

Zaštita i sigurnost informacijskih sustava

Uvod

prof. dr. sc. Krešimir Fertalj
izv. prof. dr. sc. Stjepan Groš
prof. dr. sc. Boris Vrdoljak

Motivacija

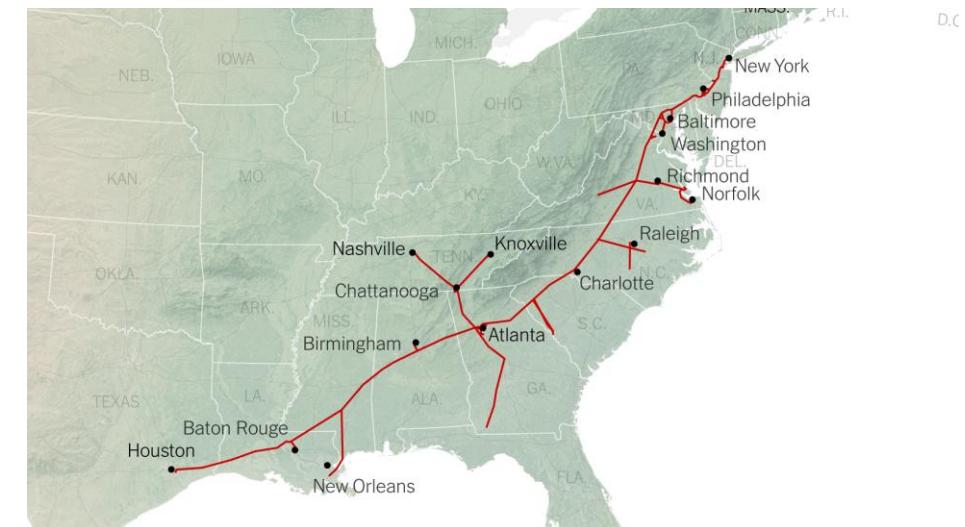
- Kontinuirani niz vijesti o napadima na organizacije
 - Mnogi incidenti se ni ne prijavljuju
- Stav je da nije pitanje **AKO** će se nekakav napad desiti već **KADA** će se desiti

Primjeri nekih značajnijih napada (1)

- SolarWinds
 - Prosinac 2020
 - Napadnuta tvrtka SolarWinds(!?)
 - Ubačen maliciozni kod u programsku podršku *Orion IT* za nadzor i upravljanje mreža
 - Mjeseci prošli prije nego je napad slučajno otkrila tvrtka **FireEye**
 - Primjer napada na dobavni lanac (engl. supply chain attack)
 - Ogroman broj žrtava, Microsoft, agencije američke Vlade, ...

Primjeri nekih značajnijih napada (2)

- *Colonial Pipeline* ucjenjivački napad
 - Svibanj 2021
 - Tvrtka koja obavlja transport nafte i derivata uz jugoistočnu obalu SAD-a
 - Vojne baze ovise o toj tvrtci
 - Tvrtka je dio kritične infrastrukture
 - Ulaz u mrežu tvrtke napravljen pomoću kompromitiranih vjerodajnica
 - Plaćena otkupnina, ali svejedno je oporavak potrajan
 - Dio otkupnine kasnije vraćen



Neki zaključci na temelju primjera

- Sve organizacije danas jako ovise o svojim informacijskim sustavima
 - Ako je narušena sigurnost informacijskog sustava tvrtka može pretrpjeti velike štete ili čak propasti.
 - To vrijedi čak i za industrijske sustave!
- Organizacije su međusobno ovisne
 - Napad na jednu organizaciju može imati katastrofalne posljedice po druge organizacije

I još neke činjenice o organizacijama

- Organizacije imaju kompleksne IT sustave
 - Teško je znati sve što se događa u tim sustavima
- Tehnologija se brzo mijenja i postaje sve kompleksnija
- Ljudi su bitna karika sustava
 - ... a ljudi grijese

O napadačima

- Jako ih je puno različitih sposobnosti, kompetencija i motivacija
- Vrlo brzo se prilagođavaju situaciji
- Teško ih je pronaći i uhvatiti
 - Najčešće nam je s druge strane nepoznata osoba ili grupa
- U pitanju je **inteligentni suparnik**

Problem s kojim se bavimo

- Kako zaštititi organizaciju od napada, u uvjetima
 - ... u kojima ne znamo što nam sve prijeti
 - ... imamo kompleksnu organizaciju gledajući tehnologiju i poslovne procese
 - ... ne poznajemo cijelu organizaciju, a možda nemamo ni kontrolu nad njom
 - ... napadači se stalno mijenjaju i poboljšavaju
 - ... tehnologija i poslovni zahtjevi idu naprijed jako brzo
 - ... imamo ograničene resurse na raspolaganju za obranu
- **OVO NIJE (SAMO) TEHNIČKO PITANJE**

Faktori koji utječu na sigurnost IS-a

- Možemo ih razvrstati u tri grupe
 - Ljudski faktori
 - Organizacijski faktori
 - Tehnološki faktor

Ljudski faktori

- Nedostatak edukacije u području sigurnosti
- Nedostatak komunikacije po pitanju sigurnosti
- Nedostatak svijesti
- Nedostatak kulture
- Neodgovarajuće ponašanje

Organizacijski faktori

- Nedostatak budžeta
- Kratki rokovi
- Nedostatak podrške menadžmenta
- Nedostatak odgovarajuće procjene rizika
- Nepostojanje sigurnosnih procedura

Tehnološki faktori

- Ranjivosti u IT imovini
- Nedovoljni ili neodgovarajući sigurnosni mehanizmi
 - primjerice vatrozidi, IDS-ovi, antivirusna podrška, lozinke, šifriranje, itd.

Teme predavanja

- Upravljanje sigurnošću informacijskog sustava
- Standardi sigurnosti informacijskog sustava
- Životni ciklus sigurnog razvoja sustava
- Analiza rizika i upravljanje rizikom informacijskog sustava
- Oblikovanje prijetnji
- Planiranje kontinuiteta poslovanja za nepredviđene slučajeve
- Sigurnost baza podataka
- Revizije informacijskog sustava
- Sigurnost elektroničkog poslovanja

Način polaganja ispita

- Kontinuirano praćenje nastave
 - Pohađanje nastave – 6 bodova
 - Međuispit – 47 bodova
 - Završni ispit – 47 bodova
 - Nema minimuma na svakoj od navedenih komponenti
- Ispitni rok
 - Pismeni ispit – 100 bodova
- **Napomena:** nema bodovnog praga na međuispitu, završnom ispitu ili pismenom ispitu

Ocjena	Bodova
Izvrstan	87,50
Vrlo dobar	75,00
Dobar	62,50
Dovoljan	50,00

Literatura

- Slajdovi će biti dostupni na stranicama predmeta
- Tijekom semestra, vezano uz specifične teme, dobijat ćete još literature

Neki izvori informacija (1)

- Stručne konferencije
 - BlackHat USA/EU/Asia/Israel [www.blackhat.com]
 - RSA Conference
- Znanstvene konferencije
 - ACM Conference on Computer and Communication Security
 - USENIX Security
 - IEEE Symposium on Security and Privacy

Neki izvori informacija (2)

- Blogovi
 - Krebs On Security, Schneier on Security, A Few Thoughts on Cryptographic Engineering
 - Symantec, Microsoft, FireEye, Kaspersky, ...
- Organizacije, udruge i certifikacije
 - SANS, ISACA, ISC2, MITRE
 - Honeynet organization
- Twitter

Osnovni pojmovi – ponavljanje

Ponavljanje – pojam sigurnosti

- Primjeri nekih izjava koje uključuju pojam „sigurnost”
 - Lozinke su sigurne
 - Web stranice tvrtke su sigurne
 - Poslužitelj je siguran
 - Tvrtka je sigurna
- Pojam „sigurnost” je složen i kontekstno osjetljiv

Ponavljanje – definicija sigurnosti

- Sigurnost (engl. security)
 - **Kontinuirani proces** čijim provođenjem se osigurava određeno stanje (sustava i/ili podataka/informacija). Željeno stanje je definirano određenim **zahtjevima**.
 - Kada su zahtjevi ispunjeni, kažemo da je sustav i/ili informacija sigurna. Ako neki od zahtjeva nije ispunjen, kažemo da se desio **incident**, odnosno, da je **narušena sigurnost**.
- U ovom predmetu zanima nas **sigurnost informacijskog sustava**
 - uključuje mnoga područja sigurnosti

Ponavljanje – osnovni sigurnosni zahtjevi

- Bez obzira o kojoj sigurnosti govorimo i dalje su bitni sigurnosni zahtjevi
- Tri temeljna zahtjeva
 - Tajnost/Povjerljivost (engl. secrecy, confidentiality)
 - Cjelovitost/Integritet (engl. integrity)
 - Raspoloživost (engl. availability)
- Dodatni zahtjevi
 - Autentičnost (engl. authenticity)
 - Neporecivost (engl. non-repudation)

Ponavljanje – Prijetnje

- Da bi se desio incident moraju postojati dva preduvjeta: ranjivost i **prijetnja**
- Prijetnja (engl. threat) je bilo kakav događaj koji može iskoristiti ranjivost te na taj način prouzročiti štetu.
 - Izvori prijetnji su ljudski (napadači) ili prirodni (potres, nestanak struje);
 - Dodatno ljudski izvori mogu biti namjerni (napadači) ili slučajni (nepažnja osobe)

Ponavljanje – Ranjivosti

- Da bi se desio incident moraju postojati dva preduvjeta: **ranjivost** i prijetnja
- Ranjivost (engl. vulnerability) je pogreška ili slabost u dizajnu sustava, implementaciji, upotrebi ili upravljanju koja se može iskoristiti za narušavanje sigurnosti sustava ili informacije.
 - Pogreške u programskoj podršci (engl. bugs), propusti u protokolima, kriva upotreba programske podrške ili nekog sustava

Načini postizanja sigurnosti

- Tako da uklonimo prijetnje i/ili ranjivosti

Kako djelovati na prijetnje?

- Dva su osnovna pristupa
 - Djelovanje na motiv i podizanje cijene djelovanja
- Djelovanje na motiv
 - Ako imamo nešto što napadač želi, riješimo se toga
 - Nije uvijek primjenjivo
- Podizanje cijene
 - Otežavanje djelovanja napadaču – uvođenje zaštitnih mjera
 - Prijetnja visokim kaznama – teško je pronaći napadače

Djelovanje na ranjivosti

- Ranjivosti se mogu ukloniti ili ublažiti **kontrolama**
 - Kontrole su zaštite koje primjenjujemo u sustavu
 - Sve kontrole svrstavamo u tri velike grupe
 - Fizičke kontrole – kamere, zaštitari, ograde, ...
 - Tehničke kontrole – vatrozidi, antivirus, ...
 - Administrativne kontrole – politike, procedure, pravilnici

Pojam informacijskog sustava

- Što je informacijski sustav?
 - Odgovor na to pitanje nije lako dati, mnoštvo je mogućih definicija
- Za naše potrebe bit će sasvim dovoljna sljedeća definicija

Informacijski sustav je bilo koji organizirani sustav za prikupljanje, organizaciju, pohranu i razmjenu informacija

Neke posljedice/zanimljivosti definicije

- Informacijski sustav ne uključuje nužno računala i/ili računalnu mrežu
 - Iako su današnji informacijski sustavi nezamislivi bez njih
- Informacijski sustav uključuje i ljudе i procese
- Informacijski sustav uključuje sve na čemu se nalaze informacije značajne za tvrtku
 - Uz napomenu: koje nisu javno objavljene!

Informacijska (i komunikacijska) tehnologija

- Čest pojam (i skraćenica) u kontekstu informacijskih sustava
 - IT, ICT
- Nema općeprihvaćenih niti univerzalnih definicija navedenih pojmove
 - Svaka definicija ima svojih prednosti i nedostataka (a neke uopće nisu dobre!)
 - Ovisno o potrebi koristi se neka od raspoloživih definicija

Definicija IT/ICT pojma

- Jedna potencijalna definicija sasvim zadovoljavajuća za naše potrebe

Informacijska tehnologija (IT) je primjena računala i telekomunikacijske opreme za pohranu, dohvat, prijenos i obradu podataka, često u poslovnom kontekstu.

- Dakle, IT je sastavni dio (čak i temelj) IS-a
 - Slijedi da je sigurnost IS-a usko vezana uz sigurnost IT-a

Operativna tehnologija

- Engl. Operational Technology (OT)
- Koristi se u upravljačkim sustavima
 - Koristi iste tehnološke temelje kao IT, ali prvenstvena zadaća je upravljanje fizičkim procesima
- Sadrži elemente koji nisu uobičajeni u IT-ju
 - PLC, SCADA, RTU, ...
- Govorimo o industriji, različitim opskrbnim sustavima, ...

IT vs. OT

- U IT-ju temeljni zahtjev je *tajnost*
 - Informacija ne smije biti poznata neovlaštenim osobama
 - Svaka organizacija ima IT
- U OT-u temeljni zahtjev je *raspoloživost*
 - Prenose se mjerena i upravljačke naredbe
 - Samo neke organizacije imaju OT
 - Niz specifičnosti u odnosu na IT
- Iako ćemo govoriti u nastavku samo o IT-ju, ne zaboravite da postoji i OT

Zašto će sigurnost dugo biti problem?

- Sustavi postaju sve kompleksniji
- Paradoks je da je jeftinije izgraditi složen sustav nego jednostavan
 - Složen sustav se potom prilagođava da izvršava jednostavnu funkcionalnost
 - Primjer: CPU-ovi
- Zbog korištenja složenih sustava napadači mogu dobiti i drugačija ponašanja

Hvala!



Zaštita i sigurnost informacijskih sustava

Sigurnost baza podataka

dr. sc. Jasenka Anzil
prof.dr.sc. Mirta Baranović
prof.dr.sc. Boris Vrdoljak

U ovoj prezentaciji koriste se i prilagođeni materijali iz predmeta Baze podataka (FER)

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Zaštićeno licencijom <http://creativecommons.org/licenses/by-nc-sa/2.5/hr/>



Creative Commons



□ slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo

□ pod sljedećim uvjetima:

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencem koja je ista ili slična ovoj.

U slučaju daljnog korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava. Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licencije preuzet je s <http://creativecommons.org/>.

Sadržaj

- ◆ baze podataka i sustavi za upravljanje bazama podataka
- ◆ dokazivanje autentičnosti korisnika baze podataka
- ◆ upravljanje pristupom
 - diskrecijsko upravljanje pristupom
 - kontekstno ovisna zaštita podataka
 - mandatno upravljanje pristupom
 - baze podataka s višerazinskom zaštitom podataka
 - upravljanje pristupom temeljeno na ulogama
- ◆ šifriranje podataka
- ◆ nadgledanje rada korisnika



Baze podataka i sustavi za upravljanje bazama podataka

Baza podataka

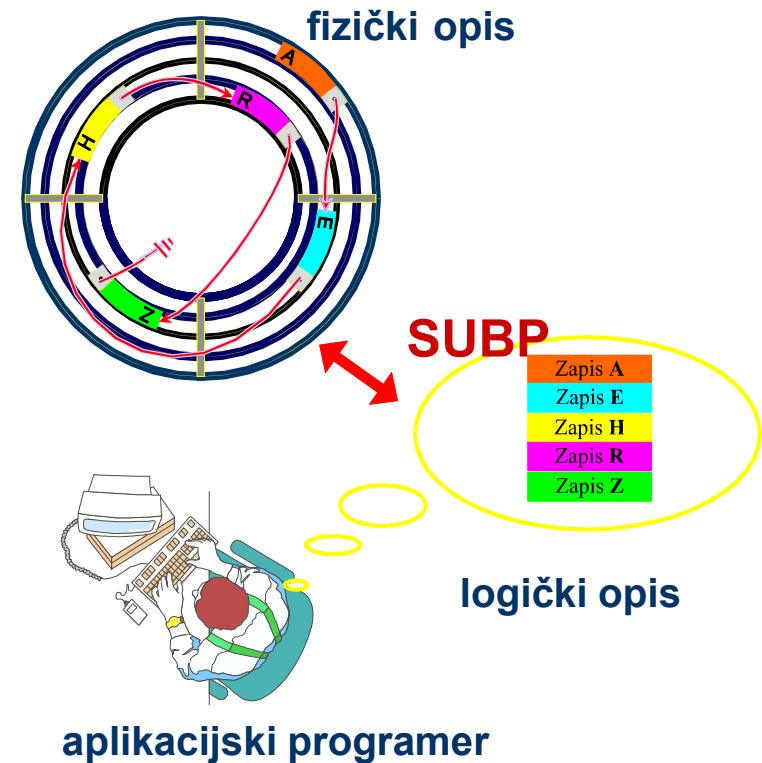
- skup podataka koji su pohranjeni i organizirani tako da mogu zadovoljiti zahtjeve korisnika. (M. Vetter)
- skup međusobno povezanih podataka, pohranjenih zajedno, uz isključenje bespotrebne zalihosti (redundancije), koji mogu zadovoljiti različite primjene. Podaci su pohranjeni na način neovisan o programima koji ih koriste. Prilikom dodavanja novih podataka, mijenjanja i pretraživanja postojećih podataka primjenjuje se zajednički i kontrolirani pristup. Podaci su strukturirani tako da služe kao osnova za razvoj budućih primjena. (J. Martin)

Sustav za upravljanje bazama podataka (SUBP)

- ◆ SUBP je programski sustav koji:
 - omogućava upravljanje podacima u bazi podataka
 - može istovremeno upravljati s više baza podataka
 - upravlja istovremenim pristupom bazi podataka od strane više korisnika/aplikacija uz osiguravanje sigurnosti i integriteta baze podataka
 - temelji se na odabranom modelu podataka (hijerarhijski, mrežni, **relacijski**, objektno-relacijski, objektno-orijentirani, ...)

Zadaće SUBP-a

- ◆ trajna pohrana podataka (*persistent storage*)
 - ◆ skriva od korisnika detalje fizičke pohrane podataka
 - ◆ osiguravanje programskog sučelja (*programming interface*)
 - ◆ omogućuje definiciju i rukovanje s podacima
 - ◆ DDL - Data Definition Language
 - ◆ DML - Data Manipulation Language
 - ◆ optimiranje metoda pristupa podacima (*query optimization*)
 - ◆ **zaštita podataka**
 - ◆ **integritet podataka (*integrity*)**
 - ◆ **pristup podacima - autorizacija, sigurnost (*security*)**
 - ◆ **potpora za upravljanje transakcijama**
 - ◆ **upravljanje istodobnim pristupom (*concurrency control*)**
 - ◆ **obnova u slučaju razrušenja (*recovery*)**



Korisnici

- pristupaju bazi tako da postavljaju upite, mijenjaju podatke i izrađuju izvještaje
- različite skupine korisnika:
 - korisnici koji često koriste bazu postavljajući standardne upite i radeći standardne promjene **koristeći programirana sučelja** (npr. službenici u banci, turističkim agencijama...)
 - korisnici koji povremeno pristupaju bazi **koristeći upitni jezik**
 - **sofisticirani korisnici** koji su dobro upoznati s bazom podataka i koriste je na složeniji način (npr. inženjeri, znanstvenici, poslovni analitičari)
 - **administratori**

Administratori

- ◆ organiziraju, nadziru i optimiziraju korištenje SUBP-a
- ◆ **odgovorni za sigurnost SUBP-a**

- ◆ administrator poslužitelja baze podataka (*database server administrator* – DBSA)
 - instaliranje i nadogradnja SUBP-a; kreiranje novog korisnika; autorizacija na razini SUBP-a

- ◆ administrator baze podataka (*database administrator* - DBA)
 - autorizacija na razini baze podataka (provodenje sigurnosne politike)
 - dodjeljivanje i ukidanje ovlasti korisniku baze podataka
 - dodjeljivanje sigurnosne klasifikacijske razine korisniku u višerazinskom sigurnosnom sustavu, u skladu s politikom organizacije



Sigurnost i zaštita baza podataka

Primjeri narušavanja sigurnosti podataka

- ◆ 2012 - Global Payments (posrednik između trgovina i kartičnih kuća) - pogodjeni MasterCard, Visa, American Express i Discover Financial Services, banke - ukradeni podaci o **1,5 milijuna računa** (neki izvori govore o 10 milijuna **kartica**)
- ◆ 2013-2014 – Yahoo – probijene lozinke – napadači došli do imena, email adresa, lozinki, telefonskih brojeva... **za 3 milijarde korisnika**
- ◆ 2015 - The Internal Revenue Service – ukradeni identiteti **104,000** poreznih obveznika – rezultat hakerskog upada u aplikaciju
- ◆ 2018 – Marriott International – ukradeni osobni podaci oko **500 milijuna korisnika** (gostiju lanca hotela) u razdoblju od 2014. do 2018. godine

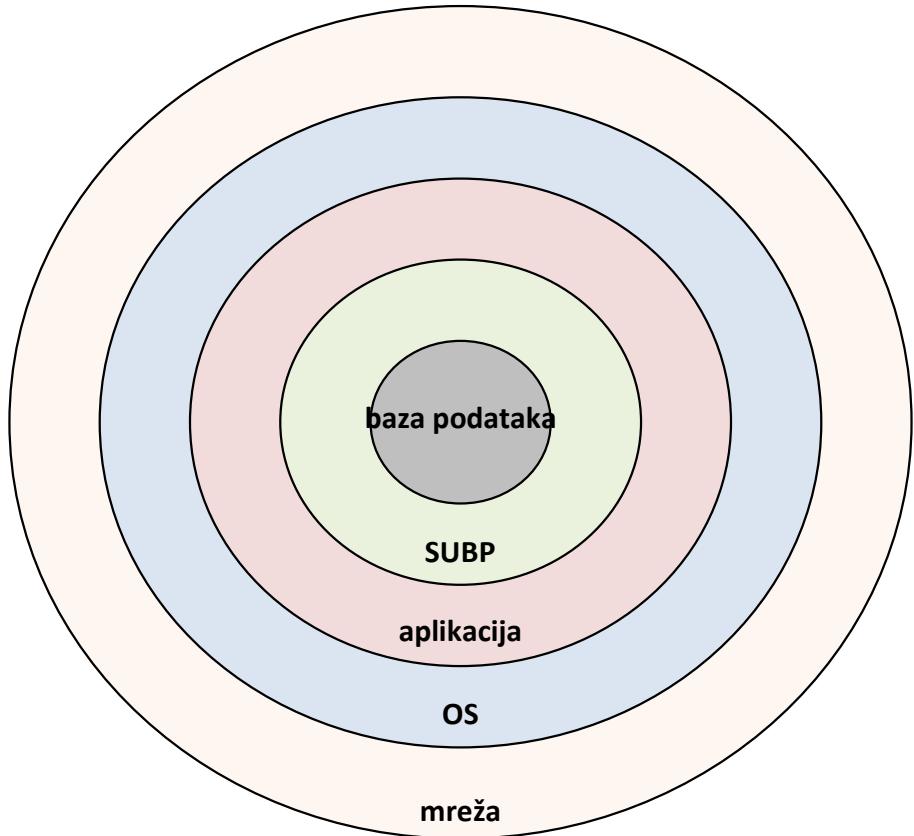
Narušavanje sigurnosti i moguće posljedice

- ◆ Oblici narušavanja sigurnosti baze podataka su:
 - neovlašteno čitanje, izmjena ili uništavanje podataka
- ◆ Moguće posljedice su:
 - krađa ili prijevara
 - gubitak tajnosti
 - odnosi se na podatke kritične za funkcioniranje organizacije
 - npr. krađa recepture - rezultira gubitkom konkurentnosti na tržištu
- ◆ gubitak privatnosti
 - odnosi se na osobne podatke
 - npr. krađa podataka o zdravstvenom stanju osobe – rezultira sudskim procesom protiv vlasnika baze podataka
- ◆ gubitak raspoloživosti
 - npr. uništenjem dijela podataka

Sigurnost baze podataka (1)

- ◆ sigurnost baze podataka se osigurava zaštitom na nekoliko razina
 - **na razini SUBP**
 - spriječiti pristup bazama podataka ili onim dijelovima baza podataka za koje korisnici nisu ovlašteni
 - **na razini operacijskog sustava**
 - spriječiti pristup radnoj memoriji računala ili datotekama u kojima SUBP pohranjuje podatke
 - **na razini računalne mreže**
 - spriječiti presretanje poruka
 - **fizička zaštita**
 - zaštita lokacije poslužitelja sustava za upravljanje bazama podataka
 - **na razini korisnika**
 - spriječiti da ovlašteni korisnici bilo nepažnjom bilo namjerno omoguće neovlaštenim osobama pristup podacima

Sigurnost baze podataka (2)



Defense-in-depth strategija

- ◆ **zaštita u više slojeva**
 - ne postoji savršeni zaštitni sloj, metoda ili proizvod
 - proboj jednog sloja ne mora značiti i narušavanje sigurnosti podataka iz baze podataka
 - nužna zaštita unutar baze podataka, čak i ako je implementiran poseban sustav zaštite baze podataka izvan baze podataka

Svaki pokušaj neovlaštenog pristupa sustavu mora biti automatski zabilježen korisničkim imenom, datumom i vremenom, a ako je to moguće i mjestom s kojeg je takav pristup pokušan.

