Design of Application Layer Services for Security Automation via a Web Service Approach

Kai-Cheong Hee and Chiung Ching Ho

***Abstract*—N-tier application design has become very common in the IT industry. Each individual layer, such as the application and data layer has its own main functionality. This design is very helpful in securing the application from unauthorized access and in protecting it from attacks to the data layer. The data layer is the core of a company's business, as all the important information of the company will be stored in the data layer and normally will be located in a secured off-line server with limited local network access. The application layer acts as the medium to exchange data between the client layer and the data layer over a network. As such, the application layer has been increasingly targeted for intrusion and attacks. This paper will introduce a method to minimize the security risks in the n-tier application design. The method introduced in this paper will mainly focus on how to secure the application layer from various attacks such as Denial of Services (DoS) attack and spoofing attacks. This is achieved through data protection such as random encryption key generation, data encryption etc. and so forth at both the client application and the application layer.**

***Index Terms*—Automation, security, web services, encrypted security token.**

I. Introduction

The information technology (IT) industry has been changing faster than previously imagined. This has been driven by increased access to Internet technology all over the world. The Internet has become extremely important for many companies and organizations which are aiming for growth. Small and medium sized businesses are turning to e-commerce systems on the Internet to widen their sales areas to a global scale. These organizations are now able to expand their business to a multinational scale without additional overhead costs.

The need to have an IT solution system to help simplify business processes is growing over the years. Almost every area or sector we can think of, from a small food stall cashier to a large organization such as engineering, banking and government sector, requires an IT solution to provide more efficient and effective services or products to satisfy the customer's needs. Advanced and developing countries alike cannot experience continuous economic growth without the usage of new IT technologies. The extreme demands of modern-day IT solutions requires continuous research in order to better meet the wide range of specification requirements. One area of research is system automation. The goal of system automation is to lessen human workload, by automating difficult tasks or processes. More and more IT personnel are relying on automated systems because such systems can provide quick and accurate results and at the same time reduce errors due to human mistakes. Unlike human operators, these systems will report the truth all the time, which produces reports that are more convincing to decision makers.

The concept of a security culture [1] was introduced by J. Malcolmson, which discussed the assumptions, values, attitudes and beliefs and the behaviours performed by an individual in an organization pertaining to security. He suggested that implementation of a security culture can help to improve the organization's awareness on the importance of security. Due to the lack of security awareness of an individual in an organization, physical security is compromised by actions such as forgetting to lock a server room door. The same lack of awareness of security in a software development environment may mean security breaches happening due to system back hole, improper firewall and network monitoring, etc. and so forth. A security culture can be inculcated through effective training. Training IT personnel on how to use and implement the system in a correct way will minimize unexpected things happening. For example, trained users would know that unblocking ports on the firewall to enable Internet connection poses a security risk.

N. R. Mathiasen, M. G. Pedersen, and S. Bødker [2] suggested using the toolbox concept on every phase in the software development life cycle to identify and analyze the potential security issues which would occur. The result of the security analysis process would be the identification of specific security design activities that is appropriate for different security issues. With this approach, the implementer would have a more structured way of identify the security breach.

Security breaches can happen any time in many unexpected places. Up till today, there is no system that can guarantee total protection from security breaches. Whenever a new security approach is introduced, it is almost inevitable that a flaw or loophole would be discovered and subsequently exploited. Since system developers are unable to provide a fool-proof security solution to thoroughly protect their system, a better approach would be to minimize or prevent the system from direct interaction with the attacker. This paper suggest a solution based on the objectives of minimizing the security breach through securing of critical data during data exchanges between the client layer and the application layer; and preventing the system from direct interaction with the attacker. The core approach is to contain the threat, and to quickly move the system to another site instead of enduring the continuous attacks in an electronic siege mode.

Manuscript received July 30, 2013; revised October 29, 2013.

The authors are with the Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia (e-mail: ccho@mmu.edu.my , alwayshee@gmail.com).

II. Research Background and Related Works

The importance of protecting the application-layer from being exposed to unauthorized user is one of the most important problems to be addressed in any n-tier application. The application-layer is designed to contain the business processes of an organization and is the place to process and maintain critical data or information received from the client-layer application. Due to the distributed nature of multinational businesses, units from different areas or countries may need to access the same system for data transfer, operations and storage. The application-layer systems are required to be exposed to the Internet, in order to communicate with the client systems. Due to this reason, attackers are given an attack vector through the Internet.

This section will discuss on some of the existing systems which implements the n-tier application and their security concerns and solutions.

A. Cloud Computing

Cloud Computing is becoming an important part of today's IT industries, as shown by F. B. Shaikh and S. Haider [3]. Cloud Computing is designed to provide a set of resources and services through the Internet to the user. With the centralized resources and services concept maintenance costs is greatly reduced, and the cloud service providers are able to offer a competitive price for their users.

The idea or concept of cloud computing was first introduced during the 1950s to share physical access and CPU time using multiple terminals for multiple users to reduce the mainframes' operational cost. The cloud concept has been enhanced to be able to place users' application on the cloud. The cloud act as the application layer and data layer of a particular business or services. The client application in the client layer will connect and communicate with the cloud systems and perform the necessary workflow processes.

