$$v_{3\min} =]\log_2 \wp[+]\log_2 p_{i\min}[+]\log_2 b_{i\ \min}[\text{ (бит)};$$

$$v_{3\max} =]\log_2 \wp[+]\log_2 p_{i\ \max}[+]\log_2 b_{i\ \max}[\text{ (бит)}.$$

Из приведенных аналитических выражений и рис. 3 видно, что предложенный метод разделения сообщения на части обеспечивает в 1,5 раза меньшую избыточность при аналогичных параметрах возобновления данных.

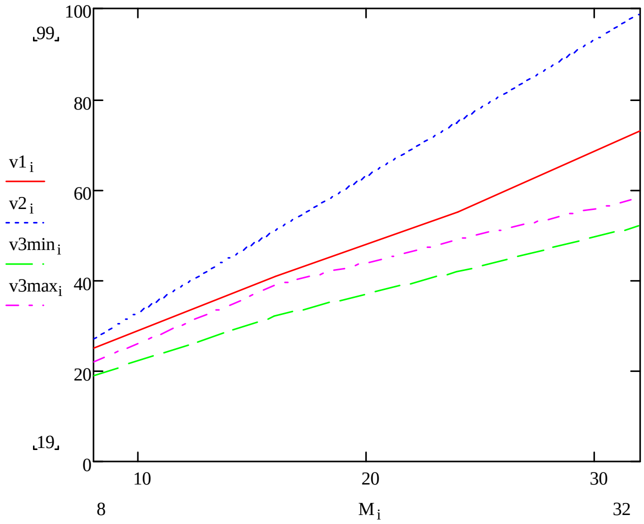

Рис. 3. Зависимость объема данных v от разрядности пакета данных M для разных пороговых схем разделения: $v1$ – схема Шамира; $v2$ – схема Асмута-Блума; $v3$ – разделение на основе системы остаточных классов

Экспериментально установлено, что уменьшить объемы данных при использовании пороговых схем для разделения сообщения на части можно следующим образом: передавая лишь пакеты с переменными коэффициентами, при этом постоянные составляющие, необходимые для возобновления данных, передаются отдельным пакетом (рис. 4).

Заключение. Предложен метод разделения сообщения на основе системы остаточных классов, которая характеризуется меньшей избыточностью в сравнении с пороговыми схемами разделения секрета:

- в 1,5 раза – при передаче служебных данных с каждой частью пакета;
- в 5 раз – при передаче служебных данных отдельным пакетом.

Еще одним преимуществом предложенного метода является то, что в результате разделения сообщения на части формируются подпакеты (остатки) разной разрядности – это дает возможность распределять их в зависимости от интегральной оценки качества маршрута.

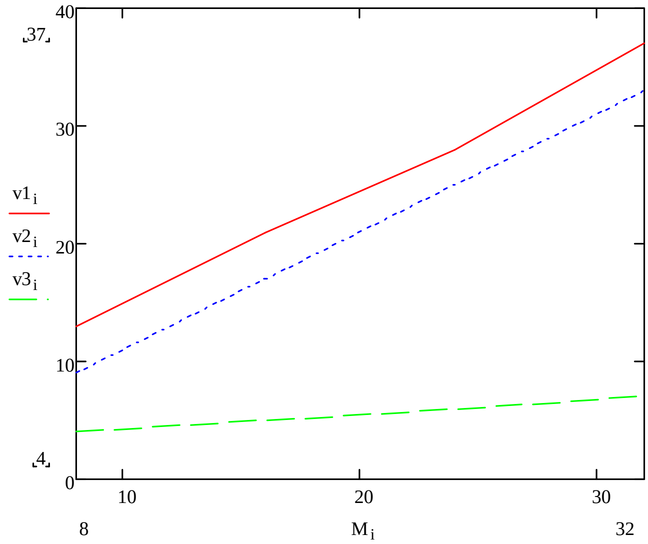

Рис. 4. Зависимость объема данных v от разрядности пакета данных M без учета постоянных составляющих: $v1$ – схема Шамира; $v2$ – схема Асмута-Блума; $v3$ – разделение на основе системы остаточных классов

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Lou, W. An efficient N-to-1 mutlipath routing protocol in wireless sensor networks: Proceedings of IEEE international Conference on Mobile Ad-hoc and Sensor Systems (MASS). – Washington, DC, November 2005.
2. Жуков, И.А. Способы повышения надежности и безопасности сбора информации в системах управления реального времени / И.А. Жуков, В.И. Дрововозов // Проблемы информатизации и управления.– 2008. – №1(23). – С. 262–276.
3. Sachenko, A. Modified Method of Noise-Immune Data Transmission in Wireless Sensors Networks / A. Sachenko, V. Yatskiv, R. Krepych // International Conference on Networks Security, Wireless Communications and Trusted Computing, “NSWCTC 2009”, 25–26 April 2009. – Wuhan, Hubei, China. – Volume 2. – P. 847–850.
4. Lou, W. SPREAD: Enhancing data confidentiality in mobile ad hoc networks, IEEE INFOCOM 2004 / W. Lou, W. Liu, Y. Fang. – HongKong, China, March 2004.
5. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.
6. Червяков, Н.И. Модулярные параллельные вычислительные структуры нейро-процессорных систем / Под. ред. Н.И. Червякова [и др.] – М.: ФИЗМАТЛИТ, 2003. – 288 с.
7. Телль, Ж. Введение в распределенные алгоритмы / Пер. с англ. В.А. Захарова. – М.: МЦНМО, 2009. – 616 с.

