CHAPTER-1

INTRODUCTION

Multi-tenancy in cloud is a double-edged sword that may lead to various security concerns in spite of its transformative contribution to resource optimization. Such security concerns are evident from a wide range of attacks reported in both the literature and the industry. As a result, the accountability and transparency of cloud service providers often become questionable to cloud tenants. To defend against security threats and build the trust of users, verifying security policies using formal verification methods, a.k.a. security auditing, has been a standard practice for years in the industry and is a desirable solution for clouds. However, security auditing in clouds presents several unique challenges. First, the dynamic and self-service nature of clouds means any auditing result may become obsolete very quickly and therefore, a runtime security auditing process is desirable for ensuring continuous protection against security threats. Second, the sheer size and high operational complexity of clouds means a runtime solution must be highly efficient and scalable in order to ensure a practical response time to users. Finally, the co-existence of a large number of tenants and users with different needs implies. That an auditing solution cannot assume users' behaviors to follow any fixed or previously known patterns.

The existing security auditing approaches still fall short to overcome such challenges. First, the retroactive approaches catch violations of security policies after the fact by verifying cloud states (e.g., configurations and logs). As a result, they cannot prevent security breaches from propagating or causing potentially irreversible damages (e.g., leaks of confidential information or denial of service). Second, the intercept-and-check approaches audit the impacts of each change request to the cloud before granting them, which leads to a substantial delay to users' requests. Third, the proactive approaches in verify potential user requests in advance, by assuming a known sequence of user requests, namely, the change plan; however, such an assumption about fixed change plans might not always be realistic, especially considering the diverse and fast evolving needs of cloud tenants and users.

CHAPTER-2

SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The existing security auditing approaches still fall short to overcome such challenges. First, the retroactive approaches catch violations of security policies after the fact by verifying cloud states (e.g., configurations and logs). As a result, they cannot prevent security breaches from propagating or causing potentially irreversible damages (e.g., leaks of confidential information or denial of service). Second, the intercept-and-check approaches audit the impacts of each change request to the cloud before granting them, which leads to a substantial delay to users' requests.

2.1.1 DISADVANTAGES

- Less secure
- Time delay
- Retroactive and can catch violations

2.2 PROPOSED SYSTEM

We present a proactive security auditing system for clouds, namely, ProSAS. First, ProSAS learns a list of cloud events (namely, critical events) that may violate a security policy by utilizing a formal verification method on the cloud state (e.g., logs and configurations). Second, it learns various (e.g., structural, probabilistic and temporal) dependency relationships between cloud events from historical data (e.g., logs). Third, it proactively verifies the future critical events, which are predicted based on their dependency relationships with the current event, against security policies, and prepares a list of allowed parameters (namely, watchlist) for those events. Finally, when a critical event actually occurs, ProSAS utilizes and recycles those verification results to efficiently enforce the security policies.

2.2.1 ADVANTAGES

- Data security
- Less time consuming
- Build trust among cloud tenants and users.
- Efficient, scalable, and adaptable to the diverse and fast-evolving needs of cloud tenants and users.
- It provides a thought-provoking and informative discussion that is relevant to professionals working in the cloud security and auditing domain.

2.3 SYSTEM SPECIFICATION

2.3.1 HARDWARE REQUIRMENTS

Processor	:	i3 and above
RAM	:	2 GB and above
Hard Drive	:	500GB and Above

2.3.2 SOFTWARE REQUIRMENTS

Operating System	: Windows XP/8/10
FRONT END	: PHP
BACK END	: MYSQL
IDE	: WAMP SERVER
Monitor	: 15'' LED
Input Devices	: Keyboard, Mouse

2.3.3 NETWORK SPECIFICATION

Network Card	: Ethernet card
Operating system	: Windows 10
Communication protocol	: TCP/IP, HTTP
Connection Type	: LAN

2. 3.4 PROGRAMMING ENVIRONMENT

Client/server environment

To design and develop the **"ONLINE AUCTION SYSTEM"**, it is essential to understand the client/server model that plays an important role in the concern, which needs the information to be retrieved in a fast and efficient way.

What is Client/Server?

The Client/Server computing model implies a form of processing when requests are submitted by a client or requests the server which processes them and returns the result to the client. The client and the server are two separate logical entities working together over a network to accomplish the task.

Conceptually, the client server architecture can be defined as a special case of Co-operative processing where on entire application is shared between the client and a server system.

Features of client/server computing

- Improved access to information due to internet
- Globalization of information
- Easier maintenance of application and data
- Graphically oriented, high interactive user interface
- Increased developer productivity through ease of tools

In our project we have divided core part into two parts. Asp pages, html pages are used as user interface (client). They gather the information from the user and process them. Ms.Access is stored in IIS, which is used as server.

Installation requirements

When installing web development to a hard drive other than ordinary PC, one need to have at least 65-70MB free space on a drive to precede installation, regardless of how much space is on installation drive.

2.3.5 SYSTEM FEATURES

Operating system : Windows 10

Web server : All OS Apache,Mysql,Php,Perl (XAMPP)

WINDOWS

WINDOWS is a powerful multitasking operating system with high security. It is user friendly and supports multithreading and lot of tools for developing in any application. This OS has number of enhancements, including performance improvement, better hardware support and closer integration with the Internet. Windows support dynamic linking. This OS has the concept of plug and play.

WEB SERVER

The Web server accepts the request and sends the HTML to the Client browser that requests it. Web browser and web server communicate through a common protocol (HTTP). The examples for web server are XAMPP(any of four different operating systems,**A**pache,**M**ySQL,**P**hp,**P**erl),WAMP(**W**indows,**A**pache,**M**ySQL,**P**hp),MAMP(**M**acintosh, **A**pache, **M**ySQL, **P**HP).

PHP

PHP stands for Hypertext Preprocessor. PHP scripts run inside Apache server or Microsoft IIS. PHP and Apache server are free. PHP code is very easy. PHP is the most used server side

scripting language. PHP files contain PHP scripts and HTML. PHP files have the extension “php”, “php3”, “php4”, or “phtml”.

Using PHP

- Generate dynamic web pages. PHP can display different content to different user or display different content at different times of the day.
- Process the contents of HTML forms. We can use a PHP to retrieve and respond to the data entered into an HTML form.
- Can create database-driven web pages. A PHP can insert new data or retrieve existing data from a database such a MySQL.

Working of PHP

PHP is a standard HTML file that is extended with additional features. Like a standard HTML file, PHP contains HTML tag that can be interpreted and displayed by a web browser. Anything we could normally place in an HTML file Java applets, Blinking text, server side scripts .we can place in PHP. However, PHP has three important features that make it unique.

- PHP contains server side scripts.
- PHP provides several built-in objects.

HYPER TEXT MARKUP LANGUAGE (HTML)

HTML is an application of the Standard Generalized Markup Language (SGML), which was approved as an international standard in the year 1986. SGML provides a way to encode hyper documents so they can be interchanged.

SGML is also a Meta language for formally describing document markup system. Infact HTML uses SGML to define a language that describes a WWW hyper document’s structure and inter connectivity.

Following the rigors of SGML, TBL bore HTML to the world in 1990. Since then, many of us have it to be easy to use but sometimes quite limiting. These limiting factors are

being addressed but the World Wide Web Consortium (aka W3c) at MIT. But HTML had to start somewhere, and its success argues that it didn't start out too badly.

MYSQL

MySQL Server is a powerful database management system and the user can create application that requires little or no programming. It supports GUI features and an entire programming language, Phpmyadmin which can be used to develop richer and more developed application. There are quite a few reasons, the first being that MySQL is a feature rich program that can handle any database related task you have. You can create places to store your data build tools that make it easy to read and modify your database contents, and ask questions of your data. MySQL is a relational database, a database that stores information about related objects. In MySQL that database means a collection of tables that hold data. It collectively stores all the other related objects such as queries, forms and reports that are used to implement function effectively.

The MySQL database can act as a back end database for PHP as a front end, MySQL supports the user with its powerful database management functions. A beginner can create his/her own database very simply by some mouse clicks. Another good reason to use MySQL as back end tool is that it is a component of the overwhelmingly popular Open source software.

