Definitive Diagnostic Data

http://www.skendric.com/seminar/

Stuart Kendrick
Sustaining Engineer
EMC Isilon
Definitive Diagnostic Data

A Rapid Problem Resolution® perspective

Advance7 is a consulting outfit which helps customers resolve critical Problems – they put an analyst at your site to coordinate your staff plus vendors to fix the issue, using the RPR methodology.

I don’t work for Advance7, and I have only a rudimentary grasp of RPR – I’ve read Paul Offord’s book and attended Advance7’s two-day Foundations in RPR seminar.

On the other hand, I have employed bits & pieces of RPR on the job, and I’ve found it effective – in fact, any time I get near a trouble-shooting job, I try to employ as much of RPR as I can manage.

A signature feature of RPR is its concept of Definitive Diagnostic Data (D³).

Over the next hour+, I plan to sketch my understanding of D³, focused particularly on the concrete technique of markers, which thread their way through D³.
Mechanics

Talk

• I encourage interactivity
• If you want to contribute, feel free to interrupt me
• Or raise your hand, and I’ll call on you
• I’m good with either approach

This deck visible at http://www.skendric.com/seminar/
Me

Multi-disciplinary IT trouble-shooter / Root Cause Analysis

http://www.skendric.com

sbk@cornella
stuart@cpvax5 (Science Applications Inc)
sbk@cornellc.cit.cornell.edu
stuart.kendrick@med.cornell.edu
skendric@fhcrc.org
stuart.kendrick@isi.loni.dot.com

student 1981
programmer 1984
desktop / server 1985
server / network 1991
multidisciplinary 1993
sustaining engineer 2013

IT Architect | ITIL Problem Manager | Problem Analyst | Device Monitoring | Transport

Geeky Highlights
PL/1 on IBM mainframes
FORTRAN on CRAY-1
Terak, DisplayWriter, IBM PC, Macintosh
Netware, Corvus Omninert, TCP-IP / IPX / AppleTalk
AppleShare, QuickMail, Farallon, NRC, Cisco, Sniffers
Solaris, Windows, Linux, Perl, SNMP, Wireshark, Cisco, Fluke
OneFS

Cornell University Ithaca 1981
SAIC San Diego 1984
Cornell University Ithaca 1985
Cornell University Ithaca 1988
Cornell Medical College Manhattan 1991
FHCRC Seattle 1993
EMC Isilon Seattle 2013

Geek credentials: I missed punch-cards by one semester ... grew up on shared machines (IBM and Cray) ... my first network ran at 1Mb/s over Cat 2 (Corvus Omninert) carrying IPX + AppleTalk with IP encapsulated in both. I bored a vampire tap (once) ... my first analyzer was a Network General Toshiba 286 laptop ... and alpha versions of EtherPeek
Recruiting

I attend SharkFest for a lot of reasons …

But one of them is recruiting.

Isilon
If you would like to hear what it is like to work at Isilon, I would enjoy sharing the pros, and the cons, of working in this space.

You may not be interested in changing jobs right now – from my point of view, I would still enjoy talking with you – perhaps your situation will change in a year or two. *Isilon invests long-term in staff; a multi-year courtship suits our style just fine.*

Richly complex product, engineering-oriented company, plenty of difficult problems to solve. Global company, numerous locations, and once you’re sufficiently senior, plenty of flexibility in terms of operating remotely, telecommuting, and visiting a base office every quarter or so.

FHcrc
My old position at the Hutch is still open … *Problem Manager / Problem Analyst / emcee of RCAs, with oversight over Change Management and post-mortems arising from Incidents. Also, Network Manager (four techs, physical layer, Ethernet/IP/WiFi transport, firewall operations, Internet connectivity, voice.)*

*Come find me during a break or in the evening. Professional networking is a good thing.*
So What Is *Definitive Diagnostic Data*?

At a first pass, D³ means inserting *markers* into the data stream you are capturing

- Markers help you find the section of trace where some interesting event occurred
- Markers function as in-band documentation on what happened and when
- Markers contribute to **concreteness**

  *I know event xyz occurred after this point here and not before*

  *Thus I know I can ignore this chunk of the trace and focus my attention on this other chunk*
Each DC runs Domain Controller and DNS services
No external forwarding
Automated drive mapping

echo off
REM This script maps drives to a cluster, unmaps them, sleeps, and then repeats
REM The idea is to trigger intermittent cluster accessibility problems and to
REM record the time of those events, to be correlated with data capture efforts
REM running in parallel outside this script
REM
REM V  Who       When        What
REM -------------------------------------------------------------
REM 1.2.0  skendric  2014-05-30  Record affinitized DC correctly
REM 1.0.1  skendric  2014-05-14  Record affinitzed DC to log file
REM 1.0.0  skendric  2014-04-29  First version

