CAT Detect: A Tool for Detecting Inconsistency in Computer Activity Timelines

By

Andrew Marrington, Ibrahim Baggili, George Mohay and Andrew Clark

Presented At

The Digital Forensic Research Conference

DFRWS 2011 USA New Orleans, LA (Aug 1st - 3rd)

CAT Detect: A tool for detecting inconsistency in computer activity timelines

Andrew Marrington, Ibrahim Baggili
College of Information Technology, Zayed University
George Mohay, Andrew Clark
Information Security Institute, Queensland University of Technology

Motivation

- Construction of timelines of computer activity is important to many investigations.
  - From system logs, file MAC times, file metadata
- Whether through deliberate tampering, software misconfiguration/bug, hardware clock skew/drift or other “natural” cause, these timestamps may be inconsistent or contradictory.

Computer Activity Timelines

- A computer activity timeline is a sequence of *events* ordered by time.
- Events have two sources:
  - System logs (e.g. the Windows Event Logs) – we call these *recorded events*
  - Timestamps from the file system and metadata (e.g. MAC times, Exif creation times) – we call these *inferred events*
- A complete-as-possible timeline of computer activity combines recorded and inferred events, ordered by time.

Inconsistent Timelines

- Out-of-sequence events
  - Events whose timestamp is inaccurate
  - These events appear to have taken place after other events, even though logically they should have taken place beforehand
- Missing events
  - Events which logically must have taken place but…
  - …are absent from our timeline!

Sources of Inconsistency

- Out-of-sequence events and missing events can be caused through “normal” operation or deliberate tampering or obfuscation.
- “Natural” causes include:
  - System logging configuration – by default many events may go unrecorded, making them “missing”
  - Clock skew and drift, and the variations in clock skew over time
  - Laptops moving between timezones etc

Detecting Inconsistency

- Approach

Tool Environment

- System log parsers
- Event inferrer

- RecordedEvents table
- InferredEvents table

- Rules base

Events

- Events are extracted from system logs (e.g. Windows Event Logs) using a log parsing tool.
  - We have been using our own parser, but other tools are now available with a great deal more maturity, e.g. GrokEVT
- Events are inferred from timestamps extracted from the file system and file metadata.
  - These could extracted from other tools.
  - We have been using our own event inferrer, which we hope to make available soon.

Normalised format

- Events are normalised into the following format:

  Time, Subject, Object, Result, Action.

- With the addition of an “Event ID” field, these are stored in two database tables: *RecordedEvents* and *InferredEvents*.

Rules Base

- The rules base includes some template events
  - e.g. template login and logoff events
- The rules base also includes rules which define the relations which should exist between certain events in order for them to be consistent
- CAT Detect checks the events in a given timeline against the rules base.
- At the moment, these rules are statically defined in-code but this should be improved upon soon.

Happened before relation…

- Some types of events must logically *happen before* other events.
- For example, the user must login before the user can create a file.
- This sort of relation is described by Leslie Lamport (1978) as the *happened-before* relation.
  - Expanded by Fidge (1991) and used in a DF context by Gladyshev and Patel (2005) and Willassen (2008).
- This relation is used as the basis of our rules.

Detecting out-of-sequence events

- A user *x* must login to the computer system before creating file *y* on the computer system's drive. Each of these is an event:
  - evtA = (t_a, *x*, system, login, success)
  - evtB = (t_b, *x*, *y*, created, success)
- The rule will specify that evtA must *happen-before* evtB (evtA → evtB).
- If $t_a < t_b$, then evtA and evtB are inconsistent.
- If $t_a > t_b$, then the events break this rule and are inconsistent.

Preconditions and Missing Events

- In some relations, the presence of the second event necessarily implies the presence of the first event.
- Extending our last example, we can say that evtA is a precondition of evtB.
- If evtA does not exist in our database, therefore, it is a *missing event* whose presence has been inferred.

Why do we need both?

