ABSTRACT

Malware attack is a very large domain of cyber security attacks. This project aims at developing a model which accurately predicts the probability that Windows operating system will be hit by a malware. It works on an operating systems dataset that has over 7 million recorded operating systems and their various features, generated by combining heartbeat and threat reports collected by Microsoft's endpoint protection solution, Windows Defender. In order to understand the working of a wide variety of models on such a problem, three different models will be developed and assessed for its accuracy at predicting malware. Three different models under consideration are recurrent neural network, LightGBM technique and lastly a factorization model over a convolutional neural network known as XGBoost. This approach is assessing vulnerability of the system rather than the attacker. If the attacker is constantly evolving and learning new techniques against the system’s defence, then efforts to defend against certain types of attacks are futile. Hence, predicting an attack in a more generic sense before it has even happened by assessing the system itself is the better alternative. Even though there are variants, a malware always targets a vulnerability or an exploit of the system to attack. If these weak points on the system are found and patched up before an attack happens, we can develop a very secure and malware proof security configuration.

**INTRODUCTION**

1.1 General Introduction

Malware is software that is aimed at intentionally causing a system to behave in a way that it should not. Its main objective is to disrupt the system’s normal functioning or cause damage to the system itself and its components or both. Malware comes in many different forms, but irrespective of the type, their objective is one - damage to a system. Malware attack is a very large domain of cyber security attacks. Cryptanalysts across the world has been in a long drawn out battle which has intensified in the past decade with malware. With increase in the technological capabilities of computers, there is also a distinct sharp increase in the capabilities of what malware can do and more importantly, how a malware can prevent from being detected.

There are two main types of defense against malware: malware detection and malware prevention. This work focus on the former and remove the problem before it even arises. There are over a billion potential systems in the world that potentially be affected by malware. Designing a technique to prevent attacks from all different types of malware can be extremely difficult as compared to detecting the causes of malware and predicting if a machine will be hit with malware.

If we check and analyze different types of malware that exist at this point in time, we may come up with a very efficient system to guard against attacks. However, this is very difficult as it requires documentation of every malware that has ever existed not to mention the very complicated and intricate system which requires different techniques against different types of malware. This makes malware detection essentially a time-series problem, but it is made complicated by the introduction of new machines, machines that come online and offline, machines that receive patches, machines that receive new operating systems, etc.

This leaves us in a predicament - stopping malware attacks by preparing ourselves against the different types of malware is practically impossible because it has to done exhaustively. Not doing so puts the system at great danger which defeats the purpose of defending a malware attack. Also, when technology grows leaps and bounds, so does the capability of malware. True exhaustive analysis of malware cannot be done in theory or practice. The list of malwares is constantly growing and evolving. True preparation against attacks takes a whole different approach.

This approach is assessing vulnerability of the system rather than the attacker. If the attacker is constantly evolving and learning new techniques against the system’s defense, then efforts to defend against certain types of attacks are futile. Hence, we try to predict an attack in a more generic sense before it has even happened by assessing the system itself. This theory is mainly based on the assumption that malware is targeted. Even though there are variants, a malware always targets a vulnerability or an exploit of the system to attack. If these weak points on the system can be found out and patch them before an attack happens, a very secure and malware proof security configuration can be developed.

.

1.2 Problem Statement

The project aims at developing a model which accurately predicts the probability that an operating system will be hit by a malware. Its primary goal is to malware detection among many operating systems and building a model which can accurately do the task at hand.

1.3 Objective

There are two main types of defence against malware - malware detection and malware prevention. We focus on the former and remove the problem before it even arises. There are over a billion potential systems in the world that potentially be affected by malware. Designing a technique to prevent attacks from all different types of malware can be extremely difficult as compared to detecting the causes of malware and predicting if a machine will be hit with malware.

We work on an operating systems dataset that has over 7 million recorded operating systems and their various features. We create a model that tries to predict which of those operating systems will be hit with any type of malware. Essentially, the model will predict the vulnerability of the system and the probability that the system will be affected soon. The dataset will have actual results of the systems after receiving status of if they’re affected which is when we can decide and calculate the accuracy of our model.

Rather than inspecting malware, we can get much better results by analyzing the system and its own vulnerabilities against targeted attacks from malware. The outcome is a predictive model that outlines the probability of a system with many parameters being affected in the near future or not. A generic solution that fits all the systems will help plug holes in many security systems - primarily windows machines as they are the most used operating system on the planet currently.

1.4 Project Deliverables

* A model that shows maximum accuracy in order to predict the presence of malware
* A seamless user interface to navigate the user to use the above said trained model.
* Incorporate information about all possible scenarios that could be present in Windows systems.

1.5 Current Scope

Malware is software that is aimed at intentionally causing a system to behave in a way that it should not. Its main objective is to disrupt the system’s normal functioning or cause damage to the system itself and its components or both. Malware comes in many different forms, but irrespective of the type, their objective is one - damage to a system.

Malware attack is a very large domain of cybersecurity attacks. Cryptanalysts across the world has been in a long drawn out battle which has intensified in the past decade with malware. With increase in the technological capabilities of computers, there is also a distinct sharp increase in the capabilities of what malware can do and more importantly, how a malware can prevent from being detected. This is what our project is aimed at - Malware detection.

1.6 Future Scope

Malware can be very devastating depending on the target entity. Key-loggers can gain confidential information on entire organizations which can have monetary impact of billions of dollars. Viruses like stuxnet can change the fate of an entire country by altering war. Catching a virus before it hits a system will improve efficiency and simplify code base of various anti virus deployments. We intend to build a model for predicting malware existence in operating systems with an impeccable accuracy. Future work could comprise of combining the works of detecting malware from the system codes written by understanding their semantics to help predict. We could also classify the malware being detected along with predictions.