Protection and security of information systems

Safety design

prof. Ph.D. Krešimir Fertalj

University of Zagreb

Faculty of Electrical Engineering and Computing

Zaštita i sigurnost informacijskih sustava

Projektiranje sigurnosti

prof. dr. sc. Krešimir Fertalj

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Creative Commons

- **you are free to:**
  - **share**—reproduce, distribute and communicate the work to the public
  - **remix**—rework the work
- **under the following conditions:**
  - **appointment**. You must acknowledge and attribute the authorship of the work in a manner specified by the author or licensor (but not in a manner that suggests that you or your use of their work has their direct endorsement).
  - **non-commercial**. You may not use this work for commercial purposes.
  - **shares under the same conditions**. If you modify, transform, or create using this work, you may distribute the adaptation only under a license that is the same or similar to this one.

In the case of further use or distribution, you must make clear to others the license terms of this work. The best way to do this is to link to this website.

Any of the above conditions may be waived with the permission of the copyright holder.

Nothing in this license infringes or limits the author's moral rights.

The text of the license was taken from http://creativecommons.org/.

Creative Commons

- **slobodno smijete:**
  - **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
  - **remiksirati** — prerađivati djelo
- **pod sljedećim uvjetima:**
  - **imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod 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/.

Threat modeling

Threat Modeling

Modeliranje prijetnji

Threat Modeling

Threat modeling

- threat modeling
  - security analysis that helps uncover the biggest security threats
  - the goal is to determine which threats should be removed and how
  - assumption - the product is not safe if the threats are not assessed and the risk is reduced

- uses:
  - better understanding of the application
    - especially new members
  - finding errors
    - estimate that MP finds 50% of errors, and the rest by testing and analyzing the code
    - complex application errors, which are rarely found otherwise (design errors)

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)

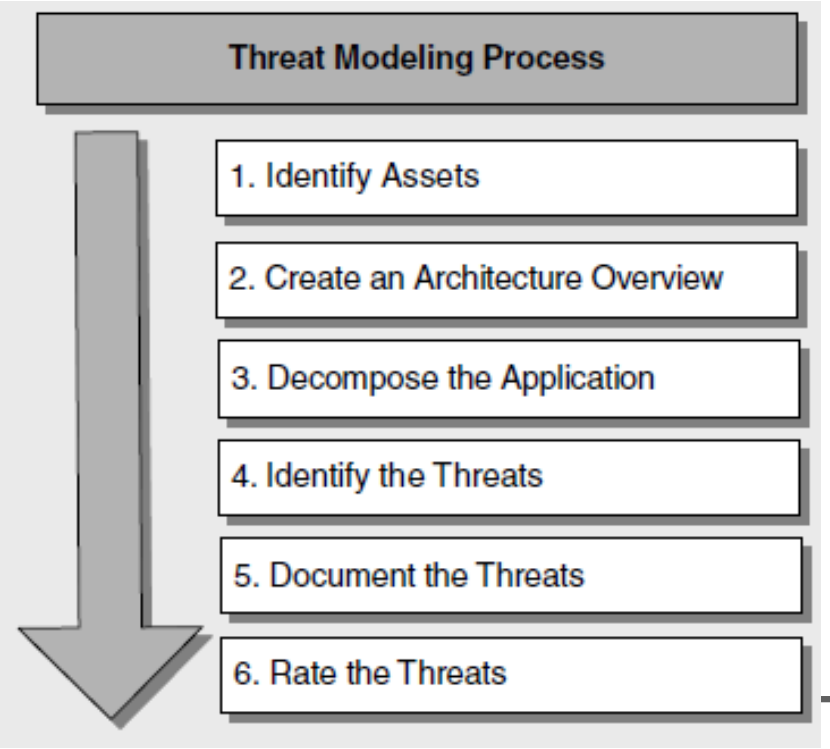
Principles and process of threat modeling

- Analyzing threats is a time-consuming job
  - it is important that it is done well
  - best iteratively

important:

  - It's easier to find a security flaw in an application's design than to change it later
  - The threat model should be current (up-to-date) - threats and ways to circumvent them

- Threat modeling process
  - Determination of protection objectives
  - Application architecture
  - Decomposition of the application
  - Determining threats
  - Documenting threats
  - Threat ranking



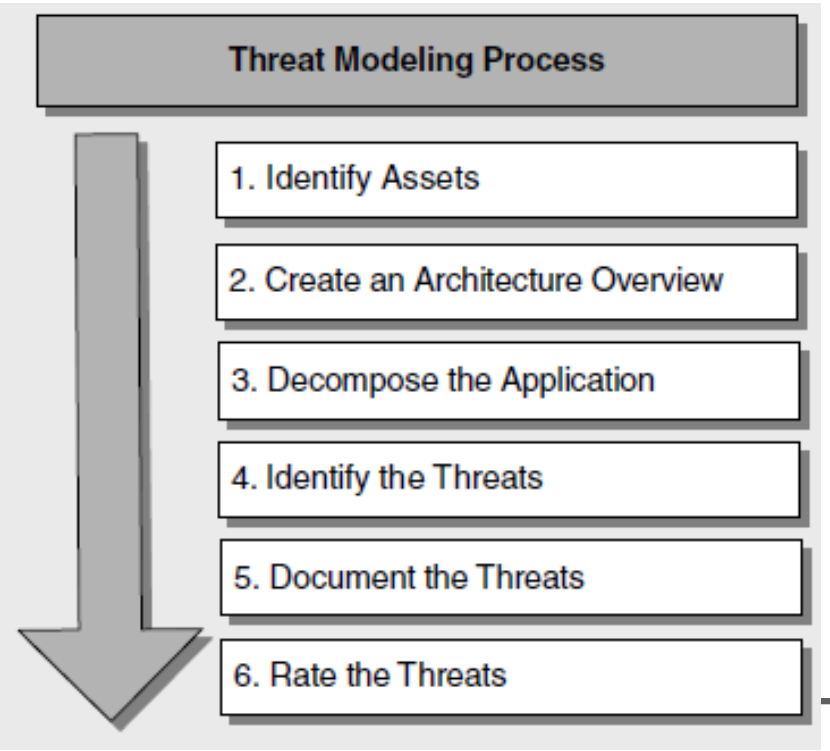
Načela i proces modeliranja prijetnji

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

važno:

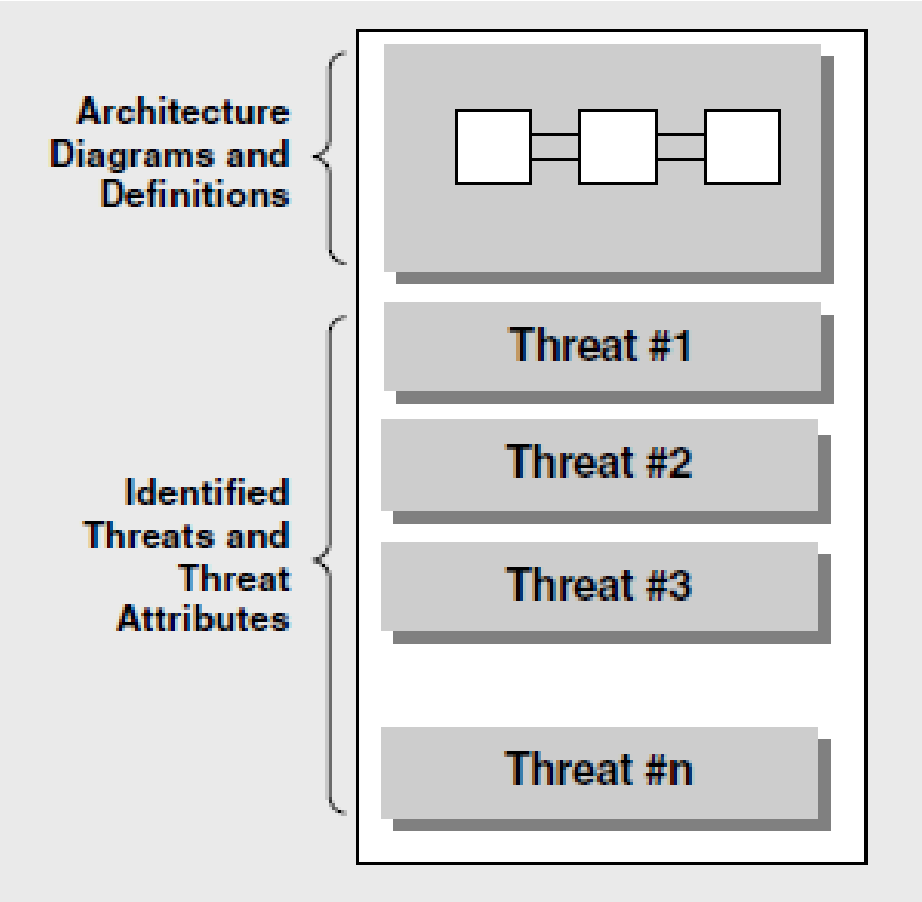
  - Jednostavnije je pronaći sigurnosni propust u dizajnu aplikacije nego mijenjati kasnije
  - Model prijetnji treba biti aktualan (ažuran) - prijetnje i načini kako ih zaobići

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

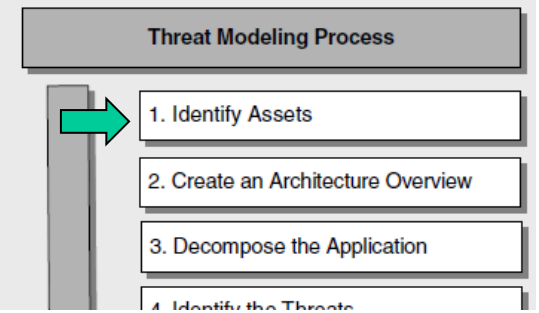


Exit

- Document with models
  - by the definition of architecture and
  - list of threats

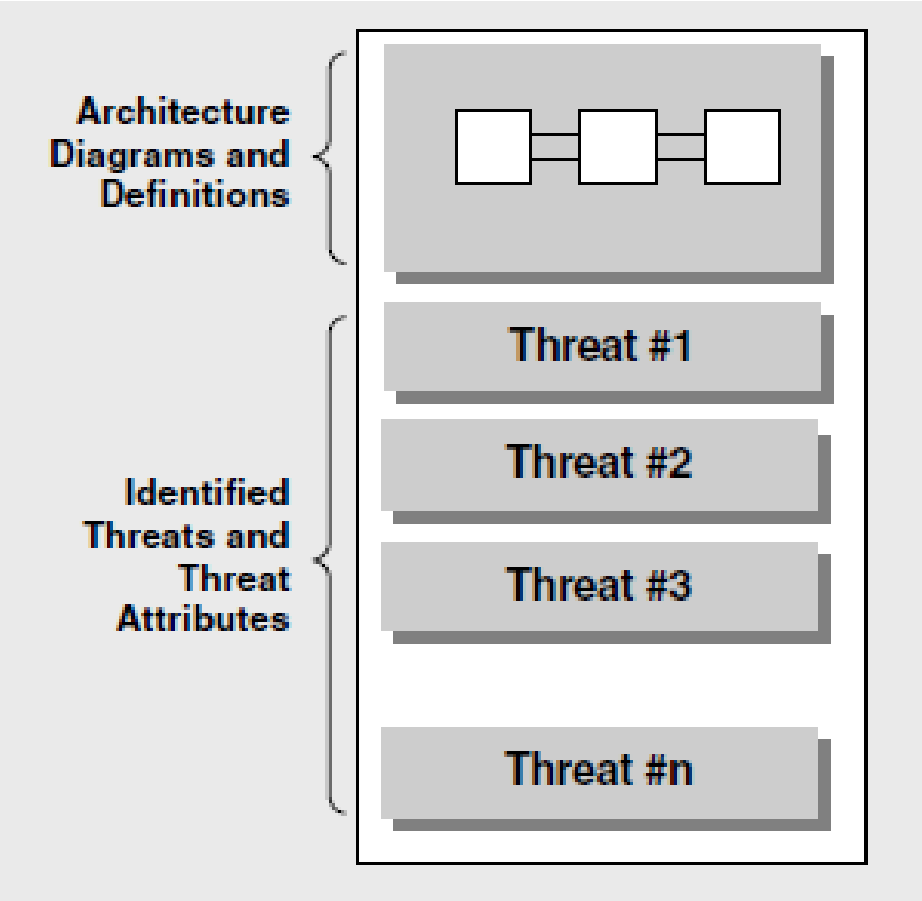


- **Step 1**–identification of resources to be protected
  - from data repositories (file, BP), …, to web pages

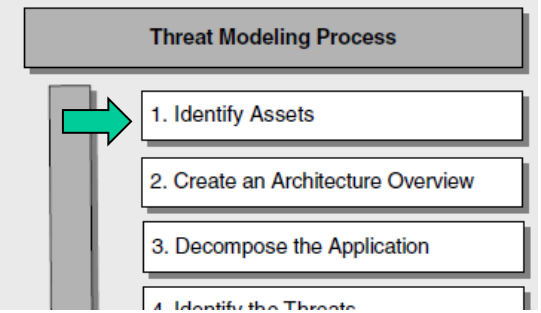


Izlaz

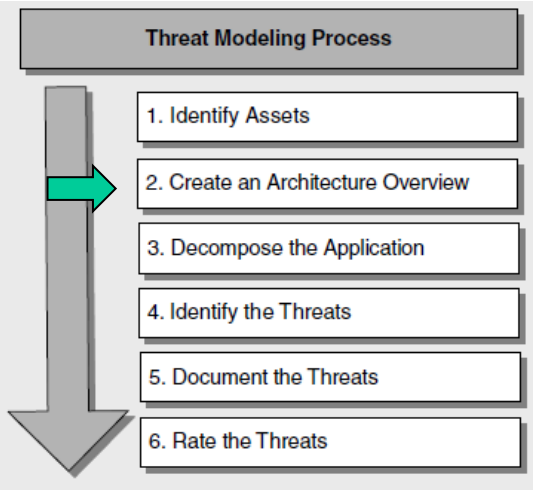
- Dokument s modelima
  - definicijom arhitekture i
  - popisom prijetnji



