A SECURE ARCHITECTURE IN HEALTH INFORMATION SYSTEMS

R. ȚOEV[1] M. SCUTARU[1] T. IACOBESCU[1] M. ROMANCA[1]

Abstract: *Hospital institutions' needs of high-availability data resources are ever growing, and accordingly there must be included reliable measures to control the access security to data bases. In this paper, we propose architecture, based on the SONA model that could tend to a hospital's data security needs. The architecture is structured in three logical components: "Client", "Server-Farm" and "Management"; thus making it very scalable. Every component needs to satisfy a set of guidelines in order to obtain maximum efficiency and different implementations can be done with similar results, this being a proof of its high capacity of integration in other networks.*

Key words: *security, public institutions, smartcard.*

1. Introduction

Due to the fact that institutions all over the world are ever evolving, the need for centralized and standard databases has become increasingly high. This fact, combined with the continuous development of software and hardware devices, now provides a means of managing and controlling these databases, therefore increasing the efficiency of companies and institutions. Furthermore, the Internet and networking technologies have evolved in such a manner that it is now possible to manage and have access to databases remotely and in a very secure manner. In [1] the author purpose a three layer client-server architecture to be used in a HIS implementation for a Radiology Information System. The server resolves the problems of authentication, authorization, data security, privacy of access, protection. The author purpose as authentication method the use of a Secure Sockets Layer (SSL) Protocol, based on TCP/IP. In [4] authors adopt a combination of SSL and Internet protocol security (IPSec) procedures to maintain data confidentiality in a distributed infant and maternity care system. The paper [7] suggests the application of PKI (public-key cryptography infrastructure) and certificates to verify the authenticity of mobile users in the context of e-business and e-health information transactions.

2. Objectives

The purpose of this work is to provide a solution for hospital institutions needing a better and secures management system due to the increasing number of medical recording stored in the hospital data-bases. The solution contains an architecture which may be used to provide medical staff secure access to patients charts databases from anywhere in the world where an Internet connection is available. The work

[1] Dept. of Electronics and Computers, *Transilvania* University of Braşov.

focuses on the dataflow and main components of this architecture such as: server farm, network infrastructure, token based clients' authentification and operation and maintenance site.

3. Architecture and Components

3.1. Architecture

The architecture (Figure 1) is designed in such a way that it can be easily integrated in any existing infrastructure and its scalability characteristics being heavily taken into consideration.

This architecture is based on the concepts of SONA (Service Oriented Network Architecture), therefore being divided in three major structural components: Client level, OAM (Operation and Maintenance) level and server farm level. SONA is Cisco's architectural approach to designing advanced network capabilities into an infrastructure.

In regard to the particular solution implemented for the hospital institution, on the client side we have end-user terminals, from which trustee users can access database information, after smartcard based login.

According to the rank in the institution, clients access data on a central located server. Thus a doctor has access at more information on a patient chart, then a staff member with a lower rank in the institution. Data access is granted based on different authentication methods and technologies as smartcard readers, USB token smartcards, shared passwords etc.

On the OAM level, administration and maintenance of the infrastructure is realized taking into consideration that the whole concept of this architecture is to perform the entire management process from a logical centre site.

The server farm is typically located on a different site than the client's. Therefore means of remote and secure access must be implemented.

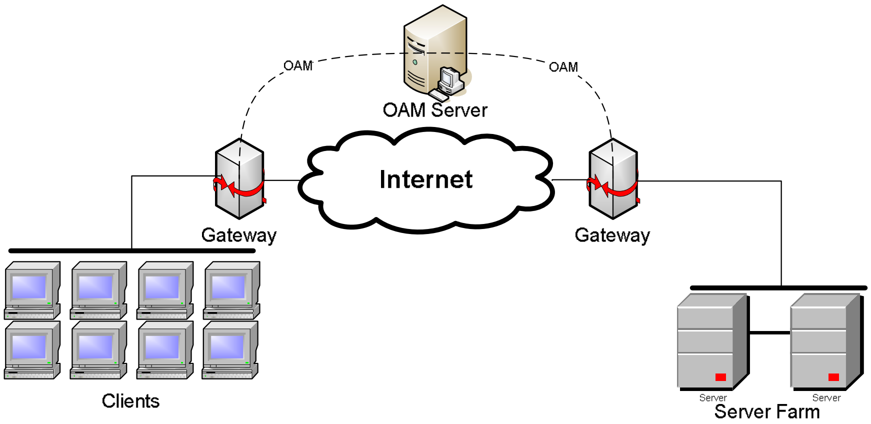

Fig. 1. *Network Architecture Concept*

3.2. Data Flow

During the communication between the network elements a strict data flow must be defined, in order to ensure the standardization of the process. On the client side it must be guarantied that all traffic is directed to a gateway. In this segment security is not a strong issue because all the traffic is internal to the institution. Security truly becomes an issue when the data passes through the network's gateway. At this point several measures have to be implemented. An example of this is to encrypt the outgoing traffic.

