УДК 004.056.57:004.89

Н.М. Кораблёв, М.В. Кушнарёв

Харьковский национальный университет радиоэлектроники, Харьков

МОДЕЛЬ ЭВРИСТИЧЕСКОГО АНАЛИЗАТОРА ВРЕДОНОСНЫХ ПРОГРАММ НА ОСНОВЕ ИСКУССТВЕННОЙ ИММУННОЙ СЕТИ

Предложена модель эвристического анализатора вредоносных программ, основным компонентом которой является искусственная иммунная сеть, с помощью которой осуществляется обучение, детектирование и анализ как существующих, так и новых модификаций вирусов. Для решения рассматриваемой задачи определяются рейтинги встречаемости для каждого фрагмента программы, показывающие, в каком количестве объектов был найден данный фрагмент. Проведено моделирование работы эвристического анализатора, показывающее эффективность предложенной модели.

Ключевые слова: *эвристический анализатор, искусственная иммунная сеть, вредоносная программа, модель, аффинность, антитело, антиген.*

Введение

На сегодняшний день создано множество алгоритмов генерации вредоносных программ, что значительно усложняет их детектирование. Разработка автоматизированных систем генерации новых модификаций вредоносных программ позволяет создателям вирусов всегда опережать детектирование и выпуск антивирусных баз. Полностью эффективных методов борьбы с ними пока еще не существует, что дает большое поле для исследований в этом направлении.

Методы выявления и распознавания вредоносных программ делятся на два класса [1, 2]: 1) сигнатурная проверка, 2) эвристический анализ. Основным способом выявления большинства компонентов вредоносных программ все еще является сигнатурная проверка [3, 4], позволяющая обнаруживать и распознавать большинство вирусов. Сигнатура – это уникальный набор характеристик, характеризующих объект (вирус). Однако этот набор характеристик заведомо меньше полного количества характеристик, присущих объекту. В данном случае это вариант хеш-функции, т.е. неоднозначного отображения большого множества объектов на небольшое множество комбинаций их признаков.

Задача эвристического анализа – распознавание неизвестных модификаций вирусов [5 – 9]. На вход эвристического анализатора (ЭА) поступает программный файл, который исследуется с использованием различных эвристических методов (синтаксический и семантический анализы, логические и статистические методы, дискретно-событийные модели, интерпретация эмулятором и др.), после чего, основываясь на полученных результатах, принимается решение о том, является ли данный файл вредоносной программой или нет.

Однако, существующие эвристические технологии, призванные помочь в определении новых модификаций вирусов, на сегодня не дают должного уровня распознавания в связи с их слабой эффективностью при работе с зашифрованными объектами. К недостаткам существующих методов обнаружения вторжений, в первую очередь, можно отнести уязвимость к новым атакам, низкую точность и скорость работы. Современные системы обнаружения вторжений плохо приспособлены к работе в реальном времени, в то время как возможность обрабатывать большой объем данных в реальном времени – это определяющий фактор использования таких систем. Указанные недостатки трудно устранить, используя только классические методы в области компьютерной безопасности.

Поэтому появилась необходимость в разработке новых эвристических методов, которые бы смогли эффективно распознавать новые модификации вирусов. Одним из возможных подходов к решению данной проблемы является использование искусст-

венной иммунной сети (ИИС) [10, 11], как фундаментальной парадигмы для решения задач классификации и кластеризации.

В работе предлагается модель ЭА, основным компонентом которого является ИИС. Для решения проблемы работы с полиморфным кодом предлагается взять в качестве входных данных ЭА протоколы, полученные в результате интерпретации вредоносных программ с помощью эмулятора.

Постановка задачи. Пусть имеется множество $P = \{p_q\}$, $q = \overline{1,Q}$ исполняемых файлов (объектов), которые могут содержать вредоносные коды. На этом множестве имеется разбиение на конечное число подмножеств (классов) C_k, $k = \overline{1,K}$:

$$\bigcup_{k=1}^{K} C_k = P, \qquad (1)$$

и подробные карты их действий (протоколы). Разбиение определено не полностью – задан лишь некоторый набор обучающей информации $I_0(C_1, C_2, ..., C_K)$ о классах C_k, $k = \overline{1,K}$. Протоколы могут содержать общие закономерности в поведении объектов. Эти закономерности представляются в виде фрагментов протоколов, являющихся значениями некоторых признаков объектов x_j, $j = \overline{1,L}$, которые сохраняются в библиотеке (этот набор всегда один и тот же для всех объектов, рассматриваемых при решении задачи). Совокупность значений признаков x_j, $j = \overline{1,L}$ определяет описание объектов (программ) $I(C_1, C_2, ..., C_K)$. Информация о вхождении объекта p_q, $q = \overline{1,Q}$ в какой-либо класс представляется в виде информационного вектора $I(p_q) = \{I_1(p_q), I_2(p_q), ..., I_K(p_q)\}$, где $I_k(p_q)$ несет информацию о принадлежности или не принадлежности объекта (программы) p_q, $q = \overline{1,Q}$ к классу C_k, $k = \overline{1,K}$:

$$I_k(p_q) = \begin{cases} 1, p_q \in C_k, \\ 0, p_q \notin C_k. \end{cases} \qquad (2)$$

Задача детектирования и анализа вредоносных программ состоит в том, чтобы для исполняемых файлов p_q, $q = \overline{1,Q}$ и набора классов C_k, $k = \overline{1,K}$ по обучающей информации $I_0(C_1, C_2, ..., C_K)$ и описаниям $I(p_q)$ отнести исходные программы к определенному классу C_k с помощью выделения и сопоставления соответствующих признаков (фрагментов программ).

Для решения этой задачи необходимо определить рейтинг встречаемости для каждого найденного фрагмента программы, который покажет, в каком количестве объектов, из всего множества представленных, был найден данный фрагмент. При этом должна быть сформирована выборка с рейтингами признаков вредоносных программ.

Модель искусственной иммунной сети

Обучение, детектирование и анализ вредоносных программ предлагается осуществлять с помощью ИИС, для которой исполняемые файлы являются антигенами Ag, а возможные решения задачи – антителами Ab. Предполагается, что все иммунные процессы происходят в некотором пространстве $S^{N \times L}$, являющимся многомерным метрическим пространством, в котором конфигурация точки s характеризуется L параметрами. Тогда математически множество свойств, характеризующих антитело или антиген, можно представить в виде L-мерного вектора. Следовательно, точка s в L-мерном пространстве однозначно определяет множество свойств, необходимых для характеристики взаимодействий антиген-антитело $(Ag - Ab)$ и антитело-антитело $(Ab - Ab)$, а значит возможность ее отнесения к тому или иному классу исполняемых программ.

Модель ИИС может быть формально представлена в виде следующего кортежа [10, 11]:

$$\begin{aligned} \text{ModAINet} = \langle &\text{Ab}, \text{N}, \text{Ag}, \text{Q}, \text{L}, \text{F}, \text{A}, \text{Sel}, \text{Clon}, \text{N}_\text{c}, \\ &\text{Mut}, \text{Edit}, \text{Supp}, \text{C}, \text{C}^*, \text{H}, \zeta, \text{M}_\text{q}, \text{M}_\text{q}^*, \text{B}, \sigma_\text{d}, \sigma_\text{s}, \text{G}, \tau \rangle, \end{aligned} \qquad (3)$$

где Ab – популяция (множество) антител $(Ab \in S^{N \times L}, Ab = Ab_{\{v\}} \cup Ab_{\{m\}})$; $Ab_{\{m\}}$ – общая память антител $(Ab_{\{m\}} \in S^{m \times L}, m \le N)$; $Ab_{\{v\}}$ – число новых v антител, помещенных в множество антител Ab $(Ab_{\{v\}} \in S^{v \times L}, v \le N))$; N – размер популяции антител; Ag – популяция (множество) антигенов $(Ag \in S^{Q \times L})$; Q – размер популяции антигенов; L – количество параметров (признаков) антитела или антигена; F – матрица аффинностей антител Ab_i по отношению к антигенам Ag_q с элементами $f_{i,q}, i = \overline{1,N}, q = \overline{1,Q}$; A – матрица аффинностей между каждой парой антител $(Ab_i - Ab_j)$ с элементами $a_{i,j}$, $i, j = \overline{1,N}$; Sel – оператор селекции; Clon – оператор клонирования; N_c – количество антител для клонирования; Mut – оператор мутации; Edit – оператор редактирования (удаление антител); Supp – оператор суппрессии (сжатие сети); C – популяция антител из N_c клонов, генерируемая

из множества антител Ab ($C \in S^{N_c \times L}$); C^* – популяция антител из C после полного созревания аффинности; H – матрица, содержащая аффинности между каждым элементом c_r^* из множества антител C^* и каждым элементом Ag_q из множества антигенов Ag с элементами $h_{r,q}$, $r = \overline{1,R}, q = \overline{1,Q}$; ζ – процент отбираемых зрелых антител; M_q – клональная память для антигена Ag_q (оставшиеся антитела после процесса клонального подавления); M_q^* – результирующая клональная память для антигена Ag_q; B – матрица аффинностей между каждой парой антител памяти M_q^* с элементами $b_{i,q}, i = \overline{1,C^*}, q = \overline{1,Q}$; σ_d – естественный порог смертности; σ_s – порог сжатия сети, G – количество поколений; τ – условие завершения работы алгоритма.

