1 Invited talk: Developments in Educational Programming Environments

This talk was about how software development can be taught in different ages. The education for software development nowadays starts already in primary or secondary school. This requires some new learning methods compared to previous ones. Students in universities are at software development courses because they have an interest to learn software development. Students in primary and secondary school require some kind of motivation because they might not be interested in learning software development. Other teaching methods are block programming and frame programming. Block programming is a visual method where you move code blocks and arrange them to build your program. It is limited in the blocks that are provided. Scratch was described as an example block programming tool. This tool is popular for younger students like in primary school. Frame programming has a similar approach. You have frames that can be moved inside the code. Basically it replaces the usage of brackets in source code. This helps to prevent syntax errors. Inside the frames code can be written, which still can have syntax errors. This approach is a full programming language approach which can solve all problems which other textual programming languages can solve.

Some tools were also presented that will help to teach programming to younger students. BlueJ was the first tool, which shows a basic class diagram. You can implement functions for the classes and create objects of these classes. This tool helps to learn the difference between classes and objects. Another presented tool was Greenfot, which shows objects in graphical representation. In this representation a objects were represented as a craps. The craps provided functions to move on the screen. This encourage students to test and use the tool. With some lines of code this tool allows to create simple computer games.

2 Session A

This section will provide a short report about the session *Malware detection and coding theory.*

2.1 Memory access patterns for malware detection

The first presentation was about using machine learning to detect malware. This research did use memory access patterns for detection. The main reason for this method is because malware nowadays tries to hide the malicious code parts. It will detect, if it is run in debug mode or run in a virtual environment. In this case the malware will not run the malicious parts this will decrease the detectability and will make the analysis more difficult. As stated memory access patterns will be used for the detection. If just the hardware access patterns are observed, the malware will not be able to detect that it is observed.
They used different sizes of memory traces with different n-grams sizes and different feature selections. The best results did provide the kNN and the ANN machine learning algorithm. They reached a detection rate over 90%. 1000 malicious programs without GUI were used as learning data from the website *virusshare.com.*

2.2 Hey TPM, Sign My Transaction

This research is about preventing man in the middle attacks in the browser on online banking. The main point of this research is to present the bank the same output as the user sees on his screen. They will make a screenshot of the screen as the users sees for the transaction. This screenshot will be send to the bank and they can check, if the screenshot fits with the provided transaction. A problem is that malicious software in the browser or on the computer could also manipulate the screenshot which is send to the bank. Another problem might be that the drivers used for the graphic card might be corrupted and display different data. To prevent such kind of issues they used the trusted platform module (TPM). This is provided by modern computers and can be used to store secret keys, password, certificates, et cetera. It can also be used to maintain the integrity of the operating system and drivers. One disadvantage of this solution is that driver updates always require new certificates to preserve the integrity.

3 Invited talk: Testdata in systems with complex infrastructure

Software testing is a relevant part of software development. As stated in the talk, tests cost a lot of time of the development. Also maintaining the already created test data requires a lot of work. For testing, the V-model is a very popular development process. At this model every development phase has an assigned testing phase. All of these testing phases require different kind of test data. Reusable data is desired for good testing. In real world software projects productions data is often used as testing data. This usually violence privacy issues.
The talk did state different ways of providing data that will not rise privacy issues. Subset is one important part of preparing test data. Relationships in the database model can be altered. Also there are some relationship in the database that are not using foreign ids. For example, a String value is used which is related to other entries in the database model. These kind of relationships have to handled as well. These relationships can be used for subsets of data. Another subset of data is vertical and horizontal subset. The vertical subset will just use a limited number of records of the data. The horizontal subset instead will just some parameters or objects in from the data.
Another important part is masking the data. This will ensure that privacy issues will be prevented. De-identification is one where ids which are used in the database will be exchanged to new ones. But this will not ensure that privacy issues will not occur. Anonymization is a one way method, where data will be replaced with new data. Also other masking methods where described and what the advantages and disadvantages of them are. The other masking methods were cross reference list, hash lookup list, random lookup, random text/number, fixed text and shuffling.
Another interesting part of the talk was the automated generated test data. The data will be generated based on generators. For example, an customer account, with a mail, a name and an id. These would require some kind of data that will be used in the generations. The id would be a number, the name a random string and the mail, a random string which full fills the mail format.

