S. D. Kulik, K. I. Tkachenko

Software Robots and Information Security

Keywords: *information security, method, computer virus, bot, internet-bot, www-bot, web-bot*

This article is about special methods of information security. These methods are a part of the tools of ensuring the information security. In this paper the problem of threats and protection against software robots discussed. The results of the researches were successfully protected by the various security documents of ROSPATENT.

С. Д. Кулик, С. И. Ткаченко

ПРОГРАММНЫЕ РОБОТЫ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Введение

Современные информационные технологии позволяют по-новому взглянуть на актуальную проблему информационной безопасности. Эффективная работа информационных систем часто зависит от различных специальных программных средств, таких как *программные роботы*, при этом сама работа информационной системы может быть нарушена действием других специфических программных объектов, таких как *вредоносные программы* или *компьютерные вирусы* [1]. Целью работы является анализ проблемы, связанной с разработкой метода (методики или концепции) обеспечения информационной безопасности с учетом таких объектов, как программные роботы. Подобные объекты изучаются экспертами-криминалистами. Рассмотрим, опираясь на работы [2–28], подробнее эту проблему.

В действующем Уголовном кодексе Российской Федерации (УК РФ) вредоносным компьютерным вирусам отведена целая глава, 28-я. Так, преступление, предусмотренное статьей 273 УК РФ (*Создание, использование и распространение вредоносных программ для ЭВМ*), наиболее опасное из содержащихся в главе 28 УК РФ, что выразилось в наиболее суровом наказании за него [3, 4]. Отметим, что в национальных законодательствах государств различных регионов мира, например Великобритании, Нидерландов, Италии и скандинавских стран, имеются аналогичные статьи за создание и распространение вирусов и вредоносных программ. В связи с этим следует быть осторожным и вести научные исследования так, чтобы не попасть под действие статьи 273 УК РФ. Некоторые необходимые рекомендации можно найти в работах [3, 4, 5].

Криминалистические средства [8, 17, 19], такие как информационные системы, в том числе автоматизированные фактографические информационно-поисковые системы (АФИПС) [4, 8, 10–13], могут быть реализованы с помощью сетевых средств и иметь удаленные средства доступа к важным фактографическим данным. Основу подсистемы поиска АФИПС составляет именно специализированный поисковый *программный робот*. Эффективность поискового робота АФИПС достаточно подробно исследована в работе [2]. На практике в силу ряда обстоятельств этот программный робот может быть создан как вредоносная программа.

Таким образом, актуальной является проблема разработки метода (методики) или хотя бы концепции обеспечения информационной безопасности, связанной с программными роботами.

Известно, что основу компьютерного вируса, как правило, составляет так называемый саморазмножающийся механизм [1]. Был выполнен анализ отечественного программного обеспечения. Разработать программу, реализующую саморазмножающийся механизм, непросто, но программист-профессионал сможет это сделать, например, на языках программирования FORTH [1], C++ или S+S [5]. Отметим, что при выполнении некоторых ограничений

имеется возможность определения авторства (идентификация преступника) по исходному тексту программы. Для установления автора текста назначается автороведческая экспертиза. Если имеется только исполняемый код программы и отсутствует ее исходный текст, то в некоторых случаях с помощью специальных программных средств удается либо полностью восстановить исходный код программы, либо только частично, например без строк комментариев.

Остановимся кратко на возможном подходе к построению методики определения вредоносных программ по исходному тексту на заданном языке программирования [3–7].

Предварительные исследования показали, что для *неизвестной программы* (НП) не всегда удается достоверно принять решение о принадлежности ее к классу вредоносных программ [3–7]. Это может быть, например, связано с неполнотой информации о *языке программирования* (ЯП), на котором создана *неизвестная программа* (НП), или с невозможностью выявить сам набор *вредоносных операторов*, или с отсутствием достоверных данных о *технической среде*, в которой выполняется *неизвестная программа*. Так, например, дополнительные исследования показали, что для интерпретаторов в некоторых случаях существует компьютерный вирус, который может превратиться из безвредной программы во вредоносную программу, искажающую, например, *фактографическую базу данных* (ФБД) [3] в автоматизированной фактографической АФИПС при неконтролируемой, несанкционированной смене языка программирования интерпретирующей системой. Существует достаточно много интерпретирующих систем, например система для языка Python [21]. Для проведения эксперимента были разработан язык программирования S+S [4] и интерпретатор для него, затем созданы и использованы две невредоносные программы. Эти две разные программы (X-60 и MIF-X-78) на практике имели полностью идентичные исходные тексты (листинги) [3, 4, 5].

В итоге получается, что любую программу можно отнести как к вредоносной, так и к невредоносной в зависимости от языка программирования. Отметим, что *компьютерный вирус* и *вредоносная программа* (например, разрушающая ФБД в АФИПС) это не одно и то же. Например, такие компьютерные вирусы, как MIF-X-01, MIF-X-02, MIF-X-03 или MIF-X-78, являются **невредоносными** программами [3, 4, 5, 9]. Имеется тесная связь между *компьютерным вирусом* и *мобильным поисковым роботом* (агентом). Поисковый робот может быть разработан как компьютерный вирус или, по крайней мере, обладать многими его признаками. Исследования компьютерных вирусов и вредоносных программ позволяют разрабатывать необходимые и эффективные средства защиты от них (см., например: [20]).

Нейросетевые технологии [14–16, 18] находят все большее применение на практике, например в программном обеспечении. В основе работы поискового робота могут быть эти нейросетевые средства. Решающее правило о принадлежности неизвестной программы к классу вредоносных программ может строиться на подходящем нейросетевом алгоритме.

