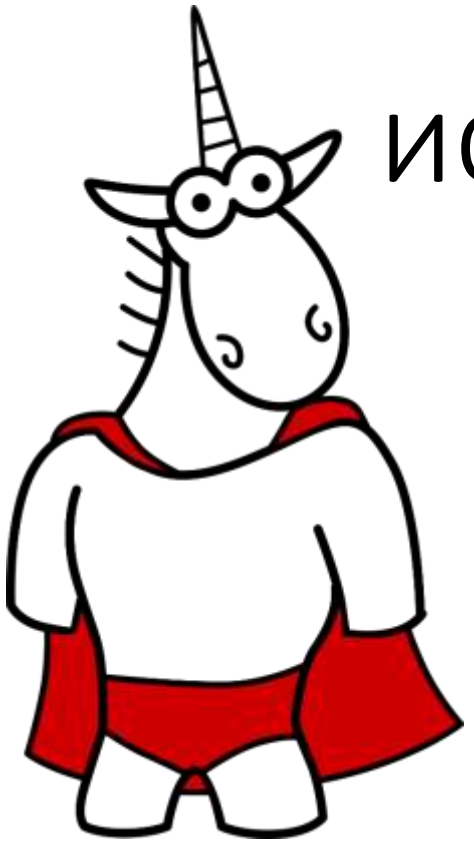


Как избежать ошибок, используя современный C++

Павел Беликов

PVS-Studio

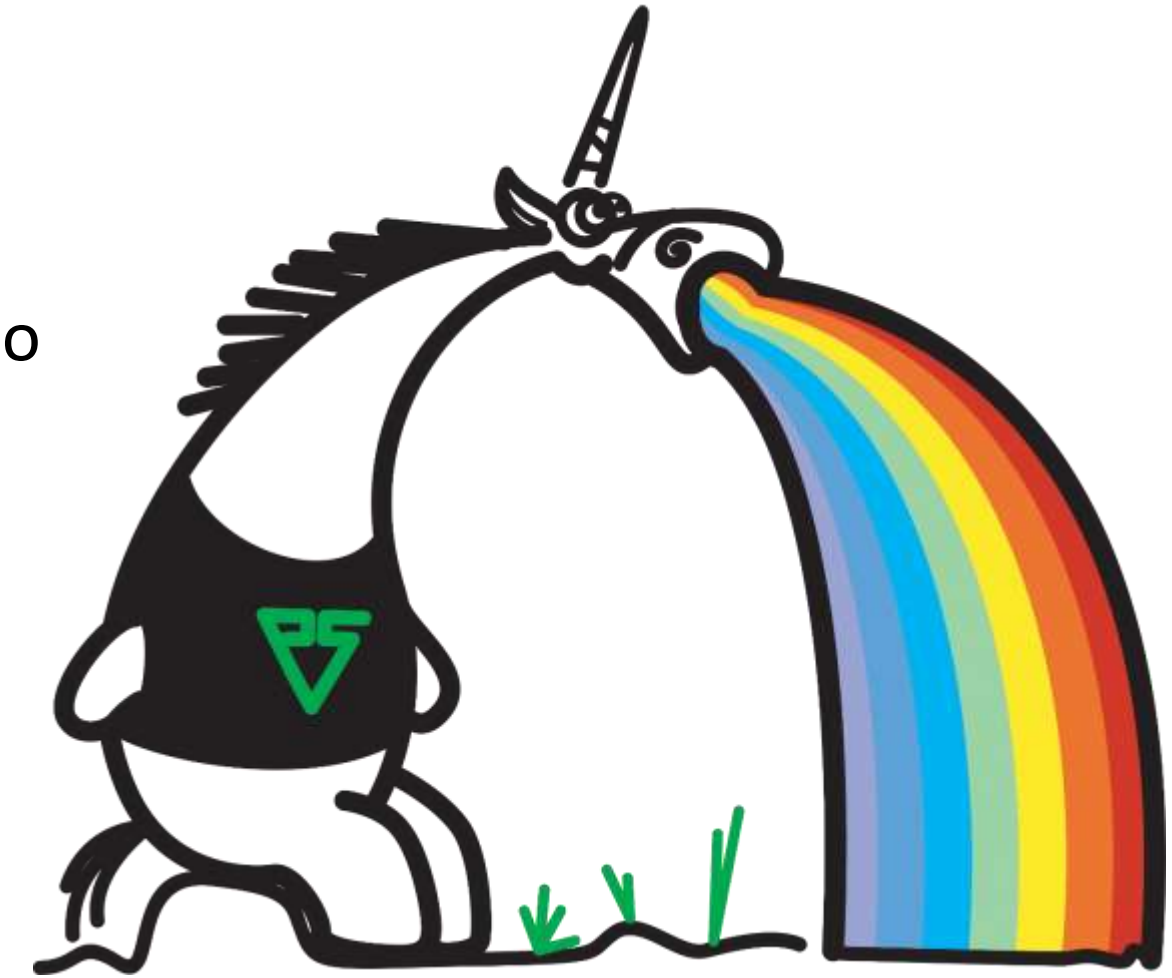
www.viva64.com

О докладчике

Павел Беликов

Разработчик в команде PVS-Studio

Мы разрабатываем статический анализатор C, C++, C# кода

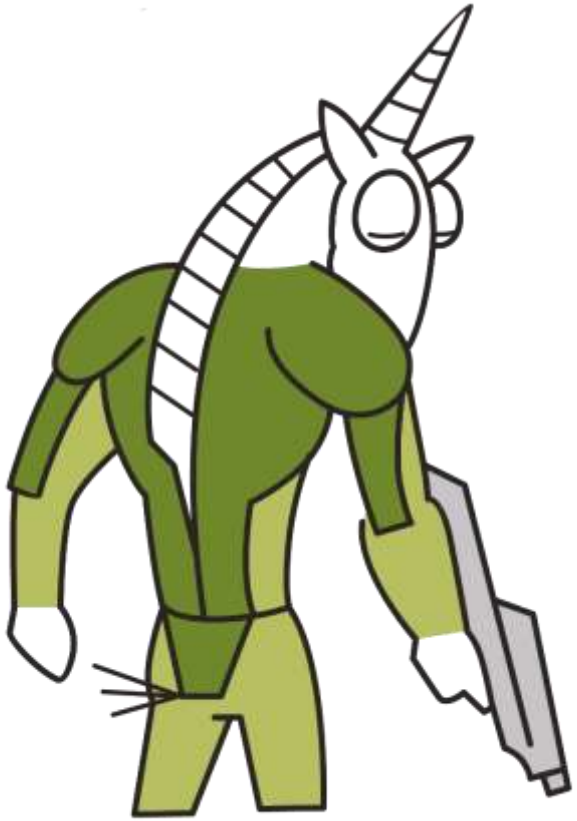


Что такое Modern C++?

- потоки, асинхронность, параллельные вычисления
- лямбды
- умные указатели
- исключения
- и, конечно же, безопасный код

Чем нам может помочь Modern C++?

- Посмотрим на ошибки из различных Open Source проектов, проверенных при помощи PVS-Studio.
- Подумаем, как обезопасить свой код.



auto

```
string str = .....;
unsigned n = str.find("ABC");
if (n != string::npos)

string str = .....;
auto n = str.find("ABC");
if (n != string::npos)
```

Чем опасно auto

```
auto n = 1024*1024*1024*5;
char* buf = new char[n];
```

Чем опасно auto

Плохо:

```
std::vector<int> bigVector;
for (unsigned i = 0; i < bigVector.size(); ++Index)
{ ... }
```

Ещё хуже:

```
std::vector<int> bigVector;
for (auto i = 0; i < bigVector.size(); ++Index)
{ ... }
```

