ABSTRACT

The increasing prevalence of ransomware attacks, as well as the advent of new variations, constitutes a severe danger to cybersecurity. Phishing emails, malicious software, illicit downloads, and the exploitation of system vulnerabilities are common methods for spreading ransomware. Ransomware detection research often employs static and dynamic analysis, but the most difficult obstacle is identifying zero-day attacks and the time it takes to do so. Dynamic analysis, although necessitating the execution of the attack, often produces more accurate findings than static analysis. This study attempts to enhance response-time ransomware detection and prevention before files are encrypted on the host system. A host-based method is used, including windowing techniques and feature extraction using n-grams and TF-IDF. The detection model was created using three machine learning algorithms: Naive Bayes, Random Forest, and Support Vector Machine. The criteria used in the evaluation were accuracy, precision, recall, F1 score, false positive rate (FPR), and false negative rate (FNR). The results show that Naive Bayes with 8 and 10 n-grams gets 100% accuracy, precision, recall, and F1-score, with 0% FPR and FNR, which is better than other models. Also, the Naive Bayes model with n-grams between 2 and 10 gives the best and most consistent evaluation matrix results after more than 20,000 API calls. The other models give the best results after more than 30,000 API calls. The suggested system can identify ransomware in the range of 25,000 to 35,000 API calls. This study also found phase variations between normal and ransomware activity, notably in the start and end stages. Although ransomware and regular activity have similar initialization periods, the ransomware phase differs significantly, favoring detection in the early stages. Thus, detection in the range of 25,000 API calls was helpful in preventing the misidentification of ransomware as regular activities. However, although using a large number of n-grams improves accuracy, it also increases data size and execution time. As a result, selecting n-grams that are neither too high nor too low is critical for maintaining a balanced performance. The framework also has numerous settings, including window size, detection threshold, and ten-window repetition. This response-time framework's use of n-grams has proven useful in identifying ransomware before it encrypts all data.

Keywords: Ransomware detection, response-time, windowing, host-based, zero-day attack, API call