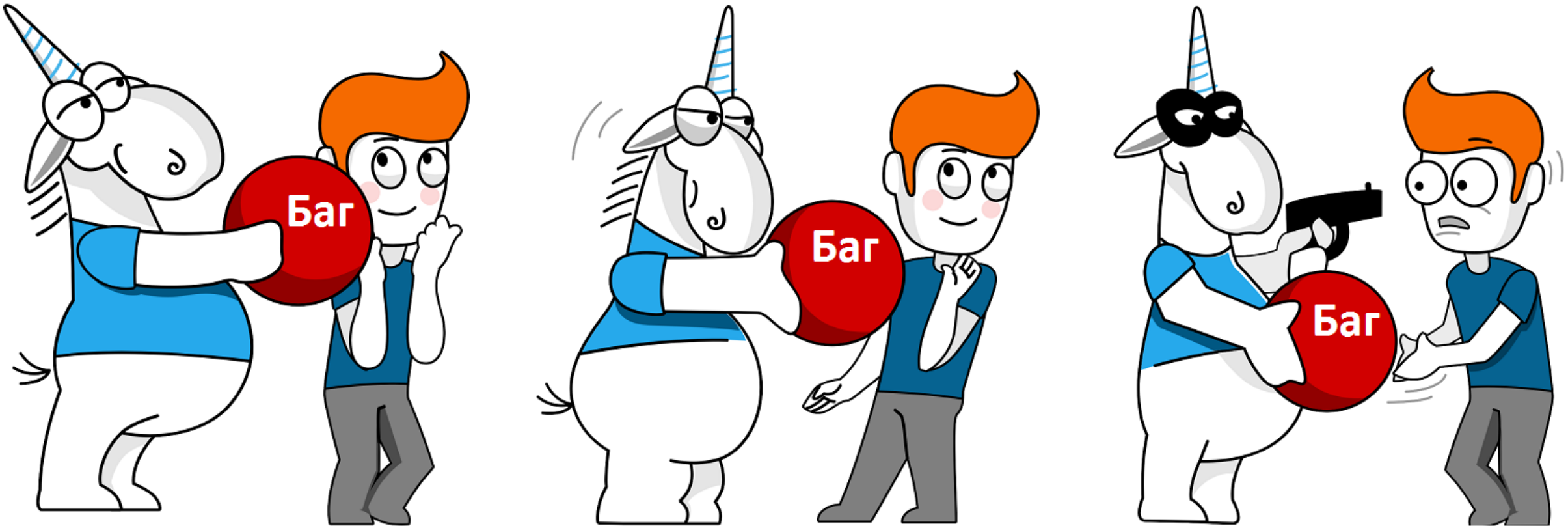
Что могут статические анализаторы, чего не могут программисты и тестировщики

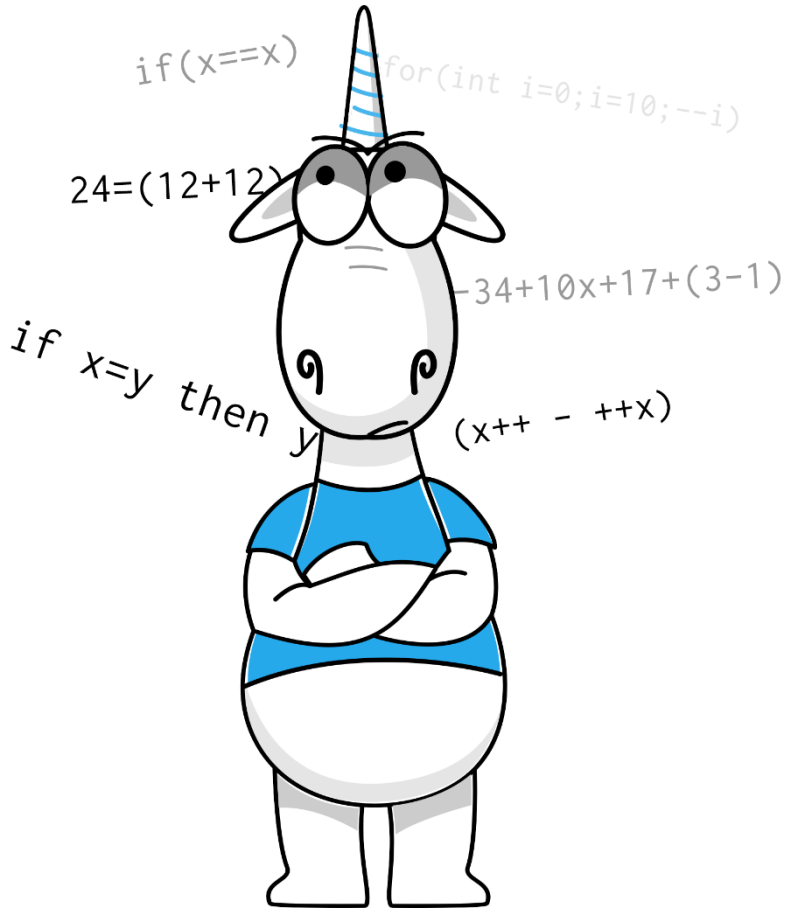
Андрей Карпов

karpov@viva64.com

Байки без пруфов: полёты



Байки без пруфов: бухгалтерия

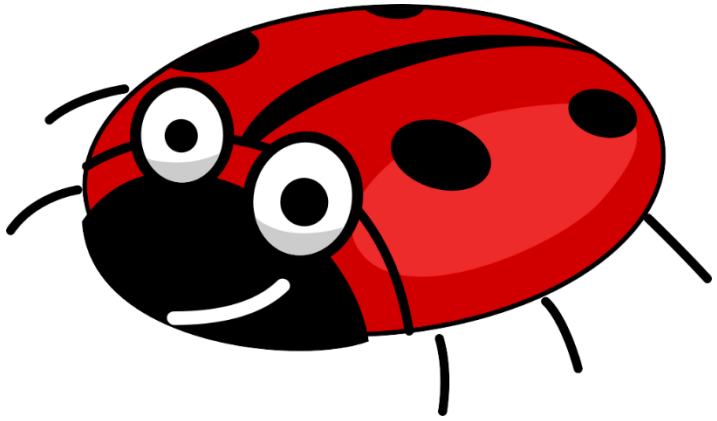


О докладчике

- Карпов Андрей Николаевич, 1981
- Присутствую на Habr под именем Andrey2008 habr.com/users/andrey2008/
- Технический директор ООО «СиПроВер»
- MVP в категории Visual C++
- Intel Black Belt Software Developer
- Один из основателей проекта PVS-Studio

Цель доклада: популяризация методологии статического анализа кода

- Название доклада – это гипербола
- Программисты и тестировщики могут находить описанные ошибки, но это неоправданно **долго, трудно, дорого**



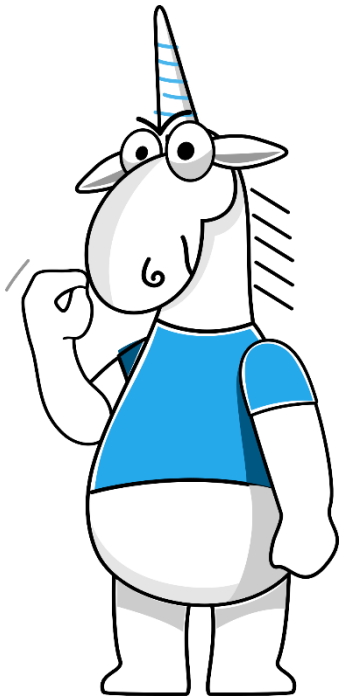
Обзор кода

Обзор кода vs Статический анализ кода

- Статический анализ – обзор кода, выполняемый программой
- БОЛЬШОЙ список анализаторов: https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

Почему статический анализ необходим?

