virus BULLETIN Covering the global threat landscape

FAME – FRIENDLY MALWARE EVALUATION FRAMEWORK

CERT Société Générale, France

(This paper was presented as a last-minute Small Talk at VB2017.)

ABSTRACT

FAME is a recursive acronym meaning 'FAME Automates Malware Evaluation'. The FAME framework is intended to facilitate analysis of malicious files, leveraging as much knowledge as possible in order to speed up and automate end-to-end analysis of malware.

INTRODUCTION

At *CERT Société Générale*, which is a banking computer emergency response team based in France, we have our fair share of malware to analyse: banking malware that targets the bank's customers, and all kinds of malware that targets our users.

We realized that there were two main issues with the process of malware analysis within our team:

1. It was taking too long to complete an analysis. Let's consider the example of a banking trojan. Even if the analyst already recognized the malware family from the spam run, he still had to submit the sample to a sandbox, wait for the sandbox analysis to be completed, download a memory dump, extract the configuration from memory, and compare the configuration with our perimeter in order to determine if we were targeted. It the malware family was unknown, the process was even more complicated.
2. Not every analyst had the same level of malware analysis skills.

In order to address these problems, we developed FAME, a malware analysis pipeline that will chain the execution of modules in order to perform end-to-end analysis. In the best-case scenario, an analyst will submit a sample, and within a few minutes FAME will be able to recognize the malware family, extract its configuration, and identify how the malware is targeting our organization.

Not a sandbox

FAME is not a malware analysis sandbox, and it will not be very useful if it is not combined with one. It already has support for both *Cuckoo Sandbox* (the cuckoo-modified version) and *Joe Sandbox*.

A framework

FAME should be seen as a framework. Instead of developing various malware analysis scripts, FAME modules can be created that are able to interact with each other. As shown in Figure 1, creating a FAME module is as simple as creating a Python class.

```
from fame.core.module import ProcessingModule


class DummyModule(ProcessingModule):
    name = "dummy"
    description = "Does nothing. Give me something useful to do!"

    def each(self, target):
        # Do something useful!

        return True
```

Figure 1: Creating a FAME module.

FAME also comes with some useful modules out-of-the-box (see [1] for the list of currently available modules). One of these is the office_macros module, which leverages oletools [2] to analyse *Office* macros.

Figure 2: The office_macros module analyses Office macros.

THREAT INTELLIGENCE INTEGRATION

Threat intelligence modules are used automatically by FAME to enrich the analysis with tags and indicators from threat intelligence platforms.

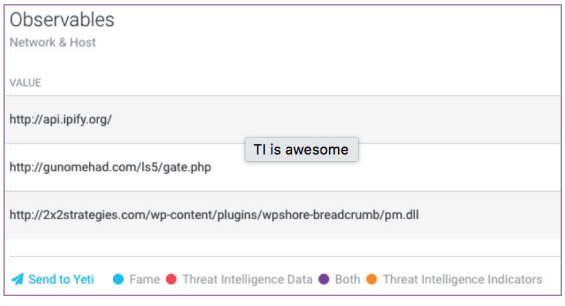

Figure 3: Observables can be added to a threat intelligence platform directly from FAME.

Observables can also be added to a threat intelligence platform directly from FAME. Currently, FAME comes with support for the YETI [3] threat intelligence platform (see Figure 4).

YETI was born out of the frustration of constantly having to answer the question 'where have I seen this artifact before?' or having to Google shady domains in order to tie them to a malware family.

In a nutshell, YETI allows an analyst to:

- Submit observables and get a pretty good guess as to the nature of the threat.
- Inversely, focus on a threat and quickly list all TTPs, observables, and associated malware.
- Skip the 'Google the artifact' stage of incident response.
- Focus on adding intelligence rather than worrying about machine-readable export formats.

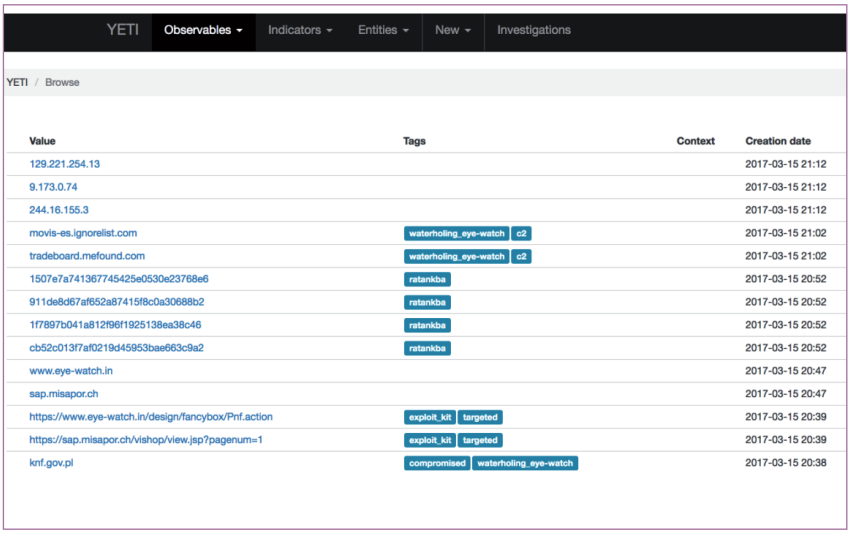

Figure 4: FAME comes with support for YETI.