Database

A database is simply a collection of used data just like phone book. MySQL database include such objects as tables, queries, forms, and more.

Tables

In MySQL tables are collection of similar data. With all tables can be organized differently, and contain mostly different information- but they should all be in the same database file. For instance we may have a database file called video store. Containing tables

named members, tapes, reservations and so on. These tables are stored in the same database file because they are often used together to create reports to help to fill out on screen forms.

Relational database

MySQL is a relational database. Relational databases tools like access can help us manage information in three important ways.

- Reduce redundancy
- Facilitate the sharing of information
- Keep data accurate.

Fields

Fields are places in a table where we store individual chunks of information.

Primary key and other indexed fields

MySQL use key fields and indexing to help speed many database operations. We can tell MySQL, which should be key fields, or MySQL can assign them automatically.

CHAPTER-3

SYSTEM DESIGN AND DEVELOPMENT

3.1 INPUT DESIGN

Input design is the process of converting the user-oriented. Input to a computer based format. The goal of the input design is to make the data entry easier, logical and free error. Errors in the input data are controlled by the input design. The quality of the input determines the quality of the system output.

The entire data entry screen is interactive in nature, so that the user can directly enter into data according to the prompted messages. The users are also can directly enter into data according to the prompted messages. The users are also provided with option of selecting an appropriate input from a list of values. This will reduce the number of error, which are otherwise likely to arise if they were to be entered by the user itself.

Input design is one of the most important phases of the system design. Input design is the process where the input received in the system are planned and designed, so as to get necessary information from the user, eliminating the information that is not required. The aim of the input design is to ensure the maximum possible levels of accuracy and also ensures that the input is accessible that understood by the user.

The input design is the part of overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify the errors.

The objectives considered during input design are:

- Nature of input processing.
- Flexibility and thoroughness of validation rules.
- Handling of properties within the input documents.
- Screen design to ensure accuracy and efficiency of the input relationship with files.

- Careful design of the input also involves attention to error handling, controls, batching and validation procedures.

Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of erroneous information.

3.2 OUTPUT DESIGN

Output design is very important concept in the computerized system, without reliable output the user may feel the entire system is unnecessary and avoids using it. The proper output design is important in any system and facilitates effective decision-making. The output design of this system includes various reports.

Computer output is the most important and direct source of information the user. Efficient, intelligible output design should improve the system's relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the data flow diagram. Human factors reduce issues for design involved addressing internal controls to ensure readability.

An application is successful only when it can provide efficient and effective reports. Reports are actually presentable form of the data. The report generation should be useful to the management for future reference. The reports are the main source of information for user's operators and management. Report generated are a permanent record of the transaction occurred. After any valid transactions; have commenced the report of the same are generations and: filed for future reference. Great care has been taken when designation the report as it plays an important role in decision-marking.

3.3 DATABASE DESIGN

A well database is essential for the good performance of the system .several tables are referenced or manipulated at various instance. The table also knows as relation; provide information pertaining to a specified entity. Normalization of table is carried out to extent

possible, while the normalizing tables, care should be taken to make sure that the number of tables do not exceed the optimum level, so that table maintenance. Is convenient and effective

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Not all of these steps will be necessary in all cases. Usually, the designer must:

- Determine the data to be stored in the database
- Determine the relationships between the different data elements
- Superimpose a logical structure upon the data on the basis of these relationships.

Within the relational model the final step can generally be broken down into two further steps that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an Object database.

In a majority of cases, the person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system.

3.4 SYSTEM DEVELOPMENT

3.4.1 DESCRIPTION OF MODULES

MODULES USED

- **TPA (Trusted Party Auditor)**
- **CSP (Cloud Service Provider)**

TPA (Trusted Party Auditor)

Here TPA has to login by using their unique user name and password. TPA is the only authorized person to access tpa module for security purpose. So others don't get rights to access this module.

Transfer:

In this module tpa view the client uploaded file and transfer them into cloud. tpa is the only authorized person to access tpa module for security purpose. So others don't get rights to access this module.

View:

In this module tpa view the client uploaded file from cloud. tpa is the only authorized person to access tpa module for security purpose. So others don't get rights to access this module.

CSP (Cloud Service Provider)

Here CSP has to login by using their unique user name and password. CSP is the only authorized person to access ttp module for security purpose. So others don't get rights to access this module.

Home View:

In this module csp view the client uploaded file in their cloud as encrypted format. If csp try edit the client file the alert will send to ttp. CSP is the only authorized person to access ttp module for security purpose. So others don't get rights to access this module.

Client:

Here client has to login by using their unique user name and password after registration. Client is the only authorized person to access this module for security purpose. So others don't get rights to access this module.

Upload:

In this module client upload their files what are all they want to store in cloud. Client is the only authorized person to access this module for security purpose. So others don't get rights to access this module.

Result View:

In this module client view their uploaded file from cloud. Client is the only authorized person to access this module for security purpose. So others don't get rights to access this module.

CHAPTER-4

TESTING AND IMPLEMENTATION

SYSTEM TESTING

System testing involves testing the entire system as a whole to ensure that all components are working together correctly. Here's an example of a system testing plan for the code:

- Verify that the application can be installed and configured correctly on the target environment.
- Test the application's ability to handle various input formats, including valid and invalid formats, to ensure that the application can handle all possible user input scenarios.
- Verify that the application can properly read and parse the input files, and that all data is processed correctly.
- Test the application's ability to generate accurate output files and that the data is formatted correctly.
- Verify that the application is scalable, and can handle large volumes of data without crashing or slowing down.
- Test the application's error-handling capabilities, including error logging, reporting, and recovery.

By performing these tests, you can ensure that the application is thoroughly tested and ready for deployment.

Types of Testing

- Unit testing
- Integration testing
- Validation testing
- Output testing
- User acceptance testing

Unit Testing

Unit testing is a process of testing individual units or components of a software application to ensure that each unit is working as expected. It involves writing test cases that verify the expected behavior of each unit of code.

The unit tests assume that the functions get_files_from_cloud() and send_request_for_file_key() are defined in the cloud_files.php file. You'll need to modify the test cases to match the actual implementation of the code being tested. Additionally, these tests only cover a few specific cases and don't provide complete test coverage for the code. You'll want to write additional tests to ensure that all of the functionality of the code is being tested

Integration Testing:

Integration testing involves testing how different parts of the system work together.The entire project was split into small program; each of this single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructors were used. It give a lot of problem by not functioning is an integrated manner.

We can also perform end-to-end testing to test the entire system, from input to output. This can involve manually testing the application using different input values and verifying that the output is correct. The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. when the frames where given for the test, the end user gave suggestion. Based on their suggestions the frames where modified and put into practice.

Validation Testing

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., Validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

- Verify that the function returns a list of integers for a valid input of a list of strings containing valid integers.
- Verify that the function raises a TypeError for an invalid input of a non-list type
- Verify that the function raises a ValueError for an invalid input of a list containing non-integer strings.
- Verify that the function returns an empty list for an empty input list.

Output Testing

Output testing is a type of testing that involves verifying that the output produced by the system under test matches the expected output. In the case of the given code, the expected output is the result of the calculation performed by the calculate function.

To perform output testing, we can write test cases that call the calculate function with different inputs and compare the output produced by the function with the expected output.

User Acceptance System

User Acceptance Testing (UAT) is a type of testing that is performed by the end-users to ensure that the system meets their requirements and expectations. An acceptance test as the objective of selling the user on validity and reliability of the system. It verifies that the procedures operate to system specification and mat the integrity of vital is maintained.

Performance Testing

Performance testing is important to ensure that the application meets the performance requirements and can handle a high load of users or requests.This project is a application based project, and the modules are interdependent with the other modules, so the testing cannot be done module by module. So the unit testing is not possible in the case of this driver. So this system is checked only with their performance to check their quality.

IMPLEMENTATION

It making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the Consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system *development* to a system *support and maintenance* mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

The process of implementing the above code can vary depending on the specific requirements and constraints of the project. However, a general outline of the implementation process might look like:

1. Requirements gathering: The first step in implementing any code is to gather the requirements. This includes understanding the problem to be solved, identifying the inputs and outputs, and determining any constraints or limitations on the solution.

