RPE-005: The Chameleons and the Snakes

Contents

[Overview of the RPE 1](#_Toc6476575)

[Attacker Role and Evaluation 1](#_Toc6476576)

[Defender Role and Evaluation 1](#_Toc6476577)

[KeyW Team Approach 2](#_Toc6476578)

[Data Sample Preparation 2](#_Toc6476579)

[Processing Stack 3](#_Toc6476580)

# Overview of the RPE

* Date: May 21 - 23, 2019
* Location: DreamPort Facility in Columbia MD
* Registration Ends: April 30, 2019
* Details: <https://dreamport.tech/event-rpe-005-the-chameleons-and-the-snakes.php>

This RPE specifically targets malware signature diversity and signature measurement for Microsoft Windows (x86 and amd64), Linux (x86 and amd64) and Android (x86 and ARM) in a simulated operational environment at a realistic pace. Participants will be challenged to assume an ‘attacker’ or a ‘defender’ role. DreamPort will publish explicitly limited PCAP files of the DreamPort network for this RPE in advance of the actual evaluation. Binary samples will be provided to support development of machine learning models.

## Attacker Role and Evaluation

The attacker team will be challenged to create (through integration, enhancement or from scratch) a single tool for altering the signature of an operational tool for the previously named platforms without altering existing functionality. For the offense, teams should strive to produce a single technology that can alter the signature of multiple executable formats (e.g. ELF, PE).

During scoring, offense teams must provide a single sample chosen from the list of available candidates (published in advance on dreamport.tech) that they have altered in an attempt to evade detection by Defense teams. Each offense sample will be combined with random artifacts (both good and bad) into a single 'security alert' and given to each defense team for analysis. Offensive teams will be evaluated according to the following criteria:

* Does your sample get flagged as malware? Benign?
* Does your sample add an excessive amount of data to the original sample (defined as more than 1.5 times the size of the original sample)?
* Do you use a known technique for alteration? Did you invent your own?
* Tool requires separate process/approach to invoke
* Tool requires elevated privileges to invoke
* Tool runs without any additional steps (e.g. unpacking)
* Tool produces Unique Hash each run against an input
* Team catches modified sample product
* Team attributes modified product to specific offense team

## Defender Role and Evaluation

Defenders will create (through integration, enhancement or from scratch) a single tool for the fully automated classification of an unknown executable as malware/benign, variant of known sample, attributed to known group (based on previously established knowledge). DreamPort will give strong consideration to any defensive team who can prove their use of machine learning (in a one on one interview) for the purposes of identification, classification or attribution of submitted samples. DreamPort will release a subset of malware binary samples for the purposes of machine learning. Defensive teams should strive to produce an extensible automated solution that can perform automated classification, at least triage analysis and potentially attribution for suspicious files or artifacts. Defense teams should focus on solutions in which can extend their behavior with minimal design changes (e.g. plugins, scripts).

During scoring, defenders must evaluate each security alert (offense sample will be combined with random artifacts, both good and bad) provided by the offense and determine:

* Is the sample benign or malware?
* Is the malware known?
* If so, can it be attributed to a known campaign?
* If so, can it be attributed to a specific author?
* Defenders should submit an analysis report for each sample detailing answers to these questions

# KeyW Team Approach

It is proposed that KeyW will develop a prototype defense solution to develop a machine learning approach to the classification of malware and clean files in a sample set of binaries.

Significant prior work has been done to use ML techniques to detect binaries. We propose a ML based detection capability that 1) uses the newly released NSA developed Ghidra tool to produce disassembled binary files as the starting point for evaluation, and 2) applies Natural Language Processing (NLP) techniques to the product of the disassembly process. This presents an unsupervised learning approach that uses clustering to split a large, unlabeled dataset of malware samples into groups of similar objects. This clustering can help optimize downstream efforts for the manual labeling of additional malware samples, such as diversified samples of the original malware samples. Furthermore, classification can potentially inform researchers investigating attribution to known malware families, campaigns, and authors.

## Data Sample Preparation

**Data corpus** – We will curate a corpus of malware samples, along with other non-malicious binaries. Samples need to include Linux, Windows, and Android (maybe iOS). There are several available collections online from which we can draw. Dreamport will also provide some initial samples.

**Diversification tools** - We identify one or more tools to diversify malware to ensure that our data corpus contains multiple diversified versions of malware samples. When source code is available we can use diversity compilers such as multicompiler or other LLVM based compilers. If source code is not available we can potentially identify a static binary rewriting/diversifying tool.