29.11.10

SU CZUN, YACKIV V.V., SACHENKO A.O. Increase of efficiency of transfer given in wireless touch networks on the basis of multitravelling routing

The method of data coding and transmission is proposed in the wireless networks on the basis of multipath routing and Residue Number System transformation, which is allowed to reduce the redundancy of data burst sharing and improved the reliability of data transmission.

004.056.57:032.26

Введение. В настоящее время проблема борьбы с компьютерной преступностью стала одной из первостепенных. Число компьютерных преступлений увеличивается ежегодно на 30–40 процентов [1]. Развитие глобальной сети Интернет способствует развитию ком-

***Безобразов Сергей Валерьевич**, к.т.н., доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*
Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

пьютерных преступлений. С каждым годом киберпреступления охватывают все новые и новые сферы, связанные с компьютерной информацией: вредоносные программы – вирусы, кража конфиденциальной информации, взлом информационных ресурсов и т.д.

Ежегодно появляется большое количество компьютерных вирусов, причем их количество постоянно увеличивается. Ущерб, наносимый вредоносными программами, составляет, по некоторым подсчетам, миллиарды долларов в год [2].

Как показала практика, традиционный подход в области обнаружения вредоносных программ, основанный на сигнатурном анализе [3, 4], не приемлем для обнаружения неизвестных компьютерных вирусов.

Основным недостатком сигнатурного анализа является необходимость в постоянном своевременном обновлении антивирусных баз, в которых хранятся сигнатуры известных вирусов. Вторым недостатком является задержка в ответной реакции антивирусной индустрии на появление нового вируса, так как для успешного обнаружения нового компьютерного вируса методом сигнатурного анализа необходимо вначале выделить уникальную сигнатуру вируса и добавить ее в антивирусные базы. Задержка ответной реакции со стороны различных антивирусных компаний на появление нового компьютерного вируса может варьироваться от нескольких часов до нескольких дней. За это время современные компьютерные вирусы способны нанести непоправимый ущерб информации и пользователям компьютеров по всему миру.

Не спасают ситуацию и разнообразные эвристические алгоритмы [5, 6, 7], разработанные для обнаружения неизвестных компьютерных вирусов. В настоящее время качество их функционирования оставляет желать лучшего. Часто такие алгоритмы классифицируют безобидные программы как вредоносные, и наоборот, пропускают вредоносные программы, считая, что это чистые файлы.

Современные исследования в области защиты информации направлены на создание таких методов и алгоритмов защиты, которые были бы способны обнаруживать и нейтрализовать неизвестные компьютерные вирусы и таким образом не только повысить уровень компьютерной безопасности, но и избавить пользователя от постоянных обновлений антивирусного ПО или его модулей.

Поэтому актуальной задачей является разработка эффективных алгоритмов построения системы обнаружения вредоносных программ, которые позволили бы обнаруживать неизвестные компьютерные вредоносные программы.

В прошлой нашей статье [8] мы рассказали о принципах построения искусственной иммунной системы для обнаружения вредоносных программ, где в качестве детекторов вирусов выступают искусственные нейронные сети. Представленная система базируется на интеграции таких методов искусственного интеллекта, как искусственные иммунные системы [9, 10], нейросетевые технологии [11, 12, 13] и эволюционное программирование.

В данной статье мы продолжаем рассказывать об интеллектуальной системе обнаружения вредоносных программ и представляем основные принципы функционирования разработанной системы.

1. Алгоритм функционирования нейросетевого иммунного детектора. На рисунке 1 представлен основной элемент обнаружения вредоносных программ – нейросетевой иммунный детектор. Он состоит из многослойной нейронной сети и арбитра.

Как уже отмечалось [8], нейронные элементы слоя Кохонена функционируют по принципу «победитель берет все» [11, 12]. Это означает, что выходное значение нейрона-победителя равняется «1», а выходные значения остальных нейронных элементов равняются «0».

