



MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

Programul de studii: Securitatea informațională

Sistem integrat pentru interceptarea și analiza traficului de rețea

Practica de licență

Student(ă):	_____	Chirița Stanislav, SI-211
Coordonator întreprindere:	_____	Jovmir Cristina, Head of CERT Gov
Coordonator de licență:	_____	Masiutin Maxim, asist.univ.
Coordonator universitate:	_____	Bulai Rodica, lector universitar

Chișinău, 2025

DECLARAȚIA DE ORIGINALITATE

Subsemnatul, Chirița Stanislav, declar pe proprie răspundere, că lucrarea de față este rezultatul muncii mele, realizată pe baza propriilor cercetări și pe baza informațiilor obținute din surse care au fost citate și indicate conform normelor etice în note și în bibliografie.

Declar că lucrarea nu a mai fost prezentată sub această formă la nici o instituție de învățământ superior în vederea obținerii titlului de inginer licențiat.

Semnătura autorului _____

CUPRINS

DECLARAȚIA DE ORIGINALITATE	2
ABREVIERI	4
INTRODUCERE.....	5
1 ANALIZA DOMENIULUI DE STUDIU.....	6
1.1 Importanța temei	6
1.2 Sisteme similare cu proiectul realizat	6
1.3 Scopul, obiectivele și cerințele sistemului	9
2 MODELAREA ȘI PROIECTAREA SISTEMUL INFORMATIC	11
2.1 Descrierea comportamentală a sistemului	11
2.1.1 Imaginea generală asupra sistemului	12
2.1.2 Modelarea vizuală a fluxurilor.....	13
2.1.3 Descrierea scenariilor de utilizare a aplicației	15
2.2 Descrierea structurală a sistemului.....	15
2.2.1 Descrierea structurii statice a sistemului	16
2.2.2 Relațiile de dependență între componentele sistemului.....	17
3 REALIZAREA SISTEMULUI.....	19
3.1 Structura sistemului	19
3.2 Integrarea serviciilor externe	20
3.3 Implementarea cerințelor funcționale.....	20
4 DOCUMENTAREA SISTEMULUI	22
4.1 Interfața de autentificare	22
4.3 Interacțiunea cu baza de date.....	23
4.4 Interfața principală a aplicației.....	24
4.5 Capturarea traficului de rețea.....	24
BIBLIOGRAFIE.....	28

ABREVIERI

IPS (Intrusion Prevention System) – Sistem de prevenire a intruziunilor

IDS (Intrusion Detection System) – Sistem de detecție a intruziunilor

SIEM (Security Information and Event Management) – Soluție integrată pentru gestionarea informațiilor și evenimentelor de securitate

GUI (Graphical User Interface) – Interfață grafică pentru utilizator

HTTP (Hypertext Transfer Protocol) – Protocol pentru transferul de date în format hipermedia pe web

DNS (Domain Name System) – Sistem distribuit care traduce numele de domeniu în adrese IP

SSL (Secure Sockets Layer) – Protocol criptografic pentru securizarea comunicațiilor pe internet (în prezent înlocuit în mare parte de TLS – Transport Layer Security)

INTRODUCERE

Practica desfășurată în cadrul **Serviciului Tehnologie Informației și Securitate Cibernetică (STISC)** a avut ca obiectiv principal aprofundarea cunoștințelor teoretice și aplicarea acestora într-un mediu real de lucru, în domeniul securității cibernetice și al analizei traficului de rețea. Activitatea s-a desfășurat pe parcursul perioadei **03.02.2025 – 28.03.2025**, timp în care am avut ocazia să mă familiarizez cu tehnologiile și metodele utilizate pentru interceptarea și analiza traficului de rețea, un domeniu esențial pentru asigurarea securității informaționale la nivel național.

Scopul principal al practicii a fost dezvoltarea unui **sistem integrat pentru interceptarea și analiza traficului de rețea**, utilizând tehnici avansate de captură și interpretare a pachetelor de date. Prin această experiență, am urmărit să înțeleg mai bine structura și vulnerabilitățile rețelelor, metodele de detecție a amenințărilor și utilizarea instrumentelor specifice pentru monitorizarea traficului. Importanța acestui subiect este majoră, având în vedere creșterea constantă a atacurilor cibernetice și necesitatea implementării unor soluții eficiente de protecție în infrastructurile critice.

1 ANALIZA DOMENIULUI DE STUDIU

Analiza traficului de rețea are un rol important în securitatea cibernetică, permițând identificarea atacurilor și optimizarea performanței rețelelor. În cadrul practicii desfășurate la **Serviciul Tehnologie Informației și Securitate Cibernetică (STISC)**, am utilizat instrumente specializate pentru interceptarea și analiza pachetelor de date, având ca scop implementarea unui sistem integrat de monitorizare a traficului.

Am folosit tehnologii precum **Wireshark**, **tcpdump**, **Zeek** și **Suricata**, fiecare având roluri specifice în captarea și interpretarea traficului de rețea. Aceste instrumente au fost utilizate pentru a analiza avantajele și limitările fiecăruia, în scopul dezvoltării unei soluții proprii optimizate. Testele efectuate au permis identificarea punctelor forte și a aspectelor care necesită îmbunătățiri, oferind o bază solidă pentru crearea unui instrument personalizat, capabil să răspundă cerințelor specifice de securitate și monitorizare a rețelei.

Rezultatele au demonstrat eficiența unei soluții automatizate pentru detectarea amenințărilor. Optimizările viitoare pot include integrarea algoritmilor de **machine learning** pentru îmbunătățirea detecției anomaliilor. Această practică a oferit o înțelegere practică asupra monitorizării traficului și a metodelor de protecție a rețelelor informatice.

1.1 Importanța temei

Implementarea unui sistem integrat pentru interceptarea și analiza traficului de rețea contribuie la identificarea anomaliilor și la reacția rapidă împotriva incidentelor de securitate. Prin monitorizarea fluxului de date, se poate determina comportamentul anormal al utilizatorilor sau dispozitivelor dintr-o rețea, prevenind astfel compromiterea infrastructurii IT. De asemenea, analiza traficului permite optimizarea performanței rețelelor și asigurarea respectării politicilor de securitate și a reglementărilor legale.

Prin această lucrare, se urmărește testarea și compararea unor instrumente existente de analiză a traficului, identificarea punctelor lor slabe și dezvoltarea unei soluții îmbunătățite, capabile să ofere o detecție mai rapidă și mai precisă a amenințărilor. Astfel, cercetarea contribuie la îmbunătățirea metodelor de protecție a rețelelor informatice și la dezvoltarea unor tehnologii mai eficiente în domeniul securității cibernetică.

1.2 Sisteme similare cu proiectul realizat

În domeniul securității cibernetică, există numeroase sisteme dezvoltate pentru interceptarea și analiza traficului de rețea. Acestea sunt utilizate atât pentru monitorizare pasivă, cât și pentru detecția activă a amenințărilor. Printre cele mai relevante soluții similare cu proiectul realizat se numără:

Wireshark – Unul dintre cele mai populare instrumente de analiză a traficului de rețea. Permite captarea și inspecția detaliată a pachetelor, fiind utilizat atât pentru depanare, cât și pentru identificarea vulnerabilităților. Totuși, Wireshark necesită o analiză manuală intensă, ceea ce poate fi un dezavantaj

pentru detecția rapidă a atacurilor. Wireshark oferă o flexibilitate remarcabilă prin posibilitatea de a defini filtre personalizate pentru captarea și analizarea pachetelor, facilitând o înțelegere mai detaliată a traficului de rețea. Datorită acestei granularități, un administrator poate identifica exact sursa problemelor de conectivitate sau poate descoperi comportamente suspecte cu precizie ridicată.

Cu toate acestea, utilizarea Wireshark necesită timp și expertiză pentru interpretarea corectă a datelor. În cazul unui volum mare de trafic, o echipă de securitate poate fi copleșită de cantitatea de informații brute, ceea ce face dificilă detectarea rapidă a atacurilor sofisticate. Din acest motiv, Wireshark este adesea folosit în combinație cu alte soluții – de exemplu, sisteme de detectare a intruziunilor (IDS/IPS) care asigură monitorizare în timp real și alerte automate.

Astfel, Wireshark rămâne un instrument esențial pentru investigații post-eveniment și pentru dezvoltarea abilităților de analiză a securității rețelei, dar nu trebuie privit ca un mijloc unic de protecție. Într-un mediu complex, el reprezintă doar o parte dintr-un ecosistem mai larg de soluții de securitate, completat de automatizări, monitorizare continuă și controale adecvate la nivelul infrastructurii.

