 ISBN: 978-81-927230-0-9

Security in web based e-Governance Application – A Case study

Sanjay G Kulkarni

Principal Systems Analyst ,
National Informatics Centre ,
Ganeshkhind Road ,Pune 411007,
Maharashtra, India
kulkarni.sg@nic.in

Abstract— ***This paper is about the security of web based*** e Governance ***application Tribe Certificate Verification Information System(TCVIS)*** . The software ***is implemented in the 8 Tribe Certificate Scrutiny Committees in Maharashtra.*** The validity certificate is issued to the applicant belonging to schedule Tribe, issued by District Magistrate or Sub divisional officer after undergoing workflow of issuing validity certificate.

Keywords— e-Governance, Validity Certificate, security, web, TCVIS

I. INTRODUCTION

It is increasingly recognised that the web applications are open windows into the IT infrastructure of the organisations. The web application allows the user to access the important resources web server, data base server etc .Every day 5000 attempts are made to hack Indian government IT infrastructure[1]. As per Gartner group study 75% of the attacks are targeted to the application level and not the infrastructure level. It is important to develop and deploy web application which is secured . This paper is about the security aspects undertaken for development and deployment of an e-Governance application called TCVIS.

II. PROJECT BACKGROUND

An applicant belonging to Scheduled Tribe wants to take benefits of the reservation of the Caste/Tribe applies for the Schedule Tribe Certificate to the Competent Authority(which could be Tehsildar, District Magistrate, Sub Divisional Officer etc). The benefits of reservation in respect of employment, education and election are admissible to backward category persons in Maharashtra only if the respective Caste / Tribe certificate is validated by Scrutiny Committee(s) set up for the purpose. There are 8 Scheduled Tribe Certificate Scrutiny Committees in Maharashtra located in 1)Pune, 2) Nashik, 3)Thane, 4)Aurangabad, 5)Nagpur, 6)Amaravati, 7)Nandurbar and 8) Gadchiroli covering all 35 Districts of Maharashtra. The procedure for issuing and Validating the Schedule Caste/Tribe is given in the CV Act 2000 of Maharashtra[4]

The Act : "An Act to provide for the regulation of the Issuance and Verification of the Caste/Tribe Certificates to the persons belonging to the Scheduled Castes, Scheduled Tribes, De-notified Tribes (Vimukta Jaties), Nomadic Tribes, Other Backward Classes and Special Backward Category and matters connected therewith or incidental thereto."

III. PROJECT OBJECTIVES

The following were objectives for developing the web based workflow system [5]

- To monitoring pendency of the cases of the Committees located in the different places.
- To know the status of the case at any point of time to the Committee Staff as well as the Applicant/Sponsoring Agency.
- To generate the Standard MIS Reports timely and query based reports as and when required.
- To automate the workflow of Tribe/Caste Verification Process to achieve the above objectives.

IV. TECHNOLOGY USED

The open source platform Linux, PHP, Apache, PostgreSQL was used in order to save the cost to the Department and to avoid vendor locking. The application software was deployed on State Data Centre Mumbai .

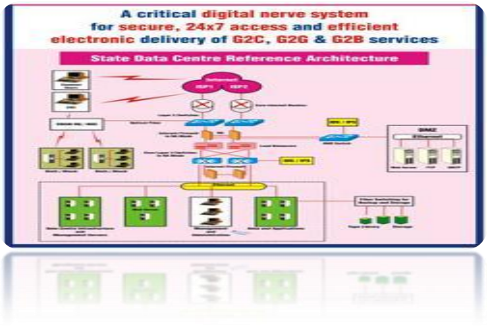

Fig 1. Architecture of SDC

The Fig. 1 shows the architecture of State Data Centre .

