IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8, August 2017

Enhanced Forensics Enabled Cloud through Secured Logging as a Service

[1] Ms.D.Gomathi.M.Sc,[2] Dr.B.Mukunthan PhD
[1] Research scholer, [1][2] Department of Computer Engineering
[1] Jairams college of Arts and Science,Karur, India, [2] Jairams college of Arts and Science, Bharathidasan University, Tiruchirapalli.

***Abstract*— The principle goal of this venture is to build up a secured logging as an administration in cloud engineering. So in the proposed strategy, security and safeguarding techniques are upgraded. The secured logging contains six noteworthy functionalities to guarantee more securities: Correctness, Confidentiality, information logs, Privacy, Preservation and VPS (Virtual intermediary server). The accuracy manages rightness information of the genuine history. Classification manages delicate data not showing amid seek. Information logs manages the information history for distinguishing fitting clients. Security conspire manages document connecting and information get to history. Safeguarding manages upgraded shading code. Lastly VPS manages the intermediary server for virtual information get to. The usage of the above given strategies are appeared in any condition, which manages enormous number of information with different clients. There are very little contrast amongst programmers and interlopers in the cloud design. Programmers are from different systems mean while interlopers are from same systems. Programmers can be kept away from and gatecrashers are can't be dodged. This is on account of gatecrashers may know about the system where they will interrupt. With the goal that secured logging as an administration is much imperative for all sort of cloud server condition so as to give appropriate login to approved client and triggers out the unapproved clients.**

***Index Terms*— Secured Logging as a Service, Data Logs, Intrusion detection system, Cloud Architecture, Privacy and preservation.**

1. INTRODUCTION

This is entitled as “Enhanced secured logging as a service using privacy preservation network in cloud architecture” is developed using VM ware as the cloud simulation tool, SQL Server as the database and C# as coding language. Ajax 2.0 used as client server tool and scripting language as java script. While you are entering the network there are large number of problem you have to face. Here introducing a new security login method that used to predict who is trying to hack your details like intruder, hacker, or user and it draw the chat based on their login method. Based on it block that particular files. Now none can access that file until admin unblock it.

RELATED WORKS

The requirement for secure logging is surely knew by the security experts, including both scientists and professionals. [1] The capacity to productively confirm all (or a few) log sections is imperative to any application utilizing secure logging strategies. In this paper, we start by analyzing best in class in secure logging and recognize a few issues inborn to frameworks in view of confided in outsider servers. [1] We at that point propose an alternate way to deal with secure logging in view of as of late created Forward- Secure Sequential Aggregate validation systems. Our approach offers both space-productivity and provable security. We additionally explore the idea of changelessness with regards to forward secure successive total validation to give better grained confirmation. At long last, we cover some involvement with a model based upon a famous code form control framework. Framework logs are an imperative piece of any protected IT framework. They record critical occasions occurred in the past, for example, client action, program execution status, framework asset use, information changes, and so forth [2] For various reasons most web administrations store data about their clients' solicitations on a long haul premise. One reason to do this is a specialist co-op's should have the capacity to build up responsibility for specific solicitations. Despite the required exertion, IP locations can regularly be connected to a real individual getting to an administration. Since the put away demand data contains individual information, [2] we have to consider client requests in regards to security. In times when organizations welcome the estimation of individual data, clients will progressively value the assurance of their profitable individual information. A great many surveys demonstrates that security worries as for the utilization of the web are on the ascent. [3] This paper gives a short diagram on the as of now existing specialized answers for unknown correspondence. The issue of mysterious correspondence is characterized, and its essential arrangement is depicted. Viable arrangements, generally in light of the essential plan, are talked about. [3] They give namelessness to associations when all is said in done, and secrecy in particular applications, similar to email and the World Wide Web. The diverse arrangements are depicted, and an examination is given. Extra comments are made as for obscurity renouncement, U.S. send out confinements, and the execution that can be gotten. [4] Many security frameworks, regardless of whether they ensure protection, secure electronictrade exchanges, or utilize cryptography for something else, don't straightforwardly avoid extortion. Or maybe, they distinguish endeavors at misrepresentation sometime later, give confirmation of that extortion with a specific end goal to convict the liable in an official courtroom, and expect that the legitimate framework will give a back channel" to hinder additionally endeavors. We trust that handled frameworks ought to perceive this key requirement for location systems, and give review capacities that can survive both effective and unsuccessful assaults. [4] Additionally, an unalterable log should make it troublesome for assailants to cover their tracks, implying that the casualties of the assault can rapidly discover that their machine has been assaulted, and take measures to contain the harm from that assault. [5] The mystery put away on the PC is the leader of a hash chain, changing by means of a cryptographic one-way work each time a section is composed to the log. This mystery is utilized to process a cryptographic message verification code (MAC) for the log each time a passage is included, and alternatively to scramble the log also. On the off chance that the framework is traded off, the assailant has no real way to recoup the mysteries used to make the MACs or decoding keys for sections in the log which have as of now been finished. He can erase the log altogether, however can't adjust it without location. [5] Later, the executive can utilize the first mystery to reproduce the hash chain and check whether the logs are as yet in place. To shield an aggressor from meddling with this procedure, this ought to occur on a different, secure machine. Macintoshes may likewise be sent to another machine as they're composed; at that point they can fill in as responsibilities regarding log passages. The clusters [19] have been dynamically reconfigured whenever the nodes move out of the cluster. Some of the objectives of clustering can be achieved using advanced neural network algorithms [3], which ensures the importance of computational methods [12] [14] such as Neural-Fuzzy mapping that are much cheaper and faster than conventional experimental methods.

