На правах рукописи

Печенкин Александр Игоревич

ВЫСОКОПАРАЛЛЕЛЬНАЯ СИСТЕМА ВЫЯВЛЕНИЯ СЕТЕВЫХ УЯЗВИМОСТЕЙ НА ОСНОВЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Специальность 05.13.19

Методы и системы защиты информации, информационная безопасность

Автореферат диссертации на соискание ученой степени кандидата технических наук

Санкт-Петербург – 2013

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет»

Научный руководитель:

Зегжда Дмитрий Петрович,
доктор технических наук, профессор

Официальные оппоненты:

Саенко Игорь Борисович,
доктор технических наук, профессор,
ведущий научный сотрудник
лаборатории проблем компьютерной
безопасности СПИИРАН

Трифаленков Илья Анатольевич,
кандидат технических наук,
директор Центра информационной
безопасности группы компаний «R-Style»

Ведущая организация: ФГУП «Главный научно-исследовательский вычислительный центр Федеральной налоговой службы»

Защита состоится « » декабря 2013 г. в часов на заседании диссертационного совета Д 212.229.27 при ФГБОУ ВПО «Санкт-Петербургский государственный политехнический университет» (по адресу 195251, Санкт-Петербург, ул. Политехническая, д.29/1, ауд. 175 главного здания)

С диссертационной работой можно ознакомиться в Фундаментальной библиотеке ФГБОУ ВПО «Санкт-Петербургский государственный политехнический университет».

Автореферат разослан « » ноября 2013г.

Ученый секретарь
диссертационного совета

Платонов Владимир Владимирович

Общая характеристика работы

Актуальность темы исследования. Использование уязвимостей – один из основных способов распространения вредоносного программного обеспечения, поэтому их поиск является одной из важнейших задач информационной безопасности. Уязвимостью называется ошибка в программе, которая может быть использована для реализации угроз безопасности. Согласно стандартам ISO 15408 и ГОСТ 15408 поиск уязвимостей является одним из обязательных этапов оценки безопасности программного обеспечения (ПО). В работе исследуется задача поиска сетевых уязвимостей (уязвимостей в реализациях сетевых протоколов), позволяющих злоумышленнику удаленно выполнить произвольный код на компьютере, подключенном к сети, в том числе Интернет, что делает их одними из наиболее критичных. На данный момент в мире более 10 млрд. устройств используют сетевые протоколы для подключения к сети Интернет, а значит, их безопасность напрямую зависит от своевременного выявления подобных уязвимостей.

Методы обнаружения уязвимостей делятся на два класса: статические, проводящие анализ кода без его исполнения, и динамические, основанные на контроле различных параметров в ходе работы ПО. Одним из основных показателей эффективности динамических методов является покрытие кода программы за минимальное время. Необходимость улучшения данного показателя привела к формированию двух направлений совершенствования методов динамического анализа:

– увеличению полноты покрытия программного кода при фиксированном времени проведения анализа;

– сокращению времени анализа для достижения заданного уровня покрытия программного кода.

В работе предложены решения по совершенствованию динамического поиска сетевых уязвимостей в обоих направлениях, основанные на применении генетических алгоритмов для максимизации покрытия программного кода и масштабировании задачи поиска на многопроцессорных кластерах для сокращения времени. Предложенные решения позволяют повысить эффективность поиска уязвимостей в реализациях сетевых протоколов и, следовательно, безопасность локальных и глобальных сетей, что делает актуальной тему настоящего исследования.

Степень разработанности темы исследования. Исследованию поиска уязвимостей в реализациях сетевых протоколов динамическими методами посвящено множество работ российских и зарубежных ученых, таких как В.А. Падарян, А.Ю. Тихонов, С. Горбунов, А. Розенблум, Р. Кеммерер, Г Бенкс, Е. Монте де Ока, М. Бишоп, С. Вален, Ж. Ву. Предложенные в данных работах методы требуют предварительно представленных оператором знаний о сетевых протоколах (исходных кодов, спецификаций протоколов в специальном формате и т.д.), что существенно сокращает возможности их практического применения. Также следует отметить, что при реализации на современных высокопроизводительных системах (например, многопроцессорных кластерах) существующие методы не позволяют эффективно использовать имеющиеся вычислительные мощности в связи с тем, что их работа требует последовательного выполнения задачи. Таким образом, имеющиеся подходы к динамическому поиску сетевых уязвимостей и алгоритмы их реализации должны становиться более универсальными по отношению к исследуемым протоколам и нуждаются в адаптации для обеспечения возможности высокопараллельной работы систем выявления сетевых уязвимостей на многопроцессорных кластерах.

Целью работы является разработка методов и средств поиска сетевых уязвимостей, обеспечивающих высокую степень покрытия кода за счет применения генетических алгоритмов и масштабирования задачи поиска на многопроцессорных кластерах. Для достижения поставленной цели в работе решались следующие задачи:

1. Исследование современных методов поиска сетевых уязвимостей.
2. Построение формальной модели сетевых уязвимостей, позволяющей сформулировать необходимое и достаточное условие наличия сетевой уязвимости.
3. Формализация постановки задачи динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления.
4. Разработка метода поиска сетевых уязвимостей на основе максимизации покрытия графа передачи управления с помощью генетического алгоритма.
5. Построение модели масштабирования динамического поиска сетевых

уязвимостей, обеспечивающей эффективное использование вычислительных ресурсов многопроцессорного кластера.

6. Разработка архитектуры и экспериментального макета высокопараллельной системы выявления сетевых уязвимостей на основе генетических алгоритмов и оценка эффективности его работы.