Sigurnost baze podataka (3)

- ◆ neki od postupaka kojima je bazu podataka moguće učiniti sigurnijom:
 - **ograničiti pristup važnim resursima** koji mogu biti pogrešno korišteni, zlonamjerno ili slučajno kao posljedica pogreške - vatrozidi; upravljanje pristupom; antivirusna zaštita...
 - upravljanje zakrpama (*patches*)
 - sustavi za otkrivanje i sprječavanje upada (*Intrusion prevention systems* - IPS , *Intrusion detection systems* - IDS)
 - onemogućiti nepotrebne komponente i servise sustava za upravljanje bazama podataka
 - ukloniti/onemogućiti nepotrebne korisničke račune i lozinke
 - izvoditi procese baze podataka pod namjenskim, neprivilegiranim korisničkim računom
 - ...

Aspekti zaštite podataka

- ◆ zakonski, socijalni i etički aspekt
 - ima li vlasnik baze podataka zakonsko pravo na prikupljanje i korištenje podataka
 - npr. smije li zdravstvena ustanova koja, u skladu sa zakonom prikuplja podatke o pacijentima, te iste podatke koristiti pri donošenju odluke hoće li svog bivšeg pacijenta zaposliti
- ◆ strategijski aspekt
 - tko definira pravila pristupa - tko određuje kakve ovlasti ima pojedini korisnik baze podataka, ...
- ◆ operativni aspekt
 - kako osigurati poštivanje pravila - kojim mehanizmima se osigurava poštivanje definiranih pravila, na koji način su lozinke zaštićene, koliko često se mijenjaju, ...

Pravni okvir

- ◆ Ustav RH - Članak 37.

Svakom se jamči sigurnost i tajnost osobnih podataka. Bez privole ispitanika, osobni se podaci mogu prikupljati, obrađivati i koristiti samo uz uvjete određene zakonom.

Zakonom se uređuje zaštita podataka te nadzor nad djelovanjem informatičkih sustava u Republici.

Zabranjena je uporaba osobnih podataka suprotna utvrđenoj svrsi njihovoga prikupljanja.

- ◆ Zakon o zaštiti osobnih podataka

- ◆ GDPR - General Data Protection Regulation

Opća uredba o zaštiti osobnih podataka koja se primjenjuje od 25. svibnja 2018. godine.

Načelo najmanje ovlasti i razdvajanje dužnosti

- ◆ **načelo najmanje ovlasti (least privilege)**
 - korisnik ima **minimalan skup dozvola** koje su neophodne za njegov trenutni zadatak
 - pojedinac ima različite razine ovlasti u različito vrijeme, ovisno o zadaći ili funkciji koju obavlja
 - spriječeno obavljanje nepotrebnih i moguće štetnih akcija
- ◆ **razdvajanje dužnosti (separation of duty - SoD)**
 - osjetljive zadatke u cijelosti ne može obaviti samo jedan korisnik
 - smanjuje se mogućnost zloporabe (važno načelo u financijskim i vojnim okruženjima)

Mehanizmi zaštite na razini SUBP

- identifikacija i dokazivanje autentičnosti
- upravljanje pristupom
- šifriranje podataka
- praćenje pristupa podacima
- maskiranje podataka



Integritet i sigurnost baze podataka

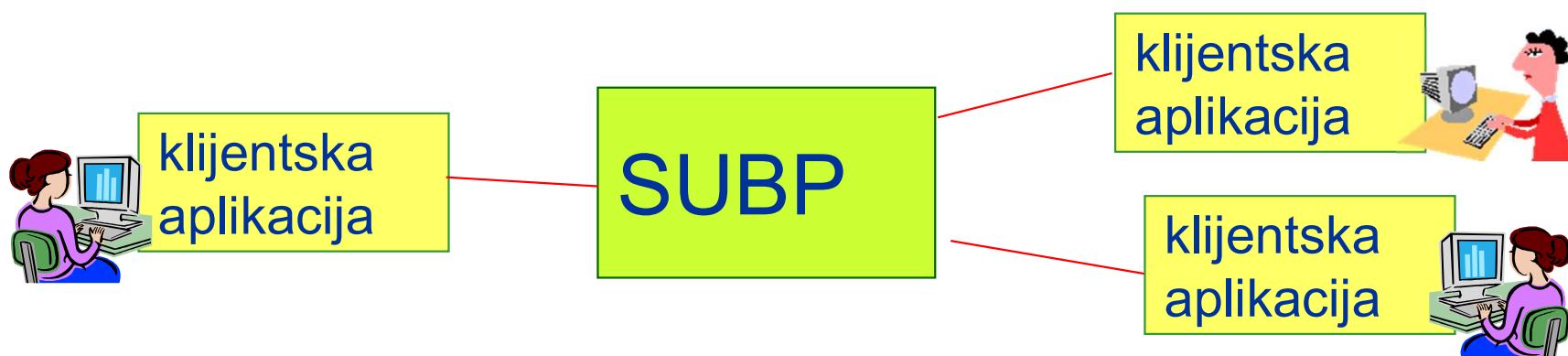
- ◆ Pojmovi *integritet* i *sigurnost* baze podataka se često spominju zajedno, međutim radi se o dva različita aspekta zaštite podataka
 - **Integritet baze podataka** (*database integrity*) - operacije nad podacima koje korisnici obavljaju su ispravne (tj. uvijek rezultiraju konzistentnim stanjem baze podataka)
 - "podaci se štite od ovlaštenih korisnika,"
 - **Sigurnost baze podataka** (*database security*) - korisnici koji obavljaju operacije nad podacima su ovlašteni za obavljanje tih operacija
 - "podaci se štite od neovlaštenih korisnika"
- Među ovim pojmovima postoje i sličnosti. U oba slučaja:
 - moraju biti definirana pravila koja korisnici ne smiju narušiti
 - pravila se pohranjuju u rječnik podataka
 - SUBP nadgleda rad korisnika i osigurava poštivanje pravila

Korisnici SUBP i ovjera autentičnosti

- ◆ administrator sustava (operacijskog sustava ili SUBP) omogućuje korisniku pristup sustavu (operacijskom sustavu ili SUBP) definiranjem jedinstvenog identifikatora korisnika (*user name*, *user ID*, *login ID*) i pripadne lozinke (*password*) koja je poznata samo dotičnom korisniku i sustavu
- ◆ korisnik koji pristupa sustavu (operacijskom sustavu ili SUBP) poznavanjem lozinke ovjerava svoju autentičnost (*authentication*)
- ◆ za ovjeru autentičnosti korisnika SUBP može koristiti
 - vlastite mehanizme
 - ili
 - vanjske mehanizme (npr. operacijski sustav)

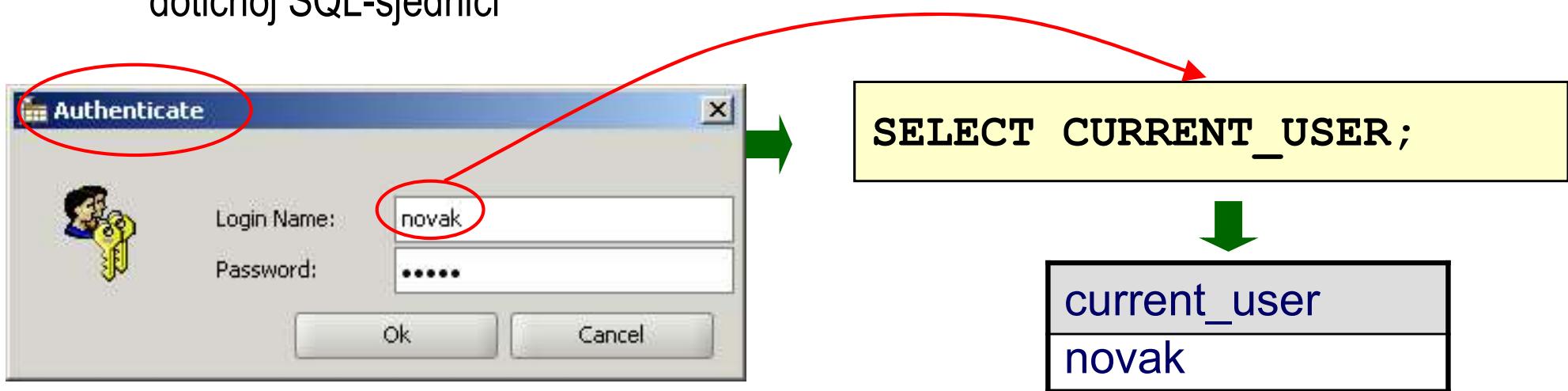
SQL-sjednica

- SQL-sjednica (*SQL-session*) je kontekst u kojem jedan korisnik obavlja niz SQL naredbi putem jedne veze (*SQL-Connection*) prema sustavu za upravljanje bazama podataka
 - SQL-sjednica **započinje** u trenutku kada korisnik, upotrebom klijentske aplikacije, **ostvari vezu** (*connect*) sa sustavom za upravljanje bazama podataka
 - SQL-sjednica **završava** u trenutku kada korisnik **prekine vezu** (*disconnect*) prema sustavu za upravljanje bazama podataka



Korisnici u SQL-u

- ◆ autentificirani korisnik
 - pri uspostavljanju SQL-sjednice korisnik se prijavljuje svojim identifikatorom korisnika, te lozinkom ovjerava svoju autentičnost
 - funkcija CURRENT_USER vraća vrijednost identifikatora korisnika koji se koristi u dotičnoj SQL-sjednici



- bilo koji korisnik (PUBLIC)
 - dodjelom dozvole "korisniku" PUBLIC, dozvolu za obavljanje operacije dobivaju svi sadašnji i budući korisnici

Korisnici u PostgreSQL-u

```
CREATE USER name [ [ WITH ] option [ . . . ] ]
```

where option can be:

```
SUPERUSER | NOSUPERUSER  
| CREATEDB | NOCREATEDB  
| CREATEUSER | NOCREATEUSER  
| [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'  
| INHERIT | NOINHERIT  
| . . .
```

- Korisnici se i u PostgreSQL sustavu definiraju na razini SUBP-a, a ne na razini pojedinačne baze podataka
- Za SUPERUSER-a ne postoje ograničenja:

```
CREATE USER the_boss WITH SUPERUSER
```

```
PASSWORD 'superSecret' ;
```

„Superuser status is dangerous and should be used only when really needed.”

- CREATEDB - korisnik dobiva ovlast kreiranja baze podataka na PostgreSQL SUBP
- NOCREATEDB - preddefinirano ponašanje.

Korisnici u PostgreSQL-u

```
...  
| CREATEUSER | NOCREATEUSER  
...
```

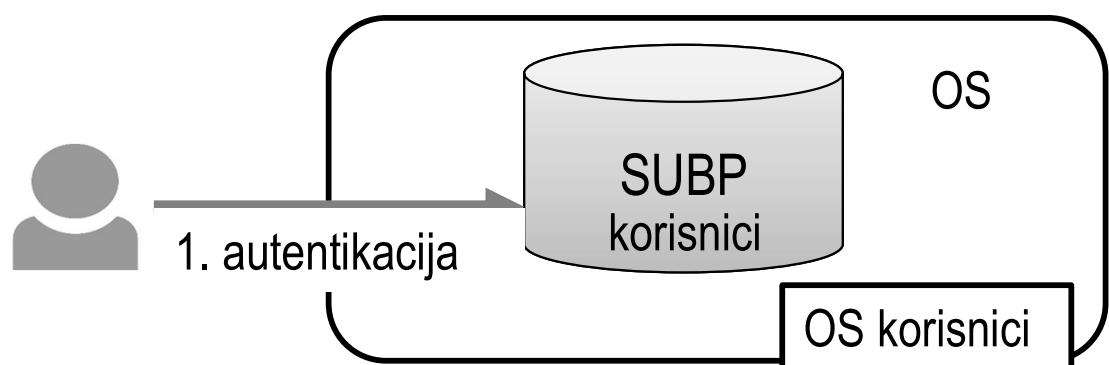
- CREATUSER - korisnik dobiva ovlast kreiranja drugih korisnika na PostgreSQL SUBP
- NOCREATEUSER - preddefinirano ponašanje.

```
CREATE USER bpadmin WITH CREATEDB CREATEUSER  
PASSWORD 'bpadminPwd' ;
```

Metode autentikacije

(PostgreSQL podržava čak 9):

- OS (trust auth)
- Vlastita (password auth)
- ...



IBM Informix: Dokazivanje autentičnosti

- ◆ interna autentikacija
 - ◆ lozinka pohranjena u poslužitelju baze podataka (tablica `sysIntAuthUsers` u bazi podataka `sysUser`) - šifrirana SHA-256 algoritmom (*Secure Hash Algorithm*)
- ◆ operacijski sustav
- ◆ posebni moduli (autentikacijski slojevi), npr:
 - Pluggable Authentication Modules (PAM)
 - za IBM Informix na UNIX i Linux OS
 - API za upravljanje autentikacijom, korisničkim računima, sjednicama i lozinkama
 - Lightweight Directory Access Protocol (LDAP) Authentication Support za Windows
 - za autentikaciju korisnika koristi se LDAP poslužitelj

Oracle: Zaštita lozinki (1)

- ◆ automatsko i transparentno **šifriranje lozinke** prije slanja preko mreže
- ◆ pohrana **sažetka (hash)** šifriranog SHA-512 algoritmom (*Secure Hash Algorithm*)
- ◆ provjera složenosti lozinke
- ◆ sprječavanje probijanja lozinke u slučaju višekratnih neuspjelih pokušaja prijave
 - ograničavanje broja neuspješnih pokušaja prijave
 - postepeno povećavanje vremena prije ponovnog pokušaja prijave - smanjenje broja pokušaja
 - isključivanje korisničkog računa

Oracle: Zaštita lozinki (2)

- definiranje sigurnosnih profila

```
CREATE PROFILE obicniKorisnik LIMIT  
    FAILED_LOGIN_ATTEMPTS 5  
    PASSWORD_LOCK_TIME 2;
```

- isključenje korisničkog računa na dva dana u slučaju pet neuspjelih pokušaja prijave
- DEFAULT profil

- povezivanje profila s korisnikom

```
CREATE USER u1 IDENTIFIED BY "pass11!1"  
    PROFILE obicniKorisnik;
```

- zaključavanje korisničkog računa

```
ALTER USER u1 ACCOUNT LOCK;
```

Parametar

FAILED_LOGIN_ATTEMPTS - broj neuspješnih pokušaja prijave prije zaključavanja korisničkog računa

PASSWORD_LIFE_TIME - trajanje lozinke (broj dana)

PASSWORD_LOCK_TIME - koliko je dana račun zaključan nakon određenog broja uzastopnih neuspjelih pokušaja prijave

PASSWORD_GRACE_TIME - broj dana do isteka lozinke (period odgode - dozvoljena prijava uz upozorenje)

PASSWORD_REUSE_TIME - mogućnost ponovnog korištenja iste lozinke (u danima)

PASSWORD_REUSE_MAX - potreban broj promjena lozinke prije ponovnog korištenja lozinke

PASSWORD_VERIFY_FUNCTION - funkcija za provjeru složenosti lozinke



Upravljanje pristupom

Upravljanje pristupom

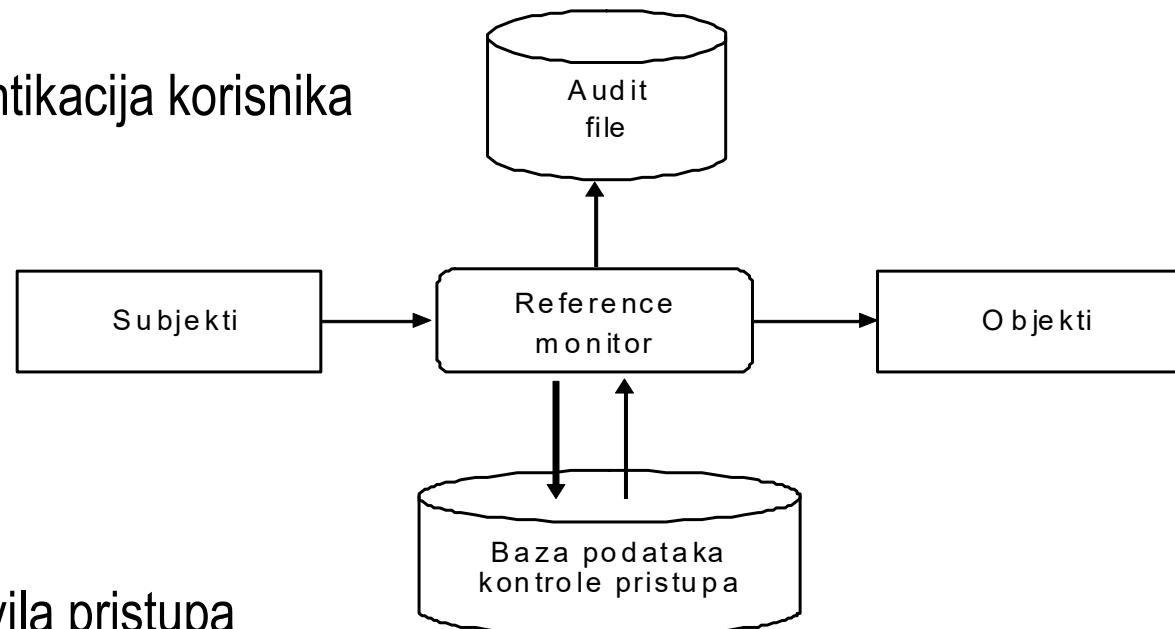
- ◆ niz postupaka kojima se utvrđuje i evidentira pokušaj pristupa, te odobrava ili odbija pristup na temelju unaprijed utvrđenih pravila
- ◆ razvoj sustava za upravljanje pristupom - višefazni proces
 - **sigurnosna politika** - pravila pristupa na visokoj razini (zakonski, socijalni, etički aspekt)
 - temeljena na načelu *treba-znati* (*need-to-know*), kompetentnosti, nadležnosti, sukobu interesa...
 - na nju mogu utjecati zakonska i etička pitanja, politike na državnoj ili korporativnoj razini ...
 - dinamična - mijenja se u skladu s promjenama poslovnih faktora, regulativa i uvjeta u okruženju
 - problem preslikavanja nejasnih i dvosmislenih zahtjeva u dobro definirana i jednoznačna pravila

Upravljanje pristupom

- ***sigurnosni model*** - formalni prikaz sigurnosne politike (strategijski aspekt)
 - ***diskrecijsko upravljanje pristupom*** (*discretionary access control* - DAC)
 - ***mandatno upravljanje pristupom*** (*mandatory access control* - MAC)
 - ***upravljanje pristupom temeljeno na ulogama*** (*role-based access control* - RBAC)
- ***sigurnosni mehanizam*** - (operativni aspekt) funkcije kojima je implementirano upravljanje pristupom

Mehanizmi upravljanja pristupom

- ◆ *reference monitor* - pouzdana komponenta koja upravlja svakim pokušajem pristupa objektu sustava i utvrđuje **je li taj pristup u skladu sa sigurnosnom politikom** utjelovljenom u bazi podataka upravljanja pristupom
 - uspoređuje sigurnosne atribute korisnika (npr. identifikator korisnika, grupa kojoj korisnik pripada, razina povjerenja iskazana korisniku) s onima koje imaju resursi (npr. oznaka osjetljivosti)
 - preuvjet: uspješna identifikacija i autentikacija korisnika



- ◆ **autorizacija** - postupak evidentiranja pravila pristupa
 - ◆ informacije koje opisuju prava pristupa moraju biti zaštićene od izmjene
 - ◆ sigurnosni mehanizmi implementirani kroz SQL

Elementi sustava za upravljanje pristupom

- ◆ **korisnik** – entitet koji koristi računalni sustav (osoba, uređaj)
 - *sjednica (session)* - instanca dijaloga korisnika sa sustavom
- ◆ **subjekt** – aktivni entitet koji može inicirati zahtjev za obavljanje operacija na objektima
 - proces koji djeluje u ime korisnika
 - korisnik može imati više aktivnih subjekata
- ◆ **objekt** - entitet sustava na kojem može biti obavljena operacija
- ◆ **operacija** - aktivan proces pozvan od strane subjekta, koji nakon poziva izvršava funkcije
- ◆ **dozvola (pravo pristupa, ovlast)** određenog načina pristupa objektu u sustavu
 - kombinacija objekta i operacije
 - za omogućavanje izvođenja iste operacije (npr. SELECT) na dva različita objekta (npr. tablice *student* i *predmet*) potrebne dvije dozvole
 - za omogućavanje obavljanja dviju različitih operacija (npr. UPDATE i SELECT) na istom objektu (npr. tablici *student*) potrebne su dvije dozvole

Elementi sustava za upravljanje pristupom - dozvola

- **pozitivna dozvola** - ako ne postoji, zatraženi pristup je odbijen
 - **negativna dozvola (tj. zabrana)** - ako postoji, zatraženi pristup je odbijen
-
- ◆ klasični pristupi upravljanja pristupom:
 - **zatvorena politika** - dozvoljen pristup za koji postoji (pozitivna) dozvola
 - **otvorena politika** - uskraćen pristup za koji postoji zabrana (tj. "negativna dozvola")
 - ako zabrana ne postoji, pristup je dozvoljen

Elementi sustava za upravljanje pristupom - dozvola

- ◆ problemi kod kombiniranog korištenja pozitivnih i negativnih dozvola:
 - *nepotpunost* - za pristup nije specificirana dozvola
 - može se izbjegići definiranjem pretpostavljene politike
 - *nekonzistentnost* - za pristup postoji i negativna i pozitivna dozvola
 - može se izbjegići definiranjem politike razrješavanja konflikata
- ◆ politike razrješavanja konflikata:
 - *nepostojanje konflikata* – postojanje konflikta smatra se pogreškom
 - *prioritetne su negativne dozvole*
 - *prioritetne su pozitivne dozvole*
 - *ništa nema prioritet* – istovremeno postojanje konfliktnih dozvola poništava te dozvole (kao da nije specificirana nikakva dozvola)



Diskrečijsko upravljanje pristupom

Diskrecijsko upravljanje pristupom

- ◆ upravljanje pristupom na temelju:
 - **identiteta korisnika** koji zahtijeva pristup i
 - eksplicitnih **pravila pristupa** koja utvrđuju tko može izvesti koje akcije na kojim objektom sustava

Način ograničavanja pristupa objektima na temelju identiteta subjekata i/ili grupa kojoj oni pripadaju. Upravljanje je diskrecijsko u smislu da je subjekt s nekom dozvolom pristupa sposoban dati tu dozvolu (možda indirektno) nekom drugom subjektu (osim ako je to ograničeno MAC-om (Mandatory Access Control))

[US Department of Defense, Trusted Computer Security Evaluation Criteria (TCSEC),
DoD 5200.28-STD, 1985]

- ◆ koncept vlasništva nad objektom
 - **vlasnik objekta određuje kome se dozvoljava pristup**

Matrica autorizacijskih pravila (Matrica pristupa)

- ◆ Lampson (1974); Harrison, Ruzzo i Ullmann - HRU model (1976)
- ◆ stanje sustava - trojka (S , O , A)
- ◆ stanje autorizacije predstavljeno matricom pristupa A
 - stupci – **objekti** (o)
 - retci – **subjekti** (s)
 - element matrice – $A[s, o]$ - **ovlasti** subjekta s na objektu o
- ◆ zahtjev (s, o, a) - pristup dozvoljen ako a postoji u elementu matrice za s i o

KORISNICI	OBJEKTI		
	<i>Datoteka</i> ₁	<i>Datoteka</i> ₂	<i>Program</i> ₁
k_1	read, write	read, write	execute
k_2	read		
k_3		read	execute

System R model

- ◆ 1970-e - IBM San Jose
- ◆ objekti - tablice i pogledi
- ◆ dozvole - *select, update, insert, delete, drop*
- ◆ subjekt koji kreira tablicu – vlasnik (ima sve ovlasti nad tablicom)
- ◆ dozvola s mogućnošću prenošenja (GRANT opcija)
 - subjekt kojemu je vlasnik dodijelio dozvolu može tu dozvolu dodijeliti drugim subjektima (bez GRANT opcije ili s GRANT opcijom)
- ◆ ukidanje dozvola
 - **kaskadno**, na temelju trenutka dodjeljivanja
 - ukidaju se sve dozvole koje je dodijelio korisnik kojem se ukida dozvola, a koje ne bi mogle biti dodijeljene bez postojanja dozvole koja se ukida
 - ukidanje dozvola se iterativno primjenjuje na sve korisnike koji su primili dozvole pristupa od svih korisnika s kojih je dozvola povučena
 - zahtijeva pamćenje povijesti i vremena dodjeljivanja dozvola

Diskrecijsko upravljanje pristupom u bazama podataka

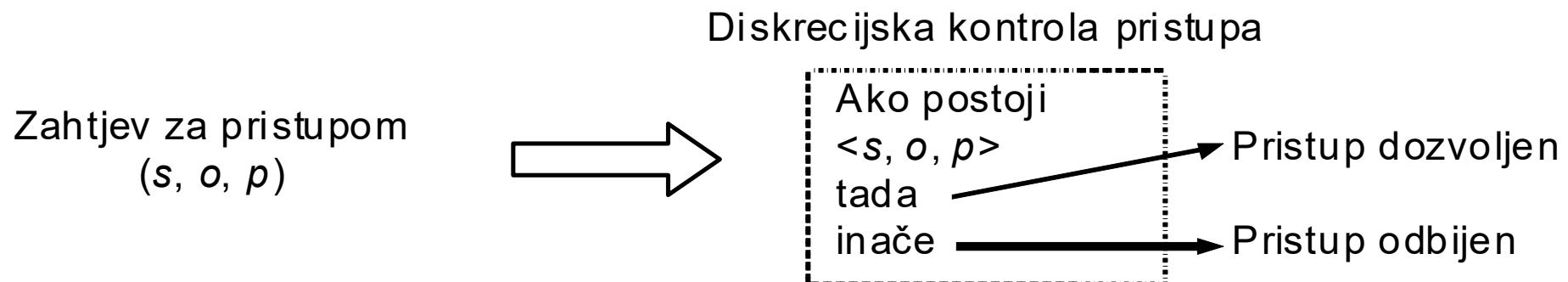
- ◆ podržano SQL standardom
- ◆ podržano u većini današnjih SUBP

