Scientific Journal Impact Factor (SJIF) : 1.711

International Journal of Modern Trends in Engineering and Research

IJMTER

ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

A SECURE AND TRUSTWORTHY RESOURCE SHARING IN SOCIAL NETWORK AND CLOUD COMPUTING

Mr Donbyntalang Dewkhaid[1], Dr.S.Uma[2], Mr.P. Samuel Kirubakaran[3], Mr. Prabhu Kavin[4]

PG Scholar, Department of CSE, Hindusthan Institute of Technology, Coimbatore, India
HOD/ PG-CSE, Department of CSE, Hindusthan Institute of Technology, Coimbatore, India
PG Scholar, Department of CSE, Hindusthan Institute of Technology, Coimbatore, India
PG Scholar, Department of CSE, Hindusthan Institute of Technology, Coimbatore, India

Abstract - Most of the business run today are online and security is of primary concern. But the online application are prone to security threats. Though security is available, for certain application. The security level is not satisfactory, and need was felt for providing a secure login service on data items like picture. The main goal of the log manager is to provide high bandwidth and low level inactivity. In many real world application secure and sensitive information must be kept in log files on an untreated machine. Hence a log manager is designed and implemented in this project to provide solution for the security issues involved in log management. In the event of attack, this secure data will be prone to attack. But the log manager designed in this project is very secure and the attacker will not gain any information about the secure data from log files and also protects from corrupting the log files. The following three steps are used by the attackers to hack: First the attacker can intercept any message sent over the internet. Second, the attacker can synthesize, replicate and replay data items in his possession. Finally, the attacker can be legitimate participant of the network or can try to impersonate legitimate hosts. Hence a log monitor is implemented for storing log files securely so that a secure read, write, delete , upload and download of the files is done.

Index Terms - Cloud computing, logging, privacy, security.

I. INTRODUCTION

Social network has become one of the most important parts of our daily life as it enables us to communicate with a lot of people. Since the creation of social networking sites such as myspace, linkedIn, facebook, orkut and twitter gives an individuals an opportunities not only to meet new people and friends but also allows resource sharing of data items like pictures between them.

Online relationships in social networks are often based on real world relationships and can therefore be used to infer a level of trust between users. The data items share between user and owner are store in a cloud which is a dynamic and enables users to store and share heterogeneous resources within the context of a social network. In spite above all, security is one of primary concern. The online applications are prone to security threats like data theft, hacking, mis-use and replicate of pictures etc. Though security is available, for certain application. The security level is not satisfactory, and need was felt for providing a secure login service on data items like picture in cloud and social network. The limitations of the existing system is that there is no security for user’s data , no authentication or security provided and high resource costs needed for the implementation.

A log manager is designed and implemented in this project to provide solution for the security issues involved in log management. The log manager designed in this project is very secure and the attacker will not gain any information about the secure data from log files and also protects from corrupting the log files.

II. LITERATURE SURVEY

In the literature a reliable delivery and filtering mechanism is adapted for Syslog feature that allows a device to be customized for receipt of syslog messages [21]. This feature provides reliable and secure delivery for syslog messages using Blocks Extensible Exchange Protocol (BEEP). Additionally, it allows multiple sessions to a single logging host, independent of the underlying transport method, and provides a filtering mechanism called a message discriminator. This module describes the functions of the Reliable Delivery and Filtering for Syslog feature and how to configure them in a network.

The National Institute of Standards and Technology, Karen Kent Murugiah Souppaya [18] recommended the Guide to Computer Security Log Management which provides practical, real-world guidance on developing, implementing, and maintaining effective log management practices throughout an enterprise. The guidance in this publication covers several topics, including establishing log management infrastructures, and developing and performing robust log management processes throughout an organization. The limitations of this project was that the publication presents log management technologies from a high-level viewpoint, and it is not a step-by-step guide to implementing or using log management technologies.

Sebastian Schmerl, et al, [28] computer networks and communication systems group , Brandenburg University of Technology, Cottbus, German describe about explorative visualization of log data to support forensic analysis and signature development .In this project, an approach for log resp. audit data representation is prpoposed, which aims at simplifying the analysis process for the security officer.