Научная новизна диссертационной работы состоит в следующем:

– разработана формальная модель сетевых уязвимостей, позволившая доказать необходимое и достаточное условие наличия сетевой уязвимости;

– формализована задача динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления;

– разработан метод поиска сетевых уязвимостей на основе максимизации покрытия графа передачи управления с помощью генетического алгоритма, и определены оптимальные значения параметров генетического алгоритма;

– создана модель масштабирования динамического поиска сетевых уязвимостей, основанного на использовании генетических алгоритмов, позволившая разработать алгоритмы масштабирования задачи и балансировки нагрузки.

Теоретическая и практическая значимость работы. Полученные теоретические и экспериментальные результаты могут быть использованы при анализе безопасности программного обеспечения как в процессе его сертификации и аттестации автоматизированных информационных систем, так и при разработке сетевых сервисов. Теоретические и экспериментальные результаты работы используются для подготовки специалистов в области защиты вычислительных систем по дисциплине "Разработка Интернет-приложений" в ФГБОУ ВПО "СПбГПУ", а также использованы в НИР "Исследование и разработка макета программного комплекса высокоскоростного процессинга (обработки) сетевого трафика в режиме реального времени" (шифр 2012-1.4-07-514-0011-007) по государственному контракту от 23 мая 2012 г. № 07.514.11.4122, при разработке метода высокоскоростного анализа сетевого трафика на многопроцессорном кластере в СПБГУТ им. М.А. Бонч-Бруевича, при проведении работ по анализу безопасности и последующей доработке телекоммуникационных систем в ЗАО "Голлард", что подтверждается соответствующими актами об использовании.

Методология и методы исследования. Для решения поставленных задач использовались системный анализ, теория алгоритмов, теория распределенных вычислений, методы математического и эволюционного моделирования, методы решения задач оптимизации.

Положения, выносимые на защиту:

1. Формальная модель сетевых уязвимостей, формулировка теоремы о необходимом и достаточном условии наличия сетевой уязвимости.
2. Формальная постановка задачи динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления.
3. Метод поиска сетевых уязвимостей на основе максимизации покрытия графа передачи управления с помощью генетического алгоритма.
4. Полученные в результате исследований оптимальные значения параметров генетического алгоритма для поиска сетевых уязвимостей.
5. Модель масштабирования динамического поиска сетевых уязвимостей, основанного на использовании генетических алгоритмов.

Степень достоверности научных положений диссертации определяется строгим теоретическим обоснованием предлагаемого аналитического аппарата, эффективностью его использования при практическом воплощении и результатами экспериментальных исследований.

Апробация результатов работы. Основные теоретические и практические результаты диссертационной работы доложены и обсуждены: на Санкт-Петербургской межрегиональной конференции "Информационная безопасность регионов России" (Институт информатики и автоматизации РАН, 2011 г.), на всероссийской конференции "Проведение научных исследований в области обработки, хранения, передачи и защиты информации" ("МСП ИТТ", 2011 г.), на 20-ой и 21-ой научно-технической конференции "Методы и технические средства обеспечения безопасности информации" (СПбГПУ, 2011 г., 2012 г.), на 15-ой международной научно-практической конференции "РусКрипто" (Ассоциация "РусКрипто" и Академия Информационных Систем, 2013 г.), на 12-ой всероссийской конференции "Информационная безопасность. Региональные аспекты. ИнфоБЕРЕГ-2013" (Академия Информационных Систем, 2013 г.). Работа представлена к присуждению премии правительства Санкт-Петербурга победителям конкурса грантов для студентов вузов,

расположенных на территории Санкт-Петербурга, аспирантов вузов, отраслевых и академических институтов, расположенных на территории Санкт-Петербурга.

Публикации. По теме диссертации опубликовано 13 научных работ, в том числе 6 в изданиях из перечня ВАК.

Объем и структура. Диссертация состоит из введения, четырех глав, заключения и списка литературы из 73 наименований.

Основное содержание работы

Во введении сформулирована и обоснована задача работы, дано определение понятия сетевой уязвимости.

В первой главе представлены результаты анализа современных методов поиска сетевых уязвимостей, описана разработанная формальная модель сетевых уязвимостей, сформулировано и доказано необходимое и достаточное условие наличия сетевой уязвимости, формализована задача динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления.

В работе рассмотрены существующие методы и средства динамического поиска сетевых уязвимостей и показано, что они используют два подхода к получению тестовых данных: генерация и мутация. Генерация — создание входных данных согласно заданным правилам, в том числе случайным образом. Такой подход используется, например, программой Sulley, которая генерирует данные по заданным оператором спецификациям протокола. Мутация — создание входных данных путем внесения модификаций в использовавшийся ранее набор данных. Такой подход реализуется в программе HotFuzz, которая модифицирует данные предварительно собранного дампа сетевого трафика. В работе показано, что зависимость существующих методов от априорных знаний (спецификации протокола, дампа сетевого трафика и т.д.) неизбежно приводит к ухудшению одного из основных показателей эффективности динамического анализа ПО – покрытия программного кода. Ограниченность рамками априорных знаний не позволяет выйти за их пределы. В свою очередь, действия злоумышленника, направленные на поиск и эксплуатацию уязвимости, заключаются как раз в обходе этих ограничений.

