



# Zaštita i sigurnost informacijskih sustava

## Projektiranje sigurnosti

prof. dr. sc. Krešimir Fertilj

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 daljnjeg 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

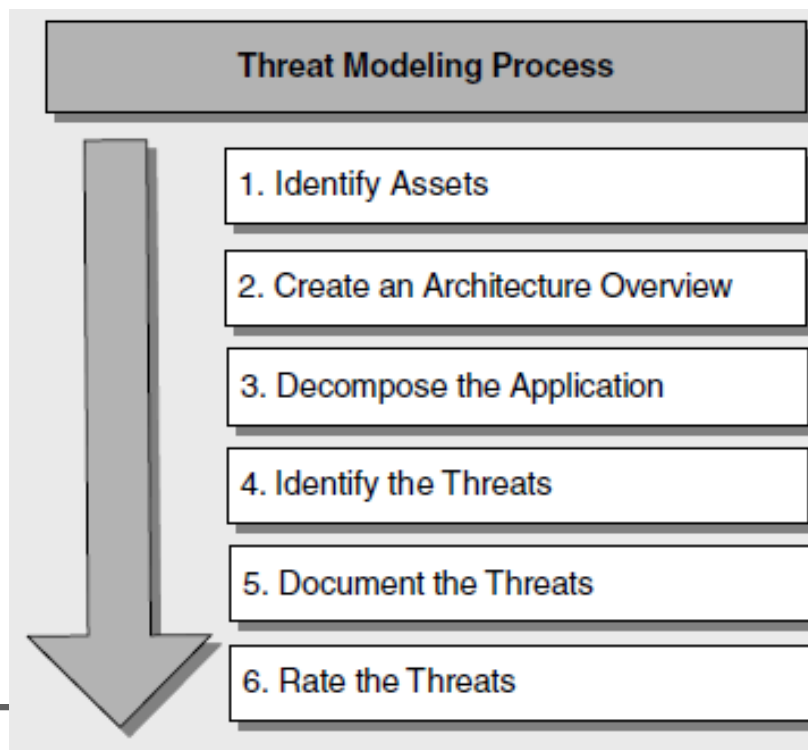
- bitno je da se obavi kvalitetno
- najbolje iterativno

važno:

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

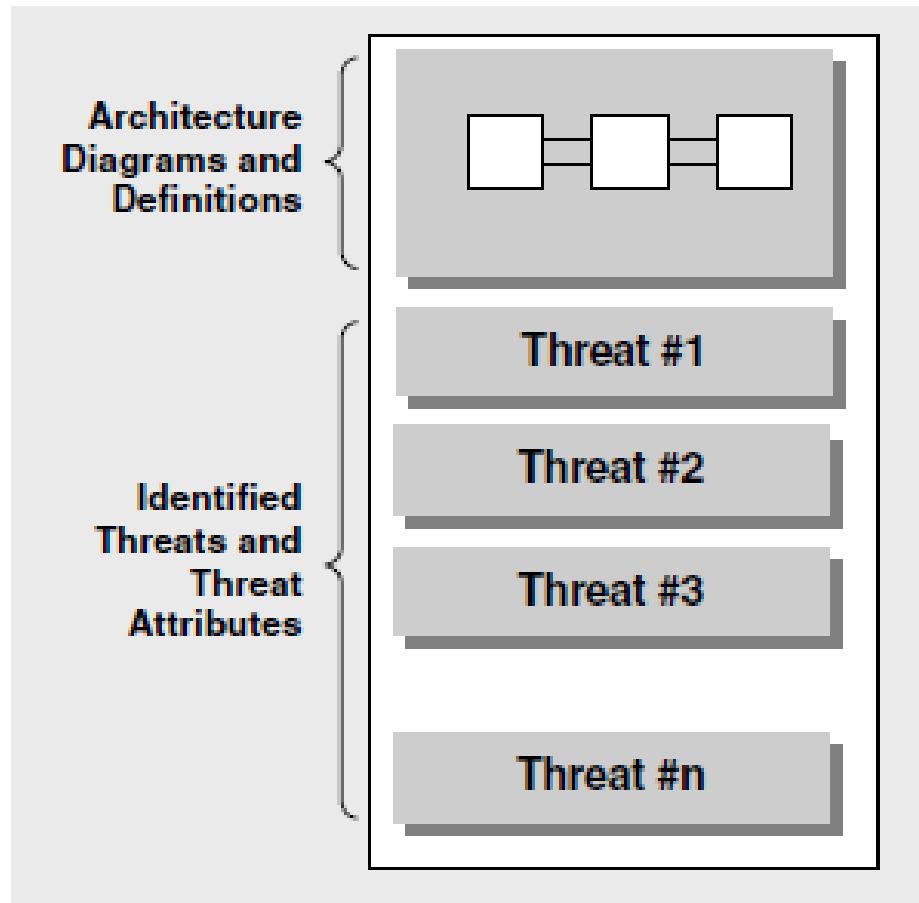
## ◆ Proces modeliranja prijetnji

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

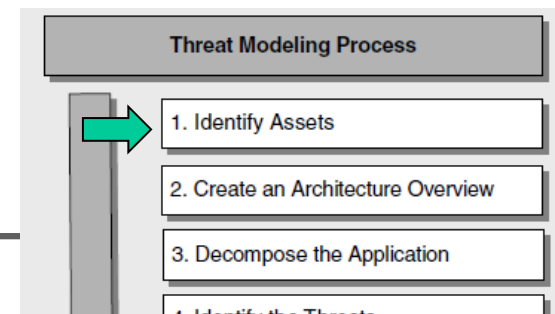


# Izlaz

- ◆ 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

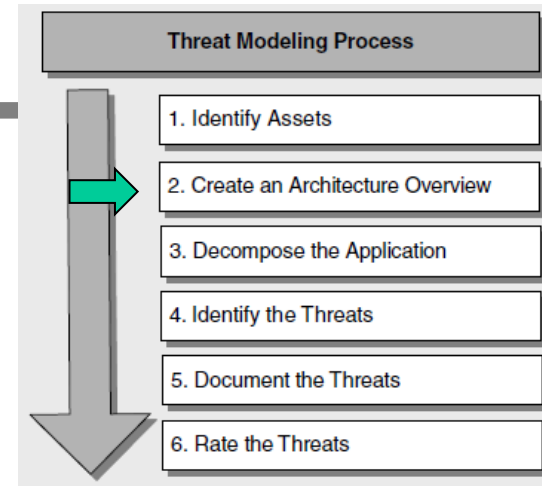
- funkcije aplikacije - što aplikacija radi
- arhitektura aplikacije i način fizičke ugradnje (konfiguracija)
- tehnologije implementacije

