Анализ сетевого трафика с помощью нейронных сетей

Введение

В современном мире наиболее актуальны вопросы информационной безопасности в компьютерных системах. Основными способами защиты от несанкционированного доступа на данных момент является антивирусное ПО и межсетевые экраны или файрволы (firewalls). Однако такое ПО действует по строго определенным алгоритмам, из-за чего они не способны распознавать угрозы новых типов. Искусственные нейронные сети, напротив, после обучения способны адаптироваться под новые типы угроз, распознавая их, ни разу с ними до этого не сталкиваясь. Данная особенность позволяет системе защиты стать более гибкой и независимой.

Целью нашей работы является разработка нейронной сети с помощью пакета MATLAB, способной в реальном времени проанализировать трафик на наличие в нём угроз безопасности для системы. Данная сеть, например, может быть размещена на сервере и адаптирована под защиту от DDoS атак.

Для анализа возможности обучения сети мы выбрали базу данных, созданную университетом MIT [2], содержащая в себе сведения об атаках различных типов. Все атаки в базе разделены на 4 основных категории:

- DOS(denial-of-service) - отказ в доступе, например, syn flood (*прим. переводчика: рассылка спама*);
- R2l - удаленный доступ с неавторизированной машины, например для получения пароля;
- U2R - попытки несанкционированного доступа к суперпользователю (root), например, различные атаки типа "buffer overflow" (переполнение буфера);
- Probing: разного рода наблюдения и зондирование, например, прослушка портов.

Всего в базе данных представлено 24 типа атак, из которых всего 10 имеют достаточное количество примеров для обучения.

Основная идея нашей работы заключается в выявлении характерных «признаков» у входящего трафика, по которым будет возможно распознавать «хороший» и «плохой» трафики и самостоятельно принимать решение о разрешении доступа к данным. Полный список и описание «признаков» вы можете найти в приложении А.

Архитектура сети

Согласно [3], по умолчанию, для задач классификации следует выбирать следующие параметры сети:

- Количество входных нейронов равно размерности входного вектора
- Количество выходных нейронов – числу классов для распознавания
- Скрытые слои:
  - количество слоёв 1;
  - количество нейронов должно быть равным или в 1.5 - 2 раза превышать размер входного слоя;
  - увеличение количества нейронов в скрытом слое ведет к увеличению производительности сети, но чем их больше, тем больше требуется вычислительных мощностей;

Для реализации нашей задачи была выбрана сеть прямого распространения с одним скрытым слоем нейронов. Причина такого выбора заключалась в простоте представления, а также относительно небольшом количестве потребляемых ресурсов, что является критическим моментом, т.к. моделирование производилось на ПК, располагающем ограниченным количеством памяти.

Обучение сети

Входными данными для нейронной сети будут

Для обучения сети входная выборка разбивается на 3 части в пропорциях 60/20/20:

- Training set – на этих примерах будет происходить обучение сети;
- Test set – на этом наборе данных будет тестироваться сеть;
- Validation set – набор данных для настройки параметров сети.

Такое разбиение обусловлено тем, что при слишком большом training set, мы будем получать наименьшую ошибку при обучении. Однако при поступлении новых данных будет наблюдаться так называемое «переобучение», т.е. бо́льшая на новых данных, поэтому часть обучающей выборки[1].

Результаты

На данный момент была разработана программа в пакете MatLAB, позволяющая создать нейронную сеть заданной архитектуры. Программа позволяет выбрать необходимое количество признаков, для распознавания атак, позволяет задавать количество примеров обучающей выборки, а также имеет возможности для проведения тестирования сети. Интерфейс программы представлен на рис. 1.

