**Ime in priimek:** **Datum:**

# Vaja4: Vrinjeni napadi

**Namen vaje je**:

- Spoznati napad XSS ter kako se pred njim zaščitimo.
- Spoznati napad ob pomoči vrivanja SQL
- Spoznati orodje SQLMAP.
- Spoznati v kako lahko v nekaj vrsticah kode zgradimo preprosti beležnik tipk.
- Spoznati kako lahko s orodjem Ettercap v Linux Kali žrtvi podtaknemo lažno DNS odgovor.

## 1. XSS – Cross Site Scripting

Pri napadih cross-site scripting poskuša napadalec spletno aplikacijo spremeniti tako, da bo ob obisku strani izvedena zlonamerna programska koda - spletni brskalnik obdela vstavljeno zlonamerno programsko kodo kot del spletne strani. Ob pomoči napada XSS lahko napadalec spreminja in poneveri podatke spletne strani in s tem obiskovalca prepriča, da je na znani spletni strani.

**Navodila in vprašanja:**

a) Odprimo spletno stran: https://cybersec.ltfe.org/xss vpišemo ime in se prijavimo.
Komentirajmo dogajanje! Kaj opazimo v URL?

b) Namesto imena v vnosno polje vnesemo testne inpute (html značke), ki jih najdete na spletni strani. Komentirajmo dogajanje!

c) Kako bi povezavo spremenili (skrili) in jo poslali morebitni žrtvi?
Pomagajmo si poljubnim URL spletni enkoderjem/dekoderjem.

d) Na kaj moramo biti kot končni uporabnik pozorni?

e) Naj moramo biti kot razvijalec pozorni pri pisanju spletnih aplikacij?

## 2. SQL injection

Skoraj vse sodobne spletne aplikacije za prikaz spletne strani uporabljajo zbirko podatkov SQL. Vrivanje in izvajanje spremenjenih ukazov SQL imenujemo vrivanje SQL injection.

**Navodila in vprašanja:**

a) Spoznajmo SQL sintakso in vrivanje stavkov.

```
SELECT * FROM users WHERE name='$username' AND password = '$password'
```

Če povpraševanje ne vrne rezultata, v zbirki ni vnesenega uporabnika s takim uporabniškim imenom in geslom.

```
SELECT * FROM users WHERE name='admin' AND password = 'krneki' OR 1 = 1
```

Originalnemu stavku smo vrinili "OR 1 = 1". Ker je ta vedno veljaven (1 = 1 vedno velja), nas aplikacija prijavi kot uporabnika admin, če le-ta seveda obstaja.

b) Odprimo spletno stran: https://cybersec.ltfe.org/sql
Poglejmo izvorno kodo (desni klik –> Inspect ali View Page Source) spletne strani.

c) Logirajmo se v spletno stran! Kakšen vrinjen SQL stavek smo napisali?

d) Odprimo še spletno stran: https://cybersec.ltfe.org/sqlcar
S pomočjo SQL injection tehnike pobrišimo bazo (tabelo)! Komentirajmo dogajanje!

## 3. SQLMAP

Sqlmap je zelo močno odprtokodno orodje za preizkušanje ranljivosti in vdiranje v baze SQL. Omogača vrsto funkcionalnosti od branja vsebin celotnih baz do prevzema nadzora sistema.

**Navodila in vprašanja:**

a) Orodje prenesemo na system iz GitHub na povezave:

```
git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev
```

b) Postavimo se v ustrezno mapo: `cd sqlmap-dev/`

c) Zaženemo čarovnika: `sqlmap –wizard`

d) Orodnje Sqlmap testiramo na povezavi: https://cybersec.ltfe.org/sqlmap/

e) Kometirajmo dogajanje! Katere podatke ste uspeli razbrati?

## 4. LOG injection

Aplikacije običajno uporabljajo dnevniške datoteke (log files) za shranjevanje zgodovine dogodkov ali transakcij za kasnejši pregled, zbiranje statističnih podatkov ali odpravljanje napak. Glede na naravo aplikacije se lahko naloga pregledovanja dnevniških datotek izvaja ročno po potrebi ali avtomatizirano z orodjem, ki samodejno izloči dnevnike za pomembne dogodke ali informacije o trendih. Vrivanje komentarjev poznamo že veliko časa in so ponavadi neškodljivi, razen v nekaterih primerih, kot se je izkazalo v nedavnem primeru Log4J ranljivosti.

**Navodila in vprašanja:**

a) Na primeru https://cybersec.ltfe.org/xss prikažite primer neškodljivega vrivanja komentarja. Vnesite primer.

b) Nedavno se je pojavila Log4J ranljivost. Stopnja ogroženosti (CVSS) te ranljivosti je ocenjena na 9.3 (https://www.cvedetails.com/cve/CVE-2021-44228/), zakaj tako visoka ocena?

c) Na preprostem primeru prikažite delovanje log4J in prikažite primer izkoriščanja log4J ranljivosti!

**Komentar, zapiski vaje:**