IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 1, Ver. III (Jan – Feb. 2016), PP 86-96
www.iosrjournals.org

Improving the Security Layer logic for a Health Care Information System

Abdelghani El malhi[1], Mohamed Ahd[1], Abdelrhani Mokhtari[2], Rachida Soulaymani Bencheik[3,4], Adil Echchelh[1],Abdelmajid Soulaymani[2]

[1] *Laboratory of electrical engineering and energetic systems Ibn Tofail University, Faculty of Sciences, Kenitra, Morocco*

[2]*Laboratory of Genetics and Biometry, Faculty of Sciences Ibn Tofail University, Faculty of Sciences, Kenitra, Morocco*

[3]*Moroccan Poison Control and Pharmacovigilance Center, Rabat, Morocco*

[4]*Faculty of Medicine and Pharmacy, University Mohammed V, Rabat, Morocco*

Abstract: *In a previous paper we suggested an information system to store, manage and treat millions of the gathered patient's information. We were able to propose a reliable application, which is able to fulfill the most important criterions, mainly measurement, monitoring, guidance, Data management and their analysis. The introduction of the Internet and the related technologies are offering countless opportunities to build an efficient healthcare system, but in the same time, Software-applications are facing adoption and securities challenges.In the software development, end users often lack knowledge of vulnerabilities, attacks and threats. Application designers, developers and testers are discovering and fixing constantly bugs, defects and flaws.The aim of this paper is to highlight the importance of integrating and adding a high level of security to the previously developed application by advocating a reliable and protected information system. To this end, we examine first the several possible sources of vulnerability. Secondly, we will present a viewpoint explaining the fundamentals principles that a secured health care system should possess. In a third part we will discuss the proposed strategy and system architecture to defend against attacks. Today, the technology could contribute actively in resolving some of problems. Connected computers are increasingly being introduced into safety-critical health care systems and as consequence have being involved in the progress but also in some accidents. Concretely in this study, the objective is to obtain a safely data management system by implementing an additional level of logical and physical data security.To summarize, this paper presents a software development concept, particularly with a web-based focus. It introduces a common application's attacks and suggests methods to defend against them.*

Keywords; *E-Health, data security, system vulnerability, database integration, e-health care system, attacks.*

I. Introduction

During the last years, the health authorities collected patient's data from the different regions in Morocco, then stockpile it in several sheets in order to analyze the different statistics and collected data. In a previous study we proposed a web application to handle and treat the patient's data. This paper examines the security aspect related to the program.

As per earlier researches, it turned out that information systems are often containing confidential data. Therefore the vulnerability revision is an important milestone. Database and web interactions could be a source of frustration and problems to the authorities. The networked E-health care system should be secure and support basically measurement, monitoring, guidance, management of data and their analysis [1].

Rakesh and Christopher presented in 2007 an integrated set of technologies, known as the Hippocratic Database, that enable healthcare enterprises to comply with privacy and security laws without impeding the legitimate management, sharing, and analysis of personal health information.[2]

The stored data in the system is very sensitive and should be thoroughly protected from attackers with a high level of security.

By Security or confidentiality we mean, informally, that the information can only be acquired by agents or processes entitled to such access [3]. Today software is everywhere, the majority of devices become networked and this forced us to care about security, confidentiality, integrity and privacy of the information that floods between systems. In parallel the security problems continue to grow. Concerned are mainly: Web browsers, web servers, database management systems and commonly used software.

Faults and Errors in a computer program produce incorrect and unexpected results, consequently the application will behave in unintended way. They can be a possible source of penetration to the program by the hackers either from inside sources or outside.

On Friday 2 June 1994 a Chinook helicopter ZD576 crashed in Kintyre killing 29 people. On Wednesday 6 February 2002 the House of Lords committee report found that there is doubt about the cause of the crash because of the possibility of a technical malfunction. Extracts from various press articles and reports illustrate the broader issues related to the development of safety critical software [4].

Patients were given massive overdoses of radiation from Therac-25, Because of concurrent programming errors, the therapy machine sometimes gave its patients radiation doses that were hundreds of times greater than normal, resulting in death or serious injuries [5].

These accidents highlight the dangers of failures and defects in the software control of safety-critical systems.

Rakesh and Christopher presented an integrated set of technologies, known as the Hippocratic Database, that enable healthcare enterprises to comply with privacy and security laws without impeding the legitimate management, sharing, and analysis of personal health information.

Statistics demonstrated that the number of vulnerabilities discovered has increased. For example, 7937 vulnerabilities were reported in 2014, whereas only 6608 were reported in 2006 and 246 in 1998 (Fig. 1).