For this purpose audit data and existing relations between audit events are represented graphically in a three dimensional space. They describe a general approach for analyzing and exploring audit or log data in the context of this presentation paradigm.

Danny Dolev And Andrew C. Yao, member of IEEE [11] proposed a model On the Security of Public Key Protocols. The use of public key encryption to provide secure network communication has received considerable attention. Such public key systems are usually very effective against a “passive” eavesdropper, namely, one who merely taps the communication line and tries to decipher the intercepted message. However, as pointed out in Needham and Schroeder an improperly designed protocol could be vulnerable to an “active” saboteur, one who may impersonate another user and may alter or replay the message. As a protocol might be compromised in a complex way, informal arguments that assert the security for a protocol are prone to errors.

David L. Wells, et al, [35] proposed an architecture of an Open Object-Oriented Database (OODB) management system. An open, incrementally extensible object oriented database management system lets developers tailor database functionality for applications. It can also serve as a platform for research.

This article describes the architecture of the open OODB system. This computational model builds database functionality as an extensible collection of transparent extensions to existing programming languages. It also describe how open OODB's system architecture is decomposed into a kernel meta-architecture and a collection of modules implementing specific behavioural extensions. Finally, it discusses the risks of the approach and report on the project’s status.

Kjetil Norvag, et al, [23], Norwegian University of Science and Technology describe about concurrency control in distributed object-oriented
database system. This simulation results are presented with two different scheduler strategies. In this work the DBsim simulator includes extensions that could make it more suitable for simulation of algorithms for object-oriented databases. Obviously, much more can be done with both the simulation model and the simulator. This includes adding new schedulers to the system, e.g., other versions of the two-phase locking scheduler, like wound-wait and wait-die. In a real system, the method of replication is used for increased reliability and performance.

III. OVERVIEW

Social network enables us to communicate with a lot of people. It does not only allows us to meet new people and friends but also allows resource sharing of data items like pictures between them. So, securely maintaining records and files over extended periods of time is very important to the proper functioning of any organization as online applications are prone to security threats like data theft, hacking, mis-use and replicate of pictures etc. Integrity of the files and that of the logging process need to be ensured at all times. In addition, as files often contain sensitive information, confidentiality and privacy of log records are equally important.

However, deploying a secure logging infrastructure involves substantial capital expenses that many organizations may find overwhelming. Delegating log management to the cloud appears to be a viable cost saving measure. In this project, a secure cloud-based management service and a framework is being implemented.

In the existing system data handling in the cloud goes through a complex and dynamic hierarchical service chain. This does not exist in conventional environments. Ordinary web framework Uses web services for request and
responses. Ordinary storage in cloud are store without providing security to data items. Most data items in existing system are store in cloud and allow only
authorise user to access the data but they once the data item is miss-used it is not known to the owner. The limitations of the existing system are :
* No security for user’s data. No authentication or security provided
* High resource costs needed for the implementation.
* Not suitable for small and medium level storage users.

Hence, the need for a secure and trustworthy resource sharing mechanism in social networks and cloud computing is felt and this project is proposed.

The objective of the proposed system is to provide a secure framework for data sharing in cloud computing environment and a secure cloud-based log management service and a propose framework is to be implemented.

The Scope of the proposed system is to implement system for storing and maintaining log records in a server operating in a cloud-based environment .The project addresses security and integrity issues not only just during the log generation phase, but also during other stages in the log management process, including log collection, transmission, storage, and retrieval. In this system, a cryptographic protocols technique is also implemented to address integrity and confidentiality issues while storing, maintaining, and querying log records.

IV. SYSTEM ARCHITECTURE DESIGN

The system architecture is show below which includes the cloud user, the log monitor, the log client where for every log client there is different log generator and the database which store the data items. The proxy address of each client is store in the log file whenever they try to download or view the data items. The log monitor can checks who have download and view the data items. The uploading , deleting of data items is done by the data owner.

