**h4 Totally Legit Sertificate**

*Opit murtautumaan weppiin. Nyt päästään kokeilemaan niitä hyökkäyksiä, joita päivällä opittiin. Paljon uusia tehtäviä!*

*Tässä läksyssä on monta uudistettua tehtävää, joita en ehtinyt testata. Jos joku aiheuttaa pulmia, ratkaise se viimeisenä. Käytä työkaluja vain harjoitusmaaleihin, ei muiden ihmisten koneisiin.*

* x) Lue/katso ja tiivistä. (Tässä x-alakohdassa ei tarvitse tehdä testejä tietokoneella, vain lukeminen tai kuunteleminen ja tiivistelmä riittää. Tiivistämiseen riittää muutama ranskalainen viiva.)
  + OWASP 2021: OWASP Top 10:2021
    - [A01:2021 – Broken Access Control](https://owasp.org/Top10/A01_2021-Broken_Access_Control/) (IDOR ja path traversal ovat osa tätä)

-Vähimmäisoikeusperiaatteen tai oletuskiellon rikkominen, jossa käyttöoikeus tulisi myöntää vain tietyille toiminnoille, rooleille tai käyttäjille, mutta se on saatavilla kenelle tahansa.

-Pääsynhallinnan ohittaminen muokkaamalla URL-osoitetta (parametrin manipulointi tai pakotettu selaaminen), sovelluksen sisäistä tilaa, HTML-sivua tai käyttämällä hyökkäystyökalua API-pyyntöjen muokkaamiseen.

-Sallii toisen henkilön tilin katselun tai muokkauksen tarjoamalla sen yksilöllisen tunnisteen (turvattomat suorat objektiviitteet).

-API:n käyttäminen puuttuvilla pääsynhallintamekanismeilla POST, PUT ja DELETE -pyyntöihin.

-Käyttöoikeuksien korottaminen. Toimiminen käyttäjänä ilman kirjautumista tai adminina kirjautuneena tavallisena käyttäjänä.

-Metatiedon manipulointi, kuten JSON Web Token (JWT) pääsynhallintamerkin uudelleenlähetys tai manipulointi, tai evästeen tai piilotetun kentän muokkaaminen käyttöoikeuksien korottamiseen tai JWT:n mitätöinnin väärinkäyttöön.

-CORS-väärinkonfigurointi mahdollistaa API-pääsyn luvattomista/luottamattomista lähteistä.

-Pakotettu selaaminen autentikoituihin sivuihin autentikoimattomana käyttäjänä tai privilegioituihin sivuihin tavallisena käyttäjänä.

**Kuinka estää:**

-Lukuun ottamatta julkisia resursseja, käytä oletusarvoisesti kieltämisperiaatetta.

-Toteuta pääsynhallintamekanismit kerran ja käytä niitä uudelleen koko sovelluksessa, mukaan lukien Cross-Origin Resource Sharing (CORS) käytön minimointi.

-Mallin pääsynhallintojen tulisi valvoa tietueen omistajuutta sen sijaan, että hyväksyttäisiin, että käyttäjä voi luoda, lukea, päivittää tai poistaa mitä tahansa tietuetta.

-Yksilölliset sovellusliiketoiminnan rajoitusvaatimukset tulisi valvoa toimialamalleilla.

-Poista web-palvelimen hakemistolistaus käytöstä ja varmista, että tiedostometatietoja (esim. .git) ja varmuuskopiotiedostoja ei ole web-juurissa.

-Kirjaa pääsynhallinnan epäonnistumiset, hälytä ylläpitäjiä tarvittaessa (esim. toistuvat epäonnistumiset).

-Rajoita API:n ja ohjaimen pääsyä määrällisesti vähentääksesi automatisoitujen hyökkäystyökalujen aiheuttamaa vahinkoa.

-Tilaan perustuvat istuntotunnisteet tulisi mitätöidä palvelimella uloskirjautumisen jälkeen. Tilaamattomat JWT-tunnukset tulisi olla lyhytaikaisia, jotta hyökkääjän mahdollisuusikkuna minimoidaan. Pidempikestoisille JWT-tunnisteille on erittäin suositeltavaa noudattaa OAuth-standardeja pääsyn peruuttamiseksi.

