



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 licencom 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

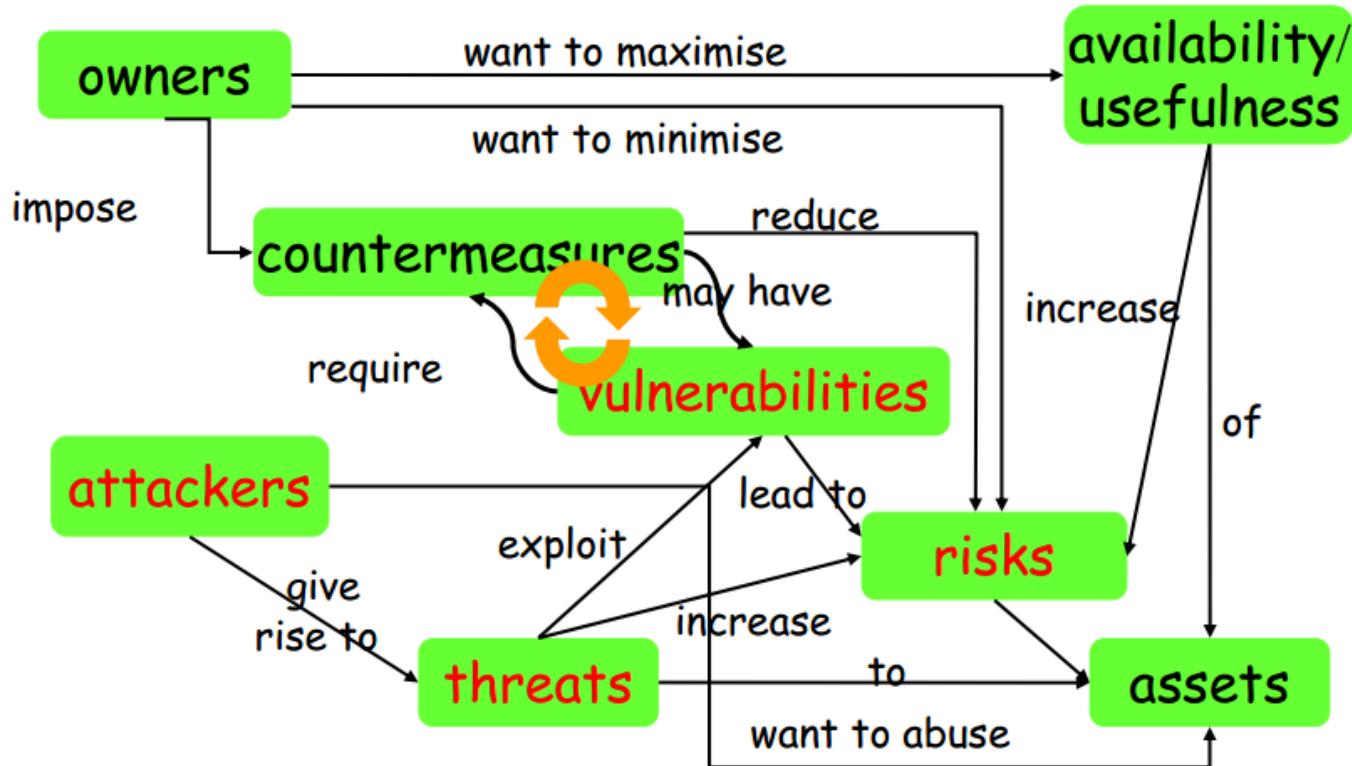
- ◆ Sigurnost programske podrške (software security)
 - Inženjerstvo softvera koji će pri napadu nastaviti ispravno raditi
 - *the science and study of protecting software (including data in software) against unauthorized access, modification, analysis or exploitation*
 - Software security = risk management
 - 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!

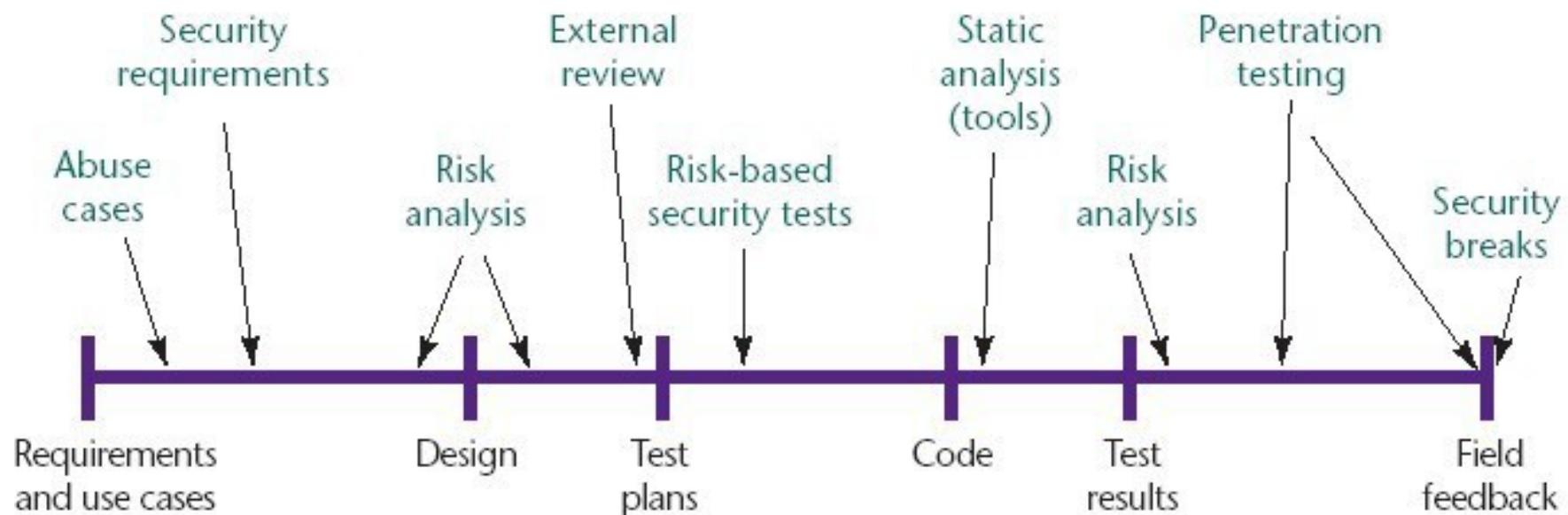
Životni ciklus sigurnog softvera

Secure Software Development Life Cycle

Životni ciklus sigurnog softvera

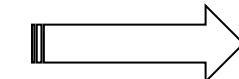
- ◆ Postupci, tehnike i metodologije
 - Sigurnost u životnom ciklusu
 - Inženjerski i projektantski principi
 - Sigurnosne tehnologije
- ◆ Pojednostavljen:

[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, ...

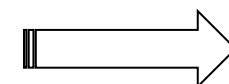
Phase One: Requirements

- ◆ 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:**

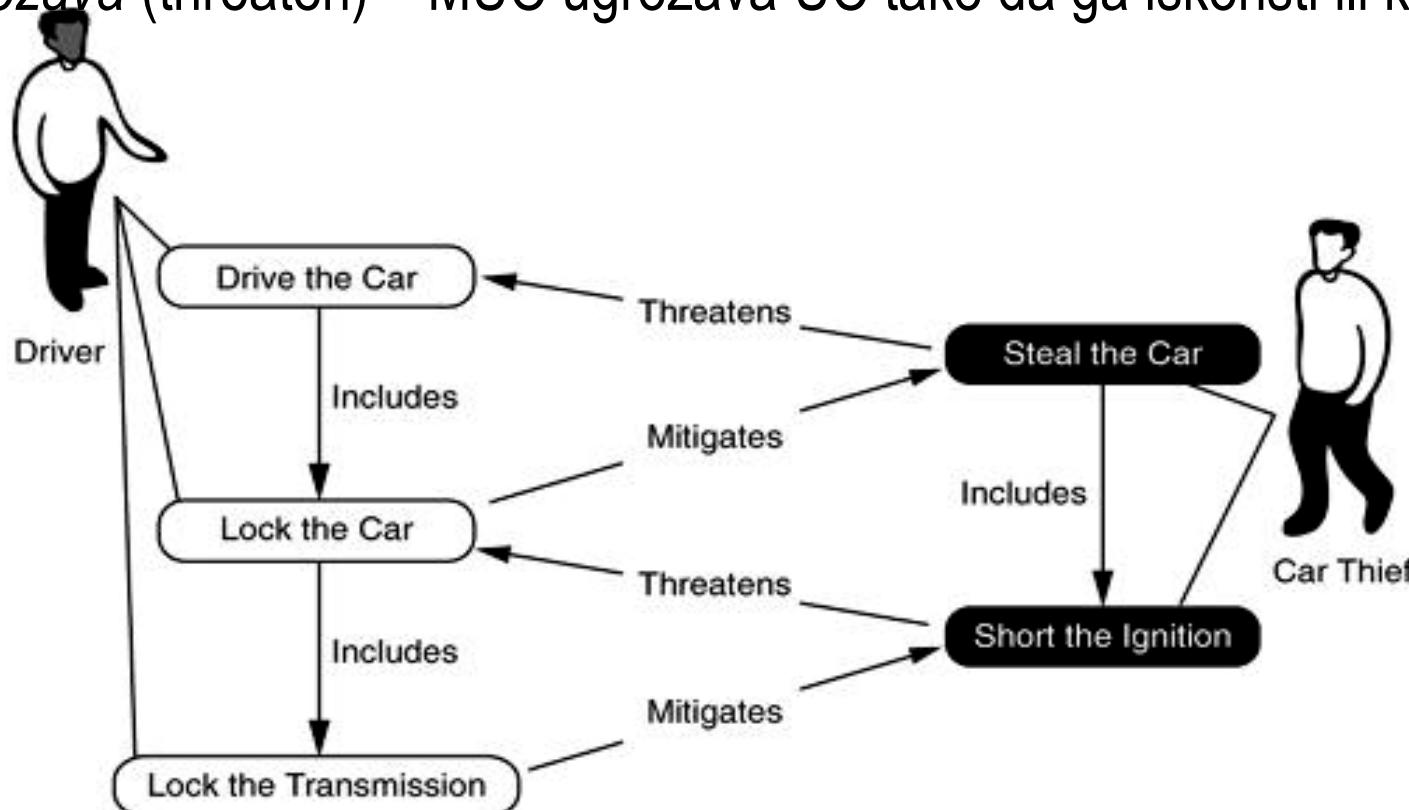
1. Napadač instalira mrežno njuškalo
2. Njuškalo sprema pakete koji sadrže "Logon", "Username", "Password"
3. Napadač čita dnevниke njuškala
4. Napadač nalazi ispravan *login / password*
5. Napadač koristi nađeni *login / password* za pristup sustavu

- ◆ **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

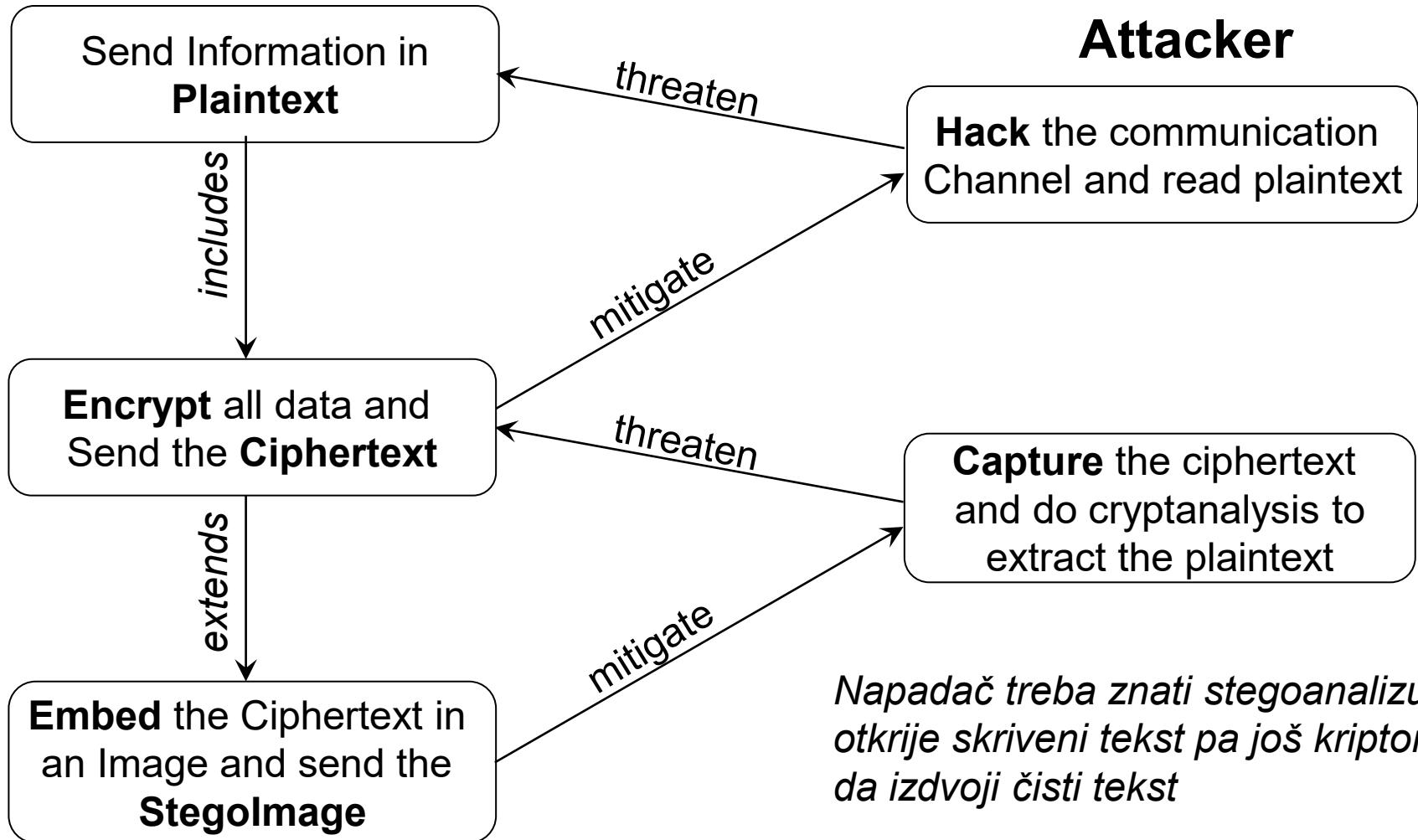
Povezivanje slučajeva zloporabe

- ◆ 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.



Zaštita i sigurnost informacijskih sustava

Projektiranje sigurnosti

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 licencom 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/>.

Modeliranje prijetnji

Threat Modeling

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

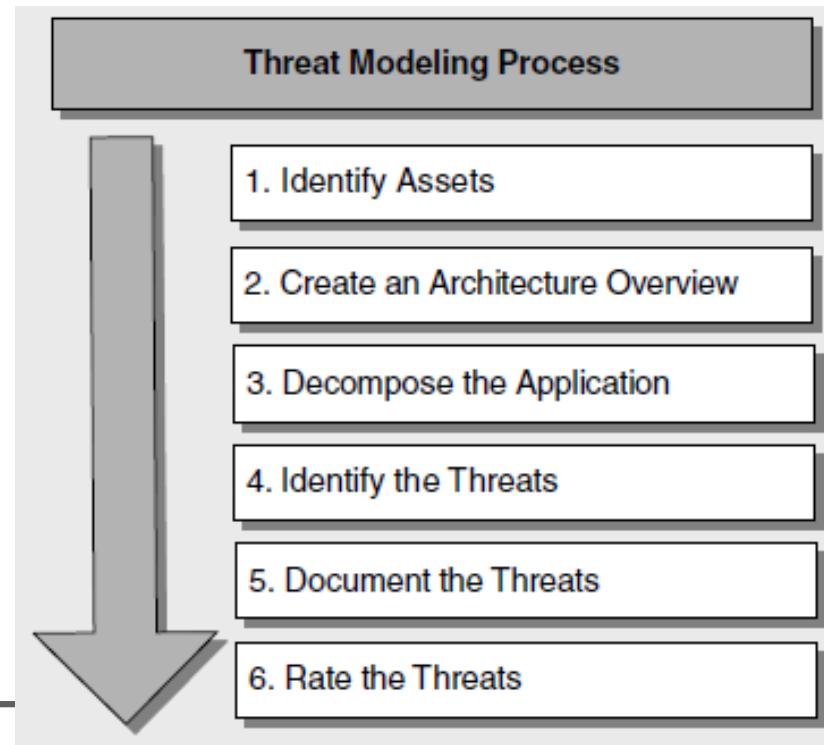
- bitno je da se obavi kvalitetno
- najbolje iterativno

važno:

- Jednostavnije je pronaći sigurnosni propust u dizajnu aplikacije nego mijenjati kasnije
- Model prijetnji treba biti aktualan (ažuran) - prijetnje i načini kako ih zaobići

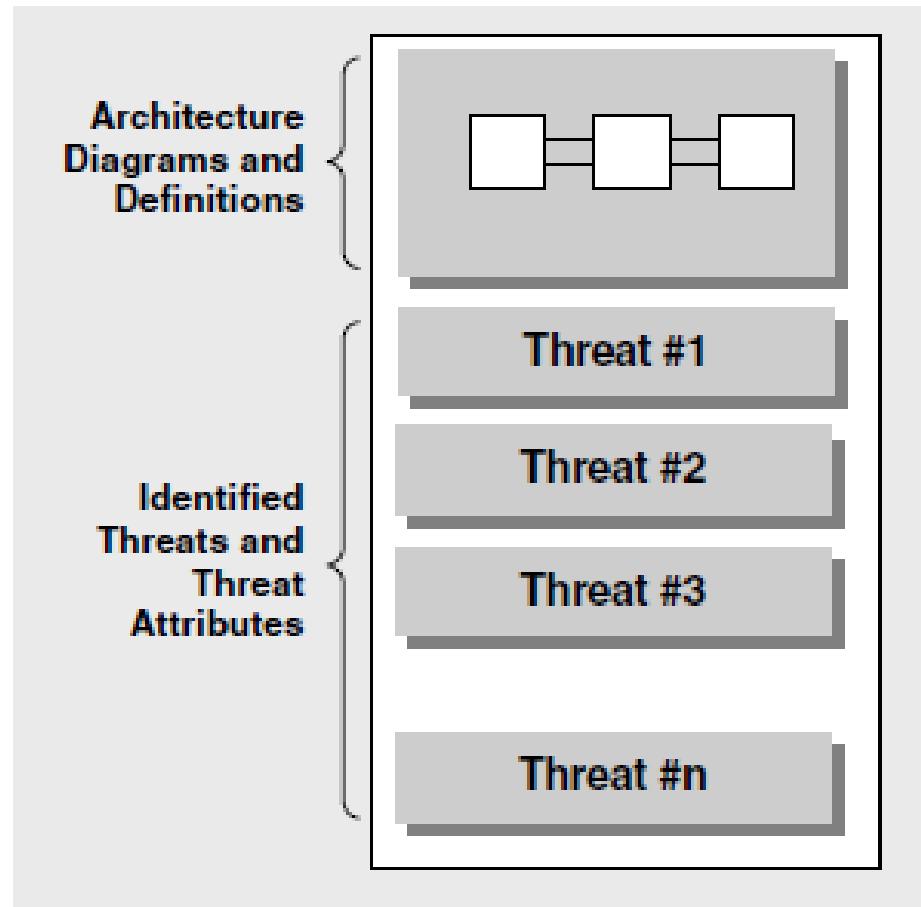
- ◆ Proces modeliranja prijetnji

- Određivanje ciljeva zaštite
- Arhitektura aplikacije
- Dekompozicija aplikacije
- Određivanje prijetnji
- Dokumentiranje prijetnji
- Rangiranje prijetnji

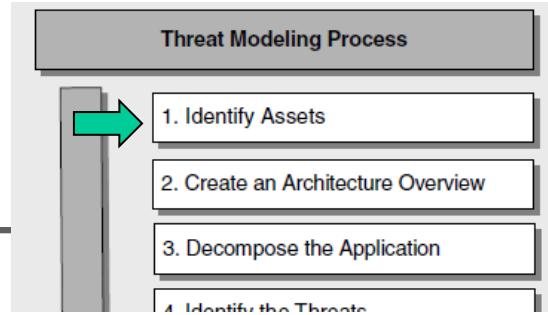


Izlaz

- ◆ Dokument s modelima
 - definicijom arhitekture i
 - popisom prijetnji

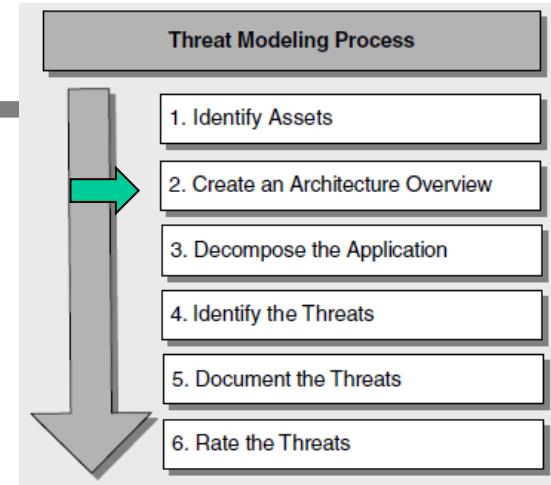


- ◆ Korak 1 – identifikacija resursa koje treba zaštititi
 - od spremišta podataka (datoteka, BP), ..., do web stranica



