SESSION MANAGEMENT IN MAINFRAME

[1]ROBIN TOMMY, [2]ASWATHY S KRISHNA, [3]ULLAS RAVI

[1,2,3]Tata Consultancy Services, Trivandrum

Abstract— This paper proposes the usage of Session Management in Mainframes. In mainframes, unattended interactive sessions can lead to resource locking and data compromises (or infiltrations). The session timeout features prevents unauthorized access to such unattended sessions. Coupled with data restoration innovation implemented in mainframes, it makes the application more user friendly, innovative, secure and saves time for the user. DB2 plays an important role in managing the sessions in mainframe. This paper also addresses how Session Management will improve the user experience and prevent unauthorized access to unattended sessions in Mainframes. Session Management is implemented using COBOL, DB2, CICS in mainframe.

Keywords— Session Management, Mainframes, DB2, Security, Session recovery, Data restoration, Terminal id.

I. INTRODUCTION

According to a recent report from the Ponemon Institute, employee negligence was identified as a top threat to information security [2]. A statistics on survey of IT professionals and assessment of cybersecurity posture shows that at least 50 percent of breaches and leaks are directly due to user error or failure to practice proper cyber hygiene [3]. Each lost or stolen record containing sensitive and confidential information costs a consolidated average of $145.10 [4]. Users cannot be relied upon to do right things to create a truly secure system. People will not be investing on most secure solution, if it is not user-friendly [3]. Session management in mainframes is an effort to streamline the user experience on a mainframes environment while contributing to the security and usability of the application. The entire process is based on the concept that by keeping track of the session's time and data that is being entered on the screen, a system will be able to lock users out depending on the time the user has remained idle, as well as restore a valid session after due authentication by redirecting users to the screen in which they were working on and populating data already entered by the user. When the timeout duration expires, a log-in page is displayed to the user. The log-in page is displayed even when the same user with a different terminal ID tries to access the application. The user is only deemed eligible for session and/or data restoration if both the user credentials as well as terminal id match the data already stored in the database.

II. IMPLEMENTATION

In mainframes, we can implement session management and data restoration by using EIBTRMID and EIBTIME variables. These variables enable us to process the system terminal ID and time. Two different databases, used for restoring session and associated data are being populated by the users of the system. The data pertaining to session start time, time last active, screen number of application in which user is working on, etc is stored in the first database to empower session restoration at a later time, after due authentication process. The second database stores data pertaining to data restoration part of the application. Some of these are the number of fields in the screen, screen number and the data that has been already entered by the user.

The algorithm, explained in due course enables the system to distinguish between users who are logging in for the first time, users who are eligible for data restoration as well as users who are not eligible for a session and or data restore. The data storage and retrieval is performed by use of a database system.

Some of the important algorithms that are implemented are as follows:

2.1 Session Timeout: It is the technique by which user is logged out of the application after a designated period of time when the session is idle.

- 2.1.1 Take a record of the time when the user logs in to the application (T1).
- 2.1.2 Maintain session time of 1 minute for the user.
- 2.1.3 When the user aid key after 1 minute, take a record of the current time (T2).
- 2.1.4 Take the difference of the two times, D(mins) = T2(min) – T1(min)
- 2.1.5 if D > 1, the user needs to login to the application.
- 2.1.6 if D < 1, the user doesn't need to login again.

2.2 Data Restoration**:** It is the technique of preserving the user's data that was keyed in before the session timeout.

- 2.2.1 Take a record of the time when the user logs in to the application (T3).
- 2.2.2 Each data keyed by the user will be stored in database as a single record with user name and terminal id as the primary key
- 2.2.3 Maintain a session time of 30 seconds (configurable)

2.2.4 After time-out, if the same user logs in again within 30 seconds (configurable), the data will be restored.

2.2.5 Data is restored by fetching the data from database based on the user name ,terminal id and map.

III. RESULTS

The session management concept in mainframes will help to prevent unauthorized access to unattended critical applications and thereby helps in preventing security breach and data breach. We have implemented the concept as explained below:

Authorized user if leaves the system unattended for some time in the middle of a work (Figure 1), can lead to unauthorized access under normal situation. However since we have implemented session management in mainframe, if the session remains unattended for some time (configurable), the unauthorized user will be able to be see a time-out page only (Figure 2). After the time-out page, the user will be re-directed to log-in page (Figure 3).

Once an authorized user logs in to the system within 30 seconds (configurable) after time-out, the log in credentials and terminal id are validated against the previous stored session's details and the data is restored in the screen.

With this innovation, it is now possible to implement data restoration with session timeout in mainframes. Therefore, implementation of data restoration helps to implement an advanced session management as proposed in [1].

