DFIRTrack

KEEPING TRACK OF TRACES

How to track incidents online (fast)

/ AGENDA

- Introduction
- Installation
- Configuration and Customization
- Overview of Main Entities
- Import, Export and Manipulation Capabilities
- Automation Capabilities
- Roadmap, Feedback and Feature Discussion

Introduction

/ Problem: Spreadsheet of Doom

	A	B	C	D	E	F
1	System	Priority	Status	Analyst	Analyse Status	Note
2	SV-ESXI-18	low	clean	lionne	done	
3	SV-ESXI-19	low	clean	stuhli	done	
4	SV-ESXI-20	low	clean	lionne	done	
5	SV-ESXI-21	low	clean	lionne	done	
6	SV-DC-01	high	compromised	lionne	triage_ongoing	a.exe
7	SV-DC-02	high	compromised	lionne	done	a.exe
8	SV-DC-03	high	compromised	stuhli	done	a.exe
9	SV-LIN-270	low	clean	stuhli	done	
10	SV-lIN-182	medium	clean	stuhli	done	
11	SV-win-459	high	compromised	stuhli	analyses_ongoing	b.exe
12	sv-WIN-622	low	clean	stuhli	done	
13	sV-win-729	low	backup	stuhli	done	recovered
14	SV-WIN-610	medium	backup	stuhli	done	recovered
15	SV-WIN-387	medium	clean	stuhli	done	
16	Sv-WIN-986	medium	compromised	stuhli	recovery_ongoing	a.exe
17	SV-WIN-375	medium	compromised	stuhli	recovery_ongoing	a.exe
18	SV-WIN-754	low	compromised	lionne	triage_ongoing	b.exe
19	SV-WIN-320	low	clean	stuhli	done	
20	SV-WIN-78	low	backup	lionne	done	recovered
21	SV-WIN-48	low	backup	stuhli	waiting_for_customer	recovered
22	SV-WIN-91	low	backup	lionne	done	recovered
23	SV-WIN-877	low	backup	lionne	done	recovered
24	SV-WIN-41	high	clean	lionne	done	
25	SV-WIN-132	high	compromised	stuhli	triage_failed	a.exe, b.exe
26	SV-WIN-744	high	compromised	stuhli	edr_installed	a.exe
27	SV-WIN-948					

/ Solution: DFIRTrack

KEEPING TRACK OF TRACES

- Incident Response system-based status tracking for Incident Response Teams
- CERT case-based incident tracking possible, but not focus
- no client separation, so one instance per customer / project
- DFIRTrack is based on Python, Django and PostgreSQL

Installation

/ Installation

Manually

> https://github.com/dfirtrack/dfirtrack/wiki/Installation#minimal-installation

Ansible

> https://github.com/dfirtrack/dfirtrack/wiki/Ansible

Docker

> https://github.com/dfirtrack/dfirtrack/wiki/Docker
> Prod and dev available
> Customization using docker/(prod|dev)/.env

/ Workshop Setup

Setup Docker

> `cd docker/prod`
> `docker-compose up -d`
> `./setup_admin.sh`
> browse to `localhost` and accept the certificate warning

Configuration and Customization

/ Main config

Configuration possibilities

- Meaning of *General options*
- *Artifactstatus* and
- *Casestatus* and their status and time implications
- Side note: working *Artifactstatus* and handling of *Artifact* entities

/ User and groups

Implications

- No extensive user management, usage of Django capabilities
- Access via *Admin* page (so only for admin user)
- Within GUI every user is allowed to do everything (no read-only user)
  - Customer only gets data exports
- Small examples of usage
- Detailed example at the end if there is time left

Overview of Main Entities

/ System

What is a system?

- Could be anything that can be compromised (and may contain forensic artifacts)
- For us mainly client and server systems
- Might also be switches, firewall appliances, USB devices, etc.

/ Artifact

What is an artifact?

- Data or file of a system to be analyzed
  - Single files (malware sample, log file etc.)
  - Disk images, memory images etc.
  - File collections (triage images)
  - THOR scan
- One artifact requires a system, a system can contain many artifacts (database relation 1:n)
- We implement the artifact lifecycle utilizing the *Artifactstatus* in our workflow

/ Case

What is a case?

- It is possible to work case-based
- You need to create the case beforehand
- Case-based working is not in focus of DFIRTrack, therefore you probably will need other tools to track cases (for example "TheHive")
- We use DFIRTrack most of the time for one incident, which equals one case
  - Sometimes we identify a second incident during the analysis and create a new case to distinguish the incidents

/ Task

What is a task?

- Basic task management / tracking (there are better solutions)
  - Was implemented before the invention of *Artifacts*, so kind of deprecated (at least with our workflow)
- A task can be linked to other main entities
- We track only high-level tasks during an incident, everything else will be tracked utilizing the corresponding status (artifacts, systems)

/ Notes, Reportitems, Analystmemo and Documentation

The dream "automated report generation" (this is the way), work in progress though

- Idea: to prepare sections for a later report
- *Notes* are notes not linked to systems
  - e.g. "T1003.001 OS Credential Dumping: LSASS Memory" overview for all affected systems
- We can create *Reportitems* on per-system-basis
  - e.g. "T1560 Archive Collected Data" for a single system
- To view these items, *Documentation* is an overview page
  - Unfortunately, not as flexible and clear as we would like it to be
- *Analystmemos* are just internal notes for a system

/ Tags

Tags are self-explanatory?

