10.48047/jocaaa.2024.32.01.26

CNN2D Algorithm for Detection of Ransomware Attacks Using Processor and Disk Usage Data

Mrs.B.Laxmi Kalpana[1], Mahi [2], P.Niharika [3], Nenavath supriya[4]

[1] Assistant Professor, Department of Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India
Email: kalpana.laxmi@gmail.com

[2,3,4,]B.Tech Student, Department of Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India

ABSTRACT:

Ransomware commonly encrypts files, disables antivirus software, and leaves the target machine and its contents useless. The methods used today to find this kind of ransomware include keeping an eye on the target system's files, system calls, and processes while also evaluating the information gathered. Tracking several processes has a significant overhead; ransomware that is more recent may disrupt the tracking and tamper with the data that is gathered. In this research, a reliable and useful method for identifying ransomware running in a virtual machine (VM) is presented. Using information gathered from the host computer for certain processor and disc I/O events, we build a detection model by applying an automatic machine learning (ML) classification to the whole VM. This method reduces the possibility of ransomware contaminating data and does away with the expense of constantly monitoring every operation on the target system. It also withstands changes in the workloads of users. It offers quick and highly probable identification of both known (used to train the ML model) as well as unidentified (not used to train) ransomware. We examined seven machine learning classifiers, and the random forest (RF) classification performed the best. The RF model identified malware at a 0.98 probability in 400 milliseconds across six distinct client loads and 22 instances of ransomware.

INTRODUCTION

Malware known as ransomware makes a target computer and its contents useless by locking or encrypting files on the computer. Cybercriminals extort victims' money using ransomware attacks. Ransomware attacks are a potential tool used by nation-state actors to damage their opponents' vital infrastructure. In order to force the victim to pony up a ransom or trade the data on the underground web, these assaults often include the exfiltration of the victims' data. Rogue ransomware assaults affected almost 70% of

10.48047/jocaaa.2024.32.01.26

enterprises in 2022. By 2031, it is anticipated that ransomware would target a company, user, or device every two seconds, up from every eleven seconds in 2021. $20 billion was spent on damage in 2021, and it's probably to surpass $265 billion, with Vicente Alarcon-Aquino serving as the assistant editor in charge of organising the manuscript's review and granting publication approval. through 2031. A number of investigators have lately looked at the identification of ransomware assaults. Using the hash values produced by antivirus software for well-known ransomware, signature-based detection looks for files on the target computer that match the hash values. However, such signature-based detection is not impervious to variants of recognised ransomware that are polymorphic or metamorphic. As a result, runtime or behavioural detection of ransomware enhances signature-based detection techniques. The behaviour of the virus—the series of activities the ransomware takes after infecting the target machine—is examined by the behavioural analysis, a dynamic study. Malware may take many different forms, but in order to swiftly encrypt as many documents as possible, ransomware has to follow a certain set of actions. A number of current ransomware programmes, including LockBit2.0, Darkside, and Backmatter, encrypt just the initial few bytes of files in order to swiftly make additional files unreadable. Ransomware's runtime behaviour will thus probably be distinguished from that that of a benign programme by the need of swiftly encrypting user data. The idea is that a system that is being attacked by ransomware has to show some kind of unchangeable abnormal behaviour. In order to encrypt data, for instance, the ransomware has to access documents from the hard drive. This causes increased activity, which may be detected by machine learning techniques that have been properly taught. When runtime detection is used to the target system, it requires constant observation of different processes, parts, and subsystems, as well as the gathering and examination of event-related data in order to spot unusual activity. By generating new processes and activities, ransomware may attempt to conceal its runtime behaviour. Nevertheless, the truth remains that, with the right analysis, an attack-affected system will show signs of increased activity. Because several processes need to be watched over and it may be difficult to pinpoint the ransomware-related activity, runtime identification is resource-intensive and invasive if monitoring is carried