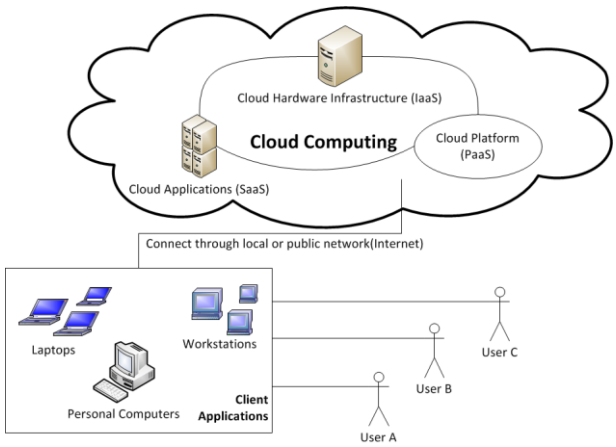

Fig. 1. Cloud Computing Architecture.

Services offered from the cloud to users can be categorized as the following, as reported by M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin [4]:

- Software as a service (SaaS), which offers software as a service to the customer
- Platform as a service (PaaS), which provides access to the platform (Virtual operating system) to allow cloud users to place their own application and data on the cloud
- Infrastructure as a service (IaaS), which leases the hardware infrastructure to the cloud user for data storage or processing, and may even include network connectivity.

To better illustrate the statement mention above, Fig. 1 shows the architecture of cloud computing.

In spite of the ubiquity and cost effectiveness of cloud services, provider such as Google Cloud, Microsoft Windows Azure Cloud, etc. and so forth are still faced with security concerns from its users.

Common question asked by cloud users would be on the trustworthiness of clouds and on the level of security assurance provided by clouds. Non-users find it difficult to trust the cloud concept as this concept is still considered new in the market and there is not enough information to support cloud computing as a reliable service to be used. In order to increase the cloud users confidence, a technique called Trusted Cloud Computing Platform (TCCP) was introduced to convince the cloud users. Direct Anonymous Attestation (DAA) and Privacy Certification Authority (Private CA) schemes were subsequently introduced to improve the TCCP model [5]. Concerns on data security were addressed by efforts such as Sealed storage ability (TSSC) [6] on the TCCP model .

The second security concern on cloud computing would be the ability for the cloud to defend against security attacks and the vulnerabilities. Cloud services are accessible by the public, putting it at increased risk as compared to a private intranet. The best approach recommended by F. B. Shaikh and S. Haider [3] would be the Transparent Cloud Protection System (TCPS) [7], a system to monitor the cloud activities closely by the cloud service provider, and perform necessary actions and decision as and when needed.

The last and most important security issues on the cloud computing would the security on the user's data. The privacy of the data and the ability to retrieve the same data when the services are corrupted is the main concern of the cloud users. In a business world, data is very sensitive, as data drives decision which will result in profit or loss. By putting the data in an unknown place with no control on it, users will have less confidence because the level of privacy and the level of backup on the cloud machines are unknown. A technique called Data Protection Framework (DPF) [3] was introduced to resolve this problem. D. Lin and A. Squicciarini [8] proposed a novel data protection framework with three major components, policy ranking at the client layer, policy integration at the application layer or the cloud services, and policy enforcement at the data layer or the cloud database. Different layers will contain different security policies and techniques to protect the data. Z. Xin, L. Song-qing, and L. Nai-wen [9] introduced a data security approach mainly in the application layer for the cloud services. They proposed that the user firstly need to have valid authentication, and subsequently the validated user is allowed to further perform the basic add, update, delete operation in the cloud services. The data received in the cloud servers will be encrypted with special encryption key, and lastly adhere to the necessary data security policy to protect the data from unauthorized access.

There are still many security issues on the cloud which is not covered in this discussion. The main objective of this discussion is to show some of the most common security breach in the cloud and the approach or model proposed to prevent or mitigate the situation.

B. Virtualization

Virtualization also means simulation of a computer resource or facility according to the Oxford Dictionaries [10]. Virtualization can occur on a hardware platform, operating system, storage device and network resources. The motivation to implement virtualization is to save on hardware maintenance cost. Multiple virtual servers exist concurrently at a same host server (Fig. 2), and dividing and sharing the same hardware resources, while the host server monitor, manage, and report the status for each of the child virtual server. Once applied to the n-tier architecture, the virtual resources would became the application layer, while the host act as the data layer, and the user who connected to the virtual resource will reside in the client layer.

Virtualization of hardware [11] is to divide an existing set of hardware devices into smaller hardware parts virtually to be used in a virtualization environment in the same machine. Hardware such as graphics cards, Random Access Memory (RAM), motherboard processors, et cetera and so forth can be simulated and used in virtualization. The hardware parameter is configurable, which means the user is permitted to decide how much hardware usage is required for the virtualization. For example, eight Gigabytes RAM in the host server can be configured to use two Gigabytes in the virtualization environment.

Virtualization normally refers to the simulation of the operating systems in a host server. Operating systems such as Linux, Windows and UNIX will be the targeted platforms. The invention of this technology is very useful for the technical experts to test their system behavior in different kinds of environment before live implementation. Huge amount of hardware maintenance cost will be saved by implementing virtualization especially for those corporations which owns many servers.

