Forensix: A Robust, High-Performance Reconstruction System

Ashvin Goel, Mike Shea, Sourabh Ahuja, Wu-chang Feng, Wu-chi Feng, David Maier, Jonathan Walpo
University of Toronto, OGI@OHSU

Tracking suspicious behaviour and troubleshooting system problems takes very long!

Where did the attack come from?

What happened as a result of the attack?

What security hole was exploited?

Its hard to capture and analyze system activity...

Motivation: need to reconstruct system activity quickly and accurately

Current logging is inaccurate

N/W tracing, File system logs, Appl logs, System logs, Process accounting

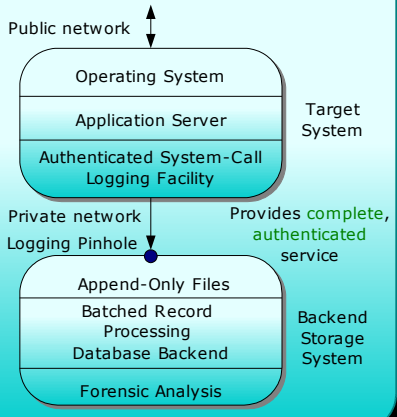
VM approaches are accurate but replay and analysis can be slow

Question: what about system-call tracing?

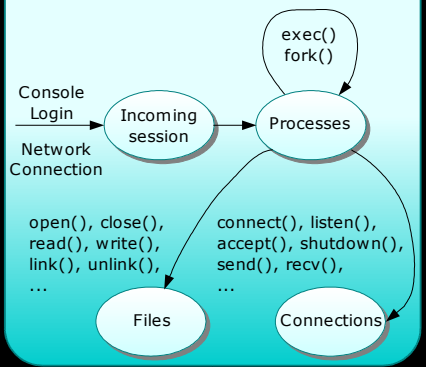
Provides:
- Completeness
- Reproducibility

Challenges:
- Authentication?
- Efficient?
- Powerful?
- Fast?

Forensix Architecture



Attributing System Activity



Powerful Queries

Query 1: Show all user sessions that executed /bin/sh from daemon processes other than sshd, telnetd, or login and group sessions by user

Query 2: Generate a system activity log for all sessions generated by Query 1

Query 3: Show all activity for a particular user session S, whee S is denoted with a source IP address and port, a user ID and a connection time-stamp

Examples of Queries

Query Name	Arguments	Output
Active_Processes	start_time, end_time	List all active processes within a given time interval.
Immediate_Children	PID	List all immediate children of a process.
Children	PID	List all children of a process.
Immediate_Parent	PID	List immediate parent of a process.
Parents	PID	List all parents of a process.
Fds_written	PID, start_time, end_time	List all file descriptors written by a process within time interval and time when they were written.
All_FDs	PID, filename, fd_list, time	List all file descriptors that refer to a filename or to other file desciptors in fd_list at a given time.
Did_Process_Write	PID, filename, start_time, end_time	Did process write to filename within a given time interval?
Writers	Filename, start_time, end_time	List all processes that wrote to filename within a given time interval.
IO	PID, fd_list	List the timing and the data for I/O performed on file descriptors in fd_list by a process.
Replay_Shell	PID	Run I/O query on file descriptors 0, 1, 2 for a shell process (replay a shell process).

Modular Queries

```
Fds_Written(PID, start_time, end_time) {
    SELECT fd, data FROM io, event
    WHERE io.parent = event.id
        AND event.pid = %1
        AND event.syscall = 4 /* write */
        AND event.date > %2
        and event.date < %3;
}
```

Did_Process_Write
uses Fds_Written, All_FDs

Writers
uses Did_Process_Write, Active_Processes

IO
similar to Did_Process_Write
but produces writes

Replay_Shell
special case of IO

Performance Overhead

Kernel Build Times

	Auditing off	Auditing on Network off	Auditing on Network on
Total Time	233.2 s	247.1 s (6%)	252.0 s (8%)
System Time	14.0 s	26.3 s	30.7 s

Webstone throughput

	Auditing off	Auditing on Network off	Auditing on Network on
Throughput (Mb/s)	296.8	276.2 (93%)	186.87 (63%)

Replay_Shell Time

Webstone test	100 s
Game community server	414 s (194 s with PID index)

Space Overhead

Kernel build test	30 GB/day
Webstone test	8.8 GB/day
Game community server	0.45 GB/day

Forensix reconstructs system activity quickly and accurately

Uses combination of:
- System-call tracing
- Separate, append-only, hardened storage
- Database for high-level querying

Future work:
- Storage efficiency
- Stepping stones
- Framework for queries