Lähde ChatGPt käännös, translate into Finnish

* + - [A10:2021 – Server-Side Request Forgery (SSRF)](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/)
* SSRF-virheet tapahtuvat aina, kun web-sovellus noutaa etäresurssin validioimatta käyttäjän toimittamaa URL-osoitetta. Se mahdollistaa hyökkääjän pakottaa sovelluksen lähettämään muokatun pyynnön odottamattomaan kohteeseen, vaikka se olisi suojattu palomuurilla, VPN:llä tai muulla verkon pääsynhallintalistalla (ACL).
* Kun modernit web-sovellukset tarjoavat loppukäyttäjille käteviä ominaisuuksia, URL-osoitteen noutaminen muodostuu yleiseksi skenaarioksi. Tämän seurauksena SSRF:n esiintyminen on lisääntymässä. Lisäksi SSRF:n vakavuus kasvaa pilvipalvelujen ja arkkitehtuurien monimutkaisuuden vuoksi.

**Kuinka estää:**

Verkkokerroksesta:

"Segmentoi etäresurssien käyttötoiminnot erillisiin verkkoihin SSRF:n vaikutusten vähentämiseksi."

-Valvo 'kiellä oletusarvoisesti' -palomuurikäytäntöjä tai verkon pääsynhallintasääntöjä, jotta kaikki muu paitsi välttämätön intranet-liikenne estetään."

Sovellus kerroksesta:

-Puhdista ja validoi kaikki asiakkaan toimittamat syöttötiedot.

-Valvo URL-kaavaa, porttia ja kohdetta positiivisen sallimislistan avulla.

-Älä lähetä raakoja vastauksia asiakkaille.

-Poista HTTP-uudelleenohjaukset käytöstä.

-Ole tietoinen URL:n johdonmukaisuudesta välttääksesi hyökkäyksiä, kuten DNS-uudelleensidonta ja 'tarkistuksen ajan, käytön ajan' (TOCTOU) kilpailutilanteet.

-Älä lievennä SSRF-ongelmaa käyttämällä kieltolistaa tai säännöllisiä lausekkeita. Hyökkääjillä on kuormalistoja, työkaluja ja taitoja kieltolistojen ohittamiseen.

Lähde ChatGPt käännös, translate into Finnish

* + PortSwigget Academy:
    - [Access control vulnerabilities and privilege escalation](https://portswigger.net/web-security/access-control) (IDOR on osa tätä)

Pääsynhallinta on rajoitusten soveltamista siihen, kuka tai mikä on valtuutettu suorittamaan toimenpiteitä tai käyttämään resursseja. Web-sovellusten kontekstissa pääsynhallinta perustuu autentikointiin ja istunnonhallintaan:

Autentikointi vahvistaa, että käyttäjä on se, kuka sanoo olevansa.

Istunnonhallinta tunnistaa, mitkä seuraavista HTTP-pyynnöistä tehdään saman käyttäjän toimesta.

Pääsynhallinta määrittää, onko käyttäjällä oikeus suorittaa yrittämäänsä toimintoa.

Rikkinäiset pääsynhallintamekanismit ovat yleisiä ja usein esittävät kriittisen turvallisuusongelman. Pääsynhallinnan suunnittelu ja hallinta on monimutkainen ja dynaaminen ongelma, joka soveltaa liiketoiminnallisia, organisaation ja laillisia rajoituksia tekniseen toteutukseen. Pääsynhallinnan suunnittelupäätökset on tehtävä ihmisten toimesta, joten virheiden mahdollisuus on suuri.

kuinka estää:

Pääsynhallinnan haavoittuvuudet voidaan estää ottamalla käyttöön monitasoinen puolustus ja soveltamalla seuraavia periaatteita:

Älä koskaan luota pelkkään hämärtämiseen pääsynhallinnassa.

Ellei resurssi ole tarkoitettu julkisesti saataville, kiellä pääsy oletusarvoisesti.

Käytä mahdollisuuksien mukaan yhtä sovelluslaajuista mekanismia pääsynhallinnan valvontaan.

Kooditasolla tee kehittäjille pakolliseksi ilmoittaa, mikä pääsy on sallittu kullekin resurssille, ja kiellä pääsy oletusarvoisesti.

Tarkasta ja testaa pääsynhallinta perusteellisesti varmistaaksesi, että se toimii suunnitellusti.

Lähde ChatGPT käännös, translate into Finnish

* + - [Server-side template injection](https://portswigger.net/web-security/server-side-template-injection)

Palvelinpuoleinen mallineen injektio tapahtuu, kun hyökkääjä pystyy käyttämään natiivia mallinekieltä syöttääkseen haitallisen kuorman mallineeseen, joka sitten suoritetaan palvelimen puolella.

