ZSIS sažetak drugog ciklusa

# Sigurnost programske podrške

**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

**Sigurna 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

**Sigurnost aplikacija (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 imovine
* **Protumjera** (countermeasure) – mjera zaštite i ublažavanja rizika

Kada je sigurnost softverski problem?

* ovisi o zahtijevanim promjenama
* mrežni problem – zahtijeva promjenu mrežnih mehanizama, kao što su mrežni protokoli
* problem OS – zahtijeva promjenu mehanizama OS kao što su 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 , *browser* kao "OS" budućnosti ?

**Uzroci problema softverske sigurnosti**

* nedostatak svijesti, značaja (awareness), nedostatak znanja
* Sigurnost kao sekundarna briga - primarna je funkcionalnost, servis, udobnost
* **Funkcionalnost** – ono što aplikacija radi
* **Sigurnost** – bavi se onim što aplikacija ne bi smjela raditi

**Sigurnosni ciljevi : CIA**

* **C**onfidentiality (povjerljivost, tajnost) - uskraćivanje “čitanja” neautoriziranim korisnicima
* **I**ntegrity (integritet) - uskraćivanje promjena neautoriziranim korisnicima
* **A**vailability (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 jedinstvenu informaciju kojom 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.*

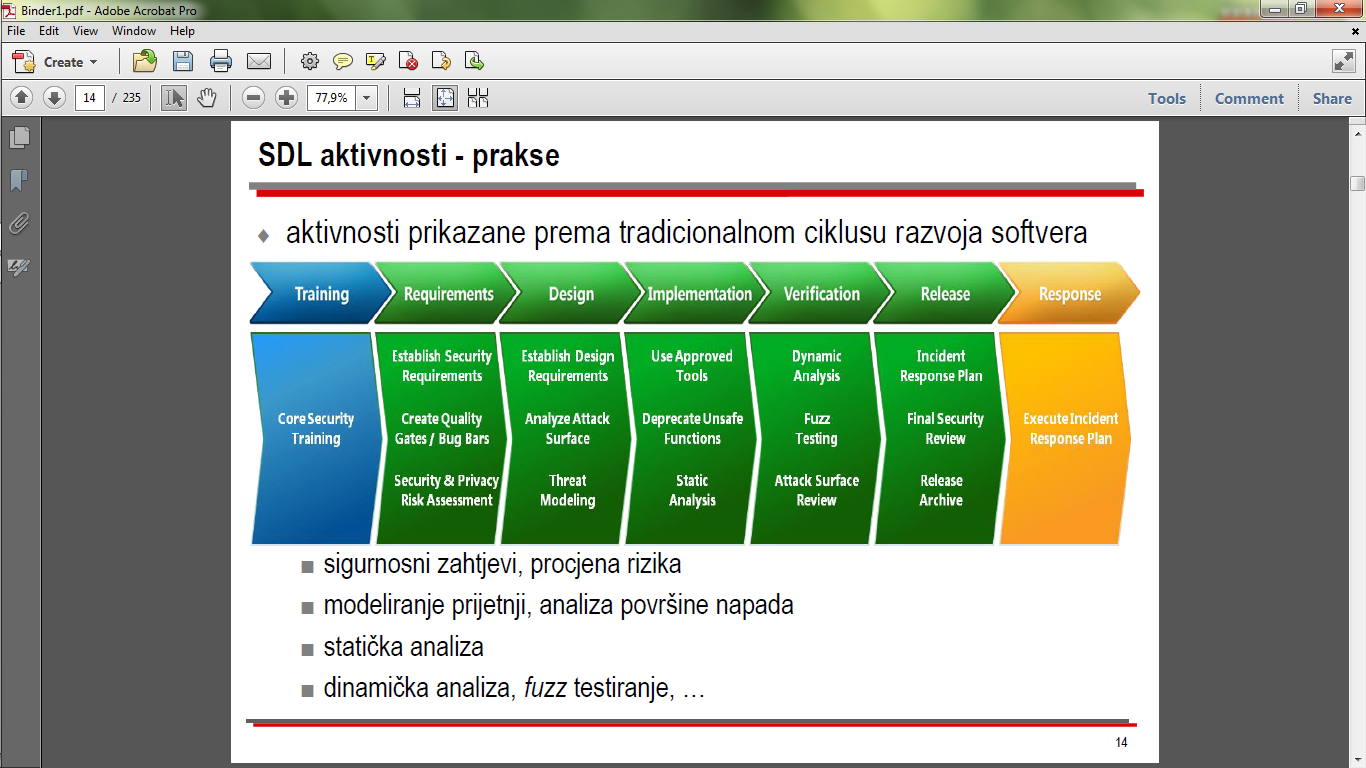
Zaštita protiv prijetnji? - *Use forensics to find & eliminate,*  *Mitigate by punishment, if possible*

Zaštita protiv ranjivosti? *Engineer secure software!*

## Životni ciklus sigurnog softvera



**Microsoft’s Trustworthy Computing Security Development LC -**  **Životni ciklus razvoja sigurnosti (skraćeno SDL)**



### 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

### 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)

**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

**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 (više u zasebnom poglavlju)**

* 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. /GS (Buffer Security Check), *StackGuard*

**SDL** **Practice 10: Static Analysis**

* osigurava inspekciju programskog koda, ali ju ne može zamijeniti ! npr. *StyleCop, CodeSmart, Ndepend*

### 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 mjerenje 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)
* *on-call* kontakt koji ima autoritet odlučivanja, 24x7
* plan servisiranja sigurnosti za izvana nabavljene 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

**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…

**Opcionalne aktivnosti**

*Nadzor, ručna inspekcija koda (code review)*

* vješti pojedinci ili sigurnosni tim ili savjetnik sigurnosti
* usmjerena 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 automatiziranom ili ručnom analizom programskog koda

*Analiza povredivosti sličnih aplikacija* - analizom dostupnih informacija na Internetu

## Sigurnosni zahtjevi

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 kkn/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

**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

**Postupci inženjerstva zahtjeva**

* REVEAL = **R**equirements **E**ngineering **VE**rification and V**AL**idation - from Praxis
* SQUARE = **S**ecurity **QUA**lity **R**equirements **E**ngineering Methodology - from CMU/SEI
* TRIAD = **T**rustworth **R**efinement through **I**ntrusion-**A**ware **D**esign from CMU/SEI
* AEGIS = **A**ppropriate and **E**ffective **G**uidance in **I**nformation **S**ecurity
* Misuse Cases ili Abuse Cases (zajednička kratica MUCs)
  + Slučajevi zloupotrebe, zloporabe (nenamjerno) ili zlostavljanja (namjerno)
  + scenariji u kojima sudionik kompromitira sustav

**Slučajevi zloporabe**

* Pogled protivnika/napadača - Dohvat podataka korisnika, Izmjena cijene, ocjene, …, Uskraćivanje usluge

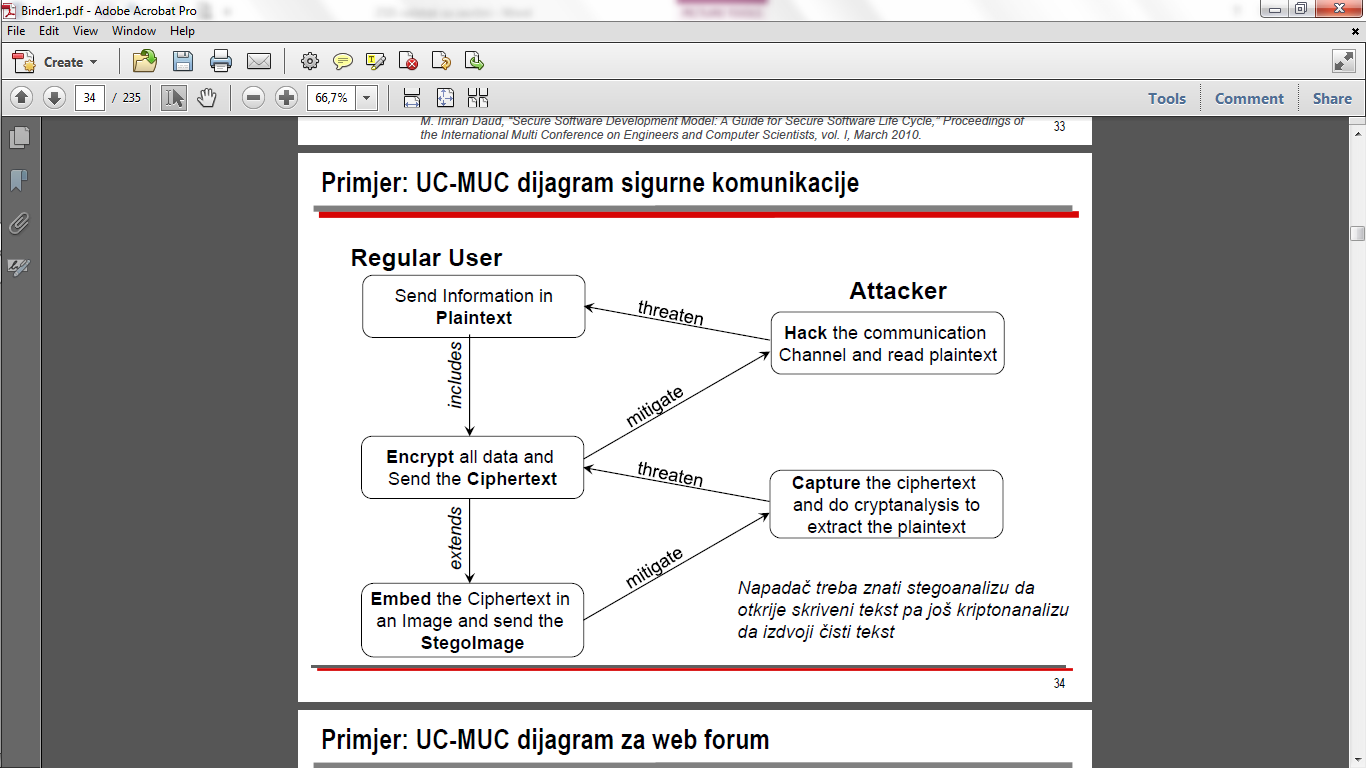
Razvoj slučajeva - Brainstorming – pretpostavke, obrasci napada, rizici

Sigurnosni zahtjevi – generalizirana forma MUCova -> Anti-zahtjevi – što o**NE**mogućiti

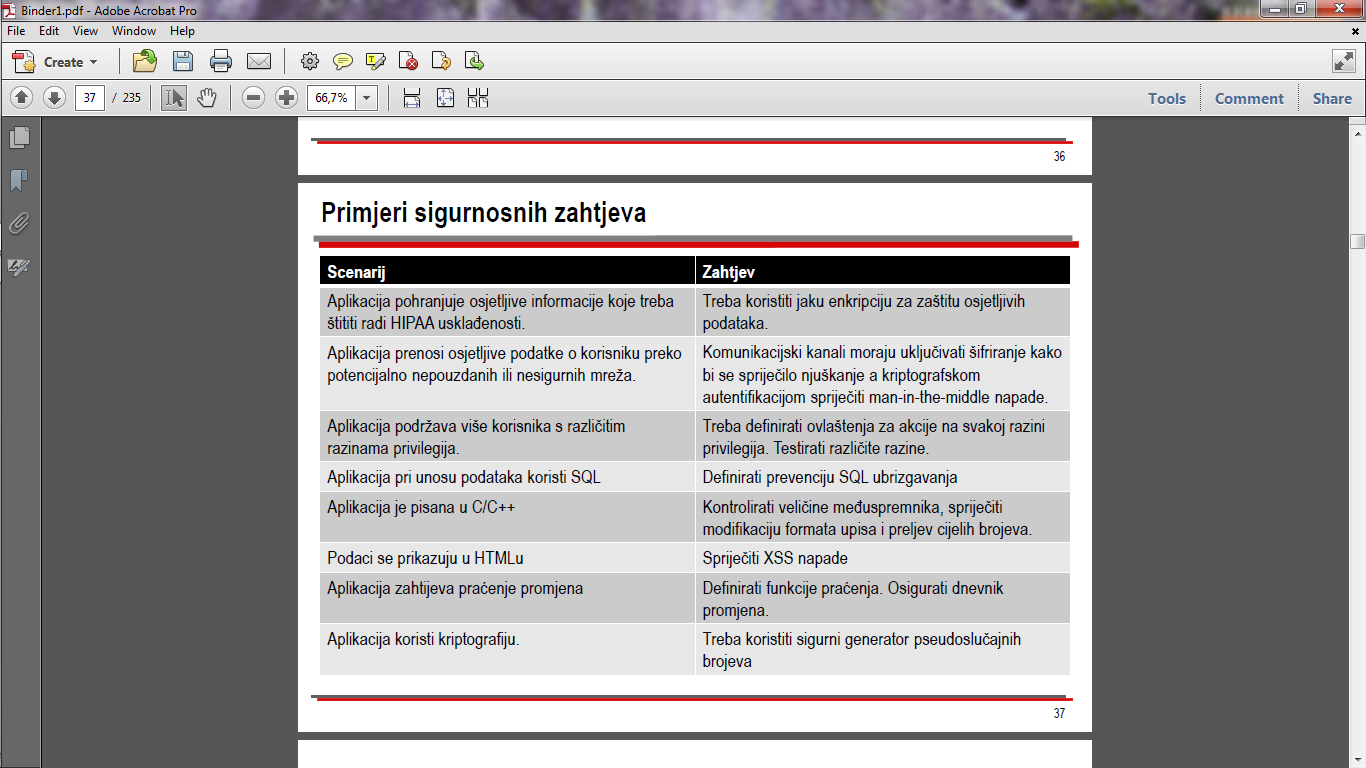
**Povezivanje slučajeva zloporabe**

* 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**



**Primjeri sigurnosnih zahtjeva**:



# Projektiranje sigurnosti

## Modeliranje prijetnji

**Modeliranje prijetnji (threat modeling)**

* sigurnosna analiza koja pomaže u otkrivanju najvećih sigurnosnih opasnosti
* cilj je odrediti koje prijetnje i na koji način treba ukloniti
* pretpostavka - proizvod nije siguran ako se ne procijene prijetnje i smanji rizik
* koristi:
  + bolje shvaćanje aplikacije - naročito novi članovi
  + pronalaženje pogrešaka
  + procjena da **MP pronađe 50% pogrešaka**, a ostatak testiranjem i analizom koda
  + pogreške složenih aplikacija, koje se rijetko pronađu drukčije (pogreške u dizajnu)

**Načela i proces modeliranja prijetnji**

Analiziranje prijetnji - dugotrajan posao, bitno je da se obavi kvalitetno i najbolje iterativno

važno:

Jednostavnije je pronaći sigurnosni propust u dizajnu aplikacije nego mijenjati kasnije

Model prijetnji treba biti aktualan (ažuran) - prijetnje i načini kako ih zaobići

Proces modeliranja prijetnji

1. Određivanje ciljeva zaštite
2. Arhitektura aplikacije
3. Dekompozicija aplikacije
4. Određivanje prijetnji
5. Dokumentiranje prijetnji
6. Rangiranje prijetnji

## Izlaz

Dokument s modelima - definicijom arhitekture i popisom prijetnji

### Korak 1

identifikacija resursa koje treba zaštititi od spremišta podataka (datoteka, BP), …, do web stranica

### Korak 2 – Pregled arhitekture

Dokumentiranje

* funkcije aplikacije - što aplikacija radi
* arhitektura aplikacije i način fizičke ugradnje (konfiguracija)
* tehnologije implementacije

Modeliranje funkcionalnosti

* slučajevi korištenja (*use case*)
* razumijevanje načina korištenja
* kontekst rada aplikacije
* primjeri: zaposlenik vidi poslovne podatke, može ažurirati osobne podatke, menadžer vidi podatke zaposlenika.