- ◆ određenom korisniku potrebno je eksplicitno dodijeliti dozvolu za obavljanje određene operacije nad određenim objektom (autorizacija)
 - ◆ dozvole su opisane trojkama **<korisnik, objekt, vrsta operacije>**, npr:
 - <horvat, ispit, čitanje>
 - <horvat, ispit, izmjena>
 - <novak, predmet, čitanje>

 - ◆ podaci o dodijeljenim dozvolama pohranjuju se u rječnik podataka

Diskrecijsko upravljanje pristupom u bazama podataka

- ◆ prije obavljanja svake operacije, **SUBP provjerava ima li korisnik dozvolu za obavljanje operacije nad objektom** (upravljanje pristupom)
 - ◆ kada korisnik *novak* pokuša obaviti operaciju čitanja objekta (tablice) *predmet*, SUBP provjerava postoji li dozvola u obliku trojke *<novak, predmet, čitanje>*



Upravljanje pristupom u SQL-u

- ◆ **Mehanizmi upravljanja pristupom**
 - naredbe za dodjeljivanje (**GRANT**) i ukidanje dozvola (**REVOKE**)
 - virtualne tablice (view)
 - pohranjene procedure
 - modifikacija upita
- ◆ **Objekti**
 - tablica (table)
 - atribut (stupac tablice, column)
 - virtualna tablica (pogled, view)
 - pohranjena procedura
 - baza podataka (neki sustavi, npr. IBM Informix)
- ◆ **Vlasnik objekta** - korisnik koji je kreirao objekt
 - implicitno dobiva dozvole za obavljanje svih vrsta operacija nad objektom, uključujući
 - dozvole za dodjeljivanje svih vrsta dozvola nad tim objektom drugim korisnicima i
 - uništavanje objekta

SUBP i baze podataka

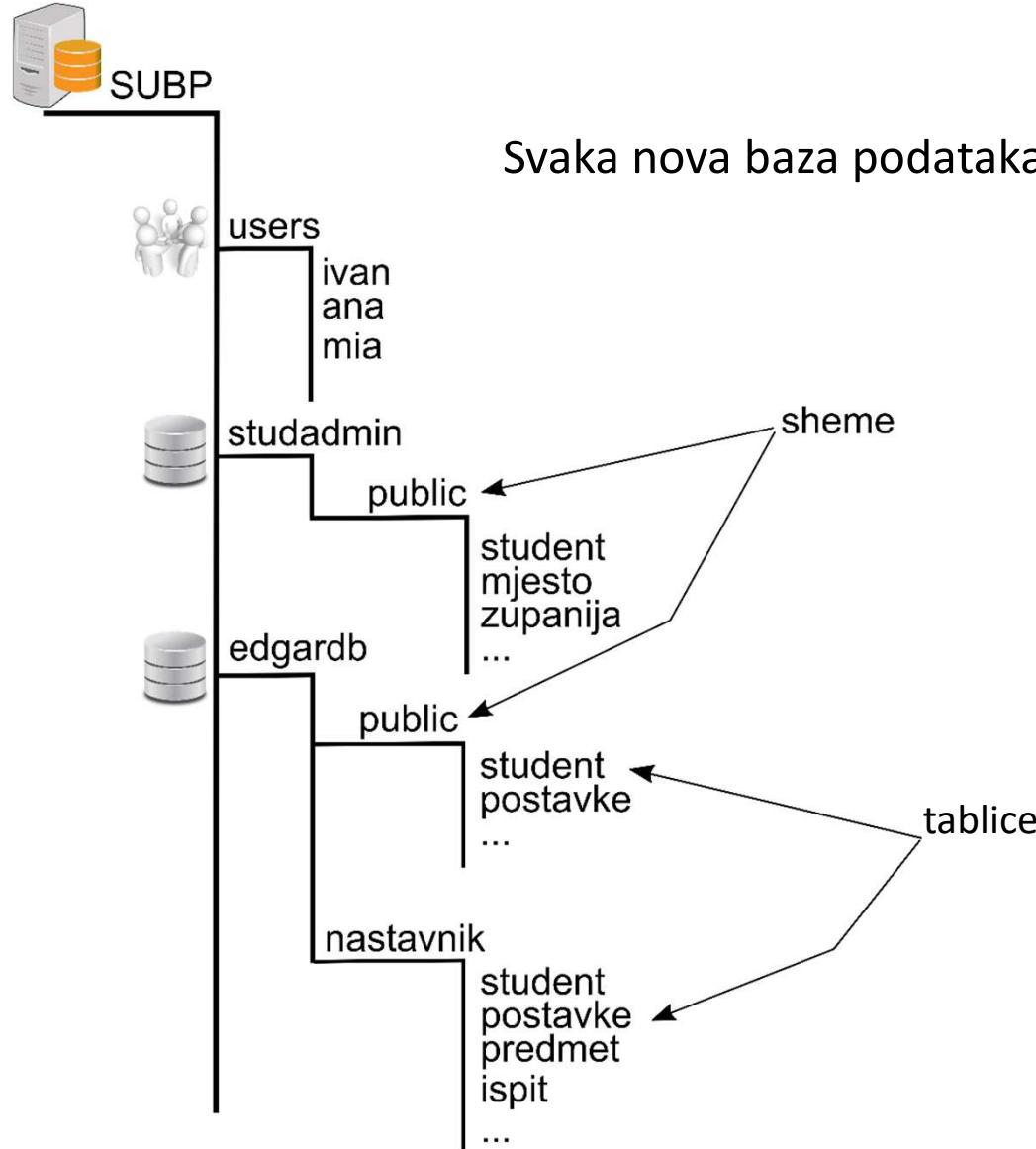
- ◆ SUBP općenito sadrži više (N) baza podataka
 - Korisnici su dijeljeni, na razini cijelog SUBP-a
 - To ne znači da imaju pravo pristupa svim bazama podataka
 - Ne mogu postojati dva korisnika „ivan”
 - Korisnik se pri spajanju na SUBP zapravo spaja na odabranu bazu podataka (npr. *studadmin*)

Sheme (eng. schemas)

◆ PostgreSQL:

- Baza podataka sadrži jednu ili više **shema**
- Sheme sadrže tablice, virtualne tablice (, funkcije, ...)
- Različite sheme mogu sadržavati istoimene tablice
- Sheme analogne s:
 - Mapama u datotečnom sustavu (s tim da se ne mogu gnijezditi)
 - Imenskim područjima (*namespaces*) u programskim jezicima

SUBP/BP/Shema



Zašto sheme?

- ◆ Omogućiti višekorisnički pristup bazi podataka, pri čemu želimo razdijeliti korisnike, odnosno pristup objektima baze podatka (tablice, funkcije, ...)
- ◆ Organizirati tablice u logičke grupe, kako bi s njima lakše upravljali (npr. javno, interno, admin, ...)
- ◆ Uspostaviti sustav dozvola (objašnjeno kasnije)

Sheme - SQL

◆ Stvaranje:

```
CREATE SCHEMA student;

CREATE TABLE student.postavke (
    username TEXT primary key,
    cm_skin TEXT not null
);
```

◆ Pristup:

```
-- schema.table
-- database.schema.table
SELECT * FROM student.postavke
```

◆ Brisanje:

```
DROP SCHEMA student;
-- cannot drop schema student because
-- other objects depend on it
DROP SCHEMA student CASCADE;
-- obrisani i sadržani objekti!!
```

Shema *public* je opcionalna, može se obrisati.

Određivanje sheme (SSP - schema search path)

- ◆ Ako se u SQL naredbi ne upotrijebi puno ime tablice, SUBP pretražuje SSP i pokušava ga odrediti:
 - Koristi se **prva** pronađena tablica
 - Ako se ne pronađe, javlja se greška (to ne znači da tablica tog imena ne postoji!)
- ◆ Prva postojeća shema u SSP se zove **trenutna shema**
- ◆ Trenutna shema se koristi za stvaranje novih objekata (tablica, ...)
- ◆ Trenutne postavke za SSP se mogu dobiti sljedećom naredbom:

```
SHOW search_path;
```

```
search_path
```

```
"$user", public
```

- Kao trenutnu shemu PostgreSQL će odrediti shemu "\$user", ako takva postoji
- Ako ne postoji, trenutna shema postaje *public*
- ◆ Ako je CURRENT_USER npr. *tibor*, tada je "\$user" shema koja se zove *tibor*

Određivanje sheme - Primjer

Vlasnik baze podataka kreirao je korisnika *tibor* i sljedeće dvije sheme:

```
CREATE USER tibor WITH PASSWORD 'tiborPwd';
CREATE SCHEMA student;
CREATE SCHEMA nastavnik;
```

Nakon uspostavljanja korisničke sjednice s bazom podataka, *tibor* obavlja sljedeću naredbu:

```
CREATE TABLE ocjena
(sifOcjena int PRIMARY KEY,
 opisOcjena CHAR(2) NOT NULL,)
```

U kojoj shemi će biti kreirana tablica ocjena?

U shemi *public* jer shema *tibor* ne postoji.

Gornja naredba je ekvivalentna naredbi:

```
CREATE TABLE public.ocjena
(sifOcjena int PRIMARY KEY,
 opisOcjena CHAR(2) NOT NULL,)
```

- ◆ Različiti SUBP imaju različita rješenja za dodjeljivanje dozvola na razini baze podataka.
- ◆ PostgreSQL:
 - CONNECT
 - Dozvoljava spajanje (uspostavljanje SQL-sjednice) na bazu podataka
 - Spojeni korisnik može obavljati operacije nad objektima za koje je dobio dozvolu od vlasnika objekta ili je njihov vlasnik
 - Preddefinirano ponašanje je da korisnik PUBLIC (pa i **tibor**) ima CONNECT dozvolu na bazu podataka u PostgreSQL SUBP
 - CREATE
 - Dozvoljava stvaranje novih shema u bazi podataka

- ◆ PostgreSQL:

- **USAGE**

- Nužan preduvjet za pristupanje objektima sadržanim u shemi. Ne podrazumijeva nikakve daljnje dozvole za konkretnе objekte u shemi.

- **CREATE**

- Dozvoljava stvaranje novih objekata (tablice, funkcije, ...) u shemi.

- Preddefinirano ponašanje:

- Korisnik nema dozvolu pristupa nijednom objektu sheme kojoj nije vlasnik.
 - Za pristup mu vlasnik sheme treba dodijeliti dozvolu USAGE
 - Za kreiranje objekata u shemi, dodatno mora dobiti CREATE
 - PUBLIC ima **CREATE** i **USAGE** dozvole za shemu *public*

Vrste dozvola u SQL-u na razini [virtualne] tablice (*tablePrivilege*)

- ◆ **SELECT [(*columnList*)]**
 - čitanje n-torki (ili vrijednosti navedenih atributa) [virtualne] tablice
- ◆ **UPDATE [(*columnList*)]**
 - Izmjena n-torki (ili vrijednosti navedenih atributa) [virtualne] tablice
- ◆ **INSERT**
 - unos n-torki [virtualne] tablice
- ◆ **DELETE**
 - brisanje n-torki [virtualne] tablice
- ◆ **ALL PRIVILEGES**
 - sve do sada navedene vrste operacija nad [virtualnom] tablicom
- ◆ itd. gore je naveden samo dio dozvola

SQL naredbe za dodjeljivanje i ukidanje dozvola

- ◆ **GRANT** *dbPrivilege* ON DATABASE name TO { PUBLIC | *userList* }
 - ◆ **REVOKE** *dbPrivilege* ON DATABASE name FROM { PUBLIC | *userList* }
-
- ◆ **GRANT** *schemaPrivilege* ON SCHEMA name TO { PUBLIC | *userList* }
 - ◆ **REVOKE** *schemaPrivilege* ON SCHEMA name FROM { PUBLIC | *userList* }
-
- ◆ **GRANT** *tablePrivilegeList* ON { *tableName* | *viewName* }
 TO { PUBLIC | *userList* }
 [WITH GRANT OPTION]
 - ◆ **REVOKE** *tablePrivilegeList* ON { *tableName* | *viewName* }
 FROM { PUBLIC | *userList* }
 [CASCADE | RESTRICT]

PostgreSQL - preddefinirane dozvole korisnika PUBLIC

- ◆ Ključna riječ PUBLIC (**<> shema public!**)
 - Označava sve korisnike, čak i one koji će tek nastati
- ◆ PgSQL - preddefinirane dozvole korisnika PUBLIC:
 - Dozvola uspostavljanja konekcije sa svim bazama na PgSQL SUBP

```
GRANT CONNECT ON DATABASE * TO PUBLIC;
```
 - Dozvole USAGE i CREATE za sve sheme public u svim bazama na PgSQL SUBP

```
GRANT ALL (USAGE, CREATE) ON SCHEMA public TO
PUBLIC;
```
 - Primijetite da PUBLIC nema nikakvu dozvolu na razini tablica u shemi **public**

Primjer

Mnogi koriste ovakav sustav (u produkciji), za nešto strože inicijalne postavke sigurnosti:

```
--ACCESS DB
REVOKE CONNECT ON DATABASE dbName FROM PUBLIC;
GRANT CONNECT ON DATABASE dbName TO user;

--ACCESS SCHEMA
REVOKE ALL      ON SCHEMA public FROM PUBLIC;
GRANT USAGE     ON SCHEMA public TO user;

--ACCESS TABLES (pretpostavka je da postoje dolje navedene uloge)
GRANT SELECT          ON ALL TABLES IN SCHEMA public TO read_only;
GRANT SELECT, INSERT,
       UPDATE, DELETE   ON ALL TABLES IN SCHEMA public TO read_write;
GRANT ALL             ON ALL TABLES IN SCHEMA public TO admin;
```

Primjer 1 (PostgreSQL):

student	matBr	ime	prez	pbr	adresa
---------	-------	-----	------	-----	--------

ispit	matBr	nazPred	datIsp	ocj
-------	-------	---------	--------	-----

Korisnik badmin treba

- kreirati bazu podataka studBaza.
- korisniku PUBLIC ukinuti dozvolu spajanja na studBaza
- korisniku PUBLIC ukinuti sve dozvole za shemu *public* u studBaza
- kreirati tablice student i ispit
- kreirati korisnike *horvat*, *novak* i *kolar* i omogućiti im spajanje na studBaza i korištenje *public* sheme u studBaza
- ◆ korisnik *horvat* treba dobiti dozvole:
 - pregled svih podataka u tablicama student i ispit
 - unos, izmjena, brisanje svih podataka u tablici ispit
- ◆ korisnik *novak* treba dobiti dozvole:
 - pregled svih podataka u tablici student
 - izmjena poštanskog broja i adrese u tablici student
- ◆ korisnik *kolar* treba dobiti dozvolu:
 - pregled svih podataka u tablici student, osim adrese

Primjer 1 (nastavak, PostgreSQL):

postgres ← naredbu obavlja korisnik **postgres (SUPERUSER)**

```
CREATE USER badmin WITH CREATEDB CREATEROLE  
    PASSWORD 'badminPwd';
```

→ korisnik badmin dobiva dozvolu kreiranja baza podataka i korisnika.

badmin ← naredbe obavlja korisnik **badmin**

```
CREATE DATABASE studbaza;  
  
REVOKE CONNECT ON DATABASE studBaza  
    FROM PUBLIC;
```

→ korisnik badmin je vlasnik baze podataka studBaza. Može ukinuti preddefiniranu dozvolu CONNECT korisniku PUBLIC.

postgres

```
REVOKE ALL ON SCHEMA public FROM PUBLIC;
```

→ vlasnik sheme *public* u svakoj bazi podataka je korisnik *postgres* (specifičnost PgSQL). Korisnik *badmin* nema ovlasti za ovu naredbu.

badmin

```
CREATE TABLE student (...);  
CREATE TABLE ispit (...);
```

→ kreiranje novih objekata u bazi.
tablice će biti kreirane u shemi public.

```
CREATE USER horvat;  
CREATE USER kolar;  
CREATE USER novak;
```

→ kreiranje korisnika s mogućnošću uspostavljanja SQL-sjednice na razini SUBP

Primjer 1 (nastavak, PostgreSQL):

bpadmin

```
GRANT CONNECT ON DATABASE studbaza TO horvat;  
GRANT CONNECT ON DATABASE studbaza TO novak;  
GRANT CONNECT ON DATABASE studbaza TO kolar;  
  
GRANT USAGE ON SCHEMA public TO horvat;  
GRANT USAGE ON SCHEMA public TO novak;  
GRANT USAGE ON SCHEMA public TO kolar;
```

```
GRANT SELECT ON student TO horvat;
```

```
GRANT SELECT, INSERT  
, UPDATE, DELETE ON ispit  
TO horvat;
```

```
GRANT SELECT ON student TO novak;
```

```
GRANT UPDATE(pbr, adresa)  
ON student TO novak;
```

```
GRANT SELECT(matBr, ime  
, prez, pbr)  
ON student TO kolar;
```

- dozvole spajanja na studBaza.
Treba jer je ukinuta CONNECT
dozvola za PUBLIC.
- dozvola korištenja sheme *public*.
Treba jer je ukinut USAGE i
CREATE za PUBLIC.
- dozvola korisniku *horvat* za pregled
podataka u tablici student
- dozvole korisniku *horvat* za pregled,
unos, izmjenu i brisanje podataka u
tablici ispit
- dozvola korisniku *novak* za pregled
podataka u tablici student
- dozvola korisniku *novak* za izmjenu
vrijednosti atributa u tablici student
- dozvola korisniku *kolar* za pregled svih
podataka u tablici student, osim adrese

Primjer 2 (PostgreSQL):

badmin

```
CREATE DATABASE studBaza;  
CREATE SCHEMA student;  
  
CREATE TABLE student.postavke (  
    username text primary key, ...);  
CREATE TABLE postavkePub(  
    username text primary key, ...);
```



korisnik badmin kreira bazu podataka studBaza, te dvije tablice, jednu u shemi student, drugu u PUBLIC shemi

Sjetimo se: PostgreSQL (*default*) daje CONNECT dozvolu korisniku PUBLIC!

tibor

```
CREATE TABLE postavkeTib (...)  
INSERT INTO postavkeTib VALUES (...);
```



Može, jer :

- ima CONNECT (bez CONNECT ne bi mogao uspostaviti SQL-sjednicu),
- ima USAGE i CREATE za shemu public u kojoj se stvara *postavketib*
- je vlasnik *postavketib* pa može obaviti INSERT

tibor

```
SELECT * FROM postavkePub;  
INSERT INTO postavkePub VALUES (...);  
SELECT * FROM student.postavke;  
INSERT INTO student.postavke ....;  
CREATE TABLE student.T2(...);  
CREATE SCHEMA moja;
```



NE može, jer:

- USAGE na shemu *public* ne uključuje dozvole za operacije nad tablicama
- Nije mu dana dozvola za *student.postavke*
- Nema dozvole (USAGE) za shemu *student*
- Nema dozvole za stvaranje sheme

Primjer 2 (nastavak):

tibor

```
DROP TABLE postavkePub;
```



ne može jer nije vlasnik objekta (niti je SUPERUSER)

kolar

```
SELECT * FROM postavkePub;
```



NE može, jer nema dozvole za operacije nad postavkePub

tibor

```
GRANT CONNECT ON DATABASE  
studBaza TO kolar;
```



ne može jer nije SUPERUSER

tibor

```
GRANT SELECT  
ON postavkeTib TO kolar;
```



Može, jer je **vlasnik** tablice postavkeTib

Primjer 2 (nastavak):

postgres

```
GRANT CREATE ON DATABASE  
studBaza TO tibor;
```

→ Može, jer je SUPERUSER

tibor

```
CREATE SCHEMA tibor;
```

→ Može, jer sad ima dozvolu

tibor

```
CREATE TABLE tibor.postavke(...);  
GRANT SELECT ON tibor.postavke TO kolar;
```

→ Može, jer je vlasnik sheme

kolar

```
SELECT * FROM tibor.postavke;
```

→ Ne može, jer nema dozvolu na
shemu (ima samo na tablicu)

tibor

```
GRANT USAGE ON SCHEMA tibor TO kolar;
```

→ Može, jer je vlasnik sheme

kolar

```
SELECT * FROM tibor.postavke;
```

→ Može

Zaštita i sigurnost informacijskih sustava

Sigurnost baza podataka, 2. dio

Dodjeljivanje prenosivih dozvola

- ◆ Ako se korisniku dozvola dodijeli uz navođenje opcije WITH GRANT OPTION, korisnik će moći dodjeljivati tu istu dozvolu ostalim korisnicima (unatoč tome što nije vlasnik objekta)

Primjer:

korisnik1

```
CREATE TABLE ispit (...);  
GRANT SELECT ON ispit TO korisnik2 WITH GRANT OPTION;  
GRANT SELECT ON ispit TO korisnik3 WITH GRANT OPTION;
```

korisnik2

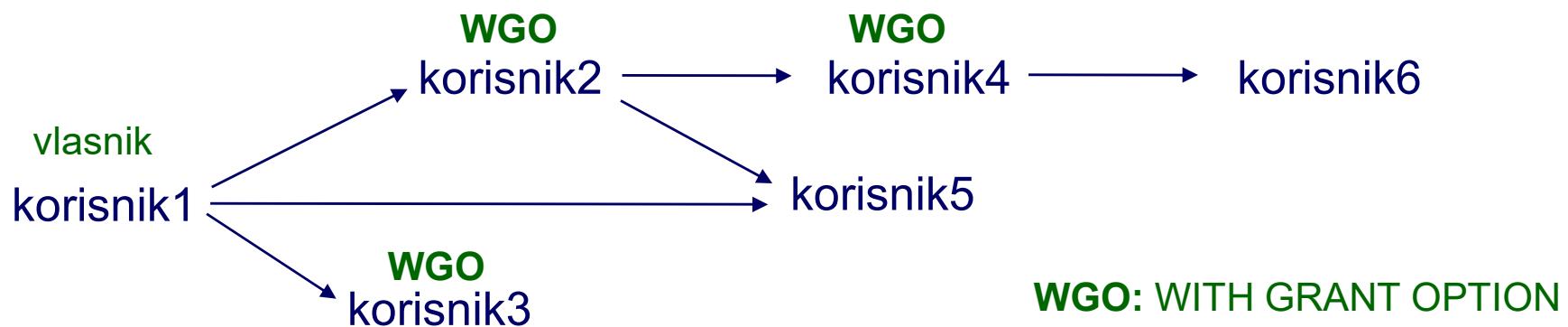
```
GRANT SELECT ON ispit TO korisnik4 WITH GRANT OPTION;  
GRANT SELECT ON ispit TO korisnik5;
```

korisnik4

```
GRANT SELECT ON ispit TO korisnik6;
```

korisnik1

```
GRANT SELECT ON ispit TO korisnik5;
```



Ukidanje dozvola

- korisnik koji je dozvolu dodijelio, tu istu dozvolu može ukinuti naredbom REVOKE

Primjer:

- vlasnik baze podataka studBaza je korisnik badmin
- vlasnik tablice mjesto je korisnik horvat

horvat

```
GRANT SELECT, UPDATE ON mjesto TO novak WITH GRANT OPTION;
```

novak

```
GRANT SELECT, UPDATE ON mjesto TO kolar;
```

- npr. naredbu:

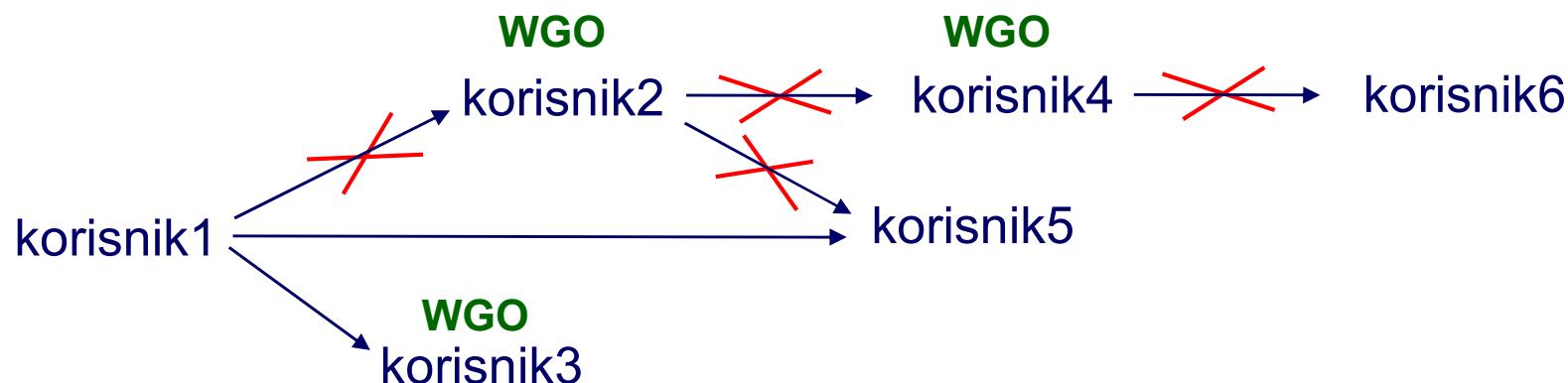
```
REVOKE UPDATE ON mjesto FROM kolar;
```
- može obaviti korisnik novak jer je novak korisnik koji je dozvolu dodijelio

