Malware Detection in Applications using a Virtual Environment

***Jagriti Kumari*[1]*, Sowmya B*[2*]**
[1]*Student,* [2]*Assistant Professor*
Department of Computer Science and Engineering,
East West College of Engineering, Bangalore, Karnataka, India.

****Corresponding Author***
***E-mail Id:-** sowmya.ab26@gmail.com*

ABSTRACT

Malware assaults amongst diverse cyber-attack on computers are deemed harmful, as they are passive and sleathy. A malware assault is a cyber-attack that initiates the action of the perpetrator on the system of the victim. Adware, spyware, keyloggers and any other malware may be used to carry out malware attacks. Spyware captures information from companies or individuals and distributes it to harmful users. The Spyware keylogger records, logs and transmits the user's keystrokes to the virus attacker. These threats must be recognised and identified to ensure adequate data protection. Early detection helps to slow the spread of malware. This paper provides a methodology for logging and testing spyware attacks.

Keywords:-*Keylogger, spyware, virtual machine.*

INTRODUCTION

Cyber-attack refers to any operation leading to the misuse of system resources or data. Malicious software is characterised as software utilised by the self-extruding script to launch cyber assault. These programmes collect personal data without user awareness. These data can be modified to attack the attacker. The attacker produces malware to change the system's functioning. Passive recording is used for collecting information such as credit or debit passwords, personal data, bank numbers and pin codes.

After the target system is infected, the virus is harmful, as user information is created, pushed keys logged, etc. Spy on and gather sensitive system data with info-stealer programmes. These data can be utilised to carry out an attack. The attacker obtains sensitive information across the web, including url, user information. The file names that the user downloads can likewise be broadcast.

Certain info stealers extract log data and report apps installed. It permanently monitors user conduct, login accounts and customs browsing. These data may be exploited to make sluggish logging and database theft of the programmes and browser settings. Current research focuses on detecting malware attacks such as keyloggers and data theft. The spywares are often combined and dissimulated into helpful adware programmes. Adware apps include plug-ins that improves current applications' operations and capabilities.

Adware is highly useful, but is likely to contain external code which the deployer has not verified sufficiently. Adware redirects the search results of a user to a comparable web application. The keylogger is a programme or script attached to a software. The keylogger is two types; it is hardware and software for keyloggers. The hardware keylogger is active when the machine is powered on. The hardware keyloggers are of various kinds, overlays, keyboard commands, etc. Only while the programme is running is the keypad software enabled. When the user executes the software, the keylogger code is activated.

LITERATURE SURVEY

The honey pot concept is used for the identification and prevention of spyware in paper [1]. Any process entering the system will also be entered on the honeypot server. This e-mail is logged by the Sweet Pot Server. When mail is sent for a permanent duration to a certain mail id, it informs you to initiate the termination signal of your host system.

The author employed mining algorithms for spyware detection in paper [2]. In this case, the recognised malware model known as N-grams was classified by five unique supervised learning techniques. Spyware covers the established pattern of keyloggers and info-stealers. The n-grams of recognised applications and spyware is extracted.

The author describes the different design patterns for anti-security applications to identify malware in paper [3]. The author uses categorization systems to classify the input file.

The classification devices detect the malware from the system's previously established classes of spyware. If new varieties of spyware emerge, they become part of a new family.

The author explains many keylogger detection approaches in paper [4]. The author proposes the user to sequence strings between successive password keys for secure access to password- protected account methods. It also mentioned strategies like bot identification, dendritic cell algorithms.

The author describes in paper [5] the design pattern of the keylogger, its use, implementation, and integration in an application. The author also stresses the evident patterns in the design, that the keylogger types may be detected and differentiated. On the basis of these observations, the keylogger will generate a confident file of all user actions. It is concluded. The malicious user may share this file. The author explains in paper [6] the frame of password securing in spyware keylogger. It employs encryption to prevent spyware for keyloggers. The keylogger logs the whole key passed by the user. It was seen. The author proposes to use the random encrypted keyboard to enter data so that an attacker does not utilise recorded data directly, but to utilise anti-encryption algorithms to remove the correct data. This study emphasises that encrypted data sharing can improve the security of data.

In paper [7], the author describes how kernel-level procedures are deeply and transparently hooking to investigate Spyware's dynamics. The kernel level system processes are known as the whole system execution, monitored by the CPU and connected to its kernel level system process each action that is legitimate or malicious. Decision tree methods, such J48, linear regression, JRIP, are used to classify malware classifications.