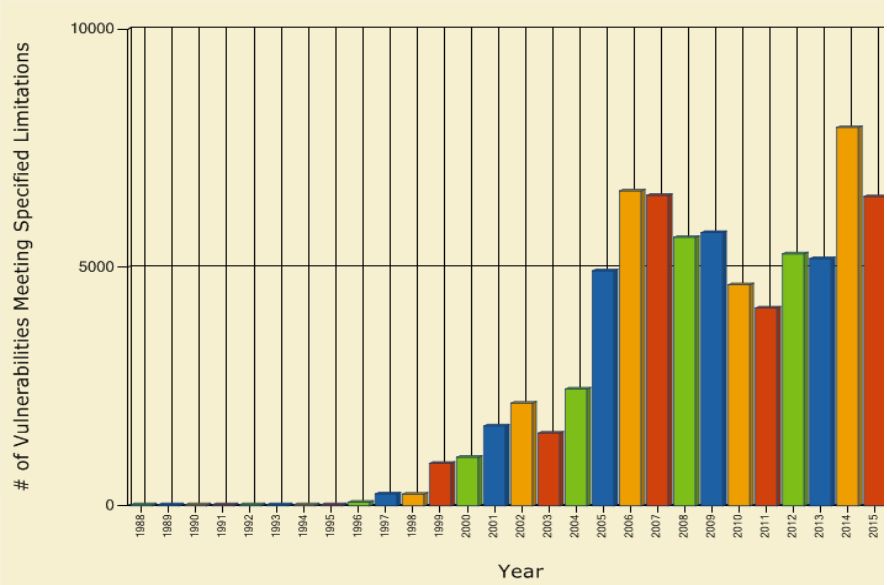

Figure 1. NVD Software Vulnerability Statistics [6]

The consequences of a class of system failures, commonly known as software vulnerabilities, violate security policies. They can cause the loss of information and reduce the value or usefulness of the system [7].

The developed information system provides an easy-to-navigate environment with the following capabilities:

- Data extraction and analysis
- Reporting and distribution
- Decision-making helper
- Accessibility and robust

While all these elements are necessary to achieve success, the focus of this paper is on ensuring more security and reliability.

Requirements are for the most stakeholders all about what the software should perform with an hyper-focus on the functionality but what can go wrong must be considered as well. After finishing the conception and the implementation, one of the most important tasks is to try finding the eventual gaps.

In addition to the software requirements, the operating Systems face escalating security challenges because connectivity is growing and proportionally the overall number of incidents is increasing. CERT and other databases keep track of reported vulnerabilities. An increasing number of individuals and organizations depend on the internet for financial and other critical services. That has made potential exploitation of vulnerabilities very attractive to criminals with suitable technical expertise. [8]

II. Material And Methods

We will discuss in this chapter the important guidelines to build a secured architecture within a tiered Model concept and how can we defend against attacks.

The first section examines the system health Check that needs to be performed in regard to a web oriented application. However the second part will discuss the various challenges to setup a secured system. It includes the different development phases including the testing. At the end of this chapter we will propose a concept to strengthen more the system security.

Kocher introduced the Trinity of Trouble to the world in 2004 in a paper about security as a new dimension. It turns out that the Trinity conspires to make managing security risks in software a major challenge [9].

- Connectivity: is the software in Ethernet? The internet is everywhere and most software is on it.
- Complexity: machines are often networked (client and server or even more distributed)
- Extensibility: Systems evolve in unexpected ways and are changed on the fly.

Researches make significant progress, the trinity of trouble-connectivity, complexity, and extensibility-goes a long way to making things difficult for system analysts, programmers and IT workers.

The trinity of trouble also makes the software more difficult. It's business-critical and causes significant impact when it fails to perform what was expected.

Three-tier Model

Organizations should employ a layered security strategy that provides necessary access to corporate information while minimizing risk and maintaining compliance. When it comes to sensitive information, the focus must go beyond authorized and unauthorized users and extend data protection from storage through transport to delivery on the endpoint to prevent sensitive data loss [10].

The challenge is how to setup a secured design to ensure the expected level of security. We opted for the 3-tier architecture. The application is organized into three major parts, each of which is distributed to a separated location in the network (Fig. 2):

1. The client machine / browser / Graphic user interface
2. The application server which consists of business and data rules.
3. The database server to store and retrieve the required data.

IT security groups tend to not understand software development but architecture. They are neither able to influence the end user to work with a specific browser nor to install the suitable security patches and appropriate plug-ins. The IT people could participate actively by making the network more secure.

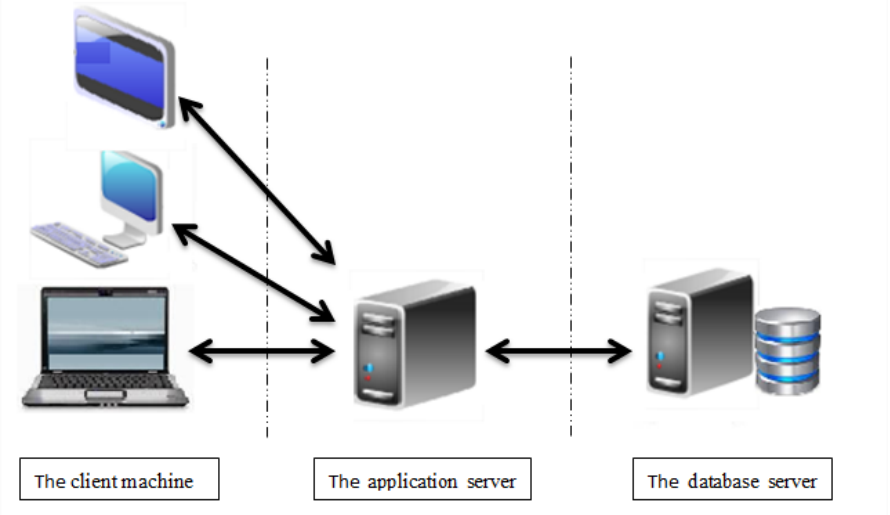

Figure 2. Three-Tier Architecture

The three-tier design has many advantages:

- Separating the application from the database server makes the process of maintaining both systems easier.
- Reusability and flexibility: the components can be distributed across the network as needed.
- Security: when the middle-tier platform is separated physically, an extra level of indirection between the client and the database is added. So no direct communication from the web application to the database server.

HTTPS:

HTTP is the standard communication protocol that a browser uses to connect. It doesn't possess an encryption mechanism. Everything in the traffic (including passwords) is transparent to sniffers.

Sniffer is a packet analyzer that can intercept and log traffic that passes over a digital network. The connection is established using either cable or wireless media. It is able to capture and record a variety of restricted personal information in a data stream such as passwords.

In the below http form "adduser_submit" the administrator submits a user's personal information to the server in order to store the related credentials in the connected database (Fig. 3).

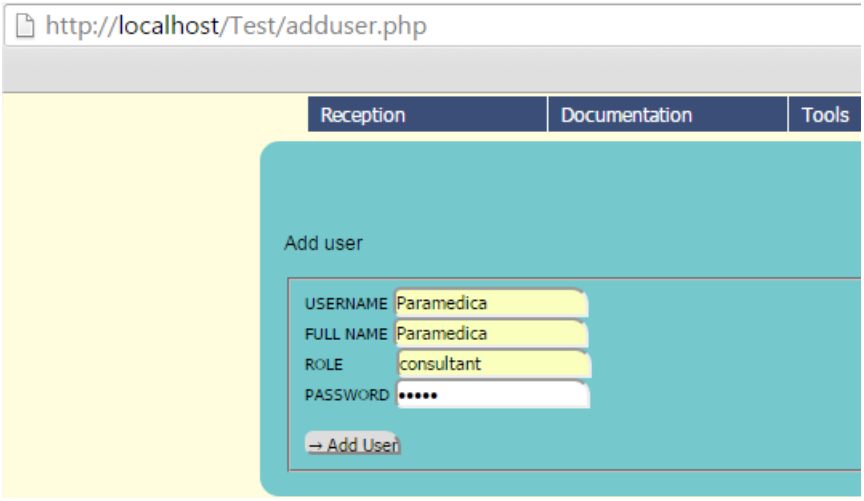

Figure 3. "adduser_submit" form

Inappropriately, even if the characters in the password field are masked (shown as black points), it is transferred over the network in a clear text and can be captured by everyone connected to the system nodes (Fig. 4).

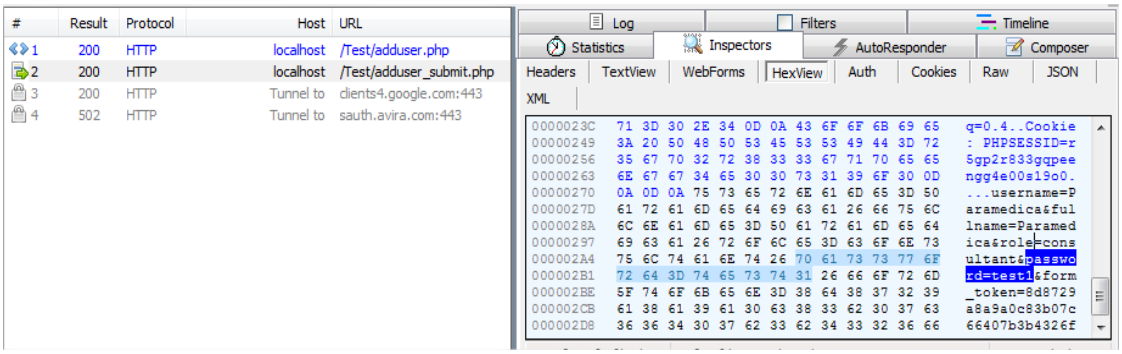

Figure 4. Fiddler session: password in clear text

Being able to send and receive data securely over a network is of growing importance. To that end, many web applications force users to use SSL certificates to encrypt traffic between the client browser and the server (application server). SSL is a secure method to encrypt data from your computer and send to the server, keep information private and safe. In order to enable the encryption of the password and personal data, an SSL certificate (containing your public key) and a server private key should be generated.

For the test purpose we do not need to buy a specific SSL certificate. Instead Apache offers "self-signed" certificate for free. It might be possible that the browser gives a security warning like "server certificate is not trusted" for using this particular certificate but this can be ignored as the test will be performed just locally.

