44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: 8 (8634) 371-905.
The Department of IT Security; post-graduate student.

УДК 681.324

Е.С. Абрамов, И.Д. Сидоров

МЕТОД ОБНАРУЖЕНИЯ РАСПРЕДЕЛЁННЫХ ИНФОРМАЦИОННЫХ ВОЗДЕЙСТВИЙ НА ОСНОВЕ ГИБРИДНОЙ НЕЙРОННОЙ СЕТИ

Описывается метод применения вероятностных суффиксных деревьев для обнаружения аномального поведения программ. Используется «отпечаток» нормального поведения приложений с целью в дальнейшем обнаружить аномальное поведение как нечто, отклоняющееся от модели. В качестве основной модели используется вероятностные суффиксные деревья.

Вероятностное суффиксное дерево; PST; обнаружение аномального поведения.

E.S. Abramov, I.D. Sidorov

METHOD OF DETECTION OF DISTRIBUTED INFORMATION IMPACTS ON THE BASIS OF HYBRID NEURAL NETWORK

Described the method of probabilistic suffix trees for detecting anomalous behavior of programs. Use "fingerprint" of the normal behavior of applications in order to further detect anomalous behavior as something deviating from the model. As a basic model uses a probabilistic suffix trees.

Probabilistic suffix tree; PST; detection of abnormal behavior.

Введение

Большинство современных систем обнаружения атак (СОА) осуществляют обнаружение атак путём контроля профилей поведения либо поиска специфических строковых сигнатур. Используя эти методы, практически невозможно создать полную базу данных, содержащую сигнатуры большинства атак. Существует три главные причины этого:

1. Новые сигнатуры необходимо создавать вручную. Сигнатуры известных атак, которые уже включены в БД, не могут гарантировать надёжной защиты без постоянных обновлений.

2. Теоретически существует бесконечное число методов и вариантов атак, и для их обнаружения понадобится БД бесконечного размера. Таким образом, имеется возможность того, что некая атака, не включённая в базу данных, может быть успешно осуществлена.

3. Современные методы обнаружения вызывают большое число ложных тревог. Таким образом, могут быть скомпрометированы легальные сетевые события.

Основное преимущество систем обнаружения атак, использующих нейронные сети, в том, что нейросеть не ограничена знаниями, которые заложил в неё программист. Они имеют возможность учиться на предшествующих событиях –

как на аномальном, так и на нормальном трафике. За счёт этого достигается высокая эффективность и адаптивность СОА.

Существуют различные варианты применения нейросетевых систем обнаружения атак (НСОА) – анализ всего сетевого трафика в защищаемом сегменте сети, анализ команд, вводимых пользователем, анализ переходов состояний и др. Далее описывается использование гибридной нейронной сети в качестве анализатора трафика.

В работе [1] показана возможность использования многослойного персептрона в качестве решающего блока в СОА, анализирующей последовательно приходящие сетевые пакеты. Описываемый далее подход предполагает использование нейросети для выявления злоупотреблений, рассредоточенных во времени и совместного создания искусственного потока данных несколькими источниками.

Распределенными по времени называются атаки, проводимые в течение одного продолжительного периода времени. При совместном нападении существует несколько злоумышленников, работающих параллельно. Каждый из них по отдельности может предпринимать действия, которые могут показаться безвредными. Атаки становятся очевидными только тогда, когда все события рассматриваются вместе. Обнаружение этих видов атак может быть очень сложным.

Моделировались два типа сетевого трафика – нормальный и аномальный. Первый содержал пакеты, появляющиеся в сети при обычной работе, а второй имитировал распределённую атаку.

Потоки данных были разделены на наборы из 150 пакетов, причём кроме данных из заголовков пакетов сетевого и транспортного уровней, как предлагается в работе [2], учитывались также и первые 50 символов полезной нагрузки прикладного уровня (в ASCII-коде). Кроме того, нормализуется представление порта назначения для уточнения используемого прикладного протокола (telnet, ftp и т.д.). Схожий метод анализа исходных данных был описан в [3].

