Transient Authentication for Cloud Security

Rishikesh Shukla, Atul Dobhal, Ankit Rai, Ankur Srivastava
Computer Department, Bharati Vidhyapeeth College of Engineering Lavale Pune, Maharashtra, India

ABSTRACT

Cloud Computing becomes the next generation architecture of IT Enterprise. In contrast to traditional solutions, Cloud computing moves the application software and databases to the large data centres, where the management of the data and services may not be fully trustworthy. This unique feature, however, raises many new security challenges which have not been well understood. In cloud computing, both data and software are fully not contained on the user's computer; Data Security concerns arises because both user data and program are residing in Provider Premises. Clouds typically have single security architecture but have many customers with different demands. Every cloud provider solves this problem by encrypting the data by using encryption algorithms. Even multi-tenancy feature of cloud give rise to security issue related to data and also the authentication of users. We provide a secure scheme which protects the sensitive data of the user stored in the cloud. We solve the problem which existing systems faces with Transient Authentication, in which a small hardware token (Mobile Phone) continuously authenticates the user’s presence over a short-range, wireless link to the machine through which the user is accessing cloud. This machine in turn provides this authentication information to cloud. When the user departs, the token and machine lose contact and the machine stops exchanging the information with cloud making the data secure. We show how to leverage this authentication framework to secure sensitive and confidential data on to the cloud.

Keywords: Cloud Computing, Transient System, Token, RSA Algorithm

I. INTRODUCTION

Cloud solutions are scalable and ubiquitous, and follow a pay per use approach at all levels. One of the main barriers to the adoption of Cloud Computing is security. User data are stored on provider servers and there is no guarantee that thiecures information will not be accessible to a third party. This can contravene legal requirements when the stored data are sensitive, as occurs in health care or banking environments. There is no guarantee that the data will be safe and secure at the server side. Here we presented a solution for s storing of data in the cloud environment through the use transient authentication.

II. METHODS AND MATERIAL

A. Literature Survey

Computing authentication requires that a user supply some proof of identity–via password, smartcard, or biometric–to advice. Unfortunately, it is infeasible to ask users to provide authentication for each request made of a device. Imagine asystem that requires the user to manually compute a message authentication code for each command. The authenticity ofeach request can be checked, but the system becomesunusable.Instead, users authenticate infrequently to devices.User authentication is assumed to hold until it is explicitlyrevoked, though some systems further limit its duration tohours or days. Regardless, in this model authentication is persistent andcreates tension between security and usability. To maximizesecurity, a device must constantly re-authenticate its user. Tobe usable, authentication must be long-lived. Unfortunately,authentication between people and their computer devices isboth infrequent and persistent. Should a device fall into thewrong hands, the imposter has the full rights of thelegitimate user. Researchers at the University of Michiganhave developed a new model, called “transientauthentication,” in which a user wears a small token,equipped with a short-range wireless link and modestcomputational

IJSRSET1623137 | Received : 28 May 2016 | Accepted : 02 June 2016 | May-June 2016 [(2)3: 458-464]

resources. This token is able to authenticate constantly on the user’s behalf. It also acts as a proximity cueto applications and services; if the token does not respond toan authentication request, the device can take steps to secure itself. This technology provides an improved method and system to maintain application data security on machines that are running or have been suspended. Applications are protected transparently by encrypting in-memory state when the user departs and decrypting this state when the user returns. This technique is effective, requiring just second’s toprotect and restore an entire machine. In the second embodiment, applications utilize an API for transient authentication, protecting only sensitive state. Ports of three applications, PGP, SSH, and Mozilla are described with respect to this API.

B. Problem Definition

Develop college based transient authentication system cloud security during the data transfer in the cluod by providing token to any device which consist of uniquecode such mobile or bluethooth device this system is also at basic college level testing in this system we are using RSA alogorithm for encryption and decryption

C. Work Done

Information i.e. username and password once and it may not be modified ever. How the system will come to know whether the person typing the user name and password is authenticated or not. One solution is to ask user to enter his username and password frequently but this Transient authentication is temporary authentication. User provides his authentication would be a tedious job. And user will remove his authentication system. Transient authentication resolves this problem. A hardware token on behalf of the actual user will communicate the authentication information frequently to the system. When the hardware token (a mobile) will be in Bluetooth range of the desktop the token i.e. the authentication information will be passed. And the moment when the device is out of the Bluetooth range the data is encrypted again and user cannot access the data.This transient authentication technique can be extended to the cloud computing for securing the data stored on to the cloud.the proposed solution on the security issue related to data stored on the cloud is “Transient Authentication”. Transient Authentication is temporary authentication. Cloud users can store their data and also can deploy their applications on to the cloud. Cloud users are dependent on the cloud provider for the security of data kept onto the cloud. But due to the multi-tenancy feature of cloud there is a great security issue related to cloud data. Unauthorized user may access the private data of cloud user. Also the issue of trust evolves. Cloud users cannot trust cloud providers. Hence to access data cloud must check whether the user accessing the data is authorized user of not.Providing username and password for the cloud user is one of the solutions. But how will the cloud confirm that the user typing the password is authorized user or not who has provided the password days back. So to overcome this, cloud user has to provide his identity after particular time duration. Also this process is tedious to the user. Thus transient authentication technique can be used in such situations. Here a mobile device, which is in the Bluetooth range of the host machine through which user will be accessing cloud, will provide the users identity to cloud on behalf of the user. The mobile device will exchange its information to the host machine and host machine in turn will pass this information to the cloud. This will prove the users identity periodically.When the mobile device is not within the Bluetooth range of the host machine there will be no exchange of data within the three entities i.e. the mobile device, the host and the cloud. Hence the user will not be able to access the data from cloud