## ◆ 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.

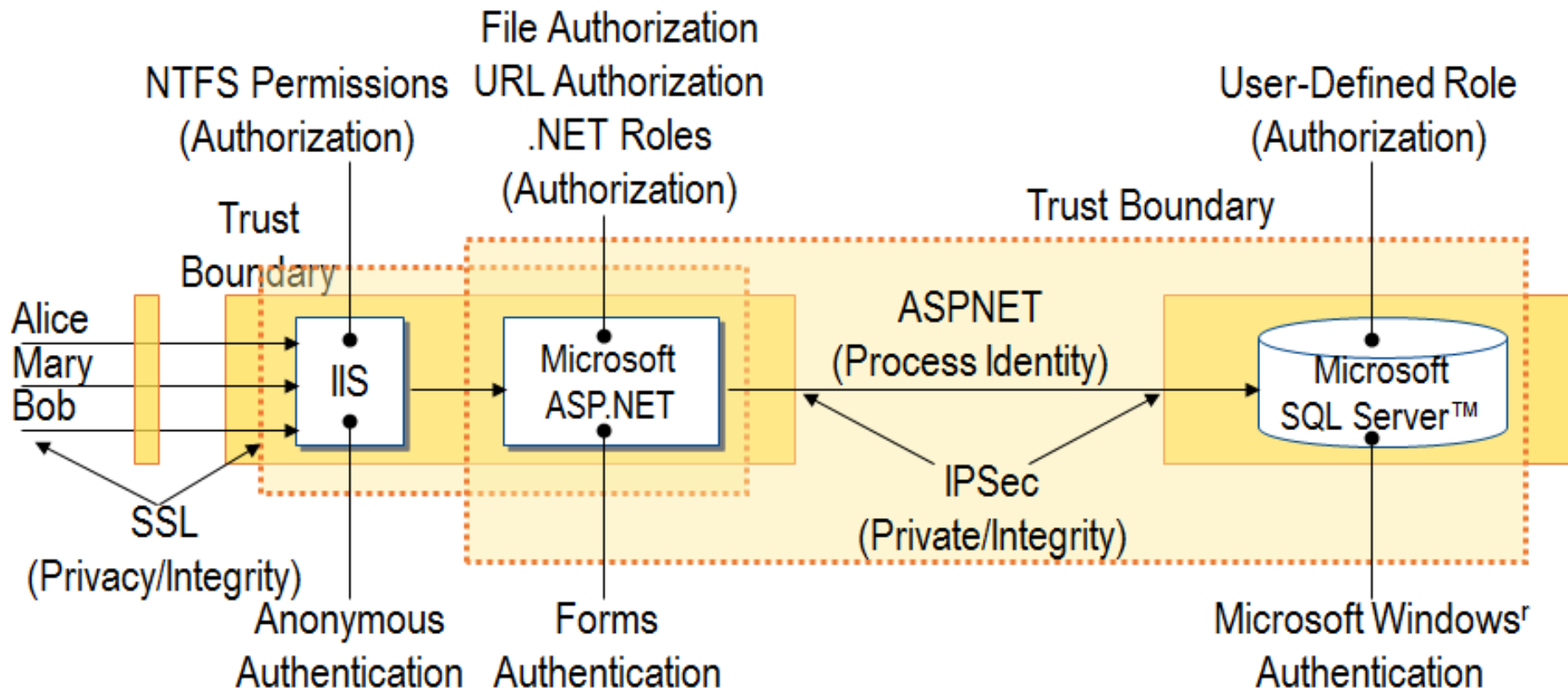
## ◆ Provjera (kršenja) poslovnih pravila

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



# Arhitektura i tehnologije implementacije

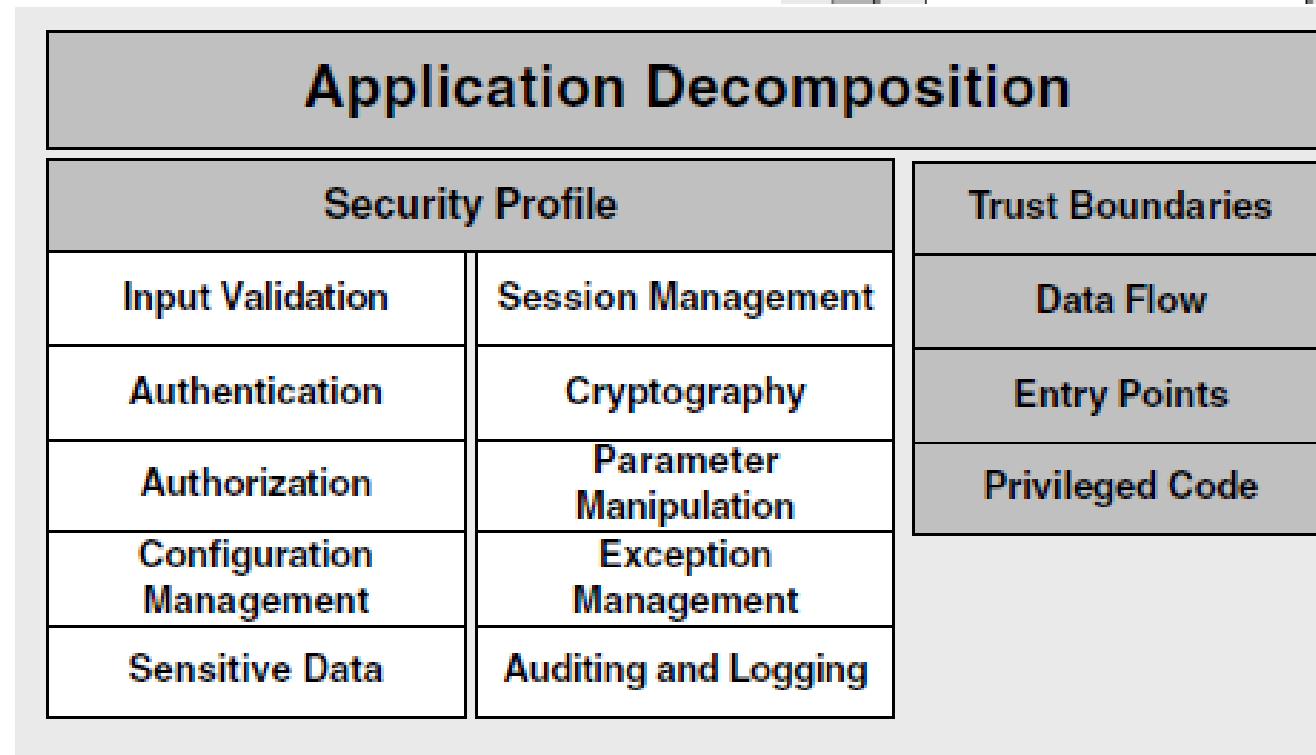
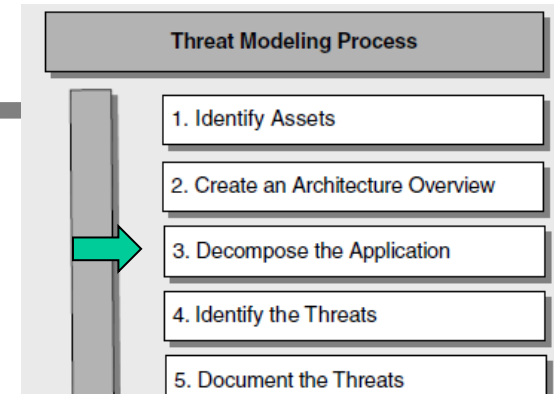
- ◆ Dijagram visoke razine - opisuje strukturu (komponente) sustava
  - ovisno o složenosti aplikacije treba izraditi detaljnije dijagrame dijelova
    - npr. dijagrame pojedinih slojeva višeslojne aplikacije
  - određivanje tehnologija implementacije na koje se nadodaju aspekti zaštite, pr.