2. Design: Based on the requirements, a design for the code should be created. This includes selecting the programming language and tools to be used, determining the overall architecture of the system, and defining the interfaces between different components.

3. Coding: Once the design is in place, the actual coding can begin. This involves writing the code that implements the various functions and features required by the system.

4. Testing: After the code has been written, it must be thoroughly tested to ensure that it works as intended. This includes unit testing, integration testing, validation testing, output testing, and performance testing, as described in the previous questions.

5. Deployment: Once the code has been thoroughly tested and any bugs or issues have been addressed, it can be deployed to the production environment. This involves

ensuring that the system is properly configured, that all necessary dependencies and libraries are installed, and that any necessary security measures are in place.

6. Maintenance: After the code has been deployed, it must be regularly maintained to ensure that it continues to function properly. This includes monitoring the system for errors or issues, addressing any bugs that arise, and making any necessary updates or modifications to keep the system up-to-date and secure.

The purpose of **Prepare for System Implementation** is to take all possible steps to ensure that the upcoming system deployment and transition occurs smoothly, efficiently, and flawlessly. In the implementation of any new system, it is necessary to ensure that the Consumer community is best positioned to utilize the system once deployment efforts have been validated. Therefore, all necessary training activities must be scheduled and coordinated. As this training is often the first exposure to the system for many individuals, it should be conducted as professionally and competently as possible. A positive training experience is a great first step towards Customer acceptance of the system.

During System Implementation it is essential that everyone involved be absolutely synchronized with the deployment plan and with each other. Often the performance of deployment efforts impacts many of the Performing Organization's normal business operations. Examples of these impacts include:

- Consumers may experience a period of time in which the systems that they depend on to perform their jobs are temporarily unavailable to them. They may be asked to maintain detailed manual records or logs of business functions that they perform to be entered into the new system once it is operational.

- Technical Services personnel may be required to assume significant implementation responsibilities while at the same time having to continue current levels of service on other critical business systems.

- Technical Support personnel may experience unusually high volumes of support requests due to the possible disruption of day-to-day processing.

Because of these and other impacts, the communication of planned deployment activities to all parties involved in the project is critical. A smooth deployment requires strong leadership, planning, and communications. By this point in the project lifecycle, the team will

have spent countless hours devising and refining the steps to be followed. During this preparation process the Project Manager must verify that all conditions that must be met prior to initiating deployment activities have been met, and that the final 'green light' is on for the team to proceed. The final process within the System Development Lifecycle is to transition ownership of the system support responsibilities to the Performing Organization. In order for there to be an efficient and effective transition, the Project Manager should make sure that all involved parties are aware of the transition plan, the timing of the various transition activities, and their role in its execution.

Due to the number of project participants in this phase of the SDLC, many of the necessary conditions and activities may be beyond the direct control of the Project Manager. Consequently, all Project Team members with roles in the implementation efforts must understand the plan, acknowledge their responsibilities, recognize the extent to which other implementation efforts are dependent upon them, and confirm their commitment.

CHAPTER 5

CONCLUSION

The continuous auditing with scalability and practical response time is important to both cloud providers and their tenants. In this paper, we proposed a proactive security auditing system, namely, ProSAS, which significantly reduces the response time and enforces the auditing results on the cloud before any violation can take effect. More specifically, ProSAS first established its models (e.g., dependency model and critical events). Second, it conducted proactive security verification by leveraging its models. Finally, it utilized those verification results to enforce security on the cloud at runtime. We integrated ProSAS to OpenStack, one of the most popular cloud management platforms, and provided guidelines to port it to other major cloud platforms. Furthermore, we evaluated the efficiency and accuracy of our method, and showed that the response time is reduced to a practical level. However, there exist several limitations in ProSAS, which we consider as future works. First, the current method of learning critical events needs a manual inspection, which could be further automated by using machine learning techniques to select the final candidate. Second, a single-step violation is not yet efficiently handled in ProSAS. An efficient runtime approach might help to address this concern.

BIBLIOGRAPHY

PHP/MYSQL REFERENCE BOOKS

1. Beginning PHP 5.3 - **Matt Doyle** Publication, first Edition, (October 26, 2009).
2. Expert PHP and MySQL - **Andrew Curioso, Ronald Bradford, Patrick Galbraith** Publication, Fourth Edition,2010
3. Beginning Php and Mysql **–W.Jason Gilmore** Publication, Fourth Edition,2010
4. PHP and MySQL 24-Hour Trainer - **Andrea Tarr** Publication, Second Edition,2011
5. Web Database Applications with PHP & MySQL, - Hugh E. Williams (Author), David Lane Publication, Second Edition,2009

References

[1] J. Aikat, A. Akella, J. S. Chase, A. Juels, M. Reiter, T. Ristenpart, V. Sekar, and M. Swift, "Rethinking security in the era of cloud computing," IEEE Security & Privacy, vol. 15, no. 3, 2017.

[2] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[3] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in ACM CCS. ACM, 2009.

[4] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-tenant sidechannel attacks in PaaS clouds," in ACM CCS, 2014.

[5] Z. Xu, H. Wang, and Z. Wu, "A measurement study on co-residence threat inside the cloud." in USENIX Security Symposium, 2015.

[6] OpenStack, "Nova network security group changes are not applied to running instances," 2015, available at: https://security.openstack.org/ossa/ OSSA-2015-021.html, last accessed on: February 14, 2018.

[7] “Neutron security groups bypass through invalid CIDR,” 2015, available at: https://security.openstack.org/ossa/OSSA-2014-014.html, last accessed on: February 14, 2018.

[8] Deloitte, “Cybersecurity and the role of internal audit,” 2019, available at: https://www2.deloitte.com/us/en/pages/risk/articles/ cybersecurity-internal-audit-role.html.

[9] KPMG, “Internal audit risk & compliance services,” 2019, available at: https://home.kpmg/xx/en/home/services/advisory/risk-consulting/ internal-audit-risk.html.

[10] T. Madi, S. Majumdar, Y. Wang, Y. Jarraya, M. Pourzandi, and L. Wang, “Auditing security compliance of the virtualized infrastructure in the cloud: Application to OpenStack,” in ACM CODASPY, 2016.

