SOSA - Studentske prezentacije - Q&A - 2024./2025.

“Ispiti će se sastojati samo od 10tak random pitanja” navedenih u nastavku.

MI (1. ciklus)

Tema: TPM

Datum: 28/03/2025

Q: Nabrojite najmanje tri namjene za koje možemo koristiti TPM.
A: TPM se koristi za sigurnu pohranu kripotgrafskih ključeva, potvrdu integriteta sustava i autentifikaciju uređaja prilikom pristupanja mreži ili servisu.

Q: Kako TPM osigurava integritet podataka tijekom pokretanja sustava?
A: TPM koristi Platform Configuration Registers (PCR) u koje sprema hash vrijednosti ključnih dijelova sustava. Ako je neka komponenta promijenjena, hash će biti drugačiji i TPM može spriječiti pristup zaštićenim podacima - kriptografskim ključevima, što će onemogućiti dekriptiranje diska i pokretanje sustava.

Q: Koje su glavne značajke TPM-a kojima se štite kriptografski ključevi?
A: Glavne značajke su:

- Generira kriptografski sigurne ključeve.
- Ključevi nikad ne napuštaju sigurno okruženje TPM-a u dešifriranom stanju.
- Kriptografske operacije koje koriste zaštićene ključeve moguće je obavljati isključivo unutar TPM-a, a on je otporan na programske napade.
- Ključevi su zaštićeni naprednim autorizacijskim sustavom koji se nalazi unutar TPM-a - također otporan na programske napade.

Q: Kako TPM omogućuje sigurnu autentifikaciju uređaja i korisnika, te gdje je to korisno?
A: TPM ima mehanizam atestacijskih ključeva kojima dokazuje svoj identitet i identitet korisnika korištenjem kriptografskih potpisa. To je korisno kada želimo da nam udaljeno računalo javi svoje PCR vrijednosti i moramo biti sigurni da te vrijednosti nisu kompromitirane i da su pročitane iz TPM-a - uz njihovo očitanje TPM daje kriptografski potpis. Atestacijski ključevi mogu autentificirati korisnika ako za korištenje atestacijskih ključeva postavimo autorizacijski sustav koji zahtjeva na primjer otisak prsta.

Q: Opišite scenarij u kojem bi TPM mogao spriječiti neovlašteni pristup osjetljivim podacima na računalu.
A: Imamo kriptirane podatke na računalu i netko dobije pristup računalu. Kako ključ za dekripciju nije pohranjen na disku u čitljivom obliku da ga možemo samo iskoristiti i ne generira se iz lozinke. Kriptografski ključ može samo TPM koristiti, pa ga moramo uvjeriti TPM da dekriptira podatke npr. unošenjem lozinke koja može imati postavljenu zaštitu tako da možemo najviše 3 puta u danu unijeti krivu lozinku - hardverska zaštita od napada riječnikom.

Tema: SGX

Datum: 28/03/2025

Q: Koji problem SGX pokušava riješiti (pojasnite)?
A: Izvršavanje osjetljivih aplikacija na nepouzdanim udaljenim računalima, pogotovo kada se koristi oblak (engl. cloud). Ne postoji garancija kako vlasnik infrastrukture nije zlonamjern, a ni da sam stroj na kojem se pokreće naša aplikacija nije kompromitiran. SGX osigurava okruženje za povjerljivo izvršavanje koje pokušava riješiti taj problem.

Q: Ukratko objasnite što je SGX.
A: Sigurnosna tehnologija dostupna na Intelovim procesorima koja osigurava postojanje okruženja za povjerljivo izvršavanje za zaštitu osjetljivih podataka i aplikacija.

Q: Zašto bismo SGX mogli opisati kao reverse sandbox?
A: Zato što štiti našu aplikaciju (tj. njezine osjetljive dijelove) od okoline (BIOS-a, OS-a, hipervizora, firmvera, ...), a ne okolinu od naše aplikacije.

