Universitatea “Ştefan cel Mare” Suceava
Inginerie Electronică, Telecomunicaţii și Tehnologii informaționale

# TEZĂ DE DOCTORAT

# Contribuții la îmbunătățirea sistemelor de securitate informațională pe diverse niveluri de comunicație

## Rezumat

Conducător științific,
prof. univ. dr. ing. Adrian GRAUR

Doctorand,
ing. Andrei-Daniel Tudosi

Suceava, aprilie 2024

# Cuprins

Capitolul 1: Introducere ..... 1
Capitolul 2: Studiu privind stadiul actual al securității informaționale ..... 3
Capitolul 3: Procesul de virtualizare ..... 5
Capitolul 4: Conceptul propus – firewall distribuit ..... 6
Capitolul 5: Soluții propuse și implementate ..... 16
Capitolul 6: Evaluarea performanțelor soluțiilor propuse ..... 51
Capitolul 7: Concluzii generale, contribuții și perspective ..... 52
Bibliografie ..... 55

# Cuvinte cheie

Aplicație de programare a interfețelor, Automatizare, Algoritm de optimizare, Analiză a riscurilor, API-uri, Comunicare securizată în sisteme, Clasificarea e-mail-urilor, Detectare de phishing, Firewall, Firewall distribuit, Managementul politicii de securitate, Managementul firewall-ului, Interfețe de rețea, IDS, Învățare automată, Mașini virtuale, Managementul rețelelor, Platformă de virtualizare, Politici de securitate, Protocol SMTP, Redirecționare, Routing, Scapy, Servere de rețea, Securitatea rețelei, Snort, VLAN.

# Capitolul 1: Introducere

Cercetarea are la bază securitatea informațională, evidențiind evoluția și implementarea soluțiilor de securitate care au ca fundament *firewall*-ul distribuit [1]. Acesta reprezintă un concept în domeniul securității informaționale care vizează protejarea rețelelor împotriva amenințărilor cibernetice. El recurge la distribuirea funcționalităților sale pe mai multe noduri sau dispozitive în rețeaua fizică.

Prezentarea problemei de cercetare subliniază necesitatea abordării avansate a securității informaționale în fața atacurilor cibernetice tot mai complexe. *Firewall*-ul tradițional, instalat pe un singur dispozitiv, poate fi depășit de atacatorii determinați [2], deoarece aceștia pot găsi vulnerabilități în sistemul de apărare de graniță și pot evita detectarea. Importanța temei și relevanța acesteia în domeniul securității informaționale sunt evidențiate prin necesitatea dezvoltării unor soluții de securitate eficiente și creative care să protejeze rețelele împotriva amenințărilor persistente avansate. Acestea reprezintă atacuri elaborate, care vizează în mod specific organizații sau ținte de înaltă valoare, atacuri ce urmăresc să rămână neobservate în rețea pentru a-și atinge scopurile.

Obiectivele cercetării sunt stabilite pentru a aborda problemele identificate și a dezvolta soluții eficiente pentru protejarea rețelelor împotriva acestor atacuri și a altor amenințări cibernetice. Obiectivele generale includ dezvoltarea și implementarea unor arhitecturi de rețea securizate bazate pe *firewall* distribuit, îmbunătățirea detecției și contracarării atacurilor și optimizarea performanțelor sistemelor de securitate. Obiectivele specifice includ dezvoltarea unor metode de grupare a fluxurilor de rețea pentru detectarea diferitelor tipuri de atacuri, proiectarea și implementarea unor sisteme automate de gestionare și actualizare a regulilor de securitate în timp real și evaluarea performanțelor soluțiilor propuse.

Metodologia folosită în această teză a fost selectată pentru a asigura o abordare sistematică și riguroasă a obiectivelor propuse. Aceasta include etapele de cercetare, colectarea și analiza datelor, utilizarea de metode și tehnici specifice pentru realizarea cercetării, precum și evaluarea și validarea soluțiilor propuse.

În cadrul cercetării se utilizează metode și tehnici precum analiza datelor obținute în urma testării soluțiilor propuse. Acest proces implică colectarea de date despre performanțele soluțiilor, analiza și interpretarea acestor date pentru a evalua eficacitatea și eficiența soluțiilor propuse. Prin implementarea și evaluarea soluțiilor propuse, cercetarea aduce contribuții semnificative în domeniul securității informaționale, cu accent pe conceptul de distribuție a protecției în rețea. Evaluarea performanțelor soluțiilor propuse permite determinarea nivelului de protecție și eficiență al acestora, și oferă date relevante pentru îmbunătățirea și optimizarea soluțiilor existente.

Prezentarea problemei de cercetare evidențiază importanța și relevanța abordării *firewall*-ului distribuit în contextul securității rețelelor. Cu toate că *firewall*-ul este recunoscut ca fiind o soluție promițătoare pentru protecția rețelelor împotriva amenințărilor cibernetice, există încă nevoia de a investiga și înțelege mai bine performanțele și funcționalitățile cheie ale acestui sistem.

Tema cercetării privind *firewall*-ul distribuit prezintă o importanță esențială în domeniul securității informaționale, având ca obiectiv protejarea rețelelor și datelor împotriva amenințărilor

cibernetice. *Firewall*-ul reprezintă o componentă critică a sistemelor de securitate, asigurând controlul și filtrarea traficului într-o rețea distribuită.

Obiectivul principal al acestei cercetări constă în analiza detaliată a performanțelor și a funcționalităților esențiale ale unui sistem de securitate în rețea, cu accent pe identificarea potențialelor îmbunătățiri în domeniul securității rețelelor.

În vederea realizării obiectivului general al cercetării, s-au identificat și urmărit mai multe obiective specifice: în primul rând, s-a analizat literatura de specialitate în domeniul auditului securității pentru a înțelege mai bine metodele și tehnicile utilizate în testarea securității soluției propuse, obținând astfel o bază solidă de cunoștințe teoretice și practice. Apoi, s-au investigat performanțele și eficiența rețelei propuse prin utilizarea unor metode și instrumente specifice, cum ar fi testarea manuală cu instrumente integrate în *Kali Linux* și testarea automatizată cu ajutorul unor instrumente precum *Nessus*, cu scopul de a identifica potențialele vulnerabilități și de a evalua rezistența la atacuri. De asemenea, s-au identificat și propus soluții pentru vulnerabilitățile *software*-ului terților utilizat în cadrul sistemului de securitate, asigurând astfel securitatea acestuia și protejând rețelele împotriva amenințărilor cibernetice. În final, s-au analizat rezultatele obținute în urma cercetării și s-au discutat implicațiile acestora în domeniul securității informaționale, evaluând critic rezultatele în contextul literaturii existente, al practicilor curente, și evidențiind contribuțiile aduse de cercetare și posibilele direcții pentru dezvoltarea ulterioară a securității informaționale.

Metodologia de cercetare adoptată în această teză reprezintă o abordare mixtă, integrând atent aspecte teoretice și experimentale pentru a oferi o perspectivă comprehensivă și riguroasă asupra problematicii analizate, concentrându-se în mod specific pe evaluarea unui sistem de securitate dintr-o rețea. Această abordare holistică servește drept fundament solid pentru analiza detaliată și evaluarea performanțelor *firewall*-ului în contextul securității cibernetice. Astfel, metodologia presupune investigarea exhaustivă a literaturii specializate pentru a obține o perspectivă detaliată asupra conceptelor, teoriilor și metodologiilor relevante în domeniul auditului securității și evaluării soluțiilor de securitate dintr-o rețea. Această analiză a literaturii de specialitate nu doar evidențiază nivelul actual al cunoștințelor în domeniu, ci și facilitează identificarea practicilor și tehnicilor cu cea mai mare eficiență și robustețe utilizate în auditul securității. Aceste informații extrase din literatură constituie baza esențială pentru dezvoltarea și aplicarea ulterioară a metodelor specifice de cercetare.

Importanța subiectului în cadrul securității informaționale derivă din imperativul dezvoltării soluțiilor de securitate de vârf, necesare pentru a contracara provocările din ce în ce mai complexe întâlnite în mediul cibernetic contemporan. Cercetarea în acest domeniu poate contribui la dezvoltarea de metode și tehnologii progresive, care să permită o protecție mai eficientă și scalabilă a rețelelor. Rezultatele obținute în urma cercetării pot avea un impact semnificativ asupra practicilor de securitate. Prin identificarea și abordarea aspectelor critice ale *firewall*-ului distribuit, se pot propune soluții creative care să asigure o protecție mai robustă împotriva amenințărilor cibernetice.

Prin aplicarea acestei metodologii, cercetarea furnizează o înțelegere a sistemelor de tip *firewall*, contribuind la dezvoltarea cunoștințelor în domeniu și direcțiilor pentru viitoare cercetări. Metodologia adoptată servește drept instrument esențial în validarea rezultatelor cercetării și în evidențierea contribuțiilor semnificative aduse în domeniul securității informaționale.

# Capitolul 2: Studiu privind stadiul actual al securității informaționale

În prezent, majoritatea soluțiilor de securitate cibernetică se bazează pe instrumente de detecție create de experți umani. Cu toate acestea, această abordare întâmpină dificultăți în a rămâne la curent cu amenințările cibernetice tot mai avansate și noi. Utilizarea noilor tehnologii și dispozitive [3] devine indispensabilă pentru a obține rezultate satisfăcătoare în domeniul securității informaționale. Inteligența artificială joacă un rol important în accelerarea procesului de identificare a noilor amenințări [4] și în furnizarea de răspunsuri eficiente, permițând blocarea atacurilor înainte ca acestea să se răspândească pe scară largă.

Mai mult, adoptarea rapidă a calculului în *cloud*, dispozitivelor *Internet of Things (IoT)* [5] și tehnologiilor de inteligență artificială (*Artificial intelligence - AI)* [6] a adus noi provocări în domeniul securității. Cu toate acestea, infractorii cibernetici au început, de asemenea, să utilizeze aceleași tehnici de *AI* pentru a-și îmbunătăți capacitatea de scanare a rețelelor, identificarea vulnerabilităților și dezvoltarea programelor *malware* greu de detectat. Un aspect important de menționat este că dezvoltarea rapidă a tehnologiei în domeniul securității informaționale aduce cu sine provocări legate de accesibilitatea și transparența informațiilor [7].

Analiza amenințărilor cibernetice, intitulată „*Wolf Security Threat Insights*", este un raport trimestrial elaborat și publicat de *HP*, furnizând o evaluare detaliată a peisajului actual al amenințărilor cibernetice. Acest document se bazează pe datele colectate de la clienții *HP* din întreaga lume și oferă o perspectivă asupra tendințelor emergente în ceea ce privește tactica, tehnicile și procedurile utilizate de infractorii cibernetici. Conform raportului publicat în ultimul trimestru al anului 2023 [8], *HP* a evidențiat că *e-mail*-urile au constituit principala sursă de amenințări cibernetice, reprezentând 75% din totalul vectorilor de atac identificați, urmate de descărcările din *browser*, care au însumat 13%, iar alte metode, precum utilizarea dispozitivelor *USB*, au reprezentat 12% din totalul amenințărilor identificate.

Conform Directoratului Naţional de Securitate Cibernetică (DNSC) [9], într-un atac cibernetic de tip *ransomware* care a avut loc în luna februarie 2023, 21 de spitale din România au fost afectate, inclusiv trei importante instituții medicale din București. Atacul a fost efectuat folosind aplicația *ransomware Backmydata*, un virus din categoria *Phobos*, care a criptat datele din serverele spitalelor ce folosesc platforma informatică *HIPOCRATE*. În urma atacului, serviciile medicale nu au putut fi înregistrate, ceea ce a condus la aglomerarea camerelor de gardă cu pacienți. Alte 79 de unități se află sub investigație, fiind suspecte în implicarea acestui atac. Acest incident subliniază vulnerabilitatea sistemelor de sănătate la amenințările cibernetice și necesitatea unei securități informatice mai robuste în sectorul medical.

În cursul anului 2023, într-o eră caracterizată de avansul rapid al tehnologiei, progresul amenințărilor cibernetice rămâne o temă de importanță fundamentală în sfera securității informaționale [10]. În această perspectivă, devine imperativ să evidențiem necesitatea manifestării unei conștientizări adecvate cu privire la categoriile predominante de atacuri cibernetice cu care se confruntă în prezent organizațiile. Unul dintre cele mai comune și insidioase tipuri de atacuri cibernetice este reprezentat de atacurile de tip *phishing* [11]. Acestea se bazează pe utilizarea metodelor de inginerie socială pentru a induce în eroare utilizatorii, cu scopul de a dezvălui informații personale sau confidențiale. Atacatorii folosesc adesea *e-mail*-uri sau mesaje

care imită în mod fals entități și organizații de încredere, cum ar fi bănci, platforme de comerț electronic sau furnizori de servicii *online*, cu scopul de a câștiga încrederea victimelor.

*Phishing*-ul este conceput în mod ingenios pentru a crea o aparență de legitimitate [12], astfel încât victimele să fie înșelate să furnizeze datele lor sensibile, cum ar fi parole, numere de card de credit, informații bancare sau detalii de identificare personală. *E-mail*-urile sau mesajele *phishing* pot conține adesea *link*-uri către *site*-uri *web* false, care sunt concepute în mod abil pentru a imita aspectul și funcționalitatea *site*-urilor originale. Un alt tip frecvent și extrem de periculos de atac cibernetic în prezent este reprezentat de atacurile de tip *ransomware* [13]. Aceste atacuri sunt concepute pentru a infecta sistemele informatice ale indivizilor sau organizațiilor și pentru a bloca accesul la fișierele și datele critice prin criptare. Atacatorii implicați în astfel de atacuri cer apoi o răscumpărare, de obicei în criptomonede, în schimbul eliberării și decriptării fișierelor.

Un alt aspect deosebit de preocupant în peisajul securității cibernetice îl reprezintă atacurile asupra rețelelor corporative [14]. Aceste atacuri vizează infrastructura de rețea a organizațiilor și pot avea consecințe grave, inclusiv accesul neautorizat la informații sensibile și lansarea altor tipuri de atacuri. Atacatorii se concentrează adesea pe identificarea și exploatarea vulnerabilităților din infrastructura de rețea și din *software*-ul utilizat de organizații.

În domeniul securității informației, sistemele de detecție joacă un rol important în identificarea și prevenirea amenințărilor cibernetice. În 2023, s-au dezvoltat și implementat o serie de sisteme de detecție avansate [15], care au abordat diverse aspecte ale securității informației.

Sistemele de detecție a intruziunilor (*IDS - Intrusion Detection Systems*) [16] reprezintă componente cheie în securitatea informației, având capacitatea de a detecta și preveni intruziunile în rețele și sisteme informatice. În 2023, aceste sisteme au evoluat semnificativ pentru a face față amenințărilor cibernetice tot mai sofisticate. Principala funcție a *IDS*-urilor constă în monitorizarea traficului de rețea în timp real [17]. Ele analizează fluxul de date care tranzitează rețeaua și caută modele, semnături sau comportamente suspecte care pot indica activități neautorizate.

Pentru a combate și proteja împotriva atacurilor cibernetice, sunt utilizate mai multe mecanisme și soluții esențiale. Printre acestea se numără *firewall*-ul [18], care acționează ca o barieră de protecție între rețeaua internă și internet, monitorizând și controlând traficul de rețea în funcție de reguli prestabilite.

Mecanismele de detecție și prevenire a intruziunilor analizează și monitorizează traficul de rețea pentru a detecta semnale de alarmă și comportamente suspecte care pot indica activități neautorizate sau atacuri în curs. Aceste sisteme [19] pot bloca sau diminua atacurile în timp real pentru a proteja infrastructura, fiind mecanisme de detecție a virușilor și a *malware*-ului ce utilizează baze de date actualizate cu semnături și modele de amenințări pentru a identifica și bloca programele maligne. Aceste soluții se bazează pe algoritmi avansați de analiză a comportamentului și euristică [20] pentru a detecta și izola amenințările cibernetice.

Sistemele de autentificare și autorizare asigură controlul accesului la resurse și informații sensibile. Acestea includ metode de autentificare multi-factor, parole robuste și politici de gestionare a accesului. Criptarea datelor [21] este un alt mecanism important, care utilizează algoritmi matematici pentru a transforma datele într-o formă eligibilă în timpul tranzitului sau stocării, asigurând confidențialitatea și integritatea informațiilor.

# Capitolul 3: Procesul de virtualizare

Tehnologia de virtualizare [22] este larg utilizată în majoritatea mediilor tehnice datorită utilității sale în îndeplinirea mai multor sarcini și datorită potențialului său de a permite o utilizare mai eficientă a resurselor alocate. Termenul de virtualizare se referă la crearea, prin intermediul *software*-ului, a unei versiuni virtuale a unui mediu sau a unei resurse fizice. Virtualizarea este realizată prin intermediul unui Monitor al Mașinii Virtuale (*VMM - Virtual Machine Monitor*) [23], care generează un strat de abstractizare între componentele *hardware* și sistemul de operare al mașinii virtuale, împărțind și gestionând resursele *hardware* pentru un număr dorit de mașini virtuale. Fiecare mașină virtuală poate fi utilizată pentru îndeplinirea diverselor sarcini și poate rula diverse sisteme de operare [24]. Mașinile virtuale pot fi fie statice, când resursele alocate fiecărei mașini sunt identice, fie variabile, când resursele sunt alocate în mod flexibil în funcție de nevoi [25].

*Software*-ul de virtualizare permite folosirea și împărțirea eficientă a *hardware*-ului computerului fizic în mai multe medii virtuale, astfel încât să se poată obține randament maxim utilizând resursele alocate în mod dinamic [26]. Această tehnologie este strâns legată de interfața de programare a aplicației (*API*-uri) [27], execuția directă și traducerea adresei. Virtualizarea *API* reprezintă un proces de utilizare a unui instrument care creează o copie virtuală a unui *API*, reflectând toate specificațiile *API*-ului de producție și folosește această copie virtuală în locul variantei originale pentru testare [28].

Pentru a implementa cu succes o soluție de virtualizare, este nevoie de o analiză a infrastructurii existente, astfel încât să se poată evalua în ce măsură aceasta răspunde nevoilor și obiectivelor organizației. Deși platforma *vSphere* de la *VMware* [29] este cea mai cunoscută și utilizată în domeniul virtualizării, nu este singura opțiune disponibilă. Există și alte platforme viabile pentru virtualizare, cum ar fi *XenSource* și *Hyper-V* [30], care sunt produse stabile, concepute cu scopul de a îndeplini cerințele organizațiilor de dimensiuni medii și mari. Este esențial să se țină cont de factori precum performanța, scalabilitatea, securitatea și suportul tehnic în evaluarea platformelor disponibile [31].

Instrumentele de monitorizare a mașinilor virtuale [32] reprezintă o necesitate în contextul în care multe companii depind de disponibilitatea serverelor virtuale. Aceste instrumente de monitorizare furnizează informații valoroase pentru a supraveghea atât serverele mașinilor virtuale, cât și mașinile virtuale găzduite pe acestea, asigurându-se că cele din urmă funcționează în mod optim în orice moment, procesul fiind ilustrat în Figura 1. Agenții de monitorizare sunt instrumente care pot detecta și monitoriza aspecte precum starea sistemului, performanța și utilizarea resurselor.

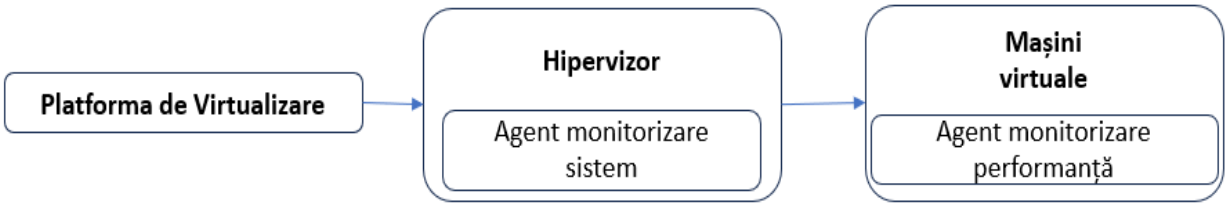



*Figura 1. Relaționare instrument monitorizare*

# Capitolul 4: Conceptul propus – firewall distribuit

Cadrul conceptual propus introduce o schimbare de paradigmă în securitatea rețelelor contemporane prin instanțierea unui sistem de *firewall* distribuit. Folosind principiile tehnologiei *open source*, această arhitectură încearcă să depășească limitele convenționale ale arhitecturilor de securitate tradiționale. Fundamentul acestei conceptualizări constă în implementarea strategică a instanțelor virtualizate, utilizând în mod specific *XCP-NG* ca platformă de virtualizare fundamentală. În acest cadru sunt încorporate instanțele *pfSense* configurate meticulos, poziționate strategic pentru a orchestra traficul de intrare și traficul intern cu o precizie care se aliniază cu complexitatea inerentă a infrastructurilor de rețea moderne.

În plus, integrarea *Snort* și *Suricata* ca *IDS* contribuie la un nivel vigilent de detectare a amenințărilor, consolidând și mai mult poziția de securitate. Acest cadru conceptual reprezintă o soluție de avangardă, pregătită să abordeze provocările contemporane reprezentate de amenințările cibernetice în continuă evoluție, cu virtuțile inerente ale modularității, scalabilității și viabilității economice intrinseci paradigmelor *open source*. În continuare se dezvăluie complexitatea acestui sistem *firewall*, prezentând nuanțele arhitecturale și justificând eficacitatea prin evaluări empirice și abordări teoretice.

În cadrul unui proiect, fie că este vorba de dezvoltarea unui produs *software* sau de implementarea unei soluții de securitate, managerii de proiect pot utiliza diverse metodologii pentru a ghida echipa și a monitoriza progresul acesteia. Una dintre metodele frecvent utilizate în acest scop este *V-MODEL* [33], prezentat în Figura 2, care urmărește îmbunătățirea calității generale a produsului final. *V-MODEL* reprezintă o reprezentare grafică a duratei de viață al dezvoltării *software* și este utilizat cu scopul de a executa și testa diferite procese într-o secvență logică.

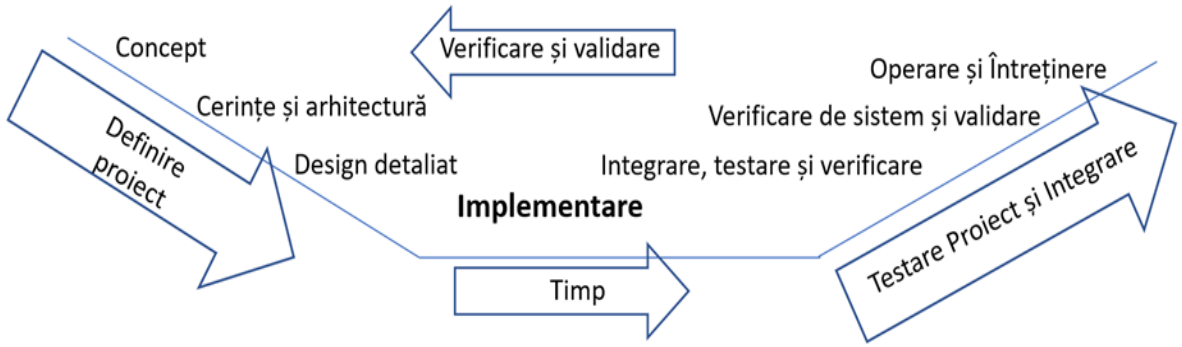



*Figura 2. Etapele V-MODEL.*

Un *firewall* distribuit plasează regulile și politicile de securitate mai aproape de aplicațiile individuale, oferind astfel o protecție mai eficientă [34]. Acest model de servicii distribuite , ilustrat în Figura 3, prezintă numeroase beneficii, printre care se numără îmbunătățirea performanței și scalabilității, reducerea costurilor asociate cu dispozitivele dedicate, implementarea microsegmentării și creșterea securității. Serviciile distribuite pot contribui la îmbunătățirea atât a performanței, cât și a scalabilității. Într-un model centralizat, pachetele sau fluxurile de date pot fi

nevoite să traverseze rețeaua centrului de date de mai multe ori pentru a trece prin dispozitivele dedicate, ceea ce poate introduce întârzieri și latență. În schimb, un model de servicii distribuite permite procesul de microsegmentare, care permite implementarea de politici detaliate de rețea și securitate.

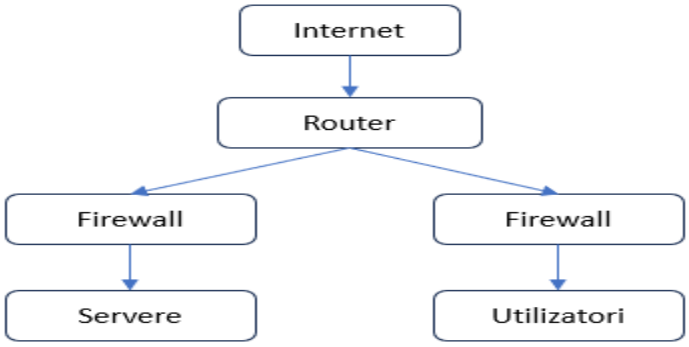



*Figura 3. Distribuirea sarcinilor în rețea*

Pentru a optimiza performanța rețelelor și a îmbunătăți utilizarea, este necesară o înțelegere detaliată a comportamentului de utilizare, semnalizării și caracteristicilor traficului în rețea. În acest scop, există o gamă variată de instrumente de testare disponibile, care permit simularea condițiilor de trafic în timp real pe o rețea.

Un simulator de trafic de rețea [35] are capacitatea de a genera trafic la nivelul rețelei, având control asupra protocoalelor și modelelor de generare a încărcăturii. De asemenea, poate genera trafic pe dispozitive specifice, independent de rețeaua de bază. Ambele metode permit utilizarea diverselor surse de trafic, o diagramă a modului de lucru este regăsită în Figura 4. Pe măsură ce intensitatea traficului crește, elementele de rețea pot prezenta diverse deficiențe, cum ar fi erori, întârzieri excesive, congestie, blocare, pierderi și degradare a calității. Simularea traficului poate oferi informații valoroase pentru a caracteriza degradarea în funcție de intensitatea și tipurile de trafic (de exemplu, voce, fax, date și video).

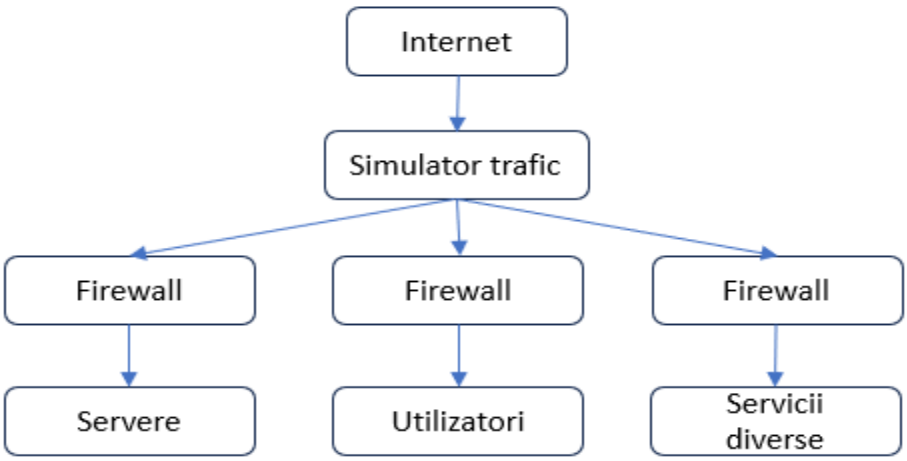



*Figura 4. Modul de funcționare al simulatoarelor de trafic*

Etapa de configurare a soluției propuse este evidențiată în Figura 5, unde este ilustrată arhitectura de sistem cu scopul de a evalua în mod exhaustiv performanța, precum și pentru a identifica avantajele și dezavantajele asociate. Această etapă importantă a reprezentat o oportunitate esențială de a ajusta și optimiza soluția în funcție de cerințele specifice și de a obține rezultatele dorite. Procesul de testare riguroasă nu doar a validat performanța, ci și a furnizat date pentru luarea deciziilor informate și îmbunătățirea continuă a soluției.

