УДК 004.056 DOI 10.18522/2311-3103-2016-9-5871

Е.С. Абрамов, Я.В. Тарасов, Е.П. Тумоян

НЕЙРОСЕТЕВОЙ МЕТОД ОБНАРУЖЕНИЯ НИЗКОИНТЕНСИВНЫХ АТАК ТИПА «ОТКАЗ В ОБСЛУЖИВАНИИ»

Представлены результаты разработки метода обнаружения низкоинтенсивных атак типа «отказ в обслуживании» на http-сервисы. Используется модель представления низкоинтенсивной атаки в виде упорядоченного по времени ряда событий с аддитивным наложением атакующего воздействия и легального трафика. Такое представление позволяет задействовать математический аппарат обработки сигналов, включая методы распознавания образов. Задача разработки метода обнаружения низкоинтенсивных атак сформулирована как последовательное решение задач выделения гомогенных групп (patterns) временного ряда на основании моделей распознавания образов (pattern recognition), и последующего построения для каждой группы отдельной модели прогнозирования. С учетом контекста решаемой задачи (требования на высокую точность классификации, скорость формирования моделей и скорость классификации) определено, что наиболее перспективным направлением решения является использование комбинированных нейросетевых моделей, выполняющих на первом этапе кластеризацию, а затем прогнозирование временного ряда внутри установленного кластера. Непосредственно для обнаружения атаки необходимо выявлять факт периодического появления однотипного набора пакетов во входящем трафике и затем определять принадлежность этого набора к определённому классу (нормальному или аномальному). Порядок следования пакетов при этом ведущей роли не играет, временная информация учитывается при разбиении входящего трафика на окна. Метод включает следующие шаги: 1) для каждого защищаемого сервиса построить отдельную гибридную ИНС; 2) Для отдельного сервиса получить набор пакетов, число которых определено величиной окна (экспериментально установленное значение); 3) сформировать вектора для шага снижения размерности (самоорганизующейся карты); 4) снизить размерность входных данных путём кластеризации вектора при помощи SOM; 5) построить вектора для MLP, в которых каждый компонент соответствует номеру кластера, которому принадлежит пакет. Т.о. входным вектором будет набор кластеризованных пакетов, сохраняющий информацию о порядке (последовательности) поступления. Для пакетов уже определена принадлежность к определённому типу; 6) проанализировать вектора на MLP, классифицировать выявленные в трафике набор, разделив их на два класса – атака или норма.

Обнаружение атак; низкоинтенсивные DDoS-атаки; отказ в обслуживании; искусственная нейронная сеть; гибридная нейронная сеть; безопасность вычислительных сетей.

E.S. Abramov, Y.V. Tarasov, E.P. Tumoyan

LOW-RATE DDOS ATTACK DETECTION USING HYBRID NEURAL NETWORK

This article presents the results of the method of detecting the low-rate DDoS-attacks on http-services. A low-rate attack model in the form of a chronological series of events with an additive superposition of attack and normal traffic is used. Such presentation allows using a mathematical signal processing apparatus, including methods of pattern recognition. The task of developing a method of detection of low-rate attacks is formulated as allocation of homogeneous groups (patterns) of the time series, based on pattern recognition models, and the subsequent construction of prediction model for each separate group. Taking into account the context of the problem being solved (the requirements of the high classification accuracy, the rate of formation of models and classification rate), the most promising direction of the solution is the use of combined neural network models, performing clustering at the first stage and then forecasting time series within the specified cluster. To detect the attacks the fact of the periodic appearance of the same type of packet's set in the incoming traffic should be identified and then the membership of the set to a certain class (normal or anomaly) should be determined. The sequence of packets is not important, the time information is taken into account in the division of the incoming traffic to the window. The method includes the following steps: 1) for each protected service a separate hybrid ANN is built; 2) for each service a set of packets is received, the number of which is determined by the size of the window (experimentally set value); 3) create a vector for reducing the dimension (by self-organizing map); 4) reduce the dimension of the input vector data by clustering using SOM; 5) create a vector for MLP, in which each

component corresponds to the number of the cluster to which the packet belongs. Thus input vector is a set of clustered packets and it stores information about the order (sequence) of their receipt. The belonging to a certain type is already identified for all packets; 6) vectors are analyzed by the MLP to classify all the identified sets of traffic, dividing them into two classes: attack or norm.

Attack detection; low-rate DDoS-attacks; denial of service; artificial neural network; hybrid neural network; computer network security.

Введение. Существующие методы обнаружения DoS-атак, основанные на статистическом анализе пороговых значений, позволяющие эффективно распознавать атаки транспортного уровня (SYN-флуд, UDP-флуд и другие), малоэффективны для обнаружения низкоинтенсивных DoS-атак (Low-Rate DoS) прикладного уровня. Это обуславливает актуальность разработки новых механизмов обнаружения низкоинтенсивных атак прикладного уровня типа «отказ в обслуживании» в компьютерных сетях при помощи методов искусственного интеллекта.

1. Общие положения. Для понимания разработанного метода обсудим связь шагов метода с фактами предметной области. Характерными представителями рассматриваемого класса DoS-атак являются атаки RUDY, SlowLoris и HTTP-flood. В результате анализа сценариев низкоинтенсивных атак, приведённого в [1], было показано, что для рассматриваемых сценариев атак характерным является наличие определенных пакетов, повторяющихся с определенной частотой по малой временной шкале. Все сценарии атак характеризуются следующими основными особенностями:

- генерация небольшого по объёму периодического трафика;
- однотипность элементов трафика, составляющих атакующее воздействие;
- невозможность отнести отдельный запрос или сетевой пакет к аномалиям.

Время между отправкой пакетов меньше времени ожидания окончания соединения, однако нельзя точно установить определенный период между отправкой пакетов, поскольку значение ожидания окончания соединения является настраиваемым параметром приложения, а очередной пакет должен прийти в любой момент из интервала ожидания.