# Korak 3 – Dekompozicija aplikacije

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



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

# Granice povjerenja i tokovi podataka

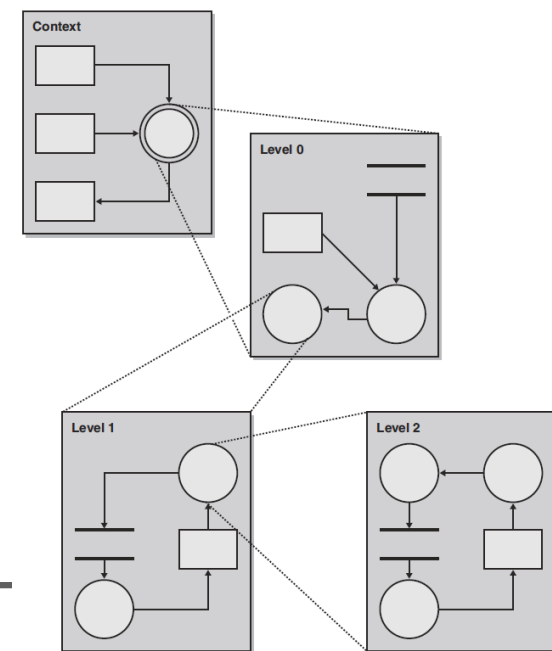
---

## ◆ 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
  - usmjereno 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**

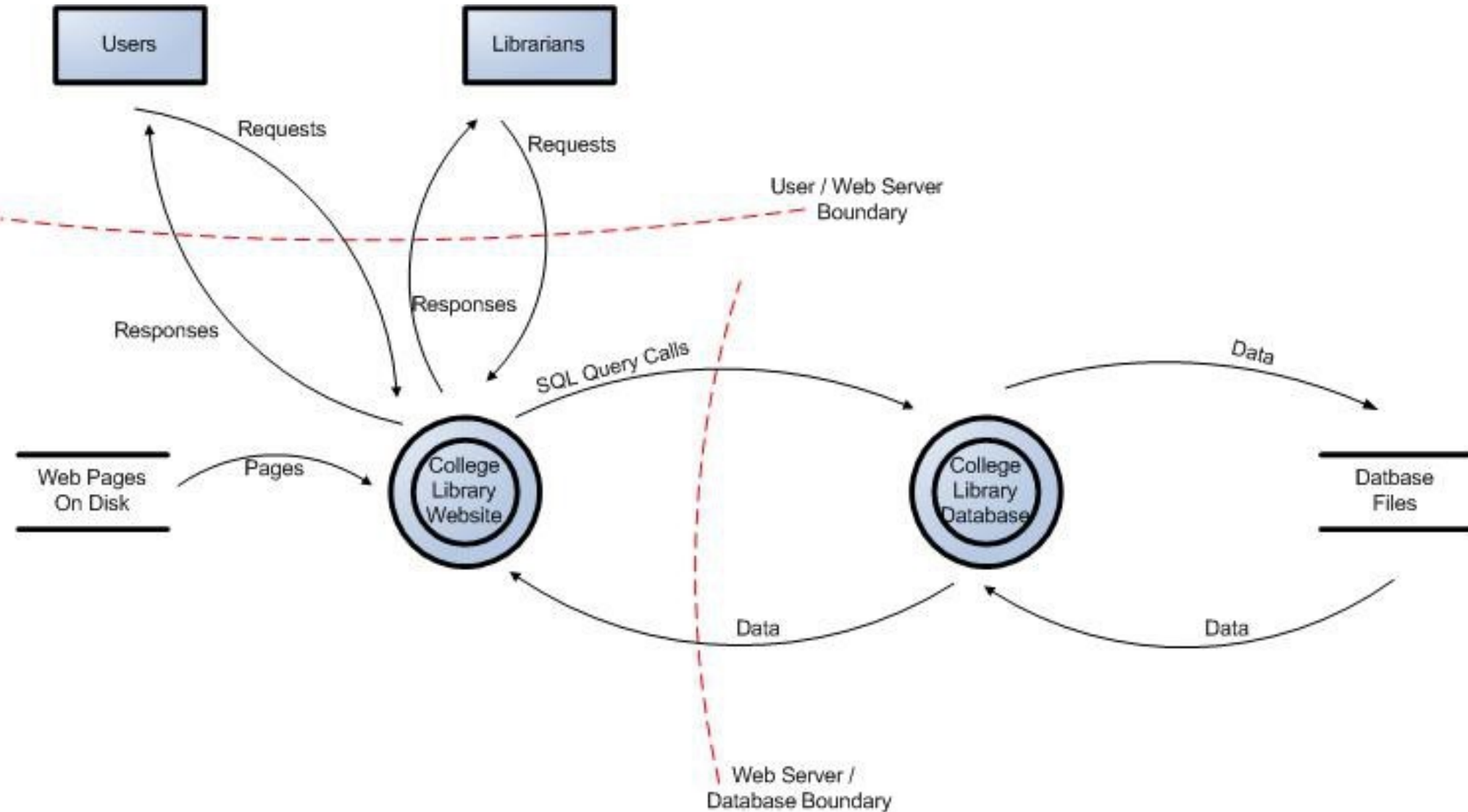
Input to the system.



