Towards Forensicator Pro

(Bringing a DevOps Mindset to DFIR to Produce an Assistive Toolchain - CADFIR)

Barry Anderson, Security Architect, Cisco Systems
@zendrag0n
Pipe Dream or Field of Dreams?

- What if, instead of obtaining actionable intelligence requiring a DFIR artisan to take hundreds of small, interrelated actions, we could follow a repeatable six-step process?
Six-step analysis process

1. Data is transferred onto SIFT workstation
2. SIFT user browses http://<SIFT-ip>/ffate/
3. SIFT user creates a case
4. Jenkins jobs “Find Evidence”
5. Jenkins transfers data from plaso to Elasticsearch

#DFIRSummit
What this talk is about

• The benefits of bringing a DevOps mindset to DFIR
• How to automate evidence processing using Jenkins
• Advantages of using the ELK stack (Elasticsearch, Logstash, Kibana)
• Forensicator FATE (ffate) a lightweight DFIR case manager
What this talk is not about

- “Nintendo Forensics” / “VB” Forensics
- Black box Automation
What is DevOps?

• Developers develop with Operations in mind
• Operations use the same tools and techniques as Developers
It can’t be that simple!

There is a lot of information on the Web, twitter (and IRC) about DevOps. Two places to start:

- [http://theagileadmin.com/what-is-devops/](http://theagileadmin.com/what-is-devops/)
Why should we do this?

• We need to move from artisans to engineers.

• We need to be able to specialize, so:

• We need to be able to collaborate.
Artisan vs Engineer

What’s the difference?

• Artisan – “a worker in a skilled trade, especially one that involves making things by hand.”
• Engineer – “a person trained and skilled in the design, construction, and use of engines or machines”

Both involve skill - the difference is – one approach scales.
Why automate?

• Not everybody is as smart as you are
• I’m not as smart as 4:30am as I am at 10:00am
• I’m not as smart when somebody’s yelling at me (while I try and complete a complex procedure)
• People make mistakes

“Under conditions of complexity, not only are checklists a help, they are required for success. There must always be room for judgment, but judgment aided – and even enhanced – by procedure.”

Atul Gawande, The Checklist Manifesto: How to Get Things Right
In case you were wondering

• The FATE in Forensicicator FATE is an acronym and stands for:

• “From Artisan To Engineer”
The moving parts

• The SIFT workstation
• The ELK stack (Elasticsearch, Logstash, Kibana)
• Jenkins
• PostgreSQL
• Forensicator FATE (ffate)
Resource Requirements

• Obviously more resources (CPU, RAM, Fiber-attached SAN) are better as forensics is resource-intensive!
• BUT you can do real work with 2 VM’s
• SSD!
Other requirements

• Firewall rules
• SMB naming
Putting the pieces together

• Start with the SIFT workstation, and ELK stack.

• Now add everything else:  
  $ git clone https://github.com/z3ndrag0n/forensicator-fate.git  $
  forensicator-fate/scripts/ffate-bundle-installer.sh  (This takes about 15 minutes, and you do have to hit the Enter key occasionally and tell the installer where to find ELK.)
What just happened?

The script we just ran installed:

- More network shares
- Jenkins and a whole library of jobs
- Forensicicator FATE
Sharing is caring
(or: A problem shared is a problem doubled)

In addition to the standard SIFT shares of /cases and /mnt, in order to share the information required for other tools and processes, the installer creates:

• /reverse
• /ioc
• /whitelist
• /blacklist
• /artifacts
Jenkins – the details

• Jenkins and plugins

• A library of Jenkins jobs, in four categories:
  – Filesystem Analysis
  – Memory Analysis
  – Find Evidence
  – Helper Tasks
Your tools, your processes

• This setup is extensible
• If you do want your tools and/or processes in the public domain, you either add them and request a (github) pull or ping me and I add them.
Jenkins – the Filesystem Analysis tasks

#DFIRSummit
### Jenkins – the Memory Analysis tasks

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>bulk_extractor_memory</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Volatility</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Volatility.0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Volatility.1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Volatility.2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Volatility.3</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Volatility.4</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Volatility.5</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Volatility.6</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Volatility.7</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Legend:**
- RSS for all
- RSS for failures
- RSS for just latest builds