Mallinekoneet on suunniteltu tuottamaan verkkosivuja yhdistämällä kiinteitä mallineita vaihteleviin tietoihin. Palvelinpuoleiset mallineen injektiohyökkäykset voivat tapahtua, kun käyttäjän syöte liitetään suoraan mallineeseen, sen sijaan että se välitettäisiin datana. Tämä antaa hyökkääjille mahdollisuuden injektoida mielivaltaisia mallineohjeita manipuloidakseen mallinekonetta, usein mahdollistaen heille täydellisen hallinnan palvelimen yli. Kuten nimi viittaa, palvelinpuoleiset mallineen injektiokuormat toimitetaan ja arvioidaan palvelimen puolella, mikä tekee niistä mahdollisesti paljon vaarallisempia kuin tyypilliset asiakaspuolen mallineen injektiot.

**Miten estää palvelinpuoleiset mallineen injektiohaavoittuvuudet**

Paras tapa estää palvelinpuoleiset mallineen injektiot on olla sallimatta käyttäjien muokata tai lähettää uusia mallineita. Tämä on kuitenkin joskus välttämätöntä liiketoimintavaatimusten vuoksi.

Yksi yksinkertaisimmista tavoista välttää palvelinpuoleiset mallineen injektiohaavoittuvuudet on aina käyttää "logiikattomia" mallinekoneita, kuten Mustache, ellei se ole ehdottoman välttämätöntä. Logiikan erottaminen esityksestä mahdollisimman paljon voi suuresti vähentää altistumistasi vaarallisimmille mallinepohjaisille hyökkäyksille.

Toinen toimenpide on suorittaa käyttäjien koodi hiekkalaatikkoympäristössä, josta mahdollisesti vaaralliset moduulit ja funktiot on poistettu kokonaan. Valitettavasti epäluotettavan koodin hiekkalaatikointi on luonteeltaan vaikeaa ja altis ohituksille.

Lopuksi toinen täydentävä lähestymistapa on hyväksyä, että mielivaltainen koodin suorittaminen on lähes väistämätöntä ja soveltaa omaa hiekkalaatikointiasi käyttämällä mallineympäristöäsi esimerkiksi tiukasti rajoitetussa Docker-säiliössä.

Lähde ChatGPT käännös artikkelin osista, translate into Finnish

* + - [Server-side request forgery (SSRF)](https://portswigger.net/web-security/ssrf)

SSRF-hyökkäykset hyödyntävät usein luottamussuhteita laajentaakseen hyökkäystä haavoittuvasta sovelluksesta ja suorittaakseen luvattomia toimia. Nämä luottamussuhteet voivat olla olemassa palvelimeen liittyen tai muihin saman organisaation taustajärjestelmiin liittyen.

Lähde ChatGPT käännös artikkelin osista, translate into Finnish

* + - [Cross-site s<pcripting](https://portswigger.net/web-security/cross-site-scripting)

Poikkisivujen skriptaus (tunnetaan myös nimellä XSS) on verkkoturvallisuuden haavoittuvuus, joka mahdollistaa hyökkääjän vaarantaa käyttäjien vuorovaikutukset haavoittuvan sovelluksen kanssa. Se mahdollistaa hyökkääjän kiertää saman alkuperän politiikan, joka on suunniteltu eristämään eri verkkosivustot toisistaan. Poikkisivujen skriptauksen haavoittuvuudet sallivat yleensä hyökkääjän esiintyä uhri-käyttäjänä, suorittaa kaikki toimet, joita käyttäjä pystyy tekemään, ja päästä käsiksi mihin tahansa käyttäjän dataan. Jos uhri-käyttäjällä on etuoikeutettu pääsy sovelluksessa, hyökkääjä saattaa pystyä saavuttamaan täyden hallinnan sovelluksen kaikkien toimintojen ja datan yli.

Estääksesi poikkisivujen skriptaus (XSS) -hyökkäykset, tehokas suojaus saattaa edellyttää useiden seuraavien toimenpiteiden yhdistelmää:

Suodata tietoja saapuessa. Siinä vaiheessa, kun käyttäjän syöte vastaanotetaan, suodata se mahdollisimman tiukasti perustuen siihen, mikä on odotettua tai kelvollista syötettä.