HTTP protocol uses by default the port 80 to communicate (Fig. 5). This can be changed by a simple configuration in Apache httpd.conf file. The biggest benefit to changing the port number is to avoid being seen by casual scans and script kiddie but a determined attacker can still find the port if they know your server's IP address. Changing the typical port number isn't efficient enough and undoubtedly doesn't provide any serious defense.

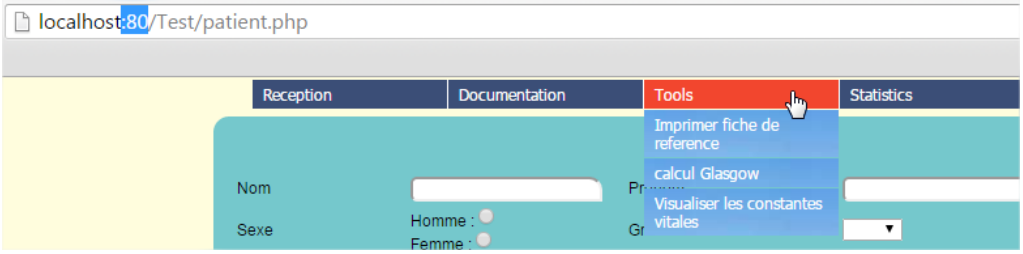

Figure 5. HTTP Default port

To setup an encrypted communication within the used web-server, below are the basic steps to create Self-Signed Certificate:

i. Create a Private Key :

- Open a command window
- Change directory to \apache\
- makecert

✓ Create a Certificate Signing Request (CSR)

✓ Create a Self-Sign certificate using the Private Key and the CSR

Alternatively, if the SSL is purchased form a provider, this step is accomplished by the certificate signing authority

ii. Import the certificate into the browser for each client:

Since the generated certificate is just a self-signed one, by the first attempt to connect to the server, the user may receive a warning by navigating in the protected pages. The certificate should be then imported as a trusted CA into the browsers.

iii. Make Apache folders accessible with SSL encryption:

The web-server httpd.conf file will be edited and instructed to access the password protected folders only with SSL encryption exclusively.

This is accomplished by putting the SSLRequireSSL directive inside of <Directory> listing in the config file.

```
DocumentRoot "C:/xampp/htdocs"
<Directory "C:/xampp/htdocs">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
    SSLRequireSSL
</Directory>
```

With this setting you will only be able to access the protected pages by typing https:// in the address.

iv. Redirect "http" to "https" for certain folders

This allows the automatically switch to https:// and encryption even if the user types http://. It is more users friendly. The generic text below needs to be added to httpd-xampp.conf and make sure to enable mod_rewrite in the apache config file.

```
RewriteCond %{HTTPS} !=on
RewriteCond %{REQUEST_URI} FOLDER
RewriteRule ^(.*) https://%{SERVER_NAME}$1 [R,L]
```

For a test purpose, we implemented the Self-Signed Certificate (Fig. 6):

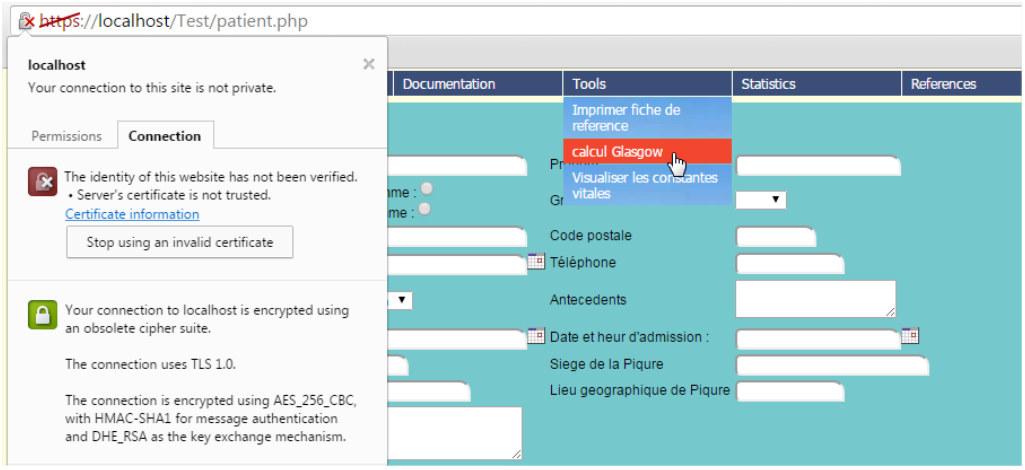

Figure 6. Apache Self signed Certificate

File Replication:

We recognize the importance of multiple strategies for prevention, detection and recovery from cyber security attacks. The administrator should be able at each time to recover the system and make it back online. Data replication is a key component of online-offline database model. It implements data synchronization between the online database installations, used by the experiments control systems, and offline databases. Both data sets are initially archived on the online databases and subsequently all changes to the data are propagated with low latency to the offline databases by the database replication solutions [11].

