**Final Project:**

Yu-Chun Chen, A13356506

Yanyu Tao, A13961185

Bolin Yang, A92111272

Shuibenyang Yuan, A14031016

Haoming Zhang, A14012520

Mingxuan Zhang, A13796895

**Introduction:**

The war against malicious software, or malware, is one of the most significant topics in the cyber security domain since it gradually becomes a well-organized market. When malware evades the computer system, the virus can harm users. This malware problem related to everyone using computers from individuals to large companies and agencies. The malware becomes more sophisticated and can frustrate the antivirus software through encryption like inserting dead code, substituting instructions, like reordering the routine to target computers and mobile devices. So, the advanced malware is hiding inside then replicate the host protections, which is unknown and hard to discover. After installation, the infectious software of malware steals the crucial private information, such as the e-mail password and the bank account, and infects other software.

The diversity and concealment of the malicious software undermine the efficiency of traditional antivirus defences using the signature-based. With the vast consumer and enterprises basis, Microsoft invests billion of money into this field and continues updating the operating system to improve the security. To counteract to such malware threats, Microsoft wants to develop new techniques to forecast whether the machine will be hit with malware. As with their previous, Malware Challenge (2015), Microsoft is providing Kagglers with a unique malware dataset to encourage open-source progress on practical techniques for predicting malware occurrences (Kaggle).

**In this project, we are going** **to assess the vulnerability of the machine (computer, server, etc.) based on its configuration.** The goal is to predict the probability of a Windows machine getting infected by different types of malware, based on various properties of that machine (Kaggle).

There are some related works. Since the cyber-threat associates with every consumer using the computers, many researchers tried similar attempts to our project, which quantify the risk to reduce the malware attack. In “Ensemble Models for Data-driven Prediction of Malware Infections,” the author investigated more than 1.4 million hosts and 50 malware in the past two years worldwide. They used ESM, an ensemble-based approach combining with the non-linear model for malware spread. They found that ESM is relatively stable and accurate even though the dataset is small. G.J. Tesauro and other two researchers from IBM also investigated malware detections of boot sector viruses from the neural networks. They applied neural network into their antivirus software, which protects millions of computer users (G.J. Tesauro, J.O. Kephart, and G.B. Sorkin). Instead of the tradition signature-based method, M.G. Schultz et al. presented a data mining framework to create a much faster detection rate. And this method can detect new malicious software, which is a vast improvement from the past (M.G. Schultz, E. Eskin, F. Zadok, and S.J. Stolfo).

**Data:**

Based on different properties of the machine, the probability of a Windows machine getting infected by various families of malware would vary in a certain amount. So, our dataset contains these properties, and the machine infections were generated by combining heartbeat and threat reports collected by Microsoft's endpoint protection solution, Windows Defender. Using the information and labels in train.csv, our object is to predict the value for HasDetections for each machine in test.csv (Kaggle).

The sampling methodology used to create this dataset was designed to meet certain business constraints, both regarding user privacy as well as the period during which the machine was running. Malware detection is inherently 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. Additionally, this dataset is not representative of Microsoft customers' devices in the wild; it has been sampled to include a much more significant proportion of malware machines (Kaggle).

Selected Columns:

|  |  |  |
| --- | --- | --- |
| Variable | Description | Variable Type |
| Census\_ProcessorCoreCount | Number of logical cores in the processor | numerical |
| Census\_PrimaryDiskTotalCapacity | Amount of disk space on primary disk of the machine in MB | numerical |
| Census\_SystemVolumeTotalCapacity | The size of the partition that the System volume is installed on in MB | numerical |
| Census\_TotalPhysicalRAM | Retrieves the physical RAM in MB | numerical |
| Census\_InternalPrimaryDiagonal  DisplaySizeInInches | Retrieves the physical diagonal length in inches of the primary display | numerical |
| Census\_InternalPrimaryDisplay  ResolutionHorizontal | Retrieves the number of pixels in the horizontal direction of the internal display | numerical |
| Census\_InternalPrimaryDisplay  ResolutionVertical | Retrieves the number of pixels in the vertical direction of the internal display | numerical |
| Census\_InternalBatteryNumberOf  Charges | NA | numerical |
| HasDetections | the ground truth and indicates that Malware was detected on the machine | categorical |
| Platform | Calculates platform name (of OS related properties and processor property) | categorical |
| Census\_IsVirtualDevice | Identifies a Virtual Machine (machine learning model) | categorical |
| AVProductsInstalled | NA | categorical |
| AVProductsEnabled | NA | categorical |
| SMode | This field is set to true when the device is known to be in 'S Mode', as in, Windows 10 S mode, where only Microsoft Store apps can be installed | categorical |
| Firewall | This attribute is true (1) for Windows 8.1 and above if windows firewall is enabled, as reported by the service. | categorical |

*Table 1: Detailed Information of Variables of Data*

In the columns, if there is "NA," it is unavailable or self-documenting columns.