Для обнаружения атаки необходимо выявить факт периодического появления некоего однотипного набора пакетов во входящем трафике и затем определить принадлежность этого набора к определённому классу (нормальному или аномальному). Порядок следования пакетов при этом ведущей роли не играет, временная информация учитывается при разбиении входящего трафика на окна.

Свойствами, позволяющими определить наличие в наблюдаемом трафике низкоинтесивной атаки, остаются следующие (по степени значимости):

1. Полезная нагрузка протокола HTTP.
2. Последовательность (порядок следования) пакетов, поступающих на сетевой узел.
3. Дельта временной метки между соседними пакетами.

Анализ первого свойства позволяет определить тип пакета, анализ второго - принадлежность пакетов к определённому сценарию.

Можно сказать, что формальная задача разработки метода заключается в разработке классификатора $\zeta(\dots)$, который выдает метки классов (атакующее воздействие и легитимные данные) для временного ряда $e_{i-Z,\dots}e_{i-1}, e_i$:

$$\zeta\left(e_{i-Z,\dots}e_{i-1}, e_i\right) = \begin{cases} C^s : s_i^{'} \neq 0 \\ C^d : d'_i \neq 0 \end{cases} \quad (1)$$

где Z – длина истории событий, а событие e_i представляет собой вектор атрибутов события т.е.

$e_i =< x_j^i : j = 0..V-1 >$, *V – количество рассматриваемых атрибутов*

Таким образом задача сводится к классификации многомерных временных рядов. Как следует из работы [2], для решения поставленной задачи регрессионные и авторегрессионные модели и методы (а также их векторные аналоги – т.е.

вариации моделей предназначенные для работы с многомерными временными рядами) не подходят. Из рассмотренных моделей наиболее перспективными представляются нейросетевые модели, а также комбинированные модели с базовой моделью на основе нейронных сетей [1, 3, 4].

2. Обоснование выбора нейросетевой модели классификации. В настоящее время существует и активно используется класс нейросетевых моделей предназначенный для решения задач классификации временных рядов представленных непосредственно в виде последовательности атрибутов событий (т.е. собственно временного ряда) – это так называемые рекуррентные нейронные сети, например, сеть Джордана, сеть Хопфилда и их варианты [5] (гл. 15).

Однако данные архитектуры сетей имеют ряд особенностей, делающих их непригодными или нежелательными для решения поставленной задачи:

1. Рекуррентные нейронные сети склонны к переобучению. Действительно, рассмотрим в качестве примера сеть Хопфилда. Мощность множества весов многослойного персептрона равна $P = \sum_{l=2}^{N} |U_{l-1}| * |U_l|$, где N – число слоев нейронной сети, а U_l – множество нейронов слоя l, мощность множества весов сети Хопфилда с аналогичным числом слоев и нейронов равна $P = \sum_{l=2}^{N} |U_{l-1}| * |U_l| + |U_l^c| * 2$, где U_l^c - количество контекстных нейронов слоя l, обычно оно равно числу нейронов слоя l. Увеличение количества оптимизируемых параметров ведет к переобучению любой искусственной сети, даже при наличии механизмов защиты от переобучения – например, регуляризации весов.

2. Большой размер обучающей выборки. Решение проблемы переобучения может быть достигнуто путем увеличения обучающей выборки за счет согласованных (в смысле решаемой задачи), но не мультиколлинеарных обучающих векторов. Необходимый размер обучающей выборки может быть оценен исключительно эмпирически (работы по вычислению размера обучающей выборки в зависимости от архитектуры сети представляются необоснованными). Однако можно указать, что это количество будет примерно кратно увеличению мощности множества весов. В условиях решаемой задачи сбор дополнительных обучающих векторов представляется нежелательным (или даже невозможным).

3. Низкая точность классификации или предсказания на коротких последовательностях событий с высоким уровнем шума. В частности в работе [6] показано, что даже с учетом улучшений, предложенных авторами, ошибка предсказания многомерных временных рядов рекуррентными нейронными сетями составляет составляет около 40 %.

Другим подходом к классификации временных рядов является выделение признаков на скользящем окне и обучение классификатора непосредственно для окна [5] (гл.13). Поясним данную идею. Пусть имеется ряд $E = \{e_i\}$, $i = 0..N-1$, где N – длина временного ряда. Тогда временной ряд разделяется на участки (окна) B, каждое окно представляет собой вектор определенной длины, длина окна имеет смысл длины истории событий Z (см. формулу 1).

1. $k = 0$
  2. for i = 0:N-1:
    3. $B^k_{i \bmod Z} = e_i$
(алгоритм 1)
  4. if i mod Z == Z-1:
    5. k = k + 1

Однако, для решения поставленной задачи основной проблемой при применении данного подхода является тот факт, что элемент классифицируемого временного ряда является вектором, т.е. $e_i = < x_j^i : j = 0..V-1 >$, V – количество рассматриваемых атрибутов. Полученные в результате «развертывания» многомерного временного ряда

вектора будут иметь B значительную размерность $|B_k| = \mathrm{Z} \cdot \mathrm{V}$, что влияет на вычислительную сложность метода в фазе обучения, ведет к сверхпотреблению памяти в фазе классификации и общей деградации производительности в обеих фазах.

3. Снижение размерности. Для решения данной частной проблемы предлагается использовать методы снижения размерности данных и кластеризации.

Рассмотрим возможности применения методов снижения размерности данных. Пусть имеется выборка объектов $\mathrm{E} = \{\mathrm{e_i}\}$, $i = 0..N-1$ (возможно упорядоченная, но не обязательно), каждый объект $\mathrm{e_i} \in \mathbb{R}^V$, где V – размерность пространства признаков. Тогда задача уменьшения размерности состоит в получении трансформирующего преобразования Ψ для представления этой выборки в пространстве меньшей размерности $R = \{\mathrm{r_i}\}$, $i = 0..N-1$, где $\mathrm{r_i} \in \mathbb{R}^y$, причем обычно y<<V при сохранении необходимых статистических свойств исходного пространства. Такое несколько расплывчатое определение связано с многообразием критериев выбора трансформирующего преобразования.