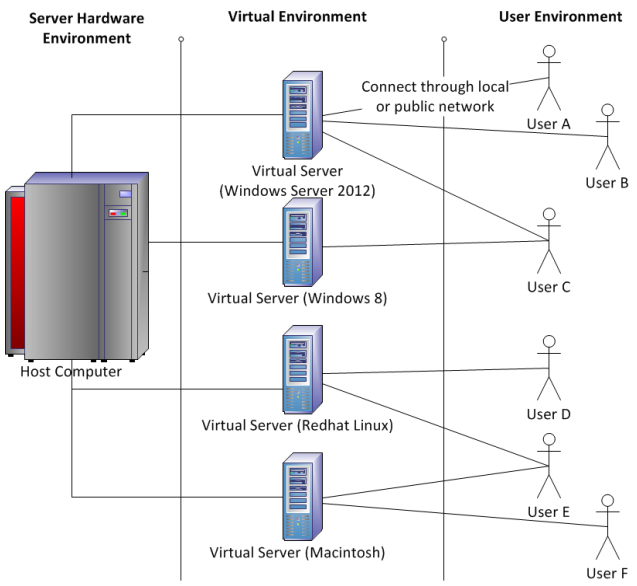

Fig. 2. Virtualization Architecture

Software virtual machine monitor (SVMM) system [12] such as VMware developed by VMware, Inc, Oracle VM Virtualbox now developed by Oracle Corporation, and Microsoft Virtual Server and Microsoft Virtual PC by Microsoft Corporation, is a good example system for creating virtualization environment. SVMM systems were designed and developed based on two foundations, the hardware virtualization configurable parameters, and the ability to install various types of operating systems on the same host. There is a second type of virtual machine monitor (VMM) system, which is called the system hardware VMM [12] but from the performance comparison with the SVMM, this approach produces lower performance. The hardware VMM still requires more research, development enhancement, and testing before it is made available for public use.

M. Pearce, S. Zeadally, and R. Hunt [13] put forward a comprehensive discussion on the security in virtualized environments. Each system would have potential security threats and risks which may compromise the system or put the system at risk. This section will not discuss all the security threats and mitigation methods identified by M. Pearce, S. Zeadally, and R. Hunt [13], but highlight three examples to better illustrate the problem faced in terms of the n-tier architecture.

The cloning problem [14] happens very frequently in virtualization. The virtual machine in the VMM is being cloned and used as a new virtual machine. The unique identity and instance of the cloned virtual machine will conflict with the original virtual machine. Once the cloning attack started, the number of virtual machine on the VMM will increase rapidly until the host server is compromise due to the uncontrollable memory and hardware usage. The original virtual machine becomes un-identical and it is difficult to perform an original installation action. The confidentially of the operating systems and its data became compromised.

Migrating an operating system to a virtual environment is also a problem [13], whereby an operating system from a physical server is being migrated into the virtual environment. The security threat would be the hardware and software simulation, where the virtual environment simulation hardware and software may not be compatible with the physical server's hardware. This will cause malfunctions on the existing systems in the operating system or the operating system itself.

The low security privilege of virtual OS in the VMM is an issue as well. The virtual operating systems needs to be fully dependent on the VMM's hardware and software simulation module. Hardware or software simulation failure in the VMM will cause all the virtual operating systems to be terminated. The level of trust between the VMM and the host server will depends on the stability and the ability of VMM and host server to handle critical situations, such as hardware malfunction on the host server or system file corruption on the VMM. The risk is very high to take this virtualization approach without any contingency plan.

Various models, approaches and mitigation plans were introduced to resolve the security threats for virtualization. M. Pearce and S. Zeadally [13] performed a thorough analysis and investigation on existing virtualization threats before identifying the mitigation plan for each of the

potential risks. The method used, such as hardening the server, threat prevention module, intrusion detection and prevention measures module etc. and so forth can help to minimize the security issues faced in VMM.

C. Server Consolidation

Server consolidation is basically similar to virtualization, with the objective of minimizing the use of servers in an organization. The only difference is that virtualization simulates the entire operating system environment in a host server with each system execute in its own different operating system environment while server consolidation (Fig. 3) will consolidate and install the systems from different servers into one particular host server to save energy and other server maintenance cost, and the consolidated systems in the host server will execute the hardware and operating system of the host server directly without any simulation. Virtualization is monitored and managed by the VMM software. In this case, the VMM software can also be consolidated into one host server while the host server is still able to support email systems, database servers, or web site hosting components, and many others software required by an organization.

With the increasingly needs to consolidate the systems into a single unit of server, the servers architecture design have changed to support multi-cores in a processor, and multi-processors in the servers as shown by H. Lv, X. Zheng, Z. Huang, and J. Duan [15]. Newly designed high performance server can even compete with mainframe server built using older architecture. This approach significantly reduces the workload of system administrators and eases their daily tasks. The system administrators no longer need to monitor and manage multiple servers but need only focus on a particular servers, which frees up their schedule and allow the spare time to be used productively.

