Distributed Forensics

Across Time and Space

Google Incident Response

Introductions

- Johan Berggren <---> Timesketch core dev
- Daniel White <---> Plaso core dev
- Aaron Peterson <---> Turbinia core dev
- Thomas Chopitea <---> dfTimewolf core dev
- Brandon Chalk <---> Incident Response
- Tri Ngo <---> Detection & Response

Agenda

Today's lesson will cover the following tools for your investigation

- **Plaso**
  - An engaging exercise
- **Timesketch**
  - An more intriguing exercise
- **GRR**
- **dfTimewolf**
  - The ultimate exercise™

Ground Rules ... <YAWN>

- Ask questions. We ***will probably*** have answers.

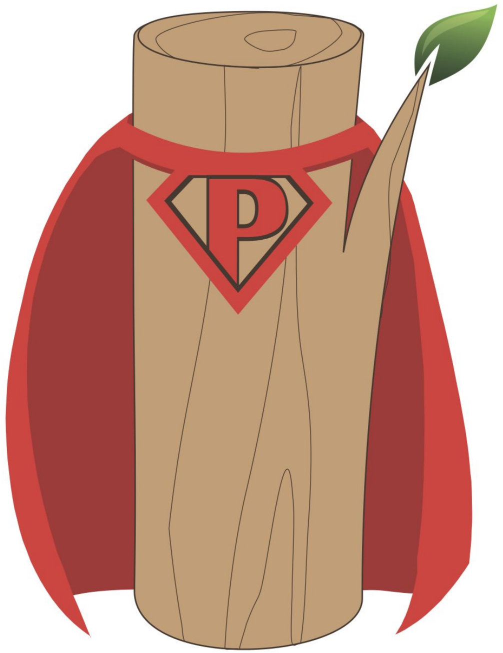
- Team up with other participants. Investigating in pairs can help.

- Don't work ahead on exercises. You'll have enough time to go through them all.

- Make sure to use the cheatsheets, they'll save you a bunch of time

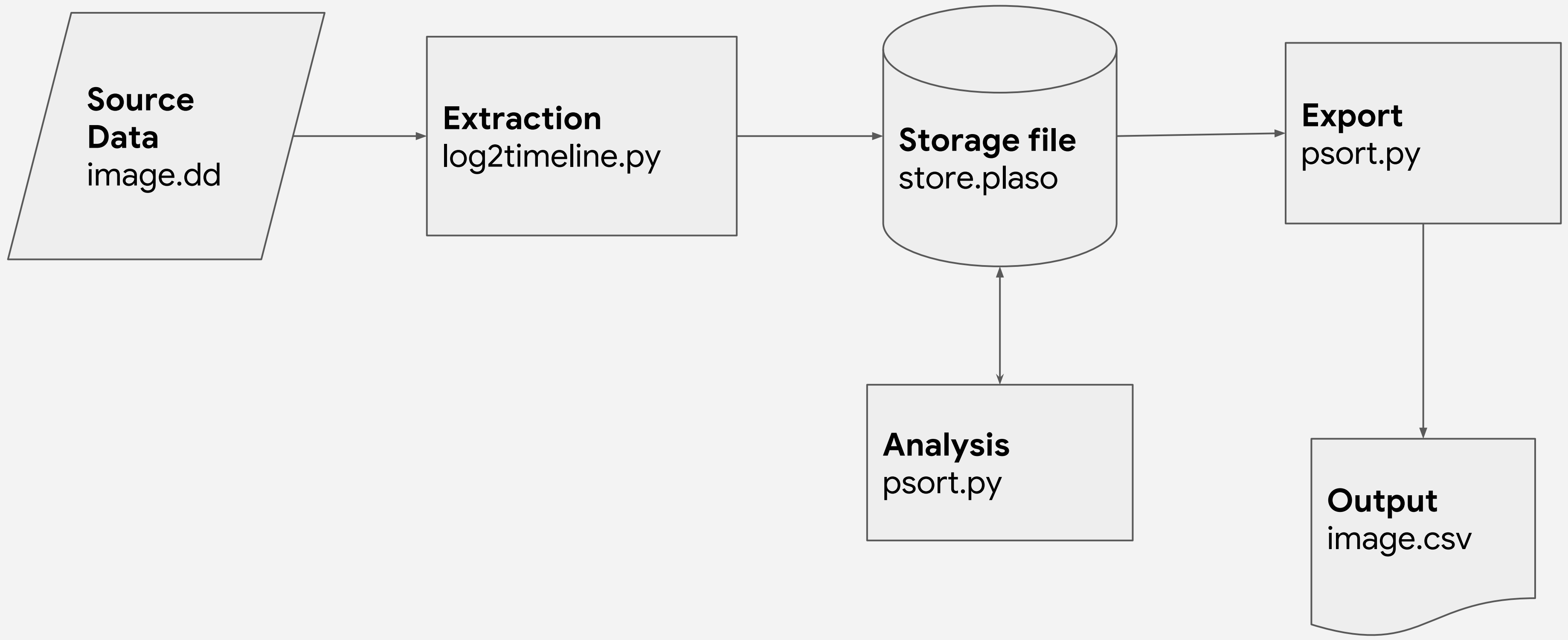
- Please poke around and experiment with the tools. And if you find a bug, let us know!

Plaso



Ye old logs

Plaso


Source Data
image.dd
Extraction
log2timeline.py
Storage file
store.plaso
Export
psort.py
Analysis
psort.py
Output
image.csv

log2timeline.py .. Event Extraction

- `$ log2timeline.py output.plaso /path/to/input/evidence`

- `$ log2timeline.py --help | less`

- Processing can take a long time
  - Less if it's a filtered extraction

- Specific options
  - `--parsers PARSER_LIST`
  - `--partitions PARTITIONS`
  - `--vss_stores VSS_STORES`

log2timeline.py .. Filtering

- **File filters**
  - Eg. `-f /usr/share/plaso/filter_windows.txt`
  - Default "triage" filter files
    - `/usr/share/plaso/filter*.txt`
  - Format: https://github.com/log2timeline/plaso/wiki/Collection-Filters

- **Artifact filters**
  - Eg. `--artifact_filters WindowsSystemRegistryFiles`
  - Definitions from Forensic Artifacts project