В основе разработанной модели сетевых уязвимостей лежит представление сетевого протокола в виде кортежа

$PROTOCOL = (PROGRAM, I)$. Программа-обработчик сетевого протокола (сетевой сервис) представляется ориентированным графом передачи управления $PROGRAM = (INSTR, E)$, который описывается множеством инструкций программного кода $INSTR$ (вершин графа) и множеством переходов $E \subset INSTR^2$ (ребер графа). Элементами множества входных данных I являются сетевые пакеты. Выполнение обработки входных данных описывается отображением $run: I \to R$, где элементы множества R – все возможные результаты обработки входных данных. Оператор $execution\big(instr_k, f_k(input_j)\big) = (instr_n, f_n(f_k(input_j)))$ описывает выполнение инструкции $instr_k$ с данными $f_k(input_j)$ и переход к выполнению следующей инструкции $instr_n$, а f_k и f_n описывают преобразования входных данных программы $PROGRAM$ в ходе её выполнения. Обработка входных данных $input_j \in I$ происходит по трассе $TRACE = trace(input_j)$. $TRACE$ является подграфом графа $PROGRAM$, а множество инструкций этой трассы определяется отображением $trace_{INSTR}: I \to \mathcal{P}(INSTR)$, где $\mathcal{P}$ – обозначение булеана.

При проявлении в сетевом сервисе уязвимостей, позволяющих злоумышленнику выполнить произвольный код, происходит либо попытка исполнения кода, не являющегося корректной инструкцией, либо попытка выхода за пределы адресного пространства сервиса. Некорректным результатом работы программы $r_k \in R$ в этом случае является попытка применения оператора $execution(instr_n, data)$ к инструкции $instr_n \notin INSTR$. В соответствии с формальным представлением программы и протокола в рамках предложенной модели уязвимость программы $PROGRAM$, реализующей сетевой протокол $PROTOCOL = (PROGRAM, I)$, представляет собой существование таких входных данных $input_j \in I$, обработка которых приводит к некорректному результату $r_k = run(input_j)$. Такие входные данные составляют множество данных $VULN$, приводящих к проявлению уязвимостей.

Определение 1: инструкция $instr_k \in INSTR$ достижима на входных данных $input_j \in I$, если $instr_k \in trace_{INSTR}(input_j)$.

Определение 2: инструкция $instr_k \in INSTR$ приводит к ошибке на входных данных $input_j \in I$, если $execution(instr_k, f(input_j)) \notin INSTR$.

На основе модели сетевых уязвимостей в работе сформулировано и

доказано необходимое и достаточное условие наличия сетевой уязвимости:

Теорема 1. Сетевая уязвимость проявляется при обработке входных данных $input_j \in I$ тогда и только тогда, когда существует инструкция $instr_k \in INSTR$, которая одновременно достижима и ошибочна на этих входных данных:

$$input_j \in VULN \Leftrightarrow \exists\, instr_k \in INSTR\colon \begin{cases} instr_k \in trace_{INSTR}(input_j), \\ execution(instr_k, f(input_j)) \notin INSTR. \end{cases}$$

Достаточность следует из определений достижимости и ошибочности инструкции, необходимость доказывается от противного. В работе выведено и доказано следующее следствие:

Следствие 1. Для обнаружения всех сетевых уязвимостей в сетевом сервисе с использованием динамических методов необходимо обеспечить полное покрытие его графа передачи управления.

В работе предлагается описывать покрытие графа передачи управления сетевого сервиса функцией $used(I_{TEST}) = |USED|$, где $USED$ – множество покрытых инструкций после тестирования на множестве входных данных I_{TEST}. Следствие 1 позволило формализовать задачу динамического поиска сетевых уязвимостей в виде задачи $used(I_{TEST}) \to max$ максимизации покрытия графа передачи управления сетевого сервиса.

В работе введены понятия скорости тестирования $v = tests/t$, где $tests$ – количество тестов, проведенных за время тестирования t, и среднего числа впервые покрытых инструкций за один тест $q = |USED|/tests$. Для функции покрытия графа передачи управления получено соотношение $used(I_{TEST}) = t \cdot v \cdot q$, позволившее сформулировать и доказать следующее утверждение:

Теорема 2. Показатель покрытия графа передачи управления при динамическом поиске сетевых уязвимостей прямо пропорционален скорости тестирования и среднему числу впервые покрытых инструкций за один тест.

Теорема 2 показывает, что для повышения показателя покрытия кода необходимо, с одной стороны, увеличить среднее число впервые покрытых инструкций за один тест, с другой стороны, повысить скорость обработки тестовых данных. Для решения первой задачи в работе предложен метод, основанный на использовании генетического алгоритма. Вторая задача решена

в работе с помощью организации параллельных вычислений на многопроцессорном кластере.

Во второй главе предложен метод поиска сетевых уязвимостей на основе максимизации покрытия графа передачи управления с помощью генетического алгоритма и представлены полученные оценки оптимальных значений параметров генетического алгоритма для поиска сетевых уязвимостей.

Формальная постановка задачи динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления сетевого сервиса ($used(I_{TEST}) \to max$) позволила предложить метод, направленный на максимизацию покрытия кода за счет применения основного механизма теории эволюционных вычислений: генетического алгоритма. Основными циклически выполняемыми этапами метода являются:

1. Генерация популяции I_k^T входных тестовых данных. На первой итерации цикла производится генерация случайной начальной популяции I_1^T, на последующих производится наследование путем применения операторов скрещивания и мутации к множеству эталонных данных: $I_k^T = inherit(I_{k-1}^E)$.
2. Анализ обработки особей популяции: получение трассы $trace(input)$, проверка условия наличия уязвимости, оценка покрытия кода $used(I_k^T)$.
3. Оценка целевой функции $fitness(input)$ для особей популяции.
4. Отбор эталонных данных $I_k^E = selection(I_k^T)$.

