УДК 004.89

Н. М. Кораблев[1], М. В. Кушнарев[2]

[1] ХНУРЭ, г. Харьков, Украина, korablev.nm@gmail.com;

[2] ХНУРЭ, г. Харьков, Украина, mauxion@gmail.com

МУЛЬТИАГЕНТНАЯ МОДЕЛЬ ИСКУССТВЕННОЙ ИММУННОЙ СИСТЕМЫ ДЛЯ РАСПОЗНАВАНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

В статье предлагается мультиагентная модель искусственной иммунной системы для определения как существующих, так и новых модификаций вредоносных программ, которая позволяет распознать вирус с минимально возможными затратами системных ресурсов. Оценка эффективности предложенной модели выполнена путем сравнительного анализа с моделями на основе искусственной нейронной сети и искусственной иммунной системы. Представлены результаты экспериментальных исследований, демонстрирующие особенности предлагаемого подхода.

МОДЕЛЬ, МУЛЬТИАГЕНТНАЯ СИСТЕМА, ИСКУССТВЕННАЯ ИММУННАЯ СИСТЕМА, ВРЕДОНОСНАЯ ПРОГРАММА, АГЕНТ, АНТИТЕЛО, АНТИГЕН, АФФИННОСТЬ, КЛОНИРОВАНИЕ, МУТАЦИЯ, СУПРЕССИЯ

Введение

В настоящее время вредоносные программы стали частью бизнеса, их создание и поддержка поставлены на коммерческую основу, что в свою очередь положительно влияет на качество применяемых в них технологий скрытия от обнаружения антивирусными программами. Одна из наиболее эффективных и часто используемых технологий скрытия – применение шифрования. Новые модификации появляются каждый день, при этом различия между модификациями в большинстве случаев заключаются в использовании разных ключей и немного измененных алгоритмов шифрования. Аналитикам антивирусных компаний приходится добавлять каждую новую модификацию в базы и выпускать их срочным обновлением, в то время как большое количество компьютеров уже оказывается зараженными. Это происходит по причине недостаточной эффективности применяемых сегодня эвристических анализаторов (ЭА) – средств распознавания неизвестных модификаций.

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 подходов к распознаванию вредоносных программ, которые должны быть основаны на анализе их поведения и действовать в обход шифрования на более высоком уровне. Новые подходы должны позволять эффективно распознавать как старые, так и новые модификации вирусов с минимально возможной загрузкой системы, а также защищать компьютерные сети без необходимости обновления антивирусного программного обеспечения [1, 2].

Полностью эффективных способов борьбы с угрозами на сегодняшний день не существует, но использование в составе ЭА искусственных нейронных сетей (ИНС) [3-5], искусственных иммунных систем (ИИС) [6-8] и мультиагентных систем (МАС) [9] позволяет идентифицировать широкий класс вирусов. В составе ЭА в настоящее время активно используются ИИС, которые относятся к классу автоматизированных вычислительных интеллектуальных систем, использующих принципы иммунной системы позвоночных. В [8] предложена модель ЭА вредоносных программ, основным компонентом которой является искусственная иммунная сеть, с помощью которой осуществляется обучение, детектирование и распознавание как существующих, так и новых модификаций вирусов.

С другой стороны, ЭА на основе ИИС можно представить в виде МАС, состоящей из множества автономных модулей – агентов, функционирующих в среде. Главное достоинство МАС – это гибкость, которая может быть дополнена и модифицирована без изменения значительной части программы [10]. В соответствии с этим в работе предлагается мультиагентная модель (МАМ) представления ИИС для определения вредоносных программ, которая является гибкой, легко масштабируемой и позволяет распознавать вирусы с минимально возможными затратами системных ресурсов.

