Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/90.html,
CWE-90: Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection').

7.9.8 Reflected Injection

Come riconoscerla

La reflection attivata con l'input non verificato dell'utente può, nella migliore delle ipotesi, dare origine a comportamenti imprevisti o causare l'instabilità del sistema. Nel peggiore dei casi, può consentire agli aggressori di iniettare ed eseguire codice dannoso, invocare metodi o classi non previsti all'interno del codice, alterare il flusso logico, manipolare i dati e altro ancora.
La reflection è una tecnica di codifica in cui classi, metodi o funzioni incorporate sono invocate a livello di codice dal loro nome. Se questo nome viene determinato dinamicamente dagli input dell'utente, questi input possono modificare il flusso di codice, invocare codice imprevisto o indesiderato e, a volte, consentire l'inserimento di nuovo codice dannoso.
Si ha reflected injection quando l'applicazione utilizza un input esterno di tipo reflection (dinamico con input via web) per selezionare le classi o il codice da utilizzare, senza effettuare i dovuti controlli.

Come difendersi

- Evitare di utilizzare qualsiasi forma di valutazione dinamica del codice e, in particolare, evitare di utilizzare la reflection se non assolutamente necessario.
- Se non è richiesta un'esecuzione dinamica, utilizzare il flusso logico per determinare quali funzioni eseguire.
- Se è necessaria un'esecuzione dinamica, applicare una lista bianca (white list) di segmenti di codice consentiti, per garantire che il codice arbitrario non possa essere eseguito

Esempio:
Codice vulnerabile:

```
function funzioneHelloWorld($name) {
  return 'Hello '.htmlentities($name);
}
// Se si immette ?function=file_get_contents&arg=/etc/passwd si potrà leggere il
// contenuto del file /etc/passwd
$funcName = isset($_GET['function']) ? $_GET['function'] : "funzioneHelloWorld";;
$arg = isset($_GET['arg']) ? $_GET['arg']: "Guest";;
echo "Output: ";
$func = new ReflectionFunction($funcName);
echo $func->invoke($arg);
```

Codice sicuro chiamato senza Reflection:

```
function funzioneHelloWorld($name) {
  return 'Hello ' . htmlentities($name);
}
$funcName = isset($_GET['function']) ? $_GET['function'] : "funzioneHelloWorld";;
$arg = isset($_GET['arg']) ? $_GET['arg']: "Guest";;
if ($funcName == "funzioneHelloWorld") {
      echo funzioneHelloWorld($arg);
} else {
      echo " Funzione non attendibile!";
}
```

Nel seguente codice la funzione funzioneHelloWorld() è chiamata con Reflection, garantendo che il nome della funzione sia attendibile:

```
$funcName = isset($_GET['function']) ? $_GET['function'] : "funzioneHelloWorld";;
$arg = isset($_GET['arg']) ? $_GET['arg']: "Guest";;
echo "Output: ";
if ($funcName == "funzioneHelloWorld") {
 $func = new ReflectionFunction($funcName);
 echo $func->invoke($arg);
} else {
```