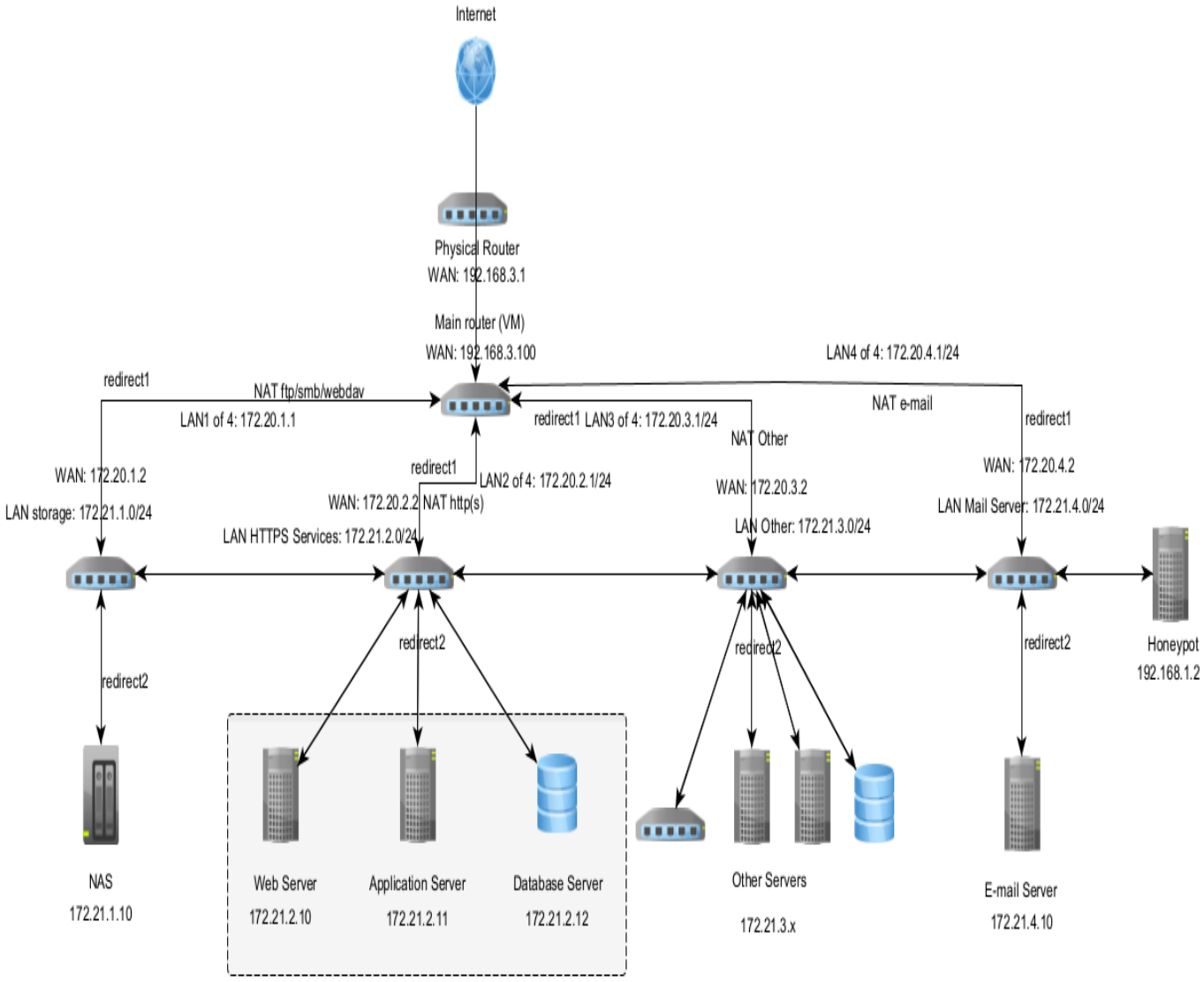



*Figura 5. Arhitectura propusă - Firewall Distribuit.*

Figura 5 ilustrează implementarea propusă, care constă într-un *firewall* distribuit virtualizat, configurat și operând conform cu parametrii normali, oferind rezultate conforme cu obiectivele propuse, precum obținerea de eficiență, scalabilitate și adaptabilitate [36]. Montajul prezentat reprezintă o soluție de securitate avansată, bazată pe un *firewall* distribuit și virtualizat cu ajutorul instrumentelor *open source*. Acesta a fost proiectat și configurat pentru a funcționa într-un mod eficient și pentru a furniza rezultate conforme cu așteptările și obiectivele stabilite anterior.

| NIC | VLAN | Auto | Link Status | MAC | MTU | SR-IOV |
|---|---|---|---|---|---|---|
| NIC 4 | - | Yes | Disconnected | 90:e2:ba:31:f6:d5 | 1500 | No |
| NIC 0 | - | Yes | Connected | 30:9c:23:64:ea:c5 | 1500 | No |
| NIC 1 | - | Yes | Disconnected | 90:e2:ba:31:f6:d0 | 1500 | No |
| NIC 2 | - | Yes | Disconnected | 90:e2:ba:31:f6:d1 | 1500 | No |
| NIC 3 | - | Yes | Disconnected | 90:e2:ba:31:f6:d4 | 1500 | No |
| NIC 0 | 10 | No | Connected | - | 1500 | No |
| NIC 0 | 20 | No | Connected | - | 1500 | No |
| NIC 0 | 30 | No | Connected | - | 1500 | No |
| NIC 0 | 40 | No | Connected | - | 1500 | No |
| NIC 0 | 42 | No | Connected | - | 1500 | No |
| NIC 0 | 100 | No | Connected | - | 1500 | No |
| NIC 0 | 200 | No | Connected | - | 1500 | No |
| NIC 0 | 300 | No | Connected | - | 1500 | No |
| NIC 0 | 400 | No | Connected | - | 1500 | No |

*Figura 6. Listă controlere de interfață de rețea.*

Conexiunile între *firewall*-uri, *server*e și clienți au fost stabilite prin intermediul plăcilor de rețea virtualizate și a *VLAN*-urilor (*Virtual Local Area Networks*) configurate la nivelul platformei de virtualizare. O listă detaliată a acestor conexiuni este prezentată în Figura 6 și Figura 7. Prima ilustrare prezintă plăcile de rețea virtualizate care au fost utilizate pentru a facilita comunicația între diferitele componente ale sistemului. Aceste plăci de rețea virtualizate oferă un mediu virtual securizat și izolat, asigurând transferul eficient și securizat al datelor între *firewall*-uri, servere și clienți. Prin intermediul acestor plăci de rețea virtualizate, se realizează o conexiune stabilă și rapidă între entitățile implicate, contribuind la performanța și securitatea sistemului. Figura 7 prezintă configurarea *VLAN*-urilor la nivelul platformei de virtualizare. *VLAN*-urile sunt utilizate pentru a segmenta și separa traficul de rețea în funcție de nevoile și cerințele specifice.

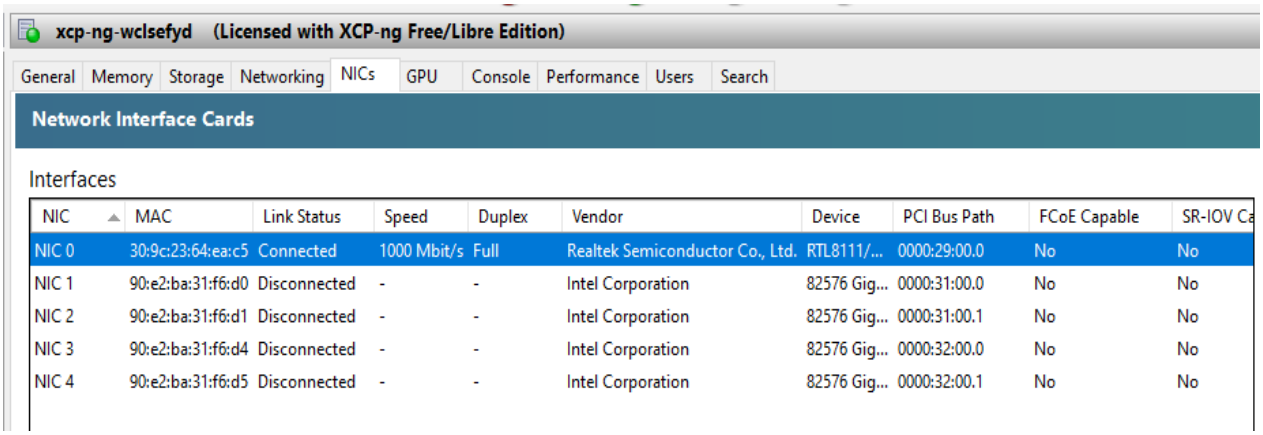



*Figura 7. Lista rețelelor virtuale principale.*

În cadrul montajului propus, configurarea a fost realizată pe un singur *SSD*, așa cum se poate observa Figura 8. Acest dispozitiv de stocare are o capacitate de 1 *TB* și a fost utilizat pentru a asigura spațiul necesar funcționării fiecărei mașini virtuale. Fiecare mașină virtuală a primit un spațiu de stocare personalizat, adaptat cerințelor și nevoilor specifice. Pe durata desfășurării testelor, s-a constatat că mașinile virtuale au avut nevoie de spațiu de stocare suplimentar. Acest aspect a fost determinat de procesele de actualizare care au avut loc pe parcurs și de *log*-urile în care sunt înregistrate evenimentele relevante.

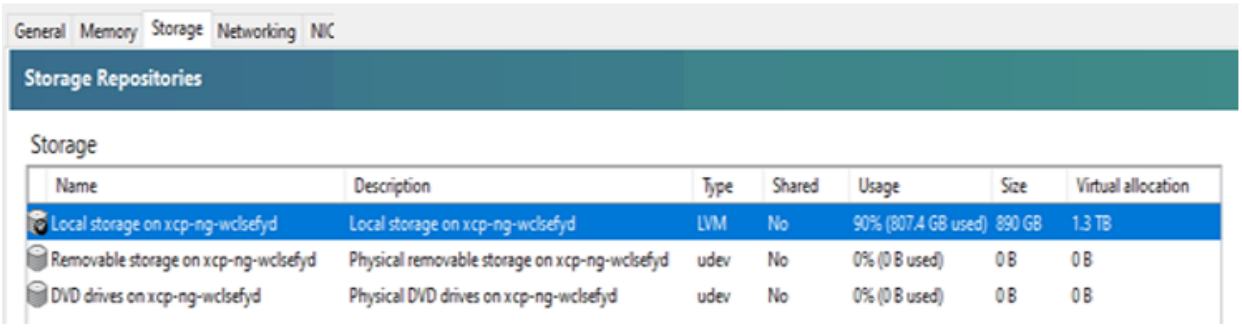



*Figura 8. Stocare montaj experimental.*

Platforma *AMD* furnizează un ansamblu de facilități *hardware* proiectate cu scopul de a susține arhitectura procesorului și de a optimiza eficiența resurselor, contribuind la sporirea performanței mașinilor virtuale. Această tehnologie, denumită *AMD-V*, reprezintă o inovație în domeniul virtualizării, dezvoltată de *AMD*, care facilitează utilizarea simultană a resurselor *hardware*. Beneficiile tehnologiei *AMD-V* [37] includ capacitatea sa de a gestiona eficient sarcini repetitive și de a optimiza exploatarea resurselor disponibile. Figura 9 detaliază aspectele tehnice ale plăcii de bază utilizată pentru a beneficia de tehnologia *AMD-V*.

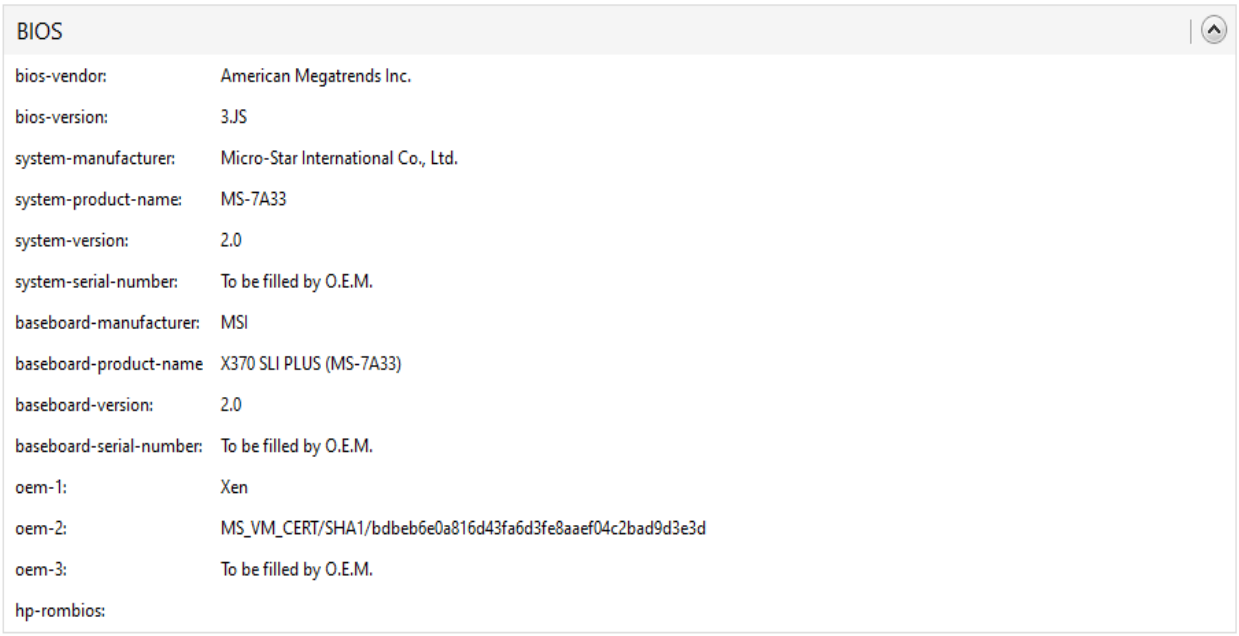



*Figura 9. BIOS folosit pentru virtualizare.*

Figura 10 prezintă interfața principală a montajului propus, oferind o platformă centrală pentru gestionarea diferitelor aspecte ale sistemului. Această interfață permite administrarea interfețelor virtualizate, a jurnalelor în care sunt stocate evenimentele din rețea și a *gateway*-urilor utilizate pentru comunicarea între *firewall*-uri. Un aspect important al acestei interfețe este flexibilitatea sa, permițând configurarea afișării informațiilor relevante în funcție de nevoile specifice. În secțiunea "*Interfaces*", marcată într-un chenar, sunt prezentate cele patru interfețe principale create cu scopul de a separa *firewall*-urile secundare și, implicit, dispozitivele conectate la acestea. Aceste interfețe au fost configurate manual într-un mod experimental, urmărind o

structură adaptată cerințelor sistemului. Prin intermediul acestor interfețe, se realizează gestionarea fluxului de date și comunicarea între diversele componente ale rețelei. Secțiunea "*Gateway*s" din chenarul corespunzător prezintă informații referitoare la starea *gateway*-urilor utilizate pentru comunicarea dintre *firewall*-urile secundare și *firewall*-ul principal. Aici, utilizatorul poate verifica existența oricăror probleme asociate acestor *gateway*-uri și poate efectua acțiuni corective în consecință. Este important să se monitorizeze aceste *gateway*-uri pentru a asigura o comunicare eficientă între componente și pentru a identifica eventualele deficiențe în timp util.

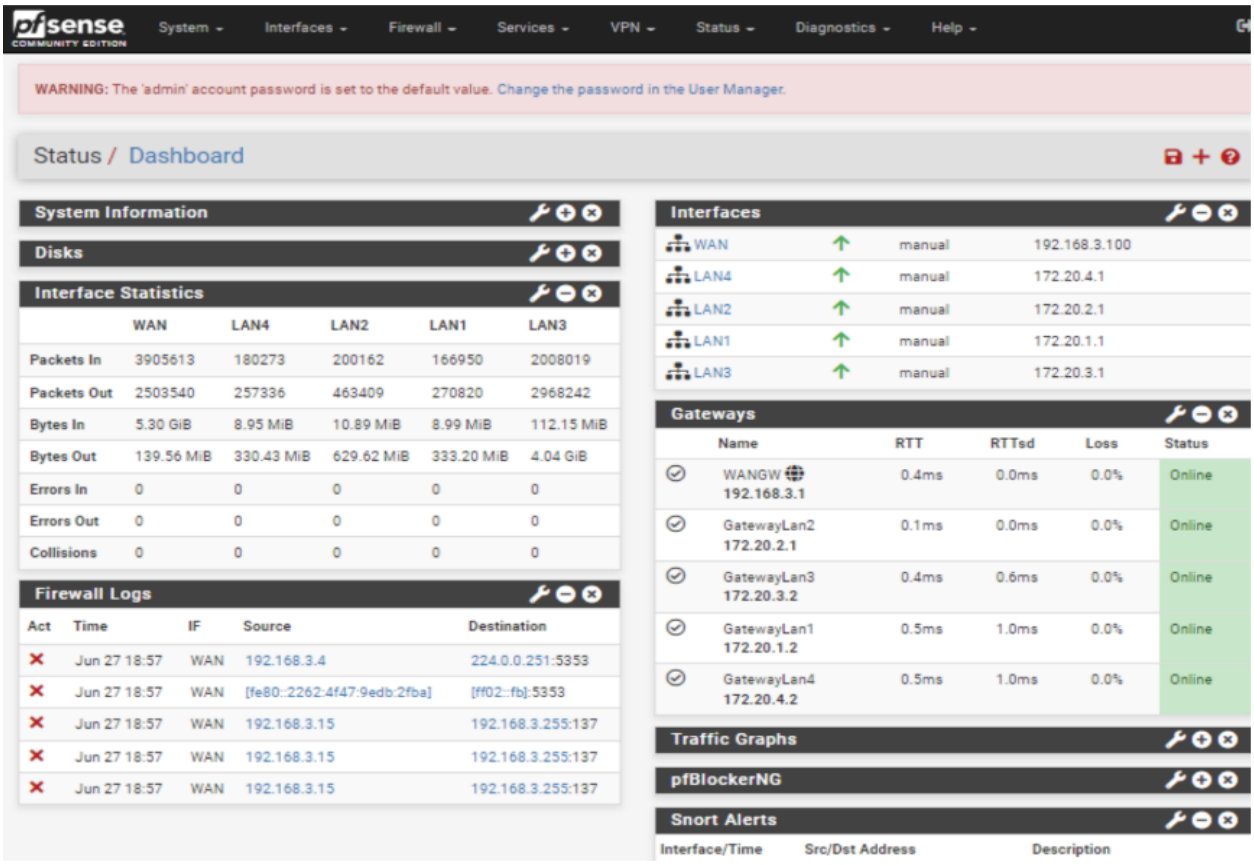



*Figura 10. Meniu principal GUI PfSense.*

Eficiența soluției propuse poate fi observată în figurile care urmează, unde se prezintă rezultatele monitorizării traficului obișnuit din rețeaua internă. Aceste date reflectă acțiunile întreprinse de *firewall* în urma analizei celor mai recente 10.000 de pachete care au intrat în rețea. Recepția traficului de Internet pe un *router*, chiar și atunci când nu este implicat în mod activ în activități de navigare, poate fi atribuită mai multor factori de bază în contextul mai larg al comunicării în rețea. Acest trafic poate fi cauzat de:

- Procese sau servicii de fundal, unde diferite dispozitive și servicii intercomunică.
- Protocoale precum *ARP* și *ICMP*, care contribuie la trafic prin descoperire, întreținere și raportarea erorilor rețelei.
- Scanări și diferite sondări, care sunt în căutare de dispozitive vulnerabile.
- Zgomot de rețea, fiind trafic aleatoriu în căutare și descoperire.
- Actualizări de dispozitive și produse *software*, care sunt periodice.

Soluția implementată a evaluat aceste pachete și a identificat că majoritatea dintre ele reprezintă trafic suspicios. Pentru a gestiona această situație, s-au realizat o serie de reprezentări grafice, prin Figurile 22-28, care ilustrează procesul de analiză efectuat de soluția propusă și rezultatele obținute în urma acestuia. În aceste grafice, sunt prezentate datele colectate și interpretate de soluție, oferind o perspectivă vizuală asupra modului în care au fost tratate pachetele în rețea.

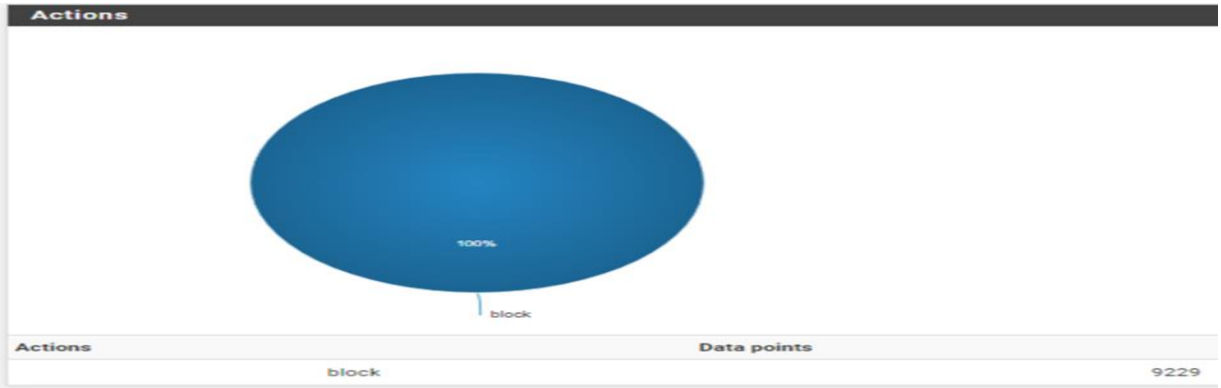



*Figura 11. Procentul de trafic suspicios blocat de firewall.*

*Firewall*-ul *pfSense* dispune de numeroase opțiuni de afișare a datelor filtrate. Figura 11 ilustrează o diagramă circulară ce vizualizează un set de date specific aplicației *pfSense*, sub panoul denumit *Actions*.

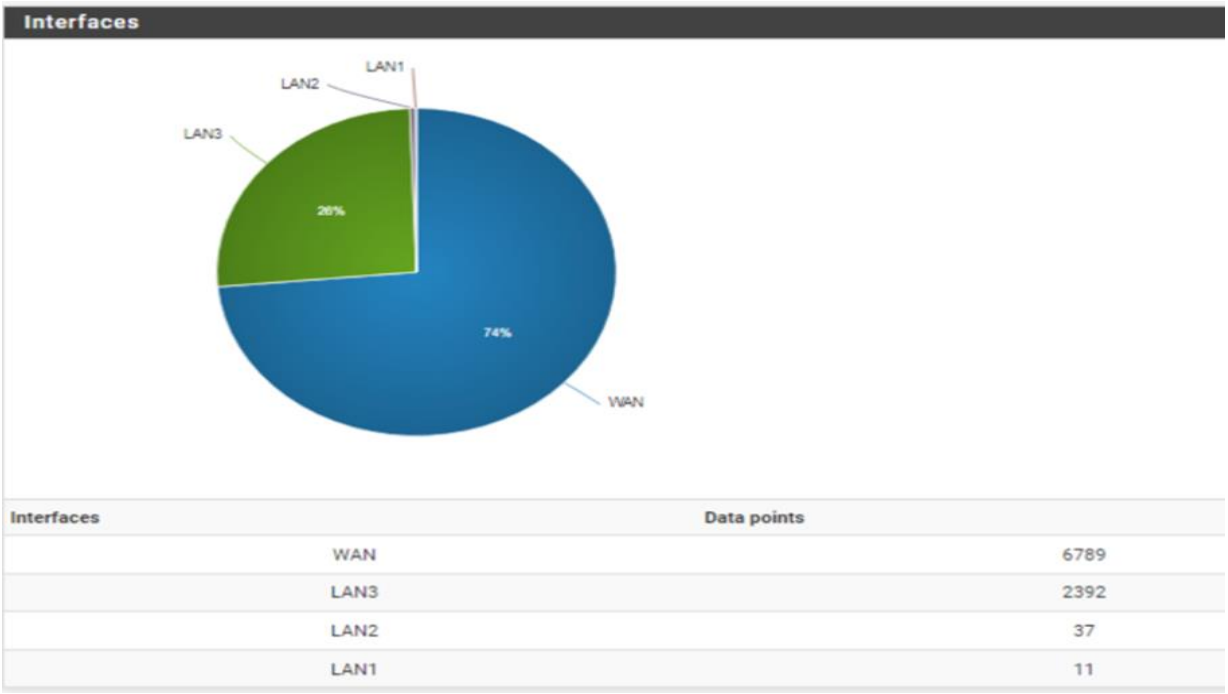
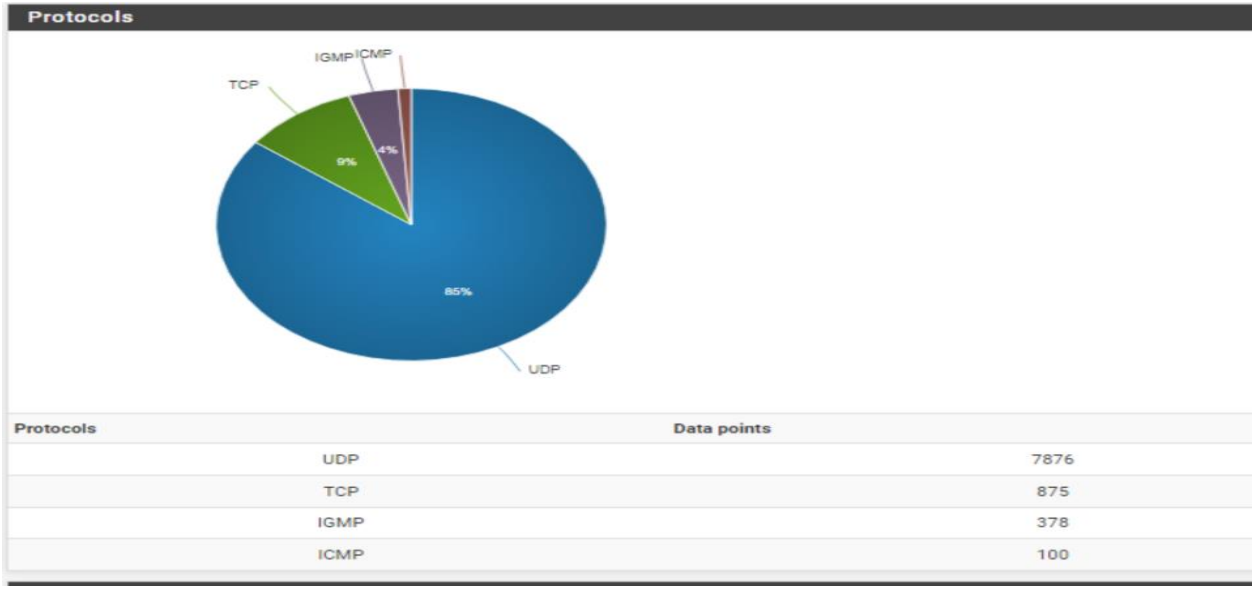



*Figura 12. Procentul traficului total pe interfețe.*

Analiza continuă cu inspectarea traficului pe interfețele utilizate în cadrul experimentului. Imaginea prezentată în Figura 12 reprezintă o captură de ecran care ilustrează o vizualizare a datelor, compusă dintr-o diagramă circulară și un tabel. Diagrama afișată descrie distribuția

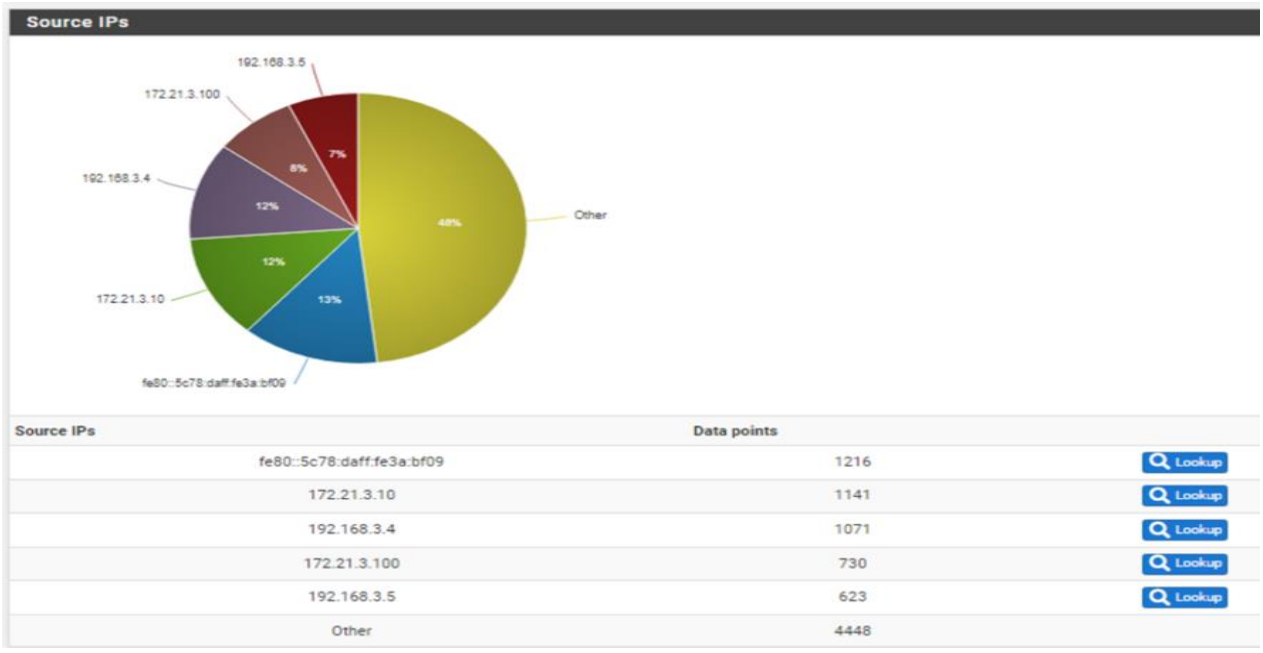
punctelor de date prin diverse interfețe, folosind culorile albastru și verde pentru a diferenția interfețele respective.




| Protocols | Data points |
|---|---|
| UDP | 7876 |
| TCP | 875 |
| IGMP | 378 |
| ICMP | 100 |

*Figura 13. Protocoalele de trafic depistate.*

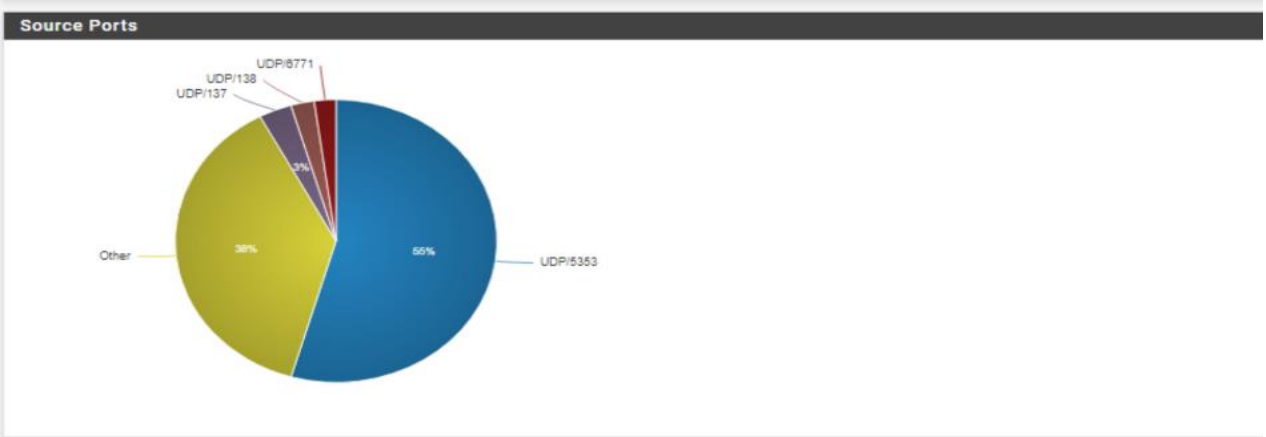
Protocoalele utilizate sunt importante pentru a determina tipul de trafic care este analizat. Figura 13 reprezintă o captură de ecran ilustrând o diagramă care vizualizează distribuția datelor pe protocoale. Diagrama arată repartizarea punctelor de date între trei protocoale: *UDP*, *TCP* și *ICMP*. Tabelul detaliază fiecare protocol și numărul asociat de pachete de date: 7876 pentru *UDP*, 875 pentru *TCP* și 100 pentru *ICMP*. Figura ilustrează eficient distribuția punctelor de date pe diverse protocoale în cadrul experimentului propus .




