Ministerul Educaţiei, Culturii și Cercetării
Universitatea Tehnică a Moldovei

Faculatea Calculatoare, informatică și microelectronică
Departamentul Ingineria Software și Automatică

RAPORT

Lucrare de Laborator nr.3
Disciplina: Analiza și Specificarea Cerințelor Software
Tema :

A efectuat:
st. gr. SI-211 Chirita S.
A verificat:
asist. univ. Tocan A.

Chişinău 2024

Cuprins:

Introducere

În contextul actual, securitatea cibernetică a devenit o preocupare esențială, iar analiza traficului de rețea joacă un rol crucial în protejarea infrastructurilor informatice de atacuri și amenințări externe. Prin interceptarea și monitorizarea traficului de rețea, se poate identifica comportamente anormale, se pot depista vulnerabilitățile și se pot lua măsuri pentru prevenirea incidentelor de securitate.
Tema acestei lucrări se concentrează pe dezvoltarea unui **sistem integrat pentru interceptarea și analiza traficului de rețea**, care va permite monitorizarea și analiza detaliată a pachetelor de date ce circulă printr-o rețea de calculatoare. Acest sistem va fi capabil să captureze și să analizeze pachetele de date, să identifice posibile atacuri, precum și să furnizeze informații utile pentru a îmbunătăți securitatea rețelei.

1.1. Obiectivele laboratorului

Obiectivele laboratorului sunt următoarele:

1. **Dezvoltarea unui sistem de interceptare a traficului de rețea** – Crearea unui sistem capabil să captureze pachetele de date care circulă prin rețea și să le analizeze în timp real. Acesta va permite monitorizarea activității de rețea și detectarea unor posibile incidente de securitate.
2. **Implementarea tehnologiilor de analiză a traficului de rețea** – Utilizarea unor tehnici și instrumente pentru analiza pachetelor de date interceptate. Sistemul va identifica tiparele anormale sau suspecte de comportament, care pot indica prezența unui atac sau a unei vulnerabilități.
3. **Crearea unei interfețe grafice (GUI)** – Dezvoltarea unei interfețe prietenoase, care va permite utilizatorilor să vizualizeze în timp real informațiile despre traficul de rețea și să interacționeze cu sistemul pentru a analiza pachetele și a identifica eventualele riscuri.
4. **Testarea performanței și securității sistemului** – Efectuarea de teste pentru a evalua performanța sistemului în condiții de trafic intens și pentru a verifica eficiența în identificarea și gestionarea potențialelor amenințări. De asemenea, vor fi verificate măsurile de securitate implementate pentru protejarea datelor procesate.
5. **Analiza vulnerabilităților și a metodelor de protecție** – Identificarea eventualelor vulnerabilități ale sistemului de interceptare și analiza unor tehnici de protecție pentru prevenirea atacurilor cibernetice, inclusiv criptarea traficului și filtrarea pachetelor suspecte.
6. **Documentarea procesului de dezvoltare și implementare** – Redactarea unui raport detaliat privind pașii parcurși în realizarea sistemului, analiza tehnologiilor utilizate și evaluarea performanței acestuia.

1.2. Descrierea sarcinilor de laborator

În cadrul acestui laborator, vor fi realizate următoarele sarcini pentru dezvoltarea și implementarea sistemului integrat de interceptare și analiză a traficului de rețea:

1. **Cercetarea tehnologiilor de interceptare a traficului de rețea**
    - Analizarea celor mai utilizate metode de capturare a pachetelor de rețea (de exemplu, utilizarea bibliotecilor libpcap pentru interceptarea pachetelor).
    - Implementarea unui sistem de interceptare a traficului de rețea care să poată captura și stoca pachetele de date într-un format ușor accesibil pentru procesare ulterioară.
2. **Analiza pachetelor de rețea**
    - Dezvoltarea unui algoritm de procesare a pachetelor pentru a analiza tipologia traficului și pentru a identifica protocoalele utilizate, sursele și destinațiile acestora.
    - Implementarea unui sistem de clasificare a pachetelor (ex: HTTP, TCP, UDP, ICMP) și analiza conținutului acestora pentru a identifica potențialele amenințări sau comportamente suspecte.
3. **Implementarea unei interfețe grafice (GUI)**
    - Crearea unei interfețe vizuale pentru utilizatori care să permită vizualizarea în timp real a traficului de rețea, analiza pachetelor capturate și generarea unor rapoarte pentru a evidenția posibilele riscuri de securitate.
    - Implementarea funcționalităților de filtrare și căutare a pachetelor pe baza diverselor criterii (tip protocol, adresă IP, porturi, etc.).
4. **Testarea și optimizarea sistemului**
    - Realizarea unor teste de performanță pentru a evalua impactul asupra resurselor sistemului (CPU, memorie, lățime de bandă) și a verifica eficiența în procesarea pachetelor într-un timp rezonabil.
    - Optimizarea codului pentru a îmbunătăți performanța și scalabilitatea sistemului.
5. **Securizarea traficului și protecția datelor**
    - Implementarea măsurilor de protecție a datelor procesate, inclusiv criptarea pachetelor și autentificarea surselor de trafic.
    - Implementarea unui sistem de filtrare a pachetelor care să blocheze sau să semnalizeze traficul considerat periculos.

Sarcina 2: Identificarea, Modelarea și Validarea Cerințelor Funcționale

În cadrul acestui capitol, ne vom concentra pe identificarea, modelarea și validarea cerințelor funcționale pentru sistemul integrat destinat interceptării și analizei traficului de rețea. Este esențial să definim cerințele funcționale ale sistemului, astfel încât să putem construi o soluție eficientă și să răspundem necesităților utilizatorilor finali. Procesul de identificare și validare a acestor cerințe reprezintă fundamentul pentru realizarea unui sistem robust și fiabil, care să îndeplinească scopul propus.

În această etapă, vom explora și metodele de modelare a cerințelor funcționale, inclusiv identificarea entităților cheie ale sistemului și relațiile dintre acestea. De asemenea, vom crea o diagramă de tip ERD (Entity-Relationship Diagram) care va ilustra structura bazei de date a sistemului, evidențiind interacțiunile dintre entități și procesele care trebuie implementate pentru a susține funcționalitățile dorite.