Опасный countof

```
#define RTL_NUMBER_OF_V1(A) (sizeof(A)/sizeof((A)[0]))
#define _ARRAYSIZE(A) RTL_NUMBER_OF_V1(A)

int GetAllNeighbors( const CCoreDispInfo *pDisp,
                     int iNeighbors[512] ) {
  ....
  if ( nNeighbors < _ARRAYSIZE( iNeighbors ) )

    iNeighbors[nNeighbors++] = pCorner->m_Neighbors[i];
  ....
}
```

Source Engine SDK

Предупреждение PVS-Studio: V511 The sizeof() operator returns size of the pointer, and not of the array, in 'sizeof (iNeighbors)' expression. Vrad_dll disp_vrad.cpp 60

Опасный countof

```
template < class T, size_t N >
constexpr size_t
countof( const T (&array)[N] ) {
  return N;
}
countof(salt); //compile-time error
```

В C++17 есть функция std::size.

Опасный countof

Функции из C++11 тоже могут быть коварны:

```
std::extent<decltype(salt)>();
//=> 0
```

std::array

```
void Foo(std::array<uint8, 16> salt)
{
  salt.size(); //=> 16
}
```

sizeof тоже опасен

```
VisitedLinkMaster::TableBuilder::TableBuilder(
    VisitedLinkMaster* master,
    const uint8 salt[LINK_SALT_LENGTH])
    : master_(master),
      success_(true) {
  fingerprints_.reserve(4096);
  memcpy(salt_, salt, sizeof(salt));
}
```

Chromium

Предупреждения PVS-Studio:

- V511 The sizeof() operator returns size of the pointer, and not of the array, in 'sizeof (salt)' expression. browser visitedlink_master.cc 968
- V512 A call of the 'memcpy' function will lead to underflow of the buffer 'salt_'. browser visitedlink_master.cc 968

Как ошибаются в простом for

```
const int SerialWindow::kBaudrates[] = { 50, 75, 110, .... };

SerialWindow::SerialWindow() : ....
{
  ....
  for(int i = sizeof(kBaudrates) / sizeof(char*); --i >= 0;)
  {
    message->AddInt32("baudrate", kBaudrateConstants[i]);
    ....
  }
}
```

Haiku Operation System

Предупреждение PVS-Studio: V706 Suspicious division: sizeof (kBaudrates) / sizeof (char *). Size of every element in 'kBaudrates' array does not equal to divisor. SerialWindow.cpp 162

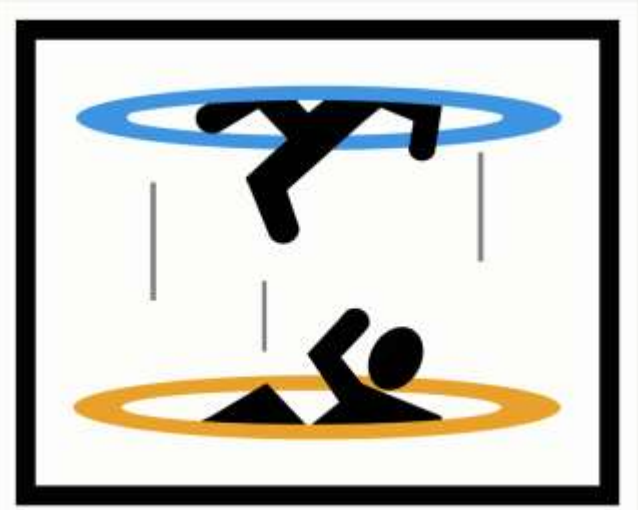
Как ошибаются в простом for

```
const int SerialWindow::kBaudrates[] = { 50, 75, 110, .... };

SerialWindow::SerialWindow() : ....
{
  ....
  for(int i = std::size(kBaudrates); --i >= 0;) {
    message->AddInt32("baudrate", kBaudrateConstants[i]);
    ....
  }
}
```

Как ошибаются в простом for

```
inline void CXmlReader::CXmlInputStream::UnsafePutCharsBack(
  const TCHAR* pChars, size_t nNumChars)
{
  if (nNumChars > 0)
  {
    for (size_t nCharPos = nNumChars - 1;
         nCharPos >= 0;
         --nCharPos)
      UnsafePutCharBack(pChars[nCharPos]);
  }
}
```



Shareaza

Предупреждение PVS-Studio: V547 Expression 'nCharPos >= 0' is always true. Unsigned type value is always >= 0. BugTrap xmlreader.h 946

Безопасные циклы в современном C++

Non-member функции:

```
char buf[4] = { 'a', 'b', 'c', 'd' };
for (auto it = rbegin(buf);
     it != rend(buf);
     ++it) {
  std::cout << *it;
}
```

Безопасные циклы в современном C++

range-based for

```
char buf[4] = { 'a', 'b', 'c', 'd' };
for (auto it : buf) {
  std::cout << it;
}
```

Безопасные циклы в современном C++

```
template <typename T>
struct reversed_wrapper {
  const T& _v;

  reversed_wrapper (const T& v) : _v(v) {}
  auto begin() -> decltype(rbegin(_v)) { return rbegin(_v); }
  auto end()   -> decltype(rend(_v))   { return rend(_v);   }
};

template <typename T>
reversed_wrapper<T> reversed(const T& v)
              { return reversed_wrapper<T>(v); }
```

Безопасные циклы в современном C++

```
char buf[4] = { 'a', 'b', 'c', 'd' };
for (auto it : reversed(buf)) {
  std::cout << it;
}
```

Можно было использовать boost::adaptors::reverse(buf).

std::string_view

```
void Foo(std::string_view s);


std::string str = “abc”;
Foo(std::string_view(“abc”, 3));
Foo(“abc”);
Foo(str);
```

std::string_view

```
inline void
CXmlReader::CXmlInputStream::UnsafePutCharsBack(
  std::wstring_view chars)
{
  for (wchar_t ch : reversed(chars))
    UnsafePutCharBack(ch);
}

Foo(wstring_view(pChars, nNumChars));
Foo(pChars);
```

enum

```
enum iscsi_param {
   ....
   ISCSI_PARAM_CONN_PORT,
   ISCSI_PARAM_CONN_ADDRESS,
   ....
};

enum iscsi_host_param {
   ....
   ISCSI_HOST_PARAM_IPADDRESS,
   ....
};
```

```
int iscsi_conn_get_addr_param(....,
   enum iscsi_param param, ....)
{
   ....
   switch (param) {
   case ISCSI_PARAM_CONN_ADDRESS:
   case ISCSI_HOST_PARAM_IPADDRESS:
   ....
   }

   return len;
}
```

enum class

```
enum class ISCSI_PARAM {
   ....
   CONN_PORT,
   CONN_ADDRESS,
   ....
};

enum class ISCSI_HOST {
   ....
   PARAM_IPADDRESS,
   ....
};
```

```
int iscsi_conn_get_addr_param(....,
 ISCSI_PARAM param, ....)
{
   ....
   switch (param) {
   case ISCSI_PARAM::CONN_ADDRESS:
   case ISCSI_HOST::PARAM_IPADDRESS:
   ....
   }

   return len;
}
```

enum class

ReactOS

```
void adns__querysend_tcp(....) {
   ...
   if (!(errno == EAGAIN || EWOULDBLOCK ||
         errno == EINTR || errno == ENOSPC ||
         errno == ENOBUFS || errno == ENOMEM)) {
   ...
}
```

Инициализация в конструкторе

```
Guess::Guess() {
  language_str = DEFAULT_LANGUAGE;
  country_str = DEFAULT_COUNTRY;
  encoding_str = DEFAULT_ENCODING;
}
Guess::Guess(const char * guess_str) {
  Guess();
  ....
}
```

LibreOffice

Предупреждение PVS-Studio: V603 The object was created but it is not being used. If you wish to call constructor, 'this->Guess::Guess(....)' should be used. guess.cxx 56

Инициализация в конструкторе

Можно написать так:

```
Guess::Guess(const char * guess_str)
{
  this->Guess();
  ....
}

Guess::Guess(const char * guess_str)
{
  Init();
  ....
}
```

