УДК 007.04

А.И. ПОВОРОЗНЮК, канд. техн. наук, НТУ "ХПИ",
М.Н. ШКАРУПА

СОВЕРШЕНСТВОВАНИЕ ЗАЩИТЫ WEB-ПРИЛОЖЕНИЙ ОТ ВТОРЖЕНИЙ НА ОСНОВЕ ЭВРИСТИЧЕСКОГО ПОДХОДА

В статті розглядаються існуючі програмні методи захисту WEB-додатків у всесвітній мережі Internet та виявляються їх вади. Робиться аналіз деяких видів атак та сучасних тенденцій щодо їх проведення. Робляться рекомендації відносно подальшого найбільш перспективного напрямку розвитку програмних засобів захисту WEB-додатків.

In present article were overviewed software-based methods of WEB-application protections at global network Internet. At this report are analyzing some kinds of attacks and present-day tendencies for their implementation. Adduced recommendation for future trends evolution of software-based instruments of WEB-application protections.

Постановка проблемы. Основным средством обеспечения безопасности на современном этапе развития сети Internet является такой класс приложений как брандмауэр (часто используемые синонимы: файервол, межсетевой экран).

В самом общем случае брандмауэр можно определить как локальное или функционально-распределенное программное, аппаратное или программно-аппаратное средство, реализующее контроль над информацией, поступающей в компьютерную систему и выходящей из нее [1].

В зависимости от уровня модели OSI, на котором происходит контроль над информацией, различают брандмауэры работающие на:

– сетевом уровне, фильтрация происходит на основе адресов отправителя и получателя пакетов, номеров портов транспортного уровня модели OSI и статических правил, заданных администратором;

– сеансовом уровне, при фильтрации не пропускаются пакеты, которые нарушают спецификации стека протоколов TCP/IP, часто используемые в злонамеренных операциях (сканирование ресурсов, взломы через неправильные реализации TCP/IP, обрыв/замедление соединений, инъекция данных);

– уровне приложений, фильтрация происходит на основе анализа данных приложения, передаваемых внутри пакета.

Следует отметить, что практически все разработчики современных брандмауэров предлагают решения, которые работают на всех отмеченных выше уровнях. Однако работа большинства "классических" брандмауэров акцентируется на сетевом и сеансовом уровне [2]. Нередко функциональные возможности работы брандмауэра на уровне приложений обеспечиваются отдельным модулем, работа которого, как правило, носит общий характер и не учитывает особенностей функционирования того или иного приложения.

Брандмауэры являются необходимым элементом первой линии обороны, но "классические" брандмауэры отлично справляются лишь с атаками на сетевом и сеансовом уровне. В случае нашествия червей или сложных атак на уровне приложений с использованием постоянно открытых портов 80 (HTTP) и 443 (HTTPS) они, как правило, беспомощны. Системы обнаружения вторжений (СОВ), входящие в состав брандмауэров, используют пассивные фильтры, через которые пропускается сетевой трафик с целью выявления активности злоумышленников. Для обнаружения атак на уровне приложений они используют технологию сигнатур и выявления аномального поведения, но в большинстве случаев они эти атаки не блокируют, а только сообщают об их осуществлении. К моменту извещения администратора предотвращать масштабные повреждения системы часто бывает слишком поздно.

Анализ литературы. Невысокая эффективность СОВ и проблемы с их управлением стали настолько заметны, что в отчете "Gartner Information Security Hype Cycle", опубликованном в июне 2003 года, эти системы названы провальными [3].

В 2006 году Gartner в очередном прессрелизе [4] посоветовали использовать системы предотвращения вторжений (СПВ), которые начали предлагать традиционные производители систем брандмауэров. В отличие от систем СОВ, которые просто следят за сетью и посылают сообщения о тревоге, сетевые СПВ блокируют атаки в момент их возникновения и только потом поднимают тревогу.

Следует отметить, что СПВ хорошо себя зарекомендовали как средство комплексной защиты компьютера. Однако создать полностью безопасную среду для конкретного приложения они еще не способны.