Критерием отбора особей на каждой итерации является наименьшее значение целевой функции $fitness(input)$, оценивающей расстояние от трассы обработки данных $trace(input)$ до ближайшего непокрытого участка кода (инструкций множества $UNUSED$). Множества покрытых и непокрытых инструкций изменяются после каждой обработки входных данных:

$$USED_{tests} = USED_{tests-1} \cup trace_{INSTR}(input_i),$$
$$UNUSED_{tests} = UNUSED_{tests-1} \setminus trace_{INSTR}(input_i).$$

Функция $I_k^T = inherit(I_{k-1}^E)$ обеспечивает выполнение операторов скрещивания (используется одноточечное скрещивание) и мутации эталонных особей для рождения особей нового поколения. В работе экспериментально подтверждены теоретические предположения о том, что использование генетического алгоритма позволяет достичь улучшения показателя покрытия кода (рисунок 1).

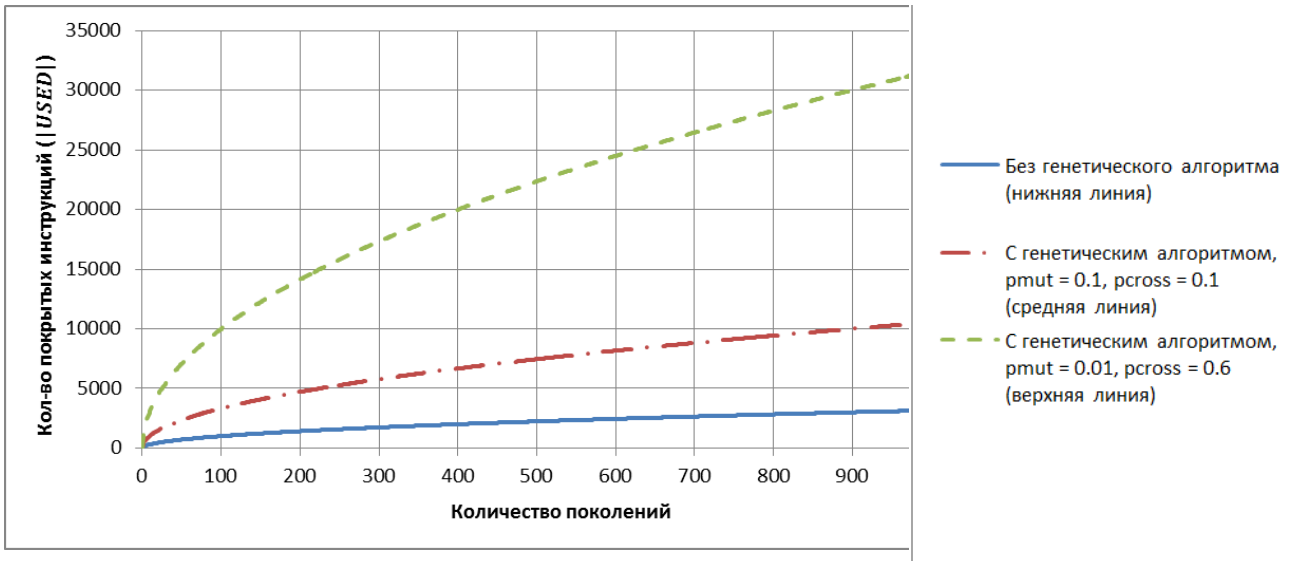


Рисунок 1 – Покрытие кода при различных значениях параметров генетического алгоритма

В работе определены значения параметров генетического алгоритма (параметров скрещивания и мутации), позволяющих достичь наилучших показателей покрытия кода, для чего проведены экспериментальные исследования на девяти протоколах операционной системы семейства Windows ($N = 9$). Для каждого из них экспериментально оценено количество покрытых инструкций при различных значениях вероятности мутации $pmut \in (0,1]$ и вероятности скрещивания $pcross \in (0,1]$.

Экспериментальные данные позволили получить значения функций $v_i(pmut, pcross)$ – средней скорости прироста покрытых инструкций графа передачи управления обработчика i-го исследуемого протокола в зависимости от параметров генетического алгоритма. Для нормирования значений введены функции относительных средних скоростей $rv_i(pmut, pcross) = \frac{v_i(pmut,pcross)}{v_i^{max}}$, где максимальная скорость для каждого протокола $v_i^{max} = \max_{pmut\in(0,1],pcross\in(0,1]}(v_i(pmut, pcross))$.

Для получения совокупной оценки по всем исследуемым протоколам введена функция зависимости выборочного среднего значения показателей относительной скорости от параметров генетического алгоритма $\overline{rv}(pmut, pcross) = \frac{\sum_{i=1}^{i=N} rv_i(pmut,pcross)}{N}$. На рисунке 2 представлен график функции $\overline{rv}(pmut, pcross)$, значения которой вычислены на основе полученных экспериментальных данных.