1. Постановка задачи

Пусть имеется множество исполняемых файлов N_j, $j=\overline{1,M}$, которые могут содержать вредоносные коды и представляются антигенами ИИС. Пусть имеется множество программных агентов S_i, $i=\overline{1,N}$, которые должны распознавать исполняемые файлы и представляются антителами ИИС. Окружающая среда *Env* представляет собой операционную систему компьютера, в которой взаимодействуют как исполняемые файлы N_j с программными агентами S_i, так и программные агенты между собой. Предполагается, что для каждого исполняемого файла N_j, $j=\overline{1,M}$ существует информационный вектор (вектор признаков) $A_j=\left[a_{j,1},a_{j,2},...a_{j,m}\right]$ из m элементов, который может содержать вредоносные коды. У каждого программного агента S_i, $i=\overline{1,N}$ также есть информационный вектор (вектор признаков) $B_i=\left[b_{i,1},b_{i,2},...b_{i,n}\right]$ из n элементов, определяющий его самостоятельные цели. При этом векторы признаков и особенности как программных агентов, так и исполняемых файлов могут отличаться друг от друга.

Предполагается, что у программных агентов S_i есть способность идентифицировать исполняемые файлы N_j в сенсорных областях *SNs* (*Sensory Neighborhoods*) с помощью двумерного массива значений аффинностей:

$$Aff_{S_i,N_j}=1/(1+D_{i,j}),i=\overline{1,N},j=\overline{1,M}, \qquad (1)$$

где $D_{i,j}=\left\|S_i-N_j\right\|$ – евклидово расстояние.

Кроме того, программные агенты также обладают способностью сообщать информацию об исполняемых файлах другим программным агентам в коммуникационных областях *CNs* (*Communication Neighborhoods*) с помощью двумерного массива значений аффинностей:

$$Aff_{S_r,S_p}=1/(1+D_{r,p}),\ r,p=\overline{1,N}, \qquad (2)$$

где $D_{r,p}=\left\|S_r-S_p\right\|$ – евклидово расстояние.

Необходимо разработать МАМ представления ИИС, с помощью которой будет выполняться распознавание вредоносных кодов в исполняемых файлах на основе организации взаимодействий как между программными агентами и исполняемыми файлами, так и программных агентов между собой.

2. Модель ИИС на основе МАС

Предлагаемая модель представления ИИС с помощью МАС приведена на рис. 1 и описывается следующим кортежем:

$$MAMAIS=$$
$$=\langle El,Attr,Env,Rl,SNs,CNs,RAct,CAct,Ev\rangle, \qquad (3)$$

где *El* – множество элементов системы, характеризующихся набором атрибутов-признаков (фрагментов программ) *Attr* и функционирующих в окружающей среде *Env*, которая представляет собой операционную систему компьютера, находятся в определенных отношениях *Rl*, позволяющих взаимодействовать друг с другом в сенсорных *SNs* и коммуникационных *CNs* областях, обладают возможностью выполнять реактивные *RAct* и коммуникативные *CAct* действия для достижения цели, изменяя свои атрибуты *Attr* в процессе эволюции *Ev*.

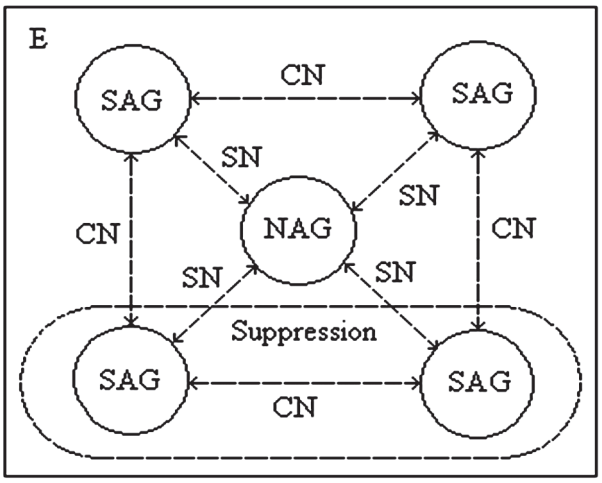

Рис. 1. Представление ИИС в виде МАС

Предлагаемая модель определяет антиген и антитело как два агента с различными особенностями и целями. Поэтому в модели МАС представления ИИС используются только два типа агентов: антигены N_j, $j=\overline{1,M}$ представлены как non-self агенты (*NAGs*), а антителам S_i, $i=\overline{1,N}$ соответствуют self агенты (*SAGs*). Следовательно, множество элементов *El* МАС состоит из self агентов *SAGs* и non-self агентов *NAGs* ($El=SAG_i\cup NAG_j$), каждый из которых характеризуется наборами атрибутов-признаков $\left\{a_{j,k}\right\}$ и $\left\{b_{i,l}\right\}$, $i=\overline{1,N}$, $j=\overline{1,M}$, $k=\overline{1,m}$, $l=\overline{1,n}$.

