УДК 004.89

Н.М. Кораблёв, М.В. Кушнарёв, О.Г. Лебедев

Харьковский национальный университет радиоэлектроники, Харьков

ОБНАРУЖЕНИЕ И АНАЛИЗ ВРЕДОНОСНЫХ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ МУЛЬТИАГЕНТНОГО ПОДХОДА

Предложено модель и структура мультиагентной системы обнаружения и анализа вредоносных программ, основными компонентами которой являются агенты двух видов: агенты-детекторы и агенты-анализаторы. Задача агента-детектора – мониторинг основных уязвимостей операционной системы и сбор данных, задача агента-анализатора – исследовать процессы и принятие решения, какие из них являются потенциальными вирусами и к какому классу вредоносных программ относятся. Проведены экспериментальные исследования, показывающие эффективность предложенного подхода.

Ключевые слова: *мультиагентная система, вредоносная программа, агент-детектор, агент-анализатор, взаимодействие агентов.*

Введение

В настоящее время количество разнообразного вредоносного программного обеспечения растет, поэтому системы обнаружения и предотвращения вторжений в компьютер не теряют свою актуальность. Существующие антивирусные продукты не могут обеспечить абсолютно надежную защиту компьютера, что, в первую очередь, связано с тем, что применяемые в антивирусах принципы поиска не позволяют обнаруживать новые разновидности вредоносных программ до их изучения аналитиками и внесения дополнений и изменений в базы антивирусных программ [1, 2].

Современные системы обнаружения вирусов используют как сигнатурный, так и эвристический методы, тем самым совмещая в себе их недостатки и преимущества. Основным способом выявления большинства компонентов вредоносных программ все еще является сигнатурная проверка. Однако, сигнатурный метод непригоден для защиты от новых и полиморфных вирусов, т.к. до тех пор, пока вирус не попал на анализ к экспертам, создать его сигнатуру невозможно. Существующие эвристические технологии, призванные помочь в определении новых модификаций вирусов, не дают должного уровня распознавания в связи с их слабой эффективностью при работе с зашифрованными объектами. К недостаткам существующих методов обнаружения вторжений также 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.

В этой связи в составе эвристических анализаторов в настоящее время активно используются искусственные нейронные сети (ИНС) [3 – 5] и искусственные иммунные системы (ИИС) [6 – 10]. Существующие эвристические технологии, использующие ИНС и ИИС, позволяют с довольно большой долей вероятности идентифицировать широкий класс вирусов. Однако полностью эффективных способов борьбы с угрозами на сегодняшний день не существует. Поэтому появилась необходимость в разработке новых подходов к обнаружению и анализу вредоносных кодов, с помощью которых можно эффективно распознавать как старые, так и новые модификации вирусов, с минимально возможной загрузкой системы. Одним из путей решения задачи обнаружения и анализа вредоносных программ является применение мультиагентных систем (МАС) [11, 12], которые позволяют решать сложные задачи и создавать программные системы, привносящие новое качество сервиса, высокую эффективность и ряд других преимуществ.

В настоящее время МАС являются одним из наиболее динамично развивающихся направлений в области искусственного интеллекта, которое сформировалось на основе результатов исследований в области сетевых технологий, распределенных компьютерных систем и параллельных вычислений. Ключевым элементом этих систем является программный агент, способный воспринимать ситуацию, принимать решения и взаимодействовать с другими агентами. Программные агенты могут создаваться в виде специализированных классов, в которых интегрированы механизмы рассуждения на основе знаний с нейросетевыми моделями и технологией обработки информации, основанной на нечеткой логике [11]. Механизмы, которые используют программные агенты, позволяют извлекать знания из данных, т.е. находить закономерности в данных и осуществлять их обобщение, формировать

правила вывода непосредственно в процессе обучения. Кроме того, они позволяют агенту асинхронно взаимодействовать с другими агентами, а также целенаправленно действовать в условиях неопределённой и динамично изменяющейся среды.

В работе предлагается система защиты компьютера от вторжений, которая базируется на мультиагентном подходе [12]. Предлагаемая система не использует базы данных сигнатур вирусов, а, следовательно, не нужно хранить огромную базу данных на локальном компьютере, или же иметь постоянный доступ к Internet, чтобы использовать базу данных, находящуюся на удаленных серверах. Применяемый в предложенной системе метод интеллектуального анализа позволяет не только обнаруживать потенциально вредоносный код без обращения к базам данных сигнатур, но и обнаруживать ранее не известные вирусы.