В соответствии с теорией иммунной сети существующие клетки (антитела) соперничают за распознавание антигена, результатом чего является адаптивный иммунный ответ. Кроме того, распознавание антитела антителом $(Ab - Ab)$ стимулирует сжатие сети.

В модели иммунной сети сжатие реализуется путем элиминации распознающих самих себя антител. Сжатие ограничивается некоторым порогом σ_s. Каждая пара $(Ag_q - Ab_i)$ будет взаимодействовать в пределах пространства $S^{N \times L}$ с аффинностью $f_{i,q}$, $i = \overline{1,N}$, $q = \overline{1,Q}$, которая отражает возможность начала адаптивного иммунного ответа. Аналогично, аффинность $a_{i,j}$ отражает степень схожести каждой пары $(Ab_i - Ab_j)$, $i, j = \overline{1,N}$.

Тогда алгоритм обучения иммунной сети может быть представлен следующим образом:

1. Для каждой итерации:

1.1. Для каждого антигена Ag_q, $q = \overline{1,Q}$, ($Ag_q \in Ag$):

1.1.1. Определяется его аффинность $f_{i,q}, i = \overline{1,N}, q = \overline{1,Q}$ для всех антител Ab_i:

$$f_{i,q} = 1/(1 + D_{i,q}), i = \overline{1,N}, q = \overline{1,Q},$$

где $D_{i,q} = \left\| Ab_i - Ag_q \right\|$.

1.1.2. Из множества антител Ab выбирается подмножество $Ab_{\{n\}}$, состоящее из n антител с наивысшей аффинностью (оператор селекции Sel).

1.1.3. Выбранные n антител подвергаются клонированию Clon в зависимости от их антигенной аффинности $f_{i,q}, i = \overline{1,N}, q = \overline{1,Q}$, образуя множество клонов C. Кратность клонирования антитела N_c регулируется в процессе работы иммунного алгоритма в зависимости от аффинности антитела $f_{i,q}, i = \overline{1,N}, q = \overline{1,Q}$ по соотношению [12]:

$$N_c(Ab_i) = \begin{cases} N_{c\_min}, \text{ если } f_{i,q} \le f_{i,q,best} * 0.3; \\ N_{c\_max}, \text{ если } f_{i,q} > f_{i,q,best} * 0.7; \\ \alpha_i * N_{c\_min} + (1 - \alpha_i) * N_{c\_max}, \text{ в ост.случ.}; \end{cases}$$

где $\alpha_i = \dfrac{f_{i,q} - f_{i,q,best} * 0.3}{f_{i,q} * 0.4}$, N_{c_min} и N_{c_max} – минимальная и максимальная кратность клонирования антитела Ab_i соответственно; $f_{i,q}$ – значение аффинности антитела Ab_i к антигенам $Ag_q, i = \overline{1,N}, q = \overline{1,Q}$; $f_{i,q,best}$ – лучшее значение аффинности, полученное в текущем поколении.

1.1.4. Множество клонов C подвергается процессу управляемой мутации Mut, в результате чего образуется множество антител C^*, в котором каждое антитело c_r^*, $r = \overline{1,N_c}$ претерпевает мутацию интенсивности α_r. Значение α_r зависит от антигенной аффинности $f_{r,q}$ родительского (по отношению к c_r^*) антитела. Чем выше аффинность, тем ниже интенсивность мутации [12]:

$$c_r^* = c_r^* + \alpha_r (Ag_q - c_r^*), \; r = \overline{1,N_c}.$$

1.1.5. Определяются аффинности $h_{r,q} = 1/(1 + D_{r,q})$ всех элементов множества C^* к антигенам Ag_q, где $D_{r,q} = \left\| c_r^* - Ag_q \right\|, r = \overline{1,N_c}$, $q = \overline{1,Q}$.

