

C# разработка и статический анализ

В чем практическая польза?

Глеб Асламов

C# Developer & Developer Advocate



Статический анализ: теория и практика



Статический анализ: теория и практика

SAST: безопасность и защищенность



Статический анализ: теория и практика

SAST: безопасность и защищенность

ГОСТ Р 71207–2024



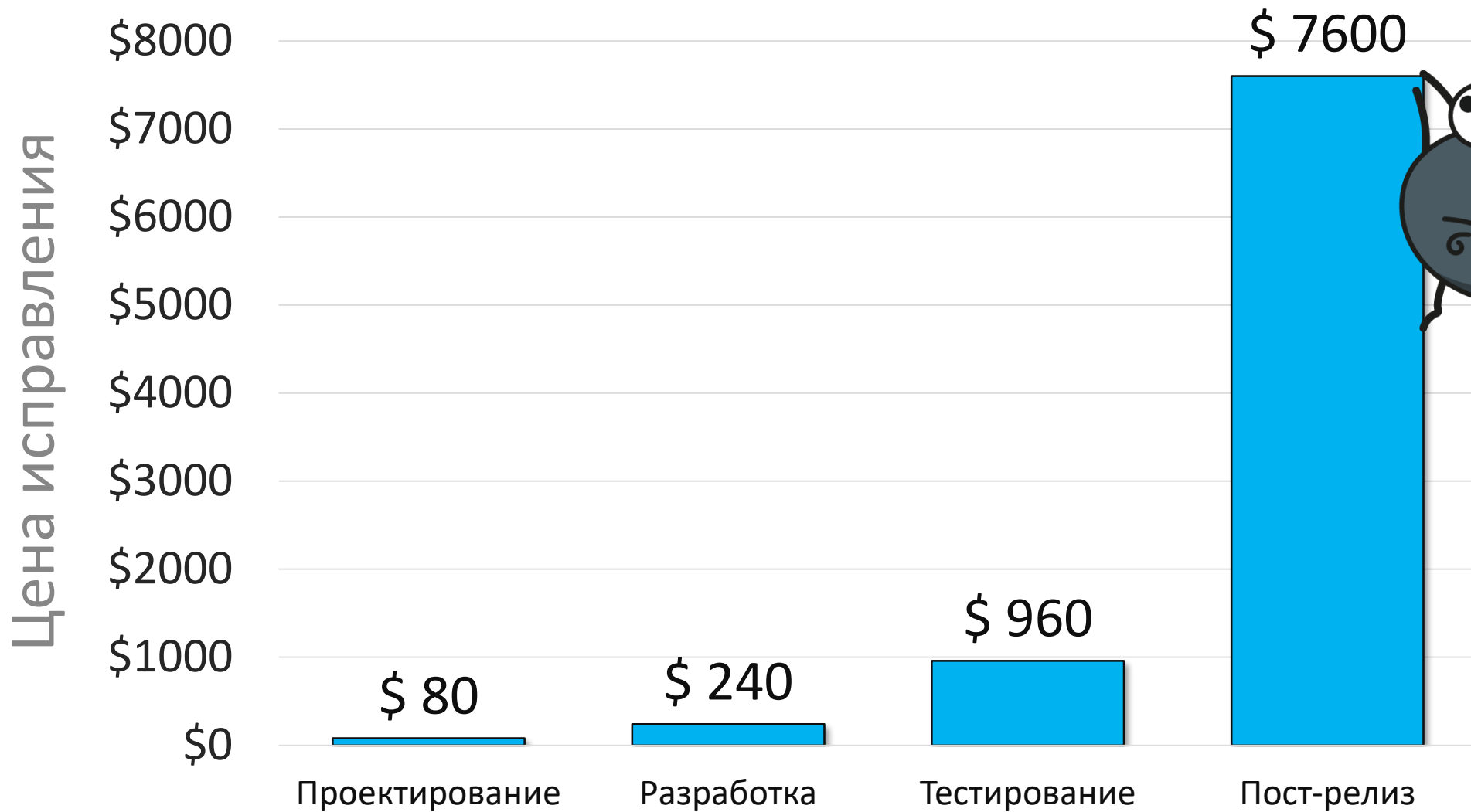
Проблематика



Рост плотности ошибок

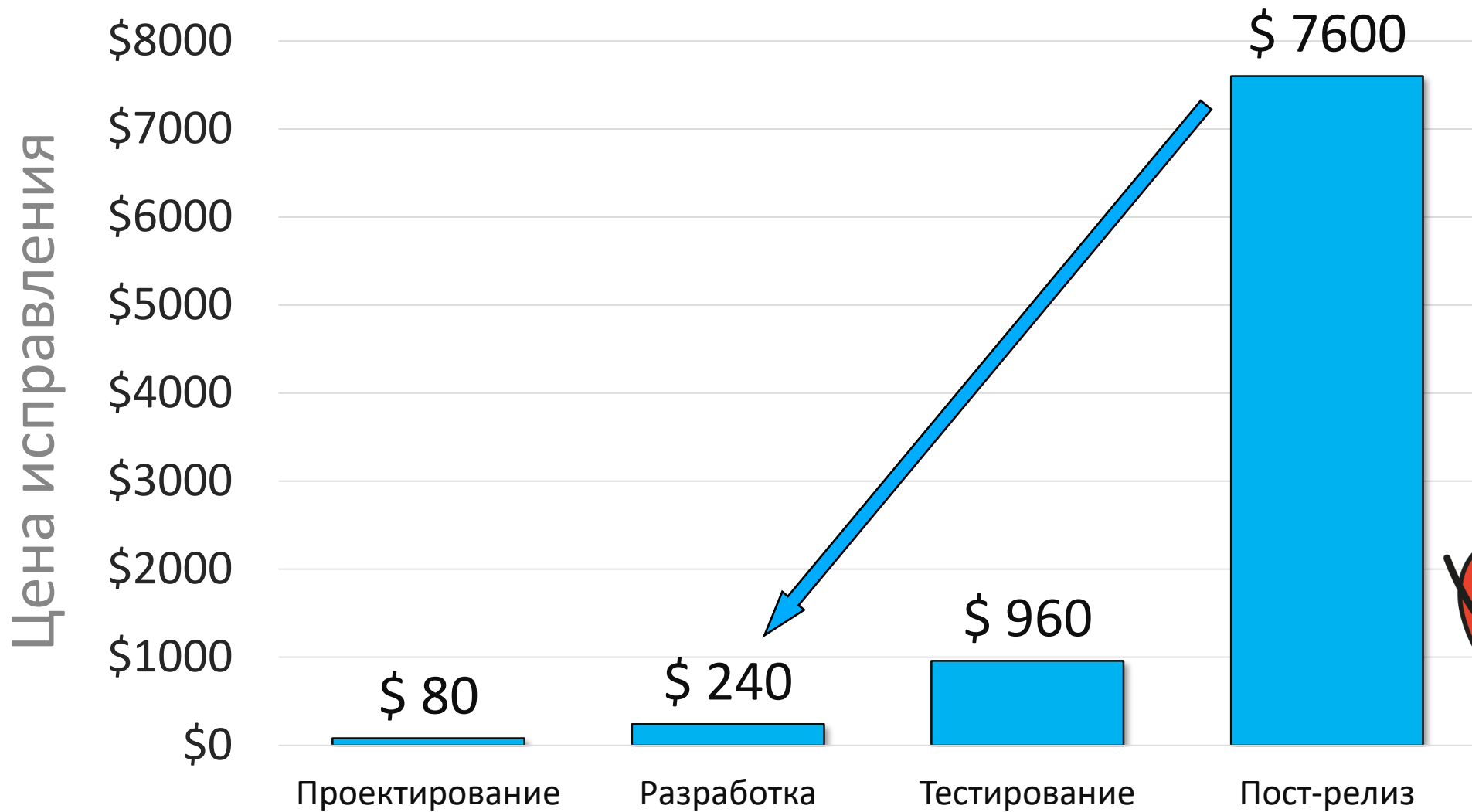
Размер проекта (число строк)	Типичная плотность ошибок
Менее 2К	0 – 25 ошибок на 1000 строк кода
2К-16К	0 – 40 ошибок на 1000 строк кода
16К-64К	1 – 50 ошибок на 1000 строк кода
64К-512К	2 – 70 ошибок на 1000 строк кода
512К и более	4 – 100 ошибок на 1000 строк кода

Сколько стоит исправить уязвимость?



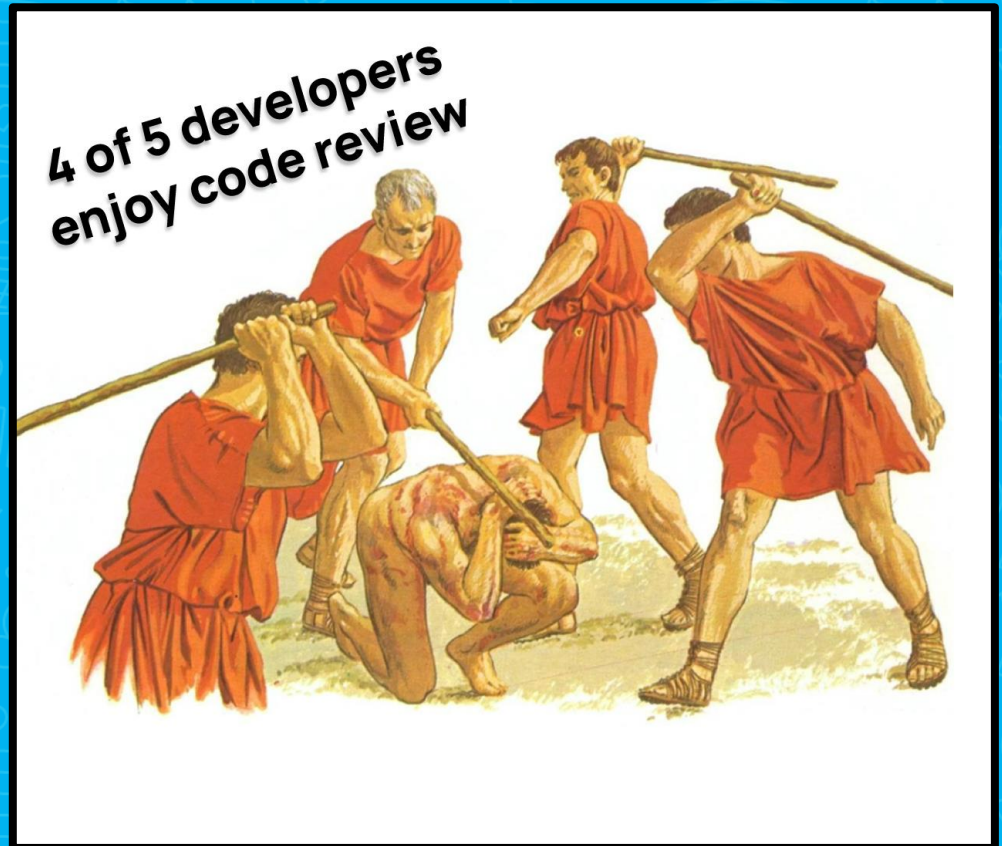
Источник - [NIST](#): National Institute of Standards and Technology

Сколько стоит исправить уязвимость?



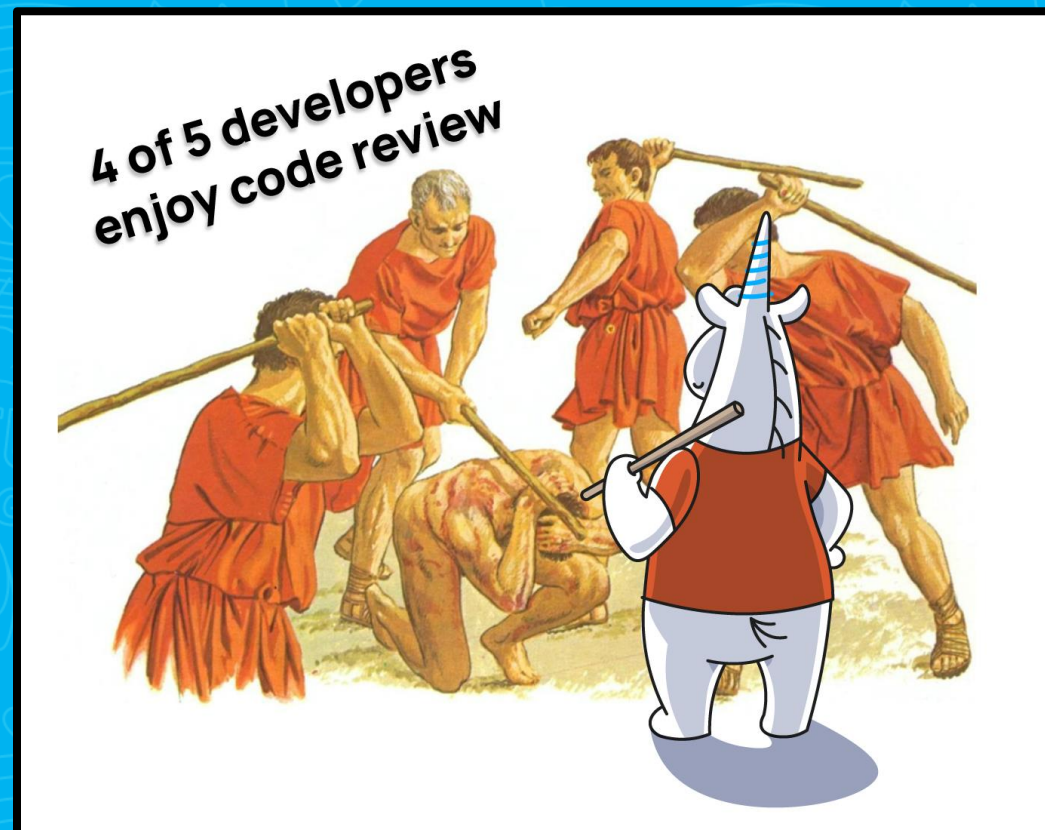
Статический анализ

- В начале был код-ревью



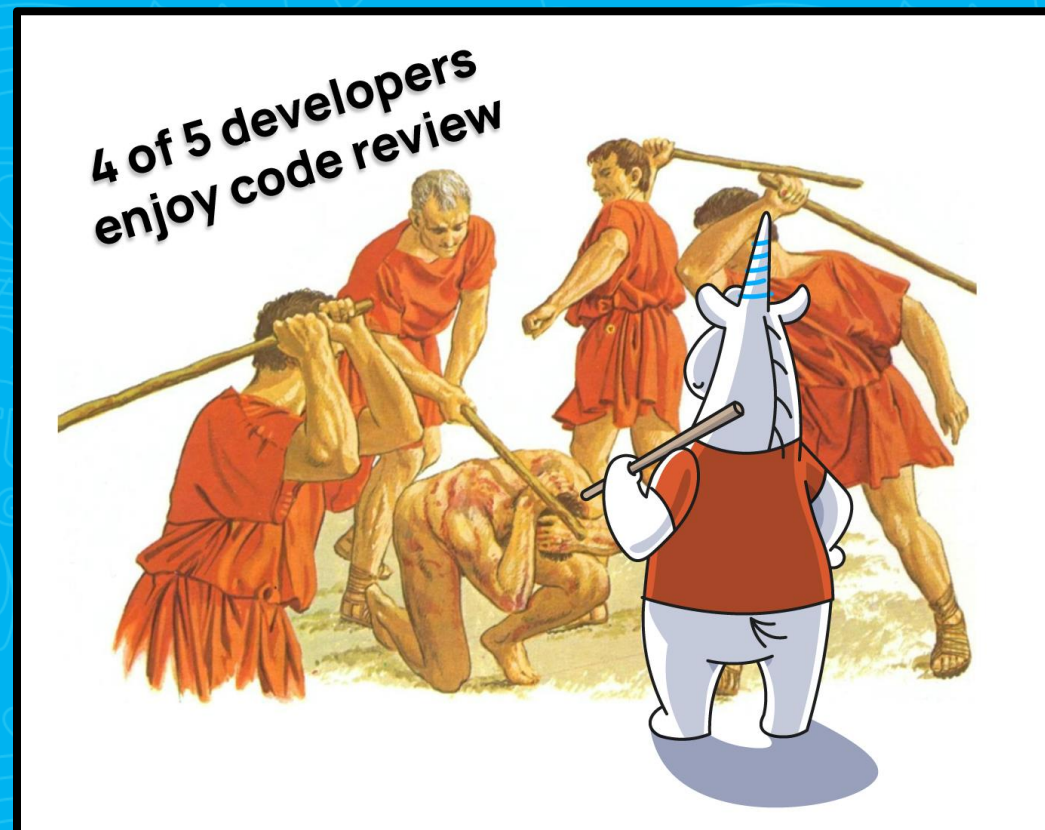
Статический анализ

- Автоматический код-ревью!



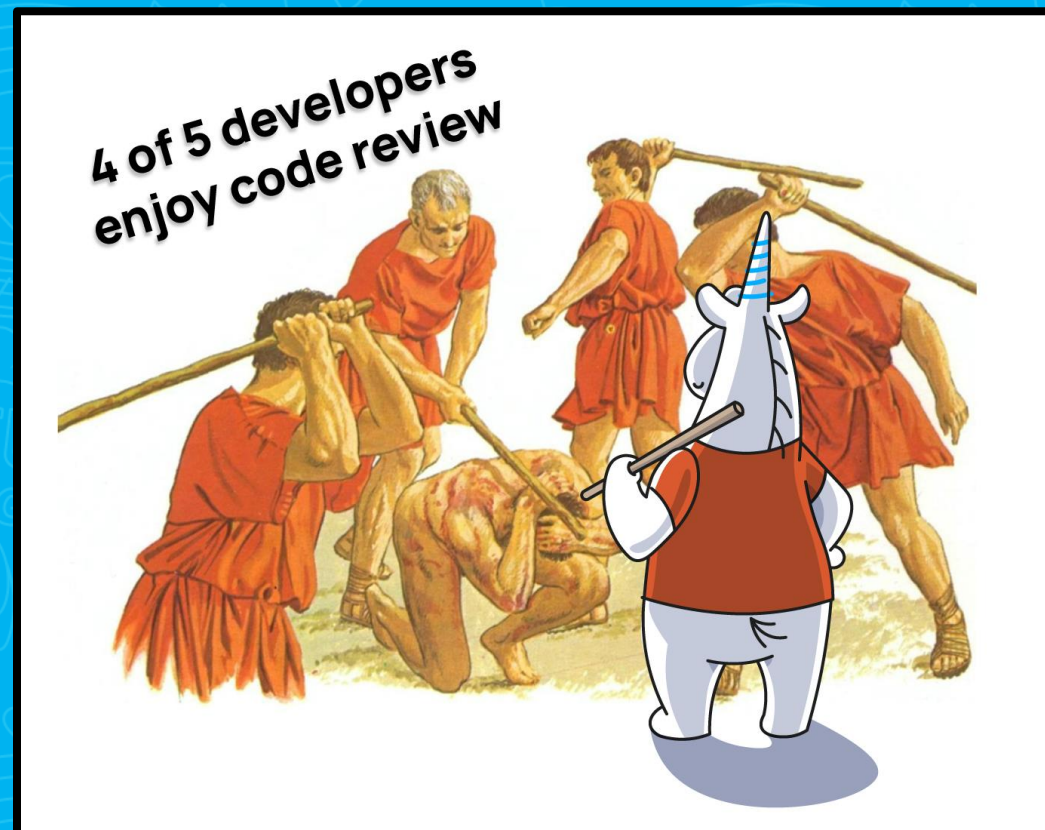
Статический анализ

- Автоматический код-ревью!
- Нужен только код



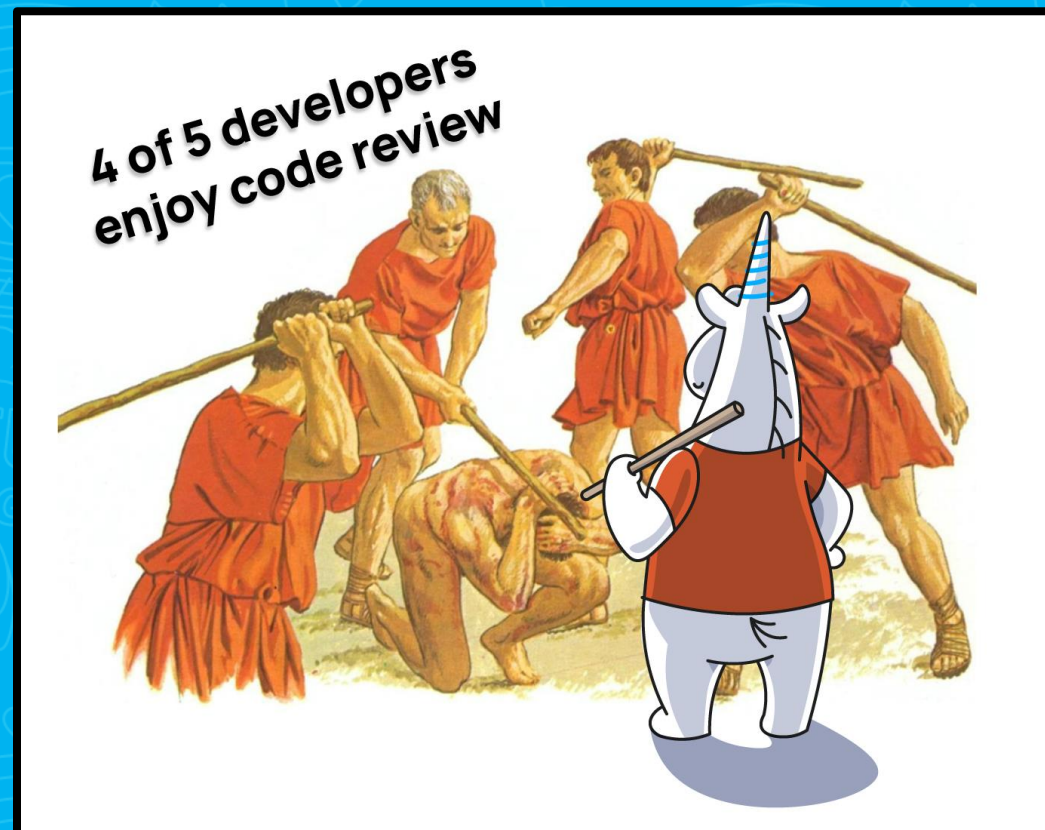
Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие



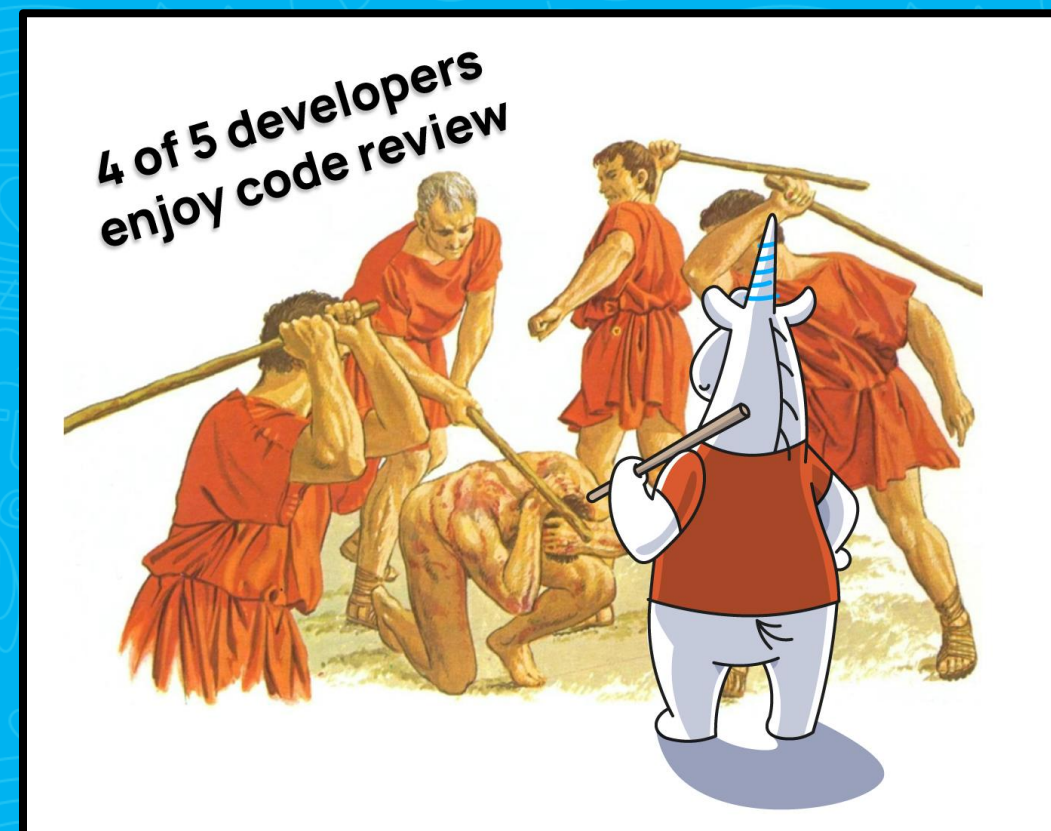
Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Ранее обнаружение ошибок



Статический анализ

- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок
- Ошибки исправляются на этапе разработки



Виды проблем

проблемы
безопасности

неправильная работа
с методами

недостижимый
код

ошибки доступа к
памяти

опечатки



неправильная работа с
типами

ошибки сериализации /
десериализации

выход за
границы

ошибки
синхронизации

переполнение
буфера

Azure SDK

```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        ExpiryTime == other.ExpiryTime &&  
        Identifier == other.Identifier &&  
        IPRange == other.IPRange &&  
        Permissions == other.Permissions &&  
        Protocol == other.Protocol &&  
        StartTime == other.StartTime &&  
        Version == other.Version;  
}
```


Azure SDK

```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        ExpiryTime == other.ExpiryTime &&  
        Identifier == other.Identifier &&  
        IPRange == other.IPRange &&  
        Permissions == other.Permissions &&  
        Protocol == other.Protocol &&  
        StartTime == other.StartTime &&  
        Version == other.Version;  
}
```

Azure SDK

```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        StartTime == other.StartTime &&  
        Version == other.Version;  
}
```

Предупреждение PVS-Studio:

V3112 An abnormality within similar comparisons. It is possible that a typo is present inside the expression 'ContentLanguage == other.ContentEncoding'

Azure SDK

```
public struct FileSasBuilder : IEquatable<FileSasBuilder> {  
    ....  
    public bool Equals(FileSasBuilder other) =>  
        CacheControl == other.CacheControl  
        && ContentDisposition == other.ContentDisposition  
        && ContentEncoding == other.ContentEncoding  
        && ContentLanguage == other.ContentEncoding  
        && ContentType == other.ContentType  
        && ExpiryTime == other.ExpiryTime  
        && FilePath == other.FilePath  
        && Identifier == other.Identifier  
        && IPRange == other.IPRange  
        && Permissions == other.Permissions  
        && Protocol == other.Protocol  
        && ShareName == other.ShareName  
        && StartTime == other.StartTime  
        && Version == other.Version  
}
```

Azure SDK

```
public struct FileSasBuilder : IEquatable<FileSasBuilder> {  
    ....  
    public bool Equals(FileSasBuilder other) =>  
        CacheControl == other.CacheControl  
        && ContentDisposition == other.ContentDisposition  
        && ContentEncoding == other.ContentEncoding  
        && ContentLanguage == other.ContentEncoding  
        && ContentType == other.ContentType  
        && ExpiryTime == other.ExpiryTime  
        && FilePath == other.FilePath  
        && Identifier == other.Identifier  
        && IPRange == other.IPRange  
        && Permissions == other.Permissions  
        && Protocol == other.Protocol  
        && ShareName == other.ShareName  
        && StartTime == other.StartTime  
        && Version == other.Version
```


BTCPay Server Check

```
if ( request.PaymentTolerance < 0
    && request.PaymentTolerance > 100)
ModelState.AddModelError(nameof(request.PaymentTolerance),
    "PaymentTolerance can only be between 0 and 100 percent");
```

BTCPay Server Check

```
if ( request.PaymentTolerance < 0
```

```
    && request.PaymentTolerance > 100)
```

```
    ModelState.AddModelError(nameof(request.PaymentTolerance),  
        "PaymentTolerance can only be between 0 and 100 percent");
```

Предупреждение PVS-Studio:

V3022 Expression 'request.PaymentTolerance < 0 && request.PaymentTolerance > 100' is always false. Probably the '||' operator should be used here.

MudBlazor

```
var exp = new Expression(CalcExpression);  
var result = exp.Eval();  
  
if (result == double.NaN)  
{  
    Current = "ERROR";  
    return;  
}
```

MudBlazor

```
var exp = new Expression(CalcExpression);  
var result = exp.Eval();
```

```
if (result == double.NaN)  
{  
    Current = "ERROR";
```

Предупреждение PVS-Studio:

```
} PVS-Studio: V3076 Comparison of 'result' with 'double.NaN' is meaningless.  
Use 'double.IsNaN()' method instead.
```


Entity Framework

```
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[j]));
    }
}
```

Entity Framework

```
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]) );
    }
}
```

eShopOnContainers

```
....  
var firstUrl = new Uri(urlHook, UriKind.Absolute);  
var secondUrl = new Uri(url, UriKind.Absolute);  
  
return firstUrl.Scheme == secondUrl.Scheme &&  
       firstUrl.Port == secondUrl.Port &&  
       firstUrl.Host == firstUrl.Host;
```

eShopOnContainers

```
....  
var firstUrl = new Uri(urlHook, UriKind.Absolute);  
var secondUrl = new Uri(url, UriKind.Absolute);  
  
return firstUrl.Scheme == secondUrl.Scheme &&  
       firstUrl.Port == secondUrl.Port &&  
       firstUrl.Host == firstUrl.Host;
```


eShopOnContainers

```
....  
var firstUrl = new Uri(urlHook, UriKind.Absolute);  
var secondUrl = new Uri(url, UriKind.Absolute);  
  
return firstUrl.Scheme == secondUrl.Scheme &&  
       firstUrl.Port == secondUrl.Port &&  
       firstUrl.Host == firstUrl.Host;
```

Предупреждение PVS-Studio:

V3001 There are identical sub-expressions 'firstUrl.Host' to the left and to the right of the '==' operator.

AWS SDK for .NET


```
var parts = path.Split('.');
for (int i = 0; i < parts.Length; i++)
{
    var part = parts[i];
    if (!(propertyValue is IList))
        throw new ArgumentException(
            "Object addressing by pathing segment '{part}'
            with indexer must be IList");

    if (!(propertyValue is IPropertyBag))
        throw new ArgumentException(
            "Object addressing by pathing segment '{part}'
            must be IPropertyBag");
    ....
}
```

AWS SDK for .NET

```
var parts = path.Split('.');
for (int i = 0; i < parts.Length; i++)
{
    var part = parts[i];
    if (!(propertyValue is IList))
        throw new ArgumentException(
            $"Object addressing by pathing segment '{part}'
                with indexer must be IList");

    if (!(propertyValue is IPropertyBag))
        throw new ArgumentException(
            $"Object addressing by pathing segment '{part}'
                must be IPropertyBag");
    ....
}
```

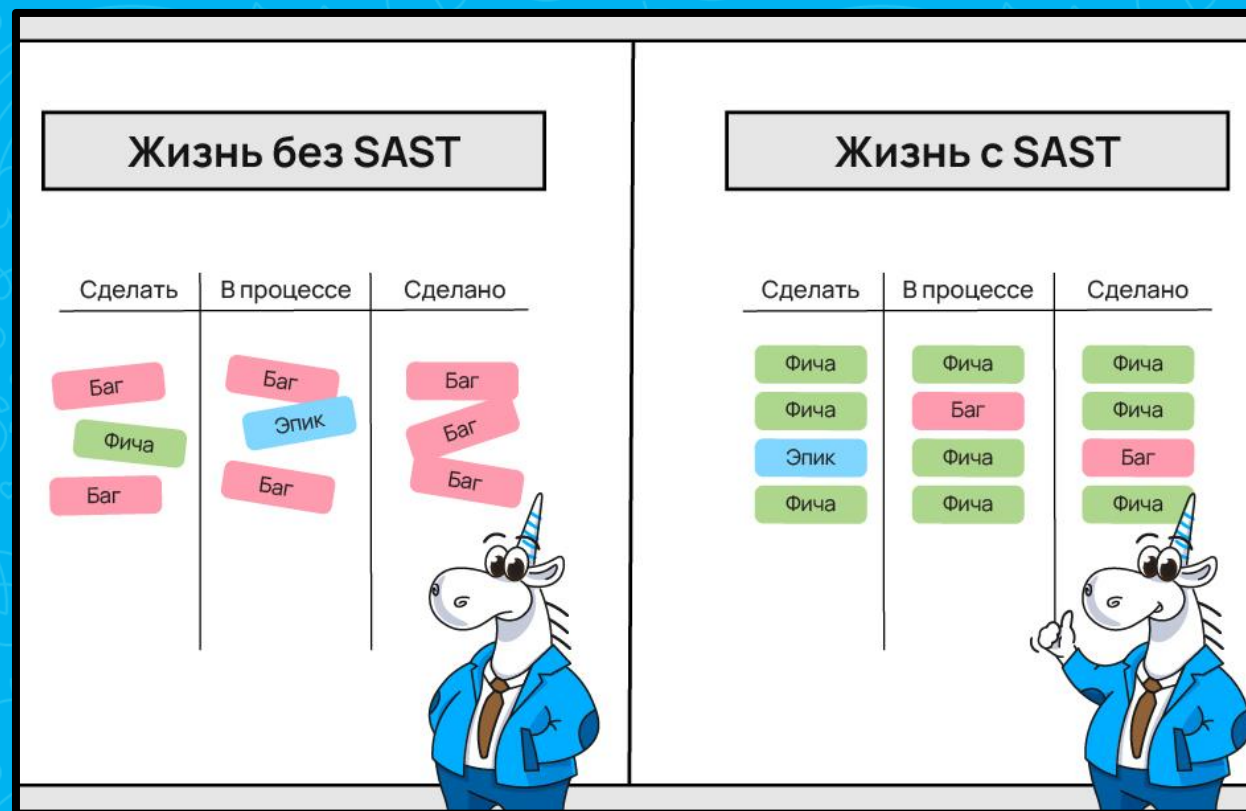


SAST: безопасность и защищенность

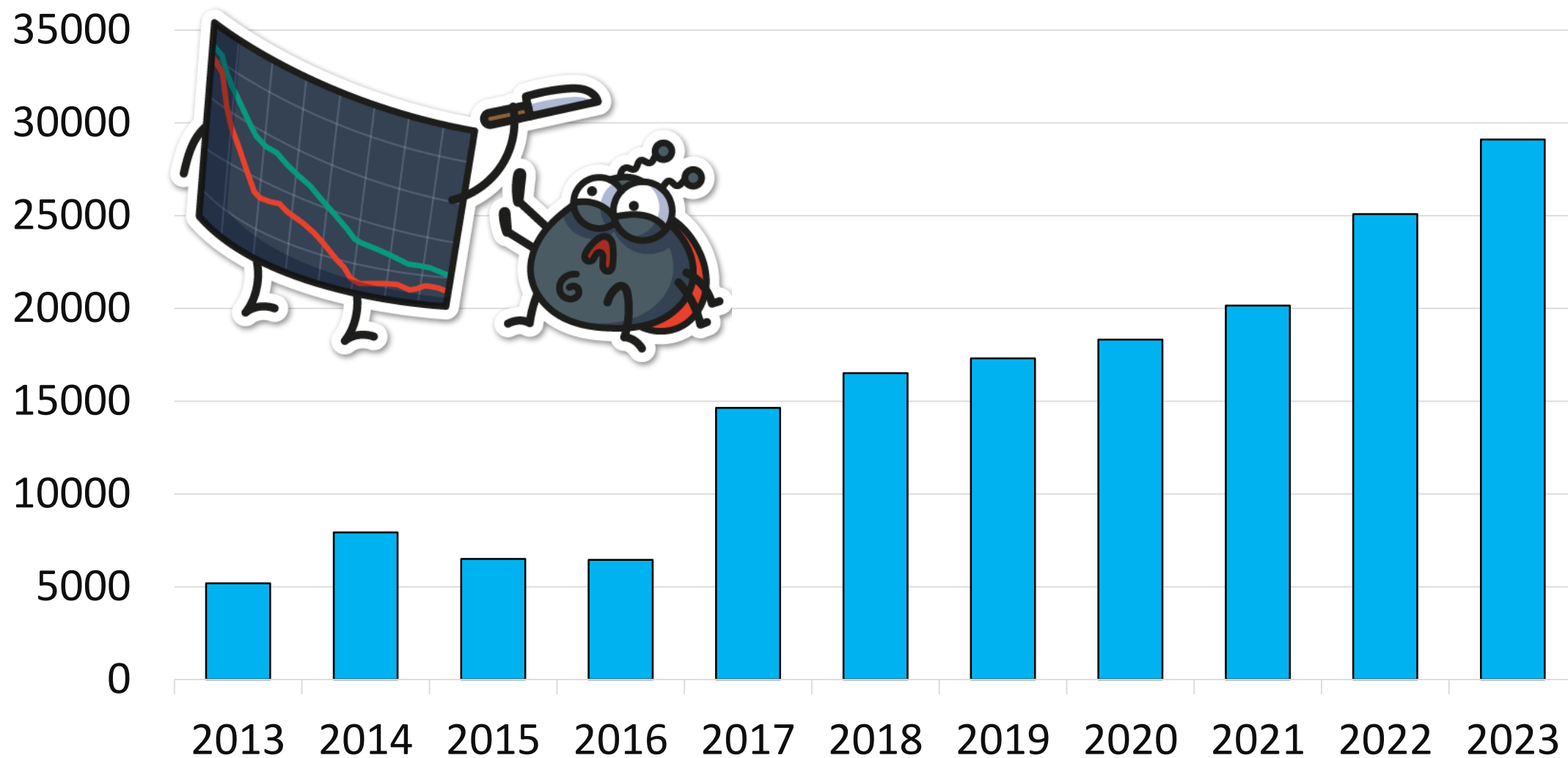


Что такое SAST?

- Static Application Security Testing
- Статический анализ, но про уязвимости
- Уязвимости – это обычные ошибки, но последствия хуже



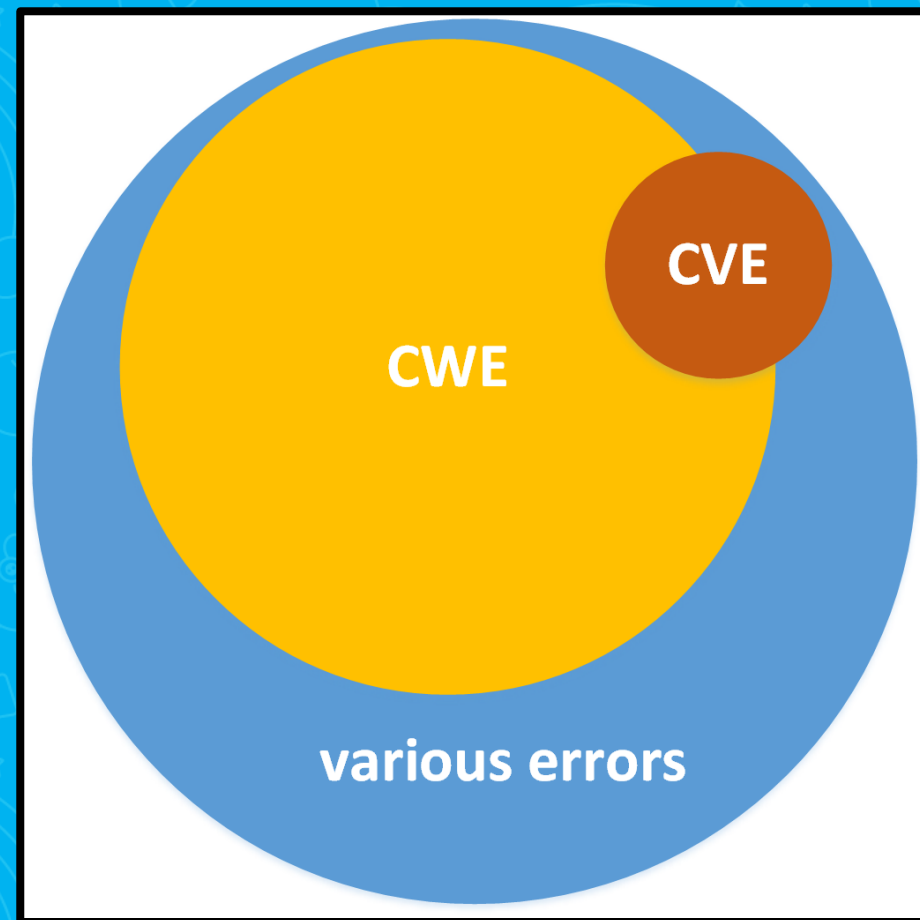
Число выявленных уязвимостей



Списки уязвимостей

CWE – Потенциальные уязвимости
Common Weakness Enumeration

CVE - Существующие уязвимости
Common Vulnerabilities and Exposures



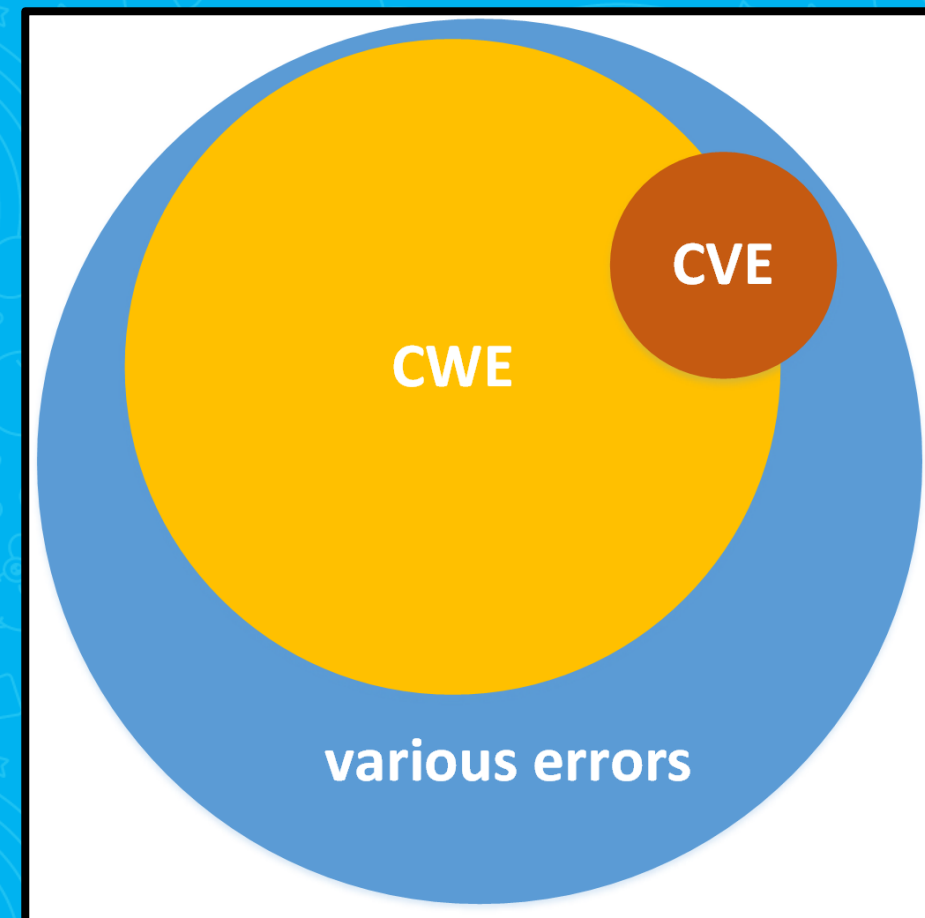
Списки уязвимостей

CWE – Потенциальные уязвимости

Common Weakness Enumeration

CVE - Существующие уязвимости

Common Vulnerabilities and Exposures



Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;
```

```
....
```

```
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```



Пример CWE в проекте FastReport

ParagraphFormat paragraphFormat; Поле

....

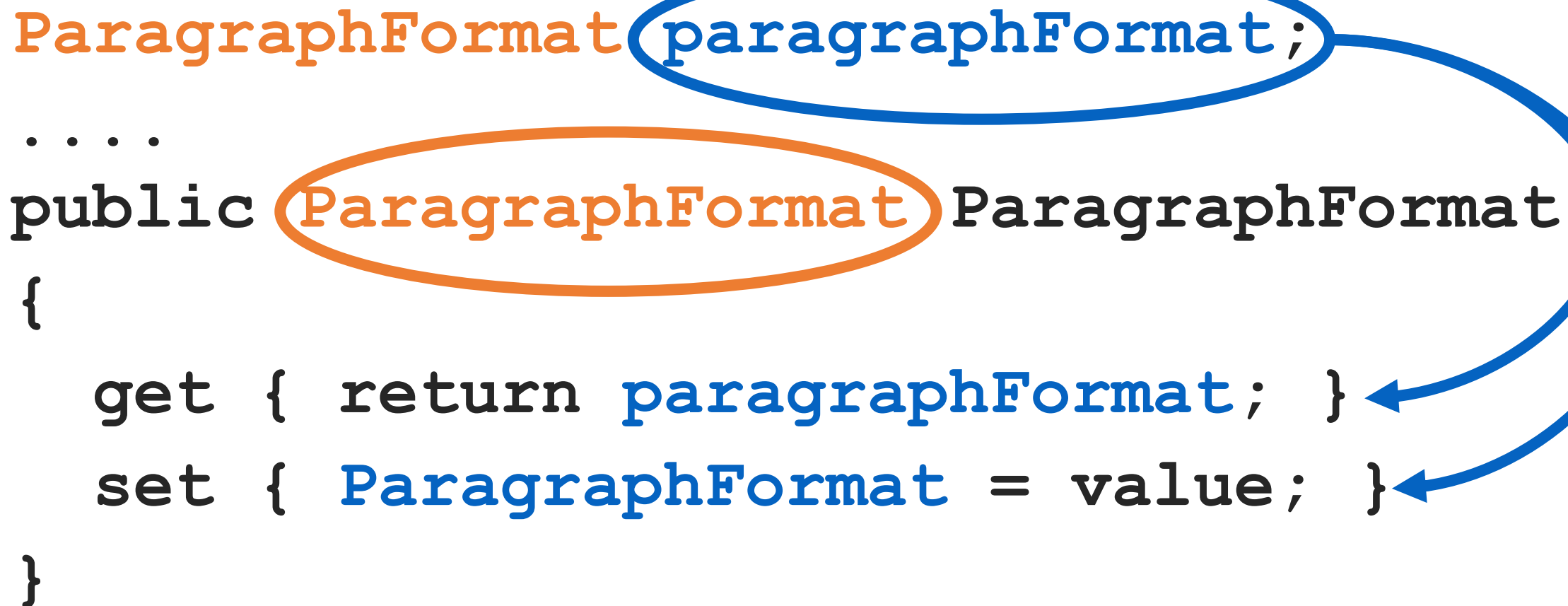
```
public ParagraphFormat ParagraphFormat
{
    get { return paragraphFormat; }
    set { ParagraphFormat = value; }
}
```

Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat; Поле  
.....  
public ParagraphFormat ParagraphFormat  
{  
    СВОЙСТВО  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```

Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```

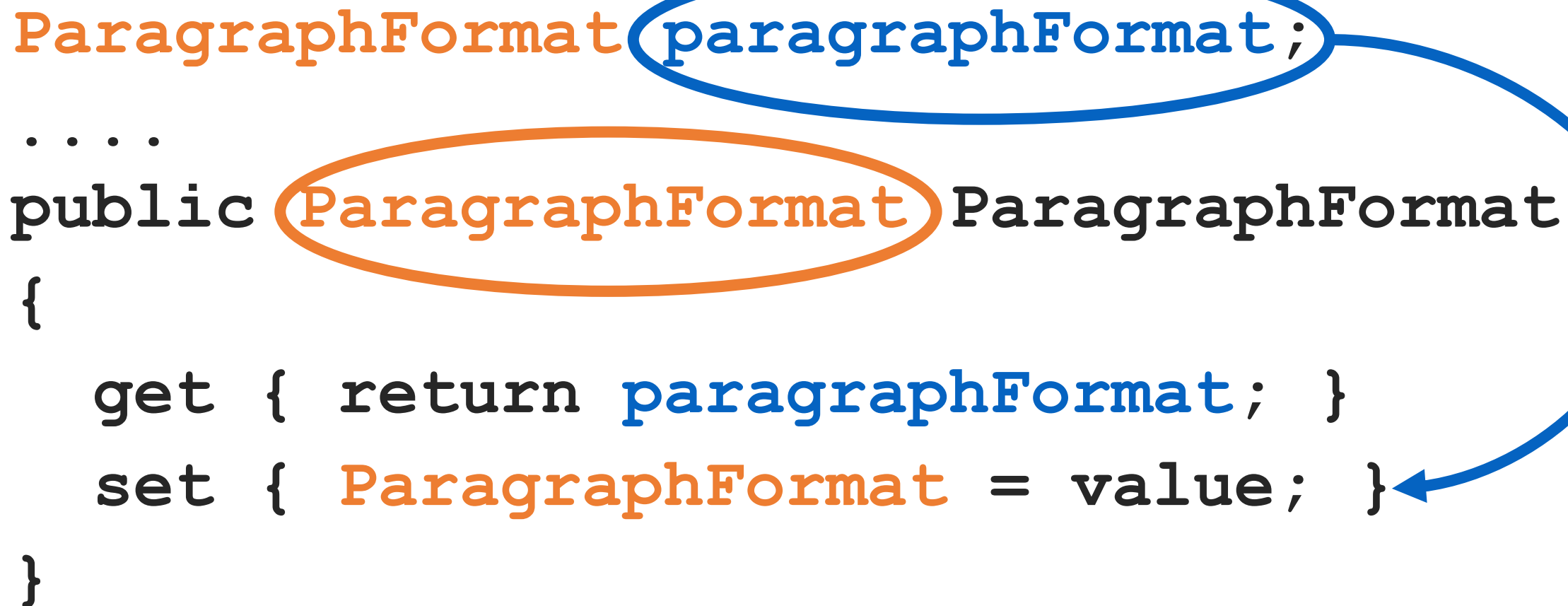


Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```

Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }  
}
```



Пример CWE в проекте FastReport

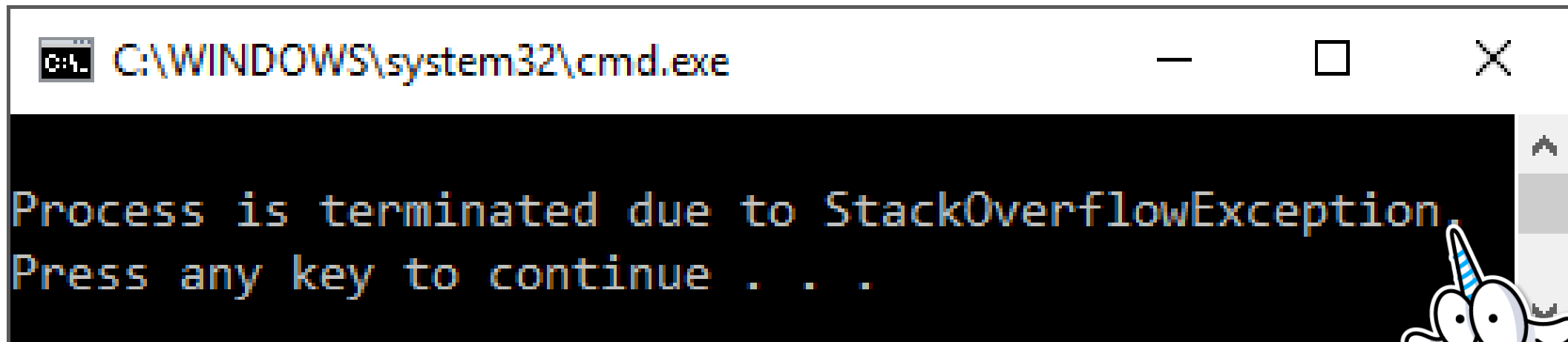
```
static void Main(string[] args)
{
    TextObject textObj = new TextObject();
    textObj.ParagraphFormat = null;

    Console.WriteLine("Ok");
}
```

Пример CWE в проекте FastReport

```
static void Main(string[] args)
```

```
{
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The main content of the window is a black background with white text that reads: "Process is terminated due to StackOverflowException. Press any key to continue . . .".

```
Console.WriteLine("Ok");
```

```
}
```



Пример CWE в проекте FastReport

```
ParagraphFormat paragraphFormat;  
...  
public ParagraphFormat ParagraphFormat  
{  
    get { return paragraphFormat; }  
    set { ParagraphFormat = value; }
```

Предупреждение PVS-Studio: V3010 [[CWE-674](#)]

Possible infinite recursion inside 'ParagraphFormat' property.

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = '" + userName + "'",
        CommandType = System.Data.CommandType.Text })
        ....
    }
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = '" + userName + "'",
        CommandType = System.Data.CommandType.Text })
        ....
    }
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = '" + userName + "'",
        CommandType = System.Data.CommandType.Text })
        ....
    }
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))  
{
```

```
.....
```

```
SELECT * FROM Users WHERE UserName = 'Иван'
```

```
CommandType = System.Data.CommandType.Text } )
```

```
.....
```

```
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = ' + userName + ' ",
        CommandType = System.Data.CommandType.Text })
        ....
    }
```


Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = '" + userName + "'",
        CommandType = System.Data.CommandType.Text })
    ....
}
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = ' " + userName + "' ",
        CommandType = System.Data.CommandType.Text })
        ....
    }
```

Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection (...))  
{  
    ....  
    String userName = Request.QueryString["name"];  
    using (var command = new SqlCommand("SELECT * FROM Users WHERE UserName = '" + userName + "'", connection))  
    {  
        CommandText =  
            "SELECT * FROM Users WHERE UserName = '" + userName + "'";  
        CommandType = System.Data.CommandType.Text;  
    }  
    ....  
}
```

Иван



Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName =
using (var command = new SqlCommand(
    Connection = connection,
    CommandText =
        "SELECT * FROM Users WHERE UserName = '" + userName + "'",
    CommandType = System.Data.CommandType.Text ))
    ....
}
```

' OR '1'='1'



Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))  
{
```

```
.....
```

```
SELECT * FROM Users WHERE UserName = 'Иван'
```

```
CommandType = System.Data.CommandType.Text } )
```

```
.....
```

```
}
```


Пример уязвимости SQL injection

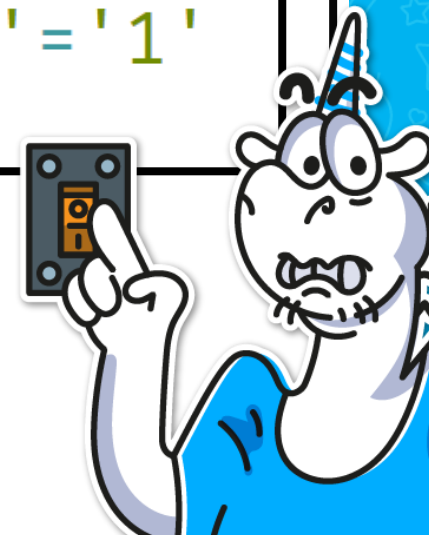
```
using (SqlConnection connection = new SqlConnection(...))  
{
```