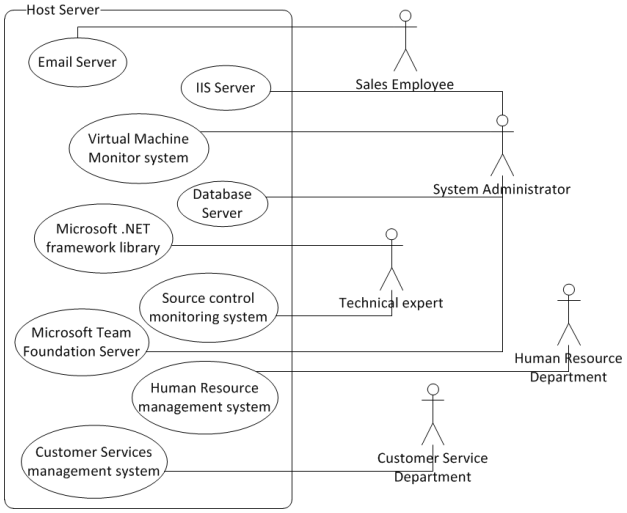

Fig. 3. Server Consolidation Architecture

No matter how good is the systemic security approach, there will always exist security threats, whether identified or yet to be identified. Three examples of frequently occurring security or operational threats for consolidated servers are provided in the following paragraphs.

The most frequently occurring security threat is the Denial of Services (DoS) attack as reported by X. Chen, S. Li, J. Ma, and J. Li [16]. The DoS attack will cause the host server to malfunction and ultimately be taken down. The network between the host servers and the outside world is the main communication medium for the DoS attack on which garbage data packages are sent to the target host servers in order to overload the server's memory and ultimately compromise the entire server. This is a very high risk for the server consolidation approach, as once the host server shuts down, all other application in this host computer will be malfunctioning and may cause the entire organization to be unable to continue operations for the time being. X. Chen et al. proposed a three steps approach, which includes determining performance parameters, calculating threat index and characterizing the threat state of service availability, to analysis the type and level of DoS for the technical expert to take further actions, such as firewall installation, DoS attack's IP diversion to dummy server, etc. and so forth.

Another security threat for consolidated servers are malware such as computer viruses, trojan horses, worms etc. and so forth. The most common action undertaken by system administrators will be to install anti-virus software on the servers with the trust that the anti-virus software can prevent and mitigate the harmful malware. M. Karresand [17] proposed the use of a "software weapon" to help solving malware related problems. N. V. N. Kumar, H. Shah, and R. K. Shyamasundar [18] proposed a method to identify and analyze each running systems' behavior in the host computer or server. If any of the system behavior changed, they will mark the system as "harmed" and any further observation or mitigation plan will be discussed and reviewed by the security expert.

Yet another point of failure for consolidated servers would be hardware. Every electronic devices will reach its end of life. Quality devices have a higher percentage to be useable for a longer time period. The stability of the hardware device is key to maintaining and continuing the host server's operation. Any hardware failure, will cause all the systems residing in the same host server to become unresponsive. Of all the hardware devices, the data storage devices are the most important for an organization. The organization's sensitive data are stored in the data storage devices, and any failure on the data storage may give permanent negative impart to the organization. Therefore, data backups are recommended to be performed frequently. Another approach known as redundant array of independent disks (RAID) [19], [20], prevents the host server from being compromised due to the failure in the main data storage as there is backup on the RAID devices. The RAID devices are designed to accept pairs of hard disk with same data image in both the devices, if one of the hard disk fails and is unable to continue operating, the RAID will automatically switch to the paired device without interrupting the current running operating systems. This will greatly reduce the percentage of system down time. The only disadvantage of the RAID technology is that both the hard disk is fully dependent on the RAID devices, and if the RAID devices are corrupted, the entire host server will still be compromised.

III. Proposed Work

Web service architecture is another type of framework

built on top of the n-tier architecture as the base foundation. Web services are typically used to facilitate communication between two electronic devices, in the offering and consuming of a software service. In this paper, we propose the usage of web services to automate security provisions This approach aims to reduce human errors while maintaining the configurable flexibility to the user.

The proposed idea in this paper is would benefit users from the company or enterprise level (B2B) who intend to develop their own in-house system, or who needs to integrate with other business partner's system for information exchange purposes with encryption for sensitive data .

The security automation via web service approach propose here will focus on the presentation layer and application layer of the n-tier architecture. The reason behind this is because in most cases, the two layers will always be designed to be exposed to the public domain and often functions with human interaction, as compared to the data layer which normally only resides in; and communicate through the private network. Fig. 4 shows the framework architecture proposed. The application server where the web services are installed, will not be restricted to only one server, and is scalable to new application servers on demand. Each application server provides the same functionality and act as the backup application server for the other server. If one server fails, the paired server will take the responsibility to share and to continue the processing of the tasks from the original server.

A set of application programming interface (API) [21] library is developed and implemented in the presentation layer and application layers. The web services in the application layer serves as the medium container for message request and response operations, and the API library will execute the tasks. This approach is to reduce the complexity of the web service designs and control the number of access point published to the public to prevent the possibility of various external harmful attacks.