Provjera (kršenja) poslovnih pravila

* Npr. korisnik pokušava promijeniti tuđe osobne podatke
* To ne bi smio ako nema dovoljnu razinu dozvola

*Dijagram visoke razine* - opisuje strukturu (komponente) sustava -> ovisno o složenosti aplikacije treba izraditi detaljnije dijagrame dijelova (npr. dijagrame pojedinih slojeva višeslojne aplikacije)

### Korak 3 – Dekompozicija aplikacije

* Izrada sigurnosnog profila (security profile)
* Određivanje
  + granica povjerenja (trust boundaries)
  + toka podataka
  + mjesta unosa
  + privilegiranog koda
* Tehnike dekompozicije - funkcionalna dekompozicija, dijagram aktivnosti, dijagram toka podataka, …

**Granice povjerenja i tokovi podataka**

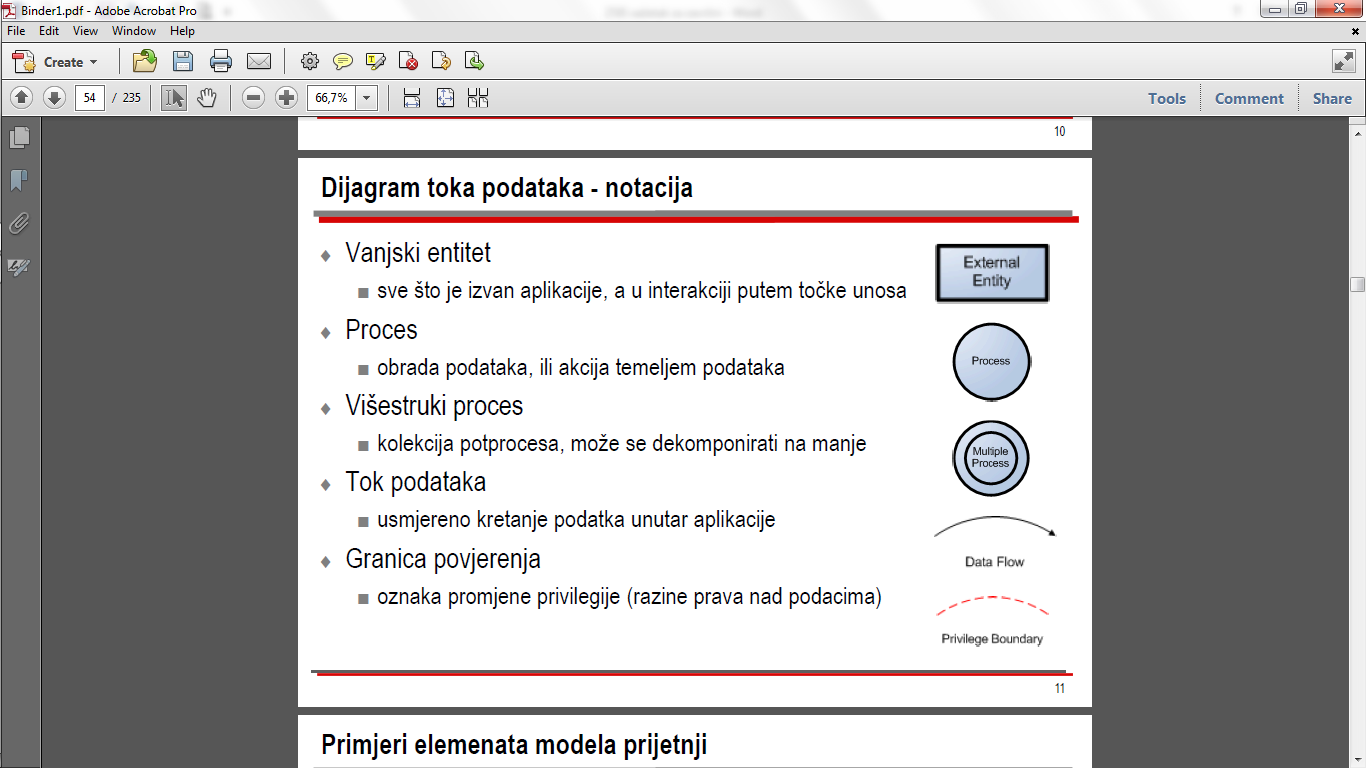
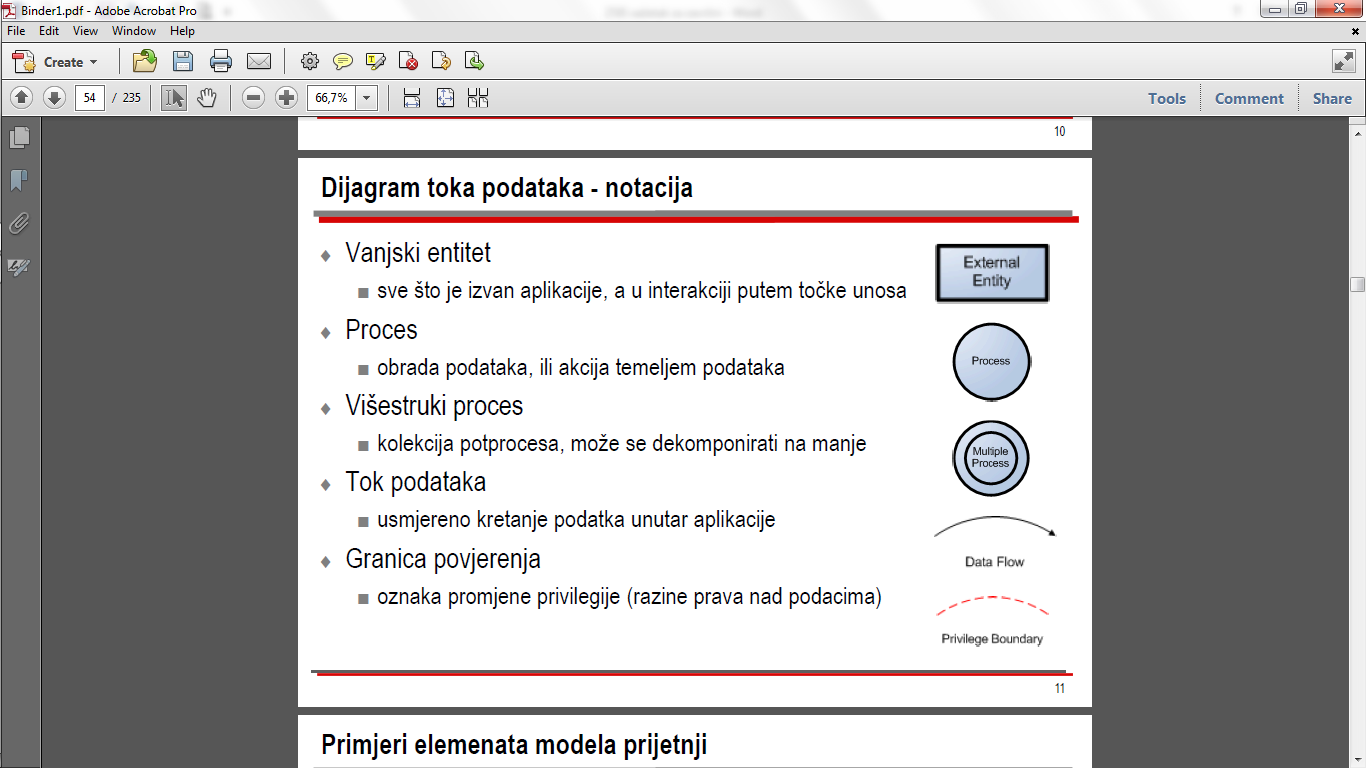
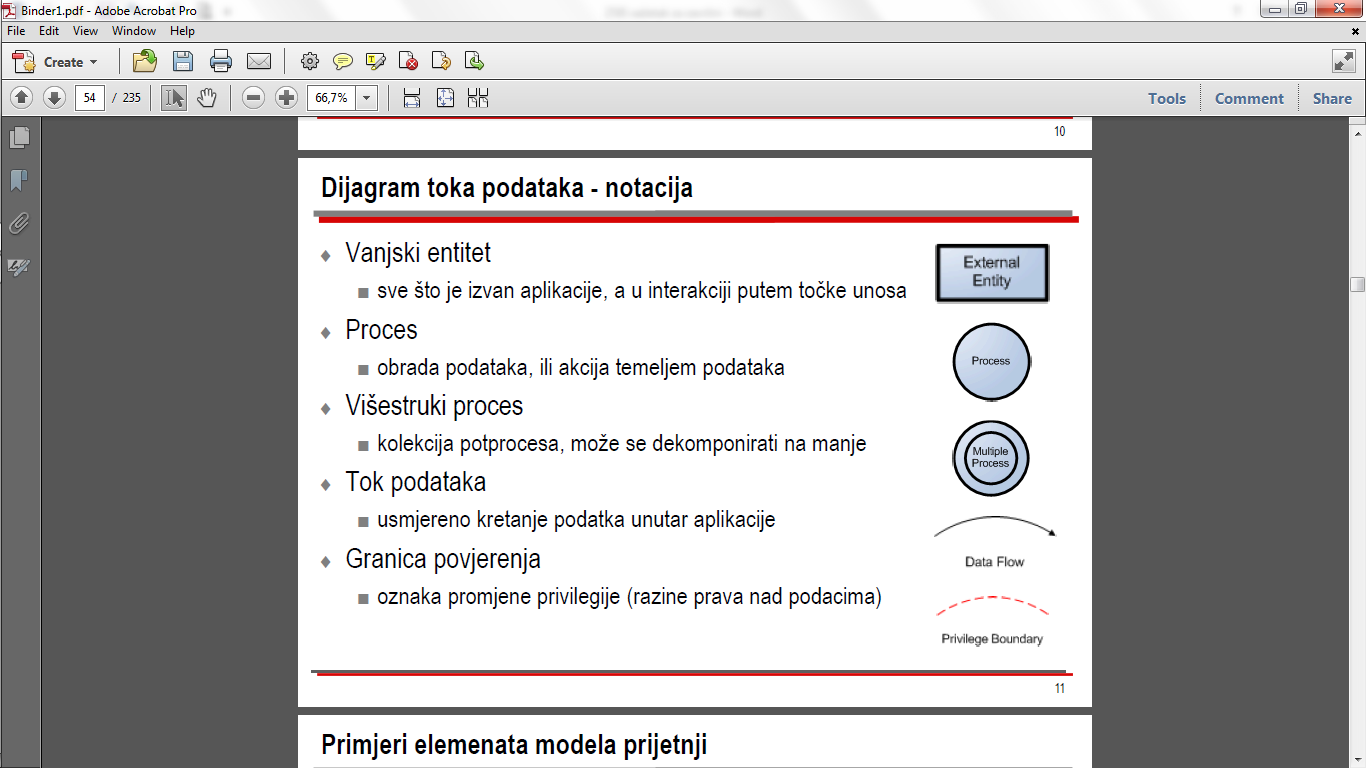
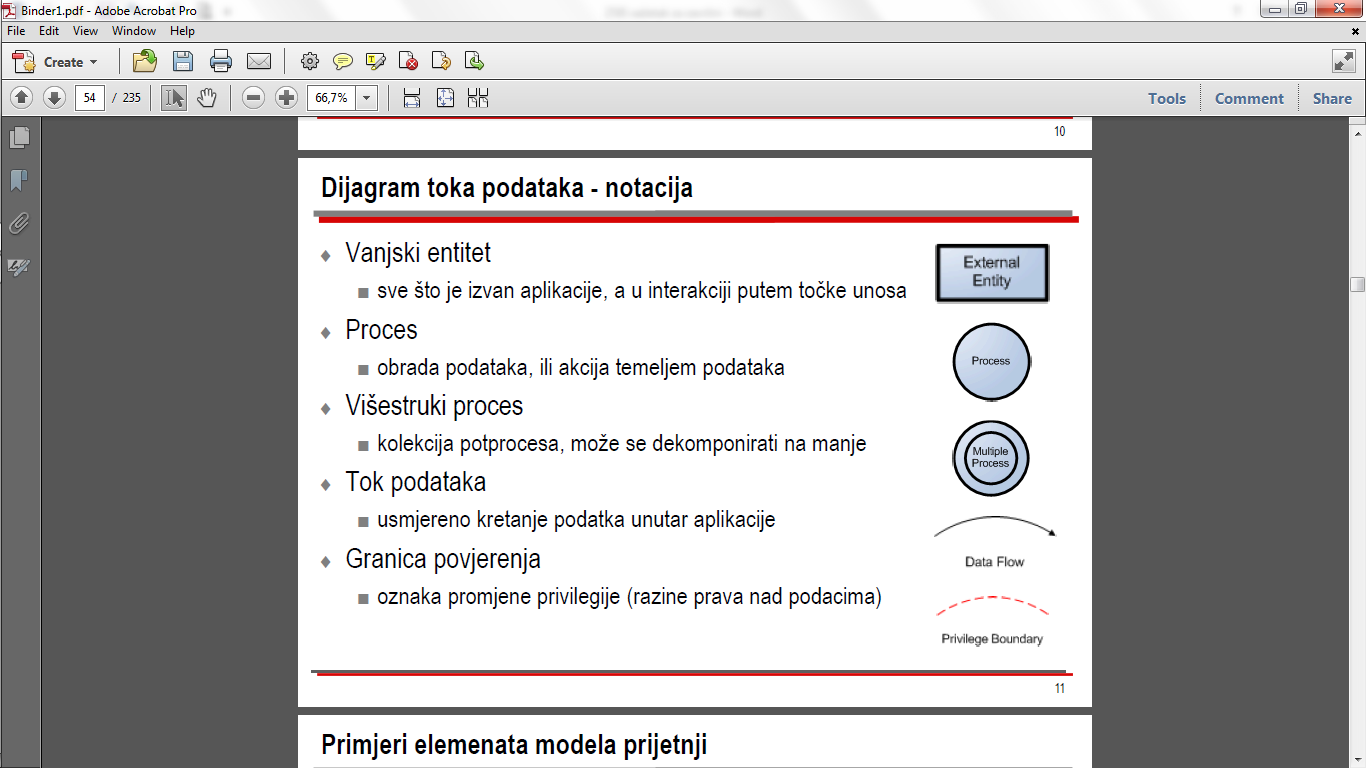
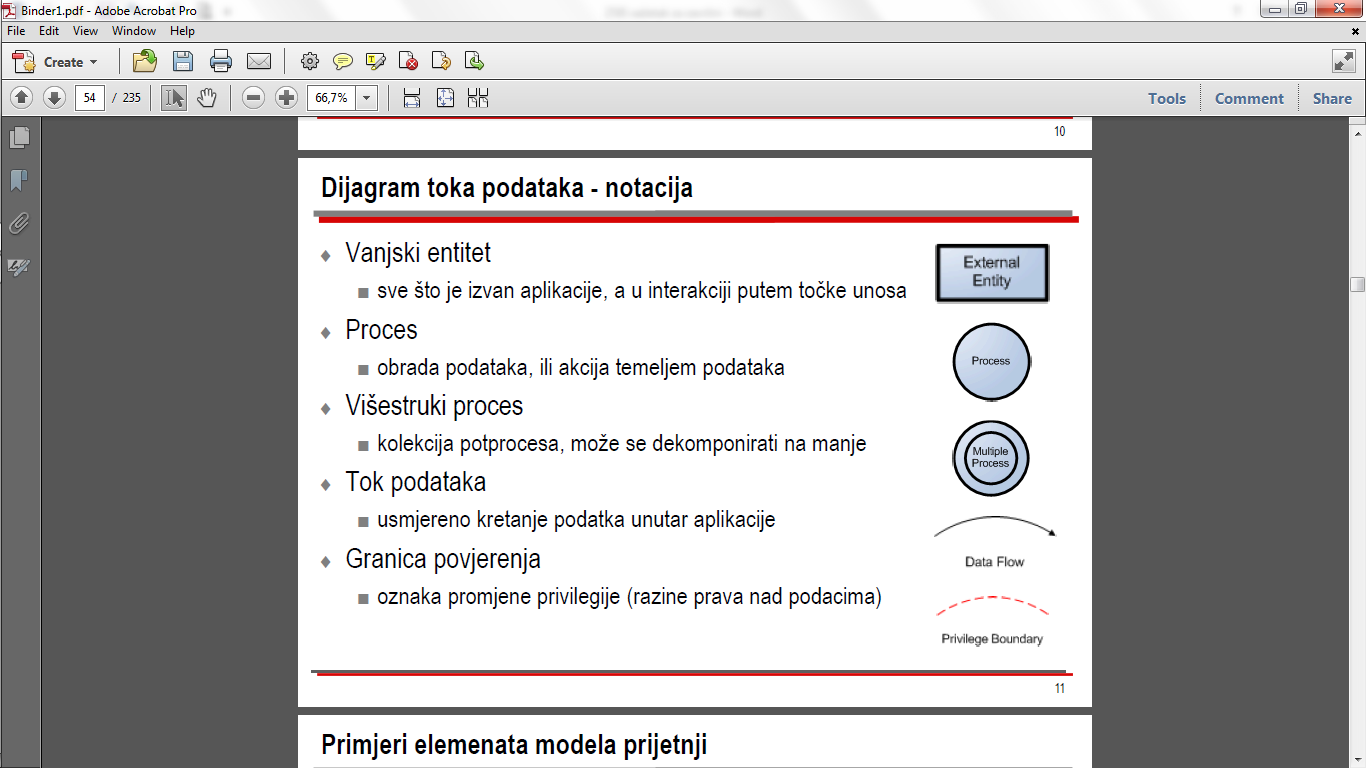
Određivanje granica povjerenja

* analiza okruženja resursa određenog dizajnom aplikacije
* za svaki podsustav, procjena je li ulazni tok ili korisnički unos povjerljiv (ako nije – razmotriti kako ih autentificirati i autorizirati)
* procjena je li pozivajući programski kod povjerljiv
* provjera povjerenja poslužitelja (server trust relationships)

Određivanje toka podataka (data flow)

* iterativna dekompozicija
* analizom tokova između podsustava, pa u dubinu (0-sistem, 1-glavne mogućnosti, 2-detalji)