## **Data Flow**

Depicts data flow from data stores, processes or interactors.

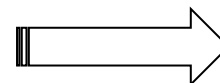
# Dijagram toka podataka - primjer



# 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

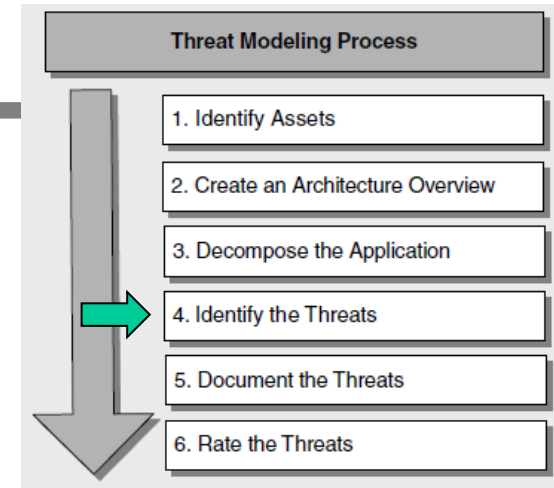


Category	Considerations
Input validation	<p>Is all input data validated?</p> <p>Could an attacker inject commands or malicious data into the application?</p> <p>Is data validated as it is passed between separate trust boundaries (by the recipient entry point)?</p> <p>Can data in the database be trusted?</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, tester 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

---

- ◆ **S**poofing – zavaravanje, lažiranje
  - preuzimanje tuđeg identiteta s ciljem pristupa resursima u mreži
  - npr. ilegalno dohvaćanje tuđih podataka prilikom autentifikacije
- ◆ **T**ampering [with Data] – zlonamjerna izmjena podataka
  - nedozvoljena izmjena npr. u bazi podataka ili prilikom prijenosa mrežom
- ◆ **R**epudiation – nepriznavanje, poricanje
  - mogućnost korisnika da porekne akciju, a da mu se to ne može dokazati
  - npr. „nisam obrisao”, „nisam naručio”, ...
- ◆ **I**nformation disclosure - otkrivanje informacija
  - neželjeno izlaganje privatnih podataka
  - npr. korisnik vidi sadržaj tuđe datoteke na što nema pravo
- ◆ **D**enial of service - uskraćivanje usluge
  - onemogućuje normalan rad sustava, relativno jednostavno i anonimno
  - npr. *flooding*, *amplification*, *protocol vulnerability*, *malformed packets*
- ◆ **E**levation 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			

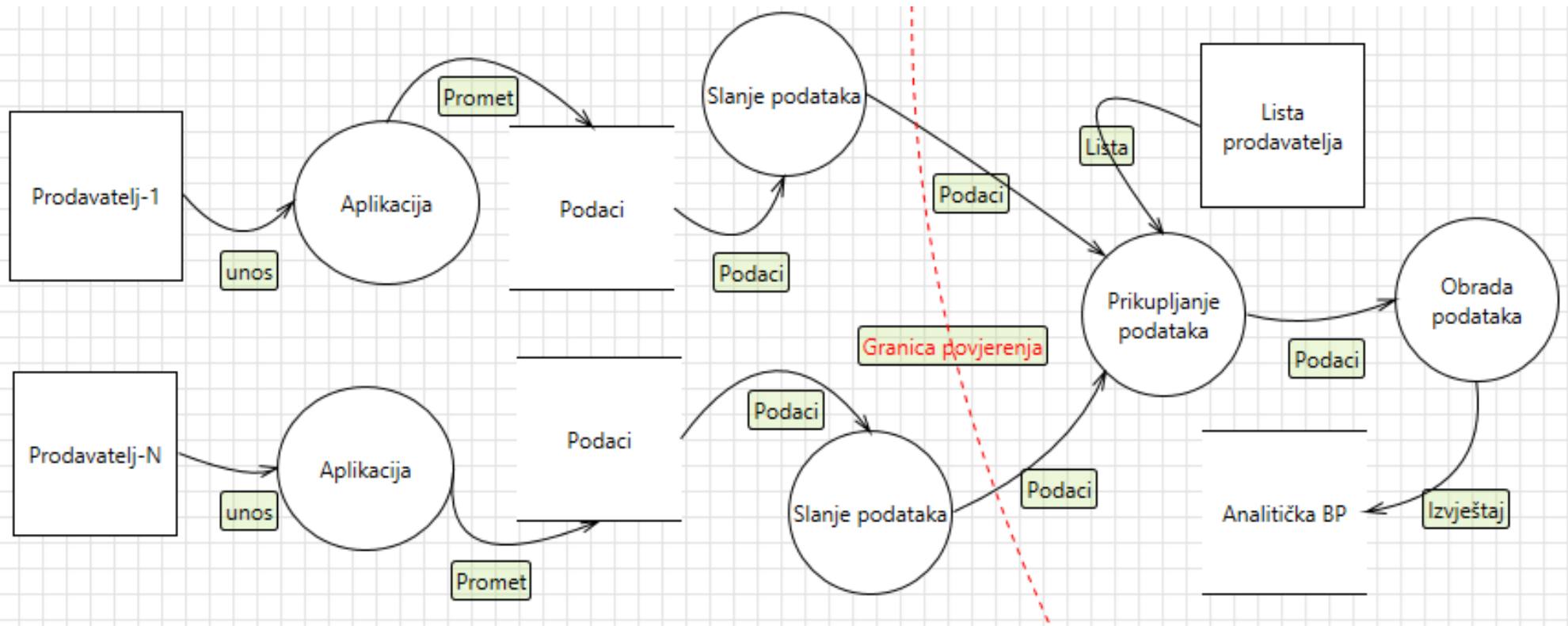
Spoofing	Authentication	<ul style="list-style-type: none"> <li>To authenticate principals: <ul style="list-style-type: none"> <li>Basic &amp; Digest authentication</li> <li>LiveID authentication</li> <li>Cookie authentication</li> <li>Windows authentication (NTLM)</li> <li>Kerberos authentication</li>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

