Инженерный вестник Дона, №3 (2017)
ivdon.ru/ru/magazine/archive/n3y2017/4354

Применение комбинированного нейросетевого метода для обнаружения низкоинтенсивных DDoS-атак на web-сервисы

Е.С. Абрамов[1], *Я.В. Тарасов*[2]

[1]*Южный федеральный университет, Ростов-на-Дону*
[2] *ЗАО «Инфосистемы Джет», Москва*

Аннотация: В статье описывается разработка и экспериментальное исследование эффективности метода обнаружения низкоинтенсивных (low-rate) атак типа «отказ в обслуживании» Используется модель низкоинтенсивных атак в виде аддитивного наложения нормальных сетевых событий и аномального трафика. Метод обнаружения заключается в последовательном выделении однородных групп временного ряда (поступающих сетевых пакетов) при помощи моделей распознавания образов и построения для каждой выделенной группы модели прогнозирования для обнаружения сценария атаки.
Ключевые слова: обнаружение атак; низкоинтенсивная атака; DoS-атака; персептрон, самоорганизующаяся карта; сетевая безопасность; распознавание образов.

Введение

Существующие методы обнаружения DoS-атак (denial of service, отказ в обслуживании), основанные на статистическом анализе пороговых значений, показали низкую эффективность для обнаружения нового класса DoS-атак прикладного уровня - низкоинтенсивных атак (low-rate-DoS) [1]. Общие сведения по проблеме и некоторые новые подходы рассматриваются в следующих работах: в [2] рассматриваются категории DoS-атак и общие подходы к обнаружению таких атак; в [3] и [4] представлен опыт создания лабораторной инфраструктуры для моделирования и изучения low-rate атак и результаты выделения характеристик атак, используемых в дальнейшем при разработке новых методов обнаружения; в [5] рассматриваются результаты исследований по разработке метода обнаружения на основе характеристик подобия атакующего трафика; вопросы эффективности типовых средств защиты информации против low-rate DDoS атак проанализированы в [1]; в [6] описывается использование многослойного персептрона (multi-layer perceptron, MLP) для обнаружения распределённых во времени сетевых событий с предварительной кластеризацией событий.

Основные представители данного класса атак - HTTP-flood [4,5], R-U-Dead-Yet? (сокращённо RUDY) [3], SlowLoris [7]. В работе [8] проведён анализ сценариев различных низкоинтенсивных атак, в результате которого показано, что во всех исследованных сценариях используются пакеты прикладных сервисов, повторяющиеся по малой временной шкале с определенной частотой.

Для рассмотренных сценариев атак характерны следующие признаки:

– генерация периодического трафика малого объёма;
– атакующее воздействие составляют однотипные элементы трафика;
– отдельный запрос или сетевой пакет нельзя определить как аномалию.

Во всех сценариях временной интервал между отдельными пакетами значительно меньше таймаута окончания соединения (connection time out). Но так как значение connection time out практически для всех приложений является реконфигурируемым параметром, нельзя заранее установить точное значение временного интервала между отправкой пакетов. Таким образом, для проведения атаки достаточно, чтобы очередной пакет мог прийти в любой момент из интервала ожидания.

Модель низкоинтенсивной DoS- атаки

Для обнаружения атаки необходимо выявлять периодическое появление определённого однотипного набора пакетов во входящем трафике, после чего относить этот набор к нормальному или аномальному классу (под аномалией будем понимать наличие низкоинтесивной атаки). При этом порядок, в котором следуют пакеты в сценарии, не будет играть значительной роли для обнаружения атаки. Информация о времени прихода пакетов учитывается при разбиении входящего трафика на "окна".

Свойствами набора пакетов, позволяющими отнести его к аномальному классу, являются, по значимости, следующие:

1) Полезная нагрузка протокола HTTP.

2) Порядок поступления пакетов.

3) Дельта времени между соседними пакетами.