From the client layer, every user is required to download and install the application from the client application setup web site. The same web site will be used by the user to launch their application each time they wish to use the client application. The web application has built in load balancing module which is used to identify and monitor the memory usage load from the application server and predefine the set of application server list arrange from least memory usage on the top and higher memory usage at the bottom. When a client launch the client application from the web site, the client application will automatically download the latest application server list and decide which web service will be connected to. The client application will do the checking periodically to maximize the performance usage in the application server. This approach indirectly protect against attackers by confusing them with the variable connectivity. The attacker is unable to identify exactly which server the client application will be connecting to for sending and receiving data. The local cache as shown in fig. 4 is mainly used for local parameter storing. All cache files will be encrypted in the client machine, to protect against unauthorized personnel from accessing the cache files directly.

The application layer will contains all the necessary web services. The web service is designed to be the only medium between the client and the application server. The system process and workflow will not execute in the web service, but will be sent to the server API for further action. The server API library contains all required base module and business module. The client request tasks such as add, update, and delete data, will be executed in this server API. The server API also includes the automation process for various core modules used for different purposes. All information, from technical information to business information will be stored in an application database.

In order to provide more structural module designs, the server and client API library will be developed by implementing some object-oriented programming (OOP) design pattern [22], such as the abstract factory design pattern used in the encryption module to provide single interface access point for various type of encryption method, the builder design pattern used in load balancing module to separate the tasks such as get server information, get server memory usage, get current web service load, etc. and so forth different application server, then combine the results and store in database, and the adapter design pattern to create a reusable class to incorporate different subclasses together. The implementation of software design patterns in the system development will give the benefit of increased structural system design and increased flexibility for future system enhancement.

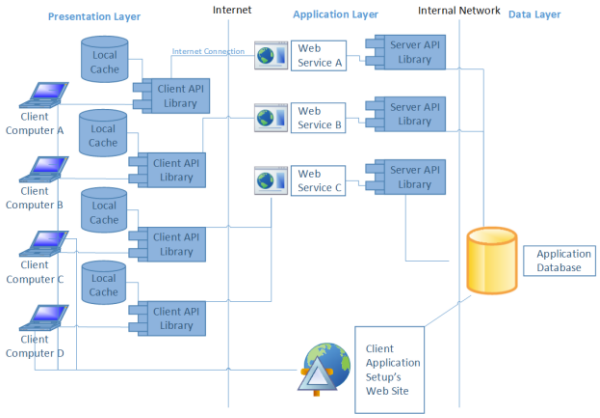

Fig. 4. Security Automation Framework's Overview Architecture.

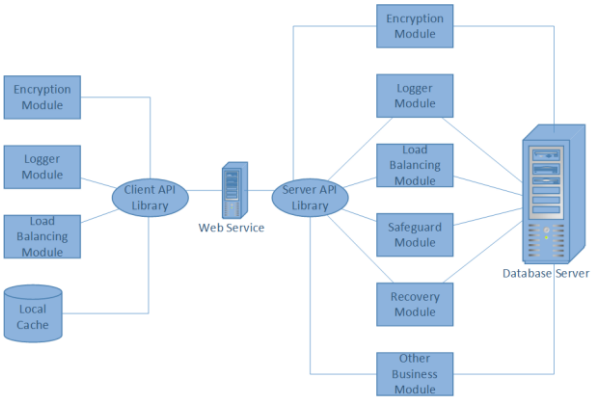

Fig. 5. Overview for API Modules

IV. Methodology

The set of library will categorized into four sections,

security, event logger, performance and system recovery. Each sections consists of its own respective modules (Fig. 5), such as the security section will consists of encryption modules and safeguard modules, event logger sections contains the logger module, performance section will contain load balancing modules, and the system recovery sections contain recovery module. Each module then decomposes into different approaches.

This paper proposes two main API, the client API which will integrate with the graphical user interface (GUI) at the presentation layer and the server API which will integrate with the web service at the application layer and communicate with the database server at the data layer.

A. Security Token as an Encryption Key

Security is an important part of the API. Basically, it will act as the gatekeeper to manage, control and monitor the security in the system. It is divided into four main modules, the security token modules, the encryption modules, the safeguard modules, and lastly the alert notification modules. Each module has its own responsibility and functionality.

The idea of implementing the security token approach is based on the work done by E. B. Fernandez, O. Ajaj, I. Buckley, N. Delessy-Gassant, K. Hashizume, and M. M. Larrondo-Petrie [23] who surveyed web services security pattern. The Security token approach basically is an approach to replace and reduce the repetitive user name and password authentication by using a token. The newly generated token will be included in the future client transaction call to the web service. The token is the proof that the user is an authenticated user, and was authorized to perform the required function. In this paper, the security token implementation is extended to being used as an encryption key for encrypting the data using different type of encryption method as shown in Fig. 6 The encryption key which is the security token, will be auto-generated by the system based on the given formula set by the user in the system. The user first sends the user name and password to the web services for authentication. The user name and password from the client API will be encrypted by using the predefine formula to generate the encryption key and chosen the encryption method. The predefine formula is configurable and is allowed to be change anytime by the system administrator. The formula will be downloaded but not stored on the client machine upon the start of client application execution. The formula will always require the current date time as the input to form the encryption key and choose the encryption method to use. The formula can be in any kind of form, and is flexible for the developer to define. The formula proposed in this paper, is shown below:

Given the date and time format,

YYYY-MM-DD hh:mm:ss:fff

To make each integer into a unique identifier, a sequence of number will be given to each integer format in the date and time fields:

$$Y_1\ Y_2\ Y_3\ Y_4 - M_1\ M_2 - D_1\ D_2\ h_1\ h_2{:}m_3\ m_4{:}s_1\ s_2{:}f_1\ f_2\ f_3.$$