Koodaa tiedot lähtiessä. Siinä vaiheessa, kun käyttäjän hallinnoima data tuotetaan HTTP-vastauksissa, koodaa tulos estääksesi sen tulkinnan aktiivisena sisältönä. Riippuen tulostusyhteydestä, tämä voi vaatia HTML-, URL-, JavaScript- ja CSS-koodauksen yhdistelmien soveltamista.

Käytä sopivia vastausotsikoita. Estääksesi XSS:n HTTP-vastauksissa, jotka eivät ole tarkoitettu sisältämään HTML:ää tai JavaScriptiä, voit käyttää Content-Type- ja X-Content-Type-Options -otsikoita varmistaaksesi, että selaimet tulkitsevat vastaukset niin kuin olet tarkoittanut.

Sisältöturvapolitiikka. Viimeisenä puolustuslinjana voit käyttää Content Security Policy (CSP) -toimenpiteitä vähentämään mahdollisesti esiintyvien XSS-haavoittuvuuksien vakavuutta.

Lähde ChatGPT käännös artikkelin osista, translate into Finnish

* + Karvinen 2020: [Using New Webgoat 2023.4 to Try Web Hacking](https://terokarvinen.com/2023/webgoat-2023-4-ethical-web-hacking/)

Artikkeli käsittelee WebGoatin asennusta ja sen käyttämistä vaihtoehtoisessa tietoliikenne porttinumerossa.

* a) Totally Legit Sertificate. Asenna OWASP ZAP, generoi CA-sertifikaatti ja asenna se selaimeesi. Laita ZAP proxyksi selaimeesi. Osoita, että hakupyynnöt ilmestyvät ZAP:n käyttöliittymään. (Ei toimi localhost:lla ilman Foxyproxya)  
    
  A screenshot of a computer program

  Description automatically generated  
  Latasin Zap crossplatform paketin ja purin sen kansioon home/ilkkah/Downloads/Zap\_2.14.0 kansioon

”sudo apt-get update”

”sudo apt-get install default-jre”

A screenshot of a computer

Description automatically generated

komento “java -jar zap\_2.14.0” käynnisti Zapin. Tässä vaiheessa siirryin tehtävään b) ja asensin FoxyProxy lisäosan Firefoxiin ja sitten palasin takaisin tähän tehtävään.

A screenshot of a computer

Description automatically generated

Kävin tarkastamassa Zapin asetuksista osoitteen (localhost) ja portin 8080

A screenshot of a computer

Description automatically generated

Foxy Proxyn astuksita lisäsin uuden ja asetin IP osoitteeksi kuvasta poiketen localhost ja portti 8080 niin kuin Zapissa oli asetuksissa.

A screenshot of a computer error

Description automatically generated

Menin osoitteeseen google.com Firefoxilla ja tulos oli tämä Firefox varoitti turvattomasta yhteydestä.

A screenshot of a computer

Description automatically generated

Loin Zapilla sertifikaatin ja lisäsin sen Firefoxiin

A screenshot of a computer

Description automatically generated

Menin Firefoxilla osoitteeseen [www.google.com](http://www.google.com) ja zap toimi nyt välimies proxyna ja tallensi liikenteen. Sertifikaatti valitusta ei Firefox enää antanut.

* b) Kettumaista. Asenna FoxyProxy Standard Firefox Addon, ja lisää ZAP proxyksi siihen.

A blue background with white text

Description automatically generated

Asensin Foxy Proxy Standardin Fitefoxin extensions valikosta.

* PortSwigger Labs. Ratkaise tehtävät. Selitä ratkaisusi: mitä palvelimella tapahtuu, mitä eri osat tekevät, miten hyökkäys löytyi, mistä vika johtuu. (Voi käyttää ZAPia, vaikka malliratkaisut käyttävät harjoitusten tekijän maksullista ohjelmaa)
  + Insecure Direct Object Reference (IDOR)
    - c) [Insecure direct object references](https://portswigger.net/web-security/access-control/lab-insecure-direct-object-references)

En oikein saanut selvää mitä tässä pitää tehdä ja en päässyt eteenpäin. Kokeilin Zapin Hudia, mutta en ottanut kuvia. En Päässyt sitä loppuun asti, kun en tajunnut kaikkia ominaisuuksia mistä keyt olisi pitänyt löytää.

Kokeilin kuitenkin tehtää mutta, en löytänyt salasanaa nimettyäni tiedostonimen, mutta tämä johtui siitä että en perehtynyt tarpeeksi ZAPin käyttöön. Yritin jotenkin väärällä tavalla. Tuntui myös, että ZAPIN käyttö hidasti konetta merkittävästi ja nettiliikennettä.

* + Path traversal
    - d) [File path traversal, simple case](https://portswigger.net/web-security/file-path-traversal/lab-simple)
    - e) [File path traversal, traversal sequences blocked with absolute path bypass](https://portswigger.net/web-security/file-path-traversal/lab-absolute-path-bypass)
    - f) [File path traversal, traversal sequences stripped non-recursively](https://portswigger.net/web-security/file-path-traversal/lab-sequences-stripped-non-recursively)  
        
      <
  + Server Side Template Injection (SSTI)
    - g) [Server-side template injection with information disclosure via user-supplied objects](https://portswigger.net/web-security/server-side-template-injection/exploiting/lab-server-side-template-injection-with-information-disclosure-via-user-supplied-objects) (Tämä on merkitty hieman vaikeammaksi, jätä viimeiseksi jos näyttää hankalalta)
  + Server Side Request Forgery (SSRF)
    - h) [Basic SSRF against the local server](https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost)  
        
      En kokeillut vielä tätä, palaan tähän
  + Cross Site Scripting (XSS)
    - i) [Reflected XSS into HTML context with nothing encoded](https://portswigger.net/web-security/cross-site-scripting/reflected/lab-html-context-nothing-encoded)
    - j) [Stored XSS into HTML context with nothing encoded](https://portswigger.net/web-security/cross-site-scripting/stored/lab-html-context-nothing-encoded)  
        
      En kokeillut vielä näitä, palaan näihin
* k) Asenna Webgoat 2023.4. (Uusi versio, jossa on eri tehtäviä kuin vanhemmissa)  
    
  <https://terokarvinen.com/2023/webgoat-2023-4-ethical-web-hacking/>

seurasin Teron ohjeita WebGoatin asennuksessa  
  
Skippasin javan asennuksen, kun se oli jo asennettu.

A screenshot of a computer program

Description automatically generated

komento ”sudo apt-get install ufw” asensin palomuurin

“sudo ufw enable” käynnistin palomuurin

“sudo ufw status” katsoin palomuurin tilan ja se oli aktiivinen

komento ”wget <https://github.com/WebGoat/WebGoat/releases/download/v2023.4/webgoat-2023.4.jar>” latasin WebGoatin koneelle

A computer screen shot of a program code

Description automatically generated

komento ”java -Dfile.encoding=UTF-8 -Dwebgoat.port=8888 -Dwebwolf.port=9090 -jar webgoat-2023.4.jar” käynnisti WebGoatin.

A screenshot of a login box

Description automatically generated

Menin selaimella osoitteeseen <http://127.0.0.1:8888/WebGoat> ja aukesi WebGoatin login näkymä.

* l) (Poistin kohdan l, helppo tehtävä johon olisi mennyt turhaa aikaa.)
* Ratkaise WebGoat 2023.4:
  + m) (A1) Broken Access Control (WebGoat 2023.4)
    - Hijack a session (1)
    - Insecure Direct Object References (4)
    - Missing Function Level Access Control (3)
    - Spoofing an Authentication Cookie (1)
  + n) (A7) Identity & Auth Failure (WebGoat 2023.4)
    - Authentication Bypasses (1)
    - Insecure Login (1)
  + o) (A10) Server-side Request Forgery (WebGoat 2023.4)
    - Server-Side Request Forgery (2)
  + p) Client side (WebGoat 2023.4)
    - Bypass front-end restrictions (2)
* Vapaaehtoinen: Ratkaise lisää WegGoat-tehtäviä.
* Vapaaehtoinen: Ratkaise lisää PortSwigger Labs -tehtäviä.

Vinkit

* Firefox ei enää ohjaa proxyyn liikennettä mihinkään 127-alkuiseen IP-osoitteeseen eikä localhostiin. Mutta Foxyproxyn avulla se onnistuu.
* [OWASP ZAP Download-sivun](https://www.zaproxy.org/download/) sopivin JAR on "Cross Platform Package", noin 250 MB.
* Proxya voi testata vaikkapa Metasploitablen weppipalveluiden kanssa.
  + Älä tee hyökkäyksiä mihinkään muualle kuin harjoitusmaaleihin
* ZAP TLS-yhteyksien avaamiseen CA-sertifikaatti löytyy Tools: Options alta, hakusanaksi "Certificate".
  + Firefoxissa Settings ja hakusanaksi "Certificate"
* ZAP hakupyynnön pysäytys
  + Puussa olevista osoitteista oikeaa nappia, menusta "Break..."
  + Tee pyyntö selaimella
  + Pyyntö näkyy ZAPissa oikealla ylhäällä "Break" välilehdellä.
  + Muokaa pyynnön tekstiä, lähetä eteenpäin yläreunan työkalupallin Play-symbolia klikkaamalla
* ZAPin oppaita pikaisella nettihaulla
  + Kulonpää 2021: [Tunkeutumistestaus vko. 5](https://kulonpaa.com/?p=338)
  + South Dakota Information Security 2018 - [OWASP ZAP Tutorial - Part 1: Intercepting Traffic](https://rkhal101.github.io/_posts/WAVS/ZAP/zap_browser_setup)
* PortSwiggerin tehtävissä voi käyttää välimiesproxyna ZAP:ia. Ei siis tarvitse ostaa noiden harjoitusten tekijän hienoa ohjelmaa.
* Viittaa
  + Viittasithan kurssiin ja tehtäväsivuun?
  + Viittasithan kaikkiin käyttämiisi artikkeleihin?
  + Viittasithan muiden tehtäväraportteihin, jos käytit lähteenä?
  + Videoihin, man-sivuihin, ohjelmiin, StackOverflowhun...
  + Viittasithan kaikkiin muihinkin lähteisiin?
* Hakkeroimaan oppii hakkeroimalla - ja kun olet ratkaissut nämä tehtävät, olet hakkeroinut taas melkoisen setin.

**LUNTO 5?**

Karvinen 2016: [PostgreSQL Install and One Table Database – SQL CRUD tutorial for Ubuntu](https://terokarvinen.com/2016/03/05/postgresql-install-and-one-table-database-sql-crud-tutorial-for-ubuntu/) (Virtuaaliympäristöä ei tarvita, voit aloittaa kohdasta "Three line install"

**Asennus**

“sudo apt-get update”

“sudo apt-get -y install postgresql”

“sudo systemctl start postgresql”

“sudo -u postgres createdb $(whoami)”

“sudo -u postgres createuser $(whoami)”

“psql”

**Apu**

“help”

\h SQL komennoille

\? psql komennoille

q exits help

“\h select”

**CRUD, Create, Read, Update, Delete**

**C: luo taulu**

“CREATE TABLE students (id SERIAL PRIMARY KEY, name VARCHAR (200));”

**C: Luo sisältöä INSERT**

“INSERT INTO students(name) VALUES ('Matti');”

“INSERT INTO students(name) VALUES ('Maija');”

“INSERT INTO students(name) VALUES ('Liisa') RETURNING id, name;”

**R: Lue: SELECT**

“SELECT \* FROM students;”

“SELECT \* FROM students WHERE name LIKE 'Ma%';

**U: Päivitä UPDATE**

“UPDATE students SET name='Tero Karvinen' WHERE name='Tero';”

“UPDATE 1”

**D: Poista DELETE**

**“DELETE FROM students WHERE name='Liisa';”**

**“DELETE 1”**

OWASP 2017: [A1:2017-Injection](https://owasp.org/www-project-top-ten/2017/A1_2017-Injection)

**Esimerkki hyökkäys skenaariot**Ensimmäinen skenaario:

String query = "SELECT \* FROM accounts WHERE custID='" + request.getParameter("id") + "'";

Toinen Skenaario:

Query HQLQuery = session.createQuery("FROM accounts WHERE custID='" + request.getParameter("id") + "'");

Molemmissa skenaariossa hyökkääjä modifioi ”id” parametriä selaimessaan lähettääkseen ’ or ’1’=’1

Esimerkki: http://example.com/app/accountView?id=' or '1'='1

Tämä muuttaa kummassakin skenaarioissa haun merkityksen ja palauttaa kaikki tiedot tietokannassa ”accounts” taulusta. Käsittääkseni tuo OR ’1’=’1’ aina tosi ja tästä syystä palautuu kaikki tiedot.

PortSwigger Academy: [SQL injection](https://portswigger.net/web-security/sql-injection)

* + Kaikki muut luvut paitsi ei "Blind SQL injection"
  + Tämä kappale kannattaa pitää näkyvissä injektioita tehdessä [SQL injection cheat sheet](https://portswigger.net/web-security/sql-injection/cheat-sheet)
* a) CRUD. Tee uusi PostgreSQL-tietokanta ja demonstroi sillä create, read, update, delete (CRUD). Keksi taulujen ja kenttien nimet itse. Taulujen nimet monikossa, kenttien nimet yksikössä, molemmat englanniksi.