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

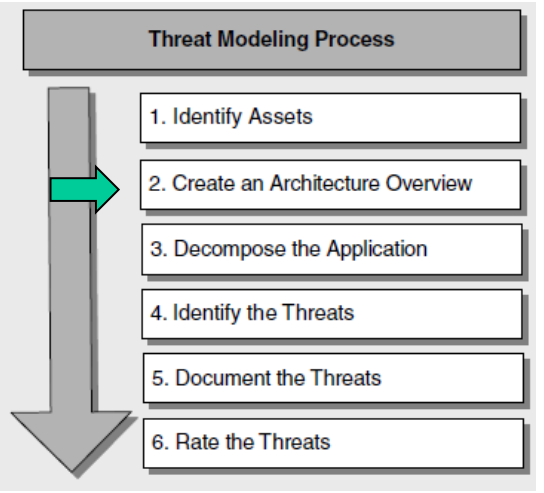


Step 2 – Architecture Review



- Documentation
  - application functions - what the application does
  - application architecture and physical installation method (configuration)
  - implementation technologies
- Functionality modeling
  - use cases (*use cases*)
  - understanding how to use it
  - the context of the application
  - examples:
    - the employee sees business data, can update personal data,
    - the manager sees the employee's data.
- Checking (violations) of business rules
  - For example the user tries to change someone else's personal data
  - He shouldn't do that if he doesn't have a sufficient level of permissions

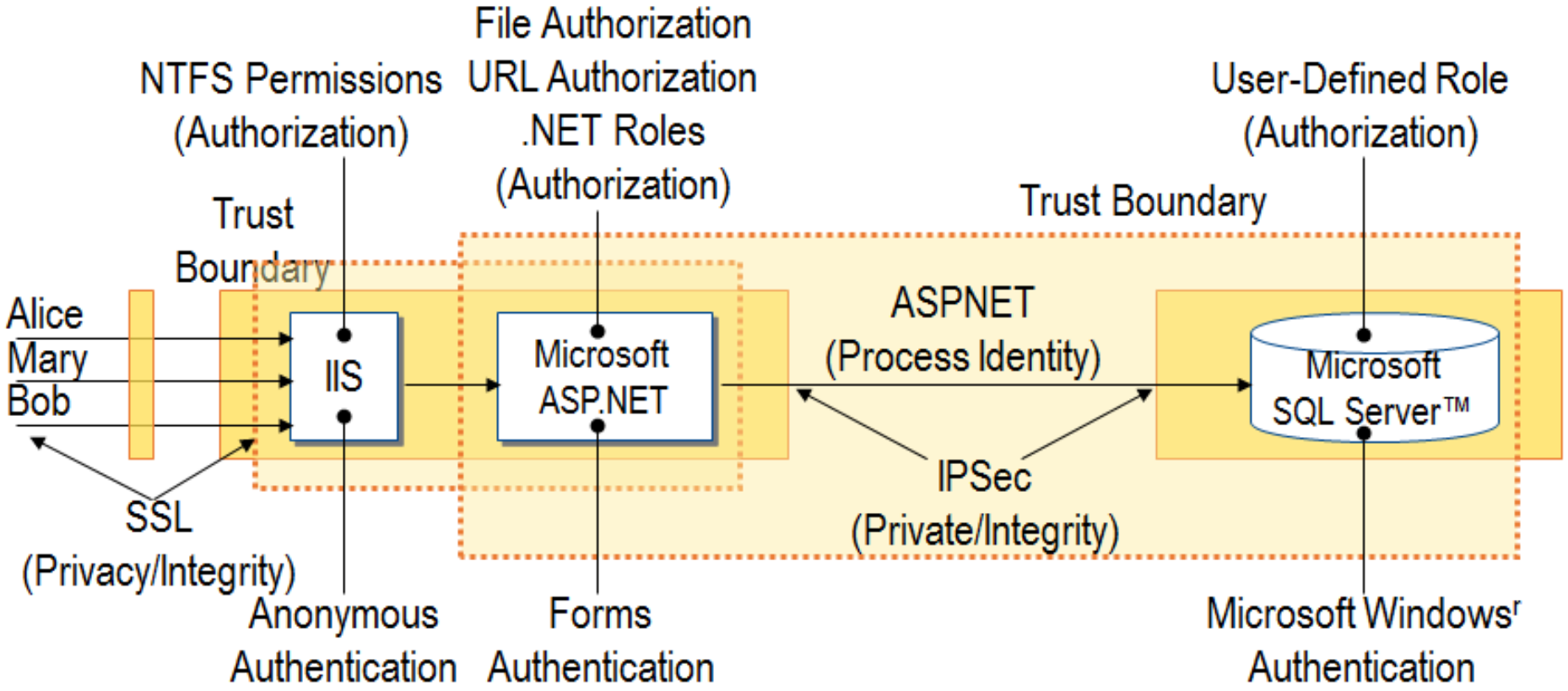
Korak 2 – Pregled arhitekture



- 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

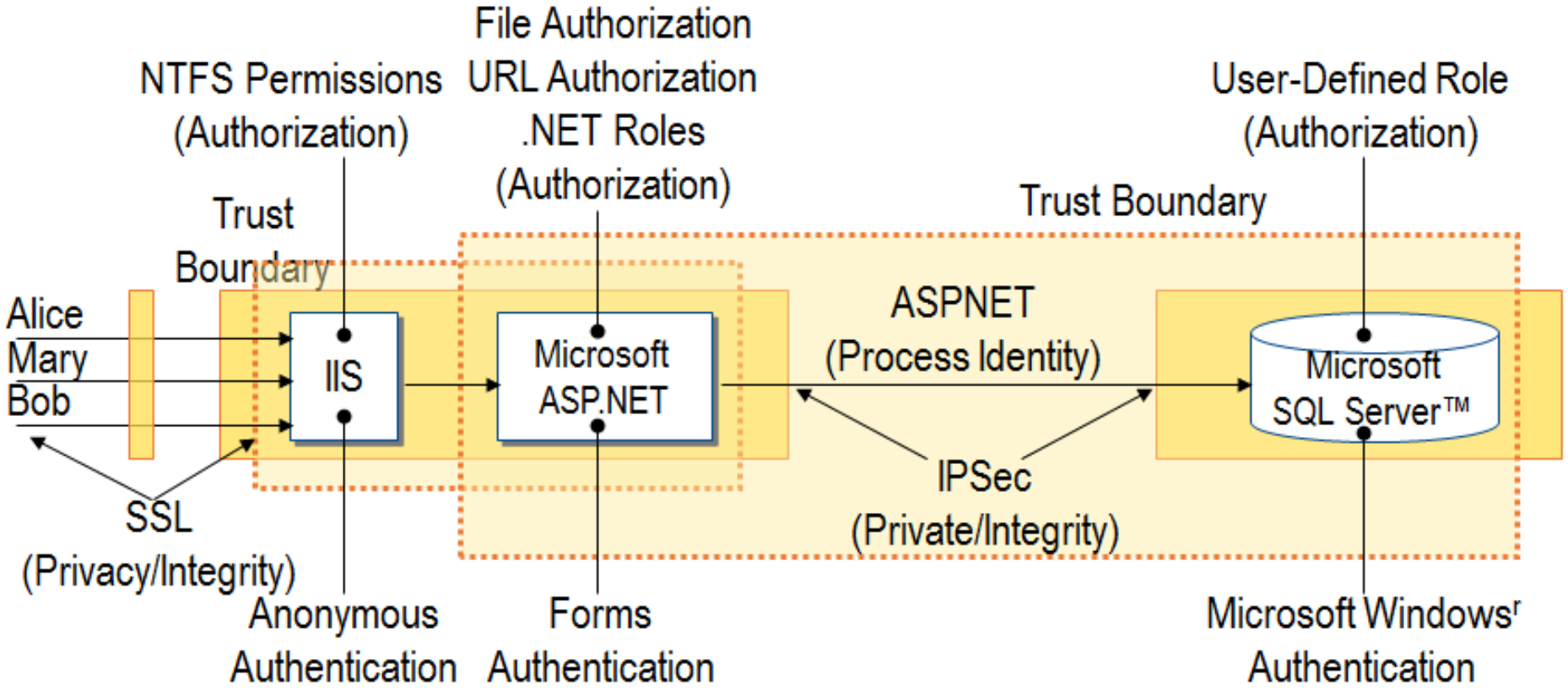
Architecture and implementation technologies

- High-level diagram - describes the structure (components) of the system
  - depending on the complexity of the application, more detailed diagrams of parts should be created
    - eg diagrams of individual layers of a multi-layered application
  - determination of implementation technologies to which aspects of protection are added, e.g.

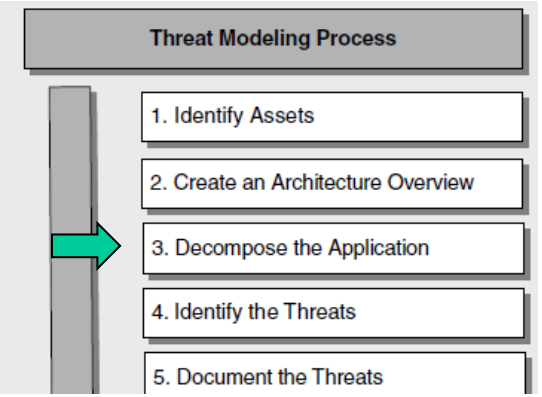


Arhitektura i tehnologije implementacije

- 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.



Step 3 – Application Decomposition

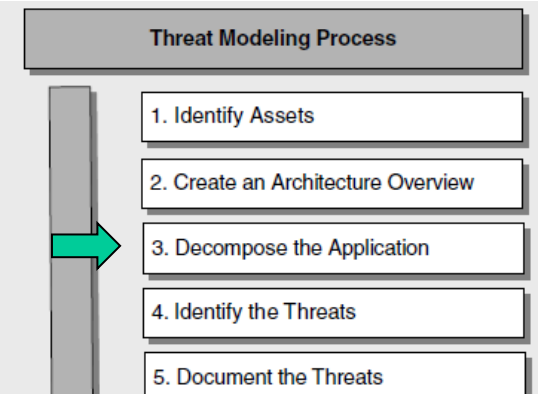


- Creating a security profile
- Determination
  - trust boundaries
  - data flow
  - places of entry
  - privileged code
- For every application

Application Decomposition		
Security Profile		Trust Boundaries
Input Validation	Session Management	Data Flow
Authentication	Cryptography	Entry Points
Authorization	Parameter Manipulation	Privileged Code
Configuration Management	Exception Management	
Sensitive Data	Auditing and Logging	

- Decomposition techniques
  - functional decomposition, activity diagram, data flow diagram, ...

Korak 3 – Dekompozicija aplikacije



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

Application Decomposition

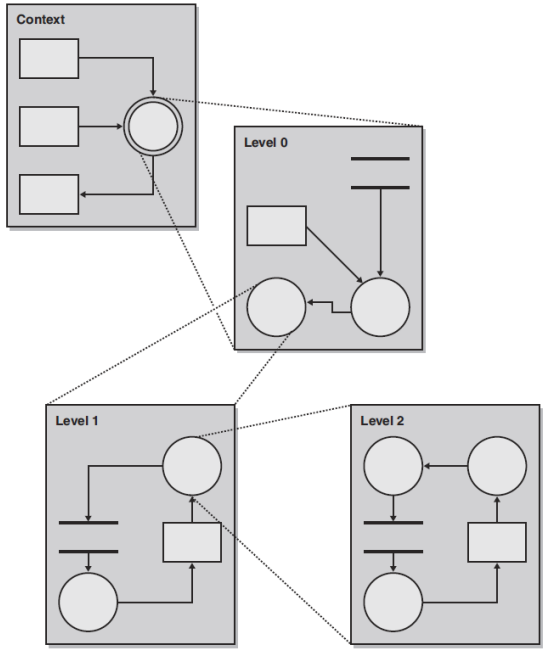
Security Profile		Trust Boundaries
Input Validation	Session Management	Data Flow
Authentication	Cryptography	Entry Points
Authorization	Parameter Manipulation	Privileged Code
Configuration Management	Exception Management	
Sensitive Data	Auditing and Logging	

- Tehnike dekompozicije
  - funkcionalna dekompozicija, dijagram aktivnosti, dijagram toka podataka, …

Confidence limits and data flows

- Determination of confidence limits
  - analysis of the resource environment determined by the application design
  - for each subsystem, an evaluation of whether the input stream or user input is confidential
    - if not - consider how to authenticate and authorize them
  - assessment of whether the calling code is confidential
  - checking server trust relationships (server trust relationships)

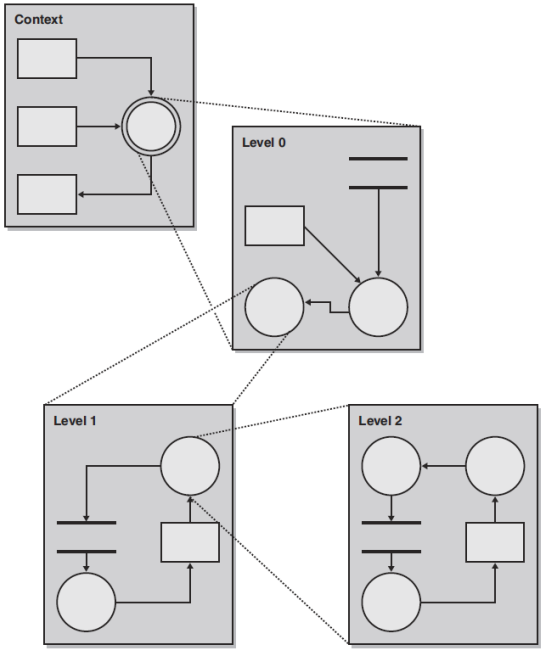
- Determination of data flow (data flow)
  - iterative decomposition
  - by analyzing flows between subsystems, and in depth
    - Levels: 0-system, 1-main capabilities, 2-details



