Нейросетевые технологии в задачах обнаружения компьютерных атак

А.В. Гришин

Аннотация. В статье рассматривается методика построения обучающей модели для нейросети с целью обнаружения сетевых компьютерных атак на основе программного комплекса «Snort». Целью исследования является построение адаптивной экспертной системы, представляющей собой комплекс системы обнаружения компьютерных атак и специальной нейронной сети, позволяющий учитывать текущие особенности рассматриваемого сетевого трафика.

Ключевые слова: системы обнаружения атак, метод сигнатурного поиска, адаптивность, кластеризация, классификация, многослойный персептрон, самоорганизующиеся карты Кохонена.

Введение

В настоящее время разработано множество программных продуктов, призванных обеспечить защиту информационных систем от несанкционированных вредоносных воздействий. На *пользовательском уровне* таковыми могут являться попытки установления контроля над системой. Для *сетевого уровня* широко распространенные атаки – процессы, разрушающие элементы информационной системы либо ставящие под угрозу их функционирование [1]. Традиционно несанкционированные воздействия выявляются по таким параметрам, как общее количество соединений с контролируемой ЭВМ, параметры трафика, временной промежуток между одинаковыми запросами и т. д. Одним из самых распространенных методов, используемых в ходе контроля за безопасностью сетевой активности систем обнаружения атак (СОА), является метод сигнатурного поиска. Этот метод наиболее эффективен в случае регулярного пополнения базы знаний информацией о компьютерных атаках, однако его реализация требует большого объема потребляемых вычислительных ресурсов, при этом он не может гарантировать стопроцентной работоспособности при отсутствии регулярных обновлений со стороны пользователя [1].

В настоящее время в научных журналах опубликован ряд работ, посвященных возможности обнаружения несанкционированных воздействий на основе нейросетевых технологий. К примеру, в [2-4] изложены во многом схожие принципы построения адаптивных механизмов обнаружения аномальной активности системы. Различаются они выбором нейронных сетей различных типов, что позволило исследовать их применимость для того или иного признака вторжения, а также несколько улучшить характеристики адаптивного модуля. Основой для них послужила работа Д. Деннинг (D. Denning), опубликованная в 1987 году и ставшая прообразом для большинства нынешних моделей обнаружения вторжений [5]. Методика работы в данном случае заключается в контроле за действиями пользователя информационной системы (исследуются такие параметры, как количество запущенных приложений, векторные характеристики работы с манипулятором «мышь» и др.)

Об использовании нейросетевых технологий для обнаружения компьютерных атак на сетевом уровне заявил в 1998 году Дж. Кэннеди (J. Cannady). В [6] излагается принцип выявле-

ния сетевых атак, основанный на анализе девяти характеристик пакета данных при помощи многослойного персептрона. Основной недостаток данной методики - невозможность выявлять атаки, проводимые в несколько стадий. Чтобы решить эту проблему, была создана адаптивная система обнаружения атак, основанная на совместной работе самоорганизующихся карт Кохонена и многослойного персептрона [7], выполняющих задачи кластеризации и классификации данных. Выявление атак, проводимых в несколько этапов, стало возможным благодаря тому, что в базу данных экспертной системы вносилась информация об изменениях в поведении конкретного объекта в течение некоторого отрезка времени [8].

Другой принцип выявления компьютерных атак основан на подсчете количества соединений наблюдаемой ЭВМ с удаленными клиентами. Количество подключений на определенный момент времени сравнивается со статистическими показателями, полученными в ходе обучения нейронной сети [9]. Основой для обучаемого модуля также стал многослойный персептрон. Один из альтернативных вариантов реализации обучающего модуля был предложен в [10]. В этом случае используются нейронные сети, построенные на основе радиальных базисных функций (radial basis networks).

В [11] подробно изложен принцип построения нечеткой экспертной системы (*fuzzy expert system*), выполняющей анализ сетевых аномалий. Данная система основана на теории нечетких множеств и имеет лингвистический вид представления правил. Она предусматривает взаимодействие базы знаний с анализатором трафика, который, получая на вход информацию, производит преобразование данных в вид нечетких лингвистических значений. С использованием нечеткого механизма логического вывода данный модуль формирует характеристику события, после чего производится преобразование полученных переменных в четкие значения. Модуль постанализа собирает необходимую информацию, классифицирует событие и составляет набор характеристик сессии, в которой была обнаружена атака.

Общей чертой, которой обладает большинство вышеуказанных разработок, является узкая специализация, поскольку в каждом конкретном случае анализируются только те параметры, которым обладает только тот или иной вид несанкционированного воздействия. При этом каждая из подобных экспериментальных систем создана в виде «монолита», без возможности использования данных разработок в универсальных системах обнаружения атак и несанкционированных воздействий. В предлагаемой статье излагается методика построения адаптивного модуля, интегрированного в уже существующую систему обнаружения компьютерных атак и выполняющего нейросетевую классификацию данных, поступающих из локально–вычислительной сети. Цель работы – показать возможность динамического автоматизированного обучения нейронной сети в условиях работы реальной системы обнаружения атак (СОА).

1. Традиционные методы классификации компьютерных атак

Рассмотрим общий механизм обработки сетевых событий на примере системы обнаружения атак (СОА) «Snort» [12]. Данный программный комплекс разработан для операционных систем семейства «Unix» и использует библиотеку «Libpcap» для получения информации из локально–вычислительной сети.

На Рис. 1 показана упрощенная функциональная схема стандартной СОА на примере «Snort». Сетевые пакеты поступают на слушающий сокет модуля «Sniffer», после чего выполняются процедуры фильтрации и классификации событий. Задача «отсева» нежелательных и заведомо «бесполезных» пакетов возложена на модуль препроцессора, который представляет собой набор функций, позволяющий избежать потребления системой чрезмерно большого объема вычислительных ресурсов. Модуль фильтрации, взаимодействуя с базой правил и базой сигнатур, осуществляет классификацию событий системы.

