WEB SECURITY

RISCHI

INPUT NON VALIDI

Idea principale: monitorare dati in ingresso (quantità → contenuto)

Possibili attacchi includono:

- SQL injection
- Cross-site Scripting (XSS)
- Cross-site Forgery (XSRF)
- Clickjacking (XSRF)

molti siti si affidano a validazioni dell'input lato client

ma solo controlli lato server possono evitare questi attacchi.

INJECTION FLAWS

Una speciale attacco di injection di input non validato

L'attaccante prova a iniettare comandi al sistema di back-end (OP. SYSTEM, DATABASE SERVER, SCRIPTING LANGUAGES)

SQL INJECTION

L'attaccante tramite input malevolo cerca a fare restituire dati a cui non dovrebbe accedere.

PER EVITARE INJECTION FLAWS:

- Filtrare gli input
- Limitare le chiamate interfacce esterne
- Scegli chiamate sicure ai sistemi esterni
- Per i database: use statement SQL precompilati
- Controllare i codici di ritorno per individuare gli attacchi.

DOMAIN COMPARISON

tutte le parti (nome del server, porta e protocollo) devono coincidere.

la parte dell'URL relativa al path non conta nulla.

Stessa origine: http://site.com e http://site.com/my/page.html

diverse origine: http://site.org e https://site.com e http://site.com:8080

CROSS-SITE SCRIPTING (XSS)

Al cuore di un tradizionale attacco XSS sta uno script vulnerabile in un sito vulnerabile: lo script legge parte della richiesta HTTP e la mostra alla pagina di risposta senza prima sanificarla.

CROSS-SITE REQUEST FORGERY (CSRF)

È un tipo di attacco che avviene quando un sito malevolo, o email malevola ecc. causano al browser di un utente, un azione non desiderata su un sito web in cui l'utente è attualmente autenticato.

CLICKJACKING

È una tecnica malevola che consiste nel ingannare un web user nell'interagire con qualcosa di diverso dal quale l'utente pensa di stare interagendo.

IMPROPER ERROR HANDLING

I messaggi di errore possono rivelare dettagli sulla tua applicazione, specialmente se contengono stack traces. Il tuo sistema dovrebbe rispondere con corti e chiari messaggi di errore all'utente.

Non distinguere tra "file non trovato" e "accesso negato".

INSECURE STORAGE

Ci possono essere diverse ragioni:

- Salvare dati critici non criptati.
- improprio salvataggio di chiavi, segreti, certificati:
- scelta sbagliata di algoritmi di cifratura
- scarsa fonte di casualità.
- cercare di inventare "nuova" criptografia
- Nessuno fornisce di cambiare la chiave

PREVENTING INSECURE STORAGE:

- Minimizzo la quantità di dati salvati
- Minimizzo l'uso della cifratura (Both)
- Uso algoritmi di cifratura affidabili.
- divido il segreto in parti

DENIAL OF SERVICE

ATTACCO: mandare 1m richieste HTTP fattibili.

PREVENIRE: testo la tua applicazione sotto grandi carichi, Registrazione il numero di richieste per host/user/session

ESERCIZI

ESERCIZIO 3 • LINK VS END-TO-END ENCRYPTION, 20/06/2006

a) Con la link è necessaria una coppia di chiavi per ogni coppia di utenti

FALSO

b) Link encryption ha messaggi in chiaro nell'host di origine e destinazione

VERO

c) con la end-to-end i messaggi sono criptati nei nodi intermedi

FALSO

d) con la link encr. i meccanismi di sicurezza sono invisibili all'utente

VERO

e) con la link encr. puoi decidere quali messaggi criptare

FALSO

d) Con la end-to-end è necessaria una coppia di chiavi per ogni nodo

FALSO

ESERCIZIO 2, 01/2023

Il tuo browser stabilisce una connessione sicura (SSL) con un web server attraverso un certificato valido:

A) NO
B) SI
C) NO
D) NO
E) SI
F) NO

ESERCIZIO 4 Web security, 8/02/2023

Vulnerabile a SQL injection.