**Background:**

Cyber attackers create malicious software, or called malware, as threatening software to cause damage or gain the essential data to a computer. In the past years, the malware market reproduces, and it becomes one of the most pressing cybersecurity issues. Hackers improve their obfuscation technology and coding method to damage the tradition protection of antivirus, which forces antivirus companies to enhance their products defencing the malware attacks. The major challenge that must overcome is to distinguish which part in the software is malware within the massive amounts of data and files.

Microsoft continues generating and running the anti-malware utilities over 150 million computers around the world, which collects millions of data to be analyzed as the potential malware. In order to analyze these data effectively, we classify them into different groups or families. This process is not an easy procedure, and the classifications are still expanding. This classification is based on their propagation process and achieved actions on the infected system (M. Asha Jerlin and C. Jayakumar). The following are the various types of malware: viruses, worms, trojans horse, spyware, adware, backdoors, rootkits, sniffers, reverse code engineering, disassemblers, debuggers, and decompiler.

The current malware analysis is the malware detection technique to catch malicious behavior in computer software and related files. There are two categories of malware analysis, one is the static method, and the other one is the dynamic method. The most significant difference between these two methods is that static method acquires the vital information by disassembling code, while dynamic method uses runtime trace report of executable files to get information (M. Asha Jerlin and C. Jayakumar). Static analysis analyzes malicious software without executing it and studies each component. Disassembler like IDA disassembles the binary files and translates the malware's code into the assembly code. This process makes an image for the analyst, which can provide the details and particular patterns to classify the family of malware. Dynamic malware analysis is the opposite method of the static method. The dynamic method analyzes malicious files and software when they are running. During the process, infected files are simulated in the sandbox environment to avoid real infection into the system. After detecting the general behavior of malware, it may be debugged during the execution. However, the biggest problem is that dynamic analysis requires much time since researchers should prepare the exact environment for malware. And this method elevates the scalability issues (Ekta Gandotra, Divya Bansal, Sanjeev Sofat).

The malware detection technique is the essential process to analyze the role of a malware. There are two methods for malware detection, signature-based, and heuristic based. Most traditional antivirus software uses Signature-based detection. Once the malware is identified and analyzed by the system, its signature will be recorded and added to the database. Then, the computer can recognize the same malware again. However, the signature detection method has limited that there are more and more virus and virus can evolve. So, the Heuristic method analyzes the unknown malware or the new derivatives of the known malware. And it needs to upgrade the data continuously, which means that it requires a high level of CPU and other software.

**Investigation:**

Scenario 1:

Scenario 2:

Scenario 3:

Scenario 4:

Scenario 5:

Additional Hypothesis:

**Works Cited:**

AshaJerlin, M., & Jayakumar, C. (2015). “A Dynamic Malware Analysis for Windows Platform-

A Survey.” *Indian Journal of Science and Technology*, *8*(27), 1.

Chanhyun Kang, Noseong Park, B. Aditya Prakash, Edoardo Serra, and V. S. Subrahmanian.

(2016). “Ensemble Models for Data-Driven Prediction of Malware Infections.”

In *Proceedings of the 2016 ACM International Conference on Web Search and*

*Data Mining.* ACM.

Gandotra, E., Bansal, D. and Sofat, S. (2014). “Malware Analysis and Classification: A Survey.” *Journal of Information Security*, **5**, 56-64. doi: 10.4236/jis.2014.52006.

Kaggle. “Microsoft Malware Prediction: Can You Predict if a Machine Will Soon Be Hit with

Malware.” https://www.kaggle.com/c/microsoft-malware-prediction.

Schultz, M., Eskin, M., Zadok, E., and Stolfo, F. (2001). “Data Mining Methods for Detection of

New Malicious Executables.” In *Proceedings of 2001 IEEE Symposium on*

*Security and Privacy*, IEEE, Oakland, CA, May 2001, 38-49.

DOI = 10.1109/SECPRI.2001.924286

Tesauro, G.J., Kephart, J.O., Sorkin, G.B. (1996). “Neural Networks for Computer Virus

Recognition.” IEEE Expert 11(4), 5–6 (1996)