Следует отметить, что несмотря на множество изменений и дополнений, вносимых в новые версии программного продукта, функционально СОА «Snort» базируется на двух

принципах работы. Первый, *элементарный анализ*, заключается в поиске соответствий параметров рассматриваемого сетевого трафика параметрам, заданным в базе правил программы. Второй, *сигнатурный анализ*, состоит в детальном поиске некоторых последовательностей байтов в пакетах, приходящих на слушающий сокет системы.

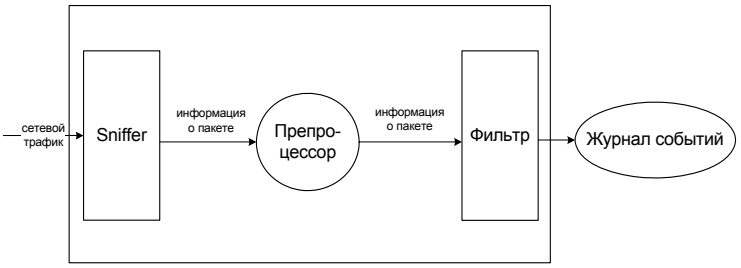


Рис. 1. Функциональная схема стандартной системы обнаружения компьютерных атак

Первый принцип основан на проверке заголовков сетевых пакетов и не предполагает анализа тела сетевого пакета в целом, является более простым и дешевым с точки зрения объема используемых системой ресурсов, а также способа его алгоритмической реализации. Например, наиболее информативной частью пакета TCP является IP–адрес отправителя (*«source IP»*), порт отправителя (*«source port»*), IP–адрес получателя (*«destination IP»*) и порт получателя (*«destination port»*). Основываясь на этих данных и сопоставляя их с набором правил собственной базы знаний, СОА способна распознавать угрозу в некотором событии. Гораздо более эффективным, но в то же время и более трудоёмким по общему количеству операций и используемых вычислительных ресурсов, является процесс сигнатурного анализа тела пакета [13]. Общая доля данных, подвергаемых сигнатурному анализу, может варьироваться в зависимости от характера поступающего на вход трафика.

При том, что СОА традиционной архитектуры стали достаточно надежным механизмом обнаружения вредоносных воздействий, в некоторых случаях информационные системы остаются уязвимыми. В частности, СОА зачастую одинаково реагирует на повторения событий, позволяя злоумышленнику действовать многократно по одной и той же схеме (к примеру, при сканировании портов). Кроме того, СОА традиционной архитектуры действуют согласно жестко заданной структуре программы и жестко заданным наборам правил – информации об атаках. При этом любое отклонение от них воспринимается системой как санкционированное событие, что является потенциальной уязвимостью системы. В этом случае необходимо добавлять новую запись в базу данных СОА, иначе попытка несанкционированного воздействия останется необнаруженной. Наконец, каким бы малым ни был период между обновлениями системы (а у СОА «Snort» он составляет приблизительно один месяц), остается некоторый промежуток времени между появлением новой атаки, выявлением ее сетевыми экспертами и добавлением информации о ней в существующие базы.

Одним из решений вышеперечисленных проблем может послужить применение нейросетевых технологий, в частности – в задаче классификации событий и кластеризации данных, поступающих на вход системы. За исключением некоторых ситуаций, оговариваемых ниже, этот подход потенциально позволяет повысить качество работы СОА.

2. Принцип действия адаптивного модуля в системе «Snort»

На Рис. 2 показана схема модифицированной системы обнаружения атак с интегрированным в нее адаптивным нейросетевым модулем, функционирование которого происходит параллельно общему функционированию системы. Эта архитектура обеспечивает автоматизацию процесса обучения с учителем.

Общие данные, поступающие на вход системы, одновременно направляются на вход линейной СОА и адаптивного модуля, после чего выходной слой нейронной сети получает желаемый отклик, выработанный в ходе непосредственного анализа трафика. В системе реализовано два режима функционирования нейронной сети: *раздельный* и *совмещенный*. *Раздельный* режим используется при нормальном (штатном) функционировании адаптивного модуля. Также он используется при начальной инициализации весов

(то есть при первичной установке численных значений для каждого из входов нейронов) [7]. Эта процедура необходима для проверки работоспособности и обеспечения сходимости при обучении. В *раздельном* режиме адаптивный модуль функционирует отдельно от общей системы. При *совмещенном* режиме функционирования корректировка весов производится параллельно работе СОА. Обучение продолжается в том случае, когда коэффициент ошибки превышает установленное значение.

Для повышения эффективности работы адаптивного модуля было предусмотрено разделение его функций между двумя подмодулями, действующими согласно общей архитектуре системы обнаружения атак, описанной в предыдущем разделе. Метод нейросетевой обработки значений на фиксированной области, соответствующий элементарному анализу, назовем методом *простой классификации*. Более трудоемкий метод, соответствующий сигнатурному анализу, назовем методом *семантической классификации* (поскольку само это определение означает семантический анализ информационной составляющей объекта данных).

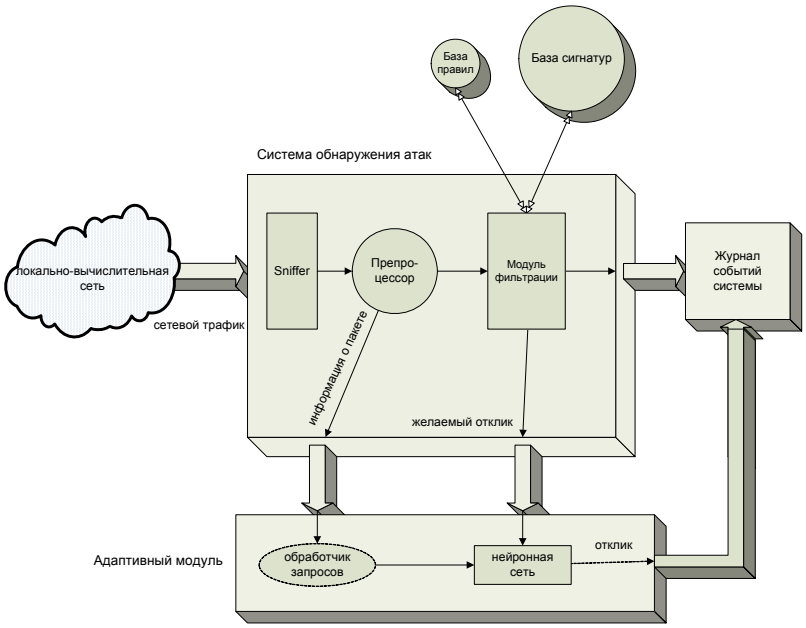


Рис. 2. Функциональная схема системы обнаружения компьютерных атак с учетом работы интегрированного адаптивного модуля

