

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:

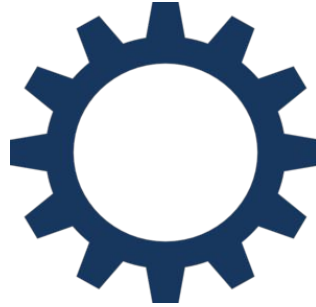


Capture Execution Trace

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



Atlantis Pintool



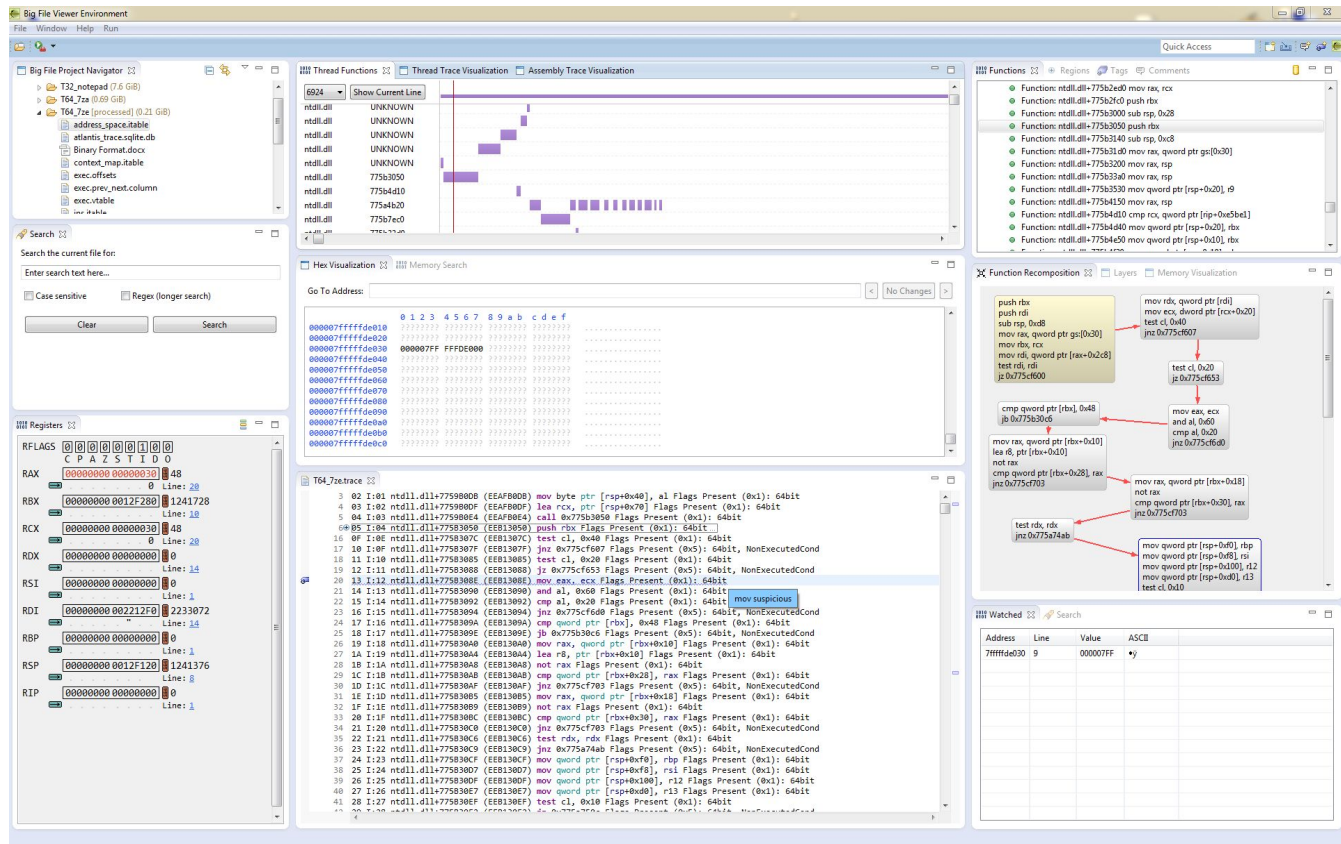
A Running Application

PC	Thread	Module	Address	Comment	Modified registers
41	0000000F	ntuserui	0040102E	MOV EAX, PTR [ESP+4], winlog	ECX=00F3000
42	0000000F	ntuserui	00401030	MOV EAX, DWORD PTR [ESP+4], 139	
43	0000000F	ntuserui	00401032	MOV EAX, DWORD PTR [ESP], EAX	
44	0000000F	ntuserui	00401034	CALL JMP, EAX	ECX=004043FD, ECX=00000020
45	0000000F	ntuserui	00401036	CALL JMP, EAX	
46	0000000F	ntuserui	00401038	JNC JNEN, 0040103D	
47	0000000F	ntuserui	0040103A	CALL JMP, EAX	
48	0000000F	ntuserui	0040103C	CALL JMP, EAX	ECX=00F31E0, ECX=78774B4, ECX=00F31D0
49	0000000F	ntuserui	0040103E	CALL JMP, EAX	
50	0000000F	ntuserui	00401040	CALL JMP, EAX	
51	0000000F	ntuserui	00401042	MOV EAX, DWORD PTR [ESP+4], 139	
52	0000000F	ntuserui	00401044	MOV EAX, DWORD PTR [ESP+4], 139	
53	0000000F	ntuserui	00401046	MOV EAX, DWORD PTR [ESP+4], 139	
54	0000000F	ntuserui	00401048	MOV EAX, DWORD PTR [ESP+4], 139	
55	0000000F	ntuserui	0040104A	MOV EAX, DWORD PTR [ESP+4], 139	
56	0000000F	ntuserui	0040104C	MOV EAX, DWORD PTR [ESP+4], 139	
57	0000000F	ntuserui	0040104E	MOV EAX, DWORD PTR [ESP+4], 139	
58	0000000F	ntuserui	00401050	MOV EAX, DWORD PTR [ESP+4], 139	
59	0000000F	ntuserui	00401052	MOV EAX, DWORD PTR [ESP+4], 139	
60	0000000F	ntuserui	00401054	MOV EAX, DWORD PTR [ESP+4], 139	ECX=00000000, ECX=00000000
61	0000000F	ntuserui	00401056	CALL JMP, EAX	
62	0000000F	ntuserui	00401058	CALL JMP, EAX	
63	0000000F	ntuserui	0040105A	CALL JMP, EAX	ECX=00000005
64	0000000F	ntuserui	0040105C	CALL JMP, EAX	