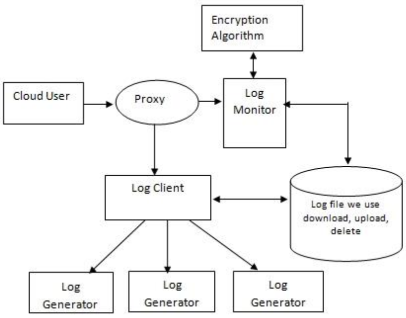

Fig 1 System architecture design

a) Jar File Organization

In the proposed system the pictures are store in the cloud storage as jar files having .jar extension .JAR file is the compressed file format. Many files can be store in a JAR file. JAR stands for the Java Archive. This file format is used to distribute a set of java classes. This file helps to reduce the file size and collect many file in one by compressing files. Downloading the files are become completed in very short duration of time because of reducing the file size. Jar file are make executable by collecting many class file of the java application in it. The jar file can execute from the java (Java Web Start).

The JAR file format is based on the popular ZIP file format. Usually these file format is not only used for archiving and distribution the files, these are also used for implementing various libraries, components and plug-ins in java applications. Compiler and JVMs (Java Virtual Machine) can understand and implement these formats for java application.

To perform basic operations for the jar file it has to used the Java Archive Tool (jar tool) which is provided by the jdk (Java Development Kit).

V. PROPOSED SYSTEM

In this project, a comprehensive solution for storing and maintaining log records in a server operating in a cloud-based environment is proposed. It addresses the security and integrity issues not only just during the log generation phase, but also during other stages in the log management process, including log collection, transmission, storage, and retrieval. The major contributions of this project are as follows the architecture of the system and development of cryptographic protocols to address integrity and confidentiality issues with storing, maintaining, and querying log records at the honest but curious cloud provider and in transit.

The advantages of the proposed system are :

* One of the main innovative features of the Cloud Information Accountability (CIA) framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication.

* Providing defences against man in middle attack, dictionary attack, disassembling Attack, Compromised JVM Attack

* It's Suitable for limited and large number of storages

The proposed system has the following application and can also be implemented on all social websites like

* Facebook
* Twitter
* Orkut
* Google++
* Online pictures seller of celebrities and pictures of rare and historic pictures.
* Finding criminal activities of misusing the download pictures.

VI. IMPLEMENTATION

a) Module Description

The modules of the proposed system are as follow-

* Log Generators
* Logging Client or Logging Relay
* Logging Cloud
* Log Monitor

b) Log Generators

These are the computing devices that generate log data. Each organization hat adopts the cloud-based log management service has a number of log generators. Each of these generators is up to with logging capability. The log files generated by these hosts are not stored locally except temporarily till such time as they are pushed to the logging client.

c)Logging Client Or Logging Relay

The logging client is a collector that receives groups of log records generated by one or more log generators, and prepares the log data so that it can be pushed to the cloud for long term storage. The log data is transferred from the generators to the client in batches, either on a schedule, or a sand when needed depending on the amount of log data waiting to be transferred. The logging client incorporates security protection on batches of accumulated log data and pushes each batch to the logging cloud. When the logging client pushes log data to the cloud it acts as a logging relay. We use the terms logging client and logging relay interchangeably. The logging client or relay can be implemented as a group of collaborating hosts. For simplicity however, we assume that there is a single logging client.

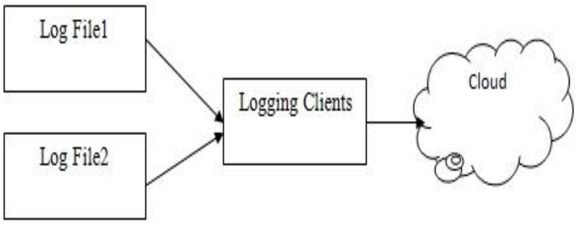

Fig 2 Logging client

d) Logging Cloud

The logging cloud provides long term storage and maintenance service to log data received from different logging clients belonging to different organizations. The logging cloud is maintained by a cloud service provider. Only those organizations that have subscribed to the logging cloud's services can upload data to the cloud.