3. Подмодуль простой классификации данных

Рассмотрим характер взаимодействия системы обнаружения атак и адаптивного модуля. СОА обеспечивает параметрический поиск и принимает решение о санкционированности того или иного события, основываясь на собственном наборе правил и сигнатур. Обозначим через E множество всех возможных событий системы, через I – подмножество известных атак ($I \subset E$). Отметим, что существует два подмножества известных атак – $I_{\text{ýà}}$ и $I_{\text{ñà}}$, соответствующих методам элементарного анализа и сигнатурного анализа СОА, причем $I_{\text{ýà}} \subset I_{\text{ñà}}$ [1]. Через U обозначим подмножество неидентифицируемых атак ($U \subset E, U \not\subset I$). Каждому событию из списка $x = (x_1 x_2 \ldots, x_n)$, попадающему под понятие атаки ($x \in (I \cup U)$), ставится в соответствие некоторая уязвимость системы $v = (v_{1,} v_{2,} ..., v_p)$ из множества всех возможных уязвимостей ($v \in V$). Если множество известных атак (или угроз) рассматривать как информационное поле, и при этом возможна идентификация каждой из известных угроз при нахождении СОА в режиме работы, то заданным угрозам $x_i \in I$ в ходе работы линейно функционирующей системы ставятся в соответствие выявленные уязвимости $v_p \in P, P \subset V$.

При поступлении на вход сообщения $x_i \in U$ (не попадающего под определение угрозы, но таковой являющегося) система должна адекватно и своевременно отреагировать на данное событие.

Работа СОА в общем виде заключается в поиске определенного соответствия некоторого качественного признака в базе знаний и некоторой характеристики пакета, поступающего на слушающий сокет сетевого интерфейса. Введем определение некоторой булевой функции $f(x_i)$, условно имеющей среди возможных значений на выходе либо «1», либо «–1», а $x_i \in E$ – некоторое событие из множества

возможных событий системы, соответствующее приходу сетевого пакета на слушающий сокет сетевого интерфейса.

Разделим совместную работу подмодуля простой классификации и системы обнаружения атак на два режима, не зависящих друг от друга.

1) Режим обучения. Для сообщений $x_i \in I$ устанавливается соответствие с некоторым элементом $v_p \in P$, при этом $f(x_i)=1$. Для сообщений $x_i \in (E \setminus I)$ значение $f(x_i)=-1$. Данные в виде вектора поступают на вход нейронной сети, выходные данные сравниваются с желаемым откликом, после чего происходит итеративное обучение по алгоритму обратного распространения.

2) Режим непосредственного функционирования. Следует отметить, что лишь в случае идеального обучения нейросети для сообщений $x_i \in I$ значение вывода всегда будет равно «1», а при $x_i \in (E \setminus I)$ – «–1». На практике вероятность реагирования системы на событие $x_i \in U$ вычисляется эмпирическим путем.

Для реализации подмодуля простой классификации данных был применен многослойный персептрон традиционной архитектуры. Для базовой модели число входных элементов первого слоя составило 10, число элементов выходного слоя – 2. Для оптимизации структуры нейронной сети и повышения эффективности ее обучения был реализован метод динамического распределения элементов на основе *алгоритма Бартлетта* [14]. При обучении процесс начинается с некоторого минимума элементов, после чего нейронная сеть тренируется до стабилизации коэффициента ошибки. На каждой стадии тренировки в скрытый слой добавляется новый элемент с малым случайным весом. Процесс повторяется до тех пор, пока добавление нового элемента не приведет к увеличению стабилизировавшейся ошибки.

Поскольку обучение происходит в течение некоторого количества временных тактов, соответствующих подаче на вход нейросети нового обучающего образа, обозначим условную итерацию через i. Множество нейронов выходного слоя обозначим через C. Обучение нейросети в нашем случае базируется на алгоритме обратного распространения, основанном на коррекции ошибок. Детальное изложение процесса обучения, а также его математическое описание приведены в [7]. Уделим некоторое внимание первому, наиболее значимому шагу – процедуре поиска значения коррекции $\Delta\omega_{jk}(i)$ для веса $\omega_{jk}(i)$ выходного слоя.

Согласно [7], для выходного слоя нейронов ($j \subset C$, нейрон k предшествует нейрону j) коррекция весов выходного слоя будет равна:

$$\Delta\omega_{jk}(i) = -\eta \frac{\partial E(i)}{\partial e_j(i)} \varphi_j\left(v_j(i)\right) y_k(i), \qquad (1)$$

где η – коэффициент скорости обучения, $E(i)$ – значение общей энергии ошибки сети, $e_j(i)$ – значение ошибки нейрона выходного слоя, представляющее собой разность между полученным и желаемым откликом нейронной сети, $\varphi_j(\ )$ – функция активации, соответствующая нейрону j, $v_j(i)$ – информационное поле, полученное на входе функции активации, $y_k(i)$ – выходной сигнал нейрона k, i – событие системы (итерация обучения).