Для определения нейрона-победителя используется Евклидово расстояние между входным и весовыми векторами. Так, Евклидово расстояние между входным и весовым вектором i-го нейронного элемента определяется следующим образом:

$$D_i = \left| X - \omega_i \right| = \sqrt{(X_1 - \omega_{1i})^2 + (X_2 - \omega_{2i})^2 + \ldots + (X_c - \omega_{ci})^2},$$

где ω_{ci} – весовой коэффициент между [illegible]-м нейроном распределительного слоя и i-м нейроном слоя Кохонена,

[illegible]$=[X_1, X_2 \ldots X_n]$ – входной образ.

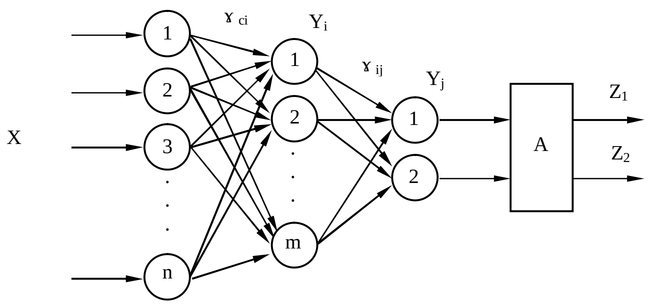

Рис. 1. Нейросетевой иммунный детектор

Нейронный элемент-победитель с номером k определяется в соответствии с минимальным Евклидовым расстоянием:

$$D_k = \min_j D_j. \tag{2}$$

Тогда выходная активность нейронов слоя Кохонена определяется

$$Y_i = \begin{cases} 1, & \text{[illegible]}\ i = k \\ 0, & \text{[illegible]} \end{cases} \tag{3}$$

Выходное значение j-го нейрона третьего слоя определяется согласно формуле:

$$Y_j = \omega_{kj} \cdot Y_k. \tag{4}$$

Арбитр принимает окончательное решение о том, является ли сканируемый файл вредоносным. Для этого он вычисляет количество чистых и вредоносных фрагментов сканируемого файла

$$\overline{Y_1} = \sum_{k=1}^{L} Y_1^k, \tag{5}$$

$$\overline{Y_2} = L - \overline{Y_1} = \sum_{k=1}^{L} Y_2^k, \tag{6}$$

где L – множество образов сканируемого файла,

Y_i^k – выходное значение i-го нейрона линейного слоя при подаче на вход сети k-го образа.

Далее определяются вероятности принадлежности сканируемого файла соответственно к чистому и вредоносному классу

$$P_T = \frac{\overline{Y_1}}{L} \cdot 100\%, \tag{7}$$

$$P_F = 1 - P_T = \frac{\overline{Y_2}}{L} \cdot 100\%. \tag{8}$$

Окончательное решение о принадлежности файла к чистому классу арбитр принимает следующим образом:

$$Z_1 = \begin{cases} 1, & \text{[illegible]}\ P_T > 80\% \\ 0, & \text{[illegible]}. \end{cases} \tag{9}$$

$$Z_2 = \begin{cases} 1, & \text{[illegible]}\ P_F > 20\% \\ 0, & \text{[illegible]}. \end{cases} \tag{10}$$

Таким образом, пространство выходных значений арбитра можно представить в табличном виде (таблица 1).

Таблица 1. Пространство выходных значений арбитра

Z_1	Z_2	класс
1	0	Чистый
0	1	Вирус
0	0	Не определено

Если выходные значения арбитра имеют нулевые значения, то сканируемый файл отправляется на дополнительную проверку другому нейросетевому иммунному детектору.

В процессе сканирования проверяемого файла на нейросетевой детектор последовательно подаются фрагменты файла по методу скользящего окна.

Алгоритм функционирования нейросетевого иммунного детектора в режиме сканирования файла можно свести к следующей последовательности шагов:

1. Устанавливаются следующие начальные значения:

$$\overline{Y_1}(k-1)=0,\qquad \overline{Y_2}(k-1)=0. \tag{11}$$

2. По методу скользящего окна последовательно подаются входные образы ($k=\overline{1,L}$) из сканируемого файла на нейронную сеть и для каждого входного образа производятся следующие вычисления:

a) определяется Евклидово расстояние между входным образом и весовыми векторами нейронов слоя Кохонена:

$$D_i=\left|X-\omega_i\right|=\sqrt{(X_1-\omega_{1i})^2+(X_2-\omega_{2i})^2+(X_n-\omega_{ni})^2}, \tag{12}$$

где $i=\overline{1,m}$;

b) определяется нейронный элемент-победитель с номером k (формула 2);

c) вычисляются выходные значения линейных нейронных элементов третьего слоя (формула 4);

d) определяется количество чистых и вредоносных фрагментов сканируемого файла:

$$\overline{Y_1}(k)=\overline{Y_1}(k-1)+Y_1^k, \tag{13}$$

$$\overline{Y_2}(k)=\overline{Y_2}(k-1)+Y_2^k. \tag{14}$$

3. Вычисляются вероятности принадлежности сканируемого файла соответственно к чистому и вредоносному классу (формулы 7 и 8 соответственно).

