INFORMACIONA BEZBEDNOST

1. UVOD U KRIPTOGRAFIJU

- **1. Uvod :**
- **Terminologija:**
  - Pošiljalac (sender): Entitet koji šalje poruku.
  - Primalac (receiver): Entitet koji prima poruku.
  - Poruka (message): Informacija koja se prenosi.
  - Otvoreni tekst (plaintext): Nešifrovani oblik poruke.
  - Šifrovani tekst (ciphertext): Tekst koji je enkriptovan.
  - Šifrovanje (encryption): Proces pretvaranja otvorenog teksta u šifrovani tekst.
  - Dešifrovanje (decryption): Proces pretvaranja šifrovanog teksta u otvoreni tekst.
  - Kriptografija (cryptography): Nauka o sigurnom komuniciranju.
  - Obezbeđivanje tajnosti poruka: Glavni cilj kriptografije.
  - Kriptoanaliza (cryptanalysis): Proces analize šifrovanih poruka radi otkrivanja tajnog sadržaja.
  - Kriptologija (cryptology): Širi pojam koji obuhvata kriptografiju i kriptoanalizu.
- **Osnovni zadaci kriptografije:**
  - Poverljivost poruka: Očuvanje tajnosti komunikacije.
  - Autentifikacija: Provera identiteta pošiljaoca i/ili sadržaja poruke.
  - Integritet: Zaštita poruke od izmena.
  - Neporecivost: Onemogućavanje poricanja slanja poruke.
- **Šifre i ključevi:**
  - Kriptografski algoritam: Matematičke funkcije korišćene za šifrovanje i dešifrovanje.
  - Simetrični algoritmi: Koriste isti ključ za šifrovanje i dešifrovanje.
  - Asimetrični algoritmi: Koriste različite ključeve za šifrovanje i dešifrovanje.
- **Kriptoanaliza:**
  - Svrha: Otkrivanje tajnih poruka ili ključeva.
  - Tipovi napada: Cyphertext-only, known-plaintext, chosen-plaintext, adaptive-chosen-plaintext, chosen-ciphertext, chosen-key, rubber-hose.
- **Kriptografija pre i posle pojave računara:**
  - Pre: Algoritmi zasnovani na karakterima.
  - Posle: Algoritmi rade nad bitovima.
- **Šifre zamene i premeštanja:**
  - Monoalfabetske, homofonske, poligramske, polialfabetske.
  - Cezarova šifra, ROT13.
- **Jednokratna sveska:**
  - Savršena šifra koja koristi slučajne nizove za enkripciju.
- **Problemi u kriptografiji pre pojave računara:**
  - Generisanje slučajnih nizova, distribucija nizova, sinhronizacija učesnika, ograničenost trake.
- **2. Protokoli** :

  su serija postupaka, s najmanje dva učesnika, namenjena obavljanju nekog zadatka. Svi učesnici moraju poznavati protokol i znati potrebne korake unapred. Protokol mora biti nedvosmislen, s dobro definisanim koracima kako bi se izbegli nesporazumi. Postoji nekoliko tipova protokola, uključujući arbitrated, adjudicated i self-enforcing.
- **Tipovi protokola - Arbitrated**
  - Arbitrated protokoli koriste arbitra ili treću osobu kojoj svi veruju, a koja nije zainteresovana za ishod protokola. Ovo može uključivati advokate, banke ili notare kao arbitre.
- **Tipovi protokola - Adjudicated**

- Adjudicated protokoli mogu biti nearbitrated ili arbitrated. Nearbitrated protokol se pokreće svaki put kada učesnici žele da kompletiraju protokol, dok se arbitrated protokol pokreće samo kada postoji spor između učesnika. Adjudicator, kao sudija, nije zainteresovan za ishod protokola i poziva se samo je potrebno utvrditi da li je protokol izvršen pravično.

- **Tipovi protokola - Self-enforcing**
  - Self-enforcing protokoli garantuju pravičnost sami po sebi i ne zahtevaju arbitra ili adjudicatora. Oni su dizajnirani tako da minimizuju mogućnost sporova i varanja.
- **Napadi na protokole**
  - Napadi na protokole mogu biti pasivni ili aktivni. Pasivni napadi uključuju prisluškivanje protokola, dok aktivni napadi uključuju izmenu ili ometanje protokola. Varalice mogu biti pasivne ili aktivne, a njihov cilj je dobijanje više informacija ili ometanje protokola.
- **<u>3. Simetrični algoritmi:</u>**
- **Komunikacija pomoću simetričnih algoritama:**
  - **Dogovor algoritma i ključa:**
    - Alice i Bob dogovaraju koji simetrični algoritam će koristiti.
    - Takođe, dogovaraju i ključ koji će koristiti za šifrovanje i dešifrovanje poruka.
  - **Šifrovanje i slanje poruke:**
    - Alice koristi dogovoreni algoritam i ključ da bi šifrovala svoju poruku.
    - Zatim šalje šifrirani tekst Bobu.
  - **Dešifrovanje poruke:**
    - Bob koristi isti algoritam i ključ da bi dešifrovao poruku koju je dobio od Alice.
- Mogućnosti napada - **Eve**:
  - **Prisluškivanje komunikacije:** Eve može pokušati da prisluškuje komunikaciju u koraku 4 kada se šifrovana poruka šalje.
  - **Known-ciphertext napad:** Eve može prisluškivati komunikaciju u koraku 1, ali to je dopušteno jer postoje javni simetrični algoritmi koji su dovoljno dobri.
  - **Prisluškivanje dogovora ključa:** Međutim, nije dopušteno prisluškivanje komunikacije u koraku 2. Alice i Bob moraju dogovoriti ključ u tajnosti.
- Mogućnosti napada - **Mallory**:
  - **Presretanje komunikacije:** Mallory može pokušati presresti komunikaciju između Alice i Boba.
  - **Slanje lažnih poruka:** Ako Mallory presretne komunikaciju u koraku 2, može slati lažne poruke.
  - **Distribucija ključeva:** Ključevi se moraju distribuirati sigurnim komunikacionim kanalima.
  - **Rizik kompromitovanja ključeva:** Ako je ključ kompromitovan, sve poruke šifrovane tim ključem postaju kompromitovane.
- **Napomena:** Kako bi se smanjio broj ključeva, poseban ključ se može koristiti za komunikaciju između svakog para učesnika u mreži. Za mrežu od n učesnika, potrebno je n(n-1)/2 ključeva.
- **<u>4. Jednosmerne funkcije:</u>**
- **Jednosmerne hash funkcije:**
  - Funkcije koje imaju ulaz promenljive dužine, ali izlaz fiksne dužine.
  - Primer jednosmerne hash funkcije je **java.lang.String.hashCode()**.
  - Collision-free funkcije:
    - Teško je generisati dva različita ulaza koji daju istu hash vrednost.
    - Funkcije su javne, a tajnost je sadržana u svojstvu jednosmernosti.
    - Promena jednog bita u ulazu rezultuje promenom u proseku polovine bitova na izlazu.
- **Message Authentication Codes (MAC)**
  - MAC kombinuje jednosmernu hash funkciju sa ključem za šifrovanje. Ključ za šifrovanje se koristi za generisanje hash vrednosti. Samo osoba koja poseduje ključ može proveriti validnost hash vrednosti.
  - Jednosmerna hash funkcija se koristi za generisanje hash vrednosti.
  - Ključ za šifrovanje se koristi za dodatnu sigurnost.

- Hash vrednost se može proveriti samo ako osoba ima odgovarajući ključ.