Ukidanje dozvola dodijeljenih temeljem WITH GRANT OPTION

- ukidanjem dozvole korisniku x (koji je dozvole dalje dodjeljivao temeljem ovlasti stečene pomoću WITH GRANT OPTION) **uz primjenu opcije CASCADE**, dozvola se ukida i svim ostalim korisnicima koji su dotičnu dozvolu stekli od korisnika x (neposredno ili posredno)

Primjer: **korisnik1**

```
REVOKE SELECT ON ispit FROM korisnik2 CASCADE;
```



- obavljanjem naredbe dozvolu gube korisnik2, korisnik4 i korisnik6
- korisnik5 će izgubiti dozvolu koju je dobio od korisnika2, ali će zadržati dozvolu koju je dobio od korisnika1
- ukoliko se opcija CASCADE ne navede**, naredba REVOKE neće uspjeti ako postoji dodatne neposredne dozvole

IBM Informix: Dodjeljivanje i ukidanje dozvola

- ◆ na razini baze podataka:

```
GRANT dbPrivilege TO { PUBLIC | userList }
```

```
REVOKE dbPrivilege FROM { PUBLIC | userList }
```

- ◆ na razini [virtualne] tablice:

```
GRANT tablePrivilegeList ON { tableName | viewName }  
    TO { PUBLIC | userList | roleList }  
    [ WITH GRANT OPTION ]
```

```
REVOKE tablePrivilegeList ON { tableName | viewName }  
    FROM { PUBLIC | userList | roleList }  
    [ CASCADE | RESTRICT ]
```

IBM Informix: Vrste dozvola na razini baze podataka

CONNECT	<ul style="list-style-type: none">uspostavljanje SQL-sjednice i obavljanje operacija nad objektima za koje je korisnik dobio dozvolu od vlasnika objekta ili je njihov vlasnik, kreiranje virtualnih i privremenih tablica
RESOURCE	<ul style="list-style-type: none">CONNECT + kreiranje novih objekata u bazi podataka (tablica, indeksa, ograničenja, pohranjenih procedura ...)
DBA	<ul style="list-style-type: none">RESOURCE + neovisno o vlasništvu i dozvolama nad objektima u bazi podataka: sve vrste operacija nad svim objektima, uništavanje svih objekata (uključujući i bazu podataka)korisnik koji kreira bazu podataka je vlasnik te baze podataka i implicitno dobiva DBA (<i>Database administrator</i>) dozvolu

Primjer:

DBA

```
GRANT DBA TO u1;  
GRANT RESOURCE TO u1;
```

- nema učinka dodjeljivanje RESOURCE dozvole korisniku koji već ima DBA dozvolu - korisnik *u₁*, i dalje ima DBA dozvolu
- ukidanjem DBA dozvole, korisnik i dalje ima CONNECT dozvolu
- za sprečavanje uspostavljanja SQL-sjednice potrebno je ukinuti i CONNECT dozvolu

```
REVOKE DBA FROM u1;  
REVOKE CONNECT FROM u1;
```

IBM Informix: Vrste dozvola na razini [virtualne] tablice

SELECT [(columnList)]	◆ čitanje n-torki (ili vrijednosti navedenih atributa) [virtualne] tablice
UPDATE [(columnList)]	◆ izmjena n-torki (ili vrijednosti navedenih atributa) [virtualne] tablice
INSERT[(columnList)]	◆ unos n-torki (ili vrijednosti navedenih atributa) [virtualne] tablice
DELETE	◆ brisanje n-torki [virtualne] tablice
REFERENCES [(columnList)]	◆ korištenje tablice (ili samo navedenih atributa) kao pozivane tablice pri definiranju stranog ključa)
INDEX	◆ kreiranje indeksa nad tablicom
ALTER	◆ izmjena strukture tablice i definiranje integritetskih ograničenja
ALL PRIVILEGES	◆ sve vrste operacija nad [virtualnom] tablicom

Primjer:

```
UPDATE nastavnik SET koef = 0.9*koef
WHERE NOT EXISTS
  (SELECT * FROM predmetGrupa
   WHERE predmetGrupa.sifNastavnik = nastavnik.sifNastavnik
     AND predmetGrupa.akGodina = 2011);
```

Potrebne dozvole:

- ◆ pristup bazi podataka
- ◆ SELECT: *nastavnik.sifNastavnik, nastavnik.koef , predmetGrupa.sifNastavnik, predmetGrupa.akGodina*
- ◆ UPDATE: *nastavnik.koef*

ORACLE: kategorije dozvola

- ◆ ORACLE: dozvole se mogu svrstati u dvije općenite kategorije:

- sistemske dozvole

- omogućavaju izvršavanje različitih tipova naredbi
 - ne odnose se na neki konkretni objekt baze podataka, već na određenu operaciju ili klasu operacija nad tipom objekta
 - neke od sistemskih dozvola su: CREATE USER; ALTER USER; DROP USER; CREATE SESSION; CREATE ANY INDEX; ALTER ANY INDEX; DROP ANY INDEX; CREATE TABLE; CREATE ANY TABLE; ALTER ANY TABLE; DELETE ANY TABLE; DROP ANY TABLE; INSERT ANY TABLE; SELECT ANY TABLE; UPDATE ANY TABLE; CREATE VIEW; CREATE ANY VIEW; DROP ANY VIEW; CREATE PROCEDURE; CREATE ANY PROCEDURE; EXECUTE ANY PROCEDURE ...

```
GRANT CREATE SESSION TO u1;
```

- korisniku *u1* dozvoljeno je uspostavljanje SQL-sjednice

- dozvole nad objektima

- omogućavaju obavljanje određenih operacija na određenom objektu baze podataka
 - npr. SELECT, UPDATE, INSERT i DELETE operacije na tablicama, ALTER, REFERENCES, INDEX i ALL na tablicama, EXECUTE na pohranjenim procedurama

SQL Server: kategorije dozvola

- ♦ administrator korisniku treba omogućiti pristup instanci SQL Server poslužitelja (CREATE LOGIN naredbom) te pristup bazi podataka (CREATE USER naredbom), npr:

```
CREATE LOGIN loginNameU1 WITH PASSWORD = 'ABCxyz' MUST_CHANGE;
USE stuSluBaza;
CREATE USER u1 FOR LOGIN loginNameU1;
```

- dozvole se mogu svrstati u dvije općenite kategorije:
 - dozvole obavljanja naredbi – upravljaju obavljanjem naredbi kao što su CREATE TABLE, CREATE VIEW, CREATE FUNCTION, CREATE PROCEDURE, itd.
 - dozvole nad objektima baze podataka – upravljaju obavljanjem operacije na postojećem objektu baze podataka, npr. DELETE, INSERT, SELECT, UPDATE, EXECUTE, REFERENCES ...

SQL Server: negativne dozvole (tj. zabrane)

- **SQL Server**: osim dodjeljivanja dozvole pristupa (GRANT), omogućeno je postavljanje zabrane, odnosno dodjeljivanje negativne dozvole (DENY)
- REVOKE ukida dozvole dodijeljene GRANT i DENY naredbama

```
DENY CREATE TABLE TO u1;
```

- korisniku u_1 zabranjuje se izvođenje CREATE TABLE naredbe

```
REVOKE CREATE TABLE FROM u1;
```

- korisniku u_1 ukida se zabrana izvođenja CREATE TABLE naredbe

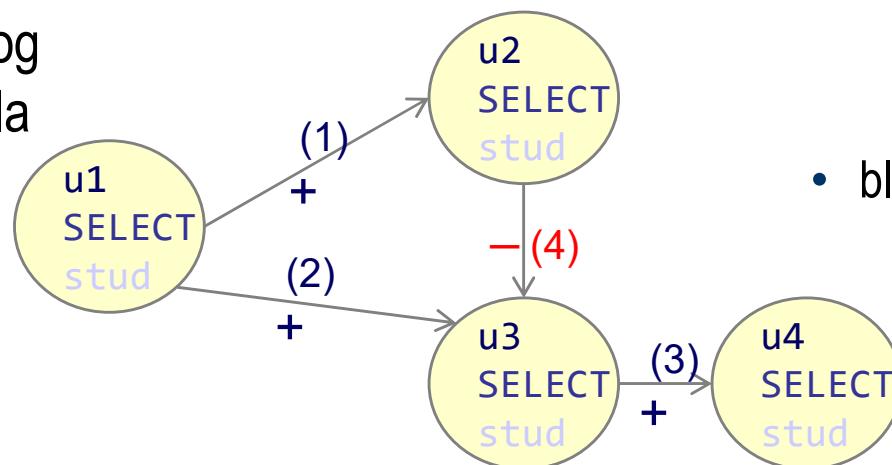
```
DENY SELECT (jmbg) ON student TO u1;
```

- korisniku u_1 zabranjuje se izvođenje SELECT naredbe nad atributom $student.jmbg$

```
REVOKE SELECT (jmbg) ON student TO u1;
```

- korisniku u_1 ukida se zabrana izvođenja SELECT naredbe nad atributom $student.jmbg$

- konflikti koji se pojavljuju zbog postojanja negativnih dozvola razrješavaju se u skladu s politikom: **prioritetne su negativne dozvole**:



- blokirane su (nisu uklonjene):
 - pozitivna dozvola koju je u_1 dodijelio korisniku u_3
 - pozitivna dozvola koju je u_3 dodijelio korisniku u_4

Upravljanje pristupom ovisno o sadržaju i kontekstu

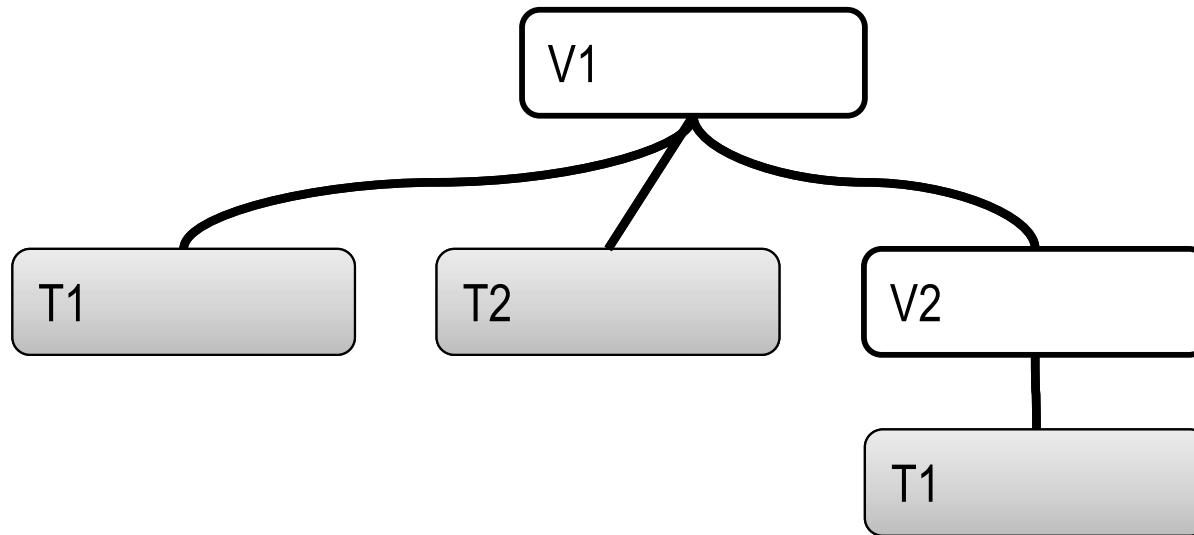
- ◆ upravljanje pristupom ovisno o **sadržaju** (*content-dependent, value-dependent, data-dependent*)
 - pristup objektu na temelju sadržaja jedne ili više njegovih komponenata
- ◆ upravljanje pristupom ovisno o **kontekstu** (*context-dependent, system-dependent*)
 - pristup objektu ovisi o trenutnom kontekstu - u obzir uzima predikate sustava (vrijeme, lokacija ...), trenutnog korisnika

Upravljanje pristupom ovisno o sadržaju i kontekstu

- ◆ dva načina implementacije upravljanja pristupom ovisnog o sadržaju i kontekstu:
 - **definiranje virtualnih tablica** koje odabiru objekt čiji sadržaj zadovoljava dani uvjet te dodjeljivanje dozvola na virtualne tablice, umjesto na temeljne tablice
 - podržano u svim komercijalnim SUBP- ovima
 - **povezivanje predikata** (ili logičke kombinacije predikata) s autorizacijama
 - predikat izražava uvjet nad sadržajem objekta koji mora biti zadovoljen kako bi pristup bio dozvoljen
 - podržan u Oracle SUBP-u

Virtualne tablice

- ◆ Tablica kojoj su shema i sadržaj definirani izrazom relacijske algebre čiji su operandi temeljne ili virtualne tablice.
 - u praksi, shema i sadržaj virtualne tablice opisuju se u obliku SQL upita



- ◆ Sadržaj virtualne tablice dinamički se određuje u trenutku obavljanja operacije nad virtualnom tablicom: ovisi o trenutačnom stanju temeljnih tablica

Virtualna tablica (primjer)

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW prosjek (sifPred  
, prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

projek

sifPred	prosOcj
?	

- Tek u trenutku obavljanja upita, SUBP dinamički određuje sadržaj virtualne tablice *projek*

```
SELECT MIN(prosOcj) AS minPros  
, MAX(prosOcj) AS maksPros  
FROM prosjek;
```

projek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

minPros	maksPros
3.00	4.00

Primjer (nastavak)

- Sadržaj virtualne tablice se ponovno određuje pri izvršavanju svakog upita koji koristi tu virtualnu tablicu

```
INSERT INTO polozeniIspit VALUES(102, 1003, 2);
```

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3
102	1003	2

```
SELECT MIN(prosOcj) AS minPros  
      , MAX(prosOcj) AS maksPros  
FROM prosjek;
```

projek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	2.67

minPros	maksPros
2.67	4.00

Svojstva virtualne tablice

- ◆ Obavljanjem naredbe CREATE VIEW u rječnik podataka se pohranjuje samo definicija virtualne tablice
 - sadržaj virtualne tablice se određuje tek za vrijeme izvršavanja upita koji koristi virtualnu tablicu
 - odnosno, sadržaj virtualne tablice uvijek odražava sadržaj temeljnih tablica u trenutku izvršavanja upita u kojem se virtualna tablica koristi
- ◆ virtualne tablice se u upitima mogu koristiti na svim mjestima gdje se mogu koristiti temeljne tablice
 - između ostalog i za kreiranje novih virtualnih tablica
- ◆ definicija virtualne tablice je trajno pohranjena u bazi podataka
- ◆ virtualna tablica je u dosegu ("vidljiva je") u svim SQL-sjednicama

Atributi virtualne tablice

- ◆ Ako se nazivi atributa u definiciji virtualne tablice ne navedu, nazivi atributa virtualne tablice određeni su nazivima atributa u SELECT naredbi kojom se definira sadržaj virtualne tablice
- ◆ tipovi podataka za atribute virtualne tablice proizlaze iz tipova podataka atributa temeljnih tablica koje se koriste u definiciji virtualne tablice

osoba

mbr	ime	prez	pbrStan
101	Ana	Kolar	23000
102	Tomo	Novak	21000
103	Tea	Ban	23000

```
CREATE VIEW zadrani1 AS
    SELECT mbr, ime, prez
        FROM osoba
       WHERE pbrStan = 23000;
SELECT * FROM zadrani1;
```

```
CREATE VIEW zadrani2 (matBr
                      , imeSt
                      , prezSt) AS
    SELECT mbr, ime, prez
        FROM osoba
       WHERE pbrStan = 23000;
SELECT * FROM zadrani2;
```

mbr	ime	prez
101	Ana	Kolar
103	Tea	Ban

matBr	imeSt	prezSt
101	Ana	Kolar
103	Tea	Ban

Atributi virtualne tablice

- ◆ Ako se u listi za selekciju pri definiciji virtualne tablice koriste izrazi, nazine atributa virtualne tablice treba eksplicitno navesti

polozeniIspit		
mbr	sifPred	ocj
100	1001	2
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW projek (sifPred  
, prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

ispravno

```
CREATE VIEW projek AS  
SELECT sifPred  
, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

neispravno

Dopušteno, ali **besmisleno** u PostgreSQLu,
atribut će se zvati „?column?”. Ako se navedu
dva izraza, onda dolazi do greške.

Materijalizirane virtualne tablice

- Materijalizirane virtualne tablice: SUBP fizički pohranjuje sadržaj virtualne tablice. Kada se promijeni sadržaj neke od temeljnih tablica pomoću kojih je virtualna tablica definirana, SUBP automatski mijenja i sadržaj materijalizirane virtualne tablice
 - prednost: virtualne tablice koje se vrlo često koriste, a čiji se sadržaj određuje složenim upitim, ne moraju se svaki puta kada neki korisnik koristi tu virtualnu tablicu ponovno izračunavati
 - nedostatak: ako se temeljne tablice pomoću kojih je virtualna tablica definirana često mijenjaju, pri svakoj izmjeni temeljnih tablica troši se dodatno vrijeme radi izmjene sadržaja virtualne tablice.
- Podržavaju rijetki: Oracle, SQL Server (*Indexed views*)
- PostgreSQL ne održava automatski podatke u materijaliziranoj virtualnoj tablici ažurnim, tj. potrebno je ručno osvježiti podatke

Implementacija virtualnih tablica

- ◆ Kako sustavi za upravljanje bazama podataka izvršavaju upite koji sadrže virtualne tablice (ako se ne radi o materijaliziranim virtualnim tablicama)?

Modifikacijom upita

- SUBP ugrađuje elemente definicije virtualne tablice u originalni SQL upit koji koristi virtualnu tablicu - umjesto originalnog SQL upita izvršava se modificirani SQL upit

Izvršavanje modifikacijom upita

- Primjer:

ispit		
mbr	predmet	ocj
100	Elektronika	3
100	Fizika	2
101	Elektronika	5
101	Fizika	2
102	Fizika	1
103	Fizika	5

stud			
mbr	ime	prez	pbrStan
100	Ivan	Kolar	52100
101	Ana	Horvat	42230
102	Jura	Novak	52100
103	Ana	Ban	52100

mjesto	
pbr	nazMjesto
42000	Varaždin
52100	Pula
42230	Ludbreg

studenti koji su položili predmet Fizika

```
CREATE VIEW polFiz AS
    SELECT stud.* , ocj
        FROM ispit, stud
       WHERE ispit.mbr = stud.mbr
         AND predmet = 'Fizika'
         AND ocj > 1;
```

korisnik obavlja:

```
SELECT * FROM polFiz;
```

↓ SUBP modificira upit

```
SELECT stud.* , ocj
    FROM ispit, stud
   WHERE ispit.mbr = stud.mbr
     AND predmet = 'Fizika'
     AND ocj > 1;
```



mbr	ime	prez	pbrStan	ocj
100	Ivan	Kolar	52100	2
101	Ana	Horvat	42230	2
103	Ana	Ban	52100	5

Primjer (nastavak)

- Ispisati prezime, ime i dobivenu ocjenu iz Fizike za studente koji su položili Fiziku, a stanuju u Puli

korisnik obavlja:

```
SELECT polFiz.prez, polFiz.ime, polFiz.ocj
  FROM polFiz, mjesto
 WHERE polFiz.pbrStan = mjesto.pbr
   AND nazMjesto = 'Pula';
```

```
CREATE VIEW polFiz AS
  SELECT stud.* , ocj
    FROM ispit, stud
   WHERE ispit.mbr = stud.mbr
     AND predmet = 'Fizika'
     AND ocj > 1;
```



SUBP modificira upit

```
SELECT stud.prez, stud.ime, ispit.ocj
  FROM ispit, stud, mjesto
 WHERE ispit.mbr = stud.mbr
   AND predmet = 'Fizika'
   AND ocj > 1
   AND stud.pbrStan = mjesto.pbr
   AND nazMjesto = 'Pula';
```



prez	ime	ocj
Kolar	Ivan	2
Ban	Ana	5

Virtualna tablica: INSERT, UPDATE, DELETE

- virtualne tablice se također mogu koristiti u naredbama INSERT, UPDATE i DELETE

```
CREATE VIEW splitStud AS  
    SELECT mbr, ime, prez, pbrStan  
        FROM stud  
    WHERE pbrStan = 21000;
```

stud			
mbr	ime	prez	pbrStan
100	Ivan	Kolar	31000
101	Ana	Horvat	21000

```
INSERT INTO splitStud  
VALUES (102, 'Jure', 'Novak', 21000);  
  
SELECT * FROM splitStud;
```



mbr	ime	prez	pbrStan
101	Ana	Horvat	21000
102	Jure	Novak	21000

```
INSERT INTO splitStud  
VALUES (103, 'Tea', 'Ban', 10000);  
  
SELECT * FROM splitStud;
```



mbr	ime	prez	pbrStan
101	Ana	Horvat	21000
102	Jure	Novak	21000

n-torka jest unesena u temeljnu tablicu,
ali se "ne vidi" u virtualnoj tablici

```
SELECT * FROM stud;
```



mbr	ime	prez	pbrStan
100	Ivan	Kolar	31000
101	Ana	Horvat	21000
102	Jure	Novak	21000
103	Tea	Ban	10000

Virtualna tablica: INSERT, UPDATE, DELETE

- ◆ SUBP ne može promijeniti "sadržaj virtualne tablice" - umjesto toga mora promijeniti sadržaj temeljnih tablica koje se koriste u definiciji te virtualne tablice

ispit

mbr	predmet	ocj
100	Elektronika	1
100	Fizika	5
101	Elektronika	1
101	Fizika	3

```
CREATE VIEW prosli AS  
SELECT * FROM ispit  
WHERE ocj > 1;
```

```
CREATE VIEW pali AS  
SELECT * FROM ispit  
WHERE ocj = 1;
```

korisnik
obavlja:

```
UPDATE prosli SET ocj = 4  
WHERE mbr = 100  
AND predmet = 'Fizika';
```



SUBP modificira upit

```
UPDATE ispit SET ocj = 4  
WHERE ocj > 1  
AND mbr = 100  
AND predmet = 'Fizika';
```

Virtualna tablica: problem migrirajućih n-torki

- ♦ n-torka se pojavljuje u virtualnoj tablici onda kada zadovoljava uvjet iz definicije virtualne tablice
 - n-torka unesena u virtualnu tablicu ili izmijenjena u virtualnoj tablici može "nestati" iz te virtualne tablice (i eventualno se "pojaviti" u nekoj drugoj virtualnoj tablici)

ispit

	mbr	predmet	ocj
t ₁	100	Elektronika	1
t ₂	100	Fizika	5
t ₃	101	Elektronika	1
t ₄	101	Fizika	3

```
CREATE VIEW prosli AS  
SELECT * FROM ispit  
WHERE ocj > 1;
```

```
CREATE VIEW pali AS  
SELECT * FROM ispit  
WHERE ocj = 1;
```

korisnik
obavlja:

```
UPDATE prosli SET ocj = 1  
WHERE mbr = 100  
AND predmet = 'Fizika';  
INSERT INTO prosli  
VALUES (102, 'Elektronika', 1);
```

- n-torka t₂ je "nestala" iz *prosli* i "pojavila" se u *pali*
- nova n-torka <102, Elektronika, 1> unesena preko *prosli* se "pojavila" u *pali*

Virtualna tablica: problem migrirajućih n-torki

- ◆ **Rješenje:** virtualne tablice koje se koriste u naredbama koje mijenjaju podatke obavezno se kreiraju uz opciju **WITH CHECK OPTION**
 - SUBP tada ne dopušta izmjenu ili unos n-torke putem virtualne tablice ukoliko n-torka nakon obavljanja operacije više ne bi pripadala virtualnoj tablici putem koje je izmijenjena ili unesena

ispit	mbr	predmet	ocj
	100	Elektronika	1
	100	Fizika	5
	101	Elektronika	1
	101	Fizika	3

```
CREATE VIEW prosli AS  
SELECT * FROM ispit  
WHERE ocj > 1  
WITH CHECK OPTION;
```

```
CREATE VIEW pali AS  
SELECT * FROM ispit  
WHERE ocj = 1  
WITH CHECK OPTION;
```

```
UPDATE prosli SET ocj = 1  
WHERE mbr = 100  
AND predmet = 'Fizika';  
pogreška
```

```
UPDATE prosli SET ocj = 4  
WHERE mbr = 100  
AND predmet = 'Fizika';  
O.K.
```

```
INSERT INTO prosli pogreška  
VALUES (102, 'Fizika', 1);
```

```
INSERT INTO prosli O.K.  
VALUES (102, 'Fizika', 3);
```

```
INSERT INTO pali pogreška  
VALUES (102, 'Fizika', 3);
```

Neizmjenjive virtualne tablice

- ◆ SUBP ne može promijeniti "sadržaj virtualne tablice" - umjesto toga mora promijeniti sadržaj temeljnih tablica koje se koriste u definiciji te virtualne tablice
 - ako je virtualna tablica definirana tako da SUBP nije u stanju **jednoznačno** odrediti koje operacije treba obaviti na temeljnim tablicama, tada je virtualna tablica **neizmjenjiva** (*non-updatable*)

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW projek (sifPred  
, prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

SELECT * FROM projek;

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

```
UPDATE projek SET prosOcj = 4.5  
WHERE sifPred = 1001;
```

?

