

Инструмент динамического анализа помеченных данных «Блесна»

Краткое руководство пользователя

Данный документ или его копии не может распространяться (полностью или частично) в любом формате без письменного разрешения ИСП РАН.

1. Описание инструмента «Блесна»

Инструмент динамического анализа помеченных данных «Блесна» предназначен чувствительных для поиска утечек 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).
- 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.
- 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 <u>lure@ispras.ru</u>.

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 РАМ модуля (Pluggable Authentication Modules for CWE Linux), который внедрена ошибка 316 в (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 РАМ модуль был добавлен в образ виртуального диска. Затем этот образ использовался OC QEMU. для запуска Debian в эмуляторе поддерживающего детерминированное воспроизведение выполнения программ. исследуемого образа диска Результат выполнения 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.

QEMU

На рисунке ниже продемонстрирована работа исследуемого РАМ 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 используемые item type могут быть получены различные данные, 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 чувствительным, как ОН может содержать И так чувствительные данные.

В данном примере тестовый РАМ 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:

Новый проект	8	Новый проект	8
Выберите размещение нового проекта и трасс импорта.	у для	Выберите данные для импорта.	
Размещение нового проекта			
Имя:		Платформа Linux	•
pam-test		/home/user/Desktop/pamtest/vmlinux	Q630D
Размещение:		Адрес загрузки ядра Linux:	
/home/user/Desktop	Обзор		
			(ie)
манифест трассы для импорта		Бинарные файлы для импорта	
	06	Размещение <u>б</u> инарных файлов:	
/nome/user/Desktop/pamtest/trace/pamtest.ini	<u>О</u> озор	/home/user/Desktop/pamtest/modules	<u>О</u> бзор
		Символьные файлы для импорта	
		Размешение символьных файлов:	
			Обзор
		Молели функций для импорта	
		/home/user/Desktop/pamtest/EuoctionModels.xml	06200
		noncouser, beskep, pancescy ancelon models. And	<u>o</u> osop
Далее	Отмена	< Назал Ла	лее > Отмена
	Новый	й проект 🛛 😣	
Сводка по проект Будет создан сле запустится его пр	у дующий новый прое редобработка.	кт. Затем этот проект будет открыт и	
Навигатор прое	кта		
★ pam-test ▼ □ pamtest ■ pamtest	:		

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.

<<u>Н</u>азад <u>З</u>авершить Отмена

	pam-test —	- Инструмент Бл	есна 1.2.0					
<u>П</u> роект <u>И</u> нструменты <u>П</u> омощь		_						_
Стек вызовов		l <u>i</u>	Экземг	іляры модулей				Ŀ
			Ζ 🔙	Сs Найти			÷	Ŷ
			Начало	▼ Конец	Имя	Загружен	Выгр	ужен
Decision and the second second								
Размер стека вызовов: Оп								
			4					•
Поиск утечек	↓ ↑ 1 ↓ ■		G	- 🗈 😣	Сs Найти		4	Ŷ
Автоматический режим								
Источник								
Ctor								
О Остановиться на <u>в</u> ходе функции								
Остановиться на выходе функции								
Построить список задач								
Список задач								
Ручной режим	4							v
Трек	Дерево вызовов 🗗	🛦 Модели функ	ций					
Вердикт								
0%	Поиск переключений	і задач (2954)				45%		×

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».

