**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 scripting](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

* + 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)  
        
      En kokeillut vielä näitä
  + 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.