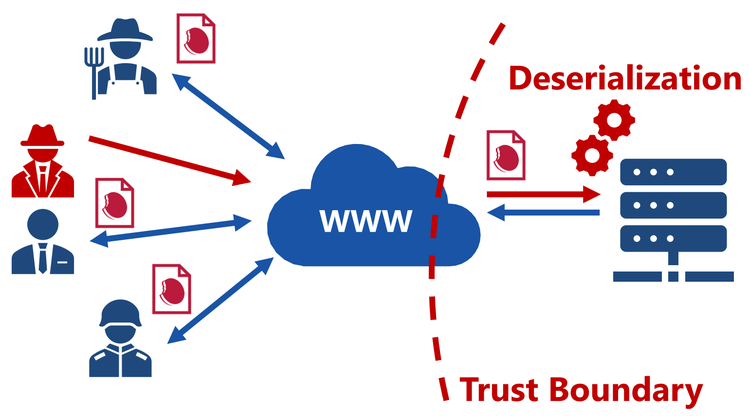
KTH ROYAL INSTITUTE OF TECHNOLOGY

Automated Detection of Object Injection Vulnerabilities

MIKHAIL SHCHERBAKOV AND MUSARD BALLIU

Introduction

Object Injection vulnerability (OIV) is an application level vulnerability that occurs when an application instantiates an object of arbitrary type based on untrusted user-supplied data, and invokes some methods of the object.



OIV in Insecure Deserialization of Untrusted Data

The impact of OIV can lead to exploitation of Remote Code Execution (RCE), Denial of Service (DoS) attacks depending on the type of gadget chain.

Gadget Chain is a graph of objects that are available on the target system and trigger malicious actions by the attack.

Research Goals

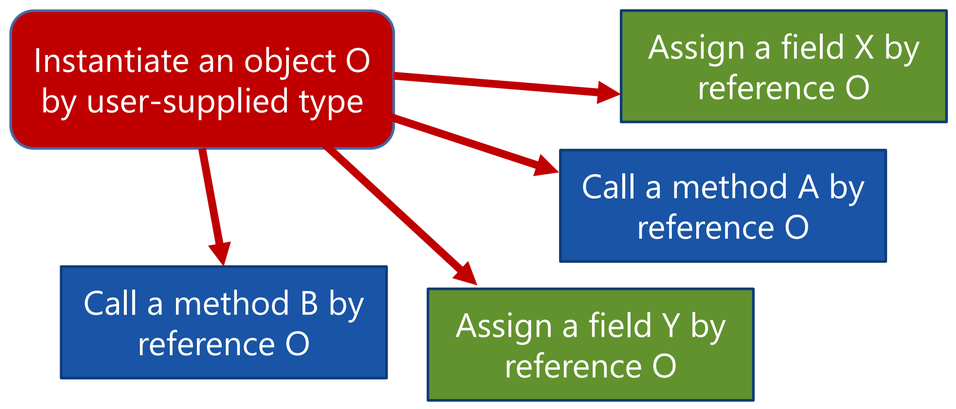
We study the code of serialization libraries in the context of .NET platform and identify formal patterns leading to this kind of attack.

The .NET Framework has a large code base that makes manual analysis impossible. The main goals are:

- Automated identification OIV patterns.
- Detection of gadget chains and payload generation by given patterns.
- Automated detection vulnerabilities in real-world .NET applications.

Methodology

We choose Common Intermediate Language (CIL) for analysis in order to be able to detect OIV patterns and new vulnerabilities in .NET Framework with no availability of source code.



Pattern of Object Injection Vulnerability (OIV)

Control Flow Analysis

- Build an index of all method calls in CIL assemblies.
- Compute paths from methods which instantiate an object of given type to API entry points.

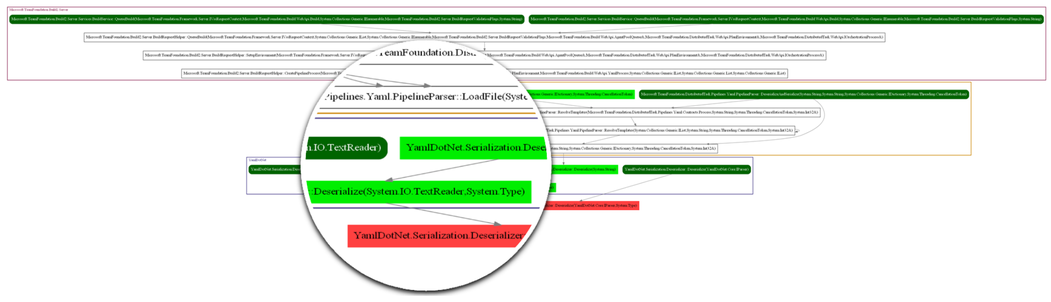
Data Flow Analysis

- Build method summaries that represent *symbolic* values in the code.
- Compositional aliases analyses via summaries.
- Taint analyses for instantiated objects.

Evaluation

Uncovered new vulnerabilities in Microsoft Azure DevOps Server:

- CVE-2019-0866 RCE via opening a malicious PDF
- CVE-2019-0872 RCE via a stored Cross Site Scripting
- CVE-2019-1306 RCE via uploading a malicious Markdown document



Call graph of CVE-2019-0866 by DeReviewer

We design and implement a automated toolchain:

- OIReviewer – static analyzer that detects new patterns of OIV in .NET Framework and third-party libraries.
- DeReviewer – static analyzer that detects usage of OIV patterns as described in build-in DSL and generates payloads for exploitation.

Mikhail Shcherbakov Lindstedtsvägen 5 mshc@kth.se
Musard Balliu 114 28 Stockholm musard@kth.se