Asian Journal of Information Technology 13 (9): 446-453, 2014
ISSN: 1682-3915

Functioning Aspects of Secure Network and Design of Log Analyser

S. Manikandan, R. Prabhu, D. Sureshbabu and P. Vijayanand
Department of Information Technology,
Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College,
Anna University, Chennai, India

Abstract: Geographically distributed and heterogeneous networks like internet with millions of entry points, users and transmission paths, depict a typical insecure network. The points where an organization's private network interfaces with the internet are vulnerable to potential security attacks. Almost all organizations take care to protect their networks through firewalls, proxy servers and the like but recently even these so called secure networks are being subjected to numerous intrusions. The logs generated by the firewalls are so huge that it is even more difficult to find the source of attack in the event of a security breach. In this study, researchers present a progressive approach to strengthen the security of a private network beginning from the server level till the gateway interface to the internet. Routers in front of the firewall can be configured using context based access control in order to perform an initial security screening. Further, the design of a log analyser is discussed which details the process of collecting and analysing logs from firewalls and application gateways, in an efficient and less time consuming manner.

Key words: Proxy server, application gateway, OS hardening, context based access control, log analyser, firewall, router

INTRODUCTION

Security is a concern of organizations with assets that are controlled by computer systems. The requirements of information security within an organization have undergone two major changes in the last several decades. The generic name for the collection of tools designed to protect data and to thwart hackers is computer security (Lysyanskaya *et al*., 2004). The second major change that affected security is the introduction of distributed systems and the use of network and communication facilities for carrying data between terminal users and computer and between computer and computer. Network security measures are needed to protect data during transmission. Virtually all business, government and academic organizations interconnect their data processing equipment with a collection of interconnected networks referred to as internet and so the term internet security may be used instead of network security. It is always important to analyse the security risks that may occur in a network before moving forward to implement security measures. Once the risks or the network vulnerabilities are studied, proper measures should be taken in order to prevent those risks (Schnorr, 1991). It is not possible to create a risk free network. So, necessary methods need to be devised to recover from any disaster caused by a security breach. Different type of security measures are being implemented by organizations but even in a so called secure network there will be hidden points of vulnerabilities through which an intrusion or security attack can happen (Nagaratnam and Lea, 1998). So, there always lies room for improvement of security. The objective of this proposal is to study the existing security measures in an organization network, analyze it to find the potential flaws and take productive steps to reconfigure and improve the system with a view to creating a more secure network without degrading system performance.

Threats to a secure network: A significant security problem for networked systems is hostile or at least unwanted trespass by users or software. Natural disasters, viruses, power faults can damage a network's data, programs and hardware. A security breach can however harm a network just as easily and quickly. User trespass can take the form of unauthorized logon to a machine or in the case of an authorized user, acquisition of privileges or performance of actions, beyond those that have been authorized. Software trespass can take the form of a virus, worm or Trojan horse. Basically security attacks are classified as active and passive attacks. Passive attacks include unauthorized reading of a

Corresponding Author: S. Manikandan, Department of Information Technology,
Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Anna University, Chennai, India

message of file and traffic analysis whereas active attacks cause more disastrous effects including denial of service, modification of messages, masquerading and replay (Bommepally *et al*., 2010; Rivest *et al*., 1978). All these attacks relate to network security because system entry can be achieved by means of the network. Over time the attacks on the internet and Internet attached systems have grown more sophisticated while the amount of skill and knowledge required to mount an attack has declined.

Attacks have become more automated and can cause greater amounts of damage. The objective of the intruder is to gain access to the system or to increase the range of privileges accessible on a system (Boneh *et al*., 2003a, b, 2001). Generally, this require an intruder to acquire information that should have been protected. In some cases, the information is in the form of user password.

With the knowledge of user password the intruder can login to the system and do actions just like a legitimate user. Certain intrusion detection schemes are more serious in that they affect an organization's network as a whole. Organization network will be connected to the internet. The entry points where the organizations private network interfaces with the internet introduces vulnerabilities. An intruder can gain access easily by disguising itself as a legitimate user within the network. This can be done by falsifying the IP address, port name, etc.