Granice povjerenja i tokovi podataka

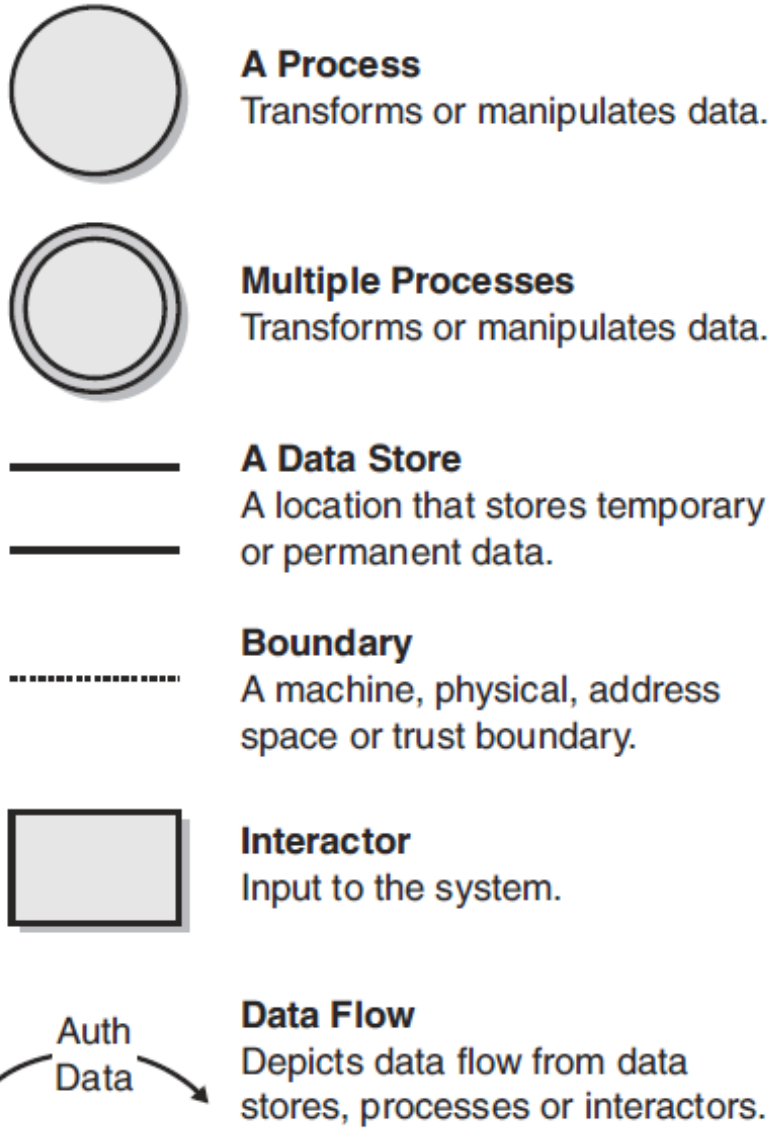
- Određivanje granica povjerenja
  - analiza okruženja resursa određenog dizajnom aplikacije
  - za svaki podsustav, procjena je li ulazni tok ili korisnički unos povjerljiv
    - ako nije – razmotriti kako ih autentificirati i autorizirati
  - procjena je li pozivajući programski kod povjerljiv
  - 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



Data flow diagram - notation

- A process, a multiple process
  - data processing, or action based on data
  - collection of subprocesses, can be decomposed
- Data storage
  - Any form of storage (file, BP, ...)
- Confidence limit
  - mark of privilege change (data rights levels)
- External entity, participant
  - everything that is outside the application, and interacting through the entry point
- Data flow
  - directed movement of data within the application



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

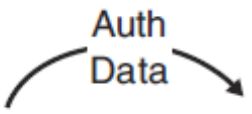
A Process
Transforms or manipulates data.

Multiple Processes
Transforms or manipulates data.

A Data Store
A location that stores temporary or permanent data.

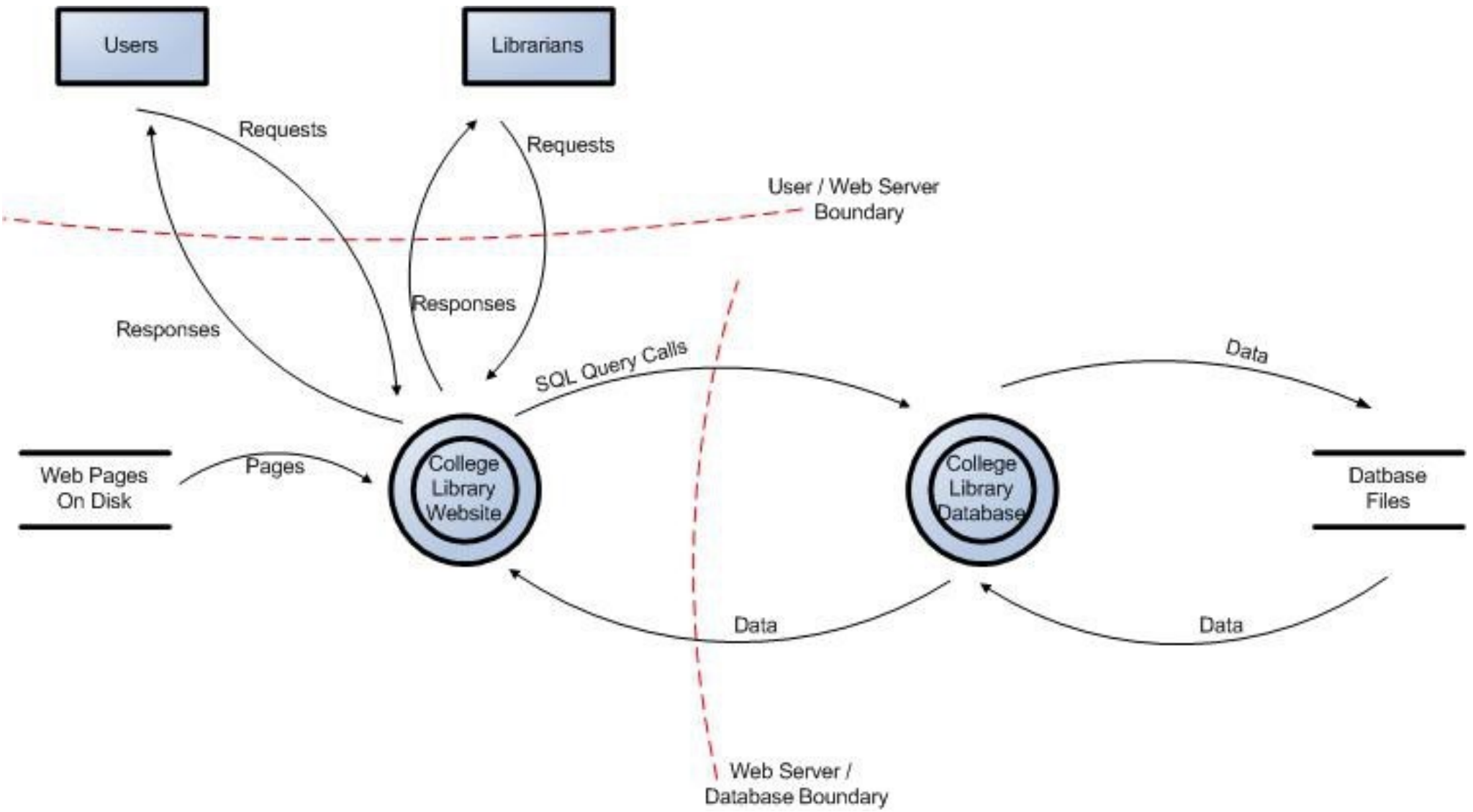
Boundary
A machine, physical, address space or trust boundary.

Interactor
Input to the system.

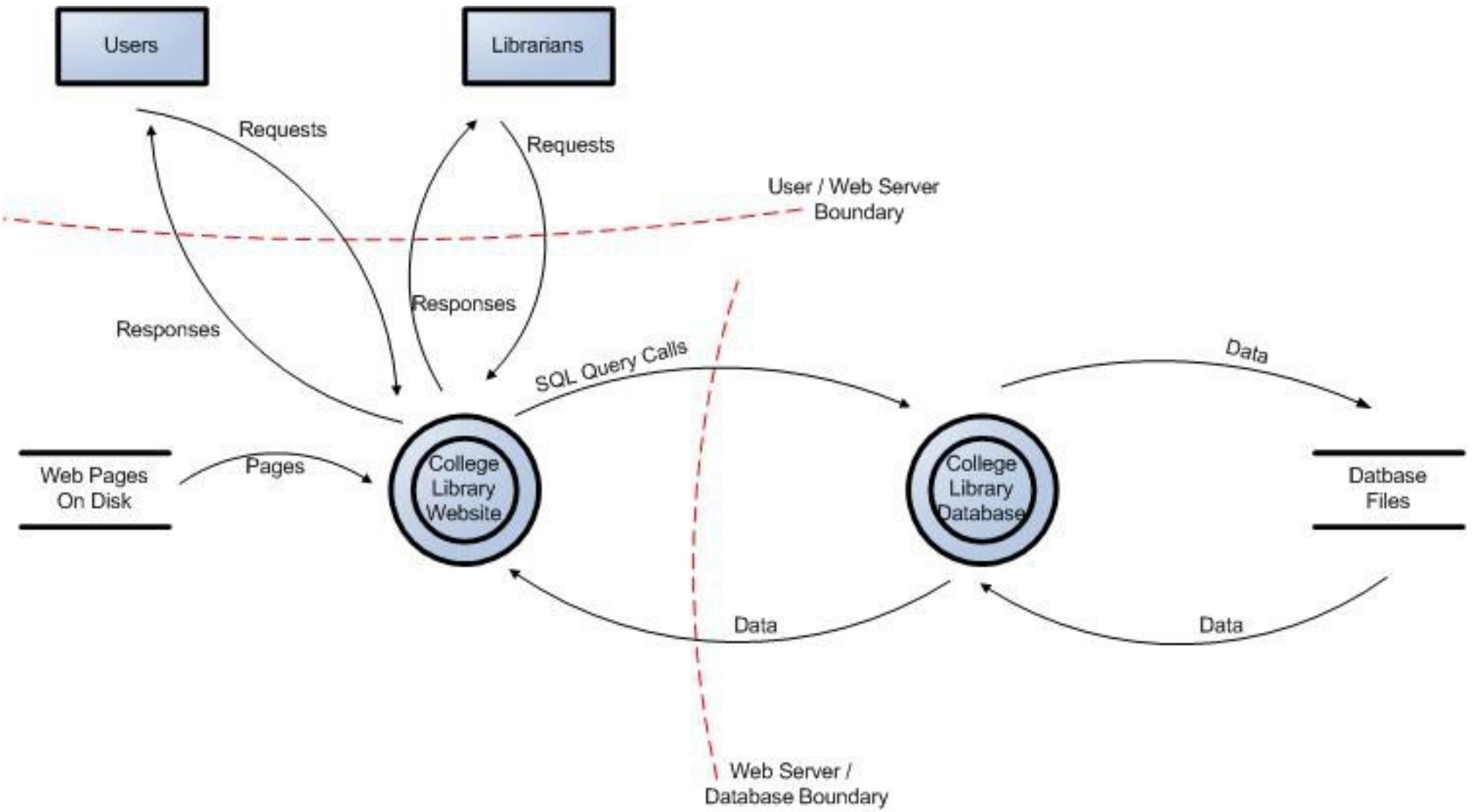


Data Flow
Depicts data flow from data stores, processes or interactors.

Data flow diagram - example



Dijagram toka podataka - primjer

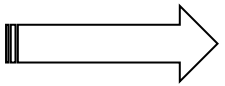


Other decomposition activities

- Determining the entry point
  - parts of the user interface, eg web application pages
  - data transfer connection points, e.g. web service interfaces, remoting components, physical ports and sockets

- Specifying a privileged code
  - that accesses certain types of secure resources or performs privileged operations
  - ex. non/secure resources: DNS servers, *registry*, *event log*, ..., printers, web services, ...
  - ex. non/safe operations: *unmanaged code calls*, reflection, serialization, ...

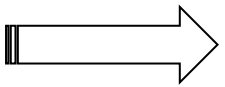
- Documenting the security profile
  - determining access to design and installation for input validation, authentication, authorization, configuration management, ...
  - examples of questions to answer when creating a profile



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 

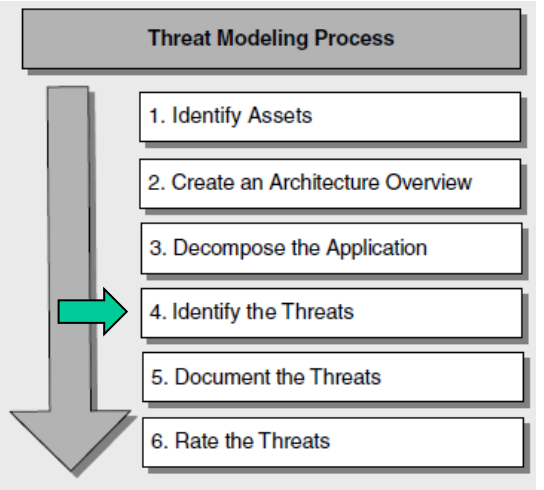
Category	Considerations
Input validation	Is all input data validated? Could an attacker inject commands or malicious data into the application? Is data validated as it is passed between separate trust boundaries (by the recipient entry point)? Can data in the database be trusted?
Authentication	Are credentials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

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

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

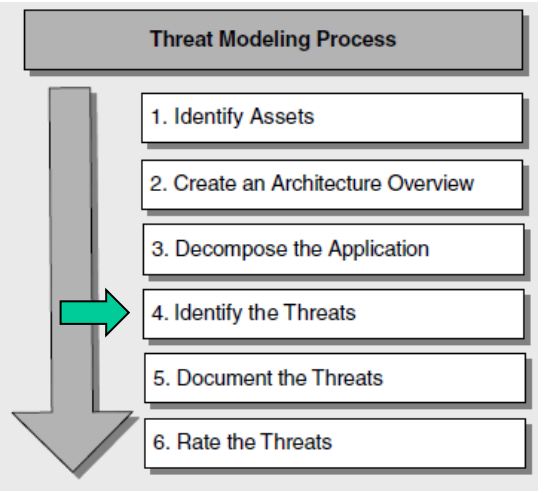
Category	Considerations
Session management	How are session cookies generated? How are they secured to prevent session hijacking? How is persistent session state secured? How is session state secured as it crosses the network? How does the application authenticate with the session store? Are credentials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

Step 4 - Threat Determination



- They are done by the development team and the testing team
  - architects, security guards, developers, testers and system administrators
- Basic approaches
  - **OYSTERS**modeling practice defined by SDL
    - acronym (*Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege*)
  - Categorized threat lists
    - list of commonly "suspicious" threats (*laundry list*)
    - grouped by categories: network, server, application
    - applying the list to your own architecture
- Other useful techniques
  - Threat Trees (*threat trees*)
    - describe what decisions an attacker must make when attacking a component
  - Attack Patterns (*attack patterns*)

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 - assessment by threat categories