		F	oam-te	st — Ино	струмент	Блес	на 1.2.0	0					θ	
<u>П</u> роект <u>И</u> нструменты <u>П</u> омощь							_							_
Стек вызовов					le		Экзем	мпляры	мод	улей				Ċ
Размер стека вызовов: 1h					🔀 🖸		Ζ 🔙	Cs	Найти	1			¥	Î
Вызов Возврат Функция	Адрес	Модул	ь	Смещен	ие	Ha	чало	Конец 0.054	1	Имя	Загру	жен	Выгружен	
		JO VIITEIT	ux	092030		00 00 87 87 87	9400000 9400000 9D4000 9F3000 9F3000	0 0054 0 0054 0 0054 0 87950 0 87800 0 87800	79D3 79D3 613F 977F 977F	bash bash libpam.so libc-2.28 libc-2.28	000 000 000 0.8 000 .so 000 .so 000	00000 00000 00000 00000 00000	7648B63 7648B63 7648B63 7648B63 7648B63 7648B63	F F F F
						B7	9F3000	9 B7BD	977F	libc-2.28	.so ≡000 .so ≡000	00000	7648863 7648863	F
						B7	A7F006	B7A9	113F	libpam.so	.0.8 = 000	00000	■ 7648B63	F 🔻
Поиск утечек	B													•
		1	<u>î</u> <u></u>	•	2	R	G I	информ	ация	о вызовах	- 🗋 🚳	Cs H	айти 🎍	Î
Автоматическии режим			0		vmlin	nux⊟	apic	_timer	_int	terrupt				
Источник			22		vmlin		SM	<pre>p_apic irg_or</pre>	tir: tor:	mer_inter	rupt			
▼ ▲ libpam.so.0.84.2			2E		vmlin		1	rcu	irq	enter				
pam_get_item			34		vmlin	nux		rc	u_n	ni_enter				
			56		vmlin	iux 🖃]	x86_	indi	irect_thu	nk_ecx			
			57		vmlin	nux nux 🗖	1	×86	ind:	1054 irect thu	nk odv			
			6B		vmlin		1	func	68	a1664	IIIK_CUX			
			7E		vmlin	nux		raw	spi	in lock i	rqsave			
			A0		vmlin	iux 🖃]	ktim	ie_ge	et_update	_offsets	now		
			B6		vmlin	iux 🗕	1		x86	_indirect	_thunk_e	dx		
		1	B/ 18		vmlin	nux nux 🗖	1	hr	TUNC time	c0a1004				
		1	.4C		vmlin		1		remo	ove hrtim	er			
Сток		1	.5C		vmlin	iux 🖃	j		time	erqueue_d	el			
		1	.68		vmlin	nux			rt	_next				
Libpam.so.0.84.2		1	.80 DE		vmlin	IUX			rt	o_erase	art			
		1	.DD (6		vmlin	iux iiix 🗖	1		 	indirect	thunk er	dx		
		1	.C7		vmlin	iux	1		fund	c 6a1664	_chank_c	ax		
		1	.D4		vmlin	nux 🖃]		ktir	ne_get				
		1	.E8		vmlin	iux 🗆]			_x86_indi	rect_thu	nk_edx		
		1	.E9		vmlin		1		+ i cl	func_6a	1664 o timor			
		2	48		vmlin		1		+	ick do un	date iif	fies64	.nart.12	
		2	53		vmlin	iux	1			raw spi	n lock	120001	-poi ci zz	
О Остановиться на входе функции		2	7F		vmlin	nux 🖃]			do_timer	_			
		2	87		vmlin	nux	1			calc_g	lobal_loa	ad		
Становиться на выходе функции		2	Ω1		vmlin		1			update_w	all_time ening adv	vance		
		2	AE		vmlin	iux	1			raw	spin lo	ck irq	save	
		2	D3		vmlin	iux 🖃]			x8	6_indire	ct_thu	nk_edx	
Список задач		2	2D4		vmlin	nux				fu	nc6a16	64		
Ручной режим			169 162		vmlin	nux				ntp_	tick_len	gth ath		
Трек		4	2B		vmlin	iux E]			time	keeping_	update		•
Верликт		Дерево в	зызово	8 6 4	• Модели	1 функ	ций							•
бердикт					-	+)								
0%														
													[

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.