Для определения принадлежности неизвестной программы к классу вредоносных программ необходимо разработать методику определения вредоносных программ по исходному тексту. Для ее реализации предлагается следующая последовательность действий при исследовании неизвестной программы [5]:

1) получить или восстановить исходный текст НП;

2) определить ЯП и техническую среду, необходимые для выполнения НП;

3) выявить для ЯП набор вредоносных операторов, приводящих к свойству вредоносности НП;

4) выяснить в НП наличие средств мутации, самомодификации;

5) определить условия (события), при которых вредоносные операторы выполняются в НП;

6) найти набор квалифицирующих признаков вредоносности для НП;

7) отнести НП к классу и типу вредоносных программ и сделать окончательный вывод о НП.

Приведенный обобщенный подход можно использовать в качестве основы для построения требуемой методики. Далее необходимо обучить экспертов применять эту методику. Исследования показывают, что для этого необходимо иметь подходящий генератор учебных вредоносных программ (возможно применение модифицированного генератора [17]). После того как эта методика будет создана, может быть разработан такой подходящий генератор текстов вредоносных программ для оценки качества подготовки эксперта по анализу неизвестных и подозрительных программ.

Перейдем к проблеме программных роботов.

Информационная безопасность, согласно определению, приведенному в законе РФ, — состояние защищенности информационной среды общества, обеспечивающее ее формирование, использование и развитие в интересах граждан, организаций, государства.

Принято считать (см. также: [24, 25]), что информационная безопасность имеет три основные составляющие [28]:

1) конфиденциальность — защита чувствительной информации от несанкционированного доступа;

2) целостность — защита точности и полноты информации и программного обеспечения;

3) доступность — обеспечение доступности информации и основных услуг для пользователя в нужное для него время.

В качестве злоумышленника в работе рассматривается субъект, оказывающий на информационный процесс воздействие путем создания и использования роботов с целью вызвать его отклонение от условий нормального протекания. Примем, как часто принимается в криптографии, что в распоряжении злоумышленника имеются все необходимые для выполнения его задачи технические и информационные средства, созданные на данный момент.

Вместо терминов *паук*, *краулер*, *бот*, *робот поисковика* [23] и т. п. названий программ будем далее использовать единый термин *программа-робот*, или просто *робот*.

Один из авторов данной статьи, С. И. Ткаченко, ранее предложил следующую концепцию обеспечения информационной безопасности, связанную с программными роботами. В основе этой концепции лежит метод обеспечения информационной безопасности интернет-ресурсов, обеспечивающих взаимодействие по протоколу HTTP и использующих язык разметки HTML, для передачи полезной текстовой информации.

Будем учитывать следующие угрозы безопасности:

U1 — загрузка на сервер при помощи программ-роботов заведомо ложной информации или информации рекламного характера в автоматизированном режиме, посредством внесения комментариев или отзывов на форумах и в интернет-магазинах. Подобные действия напрямую отражаются на таких составляющих информационной безопасности, как целостность и достоверность информации.

U2 — покупка роботами в автоматизированном режиме наиболее привлекательных мест на концертах, авиарейсах и т. п. с целью последующей перепродажи, что напрямую влияет на доступность информации и основных услуг для пользователя в нужное для него время, это также является составляющей информационной безопасности.

U3 — компрометирование учетных записей пользователей на различных интернет-ресурсах, включая, но не ограничиваясь различными личными кабинетами на сайтах магазинов, банков и т. д.

U4 — чтение, при необходимости сохранение и тиражирование в автоматизированном режиме, информации о различных ценах, котировках, событиях, что снижает привлекательность информации для пользователя на ресурсе-первоисточнике. Также данная угроза возможна при реализации угрозы U2.

U5 — фарминг (англ. farming — сельское хозяйство), или фарм (англ. farm — фермерство). Под термином «фарминг» следует понимать постоянно повторяющиеся действия злоумышленника с целью накопления определенных предметов и иных ресурсов на какой-либо игровой интернет-площадке для последующей перепродажи другим игрокам за валюты реального мира.

Задача защиты от нежелательных действий роботов возникла практически сразу, как только протокол HTTP из стека протоколов TCP/IP и язык разметки гипертекста HTML стали набирать популярность в 1998–1999 г. Соответственно, чуть позже появились средства, позволяющие различить, взаимодействует с интернет-ресурсом (HTTP-сервером) человек или программа-робот.

Необходимо выполнить анализ средств защиты от вышеперечисленных угроз.

Для выполнения комплексного анализа введем ряд показателей (критериев), по которым можно сравнить эффективность возможных методов противодействия роботам.

Делая выводы относительно различных технологий в области защиты информации и материальных ресурсов, по мнению авторов, всегда следует руководствоваться принципом разумной достаточности. Вероятность успешной атаки злоумышленника сводится практически к нулю, если стоимость получаемого в результате атаки ресурса ниже, чем стоимость совокупных затрат злоумышленника. Таким образом, оценивать эффективность метода противодействия роботам можно исходя из оценочных затрат злоумышленника на реализацию атаки. Затраты злоумышленника условно можно разделить на следующие составляющие:

- затраты на программную реализацию алгоритма работы робота;
- затраты на аппаратное обеспечение работы программы-робота.

Показатель затрат на программную реализацию можно оценить как:

$$\mathbf{P} = t_{pr} \cdot \rho_{nh},$$

где t_{pr} — выраженная в человеко-часах оценка времени, затраченного специалистом в области программирования требуемой квалификации;

ρ_{nh} — оценка средней рыночной стоимости одного человеко-часа специалиста необходимой квалификации.

