International Journal of Science Engineering and Advance Technology

A Complete System To Securely Outsource Log Records To A Cloud Provider

[1]kodari Shravan Kumar,[2] Vishnuprasad Goranthala
[1]kodarishravan514@Gmail.Com,
1,2 Balaji Institute Of Technology & Science Narsampet Warangal

Abstract:
In View Of The Fact That Log Files Hold Record Of Most System Events Including User Activities They Turn Out To Be An Important Target For Malicious Attackers. An Attacker Breaking Into A System Normally Would Try Not To Leave Traces Of His Or Her Activities Behind. As A Result The First Thing An Attacker Often Does Is To Harm Log Files Or Break Off The Logging Services. In Addition The Sensitive Information Contained In Log Files Often Directly Contributes To Confidentiality Breaches. An Illustration Of This Is When Logs Contain Database Transaction Data. Regularly Log Information Can Be Cooperative To An Attacker In Attainment Unauthorized Access To System. One Example Of This Is The Case When A User Incorrectly Enters Her Password In The Username Field While Logging Into A System. Logging Programs Will Store Up The Password As The User-Id To Evidence The Information That A User Has Failed To Log In. Last But Not Least Information In Log File Can Also Be Used To Cause Privacy Breaches For Users In The System Since The Log File Contains Record Of All Events In The System. It Is Extremely Significant That Logging Be Provided In A Protected Manner And That The Log Records Are Effectively Protected For A Predetermined Amount Of Time Maybe Even Indefinitely. Traditional Logging Protocols That Are Based On Syslog Have Not Been Designed With Such Security Features In Mind.
Keywords: Cloud Computing, Logging, Privacy, Security.

Introduction:
The Up-And-Coming Concept Of Cloud Computing Promises A Low Cost Chance For Organizations To Store And Manage Log Records In A Proper Manner. Organizations Can Subcontract The Long-Term Storage Requirements Of Log Files To The Cloud. The Challenges Of Storing And Maintaining The Log Records Become A Concern Of The Cloud Provider. Since The Cloud Provider Is Providing A Single Service To Many Organizations That It Will Advantage From Economics Of Scale. Approaching Log Records To The Cloud Though Introduces A New Challenge In Storing And Maintaining Log Records. The Cloud Provider Can Be Honest But Curious. This Means That It Can Try Not Only To Get Private Information Directly From Log Records But Also Link Log Record Related Activities To Their Sources. No Existing Protocol Addresses All The Challenges That Happen When Log Storage And Maintenance Is Pushed To The Cloud. Securely Maintaining Log Records Over Extended Periods Of Time Is Very Significant To The Proper Functioning Of Any Organization. Integrity Of The Log Files And That Of The Logging Process Need To Be Ensured At All Times. In Addition As Log Files Often Contain Sensitive Information Confidentiality And Privacy Of Log Records Are Evenly Important. However Deploying A Secure Logging Infrastructure Involves Considerable Capital Expenses That Many Organizations May Find Irresistible. Delegating Log Management To The Cloud Appears To Be A Feasible Cost Saving Measure. In This Paper We Identify The Challenges For A Secure Cloud-Based Log Management Service And Propose A Framework For Doing The Same.

Related Work:
Most Of The Approaches Are Based On Syslog Which Is The De Facto Standard For Network Wide Logging Protocol. The Syslog Protocol Uses UDP To Move Log Information To The Log Server. Thus There Is No Dependable Delivery Of Log Messages. Moreover Syslog Does Not Defend Log Records During Transit Or At The End-Points. Syslog-Ng Is A Substitute That Is Backward Compatible With Syslog. Some Of Its Features Include Support For Ipv6 Ability To Transfer Log Messages Reliably Using TCP And Sieve The Content Of Logs Using Regular Expressions. Syslog-Ng Recommend Log Record Encryption Using SSL During Transmission So As To Protect The Data From Confidentiality And Integrity Violate While In Transit. Though Syslog-Ng Does Not Guard Against Log Data Modifications When

It Resides At An End-Point. Syslog-Sign Is An Another Augmentation To Syslog That Adds Origin Authentication, Message Integrity, Replay Resistance, Message Sequencing And Detection Of Missing Messages By Using Two Additional Messages Signature Blocks And Certificate Blocks. Regrettably If Signature Blocks Connected With Log Records Get Deleted After Authentication Tamper Proof And Forward Integrity Is Only Partially Fulfilled. Syslog-Sign Also Does Not Supply Confidentiality Or Privacy During The Transmission Of Data Or At The End Points.

