OWASP za vse, ne samo za razvijalce

Milan Gabor

Inštitut za varnost podatkov in informacijskih sistemov (IVPIS), Viris d. o. o.,
Maribor, Slovenija
milan@viris.si

OWASP (Open ~~Web~~ Worldwide Application Security Project) je mednarodna neprofitna organizacija, ki se ukvarja z izboljšanjem varnosti programske opreme. Projekt je zasnovan kot odprtokodna skupnost, ki omogoča podjetjem, razvijalcem ter posameznikom dostop do varnostnih virov in orodij. Najbolj znan projekt OWASP je seznam »TOP 10«, ki identificira najpogostejše spletne varnostne grožnje. OWASP nudi tudi številne druge vire, kot so vodniki za razvoj varnih kod, orodja za testiranje penetracije, in izobraževalne vire za izboljšanje varnostne zavesti. Prednosti uporabe OWASP so številne. Prvič – OWASP spodbuja razvoj bolj varne programske opreme z izobraževanjem razvijalcev o najboljših praksah in pogostih napakah. Drugič – OWASP orodja omogočajo podjetjem, da preizkusijo svojo programsko opremo za morebitne varnostne luknje, kar zmanjšuje tveganje za napade. Tretjič – ker je OWASP odprtokoden, lahko organizacije prilagodijo OWASP vire svojim specifičnim potrebam. Skozi članek bomo izpostavili nove ranljivosti na seznamu OWASP TOP 10 in pregledali ostale projekte, ki so lahko koristni tako razvijalcem kot tudi drugim deležnikom v ciklu razvoja programske opreme. Prikazali bomo tudi našo statistiko odkritih pomanjkljivosti in dali poudarek na odpravi teh napak. Na koncu bomo podali praktične nasvete za še bolj varen razvoj programske opreme.

Ključne besede:
OWASP
varnost
spletne aplikacije
kibernetska varnost
testiranje

Univerzitetna založba Univerze v Mariboru
ISBN 978-961-286-772-0
DOI https://doi.org/10.18690/um.feri.8.2023.12

1 Uvod

V digitalni dobi, ko so spletna mesta in aplikacije temeljni del vsakodnevnega življenja, postaja vprašanje varnosti spletnih aplikacij vse bolj pereče. V tem kontekstu je OWASP ali Open Worldwide Application Security Projekt pomemben gradnik znanja in izkušenj za vse, ki se ukvarjajo z razvojem, testiranjem ter vzdrževanjem spletnih in drugih aplikacij. V začetku letošnjega leta je OWASP spremenil Web kratico v Worldwide, saj dandanes pokriva bistveno večje področje kot samo spletne aplikacije.

Zakaj pa je OWASP tako pomemben? V svetu, kjer so kibernetski napadi stalnica in kjer lahko ena sama ranljivost povzroči izgubo milijonov, je imeti zanesljiv, dosleden in ažuren vir informacij o spletni varnosti ključnega pomena. In prav tu OWASP izstopa s svojo širino in precejšnjo ponudbo na različnih področjih.

V nadaljevanju članka bomo podrobneje raziskali specifična področja, ki jih OWASP ponuja, in kako ti dokumenti in druge dobre prakse oblikujejo boljšo ter varnejšo digitalno prihodnost za vse nas.