Module 1: Desktop Application

Desktop application will be serving as the front end to the user. It will be any system for eg. a student information system which is connected to the backend through a central controller. User interacts with the system and also the transient device. It will pass the unique identification code with the data to the central controller.

Module 2: Central Controller

The central controller will consist of workflows. It will be working as the central controller or coordinator between the desktop application and different services. Workflows are to be designed for different services a user would like to access. Depending upon the users request i.e either to save the data or to retrieve the data, it will choose the proper workflow and the control is given to the respective service.

If user wants to save the data it will choose the workflow that will pass the data which also consist of the unique code to the transient device service where it will check whether the user is authenticated user or not, if yes then it will encrypt the data through encryption & decryption service and finally store into database.

If user wants to retrieve the data he will send the request with the unique code and the transient device service will check it. If user is authenticated then encrypted data is retrieved from the database and is decrypted by encryption & decryption service and the data is retrieved by the user.

Module 3: Services Designing

The three services basically to be designed are

- Encryption & decryption service
- Database service
- Transient device service
- Configuration service

a. Encryption & Decryption Service :

This service deals with the encryption and decryption of data while storing and retrieving the data to and from the database. All the requests to this service must come from transient device service and database service.

b. Database Service :

This service deals with the storing the encrypted data to the database. It will store the data only if the request is approved by the transient device service.

c. Transient Device Service :

First the user has to register his transient device and the unique code is stored in the database. Each time the user has to store or retrieve the data it will send the unique with the request. Now the task of transient device service is to check the authentication of the user. If the user is authenticated then the further transactions are allowed to be performed else the request is rejected.

d. Configuration Service :

This service deals with the configuration of the transient device and its compatibility with the services.

Module 1: Desktop Application

Desktop application will be serving as the front end to the user. It will be any system for eg. a student information system which is connected to the backend through a central controller. User interacts with the system and also the transient device. It will pass the unique identification code with the data to the central controller.

Module 2: Central Controller

The central controller will consist of workflows. It will be working as the central controller or coordinator between the desktop application and different services. Workflows are to be designed for different services a user would like to access. Depending upon the users request i.e either to save the data or to retrieve the data, it will choose the proper workflow and the control is given to the respective service.

If user wants to save the data it will choose the workflow that will pass the data which also consist of the unique code to the transient device service where it will check whether the user is authenticated user or not, if yes then it will encrypt the data through encryption & decryption service and finally store into database.

If user wants to retrieve the data he will send the request with the unique code and the transient device service will check it. If user is authenticated then encrypted data is retrieved from the database and is decrypted by encryption & decryption service and the data is retrieved by the user.

Module 3: Services Designing

The three services basically to be designed are

- Encryption & decryption service
- Database service
- Transient device service
- Configuration service

e. Encryption & decryption service :

This service deals with the encryption and decryption of data while storing and retrieving the data to and from the database. All the requests to this service must come from transient device service and database service.

f. Database service :

This service deals with the storing the encrypted data to the database. It will store the data only if the request is approved by the transient device service.

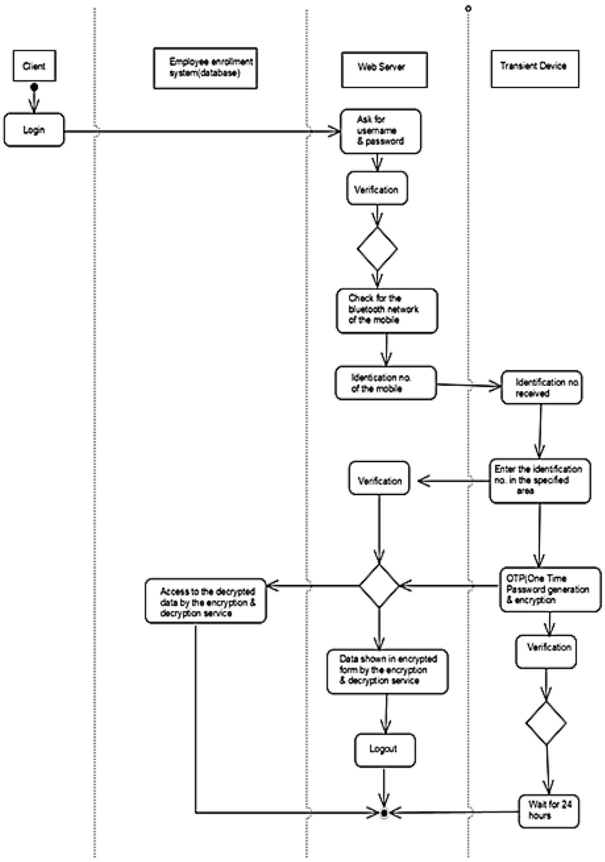
g. Transient device service :

First the user has to register his transient device and the unique code is stored in the database. Each time the user has to store or retrieve the data it will send the unique code along with the request. Now the task of transient device service is to check the authentication of the user. If the user is authenticated then the further transactions are allowed to be performed else the request is rejected.

h. Configuration service :

This service deals with the configuration of the transient device and its compatibility with the services.