4. На основании вычислений вероятностей принимается решение о принадлежности сканируемого файла к одному из классов, в соответствии с выражениями 9 и 10.

5. Если $Z_1=0$ и $Z_2=0$, то назначается другой нейросетевой иммунный детектор для повторной проверки файла.

Рассмотрим функционирование обученного нейросетевого иммунного детектора на примере сканирования двух файлов: *diskcopy.exe* и *Virus.Win32.Neshta.a*. Для его обучения использовались данные из четырех чистых и одного вредоносного файлов: *chcp.com, loadfix.com, print.exe, regedit32.exe и Virus.Win32.Virut.a.* Количество фрагментов, выбираемых из каждого файла $A=5$, т.е. размер обучающей выборки – 25 образов. При обучении установим размер входного образа 128 символов. Тогда, количество нейронов распределительного слоя равняется 128, количество нейронов слоя Кохонена примем равным 10, причем 8 первых нейронов используются для чистых входных образов, а 2 последних используются для вредоносных входных образов, т.е. p=8, r=2. Количество нейронов третьего слоя равняется 2 (рис. 1). Такое соотношение нейронов слоя Кохонена обусловлено формированием обучающей выборки, когда 80% всей обучающей выборки составляют образы чистого класса, а 20% выборки составляют образы вредоносного класса.

Методом скользящего окна, размерность которого равняется 128, последовательно подаем данные из сканируемого файла *diskcopy.exe* на вход обученного нейросетевого иммунного детектора. Данные из файла на вход нейронной сети подаются в виде кода ASCII, т.е. принимают значения в диапазоне от 0 до 255. Размер файла diskcopy.exe составляет 7168 байт. Так как размер скользящего окна равняется 128, то в результате будет сформировано следующее количество окон:

$$L=S-n+1=7168-128+1=7041, \tag{15}$$

где S – размер файла,

n – размер окна.

В результате проверки нейросетевой иммунный детектор соотнес 6161 фрагментов к классу чистых образов и 880 фрагментов – к классу вредоносных образов:

$$P_T=\frac{\overline{Y_1}}{L}\cdot 100\%=\frac{6161}{7041}\cdot 100\%=87{,}5\%, \tag{16}$$

$$P_F=1-P_T=\frac{\overline{Y_2}}{L}\cdot 100\%=\frac{880}{7041}\cdot 100\%=12{,}5\%. \tag{17}$$

Далее арбитр принимает решение о принадлежности сканируемого файла *diskcopy.exe* к тому или иному классу:

$$Z_1=1,\quad \text{если}\quad P_T>80\%,\qquad Z_2=0,\quad \text{если}\quad P_F<20\%. \tag{18}$$

Таким образом, сканируемый файл является чистым файлом.

Аналогичным образом подаем данные из сканируемого файла *Virus.Win32.Neshta.a* на вход обученного нейросетевого иммунного детектора. Размер файла *Virus.Win32.Neshta.a* составляет 32687 байт. Так как размер скользящего окна равняется 128, то в результате будет сформировано следующее количество окон:

$$L=S-n+1=32687-128+1=32560. \tag{19}$$

В результате проверки нейросетевой иммунный детектор соотнес 22466 фрагментов к классу чистых образов и 10094 фрагментов – к классу вредоносных образов:

$$P_T=\frac{\overline{Y_1}}{L}\cdot 100\%=\frac{22466}{32560}\cdot 100\%=69{,}0\% \tag{20}$$

$$P_F=1-P_T=\frac{\overline{Y_2}}{L}\cdot 100\%=\frac{10094}{32560}\cdot 100\%=31{,}0\%. \tag{21}$$

Далее арбитром принимается решение о принадлежности сканируемого файла к классу вредоносных программ, поскольку $P_F>20\%$.

Следует отметить, что время, необходимое для обучения одного детектора, составляет в среднем 1–3 секунды.

В данном разделе представлен алгоритм функционирования нейросетевого иммунного детектора. Он позволяет обнаруживать вредоносные программы, которые не входили в обучающую выборку, в то же время, оставаясь «равнодушным» к чистым файлам, не имеющим вредоносные функции.