| Source IPs | Data points | |
|---|---|---|
| fe80::5c78:daff:fe3a:bf09 | 1216 | Lookup |
| 172.21.3.10 | 1141 | Lookup |
| 192.168.3.4 | 1071 | Lookup |
| 172.21.3.100 | 730 | Lookup |
| 192.168.3.5 | 623 | Lookup |
| Other | 4448 | |

*Figura 14. Adresele IP sursă.*

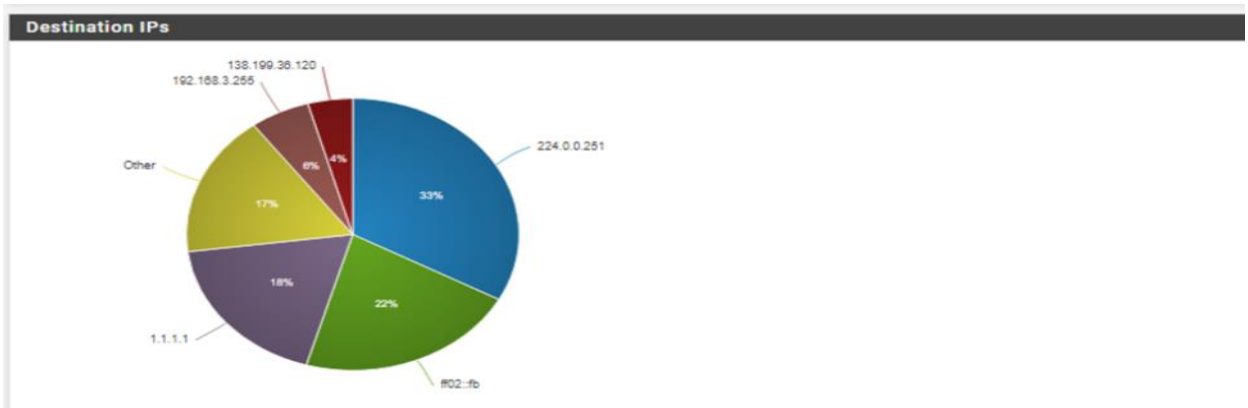
Sursele traficului prezintă interes deoarece putem identifica pe baza acestora dacă traficul este de încredere. Imaginea prezentată în Figura 14 ilustrează o vizualizare a datelor sub forma unei diagrame și a unui tabel.




| Source Ports | Data points |
|---|---|
| UDP/5353: mdns | 5031 |
| UDP/137: netbios-ns | 305 |
| UDP/138: netbios-dgm | 217 |
| UDP/6771 | 200 |
| Other | 3476 |

*Figura 15. Porturile sursă.*

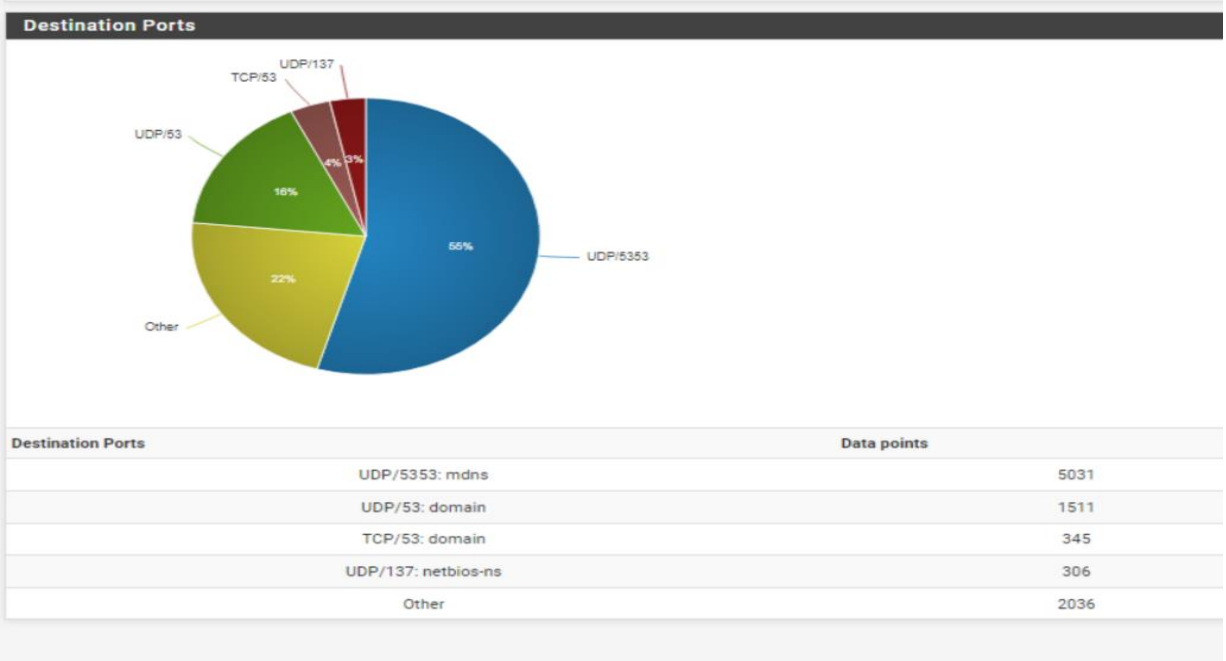
O schimbare bruscă sau o activitate neobișnuită pe anumite porturi sursă poate indica prezența unei amenințări sau a unui comportament suspect. Această monitorizare este esențială pentru detectarea și gestionarea potențialelor atacuri. Figura 15 reprezintă o captură de ecran care ilustrează o diagramă ce corespunde unui panou extras din pfSense.




| Destination IPs | Data points | |
|---|---|---|
| 224.0.0.251 | 3040 | Lookup |
| ff02::fb | 1991 | Lookup |
| 1.1.1.1 | 1704 | Lookup |
| 192.168.3.255 | 535 | Lookup |
| 138.199.36.120 | 401 | Lookup |
| Other | 1558 | |

*Figura 16. Adresele IP destinație.*

Prin analiza adreselor *IP* destinație, se poate identifica traficul provenind de la surse nedorite sau cunoscute pentru activități distructive. Această informație este esențială pentru prevenirea și gestionarea amenințărilor la adresa rețelei. În cazul unui incident de securitate, informațiile privind adresele *IP* destinație permit un răspuns rapid și eficient. Identificarea și izolarea rapidă a surselor potențiale ale amenințărilor contribuie la minimizarea impactului asupra rețelei. Imaginea prezentată în Figura 16 este o reprezentare grafică obținută din interfața *pfSense*, ce conține adresele *IP* destinație din cadrul experimentului propus.




| Destination Ports | Data points |
|---|---|
| UDP/5353: mdns | 5031 |
| UDP/53: domain | 1511 |
| TCP/53: domain | 345 |
| UDP/137: netbios-ns | 306 |
| Other | 2036 |

*Figura 17. Porturi destinație.*

Porturile destinație indică serviciile sau aplicațiile la care încearcă să acceseze traficul. Această informație este esențială pentru a identifica și gestiona corect tipul de activitate care intră în rețea. Prin gestionarea porturilor destinație, organizațiile pot controla accesul la anumite resurse sau servicii. Aceasta include restricționarea accesului la porturile destinație relevante pentru a limita expunerea la posibile amenințări. Unii atacatori încearcă să exploateze vulnerabilități specifice ale unor servicii sau aplicații prin porturile destinație. Monitorizarea și gestionarea acestor porturi pot contribui la prevenirea unor astfel de atacuri. Porturile destinație sunt esențiale pentru implementarea politicii de securitate a rețelei.

Stabilirea unor reguli specifice pentru porturile destinație permite definirea unui set coerent de reguli și restricții pentru protecția activelor și datelor organizației. Imaginea furnizată în Figura 17 prezintă o captură de ecran a unei reprezentări grafice sub formă de diagramă, și evidențiază distribuția pachetelor de date între diferite porturi. Segmentul majoritar al diagramei, colorat în albastru, reprezintă 50% din punctele de date și corespunde portului *UDP*/5353. Al doilea segment ca mărime, colorat în roșu, corespunde portului *UDP*/137 cu 27% din date. Segmentul verde, reprezentând *TCP*/57, constituie 12% din punctele de date, în timp ce segmentul galben, corespunzător portului *UDP*/6453, ocupă 7%. Segmentul etichetat ca "*Other*" este reprezentat în gri și cuprinde restul de 4% din porturile de date.

# Capitolul 5: Soluții propuse și implementate

În vederea abordării și soluționării unor provocări specifice întâmpinate în arhitecturile *firewall* prezentate anterior, s-a încercat aducerea unei contribuții prin dezvoltarea unei soluții progresive [38], ilustrată în Figura 5. Această propunere a fost prezentată și discutată în cadrul conferinței internaționale *Development and Application Systems*, reprezentând un eveniment pentru diseminarea rezultatelor obținute din cercetare. Arhitectura care se propune este concepută pe baza fundamentelor de *firewall* distribuit discutate anterior. *Firewall*-ul distribuit este alcătuit dintr-un *router* principal și patru *router*e secundare, fiecare dintre acestea fiind configurat cu un *firewall* adaptat situației specifice.

Prin această nouă abordare, s-a propus o soluție eficientă pentru gestionarea și protejarea rețelelor în fața amenințărilor persistente din mediul digital. În Figura 18 se prezintă o altă reprezentare a conceptului propus, având în vedere simplificarea fluxului de date în cadrul rețelei. Pentru a oferi o perspectivă accesibilă și coerentă asupra întregii infrastructuri de rețea, se preferă utilizarea platformei *Zabbix*, o soluție *open source* consacrată monitorizării rețelelor, serverelor, serviciilor *cloud* și mașinilor virtuale [39]. Pentru a colecta informații operaționale relevante, precum statistici despre procesor, stocare, lățime de bandă și memorie, se configurează *Zabbix Agent* pe dispozitivul țintă. Acesta poate genera alerte active în cazul unor defecțiuni ale dispozitivelor sau proceselor.

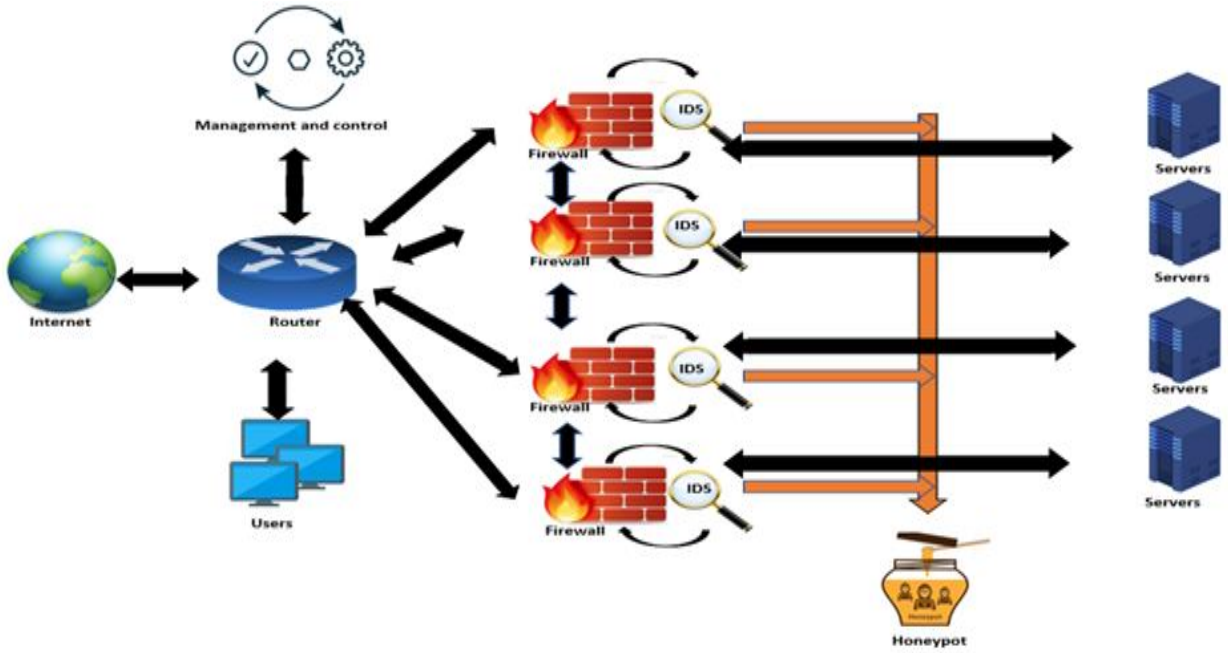



*Figura 18. Diagramă concept propus – Firewall Distribuit.*

Pentru o mai mare eficiență și simplificare, se consideră integrarea tiparelor de *firewall* care s-au dezvoltat în platforma *Zabbix* [40] pentru monitorizarea la distanță utilizând *Simple Network Management Protocol (SNMP). SNMP* este un protocol consacrat pentru citirea și actualizarea configurațiilor diferitelor dispozitive din rețea [41]. Configurarea *SNMP* în modul *Read-Write* permite setarea și manipularea valorilor în setările dispozitivului. Diagrama fluxului de date în

acest caz este ilustrată în Figura 19. Instrumentele de monitorizare propuse sunt avantajoase întrucât permit descoperirea automată și interogarea dispozitivelor pentru a extrage date relevante într-un mod accesibil.

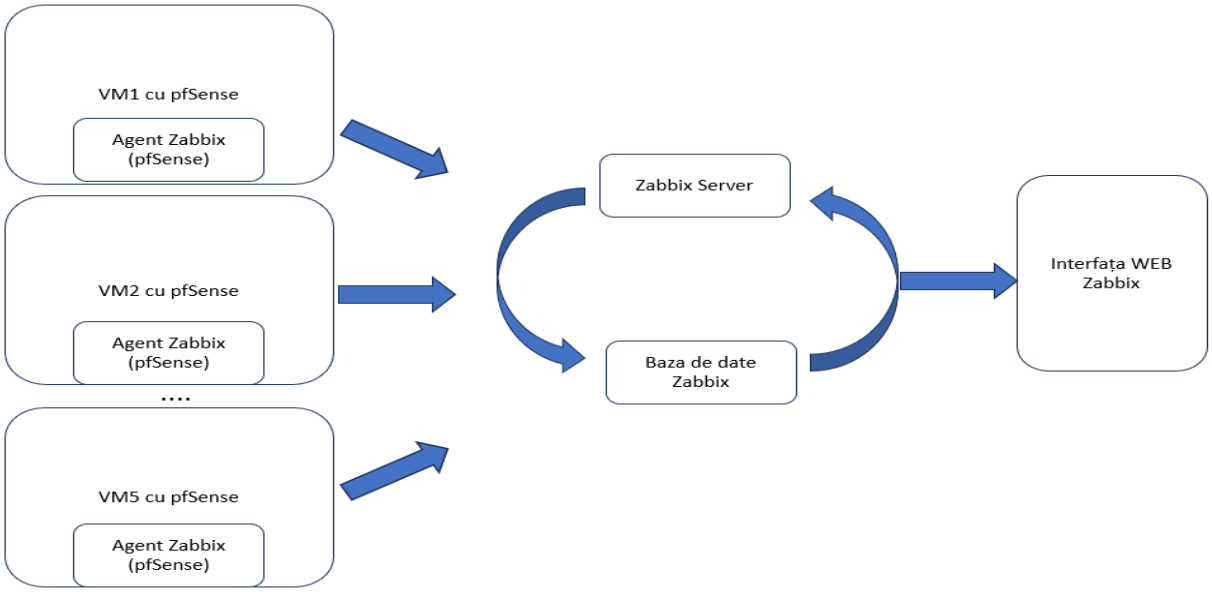



*Figura 19. Monitorizare Zabbix*

Diagrama prezentată în Figura 18 ilustrează monitorizarea traficului de rețea pentru soluția propusă. *Router*-ul principal este echipat cu un *firewall* standard, care are rolul de a respinge traficul incorect la intrarea în rețea. Astfel, doar traficul considerat normal este transmis către celelalte *firewall*-uri, care sunt configurate cu setări confidențiale din motive de securitate. *Firewall*-urile analizează traficul de intrare, însă abordările utilizate pot varia în funcție de scenariul specific al fiecărui *firewall*, astfel încât traficul să fie supus unei analize mai detaliate. În vederea împărțirii și redirecționării traficului de intrare către *firewall*-ul responsabil cu gestionarea serverului țintă, sunt necesare câteva configurații suplimentare.

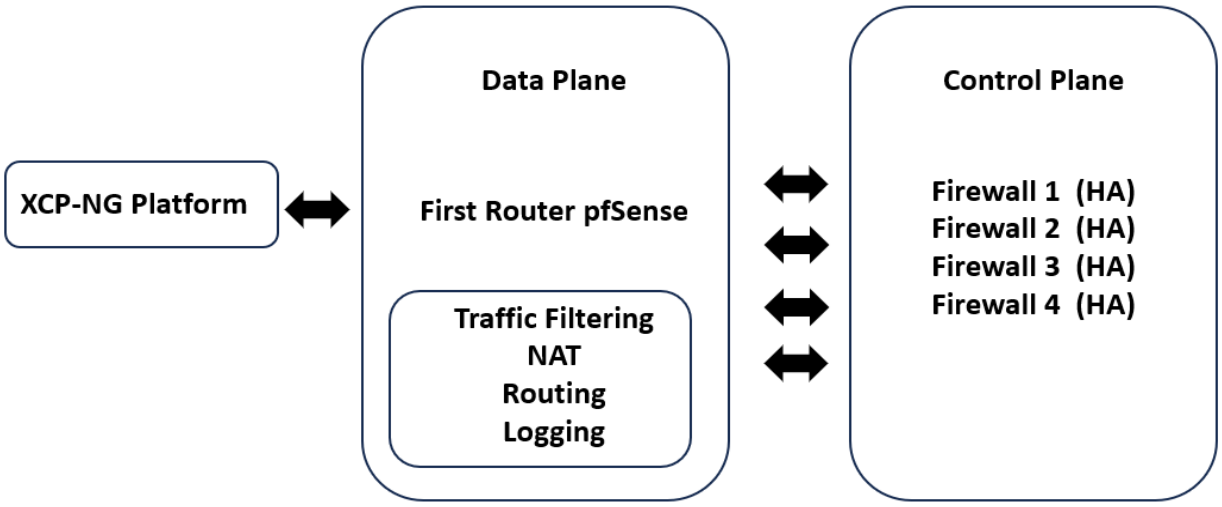



*Figura 20. Realizarea comunicării dintre planul de date și planul de control.*

În aces caz, planul de date este format din *router*-ul principal, iar planul de control include toate *firewall*-urile rămase, această propunere este ilustrată în Figura 20. *Firewall*-ul principal gestionează traficul de intrare și ieșire din rețea, aplică politicile de securitate, inclusiv filtrarea pachetelor, *NAT* și partea de rutare, dar și monitorizează și înregistrează activitățile din rețea. *Firewall*-urile secundare sunt în configurație de înaltă disponibilitate (*HA – High Availability*), ceea ce oferă redundanță și capacitate de comutare în caz de defecțiune.

În cadrul experimentului propus pentru transferul de date între rețele, pe lângă utilizarea unui *gateway*, se impune implementarea unui mecanism pentru dirijarea traficului, metoda abordată fiind cea reprezentată de definirea de rute statice. Aceste rute statice sunt configurate pe *router*-ul principal și sunt adăugate în tabelul de rutare. Integrarea unei rute statice în infrastructura rețelei aduce multiple avantaje, printre care reducerea utilizării lățimii de bandă între *router*e, evitarea suprasolicitării *router*-ului principal și, implicit, consum redus de resurse, precum și sporirea securității, întrucât numai un administrator autorizat le poate gestiona.

În cadrul transmiterii datelor prin *Internet*, există două protocoale de transport larg utilizate: *TCP* (*Transmission Control Protocol*), un protocol de transport fundamental și alternativa reprezentată de *UDP* (*User Datagram Protocol*) [42]. În cadrul cercetării propuse, s-a concentrat atenția în special asupra protocolului *UDP*, care implică un compromis între fiabilitate şi performanță. *UDP* este folosit în anumite situații în care *TCP* nu este avantajos, precum *VPN*-uri, difuzare, *streaming* media *live* și *DNS*. De asemenea, *UDP* este utilizat pentru transportul jurnalelor, datorită performanței vitezei de transfer, monitorizării datelor, moștenirii și transferului informațiilor de stare.

Una dintre soluțiile ușor de implementat pentru protecția împotriva atacului *UDP flood* este controlul accesului prin adăugarea unui strat suplimentar de autentificare utilizând diverse aplicații. Scopul principal al atacului *UDP flood* constă în faptul că serverul vizat trebuie să utilizeze resurse pentru a verifica și a răspunde la fiecare pachet *UDP* primit. Într-un interval scurt de timp, acest volum mare de pachete poate suprasolicita serverul și poate duce la o stare de refuz al serviciului de tip *DDoS* .

În continuare, s-a analizat traficul de rețea *UDP* generat de un simulator într-un atac de acest tip. S-a propus o soluție personalizată pentru contracararea acestui atac, constând în implementarea unei reguli de *firewall* care aduce îmbunătățiri semnificative în utilizarea resurselor rețelei. Studiul propus a fost prezentat în cadrul conferinței *RoEduNet 2022* [43] și se concentrează în mod specific pe atacul ce folosește simulatorul *LOIC* asupra unui server *FTP* într-o rețea care este gestionată de *PfSense* [44] ca *firewall* și *Snort* [45] ca *IDS*. În acest scenariu, atacatorul generează o mare cantitate de pachete care sunt trimise către un server specific din interiorul unei rețele, cu scopul de a supraîncărca atât ținta, cât și *firewall*-ul. Această supraîncărcare duce la disfuncționalitatea *firewall*-ului țintă, care nu mai poate opera în parametri normali. Studiul propus se concentrează pe abordarea acestei probleme a atacului *UDP* într-o rețea locală și utilizează distribuții *open source* pentru a gestiona congestia traficului de rețea.

De obicei, protocolul de transfer de fișiere folosește *TCP* ca protocol de comunicare între client și server. În cadrul testelor efectuate în cadrul montajului experimental, s-a descoperit că generarea de trafic *UDP* pe portul utilizat de *FTP* poate congestiona rețeaua și crea instabilitate. Figura 21 ilustrează mediul de lucru și modalitatea în care se intenționează testarea securității serverului *VSFTPD* în cadrul configurației propuse.

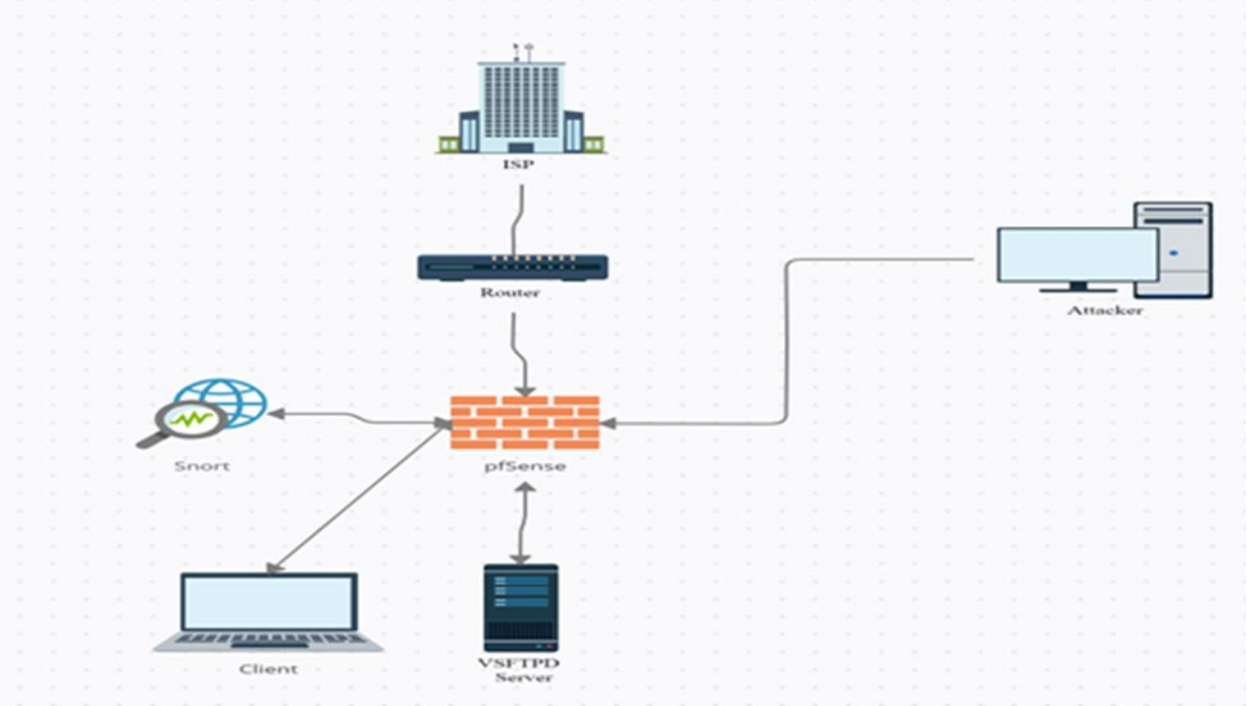



*Figura 21. Scenariu experimental atac UDP.*

*Snort* utilizează un limbaj bazat pe reguli flexibile pentru a genera alerte și pentru a bloca atacatorii pe baza adreselor *IP*. Pentru a evalua eficiența unei reguli personalizate *Snort*, a fost realizat un experiment folosind *PfSense* și *Snort* pentru a transfera un fișier de 2 *GB* între un server și un client. În primul scenariu, fără reguli personalizate, transferul a avut loc în mod normal. În al doilea scenariu, cu o regulă personalizată pentru detectarea și blocarea traficului de tip inundație *UDP*, s-a observat o îmbunătățire semnificativă a performanței rețelei, reducând impactul traficului rău intenționat și evidențiind utilitatea regulilor personalizate *Snort* în îmbunătățirea securității și eficienței rețelei.

*Tabel 1. Datele obținute în cele două scenarii*

| | | | Scenariu 1 | | | Scenariu 2 | | |
|---|---|---|---|---|---|---|---|---|
| **Sistem** | **Utilizarea resurselor** | | **Stare normală (cel mai mic - cel mai mare)** | **Începerea atacului (cel mai mic - cel mai mare)** | **În timpul atacului (Cel mai scăzut-Cel mai ridicat)** | **Stare normală (Cel mai scăzut-Cel mai ridicat)** | **Începerea atacului (cel mai mic - cel mai mare)** | **În timpul atacului (Cel mai scăzut-Cel mai ridicat)** |
| *PfSense* | *CPU* (%) | | 0.2 - 3 | 70 - 80 | 89 - 94 | 0.2 - 3 | 30 - 35 | 33 - 58 |
| | *Memorie RAM* (%) | | 10 - 11 | 10 - 12 | 10 - 12 | 10 - 11 | 10 - 16 | 10 - 16 |
| | Trafic de rețea (*Mbps*) | Recepționat | 0.2 - 0.3 | 22 - 23 | 23 - 26 | 0.2 - 0.3 | 13 - 23 | 20 – 22 |
| | | Trimis | 0.2 - 0.3 | 18 - 19 | 18 - 19 | 0.2 - 0.3 | 17 - 19 | 17 – 19 |
| *VSFTPD* | *CPU* (%) | | 0,1 - 2 | 13 - 55 | 30 - 72 | 1 - 2 | 10 - 24 | 0.2 – 4 |
| | *Memorie RAM* (%) | | 2 - 6 | 10 - 12 | 10 - 12 | 2 - 6 | 2 - 6 | 2 – 6 |
| | Trafic de rețea (*Mbps*) | Recepționat | 0.2 - 0.3 | 18 - 19 | 23 - 26 | 0.2 - 0.3 | 17 - 19 | 0.2 - 0.3 |
| | | Trimis | 0.2 - 0.3 | 18 - 19 | 23 - 26 | 0.2 - 0.3 | 17 - 19 | 0.2 - 0.3 |

Rezultatele obținute în urma acestor teste sunt prezentate în Tabel 1, unde se analizează parametrii *PfSense* și destinația, serverul *VSFTPD*. În Figura 22 sunt ilustrate diferențele între starea normală a rețelei și starea de atac, în două momente distincte: la începutul atacului și după stabilizarea acestuia, care are loc în câteva secunde. Odată ce *Snort* detectează și blochează adresa *IP* a atacatorului, serverul *VSFTPD* revine la parametrii săi normali de funcționare. Regula dezvoltată demonstrează îmbunătățiri în utilizarea resurselor, astfel încât traficul generat în cadrul atacului reușește să ajungă la serverul *VSFTPD*, dar nu se mai întoarce prin intermediul *PfSense* către sursa atacului.

Cu toate acestea, din cauza acestor dezavantaje, s-a observat o ușoară întârziere în timpul transferului de fișiere prin rețea, deoarece serverul *VSFTPD* continuă să primească trafic. În regula personalizată ce s-a implementat, s-a ales acțiunea "*drop*" pentru traficul de tip *UDP flood*, ceea ce înseamnă că toate pachetele detectate sunt blocate, iar *PfSense* înregistrează aceste pachete. Având în vedere că în configurația propusă se utilizează o rută statică și o regulă de *port forward* de la *router*-ul principal către serverul *VSFTPD*, s-a constatat că o parte din traficul de intrare este trimis în rețea, chiar dacă adresa *IP* sursă a fost blocată în *Snort*. Pentru a remedia această problemă, s-a configurat o rută statică de tip *NULL* în *PfSense*.