В рассматриваемой задаче V равно размерности пространства признаков, а y=1 – поскольку целью является получение из многомерного временного ряда одномерного временного ряда.

В работе [7] выделяется два типа методов понижения размерности:

- линейные – в которых преобразование Ψ - является, как видно, линейным. К данному типу относятся: метод главных компонент или преобразование Карунена-Лоева, факторный анализ и его варианты, анализ независимых компонент и другие;
- нелинейные – в которых преобразование Ψ формируется нелинейным способом. Например, MDS (Multi-dimensional Scaling) [9] и t-SNE (t-distributed stochastic neighbor embedding) [8]. Необходимо отметить, что эти алгоритмы допускают использование различных метрик расстояния.

Кроме того, для решения задачи понижения размерности могут также использоваться методы кластерного анализа. Действительно, пусть имеется статистический метод кластеризации который можно представить следующим преобразованием $r_i = \Psi(\mathrm{e}_i)$, где как и раньше объект $\mathrm{e_i} \in \mathbb{R}^V$, где V – размерность пространства признаков, а $\mathrm{r}_i \in \{1; L\}$, L – число меток классов. Тогда такое преобразование можно рассматривать как преобразование понижения размерности.

Общей проблемой применения данных методов является необходимость конструирования метрик расстояния D в исходном пространстве $\mathbb{R}^V$. Рассмотрим основные аспекты конструирования метрик:

1) Как неявно (см. нотацию $\mathbb{R}^V$) было показано ранее на данном пространстве можно определить метрику, хотя бы в аспекте рассмотрения данного пространства как V-мерного арифметического пространства.

2) С учетом детального описания распознаваемого класса атак и разработанной в [1] модели однозначно утверждать, что пространство признаков $\mathbb{R}^V$ не является линейным невозможно, поскольку возможно определить такие операции над признаками, которые будут удовлетворять аксиомам линейного пространства.

3) Детальное исследование данного пространства признаков выходит за пределы исследования, однако из п. 1) и 2) видно, что теоретических препятствий для использования в качестве метрики Евклидова расстояния нет.

4) Выбор конкретного алгоритма понижения размерности принципиально не меняет разработанный метод.

Таким образом в разработанном методе мы выдвигаем следующие критерии для выбора алгоритма понижения размерности:

1) Преобразование $\mathrm{r_i} = \Psi(\mathrm{e}_i)$, где объект $\mathrm{e_i} \in \mathbb{R}^V$, где V – размерность пространства признаков, а $\mathrm{r_i} \in \mathbb{R}^y$, причем **y=1**
2) Если $D(\mathrm{e}_i, \mathrm{e}_j) < D(\mathrm{e}_i, \mathrm{e}_k)$, то $d(\mathrm{r}_i, \mathrm{r}_j) < d(\mathrm{r}_i, \mathrm{r}_k)$ для $\forall i, j, k$
3) Если $D(\mathrm{e}_i, \mathrm{e}_j) = 0$, то $d(\mathrm{r}_i, \mathrm{r}_j) = 0$ для $\forall i, j$

В методе может быть использован любой из алгоритмов, удовлетворяющий данным условиям.

Для определенности в дальнейшем и для проведения экспериментальных исследований мы будем использовать для понижения размерности алгоритм кластеризации на основе самоорганизующейся карты Кохонена [17].

4. Описание разработанного метода. Таким образом, разработанный метод, как и любой другой метод машинного обучения, может быть представлен в виде последовательности двух фаз: обучение и классификация.

Назначение фазы обучения заключается в построении классификатора путем итеративной настройки параметров на множестве обучающих примеров (т.н. обучающем множестве), а также оценке (верификации) полученной модели прогнозирования на множестве проверочных примеров (тестовом множестве). Необходимо отметить, что обучающее множество и тестовое множество должны быть предварительно классифицированы экспертом (хотя бы частично) [5].

Если результат проверки обученного классификатора на тестовом множестве совпадает с ожидаемыми и достаточен для классификации осуществляется переход к следующему этапу. Фаза обучения лежит в канве общего принципа построения моделей данных, приведенного в [1] и конкретизируется только способом вычисления искомых параметров классификатора – а именно, обучением. Результатом фазы обучения является классификатор с настроенными параметрами.

Назначение фазы классификации заключается, собственно, в вычислении меток классов для неизвестных ранее наборов данных с использованием настроенного (обученного) классификатора. Результатом фазы классификации является множество меток классов для неизвестных ранее наборов данных.

Исходя из вышеизложенного, сформулируем шаги метода.

1. Для каждого защищаемого сервиса построить отдельную гибридную ИНС. В дальнейшем рассматриваем обнаружение атаки на один сервис, остальные работают аналогично.

2. Для отдельного сервиса получить набор пакетов, число которых определено величиной окна.

3. Сформировать вектора для шага снижения размерности (самоорганизующейся карты).

4. Снизить размерность входных данных. В разрабатываемом методе - кластеризовать вектора при помощи SOM.

5. Построить вектора для MLP, в которых каждый компонент соответствует номеру кластера, которому принадлежит пакет. Т.о. входным вектором будет набор кластеризованных пакетов, сохраняющий информацию о порядке (последовательности) поступления. Для пакетов уже определена принадлежность к определённому типу.

6. Проанализировать вектора на MLP, классифицировать выявленные в трафике наборы. В данном случае происходит разделение на два класса - атака или норма.

Рассмотрим шаги метода подробно.

Формально сценарий атаки описывается кортежем элементов $e_i =< x_j^i : j = 0..V-1 >$, где каждый элемент представляет собой строковые данные HTTP-запроса в ASCII-коде.