Security aspects of intrusion prevention: The most common way to protect a network from intrusion is through the implementation of firewall (Boneh and Franklin, 2001). Firewall forms a barrier through which traffic going in each directions of a network must pass. A firewall may be designed to operate as a filter at the level of IP packets or may operate at a higher protocol (Shehab *et al*., 2005). The firewall should be immune to penetration. Firewalls are basically classified into two types.

The packet filtering router applies a set of rules to each incoming and outgoing packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions. Filtering rules will be based on the information contained in a network packet like source IP address, destination IP address, source and destination transport level address, IP protocol field and Interface.

The second type of firewall the application level gateway, also called a proxy server, acts as a relay of application traffic (Boneh and Franklin, 2001; Boneh *et al*., 2003a, b). The user contacts the gateway using a TCP/IP application and the gateway asks for the username and password to login to the remote host. Although, a proxy server appears to the outside world as an internal network server, in reality it is merely another filtering device for the internal LAN. The message originating from the network will first go to the proxy server where the data frames are repackaged and the proxy inserts its own IP address as the source address.

This repackaged data frame is then passed to a packet filtering router where it validates the source address and forwards it to the destination outside the organization's network. Thus, the actual identity of the sender will remain hidden to the rest of the users outside the private network. This prevents any intruder form taking the identity of a valid user and intruding into the network (Boneh *et al*., 2003a, b).

In the present organization network, appropriately configured firewalls are placed at all places of interface with the internet (Rivest *et al*., 1978). Firewalls keep a detailed record of every packet that hit its terminal. This log report will be used to further analyses to find out attacks from prohibited sources and the frequency of those attacks.

The contributions: Existing security models assume that the initial screening of traffic occurs at the firewall level. But during times of heavy traffic the filtering process at the firewall causes the buffers in the firewall to be overloaded and subsequent packet loss results (Neuman, 1993). This reduces the speed of internet connection as well. Another issue is related to the logs collected by the firewall. These logs include the details of all sources who attempted to login to the network. Within a small interval of time these logs will grow enormously in size. So, it becomes extremely difficult for the network administrator to analyze the logs to detect an anomaly. Thus, the periodic manual log analysis becomes a tedious and time consuming task. To overcome these drawbacks in the existing system we propose certain enhancements. The proposed system as follows:

Hereby present a progressive approach starting from the server level. For the purpose of study we considered the linux servers (Naor and Nissim, 1998). These servers are additionally shielded through an OS Hardening process in an Advanced Intrusion Detection Environment (AIDE). In this way, researchers limit the options available to any user that handles the server.

Thus, the possibility of an unauthorized attack is prevented. Here by a detailed specification for configuring routers. The routers situated at the entry point of private network in front of the packet filtering firewall are subjected to configuration (Ferraiolo *et al*., 2001). Apart from the traditional method of configuration involving

Access Control List (ACL) and Extended ACL, Context Based Access Control (CBAC) is also used. Thus, protocol inspection can be performed at the router level. As an initial filtering is done at the router level, the burden at the firewall is reduced. This improves the speed and performance of the system and risk of packet loss is reduced.

At this time, it present the design aspects of a Log Analyser. The logs from firewall are collected using advanced features of syslog-ng. Thus, appropriate filters can be added to get the required logs alone along with the time stamp. The logs collected in this way are analyzed using the specialized application called Snort. Snort is configured in the Network Intrusion Detection (NIDS) mode to perform analysis.

OS HARDENING

This study includes the process of implementing OS hardening in the Linux servers as a prime step of security set up. The security tools are configured to improve system performance.

Installing and maintaining software: This includes a clean installation where any previous installations are wiped out (Gentry, 2003). Some system directories should be placed on their own partitions (or logical volumes).

This allows for better separation and protection of data. Uncheck all package groups including the package groups "Software Development" and "Web Server" unless there is a specific requirement to install software using the system installer. If the machine will be used as a web server, it is preferable to manually install the necessary RPMs instead of installing the full "Web Server" package group. Then, update the software packages using yum. This is followed by Automatic Update Retrieval and Installation with Cron (Gentry, 2003).

