

CINCAN

Codifying your malware/artifact analysis workflows

or, Building shareable, repeatable & history preserving analysis pipelines using your favourite tools + git + containers

Co-financed by the Connecting Europe
Facility of the European Union



TRAFICOM

jamk.fi

Motivation (user point of view)

Enlightenment by listening to rants from analysts

- Solving fragility - tools don't always work in every env & every time
- Teaching and sharing* the workflow
- Making the workflow repeatable
- Automating stuff
- Sharing* results
 - repeatability & transparency

*Both within team & with collaborators or maybe even the world (open-source)



Motivation (developer POV)

Riding the big productivity & collaboration tech trends

- “Githubification of infosec” - leverage technologies like Github/Gitlab, Docker containers, CI/CD, collaborative ways of working
- Once a tool packaged in a container, it can be very easily integrated into different environments & systems - solve packaging & deployment
- Automation
 - Credibility & robustness of results by verifiability & repeatability & transparency
 - Continuous testing of changes & continuous proof of “it’s still working”



Background

EU funded project, ending in 6/2020

- Original schedule 2018-2019, extended to 2020-06
- Parties
 - National Cybersecurity Center (in Finnish Transport and Communications Agency)
 - Oulu University Secure Programming Group
 - JAMK University of Applied Sciences
 - Funding from EU's INEA agency, from CEF program (Connecting Europe Facility)
 - You (users / collaborators) - open source project!



CinCan products you can use now

Tools and packaging work

- Many DFIR tools packaged for docker (see <https://gitlab.com/CinCan/tools/>) and ready to run
- open source cli tool “cincan” available to easily run DFIR tools and handle data input/output
 - `cincan run cincan/apktool d samples/selendroid-test-app.apk -o samples/cincanTest`
- Many blog posts about how to use these tools (<https://cincan.io/blog/>)



CinCan products you can use now

Other products

- CI/CD based workflow automation solution
 - Automatically run pipeline of tools when new input data appears in Gitlab
 - Uses CinCan packaged tools & Concourse CI system & Gitlab
- open source cli tool “minion”
 - rule based multi step workflows, and example rulesets



