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: Leonardo Bortoloni

Valutazione: 8/10

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:

1) Un attore malevolo può fare del cracking attraverso tecniche di reverse engineering,creando magari un emulazione di licenza o una validazione di una chiave tramite keygen.
Parziale. Un po' sintetico

2) XSS: il codice viene eseguito nella sessione della vittima tramite i suoi cookie , non riceviamo informazioni ma lui comunque esegue qualcosa ,che può essere uno script malevolo.

CSRF: inserisce una richiesta non effettuata dal utente "tra le righe" a sua insaputa, un esempio è stato l'inserire un link di pagamento amazon con il token di acquisto tramite href di un'immagine.
Sufficiente. XSS ok, CSRF esempio corretto, ma spiegazione un po' sintetica.

3) Bisogna fare l'escape di alcuni caratteri per evitare di incorrere in qualche tipo di injection ,es. code injection tramite XSS= cross site scripting , dove i caratteri da evitare sono : "<,>,etc..".
Giusto ma sintetico. Contesti HTML/SQL/shell ?

4) Il "prepared statement" è un tipo di escaping in SQL , si raddoppiano i caratteri sensibili come l'apostrofo.

1. **Buono.**

5) La client validation è meno sicura rispetto alla server side , un client potrebbe usare i develop tools e modificare alcuni parmetri non consentiti (un utente potrebbe sbloccare una sezione che normalmente è bloccata modificando il codice con i dev tools).
Corretto. Punto chiaro.

6) Un software open-source permette agli utenti di verificare se nel codice ci sono bug , anche gli attaccanti possono esaminarlo , ma gli utenti possono agire a loro volta per creare delle contromisure.
Parziale. Quali sono i vantaggi reali di sicurezza?

7) Gli app store installano applicazioni con permessi limitati,sono controllati da un unica azienda che decide cosa posso o non posso installare e non è sicuro al 100% verso i malware. I packet manager (apt,yum,etc) installano app con già permessi di root e sono app open-source.
Corretto

8) Il CSP è fondamentale nel codice per difendersi principalmente dagli attacchi cross-site scripting conosciuti come XSS,code injection,etc..
Giusto. CSP = anti-XSS, ma avrei voluto qualche dettaglio tecnico

9) Un pulsante funzione che tiene i parametri fuori dall'url , prevenendo CSRF.
Giusto ma sintetico

10)Heartbleed è stata una delle minacce più gravi nella storia di Internet perché colpiva

direttamente
OpenSSL, la libreria di crittografia utilizzata per proteggere circa i due terzi dei server mondiali.
Essa poteva portare alla luce dati sensibili , permetteva furto di chiavi private e credenziali e rimane una minaccia ancora persistente in quanto alcuni sistemi non aggiornati hanno portato a rischi di data breach prolungati.
Giusto. ma avrei voluto qualche dettaglio tecnico