Implementation of XSS attack Prevention using Ramp Secret Sharing

Mrs.Tejaswini H. Aware[1], Prof. D. S. Kulkarni[2]
Department of Computer Engineering
[1, 2]D. Y. Patil College of Engineering Ambi Pune.

Abstract*-The web applications are developed using various technologies like HTML, JavaScript, AJAX, XML etc. But the vulnerabilities at the design level in these technologies lead to security breach, resulting in theft of the users credentials. Thus, the security of these applications is becoming an important concern to ensure the users authentication and privacy. Cross site scripting attack (XSS) is also an exploitation of these vulnerabilities existing in the web applications. XSS still remains a big problem for web applications, despite the bulk of solutions provided so far. Content Security Policy (CSP) is also an approach to prevent this code injection. This paper studies the browser compatibility issues in deploying CSP to mitigate XSS vulnerabilities and also discusses how to resolve this incompatibility. A content security policy (CSP) can help Web application developers and server administrator's better control website content and avoid vulnerabilities to cross site scripting (XSS). In experiments with a prototype website, the authors CSP implementation successfully mitigated all XSS attack types in four popular browsers. Among the many attacks on Web applications, cross site scripting (XSS) is one of the most common. An XSS attack involves injecting malicious script into a trusted website that executes on a visitors browser without the visitors knowledge and thereby enables the attacker to access sensitive user data, such as session tokens and cookies stored on the browser. With this data, attackers can execute several malicious acts, including identity theft, key logging, phishing, user impersonation, and webcam activation. Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.*

Keywords-Cross-Site Scripting, Content Security Policy, Vulnerability, SQL Injection Attack, JavaScript, Scripting Language, Web Application Security

I. INTRODUCTION

In the recent years, web-based attacks have caused harm to the users of web applications. Most of these attacks occur through the exploitation of security vulnerabilities in the web-based programs. So, the mitigation of these attacks is very crucial to reduce its harmful consequence. The main issue is that if malicious content can be introduced into a dynamic web page, neither the web site nor the client is capable of recognizing that anything like this happened and prevent it.

A primary goal of CSP is to mitigate and report XSS attacks. XSS attacks exploit the browser's trust of the content received from the server. Malicious scripts are executed by the victim's browser because the browser trusts the source of the content, even when it's not coming from where it seems to be coming from CSP makes it possible for server administrators to reduce or eliminate the vectors by which XSS can occur by specifying the domains that the browser should consider to be valid sources of executable scripts. A CSP compatible browser will then only execute scripts loaded in source files received from those white listed domains, ignoring all other script (including inline scripts and event-handling HTML attributes).

As an ultimate form of protection, sites that want to never allow scripts to be executed can opt to globally disallow script execution. Even major application services such as facebook, Google, PayPal, and Twitter suffer from XSS attacks, which have grown alarmingly since they were first reported in a 2003 Computer Emergency Response Team advisory. The Open Web Application Security Project ranked XSS third on its 2013 list of top 10 Web vulnerabilities (the latest list as of February 2016), calling it the "most prevalent Web application security flaw. Underscoring the widespread risk of XSS intrusions, Whitehat Security's May 2013 Web Security Statistics Report noted that 43 percent of Web applications were vulnerable to this kind of attack Researchers have proposed a range of mechanisms to prevent XSS attacks, with content sanitizers dominating those approaches. Although sanitizing eliminates potentially harmful content from entrusted input, each Web application must manually implement it. A process prone to error. To avoid this problem, we use a different technique. Instead of sanitizing harmful scripts before they are injected into a website, we block them from loading and executing with a variation of the content security policy (CSP), which provides server administrators with a white list of accepted and approved resources. The Web application or website will block any input not on that list and thus there is no need for sanitizing. The white list also guards against data infiltration and extrusion—the unauthorized downloading of data from a website visitor's computer.

Validate user inputs and encode user outputs.