Инициализация в конструкторе

А можно так:

```
Guess::Guess(const char * guess_str) : Guess()
{
   ....
}
```

Как нельзя использовать делегирующие конструкторы

```
Guess::Guess(const char * guess_str)
   : Guess(),
     m_member(42)
{
   ....
}
```

Как нельзя использовать делегирующие конструкторы

```
Guess::Guess(const char * guess_str)
  : Guess(std::string(guess_str))
{
  ....
}
Guess::Guess(std::string guess_str)
  : Guess(guess_str.c_str())
{
  ....
}
```

О виртуальных функциях

```
class Base {
  virtual void Foo(int x);
}

class Derived : public class Base {
  void Foo(int x, int a = 1);
}
```

О виртуальных функциях

MongoDB

```
class DBClientBase : .... {
public:
  virtual auto_ptr<DBClientCursor> query(
    const string &ns,
    Query query,
    int nToReturn = 0
    int nToSkip = 0,
    const BSONObj *fieldsToReturn = 0,
    int queryOptions = 0,
    int batchSize = 0 );
};
```

О виртуальных функциях

```
class DBDirectClient : public DBClientBase {
public:
  virtual auto_ptr<DBClientCursor> query(
    const string &ns,
    Query query,
    int nToReturn = 0,
    int nToSkip = 0,
    const BSONObj *fieldsToReturn = 0,
    int queryOptions = 0);
};
```

О виртуальных функциях

```
class CWnd : public CCmdTarget {
  ....
  virtual void WinHelp(DWORD_PTR dwData, UINT nCmd = HELP_CONTEXT);
  ....
};

class CFrameWnd : public CWnd { .... };

class CFrameWndEx : public CFrameWnd {
  ....
  virtual void WinHelp(DWORD dwData, UINT nCmd = HELP_CONTEXT);
  ....
};
```

override

```
class DBDirectClient : public DBClientBase {
public:
  virtual auto_ptr<DBClientCursor> query(
    const string &ns,
    Query query,
    int nToReturn = 0,
    int nToSkip = 0,
    const BSONObj *fieldsToReturn = 0,
    int queryOptions = 0) override;
};
```

nullptr vs NULL

```
void Foo(int x, int y, const char *name);
void Foo(int x, int y, int ResourceID);
Foo(1, 2, NULL);

if (WinApiFoo(a, b) != NULL)     // Плохо
if (WinApiFoo(a, b) != nullptr) // Ура, ошибка
                                // компиляции
```

va_arg...

```
typedef std::wstring string16;
const base::string16& relaunch_flags() const;


int RelaunchChrome(const DelegateExecuteOperation&
operation)
{
  AtlTrace("Relaunching [%ls] with flags [%s]\n",
           operation.mutex().c_str(),
           operation.relaunch_flags());
  ....
}
```

Chromium

Предупреждение PVS-Studio: V510 The 'AtlTrace' function is not expected to receive class-type variable as third actual argument. delegate_execute.cc 96

va_arg...

```
cairo_status_t
_cairo_win32_print_gdi_error (const char *context)
{
  ....
  fwprintf (stderr, L"%s: %S", context,
            (wchar_t *)lpMsgBuf);
  ....
}
```

Cairo

Предупреждение PVS-Studio: V576 Incorrect format. Consider checking the third actual argument of the 'fwprintf' function. The pointer to string of wchar_t type symbols is expected. cairo-win32-surface.c 130

va_arg...

```
static void GetNameForFile(
  const char* baseFileName,
  const uint32 fileIdx,
  char outputName[512] )
{
  assert(baseFileName != NULL);
  sprintf( outputName, "%s_%d", baseFileName, fileIdx );
}
```

CryEngine 3 SDK

V576 Incorrect format. Consider checking the fourth actual argument of the 'sprintf' function. The SIGNED integer type argument is expected. igame.h 66

va_arg...

```
ReadAndDumpLargeSttb(cb,err)
  int     cb;
  int     err;
{
  ....
  printf("\n - %d strings were read, "
         "%d were expected (decimal numbers) -\n");
  ....
}
```