Packet signature checking should be globally activated. The next step involves the creation of an Advanced Intrusion Detection Environment which is an integral part of OS Hardening. This involves configuring the Advanced Intrusion Detection Environment (AIDE). AIDE is not installed by default. Install it with the command:

```
# yum install aide
Generate a new database:
# /usr/sbin/aide --init
By default, the database will be written to the file /var/lib/aide/aide.db.new.gz.n The database as well as the configuration file /etc/aide.conf and the binary /usr/sbin/aide (or hashes of these files) should be copied and stored in a secure location. Storing these copies or hashes on read-only media may provide further
confidence that they will not be altered.Install the newly-generated database.
# cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
Run a manual check:
# /usr/sbin/aide-check.
```

Manually verify the integrity of the AIDE binaries, configuration file and database. Possibilities for doing so include: use sha1sum or md5sum to generate checksums on the files and then visually compare them to those generated from the safely stored versions. This does not, of course, preclude the possibility that such output could also be faked). Mount the stored versions on read-only media and run/bin/diff to verify that there are no differences between the files3).

Copying the files to another system and performing the hash or file comparisons there may impart additional confidence that the manual verification process is not being interfered with. Traditional Unix security relies heavily on file and directory permissions to prevent unauthorized users from reading or modifying files to which they should not have access. Adhere to the principle of least privilege configure each file, directory and file system to allow only the access needed in order for that file to serve its purpose.

However, Linux systems contain a large number of files so, it is often prohibitively time-consuming to ensure that every file on a machine has exactly the permissions needed. This study introduces several permission restrictions which are almost always appropriate for system security and which are easy to test and correct.

File systems mounted on removable media also provide a way for malicious executables to potentially enter the system and should be mounted with options which grant least privilege.

Users should not be allowed to introduce arbitrary devices or setuid programs to a system. In addition while users are usually allowed to add executable programs to a system, the noexec option prevents code from being executed directly from the media itself and may therefore provide a line of defence against certain types of worms or malicious code. The default system configuration grants the console user enhanced privileges normally reserved for the root user including temporary ownership of most system devices. If not necessary, these privileges should be removed and restricted to root only. USB flash or hard drives allow an attacker with physical access to a system to quickly copy an enormous amount of data from it. If USB storage devices should not be used, the odprobe program used for automatic kernel module loading should be configured to not load the USB storage driver upon demand. Another means of disabling USB storage is to disable all USB support provided by the operating system. This can be accomplished by adding the "nousb" argument to the kernel's boot loader configuration. An attacker with physical access could try to boot the system from a USB flash drive and then

access any data on the system's hard drive, circumventing the normal operating system's access controls. To prevent this, configure the BIOS to disallow booting from USB drives.

Permissions for many files on a system should be set to conform to system policy. This study discusses important permission restrictions gshadow which should be checked on a regular basis to ensure that no harmful discrepancies have arisen. When the so-called "sticky bit" is set on a directory, only the owner of a given file may remove that file from the directory. Without the sticky bit, any user with write access to a directory may remove any file in the directory. Setting the sticky bit prevents users from removing each other's files. In cases where there is no reason for a directory to be world-writable, a better solution is to remove that permission rather than to set the sticky bit. However, if a directory is used by a particular application, consult that application's documentation instead of blindly changing modes.

Data in world-writable files can be modified by any user on the system. In almost all circumstances, files can be configured using a combination of user and group permissions to support whatever legitimate access is needed without the risk caused by world-writable files. A core dump file is the memory image of an executable program when it was terminated by the operating system due to errant behavior. In most cases, only software developers would legitimately need to access these files.

The core dump files may also contain sensitive information, or unnecessarily occupy large amounts of disk space. By default, the system sets a soft limit to stop the creation of core dump files for all users. This is accomplishedin/etc/profile with the line: ulimit-S-c0> /dev/null 2>and 1.