**Dijagram toka podataka - notacija**



**Vanjski entitet** - sve što je izvan aplikacije, a u interakciji putem točke unosa

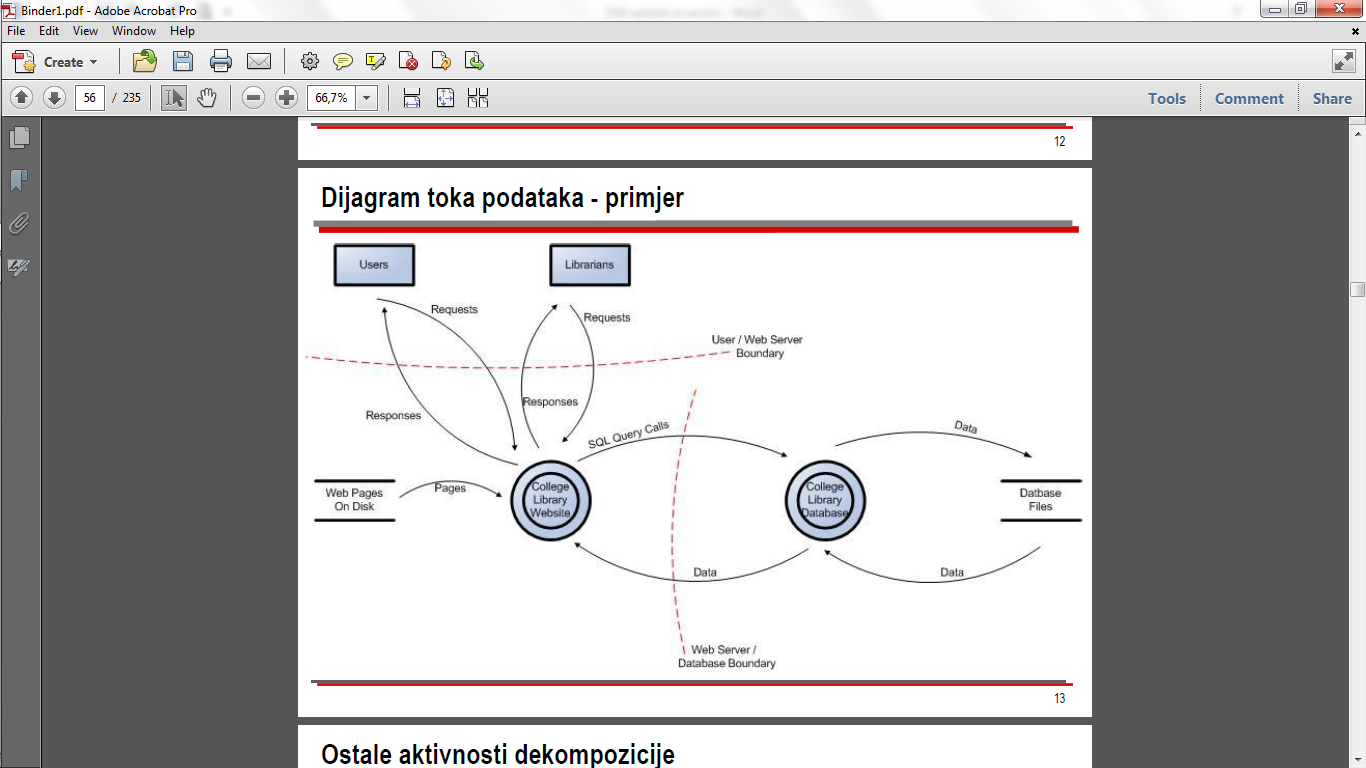
**Proces** - obrada podataka, ili akcija temeljem podataka

**Višestruki proces** - kolekcija potprocesa, može se dekomponirati na manje

**Tok podataka** - usmjereno kretanje podatka unutar aplikacije

**Granica povjerenja** - oznaka promjene privilegije (razine prava nad podacima)

Primjer dijagrama toka



**Ostale aktivnosti dekompozicije**

Određivanje točki unosa (entry point)

* dijelovi korisničkog sučelja, npr. stranice web aplikacije
* priključne točke prijenosa podataka, npr. sučelja web servisa, remoting komponente, fizički portovi i priključnice (sockets)

Određivanje privilegiranog koda

* koji pristupa određenim tipovima sigurnih resursa ili obavlja privilegirane operacije
* npr. sigurni resursi: DNS poslužitelji, *registry*, *event log*, …, pisači, web servisi, …
* npr. sigurne operacije: *unmanaged code calls*, refleksija, serijalizacija, …

Dokumentiranje profila sigurnosti

* određivanja pristupa projektiranju i ugradnji za validaciju unosa, autentifikaciju, autorizaciju, upravljanje konfiguracijom, …
* primjeri pitanja na koje treba odgovoriti pri izradi profila

### Korak 4 - Određivanje prijetnji

Odrađuju razvojni tim i tim za testiranje (arhitekti, sigurnjaci, razvojnici, testeri i sistem administratori)

Osnovni pristupi

**STRIDE** praksa modeliranja definirana SDL-om - akronim (*Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege*)

Kategorizirane liste prijetnji

* popis uobičajeno "sumnjivih" prijetnji (*laundry list*)
* grupirano po kategorijama: mreža, poslužitelj, aplikacija
* primjena liste na vlastitu arhitekturu

Ostale korisne tehnike

* Stabla prijetnji (*threat trees*) - opisuju koje odluke napadač mora donijeti pri napadu na neku komponentu
* Obrasci napada (*attack patterns*)

**STRIDE - procjena po kategorijama prijetnji**

* **Spoofing** – zavaravanje, lažiranje - preuzimanje tuđeg identiteta s ciljem pristupa resursima u mreži (npr. ilegalno dohvaćanje tuđih podataka prilikom autentifikacije)
* **Tampering** [with Data] – zlonamjerna izmjena podataka - nedozvoljena izmjena npr. u bazi podataka ili prilikom prijenosa mrežom
* **Repudiation** – poricanje - mogućnost korisnika da porekne akciju, a da mu se to ne može dokazati ( npr. „nisam obrisao”, „nisam naručio”, …)
* **Information disclosure** - otkrivanje informacija, neželjeno izlaganje privatnih podataka ( npr. korisnik vidi sadržaj tuđe datoteke na što nema pravo)
* **Denial of service** - uskraćivanje usluge - onemogućuje normalan rad sustava, relativno jednostavno i anonimno ( npr. *flooding, amplification, protocol vulnerability, malformed packets)*
* **Elevation of privilege** - povišenje ovlasti - korisnik s ograničenim ovlastima preuzima identitet korisnika s većim ovlastima

**STRIDE - postupak**

* Sustav se raščlanjuje u relevantne komponente
  + procjenjuje se osjetljivost na prijetnje svake komponente
  + prijetnje se smanjuju (mitigation) prikladnim svojstvima sigurnosti
  + ponavlja se (rekurzivno) do zadovoljavajućeg rezultata
* Utjecaj prijetnji na pojedine dijelove sustava - … analizom dijagrama toka podataka

**Stabla prijetnji**

Za svaku komponentu dobivenu dekompozicijom određuju se moguće prijetnje

utvrđuje se način na koji se prijetnje odražavaju na sustav

Primjer

* korijen predstavlja prijetnju
* djeca predstavljaju korake koje napadač mora poduzeti da bi ostvario prijetnju



Korištenje STRIDE nad stablima prijetnji je jednostavno - za svaki dio sustava se ispituje je li podložan nekoj od STRIDE kategorija (npr. može li napadač uskratiti rad procesa, vidjeti podatak itd.)

**Obrasci napada**

Općenita reprezentacija uobičajenih napada - definira cilj, uvjete, tehniku i rezultat napada

Naglasak je na tehnici napada (kod STRIDE na ciljevima napadača)

### Korak 5 - Dokumentiranje i rangiranje prijetnji

Predložak za evidenciju prijetnji

* svakako se popunjavaju opis i cilj
* rizik se ostavlja za naredni korak
* ostali atributi mogu biti opcionalni

### Korak 6 - Rangiranje prijetnji

Rangiranje – određivanje važnosti (rate the threats)

* često se koriste tehnike za određivanje rizika
* **rizik** = vjerojatnost događaja \* potencijalna šteta
* **vjerojatnost** npr. u rasponu 1-10
* **šteta** npr. u rasponu 1-10
* rizik u rasponu 1-100
* raspodjela u tri grupe (visok, srednji, nizak) koje predstavljaju prioritete

Problem:

* članovi tima ne mogu se usuglasiti oko vrijednosti

Rješenje:

* DREAD model, rangiranje rizika za zadanu prijetnju

**DREAD (model procjene rizika)**  – klasifikacija računalnih prijetnji

* ***Damage potential*** *-* moguća šteta, veličina štete bude li napad uspješan
* ***Reproducibility*** – reproduktivnost, koliko je jednostavno ponoviti napad
* ***Exploitability***– iskoristivost, trud i znanje potrebnih za uspješan napad
* ***Affected users*** – zahvaćeni korisnici, moguće uspjelim napadom, postotno
* ***Discoverability*** - mogućnost otkrivanja, teško mjerljivo

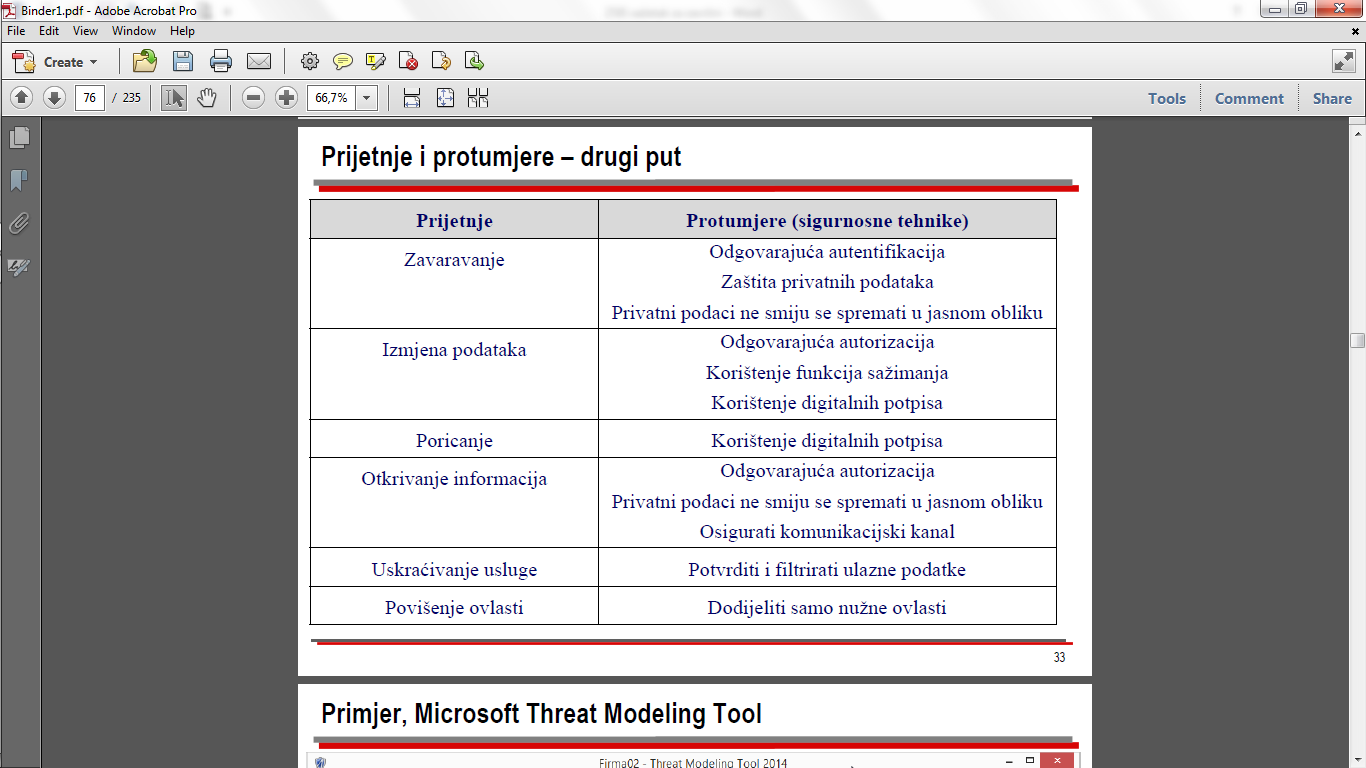
Procjena svake prijetnje po navedenim parametrima

* pojedinačno vrijednost od 1 do 10 (najmanje loše – najgore)
* ukupan rizik - prosjek 5 pojedinačnih DREAD vrijednosti

Alternativa – jednostavna shema ocjenjivanja

### Razrješenje prijetnji (nakon modeliranja)

* Popraviti (smanjenje, redukcija rizika)
  + ukloniti uzrok
* Ne učiniti ništa (prihvatiti rizik)
  + loše, ako je problem realan, kad-tad će se ostvariti te morati popraviti
* Obavijestiti korisnika te mu prepustiti odluku o korištenju (prijenos)
  + problematično
  + mnogi korisnici ne znaju koja je prava odluka, a obavijesti budu nerazumljive
  + koristiti jedino ako postoji velika potreba za korištenjem (rizične) usluge
* Uklanjanje rizičnog svojstva (izbjegavanje)
  + kad se problem ne može odmah ispraviti (npr. nema vremena i sl.),
  + ispraviti u sljedećoj verziji



## Smanjenje površine napada

Površina napada - kolekcija ulaznih točaka programskog proizvoda

* Korisnička sučelja, web servisi, izravan pristup BP, mrežni kanali, API, …
* kanali za komunikaciju s resursima = vektori napada
* mjera "napadljivosti" (attackability)

Veća površina napada = više posla zaštite = veća potencijalna šteta

Površina određuje rizik napada – mjera potencijalnog pristupa i udara

Iskustvo pokazuje da su pojedini vektori rizičniji

Npr. privilegirani (*root*) servisi, datoteke s punim pristupom (*rwxrwxrwx*), skripte (JScript , VBScript) i aktivne kontrole (ActiveX)

**Združeni model površine napada**

kontrole pristupa smanjuju - mogućnost da se dosegne sustav i broj elemenata koji su vidljivi ili se mogu koristiti

**Smanjenje površine napada (attack surface reduction)**

Glavni ciljevi

* Smanjenje količine koda koji se izvodi „po viđenju” (by default)
* Smanjenje količine koda kojem mogu pristupiti nesigurni (untrusted) korisnici „po viđenju”
* Zatvaranje pristupnih točaka (access points, entry points) – vrata koja se lako otvaraju/iskorištavaju
* Ograničavanje štete u slučaju da pristupna točka bude iskorištena
* Krajnji cilj – odbijanje budućih napada

**Uobičajena metrika softverske sigurnosti**

Razina programskog koda - brojanje bugova

* Ne ubraja bugove koji (još) nisu pronađeni
* Svi su bugovi jednake težine, iako je neke lakše iskoristiti
* Neki bugovi mogu prouzročiti više štete nego drugi

Razina proizvoda/sustava

* Brojanje koliko puta je verzija sustava spomenuta u CERT, MITRE CVE, … biltenima
* Zanemaruje specifične konfiguracije (instalirane zakrpe, uključene ili isključene standardne postavke (defaults), rad u *admin* modu)

**Mjerenje površine napada**

Mjerenje „avenija” napada (avenues of attack) - možebitno napadane mogućnosti - “more likely to be attacked” features

Mjerenje relativne sigurnosti

* Delta mjerenje – razlike između verzija istog proizvoda (npr. v1 naspram v2)
* Neupotrebljivo za usporedbu različitih aplikacija

Postupak

1. Osnovica (baseline) + tjedna mjerenja
2. Određivanje minimalne površine na početku
3. Ako se površina povećava – odrediti kako ju smanjiti

### Najbolje prakse

Redukcija koda koji se izvodi *by default*

* Isključiti mogućnost koju ne koristi barem 80% korisnika (Zaustavljen servis ne može biti napadnut)
* dinamički web sadržaj treba biti opcionalan – zahvaća samo one koji ga pokrenu
* Rješenje nije samo isključivanje - ograničenje pristupa pokrenutom kodu

Smanjenje pristupa od strane nepouzdanih (untrusted) korisnika

* Ograničenje pristupa na lokalnu mrežu ili raspon IP adresa
* Autentifikacija

Redukcija privilegija radi ograničavanja potencijalne štete

* Ukidanje privilegija koje nisu prijeko potrebne
* Pokretanje koda u sigurnosnom okviru (sandbox running code)
* ograničenjem dozvola na koje kod ima pravo, *Java.Class.SecurityManager*, ili .NET *System.Security.SecurityManager*
* Po potrebi podići dozvole – privremeno, što kraće
* Pripaziti na granice povjerenja (postupak modeliranja prijetnji) - naročito *putove* anonimnih prijetnji (anonymous threat paths) → autorizacija gdje treba
* Ne pokretati servise kao SYSTEM (demone kao *root*) ili s administratorskim pravima dok ne budu iscrpljene druge mogućnosti!

Primjer:

*Backup Operator account* – čita sve datoteke bez obzira na njihov ACL SYSTEM radi isto ali i *restore*, *debug*, "*act as part of OS*" i k tomu je *admin*

Definiranje površine napada tijekom dizajna/projektiranja

* skicirati površinu napada i ustanoviti :
* protokole, krajnje točke koje trebaju autentifikaciju i autorizaciju
* isključene (off-by-default) mogućnosti – autostart (pr. Windows \ Services, starter.exe)
* ponovno iskoristive komponente (ActiveX, COM, .NET asembliji, itd.)
* identitete procesa, instalirane korisničke račune

Ostali postupci

* Modeliranje prijetnji
* Pregled površine napada – analizator: osnovica + razlike
* Pregled dizajna – traženje prijetnji i mogućnosti redukcije
* Pregled koda – defenzivno programiranje i sigurno kodiranje

# Sigurnost u sustavima za elektroničko poslovanje

**Elektroničko poslovanje**

Primjeri elektroničkog poslovanja

* elektroničko komuniciranje s drugim poduzećima radi narudžbe proizvoda i usluga te njihovo elektroničko plaćanje
* poslovanje medu tvrtkama – B2B (Business-to-Business)
* prodavanje proizvoda i usluga preko Web sjedišta
* e-trgovina, poslovanje tvrtke s krajnjim potrošačem – B2C (Business-to- Consumer)

**Elektroničko poslovanje - razmjena poslovnih dokumenata**

Osnovni poslovni procesi u dobavnom lancu i elektronički dokumenti koji se razmjenjuju:

* Katalog *e-katalog*
* Naručivanje *e-narudžbenica*
* Otpremanje *e-otpremnica*
* Primanje *e-primka*
* Fakturiranje *e-račun*
* Plaćanje e-nalog za plaćanje

**Dobavni lanac – osnovni poslovni procesi i dokumenti**

* Suvremeni elektronički dokumenti koji se razmjenjuju u elektroničkom poslovanju većinom su u formatu XML.
* Postoje i starije norme **EDI** (*Electronic Data Interchange*), od kojih je najvažnija norma **EDIFACT** (*Electronic Data Interchange For Administration, Commerce and Transport).*
* razmjena suvremenih poslovnih elektroničkih dokumenata – tehnologije: XML i usluge Weba

**Sigurnost pri razmjeni elektroničkih dokumenata**

* **e-račun** je najrašireniji elektronički poslovni dokument
* sve zemlje članice EU trebaju omogućiti primanje **e-Računa** za porezne svrhe (PDV) ako su ispunjena dva uvjeta:
  + **primatelj se mora složiti** s primanjem računa u elektroničkom formatu;
  + **integritet** (nemogućnost izmjene) **i autentičnost** (deklarirani pošiljatelj je stvarni pošiljatelj) moraju biti osigurani pri prijenosu i arhiviranju.

Ovaj drugi zahtjev može se ispuniti bilo **naprednim elektroničkim potpisom** ili kroz elektroničku razmjenu podataka (EDI) s ugovorenim sigurnosnim mjerama.

* **u elektroničkom poslovanju** se tehnologija usluga weba počela primjenjivati **pojavom sigurnosnih normi za usluge weba**

**Digitalni potpis u RH**

U Republici Hrvatskoj digitalno potpisivanje definirano je Zakonom o elektroničkom potpisu (NN 10/02)

Dvije vrste potpisa:

* **lektronički potpis**
* **Napredni elektronički potpis**

Zamjenjuje vlastoručni potpis ili vlastoručni potpis i otisak pečata

Pouzdano jamči identitet potpisnika i udovoljava sljedećim kriterijima:

* Povezan je isključivo s potpisnikom
* Nedvojbeno identificira potpisnika
* nastaje korištenjem sredstava kojima potpisnik može samostalno upravljati i koja su isključivo pod nadzorom potpisnika
* sadržava izravnu povezanost s podacima na koje se odnosi i to na način koji nedvojbeno omogućava uvid u bilo koju izmjenu izvornih podataka

**Digitalni certifikat -** Rješava problem dokazivanja identiteta

* Povezuje identitet korisnika s njegovim javnim ključem - **potvrđuje da je određeni korisnik vlasnik određenog javnog ključa**

Skup podataka koji identificira korisnika i davatelja usluge certificiranja

Norma: - za digitalne certifikate koristi se norma **X.509**

imena su u certifikatima prikazana kao parovi: ime – vrijednost

Ako pošiljatelj potpiše poruku svojim privatnim ključem, primatelj može **znati da se radi upravo o tom pošiljatelju**:

* ako može **dešifrirati** digitalni potpis **javnim ključem pošiljatelja i**
* ako **digitalni certifikat potvrđuje** da je korišteni javni ključ upravo javni ključ tog pošiljatelja.

Pretpostavka za ovaj postupak je da korisnici imaju **povjerenje u certifikacijsko tijelo** koje je izdalo certifikat i potpisalo ga svojim privatnim ključem.

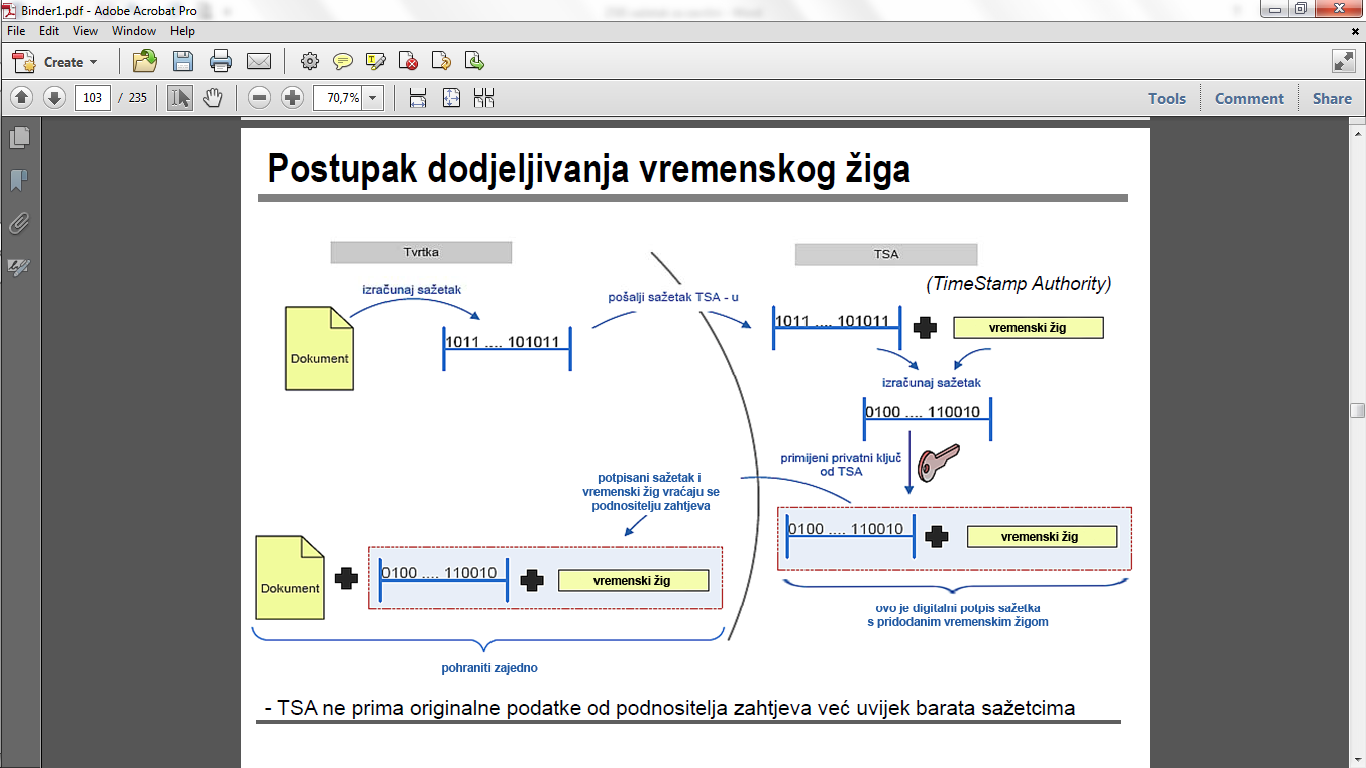
### Centar za pouzdano vremensko označavanje (TSA)

Stvara vremenske žigove kako bi se dokazalo da su određeni podatci postojali prije određenog vremena

**VREMENSKI ŽIG**

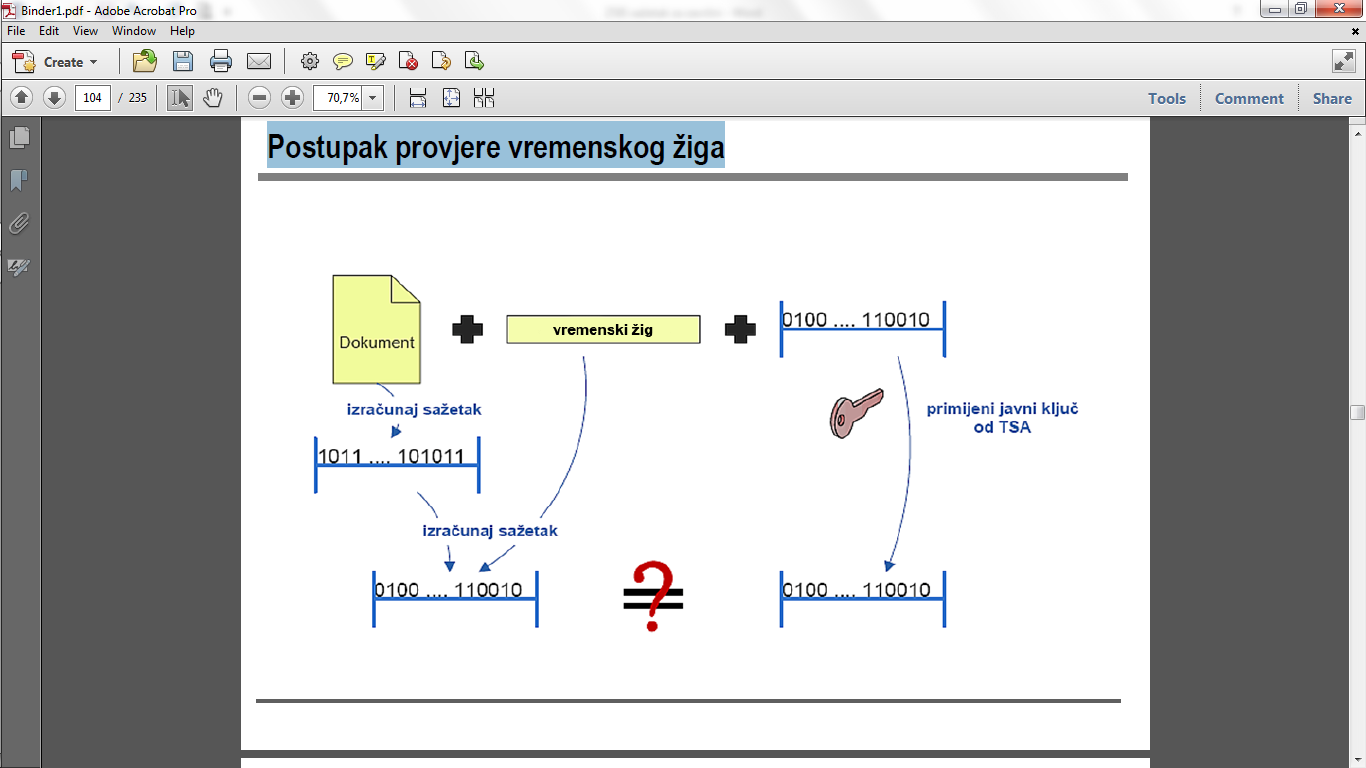
* Vremenski žig, vremenski pečat, vremenska oznaka, vremenska ovjera, engl. ***timestamp***
* Osigurava pouzdanost digitalnog potpisa i poslije isteka valjanosti ili opoziva certifikata potpisnika
* Pomoću vremenskog žiga **može se dokazati da je potpis napravljen prije isteka vrijednosti certifikata**

#### Postupak dodjeljivanja vremenskog žiga



TSA ne prima originalne podatke od podnositelja zahtjeva već uvijek barata sažetcima

#### Postupak provjere vremenskog žiga



Ako su sažetci jednaki, dokazano je da su i vremenska oznaka i dokument nepromijenjeni te da je TSA izdao vremensku oznaku

**Ne može se poreći da je podnositelj zahtjeva za vremenskom oznakom bio u posjedu originalnog dokumenta u vremenu naznačenom vremenskom oznakom**.

Ako sažetci nisu jednaki, to znači

* da su vremenska oznaka ili dokument promijenjeni
* ili da vremensku oznaku nije izdao navedeni TSA

## Sigurnost XML-dokumenata

Današnje elektroničko poslovanje uglavnom se temelji na razmjeni XMLdokumenata.

Sigurnosne norme ugrađene u XML: ***XML-Encryption*** i ***XML-Signature***

* dodaju se dokumentu bez kršenja pravila XML-a
* takvi dokumenti mogu se pregledavati korištenjem standardnih alata za XML
* sigurnost XML-dokumenata može biti implementirana i korištenjem standardnih sigurnosnih protokola
* ti algoritmi koriste binarne datoteke koje onda mogu biti interpretirane samo korištenjem posebnih alata

Za siguran prijenos dokumenata kroz mrežu može se koristiti protokol SSL

* time se **osigurava samo prijenos** podataka kroz mrežu, a **ne i pohrana**
* dokument poslan korištenjem isključivo SSL-a prestaje biti siguran onog trenutka kada stigne na odredište

Primjenom sigurnosnih mjera nad samim dokumentom korištenjem standarda za sigurnost XML-a, dokument se osigurava i u prijenosu i u kasnijoj pohrani jer se ne osigurava veza nego sami dokument

* norma *XML Digital Signature* koristi se za digitalno potpisivanje XML-dokumenata
* normom *XML Encryption* osigurava se povjerljivost

### Kanonikalizacija

* dva logički jednaka XML-dokumenta mogu biti različito zapisana
* primjerice, u jednom se nalazi **razmak viška ili prazan red viška**
* dva dokumenta logički jednaka, ali sažetak ta dva dokumenta dobiven hash-algoritmom nije jednak!
* kod digitalnog potpisivanja to za **posljedicu ima neuspješnu verifikaciju potpisa**, iako dokument logički nije promijenjen, tj. očekivalo bi se da bi verifikacija trebala biti uspješna
* kako bi se takvi problemi izbjegli, **XML-dokumente treba kanonikalizirati** tj. **svesti se na jednak (kanonički) oblik** (normiranje razmaka i sl.)

### XML-Signature (XML-DSig) - W3C norma

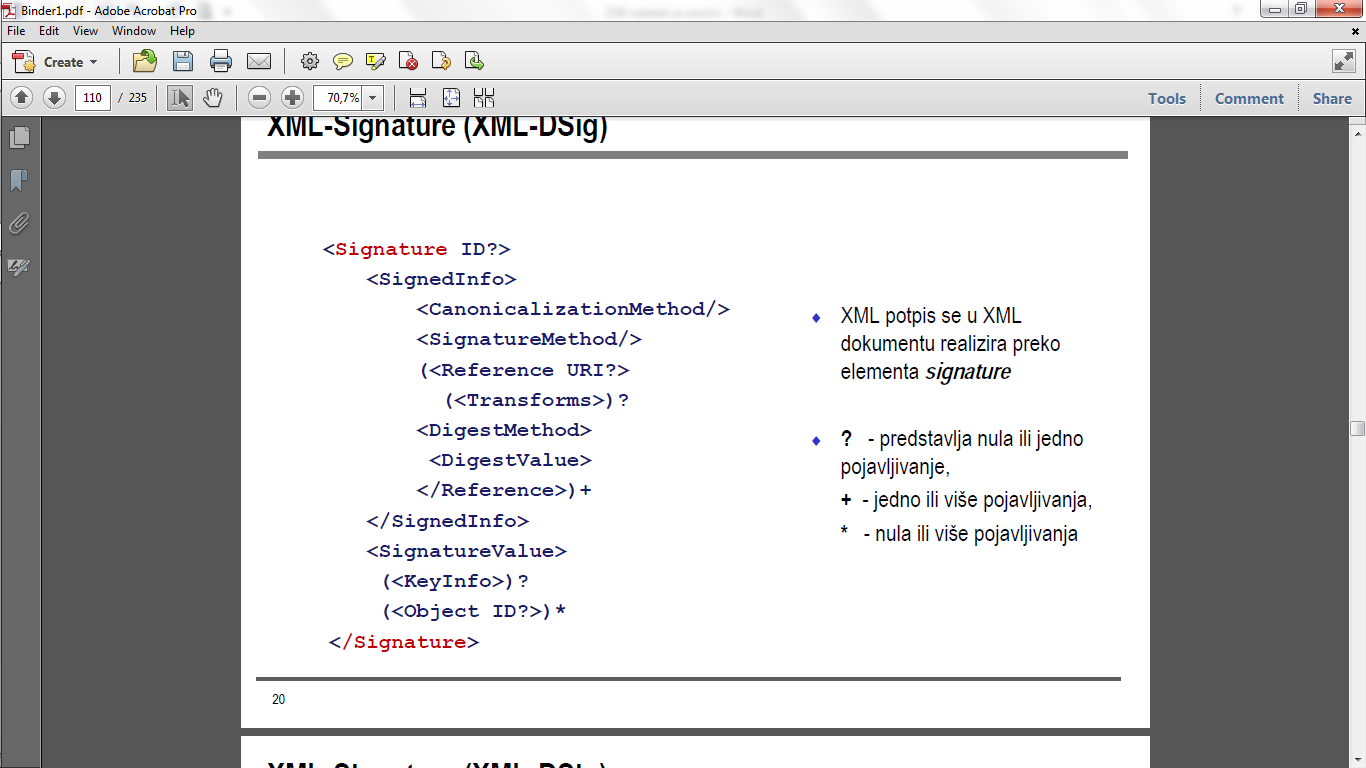
* **definira kako ugraditi digitalni potpis u XML**
* **nije** algoritam za digitalno potpisivanje
* jednim potpisom moguće je potpisati više dokumenata
* moguće je potpisati i dokumente koji nisu u formatu XML
* moguće je potpisati samo dio XML dokumenta (na taj se način omogućuje da različite dijelove jednog XML-dokumenta potpisuju različiti ljudi)

**<Signature ID?>**

**<SignedInfo>**

**<CanonicalizationMethod/> <SignatureMethod/>**

**(<Reference URI?>**

**(<Transforms>)?**

**( )**

**<DigestMethod>**

**<DigestValue>**

**R f >)+**

**</Reference>)+**

**</SignedInfo>**

**<SignatureValue>**

**(<KeyInfo>)?**

**(<Object ID?>)\***

**</Signature>**

Element *SignedInfo* – unutar svojih podelemenata **identificira podatke koji se potpisuju te različite algoritme** koji će se koristiti

*CanonicalizationMethod -* sadrži ime **algoritma kojim se radi kanonikalizacija podataka**

*SignatureValue* - sadrži vrijednost potpisa elementa *SignedInfo*

*SignatureMethod* - definira **algoritam za generiranje potpisa**

***Reference*** - **identificira resurse koji će biti potpisani i sve algoritme** koji će se koristiti **za pretprocesiranje** podataka.

Ti algoritmi su ispisani u elementu ***Transforms*** i uključuju operacije kao što su šifriranje/dešifriranje, kompresija/inflacija ili **Xpath** transformacija (XPath omogućuje potpisivanje dijela dokumenta).

**<Signature ID?>**

**<SignedInfo>**

**<CanonicalizationMethod/>**

**<SignatureMethod/>**

**(<Reference URI?>**

**(<Transforms>)?**

**<DigestMethod>**

**<DigestValue>**

**</Reference>)+**

**</SignedInfo>**

**<SignatureValue>**

**…**

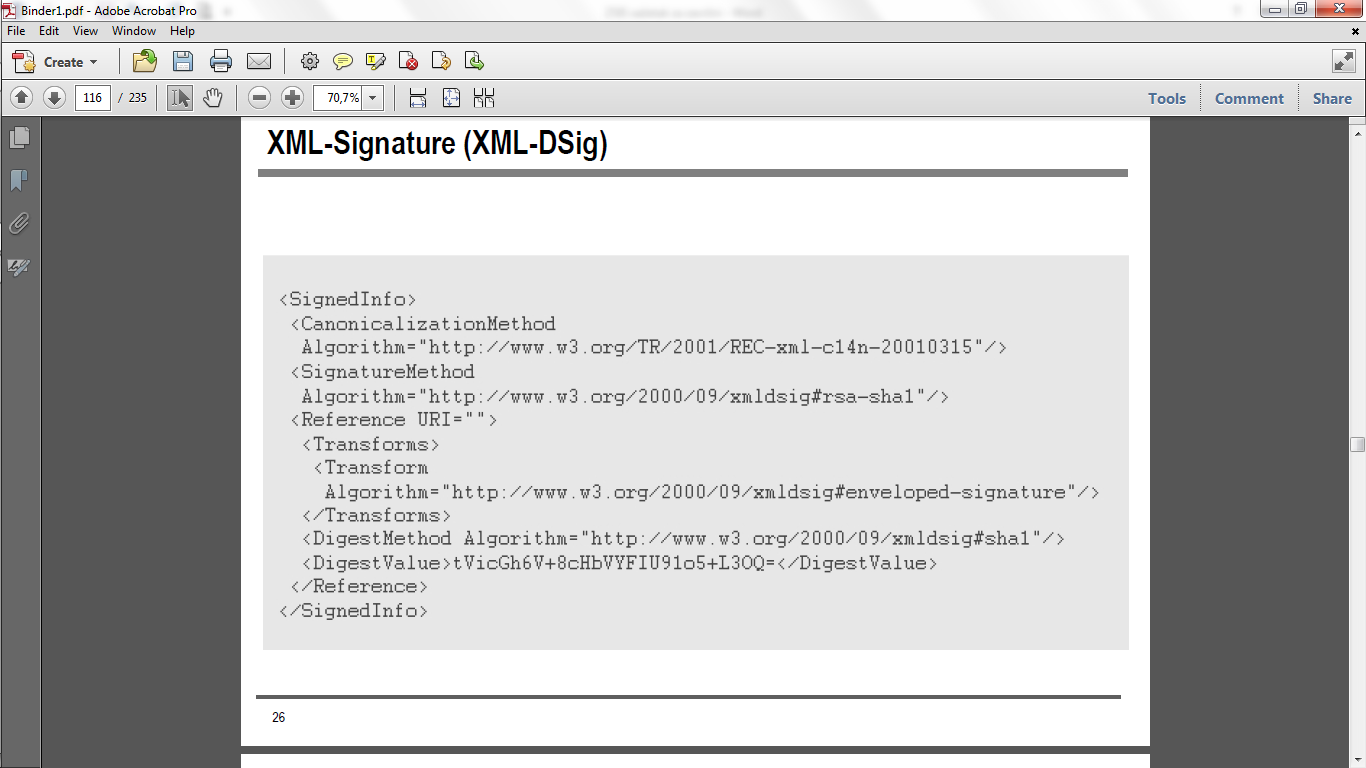
**</Signature>**

Element *Reference* ima atribut ***URI*** koji je neobavezan, ali ako potpis sadrži više elemenata *Reference* onda je URI neobavezan samo za jedan element, a ostali ga moraju imati.

Ako je sadržaj URI-ja “”, tj. prazan znakovni niz, potpisuje se dokument u kojem se nalazi element.

Svaki *Reference* uključuje :

* ***DigestMethod*** - sadrži informaciju o algoritmu koji se koristi za računanje sažetka dokumenta
* ***DigestValue*** sadrži sažetak dokumenta izračunat algoritmom navedenim u *DigestMethod*
* ***KeyInfo*** - sadrži informacije o ključu i o certifikatu



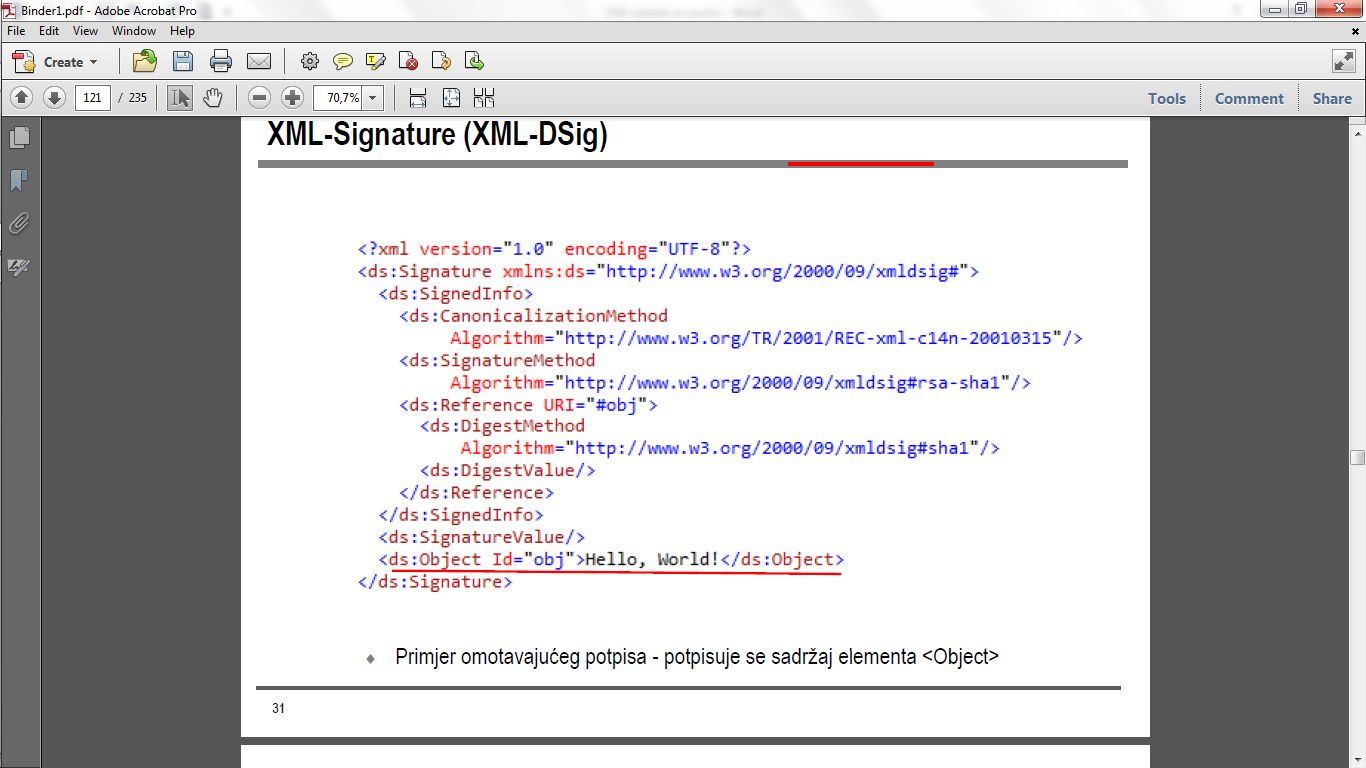
XML potpis može se pojaviti u tri osnovna oblika:

* Omotani potpis (***Enveloped***) – potpis se nalazi unutar dokumenta.
* Omotavajući potpis (***Enveloping***) – potpis omeđuje dokument koji potpisuje.
* Odvojeni potpis (***Detached***) – potpis se nalazi u zasebnom dokumentu, a URI (*Universal Resource Identifier*) određuje koji dokument potpisuje.

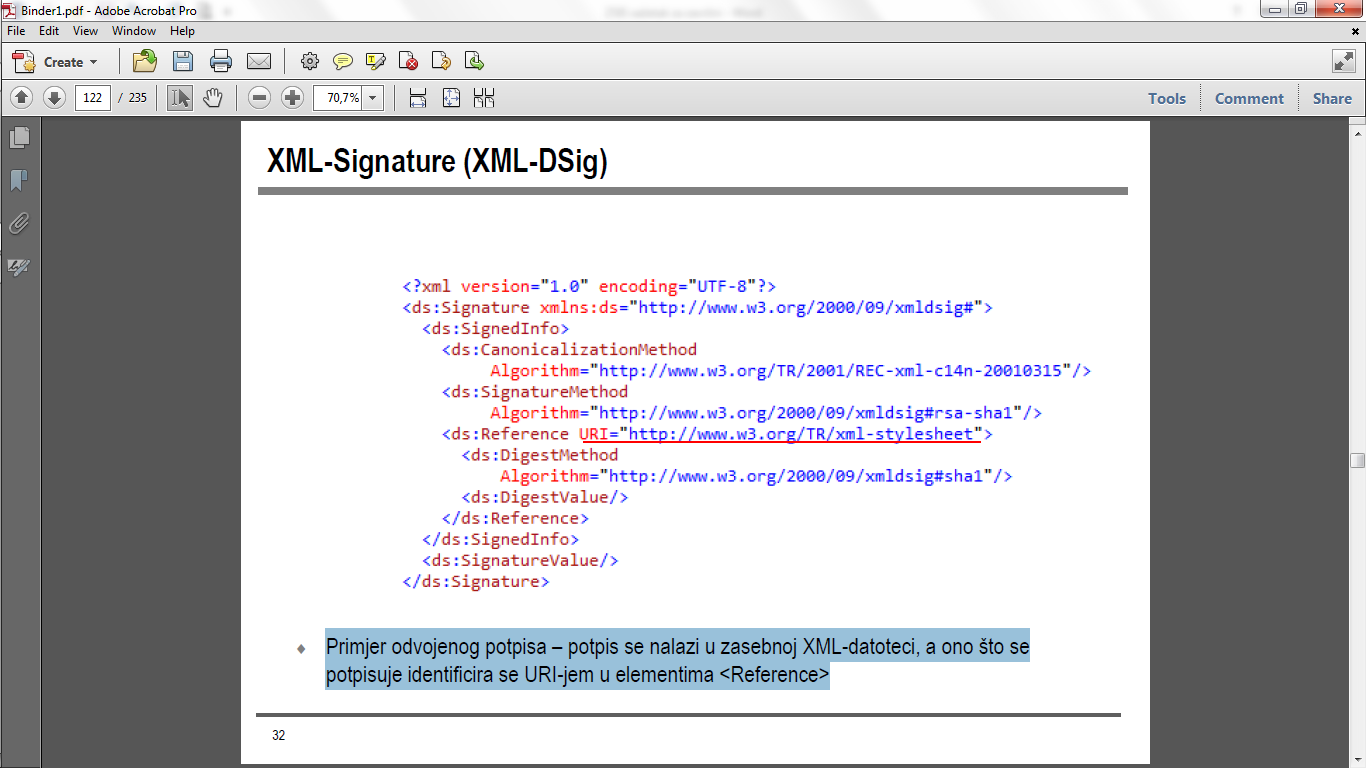
moguće je kombinacijama ta tri oblika dobiti nove

jedna od mogućih kombinacija: omotavajući potpis umetnuti u dokument tako da on potpisuje neke točno određene podatke

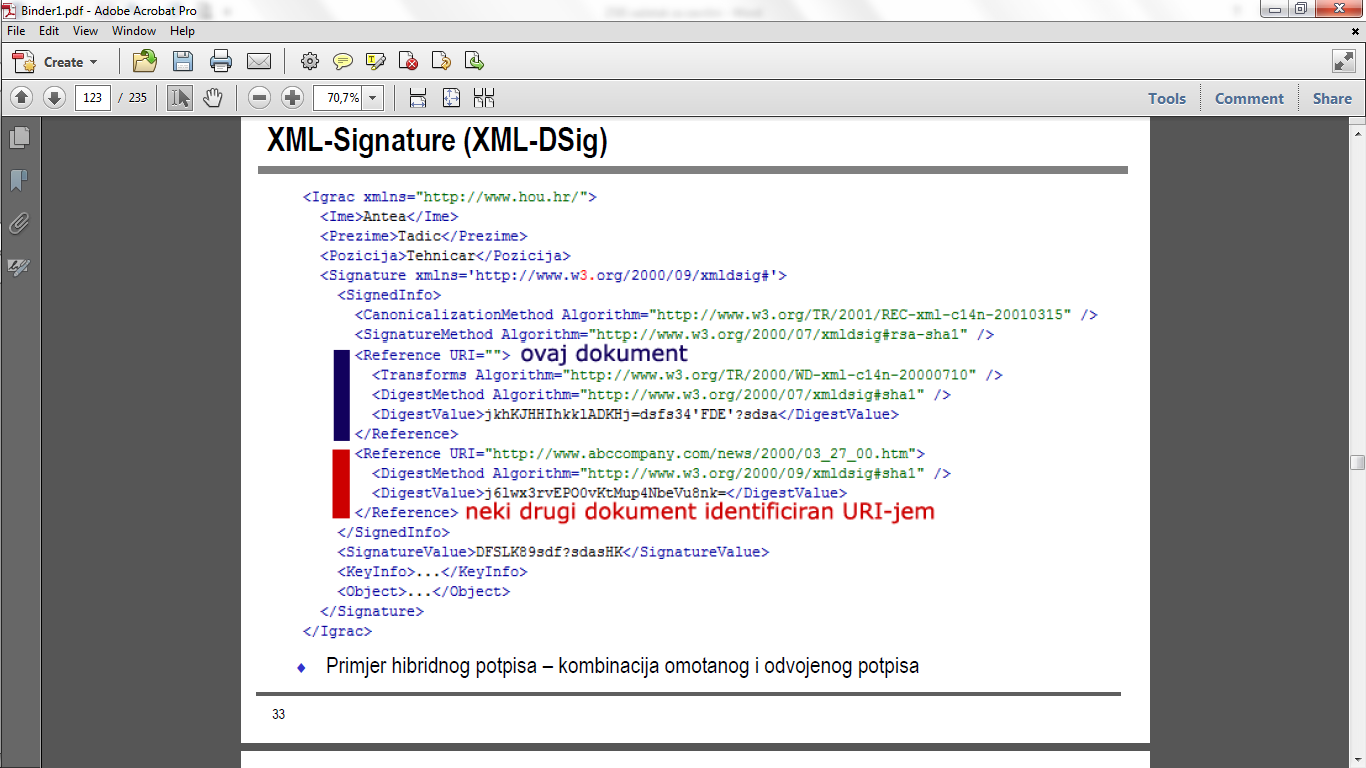
U slučaju omotavajućeg potpisa, potpisuje se sadržaj elementa *Object.*



Primjer odvojenog potpisa – potpis se nalazi u zasebnoj XML-datoteci, a ono što se potpisuje identificira se URI-jem u elementima <Reference>



Primjer hibridnog potpisa – kombinacija omotanog i odvojenog potpisa

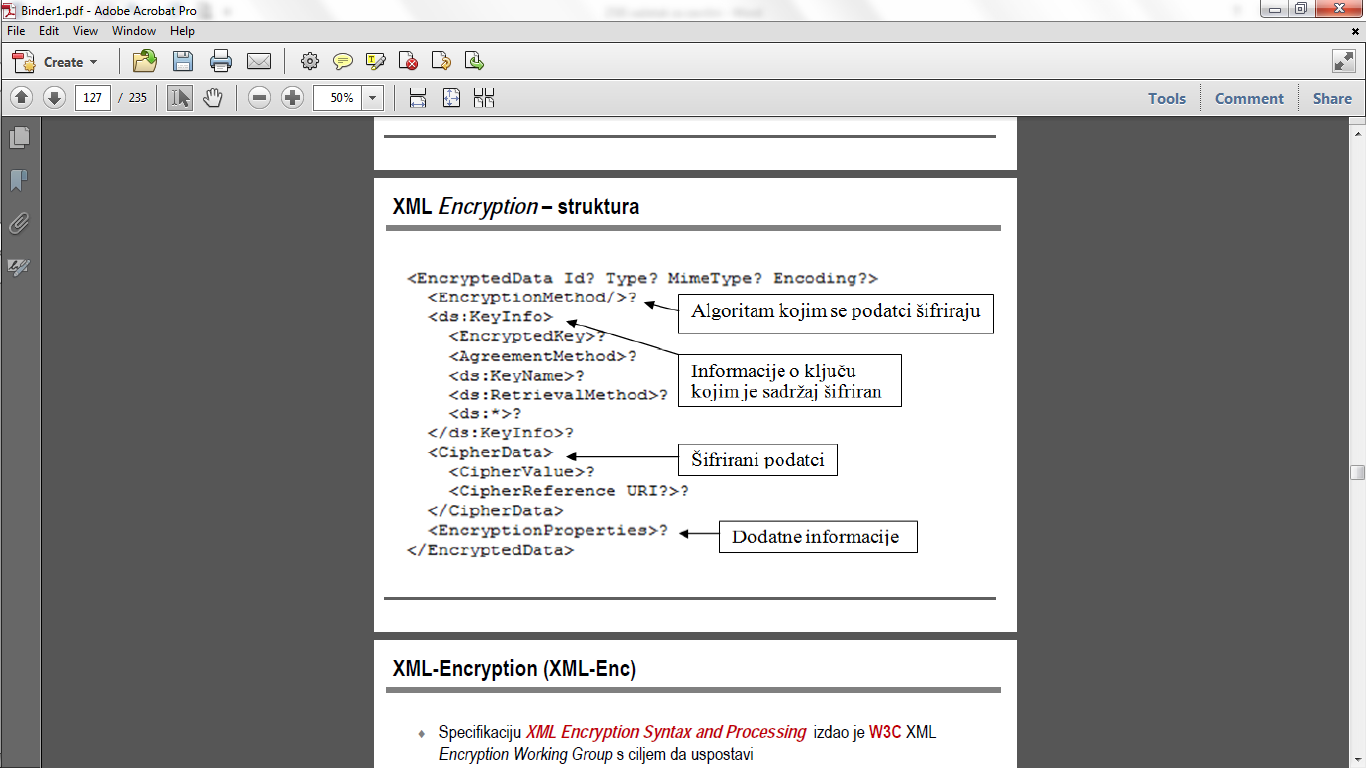


### XML Encryption (XML-Enc)

* **XML-Enc** opisuje kako šifrirani sadržaj **ugraditi u XML**
* **Nije** algoritam šifriranja
* Mogu se šifrirati i neXML-ovski dokumenti
* Moguće je šifrirati samo dio XML-dokumenta
* Različite dijelove XML-dokumenta moguće je šifrirati različitim ključevima – kontrola pristupa

Šifriranje se može izvesti na tri načina:

* korištenjem **simetrične kriptografije** – podatci se šifriraju simetričnim ključem koji su ranije sudionici komunikacije na neki (siguran) način razmijenili
* korištenjem **asimetrične kriptografije** – podatci se šifriraju javnim ključem primatelja
* korištenjem **hibridnog pristupa** – podatci se šifriraju simetričnim ključem, a taj simetrični ključ šifrira se javnim ključem primatelja; šifrirani simetrični ključ i sadržaj šifriran tim simetričnim ključem ugrađuju se u XML-dokument; ovaj je pristup najučestaliji

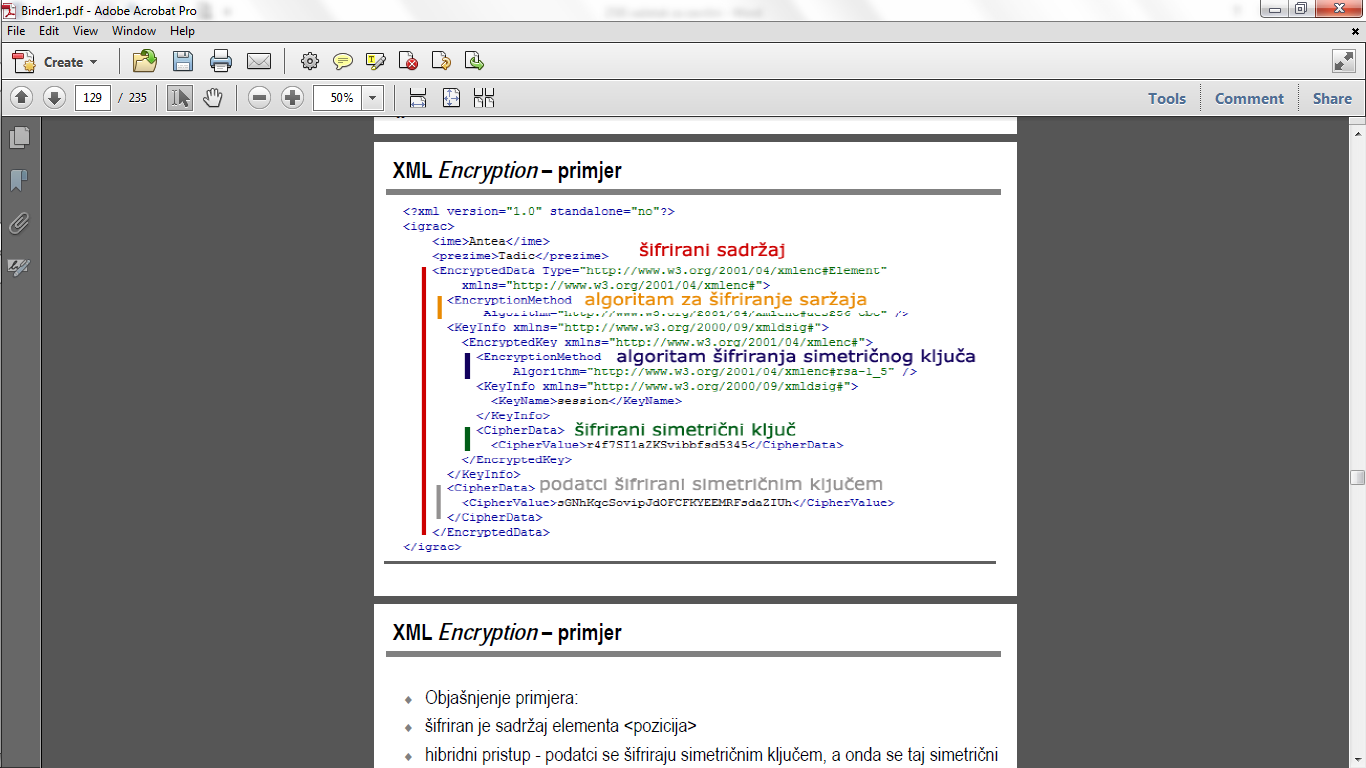


Specifikaciju ***XML Encryption Syntax and Processing*** izdao je **W3C** XML *Encryption Working Group* s ciljem da uspostavi proces šifriranja/dešifiranja digitalnih sadržaja (uključujući XML dokumente kao i njihove dijelove) i sintaksu, kako bi se prikazali:

* **šifrirani sadržaj** i
* **informacija koja omogućava** određenom primatelju **dešifriranje** primljenog sadržaja.

Rezultat šifriranja je podatkovni element koji sadrži (preko jednog od svojih elemenata djece) ili identificira (preko URI reference) šifrirane podatke.

Kad šifriramo XML element ili sadržaj elementa **šifrirani podaci** (element *EncryptedData*) **zamjenjuju element odnosno sadržaj** u šifriranoj verziji XML dokumenta.



Objašnjenje primjera:

* šifriran je sadržaj elementa <pozicija>
* hibridni pristup - podatci se šifriraju simetričnim ključem, a onda se taj simetrični ključ šifrira javnim ključem primatelja i takav se zajedno sa šifriranim sadržajem šalje na odredište
* na slici je plavom bojom označen algoritam kojim se šifrira simetrični ključ - asimetrični algoritam RSA
* zelenom je bojom prikazan šifrirani ključ
* sadržaj XML-dokumenta koji je potrebno sakriti šifrira se simetričnim algoritmom AES što je na slici prikazano narančastom bojom. Tako se dobije šifrirani sadržaj koji je na slici označen sivom bojom.
* element <EncryptedData> (crveno na slici) nalazi upravo na mjestu na kojem se prethodno nalazio element koji se šifrira

## Usluge weba

Osnovne norme za usluge weba:

* **SOAP** (engl. *Simple Object Access Protocol)-* **protokol koji definira strukturu paketa kojima se razmjenjuju strukturirani podaci** između dva udaljena programska entiteta
  + zasniva se na XML-u
  + ne uključuje sigurnosne mehanizme
* **WSDL** (engl*. Web Service Description Language*) - omogućuje **normizirani opis sučelja** usluge weba
* **UDDI** (engl*. Universal Description, Discovery and Integration) -* registar **za objavu i pretragu** postojećih usluga weba

SOAP poruke se sastoje od **tri glavna elementa**:

* **omotnica** (*envelope*) - korijenski element koji definira početak i kraj SOAP poruke
* **zaglavlje** (*header*) - nije obavezan element, a može sadržavati jedan ili više blokova metapodataka o samoj poruci
* **tijelo** (*body*) - sadrži podatke namijenjene primatelju poruke

**Usluge weba i normizacijske organizacije**

**W3C (*World Wide Web Consortium)*** je specificirao norme :

* **XML, XML Schema, XSLT**
* specifikacije za implementaciju usluga weba – norme **SOAP** i **WSDL**

**OASIS (*Organization for Advancement of Structured Information Standards) -*** normizacija jedne od najvažnijih komponenti specifikacija proširenja usluga weba – okvira **WS-Security** zaduženog **za sigurnosne mehanizme**

**WS-I (*Web Services Interoperability Organization)*** izdao je 2007. godine ***Osnovni sigurnosni profil*** *(engl.* ***Basic Security Profile****)* -skup najvažnijih tehnologija vezanih uz **sigurnost XML-a i usluga weba**

### Sigurnost usluga weba

* **jezgra normi za web usluge** - **SOAP, WSDL i UDDI** - **nije dovoljna** za uspješno ispunjavanje svih aspekata infrastrukture za e-Poslovanje.
* zahtjevi uslužno orijentiranih arhitektura nadilaze ono što skup specifikacija **SOAP+WSDL+UDDI** može pružiti
* potrebno je uvesti dodatne tehnologije kako bi se ti zahtjevi mogli uspješno zadovoljiti
* proširenja specifikacija Web usluga: **WS-Extensions**
* prepoznaju se po prefiksu "**WS-**“

***WS-Extensions***

Neka od važnijih proširenja **-** *WS-Extensions*:

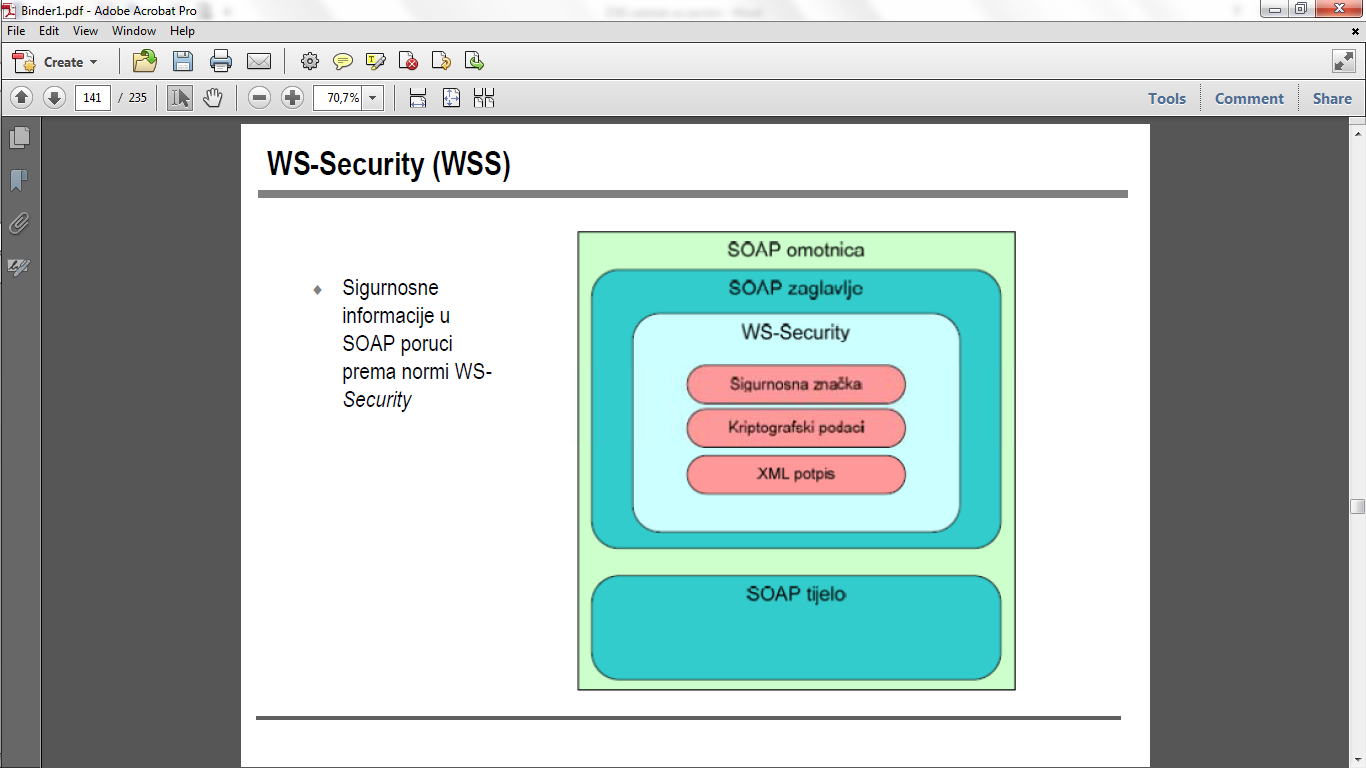
* ***WS-Security*** *–* grupa specifikacija za uvođenje sigurnosnih mehanizama koji se mogu koristiti za konstruiranje niza sigurnosnih protokola.
* **Integritet poruka** pruža ***XML Signature***, a za **osiguravanje povjerljivosti** se koristi ***XML Encription***.
* ***WS-Adressing*** *–* za imenovanje i prepoznavanje krajnjih točaka u asinkronoj komunikaciji
* ***WS-ReliableMessaging*** *–* za sigurnu dostavu SOAP poruka
* ***WS-Policy*** *–* omogućuje definiciju pravila vezanih uz sigurnost poruke, način obrade ili sam sadržaj
* ***WS-MetadataExchange*** *–* omogućuje ugradnju dodatnih metapodataka o usluzi koji bi stavili uslugu u semantički kontekst te pružali dodatne informacije kao što su kvaliteta usluge i sl.
* ***WS-Trust***
* ***WS-SecureConversation***

#### WS-Security (WSS)

**WSS** (**WS-*Security****, Web Services Security*) je komunikacijski protokol koji osigurava sigurnost primjene usluga Weba.

* razvili su ga IBM, Microsoft i VeriSign, a **OASIS** ga je objavio kao normu **WS-*Security* 1.1**.
* cilj je postizanje **sigurnosti s kraja na kraj komunikacije**, a ne samo na razini prijenosa.
* **WS-*Security*** definira **element SOAP zaglavlja** koji nosi podatke vezane uz **sigurnost**.
* **zaglavlje** može sadržavati informacije definirane normom ***XML-Signature*** koje opisuju način potpisivanja poruke, korišteni ključ i rezultat potpisa.
* kod šifriranja (kriptiranja) elemenata unutar poruke, element SOAP zaglavlja koji se odnosi na WS-*Security* sadrži informacije definirane normom ***XML-Encryption***.

WS-*Security* ne definira format potpisa ili šifriranja. Umjesto toga, specificira **način kako će se format**, definiran drugom specifikacijom, **ugraditi u SOAP poruku**.



protokol WSS sadrži i detalje o korištenju:

* jezika **SAML** (*Security Assertion Markup Language*),
* **Kerberosa** (autentifikacijski protokol) te
* digitalnih certifikata kao što su **certifikati X.509** koji se koriste u primjeni PKI infrastrukture.
* prikazuje kako pripojiti **sigurnosne tokene** na poruke u izvršavanju Web servisa
* *Kerberos* i *X.509* rješavaju problem autentifikacije

Sigurnosne informacije u SOAP poruci prema normi WS*Security*

## Sigurnost u elektroničkoj trgovini

**Sigurnosni zahtjevi kod *online* plaćanja**

* **Autentifikacija** - u transakciji online plaćanja se zna tko sudjeluje u transakciji i zna se da je osoba upravo ta za koju tvrdi da jest.
* **Integritet** - podaci iz transakcije se neće mijenjati
* **Jedinstvenost zahtjeva za plaćanjem -** omogućava trgovcu da prepozna ponovni zahtjev za istom transakcijom
* **Neporecivost transakcije** - nakon izvršavanja transakcije kupac ne može poreći da je izvršio transakciju, odnosno trgovac ne može poreći da je primio transakciju
* **Povjerljivost** – podacima o transakciji se ne može neovlašteno pristupiti
* **Privatnost i anonimnost kupca** - trgovac može vidjeti samo pseudonim ili korisničko ime kupca, ali ne i njegove privatne podatke
* **Pouzdanost sustava** - preventivne radnje u slučaju pada sustava te kod greški prilikom izvršavanja transakcije

**Kartična naplata u elektroničkoj trgovini - Sigurnost *PayPal* transakcija**

poslužitelji s osjetljivim financijskim podacima **nisu direktno povezani na Internet**

Provjera:

* *Address Verification Service* (AVS) – uspoređuju se podaci o poštanskoj adresi koju daje korisnik kreditne kartice kod izvršavanja transakcije online plaćanja ili kod povezivanja

kreditne kartice s *PayPal* računom s adresom korisnika kreditne kartice koja je zapisana

kod izdavatelja kartice

* *Card Security Code (CSC)* ili *Card Code Verification* (CCV) -troznamenkasti ili četveroznamenkasti sigurnosni kod koji se nalazi na stražnjoj strani kreditne kartice napisan obrnuto nakošeno. Kod CSC-a se uspoređuje uneseni korisnikov kod kartice s kodom kartice koji je zapisan kod izdavatelja kreditne kartice.

