ERICSSON

TTCN-3 and Titan security considerations

Technical Report

2020-09-18

The TTCN-3 code executes by design in an abstract space, with no references and access to system resources, such as memory, file system, network. Any such access can be performed via C++ or Java extensions (external functions, test ports, logger plug-ins) only. That being said, it's virtually impossible for a pure TTCN-3 code to abuse of system resources and compromise the security of the environment it is executing in.
Hence no security coding guidelines are relevant for the TTCN-3 language as such.
Obviously the Java or C++ extensions attached will have to follow their language-specific security coding guidelines.

Another possible security angle is to examine the role a Titan application is playing in a network. TTCN-3 and Titan are meant to be used and are being used to write test applications, with the following specifics:

-these applications are not deployed in a production network

-TTCN-3/Titan based application are not server based

-the execution of these applications is discontinuous and limited in time; when executing test cases, only the main controller stays resident, all other components are created when test cases start and destroyed when test cases end.

In summary, TTCN-3 test applications themselves offer an imperceptible attack surface; even if a vulnerability were exposed and exploited, the associated risk -of compromising a test application- is minimal, no production service would be disrupted or subverted.

Obviously, as for any application, vulnerabilities may result due to misconfiguration or misuse of Titan; to prevent this, general security guidelines should apply; however misuse or misconfiguration cannot be tied back to any coding practice; no coding guideline will prevent malicious usage of Titan in e.g. overloading a network to the point of denial of service.