65	0000000F	ntuserui	0040105E	CALL JMP, EAX	ECX=00F3000
66	0000000F	ntuserui	00401060	CALL JMP, EAX	ECX=00F30005
67	0000000F	ntuserui	00401062	CALL JMP, EAX	
68	0000000F	ntuserui	00401064	CALL JMP, EAX	
69	0000000F	ntuserui	00401066	CALL JMP, EAX	ECX=00F3000
70	0000000F	ntuserui	00401068	CALL JMP, EAX	
71	0000000F	ntuserui	0040106A	CALL JMP, EAX	
72	0000000F	ntuserui	0040106C	CALL JMP, EAX	ECX=00F3000
73	0000000F	ntuserui	0040106E	CALL JMP, EAX	ECX=00F3000
74	0000000F	ntuserui	00401070	CALL JMP, EAX	ECX=00F3000
75	0000000F	ntuserui	00401072	CALL JMP, EAX	ECX=00F3000
76	0000000F	ntuserui	00401074	CALL JMP, EAX	ECX=00F3000
77	0000000F	ntuserui	00401076	CALL JMP, EAX	ECX=00F3000
78	0000000F	ntuserui	00401078	CALL JMP, EAX	ECX=00F3000
79	0000000F	ntuserui	0040107A	CALL JMP, EAX	ECX=00F3000
80	0000000F	ntuserui	0040107C	CALL JMP, EAX	ECX=00F3000
81	0000000F	ntuserui	0040107E	CALL JMP, EAX	ECX=00F3000
82	0000000F	ntuserui	00401080	CALL JMP, EAX	ECX=00F3000
83	0000000F	ntuserui	00401082	CALL JMP, EAX	ECX=00F3000
84	0000000F	ntuserui	00401084	CALL JMP, EAX	ECX=00F3000
85	0000000F	ntuserui	00401086	CALL JMP, EAX	ECX=00F3000
86	0000000F	ntuserui	00401088	CALL JMP, EAX	ECX=00F3000
87	0000000F	ntuserui	0040108A	CALL JMP, EAX	ECX=00F3000
88	0000000F	ntuserui	0040108C	CALL JMP, EAX	ECX=00F3000
89	0000000F	ntuserui	0040108E	CALL JMP, EAX	ECX=00F3000
90	0000000F	ntuserui	00401090	CALL JMP, EAX	ECX=00F3000
91	0000000F	ntuserui	00401092	CALL JMP, EAX	ECX=00F3000
92	0000000F	ntuserui	00401094	CALL JMP, EAX	ECX=00F3000
93	0000000F	ntuserui	00401096	CALL JMP, EAX	ECX=00F3000
94	0000000F	ntuserui	00401098	CALL JMP, EAX	ECX=00F3000
95	0000000F	ntuserui	0040109A	CALL JMP, EAX	ECX=00F3000
96	0000000F	ntuserui	0040109C	CALL JMP, EAX	ECX=00F3000
97	0000000F	ntuserui	00		