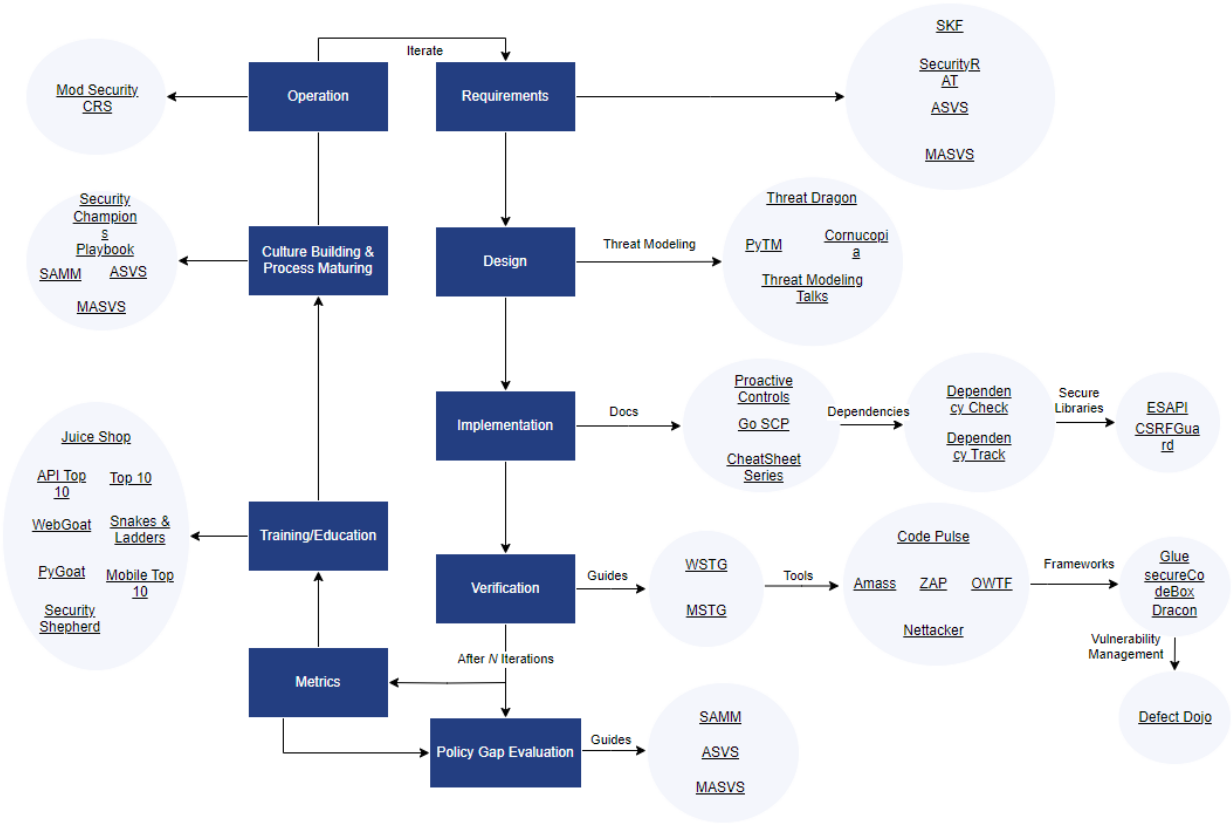
2 OWASP

OWASP je mednarodna neprofitna organizacija, ki se posveča izboljšanju varnosti spletnih in drugih aplikacij. Organizacija deluje na globalni ravni in združuje strokovnjake s področja informacijske varnosti, ki delijo svoje znanje in izkušnje. Glavni cilj OWASP je zagotoviti organizacijam, posameznikom in skupnosti sredstva za izboljšanje varnosti v vseh fazah razvojnega cikla programske opreme.

Nekatere ključne značilnosti OWASP so:

- **Odprtokoden in skupnostno usmerjen projekt**: OWASP vedno spodbuja transparentnost in sodelovanje. Njihovi projekti, orodja in viri so brezplačno dostopni vsem, kar omogoča široko uporabo in priznavanje v industriji.
- **Globalna prisotnost**: OWASP deluje na globalni ravni z lokalnimi poglavji (angl. chapter) v številnih državah, kjer člani redno organizirajo srečanja, delavnice in konference. V Sloveniji imamo tako aktivni dve poglavji, in sicer v Mariboru in Ljubljani.
- **OWASP Top 10:** Ta dokument navaja deset najpogostejših varnostnih tveganj za spletne aplikacije. Namenjen je ozaveščanju razvijalcev in varnostnih strokovnjakov o teh tveganjih. Vsakih nekaj let se ta seznam prilagodi trenutnim tveganjem in je tako vedno v koraku s trenutno situacijo v digitalnem ekosistemu.
- **Projekti in orodja:** OWASP financira in podpira številne odprtokodne projekte in orodja, ki pomagajo pri zaznavanju, preprečevanju in odpravljanju varnostnih tveganj. Primeri takšnih orodij vključujejo OWASP ZAP (Zed Attack Proxy) in Mod Security CRS.
- **Dokumentacija in viri:** OWASP ponuja številne vire, vključno z vodiči, priročniki in standardi za varno razvijanje, testiranje in vzdrževanje spletnih aplikacij.
- **Konference in dogodki:** OWASP redno organizira srečanja, usposabljanja in konference po vsem svetu, kjer se strokovnjaki in skupnost srečujejo, razpravljajo in izmenjujejo znanje o najnovejših trendih in najboljših praksah na področju varnosti aplikacij.
- **Izobraževanje:** Preko seminarjev, srečanj, delavnic in konferenc OWASP širi zavedanje o varnostnih vprašanjih in usposablja strokovnjake s področja IT-varnosti.
- **Podpora skupnosti:** Z vzpostavitvijo globalne mreže strokovnjakov OWASP spodbuja sodelovanje in izmenjavo znanja, kar pospešuje inovacije ter napredek na področju varnosti spletnih aplikacij.

Spekter projektov OWASP je širok in pokriva celotno področje SDLC (Software Development Lifecycle), kar pomeni, da se lahko začnejo stvari iz nabora OWASP uporabljati že pri sami specifikaciji in dizajnu ter skozi celotne faze cikla.



Slika 1: OWASP projekti
Vir: [1].

