Static-RWArmor: A Static Analysis Approach for Prevention of Cryptographic Windows Ransomware

Ahsan Ayub, Ambareen Siraj, Bobby Filar, and Maanak Gupta

Tennessee Tech University, Cookeville, USA

Emails: mayub42@tntech.edu, asiraj@tntech.edu, bobby@sublimesecurity.com, and mgupta@tntech.edu

IEEE International Conference on Trust, Security and Privacy in Computing and Communications

(IEEE TrustCom-2023)

CEROC

Cybersecurity Education, Research and Outreach Center

November 1-3, 2023

What is Ransomware?

- A type of malware that takes over the system by affecting the victim machine via email, remote desktop protocol, software vulnerability, etc.
- Mainly two kinds of ransomware –
  - Locker Ransomware
  - Crypto Ransomware

Ransomware Threat Landscape and Motivation

Adversaries use already-developed

- Ransomware-as-a-Service (Raas) Kits

Organizations suffer from

- Financial Loss
- Reputational Loss

Suspected means of initial access

- Software Vulnerabilities
- Credential Attacks
- Phishing
- Abuse of RDP

Ransomware attack on the colonial pipeline network in May 2021

- State of Emergency in 18 states
- USD 4.4 M worth of bitcoin paid

Portable Executable (PE) File Metadata (1/2)

- File Header
  - Target machine type, Timestamp of the file's creation, etc.
- Section Table
  - Name, Virtual address, Size of Row Data, etc.
  - Examples: "text" - the executable code of the program; "data" - the initialized data, etc.
- Import Address Table
  - A map between import libraries and function names.
  - Example: "47363b94cee907e2b8926c1be61150c7" ransomware uses "USER32.dll" import library for CharLower-BuffA", "CharUpperA", etc. functions.

Portable Executable (PE) File Metadata (2/2)

- Export Address Table
  - Functions and values that other PE files can import.
- Resources Directory Table
  - Type of resources available, e.g., manifest, icon, languages, etc.
  - Example: We checked whether “Petya” ransomware family’s sample has any presence of Russian language.

Research Questions (RQ)

RQ1. Do ransomware samples caught in the wild in a calendar year share similar structural information?

RQ2. Can we discover PE file metadata-based dissimilarities between ransomware samples and benign applications?

Distinction from Existing Related Work

Research Paper	Published Year	Hybrid Analysis	Samples' Count		Features			Technique		
			Ransomware	Benign	PE Metadata	OpCode	Hexcode	ML	DL	Yara
Medhat et al. [1]	2018	–	793	878	✓	–	–	–	–	✓
Zhang et al. [2]	2020	–	1,521	92	–	✓	–	–	✓	–
Zhang et al. [3]	2019	–	1,787	100	–	✓	–	✓	–	–
Reddy et al. [4]	2021	–	113	162	–	–	✓	✓	–	–
Hasan et al. [5]	2017	✓	360	460	✓	–	–	✓	–	–
Subedi et al. [6]	2018	–	211	239	✓	–	–	✓	–	–
Poudyal et al. [7]	2018	–	178	178	✓	–	–	✓	–	–
Poudyal et al. [8]	2018	✓	550	540	✓	✓	–	✓	–	–
Poudyal et al. [9]	2019	–	292	292	✓	✓	–	✓	–	–
Shaukat et al. [10]	2018	✓	579	442	✓	–	–	✓	–	–
Our Prior Work [11]	2021	–	727	–	✓	–	–	✓	–	✓
Our Current Work	2023	–	2,436	3,034	✓	–	–	✓	–	–

Collected Ransomware and Benign Applications

- Repository: SOREL-20M and VirusTotal
- 2,436 Ransomware
  - 2017 (and past) – 14 DLLs and 820 EXEs (total: 834).
  - 2018 – 114 DLLs and 592 EXEs (total: 606).
  - 2019 – 26 DLLs and 404 EXEs (total: 430).
  - 2020 – 90 DLLs and 352 EXEs (total: 442).
  - 2021 – 3 DLLs and 21 EXEs (total: 24).
- 3,014 Benign Application
  - A long list of Windows-based applications
  - Cloud-based backup software (Third-party)
  - File-compressing software (Third-party)