Forensic Artifacts

Machine readable repository of artifact definitions.

```yaml
name: UsersShellHistory
doc: Common unix user shell history files.
sources:
- type: FILE
  attributes:
    paths:
      - '/%%users.homedir%%/.bash_history'
      - '/%%users.homedir%%/.sh_history'
      - '/%%users.homedir%%/.zhistory'
      - '/%%users.homedir%%/.zsh_history'
labels: [History Files]
supported_os: [Linux, Darwin]
```

```yaml
name: AllUsersShellHistory
doc: Common shell history files for root and users.
sources:
- type: ARTIFACT_GROUP
  attributes:
    names:
      - UsersShellHistory
      - RootUserShellHistory
labels: [History Files]
supported_os: [Linux, Darwin]
```

psort.py .. Exporting

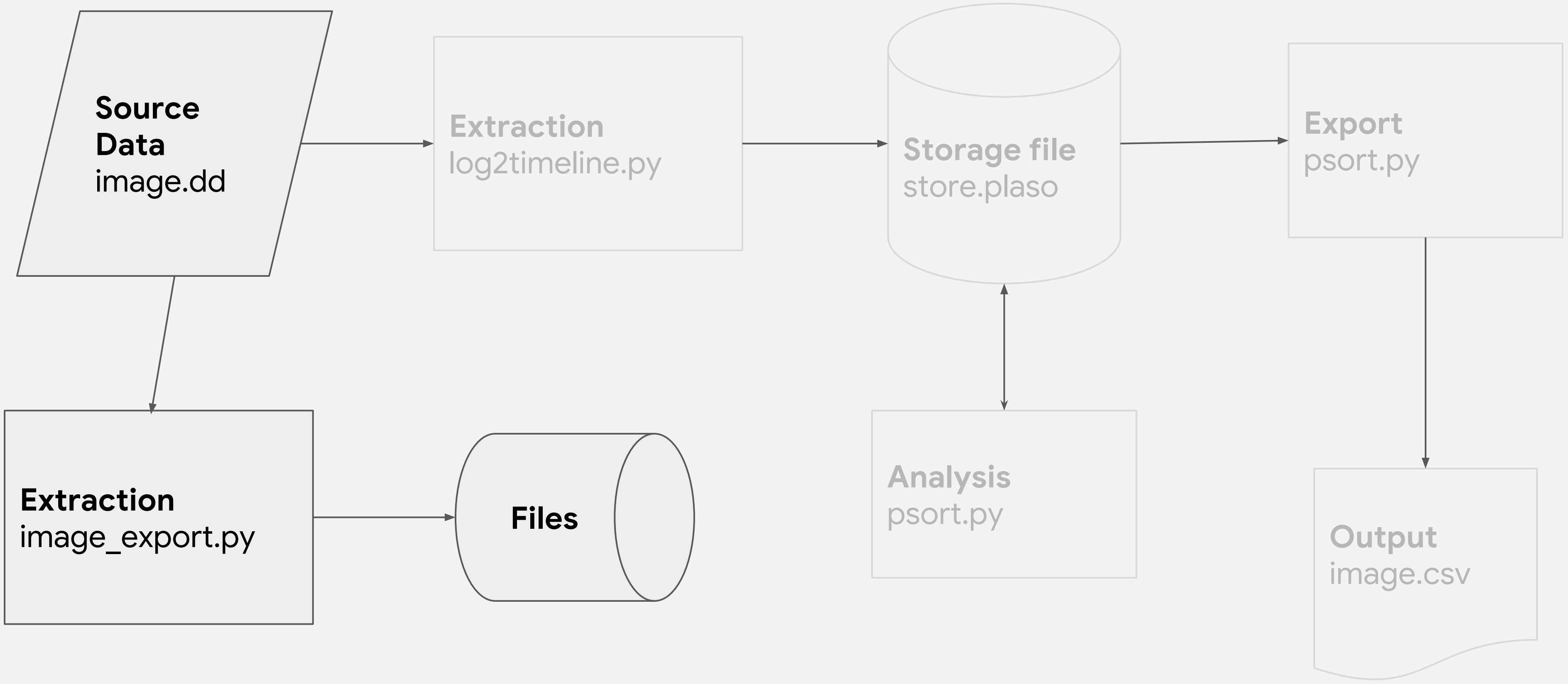
- `$ psort.py -w output.log output.plaso`
- `$ psort.py --help | less`
- `$ psort.py -o l2tcsv -w registrar.csv registrar.plaso`

- De-duplicates events

- Makes human readable
  - Expands Windows Event Log entries
  - Builds the "message" event field

- Specific options
  - `-o FORMAT`
  - `--additional_fields ADDITIONAL_FIELDS`

Plaso


Source Data
image.dd
Extraction
log2timeline.py
Storage file
store.plaso
Export
psort.py
Extraction
image_export.py
Files
Analysis
psort.py
Output
image.csv

image_export.py .. Exporting

- `$ image_export.py -w /tmp/export --names=NTUSER.DAT registrar.dd`
- `$ image_export.py --help | less`

- Exports files from source data
  - VSS

- Specific options
  - `-f FILE_FILTER`
  - `--names NAMES`
  - `--signatures IDENTIFIERS`

Bonus Features !!

- `psteal.py`
  - Plaso express
  - Runs log2timeline.py, then psort.py
- `log2timeline.py`
  - Hashing
  - Yara
- `psort.py`
  - Analysis plugins

Welcome to CFA

CYBER FORENSIC AFFORDANCES

ALL CYBER, ALL THE TIME

Time to Analyze

- SSH to your machine
  - Passphrase is "**workshop**"
  - Login with **`analyst##@<IP>`**
- Tools are pre-installed
- Source data is on a read-only disk at /mnt/case_data_readonly
  - Make local copies to work from if you need to
- Use screen/tmux
- Please don't submit artifacts to Virustotal or other online malware or network analysis service

Action Time! .. Ahmed's Request

- Generate a triage storage file and CSV output from the “registrar” image
  - Image is at `/mnt/case_data_readonly/images/registrar.dd`
- Export the malicious file “freedom_trebuchet.exe” from the registrar image
- **BONUS:** How did this malicious file come to be on the machine?

Action Time! .. Tip 1

- Generate a triage storage file and CSV output from the “registrar” image
  - Image is at `/mnt/case_data_readonly/images/registrar.dd`
  - Command line is something like:
    - **log2timeline.py --partition 2 -f /usr/share/plaso/filter_windows.txt ~/registrar.plaso /mnt/case_data_readonly/images/registrar.dd**
  - And then:
    - **psort.py -o l2tcsv -w registrar.csv registrar.plaso**

- Export the malicious file “freedom_trebuchet.exe” from the registrar image

- **BONUS:** How did this malicious file come to be on the machine?