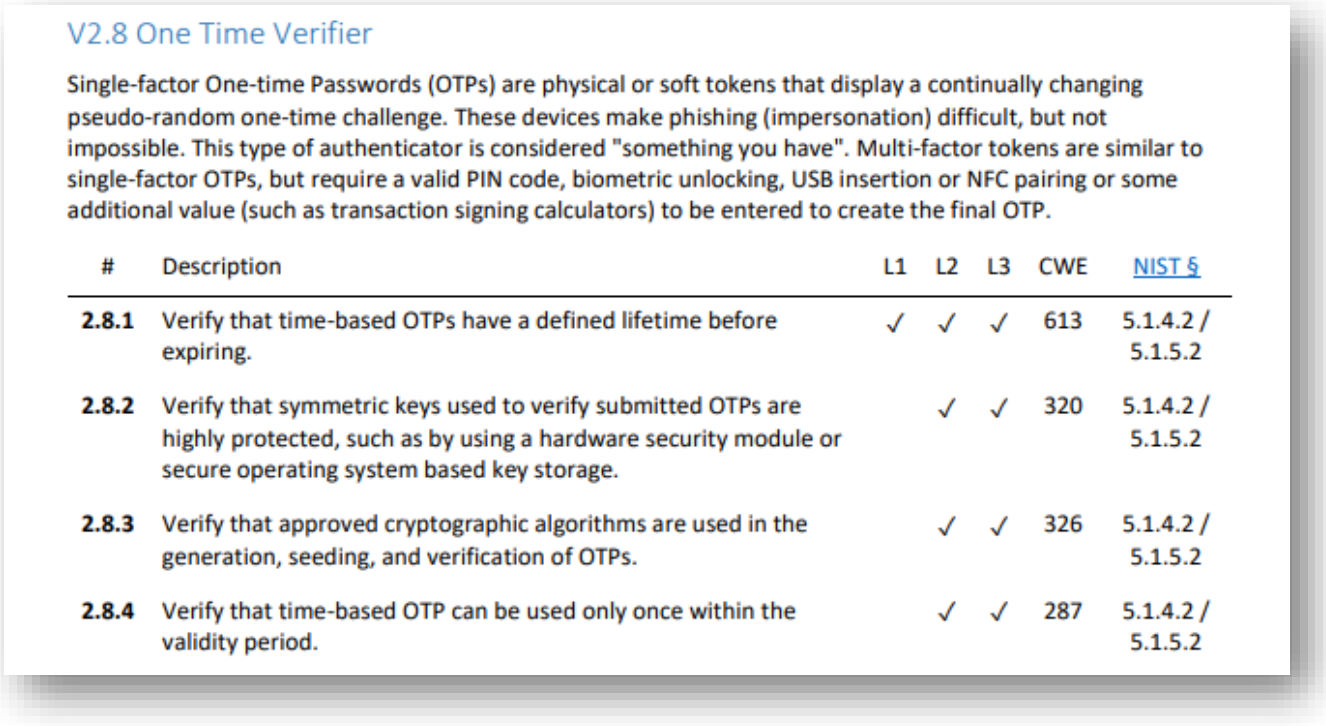
3 OWASP ASVS

OWASP ASVS (Application Security Verification Standard) je okvir, ki ga je razvila organizacija OWASP, da bi zagotovila podrobno osnovo za preverjanje varnosti spletnih aplikacij. Namenjen je pomoči organizacijam, razvijalcem in varnostnim strokovnjakom pri ugotavljanju in vzpostavljanju potrebnih varnostnih kontrol za spletne ter druge aplikacije. ASVS postavlja merila za različne ravni varnosti, ki so potrebne glede na občutljivost in kritičnost aplikacije ali podatkov, ki jih aplikacije obdelujejo.

Ključne značilnosti in vidiki OWASP ASVS vključujejo:

- **Podrobne zahteve:** ASVS vsebuje podrobne zahteve za varno kodiranje in oblikovanje za različne komponente in funkcije spletnih aplikacij, vključno z avtentikacijo, avtorizacijo, sejami, upravljanjem podatkov, konfiguracijo in drugimi področji na aplikativni ravni.
- **Tri ravni preverjanja:** ASVS določa tri različne ravni preverjanja (1, 2, 3), pri čemer je raven 1 osnovna, raven 2 standardna in raven 3 napredna. Organizacije lahko izberejo raven glede na kritičnost in občutljivost aplikacije.
- **Prilagodljivost:** Organizacije lahko uporabljajo ASVS kot vodnik za notranje preverjanje ali ga prilagodijo svojim specifičnim potrebam. Lahko pa ga tudi uporabljajo pri naročanju aplikacij pri zunanjih dobaviteljih.

- **Spodbujanje najboljših praks:** Poleg tega, da služi kot okvir za preverjanje, ASVS služi tudi kot orodje za izobraževanje in ozaveščanje o najboljših praksah na področju varnosti aplikacij.
- **Odprtokodni in skupnostno podprt:** Kot večina OWASP projektov je tudi ASVS odprtokodni, kar pomeni, da je na voljo za brezplačno uporabo, distribucijo in prilagajanje. Projekt je podprt s strani skupnosti, ki redno prispeva k izboljšavam in posodobitvam. V kolikor imate kakšen predlog za izboljšavo, je vedno dobrodošel.



V2.8 One Time Verifier

Single-factor One-time Passwords (OTPs) are physical or soft tokens that display a continually changing pseudo-random one-time challenge. These devices make phishing (impersonation) difficult, but not impossible. This type of authenticator is considered "something you have". Multi-factor tokens are similar to single-factor OTPs, but require a valid PIN code, biometric unlocking, USB insertion or NFC pairing or some additional value (such as transaction signing calculators) to be entered to create the final OTP.

#	Description	L1	L2	L3	CWE	NIST §
2.8.1	Verify that time-based OTPs have a defined lifetime before expiring.	✓	✓	✓	613	5.1.4.2 / 5.1.5.2
2.8.2	Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage.		✓	✓	320	5.1.4.2 / 5.1.5.2
2.8.3	Verify that approved cryptographic algorithms are used in the generation, seeding, and verification of OTPs.		✓	✓	326	5.1.4.2 / 5.1.5.2
2.8.4	Verify that time-based OTP can be used only once within the validity period.		✓	✓	287	5.1.4.2 / 5.1.5.2

Slika 2: Primer za uporabo OTP-žetonov
Vir: [2].