PHP/MYSQL MYSQL REFERENCE SITES

http://www.w3schools.com

http://www.tuxradar.com/practicalphp

http://phpbuddy.com/index.php

http://www.daniweb.com

http://www.pscode.com

http://dev.mysql.com

http://www.mysqltutorial.org/

www.hotscripts.com

www.freesoft.in

APPENDICES

A. DATA FLOW DIAGRAM

LEVEL: 0

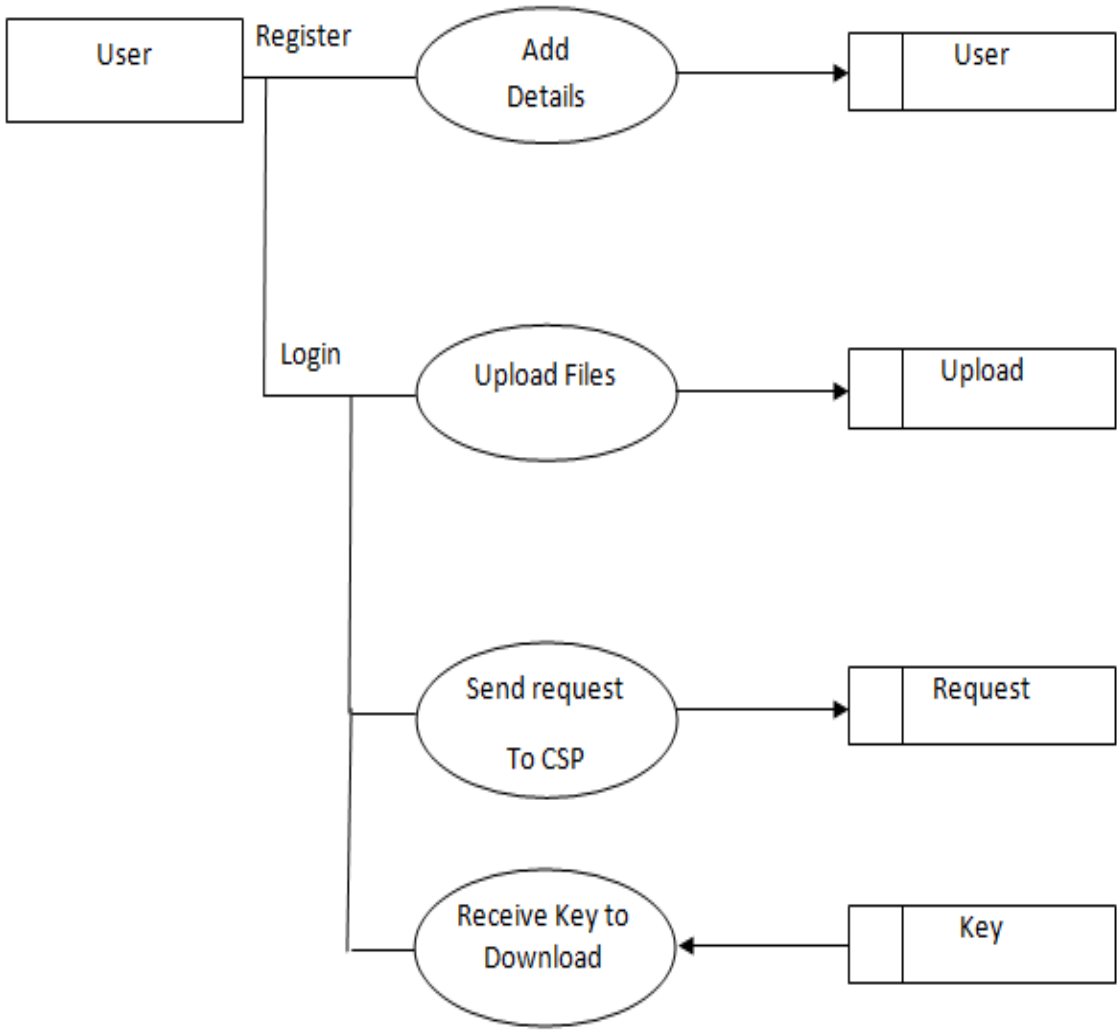

Fig A 5.1 USER LOG

LEVEL:1

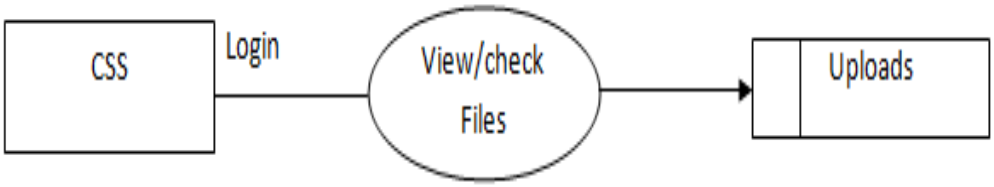

Fig CSS LOGIN

LEVEL:2

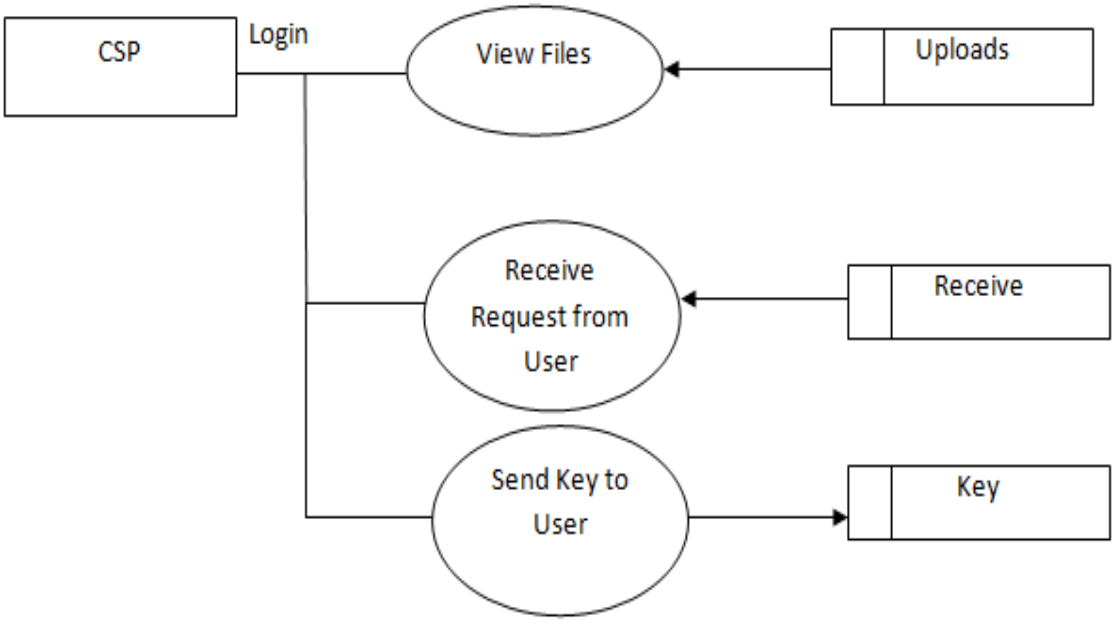

Fig A 5.2 CSP LOGIN

B.SAMPLE CODING

```
<?php
session_start();
include_once "../db/db.php";
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title><?php include_once('../title.php');?></title>
<link rel="stylesheet" type="text/css" href="../style/style.css">
</head>
<body>
 <p>
   <?php include('../header.php');?>
  </p>
<nav id="nav01"></nav>
<div id="main">
<form action="" method="post" enctype="multipart/form-data">
<h3>Key for Requested Files </h3>
```

```
<table border="1" cellpadding="5px" cellspacing="5px" width="100%" class="sortable" align="center" style="background-color:rgba(51,102,153,0.6); color:#7FFF55">
        <thead style="text-transform:uppercase;">
          <tr>
            <th>File ID</th>
            <th>File Name</th>
            <th>Checksum</th>
            <th>Key</th>
              <th>Action</th>
                  </tr>
        </thead>
        <tbody align="center">
<?php
    $result = mysql_query("SELECT * FROM upload where ctskey='yes'",$con);
    while ($row = mysql_fetch_array($result,MYSQL_ASSOC)){
    ?>
    <tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row['file_name']; ?></td>
    <td><?php echo $row['checksum']; ?></td>
        <td><?php echo $row['key']; ?></td>
        <td><a href='download1.php?nama=<?php echo $row['file_name'];?>&path=<?php echo $row['file_data'];?>'>download</a></td>
```

```
            </tr>
                    <?php
    }
?>
</tbody>
</table>
</form>
</div>
<p>
   <?php include('../footer.php');    ?>
  </p>
  <script src="script.js"></script>
</body>
</html>
<?php
session_start();
include_once "../db/db.php";
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title><?php include_once('../title.php');?></title>
<link rel="stylesheet" type="text/css" href="../style/style.css">
</head>
<body>
 <p>
   <?php include('../header.php');?>
  </p>
<nav id="nav01"></nav>
<div id="main">
<form action="" method="get" enctype="multipart/form-data">
<h3>FILES IN CLOUD </h3>
  <table border="1" cellpadding="5px" cellspacing="5px" width="100%" class="sortable" align="center" style="background-color:rgba(51,102,153,0.6); color:#7FFF55">
                    <thead style="text-transform:uppercase;">
                       <tr>
                          <th>File ID</th>
                          <th>File Name</th>
                          <th>File Type</th>
                          <th>Checksum</th>
                          <th>File Path</th>
                         <th>File Owner</th>
                            <th>Date</th>
                            <th>Key Request</th>
```

```
                </tr>
            </thead>
            <tbody align="center">
<?php
    $result = mysql_query("SELECT * FROM upload where status='Accepted' and company='".$_SESSION['company']."'",$con);
    while ($row = mysql_fetch_array($result,MYSQL_ASSOC)){
    ?>
    <tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row['file_name']; ?></td>
    <td><?php echo $row['file_type']; ?></td>
    <td><?php echo $row['checksum']; ?></td>
        <td><?php echo $row['file_data']; ?></td>
        <td><?php echo $row['user_name']; ?></td>
    <td><?php echo $row['date']; ?></td>
        <td><a href='download.php?fid=<?php echo $row['id'];?>'><b>Send Request</b></a></td>
        </tr>
            <?php
    }
?>
</tbody>
```

```
</table>
<?php
if(isset($_GET['fid'])) {
$req=$_GET['fid'];
mysql_query("UPDATE upload SET request='yes' WHERE id='".$req."'", $con) or die();
echo "<script type='text/javascript'>alert('Request Has Been Sent');</script>";
}
?>
</form>
</div>
<p>
   <?php include('../footer.php');    ?>
  </p>
  <script src="script.js"></script>
</body>
</html>
<?php
session_start();
include_once "../db/db.php";
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title><?php include_once('../title.php');?></title>
<link rel="stylesheet" type="text/css" href="../style/style.css">
</head>
<body>
 <p>
   <?php include('../header.php');?>
 </p>
<nav id="nav01"></nav>
<div id="main">
<form action="" method="post" enctype="multipart/form-data">
<h3>FILES IN CLOUD </h3>
  <table border="1" cellpadding="5px" cellspacing="5px" width="100%" class="sortable" align="center" style="background-color:rgba(51,102,153,0.6); color:#7FFF55">
                    <thead style="text-transform:uppercase;">
                       <tr>
                          <th>File ID</th>
                          <th>File Name</th>
                          <th>File Type</th>
                          <th>checksum</th>
                          <th>File Path</th>
```

```
                    <th>File Owner</th>
                <th>Date</th>
                <th>Action</th>
                  </tr>
                </thead>
                <tbody align="center">
<?php
    $result = mysql_query("SELECT * FROM upload where status='Accepted'",$con);
    while ($row = mysql_fetch_array($result,MYSQL_ASSOC)){
    ?>
    <tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row['file_name']; ?></td>
    <td><?php echo $row['file_type']; ?></td>
    <td><?php echo $row['checksum']; ?></td>
        <td><?php echo $row['file_data']; ?></td>
        <td><?php echo $row['user_name']; ?></td>
    <td><?php echo $row['date']; ?></td>
    <td><a href="<?php echo $row['file_data']; ?>">view</a></td>
    </tr>
                <?php
    }
?>
```

```
</tbody>
</table>
</form>
</div>
<p>
    <?php include('../footer.php');    ?>
  </p>
  <script src="script.js"></script>
</body>
</html>
```