```
INSERT INTO projek VALUES (1004, 2.5);
```

?

Izmjenjive virtualne tablice

- ◆ Virtualna tablica je izmjenjiva ako u glavnom SELECT dijelu definicije virtualne tablice koristi atributе iz samo jedne temeljne tablice $r(R)$ i pri tome:
 - ne sadrži eliminaciju duplikata pomoću DISTINCT
 - ne sadrži izraze u listi za selekciju (osim trivijalnih izraza koji sadrže samo ime atributa)
 - izostavljeni atributi ne smiju imati NOT NULL ograničenje ili moraju imati prepostavljenu (*default*) vrijednost
 - ne sadrži spajanje ili uniju
 - ne sadrži grupiranje i postavljanje uvjeta nad grupom (GROUP BY i HAVING)
- ◆ Prethodno navedena ograničenja se ne odnose na eventualne podupite koji se koriste unutar WHERE dijela SELECT naredbe koja se koristi za definiciju virtualne tablice

Primjeri izmjenjivih virtualnih tablica

ispit	matBr	sifPred	datIsp	ocj	sifNas
	1111	1001	29.01.2011	1	101
	1111	1001	05.02.2011	3	101
	1111	1003	28.06.2011	2	303
	1111	1002	27.06.2011	4	202
	1234	1001	29.01.2011	3	202

stud	matBr	prez	ime	pbrSt
	1111	Novak	Ivan	10000
	4444	Ban	Marko	51000
	1234	Kolar	Petar	23000

```
CREATE VIEW poloziliNista AS
    SELECT * FROM stud
    WHERE NOT EXISTS
        (SELECT * FROM ispit
            WHERE ispit.matBr = stud.matBr
            AND ocj > 1)
    WITH CHECK OPTION;
```

```
CREATE VIEW ispitiZadrana1 AS
    SELECT matBr
        , sifPred
        , datIsp
        , ocj
    FROM ispit
    WHERE matBr IN (
        SELECT matBr FROM stud
        WHERE pbrSt = 23000)
    WITH CHECK OPTION;
```

Primjeri neizmjenjivih virtualnih tablica

ispit	matBr	sifPred	datIsp	ocj	sifNas
	1111	1001	29.01.2011	1	1111
	1111	1001	05.02.2011	3	1111
	1111	1003	28.06.2011	2	3333
	1111	1002	27.06.2011	4	2222
	1234	1001	29.01.2011	3	2222

stud	matBr	prez	ime	pbrSt
	1111	Novak	Ivan	10000
	1234	Kolar	Petar	21000

```
CREATE VIEW ispitiZadrana2 AS
    SELECT ispit.matBr
        , sifPred
        , datIsp
        , ocj
    FROM ispit, stud
    WHERE ispit.matBr = stud.matBr
        AND pbrSt = 23000;
```



usporediti s izmjenjivom virtualnom
tablicom **ispitiZadrana1** s
prethodne stranice!

```
CREATE VIEW projek
    (matBr, prosOcj) AS
    SELECT matBr, AVG(ocj)
    FROM ispit
    GROUP BY matBr;
```

```
CREATE VIEW stud1 (ime_pres) AS
    SELECT ime || prez
    FROM stud;
```

```
CREATE VIEW poloziliNesto AS
    SELECT DISTINCT matBr
    FROM ispit
    WHERE ocj > 1;
```

Primjena virtualnih tablica u provođenju sigurnosne politike

- ◆ omogućavaju **prikaz samo onih informacija koje su korisniku potrebne:**
 - **zbirne informacije i/ili samo neki atributi tablice i/ili samo neke n-torce iz tablice**
 - korisniku se dodjeljuju ovlasti nad virtualnom tablicom

Primjena virtualnih tablica u kontekstu dozvola

ispit

mbrSt	nazPred	datIsp	ocj
100	Fizika	1.5.2010	3
102	Matematika	7.9.2009	1
102	Matematika	9.2.2010	5
107	Fizika	5.4.2012	4

- ◆ vlasnik tablice ispit je korisnik horvat
- ◆ korisniku novak treba omogućiti pregled samo prosječnih ocjena po predmetima
- ◆ korisniku kolar treba omogućiti pregled, unos, izmjenu i brisanje samo za ispite iz predmeta Fizika

horvat

```
CREATE VIEW prosjek (nazPred, prosOcj) AS
    SELECT nazPred, AVG(ocj)
        FROM ispit
    GROUP BY nazPred;
GRANT SELECT ON prosjek TO novak;

CREATE VIEW ispitFizika AS
    SELECT * FROM ispit
    WHERE nazPred = 'Fizika'
    WITH CHECK OPTION;
GRANT SELECT, INSERT, UPDATE, DELETE
    ON ispitFizika TO kolar;
```

zašto je nužno virtualnu tablicu
ispitFizika kreirati uz opciju
WITH CHECK OPTION?

Dodjeljivanje kontekstno ovisnih dozvola

ispit

mbrSt	sifPred	datIsp	ocj
100	100	1.5.2010	3
102	200	7.9.2009	1
102	200	9.2.2010	5
107	300	5.4.2012	4

nast

sifNast	imeN	prezN	userId
1001	Slavko	Kolar	kolar
1002	Ivo	Ban	ban
1003	Ana	Novak	novak

predaje

sifNast	sifPred
1001	100
1001	200
1002	200
1003	200
1003	300

- ◆ vlasnik tablica je korisnik horvat
- ◆ svakom nastavniku (korisnicima kolar, ban, novak) omogućiti pregled i izmjenu ispita samo iz predmeta koje predaju

horvat

LOŠE RJEŠENJE!

```
CREATE VIEW kolarIspiti AS
    SELECT * FROM ispit
    WHERE sifPred IN (
        SELECT sifPred FROM predaje
        WHERE sifNast = 1001) WITH CHECK OPTION;
GRANT SELECT, UPDATE ON kolarIspiti TO kolar;
```

- ponoviti za svakog nastavnika: banIspiti, novakIspiti, ...
- nova virtualna tablica za svakog novog nastavnika (≈ 150 na FER-u)
- svaki nastavnik upit nad tablicom ispit mora pisati na drugačiji način

Dodjeljivanje kontekstno ovisnih dozvola

ispit

mbrSt	sifPred	datIsp	ocj
100	100	1.5.2010	3
102	200	7.9.2009	1
102	200	9.2.2010	5
107	300	5.4.2012	4

nast

sifNast	imeN	prezN	userId
1001	Slavko	Kolar	kolar
1002	Ivo	Ban	ban
1003	Ana	Novak	novak

predaje

sifNast	sifPred
1001	100
1001	200
1002	200
1003	200
1003	300

horvat

ISPRAVNO
RJEŠENJE!

```
CREATE VIEW ispitiZaNastavnike AS
    SELECT * FROM ispit
        WHERE sifPred IN (
            SELECT sifPred FROM predaje, nast
                WHERE predaje.sifNast = nast.sifNast
                    AND userId = CURRENT_USER) WITH CHECK OPTION;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO kolar;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO ban;
GRANT SELECT, UPDATE ON ispitiZaNastavnike TO novak;
```

- "sadržaj" virtualne tablice ovisit će o identifikatoru nastavnika koji je ostvario SQL-sjednicu

Pohranjene procedure/funkcije (1)

- ◆ pohranjena **procedura/funkcija** je potprogram koji je pohranjen u rječniku podataka i koji se izvršava u kontekstu sustava za upravljanje bazama podataka
 - procedura je potprogram koji u pozivajući program ne vraća rezultat
 - funkcija je potprogram koji u pozivajući program vraća rezultat
 - koristi se i termin *pohranjena rutina (stored routine)*
- ◆ može biti implementirana kao:
 - SQL procedura - napisana u SQL-u
 - (napomena: u nastavku se pod pojmom *procedura* misli na *SQL proceduru*)
 - vanjska procedura (*external routine*)
 - napisana u eksternom programskom jeziku (npr. Java, C++)
 - poziva se na isti način kao SQL procedura

Pohranjene procedure/funkcije (2)

- ◆ pohranjena je kao objekt u rječniku baze podataka
- ◆ proizvođači SUBP koriste vlastite inačice jezika za definiranje pohranjenih procedura (standard postoji, ali je rijetko implementiran)
 - PostgreSQL: PL/pgSQL
 - IBM Informix: SPL (Stored Procedure Language)
 - Oracle: PL/SQL (Procedural Language/Structured Query Language)
 - Microsoft SQL Server: Transact-SQL
- ◆ Navedeni jezici proširuju mogućnosti SQL jezika proceduralnim elementima koji se koriste u strukturiranim jezicima (C, Java, ...). Osim SQL naredbi, moguće je korištenje varijabli, naredbi za upravljanje tokom programa (*if, for, while, ...*), naredbi za rukovanje iznimkama (*exception handling*)
...
- ◆ postiže se veća produktivnost programera i smanjuje mogućnost pogreške
 - programski kôd potreban za obavljanje nekog postupka koji čini logičku cjelinu implementira se i testira na samo jednom mjestu

Primjena pohranjenih procedura u provođenju sigurnosne politike

- ◆ **SQL** omogućuje zaštitu podataka od neovlaštene uporabe **na razini objekata** (tablice, atributi, virtualne tablice)
 - korisniku se može ograničiti pristup do pojedinih objekata i vrsta operacije koju nad tim objektima može obaviti (brisanje, izmjena, unos, dohvati)
 - **nije moguće ograničiti način** na koji će korisnik obavljati operacije za koje je dobio dozvolu
- ◆ **pohranjena procedura** omogućuje zaštitu podataka od neovlaštene uporabe **na razini funkcija**
 - korisniku se pridjeli **dozvola za obavljanje definirane procedure**, umjesto dozvole za pristup podacima
 - time je **precizno određen način** na koji korisnik smije obaviti operacije nad podacima
 - primjena dozvola u skladu s poslovnim pravilima ugrađenim u proceduru
 - princip najmanje ovlasti

```
CREATE PROCEDURE prijaviIspit (JMBAG CHAR(12), sifPred INTEGER, datumRok DATE)
    EXECUTE PROCEDURE obaviProvjereUzPrijavu (JMBAG, sifPred, datumRok);
    EXECUTE PROCEDURE odrediRbrIzlaz (JMBAG, sifPred, datumRok) INTO rbrIzlaz;
    INSERT INTO ispit VALUES (JMBAG, sifPred, datumRok, rbrIzlaz);
END PROCEDURE;
```

Dozvole za pohranjene procedure/funkcije

- ◆ SQL naredbe za dodjeljivanje i ukidanje dozvola za izvršavanje procedura

```
GRANT EXECUTE ON {procName | funName}  
    TO {PUBLIC | userList | roleList}  
    [WITH GRANT OPTION]
```

```
REVOKE EXECUTE ON {procName | funName}  
    FROM {PUBLIC | userList | roleList}  
    [ CASCADE | RESTRICT ]
```

Primjer

- Korisnik *novak* je službenik u banci kojem je potrebno omogućiti obavljanje **isključivo** jedne vrste bankovne transakcije: prebacivanje iznosa s jednog na drugi račun

racun

	brRacun	stanje
1001	1250.15	
1002	-300.00	
1003	10.25	

- zadatak se ne može riješiti dodjelom dozvole za obavljanje operacije UPDATE nad tablicom *racun* korisniku *novak* (zašto?)

novak

```
UPDATE racun  
SET stanje = stanje - 60.30  
WHERE brRacun = 1001;
```

[Error] ERROR: permission denied for relation racun

```
CREATE FUNCTION prebaci(sRacunaBr racun.brRacun%TYPE  
, naRacunBr   racun.brRacun%TYPE  
, iznos       racun.stanje%TYPE) RETURNS VOID AS  
...
```

PostgreSQL: PL/pgSQL

```
GRANT EXECUTE ON prebaci TO novak;
```

Mandatno upravljanje pristupom

(Mandatory Access Control – MAC)

Mandatno upravljanje pristupom

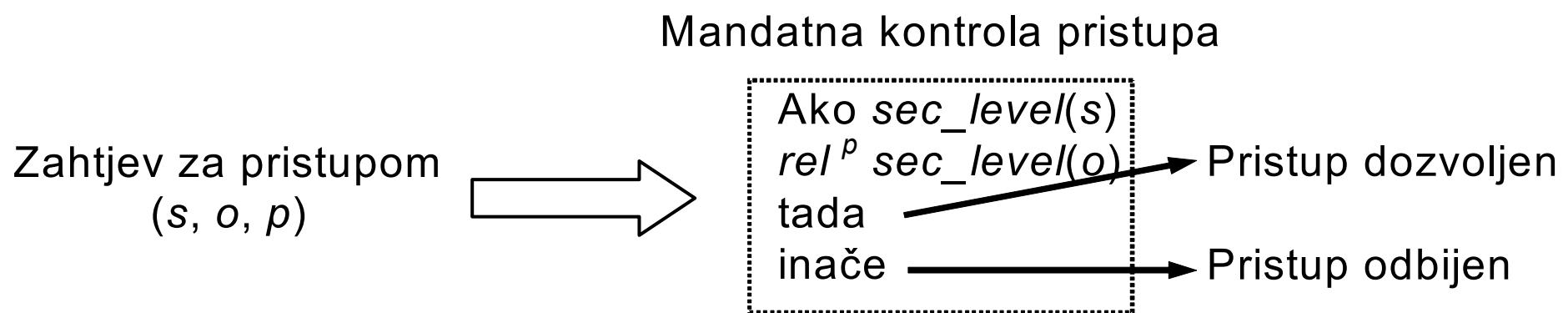
- ◆ *obavezno upravljanje pristupom (upravljanje pristupom na osnovi mandata)* - sigurnosna politika na razini sustava određuje tko ima pravo pristupa, a ne vlasnik objekata
- ◆ primjenjiva u sustavima u kojima se dozvole dodjeljuju ovisno o poziciji korisnika u hijerarhiji neke organizacije (vojska, državna uprava, ...)

“Način ograničavanja pristupa objektima temeljen je na osjetljivosti (predstavljenom oznakom (*label*)) informacija sadržanih u objektu i formalne autorizacije (tj. razine ovlasti) subjekata za informacije takve osjetljivosti.“

[US Department of Defense, Trusted Computer Security Evaluation Criteria (TCSEC), DoD 5200.28-STD, 1985]

Mandatno upravljanje pristupom

- ◆ svaki **objekt** dobiva oznaku **klasifikacijske razine** (*classification level*), npr. **povjerljivo, tajno, ...**
 - odražava osjetljivost informacije sadržane u objektu
 - ◆ svakom **korisniku** dodjeljuje se oznaka **razine ovlasti** (*clearance level*)
-
- ◆ korisnici mogu obavljati operacije nad onim objektima za koje imaju odgovarajuću razinu ovlasti:

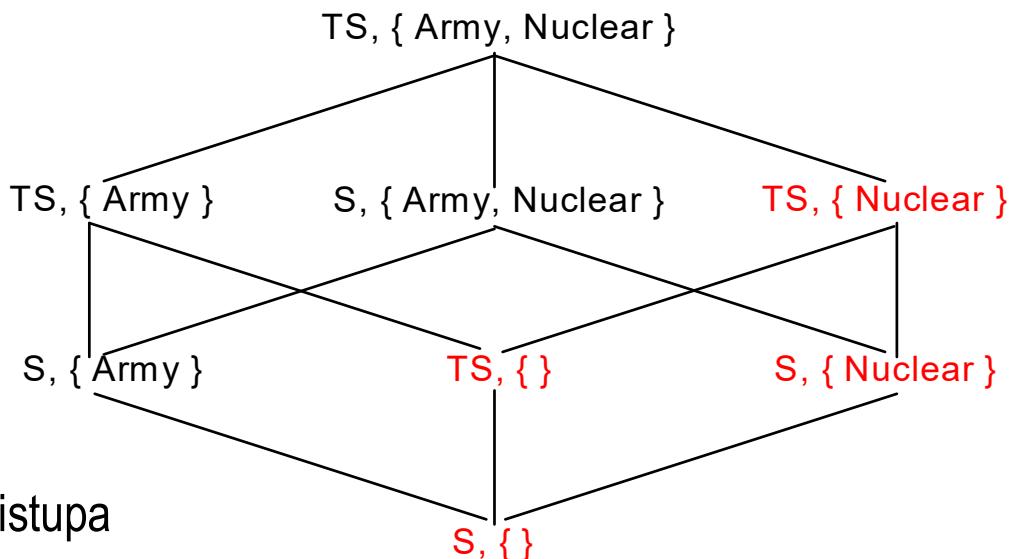


Višerazinska mandatna politika pristupa

- ◆ višerazinska politika pristupa (*multilevel security policy*) - najčešći oblik mandatne politike pristupa
- ◆ administrator sustava svakom objektu i subjektu pridružuje **klasu pristupa**
- ◆ **klasa pristupa** se sastoji od dvije komponente:
 - ◆ **sigurnosna razina** – element hijerarhijski uređenog skupa, npr. **TS > S > C > U** (*Top Secret, Secret, Confidential, Unclassified*)
 - ◆ **skup kategorija** – podskup neuređenog skupa elemenata (**funkcionalna područja ili područja kompetentnosti**), npr: { *Nuclear, Army* }

Višerazinska mandatna politika pristupa

- ◆ klasa pristupa c_1 **dominantna** je klasi pristupa c_2 , tj. $c_1 \geq c_2$ ako je
 - sigurnosna razina klase pristupa $c_1 \geq$ sigurnosne razine klase pristupa c_2
 - kategorije klase pristupa c_1 uključuju one od c_2
- klase pristupa c_1 i c_2 su neusporedive ako niti $c_1 \geq c_2$ niti $c_2 \geq c_1$



- ◆ korisnik se može prijaviti na sustav u svakoj klasi pristupa kojoj je njegova klasa pristupa dominantna
 - ◆ nakon spajanja korisnika na sustav stvara se subjekt u toj klasi pristupa
 - primjer (slika): klase pristupa s kojima se na sustav može spojiti korisnik ovlašten za $(TS, \{ Nuclear \})$:

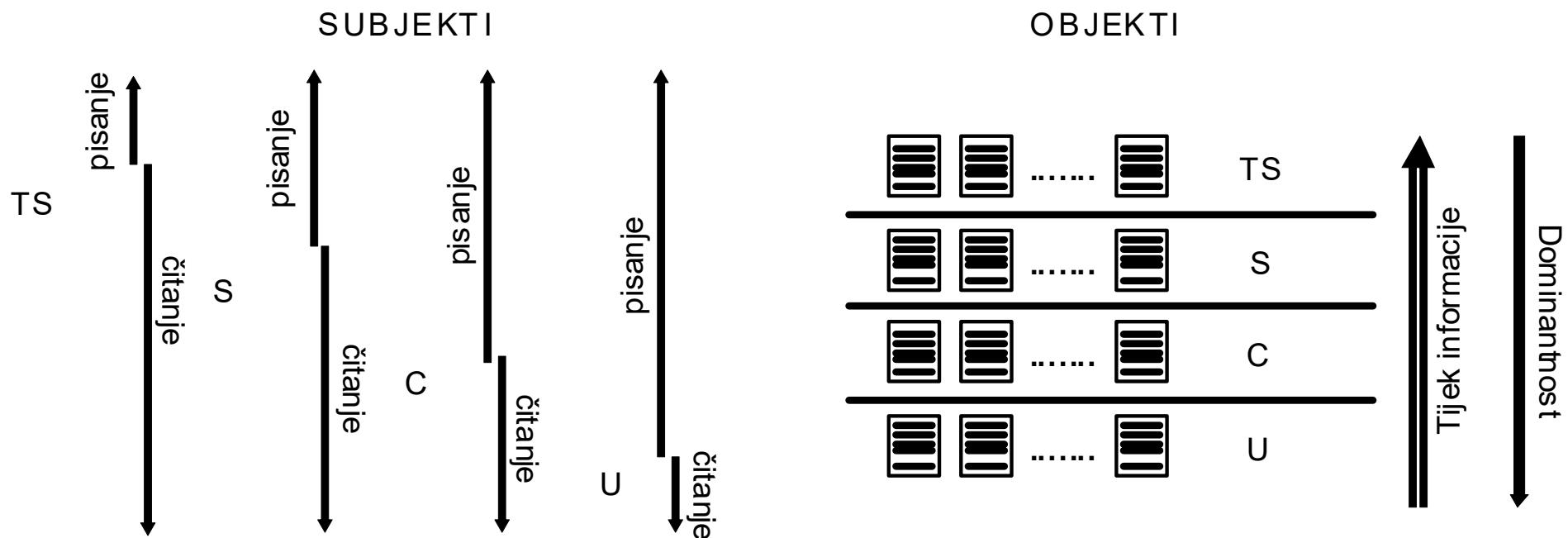
$$C = \{ \text{Nuclear, Army} \}$$
$$TS > S$$

BLP model (1)

- ◆ **Bell – La Padula Security Model (1973)**
- ◆ cilj: spriječiti tijek informacije od subjekata/objekata više razine prema subjektima/objektima nižih (ili neusporedivih) razina

BLP model (2)

- ◆ načela koja osiguravaju očuvanje tajnosti:
 - **simple property (no-read-up)** - subjektu je dozvoljeno čitanje iz objekta samo ako je klasa pristupa subjekta dominantna klasi pristupa objekta (tj. može čitati iz onih objekata kojima je njegova klasa pristupa dominantna)
 - ***-property (star-property, no-write-down)**: subjektu je dozvoljeno pisanje u objekt samo ako je klasa pristupa objekta dominantna klasi pristupa subjekta
 - nije moguće pisati u objekte koje mogu pročitati subjekti s nižom razine - spriječeno propuštanje informacija



BLP model (3)

Primjer: koriste se tri klase pristupa sastavljene od sigurnosnih razina (S, C, U)

- ◆ relacija ***ispit*** ima sigurnosnu razinu S
- ◆ korisnik u_1 ima razinu ovlasti U:
 - na sustav se može spojiti samo na razini U
 - može kreirati i čitati samo objekte razine U
 - kreira relaciju ***xyz*** koja ima sigurnosnu razinu U
- ◆ korisnik u_2 ima razinu ovlasti S:
 - na sustav se može spojiti na razini S, C ili U
 - ako radi kao S subjekt:
 - može čitati iz relacije ***ispit***
 - ne može pisati u relaciju ***xyz*** koja ima razinu U (*no-write-down*)
 - ako radi kao U subjekt:
 - ne može čitati iz relacije ***ispit*** (*no-read-up*)

Zajednička primjena DAC i MAC politike

- ◆ DAC i MAC politike nisu međusobno isključive i mogu biti primijenjene zajedno
 - diskrecijska politika djeluje unutar granica mandatne politike i može samo spriječiti neki pristup koji bi uz primjenu samo MAC politike bio dozvoljen

Mandatna politika pristupa u bazama podataka

- ◆ klasifikacija na razini: relacije; atributa; n-torke (zapis); podatka
- ◆ primjenjuju se osnovni principi MAC-a:
 - ***simple property (no-read-up)***
 - ***strong-star-property*** (umjesto **-property*): korisnik može pisati **isključivo na svojoj razini** (radi sprječavanja uništenja npr. S-podataka od strane U-korisnika)
- ◆ subjekti na različitim razinama imaju različite poglede na relaciju
 - pogled je sastavljen samo od elemenata čijom klasifikacijom dominiraju
 - nerazlikovanje NULL vrijednosti – neprikazani podatak može biti posljedica njegove NULL vrijednosti u bazi podataka **ili posljedica tajnosti (klasifikacije) podatka**
- ◆ implementacija u komercijalnim SUBP-ovima:
 - IBM Informix - Label Based Access Control (LBAC)
 - Oracle Label Security

Upravljanje pristupom temeljeno na ulogama

(Role-Based Access Control - RBAC)

Dodjeljivanje istih dozvola velikom broju korisnika

PROBLEM:

- ◆ svakom nastavniku treba dodijeliti dozvole za
 - pregled, unos i izmjenu podataka o ispitimima za predmete koje predaje, pregled podataka iz relacije *nast*, iz relacije *predaje*, itd.
 - 150 nastavnika \Rightarrow 150 puta treba obaviti niz naredbi za dodjelu dozvola:

```
GRANT SELECT, INSERT, UPDATE ON ispitiZaNastavnike TO kolar;
GRANT SELECT ON predmet TO kolar;
GRANT SELECT ON nast TO kolar;
...
-- ponoviti za svakog od 150 nastavnika
```

- ◆ za svakog novog zaposlenog nastavnika ponoviti postupak
- ◆ kada nastavnik ode u mirovinu, mora se obaviti niz REVOKE naredbi
- ◆ ako se promijene pravila pristupa (npr. odluči se da nastavnici mogu brisati "svoje" ispite), promjena se mora provesti za svakog nastavnika posebno:

```
GRANT DELETE ON ispitiZaNastavnike TO kolar;
-- ponoviti za svakog od 150 nastavnika
```

Upravljanje pristupom temeljeno na ulogama

- ◆ Osnovne postavke
 - podaci vlasništvo poduzeća
 - korisnicima nije dozvoljeno donošenje odluka o pristupu
 - za upravljanje dozvolama zadužen je administrator za sigurnost
 - odluke o pristupu temeljene na ulogama korisnika kao dijela organizacije
 - bolnica: liječnik i medicinska sestra
 - banka: blagajnik i računovodja
 - Korisnici ne mogu svoje ovlasti prosljeđivati drugim korisnicima
liječnik ne smije dopustiti medicinskoj sestri prepisivanje lijekova

Upravljanje pristupom temeljeno na ulogama

- ◆ glavna osobina RBAC modela:
 - svaki pristup podatkovnim objektima i resursima, potreban korisniku za obavljanje njegova zadatka, obavlja se **kroz uloge**
 - uloga predstavlja poslovnu funkciju unutar organizacije
 - ovlasti nad podatkovnim objektima i resursima potrebnim za obavljanje zadatka dodijeljene su ulogama umjesto pojedinim korisnicima
 - korisnik je ovlašten za obavljanje odgovarajuće uloge

RBAC u relacijskim bazama podataka - SQL standard

- ◆ kreiranje uloge:

```
CREATE ROLE roleName [ WITH ADMIN <grantor> ]  
<grantor> ::= CURRENT_USER | CURRENT_ROLE
```

- ◆ uništavanje uloge:

```
DROP ROLE roleName
```

- ◆ dodjeljivanje uloge korisniku:

```
GRANT roleName [ {, roleName } ... ]  
TO <grantee> [ {, <grantee>} ... ]  
[ WITH ADMIN OPTION ][ GRANTED BY <grantor> ]  
  
<grantee> ::= PUBLIC | <roleName> | <userIdentifier>
```

- ◆ ukidanje uloge korisniku:

```
REVOKE [ ADMIN OPTION FOR ] uloga [ {, uloga } ... ]  
FROM <grantee> [ {, <grantee>} ... ]  
[ GRANTED BY <grantor> ] <drop behavior>  
  
<drop behavior> ::= CASCADE | RESTRICT
```

- ◆ aktivacija i deaktivacija uloge:

```
SET ROLE 'roleName'  
SET ROLE NONE
```

- ◆ uloga koja je aktivna u SQL sjednici: CURRENT_ROLE

PostgreSQL uloge

- definira se uloga (*role*), npr. *nastavnik*
- dozvole se, umjesto direktno korisnicima, dodjeljuju novoj ulozi
- uloga može predstavljati jednog ili više korisnika
- uloge se, kao i korisnici, definiraju na razini cijelog SUBP-a

```
CREATE ROLE name [ [ WITH ] option [ . . . ] ]
```

where option can be:

- | CREATEDB | NOCREATEDB
- | CREATEROLE | NOCREATEROLE
- | [ENCRYPTED | UNENCRYPTED] PASSWORD 'password'
- | INHERIT | NOINHERIT
- | LOGIN | NOLOGIN

...

Nalik opcijama CREATE USER naredbe

INHERIT znači da uloga (automatski) nasljeđuje dozvole eventualnih dodatnih uloga koje su joj dodijeljene. INHERIT je preddefinirano ponašanje.

Dodjeljivanje istih dozvola velikom broju korisnika

```
CREATE ROLE nastavnik;  
GRANT SELECT, INSERT, UPDATE ON ispitiZaNastavnike TO nastavnik;  
GRANT SELECT ON nast TO nastavnik;  
GRANT SELECT ON predaje TO nastavnik;  
...
```

- svakom nastavniku, umjesto cijelog niza dozvola, dovoljno je dodijeliti dozvolu za korištenje uloge nastavnik

```
GRANT nastavnik TO kolar;  
GRANT nastavnik TO ban;  
...
```

Dodjeljivanje istih dozvola velikom broju korisnika

- ◆ uloga/korisnik aktivira drugu ulogu uz pomoć naredbe SET ROLE
- ◆ Ako je korisnik kreiran s preddefiniranom INHERIT opcijom (to je slučaj u našem primjeru) nije potrebno aktivirati ulogu jer ionako automatski ima sve njene dozvole.

