Analysis and Visualization of Massive Execution Traces

Nora Huang
March 2017

Background

Network is unsafe for the computers running on it

Vulnerabilities in the software is exploitable

Avoiding Vulnerabilities

- Build a secure software from the ground
- Detect the vulnerabilities and fix them

Security Analysis Methods

- Static Analysis (with source code)
- Dynamic Analysis (with/without source code)
- Combination of these two

Security Analysts in Our Case

Study how a program runs
without source code

Current Model

The model of Security Analysis Three Stages:

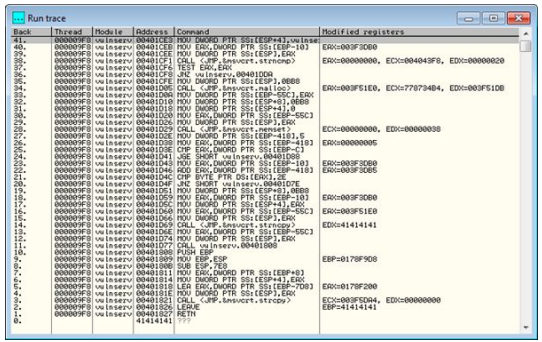
Capture Execution Trace

Capture the **instruction** and **memory change** trace of a running application

Atlantis Pintool

A Running Application

Process Trace

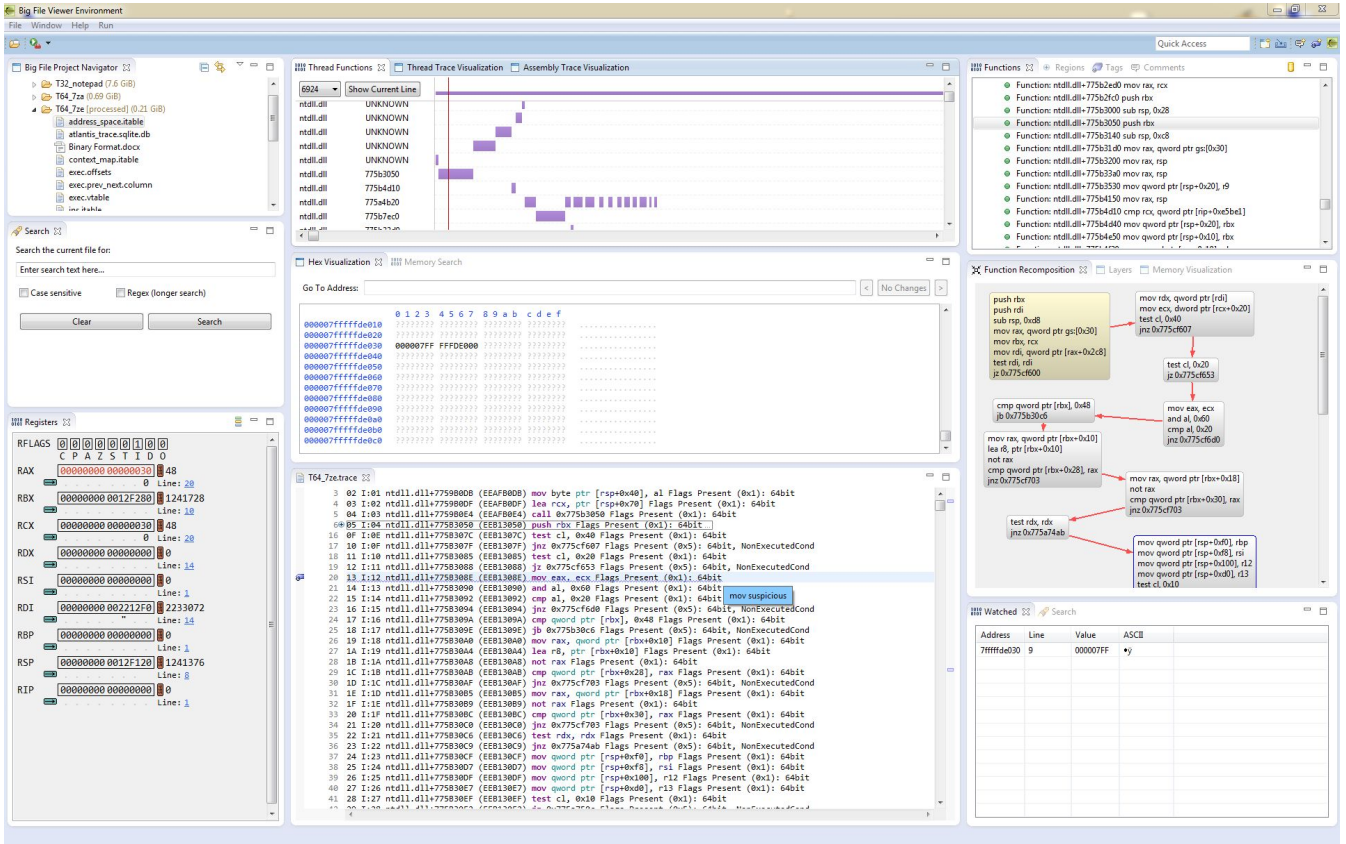


Execution Trace

Gibraltar:
Massive trace processor

- Decoded traces
- Reconstructed memory states

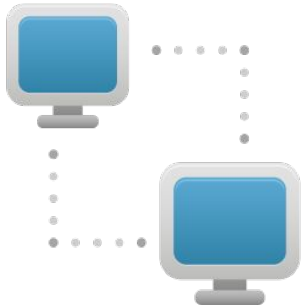
Visualize Trace



Problem

- Vulnerabilities occur when they interact with other systems.

