4,598 visible stars
(Skyandtelescope.com)

We’ll hunt through 26,479,553 ShimCache / AmCache entries
(and won’t even break a sweat)
Hunting 101

analytics

leads

qualify

triage

FP
IR

© Mandiant, a FireEye Company. All rights reserved. CONFIDENTIAL
Hunting 101

qualify

triage

FP

IR

analytics

Icu

© Mandiant, a FireEye Company. All rights reserved. CONFIDENTIAL
Hunting 102

\[ \text{#triagedleads} = \frac{\text{available resources}}{\text{triage cost/lead}} \]
Do we need a new tool?

- Enterprise-wide execution data is an extremely powerful dataset.
- But is often analyzed only superficially.
- We lack simple and efficient tools to process that dataset in an efficient and meaningful way.

```
$find ./Parsed -type file -iregex ".*shimcache.txt" -exec grep -Fi -C 5 "dropper.exe" {} \; > Output.txt
$find ./Parsed -type file -iregex ".*shimcache.txt" -exec grep -Eif ./searchTerms.txt {} \; > Output.txt
$find ./Parsed -type file -iregex ".*shimcache.txt" -exec grep -Ei "RARSFX[0-9]" {} | cut -d ',' -f 3 | uniq -c | sort -k1nr more
```

(No, that’s not efficient)
(yes, it is superficial)
Data Ingestion

- 11 x Ingestion modules available
  - CSV’s, Redline, raw System hives, raw AmCache hives, …
  - Ingest Class abstracts all the complexity, simply sub-class and you’ll have a new Ingest module in no time!
- Currently geared towards ShimCache and AmCache data.
- Future: generic execution sources (Process auditing, AppLocker, etc)
Search features

- Default set of "known bad" regex expressions included (hunting)
- Multiprocessing search module, *really fast regex searches.*
- Regularly running with a package of over 3K regexes.
- Easily extended!
Temporal Execution Correlation

- A high temporal execution correlation between file-A and file-B indicates that file-B is usually executed before/after file-A. Think dropper → payload or attacker toolchain, dump creds → RAR them up, etc.

- To measure how “strong” an execution correlation is we use a formula that exponentially decreases with distance (think force of gravity).

- Eventually the correlation strength becomes zero.
<table>
<thead>
<tr>
<th>LastModifiedDate</th>
<th>FilePath</th>
<th>File</th>
<th>Size</th>
<th>ExecFlag</th>
<th>Before</th>
<th>After</th>
<th>InvBond</th>
<th>TCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-11-20 21:29...</td>
<td>C:\windows\System32\netl.exe</td>
<td>N/A</td>
<td>True</td>
<td>593</td>
<td>1179</td>
<td>True</td>
<td>1900</td>
<td></td>
</tr>
<tr>
<td>2009-07-14 01:14...</td>
<td>C:\windows\System32\xcopy.exe</td>
<td>N/A</td>
<td>True</td>
<td>16</td>
<td>1071</td>
<td>True</td>
<td>2270</td>
<td></td>
</tr>
<tr>
<td>2009-07-14 01:14...</td>
<td>C:\windows\System32\Dwm.exe</td>
<td>N/A</td>
<td>True</td>
<td>436</td>
<td>2</td>
<td>True</td>
<td>2303</td>
<td></td>
</tr>
<tr>
<td>2015-04-10 02:42...</td>
<td>C:\Program Files\SylinkDrop.exe</td>
<td>N/A</td>
<td>True</td>
<td>200</td>
<td>49</td>
<td>True</td>
<td>2901</td>
<td></td>
</tr>
<tr>
<td>2003-11-29 05:32...</td>
<td>C:\KIX32.exe</td>
<td>N/A</td>
<td>True</td>
<td>9</td>
<td>590</td>
<td>True</td>
<td>1342</td>
<td></td>
</tr>
<tr>
<td>2010-10-12 09:24...</td>
<td>C:\Program Files\concentr.exe</td>
<td>N/A</td>
<td>True</td>
<td>203</td>
<td>75</td>
<td>-</td>
<td>2794</td>
<td></td>
</tr>
<tr>
<td>2013-08-21 04:31...</td>
<td>C:\Program Files\sftdce.exe</td>
<td>N/A</td>
<td>True</td>
<td>358</td>
<td>1</td>
<td>True</td>
<td>765</td>
<td></td>
</tr>
<tr>
<td>2015-06-26 00:30...</td>
<td>C:\windows\devco\devcon.exe</td>
<td>N/A</td>
<td>True</td>
<td>157</td>
<td>58</td>
<td>-</td>
<td>32203</td>
<td></td>
</tr>
<tr>
<td>2012-05-23 15:37...</td>
<td>C:\windows\Explorer.EXE</td>
<td>N/A</td>
<td>True</td>
<td>75</td>
<td>18</td>
<td>-</td>
<td>2353</td>
<td></td>
</tr>
<tr>
<td>2012-03-27 18:28...</td>
<td>C:\Program Files\wfcrun32.exe</td>
<td>N/A</td>
<td>True</td>
<td>196</td>
<td>49</td>
<td>-</td>
<td>2867</td>
<td></td>
</tr>
</tbody>
</table>