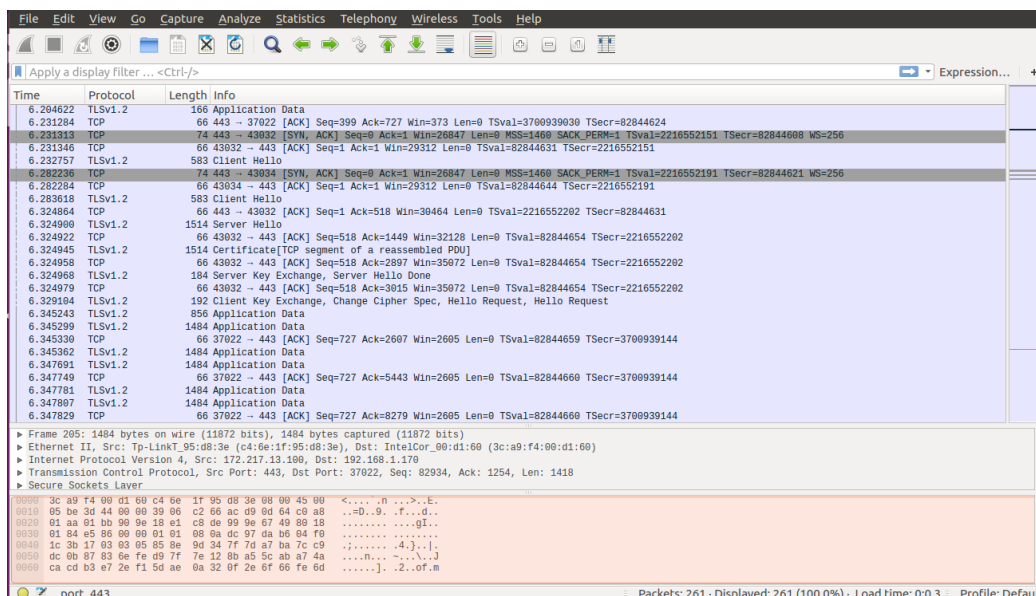


Figura 1.1 – Wireshark

tcpdump – Un instrument în linie de comandă care permite captarea și filtrarea pachetelor de date. Este eficient pentru analiza traficului în timp real, dar are o interfață limitată și necesită cunoștințe avansate pentru interpretarea rezultatelor. **tcpdump** este un instrument pentru specialiștii în securitate și administrare de rețea, însă se folosește exclusiv din linia de comandă. Datorită simplității și a consumului redus de resurse, este deseori preferat în diagnosticarea rapidă a problemelor de rețea și monitorizarea traficului în timp real, mai ales pe sisteme Linux/Unix.

Cu toate acestea, tcpdump oferă doar o interfață textuală, care poate părea intimidantă pentru utilizatorii neexperimentați. Pentru a utiliza eficient filtrele și pentru a interpreta corect datele capturate, este necesar un nivel avansat de cunoștințe de rețea (protocoale, formate de pachete etc.). În plus, pentru

analiza post-captură și interpretarea avansată a vulnerabilităților, se recomandă exportarea fișierelor pcap către soluții cu interfață grafică (de exemplu, Wireshark).

Ca și în cazul oricărui instrument de analiză a traficului, tcpdump nu oferă un mecanism automat de detectare a atacurilor. Este util în identificarea problemelor punctuale și în înțelegerea traficului la nivel de pachete, dar nu poate înlocui soluțiile automatizate de monitorizare și securitate (IDS/IPS, SIEM etc.). Prin urmare, implementarea tcpdump în cadrul unei strategii mai ample de securitate trebuie să fie dublată de alte instrumente și procese de detecție și răspuns la incidente.

```
root@kali:~# tcpdump -i eth0 -s 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
04:57:29.856624 IP 192.168.20.1.40816 > 192.168.20.255.40816: UDP, length 131
04:57:29.857265 IP kali.58048 > _gateway.domain: 22425+ PTR? 255.20.168.192.in-addr.arpa. (45)
04:57:29.858650 IP _gateway.domain > kali.58048: 22425 NXDomain 0/1/0 (80)
04:57:29.859271 IP kali.42264 > _gateway.domain: 27790+ PTR? 1.20.168.192.in-addr.arpa. (43)
04:57:29.860750 IP _gateway.domain > kali.42264: 27790 NXDomain 0/1/0 (78)
04:57:29.861150 IP kali.43676 > _gateway.domain: 36900+ PTR? 2.20.168.192.in-addr.arpa. (43)
04:57:29.862288 IP _gateway.domain > kali.43676: 36900 NXDomain 0/1/0 (78)
04:57:29.862621 IP kali.46656 > _gateway.domain: 19118+ PTR? 136.20.168.192.in-addr.arpa. (45)
04:57:30.638164 ARP, Request who-has _gateway tell kali, length 28
04:57:30.638292 ARP, Reply _gateway is-at 00:50:56:fd:dc:24 (oui Unknown), length 46
04:57:34.506805 IP 192.168.20.1.40816 > 192.168.20.255.40816: UDP, length 131
^C
11 packets captured
12 packets received by filter
1 packet dropped by kernel
```

Figura 1.2 - tcpdump

Zeek (Bro) – Un sistem de monitorizare a rețelei care nu doar capturează traficul, ci și analizează comportamentul acestuia pentru detectarea anomaliilor. Comparativ cu alte soluții, Zeek oferă un nivel mai ridicat de automatizare și posibilitatea de a genera rapoarte detaliate despre activitatea rețelei. Pe lângă rolul său de **capturare** a traficului la nivel de rețea, **Zeek (cunoscut anterior sub denumirea Bro)** se diferențiază printr-un **motor sofisticat de analiză a comportamentului**. Practic, Zeek corelează evenimentele de la nivelul aplicațiilor (HTTP, DNS, SSL etc.), transformând fluxurile de trafic în **informații structurate** care pot fi folosite pentru a identifica anomalii și potențiale amenințări de securitate.

Spre deosebire de alte unelte de tip IDS/IPS care se bazează preponderent pe semnături, Zeek adoptă o abordare **mai flexibilă**, analizând semnalele comportamentale și colectând metadate valoroase, cum ar fi cererile DNS sau parametrii tranzacțiilor HTTP. Această abordare face posibilă **detectarea atacurilor necunoscute** sau a tacticilor avansate folosite de atacatori, în condițiile în care semnăturile tradiționale nu ar oferi rezultate.

Beneficiind de un **sistem de scripturi** extensibil, Zeek permite personalizarea regulilor de monitorizare în funcție de nevoile organizației, precum și **integrarea cu alte soluții** de securitate (SIEM, platforme de automatizare etc.). Rapoartele generate de Zeek oferă o **imagine detaliată** asupra activității rețelei, fiind extrem de utile în investigațiile post-eveniment și în luarea deciziilor strategice de securitate. Astfel, Zeek reprezintă o componentă solidă într-o arhitectură de securitate modernă,

funcționând atât ca instrument de monitorizare continuă, cât și ca bază pentru detecția proactivă a amenințărilor.

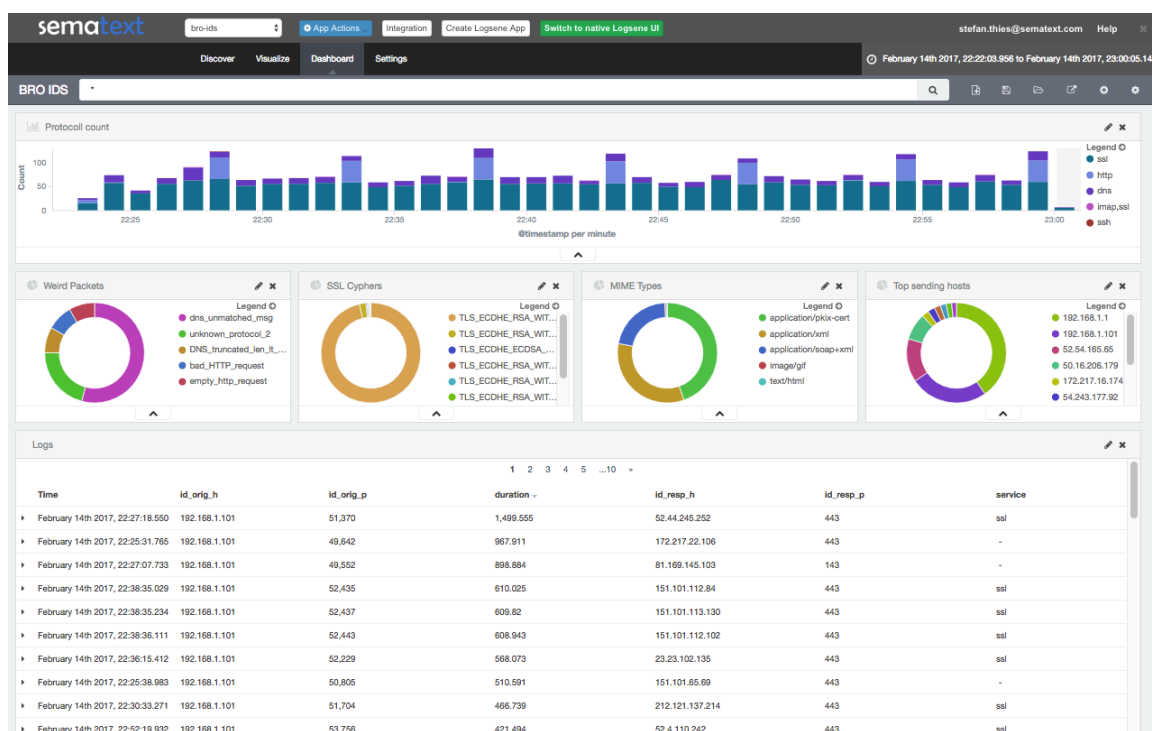


Figura 1.3 – Zeek

Suricata – Un sistem de detecție și prevenire a intruziunilor (IDS/IPS) care analizează traficul în timp real. Spre deosebire de Wireshark și tcpdump, care sunt axate pe captură, Suricata este capabilă să blocheze traficul suspect, având o abordare mai proactivă în securitatea rețelei.

Sistemul propus se bazează pe îmbinarea celor mai relevante caracteristici din diverse soluții de monitorizare și analiză a traficului (precum Wireshark, tcpdump sau Zeek), pentru a oferi o **platformă unitară și eficientă**. În loc să se limiteze la captarea pachetelor brute și la o analiză manuală extensivă, noul proiect introduce **funcționalități de filtrare și clasificare avansată**, concepute să reducă semnificativ timpul și expertiza specializată necesare în interpretarea datelor.

Prin agregarea și corelarea informațiilor din surse multiple, sistemul oferă **rapoarte sintetice** care sprijină luarea rapidă de decizii și permite **detecția proactivă** a potențialelor incidente de securitate. Astfel, **abordarea propusă** depășește limitele instrumentelor tradiționale, facilitând un **proces de analiză mai fluid și mai bine integrat** în cadrul infrastructurii de rețea.