Примером распределённой по времени атаки может служить взлом пароля методом перебора. Для FTP-сервера трассировка такой атаки будет похожа на следующую:

- `FTP Username login`
- `FTP password attempt`
- `FTP password attempt`
- `FTP password attempt`
- `Disconnect from server`
- `Same FTP Username login`
- `FTP password attempt`
- `FTP password attempt`
- `FTP password attempt`
- `Disconnect from server`
- `Same FTP Username login`
- `FTP password attempt`
- `FTP password attempt`
- `FTP password attempt`
- `Disconnect from server`

Ни одно событие по отдельности не выглядит подозрительно. Кроме того, даже два или три события не являются чем-то необычным. Тем не менее, взятые вместе, в определенной последовательности, повторяемые события являются указанием на то, что осуществляется атака. Дальнейшее осложнение обнаружения состоит в том, что эти события, хотя и в определенной последовательности, могут поступать из различных источников и чередоваться с другими, нормальными событиями.

Для обработки аномального трафика предлагается применять гибридную нейронную сеть, состоящую из самоорганизующейся сети (self-organizing map, SOM) и многослойного персептрона.

SOM используется для отображения 51-символьных векторов в узлы матрицы, в которой события аналогичных числовых символов будут сгруппированы. Фактически некоторые узлы будут представлять собой определённые сценарии атак.

После этого данные заголовков пакетов и информация о группировке подаются на вход многослойного персептрона, обученного распознавать аномальный трафик, но уже с учётом информации о событии, т.е. принадлежности пакета той или иной группе-сценарию. Это позволяет не только обнаруживать аномалии в единичных пакетах, но и выявлять принадлежность пакета к распределённой по времени атаке.

Поток данных разделяется на наборы из 150 пакетов. Учитываются первые 50 символов полезной нагрузки прикладного уровня (в ASCII-коде). Кроме этого, нормализуется порт назначения для уточнения используемого прикладного протокола (telnet, ftp и т.д.), Таким образом, формат вектора примерно следующий:

```
GET / HTTP/1.1 Host: global.ebay.com User-Agent: Mozilla; 80 – для HTTP;
Anonymous; 21 – для ftp.
```

Модуль на основе гибридной нейронной сети включает в себя реализацию двух нейронных сетей – сети Кохонена и многослойного персептрона. Работа с сетями состоит из двух этапов – обучения и распознавания (обнаружения атак).

На этапе обучения программа собирает сетевые пакеты и затем записывает их в бинарном формате Matlab-а. Для этого используется подготовленная с помощью Matlab Compiler .NET-сборка save_vectors.dll. Перед обучением сохранённые пакеты классифицируются экспертом как нормальные или атакующие. Затем на первом этапе обучения сеть Кохонена обучается без учителя для кластеризации данных, на втором этапе обучения многослойный персептрон обучается с учителем методом Левенберга-Маркуорта для классификации наборов пакетов на нормальные и атакующие. Входными данными для сети Кохонена служат сетевые пакеты, представленные следующим образом – один входной вектор соответствует одному сетевому пакету, первый компонент вектора – номер порта, остальные 50 компонентов – первые 50 байтов данных пакета. Компоненты вектора нормированы в диапазоне [0,1]. Входными данными для персептрона являются результаты распознавания группы из 150 пакетов сетью Кохонена. Каждый компонент вектора соответствует кластеру сети Кохонена, а его значение – число пакетов, принадлежащих данному кластеру. Компоненты вектора также нормированы в диапазоне [0,1]. Выходной вектор имеет вид $<y_n,y_a>$, где y_n – близость к нормальному поведению, y_a – близость к атакующему поведению.

На рис. 1 представлена модель гибридной ИНС, на рис. 2 – 4 – результаты кластеризации сетью Кохонена.

На рис. 5 приведены графики зависимости ошибки от эпохи обучения, характеризующие скорость обучения ИНС.