- **5. Asimetricni algoritmi:**
- Asimetrični algoritmi omogućavaju komunikaciju bez razmene tajnih ključeva, što čini proces komunikacije jednostavnijim. Evo osnovnih koraka u komunikaciji pomoću asimetričnih algoritama:
- **Dogovor algoritma:** Alice i Bob se dogovaraju o algoritmu koji će koristiti za enkripciju i dekripciju poruka.
- **Razmena javnih ključeva:** Bob šalje svoj javni ključ Alice.
- **Enkripcija poruke:** Alice koristi Bobov javni ključ da enkriptuje svoju poruku.
- **Slanje šifrovane poruke:** Alice šalje šifrovanu poruku Bobu.
- **Dekripcija poruke:** Bob dekriptuje poruku koristeći svoj tajni ključ.
- **6. Digitalni potpisi:**
- Digitalni potpisi su važni jer omogućavaju bezbednu komunikaciju i proveru autentičnosti dokumenata, ali imaju i praktična ograničenja, kao što su sporost asimetričnih algoritama za potpisivanje velikih dokumenata.
- Digitalni potpisi imaju nekoliko ključnih osobina:
    - **Autentičnost:** Potvrđuje da je potpisnik zaista potpisao dokument.
    - **Nije ponovo iskoristiv:** Potpis se ne može preneti na drugi dokument.
    - **Nepromenljivost potpisanog dokumenta:** Dokument ostaje nepromenjen nakon potpisivanja.
    - **Neporecivost:** Potpisnik ne može kasnije negirati svoj potpis.
- Postoje tri osnovna pristupa digitalnim potpisima:
    - **Pomoću simetričnog algoritma i arbitratora:** U ovom pristupu treći učesnik, arbitrator, koristi simetrični algoritam za potvrdu. Arbitrator je osoba kojoj obe strane veruju, a ključevi za simetrični algoritam se definiraju između svih učesnika i arbitratora.
    - **Pomoću asimetričnog algoritma:** U ovom pristupu potpisnik koristi svoj tajni ključ za enkripciju poruke, čime je i potpisuje, a primalac dekriptuje poruku korišćenjem javnog ključa potpisnika.
    - **Pomoću jednosmernih hash funkcija:** Umesto potpisivanja celog dokumenta, potpisuje se njegov hash. Ovo omogućava proveru autentičnosti i integriteta dokumenta bez potrebe za velikim količinama podataka.
- **7. Sertifikati:**
- **Šta je sertifikat:** Sertifikat je digitalna potvrda koja garantuje autentičnost identiteta entiteta, kao što su osobe, organizacije ili uređaji. U osnovi, sertifikat sadrži informacije o entitetu, kao i javni ključ tog entiteta.
- **Certificate Authority (CA):** Certificate Authority je pravno lice od poverenja koje izdaje sertifikate. Njegov javni ključ je poznat, a koristi se za potpisivanje sertifikata kako bi se osigurala njihova autentičnost.
- **Ulančavanje sertifikata (certificate chaining):** To je proces gde CA može da izda sertifikat sa naznakom da je primalac ovlašćen da izdaje dalje sertifikate. Ovo omogućava formiranje lanca sertifikata, gde se svaki sertifikat može pratiti do korenskog CA.
- **CA hijerarhija:** CA hijerarhija je lanac sertifikata koji ide od neposrednog CA do korenskog CA. Korenski CA je self-signed sertifikat koji je ugrađen u browsere i druge softvere kako bi se omogućilo proveravanje autentičnosti drugih sertifikata.
- **X.509 standard:** Ovaj standard definiše formate za sertifikate, uključujući strukturu podataka i način njihove razmene. Hijerarhijska organizacija imena, potiče od X.500 standarda, omogućava jedinstveno identifikovanje entiteta.
- **8. Ključevi:**
- **Razmena ključeva:**
    - Šifrovanje svake pojedine konverzacije posebnim ključem za sesiju.
    - Distribucija ključeva za sesije je poseban problem.
    - Može se izvesti pomoću simetričnog algoritma ili asimetričnog algoritma.
    - ***Korišćenje simetričnog algoritma:***
        - **Alice traži od Trenta novi session key:** Alice inicira zahtev za novim ključem kod Trenta.

        - **Trent generiše session key:** Trent generiše jedinstveni session key i šifruje ga dva puta, koristeći javne ključeve Alice i Boba. Obe šifrovane poruke šalje Alice.
        - **Alice dešifruje svoju kopiju session key-a:** Alice dešifruje svoju kopiju session key-a koristeći svoj tajni ključ.
        - **Alice šalje Bobu njegovu kopiju session key-a:** Alice prosleđuje Bobu njegovu kopiju session key-a.
        - **Bob dešifruje svoju kopiju session key-a:** Bob dešifruje svoju kopiju session key-a koristeći svoj tajni ključ.
        - **Korišćenje session key-a:** Nakon uspešne razmene, Alice i Bob koriste session key za dalju komunikaciju.
    - ***Korišćenje asimetričnog algoritma:***
        - **KDC (Key Distribution Center):** Postoji javna baza podataka sa svim potpisanim javnim ključevima. Kada Alice želi da pošalje poruku Bobu, prvo uzima Bobov javni ključ iz KDC.
        - **Generisanje session ključa:** Alice generiše slučajni session ključ i šifruje ga Bobovim javnim ključem, a zatim ga šalje Bobu.
        - **Dešifrovanje session ključa:** Bob dešifruje Alicinu poruku svojim tajnim ključem.
        - **Dalja komunikacija:** Nakon uspešne razmene ključeva, dalja komunikacija koristi session key.
    - Mere opreza kako bi se osiguralo da treće strane poput Eve ili Mallory ne mogu da presretnu ili modifikuju ključeve tokom razmene.
    - **Šta može Eve?**
        - Ciphertext-only napad.
    - **Šta može Mallory?**
        - Man-in-the-middle napad.
- **Interlock protokol:**
    - Protokol za međusobnu autentifikaciju pomoću razmene polovina poruka.
- **Razmena ključeva sa digitalnim potpisima:**
    - Koristi se Key Distribution Center (KDC) koji potpisuje sve javne ključeve.
    - Mallory ne može da zameni javne ključeve prilikom presretanja komunikacije.
- **Komunikacija bez prethodne razmene ključeva:**
    - Koristi se javni ključ Boba iz KDC.
    - Ključevi se šifruju javnim ključem primaoca.
- **Dužina ključeva:**
    - Dužina ključeva igra ključnu ulogu u sigurnosti kriptosistema.
    - Preporučena dužina ključeva varira u zavisnosti od algoritma i namene.
- **Upravljanje ključevima:**
    - Upravljanje ključevima je kritično za bezbednost sistema.
    - Treba obratiti pažnju na distribuciju, skladištenje, backup i "rok trajanja" ključeva.
- **9. Tipovi algoritama:**
- **Block Cipher (blok šifra):**
    - Radi sa fiksnim blokovima otvorenog i šifriranog teksta.
    - Tipično operiše nad blokovima od 64 bita.
- **Stream Cipher (tok šifra):**
    - Radi sa tokom otvorenog/šifriranog teksta po bitu, bajtu ili reči istovremeno.
    - Koristi keystream generator za generisanje niza pseudoslučajnih ključeva.
- **Self-synchronizing Stream Cipher (samosinhronizujuća tok šifra):**
    - Funkcioniše slično kao stream cipher, ali se svaki blok tretira kao shift registar.
    - Inicijalno je popunjen slučajnim sadržajem i samosinhronizuje se tokom procesa šifrovanja.
- **10. Režimi rada:**
- **Electronic Codebook (ECB):**

- Svaki blok otvorenog teksta se šifruje nezavisno istim ključem.
- Podložan je aktivnim napadima, jer isti blokovi imaju iste šifrirane rezultate.

- **Cipher Block Chaining (CBC):**
  - Otvoreni tekst svakog bloka se XOR-uje sa šifriranim tekstom prethodnog bloka.
  - Prvi blok se XOR-uje sa slučajnim inicijalizacionim blokom.
- **Cipher Feedback (CFB):**
  - Blok šifra implementirana kao samosinhronizujuća stream šifra.
  - Blok se tretira kao shift registar i XOR-uje sa ulaznim podacima iz otvorenog teksta.
- **Output Feedback (OFB):**
  - Slično kao CFB, ali se u blok registar dodaje element iz rezultata šifrovanja pre XOR-ovanja.
- **Counter (CTR):**
  - Svaki blok otvorenog teksta se XOR-uje sa brojačem koji se razlikuje za svaki blok.
  - Brojač se inkrementira za svaki naredni blok, a vrednost brojača se šifrira i XOR-uje sa otvorenim tekstom.

2. PUBLIC KEY INFRASTRUCTURE

- **1. Komunikacija pomoću simetričnih algoritama:**
  - Alice i Bob se dogovore o algoritmu i ključu.
  - Alice šifruje poruku dogovorenim algoritmom i ključem.
  - Alice šalje šifrirani tekst Bobu.
  - Bob dešifruje poruku istim algoritmom i ključem.
- **Mane komunikacije pomoću simetričnih algoritama:**
  - Razmena ključeva mora biti obavljena unapred, jer se isti ključ koristi za šifrovanje i dešifrovanje.
  - Postoji manjak poverljivosti, kontrole ili autentičnosti, jer svako ko ima pristup simetričnom ključu može šifrovati i dešifrovati poruke.
- **Komunikacija pomoću asimetričnih algoritama:**
  - Alice uzima Bobov javni ključ iz KDC.
  - Alice šifruje svoju poruku Bobovim javnim ključem.
  - Alice šalje šifrovanu poruku Bobu.
  - Bob dešifruje poruku svojim tajnim ključem.
- **Prednosti asimetričnih algoritama:**
  - Zahteva upravljanje sa manje ključeva jer svaka strana ima par ključeva, tako da je ukupan broj ključeva 2n umesto n^2.
  - Privatni ključevi se ne moraju distribuirati drugoj strani.
  - Tajni ključevi se ne prenose preko mreže, što otežava njihovu kompromitaciju.
  - Javni ključevi se mogu koristiti za šifriranje privremenih ključeva kako bi se izbeglo veće računarsko opterećenje.
  - Digitalni potpisi funkcionišu po ovom principu i omogućavaju dokazivanje neporecivosti.
- **Asimetrični vs. Simetrični:**
  - Asimetrični algoritmi imaju redak prostor ključeva i sporu brzinu, dok simetrični algoritmi imaju gust prostor ključeva i brzu brzinu.
- **Kombinacija oba pristupa:**
  - Za bezbednu razmenu dokumenata može se koristiti simetrični algoritam za šifrovanje dokumenata, dok se simetrični ključ šifrira asimetričnim javnim ključem svakog primaoca i dodaje uz dokument.
- **2. Potreba za infrastrukturom javnih ključeva (PKI):**
  - PKI je mehanizam za distribuciju i korišćenje javnih ključeva kako bi se osigurala bezbednost.
  - Postojanjem jednog popisnika kome svi veruju (certificate authority) obezbeđuje se poverenje milionima sertifikata
- **3. Digitalni sertifikati:**

- Digitalni sertifikati su elektronski dokumenti koji sadrže informacije o identitetu entiteta, poput pojedinca, organizacije ili uređaja, i njihovom javnom ključu. Oni se izdaju od strane potpisnika kome svi veruju (Certificate Authority - CA).
- X.509 verzija 3 je opšte prihvaćeni standard za digitalne sertifikate.
- Sertifikati se obično enkoduju u specijalnom binarnom formatu ASN.1 i često se MIME enkoduju kako bi se mogli prenositi ASCII znakovima, kao što je u slučaju slanja e-mailom.
- Sertifikat sadrži informacije o entitetu (subjektu), CA koji ga je izdao, period važenja, serijski broj, javni ključ subjekta i digitalni potpis CA.
- Subjekti koji koriste sertifikat moraju verovati CA i njegovom javnom ključu, koji su dobili tokom registracije.