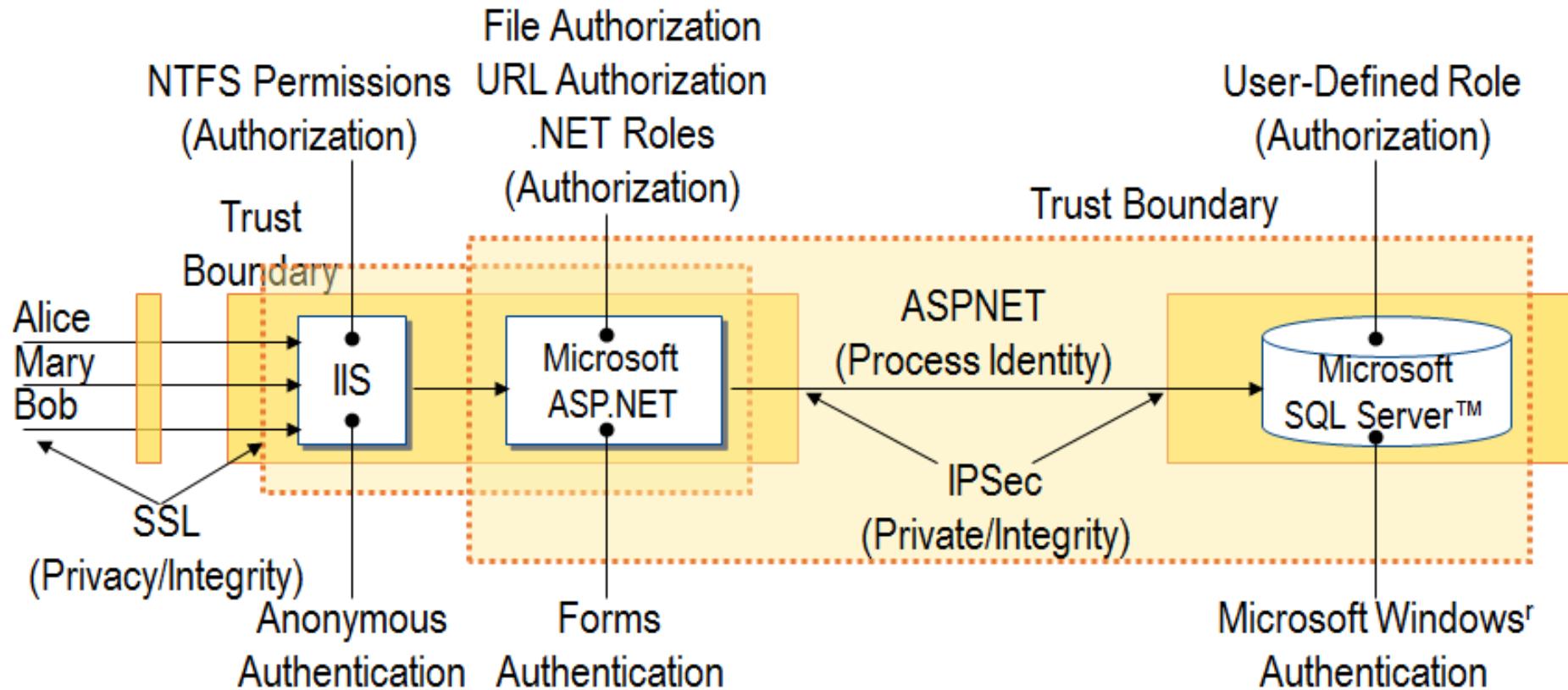
Korak 2 – Pregled arhitekture

- ◆ 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



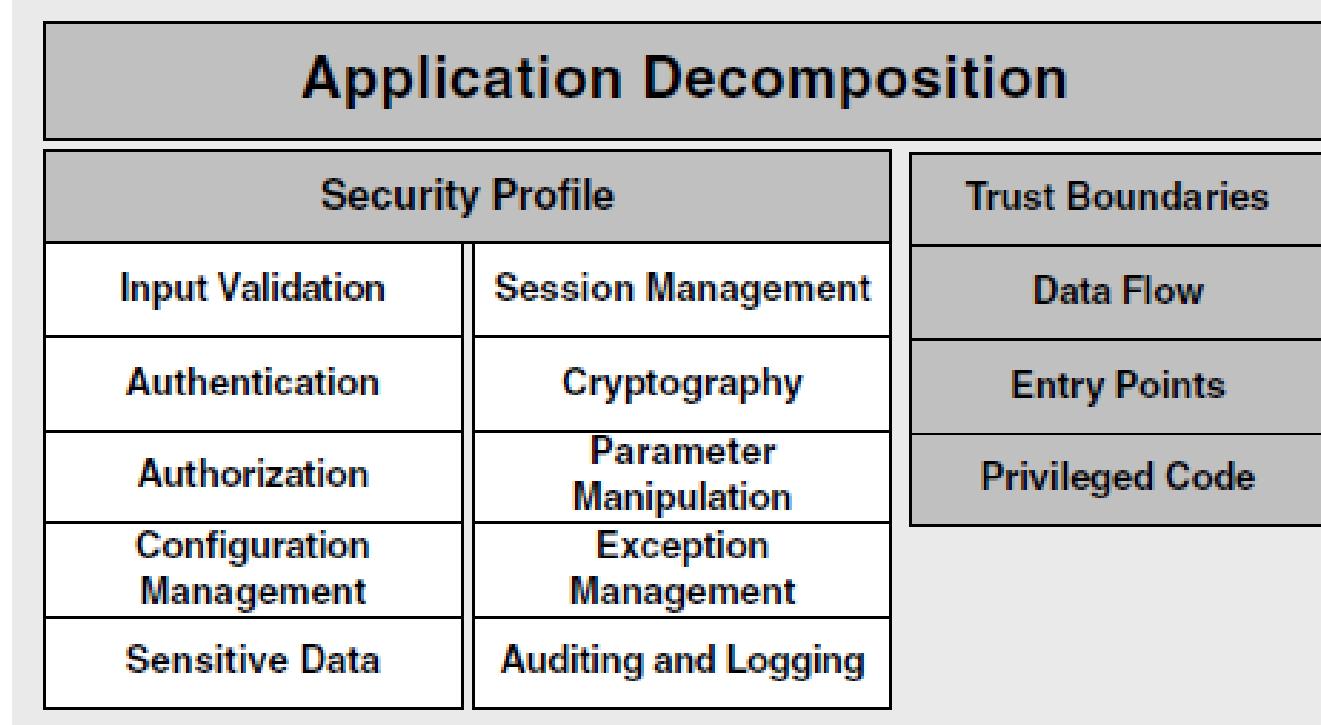
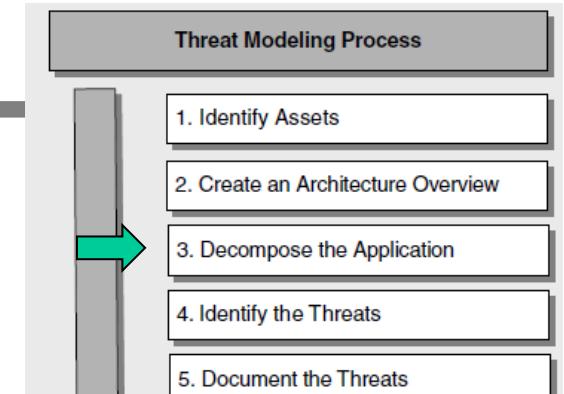
Arhitektura i tehnologije implementacije

- ◆ Dijagram visoke razine - opisuje strukturu (komponente) sustava
 - ovisno o složenosti aplikacije treba izraditi detaljnije dijagrame dijelova
 - npr. dijagrame pojedinih slojeva višeslojne aplikacije
 - određivanje tehnologija implementacije na koje se nadodaju aspekti zaštite, pr.



Korak 3 – Dekompozicija aplikacije

- ◆ Izrada sigurnosnog profila (security profile)
- ◆ Određivanje
 - granica povjerenja (trust boundaries)
 - toka podataka
 - mesta unosa
 - privilegiranog koda
- ◆ Za svaku aplikaciju

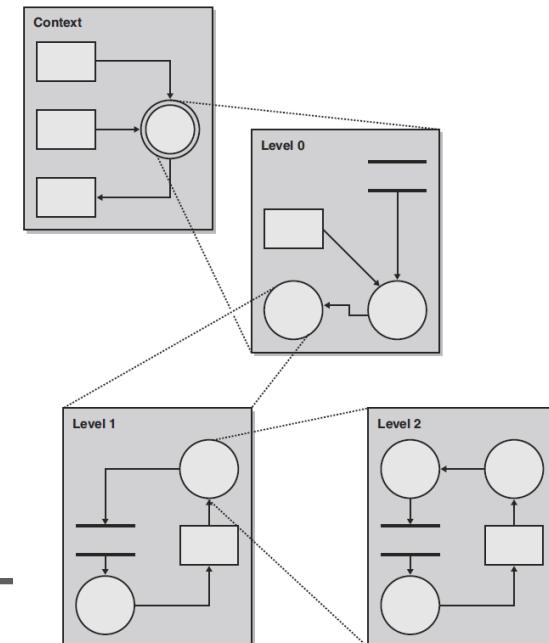


- ◆ Tehnike dekompozicije
 - funkcionalna dekompozicija, dijagram aktivnosti, dijagram toka podataka, ...

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
 - Razine: 0-sistem, 1-glavne mogućnosti, 2-detalji



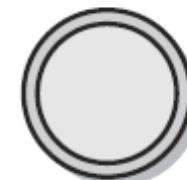
Dijagram toka podataka - notacija

- ◆ Proces, višestruki proces
 - obrada podataka, ili akcija temeljem podataka
 - kolekcija potprocesa, može se dekomponirati
- ◆ Spremište podataka
 - Bilo koji oblik pohrane (datoteka, BP, ...)
- ◆ Granica povjerenja
 - oznaka promjene privilegije (razine prava nad podacima)
- ◆ Vanjski entitet, sudionik
 - sve što je izvan aplikacije, a u interakciji putem točke unosa
- ◆ Tok podataka
 - usmjereni kretanje podatka unutar aplikacije



A Process

Transforms or manipulates data.



Multiple Processes

Transforms or manipulates data.



A Data Store

A location that stores temporary or permanent data.



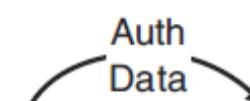
Boundary

A machine, physical, address space or trust boundary.



Interactor

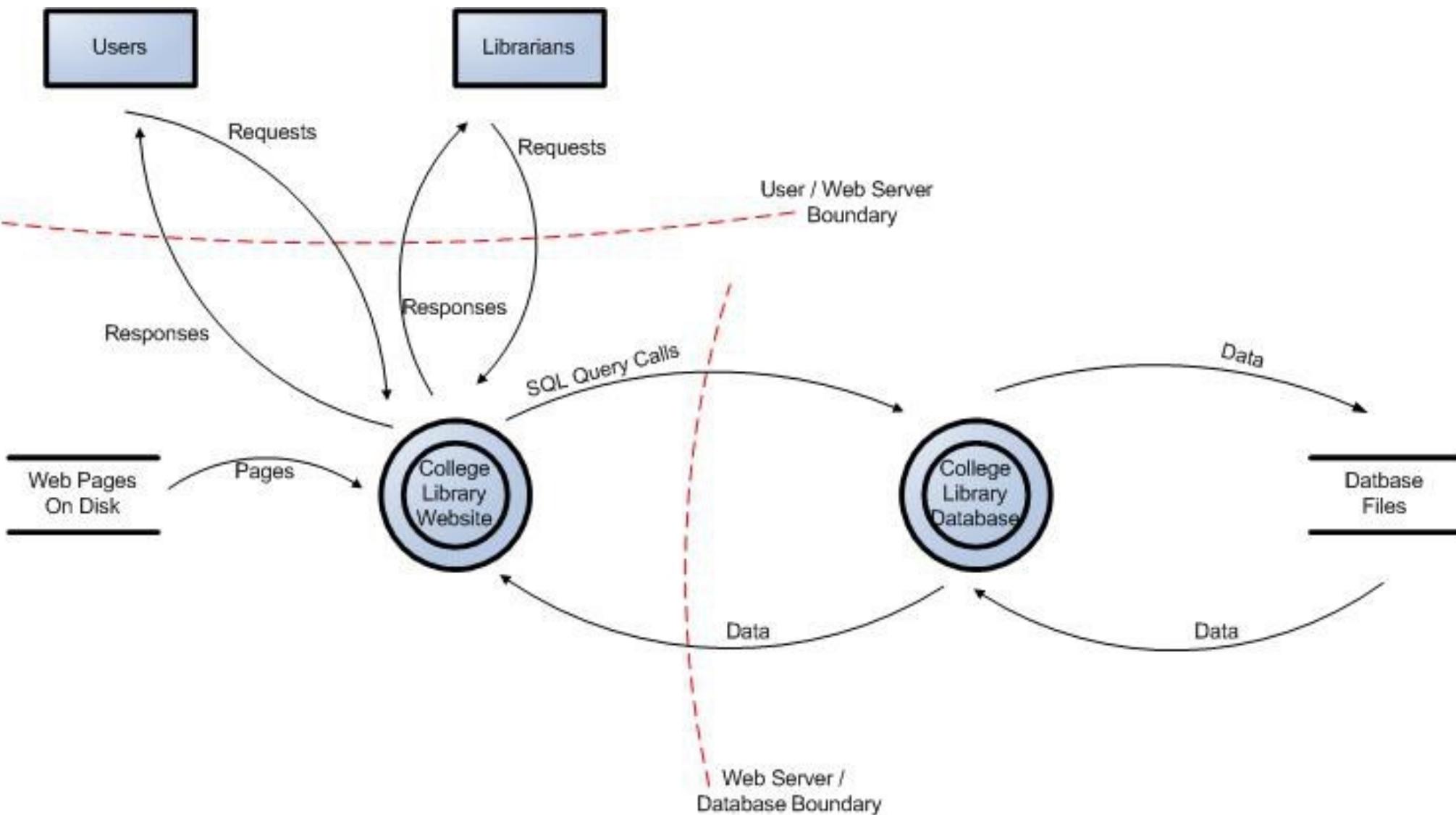
Input to the system.



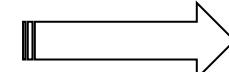
Data Flow

Depicts data flow from data stores, processes or interactors.

Dijagram toka podataka - primjer



Ostale aktivnosti dekompozicije

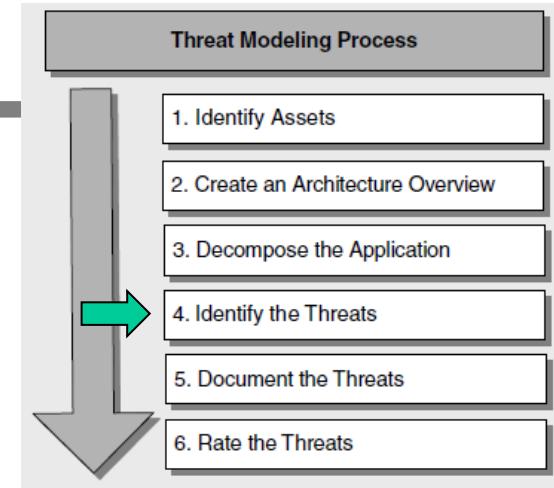
- ◆ Određivanje točki unosa (entry point)
 - dijelovi korisničkog sučelja, npr. stranice web aplikacije
 - priključne točke prijenosa podataka, npr. sučelja web servisa, remoting komponente, fizički portovi i priključnice (sockets)
- ◆ Određivanje privilegiranog koda
 - koji pristupa određenim tipovima sigurnih resursa ili obavlja privilegirane operacije
 - pr. ne/sigurni resursi: DNS poslužitelji, *registry*, *event log*, ..., pisači, web servisi, ...
 - pr. ne/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 

Category	Considerations
Input validation	<p>Is all input data validated?</p> <p>Could an attacker inject commands or malicious data into the application?</p> <p>Is data validated as it is passed between separate trust boundaries (by the recipient entry point)?</p> <p>Can data in the database be trusted?</p>
Authentication	<p>Are credentials secured if they are passed over the network?</p> <p>Are strong account policies used?</p> <p>Are strong passwords enforced?</p> <p>Are you using certificates?</p> <p>Are password verifiers (using one-way hashes) used for user passwords?</p>
Authorization	<p>What gatekeepers are used at the entry points of the application?</p> <p>How is authorization enforced at the database?</p> <p>Is a defense in depth strategy used?</p> <p>Do you fail securely and only allow access upon successful confirmation of credentials?</p>
Configuration management	<p>What administration interfaces does the application support?</p> <p>How are they secured?</p> <p>How is remote administration secured?</p> <p>What configuration stores are used and how are they secured?</p>
Sensitive data	<p>What sensitive data is handled by the application?</p> <p>How is it secured over the network and in persistent stores?</p> <p>What type of encryption is used and how are encryption keys secured?</p>

Category	Considerations
Session management	<p>How are session cookies generated?</p> <p>How are they secured to prevent session hijacking?</p> <p>How is persistent session state secured?</p> <p>How is session state secured as it crosses the network?</p> <p>How does the application authenticate with the session store?</p> <p>Are credentials passed over the wire and are they maintained by the application? If so, how are they secured?</p>
Cryptography	<p>What algorithms and cryptographic techniques are used?</p> <p>How long are encryption keys and how are they secured?</p> <p>Does the application put its own encryption into action?</p> <p>How often are keys recycled?</p>
Parameter manipulation	<p>Does the application detect tampered parameters?</p> <p>Does it validate all parameters in form fields, view state, cookie data, and HTTP headers?</p>
Exception management	<p>How does the application handle error conditions?</p> <p>Are exceptions ever allowed to propagate back to the client?</p> <p>Are generic error messages that do not contain exploitable information used?</p>
Auditing and logging	<p>Does your application audit activity across all tiers on all servers?</p> <p>How are log files secured?</p>

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** – nepriznavanje, 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
-

Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		X		X	X	
Data Stores		X	?	X	X	
Processes	X	X	X	X	X	X
Interactors	X		X			

Prijetnje i standardne protumjere

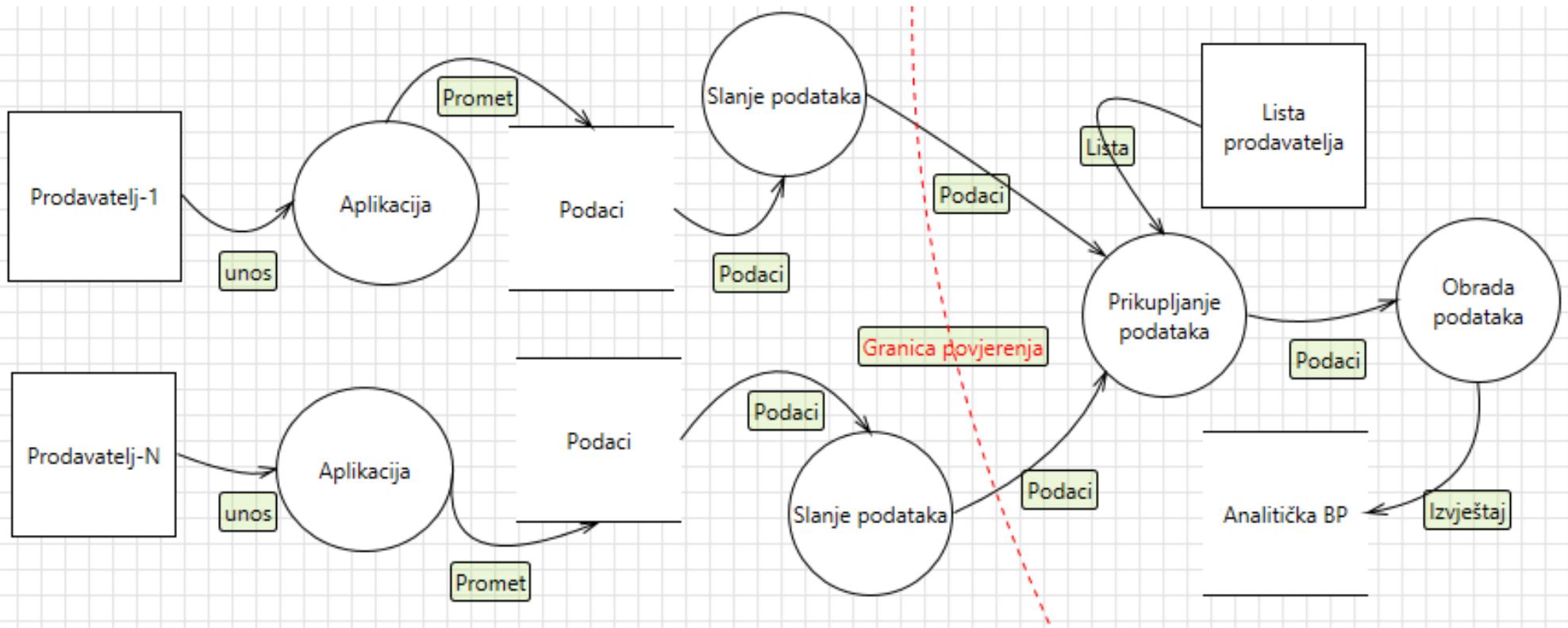
Spoofing	Authentication	<ul style="list-style-type: none"> To authenticate principals: Basic & Digest authentication LiveID authentication Cookie authentication Windows authentication (NTLM) Kerberos authentication PKI systems such as SSL/TLS and certificates IPSec Digitally signed packets <p>To authenticate code or data:</p> <ul style="list-style-type: none"> Digital signatures Message authentication codes Hashes
Tampering	Integrity	<ul style="list-style-type: none"> Windows Mandatory Integrity Controls ACLs Digital signatures Message Authentication Codes
Repudiation	Non Repudiation	<ul style="list-style-type: none"> Strong Authentication Secure logging and auditing Digital Signatures Secure time stamps Trusted third parties
Information Disclosure	Confidentiality	<ul style="list-style-type: none"> Encryption ACLS
Denial of Service	Availability	<ul style="list-style-type: none"> ACLS Filtering Quotas Authorization High availability designs
Elevation of Privilege	Authorization	<ul style="list-style-type: none"> ACLS Group or role membership Privilege ownership Permissions Input validation

„laundry list“

STRIDE - primjer

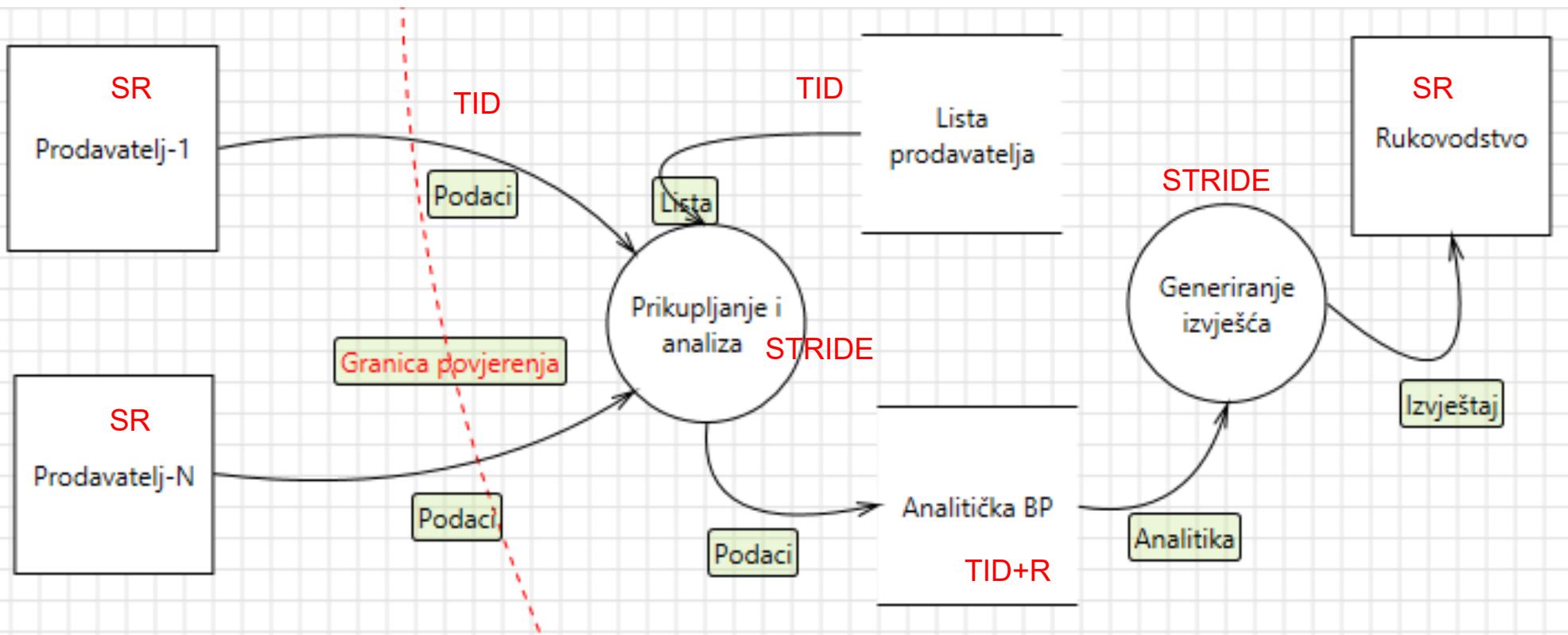
- ◆ Primjer: Uncover Security Design Flaws Using The STRIDE Approach
 - Prodavatelji prikupljaju podatke o prodaji u lokalnim evidencijama
 - Treba prikupiti datoteke o prodaji na poslužitelju
 - Te generirati tjedna izvješća
 - Za prethodno evidentirane prodavatelje
- ◆ Zahtjevi na sigurnost
 - Zaštiti podatke pri prijenosu i pohrani
 - Autentificirati i autorizirati prodavatelje
 - Aplikacija otporna na napade (unosa, injekcije, preljeva)
 - ...
- ◆ ne treba ih korisnik sve izreći – treba ih iznaci s obzirom na problem

Početni dijagram – klijent/poslužitelj



- Ponori podataka – netko ih treba čitati, procesom
- Ulančani procesi – ukazuju na ovisnost, razdvojiti
- Redundancija – generalizirati i normalizirati ponašanje (funkcionalnost)
- Pogrešni izvori podataka – provjeriti, promijeniti

Poboljšani dijagram – analiza poslužiteljske strane

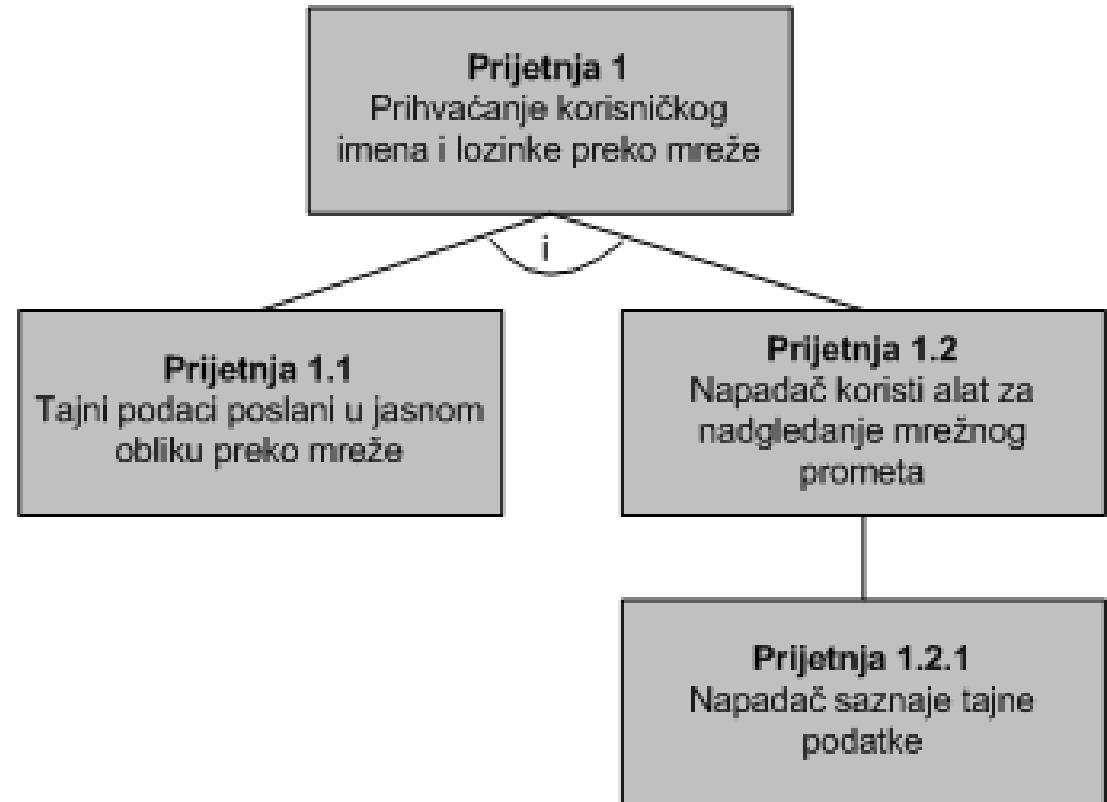


- Izbačen klijent
- Dodano *Generiranje*, posljedično i *Rukovodstvo*
- Lista postala spremište
- Integrirana obrada

◆ STRIDE ?