```
ban: SET ROLE nastavnik;
```

- ako nastavnik s identifikatorom korisnika ban ode u mirovinu

```
REVOKE nastavnik FROM ban;
```

- ako nastavnici trebaju dobiti dozvolu za brisanje "svojih" ispita

```
GRANT DELETE ON ispitiZaNastavnike TO nastavnik;
```

IBM informix: RBAC implementacija

- ◆ svakom nastavniku treba dodijeliti dozvole za pregled, unos i izmjenu podataka o ispitimima za predmete koje predaje, pregled podataka iz relacija *nast*, *predaje*, itd.
 - definira se uloga *nastavnik* i dozvole se dodijele toj ulozi
 - svakom nastavniku se dodijeli dozvola za korištenje uloge *nastavnik*

```
CREATE ROLE nastavnik;
GRANT SELECT, INSERT, UPDATE ON ispitiZaNastavnike TO nastavnik;
GRANT SELECT ON nast TO nastavnik;
GRANT SELECT ON predaje TO nastavnik;    ...
```

```
GRANT nastavnik TO kolar;
GRANT nastavnik TO novak;
...
```

- ako nastavnik s identifikatorom korisnika *novak* ode u mirovinu:
- ako nastavnici trebaju dobiti dozvolu i za brisanje "svojih" ispita:

```
REVOKE nastavnik FROM novak;
```

```
GRANT DELETE ON ispitiZaNastavnike TO nastavnik;
```

- ◆ nakon uspostavljanja SQL-sjednice, korisnik posjeduje sljedeće dozvole:
 - sve dozvole dodijeljene PUBLIC "korisniku"
 - sve dozvole dodijeljene izravno dotičnom korisniku
 - sve dozvole nad objektima kojima je dotični korisnik vlasnik
 - dozvole korisnika na razini baze podataka
 - ako namjerava koristiti i dozvole dodijeljene ulozi *nastavnik*, mora obaviti naredbu:

```
SET ROLE nastavnik;
```

Šifriranje podataka

Šifriranje podataka

- ◆ dodatna razina zaštite ako neovlašteni korisnik uspije doći do podataka iz baze podataka
 - prisluskivanjem komunikacijskih linija ili
 - zaobilaženjem sustava za upravljanje bazama podataka (npr. krađom datoteke ili diska)
- ◆ mjere:
 - prijenos šifriranih podataka (*data-in-transit*) i/ili
 - pohrana šifriranih podataka (*data-at-rest*)
- šifriranje se može koristiti kao zadnji sloj obrane radi zaštite osjetljivih i vrlo povjerljivih informacija
 - nije zamjena za ostale tehnike zaštite podataka, npr. za upravljanje pristupom

Prijenos šifriranih podataka

- ◆ **šifriranje/dešifriranje podataka u prijenosu** događa se u krajnjim točkama komunikacije između klijenta i poslužitelja
 - podaci pohranjeni u bazi podataka i podaci koje koristi klijentska aplikacija nisu šifrirani
- ◆ implementacijske mogućnosti šifriranja podataka u prijenosu:
 - specifične mogućnosti određenog SUBP-a (npr. *Oracle Advanced Security*)
 - *Connection-based methods* (npr. korištenje Secure Sockets Layer [SSL])
 - *Secure tunnels* (npr. korištenje Secure Shell [SSH] tunela)
 - mogućnosti koje podržava operacijski sustav (npr. IPSec)
- ◆ u svim, osim u prvoj kategoriji, **prijenos šifriranih podataka temelji se na industrijskim standardima i ne ovisi o proizvođaču baze podataka**
- ◆ većina metoda šifrira cijeli komunikacijski tok

Pohrana šifriranih podataka (1)

- ◆ **šifriranje vrijednosti koje su pohranjene u bazi podataka**
- ◆ šifriranje/dešifriranje moguće je obaviti na razini:
 - **aplikacije**
 - biblioteke za šifriranje/dešifriranje podataka (npr. Java Cryptographic Extensions - JCE)
 - bazi podataka se pristupa s već šifriranim podatkom - transparentno za bazu podataka
 - šifriranje/dešifriranje je možda potrebno obaviti na više mesta
 - npr. pohranjena procedura koristi podatke šifrirane u Java kôdu aplikacije
 - korištenje podataka samo iz aplikacije - ograničeno je korištenje interaktivnog alata za rad s bazom podataka, čak i uz potrebne ovlasti
 - **datotečnog sustava**
 - korištenje mogućnosti naprednog datotečnog sustava za pohranu podataka u šifriranom obliku
 - npr. Windows - Encrypted File System (EFS)
 - šifrirano je sve, a ne samo osjetljivi podaci - lošije performance
 - **sustava za upravljanje bazama podataka**

Pohrana šifriranih podataka (2)

- ◆ **šifriranje/dešifriranje na razini sustava za upravljanje bazama podataka:**
 - ugrađene rutine baza podataka (npr. T-SQL: DB_ENCRYPT i DB_DECRYPT funkcije)
 - proširenja **SUBP-a** (dodatni paketi) (npr. Oracle - DBMS_CRYPTO paket)
- ◆ podaci su neupotrebljivi dok se ne dešifriraju
 - SUBP ne može obaviti učinkovita uspoređivanja vrijednosti i temeljne operacije nad šifratima

Pohrana šifriranih podataka (2)

- ◆ neizbjegjan pad performansi, ovisno o opsegu šifriranja i korištenim algoritmima
 - ispitivanje Database Server Technologies Group: Oracle 9.2.0.1 - uzorak od 1,6 milijuna šifriranih brojeva socijalnog osiguranja
 - SELECT koji vraća sve zapise: uz korištenje DES algoritma - 200 puta sporiji
 - UPDATE zapisa: uz korištenje DES algoritma - četiri puta sporiji, uz triple DES - osam puta sporiji
- ◆ šifrirani podaci zauzimaju više prostora od originalnog teksta
- ◆ smjernice:
 - **šifrirati selektivno - samo iznimno osjetljive informacije**
 - ne šifrirati atribute koji se koriste kao ključevi ili indeksi

Upravljanje ključevima za šifriranje

- kompromitirani ključevi - mogućnost otkrivanja informacije
 - izgubljeni ključevi - gubitak informacija
-
- ◆ generiranje ključeva
 - npr. Oracle funkcija RANDOMBYTES paketa DBMS_CRYPTO
 - ◆ prijenos ključeva
 - prijenos šifriranog ključa od aplikacije do baze podataka
 - ◆ pohrana ključeva
 - u bazi podataka, u operacijskom sustavu
 - alati koji nude cjelovita rješenja vezana uz upravljanje ključem
 - korisnici upravljaju vlastitim ključevima za šifriranje
 - korištenje transparentnog šifriranja baze podataka
 - ◆ promjena ključa
 - obaviti dok se podacima ne pristupa

Transparentno šifriranje podataka

- ◆ **šifriranje/dešifriranje** podataka **obavlja SUBP** prilikom pohrane/dohvata podatka
 - nije potrebno koristiti posebne funkcije
 - transparentno za korisnike baze podataka
 - nije potrebna izmjena aplikacija radi rukovanja šifriranim podacima
- ◆ poslovi **upravljanja ključem** su **automatizirani**
 - korisnik ili aplikacija ne mora upravljati ključem za šifriranje
- ◆ implementirano u nekim sustavima za upravljanje bazama podataka:
 - Oracle - *Transparent Data Encryption* (TDE)
 - SQL Server - *Extensible Key Management* (EKM)

Praćenje rada korisnika

Praćenje rada korisnika (*auditing*)

- ◆ praćenje različitih kategorija pristupa:
 - prijava/odjava za rad s bazom podataka
 - neuspjeli pokušaji prijave
 - obavljanje DDL naredbi
 - pogreške koje dojavljuje sustav za upravljanje bazama podataka
 - promjene definicija pohranjenih procedura i okidača
 - promjene podataka o korisnicima, njihovim dozvolama i ostalih sigurnosnih atributa
 - promjene osjetljivih podataka
 - dohvati osjetljivih podataka
 - izmjene definicija snimanja traga i snimljenih podataka

Praćenje rada korisnika (auditing)

- ◆ evidentirati svaki pristup osjetljivim podacima u posebnoj datoteci za praćenje rada korisnika (*Audit Trail*)
- ◆ tipičan zapis datoteke sadrži sljedeće informacije:
 - SQL naredba koja se izvršava (*statement source*)
 - mjesto s kojeg je upućen zahtjev (terminal, IP adresa računala)
 - identifikator korisnika koji je pokrenuo operaciju
 - datum i vrijeme operacije
 - n-torke, atributi na koje se zahtjev odnosi
 - stara vrijednost n-torke
 - nova vrijednost n-torke
- ◆ sama činjenica da se prati "trag" obavljenih operacija nad podacima, često je dovoljna za sprečavanje zloporabe

Praćenje rada korisnika

- ◆ analizom prikupljenih podataka moguće je sazнати:
 - postoje li odstupanja od sigurnosne politike koju bi trebalo provoditi
 - postoje li aktivnosti "korisnika" za koje nije ovlašten
 - tko je i kako izmijenio podatke
 - što rade ovlašteni korisnici; što rade privilegirani korisnici
 - je li bilo pokušaja napada npr. SQL injekcijom, ubacivanja zlonamjernog programskog koda u pohranjene procedure ...

Selektivno praćenje

- ◆ nužnost **selektivnog** praćenja DML aktivnosti zbog mogućnosti stvaranja goleme količine podataka - praćenje aktivnosti na podskupu tablica baze podataka
- ◆ u slučaju promjena ovlasti i ostalih sigurnosnih atributa nužno implementirati obavještavanje o promjenama u stvarnom vremenu

Implementacija

- mehanizmi sustava za upravljanje baze podataka
 - **okidači** - implementacija vlastitih rješenja praćenja rada korisnika
 - **proširenja funkcionalnosti:** DB2 - praćenje tragova; SQL Server - trace functions; Sybase - native auditing
 - potrebno je izdvajanje informacija i stvaranje izvješća
- **vanjski sustavi** za praćenje rada korisnika (*third-party rješenja*)
 - prikupljanje podataka, izvještavanje i slanje upozorenja
 - podržavaju veći broj sustava za upravljanje bazama podataka (Oracle, SQL Server, Sybase, DB2, IBM Informix ...)
 - Imperva - Database Activity Monitoring
 - DAS-DBAuditor: Database Auditor
 - Ambeo - Activity Tracker, Usage Tracker, NetServer
 - nisu pod nadzorom DBA (princip razdvajanja dužnosti)
- **usporedba shema (snapshots)**
 - periodičko prikupljanje sheme (obično jednom dnevno) i usporedba s prethodnom shemom (*diff*)

Dodatna razmatranja

- ◆ mjesto **pohrane** prikupljenih podataka
 - goleme količine podataka
 - izbjegavati fizički medij na kojem su pohranjeni podaci produksijskog sustava baze podataka
- ◆ mogućnost **arhiviranja** prikupljenih podataka
 - propisi mogu zahtijevati čuvanje prikupljenih podataka više godina
 - postizanje razumnog vremena odziva prilikom analize podataka
 - utvrđivanje dinamike arhiviranja
 - arhiviranje napravljenih izvješća na temelju prikupljenih podataka
- ◆ zaštita od **neovlaštenog pristupa**:
 - prikupljenim podacima tijekom praćenja
 - napravljenim arhivama tijekom prijenosa i na mjestu pohrane

Zaštita i privatnost podataka

- ◆ **Opća uredba o zaštiti podataka EU**
(GDPR - General Data Protection Regulation)
 - Primjenjuje se **od 25.5.2018.**
 - Uredbom se utvrđuju **pravila povezana sa zaštitom pojedinaca u pogledu obrade osobnih podataka** i pravila povezana sa **slobodnim kretanjem osobnih podataka**.

Zaštita i privatnost podataka

- ◆ Opća uredba o zaštiti podataka EU - **obaveze**:
 - Klijent mora moći dati izričit pristanak na korištenje svojih podataka,
 - Mora moći biti obaviješten kada, u kojem obliku (izvorno, anonimizirani, ili pseudo-anonimizirani) i od strane koga se koriste njegovi podaci, za koju namjenu, i koliko dugo će biti pohranjeni;
 - Omogućiti klijentima uvid u njihove osobne podatke i omogućiti ispravak nepravilnosti;
 - Jamčiti da nema prijenosa podataka u zemlji izvan EU-a koji ima nedovoljnu zaštitu podataka;
 - Ispuniti "pravo na zaborav" – obrisati osobne podatke klijenta na njegov zahtjev, ako su ispunjeni propisani uvjeti.
- ◆ Za kršenje odredbi mogu se izreći upravne novčane kazne u iznosu do 20 000 000 EUR, ili u slučaju poduzetnika do 4 % ukupnog godišnjeg prometa na svjetskoj razini za prethodnu finansijsku godinu, ovisno o tome što je veće.

Zaštita i sigurnost informacijskih sustava

Upravljanje sigurnošću

izv. prof. dr. sc. Stjepan Groš

Kako se zaštитiti od incidenata?

- U uvodu smo napravili pregled nekih incidenata.
 - Možemo nabrajati još mnoštvo drugih incidenata
- Incidenti su raznoliki
 - Ovise o prijetnji, onome što organizacija posjeduje, infrastrukturi koju koristi
 - Prilično složena situacija
- PITANJE JE KAKO SE ZAŠTITI?
 - Pretpostavimo da ste dobili zadaću zaštитiti neku organizaciju/tvrtku od napada. Što učiniti? Kako krenuti?

Što organizacija može učiniti?

- Kupiti novi, veći, vatrozid?
 - Kako ga podesiti? I kako postojeći podesiti? Što dopustiti, što ne?
- Svuda instalirati antivirusni programski alat?
 - Koji? Kako znamo da nije onemogućen?
- Instalirati IDS sustave?
- Prikupljati sistemske i operativne zapise (logove)?
 - Koje? Kako? Koliko ih pohranjivati? Što s njima učiniti?
 - ...

Što organizacija može učiniti?

- Sve to, i više, moguće je...
 - Ali... to je isto kao da u rat idemo tako da samo pošaljemo jedinice u smjeru neprijatelja – s jedinom nadom da je to dovoljno...
 - Sigurnost **NIJE SAMO TEHNIČKO PITANJE**
- Zaštita organizacije zahtjeva ciljan, planiran i dugoročan pristup
 - Zaštitom (sigurnošću) organizacije **mora se upravljati**
 - Upravljanje sigurnošću mora biti sastavni dio upravljanja organizacijom!

Upravljanje organizacijom

- U svim organizacijama postoji model upravljanja (engl. governance model)
 - Ovisan o vrsti organizacije i državi
- Model upravljanja podrazumijeva jasne odgovornosti i ovlasti
- O sigurnosti se brine *Voditelj sigurnosti informacijskog sustava*

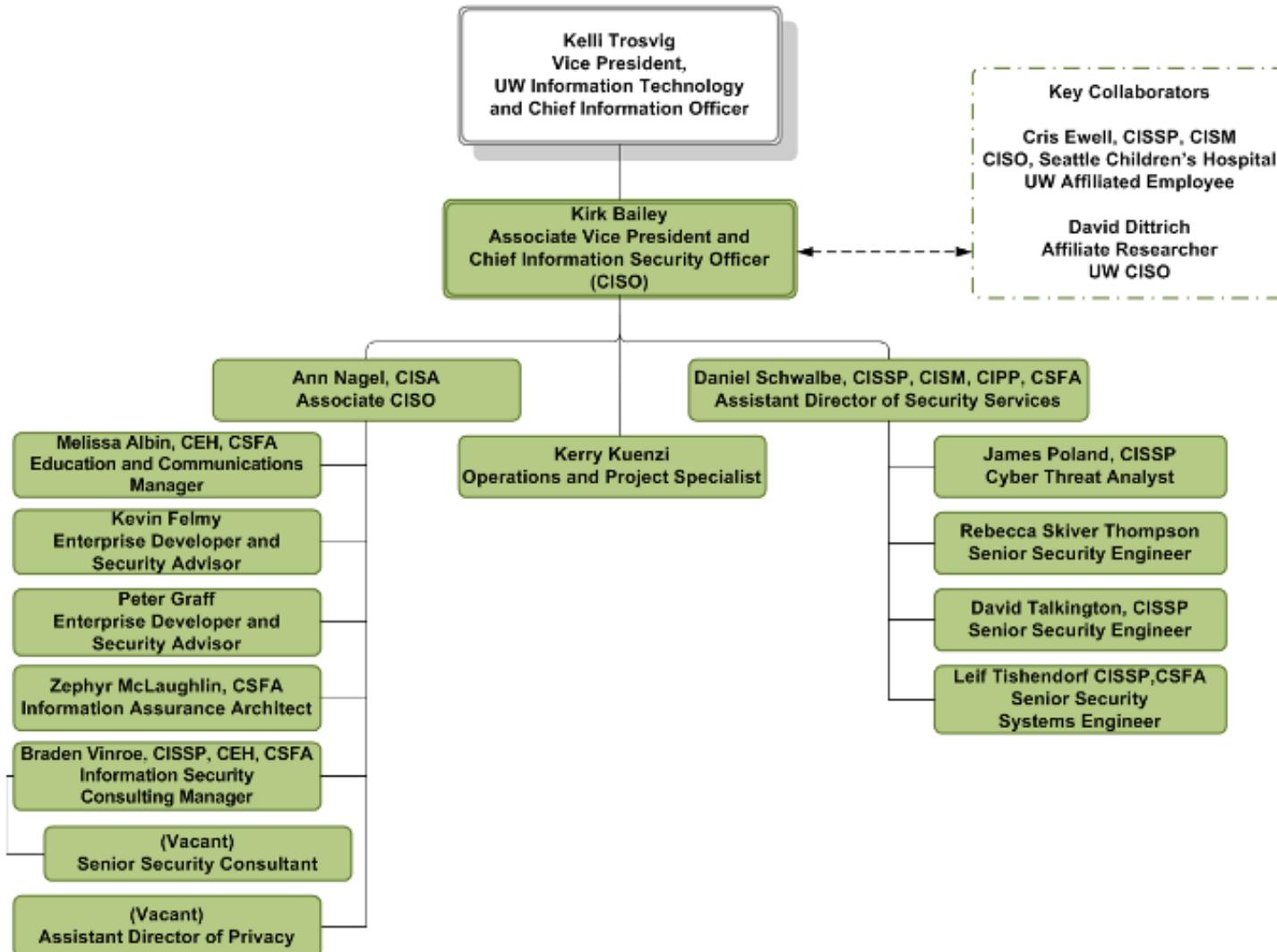
Voditelj sigurnosti informacijskog sustava

- Mora biti osoba koja razumije sigurnost
- Po mogućnosti ne smije se baviti operativnim detaljima
 - Različit od osoba koje se bave taktičkim detaljima
- Alternativna imena
 - Chief Information Security Officer, CISO
 - Chief Information System Security Officer, CISSO
 - I niz drugih imena...

Organizacijska struktura

- CISO je pri vrhu organizacijske strukture
 - Direktno odgovara predsjedniku Uprave (CEO) ili nekog od članova Uprave (CFO, COO, head of legal council)
 - U slučaju potrebe može imati direktni pristup nadzornom odboru
 - Jedna od zahtjevnijih zadaća CISO-a je komunikacija s Upravom
- U slučaju većih organizacija radi se o uredu na čijem čelu se nalazi voditelj ureda
 - Tvrtka ima voditelja sigurnosti (CSO) kojemu CISO odgovara
 - Voditelj sigurnosti zadužen je za cjelokupnu sigurnost (fizičku, informacijsku, prijevare, ...)

University of Washington – Office of CISO



Evolucija uloge voditelja sigurnosti (1)

- Od prvog CISO-a do danas uloga se mijenjala, a opseg širio
 - Posljedica je različite kompetencije osobe na toj poziciji
 - Uzrok je različito okruženje
- Faza 1 (1990-te – 2000)
 - Uvjerenje kako informacije u tvrtci nisu nikome interesantne
 - Ograničena sigurnost, naglasak na lozinkama i kontroli pristupa
 - Pojava Weba, osobna računala mijenjaju velika računala
 - Prvi imenovani CISO, Steve Katz, 1995
 - CISO je bio s tehničkom pozadinom i negdje u IT-ju

Evolucija uloge voditelja sigurnosti (2)

- Faza 2 (2000 – 2004)
 - Pojava legislative za zaštitu privatnosti i podataka
 - HIPAA, Gramm-Leach-Bliley, SOX
 - Prvi put se zahtijevalo da postoji netko zadužen za sigurnost
 - Usklađenost s regulativom
- Faza 3 (2004 – 2008)
 - Problem s ispunjavanjem regulative
 - CISO orijentiran na rizike, naglasak na „mekim vještinama”, još uvijek dio IT-ja

Evolucija uloge voditelja sigurnosti (3)

- Faza 4 (2008 – 2016)
 - Sigurnost svjesna prijetnji, društvene mreže, mobilni uređaji, računarstvo u oblaku
 - Treba li zabraniti FB? Hoće li to utjecati na zapošljavanje? LinkedIn? Što je s uređajima koji se donose u organizaciju?
 - CISO treba marketing, politiku, tehnologiju
 - Daljnje umrežavanje (tableti) zbog kojih su osjetljive informacije izloženije
- Faza 5 (2016 – 2020-te)
 - Privatnost dobija na značenju (GDPR), „outsource”, dobavni lanac
 - CISO svjestan privatnosti i podataka

Svrha CISO-a

- Nadzire i koordinira aktivnosti vezane uz sigurnost informacijskog sustava.
- Inicira primjenu dobrih praksi i prihvaćenih standarda vezanih uz sigurnost informacijskog sustava.
- Ima savjetodavnu ulogu u svezi sa sigurnosti informacijskog sustava.

Zadaće CISO-a – Izrada internih akata (1)

- Razvoj politike sigurnosti informacijskog sustava, standarda, smjernica i ostalih internih akata s ciljem postizanja i održavanja zadovoljavajuće razine sigurnosti
 - Uspostavljanje upravljačkog okvira.
- Temelji akt je **Politika sigurnosti informacijskog sustava**
 - krovni dokument koji definira što za informacijski sustav znači da je siguran.
 - Za sigurnost informacijskog sustava taj dokument je kao ustav koji je temeljni pravni akt države.
 - Temelj za ostale akte („Zakoni”)
- Bitan akt je i pravilnik o radu CISO-a, međutim on ne spada u skup internih akata koji definiraju sigurnost već je organizacijski

Zadaće CISO-a – Izrada internih akata (2)

- Može se postaviti pitanje čemu interni akti, tj. koja korist od njih?
 - Oni predstavljaju administrativne kontrole, nadopuna tehničkim kontrolama
 - Da svi dionici jasno i nedvosmisleno znaju što smiju (ili ne smiju), tko je za što zadužen, što se radi, kada i kako, ...

Zadaće CISO-a – Izrada internih akata (3)

- Vrste internih akata:
 - Politika
 - Piše ju Voditelj sigurnosti informacijskog sustava
 - Definira osnovna načela, smjernice... bez ikakvih konkretnih tehničkih detalja
 - Puna izraza tipa „Trebalo bi ...”, „Mora se...”, „Preporučljivo je...”, „Ako je moguće...”
 - Pravilnik (voditelj sigurnosti/IT)
 - Detaljnije razrađuje politiku, s konkretnijim detaljima no bez ulaska u veliku dubinu
 - Procedura (IT)
 - Detaljna razrada pojedinih dijelova pravilnika, jasno specificira korake

Zadaće CISO-a – Izrada internih akata (4)

- Primjeri internih akata
 - Politika sigurnosti informacijskog sustava
 - Po mogućnosti, što kraći dokument, no ima i vrlo dugih
 - Politika/pravilnik za izradu pričuvnih kopija
 - Politika/pravilnik za vođenje i bilježenje sistemskih i operativnih zapisa
 - Politika/pravilnik korištenja informacijskog sustava
 - Politika/pravilnik upravljanja sigurnosnim rizicima
- **Interni akti su živi dokumenti!**
- Interne akte odobrava i prihvata Uprava

Izazov pisanja politika

- Pisanje politika je složen posao
 - Mora biti efektivna te u skladu s ostalim organizacijskim politikama
- Neke preporuke za pisanje politika
 - Jasno i jednostavno napisana (ne koristiti komplikirane izraze)
 - Ne (pre)dugačka
 - U skladu s primjenjivim zakonima i regulativom
 - Razumna
 - Provediva

Sadržaj politike

- **Pregled** – o čemu je politika
- **Svrha** – zašto je politika potrebna
- **Obuhvat** – što sve politika obuhvaća
- **Kome je namijenjena** – tko je sve obvezan djelovati temeljem politike
- **Politika** – Temeljni dio dokumenta koji navodi što treba biti učinjeno
- **Definicije** – Pojmovi koji se upotrebljavaju u politici
- **Verzioniranje** – Vođenje evidencija o promjenama

Zadaće CISO-a – Provjera provođenja internih akata

- Kontroliranje provođenja politike sigurnosti informacijskog sustava i ostalih internih akata koji se odnose na sve aspekte sigurnosti informacijskog sustava
- Revizija internih sustava i pravilnika
- Po potrebi nadopuna, pojašnjenje ili izmjena pravilnika
 - Sustav se stalno mijenja te je potrebno i interne akte prilagođavati
 - Iskustvom se neke stvari ispostave teške, ili jednostavne, ili (ne)provedive

Zadaće CISO-a – Upravljanje rizicima (1)

- Temeljni alat u radu CISO-a
- Omogućava da se odrede rizici kojima je izložena organizacija
 - Procjena rizika (engl. risk assessment)
 - Potom je moguće napraviti prioritizaciju
 - Na temelju prioriteta odlučuje se što će se učiniti s identificiranim rizicima
 - Prihvati, ovladati (na neki način umanjiti), ili prenijeti na neku treću stranu
 - Uprava organizacije ima konačnu riječ po tom pitanju

Zadaće CISO-a – Upravljanje rizicima (2)

- Vrlo bitan element posla CISO-a koji se često radi na vrlo loš način
- Upravljanje rizicima je **proces** (kao i sigurnost)
 - Mora se uvijek provoditi jer se uvjeti stalno mijenjaju
- Propisan internim aktima

Zadaće CISO-a – Upravljanje incidentima (1)

- Danas nije pitanje AKO se desi incident već KADA će se desiti
- Postupak koji bi također **morao** biti definiran internim aktima
 - Tko/što/gdje/kada/kako, koga se zove, ovlasti i obaveze
 - Uključuje dijelove ili cijelu organizaciju, a CISO (možda i CEO) je koordinator
 - To recimo može ovisiti o incidentu (interni akt!)
 - Definira i što se podrazumijeva pod incidentom
 - Treba biti i uvježban, ili bar na neki način ispitan

Zadaće CISO-a – Upravljanje incidentima (2)

- Cilj upravljanja incidentima
 - Što prije detektirati incident
 - Utvrditi uzroke incidenta, posljedice i nastalu štetu te djelovati na otklanjanju incidenta i umanjivanju štete
 - Uvesti kontrole koje će spriječiti njegovo ponavljanje
- Nisu svi incidenti jednaki, a neki mogu dovesti i do krize u organizaciji

Zadaće CISO-a – Edukacija i osvještavanje

- Upozoravanje na potrebu za izobrazbom
- Davanje smjernica za izobrazbu svih osoba koje se koriste informacijskim sustavom banke, a u svezi sa sigurnosti informacijskog sustava
- Osvještavanje uključuje
 - Objašnjavati važnost sigurnosti organizacije.
 - Informiranje zaposlenika o njihovim ulogama i očekivanjima od uloge u sklopu sigurnosnih funkcija.
 - Davati smjernice u obavljanju pojedinih zadaća vezanih uz sigurnost ili rizike.
 - Edukacija korisnika.

Zadaće CISO-a – Potencijalne teme edukacije

- Politike sigurnosti i interni akti.
- Regulatorni zahtjevi.
- Socijalni inženjering.
- Kontinuitet poslovanja.
- Upravljanje katastrofalnim situacijama.
- Upravljanje sigurnosnim incidentima.
- Klasifikacija podataka, označavanje podataka i odgovarajuće rukovanje.
- Ponašanje zaposlenih.
- Fizička sigurnost.
- ...

Zadaće CISO-a – Metode provođenja edukacije

- Korištenjem sustava za e-učenje
 - Skalabilno
 - Neprilagođeno trenutnoj situaciji
- Direktnim predavanjem
 - Direktni kontakt s ljudima i mogućnosti demonstracija u tijeku predavanja
 - Ne skalira dobro
- Predavanjem putem Interneta (Skype i slično)

Zadaće CISO-a – Izvješćivanje

- CISO mora periodički izvješćivati upravu i nadzornom odboru
 - Odnosno, bilo koga koje je organizacijski odgovoran za svoj rad
 - Najčešće onaj koga CISO izvještava nije stručnjak za sigurnost
- Internim aktom o radu CISO-a ili nekim ekvivalentnim dokumentom propisani su detalji
 - Učestalost izvješćivanja
 - Sadržaj izvješća
 - Kome se izvješće sve podnosi

Zadaće CISO-a – Rad s vanjskim suradnicima

- Vanjski suradnici su
 - Revizori
 - Stručnjaci koji testiraju sigurnosti informacijskog sustava
- Rad s vanjskim suradnicima znači
 - Dogovaranje tipa i opsega poslova
 - Pružanje potrebnih informacija nužnih za provođenje ispitivanja
 - Analiza rezultata
 - Uključivanje rezultata u procjenu rizika
 - Iniciranje aktivnosti na temelju rezultata

Zadaće CISO-a – Analiza sigurnosnih potreba

- Analiziranje sigurnosnih potreba
- Na temelju analize predlaganje
 - Planiranja
 - implementacije
 - testiranja, i
 - nadziranja aktivnosti za poboljšanje sigurnosti informacijskog sustava
- **Planiranje godišnjeg budžeta**
 - Izlaganje upravi
 - Dio planiranih aktivnosti
- **Planiranje i koordiniranje analize isplativosti preporučenih i postojećih sigurnosnih rješenja**

Zadaće CISO-a – Sudjelovanje u bitnim aktivnostima

- Sudjelovanje u planiranje kontinuiteta poslovanja
 - Business Continuity Management, BCM
 - Sudjelovanje u procjeni adekvatnosti DR lokacije
- Klasifikacija informacija
 - Sve informacije u tvrtki moraju biti klasificirane na temelju tajnosti, integriteta i raspoloživosti
 - Izrađuje se pravilnik/politika te registar imovine
 - CISO može voditi navedenih postupak, ili samo sudjelovati u njemu

Zadaće CISO-a – Sudjelovanje u razvoju i održavanju IT-a (1)

- Sudjelovanje u značajnijim fazama u životnom ciklusu informacijskog sustava s aspekta sigurnosti
 - CISO je uključen kao savjetnik
 - Daje mišljenje o predloženim rješenjima
 - Po potrebi vrši analizu
 - Predlaže dodatna rješenja
- Primjeri
 - Odabir novog rješenja za udaljen pristup
 - Razvoj nove aplikacije
 - Arhitektura novog sustava

Zadaće CISO-a – Sudjelovanje u razvoju i održavanju IT-a (2)

- Praćenje raznih upozorenja i novog razvoja u računalnom kriminalu
 - Na temelju toga davanje smjernica IT-ju
 - Ažuriranje procjene rizika
- Procjena prijetnji
 - Analiza resursa, ciljeva, motiva
 - Primjer: Prijetnja DDoS napadom od anonimne skupine

Zadaće CISO-a – Sistemski i operativni zapisi

- Sistemski i operativni zapisi (logovi) vrlo su bitni sa sigurnost informacijskog sustava
 - Omogućavaju rekonstrukciju događaja
 - Omogućavaju detekciju neočekivanih događaja
- Nužno ih je držati na zasebnom mjestu
 - Kako bi se zaštitili od neovlaštenih izmjena
- Usklađeni satovi vrlo bitni za rekonstrukciju (!)
- CISO definira način upravljanja sistemskih i operativnim zapisima
 - Vrši nadzor
 - U manjim organizacijama moguće analizira logove te traži očitovanja

Nadzor sigurnosti

- Današnji trend je da tvrtke sigurnost nadziru u Sigurnosno operativnim centrima
 - Security Operations Center (SOC)
- SOC je centralizirana funkcija unutar organizacije koja korištenjem ljudi, procesa i tehnologije kontinuirano prati i poboljšava sigurnosno stanje organizacije te sprečava, detektira, analizira i odgovara na sigurnosne incidente
 - Nekada se pod tim smatrala fizička lokacija, ali danas ne treba biti
 - Trend je također iznajmljivanja usluge SOC-a
- SOC je evoluirao iz SIEM-a (security incident and event management), koji je evoluirao od centraliziranog bilježenja sistemskih i operativnih zapisa

SOC



Problemi (i izazovi) s kojima se CISO susreće (1)

- Vrlo dinamična okolina koju je teško pratiti
 - Prijetnje van organizacije se brzo mijenjaju i puno ih je
 - **Unutarnje prijetnje**
 - Informacijski sustav je vrlo dinamičan
 - Čak ni IT osoblje se ne pridržava pravila (!)
- Evaluacija rizika raznih tehnologija
 - „Cloud“ rješenja, BYOD, Mobilni uređaji, Specifične aplikacije

Problemi (i izazovi) s kojima se CISO susreće (2)

