Black Hat Europe 2019 Arsenal

AVCLASS++

Yet Another Massive Malware Labeling Tool

NTT Secure Platform Laboratories

Yuma Kurogome

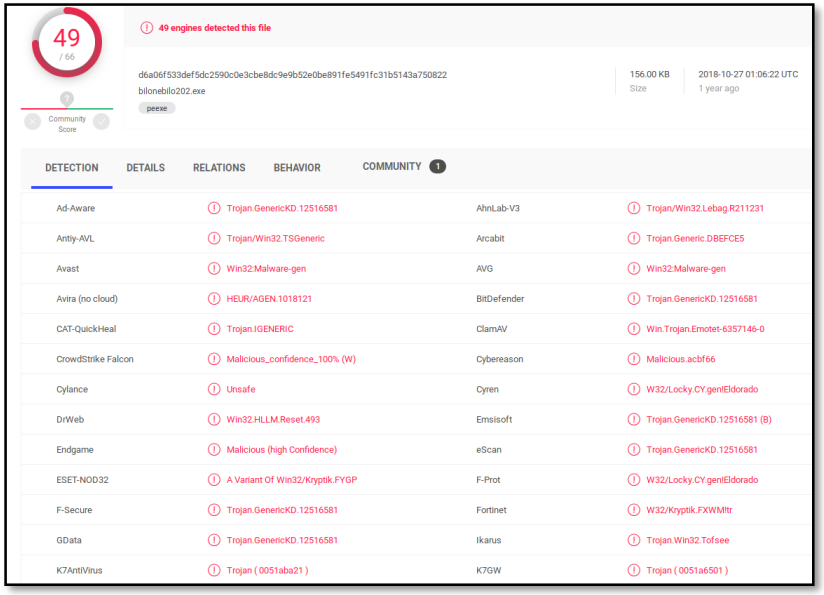
Machine Learning

- Gold rush
  - Malware detection, classification, attribution, …
- Supervised learning
  - **Labeled** training data is needed
- Unsupervised learning
  - **Labeled** data is needed, for evaluation, after all

Label	Sepal length	Sepal width	Petal length	Petal width
Setosa	5.1	3.5	1.4	0.2
Setosa	4.9	3.0	1.4	0.2
Verscicolor	6.4	3.5	4.5	1.2
…				

Labeling Problem

- Manual labeling may not scale
  - WIRESHARK  cuckoo ...
- VirusTotal (VT) comes to rescue
  - Input:
    - Malware sample
  - Output
    - Antivirus (AV) scan results
- Problem
  - Sometimes AVs return different names
  - Which AV do you trust?
  - How to aggregate multiple AV scan results to one label?



AVCLASS++

- Malware labeling tool
  - Aggregates AV scan results and output label(s)
  - Input:
    - VT report(s)
    - Malware sample(s) (optional)
  - Output:
    - Label(s)
- Based on AVCLASS
  - One of the most well-used oracles in such a scenario

AVCLASS: A Tool for Massive Malware Labeling

M Sebastián, R Rivera, P Kotzias… - … Symposium on Research …, 2016 - Springer

Labeling a malicious executable as a variant of a known family is important for security applications such as triage, lineage, and for building reference datasets in turn used for evaluating malware clustering and training malware classification approaches. Oftentimes …

☆ ❝❝ Cited by 137 Related articles All 6 versions

Features

- Automatic
  - AVCLASS++ removes manual analysis limitations on the size of the input dataset
- Vendor-agnostic
  - AVCLASS++ operates on the labels of any available set of AV engines, which can vary from sample to sample
- Cross-platform
  - AVCLASS++ can be used for any platforms supported by AV engines, e.g., Windows or Android malware

Features

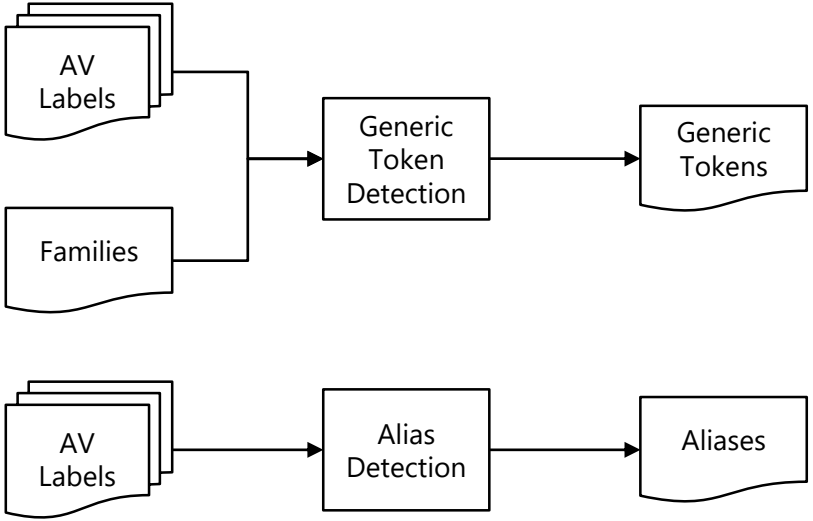
- Does not require executables
  - AV labels can be obtained from online services like VT using a sample's hash, even when the executable is not available
  - Yet, AVCLASS++ has also a potential that can improve label accuracy if there are executables
- Quantified accuracy
  - The original AVCLASS had evaluated on five publicly available malware datasets with ground truth
  - AVCLASS++ is further tuned to perform under adverse conditions
- Open source
  - Check out the QR code!