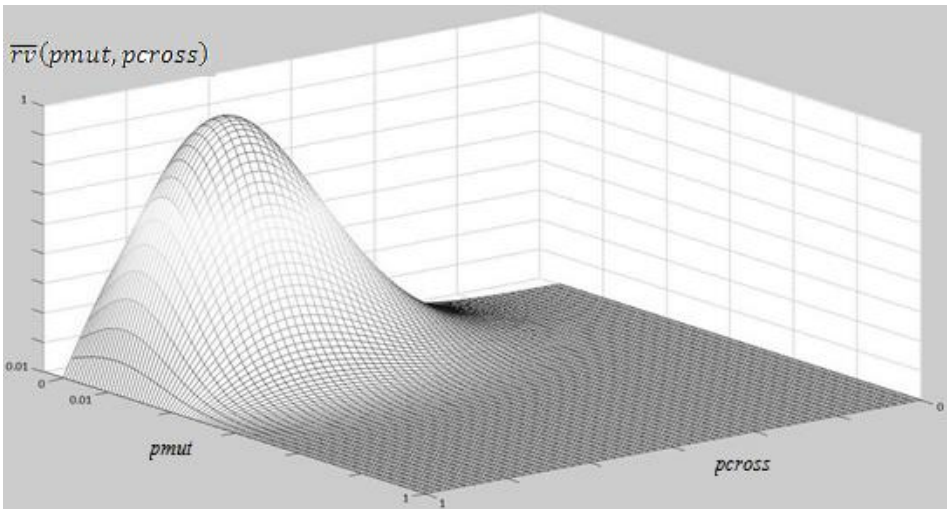


Рисунок 2 – Аппроксимированная функция $\overline{rv}(pmut, pcross)$

В результате проведенных исследований получены оптимальные значения параметров генетического алгоритма: $pmut = 0.01$, $pcross = 0.6$, при которых выборочное среднее $\overline{rv} = 0.99$ и коэффициент вариации $V = \sigma/\overline{rv} = 0.46\%$. Значения выборочного среднего и коэффициент вариации показывают, что при данных значениях параметров генетического алгоритма обеспечивается высокое покрытие кода для всех исследуемых протоколов.

В третьей главе построена модель масштабирования динамического поиска сетевых уязвимостей, основанного на использовании генетических алгоритмов, предложены алгоритмы масштабирования задачи и балансировки нагрузки при поиске уязвимостей на многопроцессорном кластере.

Теорема 2 показывает, что одним из способов улучшения показателя покрытия кода при динамическом поиске сетевых уязвимостей является повышение скорости обработки тестовых данных, которого можно достичь с помощью использования вычислительных мощностей многопроцессорных кластеров. Для решения данной задачи разработана модель, представляющая собой кортеж (K, G, VS, VA, I), где K – множество узлов многопроцессорного кластера, G – виртуальная машина генерации данных, VS – множество виртуальных машин отправки данных, VA – множество виртуальных машин обработки данных. Общее число виртуальных машин $amount_V$ равняется $amount_{VS} + amount_{VA} + 1$, где $amount_{VS} = |VS|$, а $amount_{VA} = |VA|$.

В работе предложена совокупность критериев эффективности использования вычислительных мощностей многопроцессорного кластера при решении задачи динамического поиска сетевых уязвимостей:

$$\begin{cases} V_S \longrightarrow \max, \\ V_A \longrightarrow \max, \\ |V_S - V_A| \longrightarrow \min, \\ amount_V = amount_K, \end{cases}$$

где V_S – интегральная скорость отправки данных со всех виртуальных машин множества VS, V_A – интегральная скорость обработки данных всеми виртуальными машинами множества VA, $amount_K = |K|$.

В работе показано, что для выполнения критериев необходимо обеспечить масштабирование задачи (для выполнения условий $amount_V = amount_K$ и $|V_S - V_A| \longrightarrow \min$) и балансировку нагрузки (для выполнения условий $V_S \longrightarrow \max$ и $V_A \longrightarrow \max$).

Разработанный алгоритм масштабирования задачи, описанный в работе, основывается на динамическом изменении соотношения количества виртуальных машин отправки данных и виртуальных машин обработки в соответствии с правилами, приведенными в таблице 1.

Таблица 1 – Правила алгоритма масштабирования

Условие	**Действие**
$\begin{cases} V_S > V_A, \\ amount_V < amount_K \end{cases}$	запуск виртуальной машины обработки
$\begin{cases} V_S \leq V_A, \\ amount_V < amount_K \end{cases}$	запуск виртуальной машины отправки
$\begin{cases} V_S < V_A, \\ amount_V = amount_K \end{cases}$	остановка виртуальной машины обработки, запуск виртуальной машины отправки
$\begin{cases} V_S > V_A, \\ (V_S - \frac{V_S}{amount_{VS}}) > (V_A + \frac{V_A}{amount_{VA}}) \\ amount_V = amount_K \end{cases}$	остановка виртуальной машины отправки, запуск виртуальной машины обработки

Алгоритм балансировки нагрузки $balance: I \rightarrow (VS, VA)$ ставит в соответствие данным $input_i \in I$, поступившим от машины генерации, виртуальную машину $vs_m \in VS$, которая будет выполнять их отправку, и виртуальную машину $va_n \in VA$, которая будет выполнять их обработку. Критерием выбора виртуальных машин является наименьшая загруженность вычислительного процессора: $balance(input_i) = (vs_m, va_n)$, где m и n таковы, что $\forall\, vs_j \in VS$ и $\forall\, va_k \in VA$ выполняются неравенства $load_{CURR}(vs_m) \leq load_{CURR}(vs_j)$ и $load_{CURR}(va_n) \leq load_{CURR}(va_k)$. Здесь $load_{CURR}(v)$ – загрузка процессора виртуальной машины v.

Представленная в работе модель масштабирования динамического поиска сетевых уязвимостей, основанного на использовании генетических алгоритмов, в совокупности с алгоритмами масштабирования задачи и балансировки нагрузки обеспечивает:

– минимизацию числа незадействованных вычислительных узлов многопроцессорного кластера путем запуска в динамической пропорции виртуальных машин отправки и обработки данных;

– распределение тестовых данных между виртуальными машинами отправки и обработки таким образом, чтобы достигался баланс нагрузки на вычислительные узлы кластера.