10.48047/jocaaa.2024.32.01.26

out on the target system. Additionally, ransomware that is meant to stop running processes before encrypting data has a tendency to deactivate this kind of monitoring. Hardware performance counters, or HPCs for short, are special purpose registers that keep track of system and processor events for each process or for the whole system. The number of instructions completed, cache misses, and off-chip memory accesses are only a few of the hundreds of processing and system events that may be counted by the modern processors. Performance analysis and system software optimisation are common uses for the data gathered with HPCs. Nonetheless, its use in malware detection has been the subject of numerous recent research projects. HPC data gathered for every process operating on the system was used by Alam et al. But it's not possible to monitor every operation since doing so might seriously impair the system's functionality. The data was gathered by Pundir et al. using machines. However, the scope of their tests is restricted to a single Microsoft virtual machine (VM) workload, and the detection accuracy may be greatly affected by altering the VM workload (by changing the number of programmes).

RELATED WORK

Behaviour-based analysis and detection of malware

The Internet is seriously threatened by malware like Trojan horses, worms, and spyware. While malware and its variations might differ greatly from content signatures, we found that they share several higher-level behavioural traits that are more accurate in exposing the true aim of malware. In addition to presenting the formal Malware Behaviour Feature (MBF) extraction method and suggesting a malware detection algorithm based on harmful behaviour features, this study also examines the approach of malware behaviour extraction. Our MBF-based malware detection system has been devised and deployed, and the testing findings indicate that it is capable of detecting unknown malware that has just surfaced.

A hardware-assisted runtime method for detecting crypto-ransomware, called Ran Stop

Among the various malware programmes that are now in use, crypto-ransomware is particularly dangerous since it may financially extort victims by preventing them from accessing their documents via

10.48047/jocaaa.2024.32.01.26

unauthorised encryption, keeping their papers hostage, and threatening them with legal action. Globally, this causes losses up to millions of dollars per year. The quantity of ransomware variations is increasing due to their ability to elude several antivirus programmes and software-only detection methods for malware that depend on static behavioural signatures. Our proposal in this research is to use Rain Stop, a hardware-assisted method, to identify crypto-ransomware infections in commodity processors early on. Rain Stop observes micro-architectural event sets and identifies known and undiscovered crypto-ransomware variants by using data from hardware performance counters contained in the performance monitor unit of current CPUs. This article focuses on training a recurrent neuronal network-based machine learning architecture via an LSTM model to analyse micro-architectural events in the computer's hardware domain during the execution of both benign and numerous ransomware versions. By leveraging the data from connected HPCs, we generate time series to build intrinsic statistical characteristics that enhance Rain Stop's detection accuracy and lower noise via the use of global average pooling and LSTM modelling. By analysing HPC data gathered at 20 timestamps spaced 100us apart, Rain Stop may detect ransomware early on and correctly within 2ms of the program's execution beginning. A ransomware cannot do substantial harm at this point in time, if any, because to the early identification. Furthermore, Ran Stop's detection accuracy of ransomware is 97% on average for fifty random trials when validated against innocuous programmes that have behavioural (sub-routine-centric) similarities with a crypto-ransomware.

A real-time detection system against cryptographic ransomware, called Regard

Recent times have seen the (re)emergence of ransomware, a common virus that preys on both individual and business victims in an attempt to obtain financial advantage. A significant number of files are encrypted irreversibly as a result of the current ransomware detection mechanisms' inability to offer an early warning in real-time, and the post-encryption methods (such as key extraction and file restoration) have a number of drawbacks. Furthermore, a high false positive rate is caused by the detection mechanisms currently in use, which are unable to discern between the original

10.48047/jocaaa.2024.32.01.26

purpose of file modifications and ransomware encryption. In other words, they cannot distinguish between a ransomware-caused significant file alteration and a user-initiated file operation (such as benign encryption or the compression). In response to these difficulties, we present in this paper Regard, a ransomware detection mechanism that can identify crypto-ransomware in real-time on a user's computer by (1) using decoy techniques, (2) keeping a close eye on the file system and running processes for malicious activity, and (3) avoiding the flagging of benign file changes by learning users' encryption behaviour. Using samples from the 14 most common families of ransomware to date, we test our method. With an overhead of just \(\sim \)1.9%, our studies demonstrate the effectiveness of RWGuard in real-time ransomware detection, with zero false negative and minor false positive (\(\sim \)0.1%) rates.