2.1. Definiția cerințelor funcționale

Cerințele funcționale reprezintă descrierea detaliată a comportamentului unui sistem din perspectiva utilizatorilor săi și a proceselor interne. Aceste cerințe sunt esențiale pentru dezvoltarea unui sistem, deoarece ele stabilesc ce trebuie să facă aplicația, cum trebuie să răspundă la acțiunile utilizatorilor și cum să gestioneze datele și fluxurile de informații. În contextul unui sistem integrat pentru interceptarea și analiza traficului de rețea, cerințele funcționale trebuie să fie definite în mod clar pentru a asigura monitorizarea eficientă a traficului, detecția și analiza pachetelor de date, precum și generarea de alerte și rapoarte pentru utilizatori.

Cerințele funcționale ale sistemului ar putea include:

1. **Interceptarea și analiza traficului de rețea**: Sistemul trebuie să poată intercepta pachetele de date care circulă în rețea și să le analizeze în timp real pentru a identifica tipuri de trafic sau anomalii.
2. **Generarea de alerte de securitate**: Dacă un pachet de date reprezintă o amenințare (de exemplu, un atac informatic), sistemul trebuie să poată genera alerte de securitate pentru a semnala utilizatorilor aceste evenimente.
3. **Gestionarea utilizatorilor**: Sistemul trebuie să permită administrarea utilizatorilor, inclusiv atribuirea de roluri (administrator, analist) și permisiuni specifice pentru fiecare utilizator, în funcție de nivelul de acces necesar.
4. **Generarea de rapoarte**: Utilizatorii trebuie să poată crea rapoarte bazate pe analiza evenimentelor și alertelor de securitate, pentru a documenta acțiunile și concluziile derivă din analiza traficului.
5. **Vizualizarea și filtrarea datelor**: Utilizatorii trebuie să poată vizualiza traficul interceptat, alertele generate și rapoartele într-o interfață grafică intuitivă, cu opțiuni de filtrare a datelor pe baza diferitelor criterii (ex. IP sursă, tip de alertă, dată etc.).

2.2. Metodologia de modelare a cerințelor funcționale

Pentru proiectul sistemului integrat destinat interceptării și analizei traficului de rețea, metodologia de modelare a cerințelor funcționale se va axa pe transformarea cerințelor utilizatorilor într-o soluție tehnică coerentă și eficientă. Procesul va include mai multe etape, începând cu colectarea informațiilor de la părțile interesate, pentru a înțelege așteptările utilizatorilor finali și pentru a defini nevoile sistemului. În această fază, vor fi realizate interviuri cu administratorii de rețea, experții în securitate și utilizatorii sistemului, pentru a detalia funcționalitățile dorite și scenariile de utilizare specifice. Este esențial ca toate cerințele să fie clar înțelese pentru a construi o soluție care să răspundă nevoilor utilizatorilor.

Ulterior, va fi realizată identificarea entităților majore din cadrul sistemului, cum ar fi „Sursa de trafic", „Pachet interceptat", „Alertă de securitate" și „Utilizator", dar și stabilirea relațiilor dintre aceste entități, cum ar fi „un utilizator poate genera mai multe rapoarte" sau „un pachet poate declanșa mai multe evenimente". Aceste relații vor fi detaliate în diagrame, inclusiv stabilirea cardinalității fiecărei relații, pentru a reflecta corect interacțiunile dintre elementele sistemului.

În paralel cu identificarea entităților și relațiilor, se vor defini fluxurile de date care vor circula prin sistem, precum și procesele care se vor desfășura pentru a gestiona aceste date. De exemplu, fluxul de la interceptarea unui pachet la generarea unui raport de alertă va fi detaliat pentru a înțelege pașii esențiali ai procesului. Procesele critice, cum ar fi analiza pachetelor și detecția amenințărilor, vor fi descrise pentru a evidenția modul în care fiecare componentă a sistemului va contribui la îndeplinirea cerințelor funcționale.

Pentru a ilustra aceste relații și procese, vor fi realizate diagrame specifice. Diagramele de cazuri de utilizare (Use Case Diagrams) vor descrie interacțiunile dintre utilizatori (de exemplu, administratori, analiști) și sistem. De asemenea, diagrama de flux de date (DFD) va arăta cum datele circulă între entitățile sistemului și cum sunt procesate. Diagrama de entitate-relație (ERD) va detalia structura bazei de date, iar diagramele de stare vor descrie evoluția stărilor entităților sau proceselor, cum ar fi un pachet care poate trece prin stările „interceptat", „analizat" și „raportat".

În final, toate cerințele și modelele create vor fi verificate și validate. Acest proces va implica o revizuire a cerințelor cu părțile interesate pentru a confirma că acestea reflectă nevoile reale ale utilizatorilor și sunt implementabile tehnic. Validarea va include teste pe scenarii reale și prototipurile interfeței pentru a se asigura că soluția propusă este fezabilă și funcționează așa cum a fost planificată.

2.3. Exemple de cerințe funcționale

În cadrul proiectului, cerințele funcționale definesc comportamentul sistemului integrat pentru interceptarea și analiza traficului de rețea. Aceste cerințe sunt fundamentale pentru realizarea unui sistem eficient care să îndeplinească nevoile utilizatorilor și să asigure performanță și securitate.

Unul dintre aspectele esențiale ale sistemului este capacitatea de a intercepta pachetele de rețea în timp real. Aceasta presupune capturarea tuturor pachetelor care circulă pe rețea, indiferent de sursa lor, și stocarea acestora pentru analiză ulterioară. În plus, sistemul trebuie să fie capabil să identifice și să înregistreze sursele de trafic, asociind fiecărui pachet interceptat informațiile relevante, cum ar fi adresa IP a sursei, hostname-ul și locația geografică a acesteia.

După interceptarea pachetelor, sistemul va trebui să efectueze o analiză detaliată a fiecărui pachet, verificând protocolul utilizat, dimensiunea acestuia și semnalele de comportamente anormale, care ar putea indica un atac cibernetic. Dacă se identifică astfel de comportamente, sistemul va genera automat alerte de securitate, care vor include tipul amenințării, severitatea acesteia și timpul în care a fost detectată. Aceste alerte vor fi cruciale pentru administratori și analiști, pentru a interveni rapid în caz de atacuri sau breșe de securitate.