- Выявляет многие ошибки на самом раннем этапе
- **Выявляет ошибки, которые другими методами найти очень, очень сложно**



Последуем за статическими анализаторами кода

Плохое хеширование

IronPython и IronRuby, C#

```
public static int __hash__(UInt64 x)
{
  int total = unchecked((int)(((uint)x) + (uint)(x >> 32)));
  if (x < 0)
  {
    return unchecked(-total);
  }
  return total;
}
```

Плохое хеширование

IronPython и IronRuby, C#

```
public static int __hash__(UInt64 x)
{
  int total = unchecked((int)(((uint)x) + (uint)(x >> 32)));
  if (x < 0)
  {
    return unchecked(-total);
  }
  return total;
}
```

PVS-Studio: V3022 Expression 'x < 0' is always false. Unsigned type value is always >= 0. IntOps.Generated.cs 1967

Плохое хеширование

Open Graph Drawing Framework, C++

```
template<> class DefHashFunc<void *> {
  public:
  size_t hash(const void * &key) const
    { return size_t(key && 0xffffffff); }
};
```

Плохое хеширование

Open Graph Drawing Framework, C++

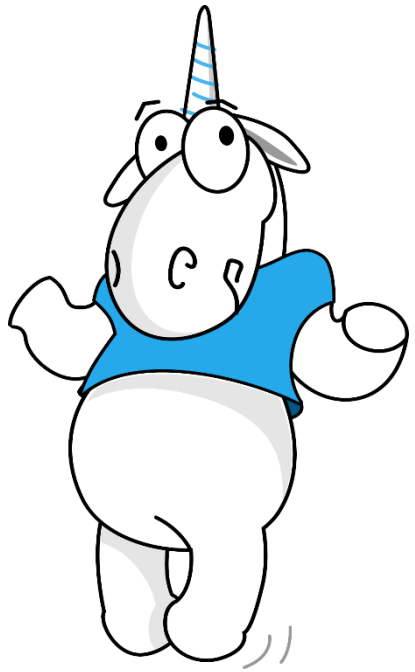
```
template<> class DefHashFunc<void *> {
  public:
  size_t hash(const void * &key) const
    { return size_t(key && 0xffffffff); }
};
```

& – побитовый AND / **&&** – логический AND

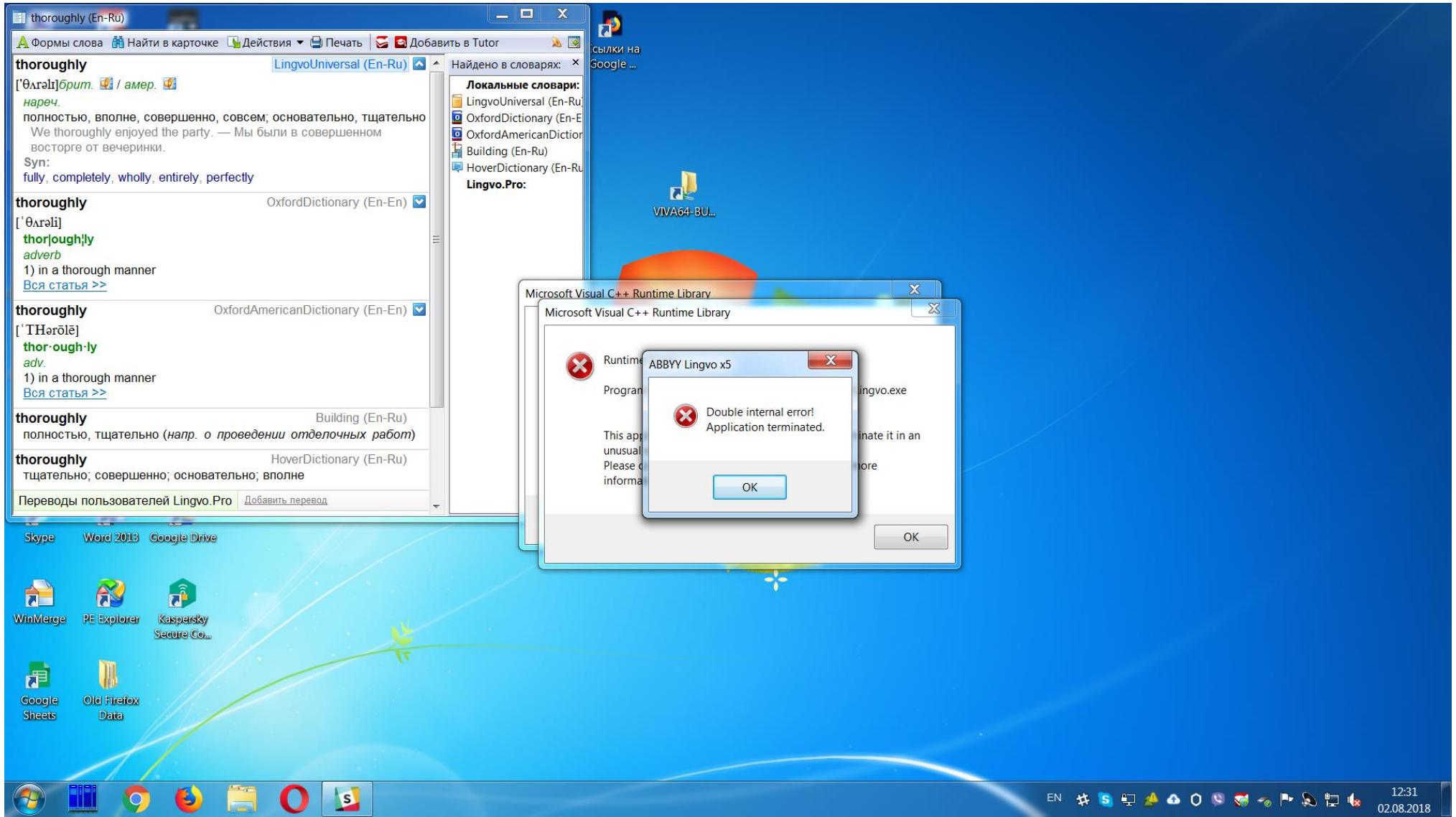
PVS-Studio: V560 A part of conditional expression is always true: 0xffffffff. hashing.h 255

Плохое хеширование

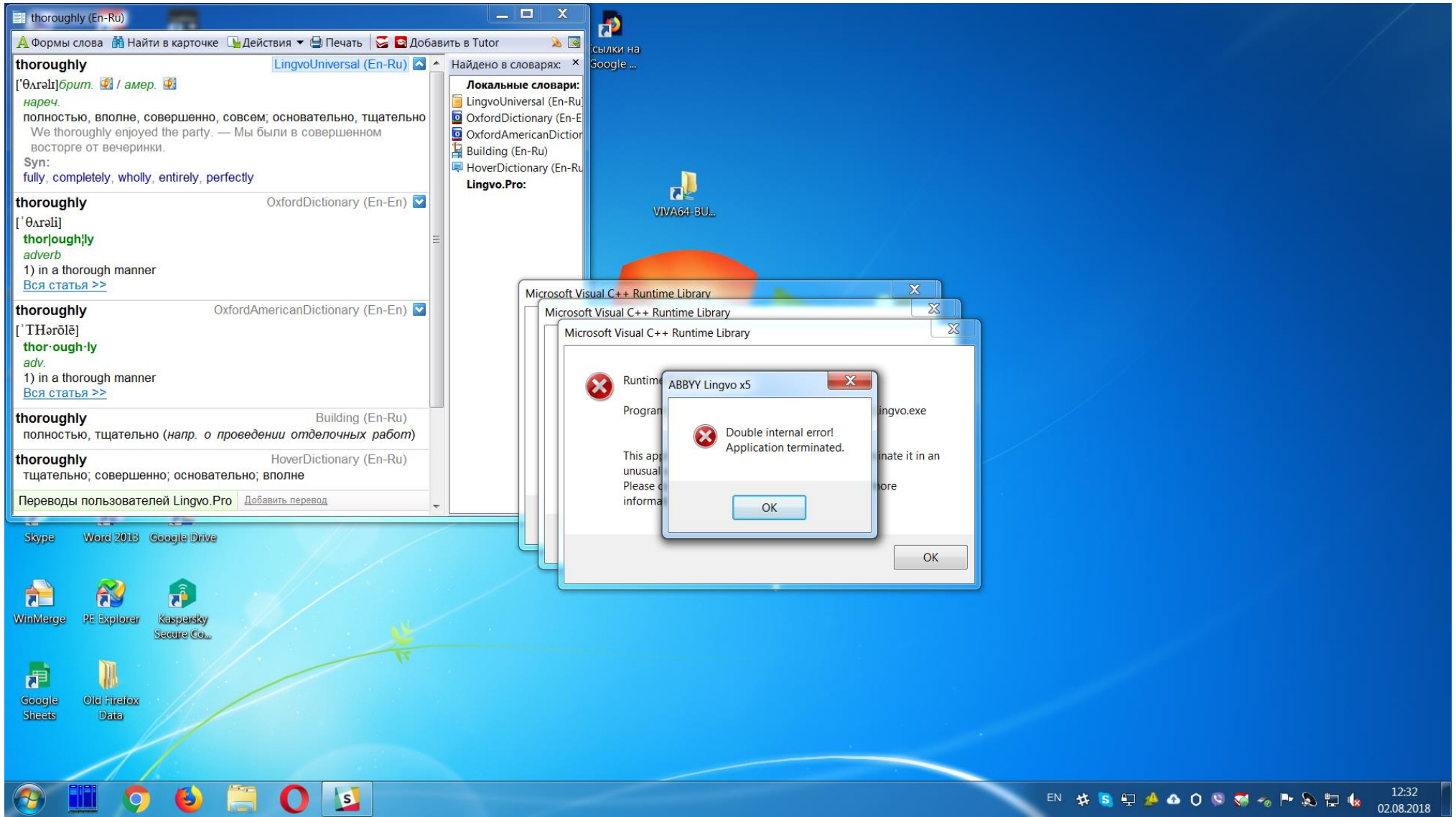
- При плохом хеше всё работает корректно, но неэффективно
- Непонятно, как обнаружить, если специально не искать замедленную работу
- Что и с чем сравнивать?