Regarding whether performance counters may be used to identify malware online

In every domain, as computers proliferate, malware also proliferates inside that domain. Systems are infested with malware such as viruses, rootkits, spyware, adware, and other types, even on the newest mobile platforms. Anti-virus software does not completely eradicate malware threats; in fact, the number of techniques to get around anti-virus (AV) software is rising. In reality, modern attackers use vulnerabilities in antivirus software to compromise computers. In this research, we investigate the viability of using current performance counters to construct a malware detection in hardware. We discover that performance counter data may be used to detect malware, and our methods of detection are resistant to even the smallest changes in malicious programmes. Consequently, we are able to identify several variants within a malware family on the Iphone ARM and Intel Ubuntu platforms, even after analysing a limited number of those versions. Furthermore, the malware detector may operate safely underneath the system software thanks to our suggested hardware changes, paving the way for less complicated and glitch-free AV systems than those using software AV. State-of-the-art internet malware detection might be improved by combining the security and robustness of hardware antivirus approaches.

METHODOLOGY

10.48047/jocaaa.2024.32.01.26

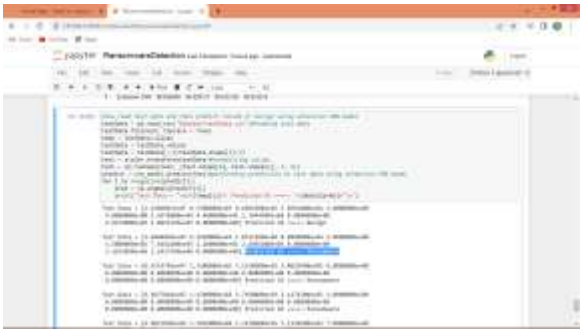
To implement this project, we have designed following modules

1. loading all packages and classes
2. defining class to normalize features
3. dataset pre-processing such as normalizing and shuffling and then displaying processed values.
4. training SVM algorithm on 80% training data and then performing prediction on 20% test data and then calculating accuracy and other metrics
5. training KNN algorithm
6. training decision tree algorithm
7. training Random Forest algorithm
8. training XGBOOST algorithm
9. training DNN algorithm
10. training LSTM algorithm
11. training CNN2D algorithm
12. displaying all algorithms performance in tabular format
13. prediction on test data

RESULT AND DISCUSSION

In above screen defining class to normalize features and then loading and displaying dataset values

In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms Extension CNN got high accuracy



10.48047/jocaaa.2024.32.01.26

In above screen reading test data and then using extension CNN algorithm object performing prediction on test data and then in output before arrow symbol =➔ we can see Test data values and after arrow symbol we can see predicted values as 'Ransomware or Benign'

CONCLUSION

This study describes a method to rapidly and correctly identify ransomware running on a virtual machine (VM) by gathering the VM's CPU and disc I/O activity events from the host computer and use machine learning methods to analyse the data. Recursive feature reduction with cross-validation is used to choose five events from a pool of over forty events to get processor-event data using the perf tool and the hardware performance counters (HPCs); virus Domb stats is used to gather disc I/O-event data for eight events. Two DL and five ML classifiers were taken into account. Three models were created for each classifier: one that just used HPC data, one that only used disc I/O data, and one that combined the two types of data into one integrated model. When it came to all seven classifiers, the integrated model fared the best. With respect to detection accuracy, the random forest, or RF, classifier outperforms the other seven classifiers and requires less training time. In general, the findings of the RF-integrated model in identifying both new and known ransomware—which was not used during training—are encouraging. Comparing our method to previous ones that track ransomware activity directly on the intended computer, we find two benefits. Because the monitoring is carried out from the host system, the target machine first experiences no overhead. Second, the data collecting process cannot be hampered or corrupted by ransomware that is operating on the target computer. This research makes a substantial contribution by evaluating the ML/DL-based models for detection on the target virtual machine (VM) under diverse user workloads. The majority of earlier research assess their models for a single workload scenario. We find that when a model is trained using collected data without any user activity if the background workload, it has minimal detection accuracy when the user engages in activities like Web browsing, playing audio/video clips, or using productivity applications like Microsoft Word, Excel, or Adobe tools. Virtual machines that are observed at their host or kvm level may use our detection approach. One benefit of this strategy is that the data