- **WITH**poofing – deceiving, faking
  - assuming someone else's identity in order to access resources in the network
  - eg illegal retrieval of other people's data during authentication
- **T**ampering [with Data] – malicious modification of data
  - unauthorized modification, for example, in the database or during network transmission
- **R**epudiation – non-recognition, denial
  - the ability of the user to deny the action without being able to prove it to him
  - eg "I did not delete", "I did not order", ...
- **AND**nformation disclosure - disclosure of information
  - unwanted exposure of private data
  - for example, the user sees the content of someone else's file to which he has no right
- **D**enial of service - denial of service
  - prevents the normal operation of the system, relatively simply and anonymously
  - for example *flooding, amplification, protocol vulnerability, malformed packets*
- **E**elevation of privilege - elevation of authority
  - a user with limited authority assumes the identity of a user with higher authority

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 - procedure

- The system is broken down into relevant components
  - the threat sensitivity of each component is assessed
  - threats are reduced (mitigation) by appropriate security features
  - it is repeated (recursively) until a satisfactory result

- The impact of threats on individual parts of the system
  - ... by analyzing data flow diagrams

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

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			

Threats and standard countermeasures

laundry list

Spoofing	Authentication	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 To authenticate code or data: • Digital signatures • Message authentication codes • Hashes
Tampering	Integrity	• Windows Mandatory Integrity Controls • ACLs • Digital signatures • Message Authentication Codes
Repudiation	Non Repudiation	• Strong Authentication • Secure logging and auditing • Digital Signatures • Secure time stamps • Trusted third parties
Information Disclosure	Confidentiality	• Encryption • ACLS
Denial of Service	Availability	• ACLs • Filtering • Quotas • Authorization • High availability designs
Elevation of Privilege	Authorization	• ACLs • Group or role membership • Privilege ownership • Permissions • Input validation

Prijetnje i standardne protumjere

„laundry list"

Spoofing	Authentication	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 To authenticate code or data: • Digital signatures • Message authentication codes • Hashes
Tampering	Integrity	• Windows Mandatory Integrity Controls • ACLs • Digital signatures • Message Authentication Codes
Repudiation	Non Repudiation	• Strong Authentication • Secure logging and auditing • Digital Signatures • Secure time stamps • Trusted third parties
Information Disclosure	Confidentiality	• Encryption • ACLS
Denial of Service	Availability	• ACLs • Filtering • Quotas • Authorization • High availability designs
Elevation of Privilege	Authorization	• ACLs • Group or role membership • Privilege ownership • Permissions • Input validation

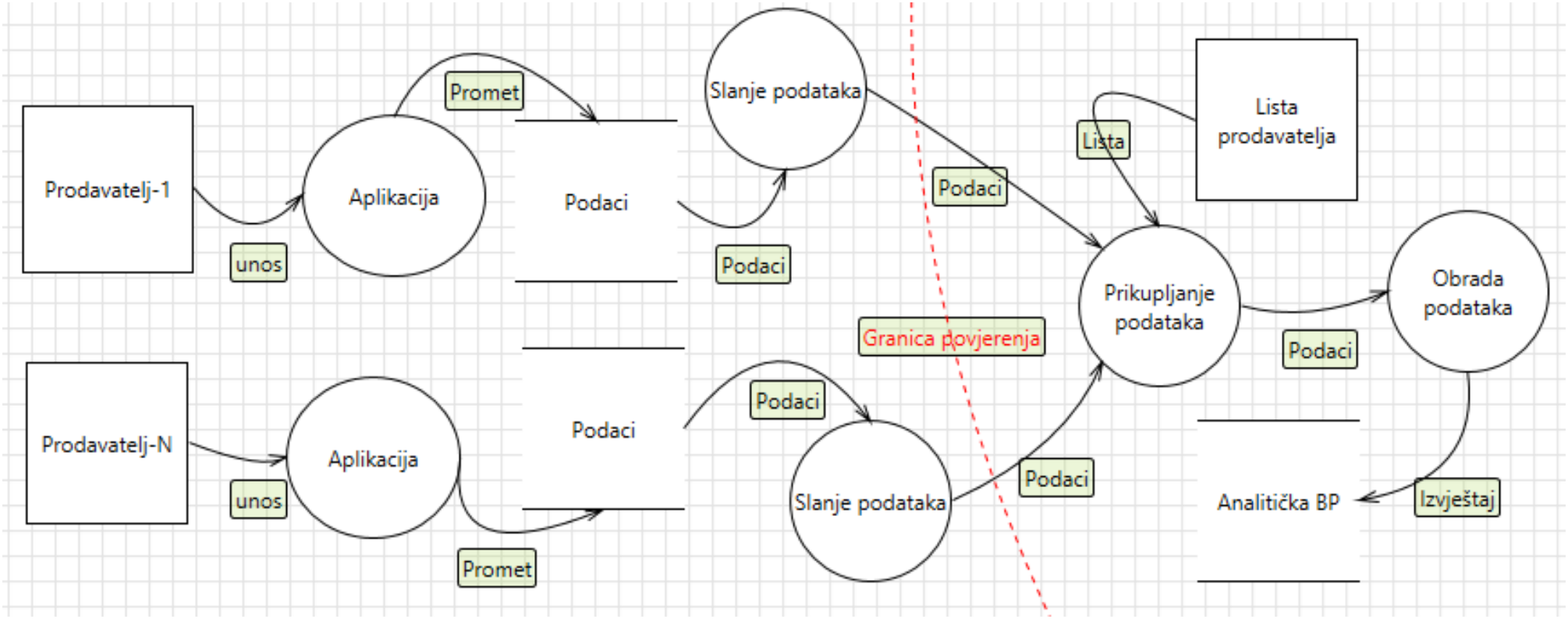
STRIDE - example

- Example: Uncover Security Design Flaws Using The STRIDE Approach
  - Sellers collect sales data in local records
  - Sales files should be collected on the server
  - And generate weekly reports
  - For previously registered sellers
- Security requirements
  - Protect data during transmission and storage
  - Authenticate and authorize sellers
  - Application resistant to attacks (intakes, injections, overflows)
  - ...
- the user does not need to pronounce them all - they need to be invented in view of the problem

STRIDE - primjer

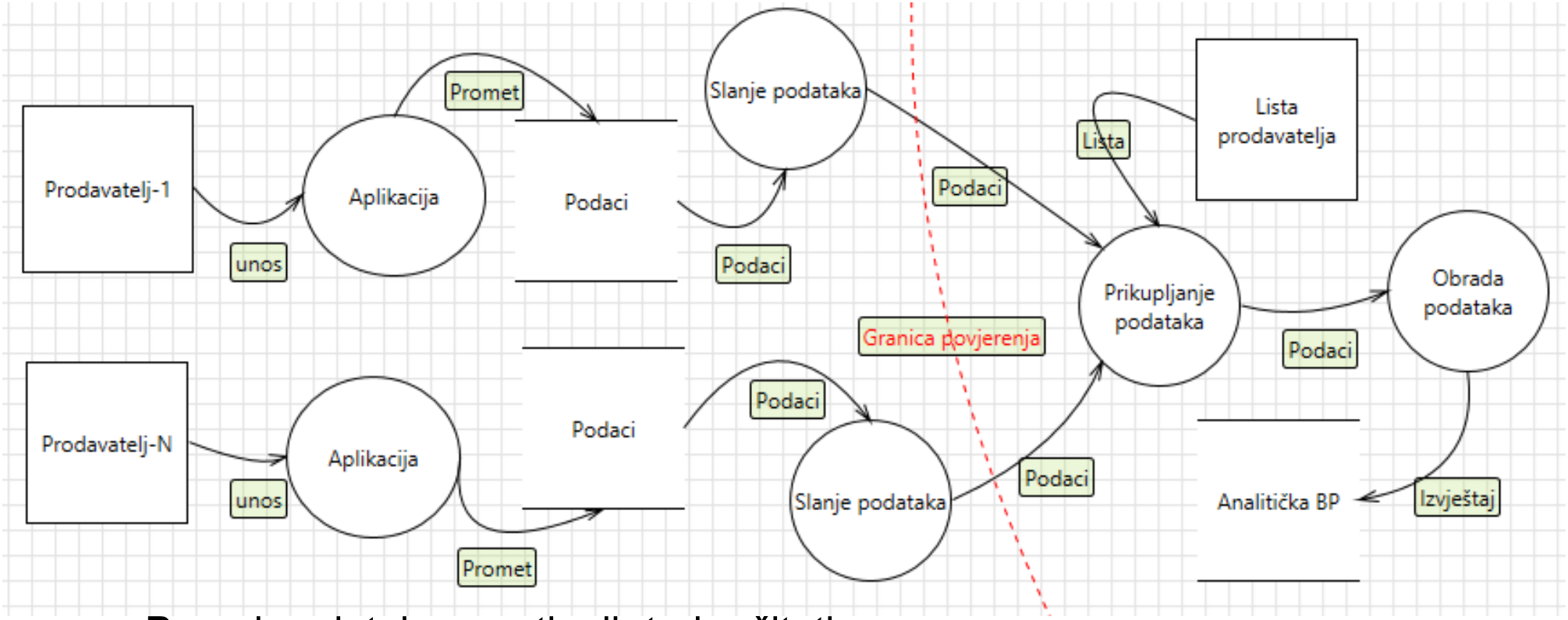
- Primjer: Uncover Security Design Flaws Using The STRIDE Approach
  - Prodavatelji prikupljaju podatke o prodaji u lokalnim evidencijama
  - Treba prikupiti datoteke o prodaji na poslužitelju
  - Te generirati tjedna izvješća
  - Za prethodno evidentirane prodavatelje
- Zahtjevi na sigurnost
  - Zaštititi 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

Initial diagram – client/server



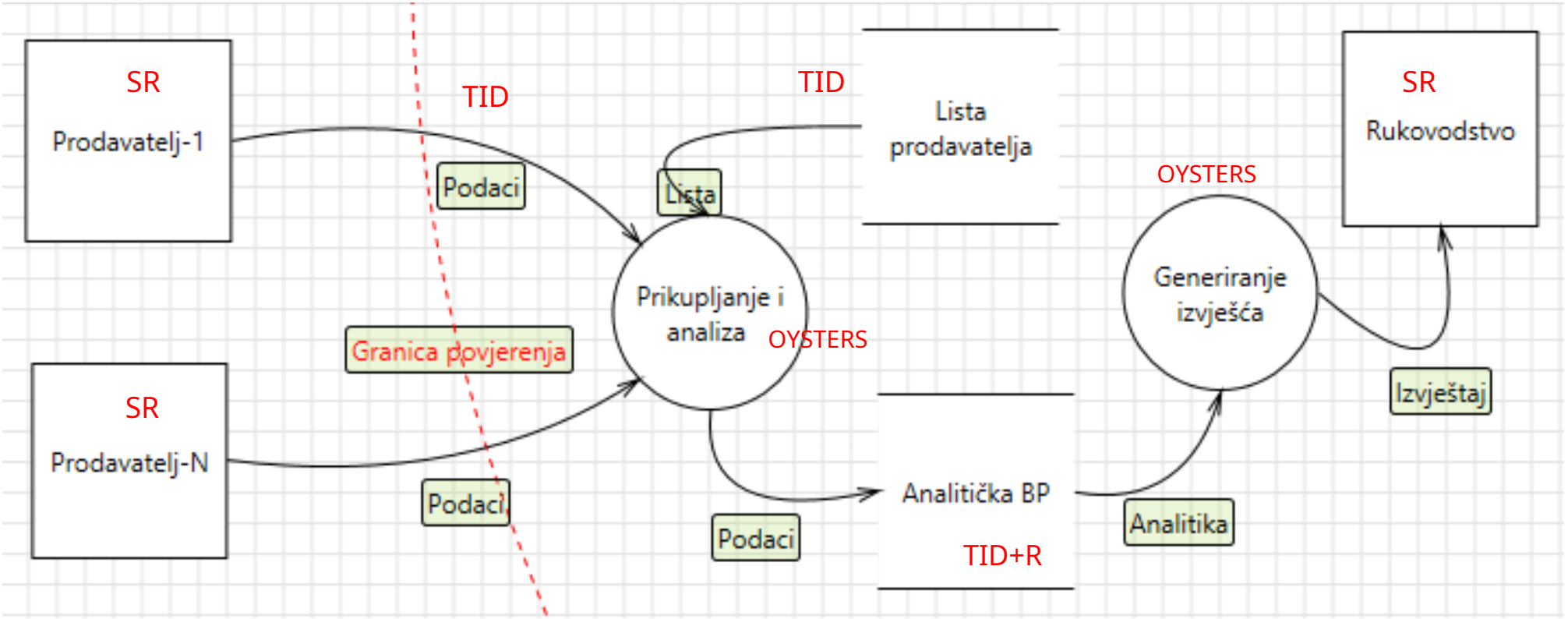
- Data sinks - someone needs to read them, by process
- Chained processes - indicate dependence, separate
- Redundancy – generalize and normalize behavior (functionality)
- Wrong data sources - check, change

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

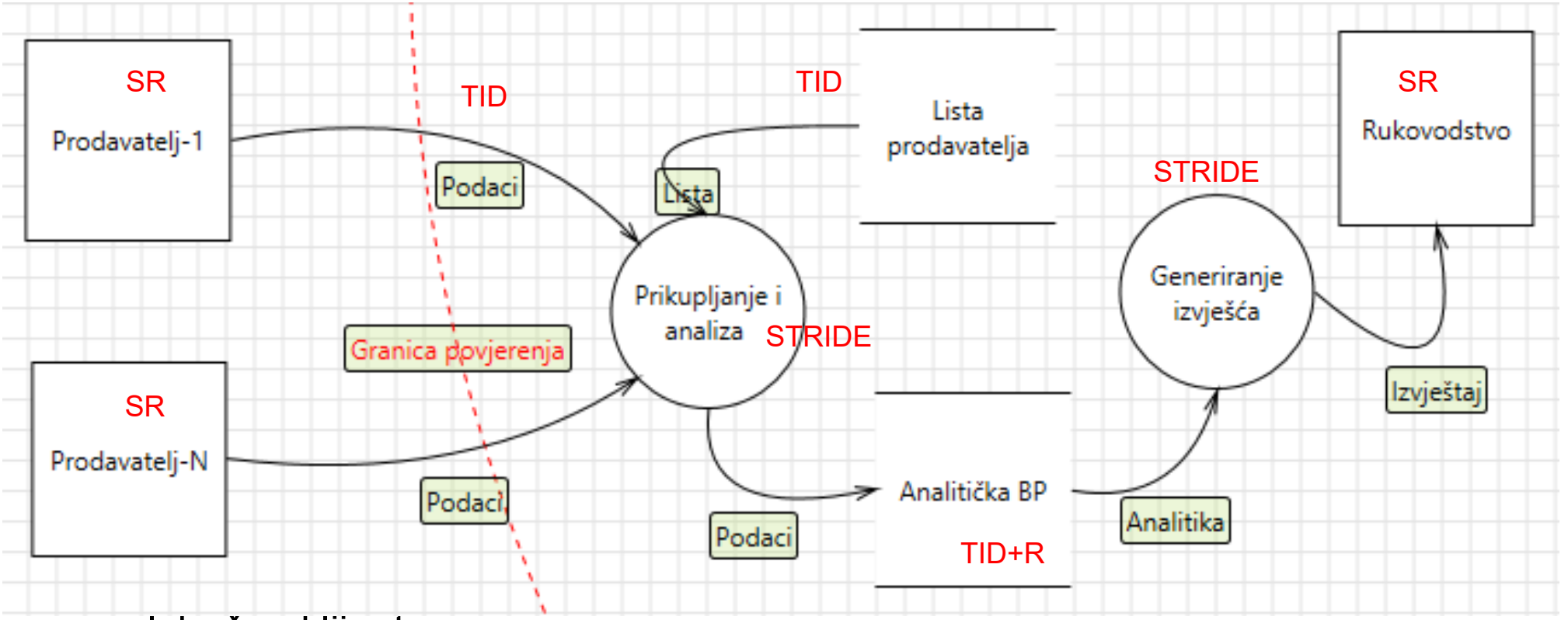
Improved diagram – server side analysis



- Client kicked out
- Added *Generation*, consequently i *Management*
- The list became a repository
- Integrated processing

- OYSTERS?