The cloud, on request from an organization can also delete log data and perform log rotation. Before the logging cloud will delete or rotate log data it needs a proof from the requester that the latter is authorized to make such a request. The logging client generates such a proof. However, the proof can be given by the logging client to any entity that it wants to authorize.

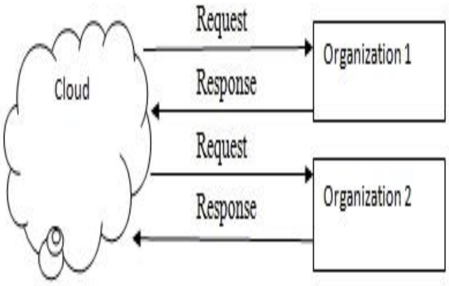

Fig 3 Logging Cloud

e) Log Monitor

These are hosts that are used to monitor and review log data. They can generate queries to retrieve log data from the cloud. Based on the log data retrieved, these monitors will perform further analysis as needed. They can also ask the log cloud to delete log data permanently, or rotate logs.

f) Pushing And Pulling Strategies

Pushing or pulling strategies have interesting tradeoffs. The pushing strategy is beneficial when there are a large number of accesses to the data within a short period of time. In this case, if the data are not pushed out frequently enough, the log file may become very large, which may increase cost of operations like copying data. The pushing mode may be preferred by data owners who are organizations and need to keep track of the data usage consistently over time. For such data owners, receiving the logs automatically can lighten the load of the data analyzers. The maximum size at which logs are pushed out is a parameter which can be easily configured while creating the logger component. The pull strategy is most needed when the data owner suspects some misuse of his data; the pull mode allows him to monitor the usage of his content immediately. A hybrid strategy cans actually be implemented to benefit of the consistent information offered by pushing mode and the convenience of the pull mode.

g) Logging Mechanism

* The Logger Structure
* Log Record Generation
* Dependability of Logs
* JARs Availability
* Log Correctness

The Logger Structure

A leverage programmable capability of JARs to conducted for automated logging. A logger component is a nested Java JAR file which stores a user’s data items and corresponding log files.

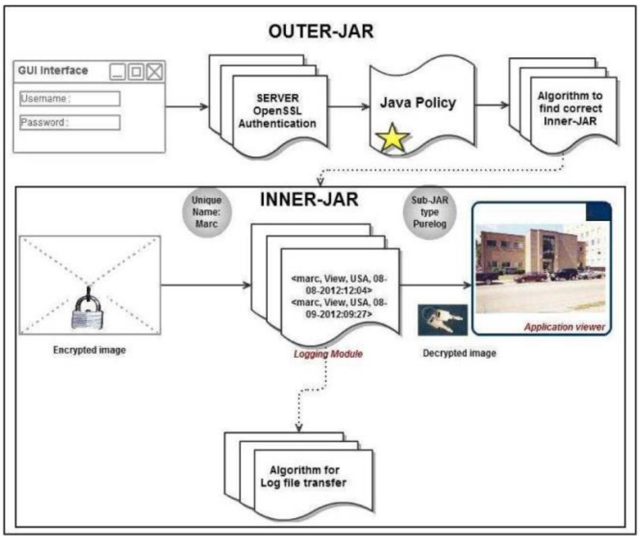
OUTER-JAR
GUI Interface
Username:
Password:
SERVER OpenSSL Authentication
Java Policy
Algorithm to find correct Inner-JAR
INNER-JAR
Unique Name: Marc
Sub-JAR type Purelog
<marc, View, USA, 08-08-2012:12:04>
<marc, View, USA, 08-09-2012:09:27>
Encrypted image
Logging Module
Decrypted image
Application viewer
Algorithm for Log file transfer

Fig 4 JAR implementation

The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file. In our context, the data owners may not know the exact Context Service Provider (CSPs) that are going to handle the data. Hence, authentication is specified according to the servers' functionality (which we assume to be known through a lookup service), rather than the server's URL or identity. For example, a policy may state that Server X is allowed to download the data if it is a storage server. The outer JAR may also have the access control functionality to enforce the data owner's requirements, specified as Java policies, on the usage of the data.

