Инструмент динамического анализа помеченных данных «Блесна»

Краткое руководство пользователя

1. Описание инструмента «Блесна»

Инструмент динамического анализа помеченных данных «Блесна» предназначен для поиска утечек чувствительных данных, например, пользовательских паролей, по трассам выполнения программ в соответствии с требованиями действующей редакции методики выявления уязвимостей и недекларированных возможностей в программном обеспечении. Трассы содержат выполненный код не только анализируемого объекта оценки, но и среды функционирования, включая код других выполнившихся процессов, что позволяет выполнять полносистемный анализ.

Основные функциональные возможности инструмента «Блесна».

1. В инструменте реализован высокоточный анализ потоков данных, который основан на классических алгоритмах теории компиляторов. Полнота контроля потоков данных позволяет избегать пропусков утечек, реализуемых через особенности работы процессора. «Блесна» учитывает особенности целевой процессорной архитектуры, такие как механизм виртуальной памяти, аппаратное переключение контекста при обработке прерываний и исключений, прямой доступ устройств к памяти (DMA).
2. Инструмент реализует полносистемный анализ, покрывающий все слои ПО программно-аппаратной платформы, как приложения, так и системный код. Анализ всего ПО ведется совместно, с учётом явных и побочных связей между программами.
3. Модульная архитектура инструмента «Блесна» позволяет разрабатывать и комбинировать различные виды анализа в виде отдельных реализованных компонентов. Инструмент использует единое промежуточное представление бинарного кода, не зависящее от целевой процессорной архитектуры. Эта особенность позволяет разрабатывать модули анализа таким образом, чтобы один модуль был пригоден для анализа кода сразу всех поддерживаемых средой процессорных архитектур.
4. Реализована поддержка процессорной архитектуры x86 / x86-64; поддерживаются ОС семейств Windows и Linux. По запросу заказчика может быть добавлена поддержка других процессорных архитектур.

5. Подавляющее большинство алгоритмов анализа, реализованных в инструменте «Блесна», распараллелены на общей памяти и показывают на современных многоядерных рабочих станциях масштабируемость, близкую к линейной.

Системные требования к инструменту «Блесна»: ОС Linux x86-64, центральный процессор с числом ядер не менее 4, ОЗУ не менее 16 Гбайт, рекомендуется не менее 2 ТБ дискового пространства.

Инструмент разрабатывается на языках программирования: C, C++, Assembler x86-64, при разработке инструмента «Блесна» используются: графическая библиотека Qt версии 5.15 (распространяется под лицензией LGPL version 3), СУБД SQLite версии 3.36 (распространяется как Public domain), Программный комплекс, обеспечивающий декодирование машинных команд процессорных архитектур x86 и x86-64 quix86 (собственная разработка ИСП РАН, свидетельство о регистрации программы для ЭВМ 2017610624 от 16.01.2017 г.).

2. Лицензирование и установка

Для скачивания дистрибутива пройдите по ссылке https://getbox.ispras.ru/index.php/s/E5bSbvFjG0FrEYv и скачайте файлы lure.protected_1.2-0.deb и aksusbd_8.11-1_amd64.deb. Для запуска тестового примера потребуется также скачать многотомный архив pamtest.zip (файлы pamtest.zip.001, pamtest.zip.002, pamtest.zip.003).

Распространяемый дистрибутив инструмента «Блесна» защищен с помощью аппаратного ключа. По вопросам определения стоимости, приобретения и использования, обращайтесь по адресу lure@ispras.ru.

2.1 Системные требования

Для установки дистрибутива требуется операционная система семейства Linux 64 бит, поддерживающая графическую библиотеку Qt5 и работу с deb пакетами, например Ubuntu 18. Дистрибутив состоит из рабочего окружения для работы аппаратного ключа, устанавливаемого из пакета aksusbd_8.11-1_amd64.deb, и программы инструмента «Блесна», устанавливаемой из пакета lure.protected_1.2-0.deb.

2.2 Установка

К примеру, для установки этих программ из командной строки в операционной системе Ubuntu 18 нужно с помощью команды cd перейти в каталог, содержащий

файлы aksusbd_8.11-1_amd64.deb и lure.protected_1.2-0.deb, а затем нужно выполнить команды:

```
sudo dpkg -i aksusbd_8.11-1_amd64.deb
sudo dpkg -i lure.protected_1.2-0.deb
```

Программа инструмента «Блесна» будет установлена в каталог /usr/local/bin и состоит из 3-х файлов: файл приложения Lure, перевода программы на русский язык Lure-ru.qm, архива машины архитектуры x86-64 lure.architecture.x86.pcma.

Для удаления инструмента «Блесна» и рабочего окружения аппаратного ключа нужно выполнить команды:

```
sudo dpkg -r lure
sudo dpkg -r aksusbd
```

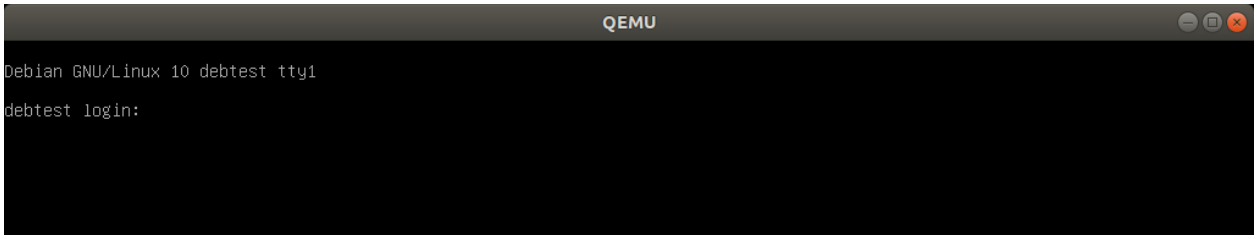
3. Примеры запуска инструмента «Блесна»

Для запуска тестового примера распакуйте многотомный архив pamtest.zip в домашней директории, например /home/user/Desktop/pamtest командами:

```
cat pamtest.zip.00* > pamtest.zip
unzip pamtest.zip
```

Работа с инструментом «Блесна», будет продемонстрирована на примере анализа работы тестового Linux PAM модуля (Pluggable Authentication Modules for Linux), в который внедрена ошибка CWE – 316 (https://cwe.mitre.org/data/definitions/316.html). Данный модуль участвует в аутентификации пользователя в операционной системе Linux, получает и выводит на экран пароль пользователя, который задан пользователем для входа в ОС, копирует его в памяти и не удаляет скопированный пароль должным образом.