Workflow

- AVCLASS++ has three phases:
  - Preparation
  - Labeling
  - Label Propagation

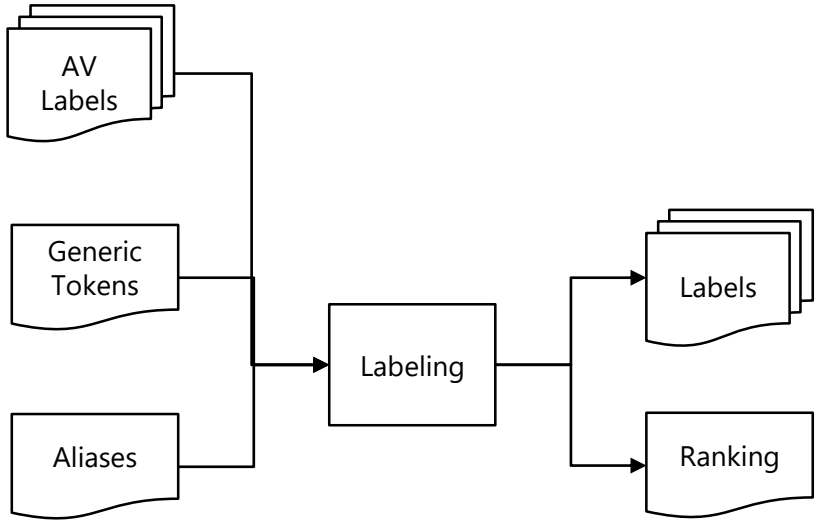
Preparation (Optional)

- AVCLASS++ requires the two files other than VT reports
  - Generic tokens
  - Aliases
- AVCLASS++ shipped with the pre-configured files
  - You don't need to touch them
  - But you can customize them



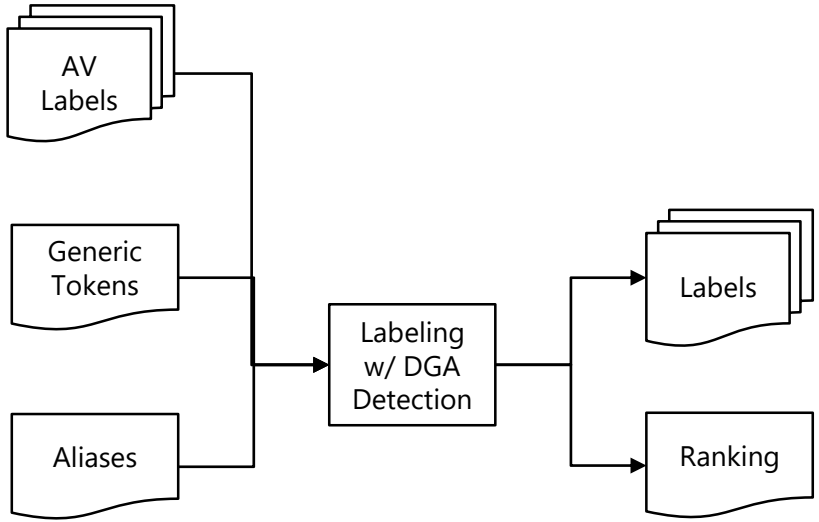
Labeling

- AVCLASS++ generates malware label(s) as follows:
  - For each scan result:
    - Remove suffixes
      - Win32/pony.c → Win32/pony
    - Filter unnecessary tokens
      - Win32/pony → pony
    - Replace aliases
      - pony → fareit
  - After that, AVCLASS++ creates a ranking
    - The most popular label would be output
    - Also, a ranking itself can be output