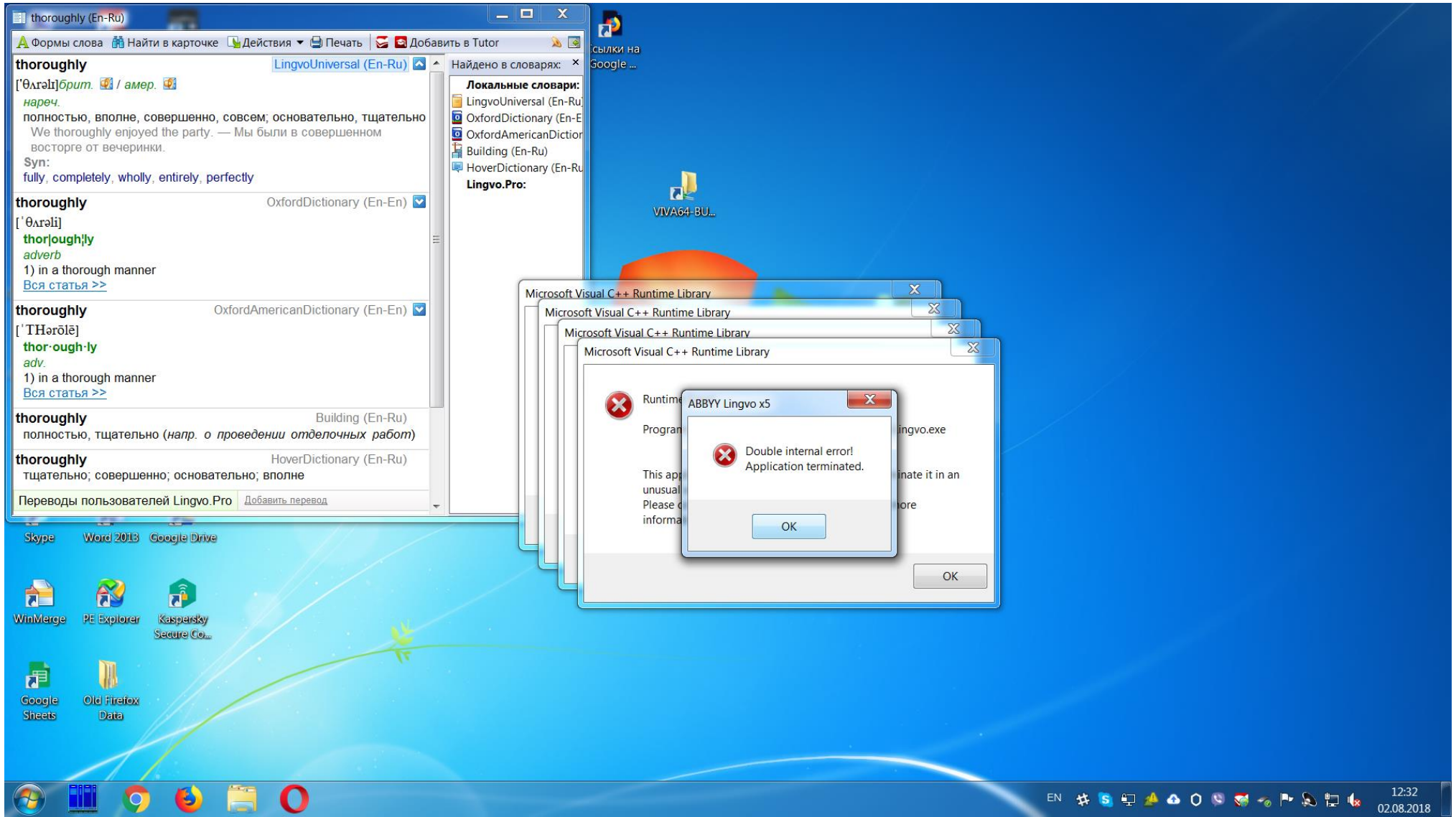
Ошибки в обработчиках ошибок



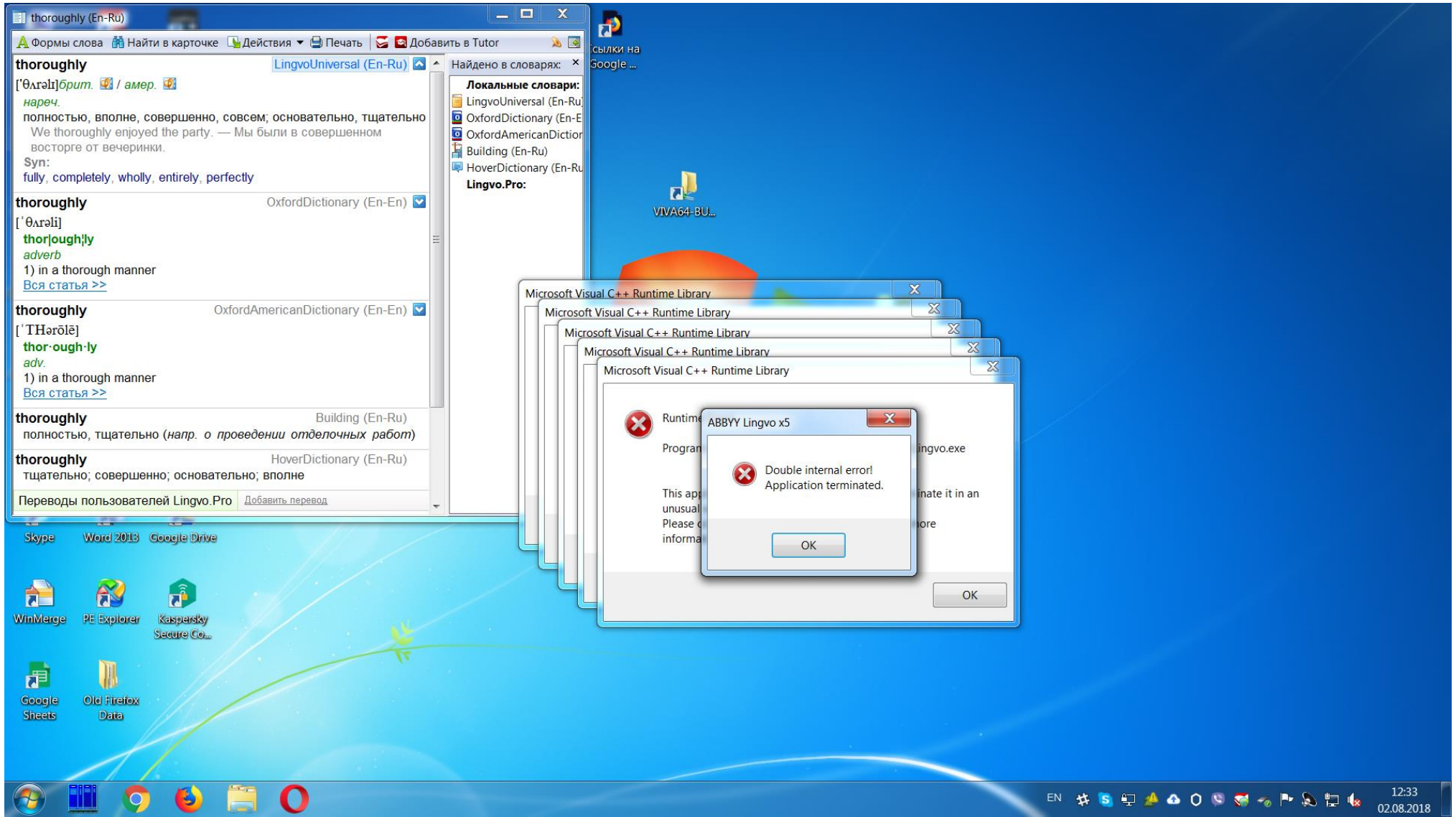
Ошибки в обработчиках ошибок



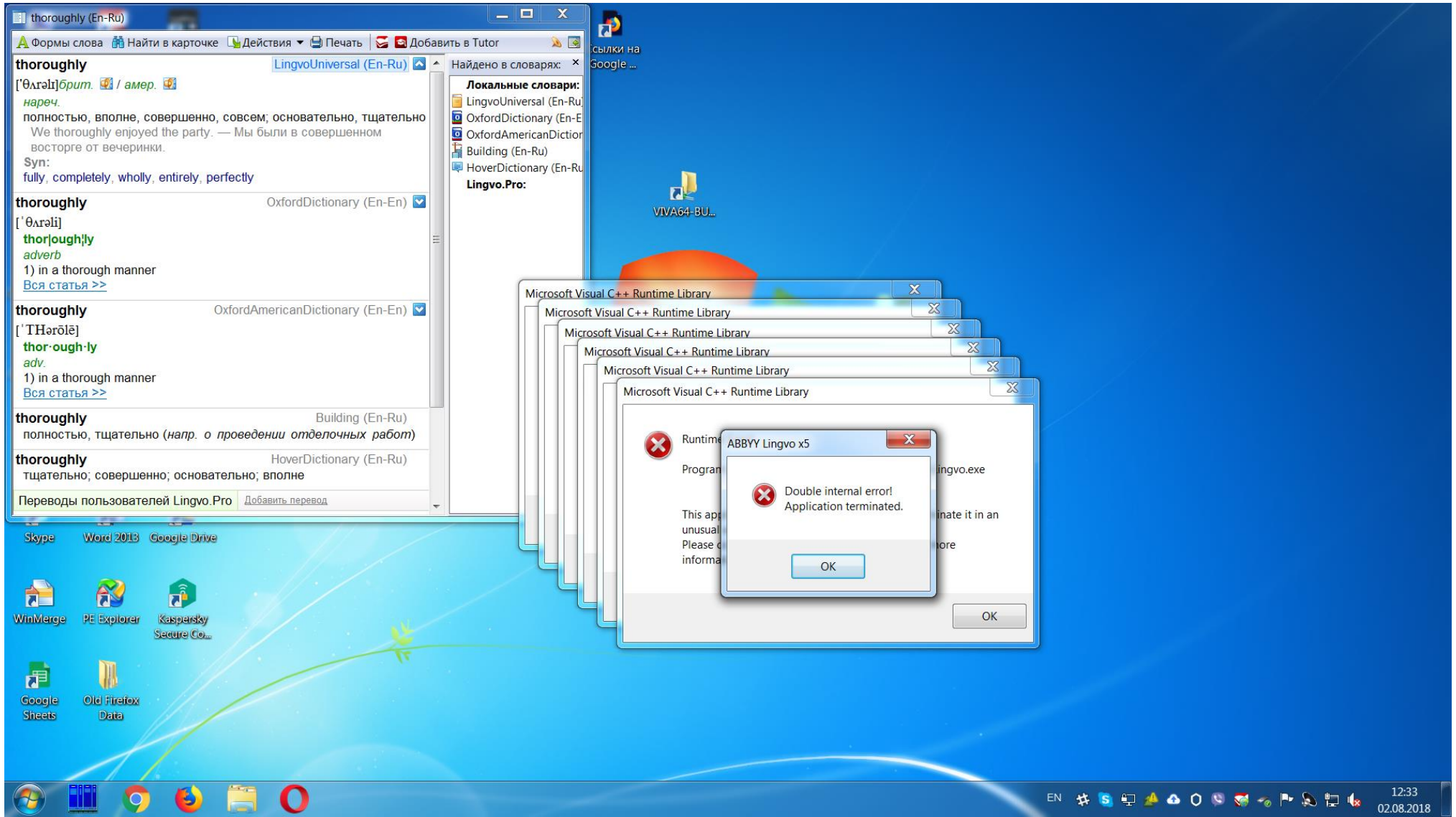
Ошибки в обработчиках ошибок



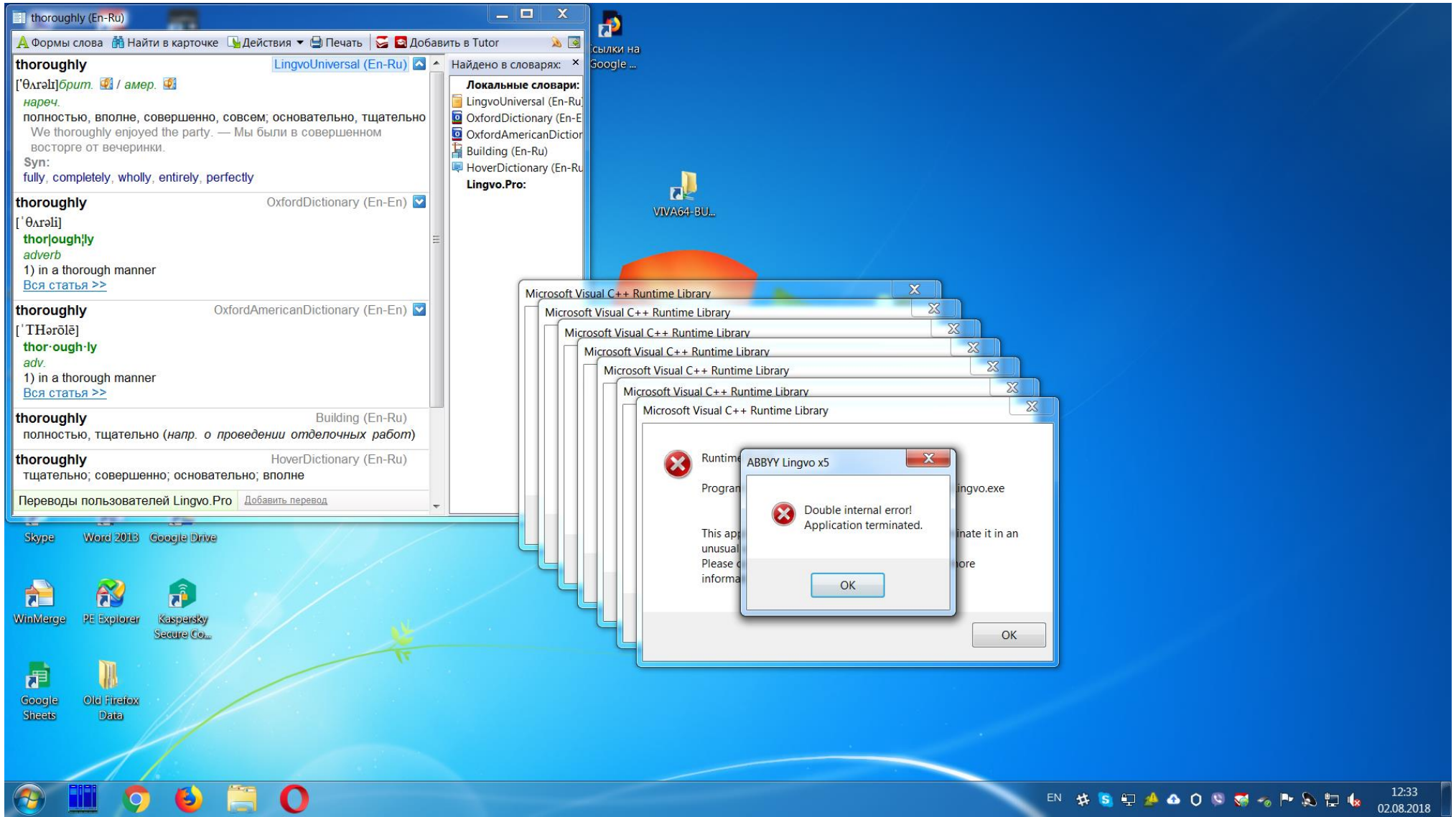
Ошибки в обработчиках ошибок



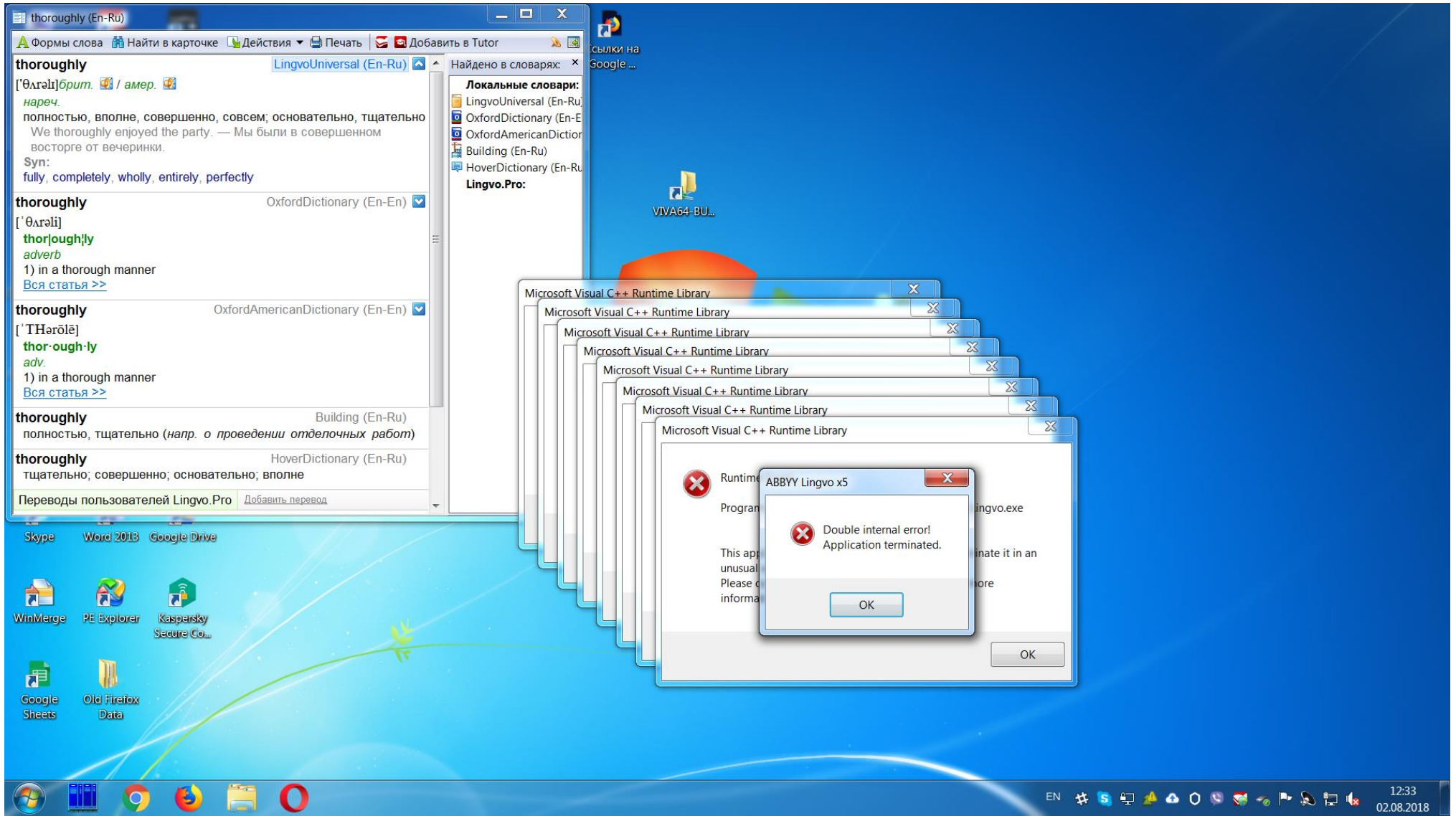
Ошибки в обработчиках ошибок



Ошибки в обработчиках ошибок



Ошибки в обработчиках ошибок



*

Ошибки в обработчиках ошибок

Spoon, Java

```
private Class load(String name) throws CtPathException {
  ....
  try {
    return Class.forName("spoon.reflect.code." + name);
  } catch (ClassNotFoundException ex) {
    throw new CtPathException(
      String.format(
        "Unable to locate element with name $s in Spoon model", name));
  }
}
```

Ошибки в обработчиках ошибок

Spoon, Java

```
private Class load(String name) throws CtPathException {
  ....
  try {
    return Class.forName("spoon.reflect.code." + name);
  } catch (ClassNotFoundException ex) {
    throw new CtPathException(
      String.format(
        "Unable to locate element with name $s in Spoon model", name));
  }
}
```

PVS-Studio: V6046 Incorrect format. A different number of format items is expected. Arguments not used: 1. CtPathStringBuilder.java 54

Ошибки в обработчиках ошибок

Rosegarden, C++

```
bool FileSource::createCacheFile()
{
  ....
  } catch (DirectoryCreationFailed f) {
    return "";
  }
  ....
}
```

Ошибки в обработчиках ошибок

Rosegarden, C++

```
bool FileSource::createCacheFile()
{
  ....
  } catch (DirectoryCreationFailed f) {
    return "";
  }
  ....
}
```

PVS-Studio: V601 The string literal is implicitly cast to the bool type. FileSource.cpp 902

Ошибки в обработчиках ошибок

OpenJDK, C

```
ClassInstancesData  *data;
data = (ClassInstancesData*)user_data;

if (data == NULL) {
  data->error = AGENT_ERROR_ILLEGAL_ARGUMENT;
  return JVMTI_VISIT_ABORT;
}
```

Ошибки в обработчиках ошибок

OpenJDK, C

```
ClassInstancesData  *data;
data = (ClassInstancesData*)user_data;

if (data == NULL) {
  data->error = AGENT_ERROR_ILLEGAL_ARGUMENT;
  return JVMTI_VISIT_ABORT;
}
```