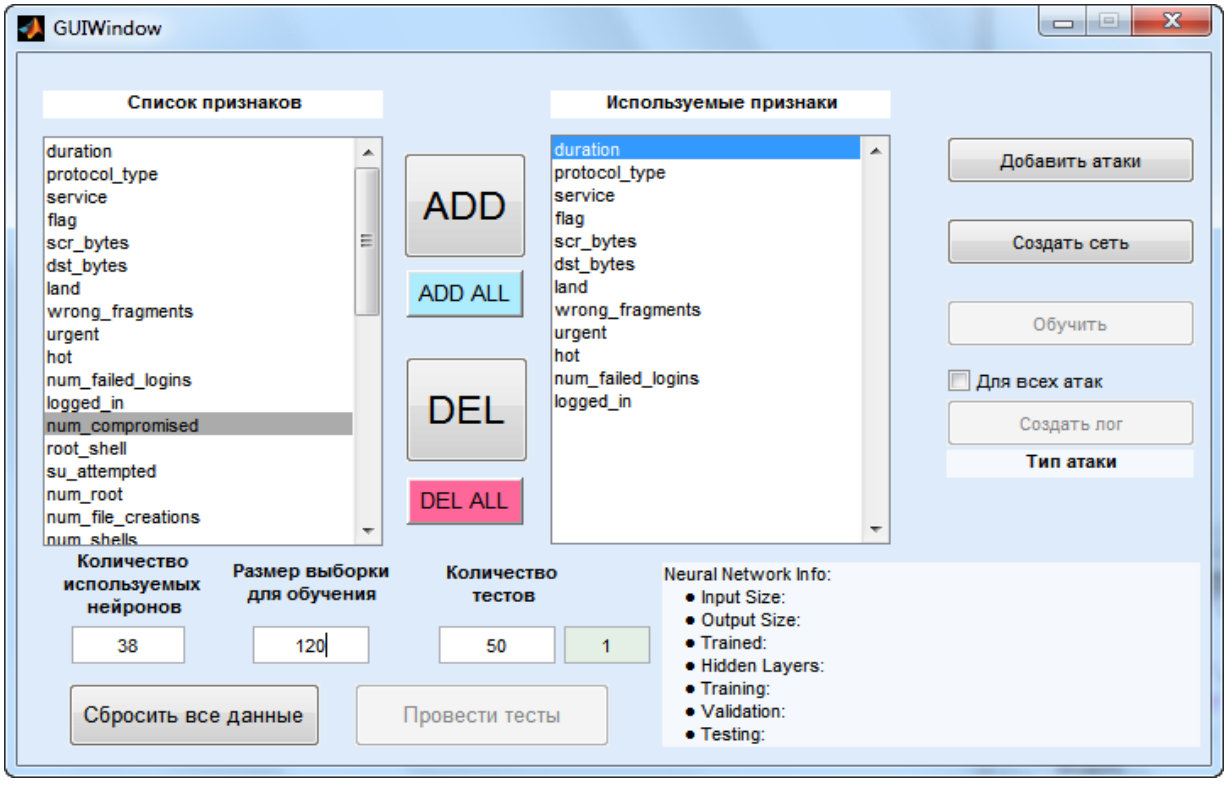

Рисунок 1 – Интерфейс программы

Для работы с программой необходимо загрузить файлы, содержащие векторы с данными, которые описывают атаки (далее *feature-векторы*). Т.к. базы данных с атаками представляют собой большой список подобных *feature-векторов*, мы не может считать все примеры для обучения, т.к. мы ограничены вычислительными возможностями и объемом памяти, поэтому программа разбивает их считывание на 3 части. Первая порция берется из самого начала файла, вторая с некоторым смещением ближе к середине файла, третья – к концу. Тем самым мы «охватываем» всю базу целиком.

Затем необходимо создать сеть, по заданным параметрам. Результат создания сети по данным на рис 1. можно посмотреть на рис 2.

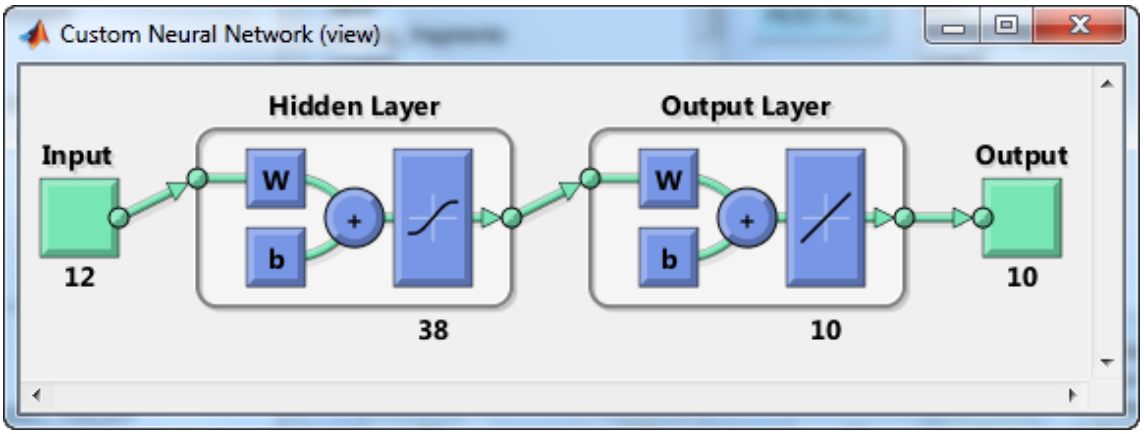

Рисунок 2 – Пример архитектуры нейронной сети

Созданная сеть (сеть 12-38-10) имеет 12 входных нейронов, описывающих *feature-векторы*, которые состоят из тех «признаков», которые мы выбрали, 38 нейронов в скрытом слое и 10 выходных нейронов, которые надо рассматривать как выходные векторы размером 1x10). Нейронная сеть принимает *feature-вектор*, обрабатывает его и формирует выходной вектор. Для определения типа атаки, определяется номер максимального элемента в данном векторе.

