Submission to Smart Event'11 Conferences

TITLE OF THE PRESENTATION:
Automatic generation of vulnerability test suite for the Java Card verifier

PRESENTING AUTHOR DETAILS:
Last Name: Aymerick
First name: Savary
Email: aymerick.savary@usherbrooke.ca
Job title:
Organisation: Université de Sherbrooke
Country: Canada
Postal address: GRIL, Département Informatique, Université de Sherbrooke, Québec, Canada

Phone number(s): 819-432-2062
Co-author(s) if applicable: Marc frappier and Jean-Louis Lanet

CONFERENCE YOU SUBMIT TO: **e-Smart**

ABSTRACT (300-500 words):
The security of Java based smart card relies essentially on the impossibility to forge pointers and to perform arithmetic operation on them. Recently papers have shown that ill formed application can retrieve important information like crypto keys by luring the operating system. The correct implementation of that component is crucial for these cards. The new standard imposes that such a component must be embedded in the card. Designer must ensure that this component is correctly implemented. Testing such a component is a difficult task due to the number of required tests.

One of the approaches is to use a formal model that allows automating the test generation. The creation of a test file containing a single fault permits to identify the precise location of the mistake in the implementation, which allows us to know where the program needs to be fixed. This work focuses on a precise aspect of bytecode verification, the type checking. This generation takes place in three stages, namely: i) the preamble, ii) the fault iii) the postamble. The preamble and the postamble are obtained through researching a sequence of instructions leading to the fault and consequently to the end of a valid program; a model checker was used to find this sequence. We have chosen to use a model checker to find this sequence. The fault is a statement that is put in a context where its precondition is false. If this fault is not detected by the Java byte code verifier, we thus know there is a mistake in the implementation and potentially an exploitable weakness. In order to create all the faults, all the preconditions rejectable by the Java byte code verifier are generated for each instruction.

The automatic generation of faults is the first step in the process of forming a test file. The faults are then used to generate the preamble and then once the context is changed by the fault we can search the postamble. Some of these stages are unfeasible, and will not be kept for testing. The set of test files obtained is usable for all implementations of the byte code verifier. In order to verify our approach, we have experimented with a small but representative subset of the byte code. We have modeled it in the language Event-B. The preamble and the postamble have been investigated using ProB and the faults have been created with a Java program. This is an efficient method to obtain a set of vulnerability tests.

KEYWORDS best describing the scope of your presentation (3 or 4)
security, vulnerability test, formal method, Java Card byte code verification

Bullet points: **The 3 or 4 key points of your presentation**
1/ Verifying the Java byte code verifier
2/ Methodology to generate test cases
3/ Use of a formal model to generate faulty CAP file
4/

Short bio (2 or 3 lines) of the author(s)

Currently student in a master degree in computer science at the University of Sherbrooke and University of Limoges. My research is done both in the SSD labs in Limoges to acquire knowledge related to smart cards security and in the GRIL labs for the formal methods.

Thank you to send a PHOTO of the speaker ATTACHED to your email.

Submission to be sent to: lperron@strategiestm.com