Первое свойство позволяет определить тип пакета, второе - отнести набор пакетов к определённому сценарию.

Задачу разработки метода обнаружения атаки можно представить как разработку классификатора наборов сетевых пакетов $\zeta(\ldots)$, который должен выдавать метки классов (аномалия или нормальные данные) для временного ряда элементов $e_{i-z,\ldots,}e_{i-1},e_i$:

$$\zeta(e_{i-z,\ldots,}e_{i-1},e_i)=\begin{cases}C^s : s_i^{'}\neq 0\\ C^d : d_i^{`}\neq 0\end{cases}, \quad (1)$$

где ζ – 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.

Исходя из этого, задача обнаружения атаки может быть сведена к классификации многомерных временных рядов. Из работы [9] следует, что регрессионные и авторегрессионные модели и методы для решения поставленной задачи не подходят. Наиболее перспективным видится использование комбинированных нейросетевых моделей [8, 10, 11].

Сейчас для классификации временных рядов, представленных в виде последовательности атрибутов событий активно используется класс нейросетевых моделей на основе рекуррентных нейронных сетей (сети Хопфилда, Джордана и их варианты) [12, стр. 15].

Сети данной архитектуры имеют некоторые особенности, которые делают их использование нежелательными для решения задачи, поставленной в исследовании:

1) Склонность рекуррентных нейронных сетей к переобучению.

2) Обучающая выборка большого размера. Специфика решаемой задачи делает сбор дополнительных обучающих векторов нежелательным, а иногда невозможным.

3) Невысокая точность предсказания (классификации) на последовательностях событий малой длины, для которых характерен высокий уровень шума [13].

Альтернативный подход к классификации временных рядов заключается в следующем:

- использование скользящего окна для выделения признаков;
- классификатор обучается непосредственно для окна [12].

Поясним идею подхода. Пусть имеется ряд $E = \{e_i\}$ $\mathbf{E = \{e_i\}}$, $i = 0..N-1$ $\mathbf{i = 0..N-1}$, где N – длина временного ряда. Разделим временной ряд на отдельные участки (окна) B, где каждое окно будет являться вектором определенной длины. Длина окна представляет собой длину истории событий Z (см. формулу 1).

1. k = 0

2. for i = 0:N-1:

3. $\mathbf{B^k_{i\ mod\ z} = e_i}$ $B^k_{i \bmod Z} = e_i$

(алгоритм 1)

4. if i mod Z == Z-1:

5. k = k + 1

Основной проблемой при применении данного подхода при решении поставленной задачи является представление элемента классифицируемого временного ряда в виде вектора, т.е. $e_i =< x^i_j : j = 0..V-1 >$, где V – число анализируемых атрибутов. Вектора, которые будут получены при

«развертывании» многомерного временного ряда, будут иметь *B*-значительную размерность $|B_k| = Z \cdot V$ $|\mathcal{B}_k| = \mathbf{Z} \cdot \mathbf{V}$, что приводит к возрастанию вычислительной сложности метода.

Однако, это представляется частной проблемой. Для её решения предлагается использовать методы кластеризации и снижения размерности данных.

Применение методов снижения размерности данных в решаемой задаче подробно рассматриваются в работах [8,14,15].

При проведении экспериментальных исследований для понижения размерности будет использоваться алгоритм кластеризации на основе самоорганизующейся карты Кохонена (self-organizing map, SOM) [16].

Метод обнаружения низкоинтенсивных атак

Как и любой метод машинного обучения, описываемый метод обнаружения низкоинтенсивных атак может быть представлен в виде двух последовательных фаз - фазы обучения и фазы классификации.

Фаза обучения подчиняется общим принципам построения моделей данных, и конкретизируется только используемым методом обучения [8].

