EXPLOITING RANSOMWARE PARANOIA FOR EXECUTION PREVENTION

Ali AlSabeh, Haidar Safa, Elias Bou-Harb, Jorge Crichigno

Presented by: Ali AlSabeh

IEEE International Conference on Communications 2020

UNIVERSITY OF
South Carolina

College of Engineering
and Computing

OVERVIEW

- Introduction
- Motivation
- Related Work
- Problem Statement
- Contribution
- Proposed Approach
- Evaluation and Results
- Conclusion and Future Work

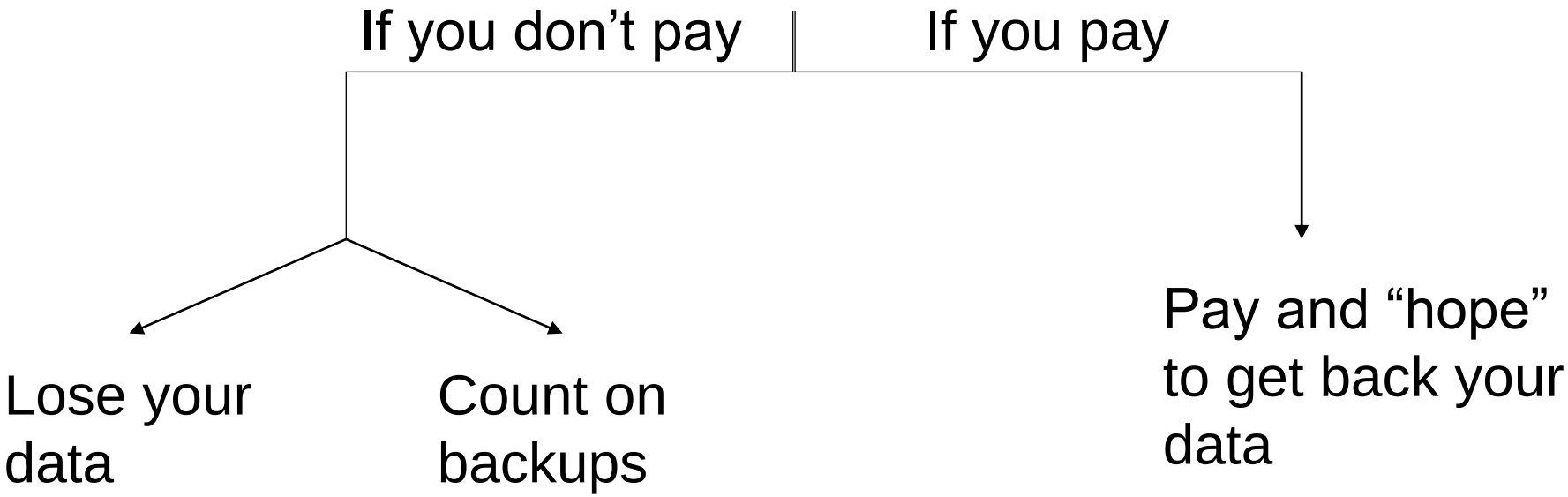
INTRODUCTION

UNIVERSITY OF
South Carolina

College of Engineering
and Computing

RANSOMWARE

- Increasingly devasting attack
- Locks and/or encrypts the victims' machine
- Ask for a ransom to return encrypted data



MOTIVATION


UNIVERSITY OF
South Carolina
College of Engineering
and Computing

MOTIVATION

Crypto-Locked: Illinois Public Health District

Last Tuesday, Champaign-Urbana Public Health District, which serves about 210,000 people in central Illinois, was hit by Netwalker ransomware, aka MailTo. "We are working to get our

HIPAA Journal @HIPAAJournal · May 22

Maze **#ransomware** gang attacks UK COVID - 19 research firms ow.ly /KdtT50yZMcI #healthcare #cybersecurity

Magellan Health, a for-profit managed health care and insurance firm, was the victim of a ransomware attack.