On the server side, at the inbound gateway, traffic coming from the clients must be processed in order to ensure the correct decryption. Further steps must be taken in order to ensure that the decrypted data reaches its designated server.

The primary function of the server is to provide the necessary data to the client. In this stage uptime is critical. The most efficient way to guarantee that the resources are 100% available is through redundancy, either local or geographical.

When implementing network redundancy, one must consider the possibility of a server failure. A backup server with identical resources and synchronized data must exist. In regard to the infrastructure, a fail over mechanism must be implemented so that traffic reaches the backup server, transparently from the clients' point of view. This is done through high availability protocols such as HSRP (Hot Standby Router Protocol) that monitors interfaces, and in case of a failure, directs all traffic through an alternate path.

In regard to the scale at which this architecture can be implemented, clients being spread over a large geographical area, a really high traffic load is estimated. Through an efficient load balancing entity, the traffic, however high it may be, can mitigate congestions. When applying load balancing, the synchronization between the servers must be taken into consideration.

3.3. Components

3.3.1. Server

In order to implement a system that takes advantage of a strong authentication method for secure network connectivity and data security, a central management system must be implemented. This system should control and organize smart card tokens throughout the institution. Without a well structured system, token based authentication solutions can be difficult or even impractical for organizations to adopt. The provisioning and life-cycle management abilities provided by token management systems make the implementation of strong authentication token solutions a reality, enabling organizations to enhance their security efficiently, in a flexible manner, and with significant cost savings.

The server, from an end-product point of view, contains the databases meant to be accessed by the clients. There are various ways of implementing these databases; SQL, MySQL and Oracle are some examples. These databases should be stored at the central location, in order to be reachable by all clients. Their structure can be distributed over multiple servers, but this brings up synchronization issues. A mechanism that ensures the correct placement of information into the database must be implemented, so that no redundant entries are present.

In this network segment we must have a central authority that stores all the users and their permissions. Most likely this implementations will be in the form of a domain controller, such as Active Directory for Microsoft Server, or OpenLDAP for Linux implementations. This service should be able to handle the complexity of large institutions such as hospitals. This implies that users should be grouped in well defined groups, such as departments, medical specialty and overall rank in institution. The controllers should maintain a database of all terminals in the institution and their purpose.

Another important component of the server structure is a certificate authority (CA) for all the digital certificates issued inside the domain.

A CA issues digital certificates that contain a public key and the identity of the owner. The matching private key is not similarly made available publicly, but kept secret by the end user who generated the key pair. The certificate is also an attestation by the CA that the public key contained in the certificate belongs to the person, organization, server or other entity noted in the certificate. A CA's obligation in such schemes is to verify an applicant's credentials, so that users and relying parties can trust the information in the CA's certificates. CAs use a variety of standards and tests to do so.

Digital Certificates bind an identity to a pair of electronic keys that can be used to encrypt and sign digital information. A Digital Certificate makes it possible to verify someone's claim that they have the right to use a given key, helping to prevent people from using false keys to impersonate other users. Used in conjunction with encryption, Digital Certificates provide a more complete security solution, assuring the identity of all parties involved in a transaction.

The Certificate Authority should have the same flexibility as the domain controller at assigning certificates to users depending on their rank in the institution.

To increase usability of the terminals, the logon mechanism for each workstation should be simplified. There are numerous ways how this could be implemented, but the most common would be the use of smartcards. These smartcards should have a central management system, typically located in the server farm that is able to deploy them according to the clients' needs. Each card must be loaded with the user credentials and clearance level inside the institution. For example the permissions of a doctor and a nurse are well differentiated.

3.3.2. Client

At the client's side it's imperative to ensure an easy to use and practical mechanism for logging into the domain. This mechanism also needs to provide mobility for the user, thus a doctor from a hospital might be able to access patients' charts and prescribe treatments from another city by simply logging into the domain.

Given these guidelines, we have several options of implementing the client side in the proposed architecture, like smartcard readers or smartcard tokens.

Tokens are used to prove one's identity electronically. The token is used in addition to or in place of a password to prove that the client is who he claims to be. Hardware tokens are typically small enough to be carried in a pocket or purse and often are designed to attach to the user's key-chain. Some may store cryptographic keys, such as a digital signature, or biometric data, such as fingerprint minutiae. Some may include small keypads to allow entry of a PIN or a simple button to start a generating routine with some display capability to show a generated key number. Special designs include a USB connector, RFID functions, or Bluetooth wireless interface to enable transfer of a generated key number sequence to a client system.