REM Generically useful startup stuff
setlocal
setlocal ENABLEDELAYEDEXPANSION

REM Assign local variables
set node1=10.25.35.100
set node2=10.25.35.101
set node3=10.25.35.102
set sleepLong=10
set sleepShort=5
set usage=usage: cycle-drive-mapping {user@domain} {password} [output file]

REM Locate binaries
set findCmd=C:\Windows\System32\find.exe
set klistCmd=C:\Windows\System32\klist.exe
set netCmd=C:\Windows\System32\net.exe
set nltestCmd=C:\Windows\System32\nltest.exe /sc_query:safra.com.br
set sleepCmd=c:\temp\sleep.exe
set wteeCmd=c:\temp\wtee.exe -a
set klistCmd=c:\windows\system32\klist.exe
Automated drive mapping

REM Grab command-line parameters
set user=%1
set password=%2
set output=%3

REM Sanity check
if not defined user (  
    echo Must specify user  
    echo %usage%  
    exit /B 1
)
if not defined password (  
    echo Must specify password  
    echo %usage%  
    exit /B 1
)

REM Assign defaults
if not defined output set output=c:\temp\cycle-drive-mapping.txt
Automated drive mapping

REM Loop forever
:BEGIN
echo. 2>&1 | %wteeCmd% %output%
echo. 2>&1 | %wteeCmd% %output%
echo. 2>&1 | %wteeCmd% %output%
echo. 2>&1 | %output%
echo. 2>&1 | %date% %time%
echo Affinitized DC:
%nltestCmd% | %findCmd% "Trusted" 2>&1 | %wteeCmd% %output%
echo Purging Kerberos ticket 2>&1 | %wteeCmd% %output%
%klistCmd% purge
echo Mapping drives 2>&1 | %wteeCmd% %output%
echo Mapping x: to %node1% at %date% %time% 2>&1 | %wteeCmd% %output%
%netCmd% use x: \\%node1%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%
echo Mapping y: to %node2% at %date% %time% 2>&1 | %wteeCmd% %output%
%netCmd% use y: \\%node2%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%
echo Mapping z: to %node3% at %date% %time% 2>&1 | %wteeCmd% %output%
%netCmd% use z: \\%node3%\ifs /user:%user% %password% 2>&1 | %wteeCmd% %output%
echo Sleeping for %sleepShort% 2>&1 | %wteeCmd% %output%
%sleepCmd% %sleepShort% 2>&1 | %wteeCmd% %output%
echo Deleting mappings at %date% %time% 2>&1 | %wteeCmd% %output%
%netCmd% use /del x: 2>&1 | %wteeCmd% %output%
%netCmd% use /del y: 2>&1 | %wteeCmd% %output%
%netCmd% use /del z: 2>&1 | %wteeCmd% %output%
echo Sleeping for %sleepLong% 2>&1 | %wteeCmd% %output%
%sleepCmd% %sleepLong% 2>&1 | %wteeCmd% %output%
goto BEGIN
Skew clock on domain controller

