***BEZBEDNOST***

***Osnovne definicije i pojmovi***

***Zakon****:*

“Informaciona bezbenost predstavlja **skup mera** koje omogućavaju da podaci kojima se rukuje putem IKT sistema budu *zaštićeni* od neovlašćenog pristupa, kao i da se zaštiti *integritet*, *raspoloživost*, *autetntičnost* i *neoporecivost* tih podataka, da bi taj sistem funkcionisao kako je predviđeno, kada je predviđeno i pod kontrolom ovlašćenog lica.”

***Tajnost*** je svojstvo koje znači da podatak nije dostupan neovlašćenim licima. (svojstvo sistema koji rukuje podacima, posledica mera koje moramo da preduzmemo da bi podatak učinili tajnim, da bi otežali pristup podacima onima kojima taj podatak nije namenjen).

***Integritet*** znači očuvanost izvornog sadržaja i kompletnosti podatka(uvodimo mehanizme koji omogućavaju da sa jedne strane sprečimo nekoga da modifikuje podatke koje čuvamo ili razmenjujemo ili bar da možemo da detektujemo da su podaci izmenjeni).

***Raspoloživost*** je svojstvo koje znači da je podatak dostupan i upotrebljiv na zahtev ovlašćenih lica onda kada im je potreban.

* Vreme rasploživosti - odnos ukupnog vremena koje posmatramo i vreme kada nam podatak treba
* Zaštita raspoloživosti - uzrok smanjene raspoloživosti može biti:
* pristup velikog broja legitimnih korisnika u istom trenutku radi obavljanja nekog posla ili radi pristupa određenim podacima
* pokušaj nelegitimnih korisnika da učine servis nedostupnim legitimnim korisnicima (*„Denial of service attack“*)

**Autentičnost** je svojstvo koje znači da je moguće proveriti i potvrditi da je podatak stvorio ili poslao onaj za koga je deklarisano da je tu radnju izvršio (provera identiteta).

**Neoporecivost** predstavlja sposobnost dokazivanja da se dogodila određena radnja ili da je nastupio odeđeni događaj, tako da ga naknadno nije moguće poreći(veoma bitno za elektronsko plaćanje).

Navedeni pojmovi predstavljaju ***mehanizme*** koje uključujemo u sistem da bi ***osigurali*** osnovne aspekte ***bezbednosti***.

***Rizik i upravljanje rizikom*** ne predstavlja tehnički deo vezan za bezbednost već predstavlja deo ***vezan za regulativu*** (zakone, standarde,..). U nekim oblastima postoje propisane mere koje se moraju poštovati ukoliko sistem radi sa nekim osetljivim podacima ili funkcionalnostima koje su od opšteg značaja(upravljanje avio saobraćajem,upravljanje postrojenjima koja rukuju hemikalijama,..).

***Rizik*** znači mogućnost narušavanja informacione bezbednosti, odnosno mogućnosti narušavanja tajnosti, integriteta, raspoloživosti, autentičnosti ili neoporecivosti podataka ili narušavanja ispravnog funkcionisanja IKT sistema (omogućava da na neki način izmerimo koliko je moguće narušiti bezbednost sistema).

***Upravljanje rizikom*** je sistematičan skup mera koji uključuje planiranje, organizovanje i usmeravanje aktivnosti kako bi se obezbedilo da rizici ostanu u propisanim i prihvatljivim okvirima (mere koje predzimamo da smanjimo rizik).

***Pojmovi uže vezani za konkretne postupke zaštite sistema***

***Autentifikacija*** - mehanizmi za proveru identiteta

***Autorizacija*** - mehanizmi provere prava na sistemu

***Cleartext(Plaintext)*** – odnosi se na čist nezaštićen tekst, polazni oblik podatka u kom je on razmiljiv

***Kodovanje(sifrovanje)*** sadržaja je pretvaranje tog sadržaja u drugi oblik.

* Veoma je bitno da ***razlog kodovanja*** nije sakrivanje podataka(čuvanje tajnosti) već ***nemogućnost kanala za prenos podataka*** da iste podatke prenese u istom obliku u kom su ti podaci izvorno predstavljeni(npr. Base64 koji se koristi kada želimo da neke podatke npr.string prenesemo kanalom kojim možemo da prenosimo samo 6-bitni sadržaj, on će osmobitne podatke da pretvori u šestobitne, koristi se puno na webu jer je jedan od kanala kojim ne možemo da prenesemo osmobitne sadržaje url ili telo http stranica).
* Bitno je da znamo kojim algoritmom su kodovani da bi ih vratili u prvobitni oblik.Međutim, i kada mehanizam nije poznat moguće je otkriti koji algoritam je u pitanju.
* Ni jedna od modernih tehnika namenjenih ***za postizanje tajnosti*** podataka se ***ne oslanja na tajnost algoritma***.

***Enkripcija*** je mehanizam namenjen za pretvaranje sadržaja u drugi oblik ali tako da njegovo razumevanje bude moguće samo autorizovanim stranama.