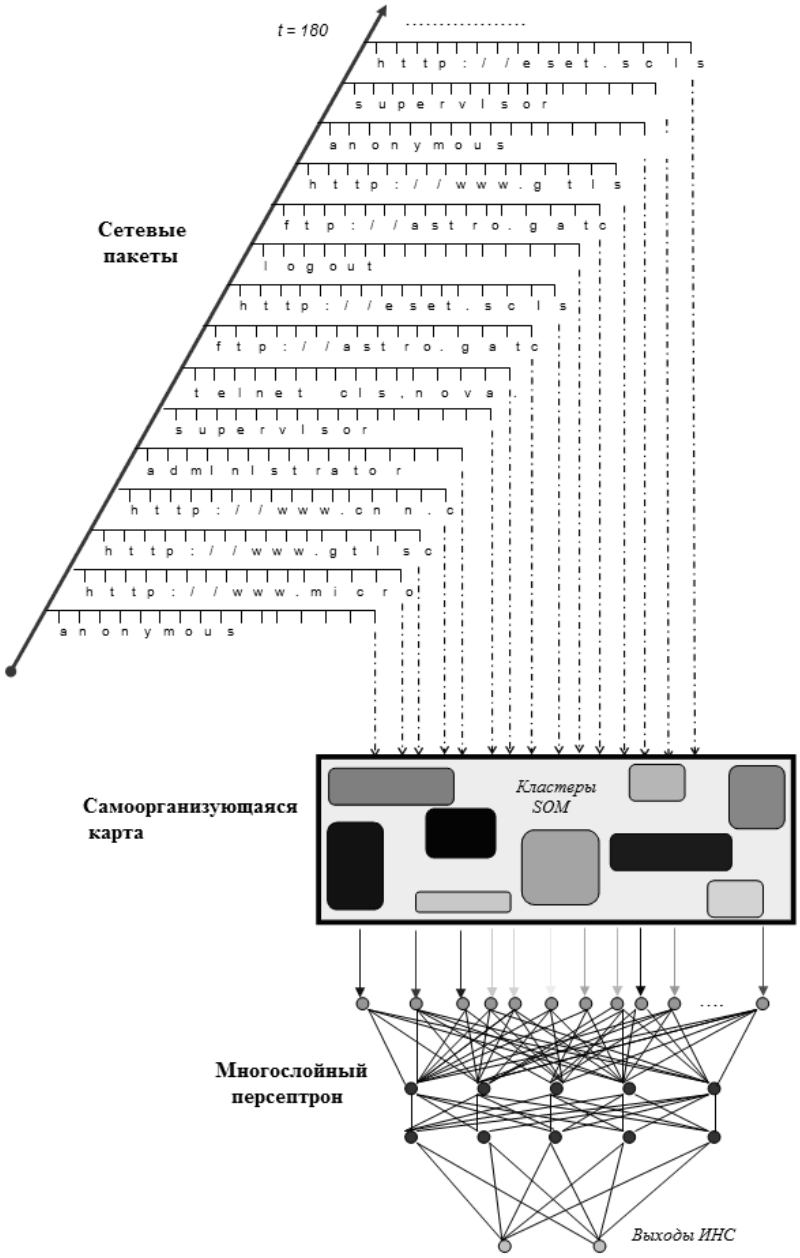

Рис. 1. Архитектура модели системы обнаружения атак

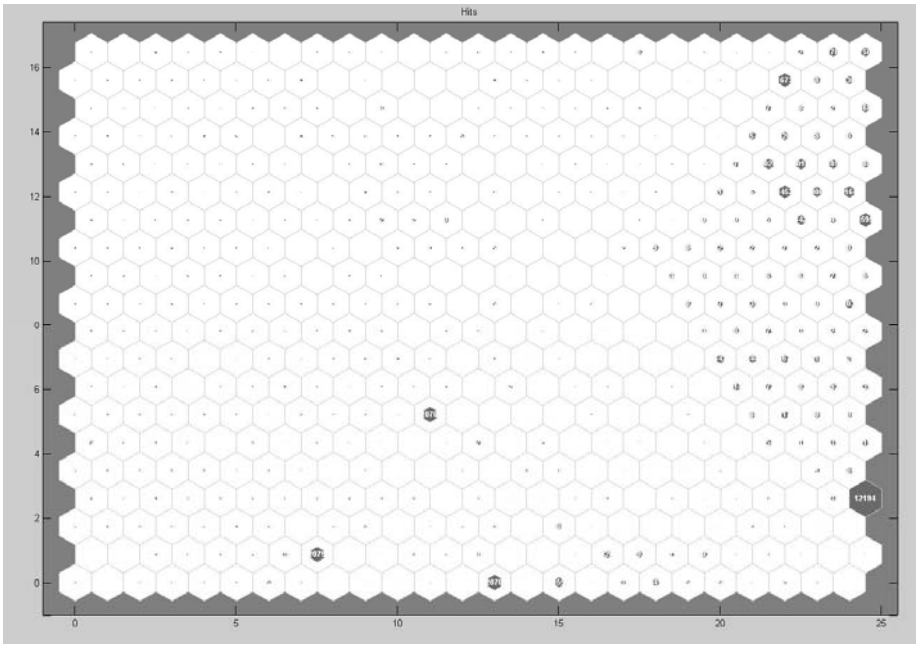

Рис. 2. Результаты классификации пакетов сетью Коханена

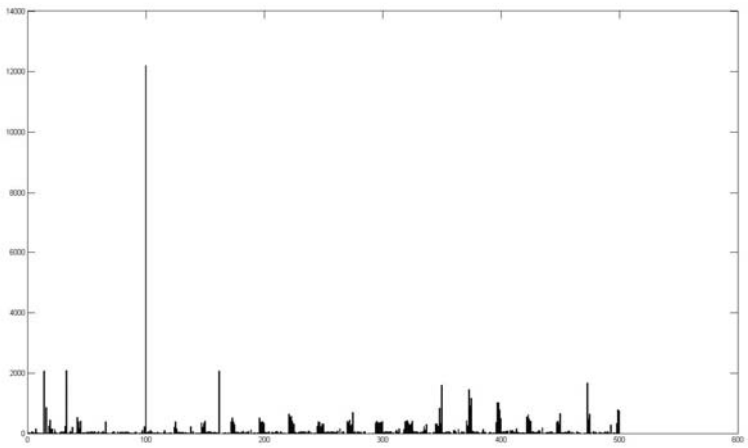