Cross site scripting has been a major threat for web applications and its users from past few years. Lot of work has been done to handle XSS attacks which include:

- Client side approaches
- Server side approaches
- Testing based approaches
- Static and dynamic analysis based approaches

Each kind of solution has been discussed in this paper. Different approaches have their own advantages and disadvantages. Major problems faced are:

- Requirement of complex frameworks
- Additional runtime overhead
- Intensive labour requirements
- Not being able to cover all types of XSS attacks
- Prone to human error
- Requires client action
- Not able to detect web content manipulation
- False positives and false negatives
- Effectives depend on completeness of specification

Based on our requirements we can choose among the possible solutions. However, there is no ideal solution for the detection and prevention of XSS attacks.

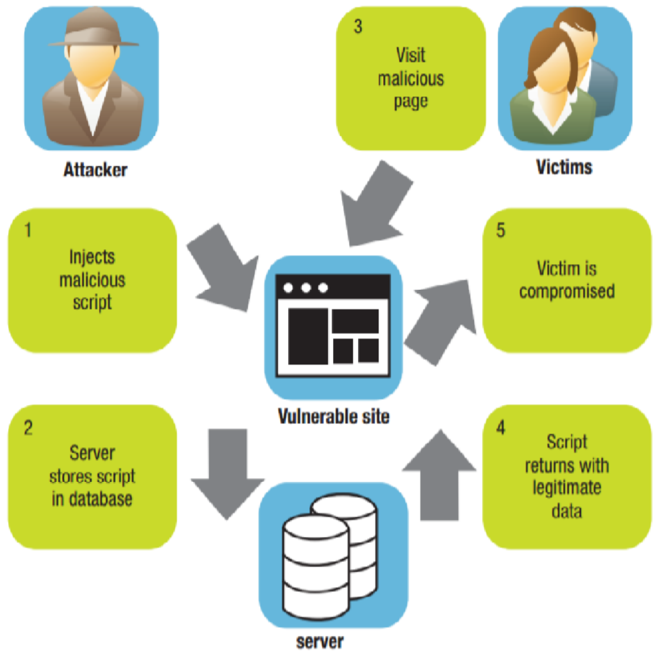


Fig.1: Block diagram of cross site scripting attack

A content security policy (CSP) can help Web application developers and server administrator's better control website content and avoid vulnerabilities to cross site scripting (XSS). In experiments with a prototype website, the authors' CSP implementation successfully mitigated all XSS attack types in four popular browsers. Among the many attacks on Web applications, cross site scripting (XSS) is one of the most common. An XSS attack involves injecting malicious script into a trusted website that executes on a visitor's browser without the visitor's knowledge and thereby enables the attacker to access sensitive user data, such as session tokens and cookies stored on the browser. With this data, attackers can execute several malicious acts, including identity theft, key logging, phishing, user impersonation, and webcam activation. Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP is designed to be fully backward compatible; browsers that don't support it still work with servers that implement it, and vice-versa. Although our CSP has many benefits, it is not intended as a primary defence mechanism against XSS attacks. Rather, it would best serve as a defence in depth mitigation mechanism. A primary defence involves tailored security schemes that

A) SYSTEM METHODOLOGY

Completeness in the sense if x and y belongs to R the honest prover who knows witness y for x succeeds in convincing the honest verifier of his knowledge. Soundness is if x and y is not belonging to R no cheating prover can convince the honest verifier that x and y belongs to R, except with some small probability. It can be captured by the existence of a knowledge extractor E to extract the witness y: given oracle access to a cheating prover P, the probability that E outputs y must be at least as high as the success probability of P in convincing the verifier. For a zero-knowledge proof of knowledge, it has the extra property of Zero-knowledge: no cheating verifier learns anything other than x and y belongs to R. It is formalized by showing that every cheating verifier has some simulator that can produce a transcript that is indistinguishable with an interaction between the honest prover and the cheating (or honest) verifier.

- ➢ The authentication server picks at random a challenge R belongs to Zp and sends R to the user
- ➢ The user computes C = e^(g,h01/(y+R) and submits (C,y,R) to his /her security device
- ➢ The security device validates C(y+R) = TG and TGy = TY.

- Upon successful validation, the security device picks a random r belongs to Zp, computes CR = H(TGr ||R||C) and ZR= r-CRtsk . It returns (CR,ZR) to the user.
- The user converts claim predicate value to its corresponding monotone span program M = (Mi,j) belongs to Zpl*m , with row labeling .
- For i = 1 to l , the user randomly picks ai; ti belongs to Zp and computes Ci = gvi hti
- Then engage in zero-knowledge protocol, all the parameters are given to the protocol for validation process.

The two type of design are as following

1. Input Design

This is a process of converting user inputs into computer based formats. The data is fed into system using simple interactive forms. The forms have been supplied with messages so that user data is read without facing any difficulty. The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. It also includes determining the recording media methods of input, speed of capture and entry into the system.

2. Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In the output design, it is determined how the information is to be displayed for immediate need and also the hard copy output. The output design should be understandable to the user and it must offer great convenience. The output of the proposed software tool is designed as opening the text file containing the translated code. The main objectives that guide the output design is that the user must be informed with proper messages and access permissions in case of any unusual key input or in case of the right key validation.

II. EXISTING SYSTEM AND PROPOSED SYSTEM

A) EXISTING SYSTEM

An XSS attack involves injecting malicious script into a trusted website that executes on a visitors browser without the visitors knowledge and thereby enables the attacker to access sensitive user data, such as session tokens and cookies stored on the browser.1 With this data, attackers can execute several malicious acts, including identity theft, key logging, phishing, user impersonation, and webcam activation.

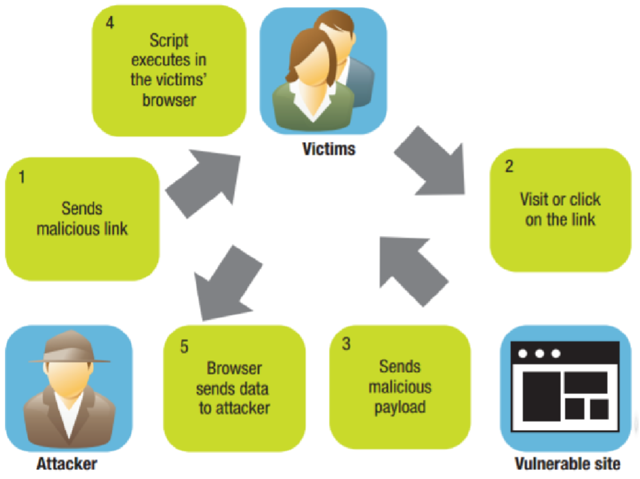

4
Script executes in the victims' browser
Victims
1
Sends malicious link
2
Visit or click on the link
5
Browser sends data to attacker
3
Sends malicious payload
Attacker
Vulnerable site

Fig.2: Existing System

Content Security Policy(CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP is designed to be fully backward compatible; browsers that don't support it still work with servers that implement it, and vice-versa. Browsers that don't support CSP simply ignore it, functioning as usual, defaultingto the standard same-origin policy for web content. If the site doesn't offer the CSP header, browsers likewise use the standard same-origin policy.

1. *DISADVANTAGES OF EXISTING SYSTEM*

- How to use the collected information in database is not addressed.
- How to make system deployed universally has also not been addressed.
- It requires modifications in the frameworks or installation of additional frameworks.
- Approved scripts have to be identified by the website.
- There is no single policy for all the documents.
- Creating policies manually is a very tough task.
- This approach incurs runtime overhead due to interception of HTTP traffic.
- It requires user-defined security policies which can be labour-intensive.

