Uvod u načela sigurnog dizajna

KRISTIAN SKRAČIĆ

Sadržaj

Što je sigurnost i zašto nam je bitna

Što su načela sigurnog dizajna i zašto su korisna

Pregled načela sigurnog dizajna
- *10 načela korisnih u praksi*

Pregled stvarnih primjera

Gdje/kako naći pomoć

Sigurnost

Svima je poznato da je sigurnost važna- ali zašto?

- Zaštita od zlobe, pogrešaka i neugodnosti
- Krađe, prijevare, uništavanja…

Sigurnost se može gledati i kroz upravljanje rizicima

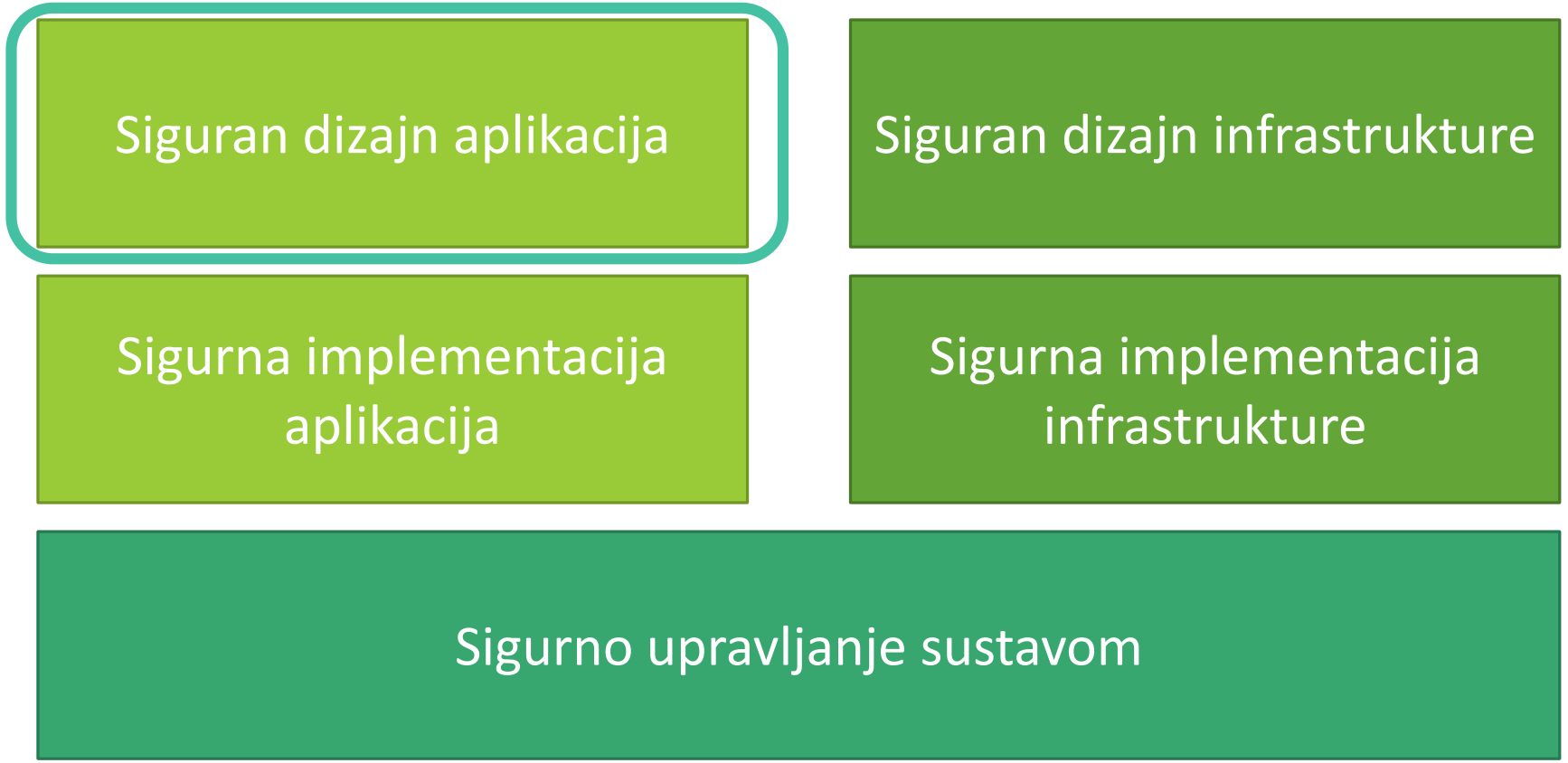
- Gubitak vremena, novca, ugleda, prednosti na tržištu
- Model osiguranja – balansiranje troškova osiguranja i rizik gubitka

Što kada se dogodi greška?

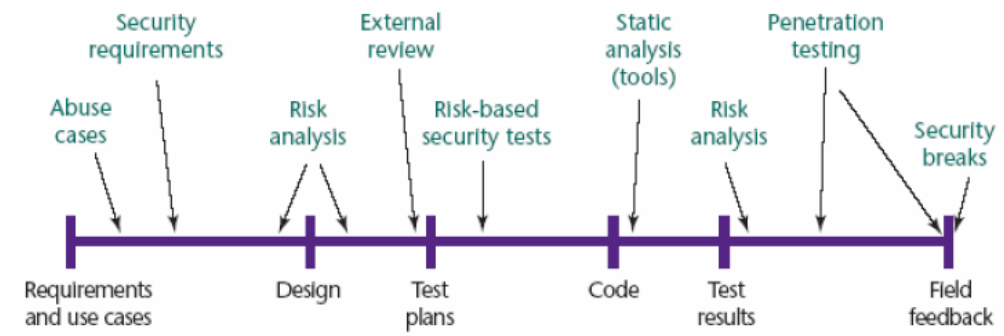
Equifax

- „*consumer credit reporting agency*”
  - Jedna od najvećih agencija tog tipa u S.A.D.
- Doživjela kompromitaciju 29. Srpnja 2017
  - Napadači dobili pristup
    - **Osobnim podatcima**: datum rođenja, adrese prebivališta, informacije sa vozačkih dozvola, te osobne identifikacijske brojeve (Social Security Number)
    - Podatcima od **143 milijuna** evidentiranih korisnika
    - Podatcima o **kreditnim karticama** za preko 209.000 korisnika
  - **Iskorištena ranjivost** u Apache Struts biblioteci (koja je korištena za izradu Equifax sustava)
    - CVE-2017-5638
      - Omogućuje izvršavanje naredbi zbog neispravnog pariranja *Content-Type* polja u HTTP zaglavlju

Aspekti sigurnosti u praksi



Životni ciklus sigurnog razvoja aplikacije



Siguran dizajn aplikacija

Što to znači ?