Action Time! .. Tip 2

- Generate a triage storage file and CSV output from the “registrar” image
  - Image is at `/mnt/case_data/registrar.dd`
  - Command line is something like:
    - **log2timeline.py --partition 2 -f /usr/share/plaso/filter_windows.txt ~/registrar.plaso /mnt/case_data_readonly/images/registrar.dd**
  - And then:
    - **psort.py -o l2tcsv -w registrar.csv registrar.plaso**

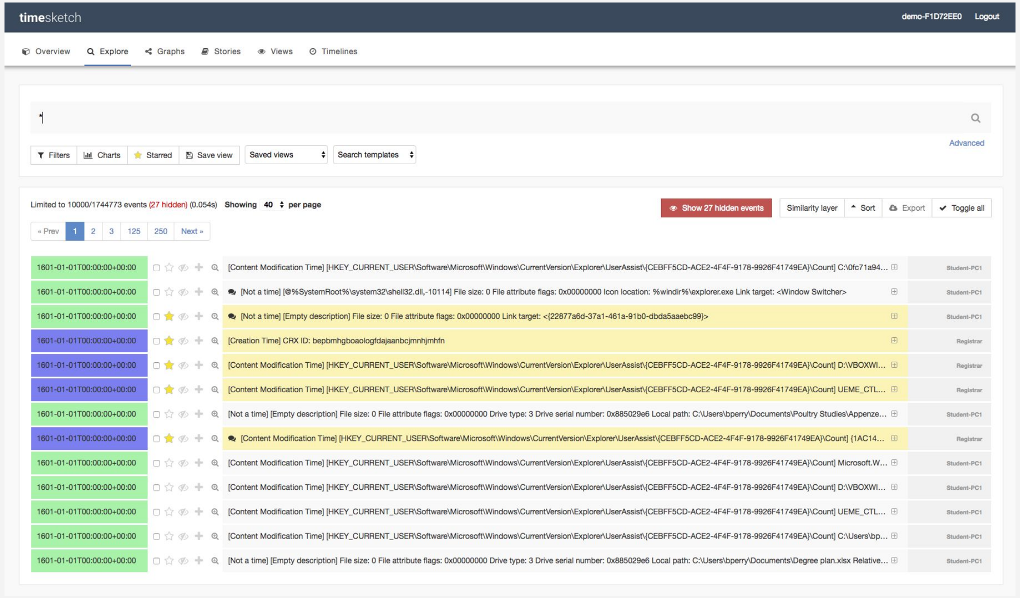
- Export the malicious file “freedom_trebuchet.exe” from the registrar image
  - Command line is:
    - image_export.py -w /tmp/export --names=freedom_trebuchet.exe /mnt/case_data_readonly/images/registrar.dd
    - File was stored in /Windows/AppPatch/Shared

- **BONUS:** How did this malicious file come to be on the machine?

Timesketch

Analyze Timelines

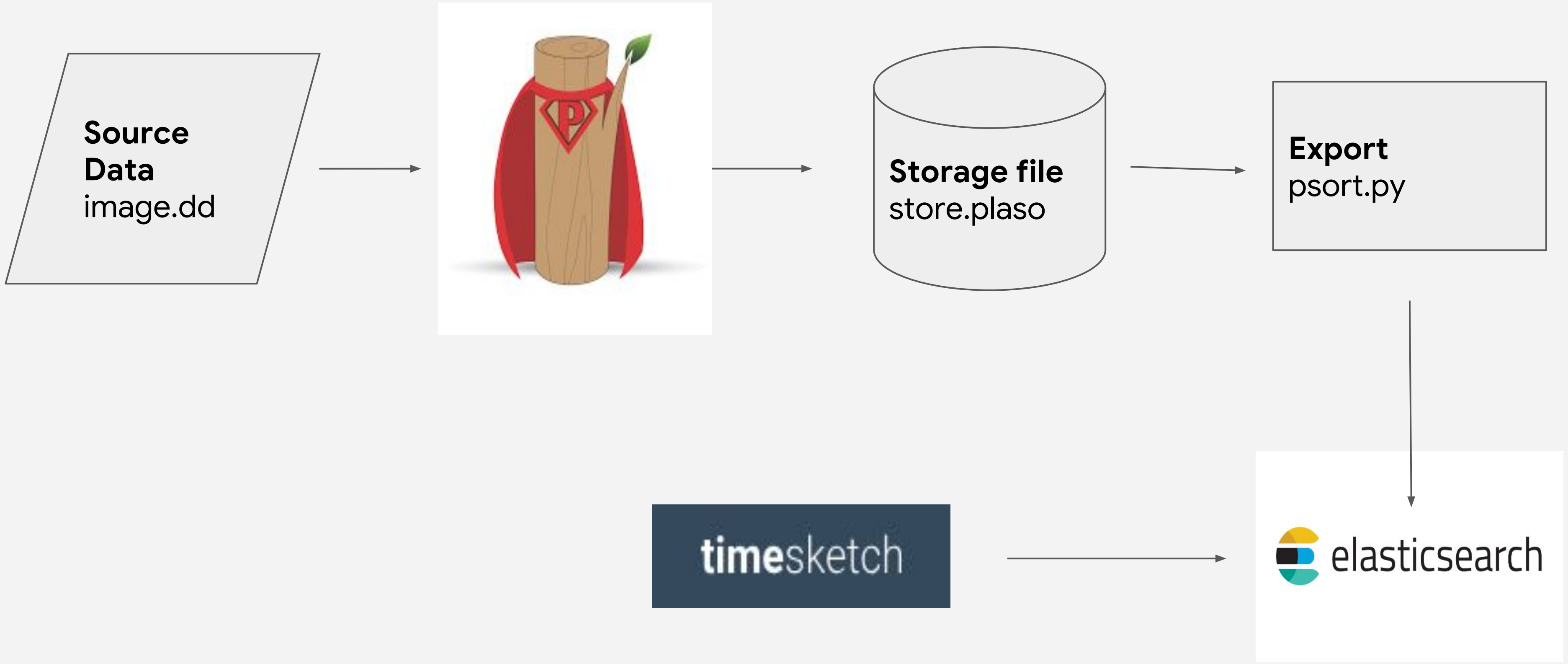
- Analysis frontend for timelines (e.g. Plaso)
- Evolution of `sed|grep|awk`
- Full text search using Elasticsearch query language
- Designed around collaboration
- Multi-user, multi-timeline and multi-case



