Implementing Secure & Reliable Software

Seminar - SS 2013

Topic Descriptions

OVERVIEW

Analysis and Verification of Runtime Behavior of Automotive SW ......................... 2
Languages for Specifying Safety-Critical Software Requirements ......................... 2
Requirements for Software in Safety-Critical Systems: ISO 26262 ........................ 2
Model-based Security Testing .............................................................................. 3
Underlay Awareness in P2P Systems: Techniques and Challenges ...................... 3
Vulnerability Assessment Approaches ................................................................. 3
Security Challenges on Virtual Machine Monitors ............................................. 4
Parallel Symbolic Execution ................................................................................ 4
SAT Solvers ......................................................................................................... 4
Exploiting Locality in Peer-to-Peer Overlay Networks ...................................... 5
On the Feasibility of Predicting Software-level Vulnerabilities ....................... 5
Security of Web Browsers ................................................................................... 6
Security of Windows COM objects ..................................................................... 6
Reliability in Wireless Sensor Networks ............................................................. 6

Contact
Thorsten Piper
Tel. 06151-16-3414
Room: S2|02 E219
piper@cs.tu-darmstadt.de
Analysis and Verification of Runtime Behavior of Automotive SW

AUTOSAR is an evolving standard for automotive software architecture, featuring a component-based design throughout its different abstraction layers (OS, Runtime Environment and Application). At this point, AUTOSAR components do not have any standardized timing relation with their tasks. This causes unpredictable and non-deterministic runtime behavior, which can only be analyzed and verified after integration phase. Within this seminar work, current approaches to runtime analysis (static or experimental) of embedded automotive systems shall be surveyed, especially focusing on worst-case execution time (WCET) analysis. An introduction to common scheduling algorithms employed by the operating systems of these systems will also be part of the work.

Languages for Specifying Safety-Critical Software Requirements

The verification of a software implementation (e.g. by testing or formal methods) requires a specification that describes how the implementation should behave. Especially for formal verification attempts, specification languages with precise semantics are required. Several such specification languages exist and some are frequently advocated for use in safety-critical software engineering, as they ease verification. “Z” is one such language. The task of this seminar report is to provide an overview on formal specification languages suitable for the development of safety-critical software, their core features, and application fields.

Requirements for Software in Safety-Critical Systems: ISO 26262

ISO 26262 is a draft standard for the functional safety of road vehicles, based on the ISO IEC 61508 standard. The standard is currently under approval, but expected to be approved and published in the near future and to have a great impact on the development and quality assurance of electronic components and software in modern automobiles. The goal of this seminar report is to provide an overview on ISO 26262 with a special focus on software requirements.
Model-based Security Testing

Software security is a software quality issue that continues to grow in importance as software systems manage continually increasing amounts of critical corporate and personal information. Testing helps to verify whether the behavior of a product or system conforms to the security features claimed by the manufacturer. Presently, developing and executing security functional tests is time-consuming and costly. Consequently, methods and tools for automating security functional testing are developed, relying on a model-based approach to automate security functional testing. Within this seminar work, the range of model-based security testing approaches shall be surveyed and compared, also emphasizing their benefits and drawbacks against non model-based testing approaches.

Underlay Awareness in P2P Systems: Techniques and Challenges

The impact of Peer-to-Peer (P2P) paradigm increases both in research and in industry. Traditional P2P systems however, suffer from inefficiency due to lack of information from the underlay, i.e. the physical network. Although there is a plethora of research on underlay awareness, this aspect of P2P systems is still not clearly structured. The student should provide a brief taxonomic survey that outlines the different steps for achieving underlay awareness.


Vulnerability Assessment Approaches

The existing vulnerability databases such as the National Vulnerability Database (NVD) or the Open Source Vulnerability Database (OSVDB) try to assess the criticality of discovered vulnerabilities. A less explored research question is about which is the right source for vulnerability studies to do such an assessment for new vulnerabilities. Within this seminar, existing vulnerability assessment approaches and their data sources should be investigated and compared with respect to the quality of the results they provide.

Security Challenges on Virtual Machine Monitors

Virtualization is one of the basic technologies behind Cloud computing. Namely the claimed secure isolation that various virtualization solutions provide enables Cloud’s multitenancy model. However, a variety of recent studies have demonstrated attacks on the Cloud resulting from insufficient isolation between Cloud’s tenants. This fact raises the question of the need of additional requirements for a more secure logical isolation between virtual machines sharing the same physical resources. The aim of the report is to consider the security challenges related to the virtualization focusing on the logical isolation it provides and to discuss how these requirements are met by the current virtualization models.


Parallel Symbolic Execution

In the realm of implementation of dependable and reliable software, formal verification constitutes a major phase in the development process. Although model checking has proven to be efficient in finding bugs and flaws in real systems it remains of limited use due to the fact that it requires concrete inputs from the user. This means that in order to completely verify a software, one has to run the model checker for every possible input. Symbolic execution, on the other hand, is a technique that works using the same principle as model checking apart from the fact that it covers all possible input ranges in one run of the verification. Although not trivially realizable, the parallelization of Symbolic execution can considerably speed up the verification process. In this seminar report, the student is required read about the concept of symbolic execution, understand the challenges faced to implement it and discuss some existing work.


SAT Solvers

SAT solvers are extensively used in the verification of software (e.g. symbolic execution, theorem proving based verification). A recent work showed promising results advocating the possible use of optimization techniques to address the scalability problems SAT
solvers suffer from. In this seminar report, you are required to thoroughly read an advanced paper, summarize it and discuss its content.


**Exploiting Locality in Peer-to-Peer Overlay Networks**

Peer-to-Peer protocols represent the basis of numerous applications. As such systems may span across continents or the whole world, various approaches exist to improve location based searches and overlay maintenance, which consider metrics based on network latencies, geographic information, or operator domains. The student should survey recent approaches and address in particular their performance and reliability aspects.


**On the Feasibility of Predicting Software-level Vulnerabilities**

It is generally accepted that software is inherently insecure, either due to coding practices, use of (known) vulnerable legacy components, etc. Furthermore, software "zero-day vulnerabilities" (i.e. previously unknown) usually reflect on big economic costs for both users and software vendors, given in part the economic benefits that exploiters obtain from the underground market [1]. Although the software community has been dealing with vulnerabilities since its beginnings, unfortunately it is still far from actually producing “secure” software. One of the most promising approaches for developing secure software, relates to the techniques to actually “predict” vulnerabilities before deploying it. This seminar topic aims to motivate a critical discussion on the feasibility of existing approaches being proposed to predict software vulnerabilities e.g., Vulture [2] and most recently Yonghee [3].


Security of Web Browsers

A browser is one of the most important software tools that is installed on almost every personal computer. Most of sensitive personal data are entered into browsers to access different online services ranging from social networks to banking. We study the security of web browsers especially when an add-on or plugin is integrated into them or a script is being run by an active web page.


Security of Windows COM objects

Windows Component Object Model (COM) modules are the building blocks of many services in a Windows PC, including codecs, plugins, multimedia filters, etc. The vulnerabilities of COM objects are especially important because of the information flow from one application to third-party software. This seminar investigates the COM object vulnerabilities especially when a certain operation is handled to a default or higher level COM object.


Reliability in Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are prone to node and link failures and unexpected communication delays. The reliability of WSNs becomes especially important when they are to be employed in a critical infrastructure system. In this seminar, we study how to make the communication, routing and processing in WSNs resilient to faults and failures.