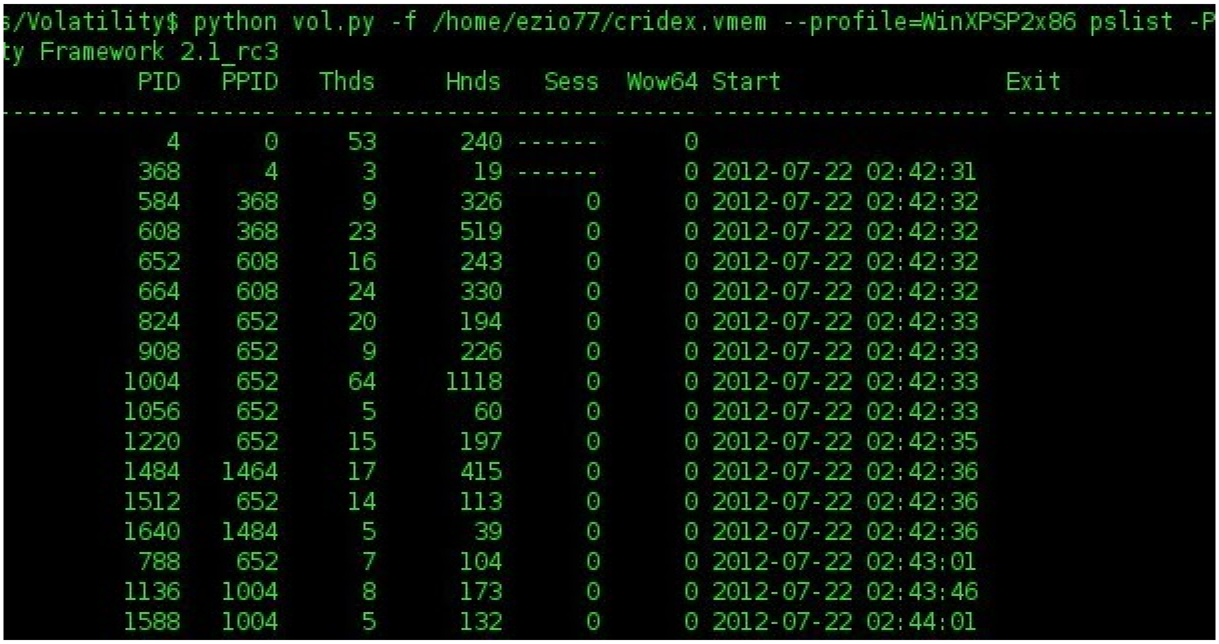
Cridex Analysis using Volatility

sempersecurus.org/2012/08/cridex-analysis-using-volatility.html



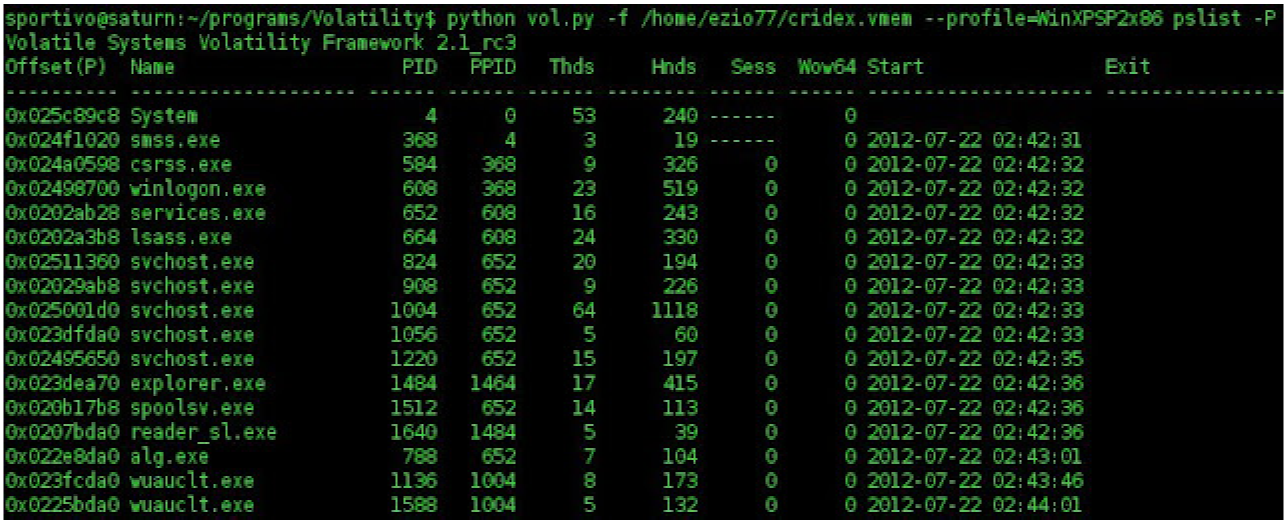
```
s/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 pslist -P
ty Framework 2.1_rc3
         PID   PPID   Thds     Hnds   Sess  Wow64 Start                Exit
------ ------ ------ ------ -------- ------ ------ -------------------- --------------
           4      0     53      240 ------      0
         368      4      3       19 ------      0 2012-07-22 02:42:31
         584    368      9      326      0      0 2012-07-22 02:42:32
         608    368     23      519      0      0 2012-07-22 02:42:32
         652    608     16      243      0      0 2012-07-22 02:42:32
         664    608     24      330      0      0 2012-07-22 02:42:32
         824    652     20      194      0      0 2012-07-22 02:42:33
         908    652      9      226      0      0 2012-07-22 02:42:33
        1004    652     64     1118      0      0 2012-07-22 02:42:33
        1056    652      5       60      0      0 2012-07-22 02:42:33
        1220    652     15      197      0      0 2012-07-22 02:42:35
        1484   1464     17      415      0      0 2012-07-22 02:42:36
        1512    652     14      113      0      0 2012-07-22 02:42:36
        1640   1484      5       39      0      0 2012-07-22 02:42:36
         788    652      7      104      0      0 2012-07-22 02:43:01
        1136   1004      8      173      0      0 2012-07-22 02:43:46
        1588   1004      5      132      0      0 2012-07-22 02:44:01
```