Word for Windows 1.1a

Предупреждение PVS-Studio: V576 Incorrect format. A different number of actual arguments is expected while calling 'printf' function. Expected: 3. Present: 1. dini.c 498

va_arg...

```
BOOL CALLBACK EnumPickIconResourceProc(
  HMODULE hModule, LPCWSTR lpszType,
  LPWSTR lpszName, LONG_PTR lParam)
{
  ....
  swprintf(szName, L"%u", lpszName);
  ....
}
```

ReactOS

Предупреждение PVS-Studio: V576 Incorrect format. Consider checking the third actual argument of the 'swprintf' function. To print the value of pointer the '%p' should be used. dialogs.cpp 66

Чем же плохи va_arg функции?

- Нельзя проверить тип аргумента
- Нельзя проверить количество аргументов
- Шаг вправо, шаг влево – undefined behavior

Что стоит использовать в C++11?

1) variadic templates

template <typename... T>

```
void Foo(T... args);
Foo(1, 2, 3, 4, 5, 6);
```

2) std::initializer_list

```
void Foo(std::initializer_list<int> args);
Foo({1, 2, 3, 4, 5, 6});
```

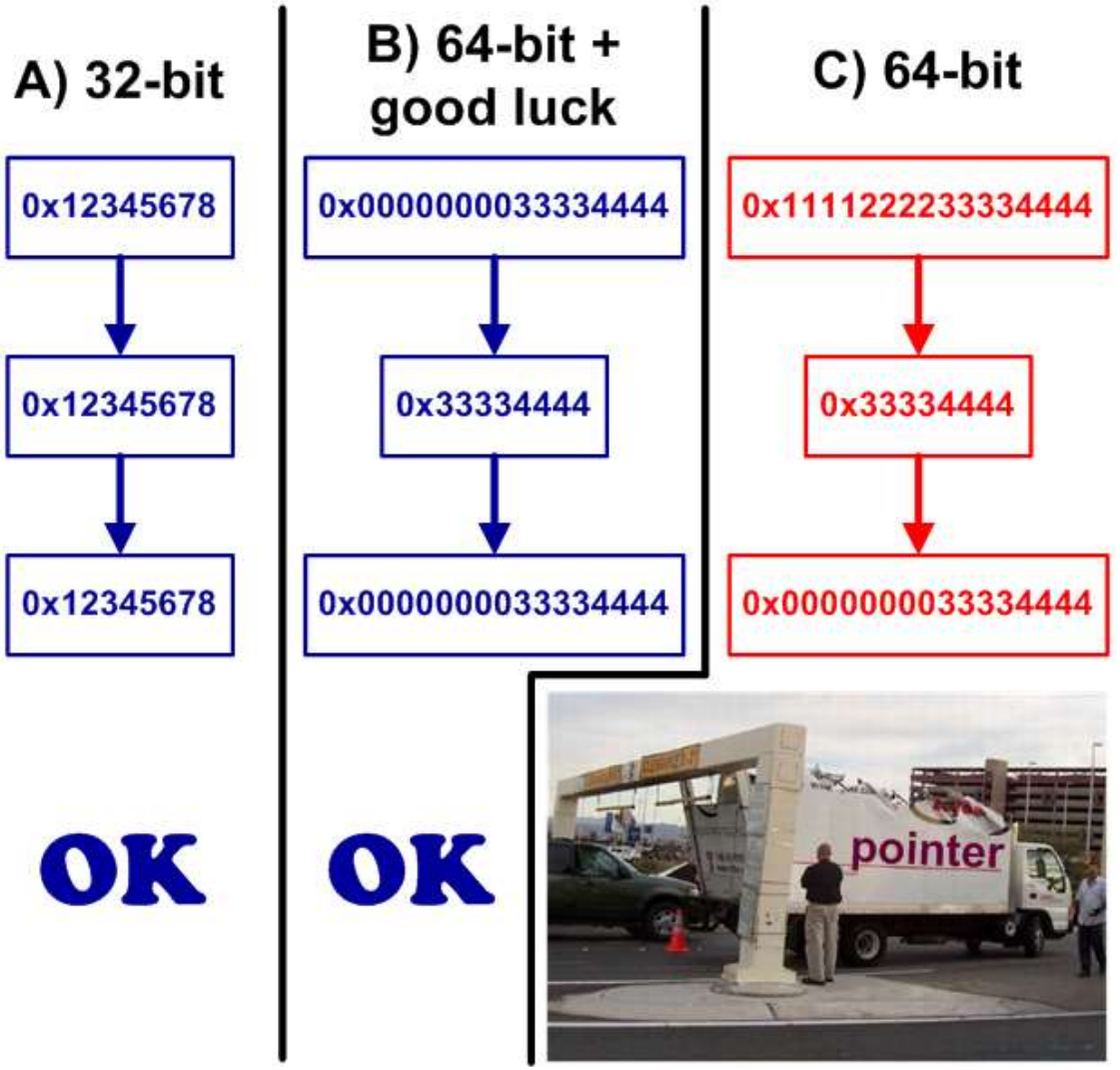
variadic templates

```
void printf(const char* s) {
  std::cout << s;
}

template<typename T, typename... Args>
void printf(const char* s, T value, Args... args) {
  while (s && *s) {
    if (*s=='%' && *++s!='%') {
      std::cout << value;
      return printf(++s, args...);
    }
    std::cout << *s++;
  }
}
```

