2024 DUBLIN

2 - 4 October, 2024 / Dublin, Ireland

SUPERCHARGE YOUR MALWARE ANALYSIS WORKFLOW

Ryan Samaroo & Jean-Pierre Vigneault

Canadian Centre for Cyber Security, Canada

contact@cyber.gc.ca

ABSTRACT

Assemblyline is an MIT-licensed open-source scalable file triage and malware analysis system developed by the Canadian Centre for Cyber Security (CCCS).

This presentation will highlight how Assemblyline is used by the CCCS to defend the Government of Canada's computer networks and electronic information. Features of interest to be discussed will include load-balancing, caching, configurability, and the integration of popular open-source tools such as CAPE Sandbox, Oletools, Yara, EmlParser, and more. New features such as malware archiving, Yara retro-hunting, querying external sources, and AI summarization will also be touched on.

The main content of the presentation will be a live demonstration of Assemblyline's capabilities involving a malicious file seen in a recent global campaign. Aspects of the system will be highlighted, such as suspicious heuristics and scoring, tagging and pivoting, recursive file analysis, and potential password extraction. The presentation will wrap up with an overview of the REST API and Python client for post-analysis workflows such as data mining and automation.

INTRODUCTION

Assemblyline is an MIT-licensed open-source scalable file triage and malware analysis system developed by the CCCS.

A historical overview of the Assemblyline project can be found at [1]. Assemblyline was designed to do the following:

- Scan millions of files per day
- Automate analysis for incident detection and response teams
- Easily integrate with existing ecosystems by exposing a comprehensive API
- Wrap a user's set of tools into the platform via services.

Assemblyline was released to open source in October 2017; the current available version, version 4, was first released in 2020.

OPEN SOURCE

The open-source nature of Assemblyline is a large part of the project.

CCCS believes in working together with the community to create tools and products that can help everyone be more secure online.

Not only is Assemblyline an open-source project, but the development team also contributes back to the various open-source projects that are used in Assemblyline.

A few of the projects that the Assemblyline team have contributed to in the past, and are still contributing to today, are Kevin O'Reilly's CAPE Sandbox, *VirusTotal*'s Yara-Python, Lief- Project's LIEF, Squiblydoo's Debloat, and the Luxembourg CERT's EmlParser.

ASSEMBLYLINE SERVICES

Alongside the core infrastructure released to open source, the Assemblyline project also has more than 50 open-source services that can be plugged into Assemblyline to dissect the various files that it could receive.

Some of these services are specifically tailored to deal with certain types of files such as *Windows* executables, PDF and *Office* documents, archives, *Android* APKs, etc. Other services are more generic and can handle all file types irrespectively.

There are also services that can connect to third-party servers and infrastructure such as:

- Anti-virus products
- Hash lookup tools (VirusTotal, MalwareBazaar)
- Dynamic analysis sandboxes (CAPE or Cuckoo, for example)
- and much more.

THE CYBER CENTRE'S CURRENT LARGEST DEPLOYMENT

The largest Assemblyline instance that the CCCS manages is deployed as a Kubernetes service in the cloud. This instance leverages node auto-scaling as well as horizontal pod auto-scaling to scale the cluster dynamically depending on the demand of the system. This combination of node and pod-level auto-scaling provides a cost-effective means to maintain operations in real time while utilizing hardware by cloud providers.

This deployment receives data from the CCCS' sensors, which are deployed across the Government of Canada's systems. Approximately 14 million files are received and scanned by Assemblyline every day, and up to 7 million are unique.

SCANNING PROCESS

Figure 1 shows the 10,000-foot view of how Assemblyline processes files:

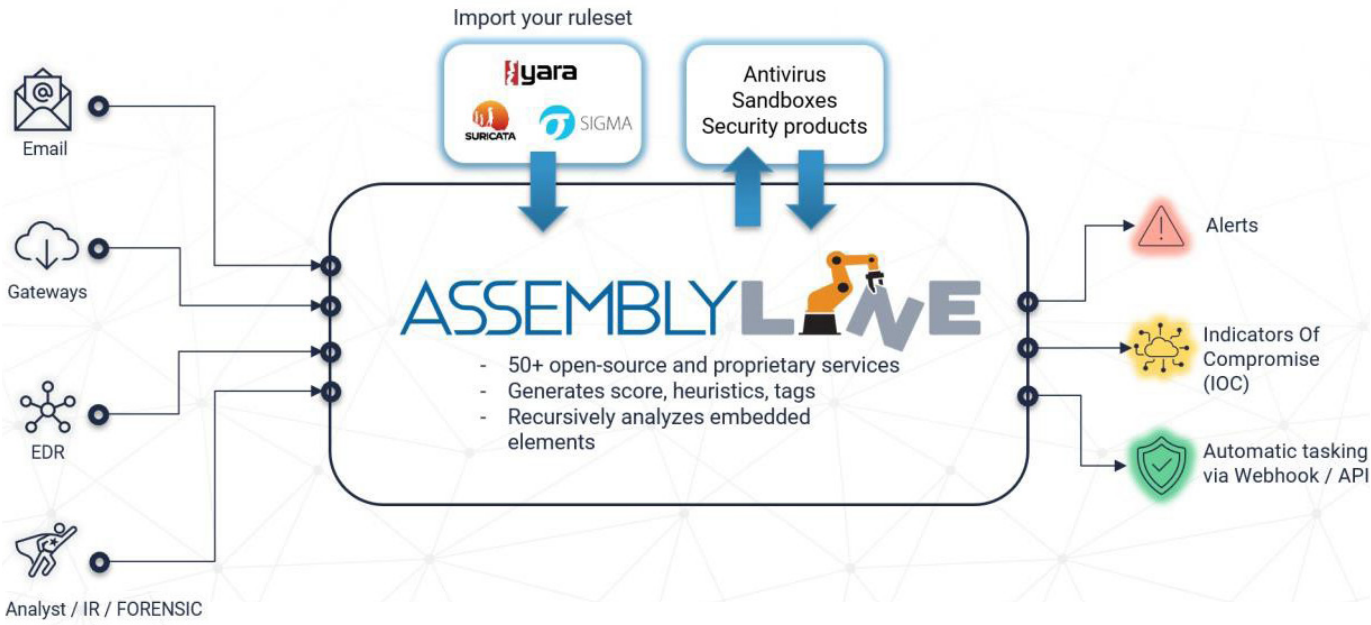

Figure 1: Overview of the scanning process.

It starts by gathering files for Assemblyline to scan. These files come from various sources such as automated threat feeds, partners submitting manually or using scripts to automate ingestion of files through Assemblyline's REST APIs.

Once a file gets in, Assemblyline will look at the file type and send it to the appropriate services out of the 50+ open-source services loaded into the system.

For services that have external rulesets – Yara, Suricata and Sigma to name a few – Assemblyline will periodically pull the rules from the open-source, commercial, and proprietary feeds that are configured and run those different rules on the files that have been submitted.

Assemblyline will also send the file to the different anti-virus products that are plugged in, as well as to a dynamic analysis farm and any other security product that has been configured.

Results for each service will be aggregated, tags will be generated, and a final score will be computed for each file. Assemblyline will recursively analyse the files that it found during analysis.

After this is all done, alerts can be generated in the system if the submitter has chosen to do so, indicators of compromise (IOCs) will be brought forward to the user, and Assemblyline can then pass on the results to another system for the IOCs to be analysed and actioned.

SAMPLE ANALYSIS EXPERIENCE

Discussing the sample analysis experience in Assemblyline is too verbose for this paper, but will be presented during the talk. Please read [2] to gain an understanding of what analysing a malware sample in Assemblyline is like.

MALWARE ARCHIVING

Assemblyline was initially designed to scan transient data. This was due to the large throughput of data that it receives and generates, and the guiding principle that alerts should be triaged as soon as possible, therefore there is no need to store files for longer than a few months.

The CCCS is increasingly moving towards big data analytics, which has created the requirement to store some data forever. With this requirement, the Assemblyline team has developed the 'Malware Archive', which is basically a feature that allows submissions to be stored forever in the system.

The most interesting aspect of this capability is that when you are viewing a submission that has been retained forever in the Malware Archive, you can view 'relations' between files in the system (archived and non-archived) and 'community' feedback about the file.

