1. Abstract:

Web applications are most widely used technique for providing an access to online services. At the same time web applications are easiest way for vulnerable acts. When a security mechanism is failed in a system, so that attacker may store malicious scripts through vulnerable web application or through blog entries into the storage of trusted web sites. These scripts may be downloaded to a client's browser unknowingly from the trusted site. As a result, users request may be re-directed or defaced to an attacker's site. Due to which sometimes, the malicious script is contracted to full access with all credentials belonging to that legitimate web site. These types of attacks are called Cross-Site Scripting (XSS) attacks.

Cross Site Scripting (XSS) attacks are the most common type of attack against web applications, which allows hackers to inject the malicious script for stealing the user's confidential information. Recent studies show that malicious code detection has become the most challenging task to protect web access from XSS attacks. There are basically, two different types of XSS attacks.

(1) Reflected (or non-persistent) and (2) Stored (or persistent).

In this project, we address both the types of XSS attacks as below:

(i) To defend from reflected attack we develop add-on module and attached with Mozilla web browser. The module works to prevent from malicious input causing XSS attack. By identifying such it will drop the web page and will not be allowed for further execution. The add-on module is implemented based on[USENIX Security] and it works by parsing a runtime URL using a regular expression. Finally, it verifies an URL using a predefined blacklist containing the HTML tags that are responsible for any malicious activity.

(ii) To defend from other type of XSS attack i.e., persistent, we verify user input data by executing a validity checking module using whitelist and escaping. The method is implemented based on the approach proposed in[USENIX Security]. A vulnerable user input data that may have secondary effect and can be occurred an XSS attack unknowingly for a legitimate user access can be stopped.