D. Activity Diagram


Client
Employee enrollment system(database)
Web Server
Transient Device
Login
Ask for username & password
Verification
Check for the bluetooth network of the mobile
Identication no. of the mobile
Identification no. received
Enter the identification no. in the specified area
Verification
OTP(One Time Password generation & encryption
Access to the decrypted data by the encryption & decryption service
Verification
Data shown in encrypted form by the encryption & decryption service
Logout
Wait for 24 hours

III. RESULTS AND DISCUSSION

Types of Tests

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.
TEST Cases:

Test case ID	1
Test case name	Display home page screen
Test case description	After clicking on application the system should open home page .
Test steps	1. Click on MyBEProject icon
Expected result	Application should provide the home page and displays following option to the user. 1. VNC Server 2. Data Server
Actual result	
Status	

Test case ID	2
Test case name	Run data server.
Test case description	Run the Data server service on mobile server
Test steps	Click on Data server
Expected result	System should display following link on the screen Http://192.168.1.1:8080
Actual result	
Status	

Test case ID	3
Test case name	Access the data service from URL
Test case description	Accessing the Data service with the help of browser.
Test steps	1. Perform test case 2 2. Open chrome browser 3. Enter link :Http://192.168.1.1:8080 and hit enter key. 4. Enter user name and Password as admin/paw
Expected result	User will successfully accessed the URL and user can see Home page.
Actual result	
Status	

Test case ID	4
Test case name	Verify options present on home page
Test case description	Verify the different operation present on the screen
Test steps	1. Perform test case 3
Expected result	System should open paw web server application home page and shows the following options on right side 1.your device 2.android device 3.update 4.Internal storage 5.SD card System should show the following options on left side 1.Phone 2.Media 3.system 4.session
Actual result	
Status	

Test case ID	5
Test case name	Perform detect mobile operation
Test case description	Perform detect mobile operation
Test steps	1.Go to the home page of paw web server application and click on find mobile option. 2. Click on start button
Expected result	2. System should play the mobile ringtone. And user can see stop option on screen.
Actual result	
Status	

Test case ID	6
Test case name	Verify stop button on find mobile page
Test case description	Verify stop button on find mobile page
Test steps	1. Perform test case no 5 2.click on stop button and check the result.
Expected result	System should stop the ringtone of your mobile.
Actual result	
Status	

Test case ID	7
Test case name	Verify system details option of paw server web application.
Test case description	Verify details of system option of paw server web application.
Test steps	1.Go to the home page of paw web server application and click on system operation. 2..Click on APK backup and check the result.
Expected result	1. system should contains the following option

	- Process list - Installed apps - APK backup - Volumes 2. system should open APK backup details it contains following information - APP name - APP version - Date - Size of app
Actual result	
Status	

Test case ID	8
Test case name	Verify VNC server URL
Test case description	Verify VNC server Functionality
Test steps	1.Start VNC server application 2. Enter URL : http://192.168.2.2.9090
Expected result	System should Display mobile screen in browser
Actual result	
Status	

Test case ID	9
Test case name	Selecting Any application icon on Mobile through Browser.
Test case description	functionality of select operation
I. REFERENCES Test steps	1.Start application 2.system should contains mobile screen displayed in browser. 3.move the cursor on any icon and click it
Expected result	The icon will select in browser as well as mobile.
Actual result	
Status	

Test case ID	10
Test case name	Verify calling from browser .
Test case description	To Verify Mobile Call can be initiated from Browser.
Test steps	1.Execute Test case no – 8 2. Click on contacts 3. Select any contact and click on dial icon
Expected result	System should connect a call to dialed contact person.
Actual result	
Status	

IV. ACKNOWLEDEMENT

I would like to take this opportunity to thank my internal guide U. C. Patkar, Head of Computer Engineering Department, BVCOEL for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

In the end our special thanks to Co-guide for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our work

V. REFERENCES

[1] M. Baker, R. Buyya, and D. Laforenza, "Grids and gridtechnologies for wide-area distributed computing,"International Journal of Software: Practice andExperience, vol.32, pp. 1437-1466, 2002.

[2] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya,"Autonomic metered pricing for a utility computing service", Future Generation Computer Systems, vol. 26,issue 8, pp. 1368-1380, October 2010.

[3] R. Sterritt, "Autonomic computing," Innovations in Systems and Software Engineering, vol. 1, no. 1,Springer, pp. 79-88. 2005.

[4] William Stallings, lawrieBrown,"Computer Security", Pearson.