Sample formula (the formula can be auto generated by the system or pre-define by the system administrator),

- Formula generated from system,

  Encryption key =
  <GUID> & (Y1 + Y2 + Y3 + Y4)
  & <Next GUID> & (M1 + M2)
  & <Next GUID> & (D1 + D2)
  & <Next GUID> & (h1 + h2)
  & <Next GUID> & (m3 + m4)
  & <Next GUID> & (s1 + s2)
  & <Next GUID> & (f1 + f2 + f3)
- Simple manual configurable formula,

  Encryption key =
  (Y1 + M1 + D1)
  & (Y2 + Y3 + M2 + D2)
  & (Y4 + h1 + h2)
  & (m3 + s1 + f2)
  & (m4 + s2 + f3)

The formula will be wiped off from the system memory once the encryption key is successfully generated. The user name and password will now be encrypted with the encryption key and the encryption method build in the client API library with compressed encrypted text ready to be transmitted to the web service. Three parameters will be sent through to the web service; the encrypted text, the current date and time the client application used for the formula input, and the server API function name to be performed. Upon reaching the web service, the web service will first decompress the encrypted text and identify the function name, and send the transaction to the respective module for further action, which is handled by the user authentication module. The user authentication module will first decrypt the encrypted text by using the same formula and the date and time parameter from the client. After decryption, the user name and password will be used for further authentication in the system. If the user credential is correct, a new randomly generated security token (encryption key) will be created and stored in the database. The security token will be valid for a time threshold set by the system administrator or will self-expire within 24 hours. After exceeding the time limit, the security token will be declared as expired and cannot be use anymore, and the user needs to re-authenticate again in order to generate a new security token. This is to minimize the percentage of an attacker from successfully guessing the encryption key. Once the encryption key is ready, the authentication module will use the first time encryption key and encryption method to encrypt the new generated encryption key and send it back to the client application. The client application receives the response from the web service, reads and stores the new encryption key in the local cache file where the new encryption key's text will remain encrypted in the local cache. Once the system is successfully logged in, the user is allowed to perform their daily tasks as required. All transaction that will transfer to the web service will be using the new encryption key. The first encryption key generated from the client application will be used by the client application for every local cache data stored in the client machine. To more effectively protect the client machine, the first encryption key will only stored in the system memory, as in the event of any unexpected system shut down, or system improper logout, the encryption key will be wiped off from the memory. A new authentication request to the web service is require when the situation mentioned above

happens.

The API library is built in with different types of encryption method, such as triples DES, Rijndael, AES, and etc. and so forth, mainly reused from the existing Microsoft .NET Framework 4.5 library [24]. The idea here is based on the predefine parameter, automatically choosing an encryption method to encrypt the data. The predefine parameter can be set to a default value by the system or can be configured by the system administrators. This approach can effectively confuse and slow down the attacker from guessing the correct encryption method. Depending on the predefine settings, each time a new transaction is sent to the web service may result in the use of a different method for the encryption and decryption process.

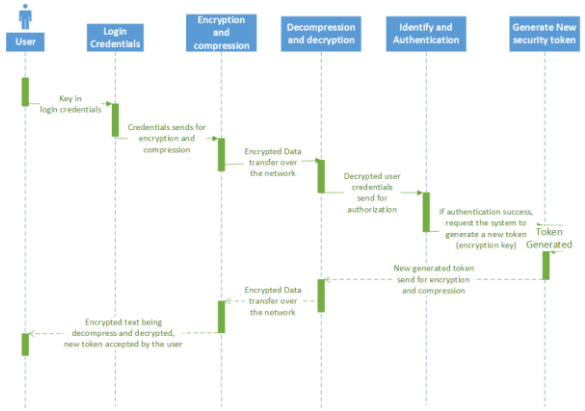

Fig. 6. Sequence diagram showing how a user request for a new security token

The safeguard module is responsible to be a security agent and monitor the web service. The main function is to prevent the DoS attack or unauthorized personnel access. The module provides two methods to handle the DoS attack. The first methods, automatically adds the harmful IP to the firewall monitor list, for further action by a designated security expert. The second method known as the "playing possum" method is used when the system successfully detects and confirmed that a DoS attack is happening on the application server. Subsequently, the system will automatically transfer all existing jobs to the other application server and disables the network adapter of the targeted server, giving the DoS the perception that the attack is successful, because the DoS main functionality is to deny all the services from the target machine, thus ceasing the attack upon success. After a period of time, (two-three hours depending on the system parameter settings), the system will try to enable the network adapter and go back online again. If the DoS still continues, the system will repeat the step to disable the network adapter again. After returning online, with no further DoS attack, the client application will be automatically notified by the web service that the targeted application server is ready for use. The client application will then automatically connect back to this application server. Other than that, the safeguard module also implements the set of predefined security policies, such as maximum password retries policy, minimum password length, dormant account monitoring, etc. and so forth. Unauthorized personnel access can be effectively prevented by applying the security policy to the system.