@CityPowerJhb · Jul 25, 2019

#Update City Power has been hit by a **Ransomware** virus. it has encrypted all our databases, applications and network. Currently our ICT department is cleaning and rebuilding all impacted applications.^GR

RELATED WORK

UNIVERSITY OF
South Carolina

College of Engineering
and Computing

ANALYSIS TECHNIQUES

- Static analysis does not execute the sample and it is achieved by inspecting:
  - Source code
  - Assembly
  - Executable file…
- Dynamic analysis executes the sample in an isolated environment and records the generated activities such as:
  - File access
  - Memory access
  - Registry access

RANSOMWARE DETECTION TECHNIQUES

- BRIDEMAID
  - Combination of static and dynamic analysis to detect ransomware in Android operating system
- UNVEIL
  - Detects ransomware by creating an artificial, yet realistic execution environment that can detect file lockers and screen lockers
- NetConverse
  - Detects ransomware from the generated network traffic using machine learning
- RansomFlare
  - Combination of behavioral-based analysis and machine learning to detect ransomware

PROBLEM STATEMENT

PROBLEM STATEMENT

- The previous approaches devote their work to **detect** a ransomware from the behaviors it generates
- However, there is no **prevention** technique that suppresses the execution of ransomware
- In our paper, we work on **preventing** a contemporary ransomware sample from the environmental artifacts it executes prior to the attack.