Рис. 3. Гистограмма с номерами нейронов победителей в сети Коханена

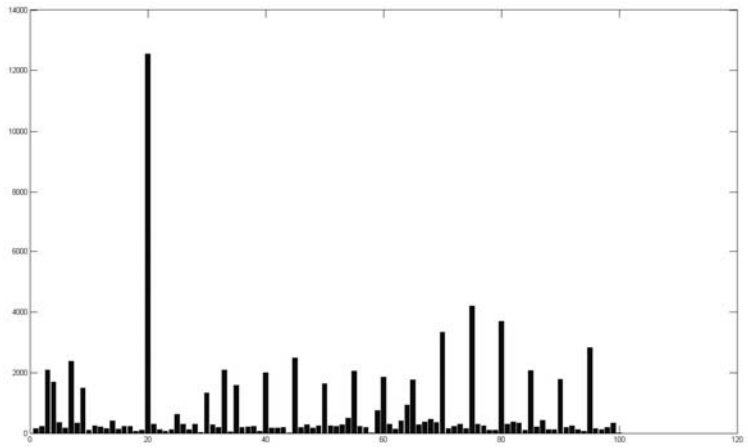

Рис. 4. Гистограмма с номерами нейронов победителей в сети Коханена (нейроны сгруппированы в 100 групп)

