International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Review paper on Resource Exhaustion Logging in IDPS

C M MADAN KUMAR[1], PRAVEEN S[2]

[1]UG student,
[2]Assistant Professor
[1,2]Dept. of Electronics and Communication Engineering, R V College of Engineering, Bengaluru, Karnataka, India

---

Abstract - *Resource exhaustion is one of the major issue in IDPS systems as they are the security solutions for the small, medium and large organizations which will be working with their confidential information as well as personal information of the employees. As it is known exhaustion can't be completely removed and their will be numerous reasons for the ex-haustion, so there is a requirement for the intelligence that minimizes the troubleshooting time taken to resolve the issue. Main objective of the project is to develop an intelligence that will decrease the trou-bleshooting time in the case resource exhaustion happened due to high traffic or attack in IDPS systems. To decrease the troubleshooting time is it required to have some relevant information at the time of exhaustion which will help the debugger to understand the reason for the exhaustion and providing a solution by taking minimum time. As the operating system running on the IDPS systems is customized Linux, it is obvious to use the many features provided by the operating system such as logrotate. Shared memory is used for the interprocess communication for the implementation.*

Key Words: IDPS, Resource Exhaustion, Shared Memory, syslog server, Lockless Queues, Logging

1. INTRODUCTION

Today, there are numerous methods for securing data like using hardware(Hard drives), software(Encryption), combining both hardware and software and means which are au- thoritative. Every one of them are focused on security improving of data. Such methods for solving the security problems are are IDS (Interruption Detection System) and IPS (Intrusion Prevention System) these systems are designed in such a way that they can be configured as IDS/IPS at a given point of time. Intrusion prevention is another methodology to resist distributed networking systems, which join the procedure firewall with the Intrusion detection appropriately, which is proactive strategy. Keep the malwares from entering the system by looking at different information record and avoidance attitude of pattern recognition sensor. At the point when an attack is identified, intrusion prevention square and log the culpable information. The essential IPS utilizes signatures to distinguish action in organize traffic and host perform detection on incoming - outgoing packets and it will obstruct that before the harm and access confidential information.

1.1 Shared Memory

Communication between processes through shared memory is a concept where two or more processes can access the common memory. And communication is achieved through this shared memory where changes made by one process can be interpreted by another process. To summarize, each process has its own address space, if any process wants to communicate to other processes with any information from its own address space, then Inter Process Communication (IPC) techniques are the only possible ways. Communication between related or unrelated processes can take place.

1.2 Lockless Queues

Current Lock-based solutions include Mutex, Semaphores and Monitors. Mutex is also a semaphore with a maximum count of 1. Mutex can be considered as a Semaphore with a maximum count of 1.Only N threads/processes can reach the crucial region at once with a Semaphore with total count with N. One of the drawbacks to using Semaphores is deadlocks. Because of programming errors a deadlock can occur and is not easy to find and fix. Many algorithms for Deadlock prevention and Deadlock detection techniques are being developed.

2. Literature work

[1] Provides Intrusion detection systems evaluate information coming through the net- work and determine if traffic is potentially harmful based on existing evidence on any previously committed network attacks. Like many antiviruses, IDS performs signature analysis on the basis of which a decision is made to send a signal to the administrator. However, all safety measures based on an analysis of signatures are subject to compromise.

[2] Security mechanisms used to identify and avoid security threats to information systems and data networks are intrusion detection and prevention systems (IDPS). Such systems are designed to automatically identify and respond to security threats by reducing the risk to computers and networks monitored there. In this paper different methodologies such as signature-based, anomaly-based, stateful protocol analysis are used in detection and prevention systems for intrusion is discussed briefly

[3] Intrusion Prevention System (IPS) have been widely applied to protect from suspicious attack. Unlike traditional

system for detecting intrusion, IPS has additional features to secure network computer system. The additional features recognizing and acknowledging suspicious threat cause warning, event notification, by responsible response. IPS has problems, however, which affect the overall system used.

[4] Provides detailed knowledge about IDPS architecture and how IDPS components can be linked to each other via standard networks or a special network configured for security software management known as a management network. By using a management network, each sensor or agent host has an additional network interface known as a management interface that links to the management network, and the hosts are configured to do so that no communication between management interfaces and other network interfaces can be transferred. Management servers, database servers and consoles are only linked to the management network.

[5] In Distributed shared memory environment, access to remotely available data may be needed in a process. The goal is, therefore, to develop a distributed shared memory programming model that will optimize the use of shared memory, at the same time, a high degree of parallelism and effectiveness. This paper proposes a naive shared memory implementation that can be used in our work.

[6] In the event of attacks and high traffic, resources in the IDPS system gets exhausted and this paper presents different types of resources that get exhausted. It also enhances knowledge about new types of DoS and DDoS attacks are being explored in cloud computing, especially XML-DoS and HTTP-DoS attacks, and potential detection and mitigation techniques are being examined.