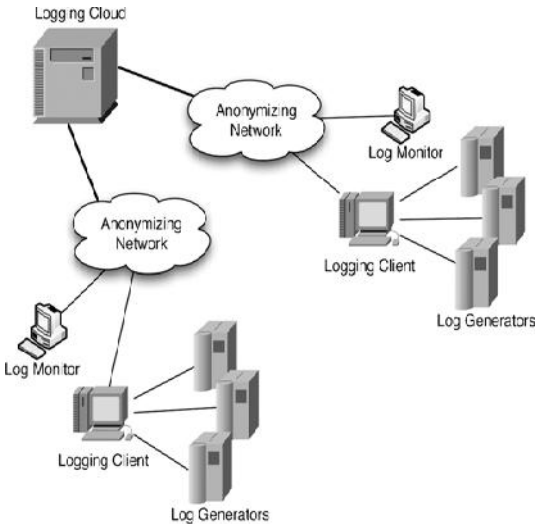
Existing System:

The Cloud Provider Can Be Honest But Curious. This Means That It Can Attempt Not Only To Get Confidential Information Directly From Log Records But Also Link Log Record Related Activities To Their Sources. No Existing Protocol Addresses All The Challenges That Happen When Log Storage And Maintenance Is Pushed To The Cloud. We Also Develop Protocols So That Log Records Can Be Transmitted And Retrieved In An Unspecified Manner Over An Existing Anonymizing Infrastructure Such As Tor. This Successfully Prevents The Cloud Provider Or Any Other Observer From Correlating Requests For Log Data With The Requester Or Generator. At Last We Develop A Proof-Of-Concept Prototype To Exhibit The Viability Of Our Approach And Discuss Some Early Experiences With It.

Proposed System:

The Chief Thought Is That Log Records Are First Processed By A Pseudonymizer Before Being Archived. The Pseudonymizer Filters Out Identifying Features From Definite Fields In The Log Record And Substitutes Them With Carefully Crafted Pseudonyms. Thus This Protocol Does Not Guarantee Correctness Of Logs. To Protect Prêt Compromise Log Data From Post Compromise Insertion, Deletion, Modification, And Reordering. Forward Integrity Is Established By A Secret Key That Becomes The Starting Point Of A Hash-Chain. The Hash-Chain Is Produced By A Cryptographically Strong One-Way Function In Which The Key Is Changed For Every Log Record. If A Unit Other Than The Logging Client Initiates The Log Delete Request It Has To Obtain The Delete Tag From The Logging Client And The Relevant Upload-Tag. Note That Deletion Or Rotation Of Log Records Is A Significant Operation And As A Result Needs To Be Undertaken After Substantial Reflection.

System Architecture:



We Suppose That The Organization Maintains The Log Generators And The Logging Client. The Log Monitor Can Be Preserved By The Same Organization Or Can Be A Separate Entity. The Logging Client Can Also Play The Role Of A Log Monitor. We Develop Our Model Assuming That The Log Monitor Is A Separate Entity That Is Trusted By The Logging Client. Since The Logging Client And Log Monitor Function Autonomous Of Each Other They Can Communicate Only In An Asynchronous Manner. This Means That If A Logging Client Wants To Post Some Data To The Log Monitor Or Vice Versa The Sender Cannot Wait For The Receiver To Be Online To Receive The Data. As A Result The Sender Has To Put Out The Data In Some Location And The Receiver Needs To Get Back The Data From There When Needed. The Logging Cloud Makes Easy This Communication By Receiving And Servicing Suitable Requests. The Logging Client And The Log Monitor Communicate With The External World Over An Unencrypted Network That Provides Anonymous Full-Duplex Communication. Our Proof-Of Concept Prototype Gets This Service From The Tor Network. Attacks On The Tor Network That Breach Secrecy Of Communicating Parties Have Been Well Studied And Solutions Proposed.

Anonymous Upload-Tag Generation:

To Make Sure That This Key Value Cannot Be Traced Back To The Logging Client That Uploaded The Data Nor Does The Log Monitor That Look For The Data. For This Purpose The Log Data Is Stored At The Cloud Indexed By An Anonymously Generated Upload-Tag. This Upload-Tag Is Shaped By The Logging Client In

 ISSN 2321-6905

Cooperation With The Log Monitor. It Has The Possessions That It Is Created By Publicly Available Information. Conversely A Given Upload-Tag Cannot Be Linked Either To The Corresponding Logging Client Or A Log Monitor. To Recover Log Data From The Cloud The Log Monitor Sends A Retrieve Request To The Logging Cloud Using An Upload Tag. The Upload-Tag Is Not Sent In An Encrypted Manner. Thus Any Adversary Can Use The Upload-Tag To Regain The Corresponding Log Data. However The Log Data Can Be Decoded If And Only If The Analogous Decryption Key Is Available.