Модель МАС представлена в виде иммунной сети Ерне [11] (рис.1), а отношения *Rl* и взаимодействия self агентов *SAGs* с non-self агентами *NAGs*, а также self агентов *SAGs* между собой определяются степенью их близости, оцениваемой значениями аффинностей (1) и (2). Реактивные действия *RAct* self агентов *SAGs* по отношению к non-self агентам *NAGs* определяются величиною порога аффинности T_{NAG} в сенсорных областях *SNs* и зависят от значений аффинностей (1). Коммуникационные действия *CAct* self агентов *SAGs* между собой определяются величиною своего порога аффинности T_{SAG} в коммуникационных областях *CNs*, зависят от значений аффинностей (2) и могут приводить к суппрессии сети.

Алгоритм распознавания вредоносных программ на основе МАМ, описывающей ИИС, может быть представлен в виде следующей последовательности выполнения операторов:

$$
\begin{aligned}
&MDA(NAGs, SAGs, Aff_{S,N}, Aff_{S,S}, SNs, CNs, T_{NAG}, T_{SAG}, p) = \\
&= NAGpresent\ (NAGs, SAGs, Aff_{S,N}) \rightarrow \\
&\rightarrow SAGSelect\ (SAGs, SNs, T_{NAG}) \rightarrow \\
&\rightarrow SAGs_{cl}\ [Clon\ (SAGs, T_{NAG}) \rightarrow Mut\ (SAGs_{cl}) \rightarrow \\
&\rightarrow Clpresent\ (NAGs, SAGs_{cl}, Aff_{S,N}) \rightarrow \\
&\rightarrow ClSel\ (SAGs_{cl}, SNs, T_{NAG}) \rightarrow \\
&\rightarrow SAGpresent\ (SAGs, SAGs_{cl}, Aff_{S,S}) \rightarrow \\
&\rightarrow Supp\ (SAGs, SAGs_{cl}, CNs, T_{SAG}) \rightarrow \\
&\rightarrow Age\ (SAGs, SAGs_{cl}) \rightarrow TermTest\ (p)],
\end{aligned}
\tag{4}
$$

где $NAGpresent(NAGs, SAGs, Aff_{S,N})$ – оператор представления self агентов $SAGs$ non-self агентам $NAGs$; $SAGSelect(SAGs, SNs, T_{NAG})$ – оператор отбора self агентов $SAGs$; $Clon(SAGs, T_{NAG})$ – оператор клонирования отобранных self агентов $SAGs$; $Mut(SAGs_{cl})$ – оператор мутации клонов; $Clpresent(NAGs, SAGs_{cl}, Aff_{S,N})$ – оператор представления клонированных self агентов $SAGs_{cl}$ non-self агентам $NAGs$; $ClSel(SAGs_{cl}, SNs, T_{NAG})$ – оператор отбора клонированных self агентов $SAGs_{cl}$; $SAGpresent(SAGs, SAGs_{cl}, Aff_{S,S})$ – оператор представления отобранных и клонированных self агентов друг другу; $Supp(SAGs, SAGs_{cl}, CNs, T_{SAG})$ – оператор суппрессии self агентов; $Age(SAGs, SAGs_{cl})$ – оператор старения; $TermTest(p)$ – процедура проверки критерия останова.