- Operativni i strateški poslovi
- Vanjski dobavljači
- Koordiniranje aktivnosti s organizacijskom jedinicom informacijske tehnologije i unutarnjom revizijom
 - Ponekad nisu na „istoj valnoj duljini“

Kompetencije CISO-a* (1)

- **Upravljačke vještine (engl. management skills)**
 - Odnosi se na one koje su neophodne da se postigne efektivna nabavka, alokacija, korištenje ljudskih resursa ili fizičkih resursa kako bi se postigao neki cilj
- **Poslovne vještine (engl. business skills)**
 - Povećati vrijednost organizacije i integrirati potrebu za sigurnošću s poslovnim ciljevima tvrtke
- **Stalno usavršavanje (engl. information systems security education)**
- **„Soft skills”**
 - Pisana i govorna komunikacija, efektivan prezentacija i slično.
 - Kritičko razmišljanje, rješavanje problema

* Whitten, Dwayne. "The chief information security officer: an analysis of the skills required for success." *Journal of Computer Information Systems* 48.3 (2008): 15.

Kompetencije CISO-a* (2)

- Vještine i znanja iz sigurnosti informacijskih sustava
- Planiranje oporavka od katastrofičnih događaja (engl. disaster recovery planning)
- Sposobnost istraživanja sigurnosnih incidenata (engl. security breach investigations)

* Whitten, Dwayne. "The chief information security officer: an analysis of the skills required for success." Journal of Computer Information Systems 48.3 (2008): 15.

Kako postati CISO

- Krenuti od dna prema vrhu
- Stjecanje tehničkih kompetencija
- Certifikati, dodatna edukacija
- Stjecanje upravljačkih kompetencija
- CISM, MBA, Specijalistički studij informacijske sigurnosti



Nadzor rada CISO-a

- Postoji definiran sustav nadzora rada CISO-a
 - Sprečavanje potencijalne štete zbog neodgovornog ili zlonamjernog ponašanja
 - Poboljšanje rada CISO-a – povratna petlja
- Tu dužnost obavljaju
 - Revizije
 - Unutarnja i vanjska
 - Osoba/funkcija nadređena voditelju sigurnosti
 - CEO, CIO, CSO, ...

Hvala!



Zaštita i sigurnost informacijskih sustava

Sigurnost programske podrške

prof. dr. sc. Krešimir Fertalj

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Zaštićeno licencijom <http://creativecommons.org/licenses/by-nc-sa/2.5/hr/>



Creative Commons



□ slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo

□ pod sljedećim uvjetima:

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, prerađu možete distribuirati samo pod licencem koja je ista ili slična ovoj.

U slučaju daljnog korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licencije preuzet je s <http://creativecommons.org/>.

Osnovni pojmovi

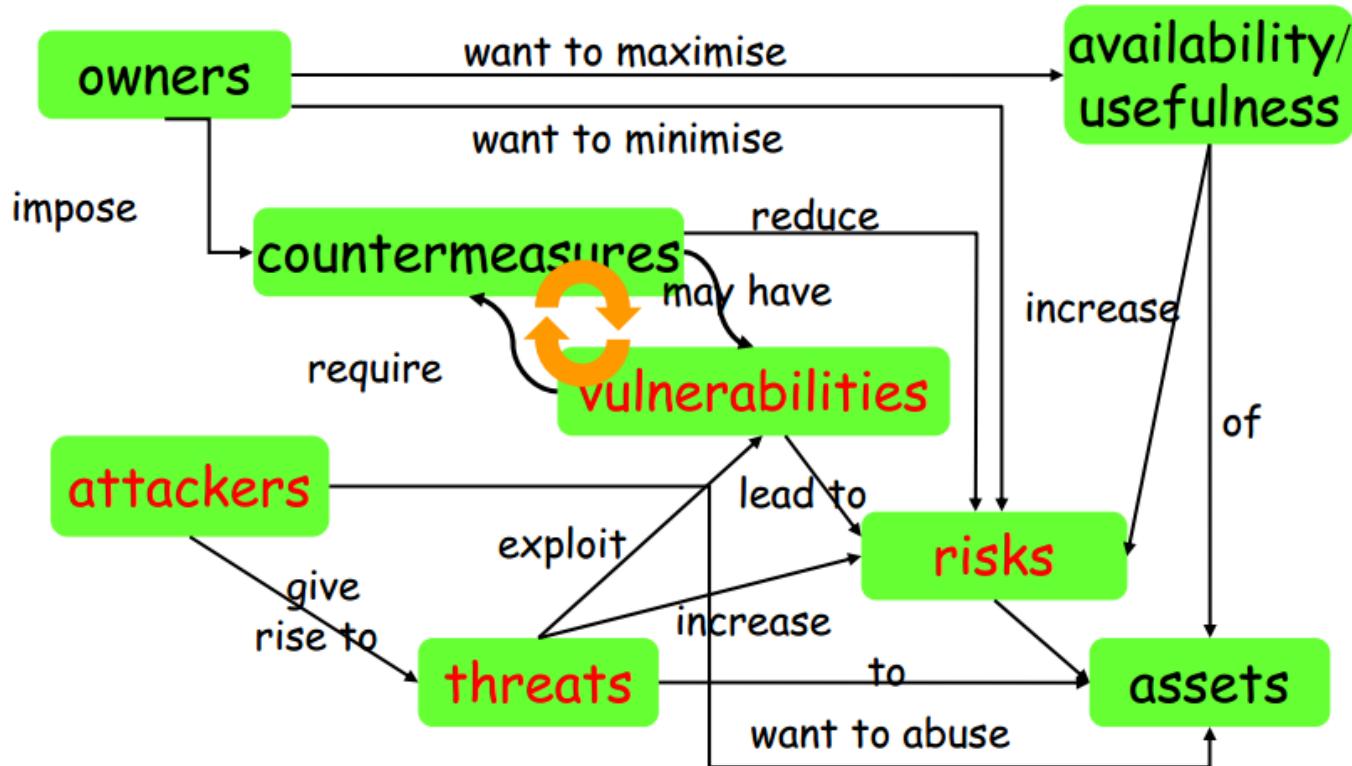
- ◆ Sigurnost programske podrške (software security)
 - Inženjerstvo softvera koji će pri napadu nastaviti ispravno raditi
 - *the science and study of protecting software (including data in software) against unauthorized access, modification, analysis or exploitation*
 - Software security = risk management
 - Management = administrative policies + patch security holes + testing + auditing
- ◆ Sigurnosna programska podrška (security software)
 - Računalni programi i knjižnice za potporu sigurnosti računala ili mreže
 - Antivirusni sw, kriptografski sw, vatrozid, sw za detekciju upada, sigurnosni dijelovi OS, ...
- ◆ **software security ≠ security software**
- ◆ Osiguranje softvera (software assurance)
 - Razina pouzdanosti da softver nema ranjivosti (bilo namjerne ili slučajne)

Sigurnost aplikacija

- ◆ Sigurnost aplikacije (application security)
 - **Mjere poduzete tijekom životnog ciklusa aplikacije** radi prevencije iznimki u odnosu na politiku sigurnosti aplikacije ili sustava uslijed pogrešaka u projektiranju, razvoju, ugradnji, nadogradnji ili održavanju aplikacije.
- ◆ Ključni pojmovi
 - Imovina, **sredstvo** (asset) – resurs
 - npr. podatci u bazi podataka/datoteci ili sistemski resursi
 - **Prijetnja** (threat) – opasnost, negativan učinak
 - **Povredivost, ranjivost** (vulnerability) – slabost koja omogućuje prijetnju
 - tj. koja napadaču dozvoljava smanjenje sigurnosti
 - **Napad** (attack, exploit) – akcija povrede sredstva
 - **Protumjera** (countermeasure) – mjera zaštite i ublažavanja rizika

Koncepti sigurnosti

- ◆ Svako razmatranje sigurnosti treba započeti
 - Inventurom dionika, resursa i prijetnji ...
 - od strane zaposlenika, klijenata, ... kriminalaca



Sigurnost kao softverski problem

- ◆ Kada je sigurnost softverski problem ?
 - ovisi o zahtijevanim promjenama
 - mrežni problem – zahtijeva promjenu mrežnih mehanizama, pr. mrežni protokoli
 - problem OS – zahtijeva promjenu mehanizama OS, pr. politika upravljanja resursima (resource management policy)
 - **softverski problem** – zahtijeva promjenu implementacije ili dizajna (softvera)
- ◆ Povećanje nesigurnosti
 - Povećanjem mrežne povezanosti sve više softvera može biti napadnuto !
 - Web aplikacije i preglednici – najslabija karika i predmet napada
 - Smanjenje razlike između OS, mreže i aplikacija
 - OS-like funkcionalnosti platformi Java i .NET
 - preglednik kao "OS" budućnosti ?

Uzroci problema softverske sigurnosti

- ◆ Glavni uzroci
 - nedostatak svijesti, značaja (awareness)
 - nedostatak znanja
- ◆ Sigurnost kao sekundarna briga
 - primarna je funkcionalnost, servis, udobnost
 - (truli) kompromis u kojem sigurnost gubi ...
- ◆ **Funkcionalnost** – ono što aplikacija radi
- ◆ **Sigurnost** – bavi se onim što aplikacija ne bi smjela raditi

Sigurnosni ciljevi : CIA

- ◆ Confidentiality (povjerljivost, tajnost)
 - uskraćivanje “čitanja” neautoriziranim korisnicima
- ◆ Integrity (integritet, cjelovitost)
 - uskraćivanje promjena neautoriziranim korisnicima
- ◆ Availability (dostupnost)
 - omogućavanje pristupa autoriziranim korisnicima, uskraćivanje ostalima
- ◆ Neporecivost odgovornosti (Non-repudiation for accountability)
 - autorizirani korisnici ne mogu odbiti, negirati, zaobići ugrađene postupke

Realizacija ciljeva: AAAA

- ◆ Autentifikacija (**authentication**)
 - ovjera, utvrđivanje vjerodostojnosti, **provjera autentičnosti**
 - proces identificiranja pojedinca, obično temeljen na korisničkim imenima i lozinkama, zasnovan na ideji da svaki pojedini korisnik ima nešto čime se razlikuje od ostalih korisnika
 - provjera je li korisnik doista onaj kojim se predstavlja
- ◆ Autorizacija (**authorization**)
 - **provjera ovlaštenosti**
 - proces davanja ili odbijanja pristupa (access) resursima
- ◆ Nadzor, praćenje (**auditing**)
 - provjera je li nešto pošlo krivo
- ◆ Djelovanje (**action**)
 - ukoliko jest, poduzeti mjere

Gdje je tu zaštita ?

- ◆ Zaštita protiv napada?
 - *Anti-virus, intrusion detection, firewalls, etc.*
- ◆ Zaštita protiv prijetnji?
 - *Use forensics to find & eliminate*
 - *Mitigate by punishment, if possible*
- ◆ Zaštita protiv ranjivosti?
Engineer secure software!

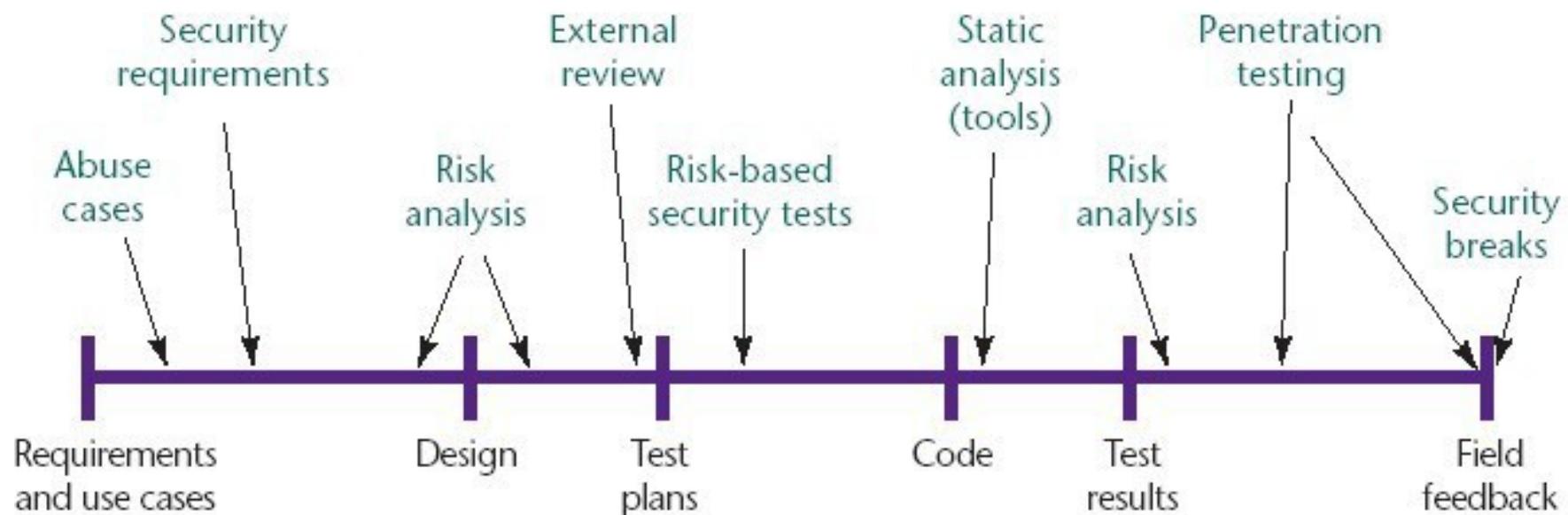
Životni ciklus sigurnog softvera

Secure Software Development Life Cycle

Životni ciklus sigurnog softvera

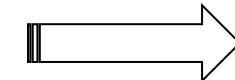
- ◆ Postupci, tehnike i metodologije
 - Sigurnost u životnom ciklusu
 - Inženjerski i projektantski principi
 - Sigurnosne tehnologije
- ◆ Pojednostavljen:

[Source: Gary McGraw, Software security, Security & Privacy Magazine, IEEE, Vol 2, No. 2, pp. 80-83, 2004.]



Modeli procesa životnog ciklusa sigurnog softvera

- ◆ Capability Maturity Models
 - CMMI for Development
- ◆ Team Software Process
 - TSP for Secure Software Development
- ◆ Correctness by Construction
- ◆ Common Criteria
- ◆ Software Assurance Maturity Model
- ◆ Building Security In – Maturity Model
 - Software Security Framework (SSF)
- ◆ **Microsoft's Trustworthy Computing Security Development LC**
 - Životni ciklus razvoja sigurnosti (skraćeno SDL)



SDL aktivnosti - prakse

- ◆ aktivnosti prikazane prema tradicionalnom ciklusu razvoja softvera



- Analiza: sigurnosni zahtjevi, procjena rizika, ...
- Dizajn: modeliranje prijetnji, analiza površine napada, ...
- Implementacija: statička analiza, ...
- Verifikacija: dinamička analiza, *fuzz* testiranje, ...
- Isporuka: plan odgovora na incidente, finalni pregled

Pre-SDL Requirements: Security Training

- ◆ SDL Practice 1: Training Requirements
 - poduka svih članova da bi znali osnove i ostali u trendu
 - tehničari (razvojnici, testeri, ...) – **barem jedan tečaj godišnje**
- ◆ Osnovni tečajevi, s temama (skraćeno)
 - Sigurni dizajn (Secure design)
 - Attack surface reduction, Principle of least privilege, Secure defaults
 - Modeliranje prijetnji (Threat modeling)
 - Overview, Design implications, Coding constraints
 - Sigurno kodiranje (Secure coding)
 - Buffer overruns, Cross-site scripting, SQL injection, Weak cryptography
 - Testiranje sigurnosti (Security testing)
 - Security and functional testing, Risk assessment, Security testing methods
 - Privatnost (Privacy)
 - Types of privacy-sensitive data, design/development/testing best practices
- ◆ Napredni tečajevi - napredni dizajn, arhitektura, trusted GUI, ...

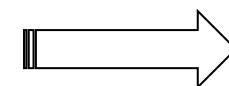
Phase One: Requirements

- ◆ SDL Practice 2: Security Requirements

- rano postavljanje pouzdanih (trustworthiness) zahtjeva
 - pri početnom planiranju
- identifikacija ključnih prekretnica (milestones) i isporuka
- specifikacija minimalnih zahtjeva na sigurnost aplikacija
- uspostava sustava za praćenje (vulnerability/work item tracking system)

- ◆ SDL Practice 3: Quality Gates/Bug Bars

- uspostava minimalno prihvatljivih razina kvalitete sigurnosti i privatnosti
- brana kvalitete (quality gate) – za svaku fazu
 - npr. ukloniti upozorenja kompilatora prije nego se napravi check-in
- prepreka za bugove (bug bar) – primjenjuje se na čitav projekt
 - npr. "bez poznatih kritičnih/važnih ranjivosti u trenutku isporuke"
- tim dokazuje sukladnost kroz Final Security Review (FSR)



Phase One: Requirements (*nastavak*)

- ◆ SDL Practice 4: Security and Privacy Risk Assessment
 - Security risk assessments (SRAs) and privacy risk assessments (PRAs)
- ◆ Procjene
 1. Dijelovi projekta koji zahtijevaju modeliranje prijetnji
 2. Dijelovi projekta koji zahtijevaju pregled dizajna
 3. Dijelovi projekta koji zahtijevaju penetracijsko testiranje
 4. Dodatno testiranje ili zahtjevi radi procjene rizika
 5. Doseg zahtjeva za *fuzz* testiranjem (pogledati praksu 12)
 6. Rangiranje utjecaja na privatnost (Privacy Impact Rating)
- ◆ Rang utjecaja (rizika) na privatnost
 - **P1** : visok – *feature/proizvod/servis* sprema ili prenosi osobne podatke, mijenja postavke ili instalira softver
 - **P2** : srednji – ponašanje koje se odnosi na privatnost je jednokratni, korisnički pokrenut prijenos podataka (npr. klik za odlazak na web)
 - **P3** : nizak – nema instalacije, promjena, prijenosa (kako je prethodno navedeno)

Phase Two: Design

- ◆ SDL Practice 5: Design Requirements
 - što ranije uklanjanje problema sigurnosti i privatnosti
 - izbjegavati "šarafljenje" ("bolting on") sigurnosti na kraju razvoja
- ◆ **razlikovati “secure features” i “security features” !**
 - sigurne mogućnosti – opća funkcionalnost koju treba osigurati (npr. unos, robusnost)
 - sigurnosne mogućnosti – f-nost koja se odnosi na sigurnost (npr. autentifikacija)
- ◆ Specifikacija dizajna treba
 - opisati mogućnosti softvera izravno izložene korisniku
 - opisati kako sigurno ugraditi funkcionalnost
- ◆ provjerava se naspram funkcionalne specifikacije koja
 - točno i potpuno opisuje korištenje mogućnosti
 - opisuje kako sigurno postaviti (deploy) *feature* ili funkciju

Phase Two: Design (nastavak)

- ◆ SDL Practice 6: Attack Surface Reduction

- redukcija rizika smanjenjem prostora za napad
- isključenjem ili restrikcijom pristupa na sistemske resurse
- primjenom principa najmanjeg prava (least privilege)
- uslojavanjem, gdje je moguće

- ◆ SDL Practice 7: Threat Modeling

- gdje postoji rizik sigurnosti
- razmatranje i dokumentiranje posljedica u planiranom operativnom okruženju
- razmatranje sigurnosti pojedinih komponenti ili aplikacije
- **glavna aktivnost dizajna u kojoj sudjeluju**
 - program/projekt menadžeri, razvojnici, testeri

Phase Three: Implementation

- ◆ **SDL Practice 8: Use Approved Tools**
 - tim određuje alate - npr. kompilator/linker opcije, upozorenja
 - savjetnik (advisor) odobrava
 - tim treba ustrajati na zadnjim verzijama dokazanih alata (oprez !)
- ◆ **SDL Practice 9: Deprecate Unsafe Functions**
 - analiza korištenih funkcija i API-ja s obzirom na sigurnost
 - stvaranje liste "zabranjenih" (banned list)
 - označavanje (npr. banned.h, strsafe.h)
 - korištenje odgovarajućih opcija prevoditelja za provjeru ili posebnih alata
 - npr. opcija kompilatora /GS (Buffer Security Check), zaseban alat *StackGuard*
- ◆ **SDL Practice 10: Static Analysis**
 - osigurava inspekciju programskog koda, ali ju ne može zamijeniti !
 - npr. alati *StyleCop*, *CodeSmart*, *Ndepend/JDepend*

Phase Four: Verification

- ◆ SDL Practice 11: Dynamic Program Analysis
 - pogonska (run-time) verifikacija koja utvrđuje da program radi kako je projektiran
 - provjera korupcije memorije, korištenje privilegija, ...
 - npr. *AppVerifier*, *ANTS profiler*, *Rational* ...
- ◆ SDL Practice 12: Fuzz Testing
 - varijanta dinamičke analize kojom se
 - nastoji izazvati zastoj unosom neispravnih ili pseudoslučajnih podataka
- ◆ SDL Practice 13: Threat Model and Attack Surface Review
 - tokom razvoja dolazi do odstupanja od specifikacija
 - ponovni pregled modela prijetnji i mjerjenje površine napada
 - verifikacija promjena u odnosu na specifikacije

Phase Five: Release

- ◆ **SDL Practice 14: Incident Response Plan**
 - plan odziva na incidente definira
 - sustained engineering (SE) tim, ili emergency response plan (ERP) ako nema resursa
 - *on-call* kontakt koji ima autoritet odlučivanja, 24x7
 - plan servisiranja sigurnosti za izvana nabavljenе komponente

- ◆ **SDL Practice 15: Final Security Review (FSR)**
 - promišljena provjera svih sigurnosnih aktivnosti, prije objave
 - nije "penetrate and test" ili aktivnost "ugradimo zanemareno i zaboravljeno" !
 - ishodi:
 - **passed FSR** – svi problemi su uočeni, te uklonjeni ili ublaženi
 - **passed FSR with exceptions** – nerazriješeni se evidentiraju i ispravljaju u narednoj objavi
 - **FSR with escalation** – projekt ne može biti objavljen, radi se plan razrješenja prije objave ili ide menadžmentu na daljnje odlučivanje

Phase Five: Release (*nastavak*)

◆ SDL Practice 16: Release/Archive

- *security advisor* potvrđuje (temeljem FSR i šire) da su zahtjevi zadovoljeni
- zasebno se potvrđuju komponente utjecaja na privatnost **P1 (praksa 4)**
- arhiviranje
 - specifikacija,
 - izvornog koda,
 - kompilata,
 - modela prijetnji,
 - dokumentacije,
 - planova odziva na incidente,
 - uvjeta licenciranja za nabavljene komponente,
 - ...

Opcionalne aktivnosti

- ◆ Nadzor, ručna inspekcija koda (code review)
 - vješti pojedinci ili sigurnosni tim ili savjetnik sigurnosti
 - usmjereni na "kritične" komponente
 - najčešće dijelova koji obrađuju ili pohranjuju osobne podatke
 - također dijelova koji se odnose na šifriranje
- ◆ Penetracijsko testiranje
 - *white box* analiza simuliranjem napada hakera
 - otkrivanje potencijalnih povredivosti uslijed pogreški u kodiranju, pogreški konfiguracije ili drugih slabosti u primjeni
 - u kombinaciji s automatiziranim ili ručnom analizom programskog koda
- ◆ Analiza povredivosti sličnih aplikacija
 - analizom dostupnih informacija na Internetu

RACI Chart – uloge (odgovoran, odobravatelj, savjetnik, informiran)

- ◆ RACI – akronim (Responsible, Accountable, Consulted, Informed)

Tasks	Architect	System Administrator	Developer	Tester	Security Professional
Security Policies		R		I	A
Threat Modeling	A		I	I	R
Security Design Principles	A	I	I		C
Security Architecture	A	C			R
Architecture and Design Review	R				A
Code Development			A		R
Technology Specific Threats			A		R
Code Review			R	I	A
Security Testing	C		I	A	C
Network Security	C	R			A
Host Security	C	A	I		R
Application Security	C	I	A		R
Deployment Review	C	R	I	I	A

Sigurnosni zahtjevi

Security Requirements

Sigurnosni zahtjevi

- ◆ Zahtjevi općenito
 - Funkcionalni zahtjevi – opisuju što softver treba moći raditi
 - Nefunkcionalni zahtjevi – sistemski, kvaliteta, ugovori, standardi, ograničenja
- ◆ **Sigurnosni – nefunkcionalni**
 - Procjene vrijednosti sustava – vrijednost sustava i podataka
 - Ispad košta 50 kn/h, gubitak podataka procjenjuje se na 20 Mkn
 - Zahtjevi za kontrolu pristupa – ograničenje na pristup podacima
 - Voditelji mogu ..., operateri mogu ... ili anon/regi/admin mogu ...
 - Zahtjevi za enkripcijom i autentifikacijom – kako, gdje i kada
 - Zahtjevi za kontrolom virusa
- ◆ Neki mogu zahtijevati funkcionalnost
 - Duljina korisničkog unosa, validacija podataka

Primjeri sigurnosnih zahtjeva

Scenarij	Zahtjev
Aplikacija pohranjuje osjetljive informacije koje trebaštiti radi HIPAA usklađenosti.	Treba koristiti jaku enkripciju za zaštitu osjetljivih podataka.
Aplikacija prenosi osjetljive podatke o korisniku prekopotencijalno nepouzdanih ili nesigurnih mreža.	Komunikacijski kanali moraju uključivati šifriranje kako bi se spriječilo njuškanje a kriptografskom autentifikacijom spriječiti <i>man-in-the-middle</i> napade.
Aplikacija podržava više korisnika s različitim razinama privilegija.	Treba definirati ovlaštenja za akcije na svakoj razini privilegija. Testirati različite razine.
Aplikacija pri unosu podataka koristi SQL	Definirati prevenciju SQL ubrizgavanja
Aplikacija je pisana u C/C++	Kontrolirati veličine međuspremnika, spriječiti modifikaciju formata upisa i preljev cijelih brojeva.
Podaci se prikazuju u HTMLu	Spriječiti XSS napade
Aplikacija zahtjeva praćenje promjena	Definirati funkcije praćenja. Osigurati dnevnik promjena.
Aplikacija koristi kriptografiju.	Treba koristiti sigurni generator pseudoslučajnih brojeva

Izvori zahtjeva

- ◆ Korisnici
- ◆ Sigurnosna implikacija funkcionalnosti
 - Zaštita od SQL ubrizgavanja za aplikacije nad BP
 - Zaštita od XSS ubrizgavanja za web aplikacije
- ◆ Regulatorna sukladnost
 - Zakon o informacijskoj sigurnosti
 - Zakon o zaštiti osobnih podataka
 - Federal Information Security Management Act (FISMA) – resursi vlade SAD
 - Sarbanes-Oxley (Sarbox ili SOX) – javna poduzeća u SAD
 - Health Insurance Portability & Accountability Act (HIPAA) – medicinski podaci

Postupci inženjerstva zahtjeva

- ◆ **SQUARE**
 - Security QUAlity Requirements Engineering Methodology from CMU/SEI
- ◆ **TRIAD**
 - Trustworth Refinement through Intrusion-Aware Design from CMU/SEI
- ◆ ...
- ◆ **SecureUML (UML, OCL), UMLintr, UMLsec**
- ◆ ...
- ◆ **Security Use Cases, Misuse Cases, Abuse Cases (MUCs)**
 - Slučajevi korištenja, zloporabe (nenamjerno) ili zlostavljanja (namjerno)
 - scenariji u kojima sudionik kompromitira sustav

Slučajevi zloporabe

- ◆ Pogled protivnika/napadača
 - Dohvat podataka korisnika
 - Izmjena cijene, ocjene, ...
 - Uskraćivanje usluge
- ◆ Razvoj slučajeva
 - Brainstorming – pretpostavke, obrasci napada, rizici
- ◆ Sigurnosni zahtjevi – generalizirana forma MUCova
 - Anti-zahtjevi – što o**N**E mogući

Primjer SZ

◆ UC1: Prijava u web trgovinu

- Primarni sudionik: Korisnik
- Dionici i interesi: Korisnik – želi kupiti proizvode
- Preduvjeti: Korisnik ima pristup webu
- Posljedice: Korisnik vidi svoj račun, ima mogućnost plaćanja i isporuke
- Sažetak: Korisnik pristupi sustavu putem korisničkog imena i lozinke

◆ MUC1: Njuškanje lozinke

- Primarni sudionik: Napadač
- Dionici i interesi: Napadač – želi dobaviti korisničke vjerodajnice
- Preduvjeti: Napadač ima pristup stroju ili mrežnom putu do sustava
- Posljedice: Napadač je dobavio jedan ili više ispravnih imena / lozinki
- Sažetak: Napadač dobavi i kasnije zlorabi neautorizirani pristup sustavu

Primjer MUC scenarija

◆ Osnovni tok:

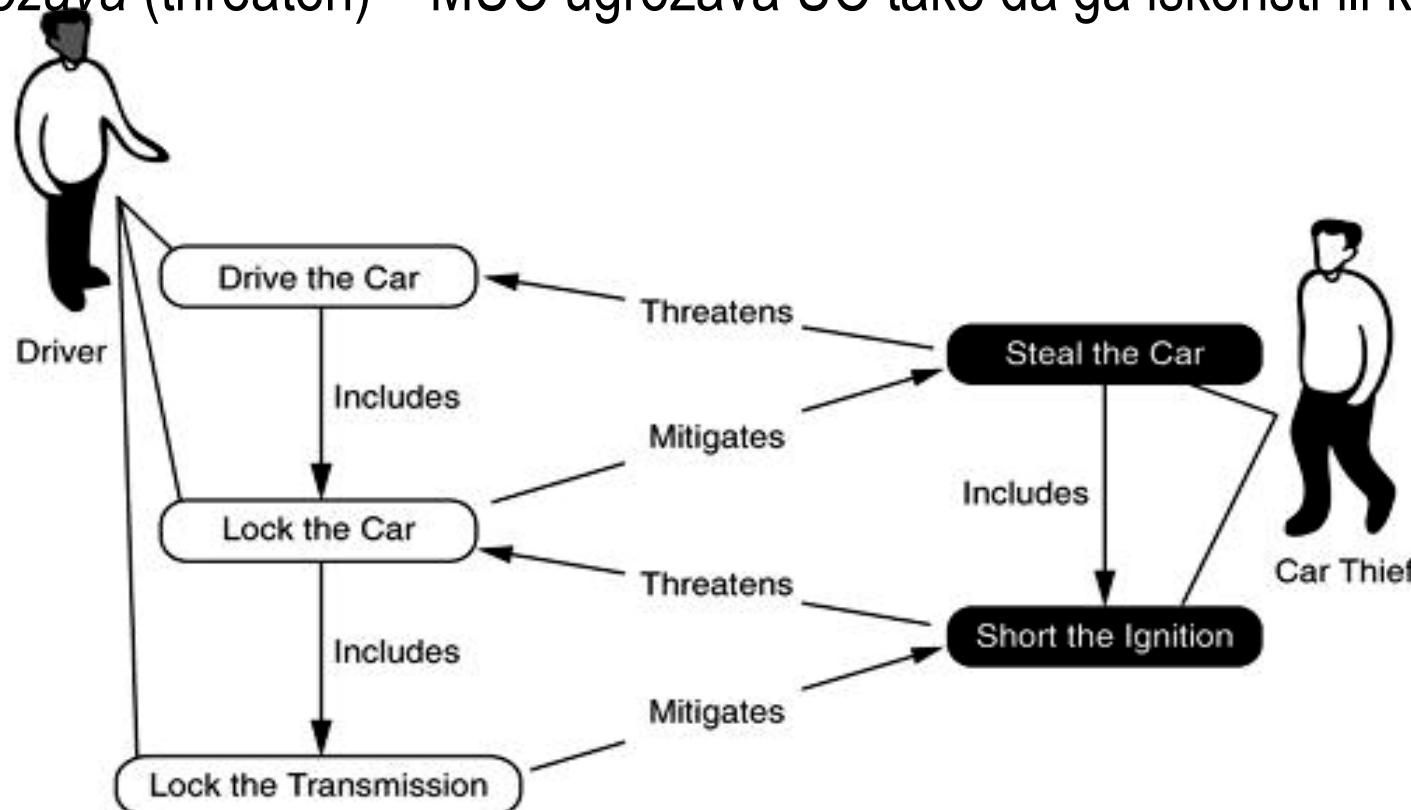
1. Napadač instalira mrežno njuškalo
2. Njuškalo sprema pakete koji sadrže "Logon", "Username", "Password"
3. Napadač čita dnevниke njuškala
4. Napadač nalazi ispravan *login / password*
5. Napadač koristi nađeni *login / password* za pristup sustavu

◆ Alternativni tokovi:

- 1a: Napadač nije na putu između korisnika i sustava
 - 1a1. Napadač koristi *ARP poisoning* ili slično da bi preusmjerio pakete
- 1b: Napadač koristi bežičnu konekciju
 - 1b1. Napadač odlazi na lokaciju korisnika
 - 1b2. Napadač koristi *wifi sniffer* za presretanje prometa

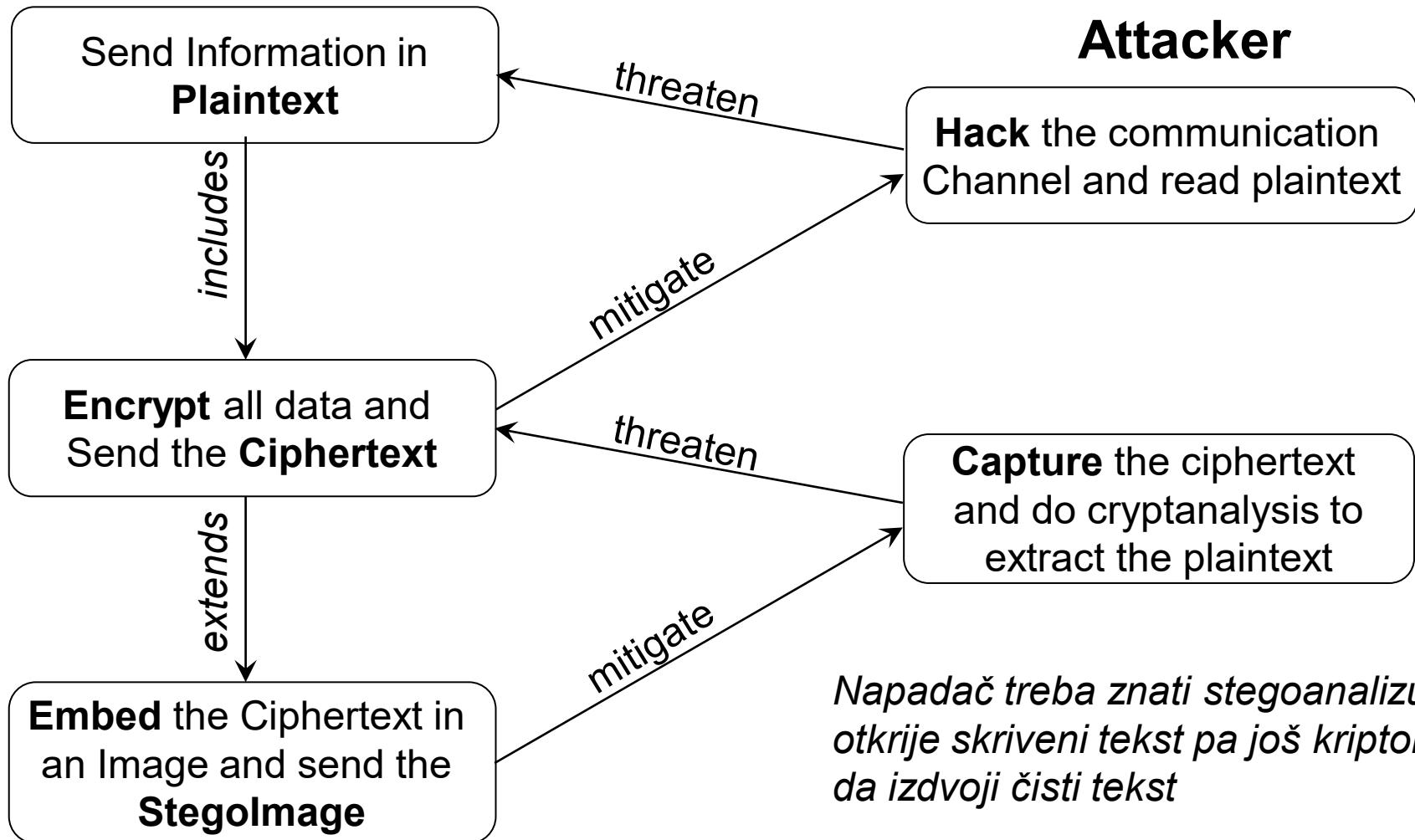
Povezivanje slučajeva zloporabe

- ◆ Proširenje dijagrama slučaja korištenja
 - Ublažava (mitigate) – UC smanjuje priliku da MUC bude uspješan
 - Ugrožava (threaten) – MUC ugrožava UC tako da ga iskoristi ili koči



Primjer: UC-MUC dijagram sigurne komunikacije

Regular User



Primjer: UC-MUC dijagram za web forum

Regular User

Send a benign message
for posting to the Forum

Attacker

Send a Message loaded with
XSS Script to post to the Forum

The message gets
posted to the Forum

Administrator

Sanitize the message for any
potential script to trigger
XSS attack and then post
to the Forum

includes

threaten

mitigate

extends

includes



Alati za softversku sigurnost (ne mrežnu)



◆ Microsoft SDL i derivati

- Attack Surface Analyzer – smanjenje površine napada
- Microsoft Threat Modeling Tool – modeliranje prijetnji
- MiniFuzz basic file fuzzing tool – fuzz testiranje
- Regular expression file fuzzing tool – testiranje potencijalnih DoS ranjivosti

◆ Statička analiza

- StyleCop <https://stylecop.codeplex.com/> # slično, FxCop
- CodeSmart <http://www.axtools.com/>
- NDepend <http://www.ndepend.com/>
- PMD Java, Checkstyle, FindBugs+Find Security Bugs

Resursi

- ◆ Open Web Application Security Project (OWASP)
 - <http://www.owasp.org>, OWASP Top Ten vulnerabilities in web applications.
- ◆ Building Security In
 - <https://buildsecurityin.us-cert.gov/bsi/home.html>
- ◆ SANS Institute
 - <http://www.sans.org/> , CWE/SANS Top 25 Most Dangerous Prog. Errors
- ◆ CERT (Computer Security Incident Response Team)
 - <http://www.cert.org/> , <http://www.cert.org/secure-coding/>, <https://www.cert.hr>
- ◆ Cloud Security Alliance
 - <https://cloudsecurityalliance.org/>
- ◆ Ostalo
 - CVE (Common Vulnerabilities and Exposures), <http://cve.mitre.org/>
 - Security Tracker , <http://securitytracker.com/>
 - US-CERT Cyber Security Bulletins <http://www.us-cert.gov/cas/bulletins/>
 - Web Application Security Consortium (WASC), <http://www.webappsec.org/>
 - MSDN, <http://msdn.microsoft.com/security>

Reference

- ◆ Noopur Davis: Secure Software Development Life Cycle Processes, Software Engineering Institute, Carnegie Mellon University, 2013
 - http://resources.sei.cmu.edu/asset_files/whitepaper/2013_019_001_297287.pdf
- ◆ Microsoft SDL , <http://www.microsoft.com/security/sdl>
 - STRIDE, <http://msdn.microsoft.com/en-us/magazine/cc163519.aspx>
 - DREAD, <http://msdn.microsoft.com/en-us/library/ff648644.aspx>
 - SDL Quick Security References, <http://www.microsoft.com/en-us/download/details.aspx?id=13759>
- ◆ BSIMM Building Security In – Maturity Model, <http://bsimm.com>
- ◆ OpenSAMM Software Assurance Maturity Model, <http://opensamm.org>
- ◆ OWASP Application Threat Modeling
 - https://www.owasp.org/index.php/Application_Threat_Modeling

- ◆ Bruce Schneier, <https://www.schneier.com/>
 - If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.
 - Unless you think like an attacker, you will be unaware of any potential threats!
 - You can't defend. You can't prevent. The only thing you can do is detect and respond.