Проблему создания безопасной среды функционирования WEB-приложений централизованно стали изучать только с 2004 года. Для этого была сформирована группа "Консорциум по проблемам безопасности WEB приложений" ("Web Application Security Consortium" group). В состав этой группы вошли ряд ведущих мировых специалистов по безопасности и разработке сетевых приложений, в том числе и ведущий разработчик популярного WEB-сервера Apache Раен Барнет (Ryan Barnett).

Уже в июне 2004 года в Internet были опубликованы первые результаты работы [6] этой группы – документ под названием "Классификация угроз" ("Threat Classification").

Издание этого документа преследовало такие цели как: определение всех известных атак на WEB-приложения, согласование терминологии, определение структурированного подхода к классификации атак.

В январе 2006 года этой же группой был опубликован документ [6] под названием "Критерии оценки брандмауэров по защите WEB-приложений" ("Web Application Firewall Evaluation Criteria").

Этот документ не имеет юридической силы и носит лишь рекомендательный характер. Он не содержит ни алгоритмов, ни каких-либо

ограничений, накладываемых на разрабатываемое программное обеспечение. Документ представляет собой набор характеристик, при помощи которых станет возможно сравнивать различные брандмауэры для защиты WEB-приложений, которые были и будут разработаны. Также документ предлагает некую общую терминологию, использование которой позволит избежать разночтений в среде разработчиков и пользователей такого класса приложений защиты.

Целью статьи является выработка рекомендаций по дальнейшей доработке СПВ с целью повышения уровня безопасности среды для функционирования WEB-приложения.

Основной раздел. Проведя детальный анализ сведений о методах осуществления атак на WEB-приложения, многие специалисты в области безопасности сходятся во мнении, что возможно создать ограниченное множество отпечатков (сигнатур) атак и при их помощи с высокой вероятностью выявлять ту или иную атаку [7].

На сегодняшний день наиболее успешным и популярным программным брандмауэром, специализирующим на защите WEB-приложений, является проект ModSecurity. Этот брандмауэр представляет собой модуль для широко распространенного WEB-сервера Apache. Модуль представляет собой гибко настраиваемый фильтр POST, GET и COOKIE параметров, передаваемых между пользователем и удаленным сервером в сети Internet.

Следует заметить, что ModSecurity требует кропотливой и точной настройки с участием специалиста по безопасности [8]. Настройки модуля по умолчанию не могут использоваться для защиты критичных ко взлому WEB-приложений (например электронных банков или электронных магазинов).

Попыткой создать наиболее полную конфигурацию для ModSecurity с учетом большинства возможных вариаций атак можно считать рекомендации, опубликованные в [9]. Это облегчило работу по настройке модуля, однако услуги специалиста для адаптации общих рекомендаций по настройке под конкретные задачи все еще нужны.

ModSecurity имеет существенный недостаток – отсутствие регулярного обновления баз сигнатур атак. К недостаткам этого модуля следует также отнести то, что он оперирует лишь жестко заданными правилами и не имеет эвристических алгоритмов для детектирования видоизмененных и предсказания новых атак. Кроме того действия ModSecurity по фильтрации содержимого не учитывают индивидуальных особенностей всех WEB-приложений, которые могут обслуживаться одним WEB-сервером.

Марсель Низамутдинов, один из ведущих российских специалистов в области безопасности, указывая на недостатки ModSecurity, в своей книге опубликовал ряд теоретических примеров успешных атак на защищенное WEB-приложение [10].

Современная инфраструктура сети Internet не позволяет регулярно и систематично обновлять базы сигнатур атак на WEB-приложения. Необходимо заметить, что даже если бы была создана централизованная оперативная система обновления баз сигнатур, то это не стало бы идеальным решением проблемы хакерских атак на WEB-приложения в долгосрочной перспективе. В современном мире компьютерной преступности хакеры все чаще стали автоматизировать свои взломы, создавая подсети компьютеров-зомби, троянских коней и червей, действия которых направлены против WEB-приложений. Молниеносность и масштабы атак с каждым днем все возрастают. И иногда обновление базы сигнатур может прийти слишком поздно, уже после того как атака состоялась.