A Java policy specifies which permissions are available for a particular piece of code in a Java application environment. The permissions expressed in the Java policy are in terms of File System Permissions. However, the data owner can specify the permissions in user-centric terms as opposed to the usual code-centric security offered by Java, using Java Authentication and Authorization Services. Moreover, the outer JAR is also in charge of selecting the correct inner JAR according to the identity of the entity who requests the data

Each inner JAR contains the encrypted data, class files to facilitate retrieval of log files and display enclosed data in a suitable format, and a log file for each encrypted item.

It support two options:

* Pure Log- its main task is to record every access to the data. The log files are used for pure auditing purpose.

* Access Log - it has two functions: logging actions and enforcing access control. In case an access request is denied, the JAR will record the time when the request is made. If the access request is granted, the JAR will additionally record the access information along with the duration for which the access is allowed.

The two kinds of logging modules allow the data owner to enforce certain access conditions either proactively (in case of Access Logs) or reactively (in case of Pure Logs).

To carry out these functions, the inner JAR contains a class file for writing the log records, another class file which corresponds with the log harmonizer, the encrypted data, a third class file for displaying or downloading the data (based on whether we have a Pure Log, or an Access Log), and the public key of the IBE key pair that is necessary for encrypting the log records.

No secret keys are ever stored in the system. The outer JAR may contain one or more inner JARs, in addition to a class file for authenticating the servers or the users, another class file for finding the correct inner JAR, a third class file which checks the JVM's validity using oblivious hashing. Further, a class file is used for managing the GUI for user authentication and the Java Policy.

Log Record Generation

Log records are generated by the logger component. Logging occurs at any access to the data in the JAR, and new log entries are appended sequentially, in order of creation LR=(r1; . . . ; rki). Each record ri is encrypted individually and appended to the log file. To ensure the correctness of the log records, we verify the access time, locations as well as actions. In particular, the time of access is determined using the Network Time Protocol (NTP) to avoid suppression of the correct time by a malicious entity. The location of the cloud service provider can be determined using IP address. The JAR can perform an IP lookup and use the range of the IP address to find the most probable location of the CSP. More advanced techniques for determining location can also be used . Similarly, if a trusted time stamp management infrastructure can be set up or leveraged, it can be used to record the time stamp in the accountability log . The most critical part is to log the actions on the users' data.

In the current system, it support four types of actions, i.e., Act has one of the following four values: view, download, timed access, and Location-based access. For each action, we propose a specific method to correctly record or enforce it depending on the type of the logging module, which are elaborated as follows:

* *View*

The entity (e.g., the cloud service provider) can only read the data but is not allowed to save a raw copy of it anywhere permanently. For this type of action, the Pure Log will simply write a log record about the access, while the Access Logs will enforce the action through the enclosed access control module. Recall that the data are encrypted and stored in the inner JAR.

When there is a view-only access request, the inner JAR will decrypt the data on the fly and create a temporary decrypted file. The decrypted file will then be displayed to the entity using the Java application viewer in case the file is displayed to a human user. Presenting the data in the Java application, viewer disables the copying functions using right click or other hot keys such as Print Screen. Further, to prevent the use of some screen capture software, the data will be hidden whenever the application viewer screen is out of focus. The content is displayed using the headless mode in Java on the command line when it is presented to a CSP.

* *Download*

The entity is allowed to save a raw copy of the data and the entity will have no control over this copy neither log records regarding access to the copy. If Pure Log is adopted, the user's data will be directly downloadable in a pure form using a link. When an entity clicks this download link, the JAR file associated with the data will decrypt the data and give it to the entity in raw form.

In case of Access Logs, the entire JAR file will be given to the entity. If the entity is a human user, he/she just needs to double click the JAR file to obtain the data. If the entity is a CSP, it can run a simple script to execute the JAR.