B) *PROPOSED SYSTEM*

A client-side tool that acts as a Web proxy, disallows requests that do not belong to the website and thus thwarts stored XSS attacks. Browser-enforced embedded policies

(BEEPs) let the Web application developer embed a policy in the website by specifying which scripts are allowed to run. With a BEEP, the developer can put genuine source scripts in a white list and disable source scripts in certain website regions. Document Structure Integrity (DSI) is a client-server architecture that restricts the interpretation of entrusted content. DSI uses parser-level isolation to isolate inline entrusted data and separates dynamic content from static content. However, this approach requires both servers and clients to cooperatively upgrade to enable protection.

1. ADVANTAGES OF PROPOSED SYSTEM

- Since the requests with waiting time D are all assigned to temporary servers, it is apparent that all service requests can guarantee their deadline and are charged based on the workload according to the SLA. Hence, the revenue of the service provider increases.
- Increase in the quality of service requests and maximize the profit of service providers.
- This scheme combines short-term renting with long-term renting, which can reduce the resource waste greatly and adapt to the dynamical demand of computing capacity

C) SECURITY MODULE DETAILS

The first part Setup is run by a trustee to generate public parameters. The second part Setup is run by the attribute issuing authority to generate its master secret key and public key. The process of authenticating the user and access control mechanism consist of mainly two steps.

1. User Key Generation Phase:-

The user key generation process consists of three parts.

- First, the user generates his secret and public key in USetup.
- Then the security device is initialized by the trustee in Device Initialization. All the public parameters generated are used during the authentication process.
- Finally the attribute issuing authority generates the user attribute secret key accord-ing to the users attribute in AttrGen. The secret generated by the attribute issuing authority determines which all data the user can access and this key is stored in the user computer. The user has to use his own computer and the USB key storage each time for accessing the cloud. Additional recovery options are also provided.

2. Access Authentication Phase

The access authentication process is an interactive protocol between the user and the cloud service provider. It requires the user to have his partial secret key, attribute secret key , Policy details and the security device each time the user is accessing the cloud.

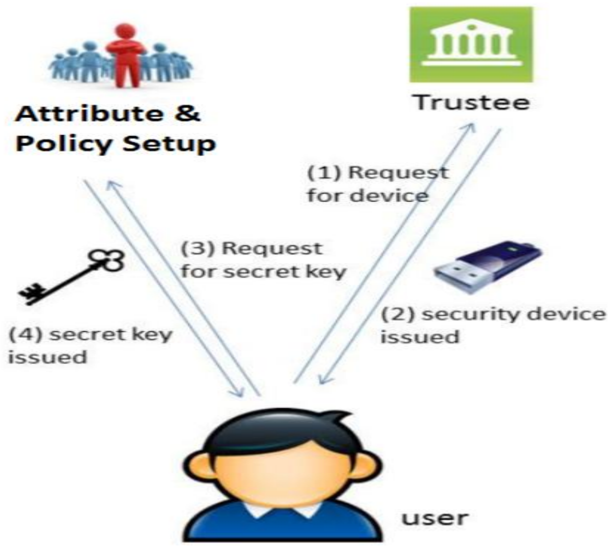
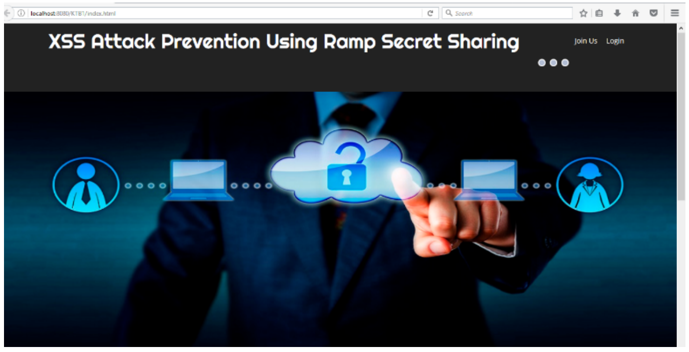