METHODOLOGY

A log and test approach incorporating the above conclusions is established. Once adware or programme is installed, data compromise is unavoidable. In order to eliminate this constraint, it is recommended to first install the programme on a clean server. Log into, check and test the existing log in many log files such as system log, nmaps log, IP table log, all application installations and installation activities based on common vulnerabilities. The virtual machine configuration can accomplish this. The application will be tested through the virtual machine setup if an application is to be installed and report whether it can be installed on the user system based on the outcome. The virtual machine is a guest operating system that works on the user's operating system without impacting the user system.

The virtual machine can perform work, such as running a programme, managing additional operating systems etc. Virtual machines have the advantage that the device may always be removed and it does not contain information from the user.

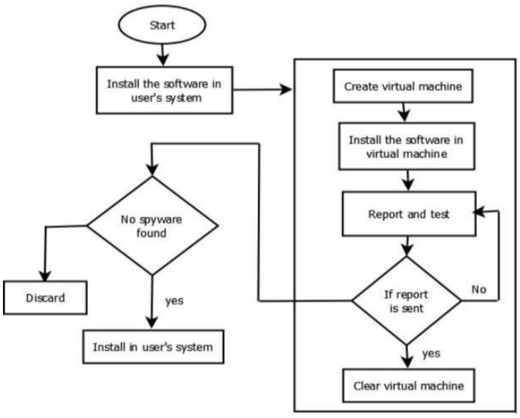


Fig.1:*-Flow chart*

The Figure 1 illustrates the work flow. Whenever a user attempts to install the software, a pop-up asks if the software needs to be tested prior to the user's system installation. The installation url is routed to the virtual machine if the user tests and install. The configuration file is called to know about the user system setup for the construction of the virtual machine. The virtual machine is generated when the configuration has been made. The programme has a virtual machine installed. The presence of the keylogger and the info -stealer malware is tested. Based on the test results, a report is created. The report is subsequently forwarded to the system of the user. The user determines on the basis of the report if the programme may be installed on a host computer. Destroy the virtual computer after you have sent the report to the user.

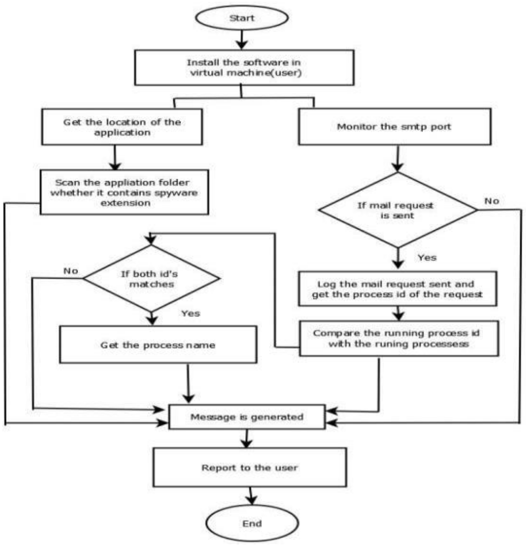
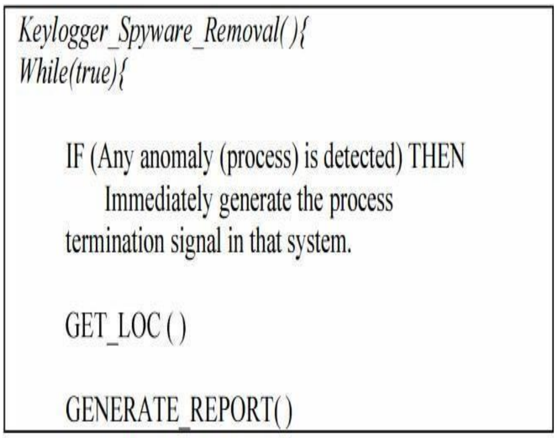

Start
Install the software in virtual machine(user)
Get the location of the application
Monitor the smtp port
Scan the appliation folder whether it contains spyware extension
If mail request is sent
No
Yes
No
If both id's matches
Yes
Log the mail request sent and get the process id of the request
Get the process name
Compare the running process id with the runing processess
Message is generated
Report to the user
End

Fig.2:*-Represents the flow of test and report*

The list of all packages installed in the application has been received for the test and report process Figure 2. Packages and programme files are checked to see if spyware extensions are included. A message is generated if it contains any malware extensions. The message generated is communicated to the user. Then the smtp port is monitored to determine if the spyware includes active keyloggers. The mail is logged if any e-mail is sent over the smtp port. We determine which process sent the mail from the mail process id.

