DeepReflect: Discovering Malicious Functionality through Binary Reconstruction

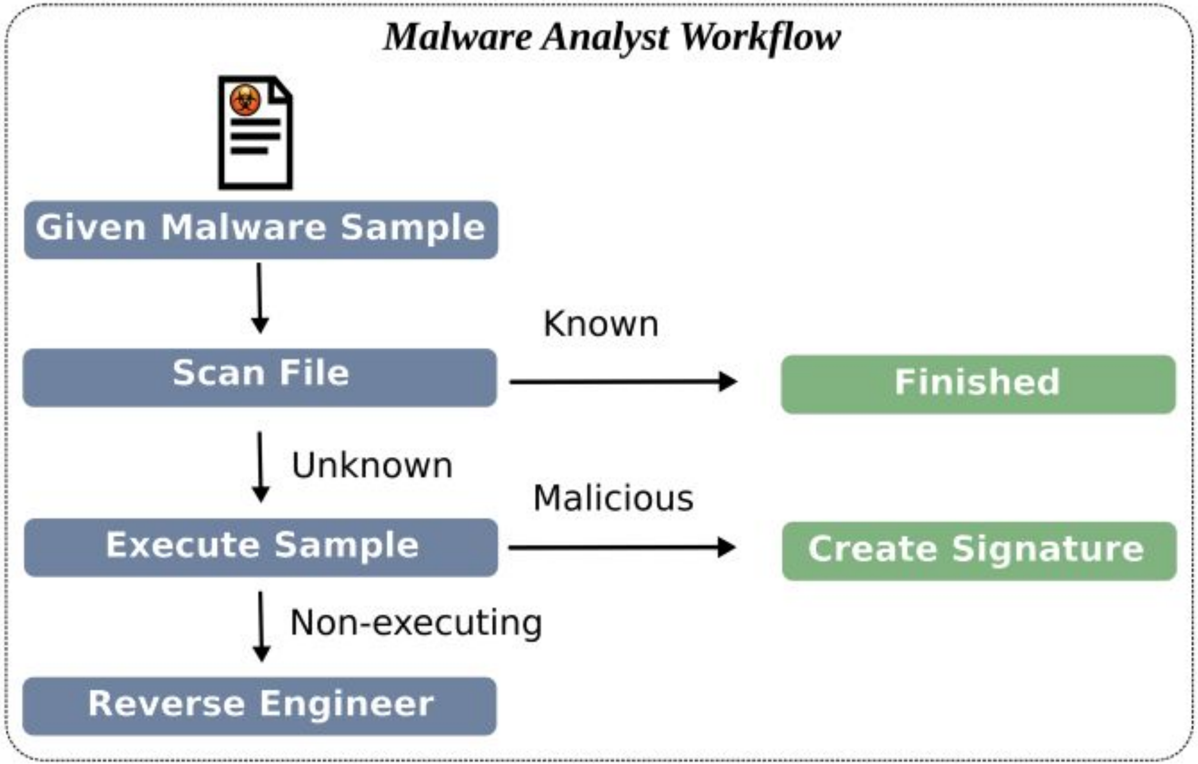
Evan Downing*, Yisroel Mirsky*†◇, Kyuhong Park*◇, Wenke Lee*

* Georgia Institute of Technology

† Ben-Gurion University

◇ Equal contributions

Motivation


Malware Analyst Workflow
Given Malware Sample
Scan File
Known
Finished
Unknown
Execute Sample
Malicious
Create Signature
Non-executing
Reverse Engineer

Overview

- Analysts want to quickly identify and label malicious functions in malware
- Cannot assume or obtain labeled dataset (too expensive timewise / doesn’t exist)
    - Thus we identify these regions via unsupervised learning
- Cannot manually label all regions all of the time (too expensive timewise)
    - The analyst labels a few regions in a semi-supervised approach, which adds a bonus of labeling these identified functions