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Bytes</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>432</td>
<td>0.0002039000</td>
<td>158</td>
<td>10.25.35.100</td>
<td>10.25.34.32</td>
<td>SMB</td>
<td>Trans2 RESPONSE, QUERY_FS_INFO</td>
</tr>
<tr>
<td>433</td>
<td>0.000418000</td>
<td>128</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Trans2 Request, QUERY_FSINFO, Query FS Attribute Info</td>
</tr>
<tr>
<td>434</td>
<td>0.000209000</td>
<td>134</td>
<td>10.25.35.100</td>
<td>10.25.34.52</td>
<td>SMB</td>
<td>Trans2 RESPONSE, QUERY_FSINFO</td>
</tr>
<tr>
<td>435</td>
<td>0.000669000</td>
<td>128</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Trans2 Request, QUERY_FSINFO, Query FS Attribute Info</td>
</tr>
<tr>
<td>436</td>
<td>0.000217000</td>
<td>134</td>
<td>10.25.35.100</td>
<td>10.25.34.52</td>
<td>SMB</td>
<td>Trans2 RESPONSE, QUERY_FSINFO</td>
</tr>
<tr>
<td>437</td>
<td>0.160387000</td>
<td>60</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>3635 &gt; 445 [ACK] Seq=1610 Ack=1316 win=62925 Len=0</td>
</tr>
<tr>
<td>438</td>
<td>4.476759000</td>
<td>97</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Logoff Andx Request</td>
</tr>
<tr>
<td>439</td>
<td>0.000279000</td>
<td>97</td>
<td>10.25.35.100</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Logoff Andx Response</td>
</tr>
<tr>
<td>440</td>
<td>0.000241000</td>
<td>93</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Tree Disconnect Request</td>
</tr>
<tr>
<td>441</td>
<td>0.000640000</td>
<td>93</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>SMB</td>
<td>Tree Disconnect Response</td>
</tr>
<tr>
<td>442</td>
<td>0.001592000</td>
<td>60</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>3635 &gt; 445 [FIN, ACK] Seq=1692 Ack=1398 win=62843 Len=0</td>
</tr>
<tr>
<td>443</td>
<td>0.000144000</td>
<td>54</td>
<td>10.25.35.100</td>
<td>10.25.34.52</td>
<td>SMB</td>
<td>445 &gt; 3635 [ACK] Seq=1398 Ack=1693 win=65535 Len=0</td>
</tr>
<tr>
<td>444</td>
<td>0.000660000</td>
<td>54</td>
<td>10.25.35.100</td>
<td>10.25.34.52</td>
<td>TCP</td>
<td>445 &gt; 3635 [ACK] Seq=1398 Ack=1693 win=65535 Len=0</td>
</tr>
<tr>
<td>445</td>
<td>0.000151000</td>
<td>60</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>3635 &gt; 445 [ACK] Seq=1693 Ack=1399 win=62843 Len=0</td>
</tr>
<tr>
<td>446</td>
<td>6.954838000</td>
<td>76</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>UDP</td>
<td>Source port: 53198 Destination port: 666</td>
</tr>
<tr>
<td>447</td>
<td>0.000378000</td>
<td>104</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>ICMP</td>
<td>Destination unreachable (Port unreachable)</td>
</tr>
<tr>
<td>448</td>
<td>3.557868000</td>
<td>90</td>
<td>10.25.35.100</td>
<td>10.25.34.24</td>
<td>NTP</td>
<td>NTP Version 4, client</td>
</tr>
<tr>
<td>449</td>
<td>0.000244000</td>
<td>90</td>
<td>10.25.35.100</td>
<td>10.25.34.24</td>
<td>NTP</td>
<td>NTP Version 3, server</td>
</tr>
</tbody>
</table>

Frame 446: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Internet Protocol Version 4, Src: 10.25.35.100 (10.25.35.100), Dst: 10.25.34.24 (10.25.34.24)
User Datagram Protocol, Src Port: 53198 (53198), Dst Port: 666 (666)

Data (34 bytes)

0000 00 50 56 a1 2f b3 00 50 56 a1 6b b1 08 00 45 00 ->...P..K...E.
0010 00 3e 9d ae 00 00 40 11 83 53 0a 19 23 64 0a 18 ..>...@ .S.#u.
0020 22 18 cf ce 02 0a 00 2a 59 e9 76 61 6c 6f 68 6f .......* Y.valoha
0030 2d 31 3a 20 53 6b 65 77 69 6e 67 20 63 6c 6f 66 -l: Skewing clock
0040 6b 20 6f 6e 20 61 61 64 6d 74 30 31 0a k on adm t01.
Drive mapping failing now
Fixing clock on domain controller

