УДК 519.6
Резник А.М., Куссуль Н.Н., Соколов А.М.

НЕЙРОСЕТЕВАЯ ИДЕНТИФИКАЦИЯ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ КОМПЬЮТЕРНЫХ СИСТЕМ

В последнее время работа многих организаций, компаний и обычных пользователей стала в значительной мере зависеть от надежного функционирования компьютерных систем. На компьютерные системы возлагается задача сохранения и обработки данных, которые зачастую имеют очень большую ценность. Получение несанкционированного доступа к ним, их уничтожение, изменение или разглашение, нелегальное использование ресурсов системы или приведение системы в недееспособное состояние — все это имеет крайне нежелательные последствия для владельцев этой информации. Поэтому проблема защиты компьютерных систем от злонамеренного вторжения является чрезвычайно актуальной.

В данной статье анализируются основные подходы к построению систем выявления вторжений, при этом особое внимание уделяется вопросам применения нейронных сетей в системах выявления аномалий. Предлагается новый подход к выявлению аномального поведения пользователя компьютерной системы на основе последовательности выполняемых им команд. Эффективность предлагаемого подхода подтверждается результатами вычислительных экспериментов, проведенных в реальной компьютерной сети Национального технического университета Украины "КПИ".

Анализ существующих подходов к защите информации в компьютерных системах. В настоящее время существует большое число механизмов защиты, которые отличаются как самим подходом к защите компьютерной системы, так и конкретными способами его реализации. Защищать компьютерную систему можно двумя путями. Первый — стараться построить абсолютно защищенную систему, то есть идти путем усложнения процессов аутентификации, введения сложных механизмов прав доступа и создания систем, которые постоянно контролируют их соблюдение. Однако такой путь имеет несколько недостатков. Во-первых, это незащищенность от собственных пользователей со злонамеренными целями. Во-вторых, сами протоколы аутентификации имеют слабые места, а пароли можно украсть или подобрать. В третьих, невозможно создать абсолютно защищенную систему, т.к. в программном обеспечении всегда имеются ошибки и слабые места.

Второй путь состоит в применении не слишком сложных механизмов аутентификации и выявлении аномалий поведения пользователя или атак на компьютерную систему в реальном времени. Именно для этого и предназначены системы выявления вторжений (intrusion detection systems — IDS). IDS обычно делят на системы, которые реагируют на уже известные атаки (так называемые системы выявления злоупотреблений) и системы выявления неизвестных ранее аномалий. Для выявления злоупотреблений обычно используются экспертные системы, работа которых основана на реализации набора известных заранее правил логического вывода [1, 2]. Работа существующих систем выявления злоупотреблений базируется на составлении шаблонов, или так называемых “подписей”, известных атак. Затем с

использованием экспертной системы в реальном времени производится анализ происходящих в системе событий и проверка их на схожесть с известными шаблонами. При этом последовательно выполняется серия проверок, позволяющих выявить возможную атаку. Следует отметить, что атаки по известным схемам встречаются довольно часто, поскольку в Internet существует большое количество хакерских узлов, где каждый желающий может получить необходимую для этого информацию. Защитные системы этого типа безусловно эффективны при выявлении известных типов атак, тем не менее при некоторых отклонениях хода атаки от определенного шаблона возникают серьезные проблемы при их выявлении. Чтобы предотвратить подобные ситуации, необходимо поддерживать слишком большую базу данных по каждой известной атаке и ее вариациям. В случае же неизвестной атаки использование систем, основанных на таких принципах, становится и вовсе нецелесообразным.