* *Timed access*

This action is combined with the view-only access, and it indicates that the data are made available only for a certain period of time. The Pure log will just record the access starting time and its duration, while the Access Log will enforce that the access is allowed only within the specified period of time. The duration for which the access is allowed is calculated using the Network Time Protocol. To enforce the limit on the duration, the Access Log records the start time using the NTP, and then uses a timer to stop the access. Naturally, this type of access can be enforced only when it is combined with the View access right and not when it is combined with the Download.

* *Location-based access*

In this case, the Pure Log will record the location of the entities. The access log will verify the location for each of such access. The access is granted and the data are made available only to entities located at locations specified by the data owner.

Dependability of Logs

First, an attacker may try to evade the auditing mechanism by storing the JARs remotely, corrupting the JAR, or trying to prevent them from communicating with the user. Second, the attacker may try to compromise the JRE used to run the JAR files.

JAR Availability

To protect against attacks perpetrated on offline JARs, the CIA includes a log harmonizer which has two main responsibilities: to deal with copies of JARs and to recover corrupted logs.

Each log harmonizer is in charge of copies of logger components containing the same set of data items. The harmonizer is implemented as a JAR file. It does not contain the user’s data items being audited, but consists of class files for both a server and a client processes to allow it to communicate with its logger components. The harmonizer stores error correction information sent from its logger components, as well as the user’s IBE decryption key, to decrypt the log records and handle any duplicate records. Duplicate records result from copies of the user’s data JARs. Since user’s data are strongly coupled with the logger component in a data JAR file, the logger will be copied together with the user’s data. Consequently, the new copy of the logger contains the old log records with respect to the usage of data in the original data JAR file. Such old log records are redundant and irrelevant to the new copy of the data. To present the data owner an integrated view, the harmonizer will merge log records from all copies of the data JARs by eliminating redundancy.

Log correctness

For the logs to be correctly recorded, it is essential that the JRE of the system on which the logger components are running remain unmodified. To verify the integrity of the logger component, it is rely on a two-step process:

* The JRE is repair before the logger is launched and any kind of access is given, so as to provide guarantees of integrity of the JRE.

* Hash codes is insert , which calculate the hash values of the program traces of the modules being executed by the logger component. This
helps in detecting the modifications of the JRE once the logger component has been launched, and are useful to verify if the original code flow of execution is altered.

VII . EXPERIMENTAL RESULTS

The proposed system allows the user to log in to any of the social networking websites and provide a secure access to data with an auditing mechanism for the user as well as the owner.

It allows the data owner to not only audit his content but also enforce strong back-end protection on demand.

Moreover, one of the main features of this project is that it enables the data owner to audit even those copies of its data that were made without his knowledge by decrypting the images and gain the user-name and ip-address of the third party who miss-used the data item.

A log file is maintained by the data owner and he/she can view who has downloaded and viewed his/her data items. The log file also gives the date, time and the location he/she has downloaded.

VII. CONCLUSION

The proposed system allows the user to log in to any of the social networking websites and provide a secure access to data with an auditing mechanism for the user as well as the owner. The system proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. The system allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Moreover, one of the main features of the proposed is that it enables the data owner to audit even those copies of its data that were made without his knowledge.

VIII. FUTURE WORK

In the future it is expected to implement a class file for authenticating the servers or the users, another class file finding the correct inner JAR and a class file which checks the JVM's validity using oblivious hashing.

It is also expected to perform in future the method of watermarking on pictures and the mechanism of stenography to encrypt the ip address, location, username, time in the downloaded picture and decrypt the picture to get back the user name and ip- address of the user if it is found misused.

The idea of the proposed system can be implemented in a cloud and protect social website and any other website in future from any data theft and provide security to the data items in the cloud storage.

REFERENCES

[1] Amman.P and Jajodia.S, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 2012.

[2] Ateniese.G, Burns.R , Curtmola.R Herring.J, Kissner.L Peterson.Z, and Song.D, "Provable Data Possession at Un trusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007.

[3] Barka. E and Lakas.A, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm., vol.2008, pp.1-8, 2013.