1.1.6. Из множества C^* выбирается ζ антител с наивысшим значением $h_{r,q}$ и помещаются в клональную память M_q.

1.1.7. Апоптоз (естественная смерть клетки): из клональной памяти M_q удаляются все элементы, для которых $h_{r,q} < \sigma_d$.

1.1.8. Определяются между собой аффинности $b_{r,p} = 1/(1 + D_{r,p})$ всех клонов памяти, где

$$D_{r,p} = \left\| c_r^* - c_p^* \right\|, r, p = \overline{1,N_c}.$$

1.1.9. Клональное подавление Edit: удаляются все клоны памяти, для которых $b_{r,p} < \sigma_s$.

1.1.10. Набор антител общей памяти $Ab_{\{m\}}$ конкатенируется с полученной клональной памятью M_q^*: для Ag_q : $Ab_{\{m\}} \leftarrow \left(Ab_{\{m\}}.M_q^*\right)$.

1.2. Определяются между собой аффинности $b_{i,q} = 1/(1+D_{i,q})$ всех антител памяти из множества $Ab_{\{m\}}$, где $D_{i,q} = \left\| Ab_{\{m\}}^i - Ab_{\{m\}}^q \right\| \forall i,q$.

1.3. Сжатие сети $Supp$: удаляются все антитела, для которых $b_{i,q} < \sigma_s$.

1.4. Находится полное множество антител: $Ab = \left\{Ab_{\{m\}}, Ab_{\{v\}}\right\}$.

2. Проверяется критерий завершения работы алгоритма τ.

В рассмотренном алгоритме шаги 1.1.1 – 1.1.7 описывают процессы клонального отбора и созревания аффинности. Шаги 1.1.8-1.3 моделируют работу иммунной сети. Алгоритм обучения останавливается после прохождения заранее определенного количества шагов.

В соответствии с приведенным алгоритмом производится обучение иммунной сети на ранее подготовленных наборах рейтингов встречаемости. При этом выполняется кластеризация входного множества данных на два подмножества, первое из которых будет соответствовать вредоносным программам исследуемого семейства, а второе – не вредоносным программам или же вредоносным программам из других семейств.

Модель эвристического анализатора

Для решения рассматриваемой задачи в составе ЭА, который выполняет вероятностное распознавание на основе взвешенной оценки некоторого количества признаков, использована ИИС. Схематически модель такого ЭА состоит из следующих частей (рис. 1):

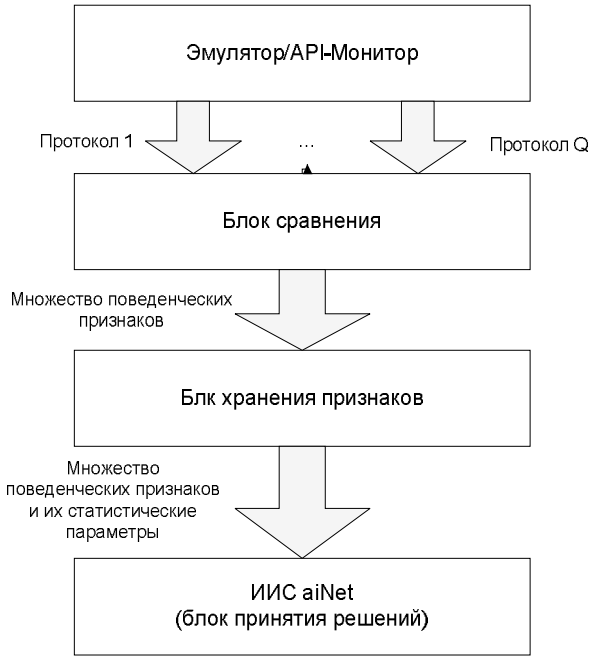

Рис. 1. Модель эвристического анализатора

1. Блок мониторинга. Функцией данного блока является мониторинг поведения вредоносных и не вредоносных объектов с целью получения протокола их работы (последовательностей вызова API функций и переданных им аргументов).

2. Блок сравнения. Данный блок принимает протоколы работы нескольких программ от блока мониторинга и сравнивает их. Результатом работы данного блока будет множество одинаковых фрагментов (признаков) в протоколах разных программ одного семейства.

3. Блок хранения признаков. Данный блок хранит в себе все признаки, выявленные блоком сравнения, и ведет статистику их встречаемости. На основе данной статистики каждому признаку присваивается рейтинг, характеризующий встречаемость данного признака, т.е. фрагмент, который был найден в протоколах всех программ, будет иметь наибольший рейтинг, а фрагмент, который найден в наименьшем количестве программ – наименьший.