Системы выявления аномалий, в отличие от экспертных систем, являются более гибкими. Они строятся на предположении, что все действия злоумышленника обязательно чем-то отличаются от поведения обычного пользователя. Иными словами, такие действия можно рассматривать как аномалии поведения. Работе системы выявления аномалий предшествует период накопления информации, в течение которого составляется некоторая концепция нормальной активности системы или пользователя. Она считается эталоном, относительно которого оцениваются все последующие действия. На этом этапе обычно определяется перечень факторов, по которым можно вести наблюдение за деятельностью пользователей в системе. Теоретически, после составления шаблона нормального поведения, можно фиксировать все параметры системы (или их необходимую часть) и сигнализировать об отклонении этих параметров от обычных значений. Однако объем информации, генерируемой в больших системах (файлы аудита, поток данных в компьютерных сетях и тому подобная информация) может достигать нескольких мегабайт за час и, разумеется, человек не в состоянии вручную обработать такое количество данных. Кроме того, сама постановка задачи — выявить аномальное поведение пользователя, чем-либо отличающееся от обычного — плохо формализуется. Поэтому в последнее время все чаще предпринимаются попытки анализа аномального поведения пользователей на основе интеллектуальных методов обработки данных, в том числе с применением нейронных сетей [3-8].

Нейронные сети в системах выявления аномалий. Искусственные нейронные сети активно применяются для решения различных задач защиты информации в компьютерных системах. Так в [6] предлагается использовать сеть Хопфилда для контроля правильности ввода пароля пользователя компьютерной сети. В [7] приводятся результаты применения нейронных сетей для классификации пользователей компьютерной системы по “компьютерному почерку“, т.е. по характерным для данного пользователя временным задержкам между нажатиями определенных клавиш. Эта задача решалась с использованием многослойной нейронной сети прямого распространения, обучаемой по методу обратного распространения ошибки. В [8] многослойная нейронная сеть персептронного типа применялась для идентификации событий в сетях на базе протокола TCP/IP и выявления злоупотреблений.

Идея применения нейронных сетей в IDS, в частности в системах выявления аномалий, состоит в том, чтобы настроить сеть на некотором “обучающем“ множестве значений вход-выход, которое характеризует поведение пользователя или системы вообще. В качестве выхода сети может служить некоторый принятый коэффициент

“нормальности поведения“ или один из параметров системы. В процессе настройки нейронной сети выявляются и фиксируются скрытые закономерности, присутствующие во входных данных. После обучения такая нейронная сеть сможет анализировать поступающую информацию о работе пользователей и выявлять отклонения от их привычного поведения в системе. Если реальное поведение пользователя отличается от ожидаемого нейронной сетью, то делается вывод о наличии аномалии в системе и возможном нарушении ее безопасности.

Так в работе [3] для построения шаблона поведения пользователя выбирались такие параметры: часы, когда пользователь обычно работает, набор внутренних узлов, с которых он начинает рабочую сессию, характеристики использования ресурсов системы и тому подобное. Эти параметры оцифровывались и в процессе обучения подавались на вход нейронной сети. Выходом сети служил единственный нейрон, принимающий значение 0 для пользователя с нормальным поведением, и значение 1 для поведения, "не похожего" на нормальное. Иными словами, сеть обучалась на парах типа (“нормальные“ параметры, 0) и (“аномальные“ параметры, 1). Поскольку для получения "ненормального" поведения нужно было бы заставить пользователя специально вести себя необычным образом, то авторы генерировали "аномальные" данные с использованием датчика случайных чисел.

Предложенный в работе [5] нейросетевой детектор атак NNID (Neural Network Intrusion Detector) идентифицирует пользователей на основе ограниченного количества команд (100 штук), выполняемых им в течение дня. При этом учитывается только количество запусков каждой команды, а их последовательность никак не учитывается. Эти величины определенным образом кодируются и подаются на вход нейронной сети. Таким образом система запоминает “шаблон“ поведения каждого из своих пользователей. Если поведение реального пользователя в сеансе работы не соответствует ни одному из шаблонов, запомненных нейронной сетью, система уведомляет об этом администратора. В компьютерной системе с 10 пользователями предложенный детектор показал неплохие результаты, но в реальных системах количество пользователей может достигать нескольких тысяч, причем большинство из них, как правило, выполняют однотипные действия. Поэтому использование одной сети и подхода, учитывающего только частоту выполнения команд без учета порядка их следования, для идентификации всех пользователей нецелесообразно.