10.48047/jocaaa.2024.32.01.26

gathering procedure is constant irrespective of the OS that is used for the virtual machine. A further benefit is that the malware operating inside the virtual machine is oblivious to the monitoring so cannot impede the process of data collecting. Cloud service providers may provide their customers further defence against ransomware threats by using our detection techniques. The continued shift of computing to virtual machines (VMs) and the cloud makes this extra ransomware defence essential and relevant. We did not specifically look at ransomware's ability to steal data for this research. We want to gather network traffic from the target virtual machine (VM) on the host system and use HPC and I/O data analysis to look into the identification of exfiltration activities. We classify as harmful any and all data gathered during ransomware operations. Nevertheless, some ransomware silently investigates the target computer and the network for a while (tens of seconds or more) before to initiating their harmful actions. The implementation of this kind of ransomware results in noisy data; the malware stays dormant for a significant amount of the experiment's duration, and the information collected matches the system load by default but is classified as harmful. This kind of information might result in a less precise detection, since our goal is to identify malware that is actively changing files. Assume we can identify as malicious the data gathered during encryption operations and as scouting the data gathered during the scouting phase. Then, in our opinion, the detection accuracy will rise and serve as a more reliable signal of ransomware in its most devastating form. In our next work, we want to explore more precise data tagging. In order to strengthen the detection models' resistance to changing user behaviour, we plan to use more degrees of workloads in our future research. We evaluated the models we described in this work using data gathered from further rounds of tests. We want to employ the algorithms in the future for real-time ransomware detection while it is still active. Although the application of our concept is limited to virtual machines, we want to modify it for standalone computers in further work. We have not assessed the models' suitability for a different system configuration, such one with more memory or CPU cores, or one with less memory. In the future, we want to look into this.

REFERENCES

10.48047/jocaaa.2024.32.01.26

[1] SR Department. (2022). Ransomware victimization rate 2022. Accessed: Apr. 6, 2022. [Online]. Available: https://www.statista.com/statistics/204457/businesses-ransomware-attack-rate/

[2] D. Braue. (2022). Ransomware Damage Costs. Accessed: Sep. 16, 2022. [Online]. Available:

[3] Logix Consulting. (2020). What is Signature Based Malware Detection. Accessed: Apr. 3, 2023. [Online]. Available: https://www.logixconsulting.com/2020/12/15/what-is-signature-based-malware-detection/

[4] W. Liu, P. Ren, K. Liu, and H.-X. Duan, ‘‘Behaviour-based malware analysis and detection,’’ in Proc. 1st Int. Workshop Complex. Data Mining, Sep. 2011, pp. 39–42.

[5] (2021). Polymorphic Malware. Accessed: Apr. 3, 2023. [Online]. Available:

[6] M. Loman. (2021). Lock file Ransomware’s Box of Tricks: Intermittent Encryption and Evasion. Accessed: Nov. 16, 2021. [Online]. Available:

[7] N. Pundir, M. Tehrani poor, and F. Rahman, ‘‘Ran Stop: A hardware assisted runtime crypto-ransomware detection technique,’’ 2020, arXiv:2011.12248.

[8] S. Mehnaz, A. Budgerigar, and E. Bertino, ‘‘Regard: A real-time detection system against cryptographic ransomware,’’ in Proc. Int. Symp. Res. Attacks, Intrusions, Defences. Cham, Switzerland: Springer, 2018, pp. 114–136.

