Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/79.html CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

7.7.2 Code Injection

Come riconoscerla

L'applicazione esegue del codice ricevuto attraverso l'input che non è stato sufficientemente verificato. Un utente in grado di inserire codice arbitrario può prendere il controllo dell'applicazione e del server, se non sono state adottate tecniche di difesa in profondità.

Come difendersi.

È vietata qualsiasi esecuzione dinamica di codice ricevuto da canali non attendibili. Se è prorpio necessario compilare ed eseguire dinamicamente del codice dinamico, occorre allore predisporre una sandbox isolata, ad esempio AppDomain di .NET o un thread isolato.
Devono essere effettuati tutti i controlli possibili per validare il codice in ingresso.
Configurare l'applicazione da eseguire utilizzando un account utente limitato che non disponga di privilegi non necessari.
Se è possibile optare per isolare tutta l'esecuzione dinamica utilizzando un account utente separato e dedicato che abbia privilegi solo per le operazioni e i file specifici utilizzati dal codice da eseguire, 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 permette di filtrare eventuale codice dannoso:

```
<%
        strHTML = "<s" & "cript>alert(document.cookie);</s" & "cript>"

        ' code injection
        Response.Write(strHTML)

        ' protetto
        Response.Write(Server.HtmlEncode(strHTML))
%>
```

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

7.7.3 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.