Experimental Setup

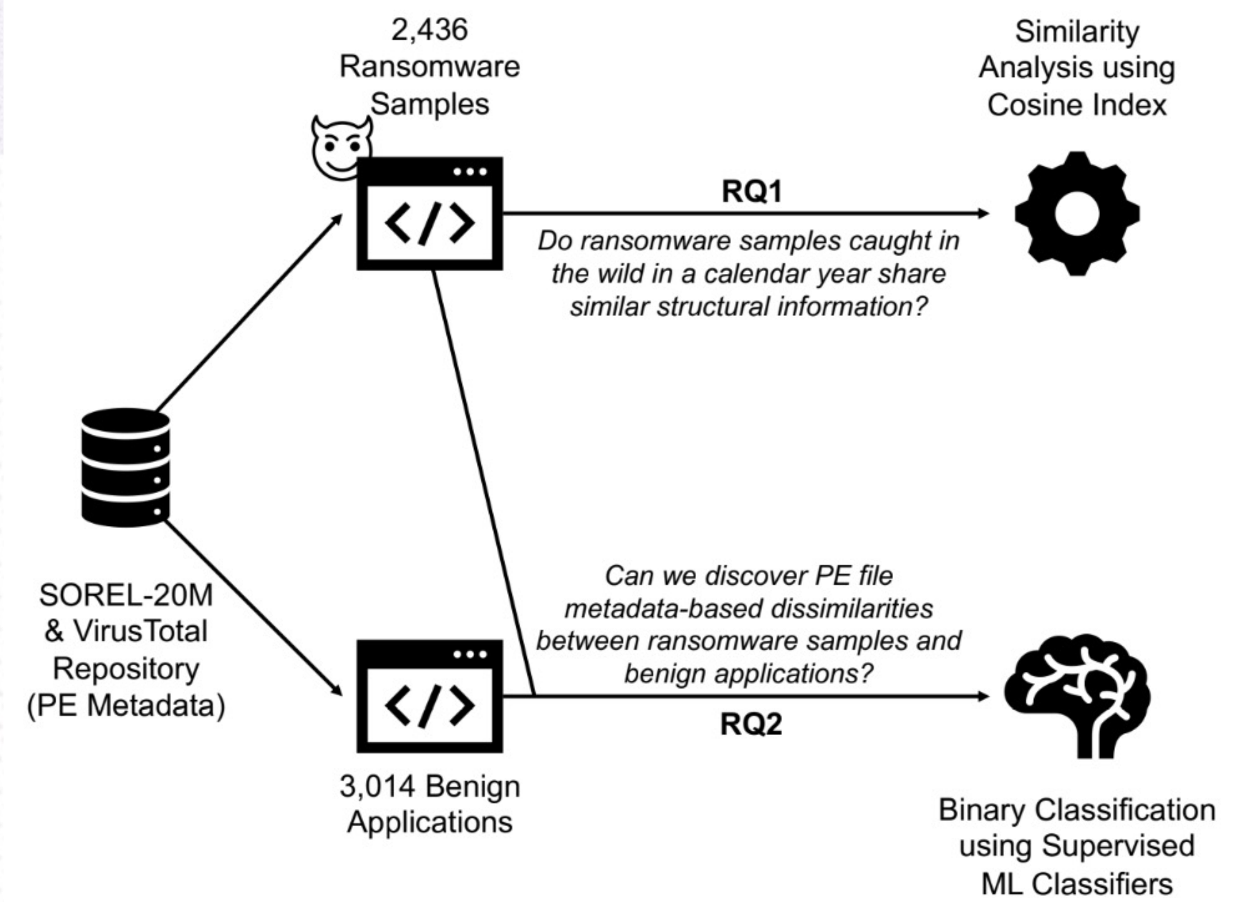

Fig. 1. Experimental methodology of detecting cryptographic ransomware through static analysis and identifying similarities among them.