Таблица 2. Результаты сканирования неинфицированных файлов

Имя файла	Детектор 1 P_T / P_F	Детектор 2 P_T / P_F	Детектор 3 P_T / P_F	Детектор 4 P_T / P_F
cacls.exe	0,78 / **0,22**	**0,93** / 0,07	**0,89** / 0,11	**0,82** / 0,18
ctfmon.exe	**0,81** / 0,19	**0,86** / 0,14	**0,87** / 0,13	**0,89** / 0,11
dbexplor.exe	**0,90** / 0,10	**0,93** / 0,07	**0,94** / 0,06	**0,90** / 0,10
dcomcnfg.exe	**0,91** / 0,09	**0,96** / 0,04	**0,96** / 0,04	**0,96** / 0,04
diskcopy.com	**0,83** / 0,17	**0,93** / 0,07	**0,92** / 0,08	**0,83** / 0,17
dllhost.exe	**0,89** / 0,11	**0,96** / 0,04	**0,98** / 0,02	**0,85** / 0,15
etm70.exe	**0,91** / 0,09	**0,94** / 0,06	**0,95** / 0,05	**0,87** / 0,13
notepad.exe	**0,84 /** 0,16	**0,91 /** 0,09	**0,92** / 0,08	**0,83 /** 0,17
soundman.exe	**0,87** / 0,13	**0,93** / 0,07	**0,94** / 0,06	**0,93** / 0,07
taskman.exe	**0,88** / 0,12	**0,92** / 0,08	**0,95** / 0,05	**0,92** / 0,08
uninlib.exe	0,58 / **0,43**	**0,81** / 0,19	**0,83** / 0,17	**0,82** / 0,18

Таблица 3. Результаты сканирования вредоносных программ

Имя файла	Детектор 1 P_T / P_F	Детектор 2 P_T / P_F	Детектор3 P_T / P_F	Детектор 4 P_T / P_F
Backdoor.Agent	**0,98** / 0,02	**0,98** / 0,02	**0,98** / 0,02	**0,96** / 0,04
Backdoor.Agobot	**0,91** / 0,09	0,58 / **0,42**	0,68 / **0,32**	**0,83** / 0,17
E-Worm.Bozori	0,64 / **0,36**	0,73 / **0,27**	0,55 / **0,45**	**0,85** / 0,15
E-Worm.Zafi	0,70 / **0,30**	0,58 / **0,42**	0,68 / **0,32**	**0,87** / 0,13
E-Worm.Mydoom	0,67 / **0,13**	0,65 / **0,35**	0,65 / **0,35**	0,79 / **0,21**
E-Worm.NetSky	0,61 / **0,39**	0,68 / **0,32**	0,57 / **0,43**	0,80 / 0,20
Exploit.DebPloit	**0,85** / 0,15	**0,92** / 0,08	**0,92** / 0,08	**0,92** / 0,08
Net-Wor.Lovesan	**0,83** / 0,17	**0,81** / 0,19	0,77 / **0,23**	0,71 / **0,29**
Net-Worm.Mytob	**0,84 /** 0,16	0,55 **/ 0,45**	0,63 / **0,37**	0,74 **/ 0,26**
Trojan.Bagle	**0,81** / 0,19	**0,85** / 0,15	0,78 / **0,22**	0,68 / **0,32**
Trojan.Daemonize	**0,93** / 0,07	**0,84** / 0,16	**0,84** / 0,16	**0,84** / 0,16
Trojan.LdPinch	**0,89** / 0,11	0,60 / **0,40**	0,76 / **0,24**	**0,81** / 0,19
Virus.Gpcode	0,73 / **0,27**	0,54 / **0,46**	0,64 / **0,36**	0,58 / **0,42**
Virus.Hidrag	0,79 / **0,21**	0,76 / **0,24**	0,75 / **0,25**	0,77 / **0,23**

Таблица 4. Сравнительный анализ различных антивирусных продуктов

Имя файла	Антивирус Касперского (актуал. базы)	Антивирус Касперского (устар. базы)	NOD32 (эвристическ. анализатор)	ИИС (на основе 4-х детекторов)
Backdoor.Win32.Agent.lw	Backdoor	OK	OK	OK
Backdoor.Win32.Agobot	Backdoor	Backdoor	Win32/Agobot	Вирус
Email-Worm.BAT.Maddas	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.JS.Gigger	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.VBS.Loding	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.Win32.Zafi.d	Email-Worm	OK	NewHeur_PE	Вирус
Net-Worm.Win32.Bozori.a	Net-Worm	OK	Win32/Bozori	Вирус
Net-Worm.Win32.Mytob.a	Net-Worm	OK	Win32/Mytob	Вирус
Trojan-Downl.JS.Psyme.y	Trojan	OK	OK	Вирус
Trojan-Downl.Win32.Bagle	Trojan	OK	Win32/Bagle	Вирус
Trojan-Proxy.Daemonize	Trojan	Trojan	OK	OK
Trojan-Proxy.Mitglieder	Trojan	Trojan	Win32/Trojan	Вирус
Trojan-Proxy.Win32.Agent	Trojan	Trojan	OK	Вирус
Trojan-PSW.LdPinch	Trojan	Trojan	Win32/PSW	Вирус
Virus.Win32.Gpcode.ac	Virus.Win32	OK	OK	Вирус
Exploit.Win32.DebPloit	Exploit	OK	OK	OK

2. Тестирование нейросетевых иммунных детекторов для обнаружения вредоносных программ. Предложенная нейросетевая искусственная иммунная система тестировалась по следующей схеме. Вначале обученные иммунные детекторы проверялись на невосприимчивость к чистым файлам. Затем они проверялись на способность обнаруживать различные вирусы и их семейства. Результаты сканирования неинфицированных файлов представлены в таблице 2. Таблица 2 демонстрирует результаты сканирования вредоносных программ. В таблицах P_T и P_F отображают вероятность того, что проверяемый файл является чистым или вредоносным соответственно.

