Poster: Leveraging Large Language Models for Detecting OS-level Ransomware

Zhuoyan Xu, Josh Dafoe, Bo Chen
Department of Computer Science, Michigan Technological University, Houghton, MI, USA
{zhuoyanx, jwdafoe, bchen}@mtu.edu

***Abstract*—This work aims at exploring the possibility of using large language models to assist in ransomware detection. Especially, we focus on the OS-level ransomware, which can compromise the OS of the victim device and is hence much more difficult to combat. Our preliminary results are encouraging. By fine-tuning a light-weight BERT model, our design has a detection accuracy of 94.1% with a false negative rate 2.3% and a false positive rate 25%.**

***Index Terms*—OS-level ransomware, flash translation layer, detection, large language model**

I. Introduction

As a special type of malware, ransomware either locks the victim systems or encrypts the victim data for extortion. In recent years, it has escalated into a global cybersecurity crisis, and active research has been conducted to mitigate it. A critical type of ransomware is high-privilege ransomware which can compromise the root privilege of the victim systems. This OS-level ransomware is hard to combat as it can simply disable the normal malware detection tools running within the OS.

Flash storage media such as SSDs, microSD cards, and MMC cards have been widely used in various types of computing devices today due to their much higher I/O throughput and lower energy consumption compared to traditional mechanical disks. The core of any flash storage medium is the flash translation layer (FTL), a firmware layer that stays between the file system and the raw flash memory. By transparently managing the unique hardware characteristics of flash memory, FTL exposes a block access interface, so that the OS can simply use a flash storage medium as a block device. To compromise data on a victim computing device, the ransomware always needs to perform I/Os over the external storage, and the I/Os will unavoidably go through the FTL. Therefore, any ransomware I/Os can be monitored inside the FTL and, most importantly, this monitoring software logic will remain intact even if the ransomware can compromise the root privilege. This makes it possible to integrate a ransomware detector within the FTL that can combat strong OS-level ransomware [1]. However, existing work [1] suffers from low accuracy and high false positive rate by applying traditional AI methods (e.g., k-Nearest Neighbors) for ransomware detection.

Large language models (LLMs) such as ChatGPT have become increasingly popular in recent years. Unlike traditional AI models, the language models extract the semantics from a vast amount of text data, making it much more superior in various aspects including reasoning, math, coding, etc. LLMs have been applied to a large number of applications, including malware detection [4]. However, they are rarely investigated in combating ransomware, especially the OS-level ransomware. This work aims to explore the possibility of integrating LLMs for FTL-based OS-level ransomware detection. We especially focus on BERT [3], because: 1) BERT is a light-weight LLM which is more suitable for low-power embedded systems, and 2) BERT supports predicting the masked tokens given its context, which well matches the ransomware detection scenario, as the ransomware I/Os captured by the FTL are the context and what needs to be predicted is a masked token, either "ransomware" or "benign software".

II. Background

Flash translation layer (FTL). Flash memory dominates the storage media of modern computing devices. It has a few unique hardware characteristics, such as erase-before-write and susceptibility to wear. FTL implements a few unique functions such as address translation, wear leveling, garbage collection, to transparently manage special hardware nature of flash memory, exposing a block access interface to the operating system running on top of it.

Large language model (LLM). An LLM is a large AI model that has been trained on a large amount of language data. Bidirectional encoder representations from transformers (BERT) is a pioneering LLM introduced in 2018. It uses an encoder-only transformer architecture and was pre-trained simultaneously on two tasks, masked language model (MLM) and next sequence prediction (NSP).

III. System and Adversarial Model

We consider a computing device equipped with a flash-based block device, such as solid-state drives (SSD), SD cards, micro SD cards, eMMC cards, etc. The ransomware can compromise the operating system running on the victim device, obtaining root privileges. Therefore, any ransomware detection software running at the privilege level ring 0 or higher would not be secure.

IV. Our Design

To detect high-privilege ransomware that is capable of compromising the operating system of the victim device, the malware detector needs to function within the FTL. Especially, the malware detector monitors the I/Os issued from the OS and determines whether there is ransomware present in real time. This requires: 1) extracting a collection of continuous I/O traces from the FTL, and 2) determine whether there is ransomware, using the collection of I/Os and a pre-trained classifier.