- **4. Certificate Revocation List (CRL):**
  - CRL je lista koju izdaje CA koja sadrži informacije o sertifikatima koji su povučeni.
  - Ona se koristi u situacijama kada je tajni ključ subjekta kompromitovan ili kada postoji drugi razlog za povlačenje sertifikata.
  - Svi subjekti koji koriste sertifikat moraju proveriti CRL pre upotrebe javnog ključa subjekta.
  - CRL sadrži informacije o serijskom broju povučenog sertifikata, vremenu povlačenja i digitalni potpis CA.
- **Vrste CRL:**
  - **Full and complete CRL**: Sadrži sve informacije o povlačenju svih sertifikata izdatih od CA.
  - **Authority Revocation List (ARL)**: Sadrži informacije o povlačenju za sve CA sertifikate izdate od CA.
  - **Distribution-point CRL**: Omogućava distribuciju CRL sa više CA ili particionisanje informacija o povlačenju.
  - **Delta CRL**: Sadrži samo razlike u povlačenim sertifikatima u odnosu na prethodnu CRL.
- **5. Alternative za CRL:**
  - **Online Certificate Status Protocol (OCSP)**: Omogućava proveru statusa sertifikata putem upita OCSP Responderu koji odgovara sa informacijom o statusu sertifikata.
  - **Simple Certificate Validation Protocol (SCVP)**: Omogućava proveru statusa sertifikata putem upita pouzdanom autoritetu.
  - **Direktorijum-based revocation**: Korisnici proveravaju status pojedinačnih sertifikata iz direktorijuma, a CA može ažurirati status sertifikata u direktorijumu.
  - **B-stablo**: CA ili drugi pouzdani server organizuje informacije o povlačenju u strukturu B-stabla kako bi se olakšala pretraga.
- **6. Enterprise PKI:**
  - Enterprise PKI (Public Key Infrastructure) koristi sertifikate kako bi se omogućilo jednostavno korišćenje javnih ključeva, ali uspostavljanje poverenja u validnost sertifikata je kompleksno.
  - Svaka grupa korisnika koju pokriva jedan CA naziva se domen, a svi korisnici unutar domena dobijaju sertifikate odgovarajućeg CA.
  - Registracioni autoritet (RA) je predstavnik CA koji autentifikuje korisnike, preuzima javne ključeve i šalje zahtev za kreiranje sertifikata CA-u.
  - Sertifikaciona polisa (CP) definiše pravila koja ukazuju na primenjivost sertifikata na određenu zajednicu ili klasu aplikacija sa zajedničkim bezbednosnim zahtevima.
  - Certification Practice Statement (CPS) opisuje kako se pravila sertifikacione polise primenjuju.
- **7. Globalni PKI:**
  - Principi koji se primenjuju za implementaciju jednog PKI mogu se proširiti kako bi podržali globalni PKI koji se sastoji od više CA koji mogu da sertifikuju druge CA.
  - Trust model ili graf poverenja definiše način na koji CA sertifikuju jedni druge.
  - Postoje različiti trust modeli, kao što su striktna hijerarhija, hijerarhija, most (bridge), višestruke tačke poverenja i mreža (mesh).
  - Interoperabilnost PKI zavisi od faktora kao što su putanja poverenja, korišćeni algoritmi, formati sertifikata i CRL, sertifikacione polise i različiti nazivi subjekata.

- **8. Oporavak ključeva:**
- **Reizdavanje ključeva:**
    - Sertifikati javnog ključa imaju definisan period važenja, nakon čega su potrebni novi sertifikati sa novim javnim ključevima.
    - Ograničen vek trajanja sertifikata pomaže u mitigaciji potencijalnih pretnji kriptoanalizom i kontroli veličine Certificate Revocation List (CRL).
    - Subjekat može zatražiti reizdavanje ključa, a CA to može automatski obaviti.
- **Oporavak ključeva:**
    - Tehnike oporavka ključeva su dizajnirane za hitne situacije kada je tajni ključ oštećen ili subjekat zaboravi lozinku.
    - Dva popularna mehanizma za oporavak ključeva su deponovanje ključeva (key escrow), gde se privatni ključ pruža trećoj strani, i enkapsulacija ključa, gde se ključ šifrira javnim ključem treće strane.
- **9. Cena PKI:**
    - Uspostavljanje PKI može delovati skupo, ali treba uporediti te troškove sa alternativama.
    - PKI omogućava zaštitu podataka koji se prenose preko nepouzdane mreže i zahteva upravljanje sa manje ključeva u poređenju sa simetričnim algoritmima.
    - Cena PKI može biti visoka kada je potrebno obezbediti globalno poverenje i interoperabilnost, ali je to neophodno za sigurnu komunikaciju. Bez korišćenja PKI, preuzimanje rizika može biti alternativa.

3. IDENTIFIKACIJA I AUTENTIFIKACIJA

- **1. Uvod:**
- **Identifikacija:**
    - Proces dodeljivanja ID-a čoveku, računaru ili drugoj mrežnoj komponenti.
    - Identifikacija je prvi korak u autentifikaciji, ali nije isto što i autentifikacija.
    - Identifikacija je sastavni deo procesa autentifikacije.
- **Autentifikacija:**
    - Proces provere identiteta subjekta.
    - Autentifikacija se sprovodi nakon identifikacije kako bi se utvrdilo da li je subjekt onaj koji tvrdi da jeste.
    - Autentifikacija je neophodna pre autorizacije.
- **Autorizacija:**
    - Utvrđivanje prava koje korisnik ima nad resursima u sistemu.
    - Zahteva uspešnu autentifikaciju subjekta.
    - Autorizacija se sprovodi nakon autentifikacije kako bi se odobrilo ili odbilo pristupanje resursima.
- **Učesnici u autentifikaciji:**
    - Prover (claimant): Onaj koji se predstavlja kao korisnik čiji se identitet autentifikuje.
    - Verifier (recipient): Onaj koji proverava identitet subjekta.
- **Protokol za autentifikaciju:**
    - Realizuje se u realnom vremenu kako bi se utvrdio identitet subjekta.
    - Cilj je potvrditi autentičnost korisnikovog identiteta.
- **Karakteristike protokola za autentifikaciju:**
    - Zanemarljiva verovatnoća da treći učesnik može da se predstavi kao provernik i izazove lažni pozitivan rezultat autentifikacije.
    - Verifier ne može koristiti informacije dobijene od provernika kako bi se predstavio kao provernik trećem učesniku.
- **2. Principi :**
- **Kombinacija metoda:**
    - Metode autentifikacije se često kombinuju radi jačeg osiguranja, npr. lozinke i tokeni ili biometrija.
- **Identifikacija vs. autentifikacija:**
    - Identifikacija je dodeljivanje ID-a, dok je autentifikacija provera identiteta.

    - Autentifikacija prethodi autorizaciji i podrazumeva identifikaciju.
- **Učesnici:**
    - Prover (claimant): Osoba koja se autentifikuje.
    - Verifier (recipient): Entitet koji proverava identitet.
- **Tipovi autentifikacije:**
    - **Nešto što znam:** Kao što su lozinke, ali podložno lažnom predstavljanju.
    - **Nešto što imam:** Kao što su fizički tokeni poput ključeva ili smart kartica, dodatna sigurnost u odnosu na lozinke.
    - **Nešto što jesam:** Biometrija, oslanja se na fizičke karakteristike, uključujući otisak prsta ili prepoznavanje lica.
    - **Nešto što radim:** Dinamička biometrija, uključujući glas, potpis, ili hod.
- **Klasifikacija šema autentifikacije:**
    - **Slabe šeme:** Jednostavne za implementaciju, ali podložne napadima.
    - **Jake šeme:** Zasnovane na kriptografskim tehnikama, pouzdanije, ali složenije.
- **3. Password-based authentication**
- **Najrašireniji metod:** Autentifikacija putem lozinki je najčešće korišćen metod.
- **Proces:** Korisnik se prijavljuje koristeći korisničko ime i lozinku, a server proverava njihovu validnost.
- **Problemi:**
    - **Nedostatak zaštite:** Ako se lozinke skladište nešifrovano, postoji rizik od neovlašćenog pristupa, što je posebno opasno ako im pristupaju administratori sistema.
    - **Koriste se loše prakse:** Neke organizacije čuvaju lozinke na papiru, što narušava sigurnost i neporecivost autentifikacije.
    - **Rizik od krađe:** Lozinke se mogu ukrasti promatranjem korisnika prilikom unosa ili prevarom legitimnih korisnika.
- **Rešenja i preporuke:**
    - **Edukacija korisnika:** Podizanje svesti korisnika o bezbednosti lozinki i praksama deljenja istih.
    - **Korišćenje menadžera lozinki:** Korišćenje alata za upravljanje lozinkama može pomoći u generisanju i čuvanju bezbednih lozinki.
    - **Korišćenje sertifikata:** Umesto lozinki, mogu se koristiti sertifikati za autentifikaciju, čime se eliminišu mnogi rizici i napadi zasnovani na lozinkama.
- **Zaštita od napada:**
    - **Napadi rečnikom:** Mitigacija ovih napada može se postići korišćenjem salt-a i hešovanja lozinki.
    - **Napadi sa ponovljenim porukama i uvođenjem kašnjenja:** Ovi napadi zahtevaju dodatne mere kao što su dodavanje slučajnih brojeva i redukovani vremenski prozori.
    - **Čitanje komunikacije (sniffing):** Korišćenje bezbednih protokola za komunikaciju može sprečiti otkrivanje lozinki tokom prenosa.
- **4. Challenge-Response šema za autentifikaciju:**
- **Princip:** Korisnik potvrđuje svoj identitet demonstrirajući poznavanje tajne informacije, pri čemu tajna informacija nije otkrivena sistemu.
- **Proces:**
    - Sistem šalje korisniku podatak koji se menja tokom vremena, kao što su timestamp, slučajan broj ili broj sekvence.
    - Korisnik izračunava odgovor na osnovu dobijenog podatka i tajne informacije.
    - Ako je odgovor ispravan, sistem smatra korisnika autentičnim.
- **Implementacija:**
    - Može se realizovati korišćenjem kriptosistema sa tajnim ključem, javnim ključem ili zero-knowledge tehnikom.
    - **Challenge-Response sa tajnim ključem:**
        - Korisnik i sistem dele tajni ključ i unapred dogovoreni simetrični algoritam.

        - Sistem šalje korisniku slučajan broj, koji korisnik šifruje tajnim ključem i vraća sistemu.
        - Sistem proverava da li je primljeni broj jednak originalnom.
    - **Challenge-Response sa javnim ključem:**
        - Sistem generiše slučajan broj, računa njegov heš, šifruje ga zajedno sa proizvoljnom porukom korisnikovim javnim ključem i šalje korisniku.
        - Korisnik dešifruje poruku, izračunava heš i poredi ga sa primljenim.
- **Mitigacija napada:**
    - Korišćenje javnog ključa za šifrovanje informacija vezanih za lozinku na serveru sprečava napade poput sniffinga.
    - Protokoli kao što je Kerberos omogućavaju autentifikaciju bez prenošenja lozinke.
- **Alternativni pristupi:**
    - **Jednokratne lozinke:** Za svako prijavljivanje se koristi nova lozinka iz unapred definisane liste.
    - **SKEY postupak:** Korisnik unosi slučajan broj, a sistem generiše i šalje niz brojeva koje korisnik koristi za autentifikaciju.
- Osim toga, važno je obratiti pažnju na preporuke za jačanje sigurnosti lozinki, kao što su definisanje minimalne kompleksnosti lozinke i korišćenje bezbednosnih pravila poput onih preporučenih od strane Centra za Internet bezbednost.
- **5. Token-based autentifikacija**
- Autentifikacija zasnovana na tokenima oslanja se na nešto što korisnik poseduje, a što se pretpostavlja da nijedan drugi korisnik nema ili ne može da pridobije. Postoji nekoliko načina za postizanje ove vrste autentifikacije:
- **Kartice za pristup:**
    - Mogu koristiti optički bar kod, magnetnu traku ili čipove koji čuvaju biometrijske podatke.
    - Kartice sa magnetnim trakama imaju svoje mane, kao što su ograničenje kapaciteta i podložnost kopiranju ili brisanju podataka.
- **Smart kartice i dongle uređaji:**
    - Koriste se za čitanje putem PC čitača kartica ili USB portova.
    - Smart kartice imaju sopstvenu mogućnost obrade i često čuvaju tajni ključ povezan sa korisnikom, što omogućava dvofaktorsku autentifikaciju uz PIN ili lozinku.
- **Softverski tokeni:**
    - Alternativa hardverskim tokenima i obično su jeftinija opcija.
    - Mogu biti ranjivi na dictionary napade, ali se noviji soft tokeni oslanjaju na kriptografiju sa javnim ključem za dodatnu sigurnost.
- **One-time password (OTP) generatori:**
    - Generišu jednokratnu kombinaciju cifara koja se menja svaki put kada istekne vreme ili kada se pritisne fizičko dugme na uređaju.
    - Često se koriste kao drugi faktor za dvofaktorsku autentifikaciju i zasnovani su na zajedničkim tajnim ključevima između servera i tokena.
- **Autentifikacija pomoću mobilnih uređaja:**
    - Koristi mobilne telefone ili tablete koji mogu primiti SMS poruke ili instalirati specijalne aplikacije za generisanje kodova.
    - Ovo se često koristi kao drugi faktor za dvofaktorsku autentifikaciju.
- **6. Biometrijska autentifikacija**
    - Biometrijska autentifikacija predstavlja automatsku identifikaciju osobe na osnovu njenih fizioloških karakteristika ili ponašanja. Dok drugi oblici autentifikacije, poput tokena ili lozinki, autentifikuju korisnika na osnovu nečega što korisnik poseduje ili zna, biometrija omogućava autentifikaciju na osnovu toga ko je korisnik.
- **Zahtevi biometrije:**
    - **Univerzalnost (Universality):** Svaka osoba treba da ima biometrijske karakteristike.

    - **Jedinstvenost (Uniqueness):** Dve osobe ne bi trebalo da budu iste u pogledu biometrijskih karakteristika.
    - **Permanentnost (Permanence):** Biometrijski podaci bi trebalo da budu relativno nepromenljivi u određenom vremenskom periodu.
    - **Naplativost/Prikupljivost (Collectability):** Biometrijska karakteristika bi trebalo da se može izmeriti.
- **Primarna upotreba biometrije:**
    - **Logički pristup sistemu:** Biometrija zamenjuje ili dopunjuje PIN-ove, lozinke i tokene prilikom pristupa računarima, mrežama ili nalozima.
    - **Fizički pristup sistemu:** Biometrija omogućava kontrolu pristupa prostorijama, trezorima itd., često se koristi na aerodromima i graničnim kontrolama.
    - **Obezbeđivanje jedinstvenosti korisnika:** Koristi se za sprečavanje dvostrukog upisa u programe socijalnih davanja ili programe beneficija zaposlenih.
    - **Sistem javne identifikacije:** Koristi se za identifikaciju kriminalaca i terorista.
- **Tipovi biometrijskih karakteristika:**
    - Otisak prsta
    - Prepoznavanje lica
    - Prepoznavanje geometrije šake
    - Skeniranje irisa (dužice)
    - Prepoznavanje glasa
- **Tipovi grešaka:**
    - **False Accept (false positive):** Verovatnoća da će prevarant biti uparen sa validnom biometrijom korisnika.
    - **False Reject (false negative):** Verovatnoća da će šablon koji je kreiran na osnovu prezentovanih podataka korisnika biti pogrešno procenjen da ne odgovara njegovom šablonu iz baze.
    - **Crossover Error Rate (CER) ili Equal Error Rate (EER):** Predstavlja presek prva dva. Niži CER ukazuje na tačniji biometrijski uređaj.
    - **Failure To Enroll (FTE):** Situacija u kojoj pojedinac nije u mogućnosti da upiše svoju biometriju kako bi se napravio odgovarajući šablon.
- **Cena tehnologija za autentifikaciju:** Iako početni troškovi biometrijskih sistema mogu biti visoki, implementacija obično rezultuje nižim administrativnim troškovima od drugih pristupa, kao što su troškovi upravljanja lozinkama. Različite tehnologije, poput čitača otiska prsta ili sistema za prepoznavanje glasa, imaju različite cene, ali je opšte mišljenje da su lozinke, ipak, "skuplje" u pogledu administrativnih troškova na duži rok.

4. KONTROLA PRISTUPA

- **1. Kontrola pristupa**
    - Kontrola pristupa je bezbednosni mehanizam prisutan u svim delovima informacionih sistema. Počevši od prvog bezbednog sistema za računanje, registra kase iz 1879. godine, gde je fioka otvarana samo prilikom unosa iznosa, do savremenih tehnika, kontrola pristupa igra ključnu ulogu u zaštiti informacija.
- **Rizici po bezbednost informacija:**
    - **Poverljivost (Confidentiality):** Čuvanje podataka od neovlašćenog čitanja.
    - **Integritet (Integrity):** Čuvanje podataka od izmena.
    - **Dostupnost (Availability):** Informacije su dostupne u trenutku kada su i potrebne.
- **Razvoj mehanizama kontrole pristupa:**
    - Početkom 1970-ih su započeti prvi radovi.
    - Standardizacija je usledila početkom 1980-ih.
    - Koncepti poput Role-based Access Control (RBAC) su razvijeni početkom 1990-ih.
- **2. Koncepti kontrole pristupa:**
    - **Korisnik (user):** Osoba koja koristi informacioni sistem.