Če povzamemo, OWASP ASVS ponuja strukturiran pristop k verifikaciji varnosti spletnih aplikacij, ki omogoča organizacijam, da razvijajo in vzdržujejo aplikacije, ki ustrezajo določenim varnostnim standardom. Za tiste, ki ne vedo, kako se lotiti postavljanja zahtev, je to zelo dobra odskočna deska za pripravo zahtev, ki jim aplikacije morajo zadoščati.

4 OWASP TOP 10

OWASP TOP 10 je eden najbolj priznanih in uporabljenih standardov v industriji kibernetske varnosti, ki izpostavlja najbolj kritične spletne varnostne ranljivosti. Namen tega projekta je informirati organizacije o tveganjih in motivirati razvijalce, da vključijo varnostne prakse v proces razvoja aplikacij. Predvsem pa je pomemben dejavnik, kako razvijalce prepričati, da se začnejo teh nevarnosti zavedati.

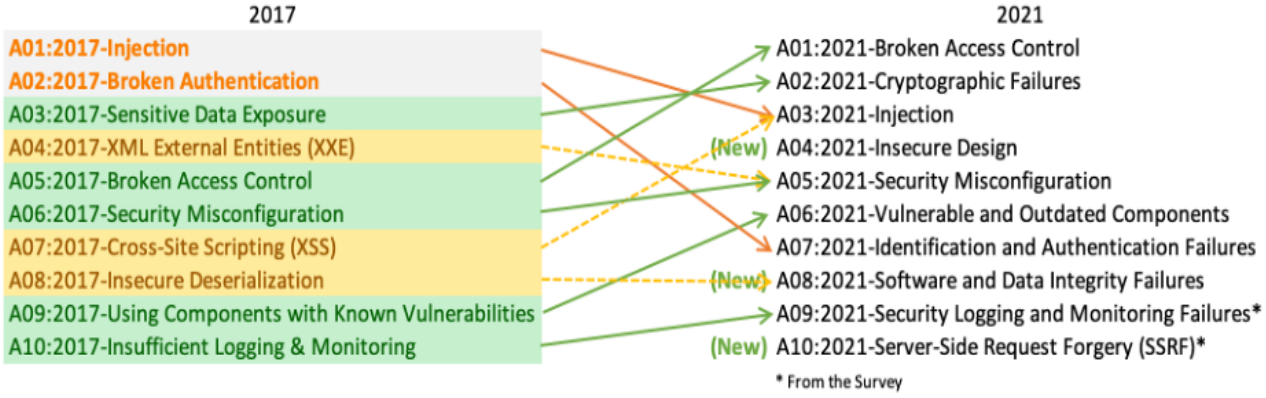
Zelo zanimiv je način, ki ga OWASP uporablja za pripravo seznama TOP 10. V grobem bi lahko ta postopek skrčili na naslednje aktivnosti:

- **Zbiranje podatkov:** OWASP sodeluje s številnimi organizacijami, strokovnjaki za varnost in raziskovalci, da zbira podatke o varnostnih incidentih in ranljivostih v spletnih aplikacijah. Večja podjetja prispevajo sezname in opise ranljivosti, na katere so naletele pri izvedbi varnostnih testiranj.

- **Analiza podatkov:** Zbrani podatki se analizirajo, da se ugotovijo najbolj razširjene in kritične ranljivosti.
- **Razvrstitev tveganj:** Ranljivosti se razvrstijo glede na njihovo kritičnost, razširjenost in potencialno škodo. Pri tem se upoštevajo različni dejavniki, kot na primer, kako lahko organizacije obravnavajo in odpravijo te ranljivosti.
- **Povratne informacije skupnosti:** Preden je nova verzija OWASP TOP 10 dokončno objavljena, se osnutek običajno deli s širšo skupnostjo. Tako OWASP zbira povratne informacije od strokovnjakov za varnost, razvijalcev in drugih zainteresiranih strani, ki jim pomagajo pri končnem oblikovanju seznama.
- **Končna izdaja:** Po zbiranju in analizi povratnih informacij skupnosti OWASP dokonča in objavi novo različico TOP 10. Dokument vključuje podrobne informacije o vsaki ranljivosti, primere, kako se te ranljivosti pojavljajo, in priporočila za njihovo odpravo.
- **Promocija in izobraževanje:** Ko je nova različica objavljena, OWASP izvaja številne izobraževalne in promocijske aktivnosti, da bi spodbudil njegovo uporabo in prepoznavnost v industriji. To vključuje delavnice, predavanja, spletne seminarje in druge izobraževalne vire.

Pomembno je razumeti, da se OWASP TOP 10 sčasoma spreminja, saj se pojavljajo nove ranljivosti in tveganja. Zato je za vse organizacije in posameznike ključnega pomena, da redno spremljajo posodobitve tega dokumenta in prilagajajo svoje varnostne prakse.