Un alt aspect important al sistemului este generarea și gestionarea rapoartelor de securitate. Utilizatorii autorizați vor putea crea rapoarte detaliate care vor include informații despre pachetele interceptate, alertele de securitate generate și măsurile luate în urma analizei. Aceste rapoarte vor fi utile atât pentru monitorizarea activității de securitate, cât și pentru raportarea incidentelor.
Sistemul trebuie să permită și gestionarea utilizatorilor, prin crearea unor conturi cu diferite nivele de acces. De exemplu, un administrator va avea drepturi complete de configurare a sistemului și vizualizare a tuturor datelor, în timp ce un analist va avea acces doar la vizualizarea alertelor și generarea rapoartelor. În acest sens, trebuie implementată o soluție eficientă pentru gestionarea permisiunilor și monitorizarea activităților utilizatorilor.

În plus, sistemul va include o interfață grafică intuitivă, care va permite utilizatorilor să vizualizeze traficul de rețea interceptat, să analizeze alertele de securitate și să genereze rapoarte, totul într-un mod accesibil și ușor de utilizat. Aceasta va asigura că utilizatorii, indiferent de nivelul lor de experiență, vor putea să interacționeze eficient cu sistemul.

De asemenea, sistemul va trebui să poată trimite alerte și notificări în timp real, prin diverse canale, cum ar fi email-uri sau mesaje interne ale aplicației, pentru a informa utilizatorii despre orice eveniment semnificativ care necesită intervenție.

2.4. Identificarea entităților și a relațiilor

În cadrul sistemului integrat pentru interceptarea și analiza traficului de rețea, identificarea entităților cheie și a relațiilor dintre acestea reprezintă un pas esențial în modelarea structurii bazei de date și a funcționalităților sistemului. Entitățile sunt elementele fundamentale care vor fi gestionate și stocate de sistem, iar relațiile dintre ele reflectă modul în care acestea interacționează și sunt interdependente.

Printre principalele entități ale sistemului se numără:

- **Sursa de trafic**: Reprezintă un nod sau un dispozitiv care generează trafic pe rețea. Această entitate este crucială pentru identificarea originilor pachetelor interceptate. Fiecare sursă de trafic are atribute precum ID-ul sursei (cheia primară), adresa IP, hostname-ul și locația geografică.
- **Pachet interceptat**: Aceasta entitate se referă la pachetele de date care sunt capturate în timpul monitorizării traficului de rețea. Atributele relevante ale acestei entități includ ID-ul pachetului (cheia primară), timpul interceptării, protocolul utilizat, dimensiunea pachetului, sursa IP și destinația IP.
- **Alertă de securitate**: Această entitate este generată atunci când sistemul detectează comportamente anormale sau potențiale atacuri cibernetice în pachetele de rețea. Atributele sale includ ID-ul alertei (cheia primară), tipul alertei, timpul declanșării, descrierea problemei identificate și severitatea alertei.
- **Eveniment analizat**: Reprezintă un eveniment în care sunt analizate pachetele de rețea pentru a detecta probleme de securitate. Evenimentele sunt asociate cu pachetele interceptate și pot fi legate de alerte de securitate. Atributele includ ID-ul evenimentului (cheia primară), ID-ul pachetului (cheia străină), timpul analizei și rezultatul acestei analize.
- **Utilizator**: Entitatea care se referă la persoanele care interacționează cu sistemul. Aceasta poate fi un administrator sau un analist. Atributele includ ID-ul utilizatorului (cheia primară), numele, prenumele, email-ul și rolul utilizatorului (administrator/analist).
- **Raport**: Reprezintă un document generat de utilizatori pentru a documenta activitățile de securitate și evenimentele din sistem. Fiecare raport este asociat cu un utilizator creator și include informații despre pachetele interceptate, alertele de securitate și măsurile luate. Atributele includ ID-ul raportului (cheia primară), data generării, utilizatorul creator (cheia străină), și descrierea raportului.

Relațiile dintre aceste entități sunt esențiale pentru modelarea fluxurilor de date și pentru implementarea corectă a funcționalităților sistemului:

- **Sursa de trafic → Pachet interceptat**: O sursă de trafic poate genera mai multe pachete interceptate. Această relație este de tipul 1:N, unde o sursă poate fi asociată cu multiple pachete interceptate.
- **Pachet interceptat → Eveniment analizat**: Un pachet interceptat poate fi analizat în mai multe evenimente, fiecare eveniment având un rezultat diferit în urma analizei. Aceasta este o relație de tipul 1:N, unde un pachet poate fi legat de mai multe evenimente de analiză.
- **Eveniment analizat → Alertă de securitate**: Un eveniment analizat poate declanșa una sau mai multe alerte de securitate, în funcție de problemele identificate în timpul analizei. Relația este de tipul 1:N, un eveniment putând să genereze mai multe alerte de securitate.

- **Utilizator → Raport**: Un utilizator poate crea mai multe rapoarte în cadrul sistemului, fiecare raport fiind asociat cu un utilizator creator. Aceasta este o relație de tipul 1:N, un utilizator având posibilitatea de a crea mai multe rapoarte.

2.5. Diagrama Relație - Entitate (ERD)

În cadrul procesului de modelare a cerințelor funcționale pentru sistemul de interceptare și analiză a traficului de rețea, este esențial să definim corect structura bazei de date. Diagrama de tip ERD (Entity-Relationship Diagram) oferă o reprezentare vizuală a entităților implicate și a relațiilor dintre acestea, esențială pentru înțelegerea modului în care datele vor fi organizate și interconectate. În această diagramă, am identificat entitățile cheie ale sistemului, cum ar fi sursa de trafic, pachetul interceptat, alerta de securitate, evenimentul analizat, utilizatorul și raportul generat, și am stabilit relațiile dintre ele.