Poboljšani dijagram – analiza poslužiteljske strane



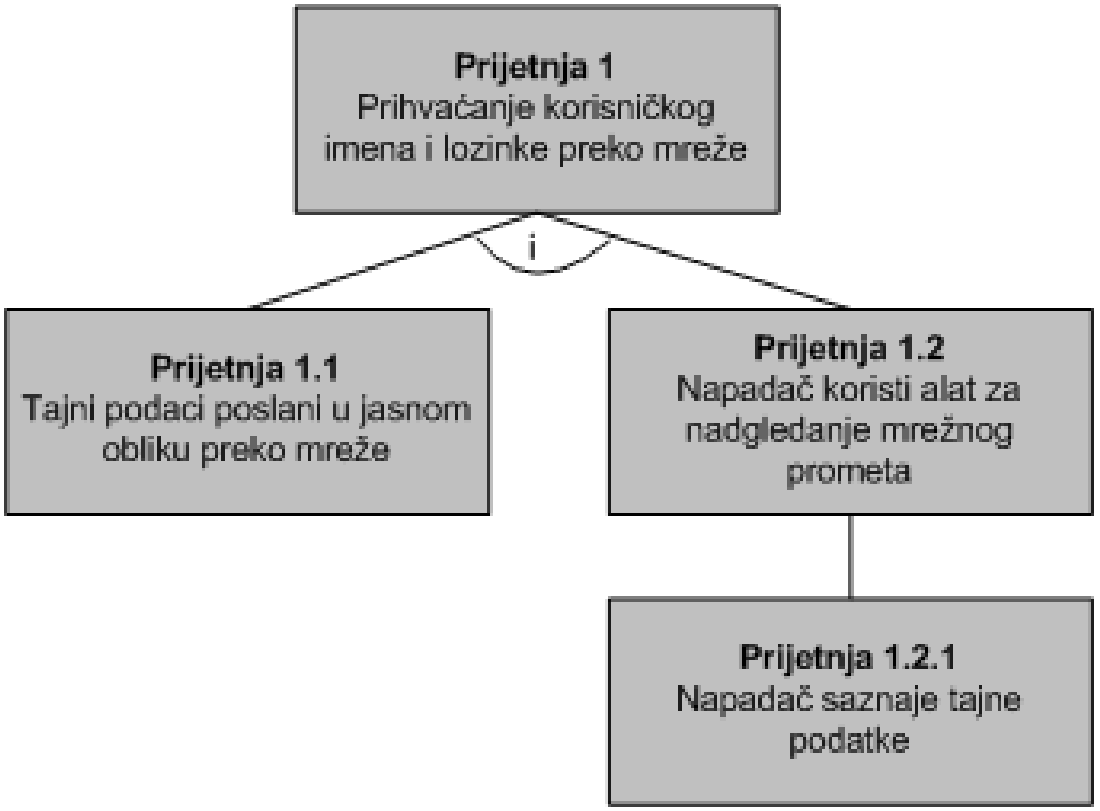
- Izbačen klijent
- Dodano *Generiranje*, posljedično i *Rukovodstvo*
- Lista postala spremište
- Integrirana obrada

- STRIDE ?

Threat trees

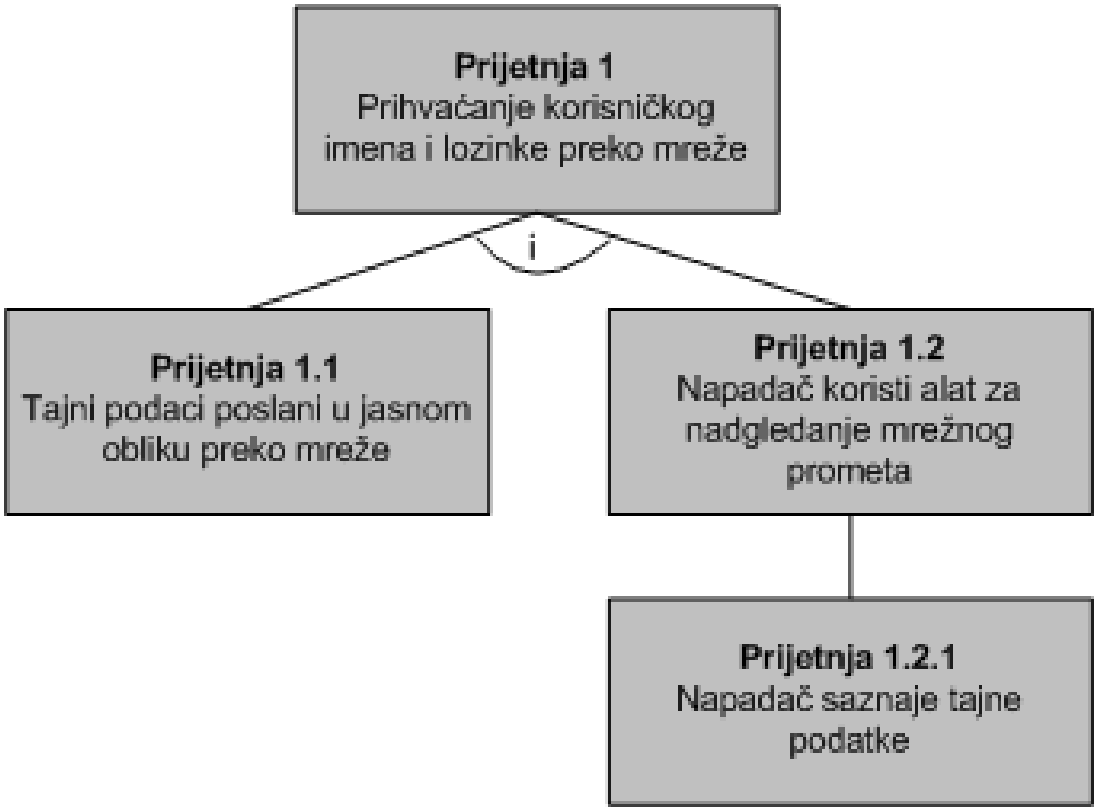
- For each component obtained
  by decomposition
  - possible threats are determined
  - the way in which threats are reflected
    on the system is determined

- Example
  - the root is a threat
  - children represent the steps
    an attacker must take to
    execute a threat



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



Threat Trees (continued)

- Alternative view

1.0 Threat 1 :
Accepting username and password over the network
1.1 Confidential data sent in clear form over the network **AND**
1.2 The attacker uses a tool to monitor network traffic
1.2.1 The attacker learns secret information

- Using STRIDE over threat trees is easy
  - for each part of the system, it is checked whether it is subject to one of the STRIDE categories
  - for example, can an attacker deny the operation of a process, view data, etc.
- What threats does the above example illustrate?

Stabla prijetnji (nastavak)