Показатель затрат на аппаратное обеспечение можно вычислить по следующей формуле:

$$\mathbf{A} = t_{cpu} \cdot \rho_{cpu} + v_{mb} \cdot \rho_{mb},$$

где t_{cpu} — оценка времени работы центрального процессора в условных единицах;

ρ_{cpu} — оценка стоимости работы процессора в условных единицах;

v_{mb} — оценка объема переданной по сети информации в мегабайтах или гигабайтах;

ρ_{mb} — оценка стоимости 1 мегабайта или гигабайта.

Следует отметить, что в случае, когда трафик отдельно не оплачивается, стоимость передачи 1 единицы информации, тем не менее, не равна нулю, так как сегодня передача по сети информации — все еще самая медленная временная составляющая работы любой программы. Таким образом, чем больше объем передаваемой информации, тем выше значение первого слагаемого в формуле. В качестве условной единицы времени работы центрального процессора можно принять количество тактов, затраченных центральным процессором, но это пока сложнооцениваемая величина, поэтому можно принять просто время выполнения программы в секундах.

Следующим критерием для сравнения методов противодействия роботам может служить стоимость полученной информации (**I**), которая напрямую зависит от времени ее получения. Например, злоумышленнику требуется загрузить весь перечень цен с определенного ресурса для анализа и поиска оптимальной по определенному условию цены. Предположим, на это ему потребуется времени t_0, а период обновления цен на ресурсе равен t_1. Тогда если $t_1 < t_0$, ценность полученной злоумышленником информации близка к нулю. Другой показательный пример:

злоумышленник создал свой ресурс для извлечения прибыли от транслируемой рекламы. Ресурс транслирует цены другого ресурса с помощью робота. В случае, если время запроса к стороннему ресурсу достаточно велико, то конечный пользователь может уйти, не дождавшись ответа, и больше не вернуться. Таким образом, злоумышленник не достигнет своей цели.

Последний критерий, который будет рассмотрен в работе, — это доступность ресурса для пользователя после внедрения средств защиты. Этот критерий можно выразить с помощью временного интервала **T**. Если принять за t_a время, затраченное пользователем на выполнение задачи без каких-либо средств противодействия роботам, а за t_b — время, затраченное пользователем с включенными средствами защиты, тогда $\mathbf{T} = t_b - t_a$.

Анализ существующих средств защиты

Наиболее популярным средством защиты является **CAPTCHA** (от англ. Completely Automated Public Turing test to tell Computers and Humans Apart — полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей) [23]. Этот компьютерный тест используется для того, чтобы определить, кем является пользователь системы: человеком или компьютером. В тесте реализован обратный тест Тьюринга, в котором машину и человека поменяли местами.

В основу теста заложена идея, основанная на том, что существует набор задач, которые пользователь-человек может с легкостью решить, но которые существенно сложнее решить компьютеру.

При применении метода вначале пользователю показывался набор букв или слов, практически не искаженных, также отсутствовало смещение букв друг относительно друга. Изображение представляло собой комбинацию изображений букв алфавита, использующегося в средстве защиты. Такой подход вполне оправдывал ожидания разработчиков, так как не требовал значительных ресурсов сервера и распространение программ распознавания текста было незначительным. С широким распространением средств сканирования изображений появилось множество программ распознавания текста на изображении, и использование незашумленного текста, предлагаемого пользователю, стало практически неэффективным. Сегодня средство защиты развивается в направлении, когда текст все более не узнаваем (сильно зашумлен) не только для роботов, но и для человека, особенно с нестопроцентным зрением. Таким образом, пользователь тратит все больше времени на распознавание текста на изображении. В связи с этим использование CAPTCHA для защиты данных, где уполномоченный пользователь может взаимодействовать с множеством ресурсов, становится практически не приемлемым. Примером таких ресурсов могут быть сайты, где представлены цены на различные товары или услуги.

Взлом защиты, основанной на CAPTCHA, сегодня ведется по двум направлениям: использование математических методов для обработки изображений с использованием обучаемой искусственной нейронной сети. Эффективность данного метода порядка 5—10 % корректно распознанных изображений к общему числу изображений (в [22] сообщается о 6%). Другое направление имеет значительно большую эффективность, порядка 80—90 %, при незначительном увеличении затрат — порядка 2 копеек за изображение, что вполне приемлемо в случаях, когда требуется, например, распространить рекламные сообщения на достаточно большом числе интернет-площадок. Суть метода заключается в использовании роботом простых API (Application Program Interface) интернет-ресурсов, которые для распознавания используют недорогую рабочую силу (пользователей) в развивающихся странах (см. рис. 1). Время распознавания изображения колеблется от 5 до 20 секунд, что вполне приемлемо для реализации большинства угроз. Ориентировочное распределение работников по странам представлено на рис. 1.

Встречается другая разновидность CAPTCHA, основанная на базе данных простых вопросов, на которые пользователь должен ответить. Характерными примерами в этом случае

являются следующие предложения: «Введите сумму двух чисел 3 + 34» или «Напишите фамилию первого человека в космосе».

