In generale, è sufficiente disabilitare la risoluzione automatica di entità esterne e disabilitare il supporto per XInclude, una parte della specifica XML che consente di creare un documento XML a partire da sottodocumenti. Questo di solito può essere fatto tramite opzioni di configurazione o sostituendo a livello di programmazione il comportamento predefinito. Consultare la documentazione per la libreria o l'API che si occupa del parsng dell'XML per dettagli su come disabilitare le funzionalità pericolose e non necessarie.

6.1.4 Insecure Deserialization

Quando dati organizzati in strutture come matrici, record, grafici, classi o altre configurazioni, devono essere archiviate o trasmesse in un'altra posizione, ad esempio attraverso una rete, devono passare attraverso un processo chiamato serializzazione. Questo processo converte e modifica l'organizzazione dei dati in un formato lineare, semplice da trasmettere e da archiviare su dispositivi di storage.
La deserializzazione, al contrario, converte il dato lineare in dato strutturato, istanziando l'oggetto per l'uso da parte del processo di destinazione.
I formati degli oggetti serializzati sono standardizzati in modo da poter essere letti da piattaforme diverse, se necessario. Alcune delle piattaforme che supportano i processi di serializzazione includono python, perl, php, ruby e Java. Anche la piattaforma Microsoft .NET supporta le funzioni di serializzazione con le classi XMLSerializer e DataContractSerializer, nonché le classi BinaryFormatter e NetDataContractSerializer, più potenti ma più vulnerabili. XML, YAML e JSON sono tra i formati di dati serializzati più comunemente utilizzati.
La vulnerabilità di deserializzazione non sicura si presenta nel momento in cui un attaccante è in grado di iniettare dati dannosi all'interno dei dati serializzati. Lo sfruttamento di tale attacco si compie quando dal dato serializzato il processo di destinazione crea un'istanza attiva.

Contromisure

Per mitigare il rischio di attacco attraverso una deserializzazione non sicura è indispensabile ridurre al minimo l'utilizzo della deserializzazione, riducendo i trasferimenti di dati non necessari tra applicazioni / sistemi, riducendo anche la quantità di file scritti su disco.
Occorre, inoltre, aderire al principio del privilegio minimo, minimizzando o disabilitando l'accesso ai privilegi amministrativi per ridurre l'impatto di un possibile attacco andato a buon fine (defense in depth).

6.1.5 Cross Site Scripting (XSS)

Il Cross Site Scripting (XSS) è una problematica solitamente riscontrabile nelle applicazioni Web e consiste nella possibilità di inserire codice HTML o client-side scripting (comunemente Javascript) all'interno di una pagina visualizzata da altri utenti. Un aggressore può, in questo modo, forzare l'esecuzione del codice Javascript all'interno del browser utilizzato dal visitatore.
L'uso più comune del Cross Site Scripting è finalizzato all'intercettazione dei cookie e/o dei token di un utente regolarmente autenticato in un portale e quindi all'appropriazione indebita delle sessioni web da esso intraprese. Con le credenziali rubate, l'attaccante si spaccerà per l'utente legittimo (spoofing).
Esistono diverse forme di Cross Site Scripting, ma il funzionamento di base è sempre lo stesso. A variare è invece la tecnica utilizzata per forzare l'esecuzione di codice Javascript nel browser del visitatore. In alcuni casi un aggressore ha la possibilità di iniettare codice persistente nella pagina web vulnerabile, ovvero codice memorizzato dal server (ad esempio su un database) e riproposto al client durante ogni singolo collegamento. In altre circostanze il codice iniettato non viene memorizzato e la sua esecuzione è resa possibile solamente invogliando l'utente, attraverso tecniche di Social Engineering, a cliccare su un link che punta alla pagina web vulnerabile. In quest'ultimo caso L'URL viene solitamente rappresentato in formato esadecimale (o altre forme) per evitare che l'utente possa identificare il codice Javascript passato come parametro alla pagina stessa. In altri casi l'aggressore può beneficiare di tecniche di url spoofing per mascherare il codice malevolo. Questa tecnica consiste nel mascherare l'url fraudolento al fine di farlo sembrare del tutto simile all'url legittimo sul quale ci si aspetta che l'utente clicchi.

Le vulnerabilità di Cross Site Scripting (XSS) possono essere in particolare sfruttate da un aggressore per: