Talos Vulnerability Report

TALOS-2021-1303

# CODESYS Development System ObjectManager.plugin ProfileInformation.ProfileData Unsafe Deserialization vulnerability

JULY 26, 2021

CVE NUMBER

CVE-2021-21866

## Summary

An unsafe deserialization vulnerability exists in the ObjectManager.plugin ProfileInformation.ProfileData functionality of CODESYS GmbH CODESYS Development System 3.5.16 and 3.5.17. A specially crafted file can lead to arbitrary command execution. An attacker can provide a malicious file to trigger this vulnerability.

## Tested Versions

CODESYS GmbH CODESYS Development System 3.5.16
CODESYS GmbH CODESYS Development System 3.5.17

## Product URLs

https://store.codesys.com/codesys.html

## CVSSv3 Score

8.8 - CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

## CWE

CWE-502 - Deserialization of Untrusted Data

## Details

The CODESYS Development System is the IEC 61131-3 programming tool for industrial control and automation technology, available in 32- and a 64-bit versions.

This vulnerability may be exploited by including a malicious profile.auxiliary file within a project file or project template, among other methods.

Deserialization occurs within the ProfileData Property in the ProfileInformation class

```
DefaultSerialization("Profile")]
StorageVersion("3.4.0.0")]

private byte[] ProfileData
{
    get
    {
        byte[] result;
        using (ChunkedMemoryStream chunkedMemoryStream = new ChunkedMemoryStream())
        {
            new BinaryFormatter
            {
                Binder = new LegacyCODESYSSerializationBinder()
            }.Serialize(chunkedMemoryStream, this._profile);
            result = chunkedMemoryStream.ToArray();
        }
        return result;
    }
    set
    {
        using (ChunkedMemoryStream chunkedMemoryStream = new ChunkedMemoryStream(value))
        {
            BinaryFormatter binaryFormatter = new BinaryFormatter();
            this._profile = (Profile)binaryFormatter.Deserialize(chunkedMemoryStream); // [1]
        }
    }
```

The `BinaryFormatter.Deserialize` method is never safe when used with untrusted input [2]. The deserialization that occurs at [1] is vulnerable to exploitation via the profile.auxiliary file within a project file or template.

[2] https://docs.microsoft.com/en-us/dotnet/standard/serialization/binaryformatter-security-guide

## Crash Information

Partial Call Stack

```
objectmanager.plugin.dll!_3S.CoDeSys.ObjectManager.ProfileInformation.ProfileData.set(byte[] value = {byte[0x00004C3F]})
objectmanager.plugin.dll!_3S.CoDeSys.ObjectManager.GenericObjectService.SetSerializableValueImpl(_3S.CoDeSys.Core.Objects.GenericObject go =
{_3S.CoDeSys.ObjectManager.ProfileInformation}, _3S.CoDeSys.ObjectManager.TypeAccess typeAccess = {_3S.CoDeSys.ObjectManager.TypeAccess},
string valueName = "Profile", object value = {byte[0x00004C3F]})
objectmanager.plugin.dll!_3S.CoDeSys.ObjectManager.GenericObjectService.SetSerializableValue(_3S.CoDeSys.Core.Objects.GenericObject go =
{_3S.CoDeSys.ObjectManager.ProfileInformation}, string valueName = "Profile", object value = {byte[0x00004C3F]})
Objects.dll!_3S.CoDeSys.Core.Objects.GenericObject.SetSerializableValue(string stValueName = "Profile", object value = {byte[0x00004C3F]})
binaryarchive.plugin.dll!ns3.Class4.method_9(System.IO.BinaryReader binaryReader_0 = {System.IO.BinaryReader}, ns6.Class6 class6_0 =
{ns6.Class6}, bool bool_0 = true)
binaryarchive.plugin.dll!ns3.Class4.method_1(System.IO.BinaryReader binaryReader_0 = {System.IO.BinaryReader}, bool bool_0 = true, out
string string_0 = null)
binaryarchive.plugin.dll!ns3.Class4.imethod_0(System.IO.BinaryReader binaryReader_0 = {System.IO.BinaryReader},
_3S.CoDeSys.Core.Objects.IArchivable iarchivable_0 = null, byte[] byte_0 = null, bool bool_0 = true, out string string_0 = null)
binaryarchive.plugin.dll!_3S.CoDeSys.BinaryArchive.BinaryArchiveReader.Load()
objectmanager.plugin.dll!_3S.CoDeSys.ObjectManager.Project.Load(System.IO.Stream stream = {System.IO.MemoryStream}, string stStreamName =
"missing_test")
objectmanager.plugin.dll!_3S.CoDeSys.ObjectManager.ObjectManager.InspectAndLoadProject(System.IO.Stream stream = {System.IO.MemoryStream},
string stStreamName = "missing_test", string stWorkingFolder = @"C:\ProgramData\CODESYS\Temporary Files\ab6ea14a-24f0-44b1-a2ab-
076041a94334", string stProjectPath = @"C:\Users\User\Documents\codesys\projects_test\missing_test.project",
_3S.CoDeSys.Core.Objects.IProjectInspectionReporter reporter = {ns2.Class57}, out int nProjectHandle = 0xFFFFFFFF)
engine.plugin.dll!ns1.Class60._3S.CoDeSys.Core.IProjects4.OpenProject(string stPath =
@"C:\Users\User\Documents\codesys\projects_test\missing_test.project", bool immediatelyUpgradeStorageFormat = false, params System.Guid[]
projectAttrs = {System.Guid[0x00000002]})
engine.plugin.dll!ns1.Class60._3S.CoDeSys.Core.IProjects.OpenProject(string stPath =
@"C:\Users\User\Documents\codesys\projects_test\missing_test.project", params System.Guid[] projectAttrs = {System.Guid[0x00000002]})
filecommands.plugin.dll!_3S.CoDeSys.FileCommands.FileCommandHelper.OpenProject(string stPath =
@"C:\Users\User\Documents\codesys\projects_test\missing_test.project", bool readOnly = false, System.Guid converterOrFilterGuid =
{System.Guid})
...
```

## Timeline

2021-05-18 - Vendor Disclosure

2021-07-26 - Public Release

## CREDIT

Discovered by Patrick DeSantis of Cisco Talos.