**Вывод**

Системы блокирования поведения имеют громадные возможности в роли дополнительного уровня защиты от самых последних и неизвестных угроз. Но до сих пор массового распространения такие программы не получили, а существующие системы блокирования поведения производят впечатление несколько недоработанных и неудобных в работе продуктов. Это связано в первую очередь со следующими причинами: большое количество ложных тревог, трудности в настройке таких систем, уменьшения производительности работы операционной системы, некоторые производители предлагают решения, которые блокируют малораспространенные угрозы, но не в состоянии блокировать реально существующие проблемы. Поэтому работы в этом направлении сейчас являются очень перспективными. И мы надеемся, что наша система блокирования поведения внесет существенный вклад в развитие этой новой, очень перспективной отрасли.

Работа поддержана грантом РФФИ № 03-07-90075.

**М. В. Аникеев И. Д. Сидоров**
Россия, г. Таганрог, ТРТУ

## ОБНАРУЖЕНИЕ АНОМАЛЬНОГО ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЯ В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS НА ОСНОВЕ АНАЛИЗА РАБОТЫ С ПРИЛОЖЕНИЯМИ

Традиционно системы обнаружения вторжений (атак) подразделяются на системы обнаружения злоупотреблений и системы обнаружения аномалий [1]. К злоупотреблениям относятся известные атаки, которые используют известные уязвимости системы. Аномалии означают любую необычную деятельность, которая потенциально может указывать на атаку. Если наблюдаемая деятельность пользователя не соответствует ожидаемому режиму работы, то говорят, что имеет место аномалия. Основной проблемой при проектировании систем обнаружения аномалий является определение, какое поведение пользователя можно считать обычным, а какое содержит признаки аномалий. Необычное, но санкционированное использование иногда может рассматриваться как аномальное, что ведёт к росту числа ложных срабатываний. С другой стороны, если проектировать систему, основываясь на известных примерах аномального поведения, система будет неспособна обнаруживать нестандартные атаки. Одним из распространённых подходов к решению этой проблемы является создание профилей стандартного поведения индивидуальных пользователей в конкретной системе вместо создания единого профиля санкционированного поведения многих пользователей. При этом поведение характерное для одного из пользователей может считаться необычным для другого и наоборот. Поскольку такие профили трудно формализовать, предлагается их создавать на основе примеров нормальной работы того или иного пользователя. В качестве средства представления профилей были выбраны нейронные сети, благодаря их способности обучаться на примерах. В ряде работ по обнаружению аномалий [2, 3, 4] в качестве информативных признаков режима работы пользователя были выбраны характеристики использования различных команд оболочки операционной системы. Поскольку для работы в операционных системах семейства Windows использование командной оболочки крайне нетипично, такой подход становится неприемлемым. В Windows аналогом работы с командной оболочкой можно считать запуск и завершение приложений, а также переход от одного активного приложения к другому. В данной статье рассматривается система обнару-

жения аномального поведения, основанная на специфике работы индивидуальных пользователей с приложениями операционной системы Windows 2000.

Вектор, характеризующий активность пользователя, формируется на основе работы перехватчика активности приложений следующим образом:

Выбирается интервал времени, за который будет подсчитываться вектор.

За время сеанса работы пользователя с операционной системой составляется протокол запуска, завершения и активизации приложений, который разбивается на участки выбранной длины, причём учитывается только время активной работы – когда запущено ненулевое число процессов и интервал между двумя событиями не превышает некоторого критического времени – порога бездействия.

На этих участках подсчитывается статистика работы с процессами. Для каждого процесса вычисляется, сколько времени он был запущен и сколько времени он был активен. Время работы за интервал подсчитывается по формуле $\sum_{i=1}^{n}(d_i - a_i)$, где n – число запусков/завершений процесса, $d_i$ – время i-того завершения, $a_i$ – время i-того запуска. Время активности вычисляется по формуле $\sum_{i=1}^{n}(z_i - x_i)$, где $x_i$ – время, когда произошла активация искомого процесса, $z_i$ – время, когда произошла активация другого процесса.

Статистика нормируется, то есть значения времени делятся на общую длительность интервала. В результате значения оказываются в интервале [0;1].

Если всего в протоколе фигурируют n программ, в итоге мы получаем вектор действительных чисел размерности 2*n.

Для формирования обучающей выборки были выбраны 5 пользователей (Artem, Igor, Juic, Max, Stas). Перехватчик активности процессов контролировал их активность в течение 3-х дней. В итоге была подсчитана статистика со следующими параметрами:

Интервал активности пользователя – 3600 секунд (1 час)

Порог бездействия пользователя – 3600 секунд

Число процессов – 85, формировалось по итогам всех протоколов работы.