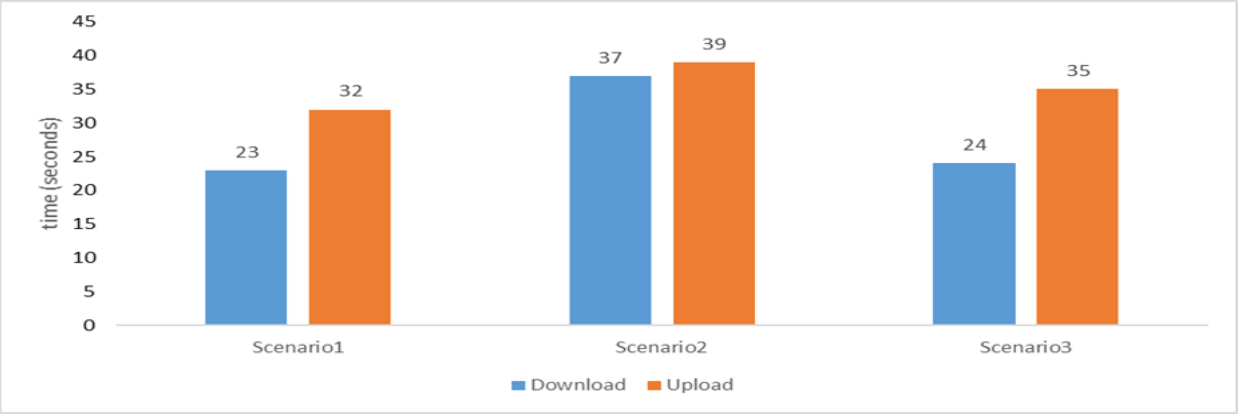



*Figura 22. Diferențele dintre durata transferului în diferite scenarii.*

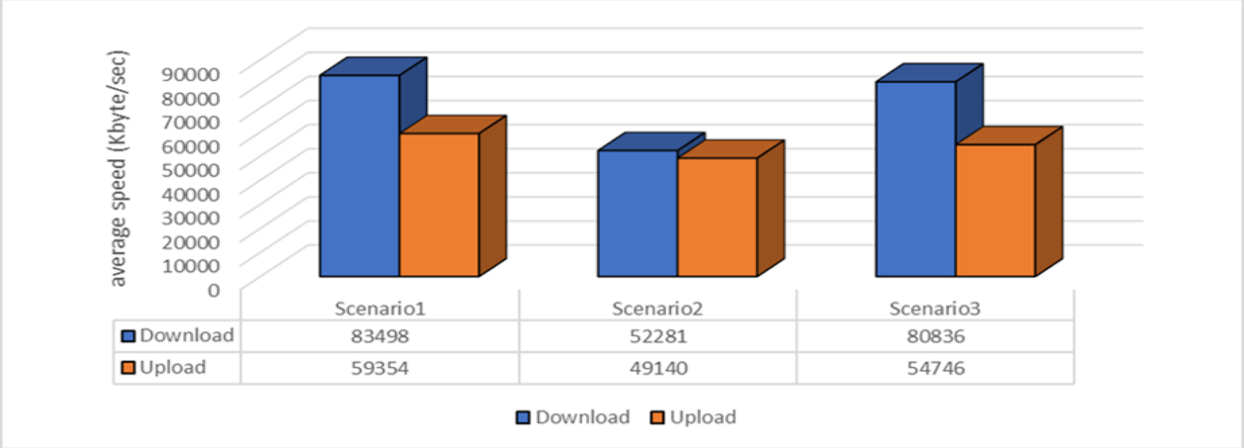



*Figura 23. Diferențele între viteza medie în diferite scenarii.*

În ceea ce privește transferul fișierelor între clientul *Windows 10* și serverul *VSFTPD* prin intermediul *PfSense*, s-au efectuat monitorizări asupra câtorva transferuri, iar rezultatele obținute sunt prezentate în Figura 22 și Figura 23. Procesul de descărcare implică transferul de la serverul *VSFTPD* la client, în timp ce procesul de încărcare reprezintă transferul de la client la server. În acest context, s-a ales investigarea a trei scenarii distincte. Primul scenariu reprezintă starea normală a rețelei, fără nicio formă de atac. În această situație, s-a obținut în medie o viteză de descărcare de 83498 *kB/s* și o viteză de încărcare de 59354 *kB/s* pentru același fișier, rezultând un timp de transfer de 23 și respectiv 32 de secunde pentru aceste operațiuni.

În al doilea scenariu, s-a inițiat un atac de tip *UDP* și s-a configurat *Snort* conform setărilor standard. În această situație, s-a observat o diferență semnificativă în timpul de transfer, deoarece viteza medie a scăzut considerabil. Astfel, în comparație cu media de 23 de secunde în starea normală, s-a înregistrat un timp de transfer de 37 de secunde pentru descărcare și unul de 39 de secunde pentru încărcare. În al treilea scenariu, *IP*-ul atacatorului a fost blocat, însă traficul continuă să fie recepționat de către *VSFTPD*. Această situație survine după câteva secunde, odată ce *Snort* detectează și blochează pachetele primite. Aici, se poate observa îmbunătățirea adusă de regula *Snort*, deoarece viteza medie de transfer se apropie de configurația normală, înregistrând o încetinire de aproximativ 5-10%.

Atacurile de tip bombă prin *e-mail* reprezintă un comportament abuziv [46], având potențialul de a provoca daune semnificative și temporare prin blocarea activităților. De obicei, atacatorii obțin adrese de *e-mail* prin intermediul formularelor *web* și ulterior inițiază atacurile asupra destinatarilor. Prin trimiterea unui număr uriaș de *e-mail*-uri, spațiul de stocare al receptorilor pe sistem sau server poate fi ocupat rapid, generând o supraîncărcare care poate duce la blocarea mașinii și oprirea funcționării întregii rețele. Această problemă afectează inaccesibilitatea altor utilizatori care utilizează serviciul și, în cazul virtualizării, poate perturba funcționarea mai multor servere aflate pe aceeași mașină. În situația în care serverul de *e-mail* este inactiv, atât serviciile de trimitere, cât și cele de primire a *e-mail*-urilor devin inutilizabile. Prin natura lor automatizată, aceste atacuri cibernetice pot provoca daune semnificative prin trimiterea a mii de *e-mail*-uri pe secundă. Datorită volumului mare de date, serverele sau sistemele devin suprasolicitate, devenind mai lente sau chiar indisponibile.

În continuare, s-a abordat problema atacurilor de tip bombă prin *e-mail* și s-au propus soluții utilizând instrumentele menționate în prezentul document. Prin capturarea și analiza traficului de rețea, s-a reușit configurarea unei reguli personalizate pentru a contracara acest tip de atac în cadrul *firewall*-ului distribuit care se propune. Soluția dezvoltată a fost prezentată în cadrul conferinței internaționale *Development and Application Systems 2022* [47].

Regula personalizată este prezentată în detaliu în articolul [47]. Performanța regulii propuse pentru combaterea atacurilor cu bombă prin *e-mail* este prezentată în cele ce urmează. S-a realizat o simulare utilizând o rețea locală în cadrul platformei *XCP-NG* [48], care reprezintă o infrastructură virtuală de tip *cloud* bazată pe *XenServer*, menționată anterior. Pentru a aduce traficul din Internet în interiorul rețelei propuse, s-au configurat *router*e, folosind *port forward*, reguli de *firewall* și *gateway* personalizat. Configurația propusă implică trimiterea traficului doar către serverul de *e-mail*, deoarece acesta este obiectivul atacatorului. În interiorul rețelei, există patru gazde: atacatorul, reprezentat de trafic simulat generat de *Scapy* și instalat pe un client *Ubuntu*, apoi sunt configurate două gazde cu *PfSense* și o mașină țintă cu *Ubuntu Server* instalată

ca server de *e-mail*. Prin această configurare experimentală, având două *firewall*-uri active, traficul incorect este blocat implicit, iar traficul legitim poate fi analizat în continuare de către *Snort*.

Pentru a evalua performanța regulii implementate, s-a generat o diagramă, prezentată în Figura 24, care ilustrează rezultatele obținute. Această diagramă evidențiază impactul soluției propuse asupra traficului de rețea și oferă o perspectivă clară asupra eficienței acesteia în detectarea și prevenirea atacurilor cu bombă prin *e-mail*. Prin simularea rețelei și analiza rezultatelor obținute, s-au putut evalua în mod obiectiv performanța regulii propuse și s-a constatat că aceasta oferă o protecție eficientă împotriva atacurilor cu bombă prin *e-mail*. Implementarea a dovedit, astfel, că soluția propusă nu numai că reușește să identifice și să blocheze atacurile cu succes, dar și să minimizeze impactul asupra traficului legitim, contribuind la eficiența generală a infrastructurii de securitate.

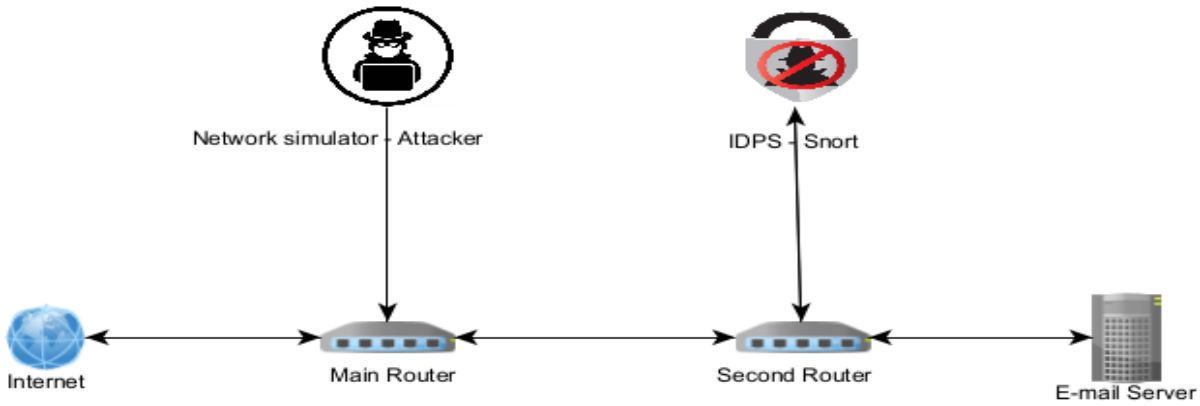



*Figura 24. Cadru experimental pentru simulare.*

În cadrul acestei cercetări, s-au parcurs mai multe faze distincte pentru a aborda problema atacurilor cu bombă prin *e-mail*. Succesiunea acestor faze este prezentată în continuare:

1. Start – S-a inițiat procesul prin lansarea traficului de atac utilizând *Scapy*, având serverul de *e-mail* ca victimă principală.
2. Capturarea traficului simulat – S-a înregistrat și colectat datele de trafic generate de atacul simulat.
3. Identificarea caracteristicilor traficului de atac – S-a analizat traficul capturat pentru a identifica caracteristicile distinctive ale atacului.
4. Inspectarea pachetelor de trafic - S-a examinat conținutul pachetelor de trafic pentru a înțelege structura și tipologia acestora.
5. *Design*-ul regulii *Snort* - S-a creat o regulă personalizată în cadrul sistemului *Snort*, adaptată specific atacului cu bombă prin *e-mail* identificat.
6. Testarea regulii create - S-au efectuat teste pentru a evalua eficacitatea regulii în detectarea și prevenirea atacului simulat.

7. Analizarea îmbunătățirii - S-au analizat rezultatele obținute pentru a evalua eficiența și performanța îmbunătățirii aduse prin implementarea regulii.
8. Final - S-a încheiat studiul, având o perspectivă asupra rezultatelor obținute și a posibilităților de optimizare în viitor.

Rezultatele obținute prin implementarea regulii personalizate sunt prezentate în Figura 25 și Figura 26. Aceste date au fost extrase din interfața grafică a platformei *PfSense*, oferind o imagine detaliată asupra evenimentelor înregistrate în cadrul rețelei. Interfața grafică *PfSense* oferă informații avansate privind comportamentul rețelei în timp real, incluzând detaliile pachetelor filtrate, resursele alocate și reacția serverului de *e-mail* la implementarea regulii personalizate.

Aceste date servesc drept bază pentru evaluarea performanței și eficacității strategiei de securitate în combaterea amenințărilor cibernetice specifice atacului cu bombă prin *e-mail*.

**Most Recent 2500 Entries from Active Log**

| Date | Action | Pri | Proto | Class | Source IP | SPort | Destination IP | DPort | GID:SID | Description |
|---|---|---|---|---|---|---|---|---|---|---|
| 2021-08-17 16:22:59 | ⚠ | 0 | TCP | | 173.92.13.211 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |
| 2021-08-17 16:22:58 | ⚠ | 0 | TCP | | 106.123.163.122 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |
| 2021-08-17 16:22:57 | ⚠ | 0 | TCP | | 242.67.71.103 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |
| 2021-08-17 16:22:57 | ⚠ | 0 | TCP | | 161.166.50.136 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |
| 2021-08-17 16:22:56 | ⚠ | 0 | TCP | | 8.5.159.247 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |
| 2021-08-17 16:22:56 | ⚠ | 0 | TCP | | 6.131.174.169 | 20 | 172.20.2.2 | 25 | 1:1000003 | MAIL BOMB ATTACK DETECTED 2021 |

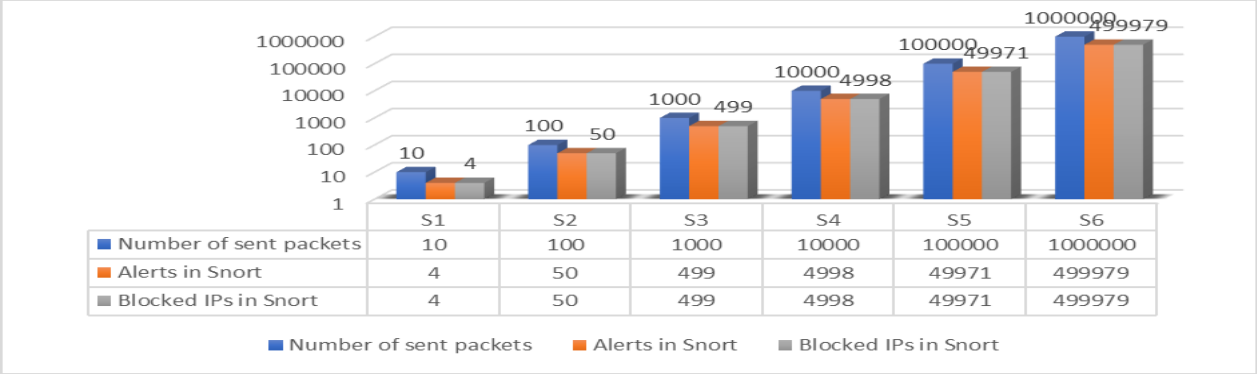
*Figura 25. Detectare atac e-mail bomb.*

**Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)**

| # | IP | Alert Descriptions and Event Times | Remove |
|---|---|---|---|
| 1 | 181.174.135.51 | MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:06:59 | ✕ |
| 2 | 175.148.142.152 | MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:00 | ✕ |
| 3 | 47.104.125.77 | MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:00 | ✕ |
| 4 | 32.169.213.158 | MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:01 | ✕ |
| 5 | 5.142.2.176 | MAIL BOMB ATTACK DETECTED 2021 -- 2021-08-17 16:07:01 | ✕ |

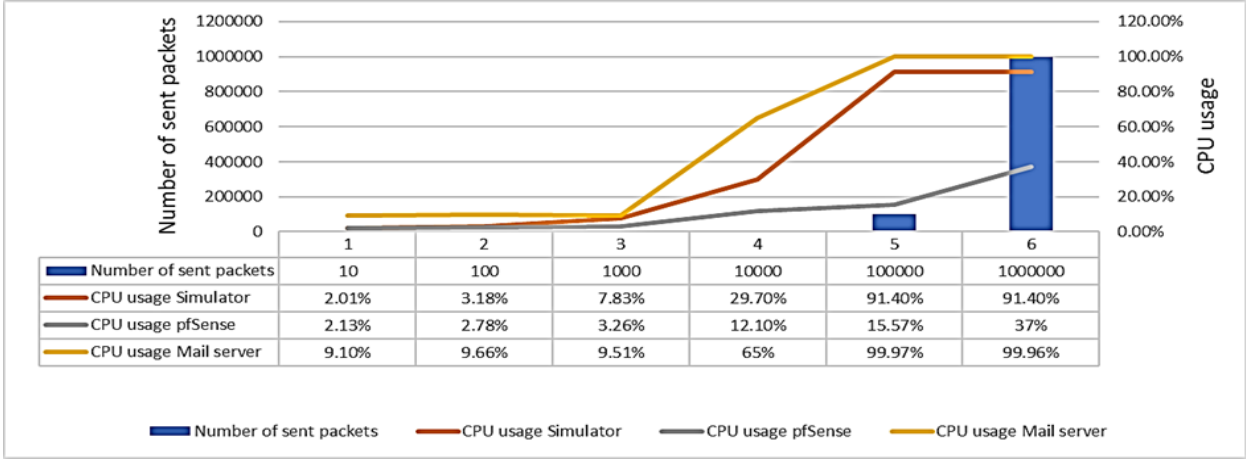
*Figura 26. Blocare sursă atac e-mail bomb.*

Diferențele semnificative dintre scenariile propuse pot fi observate în rezultatele comparative și contribuțiile prezentate în Figura 27, Figura 28 și Figura 29. Aceste grafice oferă o perspectivă clară asupra impactului implementării scenariilor și evidențiază beneficiile aduse de fiecare scenariu în parte.



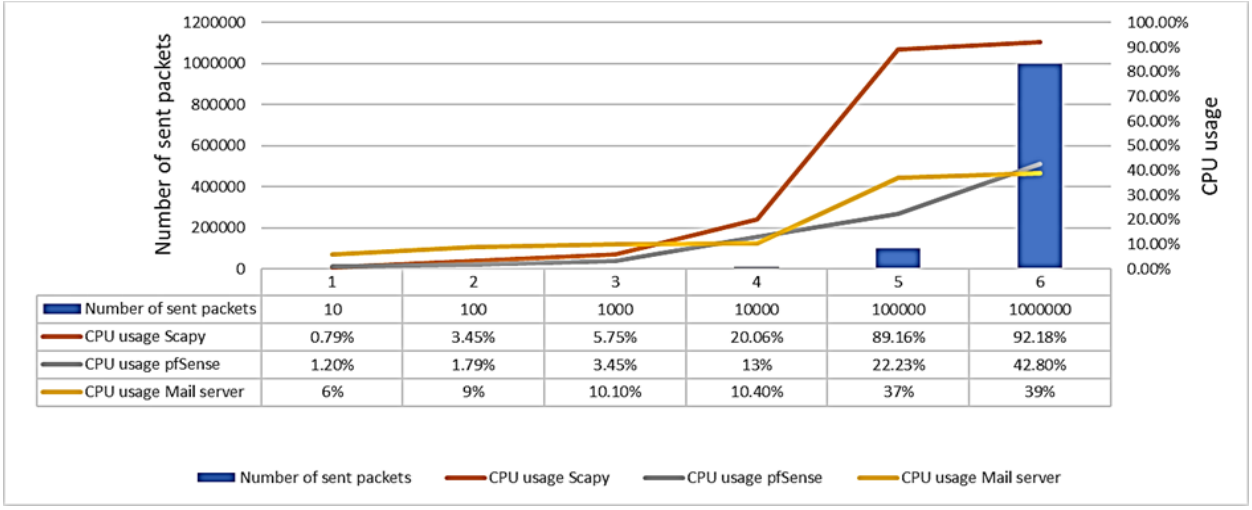

| | S1 | S2 | S3 | S4 | S5 | S6 |
|---|---|---|---|---|---|---|
| Number of sent packets | 10 | 100 | 1000 | 10000 | 100000 | 1000000 |
| Alerts in Snort | 4 | 50 | 499 | 4998 | 49971 | 499979 |
| Blocked IPs in Snort | 4 | 50 | 499 | 4998 | 49971 | 499979 |

*Figura 27. Alerte și IP-uri blocate în Snort GUI.*




| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number of sent packets | 10 | 100 | 1000 | 10000 | 100000 | 1000000 |
| CPU usage Simulator | 2.01% | 3.18% | 7.83% | 29.70% | 91.40% | 91.40% |
| CPU usage pfSense | 2.13% | 2.78% | 3.26% | 12.10% | 15.57% | 37% |
| CPU usage Mail server | 9.10% | 9.66% | 9.51% | 65% | 99.97% | 99.96% |

*Figura 28. Utilizarea CPU a VM-urilor fără Snort și regula personalizată.*




| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number of sent packets | 10 | 100 | 1000 | 10000 | 100000 | 1000000 |
| CPU usage Scapy | 0.79% | 3.45% | 5.75% | 20.06% | 89.16% | 92.18% |
| CPU usage pfSense | 1.20% | 1.79% | 3.45% | 13% | 22.23% | 42.80% |
| CPU usage Mail server | 6% | 9% | 10.10% | 10.40% | 37% | 39% |

*Figura 29. Utilizarea CPU a VM-urilor cu Snort și regula personalizată.*

Identificarea și remedierea punctelor vulnerabile într-un *firewall* distribuit prin testarea securității este fundamentală pentru protejarea informațiilor în rețea. Această metodologie implică examinarea riguroasă a *firewall*-ului pentru a identifica și remedia vulnerabilitățile, utilizând tehnici precum scanarea vulnerabilităților și testarea de penetrare, fie manual, fie cu instrumente automate precum *Nmap*, *Nessus* sau *Metasploit*. Identificarea punctelor vulnerabile permite corectarea lor prin actualizări de securitate sau alte măsuri adecvate. Auditurile de securitate sunt esențiale pentru evaluarea și remedierea vulnerabilităților, furnizând strategii pentru îmbunătățirea securității informatice.

În continuare, acest studiu propune o analiză detaliată a procesului de audit de securitate, organizată în etape distincte. În primul rând, se efectuează o sinteză succintă a literaturii relevante, inclusiv a publicațiilor recente, pentru a ilustra stadiul actual al domeniului de cercetare. Apoi, este prezentată metodologia propusă, care integrează diverse instrumente pentru identificarea breșelor de securitate, evidențiind și vulnerabilitățile publice identificate anterior și soluțiile propuse până în prezent. Rezultatele obținute sunt analizate în detaliu, iar soluțiile recomandate pentru remedierea constatărilor anterioare sunt prezentate în acest context. În final, se realizează o evaluare a rezultatelor obținute și se discută implicațiile practice și teoretice relevante.

Primul pas în testarea propusă pentru evaluarea securității *firewall*-ului distribuit este să localizăm *firewall*-ul. Deoarece nu avem adresa *IP* disponibilă, folosim o scanare simplă *Nmap* pentru a vedea ce dispozitive sunt conectate la rețea. Utilizăm comanda *Nmap* cu argumentul *-sL* pentru a face această scanare simplă. Rezultatele scanării sunt prezentate în Figura 41. Apoi, folosind *script*-urile *Nmap*, putem obține informații despre configurațiile și restricțiile *firewall*-ului. Analizând cu atenție rezultatele scanării *Nmap*, putem dezvolta un plan mai detaliat pentru identificarea și remedierea potențialelor vulnerabilități, sporind astfel securitatea sistemului.

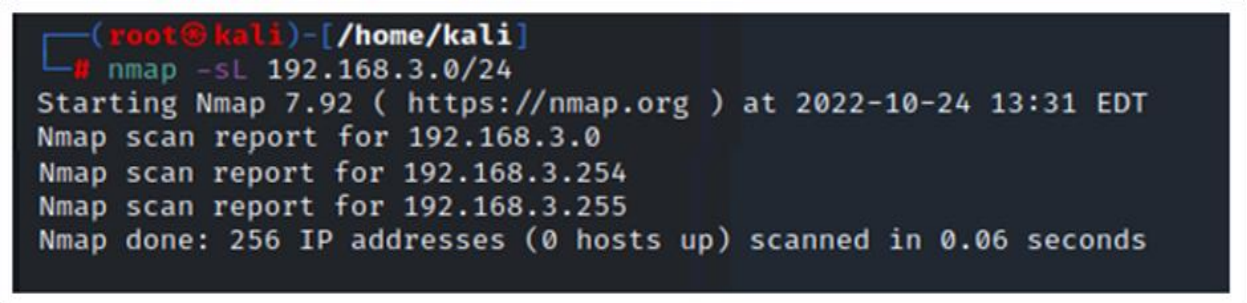



*Figura 30. Scanare inițială Nmap cu comanda -sL.*

În cadrul acestei scanări inițiale nu au fost identificate gazde active. Această situație poate afecta, din când în când, modul în care anumite sisteme de operare gestionează traficul de rețea generat de scanările porturilor. Cu toate acestea, *Nmap* dispune de câteva tehnici pe care le poate utiliza pentru a încerca localizarea acestor mașini.

Metoda următoare instruiește *Nmap* să efectueze o operațiune de *Ping* către fiecare adresă din rețeaua 192.168.3.0/24. De data aceasta, *Nmap* a returnat câteva potențiale gazde pentru scanare, prezentate în Figura 31. Opțiunea *-sn* utilizată în această comandă reprezintă un parametru pentru *Nmap*, care determină programul să efectueze doar operațiunea de *Ping* către gazdă, în locul comportamentului său implicit de a încerca scanarea gazdei prin porturi.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sn 192.168.3.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-24 13:32 EDT
Nmap scan report for 192.168.3.1
Host is up (0.00078s latency).
MAC Address: D8:0D:17:62:F1:CB (Tp-link Technologies)
Nmap scan report for 192.168.3.3
Host is up (0.067s latency).
MAC Address: 5E:78:DA:3A:BF:09 (Unknown)
Nmap scan report for 192.168.3.7
Host is up (0.046s latency).
MAC Address: 0E:47:5A:B2:4E:8B (Unknown)
Nmap scan report for 192.168.3.16
Host is up (0.075s latency).
MAC Address: C8:CB:9E:7E:A3:8D (Unknown)
Nmap scan report for 192.168.3.19
Host is up (0.00045s latency).
MAC Address: B4:2E:99:1A:A2:3A (Giga-byte Technology)
Nmap scan report for 192.168.3.20
Host is up (0.00038s latency).
MAC Address: 30:9C:23:64:EA:C5 (Micro-star Intl)
Nmap scan report for 192.168.3.100
Host is up (0.0010s latency).
MAC Address: 6E:59:2B:AB:92:A9 (Unknown)
Nmap scan report for 192.168.3.10
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.85 seconds
```

*Figura 31. Scanare inițială Nmap cu comanda -sn.*

În etapa următoare, s-a utilizat scanarea porturilor *Nmap* pentru a identifica dispozitivele disponibile pe aceste gazde specificate. Prezența porturilor deschise indică existența unui serviciu activ pe sistemul respectiv. Numărul mare de porturi deschise pe acest server se datorează faptului că adresa *IP* 192.168.3.100 este atribuită echipamentului de rețea, în acest caz fiind vorba despre *router*-ul de *internet*.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.3.100,100-102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-24 13:39 EDT
Nmap scan report for 192.168.3.100
Host is up (0.00039s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
8500/tcp open  fmtp
MAC Address: 6E:59:2B:AB:92:A9 (Unknown)

Nmap done: 3 IP addresses (1 host up) scanned in 5.62 seconds
```

*Figura 32. Scanarea Nmap pentru porturi deschise.*

Pentru a obține informații despre porturile deschise pe *firewall*-ul utilizat în experiment, trebuie rulat *Nmap* cu specificația adresei *IP* și portul de destinație. Opțiunea prezentată în Figura 32 permite obținerea acestor informații. În contextul dat s-au identificat două porturi deschise. Portul 53 este destinat serviciului *DNS*, iar portul 8500 este utilizat pentru conexiunea la interfața *web*.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sA 192.168.3.100 -p 80
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-24 13:07 EDT
Nmap scan report for 192.168.3.100
Host is up (0.00048s latency).

PORT   STATE    SERVICE
80/tcp filtered http
MAC Address: 6E:59:2B:AB:92:A9 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

*Figura 33. Scanare Nmap pentru portul 80.*

Pentru a realiza analize digitale sau testări de securitate, este prioritar să se înțeleagă infrastructura rețelei, inclusiv serverele și alte dispozitive situate între sistemul analizat și ținta în cauză. Acest aspect este esențial în identificarea amprentei digitale a rețelei respective. De exemplu, experții în securitate nu pot ataca direct un server *web* fără a verifica mai întâi existența unui *firewall.* În Figura 33, s-au prezentat rezultatele unei astfel de scanări, care au evidențiat prezența unui *firewall*. Pentru a investiga aceste aspecte, instrumentul *Traceroute* [49] se dovedește util.