# STRIDE - primjer

---

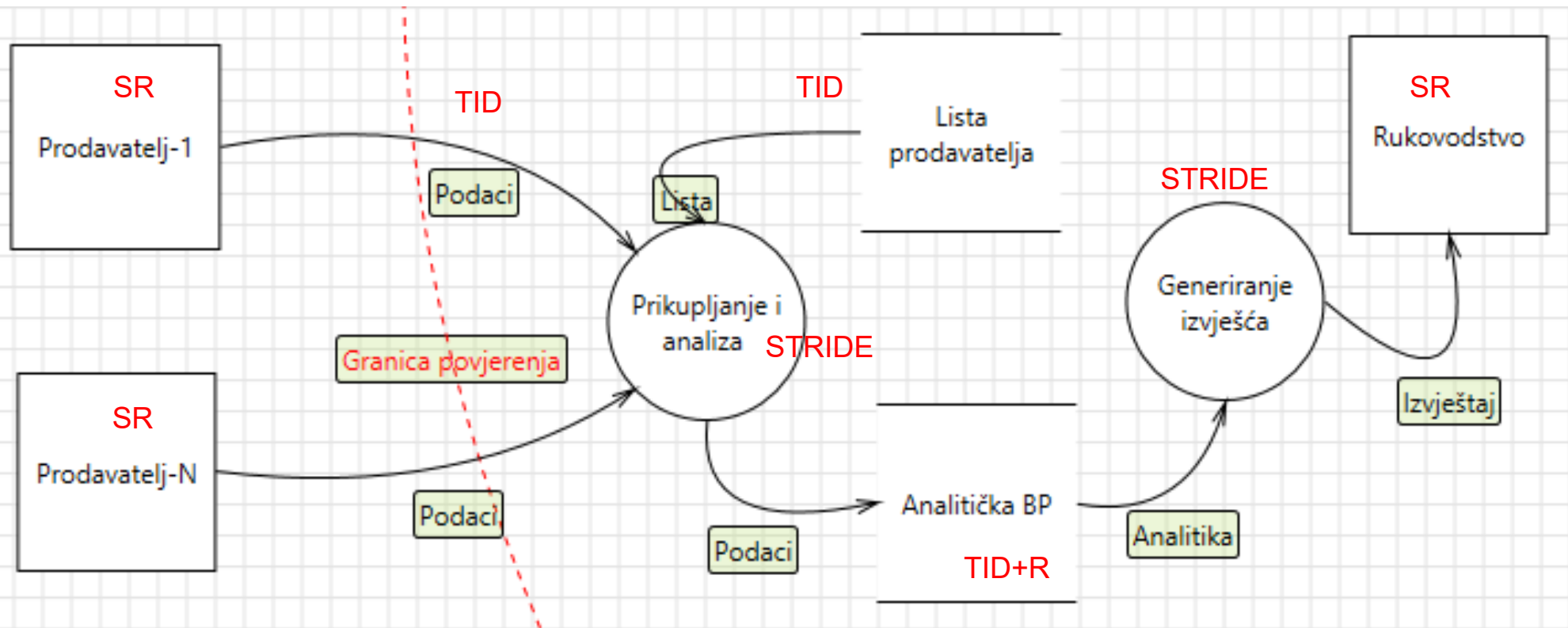
- ◆ Primjer: Uncover Security Design Flaws Using The STRIDE Approach
  - Prodavatelji prikupljaju podatke o prodaji u lokalnim evidencijama
  - Treba prikupiti datoteke o prodaji na poslužitelju
  - Te generirati tjedna izvješća
  - Za prethodno evidentirane prodavatelje
- ◆ Zahtjevi na sigurnost
  -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