The Application generates automatically patients' files, which contains all related information, most likely in form of PDF and images. By default, these documents are stored in the server itself.

The replication concept allows the administrator to copy the content of a specific directory from one location to another one or even to an external device in the network. The replication process can be scheduled to start and stop at specified times. The purpose of the file replication is to facilitate recovering by any eventual attack.

In Fig. 7 below, We illustrated the algorithm used to replicate the existing files:

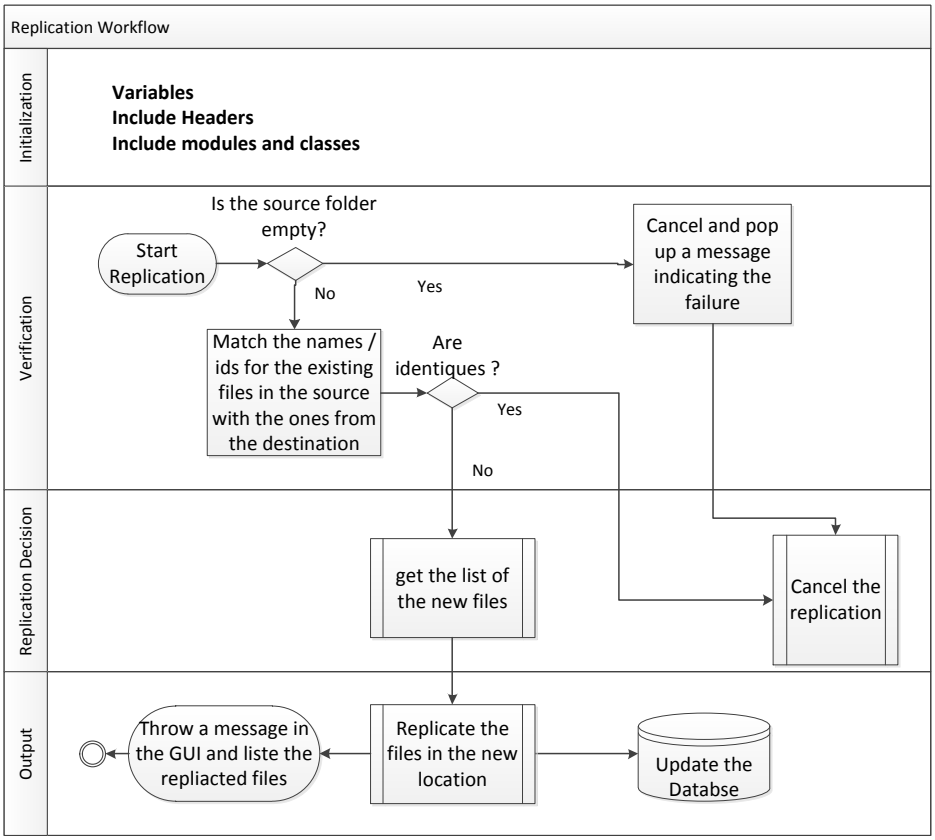

Figure 7. Replication Workflow

Results are looking like the following (Fig. 8):

Results

File Path	Status
\Abdelhalim.pdf	Match
\ahmadi.pdf	New File
\helas.pdf	New File
\Naji.pdf	Match
\Rachad.pdf	Match
\Rachidikkkk.pdf	New File
\Rachoud.pdf	Match

Figure 8. List of replicated files

Database storage (Backup and strategy)

Techniques for data integrity and availability specifically tailored to database systems must be adopted. In this respect, over the years, the database security community has developed a number of different techniques and approaches to assure data confidentiality, integrity, and availability [12].

The system administrator should work in conjunction with the DBA team to establish the appropriate and adequate backup strategy to answer the needs of the organization or the establishment. Backup frequency, database Recovery model and other related settings can be discussed further with the database vendor.

Researches confirm that policies concerning the disclosure of electronic health records can be reliably and efficiently enforced and audited at the database level [2].

We implemented the simplest scenario by establishing a Snapshot replication where the available data is solely copied to another server in the network (Fig. 9). The benefit of such implementation is the performance gain and protection of the application availability, this offers an alternate data access options.