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Bytes</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1557</td>
<td>0.0000830000</td>
<td>78</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>88 &gt; 53807 [SYN, ACK] Seq=0 Ack=1 win=64240 Len=0 MSS=1460</td>
</tr>
<tr>
<td>1558</td>
<td>0.0000120000</td>
<td>66</td>
<td>10.25.35.100</td>
<td>10.25.34.24</td>
<td>TCP</td>
<td>53807 &gt; 88 [ACK] Seq=1 Ack=1 win=131712 Len=0</td>
</tr>
<tr>
<td>1559</td>
<td>0.0000240000</td>
<td>1328</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>KRB5</td>
<td>TGS-REQ</td>
</tr>
<tr>
<td>1560</td>
<td>0.0002220000</td>
<td>171</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>KRB5</td>
<td>KRB Error: KRB5KRB_AP_ERR_SKEW</td>
</tr>
<tr>
<td>1561</td>
<td>0.0000290000</td>
<td>66</td>
<td>10.25.35.100</td>
<td>10.25.34.24</td>
<td>TCP</td>
<td>53807 &gt; 88 [ACK] Seq=1263 Ack=106 win=131712 Len=0</td>
</tr>
<tr>
<td>1562</td>
<td>0.0000310000</td>
<td>66</td>
<td>10.25.35.100</td>
<td>10.25.34.24</td>
<td>TCP</td>
<td>88 &gt; 53807 [ACK] Seq=106 Ack=1264 win=62978 Len=0</td>
</tr>
<tr>
<td>1563</td>
<td>0.0000340000</td>
<td>60</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>88 &gt; 53807 [RST, ACK] Seq=106 Ack=1264 Win=0 Len=0</td>
</tr>
<tr>
<td>1564</td>
<td>0.0403540000</td>
<td>97</td>
<td>10.25.35.100</td>
<td>10.25.34.52</td>
<td>SMB</td>
<td>Session Setup AndX Response, Error: STATUS_LOGON_FAILURE</td>
</tr>
<tr>
<td>1565</td>
<td>0.0028570000</td>
<td>60</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>3842 &gt; 445 [FIN, ACK] Seq=1216 Ack=851 win=63390 Len=0</td>
</tr>
<tr>
<td>1566</td>
<td>0.0001800000</td>
<td>54</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>445 &gt; 3842 [ACK] Seq=851 Ack=1217 win=65535 Len=0</td>
</tr>
<tr>
<td>1567</td>
<td>0.0000660000</td>
<td>54</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>445 &gt; 3842 [FIN, ACK] Seq=851 Ack=1217 win=65535 Len=0</td>
</tr>
<tr>
<td>1568</td>
<td>0.0001200000</td>
<td>60</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>3842 &gt; 445 [ACK] Seq=1217 Ack=852 win=63390 Len=0</td>
</tr>
<tr>
<td>1569</td>
<td>9.9614830000</td>
<td>66</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>53803 &gt; 445 [FIN, ACK] Seq=63 Ack=182 win=131712 Len=0</td>
</tr>
<tr>
<td>1570</td>
<td>0.0003130000</td>
<td>66</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>445 &gt; 3803 [ACK] Seq=64 Ack=182 win=64178 Len=0</td>
</tr>
<tr>
<td>1571</td>
<td>0.0000180000</td>
<td>66</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>TCP</td>
<td>53803 &gt; 445 [ACK] Seq=64 Ack=183 win=131712 Len=0</td>
</tr>
<tr>
<td>1572</td>
<td>9.5744200000</td>
<td>66</td>
<td>10.25.34.52</td>
<td>10.25.35.100</td>
<td>UDP</td>
<td>Source Port: 56336 Destination port: 666</td>
</tr>
<tr>
<td>1573</td>
<td>0.0001760000</td>
<td>97</td>
<td>10.25.34.24</td>
<td>10.25.35.100</td>
<td>ICMP</td>
<td>Destination unreachable (Port unreachable)</td>
</tr>
</tbody>
</table>

Frame 1572: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface 0

---

The text marked with the red oval highlights the relevant part of the network capture: "Fixing time sk ew on ad mt01."
How to insert markers into pcaps?

There are a lot of ways to do this.

Building out this toolkit has taken me years.

And colleagues continue to teach me new ways.
TextPing

http://www.packetiq.com/Tools/PacketIQ-TextPing.aspx
Send-UDP-Msg

http://www.skendric.com/app

Or write your own … here’s mine

vishnu> ./send-udp-msg -m "This is a test ping" rhino1 rhino2 rhino3
vishnu>
So many techniques …

Send a TCP port 2049 frame to server.company.com
host> echo Starting NFS Mount now --marker | nc -4 -w 1 server.company.com 2049
C:\Temp> echo Starting NFS Mount now --marker | ncat -4 -w 1 server.company.com 2049
*For Windows, install the open source ncat utility [http://www.insecure.org](http://www.insecure.org), part of the Nmap distribution*

Send a UDP port 666 frame to server.company.com
host> echo Starting app now --marker | nc -4 -w 1 -u server.company.com 666
C:\Temp> echo Starting app now --marker | ncat -4 -w 1 -u server.company.com 666

Create a file, the name of the file will appear in Wireshark’s Summary screen
host> touch /mnt/whatever/slowness-starting-now--marker.txt
C:\Temp> copy /y nul z:slowness-starting-now--marker.txt

Drop the message into /var/log/syslog on loghost
host> logger -n loghost.company.com Slowness starting now --marker
C:\Temp> logger -n loghost.company.com Slowness starting now --marker
*For Windows, install the freeware logger utility [http://www.monitorware.com/logger](http://www.monitorware.com/logger)*

Drop the message into the Web server’s logs
host> wget http://www.company.com/slowness-starting-now--marker.html
C:\Temp> wget http://www.company.com/slowness-starting-now--marker.html
*For Windows, wget the open source GNU wget utility*

Drop the message into database server logs
SELECT name_last, name_first FROM name_table WHERE name_last ILIKE 'slowness starting now';
CLI Ping

In a pinch, you can use ping, manually maintaining a written table associating ping packet length to message:

host> ping -n 1 -l 101 server.company.com

host> ping -n 1 -l 102 server.company.com

host> ping -n 1 -l 103 server.company.com

<table>
<thead>
<tr>
<th>Ping Packet Length</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 bytes</td>
<td>Mounting file system</td>
</tr>
<tr>
<td>102 bytes</td>
<td>Starting application</td>
</tr>
<tr>
<td>103 bytes</td>
<td>Slowness beginning now</td>
</tr>
</tbody>
</table>

Or, depending on your filters, ping a fake host … the ping won’t show up in the trace, but the failed DNS query will:

host > ping www.slowness-starting-now--marker.com
C:\Temp> ping www.slowness-starting-now--marker.com
How to find these markers?

