A Technique for Enabling and Supporting Debugging of Field Failures

James Clause and Alessandro Orso
Georgia Institute of Technology

This work was supported in part by NSF awards CCF-0541080 and CCR-0205422 to Georgia Tech.
Threats end Prince Harry's Iraq mission

Britain's Prince Harry is very disappointed about his latest military assignment. Harry will not serve in Iraq as a troop commander because of "a number of specific threats" against him, the UK's top general says. The prince would have led a troop of 12 men in four Scimitar armored reconnaissance vehicles, each with a crew of three, in the southern Iraqi city of Basra.

FULL STORY
Wednesday, May 16, 2007

(Some Queer) **OBVIOUS**
Fred Phelps to picket Falwell funeral. Vortex of douchebags likely to trigger collapse of the universe, rapture imminent

Blind man given a concealed weapon permit wants to prove a point, saying "people without sight still can carry because brains are more important than eyesight in securing public safety"

(Some Guy) **INTERESTING**
Man buys $8,000 worth of 'Forever Stamps' that can always be used to mail a letter, no matter what the current postage rate is. In related news, people still mail letters

Tinky Winky makes official comment on Jerry Falwell's recent passing

Greenpeace building replica of Noah's Ark on Mount Ararat to draw attention to global warming. "Global climate change is the biggest threat to our planet since the times of Noah. We are about to face a new flood"

Denying racism by saying you've had a 'colored boy' in your home may come off as being disingenuous

Your girlfriend breaks up with you. Do you: A) Get over it? B) Go out with friends? C) Stab yourself in the chest and crash your car into a transformer?

Prince Harry, who was kinda going to Iraq, then not going to Iraq, then really was going to Iraq, is definitely not being deployed to Iraq
You are invited to take a drink from the Firehose

Book Reviews: Security Metrics

Posted by samzenpus on Wednesday May 16, @03:35PM
from the protect-ya-neck dept.

Ben Rothke writes

“The goal of security metrics is to replace fear, uncertainty, and doubt (FUD) with a more formalized and meaningful system of measurement. The FUD factor is the very foundation upon which much of information security is built, and the outcome is decades of meaningless statistics and racks of snake oil products. Let's hope that Andrew Jaquith succeeds, but in doing so, he is getting in the way of many security hardware and software vendors whose revenue streams are built on FUD.”

Read below for the rest of Ben's
News in Brief for May 16, 2007

(05/11/07)
**Caring Technologies Receives $1 Million from NIH for Autism** Caring Technologies / TalkAutism of Boise, ID, collaborated with College of Computing Associate Professor Gregory Abowd to develop and patent a digital video recording system that lets professionals, caregivers and parents uniquely capture and then securely replay the last few moments before a behavior of interest or concern of a child with autism. [Download Article]

(05/10/07)
**College of Computing Students Win Industry Awards for Creating Next-Generation Mobile Applications**
Fifteen students at The Georgia Institute of Technology, including eleven from the College of Computing, were awarded $100,000 in cash prizes for creating next-generation mobile applications as part of the 2007 IMS Research Competition. [More]
News in Brief for May 2007

(05/11/07)
Caring Technology for Autism

Caring Technology for Autism - Caring Technology for Autism, a new recording system that lets professionals, caregivers, and parents uniquely capture and then securely replay the last few moments before a behavior of interest or concern of a child with autism.

Download Article

(05/10/07)
College of Computing Students Win Industry Awards for Creating Next-Generation Mobile Applications

Fifteen students at The Georgia Institute of Technology, including eleven from the College of Computing, were awarded $100,000 in cash prizes for creating next-generation mobile applications as part of the 2007 IMS Research Competition.

More
The application Firefox quit unexpectedly.

2007-05-16 16:04:53 -0400

EXC_BAD_ACCESS (0x0001)
KERN_PROTECTION_FAILURE (0x0002) at 0x0186af20

Thread 0 Crashed:
0  __vfprintf + 40
1  sprintf + 252
2  CreateVolFSPath(unsigned, unsigned long, char const*, unsigned long, char*) + 88
3  getattrlist_retry(unsigned, unsigned long, char const*, unsigned long, attrlist*, void*, unsigned long, unsigned long) + 68
4  GetVolFSAttributes(VolumeInfo*, unsigned long, char const*, unsigned long, unsigned long, FSAtributeInfo*, unsigned long, FSVolAttributeInfo*, unsigned char*) + 528
5  VolFSMount::_getattrs(unsigned long, char const*, unsigned long, unsigned long, FSAtributeInfo*, unsigned long, unsigned char*) + 52
6  FSMount::_getattrs(unsigned long, char const*, unsigned long, unsigned long, FSAtributeInfo*, unsigned long, unsigned char*) + 228
7  GetFSRefAttributes(FSMount*, FSRefPrivate const*, unsigned long, FSAtributeInfo*, unsigned long, char*) + 104
8  PBGetCatalogInfoSync + 156
9  FSGetCatalogInfo + 44