Q: Objasnite što je to enklava te pomoću čega se potvrđuje njena ispravnost (izvršavanje predviđenog koda na predviđeni način); koji se podatak koristi u tom procesu?
A: Enklava je skupni naziv za osjetljivi programski odsječak koji se izvršava te podatke koje obrađuje, a kojima može pristupiti isključivo pouzdan Intelov procesor. Predstavlja zaštićen i siguran kontejner koji izvršava naše aplikacije (tj. njihove osjetljive dijelove). Njena se ispravnost potvrđuje procesom udaljene atestacije, u kojem se koristi mjerni sažetak enklave.

Q: Objasnite što je to udaljena atestacija te zašto je bitna.
A: Udaljena atestacija je mehanizam koji udaljenim korisnicima daje dokaz kako se unutar SGX enklave sigurno izvršava točno očekivani/određeni kod. Udaljena atestacija omogućava nam da vjerujemo samo Intelovom procesoru te da zanemarimo OS, hipervizora ili administratore udaljenog računala.

Tema: Android

Datum: 28/03/2025

Q: Koji su osnovni principi sigurnosnog modela Androida?

A: **M - Model višestrane suglasnosti (multi-party consent - MPC):** uključuje pristanak korisnika (odobravanje dozvola), programera (deklariranje dozvola i poštivanje pravila) i platforme (provedba sigurnosnih mehanizama).

D - Slojevita sigurnost (defense in depth - DD): više slojeva zaštite, uključujući izolaciju aplikacija u jezgri OS-a, upravljanje dozvolama u posredničkom sloju i sigurnosne politike na aplikacijskom sloju.

P - Princip najmanjih privilegija (principle of least privileges - PoLP): aplikacije dobivaju minimalne potrebne dozvole, a preporučuje se traženje samo onih koje su nužne.

Q: Kako višestrana suglasnost poboljšava sigurnost platforme?

A: Višestrana suglasnost podrazumijeva da za određene radnje mora postojati pristanak tri strane: korisnika (koji odobrava dozvole), programera (koji deklarira dozvole i slijedi pravila) i same platforme (koja provodi sigurnosne mehanizme). Time se smanjuje rizik zloupotrebe jer niti jedna strana sama ne može zaobići sigurnosne kontrole.

Q: Što je pohrana s ograničenim pristupom i zašto je uvedena?

A: Pohrana s ograničenim pristupom (scoped storage) uvedena je zbog ranije prakse gdje su aplikacije imale neograničen pristup vanjskom spremniku (external storage), što je predstavljalo sigurnosni rizik (curenje podataka, zlonamjerne aplikacije). Cilj je ograničiti pristup aplikacija samo na njihove vlastite direktorije i omogućiti kontrolirani pristup zajedničkim datotekama putem MediaStore API-ja, čime se povećava privatnost i sigurnost korisničkih podataka.

Q: Kako Android postiže sigurnost međukomponentne komunikacije?

A: Android koristi Intents za komunikaciju između komponenti unutar iste ili različitih aplikacija. Sigurnost se postiže:

- Korištenjem eksplicitnih Intents koji ciljaju određenu komponentu,
- Ograničavanjem tko može slati i primati Intents,
- Provjerom i validacijom Intents izvana,
- Kontrolom pristupa putem dozvola,
- Izolacijom aplikacija (sandboxing) kako bi se spriječio neovlašten pristup.

Q: Koji su glavni sigurnosni izazovi Android platforme?

A: Izazovi su:

1. **Statičke dozvole:** korisnici odobravaju dozvole pri instalaciji, ali često ne razumiju njihove implikacije, a aplikacije mogu tražiti previše dozvola što može dovesti do zloupotrebe.

2. **Nedostatak holističke kontrole protoka informacija:** Android ne pruža potpunu kontrolu nad načinom na koji se podaci dijele između aplikacija, što može uzrokovati curenje osjetljivih podataka.
3. **Kompatibilnost i funkcionalni kompromisi** zbog pohrane s ograničenim pristupom.
4. **Potreba za boljim alatima za praćenje sigurnosti** i otkrivanje zlonamjernog ponašanja aplikacija.