					1	pam-t	est -	— Ин	стру	мент	Бл	іесна 1	.2.0						e	
<u>П</u> роект <u>И</u>	<u>И</u> нструме	нты <u>П</u> омощь																		_
Стек выз	30B0B									ĥ	ò	🗎 Эк	земг	пляры мод	улей					E
Размер сте	ека вызов	зов: 7h							8			Ζ	5	Ся Найт	И				4	È
Вызов	Возврат	Функция	Адрес		Модул	ь		Смещен	ние			Начал	0	🕶 Конец	Имя		Загруж	ен	Выгружен	A
0D2F2E8C 0D2F2E8D	? ?	stub402450 libc start main	00402 B7DC3	2450 3A50	libc-	2.28.	50	14450	h			00400	0000	005479D3	bash		0000	0000	7648B6	3F PE
0D2F49AD	?	func4027b0	00402	27B0								00400	0000	00547903	bash		0000	0000	7648B6	3F
0E3B7088	45B7662F	pam_authenticate	B7FB9	9640	libpa	m.so.(9.8	2640h				B79D4	000	B79E613F	libpa	am.so.0.	8 = 0000	9000	7648B6	3F
0E3B8261 3B0CC94C	45B764A2 45B5C9A6	func2bf0 pam_sm_authenticate	B7FB9 B7FCE	9BF0 3113	libpa pam n	am.so.(9.8 o	2BF0h 1113h				B79F3	000	B7BD077F	libc-	2.28.so	0000	9999	7648B6	3F
		·										B79F3	000	B7BD077F	libc-	2.28.50	0000	0000	7648B6	3F
												B79F3	0000	B7BD077F	libc-	2.28.50		9000	7648B6	3F 3F
												B7A7F	000	B7A9113F	libpa	am.so.0.	8 = 0000	9000	■ 7648B6	3F 🔻
_												•								•
Поиск ут	гечек			€	1	1	L	\$	*			R	N	нформация	о вызо	вах 👻		Cs	айти 🌖	主主
Автомати	ческий р	ежим	_	3B	30CC7	71E	lib	opam	.S0.	0.84	1.2	$2 \square$				stub	2060			
Список за	адач		_		80CC7	71F	1	li	bc-2	2.28.	SO					fu	nc_jmp	_9a3	f0	
Задача На	ачало	Конец Буфер			30CC7	/32 /33	lib	pam 1i	.50. hc-2		+.2 50					stub	2060	9a3	f0	
0 08	E3B6EFD	7648B63F v(0x4107E0,	•		30CC7	74D	lib	maga	. SO.	0.84	1.2	2 El				stub	2240		10	
1 0E	E3D71AB E3D7E4E	7648B63F V(0x0, 0xC) 7648B63F V(0x4107F0			80CC7	74E		li	bc-2	2.28.	so					_li	bc_mall	C		
3 25	5A4F2FE	7648B63F v(0x41A2D0,			30CC7	753		li	bc-2	2.28.	SO					fu	nc1381	165		
4 3E	B0E1436	7648B63F v(0x41A2D0,					lib		. 50.	0.84	+.2 1 2					Tunc	2400			
6 45	5B943FD	7648B63F v(0x4107E0,				798	lib		. 50.	0.84	1.2					stub	24e0 22c0			
7 45	5DB4833	7648B63F v(0x4107E0,			0CC7	799		li	bc-2	2.28.	SO)				jm	p 87720	9		
8 45	5EBDFCD	7648B63F v(0x4107E0, 7648B63E v(0x4107E0,	•		30CC7		lib	opam	.so.	0.84	1.2	$2 \square$				stub	2240			
10 45	5F63047	7648B63F v(0x4107F0,			80CC7	/BF		11	bc-2	2.28.							libc_ma	ll0C		
11 45	5F630A3	7648B63F v(0x41A190,				7EB		li	bc-2	2.20.	50	Â					func 7a	a160		
12 46	6B12BDA 6B954A2	7648B63F V(0x410/F0, 7648B63F V(0x4107F0	•		30CC7	/EE		li	bc-2	2.28.	SO)					func	1381	7d	
. -						3C2	lib	opam	.so.	0.84	1.2	2 🖂				stub	20e0			
Поиск ў	утечек			38	30008	303		li	bc-2	2.28.	SO				0.000	jm	p99cb	9		-
A B	C D E	F 0123456789ABCDEF		38		94D		p	all_ll	ny⊃∠. /mlin	. 50 111X			μ	nage	fault	niticate			
0 9		g we 12345.				999			V	/mlin	nux				fu	nc 69	fa32			
						99A			V	/mlin	nux	<				func	69fa3e			
						AC			V	/mlin	านx					do_p	age_fau	lt	le.	
					80CC2	10F			V	/mtin /mlin	iux iux					con	_reau_t	-ytoc ad	К	
					30CCA	16			V	/mlin	nux	<				rc	u all q	5		
					BOCCA	126			V	/mlin	nux	< 🖂				find	vma			
					BOCCA	A2F			V	/mlin	านx	<				VM	acache_	find		
				30	80CCA	1/8				/mtin /mlin	iux uux					hand	lo mm f	updat ault	e	
					30CCA	ABA			V	/mlin	nux					me	m carou	o fro	m task	
•			•		BOCCE	34C			V	/mlin	านx	<				pm	d_devma	o_tra	ns_unsta	able
						BA3			V	/mlin	านx					km	ap_atom:	ic.		
Bocctar	новить оуф	eh			SOCCE	BE5			V	/mtin /mlin							native	> 20+	prot	
Ручной ре	ежим		_	38	00CC0	16			V	/mlin	nux	< 🖂					kunmap_a	atomi	_p c c	
Трек				3B ∢		C2D			V	/mlin	านx	<					native_p	ote_c	lear	The second secon
Вердикт				Де	рево	вызов	вов	5 .	Ам	одели	1φ	ункций	й							
		0%																		
L																				