PROPOSED ALGORITHM

The Figure 3 shows how malware is validated. The spyware is installed first and is received in the proposed algorithm.

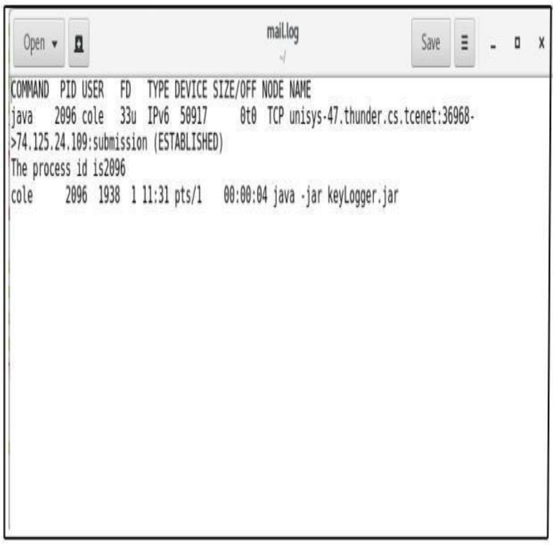


```
Keylogger_Spyware_Removal( ){
While(true){

    IF (Any anomaly (process) is detected) THEN
        Immediately generate the process
    termination signal in that system.

    GET_LOC ( )

    GENERATE_REPORT( )
```

Fig.3:*-Test and Report Process*

RESULTS AND DISCUSSION

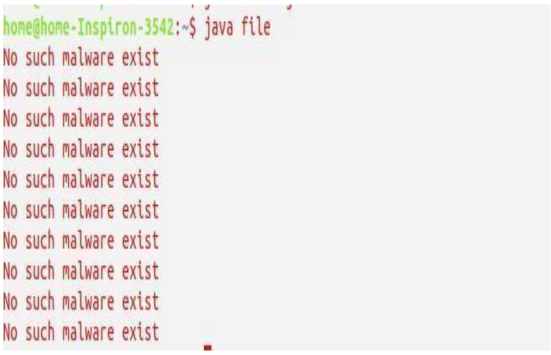
This analysis uses an application with spyware of the keylogger. When the application is running, the virus begins monitoring key presses. You log the keys and email the letters to the attacker with the user screen shot.



```
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
java     2096 cole   33u  IPv6  50917      0t0  TCP unisys-47.thunder.cs.tcenet:36968-
>74.125.24.109:submission (ESTABLISHED)
The process id is2096
cole      2096  1938  1 11:31 pts/1    00:00:04 java -jar keyLogger.jar
```

Fig.4:*-Result*

Figure 4 shows the mail process id provided to the attacker. Process id. The process id is used to identify the process by which mail is sent.



```
home@home-Inspiron-3542:~$ java file
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
No such malware exist
```

***Fig.5:**-Spyware extension analysis*

If the email request is not sent immediately, then the malware will remain silent state. In two days or even in weak days the malware can be launched. If the mail is not sent quickly, the application location will be deployed. The application analyses the files and folders in Figure 5 and determines whether spyware extensions are there. And also read and verify whether the contents of the files contain spyware notation such as @symbol, sendto, mailed etc. Thus, we may disclose that it contains spyware to the user when we analyse the software.

CONCLUSION

The malware is a major global threat. The current study effort proposes a framework for detecting the active keylogger. The active keylogger is a spyware that observes the activity of the user and actively provides the data the attacker receives. It thus has a threat to privacy. Therefore, the technique is proposed for early detection of spyware. The study suggests currently semi-automation of the spyware detection and in future complete orchestrations. This enables us to reduce the danger of spyware and data theft.

REFERENCES

1. Wazid, M., Katal, A., Goudar, R. H., Singh, D. P., Tyagi, A., Sharma, R., & Bhakuni, P. (2013, January). A framework for detection and prevention of novel keylogger spyware attacks. In *2013 7th International Conference on Intelligent Systems and Control (ISCO)* (pp. 433-438). IEEE.
2. Shahzad, R. K., Haider, S. I., & Lavesson, N. (2010, February). Detection of spyware by mining executable files. In *2010 International Conference on Availability, Reliability and Security* (pp. 295-302). IEEE.
3. Sheta, M. A., Zaki, M., & El Hadad, K. A. E. S. (2016, July). Anti-spyware security design patterns. In *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)* (pp. 465-470). IEEE.
4. Solairaj, A., Prabanand, S. C., Mathalairaj, J., Prathap, C., & Vignesh, L. S. (2016, January). Keyloggers software detection techniques. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)* (pp. 1-6). IEEE
5. Wood, C., & Raj, R. (2010, July). Keyloggers in Cybersecurity Education. In *Security and Management* (pp. 293-299).
6. Tyagi, G., Ahmad, K., & Doja, M. N. (2014, February). A novel framework for password securing system from key-logger spyware. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)* (pp. 70-74). IEEE.
7. Javaheri, D., Hosseinzadeh, M., &

Rahmani, A. M. (2018). Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access*, *6*, 78321-78332.

8. Mallikarjunan, K. N., Preethi, S. R., Selvalakshmi, S., & Nithish, N. (2019, April). Detection of spyware in software using virtual environment. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1138-1142). IEEE.
9. Menon, K. D., Raj Jain, A., & Kumar Pareek, D. (2019). Quantitative analysis of student data mining.
10. Pai H, A., HS, S., Soman, S., Pareek, D., & Kumar, P. (2019). Analysis of causes and effects of longer lead time in software process using FMEA SSRN: https://ssrn.com/abstract=3508574 or http://dx.doi.org/10.2139/ssrn.3508574
11. Pai H, Aditya and H S, Sameena and Soman, Sandhya and Pareek, Dr. Piyush Kumar, ROC Structure Analysis of Lean Software Development in SME's Using Mathematical CHAID Model (May 17, 2019).
12. HS, S., Soman, S., & Kumar Pareek, D. (2019). Fast and efficient parallel alignment model for aligning both long and short sentences
13. BR, M., Bhavya, B. R., Pareek, D., & Kumar, P. (2016). Education Data Mining: Perspectives of Engineering Students. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN*, 2347-5552
14. Kotagi, M., & Pareek, P. K. (2016). Survey on Challenges in DevOps. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN*, 2347-5552.
15. Soman, S., & Pareek, P. K. (2020). An exploratory analysis on challenges prevailing in small and medium IT firms. In *Journal of Physics: Conference Series* (Vol. 1427, No. 1, p. 012010). IOP Publishing.
16. Sangeetha, V., Vaneeta, M., Kumar, S. S., Pareek, P. K., & Dixit, S. (2021). Efficient Intrusion detection of malicious node using Bayesian Hybrid Detection in MANET. In *IOP Conference Series: Materials Science and Engineering* .1022, No. 1, p. 012077). IOP
17. Pustokhin, DA, Kumar Pareek, P, Gupta, D, Khanna, A, Shankar, K. Energy-efficient cluster-based unmanned aerial vehicle networks with deep learning-based scene classification model. *Int J Commun, Syst. 2021; 34:e4786.*
18. Swathi, K., & Shetteppanavar, P. (2017, May). An efficient machine translation model for Dravidian language. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* . 2101-2105. IEEE.
19. Aditya Pai, H., Pareek, P. K., Narasimha Murthy, M. S., Dixit, S., & Karamadi, S. (2021). An Exploratory Study for Process Optimization in IT Industry. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, 3* . 617-631. Springer Singapore
20. Soman, S., Pareek, P. K., Dixit, S., Chethana, R. M., & Kotagi, V. (2021). Exploration Study to Study the Relationships Between Variables of Secure Development Lifecycle (SDL). In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 3* (pp. 641-649). Springer Singapore
21. Suhas, G. K., Devananda, S. N., Jagadeesh, R., Pareek, P. K., & Dixit, S. (2021). Recommendation-Based Interactivity Through Cross Platform

Using Big Data. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, 3* (pp. 651-659). Springer Singapore

22. Soman, S., Pareek, P. K., Dixit, S., Kotagi, V.. (2020). An Empirical Investigation on Practicing Secure Software Development in Software Development Life Cycle in Small & Medium Level Software Firms in Bengaluru. *International Journal of Advanced Science and Technology, 29*(7s), 5164
23. Patil, S. S., Pareek, P.,K., Dinesh, H. A., Arlimatti, S.(2017). Review of relay selection techniques in multi-hop wireless sensor network with iot. *International Journal of Creative Research Thoughts (IJCRT), 5*(4).846-850