# Poboljšani dijagram – analiza poslužiteljske strane

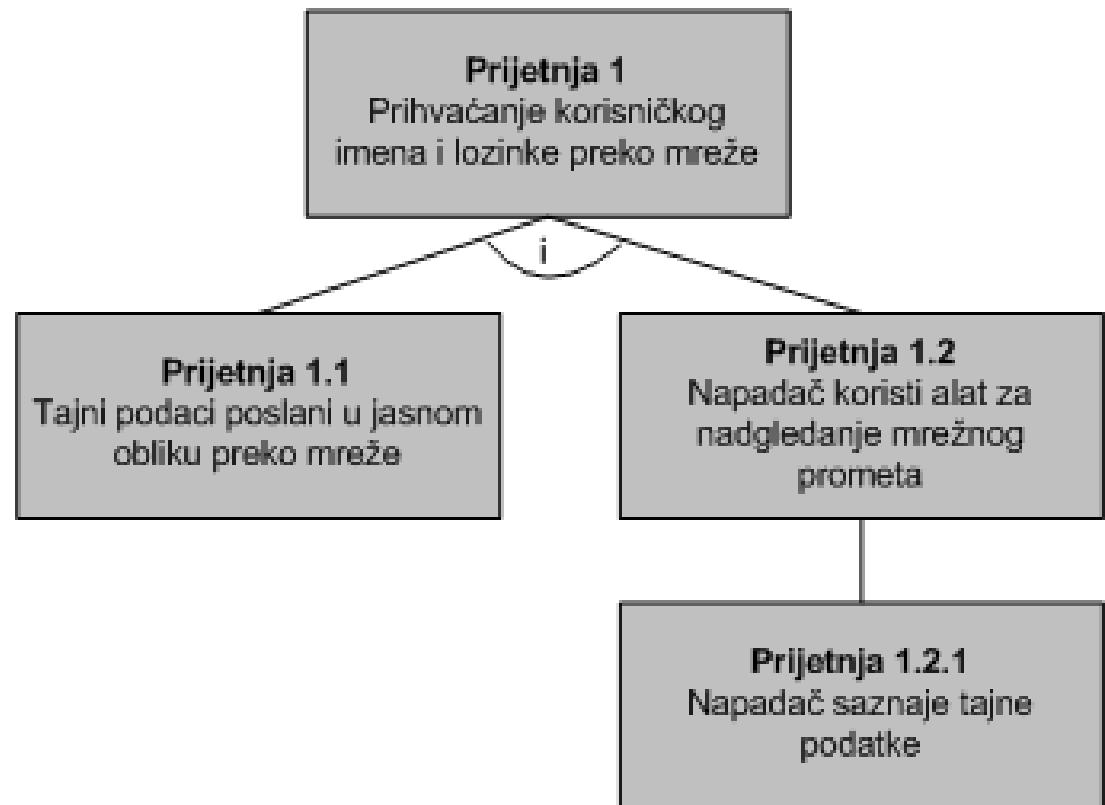


- Izbačen klijent
- Dodano *Generiranje*, posljedično i *Rukovodstvo*
- Lista postala spremište
- Integrirana obrada

◆ 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 poslani u jasnom obliku preko mreže **AND**

1.2 Napadač koristi alat za nadgledanje mrežnog prometa

1.2.1 Napadač saznaje tajne podatke

## ◆ Korištenje STRIDE nad stablima prijetnji je jednostavno

- za svaki dio sustava se ispituje je li podložen nekoj od STRIDE kategorija
- npr. može li napadač uskratiti rad procesa, vidjeti podatak itd.

## ◆ Koje prijetnje ilustrira gornji primjer ?



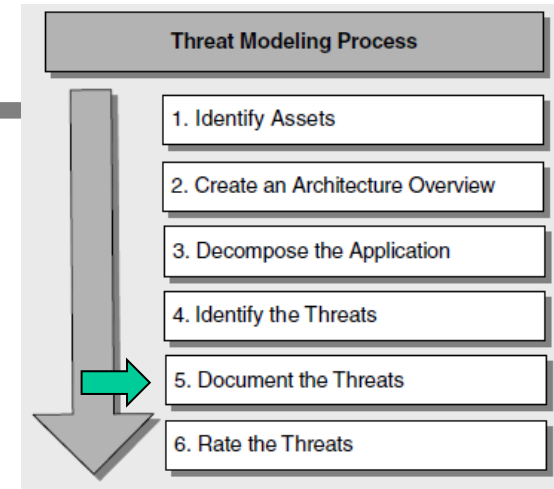
# Obrasci napada

- ◆ 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	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

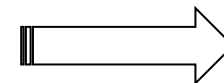
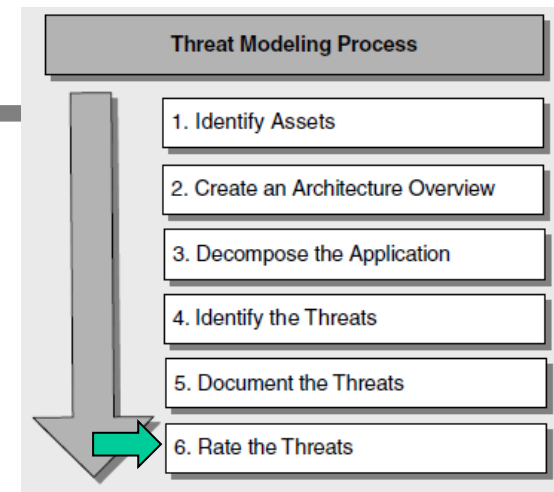


- ◆ Primjeri

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

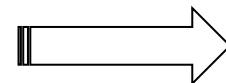
- ◆ Rangiranje – određivanje važnosti (rate the threats)
  - često se koriste tehnike za određivanje rizika
  - **rizik** = vjerojatnost događaja \* potencijalna šteta
  - **vjerojatnost** npr. u rasponu 1-10
  - **šteta** npr. u rasponu 1-10
  - rizik u rasponu 1-100
  - raspodjela u tri grupe (visok, srednji, nizak) koje predstavljaju prioritete
  
- ◆ Problem:
  - članovi tima ne mogu se usuglasiti oko vrijednosti
  
- ◆ Rješenje:
  - DREAD model, rangiranje rizika za zadanu prijetnju



# DREAD (model procjene rizika)

---

- ◆ **DREAD** – klasifikacija računalnih prijetnji
  - **D**amage potential – moguća šteta, veličina štete bude li napad uspješan
  - **R**eproducibility – reproduktivnost, koliko je jednostavno ponoviti napad
  - **E**xploitability – iskoristivost, trud i znanje potrebnih za uspješan napad
  - **A**ffected users – zahvaćeni korisnici, moguće uspješim napadom, postotno
  - **D**iscoverability – 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 ocenjivanja

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

# 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

Threat List

ID	Title	Category	Description	Short Descrip
9	Persistent Cross Site Scripting	Tampering	The web server 'Prikupljanje i an	Tampering is th
10	Cross Site Scripting	Tampering	The web server 'Prikupljanje i anali	Tampering is the