Once you’ve opened the trace, how do you find these markers?

Edit menu … Find Packet

I append the same string to all my markers … that way I can search through the trace and find them all, without having to remember unique strings for each marker.
Questions about Markers?

Questions up to this point …
So Is that It?

Inserting markers into trace files is a great start and is the RPR technique which I employ most frequently

But there’s more …
Drop markers into logs

2014-05-09T15:42:28-07:00 <23.6> valoha-1(id1) root: Skewing clock on admto1 -- marker

Now we start dropping these markers into log files, so that we can line up events recorded by operating systems and applications with pcaps
### Into ProcMon

<table>
<thead>
<tr>
<th>Time of Day</th>
<th>Process Name</th>
<th>PID</th>
<th>Operation</th>
<th>Path</th>
<th>Result</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:13:33.6847425 AM</td>
<td>svchost.exe</td>
<td>648</td>
<td>RegQueryKey</td>
<td>HKLM\System\CurrentControlSet\Control\DeviceClasses</td>
<td>SUCCESS</td>
<td>Query: HandleTags, HandleTags: 0x0</td>
</tr>
<tr>
<td>8:13:33.6847540 AM</td>
<td>svchost.exe</td>
<td>648</td>
<td>RegOpenKey</td>
<td>HKLM\System\CurrentControlSet\Control\DeviceClasses{c21f34-3458-49a9-88da-9e66}</td>
<td>SUCCESS</td>
<td>Desired Access: Query Value</td>
</tr>
<tr>
<td>8:13:33.6847767 AM</td>
<td>svchost.exe</td>
<td>648</td>
<td>RegQueryKey</td>
<td>HKLM\System\CurrentControlSet\Control\DeviceClasses</td>
<td>SUCCESS</td>
<td>Query: HandleTags, HandleTags: 0x0</td>
</tr>
<tr>
<td>8:13:33.6847967 AM</td>
<td>svchost.exe</td>
<td>648</td>
<td>RegQueryKey</td>
<td>HKLM\System\CurrentControlSet\Control\DeviceClasses{c21f34-3458-49a9-88da-9e66}</td>
<td>SUCCESS</td>
<td>Desired Access: Query Value</td>
</tr>
<tr>
<td>8:13:33.6849003 AM</td>
<td>svchost.exe</td>
<td>648</td>
<td>RegCloseKey</td>
<td>HKLM\System\CurrentControlSet\Control\DeviceClasses</td>
<td>NAME NOT FOUND</td>
<td>Desired Access: Query Value</td>
</tr>
<tr>
<td>8:13:34.1938004 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CreateFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: Read Attributes, Dispositi</td>
</tr>
<tr>
<td>8:13:34.1938209 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>QueryBasicInformationFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>CreationTime: 5/11/2014 8:17:26 AM, Last</td>
</tr>
<tr>
<td>8:13:34.1938237 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CloseFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: Read Attributes, Dispositi</td>
</tr>
<tr>
<td>8:13:34.1938752 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CreateFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>CreationTime: 5/11/2014 8:17:26 AM, Last</td>
</tr>
<tr>
<td>8:13:34.1937054 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>QueryBasicInformationFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: Read Attributes, Dispositi</td>
</tr>
<tr>
<td>8:13:34.1939221 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CloseFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>CreationTime: 5/11/2014 8:17:26 AM, Last</td>
</tr>
<tr>
<td>8:13:34.1939784 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CreateFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: Read Attributes, Dispositi</td>
</tr>
<tr>
<td>8:13:34.1938191 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>QueryBasicInformationFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>CreationTime: 5/11/2014 8:17:26 AM, Last</td>
</tr>
<tr>
<td>8:13:34.1900338 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CloseFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: General Write, Read Attri</td>
</tr>
<tr>
<td>8:13:34.1900958 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>OpenFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>AllocationSize: 0</td>
</tr>
<tr>
<td>8:13:34.1901263 AM</td>
<td>cmd.exe</td>
<td>5940</td>
<td>CreateFile</td>
<td>C:\Temp\skewing-clock-on-admt01-marker.txt</td>
<td>SUCCESS</td>
<td>Desired Access: Read Attributes, Dispositi</td>
</tr>
<tr>
<td>8:13:34.2944822 AM</td>
<td>Explorer.EXE</td>
<td>1636</td>
<td>RegQueryKey</td>
<td>HKCU\Software\Classes</td>
<td>SUCCESS</td>
<td>Query: Name</td>
</tr>
<tr>
<td>8:13:34.2945036 AM</td>
<td>Explorer.EXE</td>
<td>1636</td>
<td>RegQueryKey</td>
<td>HKCU\Software\Classes</td>
<td>SUCCESS</td>
<td>Query: HandleTags, HandleTags: 0x0</td>
</tr>
<tr>
<td>8:13:34.2945164 AM</td>
<td>Explorer.EXE</td>
<td>1636</td>
<td>RegQueryKey</td>
<td>HKCU\Software\Classes</td>
<td>SUCCESS</td>
<td>Query: HandleTags, HandleTags: 0x0</td>
</tr>
<tr>
<td>8:13:34.2945344 AM</td>
<td>Explorer.EXE</td>
<td>1636</td>
<td>RegOpenKey</td>
<td>HKCU\Software\Classes\Applications\cmd.exe</td>
<td>NAME NOT FOUND</td>
<td>Desired Access: Read Value</td>
</tr>
</tbody>
</table>
Into strace