Update 1 - August 5, 2012 - located at end of post

Update 2 - August 7, 2012 - located at end of post

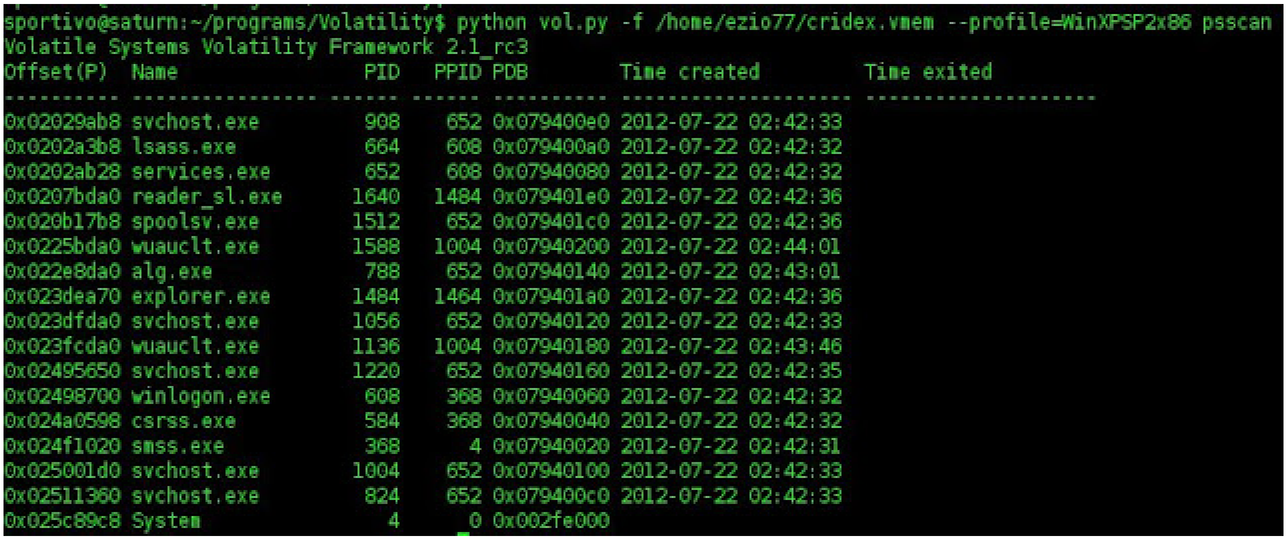
I had read previous analysis reports about Cridex from various sites as M86 Security and Kahu Security. At the time, I filed this under "another banking trojan" to track, and moved on to to other things. However Cridex once again piqued my interest when I saw an excellent analysis by Kimberly at StopMalvertising. I took particular attention to her listing of the Cridex C&C servers she observed, as several of these IP blocks were familiar to me. More on this later.   Having obtained the same Cridex sample analyzed by Kimberly, I was interested to see how Volatility could be used to analyze it. This Cridex sample had MD5 hash, 734aadd62d0662256a65510271d40048. I executed the sample and dumped the memory for analysis. A copy of this memory dump is linked at the bottom of this post.

Using the Volatility 'plist' command, we can see a list of the running processes. However it's instructive to use this in conjunction with the 'psscan' command in order to see those processes that have terminated, are unlinked, or hidden. In this case, no discrepancies between the two commands jump out at me, but I do notice a couple of things.   First, I see a process, ***reader_sl.exe, PID1640*** start exactly at the same time as its parent process, ***explorer.exe, PID1484***. I see that the parent process ID for *explorer.exe* is *1464*, which is not listed in either 'pslist' or 'psscan'. *reader_sl.exe* is a supposedly a safe process, associated with Adobe Speed Launcher, but the launch chain for this seems odd, so I'll keep note of this for now. Next, I see a second *wuauclt.exe* process start about 15 seconds after the first. This isn't a major flag, but just something to note.



```
sportivo@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 pslist -P
Volatile Systems Volatility Framework 2.1_rc3
Offset(P)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ -------------------- --------------------
0x025c89c8 System                    4      0     53      240 ------      0
0x024f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31
0x024a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32
0x02498700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32
0x0202ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32
0x0202a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32
0x02511360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33
0x02029ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33
0x025001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33
0x023dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33
0x02495650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35
0x023dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36
0x020b17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36
0x0207bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36
0x022e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01
0x023fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46
0x0225bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01
```

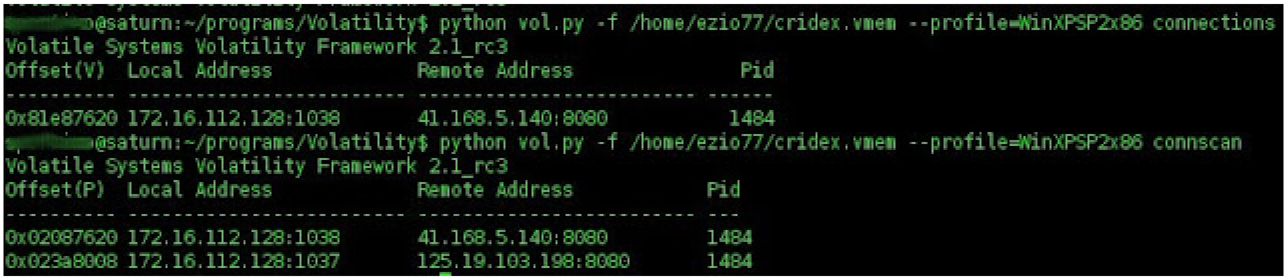
pslist command



```
sportivo@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 psscan
Volatile Systems Volatility Framework 2.1_rc3
Offset(P)  Name                PID   PPID PDB        Time created         Time exited
---------- ---------------- ------ ------ ---------- -------------------- --------------------
0x02029ab8 svchost.exe         908    652 0x079400e0 2012-07-22 02:42:33
0x0202a3b8 lsass.exe           664    608 0x079400a0 2012-07-22 02:42:32
0x0202ab28 services.exe        652    608 0x07940080 2012-07-22 02:42:32
0x0207bda0 reader_sl.exe      1640   1484 0x079401e0 2012-07-22 02:42:36
0x020b17b8 spoolsv.exe        1512    652 0x079401c0 2012-07-22 02:42:36
0x0225bda0 wuauclt.exe        1588   1004 0x07940200 2012-07-22 02:44:01
0x022e8da0 alg.exe             788    652 0x07940140 2012-07-22 02:43:01
0x023dea70 explorer.exe       1484   1464 0x079401a0 2012-07-22 02:42:36
0x023dfda0 svchost.exe        1056    652 0x07940120 2012-07-22 02:42:33
0x023fcda0 wuauclt.exe        1136   1004 0x07940180 2012-07-22 02:43:46
0x02495650 svchost.exe        1220    652 0x07940160 2012-07-22 02:42:35
0x02498700 winlogon.exe        608    368 0x07940060 2012-07-22 02:42:32
0x024a0598 csrss.exe           584    368 0x07940040 2012-07-22 02:42:32
0x024f1020 smss.exe            368      4 0x07940020 2012-07-22 02:42:31
0x025001d0 svchost.exe        1004    652 0x07940100 2012-07-22 02:42:33
0x02511360 svchost.exe         824    652 0x079400c0 2012-07-22 02:42:33
0x025c89c8 System                4      0 0x002fe000
```