The token needs at least an inherent unique identity in a protected memory that cannot be tampered and preferable is not openly accessible by other application but that original method offered by the certificate authority.

Relative to the workflow proposed by the aforementioned architecture, the certificate authority, located in the server-farm side, will load on the token digital certificates that set the level of clearance in the created domain.

3.3.3. Infrastructure

The part that connects the logical sides of client and server-farm is the infrastructure side.

As our architecture proposes, the infrastructure has two components: gateways, which apply packet-filtering techniques on data leaving or arriving at the client side or the server-farm side, and the logical path between sites.

The infrastructure component ensures security of data passing through the institutions' backbone or through the Internet from site to site.

In any type of network measures must be taken in order to prevent attempts to hack data, such as patients' insurance contracts.

"Sniffing" is observing packets passing by on a network. Sniffing is a popular way to steal data from a network, usually in form of passwords, ID names etc. Passive attacks using this method have become frequent on the Internet. The data is usually cached and hackers look for user ID and the password of a legitimate user and use the user's information to log on to the network. Once logged into the network, the hacker captures transmissions of packets. With this method the hacker can gather needed information about the network.

There is a method that hackers use in which the attack lets a hacker redirect the TCP stream through the hacker's machine. Once the hacker has redirected the TCP stream, the hacker can bypass a systems protection (One-time password, ticketing authentication). Remember that a Transport Control Protocol packet may travel over many systems before reaching the destination system. With a sniffer and a generator a person could easily access many packets.

So, in order to avoid these attacks, an efficient security mechanism must be implemented. There are various ways of preventing this kind of attacks. Identification schemes using one-time password or ticketing authentication are some examples. They ensure that institutions systems are protected from Internet attacks.

Deploying a firewall between these systems and the Internet to guard against network scans and intrusions is another way. Both methods risk attack, but combining them and adding encryption to the data stream make hacking almost impossible.

The gateway, located at the edge of each site, is designed to block unauthorized access while permitting outbound communication. It can also be a device or set of devices configured to permit, deny, encrypt, decrypt, or proxy all computer traffic between different security domains based upon a set of rules and other criteria. Gateways can be implemented in both hardware and software. They are used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. Several techniques can be implemented, such as packet filter, application gateway, circuit-level gateway or proxy server. All of these solutions have advantages and disadvantages.

Packet filtering looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.

The application gateway applies security mechanisms to specific applications, such as database access. This is very effective, but can impose performance degradation.

The circuit-level gateway applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

The proxy server intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses, but effectively breaks the server-client connection.

4. Conclusions

The design of the proposed architecture is created in the idea that the architecture needs to be scalable, easy to integrate in existing networks, secure and must have a central administration site.

These features were implemented by creating the architecture on the SONA standard, thus assuring its' scalability, and separating it into three main components.

As described in the paper each logical component has to provide a set of features that will provide the main characteristics of the architecture.

We demonstrated that there are several ways of implementing each logical component, each way having its' advantages and disadvantages, thus making it easy to integrate in existing networks and also assuring the level of security desired by the institution.

Administration can be done locally or from a remote site and the security is assured by using different techniques of fighting hacks in the system.

Based on this facts, we can state that a physical implementation of this architecture could be possible, not just in a testing environment, but in an actual hospital or network of hospitals.

References

1. Bourke, T.: *Server Load Balancing.* O'Reilly Media, Inc., 2001.
2. Cordos, A.: *Studii şi cercetări privind managementul, prelucrarea şi transmisia, informaţiilor cu aplicaţii în domeniul medical* (*The Study of Data Management, Processing and Transmition Using Applications in the Medical Field*). Ph.D. Thesis, Technical University of Cluj-Napoca, 2008.
3. Kopparapu, C.: *Load Balancing Servers, Firewalls and Cache.* John Wiley & Sons, 2002.
4. Kouri, P., Kemppainen, E.: *The Implementation of Security in Distributed Infant and Maternity Care.* In: International Journal of Medical Informatics **60** (2004) No. 2, p. 72-74.
5. Marcus, E., Stern, H.: *Blueprints for High Availability.* 2nd Edition, John Wiley & Sons, 2003.
6. McCabe, J.D.: *Network Analysis, Architecture, and Design.* In: Morgan Kauffman, 2007, p. 173-210.
7. Tan, J., Wen, H., Gyires, T.: *M-Commerce Security: The Impact of Wireless Application Protocol (WAP) Security Services on e-Business and e-Health Solutions.* In: International Journal of M-Commerce **1** (2003) No. 4, p. 43-52.