Process Tracking for Forensic Readiness

Yi-Ching Liao
Norwegian Information Security Laboratory

yi-ching.liao@hig.no
ABSTRACT

• Summarize the research on process tracking for forensic readiness
  ▪ the state-changing activities of processes
  ▪ cost-benefit analysis of process tracking
  ▪ the architecture for process tracking
  ▪ privacy implications of process tracking
PROBLEM STATEMENT

• Forensic analysis
  ▪ suffers from insufficient logging of events

• Current system loggers
  ▪ do not record enough information for incident analysis and replay

• Comprehensive process tracking
  ▪ provides precise, timely, complete, and dependable information for incident investigation and replay
  ▪ recovers the traceability links between the incident and the person or action accountable for the incident
RESEARCH QUESTIONS

1. What are the state-changing activities of processes?

2. How effective, efficient, and expensive is comprehensive process activity tracking?

3. Which hardware/software architecture facilitates process activity tracking?

4. What are privacy implications for users of systems that support comprehensive traceability?

5. How does comprehensive traceability affect evidence gathering and the legal process?
STATE-CHANGING ACTIVITIES OF PROCESSES

OVERVIEW

Research question
What are the state-changing activities of processes?

Literature review
Program execution monitors
Program comprehension through dynamic analysis
Execution replay systems

Selection of existing process tracking systems
9 process activity tracking systems for security
11 process activity tracking systems for debugging

Evaluate from the perspectives of forensic analysis and forensic readiness
Logging method
Implementation method
Tracing granularity
Replay boundary

Process activity tracking can provide sufficient evidence for investigation if the tracking is
Transition-based
System-level
Kernel-space implementation
## STATE-CHANGING ACTIVITIES OF PROCESSES

### SUMMARY

<table>
<thead>
<tr>
<th>System name</th>
<th>Logging method</th>
<th>Tracing granularity</th>
<th>Replay boundary</th>
<th>Design purpose</th>
<th>Implementation method</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReVirt [17]</td>
<td>SB</td>
<td>IL</td>
<td>SL</td>
<td>S</td>
<td>VM</td>
<td>L</td>
</tr>
<tr>
<td>AskStrider [50]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>U</td>
<td>W</td>
</tr>
<tr>
<td>Capture [45]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>K</td>
<td>W</td>
</tr>
<tr>
<td>XenLR [28]</td>
<td>TB</td>
<td>IL</td>
<td>SL</td>
<td>S</td>
<td>Hypervisor</td>
<td>L</td>
</tr>
<tr>
<td>Process Hacker [42]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>K</td>
<td>W</td>
</tr>
<tr>
<td>Process Monitor [31]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>K</td>
<td>W</td>
</tr>
<tr>
<td>Carbon Black [10]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>N/A</td>
<td>W</td>
</tr>
<tr>
<td>FileSure [9]</td>
<td>TB</td>
<td>PL</td>
<td>N/A</td>
<td>S</td>
<td>N/A</td>
<td>W</td>
</tr>
<tr>
<td>Tornado [13]</td>
<td>TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>K &amp; U</td>
<td>L</td>
</tr>
<tr>
<td>Jockey [43]</td>
<td>TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>B &amp; U</td>
<td>L</td>
</tr>
<tr>
<td>iBlog [20]</td>
<td>TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>B &amp; U</td>
<td>L</td>
</tr>
<tr>
<td>Flashback [46]</td>
<td>SB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>K</td>
<td>L</td>
</tr>
<tr>
<td>iDNA [5]</td>
<td>TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>B</td>
<td>W</td>
</tr>
<tr>
<td>ODR [1]</td>
<td>TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>B &amp; K</td>
<td>L</td>
</tr>
<tr>
<td>Respee [26]</td>
<td>SB &amp; TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>K</td>
<td>L</td>
</tr>
<tr>
<td>DoublePlay [48]</td>
<td>SB &amp; TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>K</td>
<td>L</td>
</tr>
<tr>
<td>FDR [51]</td>
<td>SB &amp; TB</td>
<td>IL</td>
<td>SL</td>
<td>D</td>
<td>H</td>
<td>Any</td>
</tr>
<tr>
<td>BugNet [35]</td>
<td>SB &amp; TB</td>
<td>IL</td>
<td>UL</td>
<td>D</td>
<td>B &amp; H</td>
<td>L</td>
</tr>
<tr>
<td>QuickRec [40]</td>
<td>TB</td>
<td>IL</td>
<td>SL</td>
<td>D</td>
<td>H &amp; K</td>
<td>L</td>
</tr>
</tbody>
</table>

*SB=State-based; TB=Transition-based
IL=Instruction level; PL=Process level
SL=System level; UL=User level
D=Debugging; S=Security
B=Binary patching; H=Hardware; K=Kernel-space; U=User-space
L=Linux; W=Windows
STATE-CHANGING ACTIVITIES OF PROCESSES

CONCLUSION AND QUESTION RAISED

• Process activity tracking can provide sufficient evidence for investigation if the tracking is
  ▪ transition-based
  ▪ system-level
  ▪ kernel-space implementation

• To strike a balance between the forensic effectiveness and efficiency, we need to
  ▪ evaluate the soundness, completeness, and cost of process activity tracking
COST-BENEFIT ANALYSIS OF PROCESS TRACKING