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



Stabla prijetnji (nastavak)

◆ Alternativni prikaz

1.0 Prijetnja 1 :

Prihvatanje korisničkog imena i lozinke preko mreže

1.1 Tajni podaci poslati u jasnom obliku preko mreže **AND**

1.2 Napadač koristi alat za nadgledanje mrežnog prometa

1.2.1 Napadač saznaće tajne podatke

◆ 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.

◆ Koje prijetnje ilustrira gornji primjer ?

Obrasci napada

- ◆ Općenita reprezentacija uobičajenih napada
 - definira cilj, uvjete, tehniku i rezultat napada
- ◆ Naglasak je na tehnici napada (kod STRIDE na ciljevima napadača)
- ◆ Primjer obrasca:

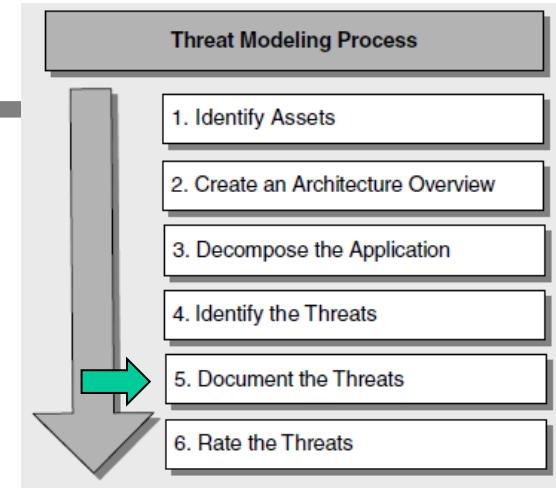
Pattern	Code injection attacks
Attack goals	Command or code execution
Required conditions	Weak input validation Code from the attacker has sufficient privileges on the server.
Attack technique	<ol style="list-style-type: none">1. Identify program on target system with an input validation vulnerability.2. Create code to inject and run using the security context of the target application.3. Construct input value to insert code into the address space of the target application and force a stack corruption that causes application execution to jump to the injected code.
Attack results	Code from the attacker runs and performs malicious action.

Korak 5 - Dokumentiranje prijetnji

- ◆ Predložak za evidenciju prijetnji
 - svakako se popunjavaju opis i cilj
 - rizik se ostavlja za naredni korak
 - ostali atributi mogu biti opcionalni

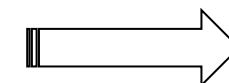
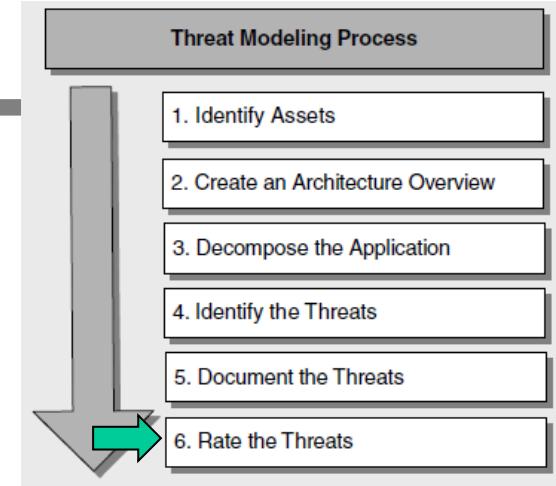
- ◆ Primjeri

Threat Description		Attacker obtains authentication credentials by monitoring the network
Threat target		Web application user authentication process
Risk		
Attack techniques		Use of network monitoring software
Countermeasures		Use SSL to provide encrypted channel
Threat Description		Injection of SQL commands
Threat target		Data access component
Risk		
Attack techniques		Attacker appends SQL commands to user name, which is used to form a SQL query
Countermeasures		Use a regular expression to validate the user name, and use a stored procedure that uses parameters to access the database.



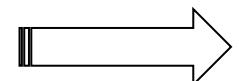
Korak 6 - Rangiranje prijetnji

- ◆ 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 zadalu prijetnju



DREAD (model procjene rizika)

- ◆ DREAD – 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
- ◆ Bolje – (jednostavna) **shema ocjenjivanja**
 - Nisko, srednje, visoko – preslikano u interval 1 do 3



Primjer jednostavne sheme ocjenjivanja

Rating	High (3)	Medium (2)	Low (1)
D	Damage potential The attacker can subvert the security system; get full trust authorization; run as administrator; upload content.	Leaking sensitive information	Leaking trivial information
R	Reproducibility The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E	Exploitability A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A	Affected users All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.	The bug is obscure, and it is unlikely that users will work out damage potential.

Primjer jednostavne sheme ocjenjivanja (nastavak)

- ◆ Zbrajaju se vrijednosti (1-3) za zadanu prijetnju
 - rezultat je u rasponu 5-15
 - pridjeljuje se rizik, npr. 5-7 nizak, 8-11 srednji, 12-15 visok
- ◆ Npr. za dvije dokumentirane prijetnje s početka priče

Threat	D	R	E	A	D	Total	Rating
Attacker obtains authentication credentials by monitoring the network.	3	3	2	2	2	12	High
SQL commands injected into application.	3	3	3	3	2	14	High

- ◆ ... nadopunjavaju se predlošci za dokumentiranje prijetnji (korak 5)

Razrješenje prijetnji (nakon modeliranja)

- ◆ Popraviti (smanjenje, redukcija rizika)
 - smanjiti posljedicu
- ◆ Ne učiniti ništa (prihvati 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

Prijetnje i protumjere – drugi put

Prijetnje	Protumjere (sigurnosne tehnike)
Zavaravanje	Odgovarajuća autentifikacija Zaštita privatnih podataka Privatni podaci ne smiju se spremati u jasnom obliku
Izmjena podataka	Odgovarajuća autorizacija Korištenje funkcija sažimanja Korištenje digitalnih potpisa
Poricanje	Korištenje digitalnih potpisa
Otkrivanje informacija	Odgovarajuća autorizacija Privatni podaci ne smiju se spremati u jasnom obliku Osigurati komunikacijski kanal
Uskraćivanje usluge	Potvrditi i filtrirati ulazne podatke
Povišenje ovlasti	Dodijeliti samo nužne ovlasti

Primjer, Microsoft Threat Modeling Tool

Firma02 - Microsoft Threat Modeling Tool

File Edit View Settings Diagram Reports Help DiagramReader

Analitika X

The diagram illustrates a threat model for a web server. A central process node labeled "Prikupljanje i analiza" (Collection and Analysis) is connected to several data nodes: "Podaci" (Data), "Lista" (List), and "Izvješta" (Report). It also receives input from "Prodavatelj-1" and "Prodavatelj-N". A red dashed line labeled "Granica povjerenja" (Trust Boundary) separates the internal components from external entities like "Lista prodavatelja" (Seller List) and "Rukovodstvo" (Management). An arrow points from the "Prikupljanje i analiza" node to a circular node labeled "Generiranje izvješća" (Report Generation), which then leads to the "Izvješta" node.

Threat List

ID	Title	Category	Description	Short Description
9	Persistent Cross Site Scripting	Tampering	The web server 'Prikupljanje i analiza' is vulnerable to persistent cross-site scripting (XSS) attacks. This threat can be exploited by injecting malicious scripts into user-supplied data, such as product descriptions or reviews, which are then displayed to other users. These scripts can steal sensitive information or manipulate the application's behavior.	Tampering is the manipulation of data that is sent to or received from a user. It includes threats like persistent XSS, reflected XSS, and session hijacking.
10	Cross Site Scripting	Tampering	The web server 'Prikupljanje i analiza' is vulnerable to cross-site scripting (XSS) attacks. This threat can be exploited by injecting malicious scripts into user-supplied data, such as product descriptions or reviews, which are then displayed to other users. These scripts can steal sensitive information or manipulate the application's behavior.	Tampering is the manipulation of data that is sent to or received from a user. It includes threats like persistent XSS, reflected XSS, and session hijacking.
11	Spoofing of Source Data Store	Spoofing	The web server 'Prikupljanje i analiza' is vulnerable to spoofing attacks. This threat can be exploited by injecting malicious data into the source data store, such as the seller list. The attacker can manipulate the data to appear as if it came from a trusted source, leading to incorrect results or security issues.	Spoofing is the act of impersonating a legitimate user or system to gain unauthorized access or manipulate data. It includes threats like man-in-the-middle (MitM) attacks and session replay.

Export Csv Clear Filters 4 Threats Displayed, 31 Total

Threat Properties

ID: 11 Diagram: Analitika Status: Not Started Last Modified: 27. 11. 14. 15:46:46

Title: Spoofing of Source Data Store Lista prodavatelja

Category: Spoofing

Notes - no entries

Element Properties

Web Server

Name: Prikupljanje i analiza
Out Of Scope:
Reason For Out Of Scope:

Configurable Attributes

Code Type: Managed
Sanitizes Input: Not Selected
Sanitizes Output: Not Selected

As Generic Process

Running As: Not Selected
Isolation Level: Not Selected
Accepts Input From: Not Selected
Implements or Uses an Authentication Mechanism: No
Implements or Uses an Authorization Mechanism: No
Implements or Uses a Communication Protocol: No

Add New Custom Attribute

Smanjenje površine napada

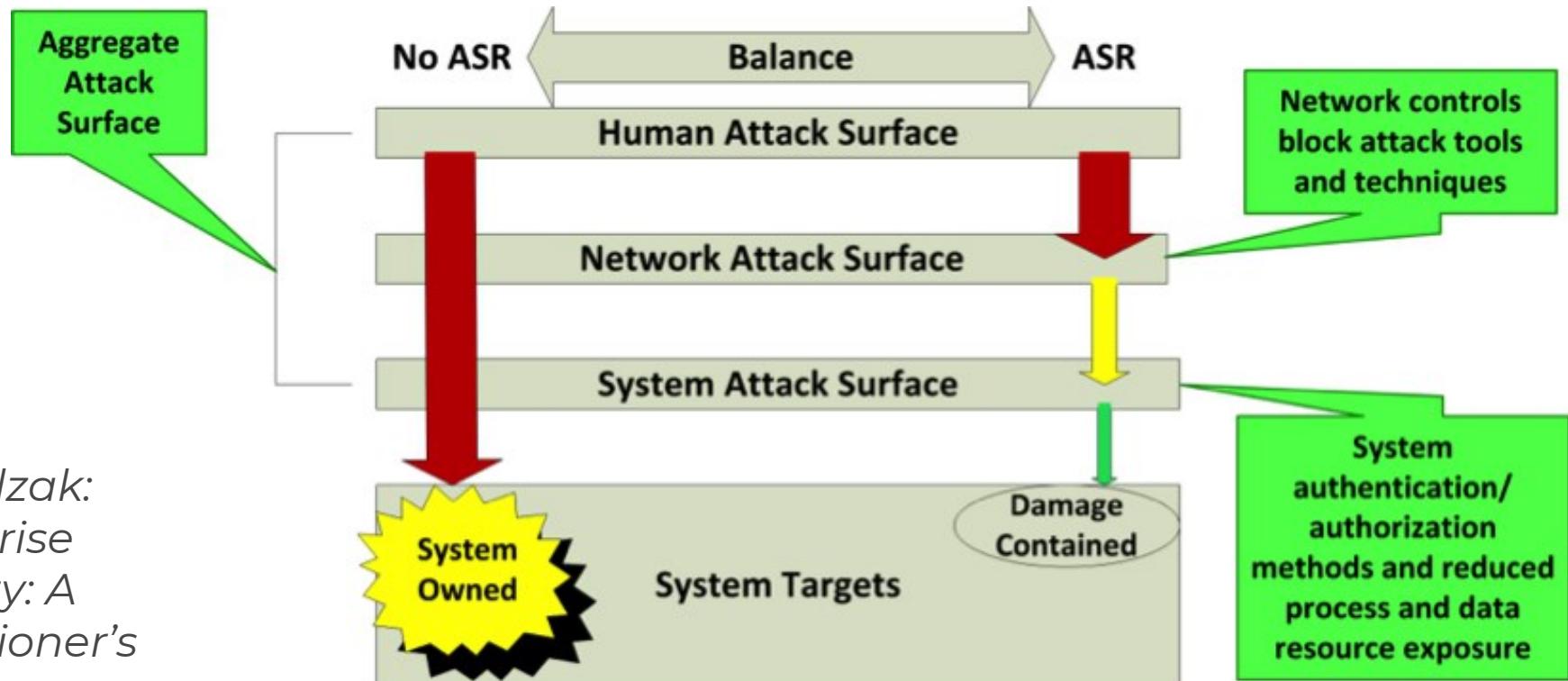
Attack Surface Reduction

Površina napada (attack surface)

- ◆ 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
 - broj elemenata koji su vidljivi ili se mogu koristiti



*Tom Olzak:
Enterprise
Security: A
practitioner's
guide*

Smanjenje površine napada (attack surface reduction)

- ◆ Glavni ciljevi
 - Smanjenje količine koda koji se izvodi „po viđenju“ (by default)
 - Smanjenje količine koda kojem mogu pristupiti nepouzdani (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
 - Computer emergency response teams (CERT), <http://www.cert.hr/>
 - Common Vulnerabilities and Exposures (CVE) dictionary, <https://cve.mitre.org/>
 - 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
 - Osnovica (baseline) + tjedna mjerenja
 - Određivanje minimalne površine na početku
 - Ako se površina povećava – odrediti kako ju smanjiti

Površina napada i pristupne točke

- ◆ Primjer usporedbe mjerenja različitih verzija

Baseline	Baseline + 1 month	Comment
3 x TCP ports	2 x TCP ports	Good; one fewer port to worry about.
1 x UDP port	2 x UDP port	Which functionality opened the new UDP port? Why is it open by default? Is it authenticated? Is it restricted to a subnet?
2 x Services (both SYSTEM)	3 x Services (2 x SYSTEM, 1 x LocalService)	Why is another service running by default? Why are any running as SYSTEM?
3 x ActiveX controls	4 x ActiveX controls	Why is the new control installed? Is it safe for scripting?
No additional user accounts	1 x application account	Turns out this is a member of the administrators group too! Why? What's the password?

*Fending Off Attacks by Reducing an Application's Attack Surface
Jason Taylor CTO, Security Innovationan SDL Pro Network member company*

Proces ASR

The Security Development Life Cycle

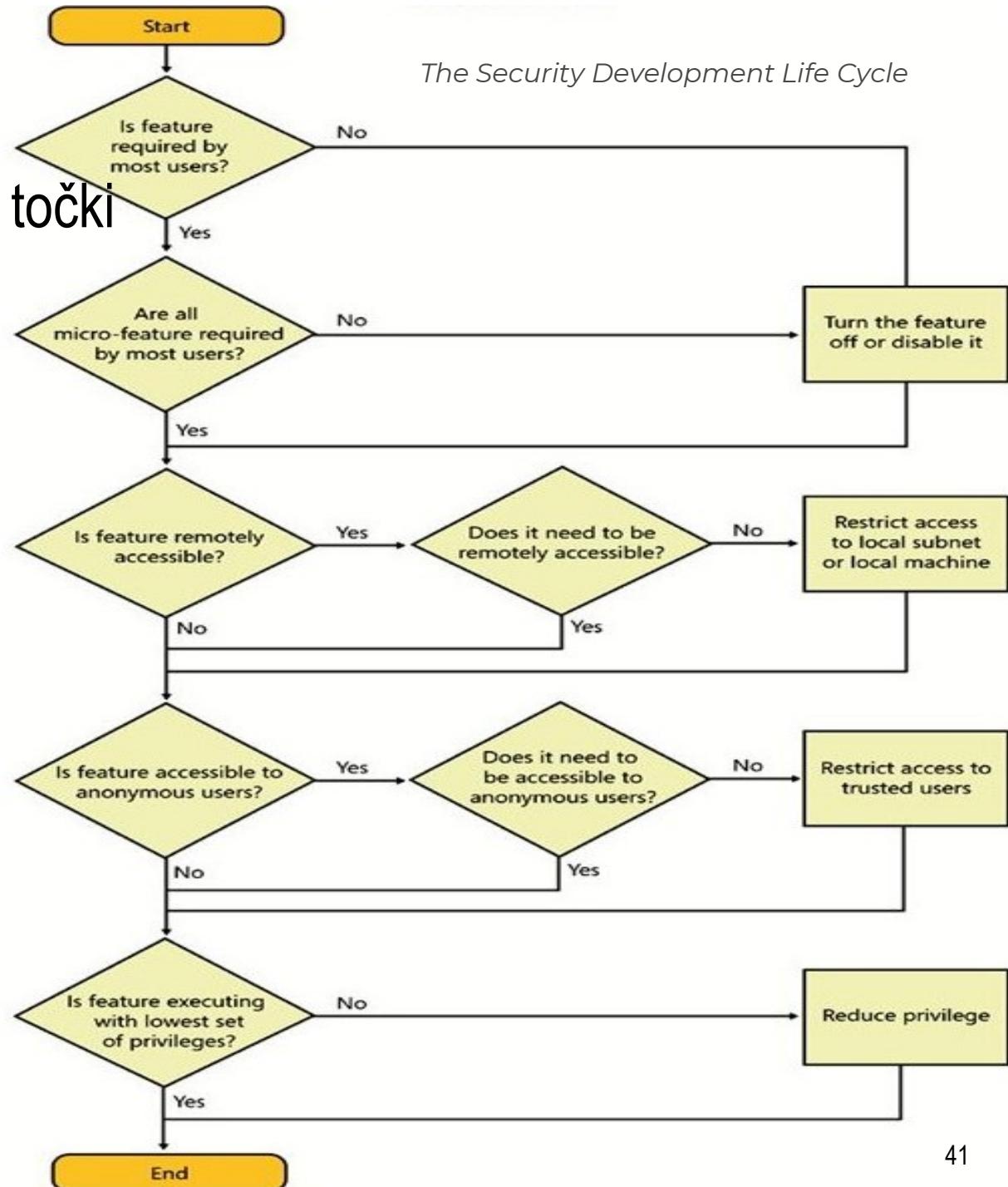
◆ Ustanovljavanje pristupnih točki

- mrežne, datoteke, ...

◆ Rangiranje točaka

- prema korisniku
- autentificirani – anonimni
- *admin* – *user*
- mrežni – *local*

◆ Podešavanje



It's Not Just About Turning Stuff Off!

Higher Attack Surface

Executing by default

Open socket

Anonymous access

Constantly on

Admin access

Internet access

SYSTEM

Uniform defaults

Large code

Weak ACLs

Lower Attack Surface

Off by default

Closed socket

Authenticated access

Intermittently on

User access

Local subnet access

Not SYSTEM!

User-chosen settings

Small code

Strong ACLs

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

Najbolje prakse (nastavak)

- ◆ Redukcija privilegija radi ograničavanja potencijalne štete
 - Uklanjanje 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 anonymnih 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*

Najbolje prakse (ostatak)

- ◆ 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

Primjer: Attack Surface Analyzer

Attack Surface Report: Table Of Contents

The screenshot shows the 'Attack Surface Analyzer' application window. At the top, there's a toolbar with icons for file operations. Below the toolbar, the title bar says 'Attack Surface Analyzer'. The main content area has a heading 'Welcome to Attack Surface Analyzer'. A text block explains that the tool scans the system to identify potential security issues and recommends scanning at least twice: a 'baseline' scan on a clean system and a 'product' scan after installing the product. It also notes that each scan generates a .CAB file for analysis. Below this, there's a section titled 'Please select an action:' with two radio button options: 'Run new scan' (unchecked) and 'Generate standard attack surface report' (checked). Under 'Select options:', there are three input fields for 'Baseline Cab', 'Product Cab', and 'Report Filename', each with a 'Browse...' button. The 'Report Filename' field contains the path 'C:\Users\kreso\Attack Surface Analyzer\GOLIJAT_1.0.0_2015-12-01_22-31-'. At the bottom right is a large 'Generate' button.

- [System Information](#)
 - [Running Processes](#)
 - [Executable Memory Pages](#)
 - [Windows](#)
 - [Impersonation Tokens](#)
 - [Kernel Objects](#)
 - [Window Stations](#)
 - [Desktops](#)
 - [Modules](#)
- [Service Information](#)
 - [Services](#)
 - [Drivers](#)
- [ActiveX, DCOM, COM, File Extensions](#)
 - [COM Controls](#)
 - [ActiveX Controls](#)
 - [DCOM Controls](#)
 - [File Registrations](#)
- [Internet Explorer](#)
 - [Pluggable Protocol Handlers](#)
 - [IE Silent Elevations](#)
 - [IE Preapproved Controls](#)
 - [Browser Helper Objects](#)
- [Network Information](#)
 - [Network Ports](#)
 - [Named Pipes](#)
 - [RPC Endpoints](#)
 - [Network Shares](#)
- [Firewall](#)
 - [Firewall Rules](#)
- [System Environment, Users, Groups](#)
 - [%PATH% Entries](#)
 - [Groups](#)

Reference

- ◆ Postupci
 - [A systematic review of security requirements engineering, Mellado et.a., 2010](#)
 - [Threat Modeling with STRIDE](#)
- ◆ Alati
 - [Attack Surface Analyzer](#) – smanjenje površine napada
 - [Microsoft Threat Modeling Tool](#) – modeliranje prijetnji
 - [Top 10 Threat Modeling Tools](#)



Zaštita i sigurnost informacijskih sustava

Provjera sigurnosti

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 licencom 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/>.