PAM or Pluggable Authentication Modules is a system which implements modular authentication for Linux programs (Sollins, 1988; Gunther, 1990). PAM is the framework which provides the system's authentication architecture and can be configured to minimize the system's exposure to unnecessary risk. A centralized authentication service is any method of maintaining central control over account and authentication data and of keeping this data synchronized between machines (Sollins, 1988). Such services can range in complexity from a script which pushes centrally-generated password files out to all machines, to a managed scheme such as LDAP or Kerberos.

Services: Running obsolete services should prioritize switching to more secure services which providethe needed functionality. If it is absolutely necessary to run one of these services for legacy reasons, care should be taken to restrict the service as much as possible, for instance by configuring host firewall software. The anacron subsystem is designed to provide cron functionality for machines which may be shut down during the normal times that system cron jobs run, frequently in the middle of the night. Laptops and workstations which are shut down at night should keep anacron enabled, so that standard system cron jobs will run when the machine boots. However, on machines which do not need this additional functionality, anacron represents another piece ofprivileged software which could contain vulnerabilities.

Therefore, it should be removed when possible to reduce system risk. The Avahi daemon implements the DNS service discovery and multicast DNS protocols which provide service and host discovery on a network.

It allows a system to automatically identify resources on the network, such as printers or web servers. This capability is also known as mDNSresponder and is a major part of Zeroconf networking. By default it is enabled. If a Mail Transfer Agent (MTA) is needed, the system default is Sendmail. Postfix which was written with security in mind is also available and is preferredIt is highly recommended that the BIND9 software be used to provide DNS service (Falcone and Castelfranchi, 2001). BIND is the internet standard Unix nameserver and while it has had security problems in the pas it is also well-maintained. If it is necessary to run the snmpd agent on the system, some best practices should be followed to minimize the security risk from the installation.

Router configuration using CBAC

ACL concepts: Masks are used with IP addresses in IP ACLs to specify what should be permitted and denied. Masks in order to configure IP addresses on interfaces start with 255 and have the large values on the left side for example, IP address 209.165.202.129 with a 255.255.255.224 mask. Masks for IP ACLs are thereverse for example, mask 0.0.0.255. This is sometimes called an inverse mask or a wildcard mask. When the value of the mask is broken down into binary (0s and 1s), the results determine which address bits are to be considered in processing the traffic. A 0 indicates that the address bits must be considered (exact match); a 1 in the mask is a "don't care". Table 1 further explains the concept.

Table 1: An example of ipacl mask

Cofiguration	Mask example
Network address	10.1.1.0
(traffic that is to be processed)	
Mask	0.0.0.255
Network address (binary)	00001010.00000001.00000001.00000000
Mask (binary)	00000000.00000000.00000000.11111111

Based on the binary mask, the first three sets (octets) must match the given binary network address exactly (00001010.00000001.00000001). The last set of numbers is "don't cares" (.11111111). Therefore, all traffic that begins with 10.1.1 matches since the last octet is "don't care".

Therefore, with this mask, network addresses 10.1.1.1 through 10.1.1.255 (10.1.1.x) are processed. Traffic that comes into the router is compared to ACL entries based on the order that the entries occur in the router. New statements are added to the end of the list. For example, access-list 101 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 permits all IP traffic from the network 10.1.1.0. In addition to defining ACL source and destination it is possible to define ports, ICMP message types and other parameters The router can display descriptive text on some of the well-known ports. access-list 102 permit tcp host 10.1.1.1 host 172.16.1.1 Border Gateway Protocol (179) chargen Character generator (19) cmdRemote commands (rcmd, 514) During configuration, the router also converts numeric values to more user-friendly values.

This is an example where you type the ICMP message type number and it causes the router to convert the number to a name. access-list 102 permit icmp host 10.1.1.1 host 172.16.1.1 1 becomes access-list 102 permit icmp host 10.1.1.1 host 172.16.1.1 timestamp-reply.

ACLs have no effect until they are applied to the interface of the router. It is a good practice to apply the ACL on the interface closest to the source of the traffic.

