SIGURNO KODIRANJE

OZNAKA DOKUMENTA	*MT14PRO02*	**DATUM IZDANJA**	*12-12-2023*
PRIMERAK BROJ	*01*	**IZDANJE**	*02*

AUTORIZACIJA	**IME I PREZIME**	**FUNKCIJA**	**POTPIS**
PRIPREMIO	Vladimir Miladinović	Menadžer ISMS	
ODOBRIO	Boris Čorni	Rukovodilac IT sektora	

SADRŽAJ

1 ZAPIS O DOPUNI

Datum	Brojevi strane(a)	Detalji izmene	Broj zahteva za izmenu dokumenta

2 DISTRIBUCIJA I KONTROLA

2.1 DISTRIBUCIJA

Rb. broj	Funkcija/odeljenje
1	Sistem administrator
2	Menadžer za ISMS
3	IT služba

2.2 KONTROLA

Nekontrolisani primerci ovog dokumenta se mogu izdati zainteresovanim stranama ili kupcima organizacije shodno odluci Direktora. Održavaće se zapisi o svakom nekontrolisanom primerku. Svaki primerak će se datirati i označiti sa **„NEKONTROLISANI PRIMERAK - UNIŠTITI NAKON UPOTREBE”** na naslovnoj stran.

3 SVRHA

Svrha bezbednog kodiranja je da bi se osiguralo da je softver napisan bezbedno, čime se smanjuje broj potencijalnih propusta u bezbednosti informacija u softveru.

Ova procedura pokriva sledeću kontrolu:

- 8.28 Secure coding

4 PODRUČJE PRIMENE

Ova politika se primenjuje na zaposlene u sektoru razvoja i proizvodnje softverskih rešenja.

5 TERMINI I DEFINICIJE

U cilju boljeg razumevanja ISMS-a organizacije, u nastavku su navedene definicije značajnijih pojmova koji se koriste u ovoj proceduri.

U dokumentaciji ISMS-a organizacije se koriste sledeće skraćenice:

U dokumentaciji ISMS-a organizacije se koriste sledeće skraćenice:

ISMS – (Information security management Sistems), Sistem menadžmenta bezbednošću informacija - ISO/IEC 27001:2022.

6 REFERENTNA DOKUMENTA

ISO 27001:2022 *Sistem menadžmenta bezbednošću informacija - Zahtevi*

7 OPIS RADA

7.1 SIGURNO KODIRANJE

Organizacija je uspostavila procese u organizaciji kako bi osigurala dobro upravljanje za sigurno kodiranje. Procese i upravljanje treba proširiti tako da pokriju softverske komponente i trećih strana, kroz upravljanje dobavljačima i softver otvorenog koda. Vodeći princip je osigurati da se kod koji je relevantan za bezbednost, poziva kada je to potrebno i da je otporan na neovlašćeno korišćenje.

Ako vlasnik aplikacije može pristupiti skriptama direktnim udaljenim pristupom serveru, to u principu može i napadač. Web serveri bi trebali biti konfigurisani da spreče pregled direktorijuma u takvim slučajevima.

Kod aplikacije je najbolje dizajniran pod pretpostavkom da je uvek podložan napadu, greškom ili zlonamernom radnjom. Osim toga, kritične aplikacije mogu biti dizajnirane da budu tolerantne na unutrašnje greške. Na primer, izlaz iz složenog algoritma može se proveriti kako bi se osiguralo da leži unutar sigurnih granica pre nego što se podaci koriste u aplikaciji kao što je sigurnosna ili financijski kritična aplikacija. Kod koji vrši provere granica je jednostavan i stoga je mnogo lakše dokazati ispravnost.

Neke web aplikacije su podložne raznim ranjivostima koje su uvedene lošim dizajnom i kodiranjem, kao što su ubacivanje baze podataka i napadi skriptiranja na više lokacija. U ovim napadima, zahtevima se može manipulisati kako bi se zloupotrebila funkcionalnost web servera.

7.1.1 Planiranje i aktivnosti pre kodiranja

Principi sigurnog kodiranja će se koristiti i za razvoj novih rešenja i za scenarije ponovne upotrebe. Principi će se primenjivati na razvojne aktivnosti kako unutar organizacije, tako i na proizvode i usluge koje organizacija isporučuje drugima.