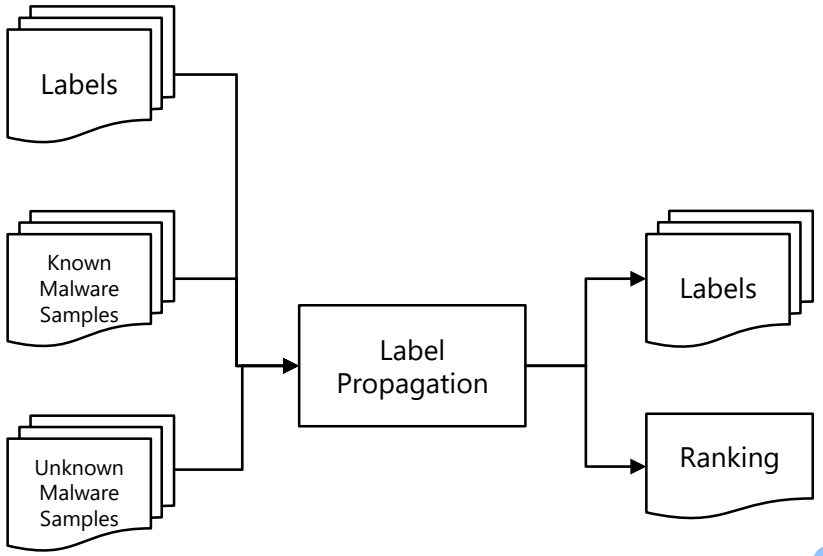
Labeling – DGA Detection

- AVCLASS fails to labeling when:
  - Malware label name is randomly generated
- Domain generation algorithm (DGA)
  - Malware uses DGA to generates new domain names
  - Some AVs use DGA-like methods to generate malware label names
- Why not divert DGA detection methods to remove such a label?
  - Meaningful characters ratio
  - N-gram normality score



Label Propagation (Optional)

- AVCLASS also fails to labeling when:
  - No AV signatures at VT submissions
  - Only generic signatures reacted
- AVCLASS++ finds labels for unlabeled samples by:
  - Extracting PE file features:
    - Headers
    - Sections
    - Imports
    - …
  - Propagating labels from similar samples

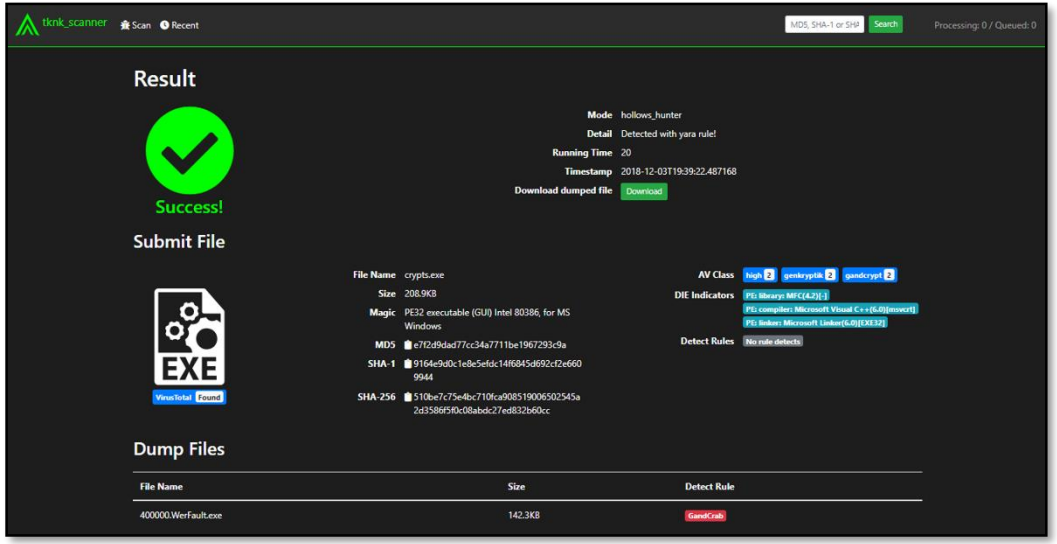


AVCLASS vs AVCLASS++

	AVCLASS	AVCLASS++
Labeling	✓	✓
Family Ranking	✓	✓
PUP Classification	✓	✓
Ground Truth Evaluation	✓	✓
Generic Token Detection	✓	✓
Alias Detection	✓	✓
DGA Detection		✓
Label Propagation		✓
Python 3 Compatible		✓

Showcase – nao-sec/tknk_scanner

- Automated malware analysis system
  - Integrated with VT, YARA, and hollows_hunter
  - Input:
    - Malware sample
    - YARA rules
  - Output:
    - Label
    - Memory dump
    - …



DEMO