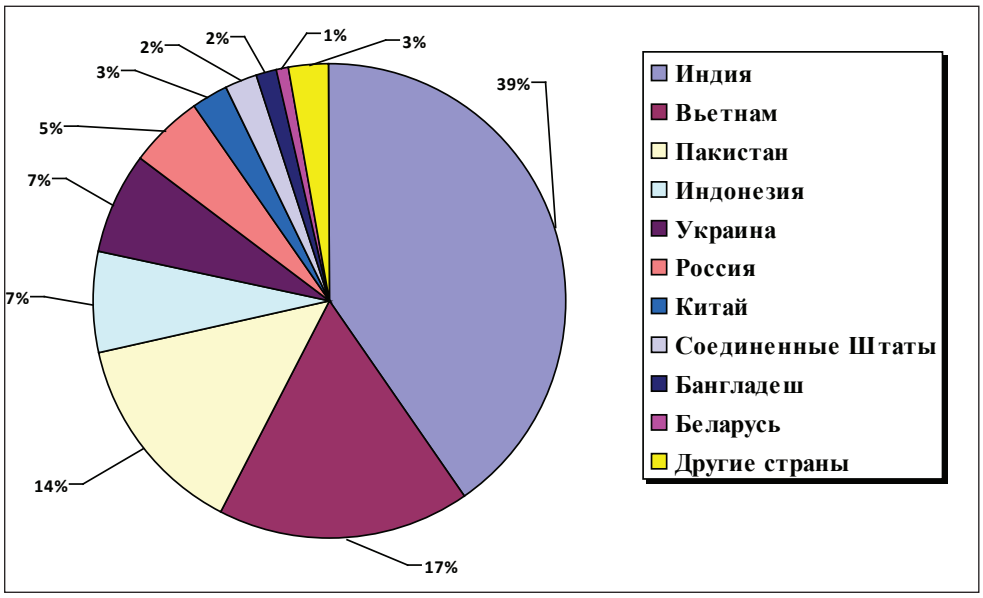

Рис. 1. Пример распределения работников по странам (данные заимствованы из [27])

Так как количество шаблонов вопросов в подавляющем большинстве случаев не превышает 100–150 штук, обход такой защиты заключается в простом выявлении базы вопросом, где не требуется серьезных навыков в области программирования. Существуют и другие разновидности CAPTCHA, встречающиеся достаточно редко.

Следующий метод защиты от роботов основан на внедрении в код HTML-страницы запросов к серверу, не несущих смысловую нагрузку с точки зрения прикладного применения интернет-ресурса. Обычно эти запросы основаны на уникальном адресе, который формируется динамически и меняется при каждом запросе, адрес, как правило, связан с индивидуальными параметрами клиентского рабочего места, например IP-адресом, также адрес зависит от данных, сохраненных браузером на стороне клиента (HTTP cookie). HTTP cookie (см., например: [26] и др.) — небольшой фрагмент данных, хранимый на компьютере пользователя и использующийся при выполнении HTTP-запросов.

Для автоматизации взаимодействия программы робота с ресурсом, защищенным подобным методом, необходимо отследить все запросы, генерируемые браузером — легитимной программой-клиентом интернет-ресурса. На основе полученной статистической информации выявить запросы, которые не несут смысловой нагрузки, например запрос, который возвращает пустое изображение, верное с точки зрения графического интерпретатора, но ничего не отображающее на мониторе компьютера. Также следует обратить внимание на запросы, адрес которых меняется при каждом обращении к одному и тому же разделу ресурса. Все вышеперечисленное позволяет в сжатые сроки выявить служебные запросы, на которых строится защита, для требуемой манипуляции данными ресурса со стопроцентной эффективностью.

Существуют вариации описанного выше метода, связанные с большими числами в JavaScript и PHP. Суть метода заключается в формировании с помощью математических преобразований некоторой строки, содержащей редко повторяющиеся данные на стороне клиента, и в дальнейшем использовании указанной строки в качестве HTTP cookie. Метод основан на различной точности вычислений с плавающей точкой в интерпретаторах языков PHP и JavaScript. Для преодоления данного

вида защиты достаточно использовать либо специальные расширения языка PHP, либо язык более высокого уровня, например C/C++. Перечень математических преобразований можно легко найти в исходном коде JavaScript-функций, так как они загружаются на сторону клиента в исходном виде.

Авторский метод защиты веб-ресурса

Суть метода, предложенного авторами, заключается в создании такой ситуации, в которой злоумышленнику пришлось бы моделировать максимально полно действия легитимного пользователя посредством веб-браузера.

Сегодня очевидно, что практически не существует средств стопроцентной защиты от действий роботов, так как любое действие легитимного пользователя, выполненное с помощью ЭВМ, теоретически можно повторить, эмулируя аналогичные управляющие сигналы. Поэтому авторами была поставлена задача выработки методики, при которой достигался бы лучший баланс экспертных оценок по описанным выше критериям среди определенных авторами существующих методик защиты.

Из описанных выше показателей (критериев) следует, что для решения указанной задачи требуется:

- максимально увеличить время, затраченное роботом на выполнение запроса к серверу;
- увеличить количество элементов системы защиты, которые программист может не учесть при реализации программы робота, тем самым предотвратив успешное получение роботом информации.

Рассмотрим следующий типичный пример последовательности шагов взаимодействия пользователя с некоторым веб-ресурсом. Предположим, требуется защитить веб-ресурс, состоящий из двух HTML-страниц. На первой странице расположена форма ввода, состоящая из нескольких полей и кнопки «Отправить». Переход на вторую страницу осуществляется при нажатии на кнопку. Вторая страница содержит информацию-ответ на параметризованный запрос пользователя на первой странице, это могут быть цены на услуги или товары.

Пользователь выполняет следующие шаги:

1. Запрос первой страницы путем нажатия на ссылку на внешнем, по отношению к данному, ресурсе или путем ввода адреса страницы в строке браузера.

2. Браузер загружает страницу и все дополнительные ресурсы, связанные с данной страницей. Такими ресурсами обычно являются файлы изображений, файлы CSS-стилей, файлы — программные модули на языке JavaScript, Flash и других языках. Следует отметить, что, например, при загрузке с сервера приведенной ниже, на рис. 2, простой формы загружается порядка десяти файлов. Обычно обработка страницы с формой требует загрузки 30—60 файлов.