Context based access control: Context-Based Access Control (CBAC) was introduced in Cisco IOS Software Release 12.0.5.T and requires the Cisco OS Firewall feature set. CBAC inspects traffic that travels through the firewall in order to discover and manage state information for TCP and UDP sessions. This state information is used in order to create temporary openings in the accesslists of the firewall. Configure ip inspect lists in the direction of the flow of traffic initiation in order to allow return traffic and additional data connections for permissible session, sessions that originated from within the protected internal network in order to do this. For traffic inspection each protocol is tied to a keyword name. The CBAC within IOS supports (Table 2).

Table 2: Defines keywords for protocols

Mask example	Descriptions
cuseeme	CU See Me protocol
ftp	File Transfer protocol
h323	H.323 protocol (for example
http	HTTP protocol
rcmd	R commands (r-exec, r-login, r-sh)
realaudio	Real audio protocol
rpc	Remote procedure call protocol
smtp	Simple mail transfer protocol
streamworks	Stream works protocol
tcp	Transmission control protocol
tftp	TFTP protocol
udp	User datagram protocol
vdolive	VDOLive protocol

An example configuration that inspects FTP, SMTP and Telnet is given. The same logic is used to inspect other protocols like ICMP, NTP, etc.

```
router1#configure
Configuring from terminal, memory or network [terminal]? Enter configuration
commands, one per line. End with CNTL/Z.
router1(config)#ip inspect name mysite ftp
router1(config)#ip inspect name mysitesmtp
router1(config)#ip inspect name mysitetcp
router1#show ip inspect config
Session audit trail is disabled
1 min (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incompletetcp connections per host is 50.
Block-time 0 min.
tcpsynwait-time is 30 sec -- tcpfinwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
Inspection name mysite
ftp timeout 3600
smtp timeout 3600
tcp timeout 3600
```

Thus the process of CBAC includes: Apply the configuration; enter the access lists; configure the inspection statements and apply the access lists to the interfaces.

Collection of logs from firewall: In this study, researchers present an advanced version of syslog application known as syslog-ng to collect logs from firewalls and application gateways (Alam and Smith, 2001; Nehinbe, 2010). The syslog-ng application is a flexible and highly scalable system logging application that is ideal for creating centralized and trusted logging solutions. Typically, syslog-ng is used to manage log messages and implement centralized logging where the aim is to collect the log messages of several devices on a single, central log server. The different devices-called

syslog-ng clients-all run syslog-ng and collect the log messages from the various applications, files and other sources. The clients send all important log messages to the remote syslog-ng server where the server sorts and stores them.

Configuring syslog-ng: The syslog-ng application is configured by editing the syslog-ng.conf file. Use any regular text editor application to modify the file. The precompiled syslog-ng packages include sample configuration files as well (Parasuraman *et al*., 2005).

Global objects (e.g., sources, destinations, log paths, or filters) are defined in the syslog-ng configuration file. Object definitions consist of; type of the object: one of source, destination, log, filter, parser, rewrite rule or template. Identifier of the object: a unique name identifying the object. When using a reserved word as an identifier, enclose the identifier in quotation marks; parameters: the parameters of the object, enclosed in braces {parameters}. Semicolon: object definitions end with a semicolon (;). The syntax is summarized as< type identifier {parameters}; >Filters perform log routing within syslog-ng: a message passes the filter if the filter expression is true for the particular message. If a log statement includes filters, the messages are sent to the destinations only if they pass all filters of the log path. For example, a filter can select only the messages originating from a particular host. Complex filters can be created using filter functions and logical booleanexpressions. A log statement or log path is used to define the relationship between the source, filter and destination. Sources, destinations and filters should be declared before defining log path. An example of a configuration file is shown:

```
sources_localhost {
tcp(ip(127.0.0.1) port(1999) max-connections(300)); };
sources_tcp {
tcp(ip(192.168.1.5) port(1999) max-connections(100)); };
destinationd_tcp {
tcp("10.1.2.3" port(1999); localport(999)); log_fifo_size(40000); };
filter demo_filter1 { host("example1"); };
logdemo_filteredlog {
source(s_localhost); source(s_tcp);
filter(demo_filter1); destination(d_tcp);};
```

Researchers fetched the collected logs into the destination files and also into a mysql database by using sql() driver.