Идентификация поведения пользователя на основе последовательностей выполняемых им команд. Для более полного и совершенного построения шаблона поведения пользователя предлагается, наряду с перечнем команд пользователя учитывать последовательность их выполнения. Это целиком оправдано, поскольку пользователи отличаются друг от друга не только набором "излюбленных" команд, но и характерным порядком их использования. Информация о последовательности действий пользователя в компьютерной системе отображается в файлах аудита (журналах системных событий), формат и содержание которых зависит от операционной системы. Предлагаемый подход был реализован для операционной системы UNIX, а именно Free BSD, поскольку это одна из наиболее популярных и защищенных систем, причем, в отличие от Windows, информация в ее файлах аудита хранится в текстовом виде. Однако это не ограничивает общности полученных результатов, поскольку для их использования на базе других платформ требуется только соответствующая “перекодировка” входных данных.

Идея предлагаемого подхода состоит в следующем. Для каждого пользователя в системе строится нейронная сеть прямого распространения (настраиваемая по правилу

обратного распространения ошибки) [9], которая обучается прогнозировать следующую команду данного пользователя на основе m предшествующих ей команд. Обученная таким образом нейросеть сможет моделировать поведение данного пользователя. Если реальное поведение пользователя в системе существенно отличается от “прогнозируемого” сетью поведения, значит с высокой вероятностью можно утверждать, что под данным именем в системе зарегистрировался посторонний пользователь.

Более строго задача формулируется следующим образом. Перенумеруем все возможные команды в системе. Рассмотрим последовательность команд, выполненных одним пользователем системы в течение одного сеанса работы (одного дня)

$$\{c_1, \cdots\cdots, c_i, c_{i+1}, \cdots, c_{i+m}, c_{i+m+1}, \cdots\cdots, c_n\},$$

где c_i — номер i-й команды в данном сеансе работы. Выделим плавающее окно (кадр) размером в m команд, подаваемых на вход нейронной сети. Ставится задача такого обучения сети, чтобы для каждого кадра (входного вектора из m элементов) на выходе нейронной сети получать следующую $(m+1)$-ю команду. Например, при $m = 2$ после последовательности команд типа (vi, c++) или (joe, gcc) логично ожидать команду вида a.out.

Таким образом, при обучении нейронной сети ей предъявляются пары вида (X_i, T_i), где $X_i = (c_i, c_{i+1}, \cdots, c_{i+m})$, $T_i = c_{i+m+1}$, $i = \overline{1, n_k - m - 1}$, $k = \overline{1, p}$, p — число сеансов, данные которых используются для обучения сети (сеансов обучения), n_k — число команд в k-м сеансе работы пользователя. Такая постановка довольно типична для задач прогнозирования, решаемых с использованием многослойной нейронной сети персептронного типа [9], структура которой показана на рис. 1. Для обучения нейронной сети можно использовать любую известную модификацию метода обратного распространения ошибки, в частности алгоритм Quick propagation или Delta-bar-Delta [10].

После обучения нейронной сети на данных нескольких сеансов обучения с ее помощью можно анализировать деятельность пользователя в сети по информации файла аудита любого другого сеанса. В режиме тестирования на вход нейронной сети также предъявляется m команд, а полученное значение выхода сети сравнивается с *реальной* $(m+1)$-й командой. Если относительное количество правильно предсказанных команд на протяжении всего сеанса выше заданного порога, то поведение пользователя считается ″нормальным", если нет — значит пользователь либо резко изменил свое поведение либо под его именем работает другой пользователь. Поскольку пользователям свойственно изменять поведение с течением времени, то в процессе работы для обеспечения адаптации к их поведению, требуется периодически дообучать сеть.

Такой подход обладает следующими преимуществами:

- независимость от количества пользователей в системе, поскольку с каждым пользователем связывается отдельная нейронная сеть;
- возможность выявления тонких закономерностей в поведении пользователя благодаря использованию информации не только о статистике выполнения команд но и об их последовательности;
- возможность приспособления к изменяемому поведению пользователей;
- возможность обучения сети на реальных данных без необходимости генерировать случайно или придумывать "аномальные" данные.