gnat> strace -p 12345 -f -tt -s 256
[...
08:35:19.764185 mprotect(0x7f5e8f841000, 4096, PROT_READ) = 0
08:35:19.764261 munmap(0x7f5e8f82e000, 65819) = 0
08:35:19.764374 set_tid_address(0x7f5e8f82b9d0) = 10506
08:35:19.764498 set_robust_list(0x7f5e8f82b9e0, 0x18) = 0
08:35:19.764612 futex(0x7f7f6f59d2e9c, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 1, NULL
08:35:19.764820 rt_sigaction(SIGRTMIN, [0x7f5e8ee40750, [], SA_RESTART|SA_SIGINFO, 0x7f5e8ee49cb0], NULL
08:35:19.765027 rt_sigaction(SIGRT_1, [0x7f5e8ee407e0, [], SA_RESTART|SA_RESTART|SA_SIGINFO,
08:35:19.765210 rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
08:35:19.765340 getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY}) = 0
08:35:19.766014 brk(0) = 0x1a2d000
08:35:19.766073 brk(0x1a4e000) = 0x1a4e000
08:35:19.766143 open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
08:35:19.766220 fstat(3, {st_mode=S_IFREG|0644, st_size=7220736, ...}) = 0
08:35:19.766287 mmap(NULL, 7220736, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f5e8e757000
08:35:19.766844 close(3) = 0
08:35:19.766966 open("Skewing-clock-on-admt01--marker", O_WRONLY|O_CREAT|O_NOCTTY|O_NONBLOCK, 0666) = 3
08:35:19.767746 dup2(3, 0) = 0
08:35:19.768110 close(3) = 0
08:35:19.768172 dup2(0, 0) = 0
08:35:19.768227 utimensat(0, NULL, NULL, 0) = 0
[...
Let’s Back Out A Bit

I chose to present this topic by diving immediately into the details of how to use markers to guide our attention during pcap analysis.

And have now expanded the use of markers to include guiding our attention during log analysis.

And guiding our attention during analysis of the output of trouble-shooting tools like ProcMon and strace.

I would like to pull back from examining the leaves to examine the entire tree.
Diagnostic Capture Plan

Who talks to whom?
Where to insert probes?
Where to gather logs / debug output?

User
Client
Access Switch
Router
Data Center Switch
Application Servers
Database Server
Directory Server (DNS, LDAP, NIS …)
Fibre Channel Switch
Storage
Definitive Diagnostic Data

Often, we solve problems through an intuitive approach, gathering a pcap here, a log file there, running an experiment, and at last identifying the root cause.

RPR suggests that for a certain class of big, hairy problem, our usual approaches not only cost a lot in terms of time but may never converge on the answer.

RPR proposes that for problems of sufficient complexity, particularly intermittent ones, we take the time to:

1. Instrument the entire path of the troubled transaction
2. Validate the data collection tools by sending a sample transaction end-to-end
   
   `UPDATE salary_table SET annual=100000 WHERE name=“Mickey Mouse”`

3. Lie in wait for the next occurrence of the problem, sending markers when it happens
4. Analyze the resulting ‘video’ of the troubled transaction, using the end-to-end view captured in D³, to definitively finger the portion of the path causing the problem

It’s a judgment call on when it is worth investing this level of effort.
Judgment Call

It’s a judgment call on when it is worth investing this level of effort.

Myself, I tend to explore for a while using my intuitive approach, before finally accepting the need to bring out the power tool: Definitive Diagnostic Data.

In fact, I tend to tell myself:

Self  This time, I’ll solve the problem using some clever shortcut, and I won’t need to spend all that time setting up a formal Diagnostic Capture Plan.

[Many hours or days later]

Self  Wish you’d started with D³, don’t you? Thought you could get away with a shortcut, didn’t you?