psscan command

The next useful Volatility command that I use for malware analysis is the 'connections' and the 'connscan' commands. Again, running both of these will allow you to see variances, as 'connscan' will show artifacts from previous connections.



```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 connections
Volatile Systems Volatility Framework 2.1_rc3
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ------
0x81e87620 172.16.112.128:1038       41.168.5.140:8080          1484
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 connscan
Volatile Systems Volatility Framework 2.1_rc3
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x02087620 172.16.112.128:1038       41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037       125.19.103.198:8080       1484
```

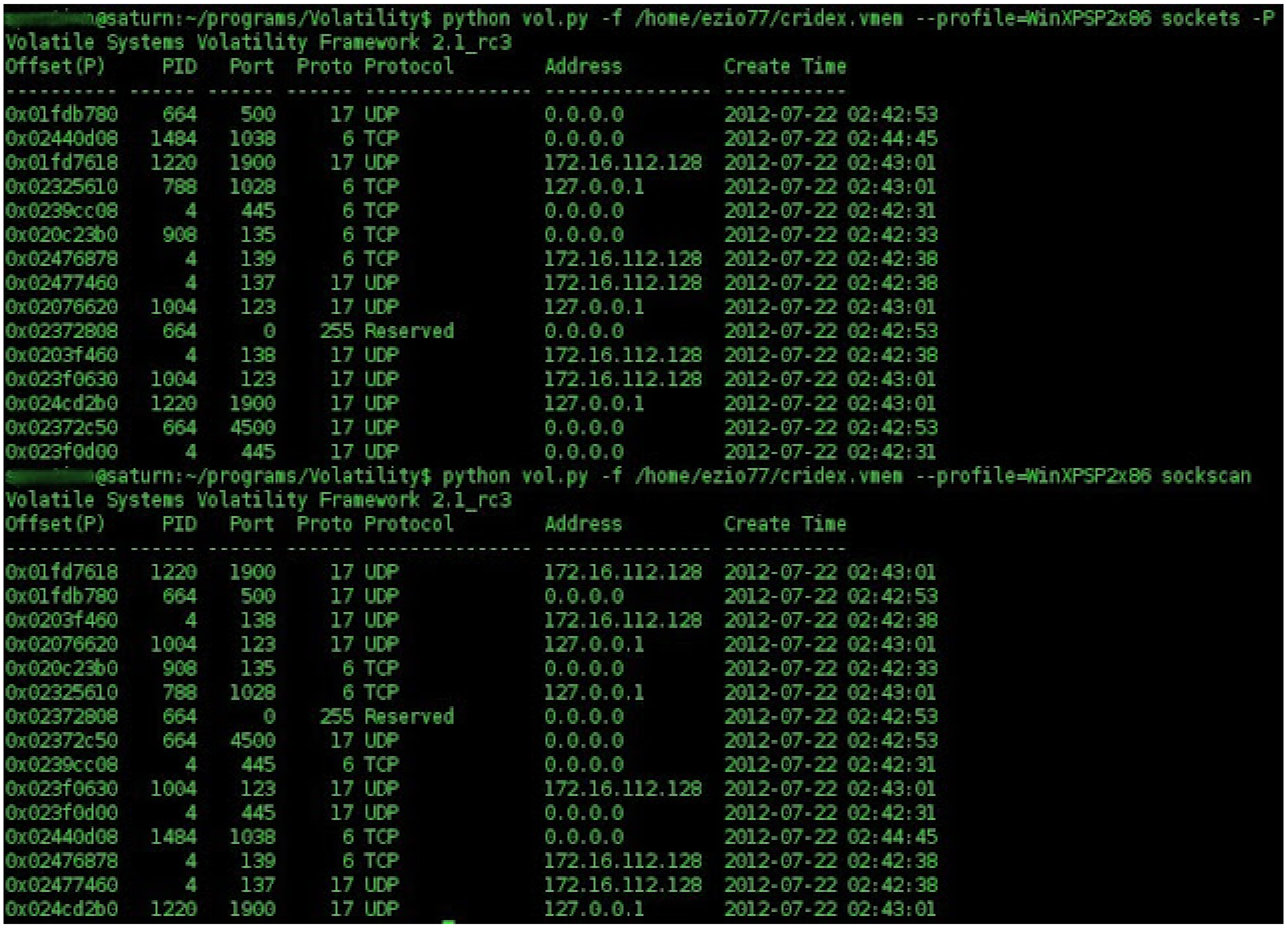
connections & connscan commands

Note that 'connections' shows that PID 1484, *explorer.exe* had an active connection to remote IP address **41.168.5.140** on port 8080. 'connscan' shows an artifact of a previous connection by PID 1484 to remote IP address **125.19.103.198**, also on port 8080. A quick 'whois' shows: **41.168.5.140**