Fig.3: 3FA Working Details

III. PROJECT IMPLEMENTATION

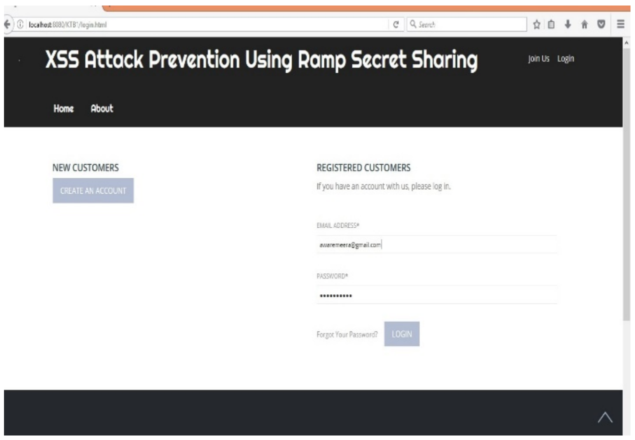
1. STEP-1

This is the project home page window which contain Join us and Login. When user click on the join us then new user registration form will display that contain the user information fields. When user fill these data then new user will register in the database.



2. STEP-2

When user click on the login then login form will display. Only already registered user can use this form to login there account which contain Email address and password.

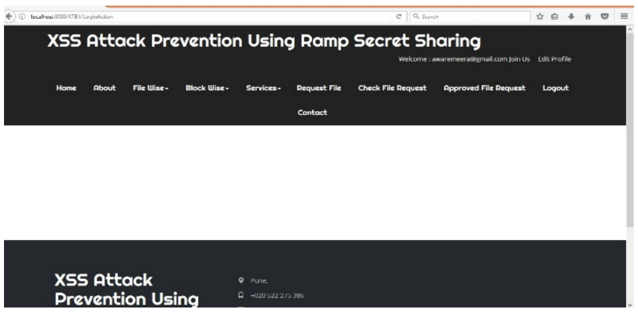

localhost:8080/KTB1/login.html
XSS Attack Prevention Using Ramp Secret Sharing
Join Us
Login
Home
About
NEW CUSTOMERS
CREATE AN ACCOUNT
REGISTERED CUSTOMERS
If you have an account with us, please log in.
Forgot Your Password?
LOGIN

EMAIL ADDRESS* awaremeera@gmail.com

PASSWORD* ••••••••••

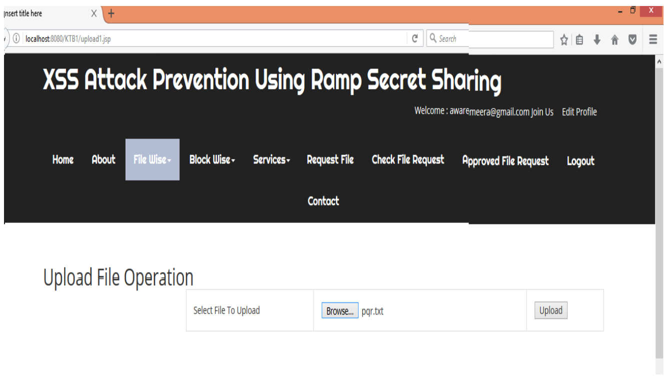
3. *STEP-3*

When login user by entering emailed and password then these window will display. Each login user has these field such as Home, About, File wise, Block wise, Services, Request File. Check file Request, Approved file Request and Logout. These all fields are work separately.



4. *STEP-4*

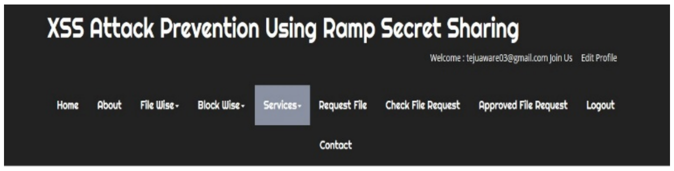
In this window, login user tries to upload new file. User click on the file wise and select the upload option then upload file operation will display. Then user browse the file which want to upload and click on the upload. After that “file upload successfully” message will display.