### PCI DSS

**PCI DSS - *Payment Card Industry Data Security Standard***

* sigurnosni standard za kartično poslovanje
* definira minimalne sigurnosne mjere i procese
* Definirao ga je *Payment Card Industry Security Standards Council*
* **Visa, MasterCard, American Express, Discover Card i JCB** su zajedno stvorili industrijski standard za sigurnost podataka kako bi zaštitili svoje korisnike.
* osigurava trgovcima, kartičnim kućama, bankama i ostalim poslovnim subjektima koji se bave kartičnim poslovanjem **zaštitu podataka vlasnika kartica**

**Banke i pružatelji usluga** moraju se **certificirati** kod kvalificiranih revizora sigurnosti, a trgovci su dužni se pridržavati PCI DSS standarda i obavljati kartično poslovanje samo s certificiranim pružateljima usluga.

PCI DSS regulira zahtjeve koji se odnose na upravljanje sigurnošću podataka, sigurnosne procedure, mrežnu arhitekturu, oblikovanje programske potpore za obradu podataka te ostale kritične zaštitne mjere u kartičnom poslovanju.

Jezgru PCI DSS-a čini skupina načela i pratećih zahtjeva oko kojih su organizirani specifični elementi sigurnosti podataka u kartičnom poslovanju.

**PCI DSS – načela i zahtjevi**

*Zahtjev 1:* **Instalirati i održavati odgovarajuću konfiguraciju vatrozida** (eng. *firewall*) **radi zaštite podataka o vlasnicima kartica.**

*Zahtjev 2:* **Ne koristiti lozinke i druge sigurnosne parametre dobivene od dobavljača softverskog sigurnosnog rješenja**

* **Promijeniti početne zaporke** postavljene od strane dobavljača

*Zahtjev 3:* **Svi pohranjeni podatci o vlasniku kartice moraju se uvijek i bezuvjetno štititi.**

* **sigurnosni kodovi kartica** (troznamenkasti ili četveroznamenkasti broj obično ispisan na

stražnjoj strani kartice) koji se koriste za potvrđivanje (verifikaciju) transakcije **i PIN brojevi ne smiju se pohranjivati**.

*Zahtjev 4:* **Tijekom prijenosa putem otvorenih, javnih mreža svi podatci o vlasniku kartice moraju se štititi šifriranjem (enkripcijom).**

* Koristiti snažne kriptografske metode i sigurnosne protokole (primjerice SSL/TLS, IPSEC, SSH) za **zaštitu osjetljivih kartičnih (korisničkih) podataka tijekom prijenosa** kroz otvorene, javne mreže (Internet, bežični prijenos, GSM i GPRS).

*Zahtjev 5*: **Nužno je koristiti i redovito osvježavati softver za zaštitu od zlonamjernog koda, odnosno antivirusni softver**

### SSL (Secure Socket Layer)

Broj kreditne kartice upisuje se preko web preglednika i putuje do web poslužitelja on-line

trgovine

* **kod elektroničkog plaćanja** potrebno je između transportnog protokola TCP i aplikacijskog protokola HTTP koristiti i **sigurnosni protokol SSL (Secure Socket Layer)**
* **SSL** osigurava **šifriranje** cjelokupne komunikacije iznad transportnog sloja.
* **protokol TLS** (eng. *Transport Layer Security*) objavljen je 1999. godine - nadogradnja na SSL 3.0

#### SSL/TLS

* koristi se za ostvarivanje sigurnije razmjene povjerljivih podataka, poput korisničkog imena i zaporke, broja kreditne kartice i sl.
* temelji se na upotrebi kriptografije te infrastrukture javnih ključeva (engl. *Public key infrastructure* - PKI)
* privatni i javni ključevi

Za kartično plaćanje preko mreže preporučuje se korištenje SSL/TLS-a sa 128- bitnom, 168- bitnom ili 256-bitnom enkripcijom.

Pri korištenju SSL/TLS-a

* adresa počinje oznakom **https://**
* sva komunikacija između preglednika i web poslužitelja se šifrira
* **HTTPS** (*Hypertext Transfer Protocol Secure) -* kombinacija protokola HTTP i SSL/TLS (*Secure Sockets Layer / Transport Layer Security)* HTTP poslužitelji “slušaju” i šalju podatke s vrata broj 80, dok HTTPS poslužitelji za to koriste vrata broj 443.

Služi za zaštitu od napada:

* prisluškivanje (eng. *eavesdropping*)
* čovjek-u-sredini (eng. *man-in-the-middle*)
* onemogućuje se presretanje i neovlašteno prisluškivanje komunikacije te eventualna krađa broja kreditne kartice
* međutim, ne rješava se problem pohrane brojeva kreditne kartice na samom poslužitelju

Provjera digitalnog certifikata poslužitelja

* poslužitelj prilikom uspostavljanja veze šalje svoj SSL/TLS certifikat klijentu
* klijentski Web preglednik provjerava autentičnost certifikata s čvorovima certifikacijskog tijela (CA - *Certificate Authority)* koji sadrže informacije o tome koji su certifikati sigurni za korištenje
* kad je CA provjerio certifikat, možemo vjerovati web poslužitelju i možemo unositi povjerljive podatke
* ne preporuča se prihvaćanje certifikata koji nisu ovjereni ili dostupni iz vjerodostojnih izvora

Kad je uspostavljena sigurna veza (certifikat zaprimljen i provjeren od strane CA), pojavljuje se ikona lokota u pregledniku i adresa počinje oznakom https://

* preglednik obično ima ugrađene sigurnosne mehanizme koji javljaju ako web sjedište nije sigurno
* neki CA su uveli SSL certifikate tipa “samo provjera domene” (*domain validation only)* za koje se radi minimalna provjera detalja u certifikatu
* za svaku uspješnu SSL konekciju – pojavljuje se ikona lokota
* **mnogi preglednici nisu jasno razlikovali certifikate s blažom validacijom od onih koji rade rigoroznu provjeru**
* korisnici nisu svjesni je li web sjedište dovoljno provjereno ili nije
* mogućnost *phishinga* – web sjedišta napravljena da bi služila za *phishing* mogu koristiti SSL da bi dobili dodatni kredibilitet

**Extended Validation Certificate** (EV) propisuje strože kriterije za provjeru identiteta

Prikazuje se ime CA koji je izdao EV certifikat - Boja (obično zelena) ukazuje na to na je EV certifikat valjan

**Phishing** - napadači pokušavaju saznati povjerljive podatke (najčešće zaporke, podatke o kreditnoj kartici ili PIN) lažno se predstavljajući kao vjerodostojan subjekt u komunikaciji.

* **lažnom porukom** elektroničke pošte ili porukom preko sustava za trenutno poručivanje korisnika se pokušava namamiti **na lažnu web stranicu**, kako bi na njoj upisao svoje korisničko ime i zaporku, PIN, broj kreditne kartice i sl.
* Npr. “*Radi provjere da Vaš račun nije neovlašteno korišten, molimo kliknite na poveznicu dolje i potvrdite svoj identitet*”
* lažne Web stranice banaka ili *online* trgovina koje vizualno izgledaju identično stvarnim stranicama
* Ako lažna stranica mimicira internetsko bankarstvo, u trenutku kada se korisnik prijavi na sustav, u pozadini ga skripta može automatski prijaviti na pravu stranicu banke, dok još vrijedi generirani OTP (one-time password). Nakon toga skripta, skriveno od korisnika, započinje prijenos novca…

Korisnik treba vjerovati HTTPS konekciji samo ako:

* Korisnik vjeruje da preglednik ispravno implementira HTTPS s ispravno unaprijed instaliranim provjerama certifikata poznatih i pouzdanih CA
* stariji preglednici nemaju instalirane certifikate
* Sjedište weba ima valjani certifikat (kojeg je potpisao CA)
* Korisnik ima povjerenje u tog CA

# Provjera sigurnosti

**Provjera ispravnosti softvera (općenito)**

* Testiranje programa, provjeravanje programa, ispitivanje programa
* otkrivanje pogrešaka odnosno nedostataka unutar programa
* uspješnost testa razmjerna je broju pronađenih pogrešaka

Prema svrsi testiranja

* Verifikacija - ovjera ispravnosti (dobra provedba)
* Validacija - potvrda valjanosti (pravi, prihvatljiv proizvod)

Prema objektu provjere

* Strukturalno (white-box testing) - izvorni kod
* Funkcionalno (black-box) – kompilat

Prema načinu provjere

* statička analiza – bez pokretanja, izvorni kod ili .NET MSIL, Java Bytecode
* dinamička analiza – pokretanjem, što ne isključuje izvorni kod

**Ključni pojmovi**

* **[„normalan” ] Test** – provjerava je li neki aspekt softvera ispravan
* **Test sigurnosti** – nastoji dokazati da neki dio ne radi kako treba
* **Pogreška** (error) – propust programera, npr. radi nerazumijevanja
  + dovodi do jednog ili više kvarova
  + razlikujemo u odnosu na "pogrešku" koja znači neželjeno stanje, tj. kvar
* **Kvar** (fault), defekt (defect), neformalno bug - neispravan dio koda
  + npr. pogrešna pretpostavka da se polje indeksira od 1 umjesto 0 izaziva kvar pristupa elementu polja
* **Zastoj** u radu (failure) – stanje izazvano jednim ili više kvarova
  + npr. prestanak rada sustava zbog kvara "buffer overrun"
* **Ispravak** (Fix) – stanje popravka

**Postupci provjere sigurnosti aplikacija**

* Nadzor (“Technical Reviews”, “Code Reviews”, "Inspections", ...)
  + testiranje nastoji izazvati zastoj, nadzor traži neispravnost
* Static Application Security Testing (**SAST**) # white-box statička
  + koja ne zahtijeva izvršenje
  + pristup izvornom kodu na klijentu i na serveru
* Dynamic Application Security Testing (**DAST**) # black-box dinamička
  + zahtijeva izvršenje
  + nema pristup izvornom kodu i pogonskoj okolini na serveru
* Interactive Application Security Testing (**IAST**) # white-box dinamička
  + dinamička s pristupom izvornom kodu i pogonskoj okolini na serveru
  + Analiza izvornog koda – statička ili dinamička s pristupom čitavom kodu



## Nadzor

**Recenzije (peer review)**

1. Inspekcija
2. Timski pregled (team review)
3. Prohod (walkthrough)

Koristi

* nalaženje defekata ranije u životnom ciklusu – do 80% prije testiranja
* nalaženje defekata s manje napora nego testiranjem
* IBM pregled 3.5 h/defekt, test 15-25 h/defekt
* nalaženje drugačijih defekata nego testiranjem – problemi dizajna i zahtjeva
* poduka razvojnika - da ne ponavljaju iste pogreške

**Inspekcija**

Formalni proces

Temeljita pokrivenost odvojenim ulogama

* Moderator – vodi sastanak, prati probleme
* Čitalac – parafrazira (prepričava) kod, nije autor
* Zapisničar – evidentira defekte
* Autor – osigurava kontekst koda, objašnjava, popravlja nakon pregleda

Kontrolne liste za specifične ciljeve

* Prikupljanje podataka za praćenje pogrešaka
* Određivanje potrebe za narednim inspekcijama
* Opsežna dokumentacija učinkovitosti
* 16-20 defekat/kLOC u inspekcije naspram 3 defekat/kLOC prohoda

**Proces inspekcije**

* Planiranje
  + autor inicira, moderator ekipira, skupa pripreme inspekcijski paket
* Priprema
  + recenzenti pregledavaju, koriste kontrolne liste i analitičke alate, označavaju defekte
* Sastanak
  + čitatelj prepričava, recenzenti komentiraju i zapitkuju, zapisničar evidentira tim zaključuje procjenu koda
* Prerada
  + autor popravlja
* Kontrola (follow-up)
  + moderator verificira korektnost promjena, autor prijavljuje kod (check-in)

**Varijante**

Timski pregled

* Timski pregled ("lagana" inspekcija)
* Osobe: moderator, recenzenti (koji nisu autori koda)
* Moduli ili manji skupovi klasa
* 1-2 sata, < 1 kLOC

Prohod (walkthrough)

* manje formalan proces
* Autor vodi sastanak i objašnjava kod
* Nedefiniran proces
* Nema kontrolnih lista ili metrike

**Drugi postupci**

Programiranje u paru

Vodič (driver) i promatrač (observer, navigator)

Zamjena uloga ali i partnera

*Peer deskcheck*

Jedan recenzent uz autora

* neformalna recenzija
* može ukIjučiti kontrolne liste i druge postupke

*Passaround* (kružno dodavanje?)

* višestruki, istodobni *Peer deskcheck*
* više recenzenata (istog koda)

## Statička provjera

SAST (Quick and Dirty)

* analiza koda bez izvršavanja
* obuhvaća sve osim testiranja

Pristupi – analizatori koda, "compiler type tools"

* *Standard compiler warnings and type checking.*
* *Lexing source checkers - look for bad names like strcpy(),gets().*
* *Parsing source code checkers.*
* *Parsing checkers with annotations.*
* *Formal proof based program verification.*

Problemi: pogrešno otkrivanje i neprepoznavanje

* **False Positives** – prijava nepostojećih bugova, nemoć pri složenom kodu
* **False Negatives** – neprepoznavanje bugova, složenost koda, slabost pravila

**Mehanizmi**

Parser, Model Builder, Analysis Engine

Parser za svaki programski jezik - generira Abstract Syntax Tree (AST)



**Prednosti i nedostaci**

Prednosti

* Potpuna pokrivenost koda (code coverage) – u teoriji
* Potencijal potvrde izostanka čitavih klasa bugova
* Hvata buhe različite u odnosu na dinamičku analizu

Slabosti

* Visok postotak pogrešnog otkrivanja
* Teško oblikovanje testa
* Složenost izgradnje (alata) – „parser za svaki jezik"
* nedovoljno kada se koriste dodatni okviri ili biblioteke
* Neimanje cjelokupnog izvornog koda u praksi (OS, shared libraries, DLLs, …)

**Alati za statičku analizu**

Simple search (lexing) tools - Flawfinder , ITS4, RATS

Parsing Tools - Fortify Source Code Analyzer, Coverity Prevent, Klocwork K7 Suite, FindBugs, splint

## Dinamička provjera

### Fuzzing – "pročešljavanje" (eng. fuzz = dlačica)

DAST (The Good, the Bad and the Ugly)

* ubrizgavanje kvara u aplikaciju (fuzzing)
* emulacijom klijenta i slanjem pripremljenih neispravnih podataka na server
* priprema neočekivanih, neispravnih ili nasumičnih podataka s ciljem rušenja
* slično regresiji, samo lošim podacima

Prednosti: - jednostavnost, nezavisnost o platformi, jeziku

Nedostaci:

* primjena na uzak skup povredivosti, pr. *Buffer overflows, Integer overflows,...*
* složena primjena na tehnologije (Web 2.0, JSON, Flash, HTML 5.0, Jscript)
* relativno dugo trajanje (permutiranja uzoraka neispravnih podataka)

**Postupci**

Glupo - Dumb (mutational) fuzzing

* dovoljno manje znanja o cilju i alatima
* pseudoslučajne anomalije ispravnih podataka
* posljedica -> potrebno više analize, redundancija nalaza

Pametno - Smart (generational) fuzzing

* zahtijeva dubinsko poznavanje cilja i specijaliziranih alata
* smišljene anomalije poznavanjem formata, standarda, ... (PDF, RFC)
* posljedica -> manja potreba za analizom

### Penetracijsko testiranje (Pen Test), etičko hakiranje

* procjena sigurnosti sustava ili mreže simuliranjem zlonamjernog napada osoba, ekipa, poželjno vanjski konzultanti
* pismena dozvola vlasnika (provedbe nezakonitih aktivnosti)

Vrste provjere prema raspoloživosti informacija

* bez dostupnih informacija (eng. *black-box test*) - kao pravi napad
* sa svim informacijama (eng. *white-box test*) - najgori slučaj, kad napadač sve zna, ili simulacija napada od strane unutrašnjeg napadača
* s djelomično dostupnim informacijama (eng. *gray-box test*) – hibrid

Kriterij početne točke testa – vanjski, unutrašnji

**Izvođenje penetracijskog testa**

**Istraživanje** (eng. *reconnaissance*)

* ispitivač pokušava prikupiti što više informacija.
* pasivno – javno dostupne informacije (npr. podaci s društvenih mreža)
* aktivno - istraživački alati (npr. *nslookup*), da bi se odredili određeni parametri.