Следует отметить, что операторы представления $NAGpresent(NAGs, SAGs, Aff_{S,N})$ и отбора $SAGSelect(SAGs, SNs, T_{NAG})$ вызываются только один раз для исходной популяции self агентов $SAGs$ и не используются в дальнейшем. Остальные операторы используются в цикле в зависимости от предельного количества популяций. Оператор представления $NAGpresent(NAGs, SAGs, Aff_{S,N})$ реализует функцию определения аффинностей $Aff_{S,N}$ между self агентами $SAGs$ и non-self агентами $NAGs$ в соответствии с (1). Оператор отбора $SAGSelect(SAGs, SNs, T_{NAG})$ используется для выделения self агентов $SAGs$ из исходной популяции, у которых аффинность с non-self агентами $NAGs$ удовлетворяет условию:

$$Aff_{S_i,N_j} \geq T_{NAG},\ i = \overline{1,N},\ j = \overline{1,M}, \tag{5}$$

т.е. отбираются те self агенты $SAGs$, которые попадают в сенсорную область SN соответствующего non-self агента NAG и значения аффинностей которых превышает заданный порог T_{NAG}. Оператор клонирования $Clon(SAGs, T_{NAG})$ используется для клонирования self агентов $SAGs$, прошедших отбор. При этом применяется пропорциональное клонирование [12], при котором количество клонов, создаваемых для каждого отобранного self агента SAG, прямо пропорционально его аффинности с non-self агентами $NAGs$, превышающей порог T_{NAG}:

$$N_{cl} = N * Aff_{S,N}, \tag{6}$$

где N – общее количество self агентов $SAGs$.

Оператор мутации клонов $Mut(SAGs_{cl})$ используется для внесения изменений в признаки клонов self агентов $SAGs_{cl}$ для достижения их большей близости j-му non-self агенту NAG_j. При использовании оператора случайной мутации изменение значения каждого k-го признака i-го клона $b_{i,k}$ производится следующим образом:

$$b_{i,k} = b_{i,k} \pm \mu_i \cdot ((Aff_{S_i,N_j})^{-1} - 1),\ i = \overline{1,N_{cl}},\ k = \overline{1,n}, \tag{7}$$

где $\propto_i$ – коэффициент мутации, который определяет характер изменения признаков i-го клона. В существующих иммунных методах наилучший эффект даёт обратно пропорциональная мутация [12], при которой коэффициент мутации зависит только от значения аффинности i-го клонированного self агента SAG_{cl_i} к j-му non-self агенту NAG_j и определяется следующим образом:

$$\mu_i = rand(0; 1 - Aff_{S_i,N_j}),\ i = \overline{1,N_{cl}}. \tag{8}$$

Следует отметить, что при случайной мутации знак $\pm$ в выражении (7) определяется случайным образом.

Оператор представления клонированных self агентов $SAGs_{cl}$ non-self агентам $NAGs$ $Clpresent(NAGs, SAGs_{cl}, Aff_{S,N})$ используется для определения значений аффинностей между мутированными клонами $SAGs_{cl}$ и non-self агентам $NAGs$ в соответствии с (1). Таким образом, вместо всей популяции non-self агентов $NAGs$ во взаимодействие с клонами вступает только небольшая группа $NAGs$, что приводит к уменьшению количества вычислительных операций.

Оператор отбора $ClSel(SAGs_{cl}, SN, T_{NAG})$ клонированных self агентов $SAGs_{cl}$ используется для проведения отбора клонов, которые попали в сенсорные области SNs non-self агентов $NAGs$ и превысили порог аффинности T_{NAG} в соответствии с условием (5), а также для определения клонов с наилучшей аффинностью с одним из non-self агентов $NAGs$.

Оператор представления отобранных и клонированных self агентов друг другу $SAGpresent(SAGs, SAGs_{cl}, Aff_{S,S})$ используется для определения значений аффинностей $Aff_{S,S}$ между ними в соответствии с (2).

Оператор суппрессии $Supp(SAGs, SAGs_{cl}, CN, T_{SAG})$ используется для поглощения self агентов, попадающих в коммуникационные области CNs других self агентов $SAGs$ и $SAGs_{cl}$ и превышающих порог аффинности T_{SAG}. С помощью этого оператора реализуется функция передачи информации о non-self агентах $NAGs$ другим self агентам $SAGs$ в коммуникационных областях CNs, которая оценивается значениями аффинностей (2) и удовлетворяет условию:

$$Af_{S_r,S_p} \geq T_{SAG}, r, p = \overline{1,N} . \quad (9)$$

Оператор старения $Age(SAGs, SAGs_{cl})$ используется для замены клонированных self агентов $SAGs$ клонами $SAGs_{cl}$, которые остались в результате отбора. Клонированные self агенты заменяется своими клонами в том случае, если их аффинность хуже аффинности клонов. В противном случае клоны удаляются из памяти, а self агенты остаются для клонирования на следующей популяции.