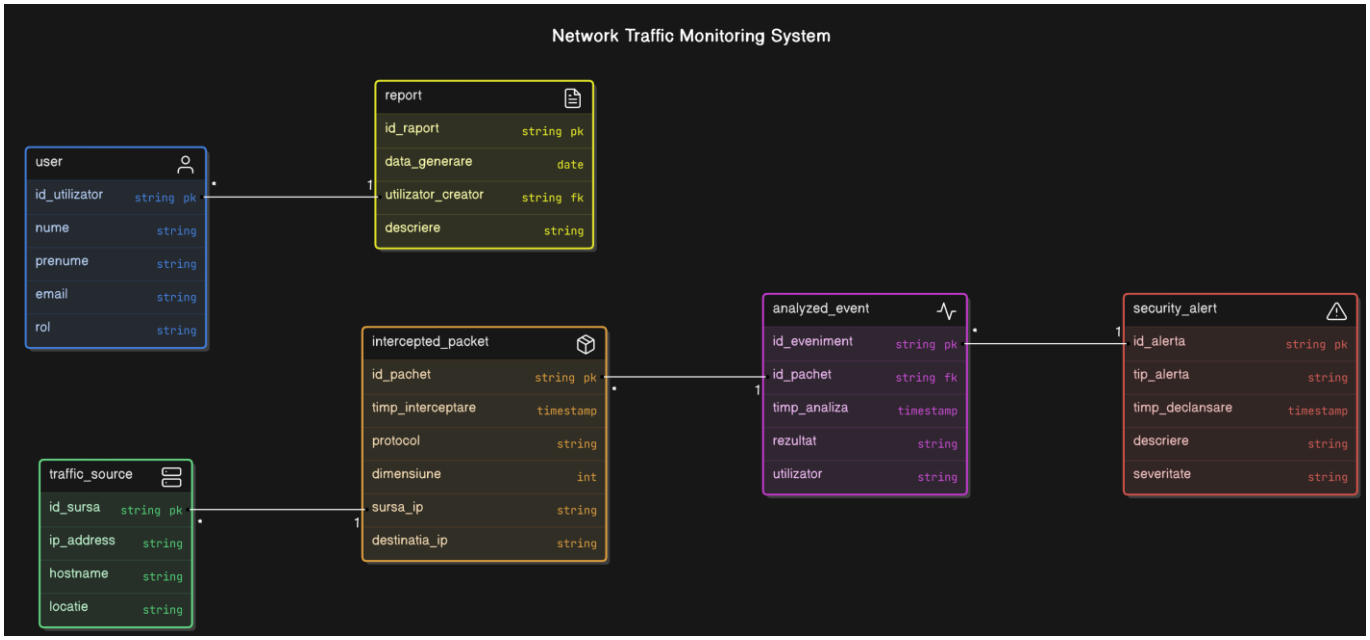

Figura 1 – Diagrama ERD

Prin această abordare, am putut clarifica modul în care informațiile sunt stocate, gestionate și interacționează în cadrul sistemului, având astfel o bază solidă pentru implementarea funcționalităților de securitate și monitorizare a traficului de rețea. Relațiile dintre entități sunt definite prin cardinalități clare, care reflectă natura interacțiunilor dintre acestea și contribuie la crearea unui sistem robust și scalabil.

Sarcina 3: Identificarea și Modelarea Fluxului de Date

Fluxul de date reprezintă modul în care informațiile circulă între diferitele componente ale sistemului, precum sursele de trafic, pachetele interceptate, evenimentele analizate și rapoartele generate. O înțelegere clară a fluxului de date este esențială pentru a garanta eficiența și securitatea sistemului, precum și pentru a asigura integrarea corectă între toate procesele implicate.

În această etapă, vom analiza pașii prin care datele sunt colectate, procesate și stocate, și vom crea o diagramă de flux de date (DFD). Această diagramă va ilustra procesele și entitățile care sunt implicate în fiecare etapă a fluxului, oferind o reprezentare vizuală a interacțiunilor și a transferului de informații în

cadrul sistemului. Identificarea clară a fluxului de date va contribui la o mai bună optimizare a resurselor și va facilita implementarea unor mecanisme eficiente de monitorizare și securitate.

3.1. Definiția fluxului de date

Fluxul de date reprezintă modul în care informațiile sunt transferate între diferitele componente ale unui sistem. În contextul sistemului integrat pentru interceptarea și analiza traficului de rețea, fluxul de date descrie procesul prin care datele sunt colectate, procesate, stocate și transmise între entitățile implicate, cum ar fi sursele de trafic, pachetele interceptate, evenimentele analizate, alertele de securitate și rapoartele generate.

Definirea fluxului de date implică stabilirea unui traseu clar prin care informațiile circulă, identificând sursele, destinațiile, procesele de transformare ale datelor și modalitățile prin care acestea sunt stocate sau distribuite. De asemenea, fluxul de date include și detalierea interacțiunilor dintre diferitele entități, asigurându-se astfel că toate procesele funcționează într-un mod coerent și eficient.

O diagramă de flux de date (DFD) este utilizată pentru a reprezenta vizual aceste interacțiuni, facilitând înțelegerea și analiza proceselor interne ale sistemului. Într-o astfel de diagramă, fiecare proces este ilustrat printr-un cerc sau dreptunghi, iar fluxul de date este reprezentat prin săgeți care conectează procesele și entitățile. Astfel, fluxul de date ajută la clarificarea modului în care informațiile sunt gestionate și transferate în cadrul sistemului.

3.2. Metodologia de modelare a fluxului de date

Metodologia de modelare a fluxului de date implică definirea și reprezentarea vizuală a modului în care informațiile circulă prin sistem, identificând procesele cheie, sursele de date, destinațiile și relațiile dintre acestea. În cadrul acestui sistem integrat pentru interceptarea și analiza traficului de rețea, modelarea fluxului de date presupune următorii pași esențiali:

1. **Identificarea proceselor cheie** – Primul pas în modelarea fluxului de date este identificarea proceselor fundamentale care se vor desfășura în sistem. Acestea pot include operații precum interceptarea pachetelor de trafic, analiza evenimentelor, generarea alertelor de securitate sau crearea rapoartelor.
2. **Determinarea entităților externe și a fluxului de date** – După definirea proceselor, următorul pas este identificarea entităților externe, cum ar fi sursele de trafic, utilizatorii sau sistemele externe cu care sistemul va interacționa. Este esențial să fie înțelese datele care sunt trimise sau primite de aceste entități și fluxul de informație pe care îl generează.
3. **Crearea diagramei de flux de date (DFD)** – Utilizând simboluri standardizate, cum ar fi cercuri pentru procese, săgeți pentru fluxuri de date și dreptunghiuri pentru entități externe, se creează o diagramă care ilustrează modul în care datele circulă între procesele interne ale sistemului și

entitățile externe. Aceasta ajută la vizualizarea clară a interacțiunilor dintre diversele componente și la înțelegerea logicii operaționale a sistemului.