Постановка задачи

Имеется множество исполняемых файлов (объектов) P_j, $j = \overline{1,F}$, которые могут содержать вредоносные коды. На этом множестве имеется разбиение на конечное число подмножеств (классов) C_k, $k = \overline{1,K}$ 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. Протоколы могут содержать общие закономерности в поведении объектов, которые представляются в виде фрагментов протоколов, являющихся признаками объектов, совокупность которых определяет описание объектов (программ) $I(C_1, C_2, ..., C_K)$.

Имеется множество программных агентов Q_i, $i = \overline{1,S}$, состоящее, возможно, из нескольких видов агентов, которые должны обнаружить и распознавать исполняемые файлы. Окружающая среда представляет собой операционную систему (ОС) компьютера, в которой взаимодействуют как исполняемые файлы P_j, $j = \overline{1,F}$ с программными агентами Q_i, $i = \overline{1,S}$, так и программные агенты между собой.

Предполагается, что для каждого исполняемого файла P_j, $j = \overline{1,F}$ существует вектор характерных поведенческих признаков $X_j = \lfloor x_{j,1}, x_{j,2}, ... x_{j,L_j} \rfloor$ из L_j элементов, который может содержать вредоносные коды. У каждого программного агента Q_i, $i = \overline{1,S}$ так же есть вектор признаков $Y_i = \lfloor y_{i,1}, y_{i,2}, ... y_{i,R_i} \rfloor$ из R_i элементов, который определяет его самостоятельные цели. При этом векторы признаков X_j и Y_i, а также особенности как программных агентов, так и исполняемых файлов могут отличаться друг от друга.

Задача детектирования и анализа вредоносных программ состоит в том, чтобы для исполняемых файлов P_j, $j = \overline{1,F}$ и набора классов C_k, $k = \overline{1,K}$ по обучающей информации $I_0(C_1, C_2, ..., C_K)$ и описаниям объектов $I(C_1, C_2, ..., C_K)$ отнести исходные программы к определенному классу C_k путем выделения и сопоставления соответствующих фрагментов программ $x_{j,l}$, $j = \overline{1,F}$, $l = \overline{1,L_j}$ с признаками программных агентов $y_{i,r}$, $i = \overline{1,S}$, $r = \overline{1,R_i}$. С этой целью необходимо разработать МАС, с помощью которой будет выполняться обнаружение и распознавание вредоносных кодов в исполняемых файлах на основе организации взаимодействий как между программными агентами и исполняемыми файлами, так и программных агентов между собой.

Модель МАС обнаружения и анализа вредоносных программ

Детектирование и анализ вредоносных программ предлагается осуществлять с помощью МАС. Общая модель МАС обнаружения и анализа вторжений (MASIDA – Multi-Agent System for Intrusion Detection and Analysis) может быть формально представлена в виде следующего кортежа [12]:

$$MASIDA = \langle El, Attr, Env, Rl, RAct, CAct, Ev \rangle, \quad (1)$$

где $El = \langle Ob \cup AgD \cup AgA \rangle$ – множество элементов системы, состоящее из объектов (исполняемых файлов) $Ob = [P_1, P_2, ... P_F]$, агентов-детекторов $AgD = [H_1, H_1, ..., H_N]$, $H_n \in Q$, $n = \overline{1,N}$ для обнаружения вредоносных программ и агентов-анализаторов $AgA = [G_1, G_1, ..., G_M]$, $G_m \in Q$, $m = \overline{1,M}$ для распознавания и анализа вредоносных программ.

Каждый исполняемый файл P_j, $j = \overline{1,F}$ характеризуется набором характерных поведенческих признаков (фрагментов программ) $X_j = \lfloor x_{j,1}, x_{j,2}, ... x_{j,L_j} \rfloor$ из L_j элементов, входящим в множество атрибутов Attr МАС: $X_j \in Attr$. Каждый агент-детектор $H_n \in Q$, $n = \overline{1,N}$ характеризуется набором атрибутов-признаков $V_n = \lfloor v_{n,1}, v_{n,2}, ... v_{n,Z_n} \rfloor$ из Z_n элементов, $Z_n \subset R$, $V \subset Y$, $V_n \in Attr$. Каждый агент-анализатор $G_m \in Q$, $m = \overline{1,M}$ характеризуется набором атрибутов-признаков $W_m = \lfloor w_{m,1}, w_{m,2}, ... w_{m,U_m} \rfloor$ из U_m элементов, $W \subset Y$, $W_n \in Attr$.