В фазе обучения строится классификатор. Это происходит путём итерационной настройки параметров классификатора на обучающем множестве. Далее в этой фазе происходит оценка (верификация) полученной модели прогнозирования временных рядов на тестовом множестве, состоящем из проверочных примеров. Как множество обучающих примеров, так и множество проверочных примеров должны быть предварительно, хотя бы частично, классифицированы экспертом.

В случае совпадения результата проверки обученного классификатора на тестовом множестве с ожидаемым результатом, и если, при этом, результат достаточен для классификации, переходят к следующей фазе. В

результате фазы обучения мы получаем классификатор с настроенными параметрами, достаточными для успешной классификации.

Целью этапа классификации является вычисление меток классов для ранее неизвестных наборов данных с применением обученного классификатора. Результат этапа классификации - набор меток классов для ранее неизвестных наборов данных.

Можно сформулировать шаги метода.

1. Построить отдельную искусственную нейронную сеть для каждого контролируемого сервиса (порта). Сети функционируют аналогично друг другу. Далее будет рассматриваться выявление атак на один сервис.

2. Для выбранного сервиса принять от источника данных некоторое множество сетевых пакетов, число которых определяется выбранным значением величины окна.

3. На шаге снижения размерности формируются вектора для самоорганизующейся карты.

4. Снизить размерность входных данных. Для разработанного метода - кластеризация векторов самоорганизующейся картой.

5. Сформировать вектора для многослойного персептрона (MLP), где каждый компонент вектора будет соответствовать номеру кластера, в который распределился пакет. Таким образом, входной вектор представляет собой набор кластеризованных сетевых пакетов, который сохраняет информацию о последовательности (порядке) поступления внутри окна. Для этих пакетов уже будет установлена их принадлежность к определённому типу.

6. Вектора анализируются на MLP, выявленные в трафике на шаге 4 наборы классифицируются. В результате осуществляется разделение на два класса - атака или норма.

Следует отметить, что разделение входящих пакетов по адресам источников оказывается ненужным, т.к. в данном случае не влияет ни на порядок поступления пакетов, ни на их содержимое.

Шаги метода подробно описаны в [8]. Необходимо отметить следующее:

1. В ходе работы использовались предварительно собранные наборы векторов двух типов - сформированные из "чистого" трафика, и из трафика, содержащего атакующие пакеты. Собранные пакеты были разбиты на окна-интервалы в соответствии с алгоритмом 1 и формулой 2.

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

где S_{byte} - скорость передачи информации в сети в байтах в секунду,

P_{min} - теоретически возможный минимальный размер пакета,

u - коэффициент уровня использования канала передачи информации.

При каждой следующей итерации работы окно сдвигается на фиксированное число пакетов. Значение сдвига определяется экспериментально на фазе обучения, и должно обеспечивать перекрытие с предыдущим окном, т.е. не превышать его размера. Выбор значения сдвига зависит от аппаратной производительности платформы и скорости защищаемой сети. Такой подход позволяет точно установить начальный набор, в котором зафиксировано начало атаки.

Пакеты из интервала-окна преобразуются во входные вектора для самоорганизующейся карты. Формат входного вектора прямо следует из перечня свойств, позволяющих отнести пакет к атакующему классу (таблица 1).

2. Самоорганизующаяся карта используется для кластеризации событий в узлы матрицы, в которых группируются пакеты определённых типов (напр., определённых прикладных протоколов).

Таблица № 1

Состав компонентов входного вектора

№ байта	Содержание
1	Дельта временной метки от предыдущего пакета, нормализованная в диапазоне 0-1
2 - 51	Данные заголовка и полезной нагрузки пакета в ASCII-2 кодировке, нормализованные в диапазоне 0-1 [6, 17]

В фазе классификации на вход самоорганизующейся карты последовательно подаются вектора из текущего окна. SOM распределяет их по кластерам. В итоге каждая компонента выходного вектора самоорганизующейся карты соответствует одному сетевому пакету. Этот выходной вектор имеет вид $<N_1, N_2, \ldots, N_i>$, где i определяется размером окна, а N указывает на то, к какому кластеру самоорганизующейся сети принадлежит данный пакет.