Kot lahko vidimo na spodnji sliki, se je seznam iz leta 2017 do 2021 nekoliko spremenil. Morda najbolj opazna in odmevna sprememba je ta, da je po letih kraljevanja vrivanje (angl. Injection) padlo s prvega mesta na tretje mesto. Nekatere ranljivosti so se združile, na seznamu pa so se pojavile nove ranljivosti, ki so bile predlagane in izglasovane s strani skupnosti.



Slika 3: Prikaz starejšega in najnovejšega seznama OWASP TOP 10
Vir: [3].

V nadaljevanju je na kratko opisana vsaka pomanjkljivost, ki je bila dodana na OWASP TOP 10 seznam za leto 2021:

- **A01:2021 – Broken Access Control** – Pomanjkljivo upravljanje dostopa omogoča napadalcem, da zaobidejo avtorizacijske mehanizme, kar lahko pripelje do neavtoriziranega izvajanja funkcij ali nepooblaščenega dostopa do podatkov.
- **A02:2021 – Cryptographic Failures** – Slabo upravljanje s kriptografskimi ključi ali pomanjkljivosti kriptografskih praks lahko pripelje do kompromitiranja zaupnih informacij, kot so gesla, ključi ali osebni podatki.
- **A03:2021 – Injection** – Varnostne ranljivosti zaradi vrivanja omogočajo napadalcem, da pošljejo zlonamerno kodo, ki se izvaja v okolju aplikacije. Najpogostejši primeri so SQL, OS in LDAP injekcije. Med vrivanje po novem sodijo tudi različni XSS-napadi.

- **A04:2021 – Insecure Design** – Varnostne pomanjkljivosti, ki izhajajo iz slabe arhitekturne zasnove aplikacije ali sistema. V kolikor je bila zasnova slaba, lahko posledično nastanejo visoki stroški s ponovnim načrtovanjem aplikacij ali sistemov in dodatnim preverjanjem zasnove.
- **A05:2021 – Security Misconfiguration** – Neprimerna, pomanjkljiva ali napačna konfiguracija varnostnih parametrov in nastavitev na vseh ravneh aplikacije ali sistema lahko še tako varno aplikacijo naredi precej nevarno.
- **A06:2021 – Vulnerable and Outdated Components** – Uporaba komponent, kot so knjižnice ali odprtokodni deli, ki vsebujejo znane varnostne ranljivosti, lahko povzroči kompromitacijo celotne aplikacije. Ugotovljeno je bilo, da se po vključitvi komponent, te v poznejših fazah ne preverjajo več tako pogosto in razvijalci tudi ne sledijo pomanjkljivostim, odkritih v njih.
- **A07:2021 – Identification and Authentication Failures** – Pomanjkljivosti v mehanizmih identifikacije in avtentikacije omogočajo napadalcem neavtoriziran dostop. Napačna uporaba avtentikacijskih mehanizmov ali nerazumevanja, kako le-ti delujejo, lahko tudi povzročijo takšen tip napak.
- **A08:2021 – Software and Data Integrity Failures** – Nezmožnost preverjanja celovitosti in pristnosti programske opreme ali podatkov lahko privede do izvrševanja nepooblaščene kode ali manipulacije podatkov. Razlog za to pomanjkljivost tiči v tem, da je bilo v zadnjih nekaj letih odkritih kar nekaj primerov, kjer so se napadalci lahko vrinili v dobaviteljsko verigo in tako infiltrirali škodljivo kodo v različne sisteme širom sveta.
- **A09:2021 – Security Logging and Monitoring Failures** – Pomanjkanje ustreznega beleženja in nadzora varnostnih dogodkov ali nezmožnost pravočasne detekcije in odziva na varnostne incidente lahko privede do velikih incidentov. Ta del je sicer povezan s konfiguracijo in postavitvijo sistemov in ni samo odvisen od razvoja programske opreme, ampak se dotika tudi DevOps področja.
- **A10:2021 – Server-Side Request Forgery (SSRF)** – Varnostna ranljivost omogoča napadalcem, da sprožijo zahteve iz zanesljivega okolja strežnika proti notranjim virom, do katerih sicer ne bi smeli imeti dostopa. Na seznam je bila uvrščena zaradi tega, ker se dostikrat pozabi na takšne tipe napadov.