Следует отметить, что время обучения нейросетевого иммунного детектора, по сравнению со временем обучения генетического иммунного детектора, меньше в сотни раз и составляет в среднем 1–3 секунды.

В обучающей выборке для первого детектора использовался вирус *Email-Worm.Win32.Mydoom*. Как видно из таблицы 2, этот детектор достаточно хорошо обнаруживает черви и классические вирусы (*Gpcode, Hidrag*). Однако он классифицирует файлы *cacls.exe* и *uninlib.exe* как компьютерные вирусы. Это явный пример нежелательного детектора. Такой детектор должен уничтожаться на стадии отбора. Для обучения остальных трех детекторов использовались различные вредоносные программы, поэтому они показывают разные результаты.

Рассмотрим результаты обнаружения различных вредоносных программ.

Как видно из таблицы 3, некоторые компьютерные вирусы (*E-Worm.Mydoom, Virus.Gpcode*, *Virus.Hidrag*) обнаруживаются всеми детекторами. Но есть и такие (*Backdoor.Agent, Trojan.Daemonize, Exploit.DebPloit*), которые обнаруживаются плохо либо вообще не обнаруживаются. Частично это связано с тем, что некоторые из них не наносят абсолютно никакого вреда (*Backdoor.Agent*). Также следует отметить, что здесь представлены результаты работы только четырех детекторов. Повышение количества детекторов увеличивает вероятность обнаружения всех вредоносных программ. Анализируя полученные результаты можно сделать вывод, что один нейросетевой иммунный детектор способен обнаружить несколько компьютерных вирусов.

В следующем эксперименте представлен сравнительный анализ результатов обнаружения компьютерных вирусов различными антивирусными продуктами. Для теста были выбраны: Антивирус Касперского версии 5 с актуальными вирусными базами; Антивирус Касперского версии 5 с устаревшими вирусными базами; антивирусный продукт NOD 32 с отключенными вирусными базами, но с задействованным эвристическим анализатором, и, разработанная нами система обнаружения вредоносных программ, построенная на основе искусственных иммунных систем с применением методов искусственных нейронных сетей. Этот эксперимент разрабатывался с целью показать незащищенность компьютерной системы с устаревшими вирусными базами и несовершенство эвристических алгоритмов. Таблица 4 отображает результаты эксперимента. Здесь «*OK*» - означает решение антивирусной программы о том, что файл является чистым.

Как видно из полученных результатов, антивирусный продукт с актуальными вирусными базами обнаружил все вирусы, которые использовались в эксперименте. Это объясняется тем, что в базах содержались сигнатуры используемых в эксперименте вирусов. Антивирус с устаревшими вирусными базами обнаружил только поло-

вину присутствующих вирусов, что явно отражает угрозу компьютерной системы перед новыми компьютерными вирусами. Антивирус NOD 32, который использовал эвристический анализатор, обнаружил только семь вирусов, что является очень низким показателем для надежной современной системы безопасности. Искусственная иммунная система, которая использовала только четыре детектора, не обнаружила три вируса, однако с увеличением количества детекторов все присутствующие вирусы были обнаружены.

Один нейросетевой иммунный детектор способен обнаруживать несколько вредоносных программ. Причем детектор приобретает способность обнаруживать принципиально новые вредоносные программы.

Рассмотрим приближенную оценку вероятности обнаружения вредоносных программ нейросетевой искусственной иммунной системой. Пусть P_i – вероятность обнаружения вредоносного файла i-м детектором. При сканировании файлового пространства r независимыми детекторами вероятность не обнаружения вредоносной программы определяется следующим образом:

$$g(r) = \prod_{i=1}^{r}(1-P_i)\,. \qquad (22)$$

Отсюда можно получить следующее выражение для оценки вероятности обнаружения вредоносной программы нейросетевой иммунной системой:

$$P(r) = 1-\prod_{i=1}^{r}(1-P_i)\,. \qquad (23)$$

Как следует из последнего выражения, с увеличением количества детекторов увеличивается вероятность обнаружения вредоносной программы.

Рассмотрим приближенную оценку количества детекторов для заданной достоверности обнаружения вредоносной программы $P(r)$. Предположим, что вероятность обнаружения вредоносной программы каждым детектором приблизительно одна и та же и равняется P_0. Тогда

$$P(r) = 1-(1-P_0)^r\,. \qquad (24)$$

Следовательно, для заданной вероятности обнаружения вредоносных программ $P(r)$ можно получить приближенную оценку количества детекторов в иммунной системе:

$$r = \frac{\ln(1-P(r))}{\ln(1-P_0)}\,. \qquad (25)$$

Отсюда следует, что с увеличением популяции нейросетевых иммунных детекторов вероятность обнаружения вредоносных программ возрастает.