Кодирование и сравнение команд. Остановимся на вопросе представления и анализа информации на входе и выходе нейронной сети. Поскольку явный критерий для упорядочения перечня возможных команд отсутствует, то команды можно перенумеровать в произвольном порядке. С целью повышения эффективности работы нейронной сети в качестве входной и выходной информации для нейронной сети целесообразно использовать двоичную p-разрядную запись номеров команд или их температурное кодирование. Следовательно, для решения этой задачи требуется многослойная нейронная сеть прямого распространения, имеющая $p \times m$ бинарных входов и p выходов. Например, для 512 команд $p = 9$.

Сравнение выхода нейронной сети и реальной следующей команды пользователя в сеансе работы выполняется покомпонентно, т.е. если

$$\forall i = \overline{1,p} \quad \left| y_i - d_i \right| \le \frac{1}{2},$$

где y_i — значение i-го выходного нейрона сети, а d_i — i-й компонент двоичной записи номера реальной следующей команды, то можно считать, что следующая команда предсказана сетью правильно.

Результаты экспериментов. Тестирование предложенного подхода проводилось в реальных условиях в компьютерной лаборатории физико-технического факультета на UNIX-сервере K6 II на протяжении 35-ти дней, а некоторые эксперименты на отдельном компьютере 486DX4-100. При этом использовались операционные системы FreeBSD 4. 0-STABLE и FreeBSD 2.2.7-RELEASE соответственно. Общее количество пользователей, зарегистрированных в системе, — 414. Из них во время тестирования были активны 172. Всего была собрана информация о 1900 рабочих сеансах.

Такой выбор операционной системы определялся следующими причинами:

- во-первых, она имеет уже встроенный в ядро механизм аудита активности пользователей и можно обойтись без создания собственной сложной системы, фиксирующей последовательность команд;
- во-вторых, FreeBSD, как и все UNIX-подобные операционные системы, широко используется во всем мире как основная серверная операционная система;
- в-третьих, благодаря схожести всех модификаций UNIX полученные результаты можно использовать для иных популярных UNIX-подобных операционных систем (Linux, OpenBSD, NetBSD, AIX и тому подобное).

Для каждого выполняемого в системе процесса ядро FreeBSD поддерживает определенную структуру данных при ведении аудита активности пользователей. По завершении процесса в режиме сбора статистики в файл аудита /var/acctount/acct добавляется новая запись. В этот файл записываются данные процессов, которые были начаты системным вызовом execve и закончили работу при обычных условиях. Чрезвычайными условиями завершения процесса считаются перегрузка системы или иные серьезные системные ошибки. Данные о таких процессах, а также о процессах, которые никогда не завершаются, также не отображаются в файле аудита. Анализ файла аудита выполнялся с помощью службы cron.

В качестве нейронной сети использовался многослойный персептрон с одним скрытым слоем. Размерность входного и выходного слоя сети составляла $9 \times m$ и 9 нейронов соответственно, поскольку в системе учитывались 512 возможных команд. В процессе проведения экспериментов прежде всего ставилась задача определения оптимальной ширины кадра (значения m) и размерности скрытого слоя нейронной сети. Оптимальной считалась такая ширина кадра, при которой средний процент

правильно спрогнозированных (угаданных) команд для всех пользователей за один выбранный день является максимальным. Сети для каждого пользователя обучались методом обратного распространения ошибки с коэффициентом скорости обучения $\alpha = 0.1$. Результаты эксперимента приведены на рис. 2.

Из рис. 2 видно, что максимальный процент правильно предсказанных команд достигается при $m = 4$ или $m = 5$, поэтому для дальнейших экспериментов было выбрано $m = 4$. Меньшее относительное число предсказанных команд при $m > 5$ $m < 4$ и логически объясняется тем, что при $m < 4$ в кадре не еще недостаточно информации для определения следующей команды, а при $m > 5$ в кадр попадают лишние команды, которые только усложняют обучение сети. Оба факторы приводят к снижению среднего процента правильно предсказанных команд.