- Let us define a third event, whereby a user *x* logs off the computer system:
  - evtC = (t_c, *x*, system, logoff, success)
- It is clear that evtB → evtC, and that evtA is a precondition of evtC.
- It is also clear that evtB (a file creation event) is not necessary for evtC to take place.

CAT Detect

- Usage Experiment

Experiment

- We extracted our data from a Windows XP test PC with several dummy user accounts created on it:
  - baddie
  - crook
  - nefarious
- We normalised the event logs and loaded them into the *RecordedEvents* table.
- We ran our event inferrer over the disk image and loaded its output into the *InferredEvents* table.

Experimental Timelines

- Timeline A – consistent timeline, no tampering
- Timeline B – inconsistent timeline, “crook” logged in but files apparently created by “baddie”
- Timeline C – inconsistent timeline, missing logon event
- Timeline D – inconsistent timeline, timestamp of logoff event altered.

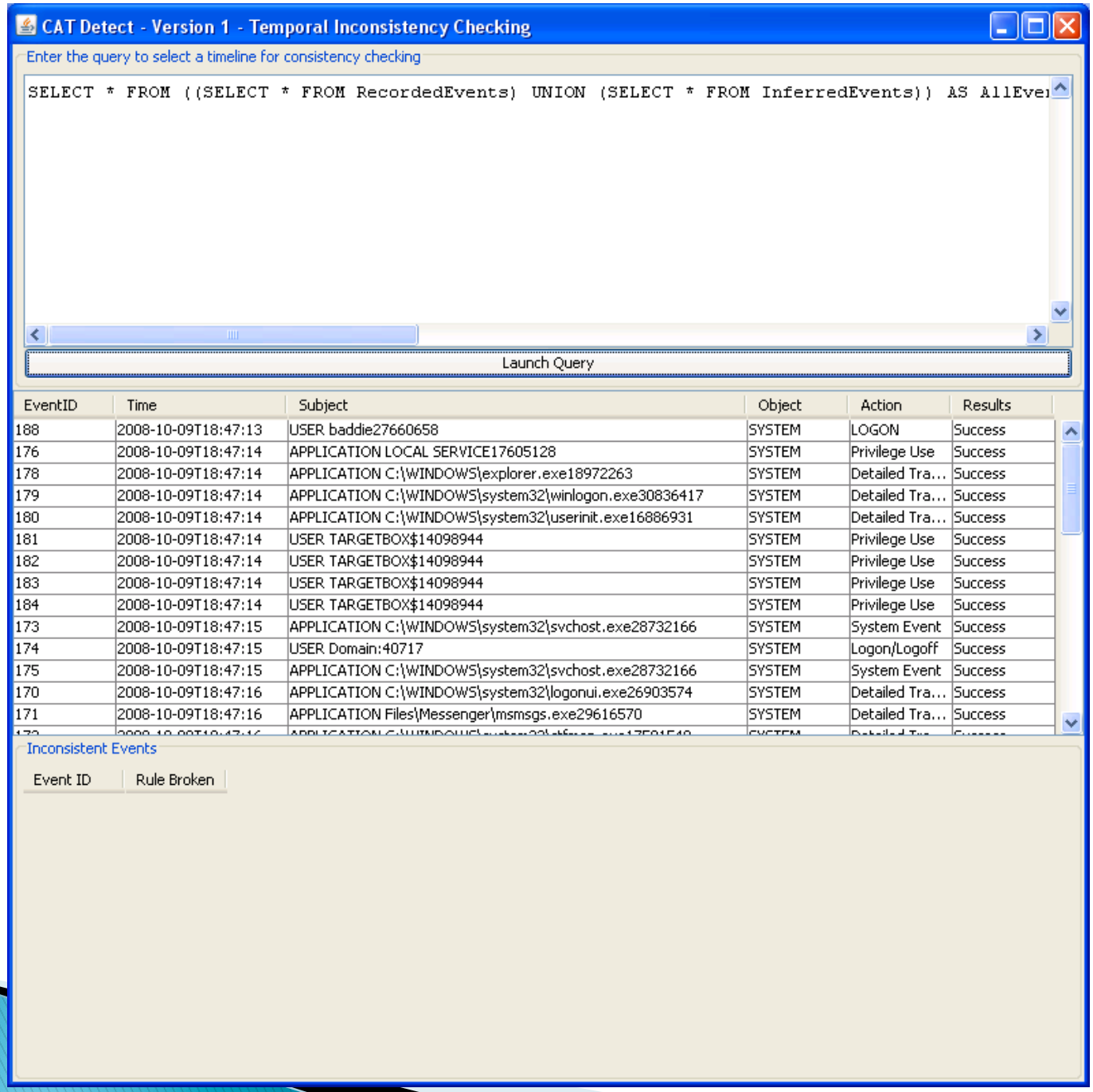
Limitations

- This release of CAT Detect has several limitations
- Most notably, as you will see, the user must provide a query to specify the timeline corresponding to the user login session in which they are interested.

Timeline A

- Query:

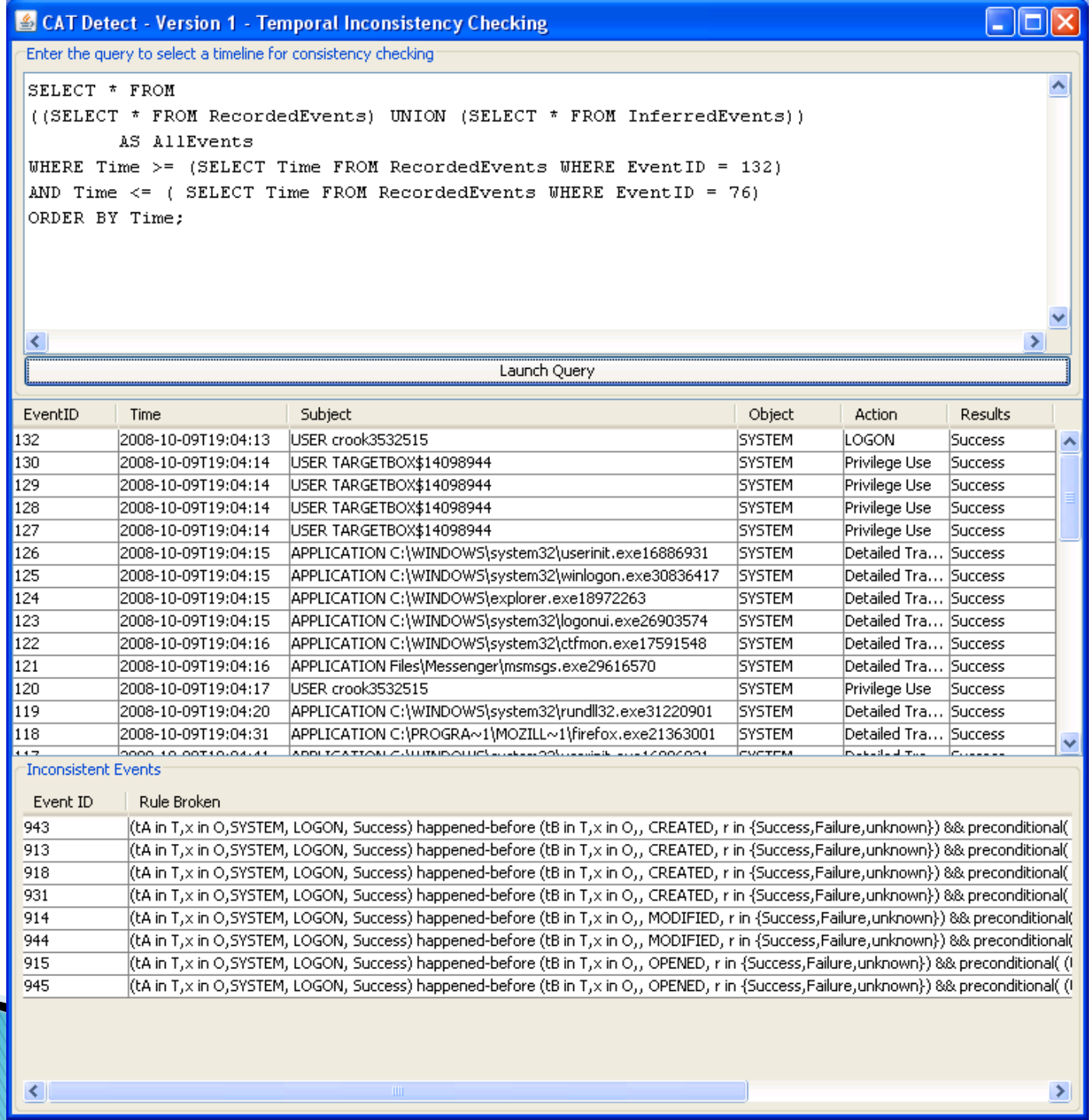
```
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION
  (SELECT * FROM InferredEvents))
  AS AllEvents
WHERE Time >= (SELECT Time FROM
  RecordedEvents WHERE EventID = 188)
AND Time <= ( SELECT Time FROM
  RecordedEvents WHERE EventID = 146)
  ORDER BY Time;
```