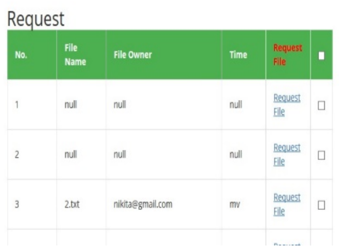


Select File To Upload pqr.txt

5. *STEP-5*

In this window, second user login there account and user will send request to download the file which uploaded in step 4. User will choose the file and click on the request file. In that user can send multiple requests to download multiple files from different users.



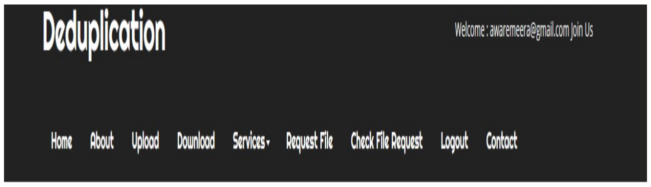


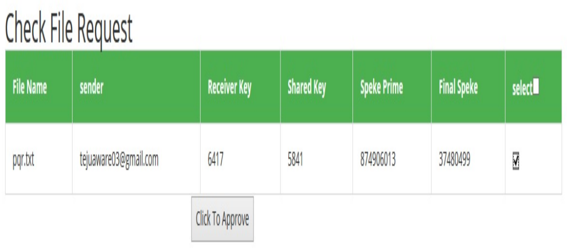
Request

No.	File Name	File Owner	Time	Request File	☐
1	null	null	null	Request File	☐
2	null	null	null	Request File	☐
3	2.txt	nikita@gmail.com	mv	Request File	☐

6. *STEP-6*

This window display the login user which got the request for file download from the step 5. This user will click on checkfile request and check the requested users. If user want to accept file download request then user click on the click to approve. In that user generate there receiver key, shared key, speke prime and final speke key.



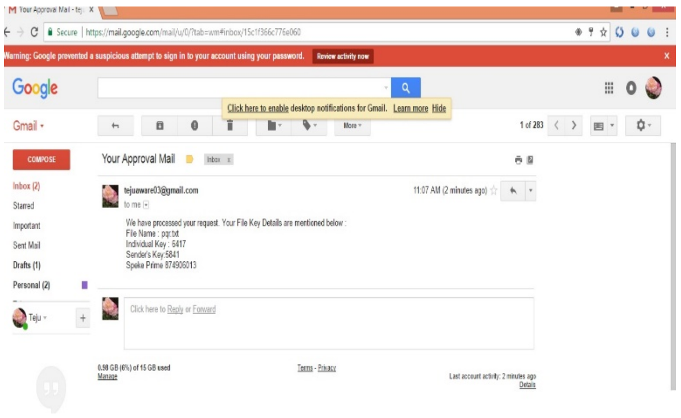


Check File Request

File Name	sender	Receiver Key	Shared Key	Speke Prime	Final Speke	select ☐
pqr.txt	tejuaware03@gmail.com	6417	5841	874906013	37480499	☑

7. *STEP-7*

When user click on the click to approve then user will send the keys on requested user’s email. These window shows the mail inbox of requested user which got the keys from approved users to download the file.


Your Approval Mail

tejuaware03@gmail.com

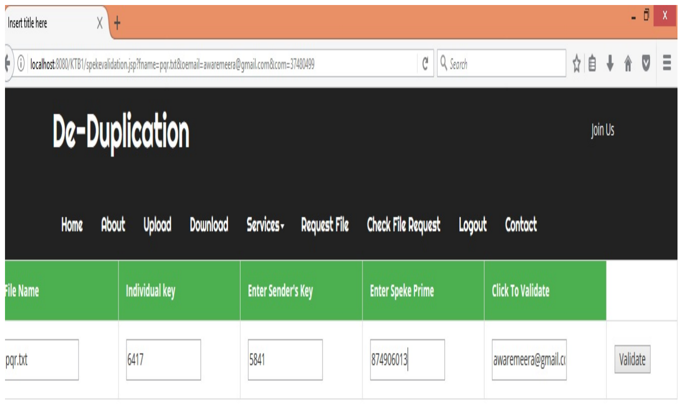
to me

We have processed your request. Your File Key Details are mentioned below :
File Name : pqr.txt
Individual Key : 6417
Sender's Key:5841
Speke Prime 874906013