4. **Definirea nivelurilor de detaliu** – Modelarea fluxului de date se face pe mai multe niveluri, începând cu o diagramă generală, care oferă o vedere de ansamblu a sistemului, și continuând cu diagrame mai detaliate, care descriu procesele interne și fluxurile de date mai specifice. Aceste niveluri permit o înțelegere graduală și mai profundă a arhitecturii sistemului.
5. **Validarea fluxurilor de date** – După realizarea diagramei de flux de date, este important să se valideze corectitudinea acesteia. Acest pas presupune verificarea coerenței între fluxurile de date și cerințele funcționale ale sistemului, precum și asigurarea faptului că toate procesele și interacțiunile sunt clar definite și că nu există informații pierdute sau inconsecvente.

3.3. Exemple de fluxuri de date

Exemplele de fluxuri de date pot include următoarele scenarii:

1. **Interceptarea pachetelor de trafic**

   Fluxul de date începe cu interceptarea pachetelor de trafic de la sursa de rețea. Aceste pachete sunt trimise către sistem pentru a fi procesate. În acest flux, pachetele conțin informații esențiale, cum ar fi adresele IP sursă și destinație, protocoalele folosite și dimensiunile acestora. Fluxul de date este generat de către sursa de trafic și se direcționează către procesul de interceptare al pachetelor.

   - **Entitate externă**: Sursa de trafic
   - **Proces intern**: Interceptarea pachetului
   - **Flux de date**: Pachet de trafic (datele pachetului, cum ar fi IP sursă, IP destinație, protocol, dimensiune)
   - **Destinație**: Pachet interceptat
2. **Analiza pachetelor de trafic**

   După interceptarea pachetelor, procesul de analiză va prelua aceste date și va verifica dacă există anomalii, vulnerabilități sau evenimente relevante, cum ar fi posibile atacuri de tip DoS sau acces neautorizat. Datele despre pachetul interceptat sunt procesate și analizate, iar rezultatul analizei poate include detectarea unor posibile alerte de securitate.

   - **Entitate externă**: Pachet interceptat
   - **Proces intern**: Analiza pachetului de trafic
   - **Flux de date**: Detalii despre pachetul de trafic (timestamp, protocol, sursa/destinația IP, etc.)
   - **Destinație**: Eveniment analizat (informațiile despre analiza pachetului)
3. **Generarea alertelor de securitate**

   Dacă analiza pachetului de trafic identifică un eveniment anormal sau un risc de securitate,

sistemul va genera o alertă. Fluxul de date în acest caz include detalii despre tipul alertelor, severitatea acestora și alte informații relevante, care sunt trimise către utilizatorul care gestionează sistemul (admin sau analist).

- **Entitate externă**: Eveniment analizat
- **Proces intern**: Generarea alertelor de securitate
- **Flux de date**: Tipul alertelor, severitatea, descrierea, timpul declanșării
- **Destinație**: Alertă de securitate

4. **Crearea rapoartelor de analiză**

   După analiza completă a traficului și generarea alertelor, un utilizator (de tip admin sau analist) poate crea un raport care sumarizează activitatea sistemului și eventualele vulnerabilități descoperite. Fluxul de date include detalii despre evenimentele analizate, alertele generate și orice alte informații relevante care sunt incluse în raport.

   - **Entitate externă**: Eveniment analizat, Alertă de securitate
   - **Proces intern**: Generarea raportului de analiză
   - **Flux de date**: Detalii despre evenimentele analizate, alerte de securitate, informații relevante pentru raport
   - **Destinație**: Raport generat (care va fi transmis utilizatorului sau stocat)

