PROCESS MONITORING SYSTEM BASED ON KNOWLEDGE OF RESOURCE UTILIZATION LIMITS

Richa Devon[1], Anju Mishra[2]

[1,2] *Department of Information Technology, Amity University, Noida, (India)*

ABSTRACT

Software security, in recent years, has become a major issue considering the huge number of unsolicited attempts to procure data from the usage of software. Various malwares attempt to dig data from end-user application software, like those used for text editing or those used as the most recent web-based applications that are connected to the internet for every action. Such threats are not only harmful to the confidentiality of data but also to the security of the system being used. The methodology proposed in this paper uses time synchronization for judging the resource utilization by software once it starts. The added resource overhead, if any, is detected and checked for unauthorized activity.

Keywords: Security Metrics, Software Security, Threat Detection, Time Synchronization

I. INTRODUCTION

Software security assurance protects the information by designing and implementing software that can be relied upon consistently [1]. Software instructs a machine's processor to execute certain set of instructions. These instructions may be legitimate or illegitimate. Instructions sent without any information to the user and by a third party for destruction of data or access of it comes under illegitimate instruction. There are various kinds of software. The three broad categories include application software, system software and computer programming tools. Application software include web based software that are more prone to security attacks that are cyber based and need to be safeguarded from such threats [2].

Some software also employs macros that can be added for additional functionalities. But at times, this provision may be exploited for running codes for acquiring details from documents. Not only can these macros send critical data to a third party but may also act as a modifying agent needed while infecting with viruses.

The hookup of such additional codes may or may not hamper the efficiency of the application in itself. But, it surely eats up on the resources of the system besides putting the user's data at crucial risk. Application software like browsers are at the ultimate risk of being infected. They are attacked and logs associated with them maybe fished for any useful data. Dumpster diving is frequent in such cases. On and off we hear of some big company suffering from a security breach. One such example is that of Adobe in 2013 when all its user details were compromised. The damage was contained but the way to the user details was through such loopholes that need to be addressed as firmly as possible.

Also, software that are not used frequently or seldom used may be tracked using the registries and exploited. These may be used for running scheduled tasks that send log files or filtered data to a remote machine. Users need to be aware of such issues and keep removing software that are not of use any longer because they may play a breeding ground for such inconspicuous codes.

II.THE PROPOSED METHODOLOGY

The proposed idea is that a monitoring system needs to be developed that calculates which processes are crossing their resource allocation limits and an alarm is raised to inform the administrator or user about the particular process or the software implementing it [4].

Every process has a start time and an end time and process scheduling handles these. However, this time along with the amount of work it has done, is not taken into consideration to check if the process that was assumed to be doing a task has done the expected task only or has there been any additions, or removals for that matter. The number of bytes processed is taken proportionally to calculate how much the threat has eaten up on the time and memory.

2.1 The Algorithm for Threat Detection

The process may be divided in to the following broad steps:

1.1 Creation of a monitoring system to record resource requirement and compare them with the utilized resources by a process dynamically.

1.2 Creating alerts for affected software

1.3 Capturing image of an environment at a regular interval

1.4 Log maintenance and error log analysis

1.5 Checking for startup or scheduled tasks associated

1.6 Calculation of proportional resource requirement in case of heavy workload

1.7 Deletion of images or time stamps from logs to lighten the memory space, in case of no error in a defined period of time.

2.2 Parameters for the Monitoring Software

The system that is being monitored whether at a smaller level or a larger level needs to have the details of the software it is using [3]. The software vendors in general provide only the requirements posed by them on the information broadcast. This requirement generally includes the processor, operating system supported, RAM required, graphics card requirements, etc. They never state what the speed of the software would be with particular software.

There are various versions for different operating system or processor, saying 32-bit or 64-bit. But hardly a table is maintained of the resources including memory it would require of that requirement. If a software states that it requires only 64MB RAM, it should also have a measurement means to state how much of that RAM would be utilized to execute a sample process of a given size. This will only facilitate the calculation of estimated time.

We generally see estimated time left in a process to complete when we are downloading a file. At this juncture it depends on the internet connection speed a user is having. At other times, it shows the estimated time for transferring a file from a folder to another whether it is on the same system, on a USB or on another system.

Such calculations are done and are possible but to use them for identifying malware and similar threats is a possible means to remove insecurities from our system.