Наиболее трудно вычисляемым параметром формулы (1) является значение частной производной общей энергии ошибки сети $\frac{\partial E(i)}{\partial e_j(i)}$, причем основная сложность заключается во множестве операций, которые необходимо произвести в общем случае. Однако можно учесть, что главной особенностью архитектуры представленной нейронной сети является наличие всего двух элементов выходного слоя. При этом желаемые отклики двух выходных нейронов будут противоположны друг другу ($f_1(x_i) = -f_2(x_i)$). Согласно [7], в частном случае предполагаемой архитектуры значение общей энергии ошибки сети может быть представлено в следующем виде:

$$E(i) = \frac{1}{2}(2f(x_i)^2 - 2f(x_i)y_1(i) + 2f(x_i)y_2(i) + y_1(i)^2 + y_2(i)^2). \qquad (2)$$

Учитывая, что в поиске значений частных производных в качестве неизвестных попеременно выступают $y_1(i)$ и $y_2(i)$ (значение $f(x_i)$ примем за константу), получим следующий результат:

$$\text{для } y_1(i):\ \Delta\omega_{jk}(i) = -\eta\left(y_1(i) - f(x_i)\right)\varphi'\left(v_1(i)\right)y_1(i),$$
$$\text{для } y_2(i):\ \Delta\omega_{jk}(i) = -\eta\left(y_2(i) + f(x_i)\right)\varphi'\left(v_2(i)\right)y_2(i). \quad (3)$$

Обучение нейронной сети по методу обратного распространения происходит в последовательном итеративном режиме, что обусловлено необходимостью корректировки весов после поступления на вход нового набора данных. Это позволяет включать режим обучения адаптивного модуля СОА в ходе непосредственного функционирования, а также проверки системы.

4. Применимость метода простой классификации

В ходе обучения модуля простой классификации была проведена запись заголовков сетевых пакетов, поступающих из сегмента Интернета. Запись сессий проводилась программным обеспечением «Snort», модифицированным с целью изменения формата и направления вывода потока данных. Для обеспечения относительно полного набора данных для обучения и проверки функционирования нейросетевого модуля было решено составить массив из 250000 уникальных строк и разделить его на две части. Первая его часть, предназначенная для обучения, включила в себя 160000 строк. При этом опытным путем было выявлено, что оптимальное отношение количества «безопасных» событий системы к количеству «несанкционированных воздействий» составляет 3 к 1. Вторая часть, предназначенная для проверки функционирования нейросетевого модуля, включила оставшиеся 90000 записей, причем в данном случае соотношение количества «безопасных» событий системы и количества «несанкционированных воздействий» составило (1:1).

В ходе обучения адаптивного модуля последовательно использовались различные методики классификации атак: в первом случае СОА в качестве учителя функционировала в *режиме проверки заголовков пакетов*, во втором случае СОА функционировала в *комбинированном режиме*, совместно использующем как простую проверку заголовков пакета, так и метод сигнатурного анализа. Поскольку комбинированный режим предлагает более эффективную обработку и анализ, чем простая проверка заголовков, нам представляется возможным выполнить сравнительную проверку эффективности работы обученного нейросетевого модуля при работе учителя как в первом, так и во втором режиме. Это позволяет эмпирически выявить основные недостатки метода простой классификации и выработать стратегию по их устранению. Для проверки работы адаптивного модуля в режиме непосредственного функционирования производился количественный подсчет общего числа несоответствий отклика нейросети значению $f(x_i)$.

В Табл. 1 показаны коэффициенты соответствия – отношение количества правильных ответов адаптивного модуля к количеству ошибок после обучения нейросети.

Табл. 1. Коэффициенты соответствия. Проверка на исходном массиве данных

Обучение / проверка	Элементарный анализ	Комбинированный режим
Элементарный анализ	0,98	0,53
Комбинированный режим	0,53	0,72

В данном случае проверка проводилась на исходном массиве данных из 160000 строк. В строках первого столбца обозначены режимы функционирования СОА на момент обучения, в столбцах первой строки обозначены режимы функционирования СОА на момент проверки.

Критерий коэффициента соответствия позволяет оценить качественную составляющую работы адаптивного модуля, однако следует отметить, что эта оценка сугубо приблизительна, поскольку она включает в себя эмпирическую погрешность, которая не может быть трактована однозначно как ухудшение показателей. Отметим, что «правильным» считается тот ответ системы, который совпадает с ответом СОА, однако сам этот ответ в некоторых случаях является неверным, поскольку экспертная система, не обладая полнотой знаний, или не используя их, далеко не всегда может дать правильную оценку тому или иному событию. Достаточно малый коэффициент соответствия, полученный при обучении в комбинированном режиме и проверке в режиме простой классификации, включает в себя те случаи несовпадения с проверяемыми данными, когда СОА не обнаруживает атаку при ее наличии.

Этот факт наглядно демонстрируется относительно высоким коэффициентом, полученным при проверке в комбинированном режиме. Таким образом, исходя из полученных данных, можно утверждать, что адаптивный модуль способен запоминать состояния системы и правильно классифицировать повторяющиеся события, то есть функционально адаптивный модуль представляет собой аналог ассоциативной памяти.

В Табл. 2 показаны коэффициенты соответствия, полученные в ходе проверки системы на тестовой части массива данных с количеством событий 90000. Соотношение количества «безопасных» событий и количества «несанкционированных воздействий», как уже было упомянуто выше, здесь составляет 1:1. В строках первого столбца обозначены режимы функционирования СОА на момент обучения, в столбцах первой строки обозначены режимы функционирования СОА на момент проверки. Данный опыт показывает эффективность работы адаптивного модуля в режиме непосредственного функционирования.