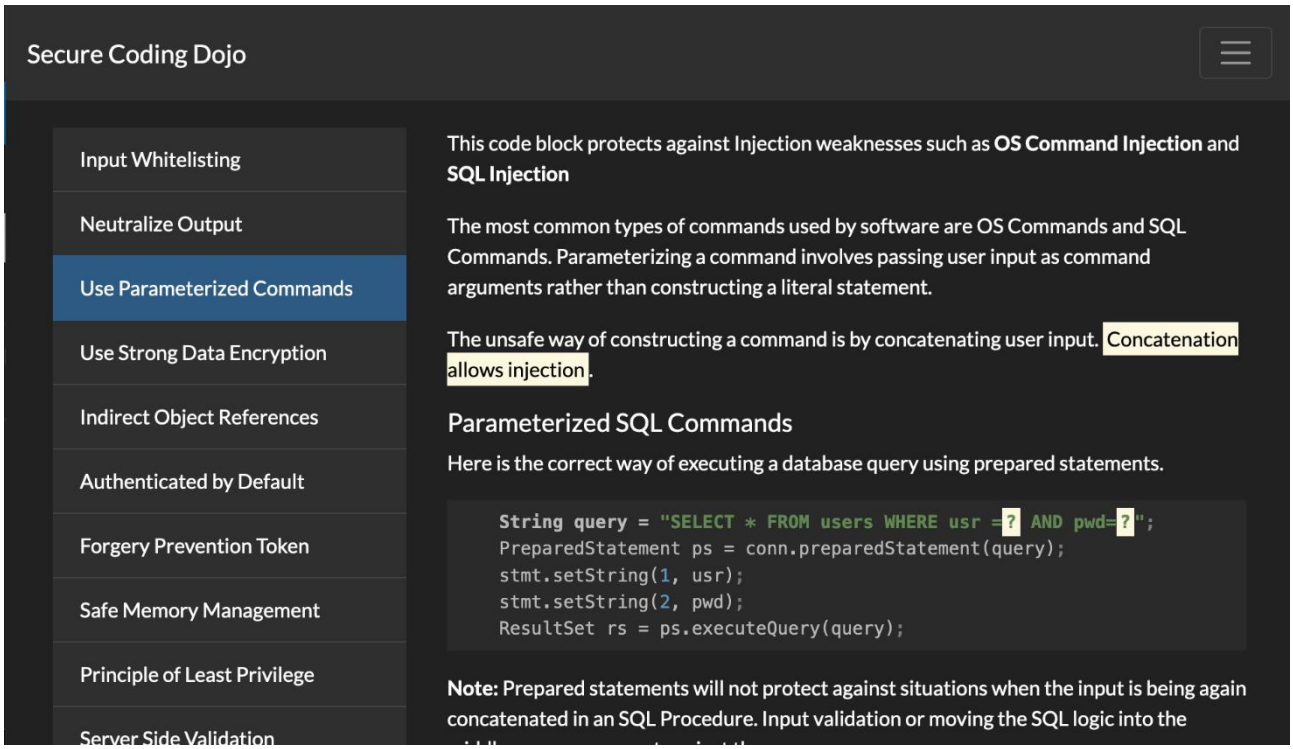
5 OWASP Secure Coding Dojo

Za varen razvoj programske opreme je ključnega pomena tudi varno kodiranje ter razvoj z uporabo dobrih in varnih razvojnih vzorcev. »Naštrikana« koda, ki ne sledi dobrim praksam, je največkrat kriva za odkrite varnostne pomanjkljivosti. Zato je projekt OWASP Secure Coding Dojo dober primer, kako razvijalce učiti in potem naknadno tudi trenirati v varnem razvoju programske opreme.

OWASP Secure Coding Dojo je zasnovan kot niz izzivov, s katerimi lahko uporabniki preizkusijo in izpopolnijo svoje znanje o varnem kodiranju. Nekatere prednosti tega projekta lahko strnemo v naslednje točke:

- **Modularna zasnova:** Dojo vsebuje različne modulirane treninge, ki pokrivajo široko paleto varnostnih tematik, vključno z mnogimi ranljivostmi iz OWASP TOP 10.
- **Praktično učenje:** Namesto zgolj teoretičnih vaj je Dojo zasnovan tako, da omogoča razvijalcem pridobivanje praktičnih izkušenj s testiranjem in odpravljanjem ranljivosti v resničnem okolju.
- **Samostojno učenje:** Razvijalci lahko delajo v lastnem tempu, kar pomeni, da lahko izbirajo med različnimi izzivi in moduli glede na svoje predhodno znanje in zanimanja.
- **Zasnovan za skupine:** Dojo je primeren tudi za skupinsko delo, kar omogoča skupinam, da se skupaj učijo in sodelujejo pri reševanju izzivov.

- **Odprtokodni:** Kot mnogi projekti OWASP je tudi Secure Coding Dojo na voljo kot odprtokodna rešitev, kar pomeni, da ga lahko organizacije prilagodijo in razširijo glede na svoje specifične potrebe.
- **Podpora skupnosti:** OWASP Secure Coding Dojo ima podporo širše skupnosti, ki prispeva k njegovi nadaljnji rasti in izboljšavam.



Slika 4: Primer uporabe Secure Coding Dojo
Vir: [4].

Če povzamemo, je OWASP Secure Coding Dojo odlično sredstvo za vsakogar, ki želi izboljšati svoje veščine varnega kodiranja, ne glede na to, ali je popoln začetnik ali izkušen strokovnjak za varnost. Omogoča praktično izobraževanje in je odlična dopolnitev tradicionalnemu varnostnemu izobraževanju.

6 OWASP Cheat Sheet Series

Ubiranje bližnjic nekako ni dobra praksa v akademski sferi in se lahko velikokrat izkaže za tek na kratke proge, vendar pa lahko dobra in kompaktna vsebina koristi predvsem tistim, ki ne marajo obširne dokumentacije in so bolj praktično naravnani. OWASP Cheat Sheet Series je zbirka dokumentov, ki jih je razvila organizacija OWASP in so zasnovani kot hitri referenčni vodiči za različna področja varnosti spletnih in drugih aplikacij. Vsak "cheat sheet" ponuja konkretne smernice, najboljše prakse in nasvete za določen vidik varnosti ter tako omogoča razvijalcem, arhitektom in varnostnim strokovnjakom, da hitro najdejo informacije o specifični varnostni temi in to koristno uporabijo pri svojem delu.