В связи с описанными выше недостатками существующих решений по защите WEB-приложений предложим рекомендации по построению системы защиты. СОВ, как уже отмечалось выше, не в состоянии создать безопасную среду функционирования WEB-приложения. СПВ имеют большой потенциал, однако точность и эффективность их работы на современном этапе их развития вызывает множество нареканий, а значит эти приложения требуют детальной проработки и внесения коренных изменений.

Доработка СПВ может быть осуществлена в двух основных направлениях: в направлении усовершенствования сигнатурного анализатора и в направлении усовершенствования анализатора аномалий.

Метод сигнатурного анализа хорошо опробован на антивирусном программном обеспечении и сейчас внедрен в СПВ, где показывает отличные результаты по отражению известных атак. Метод анализа аномалий поведения пользователя WEB-приложения заслуживает более пристального изучения.

Большая часть атак на WEB-приложения можно распознать только на уровне приложений. На рис. 1а изображена классическая схема построения защиты WEB-приложения, которая предполагает анализ данных на уровне приложений в рамках брандмауэра. Однако брандмауэр не учитывает и не может учитывать всех особенностей функционирования защищаемого WEB-приложения и, как следствие, очень часто не может отличить действия злонамеренного пользователя от действий легитимного пользователя.

Из такой ситуации возможно два выхода: "обучить" брандмауэр всем особенностям поведения пользователей каждого защищаемого WEB-приложения или же вынести контроль на уровне приложений из рамок брандмауэра в рамки самого WEB-приложения (см. рис. 1б). Второй вариант представляется наиболее логичным и удобным.

Когда анализ на уровне приложений переносится в рамки самого WEB-приложения, он приобретает иную качественную окраску, открывается ряд дополнительных возможностей для анализа и сбора информации о потенциальном атакующем.

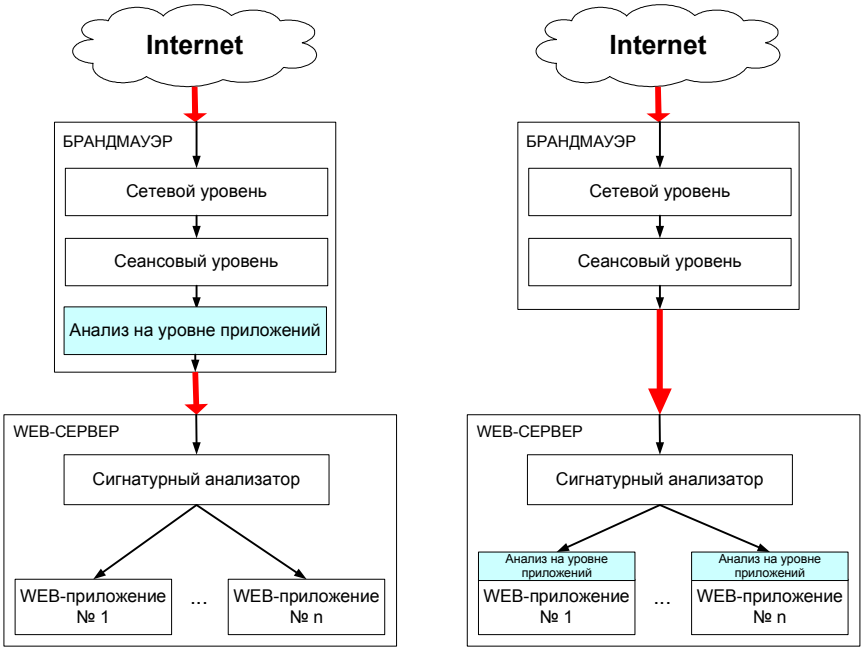

Рис. 1. Схема построения системы защите WEB-приложения
а) классическая схема, б) предлагаемая схема

В предложенной схеме анализ на уровне приложений осуществляется системой, которая состоит из набора так называемых "датчиков" и эвристического анализатора. Система "датчиков" собирает информацию в нескольких разрезах: POST-параметры, GET-параметры, COOKIE-параметры, операции с базой данных, операции с файловой системой, ошибки и предупреждения в процессе работы пользователя. Поступающая от "датчиков" информация и некоторая другая дополнительная информация о пользователе (такая как IP адрес, время начала сессии и пр.) является входной для эвристического анализатора.