Стоит заметить, что сеть 12-38-10 взята для примера. В нашей работе мы используем сеть 38-38-10. Выбор такого количества признаков обусловлен лишь тем, что мы стараемся рассматривать атаку как можно более подробно, для снижения ошибки распознавания. В дальнейшем планируется провести серию экспериментов, чтобы выявить наиболее существенные из признаков. Это позволит снизить количество используемых нейронов, соответственно, снизив общую нагрузку на работающую систему.

Тесты проводятся следующим образом: пользователь задает количество тестовых примеров N, которые будут загружены в сеть за один тест. Устанавливает количество повторений тестов M (зеленый Textbox) и запускает тестирование. Программа автоматически случайным образом считывает N x M *feature-векторов*, которые порциями по N атак подаются на вход сети. Результаты тестирования представляют собой текстовые файлы, содержащие информацию о количестве примеров в одном тесте, а также M записей о доле ошибок в каждом тесте.

Заключение

Проведенные нами тесты сейчас находятся в стадии обработки, но предварительная примерная оценка показывает, что нейронная сеть способна принимать правильные решения с относительной ошибкой 2-6%.

В дальнейшем планируется доработать архитектуру сети, а также выбрать наиболее эффективные «признаки», по которым будет обучаться сеть, уменьшив тем самым количество ошибок на выходе сети.

Планируется дальнейшее развитие программы, её применение на сетевом трафике в режиме реального времени.

Литература

1. С. Хайкин – Нейронные сети. Полный курс. Второе издание, – Издательский дом «Вильямс», 2006 г. - 1104 стр.

2. База данных университета MIT http://kdd.ics.uci.edu/databases/kddcup99/

3. Andrew Ng Lection 09: Neural networks – Learning http://www.holehouse.org/mlclass/09_Neural_Networks_Learning.html

Приложение А.

Список «признаков» для распознавания атак

Название	Описание
duration	Продолжительность соединения в секундах
protocol_type	Тип используемого протокола, т.е. TCP, UDP и пр.
service	Тип используемого обслуживания, т.е. http, ftp, telnet и пр.
flag	Флаг соединения: норма или ошибка
scr_bytes	Число байт данных от источника к получателю
dst_bytes	Число байт данных от получателя к источнику
land	1 если соединение из/на таком же хосте/порте
wrong_fragments	Количество "неверных" фрагментов
urgent	Количество срочных (urgent) пакетов
hot	Число "горячих" индикаторов
num_failed_logins	Число ошибочных попыток входа
logged_in	1 - успешный вход, 0 в противном случае
num_compromised	Число скомпрометированных условий
root_shell	1 - если получена корневая оболочка, 0 - в противном случае
su_attempted	1 - если была попытка выполнить "su root", 0 - в др. случае;
num_root	Число доступов типа "root"
num_file_creations	Число операций создания файла
num_shells	Число "подсказок оболочки"
num_access_files	Число получений доступа к контролю над файлами
num_outbound_cmds	Количество исходящих команд через FTP сессию
is_host_login	1 - если логин принадлежит к "host" списку
is_quest_login	1 - если подключение типа "гость"
count	Число подключений к этому хосту за последние 2 секунды
srv_count	Число подключений к этому сервису за последние 2 сек.
serror_rate	Процент подключений с SYN ошибками
srv_serror_rate	Процент подключений к сервису с SYN ошибками
rerror_rate	Процент подключений с REJ ошибками
srv_rerror_rate	Процент подключений к сервису с REJ ошибками
same_srv_rate	Процент подключений к такому сервису
diff_srv_rate	Процент подключений к различным сервисам
srv_diff_hast_rate	Процент подключений к различным хостам
dst_host_count	Количество соединений к локальному хосту, установленных удаленной стороной
dst_host_srv_count	Количество соединений к локальному хосту, установленных удаленной стороной и использующих одну и ту же службу
dst_host_same_srv_rate	Процентное число соединений к локальному хосту, установленных удаленной стороной и использующих одну и ту же службу
dst_host_diff_srv_rate	Процентное число соединений к локальному хосту, установленных удаленной стороной и использующих различные службы
dst_host_same_src_port_rate	Процентное число соединений к данному хосту при текущем номере порта источника

dst_host_srv_diff_host_rate	Процентное число соединений к службе разных хостов
dst_host_serror_rate	Процентное число соединений с ошибкой типа syn для данного хоста-приемника
dst_host_srv_serror_rate	Процентное число соединений с ошибкой типа SYN для данной службы приемника
dst_host_rerror_rate	Процентное число соединений с ошибкой типа REJ для данного хоста-приемника
dst_host_srv_rerror_rate	Процентное число соединений с ошибкой типа REJ для данной службы приемника