3.4. Diagrama Fluxului de Date (DFD)

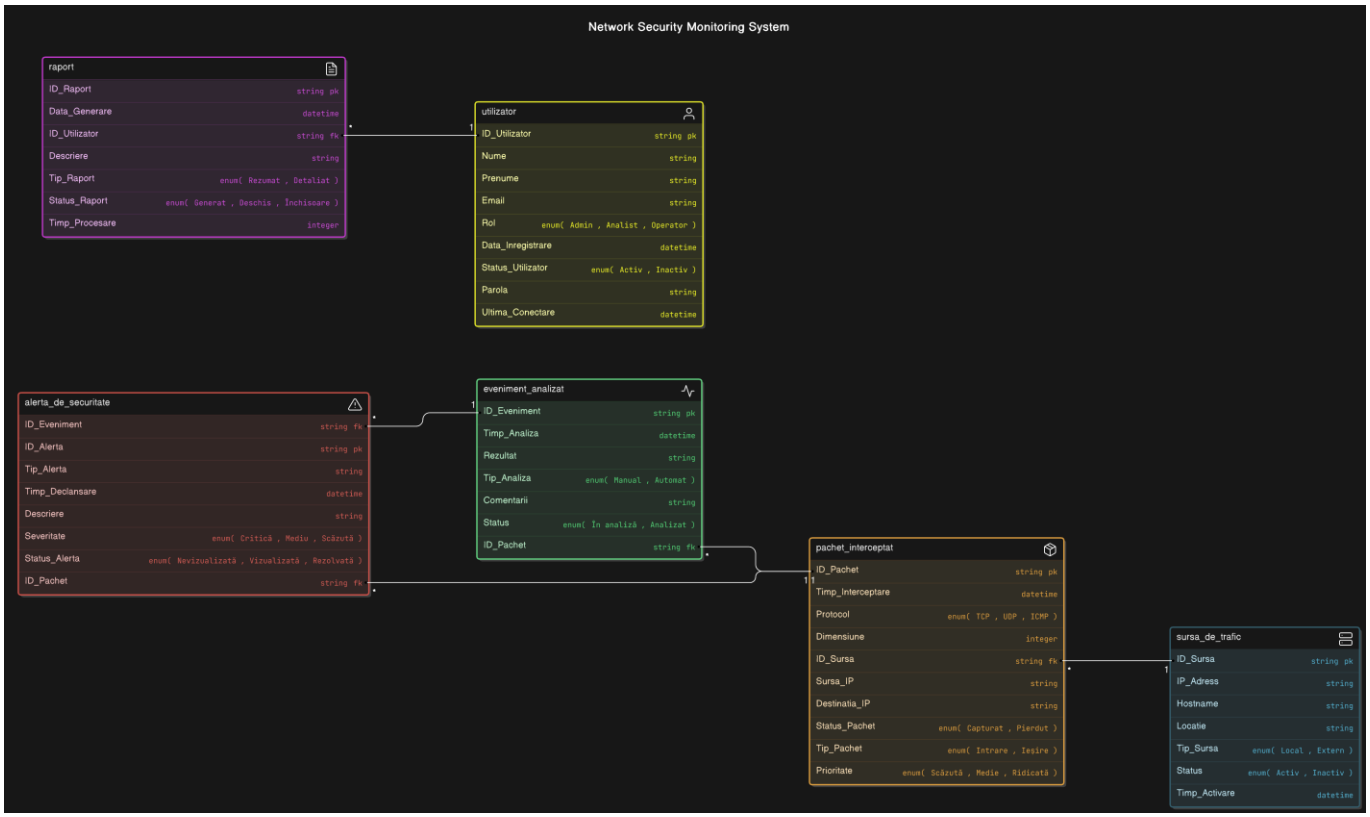

Figura 2 – Diagrama DFD

Acest tip de diagramă de flux de date (DFD) ajută la înțelegerea procesului global și a interacțiunilor dintre diferitele entități și procese ale sistemului. În contextul proiectului tău, sistemul de interceptare și analiză a traficului de rețea funcționează în următorul mod:

1. **Sursa de trafic** furnizează pachetele de date care sunt interceptate de sistem. Această sursă poate fi orice dispozitiv din rețea care trimite pachete de date, cum ar fi servere, clienți sau echipamente de rețea.
2. Pachetele interceptate sunt apoi trimise către procesul de **Analiza pachetelor**, unde sunt examinate pentru a identifica potențiale amenințări, anomalii sau comportamente suspecte. Dacă analiza detectează o problemă de securitate, sistemul generează o **alertă de securitate**.
3. **Alertele de securitate** sunt trimise către utilizatorii administrației sistemului (admin sau analist), care vor analiza situația și pot decide asupra măsurilor corective.
4. În funcție de analiza efectuată, utilizatorii pot crea un **raport** care să rezume activitatea și concluziile procesului de monitorizare. Raportul conține detalii precum descrierea alertelor, acțiunile întreprinse și rezultatele analizei.
5. **Stocarea rapoartelor** într-o bază de date permite ca acestea să fie accesibile ulterior pentru audituri, analize ulterioare sau verificări istorice. Aceste rapoarte sunt utile pentru analiza tendințelor de securitate și pentru îmbunătățirea continuă a sistemului de monitorizare.

Prin utilizarea unui DFD, devine mai ușor să se vizualizeze cum datele sunt procesate și transferate în cadrul sistemului, ceea ce face posibilă o mai bună înțelegere a fluxului de informații și a interacțiunilor între componente. Acest tip de diagramă este util și în identificarea eventualelor puncte de îmbunătățire în proces, precum și în asigurarea unei arhitecturi clare și eficiente a sistemului.

Sarcina 4: Identificarea Cerințelor Nonfuncționale

Cerințele nonfuncționale sunt caracteristici ale sistemului care nu descriu comportamentele funcționale directe, dar sunt esențiale pentru performanța și fiabilitatea acestuia. Aceste cerințe pot include performanța, scalabilitatea, securitatea, interoperabilitatea, disponibilitatea și ușurința de utilizare. În cadrul acestui capitol, vom explora modul în care cerințele nonfuncționale influențează designul și implementarea sistemului, punând accent pe cerințele specifice de performanță și securitate pentru un sistem de interceptare și analiză a traficului de rețea.

Pentru a ilustra modul în care aceste cerințe sunt implementate în sistem, vom utiliza diagrame de **Use Case**. Acestea vor ajuta la vizualizarea interacțiunilor utilizatorului cu sistemul și a scenariilor de utilizare care pot afecta performanța și securitatea. De exemplu, un **Use Case** poate descrie un scenariu în care un utilizator administrează pachetele de trafic în timp real, iar cerințele de performanță vor impune ca acest proces să fie realizat într-un interval de timp specificat.

4.1. Definiția cerințelor nonfuncționale

Cerințele nonfuncționale se referă la specificațiile care definesc modul în care un sistem trebuie să îndeplinească sarcinile sale funcționale. Aceste cerințe sunt legate de parametri precum timpul de răspuns, utilizarea resurselor, scalabilitatea și securitatea, și sunt adesea măsurabile prin teste de performanță. Spre deosebire de cerințele funcționale care descriu *ce* face sistemul, cerințele nonfuncționale se concentrează pe *cum* trebuie să îndeplinească aceste funcționalități.

În contextul proiectului nostru, cerințele nonfuncționale pot include, de exemplu, timpul maxim în care trebuie să fie procesat un pachet de trafic pentru a declanșa o alertă de securitate, sau capacitatea sistemului de a manipula un număr mare de pachete simultan, fără a afecta performanța.

Un exemplu relevant pentru cerințele nonfuncționale ar fi un **Use Case** care descrie procesul de monitorizare a traficului într-un sistem cu un număr mare de utilizatori, în care cerințele de performanță se referă la capabilitatea sistemului de a răspunde rapid și eficient, chiar și în condiții de trafic intens.