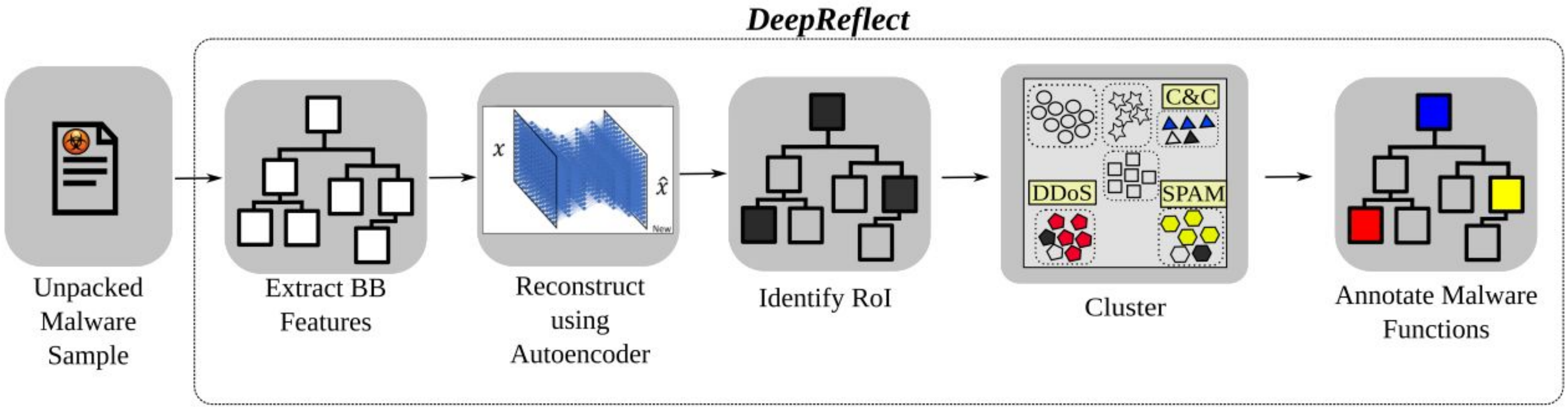
Prior Work

- ML-based solutions: Classification or detection, **not** behavior identification
- FireEye’s CAPA (July 2020)
- Eyeball strings and API calls indicative of behavior

Challenges & Insights

1. Need to distinguish between benign and malicious behaviors
    a. Use an **unsupervised** deep learning model (an autoencoder) to locate malicious **functions** in binaries
2. Understand the semantics of the identified malicious behavior
    a. Use a **semi-supervised clustering** model which classifies the identified functions
    b. Requires few labels obtained from analyst’s daily workflow

Overview of DeepReflect



Features

- Inspired from ACFG features used for bug-finding (CCS 2017)
- 18 Features:
    - **Structural**: Flow of operations (e.g., connect, send, recv, etc.)
    - **Arithmetic** Instruction Types: How mathematical operations are carried out at the higher level (e.g., encryption, obfuscation)
    - **Transfer** Instruction Types: Flow of data (arguments provided to and returned from functions)
    - **API Call** Categories: Used to execute behaviors (filesystem, registry, network, process, etc.)

Dataset

Benign Dataset

Category	Size	Category	Size
Drivers	6,123	Business Software	1,692
Games	1,567	Utilities	1,453
Education	1,244	Developer Tools	1,208
Audio	1,023	Security	1,000
Communications	994	Design	844
Digital Photo	826	Video	787
Customization	778	Productivity	730
Desktop Enhancements	699	Internet	695
Networking	612	Browsers	440
Home	390	Entertainment	257
Itunes	43	Travel	17

Malware Dataset

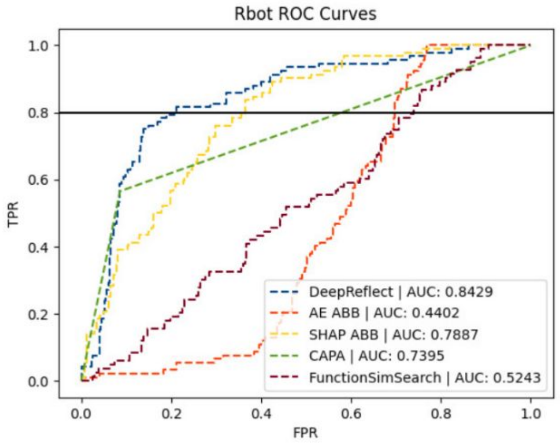
Label	virut	vobfus	hematite	sality	crytex
Size	3,438	3,272	2,349	1,313	914
Label	wapomi	hworld	pykspa	allaple	startsurf
Size	880	720	675	470	446

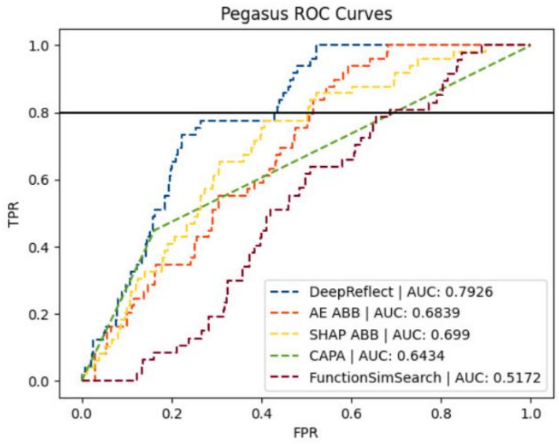
Top 10 most populous families

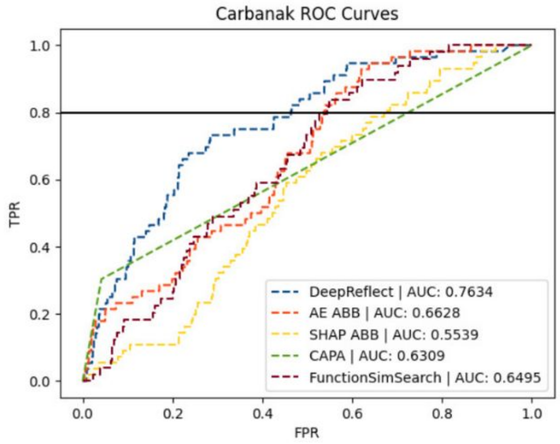
Evaluation 1: Reliability

- Ground-truth samples
  - Rbot (2004), Pegasus (2016), Carbanak (2014)
- Baseline Tools
  - VGG19 model + SHAP (deep learning comparison)
  - CAPA (FireEye)
  - FunctionSimSearch (Google Project Zero)

Evaluation 1: Reliability (cont.)