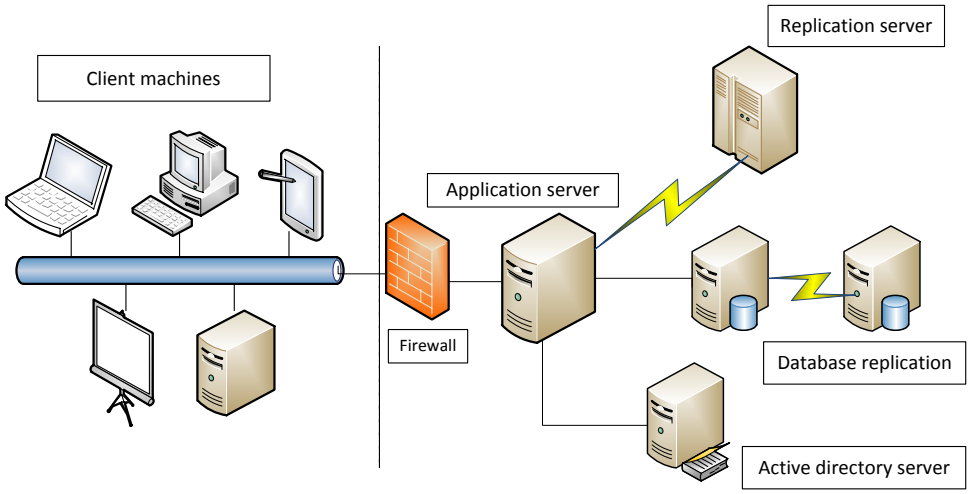

Figure 9. Application backup scenario

Web application Challenges:

The browser security Model is designed to prevent the client from any eventual attacks. We propose to verify the system to report any vulnerability. Below are the most popular checks for detecting gaps in the developed application such as SQL injections, Header Manipulation, Hidden Field Manipulation and Improper Access Controls. These vulnerabilities stem from unchecked input, which is widely recognized as the most common source of security weaknesses.

SQL Injection:

An SQL Injection can destroy a database. Users are allowed to enter their inputs in the dedicated fields in order to display data, this is done mainly by querying the corresponding tables where the information is stored. SQL statements are texts only and with a little piece of code, the database could be dramatically damaged. The data integrity can be affected by adding, modifying and deleting records.

Integrity, roughly speaking, will mean that the correctness of data is ensured: i.e., it can only be established or modified by agents or processes entitled to influence the values of the data [3]. To prevent this to happen, we defined a "blacklist" of words and characters (such as : drop, update, delete, ‘, =,…). Additionally and for more insurance we used SQL parameters which will be added to an SQL query first on the execution time.

Header Manipulation:

Headers are control information delivered from web clients to web servers on HTTP requests or vice-versa on HTTP responses. The manipulation consists of the insertion of malicious data, which has not been validated, into a header through an untrusted source (most frequently an HTTP)

The best solution is to avoid using unsecured communication protocols and protecting the headers cryptographically by implementing the SSL.

Hidden Field Manipulation:

Hidden fields are represented in HTML page as <input type=”hidden”>. Some web applications embed hidden fields to pass state information between the web server and the client side. Hidden fields often contain confidential information that should be stored normally only in a back-end database. Hidden fields aren’t transparent to the end users, but the curious attacker can discover and manipulate them. Unlike regular fields, hidden fields cannot be modified directly by typing values into an HTML form. However, since the hidden field is part of the page source, saving the HTML page, editing the hidden field value, and reloading the page will cause the Web application to receive the newly updated value of the hidden field [13].

To be in the safe side no hidden fields are integrated in the established application. This prevents the attackers from exploiting the weakness in the server's trust of client-side processing by modifying the originally sent data.

Improper Access Controls:

This will introduce security problems when the application doesn’t restrict or inaccurately restricts access to a user from an unauthorized action. Access control mechanism involves the use of several protections like the authentication and should ensure that a resource have the right and correct access. When the access control mechanism fails or not correctly applied, attackers can inappropriately gain privileges and permissions, accessing and manipulating sensitive data, setting a password, etc.

The developed information system is sufficiently tested to prevent attackers from exploiting possible weaknesses in the configuration of access controls or being able to bypass the intended protection. Many scenarios are intensively tested.

III. Results And Discussion

An information system is secure when it fulfills 3 criteria: integrity, confidentiality and availability. Integrity means that data and information cannot be modified without authorization, confidentiality ensures that unauthorized users shouldn’t have access to the data however the availability defined the continued service by preventing the disruption of service.

Though a number of security models, such as electronic signatures and encryption, are currently available to secure the transition of data across applications and sites, the Information systems face several new challenges in the conception, realization and testing.

Also, due to limited time and resources, web software engineers need support in identifying vulnerable code. A practical approach to predicting vulnerable code would enable them to prioritize security auditing efforts [14].

Below are some guidelines on handling data with a web browser (Avoidance techniques):

- The input texts should never be interpreted as a code (HTML Tags, sql requests, scripts, headers…)
- For a better security and performance all inputs need to be validate first in the client side but then also in the server.
- Store and process data using the safest mode.
- Never connect to the database as a root or owner. Use always customized users with only the necessarily privileges.
- Use SQL variables within the statements
- Verify and validate the inputs if the expected data type is entered (Regular Expressions can be used).

The browser security model should satisfy the insurance need, in the most recent browsers, scripts can interact if the two origins are the same if and only if all the following match:

Protocol: (http or https)

Domain: (example.com)

Port number

Content from different domains can’t interact straightforwardly.

1.1 Encoding and decoding:

Encoding is the process of putting a sequence of characters into a specialized format often by using an encryption algorithm (such as MD5 and SHA1 hashes) for a secured transmission and storage. Decoding is the opposite process and will be needed to convert an encoded format back into the original information.