METHODOLOGIES

Properties of Secure Logging as a Service

Secure log administration benefit in view of the distributed computing worldview. We will in this manner investigate our structure against these properties.

Correctness:

Log information is valuable just in the event that it reflects genuine history of the framework at the season of log era. The put away log information ought to be right, that is, it ought to be precisely the same as the one that was created.

Verifiability:

It must be conceivable to watch that all sections in the log are available and have not been adjusted. Every passage must contain enough data to check its credibility autonomous of others. On the off chance that a few passages are modified or erased, the capacity to independently confirm the rest of the sections (or squares of passages) makes it conceivable to recuperate some valuable data from the harmed log. Additionally, the individual sections must be connected together in a way that makes it conceivable to decide if any passages are absent.

Confidentiality:

Log records ought not be coolly perused capable or accessible to assemble delicate data. True blue hunt access to clients, for example, reviewers or framework managers ought to be permitted. What's more, since nobody can keep an assailant who has com-guaranteed the logging framework from getting to touchy data that the framework will put in future log sections, the objective is to shield the pre traded off log records from classification breaks.

Privacy:

Log records ought not be calmly traceable or linkable to their sources amid travel and away.

Gaussian Mixture and Keystroke

Distributed computing security is a developing subarea of PC security, arrange security, and, all the more extensively, data security. It alludes to an expansive arrangement of strategies, advancements, and controls conveyed to ensure information, applications, and the related framework of distributed computing. The technique utilized here for security is keystroke logging. This enables just the correct client to login at the perfect time. It is the activity of following the keys struck on a console, with the goal that the individual utilizing the console is unconscious that their activities are being observed. At whatever point a client is made, the keystroke time of writing his/her secret key ought to be noted. At the point when a client logins to sends subtle elements, the keystroke time for writing his/her secret key ought to matches with the time that is produced in the client creation. So this will gives a well security to the client's id and secret key from programmers.

ALGORITHM SLAS LOGIN

- Objective function f(x),
- x = (x1, ..., xd) T
- Make primary population of fireflies xi (I = 1, 2, ..., n)
- Light greatness Ii at xi is determined by f(xi)
- Define light absorption coefficient γ
- while (t <max generation),
- for i=1: n all n keystroke
- for j =1 : i all n c1,c2,c3
- if<Ij >Ii),
- Move firefly i towards j in d-dimension; end if
- Attractiveness varies with distance r via exp[−γr]
- Evaluate new solutions and update light intensity
- end for j
- end for i
- Rank the login and find the current best
- end while
- Post process results and visualization

RESULT AND DISCUSION

This paper manages safely keeping up log records which contains touchy and secret data. In this paper we actualize cloud based log administration to beat the substantial costs that happens while conveying a protected logging framework. This includes a secured protection signing in a cloud domain to get to the log records. For making a cloud based secure logging the accompanying design is utilized:

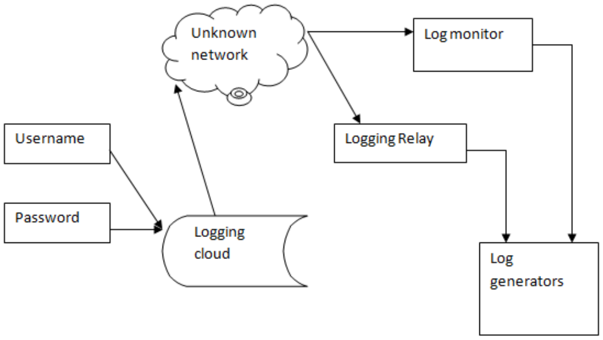


Figure 1: Architecture Diagram

The Above architecture shows fig 1 the usage of Log generator, Logging Client or Logging relay, Logging cloud, etc. Each one has the different functions. The functions are as follows:

1. Log generator:

The log generators are used for generating log data. Every cloud based secured management system has number of log generators which is used to generate log files which are stored temporarily and are then pushed to logging relay.