Limited to 10000/1744773 events (27 hidden) (0.054s) Showing 40 per page

1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] C:\0fc71a94...	Student-PC1
1601-01-01T00:00:00+00:00	[Not a time] [@%SystemRoot%\system32\shell32.dll,-10114] File size: 0 File attribute flags: 0x00000000 Icon location: %windir%\explorer.exe Link target: <Window Switcher>	Student-PC1
1601-01-01T00:00:00+00:00	[Not a time] [Empty description] File size: 0 File attribute flags: 0x00000000 Link target: <{22877a6d-37a1-461a-91b0-dbda5aaebc99}>	Student-PC1
1601-01-01T00:00:00+00:00	[Creation Time] CRX ID: bepbmhgboaologfdajaanbcjmnhjmhfn	Registrar
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] D:\VBOXWI...	Registrar
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] UEME_CTL...	Registrar
1601-01-01T00:00:00+00:00	[Not a time] [Empty description] File size: 0 File attribute flags: 0x00000000 Drive type: 3 Drive serial number: 0x885029e6 Local path: C:\Users\bperry\Documents\Poultry Studies\Appenze...	Student-PC1
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] {1AC14...	Registrar
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] Microsoft.W...	Student-PC1
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] D:\VBOXWI...	Student-PC1
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] UEME_CTL...	Student-PC1
1601-01-01T00:00:00+00:00	[Content Modification Time] [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count] C:\Users\bp...	Student-PC1
1601-01-01T00:00:00+00:00	[Not a time] [Empty description] File size: 0 File attribute flags: 0x00000000 Drive type: 3 Drive serial number: 0x885029e6 Local path: C:\Users\bperry\Documents\Degree plan.xlsx Relative...	Student-PC1

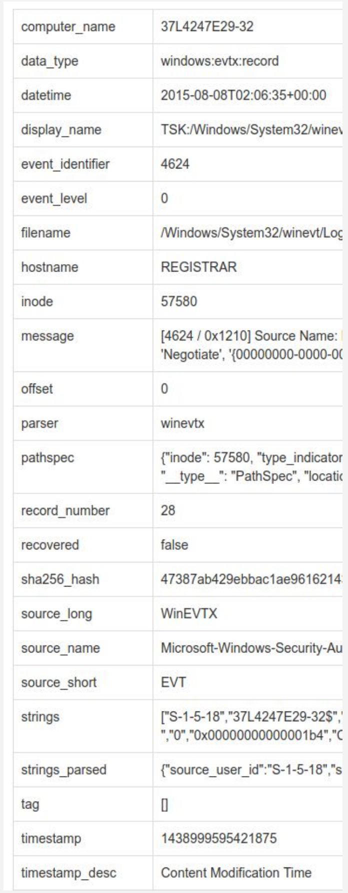
Timesketch


Source Data
image.dd
Storage file
store.plaso
Export
psort.py
timesketch
elasticsearch

Timesketch 101

- An investigation is called a sketch.
- A timeline is a collection of events from a source.
- A sketch have one or more timelines
- You search across one or more timelines
- Query language is Elasticsearch query string format or full DSL
- All fields from Plaso are searchable
    - E.g: `data_type:”windows:evtx:event” AND foobar`
- You can save searches and you can load pre canned searches from search templates to get you started