3. Выходной вектор самоорганизующейся карты является входным вектором многослойного персептрона.

4. После этого, с учётом информации о принадлежности пакета той или иной группе-сценарию, вектора подаются на вход MLP, который обучен распознавать атакующие последовательности пакетов.

В фазе классификации персептрон анализирует очередное окно, классифицируя его как атаку или норму. 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.

Для проведения экспериментальных исследований была разработана архитектура и программная реализация прототипа системы обнаружения низкоинтенсивных атак на web-сервисы на основе разработанного метода.

Система использует сенсор на основе библиотеки libpcap [18].

Экспериментальное исследование и оценка эффективности разработанного метода

Обучение сети Кохонена происходит на отдельных пакетах, последовательно выбираемых из окна. Перед подачей на SOM и MLP все данные нормируются в диапазон [0,1]. Самоорганизующаяся сеть имеет размеры 25 на 20 и использует гексагональную структуру связей нейронов.

Как показано выше, размер окна определяется по формуле 2. На эффективную величину влияют ограничения технологии Ethernet [19] и утилизация полосы пропускания сетевого канала. При проведении экспериментального исследования использовались следующие значения:

- размер окна 1500 пакетов – для утилизации канала передачи в 1% при скорости сети 100 Мбит/с [19,20];
- размер окна 30 пакетов - минимальное значение числа пакетов в сценарии, применяемое в правилах системы обнаружения атак Snort для низкоинтенсивных атак.
- размер окна 180 - соответствует скорости поступления 1 пакет в секунду.

Используется многослойный персептрон со следующей структурой – два скрытых слоя с числом нейронов 21 и 7 (подобрано в ходе экспериментов), выходной слой. Активационная функция в скрытый слоях – гиперболический тангенс, в выходном слое – линейная. Метод обучения – trainlm.

Для обучения искусственной нейронной сети моделировались два типа сетевого трафика – нормальный и атакующий, с имитацией распределённой низконтенсивной атаки с 10 адресов.

Размер нормального набора - 459565 пакетов.

Размер атакующего набора - 428890 пакетов.

Распознавание проводилось 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 таблице 2.

Таблица 2

Результаты работы прототипа системы обнаружения атак

№	Длина строки (вектора)	Величина обуч. выборки для SOM, пакетов	Величина обуч. выборки для FFNET, окон	Размер окна	Размер сдвига	Результат на тестовой выборке	
						Ошибка 1 рода	Ошибка 2 рода
1.	20	5000	4000	30	3	8.1827e-04	0.0050
2.	50	30000	24000	30	3	0.0367	0.0172
3.	20	5000	800	180	18	0	3.4378e-04
4.	50	30000	4800	180	18	0.0449	0.0449
5.	20	5000	90	1500	150	0.0554	0.5287
6.	50	30000	540	1500	150	0	0.0033
7.	20	5000	900	30	15	0.0118	0.0900
8.	50	30000	5400	30	15	0.0289	0.0232
9.	20	5000	160	180	90	0.0011	0.1124
10.	50	30000	960	180	90	0	0.1447
11.	20	5000	20	1500	750	0.1154	0.8386
12.	50	30000	120	1500	750	0	0.0471

Ошибка первого рода (ложное срабатывание) в худшем случае не превышает 0,12% в эксперименте № 11. Ошибка второго рода (пропуск цели,

фактически – результативность распознавания) в наихудшем случае составляет 0,84% в том же эксперименте. Результаты 11 эксперимента обусловлены минимальным размером анализируемых данных и обучающей выборки, использовавшимся в анализируемом эксперименте.

Как видно из таблицы, наилучшие результаты разработанный метод показал в экспериментах 1, 6, 12, что подтверждает теоретические предположения. Результаты 12 эксперимента показывают теоретическую возможность эффективного применения метода на высоких скоростях поступления пакетов [25,26].