netname: NEOTEL
descr: NEOTEL PTY LTD
country: ZA

125.19.103.198
descr: Bharti Tele-Ventures Limited
descr: NEW DELHI
country: IN

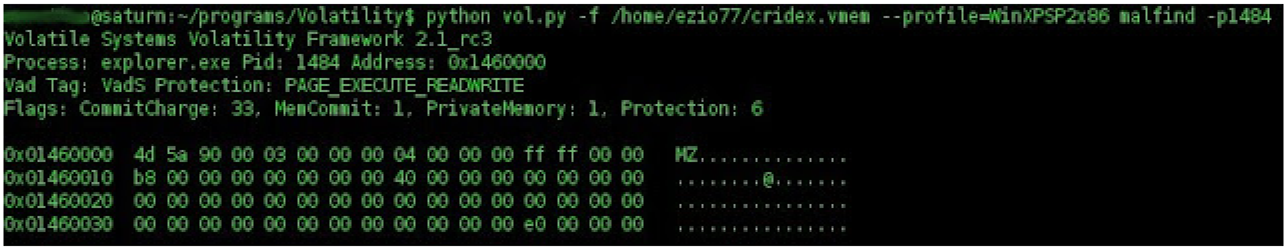
Next, running 'sockets' and 'sockscan' will show any listening sockets that may have been initiated by a running process. As in 'conscan', 'sockscan' will show any detected artifacts from previous sockets. In this case, we see that PID 1484, *explorer.exe*, opened a listening socket on port 1038 approx. 2 minutes after PID 1484 was created.



```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 sockets -P
Volatile Systems Volatility Framework 2.1_rc3
Offset(P)     PID   Port  Proto Protocol        Address         Create Time
---------- ------ ------ ------ --------------- --------------- -----------
0x01fdb780    664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53
0x02440d08   1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45
0x01fd7618   1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01
0x02325610    788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01
0x0239cc08      4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31
0x020c23b0    908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33
0x02476878      4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38
0x02477460      4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38
0x02076620   1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01
0x02372808    664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53
0x0203f460      4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38
0x023f0630   1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01
0x024cd2b0   1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01
0x02372c50    664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53
0x023f0d00      4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 sockscan
Volatile Systems Volatility Framework 2.1_rc3
Offset(P)     PID   Port  Proto Protocol        Address         Create Time
---------- ------ ------ ------ --------------- --------------- -----------
0x01fd7618   1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01
0x01fdb780    664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53
0x0203f460      4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38
0x02076620   1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01
0x020c23b0    908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33
0x02325610    788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01
0x02372808    664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53
0x02372c50    664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53
0x0239cc08      4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31
0x023f0630   1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01
0x023f0d00      4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31
0x02440d08   1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45
0x02476878      4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38
0x02477460      4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38
0x024cd2b0   1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01
```

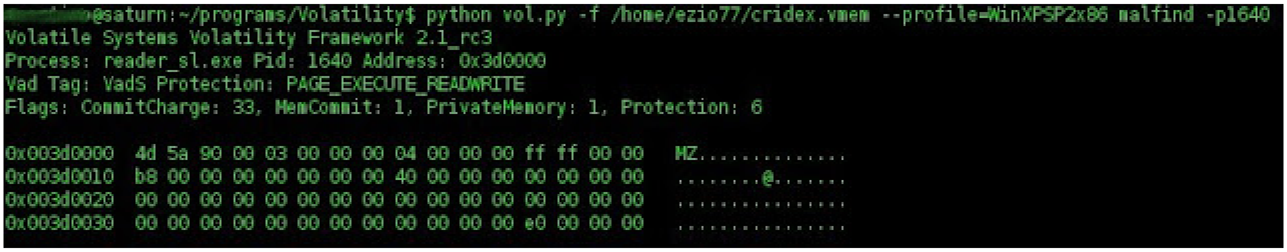
sockets and sockscan commands

Running the 'malfind' command against our two suspect processes yields the following:



```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 malfind -p1484
Volatile Systems Volatility Framework 2.1_rc3
Process: explorer.exe Pid: 1484 Address: 0x1460000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x01460000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x01460010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x01460020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x01460030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................
```



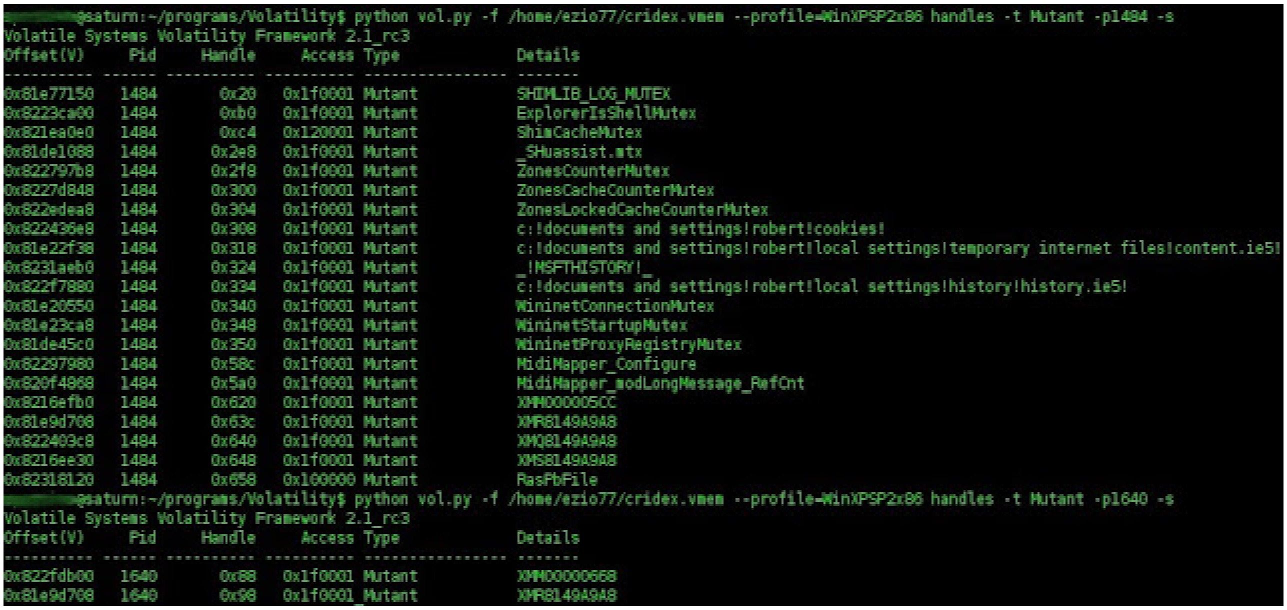
```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 malfind -p1640
Volatile Systems Volatility Framework 2.1_rc3
Process: reader_sl.exe Pid: 1640 Address: 0x3d0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 33, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x003d0000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x003d0010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x003d0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x003d0030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................
```

malfind command on PID 1484 & 1640

In this output, we see that the *explorer.exe*, PID1484 and *reader_sl.exe*, PID1640 processes have a PE section located at 0x1460000 and 0x3d0000 respectively. By using the "-D" switch, 'malfind' can dump those identified segments to a dump directory for further analysis.