- You can tag every entity
- Flexible structuring of entities, e.g.
  - Initial system (system identified by the customer at beginning of an incident)
  - Priorities
  - IoCs (one tag per identified IoC)
  - MITRE ATT&CK techniques / tactics / software
  - Scanner license created
  - Basically everything, that is not considered initially by DFIRTrack (like OS, location, etc.)

Import, Export and Manipulation Capabilities

/ System Creator and Modificator

Creator

> Fast creation of systems (obviously)
> Basic error correction implemented

Modificator

> Modification (of a subset) of systems
> Main goal is to avoid accidental changes

/ Artifact Creator

Slightly different than the System Creator

- Creates multiple *Artifacts* (also obvious)
- Basically, a concatenation of *Artifacttype* and *System* that results in *Artifacts*

/ Case, Task and Tag Creator

No example provided, because self-explanatory

- Does the same than the other *Creators*
- Just to understand the implications from a database point of view
  - Case and Tag creator combines existing *Case* or *Tag* entitites with existing *System* entities (creates the m:n relationship)
  - Task creator creates new *Task* entities (for existing *Tasknames*) and links to existing *System* entities

/ System Exporter and Artifact Exporter

System Exporter

- Creates CSV and XLS exports for Systems
- aka system list aka *Spreadsheet of Doom*
- Freely configurable (global and persistent)
- Either ad hoc download or per scheduled task (explanation follows later)
- Main reporting instance within our workflow

Artifact Exporter

- Only XLS export available
- Otherwise equivalent to System Exporter
- We use this file for requesting artifacts within our workflow
  - Export only specific *Artifactstatus*

/ System Importer

Imports systems via CSV

- For our workflow, the automatic import implements the interface to other tool results
- Freely configurable (global and persistent)
- Main goal is to avoid accidental changes, but differentiation: newly added systems vs. existing systems
  - The complex setting options result from this
  - This feature should be tested intensively for your own processes before you automate it and run it in production
- Either ad hoc upload via GUI or file system or per scheduled task (explanation follows later)

Automation Capabilities

/ Scheduled Tasks

Fully automated processes

- Uses Django Q
- Available for System Exporter (CSV and XLS), System Importer, Artifact Exporter and for the *Status* page
- Not very intuitive, therefore shortcut buttons exist
- Export path is defined in *Main Config*
- Post-processing (e.g. via cron job) is not part of DFIRTrack

/ Workflows

Semi-automated tasks

- Mainly used for *Artifacts* (but *Tasks* also possible)
- Idea: often the same standard *Artifacts* required per *System*
  - DFIRTrack GUI would require 2 single steps
- possible in several places
  - System creation
  - System detail
  - System Creator
  - System Modificator (this is a good place for adding – also single – artifacts for multiple systems in case they were forgotten, because it is more convenient than using the Artifact Creator)

Roadmap, Feedback and Feature Discussion

/ Roadmap, Feedback and Feature Discussion

Roadmap

- Lots of smaller issues
- Timesketch integration
- Improvement of documentation and report features
- Potentially MITRE ATT&CK tactics and techniques (not sure, currently represented via Tags in our workflow)
- We are aware that there are some legacy features, but whenever you remove them, you suddenly need them in the next project...

Feedback

- DFIRTrack is used by a small community
- Currently the feedback is very restrained
- Missing functionalities or bugs are mostly discovered during projects by us
- Sorry we are all no frontend developers, contribution is very welcome

How you can reach us

/ Who we are

Lionne-Jeremias Stangier

> E-mail lionne.stangier@sva.de
> Github lejurn
> Twitter lejurn
> Keybase lejurn

Mathias Stuhlmacher

> E-mail mathias.stuhlmacher@sva.de
> Github stuhli
> Twitter stuhlonsky
> Keybase stuhli

Backlog

/ User and groups

Example

- New user with extended permissions (between admin 'dfirtrack' and 'stuhli')
  - 'Staff status' is necessary to access admin page at all
  - Useful groups are 'dfirtrack' and 'django_q'

/ Status

Kind of overview page

- Was relevant during a project
- Also makes use of scheduled task

/ Assignments

Counter to pick up "tasks"

- Should be self-explanatory
- Independent from 'grab' and 'free' the entity status has to be updated

/ Filter

Work in progress

- At the moment the main entities can be filtered
  - Cases
  - Tags
  - User assignment
- The current status works, but we are not yet completely satisfied with it

/ Fixtures

Might support recurring base configurations during different deployments

- Uses Django feature to serialize database contents to files
- Files can be imported (de-serialized) during deployment
- Thinkable use cases are
  - Users and groups
  - Scheduled tasks
  - Status (for *Systems*, *Artifacts*, etc.)
  - *Tags*
  - Main Config