Provjera sigurnosti

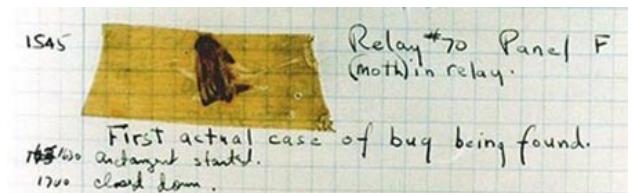
Security Testing

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
 - staticka analiza - bez pokretanja, izvorni kod ili .NET MSIL, Java Bytecode
 - dinamička analiza - pokretanjem, što ne isključuje izvorni kod

Ključni pojmovi

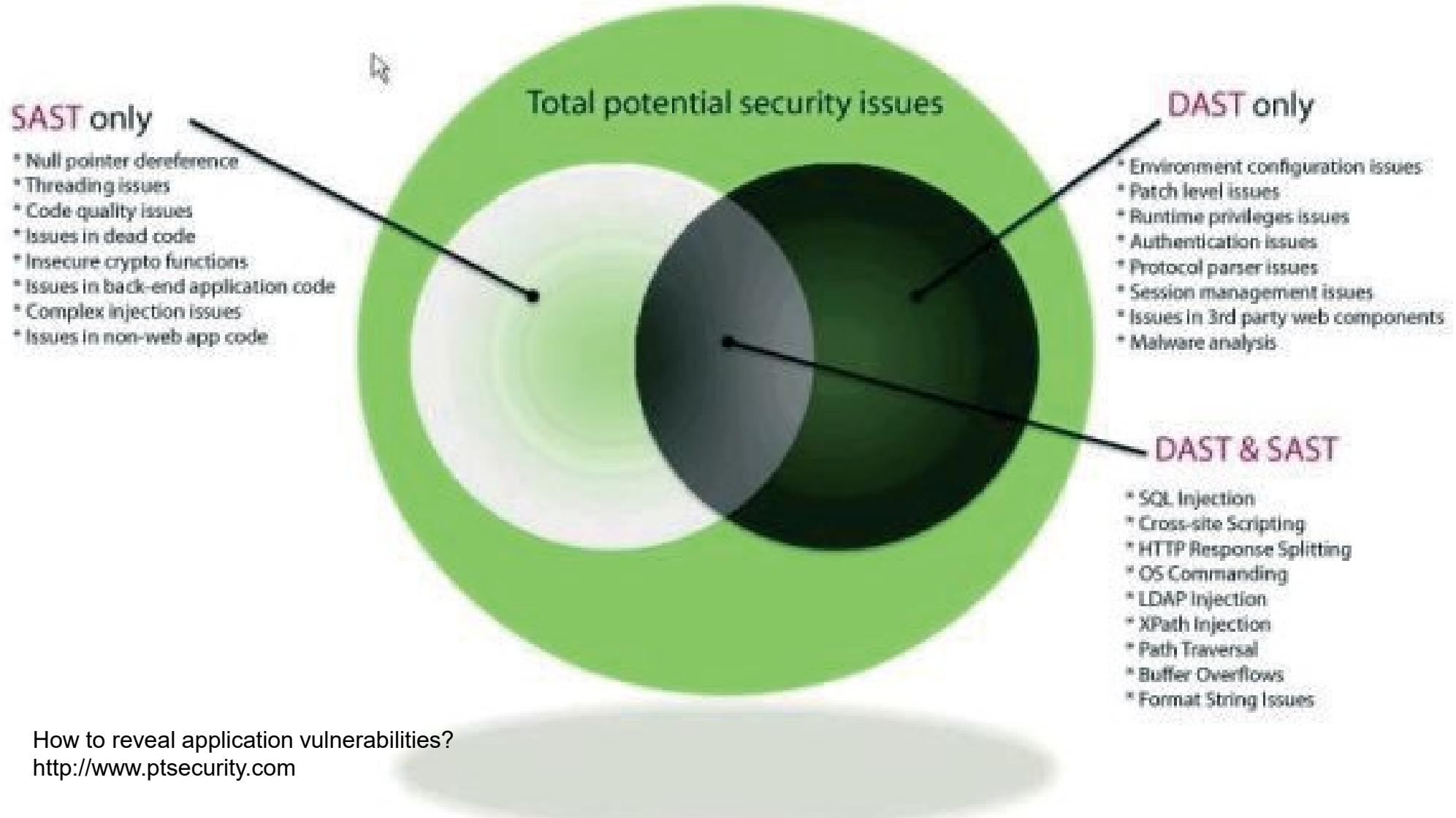
- ◆ [„normalan”] **Test** - provjerava je li neki aspekt softvera ispravan
- ◆ **Test sigurnosti** - nastoji dokazati da neki dio ne radi kako treba
- ◆ **Pogreška** (error) - propust programera, npr. radi nerazumijevanja
 - dovodi do jednog ili više kvarova
 - razlikujemo u odnosu na "pogrešku" koja znači neželjeno stanje, tj. kvar
- ◆ **Kvar** (fault), **defekt** (defect), **neformalno bug** - neispravan dio koda
 - npr. pogrešna prepostavka 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

How to reveal application vulnerabilities?



Nadzor

Security Review, Code Reviews, Inspection

Revizija, recenzija (peer review)

- 
- ◆ Varijante
 - Inspekcija (inspection)
 - Timski pregled (team review)
 - Prohod (walkthrough)
 - ◆ Koristi
 - našaženje defekata ranije u životnom ciklusu - do 80% prije testiranja
 - našaženje defekata s manje napora nego testiranjem
 - IBM - pregled 3.5 h/defekt, test 15-25 h/defekt
 - našaženje drugačijih defekata nego testiranjem - problemi dizajna i zahtjeva
 - poduka razvojnika - da ne ponavljaju iste pogreške

- ◆ 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 defekt/kLOC inspekcije naspram 3 defekt/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
 - čitalac prepričava, recenzenti komentiraju i zapitkuju, zapisničar evidentira
 - tim zaključuje procjenu koda
- ◆ Prerada
 - autor popravlja
- ◆ Kontrola (follow-up)
 - moderator verificira korektnost promjena, autor prijavljuje kod (chek-in)

Varijante

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

- ◆ Prohod (walkthrough)
 - Autor vodi sastanak i objašnjava kod
 - Manje formalan proces
 - 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*
 - Samo jedan recenzent uz autora
 - neformalna recenzija
 - može uključiti kontrolne liste i druge postupke
- ◆ *Pass around* (kružno dodavanje?)
 - višestruki, istodobni *Peer deskcheck*
 - više recenzenata (istog koda)

Statička provjera

Static Analysis

Statička analiza

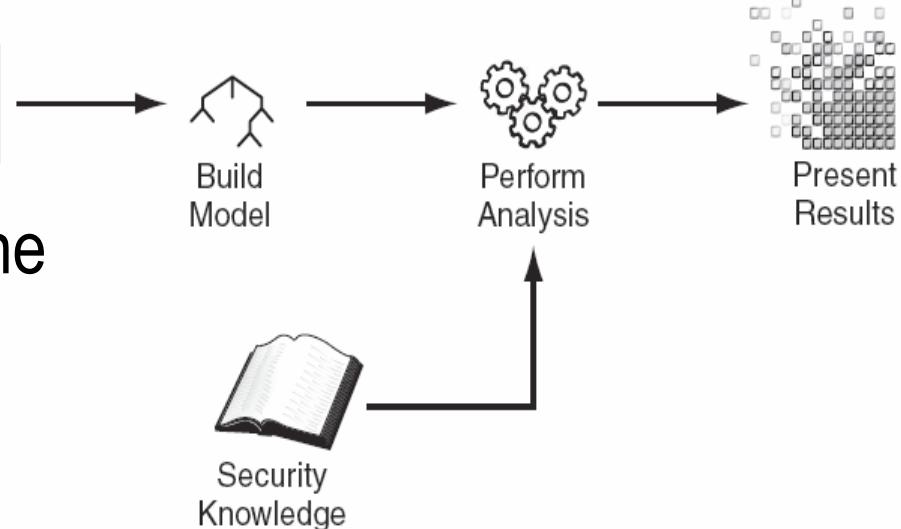
- ◆ SAST (Quick and Dirty)
 - Analiza koda bez izvršavanja
 - Obuhvaća sve osim testiranja
 - Korištenje analizatora koda
 - Može biti dio revizije koda
- ◆ Ograničenja: pogrešno otkrivanje i pogrešno neprepoznavanje
 - **False Positives** - nepostojeći bugovi, nemoć pri složenom kodu ili vanjskom
 - **False Negatives** - neprepoznavanje bugova, složenost koda, slabost pravila

Vrste statičke analize

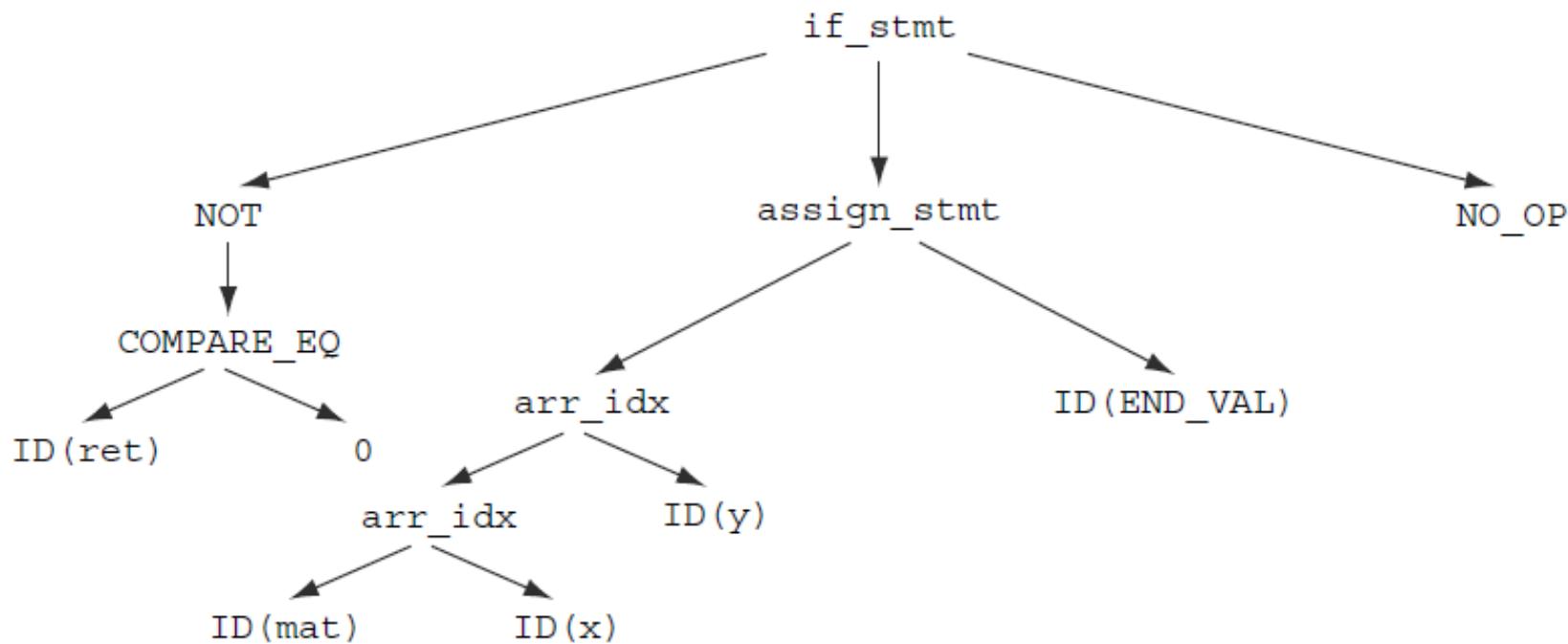
- ◆ Provjera tipova (Type checking) - dio programskog jezika
 - pr. int i = "abc";
- ◆ Provjera stila (Style checking) - dobre prakse
 - Pravila - praznine/proredi, nazivlje, komentari, ...
- ◆ Razumijevanje programa (Program understanding) - zaključivanje značenja
 - Sva korištenja metode, nalaz deklaracije globalnih varijabli, ...
- ◆ Provjera svojstava (Property checking) - osiguranje da nema lošeg ponašanja
 - Npr. curenje memorije : if (malloc() == NULL)
- ◆ Verifikacija programa (Program verification) - osiguranje ispravnog ponašanja
 - Npr. free(mem);
- ◆ Traženje pogrešaka (Bug finding) - otkrivanje mogućih pogrešaka
 - Obrasci bugova

Mehanizmi statičke analize

```
if (fgets (buf, 10, Source) != buf) {  
    strcpy (buf, "buf");  
    system (othr);
```



- ◆ Parser, Model Builder, Analysis Engine
- ◆ Parser za svaki programski jezik
 - generira Abstract Syntax Tree (AST)



Tehnike analize

◆ Leksička analiza (Lexical Analysis) i parsiranje

- Primjer, programski odsječak
- Pripadni slijed simbola (tokena)

```
if (ret) // probably true  
mat[x][y] = END_VAL;
```

```
IF LPAREN ID(ret) RPAREN ID(mat) LBRACKET ID(x) RBRACKET  
LBRACKET ID(y) RBRACKET EQUAL ID(END_VAL) SEMI
```

- Izdvojen sintaksnim pravilima, parsiran prema produkcijama gramatike

```
if { return IF; }  
( { return LPAREN; }  
) { return RPAREN; }  
[ { return LBRACKET; }  
] { return LBRACKET; }  
= { return EQUAL; }  
; { return SEMI; }  
/[ \t\n]+/ { /* ignore whitespace */ }  
//.*/ { /* ignore comments */ }  
/[a-zA-Z][a-zA-Z0-9]*"/ { return ID; }
```

```
stmt := if_stmt | assign_stmt  
if_stmt := IF LPAREN expr RPAREN stmt  
expr := lval  
assign_stmt := lval EQUAL expr SEMI  
lval = ID | arr_access  
arr_access := ID arr_index+  
arr_idx := LBRACKET expr RBRACKET
```

Tehnike analize (2)

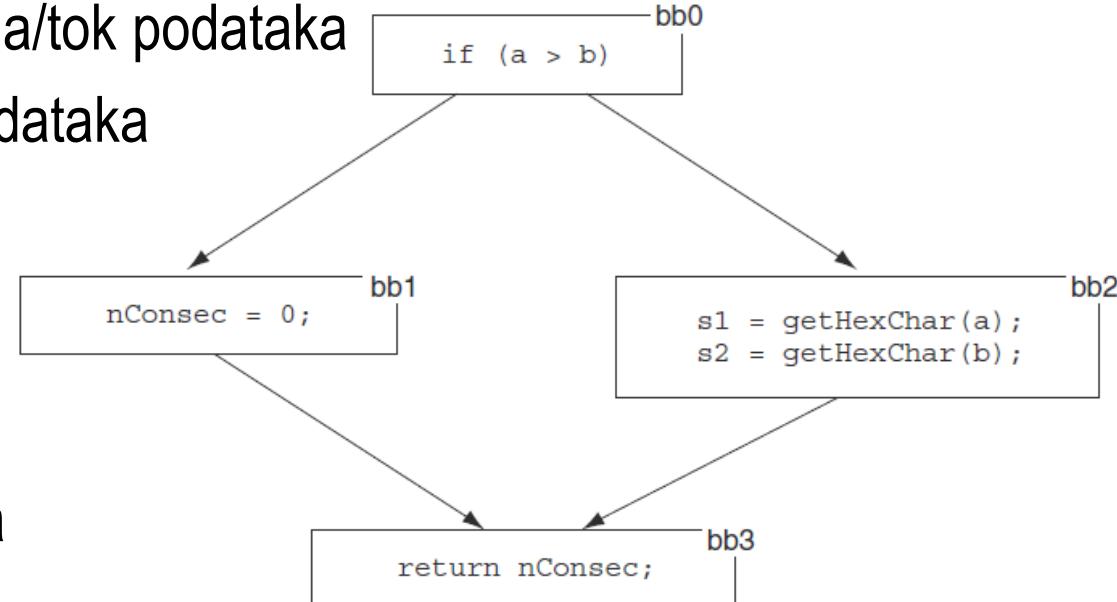
◆ Analiza toka podataka (Data Flow Analysis)

- Prikupljanje informacija o kolanju podataka pri izvršenju programa dok je zaustavljen
- Semantička analiza - na temelju AST i tablice simbola

- Osnovni blok - slijed naredbi koji se ne zaustavlja ili grana osim na kraju
- Control Flow Analysis – kontrola/tok podataka
- Control Flow Path - putanja podataka

◆ Graf kontrole toka

- Control Flow Graph (CFG)
- Primjer: graf s 4 osnovna bloka



Tehnike analize (3)

◆ Analiza „mrlja” (Taint Analysis)

- Identifikacija varijabli *uprlijanih* korisničkim unosom
- Praćenje njihove propagacije prema moguće ranjivim funkcijama (*sink*)
- Ako nisu dezinficirane prije odvoda - ranjivost

◆ Pravila propagacije mrlja

- Pravila izvora (source rules) - unos podataka
 - Naredbe *read()*, *getenv()*, *getpass()*, *gets()*.
- Pravila sливника (sink rules) - lokacije koje ne bi smjele primiti prljave podatke
 - Primjer, Java *Statement.executeQuery()*, C *strcpy()*
- Pravila propuštanja (pass-through)
 - Ako je *string* zaprljan i *trim(string)* će biti zaprljan
- Pravila čišćenja (cleanse rules) - validacija unosa
- Pravila početka (entry-point rule) - slično izvoru, npr. *main(...)*

Primjer statičke analize

```
if ( fgets ( buf , sizeof(buf) , stdin) == buf ) {  
    strcpy ( othr , buf );  
    system ( othr );  
}
```

The diagram illustrates the flow of tainted data through the code. It starts with a call to `fgets` (labeled 1), which taints the variable `buf`. This tainted `buf` is then used as the source for a `strcpy` operation (labeled 2), which copies its value into the variable `othr`. Finally, the variable `othr` is passed as an argument to the `system` function (labeled 5), which executes the command stored in `othr`.

- 1 A source rule for `fgets()` taints `buf` other
- 2 Dataflow analysis connects uses of `buf`
- 3 A pass-through rule for `strcpy` taints
- 4 Dataflow analysis connects uses of `othr`
- 5 Because `othr` is tainted, a sink rule for `system()` reports a command injection vulnerability

Prednosti i nedostaci statičke analize

◆ Prednosti

- Potpuna pokrivenost koda (code coverage) - u teoriji
- Potencijal potvrde izostanka čitavih klasa bugova
- Hvata bugove 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

- https://www.owasp.org/index.php/Source_Code_Analysis_Tools

- StyleCop <https://github.com/StyleCop/StyleCop> - C#
- CodeSmart <http://www.axtools.com/> - C#, C++, VB.NET
- NDepend <http://www.ndepend.com/> - C#, jDepend za Javu

- VS Code Analysis (FxCop, Roslyn analyzers) - C#, C/C++, ...
- PMD - Java, C, C++, C#, Groovy, PHP, Ruby, Fortran, JavaScript, PLSQL, ...
- Fortify Source Code Analyzer - 25 jezika
- Checkstyle - Java
- Klocwork K7 Suite - Java
- FindBugs, Find Security Bugs - Java
- Coverity Prevent - C#, clang, gcc

Primjer: StyleCop

The screenshot shows a Visual Studio interface with the following components:

- Code Editor (UserModel.cs):** Displays C# code for a `UserModel` class.
- Toolbox:** Standard Visual Studio toolbox items.
- Server Explorer, Toolbox, SQL Server Object Explorer:** Standard Visual Studio toolbars.
- StyleCop 5.0 Project Settings Dialog:** Opened over the code editor.
 - Rules Tab:** Shows the configuration for running StyleCop rules on the project.
 - Enabled rules section:** Lists categories like C#, Documentation Rules, Layout Rules, etc., with specific rules like SA1000, SA1001, etc., checked.
 - Detailed settings section:** Contains checkboxes for "Analyze designer files" (checked) and "Analyze generated files".
 - Note:** A note at the bottom indicates whether to include designer files (*.Designer.cs).
 - Buttons:** OK, Cancel, Apply.
- Error List:** Shows 0 Errors and 55 Warnings.

Primjer: IBM Rational AppScan Source Edition for Security

The screenshot shows the IBM Rational AppScan Source Edition for Security interface. The main window displays a trace analysis for a JSP file named `feedbacksuccess.jsp`. The trace shows a flow from a user input source (e.g., `getAttribute()`) through Java objects (`java.lang.StringBuilder`) to a sink (e.g., `JspWriter.print()`). The interface includes a left sidebar for findings, a bottom code editor showing the JSP code, and a right panel for finding details and mitigation.

Trace Details:

- Source: `javax.servlet.jsp.JspWriter.print`
- Trace Flow:
 - From `javax.servlet.jsp.JspWriter.print` to `javax.servlet.ServletRequest.getAttribute`
 - From `ServletRequest.getAttribute` to `java.lang.StringBuilder.append`
 - From `StringBuilder.append` to `java.lang.StringBuilder.toString`
 - From `toString` to `JspWriter.print`
- Context: `request.getAttribute("message_feedback") != null ? ", "+request.getAttribute("message_feedback") : "They will be reviewed by our Customer Service staff and given the full attention that they deserve."`
- Line: 49
- Code Snippet:

```
49 out.print( (request.getAttribute("message_feedback")!=null? ", "+request.getAttribute("message_feedback") : "They will be reviewed by our Customer Service staff and given the full attention that they deserve") );
50 out.write( "They will be reviewed by our Customer Service staff and given the full attention that they deserve" );
51 String email = (String) request.getParameter("email_addr");
52 boolean regExMatch = email!=null && email.matches(ServletUtil.EMAIL_REGEX);
53 if (email != null && email.trim().length() != 0 && regExMatch) {
54 out.write( "\r\n\t\tOur reply will be sent to your email: " );
55 out.print( ServletUtil.sanitizeBasic(email.toLowerCase()) /*email.toLowerCase() */ );
56 out.write("\r\n\t\t");
57 } else {
58 out.write("\r\n\t\tHowever, the email you gave is incorrect (" );
59 out.print(email.toLowerCase() />ServletUtil.sanitizeWebEmail.toLowerCase() ) );
```

Finding Detail:

- Context: `request.getAttribute("message_feedback") != null ? ", "+request.getAttribute("message_feedback") : "They will be reviewed by our Customer Service staff and given the full attention that they deserve"`
- Classification: `CrossSiteScripting`
- Vulnerability Type: `Injection`
- Severity: `High`
- Bundle: `JspWriter`

Mitigation:

To defend against these problems, the application should apply appropriate validation and encoding on the string. The validation mechanism should ensure that the string does not contain malicious data and code. HTML entity encoding should be applied to data that is not intended to be interpreted as scripts. For more details on validation and encoding, please refer to [Validation_Required](#) and [Validation_EncodingRequired](#).

Example:

The following JSP page prints the user provided name in the resulting welcome page.

```
welcome.jsp
<html>
<head>
<title>Welcome</title>
</head>
<body>
<%
    String name = request.getParameter("name");
    out.println("Hello " + name);
%>
</body>
</html>
```

Primjer: NDepend

The screenshot shows the Microsoft Visual Studio interface with the NDepend extension installed. The NDepend menu is open, displaying various analysis options and status information.

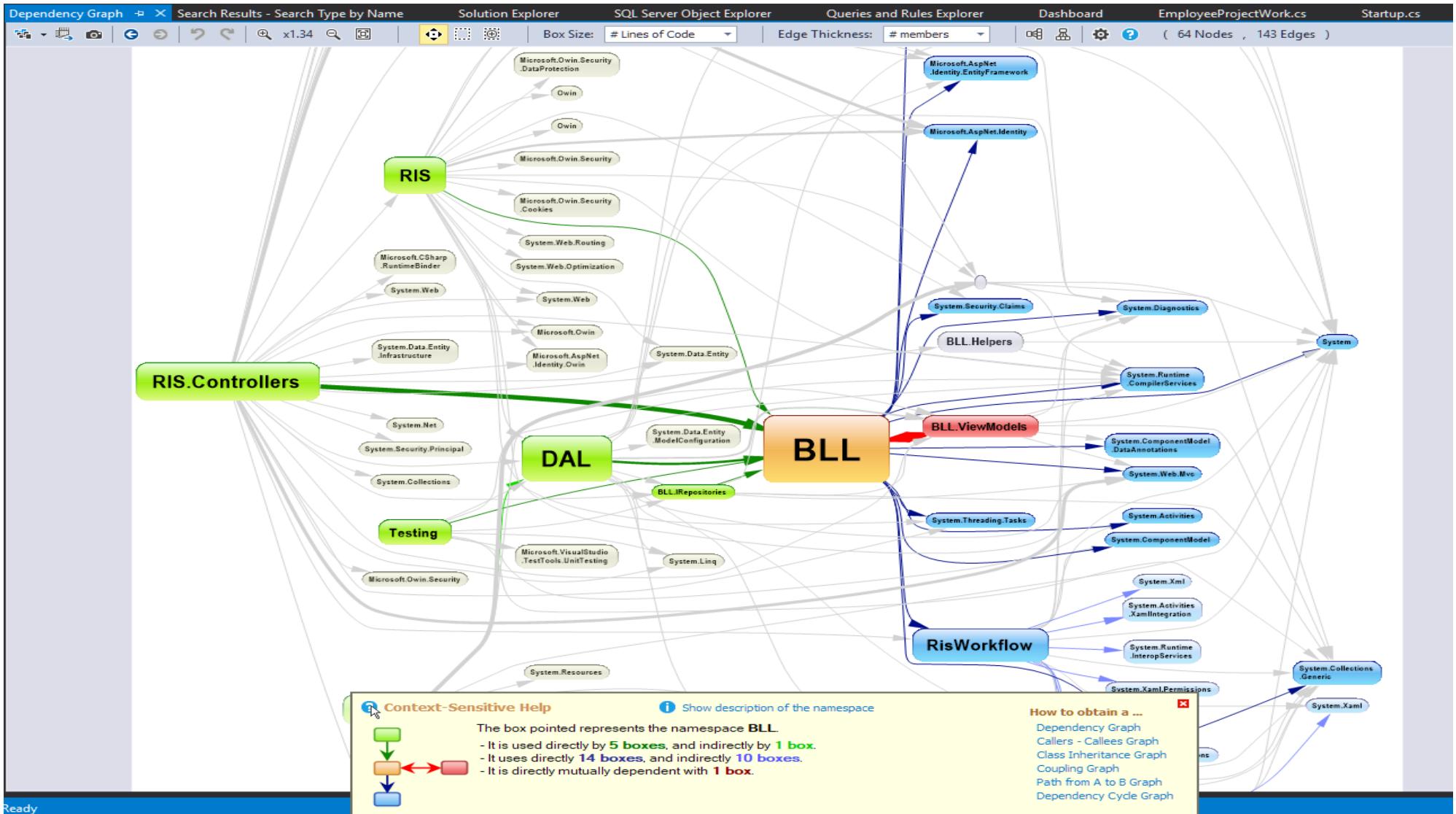
Open NDepend Context Menu:

- Dashboard
- Rules
- Issues
- Graph
- Matrix
- Diff
- Trend
- Metrics
- Coverage
- Search
- Project
- Analyze
- Report
- Tools
- Options...
- Windows
- Help

Status Summary:

- View Explorer Panel
- View Editor Panel
- New ...
- 2 Quality Gates Fail
- 0 Quality Gate Warn
- 9 Quality Gates Pass
- 4 Critical Rules Violated
- 29 Rules Violated
- 112 Rules Ok
- More Selections
- Code Smells
- Object Oriented Design
- Design
- Architecture
- Dead Code
- Visibility
- Immutability
- Naming Conventions
- Source Files Organization
- .NET Framework Usage
- Defining JustMyCode
- Trend Metrics
- Code Diff Summary
- Statistics
- Samples of Custom rules
- Rules extracted from Source Code

Primjer: zavisnost komponenti aplikacije



Dinamička provjera

Dynamic Analysis
Fuzzing
Penetration Testing

Fuzzing - "pročešljavanje" (eng. fuzz = dlačica)

- ◆ DAST (The Good, the Bad and the Ugly)
 - ubrizgavanje kvara u aplikaciju (fuzzing, fuzz testing)
 - slanje neispravnih, neočekivanih ili nasumičnih podataka ulazu programa
 - slično regresiji, samo s lošim podacima
 - „češljanje” aplikacija, protokola, datoteka
- ◆ 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 uzorka 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
 - podaci generirani na temelju modela
 - zahtijeva dubinsko poznavanje cilja i specijalizaciju
 - smišljene anomalije poznavanjem formata, standarda
 - posljedica
 - manja potreba za analizom
 - manje dupliciranje nalaza

Standard HTTP GET request

GET /index.html HTTP/1.1

Anomalous requests

AAAAAA...AAAA /index.html
HTTP/1.1

GET %n%n%n%n%n%n.html HTTP/1.1

GET /AAAAAAAAAAAAAA.html HTTP/1.1

GET /index.html

HTTTTTTTTTTTTP/1.1

```
GET /index.html  
HTTP/1.1.1.1.1.1.1.1.1.1
```

Alati za dinamičku analizu

- https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools
- CERT Basic Fuzzing Framework (BFF) i Failure Observation Engine (FOE)
 - Otvoreni kod <https://github.com/CERTCC/certfuzz>
- Peach Fuzzer - *automated security testing platform*
- WebScarab - analiza aplikacija koje koriste HTTP/HTTPS, *intercepting proxy*
- Burp - web aplikacije, buffer overflow, CSS, SQL injection, ...
- Fuddly - *fuzzing and data manipulation framework (for GNU/Linux)*
- Hongfuzz - *general fuzzer*
- *Profileri, ...*

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)
-
- ◆ Svrha
 - Potvrda funkcionalnosti sigurnosnih kontrola
 - Pravovremeno uočavanje sigurnosnih propusta
 - Prevencija sigurnosnih incidenata
 - Opravdavanje investicije
 - Ispunjavanje regulatornih zahtjeva

Pristup penetracijskom testiranju

- ◆ 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 - s udaljene lokacije (Interneta) prema javno dostupnim sustavima
 - Unutrašnji - s intraneta, simulacija incidenta neovlaštenog pristupa unutrašnjoj mrežnoj infrastrukturi
- ◆ Ostali kriteriji
 - Opseg, prikrivenost, tehnike, agresivnost

Izvođenje penetracijskog testa

- ◆ **Istraživanje** (eng. *reconnaissance*), izviđanje
 - ispitivač pokušava prikupiti što više informacija.
 - pasivno - javno dostupne informacije (npr. podaci s društvenih mreža, Google)
 - aktivno - istraživački alati (npr. *nslookup*), da bi se odredili određeni parametri
- ◆ **Skeniranje** (eng. *scanning*)
 - ispitivač skenira otvorene portove (*port scanning*) korištenjem alata (npr. Nmap)
 - cilj - enumeracija servisa, verzije enumeriranih servisa i OS (*OS and service fingerprinting*).
 - skeniranje ranjivosti (*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 zločudne *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

Alat	Osnovna funkcionalnost	Faza
Harvester	Istraživanje javno dostupnih informacija korištenjem tražilica, društvenih mreža itd.	Istraživanje
Nmap	Istraživanje mreže i skeniranje portova.	Skeniranje
QualysGuard (Network)	Mrežno skeniranje poznatih ranjivosti.	Skeniranje
QualysGuard (WAS)	Skeniranje ranjivosti web aplikacija.	Skeniranje
ASPAuditor	Identifikacija ranjivih i loše konfiguiranih ASP.NET poslužitelja.	Skeniranje
Nikto	Skeniranje web poslužitelja na razne propuste, kao što opasne datoteke itd.	Skeniranje
ZAP	Multifunkcionalni alat za penetracijsko testiranje web aplikacija.	Skeniranje
Sqlmap	Otkrivanje i iskorištavanje ranjivosti SQL umetanja.	Provodenje napada
Metasploit	Multifunkcionalna platforma za iskorištavanje ranjivosti.	Provodenje napada
HTTP DoS	Uskraćivanje usluge na aplikacijskom sloju.	Provodenje napada
Hydra	Udaljeno probijanje lozinki.	Provodenje napada
Wireshark	Snimanje i analiza mrežnog prometa.	Provodenje napada

Primjer: Nmap i QualysGuard

```
root@bt:~# nmap -sS -sV 22.22.22.22
Host is up (0.0027s latency).
Not shown: 992 filtered ports
PORT      STATE    SERVICE          VERSION
21/tcp     open     ftp              Microsoft ftpd
25/tcp     open     smtp?
80/tcp     open     http             Microsoft IIS httpd 6.0
110/tcp    open     pop3            Microsoft Windows 2003 POP3 Service1.0
443/tcp    open     ssl/http        Microsoft IIS httpd 6.0
3389/tcp   open     microsoft-rdp Microsoft Terminal Service
```

Summary of Vulnerabilities

Vulnerabilities Total	24	Security Risk (Avg)	5.0
by Severity			
Severity	Confirmed	Potential	Information Gathered
5	1	0	 5
4	0	0	
3	0	0	
2	1	0	
1	0	0	
Total	2	0	

5 Microsoft Windows Remote Desktop Protocol Remote Code Execution

Category	Confirmed	Potential
Information gathering	0	0
TCP/IP	0	0
Windows	2	0
Web server	0	0
CGI	0	0
Total	2	0

QID: 90783
Category: Windows
CVE ID: CVE-2012-0002, CVE-2012-0152
Vendor Reference: MS12-020
Bugtraq ID: -
Service Modified: 03/29/2012
User Modified: -

Primjer: Hydra, Wireshark

- ◆ Napad rječnikom alatom Hydra

```
[STATUS] 446.66 tries/min, 187152 tries in 06:59h, 0 todo in 00:01h
[STATUS] attack finished for 22.22.22.22 (waiting for children to finish)
1 of 1 target successfully completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2012-06-10 17:20:30
```

- ◆ Autentifikacijski podaci tijekom napada snimljeni alatom Wireshark

23992	531.375	192.168.235.128	FTP	72	Request: PASS zagreb0702!
23993	531.375	192.168.235.128	TCP	60	ftp > 48027 [ACK] Seq=83068 Ack=3674
23994	531.481	192.168.235.128	FTP	87	Response: 331 Password required for
23995	531.481	192.168.235.128	FTP	72	Request: PASS zagreb0703!
23996	531.481	192.168.235.128	TCP	60	ftn > 48028 [ACK1 Seq=85051 Ack=3762

File Transfer Protocol (FTP)

PASS zagreb0703!\r\n

Request command: PASS

Request arg: zagreb0703!

0010

...@.U.....>

Primjer: Metasploit

```
msf > use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf auxiliary(ms12_020_maxchannelids) > show options
Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):
Name      Current Setting  Required  Description
----      -----          -----      -----
RHOST                yes        The target address
RPORT      3389           yes        The target port
msf auxiliary(ms12_020_maxchannelids) > set RHOST 11.11.11.11
RHOST => 11.11.11.11
msf auxiliary(ms12_020_maxchannelids) > exploit
[*] 11.11.11.11:3389 - Sending MS12-020 Microsoft Remote Desktop Use-After-Free DoS
[*] 11.11.11.11:3389 - 210 bytes sent
[*] 11.11.11.11:3389 - Checking RDP
[+] 11.11.11.11:3389 seems down
A problem has been detected and Windows has been shut down to prevent damage to your computer.
If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:
Check to be sure you have adequate disk space. If a driver is identified in the Stop message, disable the driver or check with the manufacturer for driver updates. Try changing video adapters.
Check with your hardware vendor for any BIOS updates. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup options, and then select Safe Mode.
Technical information:
*** STOP: 0x00000008E (0xC0000005,0x8DA6F987,0x931778F0,0x00000000)
***      termdd.sys - Address 8DA6F987 base at 8DA6E000, DateStamp 4ce7a116
Collecting data for crash dump ...
Initializing disk for crash dump ...
```

Alati za penetracijsko testiranje i detekciju upada

- ◆ Pentest
 - https://www.owasp.org/index.php/Category:Penetration_Testing_Tools
 - Aircrack-ng - WIFI skaner - otvoreni kod
 - Burp Suite - skaner web ranjivosti
 - Cain & Abel - „password recovery tool” - packet sniffer, password cracker, ...
 - Ettercap - suite for man in the middle attacks - otvoreni kod
 - John The Ripper - password cracker
 - Nessus - skener ranjivosti - free trial
 - Kismet - mrežni detektor, packet sniffer, IDS - freeware
 - Zed Attack Proxy (**ZAP**) - web application security scanner - Apache 2 License
- ◆ ID/PS
 - <https://www.comparitech.com/net-admin/network-intrusion-detection-tools/>
 - Snort, OSSEC, ..., Solarwinds Log and Event Manager, ...

Primjer: Snort

Services / Snort / Alerts ?

Snort Interfaces Global Settings Updates **Alerts** Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

[Clear all interface log files](#)

Alert Log View Settings

Interface to Inspect: WAN Auto-refresh view Choose interface.. Alert lines to display.

Alert Log Actions:

Alert Log View Filter +

Last 1000 Alert Log Entries

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066	<input type="button" value="Q"/> <input type="button" value="⊕"/>	16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465	<input type="button" value="Q"/> <input type="button" value="⊕"/>	5060	140:26	(spp_sip) Method is unknown
2017-07-21 09:26:30	2	UDP	Potentially Bad Traffic	163.172.22.169	52428	<input type="button" value="Q"/> <input type="button" value="⊕"/>	5060	140:26	(spp_sip) Method is unknown
2017-07-21 01:03:28	2	UDP	Potentially Bad Traffic	163.172.17.76	46834	<input type="button" value="Q"/> <input type="button" value="⊕"/>	5060	140:26	(spp_sip) Method is unknown
2017-07-20 20:36:37	2	UDP	Potentially Bad Traffic	163.172.22.169	54788	<input type="button" value="Q"/> <input type="button" value="⊕"/>	5060	140:26	(spp_sip) Method is unknown
2017-07-20 08:31:30	2	UDP	Potentially Bad Traffic	163.172.17.76	59571	<input type="button" value="Q"/> <input type="button" value="⊕"/>	5060	140:26	(spp_sip) Method is unknown

Reference

- OWASP Category: Vulnerability Scanning Tools
- OWASP Code Review Guide
- OWASP Testing Guide
- ◆ Još alata ...
 - <http://sectools.org/>
 - https://www.owasp.org/index.php/Appendix_A:_Testing_Tools
 - [Software Security Assessment Tools Review, 2009](#)



Zaštita i sigurnost informacijskih sustava

Upravljanje (softverskim) rizikom

prof. dr. sc. Krešimir Fertalj

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 licencom 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/>.

Rizik i upravljanje 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 - jednostavne liste rizika, jedan član ekipe (ne voditelj)
 - Veliki projekti - formalno upravljanje rizikom, *risk officer*, puno radno vrijeme

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

Procjena 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.

Prepoznavanje rizika

- ◆ Rizici rokova
 - ◆ Rizici planiranja
 - ◆ Rizici organizacije i upravljanja
 - ◆ Rizici razvojnog okruženja
 - ◆ Rizici krajnjeg korisnika
 - ◆ Rizici naručitelja
 - ◆ Rizici (pod)ugovaratelja
 - ◆ Rizici zahtjeva
 - ◆ Rizici aplikacije
 - ◆ Rizici vanjskih utjecaja
 - ◆ Rizici razvojne ekipe
 - ◆ Rizici dizajna i ugradnje
 - ◆ Rizici procesa
-
- ◆ Primjeri najčešćih grupa rizika navedeni su u dodatku
 - ◆ Osim općenitih rizika, svaki projekt nosi svoje vlastite rizike
 - npr. važan član tima prijeti otkazom ne bude li mogao dovoditi svog psa na posao

Analiza 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
 - alternativno financijski gubitak u novčanim jedinicama
- ◆ **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*.

Primjer procjene rizika

Rizik	Vjerovatnost gubitka	Veličina gubitka (u tjednima)	Izloženost riziku (u tjednima)	
Preoptimističan plan razvoja	50%	5	2.5	
Dodatni zahtjevi za potpunim podržanjem automatskog ažuriranja programskih verzija	5%	20	1.0	
Dodatne funkcionalnosti prema zahtjevima marketinga (posebne funkcionalnosti nepoznate)	35%	8	2.8	
Nestabilan grafički podsustav korisničkog sučelja	25%	4	1.0	Identificirani su potencijalni rizici, te njihov utjecaj na vremenska kašnjenja projekta.
Neprikladan dizajn koji zahtjeva redizajn	15%	15	2.25	
Odobrenje projekta traje dulje od očekivanog	25%	4	1.0	Utjecaj, 1 do 20 tjedana kašnjenja s vjerovatnošću pojedinog rizika 5% do 50%.
Sredstva za rad nisu dostupna na vrijeme	10%	2	0.2	
Izvješća od strane menadžmenta zahtjevaju više razvojnog vremena od očekivanog	10%	1	0.1	
Kašnjenje kontraktora u isporuci grafičkog podsustava	10-20%	4	0.4-0.8	
Novi programerski alat ne donosi obećane uštede	30%	5	1.5	

Procjena veličine gubitka

- Obično je lakše procijeniti veličinu gubitka od vjerojatnosti pojave
- ◆ Za primjer u prethodnoj tablici,
 - neka je procijenjeno da će projekt biti odobren 1.veljače ili 1.ožujka, ovisno o tome kada će nadzorni raspravljati o prijedlogu projekta
 - Veličina rizika odobrenja 1. ožujka je točno jedan mjesec
- ◆ 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 dajem/dobivate 125 eur, inače ja dobivam 100 eur"
 - Oklada se prepravlja sve dok obje strane ne budu zadovoljne
 - Vjerojatnost rizika je rezultat dijeljenja dobitka ponuditelja oklade i ukupnog iznosa.
 - Za navedeni primjer rizika, vjerojatnost = $100 / (100 + 125) = 44\%$
- ◆ **Postupak pridjevne kalibracije ("adjective calibration")**
 - odredi se razina rizika opisno (npr. vrlo vjerojatno, vjerojatno, ..., malo vjerojatno)
 - zatim se opisne procjene kvantificiraju [Boehm 1989] # bolje kvalitativno

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

Procjena rizika uređena prema prioritetima

Rizik	Vjerovatnost gubitka	Veličina gubitka (u tjednima)	Izloženost riziku (u tjednima)
Dodatne funkcionalnosti prema zahtjevima marketinga (posebne funkcionalnosti nepoznate)	35%	8	2.8
Preoptimističan plan razvoja	50%	5	2.5
Neprikladan dizajn koji zahtijeva redizajn	15%	15	2.25
Novi programerski alat ne donosi obećane uštede	30%	5	1.5
Dodatni zahtjevi za potpunim podržanjem automatskog ažuriranja programskih verzija	5%	20	1.0
Nestabilan grafički podsustav korisničkog sučelja	25%	4	1.0
Odobrenje projekta traje dulje od očekivanog	25%	4	1.0
Kašnjenje kontraktora u isporuci grafičkog podsustava	10-20%	4	0.4-0.8
Sredstva za rad nisu dostupna na vrijeme	10%	2	0.2
Izvješća od strane menadžmenta zahtijevaju više razvojnog vremena od očekivanog	10%	1	0.1

Komentar prioriteta prema izloženosti

- ◆ Poredak rizika u tablici daje grubu procjenu prioriteta rizika
 - Uspješno rješavanje prvih 5 rizika donosi uštedu od 9.8 tjedana
 - Rješavanje 5 posljednjih štedi od 2.1 do 3.7 tjedana
- ◆ Tablični sort je **gruba procjena** prioriteta
 - Rizici s velikim iznosima gubitaka bi možda trebali biti bliže vrhu
 - Npr., "Dodatni zahtjevi ..." ima vjerojatnost 5%, ali povlači gubitak od 20t
 - nužno je osigurati da se taj rizik ne dogodi iako je malo vjerojatan
- ◆ Ponekad je važnija **kombinacija** u odnosu na pojedinačne rizike
 - Primjer, **Nestabilnost sučelja ...** i **Kašnjenje ugovaratelja ...**
 - Kombinacija ima veći rizik nego pojedinačni

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

Kontrola rizika

- ◆ 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
- ◆ Razrješenje rizika (risk resolution)
 - Izbjegavanje - Avoidance (eliminate, withdraw from or not become involved)
 - Preusmjeravanje, dijeljenje - Sharing (transfer, outsource or insure)
 - Smanjenje - Reduction (optimize – mitigate)
 - Prihvaćanje - Retention (accept and budget)

Plan upravljanja rizikom i razrješenje rizika

- ◆ Plan upravljanja rizikom
 - 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
 - Što učiniti ?

Opći postupci razrješenja rizika

- ◆ **Izbjegavanje rizika - ne preuzeti rizik ili ukloniti uzrok**
 - Pr.1.a: preuzeti odgovornost samo za poznati dio, nepoznato prepustiti klijentu ?!
 - Pr.1.b: promijeniti doseg projekta – problem postane dio druge verzije ili projekta
 - Pr.2.a: nagovoriti grupu koja pretendira na prostor da odustane (potpuno)
 - Pr.2.b: navesti konkurenčiju 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

Opći postupci razrješenja rizika (nastavak)

◆ 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 neizbjegljiva, treba ju provesti u trenutku kada najmanje utječe na rad i organizirati pomoć pri pakiranju i selidbi

◆ Prihvatanje rizika

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

Opći postupci razrješenja rizika (ostalo)

- ◆ 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 iznenadenje u slučaju da se rizik dogodi
- ◆ Evidencija rizika
 - zbarka planova razrješenja, što se može iskoristiti u budućim projektima

[neki] Mehanizmi kontrole

Rizična situacija	Način kontrole
Neplansko dodavanje novih programskih karakteristika	Orientirati se prema klijentu Koristiti postupke postupnog razvoja Kontrolirati skup mogućnosti aplikacije Osigurati rezervni dizajn
Pretjerivanje u zahtjevima ili razvoju	Ne ispunjavati sve zahtjeve Postaviti vremenski okvir za zahtjeve Kontrolirati skup mogućnosti aplikacije Koristiti postupnu isporuku Koristiti sustav testnih prototipova Dizajnirati prema vremenskom planu Odvojiti vrijeme za pitanja i odgovore, te posvetiti pozornost temeljima osiguranja kvalitete
Smanjenje kvalitete zbog prekratkog vremenskog roka Preoptimističan plan razvoja	Koristiti postupke za vremensku procjenu, višestruke procjene, te alate za automatsku procjenu Koristiti taktiku pregovaranja Dizajnirati prema vremenskom planu Koristiti postupke postupnog razvoja
Neprikladan dizajn	Dizajn postaviti kao zasebnu aktivnost za koju će planom biti predviđeno vrijeme Koristiti inspekcije dizajna
Sindrom srebrnog metka	Biti skeptičan prema zahtjevima koji se odnose na produktivnost Napraviti program za mjerjenje kvalitete programske podrške Utemeljiti grupu koja će brinuti o programskim alatima
Razvoj okrenut istraživanjima	Ne pokušavati istovremeno istraživati i maksimizirati razvojnu brzinu Koristiti plan orientiran prema rizičnim situacijama
Loše osoblje	Zaposliti najtalentiranije osoblje Zaposliti i rasporediti ključne djelatnike puno prije početka projekta Obučiti djelatnike Izgraditi tim
Pogreške kontraktora	Provjeriti reference kontraktora

Nadgledanje rizika

- ◆ Nestabilnost rizika
 - rizici se pojavljuju, povećavaju/smanjuju, nestaju s vremenom
 - trajno nadgledanje i mjerjenje
- ◆ „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

Ovaj tjedan	Prošli tjedan	Broj tjedana na listi	Rizik	Koraci koji su poduzeti da bi se rizik smanjio
1	1	5	Neplansko dodavanje novih programskih karakteristika	Prihvaćen je postupni sustav isporuke; potrebno obrazložiti marketingu i krajnjim korisnicima.
2	5	5	Neprikidan dizajn, koji zahtijeva redizajn	U tijeku je redizajn.
3	2	4	Probni voditelj projekta još nije preuzeo posao	Posao je ponuđen najboljem kandidatu; čeka se prihvatanje ponude.
4	7	5	Nestabilna grafički podsustav sučelja	Dizajn grafičkog sučelja je stavljen u prednji plan projekta; dizajn još nije završen.
5	8	5	Kontraktor kasni u isporuci grafičkog podsustava	Iskušna osoba je imenovana za vezu s kontraktorom; još uvijek nema odgovora od kontraktora.
6	4	2	Razvojni alati kasne u isporuci	5 od 7 razvojnih alata je OK. Obaviještena je grupa određena za nabavu razvojnih.
7	-	1	Spor ciklus recenzije od strane menadžera.	Evaluacija u tijeku.
8	-	1	Ciklus recenzije od strane korisnika je spor	Evaluacija u tijeku.
9	3	5	Preoptimističan plan	Prva je etapa projekta dovršena na vrijeme.
10	9	5	Dodatni zahtjev za automatskim ažuriranjem verzije	Istražiti izvedivost ručnog ažuriranja verzije.
-	6	5	Voditelj dizajna je preokupiran zahtjevima za podrškom prethodnom projektu	Prethodni projekt je dobio novog voditelja.

Lista 10 najvećih rizika

Kvalitativna procjena rizika

- ◆ numerički, ali relativnim vrijednostima
- ◆ Matrica preddefiniranih vrijednosti

		Impact		
		Low	Medium	High
Probability	Low	Low Risk	Low Risk	Medium Risk
	Medium	Low Risk	Medium Risk	High Risk
	High	Medium Risk	High Risk	High Risk

Qualitative risk assessment table

- razina rizika kao suma vrijednosti sredstva (AV), ranjivosti (V) i prijetnje (T), pr.