A. Programeri ne smiju raditi pogreške (treba bolje testirati kod)

B. Komponente od kojih su aplikacije građene moraju biti testirane i pažljivo odabrane

C. Ne smijemo držati osjetljive podatke u aplikacijama

D. Redovito raditi sigurnosno testiranje sustava kako bi se što prije uočile ranjivosti

Aplikacije je sigurna

- Ukoliko se ponaša na očekivani način

Ipak …

Adult Friend Finder

- Stranica za dating…
- Doživjela kompromitaciju u rujnu 2016
  - Napadači pronašli ***Local File Inclusion vulnerability (LFI)*** ranjivost
    - Omogućuje dohvat proizvoljnih datoteka sa sustava
    - Napadač dohvatili
      - **/etc/passwd** datoteku
      - **SQL shemu** sustava generiranu mjesec dana ranije
        - Shema otkrila:
          - Imena tablica, internih IP adresa,…
          - **Lozinku od 6 znakova za pristup svim dijelovima sustava**
    - Napadač dohvatio povijesne podatke unazad čak do 20 godina za nekoliko sustava
      - *Adult Friend Finder, Penthouse.com, Cams.com, iCams.com and Stripshow.com*
    - U ukradenim podatcima su bili korisnička imena i lozinke korisnika servisa
      - Korišten zastarjeli **SHA-1** algoritam
        - Konačna analiza kompromitacija je završila u Studenom 2016 – do kad je 99% lozinki bilo razbijeno

Načela sigurnog dizajna

Što je „načelo” ?

- …temeljna istina ili prijedlog koji služi kao temelj vjere ili djelovanja…

Određujemo načelo sigurnog dizajna kao

- Deklarativnu izjavu s namjerom **vođenja sigurnosnih odluka** u dizajnu kako bi se postigli ciljevi sustava

Postoji više načela sigurnog dizajna

- U nastavku su odabrani osnovni
  - Više njih može se pronaći u literaturi, Internetu, praksi (linkovi dani na kraju prezentacije)

Načela sigurnog dizajna

(Principles of Secure Design)

1. Najmanji razina pristupa (engl. Least Privilege)
2. Razdvajanje odgovornost (engl. Separate responsibilities)
3. Ograničiti povjerenje (engl. Trust cautiously)
4. Ekonomičnosti rješenja (engl. Economy of Mechanism)
5. Revizija ključnih događaja (engl. Audit sensitive events)
6. Primjena sigurnih *default* vrijednosti (engl. Secure and Fail Safe Defaults)
7. Izbjegavat zabitnost (engl. Avoiding Obscurity)
8. Dubinska obrana (engl. Defense in depth)
9. Korištenje poznatih i provjerenih metoda zaštite (engl. Use proven and secure security methods)
10. Osigurati najslabiju kariku (engl. Secure the weakest link)

Načelo najmanje razina pristupa 1/2

Principle of Least Privilege

Zašto	Viša razina pristupa omogućuje zlonamjerni ili slučajni pristup zaštićenim resursima
Načelo	Koristiti najmanju razinu pristupa koja je potrebna za obavljanje tražene funkcionalnosti – ne više od toga
Nedostatci	Složenost održavanja, niska učinkovitost/produktivnost
Primjer	Blagajnik ne može pisati čekove. Pokretanje mrežnih procesa s posebnim korisničkim računima koji imaju ovlasti isključivo na resurse koje koristi proces

Načelo razdvajanja odgovornosti

Principle of Separate Responsibilities

Zašto	Postići kontrolu i ustanoviti odgovornost, te ograničiti utjecaj uspješnih napada Učiniti napade manje „atraktivnim”
Načelo	Odvojiti odgovornosti i prava pristupa
Nedostatci	Viši troškovi razvoja i testiranja, veća operativna složenost, rješavanje problema postaje teže
Primjer	Administratori modula za naplatu nemaju pristup niti kontrolu nad modulom za upravljanje narudžbama

Načelo ograničenog povjerenja

Trust cautiously

Zašto	Mnogi sigurnosni problemi nastaju ubacivanjem zlonamjernih posrednika u komunikacijski kanal
Načelo	Pretpostaviti da su nepoznati entiteti nepovjerljivi Ustanoviti jadan proces za uspostavljanje povjerenja Validirati entitet s kojim se komunicira
Nedostatci	Visoka operativna složenost, duže vrijeme implementacije
Primjer	Ne prihvaćati konekcije od nepoznatih ili nepovjerljivih izvora Koristiti certifikate (ili druge vjerodajnice) za poznate/pouzdane korisnike

Načelo Ekonomičnosti rješenja 1/2

Principle of Economy of Mechanism

- *Or simplest solution possible*

Zašto	Sigurnost zahtjeva razumijevanje onoga što je stvoreno Složena rješenja se rijetko kada razumiju – jednostavnost olakšava analizu
Načelo	Razvijati jednostavna rješenja Izbjegavati kompleksne scenarije, nepotrebne funkcionalnosti, implicitno ponašanje …
Nedostatci	Zahtjeva donošenje teških odluka vezane za odabir (ključnih) funkcionalnosti Dizajn jednostavnih rješenja (ili modeliranje složenih rješenja na jednostavan način) je kompleksan … ☺
Primjer	U nastavku…