* Rezultat enkripcije je ***Cyphertext***.
* Moderini mehanizmi za enkripciju, za očuvanje tajnosti u podacima, koriste neki algoritam kojim enkriptuju polazni sadržaj ***uz pomoć određenog ključa***. ***Tajnost ključa*** je nešto što ***obezbeđuje tajnost*** ukupnih ***eknriptovanih podataka*** a ne sam algoritam. Ali takođe, algoritam i dalje mora da bude takav da ne sadrži neke manjkavosti koje bi omogućile nekome da pogodi ključ ili nekako dođe do njega pa samim tim dođe i do sadržaja.
* Svi algoritmi za enkripciju polaze od matematike.
* ***Ključ*** je nešto što se kombinuje sa izvornim sadržajem upotrebom odabranog algoritma.

***Dekripcija*** vraća kriptovan podataka u prvobitan oblik(plain tekst).

***Vrste algoritama za enkripcij***

***ENKRIPCIJA SIMETRIČNIM KLJUČEM***

* + ***Isti ključ*** se koristi i za enkripciju i za dekripciju
  + Prednost: prilično ***brzi*** algoritmi
  + Danas postoji implementacija određenih funkcija koje se koriste u ovim algoritmima i u samim računarima zbog čega su takođe dosta brzi.
  + Mana: i pošiljalac i primalac moraju znati isti ključ, što znači da moraju na neki način da se sastanu i ***razmene ključeve***, što nije uvek moguće učiniti na bezbedan način.

***ENKRIPCIJA ASIMETRIČNIM KLJUČEVIMA***

* + ***Dva različita ključa***, gde se jedan korisi za enkripciju a drugi za dekripciju.
  + Između ovih ključeva postoji neki ***međusoban odnos***, oni imaju veze jedan sa drugim ali su tako kreirani da je veoma teško da od jednog dobijemo drugi.
  + U praksi od para ključeva jedan se zove ***javni ključ*** kojeg možemo da delimo sa bilo kim na bilo koji način ne moramo nigde da se nadjemo na istom mestu da bi ga prosedili primaocima naših poruka.
  + Drugi se naziva ***privatni ključ*** i od njega zavisi tajnost poruka, pa ga stoga čuvamo.
  + Kada neko ***šalje poruku*** nama, on ga enkriptuje našim javnim ključem, i samo mi možemo da ga dekriptujemo našim privatnim ključem, takođe ako mi šaljemo poruku, nju enkriptujemo javnim ključem onoga kome šaljemo, a on ga dekriptuje svojim tajnim ključem.
  + Ako enkriptujemo poruku privatnim ključem i postavimo taj sadržaj bilo gde, taj sadržaj će moći da bude enkriptovan samo našim javnim ključem, i na taj način je moguće proveriti ***da li smo mi ti koji smo poslali tu poruku*** jer je javni ključ povezan samo sa našim privatnim ključem.
  + ***Problem***: Kako da znamo tačno čiji je to javni ključ, bilo ko je mogao da postavi javni ključ isto kao i mi.

Problemom o proveri čiji je javni ključ se bavi ***infrastuktura javnih ključeva***(Public Key Infrastucture - PKI).

Postoje dva načina da obezbedimo proveru porekla javnih ključeva:

* ***Centralizovani pristup*** – podrazumeva postojanje ***sertifikcionih tela*** (CA), neke treće strane kojoj svi verujemo. Sertifikaciono telo uzima podatke o nama i o našem javnom ključu i potpiše ga elektronski. Rezultujući dokument se zove ***sertifikat***. Za sertifikat možemo da proverimo ko je njegov autor. Ako tom autoru verujemo i ako u tom sertifikatu piše da je taj ključ nečiji možemo da verujemo da taj ključ pripada baš njemu.
* ***Decentralizovani pristup*** – oblik mreže poverenja(Web of trust) podrazumeva da svako od nas veruje nekome za koga zna da njegov autoritet može da proveri direktno i onda nekako posredno možemo da proverimo identitet nekoga ko nam nije direktno poznat.

***HASH***

* Funkcija koja pretvara neki izvorni sadržaj u neki drugi sadržaj fiksne dužine.
* Prednost: ove funkcije se kreiraju tako da sa jedne strane nikako ne možemo od

hash-ovanog sadržaja dobiti originalan.

* Takođe male promene originalnog sadržaja treba da dovedu do velikih promena u rezultatu. Zbog toga je primarna namena hash funkcija provera integriteta sadržaja tj. da je sadržaj ostao nepromenjen (kada preuzimamo nešto pored linka za preuzimanje postoji i objavljen hash fajla).

