TPT Severe Issues

Introduction

The following document contains a list of known severe issues of TPT. By severe issues we mean issues/bugs in particular versions of TPT that:

1. might cause malfunctions in the behavior of TPT
2. are hard or even impossible to find by the TPT user herself/himself
3. cause the risk that bugs/defects in a SUT (system under test) are not detected by TPT in cases where TPT would have been able to reveal these bugs/defects in the SUT without the aforementioned malfunction in the behavior of TPT.

Usually these severe issues address the situations where the problem might appear and have well-defined workarounds.

ISSUE # 31924

TITLE:
RMI API is vulnerable to "CWE - 502 : Deserialization of Untrusted Data" due to used libraries.
Execution of arbitrary code is possible.

ISSUE DETECTION:
02-November-2021

AFFECTED VERSIONS OF TPT:
TPT 7 to TPT 17u1

PRECONDITIONS:
- RMI/Remote API is activated in preferences and library commons-collections-3.2.1.jar is present
  in your TPT installation directory in "lib\commons-collections-3.2.1.jar".
- firewall allows access of RMI API from other computers

DETAILS:
If RMI API is enabled and exposed by your firewall an unauthenticated attacker could execute arbitrary
code on the remote machine. Java cannot verify that serialized objects are well formed. In some cases
an attacker can use modified data to execute code during deserialization. There are some known "gadgets"
that can be used by an attacker. One gadget is the libray commons-collections version 3.2.1. The attack
can be prevented by restricting deserialization by using deserialization filters or upgrading the libraries
to unvulnerable versions.

EFFECT OF THE ISSUE:
An unauthenticated attacker could execute arbitrary code on the remote machine.

WORKAROUND:
Disable RMI API or replace "lib\commons-collections-3.2.1.jar" by commons collection version 3.2.2.

RESOLVED IN:
TPT 15u5 TPT 16u4, TPT 17u2

ISSUE # 30102
=============
TITLE:
When comparing INT64 signals using Min/Max or Signal Comparison assesslet with values larger than 2^53 or smaller than -2^53, the computation can incorrectly compare the signal with the reference(s) which might lead to PASSED results even if the specified bounds are exceeded.

ISSUE DETECTION:
11-January-2021

AFFECTED VERSIONS OF TPT:
TPT 8 - TPT 16

PRECONDITIONS:
The Min/Max or Signal Comparison assesslet is used with INT64 signals with values larger than 2^53 or smaller than -2^53.

DETAILS:
If signal or reference values are larger than 2^53 or smaller than -2^53, the difference can be missed because the values are converted and compared as double values. (Since values larger/smaller than 2^53/-2^53) cannot be converted to double without losing precision, floating point precision problems might occur in such cases.)

EFFECT OF THE ISSUE:
Values outside of the specified bounds might be overlooked by TPT leading to PASSED results, but should be FAILED.

WORKAROUND:
Avoid using INT64 signals in Min/Max or Signal Comparison assesslets if the values are larger than 2^53 or smaller than -2^53.

RESOLVED IN:
TPT 15u4, TPT 16u1

ISSUE # 30063
=============
TITLE:
When iterating in assessment scripts via an inlined loop and applying signal processing functions like TPT.average(), TPT.min(), ... with changing argument in each loop, the result of the first iteration is incorrectly used for the following iterations.

ISSUE DETECTION:
16-December-2020

AFFECTED VERSIONS OF TPT:
TPT 8 - TPT 16

PRECONDITIONS:
The usage of the signal processing function must be applied on an expression or a signal that is being changed while iterating through an inlined loop or list comprehension.

DETAILS:
For faster computation of timed expressions of the form
  foo(t) := TPT.average(...)
results of signal processing functions are being cached.
The cached result is invalidated as soon as a new line is reached.
When iterating through an inlined loop, the same expression is evaluated multiple times.
If during this inlined iteration a variable of the expression is changed, the cached result of the signal processing function is used instead of a newly calculated.

Affected are inlined expressions of the form
  for x in range(n): print TPT.min(...+x)
  while x < n: x=x+1;print TPT.min(...+x)
as well as list comprehension:
  my_list = [TPT.min(...+x) for x in range(n)].

EFFECT OF THE ISSUE:
Old cached values are used instead of recalculating the value in every iteration,
which results in wrong computation results.

WORKAROUND:
Avoid using inlined loops or list comperhensions and use loops with indentation in multiple code lines instead.

RESOLVED IN:
TPT 15u4 TPT 16u1

ISSUE # 24165
==========
TITLE:
TPT-VMAPI corrupts values at runtime for array input or read/write output channels that have a specified dimension in TPT less than the specified dimension in the SUT(C code).

ISSUE DETECTION:
11-Jun-18
AFFECTED VERSIONS OF TPT:
TPT 8 – TPT 12

PRECONDITIONS:
- Test execution by means of EXE-platform, CANoe-platform, ASCET-platform (excluding ASCET@FUSION), C-code-platform.
- Test model contains array channels (including struct channels

containing arrays or array channels containing structs) that are configured as input channels.
- The dimension of these array input channels in TPT is smaller than the dimension in the SUT.

DETAILS:
Assume that the test model uses the feature of TPT which allows the dimension of array channels in TPT (logical dim) to be smaller than the dimension in the SUT (physical dim). If one or many of such the array channels are read from the SUT to TPT per test cycles (periodically) during test execution TPT potentially corrupts some *other* channels/parameter values at runtime. For each test model the number of affected channels/parameters is fixed, but unpredictable.

EFFECT OF THE ISSUE:
Channels and/or parameters can be corrupted at runtime under the specified preconditions. The signals recorded for these channels/parameters in the TPTBIN file are corrupted afterwards and the subsequent assessment will analyze the results based on these potentially corrupted data.

WORKAROUND:
Regenerate the testframe.c testdriver to ensure that the dimension of signals in TPT matches the dimension of signals in the SUT.

RESOLVED IN:
TPT11u3, TPT12u2