Q: Koje su najčešće prijetnje Android aplikacijama?
A: Najčešće prijetnje su:

1. **Zlonamjerne aplikacije** (virusi, trojanci, špijunski programi, ucjenjivački softver), često distribuirane putem neslužbenih trgovina, phishinga ili sideloadinga.
2. **Phishing i društveni inženjering:** lažne poruke i web stranice koje pokušavaju prevariti korisnike da otkriju osjetljive podatke.
3. **Nedovoljno šifriranje podataka** (npr. pohrana tokena bez korištenja Android Keystore ili EncryptedSharedPreferences).
4. **Loša autorizacija i neadekvatna kontrola pristupa** na serveru i klijentu.

Tema: Sigurnost Dockera

Datum: 04/04/2025

Q: Kako se postiže izolacija procesa između kontejnera?
A: Izolacija procesa između kontejnera postiže se korištenjem razdvojenog imenskog prostora za identifikacijske brojeve procesa jezgre Linux zvanog *PID namespace*. On osigurava da procesi unutar imenskog prostora mogu vidjeti samo druge procese unutar tog prostora, a ne vide vanjske procese. Svaki kontejner ima svoj imenski prostor za identifikacijske brojeve procesa.

Q: Koja je uloga opcije *nodev* prilikom montiranja datotečnog sustava kontejnera?
A: Opcija nodev se koristi prilikom montiranja datotečnih sustava kontejnera i njezina uloga jest da spriječi stvaranje i korištenje datoteka uređaja prilikom pokretanja slike kontejnera. Konkretnije, ukoliko takve datoteke postoje, ova opcija će osigurati da se one promatraju kao obične datoteke.

Q: Kako se ostvaruje mrežna veza između Docker kontejnera i koje je njezino sigurnosno ograničenje?
A: Mrežna veza između Docker kontejnera ostvaruje se virtualnim Ethernet mostom, a njezino sigurnosno ograničenje jest činjenica da ne provodi filtraciju paketa, što ju čini pogodnom za napade čovjeka u sredini i *MAC flooding* napade.

Q: Koje dvije vrste sustava za dodatnu zaštitu jezgre postoje i na koji način oni pružaju zaštitu?
A: Dvije vrste sustava za dodatnu zaštitu jezgre su:

1. **Linux sposobnosti** koje dijele privilegije administratora na sposobnosti koje se mogu omogućiti ili onemogućiti.
2. **Linux sigurnosni model** koji pruža okvir koji omogućuje uključivanje raznih sigurnosnih mehanizama, poput *AppArmoura* i *SELinuxa*, u sustav.

Q: Objasnite način rada sigurnosnog modela AppArmour.
A: Sigurnosni model AppArmour stvara sigurnosne profile za aplikacije koji ograničavaju njezine mogućnosti. Sigurnosni profili mogu biti postavljeni u dva načina: *enforcement*, u kojem je poštivanje pravila obavezno, i *complain*, u kojem se pravila smiju prekršiti, no to će se zabilježiti.

Tema: Modeliranje prijetnji
Datum: 04/04/2025

Q: Što je modeliranje prijetnji?
A: Proces korišten za analaziranje potencijalnih napada i prijetnji koji na strukturiran način osigurava softver.

Q: Koji je glavni razlog modeliranja prijetnji?
A: Određivanje granica sigurnog korištenja sustava. Korisnik mora znati u kojim slučajevima korištenja je aplikacija sigurna, a u kojim nije.

Q: Koji su zadaci eksperta modeliranja prijetnji?
A: Vodi projekt modeliranja prijetnji, upoznaje dionike i ostale na projektu s prijetnjama, postiže dijeljeno razumijevanje oko sigurnosnih rizika.

Q: Koji su koraci tipičnog projekta modeliranja prijetnji?
A: Koraci su:
1. Određivanje ciljeva projekta s dionicima
2. Kreiranje modela sustava
3. Otkrivanje prijetnji i njihovo analiziranje
4. Pregled rangiranih prijetnji te provjera kvalitete rezultata s dionicima.