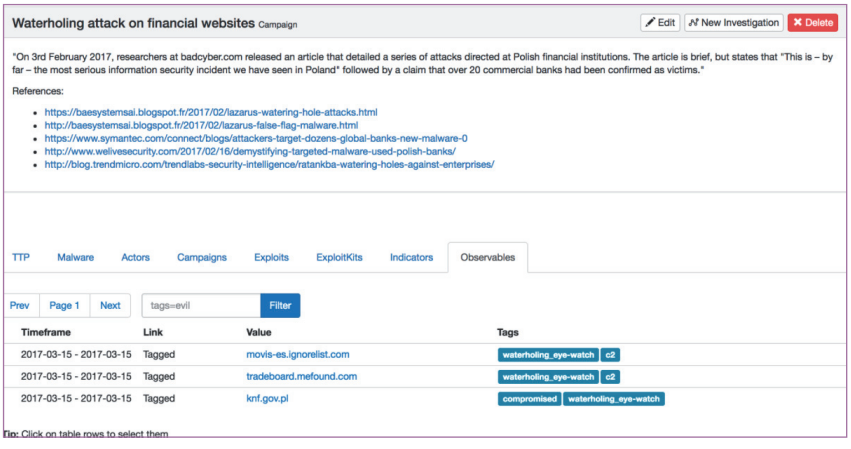

Figure 5: YETI offers the possibility to search instantly through millions of observables.

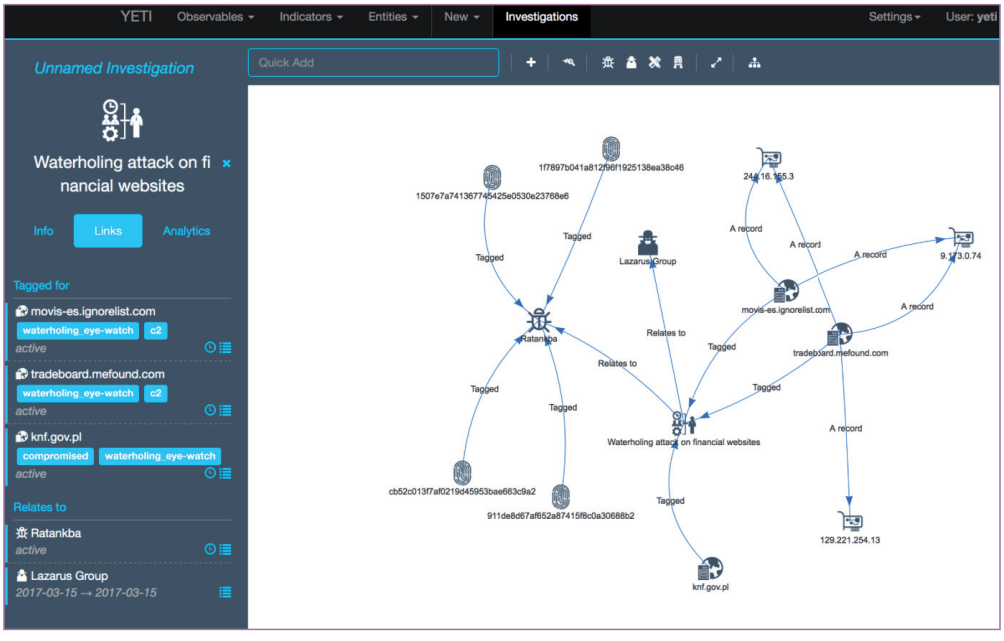

Figure 6: Relationship maps.