Рассмотрим теоретическую и экспериментальную оценки вероятности обнаружения вредоносной программы в зависимости от количества детекторов. Экспериментальная оценка определяется на основе следующего соотношения:

$$P(k) = \frac{n(k)}{n}\,, \qquad (26)$$

где $k=1,r$,

r – общее количество детекторов,

$n(k)$ – количество вредоносных программ, обнаруживаемых k-детекторами,

n – общее количество вредоносных программ.

Для получения приближенной теоретической оценки необходимо определить вероятность обнаружения детектором вредоносной программы. Предположим, что вероятность обнаружения вредоносной программы каждым детектором является приблизительно одинаковой и равняется P_0. Тогда, согласно выражению (24),

$$P_0 = 1-(1-P(r))^{1/r} \qquad (27)$$

Пусть $r = 24$, тогда $P(r) = 0{,}78$. Отсюда $P_0 = 0{,}0325$. Тогда теоретическая оценка вероятности обнаружения вредоносной программы в зависимости от количества детекторов определяется следующим образом:

$$P(k) = 1-(1-P_0)^k\,. \qquad (28)$$

На рис. 2 приведены результаты экспериментальной и теоретической вероятности обнаружения вредоносной программы в зависимости от количества детекторов.

Как следует из рисунка, теоретическая вероятность хорошо аппроксимирует экспериментальную вероятность обнаружения вредоносных программ.

Отсюда следует, что применение нейросетевой искусственной иммунной системы позволяет корректно и точно обнаруживать вредоносные программы и не генерировать ложные срабатывания. Как продемонстрировали эксперименты, совокупность нейросетевых иммунных детекторов позволяет своевременно и надежно защитить компьютерную систему от заражения компьютерными вирусами.

Заключение

1. Предложен алгоритм функционирования нейросетевых иммунных детекторов, который характеризуется вероятностным принципом работы, а также тем, что окончательный результат клас-

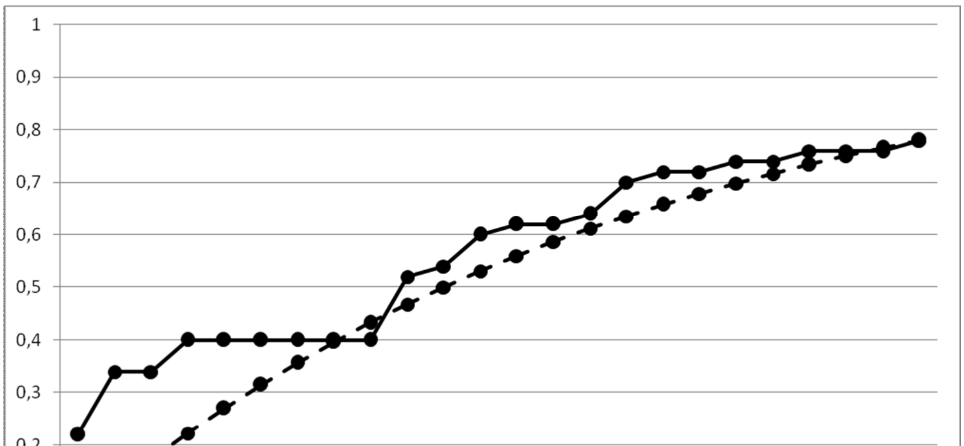

Рис. 2. Вероятности обнаружения вредоносных программ детекторами

сификации происходит после подачи всех образов сканируемого файла на нейронную сеть. Отличительной особенностью алгоритма является способность НИД обнаруживать неизвестные вредоносные программы.

2. Получено приближенное выражение для оценки вероятности обнаружения вредоносной программы искусственной иммунной системой. Показано, что с увеличением количества детекторов увеличивается вероятность обнаружения. Предложена приближенная оценка количества детекторов для заданной вероятности обнаружения вредоносной программы.
3. Проведены эксперименты по тестированию нейросетевой искусственной иммунной системы. Они показали способность нейросетевых иммунных детекторов обнаруживать разнотипные неизвестные вредоносные программы. В отличие от известных антивирусных программ нейросетевая искусственная иммунная система обнаруживает в среднем в 1,5 раза больше неизвестных вредоносных программ.
4. Приведены теоретическая и экспериментальная оценки вероятности обнаружения вредоносной программы в зависимости от количества детекторов.