    - **Subjekat (subject):** Računarski proces koji obavlja zadatke za korisnika.
    - **Objekat (object):** Resurs informacionog sistema koji je dostupan korisniku.
    - **Operacija (operation):** Aktivan proces pokrenut od strane subjekta.
    - **Dozvola (permission):** Dopuštenje da se obavi određena operacija u okviru sistema.
- **Minimalne privilegije (least privilege):**
    - Selektivno dodeljivanje dozvola korisnicima tako da nemaju više privilegija nego što je minimalno neophodno za obavljanje njihovog posla.
    - Određivanje minimalnih privilegija je administrativni zadatak koji uključuje identifikaciju funkcija vezanih za radno mesto ili korisnika, specifikaciju dozvola potrebnih za obavljanje svake funkcije i restrikciju korisnika na određeni domen uz dodeljene privilegije.
    - Ovaj princip zahteva striktno pridržavanje kako bi se minimizirali rizici.
- Kontrola pristupa je dinamička i evolutivna oblast koja se neprekidno prilagođava kako bi se održala sigurnost informacionih sistema u promenljivom okruženju.
- **3. Politika kontrole pristupa**
- **Politika:**
    - Definiše zahteve visokog nivoa koji određuju ko može pristupiti čemu i pod kojim uslovima.
    - Može se definisati za različite aplikacije, ali često je deo realnog sistema, poput finansijskih institucija, vojnih ili zdravstvenih organizacija.
    - Podleže promenama tokom vremena, odražavajući promene u organizacionoj strukturi i načinu rada.
- **Mehanizmi kontrole pristupa:**
    - Sprovode politiku kontrole pristupa.
    - Zahtevaju čuvanje bezbednosnih atributa o korisnicima i resursima.
    - Primeri atributa korisnika su identifikacioni brojevi, grupe i uloge, dok su atributi resursa nivoi osetljivosti i pristupne liste.
    - Mehanički procesi kontrole pristupa mogu uključivati poređenje vrednosti bezbednosnih atributa ili poklapanje u bezbednosnim atributima.
- **4. Mehanizmi kontrole pristupa - referentni monitor:**
    - Predstavlja apstraktni pogled na podsistem za kontrolu pristupa.
    - Služi kao vodič za dizajn, implementaciju i analizu bezbednih IT sistema.
    - Zahtevaće da se uvek pozove, ne može ga se zaobići, mora biti otporan na neovlašćene izmene i njegova korektna implementacija mora biti proveriva.
- **Tri dodatna uslova za sistem kontrole pristupa:**
    - **Fleksibilnost:** Sistem mora podržati politiku kontrole pristupa u organizaciji.
    - **Upravljivost:** Sistem mora biti jednostavan za korišćenje i upravljanje.
    - **Skalabilnost:** Sistem mora efikasno funkcionisati i za veliki broj korisnika i resursa.
- Implementacija politike kontrole pristupa na sistemu često zahteva dodatne napore u prilagođavanju kupljenih referentnih monitora ili mehanizama kontrole pristupa kako bi odgovarali specifičnim zahtevima organizacije.
- **5. Modeli kontrole pristupa**
    - Modeli kontrole pristupa su apstraktne strukture koje se zasnivaju na konceptima kontrole pristupa i pružaju jednostavniji i standardizovan pristup implementaciji kontrole pristupa. Evo pregleda nekih od najpoznatijih modela:
- **Lampson model**:
    - Koristi "matricu pristupa" koja ima redove za subjekte i kolone za objekte.
    - Operacije su definisane kao skupovi dozvoljenih operacija.
    - Dva česta mehanizma za implementaciju su liste sposobnosti i liste kontrola pristupa.
- **Bell-LaPadula model**:
    - Formalizuje vojne principe kontrole pristupa.
    - Definiše objekte kao dokumente sa različitim nivoima poverljivosti i korisnike sa različitim nivoima pristupa.

    - Osnovna pravila uključuju "No read up" i "No write down".
    - Ima nedostatak u odlučivosti, što znači da ne može pouzdano garantovati da će konfiguracija ostati ispravna.
- **TCSEC standard**:
    - Definiše standarde za kontrolu pristupa u okviru američkog Ministarstva odbrane.
    - Uključuje Discretionary Access Control (DAC) i Mandatory Access Control (MAC) modele.
- **Discretionary Access Control (DAC)**:
    - Vlasnici objekata dodeljuju prava pristupa subjektima.
    - Korisnici mogu delegirati prava drugim korisnicima.
    - Model nije odlučiv.
- **Mandatory Access Control (MAC)**:
    - Baziran je na Bell-LaPadula modelu.
    - Subjektima i objektima su dodeljeni bezbednosni nivoi.
    - Model je odlučiv i koristi se za višenivojske sisteme.
- **Biba model**:
    - Fokusira se na integritet podataka umesto na poverljivost.
    - Uvodi ograničenja za čitanje i pisanje bazirana na nivoima integriteta.
- **Clark-Wilson model**:
    - Fokusira se na integritet u komercijalnim primenama.
    - Obezbeđuje dobro formirane transakcije i razdvajanje zaduženja kako bi se osigurala konzistentnost podataka.
- **Politika kineskog zida**:
    - Cilj je sprečiti protok podataka koji mogu izazvati sukob interesa, posebno u finansijskim konsultacijama.
- **Brewer-Nash model**:
    - Definiše pravila za čitanje i pisanje podataka na osnovu organizacija i skupova podataka.
- **Domain Type Enforcement model**:
    - Subjektima se dodeljuju domeni, a objektima tipovi.
    - Dozvole su vezane za domene i tipove.

5. KONTROLA PRISTUPA – ROLE BASED ACCESS CONTROL

- **1. Uvod:**
- Role Based Access Control (RBAC) predstavlja model kontrole pristupa koji se zasniva na ulogama ili radnim mestima u organizaciji. Ovaj model je razvijen krajem 1980-ih i početkom 1990-ih kao odgovor na nedostatke diskrecionih i mandatornih modela kontrole pristupa.
- **Osnovni koncepti RBAC-a**:
    - **Uloga (role)**: Predstavlja radno mesto ili funkciju koju korisnik ima u organizaciji.
    - **Dodela uloga**: Subjekti (korisnici) imaju aktivne uloge koje im omogućavaju izvršavanje određenih transakcija.
    - **Autorizacija uloga**: Subjekti mogu koristiti samo uloge koje su im autorizovane.
    - **Autorizacija transakcija**: Subjekti mogu izvršavati transakcije samo ako su te transakcije autorizovane za njihove aktivne uloge.
- **Administracija u RBAC-u**:
    - Administracija u RBAC-u je pojednostavljena u odnosu na tradicionalne modele kontrole pristupa. Ovo se postiže centralizacijom veza između korisnika, uloga i dozvola.
    - Uloga se definiše kao skup dozvola potrebnih za obavljanje određenog posla ili zadatka na radnom mestu.
    - Centralizovana administracija se često sprovodi na serveru gde su definisane uloge, dozvole i korisnici.
- **Prednosti RBAC-a**:

    - **Smanjenje kompleksnosti**: Centralizovana administracija smanjuje broj veza koje se moraju održavati za kontrolu pristupa, što olakšava upravljanje.
    - **Jednostavnija administracija promena**: Kada korisnik promeni poziciju na poslu, samo se veza korisnik-uloga menja, dok ostale veze ostaju nepromenjene.
    - **Sprečavanje konflikta interesa**: RBAC omogućava efikasno sprečavanje konflikta interesa pravilnim dodeljivanjem uloga i dozvola.
- **Varijante RBAC-a**:
    - **RBAC0**: Osnovni elementi RBAC sistema.
    - **RBAC1**: Dodatak RBAC0-u koji uključuje hijerarhije uloga.
    - **RBAC2**: Dodatak RBAC0-u koji uključuje ograničenja kao što je razdvajanje zaduženja (Separation of Duties - SoD).
    - **RBAC3**: Kombinacija RBAC0, RBAC1 i RBAC2.
- RBAC predstavlja moćan model kontrole pristupa koji se često koristi u organizacijama kako bi se efikasno upravljalo pravima korisnika u skladu sa njihovim ulogama i odgovornostima.
- **2. Osnovni model RBAC-a** sastoji se od pet osnovnih koncepata:
    - **Korisnici**: Predstavljaju entitete (osobe, procesi ili aplikacije) koji zahtevaju pristup resursima sistema.
    - **Uloge**: Definišu skup privilegija ili ovlašćenja koje korisnici mogu imati. Uloge se dodeljuju korisnicima u skladu sa njihovim radnim zadacima ili odgovornostima.
    - **Dozvole**: Predstavljaju operacije koje su dozvoljene nad određenim objektima sistema. Dozvole se sastoje od konkretnih operacija koje korisnici mogu izvršavati nad određenim resursima.
    - **Veze**: Veze između korisnika, uloga i dozvola su mnogo-na-mnogo (n:m). To znači da korisnik može imati više uloga, uloga može imati više korisnika, uloga može imati više dozvola, i dozvola može biti u više uloga.
    - **Granularnost dozvola**: Dozvole se mogu fino podešavati prema potrebama sistema. Svaka dozvola može biti posmatrana kao atomična operacija u sistemu, što omogućava preciznu kontrolu pristupa.
- Primeri upotrebe ovih koncepata u bankarskom sistemu:
    - **Šalterski radnik**: Može izvršavati transakcije kao što su isplata novca (withdraw) ili uplata novca na račun (deposit). Potrebna su mu prava za čitanje i pisanje (read i write) nad podacima o računima. Nakon obavljanja transakcije, ne sme vršiti izmene na podacima.
    - **Supervizor**: Može ispravljati rezultate transakcija, takođe uz prava za čitanje i pisanje nad podacima o računima. Međutim, ne može samostalno izvršavati transakcije kao što su withdraw ili deposit.
- U RBAC-u, veza između korisnika i subjekta je jedan-na-mnogo (1:n). Subjekat predstavlja tipično aktivni entitet kao što je program, a vezuje se za jednu sesiju. Korisnik može imati više istovremenih subjekata, što omogućava dinamičku komponentu RBAC-a gde korisnik može aktivirati podskup mogućih uloga radi ispunjenja principa minimalnih privilegija.
- Implementacija RBAC-a zavisi od korišćene tehnologije i sistema. Koncepti RBAC-a se moraju mapirati na koncepte sistema, kao što su prava pristupa fajlovima, korisnici, grupe itd. Ovo mapiranje omogućava da se apstraktni model RBAC-a konkretizuje u stvarne ACL (Access Control List) liste na nivou sistema.
- **3. Hijerarhijski RBAC** (Role-Based Access Control) uvodi dodatnu složenost u osnovni model dodavanjem hijerarhije uloga. Ovo omogućava da uloge nasleđuju dozvole od drugih uloga, što olakšava administraciju sistema i odražava organizacionu strukturu, raspodelu zaduženja i odgovornosti unutar organizacije.
- Glavne karakteristike hijerarhijskog RBAC-a su:
    - **Hijerarhija uloga**: Uloge su organizovane u hijerarhijsku strukturu gde uloge na višim nivoima nasleđuju dozvole od uloga na nižim nivoima. Na primer, ako uloga A nasleđuje ulogu B, tada sve dozvole koje ima uloga B pripadaju i ulozi A.
    - **Složenija administracija, ali dugoročne prednosti**: Iako zahteva složeniju pripremu, korišćenje hijerarhije uloga se isplati na duži rok kroz jednostavniju administraciju. Hijerarhijska struktura olakšava upravljanje dozvolama i korisnicima u sistemu.