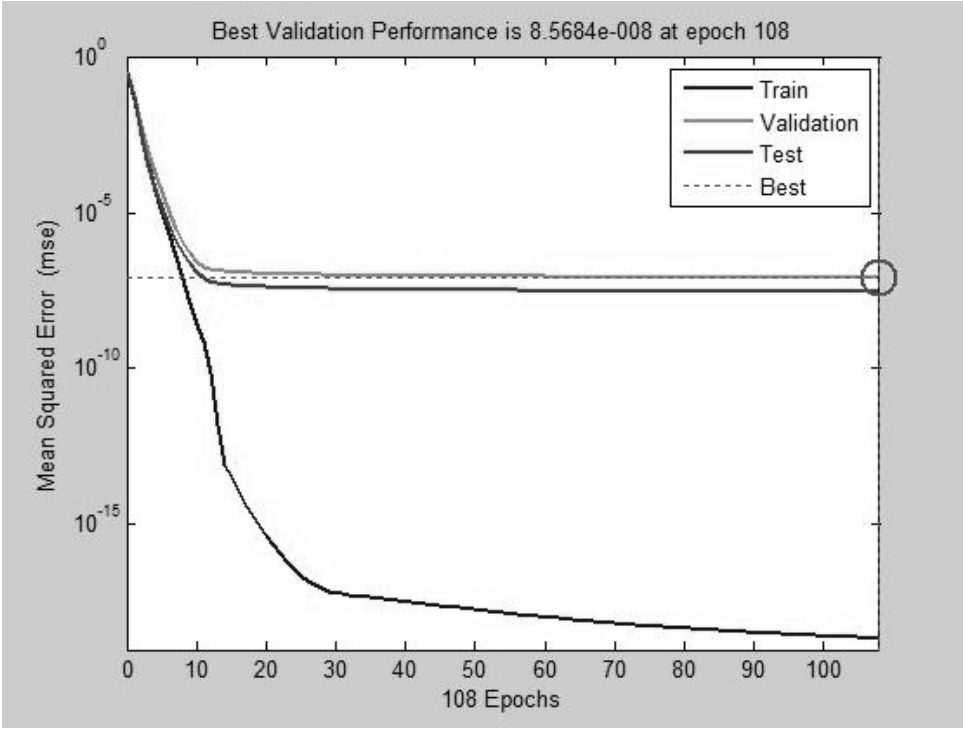

Рис. 5. Ошибка на различных выборках при обучении персептрона

Полученные искусственные нейронные сети преобразуются в текстовый файл, содержащий веса обученных нейронных сетей. Затем эти коэффициенты используются программой для имитации ИНС и классификации полученных сетевых пакетов. Для распознавания в режиме реального времени необходимо в программе включить соответствующий режим установкой галочки «Обнаруживать атаки». При включённом распознавании программа каждые 150 пакетов будет производить классификацию полученных пакетов. Ответы ИНС 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 сеть не может классифицировать пакет.

Результаты анализа («норма», «атака», «невозможно классифицировать») выводятся пользователю в списке каждые 150 пакетов.

Для экспериментов была выбрана конфигурация типичного Web-сервера, на котором запущены службы web-, ftp-, torrent-серверов. Такая конфигурация является одной из наиболее часто встречающихся и соответственно подвергающихся атакам.

Описание экспериментов

1. Этап сбора данных. На этом шаге было собрано 33 000 пакетов нормального трафика (Web, torrent), 33 000 пакетов атакующего трафика (FTP-bruteforce) и 33 000 пакетов трафика, содержащего DDoS-атаку (TCP SYN Flood атака, пакеты с случайным номером порта и нулевыми данными).

2. Этап формирования выборок. На этапе формирования выборок из собранных пакетов были сформированы обучающая (12 000 нормальных + 12 000 атакующих = 24 000 пакетов) и тестовая (остальные пакеты). Размер обучающей выборки лимитировался, в первую очередь, размером памяти и временем, необходимым для обучения карты Кохонена – самой ресурсозатратной части.

3. Этап обучения. Обучение карты Кохонена производится на отдельных пакетах. Время обучения ~50 мин. Перед подачей на сеть Кохонена и на персептрон все данные нормировались в диапазон [0,1]. Сеть Кохонена имеет гексагональную структуру связей нейронов (предлагается Матлабом по умолчанию) и размеры 25 на 20 (всего 25*20=500 кластеров, взято из статьи).

Обучение персептрона производится на группах по 150 пакетов. Применяются два способа формирования векторов. Первый состоит в том, что каждая компонента вектора соответствует сетевому пакету и указывает на то, к какому кластеру сети Кохонена относится данный пакет. Второй состоит в том, что каждая компонента вектора соответствует кластеру сети Кохонена и показывает, сколько пакетов из группы в 150 штук были опознаны как принадлежащие этому кластеру. Первый подход учитывает порядок следования пакетов, второй – его игнорирует.

Размерность входного вектора для персептрона задаётся способом формирования вектора (150 для первого, 500 для второго), выходной вектор имеет размерность 2. Нормальные векторы обучались на выход <1,0>, атакующие – на <0,1>. То есть можно сказать, что первый выход – близость к нормальному поведению, второй – близость к аномальному поведению.

Персептрон имеет следующую структуру – два скрытых и выходной слой, активационная функция в скрытых 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.

4. Этап распознавания. 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 работы модели представлены на рис. 6.