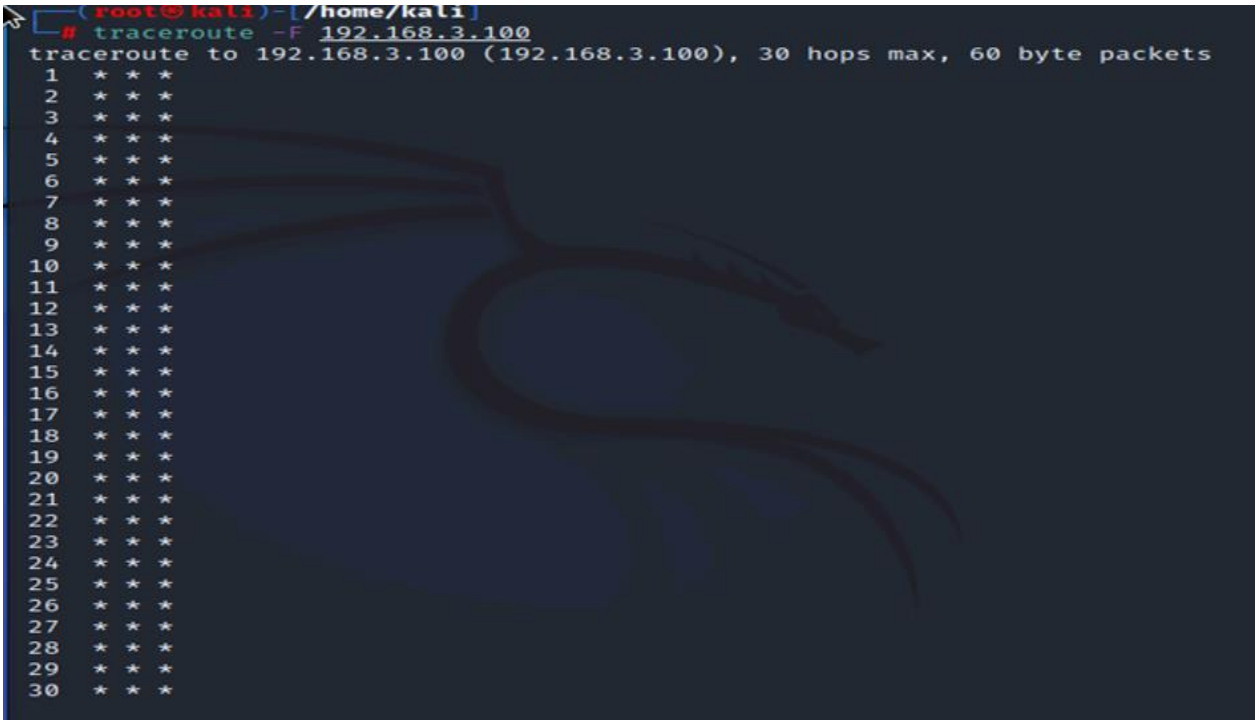



*Figura 34. Rezultatul Traceroute.*

Atunci când se utilizează comanda *Traceroute* pentru a examina un dispozitiv și rezultatele sunt afișate sub forma unor asteriscuri în consolă, situația similară întâlnită și prezentată în Figura 34, indică faptul că dispozitivul nu a fost configurat să răspundă la traficul *ICMP*/*UDP*. Acest rezultat nu înseamnă că nu a existat trecere de trafic. O altă posibilitate este că o problemă în rețea a cauzat pierderea pachetelor. Aceste rezultate, în general, indică *timeout*-uri ale pachetelor sau trafic respins de un *firewall,* așa cum este cazul în configurația propusă. În aceste condiții, este esențial să se efectueze o analiză detaliată a configurației *firewall*-ului pentru a identifica eventualele reguli care pot afecta trecerea pachetelor și să se implementeze ajustări corespunzătoare.

În continuare, s-a utilizat instrumentul *Netcat* [50] pentru a testa preluarea de *bannere*. Aceasta este o tehnică utilizată pentru a identifica porturile deschise și serviciile de rețea disponibile pe un sistem informatic. Administratorii de sistem pot utiliza această tehnică pentru a cataloga dispozitivele și serviciile din rețelele lor. Cu toate acestea, un atacator poate folosi *banner grabbing* pentru a identifica *site*-uri de rețea care rulează versiuni de programe sau sisteme de operare care prezintă vulnerabilități de securitate cunoscute. Astfel, s-au comparat rezultatele obținute cu *Netcat*, prezentate în Figura 35, cu rezultatele obținute cu *Nmap*, prezentate în Figura 36.

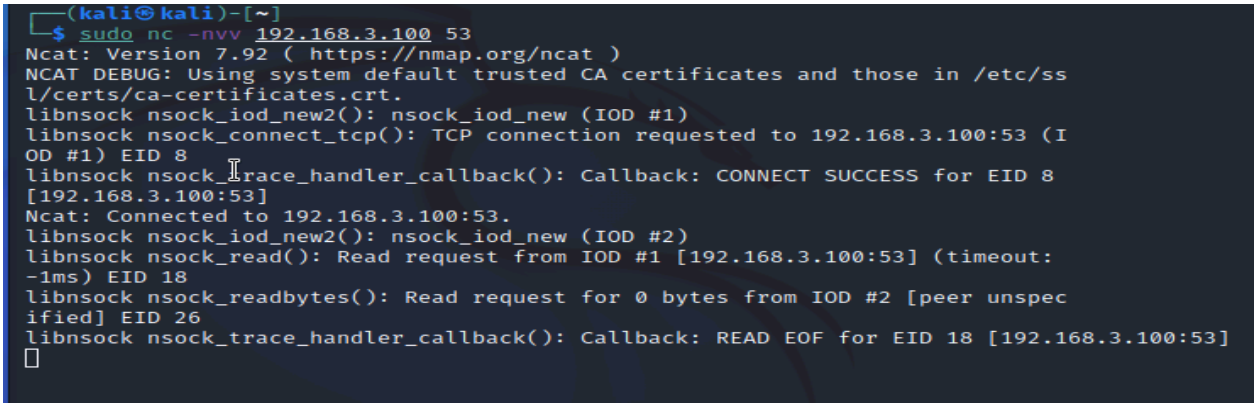



*Figura 35. Netcat Banner grabbing.*

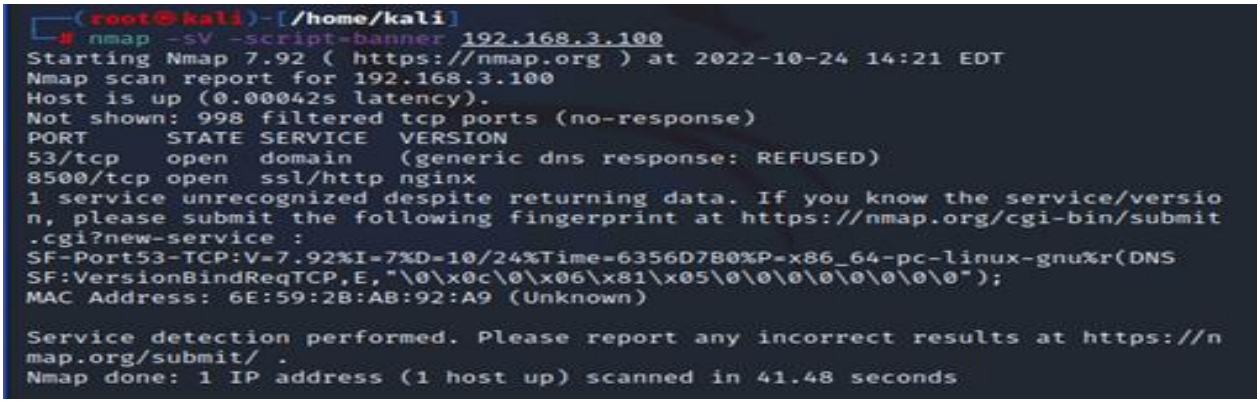



*Figura 36. Nmap Banner grabbing.*

Procesul de scanare prezentat în Figura 35 și Figura 36 nu a furnizat informații suplimentare referitoare la etapele anterioare. *Nmap* a indicat că există o singură gazdă activă, și anume *firewall*-ul principal din experimentul propus.

Pentru a adăuga o abordare suplimentară, s-a utilizat *Firewalk* [51], un *script* distinct care se poate executa în configurația propusă. Acest instrument este util în evaluarea nivelului de securitate al *hardware*-ului de filtrare a pachetelor, cum ar fi cel găsit în sistemele de *firewall.*

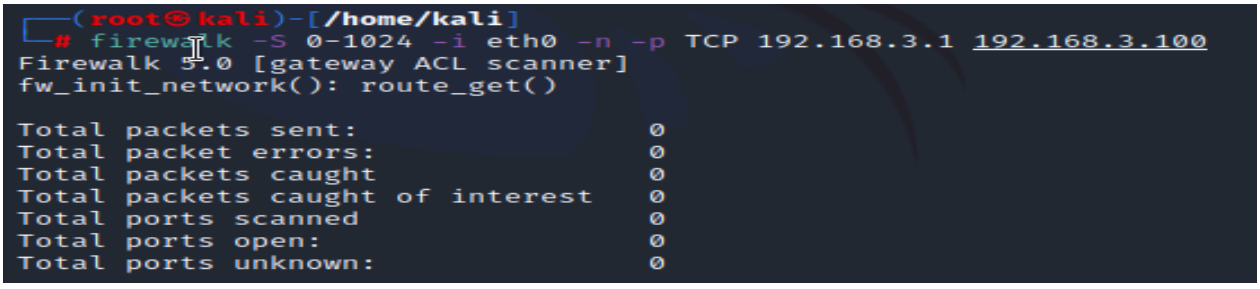



*Figura 37. Rezultate obținute cu Firewalk.*

În Figura 37 sunt prezentate rezultatele scanării. Parametrii utilizați în acest scenariu se referă la intervalul de porturi scanate (de la 0 la 1024, care reprezintă porturile bine-cunoscute pentru scanare), *eth0* indică interfața pe care rulează *Firewalk*, opțiunea **-n** a fost utilizată pentru a evita

rezolvarea numelor de gazdă în adrese *IP* folosind serviciul *DNS*, *TCP* indică protocolul utilizat, iar adresele *IP* ulterioare se referă la sursa și destinația scanării. De obicei, pachetele de date asociate cu acțiunile de scanare, care sunt interzise de o listă de control al accesului (*ACL*) [52] sau de un *firewall,* sunt pierdute sau respinse. Dacă acestea sunt permise să treacă, ele expiră și generează un mesaj *ICMP* de depășire a timpului.

În acest scenariu, soluția implementată de *firewall*-ul distribuit a reușit să blocheze activitățile de recunoaștere încercate prin *script*. Deși există și alte *script*-uri disponibile în *Nmap*, *Netcat* și *Firewalk*, majoritatea nu au furnizat informații utile în cadrul experimentului propus. S-a ales acest *script* întrucât este *open source*, gratuit și are sprijinul unei comunități active. În cadrul acestui studiu, am constatat că testarea manuală a securității este costisitoare și consumă mult timp. Pentru a realiza experimentul, s-a examinat documentația legată de aceste instrumente, s-a analizat condițiile probabile și s-au adaptat testele pentru a obține rezultatele dorite. Apoi, s-a efectuat un test de penetrare automatizat și s-au colectat informații pentru a înțelege mai bine problemele de securitate care pot fi identificate în configurația propusă.

Pentestul automatizat oferă un avantaj evident în ceea ce privește viteza și scalabilitatea față de testele de penetrare manuale, însă adâncimea și eficacitatea acestora nu pot fi egale. Pentru a evalua *firewall*-ul distribuit și a înțelege în profunzime eventualele vulnerabilități întâlnite, s-a efectuat un pentest automatizat, repetând procesul de mai multe ori pentru a obține rezultate consistente și pentru a observa orice modificări. Durata scanării complete a arhitecturii a fost de aproximativ 15 minute, iar rezultatele obținute sunt prezentate mai jos. Pentestul automatizat a oferit o perspectivă cuprinzătoare asupra vulnerabilităților identificate și a capacității de apărare a *firewall*-ului distribuit. Cu toate acestea, testele manuale de penetrare rămân esențiale pentru identificarea unor vulnerabilități complexe și pentru evaluarea interacțiunii dintre sistemele componente. Pentestul automatizat reprezintă un instrument valoros, ce ar trebui utilizat împreună cu testele manuale în cadrul unei abordări integrate a testării și evaluării securității.

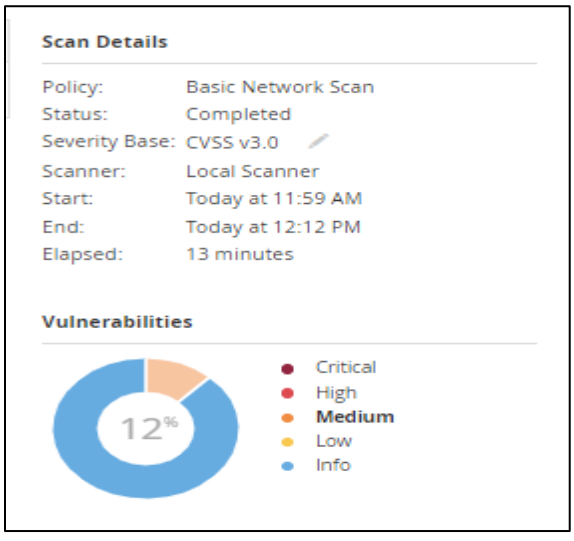



*Figura 38. Rezultatele scanării - vulnerabilități medii.*

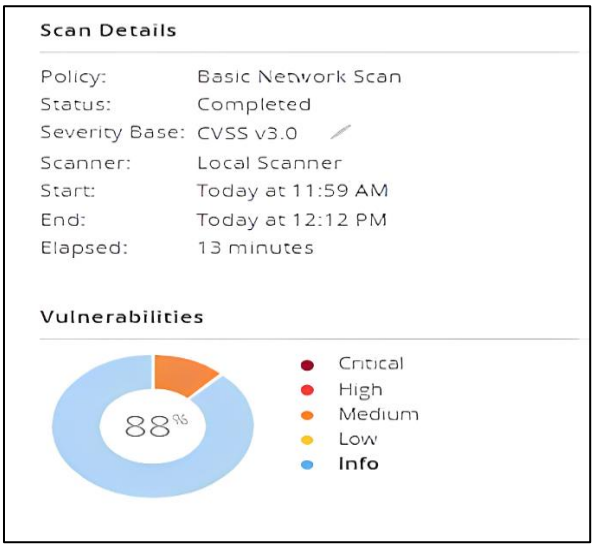



*Figura 39. Rezultatele scanării – Info.*

În Figura 38 și Figura 39, se prezintă rezultatele scanării efectuate de *Nessus*. Din totalul de 37 de obiecte analizate, 12% reprezintă vulnerabilități de nivel mediu, în timp ce restul de 88% reprezintă probleme informative care necesită atenție, dar nu prezintă un risc critic. Aceste constatări subliniază importanța analizei atente a rezultatelor scanării și a prioritizării remedierii în funcție de gravitatea și factorul de risc asociat fiecărei vulnerabilități identificate.

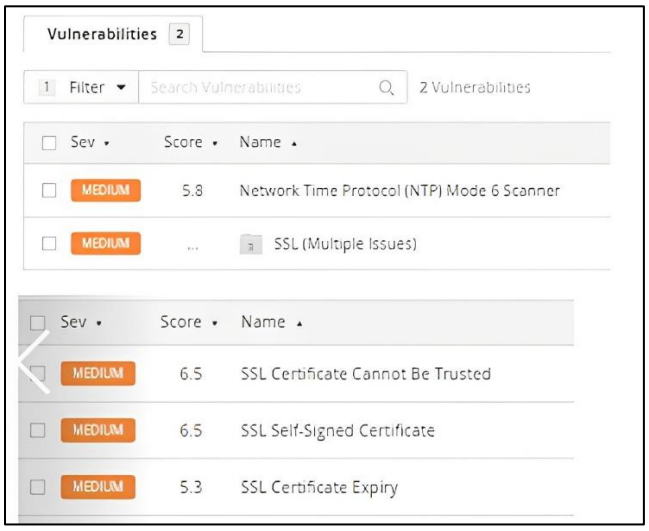



*Figura 40. Vulnerabilități medii detectate.*

Figura 40 ilustrează cele mai semnificative amenințări identificate în cadrul experimentului propus. Serviciul *Nessus* furnizează o descriere concisă și soluții pentru fiecare vulnerabilitate descoperită. În cazul prezentat, remedierea acestor probleme a fost ușoară datorită performanței remarcabile asigurate de *pfSense*.


*Figura 41. Log-urile Firewall în timpul experimentului.*

În Figura 41 se poate observa portul pe care scanerul *Nessus* a efectuat testele din cadrul scenariului propus. Protocolul *TCP* asigură livrarea pachetelor de date pe portul 53684 în aceeași ordine în care au fost transmise, oferind o comunicare garantată. În schimb, portul *UDP* 53684 nu beneficiază de aceleași garanții de comunicare ca *TCP*.

În contextul prezentat, în Tabel 2 sunt expuse problemele descoperite prin intermediul comunităților publice [53]. Aceste probleme reprezintă diverse tipuri de vulnerabilități identificate în versiunile anterioare ale platformei *pfSense*. În experimentul prezentat, s-a utilizat versiunea 2.6.0 a *pfSense*, fiind ultima versiune stabilă disponibilă pentru public.

*Tabel 2. Vulnerabilități pfSense*

| ID-ul vulnerabilității | Detaliile vulnerabilității | Data publicării | Data ultimei actualizări |
|---|---|---|---|
| CVE-2022-42247 | S-a descoperit că pfSense [2.5.2] conține o vulnerabilitate de *scripting cross-site* (*XSS*) în componenta browser.php. Această vulnerabilitate permite atacatorilor să execute *script*-uri *web* sau *HTML* arbitrar prin intermediul unui *payload* manipulat injectat în numele unui fișier. | 2022-10-03 | 2022-10-05 |
| CVE-2022-23993 | În versiunile anterioare lui [2.6.0] pentru *pfSense* și [22.01] pentru *pfSense Plus*, există o vulnerabilitate în calea /usr/local/www/pkg.php, unde se folosește $_REQUEST['pkg_filter'] într-o comandă *PHP echo*, cu potențialul de a cauza *Cross-Site Scripting*. | 2022-01-26 | 2022-04-29 |
| CVE-2021-41282 | Fișierul diag_routes.php aflat în versiunea pfSense [2.5.2] permite injectarea de date *sed*. Se intenționează ca utilizatorii autentificați să poată vizualiza date despre rutele setate în *firewall*. Datele sunt recuperate prin executarea utilitarului *netstat*, iar apoi ieșirea acestuia este analizată prin intermediul utilitarului *sed*. Deși sunt utilizate mecanismele obișnuite de protecție împotriva injecției de comenzi (de exemplu, utilizarea funcției *escapeshellarg* pentru argumente), este totuși posibil să se injecteze cod specific *sed* și să se scrie un fișier arbitrar într-o locație arbitrară. | 2022-03-01 | 2022-07-12 |
| CVE-2021-27933 | Versiunea de *pfSense* [2.5.0] permite *scripting cross-site* prin intermediul câmpului *Description* din *services_wol_edit.php.* | 2021-04-28 | 2021-05-01 |
| CVE-2021-20729 | Vulnerabilitatea de *script*-ing încrucișat în *pfSense CE* și *pfSense Plus* (versiunile [2.5.2] și anterioare ale software-ului *pfSense CE* și versiunile [21.05] și anterioare ale software-ului pfSense Plus) permite unui atacator de la distanță să injecteze un script arbitrar prin intermediul unui *URL* malițios. | 2022-03-31 | 2022-04-08 |
| CVE-2020-26693 | A fost descoperită o vulnerabilitate *XSS* stocată în versiunea [2.4.5-p1] care permite unui atacator autentificat să execute *script*-uri *web* arbitrare prin exploatarea funcției load_balancer_monitor.php. | 2021-06-01 | 2021-06-09 |
| CVE-2016-10709 | *pfSense* înainte de versiunea [2.3] permite utilizatorilor autentificați de la distanță să execute comenzi arbitrare ale sistemului de operare prin intermediul unui caracter '\|' în parametrul grafic status_rrd_graph_img.php, legat de _rrd_graph_img.php. | 2018-01-22 | 2018-02-09 |
| CVE-2011-5047 | Vulnerabilitatea *XSS* în interiorul fișierului *status_rrd_graph.php* din *pfSense* înainte de versiunea [2.0.1.0] permite atacatorilor de la distanță să injecteze *script*-uri *web* sau *HTML* arbitrare prin intermediul parametrului *style*. | 2012-01-03 | 2017-08-29 |
| CVE-2011-4197 | Fișierul etc/inc/certs.inc din implementarea *PKI* în pfSense înainte de versiunea [2.0.1] creează fiecare certificat X.509 cu o valoare adevărată pentru constrângerea de bază *CA*, ceea ce permite atacatorilor de la distanță să creeze subcertificate pentru subiecți arbitrari prin utilizarea cheii private. | 2012-01-03 | 2017-08-29 |

Anterior s-au abordat diverse metode pentru identificarea vulnerabilităților în infrastructura de securitate. Figura 42 ilustrează procesul de descoperire a punctelor slabe de securitate în arhitectura propusă, unde s-au utilizat diferite instrumente de scanare a securității pentru a identifica potențialele probleme de securitate.

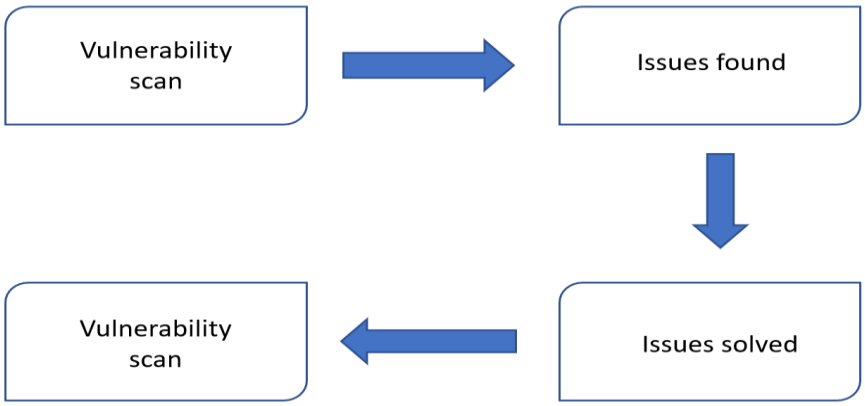



*Figura 42. Diagrama de evaluare a securității.*

În continuare, s-au examinat soluțiile propuse pentru a aborda problemele identificate și evidențierea beneficiilor *firewall*-ului distribuit. Există mai multe abordări posibile pentru a rezolva aceste probleme. În primul rând, o soluție eficientă este utilizarea unui instrument de capturare a pachetelor, cum ar fi *Wireshark*. Utilizând *Wireshark* în timpul scanării, se poate analiza traficul și se pot crea reguli personalizate pentru *Suricata* sau *Snort*, care pot fi folosite pentru a bloca metodele de scanare simulate. Totuși, trebuie avut în vedere că există și alte instrumente de scanare care pot utiliza tehnici diferite pentru a ocoli regulile propuse.

O altă perspectivă importantă este remedierea problemelor identificate prin intermediul configurațiilor personalizate ale *firewall*-ului principal. Utilizând instrumentul *Nmap*, s-au identificat *router*-ul din rețea și porturile deschise care pot fi exploatate. După îmbunătățirea configurării *pfSense*, s-au obținut următoarele rezultate ale scanării *Nmap*:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sL 192.168.3.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-07 04:47 EST
Nmap scan report for 192.168.3.0
Nmap scan report for 192.168.3.1
Nmap scan report for 192.168.3.254
Nmap scan report for 192.168.3.255
Nmap done: 256 IP addresses (0 hosts up) scanned in 0.07 seconds
```

*Figura 43. Scanare Nmap cu opțiunea -sL.*

Prin implementarea modificărilor aduse, s-a reușit ascunderea *firewall*-ul de scanarea rețelei, așa cum se poate observa în Figura 43 și Figura 44. Acest aspect este deosebit de important, deoarece în absența unui *router*, procesele ulterioare de *pentesting* devin ineficiente din cauza lipsei de informații disponibile. *PfSense*, în calitatea sa de *firewall,* oferă multiple protecții împotriva diferitelor tipuri de probleme care pot apărea într-o rețea locală, confirmând astfel eficiența sa ca soluție.

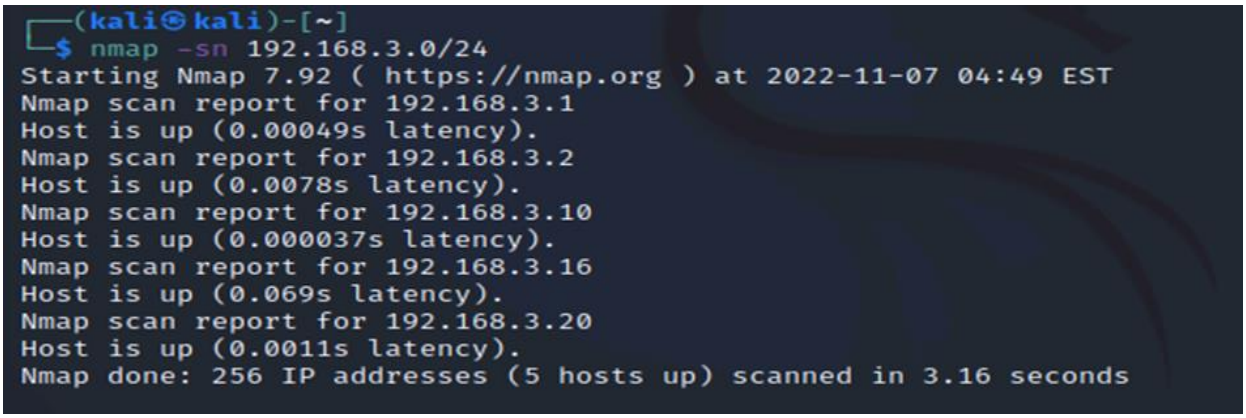



*Figura 44. Scanare Nmap cu opțiunea -sn.*

În ceea ce privește testarea automatizată cu ajutorul instrumentului *Nessus*, s-a reușit rezolvarea problemelor identificate și obținerea următoarelor rezultate, prezentate în Figura 45.

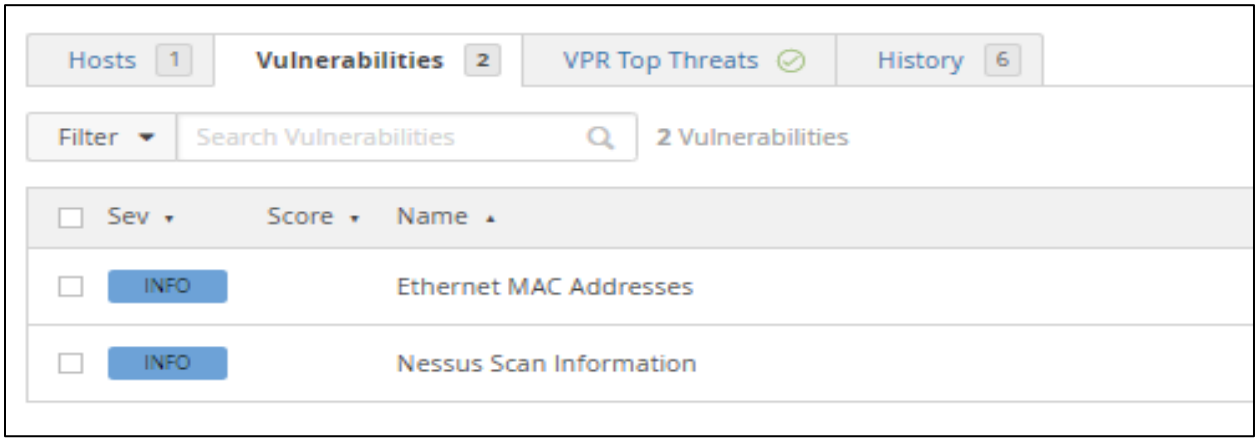



*Figura 45. Raportul Nessus după ajustări.*

În Figura 45 este prezentat raportul rezultat în urma rezolvării problemelor identificate anterior. Cele mai multe probleme au fost remediate prin optimizarea *firewall*-ului cu utilizarea diverselor tehnici, cum ar fi ajustarea politicii de securitate, eliminarea serviciilor neutilizate și editarea unor fișiere specifice prin intermediul interfeței de linie de comandă.

Raportul de risc este esențial în studiul propus din mai multe motive. Acesta utilizează diferite scenarii și evaluează amenințările în funcție de probabilitatea apariției lor. Se bazează pe expertiză profesională, intuiție și experiență, ordonând riscurile în funcție de severitatea lor.

Raportul de risc este calculat folosind următoarea formulă [54]:

*Risc = Impact x Amenințare x Vulnerabilitate*

Pentru evaluarea riscului, se poate utiliza următoarea scală: vulnerabilitățile de nivel scăzut au un scor de impact cuprins între 1 și 16, vulnerabilitățile de nivel moderat sunt cuprinse de un scor între 17 și 32, vulnerabilitățile de nivel înalt sunt în intervalul 33-48, iar vulnerabilitățile critice se situează între 49 și 64.

Implementarea de actualizări de securitate este o tehnică comună pentru protecția sistemelor, însă nu este întotdeauna eficientă în fața vulnerabilităților de tip *zero-day*. Vulnerabilitățile *zero-day* sunt deficiențe de securitate neanunțate, care pot fi exploatate de atacatori pentru a obține acces neautorizat la sistemele vulnerabile. *Software*-ul antivirus și *firewall*-urile sunt esențiale pentru prevenirea intruziunilor, iar *firewall*-urile monitorizează traficul în rețea pentru a preveni accesul neautorizat și comunicările ostile.

Pentru a face față acestei provocări, se propune o metodă progresivă pentru automatizarea dezvoltării regulilor de *firewall* dinamice, folosind un *API* și un *script Python*. Această abordare are scopul de a menține și actualiza în mod autonom regulile de *firewall*, îmbunătățind astfel securitatea rețelei și simplificând identificarea și atenuarea vulnerabilităților de atac *zero-day*. Metoda propusă a fost evaluată pe o rețea simulată și s-a constatat că este eficientă în detectarea vulnerabilităților *zero-day* și prevenirea accesului neautorizat. Studiul a fost valorificat prin publicarea în cadrul revistei *AECE* [55], dobândind astfel validarea rezultatelor.

Regulile automate de *firewall* dinamice permit organizațiilor să reacționeze în mod proactiv la amenințările emergente în timp real, oferind o apărare eficientă împotriva exploatărilor *zero-day* și a altor atacuri sofisticate [56]. Se propune o soluție nouă numită *Sistemul de Reguli Dinamice Automate*, care urmărește automatizarea creării și actualizării regulilor de *firewall* pe baza analizei în timp real a traficului de rețea în curs de intrare. Pentru a realiza acest lucru, se folosesc un *API* și un *script Python* pentru a colecta și procesa datele provenite de la *Snort*, care identifică anomalii în traficul de rețea. Prin utilizarea datelor colectate, *script*-ul *Python* generează și implementează automat noi reguli de *firewall* în rețeaua distribuită. Această generare dinamică a regulilor permite rețelei să răspundă prompt la amenințările emergente, oferind o apărare robustă împotriva exploatărilor *zero-day* și a altor tehnici avansate de atac.

Figura 46 ilustrează configurația experimentală, care cuprinde o rețea locală și mai multe componente interconectate. Aceasta include o mașină virtuală care servește ca atacator, echipată cu un simulator de trafic de rețea. Una dintre mașinile virtuale este configurată pentru a rula *pfSense* și *Snort*, în timp ce o altă mașină virtuală rulează *script*-ul *Python* responsabil de procesarea informațiilor colectate. Împreună, aceste două mașini alcătuiesc mecanismul *firewall* propus.

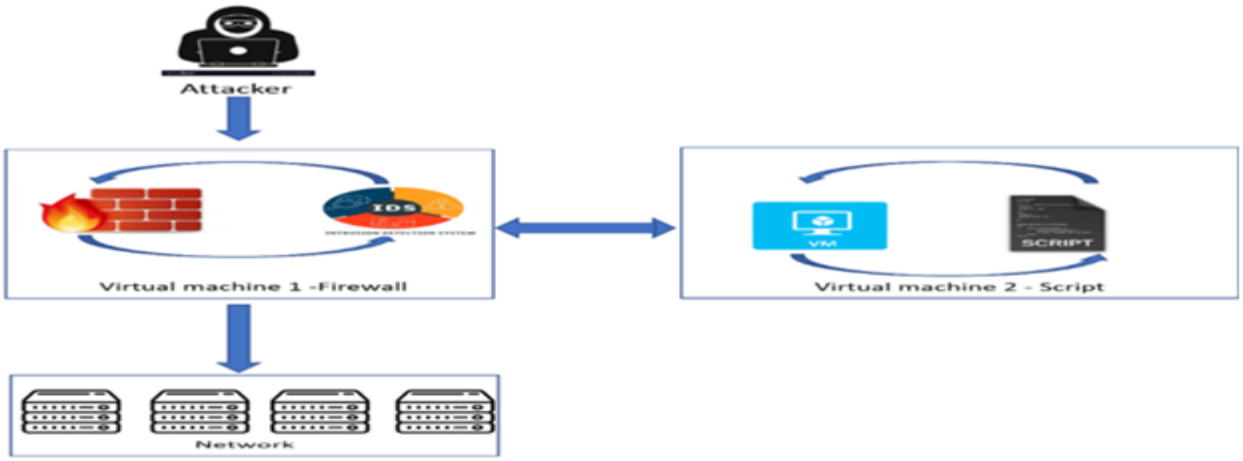



*Figura 46. Circuitul traficului din cadrul rețelei.*

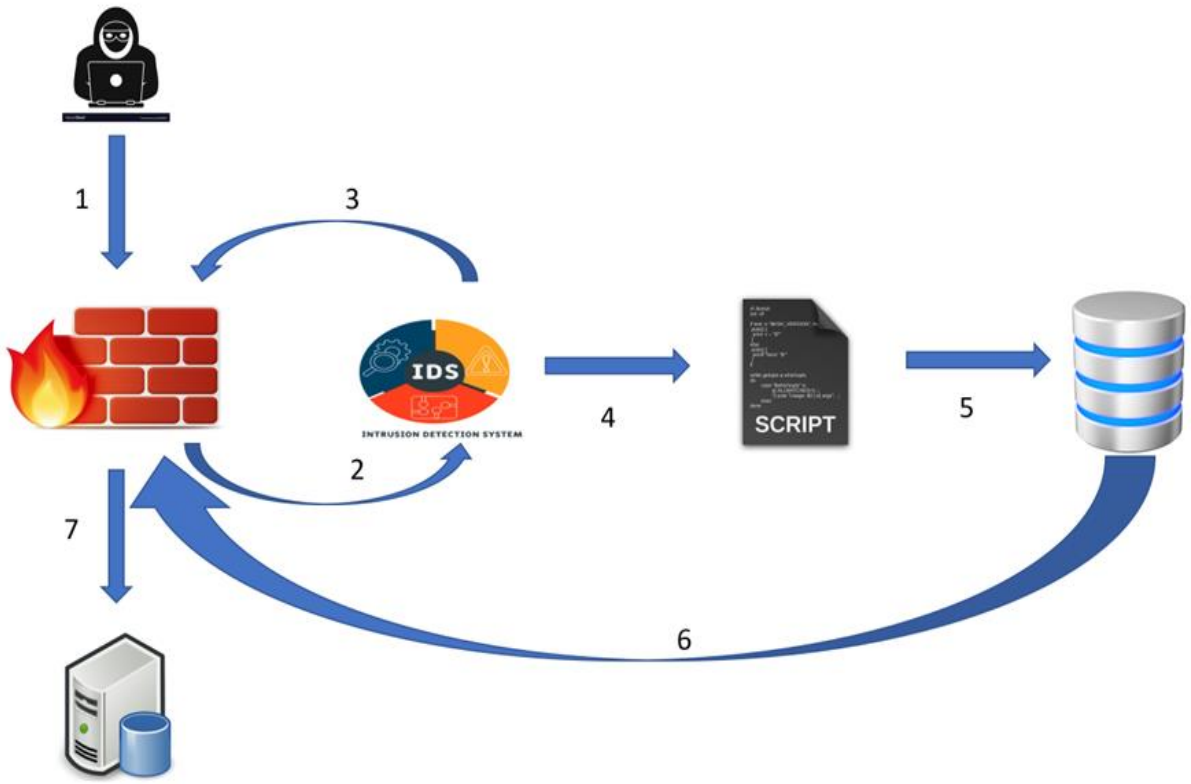



*Figura 47. Procesul propus pentru crearea de reguli automate.*

În Figura 47 este ilustrată diagrama pentru procesul de automatizare propus. *Script*-ul furnizat automatizează gestionarea regulilor de *firewall* prin analizarea jurnalelor de trafic de intrare, crearea de reguli noi pentru blocarea unor adrese *IP* specifice și eliminarea regulilor expirate. Acesta preia *log*-urile prin intermediul unui *API*-uri, le procesează pentru a extrage adrese *IP* și porturi, verifică dacă o adresă *IP* este deja blocată, creează reguli noi, dacă este necesar, și verifică setul de reguli actualizat. De asemenea, ține evidența regulilor create prin stocarea informațiilor relevante și șterge regulile expirate.

*Script*-ul funcționează într-un ciclu, resetând variabilele și așteptând un interval specificat între iterații. În ansamblu, oferă o abordare automatizată pentru menținerea unei configurații eficiente a *firewall*-ului pe baza analizei traficului în timp real.

S-a testat soluția propusă într-un mediu de rețea realist pentru asigurarea eficacității și eficienței sale. *Script*-ul *Python* a fost testat pentru automatizarea creării de reguli dinamice de *firewall*, așa cum reiese din Figura 48. S-au simulat mai multe scenarii de atac, inclusiv amenințări cunoscute și noi, pentru a testa capacitatea soluției propuse de a răspunde și de a reacționa cu precizie la diverse atacuri de rețea în timp real, îmbunătățind securitatea rețelei.

Prin monitorizarea consumului de resurse și a stabilității generale a rețelei, s-a obținut o imagine detaliată asupra comportamentului soluției în condiții de stres. Cu toate acestea, într-un efort continuu de a asigura securitatea cibernetică în fața unor amenințări în evoluție, viitoarele direcții de dezvoltare vizează îmbunătățiri ale performanțelor, extinderea capacităților de analiză a traficului și integrarea cu tehnologii emergente în domeniul securității rețelelor.

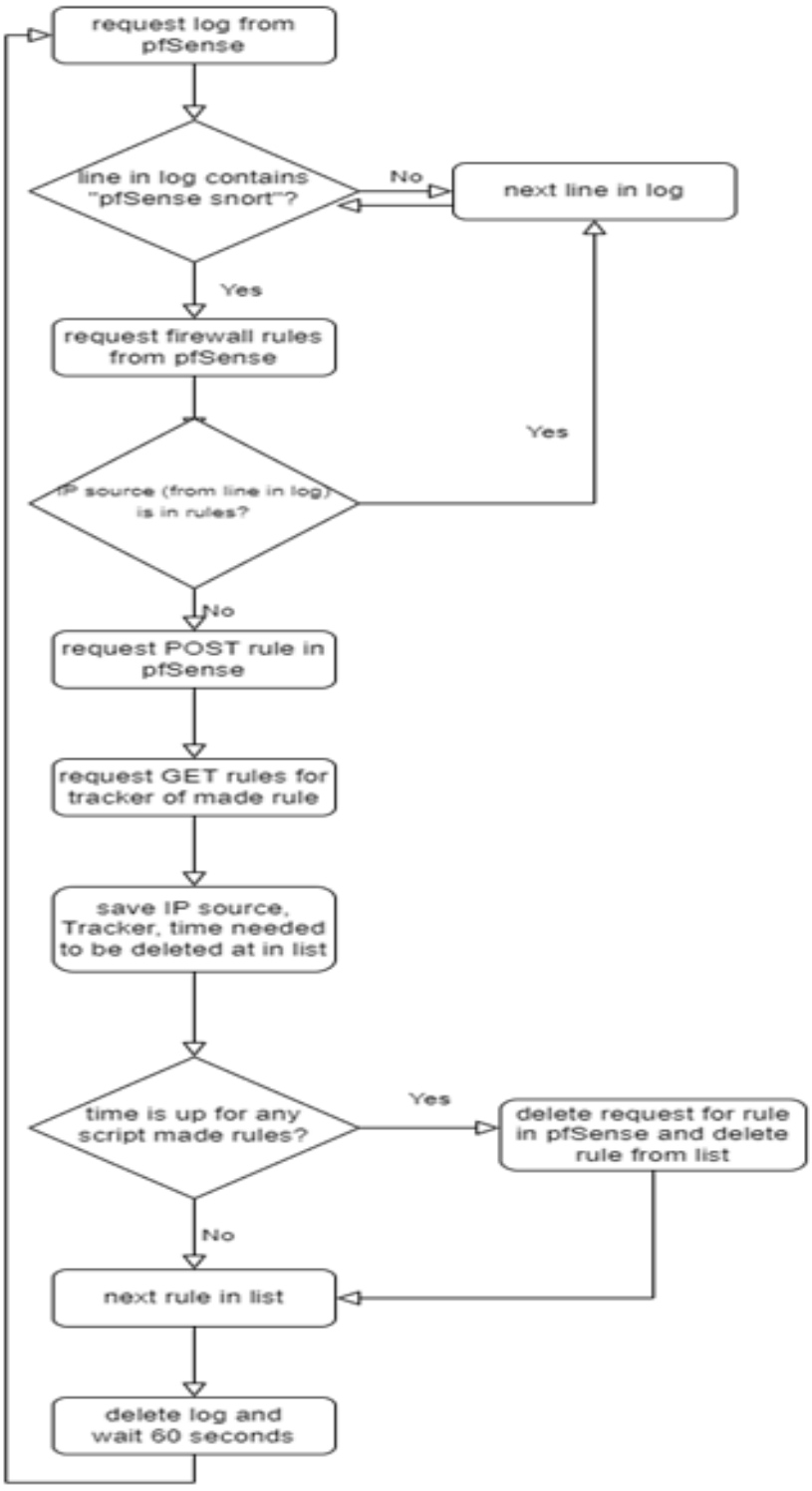



*Figura 48. Diagramă proces creare reguli Firewall automate.*

Platforma *POSTMAN* servește drept instrument fundamental pentru comunicarea cu utilizatorii prin intermediul consolei, furnizând notificări detaliate cu privire la orice schimbare în starea sau mesajul unei reguli atunci când aceasta suferă modificări prin intermediul *script*-ului asociat.

```
{
    "status": "ok",
    "code": 200,
    "return": 0,
    "message": "Success",
    "data": [
        {
            "id": "",
            "tracker": "1666112103",
            "type": "pass",
            "interface": "wan",
            "ipprotocol": "inet",
            "tag": "",
            "tagged": "",
            "max": "",
            "max-src-nodes": "",
            "max-src-conn": "",
            "max-src-states": "",
            "statetimeout": "",
            "statetype": "keep state",
            "os": "",
            "source": {
                "address": "192.168.3.0/24"
            },
            "destination": {
                "any": ""
            },
            "descr": "Retea fizica",
            "created": {
                "time": "1666112103",
                "username": "admin@172.20.1.2 (Local Database)"
            },
            "updated": {
                "time": "1666112118",
                "username": "admin@172.20.1.2 (Local Database)"
```


*Figura 49. Rezultat comandă GET pentru reguli.*

Figura 49 prezintă rezultatele executării comenzii *GET* în *POSTMAN*, efectuată pentru a obține informațiile din baza de date *pfSense*.


```
{
    "status": "ok",
    "code": 200,
    "return": 0,
    "message": "Success",
    "data": [
        "Dec 28 16:23:00 pfSense newsyslog[37782]: logfile turned over due to size>500K",
        "Dec 28 16:23:00 pfSense sshguard[95104]: Exiting on signal.",
        "Dec 28 16:23:00 pfSense sshguard[39265]: Now monitoring attacks.",
        "Dec 28 16:23:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:23:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:23:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:23:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:23:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:23:58 pfSense snort[49267]: [120:3:2] (http_inspect) NO CONTENT-LENGTH OR
        "Dec 28 16:24:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
        "Dec 28 16:24:58 pfSense snort[49267]: [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP
```


*Figura 50. Rezultat comandă GET pentru log-uri Snort.*

Metoda prin care *pfSense* trimite *log*-urile *Snort* este una dintre cele mai importante componente ale acestei soluții. Rezultatele generate de *POSTMAN* atunci când *log*-urile sunt exportate sunt prezentate în Figura 50.

Următorul pas după exportarea și analizarea logurilor de către *script*, este construirea regulii în *pfSense* și apoi transmiterea acesteia. În Figura 51 sunt arătați parametrii folosiți, iar în Figura 52, se arată modul în care operația este realizată folosind *POSTMAN*.

Query Params

| | KEY | VALUE |
|---|---|---|
| ☑ | dst | any |
| ☑ | interface | wan |
| ☑ | ipprotocol | inet |
| ☑ | src | 220.240.233.57 |
| ☑ | srcport | any |
| ☑ | type | block |
| ☐ | disabled | false |
| ☑ | descr | test |
| ☑ | protocol | any |
| ☑ | apply | true |
| | Key | Value |

*Figura 51. Parametri utilizați pentru reguli*


```
{
    "status": "ok",
    "code": 200,
    "return": 0,
    "message": "Success",
    "data": {
        "type": "block",
        "interface": "wan",
        "ipprotocol": "inet",
        "source": {
            "address": "220.240.233.57"
        },
        "destination": {
            "any": ""
        },
```


*Figura 52. Rezultat comandă POST pentru reguli.*

Detaliile privind construcția și utilizarea regulilor sunt expuse în detaliu în Figura 53, furnizând un cadru vizual clar și comprehensibil al acestui proces în cadrul platformei *POSTMAN*. Prin intermediul acestei abordări, utilizatorii sunt informați și ghidați eficient în toate etapele procesului, facilitând astfel o gestionare precisă și o utilizare eficientă a regulilor în contextul securității cibernetice.