- Alternativni prikaz

```
1.0  Prijetnja 1 :
     Prihvaćanje 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 podložan nekoj od STRIDE kategorija
  - npr. može li napadač uskratiti rad procesa, vidjeti podatak itd.
- Koje prijetnje ilustrira gornji primjer ?

Attack patterns

- General representation of common attacks
  - defines the aim, conditions, technique and result of the attack
- The emphasis is on attack technique (with STRIDE on the attacker's goals)
- Example form:

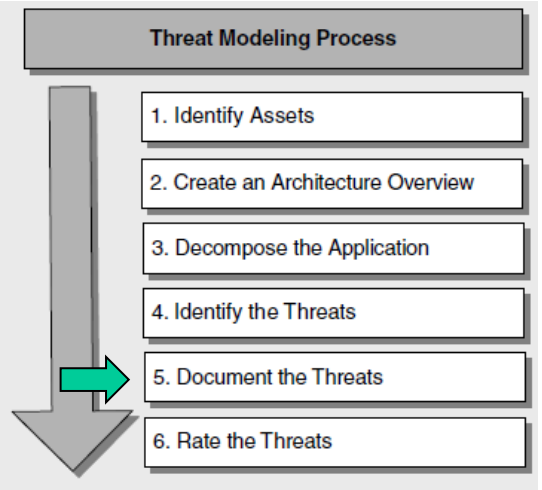
Pattern	Code injection attacks
Attack goals	Command or code execution
Required conditions	Weak input validation Code from the attacker has sufficient privileges on the server.
Attack technique	1. Identify program on target system with an input validation vulnerability. 2. Create code to inject and run 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.

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.

Step 5 - Documenting the threats

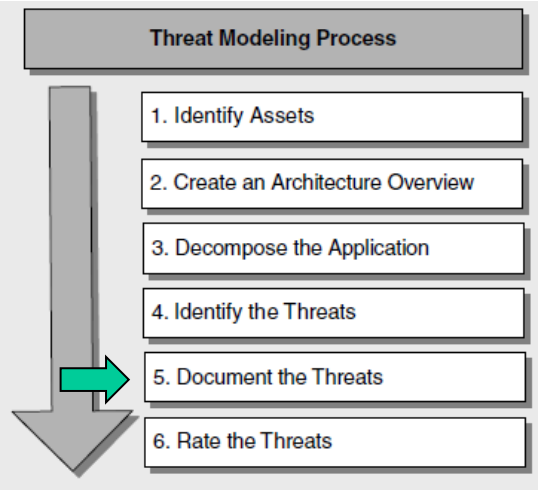


- Threat log template
  - Be sure to fill in the description and goal
  - the risk is left for the next step
  - other attributes may be optional
- Examples

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 5 - Dokumentiranje prijetnji

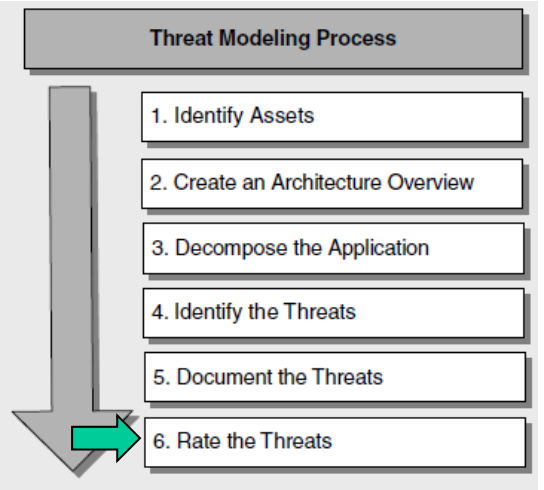


- Predložak za evidenciju prijetnji
  - svakako se popunjavaju opis i cilj
  - rizik se ostavlja za naredni korak
  - ostali atributi mogu biti opcionalni
- Primjeri

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.

Step 6 - Threat Ranking

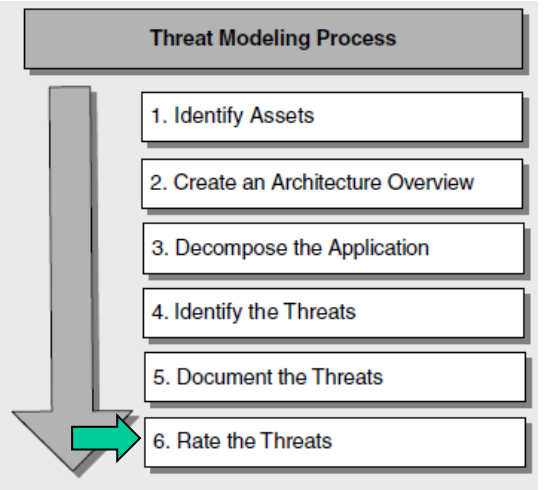


- Ranking - determination of importance (rate the threats)
  - techniques are often used to determine risk
  - **risk**=probability of event * potential damage
  - **probability**eg in the range 1-10
  - **too bad**eg in the range 1-10
  - risk in the range 1-100
  - distribution into three groups (high, medium, low) representing priorities

- An issue:
  - team members cannot agree on values

- Solution:
  - DREAD model, risk ranking for a given threat

Korak 6 - Rangiranje prijetnji



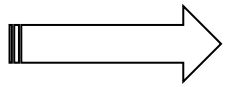
- Rangiranje – određivanje važnosti (rate the threats)
  - često se koriste tehnike za određivanje rizika
  - **rizik** = vjerojatnost događaja * potencijalna šteta
  - **vjerojatnost** npr. u rasponu 1-10
  - **šteta** npr. u rasponu 1-10
  - rizik u rasponu 1-100
  - raspodjela u tri grupe (visok,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

DREAD (risk assessment model)

- DREAD– classification of computer threats
  - ***D****harm potential*–possible damage, amount of damage if the attack is successful
  - ***R****eproducibility*–reproducibility, how easy it is to repeat the attack
  - ***E****xploitability*–exploitability, effort and knowledge required for a successful attack
  - ***AND****affected users*–affected users, possibly by a successful attack, percentage
  - ***D****iscoverability*–detectability, difficult to measure

- Assessment of each threat according to the specified parameters
  - individual value from 1 to 10 (least bad - worst)
  - total risk - average of 5 individual DREAD values

- Better – (simple)grading scheme



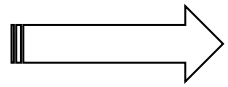
  - Low, medium, high – mapped to interval 1 to 3

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 uspjelim 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



Example of a simple grading scheme

	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

	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.

Example of a simple grading scheme (continued)

- The values (1-3) for the given threat are added up
  - the score is in the range 5-15
  - risk is assigned, eg 5-7 low, 8-11 medium, 12-15 high

- For example for two documented threats from the beginning of the story

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

- ... threat documentation templates are updated (step 5)

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)

Threat resolution (after modeling)

- Fix (reduction, risk reduction)
  - reduce the consequence
- Do nothing (accept the risk)
  - bad, if the problem is real, it will happen at some point and will have to be fixed
- Notify the user and leave the decision on use to him (transfer)
  - problematic
  - many users do not know what the right decision is, and the notifications are incomprehensible
  - use only if there is a great need to use the (risky) service
- Removal of the risky property (avoidance)
  - when the problem cannot be corrected immediately (e.g. there is no time, etc.),
  - to be corrected in the next version