Anonymous Upload:

The Logging Client Sends A Formatted Message Containing The Upload-Tag, A Delete Tag And A Batch Of Previously Prepared Log Data. The Delete-Tag Is Used Later On To Delete Or Rotates The Log Data If The Logging Client Or Another Entity Authorized The Logging Client Needs To Do So. Since All Messages To The Logging Cloud Are Over Anonymous Channels Any Unit Can Potentially Ask The Logging Cloud To Delete A Log Data. To Avoid This From Happening The Logging Cloud Challenges The Supplicant Of Such A Delete Operation To Prove That It Has The Necessary Authorization. Of Course This Can Be Achieved Unimportantly By The Logging Client Since It Can Be Anonymously Genuine By The Logging Cloud.

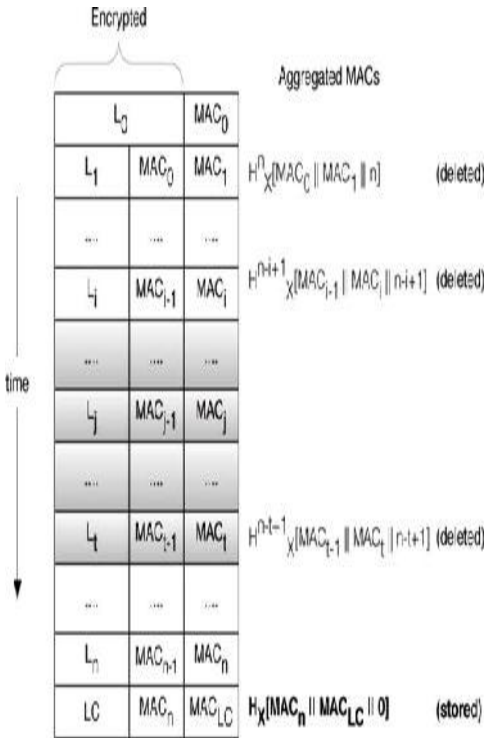
Anonymous Retrieve:

This Protocol Is Straightforward. The Unit That Needs To Download Log Data Most Of The Time The Log Monitor Sends A Retrieve Request Anonymously Jointly With The Upload-Tag Equivalent To The Desired Log Data. The Logging Cloud Gets The Data From Its Storage Space And Sends It Over The Anonymous Channel To The Requester. The Cloud Provider Does Not Have To Validate The Requester. This Is Because By Asset Of The Log Batches Being Encrypted The Retrieved Data Is Useful Only To Those Who Have The Valid Decryption Keys.

Anonymous Delete:

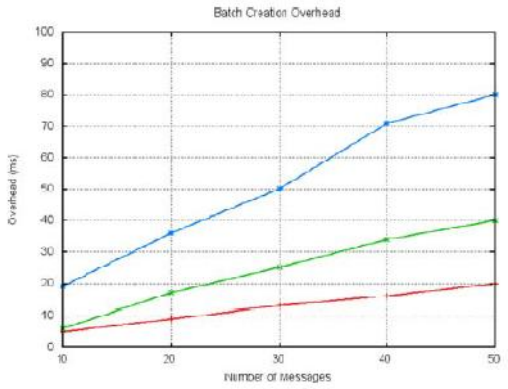
To Delete Log Data The Delete Requester Sends A Suitable Delete Message To The Logging Cloud. In Response The Logging Cloud Throws A Challenge To The Requester. The Requester Proves Approval To Delete By Presenting A Correct Delete Tag.

Log Batch Uploaded On Cloud:



We Use A Dependable Delivery Mechanism Of Log Records From Log Generators To The Logging Clients Which Though Not Explicitly Indicated Can Be Amplified By Cryptographically Strong Message Digests And Authentication Of End Points Of Communication The Records Produced By A Generator Are The Ones Received By The Logging Client And Used In The Log Batch Preparation. Our Scheme Can Also Detect Truncation Attacks. If The Attacker Deletes Some Trailing Log Records On The Logging Client The Existing Collective Message Authentication Code Will Be Cause To Be Unacceptable. An Adversary That Is Camouflaged As A Logging Client Will Be Detected By The Security In Place At The Organization.