Close  Report…  Attach Debugger…
Field failures: Anomalous behavior (or crashes) of deployed software that occur on user machines
Problem Report for Firefox

Problem and system information:
Date/Time: 2007-05-16 16:00:01.424 -0400
OS Version: 10.4.9 (Build 8P135)
Report Version: 4
Command: firefox-bin
Path: /Applications/Firefox.app/Contents/MacOS/firefox-bin
Parent: WindowServer [59]
Version: 1.5 (1.5)
PID: 947
Thread: 0
Exception: EXC_BAD_ACCESS (0x0001)
Codes: KERN_PROTECTION_FAILURE (0x0002) at 0x0186af20

Thread 0 Crashed:
    0  libSystem.B.dylib 0x0010d50  vfprintf_140

Please describe what you were doing when the problem happened:

Your report will help Apple improve this software. Your personal information is not sent with this report. You will not be contacted in response to this report. For Apple product support, visit www.apple.com/support or contact your Apple dealer.

Send to Apple...
Problem Report for Firefox

Problem and system information:
Date/Time: 2007-05-16 16:00:01.424 -0400
OS Version: 10.4.9 (Build 8P135)
Report Version: 4

Command: firefox-bin
Path: /Applications/Firefox.app/Contents/MacOS/firefox-bin
Parent: WindowServer [59]

Version: 1.5 (1.5)
PID: 947
Thread: 0

Exception: EXC_BAD_ACCESS (0x0001)
Codes: KERN_PROTECTION_FAILURE (0x0002) at 0x0186af20

Thread 0Crashed:

Please describe what you were doing when the problem happened:

Your report will help Apple improve this software. Your personal information is not sent with this report. You will not be contacted in response to this report. For Apple product support, visit www.apple.com/support or contact your Apple dealer.
Problem and system information:

- Date/Time: 2007-05-16 16:00:01.424 -0400
- OS Version: 10.4.9 (Build 8P135)
- Report Version: 4

- Command: firefox-bin
- Path: /Applications/Firefox.app/Contents/MacOS/firefox-bin
- Parent: WindowServer [59]

- Version: 1.5 (1.5)
- PID: 947
- Thread: 0

Exception: EXC_BAD_ACCESS (0x0001)
Codes: KERN_PROTECTION_FAILURE (0x0002) at 0x0186af20

Thread 0 Crashed:

Please describe what you were doing when the problem happened:

Your report will help Apple improve this software. Your personal information is not sent with this report. You will not be contacted in response to this report. For Apple product support, visit www.apple.com/support or contact your Apple dealer.
Our solution
Our solution

Record
Our solution

- Record
- Replay
Our solution

- Record
- Replay
- Minimize
Our solution

- Record
- Replay
- Minimize
- Debug
Usage Scenario

In house

- Develop
- Replay / Debug

In the field

- Record (on line)
- Replay / Minimize (off line)

Execution repository
Existing record / replay approaches

**Deterministic debugging**
(e.g. Chen et al. 01, King et al. 05, Narayanasamy et al. 05, Netzer and Weaver 94, Srinivasan et al. 04, VMWare)

- Replay an entire execution by recording every component of an application

**Regression testing**
(e.g. Elbaum et al. 06, Orso et al. 06, Orso and Kennedy 05, Saff et al. 05, Mercury WinRunner)

- Replay only a portion of an execution by recording events for specific subsystems

Both types of techniques are not amenable to minimization and may cause unacceptable overhead.
Outline

✓ Motivation & background

• Our technique
  • record
  • replay
  • minimization

• Empirical evaluation

• Conclusion & future work
• Goal: develop an approach that has low overhead and is amenable to minimization

• Key insight: avoid focusing on low-level (internal) events
  • expensive (large number of events)
  • not amenable to minimization (high interdependence)
• Goal: develop an approach that has low overhead and is amenable to minimization

• Key insight: avoid focusing on low-level (internal) events
  • expensive (large number of events)
  • not amenable to minimization (high interdependence)

Focus on high-level (external) interactions with the environment
  • efficient (fewer, more “expensive” interactions)
  • amenable to minimization (low interdependence)
Environment interactions
Environment interactions