[4] Boneh.D and Franklin.M.K, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213- 229, 2010.

[5] Bose.R and Frew.F, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1- 28, Mar. 2012.

[6] Buneman.P, Chapman.A, and Cheney.J,"Provenance Management in Curate Databases," Proc. ACM IGMOD Int'l Conf. Management of Data(SIGMOD '06), pp. 539-550, 2013

[7] Bellare.M and Yee B.S., "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

[9] Blakley.G.R,"Safeguarding cryptographic keys," in Proc. Nat. Comput. Conf., Jun. 1979, p. 313.

[10] Dingledine.R, Mathewson.N, and Syverson.P, "Tor: The second Generation onion router," in Proc. 12th Ann. USENIX Security Symp., Aug. 2004, pp. 21–21.

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

[12] Droms.R, Dynamic Host Configuration Protocol, Request for Comment RFC 2131, Internet Engineering Task Force, Network Working Group Mar. 1991.

[13] Eckert.C and Pircher.A, "Internet anonymity: Problems and solutions,"in Proc. 16th IFIP TC-11 Int. Conf. Inform. Security, 2001, pp. 35–50 .

[14] Flegel.U, "Pseudonymizing unix log file," in Proc. Int. Conf. Infrastruture Security, LNCS 2437. Oct. 2002, pp. 162–179.

[15] Global Internet Freedom Consortium. (2012, Mar.).

[16] Holt.J.E, "Logcrypt: Forward security and public verification for secure audit logs," in Proc. 4th Australasian Inform. Security Workshop, 2006,pp. 203–211.

[17] Herzberg.A,Jarecki.S, Krawczyk.H, and Yung.M, "Proactive secret sharing or: How to cope with perpetual leakage," in Proc. 15th Ann. Int.Cryptology

Conf., Aug. 1995, pp. 339–352.

[18] Kent.K and Souppaya. M (1992). Guide to Computer Security Log Management, NIST Special Publication 800-92 .

[19] Kelsey.J, Callas.J, and Clemm.A, Signed Syslog Messages, Request for Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.

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

[21] Ma.D and Tsudik.G, “A new approach to secure logging,” ACM Trans.Storage, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.

[22] New.D and Rose.M, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.

[23] Norvag.K, Sandsta.O, and Bratbergsengen.K, “Concurrency control In distributed object oriented database systems,” in Proc. 1st East-Eur. Symp. Adv. Databases Inform. Syst., Sep. 1997, pp. 32–32.

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

[25] PCI Security Standards Council. (2006, Sep.) Payment Card Industry (PCI) Data Security Standard—Security Audit Procedures Version 1.1

[26] Project:AN.ON—Anonymity Online. (2012, Mar.). JAP Anonymity and Privacy

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

[28] Sebastian Schmerl, Michael Vogel, René Rietz, and Hartmut König“Explorative Visualization of Log Data”,2010 Fifth International Workshop.

[29] Schneier.B and Kelsey.J, “Security audit logs to support computer forensics,” ACM Trans. Inform. Syst. Security, vol. 2, no. 2, pp. 159–176, May 1999.

[30] Shamir.A, “How to share a secret,” Communication. ACM, vol. 22, no.11, pp. 612–613, Nov. 1979.

[31] The Tor Project, Inc. (2011, Sep.) Tor: Anonymity Online

[32] Teranishi.I, Furukawa.J, and Sako.K, “k-times anonymous Authentication (extended abstract),” in Proc. 10th Int. Conf. Theor. Appl Cryptology Information Security, LNCS 3329. 2004, pp. 308–322.

[33] U.S. Department of Health and Human Services. (2011, Sep.). HIPAA—General Information .

[34] Ultrareach Internet Corporation. (2012, Mar.). Ultrasurf—Privacy, Security, Freedom

[35] Wells.D.L, Blakeley.J.A, and Thompson.C.W, “Architecture of an open object-oriented database management system,” IEEE Comput., vol. 25, no. 10, pp. 74–82, Oct. 1992.