Ядром эвристического анализатора является нейронная сеть адаптивно-резонансной теории (АРТ). Для решаемого класса задач наиболее всего подходит сеть АРТ-1 [11]. Главной задача, которая возложена на нейронную сеть, является задача выявления атак, которые не были выявлены на этапе сигнатурного анализа.

Нужно особо отметить, что нет необходимости обучать нейронную сеть всем известным на сегодняшний день видам атак. "Знать" существующие атаки должен сигнатурный анализатор, а эвристический анализатор должен "уметь отличать" поведение легитимного пользователя от поведения

злонамеренного пользователя. Иными словами, сигнатурный анализатор должен "уметь видеть" аномалии поведения.

Однако очень важно, чтобы эвристический анализатор не был слишком "жестким" в определении отклонения поведения пользователя от нормального и в то же время он не должен быть слишком "мягким". Эту проблему можно решить двумя способами: подбором оптимального коэффициента подобия, использования различных коэффициентов подобия для различных категорий пользователей. Первый вариант недостаточно гибкий для нашей задачи, поэтому обратимся ко второму варианту.

Выделим четыре базовые категории пользователей WEB-приложения с точки зрения безопасности: проверенные пользователи (им разрешены любые действия), обычные пользователи (доступ открыт в нормальном режиме), "подозрительные" пользователи (пользователи за которыми была замечена некоторая подозрительная активность, однако собранных данных недостаточно, чтобы определить пользователя как взломщика) и взломщики (доступ блокируется полностью). К проверенным пользователям относятся администратор системы и один или более операторов, управляющих системой. От этих пользователей не может исходить угроза и поэтому эвристический анализатор не проверяет действия таких пользователей. Действия обычных пользователей проходят эвристическим анализатором проверку в штатном режиме. Действия "подозрительных" пользователей подвергаются более жесткой проверке эвристическим анализатором. Действия взломщиков блокируются полностью.

Для реализации поставленных перед ним задач эвристический анализатор должен состоять из количества нейронных сетей АРТ-1 равного количеству защищаемых страниц WEB-приложения (в среднем до 25) и еще одной нейронной сети, которая отвечает за отнесение пользователя к той или иной категории (и соответственно, определяющая коэффициент подобия для сетей, которые защищают страницы WEB-приложения).

Перед тем как WEB-приложение будет открыто для доступа из Internet, администратор проводит обучение нейронных сетей, защищающих страницы (каждая свою страницу). Администратор активизирует режим обучения и начинает работу с приложением, стараясь инициировать "крайние ситуации", т. е. такие ситуации, когда значения какого-либо параметра (например, размер передаваемого файла, количество GET параметров в одном запросе и пр.) достигают сначала разрешенного минимума, а потом разрешенного максимума. Такой режим работы с приложением обучает нейронную сеть понятию "нормальное поведение" пользователя. Понятие "нормального поведения" для разных страниц сайта может сильно отличаться, поэтому для каждой защищаемой страницы сайта предусмотрена своя нейронная сеть.

Обученная нейронная сеть может выявлять аномалии поведения пользователей и относить последних к одной из трех категорий: обычные пользователи (не найдено соответствий среди образов, которые запомнила

сеть ранее), "подозрительные" пользователи (найдено соответствие запомненному ранее вектору в рамках коэффициента подобия) или взломщики (найдено точное соответствие запомненному ранее вектору). После окончания обучения администратор переводит сеть в рабочий режим и открывает доступ к WEB-приложению из Internet.