4. Графический интерфейс инструмента «Блесна»

Главное меню программы состоит из следующих пунктов:

- 1. Проект:
 - 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.
 - Сбросить проект выполняет сброс данных алгоритмов предварительной обработки проекта.
- 3. Помощь:
 - О Блесне показывает диалоговое окно с информацией об инструменте «Блесна».

		pam-	test — Ин	струмент	г Блес	:на 1.2	.0				6	
<u>П</u> роект <u>И</u> нструменты <u>П</u> омощь												_
Стек вызовов						Экзе	емпл	пяры мод	улей			b
Размер стека вызовов: 1h				🔀 🖸		Ζ 5		Ся Найт	И			Ê
Bu3os Bo3spar Функция 00000000 000012CF apic timer_interrupt	Адрес C169ED38	Модуль vmlinux	Смеще 69ED3	ние 8h	Ha O O B B B B B B B B B B B B	ayano 040000 040000 79D400 79F300 79F300 79F300 79F300 79F300	- - - - - - - - - - - - - -	Конец 005479D3 005479D3 005479D3 B79E613F B7BD077F B7BD077F B7BD077F B7BD077F B7A9113F	Имя bash bash libpam.so. libc-2.28. libc-2.28. libc-2.28. libc-2.28. libc-2.28. libc-3.28.	3arpyxe 00000 00000 00000 00.8 00000 so 00000 so 00000 so 00000 so 00000 so 00000 0.8 00000	н Выгружен 000 7648863 000 7648863 000 7648863 000 7648863 000 7648863 000 7648863 000 7648863 000 7648863	▲ 3F 3F 3F 3F 3F 3F 3F 3F
Поиск утечек	6					et 1					P- U-Sev B	
Автоматический режим		TL	9 k	¥ [•	ĸ		инд	рормация	о вызовах 🔻		Ls Наити 🖉	X
Источник		0 22 24 26 34 56 57 6A 6B 7E A0 B6 B7 118 14C 15C 168 180 185 1C6 1C7 1D4 1E8 1E9 228 248 248		VML1 VMLi VMLi VMLi VMLi VMLi VMLi VMLi VMLi	nux EE nux E E nux E E nux E E nux E E nux E E E nux E E E E E E E E E E E E E E E E E E E		<u>c t</u> mp_ ir	1mer 1n apic_ti q_enter rcu_irq rcu_irq rcu_irq x86_ind func_6 raw_sp ktime_g time_time raw_sp trime_rem time time_rem time time_rem time time time time time time time ti	terrupt mer_intern _enter mi_enter irect_thur al654 irect_thur al664 in_lock_in et_update _indirect_ c_6al664 er_run_que ove_hrtime erqueue_d6 b_erase k_text_sta _indirect_ c_6al664 me_get _x86_indin func_6al k_sched_d0 ick_do_upc raw spir	rupt nk_ecx nk_edx offsets_n _thunk_edx eues er el art _thunk_edx hunk_edx hunk_edx hunk_edx ithunk_edx ithunk_edx ithunk_edx ithunk_edx ithunk_edx	ow _edx es64.part.12	2
О Остановиться на <u>в</u> ходе функции		27F		vmli	nux E	Ξ			do_timer	lobal load		
Остановиться на выходе функции		287		vmli	nux 🗆	-			update wa	all time		
Построить список задач		2A1 2AE 2D3		vmli vmli vmli	nux E nux nux E	=			timekee rawx80	eping_adva _spin_lock 5_indirect	nce _irqsave _thunk_edx	
Список задач	_	2D4 309		vmli vmli	nux				fur nto 1	nc6al664 tick lengt	h	
Ручной режим	_	3E2		vmli	nux	_			ntp_1	tick_lengt	h date	
Трек	•	428		VIIIC1	Hux				CTIIIG	veehrud_ub	uale	×
Вердикт	Д	ерево вызо	вов	Å Модел	и фун	кций						
0%											[

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.
- 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.
- После завершения обработки проекта станет доступна различная информация о трассе проекта и можно приступить к анализу трассы. Для этого используются следующие окна графического интерфейса:
 - Стек вызовов для отображения стека вызовов текущей функции в трассе.
 - Экземпляры модулей для отображения модулей программ, найденных в трассе.
 - 3) Поиск утечек для поиска утечек в анализируемой трассе программы.
 - 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 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.