Načelo Ekonomičnosti rješenja 2/3

Primjeri

- IPSEC specifikacija je složena što je rezultiralo
  - brojnim bugovima,
  - djelomičnim implementacijama i nekompatibilnosti između različitih proizvođača

- HTTP Request Smuggling (CWE-444)
  - Napadi krijumčarenja HTTP zahtjeva su izvedivi jer
    - ne postoje stroži uvjeti za rješavanje nepropisnih ili nedosljednih HTTP zaglavlja
  - To može dovesti do
    - nedosljednih implementacija u kojima proxy
    - vatrozid tumači isti tok podataka kao drugi skup zahtjeva od krajnjih točaka u tom toku

Načelo Ekonomičnosti rješenja 3/3

Sigurnosni mehanizmi trebaju biti što je moguće jednostavniji

Jednostavnije održavati i pronaći greške

- *KISS - "Keep it simple, stupid"*
- *YAGNI - "You Aren't Gonna Need It"*

"Keep it simple, as simple as possible, but not simpler." **– Albert Einstein**

"There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. And the other way is to make it so complicated that there are no obvious deficiencies." – **C.A.R. Hoare**

Načelo revizije ključnih događaja

Audit sensitive events

Zašto	Potrebno je stvoriti zapis aktivnosti na sustavu, spriječiti nepoželjne radnje i događaje, te uspostaviti mehanizam za nadzor sustava
Načelo	Pohraniti sve bitne aktivnosti na medij koji je otporan na zlonamjerne izmjene (engl. tamper resistant storage)
Nedostatci	Složena implementacija Potencijalno visoki troškovi opreme Niže performanse cjelokupnog sustava
Primjer	Snimanje svih promjena "osnovnih" poslovnih subjekata u ***append-only*** medij (user;IP;timestamp;entity;event)

Načelo Primjena sigurnih *default* vrijednosti 1/2

Principle of Secure and Fail Safe Defaults

Zašto	Standardne (*default*) lozinke, mrežni portovi i pravila su „otvorena vrata” za napadače Ispadi sustava često vode do primjene tih standardne (*default*) postavki
Načelo	Prisiliti izmjene osjetljivih parametara Razmišljati o ispadima/kvarovima prilikom projektiranja sustava
Nedostatci	Manjak komfora za krajnjeg korisnika
Primjer	Maknuti default lozinke na uređajima prilikom prve prijave

Načelo Primjena sigurnih *default* vrijednosti 2/2

Ako pristup nije eksplicitno odobren, pristup bi trebao biti odbijen

- Štoviše, ako sustav nije u mogućnosti dovršiti zadatak
  treba se **vratiti na početno stanje** !

Primjer:

- Običan korisnik ne može mijenjati mail-datoteke drugih korisnika
- Ako mail-aplikacija ne može isporučiti poruku, treba prijaviti grešku

Načelo izbjegavanja zabitnosti

Avoiding Obscurity

Zašto	Skrivanje je težak zadatak – netko uvijek može (slučajno ili namjerno) razotkriti nešto što je skriveno
Načelo	Pretpostaviti da se borimo protiv napadača sa savršenim znanjem iz svih tehničkih domena vezanih za informacijske sustave
Nedostatci	Zahtjeva značajne količine vremena i truda
Primjer	Pretpostaviti da će napadač pogoditi • „port knock” sekvencu za pristup sustavu • encoding lozinke • skrivene podatke u datotekama