For more security and to maintain the confidentiality, we implemented the encryption of passwords in the database using the SHA1 encryption algorithm (Fig. 10).

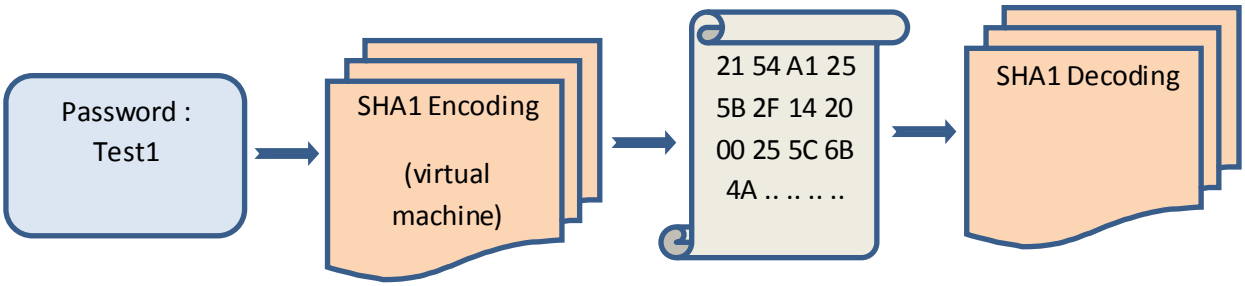

Figure 10 Encoding and Decoding Processes

SHA-1 stands for the Secure Hash Algorithm 1 which is a popular one-way cryptographic hash algorithm. The hash value is 40 digits long and rendered as a hexadecimal number. SHA-1 is also frequently used in environments where the need for data integrity is high. Even if you have access to the tables in the database you won't be able to view the passwords in clear texts.

The users table looks like the following (Fig. 11):

user_id	username	fullname	role	password
2	mali	Malhi Ali	viewer	b444ac06613fc8d63795be9ad0beaf55011936ac
3	Paramedica	Paramedica	consultant	b444ac06613fc8d63795be9ad0beaf55011936ac
4	amalhi	MALHI Abdelghani	admin	d033e22ae348aeb5660fc2140aec35850c4da997
5	arachida	Alaoui Rachida	viewer	40b4f25b1fd956b576d880db2b41182e0444bd1d

Figure 11. Password encryption in the database

1.2 Availability & accessibility:

This is developed based on a simple user sign up form with proper field validations using the regular expressions. After a successful login the user will maintain the logged-in status throughout the different pages in the same session.

After registration the user will receive an email to activate his proper account. He will get registered only and after his confirmation. Beside from that and to ensure more privacy it's possible to reset or update a password to a new one either by the user itself or by the administrator.

As the system is an extensible one and for more security the user's credentials and information can be stored within a separated active directory server. The password can be then more complicated by the use of policy algorithms.

From a security perspective the user's form does the basic checks of the given type and length. More importantly however is the setting of the token in the form itself and as a session variable. This will ensure identifying correctly the form that is being posted and will ensure that it's in fact the truthful form and not a foreign malicious one. In addition to that, it prevents multiple postings so the database is not flooded by somebody hitting the refresh button.

The application availability is calculated based on the percentage of time the software is accessible and available for use.

Availability % = (Uptime ÷ (Uptime + Downtime))*100.

1.3 Efficiency

The goal is to replace the informal methods we used in the past to design and develop applications with precise software and security engineering. The ability to extend, customize and re-design the framework is an enormous advantage. To develop an E-Health information system, the multidisciplinary collaboration is essential. A standard model can be used for general purpose but depends on the use case the system is extensible and it can be customized to fulfill the need.

The achievement to have a networked server platform can help enormously the health care professionals to measure and monitor the daily progress. The application is offering a timely guidance in order to uptake the satisfactory decision by evaluating the entered patient's information [15] [AJASR]. To ensure a high quality and reliability during the realization process, the developers, architects and testers should focus constantly on the following:

- Usability: The ability to make the information system features more friendly and easy to use. On the same time the needs and requirements requested by specified users to achieve specified goals are guaranteed with a high level of effectiveness, efficiency and satisfaction.
- Maintainability: This defines how is it easy and rapid to restore the service to an operational status following a failure and based on prescribed procedures.
- Scalability: The capability of the application to continue functioning well even if it has been increased in size or volume. The objective is to meet the user need and to handle a growing amount of work and tasks or even increase its level of performance and efficiency
- Portability: The web-program can be used in different operating systems and browsers without requiring any major change or rework.

IV. Conclusion

The proposed approach to defend against attacks approves the need to secure the developed health care information system.

Security and privacy are implicit and shouldn't be considered as additional negative costs.

An increased understanding of the nature of vulnerabilities, their manifestations, and the mechanisms that can be used to eliminate and prevent them can be achieved when this will be considered from the beginning of the conception and implementation.

Security reviews generally are executed at the end of Software development lifecycle but in order to obtain high quality software, we did in this approach the focus on quality and testability from the beginning.

That way, product managers, business analysts, developers, testers must behave similarly in the roles.