В четвертой главе представлено описание архитектуры и макета высокопараллельной системы поиска сетевых уязвимостей и приведены результаты экспериментальных исследований работы системы.

Предложенная в работе архитектура высокопараллельной системы выявления сетевых уязвимостей (рисунок 3) основывается на разработанной модели масштабирования. Возможность масштабирования и балансировки нагрузки обеспечивается за счет наличия непрерывного многосвязного цикла в процессе динамического поиска.

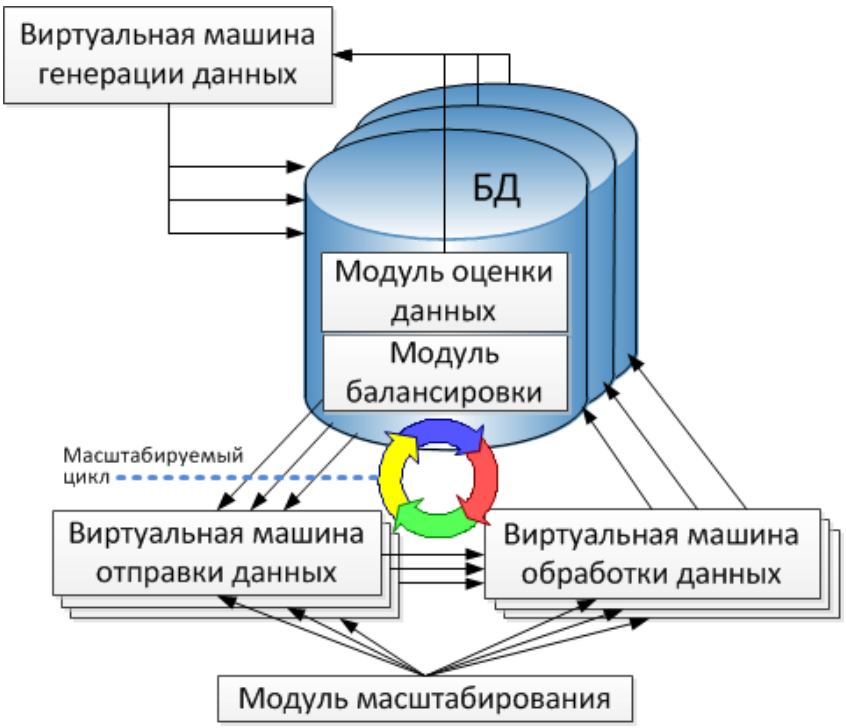


Рисунок 3 – Масштабируемая архитектура высокопараллельной системы выявления сетевых уязвимостей на основе генетических алгоритмов

Исследование полноты покрытия кода, обеспечиваемого использованием генетического алгоритма, проводилось на реализации протокола FTP операционной системы Windows Server 2012. Разработанный макет сравнивался с программными средствами Sulley и HotFuzz, также реализующими динамический поиск сетевых уязвимостей (рисунок 4). Из-за отсутствия возможности распределенной работы вышеуказанные средства тестировались на одном персональном компьютере (процессор Core i7 – 4 выч. ядра, 8 Гб RAM). Исследование экспериментального макета проводились на персональном компьютере и многопроцессорном кластере (суммарно 160 выч. ядер, 600 Гб RAM).

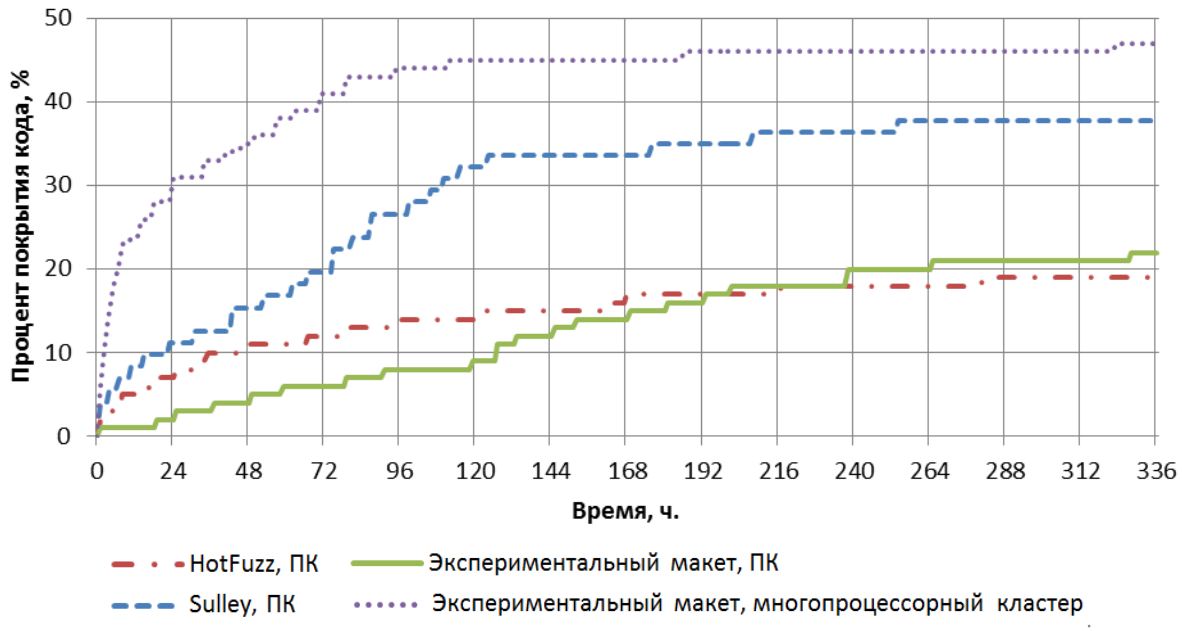
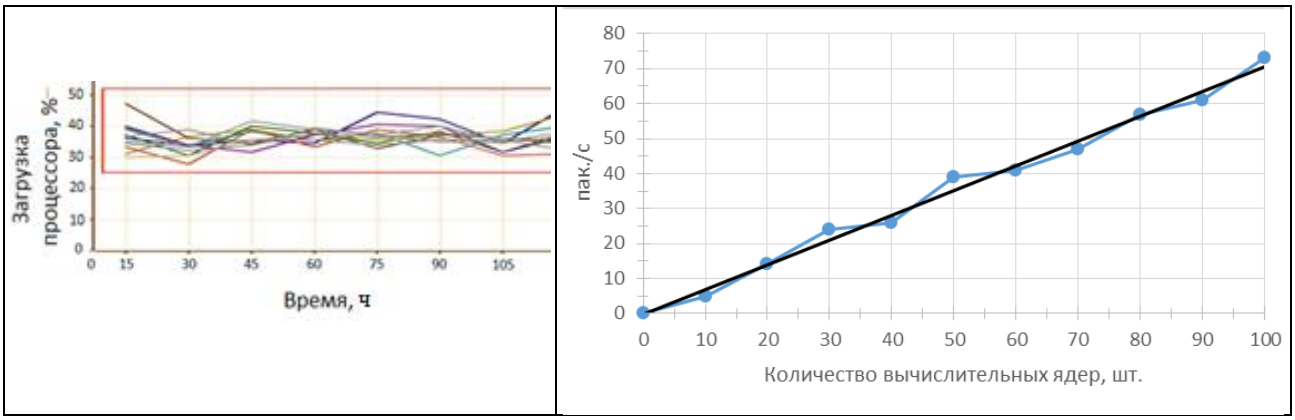