- AV u rasponu od 0 (mala) do 4 (vrlo velika)
- V i T raspon od 0 (niska razina) do 2 (visoka razina)
- $R = AV + V + T$

■ Vrijednosti 0-8

- Nisko (M): 0 - 2
- Srednje (S): 3 - 5
- Visoko (V): 6 - 8

	Prijetnja	0			1			2		
	Ranjivost	0	1	2	0	1	2	0	1	2
Vrijednost resursa	0	0	1	2	1	2	3	2	3	4
	1	1	2	3	2	3	4	3	4	5
	2	2	3	4	3	4	5	4	5	6
	3	3	4	5	4	5	6	5	6	7
	4	4	5	6	5	6	7	6	7	8

■ Karakteristike

- Proizvoljno određivanje parametara, jednostavnost
- zanemarivanje većih rizika zbog (funkcije) distribucije (ne gleda vjerojatnost i posljedicu)

Dodatak

Vrste rizika

Rizici vremenskih rokova

- Neplansko dodavanje karakteristika koje narušavaju dizajn
 - Nepotrebno usavršavanje pojedinih dijelova aplikacije
 - Smanjenje kvalitete radi sustizanja vremenskih rokova
 - Preoptimistični vremenski rokovi razvoja
 - Neprikladan dizajn programske podrške
 - Sindrom srebrnog metka
 - Istraživački orijentiran razvoj
 - Slab razvojni tim
 - Pogreške za koje su odgovorni podugovarači
 - 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/OP u odnosu na surrogate).
- Potreban napor veći je od predviđenog.
- Pretjerani pritisak na članove tima zbog rokova smanjuje njihovu produktivnost.
- Vremenski rok je promijenjen bez potrebnih preinaka u procesu ili resursima.
- 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.
- Napuštanje plana projekta pod pritiskom rokova.

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.
- Isteč 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 zahtjeva 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 zahtjeva 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.

Rizici aplikacije

- Moduli koji uzrokuju najviše pogrešaka iziskuju više napora od očekivanog.
- Loša kvaliteta aplikacije iziskuje više napora od očekivanog.
- (Nepotrebno) usavršavanje produljuje vrijeme razvoja.
- Razvoj pogrešnih programskih funkcija zahtjeva redizajn i implementaciju.
- Neprihvatljivo korisničko sučelje zahtjeva redizajn.
- Problemi odziva ili količine podataka zahtjevaju više vremena od očekivanog.
- Strogi zahtjevi na kompatibilnost ...
- Zahtjevi postavljeni na sučelja s drugim sustavima ...
- Zahtjev na prenosivost na druge OS ...
- Rad u nepoznatom/neprovjerenom okruženju uzrokuje nepredviđene probleme.
- Razvoj nove atipične komponente traje predugo.
- Oslanjanje na nezrelu tehnologiju produljuje razvoj programske podrške.

Rizici razvojne ekipe

- Zapošljavanje članova razvojnog tima traje dulje od predviđenog.
- Preuvjeti (npr. osposobljavanje, završetak drugih projekata, radne dozvole) ...
- Loši odnosi članova tima i/ili uprave usporavaju odlučivanje i provođenje.
- Članovi tima nisu dovoljno angažirani, a posljedica su slabe performanse.
- Niski moral i motiviranost smanjuju produktivnost.
- Nedostatak potrebnih specijalizacija uzrokuje nedostatke i preradu.
- Članovi trebaju dodatno vrijeme za upoznavanje s alatom, okruženjem, opremom.
- Vanjski suradnici napuštaju projekt prije završetka projekta.
- Stalni zaposlenici odlaze prije završetka projekta.
- Novi članovi se kasno uključuju u projekt, što smanjuje učinkovitost postojećih.

Rizici razvojne ekipe (nastavak)

- Članovi tima ne rade učinkovito zajedno.
- Sukobi između članova tima – loša komunikacija, dizajn, prerada, ...
- Problematični članovi – narušena motivacija i odnosi ekipe.
- Najsposobniji članovi nisu dostupni zbor političkih ili drugih razloga.
- Nije moguće angažirati članove s potrebnim osobinama.
- Ključni djelatnici nisu dostupni puno radno vrijeme.
- Nije dostupan dovoljan broj djelatnika za rad na projektu.
- Zadaci povjereni članovima tima nadilaze njihove snage.
- Djelatnici rade sporije od očekivanog.
- Menadžeri „sabotiraju“ projekt - neučinkovito planiranje.
- Tehničko osoblje „sabotira“ projekt ... neiskoristiv posao ili loša kvaliteta

Rizici dizajna i ugradnje

- Pojednostavljeni dizajn prema zahtjevima – redizajn i ponovna ugradnja.
- Presložen dizajn uzrokuje nepotrebnu i neproduktivnu implementaciju.
- Loš dizajn uzrokuje ponovni dizajn i ponovnu implementaciju.
- Neprikladni postupci – dodatna poduka, rad, ispravljanje pogrešaka.
- Implementacija u neprikladnom / zastarjelom jeziku – produktivnost ...
- Neodgovarajuće programske knjižnice – nadomjesne ili razvoj vlastitih.
- Nekvalitetan prog. kod / knjižnice - dodatno testiranje, ispravke i prerada.
- Precijenjene vremenske uštede očekivane od alata.
- Teškoća integracije zasebno razvijenih komponenti – redizajn, dorada.

Rizici razvojnog procesa

- Prevelika količina administrativnog posla.
- Neprecizno praćenje razvoja – prekasna procjena stvarnog stanja.
- Skraćeno testiranje u raznim fazama – značajnije prerađe u kasnijim.
- Neprecizno praćenje kvalitete – prekasno saznanje o lošoj kvaliteti.
- Premalo formalizma (standarda) - loša komunikacija, kvaliteta - prerađe.
- Previše formalizma (slijepo pridržavanje) - nepotreban dodatni posao.
- Izvješća o tijeku razvoja „prema gore“ iziskuju više vremena od očekivanog.
- Loša analiza poslovnih rizika nije otkrila najveće projektne rizike.
- Analiza projektnih rizika iziskuje više vremena od očekivanog.



Zaštita i sigurnost informacijskih sustava

Planiranje kontinuiteta poslovanja za nepredviđene slučajeve

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 licencom 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.

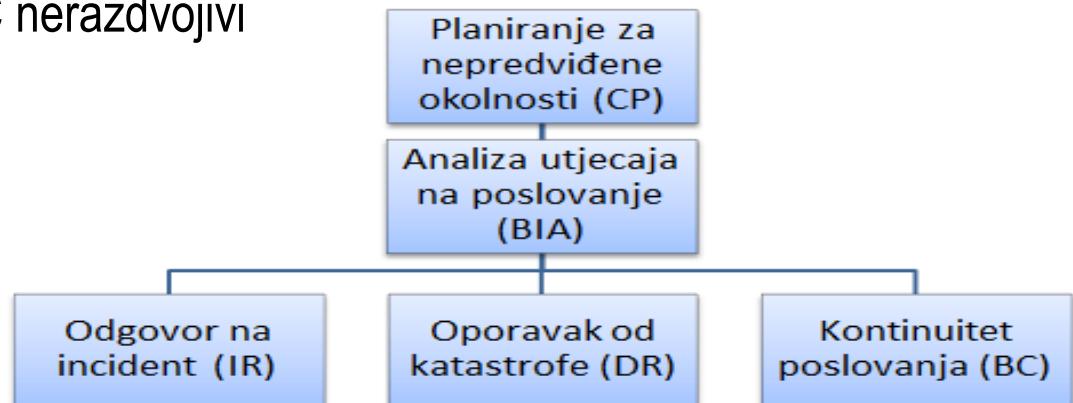
Tekst licencije preuzet je s <http://creativecommons.org/>.

Osnovni pojmovi

- ◆ Štetni događaj (adverse event)
 - Događaj s negativnim posljedicama koji bi mogao ugroziti resurse ili operacije organizacije – napad, sabotaža, potres, poplava, požar, curenje plina, radijacija, ...
 - Mogući kandidat za incident
- ◆ Incident
 - Štetni događaj koji može rezultirati gubitkom informacijske imovine, ali trenutno ne prijeti održivosti čitave organizacije
 - Jasno identificirani napad na informacijsku imovinu koji može ugroziti njenu povjerljivost, cjelovitost ili raspoloživost
- ◆ Katastrofa (disaster)
 - Štetni događaj koji bi mogao ugroziti održivost čitave organizacije
 - Eskalira iz incidenta ili odmah bude proglašena

Planiranje za nepredviđene situacije

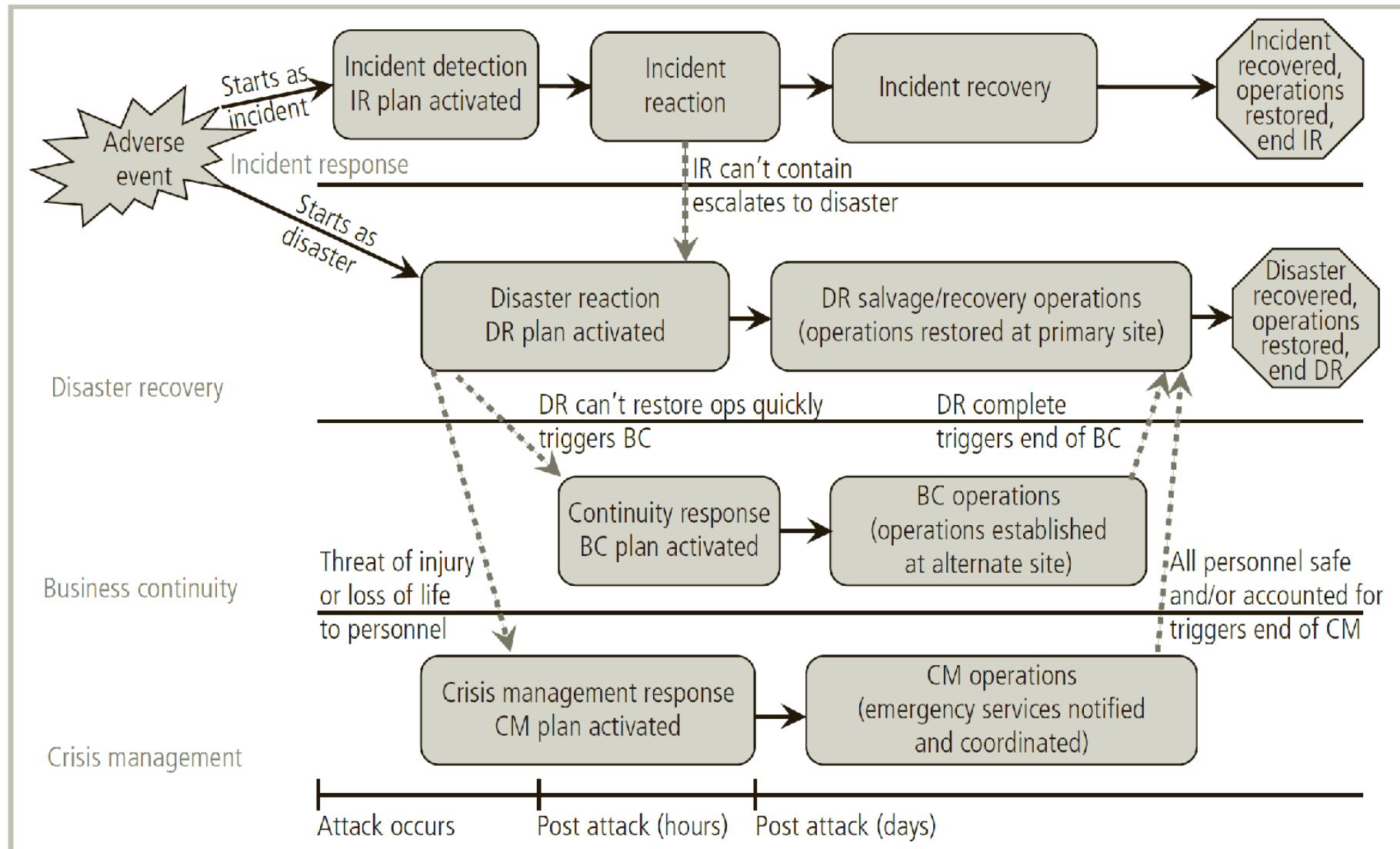
- ◆ Planiranje za nepredviđene situacije (Contingency planning - CP)
 - više rukovodstvo odredi što kada štetni događaj postane incident ili katastrofa
- ◆ Elementi
 - Analiza utjecaja na poslovanje (Business Impact Analysis - BIA)
 - Planiranje odgovora na incidente (IR), oporavka od katastrofe (DR) i kontinuiteta poslovanja (BC)
 - Planiranje nastavka poslovanja (Business Resumption Planning – BRP) = DRP + BCP
 - Smatra se da su planovi DR i BC nerazdvojivi

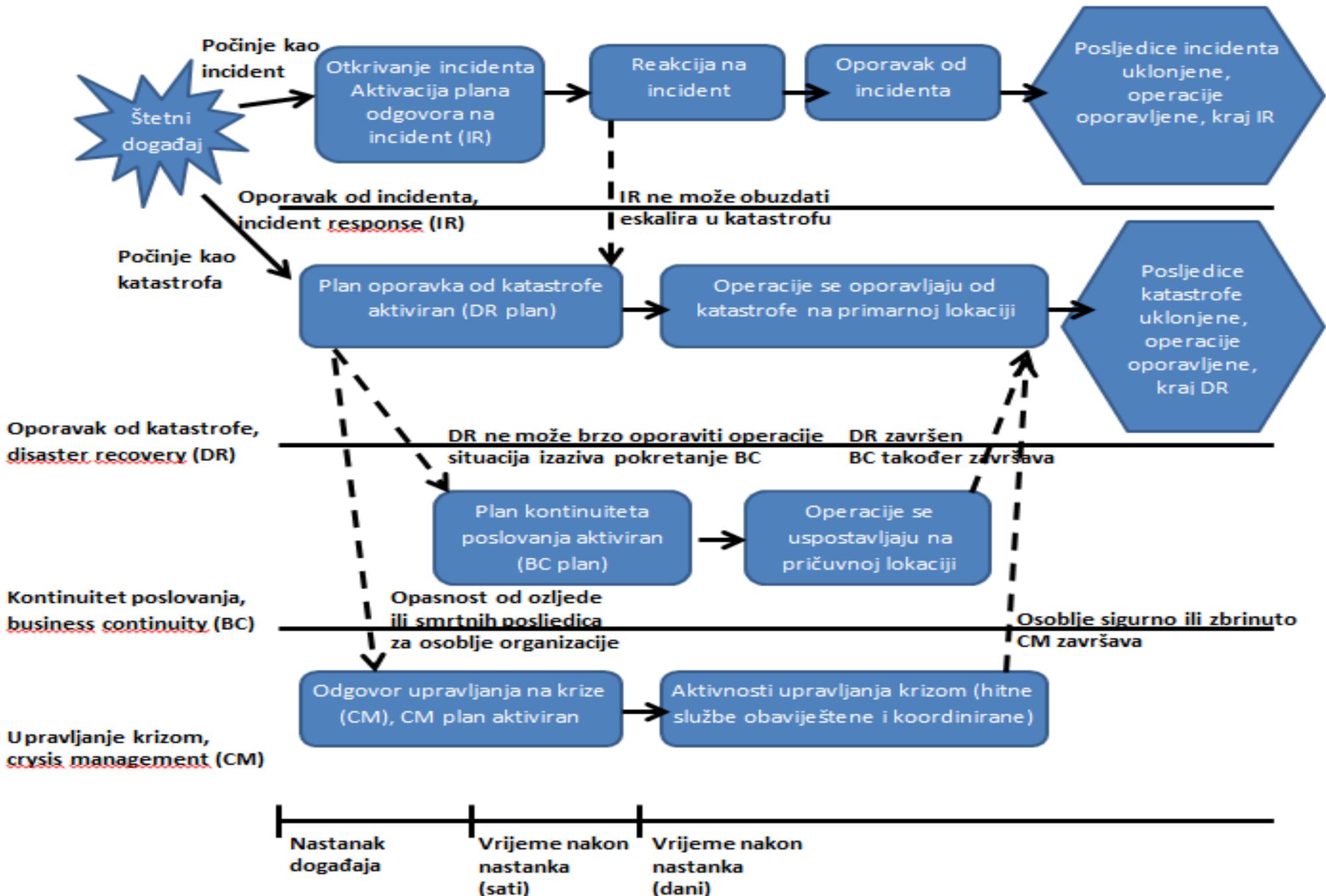


Planovi

- ◆ Plan za nepredviđene situacije (contingency plan)
 - Organizacija priprema kako bi se preduhitrili, reagirali i oporavili od događaja koji su prijetnja sigurnosti i informacijskoj imovini, te postupno doveli organizaciju u normalan tok rada
- ◆ Plan za odgovora na incident (Incident Response Plan – IR plan)
 - Prva, neposredna reakcija - ako situacija eskalira proširuje se na DRP i/ili BCP
- ◆ Plan oporavka od katastrofe (Disaster Recovery Plan – DR plan)
 - Obnavljanje sustava **na originalnoj lokaciji** nakon pojave katastrofe
- ◆ Plan kontinuiteta poslovanja (Business Continuity Plan – BC plan)
 - Konkurentno, održivost ključnih poslovnih funkcija, kad je šteta velika ili traje
 - Uspostavlja kritične poslovne funkcije **na alternativnom mjestu - pričuvnoj lokaciji**
- ◆ Dodatno, upravljanje krizom (Crisis Management – CM)
 - Bavljenje ozljedama, traumama i gubitkom života kao posljedicama katastrofe

Raspored planiranja nepredviđenih situacija





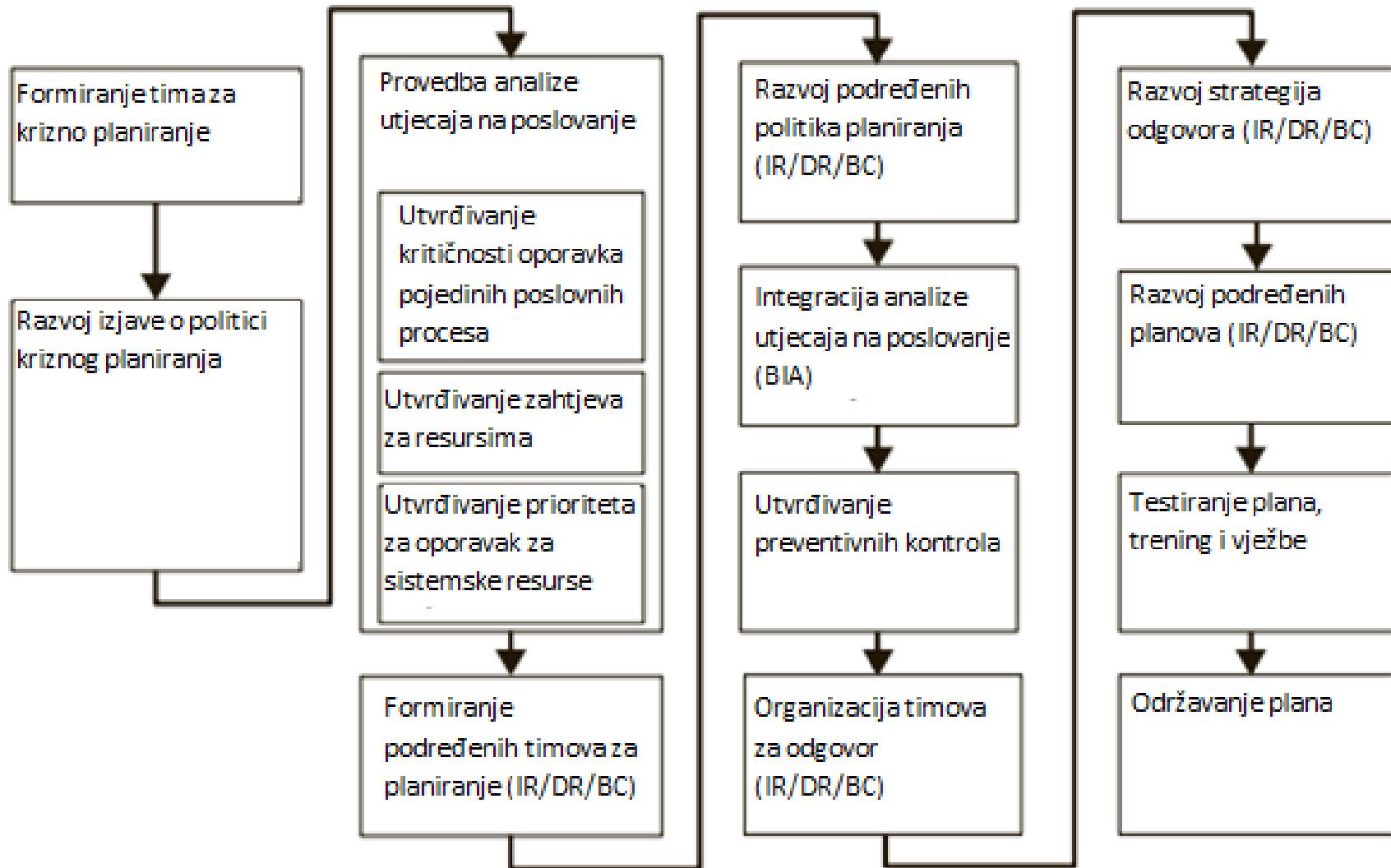
Tim za upravljanje planiranjem nepredviđenih situacija (CPMT)

- ◆ Tim za upravljanje planiranjem u nepredviđenim situacijama (CPMT)
 - Grupa viših menadžera i članova projekta organizirani da pro/vode sve napore CP
 - Formiranje tima i dodjela uloga prije nego započne planiranje
- ◆ Prvak, šampion (champion)
 - Viši rukovoditelj – potpora, promicanje, podržavanje
 - Idealno CIO (voditelj informatike) ili CEO (izvršni direktor)
- ◆ Voditelj projekta (project manager)
 - Srednji rukovoditelj ili CISO (chief information security officer)
- ◆ Članovi tima
 - Rukovoditelji ili predstavnici: poslovanje, IT, informacijska sigurnost

Cjelokupni proces planiranja za nepredviđene situacije

- ◆ Razvoj politike CP
 - Osiguranje autoriteta i smjernica za učinkovito planiranje
- ◆ Provedba BIA
 - Identifikacija i određivanje prioriteta ključnih IS za poslovne procese organizacije
- ◆ Određivanje preventivnih kontrola
 - Mjere za smanjenje učinaka poremećaja sustava i povećanje dostupnosti
- ◆ Izrada strategija za nepredviđene situacije
 - Strategije oporavka za brzi i učinkovit oporavak
- ◆ Razvoj plana za nepredviđene situacije
 - Detaljne preporuke i procedure za obnovu objekata prema zahtjevima za svaku organizacijsku cjelinu
- ◆ Osiguranje plana provjere, treninga i uvježbavanja
 - Provjera sposobnosti oporavka, trening i uvježbavanje osoblja
- ◆ Osiguranje održavanja plana
 - Periodičko ažuriranje sukladno poboljšanjima sustava i organizacijskim promjenama

Glavni koraci planiranja za nepredviđene situacije



Glavni koraci (2)

- ◆ Formiranje tima za krizno planiranje (CPMT)
 - Predstavnici upravljačke razine, poslovnih procesa te podređenih timova
- ◆ Razvoj izjave o politici CP
 - formalizirana politika – vodič za planiranje i ponašanje u slučaju nepredviđenih situacija
- ◆ Provedba analize utjecaja na poslovanje
 - Prepoznavanje poslovnih funkcija i IS kritičnih za poslovanje te određivanje njihovih prioriteta
- ◆ Formiranje podređenih timova
 - za planiranje koji će razviti IR, DR i BC planove, ne nužno istih za provođenje
- ◆ Razvoj podređenih politika
 - Timovi za područje IR, DR i BC
- ◆ Integracija analize utjecaja na poslovanje (BIA)
 - Svaki od podređenih timova treba procijeniti aspekte BIA važne za njihovo područje

Glavni koraci (3)

- ◆ Utvrđivanje preventivnih kontrola
 - Procjena protumjera i zaštitnih mjera za smanjenje rizika i posljedica štetnih događaja na podatke, poslovne procese i osoblje
- ◆ Organiziranje timova za odgovor
 - Navod vještina potrebnih za odgovor IR, DR i BC te odabir potrebnog osoblja
- ◆ Razvoj strategija odgovora (contingency strategies)
 - Pr. planovi izrade pričuvnih kopija i oporavka podataka, organizaciju alternativnih lokacija, ...
- ◆ Razvoj podređenih planova
 - Aktivnosti za svako od područja (IR, DR, BC)
- ◆ Testiranje plana, trening i vježbe
 - Provjera učinkovitosti svakog od podređenih planova
- ◆ Održavanje plana
 - Periodička provjera, procjena plana te ažuriranje