Тестовый PAM модуль был добавлен в образ виртуального диска. Затем этот образ использовался для запуска ОС Debian в эмуляторе QEMU, поддерживающего детерминированное воспроизведение выполнения программ. Результат выполнения исследуемого образа диска в эмуляторе QEMU сохраняется в виде трассы. В данном примере запись трассы была начата с момента начала аутентификации пользователя в системе и до вывода приглашения ввода новой команды в терминале.



На рисунке ниже продемонстрирована работа исследуемого PAM модуля и вывод им пароля пользователя на экран. В данном примере использовалось имя пользователя user и пользовательский пароль `qwe`12345`.

Снятая трасса находится в каталоге trace, модули выполненных в трассе программ в каталоге modules (включая тестовый PAM модуль pam_my32.so), используемые модели функции в файле FunctionModels.xml, образ ядра Linux в файле vmlinux.

В прилагаемых моделях функций описана модель стандартной системной функции Linux pam_get_item с сигнатурой:

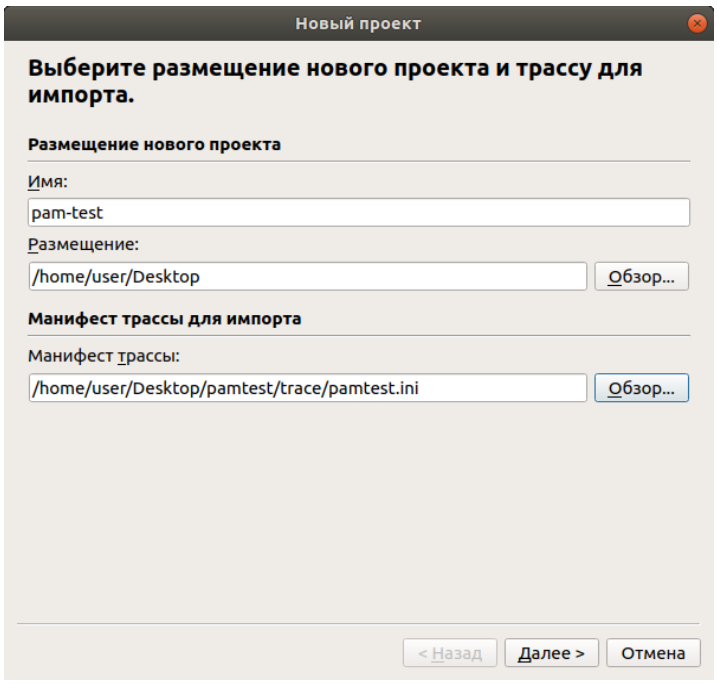
```
int pam_get_item(const pam_handle_t *pamh, int item_type, const void
**item);
```

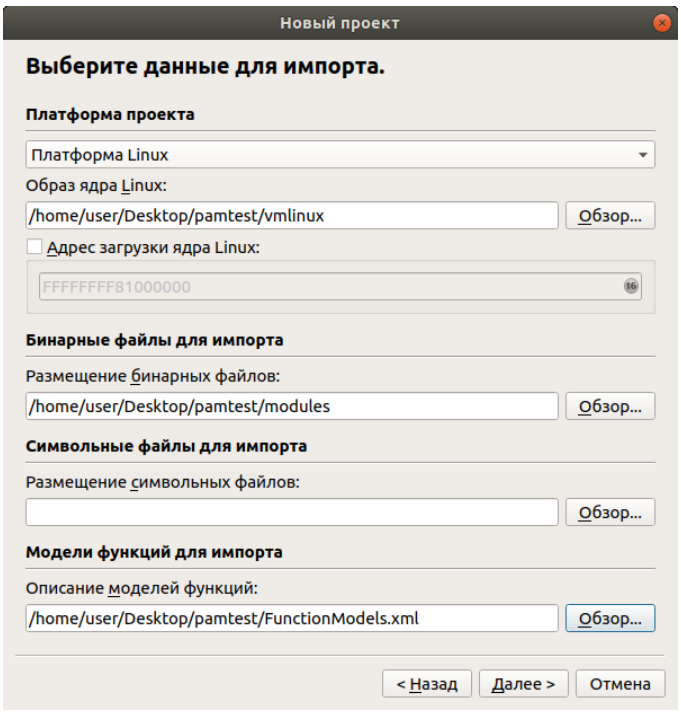
С помощью этой функции в зависимости от значения параметра функции item_type могут быть получены различные данные, используемые при аутентификации пользователей, которые возвращаются через параметр функции item. В модели функции параметрам функции item_type и item соответствуют входные параметры модели item_type и item_ptr_ptr. Параметр модели item_ptr, который получается разыменованием параметра item_ptr_ptr, является выходным. Параметр модели item, который получается разыменованием параметра item_ptr, является выходным и чувствительным, так как он может содержать чувствительные данные.

В данном примере тестовый PAM модуль pam_my32.so через параметр item функции pam_get_item получает пароль пользователя, который выводится на экран, а затем копируется в памяти.