Анализ техник противодействия низкоинтенсивным атакам

Рассмотрим техники, применяемые сегодня для обнаружения и противодействия low-rate DDoS атакам. Для этого используют конфигурирование правил межсетевых экранов и систем обнаружения вторжений (IDS). Правила межсетевого экранирования рассматриваются на примере iptables [21], правила систем обнаружения вторжений – на примере Snort [22].

1) Статический лимит соединений

```
iptables -I INPUT -p tcp --syn --dport 80 -m connlimit \
--connlimit-above 5 --connlimit-mask 32 -j DROP
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

Правила ограничивают число соединений пятью с одного узла. При превышении этого лимита попытка соединения запрещается. Если у атакующего достаточно большой ботнет, он всё равно может успешно атаковать. Также значение connlimit должно быть очень низким, чтобы защитить web-сервер. Это может существенно затруднить использование сервера легитимными клиентами и вызывает большое число ложных

срабатываний (false positives). Такие правила затрудняют или исключают использование NAT и proxy-серверов.

2) Динамический лимит соединений

```
iptables -I INPUT -p tcp -m state --state NEW --dport 80\
-m recent --name slowloris --set
iptables -I INPUT -p tcp -m state --state NEW --dport 80\
-m recent --name slowloris --update \
--seconds 15 --hitcount 10 -j DROP
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Первые два правила проверяют новые соединения и число соединений с одного ip-адреса. Если с одного ip–адреса создаётся более 10 соединений в течение 15 секунд, то такие пакеты блокируется. Для такого набора правил характерно неприемлемо большое число ложных срабатываний [23]. Легитимные подключения хостов через один NAT и proxy-сервер также будут удовлетворять этим правилам и будут блокироваться межсетевым экраном.

3) Применение правил IDS

Правила Snort, входящие в состав набора правил по умолчанию [24], основаны на поиске строковых шаблонов, специфичных для определённых сценариев атак, и контроле пороговых значений пакетов в единицу времени.

```
alert tcp any any -> any any (msg:"low-rate DDoS";
flow:to_server,established; content:”some DoS tool user-agent specific content”
detection_filter:track by_src, count 20, seconds 20;
metadata:service http; classtype:attempted-dos; sid:1234572; rev:2;)
```

Такие правила легко обходятся, поскольку параметры сценария и user-агента легко реконфигурируются. Кроме того, как и в случае правил

межсетевого экрана, основанных на контроле числа соединений, эти правила IDS генерируют много ложных срабатываний.

Заключение

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

Результаты сравнительного анализа применяемых методов обнаружения low-rate атак показывают, что методы защиты серверов, основанные на изменении конфигурации сервера или применении правил межсетевых экранов и IDS не позволяют эффективно защищаться от low-rate DDoS. Для этих методов характерно высокий уровень ложных срабатываний (ошибок первого рода).

Экспериментальная оценка эффективности предложенного в статье метода показала высокий процент обнаружения атак за счёт снижения числа необнаруженных атак (ошибок второго рода) и низкий уровень ложных срабатываний, при этом скорость работы метода зависит лишь от скорости обработки поступающих пакетов.

Литература

1. Moustis D., Kotzanikolaou P. Evaluating security controls against HTTP-based DDoS attacks Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), 2013. URL: ieeexplore.ieee.org/abstract/document/6623707/

2. C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art,” Elsevier Computer Networks, vol. 44, no. 5, pp. 643–665, April 2004.