Design of log analyzer and results: The logs collected from the firewalls are systematically stored in files. The required rules conforming to the security requirements of the organization are written into a configuration file. Each log file is matched against the set of rules and whenever a rule violation is encountered the event is reported. This analyser takes the log file (Alam and Smith, 2001) as the input and output network usage statistics in the form of various reports. The log analyser the designed using the open source language called SNORT. Snort can be configured to operate in three different modes.

Sniffer mode: This simply reads the packets off of the network and displays them for you in a continuous stream on the console (screen).

Packet logger mode: This logs the packets into a disk which can be referred as required thus offering more flexibility over sniffer mode.

NIDS mode: Network Intrusion Detection or NIDS mode is the most complex and configurable configuration which allows snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

For the study, researchers have configured snort to work in NIDS mode. This can be executed from command line. There are a number of ways to configure the output of snort in NIDS mode. The default logging and alerting mechanisms are to log in decoded ASCII format and use full alerts. The full alert mechanism prints out the alert message in addition to the full packet headers. There are several other alert output modes available at the command line as well as two logging facilities. It provides support for real time statistics and can process as many as 1,50,000 lines of log entries per second.

Experimental evaluation: In this study, researchers present the results of using Log Analyser. It generates legible reports of usage of the network in the form of graphs as shown in Fig. 1.

It gives a detailed view about the unique users in a particular hour of the day, how long they logged into a particular application in the network, etc. Thus if the user tried to access any unauthorized application it will be immediately reported. Since, the result comes in a graphical format it can be easily understood. Details about the graph can be read from the related files of a particular user.

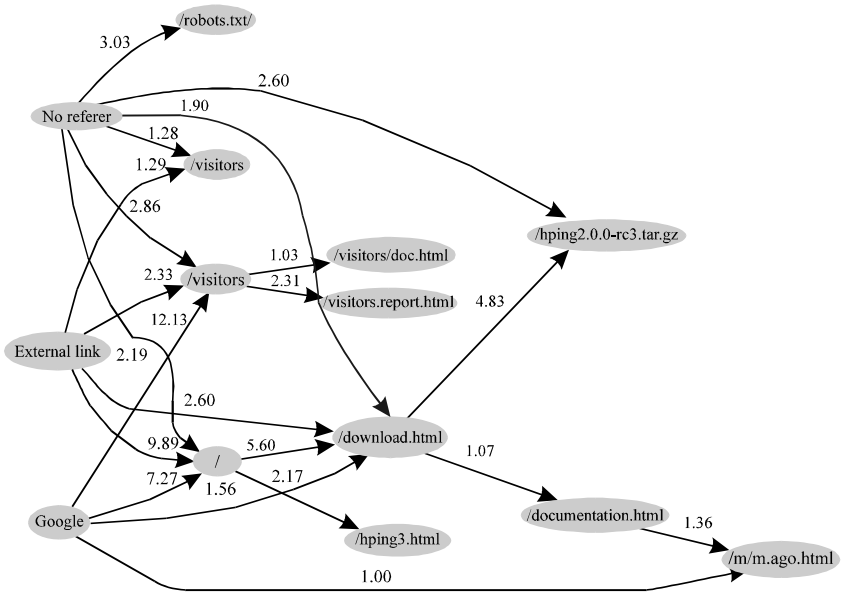

Fig. 1: Experimental results in the implementation. This illustrates the usage report of visitors in the organization'snetwork along with the applications they accessed

CONCLUSION

Researchers have studied the current security implementation in an organization's network and the various security vulnerabilities were identified. Server level security is implemented and an advanced intrusion detection environment is created through the process of OS Hardening. After OS Hardening is performed, unauthorized access into the server environment is reduced to almost nil. Strengthening of entry point routers through CBAC reduces the processing at the firewalls and the availability of the network has been increased to an appreciable level. Further, the implementation of log analyzer eases the task of manual analysis of log files. The log analyzer automatically checks the log file based on the rule base and thus presents a time effective and efficient way of log analysis apart from providing user friendly reports in the form of graphs.