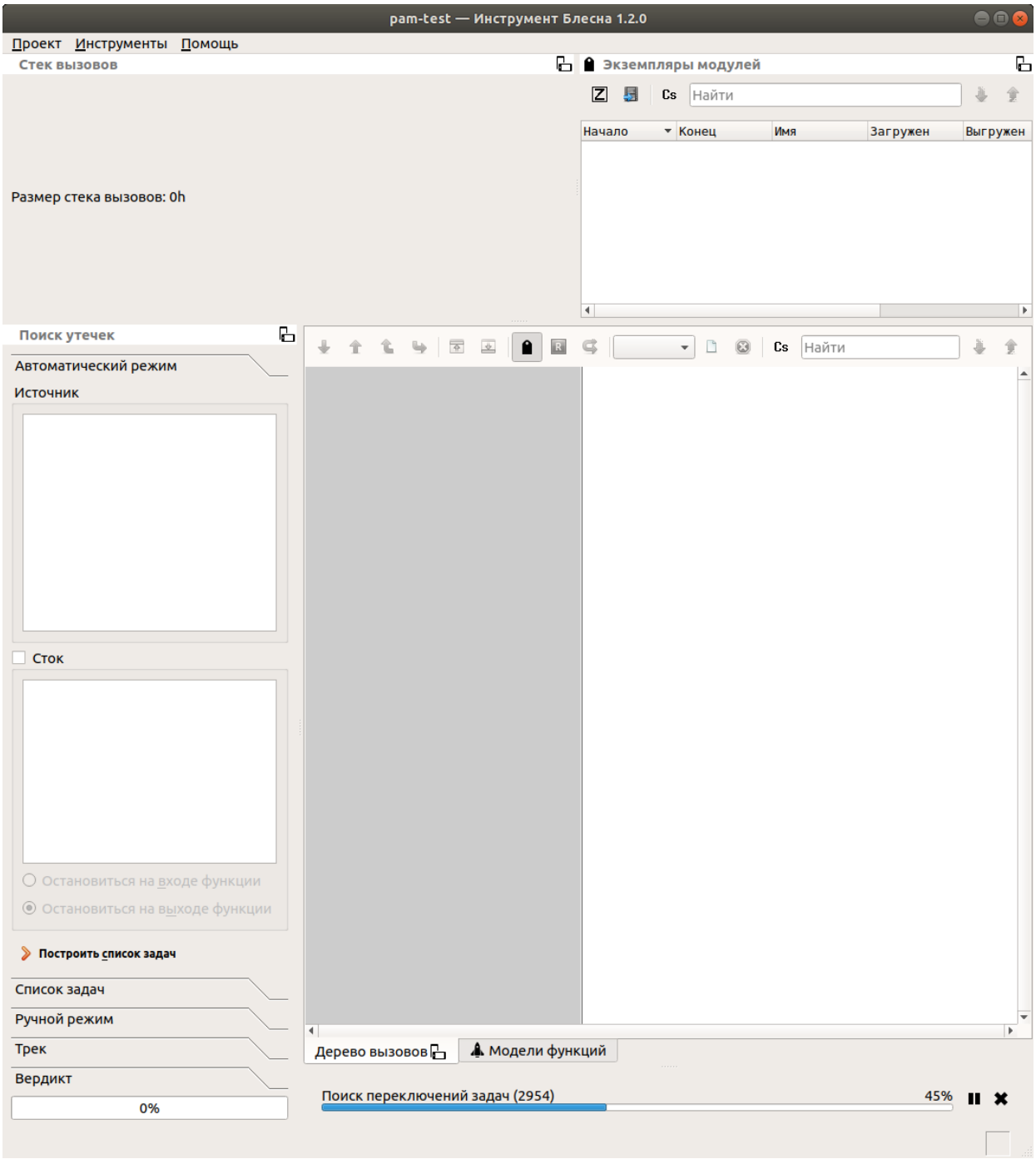
На основе приложенных данных создадим проект в инструменте «Блесна», в предположении, что все указанные выше каталоги и файлы тестового проекта

находятся в каталоге /home/user/Desktop/pamtest. После запуска программы выбираем пункты меню Проект –> Новый проект. В результате запустится мастер создания нового проекта:

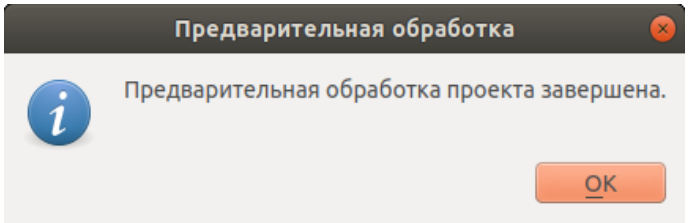




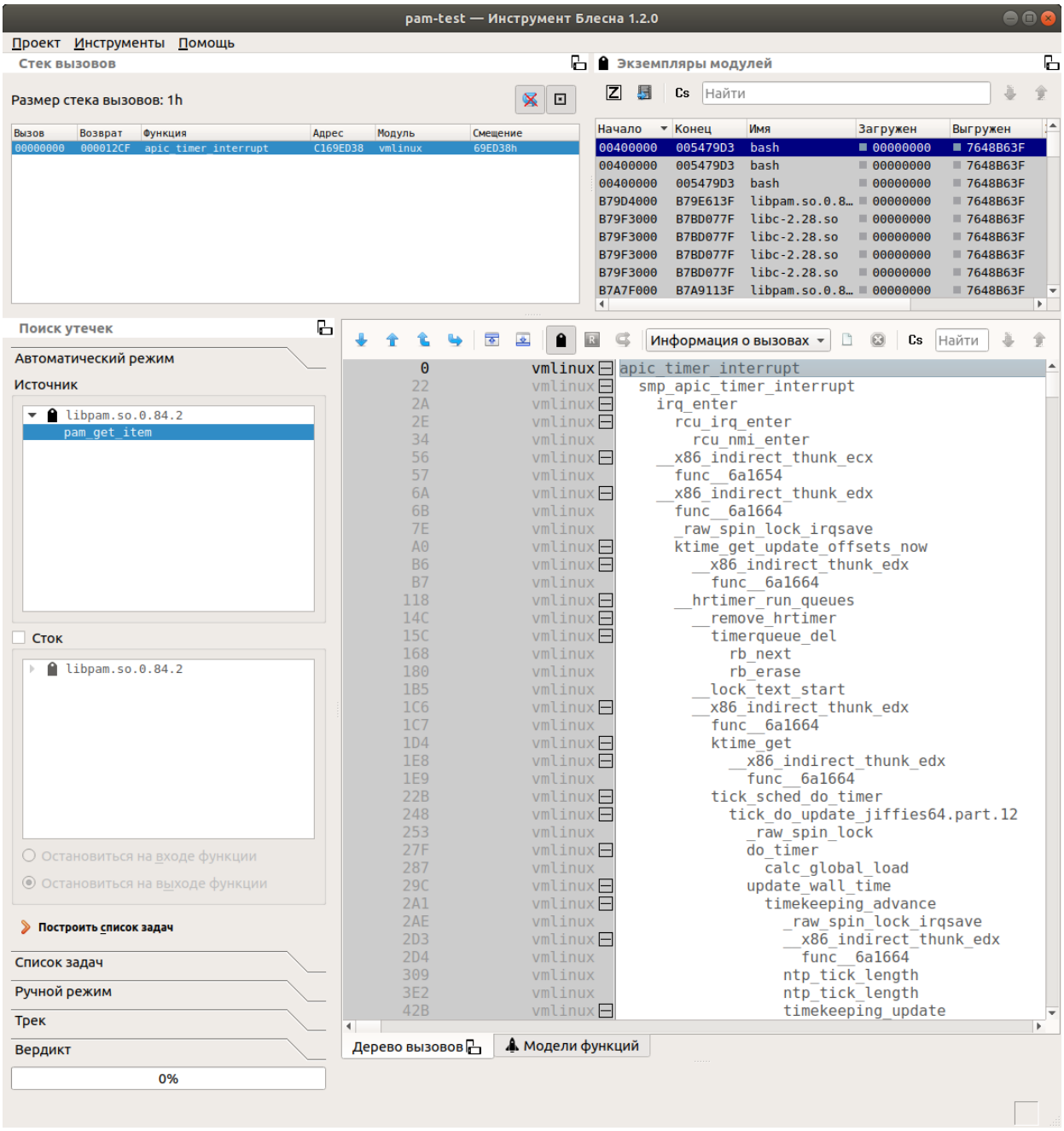
После создания проекта этот проект будет автоматически открыт и запустится предобработка проекта. Данная операция выполняется в многопоточном режиме и может занимать все доступные вычислительные ресурсы машины.