    - **Motivacija za formiranje hijerarhije uloga**: Uloge u organizaciji često imaju preklapajuće funkcije, a hijerarhijska struktura pomaže u organizovanju tih funkcija i pojednostavljuje upravljanje pristupom.
- Postoje različite šeme nasleđivanja u hijerarhijskom RBAC-u:
    - **Direktno nasleđivanje dozvola**: U ovom modelu, uloga je imenovani skup dozvola, i uloga r2 nasleđuje ulogu r1 ako je skup dozvola r1 podskup skupa dozvola r2.
    - **Nasleđivanje dozvola i korisnika**: U ovom modelu, uloga obuhvata i dozvole i korisnike. Uloge na vrhu hijerarhije imaju više dozvola i manje korisnika, dok uloge na nižim nivoima imaju manje dozvola i više korisnika.
    - **Zadržavanje korisnika i indirektno nasleđivanje dozvola**: U ovom modelu, dozvole se dodeljuju grupama korisnika, a grupe se mapiraju na uloge. Administracija se svodi na upravljanje relacijama nad korisnicima.
- Dodatno, hijerarhijski RBAC može uključivati ograničene hijerarhije koje omogućavaju jednostruko nasleđivanje i opšte hijerarhije koje dozvoljavaju višestruko nasleđivanje. Ograničene hijerarhije su češće u komercijalnim proizvodima jer su lakše za implementaciju i upravljanje.
- Uvod u tipove uloga, kao što su uloge sa kvalifikatorima kao što su lokacija, organizaciona jedinica, region itd., omogućava dodatnu fleksibilnost u organizaciji i upravljanju pristupom resursima. Na primer, možemo imati globalne uloge koje imaju pristup svim podacima o kreditima, dok lokalne uloge imaju pristup samo određenim regionalnim podacima. Ograničene hijerarhije često podržavaju ove tipove uloga.
- **4. RBAC sa ograničenjima**, posebno razdvajanje zaduženja (SoD), donosi dodatne složenosti u upravljanju pravima pristupa. Ova vrsta RBAC-a je posebno korisna u situacijama gde je važno osigurati da kritične operacije zahtevaju prisustvo više osoba kako bi se povećala bezbednost sistema. Evo nekoliko ključnih pojmova i pristupa u RBAC-u sa ograničenjima:
    - **Razdvajanje zaduženja (SoD)**: Ovo je princip koji zahteva da određene kritične operacije obavljaju dva ili više različitih lica, čime se eliminiše zavisnost o jednoj osobi i povećava sigurnost sistema.
    - **Statički SoD**: Ograničenja se postavljaju u trenutku kada se korisniku dodeli uloga. Na primer, korisniku može biti zabranjeno da istovremeno ima dodeljene određene uloge.
    - **Dinamički SoD**: Ograničenja se primenjuju u toku korišćenja sistema. Na primer, korisnik ne sme imati istovremeno aktivne uloge koje su međusobno isključive.
    - **SoD baziran na objektima**: Ograničenja se primenjuju ne samo na uloge ili korisnike, već i na konkretne objekte u sistemu. Na primer, jedna osoba ne sme imati uloge koje omogućavaju pristup istom objektu.
    - **Međusobno isključivanje uloga**: Ovaj pristup koristi skupove uloga koje se međusobno isključuju. Na primer, jedna uloga može onemogućiti dodeljivanje druge uloge korisniku.
    - **Pravila za bezbedno dodeljivanje privilegija**: Ova pravila osiguravaju da nijedna uloga ne sadrži sve dozvole potrebne za izvršenje određene kritične operacije, čime se garantuje da SoD ostaje očuvan.
- RBAC sa ograničenjima omogućava precizniju kontrolu pristupa u sistemu, posebno u situacijama gde je važno sprečiti konflikte interesa ili minimizirati rizike vezane za sigurnost. Ovi pristupi se često koriste u bankarskoj industriji, vojsci, kao i u drugim oblastima gde je sigurnost od ključnog značaja.
- **5. Attribute Based Access Control (ABAC)** je model koji definiše ovlašćenja na osnovu atributa subjekata, objekata i okruženja. Ovaj model kontroliše pristup objektima procenjujući pravila u odnosu na kombinaciju atributa subjekta i objekta, operacija i okruženja relevantnog za zahtev. Evo detaljnijeg pregleda ključnih elemenata ABAC modela:
    - **Atributi**: Atributi predstavljaju karakteristike subjekata, objekata i okruženja koje se koriste za donošenje odluka o pristupu. Oni mogu obuhvatiti različite informacije kao što su identitet subjekta, identitet objekta, organizacija, vreme pristupa, lokacija, nivo bezbednosti i drugi relevantni podaci.
    - **Model arhitekture**: Arhitektura ABAC modela uključuje mehanizme kontrole pristupa koji evaluiraju pravila na osnovu atributa subjekata, objekata i okruženja kako bi odredili dozvoljeni pristup. Ovi mehanizmi rade na osnovu definisanih polisa koje regulišu pristup informacijama.

- **Model polisa**: Polise definišu pravila i odnose koji regulišu dozvoljeno ponašanje unutar organizacije u vezi sa pristupom resursima. Ove polise se sastoje od pravila koja opisuju uslove pod kojima je pristup dozvoljen ili odbijen na osnovu atributa subjekata, objekata i okruženja.

- ABAC model omogućava fleksibilniju kontrolu pristupa u odnosu na RBAC (Role Based Access Control) model, jer ne zahteva predefinisane uloge. Umesto toga, pristup se određuje dinamički na osnovu kombinacije atributa subjekta, objekta i okruženja. To omogućava preciznije kontrolisanje pristupa u složenim okruženjima gde su potrebna detaljnija pravila za kontrolu pristupa.
- Na primer, u ABAC modelu, umesto da se korisnicima dodeljuju fiksne uloge poput "Odrasla osoba" ili "Dete", pristup se određuje na osnovu atributa kao što su uzrast korisnika i ocena sadržaja koji želi da pristupi. Ovo omogućava dinamičko upravljanje pravima pristupa u skladu sa promenljivim uslovima i potrebama organizacije.

6. STANDARDI ZA AUTENTIFIKACIJU

- **1. X.509 Authentication Service**:
  - Hijerarhijski direktorijumski servis X.500 koristi se za čuvanje informacija o korisnicima u svrhu autentifikacije.
  - Informacije o korisnicima smeštene su u sertifikate potpisane tajnim ključem treće strane kojoj svi veruju.
  - Definiše tri tipa autentifikacionih procedura: One-way authentication, Two-way authentication i Three-way authentication.
  - One-way i two-way autentifikacija se koriste na webu u okviru SSL protokola.
  - **One-way authentication**: Provera identiteta provera i slanje relevantnih informacija verifieru radi autentifikacije.
  - **Two-way authentication**: Obostrana autentifikacija koja omogućava proveri da potvrdi svoj identitet.
  - **Three-way authentication**: Uključuje i treću poruku kojom se sinhronizuje časovnik.
- **2. HTTP autentifikacija**:
  - Postoje dva tipa autentifikacije po HTTP standardu: Basic Authentication i Digest Access Authentication.
  - **HTTP Basic Authentication**:
    - Klijenti se identifikuju na osnovu korisničkog imena i lozinke.
    - Klijent šalje korisničko ime i lozinku kodirane Base64 algoritmom u zaglavlju HTTP zahteva.
    - Nizak nivo zaštite, jer lozinke putuju kao otvoreni tekst preko mreže.
  - **HTTP Digest Access Authentication**:
    - Korisničko ime i lozinka se ne šalju preko mreže, već samo njihov hash kod.
    - Radi po challenge-response principu, gde server šalje kodiranu informaciju, a klijent odgovara sa korisničkim imenom, lozinkom i kodiranom informacijom.
    - Lozinka nije otkrivena prisluškivanjem, ali preneti sadržaji i dalje nisu zaštićeni od prisluškivanja.
- Ove metode autentifikacije se koriste za obezbeđivanje pristupa resursima preko HTTP protokola, pri čemu Digest Access Authentication pruža viši nivo bezbednosti u odnosu na Basic Authentication.
- **3. NTLM (NT LAN Manager)**:
- **Verzije**:
  - **Lan Manager (LM)**: Koristi slab metod za heširanje lozinki.
  - **New Technology Lan Manager (NTLMv1)**: Koristi DES-ECB za generisanje challenge-a, ali je poboljšan u odnosu na prethodnu verziju.
  - **NTLMv2**: Podržava bolje kriptografske algoritme poput HMAC-MD5 i dizajniran je da poboljša bezbednost u odnosu na prethodne verzije.
- **Tok komunikacije**:
  - Korisnik unosi naziv domena, korisničko ime i lozinku.
  - Računar izračunava hash lozinke i odbacuje originalnu lozinku.
  - Korisnik šalje korisničko ime serveru.
  - Server generiše nasumični broj (challenge) i šalje ga klijentu.