PVS-Studio: V522 Dereferencing of the null pointer 'data' might take place. util.c 2424

Ошибки в обработчиках ошибок

- Сложно написать тест, чтобы проверить обработчики ошибок
- На самом деле, их никто и не проверяет :)
- Статический анализ хорошо дополняет другие методологии

- Альтернатива: Фаззинг. На практике не всегда применимо, особенно, если речь идёт о GUI. Вспомним:

```
throw new CtPathException(
  String.format(
    "Unable to locate element with name $s in Spoon model", name));
```

Android, C

```
static void FwdLockGlue_InitializeRoundKeys() {
  unsigned char keyEncryptionKey[KEY_SIZE];
  ....
  memset(keyEncryptionKey, 0, KEY_SIZE); // Zero out key data.
}
```

CWE-14: Компилятор удаляет код для затирания буфера

Android, C

```
static void FwdLockGlue_InitializeRoundKeys() {
  unsigned char keyEncryptionKey[KEY_SIZE];
  ....
  memset(keyEncryptionKey, 0, KEY_SIZE); // Zero out key data.
}
```

PVS-Studio: V597 CWE-14 The compiler could delete the 'memset' function call, which is used to flush 'keyEncryptionKey' buffer. The memset_s() function should be used to erase the private data. FwdLockGlue.c 102

CWE-14: Компилятор удаляет код для затирания буфера

- Подробнее: https://cwe.mitre.org/data/definitions/14.html
- При отладке Debug-версии этой ошибки нет
- Если не знаешь про этот паттерн, то нельзя найти ошибку во время обзора кода
- Невозможно написать тест, если не искать эту ошибку целенаправленно
- Эта ошибка везде. Я находил её в таких проектах, как: Android, XNU kernel, MySQL, Sphinx, Tizen, FreeBSD Kernel, Linux Kernel, Haiku Operation System, Qt, Apache HTTP Server и т.д.

Android, C

```
char c;
printf("%s is already in *.base_fs format, just copying ....);
rewind(blk_alloc_file);
while ((c = fgetc(blk_alloc_file)) != EOF) {
  fputc(c, base_fs_file);
}
```

Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ
0	0	спец. NOP	32	20	сцеп. SP (Пробел)	64	40	@	96	60	`	128	80	Ђ	160	A0		192	C0	А	224	E0	а
1	1	спец. SOH	33	21	!	65	41	A	97	61	a	129	81	Ѓ	161	A1	Ў	193	C1	Б	225	E1	б
2	2	спец. STX	34	22	"	66	42	B	98	62	b	130	82	‚	162	A2	ў	194	C2	В	226	E2	в
3	3	спец. ETX	35	23	#	67	43	C	99	63	c	131	83	ѓ	163	A3	Ј	195	C3	Г	227	E3	г