***Čuvanje lozinki u sistemu***

1. ***U IZVORNOM OBLIKU***
   * Najjednostavnije
   * Problem: ako neko dođe do mesta gde su ti podaci skladišteni onda će imati i sve podatke i lozinke
2. ***U ENKRIPTOVANIM OBLIKU***
   * Moramo čuvati i ključ za enkripciju ali moramo ga čuvati na razčitiom mestu od podataka, jer i da neko uspe da dodje do baze gde su ostali podaci, nece moći da dekriptuje lozinke jer neće doći i do ključa
   * Problem: Vrlo je verovatno će, u slučaju da neko i dođe do naše baze, uspeti i da dođe do našeg ključa na file sistemu.
3. ***U HASHOVANOM OBLIKU***
   * Prednost: jednosmerna funkcija ukoliko dođe do naše baze neće moći da prepozna i lozinke. U tom slučaju kada se korisnik prvi put prijavljuje na sistem, mi čuvamo njegovu lozinku u hash-ovanom obliku, i prilikom svake prijave korisnika (autentifikacija) unetu lozinku hash-ujemo istim algoritmom i poredimo sa hash-om koji je sačuvan kao njegova lozinka. Nema potrebe da čuvamo kljuc, sve je u bazi, a to sto čuvamo je takvo da ukoliko neko dođe do naše baze neće moći da dođe i do lozinke.
   * Međutim, i dalje postoje načini da se sistem kompromituje od strane napadača. Jedan od načina jeste da smisli neku lozinku, napravi njen hash i proveri da li se poklapa sa nekom lozinkom iz baze. Iz tog razloga često se prave hash funkcije koje proizvode stringove koji imaju dosta karaktera kako bi napadaču bilo teže prepoznati hash neke lozinke.
   * Napadač takođe može napraviti rečnik u hash-ovanom obliku i proveriti da li neko koristi neku od tih reči za svoju lozinku.
   * Može analizairati da li ima hash-eva koji se ponavljaju tj. da više korisnika ima istu lozinku, tada je ta lozinka veoma jednostavna.
   * Napadač u tim slučajevima može unapred da sačuva hash-eve za česte lozinke ili čak čita rečnike i poredi sa hash-evima iz baze (RAINBOW TABLE).
4. ***U HASHOVANOM OBLIKU SA DODATKOM***
   * Dodatak se zove ***salt***.
   * Kada se korisnik prvi put prijavljuje na sistem i unese željenu lozinku, na taj string dodamo neki ***nasumično odabran*** salt,tako dobijeni string hash-ujemo i sačuvamo ga,kao i nasumično odabrani salt. Kada se taj korisnik ponovo prijavljuje, onda nalepimo salt na ono sto je uneto i onda poredimo sa onim hashom koji je sačuvan. Iako se čuva na istom mestu kao hash, suština je da salt nije osetljiv podatak, napadač neće ništa dobiti time što zna koji je salt, cilj je da sve i da je lozinka korisnika jednostavna i da drugi korisnici imaju istu lozinku, taj salt koji se nasumično bira, će učiniti da rezultat hasha bude različit (mala promena podatka velika promena hash-a).
   * Problem: I dalje nije rešen brute force napad, kao ni napad pomoću rainbow tabela napravljenih pomoću rečnika. Hash funkcija je veoma ***brza*** te omogućava napadaču da može ***puno pokušaja*** da napravi za ***kratak vremenski period***.
5. ***SPECIJALIZOVANI ALGORITAM***
   * Osim onih osnosvnih hash funkcija koje su kreiranje za proveru integriteta podataka i pravljene da budu brze. Vraćamo se na to da je bolje da koristimo neki specijalizovani algoritam koji je namenjen za čuvanje lozinku. Ovi algoritmi kombinuju razne tehnike, i dodavanje salta, i razne druge algoritme hash-ovanja. Ideja je da ako je neki od algoiritama hash-ovanja lako kompromitovati da imamo neki u lancu koji je teže komprovitovati. Zatim čitav taj postupak ponavljamo više puta, sa ciljem da se namerno uspori čitav proces da bi onom ko zna algoritam kojim su čuvane lozinke, treba dugo vreme za nasumične pokušaje. Tako smo otežali i brute force.

***Dodatne mere bezbednosti***

1. ***MULTI-FACTOR AUTENTIKACIJA***
   * Lozinka nije jedina stvar koja je potrebna za logovanje. Kombinuje se ono što korisnik zna (lozinka) i ono što korisnik ima (TAN tablicu, digitalni sertifikat, telefon).Ako korisnik uspešno unese korisnicno ime i lozinku, dobija nasumičan kod sms-om i ako vrati taj kod koji je poslat on ima i telefon koji je njegov i tako su kombinovane dve provere da bi potvrdili da je to taj korisnik.
   * Možemo ovakvu autentifikaciju pogrešno implementirati i tako uništiti sve zbog čega ovaj postupak postoji (za promenu lozinke dobicete sms - ili telefon ili lozinka za potvrdu).
2. ***OAUTH***
   * omogućiti prijavu na servis preko nekog drugog servisa pa se cela autentifikacija prepusta tim servisima.

Čuvanje lozinke je besmisleno ukoliko neko može da presretne saobraćaj pa moramo da zaštitimo komunikaciju korisnika i servisa. Zato se korisit HTTPS ali on koristi TLS protokol za enkripciju koji zahteva da imamo sertifikat koji možemo kupiti(pomocu CA) , samopotpisati (ali svi korsinici dobijaju obavestenje o tome, korisiti se za testiranja), ili koristiti besplatan (za određeni domen koji posedujemo ili aplikaciju koju pravimo)