Q: O čemu govori manifest modeliranja prijetnji?
A: Govori o važnosti modeliranja prijetnji te govori o vrijednostima, pozitivnim i negativnim uzorcima koje je potrebno pratiti tijekom modeliranja prijetnji.

Tema: Modeliranje prijetnji - STRIDE
Datum: 04/04/2025

Q: Nabrojite i ukratko opišite svaku kategoriju modela prijetnji STRIDE.
A: **S – Spoofing**: Lažno predstavljanje čime se dobiva pristup povjerljivim podacima.
T – Tampering: Neovlaštena manipulacija podacima.
R – Repudiation: Poricanje izvršavanja neovlaštene operacija.
I – Information Disclosure: Neovlašteno ili nenamjerno otkrivanje osjetljivih podataka.
D – Denial of Service: Postupak preopterećenja mreže prometom ili zahtjevima čime se smanjuje performansa ili onemogućava pristup.
E – Elevation of Privileges: Korisnik ili proces dobije veća prava pristupa nego što mu je namijenjeno.

Q: Koji su koraci projekta metodom modeliranja prijetnji STRIDE.
A: Koraci su:

1. Dizajn modela sustava
2. Identificiranje potencijalnih prijetnji i ranjivosti
3. Implementacija sigurnosnih zaštita za smanjivanje rizika od prijetnji

Q: Navedite prednosti i mane metode modeliranja prijetnji STRIDE.
A: **Prednosti**: strukturirana analiza prijetnji, može se koristiti u svim fazama razvoja softvera, dijagrami protoka podataka ga čine jednostavim za korištenje i omogućavaju vizualizaciju prijetnji.
Mane: vremenski zahtjevni, potrebna analiza svakog elementa zasebno, ne identificira sve prijetnje.

Q: Kako napadači koriste Spoofing i Denial of Service te koje su metode zaštite?
A: **Spoofing**: Napadač se lažno predstavlja E-mailom, telefonskim pozivom, krade identitet, lozinke, lažira IP adresu.
Zaštita: autentifikacija (npr. MFA), filtriranje e-pošte, tokeni.
Denial of Service (DoS): Napadač zatrpava mrežu zahtjevima, moguće s više zaraženih računala što je poznato kao DDoS napad.
Zaštita: Ograničavanje zahtjeva, CAPTCHA, AWS Shield ili Cloudflare.

Q: Zašto su granice povjerenja važne u dijagramima protoka podataka I kako one pomažu u identificiranju prijetnji?
A: Granice povjerenja predstavljaju mjesta gdje se mijenja razina sigurnosti ili prava, omogućuju identifikaciju ranjivih točki u sustavu gdje su moguće potencijalne sigurnosne ranjivosti.

Tema: Modeliranje prijetnji: Stablo napada

Datum: 11/04/2025

Q: Što je stablo napada i po čemu se razlikuje od grafa napada?
A: Stablo napada je dijagram modeliranja prijetnja koji prikazuje moguće napade i protumjere u strukturi stabla.

Q: Nabrojite prednosti modela stabla napada nad drugim modelima prijetnja.
A: Prednosti su grananje koje može pokriti svaki napad i preduvjet za napad na sustav, jednostavnost i intuitivnost pri čitanju i analizi, te skalabilnost za lagano proširivanje stabla novim napadima.

Q: Objasnite strukturu stabla napada.
A: Korijenski čvor stabla je glavni cilj napada na sustav, listovi su pod-napadi ili koraci za izvršavanje dubljih napada, bridovi pokazuju smjer i zavisnost napada.

Q: Opišite načine definiranja metrika napada u stablu napada.
A: Metriku stabla možemo definirati Booleovim, tj. logičkim vrijednostima te kontinuiranim, tj. numeričkim vrijednostima. Svaki čvor sadrži vlastite vrijednosti za definirane metrike.

Q: Napravite primjer jednostavnog stabla napada koristeći kontinuirane vrijednosti čvorova te označite najefektivniji put po tim svojstvima
A: Primjer stabla napada otvaranja sefa, kontinuirane vrijednosti su cijene ostvarenja napada na čvorovima. Najefektivniji put je onda put napada u kojem je zbroj cijena čvorova minimalan.