Naj naštejemo nekaj dobrih strani OWASP Cheat Sheet Series, ki jih lahko strnemo v naslednje alineje:

- **Kratek in jedrnat format:** "Cheat sheets" so namenjeni hitremu iskanju informacij in se osredotočajo na bistvene informacije, potrebne za zagotavljanje varnosti.

- **Pokrivanje širokega spektra tem:** Serija vsebuje vodiče za različne vidike varnosti, kot so avtentikacija, CSRF-zaščita, šifriranje podatkov, varna konfiguracija, preprečevanje napadov SQL-injekcije in mnogo drugih.
- **Praktične smernice:** Vsak "cheat sheet" ponuja praktične smernice in primere, kako pravilno implementirati določene varnostne mehanizme ali kako preprečiti določene ranljivosti.
- **Redno posodobljeno:** Zbirka se redno posodablja, da odraža najnovejše varnostne standarde, ranljivosti in grožnje ter je podkrepljena s kopico strokovnjakov, ki se dnevno srečujejo s takšnimi grožnjami.
- **Podpora skupnosti:** Kot večina OWASP projektov je tudi Cheat Sheet Series odprtokodni projekt, ki ga podpira in redno posodablja široka skupnost varnostnih strokovnjakov.

Slika 5: Primer priporočil za Javo
Vir: [5].

V bistvu je lahko OWASP Cheat Sheet Series zelo dragoceno in priročno sredstvo za vsakogar, ki dela na področju razvoja spletnih in drugih aplikacij in želi hitro dostopati do zanesljivih informacij o specifičnih varnostnih temah. Seveda pa lahko vsak strokovnjak doda ali dopolni področja, če misli, da so pomanjkljivo spisana ali netočna.

7 Rezultati naših izvedenih pregledov aplikacij

Pri pregledu rezultatov naših testiranj spletnih aplikacij v zadnjih dveh letih lahko ugotovimo, da je večina odkritih pomanjkljivosti na seznamu OWASP TOP 10. Ko smo izluščili rezultate in pripravili kratko analizo teh pomanjkljivosti, lahko iz naših odkritih pomanjkljivosti izpostavimo naslednje:

- **Vrivanje kode** – V veliki večini primerov smo odkrili različna vrivanja skripte kode, in sicer Reflected in Stored XXS, v nekaterih redkih primerih tudi primerke DOM vrivanja**.** V enem primeru smo uspeli na platformi pripraviti takšen vnos XSS, ki bi lahko imel posledico avtomatskega širjenja po celotni platformi za vse uporabnike (Wormable Stored XXS). V vseh primerih je težava pri pomanjkanju validacije vhodnih podatkov in nerazumevanja pomembnosti te validacije tako na odjemalčevi kot strežniški strani. SQL-vrivanje smo sicer našli, vendar so bile pri večini pregledov aplikacije starejšega izvora, ki se jih prej nikoli ni pregledalo.
- **Pomanjkljiva kontrola dostopa** – tukaj je bila večina pomanjkljivosti odkrita na način, da se na strežniku ni preverjalo, ali ima prijavljeni uporabnik pravice klica določene funkcionalnosti ali dostopa do podatkov. V nekaj primerih je bilo odkrito, da so se pravice preverjale samo na strani odjemalca, na strani strežnika pa ne in je imel uporabnik poljuben dostop do podatkov na strežniku. V nekaterih primerih so razvijalcu samo iz uporabniškega vmesnika odstranili opcije za izbiro, na strežniku pa potem niso več preverjali, ali ima uporabnik pravice za klic funkcionalnosti.
- **Poljubno preusmerjanje** – veliko število aplikacij omogoča preusmerjanje na druge URL-naslove po prijavi, odjavi ali drugih akcijah. Pri testiranju je bilo velikokrat ugotovljeno, da je možno uporabnika poljubno preusmerjati na potencialno zlonamerne strani, saj ni bilo preverjanja na strani strežnika, katere preusmeritve so dovoljene in katere ne.
- **Neustrezne kontrole za strojno izpolnjevanje obrazcev** – pri skoraj večini aplikacij smo ugotovili, da nimajo zaščite pred avtomatskimi orodji ali boti, ki so lahko brez omejitev klicali končne API-vmesnike ali spletne strani, ter tako ugibali gesla, ali povzročali večjo obremenitev na strežniku.
- **Neustrezna politika gesel** – pri večjem številu analiz smo odkrili, da aplikativni vmesniki sporočajo uporabniku, kakšna je politika gesel. Pri nastavitvi gesla se je potem samo geslo preverjalo na strani uporabnika, na strežniku pa se pri klicu API-vmesnika to ni več preverjalo in smo si lahko nastavili zelo enostavna gesla, kot na primer a ali 1.
- **Neustrezno obravnavanje napak** – ta tip pomanjkljivosti je izstopal pri vseh vrstah aplikacij in uporabljenih tehnologijah. Razvijalci imajo še vedno radi prenos in razkrivanje napak na uporabniški strani, pa naj gre to za klice API-vmesnika ali preko prikaza na spletnih straneh. Te napake razkrivajo velikokrat celotne stack-trace, tehnologije in tudi poti, kjer je koda shranjena na strežniku.