PILOTING

European CERTs wanted, but also SoCs etc

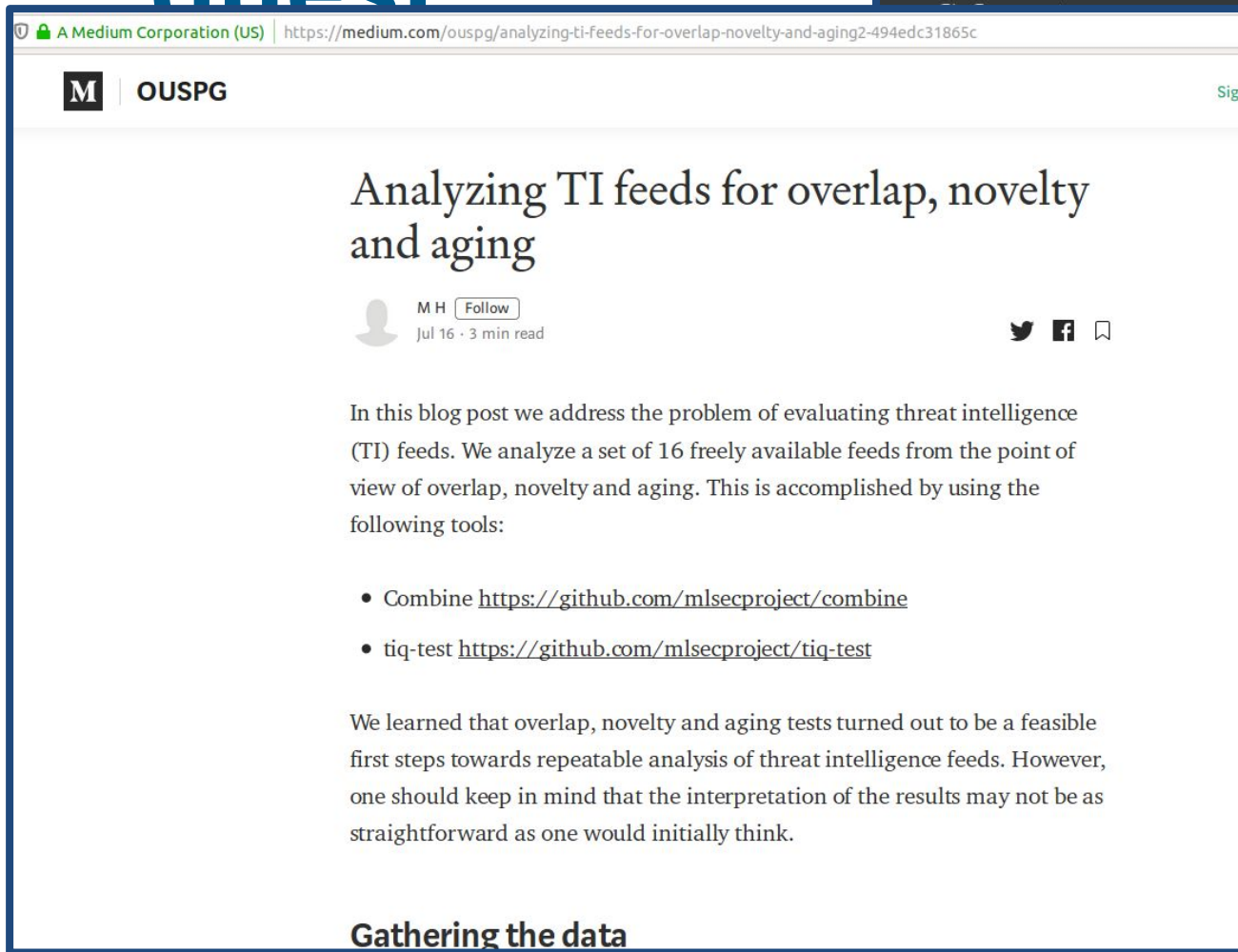
- We want as many users as possible
- Sign up, hear our 20 minute intro / pitch, try out some of our tools & give us feedback!