1.3 Scopul, obiectivele și cerințele sistemului

Scopul principal al acestui proiect este dezvoltarea unui **sistem integrat pentru interceptarea și analiza traficului de rețea**, care să permită monitorizarea și detecția eficientă a activităților suspecte. Sistemul trebuie să ofere o metodă rapidă și automatizată de captură, filtrare și interpretare a pachetelor de date, contribuind astfel la îmbunătățirea securității cibernetice prin prevenirea și investigarea amenințărilor informatice.

- a) **Captarea traficului de rețea** – Dezvoltarea unui modul care să intercepteze pachetele de date în timp real folosind tehnologii precum **tcpdump**, **Wireshark**, **Zeek** sau **Suricata**;
- b) **Filtrarea și clasificarea pachetelor** – Implementarea unor algoritmi care să extragă și să analizeze doar informațiile relevante, eliminând traficul nesemnificativ;
- c) **Deteția activităților suspecte** – Identificarea anomaliilor și a potențialelor atacuri informatice prin metode bazate pe semnături și comportament;
- d) **Stocarea și gestionarea datelor** – Crearea unei baze de date sau a unui sistem de logare pentru păstrarea și analiza ulterioară a traficului interceptat;
- e) **Interfață de utilizator (GUI)** – Dezvoltarea unei interfețe intuitive care să permită utilizatorilor să vizualizeze datele capturate și să genereze rapoarte relevante;
- f) **Automatizarea procesului** – Integrarea unor funcționalități care să reducă necesitatea intervenției manuale, crescând astfel eficiența detecției.

Pentru ca sistemul să funcționeze eficient, acesta trebuie să îndeplinească o serie de cerințe esențiale, atât funcționale, cât și non-funcționale. Din punct de vedere funcțional, sistemul trebuie să fie capabil să captureze și să analizeze traficul de rețea în timp real, asigurând în același timp filtrarea și clasificarea pachetelor de date pe baza protocoalelor utilizate și a adreselor IP. De asemenea, este necesară integrarea unor mecanisme eficiente de detecție pentru identificarea anomaliilor sau a posibilelor atacuri informatice. Un alt aspect important este posibilitatea de stocare și raportare a datelor relevante, permițând astfel investigarea ulterioară a incidentelor de securitate. Pentru a facilita utilizarea, sistemul trebuie să dispună de o interfață grafică care să permită analiza și gestionarea traficului interceptat într-un mod intuitiv.

Pe lângă cerințele funcționale, sistemul trebuie să respecte și o serie de cerințe non-funcționale. Acesta trebuie să fie scalabil, astfel încât să permită extinderea funcționalităților în funcție de necesități. Performanța trebuie optimizată pentru a evita introducerea unor latențe semnificative în rețea, menținând un echilibru între eficiența analizelor și resursele consumate. Interfața utilizatorului trebuie să fie intuitivă și ușor de utilizat, astfel încât specialiștii în securitate să poată interpreta rapid datele obținute. Nu în ultimul rând, sistemul trebuie să respecte normele legale privind interceptarea și analiza traficului de rețea, asigurând conformitatea cu reglementările în vigoare.

2 MODELAREA ȘI PROIECTAREA SISTEMUL INFORMATIC

Modelarea și proiectarea sunt etape centrale în dezvoltarea oricărui sistem informatic, inclusiv a unui sistem integrat pentru interceptarea și analiza traficului de rețea. În această fază, se stabilește în detaliu modul în care sistemul își va îndeplini funcționalitățile, se definesc structurile interne și se proiectează interacțiunile dintre componente.

Modelarea datelor implică crearea unei reprezentări arhitecturale a funcționalităților și a fluxurilor de date. Aceasta se realizează prin identificarea entităților-cheie (de exemplu, module de interceptare a pachetelor, componente de analiză și raportare) și definirea relațiilor dintre acestea. Proiectarea se concentrează pe determinarea arhitecturii generale, a modulelor, interfețelor și tehnologiilor care vor asigura îndeplinirea cerințelor de securitate, performanță și scalabilitate.

Un instrument de mare importanță în procesul de modelare și proiectare este limbajul unificat de modelare (UML). Acesta oferă un set de diagrame și simboluri standardizate pentru descrierea și documentarea aspectelor funcționale și structurale ale sistemului. De exemplu:

- Diagrame de clasă - evidențiază entitățile (modul de interceptare, modul de analiză, modul de stocare a datelor) și relațiile dintre acestea;
- Diagrame de activitate - ilustrează fluxurile de lucru, cum ar fi procesul de captare a traficului și secvența operațiunilor de analiză;
- Diagrame de secvență - evidențiază interacțiunile dintre componente într-un anumit scenariu de interceptare sau de analiză.

Prin utilizarea UML, echipa de dezvoltare a proiectului și părțile interesate relevante (de exemplu, departamentul IT, specialiștii în securitate) obțin o viziune comună asupra modului în care va funcționa sistemul și asupra modului în care diferitele componente vor comunica între ele. Această abordare standardizată îmbunătățește comunicarea și coordonarea, contribuind la identificarea potențialelor incertitudini sau probleme într-un stadiu incipient. Diagramele UML pot servi, de asemenea, ca bază pentru testare, permițând identificarea și corectarea rapidă a erorilor și optimizarea performanței sistemului.

Prin urmare, modelarea și proiectarea sistemului integrat de interceptare și analiză a traficului de rețea oferă o structură clară și eficientă, asigurând îndeplinirea cerințelor funcționale și nefuncționale. Utilizarea în consecință a UML și a metodelor de proiectare adecvate asigură un proces de dezvoltare coerent, permițând ulterior implementarea, testarea și întreținerea ușoară a instalației.

2.1 Descrierea comportamentală a sistemului

Comportamentul sistemului integrat de interceptare și analiză a traficului de rețea poate fi înțeles cel mai bine prin ilustrarea modului în care componentele sale interacționează și colaborează pentru a realiza obiectivele de monitorizare și detecție. În această etapă, accentul cade pe definirea fluxurilor de

activități și pe evidențierea modului în care datele parcurg succesiv diferitele module ale sistemului. Pentru a asigura o reprezentare clară și standardizată a comportamentului, sunt utilizate diagrame UML (diagrame de activitate, diagrame de secvență), care fac posibilă identificarea pașilor logici și a interacțiunilor relevante.

Din punct de vedere comportamental, sistemul poate fi descris în trei etape majore:

Captarea traficului de rețea

- Modulul de interceptare preia pachetele care trec prin interfața de rețea, fie în mod promiscuu (prin care se colectează toate pachetele), fie pe baza unor filtre specifice;
- Interceptarea pachetelor se realizează în timp real, iar sistemul trebuie să gestioneze aceste date rapid, fără a introduce latențe considerabile în rețea;
- Acest flux inițial reprezintă punctul de intrare în sistem și declanșează următoarele activități, precum filtrarea și clasificarea datelor;

Analiza și filtrarea datelor

- Odată ce pachetele au fost colectate, modulul de analiză preliminară le segregă după criterii precum tipul protocolului (TCP, UDP, ICMP, HTTP etc.), sursa și destinația;
- Pachetele identificate ca fiind irelevante (de exemplu, trafic de rutină, fără semnificație pentru securitate) pot fi trecute direct în arhivă, pentru a nu încărca modulul de detecție;
- Pachetele considerate potențial suspecte sunt supuse unei analize mai aprofundate, cu ajutorul mecanismelor bazate pe semnături (compararea cu reguli cunoscute, după modelul Suricata) sau pe comportament (analiza anomaliilor, după modelul Zeek);
- Datele considerate importante sunt transferate către modulul de stocare și raportare, unde sunt păstrate pentru investigații ulterioare.

Prin intermediul diagramelor de activitate (Activity Diagrams) sunt reprezentate succesiunea și ramificarea fluxurilor ce descriu traseul pachetelor, de la momentul captării până la eventuala lor marcare ca „suspecte” sau „inofensive”. În plus, diagramele de secvență (Sequence Diagrams) oferă o perspectivă detaliată asupra modului în care componentele (modulul de captare, modulul de filtrare, modulul de detecție, baza de date și interfața de utilizator) interacționează în timp, punând în evidență ordinea exactă a apelurilor și a transferurilor de date.

2.1.1 Imaginea generală asupra sistemului

În **Figura 2.1** („Interacțiune cu utilizatorul/analistul”), se observă diagrama de caz de utilizare care ilustrează principalele acțiuni pe care un analist le poate întreprinde în cadrul aplicației pentru interceptarea și analiza traficului de rețea. Actorul, reprezentat de „Utilizatorul/Analistul”, interacționează cu sistemul printr-o serie de funcționalități esențiale:

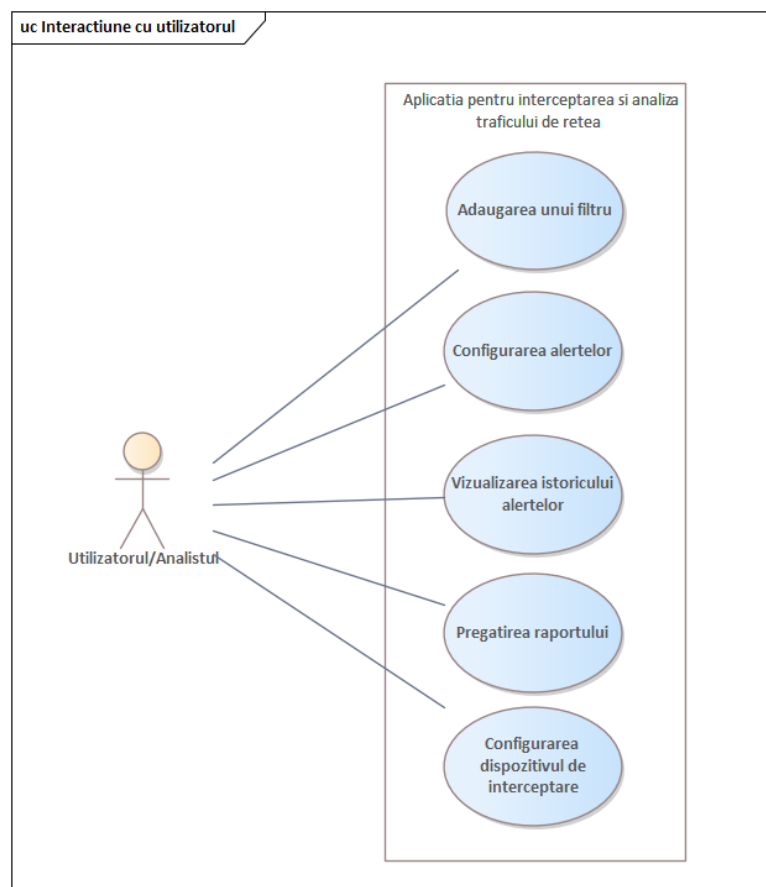


Figura 2.1 - Interacțiune cu utilizatorul/analistul

Funcționalitatea de adăugare a unui filtru oferă posibilitatea personalizării modului de captare a traficului, prin specificarea unor reguli precise de filtrare bazate, de exemplu, pe adrese IP, protocoale sau porturi, astfel încât analistul să poată izola informațiile relevante. Opțiunea de configurare a alertelor reprezintă o modalitate de stabilire a unor praguri și condiții ce declanșează notificări automate, utile pentru depistarea rapidă a activităților suspecte și a anomaliilor. În plus, vizualizarea istoricului alertelor permite accesul la evenimentele de securitate semnalate anterior, contribuind la identificarea tiparelor de atac și la evaluarea riscurilor asociate. Prin funcția de pregătire a raportului, se pot genera documente detaliate despre activitățile de monitorizare, statisticile traficului și potențialele incidente de securitate. De asemenea, configurarea dispozitivului de interceptare facilitează gestionarea setărilor tehnice ale echipamentului sau aplicației de captare, asigurând că datele sunt preluate eficient și corespunzător nevoilor de analiză.

2.1.2 Modelarea vizuală a fluxurilor

În **Figura 2.2**, este ilustrat un exemplu de diagramă de activitate care reflectă procesul de interceptare și partajare a traficului în funcție de filtrele definite. Diagrama pune în evidență pașii logici parcurși de aplicație imediat după inițierea filtrelor de captare: dacă sunt introduse reguli de filtrare, sistemul începe procesul de interceptare a traficului conform acestora, iar în absența unor reguli specifice, se optează pentru captarea completă (tot traficul). Ulterior, există o decizie suplimentară privind aplicarea unei filtrări personalizate înainte de partajarea fluxului de date; în cazul în care se optează pentru filtrarea

suplimentară, se partajează doar traficul care corespunde criteriilor stabilite, în timp ce, în absența acestor criterii, se partajează integral pachetele colectate. Această modelare vizuală a fluxurilor ajută la înțelegerea mai clară a logicii interne a sistemului și oferă o structură ușor de urmărit atunci când se dorește extinderea sau optimizarea procesului de monitorizare și prelucrare a datelor.

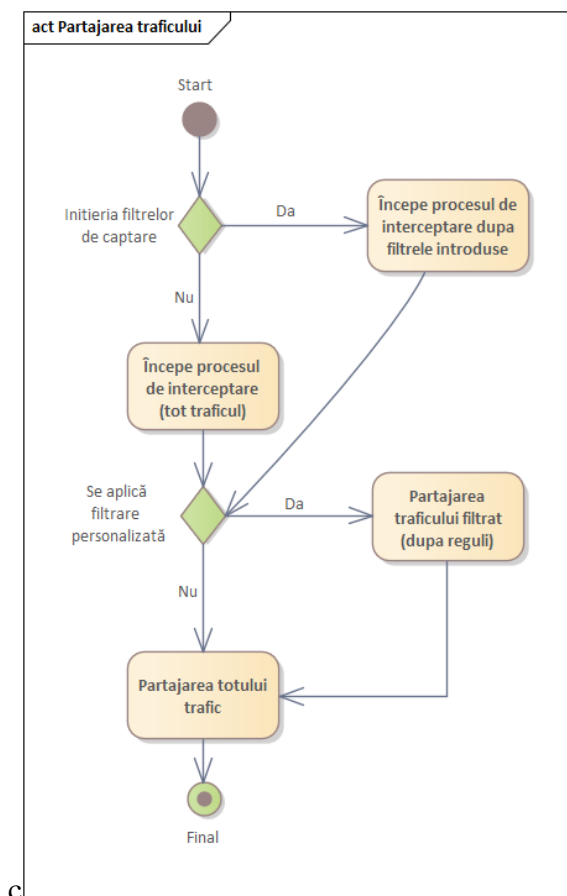


Figura 2.2 – Procesul de interceptare

Fluxul prezentat subliniază conceptul de „filtrare progresivă” în contextul interceptării datelor de rețea, demonstrând cum, în funcție de setările definite la pornirea aplicației, procesul poate evolua către o captare globală sau, dimpotrivă, către una orientată strict pe tipul de trafic vizat. Parcurgerea decizională din diagrama de activitate scoate în evidență flexibilitatea sistemului, permițând combinarea regulilor de filtrare pentru a obține un control granular asupra pachetelor interceptate. Astfel, dacă echipa de securitate activează anumite condiții de filtrare — cum ar fi protocoalele de comunicare suspecte sau adresele IP țintă — aceste reguli sunt aplicate din momentul inițial al captării, iar fluxul de date este redirecționat automat pentru analiză avansată. În situația opusă, fără introducerea niciunui filtru inițial, aplicația colectează toate pachetele, oferind o imagine de ansamblu cuprinzătoare asupra traficului. Prin prezentarea acestei diagrame, se arată cât de importantă este stabilirea de reguli flexibile încă de la începutul procesului, ceea ce permite adaptarea dinamică la diverse tipare de trafic și nevoi de securitate.

2.1.3 Descrierea scenariilor de utilizare a aplicației

În **Figura 2.4** se regăsește o diagramă de secvență care ilustrează etapele implicate în configurarea și gestionarea traficului de rețea, precum și schimbul de mesaje între actorul „Analist”, „Sistem”, sursa de „Trafic Rețea” și „Modulul Analiză”.

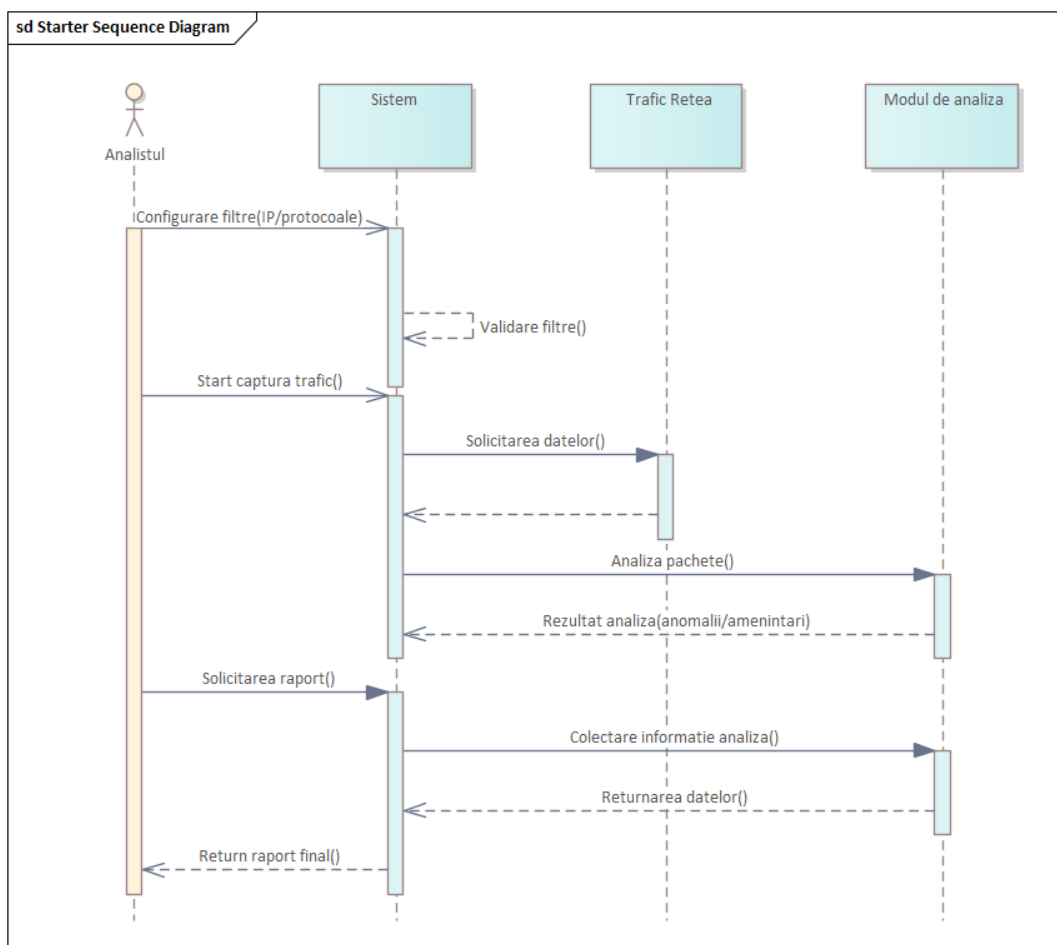


Figura 2.3 - Diagrama de secvență pentru configurarea, capturarea și analiza traficului de rețea

În primă instanță, analistul stabilește filtrele (adrese IP, protocoale) și pornește procesul de captură, moment în care sistemul validează setările și solicită date de la sursa de trafic. Pachetele interceptate sunt apoi transmise modulului de analiză, care verifică eventualele anomalii sau amenințări și returnează rezultatele. Ulterior, la cererea analistului, sistemul colectează informațiile obținute și afișează raportul final. Structura acestei diagrame de secvență subliniază interacțiunile logice dintre componente și evidențiază fluxul complet de acțiuni, de la definirea filtrelor până la prezentarea concluziilor despre securitatea traficului monitorizat.