2017-04-01 21:00:18,076 INFO (*) Note that context AppCompat data is pulled from first match in DB as an example (dates, paths, sizes, of other correlating files with the same FileName could be different)
2017-04-01 21:00:18,077 INFO Done
<table>
<thead>
<tr>
<th>FileName(*)</th>
<th>Size</th>
<th>ExecFlag</th>
<th>Before</th>
<th>After</th>
<th>InvBond</th>
<th>TCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>net1.exe</td>
<td>N/A</td>
<td>True</td>
<td>593</td>
<td>1179</td>
<td>True</td>
<td>1900</td>
</tr>
<tr>
<td>xcopy.exe</td>
<td>N/A</td>
<td>True</td>
<td>16</td>
<td>1071</td>
<td>True</td>
<td>2270</td>
</tr>
<tr>
<td>Dwm.exe</td>
<td>N/A</td>
<td>True</td>
<td>436</td>
<td>2</td>
<td>True</td>
<td>2303</td>
</tr>
<tr>
<td>Sysln.exe</td>
<td>N/A</td>
<td>True</td>
<td>200</td>
<td>49</td>
<td>True</td>
<td>2901</td>
</tr>
<tr>
<td>KIX32.exe</td>
<td>9</td>
<td>True</td>
<td>590</td>
<td></td>
<td>True</td>
<td>1342</td>
</tr>
<tr>
<td>conce.exe</td>
<td>203</td>
<td>False</td>
<td>75</td>
<td></td>
<td>-</td>
<td>2794</td>
</tr>
<tr>
<td>sftd.exe</td>
<td>358</td>
<td>True</td>
<td>1</td>
<td></td>
<td>True</td>
<td>765</td>
</tr>
<tr>
<td>devcon.exe</td>
<td>N/A</td>
<td>True</td>
<td>436</td>
<td>2</td>
<td>True</td>
<td>2303</td>
</tr>
<tr>
<td>Explorer.EXE</td>
<td>N/A</td>
<td>True</td>
<td>15</td>
<td></td>
<td>True</td>
<td>1342</td>
</tr>
<tr>
<td>wfcrun32.exe</td>
<td>N/A</td>
<td>True</td>
<td>358</td>
<td>1</td>
<td>True</td>
<td>765</td>
</tr>
</tbody>
</table>
root@siftworkstation:/mnt/hgfs/appcompatprocessor# ./A
root@siftworkstation:/mnt/hgfs/appcompatprocessor# l
Time Stacking

\[
\frac{\#\text{Exec IN}}{\#\text{Exec OUT1} + \text{OUT2}}
\]

Everything that executed during the known attacker activity (IN)

Attacker toolkit file names will bubble up

Breach

2016

Attacker activity

2017

Everything that executed before OUT1

Everything that executed after OUT2
(Reconscan) + Fevil

- Reconnaissance sessions are detected based on a list of recon commands available in ./reconFiles.txt which the user can tweak if required.

- For every host all recon files are identified and based on temporal execution distance split into **recon ‘sessions’**. Recon session weight is calculated using the overall length or the recon session along the same weight formula used by tcorr.

- **Intuitively**: length of the session (# recon commands) and cohesiveness (are commands one after the other or spread apart?) will determine the overall recon session weight.

---

**Strong cohesiveness**

- Scrnsvr.scr
- Mimi64.exe
- dsquery.exe
- Net1.exe
- Net.exe
- hostname.exe
- whoami.exe

**Weak cohesiveness**

- dsquery.exe
- Scrnsvr.scr
- Net1.exe
- Net.exe
- Outlook.exe
- calc.exe
- notepad.exe
- hostname.exe
- Excel.exe
- shutdown.exe
- whoami.exe

© Mandiant, a FireEye Company. All rights reserved. CONFIDENTIAL
Reconsan + (Fevil)

- Step 1: Grab all files executed within or close to all recon sessions
- Step 2: Perform frequency analysis on file name.
- Step 3: Sort and triage!
- Future: Replace frequency analysis with Time Stacking
Closing remarks

• Hopefully AppCompatProcessor will:
  • improve your hunting capabilities
  • give you interesting ideas on how to improve other analytics
  • reduce your triaging cost / lead!
  • lower the entry barrier for all you crazy hunting ideas!

(looking fwd to these)
QA?

- [GitHub link](https://github.com/mbevilacqua/appcompatprocessor)