Načelo Dubinske obrane 1/2

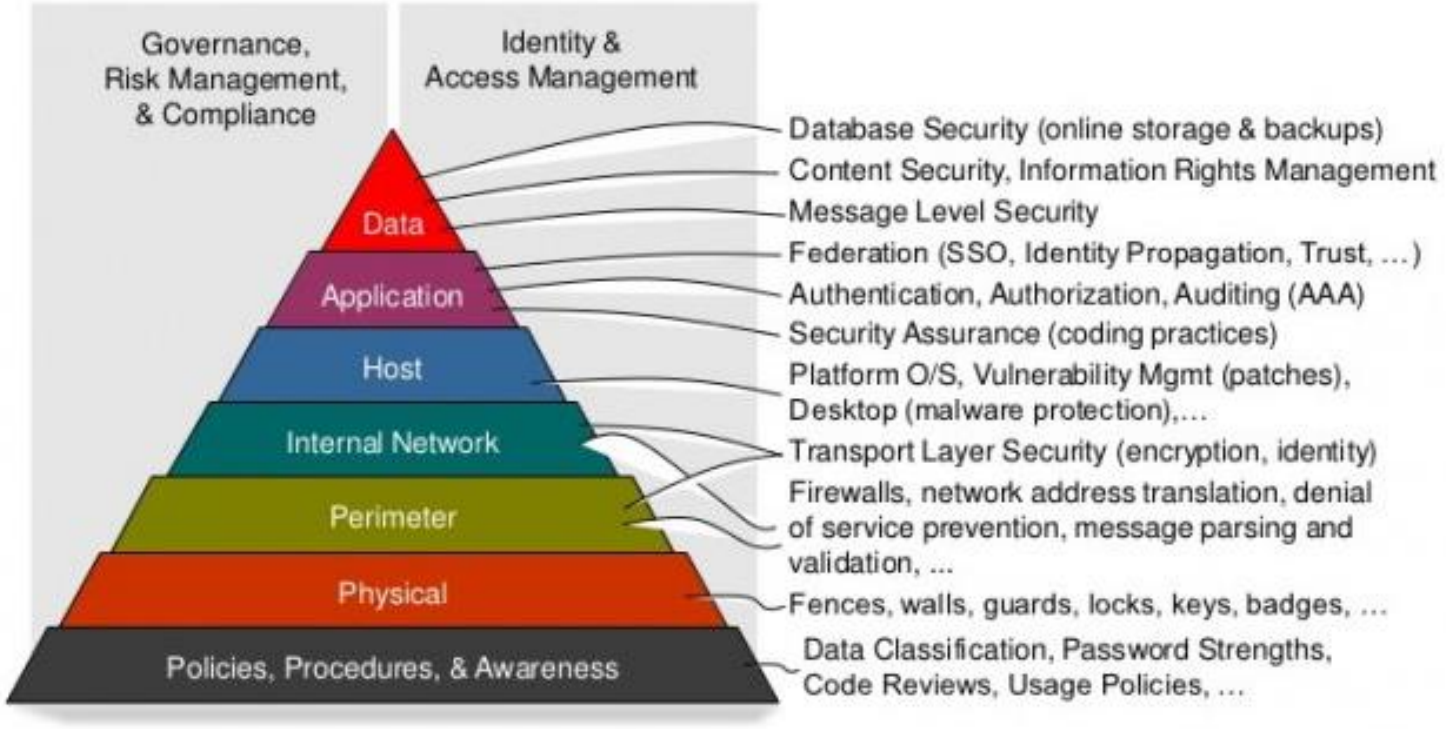
Principle of Defense in depth

Zašto	Sustavi ipak bivaju napadnuti, proboji sustava se događaju, pogreške se rade Bitno je minimizirati utjecaj takvih događaja na sustav
Načelo	Ne oslanjati se na sigurnost jedne centralne točke ispada/kompromitacije Potrebno je osigurati svaku razinu sustava, svaki element/čvor u sustavu, te zaustaviti propagaciju i utjecaj pogreške na ostatak sustava
Nedostatci	Izuzetno složeno za održavanje i projektiranje, ponekad može dovest ido sporijeg oporavka, mnogo redundantnih politika
Primjer	Odvojiti prava pristupa korisničkom sučelju, servisima, bazi podataka, operacijskom sustavu

Načelo Dubinske obrane 2/2

Što više linija obrane imamo

- to je bolja obrana
- osobito ako su dodatne linije različite prirode



Načelo Korištenje poznatih i provjerenih metoda zaštite

Use proven and secure security methods

Zašto	Stvaranje metoda/tehnologija za očuvanje sigurnosti pojedinih dijelova sustava je složen posao – posao za stručnjake koji se samo time bave
Načelo	Izbjegavati stvaranje vlastitih metoda zaštite, već koristiti postojeće metode
Nedostatci	Potrebno vrijeme za istraživanje/učenje postojećih metoda, te naučiti kada koju primijeniti
Primjer	Ne izmišljati vlastite alternative za kriptografske algoritme i protokole

Načelo osiguranja najslabije karike

Principle of Securing the weakest link

Zašto	Najslabija karika prva puca Obično je prva koju napadači odabiru
Načelo	Pronaći najslabiju kariku u sustavu i ojačati ju Ponoviti kroz više iteracija (tako izgraditi model ugroza)
Nedostatci	Zahtjeva značajan trud Često otkriva probleme u najnezgodnije vrijeme … ☺
Primjer	Problem čuvanja tajnosti osjetljivih podataka riješen korištenjem kriptografskih metoda prilikom razmjene podataka No osjetljivi podatci se čuvaju u nešifriranoj bazi i arhivama

Yahoo! data breach

Popularna tražilica Yahoo!

- Kompromitirana 2014. godine
- Napad objavljen tek 2016., a detalji poznati 2017. godine

Navodni tijek napada:

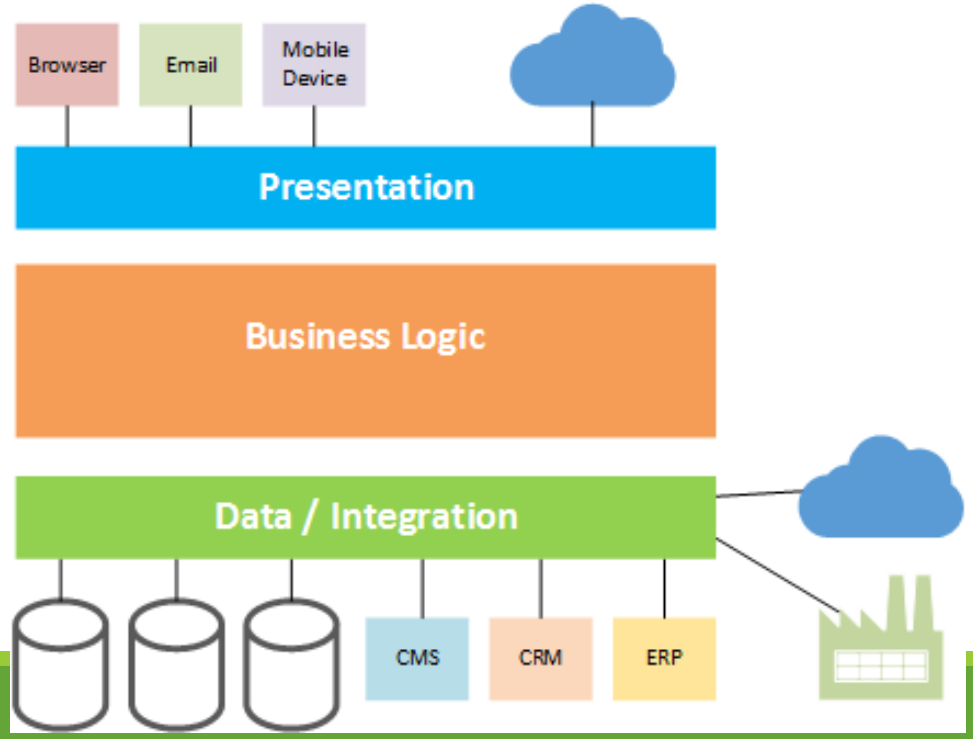
1. Serija **spear-phishing email** poruka kroz 2014., usmjerena na Yahoo! zaposlenike
2. Pronađena **baza podataka Yahoo korisnika**, te **Account Management Tool** (koji se koristi za uređivanje baze)
   - *Baza sadržala imena korisnika, telefonske brojeve i mailove, pitanja za promjenu lozinki, adrese za resetiranje lozinke, te određene kriptografske vrijednosti*
3. Instaliran **backdoor** za kasniji pristup sustavu
4. Ukradene kriptografske vrijednosti omogućili su napadaču da **pristupi korisničkim računima** (email i drugo) bez lozinke