2.2 Descrierea structurală a sistemului

În prima instanță, sistemul poate fi împărțit în componente precum modulul de captură (responsabil de interceptarea pachetelor de date), modulul de analiză (destinat verificării și detectării anomaliilor), bazele de date pentru logarea activităților și interfața de utilizator, prin care specialiștii configurează filtrele

și vizualizează rapoarte. Fiecare dintre aceste componente are un rol bine definit și un set de responsabilități clar delimitate, fiind conectate într-o manieră ierarhică sau prin relații de colaborare.

Pentru ilustrarea acestor legături, se utilizează de regulă diagrame UML adecvate, precum diagrame de componente (Component Diagrams) sau diagrame de clasă (Class Diagrams). Aceste reprezentări grafice asigură o imagine de ansamblu asupra modului în care sunt organizate părțile sistemului și asupra modului în care acestea interacționează pentru a atinge obiectivele de securitate. De pildă, o diagramă de componente poate arăta modulul de captură conectat la modulul de analiză, care, la rândul lui, interacționează cu baza de date și notifică interfața de utilizator în cazul detecției unor comportamente suspecte. Această abordare vizuală facilitează înțelegerea rapidă a structurii interne, fiind un punct de reper pentru echipa de dezvoltare și pentru părțile interesate.

2.2.1 Descrierea structurii statice a sistemului

În **Figura 2.4** este prezentată diagrama de clasă asociată sistemului de interceptare și analiză a traficului, evidențiind principalele entități, atribute și metode implicate. Modelul cuprinde clase precum „Utilizator”, „SistemInterceptare”, „Filtru”, „Pachet”, „ModulAnaliză”, „BazaDateLoguri”, „Raport”, „RezultatAnaliza” și „Alerta”, fiecare având un rol specific în ciclul de captură și procesare. Clasa „Utilizator” administrează configurațiile și rapoartele, în timp ce „SistemInterceptare” inițiază și oprește capturarea, gestionând regulile de filtrare prin obiecte de tip „Filtru”. Pachetele de date interceptate sunt modelate prin clasa „Pachet”, care reține informațiile despre sursa, destinația și protocolul folosit.

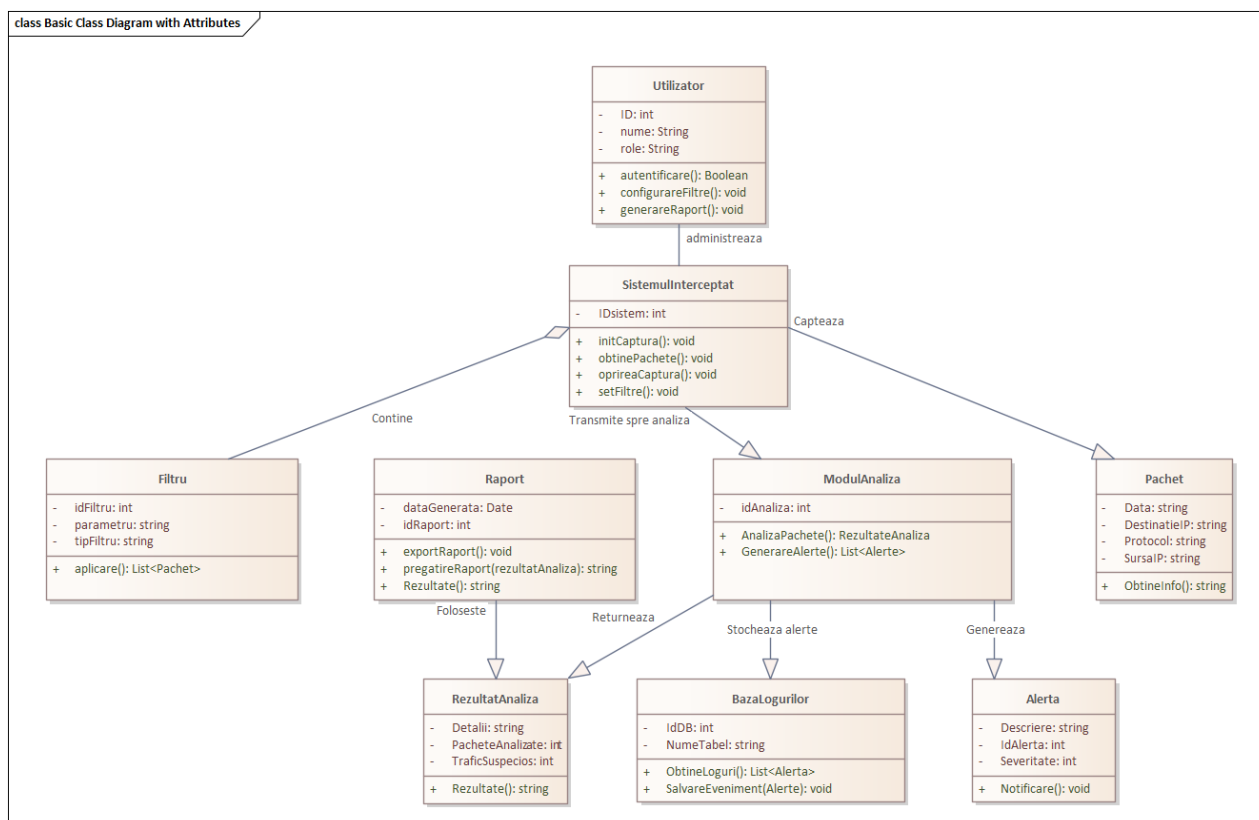


Figura 2.4 - Diagrama de clasă a sistemului de interceptare și analiză a traficului.

Analiza pachetelor are loc în cadrul clasei „ModulAnaliză”, rezultatele fiind centralizate în instanțe de tip „RezultatAnaliza” și, dacă e cazul, înregistrate sub forma unor „Alerta” în „BazaDateLoguri”. Generarea documentației se efectuează prin intermediul clasei „Raport”, responsabilă de compilarea și exportul informațiilor esențiale, astfel încât procesul de monitorizare și investigare a traficului să fie susținut în mod unitar și eficient.

Relațiile dintre clase evidențiază modul în care sistemul gestionează capturarea, analiza și raportarea traficului de rețea. Clasa *Utilizator* interacționează cu *SistemInterceptare* pentru a configura filtrele și a genera rapoarte. *SistemInterceptare* conține metode pentru inițierea și oprirea capturii de date, transmițând pachetele către *ModulAnaliză* pentru procesare.

Clasa *ModulAnaliză* analizează pachetele interceptate și generează alerte pentru activități suspecte. Rezultatele analizei sunt utilizate pentru generarea rapoartelor, care sunt apoi stocate în *BazaLogurilor*. Clasa *Filtru* permite selecția pachetelor relevante, iar *Raport* și *RezultatAnaliza* oferă o analiză detaliată a traficului. *BazaLogurilor* are rolul de a stoca evenimentele și alertele pentru referință ulterioară.

2.2.2 Relațiile de dependență între componentele sistemului

Diagrama ilustrează procesul de autentificare și autorizare într-un sistem software, utilizând o mașină de stare UML cu un stat compozit. Procesul începe dintr-un nod inițial și se desfășoară prin mai multe etape esențiale.

Prima etapă implică accesarea paginii de logare, unde utilizatorul este invitat să introducă datele necesare autentificării. În acest moment, sistemul intră într-un stat compozit denumit „Sistemul”, care conține mai multe stări interne.

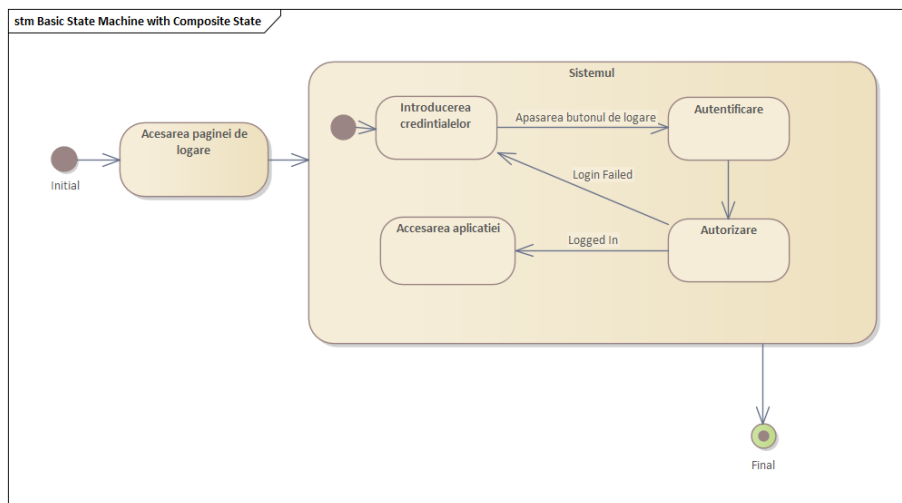


Figura 2.5 - Diagrama de stare pentru procesul de logare

În interiorul acestui stat, utilizatorul introduce credențialele și apasă butonul de logare, moment în care sistemul declanșează procesul de autentificare. Dacă datele sunt incorecte, tranziția către „Introducerea credențialelor” este reluată printr-un eveniment de eșec al autentificării („Login Failed”). Dacă

autentificarea are succes, sistemul trece la procesul de autorizare, verificând drepturile de acces ale utilizatorului.