Кластеризуемые пакеты разбиваются на непересекающиеся окна. Размер окна определяется по формуле:

$$w = u * \frac{S_{byte}}{8 * P_{min}}, \quad (2)$$

где S_{byte} – скорость передачи информации в сети в байтах в секунду; P_{min} – минимальный теоретический размер пакета; u – коэффициент, показывающий уровень утилизации канала передачи информации.

Фаза обучения в разработанном методе заключается в последовательном выполнении следующих этапов:

1. Обучение параметров алгоритма снижения размерности данных R(.). На этапе обучения SOM из собранных пакетов формируются обучающая (из нормальных и атакующих пакетов) и тестовая (все остальные пакеты) выборки. При разбиении на окна информация о том, содержится ли в окне атакующий сценарий, сохраняется и учитывается в дальнейшем. Окна размечаются экспертом как нормальные или атакующие на основании выполнения следующих условий:
   а) бнаружено наличие в окне набора однотипных пакетов, появляющихся с периодичностью, достаточной для поддержания открытых соединений.
   б) "Мощность" такого набора достаточна для того, чтобы отнести её к аномалии.

Размер обучающей выборки лимитируется размером памяти и временем, необходимым для обучения карты Кохонена – самой ресурсозатратной части.

2. Снижение размерности обучающих данных T' = R(T). Сеть Кохонена обучается на отдельных пакетах, которым ставятся в соответствие номера кластеров.

3. Формирование обучающего и тестового множества T'-> [T1,T2]. Выходной вектор SOM является входным вектором MLP. Первый компонент входного вектор для MLP – номер кластера, в который распределился первый пакет, второй компонент – номер кластера, в который распределился второй пакет, и так далее. Компоненты вектора также нормированы в диапазоне [0, 1]. Размерность входного вектора для персептрона зависит от числа выходов SOM (т.е. размера окна), выходной вектор имеет размерность 2.

4. Обучение нейросетевого классификатора arg min N(T1,W). На этапе обучения входные вектора помечаются как нормальные или атакующие на основании информации из шага 4 метода [18, 19]:

- ♦ есть в текущем окне нет атакующего сценария, вектор помечается как «норма»;
- ♦ есть в текущем окне есть атакующий сценарий, вектор помечается как «атака».

Выходной вектор имеет вид $<y_n,y_a>$, нормальные вектора обучаются на выход <1,0>, атакующие – на <0,1>.

5. Валидация классификатора E = Error(N(T2,W),?) на тестовой выборке.

6. Если точность классификатора согласуется с ожидаемой оценкой – переход к этапу 7, иначе к этапу 2.

7. Сохранение параметров алгоритма снижения размерности данных R и нейросетевого классификатора W. Переход к фазе классификации.

Фаза классификации, в целом, повторяет этапы фазы обучения.

1. Снижение размерности обучающих данных T' = R(T). Сеть Кохонена обучается на отдельных пакетах, которым ставятся в соответствие номера кластеров.

В общем случае атаки могут быть направлены на различные сетевые сервисы, характеризующиеся разными значениями портов приемника (destination port). При этом сценарий атаки не изменяется. Поэтому предлагается контролировать каждый порт отдельно, и, соответственно, использовать отдельный кластеризатор и классификатор.

Необходимо отметить, что разделение входящих пакетов по адресам источников представляется ненужным, т.к. в решаемой задаче не влияет ни на последовательность поступления пакетов, ни на их содержимое. Подробнее обоснование этого приведено в [1].

2. Для проведения исследований используются предварительно сохранённые наборы векторов двух типов - полученные на основе "чистого" трафика, и полученные на основе атакующих пакетов, содержащих трафик низкоинтенсивной атаки. Сохранённые пакеты разбивались на интервалы-окна в соответствии с алгоритмом 1 и формулой 2.

Окно сдвигается на определённую величину на каждой итерации работы метода, обеспечивая перекрытие с предыдущим окном. Величина сдвига зависит от аппаратной производительности платформы и подбирается на этапе экспериментального исследования (развёртывания системы в реальной защищаемой сети на фазе обучения). Это позволяет точно установить факт начала атаки.

3. Каждый пакет из окна преобразуется в формат входного вектора для SOM. Входной вектор содержит следующие компоненты, выделенные на основе анализа набора независимых атрибутов атаки [1] и прямо вытекающих из перечня свойств, позволяющих определить наличие атаки (п.1 данной статьи):

№ байта	Содержание
1	Дельта временной метки от предыдущего пакета, приведённая в диапазон 0-1
2 - 51	Порция строковых данных пакета в ASCII-2 кодировке, приведённая в диапазон 0-1

4. SOM используется для кластеризации событий в узлы матрицы, в которых будут сгруппированы события (пакеты) определённых типов.

На этапе классификации на вход SOM последовательно подаются вектора из текущего окна, которые распределяются по кластерам. В результате каждая компонента выходного вектора SOM соответствует сетевому пакету. Выходной вектор SOM имеет вид $<N_1, N_2, \ldots, N_i>$, где i определяется размером окна, а N указывает на то, к какому кластеру сети Кохонена принадлежит данный пакет.

5. Выходной вектор SOM является входным вектором MLP.

6. После этого вектора подаются на вход многослойного персептрона, обученного распознавать атакующие последовательности пакетов, но уже с учётом информации о событии, т.е. принадлежности пакета той или иной группе-сценарию [19].

На этапе классификации MLP будет анализировать очередное окно пакетов, классифицируя его как нормальный или атакующий сценарий. Ответы интерпретируются следующим образом:

- если $y_n>0.7$ и $y_a<0.3$, то набор пакетов нормальный;
- если $y_n<0.3$ и $y_a>0.7$, то набор пакетов атакующий;
- иначе – ИНС не может классифицировать пакет.

Результаты анализа («норма», «атака», «невозможно классифицировать») выводятся по каждому окну.

5. Описание реализации разработанного метода. Исходя из описания метода был разработан прототип системы обнаружения низкоинтенсивных атак на web-сервисы на основе протокола http.