DFIRSummit
Jenkins – Find Evidence

#DFIRSummit
Jenkins – the Helper tasks

#DFIRSummit
(Timeline) Data Visualisation with ELK

#DFIRSummit
Forensicator FATE

• With Jenkins, we’ve automated the evidence processing steps
• Can we take this even further, abstracting away even this complexity?
• Could a neophyte DFIR Engineer create new cases and “hit the Find Evidence button?”
• Better still, could an Examiner start the analysis process for the analyst?
<table>
<thead>
<tr>
<th>id</th>
<th>casename</th>
<th>memory image</th>
<th>disk image</th>
<th>disk name</th>
<th>timezone</th>
<th>volatility_profile</th>
<th>notes</th>
<th>keywords</th>
<th>actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>xp-tduong-10.3.58.7</td>
<td>/cases/xp-tduong-10.3.58.7</td>
<td>/cases/xp-tduong-10.3.58.7</td>
<td>C:</td>
<td>EST/EDT</td>
<td></td>
<td></td>
<td></td>
<td>View Evidence</td>
</tr>
<tr>
<td>6</td>
<td>win7-64-nfury-10.3.58.6</td>
<td>/cases/win7-64-nfury-10.3.58.6/win7-64-nfury-memory/win7-64-nfury-memory.raw.001</td>
<td>/cases/win7-64-nfury-10.3.58.6/win7-64-nfury-memory/win7-64-nfury-memory.raw.001</td>
<td>C:</td>
<td>EST/EDT</td>
<td></td>
<td></td>
<td></td>
<td>Find Evidence</td>
</tr>
<tr>
<td>7</td>
<td>sift</td>
<td>/cases/sift408pc-memory.img</td>
<td></td>
<td>C:</td>
<td></td>
<td>Australia/Melbourne</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Create new case

#DFIRSummit
Creating a new case

#DFIRSummit
<table>
<thead>
<tr>
<th>IOC</th>
<th>Size</th>
<th>Hash</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/loc/5477b392-e565-45c5-9c4-b4-f561d6daddec0ioc</td>
<td>5718</td>
<td>8a591ee34fcf3bedf39dde345</td>
<td>WARP (FAMILY)</td>
</tr>
<tr>
<td>/loc/7936b03c-49fd-4a3f-a89f-32b3d6abe046ioc</td>
<td>3661</td>
<td>905058ae76ba7d5b7e21d61603b3</td>
<td>Trojan-Tinba-Zusy</td>
</tr>
<tr>
<td>/loc/100c9f3c-970b-40a-8e0c-5772dd90a20ioc</td>
<td>7628</td>
<td>2aa945c35a306d263cd85a5219bedbd</td>
<td>ZEUS ANALYTICDNS.COM (BACKDOOR)</td>
</tr>
<tr>
<td>/loc/7bf353727-75sf-4eecc-8a5e-83435655f70ioc</td>
<td>2289</td>
<td>11a0160e6889f8f5dcd75ab8650c0</td>
<td>Backdoor-r57shell</td>
</tr>
<tr>
<td>/loc/ad521068-6f14-aab1-899c-11007a18ec73ioc</td>
<td>12814</td>
<td>43272f2ceca352f1e0100e8e1117802</td>
<td>WEB2C-DIV (FAMILY)</td>
</tr>
<tr>
<td>/loc/8698bb7fa02-46df-8e19-e50ce370fedioc</td>
<td>17173</td>
<td>6cd241d40a1558090ac66d591e6b4e9</td>
<td>COOKIEBAG (FAMILY)</td>
</tr>
<tr>
<td>/loc/d5e5f31c30-41ae-b381-c3c473040c39ioc</td>
<td>29323</td>
<td>a9cfeed8df6e47bc6da9a0220f20fc</td>
<td>STARSPOUND (FAMILY)</td>
</tr>
<tr>
<td>/loc/a461f381-8612-4c1a-8duc-6bcbaca02d80ioc</td>
<td>11581</td>
<td>788a416c6e19e6f56a789926c77846e</td>
<td>WEB2C-UGX (FAMILY)</td>
</tr>
<tr>
<td>/loc/25a50f37-ea8c-31a9-ac8d-4668df520dd1ioc</td>
<td>5893</td>
<td>b87af802a28e9f3d61261683768e75</td>
<td>FWS-Zbot.gen.x.Dximizer.ZIP+PWD-Mail</td>
</tr>
<tr>
<td>/loc/4d1ce5f-fe47-7ba4-be0e-81d54773a8aioc</td>
<td>8710</td>
<td>178b22e270c21823ad9409fe1f22d</td>
<td>WEB2C-AUSOV (FAMILY)</td>
</tr>
<tr>
<td>/loc/326c3286-1a3c-403f-3b7f-6f1f3183b1ioc</td>
<td>6867</td>
<td>47ac4fcd3d274062b175fca340a395b3</td>
<td>Morto</td>
</tr>
<tr>
<td>/loc/d624090a-adf6-46ee-a39c-649df5b8815ioc</td>
<td>16614</td>
<td>fd673a1d951a7317a8089470a9899f8</td>
<td>AURIGA (FAMILY)</td>
</tr>
<tr>
<td>/loc/326a675-6c98-4e42-8cb0-3dbf79517878f</td>
<td>15269</td>
<td>b224008fd70bd7b90ade346a59db2a3</td>
<td>Pnmocup DLL imports</td>
</tr>
<tr>
<td>/loc/d165316c-c0dd-499c-436e-c5392a5965c0ioc</td>
<td>31554</td>
<td>0f2b70f75595b77d528b1f6c19e8a354</td>
<td>TARSIS-ECLIPSE (FAMILY)</td>
</tr>
<tr>
<td>/loc/ece1846e-8cb3-44dc-a520-0da4866989ioc</td>
<td>7170</td>
<td>554ee5fa434d234702c00d776bc33</td>
<td>WECB-TOCK (FAMILY)</td>
</tr>
<tr>
<td>/loc/4ac26f60-f4c0-48b4-ba1f-1ad4ac9f36c0ioc</td>
<td>6845</td>
<td>6042854409f9649a62642f47ad9b81</td>
<td>SEASALT (FAMILY)</td>
</tr>
<tr>
<td>/loc/baa24c6a-a232-4919-b7e5-08c4809618ioc</td>
<td>75271</td>
<td>cdd3da050c5006a76a07b3d0a24b40</td>
<td>Sykipot/WyksolTrojan</td>
</tr>
<tr>
<td>/loc/86e98ec-7413-453b-932b-b55f957a6b0aio</td>
<td>16117</td>
<td>b7c78202c89e01c8a3ab37b1d11808</td>
<td>TABMSGSQL (FAMILY)</td>
</tr>
<tr>
<td>/loc/5add2090-312b-4628-a4f8-c4221ca2c9aio</td>
<td>9015</td>
<td>95d21b5c882693e3aa6b5d379d95f</td>
<td>DUQU</td>
</tr>
<tr>
<td>/loc/8d23e0a-a059-45a4-b416-67e4d600944fic</td>
<td>258764</td>
<td>110b05d5616f2d2666937226b1f6d70</td>
<td>Appendix E - APT1 File Hashes</td>
</tr>
<tr>
<td>/loc/1623e44-7016-46bb-30b2-077a75b8be81ioc</td>
<td>11493</td>
<td>4662c59e35c381c830e63895c57</td>
<td>SHYLOCK</td>
</tr>
<tr>
<td>/loc/35ac9207-155e-4272-8dc0-d9f88686ca5ioc</td>
<td>2935</td>
<td>5823476d98282a5e9a3a5a65a66c9</td>
<td>Debugger persistence mechanism possibly abused by malware</td>
</tr>
</tbody>
</table>