CONTRIBUTION

UNIVERSITY OF
South Carolina

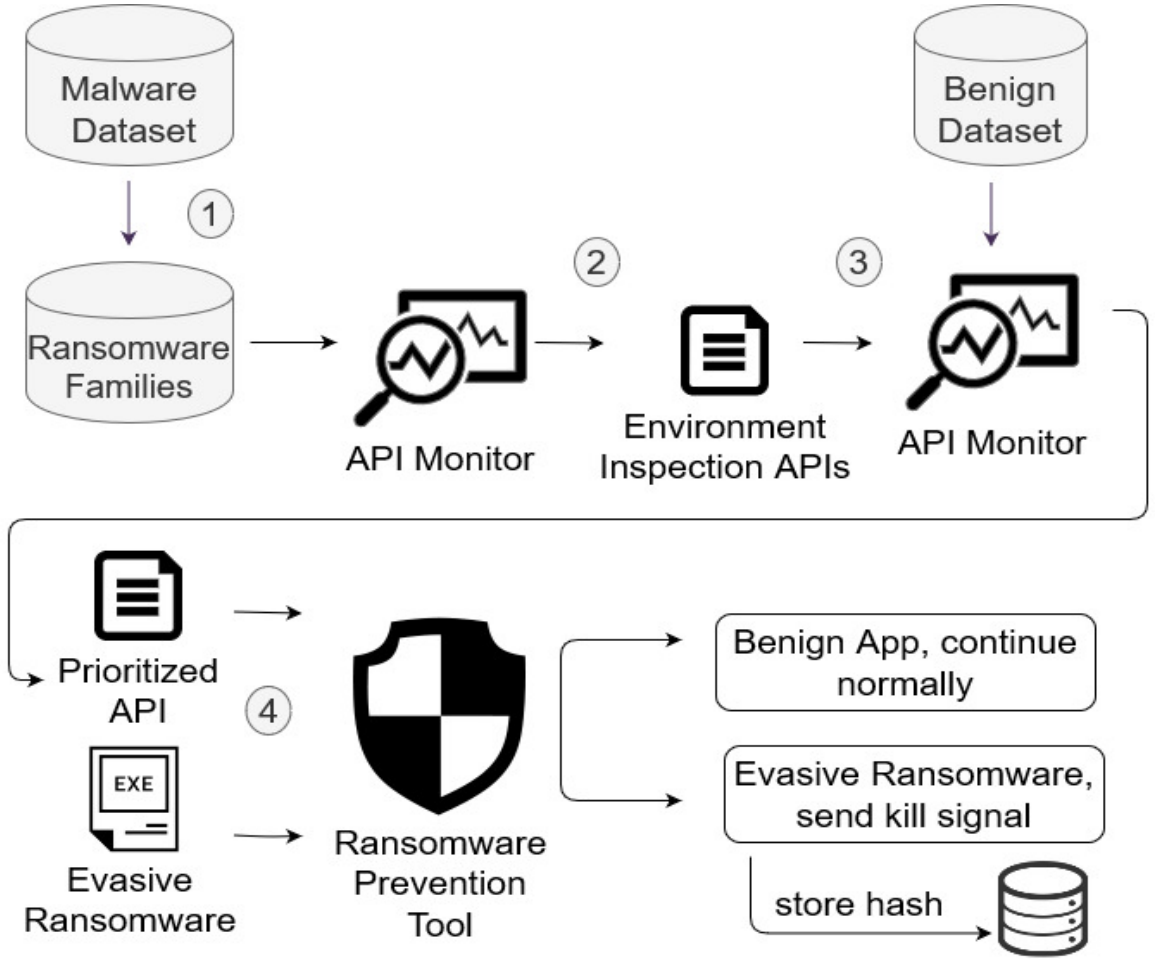
College of Engineering
and Computing

CONTRIBUTION

- Exploring the behavior of contemporary ransomware by collecting relevant artifacts related to fingerprinting the execution environment
- Designing and developing a host-based approach which can detect and prevent contemporary ransomware through monitoring their “paranoia”
- Executing empirical evaluations using real ransomware datasets
  - Training: 91% accuracy
  - Testing: 84% accuracy

PROPOSED APPROACH

PROPOSED APPROACH



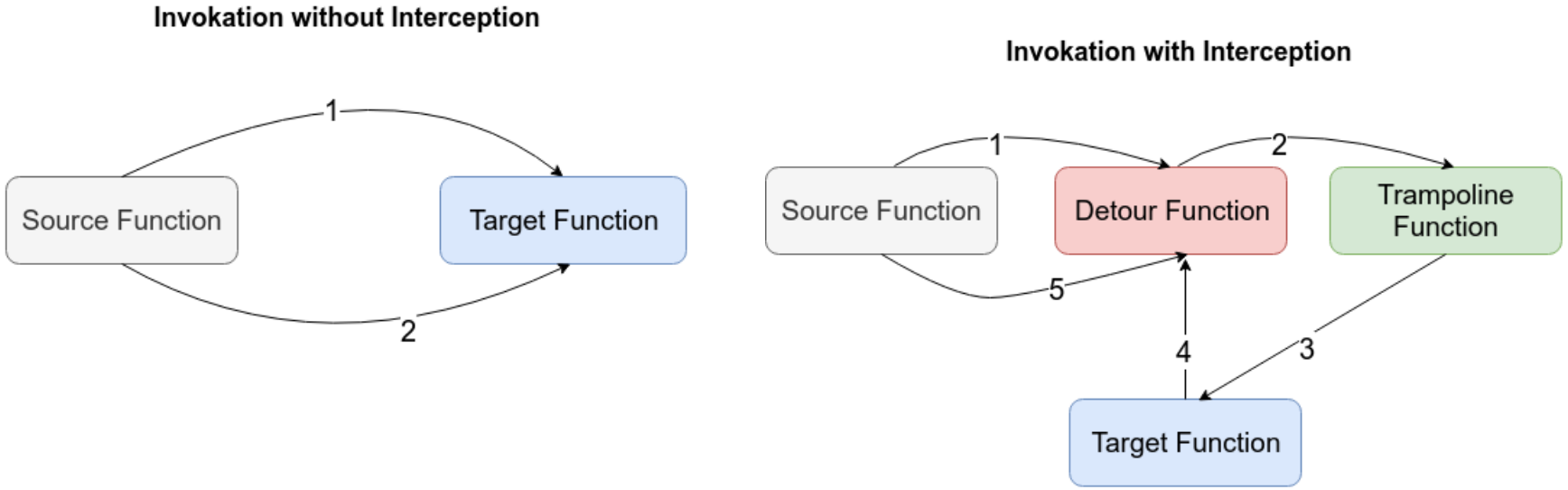
GetWindowsDirectory
EnumPrintersW
EnumServicesStatusW
C:\agent.py
SystemBiosVersion

DATASET COLLECTION

- The collected ransomware samples were from multiple sources
- The collected samples were inconsistent (file types, compatibility) and they lack metadata
- Using VirusTotal API, we did the following:
  - Performed data cleaning to filter out incompatible samples w.r.t our execution environment
  - Associated meta-data labels to map each ransomware sample to its corresponding family