После завершения предобработки проекта появится сообщение:



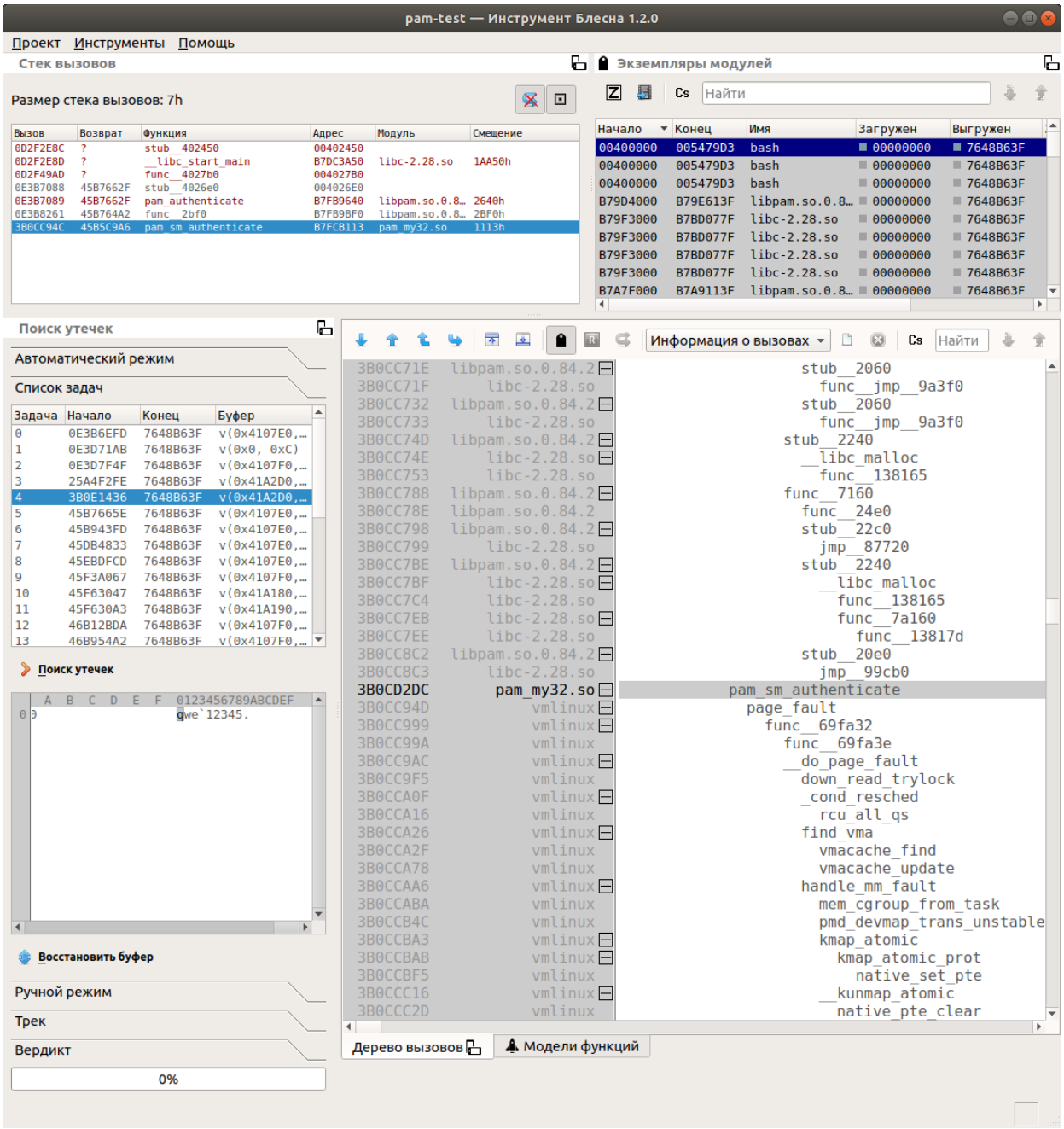
Для проверки утечки чувствительных данных в окне «Поиск утечек» выбираем в качестве функции источника функцию pam_get_item, а затем нажмем на кнопку «Построить список задач».



В результате для каждого из найденных в трассе вызовов функции pam_get_item будет определен буфер, соответствующий параметру модели item и содержащий чувствительные данные, а также диапазон шагов, на которых этот буфер будет отлеживаться. В данном примере это будут шаги от момента завершения вызова функции pam_get_item и до конца трассы. Пара <буфер, диапазон шагов> формирует задачу.

По двойному щелчку левой кнопки мыши на задаче в Списке задач мы можем перейти на вызов функции, который соответствует начальному шагу задачи. Для задачи с номером 4 это будет вызов функции в модуле pam_my32.so. С помощью кнопки «Восстановить буфер» можно восстановить значение буфера выбранной

задачи. Например, восстановив буфер для задачи с номером 4, мы увидим, что он содержит пользовательский пароль `qwe`12345`.