După autorizare, utilizatorul poate accesa aplicația, finalizând astfel procesul de autentificare și intrând într-un nou stat funcțional. Diagrama se încheie cu un nod final, indicând terminarea procesului și tranziția către starea normală de utilizare a aplicației.

3 REALIZAREA SISTEMULUI

Sistemul integrat pentru interceptarea și analiza traficului de rețea a fost dezvoltat pentru a răspunde nevoii de monitorizare, captură și examinare a pachetelor de date transmise în rețelele locale, oferind o soluție practică și eficientă atât pentru analiza de securitate, cât și pentru scopuri educaționale sau administrative.

Aplicația este compusă din trei componente principale: motorul de captură și analiză a traficului de rețea, interfața grafică de utilizator (GUI) și sistemul de stocare a datelor, toate dezvoltate în limbajul Python. Pentru partea de captură și procesare a pachetelor, s-au utilizat biblioteci specializate precum `scapy` și `socket`, care permit interceptarea pachetelor în timp real și extragerea informațiilor esențiale (adrese IP, porturi, protocoale, dimensiuni, timestamp-uri). Interfața grafică a fost construită folosind biblioteca **Tkinter**, oferind utilizatorului o experiență intuitivă, stilizată sub forma unei ferestre de login.

Toate datele capturate sunt stocate într-o **bază de date locală**, care permite păstrarea istoricului sesiunilor de monitorizare și realizarea de căutări și analize ulterioare. Această bază de date asigură persistența datelor și facilitează exportul informațiilor în diverse formate, cum ar fi `.pcap` sau `.txt`. Această abordare completă, bazată pe Python, oferă flexibilitate în dezvoltare, portabilitate pe diferite sisteme de operare și o bază solidă pentru extinderea funcționalităților în viitor.

3.1 Structura sistemului

Sistemul a fost construit într-o manieră modulară, cu o arhitectură clar definită, care să permită atât o dezvoltare coerentă, cât și o posibilitate facilă de extindere ulterioară. Întregul proiect este implementat în limbajul Python, integrând mai multe componente care colaborează pentru a asigura funcționalitatea completă a aplicației.

Captura traficului de rețea este realizată cu ajutorul unor biblioteci specializate precum `scapy` și `socket`, care permit interceptarea pachetelor în timp real. Informațiile extrase din aceste pachete – adrese IP, porturi, protocoale, dimensiuni și timestamp-uri – sunt prelucrate imediat de un motor de analiză care clasifică datele în funcție de criterii specifice și evidențiază protocoalele implicate.

Datele analizate sunt afișate prin intermediul unei interfețe grafice dezvoltate în Tkinter. Aceasta a fost concepută sub forma unei ferestre stilizate, inspirată de un ecran de login, oferind o experiență vizuală prietenoasă și intuitivă. Utilizatorul poate selecta interfața de rețea dorită, poate porni sau opri capturarea traficului și are acces la informațiile prezentate într-un tabel clar structurat.

Pentru a asigura persistența datelor și posibilitatea de a reveni ulterior asupra informațiilor capturate, aplicația include o bază de date locală în care sunt stocate toate sesiunile de monitorizare. Aceasta permite interogarea, filtrarea, precum și exportul datelor în formate precum `.pcap` sau `.txt`.

În plus, sistemul include un mecanism care extrage denumirile reale ale interfețelor de rețea disponibile pe sistem, prin integrarea cu tehnologia WMI, îmbunătățind astfel considerabil experiența utilizatorului în etapa de selecție a sursei de trafic.

3.2 Integrarea serviciilor externe

Pentru a extinde funcționalitățile aplicației și pentru a simplifica anumite procese tehnice, sistemul integrează servicii externe care contribuie la îmbunătățirea experienței utilizatorului și la creșterea eficienței operaționale. Una dintre componentele externe importante este Windows Management Instrumentation (WMI), utilizată pentru a obține informații detaliate despre interfețele de rețea disponibile pe sistemul de operare. Această integrare oferă posibilitatea afișării denumirilor reale ale interfețelor, în locul identificatorilor criptici, facilitând astfel alegerea corectă a sursei de captură de către utilizator.

Un alt exemplu de integrare externă este reprezentat de utilizarea unor biblioteci și instrumente open-source specializate în manipularea pachetelor de rețea. Astfel, biblioteca scapy permite nu doar capturarea pachetelor, ci și analiza detaliată a conținutului acestora, iar socket este utilizată pentru stabilirea conexiunii directe cu interfețele de rețea. Aceste biblioteci externe sunt esențiale pentru buna funcționare a aplicației și pentru oferirea unei game largi de funcționalități în cadrul unei soluții locale.

În ceea ce privește gestionarea și persistența datelor, aplicația se bazează pe un sistem de baze de date local, care permite înregistrarea, accesarea și exportarea datelor capturate. Acest mecanism funcționează independent, fără a fi nevoie de o conexiune la servicii de cloud, însă poate fi adaptat ulterior pentru integrarea cu soluții externe de stocare sau analiză, în funcție de nevoile utilizatorului sau ale instituției care implementează sistemul.

Prin aceste integrări, aplicația reușește să combine capabilitățile mediului Python cu avantajele oferite de resursele externe, fără a compromite portabilitatea, securitatea sau independența sistemului. Această deschidere către extensibilitate tehnologică oferă un potențial ridicat pentru dezvoltări viitoare și adaptări în contexte diverse de utilizare.

3.3 Implementarea cerințelor funcționale

Realizarea cerințelor funcționale a presupus definirea clară a comportamentului pe care aplicația trebuie să îl aibă în raport cu utilizatorul și cu traficul de rețea observat. Procesul a început cu delimitarea precisă a funcțiilor esențiale: capturarea traficului, vizualizarea în timp real, gestionarea sesiunilor, salvarea datelor și interacțiunea intuitivă printr-o interfață grafică.

Captura de trafic a fost implementată ca o acțiune asincronă, rulând într-un fir separat de execuție, astfel încât interfața grafică să nu fie blocată în timpul procesării pachetelor. Această decuplare între logică și prezentare asigură o experiență fluidă pentru utilizator, indiferent de volumul de date procesate. Captura poate fi pornită și oprită în orice moment, fără întreruperi bruște sau pierderi de date, datorită sincronizării controlate între fluxurile de execuție.

Vizualizarea pachetelor în interfață este realizată prin actualizarea unui tabel dinamic, unde fiecare rând reprezintă un pachet interceptat. Pentru optimizarea performanței, actualizările sunt realizate în loturi mici, astfel încât redarea vizuală să nu încetinească pe măsură ce traficul se intensifică. Informațiile afișate sunt normalizate pentru lizibilitate, iar protocoalele sunt evidențiate distinct, facilitând recunoașterea rapidă a tipurilor de trafic.

Sistemul de salvare a sesiunilor în baza de date a fost conceput astfel încât fiecare captură să fie asociată cu un identificator unic, permițând gestionarea mai multor sesiuni în paralel. Fiecare pachet este salvat cu toate metadatele esențiale, iar structura bazei de date permite interogări flexibile în funcție de protocol, IP, port sau interval de timp. Acest nivel de granularitate în organizarea datelor a fost implementat pentru a răspunde cerinței de analiză post-eveniment.

Interfața grafică, pe lângă funcțiile de control, include și validări interne, mesaje de feedback și blocaje logice care previn erorile utilizatorului. Spre exemplu, captura nu poate fi pornită fără selectarea unei interfețe, iar utilizatorul este notificat vizual în cazul în care accesul la rețea este restricționat de sistemul de operare sau de drepturile de utilizator.

Fiecare cerință funcțională a fost transpusă în cod printr-o abordare centrată pe stabilitate, ușurință în utilizare și extensibilitate. Această implementare atent planificată transformă aplicația dintr-un simplu instrument de captură într-un mediu complet pentru observarea și studierea comportamentului rețelelor.

4 DOCUMENTAREA SISTEMULUI

Acest capitol descrie modul de utilizare al aplicației Sentinel, prezentând elementele vizuale, componentele funcționale și modul în care aplicația interacționează cu utilizatorul și cu baza de date. Documentarea este susținută prin capturi de ecran care exemplifică interfețele grafice și operațiunile esențiale ale aplicației.

4.1 Interfața de autentificare

Pagina de autentificare reprezintă punctul de acces principal în aplicație pentru utilizatorii existenți. Aceasta a fost concepută într-un stil modern, minimalist, cu accent pe lizibilitate și claritate. Interfața conține două câmpuri de intrare: unul pentru adresa de email și unul pentru parola utilizatorului. Lângă fiecare câmp este plasată o pictogramă sugestivă pentru tipul de informație solicitată.