3. E. Damon, J. Dale, E. Laron, J. Mache, N. Land, and R. Weiss, “Hands-on denial of service lab exercises using slowloris and rudy,” in Proceedings of the

2012 Information Security Curriculum Development Conference, ser. InfoSecCD ’12. New York, NY, USA: ACM, 2012, pp. 21–29. URL: doi.acm.org/10.1145/2390317.2390321

4. Ievgen Duravkin; Anastasiya Loktionova; Anders Carlsson Method of slow-attack detection 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology URL: ieeexplore.ieee.org/document/6992341/

5. Maurizio Aiello; Enrico Cambiaso; Silvia Scaglione; Gianluca Papaleo A similarity based approach for application DoS attacks detection 2013 IEEE Symposium on Computers and Communications (ISCC). URL: ieeexplore.ieee.org/document/6754984/

6. Абрамов Е.С., Сидоров И.Д., Метод обнаружения распределённых информационных воздействий на основе нейронной сети // Известия ЮФУ. Технические науки, 2009, №. 11 (100) С. 154-164.

7. “Slowloris http dos,” URL: ha.ckers.org/slowloris, Tech. Rep.

8. Тарасов Я.В. Модель низкоинтенсивной сетевой атаки "отказ в обслуживании" // МГТУ им. Баумана, В сборнике трудов VII всероссийской научно-технической конференции "Безопасность информационных технологий " (BIT - 2016), с. 75-80.

9. Chuchueva I.A. Model prediction of time series based on a sample of maximum similarity // PhD degree work. M.:2012, 147 c.

10. Fogler H.R. A pattern recognition model for forecasting // Management science. 1974, No.8. pp. 1178 – 1189.

11. Discovering Patterns in Electricity Price Using Clustering Techniques / F. Martinez Alvarez [at al.] // ICREPQ International Conference on Renewable Energies and Power Quality, Spain, Sevilla, 2007: URL: icrepq.com/icrepq07/245-martinez.pdf.

12. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999, 842 p.

13. Lee Giles, Steve Lawrence, Ah Chung Tsoi Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference // Machine Learning July 2001, Volume 44, Issue 1, pp 161–183.

14. Fodor,I. (2002) "A survey of dimension reduction techniques". Center for Applied Scientific Computing, Lawrence Livermore National, Technical Report UCRL-ID-148494, 26 p.

15. Van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). "Visualizing High-Dimensional Data Using t-SNE" (PDF). Journal of Machine Learning Research 9: pp. 2579–2605.

16. Kohonen, T. Self-Organizing Maps. Third, extended edition. Springer, 2001, 487 p.

17. Ghost, A.K., et al. “Detecting Anomalous and Unknown Intrusions Against Programs in Real-Time”. DARPA SBIR Phase I Final Report. Reliable Software Technologies, pp. 259-268.

18. Command-line packet analyzer tcpdump. URL: tcpdump.org/

19. Robert Graham, "What's the max speed on Ethernet?" // URL: blog.erratasec.com/2013/10/whats-max-speed-on-ethernet.html#.UlbwuNK8Dp8

20. Stephen Northcutt, Judy Novak (2002) “Network Intrusion Detection An Analyst's Handbook” // Sams Publishing, 346 pp.

21. “The netfilter iptables project,” URL: netfilter.org/projects/iptables/index.html.

22. “The snort project,” URL: snort.org.

23. “Slowloris dos mitigation guide,” URL:.funtoo.org/wiki/Slowloris DOS Mitigation Guide.

24. “Snort: snort-rules,” URL: snort.org/snort-rules/.

25. Бабенко Г.В., Белов С.В. Анализ трафика TCP/IP на основе методики допустимого порога и отклонения // Инженерный вестник Дона, 2011, №2 URL: ivdon.ru/ru/magazine/archive/n2y2011/446.

26. Георгица И.В., Гончаров С.А., Мохов В.А. Мультиагентное моделирование сетевой атаки типа DDoS // Инженерный вестник Дона, 2013, №3 URL: ivdon.ru/ru/magazine/archive/n3y2013/1852.

References

1. Moustis D., Kotzanikolaou P. Evaluating security controls against HTTP-based DDoS attacks. Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), 2013 URL:http://ieeexplore.ieee.org/abstract/document/6623707/

2. C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art,” Elsevier Computer Networks, vol. 44, no. 5, pp. 643–665, April 2004.