C. SAMPLE FORMS

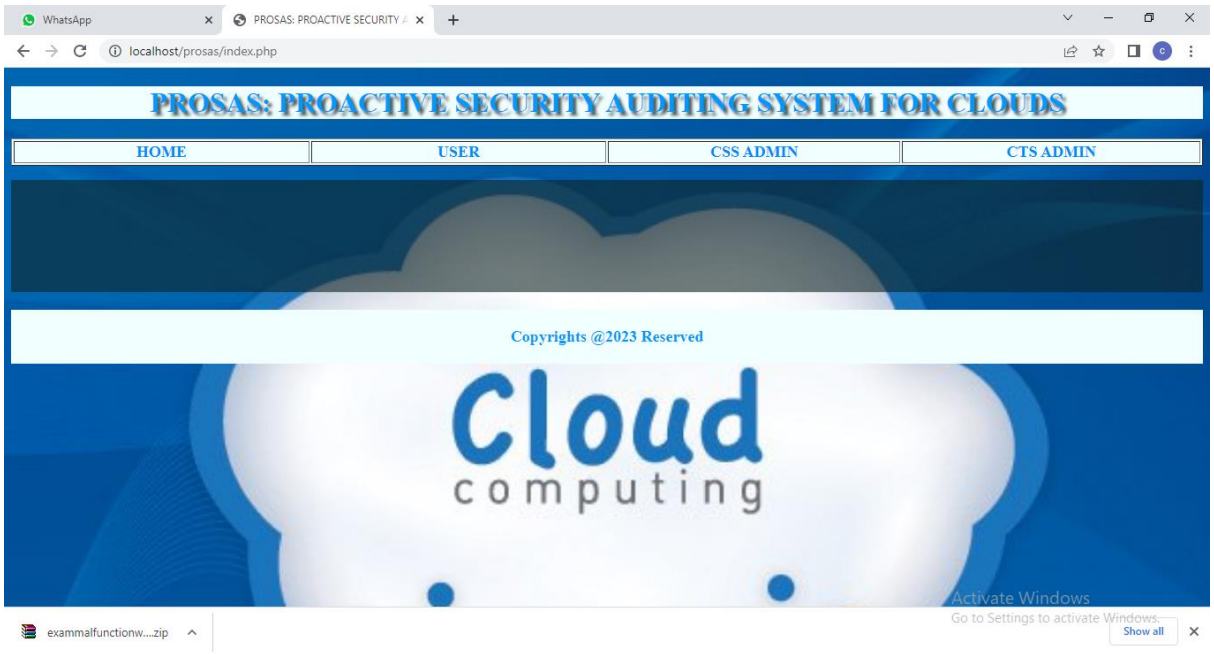

Fig C 5.1

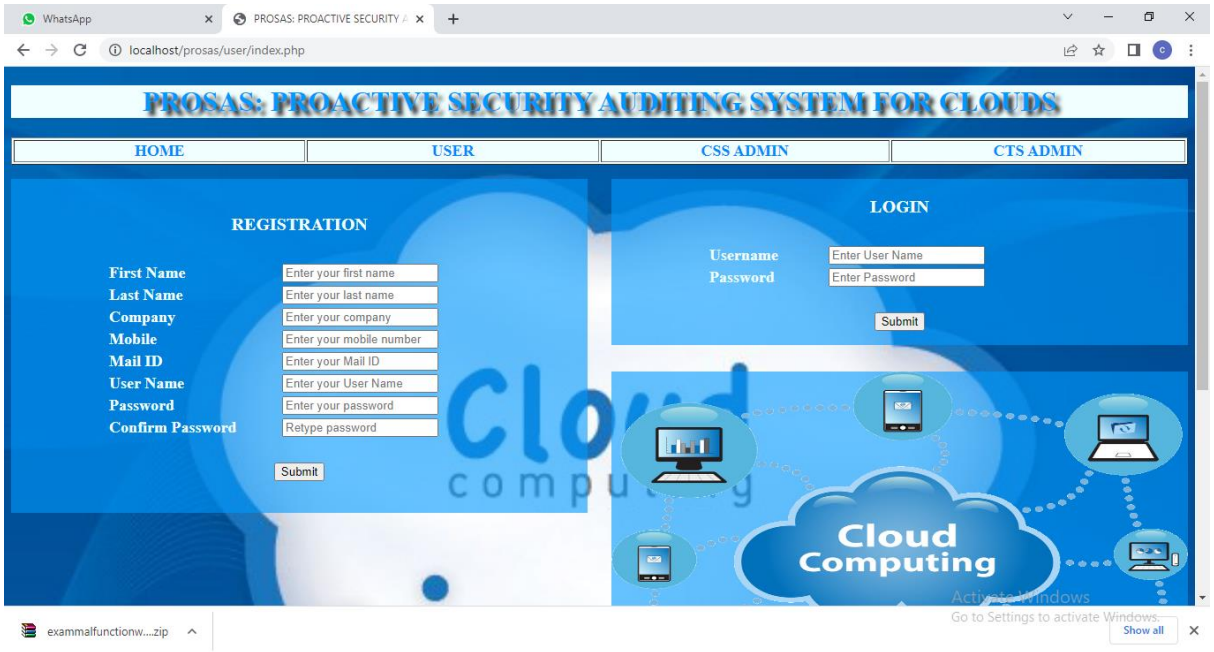

Fig C 5.2

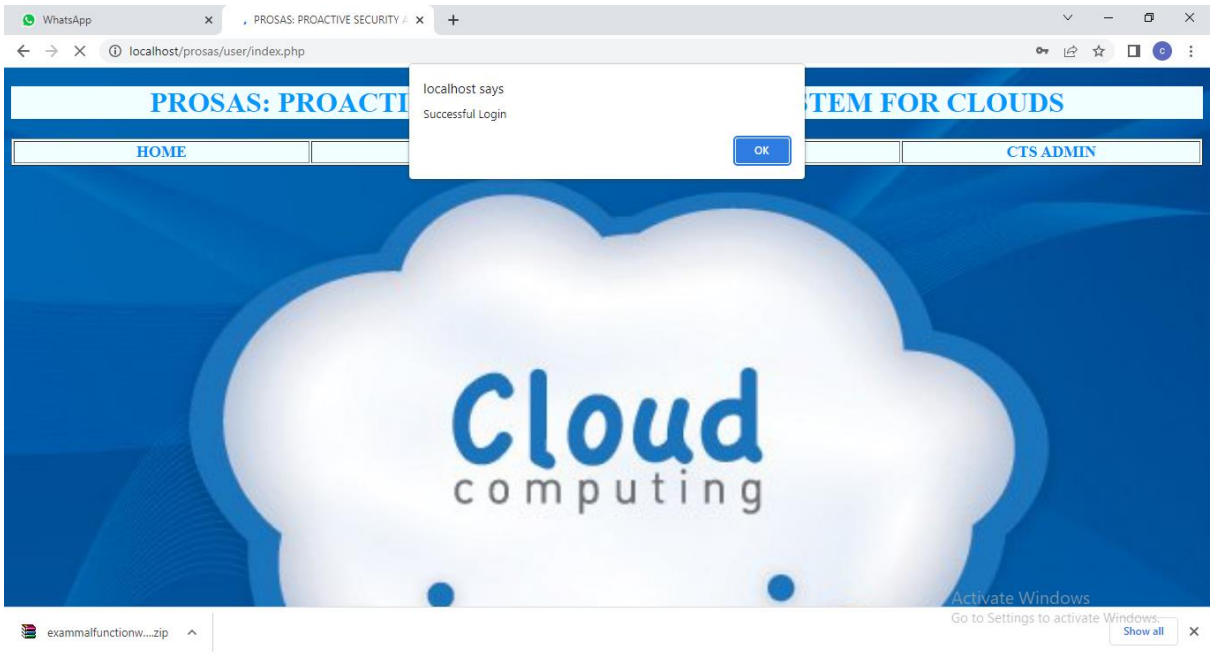

Fig C 5.3

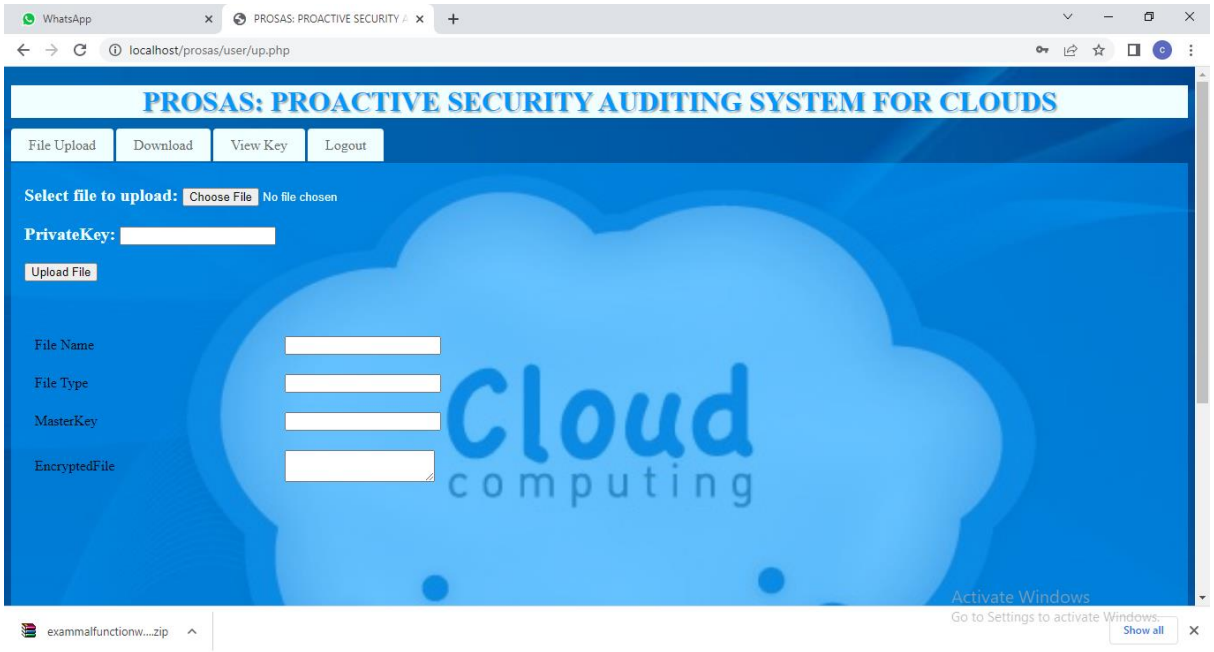

Fig C 5.4

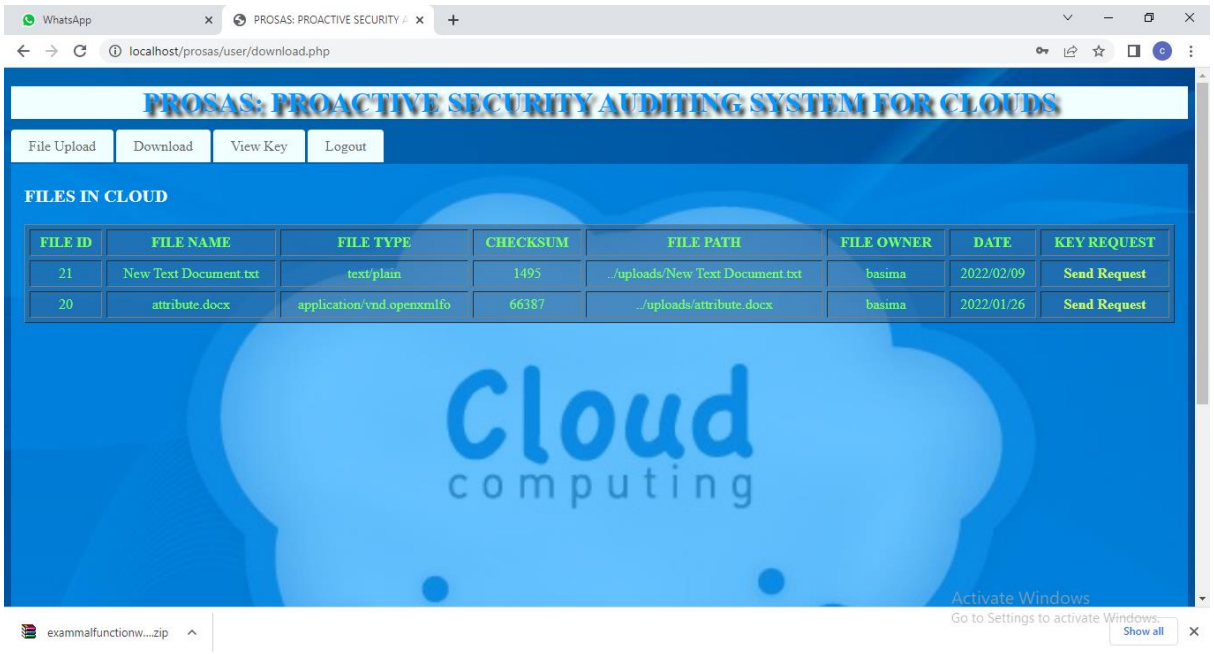

Fig C 5.5

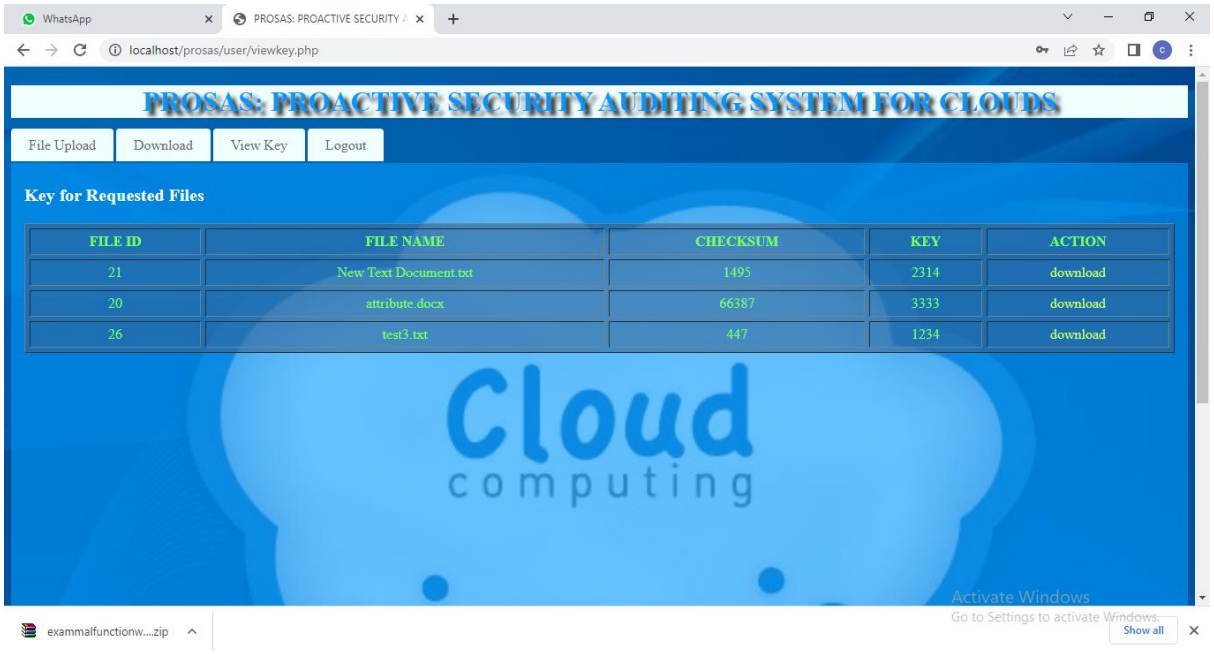

Fig C 5.6

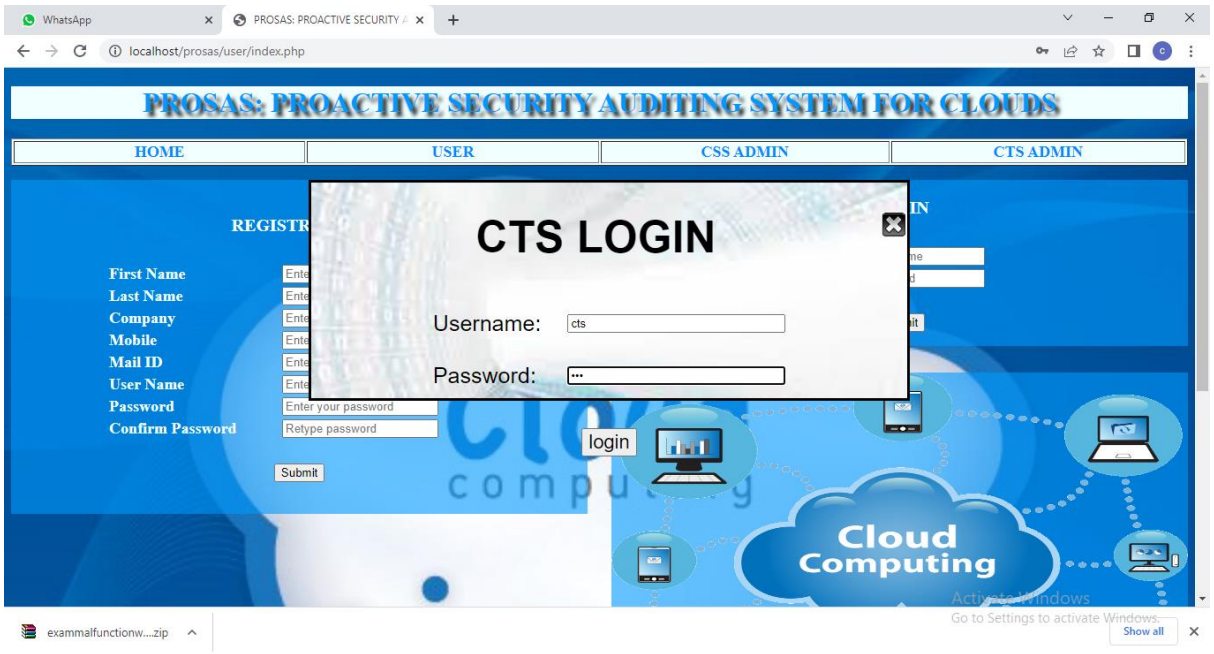