[7] Shared memory in scheduling designed for multi-core processors, recent developments in the shared memory scheduler setting have gained more prominence in multi-core systems with workload output relying heavily on which processing tasks are scheduled to run on the same or different sub-set of core. This paper clearly states about how problem arises while using shared memory and exhibits implementation of an optimized multi-core scheduler for which we executed applications using allocation created by our new to-core processor mapping algorithms on used ARM FL-2440 Board hardware architecture with Linux software in Large Systems running state.

[8] A synchronization method has been proposed in this paper which reduces contention, removes any possibility of deadlock and performs better than other models as the number of threads and the number of transactions increases. Message Passing is based on this technique. C implementation of this model takes advantage of the atomic actions of copying integer value to a memory location. It is proven that there can be no deadlock in the parallel system using this model for each process and that they are better than current models.

[9] Detection of zombies attack is extremely difficult in cloud system. The program of intrusion & prevention (IDPS) is used to detect potential attacks, gather information about them and then seek to avoid their occurrence and eventually report them to the network administrator. Any company uses these systems to identify the vulnerabilities in its security policies, log current attacks and threats and prevent an person from breaching security policies. This paper shows how IDPS systems plays a role in cloud based technologies.

[10] This paper addresses the problem of modelling lock-free concurrent queue data structures' energy behaviour. Our main contribution is a way to model lock-free queue implementation energy behaviour and parallel applications that use it. We decompose energy by focusing on steady state behaviour power dissipation activity which can be modelled separately and subsequently recombined into many useful metrics, such as energy per service.

[11] Bounded single-producer single-consumer FIFO queues are one of the simplest si-multaneous data-structure, requiring no more than sequential consistency for proper operation. Still, continuity of series is impossible multiprocessor shared memory theory and implementation by memory barriers contributes to substantial overhead efficiency and resources. This paper reviews the proof of optimisation and correctness of limited FIFO queues in the context of weak consistency of memory, building on the recent axiomatic formalization of Memory System C11.

[12] This paper presents a new lock-free algorithm, which offers many of the advantages of non-blocking algorithms while avoiding the true overhead non-blocking behaviour: the lock-free algorithm closely synchronizes with the data structure and demonstrates supe- rior application efficiency to all other students. The success of this algorithm means that a re-examination of the nonblocking literature may also benefit other practical problems.

[13] Operating systems and applications produce log-messages. Such communications provide vital information about the system's safety and operation. The communications in an intranet are also of great value for security monitoring, audit-checks and forensics. So, a system of logging which generates, relays, collects and archives log messages, must be monitored and managed just like all other components of the ICT infrastructure, to ensure that it is operating normally i.e., the logs are being collected and archived as desired.

[14] Monitoring logs is one of the most glamorous IT security work. It's boring, and gruelling at times. But if you forget or do something wrong, you can be targeted without ever realizing something.This Article gives breif information about how logs should be managed and what are the uses of using logrotate.

3. IMPLEMENTATION

Lists, Arrays and buffers are the some resources that can get exhausted due to high traffic/attack in IPDS systems. At the instant of resource exhaustion, system event get raised. In the IDPS systems there is one primary and many secondary processes running in the system. For the better performance of the system it is verified that the logging is only done by primary process. At the time of exhaustion system events are raised by the secondary processes, so there is a medium required for the communication between the primary and secondary processes in the system. Shared memory is used as the medium which is allocated at the time of booting. Therefore, the sysevents raised by secondary processes is written to the shared memory and from the same shared memory primary process reads the sysevents using the functions.

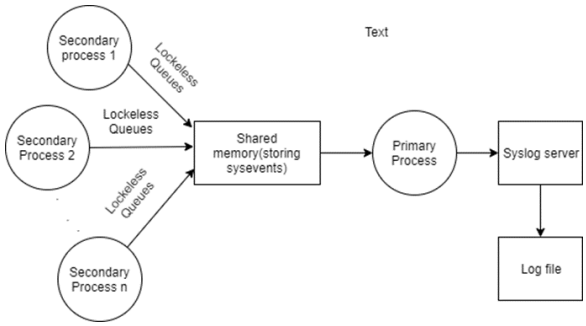

Figure 3.1: Logging Implementation

AS both primary and secondary processes work in tandem with the shared memory there is possibility of raising of synchronization issues, therefore to solve this Lockless queues(Explained in chapter 3) are used by each secondary process. It is designed in such a way that each secondary process will have their own lockless queues where sysevents raised by the particular processes are stored as shown in Figure 3.2.