4. Блок принятия решений. Основной компонент ЭА. Функция данного компонента – принятие решения о принадлежности или непринадлежности рассматриваемой программы к некоторому семейству вредоносных программ.

ЭА работает в двух режимах: обучение и распознавание. В режиме обучения происходит настройка ЭА на распознавание поведений вредоносных программ, при этом выполняются следующие действия:

1. Множество исполняемых файлов (вирусов), сходных по функционалу и принадлежащих к одному семейству, исследуется с помощью эмуляции и составляется подробные карты их действий (протоколы).

2. Протоколы сравниваются между собой для того, чтобы выявить общие закономерности в поведении объектов.

3. Найденные закономерности представляются в виде фрагментов протоколов и сохраняются в библиотеке. Далее подсчитывается рейтинг встречаемости для каждого найденного фрагмента. Этот рейтинг показывает, в каком количестве объектов, из всего множества представленных, был найден данный фрагмент. Происходит формирование выборки с рейтингами признаков вредоносных программ для обучения ИИС на положительные вердикты.

4. Выборка не вредоносных программ исследуется с помощью эмуляции аналогичным образом. После этого в протоколах работы не вредоносных программ производится поиск фрагментов поведе-

ния вредоносных программ, предварительно сохраненных в библиотеке. Для всех не вредоносных программ так же формируется выборка с рейтингами признаков вредоносных программ, которая будет использована для обучения ИИС на отрицательные вердикты.

5. Производится обучение ИИС на ранее подготовленных наборах рейтингов встречаемости. При этом выполняется кластеризация входного множества данных на два подмножества, первое из которых будет соответствовать вредоносным программам исследуемого семейства, а второе – не вредоносным программам или же вредоносным программам из других семейств.

Обученная иммунная сеть представляется множеством антител памяти $\mathrm{Ab}_{\{m\}}$ и матрицей их аффинностей B.

Множество $\mathrm{Ab}_{\{m\}}$ интерпретирует внутренние отображения антигенов, поданные на вход сети. Матрица B описывает связи между антителами, и показывает общую структуру иммунной сети. Имея матрицу аффинностей антител памяти B, можно определить структуру иммунной сети, а также принадлежность каждого антитела к соответствующему классу.

Путем измерения аффинностей антител из множества клеток памяти к антигенам из обучающего набора можно определить, какие из антител распознают антигены, соответствующие вредоносным программам из данного семейства, а какие – распознают не вредоносные программы (или вредоносные программы других семейств).

Таким образом, осуществляется идентификация полученных кластеров.

В режиме распознавания на вход уже обученной ИИС подается вектор данных, представляющий собой новый антиген, с которым обученная ИИС ранее не сталкивалась. Измерив аффинности данного антигена к антителам, содержащимся в памяти ИИС, можно определить, к какому кластеру отнести данный антиген, и таким образом вынести вердикт о принадлежности или не принадлежности антигена к множеству вредоносных программ исследуемого семейства.

В режиме распознавания выполняются следующие действия:

1. Исполняемый файл исследуется с помощью эмулятора.

2. В протоколе исследуемого файла производится поиск фрагментов поведения вредоносных программ из библиотеки. Из библиотеки извлекаются рейтинги для найденных фрагментов.

3. Рейтинги подаются на входы ИИС.

Схема движения данных представлена на рис. 2.