CONCLUSION

It is a well known fact that there is no other computer system which is more secure and safe as mainframes. Apparently, drawback of mainframe is that it is not updated with current trends of technology. Utilization of such innovations adds another layer of protection for application and the users. The use of session management prevents unauthorized access. So we need to ensure that timeouts happen appropriately in an application. Data restoration on the other hand, promotes user friendliness by preserving the data that the user keyed in before session expires. But to access this data, the user needs to log in again to the application, thereby preventing unauthorized access to the resources.

FIGURES

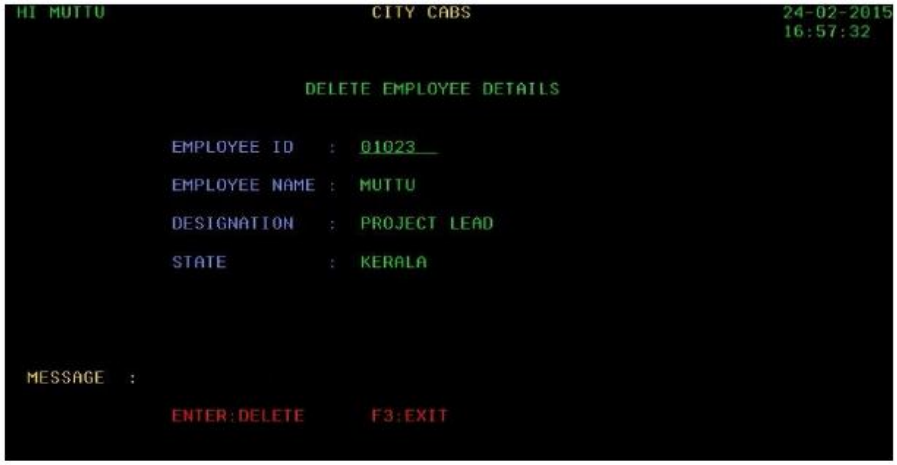

Figure 1: Application in use before time-out

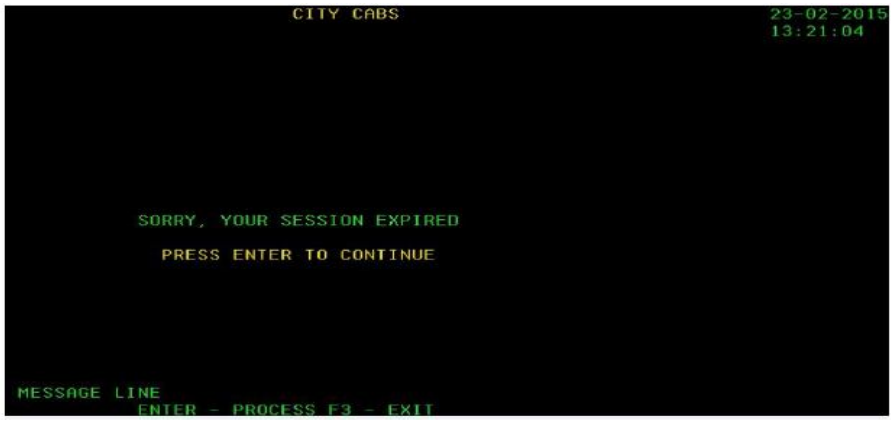

Figure 2: Alert page for time-out

Figure 3: Login Screen after session time-out

REFERENCES

[1] **S**ingle sign on – innovative mainframe CICS application *Vol.1: Issue.2, October-November 2012 pp- 7-10 ; www,ijrde.com*

[2] *http://complianceandethics.org/tag/data-breach/*

[3] *http://www.federaltimes.com/story/government/cybersecurity/2015/04/14/the-user-knows-nothing/25776507/*

[4] *http://www.insurancejournal.com/news/national/2014/05/07/328512.htm*

[5] Live tiles in mainframe, AIRCC, *vol 5, Number 9, April 2015; http://airccse.org/V5N40.html*

[6] Internet of Things (IOT) Expanding the Horizons of Mainframes, *IEEE, IT Convergence and Security (ICITCS), 2015 5th International Conference 24-27 Aug. 2015*

[7] IBM Mainframe Handbook by Alexis Leon.

[8] Roul Mendes, Doug Lowe. Murach's CICS for COBOL Programmers

[9] CICS transaction server for z/OS, Version 4.2 – IBM.

[10] Taruna Yadav, Soumya Nayak, Sneha Mahajan, Anuja Munjewar, Kunal Jain, *User friendly Help and Error map in CICS Mainframes*

[11] Mike Ebbers , John Kettner , Wayne O'Brien , Bill Ogden, *Introduction to new mainframes : z/OS.*

[12] Roul Mendes, Doug Lowe. *Murach's CICS for COBOL Programmers.*

[13] Introduction to the new mainframe: Z/OS Basics – An IBM redbooks publication.

[14] esigning and programming CICS application.

★★★