Процедура проверки критерия останова $TermTest(p)$ используется для завершения работы алгоритма, которое может произойти в случае достижения состояния полной близости для self агентов, прошедших отбор, либо достижения предельного количества популяций self агентов.

Процесс распознавания вредоносных программ с помощью предлагаемой модели представляется следующим образом:

1. Формирование популяции self агентов $SAGs$ случайным образом.
2. Для каждого non-self агента NAG определяется его близость со всеми self агентами $SAGs$ в сенсорных областях SNs в соответствии с (1).
3. Из множества self агентов $SAGs$ выбирается подмножество, удовлетворяющее условию (5).
4. Выбранные self агенты подвергаются клонированию в соответствии с (6).
5. Множество клонов подвергается процессу мутации в соответствии с (7), образуя множество мутированных self агентов.
6. Определяется близость всех элементов этого множества к non-self агентам в соответствии с (1).
7. Из этого множества выбираются self агенты с наивысшим значением близости, удовлетворяющие условию (5), и помещаются в клональную память.
8. Определяется близость всех клонов памяти между собой в соответствии с (2).
9. Сжатие сети: поглощаются все self агенты, для которых выполняется условие (9).
10. Клональное подавление: удаляются все клоны памяти, для которых не выполняется условие (5).
11. Проверка критерия останова. В случае его достижения – переход к 12, иначе переход к 4.
12. Конец.

В соответствии с приведенным алгоритмом производится распознавание вредоносных программ. При этом выполняется кластеризация входного множества исполняемых файлов на два подмножества, первое из которых будет соответствовать вредоносным программам исследуемого семейства, а второе – невредоносным программам или же вредоносным программам из других семейств.

3. Экспериментальные исследования

Для проведения экспериментальных исследований использовался специализированный набор инструментов: разработанные утилиты Threader и Matcher, которые предназначены для преобразования и исследования информации. Для сравнительного анализа предлагаемой МАМ с другими моделями (ИНС [5, 6] и ИИС [7, 8]) было выбрано 5 вредоносных программ, принадлежащих к классу загрузочных вирусов, которые однозначно были распознаны при использовании всех моделей.

На этапе обработки данных производился запуск вредоносных программ на эмуляторе и получение протоколов их работы. Полученные протоколы были проанализированы с помощью утилиты Threader, а затем сравнивались попарно каждый с каждым при помощи утилиты Matcher. Результатом работы явилось множество общих для всех входных протоколов фрагментов (характерных поведенческих признаков).

В качестве критериев эффективности были выбраны следующие системные ресурсы: 1) время анализа программ; 2) загрузка центрального процессора (ЦП); 3) загрузка оперативной памяти (ОП). Сравнительные результаты потребления системных ресурсов при моделировании выбранных вредоносных программ по указанным критериям с помощью различных подходов приведены в табл. 1.

Таблица 1

Потребление системных ресурсов

Наименование вируса	Время анализа программ, с			Загрузка ЦП, %			Загрузка ОП, %		
	ИНС	ИИС	МАМ	ИНС	ИИС	МАМ	ИНС	ИИС	МАМ
HackTool.Win32.BruteForce.ben	7	12	8	33	44	32	52	57	50
Worm.Win32.AutoRun.faka	11	9	6	35	42	35	38	42	39
Virus.Win9x.Spaces.1245	9	10	7	13	26	13	40	39	38
not-a-virus:Downloader.Win32.LMN.uhv	8	10	6	31	38	28	36	35	37
not-a-virus:AdWare.Win32.ScreenSaver.wym	8	11	8	40	45	38	24	27	25

Из таблицы видно, что для тестируемых вредоносных программ с помощью трех основных подходов (ИНС, ИИС и МАМ) как по времени анализа, так и по загрузке ЦП и ОП наилучшие средние показатели имеет предлагаемая МАМ представления ИИС. Таким образом, использование предложенной МАМ, описывающей ИИС, позволяет распознавать новые модификации вредоносных программ и, следовательно, успешно решать возложенную на нее задачу с минимально возможными затратами системных ресурсов.