CAT Detect - Version 1 - Temporal Inconsistency Checking
Enter the query to select a timeline for consistency checking
SELECT * FROM ((SELECT * FROM RecordedEvents) UNION (SELECT * FROM InferredEvents)) AS AllEve
Launch Query
EventID
Time
Subject
Object
Action
Results
188 2008-10-09T18:47:13 USER baddie27660658 SYSTEM LOGON Success
176 2008-10-09T18:47:14 APPLICATION LOCAL SERVICE17605128 SYSTEM Privilege Use Success
178 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\explorer.exe18972263 SYSTEM Detailed Tra... Success
179 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\system32\winlogon.exe30836417 SYSTEM Detailed Tra... Success
180 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\system32\userinit.exe16886931 SYSTEM Detailed Tra... Success
181 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
182 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
183 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
184 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
173 2008-10-09T18:47:15 APPLICATION C:\WINDOWS\system32\svchost.exe28732166 SYSTEM System Event Success
174 2008-10-09T18:47:15 USER Domain:40717 SYSTEM Logon/Logoff Success
175 2008-10-09T18:47:15 APPLICATION C:\WINDOWS\system32\svchost.exe28732166 SYSTEM System Event Success
170 2008-10-09T18:47:16 APPLICATION C:\WINDOWS\system32\logonui.exe26903574 SYSTEM Detailed Tra... Success
171 2008-10-09T18:47:16 APPLICATION Files\Messenger\msmsgs.exe29616570 SYSTEM Detailed Tra... Success
Inconsistent Events
Event ID
Rule Broken

Timeline B

- Query:

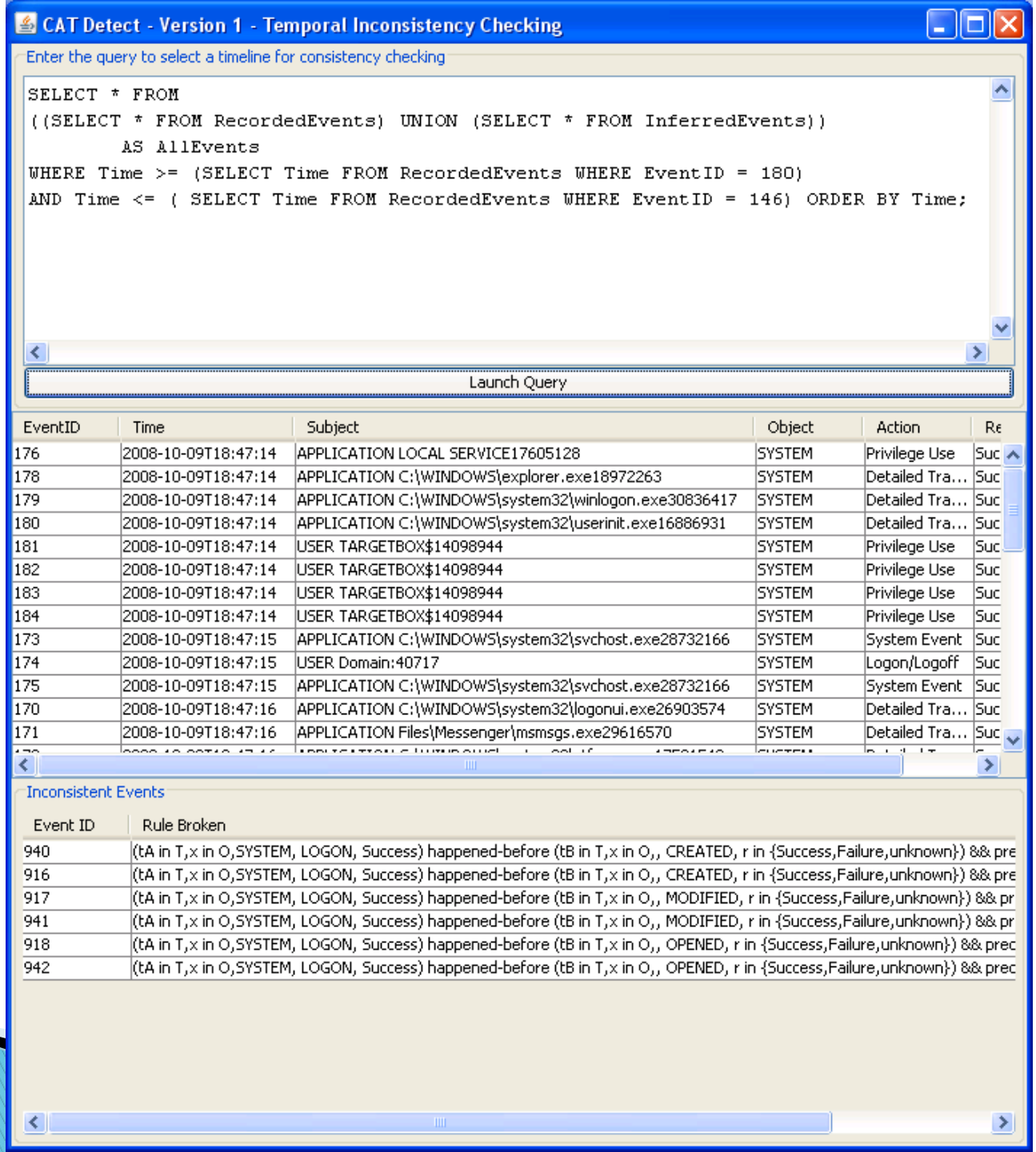
```
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION
  (SELECT * FROM InferredEvents))
  AS AllEvents
WHERE Time >= (SELECT Time FROM
  RecordedEvents WHERE EventID = 132)
AND Time <= ( SELECT Time FROM
  RecordedEvents WHERE EventID = 76)
ORDER BY Time;
```