```
    ....  
    String userName = Request.Form["userName"] ;
```

```
    SELECT * FROM Users WHERE UserName = '' OR '1'='1'
```

```
    CommandType = System.Data.CommandType.Text } )
```

```
    ....  
}
```



Пример уязвимости SQL injection

```
using (SqlConnection connection = new SqlConnection(...))
{
    ....
    String userName = Request.Form["userName"];
    using (var command = new SqlCommand() {
        Connection = connection,
        CommandText =
            "SELECT * FROM Users WHERE UserName = '" + userName + "'",
        CommandType = System.Data.CommandType.Text })
```

Предупреждение PVS-Studio: V5608 Possible SQL injection.

Potentially tainted data in the 'userName' variable is used to create SQL command.

ГОСТ Р 71207



ГОСТ Р 71207–2024

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Статический анализ программного обеспечения

Общие требования

Введён в действие 01.04.2024

Введён впервые

ГОСТ Р 71207–2024

Защита информации

РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Статический анализ программного обеспечения

Общие требования

Введён в действие 01.04.2024

Введён впервые

To Be Continued

ГОСТ Р 56939



ГОСТ Р 56939 - 2016

Защита информации

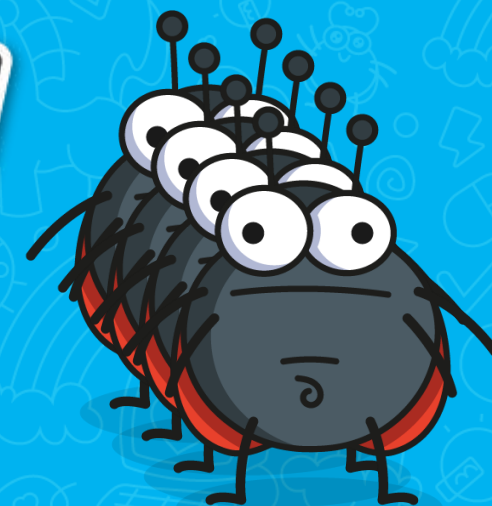
РАЗРАБОТКА БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Общие требования

Введён в действие в 2016 г.

ГОСТ Р 56939 - 2016

- Направлен на предотвращение и устранение уязвимостей программ
- Меры по разработке безопасного ПО:
 - ...
 - Динамический анализ кода
 - **Статический анализ кода**
 - ...



В соответствии с требованиями ГОСТ Р 56939...

- **Разработчик ПО должен проводить регулярный поиск уязвимостей** на этапе эксплуатации жизненного цикла ПО.
- В состав применяемых инструментальных средств разработчик ПО **должен включать статический анализатор.**
- Рекомендуется настраивать конфигурацию статического анализатора и **применять** специализированные **методы статического анализа для** более глубокого **поиска ошибок.**



ГОСТ Р 71407–2024

ГОСТ Р 71207–2024

Описывает:

- Термины;

ГОСТ Р 71207–2024

Описывает:

- Термины;
- Порядок внедрения и выполнения статического анализа;

ГОСТ Р 71207–2024

Описывает:

- Термины;
- Порядок внедрения и выполнения статического анализа;
- Классификацию ошибок, обнаруживаемых статическими анализаторами;

ГОСТ Р 71207–2024

Описывает следующие требования:

ГОСТ Р 71207–2024

Описывает следующие требования:

- К выполнению анализа;

ГОСТ Р 71207–2024

Описывает следующие требования:

- К выполнению анализа;
- К методам анализа;

ГОСТ Р 71207–2024

Описывает следующие требования:

- К выполнению анализа;
- К методам анализа;
- К инструментам анализа;

ГОСТ Р 71207–2024

Описывает следующие требования:

- К выполнению анализа;
- К методам анализа;
- К инструментам анализа;
- К специалистам, участвующим в выполнении анализа;

ГОСТ Р 71207–2024

Описывает следующие требования:

- К выполнению анализа;
- К методам анализа;
- К инструментам анализа;
- К специалистам, участвующим в выполнении анализа;
- К методике проверки статических анализаторов на соответствие стандартам.

Актуальность для команд



Если занимаетесь разработкой безопасного ПО (ГОСТ 56939)

- Правильные ли процессы?
- Подходят ли инструменты анализа кода?
- На что обращать наибольшее внимание?
(про критические ошибки)



Если хотите улучшить процесс разработки

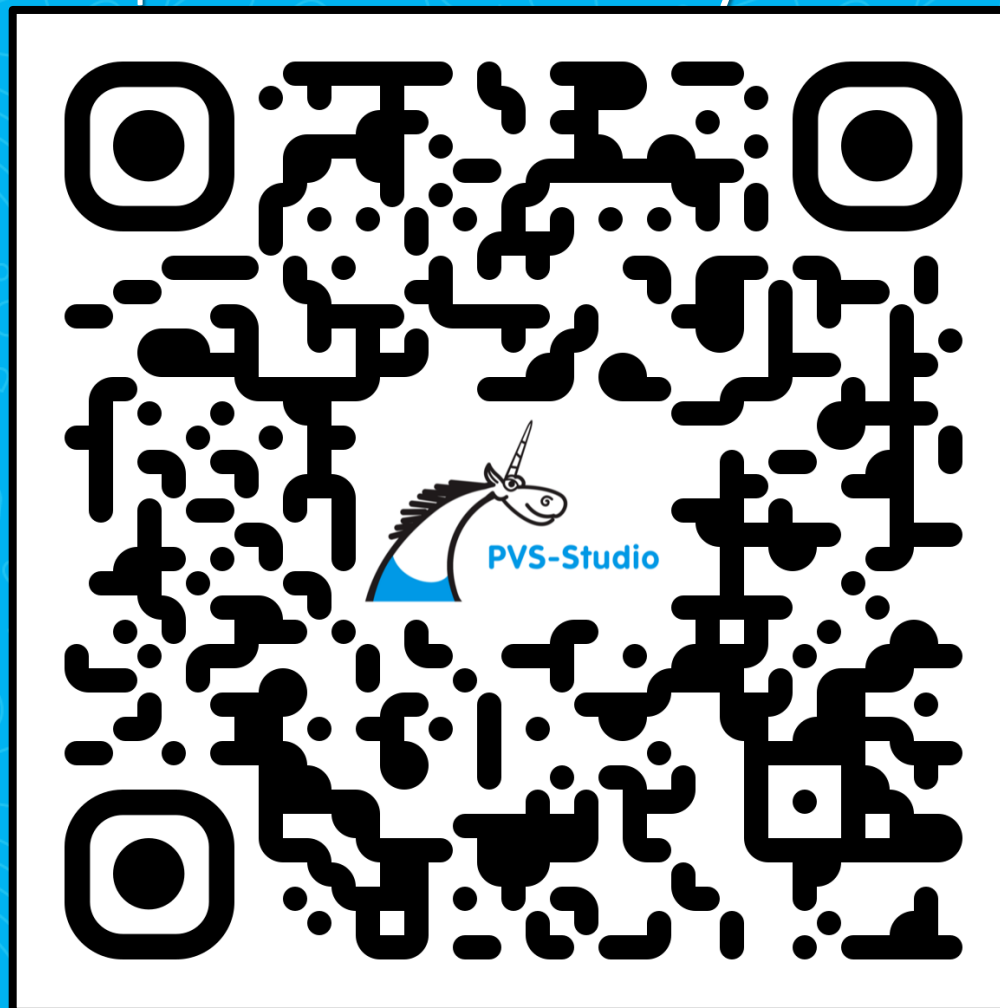
- Стоит ли менять инструменты / процессы
- Какие практики использовать?



Полная версия.

ГОСТ Р 71207–2024. Статический анализ программного обеспечения.

Общее описание и актуальность



Несколько примеров



Анализ потока данных

- Определяет предположительное значение переменных и констант.
- Примеры артефактов:
 - диапазон значений
 - точное значение
 - множество значений



Анализ потока данных (RavenDB)

```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        . . . .
    }
}
```

Анализ потока данных (RavenDB)

```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        if ( expr.Arguments.Count != 1)
        {
            throw new InvalidOperationException("....");
        }
        ....
    }
}
```

Анализ потока данных (RavenDB)

```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        if ( expr.Arguments.Count != 1)
        {
            throw new InvalidOperationException("...");
        }
    }
}
```

Предупреждение PVS-Studio:

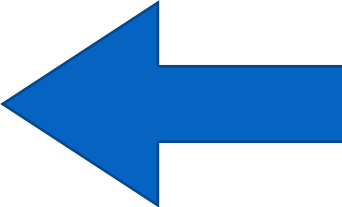
V3022 Expression 'expr.Arguments.Count != 1' is always true.

Межмодульный анализ (Barotrauma)

```
// Функция Remove() делает ссылку Sprite нулевой
partial class DecorativeSprite : ISerializableEntity
{
    public Sprite Sprite { get; private set; }
    ....
    public void Remove()
    {
        Sprite?.Remove();
        Sprite = null;
        ....
    }
}
```

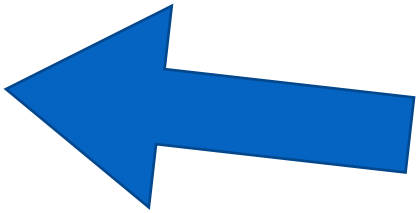
Межмодульный анализ (Barotrauma)

```
// Функция Remove() делает ссылку Sprite нулевой
partial class DecorativeSprite : ISerializableEntity
{
    public Sprite Sprite { get; private set; }
    ....
    public void Remove()
    {
        Sprite?.Remove ();
        Sprite = null;
        ....
    }
}
```



Межмодульный анализ (Barotrauma)

```
// Функция Remove() делает ссылку Sprite нулевой
partial class DecorativeSprite : ISerializableEntity
{
    public Sprite Sprite { get; private set; }
    ....
    public void Remove()
    {
        Sprite?.Remove();
        Sprite = null;
        ....
    }
}
```

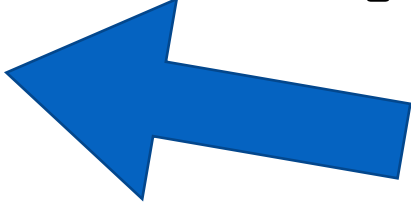


Межмодульный анализ (Barotrauma)