Как указано выше, в общем случае атаки могут быть направлены на различные сетевые сервисы. С точки зрения обработки пакетов стеком, сервисы характеризуются разными значениями портов приемника (destination port). При этом сценарий атаки не изменяется. Поэтому предлагается контролировать каждый порт отдельно, и, соответственно, использовать отдельный кластеризатор и классификатор. В прототипе будет рассматриваться один сервис, работающий по 80 порту.

Структурная схема прототипа системы обнаружения атак, реализующей метод, в общем представлена на рис. 1.

Система использует сенсор, работающий на основе библиотеки libpcap [10]. Библиотека позволяет использовать быстрый движок перехвата пакетов и гибкую систему выражений фильтрации. С её помощью отфильтровывается трафик, направленный на защищаемый сервис, и передаётся в подсистему анализа.

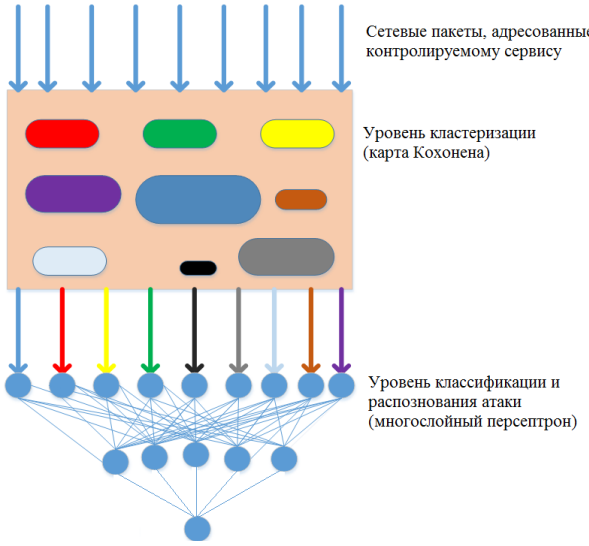

Рис. 1. Архитектура программного комплекса обнаружения LR-DOS атак

5.1. Параметры эксперимента. Обучение карты Кохонена производится на отдельных пакетах, последовательно выбираемых из окна. Перед подачей на сеть Кохонена и на персептрон все данные нормировались в диапазон [0,1]. Сеть Кохонена имеет гексагональную структуру связей нейронов и размеры 25 на 20 (всего 25*20=500 кластеров).

Размер окна вычисляется по формуле 2.

Стандарт Ethernet требует, чтобы между пакетами был 12-байтовый период "тишины", так и определяется окончание одного пакета и начало следующего. В конце каждого пакета также нужно передавать CRC-код (4 байта) для проверки целостности передачи, а в начале пакета - обязательную преамбулу из 8 байт. Есть и еще одно ограничение - минимальный размер пакета 60 байт [11].

Если учесть все ограничения, то пакеты должны быть минимум по 84 байта.

Для 1 Гбит/с мы получаем теоретическое ограничение 1 000 000 000/84*8 = 1 488 095 пакетов в секунду.

Для 100 Мбит/с - 1 00 000 000/84*8 = 148810 пакетов в секунду.

Выбор коэффициента *u*, показывающего уровень утилизации канала передачи информации, зависит от загруженности сети в нормальном режиме работы.

Нормальным режимом будем считать загруженность канала в диапазоне от 3 % до 10 %. В работе [12] высказано мнение, что для сети с предельной скоростью 100 Мбит/с с точки зрения эффективной работы системы обнаружения атак в этом диапазоне находятся значения от нормальных до высоких. Предельной загруженностью, при которой возможен анализ без потерь пакетов, в работе [12] полагается 40 %. Поскольку трафик, генерируемый при низкоинтенсивной атаке, является минимальным, будем экспериментировать на нижнем пределе загруженности.

Соответственно, 0,01*148810 ~ 1500 пакетов в секунду при нормальной утилизации в 1 %.

Кроме того, используем ещё два значения:

- размер окна 30 пакетов – минимальное значение, встречающееся в правилах IDS Snort для низкоинтенсивных атак.
- размер окна 180 – соответствует скорости поступления 1 пакет в секунду.

Для анализа строковых данных используются первые 50 символов необработанных данных полезной нагрузки пакета, так как основная часть значимых для анализа данных в собранных сетевых пакетах содержится в самом начале пакета. Символы представляются в ASCII-коде. Подобный метод анализа "сырых" сетевых данных был описан в [13] и [16].

Число нейронов (кластеров) в карте Кохонена som_size = 500.

Персептрон имеет следующую структуру – два скрытых и выходной слой, активационная функций в скрытый слоях – гиперболический тангенс, в выходном слое – линейная. Число нейронов в скрытых слоях – 21 и 7 (подобрано в ходе экспериментов). Метод обучения – trainlm – быстрый и затратный по памяти, но при рассматриваемых входных данных проблем с перерасходом памяти не наблюдалось.

5.2. Экспериментальное исследование. Для обучения искусственной нейронной сети моделировались два типа сетевого трафика – нормальный и аномальный. Первый содержал пакеты, появляющиеся в сети при обычной работе, а второй имитировал распределённую атаку [14].

Сбор данных проводилось в 2 этапа.

1. Этап сбора нормального поведения. На атакуемом сервере создан php скрипт, который выполняет «тяжелый» sql запрос (Цикл в N итераций). С атакующего сервера запускался Apache BanchMark для создания легитимных подключений к серверу. 10 потоков по 100 запросов.

2. Этап сбора аномального поведения. Для имитации трафика, возникающего во время атаки, на атакующем сервере запускается 3 копии скрипта slowloris с разницей в параметре отвечающем за задержку между повторными подключениями [20]. Пример запуска:

```
./slowloris.pl -dns google.com -port 80 -timeout 500 -num 100500
```

- Размер нормального набора – 459565 пакетов.
- Размер атакующего набора – 428890 пакетов.

Архитектура тестового стенда, на котором проводились экспериментальные исследования, представлена на рис. 2 [15].