Bugs can be statically discovered and require a fix in the source code but flaws should be remedied dipper in the architecture level. Often bugs are easier to fix but flaws may require significant engineering to ensure that the design is fixed without causing other problems in the system.

Human, organization and technology are the essential components of Information Systems. Those three evaluation factors can be evaluated throughout the whole system development life cycle namely planning, analysis, design, implementation, operation and maintenance." [16].

To this topic, there are a variety of methods we worked with in order to ensure the security of critical systems.

- Eliminating security flaws, bugs & defects causing vulnerabilities in systems
- Identifying and proactively preventing intrusions from occurring
- Preserving essential services when systems have been penetrated
- Providing decision makers with information required for defense strategy.

In the worst case thanks the replication procedure we will be able to recover in a very short time.

As the system holds sensitive data, a restricted and limited entrance is indispensible; it was ensured in different layers:

- Physical Layer: enforced by restricting the entrance to the server itself (application and / or database). The Operating Systems offer several access control technics. In addition to that the Three-tier Model presents a useful structural approach.
- Restricted access to the application: each user needs his own credential information to login in the software, usually a username and a password. The login request is sent to the database or active directory and based on the response the system will define the permissions and the allowed actions.
- In the web application it's often required to use a secured communication protocols like HTTPS /SSL instead of HTTP.

There are several reasons for placing a high value on caring the privacy, confidentiality, and security of health information system. All these 3 above conditions are fulfilled by the developed applications. This implies a real milestone in the efficiency of the system.

References

[1] Abdelghani El Malhi, Adil Echchelh, Nesma Nekkal, Rassam Ahmed, Abdelrhani Mokhtari, Rachida Soulaymani Bencheikh, Abdelmajid Soulaymani "Modeling of actions to take after a scorpion sting and developing a web based information system to track the different indicators systematically". ESJ - April 2014.

[2] Rakesh Agrawal and Christopher Johnson "Securing electronic health records without impeding the flow of information" International journal of medical informatics Volume 76, Issues 5-6, Pages 471–479 May–June, 2007

[3] Riccardo Focardi & Roberto Gorrieri "Foundations of Security Analysis and Design" ISBN: 978-3-540-42896-1 - Springer - 2001

[4] Simon Rogerson "The Chinook Helicopter Disaster" ETHIcol in the IMIS Journal Volume 12 No 2 (April 2002)

[5] Leveson, Nancy G.; Turner, Clark S. "An Investigation of the Therac-25 Accidents" (PDF). IEEE Computer 26 (7): 18–41. (July 1993).

[6] Statistics Results Page National vulnerability Database (NIST) - Last updated: 1/19/2016 at https://web.nvd.nist.gov/view/vuln/statistics-results?adv_search=true&cves=on

[7] Ivan Victor Krsul "software vulnerability analysis" A Thesis Submitted to the Faculty of Purdue University May 1998

[8] Omar H. Alhazmi, Yashwant K. Malaiya, Colorado State University, "Quantitative Vulnerability Assessment of Systems Software" RAMS 2005 - 0-7803-8824-0/05 ©2005 IEEE
[9] Paul Kocher, Ruby Lee, Gary McGraw, Anand Raghunathan and Srivaths Ravi "Security as a New Dimension in Embedded System Design" Proceedings of the 41st annual Design Automation Conference Pages 753-760 . ISBN:1-58113-828-8 - 2004
[10] Bill Morrow "BYOD security challenges: control and protect your most sensitive data" Network Security Volume 2012, Issue 12, Pages 5–8, December 2012
[11] Zbigniew Baranowski, Lorena Lobato Pardavila, Marcin Blaszczyk, Gancho Dimitrov and Luca Canali "Evolution of Database Replication Technologies for WLCG" Journal of Physics: Conference Series: 664042032, Volume 664, 2015.
[12] Bertino, E. & Sandhu, R. "Database security - concepts, approaches, and challenges" Dependable and Secure Computing, IEEE Transactions (Volume:2 , Issue: 1) - Jan.-March 2005
[13] V. Benjamin Livshits and Monica S. Lam "Finding Security Vulnerabilities in Java Applications with Static Analysis" the Proceedings of the 14th USENIX Security Symposium, July 31–August 5, 2005.
[14] Lwin Khin Shar, Briand, L.C. ; Hee Beng Kuan Tan "Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning" Dependable and Secure Computing, IEEE Transactions on (Volume:12 , Issue: 6) - Novembre 2014
[15] Abdelghani El Malhi, Adil Echchelh, faical EL Hattimy, Abdelrhani Mokhtari, Rachida Soulaymani Bencheikh, Abdelmajid Soulaymani "Conception and creation of a data model to import existing reports about stung patients from spreadsheet into a relational database" AJSR issue Jan 2016
[16] Yusof MM, Kuljis J, Papazafeiropoulou A, Stergioulas LK. Int J Med Inform "An evaluation framework for health information systems: human, organization and technology-fit factors (HOT-fit)." 77(6):386-398 - 2008.