The alert notification module is an important design features in this system. The main functionality is to automatically notify the system administrator on any emergency and danger in the system. The alert notification module contains two type of notification. The first notification approach is by sending a critical email notification to the system administrator, although this approach will not be effective enough for very urgent and critical issues such as the system under attack. The next approach is to notify the system administrator by short message service (SMS) notification. This approach can give an immediately response to the respective personnel, so they can make an immediately decision and plan to mitigate the critical problem happening in the system. The level of notification to the respective personnel will be based on the system parameter settings. The system administrator also allowed to manually configure the system parameter settings as per their requirements.

B. Event Logger

An event logger module is useful to log the basic system information such as system error, and to have an advance system log, which logs user behavior on how the system is being used, the frequency of the server being attack etc. and so forth for further system analysis by the system administrator. The event logger is divided into three main layers, the client logger, the application logger and the data logger.

Client logger mainly focuses on logging the client application error and the user behavior on using the system, such as the time normally the user use to log in to the system, the function the user use the most, and even the emotional state of the user while waiting for a response (some user will keep on clicking the button when the system is slow to response). All this information is very useful for the developer to analyze, design and enhance the system to be more usable and user friendly. All the log information will automatically be transferred back to the application server through the web service and is stored in the database server.

The application logger focuses on logging the system error and the behavior of the system processes, such as the shortest and longest time for processing a task, the lesser and biggest memory usage for processing a task, etc. and so forth. The log is useful for task clustering and application server allocation especially for use by certain difficult processes. This will maximize the usage of the application server.

The data logger only records the database transaction error and information. From the log report, the system administrator will know which database table is heavily loaded, and which database table has a bad design with symptoms such as frequent primary key constant error. From the data logger, the system administrator also can identify the table usage for each particular user and the activity done by the user.

C. Performance

This section will mainly discuss on the task division among different application servers. All the server information will be stored in the database periodically. Each time a user access the client application web site and launches the client application, the web application will get the latest application server information and prepare the set

of application server list. Once the client application launch from the client machine, the first thing the client application does is to download the latest application server list. The list will be used by the client application to decide which application server it will connect to. The server list will not be restricted to only renew during the client application launch time, but will continue to update the latest application server list from the application server's web service.

The module is built in with the automatic change server connectivity functionality, which will be intelligent enough to automatically change the current connected application to the new appoint application server connectivity. There are only two situations where this sub module will be executed; first the current connected application server load is no longer the lowest among all the available application servers, and secondly the connected application server no longer responding to the client request. Therefore, within the entire period of user usage on the client application, the application server will not be fixed to only one access point. This approach can effectively organize and divide the tasks equally to maximize the processing speeds while reducing the connectivity down time by providing backup connectivity to let the client system continue operate without interrupting the user daily work tasks.

Data compression for every transaction sends from the client application to the application server. Data compression approach will significantly reduce the size of the message and reduce the sending time period. The system will operate the tasks efficiencies especially when process data size is big and the data that will compress is purely text data. This approach will greatly improve the system performance and reduce the time the data expose to the public which will giving the possibility for someone to sneak in and steal the data from the public network.

D. System Recovery

Recovery module mainly focuses on restoring the system into the last known stabile original states. This is done by replacing the corrupted system data files with the default system data files; and by periodical automatic backup of the system files, executable files or data files. The system parameter such as number of version copy to be kept, and the frequency to back-up the system can be configured by the system administrator. All modules are designed to perform the task automatically without human interaction. The modules will periodically check the system data files to make sure the entire system is in a healthy mode and notifies the system administrator if the modules find any issues.

V. Conclusions and Future Works

The proposed automation framework focuses on the security token as an encryption key which is only generated from the application server and encrypts the data by selecting the various encryption method based on the predefine system parameter. This approach is very effective as the attacker have no way to know which encryption method and the encryption key that will be used to decrypt the data. The encryption key is continuously changing to increase the difficulty to break the encryption key. However, there is the possibility of reverse-engineering on the encrypted data to decipher the code and get the key. This area is still not under consideration of the proposed framework and will be addressed in a future work. Other areas of improvement may include the decomposition of the client message into sub-messages to be sent through the public network with each pieces to different application server, and it will be re-combined back to original in the application server This approach will increase the difficulty level of breaking the encrypted data.

References

[1] J. Malcolmson, "What is security culture? Does it differ in content from general organisational culture?" in *Proc. 43rd Annual 2009 International Carnahan Conference on Security Technology, 2009*, 2009, pp. 361–366.

[2] N. R. Mathiasen, M. G. Pedersen, and S. Bødker, "While working around security," in *Pro. 3rd International Conference on Human Computer Interaction*, New York, NY, USA, 2011, pp. 1–9.

[3] F. B. Shaikh and S. Haider, "Security threats in cloud computing," in *Proc. 2011 International Conference for Internet Technology and Secured Transactions (ICITST)*, 2011, pp. 214–219.

[4] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Commun. Acm*, vol. 53, no. 4, pp. 50, Apr. 2010.

[5] H.-Z. Wang and L.-S. Huang, "An improved trusted cloud computing platform model based on DAA and privacy CA scheme," in *Proc. 2010 International Conference on Computer Application and System Modeling (ICCASM)*, 2010, vol. 13, pp. 33–39.