```
public void RecreateSprites ()
{
    for (int i = 0; i < DecorativeSprites.Count; i++)
    {
        var decorativeSprite = DecorativeSprites[i];
        decorativeSprite.Remove ();
        var source =
            decorativeSprite.Sprite.SourceElement;
        . . . .
    }
}
```

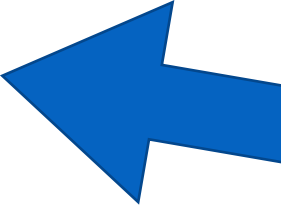
Межмодульный анализ (Barotrauma)

```
public void RecreateSprites ()
{
    for (int i = 0; i < DecorativeSprites.Count; i++)
    {
        var decorativeSprite = DecorativeSprites[i];
        decorativeSprite.Remove();
        var source =
            decorativeSprite.Sprite.SourceElement;
        ....
    }
}
```



Межмодульный анализ (Barotrauma)

```
public void RecreateSprites ()
{
    for (int i = 0; i < DecorativeSprites.Count; i++)
    {
        var decorativeSprite = DecorativeSprites[i];
        decorativeSprite.Remove ();
        var source =
            decorativeSprite.Sprite.SourceElement;
        ....
    }
}
```



Межмодульный анализ (Barotrauma)

```
public void RecreateSprites ()
{
    for (int i = 0; i < DecorativeSprites.Count; i++)
    {
        var decorativeSprite = DecorativeSprites[i];
        decorativeSprite.Remove ();
        var source =
            decorativeSprite.Sprite.SourceElement;
```

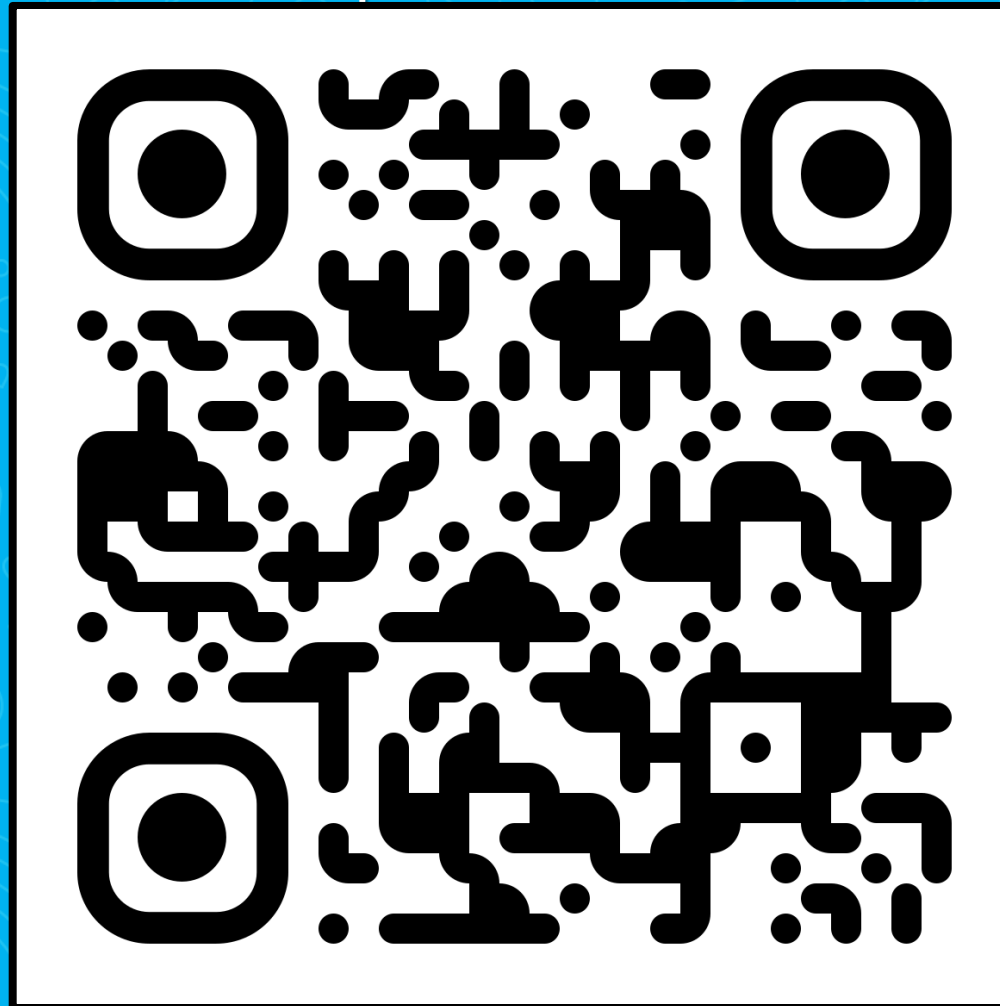
Предупреждение PVS-Studio:

V3080. Possible null dereference. Consider inspecting 'decorativeSprite.Sprite'.

Полная версия.

ГОСТ Р 71207–2024 - Статический анализ программного обеспечения.

Терминология



Критическая ошибка (!)

- Самое важное понятие
- Ошибка, которая может привести к нарушению безопасности.
- Мы называли такие ошибки потенциальными уязвимостями.



Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```

Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
    CredentialsUserName = userName;
    CredentialsUserPassword = password;
    CredentialsDomain = domain;
}
```


Ошибки непроверенного использования чувствительных данных (ONLYOFFICE Community Server)

```
public void SetCredentials
(string userName, string password, string domain) {
    if (string.IsNullOrEmpty(userName)) {
        throw new ArgumentException
            ("Empty user name.", "userName");
    }
    if (string.IsNullOrEmpty("password")) {
        throw new ArgumentException
            ("Empty password.", "password");
    }
}
```

Предупреждение PVS-Studio:

V3022 Expression 'string.IsNullOrEmpty("password")' is always false.

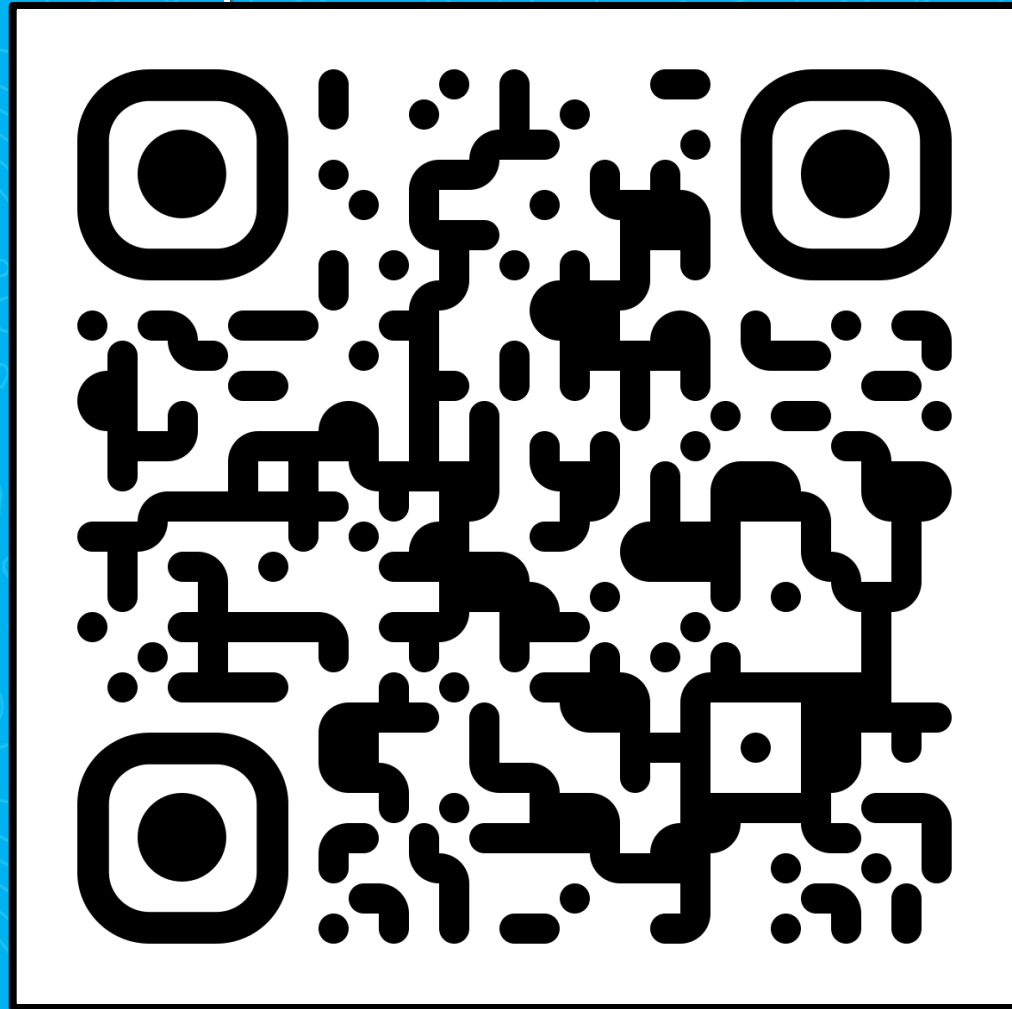
```
CredentialsDomain = domain;
```

```
}
```

Полная версия.

ГОСТ Р 71207–2024 — Статический анализ программного обеспечения.

Критические ошибки



Q&A



Задавайте
вопросы



Глеб Асламов