We now enumerate the mutant/mutex objects for the two processes under review. Note that I used the Volatility 'handles' command, with a subtype selection of "Mutant" in order to specifically select the mutant/mutexes associated with PID 1484 and 1640. The 'mutantscan' command will give additional information such as its signaled state, its client ID, and which thread acquired the mutant.



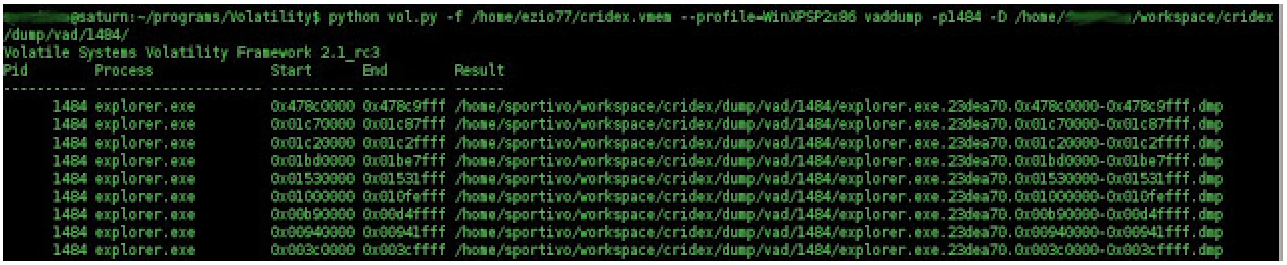
```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 handles -t Mutant -p1484 -s
Volatile Systems Volatility Framework 2.1_rc3
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x81e77150   1484       0x20   0x1f0001 Mutant           SHIMLIB_LOG_MUTEX
0x8223ca00   1484       0xb0   0x1f0001 Mutant           ExplorerIsShellMutex
0x821ea0e0   1484       0xc4   0x120001 Mutant           ShimCacheMutex
0x81de1088   1484      0x2e8   0x1f0001 Mutant           _SHuassist.mtx
0x822797b8   1484      0x2f8   0x1f0001 Mutant           ZonesCounterMutex
0x8227d848   1484      0x300   0x1f0001 Mutant           ZonesCacheCounterMutex
0x822edea8   1484      0x304   0x1f0001 Mutant           ZonesLockedCacheCounterMutex
0x822436e8   1484      0x308   0x1f0001 Mutant           c:!documents and settings!robert!cookies!
0x81e22f38   1484      0x318   0x1f0001 Mutant           c:!documents and settings!robert!local settings!temporary internet files!content.ie5!
0x8231aeb0   1484      0x324   0x1f0001 Mutant           _!MSFTHISTORY!_
0x822f7880   1484      0x334   0x1f0001 Mutant           c:!documents and settings!robert!local settings!history!history.ie5!
0x81e20550   1484      0x340   0x1f0001 Mutant           WininetConnectionMutex
0x81e23ca8   1484      0x348   0x1f0001 Mutant           WininetStartupMutex
0x81de45c0   1484      0x350   0x1f0001 Mutant           WininetProxyRegistryMutex
0x82297980   1484      0x58c   0x1f0001 Mutant           MidiMapper_Configure
0x820f4868   1484      0x5a0   0x1f0001 Mutant           MidiMapper_modLongMessage_RefCnt
0x8216efb0   1484      0x620   0x1f0001 Mutant           XMM000005CC
0x81e9d708   1484      0x63c   0x1f0001 Mutant           XMR8149A9A8
0x822403c8   1484      0x640   0x1f0001 Mutant           XMQ8149A9A8
0x8216ee30   1484      0x648   0x1f0001 Mutant           XMS8149A9A8
0x82318120   1484      0x658   0x100000 Mutant           RasPbFile
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 handles -t Mutant -p1640 -s
Volatile Systems Volatility Framework 2.1_rc3
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x822fdb00   1640       0x88   0x1f0001 Mutant           XMM00000668
0x81e9d708   1640       0x98   0x1f0001 Mutant           XMR8149A9A8
```

process mutexes for PID 1484 & 1640

Via some Google queries, we learn that several of these mutex objects have been seen in other malware, notably:

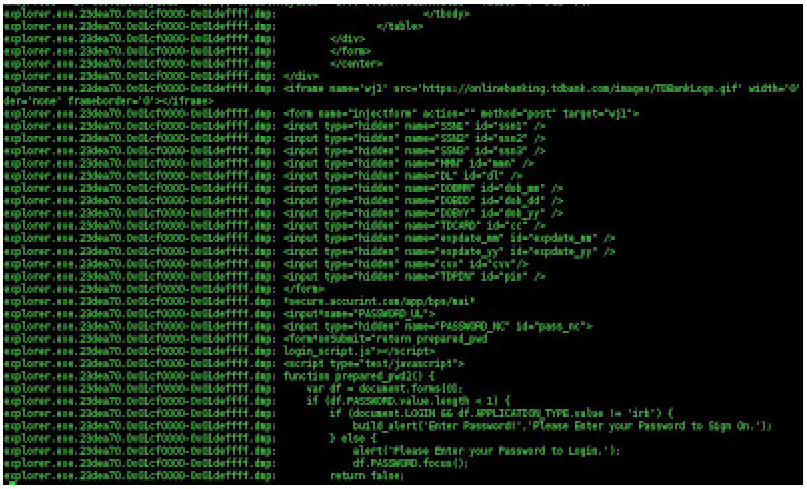
- 746bbf3569adEncrypt
- _SHuassist.mtx
- SHIMLIB_LOG_MUTEX
- XMR8149A9A8

Next, we'll dump the VAD segments of each of these processes, run 'strings', and look for anything interesting.