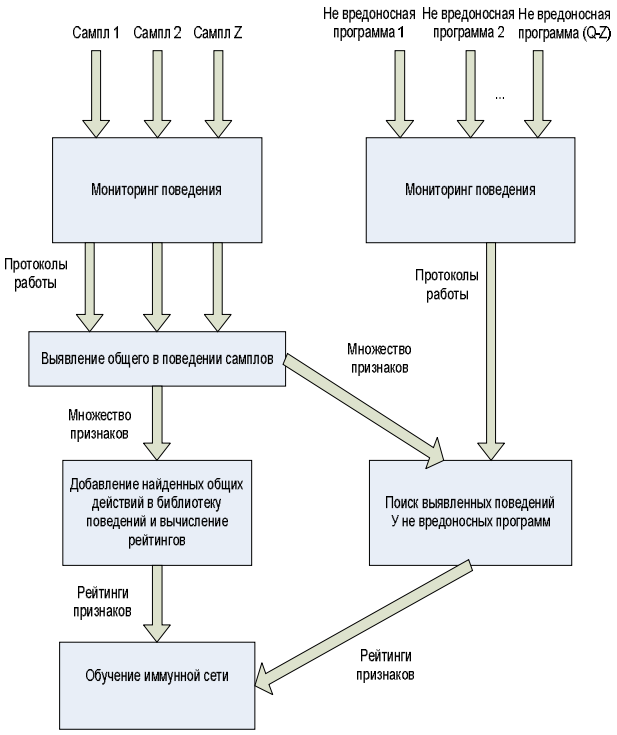

Рис. 2. Схема движения данных

Моделирование работы эвристического анализатора

Для моделирования работы эвристического анализатора был выбран следующий набор инструментов:

1) моделирование блока мониторинга производилось с использованием специального программного обеспечения, которое эмулирует работу центрального процессора, API функций ОС и ее внутренних структур и идеально подходит для задачи мониторинга поведения программ и сбора необходимых данных;

2) моделирование блока сравнения признаков производилось с помощью специально разработанных утилит Threader и Matcher, которые предназначены для преобразования и исследования информации, получаемой от эмулятора системы;

3) моделирование ИИС проводилось при помощи программного пакета MATLAB R2013a. Реализованная ИИС получает в качестве входных данных файл Ag_matrix.dat, содержащий обучающую выборку. Результатом работы алгоритма являются:

– множество антител памяти $\mathrm{Ab}_{\{m\}}$ (записывается в файл M_matrix.dat);

– матрица аффинностей между антителами памяти $B = \left[b_{i,q} \right]$ (используется для построения дендрограммы и гистограммы минимального связывающего дерева);

– графическая интерпретация обученной ИИС, а именно, дендрограмма и гистограмма минималь-

ного связывающего дерева (строятся при помощи средств программного пакета MATLAB R2013a);

4) анализ данных, полученных в результате обучения ИИС, а также распознавание новых антигенов производилось посредством специально разработанной утилиты aiNet_Help_Tool. Разработанная утилита принимает в качестве входных параметров файлы Ag_matrix.dat, M_matrix.dat, а также файл, содержащий данные для распознавания новых антигенов (New_Ag.dat).

На этапе получения данных для обработки предполагался запуск вредоносных программ на эмуляторе и получение протоколов их работы. Для эксперимента были взяты следующие вредоносные программы, а именно, семейство троянских программ, предназначенных для похищения паролей, включая следующие модификации:

Trojan-Spy.Win32.Zbot.jqye;
Trojan.Win32.Cidox.afaq;
Trojan.Win32.ShipUp.iwr;
Trojan-Dropper.Win32.Baky.c;
Trojan.FakeAV;
Trojan.Flame.A.

Данные модификации были запущены на эмуляторе, и для каждой из них был получен протокол. Была выполнена предварительная обработка полученных протоколов работы вредоносных программ. Полученные протоколы были проанализированы с помощью программы Threader, а затем сравнивались попарно каждый с каждым при помощи специально разработанной утилиты Matcher. Результатом работы данной программы явилось множество общих для всех входных протоколов фрагментов (характерных поведенческих признаков). Было выполнено распознавание вредоносных программ семейства Trojan. Для этого была сформирована обучающая выборка из рейтингов встречаемости признаков во вредоносных и не вредоносных программах и был запущен процесс обучения иммунной сети со следующими параметрами:

– порог сжатия иммунной сети: $\sigma_s = 0{,}1$;
– процент отбираемых зрелых антител: $\zeta = 20\%$;
– порог естественной смертности: $\sigma_d = 0{,}02$;
– количество отбираемых антител с максимальной аффинностью для каждого антитела (селекция): $n = 2$;
– количество поколений обучения: $G = 20$.

Графическая интерпретация обученной иммунной сети приведена на рис. 3, 4. Исходя из графической интерпретации обученной иммунной сети, можно сделать вывод о наличии во входных данных двух кластеров. Первый кластер составляют антитела: 4, 8, 9, 10, 6, 11, 3; второй – антитела: 1, 2, 5, 7, 12. Дальнейшая обработка полученных данных, а также распознавание новых антигенов проведено при помощи утилиты aiNet_Help_Tool. В качестве новых антигенов для распознавания сетью использовались две вредоносные программы и две не вредоносные программы, а именно:

1-й антиген – Trojan.OlympicGames;
2-й антиген – Win32.Worm.Prolaco.S;
3-й антиген – Mirabilis ICQ;
4-й антиген – Opera Web Browser.

