Injections

PHP Object Injection

PHP Object Injection

- PHP Object Injection attack generally allows an attacker to perform various other different attacks such as Application Denial of Service, SQL Injection, Code Injection.
- This vulnerability usually occurs when the user input is not sanitized properly before passing it to the unserialize()PHP function.
- PHP contains the object serialization feature that could be exploited by attackers to pass serialized strings to a vulnerable unserialize() call.
- Hence, resulting in an arbitrary PHP object injection.

Attack

This attack is only successful if the following conditions are met:

- The web application must have a class which implements a PHP magic method such as __construct and __destruct, which could be used for attacking.
- Classes which are used during the attack should be declared while calling the unserialize() function, if this condition is not met then the object autoloading must be supported for those classes.

Consequences

- The attacker could deface the website.
- The attacker could send spam mails using the vulnerable website.
- The attacker could obtain the database credentials.
- The attacker could gain access to the confidential information.

Mitigations

- Usage of JSON functions instead of the unserialize() function is an important mitigation step to be followed.
- Input Validation must be implemented.
- Source Code Analysis must be performed at the Implementation phase of the Secure SDLC.