4	4	спец. EOT	36	24	$	68	44	D	100	64	d	132	84	„	164	A4	¤	196	C4	Д	228	E4	д
5	5	спец. ENQ	37	25	%	69	45	E	101	65	e	133	85	…	165	A5	Ґ	197	C5	Е	229	E5	е
6	6	спец. ACK	38	26	&	70	46	F	102	66	f	134	86	†	166	A6	¦	198	C6	Ж	230	E6	ж
7	7	спец. BEL	39	27	'	71	47	G	103	67	g	135	87	‡	167	A7	§	199	C7	З	231	E7	з
8	8	спец. BS	40	28	(	72	48	H	104	68	h	136	88	€	168	A8	Ё	200	C8	И	232	E8	и
9	9	спец. Табуляция	41	29	)	73	49	I	105	69	i	137	89	‰	169	A9	©	201	C9	Й	233	E9	й
10	0A	спец. LF (Возвр. каретки)	42	2A	*	74	4A	J	106	6A	j	138	8A	Љ	170	AA	Є	202	CA	К	234	EA	к
11	0B	спец. VT	43	2B	+	75	4B	K	107	6B	k	139	8B	‹	171	AB	«	203	CB	Л	235	EB	л
12	0C	спец. FF	44	2C	,	76	4C	L	108	6C	l	140	8C	Њ	172	AC	¬	204	CC	М	236	EC	м
13	0D	спец. CR (Новая строка)	45	2D	-	77	4D	M	109	6D	m	141	8D	Ќ	173	AD	-	205	CD	Н	237	ED	н
14	0E	спец. SO	46	2E	.	78	4E	N	110	6E	n	142	8E	Ћ	174	AE	®	206	CE	О	238	EE	о
15	0F	спец. SI	47	2F	/	79	4F	O	111	6F	o	143	8F	Џ	175	AF	Ї	207	CF	П	239	EF	п
16	10	спец. DLE	48	30	0	80	50	P	112	70	p	144	90	ђ	176	B0	°	208	D0	Р	240	F0	р
17	11	спец. DC1	49	31	1	81	51	Q	113	71	q	145	91	‘	177	B1	±	209	D1	С	241	F1	с
18	12	спец. DC2	50	32	2	82	52	R	114	72	r	146	92	’	178	B2	І	210	D2	Т	242	F2	т
19	13	спец. DC3	51	33	3	83	53	S	115	73	s	147	93	“	179	B3	і	211	D3	У	243	F3	у
20	14	спец. DC4	52	34	4	84	54	T	116	74	t	148	94	”	180	B4	ґ	212	D4	Ф	244	F4	ф
21	15	спец. NAK	53	35	5	85	55	U	117	75	u	149	95	•	181	B5	µ	213	D5	Х	245	F5	х
22	16	спец. SYN	54	36	6	86	56	V	118	76	v	150	96	–	182	B6	¶	214	D6	Ц	246	F6	ц
23	17	спец. ETB	55	37	7	87	57	W	119	77	w	151	97	—	183	B7	·	215	D7	Ч	247	F7	ч
24	18	спец. CAN	56	38	8	88	58	X	120	78	x	152	98	￿	184	B8	ё	216	D8	Ш	248	F8	ш
25	19	спец. EM	57	39	9	89	59	Y	121	79	y	153	99	™	185	B9	№	217	D9	Щ	249	F9	щ
26	1A	спец. SUB	58	3A	:	90	5A	Z	122	7A	z	154	9A	љ	186	BA	є	218	DA	Ъ	250	FA	ъ
27	1B	спец. ESC	59	3B	;	91	5B	[	123	7B	{	155	9B	›	187	BB	»	219	DB	Ы	251	FB	ы
28	1C	спец. FS	60	3C	<	92	5C	\	124	7C	|	156	9C	њ	188	BC	j	220	DC	Ь	252	FC	ь
29	1D	спец. GS	61	3D	=	93	5D	]	125	7D	}	157	9D	ќ	189	BD	S	221	DD	Э	253	FD	э
30	1E	спец. RS	62	3E	>	94	5E	^	126	7E	~	158	9E	ћ	190	BE	s	222	DE	Ю	254	FE	ю
31	1F	спец. US	63	3F	?	95	5F	_	127	7F		159	9F	џ	191	BF	ї	223	DF	Я	255	FF	я

Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ	Dec	Hex	Символ
0	0	спец. NOP	32	20	сцеп. SP (Пробел)	64	40	@	96	60	`	128	80	Ђ	160	A0		192	C0	А	224	E0	а
1	1	спец. SOH	33	21	!	65	41	A	97	61	a	129	81	Ѓ	161	A1	Ў	193	C1	Б	225	E1	б
2	2	спец. STX	34	22	"	66	42	B	98	62	b	130	82	‚	162	A2	ў	194	C2	В	226	E2	в
3	3	спец. ETX	35	23	#	67	43	C	99	63	c	131	83	ѓ	163	A3	Ј	195	C3	Г	227	E3	г
4	4	спец. EOT	36	24	$	68	44	D	100	64	d	132	84	„	164	A4	¤	196	C4	Д	228	E4	д
5	5	спец. ENQ	37	25	%	69	45	E	101	65	e	133	85	…	165	A5	Ґ	197	C5	Е	229	E5	е
6	6	спец. ACK	38	26	&	70	46	F	102	66	f	134	86	†	166	A6	¦	198	C6	Ж	230	E6	ж
7	7	спец. BEL	39	27	'	71	47	G	103	67	g	135	87	‡	167	A7	§	199	C7	З	231	E7	з
8	8	спец. BS	40	28	(	72	48	H	104	68	h	136	88	€	168	A8	Ё	200	C8	И	232	E8	и
9	9	спец. Табуляция	41	29	)	73	49	I	105	69	i	137	89	‰	169	A9	©	201	C9	Й	233	E9	й
10	0A	спец. LF (Возвр. каретки)	42	2A	*	74	4A	J	106	6A	j	138	8A	Љ	170	AA	Є	202	CA	К	234	EA	к
11	0B	спец. VT	43	2B	+	75	4B	K	107	6B	k	139	8B	‹	171	AB	«	203	CB	Л	235	EB	л
12	0C	спец. FF	44	2C	,	76	4C	L	108	6C	l	140	8C	Њ	172	AC	¬	204	CC	М	236	EC	м
13	0D	спец. CR (Новая строка)	45	2D	-	77	4D	M	109	6D	m	141	8D	Ќ	173	AD	-	205	CD	Н	237	ED	н
14	0E	спец. SO	46	2E	.	78	4E	N	110	6E	n	142	8E	Ћ	174	AE	®	206	CE	О	238	EE	о
15	0F	спец. SI	47	2F	/	79	4F	O	111	6F	o	143	8F	Џ	175	AF	Ї	207	CF	П	239	EF	п
16	10	спец. DLE	48	30	0	80	50	P	112	70	p	144	90	ђ	176	B0	°	208	D0	Р	240	F0	р
17	11	спец. DC1	49	31	1	81	51	Q	113	71	q	145	91	‘	177	B1	±	209	D1	С	241	F1	с
18	12	спец. DC2	50	32	2	82	52	R	114	72	r	146	92	’	178	B2	І	210	D2	Т	242	F2	т
19	13	спец. DC3	51	33	3	83	53	S	115	73	s	147	93	“	179	B3	і	211	D3	У	243	F3	у
20	14	спец. DC4	52	34	4	84	54	T	116	74	t	148	94	”	180	B4	ґ	212	D4	Ф	244	F4	ф
21	15	спец. NAK	53	35	5	85	55	U	117	75	u	149	95	•	181	B5	µ	213	D5	Х	245	F5	х
22	16	спец. SYN	54	36	6	86	56	V	118	76	v	150	96	–	182	B6	¶	214	D6	Ц	246	F6	ц
23	17	спец. ETB	55	37	7	87	57	W	119	77	w	151	97	—	183	B7	·	215	D7	Ч	247	F7	ч
24	18	спец. CAN	56	38	8	88	58	X	120	78	x	152	98	￿	184	B8	ё	216	D8	Ш	248	F8	ш
25	19	спец. EM	57	39	9	89	59	Y	121	79	y	153	99	™	185	B9	№	217	D9	Щ	249	F9	щ
26	1A	спец. SUB	58	3A	:	90	5A	Z	122	7A	z	154	9A	љ	186	BA	є	218	DA	Ъ	250	FA	ъ
27	1B	спец. ESC	59	3B	;	91	5B	[	123	7B	{	155	9B	›	187	BB	»	219	DB	Ы	251	FB	ы
28	1C	спец. FS	60	3C	<	92	5C	\	124	7C	|	156	9C	њ	188	BC	j	220	DC	Ь	252	FC	ь
29	1D	спец. GS	61	3D	=	93	5D	]	125	7D	}	157	9D	ќ	189	BD	S	221	DD	Э	253	FD	э
30	1E	спец. RS	62	3E	>	94	5E	^	126	7E	~	158	9E	ћ	190	BE	s	222	DE	Ю	254	FE	ю
31	1F	спец. US	63	3F	?	95	5F	_	127	7F		159	9F	џ	191	BF	ї	223	DF	Я	255	FF	я

Windows кодировка **CP1251**.

Буква 'я' имеет код 255.

Злоключения буквы 'я'

Android, C

```
char c;
printf("%s is already in *.base_fs format, just copying ....);
rewind(blk_alloc_file);
while ((c = fgetc(blk_alloc_file)) != EOF) {
  fputc(c, base_fs_file);
}
```

PVS-Studio: V739 CWE-20 EOF should not be compared with a value of the 'char' type. The '(c = fgetc(blk_alloc_file))' should be of the 'int' type. blk_alloc_to_base_fs.c 61

Злоключения буквы 'я'

- У программиста будут работать его тесты
- Программист/тестеровщик не догадается и не проверит, что он неверно работает с каким-то незнакомым ему алфавитом

Никто не тестирует функции сравнения

- Программисты редко проверяют функции сравнения, ибо:
  - лень
  - они простые – там ошибок нет
- Программисты редко пишут юнит-тесты на функции сравнения, ибо:
  - лень
  - они простые – там ошибок нет

- Proof: "Зло живёт в функциях сравнения" https://www.viva64.com/ru/b/0509/

Никто не тестирует функции сравнения

Hibernate, Java

```
public boolean equals(Object other) {
    if (other instanceof Id) {
        Id that = (Id) other;
        return purchaseSequence.equals(this.purchaseSequence) &&
            that.purchaseNumber == this.purchaseNumber;
    }
    else {
        return false;
    }
}
```

PVS-Studio: V6009 Function 'equals' receives odd arguments. Inspect first argument. PurchaseRecord.java 57

Никто не тестирует функции сравнения

Apache Hive, Java

```
Collections.sort(list, new Comparator<ServiceInstance>() {
  @Override
  public int compare(ServiceInstance o1, ServiceInstance o2)
  {
    return o2.getWorkerIdentity().compareTo(o2.getWorkerIdentity());
  }
});
```

PVS-Studio: V6009 Function 'compareTo' receives odd arguments. Inspect first argument. PurchaseRecord.java 57

Никто не тестирует функции сравнения

- Ещё раз прошу прочитать "Зло живёт в функциях сравнения" https://www.viva64.com/ru/b/0509/
- Вы поймёте, что это серьёзно и повсеместно

- **Всё равно никто не будет вручную писать тесты, это муторно.** Рассмотрим пример.