Fig C 5.7

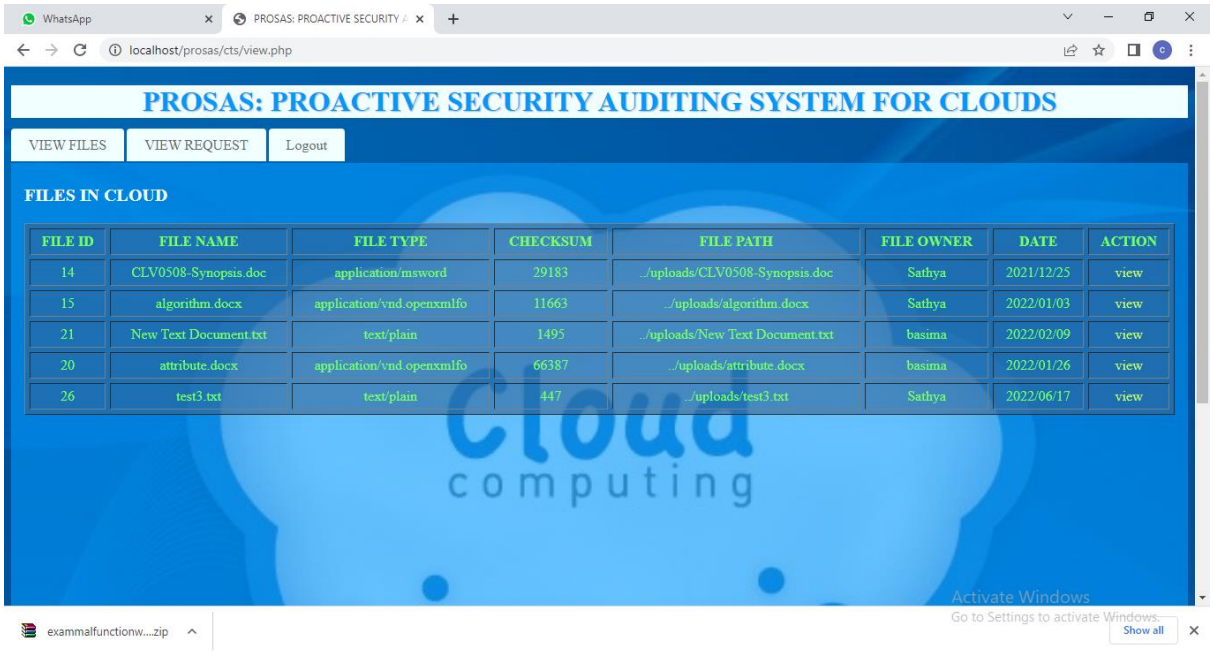

Fig C 5.8

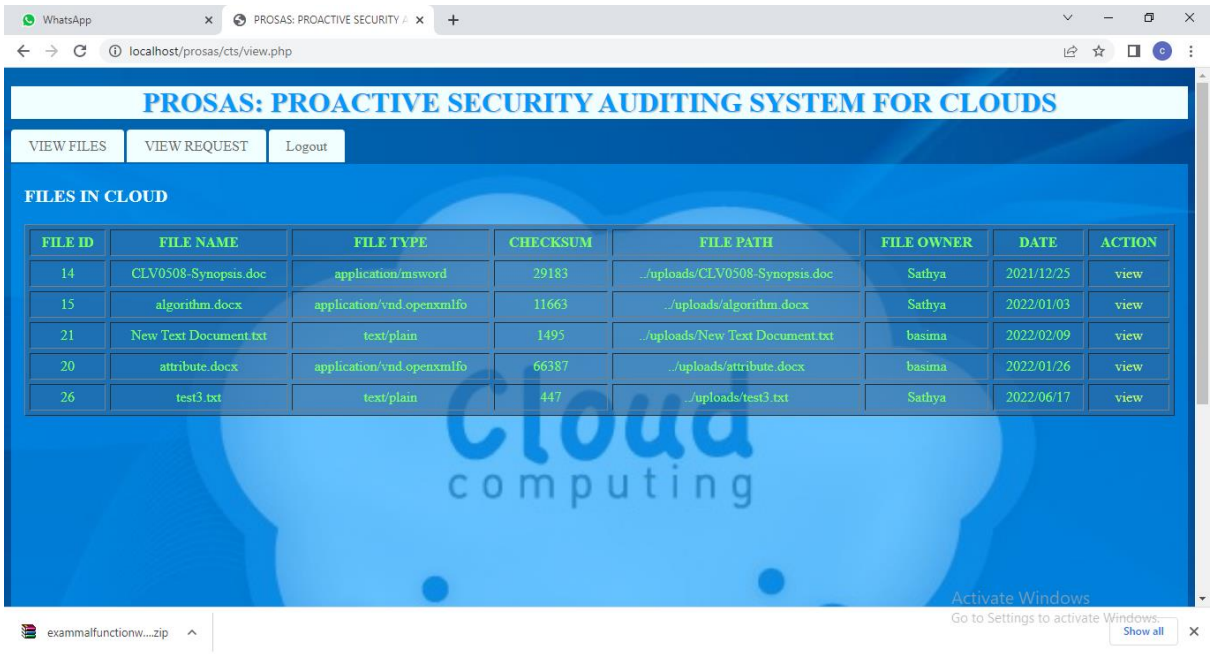

Fig C 5.9

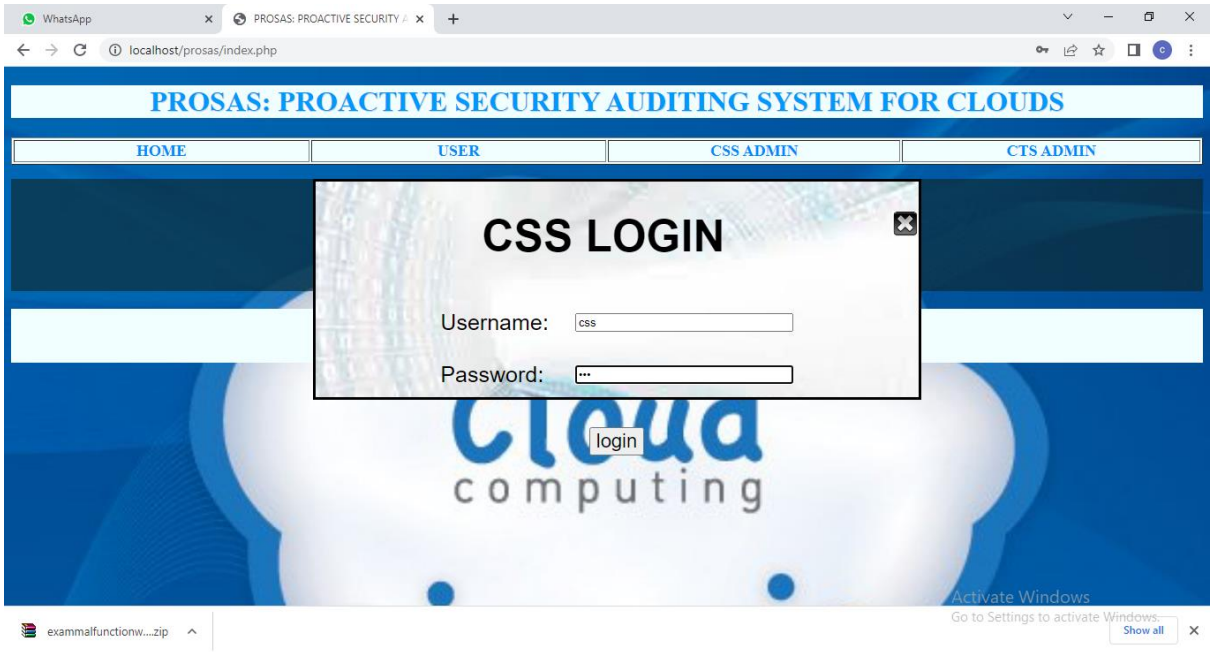

Fig C 5.10

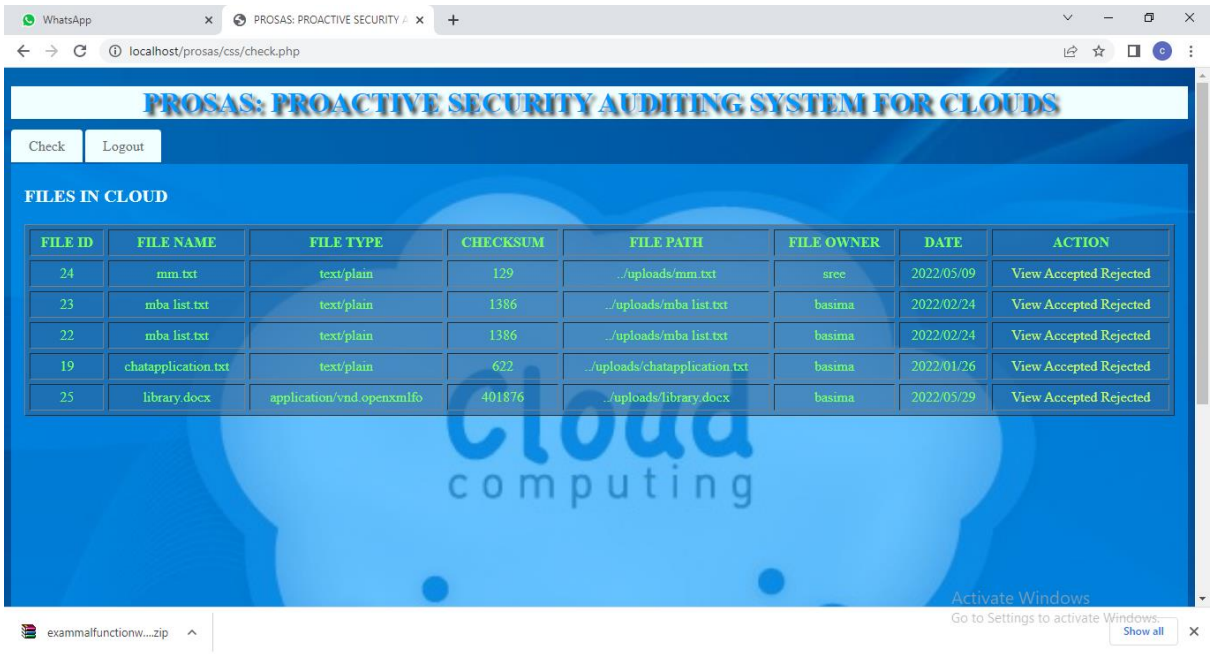
WhatsApp
PROSAS: PROACTIVE SECURITY
localhost/prosas/css/check.php
PROSAS: PROACTIVE SECURITY AUDITING SYSTEM FOR CLOUDS
Check
Logout
FILES IN CLOUD
FILE ID | FILE NAME | FILE TYPE | CHECKSUM | FILE PATH | FILE OWNER | DATE | ACTION
24 | mm.txt | text/plain | 129 | ../uploads/mm.txt | sree | 2022/05/09 | View Accepted Rejected
23 | mba list.txt | text/plain | 1386 | ../uploads/mba list.txt | basima | 2022/02/24 | View Accepted Rejected
22 | mba list.txt | text/plain | 1386 | ../uploads/mba list.txt | basima | 2022/02/24 | View Accepted Rejected
19 | chatapplication.txt | text/plain | 622 | ../uploads/chatapplication.txt | basima | 2022/01/26 | View Accepted Rejected
25 | library.docx | application/vnd.openxmlfo | 401876 | ../uploads/library.docx | basima | 2022/05/29 | View Accepted Rejected
cloud computing
Activate Windows
Go to Settings to activate Windows.
exammalfunctionw....zip
Show all

Fig C 5.11