3. E. Damon, J. Dale, E. Laron, J. Mache, N. Land, and R. Weiss, “Hands-on denial of service lab exercises using slowloris and rudy” in Proceedings of the 2012 Information Security Curriculum Development Conference, ser. InfoSecCD ’12. New York, NY, USA: ACM, 2012, pp. 21–29. URL: doi.acm.org/10.1145/2390317.2390321

4. Ievgen Duravkin; Anastasiya Loktionova; Anders Carlsson Method of slow-attack detection 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology URL: ieeexplore.ieee.org/document/6992341/

5. Maurizio Aiello; Enrico Cambiaso; Silvia Scaglione; Gianluca Papaleo A similarity based approach for application DoS attacks detection 2013. IEEE Symposium on Computers and Communications (ISCC) URL: ieeexplore.ieee.org/document/6754984/

6. Abramov E.S., Sidorov I.D. Izvestiya YuFU. Tekhnicheskie nauki, 2009, No. 11 (100).

7. “Slowloris http dos,” URL: ha.ckers.org/slowloris, Tech. Rep.

8. Tarasov Y.V., Model' nizkointensivnoy setevoy ataki "otkaz v obsluzhivanii". MGTU im. Baumana, v sbornike trudov VII vserossijskoj nauchno-tehnicheskoj konferencii "Bezopasnost' informacionnyh tehnologij " (BIT - 2016). Pp. 75-80.

9. Chuchueva I.A. Model prediction of time series based on a sample of maximum similarity PhD degree work. M.:2012, 147 p.

10. Fogler H.R. Management science. 1974, No.8. pp. 1178 – 1189.

11. Discovering Patterns in Electricity Price Using Clustering Techniques. F. Martinez Alvarez [at al.] ICREPQ International Conference on Renewable Energies and Power Quality, Spain, Sevilla, 2007: URL:http://www.icrepq.com/icrepq07/245-martinez.pdf.

12. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999, 842 pages.

13. Lee Giles, Steve Lawrence, Ah Chung Tsoi Machine Learning July 2001, Volume 44, Issue 1, pp. 161–183.

14. Fodor,I. (2002) "A survey of dimension reduction techniques". Center for Applied Scientific Computing, Lawrence Livermore National, Technical Report UCRL-ID-148494, 26 p.

15. Van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). "Visualizing High-Dimensional Data Using t-SNE" (PDF). Journal of Machine Learning Research 9: PP. 2579–2605.

16. Kohonen, T. Self-Organizing Maps. Third, extended edition. Springer, 2001, 487 p.

17. Ghost, A.K., et al. “Detecting Anomalous and Unknown Intrusions Against Programs in Real-Time”. DARPA SBIR Phase I Final Report. Reliable Software Technologies, pp. 259-268.

18. Command-line packet analyzer tcpdump. URL: tcpdump.org/.

19. Robert Graham, "What's the max speed on Ethernet?" URL: blog.erratasec.com/2013/10/whats-max-speed-on-ethernet.html#.UlbwuNK8Dp8

20. Stephen Northcutt, Judy Novak (2002) “Network Intrusion Detection An Analyst's Handbook”. Sams Publishing, 346 pp.

21. “The netfilter iptables project,” URL: netfilter.org/projects/iptables/index.html.

22. “The snort project,” URL: snort.org.

23. “Slowloris dos mitigation guide,” URL: funtoo.org/wiki/Slowloris DOS Mitigation Guide.

24. “Snort: snort-rules,” URL: snort.org/snort-rules/.

25. Babenko G.V., Belov S.V. Inženernyj vestnik Dona (Rus), 2011, №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/446.

26. Georgica I.V., Goncharov S.A., Mohov V.A. Inženernyj vestnik Dona (Rus), 2013, №3. URL: ivdon.ru/ru/magazine/archive/n3y2013/1852.