Поиск утечек	6	Поиск утечек		Ē
Автоматический режим		Автоматический	режим	
Источник		Список задач		
✓ ▲ libpam.so.0.84.2		Ручной режим		
pam_get_item		<u>Н</u> ачальный шаг:	Θ	
		<u>Б</u> уфер:		
		<u>К</u> онечный шаг:	7648B63F	
		<u>П</u> оиск утечек		
Сток				
libpam.so.0.84.2				
О Остановиться на входе функции				
• Остановиться на выходе функции				
Построить <u>с</u> писок задач				
Список задач				
Ручной режим				
Трек		Трек		
Вердикт		Вердикт		
0%			0%	
·				

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.

Поиск	утечек			Поиск	утечек		F
Автома	тический р	режим		Автома	тический ре	жим	
Список	задач			Список	задач		
Задача	Начало	Конец	Буфер	Ручной	режим		
Θ	0E3B6EFD	7648B63F	v(0x4107E0,				
1	0E3D71AB	7648B63F	v(0x0, 0xC)	Трек			
2	0E3D7F4F	7648B63F	v(0x4107F0,	320202	lllar	Измонония	
3	25A4F2FE	7648B63F	v(0x41A2D0,	2	05307545	• v(0×4107E0_0×5)	
4	3B0E1436	7648B63F	v(0x41A2D0,	2	0E3D7F4F	 V(0x410/F0, 0x5) ECX 	
5	45B7665E	7648B63F	v(0x4107E0,	2	0E3D7F55	▼ EUX	
5	45B943FD	7648863F	V(0x410/E0,	2	0E3D7F65	■ M(0X//EF664C, 0X4)
/	450B4833	7648863F	V(0x410/E0,	2	0E3D7F73	- ECX	
8	45EBDFCD	7648803F	V(0X4107E0,	Θ	0E3B6EFD	v(0x4107E0, 0x5)	
10	45F5A007	7640003F	V(0X410/F0,	Θ	0E3B6F1D	 r(0x15E, 0x5) 	
11	45F63043	7648B03F	V(0X41A100,	2	0E3D842B	ECX	
12	45F050A5	7648B63F	v(0x41A190,	2	0E3D842C	 m(0x77EF664C, 0x4)
13	46B954A2	7648B63F	v(0x4107F0,	v 2	0E3D8465	- ECX	
10	10000 1112	10102031		0	0E3B704A	ECX	
> Пои	ск утечек			Θ	0E3B704D	- ECX	
					0E3B7058	♦ CL	
A	BCD	E F 01234	456789ABCDEF	• •	0E3B705B	= CI	
0 9		gwe	12345.	2	0E3D8989	★ m(0x77EE666C 0x1	
				2	02300305	• ETP	'
				2	02300301	• m(0x77556640 0x4	、
				2	0E3D0994	 III(0X//EF0048, 0X4 ETD 	'
				2	0E3D8994	= EIP	
				2	0E3D8995	EBX	
				2	0E3D8996	♥ EIP	
				2	0E3D8997	= EBX	
				2	0E3D899E	AL	
				2	0E3D899F	🕈 m(0x77EF6610, 0x1)
				2	0E3D89A0	- AL	
				2	0E3D89A8	 m(0x77EF65EC, 0x4)
4				2	0E3D89A8	= EIP	
4			P.	2	0E3D8F0B	• EIP	
🚖 Boco	тановить бу	ben		2	0E3D8E1C	m(0x36EE20EC 0x4)
<u> </u>				2	0E3D8E1C	= FTP	^
Рициой	DAWIAM			2	000000000000000000000000000000000000000	• m(0x35000000 0x4	、 III
гучной	PCMIM		<u> </u>	- 2	05200531	 m(0x35C00FE4, 0x4 ETD 	1
Трек				2	0E3D969F	* EIP	-
-					0F3D96A2	• FIP	
вердик	а		<u> </u>	Вердик	т		_
		0%				0%	
		0%				0%	

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.

Ha 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 «Х» 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

14

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.

- Для хранения кода используется система контроля версий git, и изменения в основной ветке проходят инспекцию кода (code review) другими разработчиками.
- 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.
- 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.
- 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.

- 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.
- 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» — <u>lure@ispras.ru</u>.