CAT Detect - Version 1 - Temporal Inconsistency Checking
Enter the query to select a timeline for consistency checking
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION (SELECT * FROM InferredEvents))
AS AllEvents
WHERE Time >= (SELECT Time FROM RecordedEvents WHERE EventID = 132)
AND Time <= (SELECT Time FROM RecordedEvents WHERE EventID = 76)
ORDER BY Time;
Launch Query
EventID Time Subject Object Action Results
132 2008-10-09T19:04:13 USER crook3532515 SYSTEM LOGON Success
130 2008-10-09T19:04:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
129 2008-10-09T19:04:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
128 2008-10-09T19:04:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
127 2008-10-09T19:04:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Success
126 2008-10-09T19:04:15 APPLICATION C:\WINDOWS\system32\userinit.exe16886931 SYSTEM Detailed Tra... Success
125 2008-10-09T19:04:15 APPLICATION C:\WINDOWS\system32\winlogon.exe30836417 SYSTEM Detailed Tra... Success
124 2008-10-09T19:04:15 APPLICATION C:\WINDOWS\explorer.exe18972263 SYSTEM Detailed Tra... Success
123 2008-10-09T19:04:15 APPLICATION C:\WINDOWS\system32\logonui.exe26903574 SYSTEM Detailed Tra... Success
122 2008-10-09T19:04:16 APPLICATION C:\WINDOWS\system32\ctfmon.exe17591548 SYSTEM Detailed Tra... Success
121 2008-10-09T19:04:16 APPLICATION Files\Messenger\msmsgs.exe29616570 SYSTEM Detailed Tra... Success
120 2008-10-09T19:04:17 USER crook3532515 SYSTEM Privilege Use Success
119 2008-10-09T19:04:20 APPLICATION C:\WINDOWS\system32\rundll32.exe31220901 SYSTEM Detailed Tra... Success
118 2008-10-09T19:04:31 APPLICATION C:\PROGRA~1\MOZILL~1\firefox.exe21363001 SYSTEM Detailed Tra... Success
Inconsistent Events
Event ID Rule Broken
943 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && preconditional(
913 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && preconditional(
918 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && preconditional(
931 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && preconditional(
914 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, MODIFIED, r in {Success,Failure,unknown}) && preconditional(
944 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, MODIFIED, r in {Success,Failure,unknown}) && preconditional(
915 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, OPENED, r in {Success,Failure,unknown}) && preconditional((
945 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, OPENED, r in {Success,Failure,unknown}) && preconditional((

Timeline C

- Query:

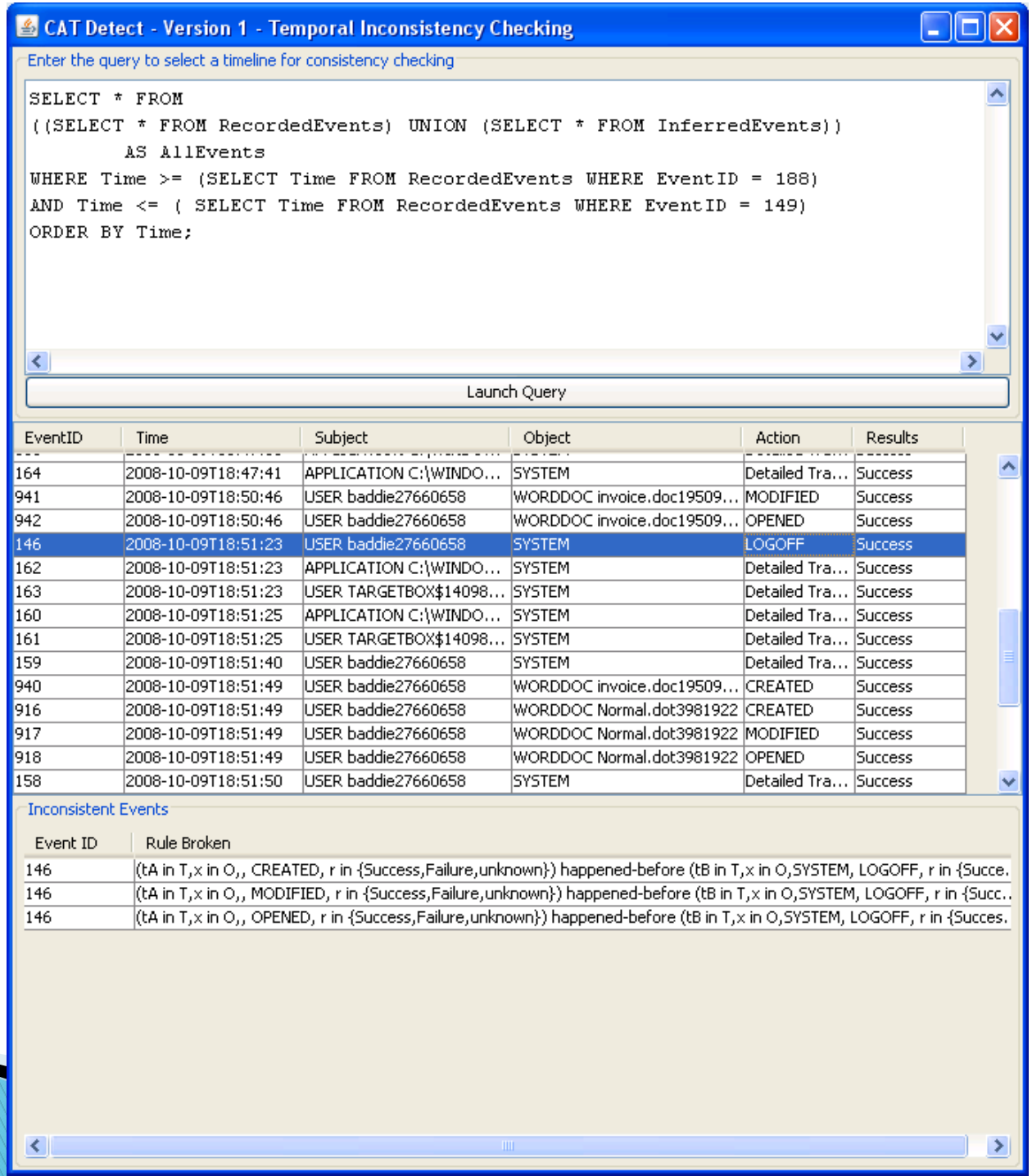
```
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION
  (SELECT * FROM InferredEvents))
  AS AllEvents
WHERE Time >= (SELECT Time FROM
  RecordedEvents WHERE EventID = 180)
AND Time <= ( SELECT Time FROM
  RecordedEvents WHERE EventID = 146)
  ORDER BY Time;
```


