Parth Parab
10444835
December 7, 2020

Early-stage malware prediction using recurrent neural networks

Review Essay - CS 573 Assignment 2

In this day and age where most of the wars are fought online in the form of cyber attacks and cyber ransoms, it has become all the more important for governments, private companies and even everyday users to secure their digital devices from being caught up in it. Recent events like the WannaCry attack in 2017 which was able to generate a financial loss of billions of dollars, have raised concerns about the security of computer systems and stirred the need for more effective malware detection software. The aim of this paper is to take into consideration the ever changing behavior of malware and build a system to predict its behavior using artificial recurrent neural networks. Using snapshots of behavioral data, unlike static malware analysis which examines the file after it's been installed, this system is able to predict whether an executable is malicious or benign within the first 5 seconds of execution with 94% accuracy. In this essay, we discuss the capabilities of using active machine data to implement machine learning solutions for developing a system which not only detects existing malware in the system which tends to obfuscate the code to pass as design data but also discerns previously not seen 'zero-day' malware with no existing footprint.

The need for such a revised and advanced malware detecting system arises because of two reasons - Rising threat of cyber-attacks as more and more devices get connected to the internet and the incapability of endpoint anti-virus systems to protect against such vulnerability. In anti-virus programs, automated malware detection usually compares features extracted from the code of an incoming file to a documented list of malware signatures. However, unless it shares the code with previously established strains, this method of filtering using static data is not appropriate for detecting completely new ('zero-day') malware. Malware nowadays perform obfuscating of code so that they pass undetectably by the anti-virus system. Alternatively, approaches with a behavioral analysis consider that malware cannot escape leaving a visible

footprint as a consequence of the actions required to achieve its goals. In contrast with static analysis, although dynamic data can lead to more precise result, executing the malware involves a time penalty. While a single file is evaluated, and eventually, by the end of the review window, the malicious payload is possibly delivered, so the chance to block malicious behavior is missed. Furthermore, there are also certain approaches which track "live" operation on the local network or machine to prevent this wait time. These detection systems tend to either search for characteristics that suggest a specific form of malware (e.g., ransomware) or flag deviations from a "standard" conduct base line. Both of these methods suffer from particular weaknesses. Searching for specific behaviors is similar to conventional methods of comparing incoming files with known variants and new types of malware which fail to be identified. In reality, it all comes down to human analysts who often examine anomalous events, leaving the model vulnerable to human error.

The paper suggests building a method that could be integrated into an end-point solution, a behavior-based model to predict whether a file is malicious during the first few seconds of file execution. In order to predict malicious behavior using computer activity data, The authors have used a recurrent neural network (RNN) model to demonstrate that its capabilities are superior to other machine learning solutions previously used for malware detection because they are able to process time series data, thus capturing information that changes over time as well as raw input feature values. To advance malware detection to a more accurate approach that can respond in seconds, the authors have suggested a model that uses only short sequences of the original dynamic data to investigate whether this is adequate to judge a file with a high degree of accuracy as malicious. For feature inputs in the model the function uses 10 computer activity data metrics. For 20 seconds, it takes a snapshot of the metrics every second while the sample runs, starting in 0 seconds, so that we have two feature sets or a 2 sequence length at 1 second. The metrics used were - system CPU usage, user CPU use, packets sent, packets received, bytes sent, bytes received, memory use, swap use, the total number of processes currently running and the maximum process ID assigned. In order to train this model, the authors have used VirusTotal which obtained 1000 malicious and 600 "trusted" Windows 7 executables along with 800 trusted samples from device files of a new 64-bit installation of Windows 7. The reason for using VirusTotal is because it runs files through around 60 anti-virus engines and reports the number of engines that detected the file as malicious. The final dataset comprised of 2345 benign and 2286 malicious samples. On running the model on different Machine Learning Algorithms apart from RNN for behavioral malware classification: Random Forest, J48 Decision Tree, Gradient Boosted Decision Trees, Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbor and Multi-Layer Perceptron algorithms the accuracy pattern shows that the 10-fold

cross-validation on the training set and the test set progresses as the execution time progresses. Over the 20 seconds of execution on the training set, Random Forest achieves the highest accuracy, but after 1 second of execution, the RNN achieves the highest precision on the unseen test set and exceeds all other algorithms on the unseen test set. The RNN consistently performs better than all other algorithms after 1 second. The RNN correctly classifies 91% of unseen samples using 4 seconds of data and achieves 96% accuracy at 19 seconds in execution, while the maximum accuracy is 92% expected by any other algorithm at any time. It continues to maintain this level of accuracy when simulating 'zero-day' malware detection.

In my opinion, the work done in this paper is remarkable and in the right direction to keep our systems safe. Using Machine Learning to learn patterns of malware behavior in-order to detect them within 5 seconds of execution to ensure that the file is flagged and removed before it gets to sensitive information the system might have is very intuitive. While this paper mostly focuses on API Data calls it would be reasonable to also focus on other aspects of malware activity like Network data and other sensitive data it accesses during execution. Although, this model does perform poorly for specific versions of Trojans, with enough time and data training this can be overcome. One of the aspects which sets it aside from other such systems is it ability to detect ransomware, which is a growing threat. The authors were able to detect and block all of the 492 ransomware samples tested with less than 33% of user data being lost in each instance.

In conclusion, this paper has presented a new and improved technique to replace traditional malware detection systems by implementing a dynamic malware detection approach with a low time penalty by snapshotting the footprint of the executable file. It has also shown us how we can use Neural Networks to detect such malware which tends to obfuscate its signature and mask it's behavioral data which opens new horizons of using Machine Learning and it's applications to make our systems secure and ready to face new and dangerous malware.

References :

Paper:

https://www.sciencedirect.com/science/article/pii/S0167404818305546?via%3Dihub

Other links:

https://core.ac.uk/download/pdf/46908978.pdf

https://www.csoonline.com/article/2615925/security-your-quick-guide-to-malware-types.html

https://en.wikipedia.org/wiki/WannaCry_ransomware_attack