“sudo apt-get update”

“sudo apt-get -y install postgresql”

“sudo systemctl start postgresql”

“sudo -u postgres createdb $(whoami)”

“sudo -u postgres createuser $(whoami)”

A screen shot of a computer

Description automatically generated

Tietokanta ja käyttäjä on luotu ja käynnistin tietokannan ”psql” komennolla

A black background with white text

Description automatically generated

“CREATE TABLE users (id SERIAL PRIMARY KEY, email VARCHAR (200));” Tuli virheilmoitus, enkä saanut luotua taulua users.

* b) SQLi me. Kuvaile yksinkertainen SQL-injektio, ja demonstroi se omaan tietokantaasi psql-komennolla. Selitä, mikä osa on käyttäjän syötettä ja mikä valmiina ohjelmassa. (Tässä harjoituksessa voit vain kertoa koodista, ei siis tarvitse välttämättä koodata sitä ohjelmaa, joka yhdistää käyttäjän syötteen SQL:n)

Helppo SQL injektio, muuttaa selaimessa palautuvaa osoitetta esim. -- joka tarkoittaa kommenttia, jolloin takana olevaa SQL koodia ei lueta.

* PortSwigger Labs
  + c) (Alakohta c poistettu, tämänhän ratkoimme jo aiemmin: [SQL injection vulnerability in WHERE clause allowing retrieval of hidden data](https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data))
  + d) [SQL injection vulnerability allowing login bypass](https://portswigger.net/web-security/sql-injection/lab-login-bypass)  
      
    A screenshot of a computer

    Description automatically generated

Kaapasin liikenteen Zapilla ja löysin sieltä oikean kohdan jossa syöttämäni tiedot näkyivät. Muokkasin kohtaa username=administrator’--, joka siis kommentoi password kohdan pois ja näin sain ratkaistuksi tehtävän. En katsonut ohjetta tähän, mutta tämän ratkaisuhan luki saamissamme Portswigger teoria materiaalissa.

* + e) [SQL injection attack, querying the database type and version on Oracle](https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-oracle)

A screenshot of a computer

Description automatically generated

Katsoin ratkaisun mutta en silti saanut syötettyä annettua injektiota oikeaan paikkaan. Yritin muokata category parametria, seuraavalla injektiolla '+UNION+SELECT+'abc','def'+FROM+dual—tuloksetta.

* + f) [SQL injection attack, querying the database type and version on MySQL and Microsoft](https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-mysql-microsoft)  
      
    Kun en pääsyt aikaisemmasta eteenpäin. Katsoin vihjeen, ratkaisun ja videon miten tämä tehdään.

A screenshot of a computer

Description automatically generated

Kokeilin syöttää videon ohjeiden muikaisesti Gifts kaegorian perään '+UNION+SELECT+'abc','def'#

Serverin vastaus oli OK ja kävin katsomassa nettisivua Firefoxilla ja tosissaan sinne oli tulostunut alas abc ja def ja näin pystyi todentamaan, että SQL haavoittuvuus oli löytynyt.

A black text on a white background

Description automatically generated

MySQL versio 8.0.35 joka varmaankin pyöri Linuxin vanhassa versioss 20.04.1 Muistaakseni LTS versio, joka alkaa kyllä olemaan jo vanha.

g) [SQL injection attack, listing the database contents on non-Oracle databases](https://portswigger.net/web-security/sql-injection/examining-the-database/lab-listing-database-contents-oracle)  
  
Onko otsikossa virhe? linkki ohjasi nimenomaan contets on Oracle database?  
  
Katsoin vihjee ja ratkaisun

* + +UNION+SELECT+'abc','def'+FROM+dual--  
      
    Tämän kum syötti requesteriin ja lähetti, tulostui taas abc ja def sivulle, haavoittuvuus oli olemassa.

Kokeilin syöttää toisen vastauksessa olevan SQL injektion, serveri vastasi 200 ok ja katsoin tuloksen Firefoxissa.

A white background with black and white text

Description automatically generated

En kuitenkaan löytänyt sieltä heti silmiinpistävää kategoriaa, joka sisältäisi käyttäjänimet ja salasanat. Joten päätin katsoa ensimmäisen Community solution videon, jota Portswigger tarjosi, video varmaankin oli vanhalle tehtävälle, koska taulujen nimet eivät täsmänneet.