Табл. 2. Коэффициенты соответствия. Проверка на тестовом массиве данных

Обучение / проверка	Элементарный анализ	Комбинированный режим
Элементарный анализ	0,96	0,52
Комбинированный режим	0,45	0,69

Как видно, коэффициенты, полученные при проверке на новом массиве в режиме простой классификации, несколько отличаются в худшую сторону. Это вполне закономерно, поскольку в тестовом массиве встречаются данные, которые существенно отличаются от тех, что использовались при обучении нейронной сети. Однако важно отметить, что полученные коэффициенты остаются практически неизменными (три опыта, проведенных на разных векторах данных высокой размерности, показали, что значение всех коэффициентов колеблется в пределах 0,005). Это говорит о стабильности работы данной методики на разрозненных данных.

Таким образом, можно сделать вывод, что нейросетевой подход оправдан в случае его применения в области задач классификации заголовков сетевых пакетов. Для эффективного анализа данных нейросетевым модулем необходимо обеспечить поступление на его вход векторов данных $x_{1,}x_{2,}...,x_i$ ($i \in I$, где I – множество событий, или обучающих выборок) размерности N, при этом каждый из векторов должен иметь определенное фиксированное количество элементов, а каждый из них должен соответствовать своему функциональному назначению (другими словами, поступающие образы не могут быть инвариантными). Данная реализация адаптивного модуля (с учетом условия сходимости при обучении, работы на сходном с обучающим наборе данных, а также условий, приведенных выше) позволяет с некоторой степенью риска заменить функционал СОА на повторяющихся наборах данных. Более того, адаптивный модуль, реализованный в виде многослойного персептрона, представляет собой аналог элемента ассоциативной памяти (адаптивная модель способна запоминать состояния, выявляемые обучающей системой методом сигнатурного анализа, что позволяет классифицировать некоторые события $x_i \in U$ как атаку). Однако эффективность работы адаптивного модуля падает в том случае, если на вход нейросетевого модуля поступает набор векторов данных с высокой степенью разобщенности. Кроме того, поведение адаптивного модуля при непосредственном функционировании трудно прогнозировать, вследствие чего необходимо определить надлежащую степень ответственности и обеспечить возможность «перестраховки» и сравнения искомого и полученного результатов в ходе параллельного функционирования СОА и ее надстройки в виде нейросетевой экспертной системы. Также следует отметить, что эффективность метода простой классификации достаточно мала по сравнению с показателями метода сигнатурного анализа, основанного на обработке данных тела сетевого пакета.

5. Подмодуль семантической классификации; постановка задачи

В силу того, что многие важные события оказываются вне поля зрения системы, было решено внедрить в нее дополнительный адаптивный модуль, на который была возложена за-

дача обработки массива данных тела сетевого пакета, приходящего на слушающий сокет СОА. В качестве инструментария для реализации модуля был выбран алгоритм самоорганизующихся карт SOM [15]. Метод семантической классификации, призванный создать аналог метода сигнатурного анализа при распознавании атак, позволяет повысить точность распознавания поступающих образов входного трафика. Однако в связи с тем, что размер пакетов транспортного уровня жестко не фиксирован, возникает проблема выбора оптимального количества элементов нейросети. Зачастую размер TCP-пакета вместе с заголовком ограничивается сотней байтов, однако в сетевом трафике встречаются пакеты, размерностью превышающие это значение более, чем в сто раз. Высокая степень потребления вычислительных ресурсов нейронными сетями не позволяет построить полноценную модель сопоставимой размерности для обработки данных высоких порядков, что говорит о необходимости упрощения вводимой информации, с одной стороны, и повышения её ценности – с другой.

Одним из преимуществ метода сигнатурного анализа при поиске атак является отсутствие линейной индексной привязанности к тому или иному проверяемому элементу, поскольку зачастую искомый набор символов может менять свое местоположение.

На листингах 1 и 2 показан случай повторения одной и той же атаки («EXPLOIT ssh CRC32 overflow»), зафиксированной СОА «Snort» с периодичностью в полсекунды. Ниже приведены два массива данных, в шестнадцатеричном виде представляющих фрагменты тела сетевого пакета. Курсивом выделены искомые фрагменты кода (сигнатуры). При помощи фильтра в записанной сессии было выбрано несколько событий, когда СОА, основываясь на базе знаний, идентифицировала пакет как атаку, однако искомая последовательность символов располагается с варьирующимся смещением.

В первом случае искомый фрагмент сетевого пакета, содержащий в себе сигнатуру атаки, начинается с 87 порядкового элемента (байта), однако в событии, показанном на листинге 2, первый элемент искомой сигнатуры встречается гораздо раньше (порядковый элемент 50). В обоих случаях СОА, выступая в качестве обучающей системы, приняла верное решение (данный пакет отнесен ко множеству несанкционированных воздействий), однако статичная нейронная сеть не сможет адекватно отреагировать на появление входного вектора, по своим характеристикам полностью отличающегося от «эталонного», использующегося при процессе корректировки весов.

Листинг 1. Фрагмент тела сетевого пакета. Атака: EXPLOIT ssh CRC32 overflow (сигнатура | 00 01 | W | 00 00 00 18 |). Искомый фрагмент кода обнаружен на 87 порядковом байте.

1	00	*5E*	40	98	8F	00	03	0D	38	34	*C2*
08	00	45	00	00	93	*E6*	24	00	00	*FF*	11
16	8C	0A	01	2C	58	*EF*	*C0*	98	8F	1A	73
1A	73	00	7F	97	*CE*	42	54	2D	53	45	41
52	43	48	20	2A	20	48	54	54	50	2F	31
2E	31	0D	0A	48	6F	73	74	3A	20	32	33
39	*2E*	31	39	32	*2E*	31	00	01	*2E*	31	34
33	3A	**00**	**01**	**57**	**00**	**00**	**00**	**18**	6F	72	74
3A	20	34	30	0	0	0	0	0	0	0	0

Листинг 2. Фрагмент тела сетевого пакета. Атака: EXPLOIT ssh CRC32 overflow (сигнатура | 00 01 | W | 00 00 00 18 |). Искомый фрагмент кода обнаружен на 51–м порядковом байте.