Элементы системы функционируют в окружающей среде Env, которая представляет собой ОС компьютера, и находятся в определенных отношениях Rl, которые позволяют им взаимодействовать друг с другом, обладают возможностью выполнять реактивные действия $RAct$ над объектами и коммуникативные действия $CAct$ между агентами для достижения общей цели, изменяя свои атрибуты в процессе эволюции Ev.

МАС обнаружения и анализа вредоносных программ может быть представлена в виде следующей структуры (рис. 1).

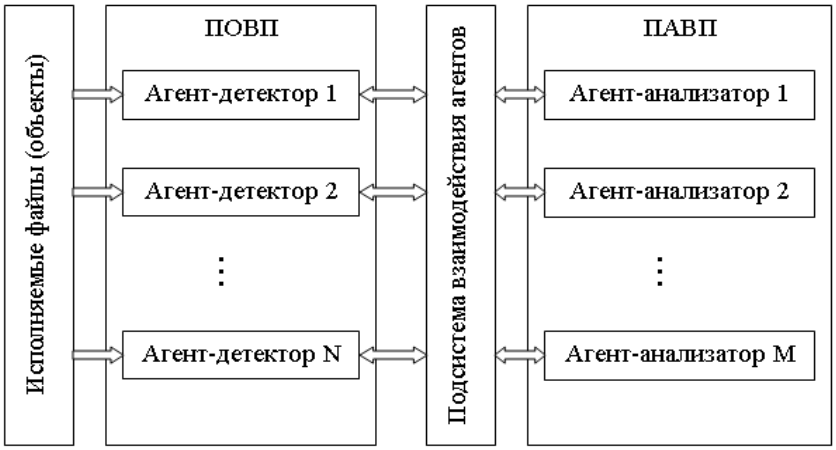

Рис. 1. Структура МАС обнаружения и анализа вредоносных программ

В предлагаемой структуре МАС можно выделить две основных подсистемы:

1) подсистему обнаружения вредоносных программ (ПОВП);

2) подсистему анализа вредоносных программ (ПАВП).

Обе подсистемы состоят из набора агентов, каждый из которых работает независимо друг от друга и решает общую задачу, возложенную на систему. Выделена также отдельная подсистема, отвечающая за взаимодействие агентов в системе.

ПОВП обеспечивает сбор данных, которые используются для анализа ситуации, и представляет собой совокупность программных функций и алгоритмов, обеспечивающих ПАВП данными для анализа, в качестве которых могут выступать байтовый код файла, текстовые строки внутри файла, API Log, единичное действие программы в рамках ОС или целая цепочка таких действий и др.

В данной системе ПОВП представлена набором агентов-детекторов H_n, $n=\overline{1,N}$, которые занимаются мониторингом ОС. Агент-детектор является программным агентом, и с технической точки зрения представляет собой драйвер, работающий в режиме ядра ОС и осуществляющий путем сопоставления атрибутов-признаков $x_{j,l}$, $j=\overline{1,F}$, $l=\overline{1,L}$ исполняемой программы с атрибутами-признаками $v_{n,z}$, $n=\overline{1,N}$, $z=\overline{1,Z}$ агента-детектора низкоуровневый перехват стандартных API-функций, вызов которых может привести к потенциальной угрозе безопасности компьютера.

Драйвер, перехвативший функцию, изменяющую реестр, формирует пакет, в котором содержится необходимая для дальнейшего анализа информация о процессе, вызвавшем данную функцию, и параметрах реестра, которые он запрашивал на изменение.