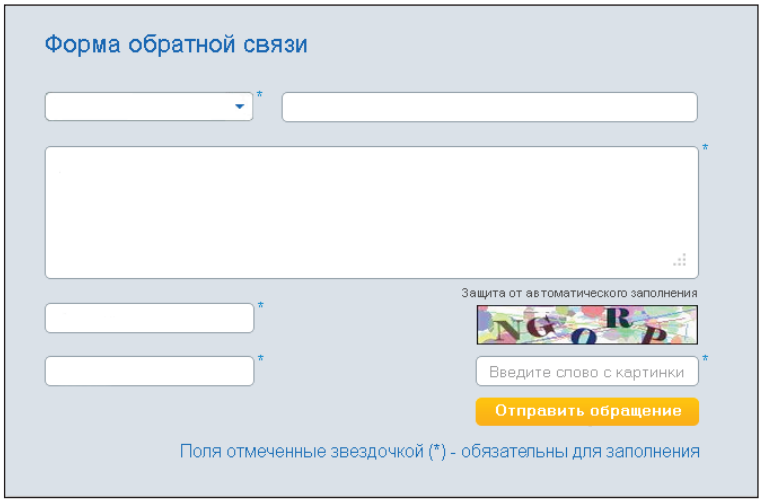

Рис. 2. Пример HTML-страницы

3. Пользователь производит ввод данных, используя различные управляющие устройства: мышь, клавиатуру, сенсорный экран и т. п.

4. Пользователь ожидает завершения процесса обработки сервером введенных данных и отображения новой страницы.

Для обхода роботом защиты описанной выше страницы достаточно:

- загрузить изображение слова единственного элемента защиты;
- передать изображение серверу, занимающемуся распознаванием изображений, и получить ответ;
- сформировать требуемые для передачи серверу-жертве данные.

Роботу потребуется около 5—15 секунд для успешного выполнения задачи.

Для разработки подобного робота злоумышленнику, согласно экспертным оценкам, потребуется примерно 3—4 часа. Ресурсов для реализации атаки также потребуется минимальное количество.

Авторский метод защиты основывается на следующих исходных данных:

- О каждом подключенном клиенте можно получить следующую, но не ограничиваясь данной, информацию: Ip — например 127.0.0.4 / браузер — Firefox 27.0 / операционная система — Windows x.x / поддержка cookies — да / поддержка https — да / поддержка javascript — да / flash — 12.0 / Разрешение экрана — 2560Ч1440Ч24.
- Cookies передаются при HTTP GET/POST запросе любого файла, а не только файлов, содержащих HTTP-код.
- С помощью JavaScript и событийной DOM-модели веб-браузера можно получить исчерпывающую информацию о манипуляциях мышью, клавиатурой, пером и т. п., касающуюся легитимного пользователя, на странице веб-ресурса.

Рассмотрим, на каких основных составляющих (компонентах) строится предлагаемая система защиты страницы. Можно выделить следующие компоненты:

- Факт загрузки всех файлов, ассоциированных со страницей. С помощью специального модуля расширения веб-сервера анализируются все запросы, поступающие на сервер, до дальнейшей передачи другим модулям обработки. При запросе основного файла страницы, содержащего HTML-код, выставляется флаг о начале обработки комплексного запроса, флаг снимается, когда загружаются все элементы страницы. Если какой-либо элемент страницы не загружен до запроса следующей страницы, система защиты принимает решение о том, что страницу загружал робот.
- Координаты и временные задержки между действиями пользователя также являются элементами системы защиты. При разработке страницы настраивается модуль, отслеживающий все запросы к серверу. В конфигурационный файл записывается информация об относительном расположении элементов на экране пользователя, причем, если элементов много, а монитор имеет низкое разрешение, модуль сам учитывает события пролистывания содержимого страницы. Также определяются временные задержки, которые требуются легитимному пользователю для заполнения формы. Если сервер получает запрос с данными формы, но при этом не получал запросов с необходимыми данными о заполнении полей формы и перемещениях мыши, система защиты также принимает решение о том, что страницу загружал робот. Системы защиты не пропустят запрос и в том случае, если не были выдержаны временные задержки.
- Третьим элементом защиты является комбинация изображения двух-трех различных предметов, животных, букв, цифр и предложения, содержащего просьбу нажать на одно из приведенных изображений предметов. Изображения предметов не зашумляются, как это делают обычно, но при этом поворачиваются вокруг собственной оси, также одному и тому же названию предмета в базе изображений сопоставляются несколько вариантов изображений. Если переданные координаты не совпадают с изображением, система защиты не пропустит запрос. Координаты изображения в целом меняются, причем с помощью относительных инструкций языка HTML, а не позиционных.

Выполним анализ предложенного метода. Программа-робот не сможет выполнять запросы быстрее, чем выполняет запросы легитимный пользователь. Действительно, для прохождения системы защиты роботу потребуется загрузить все составляющие страницы, содержащей форму, а не только выполнить параметризованный запрос к странице с ответом. Более того, роботу потребуется определить расположение ключевых элементов на странице и смоделировать действия пользователя на клавиатуре и с помощью мыши. Таким образом, робот будет затрачивать в несколько раз больше времени на одну итерацию, и это при условии, что получится успешно сопоставить координаты требуемого изображения предмета с вопросом. Если учитывать, что существует ресурс, который с помощью дешевой рабочей силы позволяет получить координаты нужного изображения предмета, это еще добавит, как минимум, 4–15 секунд. С другой стороны, пользователю требуется сделать лишь одно дополнительное нажатие мышью либо пером или пальцем, если используется сенсорная панель, что также делает метод более привлекательным по сравнению с CAPCHA, где для ввода слов используются сенсорные экраны. Следует отметить, что во многих случаях будет достаточно только первых двух элементов защиты и пользователю вообще не придется выполнять дополнительных действий. Другим преимуществом метода является возможность значительно расширить множество используемых изображений. Таким образом, программное распознавание образов становится трудновыполнимой задачей. Также роботу придется произвести лингвистический анализ предложения-просьбы для выявления предмета предложения. Очевидно, для реализации успешной атаки злоумышленнику потребуется значительно больше времени, чем для реализации атаки на ресурсы, защищенные другими методами.