00	*1E*	*BE*	*FF*	*D0*	*B4*	00	1D	60	80	4A	22
88	64	11	00	7C	96	00	42	00	21	45	00
00	40	*BA*	51	40	00	40	11	58	78	*D5*	8D
9E	31	*D4*	01	*E0*	22	*CD*	8B	63	19	*AF*	56
01	00	**00**	**01**	**57**	**00**	**00**	**00**	**18**	00	00	00
00	00	06	*6E*	6F	76	65	6C	6C	3	31	31
32	03	00	01	37	03	*6E*	65	74	00	00	01
00	01	00	00	00	00	00	00	00	00	00	00

Таким образом, в задаче поиска компьютерных атак элементы входных данных отображают нестатические свойства. Сигналы, поступающие на слушающий сокет, имеют динамический характер, хотя их элементы по большей части взаимозависимы. Следовательно, для успешной и быстрой обработки входных данных необходимо уменьшить их размерность, разложить параметры входного вектора по глобальным признакам и придать инвариантность системе, то есть выполнить операцию предобработки.

6. Предобработка и представление входных данных

Представим данные, описывающие классы образов, в виде аналитически определяемых многообразий. Таким образом, по терминологии линейной алгебры, некоторые группы преобразований можно осуществить в автоматическом режиме при условии, что рассматриваемые данные принадлежат к классам линейных подпространств [15]. Если считать, что подпространство определяется базисными векторами, в качестве которых выбраны векторы представления образов, то входной образ может быть классифицирован в соответствии с его сходством с определенной общей линейной комбинацией базисных векторов.

Имея на входе разобщенный массив данных, мы можем выделить из него некоторые, наиболее характерные признаки, что достигается описанием используемых входных данных через собственные векторы. Если наборы значений каждого события x_i, поступающие на слушающий сокет сетевого интерфейса, рассматривать как вещественные векторы (при этом все линейные комбинации множества I векторов x_i образуют линейное подпространство), то для каждого из таких векторов возможно посчитать корреляционную матрицу [15]:

$$\tilde{N}_{\delta\delta} = \frac{1}{N}\sum_{t\in I} x(i)x^T(i), \qquad (4)$$

где I – множество индексов (по сути являющееся множеством событий системы), i – событие системы, N – количество событий системы.

Если собственные векторы матрицы C_{xx} обозначить через $u_k, k = 1,2,...,n$ (u_k – вещественные векторы n–мерного пространства), то любой вектор x', поступающий на вход системы, можно разложить в ряд по этим векторам [15]:

$$x' = \sum_{k=1}^{p}(u_k^T x)u_k + \varepsilon, \qquad (5)$$

где число $p \le n$, а ε – остаточный член.

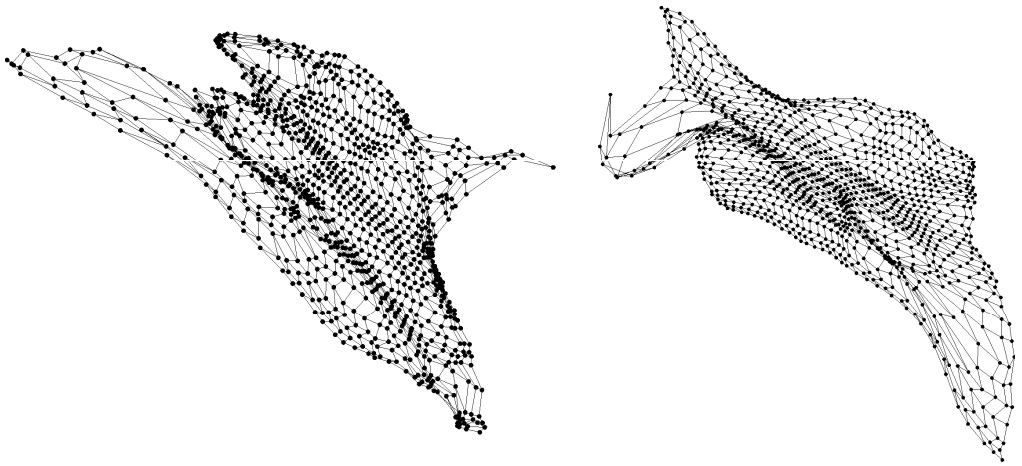
Для того чтобы получить приближенное представление входного вектора, можно использовать ограниченный набор наибольших членов разложения $u_k^T x$, что по сути является сжатым описанием с некоторой статистической точностью.

В дальнейшем кластеризация проходит довольно благополучно, большинство узлов решетки распределяется равномерно по всему периметру карты, однако при этом в режиме непосредственного функционирования адаптивный модуль показывает низкий коэффициент соответствия – 0.45. По всей видимости, связано это, в первую очередь, с тем, что при внесении дополнительных значимых свойств в рассматриваемые данные при кластеризации резко падает масштаб карты, поскольку в процессе группировки узлы сосредотачивают в себе глобальные признаки векторов, не акцентируя внимание на более мелких, но от этого не менее значимых. Таким образом, для повышения эффективности метода семантической классификации необходимо производить предварительное разделение сетевых пакетов по количеству байтов в их теле.

Главный недостаток описанного выше метода заключается в необходимости использования довольно трудоемкого алгоритма нахождения собственного вектора матрицы данных (как один из самых «быстрых», был выбран метод Данилевского [16]), который необходимо применять не только в процессе многоступенчатого формирования самоорганизующейся карты, но и при каждом поступлении нового сетевого пакета на вход СОА.

Для экономии временных затрат, с некоторым допуском погрешности, был применен простейший вейвлет-метод сжатия информации, основанный на вычислении полусумм и полуразностей. В массиве данных вычисляются средние величины соседних пар, которые рассматриваются как крупномасштабное представление входного образа. Одновременно с ними считаются полуразности этих же пар, которые используются для восстановления менее существенных деталей и заменяются нулями по прошествии нескольких итераций, в случае, если их значение становится достаточно малым. Подробно данный метод описан в [17].