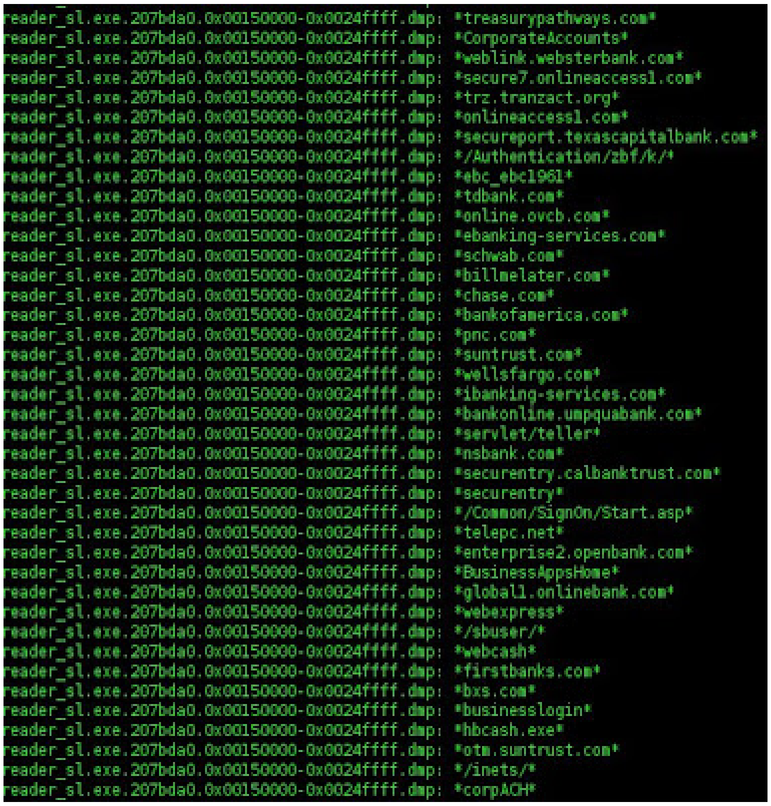
```
[redacted]@saturn:~/programs/Volatility$ python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 vaddump -p1484 -D /home/[redacted]/workspace/cridex
/dump/vad/1484/
Volatile Systems Volatility Framework 2.1_rc3
Pid        Process              Start      End        Result
---------- -------------------- ---------- ---------- ------
      1484 explorer.exe         0x478c0000 0x478c9fff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x478c0000-0x478c9fff.dmp
      1484 explorer.exe         0x01c70000 0x01c87fff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x01c70000-0x01c87fff.dmp
      1484 explorer.exe         0x01c20000 0x01c2ffff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x01c20000-0x01c2ffff.dmp
      1484 explorer.exe         0x01bd0000 0x01be7fff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x01bd0000-0x01be7fff.dmp
      1484 explorer.exe         0x01530000 0x01531fff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x01530000-0x01531fff.dmp
      1484 explorer.exe         0x01000000 0x010fefff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x01000000-0x010fefff.dmp
      1484 explorer.exe         0x00b90000 0x00d4ffff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x00b90000-0x00d4ffff.dmp
      1484 explorer.exe         0x00940000 0x00941fff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x00940000-0x00941fff.dmp
      1484 explorer.exe         0x003c0000 0x003cffff /home/sportivo/workspace/cridex/dump/vad/1484/explorer.exe.23dea70.0x003c0000-0x003cffff.dmp
```

vaddump command



```
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:                         </tbody>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:                 </table>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         </div>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         </form>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         </center>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: </div>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <iframe name='wj1' src='https://onlinebanking.tdbank.com/images/TDBankLogo.gif' width='0'
der='none' frameborder='0'></iframe>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <form name="injectform" action="" method="post" target="wj1">
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="SSN1" id="ssn1" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="SSN2" id="ssn2" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="SSN3" id="ssn3" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="MMN" id="mmn" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="DL" id="dl" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="DOBMM" id="dob_mm" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="DOBDD" id="dob_dd" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="DOBYY" id="dob_yy" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="TDCARD" id="cc" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="expdate_mm" id="expdate_mm" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="expdate_yy" id="expdate_yy" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="cvv" id="cvv"/>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="TDPIN" id="pin" />
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: </form>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: "secure.accurint.com/app/bps/mai"
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input"name="PASSWORD_UL">
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <input type="hidden" name="PASSWORD_NC" id="pass_nc">
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <form"onSubmit="return prepared_pwd
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: login_script.js"></script>
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: <script type="text/javascript">
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: function prepared_pwd2() {
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:     var df = document.forms[0];
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:     if (df.PASSWORD.value.length < 1) {
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         if (document.LOGIN && df.APPLICATION_TYPE.value != 'irb') {
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:             build_alert('Enter Password!','Please Enter your Password to Sign On.');
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         } else {
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:             alert('Please Enter your Password to Login.');
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:             df.PASSWORD.focus();
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp:         return false;
```