Pogled arhitekture aplikacija s aspekta sigurnosti

Višeslojne aplikacije

Multi-tier / N-tier aplikacije

- Slojeviti način izrade aplikacija
- Arhitektonski oblikovni obrazac koji u svakom sloju sadrži određen skup poslovnih funkcionalnosti
- Najčešće se sastoji od:
  - Prezentacijski sloj (engl. Presentation layer)
  - Sloj poslovne logike (engl. Application layer)
  - Podatkovni sloj (engl. Data layer)
  - Mrežni sloj (engl. Network Layer)



Rizik: Više pogleda na isti sustav

Višeslojne aplikacije se rade u timovima, u kojima svaka grupa (ili pojedinac) imaju viziju za svoj dio sustava:

- Administratori sustava
- Administratori baza podataka
- Programeri
- QoA
- Analisti za sigurnost
- Marketing
- PR
- …

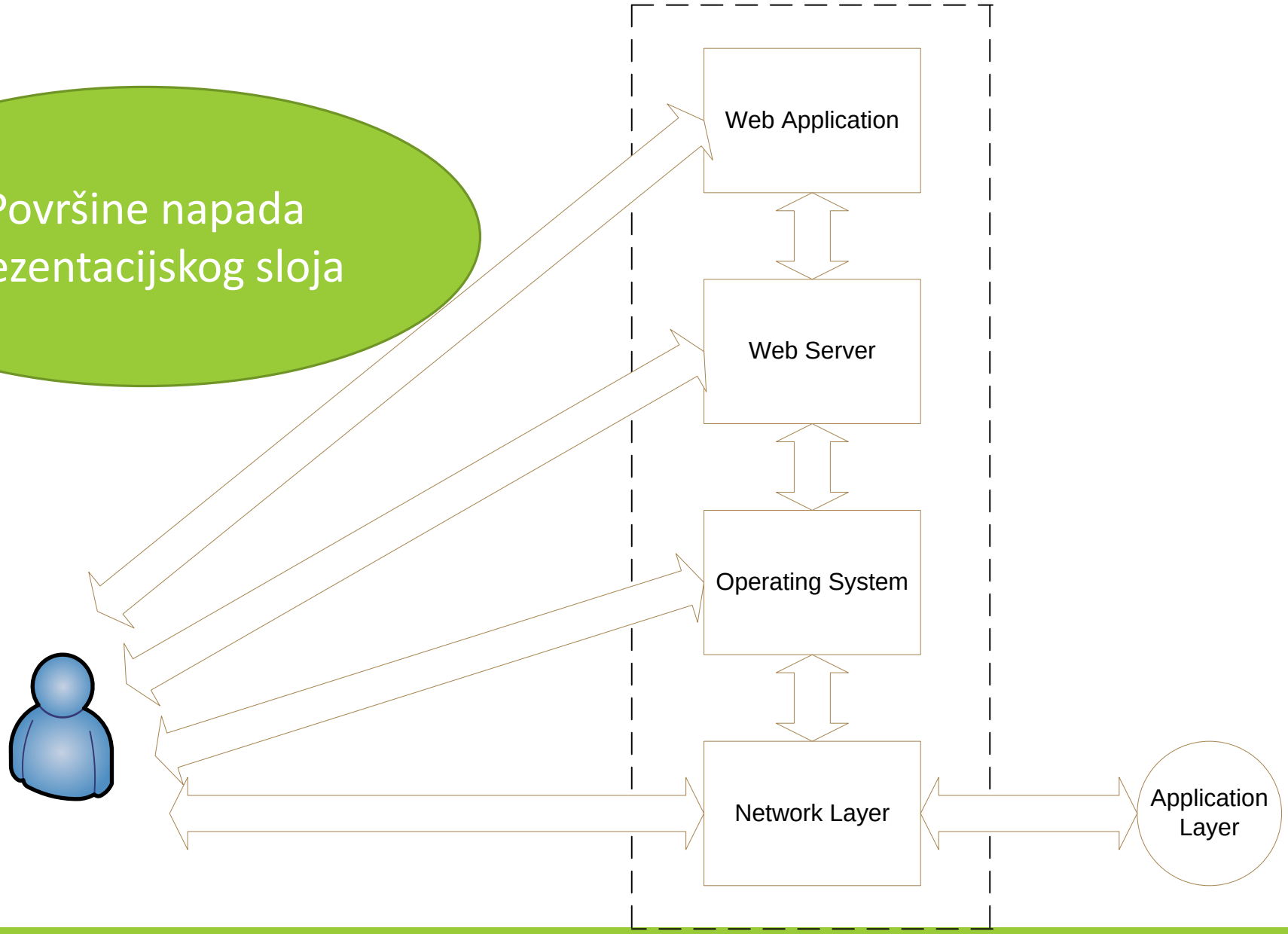
- Web masters
- Poslovni analitičari
- Help Desk
- Poslovne jedinice
- Pravna služba
- HR
- Projekt menadžer
- …

Rizik: Vlasništvo

Spriječeni smo u pravilnom osiguravanju "sustava" zbog vlasničkih problema, jer "naš sustav" je jedna komponenta u višeslojnoj aplikaciji.

Često nema odgovornosti za "Sigurnost sustava" koja uključuje svaki sloj, interkonekcije i konačnu usluge koju sustav pruža.