8 Varnostna priporočila za bolj varne aplikacije

Varen razvoj aplikacij je proces, kjer so varnostne prakse integrirane skozi celotno življenjsko dobo razvoja aplikacij in zato so priporočila, kako zagotoviti ta proces, zelo priročna. Takšna priporočila nam lahko pomagajo pri zagotavljanju, da so aplikacije odporne na zlonamerne napade in varujejo občutljive podatke. Če bi na podlagi naših izkušenj podali priporočila, bi jih lahko strnili v naslednje alineje:

- **Zaščita podatkov:** Vsa komunikacija, ki vključuje občutljive informacije (kot so podatki o plačilu ali osebne informacije), naj poteka preko šifrirane povezave (npr. HTTPS).
- **Zanesljivo shranjevanje gesel:** Uporabite močne algoritme za zgoščevanje gesel in dodajte sol. Nikoli ne shranjujte gesel v čisti obliki ali jih shranjujte v vmesnih shrambah.

- **Validacija vhodnih podatkov:** Vedno preverite, filtrirajte in očistite vhodne podatke. Nikoli ne zaupajte podatkom, ki prihajajo od uporabnikov ali drugih aplikacij in API vmesnikov brez predhodne validacije.
- **Validacija na strani strežnika**: Čeprav lahko izvajate validacijo vhodnih podatkov na strani odjemalca, je ključno, da to ponovno izvedete na strani strežnika, saj lahko napadalci obidejo preverjanje na strani odjemalca.
- **Omejitev dostopa:** Zagotovite, da so funkcije in podatki dostopni le tistim uporabnikom, ki imajo ustrezne avtorizacije. To velja tako za uporabniški vmesnik kot za podatkovne vire in API-je.
- **Redne varnostne posodobitve:** Redno posodabljajte vso programsko opremo, vključno z operacijskimi sistemi, spletnimi strežniki, bazami podatkov in vsemi uporabljenimi knjižnicami.
- **Obramba v globini:** Ne zanašajte se samo na eno varnostno rešitev. Uporabljajte večplastno obrambo, kot so požarni zidovi, WAF (Web Application Firewall), IDS/IPS sistemi in druge varnostne rešitve.
- **Zmanjšajte površino napada:** Omejite število dostopnih točk v vaši aplikaciji, onemogočite nepotrebne storitve in funkcionalnosti ter redno preverjajte konfiguracijo sistema.
- **Omejitev pravic in privilegijev:** Uporabite načelo najmanjših privilegijev; to pomeni, da uporabnikom in procesom dodelite le tiste pravice, ki jih resnično potrebujejo.
- **Pravilno upravljanje napak:** Ne razkrivajte preveč informacij v sporočilih o napakah. Tehnične podrobnosti lahko napadalcu ponudijo dragocene informacije, ki jim lahko koristijo potem pri modeliranju napadov.
- **Varnost API-jev:** Če razvijate ali uporabljate API-je, upoštevajte "OWASP API Security Top 10", da se zavedate in obravnavate specifična tveganja, povezana z API-ji.
- **Varnost zasnovana od začetka:** Vključite varnost že na začetnih stopnjah razvojnega cikla, ne pa kot naknadni dodatek, saj boste s tem prihranili čas in denar.
- **Izobraževanje in usposabljanje:** Redno usposabljajte svoje razvijalce in druge člane ekipe o varnostnih najboljših praksah ter aktualnih grožnjah.
- **Odzivanje na incidente:** Pripravite načrt odzivanja na varnostne incidente. Hitro in učinkovito odzivanje na varnostne grožnje lahko zmanjša potencialno škodo in pokaže resnost organizacije, da je pripravljena tudi na takšne izjemne dogodke.

9 Zaključek

V svetu, ki je vse bolj povezan in digitaliziran, OWASP igra ključno vlogo pri zagotavljanju, da spletne in druge aplikacije, ki jih vsakodnevno uporabljamo, ostanejo varne pred grožnjami. Njihovi viri, orodja in skupnostna naravnanost pomenijo, da je OWASP ne le vir informacij, ampak tudi središče globalnega gibanja za varnejši internet.

OWASP je ključni vir za vsakogar, ki je vpleten v razvoj, testiranje, vzdrževanje ali preverjanje varnosti spletnih aplikacij. Zaradi svoje odprtokodne narave in skupnostnega pristopa je postal OWASP eden izmed vodilnih virov na področju varnosti spletnih aplikacij, zato se vsekakor izplača preveriti projekte, dokumente in druge vire, ki jih OWASP ponuja posameznikom in organizacijam.

Zaradi vsega naštetega je vredno preveriti, kako lahko OWASP vključite v svoje projekte in s tem prispevate k bolj varnemu ekosistemu.

Literatura

[1] https://owasp.org/projects/, OWASP projects spletna stran, obiskano 1. 8. 2023.

[2] https://github.com/OWASP/ASVS/raw/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf, Dokument s spletne strani, obiskano 1. 8. 2023.

[3] https://owasp.org/Top10/, OWASP spletna stran, obiskano 1. 8. 2023.

[4] https://owasp.org/www-project-top-ten/, OWASP spletna stran, obiskano 1. 8. 2023.

[5] https://cheatsheetseries.owasp.org/cheatsheets/Java_Security_Cheat_Sheet.html, Spletna stran, obiskano 1. 8. 2023