Выводы

Рассмотрено решение актуальной задачи распознавания как существующих, так и новых

модификаций вредоносных программ на основе ИИС, моделью которой является МАС. Поскольку МАС имеют некоторое сходство с ИИС, что обеспечивает применение принципов функционирования ИИС к МАС, то в модели ИИС на основе МАС использованы только два типа агентов: антигены смоделированы как non-self агенты *NAGs* , а антитела – как self агенты *SAGs* .

Предложенная МАМ представления ИИС для распознавания вредоносных программ является гибкой, легко масштабируемой и позволяет распознать вирус с минимально возможными затратами системных ресурсов.

Были проведены сравнительные экспериментальные исследования предложенной МАМ с моделями на основе ИНС и ИИС на примере вредоносных программ, принадлежащих к классу загрузочных вирусов, которые показали, что по потреблению основных системных ресурсов (время анализа, загрузка ЦП и ОП) наилучшие средние показатели имеет предлагаемая модель.

Дальнейшие исследования ориентированы на разработку моделей обнаружения и распознавания вредоносных программ на основе агентно-ориентированного подхода.

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.** *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. **4.** *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. **5.** *Гаврилов А.В.* Применение постоянно модифицирующихся нейронных сетей для защиты программного обеспечения / А.В. Гаврилов // Нейрокомпьютеры: разработка, применение. – 2008. – № 1-2. – С. 90-101. **6.** *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. **7.** *Zekri M.*, Souici-Meslati L. Artificial Immune System for Intrusion Detection / M. Zekri, L. Souici-Meslati // Evolutionary Computation. – 2011. – V. 13, № 2. – 145-153. **8.** *Кораблев Н.М.* Модель эвристического анализатора вредоносных программ на основе искусственной иммунной сети / Н.М. Кораблев, М.В. Кушнарев // Системи обробки інформації: зб. наук. праць. – 2013. – Вип. 8 (115). – С. 216-222. **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.** *Alkhateeb F.* Multi-Agent Systems – Modeling, Interactions, Simulations and Case Studies / F. Alkhateeb, E. Al Maghayreh, I. Abu Doush // Published by InTech, Rijeka, Croatia. – 2011. – 502 p. **11.** *Jerne N.K.* Idiotypic networks and Other Preconecived Ideas / N. K. Jerne // Immunological review . – 1984. – V. 79. – P. 5–24. **12.** *Dasgupta D.* Immunological computation, theory and applications / D. Dasgupta, L.F. Nino – CRC Press, 2009. – 298 p.

Поступила в редколлегию 21.02.2014

УДК 004.89

Мультиагентна модель штучної імунної системи для розпізнавання шкідливих програм / М.М. Корабльов, М.В. Кушнарьов // Біоніка інтелекту: наук.-техн. журнал. – 2014. – № 1 (82). – С. 90–94.

Запропоновано мультиагентну модель подання штучної імунної системи, яка використовується для розпізнавання шкідливих кодів у файлах, що виконуються, на основі організації взаємодій як між програмними агентами та програмними файлами , так і програмних агентів між собою. Запропонована модель є гнучкою, легко масштабується і дозволяє розпізнавати віруси з мінімально можливими витратами системних ресурсів. Представлені результати порівняльних експериментальних досліджень, що демонструють ефективність запропонованого підходу.

Табл. 1. Іл. 1. Бібліогр.: 12 найм.

UDK 004.89

Multi Agent Model of an Artificial Immune System to Malware Detection / N.M. Korablyov, M.V. Kushnaryov // Bionics of Intelligence: Sci. Mag. – 2014. – № 1 (82). – P. 90–94.

Multi agent model representation an artificial immune system which is used for the detection of malicious code in executable files based on the organization as interactions between software agents and executable files, and software agents to each other is proposed. The proposed model is flexible, scalable and easily allows you to identify viruses with the lowest possible cost of system resources. Results of comparative experimental researches that demonstrate efficiency of the proposed approach is presented.

Tab. 1. Fig. 1. Ref.: 12 items.