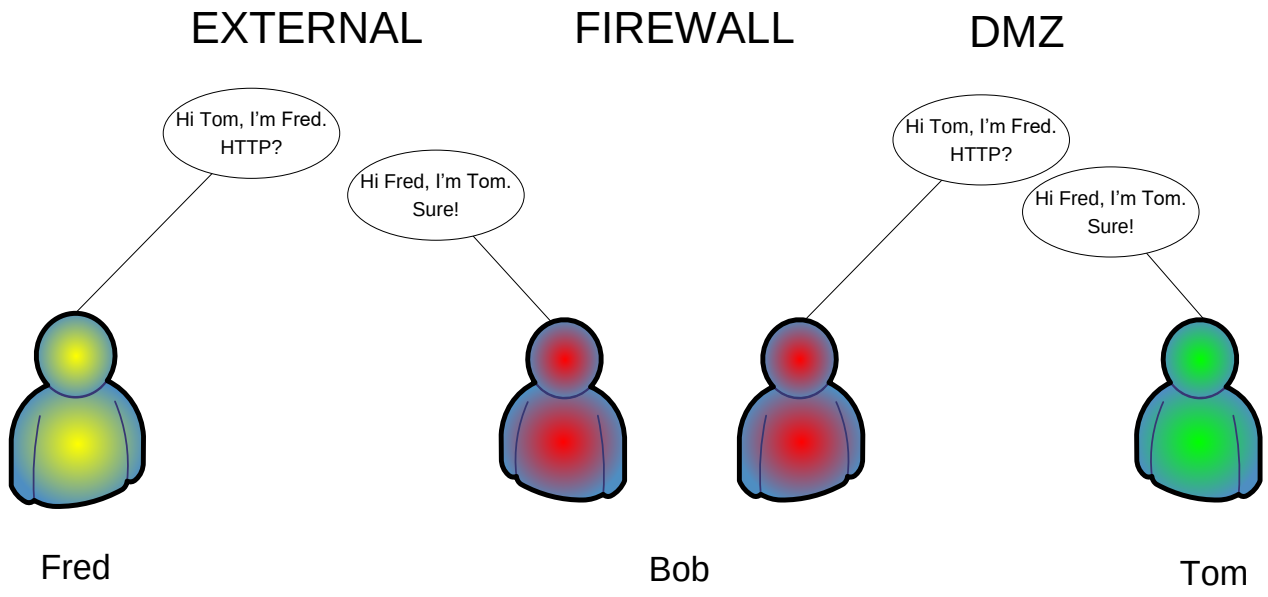
Površine napada prezentacijskog sloja



1. Razdvajanje mreža

Nema izravnog prometa između vanjskih i unutarnjih mreža (zona)