REFERENCES

Alam, M.M. and C.E. Smith, 2001. A scalar network analyzer based on detector-log amplifier RFIC chips. Proceedings of the IEEE SoutheastCon, 30 March-1-April, 2001, Clemson, SC., pp: 283-291.

Bommepally, K., T.K. Glisa, J.J. Prakash, S.R. Singh and H.A. Murthy, 2010. Internet activity analysis through proxy log. Proceedings of the National Conference on Communications, January 29-31, 2010, Chennai, pp: 1-5.

Boneh, D. and M. Franklin, 2001. Identity-based encryption from the weil pairing. Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, August 19-23, 2001, Santa Barbara, CA., USA., pp: 213-219.

Boneh, D., B. Lynn and H. Shacham, 2001. Short Signatures from the Weil Pairing, Advances in Cryptology-Asiacrypt' 2001. Springer-Verlag, Berlin, Heidelberg, pp: 514-532.

Boneh, D., C. Gentry, B. Lynn and H. Shacham, 2003a. A survey of two signature aggregation techniques. RSA Cryptobytes, 6: 1-10.

Boneh, D., C. Gentry, B. Lynn and H. Shacham, 2003b. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Advance in Cryptology-Eurocrypt, Biham, E. (Ed.). Springer-Verlag, Berlin, Heidelberg, pp: 416-432.

Falcone, R. and C. Castelfranchi, 2001. The human in the loop of a delegated agent: The theory of adjustable social autonomy. IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans, 31: 406-418.

Ferraiolo, D.F., R. Sandhu, S. Gavrila, D.R. Kuhn and R. Chandramouli, 2001. Proposed NIST standard for role-based access control. ACM Trans. Inform. Syst. Secur., 4: 224-274.

Gentry, C., 2003. Certificate-based encryption and the certificate revocation problem. Comput. Sci., 2656: 272-293.

Gunther, C.G., 1990. An identity-based key exchange protocol. Proceedings of the EUROCRYPT'89, LNCS 434, April 10-13, 1990, Houthalen, Belgium, pp: 29-37.

Lysyanskaya, A., S. Micali and L. Reyzin and H. Shacham, 2004. Sequential aggregate signatures from trapdoor permutations. Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, May 2-6, 2004, Interlaken, Switzerland, pp: 74-90.

Nagaratnam, N. and D. Lea, 1998. Secure delegation for distributed object environments. Proceedings of the 4th USENIX Conference on Object-Oriented Technologies and Systems, April 27-30, 1998, Santa Fe, New Mexico, pp: 101-115.

Naor, M. and K. Nissim, 1998. Certificate revocation and certificate update. Proceedings of the 7th USENIX Security Symposium, January 26-29, 1998, San Antonio, Texas, pp: 217-228.

Nehinbe, J.O., 2010. Log analyzer for network forensics and incident reporting. Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS), January 27-29, 2010, Liverpool, pp: 356-361.

Neuman, B.C., 1993. Proxy-based authorization and accounting for distributed systems. Proceedings the 13th International Conference on Distributed Computing Systems, May 25-28, 1993, Pittsburgh, PA., pp: 283-291.

Parasuraman, R., S. Galster, P. Squire, H. Furukawa and C. Miller, 2005. A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with RoboFlag. IEEE Trans. Syst. Man Cybernet.-Part A: Syst. Hum., 35: 481-493.

Rivest, R.L., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM., 21: 120-126.

Schnorr, C., 1991. Efficient signature generation by smart cards. J. Cryptol., 4: 161-174.

Shehab, M., E. Bertino and A. Ghafoor, 2005. Secure collaboration in mediator-free environments. Proceedings of the 12th ACM Conference on Computer and Communications Security, November 7-10, 2005, Alexandria, VA., USA., pp: 58-67.

Sollins, K.R., 1988. Cascaded authentication. Proceedings of the IEEE Symposium on Security and Privacy, April 18-21, 1988, Oakland, CA., pp: 156-163.