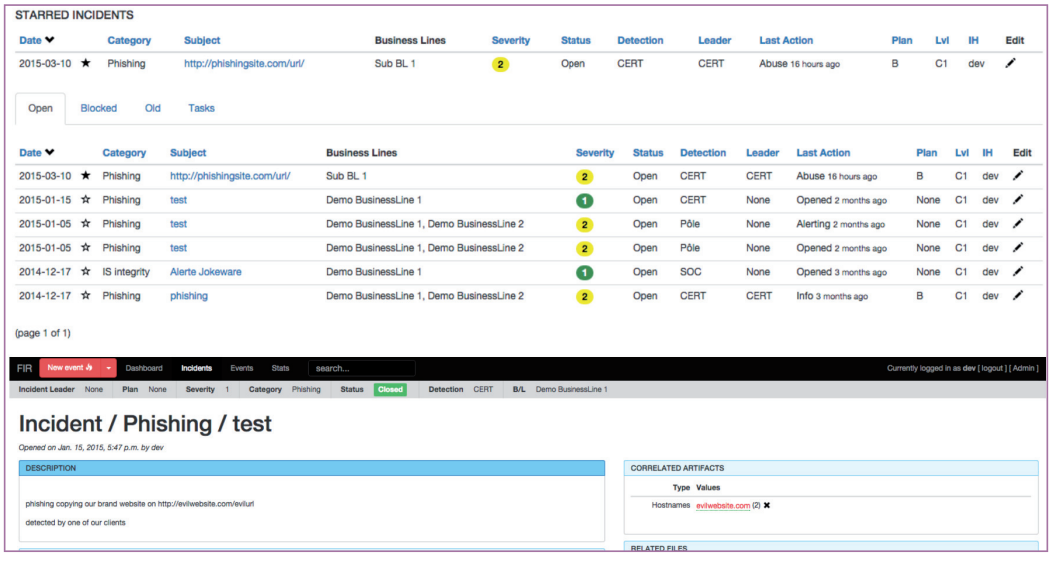

Figure 7: Open-sourcing with the FIR (Fast Incident Response) incident management platform [4].

- Visualize relationship graphs between different threats.

From an operational standpoint, YETI offers the possibility to search instantly through millions of observables while easily tracking campaigns linking to related observables, malware toolsets, threat actors and their TTPs (Figure 5).

There is even the possibility to enrich investigations by generating visually pleasing maps illustrating the relationships, such as the one shown in Figure 6.

OPEN SOURCE

We were particularly pleased with our open-sourcing experience with the FIR (Fast Incident Response) incident management platform [4] (see Figure 7), and as a result of this we decided to release FAME – hoping it will help other incident response teams with their malware analysis needs.

We are looking forward to hearing ideas from the open-source community, and to using some of the awesome

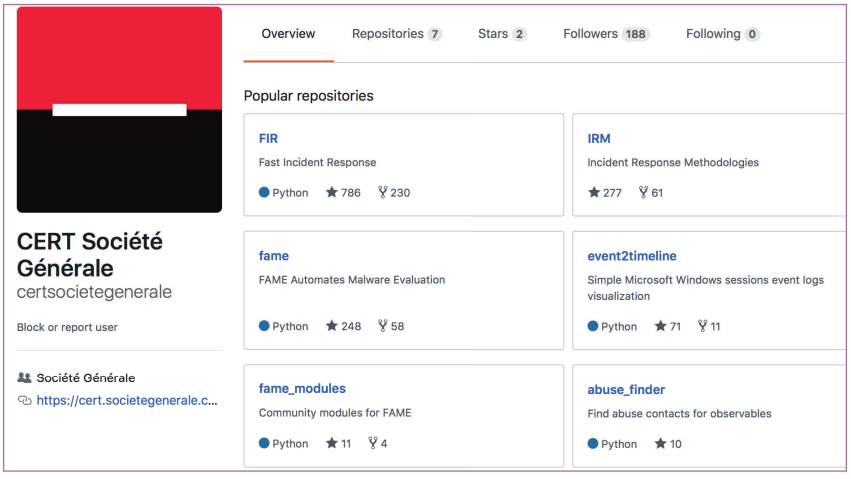

Figure 8: CERT Société Générale GitHub.

modules that we are sure will be created. More details can be found on the FAME Community page [5].

If you are involved in incident response or malware analysis, or if it simply sounds tempting, you can get a custom-tailored FAME framework today from the *CERT Société Générale GitHub* [6], with all the accompanying documents at [7].

REFERENCES

[1] fame_modules repository. https://github.com/certsocietegenerale/fame_modules.

[2] oletools. https://github.com/decalage2/oletools.

[3] YETI. https://yeti-platform.github.io.

[4] FIR – Fast Incident Response incident management platform. https://github.com/certsocietegenerale/FIR.

[5] FAME community page. https://certsocietegenerale.github.io/fame/community.

[6] CERT Société Générale GitHub. https://github.com/certsocietegenerale/fame.

[7] https://fame.readthedocs.io.

Editor: Martijn Grooten

Head of Testing: Peter Karsai

Security Test Engineers: Scott James, Tony Oliveira, Adrian Luca, Ionuț Răileanu, Chris Stock

Sales Executive: Allison Sketchley

Editorial Assistant: Helen Martin

Developer: Lian Sebe

Tel: +44 (0)1235 555139 Email: editor@virusbulletin.com

Web: https://www.virusbulletin.com/