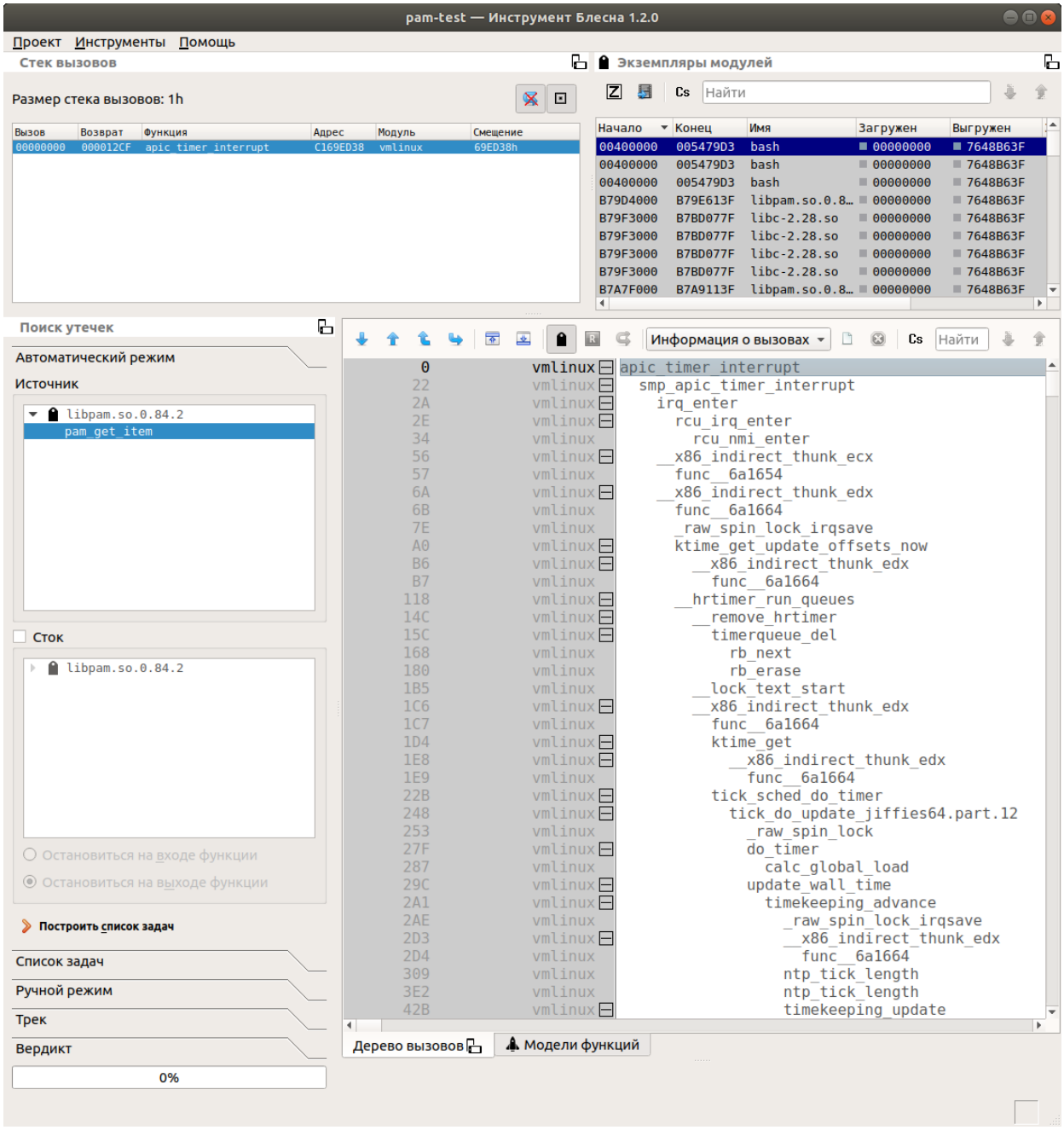
4. Графический интерфейс инструмента «Блесна»

Главное меню программы состоит из следующих пунктов:

1. Проект:
   1) Новый проект – для создания нового проекта анализируемой программы.
   2) Открыть проект – для открытия уже существующих проектов.
   3) Закрыть проект – закрывает текущий проект.
   4) Недавние проекты – отображает 9 последних открытых проектов.

5) Выйти из инструмента Блесна – закрывает программу «Блесна».

2. Инструменты:
   1) Предварительная обработка – выполняет повторный запуск алгоритмов предварительной обработки проекта.
   2) Сбросить проект – выполняет сброс данных алгоритмов предварительной обработки проекта.
3. Помощь:
   1) О Блесне – показывает диалоговое окно с информацией об инструменте «Блесна».



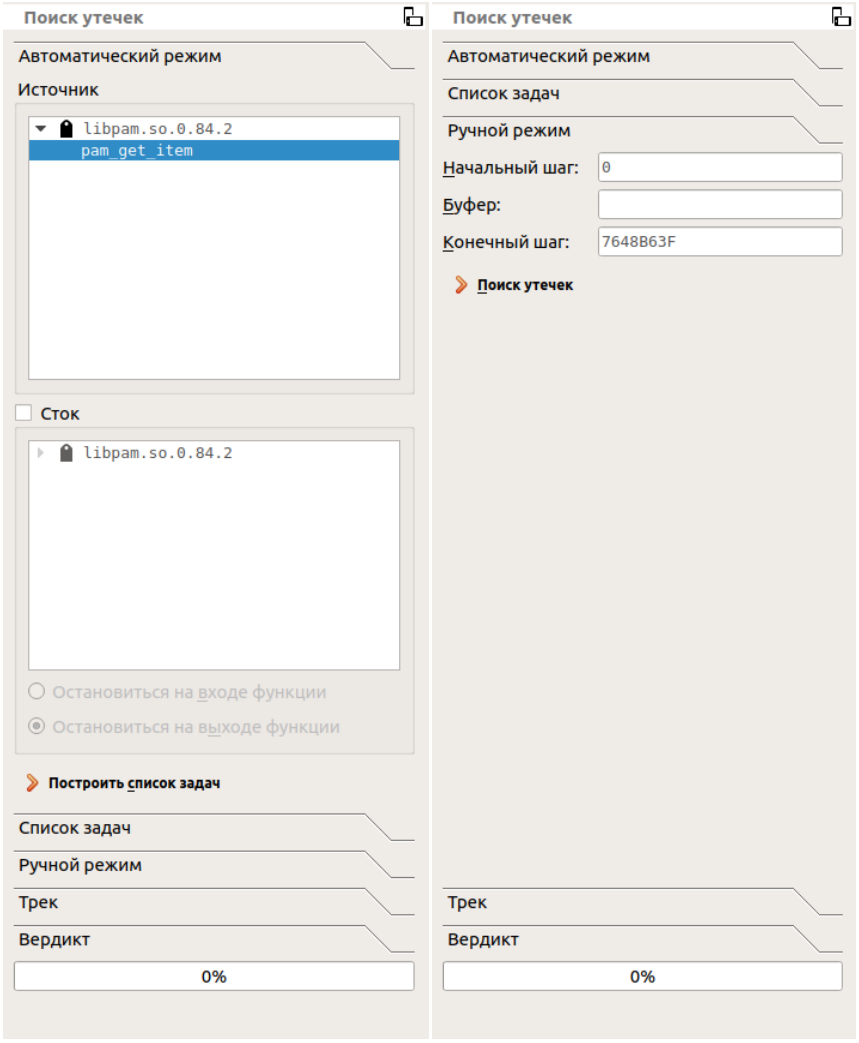
Работа с проектом в инструменте «Блесна» состоит из следующих этапов:

1. После выбора пункта меню «Новый проект» будет запущен мастер создания нового проекта, который на основе описанных в пункте 3 данных создает новый проект.
2. После создания проект автоматически открывается и выполняется предварительная обработка созданного проекта, в ходе которой происходит запуск различных алгоритмов анализа, которые выполнят следующие операции:
   1) Определение точек переключения контекстов процессов, потоков.
   2) Разбиение трассы по процессам, потокам, адресным пространствам.
   3) Обработка прерываний в трассе.
   4) Построение информации о вызовах в трассе.
   5) Построение информации о функциях в трассе.
   6) Анализ обращений в память.
   7) Построение высокоуровневого представления СМП.
   8) Распознавание модулей в трассе.
   9) Импортирование моделей функций.
3. После завершения обработки проекта станет доступна различная информация о трассе проекта и можно приступить к анализу трассы. Для этого используются следующие окна графического интерфейса:
   1) Стек вызовов – для отображения стека вызовов текущей функции в трассе.
   2) Экземпляры модулей – для отображения модулей программ, найденных в трассе.
   3) Поиск утечек – для поиска утечек в анализируемой трассе программы.
   4) Дерево вызовов – для отображения последовательности вызовов в трассе.
   5) Модели функций – для работы с моделями функций в трассе.

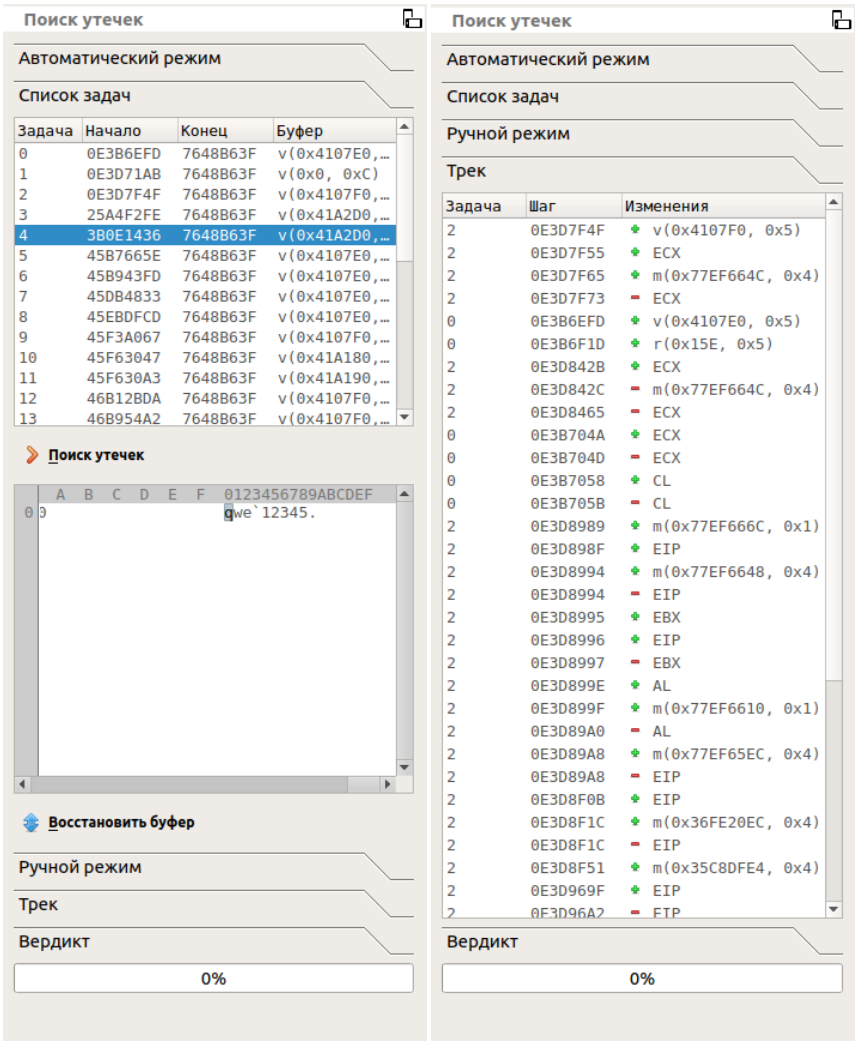
Пользователь может осуществлять поиск утечек чувствительных данных в 2-х режимах: автоматическом и ручном.