Tema: Dizajn arhitekture s naglaskom na sigurnost

Datum: 11/04/2025

Q: Navedite i objasnite tri sigurnosna zahtjeva koja bi trebao implementirati svaki sustav
A: Treba nabrojat tri od ovih sedam:

1. **Autentifikacija (authentication):** Proces provjere identiteta korisnika kojim se omogućuje da samo ovlašteni korisnici mogu pristupiti resursima.
2. **Autorizacija (authorization):** Dolazi nakon autentifikacije i određuje što korisnik smije raditi unutar sustava.
3. **Povjerljivost (confidentiality):** Podaci moraju biti dostupni samo ovlaštenim korisnicima.
4. **Integritet podataka (integrity):** Jamči da podaci ostanu točni, potpuni i nepromijenjeni tijekom prijenosa i pohrane.
5. **Odgovornost (accountability):** Sustav mora omogućiti da se prate aktivnosti korisnika.
6. **Dostupnost (availability):** Sustav mora neprekidno raditi kako bi podaci bili dostupni.
7. **Neporecivost (non-repudiation):** Korisnici ne mogu negirati da su obavili neku radnju

Q: Zašto je bitno da se razmišlja o sigurnosti u samom početku SDLC-a

A: Zato što popravak ranjivosti postaje kompleksniji i skuplji s rastom sustava jer sve više stvari može ovisiti o ranjivoj komponenti, također što duže ranjivost postoji to je veća šansa da će je napadač pronaći i iskoristiti.

Q: Koja je razlika između bug-a i ranjivosti u arhitekturi
A: Greška u kodu (bug) je kada nešto ne radi kako je zamišljeno (zanemarivanje nekog uvjeta) i popravlja se izmjenom koda bez izmjene arhitekture. Ranjivost u arhitekturi je loša odluka u dizajnu sustava (autentifikacija na klijentskoj strani) i popravak zahtijeva redizajn dijela sustava.

Q: Objasnite prednosti i mane korištenja eksternih komponenti
A: **Prednosti:** ne moramo nešto raditi iz nule što omogućuje jeftiniji i brži razvoj sustava, često imaju dobru dokumentaciju i primjere korištenja.
Mane: Povećava se broj mjesta gdje napadači mogu pokušati kompromitirati sustav, moraju postojati odgovorne osobe koje prate promjene

Q: Objasnite Tactic-Oriented Architectural Analysis (ToAA)
A: Analitičar na temelju sigurnosnih taktika postavlja pitanja arhitektu koji je jako dobro upoznat sa sustavom i na taj način u kratkom vremenu se može otkriti koje dijelove sustava treba popraviti. Proces je apstraktan i zbog toga možda nije povezan s kodom.

Tema: Sigurnost mikroservisne arhitekture
Datum: 11/04/2025

Q: Zašto mikroservisi imaju veće sigurnosne izazove od monolita?
A: Zbog distribuirane prirode sustava postoji više vanjskih sučelja pa je površina napada veća nego kod monolitna. Komunikacija i autentifikacija između servisa također povećava kompleksnost i uvodi dodatne izazove.

Q: Objasnite što je API Gateway i navedite njegov značaj u sigurnosti
A: API Gateway je servis s ulogom jedine točke ulaska u sustav. Smanjuje sigurnosne ranjivosti na način da centralizira proces autentifikacije, validira zahtjeve i ograničava njihov broj te skriva unutarnju arhitekturu sustava.

Q: Navedite neke metode autentifikacije u mikroservisnoj arhitekturi
A: OAuth 2.0, OpenID Connect i JSON Web Token

Q: Objasnite princip najmanjih privilegija
A: Korisnici ili procesi trebaju dobiti minimalne privilegije potrebne za obavljanje svojih zadataka. Svaku drugu privilegiju bi mogli zlouporabiti (oni ili napadač kroz njih)

Q: Objasnite kako se sigurnost uklapa u DevSecOps
A: Sigurnosne provjere ubacuju se u svaki korak razvoja programske podrške, sa što je više moguće automatizacije testova.