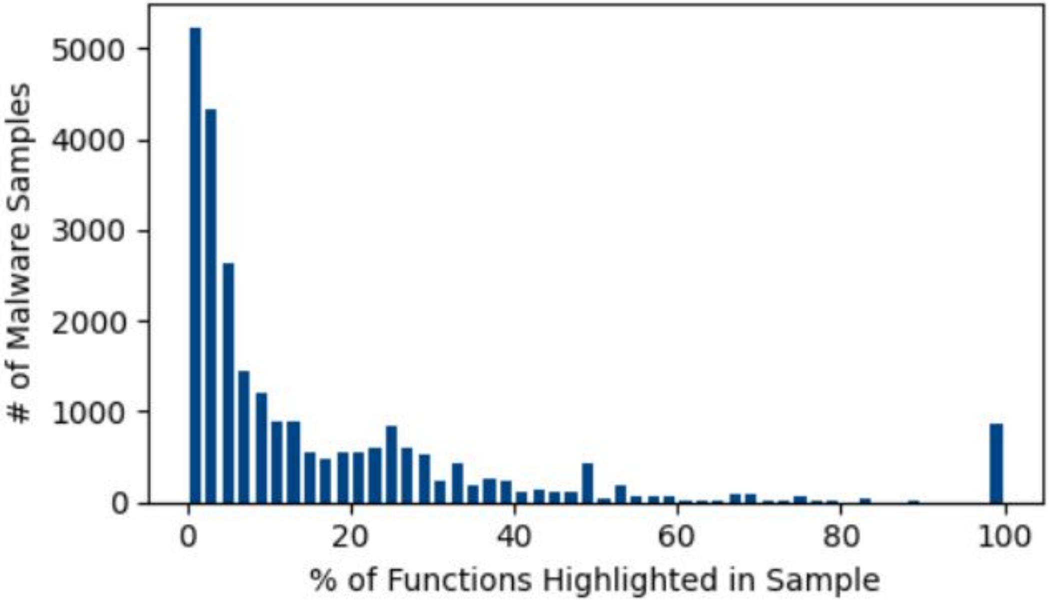




Evaluation 2: Cohesiveness

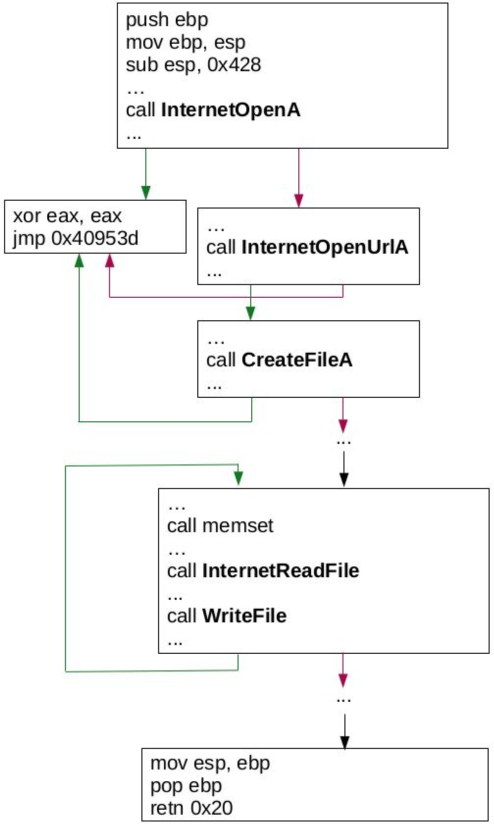
- DeepReflect identified ~600k malicious functions in ~25k malware samples
- HDBSCAN produced ~22k clusters
  - Largest cluster: ~6k functions
  - Noise points: ~60k functions
- Analysts labeled 119 functions via MITRE (60% malicious, 40% benign)
- Clustering matches 89.7% of an analyst’s manually-clustered functions

Evaluation 3: Focus



Evaluation 4: Insights

- Different (unrelated) malware families share the same functions
- 1.7k clusters had at least one singleton sample
- Novel malware families form new clusters



Evaluation 5: Robustness

- Used OLLVM on Rbot and enabled combinations of obfuscations
  - Control-flow flattening
  - Instruction substitution
  - Bogus control-flow
- Mimicry-like attack
- DeepReflect’s results weren’t significantly affected

Discussion

- Obfuscation
- Adversarial ML attacks
- Training Data Quality
- Human Error

Questions & Comments

Email:
edowning3@gatech.edu

Implementation & Dataset:
https://github.com/evandowning/deepreflect