Система "датчиков" собирает и предает в виде двоичного вектора на вход нейронной сети некоторую информацию: количество и суммарный объем GET-параметров, количество и суммарный объем POST-параметров, количество и суммарный объем COOKIE-параметров, MIME тип переданных файлов, номер ответа из заголовка HTTP, имена затронутых таблиц в базе данных, действия проводимые с таблицами в базе данных, номера ошибок, которые возникли при работе скриптов или нуль, если таковых не возникло. Вектор сравнивается с изображениями, отложенные в памяти нейронной сети в процессе обучения, и делает вывод нормальное ли поведение пользователя, которое привело к возникновению такого вектора, или нет. Если сеть не может точно сказать, что происходит атака, однако степень отклонения от модели нормального поведения достаточно велика, то данные о потенциальном атакующем запоминаются отдельной сетью, отвечающей за отнесение пользователей к той или иной категории. В следующий раз, когда пользователь вернется на сайт и будет производить некоторые вызывающие подозрения действия, нейронная сеть классифицирует его как "подозрительного" пользователя и ужесточит анализ отклонения от нормальной модели поведения путем коррекции коэффициента подобия. Если подозрения подтвердятся, то пользователь будет переведен в категорию взломщиков, и тогда все его последующие действия будут заблокированы.

Несмотря на то, что нейронная сеть, отвечающая за отнесение пользователя к той или иной категории, и нейронная сеть, которая отвечает за защиту конкретной страницы сайта, имеют разные функциональные назначения, работают они по одному и тому же принципу. Отличие этих сетей состоит лишь в тех данных, которые они запоминают. Первая сеть запоминает данные о подозрительном пользователе (IP адрес, страна, наименование и версия браузера, наименование и версия операционной системы, язык системы, есть ли поддержка Flash, есть ли поддержка Java, разрешение экрана, глубина цвета, стартовая страница браузера), а вторая – данные о нормальном поведении пользователей системы. Анализируя выход первой нейронной сети, мы делаем вывод о необходимости повышения коэффициента подобия при проверке при помощи второй нейронной сети. Анализируя выход второй нейронной сети, мы делаем вывод о том нормальное ли поведение пользователя или же оно отклоняется от нормальной схемы. Поэтому ниже будет приведен пример функционирования лишь первой нейронной сети. Будет показано, что эта нейронная сеть способна также распознать ранее проявившего активность потенциального взломщика даже если он предпринял меры, затрудняющие его идентификацию.

В реальной нейронной сети, отвечающей за идентификацию пользователя, входной вектор достаточно длинный, мы же ради простоты изложения ограничимся вектором длинной в пять бит.

Первые два бита нашего вектора определяют наименование клиента пользователя (00 – Internet Explorer, 01 – FireFox, 10 – Opera, 11 – другой), третий бит определяет установлен ли на компьютере Flash плеер (0 – не установлен, 1 – установлен), четвертый бит определяет язык системы (0 – русский, 1 – другой) и наконец последний бит определяет тип операционной системы (0 – UNIX подобная, 1 – Windows).

Предположим, что у нас имеются данные о трех пользователях, к которым необходимо применить более жесткий эвристический анализ: $X_1 = (00001)^T$, $X_2 = (00110)^T$, $X_3 = (01111)^T$. Также у нас имеются данные о пользователе X_3, который с целью маскировки сменил свой браузер с FireFox на некий другой браузер и пытается совершить некоторые несанкционированные действия. Входной вектор для этого пользователя будет иметь вид $X_4 = (11111)^T$.

На рис. 2 изображена упрощенная схема сети АРТ-1 [11].