Binary Samples

Non-malicious

Malware

Standard Compile

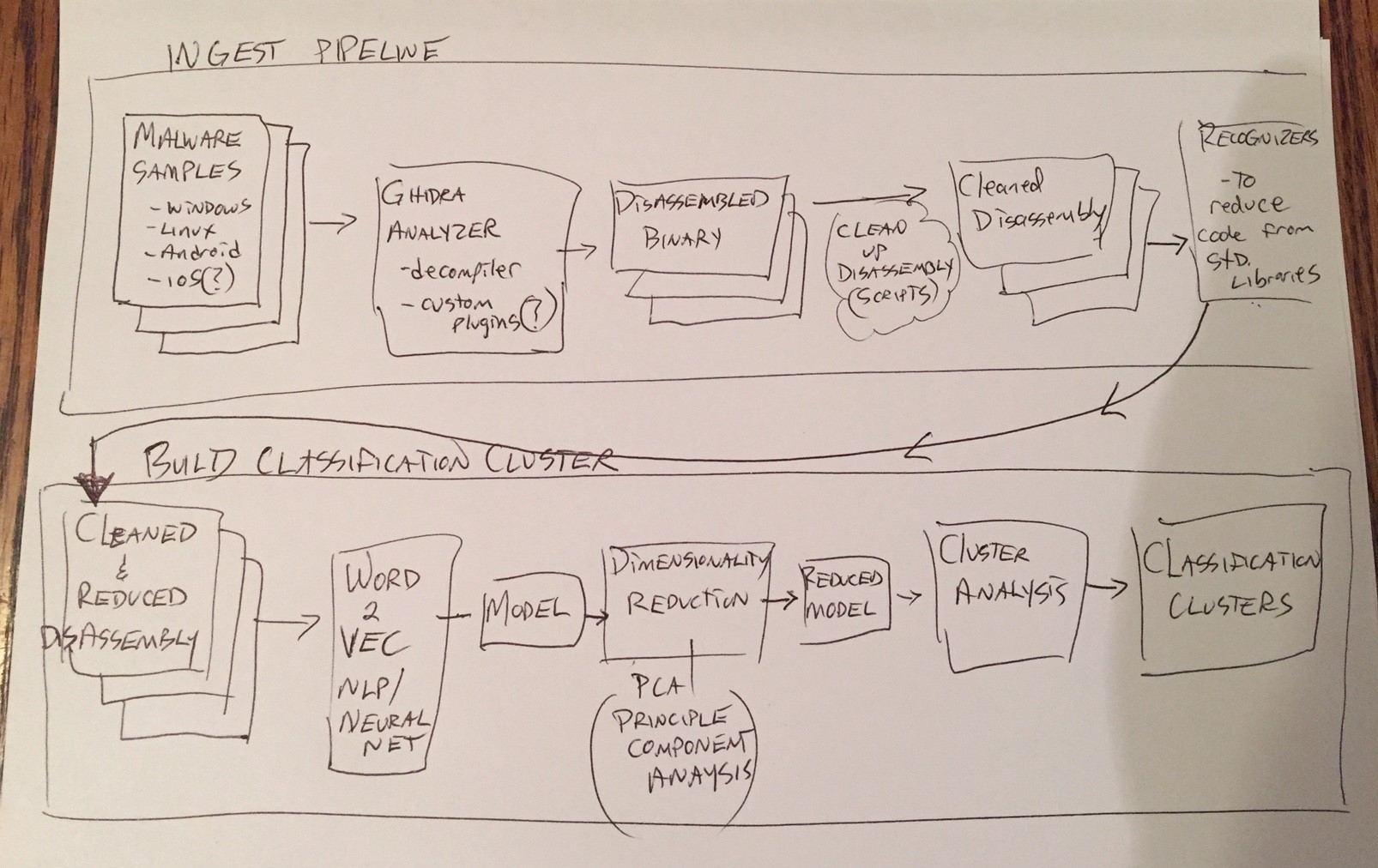
Diversity Compile

Standard Compile

Diversity Compile

## Processing Stack

**Ingest Pipeline** – We will develop an ingest pipeline that takes malware samples, disassembles them with Ghidra, provides hooks for cleanup scripts (to correct and prepare disassembly as necessary), and provides hooks for code recognizers (recognizes code from standard libraries to reduce it from the to-be-analyzed code). Pipeline inputs are malware samples. Pipeline outputs are cleaned and reduced disassembly from malware samples.

**Clustering Pipleline** – We will develop a processing pipeline that uses Word2Vec to process cleaned and reduced disassembly and develop a multi-dimensional shallow neural network of features discovered in the assembly. Pipleline will employ a dimensionality reduction phase (using a tool such as PCA – principle component analysis) to produce a reduced model. Cluster analysis will be used to define classification groups. Pipeline inputs are malware samples. Pipeline outputs are cleaned and reduced disassembly from malware samples.

**References**

Pai, Swathi, "A Comparison of Clustering Techniques for Malware Analysis" (2015). Master's Projects. 404. <https://scholarworks.sjsu.edu/etd_projects/404>

Per Larsen, Andrei Homescu, Stefan Brunthaler, Michael Franz. SoK: Automated Software Diversity. In IEEE Symposium on Security and Privacy, 2014. <http://www.cse.usf.edu/~xou/sec/software_diversity.pdf>