The monitoring software need not have a heavy database loaded to it because it is not maintaining huge records of threats that have harmed systems at other places on the planet [8]. The system has its own comparison with variables provided by the respective software vendors. Hence, the database of such variables will be much smaller compared to the history of threats. Also, the size of the database in this monitoring system will be dependent on the number and size of software that a user is choosing.

We cannot rule out the existence of anti-viruses but having such a system will safeguard our data from unknown threats even when the database of the anti-virus we are using has not been updated.

2.3 Alerts for Affected Software

The monitoring system needs to create alerts when a software initiates more processes than it had stated initially. If these processes cross the upper bound of memory requirement or take more time than they are supposed to be, then a different kind of alert needs to be raised. Though time is not provided by the vendors, it can be calculated if more information about process requirement is provided with the software by the vendors.

The types of alerts can be as follows:

2.4 Software Initiates More Number of Processes than it Stated as the Maximum in its Vendor Brochure

2..4.1 A process takes up more time than calculated in the given conditions of the working environment including the operating system, processor speed, RAM, etc. This calculation takes into account the values provided by the vendor as proposed in this study to enhance security. To counter the case of software bugs or vendors' miscalculation, the system needs to foresee the repercussions of the event and list them out for the administrator to approve or disapprove a similar threat automatically in future.

2.4.2 An alarm also needs to be raised when the memory consumed by the process is more than the maximum stated for that particular process.

These alarms can be used to warn the user or administrator of any processes that may be working in the background to either change the contents of the user's system or to send the data filtered and found useful to some third party.

Once an alarm is raised, the user can get to see which process was causing the alarm. In extreme cases of urgency, the process can be halted right away. In another scenario, the administrator maybe asked for permissions. However, the automated system is better for security as human may not identify the threat and let the process proceed with its fallacies [5].

2.5 Image of System

The system maintains recovery tools to roll back to the last stable state. This feature maybe exploited to maintain image of the system whenever software starts afresh. The new processes before starting may render the monitoring system an image of the system variables just before the start.

As soon as an alarm is raised, the administrator maybe informed or in the better case of automated response mechanism, the alarming processes maybe stopped and the system may be restored to the last image. This mechanism helps in successful and smooth running of the system instead of a crash.

If the system waits for the user to instruct them for the next action, there may be chances that some information maybe leaked. To avoid such scenarios, an automated response is preferred over the user intervention option.

Once the image is saved, it stays there for as much time as it takes to form two new set of images. Once the second new set of images is formed, another process may delete the previous set of images. An alternative approach maybe to simultaneously update the variables. These variables are system checkpoints to store registry entries of the system at any time.

An image once saved can also be proposed to induce optimum utilization of memory. As the advancements of intelligent systems are encouraged, the future systems are expected to manage their working not only on the scheduling algorithms of this decade but also utilize intelligent learning mechanisms to learn which processes should be implemented and when [6] [7].

2.6 Log Maintenance and Error Log Analysis

The system maintains logs of all the processes that take place in its environment. All these logs are impossible to be manually checked for errors or malicious activities. Hence, it is always beneficial to concentrate our attention on the error logs. These logs can have the minimal format of timestamp and error number with process id. The overhead because of it can be reduced by lowering the scan rate of processes that have been running for a considerable amount of time.

The error logs enlist various kinds of errors in them. Some are genuine and legitimate. Others are due to the malicious processes attached to software or hidden in system files orsome hookup with the aid of rootkits. All such tactics used for keeping the malware and other threats safe from being detected can be failed by maintaining the error log using our monitoring system. This monitoring system notes down the software that initiated the malign process by backtracking its source and also noted down the extra time it took which is calculated. The resources are also judged.

The administrator's duties include taking actions in case of any kind of threat to remove the source of the unwanted activity whichever process it might be. The software vendor has its own duties to provide with the updated requirements of the software in case there are updating and patches.

The operating system that is most currently in use widely relies heavily on internet connection for its apps. In future, not only the apps and packages like Microsoft Office but others also would be solely dependent on the internet with time. To deal with such issues in future, we need to deal with them in the same fashion. The database of anti-viruses would not hold of much use because they update their database after the appearance of a threat elsewhere. They do not look for loopholes. Hence, a monitoring system is very much a need.

Time synchronization based error detection and removal can be done easily and is by far preferred over the conventional ways and means. The error logs not only help in maintaining the system but also help in identifying the root cause. Most of the times, the repercussions of a malware are handled every time it pops up an unwanted problem but does not deal with the root of the pop-up windows. That stays unknown in conventional systems.