В процессе экспериментов оказалось, что оптимальная размерность скрытого слоя составляет $Nh = 15$ (рис. 3). При этом значении уже наступает насыщение и дальнейшее увеличение Nh не ведет к увеличению количества угаданных команд.

Для определения способности нейронной сети отличить нормальное поведение от аномального был проведен такой основной эксперимент: в течение месяца велось наблюдение за активностью авторизованных пользователей, потом робота сети одного пользователя исследовалась на данных другого. Таким образом, на реальных данных была смоделирована распространенная ситуация, когда пароль пользователя был украден или утерян и под его именем в системе работал некто другой. При этом в первые 25 дней работал законный пользователь, в следующие 33 — другой человек. Типичные результаты отображены на рис. 4 и 5:

Как видно из рис. 4 и 5, нейронная сеть способна отличить аномальные данные от нормальных: если робота авторизованного пользователя характеризуется достаточно высоким процентом правильно предсказанных команд, то для "злоумышленника" этот показатель значительно ниже. Поскольку значение процента правильно предсказанных команд резко падает (часто к нулю) для сеансов, проведенных другим пользователем, то можно ввести некоторый критерий аномальности сеанса (например, резкое уменьшение относительного количества правильно предсказанных команд за определенный период времени).

Заключение. В работе предложен новый подход к выявлению аномальной деятельности пользователей в системе, основанный на применении нейронных сетей обратного распространения ошибки для идентификации характера поведения пользователя в системе по последовательности выполняемых им команд. Эффективность такого подхода в системах выявления аномалий показана на примере подмены пользователя и выявления такой подмены нейронной сетью. Результаты экспериментов, проведенных в реальной компьютерной системе с большим числом пользователей свидетельствуют о правильности избранного подхода.

Начатые исследования планируется продолжить в направлении изучения влияния различных параметров сеанса работы пользователя, способов представления (кодирования) команд, определения гибкого критерия аномальности сеанса для снижения уровня ошибок системы выявления аномалий, определения оптимального значения коэффициента обучения. Особое внимание необходимо уделить проблеме построения шаблонов поведения для пользователей с низкой активностью.

Литература

1. *Denning D.* An Intrusion-Detection Model// IEEE Transactions on Software Engineering, Vol. SE-13, 1987, No. 2, P.276-283.

2. *Sebring M., Shellhouse E., Hanna M.*, Whitehurst R. Expert Systems in Intrusion Detection: A Case Study//Proc. of the 11th National Computer Security Conference (Washington, USA), 1988, P. 315-321.

3. *Tan K.* The Application Of Neural Networks To UNIX Computer Security// Proc. of the IEEE International Conference on Neural Networks (Perth, Western Australia, Australia). — 1995. -Vol. 1. — P. 476-481.

4. *Debar H., Dorizzi B.* An Application of a Recurrent Network to an Intrusion Detection System// Proc. of 6-th International Joint Conference on Neural Networks (Baltimore, Maryland, USA). — 1992.— V. II.— P. 478-483.

5. *Ryan J., Lin M.-J., Miikkulainen R.* Intrusion Detection with Neural Networks// Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, 1998, P. 254-272.

6. *Neural* Networks for System Security/ Lozano C.M., Lopez F., Lopez J., and others//Proc. Of 5th European Congress on Intelligent Techniques and Soft Computing (Aachen, Germany). — 1997.— V.1.— P. 410-414.

7. *Obaidat M.S., Macchairolo D.T.* A multilayer neural network system for computer access security// IEEE Trans. on Syst., Man. and Cybern.— 1994. V.24.— N 5.— P. 806-813.

8. *Tan K.M., Collie B.S.* (1997). Detection and Classification of TCP/IP Network Services// Proc. of the Computer Security Applications Conference (San Diego, USA). P. 99-107.

9. *Reed R.D., Marks R.J.* II Neural Smithing. Supervised Learning in Feedforward Artificial Neural Networks. — Cambridge, Massachusetts, London: A Bradford Book. — 1999. — 346 p.

10. *Eberhart R., Simpson P., Dobbins R.* Computational Intelligence PC Tools: Academic Press, Inc., 1996.-464p.