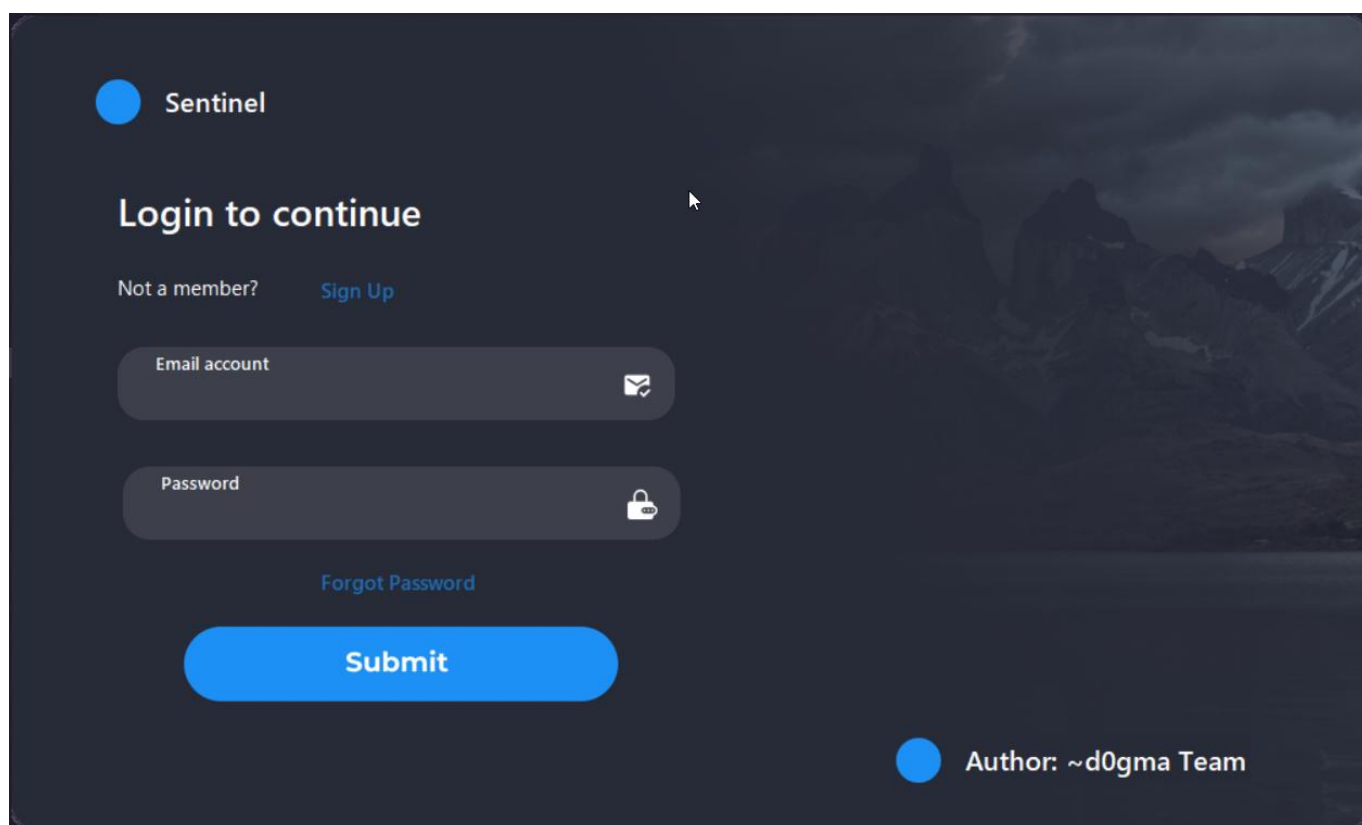


Figura 4.1 – Interfața de autentificare în aplicație

În partea superioară a paginii este plasat un logo stilizat și numele aplicației. Sub formularul de login există opțiunea de a accesa pagina de înregistrare pentru utilizatorii care nu dețin cont, precum și linkul de resetare a parolei pentru utilizatorii care și-au uitat acreditivile de conectare.



4.2 Interfața de înregistrare


Interfața de înregistrare permite crearea unui cont nou de utilizator. Aceasta este accesibilă prin linkul "Sign Up" aflat pe pagina de logare. Formularul de înregistrare solicită utilizatorului completarea următoarelor câmpuri: prenume, nume, adresă de email, parolă și confirmarea parolei. Figura 4.1 ilustrează această interfață de logare.



Sentinel

Create new account

Already a member? [Login](#)

First name  Last name 

Email account 

Password  Confirm Password 

Submit

Author: ~d0gma Team

Figura 4.2 – Formularul de creare a unui cont nou în aplicația Sentinel

Pentru a asigura o experiență intuitivă, fiecare câmp este însoțit de o pictogramă reprezentativă, iar layout-ul este echilibrat și ușor de urmărit. Butonul de trimitere este stilizat modern, colorat într-o nuanță albastră, care atrage atenția utilizatorului și confirmă acțiunea de înregistrare. Această interfață este ilustrată în Figura 4.2.

4.3 Interacțiunea cu baza de date

După completarea formularului de înregistrare, datele utilizatorului sunt transmise către backend, unde sunt validate și procesate. Aplicația utilizează o bază de date PostgreSQL, iar datele sunt inserate într-o tabelă denumită `users`, care conține următoarele câmpuri:

- `id` – identificator unic generat automat (UUID);
- `first_name` – prenumele utilizatorului;
- `last_name` – numele de familie;
- `email` – adresa de email;
- `password` – parola criptată cu un algoritm SHA-512.

Această structură oferă un echilibru între securitate și eficiență. În Figura 4.3 este prezentată o interogare SQL care returnează conținutul tabelii `users`, confirmând existența unui utilizator înregistrat.

```
user_database=> select * from users;
```

id	first_name	last_name	email	password
79bc4fd8-153e-4a3f-9ffb-ca74918837b0	qq	qq	qq	d5ce2b19fbd14a25deac948154722f33efd37b369a32be8f03ec2be8ef7d3a5

(1 row)

Figura 4.3 - Conținutul bazei de date PostgreSQL după înregistrarea unui utilizator

4.4 Interfața principală a aplicației

După autentificare, utilizatorul este direcționat către interfața principală a aplicației Sentinel. Această interfață centralizează accesul către modulele disponibile: `PacketSentinel` (captură trafic rețea), `HashSentinel` (atacuri brute-force asupra hash-urilor) și `DirSentinel` (scanare directoare).

Interfața are un aspect întunecat, profesional, aliniat cu tema generală de securitate. În partea centrală a ferestrei este plasată zona de control pentru capturarea traficului. Utilizatorul poate selecta interfața de rețea activă dintr-un meniu dropdown, apoi poate porni sau opri procesul de captură. Datele capturate sunt afișate în timp real într-un terminal grafic integrat. Figura 4.4 prezintă interfața principală înainte de inițierea capturii.

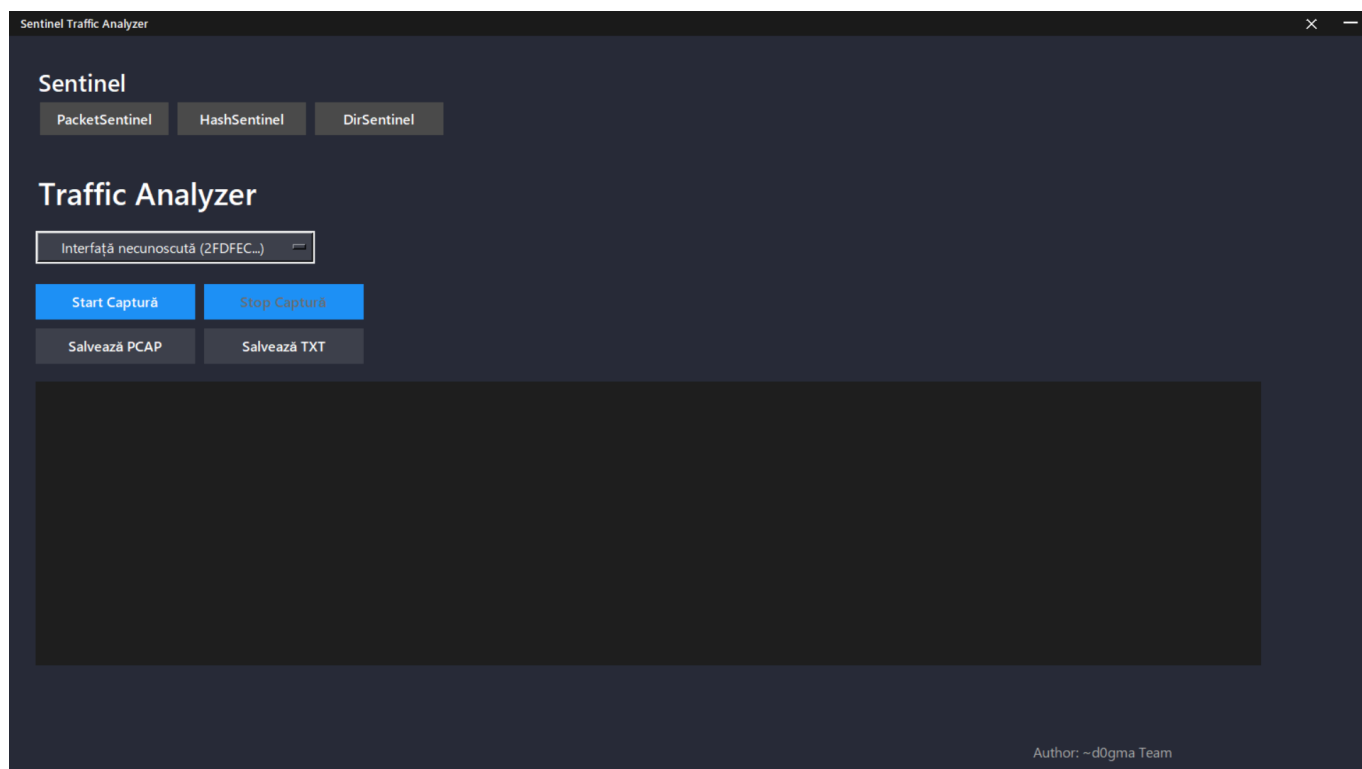


Figura 4.4 - Interfața principală a aplicației

4.5 Capturarea traficului de rețea

După selectarea interfeței de rețea și inițierea procesului de captură, aplicația interceptează în timp real pachetele care circulă prin acea interfață. Pentru fiecare pachet capturat, aplicația extrage și afișează următoarele informații:

- Ora capturii;
- Adresa IP sursă și destinație;
- Protocolul utilizat (TCP/UDP);
- Portul sursă și destinație;
- Dimensiunea pachetului.

