flusso di controllo e della memoria		del codice ci protegge da intere classi di attacco.
	Sfruttare il sistema operativo per la protezione della memoria.	I sistemi operativi di ultima generazione possiedono meccanismi intrinsechi di protezione della memoria.
	Utilizzare il sandboxing	• I sistemi operativi moderni supportano il sandboxing in vari modi (AppArmor su Linux, AppContainer o il modello MOICE su Windows, Sandboxlib su Mac OS). • Non utilizzare per l'esecuzione l'account "nobody", crearne uno nuovo per ciascuna applicazione.
Attacchi di command-injection	Porre la dovuta attenzione durante la fase implementativa della logica di business	Validare l'input in termini di forma e dimensione attesa. Non bonificare. Tracciare l'input nel log e scartarlo se non viene riconosciuto.

Tabella 19 - STRIDE: Indirizzamento dell'Elevation of privilege

Confusione tra dati/codice. Accade spesso che i dati vengono trattati come codice. Attacchi come gli XSS sfruttano l'unione tra codice HTML e dati (un file html che contiene sia codice, come Javascript, che dati, come il testo da visualizzare e talvolta anche istruzioni di formattazione per il testo stesso). Esistono alcune strategie per affrontare tale problematica. Il primo consiste nell'adottare modalità/strumenti che aiutano a mantenere separati codice e dati (ad esempio, i prepared statement in SQL indicano al database quali dichiarazioni aspettarsi e dove saranno posizionati i dati). Un'altra strategia è validare i dati prima di inviarli. Ad esempio, se si stanno inviando dati attraverso una pagina web, è necessario assicurarsi che questi non contengano caratteri come <, >, # o & e quant'altro.
Attacchi di compromissione del flusso di controllo e/o della memoria. Questo insieme di attacchi generalmente sfrutta il "weak typing" e le strutture statiche presenti nei linguaggi simili al C per consentire ad un aggressore di introdurre del codice per poi eseguirlo. Se si utilizza un linguaggio sicuro, come Java o C#, molti di questi attacchi sono più difficili da portare. I sistemi operativi più moderni tendono a incorporare funzioni di protezione e di randomizzazione della memoria, come ad esempio l'Address Space Layout Randomization (ASLR). A volte queste funzioni sono facoltative e richiedono un compilatore o un linker switch. In molti casi, queste funzioni sono disponibili gratuitamente e pertanto dovrebbero essere utilizzate. L'ultima serie di controlli utili a contrastare la compromissione della memoria sono le sandbox. Le Sandbox sono funzioni del sistema operativo progettate per proteggere da un programma danneggiato il sistema operativo stesso e il resto dei programmi dell'utente in esecuzione su di esso.
Attacchi di command-injection. Gli attacchi di command-injection (iniezione di comando) sfruttano i dati di input come vettore di attacco (un aggressore fornisce un carattere di controllo, seguito da una serie di comandi). Per esempio, nella SQL injection, un apice chiude spesso un'istruzione dinamica SQL e quando si tratta di script shell unix, la shell può interpretare un punto e virgola come la fine dell'input, prendendo come comando qualsiasi cosa viene dopo.