Для последующего анализа и сравнения методов защиты относительно критериев и угроз безопасности определим следующую шкалу экспертных оценок, представленную в таблице 1.

Таблица 1.

Оценка	**Угроза безопасности**	**Критерий эффективности**
0	Метод не способен противодействовать	Не эффективен
1	Применение возможно, но не рекомендуется	Результаты удовлетворительные
2	Применение метода рекомендуется	Хорошая эффективность
3	Наилучшим образом подходит для применения	Наилучшая эффективность

Экспертные оценки описанных методов относительно критериев и угроз безопасности, описанных в работе, представлены в таблицах 2 и 3.

Таблица 2.

Критерий	**Метод**				
	CAPTCHA на основе текста	**CAPTCHA на основе вопроса**	**Метод на основе скрытых запросов**	**Метод на основе больших чисел**	**Авторский метод**
Затраты на программную реализацию робота (**P**)	1	1	1	0	3

Затраты на аппаратное обеспечение робота (**A**)	2	2	1	1	2
Стоимость полученной информации (**I**)	2	2	1	1	3
Доступность ресурса для пользователя (**T**)	1	1	3	3	2

Таблица 3.

Угроза безопасности	**Метод**				
	CAPTCHA на основе текста	**CAPTCHA на основе вопроса**	**Метод на основе скрытых запросов**	**Метод на основе больших чисел**	**Авторский метод**
Загрузка на сервер ложной информации (**U1**)	2	2	1	0	2
Покупка роботами (**U2**)	1	1	2	1	3
Компрометирование учетных записей (**U3**)	2	2	2	1	3
Тиражирование (**U4**)	0	0	2	1	3
Фарминг (**U5**)	0	0	2	1	3

Из приведенных оценок и анализа методов следует, что предложенный метод более эффективен относительно описанных классов критериев и угроз безопасности.

Выводы

Таким образом, очень кратко представлена в настоящей статье концепция защиты от программных роботов. Намечены пути решения проблемы обеспечения информационной безопасности в среде, где работают программные роботы и пользовательские невредоносные программы. Проанализировано представление проблемы компьютерных вирусов в УК РФ. Обращено внимание на то, что компьютерный вирус и вредоносная программа — это разные объекты. Намечены пути построения методики для определения вредоносной программы. Предложен новый метод защиты от программ-роботов. Выполнены анализ и оценка эффективности этого метода. На данном этапе получены некоторые результаты, которые способствуют эффективному

решению задачи обеспечения информационной безопасности. В процессе выполненной работы и проведенных исследований были успешно получены необходимые охранные документы РОСПАТЕНТа, например [9].

СПИСОК ЛИТЕРАТУРЫ:

1. *Козлов Д. А., Парандовский А. А., Парандовский А. К.* Энциклопедия компьютерных вирусов. М.: СОЛОН-Р, 2001. — 457 с.
2. *Таха Х.* Введение в исследование операций. М.: Вильямс, 2005. — 912 с.
3. *Кулик С. Д., Фролов Д. Б.* Правовые вопросы разработки вирусов и поисковых роботов АФИПС // Безопасность информационных технологий. 2001. № 1. С. 40—45.
4. *Кулик С. Д., Фролов Д. Б.* Защита АФИПС и правовые вопросы разработки вредоносных программ // Безопасность информационных технологий. 2001. № 3. С. 35—38.
5. *Кулик С. Д.* Анализ вредоносных программ и вирусов // Судебная экспертиза: дидактика, теория, практика. Сборник научных трудов. М.: Московский университет МВД России, 2010. Вып. 6. С. 180—195.
6. *Кулик С. Д.* Возможный подход к построению методики определения вредоносных программ по исходному тексту // Научная сессия МИФИ-2002. IX Всероссийская научно-практическая конференция «Проблемы информационной безопасности в системе высшей школы». Сборник научных трудов. М.: МИФИ, 2002. С. 54—55.
7. *Кулик С. Д.* Возможность построения методики определения вредоносных программ // Актуальные проблемы управления — 2001: Материалы международной научно-практической конференции: Вып. 4. М.: ГУУ, 2001. С. 189—190.
8. *Вайрадян А. С., Ковшов В.К., Подласов В. С., Пчелинцев И. П., Челышев М. М.* Подход к построению одного класса фактографических информационно-поисковых систем // VIII Всесоюзное совещание по проблемам управления. Таллин: Наука, 1980. С. 346—347.
9. *Кулик С. Д.* Свидетельство на программу Российской Федерации № 2000610206 «Самомодифицирующий вирус-мутант МИФ-ИКС-01, изменяющий свою сигнатуру» (MIF-X-01) / С. Д. Кулик (Россия). Заявка № 2000610049; Заяв. 27.01.2000; Зарегистр. 22.03.2000. Бюл. № 2 (31). С. 163—164. (РОСПАТЕНТ)
10. *Лахути Д. Г.* Автоматизированные документально-фактографические информационно-поисковые системы // Итоги науки и техники. М.: ВИНИТИ, 1988. С. 6—79.
11. *Попов И.И.* Применение адаптивных СУБД для реализации документальных и фактографических информационных систем // Математическое и информационное обеспечение систем принятия решений. М.: Энергоатомиздат, 1988. С. 19—28.
12. *Соколов А. В.* Информационно-поисковые системы. М.: Радио и связь,1981. — 152 с.
13. *Гайдамакин Н. А.* Система представления и обработки данных фактографических АИС // Автоматизированные информационные системы, базы и банки данных. М.: Гелиос АРВ, 2002. С. 27—32.
14. *Галушкин А. И.* Теория нейронных сетей. М.: ИПРЖР, 2000. — 416 с.
15. *Хайкин С.* Нейронные сети: полный курс. М.: Вильямс, 2008. — 1104 с.
16. *Ту Дж., Гонсалес Р.* Принципы распознавания образов. М.: Мир, 1978. — 411 с.
17. *Кулик С. Д., Ткаченко К. И.* Разработка генераторов для обеспечения информационной безопасности // Безопасность информационных технологий. 2010. № 1. С. 87—89.
18. *Тархов Д. А.* Нейронные сети как средство математического моделирования. М.: Радиотехника, 2006. — 48 с.
19. *Пахомов А. В.* Коллекции в правоохранительных органах России. М.: Юрлитинформ, 2001. 136 с.
20. *Кулик С. Д., Ткаченко С. И.* Подход к защите от программных роботов // Естественные и технические науки, 2014. № 3(71). С. 178—179.
21. *Лутц М.* Программирование на Python. Т. I. 4-е изд. Пер. с англ. СПб.: Символ-Плюс, 2011. — 992 с.
22. Взлом каптчи файлообменника [Электронный ресурс]. URL: http://habrahabr.ru/post/67194/ (дата обращения: 18.04.2014).
23. SEO: «Поисковая оптимизация от А до Я» Основы (версия от 27 марта 2014 года) [Электронный ресурс]. URL: http://yadi.sk/d/Hmo2EBdLLGr9M (дата обращения: 18.04.2014).
24. Руководящий документ Гостехкомиссии России «Защита от несанкционированного доступа к информации. Термины и определения». М.: ГТК РФ, 1992. — 13 с.
25. ГОСТ Р ИСО/МЭК 13335-1-2006. Национальный стандарт РФ. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий. Введен ВПЕРВЫЕ. Дата введения с 01.08.2007. М.: Стандартинформ, 2007. — 22 с.
26. HTTP cookie [Электронный ресурс]. URL: http://ru.wikipedia.org/wiki/HTTP_cookie (дата обращения: 24.04.2014).
27. Сервис по ручному распознаванию текста с изображений. Распределения работников по странам [Электронный ресурс]. URL: http://antigate.com/#workers (дата обращения: 14.04.2014).
28. Информационная безопасность [Электронный ресурс]. URL: http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RIt(uwsg.outtg9!hlnuvgxtuxy; (дата обращения: 24.04.2014).

REFERENCES:

1. *Kozlov D. A., Parandovskiy A. A., Parandovskiy A. K.* Entsiklopediya komp'yuternykh virusov. M.: SOLON-R, 2001. 457 p.
2. *Kulik S. D.* Issledovaniye poiskovogo robota dlya faktograficheskogo poiska // Nauchno-tekhnicheskaya informatsiya. 2003. Series.2. №3. P. 21–27.
3. *Kulik S. D., Frolov D. B.* Pravovyye voprosy razrabotki virusov i poiskovykh robotov AFIPS // Bezopasnost' informatsionnykh tekhnologiy. 2001. №1. P. 40–45.
4. *Kulik S. D., Frolov D. B.* Zashchita AFIPS i pravovyye voprosy razrabotki vredonosnykh programm // Bezopasnost' informatsionnykh tekhnologiy. 2001. №3. P. 35–38.
5. *Kulik S. D.* Analiz vredonosnykh programm i virusov // Sudebnaya ekspertiza: didaktika, teoriya, praktika. Sbornik nauchnykh trudov. M.: Moskovskiy universitet MVD Rossii. 2010. Vypusk 6. P. 180–195.
6. *Kulik S. D.* Vozmozhnyy podkhod k postroyeniyu metodiki opredeleniya vredonosnykh programm po iskhodnomu tekstu // Nauchnaya sessiya MIFI-2002. IX Vserossiyskaya nauchno-prakticheskaya konferentsiya "Problemy informatsionnoy bezopasnosti v sisteme vysshey shkoly". Sbornik nauchnykh trudov. M.: MIFI, 2002. P. 54–55.
7. *Kulik S. D.* Vozmozhnost' postroyeniya metodiki opredeleniya vredonosnykh programm // Aktual'nyye problemy upravleniya -2001: Materialy mezhdunarodnoy nauchno-prakticheskoy konferentsii: Vypusk 4. M.: GUU, 2001. P. 189–190.
8. *Kulik S. D.* Ob"yekty intellektual'noy sobstvennosti Rossii (obzor programmnogo obespecheniya): Nauchnoye izdaniye. M.: Kompaniya Sputnik+, 2001. 159 p.
9. *Kulik S. D.* Certificate program at the Russian Federation №2000610206 "Samomodifitsiruyushchiy virus-mutant MIF-IKS-01, izmenyayushchiy svoyu signaturu" (MIF-X-01)/ *S. D. Kulik* (Russia). Zayavka №2000610049; Zayavleno 27.01.2000; Zaregistrirovano 22.03.2000. Bull. №2 (31). P. 163–164. — (ROSPATENT).
10. *Kulik S. D., Nikonets D. A., Tkachenko K. I., Lukyanov I. A.* Metody i sredstva povysheniya effektivnosti informatsionnykh sistem (neyronnyye seti, kriminalistika, formirovaniye faktograficheskikh dannykh, morfologicheskiy analiz). Tom 3: Prilozheniya. / Izdatel'stvo Radiotekhnika (Deponirovano v VINITI 05.05.2011, № 208-B2011; Bibl. Ukazatel' №7(473)). M. 2011. 229 p.
11. *Kulik S. D.* Faktograficheskiye sistemy (metody postroyeniya, modeli, strategii poiska i programmnoye obespecheniye) / Izdatel'stvo Radiotekhnika (Deponirovano v VINITI 23.06.2003, №1205-B2003; Bibl. Ukazatel' № 8(378), 2003). M. 2003. 325 p.
12. *Kulik S. D.* Issledovaniye effektivnosti faktograficheskogo poiska v informatsionnykh sistemakh / Izdatel'stvo Radiotekhnika (Deponirovano v VINITI 29.07.2004, №1326-B2004; Bibl. Ukazatel' №9(391), 2004). M. 2004. 251 p.
13. *Kulik S. D., Nikonets D. A., Tkachenko K. I.,* Zhizhilev *A. V.* Ustroystvo opredeleniya poddel'nykh dokumentov // Bezopasnost' informatsionnykh tekhnologiy. 2009. №1. P. 114–115.
14. *Kulik S. D., Nikonets D. A., Tkachenko K. I., Lukyanov I. A.* Metody i sredstva povysheniya effektivnosti informatsionnykh sistem (neyronnyye seti, kriminalistika, formirovaniye faktograficheskikh dannykh, morfologicheskiy analiz). Tom 1: Kriminalistika. / Izdatel'stvo Radiotekhnika (Deponirovano v VINITI 05.05.2011, №206-B2011; Bibl. Ukazatel' №7(473)). M. 2011. 300 p.
15. *Kulik S. D., Nikonets D. A.* Primery ispol'zovaniya neyrosetevogo algoritma v metodikakh dlya eksperta-pocherkoveda // Neurocomputers: razrabotka i primeneniye. 2009. №9. P. 61–85.
16. *Kulik S. D.* Primeneniye neyronnykh setey v avtomatizirovannykh faktograficheskikh informatsionno-poiskovykh sistemakh // Neurocomputers: razrabotka i primeneniye. 2002. №5-6. P. 3–12.
17. *Kulik S. D., Tkachenko K. I.* Razrabotka generatorov dlya obespecheniya informatsionnoy bezopasnosti // Bezopasnost' informatsionnykh tekhnologiy. 2010. №1. P. 87–89.
18. *Kulik S. D.* Neyronnyye seti v avtomatizirovannykh faktograficheskikh informatsionno-poiskovykh sistemakh // Neurocomputers: razrabotka i primeneniye. 2007. №2-3. P. 60–66.
19. *Kulik S. D.* Patent for invention №2208837, Russian Federation (RU), G06 F17/30. Ustroystvo dlya imitatsionnogo modelirovaniya znacheniy funktsii vykhoda avtomatizirovannoy faktograficheskoy informatsionno- poiskovoy sistemy kriminalisticheskogo naznacheniya. — Zayavka №2001129139/09; Zayavleno 30.10.2001; Zaregistrirovano 20.07.2003; Priority 30.10.2001; Bull. №20. Part 3. P. 752–753. (ROSPATENT).
20. *Kulik S. D.* Certificate program at the Russian Federation №2000610153 "Zashchita bazy dannykh ot razrusheniy virusom ili khakerom" (VDPRO) /*S. D. Kulik* (Russia). Zayavka №991042; Zayavleno 30.12.1999; Zaregistrirovano 23.02.2000. Bull. №2(31). P. 120–121. — (ROSPATENT).
21. *Lutz M.* Programmirovaniye na Python, tom I, 4-ye izdaniye. SPb.: Simvol-Plyus. 2011. 992 p.
22. Vzlom kaptchi fayloobmennika [Elektronnyy resurs]. — URL: http://habrahabr.ru/post/67194/ (data obrashcheniya: 18.04.2014).
23. SEO: Poiskovaya optimizatsiya ot A do YA» — Osnovy (versiya ot 27 marta 2014 goda) [Elektronnyy resurs]. — URL: http://yadi.sk/d/Hmo2EBdLLGr9M (data obrashcheniya: 18.04.2014).
24. Rukovodyashchiy dokument Gostekhkomissii Rossii «Zashchita ot nesanktsionirovannogo dostupa k informatsii. Terminy i opredeleniya». M.: GTK RF. 1992. 13 p.
25. GOST R ISO/MEK 13335-1-2006. Natsional'nyy standart RF. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Chast' 1. Kontseptsiya i modeli menedzhmenta bezopasnosti informatsionnykh i telekommunikatsionnykh tekhnologiy. Vveden VPERVYYe. Data vvedeniya s 01.08.2007. M.: Standartinform. 2007. 22 p.
26. HTTP cookie [Elektronnyy resurs]. — URL: http://ru.wikipedia.org/wiki/HTTP_cookie (data obrashcheniya: 24.04.2014).
27. Cervis po ruchnomu raspoznavaniyu teksta s izobrazheniy. Raspredeleniya rabotnikov po stranam. [Elektronnyy resurs]. — URL: http://antigate.com/#workers (data obrashcheniya: 14.04.2014).
28. Informatsionnaya bezopasnost'. [Elektronnyy resurs]. — URL: http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RIt(uwsg.outtg9!hlnuvgxtuxy; (data obrashcheniya: 24.04.2014).