Anatomy of an Event

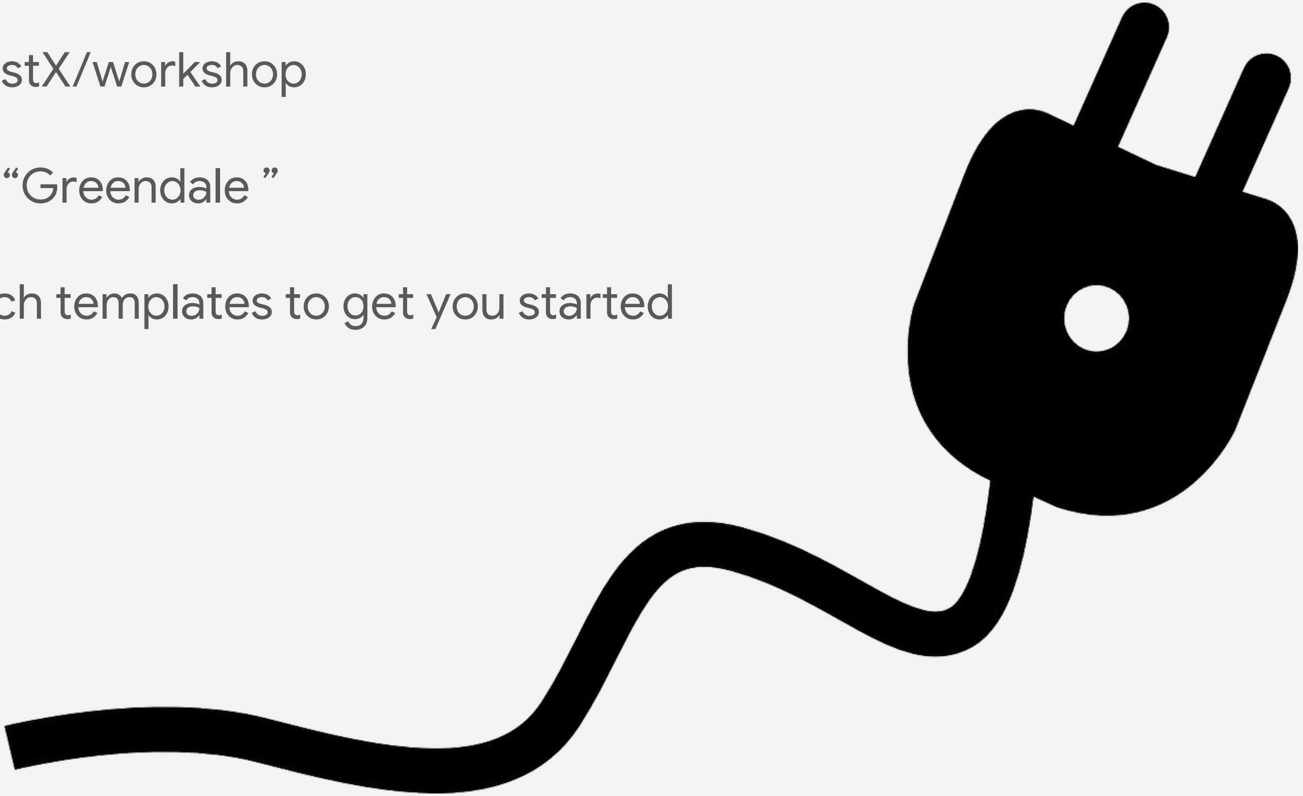


computer_name	37L4247E29-32
data_type	windows:evtx:record
datetime	2015-08-08T02:06:35+00:00
display_name	TSK:/Windows/System32/winev
event_identifier	4624
event_level	0
filename	/Windows/System32/winevt/Log
hostname	REGISTRAR
inode	57580
message	[4624 / 0x1210] Source Name: 'Negotiate', '{00000000-0000-0
offset	0
parser	winevtx
pathspec	{"inode": 57580, "type_indicator "__type__": "PathSpec", "locati
record_number	28
recovered	false
sha256_hash	47387ab429ebbac1ae9616214
source_long	WinEVTX
source_name	Microsoft-Windows-Security-Au
source_short	EVT
strings	["S-1-5-18","37L4247E29-32$"," ","0","0x0000000000001b4","C
strings_parsed	{"source_user_id":"S-1-5-18","s
tag	[]
timestamp	1438999595421875
timestamp_desc	Content Modification Time

- **data_type**
  - Indication of what sort of thing the event is
  - eg. `windows:evtx:record`
- **filename**
  - File the event was extracted from
  - eg. `/Windows/System32/winevt/Logs/Security.evtx`
- **event_identifier**
  - Example event-specific attribute
  - eg. `4624`
- **message**
  - Human readable summary of the event, generated from attributes by psort
  - eg. `[4624 / 0x1210] Source Name: Microsoft-Windows-Security-Auditing Strings: ....`

Connect to Timesketch

- https://timesketch.cyberforensicaffordances.club/
- Login with your analystX/workshop
- Our sketch is named “Greendale ”
- There are some Search templates to get you started



Action Time .. Investigate!

- How did the intruders get on to the registrar’s machine?
- How did Student-PC1 get compromised?
- Is there any other evidence of attacker activity you can find?

GRR

GRR
RAPID RESPONSE

gather all the things

GRR Overview

- Remote forensics tool
- Clients connect to a GRR server
- Users (you) interact with the server which handles interactions with clients
- Clients upload stuff (files, information) to the server
- Users download stuff (files, information) to analysis systems

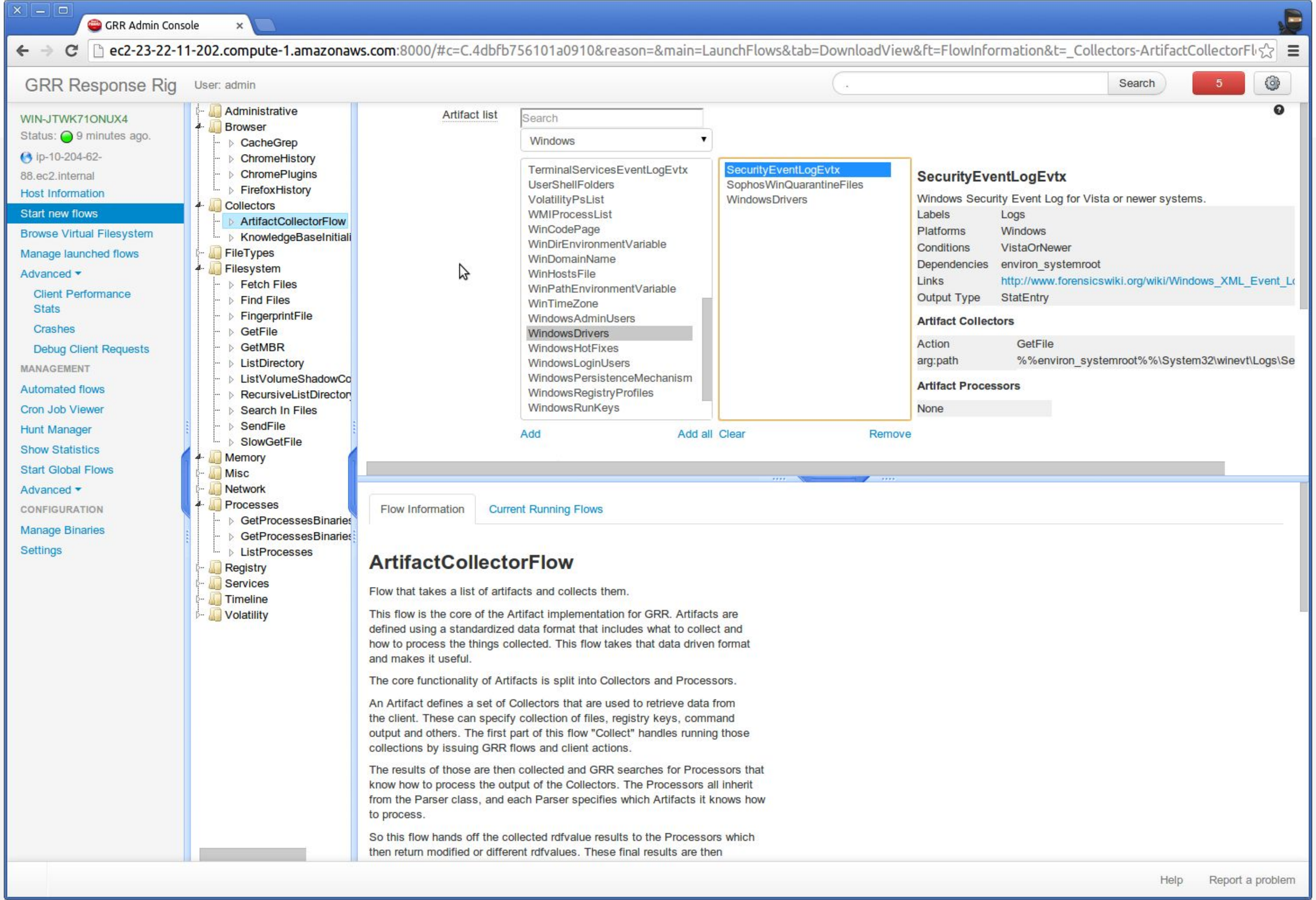
Flows and Hunts

- “Flows” are scheduled on clients to do collection
  - Upload a file
  - Upload an artifact
  - List open sockets

- “Hunts” run the same flow on many/all connected clients
  - Eg. Upload the contents of the `UserShellHistory` artifact

- Everything is asynchronous

- Manual interaction *isn’t all that scalable*



SecurityEventLogEvtx

Windows Security Event Log for Vista or newer systems.