    - Klijent šifruje challenge hash-om korisničke lozinke i šalje rezultat serveru (response).
    - Domen kontroler koristi korisničko ime da dobije hash korisnikove lozinke i koristi ga da šifruje challenge. Potom upoređuje šifrovani challenge sa odgovorom klijenta.
    - Ako su isti, korisnik je autentifikovan i domen kontroler obaveštava server o tome.
- **LM**:
    - Koristi ANSI printabilne karaktere za formiranje lozinke.
    - Lozinke duže od 7 karaktera se dele na dva dela i svaki deo se hešuje odvojeno.
    - LM hash ne sadrži salt.
- **NTLMv1**:
    - Računa MD4 hash lozinke, zatim deli hash na tri dela i šifruje ih DES algoritmom zajedno sa challenge porukom.
    - NTLM hash postaje isto što i lozinka, omogućavajući napadima poput "pass-the-hash".
- **NTLMv2**:
    - Dizajniran je da eliminiše mogućnost dictionary napada.
    - Koristi klijentski nonce uz serverski challenge tokom generisanja odgovora.
    - Dodatni nonce menja veličinu odgovora.
    - Koristi HMAC-MD5 za generisanje odgovora, što povećava bezbednost u odnosu na NTLMv1.
- NTLM je bio značajan sistem za autentifikaciju u Windows okruženju, ali se danas preporučuje upotreba modernijih i sigurnijih protokola poput Kerberos-a ili NTLMv2 radi poboljšanja bezbednosti sistema.
- **4. Kerberos**:
- **Povijest**:
    - Nastao je na MIT-u 1980-ih.
    - Verzije 1-3 su se koristile samo interno na MIT-u.
    - Verzija 4 je bila u javnoj upotrebi.
    - Verzija 5 je ispravila neke nedostatke i postala je široko rasprostranjena, posebno u Windows okruženju.
- **Funkcionalnosti**:
    - Omogućava single sign-on (SSO), što znači da korisnik mora da se prijavi samo jednom, a potom ima pristup svim resursima na mreži u skladu sa svojim pravima.
    - Efikasno upravljanje velikim brojem korisničkih naloga.
    - Lozinke se nikad ne šalju kao otvoreni tekst.
- **Centar za distribuciju ključeva (KDC)**:
    - Sastoji se od tri glave: baza podataka, server za proveru identiteta i server za izdavanje karata.
    - Baza podataka čuva proverene parametre svih učesnika Kerberos sistema.
    - Kerberos Principal je jednoznačni identifikator učesnika, a sastoji se od identity/instance@realm.
    - KDC izdaje TGT kartu svim klijentima prilikom prijave na sistem.
- **Kerberos KDC**:
    - Sastoji se iz tri komponente: baza principala, server za autentifikaciju (AS) i server za dodelu karata (TGS).
    - AS izdaje TGT kartu svim klijentima prilikom prijave na sistem.
    - TGS je zadužen za izdavanje ST karata namenjenih pojedinim resursima.
- **Ticket-granting ticket (TGT)**:
    - Sadrži ime principala koji traži pristup, ime principala kome se želi pristupiti, timestamp, rok važenja karte i tajni ključ za komunikaciju sa resursom.
- **Tok komunikacije**:
    - KRB_AS_REQ zahtev.
    - KRB_AS_REP odgovor.
    - KRB_TGS_REQ zahtev.
    - KRB_TGS_REP odgovor.
- **Tipovi ticketa**:

    - Renewable Ticket.
    - Post Dated Ticket.
    - Proxiable Ticket.
    - Forwardable Ticket.
- **Cross Realm Authentication**:
    - Omogućava autentifikaciju između organizacija.
- **Bezbednost**:
    - Koristi se zaštita podataka kako bi se sprečile različite vrste napada poput pass-the-ticket, silver ticket i golden ticket napada.
- Kerberos je sistem za autentifikaciju koji se široko koristi u Windows okruženju i omogućava bezbedan pristup mrežnim resursima uz efikasno upravljanje pravima korisnika.

7. BEZBEDNA KOMUNIKACIJA U TCP/IP MREŽAMA

- **1. TCP/IP stek i bezbednost:**
- Sigurna komunikacija može se postići na različitim nivoima TCP/IP steka, pri čemu su potrebne usluge kao što su poverljivost, neporecivost, integritet, autentifikacija, autorizacija i upravljanje ključevima.
- **Implementacija bezbednosti na aplikativnom nivou**:
    - Dobre osobine:
        - Implementacija se vrši na krajnjim tačkama komunikacije, tj. računarima.
        - Aplikaciji nije potrebno oslanjati se na sigurnosne servise operativnog sistema.
        - Omogućava kompletni pristup podacima koji se štite.
    - Loše osobine:
        - Potrebna je implementacija za svaku aplikaciju posebno.
        - Može biti komplikovana izmena postojećih aplikacija.
        - Postoji velika verovatnoća pravljenja grešaka.
    - Primer: Pretty Good Privacy (PGP) za šifrovanje email poruka.
- **Implementacija bezbednosti na transportnom nivou**:
    - Dobre osobine:
        - Implementacija se vrši na krajnjim tačkama komunikacije, tj. računarima.
        - Ne zahteva modifikaciju svake aplikacije.
        - Omogućava kompletni pristup podacima koji se štite.
        - Sve aplikacije koriste isti stepen sigurnosti.
    - Loše osobine:
        - Zahteva (neznatne) izmene postojećih aplikacija radi korišćenja sigurnosnih usluga transportnog sloja.
    - Primer: Transport Layer Security (TLS) za sigurnu komunikaciju preko TCP protokola.
- **Implementacija bezbednosti na mrežnom nivou**:
    - Dobre osobine:
        - Zahteva još manje izmene u aplikacijama.
        - Svi transportni protokoli koriste istu infrastrukturu.
        - Omogućava pravljenje virtuelnih privatnih mreža (VPN).
    - Loše osobine:
        - Teško je obezbediti uslugu neporecivosti na ovom nivou.
        - Teško je obezbediti kontrolu na nivou korisnika na višekorisničkim računarima.
    - Primer: IP Security (IPSec) za sigurnu komunikaciju preko IP protokola.
- **Implementacija bezbednosti na nivou veze**:
    - Koristi se ako postoji namenska veza između dva uređaja na mreži, kao što su računari ili ruteri.
    - Dobre osobine:
        - Hardverski uređaj za šifrovanje omogućava visoku brzinu rada.