Methodology to Address RQ1: Similarity Analysis

- We selected “Imports” and ”Function Names” as features spaces to compute similarity among ransomware.
- We chose ”Cosine Index” to report the similarities.

$$\text{Cosine Index} = \frac{xy^T}{||x||||y||}$$

Methodology to Address RQ2: Binary Classification (1/2)

- Selected features for experiments:
  1. Imports: Applied Principal Component Analysis (PCA) on the 2,576 unique numbers of imports for all the ransomware samples and benign applications.
  2. Function Names: Applied Principal Component Analysis (PCA) on the 105,546 unique numbers of function names for all the ransomware samples and benign applications.
  3. Imports and Function Names Combined.
  4. Numeric feature set: For every ransomware and benign application, we compute “imports’ count”, “function names’ count”, “section names’ count”, “sample size”, “Code Size”, “Initialized Data Size”, “Uninitialized Data Size”, and “Resource Languages w/ PCA”.

Methodology to Address RQ2: Binary Classification (2/2)

- We selected tree-based ML classification algorithms: Support Vector Classifier (SVC), Decision Tree, Random Forest, AdaBoost, and Gradient Boosting.
- We utilized Scikit Learn, a machine learning package, for the implementation.

Addressing RQ1 – Similarity Analysis

Year	Imports				Function Names			
	min	median	mean	max	min	median	mean	max
2017	0.095	0.71	0.595	1.0	0.014	0.66	0.55	1.0
2018	0.12	0.76	0.63	1.0	0.019	0.75	0.59	1.0
2019	0.295	0.47	0.55	1.0	0.068	0.35	0.48	1.0
2020	0.34	0.77	0.73	1.0	0.34	0.823	0.77	1.0
2021	0.267	0.878	0.795	1.0	0.08	0.8	0.73	1.0

Fig. 2. Cosine index similarity of ransomware samples per calendar year based on Imports and Function Names.

Addressing RQ2 – Binary Classification (1/4)

Model	Accuracy	Precision	Recall	F1
SVC (rbf)	0.6917	0.6953	0.7009	0.6902
SVC (poly)	0.6416	0.7112	0.6813	0.6369
Decision Tree	0.8179	0.8136	0.8171	0.8136
Random Forest	0.8286	0.824	0.8261	0.824
AdaBoost	0.7721	0.7662	0.7608	0.7628
Gradient Boosting	0.8179	0.8125	0.8151	0.8132

Fig. 3. Performance of a suite of Machine Learning algorithms for binary classification task with **Imports** feature space.

Addressing RQ2 – Binary Classification (2/4)

Model	Accuracy	Precision	Recall	F1
SVC (rbf)	0.7022	0.7372	0.6556	0.6509
SVC (poly)	0.5861	0.293	0.5	0.3695
Decision Tree	0.8712	0.8682	0.8669	0.8669
Random Forest	0.8839	0.8825	0.8781	0.8795
AdaBoost	0.8337	0.8343	0.8215	0.8249
Gradient Boosting	0.8622	0.8595	0.856	0.8571

Fig. 4. Performance of a suite of Machine Learning algorithms for binary classification task with **Function Names** feature space.

Addressing RQ2 – Binary Classification (3/4)

Model	Accuracy	Precision	Recall	F1
SVC (rbf)	0.7063	0.7346	0.662	0.6596
SVC (poly)	0.6323	0.7703	0.5574	0.4877
Decision Tree	0.8667	0.8631	0.8632	0.8625
Random Forest	0.8839	0.8828	0.8777	0.8793
AdaBoost	0.8281	0.8333	0.8111	0.8165
Gradient Boosting	0.8644	0.862	0.858	0.8592

Fig. 5. Performance of a suite of Machine Learning algorithms for binary classification task with **Imports and Function Names** feature space.