API MONITORING AND COLLECTING ENVIRONMENT ARTIFACTS

- To study the behavior of an application we monitored the called APIs using Microsoft Detour library
- The collected APIs were filtered to include the ones mainly related to environment fingerprinting

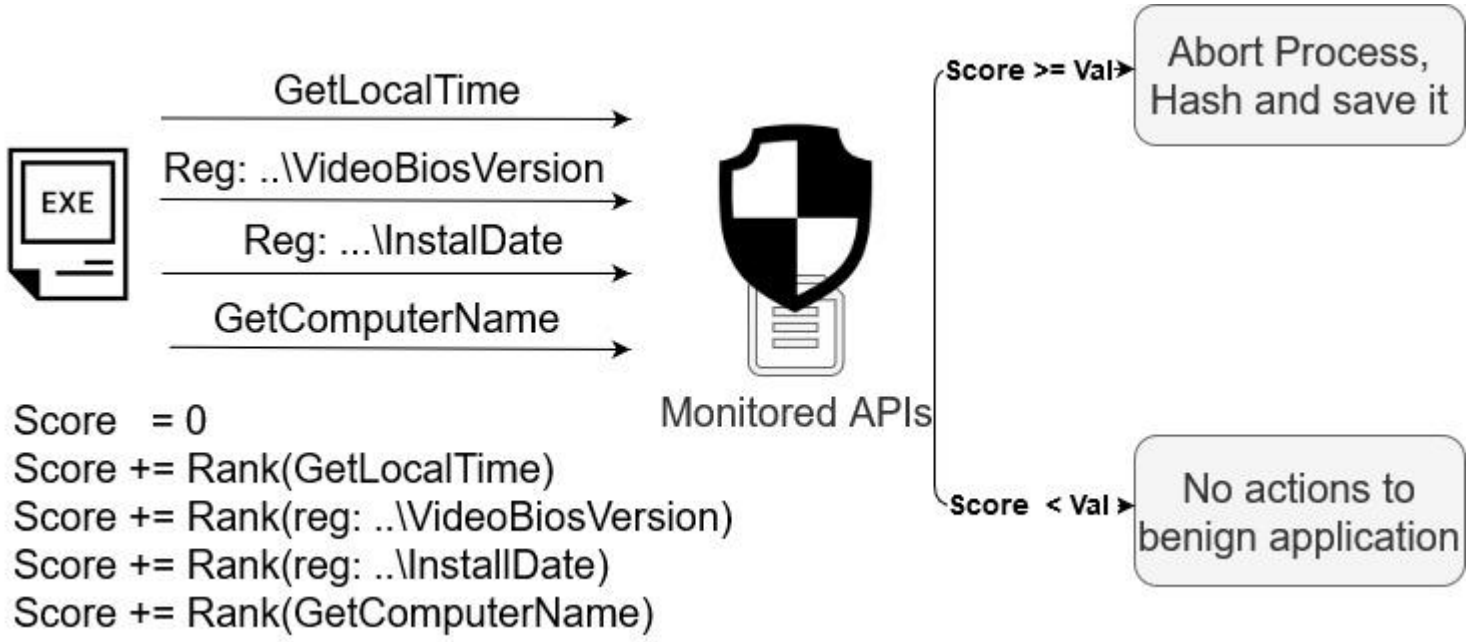


MANAGING FALSE POSITIVES USING PRIORITIZED COLLECTED APIS

- To address false positives, the collected APIs were monitored against benign applications
- A rank is assigned to each API
- The more the API is called by evasive ransomware, the more its rank will be close to 10
- Similarly, the more the API is called by benign applications, the less its rank will be

MANAGING FALSE POSITIVES USING PRIORITIZED COLLECTED APIS

- Every monitored program will have a score that is initially zero and is incremented by the rank of each called API
- Once the score exceeds a threshold, the monitored program is killed