Katsoin vastauksesta, että sieltä olisi pitänyt löytyä USERS\_ABCDEF niminen taulu. Syötin alla olevan Zapin requesteriin.  
  
'+UNION+SELECT+column\_name,NULL+FROM+all\_tab\_columns+WHERE+table\_name='USERS\_ABCDEF'—

A screenshot of a computer

Description automatically generated

Sain tällaisen vastauksen, joku ei nyt menny oikein.

A screenshot of a computer

Description automatically generated

Muokkasin komentoa USERS\_EUVJOW muotoon, jonka löysin sivujen table listauksesta, joka oli tulostunut tuolla aikasemmassa vaiheessa. ja lähetin komennon. Tämä taulunimi siis poikkesi videon taulun nimestä.

A close-up of a computer screen

Description automatically generated

Taas tällainen vastaus, mutta nyt katsoin sivun alas ja siellä oli:

A screenshot of a white text

Description automatically generated

PASSWORD\_DJAGUI

USERNAME\_SNFRGW

Näistä muodostin injektion: ’ UNION select USERNAME\_SNFRGW, PASSWORD\_DJAGUI from USERS\_EUVJOW—

se toimi:

A screenshot of a computer code

Description automatically generated

Kirjauduin adminina sisään ja labra oli ratkaistu

* + h) [SQL injection UNION attack, determining the number of columns returned by the query](https://portswigger.net/web-security/sql-injection/union-attacks/lab-determine-number-of-columns)
  + i) [SQL injection UNION attack, retrieving data from other tables](https://portswigger.net/web-security/sql-injection/union-attacks/lab-retrieve-data-from-other-tables)
  + j) [SQL injection UNION attack, retrieving multiple values in a single column](https://portswigger.net/web-security/sql-injection/union-attacks/lab-retrieve-multiple-values-in-single-column)
* k) Vapaaaehtoinen: Mitmproxy. Demonstroi mitmproxy:n käyttöä terminaalista (TUI tai CLI).
* l) Vapaaehtoinen: Attack lab. Asenna vapaavalintainen kone [Vulnhubista](https://www.vulnhub.com/) ja murtaudu siihen.
* Vinkit
  + Aloita ajoissa, niin voit tehdä tehtäviä monena päivänä.
  + Useimmat SQL-injektiotehtävät tässä läksyssä voinee ratkaista pelkällä selaimella, joskin välimiesproxylla se lienee mukavampaa.
  + Tehtäviä saa täydentää. Jos edellisen kerran IDORit ja muotti-injektiot tuntuvat oppitunnin jälkeen selkeämmiltä, voit tehdä ne nyt. Merkitse silloin vanhan raportin sekaan viittaus [Update 2023-11-21 w47 Tue: Jatkoin tehtävän tekoa tänään, uusi ratkaisu raportin lopussa] ja siitä linkki raportin loppuun. On tärkeää, että raportti säilyy täsmällisenä ja totuudenmukaisena kuvauksena tapahtumista. Historia ei voi muuttua.
  + Tarkista, että jokainen tehtäväraporttisi viittaa tähän tehtäväsivuun ja kaikkiin muihinkin lähteisiin.
  + Haastava kohde?
    - Mieti vielä eri vaihtoja ja lähestymistapoja.
    - Mitä lukuläksy opettaa aiheesta?
    - Tee muita tehtäviä ensin, jätä vaikeat viimeiseksi.
    - Löytyykö jostain eri tehtävän ratkaisu, joka on samantapainen?
    - Yritä kovemmin, ja raportoi selkeämmin.
    - Tee pari punnerrusta tai jaloittele ulkona.
    - Jos muu ei auta: walktrough.
      * Jos tehtävä ei ratkea, niin lopuksi kannattaa etsiä esimerkkiratkaisu juuri tähän tehtävään, ja käyttää sitä.
      * Moniosaisessa tehtävässä (esim jotkut VulnHub-koneet) voi katsoa läpikävelystä yhden vaiheen, ja sitten kokeilla seuraavaa.
      * Läpikävelyohjeeseen pitää viitata. Raporttiin tulee selkeästi merkitä, missä vaiheessa on käytetty läpikävelyohjetta.
  + VulnHub
    - Difficulty: Easy. Toki saat ottaa minkä koneen vain, mutta helpoista kannattaa aloittaa.
    - Voit kokeilla eri koneita.
    - Jos jäät jumiin, läpikävely koneelle foo löytynee hakukoneesta: vulnhub foo walktrough