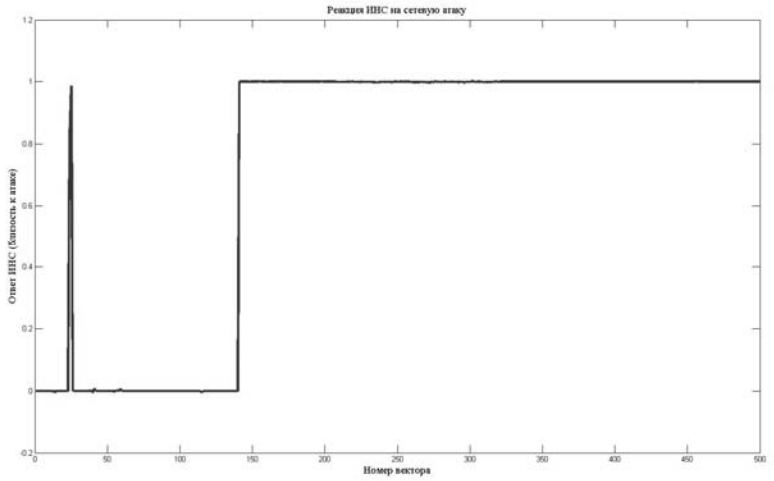

Рис. 6. Результат работы модели СОА (реакция на атаку)

Описания параметров и их значения по умолчанию:

– число пакетов для формирования одного вектора перспептрона packets_on_vector = 150;
– размерность сети Кохонена;
– som_x = 25;
– som_y = 20;
– число нейронов (кластеров) в карте Кохонена;
– som_size = 500;
– число объединяемых вместе кластеров сети Кохонена;
– cluster_step = 5;
– размерность скрытых слоёв ИНС: [21 7];
– активационные функции слоёв ИНС: ['tansig' 'tansig' 'purelin'].

Таблица 1

Результаты распознавания для первого способа, все векторы

	Общее число векторов	Верно классифицировано	Неверно классифицировано	Процент ошибки
Нормальные	140	99	56	40%
Атака	360	326	34	9,4%
Всего	500	410	90	18%

Ошибка первого рода составила 40%, второго – 9,4%, общая ошибка – 18%.

Таблица 2

Результаты распознавания для первого способа, норма и FTP brute

	Общее число векторов	Верно классифицировано	Неверно классифицировано	Процент ошибки
Нормальные	140	99	41	29,3%
Атака	140	137	3	2,1%
Всего	280	236	44	15,7%

Ошибка первого рода составила 29,3%, второго – 2,1%, общая ошибка – 15,7%.

Таблица 3

Результаты распознавания для второго способа, все вектора

	Общее число векторов	Верно классифицировано	Неверно классифицировано	Процент ошибки
Нормальные	140	138	2	1,4%
Атака	360	360	0	0%
Всего	500	498	2	0,4%

Ошибка первого рода составила 1,4%, второго – 0%, общая ошибка – 0,4%.

Сравнение с аналогами

В качестве аналога была выбрана свободно распространяемая система обнаружения атак Snort. Snort – облегченная система обнаружения вторжения, потому что разработана, прежде всего, для небольших локальных сетей. Программа может исполнять анализ протокола и может использоваться, чтобы обнаружить разнообразные нападения и исследовать проблемы типа переполнения буфера, скрытых просмотров порта, CGI нападения, попыток определения OS и т.п. Программа также имеет механизм обнаружения, который использует модульную сменную архитектуру, посредством чего определенные дополнения к программе могут быть добавлены или удалены из механизма обнаружения.

Snort может работать в трех режимах:

1. Как пакетный снифер, подобно TCPdump.
2. Как регистратор пакетов.
3. Как развитая система обнаружения вторжения.

Система не имеет графического интерфейса. Предупреждения об атаках могут сохраняться в текстовый файл или же в базу данных на основе MySQL, в том числе и с использованием удалённого доступа.

Snort использует правила (указанные в файлах «правил»), чтобы знать какой трафик пропустить, а какой задержать. Этот инструмент достаточно гибок, позволяет записывать новые правила и соблюдать их. Параметры правила по сути представляют собой сложные сигнатуры и включают в себя как данные из заголовков пакетов, так и строковые данные из поля данных пакетов. Такой формат правила позволяет контролировать аномалии трафика и вести сигнатурный поиск.

Существует мнение, что специальные средства для обнаружения DoS-атак не требуются, поскольку факт DoS-атаки невозможно не заметить. Во многих случаях это действительно так. Однако достаточно часто отмечались успешные атаки, которые были замечены жертвами лишь через 2-3 суток. Бывало, что негативные последствия атаки (типа *флуд*) заключались в излишних расходах по оплате трафика, что выяснялось лишь при получении счёта. Кроме того, многие методы обнаружения атак неэффективны вблизи цели атаки, но эффективны на магистральной сети. Методы обнаружения можно разделить на несколько больших групп.

1. Анализ сессий и периодов активности. Данная эвристика основана на том, что ресурс используется людьми, которым присущи индивидуальные особенности работы в сети. Большинство людей имеет определенный распорядок дня (работа, учеба, семья), который определяет время сетевой активности. Также каждому человеку присущи определенные интересы, это тоже можно использовать для формирования его «портрета поведения» в сети. Благодаря особенностям темперамента и характера человека, можно определить такие параметры, как частота запросов в сессии, промежутки времени между запросами и т.д. Итак, для каждого пользователя ресурса можно определить характерный только для него «портрет поведения».

2. Анализ суточной статистики. В основе алгоритмов лежит разбиение аудитории ресурса на зоны, характеризующиеся постоянными периодами активности. По сути, данный метод является обобщением предыдущей эвристики.

3. Анализ заголовков сетевых пакетов. Для повышения эффективности атаки злоумышленник может посылать сетевые пакеты, содержащие нестандартные значения служебных полей. Данная эвристика направлена на выявление таких пакетов среди общего сетевого трафика.

Правила Snort для обнаружения распределённой атаки «перебор паролей»:

```
# Druber - ftp hack
alert tcp any any -> $HOME_NET 21 ( msg:"FTP Potential Brute
Force Attack"; flow:to_server; flags:S; threshold:type thre-
shold, track by_src, count 20,
seconds 60; classtype:suspicious-login; sid:3000002; rev:5; )

# cover other ftp daemons like freeftpd and warftpd
alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (msg:"ET SCAN Po-
tential FTP Brute-Force attempt";
flow:from_server,established; dsize:<100; content:"530 ";
depth:4; pcre:"/530\s+(Login|User|Failed|Not)/smi"; class-
type:unsuccessful-user; threshold: type threshold, track
by_dst, count 5, seconds 300; sid:2002383; rev:10;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:“FTP sa brute
force failed login unicode attempt”;
flow:to_server,established; content:“PASS”; reference:url;
threshold:type threshold, track by
_src, count 6000, seconds 1200; priority: 1; class-
type:attempted-user; sid:2000001;)
```

Из анализа правил видно, что для эффективного противодействия необходимо знать тип, характер и другие показатели DoS-атаки, а оперативно получить эти сведения как раз и позволяют системы обнаружения распределённых атак.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Cannady, J.* (1998) Applying Neural Networks to Misuse Detection. Proceedings of the 21st National Information Systems Security conference. P. 368 – 381.

2. *Абрамов Е.С., Макаревич О.Б., Бабенко Л.К., Пескова О.Ю.* Разработка архитектуры СОА на основе нейронной сети // Материалы VI Международной научно-практической конференции «Информационная безопасность», – Таганрог. – 2004. – С. 81 – 86.

3. *Райан Д., Менг-Джанг Лиин* Обнаружение атак с помощью нейросетей.- Режим доступа [http://neurnews.iu4.bmstu.ru/, свободный], 03.12.2009.

Абрамов Евгений Сергеевич
Технологический институт Федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: juic@mail.ru.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 8(8634)371-905.
Кафедра безопасности информационных технологий; доцент.

Abramov Eugene Sergeevich
Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.
E-mail: juic@mail.ru.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: 8(8634) 371-905.
The Department of IT Security; associate professor.

Сидоров Игорь Дмитриевич
Технологический институт Федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: idsidorov@gmail.com.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 8(8634)371-905.
Кафедра безопасности информационных технологий; ассистент.

Igor D. Sidorov
Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.
E-mail: idsidorov@gmail.com.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: 8(8634) 371-905.
The Department of IT Security; assistant.

УДК 004.056.53

А.П. Росенко

ОБ ОДНОМ ПОДХОДЕ К ОПРЕДЕЛЕНИЮ ВЕРОЯТНОСТЕЙ ПОСЛЕДСТВИЙ ОТ ВОЗДЕЙСТВИЯ НА АИС УГРОЗ БЕЗОПАСНОСТИ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ

В работе предложен один из подходов к определению вероятностей последствий от воздействия на АИС угроз безопасности конфиденциальной информации. Показано, что процесс воздействия на АИС различных дестабилизирующих