Labels	Logs
Platforms	Windows
Conditions	VistaOrNewer
Dependencies	environ_systemroot
Links	http://www.forensicswiki.org/wiki/Windows_XML_Event_L
Output Type	StatEntry

Artifact Collectors

Action	GetFile
arg:path	%%environ_systemroot%%\System32\winevt\Logs\Se

Artifact Processors

None

ArtifactCollectorFlow

Flow that takes a list of artifacts and collects them.

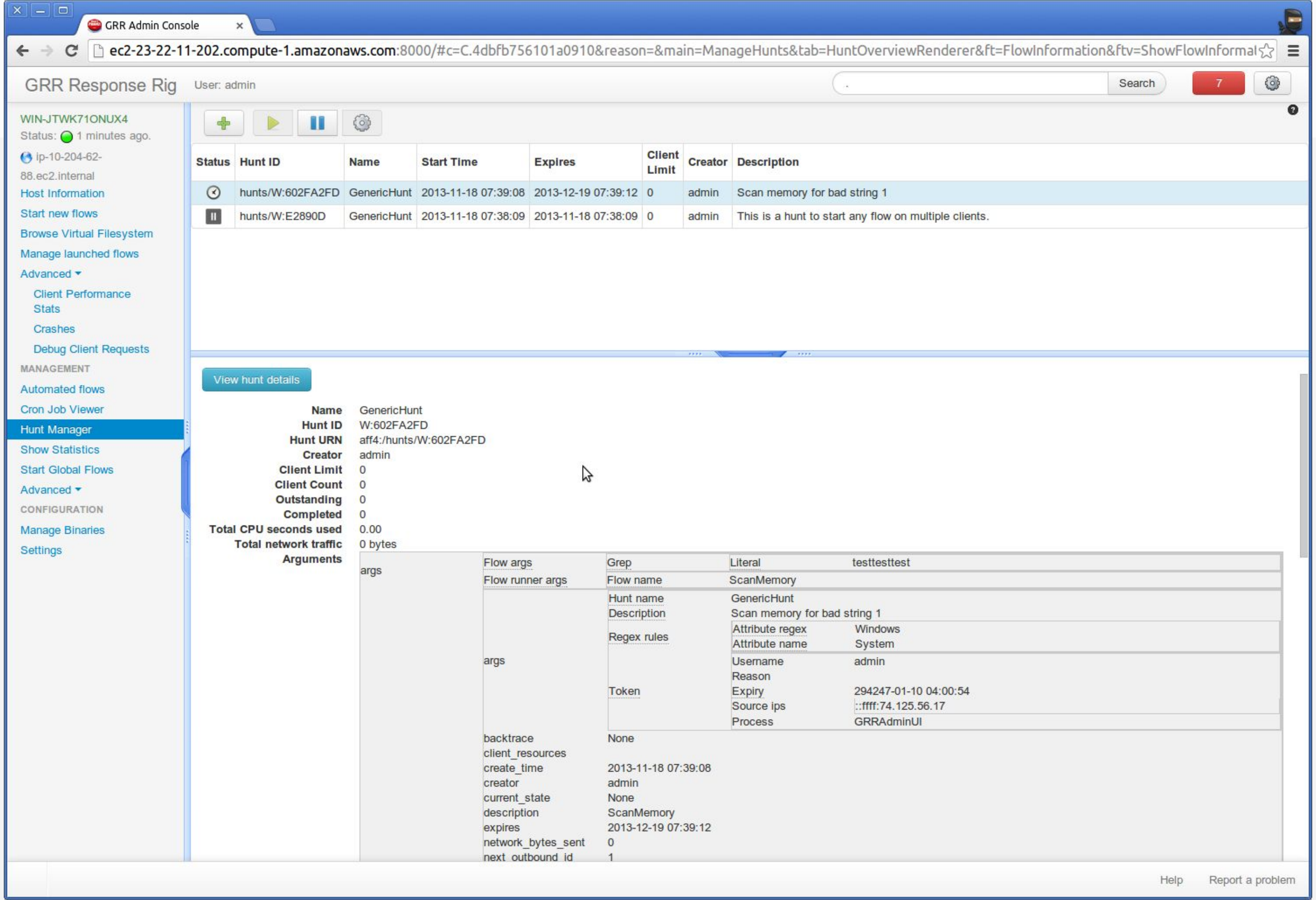
This flow is the core of the Artifact implementation for GRR. Artifacts are defined using a standardized data format that includes what to collect and how to process the things collected. This flow takes that data driven format and makes it useful.

The core functionality of Artifacts is split into Collectors and Processors.

An Artifact defines a set of Collectors that are used to retrieve data from the client. These can specify collection of files, registry keys, command output and others. The first part of this flow "Collect" handles running those collections by issuing GRR flows and client actions.

The results of those are then collected and GRR searches for Processors that know how to process the output of the Collectors. The Processors all inherit from the Parser class, and each Parser specifies which Artifacts it knows how to process.

So this flow hands off the collected rdfvalue results to the Processors which then return modified or different rdfvalues. These final results are then



GRR Admin Console

ec2-23-22-11-202.compute-1.amazonaws.com:8000/#c=C.4dbfb756101a0910&reason=&main=ManageHunts&tab=HuntOverviewRenderer&ft=FlowInformation&ftv=ShowFlowInformat

