ABSTRACT:

The advancement in computer networking and popularity of web application is increasing day by day. Now a days, people relies a lot on various web applications software such as online banking, shopping, webmail, online auction, online examination etc. to do various activities. But due to some underlying deficiencies in the technology, its security issue is also a major concern for the researchers and developers. Attackers logically intrude into the system and destruct the normal and correct functioning of the system. One of the most harmful web application threats is cross-site scripting. In this case, malicious codes are written and inserted in a web document by attackers mainly to steal the session detail or sometimes, for website re-direction to the attacker's site. Attackers execute their attack by storing such malicious code in the server or injecting them in the client's web browser or sometimes, execute such malicious script without storing.

In this project work, we study the various classes of cross-site scripting attacks and present practices of their detection. We observe and identify a suitable and popular method proposed in 'Detection of Cross-site scripting under multiple scenario' by Debashish Das, Dr Utpal Sharma and Dr Dhurba Kumar Battacharya ' and implemented for detection of cross-site scripting attacks. The method consists of a detection module, attached to the web browser and performs additional checking to identify injecting codes of malicious nature. Such checking is called mock-browser and it is performed before the actual browsing of a web application document. To implement the method, we prepared a web-application profile during learning phase with legitimate access. The profile consists of the execution sequences of web document browsing for legitimate access. At run-time the detection method match a sequence retrieved from the web document using the profile in a trie data structure pattern. Here an outgoing or incoming web application document related to the web application transaction, the sequence of function statement called execution sequence, available in the document is extracted. A sequence searching algorithm is executed to search the sequence in the respective WAEP before actual execution by the web browser. Based on the searching results (existence or percentage of matched characters of dynamic scripts or expression) a web application is identified as legitimate or malicious. Our project work is based on this and we have implemented this module on the client side to identify malicious script execution while browsing a web document.