narrowing

```
char* ptr = …;
int n = (int)ptr;
....
ptr = (char*) n;
```



narrowing

```
virtual int GetMappingWidth( ) = 0;
virtual int GetMappingHeight( ) = 0;

void CDetailObjectSystem::LevelInitPreEntity()
{
   ....
   float flRatio = pMat->GetMappingWidth() /
                   pMat->GetMappingHeight();
   ....
}
```

Source Engine SDK

Предупреждение PVS-Studio: V636 The expression was implicitly cast from 'int' type to 'float' type. Consider utilizing an explicit type cast to avoid the loss of a fractional part. An example: double A = (double)(X) / Y;. Client (HL2) detailobjectsystem.cpp 1480

narrowing

```
virtual int GetMappingWidth( ) = 0;
virtual int GetMappingHeight( ) = 0;

void CDetailObjectSystem::LevelInitPreEntity()
{
   ....
   float flRatio { pMat->GetMappingWidth() /
                   pMat->GetMappingHeight() };
   ....
}
```

No news is good news

```
void AccessibleContainsAccessible(....)
{
  auto_ptr<VARIANT> child_array(
            new VARIANT[child_count]);
  ...
}
```

Chromium

Предупреждение PVS-Studio: V554 Incorrect use of auto_ptr. The memory allocated with 'new []' will be cleaned using 'delete'. interactive_ui_tests accessibility_win_browsertest.cc 171

std::unique_ptr

```
void text_editor::_m_draw_string(....) const
{
   ....
   std::unique_ptr<unsigned> pxbuf_ptr(
         new unsigned[len]);
   ....
}
```

nana

Предупреждение PVS-Studio: V554 Incorrect use of unique_ptr. The memory allocated with 'new []' will be cleaned using 'delete'. text_editor.cpp 3137

std::unique_ptr

```
void text_editor::_m_draw_string(....) const
{
   ....
   std::unique_ptr<unsigned[]> pxbuf_ptr(
         new unsigned[len]);
   ....
}
```

No news is good news

```
template<class TOpenGLStage>
static FString GetShaderStageSource(TOpenGLStage*
Shader)
{
  ....
  ANSICHAR* Code = new ANSICHAR[Len + 1];
  glGetShaderSource(Shaders[i], Len + 1, &Len, Code);
  Source += Code;
  delete Code;
  ....
}
```