V. SECURITY

The security of the web based application is important for continuation of the application and Government business. The Open Web Application Security Project (OWASP) project is non profitable organisation dedicated to provide unbiased , practical information about web security. The application security affects all organisations OWASP top 10 are prevalent in most of the industries which are listed in table 1

TABLE I

OWSAP TOP 10 APPLICATION SECURITY RISKS

No.	Security Risk
A1	Injection
A2	Cross Site Scripting (XSS)
A3	Broken authentication and session management
A4	In secure direct object references
A5	Cross site request Foregery
A6	Security Misconfiguration
A7	Insecure Cryptographic
A8	Failure to restrict URL Access
A9	Insufficient transport layer protection
A10	Unvalidated redirects and forwards

The majority of occurring software security holes in web applications may be sorted into just two categories: Failure to deal with metacharacters, and authorization [3].

1) *Injection :* Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query through an HTTP request. The most common Header Manipulation attacks is HTTP Response Splitting.

VI. MEASURES TAKEN

The following precautions were taken for the application security

1) *Physical Safety :* The application is hosted on State Data Centre which is guarded by security guard 24X7 .The physical security involves protection of computer system , storage media, netwrok equipments, theft , fire and flood. The State data centre access mechanism is in place as well as DR in place for data loss .
2) *Role Based Access*: Role based access was given in the application.
3) *Input validation* : The input was validated on client side and server side with positive or whitelist validation.
4) *Log :* Access log was maintained to know who has access the application.
5) *Salted md5 :* The salted md5[6] technique was used for loging in the system.
6) *Hardening the system :* All latest patches of OS were downloaded and implemented . The Directory with the application was given readonly rights . All unwanted services were stopped also unnecessary files were removed .
7) *Audit trials :* The log of the audit trials was used for the changes given in the previledges deletion of user, additiion of the user, addditition of the previledges , delition of the previledges. The trail was also kept in the change of the data.
8) *Security Audit :* The security audit was carried out and the vulnerability in the code was removed. Code reviews were also made by the team to find out the vulnerability.
9) *Checking IP :* Checking IP for different browser on same session ID
10) *Server Signature:* The signature was made off.
11) *Error messages:* Error messages were supressed from users of the application.

There are many more security aspects used in the application at the code level and exhaustive list cannot be given as it depends on application to application and also on the scripting language used. Some importanat aspects used are mentioned above. when designing the secured system basic security thinking needs to be Trust Nobody and nothing ,Assume a worst case scenario,Apply-Defense in Depth,Keep it Simple Stupid(KISS),principle of least previledge, Attackers can smell obscurity, Read the manual , If it wasn't tested, it doesn't work, It's always your fault! [7]

VII. CONCLUSIONS

It is increasingly recognised that the web applications are open windows into the IT infrastructure of the organisations. It is difficult to guarntee any application of being secured. Procurement of hardware , manpower etc may add to the cost interms of maintenance .This paper gives information of the security precautions observed in an e-Govenance application called TCVIS. Security is important for business countinuity and hence needs to be taken care of while developing an application software.

REFERENCES

1. Government Technologies handbook 2010 Article on Essentials of Penetration Tests and why it is important
2. Sverre H. Huseby, " Common Security Problems in the Code of Dynamic Web Applications" , Web application security consortium, 01-06-2005, http://www.webappsec.org/projects/articles/062105.shtml
3. OWASP top 10 https://www.owasp.org/index.php/Main_Page
4. Kulkarni Sanjay G. "ICT Enabled Tribe Certificate Verification in Maharashtra Pg 9-11 Informatics Vol. 22 No. 4 April 2014. http://informatics.nic.in
5. Kulkarni Sanjay G.,Praveen rao, Ms. Kamat P.V.Use of ICT in Caste / Tribe Certificate Verification Process in Maharashtra (G2G and G2C) Stoch held in September 2011.
6. Salted Password Hashing - Doing it Right https://crackstation.net/hashing-security.htm#salt
7. Servive the deep end PHP Security http://phpsecurity.readthedocs.io/en/latest/Introduction.htm