I like to think that optimism is a desirable character trait … 😊
The Fantasy …

Ideally, you’re sitting next to the end-user’s computer, reading {insert your favorite book here} when she says “There, see, it’s hanging right now!” And you ask her to double-click on the desktop icon which fires off your marker:

```
UPDATE salary_table SET annual=100000 WHERE name=“Mickey Mouse”
```

Then, you walk back to your desk and double-click the icon which automatically logs into each sniffer, copies the latest pcap (neatly named), logs into each server, copies the last 15 minutes of logs … ProcMon … strace … and collects all these data files into a folder named Analyze-Me.

On your four 30” monitors, you open all the files, organized from left to right reflecting the Diagnostic Capture Plan diagram, and you search in each one for ‘Mickey Mouse’, lining them all up. Immediately, you can see that the Database Server took two minutes to return the OK symbol, and during that time, the DBM was logging messages like “Table salary locked exclusive, waiting”, you walk over to the database manager, describe what you see, he smacks his forehead and says “Duh! Of course! I’ve been meaning to fix that for months … [type, type] … OK try it now, I bet it works” …

and … you earn your paycheck that day.
The Reality …

This is not as easy as it looks. –The Man in Black

- The application encrypts its traffic over the wire, so you can’t see Mickey Mouse in the pcaps
- So when the user clicks the icon, she’s sending a custom ping packet through the front-half of the transaction, while you attempt, as simultaneously as you can, to send another custom ping packet through the back-half of the transaction – they won’t be lined up perfectly, but hey, the best you can do
- Simultaneously, you give your colleagues a shout and they insert custom log messages into the logs of various applications along the way: Mickey Mouse is Here … again not perfectly lined up

And then you discover:

- One of the sniffers was hung
- You fat-fingered the marker insertion at one of the steps
- You forgot to copy the logs off a particular server and by the time you notice, they have been overwritten
More tips

Automate as much typing as possible:

Batch files (PowerShell, bash, Perl, Python, whatever)

So, instead of typing:

```
echo Starting NFS Mount now --marker | ncat -4 -w 1 server.company.com 2049
```

Write shell scripts to do the same: `ins-start-marker` and `ins-end-marker`

The checklist is your friend: [The Checklist Manifesto](http://www.skendric.com/philosophy/uptime/DaPlan-Hobbes.pdf) by Atul Gawande

e.g.

When the pathology starts, we’re going to do the following:

- Bob   double-clicks on ‘Pathology-Starting’ icon
- Sarah types `ins-start-marker` on the Web Server
- Jiang types `ins-start-marker` on the Application Server

...
Back Out Even Further

Let’s back out and look at the entire forest, or at least at a grove of trees
How Do Techs Fix Issues?

Oh boy, that’s a big question. But let’s take a stab at answering it. A tech might start asking themselves, or the person reporting the problem, questions similar to the following:

• What makes you think there is an issue?
• What are you expecting that you’re not getting?
• Has it ever performed well?
• What changed recently? Software or hardware? Load?
• Can it be expressed in terms of latency or run time?
• Does the problem affect other people or applications?
• What is the environment? What software and hardware is used? Versions? Configuration?
• …

Most issues get fixed somewhere during the process of asking these questions and uncovering the answers …
Anti-Patterns

As the issue resists resolution, less skilled techs will start employing less effective approaches.

Street Lamp Method
The student comes across his professor on the Arts Quad at night, down on his hands & knees, staring at the sidewalk. “What are you doing, sir?” “Looking for my car keys”. The student joins the professor but after looking unsuccessfully in widening circles, asks him “Do you recall precisely where you were when you dropped the keys?” “Yes, over there, in the middle of the quad” points the professor, toward the dimly perceived middle of the grassy acre. “Well, why are you looking here?” asks the student. “Because the light is better here” responds the professor.

More formally:

:START
1. List available tools
2. Examine the output of each one, looking for clues
3. Purchase more tools
4. Goto START

Use The Force, Luke
“I know that we are experiencing a broadcast storm … you should check your {switch | router | firewall | server | client | application | whatever-belongs-to-some-other-group}”

I enjoyed Star Wars … but it was fiction, not real … that distinction is hard for human brains to make. --sk
When It Really Hurts

What happens when your technical teams (desktop, server, network, storage, database, application …) have looked at the issue and are stumped?

Or worse, have tried something, and it didn’t help …

Or even worse … are now avoiding the issue …

Your vendor tells you to upgrade to the next biggest model / version …

Tensions rise, people point fingers … blame-based language …

What’s next?
Rapid Problem Resolution ®

Advance7 is a consulting outfit which helps customers resolve critical Problems – they put an analyst at your site to coordinate your staff plus vendors to fix the issue, using the RPR methodology. They tend to play in the Fortune 1000 + government space.

RPR is an evidence-based trouble-shooting approach.


Advance7 designed RPR to work against Grey Problems

Most Problems are not Grey … unless the Problem is Grey, RPR is overkill.

So what are Grey Problems?