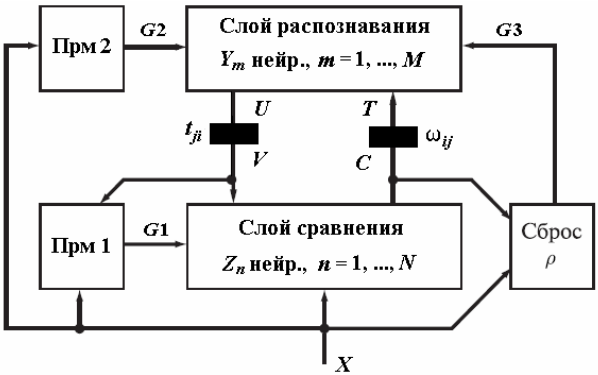

Рис. 2. Упрощенная схема нейронной сети АРТ-1

Входной вектор сети $X = (X_1, ..., X_n, ..., X_N)$ имеет N компонент. В слое распознавания запоминается M классов образов, по одному классу на каждый нейрон $m = 1, ..., M$. Основную работу по классификации производят слой сравнения и слой распознавания. Схемы приемников (Прм 1, Прм 2) и схема сброса управляют режимом работы сети и генерируют разрешающие сигналы $G1$, $G2$ и сигнал сброса $G3$ соответственно. Матрица непрерывных весов и матрица двоичных весов на рис. 2 обозначены ω_{ij} и t_{ji} соответственно.

Входной двоичный вектор X, при прохождении через сеть, претерпевает такие преобразования: $X \to C \to T \to U \to V$. Здесь C – выходной вектор слоя сравнения, T – входной вектор слоя распознавания, U – выходной сигнал слоя

распознавания, V – входной вектор для слоя распознавания и сигнал запрещения для Прм 1.

Параметр подобия возьмем $\rho = 0{,}6$. Матрицы ω_{ij} и t_{ji} инициализируются начальными значениями согласно:

$$0 < \omega_{ij} < \frac{\lambda}{\lambda - 1 + N}; \qquad \frac{\beta - 1}{d} < t_{ji} \le 1,$$

где $\lambda \in (1, 2]$; β – константа; $d > 0$.

Размерность вектора $N = 5$, примем $\lambda = 1{,}5$. Получим матрицы весов связей $\omega_{ij} = 0{,}2$; $t_{ji} = 1$, $i = \overline{1, 5}$; $j = \overline{1, 4}$.

Обучим сеть первым трем векторам. При поступлении на слой сравнения вектора X_1 на выходе слоя сравнения получаем вектор $C_1 = X_1$. На всех входах слоя распознавания имеем сигнал:

$$T_m = \sum_{i=1}^{5} \omega_{1i} C_i = 0{,}2 \cdot 0 + 0{,}2 \cdot 0 + 0{,}2 \cdot 0 + 0{,}2 \cdot 0 + 0{,}2 \cdot 1 = 0{,}2, \; m = \overline{1, M}.$$

Нейроном-победителем становится нейрон с наименьшим индексом, т.е. нейрон Y_1. Веса связей t_{1n} ($n = \overline{1, N}$) принимают значения [12]: (0, 0, 0, 0, 1).

Вычислим параметр подобия для вектора X_1:

$$S = \frac{1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 1}{0 + 0 + 0 + 0 + 1} = 1.$$

Т.к. $S > \rho$, то значит поданный на вход вектор X_1 создаст первый сохраненный в памяти образ. Соответственно будет откорректирована матрица ω_{ij}:

$$\omega_{1i} = \frac{1{,}5 \cdot 0}{0{,}5 + 0 + 0 + 0 + 0 + 1} = 0, \; i = \overline{1, 4}; \qquad \omega_{15} = \frac{1{,}5 \cdot 1}{0{,}5 + 0 + 0 + 0 + 0 + 1} = 1.$$

Далее на вход будут поданы вектора X_2 и X_3, которые также будут запомнены сетью. В результате обучения матрицы ω_{ij} и t_{ji} будут иметь вид:

$$\omega_{ij} = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0{,}6 & 0{,}6 & 0 \\ 0 & 0{,}33 & 0{,}33 & 0{,}33 & 0{,}33 \\ 0{,}2 & 0{,}2 & 0{,}2 & 0{,}2 & 0{,}2 \end{vmatrix}; \qquad t_{ji} = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{vmatrix}.$$