    - Loše osobine:
        - Samo je praktično za namenske veze gde su učesnici fizički povezani.
    - Primer: Veza bankomata sa centralom putem namenske veze.
- **2. PGP (Pretty Good Privacy):**
- PGP je protokol koji kombinuje simetrično i asimetrično šifrovanje, a namenjen je zaštiti elektronske pošte. Nastao je 1991. kao reakcija na predlog zakona koji bi omogućio pristup otvorenom tekstu svih poruka američkoj vladi. Glavne karakteristike PGP-a uključuju:
- **Asimetrično šifrovanje**: Ključ za šifrovanje od 128 bita se prenosi asimetričnim algoritmima poput ElGamala ili RSA.
- **Simetrično šifrovanje**: Koristi se za šifrovanje sadržaja poruka. Simetrični ključ se šifruje asimetričnim algoritmima i šalje primaocu.
- **Digitalno potpisivanje poruka**: Koristi se MD5 heš otvorenog teksta koji se potpisuje privatnim ključem pošiljaoca.
- **Repozitorijumi javnih ključeva**: PGP koristi repozitorijume javnih ključeva kao što su wwwkeys.pgp.net, keys.openpgp.org itd.
- **Poverenje u javne ključeve**: Koristi se "web of trust" za poverenje u javne ključeve umesto stroge hijerarhije sertifikata.
- **Bezbednost**: Bezbednost PGP-a zavisi od izbora simetričnog i asimetričnog algoritma, kao i heš funkcije. OpenPGP je rezultat standardizacije u okviru IETF-a (RFC 2440) i pruža standardne formate za šifrovane poruke, potpise i sertifikate.
- Otvorene implementacije PGP-a uključuju GnuPG, Enigmail plugin za Mozilla Thunderbird, KMail itd.
- **3. TLS (Transport Layer Security)** je kriptografski protokol koji omogućava sigurnu komunikaciju preko računarskih mreža. On je razvijen s ciljem podrške kriptografskoj bezbednosti, interoperabilnosti, proširivosti i relativne efikasnosti. TLS je evolucija SSL (Secure Sockets Layer) protokola, koji je prvobitno razvio Netscape.
- TLS se sastoji od nekoliko slojeva, uključujući Record Protocol i Handshake Protocol:
    - **TLS Record Protocol**: Ovaj protokol se oslanja na TCP i podržava protokole višeg nivoa. Koristi simetrične algoritme za šifrovanje podataka kako bi se osigurala poverljivost i integritet poruka. Takođe, vrši proveru integriteta poruka korišćenjem hash funkcija.
    - **TLS Handshake Protocol**: Ovaj protokol omogućava autentifikaciju klijenta i servera, kao i dogovor o korišćenim algoritmima i ključevima za šifrovanje. Sesija se uspostavlja putem razmene informacija o identitetu, sertifikatima, i generisanju sesijskog ključa. Cilj je osigurati da komunikacija ostane sigurna od prisluškivanja.
- TLS takođe sadrži protokole kao što su Alert Protocol, Change Cipher Spec Protocol i Heartbeat Protocol, koji se koriste za različite svrhe kao što su upozorenja o greškama, promena specifikacija šifrovanja i održavanje veze aktivnom.
- Neki od napada na SSL/TLS uključuju napade na Handshake protokol, napade na podatke i application data protokole, kao i napade na PKI (Public Key Infrastructure), što ukazuje na važnost stalnog nadgledanja i unapređenja sigurnosti ovih protokola.
- HTTPS (HTTP Secure) je protokol koji se oslanja na TLS kako bi omogućio sigurnu komunikaciju preko HTTP protokola. Koristi se za šifrovanje URL-ova, sadržaja dokumenata, informacija unetih u forme, kolačića (cookies) i zaglavlja HTTP zahteva. Razlika između HTTP i HTTPS je u tome što se HTTPS koristi za sigurnu komunikaciju preko SSL/TLS protokola, dok je HTTP nešifrovan i otvoren za prisluškivanje.
- **4. IPSec (Internet Protocol Security)** je skup proširenja IPv4 koji pruža zaštitu privatnosti, integriteta, provere identiteta i neporecivosti. On je integralni deo IPv6 i koristi se na mrežnom sloju TCP/IP steka.
- IPSec koristi sledeće komponente:
    - **Diffie-Hellman**: Za razmenu ključeva.
    - **Algoritme za digitalno potpisivanje**: Koristi se pri Diffie-Hellman razmeni ključeva radi potvrde identiteta učesnika i sprečavanja napada man-in-the-middle.
    - **Algoritme za šifrovanje**: Kao što su DES, 3DES i AES za šifrovanje podataka.

    - **Hash funkcije**: Kao što su MD5 i SHA, koje se koriste kao osnova za HMAC funkcije.
    - **Sertifikate koje potpisuje CA**: Za proveru identiteta učesnika u komunikaciji.
- IPSec sadrži dva nezavisna protokola:
    - **AH (Authentication Header)**: Pruža usluge integriteta, provere identiteta i neporecivosti. AH zaglavlje se smešta između IP zaglavlja i podataka koji slede. AH zaglavlje sadrži polja kao što su security parameters index, sequence number i authentication data.
    - **ESP (Encapsulated Security Payload)**: Pruža integritet, identitet, neporecivost i poverljivost podataka. ESP se smešta posle IP zaglavlja i enkapsulira sve podatke iz protokola višeg sloja. Polja u ESP zaglavlju uključuju security parameters index, sequence number, payload data, padding, next header i authentication data.
- Postoje dva režima rada IPSec-a:
    - **Transportni režim**: Šifruju se samo podaci u IP paketu, dok se IP zaglavlje ne menja. Ruteri vide source i destination IP.
    - **Tunelovanje**: Enkapsulira ceo IP paket u novi IP paket. To se koristi za sigurnu komunikaciju između dva gateway-a ili za formiranje VPN-a.
- Za uspostavljanje veze i razmenu ključeva, IPSec koristi protokole kao što su IKE (Internet Key Exchange), Photuris, i SKIP. IKE kombinuje protokole kao što su ISAKMP, Oakley, i SKEME, i sastoji se od dve faze: uspostavljanje IKE SA parametara i uspostavljanje IPSec SA parametara.
- IPSec je važan za sigurnost mrežne komunikacije, ali može imati uticaj na potrošnju resursa, uključujući procesorsko vreme i povećanje mrežnog saobraćaja.

8. OAuth 2.0

- **1. OAuth 2.0** je delegacioni protokol koji omogućava korisnicima da dozvole aplikacijama pristup njihovim resursima u njihovo ime. Ključni učesnici u komunikaciji u okviru OAuth 2.0 su:
    - **Vlasnik resursa (Resource Owner)**: Osoba koja ima pristup određenom resursu ili API-ju i koja može delegirati pristup tom resursu ili API-ju. Obično je to korisnik koji ima pristup web pretraživaču.
    - **Zaštićeni resurs (Protected Resource)**: Web servis (API) koji ima implementirane bezbednosne kontrole i štiti resurse vlasnika. Pruža pristup resursima na zahtev vlasnika.
    - **Klijentska aplikacija (Client)**: Aplikacija koja želi da pristupi zaštićenom resursu u ime vlasnika. Može biti web server ili nativna aplikacija.
    - **Autorizacioni server (Authorization Server)**: Server koji generiše tokene za klijenta, autentifikuje vlasnike resursa i klijente, i upravlja autorizacijom. Pruža mehanizme za autentifikaciju i autorizaciju korisnika i aplikacija.
- Proces komunikacije u OAuth 2.0 protokolu uključuje sledeće korake:
    - Vlasnik resursa ukazuje klijentskoj aplikaciji da želi da aplikacija izvrši određenu operaciju u njegovo ime.
    - Klijentska aplikacija zahteva autorizaciju od vlasnika resursa na autorizacionom serveru.
    - Vlasnik resursa daje prava klijentskoj aplikaciji.
    - Klijentska aplikacija dobija token od autorizacionog servera.
    - Klijentska aplikacija koristi dobijeni token da pristupi zaštićenom resursu.
- OAuth tokeni predstavljaju delegirana prava koja su dodeljena klijentu od strane vlasnika resursa. Autorizacioni server izdaje ove tokene, a klijent ih koristi za pristup zaštićenim resursima. Tokeni treba da budu formatirani na način koji je neprepoznatljiv klijentu i obično se koriste zaštite poput TLS-a za obezbeđivanje poruka i tokena.
- Važno je napomenuti šta OAuth 2.0 jeste i šta nije:
    - OAuth 2.0 je protokol koji definiše zajedničke koncepte i komponente za delegiranje prava aplikacijama, ali nije definisan van HTTP protokola. Takođe, on nije autentifikacioni protokol i ne definiše format tokena, kriptografske metode ili mehanizme za procesiranje informacija o autorizaciji. Umesto toga, on ostavlja ove detalje implementaciji sistema koji koristi OAuth 2.0.
- **2. OAuth 2.0 sekvenca**
- Ovo su koraci u OAuth 2.0 sekvenci:

    - **Korak 1:** Vlasnik resursa ukazuje klijentskoj aplikaciji da želi da aplikacija izvrši operaciju za njega. Na primer, vlasnik resursa može želeti da aplikacija učita slike sa servisa kako bi se mogle štampati. Klijentska aplikacija, shvatajući da je potreban OAuth token, šalje vlasnika resursa na autorizacioni server sa zahtevom za delegacijom prava.
    - **Korak 2:** Vlasnik resursa se autentifikuje. Autentifikacija se obavlja direktno sa autorizacionim serverom u web pretraživaču, a ne kroz klijentsku aplikaciju.
    - **Korak 3:** Vlasnik resursa autorizuje klijentsku aplikaciju. Vlasnik resursa bira koje delove odgovornosti će dati klijentskoj aplikaciji.
    - **Korak 4:** Autorizacioni server redirektuje korisnika nazad na klijentsku aplikaciju. Parametar **code** sadrži jednokratnu vrednost koja predstavlja rezultat autorizacije korisnika. Klijentska aplikacija proverava da li se parametar **state** podudara sa parametrom poslatim u prethodnom koraku.
    - **Korak 5:** Klijentska aplikacija pravi HTTP POST zahtev na autorizacioni server. Šalje **client_id** i **client_secret** kao HTTP Basic authorization zaglavlje, zajedno sa **code**. Ovaj zahtev se obavlja direktno između klijentske aplikacije i autorizacionog servera bez učešća web pretraživača i vlasnika resursa.
    - **Korak 6:** Autorizacioni server proverava klijentske kredencijale iz Authorization zaglavlja i vrednost **code** parametra. Ako je zahtev validan, autorizacioni server izdaje token.
    - **Korak 7:** Klijentska aplikacija može dalje koristiti token za pristup zaštićenom resursu. Kada šalje zahtev za pristup, uključuje token u Authorization zaglavlje.
- OAuth 2.0 koristi pojmove kao što su **scopes** i **refresh token**:
    - **Scopes** su stringovi koji predstavljaju dozvole koje token ima. Klijentska aplikacija može tražiti određene scope-ove, a vlasnik resursa ih odobrava.
    - **Refresh token** se takođe izdaje od strane autorizacionog servera i koristi se da se dobije novi token za pristup, bez potrebe da se uključuje vlasnik resursa u proces.