https://mail.google.com/mail/u/0/?tab=wm#inbox/15c1f366c776e060
Warning: Google prevented a suspicious attempt to sign in to your account using your password.
Review activity now
Click here to enable desktop notifications for Gmail. Learn more Hide
COMPOSE
Inbox (2)
Starred
Important
Sent Mail
Drafts (1)
Personal (2)
11:07 AM (2 minutes ago)
Click here to Reply or Forward
0.98 GB (6%) of 15 GB used
Terms - Privacy
Last account activity: 2 minutes ago

8. STEP-8

When requested user got the keys from the email then user will copy these keys and enter in each keys separately in there place shown in window. After that user will click on the validate then "file download successfully" message will print on the window.


localhost:8080/KTB1/spekevalidation.jsp?fname=pqr.txt&oemail=awaremeera@gmail.com&com=37480499

File Name	Individual key	Enter Sender's Key	Enter Speke Prime	Click To Validate	
pqr.txt	6417	5841	874906013	awaremeera@gmail.c	Validate

IV. APPLICATION

1. Manufacturing and Engineering

The Manufacturing and Engineering sectors include a wide range of market segments, from aerospace to automotive. Manufacturing organizations face a number of computing challenges as they seek to optimize their IT environments, including high infrastructure costs and complexity to poor visibility into capacity and utilization. Today's design engineers need access to unrestrained, flexible computing capacity on demand, so that design cycles can be as fast, cheap, and productive.

2. Geospatial Sciences and Technologies

Due to the continuous growth of GIS sciences and technologies, there have been even more geospatial and non- spatial data involved due to increase in number of data sources and advancement of data collection methodologies. Spatial analysis and Geo-computation are getting intricate and computationally demanding. The Department of Space, Government of India, adopted Aneka as the Cloud computing platform supporting the development of high performance GIS applications [6]. Aneka enables a new approach to complex analyses of massive data and computationally

3. Health and Life Science

With the high volume and density of data, along with the growing complexity of IT ecosystem and the pressures of competition and regulatory groups, life sciences organizations need IT infrastructure and management tools that can respond quickly to changing needs and, more importantly, enable rather than hamper the ability to innovate.

4. IT Education and Research

As the IT field is rapidly moving towards Cloud Computing, software industry's focus is shifting from developing applications for PCs to Data Centres and Clouds that enable millions of users to make use of software simultaneously. This is creating a huge demand for manpower with skills in this area. Educational and research organizations require a platform that can support

- Multiple models of application programming
- Multiple types of Cloud deployments (private, public, or hybrid)
- Extensible framework enabling educators/researchers to develop their own programming models and application schedulers.

V. CONCLUSION

Through performing the above experiments, it has been concluded that CSP can mitigate XSS assaults. However the difficulty in imposing CSP within the internet sites is observed to be browser compatibility disorders as there is no one single typical CSP header that may be outlined for all of the browsers. Also, the website developer does now not comprehend about the browser where customer is going to open the net web page of his internet site. The consumer can use chrome, IE, safari, firefox or any different browsers. If the developer use header (content material-safety-coverage :default-src self), then XSS attack can be mitigated in case of Chrome, Opera and Firefox; but code injection can also be carried out in Safari, IE and other browsers. In a similar fashion, if the developer use header(X-content material-protection-policy: sandbox ..) , then XSS attack may also be mitigated in case of most effective IE however code injection can also be applied in different browsers. As a result, to resolve this incompatibility, a code is written in personal home page that's to be deployed in commencing of the webpage where CSP is to be implemented.