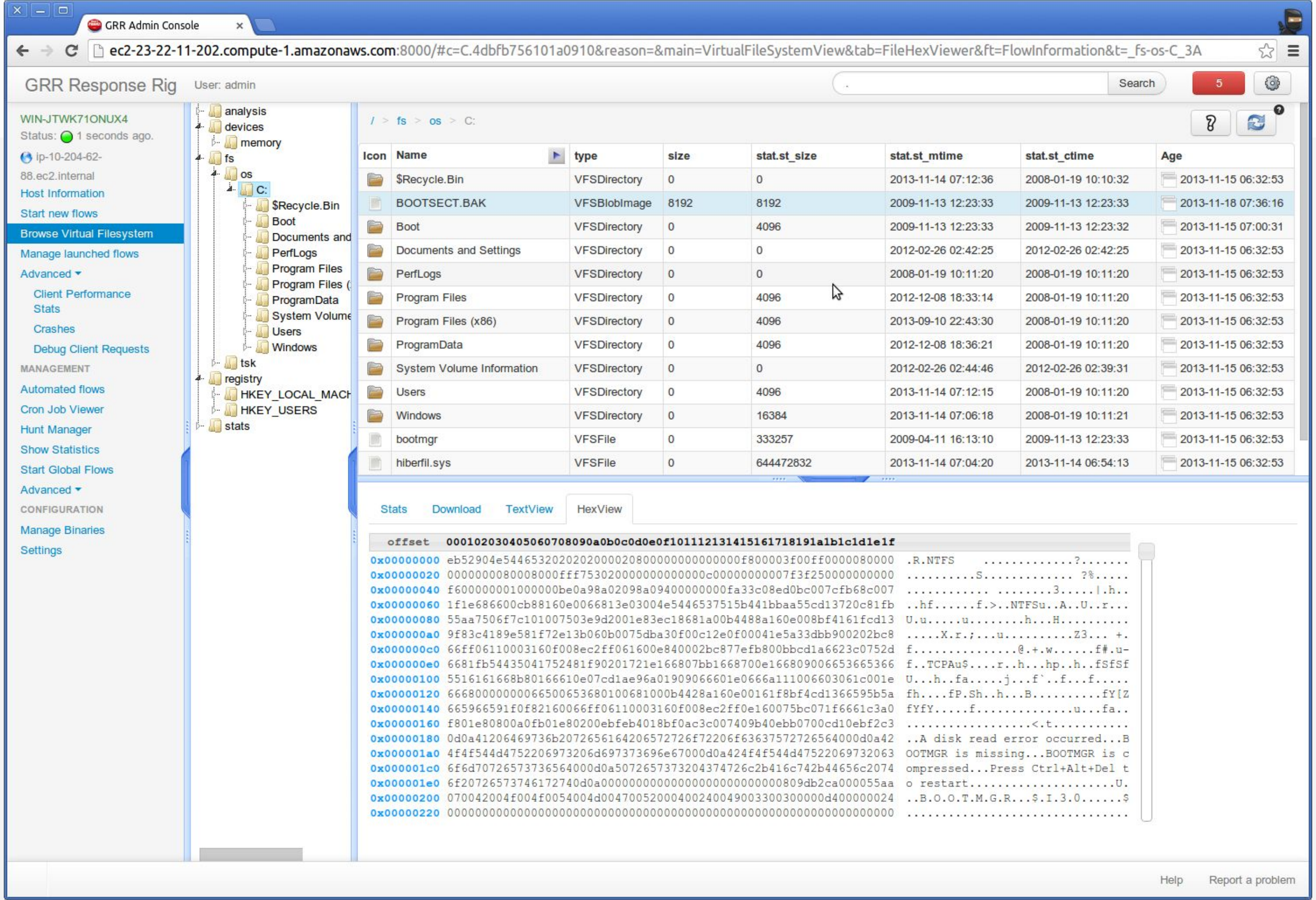
GRR Response Rig User: admin

Status	Hunt ID	Name	Start Time	Expires	Client Limit	Creator	Description
🕒	hunts/W:602FA2FD	GenericHunt	2013-11-18 07:39:08	2013-12-19 07:39:12	0	admin	Scan memory for bad string 1
⏸	hunts/W:E2890D	GenericHunt	2013-11-18 07:38:09	2013-11-18 07:38:09	0	admin	This is a hunt to start any flow on multiple clients.

Name	GenericHunt
Hunt ID	W:602FA2FD
Hunt URN	aff4:/hunts/W:602FA2FD
Creator	admin
Client Limit	0
Client Count	0
Outstanding	0
Completed	0
Total CPU seconds used	0.00
Total network traffic	0 bytes

Arguments

args	Flow args	Grep	Literal	testtesttest
	Flow runner args	Flow name	ScanMemory	
	args	Hunt name	GenericHunt	
		Description	Scan memory for bad string 1	
		Regex rules	Attribute regex	Windows
			Attribute name	System
		Token	Username	admin
			Reason	
			Expiry	294247-01-10 04:00:54
			Source ips	::ffff:74.125.56.17
			Process	GRRAdminUI
	backtrace	None		
	client_resources			
	create_time	2013-11-18 07:39:08		
	creator	admin		
	current_state	None		
	description	ScanMemory		
	expires	2013-12-19 07:39:12		
	network_bytes_sent	0		
	next_outbound_id	1		