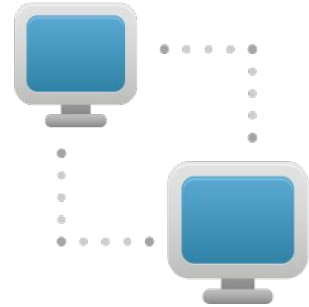
- 8

Visualize Trace



Problem

- Vulnerabilities occur when they interact with other systems.
- Interaction form is various



Our Goal

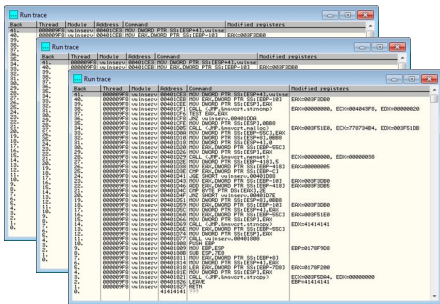
- Keep the Current working model
- Update the tools to solve the problem

Comprehensive Pintool



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

New Traces file processor

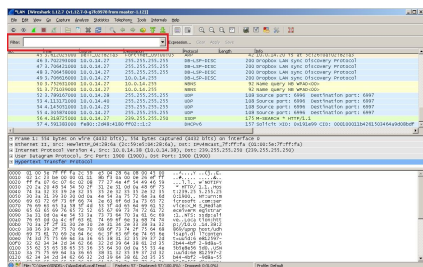


Multiple Massive Execution Traces



Gibraltar: Trace files processor

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



Communication Traces

Dual Traces View

The screenshot displays the 'Dual Traces View' in a debugger, showing two parallel traces of program execution. The left trace is the 'Instruction Trace' and the right trace is the 'Register Trace'.

Instruction Trace (Left):

- Go To Address:** A search bar with 'No Changes' and a dropdown arrow.
- Search:** A search bar with 'No Changes' and a dropdown arrow.
- Registers:** A list of registers (RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, RIP) with their current values and addresses.
- Instruction List:** A list of instructions with their addresses, mnemonics, and comments. The instruction at address 0000000000000000 is highlighted.
- Registers:** A list of registers (RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, RIP) with their current values and addresses.

Register Trace (Right):

- Go To Address:** A search bar with 'No Changes' and a dropdown arrow.
- Search:** A search bar with 'No Changes' and a dropdown arrow.
- Registers:** A list of registers (RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, RIP) with their current values and addresses.
- Register List:** A list of registers (RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, RIP) with their current values and addresses.
- Register Values:** A list of register values (RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, RIP) with their current values and addresses.

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!