ACKNOWLEDGMENT

At the outset, I thank the Lord Almighty for the grace, strength and hope to make my endeavour a success. I also express my gratitude to Prof. D. S. Kulkarni , Head of the Department and my project Guide for providing me with adequate facilities, ways and means by which I was able to complete this Paper. I express my sincere gratitude to him for his constant support and valuable suggestions without which the successful completion of this project would not have been possible. I thank Prof. Anupkumar Bhongale, of PG Coordinator for his boundless cooperation and helps extended for this Paper. I express my immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Engineering for their cooperation and support. Last but not the least, I thank to IJSART convener and especially my family members who in one way or another helped me in the successful completion of this work. I wish you the best of success. Thank You

REFERENCES

[1] Haneet kour and LalitSen Sharma, "Tracing out cross site scripting vulnerabilities in modern scripts", IJANA Vol7 Issue5, pp. 2862-2867, 2016.

[2] Kailas Patil and Braun Frederik, "A Measurement Study of the Content Secu-rityPolicy on Real-World Applications", International Journal of Network Security, Vol.18, No.2, pp.383-392, Mar-2016.

[3] Isatou Hydara and et al., "Current state of research on cross-site scripting (XSS) A systematic literature review", ELSEVIER Information and Software Technology 58 (170186), pp. 270-276, 2015.

[4] Faisal Mushtaq and S. Aranganathan, " Secure Access Control Requirement Analysis in Cloud Computing",International Journal of Advance Research in Computer Science and Management Studies, Volume 3, Issue 3, pp.328-333, March-2015.

[5] S Divya Bharathy and T Ramesh, "Securing Data Stored in Clouds Using Privacy Preserving Authenticated Access Control", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, pp. 1069-1074, April-2014.

[6] Dang Nguyen, Jaehong Park, and Ravi Sandhu, "Adopting Provenance-Based Access Control in OpenStack Cloud IaaS", Springer International Publishing Switzerland, pp. 15-27, 2014.

[7] Jyoti Malik, Dhiraj Girdhar, Ratna Dahiya, and G. Sainarayanan, "Multifactor Authentication Using a QR Code and a One-Time Password", J Inf Process Syst, Vol.10, No.3, pp.483~490, September 2014.

[8] Abhijit Kumar Nag and Dipankar Dasgupta, "An Adaptive Approach for Continuous Multi-factor Authentication in an Identity Eco-System", 9th Cyber and Information Security Research Conference, pp.65-68, 2014.

[9] Amit Singh and S Sathappan, "A Survey on XSS web-attack and Defence Mechanisms", IJARCSSE, Vol 3 issue 4, pp. 105-109, March 2014.

[10] Sid Stamm et. al., "Reining in the Web with Content Security Policy", Proceedings of the 19th international conference on www. ACM, USA pp 921-930, 2013.

[11] Jake Meredith, Content security policy best practices, https://www.isecparteners.com, July 14, 2013.

[12] Natarajan Meghanathan, "Review of Access Control Models for Cloud Computing", ICCSEA, SPPR, CSIA, WimoA, pp. 77-85, 2013.

[13] S.Shalini and S.Usha, "Prevention of XSS attacks on web applications in the client side", IJCSI, Vol. 8, Issue 4, No1, pp. 24-31 July 2011.

[14] Z. Mao, N. Li, and I. Molloy, "Defeating cross-site request forgery attacks with browser-enforced authenticity protection, In Financial Cryptography and Data Security" 13th International Conference, FC 2009, Accra Beach, Barbados, pp. 23-26, 2009. Revised Selected Papers, pp 238 255, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] M. T. Louw and V N. Venkatakrishnan, "Blueprint: Robust Prevention of Cross-Site Scripting Attacks for existing browser". Proc. 30th IEEE Symp Security and Privacy (SP 09), IEEE CS, pp.331-346, 2009.