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

Threat Properties 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

---

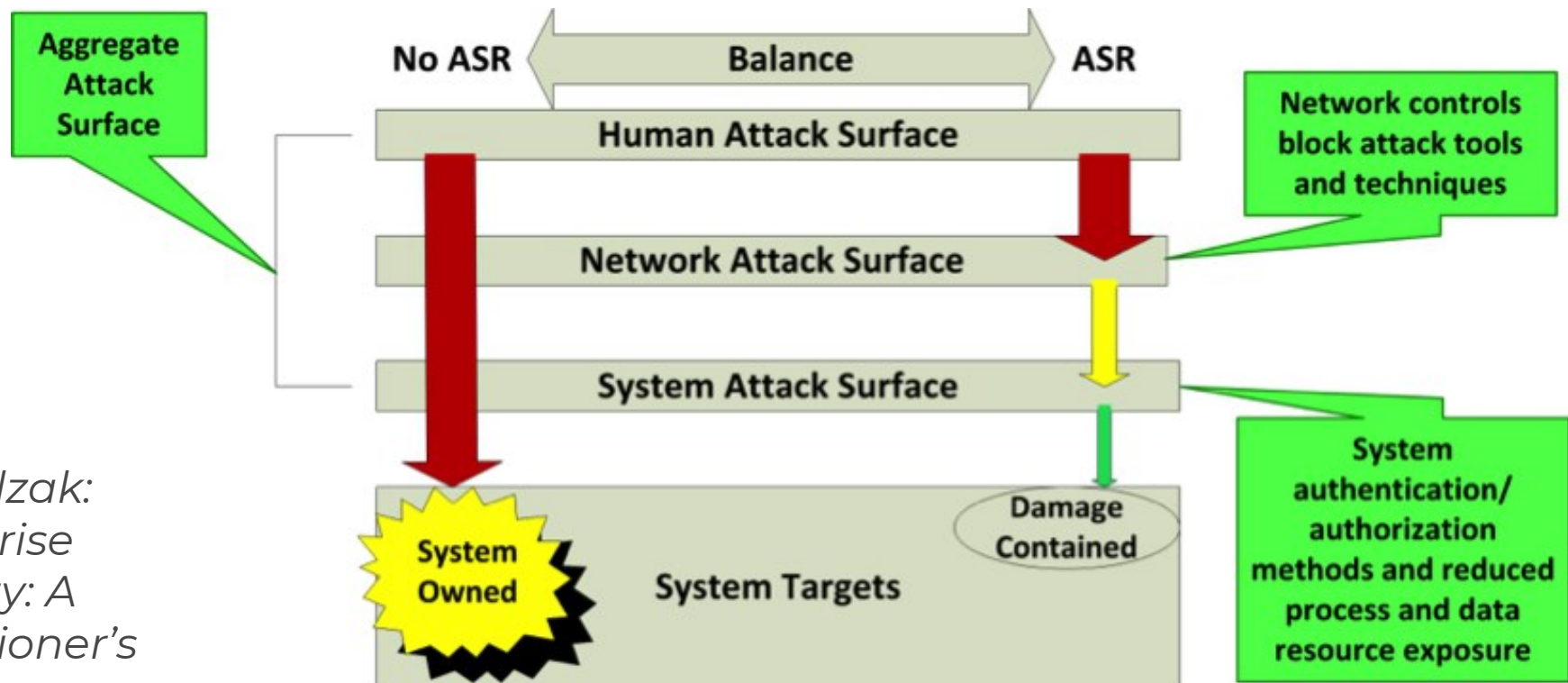
# Površina napada (attack surface)

---

- ◆ **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 (*rw-rw-rw-*), skripte (JavaScript, 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/>
  - Zanimaju specifične konfiguracije
    - Instalirane završne
    - 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

## ◆ Ustanovljavanje pristupnih točki

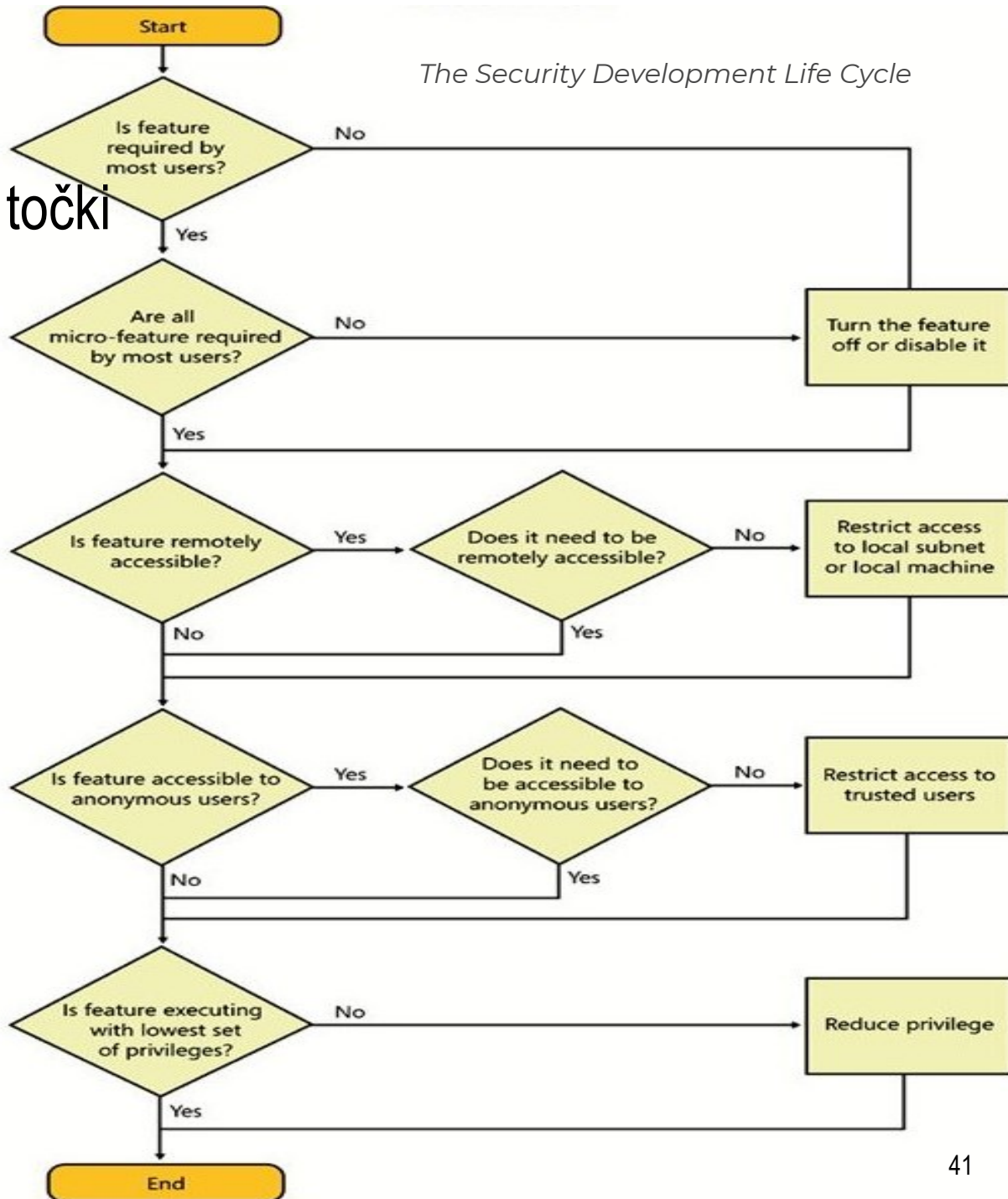
- mrežne, datoteke, ...