CAT Detect - Version 1 - Temporal Inconsistency Checking
Enter the query to select a timeline for consistency checking
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION (SELECT * FROM InferredEvents))
AS AllEvents
WHERE Time >= (SELECT Time FROM RecordedEvents WHERE EventID = 180)
AND Time <= (SELECT Time FROM RecordedEvents WHERE EventID = 146) ORDER BY Time;
Launch Query
EventID Time Subject Object Action Re
176 2008-10-09T18:47:14 APPLICATION LOCAL SERVICE17605128 SYSTEM Privilege Use Suc
178 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\explorer.exe18972263 SYSTEM Detailed Tra... Suc
179 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\system32\winlogon.exe30836417 SYSTEM Detailed Tra... Suc
180 2008-10-09T18:47:14 APPLICATION C:\WINDOWS\system32\userinit.exe16886931 SYSTEM Detailed Tra... Suc
181 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Suc
182 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Suc
183 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Suc
184 2008-10-09T18:47:14 USER TARGETBOX$14098944 SYSTEM Privilege Use Suc
173 2008-10-09T18:47:15 APPLICATION C:\WINDOWS\system32\svchost.exe28732166 SYSTEM System Event Suc
174 2008-10-09T18:47:15 USER Domain:40717 SYSTEM Logon/Logoff Suc
175 2008-10-09T18:47:15 APPLICATION C:\WINDOWS\system32\svchost.exe28732166 SYSTEM System Event Suc
170 2008-10-09T18:47:16 APPLICATION C:\WINDOWS\system32\logonui.exe26903574 SYSTEM Detailed Tra... Suc
171 2008-10-09T18:47:16 APPLICATION Files\Messenger\msmsgs.exe29616570 SYSTEM Detailed Tra... Suc
Inconsistent Events
Event ID Rule Broken
940 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && pre
916 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, CREATED, r in {Success,Failure,unknown}) && pre
917 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, MODIFIED, r in {Success,Failure,unknown}) && pr
941 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, MODIFIED, r in {Success,Failure,unknown}) && pr
918 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, OPENED, r in {Success,Failure,unknown}) && prec
942 (tA in T,x in O,SYSTEM, LOGON, Success) happened-before (tB in T,x in O,, OPENED, r in {Success,Failure,unknown}) && prec