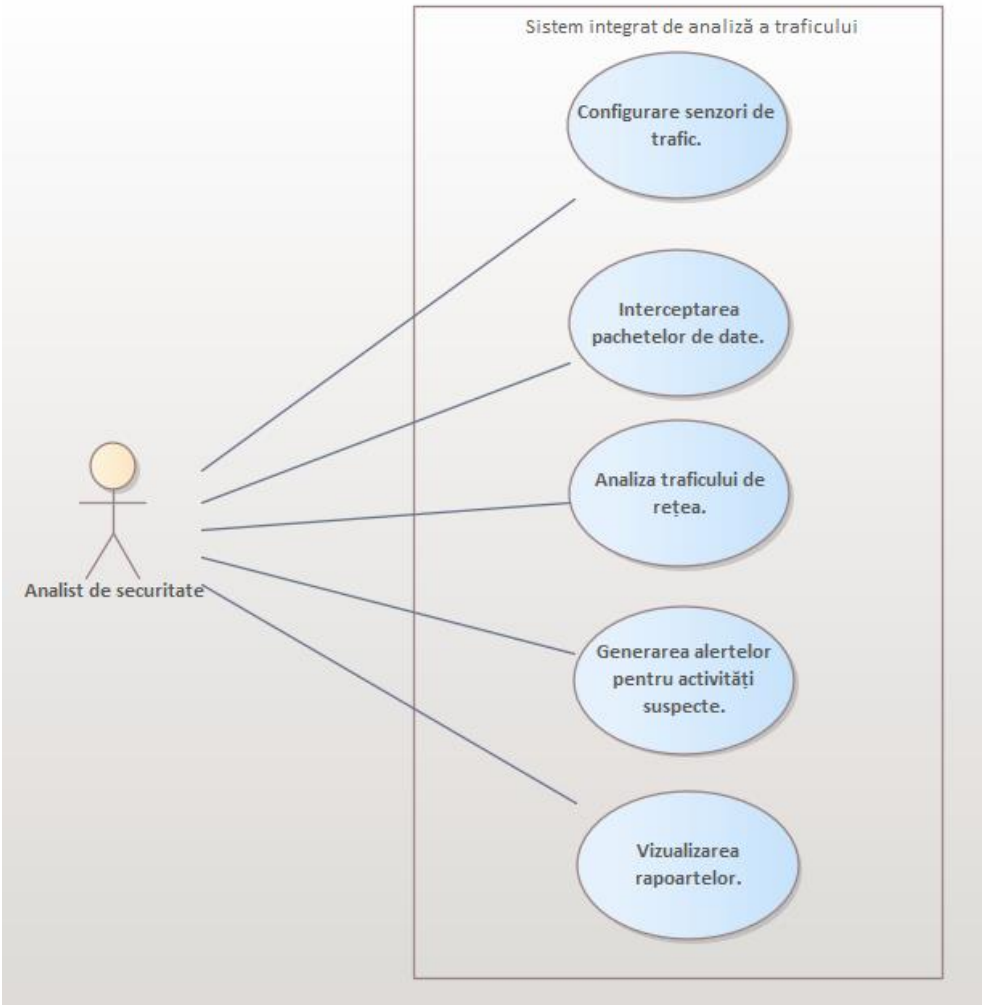

Figura 3 – Use Case diagrama

Diagrama **Use Case** poate ilustra interacțiunile dintre utilizatori (de exemplu, administratori sau analisti de securitate) și subsistemele care procesează traficul, în timp ce cerințele de performanță, scalabilitate și securitate vor fi derivate din aceleași interacțiuni.

4.2. Exemple de cerințe nonfuncționale

Cerințele nonfuncționale sunt acele caracteristici ale sistemului care nu se referă direct la funcțiile sau comportamentele acestuia, ci la modul în care sistemul trebuie să se comporte în diferite condiții. Aceste cerințe sunt esențiale pentru a asigura performanța, fiabilitatea, scalabilitatea și utilizabilitatea sistemului pe termen lung.

Pentru proiectul de interceptare și analiză a traficului de rețea, câteva exemple de cerințe nonfuncționale ar putea include:

1. **Performanță:** Sistemul trebuie să poată analiza și procesa cel puțin 100 de pachete de rețea pe secundă, fără a depăși un timp de răspuns de 3 secunde pentru fiecare pachet interceptat. Aceasta asigură că aplicația este suficient de rapidă pentru a face față traficului de rețea intens.
2. **Scalabilitate:** Sistemul trebuie să poată fi extins pentru a analiza pachete din mai multe surse de trafic simultan, fără a afecta performanța generală. Acesta ar trebui să fie capabil să gestioneze o creștere de 50% a volumului de trafic pe an, fără necesitatea unor modificări majore ale arhitecturii.
3. **Fiabilitate:** Sistemul trebuie să fie disponibil 99,9% din timp, cu un timp maxim de nefuncționare (downtime) de 8 ore pe an. În plus, toate datele procesate de sistem trebuie să fie stocate în siguranță, fără pierderi sau coruperea acestora.
4. **Securitate:** Datele procesate de sistem trebuie să fie criptate atât în tranzit, cât și la stocare. Accesul la sistem trebuie să fie restricționat doar utilizatorilor autorizați, iar orice încercare de acces neautorizat trebuie să fie monitorizată și înregistrată într-un jurnal de securitate.
5. **Usabilitate:** Interfața grafică a utilizatorului (GUI) trebuie să fie intuitivă și ușor de folosit, astfel încât chiar și utilizatorii cu experiență limitată să poată naviga cu ușurință în sistem. Acesta ar trebui să includă un ghid de utilizare și să permită accesul rapid la informațiile critice.
6. **Compatibilitate:** Sistemul trebuie să fie compatibil cu majoritatea browserelor web moderne (Google Chrome, Mozilla Firefox, Safari, Edge) și să funcționeze pe cele mai utilizate platforme de operare (Windows, Linux, macOS).
7. **Mentenabilitate:** Sistemul trebuie să fie ușor de întreținut și actualizat. Codul sursă trebuie să fie bine documentat, iar procesul de implementare a noilor funcționalități și corectarea erorilor să poată fi realizat fără întreruperea serviciilor existente.

Sarcina 5: Decompoziția Lucrărilor la Realizarea Cerințelor

5.1. Definiția decompoziției lucrărilor

Decompoziția lucrărilor se referă la procesul prin care un proiect complex este împărțit în sarcini și sub-sarcini mai mici și mai gestionabile. Scopul acestui proces este de a facilita planificarea și implementarea etapelor într-o manieră organizată, eficientă și ușor de urmărit. În contextul proiectului nostru de interceptare și analiză a traficului de rețea, decompoziția lucrărilor presupune separarea

cerințelor și a activităților în funcții clare, care pot fi atribuite diferitelor echipe sau persoane. Această metodă contribuie la reducerea riscurilor, asigură o mai bună urmărire a progresului și permite identificarea timpurie a problemelor.