Tema: Sistemski i operativni zapisi (logiranje)

Datum: 11/04/2025

Q: Kojih je to 6 pitanja na koje mora moći odgovoriti zapis aplikacije koji prati izdvojene generalne sigurnosne smjernice?
A:

- Tko je odgovoran,
- Što se dogodilo,
- Kada se događaj dogodio,
- Gdje se događaj dogodio,
- Zašto se događaj dogodio,
- Kako se događaj dogodio

Q: Navedite dvije moguće razine zapisa te na primjeru objasnite gdje bi se koristile.
A: (Dvije od ovih 4) :

- Katastrofalna (Fatal)
- Greške (Error)
- Informativna (Info)
- Razvojna (Debug)

Q: Koje informacije ne smiju biti zabilježene u zapisu sustava?
A: Privatne korisničke informacije kao OIB, lozinka, adresa, JMBAG, …

Q: Što je to sanitizacija zapisa i koje su dvije moguće arhitekture?
A: Sanitizacija zapisa je proces filtriranja privatnih podataka iz zabilježenih događaja kako bi se prikrili privatni podatci od osobe koja nadgleda zapise. Dvije vrste sanitizacije zapisa su sanitizacija prilikom zabilježavanja zapisa i sanitizacija prilikom pregledavanja zapisa.

Q: Navedi tri biblioteke koji se koriste kod implementacija sustava za zapisivanje događaja.
A: log4j, log4c, syslog, Logguru, Winston, …

Tema: Programski jezik Rust

Datum: 18/04/2025

Q: Objasnite od kojih vrsta grešaka štiti Rust?
A: Rust štiti od grešaka korupcije i nepravilnog korištenja memorije.

Q: Objasnite situacije u kojima je Rust potencijalno dobar odabir jezika?
A: Kada treba brzina sistemskog programskog jezika, a sigurnost je kritična.

Q: Objasnite vlasništvo (eng. *Ownership*) u kontekstu Rusta?
A: Vlasništvo je svojstvo koje varijabla ima nad memorijom i pokušaj korištenja varijable bez vlasništa će rezultirati greškom prilikom prevođenja.

Q: Koje dozvole imaju varijable i reference unutar *borrow checkerovih* pravila u programskom jeziku Rust?
A: Read, write, ownership (čitanje, pisanje, vlasništvo).

Q: Što se događa s varijablom u Rustu kada se u funkciju pošalje izmjenjiva referenca na tu varijablu?
A: Ta varijabla gubi sva prava nad memorijom, a prava joj se vraćaju nakon završetka funkcije.

Tema: Korištenje LLM-ova u programiranju

Datum: 18/04/2025

Q: Zašto je primjena LLM-ova u programiranju sigurnosni izazov?
A: Modeli su trenirani na velikim količinama koda s internet za koje nije nužno da su sigurni. Također, neki dijelovi koda mogu biti sigurni izolirano, ali postati ranjivi kad ih integriramo u veći sustav.

Q: Što je SVEN?
A: SVEN je metoda za kontrolirano generiranje koda uz fokus na sigurnost. Dva načina rada su SVENsec i SVENvul.

Q: Navedite 3 najčešće ranjivosti koje Codex generira.
A: CWE-089 (SQL Injection), CWE-798 (Use of Hard-coded Credentials) i CWE-022 (Path Traversal).

Q: Što pokazuju rezultati istraživanja korištenja LLM-a za generiranje C koda?
A: LLM ne povećava broj ranjivosti u značajnoj mjeri, a neke ranjivosti su čak i rjeđe u grupi koja koristi AI. Zaključeno je da prevagnu prednosti generiranja koda LLM-om.

Q: Koje su razlike u ranjivostima generiranog koda između ChatGPT-a i StackOverflowa?

A: ChatGPT generira manje ranjivosti, ali se ranjivosti dosta razlikuju (malo preklapanje). Ipak najčešće ranjivosti, i to one s MITRE Top 25 liste se preklapaju.