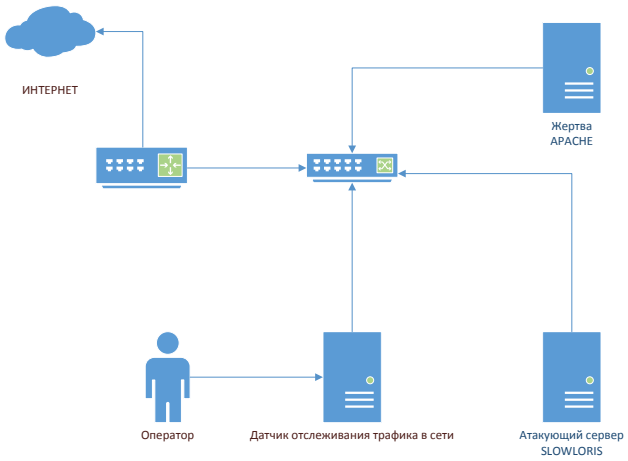

Рис. 2. Общая схема экспериментального стенда

На рис. 3 представлен график загруженности полосы пропускания в штатном режиме работы, на этапе сбора нормального поведения.

На рис. 4 приведены графики для этапа сбора атакующего трафика - потери пакетов из-за недоступности сервиса для новых соединений (красная линия) и нормальный трафик, обрабатываемый сервисом (синяя линия).

Рис. 3. Загруженность полосы пропускания (нормальная)

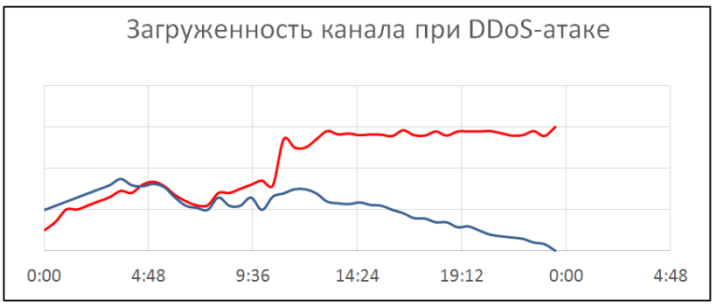

Рис. 4. Загруженность канала под атакой

На рис. 3 и 4 по оси абсцисс откладывается время суток, начиная с 00.00 часов, по оси ординат максимум является 100% загруженность полосы пропускания.

В ходе экспериментов исследовалось влияние следующих значений на эффективность метода.

Размер строковых данных:

- 20 байт;
- 50 байт.

Величина обучающей выборки:

- 5000 пакетов;
- 30000 пакетов.

Размер окна (в пакетах):

- 30;
- 180;
- 1500.

Размер сдвига:

- на 10 %;
- на 30 %.

Распознавание осуществлялось на тестовой выборке. Оценивалась близость к эталону. Распознавание считалось успешным, если абсолютная разница между эталонными и фактическими значениями для каждой компоненты выходного вектора не превосходила 0.3.

Результаты экспериментального исследования представлены в табл. 1. На рис. 5 результаты работы кластеризатора.

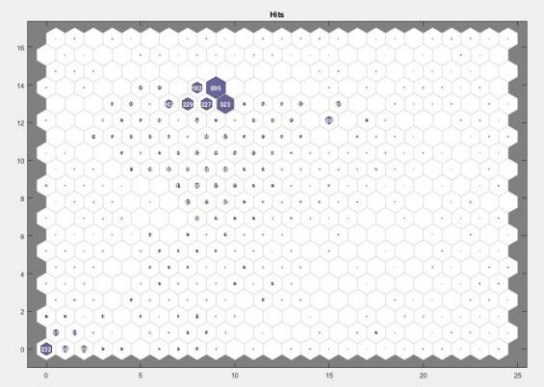

Рис. 5. Результаты кластеризации пакетов сетью Кохонена

На рис. 6 представлены значения ошибки на различных выборках при обучении персептрона.

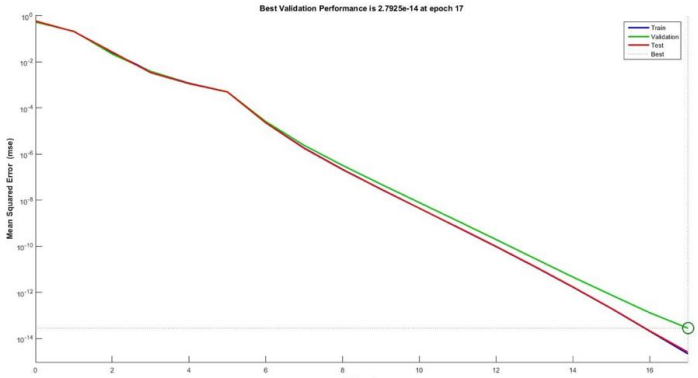

Рис. 6. Ошибка на различных выборках при обучении персептрона

Таблица 1

Результаты работы прототипа системы обнаружения атак

№	Длина строки	Величина обуч. выборки для SOM, пакетов	Величина обуч. выборки для FFNET, окон	Размер окна	Размер сдвига	Время обучения	Результат на тестовой выборке	
							Ошибка 1 рода	Ошибка 2 рода
	20	5000	4000	30	3	00:01:34	8.1827e-04	0.0050
	50	30000	24000	30	3	00:26:48	0.0367	0.0172
	20	5000	800	180	18	00:01:48	0	3.4378e-04
	50	30000	4800	180	18	00:18:33	0.0449	0.0449
	20	5000	90	1500	150	00:01:11	0.0554	0.5287
	50	30000	540	1500	150	00:20:33	0	0.0033
	20	5000	900	30	15	00:01:18	0.0118	0.0900
	50	30000	5400	30	15	00:22:44	0.0289	0.0232
	20	5000	160	180	90	00:01:05	0.0011	0.1124
	50	30000	960	180	90	00:16:06	0	0.1447
	20	5000	20	1500	750	00:01:05	0.1154	0.8386
	50	30000	120	1500	750	00:15:21	0	0.0471

Видно, что число ложных срабатываний не превышает 0,115 % в самом худшем случае, что было обусловлено минимальным размером анализируемых данных и обучающей выборки. Значение пропуска цели поднималось до 0,84 % также в одиннадцатом эксперименте. Лучшие результаты показана в экспериментах 1, 6, 12, что подтверждает теоретические предположения.

