Principal Investigator: Pande, Santosh
Organization: GA Tech Res Corp - GIT
Submitted By: Pande, Santosh - Principal Investigator
Title: CT-ISG: Intrusion Tolerant Software: Achieving, Confidentiality, Availability and Integrity Simultaneously

Project Participants

Senior Personnel
Name: Pande, Santosh
Worked for more than 160 Hours: Yes
Contribution to Project:

Post-doc

Graduate Student
Name: Zhang, Kun
Worked for more than 160 Hours: Yes
Contribution to Project:
   Kun worked on extending the MICRO papers to make the work applicable to full C solutions.

Undergraduate Student

Technician, Programmer

Other Participant

Research Experience for Undergraduates

Organizational Partners

Other Collaborators or Contacts
None

Activities and Findings

Research and Education Activities:
The main research activities undertaken during the reporting period of the project include:

1. Development of a concept of intrusion sensors to augment our year II work on anomaly detection.

2 New Research on Memory Trust Properties of Mobile Code

On the educational side, the following activity was undertaken:
1. An undergraduate project to deal with logging and on-the-fly analysis of event data during execution.

Findings:
These finds summarize the last phase of the project:

1. Intrusion sensors
Intrusion sensors are a novel concept of triggering monitoring and analysis activity based on observations of the past, a previous few steps of execution and perhaps expected control and data-flow behaviors of the executing application - when one of the set conditions is met at the current program point. Intrusion sensors allow the generalization branch-co-relations anomaly detector developed this research in their 2nd year. The intrusion sensor is can be invoked via each sentries (some API abstraction) posted at key program points either by a user or the trusted compiler. Intrusion sensors execute inside the secure domain as tamper-resistant hardware functions. It can gather the necessary execution information by configuring and utilizing the hardware program counters and by generating appropriate calls for further analysis. The sensors have an ability to set up different sets of programmable performance counters at different program points which can lead to different types of monitoring and analysis. The intrusion sensors can also request collaborative support from other ones - regarding state change and such combined information can powerfully managed and can contribute to nearby state behaviors observations. Intrusion sensors in such cases can collaboratively perform the statistical functions such as co-relation detection, some type of multi-variate analysis etc. Intrusion sensors research developed in this phase of the research allows generalization of both branch-based anomaly detection by leveraging the larger number of programmable performance counters. Instead of limiting to only branches, one could use any other program metric such as load/stores or cache misses etc. to detect anomaly and also create and co-relate the anomaly models at different execution points.

In short, intrusion sensors provide a unifying basis for programming and experimenting with different types of intrusion detection schemes based on performance data of the application.

2. Memory Trust Properties of Mobile Code
After working on the memory integrity in our 2nd year, we decided to take more global view of the memory and tackle some issues regarding the trust properties going beyond normal single or multi-threaded shared memory spaces.

Mobile apps are everywhere Mobile code will the biggest application driver wherein an app migrates on a peer to peer basis performing the computation. In such cases different pieces of memory/data are touched by the mobile app on different mobile platforms. Often data control policies exist on distributed systems but they only deal with access control. In this mobile apps's scenario memory is an integrated part of application's semantics and can be taken into account into building trust into the execution.

A new programming model: Towards the above, first a new programming model was introduced in which a Java mobile agent carries mobile code and its state with it from one mobile peer to peer. At each peer, it will perform a partial computation using the local data mentioned in the computation and carry the partial solution. The mobile agent migrates from one peer to the other; the order of migration is determined by the program's semantics and restrictions imposed by the data trust semantics and thus by a compiler. Interesting data policies can be stipulated as a part of the application semantics and are enforced.

Maintaining trust semantics and obeying it poses interesting challenges. Unlike traditional access control mechanisms, trust semantics could look like: trust friend's, or: trust friend friend; Moreover policies about access to only certain field' of data, certain data not being accessible after the access of some other data could be interesting.

Using the above stipulation, the compiler generates smart mobile code which can be efficiency scheduled from peer to peer for execution; number of migrations should be minimized; minimize the partial results stored across mobile peers. Performing integrity checking of such code is very challenging and this is where an interesting solution exists. In order to check the integrity of local value A in located in mobile host Z one has to construct all possible program orders to check possible changes to A's memory state. Tracking all such causality state changes can be super expensive leading to state explosion and is infeasible in distributed asynchronous setting. However, by leveraging and tracking compiler generated migrations, one need not check-all the causality checks saving a tremendous overhead. Remember the compiler will create one migration schedule moving the mobile code from
one peer to the next.
The state saving generated by the above scheme over the baseline one is big.

Educational Activity:

Thirdly, an undergraduate class was given by the PI to efficiently gather, summarize and represent collected events. The idea is to use this compacted trace in real time to perform forensics. Several challenges are encountered. It must be compact and accurate. Events must be classified and causally ordered, the representation should be hierarchical, and has time steps. Several optimizations had to be done for compaction and smarts built-in into the query engine. Characterization about compaction and time complexity of queries and their accuracy was done through measurement.

Training and Development:

Two Ph.D. students (Tao Zhang and Xiaotong Zhuang) have graduated who have worked on the project. Both are in IBM T. J. Watson Research Center in compiler analysis group. The third PhD student Kun Zhang has also graduated in December 2009 and she has taken up a position in Qualcomm, Inc.

Apart from previous technical works of the years this new work extended into mobile code security and carried forward the techniques to mobile code. This is an emerging area in security. Future of mobile apps is very important esp. peer to peer. Getting both traditional view of memory and trust view in mobile apps should be broaden Kun's perspectives.

Outreach Activities:

None

Journal Publications


Books or Other One-time Publications

Web/Internet Site

Other Specific Products
Contributions

Contributions within Discipline:
The notion of intrusion sensor provides a unifying basis to different types of intrusion detection systems. One could combine different artifacts through the chosen performance counters in a unified manner. Such a unification would allow us to better understanding of the contributing factors and also experiment with detection of intrusions using different program artifacts.

Contributions to Other Disciplines:
Providing semantic basis to mobile code data spaces is a very important problem and solving such a hard problem of integrity in such setting is very important for paving way for secure app development. The hard problem of integrity checking was exposed by this work and then solved by falling back upon the generated migration schedule. Thus, the work showed that the integrity checking is tractable for mobile code data spaces modelled in a particular manner. Such a result would open up the development of mobile peer-to-peer apps.

Contributions to Human Resource Development:
Two PhDs have graduated and joined IBM Research at Watson Research Center in past. In addition, in 2009, Kun Zhang graduated and joined Qualcomm, Inc.

Contributions to Resources for Research and Education:

Contributions Beyond Science and Engineering:
Establishment of memory integrity and trust properties in mobile peer-to-peer application spaces in a tractable manner would pave for confident way of developing many such mobile apps

Conference Proceedings

Categories for which nothing is reported:
Organizational Partners
Any Book
Any Web/Internet Site
Any Product
Contributions: To Any Resources for Research and Education
Any Conference