Число векторов, полученных в результате, представлено в табл. 1.

Таблица 1

| Пользователь | Число векторов |
|---|---|
| Artem | 10 |
| Igor | 10 |
| Juic | 19 |
| Max | 16 |
| Stas | 17 |
| Итого | 72 |

Для каждого пользователя была сформирована обучающая выборка и каждому пользователю была поставлена в соответствие нейронная сеть. Каждая нейронная сеть была обучена выдавать на выходе вектор <1;0> при предъявлении векторов с признаками характерной работы соответствующего пользователя, и вектор <0;1> при предъявлении «чужих» векторов. Максимально допустимое значение ошибки – 0.001. Обучение проводилось методом гибкого обратного распространения (resilient backpropagation) [5], нейросети затратили на обучение от 8 до 18 итераций (эпох). Тестовая выборка была сформирована из протоколов работы тех же пользователей в течение следующего дня. Число векторов в тестовой выборке приведено в табл. 2.

Классификация идёт следующим образом:

(свой > 0.7) И (чужой < 0.3) = свой

(свой < 0.3) И (чужой > 0.7) = чужой

(НЕ свой) И (НЕ чужой) =

((свой < 0.7) ИЛИ (чужой > 0.3)) И ((свой > 0.3) ИЛИ (чужой < 0.7)) = не классифицирован

Таблица 2

| Пользователь | Число векторов |
|---|---|
| Artem | 2 |
| Igor | 8 |
| Juic | 12 |
| Max | 11 |
| Stas | 5 |
| Итого | 38 |

Результаты тестирования приведены в табл. 3.

Таблица 3

| Тест | Предъявлено | Правильно классиф. | Неверно классиф. | Не классиф. |
|---|---|---|---|---|
| Всего | 190 | 177 | 5 | 8 |
| Свой пользователь | 38 | 34 | 2 | 2 |
| Чужой пользователь | 152 | 143 | 3 | 6 |

По данным табл. 3 можно оценить вероятность ошибки первого рода (ложная тревога) диапазоном 6-10%, вероятность ошибки второго рода диапазоном 2-6%.

Таким образом, предложенный метод позволяет достаточно эффективно сигнализировать о нестандартном поведении пользователей при работе с компьютерами на базе Windows. Такое нестандартное поведение нельзя однозначно ассоциировать с вторжением. Однако при проявлении других признаков атаки, данные такой системы могут ускорить поиск её источника или получить о ней дополнительную информацию.

Библиографический список

1. *Denning D. E.* An intrusion-detection model // Proc. IEEE Symposium on Security and Privacy. 1986. PP. 118-131.

2..*Lane T., Brodley C. E.* An application of machine learning to anomaly detection // Proc. 20th NIST-NCSC National Information Systems Security Conference. 1997. PP. 366-380.

3. *Ryan J., Lin M.-J., Miikkulainen R.* Intrusion Detection with Neural Networks // Advances in Neural Information Processing Systems. The MIT Press. Vol.10. 1998.

4. *Marin J. A., Ragsdale D., Surdu J. A* Hybrid Approach to Profile Creation and Intrusion Detection // Proc. of DARPA Information Survivability Conference and Exposition. – 2001.

5. *Riedmiller M., Braun H. A* direct adaptive method for faster backpropagation learning: The RPROP algorithm // Proc. of the IEEE International Conference on Neural Networks. 1993. PP. 586-591.

**Е.С. Абрамов, М. В. Аникеев, О. Б. Макаревич**
Россия, г. Таганрог, ТРТУ

## ПОДГОТОВКА ДАННЫХ ДЛЯ ИСПОЛЬЗОВАНИЯ В ОБУЧЕНИИ И ТЕСТИРОВАНИИ НЕЙРОСЕТЕЙ ПРИ ОБНАРУЖЕНИИ СЕТЕВЫХ АТАК

Большинство современных подходов к проблеме обнаружения вторжений [1] подразумевают использование анализа, основанного на жестких правилах. Такой анализ основан на наборах правил, которые либо заранее встраиваются в систему разработчиками, либо создаются самой системой или системным администратором в процессе эксплуатации. Наиболее традиционной формой систем обнаружения атак (СОА), основанных на правилах, считаются экспертные системы [6]. Экспертные системы успешно используют внесённые в них знания специалистов в области защиты информации для выявления сетевой активности с признаками