TRAFICOM

jamk.fi

Quality of Threat Intelligence side quest



M | OUSPG

Analyzing TI feeds for overlap, novelty and aging

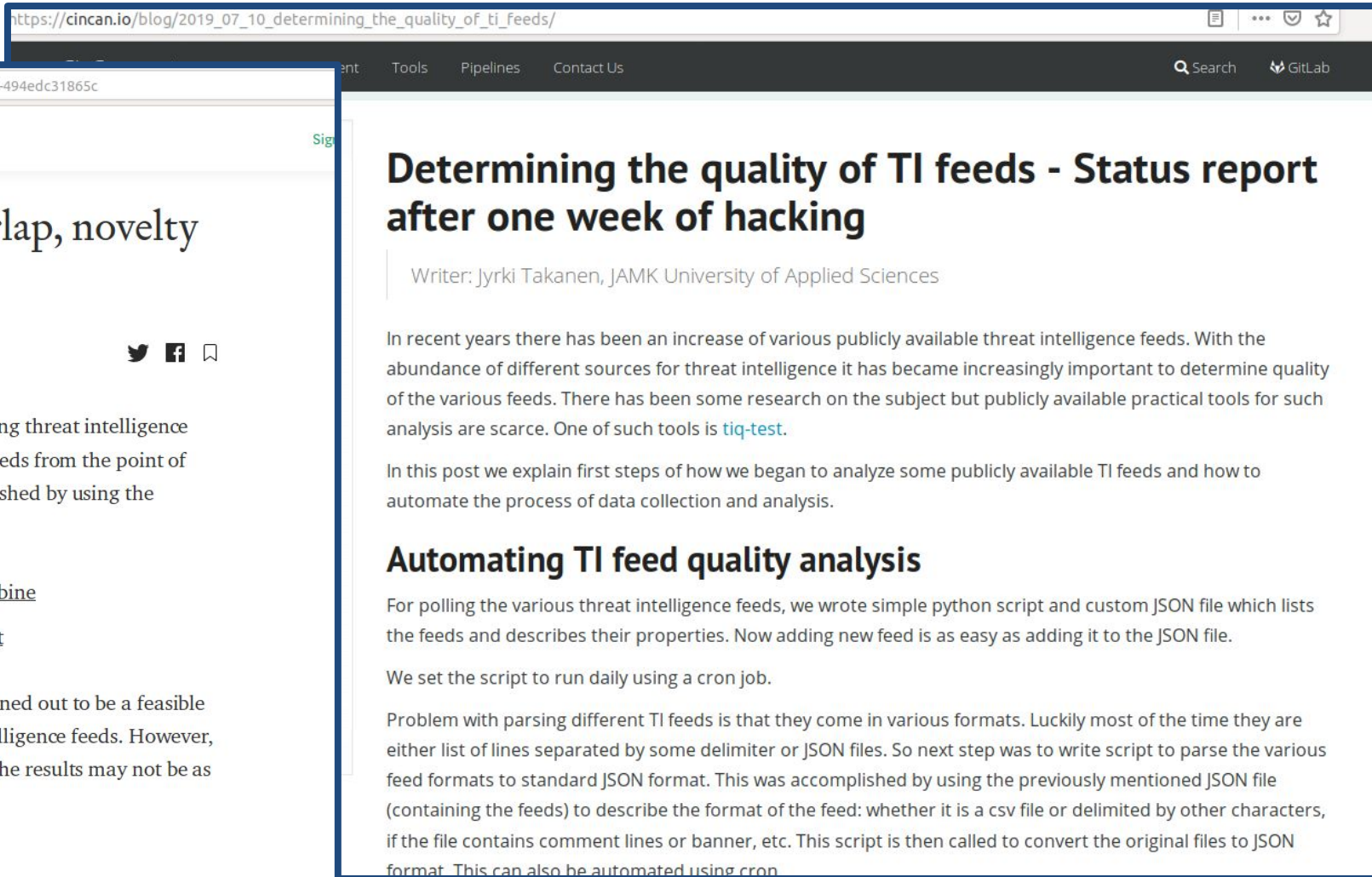
M H Follow
Jul 16 · 3 min read

In this blog post we address the problem of evaluating threat intelligence (TI) feeds. We analyze a set of 16 freely available feeds from the point of view of overlap, novelty and aging. This is accomplished by using the following tools:

- Combine <https://github.com/mlsecproject/combine>
- tiq-test <https://github.com/mlsecproject/tiq-test>

We learned that overlap, novelty and aging tests turned out to be a feasible first steps towards repeatable analysis of threat intelligence feeds. However, one should keep in mind that the interpretation of the results may not be as straightforward as one would initially think.

Gathering the data



https://cincan.io/blog/2019_07_10_determining_the_quality_of_ti_feeds/

Determining the quality of TI feeds - Status report after one week of hacking

Writer: Jyrki Takanen, JAMK University of Applied Sciences

In recent years there has been an increase of various publicly available threat intelligence feeds. With the abundance of different sources for threat intelligence it has become increasingly important to determine quality of the various feeds. There has been some research on the subject but publicly available practical tools for such analysis are scarce. One of such tools is [tiq-test](#).

In this post we explain first steps of how we began to analyze some publicly available TI feeds and how to automate the process of data collection and analysis.

Automating TI feed quality analysis

For polling the various threat intelligence feeds, we wrote simple python script and custom JSON file which lists the feeds and describes their properties. Now adding new feed is as easy as adding it to the JSON file.

We set the script to run daily using a cron job.

Problem with parsing different TI feeds is that they come in various formats. Luckily most of the time they are either list of lines separated by some delimiter or JSON files. So next step was to write script to parse the various feed formats to standard JSON format. This was accomplished by using the previously mentioned JSON file (containing the feeds) to describe the format of the feed: whether it is a csv file or delimited by other characters, if the file contains comment lines or banner, etc. This script is then called to convert the original files to JSON format. This can also be automated using cron.



TO FOLLOW/JOIN THE PROJECT

- <https://gitlab.com/cincan>
- <https://cincan.io>
- Twitter: @CinCanProject
- email contact cincan@traficom.fi

Co-financed by the Connecting Europe
Facility of the European Union



TRAFICOM

jamk.fi