Применение вейвлет–метода сжатия информации позволяет уменьшить размерность матрицы данных и сократить время на их обработку (в общей сложности удалось достичь сокращения времени на 40%).



а) б)

Рис. 3. Формирование самоорганизующейся карты на разных этапах
(а) 10000 шагов обучения, (б) 50000 шагов обучения

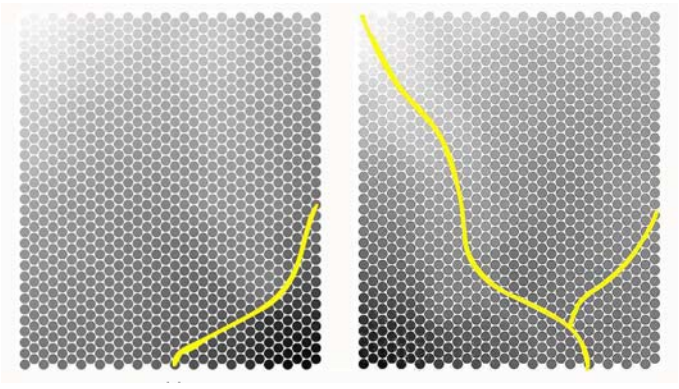
7. Применимость метода семантической классификации

На Рис. 3 показано графическое представление самоорганизующейся карты размерностью 32×32 после нескольких этапов обучения, полученное с помощью отображения Сэммона.

Для построения карты на вход адаптивного модуля было направлено 100000 векторов данных одинаковой размерности, при этом соотношение повторений к общему количеству событий составило 2:5. Как можно видеть из графического представления на Рис. 3(а), основа структуры карты была сформирована уже на шаге 10000. Существенных различий между отображениями карт, полученных после 50000 итераций и после 100000 итераций, не наблюдается. «Отростки» от основной плоскости в данном случае образуются из–за присутствия в выборке сетевого трафика некоторых пакетов, не характерных для общей массы. «Повернутый» вид карты объясняется тем, что компоненты входных векторов существенно отличаются по диапазонам изменения. Ориентация улучшается при добавлении взвешенного евклидова расстояния в уравнение, которое описывает процесс обучения [15].

На Рис. 4 показаны кластерные карты компонентных плоскостей, полученные после 10000 итераций и 50000 итераций соответственно. Значения компонент здесь отображены с помощью оттенков серого цвета (более темные оттенки обозначают скопление элементов наилучшего соответствия), примерные границы решений выделены линиями.

Подача обучающих векторов происходила по такому же принципу, как и в случае обучения модуля простой классификации, за исключением размерности и характера данных, поступающих на вход. При обучении адаптивного модуля в двух случаях последовательно использовались различные методы классификации атак: в первом случае СОА функционировала в режиме сигнатурного поиска по телу пакетов, во втором – в комбинированном режиме, совместно использующем оба метода.



а) б)

Рис. 4. Кластерная карта событий системы

(а) после 10000 шагов обучения
(б) после 50000 шагов обучения

Для проверки работы адаптивного модуля в режиме непосредственного функционирования производился количественный подсчет общего числа несоответствий отклика нейросети значению $f(x_i)$.

В Табл. 3 показаны коэффициенты соответствия – отношение количества правильных ответов адаптивного модуля к количеству ошибок после построения самоорганизующейся карты. В данном случае проверка проводилась на исходном массиве данных из 160000 строк (проверка проводилась на том же самом массиве, на котором производилось и обучение). В строках первого столбца обозначены режимы функционирования СОА на момент обучения, в столбцах первой строки обозначены режимы функционирования СОА на момент проверки.

Табл. 3. Коэффициенты соответствия. Проверка на исходном массиве данных

Обучение / проверка	Сигнатурный анализ	Комбинированный режим
Сигнатурный анализ	0,75	0,69
Комбинированный режим	0,72	0,80

В Табл. 4 показаны коэффициенты соответствия – отношение количества правильных ответов адаптивного модуля к количеству ошибок после построения самоорганизующейся карты. В данном случае, аналогично методу простой классификации, использовался тестовый массив данных из 90000 строк. Отношение количества «безопасных» событий к количеству «несанкционированных воздействий» составило 1:1. В строках первого столбца обозначены режимы функционирования СОА на момент обучения, в столбцах первой строки обозначены режимы функционирования СОА на момент проверки.

Табл. 4. Коэффициенты соответствия. Проверка на тестовом массиве данных

Обучение / проверка	Сигнатурный анализ	Комбинированный режим
Сигнатурный анализ	0,73	0,65
Комбинированный режим	0,72	0,79

Следует отметить, что коэффициенты, указанные в Табл. 4, получены при проверке массива данных приведенной размерности, подвергнутого сжатию. Эквивалентность значений, полученная при проверке на двух массивах данных в случае обучения в комбинированном режиме и проверки в режиме сигнатурного анализа, косвенно свидетельствует о наличии некоторого гарантированного порога стабильности правильного отклика модуля семантической классификации.

Заключение

В статье изложена методика построения нейросетевого адаптивного модуля, функционально дополняющего систему обнаружения компьютерных атак. В рамках исследовательской работы была построена комплексная экспертная система, подразумевающая режим автоматического обучения с учителем нейронной сети, а также режим ее автономного функционирования. По характеру обрабатываемых данных задача классификации событий была разделена соответственно между двумя подмодулями, выполняющими элементарный анализ заголовка сетевого пакета и семантический анализ его наполнения. При тестировании адаптивного модуля, обученного и функционирующего в режиме проверки заголовков пакетов на исходном массиве данных, был получен высокий коэффициент соответствия - 0,98. Кроме того, в ходе эксперимента было проведено сравнение показателей разных режимов проверки и обучения нейросети. Полученные данные позволили выявить, что адаптивный модуль способен запоминать состояния системы, однако для режима проверки содержимого сетевых пакетов коэффициент соответствия составил только 0,72. Таким образом, экспериментально было показано, что при выполнении задач элементарного анализа адаптивный модуль способен сохранять информацию о состояниях системы и с некоторой степенью риска функционально заменять собой систему обнаружения атак. Однако эффективность выявления несанкционированных событий адаптивным модулем достаточно мала по сравнению с эффективностью аналогичного сигнатурного метода, основанного на обработке данных тела сетевого пакета. Для решения этой проблемы был введен дополнительный адаптивный модуль. Метод семантической классификации, призванный соз-

дать аналог метода сигнатурного анализа при распознавании атак, позволяет повысить точность распознавания поступающих образов входного трафика до 0,8.