**Skeniranje** (eng. *scanning*)

* ispitivač skenira otvorene portove (eng. *port scanning*) korištenjem alata (npr. Nmap)
* cilj - enumeracija servisa, verzije enumeriranih servisa i OS (eng. *OS and service fingerprinting*).
* skeniranje ranjivosti (eng. *vulnerability scanning*), automatiziranim alatima (npr. OpenVAS)

**Dobivanje pristupa** (eng. *obtaining access*)

* iskorištavanje ranjivosti, ručno ili alatom (npr. Metasploit),
* ovisno o dogovoru s vlasnikom, neke ranjivosti se neće iskorištavati (npr. rušenje poslužitelja)

**Zadržavanje pristupa** (eng. *maintaining access*)

* ispitivač instalira zloćudne *backdoor* i *rootkit* programe za daljnji pristup sustavu
* ova i naredna faza se u praksi najčešće ne provode ali predstavljaju scenarij realnog napada

**Brisanje tragova** (eng. *erasing evidence*)

* ispitivač pokušava izbrisati dnevničke zapise koji bi ukazivali na njihov neovlašteni pristup

# Upravljanje (softverskim) rizikom

Rizik - uvjet koji može dovesti do nekih gubitaka ili može ugroziti uspješnost projekta

Upravljanje rizikom - suočavanje s brigom prije nego što ona preraste u problem ili krizu

Upravljanje rizikom sastoji se od

* identifikacije rizika,
* odluke kako postupiti u slučaju pojedinog rizika, te
* uklanjanja rizika i rukovanja posljedicom rizika

Aktivnosti upravljanja trebaju odgovarati veličini projekta

* Mali projekti - koristeći jednostavne liste rizika
* Veliki projekti - formalno upravljanje rizikom

**Aktivnosti upravljanja rizikom**

* Procjena rizika (risk assessment)
* Identifikacija rizika - utvrđivanje liste rizika, naročito onih koji bi mogli utjecati na vremenska kašnjenja
* Analiza rizika - procjena vjerojatnosti i utjecaja pojedinog rizika, te procjena rizika za različite alternative
* Određivanje prioriteta rizika - utvrđivanje liste prioriteta rizika prema utjecaju, npr. na vremenska kašnjenja
* Kontrola rizika (risk control)
* Planiranje upravljanja rizikom - plana postupanja u slučaju pojave pojedinih rizičnih situacija
* Razrješenje rizika (risk resolution) - izvođenje plana da bi se uklonila rizična situacija koja je nastupila
* Nadziranje, praćenje rizika (risk monitoring) - nadgledanje situacija, prepoznavanje novih i njihovo uključivanje u proces upravljanja

## Prepoznavanje rizika

**Identifikacija rizika**

Opis rizika izjavama oblika uzrok-posljedica

Prati se zabrinjavajuće stanje i procjenjuje moguća posljedica

Na primjer,

* rizik se može smatrati stanjem: «korisnici se ne slažu s zahtjevima postavljenim na proizvod»
* ili posljedicom: «moguće je zadovoljiti samo najvažnije korisnike».
* kombiniranje izjava u uzročno-posljedičnu formulaciju: «Budući da se korisnici ne slažu oko zahtjeva postavljenih na proizvod, moguće je zadovoljiti samo najvažnije korisnike».

Jedan uvjet može dovesti do nekoliko posljedica, a nekoliko uvjeta može doprinijeti istoj posljedici.

Neki od najčešćih rizika:

* Rizici rokova, planiranja, organizacije i upravljanja, razvojnog okruženja..
* Rizici krajnjeg korisnika, naručitelja, (pod)ugovaratelja….
* Rizici zahtjeva, aplikacije, vanjskih utjecaja, razvojne ekipe….

**Rizici vremenskih rokova**

* Neplansko dodavanje karakteristika koje narušavaju dizajn
* Nepotrebno usavršavanje pojedinih dijelova aplikacije
* Gubitak kvalitete radi sustizanja vremenskih rokova
* Preoptimistični vremenski rokovi razvoja
* Neprikladan dizajn programske podrške
* Slab razvojni tim
* Razilaženje između korisnika i razvojnog tima

Najjednostavniji način evidentiranja rizika

* liste rizika poredanih prema utjecaju na kašnjenje

**Rizici planiranja (vremenskog rasporeda)**

* Vremenski plan, resursi i definicija proizvoda nisu usuglašeni.
* Vremenski plan je preoptimističan (kao u *najboljem slučaju*).
* Vremenski plan ne obuhvaća sve postupke koje treba provesti.
* Plan se radi uz pogrešnu pretpostavku o sastavu razvojne ekipe.
* Aplikacija je veća od predviđenog (npr. LOC, FP, u odnosu na surogate).
* Potreban napor veći je od predviđenog (npr. po prog. retku, funkciji, modulu).
* Vremenski zaostatak jednog dijela uzrokuje zaostajanje zavisnih dijelova.
* Nepoznati dijelovi aplikacije zahtijevaju više vremena nego što je predviđeno

**Rizici organizacije i upravljanja**

* Uprava ili više rukovodstvo ne podržavaju projekt.
* Odlazak zaposlenika smanjuje kapacitet razvojnog tima.
* Menadžment i marketing inzistiraju na odlukama koje produljuju rok završetka.
* Neučinkovitost razvojnog tima smanjuje produktivnost.
* Sporo donošenje odluka.
* Smanjenje proračuna narušava planove za razvoj projekta.
* Donošenje odluka koje smanjuju motivaciju članova ekipe.
* Netehnički dijelovi projekta traju dulje od očekivanog (npr. odobrenje, nabava).
* Loše vođenje projekta i loš nadzor nad napretkom projekta.

**Rizici razvojnog okruženja**

* Oprema ne dolazi na vrijeme.
* Oprema je dostupna, ali neprikladna (npr. Internet, uredska oprema).
* Oprema je nagurana, bučna ili ometa rad.
* Razvojni alati ne odgovaraju tipu problema koji se rješava.
* Razvojni alati nisu dostupni u potrebno vrijeme.
* Razvojni alati ne rade prema očekivanjima.
* Preduga krivulja učenja novih razvojnih alata.
* Istek licenci.

**Rizici krajnjeg korisnika**

* Rizici vezani uz krajnjeg korisnika
* Korisnici inzistiraju na novim zahtjevima.
* Korisnici zahtijevaju redizajn aplikacije.
* Korisnici ne pružaju potrebne informacije.
* Rizici vezani uz korisničke zahtjeve
* Nerazumijevanje zahtjeva.
* Nepotvrđeni zahtjevi., neprovjereni zahtjevi.
* Prešućeni zahtjevi, promjene zahtjeva.

**Rizici naručitelja, rizici ugovaratelja, vanjski utjecaji**

* Rizici vezani uz naručitelja
* Spor odziv naručitelja (prema planovima i specifikacijama).
* Klijent ne želi ili nije sposoban sudjelovati u odlučivanju.
* Klijent zahtijeva tehnička rješenja koja produljuju trajanje.
* Klijent upravlja razvojnim procesom.
* Nekvalitetne ili nekompatibilne komponente koje nabavlja klijent samostalno.
* Klijent izbjegava primopredaju, iako programska podrška ispunjava sve specifikacije.
* Klijent zahtijeva neprovedivu brzinu razvoja.
* Rizici vezani uz ugovaratelja ili podugovatelja
* Ugovaratelj kasni s isporukom komponenti.
* Isporučene komponente su nekvalitetne pa ih treba poboljšati.
* Kontraktor je neprofesionalan ili nedovoljno angažiran.
* Rizici vezani uz vanjske utjecaje
* Razvoj ovisi o pravnim propisima koji se nepredviđeno mijenjaju.
* Razvoj ovisi o tehničkim standardima, koji se nepredviđeno mijenjaju.

(….)

## Analiza rizika i određivanje prioriteta rizika

* Nakon utvrđivanja liste rizika projekta
* Analiza svakog rizika pojedinačno
* Utvrđivanje utjecaja na projekt

Primjena

* odabir između nekoliko razvojnih opcija ili
* utvrđivanje rizika već odabrane razvojne opcije

### Dokumentiranje rizika

Predložak za dokumentiranje pojedine izjave o riziku

**ID**: Jedinstveni identifikator

**Datum otvaranja**: Datum kada je rizik identificiran

**Datum zatvaranja**: Datum kada je rizik zatvoren

**Opis**: Opis rizika u obliku «uvjet-posljedica»

**Vjerojatnost**: Vjerojatnost da će rizik postati problem

**Učinak**: Potencijalna šteta ako se problem ostvari

**Izloženost**: Vjerojatnost \* učinak

**Plan razrješenja**: izbjegavanje, smanjenje, transfer, prihvaćanje rizika

**Nositelj**: Osoba odgovorna za razrješenje rizika

**Rok**: Datum do kojeg plan ublaživanja mora biti završen

Umjesto strukturiranog dokumenta - tablica s listom rizika

### Vrednovanje rizika

Rizik kao "neočekivani gubitak"

* **Vjerojatnost** gubitka se kreće u rasponu od 0.01 do 1.0 (do 100%)
* **Veličina** gubitka, učinak
  + zanima nas vremenski raspored – izraženo u danima/tjednima/mjesecima
* **Izloženost**, utjecaj (risk exposure, risk impact)
  + Kad nas zanima raspored, računa se (skalira) kašnjenje
  + Izloženost = Vjerojatnost \* Učinak

Primjer: vjerojatnost 25% da će nešto trajati 4 tjedana dulje – izloženost 1 tj.

Ponekad nije potrebno precizno kvantificirati rizik.

Vjerojatnost i učinak mogu biti *visoko, srednje* ili *nisko*.

### Procjena veličine gubitka

Obično je lakše procijeniti veličinu gubitka od vjerojatnosti pojave

Kada nije jednostavno izravno procijeniti veličinu gubitka moguće gubitke podijeliti u manje,

te procijeniti njihovu veličinu, a zatim agregirati pojedinačne procjene podgubitaka.

Primjerice, ako se koriste tri nova programska alata, za svaki alat zasebno procijeniti veličinu gubitka, a zatim zbrojiti previđanja za pojedine alate.

### Procjena vjerojatnosti gubitka

Procjena vjerojatnosti obično subjektivna - postupci za povećanje točnosti

Najupućenija osoba procijeni vjerojatnost svakog pojedinačnog gubitka

*Delphi* ili neki drugi postupak kojim se postiže konsenzus

* Svaki član grupe zasebno procjenjuje svaki rizik
* Diskutiraju se (argumentiraju) procjene, naročito ekstremne
* Postupak procjene se ponavlja do konvergencije

Metoda klađenja

* Npr. "Ako dodaci budu gotovi na vrijeme dobit ćete 125kn, inače ja dobivam 100kn"
* Oklada se prepravlja sve dok obje strane ne budu zadovoljne
* Vjerojatnost rizika je rezultat dijeljenja dobitka ponuditelja oklade i ukupnog iznosa.
* Za navedeni primjer, vjerojatnost = 100 kn / (100 + 125) kn = 44%.

Postupak pridjevne kalibracije ("adjective calibration")

* odredi se razina rizika opisno (npr. vrlo vjerojatno, vjerojatno, ... malo vjerojatno)
* zatim se opisne procjene kvantificiraju [Boehm 1989].

### Vremenski gubici cijelog projekta i vremenske zalihe

Izloženost riziku je *očekivana vrijednost vremenskih gubitaka*

Statistički, očekivani gubitak je umnožak vjerojatnosti i veličine gubitka

U primjeru, gubitak zbog neprikladnog dizajna = 15%\*15t = 2.25 tjedna

Ukupni gubici prije poduzimanja koraka za upravljanje rizikom zbrajanjem pojedinačnih gubitaka, za primjer u tablici 12.8 do 13.2 tjedna

Vremenski plan treba prilagoditi očekivanim vremenskim gubicima

* nakon izrade plana upravljanja rizikom
* postaviti očekivane vremenske gubitke kao vremensku rezervu projekta
* alternativno, vremenski plan s +/-odstupanjima za svaki rizik
* ažurira se vremenski plan svaki put kad se neki rizik ostvari

### Utvrđivanje prioriteta rizika

Postavljanje prioriteta rizika – usmjeravanje upravljanja

U projektima se obično 80% budžeta troši na ispravljanje 20% problema, pa je zato neophodno usredotočiti se na 20% najvažnijih [Boehm 1989].

"S obzirom da je bezuspješno pokušati eliminirati rizik, a upitno minimizirati ga, ključno je da svaki preuzeti rizik bude onaj *pravi*" [Peter Drucker]

Jednostavnije je usredotočiti se samo na vremenske rizike, nego na sve vrste rizika odjednom!

Trivijalno, silaznim sortiranjem prema izloženosti

### Točnost procjene i zanemarivanje rizika

Poredak rizika prema prioritetu je samo aproksimacija jer su svi podaci koji se koriste samo *procjene*.

Točnost prioriteta zavisi o točnosti procjena vjerojatnosti i veličina

Pretvorba procjena u brojeve stvara dojam da je lista prioriteta točna, iako ne može biti točnija od subjektivnih podataka na temelju kojih je dobivena!

Zanemarivanje rizika

* Nema smisla trošiti vrijeme na rizike koji nose male gubitke
* Da se ne bi više potrošilo na bavljenje rizikom nego što iznosi njegov gubitak

## Kontrola rizika

* Planiranje upravljanja rizikom - plana postupanja u slučaju pojave pojedinih rizičnih situacija
* Razrješenje rizika (risk resolution) - izvođenje plana da bi se uklonila rizična situacija koja je nastupila
* Nadziranje, praćenje rizika (risk monitoring) - nadgledanje situacija, prepoznavanje novih i njihovo uključivanje u proces upravljanja
* Razrješenje rizika (risk resolution)
* Avoidance (eliminate, withdraw from or not become involved) - izbjegavanje
* Sharing (transfer – outsource or insure) – dijeljenje / prijenos
* Reduction (optimize – mitigate) – smanjenje, redukcija
* Retention (accept and budget) – zadržavanje, prihvaćanje

**Plan upravljanja rizikom i razrješenje rizika**

* Radi se plan djelovanja za svaki utvrđeni visoki rizik
* Plan može biti samo izjava „**tko, što, gdje, kada, zašto i kako**” postupiti
* Plan treba sadržavati opće odredbe za nadzor rizika, zatvaranje rizika koji su riješeni i identifikaciju novih rizika.

Razrješenje rizika - ovisi o posebnostima pojedinog rizika

Na primjer,

* Pr.1: rizik neodgovarajućeg dizajna u neistraženom problemskom području
* Pr.2: rizik "gubitka" radnog prostora, preseljenjem radi druge ekipe

**Opći postupci razrješenja rizika**

Izbjegavanje rizika - ne preuzeti rizik ili ukloniti uzrok

* Pr.1.a: preuzeti odgovornost samo za dizajn sustava koji je poznat, a dizajn nepoznatog dijela prepustiti klijentu ?!
* Pr.1.b: promijeniti doseg projekta, tako da npr. problematični dizajn postane dio neke druge verzije ili zasebnog „istraživačkog” projekta
* Pr.2.a: nagovoriti grupu koja pretendira na prostor da odustane (potpuno)
* Pr.2.b: navesti konkurenciju da se preseli u neki drugi prostor

Preusmjeravanje rizika - rizik u jednom dijelu nije rizik u nekom drugom

posljedice i/ili upravljanje prenesu se u drugi dio projekta ili na treću stranu

* Pr.1.a: najam usluge (outsourcing) rizičnog dijela
* Pr.1.b: prijedlog klijentu da se uključi/revidira dizajn - preuzme dio odgovornosti
* Pr.2.a: prijedlog da druga grupa zamijeni radni prostor
* Pr.2.b: pristati na premještaj ali uz odgodu do boljeg trenutka ili kraja projekta

Kontrola rizika – smanjenje rizika

Prihvatiti mogućnost rizika i razviti rezervni plan

* Pr.1: osigurati dovoljno članova za testiranje loše projektiranog sustava, planirati dodatno vrijeme za ispravak pogrešaka
* Pr.2: ako je selidba neizbježna, treba ju provesti u trenutku kada najmanje utječe na rad i organizirati pomoć pri pakiranju i selidbi

Preuzimanje rizika

* Prihvatiti mogućnost da se rizik može dogoditi i ne činiti ništa
* Prikladno ako su posljedice male, a napor izbjegavanja velik

Prikupljanje informacija o riziku

* Ako se ne zna koliko je rizik ozbiljan, treba ga istražiti.
* Pr.1: prototip za test izvedivosti, ili vanjska evaluacija dizajna
* Pr.2: suradnja s organizatorom premještaja – zamjenski prostor

Objavljivanje rizika

* Upoznati dionike s rizikom i posljedicama – uprava, korisnici, …
* minimizirati njihovo iznenađenje u slučaju da se rizik dogodi

Evidencija rizika

* zbirka planova o djelovanju u rizičnim situacijama, što se može iskoristiti u budućim projektima

**Nadgledanje rizika**

Nestabilnost rizika

* rizici se pojavljuju, povećavaju/smanjuju, nestaju s vremenom
* trajno nadgledanje i mjerenje
* „lista najvećih 10” (Top 10)
* jedna od najboljih strategija za nadgledanje
* ne nužno točno 10 rizika
* sadržaj - status rizika, broj pojavljivanja, koraci od prethodnog ažuriranja
* ažuriranje jednom tjedno (ili prema iteraciji životnog ciklusa projekta)
* najvažniji aspekt - osiguranje redovitog uvida, redovno razmišljanje o rizicima i uzbunjivanje u slučaju promjena u važnosti rizika

## Ostala razmatranja

Napraviti sažetak rizika nakon svake veće etape projekta

Rezime po završetku projekta koristi tek u narednom!

Djelatnik zadužen za praćenje rizika (risk officer)

* korisno je imati osobu koja će stalno tražiti razloge zbog kojih projekt može propasti.
* veliki projekti (50 ili više članova) - puno radno vrijeme
* mali projekti – dodatna uloga člana ekipe
* ne smije biti voditelj projekta!