Addressing RQ2 – Binary Classification (4/4)

Model	Accuracy	Precision	Recall	F1
SVC (rbf)	0.6207	0.3104	0.5	0.383
SVC (poly)	0.6207	0.3104	0.5	0.383
Decision Tree	0.8911	0.8846	0.8845	0.8839
Random Forest	0.9175	0.9199	0.9047	0.9105
AdaBoost	0.8601	0.8561	0.8457	0.8489
Gradient Boosting	0.9132	0.9107	0.9049	0.9069

Fig. 6. Performance of a suite of Machine Learning algorithms for binary classification task with **PE Numeric** feature space.

Summary

- We investigated how similar ransomware samples collected in the same calendar year are based on the Cosine Index from 2017 to 2021.
- We built ML classifiers to find the structural dissimilarities and achieve 91.75%, 91.99%, 90.47%, and 91.05% at best for accuracy, precision, recall, and F1 scores.
- We encourage the organizations to use the 3-2-1 rule, that is to keep 3 back-ups of their data: 2 on different storage types while 1 on offsite.

References (1/2)

1. May Medhat, Samir Gaber, and Nashwa Abdelbaki. A new static- based framework for ransomware detection. In IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, pages 710–715, 2018.
2. Bin Zhang, Wentao Xiao, Xi Xiao, Arun Kumar Sangaiah, Weizhe Zhang, and Jiajia Zhang. Ransomware classification using patch-based cnn and self-attention network on embedded n-grams of opcodes. Future Generation Computer Systems, 110:708–720, 2020.
3. Hanqi Zhang, Xi Xiao, Francesco Mercaldo, Shiguang Ni, Fabio Mar- tinelli, and Arun Kumar Sangaiah. Classification of ransomware families with machine learning based onn-gram of opcodes. Future Generation Computer Systems, 90:211–221, 2019.
4. Bheemidi Vikram Reddy, Gutha Jaya Krishna, Vadlamani Ravi, and Dipankar Dasgupta. Machine learning and feature selection based ransomware detection using hexacodes. In Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applica- tions (FICTA 2020), Volume 1, pages 583–597. Springer, 2021.
5. Md Mahbub Hasan and Md Mahbubur Rahman. Ranshunt: A support vector machines based ransomware analysis framework with integrated feature set. In 2017 20th International Conference of Computer and Information Technology (ICCIT), pages 1–7. IEEE, 2017.
6. Kul Prasad Subedi, Daya Ram Budhathoki, and Dipankar Dasgupta. Forensic analysis of ransomware families using static and dynamic analysis. In IEEE Security and Privacy Workshops, 2018.
7. Subash Poudyal, Kul Prasad Subedi, and Dipankar Dasgupta. A framework for analyzing ransomware using machine learning. In IEEE Symposium Series on Computational Intelligence, 2018.

References (2/2)

8. Subash Poudyal and Dipankar Dasgupta. Ai-powered ransomware detection framework. In IEEE Symposium Series on Computational Intelligence, 2020.
9. Subash Poudyal, Dipankar Dasgupta, Zahid Akhtar, and K Gupta. A multi-level ransomware detection framework using natural language processing and machine learning. In International Conference on Malicious and Unwanted Software" MALCON, 2019.
10. Saiyed Kashif Shaukat and Vinay J Ribeiro. Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. In 2018 10th International Conference on Communication Systems & Networks (COMSNETS), pages 356–363. IEEE, 2018.
11. Md Ahsan Ayub and Ambareen Siraj. Similarity analysis of ransomware based on portable executable (pe) file metadata. In IEEE Symposium Series on Computational Intelligence, pages 1–6, 2021.

Acknowledgement

The work reported in this paper has been supported by Cybersecurity Education, Research & Outreach Center (CEROC) at Tennessee Tech University. Additionally, it is partially supported by the NSF grants 2025682 and 2230609.

THANK YOU!

Implementation

https://github.com/AhsanAyub/deep_static_ransomware_analysis