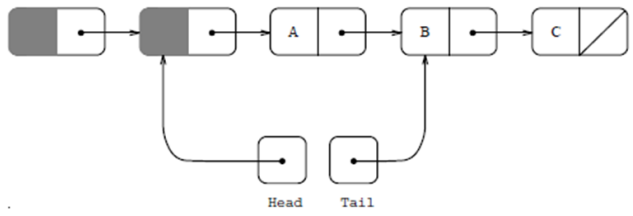

Figure 3.2: Lockless Queue

Figure 3.2 shows a queue as defined in chapter 3. There is a loop in the middle for enqueuing element C, and the tail pointer was not yet updated. A strict policy regarding the location of the tail pointer is employed and shown in the Figure 3.2 it always points to the last node on the list, or the one immediately preceding it. This is accomplished by the second CSW instruction, which attempts to update the tail pointer if its own node is not enqueued by the process.

Lockless queues consists of nodes of sysevent type and sysevents raised by each secondary process is added to the queue corresponding to the process. It is known that system event is raised only when there is problem in the system normal operation, therefore continuous raising of sysevents may lead to crashing of the system. To overcome this issue throttling concept is used. To avoid continuous raising of sysevents a time interval is fixed between raising of two consecutive sysevents, this will give a system fair amount of time to work with large number of sysevents. Throttling of sysevents makes sure that sysevents are raised in controlled manner.

There many log files already present in the system which has the logs related to many modules that comes under many severity levels. Resource exhaustion is considered as one of the emergency event happening in the system that may lead to the serious problem. In the present situation, if logs related to resource exhaustion are also directed to the already existing file they get lost in the pool of large number of logs related to the other modules, so there is a requirement for the new log file only to have a logs related to re- source exhaustion which will help to troubleshoot the issue. The new log file created is configured to be persistent across the reboot that helps the collection of logs related to many previously raised sysevents related to resource exhaustion.

Log files are the most valuable tools for security on the Linux system. The logrotate program is used to provide an up-to - date record of events happening on the system to the administrator. The logrotate utility can also be used to back up log files, so copies can be used to set patterns for use with the system. The Logrotate software is a manager of log data. This is used to process (or rotate) log files periodically, by deleting the oldest files from the system and generating new log files. It can be used to rotate depending on the age of the file or the size of the file, which will normally run automatically via the application cron. Additionally, the logrotate software can be used to compress log files and to customize user emails when they are rotated.

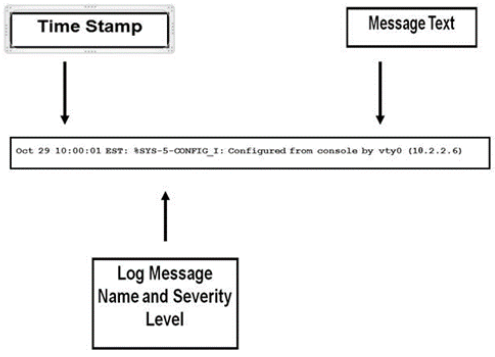

Figure 4.3: Log message format

Information that should be logged at the instant of resource exhaustion can generic to all the resources present in the system but only generic information won't help in the successful troubleshooting of the issue. Therefore, specific information related to the particular resource for which sysevent is raised is also necessary to debug the issue. To support generic information logging a function is created which can called in the event of exhaustion of any resource, then primary process can log the specific information while processing the particular sysevent related to the resources. The logs that gets printed into the log file will have particular format(as shown in Figure 4.3) that is supported by the syslog server.

4. CONCLUSIONS

Resource exhaustion happens in IDPS systems due to improper resource management or due to high inbound traffic or due to attacks. So at the time of exhaustion there is requirement of some information related to the resource which will help in successful troubleshooting of the issue. To achieve this, an intelligence is built on to the IDPS system in which processes which encounters exhaustion raises a system event and the generic and specific information related to the resource is logged to the new log file. Simultaneously the intelligence is implemented on to the IDPS systems to test its working. The informa-tion provided in the log file is made use to debug the issue by consuming minimum time and resulting in the effective management of resources in the IDPS systems. At the time of resource exhaustion a system event is raised by the encountered process. To efficient management of the system events throttling is used. A time interval of 2 minutes is fixed between two consecutive raising of sysevents. The secondary process that encounters resource exhaustion will raise a sysevent and add it to the lockless queues. These queues will be written to the shared memory between primary and secondary processes. Lockless queues are used to overcome the synchronization issues that can arise while reading and writing to the shared memory. Primary process will read the sysevents in the shared memory and log the information related those resources which will help in troubleshooting the issue.

ACKNOWLEDGEMENT

We would like to thank the all faculty members of the institute who helped us lot in calculating the facts and figures related to our paper. We would also like to thank the anonymous reviewers who provided helpful feedback on my manuscript.

REFERENCES

[1] A. A. Titorenko and A. A. Frolov, "Analysis of modern intrusion detection system," in 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), IEEE, 2018. DOI: 10.1109/eiconrus.2018.8317049.