## ◆ Rangiranje točaka

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

## ◆ Podešavanje

*The Security Development Life Cycle*



# 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 nepouzdatih (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* anonimnih prijetnji (anonymous threat paths) → autorizacija gdje treba
  - **Ne pokretati servise kao SYSTEM (demone kao *root*) ili s administratorskim pravima dok ne budu iscrpljene druge mogućnosti!**
- ◆ Primjer:
  - *Backup Operator account* – čita sve datoteke bez obzira na njihov ACL
  - SYSTEM radi isto ali i *restore*, *debug*, "*act as part of OS*" i k tomu je *admin*

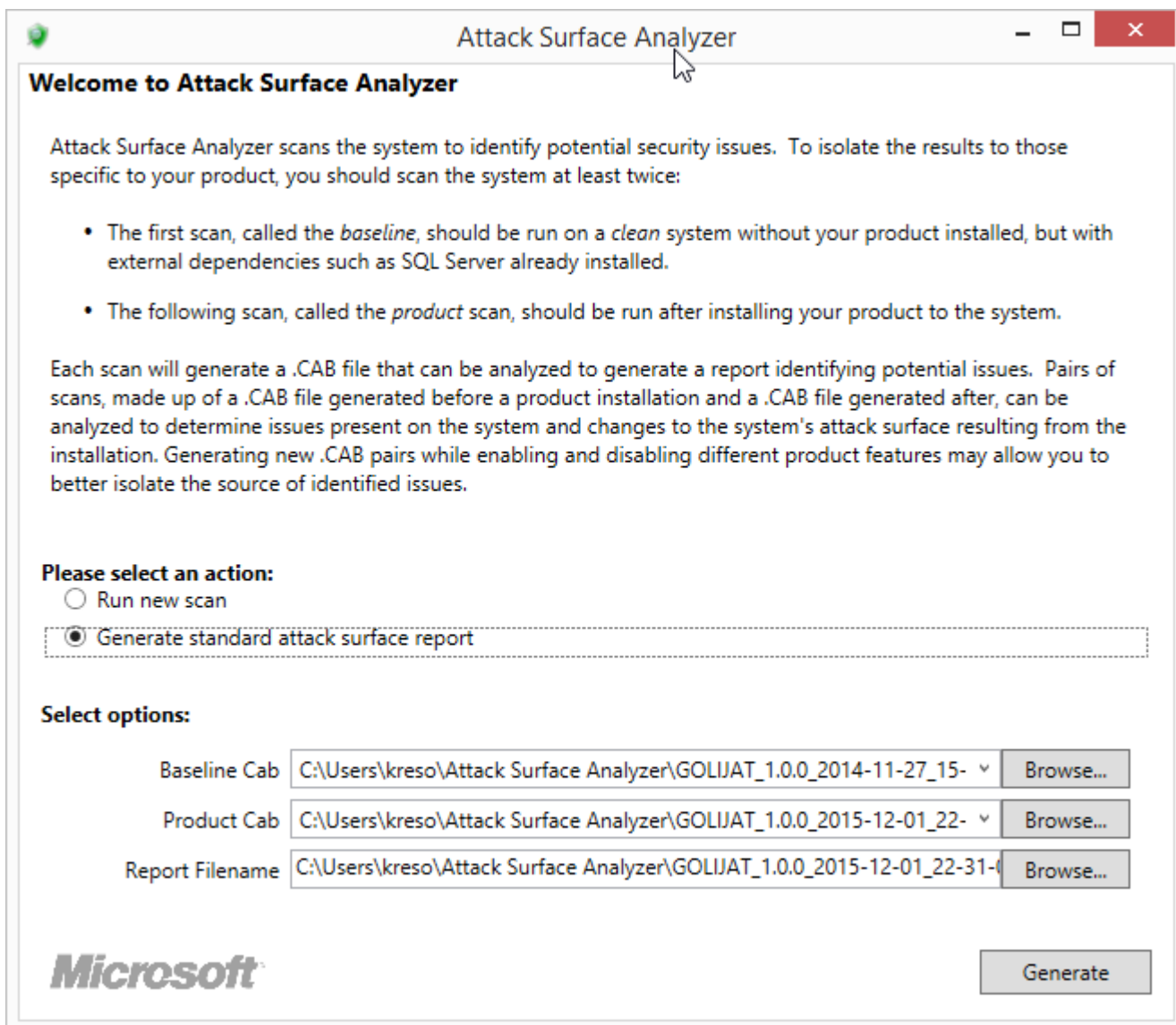
# Najbolje prakse (ostatak)

---

- ◆ 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



**Welcome to Attack Surface Analyzer**

Attack Surface Analyzer scans the system to identify potential security issues. To isolate the results to those specific to your product, you should scan the system at least twice:

- The first scan, called the *baseline*, should be run on a *clean* system without your product installed, but with external dependencies such as SQL Server already installed.
- The following scan, called the *product* scan, should be run after installing your product to the system.

Each scan will generate a .CAB file that can be analyzed to generate a report identifying potential issues. Pairs of scans, made up of a .CAB file generated before a product installation and a .CAB file generated after, can be analyzed to determine issues present on the system and changes to the system's attack surface resulting from the installation. Generating new .CAB pairs while enabling and disabling different product features may allow you to better isolate the source of identified issues.

**Please select an action:**

☐ Run new scan

☒ Generate standard attack surface report

**Select options:**

Baseline Cab: C:\Users\kreso\Attack Surface Analyzer\GOLIJAT\_1.0.0\_2014-11-27\_15-

Product Cab: C:\Users\kreso\Attack Surface Analyzer\GOLIJAT\_1.0.0\_2015-12-01\_22-

Report Filename: C:\Users\kreso\Attack Surface Analyzer\GOLIJAT\_1.0.0\_2015-12-01\_22-31-

**Microsoft**

- [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](#)