Никто не тестирует функции сравнения

```
bool Compare(const FPooledRenderTargetDesc& rhs, bool bExact) const
{
  ....
  return Extent == rhs.Extent
    && Depth == rhs.Depth
    && bIsArray == rhs.bIsArray
    && ArraySize == rhs.ArraySize
    && NumMips == rhs.NumMips
    && NumSamples == rhs.NumSamples
    && Format == rhs.Format
    && LhsFlags == RhsFlags
    && TargetableFlags == rhs.TargetableFlags
    && bForceSeparateTargetAndShaderResource ==
         rhs.bForceSeparateTargetAndShaderResource
    && ClearValue == rhs.ClearValue
    && AutoWritable == AutoWritable;
}
```

Unreal Engine 4, C++

PVS-Studio: V501 There are identical sub-expressions to the left and to the right of the '==' operator: AutoWritable == AutoWritable rendererinterface.h 180

Никто не тестирует функции сравнения. Что делать?

- Оформление кода
- Статический анализ кода
- Генераторы тестов
  - Андрей Сатарин – EqualsVerifier, ErrorProne и все-все-все
  - https://youtu.be/jeCpYOEuL64

Никто не тестирует функции сравнения

```
bool Compare(const FPooledRenderTargetDesc& rhs, bool bExact) const
{
  ....
  return    Extent              == rhs.Extent
         && Depth               == rhs.Depth
         && bIsArray            == rhs.bIsArray
         && ArraySize           == rhs.ArraySize
         && NumMips             == rhs.NumMips
         && NumSamples          == rhs.NumSamples
         && Format              == rhs.Format
         && LhsFlags            == RhsFlags
         && TargetableFlags     == rhs.TargetableFlags
         && bForceSeparateTargetAndShaderResource ==
              rhs.bForceSeparateTargetAndShaderResource
         && ClearValue          == rhs.ClearValue
         && AutoWritable        == AutoWritable;
}
```

"Табличное форматирование"

Стало лучше, но не идеально.

Стоит использовать, но это не отменяет необходимость статического анализа кода.

Бла-бла код

Hibernate, Java

```
annotationToXml.put( NamedNativeQuery.class, "named-native-query" );
annotationToXml.put( NamedNativeQueries.class, "named-native-query" );
annotationToXml.put( NamedStoredProcedureQuery.class, "named-stored-procedure-query" );
annotationToXml.put( NamedStoredProcedureQueries.class, "named-stored-procedure-query" );
annotationToXml.put( SqlResultSetMapping.class, "sql-result-set-mapping" );
annotationToXml.put( SqlResultSetMappings.class, "sql-result-set-mapping" );
annotationToXml.put( ExcludeDefaultListeners.class, "exclude-default-listeners" );
annotationToXml.put( ExcludeSuperclassListeners.class, "exclude-superclass-listeners" );
annotationToXml.put( AccessType.class, "access" );
annotationToXml.put( AttributeOverride.class, "attribute-override" );
annotationToXml.put( AttributeOverrides.class, "attribute-override" );
annotationToXml.put( AttributeOverride.class, "association-override" );
annotationToXml.put( AttributeOverrides.class, "association-override" );
annotationToXml.put( AttributeOverride.class, "map-key-attribute-override" );
annotationToXml.put( AttributeOverrides.class, "map-key-attribute-override" );
annotationToXml.put( Id.class, "id" );
annotationToXml.put( EmbeddedId.class, "embedded-id" );
annotationToXml.put( GeneratedValue.class, "generated-value" );
annotationToXml.put( Column.class, "column" );
annotationToXml.put( Columns.class, "column" );
annotationToXml.put( Temporal.class, "temporal" );
annotationToXml.put( Lob.class, "lob" );
annotationToXml.put( Enumerated.class, "enumerated" );
annotationToXml.put( Version.class, "version" );
annotationToXml.put( Transient.class, "transient" );
annotationToXml.put( Basic.class, "basic" );
annotationToXml.put( Embedded.class, "embedded" );
```

Бла-бла код

Hibernate, Java

```
private static final Map<Class, String> annotationToXml;
....
annotationToXml.put(AttributeOverride.class, "attribute-override");
....
annotationToXml.put(AttributeOverride.class, "association-override");
....
annotationToXml.put(AttributeOverride.class, "map-key-attribute-override");
```

- V6033 An item with the same key 'javax.persistence.AttributeOverride.class' has already been added. Check lines: 188, 186. JPAOverriddenAnnotationReader.java 188
- V6033 An item with the same key 'javax.persistence.AttributeOverride.class' has already been added. Check lines: 190, 186. JPAOverriddenAnnotationReader.java 190

Бла-бла код

EA WebKit, C++

```
value.start_fragment_union_rect.size.width =
  std::max(descendant.offset_to_container_box.left +
        descendant.fragment->Size().width -
        value.start_fragment_union_rect.offset.left,
      value.start_fragment_union_rect.size.width);
value.start_fragment_union_rect.size.height =
  std::max(descendant.offset_to_container_box.top +
        descendant.fragment->Size().height -
        value.start_fragment_union_rect.offset.top,
      value.start_fragment_union_rect.size.width);
```

Бла-бла код

EA WebKit, C++

```
value.start_fragment_union_rect.size.width =
  std::max(descendant.offset_to_container_box.left +
        descendant.fragment->Size().width -
        value.start_fragment_union_rect.offset.left,
      value.start_fragment_union_rect.size.width);
value.start_fragment_union_rect.size.height =
  std::max(descendant.offset_to_container_box.top +
        descendant.fragment->Size().height -
        value.start_fragment_union_rect.offset.top,
      value.start_fragment_union_rect.size.width);
```

PVS-Studio: V778 CWE-682 Two similar code fragments were found. Perhaps, this is a typo and 'height' variable should be used instead of 'width'. ng_fragment_builder.cc 326

Бла-бла код

- Чаще всего причина ошибок в таком коде – это Copy-Paste
- Юнит-тесты писать сложно и неинтересно
- Полноценный обзор кода почти бесполезен или будет **неоправданно дорог**
- Могут помочь разные способы тестирования, но статический анализ - самая лучшая первая линия обороны

Ошибки в тестах

Entity Framework, C#

```
for (var i = 0; i < result.Count; i++)
{
  ....
  for (var j = 0; j < expectedInnerNames.Count; j++)
  {
    Assert.True(
      result[i]
      .OneToMany_Optional.Select(e => e.Name)
      .Contains(expectedInnerNames[i])
    );
  }
}
```

PVS-Studio: V3081 The 'j' counter is not used inside a nested loop. Consider inspecting usage of 'i' counter. EFCore.Specification.Tests ComplexNavigationsQueryTestBase.cs 2393

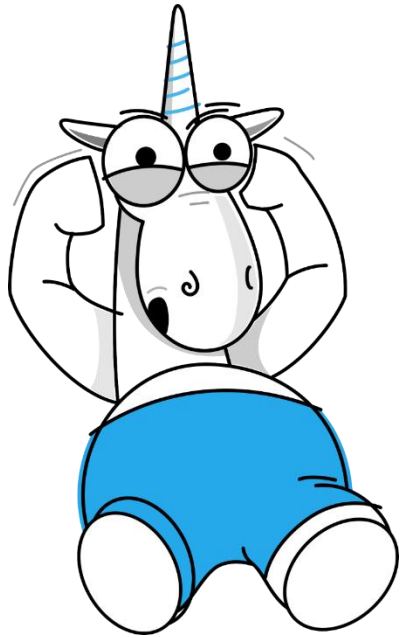
Ошибки в тестах

```
for (int i = 0; i < 20; i++)
{
  ....
  if (i % 2 == 0)
  {
    thread1.Start();
    thread2.Start();
  }
  else
  {
    thread1.Start();
    thread2.Start();
  }
  ....
```

.NET Compiler Platform ("Roslyn"), C#

PVS-Studio: V3004 The 'then' statement is equivalent to the 'else' statement. GetSemanticInfoTests.cs 2269

Ошибки в тестах



- Тесты для тестов?
- Тесты для тестов тестов? :)
- Никто не тестирует тесты!

- Многие тесты ничего на самом деле не проверяют
- Здесь статический анализ отлично дополняет юнит-тесты

Неправильное ограничение по количеству итераций или времени

Jenkins, Java

```
public final R getSomeBuildWithWorkspace() {
  int cnt=0;
  for (R b = getLastBuild(); cnt<5 && b!=null; b=b.getPreviousBuild())
  {
    FilePath ws = b.getWorkspace();
    if (ws!=null)   return b;
  }
  return null;
}
```

PVS-Studio: V6007 Expression 'cnt < 5' is always true. AbstractProject.java 557

Неправильное ограничение по количеству итераций или времени

FreeBSD Kernel, C

```
#define  AE_IDLE_TIMEOUT    100
....
int i;
....
/* Wait for IDLE state. */
for (i = 0; i < AE_IDLE_TIMEOUT; i--) {
  val = AE_READ_4(sc, AE_IDLE_REG);
  if ((val & (AE_IDLE_RXMAC | AE_IDLE_DMAWRITE)) == 0)
    break;
  DELAY(100);
}
```

PVS-Studio: V621 Consider inspecting the 'for' operator. It's possible that the loop will be executed incorrectly or won't be executed at all. if_ae.c 1663

Неправильное ограничение по количеству итераций или времени

- Незаметно, ведь вроде как работает....
- Сложно тестировать

Хождение по тонкому льду

Umbraco, C#

```
internal static ImageCropData GetCrop(....)
{
  var imageCropDatas = dataset.ToArray();
  if (dataset == null || imageCropDatas.Any() == false)
    return null;
  ....
}
```