To develop this pre-trained classifier, we leverage large language models. As the large language model is not originally designed for detecting ransomware on the FTL, we should fine-tune a pre-trained LLM using unique ransomware I/O traces captured on the FTL. Our fine-tuning process includes a few key steps, elaborated below:

Dataset preparation: We first collect ransomware I/O traces in the FTL. Each trace should contain sufficient information to identify an I/O access, e.g., the nature of the operation, I/O location, and I/O length. The traces can be collected by running ransomware samples and capturing the respective I/O traces on the FTL.

Tokenization and encoding: Each I/O trace consists of operation type (0 for read, 1 for write), starting address, and operation length. The tokenizer can effectively parse each I/O trace based on the provided vocabulary, generating tokens. Encoding further converts the tokens into token IDs suitable for the LLM embedding layer.

Fine-tuning: On top of the pre-trained model's [CLS] token output, we add a classification layer to distinguish between the two classes: label 0 for benign software and label 1 for ransomware. Given this extended model, we simply use the prepared dataset to fine-tune the model.

V. Preliminary Experimental Evaluation

Ransomware I/O trace dataset. Chen et al. [2], [1] provided a malware I/O trace dataset "MITON", which contains I/O traces captured from open-source FTL when ransomware or regular software (including compression/ encryption/ deletion software, etc.) is running. We generated a new ransomware I/O trace dataset by 1) extracting traces specific for ransomware from MITON, and 2) adding I/O traces of additional ransomware samples of recent years, collected from MalwareBazaar and VirusShare, and 3) reusing part of the benign software samples. The extra ransomware I/O traces were collected from open source FTL following the guideline of Chen et al. [2]. Our final dataset for experimental evaluation contains 139 ransomware samples and 31 benign software samples.

Fine-tuning BERT for ransomware detection. We used Google BERT for our experimental evaluation. Especially, we used the pre-trained bert-base-uncased (110M parameters) available in Hugging Face. We fine-tuned it to support ransomware detection on the FTL. We utilized the pre-trained BertTokenizer from bert-base-uncased to tokenize and encode the I/O traces. The ransomware I/O trace dataset was divided into 70% training data and 30% test data.

Accuracy	Precision	Recall	F1	FNR	FPR
0.941	0.955	0.977	0.966	0.023	0.250

TABLE I: Preliminary experimental results.

The bert-base-uncased model was fine-tuned by running the trainer class of Hugging Face Transformers using the training data. As our input is a time sequence, which is more complicated than normal text input, we set the training parameters to train for 10 epochs using a batch size of 8. We also used weight decay and warm-up steps to improve generalization and convergence. In our experiment, we added a newly initialized classification layer followed by a softmax function on top of the [CLS] token's output to perform binary classification. In addition to preparing and tokenizing the input data, we also fine-tuned the entire paramaters, including the pretrained BERT layers.

Experimental results. The preliminary experimental results are presented in Table I. We have a few observations: 1) The effectiveness of our ransomware detection approach is demonstrated by high accuracy (94.1%), precision (95.5%), recall (97.7%) and F1 score (96.6%). 2) Our approach maintains a low level of "detection missing" (false negative rate 2. 3%), while the false positive rate is acceptable (25%).

VI. Discussion

Incorporating an LLM into the FTL may present some challenges. Specifically, the limited hardware capabilities of a flash memory controller may restrict the feasibility of efficiently running an LLM within the FTL. This limitation should be evaluated and addressed in future work. In addition, future work should explore the use of a larger training dataset, which is more balanced, to make the solution more robust. Furthermore, our approach should be evaluated against previous works targeting in-FTL ransomware detection.

Acknowledgment. This work was supported by the US National Science Foundation under grant numbers 2225424-CNS and 2043022-DGE.

References

[1] N. Chen and B. Chen. Defending against os-level malware in mobile devices via real-time malware detection and storage restoration. *Journal of Cybersecurity and Privacy*, 2(2):311–328, 2022.

[2] N. Chen, W. Xie, and B. Chen. Combating the os-level malware in mobile devices by leveraging isolation and steganography. In *Proceedings of Applied Cryptography and Network Security Workshops*, pages 397–413. Springer, 2021.

[3] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] P.M.S. Sánchez, A.H. Celdrán, G. Bovet, and G.M. Pérez. Transfer learning in pre-trained large language models for malware detection based on system calls. In *Proceedings of MILCOM*, pages 853–858. IEEE, 2024.

Poster: Leveraging Large Language Models for Detecting OS-level Ransomware

Zhuoyan Xu, Josh Dafoe, Bo Chen
Department of Computer Science, Michigan Technological University

Michigan Tech

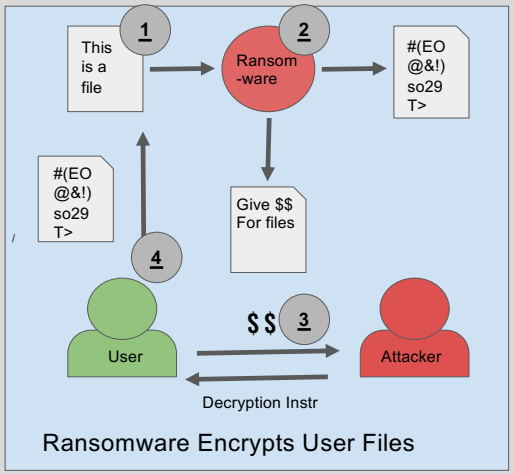
SnP Lab

Abstract

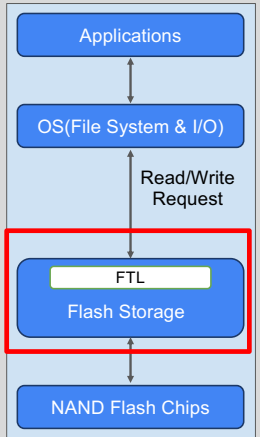
This work aims at exploring the possibility of using large language models to assist in ransomware detection. Especially, we focus on the OS-level ransomware, which can compromise the OS of the victim device and is hence much more difficult to combat. Our preliminary results are encouraging. By fine-tuning a light-weight BERT model, our design has a detection accuracy of 94.1% with a false negative rate 2.3% and a false positive rate 25%.

Introduction and Background

- The **Flash Translation Layer** (FTL) is the flash storage firmware layer.
- FTL is **isolated** from the host OS, preventing compromize from OS level ransomware.
- The FTL **processes all I/O operations**. This includes the ransomware encryption operations.



Ransomware Encrypts User Files



FTL Isolation

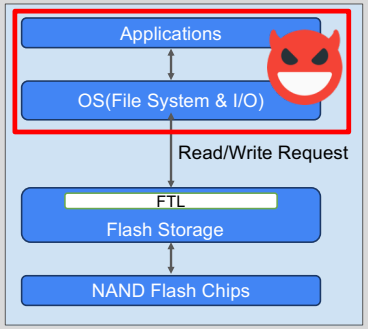
- **Ransomware will encrypt user files.** Some modern variants will have OS level privileges.

Large Language Models

- **Large Language Models** (LLMs) have demonstrated exceptional semantic reasoning capabilities.
- The BERT model uses an encoder-only transformer architecture and was pre-trained simultaneously on masked language model (MLM) and next sequence prediction (NSP).

Model

- We consider a computing device equipped with a flash-based block device.
- **The ransomware can control the operating system**
- Therefore, any ransomware detection software running at the privilege level ring 0 or higher would not be secure.



Preliminary Evaluation

We extended an existing dataset "MITON" significantly, to now incorporate I/O traces from 139 ransomware samples and 31 benign samples.

Next, we fine-tuned the model bert-base-uncased, and evaluated different metrics.

Accuracy	Precision	Recall	F1	FNR	FPR
0.941	0.955	0.977	0.966	0.023	0.250

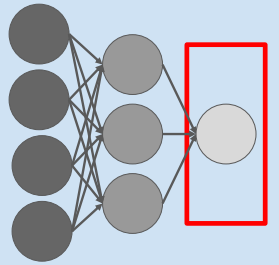
The evaluation of our fine-tuned model.

We observe that **almost all ransomware files are successfully detected.**

Design

- We aim to enable ransomware detection within the FTL, which is isolated from the host OS.
- The FTL is limited to observing the I/O requests to the flash storage.
- The ransomware detector will monitor I/O in real time
- To do this, the ransomware detector will need to do the following:
  - Extract continuous I/O traces from the FTL.
  - **Use I/O data and a pre-trained classifier to determine if there is ransomware.**

1. **Dataset preparation:** Collect the I/O traces
2. **Tokenization and Encoding:** Map the I/O data to corresponding tokens.
3. **Fine Tuning:** We add a classification layer to perform binary classification. We fine-tune the extended model.



Our Additional Binary Classification Layer

This work was supported by the US National Science Foundation under grant numbers 2225424-CNS and 2043022-DGE.