Such generators allow to create test data in any size.

For the integration testing proxy methods were explained to provide the data from proxies. This allows to change the data for different test scenarios. The proxy has to be maintained as well. One solution for this is to let it maintain all different proxies by the related developers because they know how the data looks like. This will bring the issue that some data might have to be related to each other provide real test data. This can be solved by having a test data management which will provide the shared data to all related data providers. Another solution to provide the data is using the production data and use some kind masking between the environments. Typical environments are development, integrity testing, user acceptance testing and production environments. This will help to provide real like data without violating privacy issues. But the test data still requires maintenance because new data might be introduced in the development phase and these will not exist in the production environment.

This talk did provide a good overview of different methods to provide test data for software development.

4 Session C: Intrusion detection

This session was about intrusion detection. The first presentation *Constrained Row-Based Bit-Parallel Search in Intrusion Detection* did show a new solution to detect attacks. The main problem they solve is to prevent known attacks with simple changes. For example, some attacks might be repeated with some bits are changed, but will still use the same exploit. They used the bit-parallelism algorithm and modified it, to allow n count disparities. The algorithm was explained and they made some tests with that algorithm to proof the concept. But they missed to do tests with real bit operations, which would provide the efficiency that is the big advantage of bit based algorithm.
The other talk *Data-driven Approach to Information Sharing using Data Fusion and Machine Learning for Intrusion Detection* did show a complete different approach for intrusion detection. It uses machine learning for intrusion detection, but will not just use already known attack patterns. They did show a system which will also include manual created data from other sources. For example, a source can be from news about new exploit mechanism. These new inputs will be used to improve the detection rate from the machine learning algorithm. This was explained as cyclic machine learning because of the cycle inside the system.

5 Invited Talk: Computations and Cryptology

This talk did give an overview about the history of cryptology. It did talk about the encryption systems used in world war II. The presentation did explain the main principles on how these systems got decrypted using plain text attacks. Famous people for cryptology were presented like Alan Turing who was involved in decrypting the enigma. The last part of the presentation was about quantum computing which would solve the hard problem of square root. This would make a lot of cryptography algorithm obsolete. But for now there are no quantum computer that can be used.

6 Session D: Cellular and network security

The first talk was *Security Vulnerabilities of Cellular Communication Systems*. This talk did state out vulnerabilities from GSM, UMTS and LTE. It also did explain some attacks that exists. The main point of this presentation was that the same vulnerabilities do occur in newer systems. The same mistakes will be repeated on and on.
The other talk was *An End-to-End Security Model of Inter-Domain Communication in Network Function Virtualization*. It did present a virtualization method mainly used to virtulize networks. The talk did present an abstract security model to secure the virtual network. It mainly had a vertical and a horizontal security model. The vertical was between the different layers and the horizontal was between different domains.

7 Invited talk: Game Development in Practice – Towards 10 years with Turbo Tape Games

This talk was not about a research. The CEO of Turbo Tape Games did present his history of game developing. All games he did develop were presented. It mainly had two different parts of games. The classical games for entertaining and games for education. He did show some pitfalls that can occur in a game development company. It was an entertaining presentation.

8 Session E: Game Development in Practice – Towards 10 years with Turbo Tape Games

The last session started with the presentation *On trends in low-level exploitation*. This paper was an overview of existing exploits. Different kind of exploits were explained. Classic examples like a buffer overflow was explained. But also the methods to exploit a vulnerability that allows to modify the program counter. The prevention mechanics like NX-bit, ASLR, et cetera were explained as well. One interesting example did show that some vulnerabilities just exist because the compiler does some optimizations in the code. In the example the optimization did remove the sanitization method.
The last presentation was *Decryption phase in Norwegian electronic voting*. This did show a solution to prevent manipulation on electronic voting. Also did it provide a solution to prevent that the votes can be tracked back to the voting person. It uses a zero knowledge protocol and a shuffling system to prevent these kind of manipulations.