В автоматическом режиме пользователь выбирает функцию источник чувствительных данных. В качестве чувствительных данных будут использованы входные и выходные параметры выбранной функции, у которых установлен флаг Чувствительный (Sensitive) в модели этой функции. Затем пользователь выбирает, до какого момента будет происходить отслеживание чувствительных данных. Если галочка «Сток» выключена, то чувствительные данные будут

отслеживаться от шага начала вызова функции источника для входных чувствительных данных и от шага завершения вызова функции источника для выходных чувствительных данных и до конца трассы. Если галочка «Сток» включена, то пользователь должен выбрать функцию сток и параметр «Остановиться на входе функции» или «Остановиться на выходе функции». Если выбран параметр «Остановиться на входе функции», то чувствительные данные будут отслеживаться от шага начала вызова функции источника для входных чувствительных данных и от шага завершения вызова функции источника для выходных чувствительных данных и до шага в трассе, на котором произошел первый после вызова функции источника вызов функции стока. Если выбран параметр «Остановиться на выходе функции», то чувствительные данные будут отслеживаться до шага в трассе, на котором завершился первый после вызова функции источника вызов функции стока. Если вызова функции стока после окончания вызова функции источника в трассе нет, то чувствительные данные будут отслеживаться до конца трассы.



После завершения выбора функции источника и функции стока нужно нажать на кнопку «Построить список задач». В результате откроется вкладка «Список задач» и будет сформирован пронумерованный список задач по отслеживанию чувствительных данных. Для каждого вызова функции источника определяются буферы, в которых содержатся данные чувствительных параметров функции источника, а так же диапазон шагов на котором происходит отслеживание чувствительных данных. Пара буфер – диапазон шагов формирует задачу на отслеживание чувствительных данных. Для выбранной задачи становится активной кнопка «Восстановить буфер», нажатие на которую запускает алгоритм восстановления содержимого буфера выбранной задачи. Также по двойному щелчку левой кнопки мыши по выбранной задаче можно выполнить переход на вызов функции, соответствующий начальному шагу задачи.



На вкладке «Ручной режим» пользователь может самостоятельно задать буфер с чувствительными данными и диапазон шагов для отслеживания. В этом случае будет отслеживаться только заданный буфер.

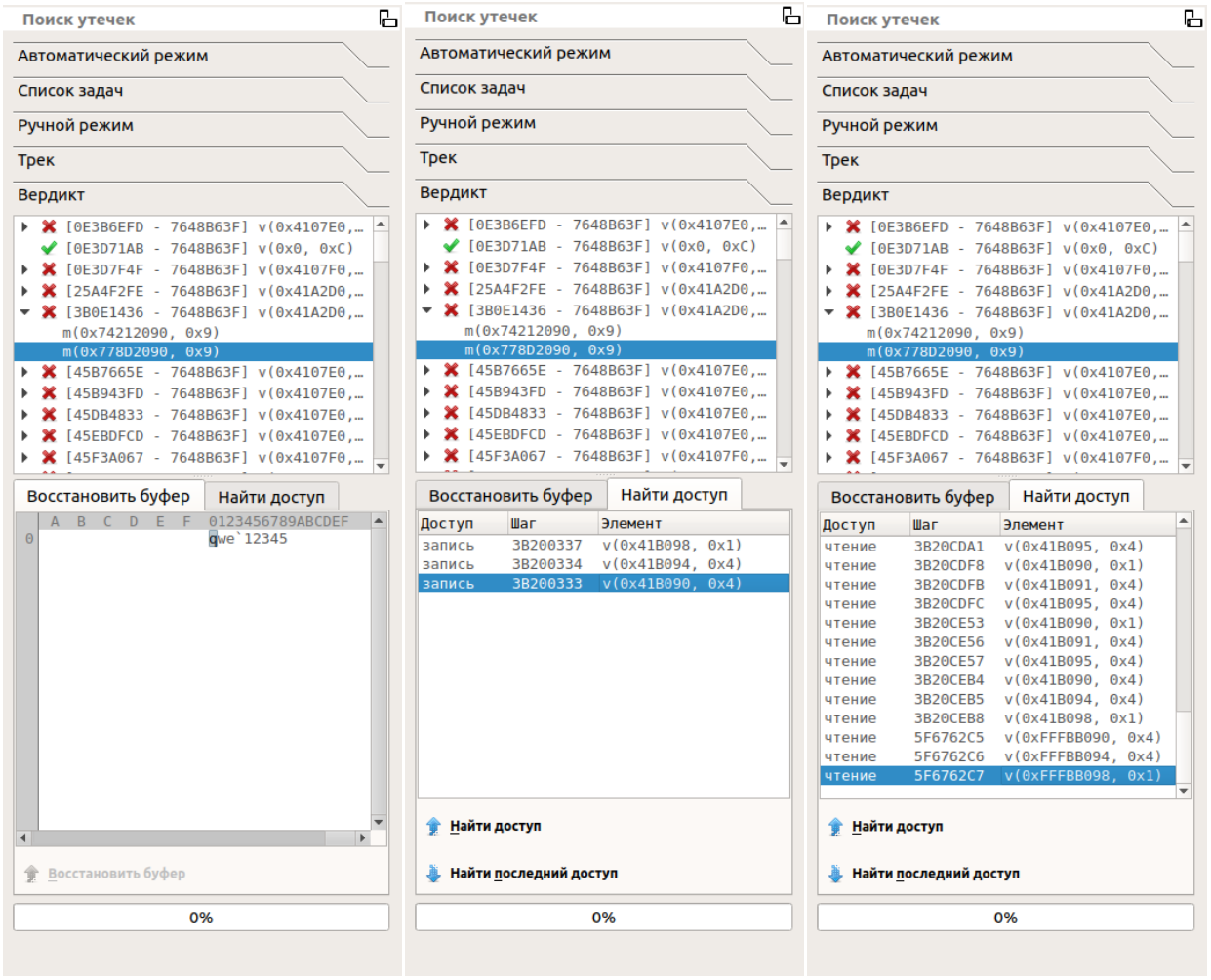
После формирования списка задач в автоматическом режиме или задания буфера и диапазона шагов в ручном режиме нужно нажать на кнопку «Поиск утечек». В результате откроется вкладка «Трек» и произойдет запуск алгоритма анализа для сформированных задач. Так же с помощью пункта «Запустить в ручном режиме» контекстного меню в списке задач можно запустить анализ только для выбранной задачи.

На вкладке «Трек» можно отследить работу алгоритма анализа. Промежуточный результат алгоритма анализа выдается в виде тройки <Задача, Шаг, Изменение> для автоматического режима или пары <Шаг, Изменение> для ручного режима. Задача указывает на номер задачи, которая выполняется алгоритмом анализа, Шаг указывает на номер шага в трассе, на котором произошло изменение отслеживаемого множества чувствительных данных. Иконкой «+» помечаются новые элементы, которые добавлены в отслеживаемое множество, а иконкой «-» помечаются элементы, которые убраны из отслеживаемого множества. По двойному щелчку мыши на этих элементах можно перейти на шаг в трассе, на котором произошло выбранное изменение отслеживаемого множества, и увидеть соответствующий этому шагу вызов функции в дереве вызовов/стеке вызовов.