Вредоносная программа – это, с одной стороны, файл с определенным содержимым, с другой – совокупность действий, производимых в ОС, с третьей – совокупность конечных эффектов в ОС. Поэтому и идентификация программы может быть произведена на разных уровнях: по цепочкам байт, по действиям, по влиянию на ОС и т.д. В соответствии с этим выделяются следующие способы сбора данных для обнаружения вредоносных программ, каждый из которых реализует отдельный агент-детектор и с помощью которых можно выявить большинство вирусов:

– монитор списка автозагрузки;

– монитор загрузочного раздела;

– монитор раздела HKLM\SOFTWARE\ Microsoft\WindowsNT\CurrentVersion\Winlogon реестра;

– монитор системного файла C:\windows\ system32\drivers\etc\hosts.

На один вирус может сработать больше одного агента-детектора, следовательно, необходимо учитывать и комбинации срабатываемых детекторов. Например, если вирус пытается прописать себя в список автозагрузки, ничто не мешает ему прописать себя в C:\windows\system32\drivers\etc\hosts адрес сайта, откуда он может сам закачаться в случае, если его обнаружат и удалят из списка автозагрузки. Так же существуют комбинации детекторов, которые никогда не сработают, например, если вирус пытается прописаться в загрузочный раздел, значит он запустится до загрузки ОС, следовательно, ему

незачем пытаться прописать себя еще и в список автозагрузки.

ПАВП, состоящая из агентов-анализаторов G_m, $m=\overline{1,M}$, – это совокупность модулей, отвечающих за принятие решения. Это набор алгоритмов, каждый из которых анализирует имеющиеся в его распоряжении данные и выносит о них суждение, в соответствии с которым антивирус (либо другое защитное ПО) предпринимает установленные его политикой безопасности реактивные действия $RAct$: оповещает пользователя, запрашивает у него дальнейшие указания, помещает файл в карантин, блокирует несанкционированное действие программы и др.

Синхронизация работы агентов-детекторов с другими подсистемами МАС осуществляется путем посылки уведомлений о событии в подсистему взаимодействия агентов (ПВА), которая использует «доску объявлений» для выполнения коммуникативных действий $CAct$ между агентами. Сформированный пакет с данными драйвер помещает в буфер, адрес которого размещается на «доске объявлений», откуда эту информацию могут получить и использовать далее агенты-анализаторы. «Доска объявлений» запрашивает данные из участка памяти в пространстве ядра ОС и копирует в свое адресное пространство для предоставления этой информации агентам-анализаторам.

Таким образом, в предлагаемой МАС обнаружения и анализа вредоносных программ используются два вида агентов: агент-детектор и агент-анализатор. Основная задача агента-детектора – мониторинг основных уязвимостей ОС, и в случае обнаружении аномальной активности – размещение информации о процессе, инициирующем эту аномальную активность, на «доске объявлений».

Агент-анализатор является интеллектуальным программным агентом, который использует эвристический анализ.

Задача агента-анализатора – исследовать процессы, размещенные на «доске объявлений», и принятие решения, какие из процессов являются потенциальными вирусами и к какому классу вредоносных программ они относятся. Информация о вхождении исполняемого файла в какой-либо класс представляется в виде вектора

$$I(P_j)=\{I_1(P_j),I_2(P_j),...,I_K(P_j)\},$$

где $I_k(P_j)$, который несет информацию о принадлежности или не принадлежности объекта (программы) P_j, $j=\overline{1,F}$ к классу C_k, $k=\overline{1,K}$:

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

Для эффективного распознавания вредоносных программ и при большом количестве агентов необходима достаточно сложная аналитическая система – наподобие экспертной системы, в которой могут использоваться различные технологии. Поэтому ПАВП можно представить в виде эвристического анализатора, который может быть реализован в виде ИНС [3, 5] или ИИС [10].

Подсистема МАС обнаружения и анализа вредоносных программ, отвечающая за взаимодействие агентов, – ПВА, также является интеллектуальной, т.к. обладает соответствующими свойствами: сама принимает решения о том, какой из агентов-анализаторов наиболее точно сможет определить вирус, и обладает базой знаний. В качестве базы знаний используются соответствия атрибутов-признаков $v_{n,z}$, $n=\overline{1,N}$, $z=\overline{1,Z}$ агентов-детекторов атрибутам-признакам $w_{m,u}$, $m=\overline{1,M}$, $u=\overline{1,U}$ агентов-анализаторов, а также правила выбора одного или нескольких агентов-анализаторов для принятия решения.