Icon	Name	type	size	stat.st_size	stat.st_mtime	stat.st_ctime	Age
	$Recycle.Bin	VFSDirectory	0	0	2013-11-14 07:12:36	2008-01-19 10:10:32	2013-11-15 06:32:53
	BOOTSECT.BAK	VFSBlobImage	8192	8192	2009-11-13 12:23:33	2009-11-13 12:23:33	2013-11-18 07:36:16
	Boot	VFSDirectory	0	4096	2009-11-13 12:23:33	2009-11-13 12:23:32	2013-11-15 07:00:31
	Documents and Settings	VFSDirectory	0	0	2012-02-26 02:42:25	2012-02-26 02:42:25	2013-11-15 06:32:53
	PerfLogs	VFSDirectory	0	0	2008-01-19 10:11:20	2008-01-19 10:11:20	2013-11-15 06:32:53
	Program Files	VFSDirectory	0	4096	2012-12-08 18:33:14	2008-01-19 10:11:20	2013-11-15 06:32:53
	Program Files (x86)	VFSDirectory	0	4096	2013-09-10 22:43:30	2008-01-19 10:11:20	2013-11-15 06:32:53
	ProgramData	VFSDirectory	0	4096	2012-12-08 18:36:21	2008-01-19 10:11:20	2013-11-15 06:32:53
	System Volume Information	VFSDirectory	0	0	2012-02-26 02:44:46	2012-02-26 02:39:31	2013-11-15 06:32:53
	Users	VFSDirectory	0	4096	2013-11-14 07:12:15	2008-01-19 10:11:20	2013-11-15 06:32:53
	Windows	VFSDirectory	0	16384	2013-11-14 07:06:18	2008-01-19 10:11:21	2013-11-15 06:32:53
	bootmgr	VFSFile	0	333257	2009-04-11 16:13:10	2009-11-13 12:23:33	2013-11-15 06:32:53
	hiberfil.sys	VFSFile	0	644472832	2013-11-14 07:04:20	2013-11-14 06:54:13	2013-11-15 06:32:53

```
  offset   000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
0x00000000 eb52904e544653202020200000208000000000000000f800003f00ff0000080000  .R.NTFS    .............?.......
0x00000020 000000008000800fff75302000000000000c00000000007f3f250000000000000  ..........S............. ?%.....
0x00000040 f6000000001000000be0a98a02098a09400000000fa33c08ed0bc007cfb68c007  ........... ........3.....|.h..
0x00000060 1f1e686600cb88160e0066813e03004e5446537515b441bbaa55cd13720c81fb  ..hf......f.>..NTFSu..A..U..r...
0x00000080 55aa7506f7c101007503e9d2001e83ec18681a00b4488a160e008bf4161fcd13  U.u.....u........h...H..........
0x000000a0 9f83c4189e581f72e13b060b0075dba30f00c12e0f00041e5a33dbb900202bc8  .....X.r.;...u..........Z3... +.
0x000000c0 66ff06110003160f008ec2ff061600e840002bc877efb800bbcd1a6623c0752d  f..............@.+.w......f#.u-
0x000000e0 6681fb54435041752481f90201721e166807bb1668700e166809006653665366  f..TCPAu$....r..h...hp..h..fSfSf
0x00000100 5516161668b80166610e07cd1ae96a01909066601e0666a111006603061c001e  U...h..fa.....j...f`..f...f.....
0x00000120 666800000000665006536801006810000b4428a160e00161f8bf4cd1366595b5a  fh....fP.Sh..h...B..........fY[Z
0x00000140 665966591f0f82160066ff06110003160f008ec2ff0e160075bc071f6661c3a0  fYfY.....f..............u...fa..
0x00000160 f801e80800a0fb01e80200ebfeb4018bf0ac3c007409b40ebb0700cd10ebf2c3  ..................<.t...........
0x00000180 0d0a41206469736b2072656164206572726f72206f63637572726564000d0a42  ..A disk read error occurred...B
0x000001a0 4f4f544d4752206973206d697373696e67000d0a424f4f544d47522069732063  OOTMGR is missing...BOOTMGR is c
0x000001c0 6f6d7072657373656400d0a507265737320437472c2b416c742b44656c2074  ompressed...Press Ctrl+Alt+Del t
0x000001e0 6f2072657374617274000d0a0000000000000000000000000809db2ca000055aa  o restart....................U.
0x00000200 07004200 4f004f0054004d004700520000400240049003300300000d400000024  ..B.O.O.T.M.G.R...$.I.3.0......$
0x00000220 0000000000000000000000000000000000000000000000000000000000000000  ................................
```

dfTimewolf

because … wolves

dfTimewolf Overview

Goal: Automate manual, repetitive workflows as much as possible

- CLI tool acting as glue between different APIs and tools
- Uses "*modules*" (GRR, plaso, Timesketch, GCP...)
- Modules are chained through "*recipes*":
    - GRR ⟶ plaso ⟶ Timesketch
- Recipes define parameters for each module
    - Can be overridden through the CLI for one-offs

GRR & dfTimewolf

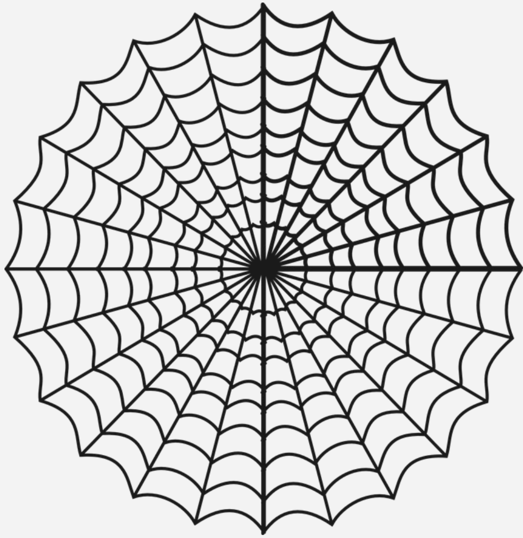
- Dftimewolf can easily launch GRR Hunts and Flows and collect results
- It can process the results with Plaso
- It can send a plaso output file directly to timesketch

This is exactly what the `grr_artifact_hosts` recipe does!

Let's collect some artifacts!

“Launch artifact collection on GRR hosts, collect results, process them through plaso, send results to Timesketch”

```
$ dftimewolf grr_artifact_hosts host1,host2
[--artifact_list, --sketch_id]
```



Greendale-as-a-Service

- As part of the “GaaS” program, Greendale has moved some of its infrastructure to the cloud
- Students can use their own GaaS instances (Ubuntu VMs) through SSH
- All GaaS instances run GRR, but there’s no other logging.
- Greendale’s SOC gets an alert that brute-force attacks were attempted on one of the GaaS servers, **greendale-webserver**

“Please investigate”

Forensicate !!

Using dftimewolf, collect evidence and answer these questions:

1. Was the bruteforce attack on **`greendale-webserver`** successful?
   a. Hint: Use the **`grr_artifact_hosts`** recipe to build a timeline from authentication logs
2. Identify the next computer to investigate
   a. Hint: Use Timesketch to identify which host the key is usually used from.
3. How were SSH keys exfiltrated from **`mccloud-gaas`**?
   a. Hint: Do a targeted GRR artifact collection with **`AllUsersShellHistory`**
4. ***Bonus**: Can you use dftimewolf to recover the SSH key archive?*
   a. Hint: Use the **`grr_fetch_files`** recipe.

Links & Contact

- **dfTimewolf**
  - https://github.com/log2timeline/dftimewolf
  - log2timeline-discuss@googlegroups.com
  - Apache License v2
- **GRR**
  - https://github.com/google/grr
  - grr-users@googlegroups.com
  - Apache License v2
- **Plaso**
  - https://github.com/log2timeline/plaso
  - log2timeline-discuss@googlegroups.com
  - Apache License v2
- **Timesketch**
  - https://github.com/google/timesketch
  - https://demo.timesketch.org
  - timesketch-dev@googlegroups.com
  - Apache License v2

Thanks from the cyber pony