Experimental Results:



We Designed A Set Of Simulations That Allows Us To Analyze The Efficiency Of Different Encryption Settings In Our Prototype. We Created Test Situations To Understand The Time Overhead That Is Introduced By The Log Record Preparation Process. In Our Experiments We Used Batch Sizes Of 10, 20, 30, 40, And 50 Encrypted Log Records. For Each Log Batch We Deliberate Three Different Log Preparation Settings. First We Deliberate The Time That It Takes To Fill Up The Log Batch With Log Records Without Any Security Related Preparation. Therefore The First Experiment Provides Us The Best Attainable Performance In Our Prototype. Second We Measured An Overhead For Filling Up Log Batch With Biased Security Related Log Preparation. This Includes Unencrypted Log Records No Confidentiality With Calculated MAC And Aggregated MAC Functions Correctness, Tamper Resistance And Verifiability Assurances. Third We Intended An Overhead For Security Preparation That Includes Encrypted Log Records With Corresponding MAC And Aggregated MAC Functions Correctness, Tamper Resistance, Verifiability And Confidentiality Assurances.

Conclusion:

We Proposed A Complete System To Firmly Outsource Log Records To A Cloud Provider. We Reviewed Existing Solutions And Identified Troubles In The Current Operating System Based Logging Services Such As Syslog And Sensible Difficulties In Some Of The Existing Secure Logging Techniques. We Then Proposed A Complete Scheme That Addresses Security And Integrity Issues Not Just During The Log Generation Phase But Also During Other Stages In The Log Management Process Including Log Collection, Transmission, Storage And Retrieval. One Of The Unique Confronts Is The Problem Of Log Privacy That Arises When We Outsourced Log Management To The Cloud. Log Information In This Case Should Not Be Casually Linkable Or Noticeable To Their Sources During Storage, Retrieval And Deletion. We Provided Anonymous Upload, Retrieve And Delete Protocols On Log Records In The Cloud Using The Tor Network. The Protocols That We Developed For This Purpose Have Potential For Usage In Many Different Areas Including Anonymous Publish-Subscribe.

References:

[1] K. Kent And M. Souppaya. (1992). *Guide To Computer Security Log Management, NIST Special Publication 800-92* [Online]. Available: Http://Csrc.Nist.Gov/Publications/Nistpubs/800-92/SP800-92.Pdf

[2] U.S. Department Of Health And Human Services. (2011, Sep.). *HIPAA—General Information* [Online]. Available: Https://Www.Cms.Gov/ Hipaageninfo

[3] PCI Security Standards Council. (2006, Sep.) *Payment Card Industry (PCI) Data Security Standard—Security Audit Procedures Version 1.1* [Online]. Available: Https://Www.Pcisecuritystandards.Org/Pdfs/ Pci−Audit−Procedures−V1-1.Pdf

[4] Sarbanes-Oxley Act 2002. (2002, Sep.). *A Guide To The Sarbanes-Oxley Act* [Online]. Available: Http://Www.Soxlaw.Com/

[5] C. Lonvick, *The BSD Syslog Protocol*, Request For Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.

[6] D. New And M. Rose, *Reliable Delivery For Syslog*, Request For Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.

[7] M. Bellare And B. S. Yee, "Forward Integrity For Secure Audit Logs," Dept. Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.

[8] Balabit IT Security (2011, Sep.). *Syslog-Ng—Multiplatform Syslog Server And Logging Daemon* [Online]. Available: Http://Www.Balabit. Com/Network-Security/Syslog-Ng

[9] J. Kelsey, J. Callas, And A. Clemm, *Signed Syslog Messages*, Request For Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.

[10] 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.

[11] U. Flegel, "Pseudonymizing Unix Log File," In *Proc. Int. Conf. Infrastruture Security*, LNCS 2437. Oct. 2002, Pp. 162–179.

[12] C. Eckert And A. Pircher, "Internet Anonymity: Problems And Solutions," In *Proc. 16th IFIP TC-11 Int. Conf. Inform. Security*, 2001, Pp. 35–50 .

[13] M. Rose, *The Blocks Extensible Exchange Protocol Core*, Request For Comment RFC 3080, Internet Engineering Task Force, Network Working Group, Mar. 2001.

[14] 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.

[15] J. E. Holt, "Logcrypt: Forward Security And Public Verification For Secure Audit Logs," In *Proc. 4th Australasian Inform. Security Workshop*, 2006, Pp. 203–211.

[16] R. Dingledine, N. Mathewson, And P. Syverson, "Tor: The Secondgeneration Onion Router," In *Proc. 12th Ann. USENIX Security Symp.*, Aug. 2004, Pp. 21–21.