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

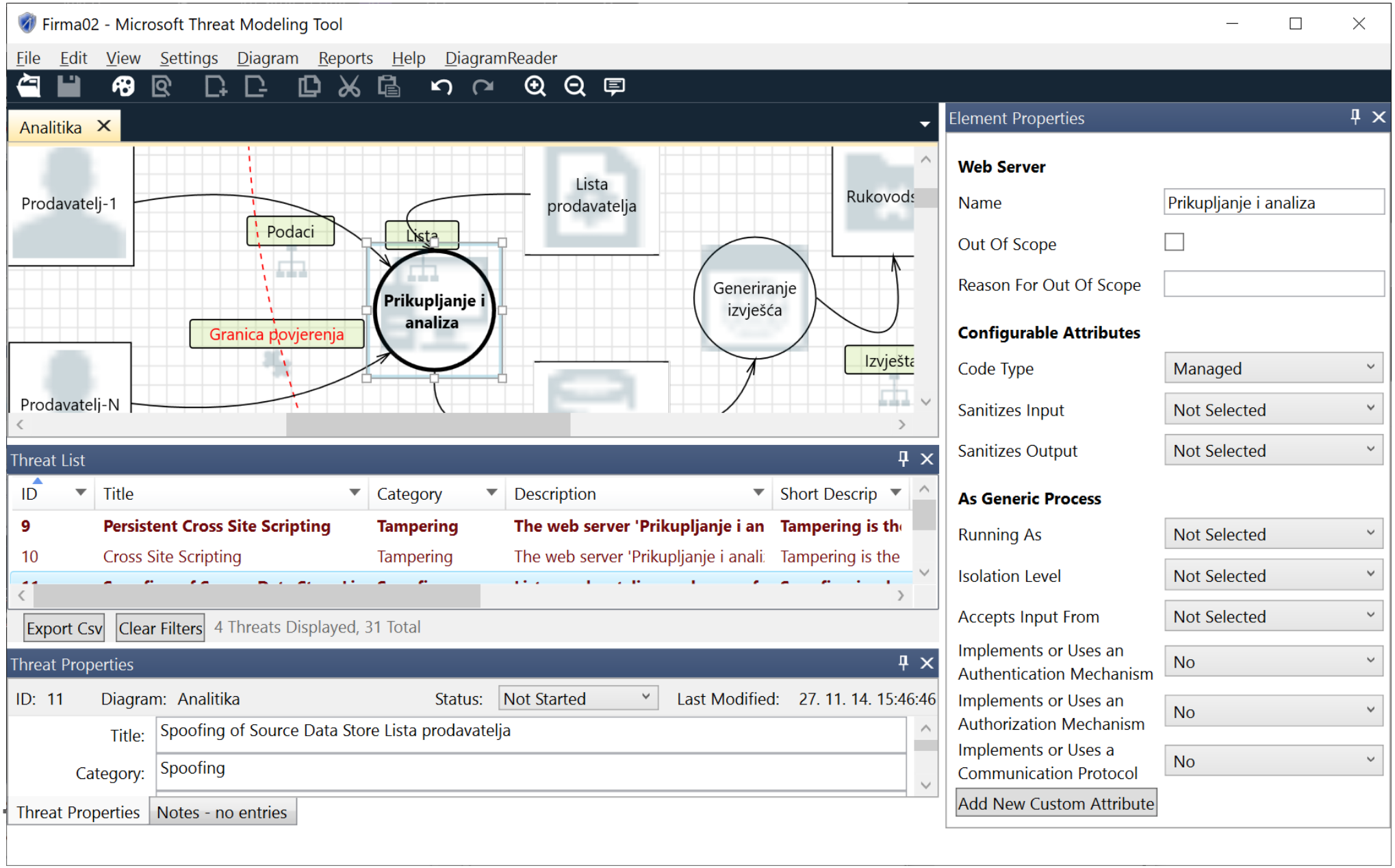
Threats and countermeasures - another time

Threats	Countermeasures (security techniques)
Deception	Appropriate authentication Protection of private data Private data must not be stored in a clear form
Modification of data	Appropriate authorization Use of compression functions Use of digital signatures
Denial	Use of digital signatures
Disclosure of information	Appropriate authorization Private data must not be stored in a clear form Provide a communication channel
Denial of Service	Validate and filter input data
Increase in authority	Grant only the necessary powers

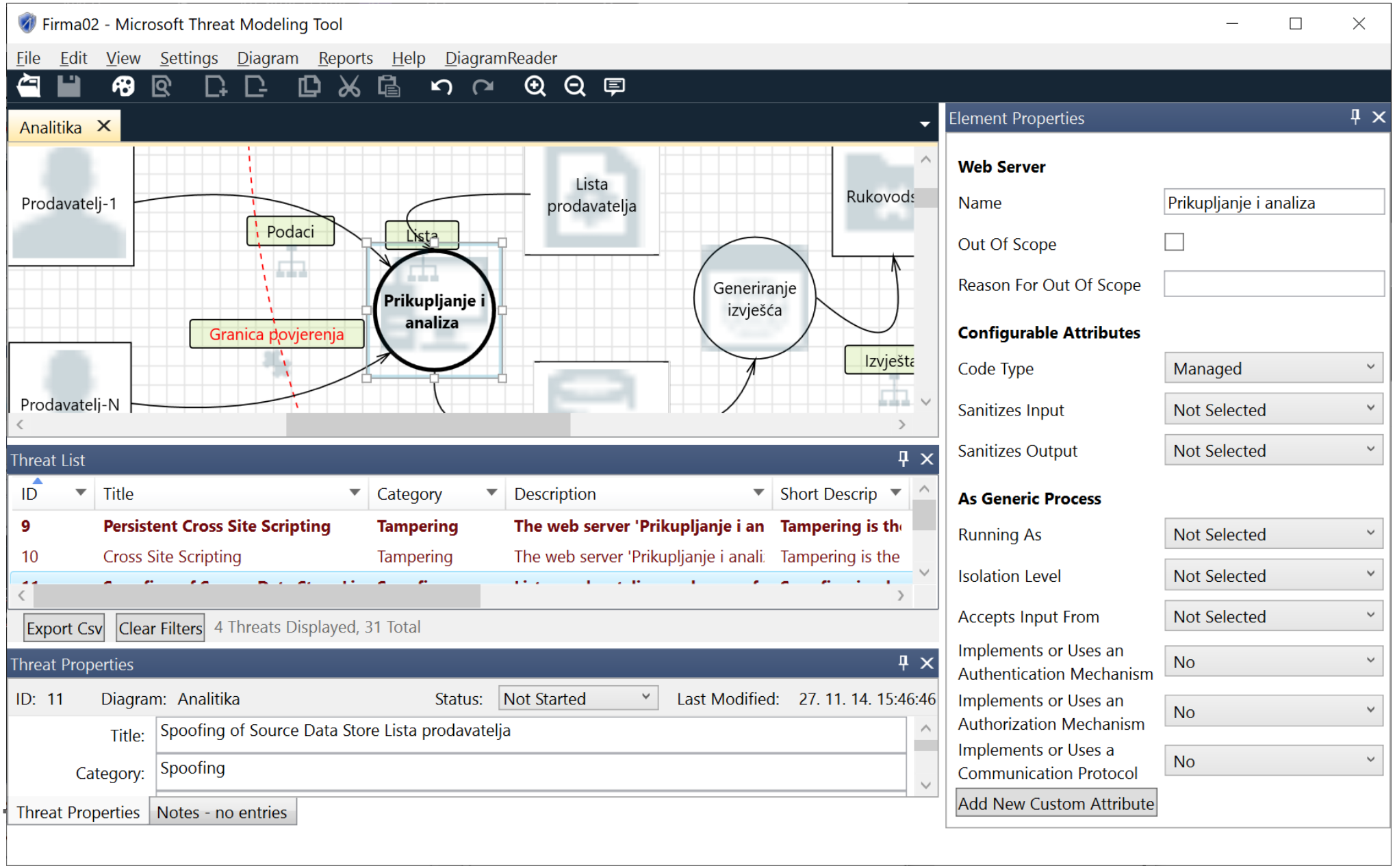
Prijetnje i protumjere – drugi put

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

Example, Microsoft Threat Modeling Tool



Primjer, Microsoft Threat Modeling Tool



Reducing the attack surface

Attack Surface Reduction

Smanjenje površine napada

Attack Surface Reduction

attack surface

- Attack surface- a collection of program product entry points
  - User interfaces, web services, direct access to BP, network channels, API, …
  - resource communication channels = attack vectors
  - attackability measure

- Larger attack surface = more protection work = more potential damage

- Surface area determines attack risk – a measure of potential access and impact
  - Experience shows that certain vectors are more risky
  - For example privileged (*root*) services, files with full access (*rwxrwxrwx*), scripts (JScript, VBScript) and active controls (ActiveX)

Površina napada (attack surface)

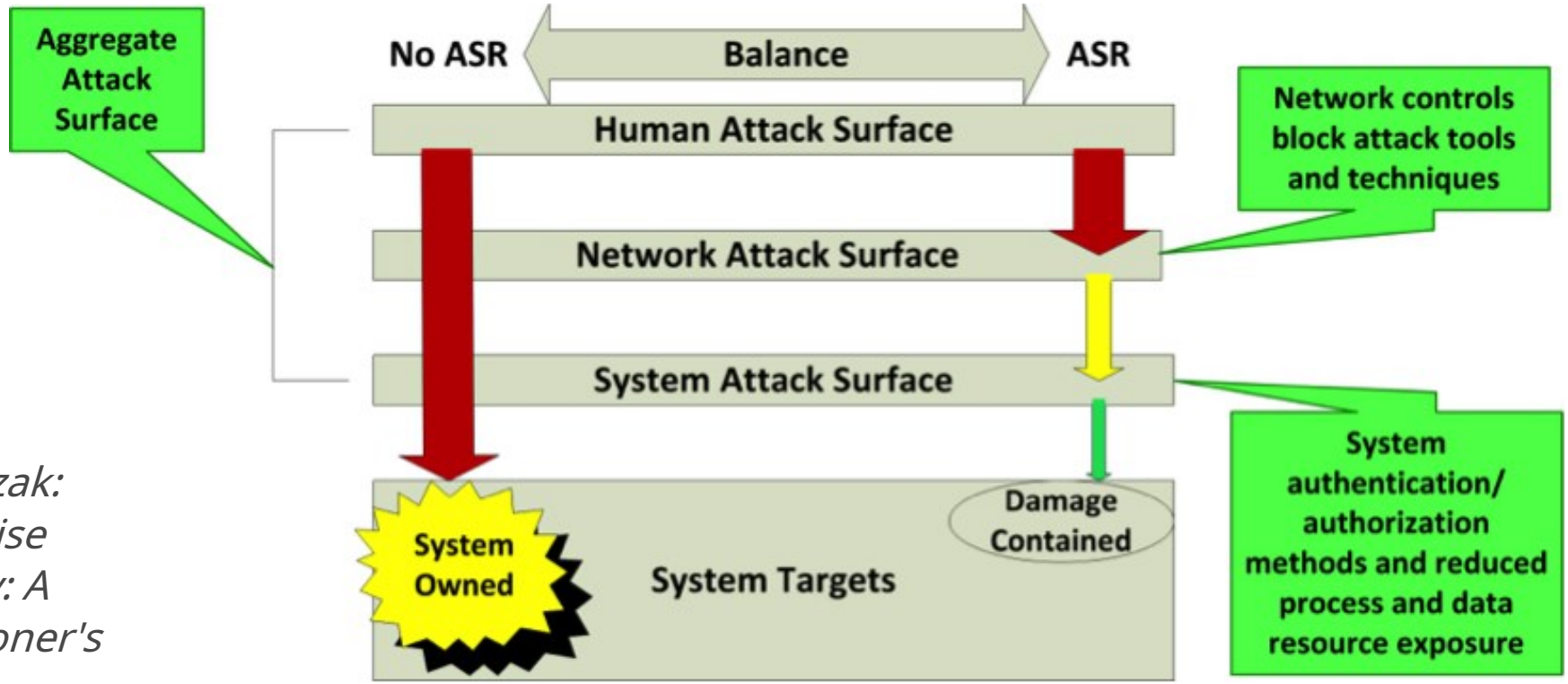
- Površina napada - kolekcija ulaznih točaka programskog proizvoda
  - Korisnička sučelja, web servisi, izravan pristup BP, mrežni kanali, API, …
  - kanali za komunikaciju s resursima = vektori napada
  - mjera "napadljivosti" (attackability)

- 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)

Combined attack surface model

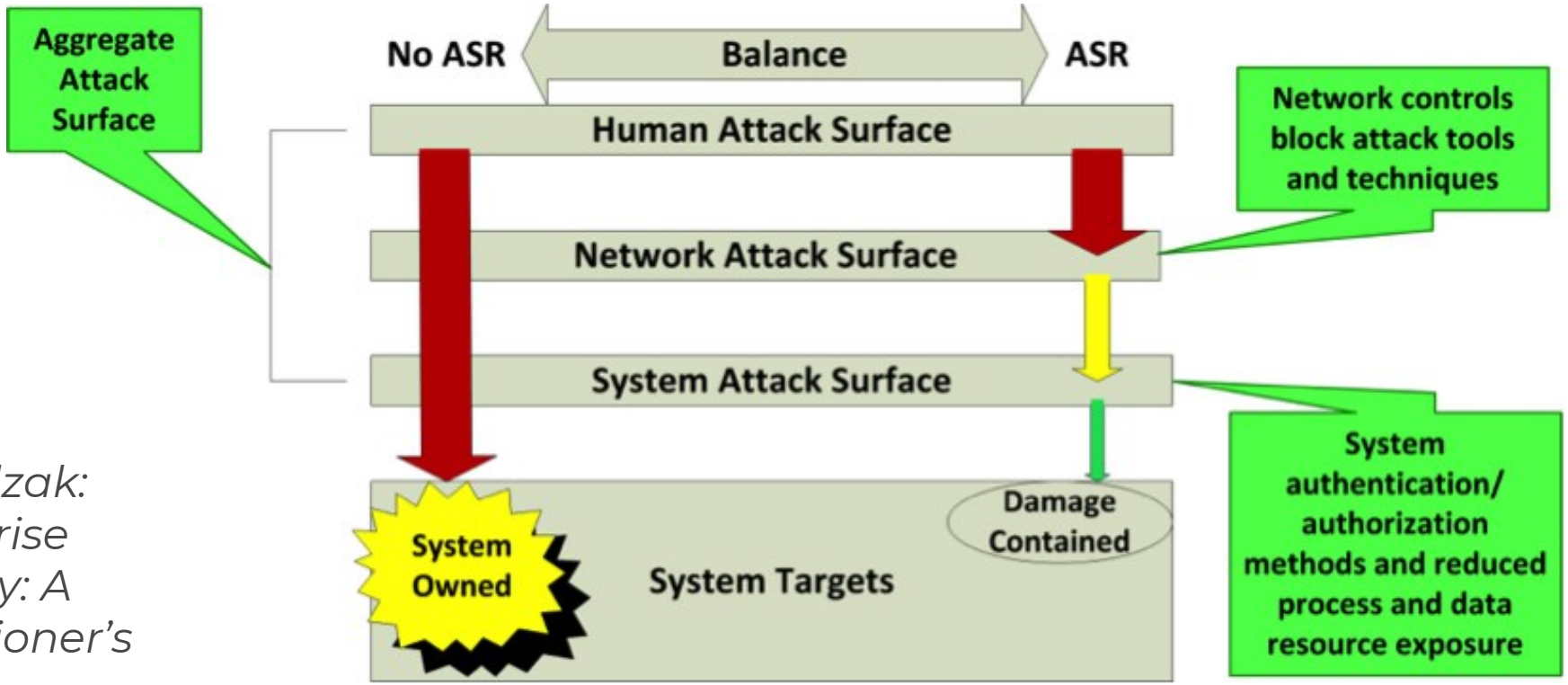
- access controls reduce
  - possibility to reach the system
  - the number of elements that are visible or usable



Tom Olzak: Enterprise Security: A practitioner's guide

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

Attack surface reduction

- Main objectives
  - Reducing the amount of code that is executed "by default"
  - Reducing the amount of code that can be accessed by untrusted users, "on sight"
  - Closing access points (access points, entry points) - doors that are easily opened/used
  - Limiting the damage in case the access point is exploited

- The ultimate goal – repelling future attacks

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

A common software security metric

- Code level - bug count
  - It does not count bugs that have not (yet) been found
  - All bugs are of equal severity, although some are easier to exploit
  - Some bugs can cause more damage than others
- Product/system level
  - Counting how many times the system version is mentioned in CERT, MITER CVE, … bulletins
    - Computer emergency response teams (CERT),http://www.cert.hr/
    - Common Vulnerabilities and Exposures (CVE) dictionary,https://cve.mitre.org/
  - Ignores specific configurations
    - Installed patches
    - Defaults on or off
    - Work in*admin*fashion

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

Attack surface measurement

- Measuring avenues of attack
  - "more likely to be attacked" features
- Measuring relative safety
  - Delta measurement – differences between versions of the same product (eg v1 vs v2)
  - Unusable for comparing different applications
- Procedure
  - Baseline + weekly measurements
  - Determining the minimum area at the beginning
  - If the area increases – determine how to reduce it

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

Attack surface and access points

- Example of comparison of measurements of different versions

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

Površina napada i pristupne točke

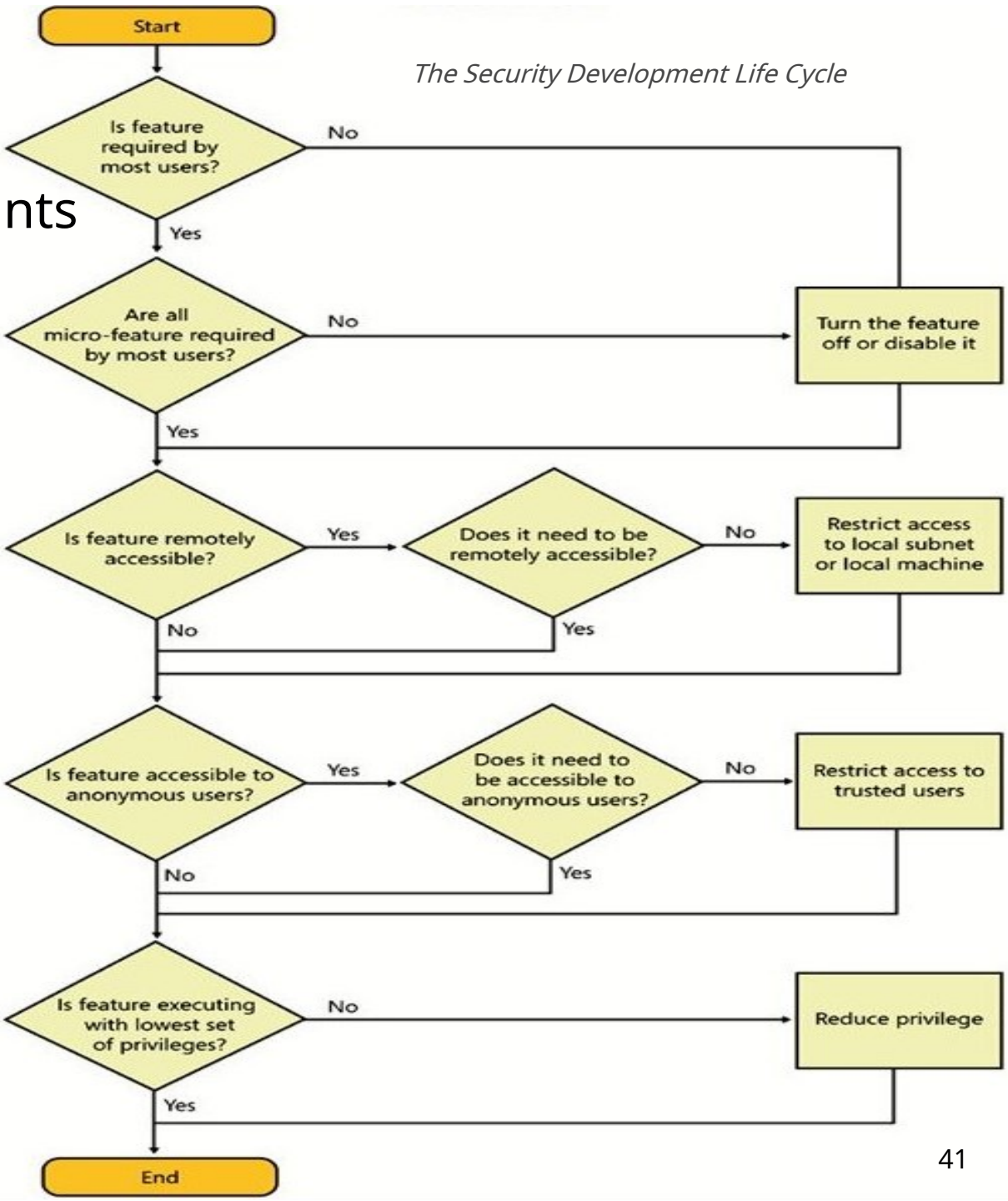
- Primjer usporedbe mjerenja različitih verzija

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

ASR process

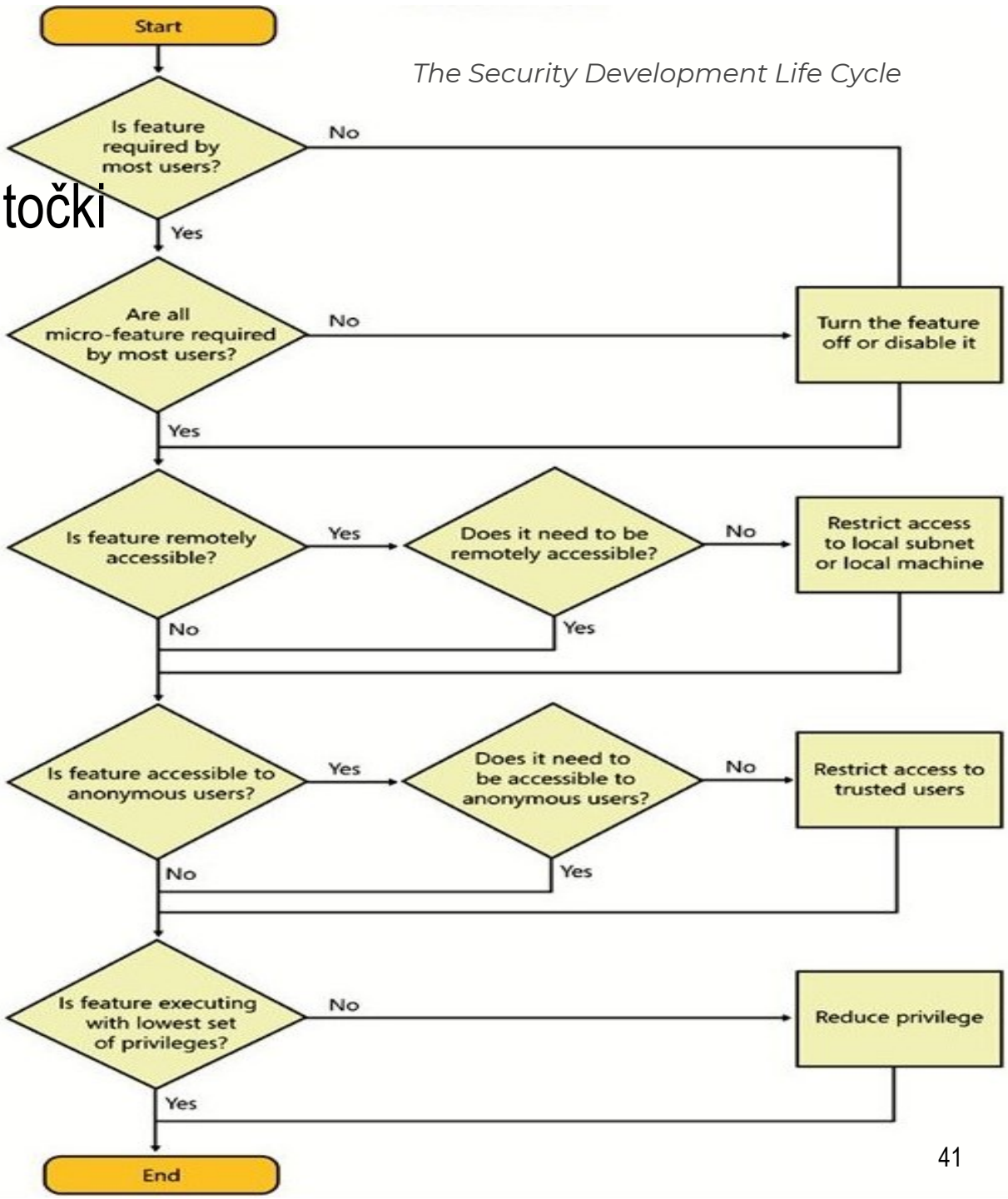
- Establishment of access points
  - network, files, …
- Ranking points
  - according to the user
  - authenticated – anonymous
  - *admin–user*
  - online -*local*
- Adjustment



The Security Development Life Cycle

Proces ASR

- 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	Lower Attack Surface
Executing by default	Off by default
Open socket	Closed socket
Anonymous access	Authenticated access