Каждый агент в системе имеет свой уникальный идентификатор. Таким образом, можно однозначно идентифицировать, какой из агентов-детекторов обнаружил аномальную активность и готов предоставить соответствующую для анализа информацию. В зависимости от того, какие агенты-детекторы сработали исходя из заложенной базы знаний, принимается решение о выборе наиболее подходящего агента-анализатора (или агентов-анализаторов).

В случае, если комбинации из срабатываемых агентов-детекторов нет в базе знаний, выбираются несколько агентов-анализаторов, покрывающих все агенты-детекторы. Данные агенты-анализаторы выполняют анализ по очереди с целью наименьшей загрузки системы. Более приоритетным считается агент-анализатор, вероятность ложных результатов которого наименьшая Результаты каждого из агентов-анализаторов анализируются и в зависимости от результата база знаний корректируется или пополняется.

Экспериментальные исследования

Экспериментальные исследования проводились с использованием специального программного обеспечения, которое эмулирует работу центрального процессора, API функций ОС и ее внутренних структур, и подходит для задачи мониторинга поведения программ и сбора необходимых данных. Для проведения экспериментальных исследований использовался специализированный набор инструментов: разработанные утилиты Threader и Matcher, которые предназначены для преобразования и исследования информации.

Для сравнительного анализа предлагаемой МАС с другими моделями (ИНС и ИИС) было выбрано 20 вредоносных программ, принадлежащих разным классам вирусов (загрузочные, файловые, сетевые и макровирусы), которые однозначно были распознаны при использовании всех подходов.

На этапе получения и обработки данных производился запуск вредоносных программ на эмуляторе и получение протоколов их работы. Была выполнена предварительная обработка полученных протоколов работы вредоносных программ. 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 (ИНС, ИИС и МАС) приведены в табл. 1.

Таблица 1

Потребление системных ресурсов

Класс вирусов	Наименование вируса	Время анализа программ, с			Загрузка ЦП, %			Загрузка ОП, %		
		ИНС	ИИС	МАС	ИНС	ИИС	МАС	ИНС	ИИС	МАС
Загрузочные	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
Файловые	Trojan-PSW.Win32.Tepfer.kcoi	14	9	7	24	22%	22%	30	32	32
	Trojan-PSW.Win32.Tepfer.kcta	7	9	5	27	28	20	34	36	30
	Trojan.Win32.KillFiles.brrq	10	12	8	25	25	22	26	25	34
	Trojan.Win32.Bublik.axmx	7	10	7	33	31	26	32	33	29
	Trojan-Ransom.Win32.Foreign.dbdo	8	9	6	28	26	28	40	36	40
Сетевые	Net-Worm.Win32.Kolab.body	8	10	7	28	32	32	35	44	44
	Worm.Win32.Vobfus.duxh	7	8	5	24	24	20	27	38	35
	P2P-Worm.Win32.Palevo.hbih	7	8	6	22	27	24	31	33	40
	Worm.Win32.Cridex.pjd	8	7	4	33	22	36	36	46	32
	Worm.Win32.WBNA.ahzb	9	11	5	37	35	38	40	54	43
Макро-вирусы	Trojan-Ransom.Win32.Foreign.cyyy	12	11	5	21	20	16	15	13	16
	Trojan.Win32.Jorik.ZAccess.pmf	13	14	7	24	22	23	15	16	14
	Trojan-Spy.Win32.Zbot.lwos	12	10	5	16	28	14	12	35	21
	Trojan-PSW.Win32.Tepfer.ljni	12	6	7	27	19	28	21	24	24
	Backdoor.Win32.Ruskill.udm	13	9	6	24	26	24	20	28	20

Из приведенных результатов можно сделать вывод о том, что нейросетевой подход показал лучшие результаты, но показатели мультиагентной системы имеют очень незначительные отличия. Результаты сравнительного анализа показали, что МАС определяет вредоносные программы с меньшим потреблением системных ресурсов в сравнении с нейросетевым и иммунным подходами.

В качестве новых программ для распознавания МАС были представлены две вредоносные программы и две не вредоносные программы: 1) Trojan.OlympicGames; 2) Win32.Worm.Prolaco.S; 3) Mirabilis ICQ и 4) Opera Web Browser. Представленные программы были распознаны верно: 1-я и 2-я программы оказались вредоносными, а 3-я и 4-я – не вредоносными.