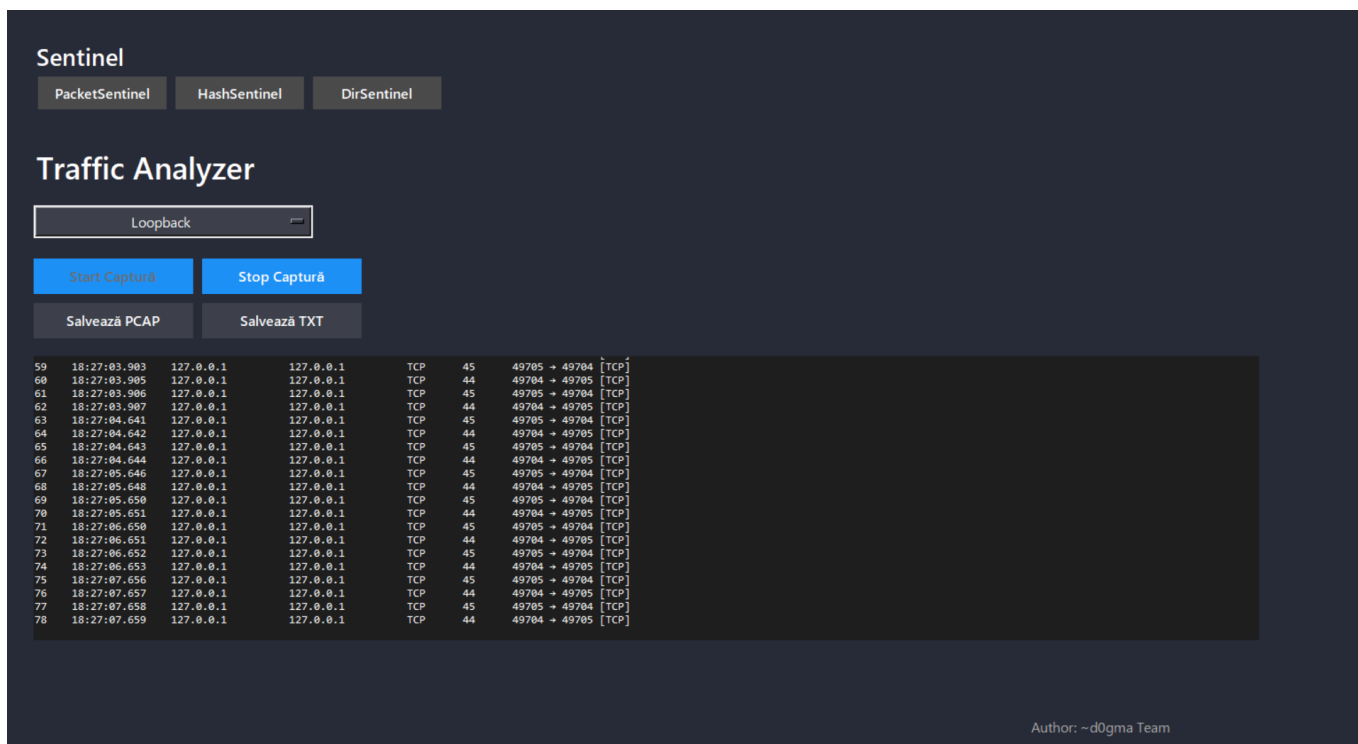


Figura 4.5 - Capturarea traficului în timp real și afișarea în interfață

Utilizatorul poate opri oricând captura, iar rezultatele pot fi salvate în format `.pcap` (compatibil Wireshark) sau `.txt`, pentru analiză ulterioară. În Figura 4.5 este ilustrat un exemplu de sesiune activă de captură pe interfața Loopback.

5 Analiza de risc, evaluarea și impactul proiectului

În cadrul realizării modulului PacketSentinel, care are rolul de a intercepta, afișa și analiza traficul de rețea în timp real, au fost identificate și gestionate o serie de riscuri relevante pentru aplicațiile de tip analizator de trafic.

5.1 Riscuri tehnice

- **Acces insuficient la interfețele de rețea (permisiuni)**

Un risc major a fost imposibilitatea de a accesa corect interfețele de rețea sub Windows, din cauza lipsei de permisiuni administrative sau a unor servicii Npcap oprite. Pentru a rezolva acest aspect, aplicația verifică interfețele disponibile cu Scapy și oferă feedback utilizatorului în caz de eroare;

- **Suprasarcină în captură și pierdere de pachete**

Captura rapidă și voluminoasă poate depăși capacitatea de procesare, ducând la pierderea unor pachete. Pentru a preveni acest risc, aplicația utilizează thread-uri separate pentru captură și afișare și oferă posibilitatea filtrării prin expresii BPF;

- **Filtrare inexactă sau absența interpretării pachetelor**

Filtrul introdus de utilizator (ex: tcp, udp port 53) putea duce la rezultate irelevante sau incomplete. Acest risc a fost redus prin afișarea dinamică a pachetelor capturate și prin permiterea ajustării filtrelor în timp real;

- **Erori în interpretarea interfețelor de rețea**

În lipsa unei mape clare între denumirile Scapy și cele afișate în sistem, exista riscul selectării greșite a interfeței. Acest risc a fost eliminat prin implementarea unei funcții de corelare între descrierea interfeței, adresa IP și numele tehnic (GUID).

5.2 Riscuri de utilizare

- **Utilizarea necorespunzătoare în medii neautorizate**

Aplicația, prin natura ei, poate fi utilizată pentru interceptarea datelor din rețele în care utilizatorul nu are drepturi. Acest risc este adresat prin menționarea clară a scopului educațional și prin faptul că aplicația nu decodează payloadul pachetelor (nu analizează conținutul datelor propriu-zise).

5.3 Funcționalități implementate

Modulul PacketSentinel oferă următoarele funcționalități validate:

- **Listarea interfețelor disponibile**

Se afișează denumirea clară (ex: "Wi-Fi Intel AX200") împreună cu adresa IP aferentă, facilitând alegerea interfeței corecte.

- **Capturarea traficului în timp real**

Traficul este interceptat cu ajutorul bibliotecii Scapy și este afișat într-o fereastră text în interfața grafică, cu suport pentru filtre BPF (ex: tcp, port 80, icmp, etc.).

- **Afișarea grafică a protocoalelor capturate**

Este integrat un grafic în timp real care arată proporția protocoalelor (TCP, UDP, Other), oferind o viziune de ansamblu asupra rețelei.

- **Calcularea ratei de trafic**

Aplicația afișează rata de trafic exprimată în pachete pe minut, actualizată periodic, însoțită de pictograme informative.

- **Salvarea datelor capturate**

Se poate salva traficul în format .pcap, pentru analiză ulterioară în Wireshark, precum și în format .txt, pentru arhivă sau documentație.

5.4 Evaluarea performanței

Testele efectuate au demonstrat că aplicația:

- poate intercepta pachete pe rețele Wi-Fi și Ethernet;
- nu consumă excesiv resurse în timpul capturii;
- poate funcționa în paralel cu alte instrumente de rețea;
- menține o interfață responsivă și clară.

5.5 Impactul educațional

PacketSentinel este o unealtă extrem de utilă în învățarea și înțelegerea modului în care funcționează traficul de rețea. Aplicația permite studenților:

- să observe traficul generat de aplicații reale (navigare, ping, DNS, etc.);
- să testeze ipoteze de rețea în medii de laborator;
- să vizualizeze pachete live fără a avea nevoie de Wireshark sau instrumente complexe.

5.6 Impactul în cercetare

Proiectul poate fi extins pentru:

- analiză de tip port scanning și detecție de atacuri simple (SYN flood, ARP spoofing);
- integrare cu sisteme de alertare automată în caz de trafic anormal;
- studii privind comportamentul rețelelor în scenarii simulate.

5.7 Impactul profesional

Cunoștințele și abilitățile dezvoltate în timpul realizării acestui modul sunt direct aplicabile în:

- domenii precum administrarea rețelelor, DevSecOps, testare de penetrare;
- poziții care implică analiză de trafic, monitorizare rețele și troubleshooting;
- securitatea rețelelor și detectarea timpurie a amenințărilor.

BIBLIOGRAFIE

- [1] Bejtlich, R. (2005). *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Addison-Wesley.
- [2] Orebaugh, A., Ramirez, G., & Beale, J. (2006). *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress.
- [3] Scapy Documentation. (2024). [Online]. Available: <https://scapy.readthedocs.io/>
- [4] Kurose, J. F., & Ross, K. W. (2021). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.
- [5] Paxson, V. (1999). "Bro: A System for Detecting Network Intruders in Real-Time," *Computer Networks*, vol. 31, pp. 2435–2463.
- [6] Suricata IDS. *Suricata – Open Source Threat Detection Engine*. [Online]. Available: <https://suricata.io/>
- [7] Nmap Network Scanning. (2024). *Nmap Guide to Network Discovery and Security Scanning*. [Online]. Available: <https://nmap.org/book/>
- [8] Python Software Foundation. (2024). *Scapy Packet Manipulation Library*. [Online]. Available: <https://pypi.org/project/scapy/>
- [9] RFC 791 – *Internet Protocol*. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc791>
- [10] RFC 793 – *Transmission Control Protocol*. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc793>
- [11] Go Authors. (2024). *Official Go Documentation*. [Online]. Available: <https://golang.org/doc/>
- [12] Tkinter GUI Docs. (2024). *Python Tkinter GUI Reference*. [Online]. Available: <https://docs.python.org/3/library/tkinter.html>
- [13] Stallings, W. (2017). *Network Security Essentials* (6th ed.). Pearson.
- [14] Wireshark. (2024). *Wireshark User Guide*. [Online]. Available: <https://www.wireshark.org/docs/>