venga attribuito il più basso livello di “diritti” che possano, detenere rimanendo comunque in grado di compiere il proprio lavoro.

Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/94.html Improper Control of Generation of Code ('Code Injection') CWE-94

7.6.3 Command Injection

Come riconscerla

Sfruttando questa vulnerabilità un aggressore potrebbe eseguire comandi di sistema arbitrari sull’application server. Il danno che potrebbe essere arrecato comprende:

- la possibilità di modificare i permessi all’interno di file o directory nel file system (read / create / modify / delete);
- la possibilità di instaurare connessioni di rete non autorizzate verso il server;
- la possibilità di gestire i servizi di sistema, avviandoli, fermandoli o rimuovendoli;
- la completa acquisizione del controllo del server da parte dell'attaccante.

Attraverso questa vulnerabilità l'applicazione viene portata ad eseguire i comandi dell’attaccante. L'operazione spesso viene effettuata utilizzando stringhe di input controllate dall'utente, sulle quali non viene effettuata alcuna verifica.
Potrebbero così essere eseguiti direttamente sul server comandi anche molto pericolosi per il sistema o per la sicurezza dei dati.

Come difendersi

- Rimodulare il codice per evitare una qualsiasi esecuzione diretta di script di comandi. Per effettuare operazioni di sistema, utilizzare eventualmente API fornite dalla piattaforma.
- Se non è possibile fare a meno di lanciare shell dei comandi, assicurarsi tuttavia di eseguire solo stringhe statiche, che non includano l'input dell'utente.
- Validare tutti gli input, indipendentemente dalla loro provenienza. La convalida dovrebbe essere basata su una white list: dovrebbero essere accettati solo i dati che adattano a una struttura specificata, e scartati i dati che non rientrano in questa categoria. I parametri devono essere limitati a un set di caratteri consentito e i caratteri riconosciuti come estranei devono essere filtrati e neutralizzati (escaping). Oltre ai caratteri, occorre controllare il tipo del dato, la sua dimensione, l'intervallo di validità (range), il formato ed eventuali valori attesi (white list).
- Configurare l'applicazione da eseguire utilizzando un account utente limitato che non disponga di privilegi non necessari.
- L’esecuzione del codice dovrebbe utilizzare un account utente separato e dedicato, fornito dei soli privilegi strettamente necessari, in base al principio denominato "Principle of Least Privilege". Il principio stabilisce che agli utenti venga attribuito il più basso livello di “diritti” che possano detenere rimanendo comunque in grado di compiere il proprio lavoro.

Esempio:
Il seguente codice è vulnerabile, poiché se al parametro “comando” viene passato il valore "/ sbin / shutdown" e il server Web è in esecuzione come root, la macchina che esegue il server Web verrà arrestata e non sarà disponibile per richieste future:

```
public IActionResult Run(string nomeFile)
{
  Process p = new Process();
  p.StartInfo.FileName = nomeFile; // Non sicuro
  p.StartInfo.RedirectStandardOutput = true;
  p.Start();
  string output = p.StandardOutput.ReadToEnd();
```