Analiza utjecaja na poslovanje

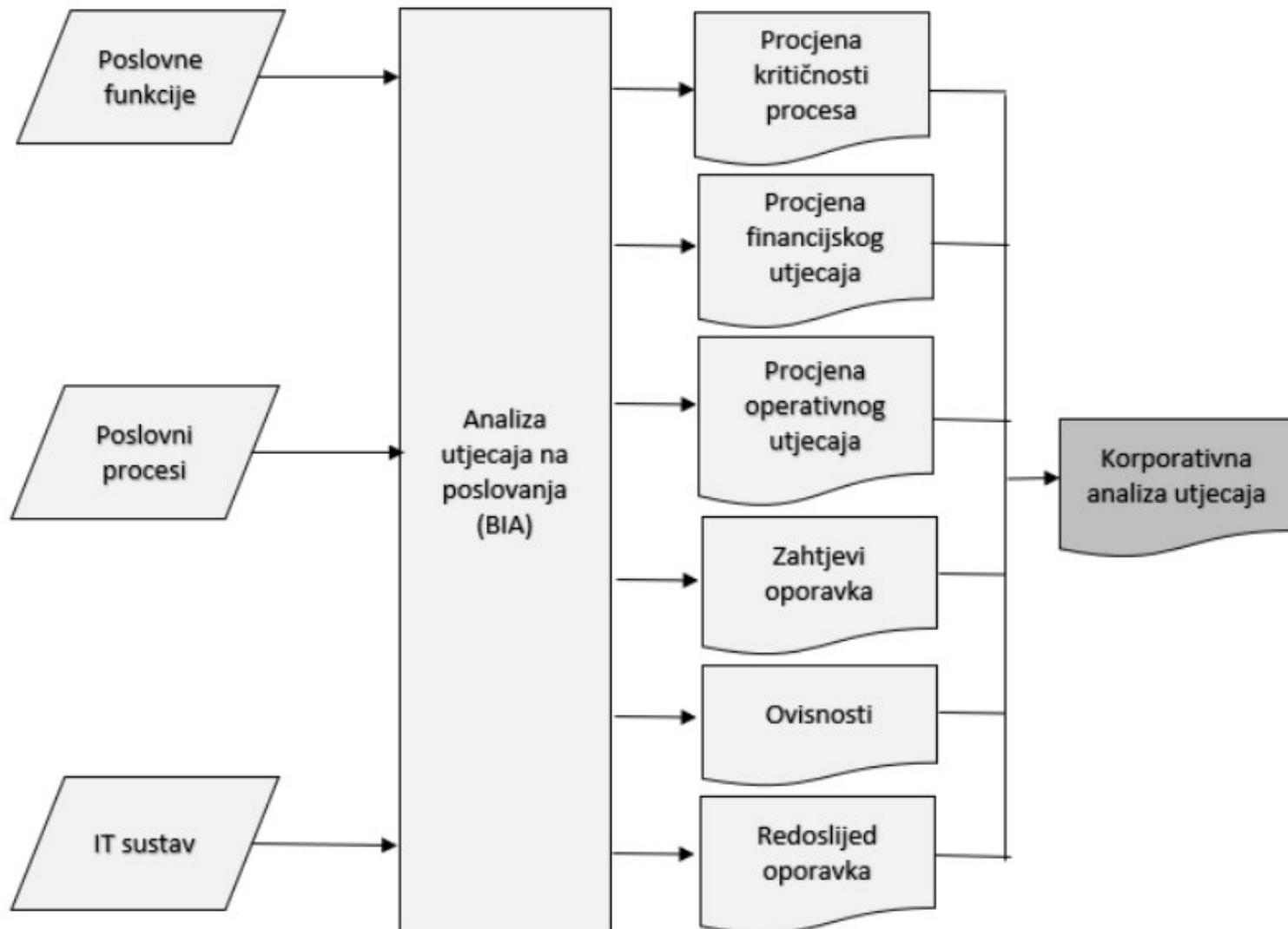
Analiza utjecaja na poslovanje

- ◆ Analiza utjecaja na poslovanje (Business Impact Analysis - BIA)
 - Ustanovljava organizacijske funkcije i njihove prioritete, kao i informacijske sustave koji podržavaju kritične poslovne procese
 - Upravljanje rizikom usmjerava se na prijetnje, ranjivosti i napade radi određivanja kontrola za zaštitu informacija
 - **BIA prepostavlja da kontrole mogu biti zaobiđene, neučinkovite**
- ◆ Nastoji odgovoriti kako će to utjecati
 - **Doseg**: koje organizacijske cjeline i sustave obuhvatiti
 - **Plan**: podaci mogu biti obimni – uvažiti relevantne
 - **Ravnoteža**: objektivno-subjektivno, naglasak na znanju i iskustvu osoblja
 - **Cilj**: odrediti ključne donositelje odluka – informacije za donošenje
 - **Praćenje**: povremena provjera da vlasnici procesa i donositelji odluka podržavaju proces i rezultat BIA

Koraci BIA

- ◆ NIST SP 800-34 (National Institute of Standards and Technology)
 - Identifikacija ključnih poslovnih procesa i funkcija,
 - Utvrđivanje međuvisnosti informacijskih sustava i poslovnih procesa,
 - Utvrđivanje prioriteta i klasifikacija poslovnih procesa i funkcija,
 - Utvrđivanje utjecaja prekida poslovnih procesa na sveukupne poslovne operacije, s naglaskom na financijske i operativne utjecaje,
 - Utvrđivanje zahtijevanih vremena oporavka,
 - Utvrđivanje preduvjeta za oporavak poslovanja,
 - Utvrđivanje redoslijeda oporavka pojedinih procesa i funkcija.

Rezultat BIA: korporativna analiza utjecaja na poslovanje



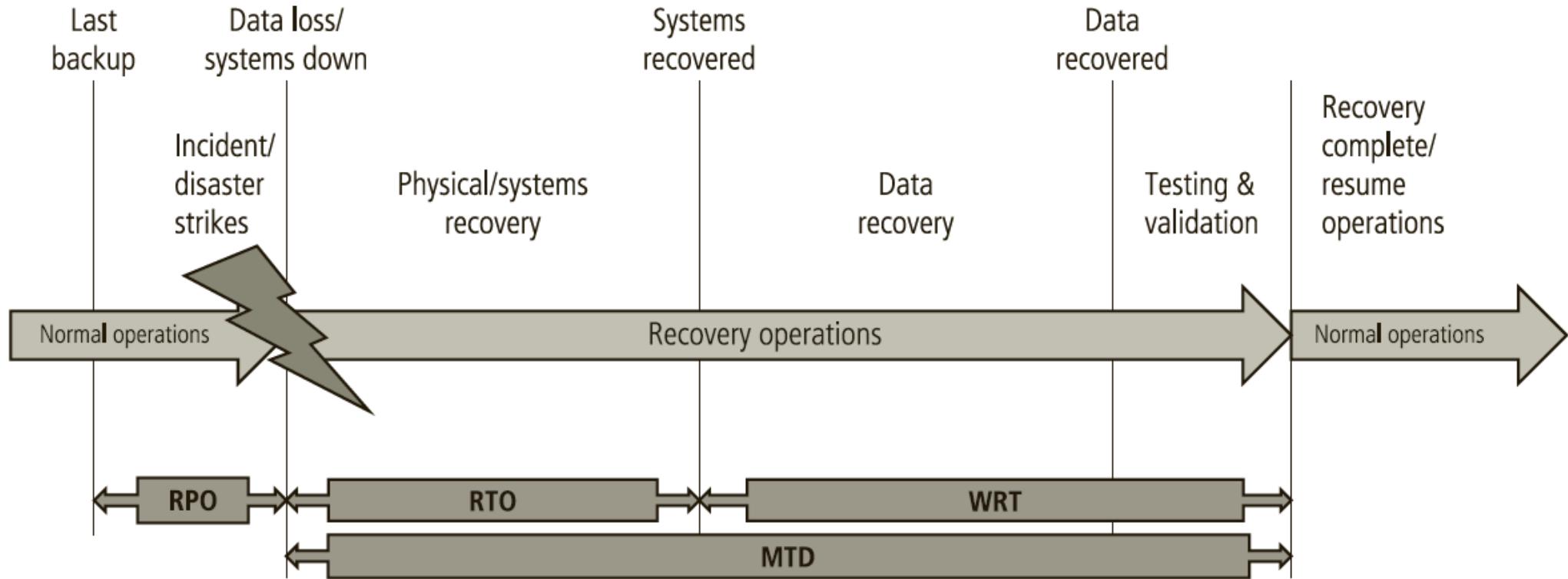
Identifikacija poslovnih procesa i funkcija te procjena utjecaja

- ◆ **Kritične funkcije** (critical functions) - neophodne za poslovanje org. (core)
 - IT gledište - prekid ima ozbiljne/trajne sigurnosne, operativne i financijske učinke
 - Prihvatljivo vrijeme oporavka mjeri se satima
- ◆ **Bitne funkcije** (essential functions) - vrlo važne, ali ne ključne
 - Pr. isplata plaće zaposlenicima
 - Prihvatljivo vrijeme oporavka u IT segmentu – dan ili dva
- ◆ **Potrebne funkcije** (necessary functions)
 - Nedostupnost u duljem razdoblju može imati značajan učinak
 - Pr. E-pošta ili pristup Internetu, funkcije potpore poslovnim procesima
 - Prihvatljivo vrijeme oporavka mjeri se danima
- ◆ **Poželjne funkcije** (desirable functions) - mali učinak na poslovanje
 - Pomoćne funkcije koje su se razvile vremenom kao potpora poslovanju
 - Prekid može biti prilika za njihovu reviziju – može se ispostaviti da nisu potrebne
 - Prihvatljivo vrijeme oporavka – tjednima ili mjesecima

Zahtjevi oporavka

- ◆ **Ciljana točka oporavka - RPO** (Recovery Point Objective)
 - Vremenska tolerancija gubitka podataka, stanje povrata oporavkom pričuvne kopije podataka
 - Vrijeme između posljednjeg backupa i prekidnog događaja
 - Pr. tjedni backup + ispad u subotu → RPO = 1 tjedan
- ◆ **Ciljano vrijeme oporavka - RTO** (Recovery Time Objective)
 - Maksimalno vrijeme za oporavak resursa koji podržavaju misiju organizacije
 - Računalni sustavi, proizvodni uređaji, telekomunikacije, zgrade i radni prostor
 - Vrijeme između prekidnog događaja i oporavka sustava/resursa
- ◆ **Vrijeme oporavka rada - WRT** (Work Recovery Time)
 - Vrijeme potpunog oporavka poslovne funkcije nakon oporavka resursa
 - Obnova podataka (elektronički restore i ručni unos) + testiranje i validacija
- ◆ **Maksimalno prihvatljivo vrijeme ispada - MTD** (Maximum Tolerable Downtime)
 - Maksimalno podnošljiv zastoj/ispad sustava mjerен trajanjem neraspoloživosti poslovnih procesa
 - Period između prekidnog događaja i početka normalnog poslovanja
 - $MTD = RTO + WRT$

Analiza i postavljanje prioriteta poslovnih procesa



Međuvisnosti poslovnih funkcija

- Kako će prekid određene poslovne funkcije utjecati na ostale i kada će to biti?
- Je li ta funkcija vezana za neke specifične resurse (određeni dobavljači, oprema)?
- Koje su ključne osobe za obavljanje te funkcije? Što kada su te osobe nedostupne?
- Kako se ta funkcija obavlja – kontinuirano, povremeno, na dnevnoj ili tjednoj bazi? Postoji li neko kritično vrijeme kada je neophodna za poslovanje?
- Koji su IT resursi neophodni za obavljanje te funkcije?
- Postoje li neke ručne, zaobilazne procedure kojima se ona može izvršavati i ako informacijski sustav nije dostupan?

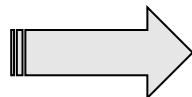
Izvješće o analizi utjecaja

- Ključni procesi i funkcije,
- Međuvisnosti procesa i IT resursa,
- Kritičnost odnosno razina utjecaja na poslovanje,
- Ključne uloge i odgovornosti osoba zaduženih za njihovu provedbu,
- Zahtijevana vremena oporavka,
- Financijski, operativni, pravni, personalni učinci nedostupnosti,
- Ručne procedure za nastavak poslovanja u slučaju nedostupnosti.

Odgovor na incident

Planiranje odgovora na incidente (IRP)

- Identifikacija i klasifikacija incidenata te odgovarajućih odgovora
- ◆ Tim za planiranje odgovora na incident (IR team)
 - Razvija planove za odgovor na incident
- ◆ Tim za odgovor na incident
 - Computer Security Incident Response Team (CSIRT)
 - Izvodi planove, kao reakciju na incident
- ◆ Faze odgovora na incident
 - Planiranje (planning)
 - Detekcija (detection)
 - Reakcija (reaction)
 - Oporavak (recovery)



Uspostava tima za odgovor na incidente

- Srodni pojmovi
- ◆ Computer Security Incident Response Team (**CSIRT**)
 - usluga odgovorna za zaprimanje, pregled i odgovor na prijavu incidenata računalne sigurnosti – organizacijsko tijelo, ali može biti i vanjsko
- ◆ Tim za odgovor na incidente informacijske sigurnosti
 - Information Security Incident Response Team (**ISIRT**)
 - prema normi ISO/IEC 27035:2011 (više ne vrijedi)
 - tim odgovarajuće vještih i pouzdanih članova organizacije koji tijekom svog životnog ciklusa rješavaju incidente informacijske sigurnosti
- ◆ Computer Emergency Response (**CERT**)
 - tim za IKT incidente, organizacijski, češće nacionalni, gdje može biti i drugčije nazvan
 - Pr. <https://www.cert.hr/> , https://www.cert.hr/csirt_specifikacija/

Politika odgovora na incidente

- ◆ NIST 800-61, Rev. 2, The Computer Security Incident Handling Guide
 - Izjava o svrsi i ciljevima politike
 - Doseg – na koga se što odnosi te u kojim okolnostima
 - Definicija incidenata i povezanih pojmove
 - Organizacijska struktura, definicija uloga, odgovornosti i ovlasti
 - Zapljena ili isključivanje opreme, nadzor sumnjivih aktivnosti, prijava počinitelja
 - Dijeljenje informacija (što, tko, kada, kako)
 - Postupak eskalacije
 - Postavljanje prioriteta ili ocjene ozbiljnosti incidenata
 - Mjerenje učinaka (kontrola pristupa, sigurnosne stijene, DNS, ...)
 - Izvještavanje i formulari

Planiranje odgovora na incident

- ◆ Prepostavka je da postoji CSIRT
 - Kompetencije, dežurstva, ...
- ◆ Format i sadržaj
 - Organizirane upute o procedurama postupanja
 - ... za vrijeme i nakon incidenta
- ◆ Smještaj – zaštita IR plana
 - Pri ruci, ali tako da ih napadač ne otkrije
 - Fizički registratori blizu administratorskih stanica, ormari, šifrirane datoteke
- ◆ Testiranje
 - Kontrolne liste, strukturirani prohod (walk-through), simulacija, potpuni prekid
- ◆ The more you sweat in training, the less you bleed in combat.
- ◆ Training and preparation hurt.
- ◆ Lead from the front, not the rear.
- ◆ You don't have to like it, just do it.
- ◆ Keep it simple.
- ◆ Never assume.
- ◆ You are paid for your results, not your methods.

Detekcija incidenta

- ◆ Indikatori mogućih incidenata

- Nepoznate datoteke
- Nepoznati procesi
- Neuobičajeno trošenje računalnih resursa
- Neuobičajen pad sustava

- ◆ Indikatori vjerojatnih incidenata

- Aktivnosti u neuobičajena vremena (mrežni promet ili pristup datotekama „kada ih nitko ne koristi“)
- Pojava novih vjerodajnica
- Napadi prijavljeni od strane korisnika
- Notifikacije IDPS (Intrusion Detection / Prevention System)

Detekcija incidenta (2)

- ◆ Indikatori **izvjesnih** incidenata
 - Korištenje neaktivnih vjerodajnica
 - Izmjene dnevničkih zapisa (u odnosu na rezervnu kopiju)
 - Prisustvo hakerskih alata
 - Dojava partnera ili parnjaka (*partner, peer*)
 - Poruka hakera – „gotcha“ na web stranici ili email poruka sa „sigurnog“ računa
- ◆ Drugi indikatori
 - Gubitak raspoloživosti - nedostupan sustav
 - Gubitak integriteta - korumpirane datoteke ili podaci
 - Gubitak povjerljivosti - obavijest o curenju podataka ili otkrivanje podataka za koje se mislilo da su zaštićeni
 - Kršenje politike – događaji u suprotnosti s org. politikama sigurnosti
 - Kršenje zakona – prekršen je zakon u čemu su sudjelovala org. sredstva

Reakcija – ključni pojmovi

- ◆ Poruka upozorenja (alert message)
 - Opis incidenta s dovoljno informacija
 - Da svaka osoba zna koji dio IR plana provesti bez da uspori obavješćivanje
- ◆ Popis upozorenja (alert roster)
 - Kontakti koje treba obavijestiti o događaju incidenta
 - Hijerarhijski popis (hierarchical roster)
 - Popis upozorenja u kojem prva osoba poziva nekoliko drugih, a one dalje
 - brže ali nepreciznije
 - Slijedni popis (sequential roster)
 - Popis upozorenja u kojem jedna osoba poziva svaku na popisu
 - točnije ali dugotrajnije

Reakcija - postupak

- ◆ Pomoćna služba (help desk), korisnik ili administrator sustava
 - Pozivaju „prave ljude” s popisa upozorenja
- ◆ Dokumentiranje incidenta
 - Tko, što, kada, gdje, zašto i kako
 - Studijski slučaj, učenje
 - Dokaz za ispravno postupanje
 - Podloga za simulacije u budućnosti
- ◆ Strategije suzbijanja incidenata i povrata kontrole
 - Filtriranje poruka, blokiranje priključnica, onesposobljavanje vjerodajnica, rekonfiguriranje sig. stijene, privremeno zaustavljanje servisa i procesa

Oporavak od incidenta

- ◆ Ulaganje napora po prioritetima – slijedenjem plana
- ◆ Procjena štete
 - Trenutno, danima, tjednima
 - Procjena sustava i pohrane podataka
 - Proučavanje dnevnika (log), računalna forenzika, prikupljanje dokaza
- ◆ Oporavak
 - Identifikacija ranjivosti
 - Instalacija, zamjena, nadogradnja zaštite
 - Oporavak podataka, usluga, procesa
 - Kontinuirano praćenje/nadzor sustava
 - Obnavljanje povjerenja
- ◆ Naknadna revizija (After Action Review - AAR)

Oporavak od katastrofe

Katastrofa

- neželjeni i neočekivani štetni događaj koji organizaciji
 - onemogućuje obavljanje kritičnih poslovnih funkcija
 - kroz neodređeni vremenski period i
 - rezultira velikom štetom (ne samo finansijskom) za njezino posovanje
-
- ◆ Neki primjeri
 - nedostupnost glavne lokacije organizacije zbog prirodne katastrofe ili požara,
 - nedostupnost IT infrastrukture na glavnoj lokaciji zbog kvara hardvera ili softvera većih razmjera,
 - nedostupnost ključnih djelatnika organizacije zbog epidemije,
 - dugotrajni prekid isporuke električne energije,
 - prekid ključnih usluga dobavljača

Sadržaj plana oporavka od katastrofe (DR plan)

- ◆ Popis IT sredstava
 - inventura hardvera, sustava i aplikacija
- ◆ Procjena rizika
 - za svaki ključni IS; vjerojatnost, posljedice
- ◆ Klasifikacija važnosti
 - kritični, ostali
- ◆ RPO i RTO
- ◆ Popis aktivnosti – procedure uspostave nastavka poslovanja
 - Kratkoročne – osnovne funkcionalnosti
 - Dugoročne – poslovanje se vraća u uobičajeno stanje

Aktivnosti oporavka

- ◆ Oporavak hardvera
 - Zamjena komponenti na glavnoj ili pričuvnoj lokaciji
 - Poslužitelji, mrežna oprema, vatrozid, IP/DS
- ◆ Oporavak operacijskih sustava
 - OS i glavni servisi (npr. DNS, AD)
- ◆ Oporavak baza podataka i arhivskih zapisa
- ◆ Oporavak spremišta podataka
 - Storage, pričuvni hardver (Storage Area Network – SAN)
- ◆ Oporavak aplikacija
 - Podaci, sinkronizacija s pričuvnom lokacijom, provjera
- ◆ Testiranje procedura oporavka

Razine oporavka od katastrofe ([IBM, 2007](#))

- ◆ Razina 0 – bez pohrane podataka na pričuvnoj lokaciji
 - Podatci se ne pohranjuju na drugoj lokaciji
 - Oporavak je moguć samo korištenjem sustava na primarnoj lokaciji
- ◆ Razina 1 – Izrada pričuvne kopije podataka s hladnom lokacijom
 - Podatci se pohranjuju na diskove/trake i fizički šalju na pričuvnu lokaciju
 - Pickup Truck Access Method (PTAM)
 - Pričuvna hladna lokacija (cold site)
 - samo osnovna infrastruktura poput namještaja, napajanja, mrežnih ormara i utičnica
 - uspostava HW i SW, pa vraćanje pričuvnih kopija podataka
 - Jeftino rješenje, nastavak rada obično moguć tek nakon nekoliko dana

Razine oporavka od katastrofe (BC tier 2 - 4)

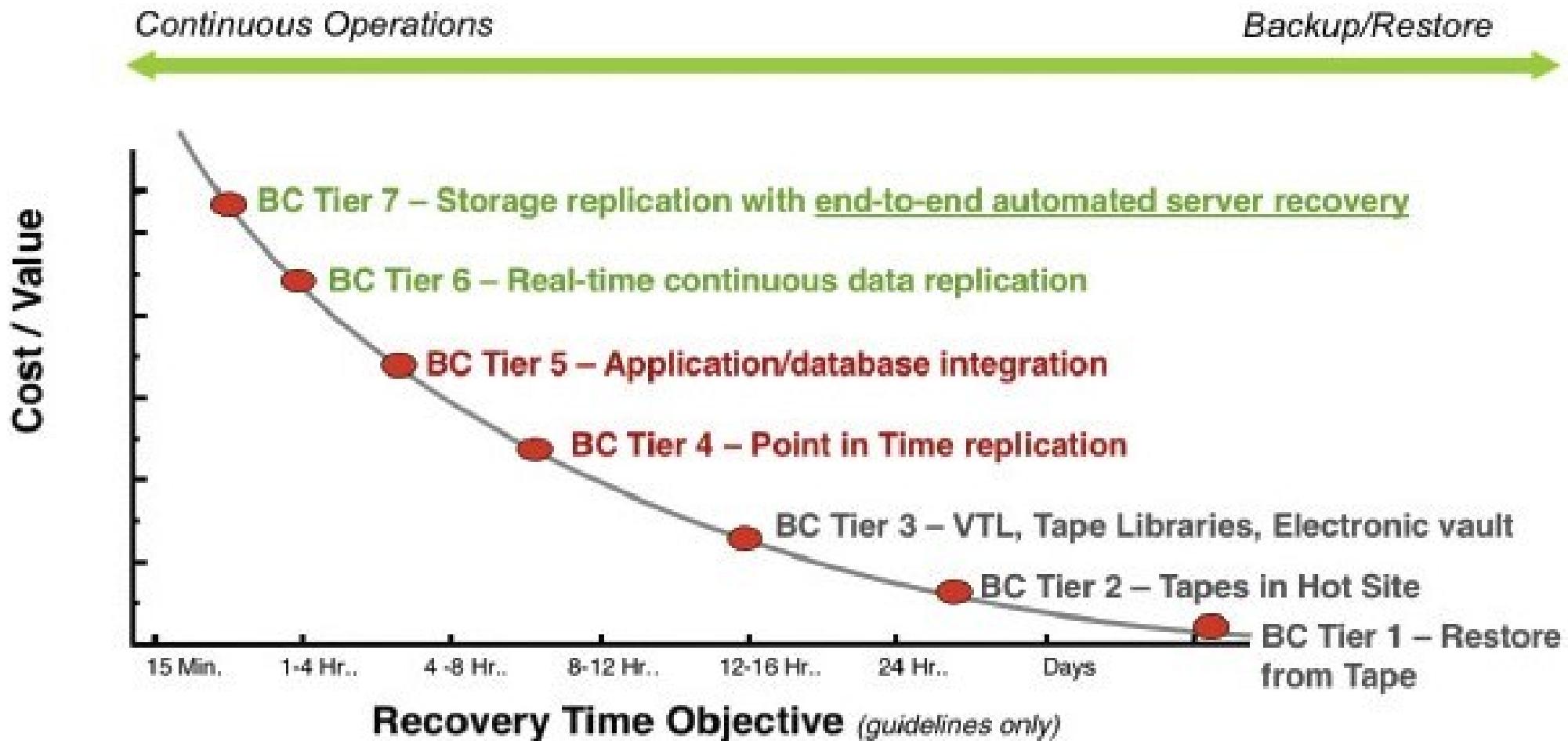
- ◆ Razina 2 – Izrada pričuvne kopije podataka s vrućom lokacijom
 - Pričuvne kopije se fizički šalju na pričuvnu lokaciju - PTAM
 - Pričuvna vruća lokacija (host site)
 - na kojoj je instaliran i aktivan pričuvni sustav s odgovarajućim HW i SW, pa vraćanje podataka
 - Skuplje rješenje, nastavak rada unutar 24 sata
- ◆ Razina 3 – Elektronička pohrana (Electronic vaulting)
 - BC2 + Kritični podaci elektronički na pričuvnu lokaciju (remote backup service)
 - Efikasnije, nastavak rada za desetak sati
- ◆ Razina 4 – Aktivna pričuvna lokacija
 - Svi podatci periodički elektronički kopirani na pričuvnu lokaciju (point-in-time copies)
 - *Batch/Online Database Shadowing and Journaling, Global Copy, FlashCopy, ...*
 - Gubitak podataka do nekoliko sati