The following sides are cribbed from Advance7 -- credit to Paul Offord & his colleagues.
The Grey Problem

The majority of issues that are passed to 2nd and 3rd line technical support teams are investigated in a straightforward manner. The nature of the issue or an indication from a monitoring system identifies the failing component, and the issue is assigned to the correct technical support team. Q1: the bulk of support work falls into this area. Q2 is harder but tends to be resolved by experienced support staff. Q3 are sticky; we tend not to solve these.

An intermittent response-time or error issue is not so easily handled due to its transient nature. Not only does the cause sneak under the radar of monitoring systems, but investigation often starts after the issue has passed, making it impossible to use many of the tools available. The result is a recurring problem where the causing technology is unknown: Q4, aka the Grey Problem. The Rapid Problem Resolution methodology targets Q4.
Grey Problem Characteristics

Because the causing technology is unknown, a grey problem will bounce between Technical Support Teams as each in turn produces evidence (often in the form of a health check) to prove that their technology is not to blame.

Typical characteristics of a grey problem
- An ever-growing number of people become involved
- Long meetings to discuss what might be the cause
- Support people shy away from becoming involved
- Repeated changes with no clear reason or objective
- Technical Support Teams hire their vendors to perform health checks

Consequences of grey problems
- An ever growing backlog of problems
- A fog that hinders the investigation of other, more urgent problems
- A growing pool of problems that escalate into Major Incidents as patterns of use and business priorities change
- Wasted IT budget as money is spent on poorly targeted upgrades
- Barriers to integration due to concerns about the stability of component systems
- Loss of confidence and satisfaction with the IT department
- Pressure to outsource IT services
- Reduced customer satisfaction
- Higher costs as the business adjusts to accommodate the problem
- Higher IT staffing costs
## RPR Roles & Responsibilities

<table>
<thead>
<tr>
<th>Who</th>
<th>What</th>
</tr>
</thead>
</table>
| **Facilitator**<br>(often a Problem Manager) | • Accountable for  
  o Owns the RCA  
  o Acquire resources  
  o Use and execute the methodology  
  o Communicate within the team  
  o Report & escalate to leadership  
  o Schedule meetings |
| **Problem Analyst**<br>(often a senior tech) | • Responsible for  
  o Unify & synthesize information from SMEs  
  o Keep team on track technically  
  o Breadth & depth |
| **Subject Matter Experts** | • Responsible for  
  o Strong fundamental knowledge of area  
  o Facilitating access  
  o Capturing data  
  o Analyzing |
| **SME Desirable Characteristics** | • Skills / Predilections  
  o Problem solving skills  
  o Inquiring mind – passion for understanding how things work  
  o Determination & stamina – pursuing a tough problem can be wearing  
  o T-shaped – broad background in IT with specialization in one or two particular areas |
Simplified RPR

1. Understand the Symptoms
2. Choose One Symptom
3. Understand the Symptom Environment
4. Design Diagnostic Capture Plan
5. Capture *Definitive Diagnostic Data*
6. Analyze Captured Data
7. Identify Fix
8. Implement Fix
9. Verify Fix
The Full RPR Methodology

See the Rapid Problem Resolution book and Advance7’s Web site for more information
Summary

1. Instrument the entire path of the troubled transaction

2. Validate the data collection tools by sending a sample transaction end-to-end
   *Increase Mickey Mouse’s salary to 100,000* --marker

3. Lie in wait for the next occurrence of the problem.
   Ideally, the troubled transaction shows up in a recognizable way at each probe: “I was updating the customer’s address from 100 Main St to 200 Broad St when my application froze” – search for Broad St.
   But you may need to manually insert markers into the data stream to build a richer trail of bread crumbs

4. Analyze the resulting ‘video’ of the troubled transaction, using the end-to-end view captured in D³, to **definitively** finger the portion of the path causing the problem
Thank you!

On-Line Resources

**Rapid Problem Resolution** by Paul Offord
LinkedIn **Protocol Analysis & Troubleshooting Group**
Old Comm Guy [http://www.lovemytool.com](http://www.lovemytool.com)

Trouble-shooting & Training Outfits

James Baxter [http://www.packetiq.com](http://www.packetiq.com)
Tony Fortunato [http://www.thetechfirm.com](http://www.thetechfirm.com)
Chris Greer [http://www.packetpioneer.com](http://www.packetpioneer.com)
Paul Offord [http://www.advance7.com](http://www.advance7.com)
Mike Pennacchi [http://www.nps-llc.com](http://www.nps-llc.com)
Ray Tompkins [http://www.gearbit.com](http://www.gearbit.com)

…

Conferences

Sharkfest [http://www.sharkfest.org](http://www.sharkfest.org)

Follow-up

stuart.kendrick.sea {at} gee mail dot com
This deck visible at [http://www.skendric.com/seminar](http://www.skendric.com/seminar)