Timeline D

- Query:

```
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION
  (SELECT * FROM InferredEvents))
  AS AllEvents
WHERE Time >= (SELECT Time FROM
  RecordedEvents WHERE EventID = 188)
AND Time <= ( SELECT Time FROM
  RecordedEvents WHERE EventID = 149)
ORDER BY Time;
```


CAT Detect - Version 1 - Temporal Inconsistency Checking
Enter the query to select a timeline for consistency checking
SELECT * FROM
((SELECT * FROM RecordedEvents) UNION (SELECT * FROM InferredEvents))
AS AllEvents
WHERE Time >= (SELECT Time FROM RecordedEvents WHERE EventID = 188)
AND Time <= (SELECT Time FROM RecordedEvents WHERE EventID = 149)
ORDER BY Time;
Launch Query
EventID Time Subject Object Action Results
164 2008-10-09T18:47:41 APPLICATION C:\WINDO... SYSTEM Detailed Tra... Success
941 2008-10-09T18:50:46 USER baddie27660658 WORDDOC invoice.doc19509... MODIFIED Success
942 2008-10-09T18:50:46 USER baddie27660658 WORDDOC invoice.doc19509... OPENED Success
146 2008-10-09T18:51:23 USER baddie27660658 SYSTEM LOGOFF Success
162 2008-10-09T18:51:23 APPLICATION C:\WINDO... SYSTEM Detailed Tra... Success
163 2008-10-09T18:51:23 USER TARGETBOX$14098... SYSTEM Detailed Tra... Success
160 2008-10-09T18:51:25 APPLICATION C:\WINDO... SYSTEM Detailed Tra... Success
161 2008-10-09T18:51:25 USER TARGETBOX$14098... SYSTEM Detailed Tra... Success
159 2008-10-09T18:51:40 USER baddie27660658 SYSTEM Detailed Tra... Success
940 2008-10-09T18:51:49 USER baddie27660658 WORDDOC invoice.doc19509... CREATED Success
916 2008-10-09T18:51:49 USER baddie27660658 WORDDOC Normal.dot3981922 CREATED Success
917 2008-10-09T18:51:49 USER baddie27660658 WORDDOC Normal.dot3981922 MODIFIED Success
918 2008-10-09T18:51:49 USER baddie27660658 WORDDOC Normal.dot3981922 OPENED Success
158 2008-10-09T18:51:50 USER baddie27660658 SYSTEM Detailed Tra... Success
Inconsistent Events
Event ID Rule Broken
146 (tA in T,x in O,, CREATED, r in {Success,Failure,unknown}) happened-before (tB in T,x in O,SYSTEM, LOGOFF, r in {Succe.
146 (tA in T,x in O,, MODIFIED, r in {Success,Failure,unknown}) happened-before (tB in T,x in O,SYSTEM, LOGOFF, r in {Succ..
146 (tA in T,x in O,, OPENED, r in {Success,Failure,unknown}) happened-before (tB in T,x in O,SYSTEM, LOGOFF, r in {Succes.

CAT Detect

- DFRWS Release

CAT Detect

- The CAT Detect DFRWS release is available to download from SourceForge now:

http://sourceforge.net/projects/catdetect/files/

- You will need Java 6 and MySQL 5.5 or later to run CAT Detect.
- CAT Detect's source code is available under the Academic Free License.

Disclaimer

- This is not final, production-quality software.
- We have not released CAT Detect because it is perfect – but rather, because it is far from perfect.
  - A to do list will be posted on the SourceForge list soon!
- If you are interested in contributing your ideas and effort to developing tools to detect inconsistencies in timelines, any input would be welcome.

Future Work

- Dynamic rules specification.
- Better parser integration (e.g. built-in Windows Event Log parser or close integration/front-end for GrokEVT or output).
- Automatic detection of user sessions.
- Obtain data sources and parsers to apply CAT Detect to non-Windows operating systems and newer versions of Windows.
- Fix some “unfortunate” user interface issues!