[9] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Seth Madhavan, and S. Stolfo, ‘‘On the feasibility of online malware detection with performance counters,’’ ACM SIGARCH Compute. Archit. News, vol. 41, no. 3, pp. 559–570, Jun. 2013.

[10] A. Tang, S. Seth Madhavan, and S. J. Stolfo, ‘‘Unsupervised anomaly-based malware detection using hardware features,’’ in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2014, pp. 109–129.

[11] S. Das, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, ‘‘SoK: The challenges, pitfalls, and perils of using hardware performance counters for

10.48047/jocaaa.2024.32.01.26

security,'' in Proc. IEEE Symp. Secure. Privacy (SP), May 2019, pp. 20–38.

[12] S. P. Kadiyala, P. Jadhav, S.-K. Lam, and T. Srikanthan, ''Hardware performance counter-based fine-grained malware detection,'' ACM Trans. Embedded Compute. Syst., vol. 19, no. 5, pp. 1–17, Sep. 2020.

[13] B. Zhou, A. Gupta, R. Jahanshahi, M. Egale, and A. Joshi, ''Hardware performance counters can detect malware: Myth or fact?'' in Proc. Asia Conf. Compute. Common. Secure., May 2018, pp. 457–468.

[14] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, ''On the classification of Microsoft-windows ransomware using hardware profile,'' Peer Compute. Sci., vol. 7, p. e361, Feb. 2021.

[15] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, and A. Chattopadhyay, ''RATAFIA: Ransomware analysis using time and frequency informed autoencoders,'' in Proc. IEEE Int. Symp. Hard. Oriented Secure. Trust (HOST), May 2019, pp. 218–227.

[16] K. Thumbpad, R. Boppana, and P. Lama, ''HPC 41 events 5 rounds,'' Harvard Dataverse, 2022, doi: 10.7910/DVN/MA5UPP.

[17] K. Thumbpad, R. Boppana, and P. Lama, ''IO 41 events 5 rounds,'' Harvard Dataverse, 2022, Doi: 10.7910/DVN/GHJFUT.

[18] K. Thumbpad, R. Boppana, and P. Lama, ''HPC 5 events 7 rounds,'' Harvard Dataverse, 2022, Doi: 10.7910/DVN/YAYW0J.

[19] K. Thumbpad, R. Boppana, and P. Lama, ''Io 5 events 7 rounds,'' Harvard Dataverse, 2022, Doi: 10.7910/DVN/R9FYPL.

[20] K. Thumbpad, R. Boppana, and P. Lama, ''Scripts to reproduce results,'' Harvard Dataverse, 2023, Doi: 10.7910/DVN/HSX6CS.

[21] M. Rhode, P. Burnap, and A. Wedgbury, ''Real-time malware process detection and automated process killing,'' Secure. Common. Newt., vol. 2021, pp. 1–23, Dec. 2021.

[22] A. Kharrazi and E. Karda, ''Redemption: Real-time protection against ransomware at end-hosts,'' in Proc. Int.

10.48047/jocaaa.2024.32.01.26

Symp. Res. Attacks, Intrusions, Defences. Cham, Switzerland: Springer, 2017, pp. 98–119.

[23] F. Mabolo, J.-M. Robert, and A. Salishan, ‘‘An efficient approach to detect torrent locker ransomware in computer systems,’’ in Proc. Int. Conf. Cryptal. Newt. Secure. Springer, 2016, pp. 532–541.

[24] K. Lee, S. Lee, and K. Yim, ‘‘Machine learning based file entropy analysis for ransomware detection in backup systems,’’ IEEE Access, vol. 7, pp. 110205–110215, 2019.

[25] C. J. Chew and V. Kumar, ‘‘Behaviour based ransomware detection,’’ in Proc. Int. Conf. Compute. Their Appl., in Epic Series in Computing, vol. 58. 2019, pp. 127–136