Заключение. В статье представлено описание разработанного метода обнаружения низкоинтенсивных атак «отказ в обслуживании», основанного на использовании гибридной нейронной сети.

Метод демонстрирует высокий процент обнаружения атак за счёт снижения числа необнаруженных атак (ошибок второго рода), при этом скорость работы зависит лишь от скорости обработки поступающих пакетов

Разработана архитектура и прототип программного комплекса, предназначенного для обнаружения низкоинтенсивных атак типа «отказ в обслуживании», произведено экспериментальное исследование эффективности разработанного комплекса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Тарасов Я.В.* Модель низкоинтенсивной сетевой атаки "отказ в обслуживании" // Сборник трудов VII Всероссийской научно-технической конференции «Безопасные информационные технологии» (БИТ – 2016). – М.: МГТУ им. Н.Э. Баумана.
2. *Чучуева И.А.* Модель прогнозирования временных рядов по выборке максимального подобия: дисс. … канд. техн. наук. – М., 2012.
3. *Fogler H.R.* A pattern recognition model for forecasting // Management science. – 1974. – No. 8. – P. 1178-1189.
4. *F. Martinez Alvarez [at al.]* Discovering Patterns in Electricity Price Using Clustering Techniques // ICREPQ International Conference on Renewable Energies and Power Quality, Spain, Sevilla, 2007. –8 p. URL: http://www.icrepq.com/icrepq07/245-martinez.pdf (дата обращения 15.1.2016).
5. *Haykin.* Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey, 1999. – 2nd ed. – 842 p.
6. *C. Lee Giles, Steve Lawrence, Ah Chung Tsoi*. Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference // Machine Learning. – July 2001. – Vol. 44, Issue 1. – P. 161-183.
7. *Fodor I.* A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National, Technical Report UCRL-ID-148494. – 2002.
8. *Van der Maaten, L.J.P., Hinton, G.E*. Visualizing High-Dimensional Data Using t-SNE // Journal of Machine Learning Research. – Nov. 2008. – No. 9. – P. 2579-2605.
9. *Borg I., Groenen P*. Modern Multidimensional Scaling: theory and applications – 2nd ed. – New York: Springer-Verlag, 2005. – P. 207–212. ISBN 0-387-94845-7.
10. Command-line packet analyzer tcpdump. http://www.tcpdump.org/ (дата обращения 03.12.2016).
11. *Robert Graham*. What's the max speed on Ethernet? http://blog.erratasec.com/2013/10/whats-max-speed-on-ethernet.html#.UlbwuNK8Dp8 (дата обращения 03.12.2016).
12. *Stephen Northcutt, Judy Novak*. Network Intrusion Detection An Analyst's Handbook. – Sams Publishing, 2002. – 346 p.
13. *Ghost A.K., et al.* Detecting Anomalous and Unknown Intrusions Against Programs in Real-Time // DARPA SBIR Phase I Final Report. Reliable Software Technologies.
14. *Тарасов Я.В., Макаревич О.Б.* Моделирование и исследование низкоинтенсивных DoS-атак на BGP-инфраструктуру // Известия ЮФУ. Технические науки. – 2013. – № 12 (149). – С. 101-111.
15. *Тарасов Я.В.* Метод обнаружения низкоинтенсивных DDOS-атак на основе гибридной нейронной сети, инфраструктуру // Известия ЮФУ. Технические науки. – 2014. – № 8 (157). – С. 47-87.

16. *Абрамов Е.С., Сидоров И.Д.* Метод обнаружения распределённых информационных воздействий на основе гибридной нейронной сети // Известия ЮФУ. Технические науки. – 2009. – № 11 (100). – С. 154-164.
17. *Kohonen T.* Self-Organizing Maps. Third, extended edition. – Springer, 2001.
18. *Абрамов Е.С., Аникеев М.В., Макаревич О.Б.* Использование аппарата нейросетей при обнаружении сетевых атак // Известия ТРТУ. – 2004. – № 1 (36). – С. 130.
19. *Абрамов Е.С., Аникеев М.В., Макаревич О.Б.* Подготовка данных для использования в обучении и тестировании нейросетей при обнаружении сетевых атак // Известия ТРТУ. – 2003. – № 4 (33). – С. 204-206.
20. *Aiello M., Cambiaso E., Scaglione S., Papaleo G.* A similarity based approach for application DoS attacks detection // 2013 IEEE Symposium on Computers and Communications (ISCC).