Razine oporavka od katastrofe (BC tier 5 - 7)

- ◆ Razina 5 – Integritet transakcija
 - Aplikacijski podatci i podatci iz BP se na transakcijskoj razini preslikavaju na diskove na pričuvnoj lokaciji (two-phase commit, remote replication, ...)
 - Oporavak ovisan o korištenom softveru
- ◆ Razina 6 – Minimalni ili nikakav gubitak podataka
 - Svi podatci (neovisno o aplikaciji) se „trenutno“ kopiraju s primarne na pričuvnu
 - Elektronički (real-time storage mirroring, server mirroring), najčešće zrcaljenjem diska (disk-mirroring)
- ◆ Razina 7 – Potpuno automatizirano rješenje
 - Nadgradnja razine 6 pri kojoj u slučaju katastrofe IS automatski nastavlja raditi na hardverskoj infrastrukturi, aplikacijama i podatcima koji se nalaze na pričuvnoj lokaciji bez ikakvog prekida ili gubitka podataka

Razine oporavka i kontinuitet poslovanja

- ◆ BC1-3 backup/restore, BC4-5 brzi oporavak, BC6-7 kontinuirana dostupnost



Varijante pričuvne lokacije

- ◆ Cold – infrastruktura, Warm – bez aplikacija, Hot – potpuna konfiguracija



Hladna lokacija

- malo ili bez opreme
- nema mrežne veze
- nije spremna za automatsko preuzimanje
- nema sinkronizacije podataka
- velik rizik gubitka podataka
- jeftino



Topla lokacija

- djelomično dostupna oprema
- mrežna veza aktivna
- preuzimanje unutar nekoliko sati
- dnevna sinkronizacija
- mali gubitak podataka
- financijski isplativo



Vruća lokacija

- potpuno dostupna oprema
- mrežna veza aktivna
- preuzimanje unutar nekoliko minuta
- gotovo trenutna sinkronizacija
- bez gubitka podataka
- skupo

Procedure za prelazak s primarne na pričuvnu lokaciju i obrnuto

◆ ***Failover (activation)***

- Automatski nastavak rada na pričuvnom poslužitelju, računalnoj ili mrežnoj komponenti u slučaju kvara na primarnom P/RK/MK
- Pravi automatizirani *failover* moguć samo na razini BC7

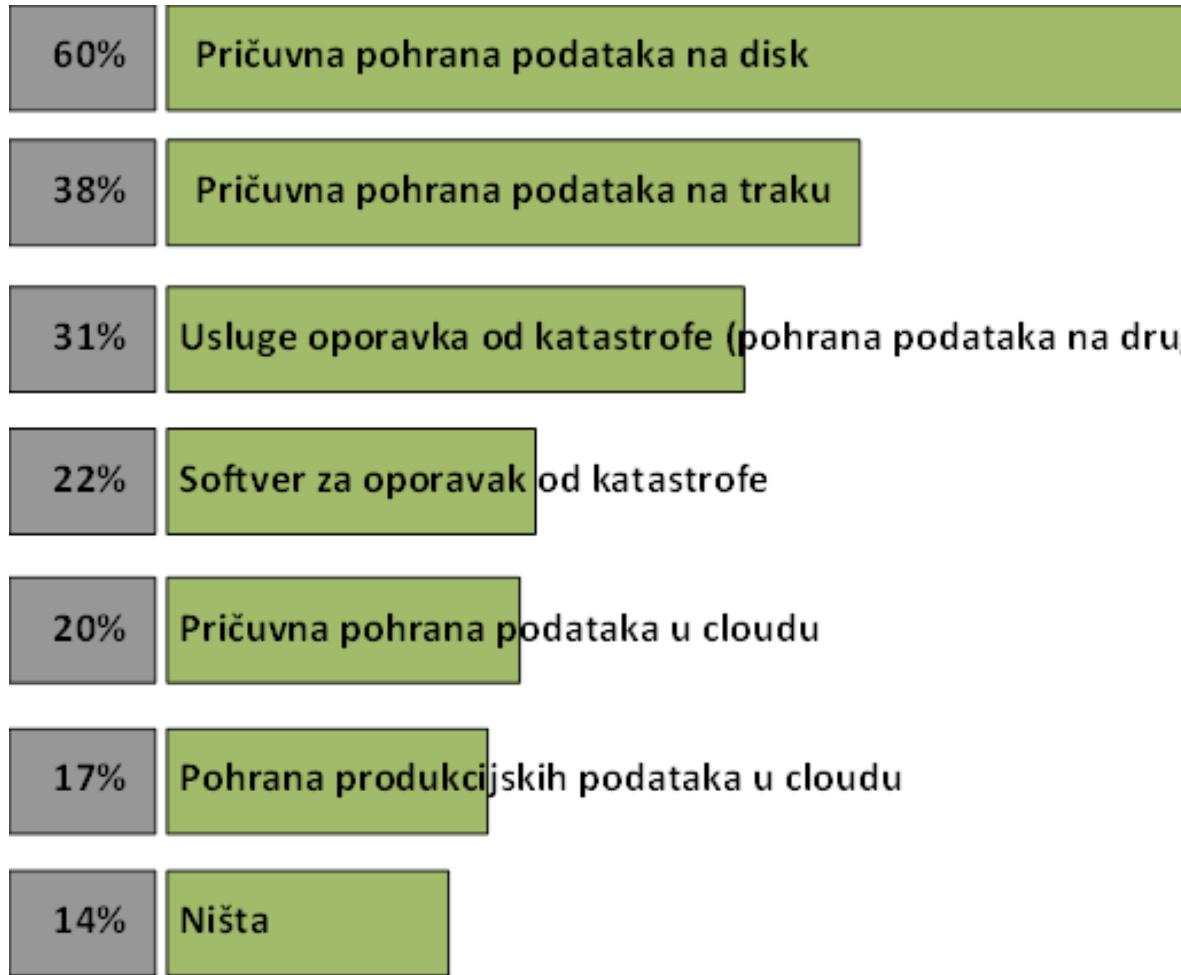
◆ ***Switchover (role switch)***

- Kontrolirana zamjena uloga, najčešće ručno u planirano vrijeme
- Priprema za održavanje – instalacija zakrpa, nadogradnji, ...
- Također za prelazak na pričuvnu kada je *failover* prekomplikiran ili preskup

◆ ***Fallback***

- Nakon osposobljavanja sustava na primarnoj lokaciji
- Vraćanje promjena u podacima i aplikacijama
- U idealnom slučaju (BC7) automatski
- U praksi uz manji ili veći gubitak podataka, ovisno o rješenju

Alati i tehnologije za oporavak od katastrofe



- ◆ DRaaS - DR as a Service
- ◆ BaaS - Backup as a Service
 - IBM BaaS
 - Veeam Cloud Connect Replication
 - MS Azure Site Recovery
- ◆ DR Tools
 - Zerto
 - Carbonite
 - Arcserve
 - Veritas
 - Datto
- ◆ Virtualizacija
 - VMware ESX, ESXi
 - MS Hyper-V

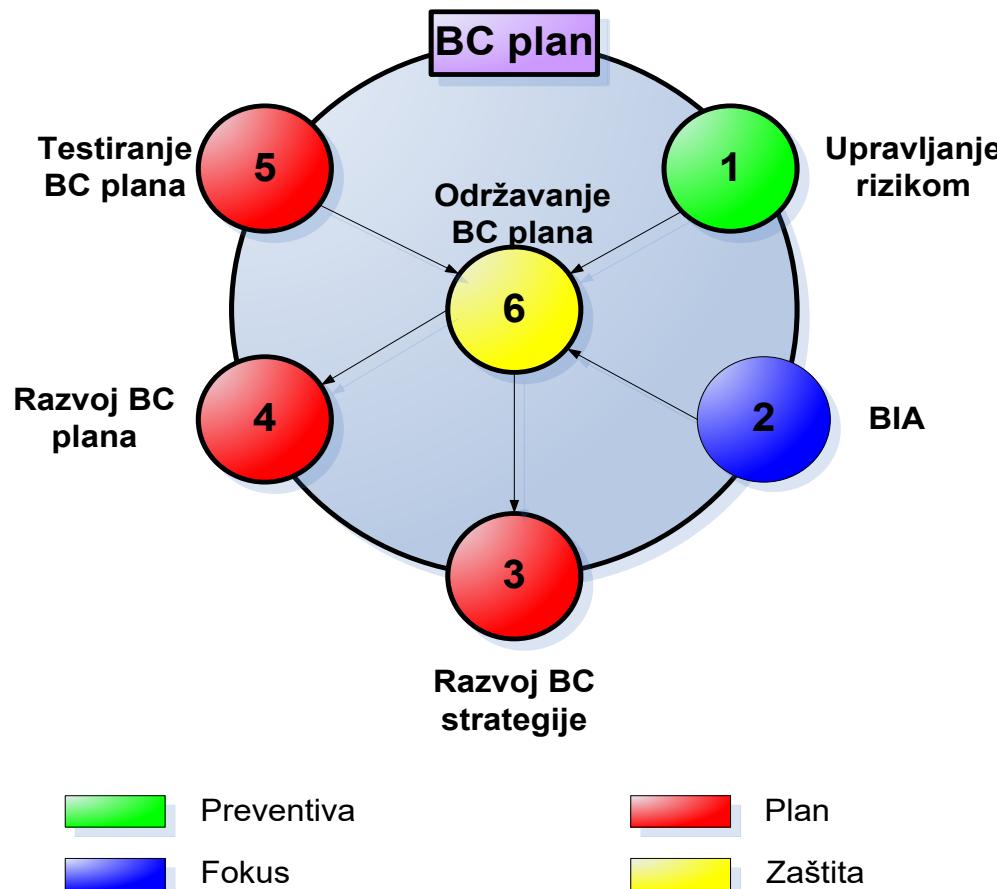
Kontinuitet poslovanja

Planiranje kontinuiteta poslovanja

- ◆ Napor organizacije da nastavi s kritičnim funkcijama u slučaju ispada primarne lokacije
 - Više rukovodstvo – razvoj i implementacija BC politike, plana te timova
- ◆ Uspostava sustava upravljanja kontinuitetom poslovanja (Business Continuity Management System - BCMS), prema normi:
 - ISO 22301 Security and resilience — Business continuity management systems — Requirements
 - ISO 22313 - Security and resilience — Business continuity management systems — Guidance on the use of ISO 22301

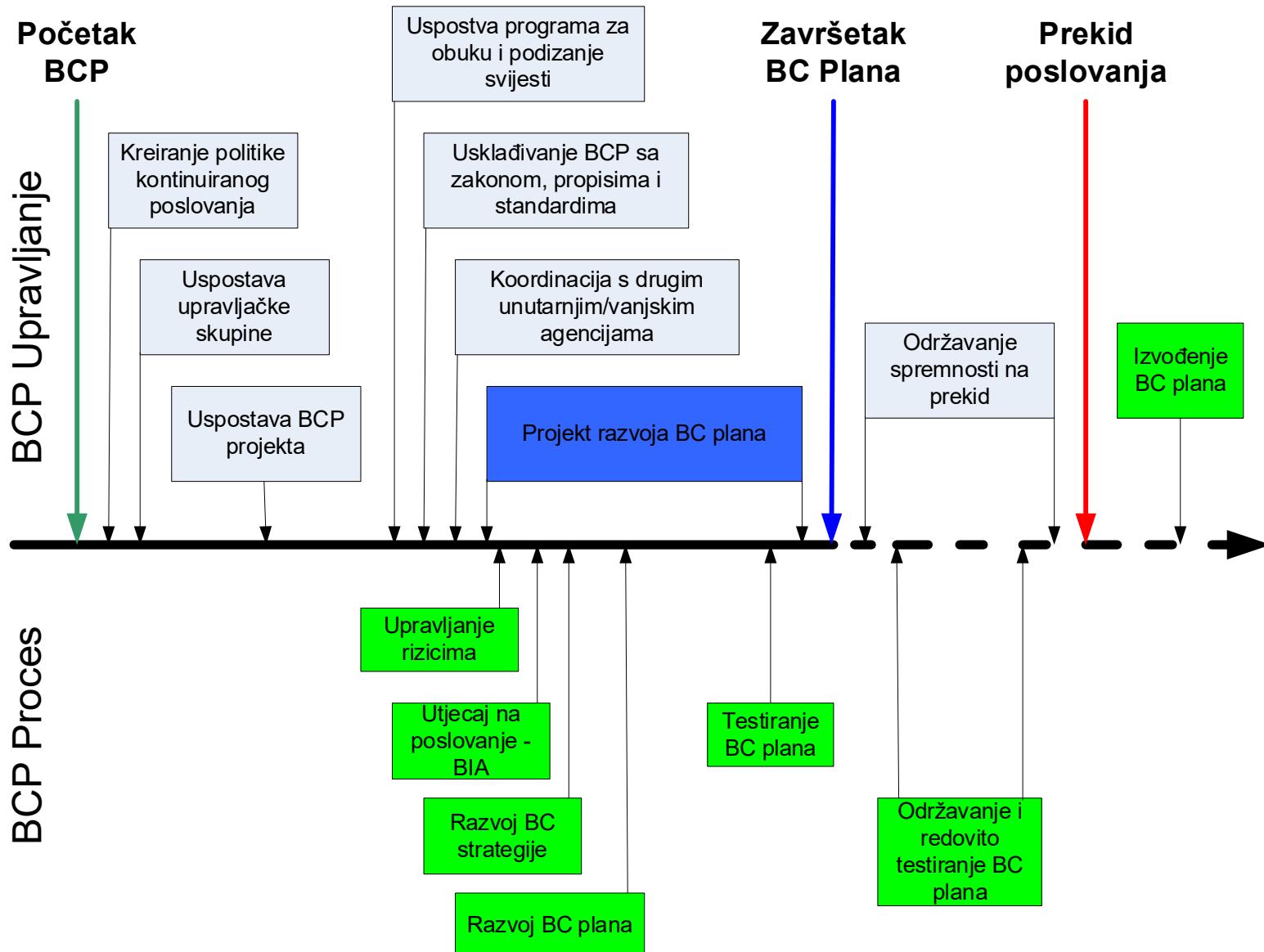
Proces planiranja kontinuiteta poslovanja

- ◆ Proces slijedi četiri ključna načela: *Fokus, Preventiva, Plan, Zaštita*
 - koji se implementiraju u BC programu kroz proces planiranja u šest koraka:



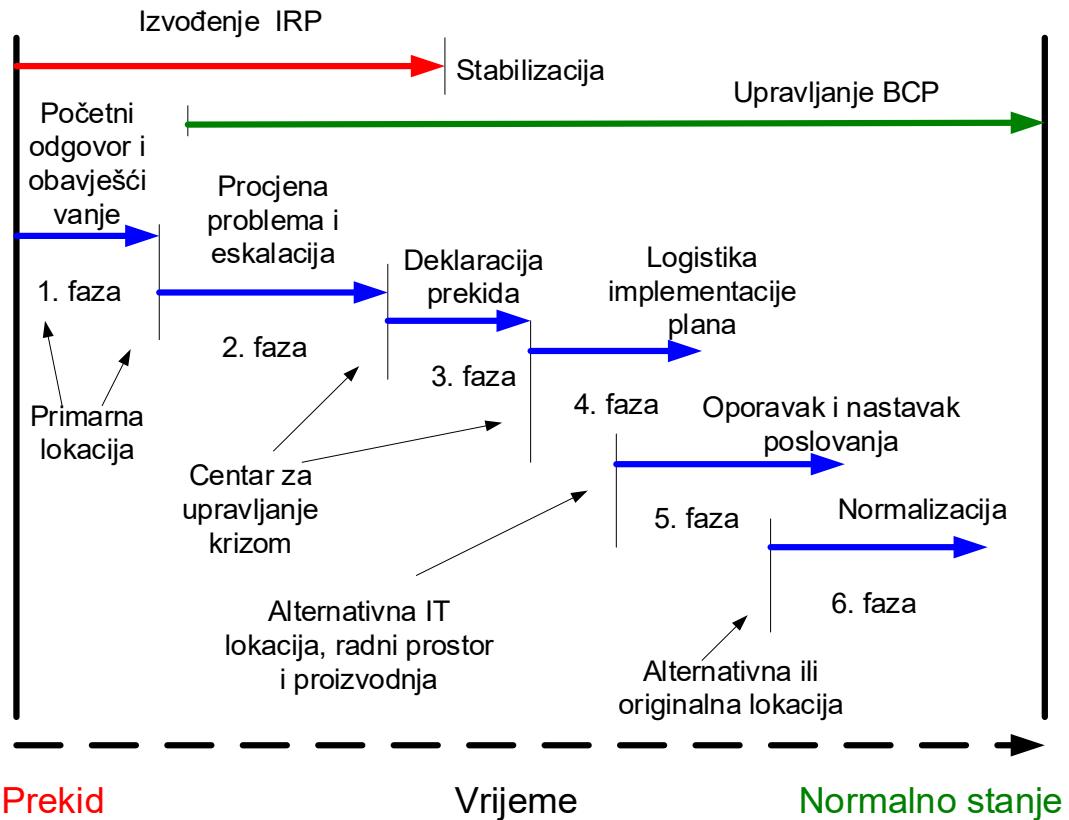
Planiranje kontinuiteta poslovanja

- ◆ Upravljanje rizikom
 - Procjena prijetnji i rizika za kontinuitet poslovanja, kontrola rizika
- ◆ Analiza posljedica na poslovanje (BIA)
 - Identifikacija ključnih poslovnih funkcija i procesa, analiza mogućih posljedica
 - Identifikacija zahtjeva za oporavak nakon pojave katastrofe
- ◆ Razvoj strategije kontinuiranog poslovanja
 - Ocjena zahtjeva za oporavak prekinutih ključnih poslovnih procesa.
 - Ustanovljavanje rješenja koja zadovoljavaju zahtjeve, odabir isplativih rješenja
- ◆ Razvoj BC plana
 - Zaštita ključnih procesa i sredstava od različitih prijetnji i rizika
 - Oporavak ključnih poslovnih procesa i resursa na siguran i vremenski prihvatljiv način
- ◆ Testiranje BC plana
 - Testiranje sposobnosti i učinkovitosti tima za oporavak
 - Testiranje sposobnosti i učinkovitosti dobavljača robe i usluga
- ◆ Održavanje BC plana



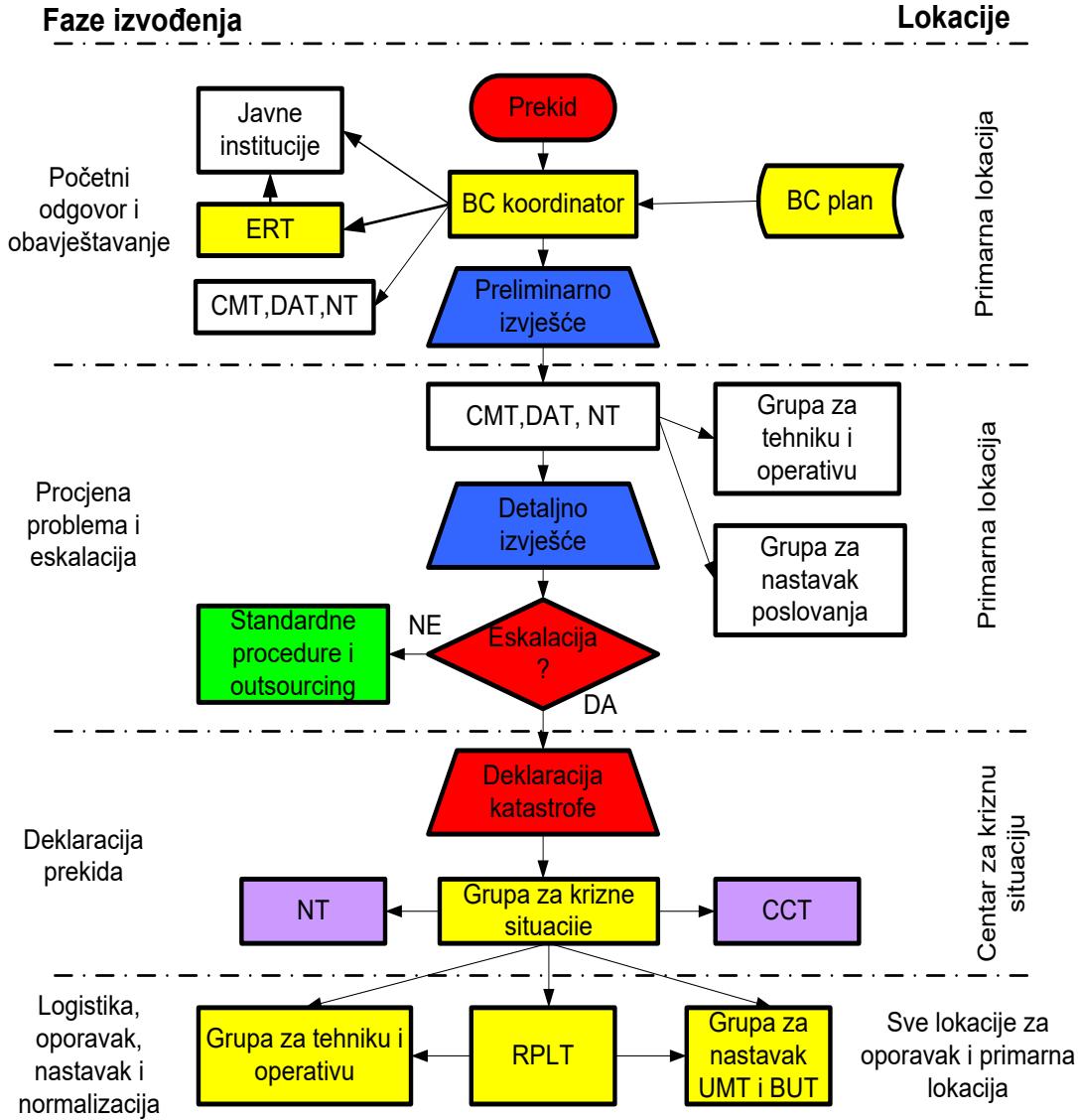
Izvođenje plana BC

- ◆ Početni odgovor i obavijest
 - preliminarno izvješće o problemu
- ◆ Procjena problema i eskalacija
 - detaljno izvješće o problemu
- ◆ Izjava o katastrofi / prekidnom događaju
 - proglašenje katastrofe / prekidnog događaja
- ◆ Implementacija plana logistike
 - mobilizacija timova, backup medija, kritičnih resursa i uređaja
- ◆ Oporavak i nastavak poslovanja
 - oporavak kritičnih IT i ne-IT resursa i nastavak procesa
- ◆ Normalizacija
 - operativni status kakav je bio prije pojave prekida



Uloge i odgovornosti pri izvođenju plana BC

- ◆ ERT – Emergency Response Team
- ◆ CMT – Crisis Management Team
- ◆ DAT – Data Team
- ◆ NT – Notification Team
- ◆ CCT – Command & Control Team
- ◆ RPLT – Resource Procurement and Logistics Team
- ◆ UMT – User Management Team
- ◆ BUT – Business Unit Team



Reference

- ISO/IEC 27031:2011 Guidelines for information and communication technology readiness for business continuity
 - Application of ISO/IEC 27002 to information and communication technology readiness for business continuity
- ISO/IEC 27035:2016+ — Information technology — Security techniques — Information security incident management
- NIST Special Publication (SP) 800-34, Revision 1, Contingency Planning Guide for Federal Information Systems
- NIST 800-61, Rev. 2, The Computer Security Incident Handling Guide
- ISO 22301 Security and resilience — Business continuity management systems — Requirements
- ISO 22313 - Security and resilience — Business continuity management systems — Guidance on the use of ISO 22301