**Indicators of Compromise**

#DFIRSummit
Reverse Engineering

#DFIRSummit
“How did the binary get in the RE share?”

• Short answer: I put it there.

• Slightly longer answer: I put it there after seeing it was interesting based on the results of some of the jobs run.
What’s next?

Coming for the DFIR Summit release this week are:

• openioc_scan support
• DFIR in the cloud – the Inception Jenkins job

#DFIRSummit
What about after that?

• More Automation
• Color
• “Packets Never Lie”
• Compartmentalization
More Automation

• Automation of the investigation beyond the evidence processing phase (e.g. IOC generation)
• Mac Forensics
• Linux Forensics
• Mobile Forensics
• Network Forensics
• one FindEvidence to rule them all
• Binary analysis, including AV scanning
Color

• The colorized supertimeline visualisation is a highly effective way of seeing patterns in large volumes of data quickly.

• While ELK is a superior solution for storing huge volumes of data, the colorised view of the timeline is a desirable feature for the Kibana dashboard.
“Packets Never Lie”

• Integrating evidence found on the Network with evidence found (or not) on the systems under investigation.
Compartmentalisation

• The described setup, (like the SIFT workstation itself) makes certain assumptions about the environment in which it is deployed.
• These assumptions may no longer hold as cooperation is made easier and more widespread.
• An expert consulted for their expertise on one case, may not have clearance to view another case’s details.
• Likewise, an expert consulted for their particular expertise may not have the clearance to view all the details even of the case for which they are consulted.
Requesting features, Reporting bugs

• You can use the issues page on github:
  https://github.com/z3ndrag0n/forensicator-fate/issues
• You can tweet at me (@zendrag0n)
• You can tweet at the project (@ffate_project)
Tracking progress

• Watch the project on github:
  https://github.com/z3ndrag0n/forensicator-fate
• Follow me on twitter (@zendragon)
• Follow the project on twitter (@ffate_project)
Acknowledgements

• Rob Lee
• Ovie Carroll and Brett Padres
• Bianca Munoz-Greco
• Chris Walker
• You