2. Logging Client or Logging Relay:

The logging relay is used for receiving the group of records generated by one or more log generators. The register data is then transferred from the generators to the client in batches depending on amount of log data waiting to be transferred.

3. Logging Cloud:

It is the long term storage and maintenance service to log data received from different logging clients belonging to different organizations. It is maintained by the cloud service provider.

4. Log monitor:

They are used for monitoring and reviewing the log data. They can generate queries to retrieve log data from the cloud. Based on the log data the log monitors will analyze the log data. The log data from the log cloud will be permanently deleted once the log monitors requests. Based on the above described architecture the cloud based secured logging system is developed. In this system the logging clients and the log generators can communicate with each other in an authenticated way. If an unauthorized person or an attacker or a hacker occurs the system will monitor and send the message to the corresponding authenticated users.

This is implemented by the following steps
1. Basically when a user logging into the cloud environment to access logs data, he or she should enter their username and password.
2. In order to improve the given system we used many security measures to avoid the occurrence of unauthorized users.
3. The security measures are Keystroke Logging, Color code Security mechanism, Last accessed Login date, Last uploaded log data and Last accessed system IP.
4. Let us begin with the security measure

KEYSTROKE LOGGING:

This security method gets the time of the user typing the password for every key stroke in the keyboard.This uses the following methods and this event starts on Onkeydescription() function occurs.

5. The next security mechanism is Color code security mechanism. It is nothing but the user should select three colors for the logging and encrypted codes for the colors are generated. Every time when the user login into the cloud the user should select the colors and the code should matches the encrypted code.

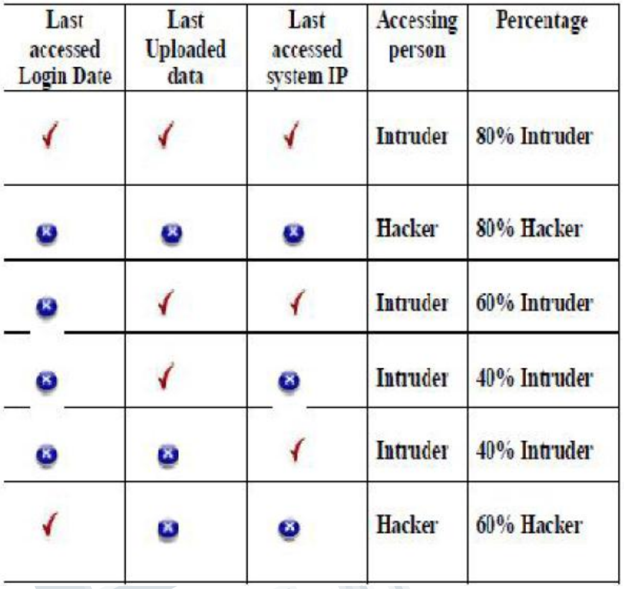
It is implemented by the following
method.

Consider the Color 1 as a Red
Color 2 as a Green and
Color 3 as a Blue.
The codes for colors be Color 1 (Red)
Code = 247861
Color 2 (Green) Code = 145782
Color 3 (Blue) Code = 547643

Then the encrypted code for the above three colors
be 941286.

Every time when the users sign up with the above colors then the encrypted code should match so that high security will be provided.

If the user fails in logging from the above two mechanism then he/she should undergo the security mechanisms Last accessed Login date, Last uploaded log data and Last accessed system IP to find whether the person is a intruder (one of the member of the cloud network) or the hacker. From the above mentioned security mechanism we can able to find whether the logging person is hacker or an intruder.
The below table shows how it is implemented.