Zabrani sve što nije izričito naglašeno da smije proći

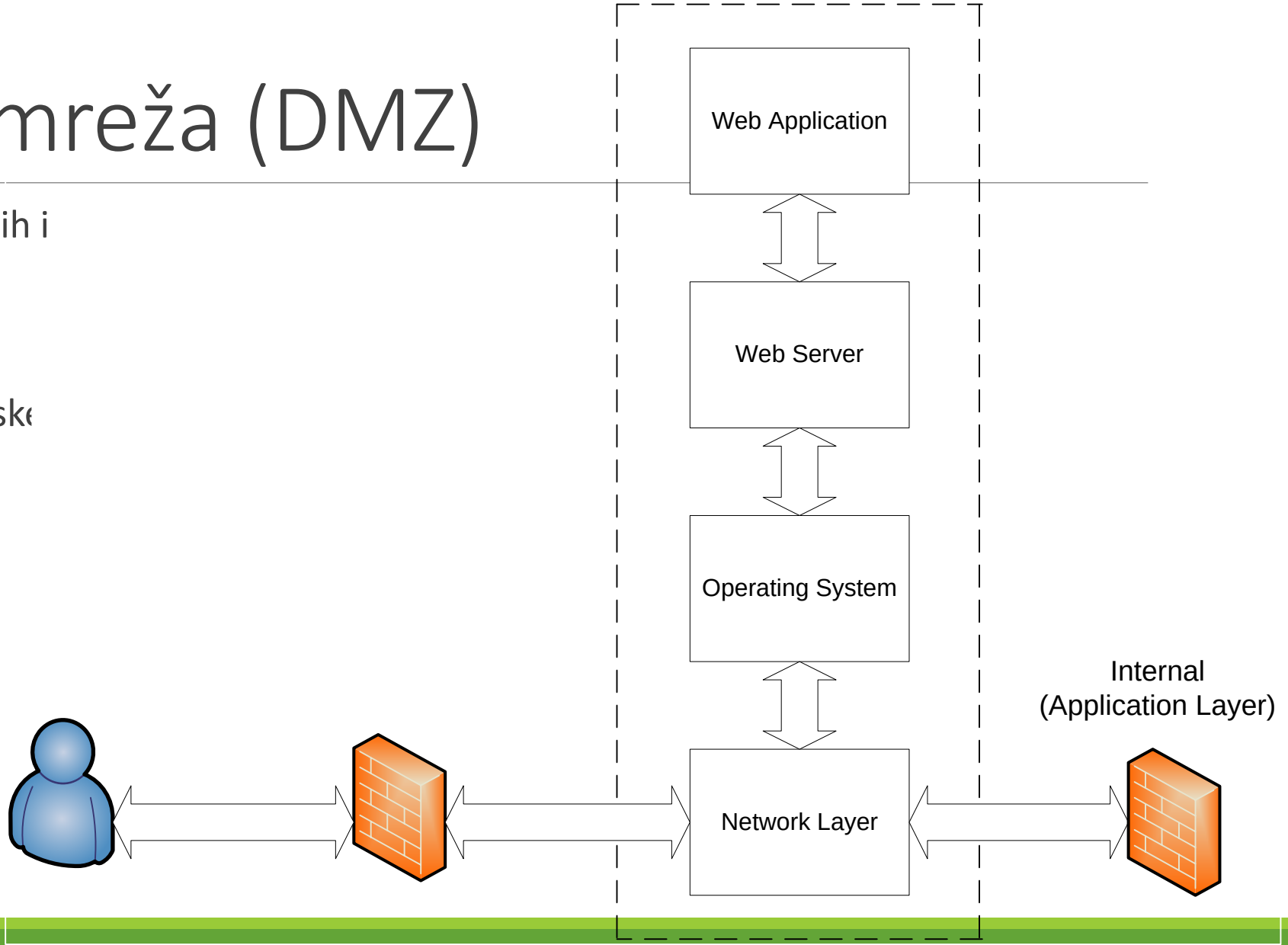


RULE: External to DMZ/Tom on service HTTP

2. Izolirana mreža (DMZ)

„Među-mreža” između vanjskih i internih mreža

Ne miješati testne i produkcijske mreže



3. Sterilno okruženje

Tretirati sustave unutar DMZ kao nesigurne

Prakticirati dobra pravila održavanja

- Ne držati osjetljive datoteke gdje im nije mjesto (debugs, dumps, temporary files…)

4. Paziti na upotrebu

- Servere koristiti samo za jednu svrhu
- Maknuti nepotrebne/nekorištene servise i aplikacije

6. Izolacija operacijskog sustava

Nikada ne instalirati aplikaciju na particiji na kojoj je operacijski sustav

Koristiti ACL liste za pristup svim direktorijima

7. Monitoriranje

- Koristiti IDS/IDP
- Logove prikupljati na odvojeni server
- Aplikacije koje samostalno stvaramo moraju generirati logove

8. Šifriranje podataka

Razumjeti koji tip podataka se u sustavu koristi

- Te ga klasificirati

Koristiti standardne kriptografske protokole za zaštitu podatka u tranzitu (SSL/TLS)

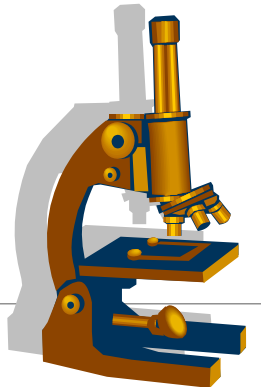
Koristiti standardne kriptografske algoritme za zaštitu podataka na disku

- i bazi podataka !

Šifriranje diska ne sprječava krađu podataka

- Već samo otežava krađu podataka u slučaju (fizičke) krađe diska

9. Praćenje korisnika u sustavu



Parametrizirati korisničko ime

Osigurati jedinstvenost identifikatora sesije

- i parametrizirati da

Slati korisničko ime i identifikator sesije kroz sve slojeve aplikacije

10. Ograničenja na bazi podataka

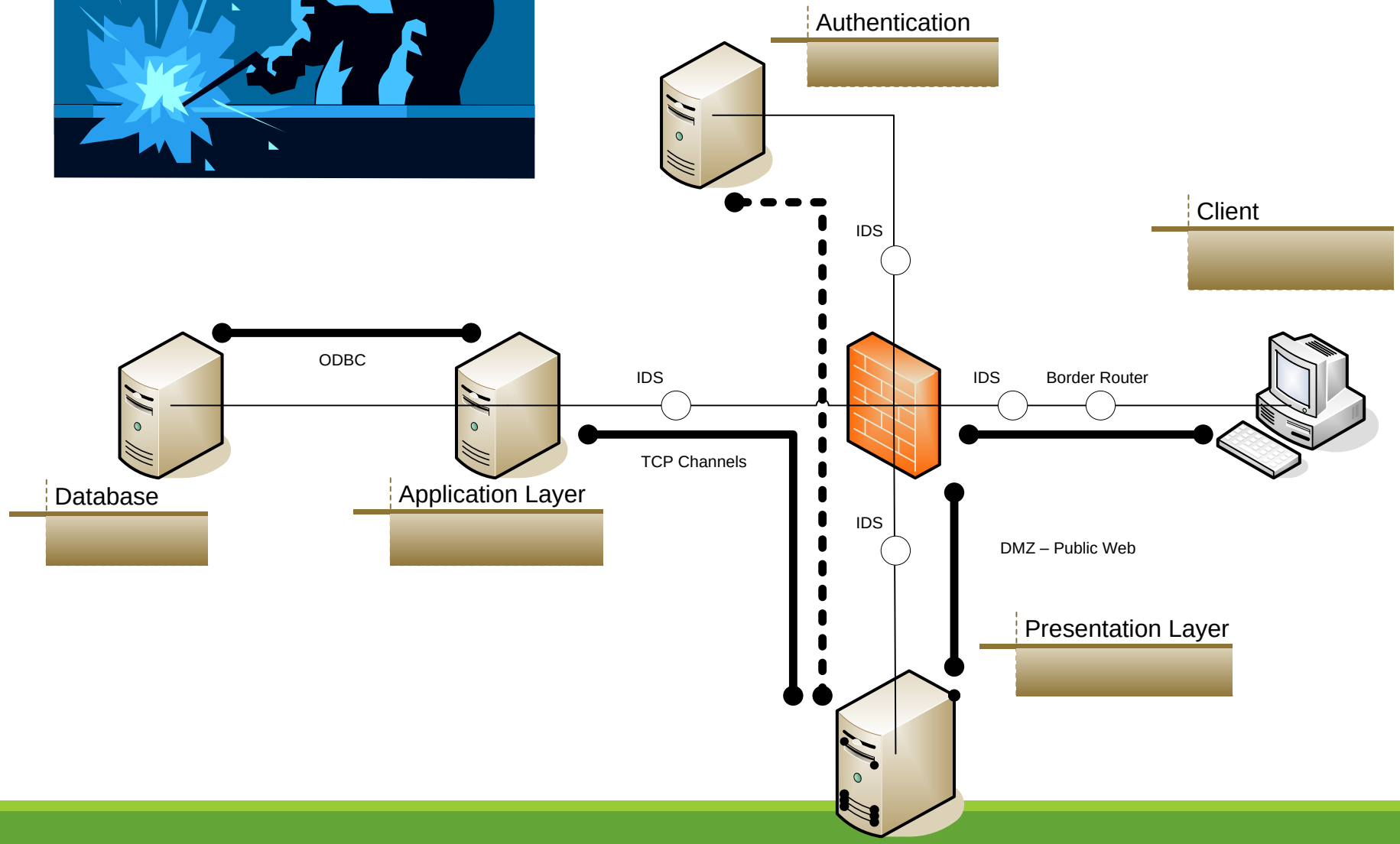
Korisnici ne mogu pristupiti bazi podataka