'strings' output section from PID 1484, *explorer.exe*



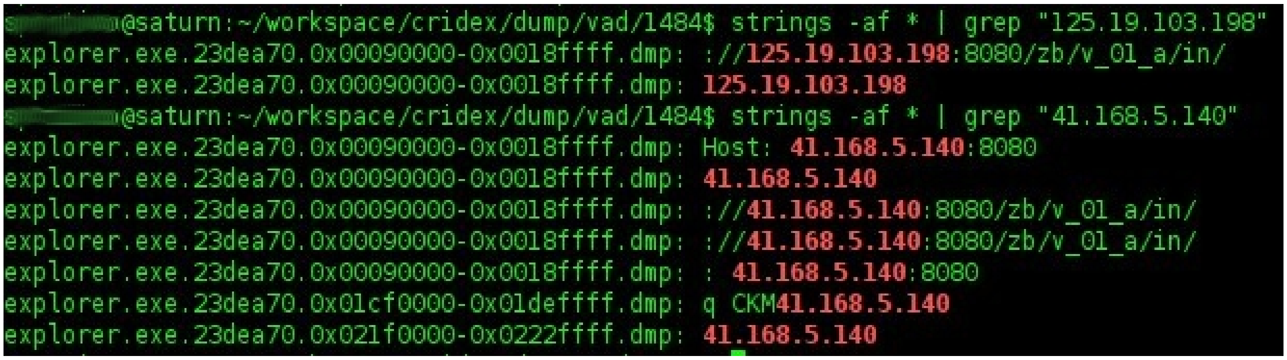
```
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *treasurypathways.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *CorporateAccounts*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *weblink.websterbank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *secure7.onlineaccess1.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *trz.tranzact.org*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *onlineaccess1.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *secureport.texascapitalbank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: */Authentication/zbf/k/*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *ebc_ebc1961*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *tdbank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *online.ovcb.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *ebanking-services.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *schwab.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *billmelater.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *chase.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *bankofamerica.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *pnc.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *suntrust.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *wellsfargo.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *ibanking-services.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *bankonline.umpquabank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *servlet/teller*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *nsbank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *securentry.calbanktrust.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *securentry*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: */Common/SignOn/Start.asp*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *telepc.net*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *enterprise2.openbank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *BusinessAppsHome*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *global1.onlinebank.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *webexpress*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: */sbuser/*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *webcash*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *firstbanks.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *bxs.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *businesslogin*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *hbcash.exe*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *otm.suntrust.com*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: */inets/*
reader_sl.exe.207bda0.0x00150000-0x0024ffff.dmp: *corpACH*
```

'strings' output section from PID 1640, *reader_sl.exe*

Note the advantage of dumping the VAD segments as opposed to the entire process memory is that you can see which VAD node section the 'strings' hit was located. In this section, we find a list of banks and financial institutions.   Here is the contents of the Cridex configuration specifically containing references to financial institutions.

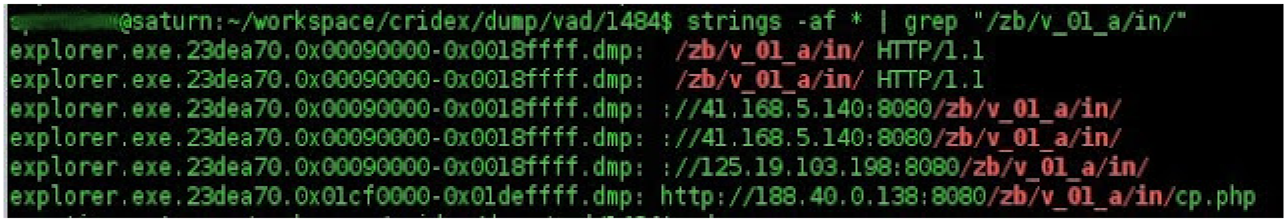
In addition to the list above, examining these VAD dumps also shows HTML code referencing or representing web pages of various financial organizations. The code seems to indicate that these sections are part of the web injection code that is used to obtain personal information from the banking customer. In my test of Cridex, I did not launch a web browser or continue additional interaction with my infected host. If I had visited a URL containing these strings, it is believed that Cridex would attempt to log or capture my input, and redirect that personal information back to the controller.

While we're looking for strings, let's see what shows up for the IP addresses **41.168.5.140** & **125.19.103.198** that were seen in the Volatility "connscan" command.



```
[redacted]@saturn:~/workspace/cridex/dump/vad/1484$ strings -af * | grep "125.19.103.198"
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://125.19.103.198:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: 125.19.103.198
[redacted]@saturn:~/workspace/cridex/dump/vad/1484$ strings -af * | grep "41.168.5.140"
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: Host: 41.168.5.140:8080
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: 41.168.5.140
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://41.168.5.140:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://41.168.5.140:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: : 41.168.5.140:8080
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: q CKM41.168.5.140
explorer.exe.23dea70.0x021f0000-0x0222ffff.dmp: 41.168.5.140
```

Searching for the directory path after the IP addresses gives us another related IP address, 188.40.0.138:

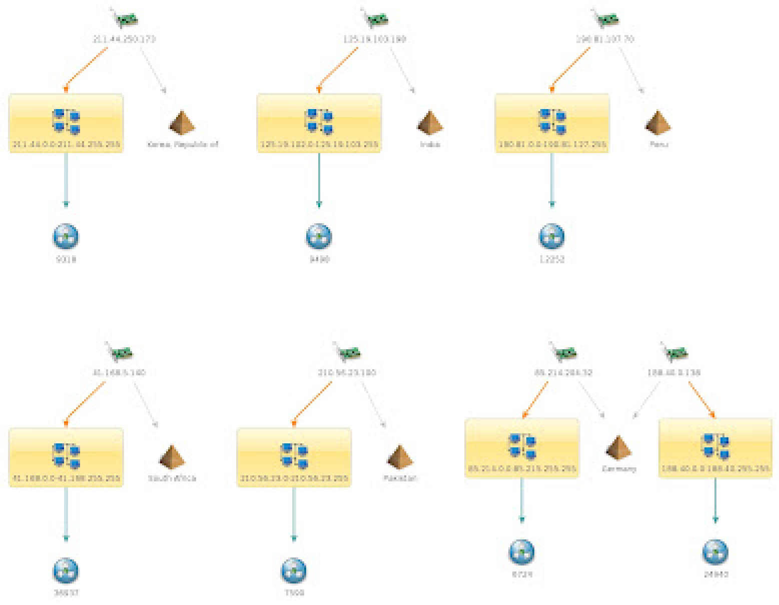


```
[redacted]@saturn:~/workspace/cridex/dump/vad/1484$ strings -af * | grep "/zb/v_01_a/in/"
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp:  /zb/v_01_a/in/ HTTP/1.1
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp:  /zb/v_01_a/in/ HTTP/1.1
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://41.168.5.140:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://41.168.5.140:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x00090000-0x0018ffff.dmp: ://125.19.103.198:8080/zb/v_01_a/in/
explorer.exe.23dea70.0x01cf0000-0x01deffff.dmp: http://188.40.0.138:8080/zb/v_01_a/in/cp.php
```

So via various string searches and some grepping in the VAD dump directory for PID1484 & PID1640 we find these IP addresses of interest:

- 190.81.107.70
- 41.168.5.140
- 85.214.204.32
- 210.56.23.100
- 211.44.250.173
- 125.19.103.198
- 188.40.0.138

Maltego lets me draw a pretty picture of the IPs, country of registration, and ASN.