- Interaction form is various



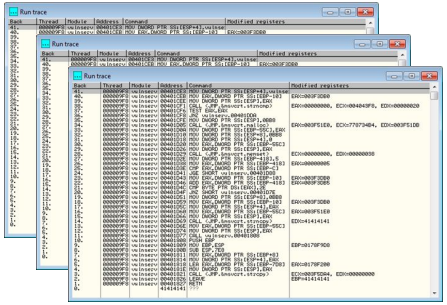
Our Goal

- Keep the Current working model
- Update the tools to solve the problem

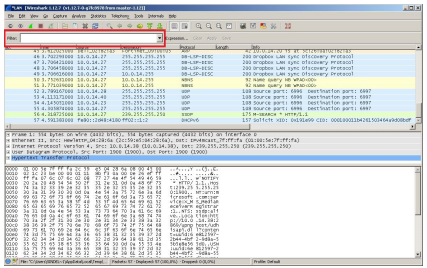
Comprehensive Pintool

- Able to capture execution and communication trace from running applications
- The communication form of the applications are various

New Traces file processor



Multiple Massive Execution Traces

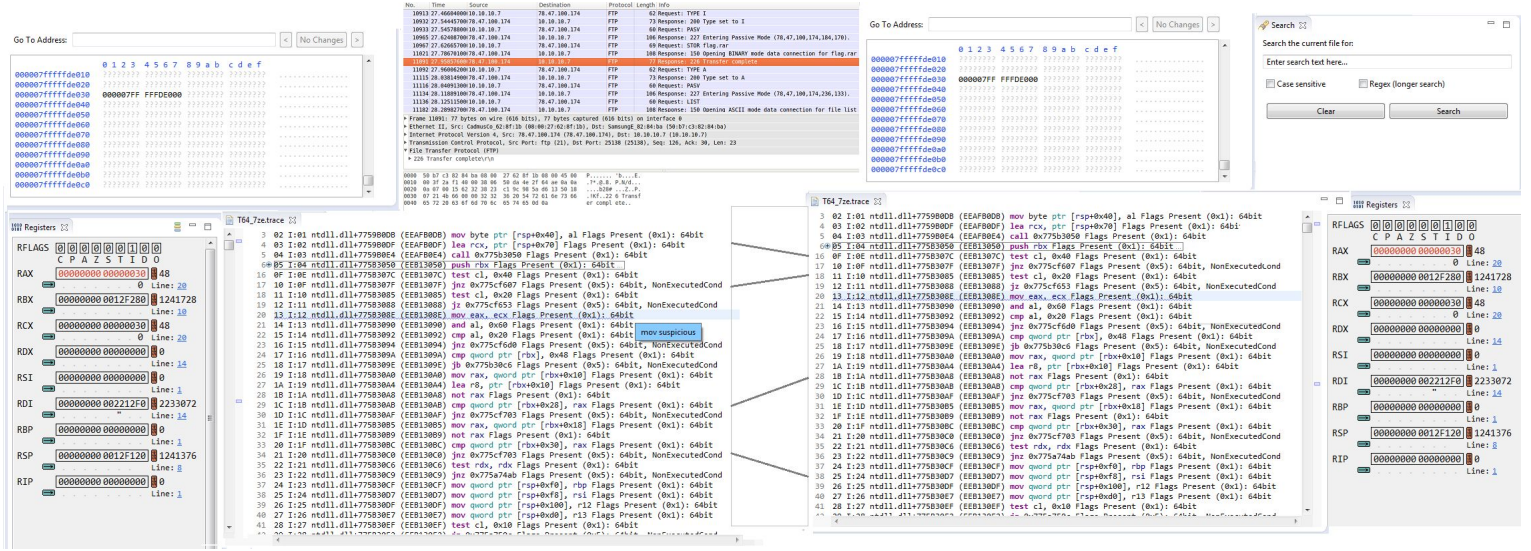


Communication Traces

Gibraltar: Trace files processor

Information in the traces should be summarized, integrated, compacted and filtered which is suitable for interactive visualization

Dual Traces Viewer



1. Dual Instruction View
2. Dual Memory Views
3. Dual Register View
4. Application Synchronisation Points
5. Message View
6. Text Search View

All views should be synchronised and updated when the user navigates the instruction lines.

Next Steps

- Get the dual execution traces along with the communication trace
- Analysis the traces, find a way to get the synchronisation points
- Design the new data structure for multi traces
- Design the viewer layout
- More research on view synchronisation methods

Verification method

- Case study - local traces, traces from DRDC

Other Functionalities

- Automatic Vulnerabilities detection
- Instruction View Scale
- Overview of the execution
- Dashboard for Statistics, such as most frequently access memory


Thank you!