Разработанная система может быть использована при построении как новых систем защиты компьютеров от вредоносных программ, так и в дополнении к уже имеющимся средствам.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Киберпреступность // Центр исследования компьютерной преступности [Электронный ресурс]. – 2007. – Режим доступа: http://www.crime-research.ru/news/05.09.2007/3793/. – Дата доступа: 27.11.2007.
2. Пресс-Центр // Антивирус ВирусБлокАда [Электронный ресурс]. – 2005. – Режим доступа: http://www.anti-virus.by/press/viruses/1485.html. – Дата доступа: 25.08.2007.
3. Касперский, Е. Компьютерное зловредство / Е. Касперский. – СПб.: Питер, 2007. – 208 с.
4. Касперский, К. Записки исследователя компьютерных вирусов / К. Касперский. – СПб.: Питер, 2006. – 316 с.
5. Куприянов, А.И. Основы защиты информации / А.И. Куприянов, А.В. Сахаров. – М.: Академия, 2006. – 256 с.
6. Зайцев, О.В. Rootkits, spyware/adware, keyloggers & backdoors: Обнаружение и защита / О.В. Зайцев. – СПб.: BHV-Санкт-Перербург, 2006. – 304 с.
7. Проактивность как средство борьбы с вирусами // Интернет-безопасность [Электронный ресурс]. – 1996. – Режим доступа: http://www.viruslist.com/ru/analysis?pubid=189544544. - Дата доступа: 15.05.2008.
8. Безобразов, С.В. Нейросетевая искусственная иммунная система для обнаружения вредоносных программ: принципы построения / С.В. Безобразов, В.А. Головко // Вестник БрГТУ. Физика, математика, информатика. – 2009.
9. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2005. – 1424 с.
10. Дасгупта, Д. Искусственные иммунные системы и их применение / Д. Дасгупта; под ред. Д. Дасгупта. – М.: Физматлит, 2006. – 344 с.
11. Головко, В.А. Нейронные сети: обучение, организация, применение / В.А. Головко // Нейрокомпьютеры и их применение : учеб. пособие / В.А. Головко. – М., 2001 – 256 с.
12. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.
13. Яхъяева, Г.Э. Нечеткие множества и нейронные сети / Г.Э. Яхъяева. – М.: Бином. ЛЗ, 2008. – 316 с.

[illegible] 11.11.10

BEZOBRAZOV S.V., GOLOVKO V.A. The Neuronet Immune System for Malware Detection: the Principles of Construction

In this paper we propose the principles of the neural network immune system functioning for detection of unknown malware. Research results are submitted.

[illegible] 004.5;621.38

[illegible]

[illegible]

Введение. Традиционные способы описания геометрических объектов и, в частности, многоугольников основаны на использовании методов вычислительной геометрии [1] и имеют практическое применение, например, при автоматизированном проектировании топологии интегральных схем [2, 3]. Однако в последнее время появились альтернативные способы описания многоугольников, основанные на использовании булевых формул [4, 5].

В настоящей работе предлагается способ решения одной из задач, лежащих в основе изложенного в работе [5] метода построения канонической булевой формулы. Тем самым указанный метод может быть легко доведен до формы алгоритма и, далее, переведен в форму программ на каком-либо языке программирования.

1. Основные определения, постановка задачи. Многоугольник, расположенный на плоскости, задается своей *границей* – замкнутой не пересекающейся ломаной линией, состоящей из отрезков прямых или *сторон* многоугольника. Эту границу можно определить последовательностью *угловых точек* или *вершин* многоугольника, получаемых при обходе его по границе справа: $p_1, p_2, \ldots, p_n$ (рис. 1).

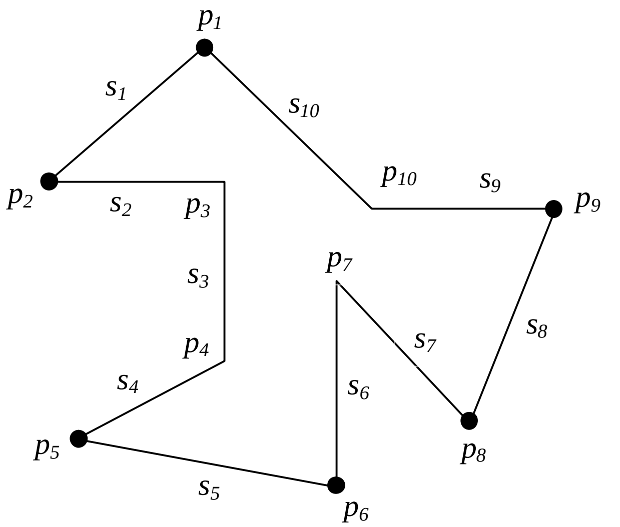

Рис. 1. Угловые точки и стороны многоугольника

***Бутов А.А.**, к.т.н., доцент кафедры экономической каибернетики Белорусского государственного университета информатики и радиоэлектроники.*
Беларусь, БГУИР, 220013, г. Минск, ул. П. Бровки, 6.