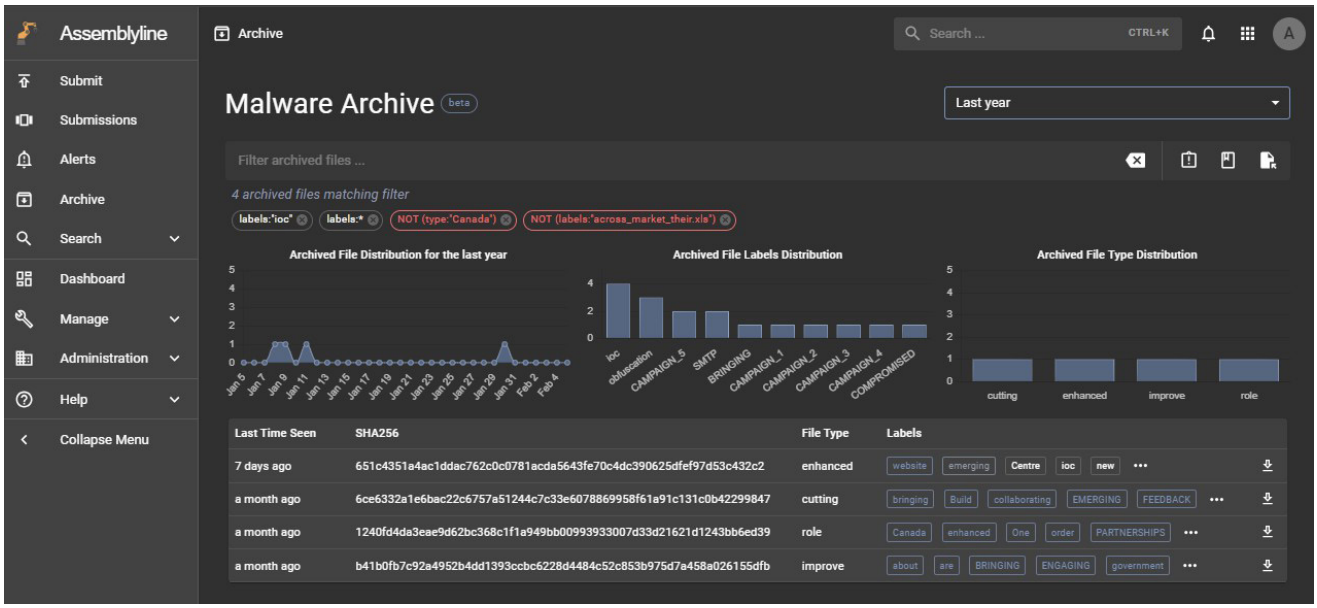

Figure 2: Malware Archive view.

Relations could be attributions, labels, or file types for files that match a given query in the archive. Community feedback can be viewed per submission, where users can add labels and comments, which subsequently can be searchable in the archive view.

YARA RETRO-HUNTING

While Assemblyline facilitates Yara scanning at submission time via the Yara service, the system did not provide a means to retrospectively scan files in its database against new rules that had been added or to test new rules against a corpus of files. This is where the idea of performing 'retro-hunting' using Yara rules was born, similar to the service provided by *VirusTotal*. This feature allows analysts and researchers to add rules where the rule will be applied to files both currently in the system and those that will be submitted later, and it will track hits over the lifetime of the Yara rule.

For more information on Retrohunt, see [3].