Last accessed Login Date	Last Uploaded data	Last accessed system IP	Accessing person	Percentage
✓	✓	✓	Intruder	80% Intruder
⊗	⊗	⊗	Hacker	80% Hacker
⊗	✓	✓	Intruder	60% Intruder
⊗	✓	⊗	Intruder	40% Intruder
⊗	⊗	✓	Intruder	40% Intruder
✓	⊗	⊗	Hacker	60% Hacker

CONCLUSION

In this way we are inferring that all the outcome gotten by the conferred unique. In this paper, We incorporate a recently proposed edge plan and codes over examples for secured logging. The plan underpins encoding, sending, and fractional unscrambling operations distributed for shading code. To decode a message of k hinders that are scrambled and encoded ton code word images, every server just needs to in part unscramble two codeword images in our framework. The precision level of discovering Hacker, Intruder and User is high. Besides, every capacity server freely performs encoding and re-encryption and every server autonomously perform halfway decoding. Our capability of framework and some recently planned content addressable document frameworks and capacity frameworks are extremely good. Our capacity servers go about as capacity hubs in a substance addressable capacity framework for putting away substance addressable squares. Our key servers go about as get to hubs for giving a front-end layer, for example, a conventional document framework interface. Additionally contemplate on natty gritty collaboration is required.

ACKNOWLEDGMENT

I am heart fully thankful to my project guide Dr.B.MUKUNTHAN,MCA.,Ph.D., Research Advisor, PG & Research Department of Computer Science, Jairams Arts and Science College, Karur, for his valuable guidance, suggestions, advice, and mentoring during my graduate career.

REFERENCES

1. A new approach to secure logging, d. Ma and g. Tsudik, "a new approach to secure logging," acm trans. Storage, vol. 5, no. 1, pp. 2:1–2:21, mar. 2009.

2. Pseudonymizing unix log file, u. Flegel, "pseudonymizing unix log file," in proc. Int. Conf. Infrastructure security, lncs 2437. Oct. 2002, pp. 162–179.

3. Mukunthan. B, "A Neural Network Approach for Precise Pattern Identification of Human DNA", International Journal of Neural Networks and Applications, Vol. 5, Issue 1, pp.21-27, 2012.

4. Internet anonymity: problems and solutions, c. Eckert and a. Pircher, "internet anonymity: problems and solutions," in proc. 16th ifip tc- 11 int. Conf. Inform. Security, 2001, pp. 35–50

5. Security audit logs to support computer forensics, b. Schneier and j. Kelsey, "security audit logs to support computer forensics," acm trans. Inform. Syst. Security, vol. 2, no. 2, pp. 159– 176, may 1999.

6. logcrypt: forward security and public verification for secure audit logs, j. E. Holt, "logcrypt: forward security and public verification for secure audit logs," in proc. 4th Australasian inform. Security workshop, 2006, pp. 203–211.

7. R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," inProc. 12th Ann. USENIX Security Symp., Aug. 2004, pp. 21–21.

8. The Tor Project, Inc. (2011, Sep.) Tor: Anonymity Online [Online]. Available: http://www.torproject.org

9. D. Dolev and A. Yao, "On the security of public key protocols," IEEE Trans. Inform. Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983.

10. A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, Nov. 1979.

11. G. R. Blakley, "Safeguarding cryptographic keys," inProc. Nat. Comput. Conf., Jun. 1979, p. 313.

12. Mukunthan. B and Nagaveni. N, "Identification of Unique Repeated Patterns, Location of Mutation in DNA Finger Printing Using Artificial Intelligence Technique", International Journal of Bioinformatics Research and Applications, Vol. 10, Issue. 2, pp. 157-176, 2014, doi.org: 10.1504/IJBRA.2014.059516

13. R. Ostrovsky and M. Yung, "How to withstand mobile virus attack," in Proc. 10th Ann. ACM Symp. Principles Distributed Comput., Aug. 1991, pp. 51–59.

14. Mukunthan. B and Pushpalatha. A, "Automation of DNA Finger Printing for Precise Pattern Identification using Neural- Fuzzy Mapping Approach", International Journal of Computer Applications, Vol. 13, Issue. 3, pp.16-24, 2011.

15. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," inProc. 15th Ann. Int. Cryptology Conf., Aug. 1995, pp. 339–352.

16. I. Teranishi, J. Furukawa, and K. Sako, "ktimes anonymous authen-tication (extended abstract)," in Proc. 10th Int. Conf. Theor. Appl. Cryptology Inform. Security, LNCS 3329. 2004, pp. 308–322.

17. D. L. Wells, J. A. Blakeley, and C. W. Thompson, "Architecture of an open objectoriented database management system,"IEEE Comput., vol. 25, no. 10, pp. 74–82, Oct. 1992.

18. K. Nørv˚ ag, O. Sandst˚ a, and K. Bratbergsengen, "Concurrency control in distributed object oriented database systems," in Proc. 1st East-Eur. Symp. Adv. Databases Inform. Syst., Sep. 1997, pp. 32–32.