2.7 Startup and Scheduled Tasks Associated

The system that any user uses probably has a provision of scheduled tasks and startup tasks. Most of the times, some newly downloaded software will force some startup processes without being required by the user. To overcome this issue with the software, the logs also enlist the overhead time used by particular software without being initiated by the user.

Some startup tasks are initiated by files stored in system files folder. Such folders are generally ruled out from suspicion while checking for errors. Wrong deletion of files from these may immensely harm the functioning of the system and may even lead to crash of the complete system with total loss of important data. The proposed monitoring system does not differentiate between processes on any basis. It only compares the requirements with the dynamic values of resource allocation units and time consumption.

The vender can regularly update their definitions for the user. Any hacks into the previous software or interference by a third party or twisted version by a third party may be detected because of the inbuilt directive to update from a specified website only. This website is visible to the user every time an action is taken so that they know if their files have been tampered.

2.8 Calculation of Resource Requirement

The various requirements of any process can be encompassed in memory, RAM, time consumed, internet services requirement and so on [9]. Once the vendor has provided details of the requirements of the software, or even in the ulterior states, the operating system comes with the details of requirement of its individual processes. In some cases, we may assume that the vendor will provide updated inputs on the enhanced requirements.

The original resource requirements stated be the software vendor need to be considered to compute the required resources at any later stage. The unit for requirement can be measured in terms of time taken for a certain amount of memory. The direct proportion assumption can be utilized till a fixed threshold value of space or time is reached. After that, a downward graph of requirement fulfillment occurs.

The calculation is done of memory required, RAM space required and time that will be consumed for a process. Complex algorithms need to determine how this time is affected by concurrent processes running in parallel.

The calculation is further utilized in detecting any abnormal behavior or anomalies in the working of the system. The abnormalities are logged into the log files and errors are entered in the error logs.

2.9 Deletion

The deletion procedure is triggered when the second set ofprocess status are ready and updated. The entities to be deleted include system images, logs of errors and further on, infected software is banned from further activity until review. The infected software may be removed or the vendor may be contacted to review the state of the error.

2.10 Process of Time Synchronization

The process deals with an algorithm of steps such that starting time of a process is noted and the dynamically process duration is incremented with the clock timer. Once it reaches the threshold value stated by the software vendor for resource utilization, an alarm is raised.

Simultaneously, a log is maintained of all the processes running and errors if any. This log further helps in dealing with errors. Recovery is done by going to the last successful system image in case of threats have

succeeded partially. Also, the infected software is banned from further action until it is reviewed. Review can be done by the administrator or by the software vendor via its automated review services.

III. CONCLUSION

The future prospects of time scheduled cyber threat detection are bright as the whole paradigm of our regular systems is going to shift to online internet based work. The operating system and drives are all going to be shifted to remote servers and only limited working is to be done on a system. Cloud based storage will also enhance efficiency. The association of the proposed scheme with the host intrusion detection system and open source intrusion detection system will not only implore that there will be only negligible amount of intrusion but also an optimized state of work environment.

REFERENCES

[1] Song Huang, Zhan Wei Hui, "A Case Study of Software Security Test Based On Defects Threat Tree Modeling", International Conference on Multimedia Information Networking and Security, 2010

[2] Liu Li, Wang Chunlei, "A Method for Modeling and Analyzing the SecurityAttributes of Service-Oriented Software System", International Conference on Computer Science and Network Technology, 2011

[3] Aleem Khalid Alvi, Mohammad Zulkernine, "A Natural Classification Scheme for Software Security Patterns", Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2011

[4] Sen-Tarng Lai, "An Analyzer-based Software Security Measurement Model for Enhancing Software System Security", Second WRI World Congress on Software Engineering, 2010

[5] Mohamed Almorsy, John Grundy, "Automated Software Architecture Security Risk Analysis using Formalized Signatures", ICSE , San Francisco, CA, USA, 2013

[6] Atsuo HAZEYAMA, Hiroto SHIMIZU, "Development of a Software Security Learning Environment", 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2012

[7] Shareeful Islam, Paolo Falcarin, "Measuring Security Requirements for Software Security", Proceedings of the 10th IEEE International Conference On Cybernetic Intelligent Systems, September 1-2, London, UK, 2011

[8] Francisco, Arnaldo, "Security Engineering Approach to Support Software Security", IEEE 6th World Congress on Services, 2010.

[9] Athena Abdi, AfshinSouzani, "Using Security Metrics in Software Quality Assurance Process", 6'th International Symposium on Telecommunications (IST'2012), 2012.