Cridex IP addresses, ASN, and country of registration.

Doing some additional research, I noted that at one time or another, several domain names (now suspended) utilized all of the above listed Cridex IPs (except for 188.40.0.138). In fact, these domains each utilized the same 11 to 14 IP addresses, including the Cridex IPs for their DNS "A" records during their brief activity. Looking at the 'whois' for a sample of these domains shows an entirely different set of IPs used for their NS records... but I digress.

```
domain:       VALIDATORONMEE.RU
nserver:      ns1.validatoronmee.ru. 62.213.64.161
nserver:      ns2.validatoronmee.ru. 195.62.52.69
nserver:      ns3.validatoronmee.ru. 62.76.191.172
nserver:      ns4.validatoronmee.ru. 41.66.137.155
nserver:      ns5.validatoronmee.ru. 83.170.91.152
nserver:      ns6.validatoronmee.ru. 85.214.204.32
state:        REGISTERED, NOT DELEGATED, UNVERIFIED
person:       Private Person
registrar:    NAUNET-REG-RIPN
admin-contact: https://client.naunet.ru/c/whoiscontact
created:      2012.04.10
paid-till:    2013.04.10

domain:       POLUICENOTGO.RU
nserver:      ns1.poluicenotgo.ru. 62.76.41.3
nserver:      ns2.poluicenotgo.ru. 62.213.64.161
nserver:      ns3.poluicenotgo.ru. 195.88.242.10
```

```
nserver:      ns4.poluicenotgo.ru. 41.66.137.155
nserver:      ns5.poluicenotgo.ru. 83.170.91.152
nserver:      ns6.poluicenotgo.ru. 85.214.204.32
state:        REGISTERED, NOT DELEGATED, UNVERIFIED
person:       Private Person
registrar:    NAUNET-REG-RIPN
admin-contact: https://client.naunet.ru/c/whoiscontact
created:      2012.04.15
paid-till:    2013.04.15

domain:       VITALITYSOMER.RU
nserver:      ns1.vitalitysomer.ru. 62.213.64.161
nserver:      ns2.vitalitysomer.ru. 195.62.52.69
nserver:      ns3.vitalitysomer.ru. 62.76.191.172
nserver:      ns4.vitalitysomer.ru. 41.66.137.155
nserver:      ns5.vitalitysomer.ru. 83.170.91.152
nserver:      ns6.vitalitysomer.ru. 85.214.204.32
state:        REGISTERED, NOT DELEGATED, UNVERIFIED
person:       Private Person
registrar:    NAUNET-REG-RIPN
admin-contact: https://client.naunet.ru/c/whoiscontact
created:      2012.04.10
paid-till:    2013.04.10
```

There is much more that you can do with this Cridex memory dump. For example, you can use 'apihooks' on the two processes, then drop into 'volshell' and browse through the pages. You could find the loaded DLLs, or extract a process of interest.

For your added research, I've posted a link to the Cridex memory image below. I didn't extract other forensic objects for this sample, but as I mentioned in my last post, I plan to do that for other samples going forward.

Update 1 - August 5, 2012

In the comments section, Tamer Hassan posted a question referencing PID 1464. That PID is most likely a terminated process where 'psscan' didn't find any associated remnants. However it might be interesting to search for references to executable files. Since we know that PID 1464 was the parent to PID 1484, it's worth looking for registry artifacts typically used by malware. Volatility allows you to carve through the the registry that is resident in memory and display subkeys, values, and data. In this example, I looked for keys and values associated with ***"Software\Microsoft\Windows\CurrentVersion\Run"*** This is accomplished via the 'printkey' command:

```
python vol.py -f /home/ezio77/cridex.vmem --profile=WinXPSP2x86 printkey -K
"Software\Microsoft\Windows\CurrentVersion\Run"
```

Since 'printkey' will go through all hives, you will get multiple hits related to the key in your search. After displaying multiple hives each with a Last Update date of either 2012-04-12 or 2012-04-13, you'll see the following:

```
Registry: \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-07-22 02:31:51
Subkeys:
Values:
REG_SZ        KB00207877.exe  : (S) "C:\Documents and Settings\Robert\Application
Data\KB00207877.exe"
```

Perhaps KB00207877.exe was PID 1464? It's not clear via Volatility at this point, but it's most likely just a copy of the original with an updated registry key. Referring to Microsoft's encyclopedia entry for "Worm:Win32/Cridex.G", they reference:

subkey: HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Sets value: "KB<eight-digit number>.exe"
With data: "%AppData%\KB<eight-digit number>.exe"

Additionally, the VirusTotal analysis for this sample shows references to this naming convention as well. (Scroll to bottom and select "Additional Information")

In any case, it's good info for further analysis, including examining other registry hives.

Update 2 - August 7, 2012

Michael Ligh, was kind enough to drop me a note about the parent of 'explorer.exe'. Michael is one of the key contributors to the Volatility project, as well as one of the authors of the "Malware Analyst's Cookbook and DVD" . He referenced an excerpt from his book where it explains that the parent of 'explorer.exe' is 'userinit.exe', which upon completion, will terminate, leaving 'explorer.exe' without a parent. From the "Malware Analyst's Cookbook", pg 585:

> *Details aren't available for the process with Pid 1536 (which appears to have created explorer.exe). However, based on what you know about the boot sequence, Pid 1536 probably belonged to userinit.exe—but it has since exited. Winlogon.exe launches userinit.exe, which in turn launches explorer.exe. Once userinit.exe is finished, it terminates, leaving explorer.exe without a parent process. It is still possible to determine a process's parent, even after the parent exits, by looking at the _EPROCESS.InheritedFromUniqueProcessId field.*

Many thanks Michael!

----

cridex_memdump.zip (40MB)