По результатам работы утилиты aiNet_Help_Tool антитела 1, 2, 5, 7, 12 распознали вредоносные антигены (2-й кластер); антитела 3, 4, 6, 8, 9, 10, 11 распознали не вредоносные антигены или антигены из других вредоносных семейств (1-й кластер). Новые антигены распознаны верно: 1-й и 2-й являются вредоносными, а 3-й и 4-й – не вредоносными.

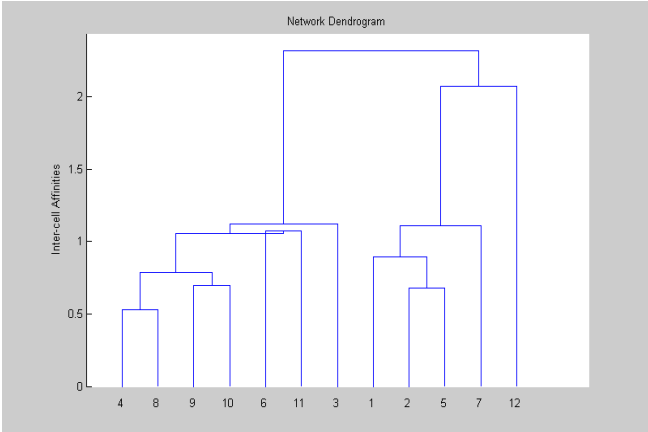

Рис. 3. Дендрограмма обученной иммунной сети

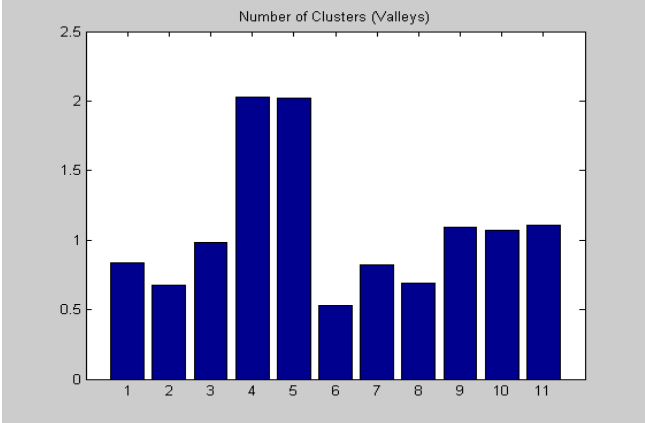

Рис. 4. Гистограмма минимального связывающего дерева обученной иммунной сети

Можно сделать вывод, что разработанный ЭА на основе ИИС способен распознавать новые модификации вредоносных программ из рассматриваемого семейства. Следовательно, разработанный ЭА на основе ИИС успешно решает возложенную на него задачу.

Выводы

Рассмотрено решение актуальной задачи выявления и распознавания как существующих, так и новых модификаций вредоносных программ на ос-

нове ИИС. Рассмотрена модель ИИС и предложен алгоритм ее обучения, детектирования и анализа вредоносных программ.

Предложена модель ЭА, основным компонентом которого является ИИС, используемая для принятия решения о принадлежности или непринадлежности рассматриваемой программы к семейству вредоносных программ. Рассмотрена работа ЭА в режимах обучения и распознавания. В режиме обучения происходит настройка ЭА на распознавание поведений вредоносных программ. В режиме распознавания исполняемый файл исследуется с помощью эмулятора с целью поиска фрагментов поведения вредоносных программ из библиотеки, из которой извлекаются рейтинги для найденных фрагментов, подаваемые на входы ИИС.

Для экспериментальных исследований были взяты вредоносные программы семейства троянских программ, предназначенные для похищения паролей. Результаты моделирования показали, что предложенный ЭА на основе ИИС способен распознавать новые модификации вредоносных программ.

Дальнейшие исследования ориентированы на разработку моделей обнаружения и анализа вредоносных программ на основе агентно-ориентированного подхода к представлению ИИС.

Список литературы

1. Шибаева Т.А. Защита от внедрения и запуска вредоносных программ / Т.А. Шибаева, А.Ю. Щеглов, А.А. Оголюк // Вопросы защиты информации. – 2011. – № 2. – С. 26-30.