PVS-Studio: V3095 The 'dataset' object was used before it was verified against null. Check lines: 48, 49. ImageCropperBaseExtensions.cs 48

Хождение по тонкому льду

EFL, C

```
filter->data_count++;
array = realloc(filter->data,
  sizeof(Edje_Part_Description_Spec_Filter_Data) *
  filter->data_count);
array[filter->data_count - 1].name = name;
array[filter->data_count - 1].value = value;
filter->data = array;
```

PVS-Studio: V522 There might be dereferencing of a potential null pointer 'array'. edje_cc_handlers.c 14249

Хождение по тонкому льду

Oracle MySQL 5.1.x, C

```
typedef char my_bool;

my_bool
check_scramble(const char *scramble_arg, const char *message,
               const uint8 *hash_stage2)
{
  ....
  return memcmp(hash_stage2, hash_stage2_reassured, SHA1_HASH_SIZE);
}
```

CVE-2012-2122 (Не мы нашли, но могли бы)

PVS-Studio: V642 Saving the 'memcmp' function result inside the 'char' type variable is inappropriate. The significant bits could be lost breaking the program's logic. password.c

Хождение по тонкому льду

- Ошибки проявляют себя очень редко
- Сложно сделать тест. Например, эмулировать, что память кончилась
- Ручное тестирование – вообще не вариант (попробуй дождись сильную фрагментацию памяти и невозможность выделить массив)

- P.S. Рекомендую заметку «Почему важно проверять, что вернула функция malloc» https://www.viva64.com/ru/b/0558/

Промежуточные итоги

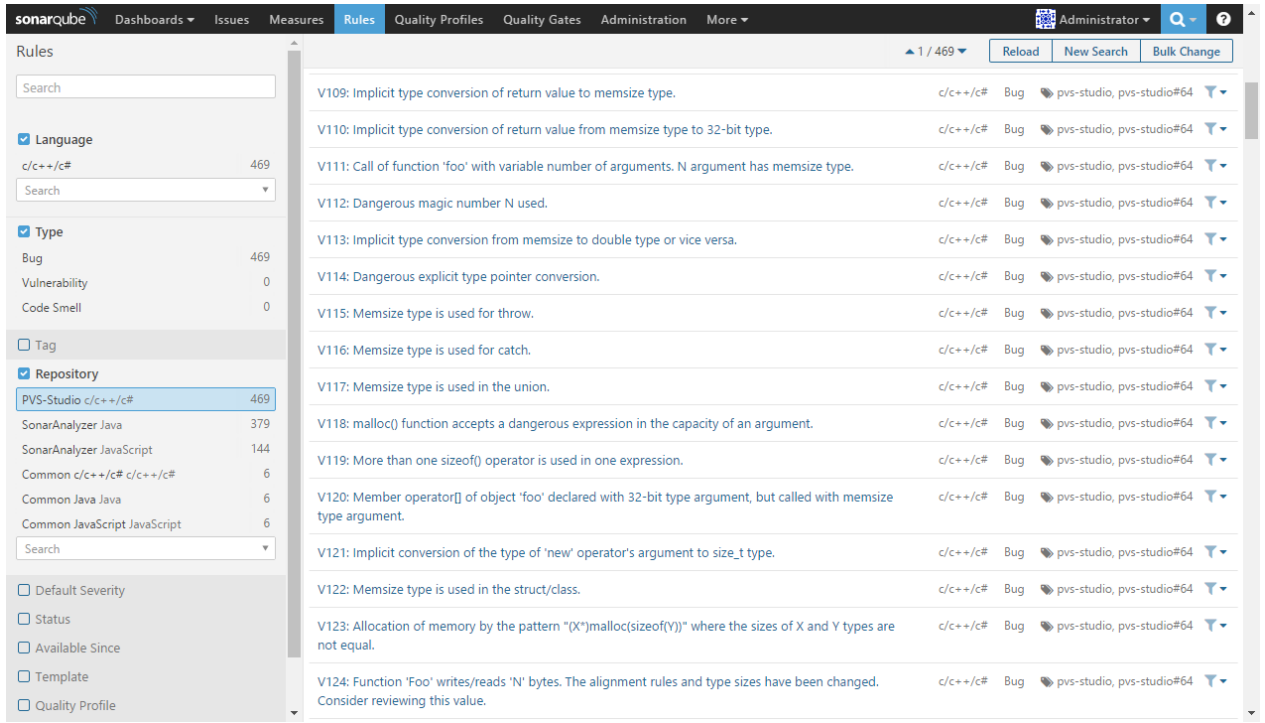
- Статический анализ не лучше и не хуже, чем другие методы поиска и устранения ошибок
- Это неотъемлемая часть цикла разработки качественного ПО
- Разные методики дополняют друг друга

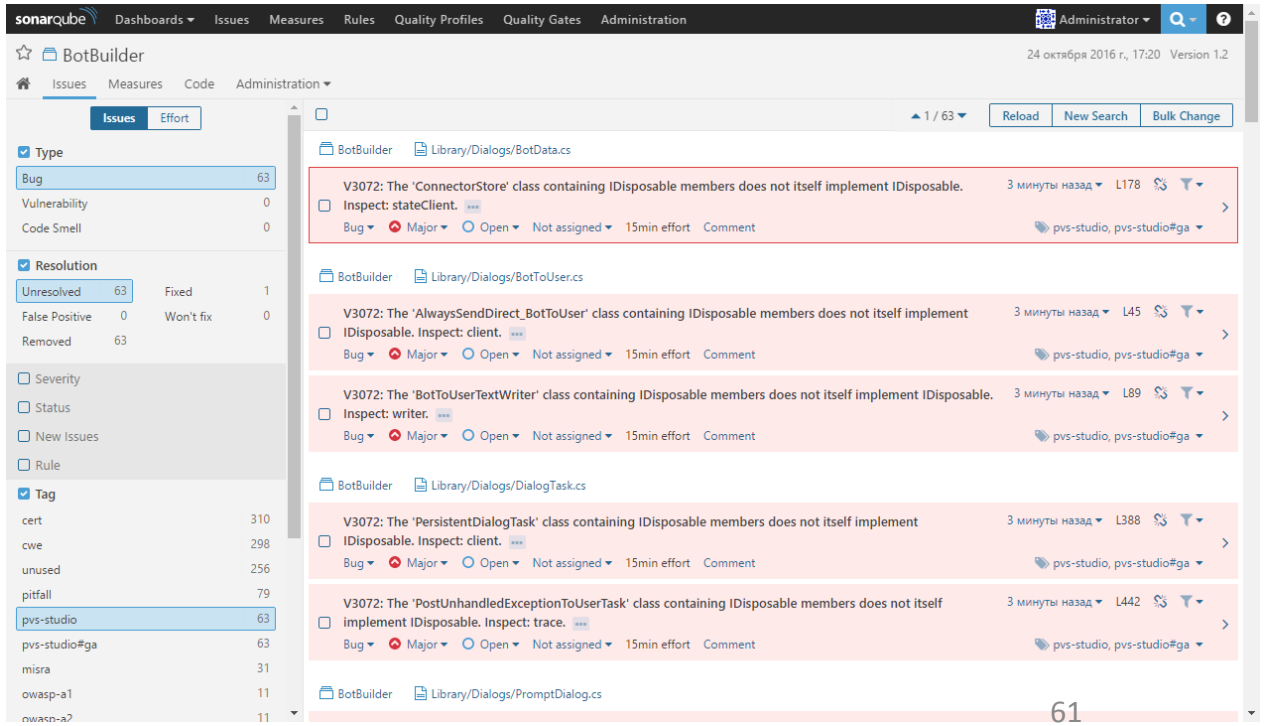
Важно

- Регулярный анализ
- Аналогия: предупреждения компилятора

«*Одно кольцо соберёт их …*»

- Что такое SonarQube, и зачем он нужен
- PVS-Studio как плагин для SonarQube





100500 предупреждений

- Разметка имеющихся предупреждений как неактуальных для начала
- Правим ошибки в новом или изменённом коде
- Потихоньку устраняем технический долг
- Обязательно ночные прогоны и рассылка уведомлений
- Если очень большой проект - инкрементальный анализ

- PVS-Studio – это инструмент для выявления ошибок и потенциальных уязвимостей
- C, C++, C# и Java
- Windows, Linux и macOS

- Смежные умные слова: SAST, CWE, SEI CERT, MISRA, DevSecOps, SonarQube, IncrediBuild, CI.

Как быстро попробовать PVS-Studio

- Скачать PVS-Studio: https://www.viva64.com/ru/pvs-studio-download/
- Быстрый старт (мониторинг компиляции)
  - Windows утилита: CLMonitoring
  - Linux/macOS утилита: pvs-studio-analyzer

Ответы на вопросы

E-Mail: karpov@viva64.com
Twitter: @Code_Analysis
Instagram: @pvs_studio_unicorn