REFERENCES

1. *Tarasov Ya.V.* Model' nizkointensivnoy setevoy ataki "otkaz v obsluzhivanii" [Neural network method of detection of low-rate dos-attacks on web-services], *Sbornik trudov VII Vserossiyskoy nauchno-tekhnicheskoy konferentsii «Bezopasnye informatsionnye tekhnologii» (BIT – 2016)* [proceedings of VII All-Russian Scientific and Technical Conference "Safety of information technology" (BIT - 2016). Moscow: MGTU im. N.E. Baumana.
2. *Chuchueva I.A.* Model' prognozirovaniya vremennykh ryadov po vyborke maksimal'nogo podobiya: diss. kand. tekhn. nauk [Model prediction of time series based on a sample of maximum similarity. Dr. of eng. sc. diss. Moscow, 2012.
3. *Fogler H.R.* A pattern recognition model for forecasting, *Management science*, 1974, No. 8, pp. 1178-1189.
4. *F. Martinez Alvarez [at al.]* Discovering Patterns in Electricity Price Using Clustering Techniques, *ICREPQ International Conference on Renewable Energies and Power Quality, Spain, Sevilla, 2007*. 8 p. Available at: http://www.icrepq.com/icrepq07/245-martinez.pdf (accessed 15 January 2016).
5. *Haykin.* Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey, 1999. 2nd ed., 842 p.
6. *C. Lee Giles, Steve Lawrence, Ah Chung Tsoi*. Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference, *Machine Learning*, July 2001, Vol. 44, Issue 1, pp. 161-183.
7. *Fodor I.* A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National, Technical Report UCRL-ID-148494. 2002.
8. *Van der Maaten, L.J.P., Hinton, G.E*. Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*, Nov. 2008, No. 9, pp. 2579-2605.
9. *Borg I., Groenen P.* Modern Multidimensional Scaling: theory and applications 2nd ed. New York: Springer-Verlag, 2005, pp. 207-212. ISBN 0-387-94845-7.
10. Command-line packet analyzer tcpdump. Available at: http://www.tcpdump.org/ (accessed 03 December 2016).
11. *Robert Graham*. What's the max speed on Ethernet? Available at: http://blog.erratasec.com/2013/10/whats-max-speed-on-ethernet.html#.UlbwuNK8Dp8 (accessed 03 December 2016).
12. *Stephen Northcutt, Judy Novak*. Network Intrusion Detection An Analyst's Handbook. Sams Publishing, 2002, 346 p.
13. *Ghost A.K., et al.* Detecting Anomalous and Unknown Intrusions Against Programs in Real-Time, *DARPA SBIR Phase I Final Report. Reliable Software Technologies*.
14. *Tarasov Ya.V., Makarevich O.B.* Modelirovanie i issledovanie nizkointensivnykh DoS-atak na BGP-infrastrukturu [Modeling and study of low-intensity DOS-attacks on BGP-infrastructure], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 12 (149), pp. 101-111.
15. *Tarasov Ya.V.* Metod obnaruzheniya nizkointensivnykh DDOS-atak na osnove gibridnoy neyronnoy seti, infrastrukturu [Method of detection of low-rate dos-attacks based on hybrid neural network], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 8 (157), pp. 47-87.

16. *Abramov E.S., Sidorov I.D.* Metod obnaruzheniya raspredelennykh informatsionnykh vozdeystviy na osnove gibridnoy neyronnoy seti [Metod obnaruzheniya raspredelennykh informatsionnykh vozdeystviy na osnove gibridnoy neyronnoy seti], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 11 (100), pp. 154-164.
17. *Kohonen T.* Self-Organizing Maps. Third, extended edition. Springer, 2001.
18. *Abramov E.S., Anikeev M.V., Makarevich O.B.* Ispol'zovanie apparata neĭroseteĭ pri obnaruzhenii setevykh atak [Ispol'zovanie apparata neyrosetey pri obnaruzhenii setevykh atak], *Izvestiya TRTU* [Izvestiya TSURE], 2004, No. 1 (36), pp. 130.
19. *Abramov E.S., Anikeev M.V., Makarevich O.B.* Podgotovka dannykh dlya ispol'zovaniya v obuchenii i testirovanii neĭroseteĭ pri obnaruzhenii setevykh atak [Preparing data for use in training and testing of neural networks in the detection of network attacks], *Izvestiya TRTU* [Izvestiya TSURE], 2003, No. 4 (33), pp. 204-206.
20. *Aiello M., Cambiaso E., Scaglione S., Papaleo G.* A similarity based approach for application DoS attacks detection, *2013 IEEE Symposium on Computers and Communications (ISCC).*

Статью рекомендовал к опубликованию д.т.н., профессор К.Е. Румянцев.

Абрамов Евгений Сергеевич – Южный федеральный университет; e-mail: abramoves@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371905; кафедра безопасности информационных технологий; зав. кафедрой.

Тумоян Евгений Петрович – e-mail: eptumoyan@sfedu.ru; кафедра безопасности информационных технологий; доцент.

Тарасов Ярослав Викторович – ЗАО «Инфосистемы Джет»; e-mail: info@jet.msk.su; 125252, г. Москва, ул. 2-я Песчаная, 2/1, корп. 50; тел.: 84954117601, факс: 84954117602; директор по развитию бизнеса компании «Инфосистемы Джет».

Abramov Evgeny Sergeevich – Southern Federal University; e-mail: abramoves@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +88634371905; the department of information security; head of the department.

Tumoyan Evgeny Petrovich – e-mail: eptumoyan@sfedu.ru; the department of information security; associate professor.

Tarasov Yaroslav Viktorovich – Jet Infosystems; e-mail: info@jet.msk.su; 2/1, 2nd Peschanaya street, build. 50, Moscow, 125252, Russia; phone: +74954117601, fax: +74954117602; director of Business Development in Jet Infosystems.

УДК 519.688 DOI 10.18522/2311-3103-2016-9-7181

С.Л. Беляков, А.В. Боженюк, М.Л. Белякова, А.А. Глушков

ДИНАМИЧЕСКАЯ ГЕОИНФОРМАЦИОННАЯ МОДЕЛЬ ДЛЯ ЗАДАЧ УПРАВЛЕНИЯ МАТЕРИАЛЬНЫМИ ПОТОКАМИ*

Статья посвящена разработке геоинформационной модели для задач управления материальными потоками. Модель используется при разработке геоинформационной системы, ориентированной на информационную поддержку принятия решений. Специфическим требованием к модели является обеспечение заданного уровня достоверности решений. Рассмотрены существующие методики синтеза геоинформационных моделей. Указаны недостатки геоинформационных моделей для картографирования. Главным из них является статический характер пространственных данных, описывающих реальную среду. Модели для систем поддержки принятия решений обладают несомненным преимуществом в построении динамических представлений пространственных данных, но не предусматри-

* Работа поддержана грантами РФФИ, проекты № 15-01-00149 и № 15-07-00185.