Атакующий $X_4 = (11111)^{\mathrm{T}}$, сменив свой браузер, пытается остаться незамеченным сетью. В этом случае $T_1 = 1$, $T_2 = 1{,}2$, $T_3 = 1{,}32$. Выбираем нейрон Y_3 с максимальным значением T. Рассчитаем для него $S = 0{,}8 > \rho$. Таким образом, вектор X_4 был правильно отнесен к третьему классу, т.е. пытавшийся остаться незамеченным атакующий все же был правильно классифицирован.

Нейронные сети, которые защищают отдельные WEB страницы сайта, работают по тому же принципу, но результат их работы интерпретируется с точностью до наоборот: если было найдено соответствие входного вектора вектору в памяти сети, то это нормальная ситуация, а если соответствия не было найдено, то, возможно, мы имеем дело с атакой и необходимо "насторожиться" и более пристально следить за текущим пользователем.

Подводя итоги, перечислим достоинства и недостатки предложенной схемы.

Достоинства предложенной схемы: система имеет способность к обнаружению новых типов атак, не требует обновления сигнатур, т.к. основана на аномалиях поведения, дает возможность отслеживать действия пользователя, неоднократно совершающего попытки взлома, полностью адаптируется под особенности защищаемого WEB-приложения.

Недостатки: зависимость от языка программирования, на котором написано WEB-положение, необходимость начального обучения нейронной сети, возможность ложных срабатываний при недостаточном периоде обучения.

Выводы. Существующие брандмауэры, предназначенные для защиты WEB-приложений, не отвечают требованиям, предъявляемых к безопасности WEB-приложения в сфере электронной коммерции. Требуются изменения схемы построение защиты, а именно, необходимо создание эффективных работающих на уровне приложения эвристических механизмов, которые будут фокусироваться на анализе аномалий поведения пользователя, что позволит выявлять атаки, с которыми не может справиться сигнатурный анализ. Несмотря на существенные недостатки метода эвристического анализа аномалий, работа, направленная на его усовершенствование, является на сегодняшний день наиболее перспективным направлением в СПВ.

Список литературы: **1.** *McGraw G.* Building Security In. – New-York: Addison-Wesley, 2006. – 448 p. **2.** *Shah S.* Web hacking. – New-York: Addison-Wesley, 2004. – 376 p. **3.** *Heiser J., Firstbrook P., Scholtz T.* Gartner Information Security Hype Cycle: http://www.gartner.com/5_about/press_releases/pr11june2003c.jsp, 2003 **4.** *Paul E., Amrit T.* Make Your IPS Work for You With Improved Tuning: http://www.gartner.com/5_about/press_releases/august2006.jsp, 2006 **5.** *Auger R., Barnett R.* Threat Classification: http://www.webappsec.org/projects/threat/v1/WASC-TC-v1_0.txt, 2004. **6.** *Robert A., Ryan B.* Web Application Firewall Evaluation Criteria: http://www.webappsec.org/projects/wafec/v1/wasc-wafec-v1.0.txt, 2006. **7.** *Bragg R., Strassberg K.* Network Security. – Osborne: Mcgraw-Hill, 2006. – 896 p. **8.** *Coar C., Bowen R.* Apache Cookbook. – Sebastopol: O'Relly, 2003. – 254 p. **9.** *Ristic I.* Apache Security. – Sebastopol: O'Relly, 2005. – 420 p. **10.** *Низамутдинов М. Ф.* Тактика защиты и нападения на WEB-приложения. – СПб.: БХВ-Петербург, 2005. – 432 с. **11.** *Заинцев И. В.* Нейронные сети. Основные модели. – СПб.: БХВ-Петербург, 1999. – 458 с. **12.** *Руденко О. Г., Бодянський Є. В.* Штучні нейронні мережі. – Х: ТОВ "Компанія СМІТ", 2006. – 404 с.

Поступила в редакцию 12.04.2007