EVALUATION AND RESULTS

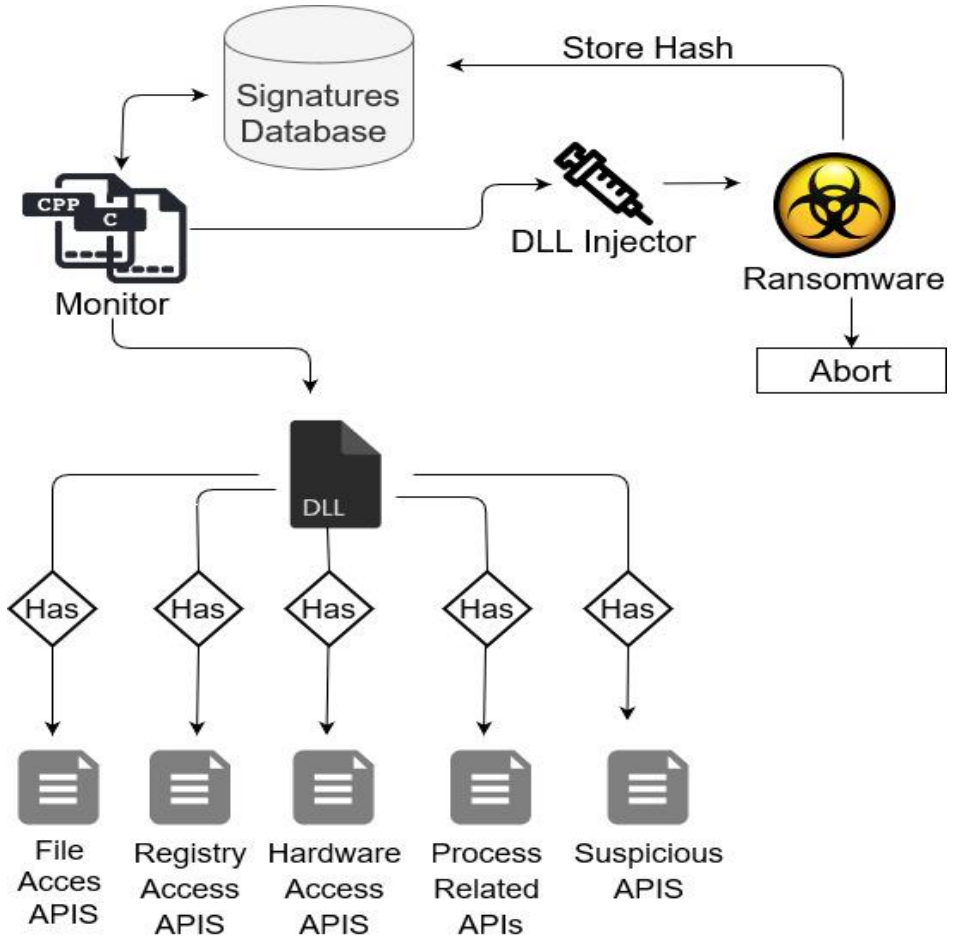
UNIVERSITY OF
South Carolina

College of Engineering
and Computing

ENVIRONMENTAL SETUP

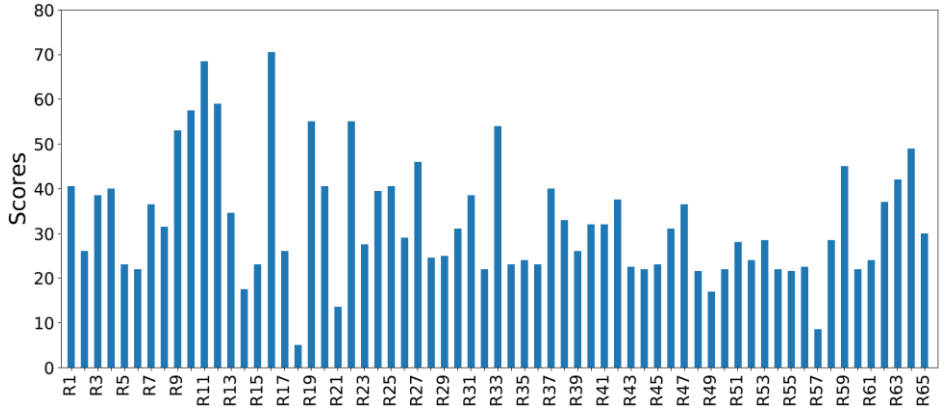
- Our approach is currently designed to operate solely on Windows operating system
- The approach was tested on virtual box running Windows 10 with 8 GB of RAM and 50 GB of hard disk space
- 117 ransomware samples from
  - 30 different ransomware families (wannacry, cryptolocker, locky)
- 98 benign applications
  - built-in Windows applications (notepad, chrome, Skype)
  - Random applications marked as safe by Virus Total