После завершения работы алгоритма анализа произойдет переключение на вкладку «Вердикт», на которой можно посмотреть результат. Если после завершения прохода алгоритма анализа по диапазону шагов задачи отслеживаемое множество пусто, то утечка данных из отслеживаемого в задаче буфера не обнаружена. Такие задачи помечаются иконкой «V». Если же отслеживаемое множество задачи оказывается не пустым, то такая задача помечается иконкой «X» и элементы, составляющие отслеживаемое множество на момент завершения работы алгоритма анализа, добавляются к этой задаче.

Пользователь может выбрать такой элемент, а затем нажать на кнопку «Восстановить буфер» на вкладке «Восстановить буфер» – в результате будет запущен алгоритм восстановления содержимого буфера выбранного элемента. По двойному щелчку мыши на выбранном элементе либо нажав кнопку «Найти доступ» на вкладке «Найти доступ» пользователь может запустить алгоритм поиска доступа на запись в выбранный элемент начиная с последнего шага задачи вверх по трассе. Результат поиска выдается в виде тройки: тип доступа, шаг на котором произошел доступ, элемент к которому произошел доступ. По двойному щелчку по результату поиска можно перейти на шаг в трассе, где

произошел доступ, и увидеть соответствующий этому шагу вызов функции в дереве вызовов/стеке вызовов.



Нажав на кнопку «Найти последний доступ» пользователь может запустить алгоритм поиска доступа на чтение/запись в выбранный элемент начиная с последнего шага в трассе, который будет найден в ходе вызова «Найти доступ», и вниз по трассе, до тех пор, пока содержимое выбранного элемента не будет полностью перезаписано.

На вкладке «Вердикт» пользователю отображаются элементы, в которые произошла утечка чувствительных данных. С помощью восстановления буфера, он может проверить какие данные содержатся в этих элементах. Выбрав интересующие его элементы, пользователь с помощью вкладки «Трек» может определить на каких шагах трассы произошло добавление этих элементов в отслеживаемое множество. А по двойному щелчку мыши на элементах вкладки «Трек» найти вызов функции в окнах «Дерева вызовов», «Стек вызовов», в котором произошел доступ к выбранному элементу.

5. Описание процессов, обеспечивающих поддержание жизненного цикла ПО

5.1 Процессы разработки и совершенствования ПО

Разработка инструмента динамического анализа помеченных данных «Блесна» ведется по методологии Agile с привлечением современных средств повышения качества кода.

1. Для хранения кода используется система контроля версий git, и изменения в основной ветке проходят инспекцию кода (code review) другими разработчиками.
2. Для проверки работоспособности системы созданы тесты, причем используются как модульные тесты, проверяющие функционал отдельных компонентов, так и интеграционное тестирование, при котором проверяется корректность работы системы при типичных вариантах использования.
3. При разработке используется практика непрерывной интеграции (continuous integration): при помощи сервера тестирования Jenkins происходят регулярные автоматические сборки с последующим запуском тестов, упомянутых в предыдущем пункте.
4. При написании кода разработчики должны придерживаться строгих правил оформления кода; нарушение этих правил не позволит пройти этап инспекции кода.
5. Периодически производится проверка кода среды на предмет наличия ошибок статическим анализатором Svace, разработанным в ИСП РАН.

5.2 Поддержка пользователей ПО

Пользователи, которым достаточно для решения их задач возможностей, уже реализованных в инструменте «Блесна», могут приступать к работе с инструментом после краткого обучения, проводимого на стороне разработчика (ИСП РАН). В ходе своей работы инструмент «Блесна», автоматически ведёт журналы действий и ошибок, которые могут быть отправлены разработчику в случае обнаружения некорректного поведения или возникновения запроса на улучшение и доработку.

Помимо этого, ИСП РАН предлагает разработку специализированных модулей анализа на своей стороне по запросу заказчика.

Информация о сбоях в работе инструмента «Блесна», проблемах производительности, ошибках целевого функционала передаются

пользователями среды непосредственно ответственным сотрудникам ИСП РАН, без использования публичных Интернет-ресурсов управления ошибками. Это обеспечивает должный уровень конфиденциальности для контрольных примеров (фрагменты исследуемых динамических профилей, снимки памяти), передаваемых пользователем инструмента в ИСП РАН для оценки и исправления программного дефекта.

Со своей стороны ИСП РАН постоянно совершенствует разрабатываемый инструмент, применяя в жизненном цикле разработки передовые методики. Добавление новых и улучшение существующих алгоритмов ведется в инициативном порядке. Обновления инструмента «Блесна» передаются пользователям среды через согласованные с ними каналы распространения обновлений.

5.3 Необходимый персонал для разработки и поддержки

Для разработки и поддержки программного продукта необходима соответствующая квалификация разработчиков. Это вызвано следующими причинами.

1. Специфическая предметная область, требующая глубоких знаний одновременно в нескольких областях: устройство современной аппаратуры и операционных систем, компиляторные технологии, компьютерная безопасность, технологии разработки ПО.
2. Наличие среди алгоритмов, реализованных в среде анализа, алгоритмов, впервые разработанных сотрудниками ИСП РАН.
3. Требования к производительности системы, из-за чего необходимо применять эффективные алгоритмы, в т.ч. хорошо масштабируемые по нескольким вычислительным ядрам.

В силу приведенных причин коллектив разработчиков инструмента «Блесна» формируется из специалистов, получивших профильное образование: выпускников ВМК МГУ (магистерская программа «Компиляторные технологии») и ФУПМ МФТИ (магистерская программа «Системное программирование»).

Для гарантийного обслуживания задействовано 3 научных сотрудника, для Технической поддержки задействованы 3 научных сотрудника, для модернизации программного обеспечения задействованы 5 научных сотрудников.

Адрес электронной почты, по которому можно обратиться по вопросам, связанным с инструментом динамического анализа помеченных данных «Блесна» — lure@ispras.ru.