Рисунок 4 – Покрытие кода сетевого сервиса

Наибольший процент покрытия кода в рамках исследований продемонстрировал экспериментальный макет, что подтверждает корректность разработанной теоретической базы. Однако стоит отметить низкую скорость покрытия кода на начальном этапе тестирования в случае, если не используются возможности параллельных вычислений. Это вызвано отсутствием априорной информации о структуре протокола, используемой остальными исследуемыми методами (спецификация протокола, дамп сетевого трафика и т.д.). С течением времени выигрыш от наличия априорных знаний нивелируется результатами эволюционного развития.

Низкая начальная скорость покрытия кода полностью компенсируется эффективным использованием вычислительных мощностей многопроцессорного кластера, которое стало возможным благодаря масштабированию динамического поиска сетевых уязвимостей. Разработанный алгоритм обеспечивает линейность масштабирования при увеличении числа вычислительных узлов кластера (рисунок 5б). На рисунке 5а представлен график загрузки вычислительных узлов кластера при работе экспериментального макета. Каждая линия отображает загрузку одного процессора. Из графика видно, что предложенный алгоритм балансировки нагрузки обеспечил загрузку всех вычислительных узлов кластера, близкую к 50%, что является максимально возможным значением для процессоров с технологией Hyper Threading.

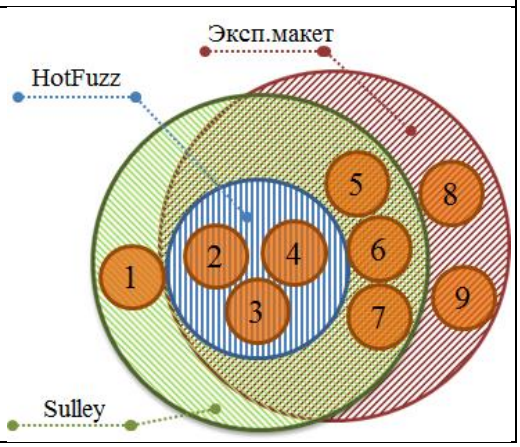


а) балансировка нагрузки б) масштабирование

Рисунок 5 – Результаты балансировки нагрузки и масштабирования системы выскокопараллельного поиска сетевых уязвимостей

В качестве объекта экспериментальных исследований по обнаружению уязвимостей была выбрана реализация специального протокола взаимодействия между агентами системы мониторинга состояния компьютерной сети. Проведенные испытания показали, что использование предложенных в работе методов и моделей позволило покрыть участки кода, не покрытые другими сравниваемыми средствами, и вследствие этого обнаружить уязвимости, которые они выявить не смогли (таблица 2).

Таблица 2 – Результаты обнаружения уязвимостей

Средство	**Предварит. исследования**	**Кол-во обнаруж. уязв. (обнар. только данным средством)**	**Соотношение множеств обнаруженных уязвимостей**
Sulley	Описание спецификации протокола	7 (1)	
HotFuzz	Сбор дампа трафика 5 клиентов за 3 часа	3 (0)	
Эксп. макет	нет	8 (2)	

В заключении приведены результаты и выводы, полученные автором в ходе выполнения работы.