Таким образом, разработанная МАС обнаружения и анализа вредоносных программ успешно решает возложенную на нее задачу и способна распознать новые модификации вирусов.

Выводы

Рассмотрено решение актуальной задачи выявления и распознавания как существующих, так и новых модификаций вредоносных программ на основе использования мультиагентного подхода.

Предложены модель и структура МАС обнаружения и анализа вредоносных программ, основными компонентами которой являются ПОВП, представленная набором агентов-детекторов и используемая для мониторинга ОС и сбора данных, а также ПАВП, представленная набором агентов-анализаторов и используемая для анализа и принятия решения о принадлежности или не принадлежности рассматриваемой программы к семейству вредоносных программ. Синхронизация работы ПОВП с ПАВП в МАС осуществляется с помощью ПВА, которая использует «доску объявлений» для выполнения коммуникативных действий между агентами.

Проведены экспериментальные исследования на примере вредоносных программ, принадлежащих разным классам вирусов. Результаты сравнительного анализа показали, что МАС определяет вредоносные программы с меньшим потреблением системных ресурсов в сравнении с нейросетевым и иммунным подходами, успешно решает возложенную на нее задачу и способна распознать новые модификации вирусов.

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. Гаврилов А.В. Применение постоянно модифицирующихся нейронных сетей для защиты программного обеспечения // А.В. Гаврилов / Нейрокомпьютеры6 разработка, применение. – 2008. – № 1-2. – С. 90-101.

4. 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.

5. 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.

6. Безобразов, С.В. Искусственные иммунные системы для защиты информации: сравнительный анализ методов негативной и позитивной селекций детекторов // Инженерный вестник. – 2006. – № 1(21). – С. 76-82.

7. Bezobrazov, S. Artificial immune system approach for malware detection: neural networks applying for immune detectors construction / S. Bezobrazov, V. Golovko // Inernational journal of «Computing». – 2008. – Vol. 7, no. 2. – P. 44-50.

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. Zekri M. Artificial Immune System for Intrusion Detection / M. Zekri, L. Souici-Meslati // Evolutionary Computation. – 2011. – V. 13, № 2. – Pp. 145-153.

10. 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.

11. 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. – Ч. 1. – С. 190-201.

12. Alkhateeb F. Multi-Agent Systems – Modeling, Interactions, Simulations and Case Studies [Текст] / F. Alkhateeb, E. Al Maghayreh, I. Abu Doush // Published by InTech, Rijeka, Croatia. – 2011. – 502 p.

Поступила в редколлегию 10.12.2014

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

ВИЯВЛЕННЯ ТА АНАЛІЗ ШКІДЛИВИХ ПРОГРАМ З ВИКОРИСТАННЯМ МУЛЬТИАГЕНТНОГО ПІДХОДУ

М.М. Корабльов, М.В. Кушнарьов, О.Г. Лебедєв

Запропоновано модель і структура мультиагентної системи виявлення та аналізу шкідливих програм, основними компонентами якої є агенти двох видів: агенти-детектори та агенти-аналізатори. Завдання агента-детектора – моніторинг основних вразливостей операційної системи і збір даних, завдання агента-аналізатора - дослідити процеси і прийняття рішення, які з них є потенційними вірусами і до якого класу шкідливих програм відносяться. Проведено експериментальні дослідження, що показують ефективність запропонованого підходу.

Ключові слова: *мультиагентна система, шкідлива програма, агент-детектор, агент-аналізатор, взаємодія агентів.*

DETECTION AND ANALYSIS OF MALWARE USING MULTI-AGENT APPROACH

N.M. Korablyov, M.V. Kushnaryov, O.G. Lebedev

The model and the structure of multi-agent systems of detecting and analyzing malware were proposed. The main components are two types of agents: agents-detectors and agents-analyzers. The task of agent-detector – monitoring key vulnerabilities of the operating system and data collection. The task of agent-analyzer – explore processes and make decision, which of them are potential viruses and to which class of malware are. Experimental researches were provided and they show the effectiveness of the proposed approach.

Keywords: *multi-agent system, malware, agent-detector, agent-analyzer, interaction of agents.*