5.2. Metodologia de decompoziție

Metodologia de decompoziție a lucrărilor pentru realizarea cerințelor se bazează pe utilizarea unor tehnici de **descompunere ierarhică**. Aceste tehnici permit împărțirea unui obiectiv mare în părți mai mici, fiecare având scopuri specifice și rezultate clare.

Procesul de decompoziție poate fi împărțit în mai multe etape:

1. **Definirea obiectivelor majore:** În această etapă, se identifică activitățile principale ale proiectului. De exemplu, analiza cerințelor de securitate a rețelei, dezvoltarea infrastructurii de capturare a datelor, crearea interfeței utilizatorului etc.
2. **Împărțirea obiectivelor majore în sarcini specifice:** Fiecare obiectiv major este împărțit în sarcini și sub-sarcini care pot fi realizate de echipele sau persoanele respective. De exemplu, pentru "analiza cerințelor de securitate", sarcinile ar putea include analiza vulnerabilităților, analiza tipurilor de atacuri, crearea unui plan de protecție etc.
3. **Detalierea sarcinilor:** Fiecare sarcină este detaliată în pași mici și gestionabili, cu termene, resurse necesare și responsabilități clar definite. De exemplu, în cazul "capturării datelor", pașii pot include selecția instrumentelor de capturare, configurarea filtrelor de trafic, analiza protocoalelor și implementarea procesului de colectare a datelor.

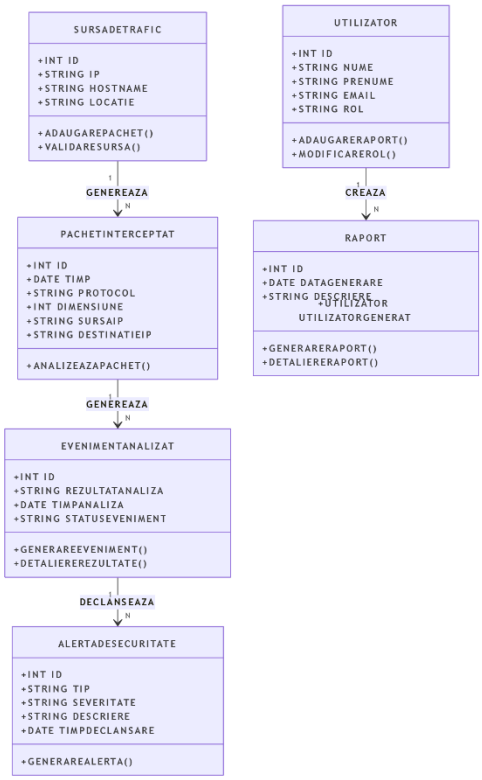

Figura 4 – Diagrama de clasa

În cadrul acestui subcapitol, diagramă de tip **Clasa** prezentată anterior ilustrează structura internă a sistemului de interceptare și analiză a traficului de rețea, detaliind principalele clase, atributele acestora și relațiile dintre ele.

Această diagramă ajută la vizualizarea modului în care diferitele componente ale sistemului sunt organizate și interacționează între ele. Fiecare clasă reprezintă o entitate cheie a sistemului, iar relațiile dintre clase reflectă fluxurile de date și dependențele dintre acestea.

De exemplu, clasa **PachetInterceptat** poate conține atribute precum **Timp_Interceptare** și **Protocol**, iar relațiile cu alte clase, precum **SursaDeTrafic** și **EvenimentAnalizat**, sunt esențiale pentru a înțelege cum sunt gestionate datele de trafic și cum sunt legate între ele diferitele operații ale sistemului.

Acest tip de diagramă este fundamental pentru a oferi o viziune clară asupra structurii obiectelor și a comportamentului acestora, sprijinind procesul de implementare și testare al sistemului.

5.3. Exemple de decompoziție a sarcinilor

Un exemplu de decompoziție a sarcinilor poate include procesul de interceptare a pachetelor de date. Această sarcină principală poate fi împărțită în subactivitățile de configurare a monitorizării rețelei, capturarea pachetelor, stocarea acestora într-o bază de date și analiza ulterioară a datelor capturate. Fiecare dintre aceste subactivități poate fi tratată ca o unitate separată de muncă, cu obiective și livrabile specifice.

De asemenea, analiza alertelor de securitate poate fi decompozită în mai multe etape, precum detectarea evenimentelor de securitate, generarea alertelor corespunzătoare și evaluarea impactului acestora asupra sistemului. Fiecare etapă implică diferite clase și funcționalități, iar procesul de decompoziție ajută la identificarea clară a responsabilităților și a fluxurilor de lucru.
În mod similar, gestionarea utilizatorilor și generarea rapoartelor de securitate pot fi de asemenea descompuse în activități precum autentificarea utilizatorilor, alocarea permisiunilor, generarea și personalizarea rapoartelor. Fiecare dintre aceste activități poate fi tratată separat, dar interacțiunile între ele trebuie modelate și înțelese pentru a asigura o implementare coerentă a sistemului.

Sarcina 7: Concluzii

În concluzie, laboratorul privind realizarea unui sistem integrat pentru interceptarea și analiza traficului de rețea a fost o oportunitate esențială pentru a aplica concepte fundamentale de modelare a cerințelor funcționale și nonfuncționale, precum și pentru a înțelege importanța decompoziției sarcinilor și fluxurilor de date într-un sistem complex. Prin realizarea diagramelor ERD, flux de date, și UML, am dobândit o viziune detaliată asupra arhitecturii și funcționalității sistemului propus.

Identificarea și modelarea cerințelor funcționale și nonfuncționale ne-au permis să definim cu exactitate cerințele sistemului, asigurându-ne că soluția va răspunde nevoilor utilizatorilor finali și va respecta standardele de performanță, securitate și scalabilitate.

De asemenea, prin decompoziția lucrărilor și stabilirea unui flux clar al sarcinilor, am asigurat o abordare structurată și eficientă a procesului de implementare. Diagramele UML și exemplele de fluxuri de date oferă o bază solidă pentru dezvoltarea continuă a sistemului și permit o înțelegere clară a interacțiunilor dintre componentele acestuia.