Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Security Evaluation of Library-based Access Control Mechanism

Description of Master Thesis

Dr.-Ing. Mohammad Hamad, May 25, 2022

Title: "Security Evaluation of Library-based Access Control Mechanism"
Supervisor: Dr.-Ing. Mohammad Hamad
Period: 24 weeks
Start Date: ASAP

Context

Complex systems, by necessity, use layers upon layers of existing code usually in the form of libraries. This results in numerous interactions between the main application and externally provided code, via standardized APIs. By monitoring these interactions, we can create an execution model of the application. In other words, we can create a state machine that contains the sequence of library function invocations, supplied arguments, return values etc. This model will then reflect the behavior of the application.
The premise is that by comparing actual execution behavior with that predicted by our execution model, we can detect possible attacks aiming at taking over our application, or other off-nominal situations that are the result of bugs triggered by anomalous input or unforeseen events.
In [1] we see that by adding wrappers to existing functions we can add code that records information such as the invocation of a particular library function, the supplied arguments etc., This information can be used in conjunction with a fairly comprehensive test suite to create the execution state model.
However, as wrappers are inherently only suitable for cooperative monitoring (since the wrapped destination function is in the same address space as the invoking code, this code can simply jump directly to the desired function bypassing the wrapper), a more robust - kernel based - mechanism should be used for enforcing the execution model. The mechanism to be used is based on the concept of gates that ensures the transferring execution from one library to another involves passing through a gate. Direct jumps across libraries are controlled by the kernel and will result in a fault. Once a fault is detected the kernel is invoked automatically to process the fault, which in turn calls the appropriate gate code. This mechanism ensures that (a) gates cannot be bypassed, and (b) that the gate code is always invoked by the kernel and execution always begins from the beginning of the code (i.e. there is no way to jump in the middle of the code).
The objective of this research if to consider the a fairly small but not trivial application and use wrappers to construct its execution model. Then a vulnerability in this application (either pre-existing, or intentionally introduced) will be exploited in order to perform a zero-day attack. The result of the attack will be on the unprotected application will then be compared with the behavior of the application with the gate mechanism enforcing compliance with the previously generated execution model. Multiple variations of the attack may be performed in order to explore the limitations of the protection mechanism.

References

[1] M. Tsantekidis and V. Prevelakis. Efficient monitoring of library call invocation. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 387–392. IEEE, 2019.