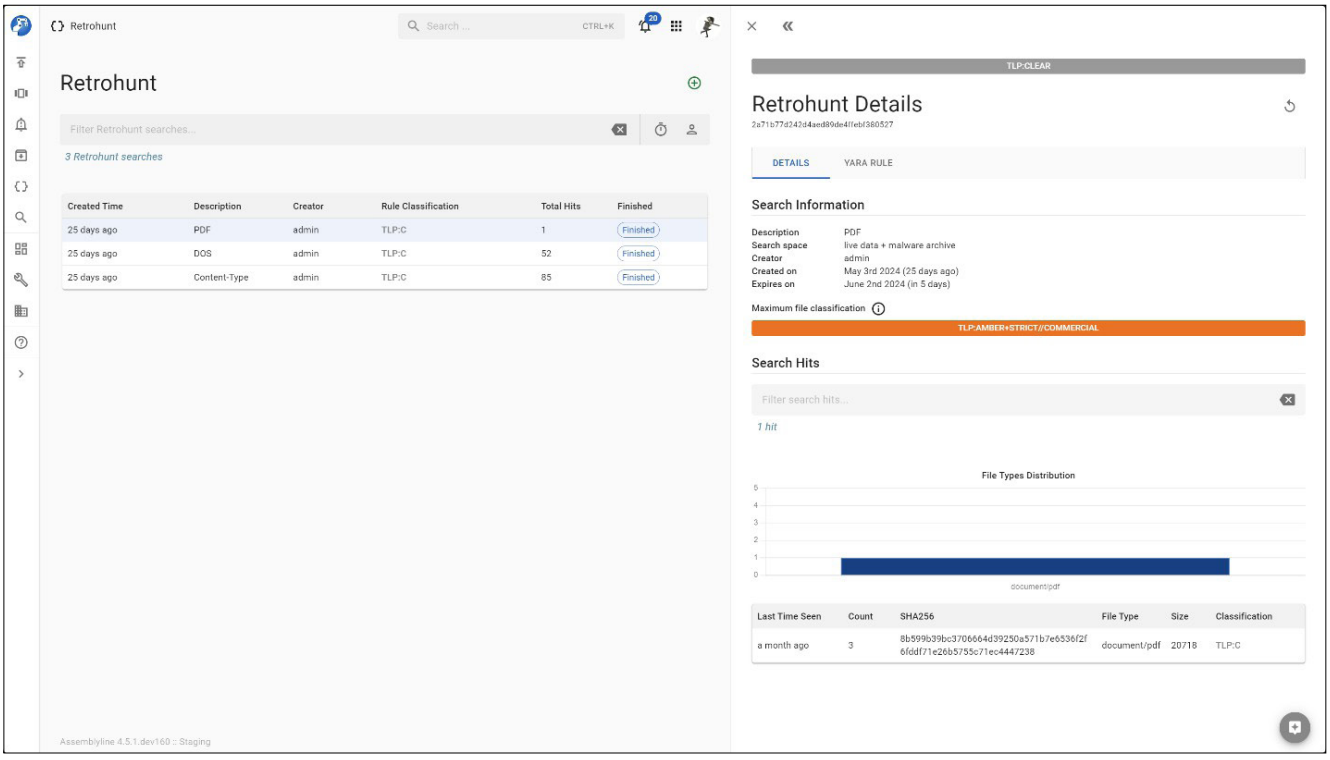

Figure 3: Yara Retro-hunting view.

QUERYING EXTERNAL SOURCES

A capability that was recently added to Assemblyline is the ability to query different systems for data found in Assemblyline. This allows for querying external systems for certain data types and enriching the data in Assemblyline with the data from external systems.

Examples of this is the connector with MalwareBazaar, a free public lookup service for malware, or a connector for a threat intelligence system like a MISP instance.

CONCLUSION

Assemblyline is a free, open-source, MIT-licensed malware analysis and triage tool developed by the Canadian Centre for Cyber Security that was designed, built, and is maintained to address the scanning of millions of files seen on the Government of Canada's systems every day.

The cybersecurity community has embraced this tool to fit a variety of use cases, such as in large enterprise SOCs, Master's and Ph.D.-level file scanning for big data generation, reverse engineering teams testing out new configuration extraction and attributions, and individual users developing hobby tools for their own private labs.

The Cyber Centre believes that cybersecurity is a team sport, and in the spirit of teamwork, we strongly encourage collaboration between varying levels of the community so that we can all contribute and benefit from a safer cyberspace for everyone.

REFERENCES

[1] Garon, S. A Little Bit Of History. Assemblyline Blog Entry #2. https://medium.com/@steve.garon/a-little-bit-of-history-b9383f90602.

[2] Hardy-Cooper, K. Static Analysis Showcase. Assemblyline Blog Entry #3. https://medium.com/@kevin.hardy-cooper/static-analysis-showcase-13f9224cbbc.

[3] Using Retrohunt. https://cybercentrecanada.github.io/assemblyline4_docs/user_manual/retrohunt.