Planiranje i preduslovi bezbednog kodiranja moraju uključiti:

a) očekivanja specifična za organizaciju i odobrena načela za bezbedno kodiranje koja će se koristiti i za razvoj koda unutar i za eksterne izvore;

b) uobičajene prakse i ranije prakse kodiranja i nedostatke koji dovode do ranjivosti bezbednosti informacija;

c) konfigurisanje razvojnih alata, kao što su integrisana razvojna okruženja (IDE), kako bi se pomoglo u stvaranju bezbednog koda;

d) smernice koje su izdali vendori razvojnih alata i izvršnih okruženja prema potrebi;

e) održavanje i korišćenje ažuriranih razvojnih alata (npr. kompajlera);

f) kvalifikacije programera za pisanje bezbednog koda;

g) bezbedan dizajn i arhitekturu, uključujući modeliranje pretnji;

h) osigurati standarde kodiranja i, gde je relevantno, obavezati na njihovu upotrebu;

i) korišćenje kontrolisanog okruženja za razvoj.

7.1.2 Tokom kodiranja

Razmatranja tokom kodiranja trebaju uključivati:

• bezbedne prakse kodiranja specifične za korišćene programske jezike i tehnike;

• korišćenje tehnika bezbednog programiranja, kao što su programiranje u paru, refaktorisanje, recenzije kolega, bezbednosne iteracije i razvoj vođen testom;

• korištenje tehnika strukturiranog programiranja;

• dokumentovanje koda i uklanjanje programskih nedostataka, koji mogu omogućiti iskorištavanje ranjivosti bezbednosti informacija;

• zabrana korišćenja nesigurnih tehnika dizajna (npr. korišćenje hard-coded lozinki, neodobrenih uzoraka koda i neovlašćenih web servisa).

Testiranje će se sprovoditi tokom i nakon razvoja.

Pre nego što softver postane operativan, mora se proceniti:

• površina napada i princip najmanje privilegija;

• sprovođenje analize najčešćih programskih grešaka i dokumentovanje da su one ublažene.

7.1.3 Provera i održavanje

Nakon što je kod postao operativan:

• ažuriranja treba da budu bezbedno upakovana i distribuirana;

• treba tretirati prijavljene ranjivosti sistema bezbednosti informacija;

• greške i sumnjive napade treba evidentirati i evidencije redovno preispitivati kako bi se izvršila prilagođavanja koda po potrebi;

• izvorni kod treba da bude zaštićen od neovlašćenog pristupa i izmena.

Ako se koriste eksterni alati i biblioteke, organizacija mora razmotriti:

a) osiguranje da se eksternim bibliotekama upravlja (npr. održavanjem inventara korišćenih biblioteka i njihovih verzija) i da se redovno ažuriraju prema ciklusima izdanja;

b) odabir, autorizaciju i ponovnu upotrebu dobro proverenih komponenti, posebno komponenti za autentifikaciju i kriptografiju;

c) licencu, sigurnost i istoriju eksternih komponenti;

d) osiguravanje da se softver može održavati, pratiti i da potiče iz dokazanih, renomiranih izvora;

e) dovoljno dugoročna dostupnost razvojnih resursa i artefakata.

Ukoliko je potrebno modifikovati softverski paket, uzeće se u obzir:

• rizik od kompromitovanja ugrađenih kontrola i procesa integriteta;

• dobiti saglasnost proizvođača ili prodavca;

• mogućnost dobijanja potrebnih izmena od dobavljača kao standardnih ažuriranja programa;

• uticaj ako organizacija postane odgovorna za buduće održavanje softvera kao rezultat promena;

• kompatibilnost sa drugim softverom koji se koristi.

8 ODGOVORNOSTI I OVLAŠĆENJA

Razvojni sektor je odgovoran za upravljanje bezbednim kodiranjem. Programeri su dužni da se pridržavaju smernica iz ove politike. Razvojni sector je dužan da pre nabavke alata za razvoj preispita sve potencijalne pretnje, ranjivosti i rizike koji mogu da se ostvare eksploatacijom identifikovane ranjivosti od strane pretnje.

Svaki pokušaj osoblja da zaobiđe ili na drugi način povredi ovu politiku ili bilo koju prateću politiku tretiraće se kao kršenje bezbednosti i podleže istrazi. Rezultati istrage mogu povlačiti pismenu opomenu, suspenziju, prekid, a moguće i krivične i/ili građanske kazne.

9 ZAPISI

Sve informacije vezane za dokumentovani postupak upravljanje uvođenjem, razvojem i održavanjem informacionih sistema formiraju deo dokumentovanih informacija organizacije Meridian Tech d.o.o. Beograd. Dokumentovane informacije se moraju zadržavati i održavati u skladu sa ovim postupkom.

U realizaciji ovog postupka nastaju zapisi, odnosno dokumentovane informacije navedeni u tabeli.

Naziv	Referentna oznaka obrasca	Vreme čuvanja	Mesto čuvanja	Odgovorno lice

10 PRILOZI

Nema.