Constantly on	Intermittently on
Admin access	User access
Internet access	Local subnet access
SYSTEM	Not SYSTEM!
Uniform defaults	User-chosen settings
Large code	Small code
Weak ACLs	Strong ACLs

It's Not *Just* About Turning Stuff Off!

Higher Attack Surface	Lower Attack Surface
Executing by default	Off by default
Open socket	Closed socket
Anonymous access	Authenticated access
Constantly on	Intermittently on
Admin access	User access
Internet access	Local subnet access
SYSTEM	Not SYSTEM!
Uniform defaults	User-chosen settings
Large code	Small code
Weak ACLs	Strong ACLs

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

Best practices

- Reduction of the code being executed *by default*
  - Turn off an option that is not used by at least 80% of users
  - A stopped service cannot be attacked
    - dynamic web content should be optional - it affects only those who initiate it
  - The solution is not just to switch off
    - restriction of access to running code

- Reducing access by untrusted users
  - Restricting access to a local network or IP address range
  - Authentication

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

Best practices (continued)

- Privilege reduction to limit potential damage
  - Revoking privileges that are not absolutely necessary
  - Running code in the security box (sandbox running code)
    - by limiting the permissions to which the code is entitled,
    - *Java.Class.SecurityManager*, or .NET *System.Security.SecurityManager*
  - If necessary, pick up permits - temporarily, as short as possible
  - Paying attention to confidence limits (threat modeling procedure)
    - particularly *roads* of anonymous threats (anonymous threat paths) → authorization where necessary
  - Do not run services as SYSTEM (daemons as *root*) or with administrator rights until other options are exhausted!
- Example:
  - *Backup Operator account* – reads all files regardless of their ACL
  - SYSTEM works the same but also *restore*, *debug*, "*act as part of OS*" that's it *admin*

Najbolje prakse (nastavak)

- Redukcija privilegija radi ograničavanja potencijalne štete
  - Ukidanje privilegija koje nisu prijeko potrebne
  - Pokretanje koda u sigurnosnom okviru (sandbox running code)
    - ograničenjem dozvola na koje kod ima pravo,
    - *Java.Class.SecurityManager*, ili .NET *System.Security.SecurityManager*
  - Po potrebi podići dozvole – privremeno, što kraće
  - Pripaziti na granice povjerenja (postupak modeliranja prijetnji)
    - naročito *putove* anonimnih prijetnji (anonymous threat paths) → autorizacija gdje treba
  - Ne pokretati servise kao SYSTEM (demone kao *root*) ili s administratorskim pravima dok ne budu iscrpljene druge mogućnosti!
- Primjer:
  - *Backup Operator account* – čita sve datoteke bez obzira na njihov ACL
  - SYSTEM radi isto ali i *restore*, *debug*, "*act as part of OS*" i k tomu je *admin*

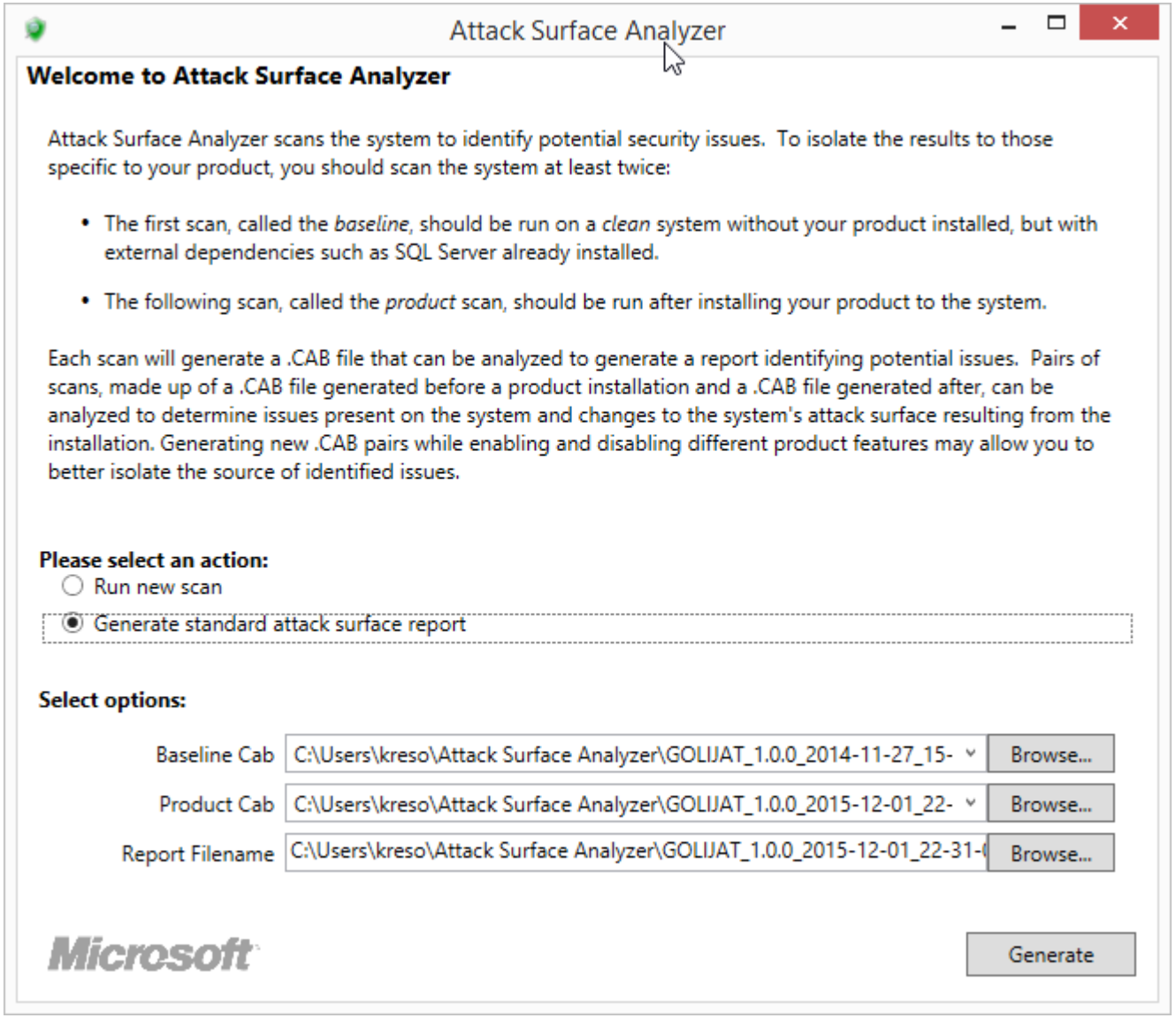
Best practices (rest)

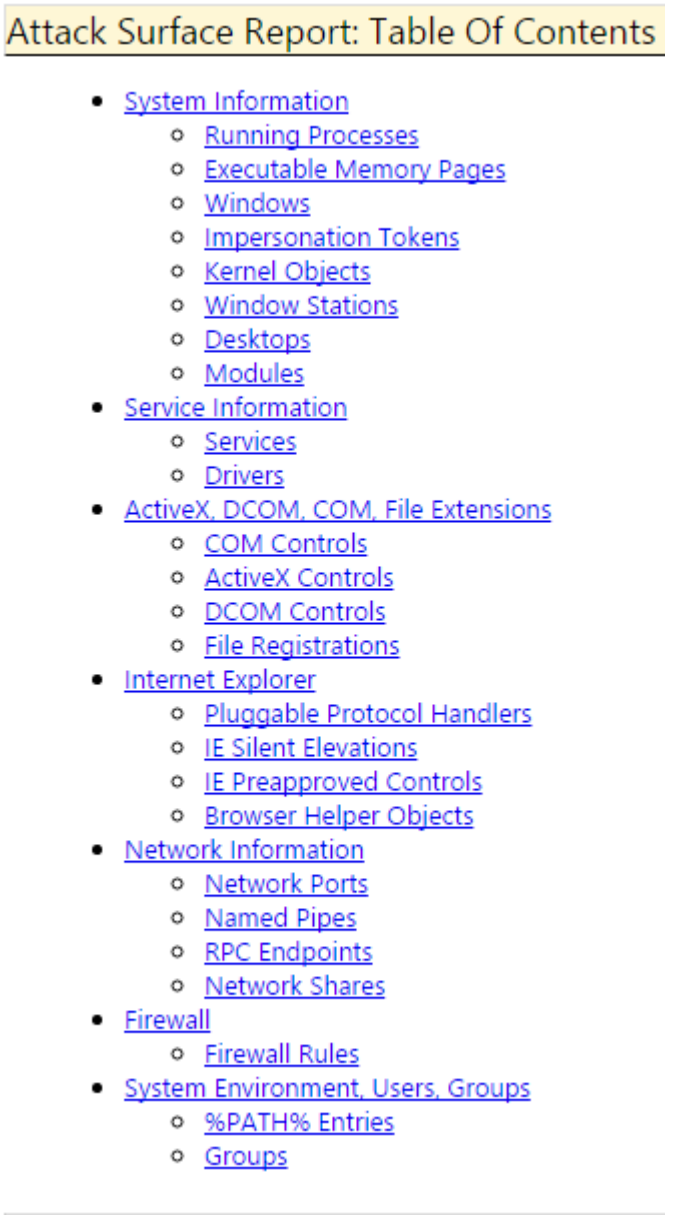
- Defining the attack surface during design/engineering
  - sketch the attack surface and establish
    - protocols
    - endpoints that need authentication and authorization
    - off-by-default options – autostart (eg Windows \ Services, starter.exe)
    - reusable components (ActiveX, COM, .NET assemblies, etc.)
    - process identities
    - installed user accounts
- Other procedures
  - Threat modeling
  - Attack surface overview – analyzer: base + differences
  - Design review – looking for threats and mitigation opportunities
  - Code Review - Defensive Programming and Secure Coding

Najbolje prakse (ostatak)

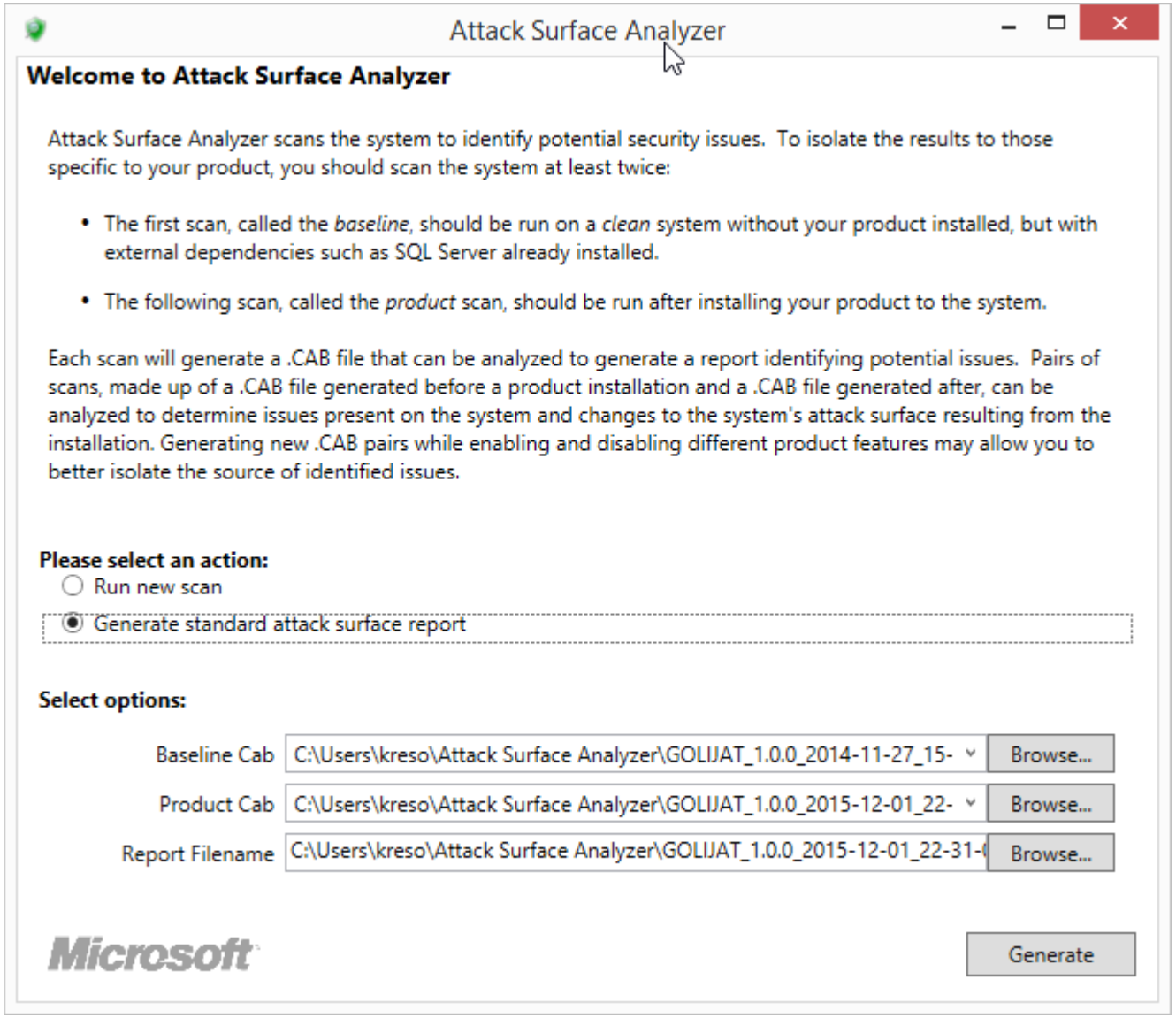
- Definiranje površine napada tijekom dizajna/projektiranja
  - skicirati površinu napada i ustanoviti
    - protokole
    - krajnje točke koje trebaju autentifikaciju i autorizaciju
    - isključene (off-by-default) mogućnosti – autostart (pr. Windows \ Services, starter.exe)
    - ponovno iskoristive 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

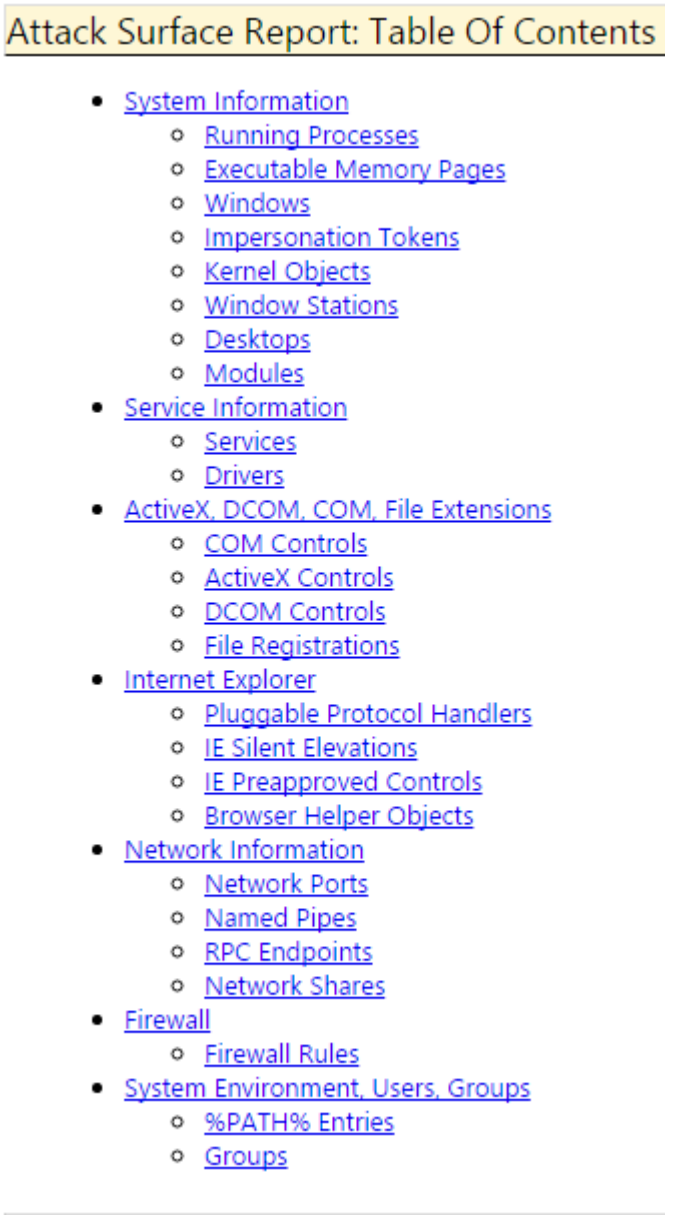
Example: Attack Surface Analyzer





Primjer: Attack Surface Analyzer





References

- Procedures
  - A systematic review of security requirements engineering, Mellado et.a., 2010
  - Threat Modeling with STRIDE

- Tools
  - Attack Surface Analyzer
  - Microsoft Threat Modeling Tool
  - Top 10 Threat Modeling Tools in 2021

Reference

- Postupci
  - A systematic review of security requirements engineering, Mellado et.a., 2010
  - Threat Modeling with STRIDE
- Alati
  - Attack Surface Analyzer
  - Microsoft Threat Modeling Tool
  - Top 10 Threat Modeling Tools in 2021