2. Новиков Е.А. Сравнительный анализ методов обнаружения вторжений / Е.А. Новиков, А.А. Краснопевцев // Безопасность информационных технологий. – 2012. – № 1. – С. 47-50.

3. Харченко С.С. Сигнатурный анализ программного кода / С.С. Харченко, Е.М. Давыдова, С.В. Тимченко // Ползуновский вестник. – 2012. – № 3. – С. 60-64.

4. Медведев Н.В. Применение метода статического сигнатурного анализа для выявления дефектов безопасности веб-приложений / Н.В. Медведев, А.С. Марков, А.А. Федин // Наука и образование: электронное научно-техническое издание. – 2012. – № 9. – С. 21-31.

5. Абрамов Е.С. Метод обнаружения распределенных информационных воздействий на основе гибридной нейронной сети / Е.С. Абрамов, И.Д. Сидоров // Известия Южного федерального университета. Технические науки. – 2009. – Т. 100. – № 11. – С. 154-164.

6. Емельянова Ю.Г. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы / Ю.Г. Емельянова, А.А. Талалаев, И.П. Тищенко, В.П. Фраленко // Программные системы: теория и приложения. – 2011. – Т. 2. – № 3. – С. 3-15.

7. Гаврилов А.В. Применение постоянно модифицирующихся нейронных сетей для защиты программного обеспечения / А.В. Гаврилов // Нейрокомпьютеры: разработка, применение. – 2008. – № 1-2. – С. 90-101.

8. Гаврилов А.В. Применение иммунных систем в целях защиты корпоративной информации от нецелевого использования / А.В. Гаврилов, А.В. Тихомиров // Известия Южного федерального университета. Технические науки. – 2010. – Т. 108. – № 7. – С. 154-163.

9. Войцехович Л.Ю. Применение мультиагентной системы с нейросетевым классификатором для выявления атак в трафике TCP/IP / Л.Ю. Войцехович, В.А. Головко, Курош Мадани // Нейроинформатика. – 2011. – Часть 1. – С. 190-201.

10. Dasgupta D. Immunological computation, theory and applications / D. Dasgupta, L.F. Nino. – CRC Press, 2009. – 298 p.

11. De Castro L.N. AiNet: An Artificial Immune Network for Data Analysis / L.N. De Castro, F.J. Von Zuben // Data Mining: A Heuristic Approach. / Eds.: H.A. Abbass, R.A. Sarker, C.S. Newton. –Hershey: Idea Group Publishing, 2001. – P. 231-259.

12. Designing Ensembles of Fuzzy Classification Systems: An Immune-Inspired Approach / P.A.D. Castro, G.P. Coelho, M.F. Caetano [et al.] // Springer Lecture Notes in Computer Science. – 2005. – Vol. 3627. – P. 469-482.

Поступила в редколлегию 12.08.2013

Рецензент: д-р техн. наук, проф. С.Г. Удовенко, Харьковский национальный университет радиоэлектроники, Харьков.

МОДЕЛЬ ЕВРИСТИЧНОГО АНАЛІЗАТОРА ШКІДЛИВИХ ПРОГРАМ НА ОСНОВІ ШТУЧНОЇ ІМУННОЇ МЕРЕЖІ

М.М. Корабльов, М.В. Кушнарьов

Запропоновано модель евристичного аналізатора шкідливих програм, основним компонентом якої є штучна імунна мережа (ШІМ), за допомогою якої здійснюється навчання, детектування та аналіз як існуючих, так і нових модифікацій вірусів. Для вирішення даної задачі визначаються рейтинги для кожного фрагмента програми, що зустрічається, які показують, в якій кількості об'єктів був знайдений даний фрагмент. Проведено моделювання роботи евристичного аналізатора, що показує ефективність запропонованої моделі.

Ключові слова: *евристичний аналізатор, штучна імунна мережа, шкідлива програма, модель. афінність, антитіло, антиген.*

MODEL HEURISTIC ANALYZER MALWARE BASED ON ARTIFICIAL IMMUNE NETWORK

N.M. Korablyov, M.V. Kushnaryov

A model of the heuristic malware analyzer, the main component of which is an artificial immune network (AINet), it use for training, detection and analysis of both existing and new variants of viruses. To solve the problem of occurrence ratings are determined for each program's fragment that show how many objects of this fragment was found. The simulation of the heuristic analyzer was made, it showing the effectiveness of the proposed model.

Keywords: *heuristic analyzer, artificial immune network, malicious program, model, affinity, antibody, antigen.*