Для случая семантического анализа тела пакета были предложены методы предобработки входных данных, позволяющие сократить их размерность и придать свойство инвариантности входному образу.

Полученные результаты могут быть использованы для повышения качества систем обнаружения компьютерных атак и придания им свойства адаптивности. На практике было показано, что вышеизложенная методика показывает стабильную работу при статичном использовании системы в условиях определенного сетевого сегмента. Это обусловлено тем, что характер данных, поступающих на некоторый узел локально-вычислительной сети, показывает единообразие как при элементарном анализе и простой классификации, так и при сигнатурном анализе и семантической классификации. Вместе с тем важным недостатком данной методики является отсутствие универсальности ее применения. Для успешного функционирования системы обнаружения атак, построенной по данной идеологии, необходимо проводить длительное обучение и последующую проверку адаптивного модуля, что зачастую может не сочетаться с техническими требованиями общих систем защиты, установленных на реальных объектах информатизации. Для преодоления этого недостатка необходимо усовершенствовать методику обучения и проверки работы адаптивного модуля на статичном узле локально–вычислительной сети. Также одним из главных направлений в развитии методики является улучшение показателей работы нейронной сети в режиме семантической классификации, достигаемое за счет применения более совершенных методов предобработки входных образов.

Литература

1. Лукацкий А.В. Обнаружение атак / А.В. Лукацкий – изд. 2–е, перераб. и доп. – СПб.: БХВ–СПб., 2003. – 608 с.
2. Ryan J., Lin Meng–Jang, Miikkulainen R. Intrusion Detection with Neural Networks // Proceedings, Advances in Neural Information Processing Systems 10 (1997). – Cambridge: MIT Press, 1997. С. 943–949.
3. Botha M., von Solms R. The utilization of trend analysis in the effective monitoring of information security. Part 1: the concept // Information Management & Computer Security, Volume 9, Issue 5. – MCB UP Ltd, 2001. С. 237 – 242.
4. Веселов В.В., Елманов О.А., Карелов И.Н. Мониторинг информационных систем на основе нейросетевых технологий // Нейрокомпьютеры и их применение, кн. 27: Нейрокомпьютеры в информационных и экспертных системах. – М.: Радиотехника, 2007. С. 39 – 41.
5. Denning D. E. An intrusion detection model // Proceedings, IEEE Transactions on Software Engineering – Special issue on computer security and privacy, Volume 13 Issue 2, 1987. С. 222–232.
6. Cannady J. Artificial Neural Networks for Misuse Detection // Proceedings, National Information Systems Security Conference (NISSC'98), October, Arlington, VA, 1998. С. 443–456.
7. Хайкин С. Нейронные сети. Полный курс / Саймон Хайкин; пер. с анл. – М.: Вильямс, 2006. – 1103 с.
8. Bivens A., Palagiri C., Smith R., Szymansky B., Embrechts M. Network–Based Intrusion Detection Using Neural Networks // Proc. Intelligent Engineering Systems through Artificial Neural Networks ANNIE–2002, St. Louis, MO, Volume 12. – New York: ASME Press, 2002. С. 579–584.
9. Васильев В.И., Хафизов А.Ф. Нейронные сети при обнаружении атак в сети Internet (на примере атаки SYNFLOOD) // Нейрокомпьютеры и их применение, кн. 27: Нейрокомпьютеры в информационных и экспертных системах. – М.: Радиотехника, 2007. С. 34 – 38.
10. Hofmann A., Sick B. Evolutionary Optimization of Radial Basis Function Networks for Intrusion Detection // Proceedings, International Joint Conference on Neural Networks (Volume 1), 2003 . С. 415 – 420.
11. Kim Jung–Sun, Kim Dong–Geun, Noh Bong–Nam. A fuzzy logic based expert system as a network forensics // Proceedings, IEEE International Conference on Fuzzy Systems, 25–29 July 2004 (Volume 2), 2004. С. 879–884.
12. Бил Д. Snort 2.1. Обнаружение вторжений / Джей Бил; пер. с англ. под. ред. А. П. Караваева. – М.: Бином, 2006. – 655 с.
13. Норткат С., Новак Д. Обнаружение нарушений безопасности в сетях / Стивен Норткат, Джуди Новак; пер. с англ. под ред. В.С. Иващенко. – М.: Вильямс, 2003 – 448 с.
14. Bartlett P.L. For valid generalization, the size of the weights is more important than the size of the network // Advances in Neural Information Processing Systems 9. – Cambridge: MIT Press, 1997. С. 134–140.
15. Кохонен Т. Самоорганизующиеся карты / Тойво Кохонен; пер. 3–го англ. изд. – М.: Бином, 2008. – 655 с.
16. Березин И.С., Жидков Н.П. Методы вычислений. Т.2. / Березин И.С., Жидков Н.П.– М.: Наука, 1967.–305с.
17. Сэломон Д. Сжатие данных, изображений и звука / Дэвид Сэломон; пер. с англ. – М.: Техносфера, 2004. – 365 с.

Гришин Алексей Васильевич. Младший научный сотрудник ФГУП «Конструкторское бюро полупроводникового машиностроения». Окончил Московский государственный университет приборостроения и информатики в 2006 году. Автор пяти научных работ. Область научных интересов: защита информации, криптография. E–mail: gab@land.ru.