Заключение

В работе получены следующие основные результаты:

1. Выявлены основные недостатки методов анализа безопасности сетевого программного обеспечения, и определены направления совершенствования динамического поиска сетевых уязвимостей.
2. Построена формальная модель сетевых уязвимостей, позволившая доказать необходимое и достаточное условие наличия сетевой уязвимости.
3. Формализована задача динамического поиска сетевых уязвимостей в виде задачи максимизации покрытия графа передачи управления.
4. Разработан метод динамического поиска сетевых уязвимостей на основе максимизации покрытия графа передачи управления с использованием генетического алгоритма, и получены оценки оптимальных значений параметров генетического алгоритма для поиска сетевых уязвимостей.
5. Построена модель масштабирования динамического поиска сетевых уязвимостей, основанного на использовании генетических алгоритмов, обеспечивающая эффективное использование вычислительных ресурсов многопроцессорного кластера при поиске уязвимостей.
6. Разработаны архитектура и экспериментальный макет высокопараллельной системы поиска сетевых уязвимостей на основе генетических алгоритмов на многопроцессорном кластере, и проведена оценка эффективности его работы.

Перспективы дальнейшей разработки темы диссертации заключаются в применении разработанных методов, моделей и алгоритмов для построения систем анализа безопасности сетевых подсистем ПО, а также в расширении области применения разработанного метода динамического поиска сетевых уязвимостей на основе генетического алгоритма и модели масштабирования динамического поиска сетевых уязвимостей на другие классы уязвимостей.

Список работ, опубликованных автором по теме диссертации:

1. Печенкин, А. И. Моделирование высокоскоростной параллельной обработки сетевого трафика на многопроцессорном кластере [Текст] / А. И. Печенкин, Д. С. Лаврова // Журнал "Проблемы информационной безопасности. Компьютерные системы". — СПб.: Изд-во Политехн. ун-та, 2012. — №4. — С. 33-39.

2. Печенкин, А. И. Параллельный анализ безопасности сетевого трафика на многопроцессорном кластере [Текст] / А. И. Печенкин, Д. С. Лаврова // Журнал "Проблемы информационной безопасности. Компьютерные системы". — СПб.: Изд-во Политехн. ун-та, 2013. — №1. — С. 55-62.

3. Печенкин, А. И. Архитектура масштабируемой системы фаззинга сетевых протоколов на многопроцессорном кластере [Текст] / А. И. Печенкин, А. В. Никольский // Журнал "Проблемы информационной безопасности. Компьютерные системы". — СПб.: Изд-во Политехн. ун-та, 2013. — №1. — С. 63-72.

4. Печенкин, А. И. Применение генетических алгоритмов для поиска уязвимостей в сетевых протоколах методом фаззинга/ А. И. Печенкин // Журнал "Системы высокой доступности". — М.: Изд-во Радиотехника, 2013. — №3. — С. 63-70.

5. Печенкин, А. И. Моделирование поиска уязвимостей методом фаззинга с использованием автоматного представления сетевых протоколов [Текст] / А. И. Печенкин, Д. С. Лаврова // Журнал "Проблемы информационной безопасности. Компьютерные системы". — СПб.: Изд-во Политехн. ун-та, 2013. — №2. — С. 59-67.

6. Печенкин, А. И. Безопасность АСУ ТП энергосистем, использующих промышленные протоколы передачи данных [Текст] / П. Д. Зегжда, Т. В. Степанова, А. И. Печенкин // Журнал "Известия

Российской Академии Наук. Энергетика". — М.: Изд-во Наука, 2013. — №5. — С. 59-64.

7. Печенкин, А. И. Анализ достоверности результатов поиска уязвимостей в программных продуктах [Текст] / А. И. Печенкин // Сб. материалов 19-й научно-технической конференции "Методы и технические средства обеспечения безопасности информации". — СПб.: Изд-во Политехн. ун-та, 2010. — С. 123-124.

8. Печенкин, А. И. Мониторинг сетевой активности вредоносного программного обеспечения [Текст] / Д. А. Москвин, А. И. Печенкин // Сб. материалов XII Санкт-Петербургской международной конференции "Региональная информатика". — СПб.: Изд-во СПИИРАН, 2010. — С. 125-126.

9. Печенкин, А. И. Универсальная платформа распределенных вычислений на базе АРМ пользователей сети Интернет, облако своими руками [Текст] / А. И. Печенкин // Сб. материалов 20-й научно-технической конференции "Методы и технические средства обеспечения безопасности информации". — СПб.: Изд-во Политехн. ун-та, 2011. — С.148-149.

10. Печенкин, А. И. Обнаружение несанкционированной сетевой активности вредоносного программного обеспечения [Текст] / А. И. Печенкин // Материалы VII межрегиональной конференции "Информационная безопасность регионов России". — СПб.: СПОИСУ, 2011. — С.87-88.

11. Печенкин, А. И. Построение облачных вычислений на базе АРМ пользователей сети Интернет [Текст] / А. И. Печенкин // Материалы VII межрегиональной конференции "Информационная безопасность регионов России". — СПб.: СПОИСУ, 2011. — 87 с.

12. Печенкин, А. И. Применение генетических алгоритмов для повышения эффективности поиска уязвимостей в системном и прикладном ПО [Текст] / Д. А. Москвин, А. И. Печенкин, Д. В. Мельницкий // Сб. материалов 21-й конференции "Методы и технические средства обеспечения безопасности информации". — СПб.: Изд-во Политехн. ун-та, 2012. — С. 165-167.

13. Печенкин, А. И. Применение генетических алгоритмов для повышения эффективности поиска уязвимостей в системном и прикладном ПО [Текст] / А. И. Печенкин, В. А. Мацуев // Сб. материалов 21-й научно-технической конференции "Методы и технические средства обеспечения безопасности информации". — СПб.: Изд-во Политехн. ун-та, 2012. — С. 71-73.