[2] D. Mudzingwa and R. Agrawal, "A study of methodologies used in intrusion detec- tion and prevention systems (IDPS)," in 2012 Proceedings of IEEE Southeastcon, IEEE, 2012. DOI: 10.1109/secon.2012.6197080.

[3] L. Hui and C. Yonghui, "Research intrusion detection techniques from the perspec- tive of machine learning," in 2010 Second International Conference on Multimedia and Information Technology, IEEE, 2010. DOI: 10.1109/mmit.2010.161.

[4] W. Bul 'ajoul, A. James, and S. Shaikh, "A new architecture for network intrusion detection and prevention," IEEE Access, vol. 7, pp. 18 558–18 573, 2019. DOI: 10. 1109/access.2019.2895898.

[5] D. Dangi, S. Bhandari, and A. Bhagat, "Analysis of shared memory in distributed and non distributed environment," in 2016 Fifth International Conference on Eco- friendly Computing and Communication Systems (ICECCS), IEEE, 2016. DOI: 10. 1109/eco-friendly.2016.7893253.

[6] A. Bonguet and M. Bellaiche, "A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing," Future Internet, vol. 9, no. 3, p. 43, 2017. DOI: 10.3390/fi9030043.

[7] J. Shan, Y. Chen, Q. Diao, and Y. Zhang, "Parallel information extraction on shared memory multi-processor system," in 2006 International Conference on Par- allel Processing (ICPP'06), IEEE, 2006. DOI: 10.1109/icpp.2006.58.

[8] L. Beringer, G. Stewart, R. Dockins, and A. W. Appel, "Verified compilation for shared-memory c," in Programming Languages and Systems, Springer Berlin Hei- delberg, 2014, pp. 107–127. DOI: 10.1007/978-3-642-54833-8_7.

[9] A. Sawant, "A comparative study of different intrusion prevention systems," in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, 2018. DOI: 10.1109/iccubea.2018.8697500.

[10] A. Atalar, A. Gidenstam, P. Renaud-Goud, and P. Tsigas, "Modeling energy con- sumption of lock-free queue implementations," in 2015 IEEE International Parallel and Distributed Processing Symposium, IEEE, 2015. DOI: 10.1109/ipdps.2015.31.

[11] N. M. Le, A. Guatto, A. Cohen, and A. Pop, "Correct and efficient bounded FIFO queues," in 2013 25th International Symposium on Computer Architecture and High Performance Computing, IEEE, 2013. DOI: 10.1109/sbac-pad.2013.8.

[12] S. Lumetta and D. Culler, "Managing concurrent access for shared memory ac- tive messages," in Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing, IEEE Comput. Soc. DOI: 10.1109/ipps.1998.669925.

[13] H. Tsunoda and G. M. Keeni, "Managing syslog," in The 16th Asia-Pacific Network Operations and Management Symposium, IEEE, 2014. DOI: 10. 1109 / apnoms. 2014.6996575.

[14] "Logging and log management," Network Security, vol. 2013, no. 2, p. 4, 2013. DOI:10.1016/s1353-4858(13)70026-6.

[15] S. S. Tirumala, H. Sathu, and A. Sarrafzadeh, "Free and open source intrusion detection systems: A study," in 2015 International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, 2015. DOI: 10.1109/icmlc.2015.7340923.

[16] S. Xin, "Research of intrusion detection system," in 2013 International Conference on Computational and Information Sciences, IEEE, 2013. DOI: 10.1109/iccis. 2013.385.

[17] P. Xu, W. Zhuang, X. Li, J. Liu, M. Sun, L. Huang, H. Zhang, J. Pan, and X. Zhang, "The research of distributed shared memory technology in power system," in 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, 2017. DOI: 10.1109/itnec.2017.8285008.

[18] J. J. Tithi, D. Matani, G. Menghani, and R. A. Chowdhury, "Avoiding locks and atomic instructions in shared-memory parallel BFS using optimistic paralleliza- tion," in 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, IEEE, 2013. DOI: 10.1109/ipdpsw.2013.241.

[19] K. Kowalski and M. Beheshti, "Analysis of log files intersections for security en- hancement," in Third International Conference on Information Technology: New Generations (ITNG'06), IEEE, 2006. DOI: 10.1109/itng.2006.32.

[20] J. Andrews, "Testing using log file analysis: Tools, methods, and issues," in Pro- ceedings 13th IEEE International Conference on Automated Software Engineering (Cat. No.98EX239), IEEE Comput. Soc. DOI: 10.1109/ase.1998.732614.

[21]]G. Jones, "Penetrating the cloud," Network Security, vol. 2013, no. 2, pp. 5–7, 2013.DOI: 10.1016/s1353-4858(13)70028-x.K. Elissa, "Title of paper if known," unpublished.