ESERCITAZIONE FINALE
(21 gennaio 2026)

Corso: **3770/10840604-011/606/DEC/25**
Titolo: **ESPERTO IN SICUREZZA INFORMATICA – ED. ROVIGO**
Sede: **ROVIGO (RO), Via N. Badaloni 2**

Modulo 3: **MONITORAGGIO DELLA SICUREZZA DEL SISTEMA INFORMATIVO**

Docente: Davide Gessi

Corsista: Giulia Bala

Valutazione: 10/10

Securing Software Esercitazione

1. **Attraverso quali tecniche un attore malevolo potrebbe “craccare” un software, cioè bypassare la registrazione o il pagamento per poterlo usare gratuitamente?**
2. **Distingui la natura dei due tipi di attacchi “cross-site” discussi:**
   - Cross-Site Scripting (XSS)
   - Cross-Site Request Forgery (CSRF)
3. **Perché è necessario “fare l'escape” (cioè neutralizzare) alcuni caratteri nei dati di input?**
4. **Nel contesto di SQL, che cos’è una prepared statement (istruzione preparata)?**
5. **Perché la validazione lato client (client-side validation) è considerata meno sicura rispetto a quella lato server (server-side validation)?**
6. **Riferimento alla vignetta GeekHero**
   Dal punto di vista pratico, la vignetta sopra ha probabilmente ragione nel rappresentare il comportamento della maggior parte degli utenti verso il software open-source.
   Tuttavia, anche se questo fosse la tua opinione, perché potrebbe comunque essere una buona idea usare (o sviluppare) più software open-source, dal punto di vista della sicurezza informatica?
7. **In che modo i package manager (come apt, yum, npm, ecc.) sono simili agli app store (Apple App Store, Google Play Store, Microsoft Store, ecc.) dal punto di vista della cybersecurity?**
8. **Contro quale tipo di minaccia aiuta a difendersi l’uso del campo Content-Security-Policy (CSP) nel nostro codice sorgente?**
9. **Fornisci un esempio concreto di una situazione in cui potresti voler usare il metodo HTTP POST invece del metodo GET.**
10. **Heartbleed (CVE-2014-0160)**
    Il bug noto come Heartbleed, scoperto nel 2014, generò un’enorme preoccupazione su Internet: fu uno dei primi casi in cui una vulnerabilità informatica venne diffusa anche dai media generalisti, mentre i ricercatori di sicurezza cercavano di avvisare il pubblico e incoraggiare un aggiornamento urgente dei sistemi.
    Leggi informazioni su Heartbleed, ad esempio dalla pagina Wikipedia o da altre fonti affidabili (come un video divulgativo).
    Perché Heartbleed rappresentava una minaccia così grave per la sicurezza degli utenti?
    Qua la vignetta obbligatoria xkcd

Risposte:

Un attore malevolo può “craccare” un software per renderlo gratuito tramite reverse engineering, modificando o rimuovendo i controlli di licenza, applicando patch al binario per saltare le verifiche di pagamento, generando chiavi di attivazione false o simulando il server di licenza. **Ottimo.** Risposta molto completa, “simulare il server di licenza” è una chicca.

Il Cross-Site Scripting (XSS) consiste nell’iniettare codice malevolo, solitamente JavaScript, che viene eseguito nel browser dell’utente vittima. Il Cross-Site Request Forgery (CSRF) invece sfrutta l’autenticazione già attiva dell’utente per indurlo a compiere azioni indesiderate su un sito fidato senza il suo consenso.**Perfetto.** XSS e CSRF spiegati in modo pulito e tecnico

È necessario fare l’escape di alcuni caratteri nei dati di input per evitare che questi vengano interpretati come codice eseguibile dal sistema, prevenendo vulnerabilità come XSS, SQL injection o command injection. **Ottimo.** anche *command injection*: segnale di buona visione d’insieme.

Nel contesto SQL, una prepared statement è un’istruzione precompilata in cui la struttura della query è separata dai dati forniti dall’utente, riducendo il rischio di SQL injection perché i parametri non vengono interpretati come codice.**Perfetto.** Definizione corretta

La validazione lato client è considerata meno sicura perché può essere facilmente aggirata o disabilitata dall’utente, mentre solo la validazione lato server è realmente affidabile e sotto il controllo dell’applicazione. **Perfetto.** Senza ambiguità

Dal punto di vista della sicurezza il sistema open-source è vantaggioso perché il codice è pubblico e può essere analizzato da più esperti, le vulnerabilità vengono individuate più rapidamente e si riduce il rischio di backdoor nascoste. **Molto buono.** Corretta e matura; la citazione del problema backdoor è centrata.

Dal punto di vista della cybersecurity i package manager sono simili agli app store perché centralizzano la distribuzione del software, permettono la verifica delle firme e facilitano aggiornamenti e patch di sicurezza automatiche. **Ottimo.** Chiaro il legame con firme, patch e sicurezza operativa.

L’uso del campo Content-Security-Policy (CSP) aiuta a difendersi principalmente da attacchi di tipo XSS, limitando le sorgenti da cui il browser può caricare ed eseguire script e altre risorse. **Corretto.** Essenziale e preciso.

Un esempio concreto di utilizzo del metodo HTTP POST invece del GET è l’invio di credenziali di login e di dati sensibili, oppure operazioni come un pagamento online. **Buono**

Heartbleed rappresentava una minaccia molto grave perché permetteva a un attaccante di leggere porzioni arbitrarie della memoria dei server vulnerabili, esponendo informazioni sensibili come password, cookie di sessione e persino chiavi private SSL, senza lasciare tracce nei log e colpendo una libreria crittografica estremamente diffusa come OpenSSL. **Perfetto**