Streams
Environment interactions

Streams

Files
Environment interactions

Streams

Files
Environment interactions

Interaction events:

- **FILE** — interaction with a file
- **POLL** — checks for availability of data on a stream
- **PULL** — read data from a stream
Event log:

Environment data (files):

Environment data (streams):
Event log:
FILE foo.1

Environment data (streams):

Environment data (files):
Event log:
FILE foo.1

Environment data (streams):

Environment data (files):
Event log:
FILE foo.1
POLL KEYBOARD NOK

Environment data (files):

Environment data (streams):
Event log:

FILE foo.1
POLL KEYBOARD NOK

Environment data (files):

Environment data (streams):
Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK

Environment data (streams):
KEYBOARD: {5680}

Environment data (files):

Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK

Environment data (streams):
KEYBOARD: {5680}

Environment data (files):
Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 5

Environment data (streams):
KEYBOARD: {5680}hello

Environment data (files):
FILE foo.1
Event log:

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 5

Environment data (streams):
KEYBOARD: {5680}hello

Environment data (files):
Event log:

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 5
POLL NETWORK OK

Environment data (streams):
KEYBOARD: \{5680\}hello
NETWORK: \{3405\}

Environment data (files):
Event log:

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 5
POLL NETWORK OK

Environment data (streams):

KEYBOARD: {5680}hello

NETWORK: {3405}

Environment data (files):

FILE foo.1
Event log:

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 5
POLL NETWORK OK

Environment data (streams):
KEYBOARD: {5680}hello
NETWORK: {3405}

Environment data (files):
FILE foo.1
Environment data:

**Event log:**

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2

... PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

**Environment data (streams):**

KEYBOARD: {5680}hello  l {4056}c  l {300}...

NETWORK: {3405}<html><body>...  l {202}...

**Environment data (files):**

foo.1  foo.2  bar.1  ...

foo.1
foo.2
bar.1
**Event log:**

FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
...
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

**Environment data (streams):**

KEYBOARD: {5680}hello l {4056}c l {300}...

NETWORK: {3405}<html><body>... l {202}...

**Environment data (files):**

foo.1
foo.2
bar.1
Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
...
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

Environment data (streams):
KEYBOARD: {5680}hello I {4056}c I {300}...
NETWORK: {3405}<html><body>... I {202}...

Environment data (files):
foo.1
foo.2
bar.1
...
Event log:
FILE foo.1  ✔
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
...
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

Environment data (streams):
KEYBOARD: {5680}hello  l {4056}c  l {300}...
NETWORK: {3405}<html><body>...  l {202}...

Environment data (files):
foo.1  
foo.2  
bar.1  
...
Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
...
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

Environment data (streams):
KEYBOARD: {5680}hello l {4056}c l {300}...
NETWORK: {3405}<html><body>... l {202}...

Environment data (files):
foo.1
foo.2
bar.1
...
Event log:
FILE foo.1 ✔
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
...
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK
...

Environment data (streams):
KEYBOARD: {5680}hello  \{4056}c  \{300}...
NETWORK: {3405}<html><body>...  \{202}...

Environment data (files):
...
Event log:
FILE foo.1 ✔
POLL KEYBOARD NOK ✔
POLL KEYBOARD OK ✔
PULL KEYBOARD 1 ✔
POLL NETWORK OK ✔
PULL NETWORK 1024 ✔
FILE bar.1 ✔
POLL NETWORK NOK ✔
POLL NETWORK OK ✔
FILE foo.2 ✔
...
PULL NETWORK 1024 ✔
FILE foo.2 ✔
POLL KEYBOARD NOK ✔
...

Environment data (streams):
KEYBOARD: {5680}hello {4056}c {300}...
NETWORK: {3405}<html><body>... {202}...

Environment data (files):
foo.1
foo.2
bar.1
...
Minimize

Goal: focus debugging effort
Minimize

Goal: focus debugging effort
Minimize

Goal: focus debugging effort

Execution recording

Time minimization
Minimize

Goal: focus debugging effort
Minimize

Goal: focus debugging effort
Minimize

Goal: focus debugging effort

Execution recording

Time minimization

Execution recording

Data minimization

Execution recording
Minimize: time

**Event log:**
- FILE foo.1
- POLL KEYBOARD NOK
- POLL KEYBOARD OK
- PULL KEYBOARD 1
- POLL NETWORK OK
- PULL NETWORK 1024
- FILE bar.1
- POLL NETWORK NOK
- POLL NETWORK OK
- FILE foo.2
- PULL NETWORK 1024
- FILE foo.2
- POLL KEYBOARD NOK

**Environment data (streams):**
- KEYBOARD: \{5680\}hello \{4056\}c \{300\}...
- NETWORK: \{3405\}<html><body>... \{202\}...

**Environment data (files):**
Minimize: time

Event log:
FILE foo.1
POLL KEYBOARD NOK
POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1
POLL NETWORK NOK
POLL NETWORK OK
FILE foo.2
PULL NETWORK 1024
FILE foo.2
POLL KEYBOARD NOK

Environment data (streams):
KEYBOARD: {5680}hello 1 {4056}c 1 {300}...
NETWORK: {3405}<html><body>... 1 {202}...

Environment data (files):
Minimize: time

Event log:

FILE foo.1

POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1

POLL NETWORK OK
FILE foo.2
PULL NETWORK 1024
FILE foo.2

Environment data (streams):
KEYBOARD: {5680}hello {4056}c {300}...
NETWORK: {3405}<html><body>... {202}...

Environment data (files):
Minimize: time

**Event log:**

FILE foo.1

POLL KEYBOARD OK
PULL KEYBOARD 1
POLL NETWORK OK
PULL NETWORK 1024
FILE bar.1

POLL NETWORK OK
FILE foo.2
PULL NETWORK 1024
FILE foo.2

**Environment data (streams):**

KEYBOARD: {5000}hello {1050}c {500}...

NETWORK: {5105}<html><body>...{202}...

**Environment data (files):**
Minimize: data

Data minimization
- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

Data minimization

Whole entities
Chunks
Atoms

Environment
Minimize: data

Data minimization

Whole entities

Chunks

Atoms

Environment

Atoms

Chunks

Whole entities

Entities
Minimize: data

Data minimization

Whole entities
Chunks
Atoms

Environment
Minimize: data

Data minimization

- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

- Data minimization
  - Whole entities: X
  - Chunks
  - Atoms

- Environment
Minimize: data

Data minimization
- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

Data minimization

Whole entities

Chunks

Atoms

Environment
Minimize: data

Data minimization

- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

Data minimization
- Whole entities
- Chunks
- Atoms

Environment

Minimize: data

Data minimization
- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

Data minimization

Whole entities

Chunks

Atoms

Environment
Minimize: data

Data minimization

- Whole entities
- Chunks
- Atoms

Environment
Minimize: data

Data minimization

Whole entities

Chunks

Atoms

Environment
The tool: ADDA

Assisting the Debugging of Deployed Applications

- Record and Replay:
  - Works on x86 (c-lib based) binaries
  - Based on dynamic instrumentation (Pin)
  - Maps c-library calls to interaction events

- Minimization:
  - Set of extensible scripts
Limitations

Two main limitations:

• **Technique:**
  May not replay non-deterministic failures

• **Implementation:**
  Does not handle window system events (yet)
Empirical evaluation

- Research questions
  - Can ADDA produce *minimized executions* that can be used to debug the original failure?
  - How much *overhead* does ADDA impose?

- Subject:
  - Pine — widely-used email / news client

- Data:
  - Two real field failures from Pine’s history
  - Set of 20 failing executions, 10 per failure
Empirical evaluation

- Research questions
  - Can ADDA produce minimized executions that can be used to debug the original failure?
  - How much overhead does ADDA impose?

- Subject:
  - Pine — widely-used email / news client

- Data:
  - Two real field failures from Pine’s history
  - Set of 20 failing executions, 10 per failure
Minimization results

Average value after minimization

- # entities
- streams size
- files size

- Header-color fault
- Address book fault
Minimization results

Moreover, these results are conservative: recorded executions only contain the minimal amount of data needed to perform an action.
Minimization results

Overhead

- Offline: less than 75 minutes for minimization
- Online: negligible overhead while recording

Moreover, these results are conservative: recorded executions only contain the minimal amount of data needed to perform an action.
Specific Example: Address Book Failure

- Complete execution
  - 34 entities (files and streams)
  - ≈800kb
- Minimized execution
  - 5 partial entities (4 files, 1 stream)
  - ≈72kb
Future work

- More studies: additional applications and real users
- Extend technique / implementation
  - Support windowing system
  - Investigate ad-hoc minimization algorithms
- Include non-deterministic events (if needed)
Conclusions

• Novel approach that supports debugging field failures

• Prototype implementation for x86 binaries

• Preliminary empirical evaluation: for the cases considered, our technique can
  1. minimize failing executions,
  2. preserve their failing behavior, and
  3. impose low overhead on users
Questions?