Programeri nemaju pristup produkciji

Maknuti pristup „stored procedurama” koje su namijenjena za sistemašenje

Sav pristup podatcima omogućiti kroz vlastite procedure

Maknuti programski kod iz baze podataka


Authentication
Client
IDS
ODBC
IDS
IDS
Border Router
TCP Channels
Database
Application Layer
IDS
DMZ – Public Web
Presentation Layer

NIST Guides to Security

http://csrc.nist.gov/publications/PubsSPs.html

SP 800-12

"An Introduction to Computer Security: The NIST Handbook"

SP 800-14

"Generally Accepted Principles and Practices for Securing Information Technology Systems"

SP 800-123

"Guide to General Server Security"

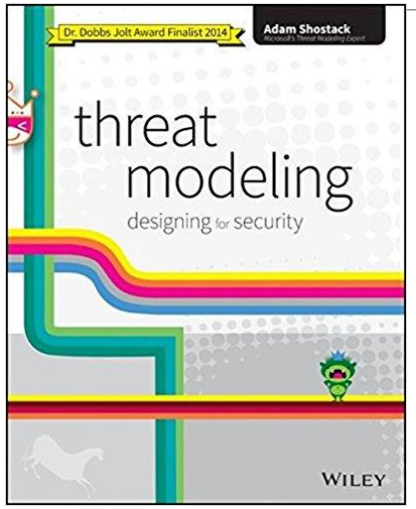
Section 2.4 Server Security Principles

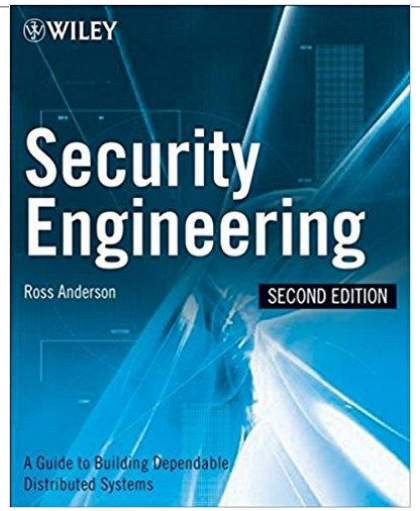
Više informacija

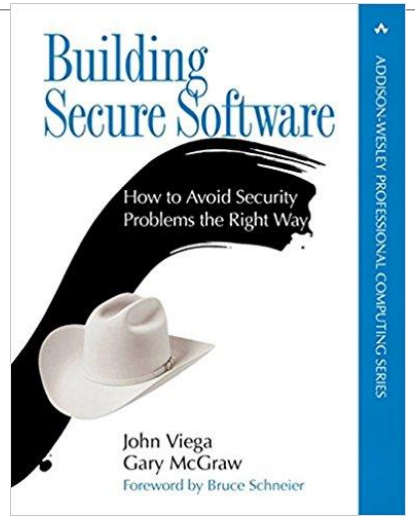
O načelima sigurnog dizajna

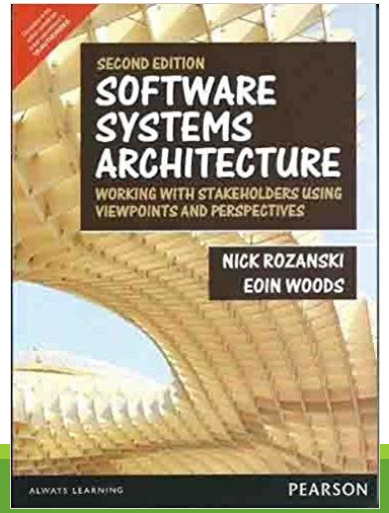
- Viega & McGraw (10)
- OWASP (10)
- NIST (33)
- NCSC (44)
- Cliff Berg’s set (185)

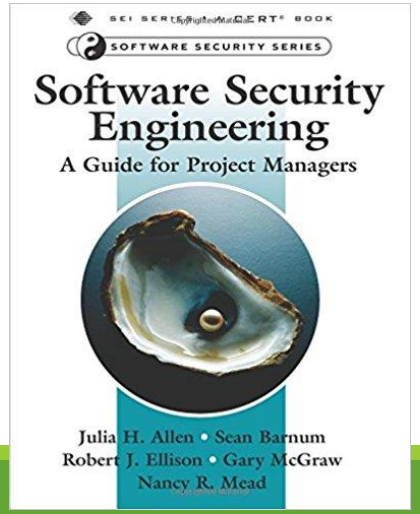
Literatura

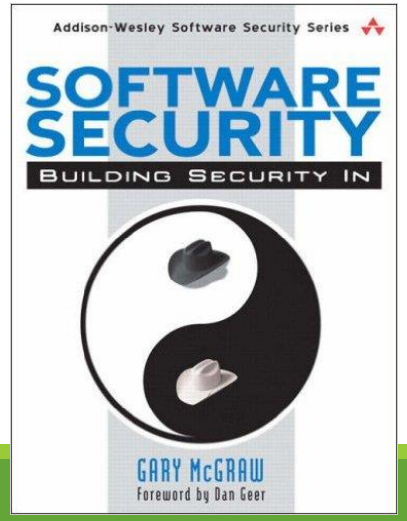












- https://www.cis.hr/dokumenti/dokumenti/
- https://www.cis.hr/edicija/edicija.html

Hvala na pažnji ! ☺

Diskusija

Pitanja