OVERVIEW

<table>
<thead>
<tr>
<th>Research question</th>
<th>How effective, efficient, and expensive is comprehensive process activity tracking?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature review</td>
<td>Kernel tracing for forensics for reverse engineering for system behavior analysis</td>
</tr>
<tr>
<td>Selection of existing kernel tracing systems</td>
<td>Strace SystemTap LTTng</td>
</tr>
<tr>
<td>Cost-benefit analysis</td>
<td>Benefit analysis: the comprehensive coverage Cost analysis: the performance and storage overheads</td>
</tr>
<tr>
<td>Kernel tracing systems can meet the two objectives of forensic readiness</td>
<td>Maximize the capability of collecting credible evidence Minimize the cost of investigation</td>
</tr>
</tbody>
</table>
COST-BENEFIT ANALYSIS OF PROCESS TRACKING

BENEFIT ANALYSIS RESULTS

Comprehensive coverage in 32-bit architecture

Comprehensive coverage in 64-bit architecture
COST-BENEFIT ANALYSIS OF PROCESS TRACKING
COST ANALYSIS RESULTS: PERFORMANCE OVERHEAD

Performance overhead per successfully traced system call (sec)

- strace
- SystemTap (program-specific)
- SystemTap (system-wide)
- LTTng

32-bit and 64-bit systems compared.
COST-BENEFIT ANALYSIS OF PROCESS TRACKING
COST ANALYSIS RESULTS: STORAGE OVERHEAD

Storage overhead (filtered) per successfully traced system call (MB)
COST-BENEFIT ANALYSIS OF PROCESS TRACKING
CONCLUSION AND QUESTION RAISED

• Kernel tracing systems can meet the two objectives of forensic readiness (Tan, 2001)
  ▪ maximize the capability of collecting credible digital evidence
  ▪ minimize the cost of investigation

• However
  ▪ high performance and storage overheads caused by dynamic instrumentation

• For cost-benefit trade-off, we need to
  ▪ design the architecture for flexible and adjustable process tracking
ARCHITECTURE FOR PROCESS TRACKING

OVERVIEW

Research question
Which hardware/software architecture facilitates process activity tracking?

Literature review
Digital event reconstruction systems
Digital forensics frameworks

Set design goals
Completeness
Pertinence
Reliability
Privacy preservation

Feasibility study
Kernel tracing systems
System model building tool
Visualization tool

Employing kernel tracing systems in readiness phase of digital forensics frameworks
Ensure the potential evidence is readily available in an acceptable form when an incident or a crime occurs.
ARCHITECTURE FOR PROCESS TRACKING
PROTOTYPE FRAMEWORK

Deployment Phase
Receiving Detection Alerts / Authorization Notifications

- Detection Alerts from NIDS, firewalls, etc.
- Authorization Notifications from Related Authorities

Readiness Phase
Collecting System Call Traces

- Static Instrumentation for Background Potential Evidence
- Dynamic Instrumentation for Foreground Potential Evidence

Investigation Phase 1
Preserving States / Crime Scenes

- Securing Trace Storage
- Remote Logging

Investigation Phase 2
Recognizing Evidence

- Examining Personally Identifiable Information
- Translating System Call Traces into Events

Investigation Phase 3
Reconstructing Events

- Sequencing and Translating Events to Higher Layer of Abstraction
- Hypothesis Testing
ARCHITECTURE FOR PROCESS TRACKING FEASIBILITY STUDY: SYSTEM MODEL

- Generated System Model
  - `ls` command
  - each edge label
    - transition probability
  - the state transitions of resource usage
ARCHITECTURE FOR PROCESS TRACKING
FEASIBILITY STUDY: RECONSTRUCTING EVENTS

• **Reconstructing Events**
  - play back the system activity history as an animation
    - Gource
      » generates a dynamic tree to animate the software development history
      » user who commits the update floating near the files
      » color the update actions (add, modify, and delete)
      » animate the history by the timelines
ARCHITECTURE FOR PROCESS TRACKING

CONCLUSION AND QUESTION RAISED

- Employing kernel tracing systems in readiness phase of digital forensics frameworks can
  - ensure the potential evidence is readily available in an acceptable form when an incident or a crime occurs

- Interpret the meaning of digital events
  - cause and effect analysis
  - layers of abstraction
# PRIVACY IMPLICATIONS OF PROCESS TRACKING

## OVERVIEW

<table>
<thead>
<tr>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are privacy implications for users of systems that support comprehensive traceability?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Literature review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surveillance impact assessment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Define problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compliance assessment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The Goal/Question/Metric paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social impact model</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Developing metrics for surveillance impact assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensure the negative consequences are minimized to acceptable levels</td>
</tr>
</tbody>
</table>
PRIVACY IMPLICATIONS OF PROCESS TRACKING SURVEILLANCE IMPACT ASSESSMENT

• **Objective**
  - identify and assess the impacts posed by surveillance technologies on different dimensions of privacy
PRIVACY IMPLICATIONS OF PROCESS TRACKING
THE GOAL/QUESTION/METRIC PARADIGM

Strength
- By whom: Nominal
- For whom: Nominal

Immediacy
- When: Nominal
- Where: Ordinal

Number
- Of what: Nominal
- How: Interval

By whom: Nominal
For whom: Nominal
Why: Ordinal
When: Nominal
Where: Ordinal
Of what: Nominal
How: Interval
PRIVACY IMPLICATIONS OF PROCESS TRACKING

CONCLUSION AND QUESTION RAISED

• Social Impact
  - an ongoing chain process of continuing influences

• Developing Metrics for Surveillance Impact Assessment
  - can ensure the negative consequences are minimized to acceptable levels

• Metric Validation through a Feasibility Study
  - utilize the metrics to compare the impacts between
    - kernel tracing systems
    - application-level logging systems
  - provide credible information for decision-making
Thank you

Yi-Ching Liao

Norwegian Information Security Laboratory

yi-ching.liao@hig.no