[6] G. Cheng and A. K. Ohoussou, "Sealed storage for trusted cloud computing," in *Proc. 2010 International Conference on Computer Design and Application (ICCDA)*, 2010, vol. 5, pp. 335–339.

[7] F. Lombardi and R. Di Pietro, "Transparent security for cloud," in *Proc. 2010 ACM Symposium on Applied Computing*, New York, NY, USA, 2010, pp. 414–415.

[8] D. Lin and A. Squicciarini, "Data protection models for service provisioning in the cloud," in *Proc. 15th ACM symposium on Access control models and technologies*, New York, NY, USA, 2010, pp. 183–192.

[9] X. Zhang, S.-Q. Lai, and N.-W. Liu, "Research on cloud computing data security model based on multi-dimension," in *Proc. 2012 International Symposium on Information Technology in Medicine and Education (ITME)*, 2012, vol. 2, pp. 897–900.

[10] Virtualize: definition of virtualize in Oxford dictionary (British & World English). [Online]. Available: http://oxforddictionaries.com/definition/english/virtualize?q=Virtualization#virtualize__4..

[11] C.-H. Huang and P.-A. Hsiung, "Hardware resource virtualization for dynamically partially reconfigurable systems," *IEEE Embedded Systems Letters*, vol. 1, no. 1, pp. 19–23, 2009.

[12] K. Adams and O. Agesen, "A comparison of software and hardware techniques for x86 virtualization," in *Proc. 12th international conference on Architectural support for programming languages and operating systems*, New York, NY, USA, 2006, pp. 2–13.

[13] M. Pearce, S. Zeadally, and R. Hunt, "Virtualization: Issues, security threats, and solutions," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 17:1–17:39, Mar. 2013.

[14] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: security challenges in virtual machine based computing environments," in *Proc. 10th Conference on Hot Topics in Operating Systems - Volume 10*, Berkeley, CA, USA, 2005, pp. 20–20.

[15] H. Lv, X. Zheng, Z. Huang, and J. Duan, "Tackling the challenges of server consolidation on multi-core systems," in *Proc. 2010 IEEE International Symposium on Workload Characterization (IISWC)*, 2010, pp. 1–10.

[16] X. Chen, S. Li, J. Ma, and J. Li, "Quantitative threat assessment of denial of service attacks on service availability," in *Proc. 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2011, vol. 1, pp. 220–224.

[17] M. Karresand, "Separating Trojan horses, viruses, and worms - a proposed taxonomy of software weapons," in *Proc. Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, 2003, pp. 127–134.

[18] N. V. N. Kumar, H. Shah, and R. K. Shyamasundar, "Can we certify systems for freedom from malware," in *Proc. 2010 ACM/IEEE 32nd*

International Conference on Software Engineering, 2010, vol. 2, pp. 175–178.
[19] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. K. Rao, "A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors," *Trans. Storage*, vol. 4, no. 1, pp. 1:1–1:42, May 2008.
[20] J. Wan, J. Wang, Q. Yang, and C. Xie, "S2-RAID: A new RAID architecture for fast data recovery," in *Proc. 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 2010, pp. 1–9.
[21] C. R. B. de Souza, D. Redmiles, L.-T. Cheng, D. Millen, and J. Patterson, "Sometimes you need to see through walls: a field study of application programming interfaces," in *Proc. the 2004 ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 2004, pp. 63–71.
[22] M. Huaxin and J. Shuai, "Design patterns in software development," in *Proc. 2011 IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, 2011, pp. 322–325.
[23] E. B. Fernandez, O. Ajaj, I. Buckley, N. Delessy-Gassant, K. Hashizume, and M. M. Larrondo-Petrie, "A survey of patterns for web services security and reliability standards," *Future Internet*, vol. 4, no. 4, pp. 430–450, Apr. 2012.
[24] System.Security.Cryptography Namespace (). [Online]. Available: http://msdn.microsoft.com/en-us/library/system.security.cryptography.aspx.

Kai-Cheong Hee received his bachelor of computer science (Hons.) degree from University Tunku Abdul Rahman, Selangor, Malaysia in 2005. He is currently working as a senior software engineer in Unixus Solutions Sdn Bhd, an IT company mainly focusing on developing logistic business solutions. At the same time, he is pursuing his master degree in computer science majoring in software engineering and software architecture in Multimedia University, Cyberjaya, Malaysia. He had worked before as the technical team leader in Consolsys Technologies Sdn Bhd, Kuala Lumpur, Malaysia for 4 years; focusing on developing banking solutions. Previously, he was the Senior Executive I in the City-link Express in-house developing team. Since his first job, he has been involved in the framework architecture design which has been used successfully by a number of clients.

Chiung Ching Ho received his bachelor of computer science and master of science (computer science) from Universiti Putra Malaysia. Ho is currently working as a lecturer in the Faculty of Computing and Informatics, Multimedia University. He is currently completing his PhD in Information Technology, pending a viva voce. His main research area is in the field of multimodal biometrics, smart mobile computing, and software design patterns. Ho is the author of a number of refereed indexed conference proceedings and indexed journals articles. He is professionally affiliated to Standards Malaysia as a member of the technical committee on health information standards, a IEEE senior member and also a founding committee member of ITSIM. He has completed a number of funded research grants and consultancy projects and has served as a reviewer for a number of conferences and indexed journals.