ENVIRONMENTAL SET UP

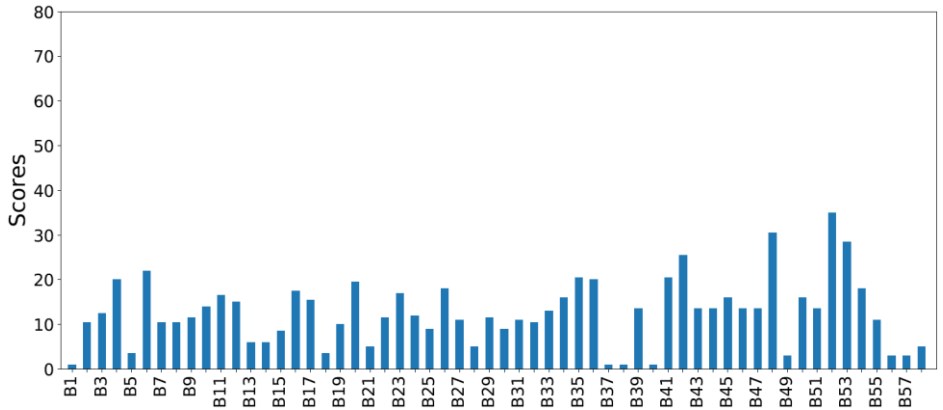


TRAINING DATA

- During this phase, fingerprinting-related APIs were collected and ranked
- The scores of ransomware samples are relatively higher than that of benign samples