Unreal Engine 4

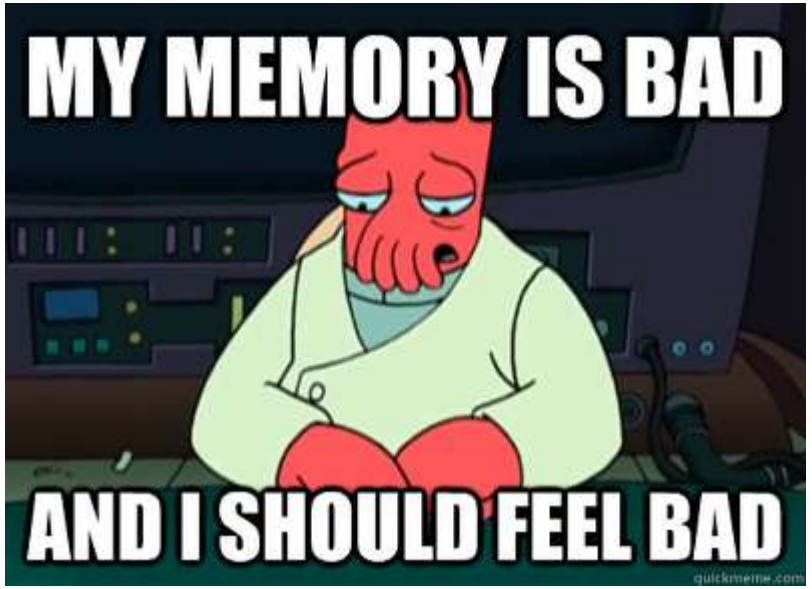
Предупреждение PVS-Studio: V611 The memory was allocated using 'new T[]' operator but was released using the 'delete' operator. Consider inspecting this code. It's probably better to use 'delete [] Code;'. openglshaders.cpp 1790

No news is good news

```
bool CxImage::LayerCreate(int32_t position)
{
   ....
   CxImage** ptmp = new CxImage*[info.nNumLayers + 1];
   ....
   free(ptmp);
   ....
}
```

CxImage

Предупреждение PVS-Studio: V611 The memory was allocated using 'new' operator but was released using the 'free' function. Consider inspecting operation logics behind the 'ptmp' variable. ximalyr.cpp 50



No news is good news

```
int settings_proc_language_packs(....)
{
  ....
  if(mem_files) {
    mem_files = 0;
    sys_mem_free(mem_files);
  }
  ....
}
```

Fennec Media

Предупреждение PVS-Studio: V575 The null pointer is passed into 'free' function. Inspect the first argument. settings interface.c 3096

Ошибки свойственны не только для new

```
ETOOLS_API int __stdcall ogg_enc(....) {
  format = open_audio_file(in, &enc_opts);
  if (!format) {
    fclose(in);
    return 0;
  };
  out = fopen(out_fn, "wb");
  if (out == NULL) {
    fclose(out);
    return 0;
  }
}
```

RAII

```
void Cpp11()
{
  auto deleter = [](FILE* f) {fclose(f);};
  std::unique_ptr<FILE, decltype(deleter)> p(fopen("1.txt", "w"),
                                             deleter);
  ....
}

void Cpp14()
{
  auto deleter = [](FILE* f) {fclose(f);};
  auto p = std::make_unique(fopen("1.txt", "w"), deleter);
  ....
}
```

Что в итоге?

- Да, с помощью Modern C++ можно избавиться от ряда неприятных ошибок
- Можно выкинуть старые костыли, а-ля #define countof
- Избавиться от ручного управления памятью

Что в итоге?

Вместе с новыми стандартами пришли и новые ошибки.

Да и старые не все исчезли.

Что делать? То же самое:

- тесты
- код-ревью
- и статический анализ, конечно

PVS-Studio

- Более 350 диагностик для C++
- Более 100 диагностик для C#
- Интеграция с Visual Studio
- Поддержка любой билд-системы
- Работает в Windows и Linux

Ответы на вопросы

E-Mail: belikov@viva64.com

PVS-Studio: http://www.viva64.com/ru/pvs-studio/

Наш блог на хабре: https://habrahabr.ru/company/pvs-studio/