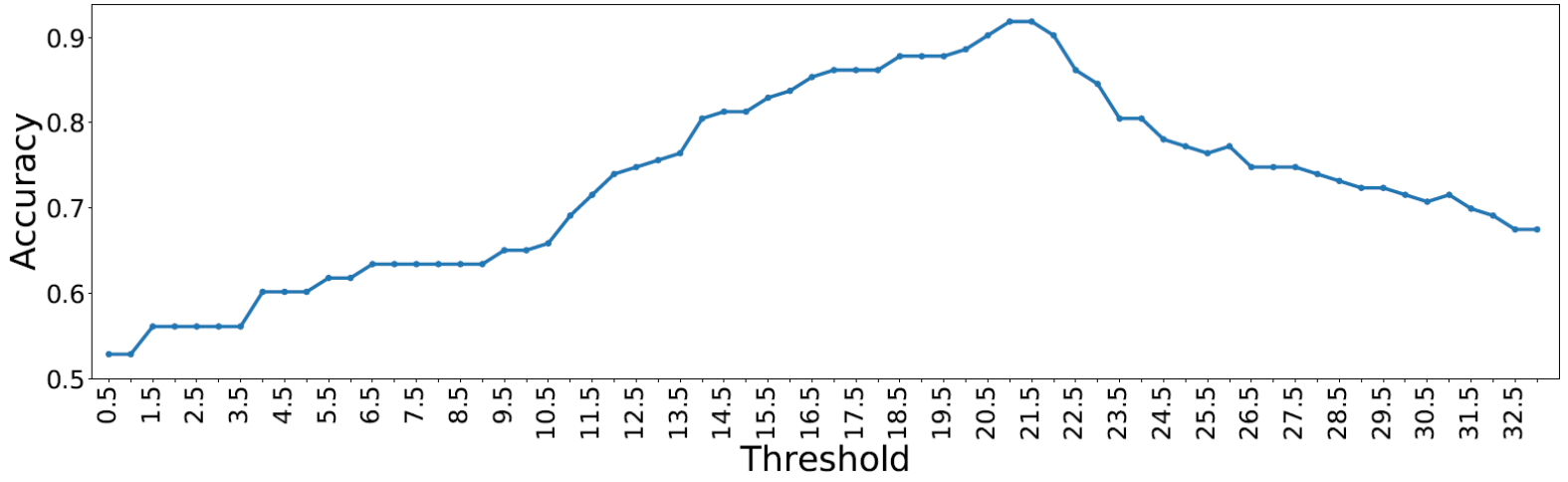
Ransomware samples



Benign samples

THRESHOLD

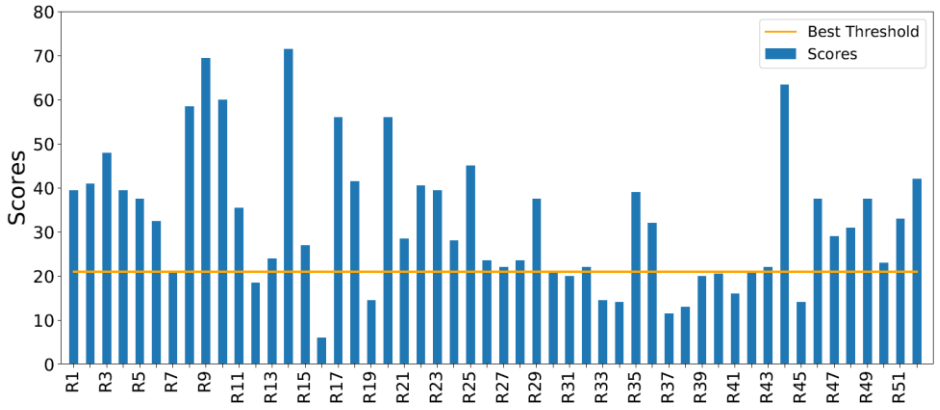
- To differentiate between ransomware and benign samples, a threshold is set
- A threshold value of 21 has the best accuracy (91%) in the training data set
- This value is used as score limit in the testing phase



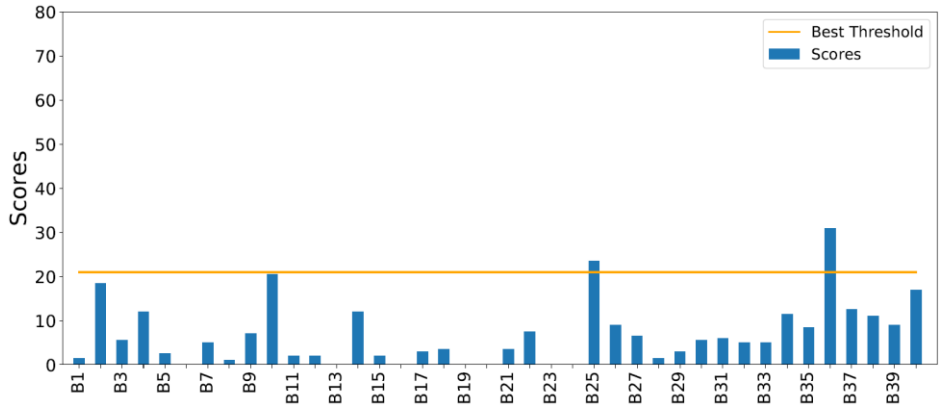
Engineering and Computing

TESTING DATA

- With zero-day ransomware samples, and a threshold value of 21, the accuracy was 84%
- False negative rate = 22%
  - Due to the presence of non-evasive ransomware samples
  - However, can be detect using a regular IDS



Ransomware samples



Benign samples

Engineering and Computing

CONCLUSION AND FUTURE WORK

CONCLUSION

- We addressed evasive ransomware that perform environmental fingerprinting checks
- We explored fingerprinting artifacts on CPU, registries, memory…
- We performed empirical evaluations and showed that our approach is capable of detecting and preventing evasive ransomware

FUTURE WORK

- Conduct extensive evaluations on a broader set of samples and evasive APIs
- Explore deferring techniques to delay/suppress the execution of contemporary ransomware
- Enhance the developed prototype to make it more generic on various operating systems

THANKS!