```
{
    "status": "ok",
    "code": 200,
    "return": 0,
    "message": "Success",
    "data": {
        "type": "block",
        "interface": "wan",
        "ipprotocol": "inet",
        "source": {
            "address": "220.240.233.57"
        },
        "destination": {
            "any": ""
        },
        "descr": "test",
        "tracker": 1672316774,
        "created": {
            "time": 1672316774,
            "username": "admin@192.168.3.19 (API)"
        },
        "updated": {
            "time": 1672316774,
            "username": "admin@192.168.3.19 (API)"
        }
    }
}
```


*Figura 53. Creare cu succes regulă firewall.*

Experimentele au implicat simularea diverselor scenarii de trafic de rețea folosind instrumentul de testare a *API POSTMAN*. Prin emularea condițiilor reale, s-a putut evalua modul în care *script*-ul propus a creat și gestionat reguli de *firewall* pe baza traficului de intrare detectat de *Snort*. Pentru experimentul propus, s-a simulat trafic prin *LOIC* către *firewall*-ul propus pentru a declanșa alerta *Snort*.

După cum s-a discutat anterior, alerta a activat *script*-ul pentru a crea, încărca și aplica regula de *firewall*. S-au luat în considerare următorii parametri: ținta selectată a fost *firewall*-ul, cu adresa *IP* 192.168.3.100, portul destinație a fost 80, care este portul implicit pentru traficul *HTTP*. Metoda a fost *UDP*, ceea ce înseamnă că pachetele de atac au fost trimise ca datagrame *UDP*, care nu necesită stabilirea unei conexiuni și sunt folosite în mod obișnuit pentru aplicații în timp real care necesită o latență redusă. *LOIC* a utilizat 100.000 de fire de execuție, fiecare fir de execuție reprezentând o sursă de atac către sistemul țintă și a trimis pachete cât mai rapid posibil. Viteza maximă oferită de *LOIC* a fost utilizată în cazul prezentat, ceea ce înseamnă că pachetele au fost trimise la o rată maximă posibilă.

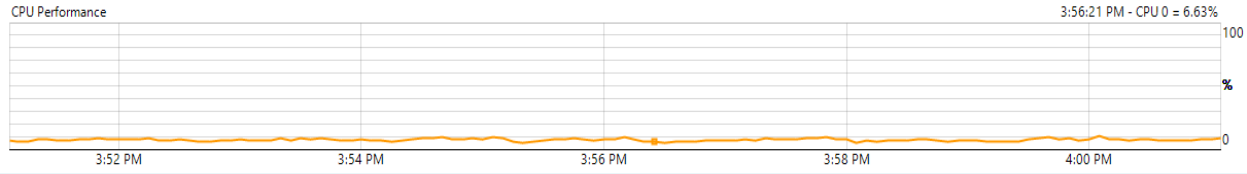



*Figura 54. Utilizare CPU în mod normal.*

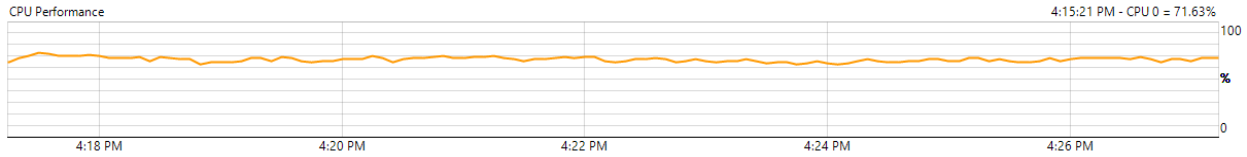



*Figura 55. Utilizare CPU în timpul atacului.*

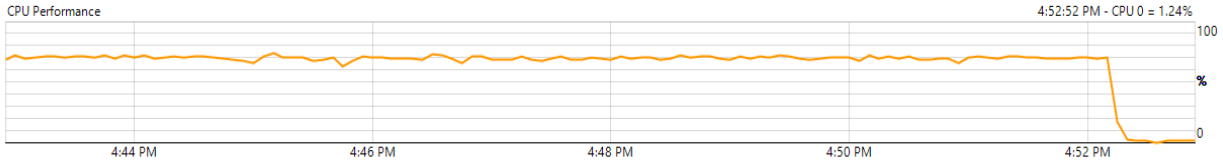



*Figura 56. Utilizare CPU după rularea script-ului.*

În cadrul experimentului propus, s-a monitorizat activitatea pe *firewall*-ul folosit pentru a evidenția avantajele utilizării *script*-ului sugerat. *Log*-urile utilizate în acest scop sunt incluse în platforma utilizată pentru virtualizare, care este utilizată cu scop de monitorizare.

Conform datelor prezentate în Figura 54, Figura 55 și Figura 56 atacul *LOIC* asupra *firewall*-ului a efectuat o creștere semnificativă a procentului de utilizare a *CPU*-ului, de la 6,63% la 71,6%. Sunt prezentate dovezi incontestabile că atacul ce a folosit *LOIC* a avut un efect semnificativ asupra *firewall*-ului și a reușit să-l inunde cu trafic. Apoi, rularea unui *script* care a construit o regulă de *firewall* pentru a interzice traficul dăunător de la *LOIC* a determinat o scădere semnificativă a procentului de utilizare a *CPU*-ului, de la 71,63% la 1,24%, așa cum reiese din Figura 56.

*Tabel 3. Analiza desfășurării experimentului*

| **Timp** | **Tipul de trafic** | ***IP* sursă** | ***IP* destinație** | **Port destinație** | **Număr de pachete** |
|---|---|---|---|---|---|
| 10:00:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 1,000 |
| 10:01:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 2,000 |
| 10:02:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 3,000 |
| 10:03:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 4,000 |
| 10:04:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 5,000 |
| 10:05:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 6,000 |
| 10:06:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 7,000 |
| 10:07:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 8,000 |
| 10:08:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 9,000 |
| 10:09:00 | Atac | LOIC (aleatoriu) | *Firewall* (192.168.3.100) | 80 | 10,000 |
| 10:10:00 | Regula de blocare a firewall-ului | *Firewall* (192.168.3.100) | LOIC (aleatoriu) | 80 | N/A |
| 10:11:00 | Normal | *Firewall* (192.168.3.100) | N/A | N/A | N/A |

Datele prezentate în Tabel 3 au fost colectate în cadrul unui experiment, al cărui scop a fost de a ilustra eficacitatea unui *script* în protejarea împotriva unui atac *LOIC* asupra unui *firewall*. Obiectivul experimentului a fost de a simula un atac asupra *firewall*-ului prin trimiterea de datagrame *UDP* către *firewall* cu ajutorul *LOIC*, în scopul inundării *firewall*-ului cu trafic. Experimentul a fost conceput pentru a imita un atac asupra *firewall*-ului.

În continuare se prezintă abordarea propusă în dezvoltarea unui sistem de management pentru un *firewall* distribuit, care oferă multiple beneficii ce sunt detaliate ulterior. Pentru a obține datele privind utilizarea *CPU* și lățimea de bandă a rețelei pentru fiecare mașină virtuală și algoritmul de optimizare, s-a utilizat un *script* specific și s-au procesat informațiile colectate. *API*-ul a fost utilizat pentru a accesa datele din platforma de virtualizare și pentru a ajusta procesul de optimizare al *firewall*-ului. Astfel, soluția propusă poate monitoriza întreaga rețea și poate lua decizii bazate pe parametrii stabiliți pentru a asigura o securitate de înaltă performanță. *Script*-ul dezvoltat gestionează mașinile virtuale din cadrul platformei de virtualizare, utilizând *API*-ul pentru a obține date și a modifica algoritmul de optimizare al *firewall*-ului.

Pentru gestionarea completă a platformei *XCP-NG,* se utilizează interfața *API*, numită *XAPI* [48]. *XAPI* utilizează o bază de date pentru citire/scriere pe magistrală, permițând replicarea numai pentru citire la clienți. Un fișier *XML* numit *State.db* este folosit pentru a păstra toate setările și metadatele pentru grupul configurat, inclusiv informații despre gazde și altele.

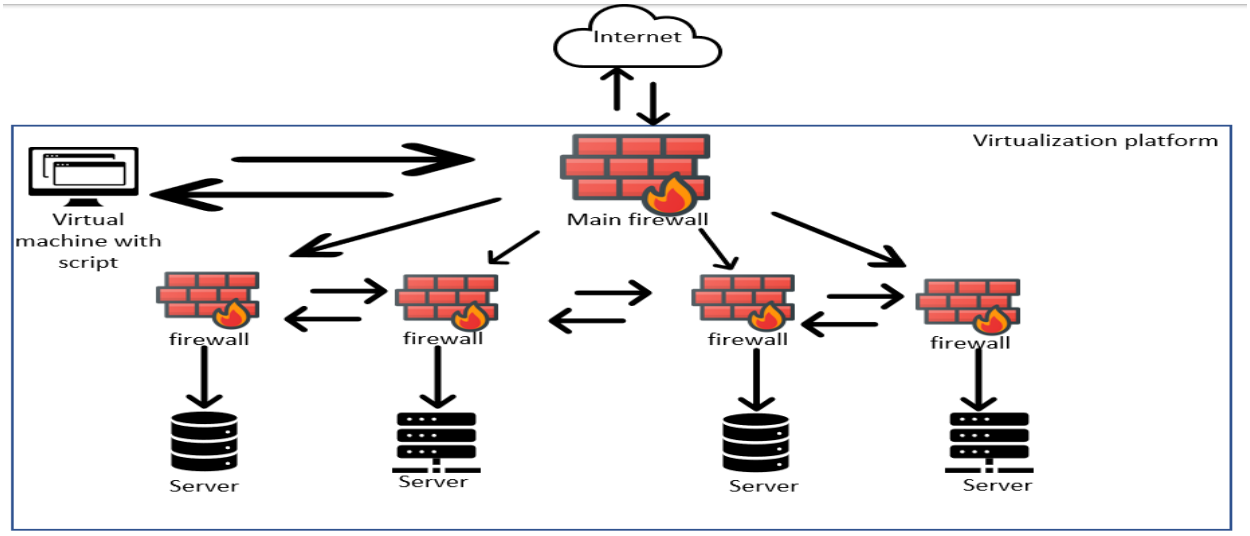



*Figura 57. Arhitectura propusă pentru sistemul de management.*

În vederea optimizării eficienței *script*-ului, s-a decis găzduirea *script*-ului pe o mașină virtuală separată în cadrul infrastructurii virtualizate. Amplasarea acestei mașini virtuale, care îndeplinește multiple scopuri, este ilustrată în Figura 57. S-a considerat că această mașină virtuală se află la același nivel cu *router*-ul principal, permițându-i să fie utilizată în toate straturile configurației propuse. Pentru a asigura o protecție suplimentară, s-a luat în considerare posibilitatea de a crea o clasă virtuală exclusiv pentru această mașină virtuală, la care administratorul de rețea poate avea acces doar prin intermediul unei rute statice.

Modul de optimizare, care este controlat prin intermediul *script*-ului propus, determină intervalul de timp în care expiră intrările din tabela de stare ale *firewall*-ului. Opțiunea "*Normal*"

se referă la algoritmul de optimizare standard, care este optim în majoritatea situațiilor. Pentru legăturile cu latență ridicată, cum ar putea fi cele realizate prin satelit, folosim opțiunea "*High Latency*", în care conexiunile inactive expiră mai târziu decât în modul implicit. Modul "*Aggressive*" este utilizat atunci când dorim să eliminăm conexiunile inactive mai rapid din tabela de stare. Această opțiune poate îmbunătăți performanța în implementările cu trafic intens și multe conexiuni, precum serviciile *web*, chiar dacă solicită o utilizare mai intensă a resurselor *CPU* și a memoriei. Opțiunea finală este "*Conservative*", în care *firewall*-ul încearcă să mențină toate conexiunile legitime, cu costul unei utilizări mai mari a memoriei și a procesorului. Acest mod poate fi util în aplicații care necesită conexiuni *UDP* de lungă durată, dar în mare parte inactive, cum ar fi *VoIP*. În continuare, s-a prezentat *script*-ul pentru scenariul propus, acesta fiind inclus și într-o conferință internațională cu tema creării de rețele în educație și cercetare [57].

În vederea funcționării eficiente, un *firewall* utilizează o tabelă de stare pentru a stoca dinamic informații despre conexiunile active permise de regulile sale. Fiecare intrare în tabel definește o conexiune în funcție de următoarele criterii. În ceea ce privesc protocoalele, *TCP*, *UDP* și *ICMP* sunt modalitățile implicite prin care serviciile se conectează și comunică între ele. Fiecare dispozitiv are asignată o adresă *IP*, atât pentru dispozitivele locale, cât și pentru cele de la distanță.

*Script*-ul *Bash* propus permite modificarea algoritmului de optimizare pe mașinile virtuale *pfSense*, care sunt instalate pe *XCP-NG*, utilizând *API*-ul. În configurația propusă, se folosesc cinci *router*e, unul dintre acestea fiind *router*-ul principal, folosit pentru rutare și redirecționare, iar celelalte pentru filtrarea pachetelor. Schimbarea stării unui algoritm este reprezentată în Figura 58 și este detaliată în acest capitol.

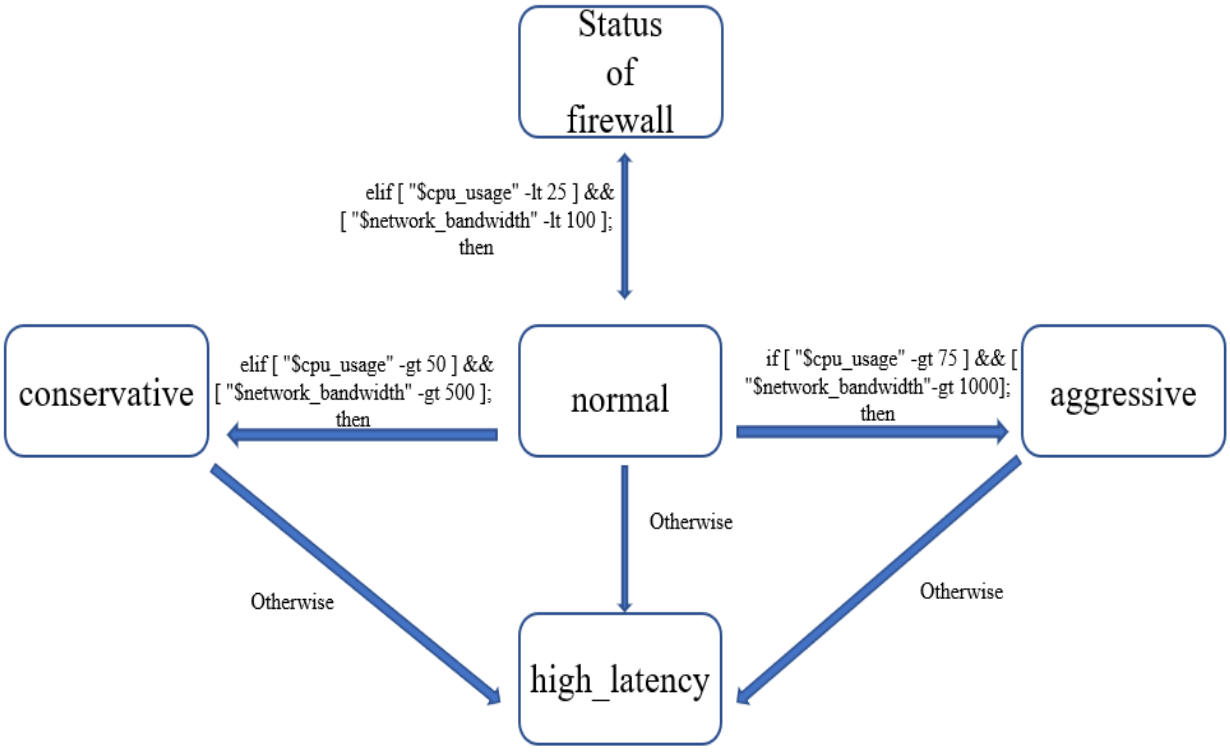



*Figura 58. Diagrama algoritmului propus.*

Există multipli parametri care pot fi utilizați pentru a evidenția beneficiile *script*-ului. În ceea ce privește utilizarea resurselor, se poate măsura consumul de *CPU* și de memorie al mașinilor virtuale înainte și după rularea *script*-ului, astfel încât să se compare rezultatele și să se demonstreze eficiența acestuia în optimizarea utilizării resurselor. Pentru testarea performanței unui *script* de rețea, este esențial să se măsoare performanța rețelei. Prin măsurarea lățimii de bandă și a latenței mașinilor virtuale înainte și după rularea *script*-ului, se pot compara rezultatele și demonstra că *script*-ul este eficient în optimizarea performanței rețelei.

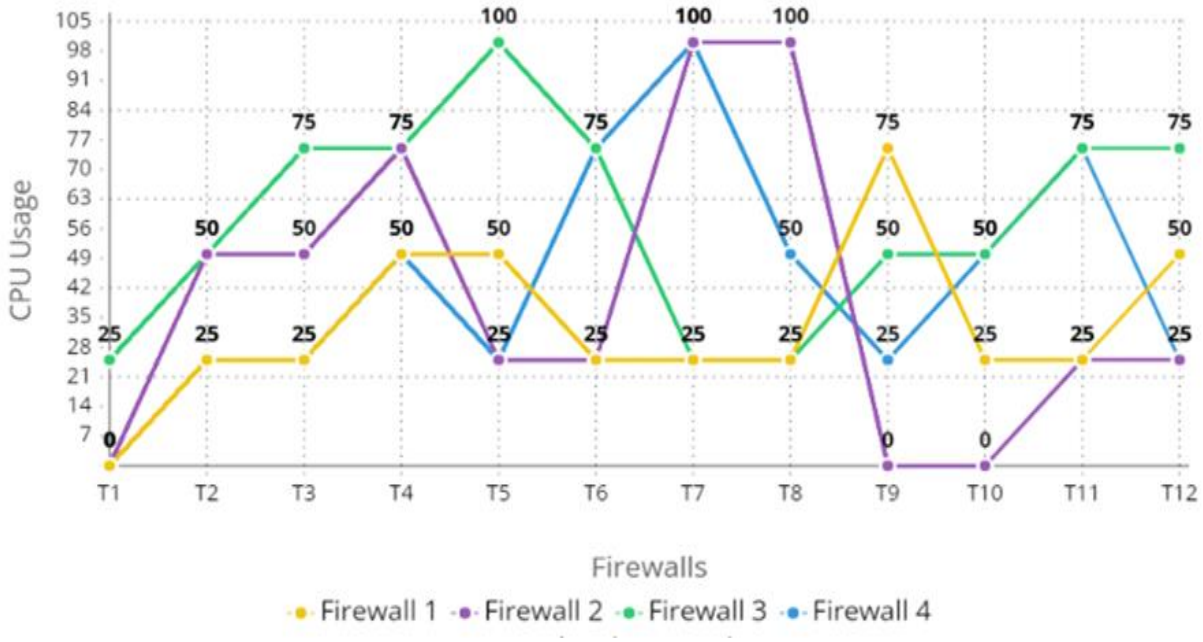



*Figura 59. Modificarea politicii de Firewall în timp real.*

Figura 59 prezintă măsurătorile analizate în cadrul experimentului propus. Durata monitorizării *firewall*-urilor a fost de 12 ore, în timpul cărora s-au înregistrat și afișat modificările de optimizare. Datorită traficului care a trecut prin *firewall*-urile propuse, algoritmul de optimizare a fost modificat prin intermediul *script*-ului utilizat. În cadrul studiului, s-a observat că un trafic intens, îndreptat către un singur dispozitiv *firewall*, a cauzat congestie în rețea, afectând comunicarea fluidă. Pentru a remedia această problemă, s-a decis să se împartă încărcarea între mai multe dispozitive *firewall,* optimizând astfel procesarea sarcinilor.

Automatizarea autentificării utilizatorilor într-un *firewall* distribuit constă în integrarea unui sistem de autentificare în structura *firewall*-ului, facilitând gestionarea centralizată a accesului utilizatorilor la resursele rețelei și consolidând securitatea acestui proces. Această procedură poate fi realizată prin intermediul *scripting*-ului și tehnologiei *Lightweight Directory Access Protocol* (*LDAP*) [58]. *LDAP* este o tehnologie utilizată pentru gestionarea centralizată a autentificării utilizatorilor și a accesului lor la resursele rețelei, fiind, în general, folosită pentru autentificarea utilizatorilor în serviciile director. Integrarea *script*-urilor și a tehnologiei *LDAP* într-un *firewall* distribuit poate contribui la îmbunătățirea performanței și securității rețelei, prin gestionarea centralizată a autentificării utilizatorilor și a accesului la resursele rețelei.

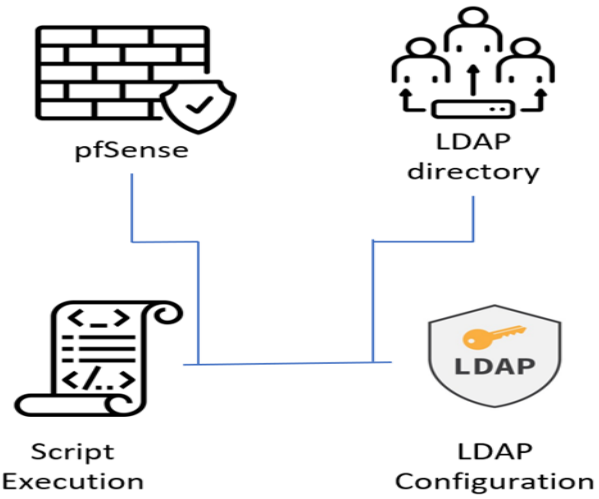



*Figura 60. Diagrama poziției script-urilor în rețeaua propusă.*

Configurarea *firewall*-ului *pfSense* utilizează autentificarea *LDAP* pentru gestionarea conturilor de utilizator, așa cum este exemplificat în Figura 60. În vederea automatizării acestei configurații, se execută un *script* care integrează autentificarea *LDAP* și definește regulile de *firewall* pe baza rolurilor utilizatorilor. *Script*-ul interacționează direct cu directorul *LDAP* pentru autentificarea utilizatorilor și preluarea atributelor acestora, urmând apoi să configureze *firewall*-ul *pfSense* pentru a restricționa accesul utilizatorilor în funcție de aceste atribute.

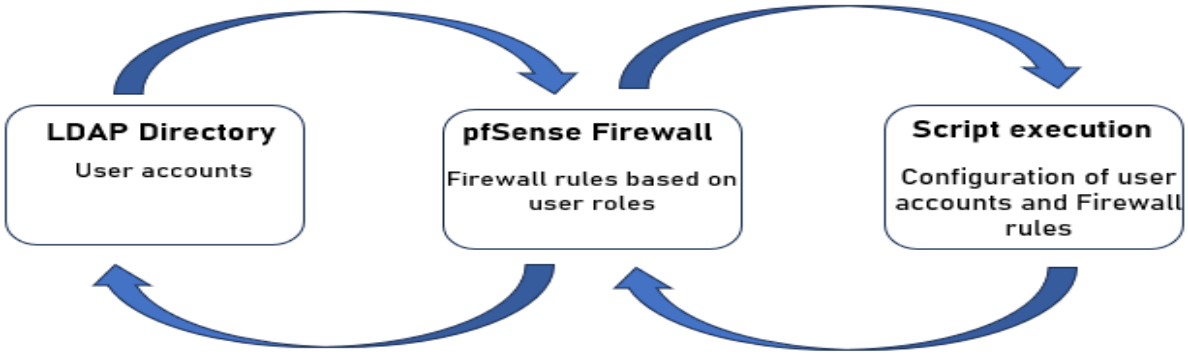



*Figura 61. Arhitectura propusă experiment LDAP.*

Evidențierea prezenței directorului *LDAP*, care găzduiește conturile de utilizator și atributele acestora, utilizate în procesul de autentificare a utilizatorilor este ilustrată în Figura 61. În același timp, *firewall*-ul *pfSense* este configurat cu reguli de *firewall* adaptate în funcție de rolurile de utilizator. Etapa de executare a *script*-ului automatizează configurarea conturilor de utilizator și a regulilor de *firewall,* folosind directorul *LDAP* ca sursă principală de informații despre utilizatori.

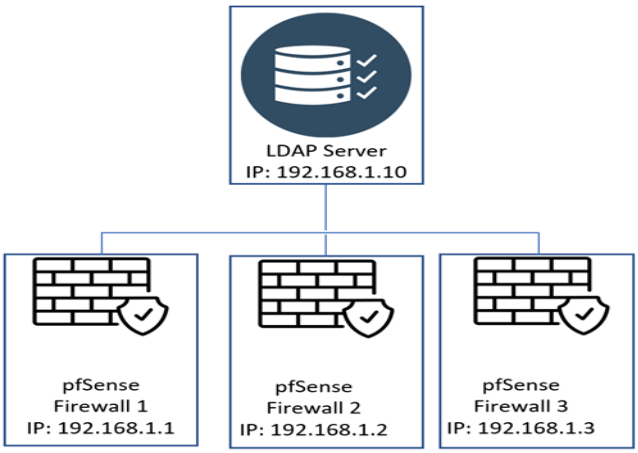



*Figura 62. Configurarea serverului LDAP pentru Firewall distribuit.*

Serverul *LDAP* este implementat ca un sistem independent de *firewall*-urile *pfSense*, primind adresa *IP* 192.168.1.10 în cadrul rețelei, așa cum este prezentat în Figura 62. *Firewall*-urile *pfSense* din configurarea *firewall*-urilor distribuite sunt identificate prin adresele *IP* 192.168.1.1, 192.168.1.2 și 192.168.1.3. *Script*-ul modificat este executat pe fiecare *firewall pfSense* și stabilește o comunicare în rețea cu serverul *LDAP* în vederea autentificării utilizatorilor.

Prin evaluarea acestor aspecte și prin comparația *script*-ului propus cu metoda tradițională de configurare manuală, se pot demonstra avantajele și beneficiile aduse de utilizarea acestui *script* într-un context real.

*Tabel 4. Analiza comparativă a timpului*

| Metoda | Timp pentru configurarea unui *firewall* (minute) | Timp pentru configurarea a trei *firewall*-uri (minute) |
|---|---|---|
| Manual | 60 | 180 |
| *Script* | 15 | 45 |

În Tabel 4 s-a comparat implementarea unui singur *firewall* cu cea a unei rețele de *firewall*-uri distribuite în cadrul scenariului analizat. Așa cum s-a menționat anterior, scalabilitatea joacă un rol esențial atunci când există un număr mare de *firewall*-uri. Sunt situații în care administratorii de rețea trebuie să configureze multiple *firewall*-uri în diverse scenarii, iar utilizarea *script*-ului propus poate aduce economii semnificative de timp în acest proces. Durata necesară pentru configurarea unui *firewall* este o sarcină dificil de estimat, întrucât acest lucru depinde de specificul scenariului și de nivelul de expertiză al persoanei care se ocupă de configurarea și rularea *script*-urilor.

*Tabel 5. Estimări ale rezultatelor soluției propuse*

| | Configurație manuală | Configurație automatizată |
|---|---|---|
| Timp petrecut (ore) | 24 | 4 |
| Numărul de erori | 6 | 1 |
| Disponibilitatea *firewall*-ului (minute) | 3,000 | 3,600 |
| Numărul de modificări ale configurației | 12 | 12 |
| Timpul mediu între defecțiuni (*MTBF*) (ore) | 200 | 400 |

În Tabel 5 s-au inclus mai multe informații relevante pentru a compara procesul de configurare manuală cu cel automat în ceea ce privesc *firewall*-urile. S-au înregistrat numărul de erori întâlnite în timpul ambelor procese, disponibilitatea *firewall*-urilor pe durata configurației, numărul de modificări de configurare efectuate și timpul mediu între defecțiuni (*Mean Time Between Failures - MTBF*) al *firewall*-urilor, care reprezintă perioada de timp în care acestea funcționează fără probleme.

În continuare se prezintă o descriere detaliată a unui *IDS* pentru mediile de rețea, care combină tehnici de interceptare a pachetelor și de învățare automată. Sistemul utilizează un *firewall* primar pentru a intercepta și redirecționa traficul de Internet către *firewall*-urile secundare, responsabile de protejarea diferitelor părți ale rețelei.

Acest *script* [59] interacționează cu *firewall*-ul *pfSense* prin capturarea traficului de rețea de la *firewall*-ul principal (așa cum este specificat de parametrul *interface='eth0'* în funcția *pyshark.LiveCapture*). Figura 63 prezintă întregul proces propus. *Script*-ul filtrează pachetele capturate în funcție de adresa *IP* de destinație, determinând care pachete aparțin fiecărui *firewall* secundar. Odată filtrate, pachetele sunt redirecționate către *Snort*, care este implementat pe fiecare *firewall pfSense* pentru detectarea intruziunilor.

*Snort* generează alerte pentru orice intruziuni detectate, care sunt verificate de *script* și înregistrate într-un dicționar "intrusions". De asemenea, *script*-ul antrenează un model de învățare automată *Random Forest Classifier* pentru a detecta intruziunile utilizând setul de date "*intrusion_detection_data.csv*". Odată antrenat, *script*-ul utilizează modelul antrenat pentru a detecta orice intruziuni pe care *Snort* le-ar fi putut omite. În final, *script*-ul stabilește un prag pentru numărul de intruziuni detectate într-un interval de 60 de secunde. Dacă numărul de intruziuni detectate pentru un anumit *ID* de semnătură depășește acest prag, se generează o alertă pentru a informa utilizatorul că a fost detectată o intruziune.

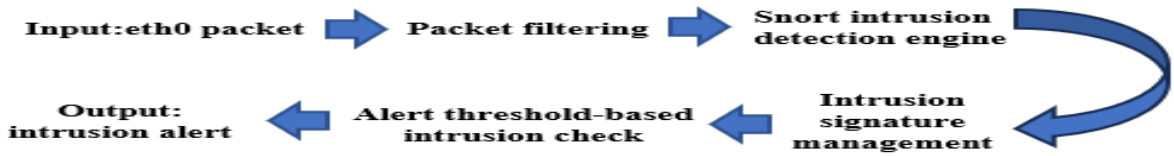



*Figura 63. Diagramă mod de operare script propus.*

Pentru a vizualiza scorul de acuratețe, se poate utiliza biblioteca *matplotlib* pentru a crea un grafic de bare sau un grafic liniar. Un exemplu este prezentat mai jos, în Figura 64:

```
    import matplotlib.pyplot as plt
    accuracy = [0.87, 0.91, 0.92, 0.89, 0.93]
    firewalls = ['FW2', 'FW3', 'FW4', 'FW5', 'Main FW']
    plt.bar(firewalls, accuracy)
    plt.title('Accuracy of IDS model on different firewalls')
    plt.xlabel('Firewall')
    plt.ylabel('Accuracy')
plt.show()
```

*Figura 64. Grafic acurateţe firewall-uri*

Evaluarea unui model de detectare a intruziunilor pe un set de date de testare furnizează valorile de acuratețe ale codului. Modelul a fost antrenat pe un set distinct de date de antrenament, folosind un algoritm de învățare supervizată, precum *Random Forest*, scopul antrenării fiind de a distinge între traficul normal și cel intruziv în rețea. Precizia este ilustrată în Figura 65. După antrenament, acuratețea modelului este evaluată în mod obișnuit pe un set distinct de date de testare, care constă din exemple etichetate și care nu au fost utilizate în timpul antrenamentului. Scorul de acuratețe cuantifică proporția de exemple clasificate corect din setul de date de testare. De exemplu, o acuratețe de 0,87 indică faptul că 87% din exemplele de test au fost clasificate corect de către model.

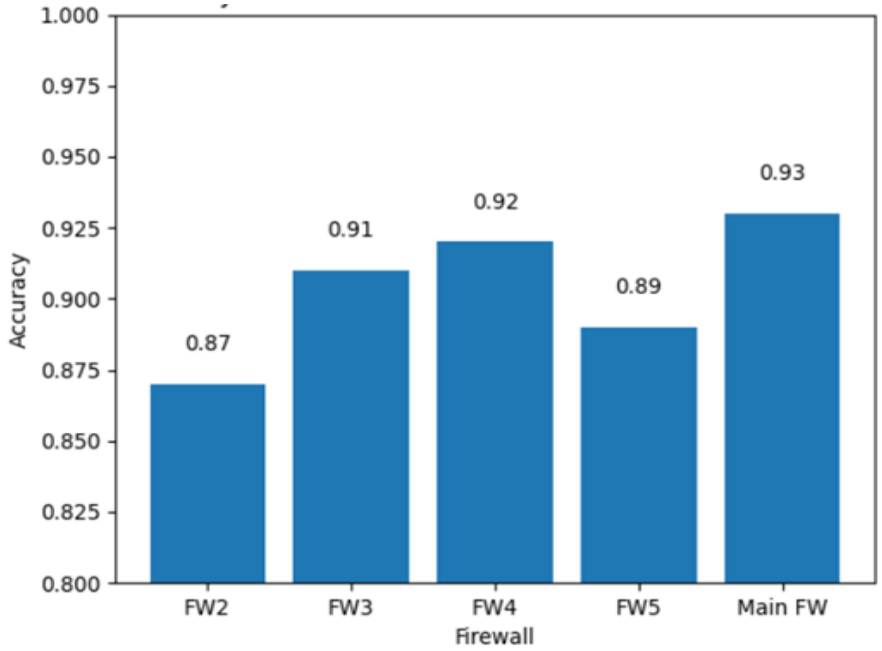



*Figura 65. Precizia modelului de detectare a intruziunilor pe diferite firewall-uri.*

Atacurile de tip *phishing* reprezintă o amenințare semnificativă la adresa securității rețelelor, iar tehnicile eficiente de detectare și combatere sunt vitale pentru protejarea informațiilor sensibile. Acest subcapitol prezintă un cadru nou de clasificare a *e-mail*-urilor pentru detectarea *phishing*-ului în medii de rețea virtualizate, cu accent deosebit pe arhitecturile de *firewall* distribuit. Cadrul utilizează puterea clasificatorului *Naive Bayes* [60] pentru a analiza textul preprocesat al *e-mail*-urilor și pentru a le clasifica în mod precis ca *phishing* sau legitime. Integrând acest cadru într-o configurație de *firewall* distribuit, traficul de *e-mail* poate fi controlat și filtrat dinamic la mai multe puncte de rețea, îmbunătățind în ansamblu postura de securitate. *Script*-ul demonstrează implementarea practică a cadrului și oferă un exemplu de utilizare într-o configurație de *firewall* distribuit.


*Figura 66. Diagrama modului de lucru - Detectare phishing.*

În Figura 66 este ilustrată diagrama fluxului de date al traficului de intrare de la Internet în serverul de *e-mail*. Mașina virtuală care rulează *script*-ul de clasificare a *e-mail*ului este plasată în aceeași segment de rețea ca și *router*-ul secundar (172.20.4.X). Plasând mașina virtuală în acest segment, aceasta poate intercepta și analiza eficient traficul de *e-mail* înainte ca acesta să ajungă la serverul de *e-mail* (172.21.4.10). În plus față de clasificarea *e-mail*-ului, *script*-ul propus generează reguli de *firewall* pe baza rezultatelor clasificării *e-mail*-ului. Când este primit un *e-mail* nou, acesta este preprocesat folosind aceleași proceduri ca și datele de antrenament. Clasificatorul prezice apoi clasificarea *e-mail*-ului, indicând dacă este "*phishing*" sau "legitim".

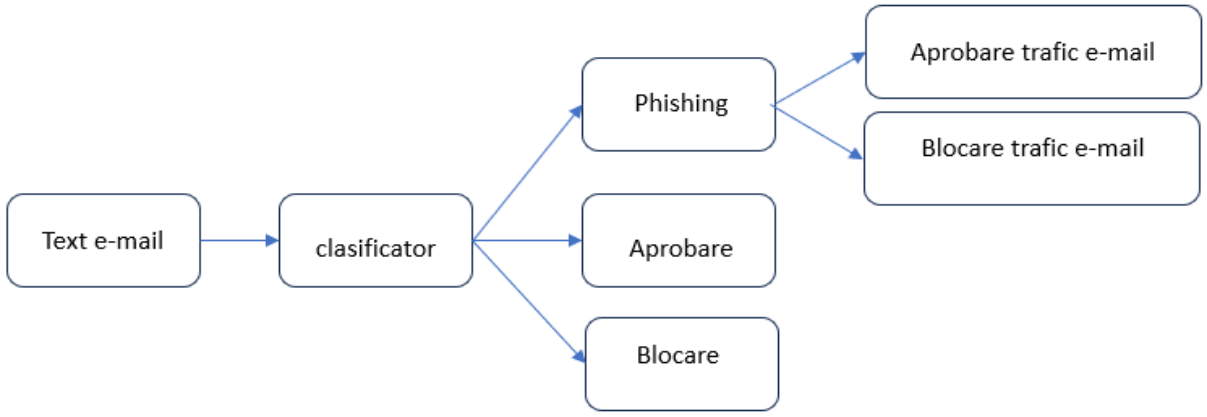



*Figura 67. Schema logică detectare phishing*

Modul de lucru este ilustrat în Figura 67. Rolul *script*-ului este acela de a detecta și de a răspunde la amenințările reprezentate de *e-mail*-uri de *phishing*. Prin preprocesarea și clasificarea automată a conținutului *e-mail*-urilor, *script*-ul poate identifica rapid și eficient *e-mail*-urile suspecte și poate genera automat reguli de *firewall* pentru a bloca traficul către potențiale surse de *phishing*. În esență, *script*-ul acționează ca un filtru adăugat în linia de apărare a sistemului, ajutând la prevenirea accesului neautorizat și la protejarea datelor și a infrastructurii împotriva amenințărilor cibernetice.

*Tabel 6. Măsurători de performanță ale cadrului de clasificare a e-mailurilor*

| **Metrică** | **Acuratețe** | **Precizie** | ***Recall*** | **Scor *F1*** |
|---|---|---|---|---|
| *Phishing E-mails* | 95.3% | 96.8% | 94.2% | 95.4% |
| *Legitimate E-mails* | 97.1% | 93.5% | 97.1% | 95.4% |

În Tabel 6 este furnizat un rezumat al indicatorilor de performanță, inclusiv acuratețe, precizie, amintire și scor F1, pentru *e-mail*-urile de tip *phishing* și legitime. Acești indicatori oferă informații cu privire la eficacitatea cadrului de clasificare a *e-mail*-ului în identificarea și clasificarea precisă a *e-mail*-urilor.

În continuare se oferă o analiză cuprinzătoare a unui script pentru monitorizarea și remedierea *Snort* în rețelele ce utilizează un *firewall* distribuit. Cadru oferă o soluție completă pentru monitorizarea continuă prin utilizarea *script*-ului *Python* și a bibliotecilor cheie, *subprocesses*, *requests* și *time*. *Script*-ul conține funcții pentru evaluarea conectivității rețelei, disponibilitatea

serverului de baze de date, limitările resurselor și configurația *Snort*. Prin efectuarea acestor teste în mod periodic, cadru identifică eficient problemele potențiale care ar putea compromite funcționarea optimă a lui *Snort*.

Monitorizând sistematic starea în care se află Snort, cadrul propus operează într-un mod metodic. Utilizând un *script* care coordonează o serie de proceduri, identifică proactiv potențiale probleme care pot apărea în timpul funcționării *Snort*. Prin executarea acestor proceduri ilustrate în Figura 68, cadrul își propune să reducă impactul problemelor legate de rețea, indisponibilitatea serverului de baze de date, constrângerile de resurse și inconsistențele de configurare asupra eficacității Snort.

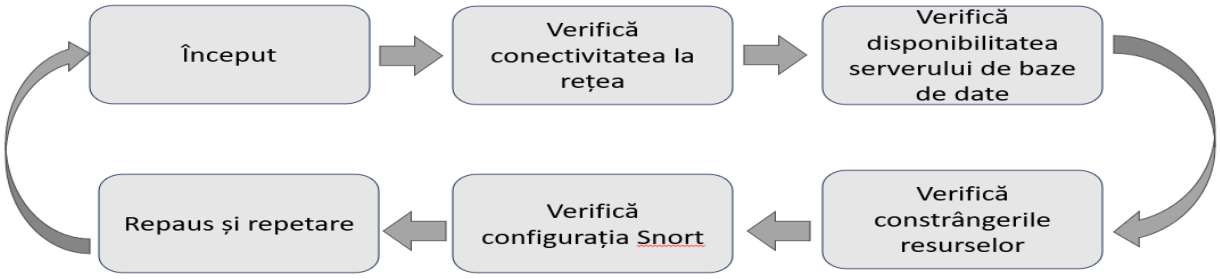



*Figura 68. Procesul de monitorizare Snort*

*Tabel 7. Conectivitatea rețelei*

| **Metrică** | **Rezultatele simulării** |
|---|---|
| Total cicluri monitorizare | 100 |
| Cicluri cu conectivitate funcțională | 99 |
| Cicluri cu probleme de conectivitate | 1 |
| Probleme detectate | Cycle 98 |
| | |
| Acțiuni de rezolvare | Acțiunea 1: Resetați adaptorului de rețea.<br>Acțiunea 2: Reporniți router-ul. |

Rezultatele simulărilor prezentate în Tabelul 7 arată că dintr-un total de o sută de cicluri de monitorizare, problemele de conectivitate la rețea au fost detectate doar o singură dată, și anume în Ciclul 98. Cadrul a răspuns rapid la această problemă prin executarea acțiunilor de rezoluție, cum ar fi resetarea adaptorului de rețea și repornirea *router*-ului.

*Tabel 8. Disponibilitatea serverului de baze de date*

| **Metrică** | Rezultatele simulării |
|---|---|
| Total cicluri monitorizare | 100 |
| Cicluri cu server accesibil | 99 |
| Cicluri cu probleme de conectivitate | 1 |
| Probleme detectate | Cycle 27 |
| Acțiuni de rezolvare întreprinse | Acțiunea 1: Verificați log-urile serverului bazei de date.<br>Acțiunea 2: Reporniți serverul bazei de date. |

Tabelul 8 dezvăluie că doar un ciclu, Ciclul 27, a avut probleme de conectivitate la serverul de baze de date. Verificarea înregistrărilor serverului de baze de date și restaurarea acestuia au fost printre acțiunile de inițiate de soluția propusă.

*Tabel 9. Constrângeri de resurse (CPU)*

| Metrică | Rezultatele simulării |
|---|---|
| Total cicluri de monitorizare | 100 |
| Utilizare medie a CPU | 65% |
| Cicluri care depășesc pragul | 2 |
| Cicluri de depășire a pragului | Cycle 57, Cycle 82 |
| Acțiuni de rezolvare întreprinse | Acțiunea 1: Optimizați regulile Snort.<br>Acțiunea 2: Măriți resursele sistemului. |

După cum se arată în Tabelul 9, constrângerile de resurse, în special utilizarea *CPU*-ului, au fost monitorizate. Utilizarea medie a *CPU*-ului în timpul simulării a fost 65%. Totuși, două cicluri, Ciclurile 57 și 82, au depășit pragul prestabilit. Pentru a aborda aceasta, cadrul a recomandat creșterea resurselor sistemului și optimizarea regulilor *Snort* ca acțiuni de rezoluție.

*Tabel 10. Configurarea Snort*

| Metrică | Rezultatele simulării |
|---|---|
| Total configurări greșite detectate | 3 |
| Configurări greșite detectate | Nealinierea interfețelor de rețea.<br>Definiții incompatibile de reguli.<br>Preprocesoare greșite. |
| Acțiuni de rezolvare întreprinse | Acțiunea 1: Realiniați interfețele de rețea.<br>Acțiunea 2: Corectați definițiile regulilor.<br>Acțiunea 3: Activați preprocesoarele necesare. |

Tabelul 10 subliniază detectarea a trei configurații incorecte *Snort*. Interfețele de rețea nealiniate, definițiile incompatibile ale regulilor și absența preprocesorilor pot fi exemple de astfel de cazuri. Prin realinierea interfețelor de rețea, corectarea definițiilor regulilor și activarea preprocesorilor necesari, cadrul a corectat cu succes aceste configurații incorecte.

Analiza datelor obținute în urma testării soluțiilor propuse reprezintă o etapă esențială în procesul de evaluare a eficacității și adecvării soluțiilor propuse într-un context specific. Această analiză are ca scop evaluarea rezultatelor obținute în cadrul testelor și extragerea de informații relevante pentru a lua decizii informate și a aduce îmbunătățiri soluțiilor.

În acest capitol s-a realizat analiza și implementarea diverselor soluții în domeniul securității informaționale, cu un accent deosebit pe *firewall*-urile distribuite, evidențiază eficacitatea și adaptabilitatea acestor instrumente în gestionarea amenințărilor cibernetice. Prin explorarea arhitecturii de rețea securizate bazate pe *firewall*-uri distribuite și aplicarea lor în decongestionarea serviciilor *FTP*, împreună cu dezvoltarea de reguli *Snort* pentru contracararea atacurilor de tip *flood SMTP*, s-a demonstrat capacitatea acestor soluții de a răspunde rapid și eficient provocărilor din mediul online. În plus, investigarea vulnerabilităților de securitate și implementarea unor sisteme automate de gestionare a regulilor dinamice reflectă angajamentul continuu în furnizarea unui nivel ridicat de protecție și adaptabilitate în cadrul infrastructurii IT. Astfel, *firewall*-urile distribuite reprezintă nu doar un element esențial în asigurarea securității rețelelor, ci și un instrument indispensabil pentru abordarea dinamică a amenințărilor cibernetice într-un mediu în continuă schimbare. Această diversitate de abordări și soluții reflectă un angajament ferm în dezvoltarea și implementarea unor mecanisme de securitate robuste și eficiente, menite să asigure protecția activelor și datelor organizației în fața amenințărilor din ce în ce mai sofisticate.

# Capitolul 6: Evaluarea performanțelor soluțiilor propuse

În vederea protejării rețelelor împotriva atacurilor cibernetice, se pot considera *firewall*-uri distribuite, fiind capabile să protejeze mai multe puncte de intrare în rețea. Configurația unei astfel de soluții impune evaluarea performanței *firewall*-ului distribuit și analiza funcționalităților cheie care asigură securitatea rețelei. Prin intermediul unei astfel de analize, se pot identifica punctele slabe în configurația *firewall*-ului distribuit și se pot dezvolta strategii pentru îmbunătățirea securității rețelei. Aspectele esențiale care pot fi examinate în cadrul unei astfel de analize includ performanța *firewall*-ului distribuit în condiții de trafic intens, configurarea politicilor de securitate și monitorizarea traficului de rețea. De asemenea, se poate evalua eficacitatea mecanismelor de securitate implementate, cum ar fi filtrele de adrese *IP*, inspecția traficului *SSL* sau detecția intruziunilor.

Metodologia de evaluare a performanțelor este un proces structurat și sistematizat pentru a măsura și evalua performanța unui sistem, unui proces sau a unei activități într-un mod obiectiv și cu precizie. Aceasta implică utilizarea unor metrici și indicatori de performanță relevanți, colectarea datelor și analiza rezultatelor pentru a determina eficiența și eficacitatea sistemului sau procesului evaluat. Metodologia folosită în cadrul acestei cercetări este formată din o serie de pași, după cum urmează: definirea obiectivelor, identificarea indicatorilor de performanță, colectarea datelor, analiza datelor, compararea rezultatelor, interpretarea rezultatelor împreună cu luarea deciziilor și monitorizarea continuă. Această metodologie se bazează pe seria de pași menționați anterior, care sunt ilustrați prin exemple concrete în această teză. În cazul unui sistem de securitate informațională, un obiectiv poate fi reducerea numărului de incidente de securitate cu un procentaj cât mai mare într-un an de implementare.

Analiza datelor obținute în urma testării soluțiilor propuse a implicat o serie de pași și tehnici pentru evaluarea performanței și adecvării acestor soluții în contextul specific al securității informaționale. Datele obținute din testele efectuate au fost organizate și pregătite pentru a permite o analiză riguroasă și obiectivă. Astfel, au fost identificate și calculate metrici relevante pentru a evalua performanța și eficacitatea soluțiilor propuse. De exemplu, pentru a măsura eficiența sistemului propus, s-au utilizat metrici precum timpul de răspuns al sistemului la atacuri, rata de detecție a intruziunilor și rata de alarme false. Astfel de metrici au oferit o imagine clară asupra performanței soluțiilor și au permis compararea rezultatelor obținute cu standardele predefinite.

Compararea rezultatelor obținute în urma testelor cu criteriile de acceptare a reprezentat un alt aspect esențial al analizei datelor. Astfel, rezultatele au fost comparate cu performanța sistemelor similare existente sau cu standardele și cerințele predefinite. Această comparație a permis să aflăm dacă soluțiile propuse îndeplinesc sau chiar depășesc așteptările și cerințele impuse de contextul specific al securității informaționale, datele fiind utilizate pentru diferite participări la conferințe internaționale. Ca ultim pas, rezultatele și concluziile obținute în urma analizei datelor au fost documentate și comunicate către părțile interesate. Aceasta a implicat crearea de rapoarte și prezentări care să ofere o imagine clară și detaliată asupra performanței soluțiilor propuse. De asemenea, au fost evidențiate avantajele și beneficiile soluțiilor identificate, ceea ce a contribuit la luarea deciziilor informate și la adoptarea măsurilor corespunzătoare pentru îmbunătățirea securității informaționale într-un context specific.

# Capitolul 7: Concluzii generale, contribuții și perspective

În cadrul acestei teze s-au abordat o serie de aspecte și probleme relevante în domeniul securității informaționale și s-au adus mai multe contribuții care pot îmbunătăți înțelegerea și practicile actuale în acest domeniu. Prin cercetarea și analiza efectuate, s-au obținut rezultate valoroase și s-au identificat soluții progresive pentru a face față amenințărilor și atacurilor cibernetice. Mai jos sunt prezentate principalele contribuții aduse în acest domeniu.

C1. S-a configurat, implementat și optimizat o arhitectură de *firewall* distribuit, ca soluție de securitate alternativă la variantele *firewall* tradiționale. Arhitectura propusă are la bază instrumente de securitate *firewall* constând într-un *router* principal și patru *router*e secundare, fiecare configurat cu un *firewall* adaptat contextului specific. Prin această structură, s-a realizat o distribuție eficientă a sarcinilor și o redirecționare optimizată a traficului către *router*-ele secundare. Fiecare *router* secundar este echipat cu un *firewall* configurat individual, iar prin implementarea unui sistem de detectare și prevenire a intruziunilor, traficul este supus unei filtrări meticuloase, în conformitate cu politicile de securitate predefinite. Prin acest proces, fiecare pachet de date este evaluat, iar o decizie este luată cu privire la permisibilitatea, respingerea sau blocarea sa, în concordanță cu regulile și politicile de securitate specificate în cadrul *firewall*-ului. Prin distribuirea sarcinilor între *router*-ele secundare și prin aplicarea riguroasă a politicilor de filtrare, soluția propusă permite un control granular, oferind de asemenea o protecție optimă împotriva amenințărilor cibernetice [38].

C2. S-a analizat impactul congestiei de rețea generată de atacuri cibernetice, punând accent asupra dezvoltării unei soluții în cadrul serviciilor *FTP*. În timpul analizei atent detaliate a traficului produs în timpul unui atac de tip *UDP flood* s-au evidențiat consecințele negative asupra resurselor de rețea. S-au detaliat procesele de implementare și configurare a unei reguli personalizate de *firewall* în sistemul existent. S-a propus o regulă personalizată pentru gestionarea și prevenirea acestor atacuri, observând îmbunătățiri semnificative în eficiența utilizării resurselor de rețea. Prin prezentarea detaliilor implementării și prin demonstrarea soluției, s-a subliniat importanța consolidării securității în cadrul protocoalelor *FTP*. Astfel, contribuind la o perspectivă nouă și la soluții practice pentru abordarea congestiei de rețea în contextul critic al serviciilor esențiale, în special în domeniul medical [43].

C3. S-a realizat un studiu cu scopul de a identifica și contracara atacurilor de tip *e-mail bomb* asupra unui server ce folosește protocolul *SMTP.* În cadrul acestui demers, s-a propus o soluție de introducere a unei noi reguli în cadrul sistemului de detectare și prevenire a intruziunilor *Snort*, cu accent pe detectarea și prevenirea acestui tip specific de atac. Evaluarea performanței regulii în detectarea atacurilor de *e-mail bomb*, împreună cu prezentarea datelor și analiza rezultatelor obținute în cadrul testelor și experimentelor, subliniază eficacitatea soluției propuse. S-a explorat, de asemenea, modul în care regula *Snort* poate genera alerte în cazul detectării unor conexiuni *TCP* suspecte asociate cu astfel de atacuri, evidențiind rolul esențial al alertelor în semnalarea și răspunsul rapid la potențiale amenințări. În ansamblu, contribuția demersului propune o soluție specifică și eficientă pentru abordarea amenințărilor de securitate reprezentate de atacurile de tip *e-mail bomb* asupra unui server ce folosește protocolului *SMTP* [47].

C4. S-a efectuat un audit de securitate ce evidențiază analiza detaliată efectuată asupra performanței unui *firewall* distribuit, cu accent pe soluționarea problemelor identificate într-un *firewall* complex, utilizat într-o configurație distribuită. Prin aplicarea metodelor de testare manuale și automate, s-a avut ca obiectiv identificarea și remedierea vulnerabilităților sistemului. S-au adoptat abordări diferite pentru detectarea și rezolvarea acestor probleme minore, prezentând rezultatele studiului sub forma unui raport de risc. Scopul principal al studiului a constat în a furniza un raport de securitate adecvat unui *firewall* distribuit, contribuind astfel la îmbunătățirea integrității și eficienței acestui tip de sistem [61].

C5. S-a realizat și implementat un sistem automatizat de reguli dinamice pentru *firewall*-uri distribuite, cu un accent deosebit pe gestionarea eficientă a vulnerabilităților de tip *zero-day.* S-a propus automatizarea generării și actualizării regulilor *firewall*-ului, utilizând un script *Python* care rulează pe o mașină virtuală conectată la un *firewall*. Sistemul autonom dezvoltat, colectează și analizează jurnalele de securitate din rețea, ulterior, pe baza acestora dezvoltă și aplică reguli *firewall* pentru a bloca traficul suspicios produs de un simulator de trafic de rețea [55].

C6. S-a conceput o alternativă de filtrare și analizare a traficului în cadrul unui sistem de *firewall* distribuit, configurat cu *Snort.* Acest sistem integrează tehnici de *machine learning*, utilizând clasificatorul *Random Forest.* Sistemul propus funcționează în mod continuu, monitorizează intruziunile detectate și emite alerte în timp real atunci când sunt depășite limitele predefinite. Pentru evaluarea performanței sistemului, s-a efectuat calculul acurateței utilizând un set de date cuprinzător. S-au atins rate înalte de acuratețe, minimizând simultan alarmele false. S-a urmărit în special detectarea intruziunilor, prevenirea răspândirii acestora și reducerea traficului de rețea inutil [59].

C7. S-a implementat o platformă de clasificare a *e-mail*-urilor, cu accent pe detectarea *phishing*-ului în mediul de rețea virtualizat. Platforma utilizează clasificatorul *Naive Bayes* pentru a analiza textul preprocesat al *e-mail*-urilor și a le clasifica cu precizie drept *phishing* sau legitime. Integrarea acestei platforme într-o configurație de *firewall* distribuit a permis controlul și filtrarea dinamică a traficul *de e-mail* la mai multe puncte ale rețelei, îmbunătățind astfel securitatea generală. Se utilizează *Snort* pentru a detecta potențialele intruziuni pe baza pachetelor filtrate, orice alertă generată de acesta fiind înregistrată și analizată. Se folosește un set de date preexistent pentru a instrui un model *Random Forest* destinat detecției de intruziuni. Acuratețea modelului este evaluată pe un set de testare [62].

C8. S-a dezvoltat un sistem de management al algoritmilor de *optimizare* pentru *firewall*-ul distribuit. Sistemul propus pentru managementul *firewall*-ului colectează date despre utilizarea *CPU*-ului și lățimea de bandă a interfeței *WAN* pentru fiecare mașină virtuală, aplicând ulterior un algoritm de optimizare. Prin utilizarea interfeței de programare a aplicației (*API*), s-a creat un instrument ce accesează datele din platforma de virtualizare și ajustează procesul de optimizare al *firewall*-ului. Astfel, soluția devine capabilă să monitorizeze întreaga rețea și să ia decizii bazate pe parametrii colectați [57].

C9. S-a implementat un cadru bazat pe *script*-uri pentru monitorizarea și remedierea instrumentului *Snort* în mediul propus, punând la dispoziție funcții specializate pentru evaluarea

conectivității rețelei, verificarea disponibilității serverului de tip baze de date, analiza limitărilor de resurse și verificarea configurării acestuia. Efectuând periodic aceste teste, s-au identificat problemele potențiale ce ar putea afecta funcționarea optimă a instrumentului *Snort* în mediul de *firewall* distribuit. Contribuția adusă în această lucrare se concentrează pe mediile de *firewall* distribuit și adoptând o metodologie bazată pe *script*-uri [63].

C10. S-a explorat și dezvoltat o alternativă la configurarea autentificării utilizatorilor pe platforma *pfSense,* în cazul în care există mai multe instanțe. Prin intermediul *script*-ului personalizat și integrarea cu *LDAP*, s-a propus o soluție pentru automatizarea procesului de autentificare a utilizatorilor. Metoda, a permis configurarea automată a autentificării *LDAP*, aducând beneficii semnificative, precum reducerea timpului de configurare, consolidarea consistenței și centralizarea gestionării utilizatorilor în cadrul rețelei distribuite. Studiul analizează modul în care această tehnologie poate îmbunătăți securitatea sistemelor de *firewall* distribuit, reprezentând astfel o contribuție pentru administratorii de sisteme, specialiștii în securitatea rețelei și cercetătorii din domeniul securității informațiilor [64].

Dezvoltarea conceptului de *firewall* distribuit reprezintă un progres, furnizând o perspectivă nouă asupra securității rețelelor și oferind o soluție scalabilă și dinamică adaptată cerințelor complexe ale mediului de rețea contemporan.

Astfel, soluțiile avansate de detecție și prevenire a atacurilor, integrate cu tehnologii emergente precum inteligența artificială și învățarea automată, demonstrează o abordare holistică și proactivă în combaterea amenințărilor cibernetice în evoluție. Implementarea sistemelor automate de gestionare a regulilor și configurărilor reprezintă un alt progres important, facilitând o administrare eficientă și coezivă a securității în contextul unor rețele distribuite din ce în ce mai complexe.

Evaluarea riguroasă a performanțelor soluțiilor propuse confirmă nu doar eficacitatea acestora, ci și reziliența lor în fața provocărilor critice, oferind astfel nu doar o evoluție, ci și un fundament robust pentru viitorul securității cibernetice. Aceste concluzii evidențiază nu doar importanța cercetării tehnice în domeniul securității informaționale, ci și nevoia continuă de inovare și adaptare pentru a face față amenințărilor în schimbare din peisajul cibernetic contemporan.

În cadrul cercetării s-a dezvoltat și propus o abordare nouă pentru mecanismul de *firewall* distribuit care îmbunătățește semnificativ securitatea rețelelor. Acest concept oferă o abordare mai eficientă și scalabilă în protejarea rețelelor împotriva atacurilor cibernetice prin distribuirea funcționalității *firewall*-ului pe mai multe noduri în rețea. Prin implementarea acestui concept de *firewall* distribuit, se obține o securitate sporită și o rezistență mai mare la atacuri.

Distribuirea funcționalității *firewall*-ului pe mai multe noduri permite o gestionare mai eficientă a traficului de rețea și o filtrare mai precisă a acestuia. Fiecare nod în rețea poate fi configurat individual pentru a analiza și controla traficul care trece prin el, oferind astfel o protecție mai robustă la nivelul fiecărui punct de acces în rețea. Prin utilizarea unui *firewall* distribuit, se elimină vulnerabilitatea asociată cu punctele unice de eșec și se creează o arhitectură rezistentă la atacuri. Dacă un nod al *firewall*-ului este compromis sau inoperabil, celelalte noduri din rețea pot prelua rapid funcționalitatea și să continue să protejeze rețeaua.

# Bibliografie


[1] A. Andalib and S. M. Babamir, “Anomaly detection of policies in distributed firewalls using data log analysis,” *J Supercomput*, vol. 79, no. 17, pp. 19473–19514, Nov. 2023, doi: 10.1007/s11227-023-05417-7.

[2] B. Rajkumar and G. Arunakranthi, “Evolution for a secured path using NexGen firewalls,” in *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)*, IEEE, Feb. 2023, pp. 1–6. doi: 10.1109/OTCON56053.2023.10113935.

[3] R. Kaur, D. Gabrijelčič, and T. Klobučar, “Artificial intelligence for cybersecurity: Literature review and future research directions,” *Information Fusion*, vol. 97, p. 101804, Sep. 2023, doi: 10.1016/j.inffus.2023.101804.

[4] M. J. Hossain Faruk, S. Tahora, M. Tasnim, H. Shahriar, and N. Sakib, “A Review of Quantum Cybersecurity: Threats, Risks and Opportunities,” in *2022 1st International Conference on AI in Cybersecurity (ICAIC)*, IEEE, May 2022, pp. 1–8. doi: 10.1109/ICAIC53980.2022.9896970.

[5] L. A. Maghrabi, “Automated Network Intrusion Detection for Internet of Things Security Enhancements,” *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3369237.

[6] J. Arsenyan and A. Piepenbrink, “Artificial Intelligence Research in Management: A Computational Literature Review,” *IEEE Trans Eng Manag*, vol. 71, pp. 5088–5100, 2024, doi: 10.1109/TEM.2022.3229821.

[7] N. Humaidi and S. H. Abdallah Alghazo, “Procedural Information Security Countermeasure Awareness and Cybersecurity Protection Motivation in Enhancing Employee’s Cybersecurity Protective Behaviour,” in *2022 10th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, Jun. 2022, pp. 1–10. doi: 10.1109/ISDFS55398.2022.9800834.

[8] Hewlett-Packard, “Wolf Security Threat Insights,” Wolf Security Threat Insights office. Accessed: Feb. 15, 2024. [Online]. Available: https://www.hp.com/us-en/services/workforce-solutions/workforce-security-solution.html

[9] Directoratul Naţional de Securitate Cibernetică, “UPDATE: Un atac cibernetic de tip ransomware a afectat spitale din România.” Accessed: Feb. 18, 2024. [Online]. Available: https://www.dnsc.ro/citeste/atac-cibernetic-ransomware-spitale-Romania

[10] L. Meshkat, R. L. Miller, C. Hillsgrove, and J. King, “Behavior Modeling for Cybersecurity,” in *2020 Annual Reliability and Maintainability Symposium (RAMS)*, IEEE, Jan. 2020, pp. 1–7. doi: 10.1109/RAMS48030.2020.9153685.

[11] S. Merugula, K. S. Kumar, S. Muppidi, and Ch. Vidyadhari, “Stop Phishing : Master Anti-Phishing Techniques,” in *2022 IEEE North Karnataka Subsection Flagship International*

*Conference (NKCon)*, IEEE, Nov. 2022, pp. 1–5. doi: 10.1109/NKCon56289.2022.10126569.

[12] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," *IEEE Access*, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[13] S.-J. Lee, H.-Y. Shim, Y.-R. Lee, T.-R. Park, S.-H. Park, and I.-G. Lee, "Study on Systematic Ransomware Detection Techniques," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, IEEE, Feb. 2022, pp. 297–301. doi: 10.23919/ICACT53585.2022.9728909.

[14] S. Vasudevan, A. Piazza, and M. Carr, "A Decade of Studies on Cyber Security Training in Organizations using Social Network Analysis: A Systematic Literature Review Through Keyword co-Occurrence Network," in *2023 International Conference on Business Analytics for Technology and Security (ICBATS)*, IEEE, Mar. 2023, pp. 1–6. doi: 10.1109/ICBATS57792.2023.10111127.

[15] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2023, doi: 10.1109/COMST.2023.3280465.

[16] M. Parihar and C. Fung, "IDS with deep learning techniques," in *2023 7th Cyber Security in Networking Conference (CSNet)*, IEEE, Oct. 2023, pp. 1–4. doi: 10.1109/CSNet59123.2023.10339748.

[17] P. Shrivastava and R. K. Yadav, "Cyber-Attacks Detection Using Intelligent Intrusion System (IDS) Along With Deep Learning: Novel Approach," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, Jul. 2023, pp. 1–7. doi: 10.1109/ICCCNT56998.2023.10306742.

[18] J. Liang and Y. Kim, "Evolution of Firewalls: Toward Securer Network Using Next Generation Firewall," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, Jan. 2022, pp. 0752–0759. doi: 10.1109/CCWC54503.2022.9720435.

[19] V. Clincy and H. Shahriar, "Web Application Firewall: Network Security Models and Configuration," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, Jul. 2018, pp. 835–836. doi: 10.1109/COMPSAC.2018.00144.

[20] A. A. El Tawil and K. Samrouth, "IEWS: a Free Open Source Intelligent Early Warning System Based on Machine Learning," in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, May 2023, pp. 1–3. doi: 10.1109/ISDFS58141.2023.10131709.

[21] T. N. Nguyen and T. T. T. Le, “Authentication and Encryption algorithms for data security in Cloud computing: A comprehensive review,” Jan. 2022, pp. 57–63. doi: 10.15439/2021R7.

[22] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, “Recent Advances of Resource Allocation in Network Function Virtualization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 295–314, Feb. 2021, doi: 10.1109/TPDS.2020.3017001.

[23] O. Sri Nagesh, T. Kumar, and V. R. Vedula, “A Survey on Security Aspects of Server Virtualization in Cloud Computing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 3, p. 1326, Jun. 2017, doi: 10.11591/ijece.v7i3.pp1326-1336.

[24] S. A. Babu, M. J. Hareesh, J. P. Martin, S. Cherian, and Y. Sastri, “System Performance Evaluation of Para Virtualization, Container Virtualization, and Full Virtualization Using Xen, OpenVZ, and XenServer,” in *2014 Fourth International Conference on Advances in Computing and Communications*, IEEE, Aug. 2014, pp. 247–250. doi: 10.1109/ICACC.2014.66.

[25] N. Jain and S. Choudhary, “Overview of virtualization in cloud computing,” in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, IEEE, Mar. 2016, pp. 1–4. doi: 10.1109/CDAN.2016.7570950.

[26] F. Sierra-Arriaga, R. Branco, and B. Lee, “Security Issues and Challenges for Virtualization Technologies,” *ACM Comput Surv*, vol. 53, no. 2, pp. 1–37, Mar. 2021, doi: 10.1145/3382190.

[27] D. C. D’Elia, S. Nicchi, M. Mariani, M. Marini, and F. Palmaro, “Designing Robust API Monitoring Solutions,” *IEEE Trans Dependable Secure Comput*, pp. 1–1, 2021, doi: 10.1109/TDSC.2021.3133729.

[28] S. Baucke, J. Kempf, R. Ben Ali, A. Ramachandran, and S. Seetharaman, “Cloud API support for self-service Virtual Network Function (VNF) deployment,” in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, IEEE, Nov. 2015, pp. 40–46. doi: 10.1109/NFV-SDN.2015.7387404.

[29] S. S. Kolahi, V. S. Hora, A. P. Singh, S. Bhatti, and S. R. Yeeda, “Performance Comparison of Cloud Computing/IoT Virtualization Software, Hyper-V vs vSphere,” in *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, IEEE, Feb. 2020, pp. 1–6. doi: 10.1109/ASET48392.2020.9118185.

[30] Citrix, “Citrix Hypervisor.” [Online]. Available: https://developer.cloud.com/citrixworkspace/citrix-hypervisor/docs/overview

[31] Z. Xu, L. Yang, and J. Lei, “Conception and Design of Desktop Virtualization Cloud Platform for Primary Education: Based on the Citrix Technology,” in *2015 International*

*Conference of Educational Innovation through Technology (EITT)*, IEEE, Oct. 2015, pp. 226–230. doi: 10.1109/EITT.2015.55.

[32] N. Saswade, V. Bharadi, and Y. Zanzane, "Virtual Machine Monitoring in Cloud Computing," *Procedia Comput Sci*, vol. 79, pp. 135–142, 2016, doi: 10.1016/j.procs.2016.03.018.

[33] Iris Graessler, "A new V-Model for interdisciplinary product engineering," *Ilmenau Scientific Colloquium, Technische Universität Ilmenau*, 2017.

[34] A. Makhdoomi, N. Jan, Palak, and N. Goel, "Conventional and next generation firewalls in network security and its applications," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, IEEE, Nov. 2022, pp. 964–969. doi: 10.1109/ICCCIS56430.2022.10037674.

[35] D. A. B. Fernandes, M. Neto, L. F. B. Soares, M. M. Freire, and P. R. M. Inácio, "On the self-similarity of traffic generated by network traffic simulators," in *Modeling and Simulation of Computer Networks and Systems*, Elsevier, 2015, pp. 285–311. doi: 10.1016/B978-0-12-800887-4.00010-9.

[36] Andrei-Daniel Tudosi, "Contributions to the improvement of information security systems at various levels of communication," in *Physical And Technological Problems Of Transmission, Processing And Storage Of Information In Infocommunication Systems*, Chernivtsi - Ukraine: Chernivtsi National University, Oct. 2021, pp. 45–47.

[37] L. Bouali, E. Abd-Elrahman, H. Afifi, S. Bouzefrane, and M. Daoui, "Virtualization Techniques: Challenges and Opportunities," 2016, pp. 49–62. doi: 10.1007/978-3-319-50463-6_5.

[38] A.-D. Tudosi, D. G. Balan, and A. D. Potorac, "Secure network architecture based on distributed firewalls," in *2022 International Conference on Development and Application Systems (DAS)*, IEEE, May 2022, pp. 85–90. doi: 10.1109/DAS54948.2022.9786092.

[39] C.-M. Petruti, B.-A. Puiu, I.-A. Ivanciu, and V. Dobrota, "Automatic Management Solution in Cloud Using NtopNG and Zabbix," in *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, IEEE, Sep. 2018, pp. 1–6. doi: 10.1109/ROEDUNET.2018.8514142.

[40] A. R. H. Velasco, E. E. G. Malla, R. D. C. C. Herrera, and F. D. M. Arévalo, "Real-time monitoring and alerting system using Zabbix and Grafana software for wireless Internet access service management.," in *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2023, pp. 1–6. doi: 10.23919/CISTI58278.2023.10211432.

[41] E. Safrianti, L. O. Sari, and N. A. Sari, "Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model," in *2021 3rd International Conference on Research and Academic Community Services (ICRACOS)*, IEEE, Oct. 2021, pp. 122–127. doi: 10.1109/ICRACOS53680.2021.9701973.

[42] R. R. Nuiaa, S. Manickam, and A. H. Alsaeedi, “Distributed reflection denial of service attack: A critical review,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, p. 5327, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5327-5341.

[43] A.-D. Tudosi, A. Graur, D. G. Balan, and A. Dan Potorac, “Network Congestion Solution for FTP Services Based on Distributed Firewall and Snort,” in *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*, IEEE, Sep. 2022, pp. 1–4. doi: 10.1109/RoEduNet57163.2022.9921099.

[44] Netgate, “pfSense® - World’s Most Trusted Open Source Firewall.” [Online]. Available: https://www.pfsense.org/

[45] İ. Gündoüdu, A. A. Selçuk, and S. özarslan, “Effectiveness Analysis of Public Rule Sets Used in Snort Intrusion Detection System,” in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, IEEE, Jun. 2021, pp. 1–4. doi: 10.1109/SIU53274.2021.9477698.

[46] H. Song, D. Ding, Q.-L. Han, and X. Ge, “Trust-Based Distributed Entropy Filtering for State-Saturated Nonlinear Systems with Hybrid Cyber-Attacks and Non-Gaussian Noises,” in *2024 Australian & New Zealand Control Conference (ANZCC)*, IEEE, Feb. 2024, pp. 164–168. doi: 10.1109/ANZCC59813.2024.10432849.

[47] A.-D. Tudosi, D. G. Balan, and A. Dan Potorac, “New Snort rule for detection and prevention of SMTP e-mail bomb attacks,” in *2022 International Conference on Development and Application Systems (DAS)*, IEEE, May 2022, pp. 78–84. doi: 10.1109/DAS54948.2022.9786213.

[48] Linux Foundation, “API - XCP-ng documentation.” Accessed: Jan. 11, 2023. [Online]. Available: https://xcp-ng.org/docs/api.html

[49] P. Marchetta, A. Montieri, V. Persico, A. Pescape, I. Cunha, and E. Katz-Bassett, “How and how much traceroute confuses our understanding of network paths,” in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, IEEE, Jun. 2016, pp. 1–7. doi: 10.1109/LANMAN.2016.7548847.

[50] I. Kostaras, C. Drabo, J. Juneau, S. Reimers, M. Schröder, and G. Wielenga, “The NetCAT Program on Testing,” in *Pro Apache NetBeans*, Berkeley, CA: Apress, 2020, pp. 431–440. doi: 10.1007/978-1-4842-5370-0_14.

[51] Kali Linux, “Tool Documentation: firewalk Usage Example.” Accessed: Jul. 01, 2023. [Online]. Available: https://www.kali.org/tools/firewalk/

[52] J. Daly, A. X. Liu, and E. Torng, “A Difference Resolution Approach to Compressing Access Control Lists,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 610–623, Feb. 2016, doi: 10.1109/TNET.2015.2397393.

[53] U.S. Department of Homeland Security (DHS), “CVE.” Accessed: Dec. 12, 2022. [Online]. Available: https://www.cve.org/ResourcesSupport/FAQs

[54] Pratum, “Risk Assessment: Likelihood & Impact.” [Online]. Available: https://pratum.com/blog/443-risk-assessment-likelihood-impact

[55] A.-D. , Tudosi, A. Graur, D. G. Balan, A. D. Potorac, and R.-C. Tarabuta, “Design and Implementation of an Automated Dynamic Rule System for Distributed Firewalls,” *Advances In Electrical And Computer Engineering*, vol. 23, no. 3, pp. 29–38, Aug. 2023.

[56] P. Čisar, B. Popović, K. Kuk, S. M. Čisar, and I. Vuković, “Machine Learning Aspects of Internet Firewall Data,” 2022, pp. 43–59. doi: 10.1007/978-94-024-2174-3_4.

[57] A.-D. Tudosi, A. Graur, D. G. Balan, and A. D. Potorac, “Design and Implementation of a Distributed Firewall Management System for Improved Security,” in *2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, IEEE, Sep. 2023, pp. 1–6. doi: 10.1109/RoEduNet60162.2023.10274920.

[58] M. A. Thakur and R. Gaikwad, “User identity & lifecycle management using LDAP directory server on distributed network,” in *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, Jan. 2015, pp. 1–3. doi: 10.1109/PERVASIVE.2015.7086970.

[59] A.-D. Tudosi, A. Graur, D. G. Balan, A. Dan Potorac, and R. Tarabuta, “Distributed Firewall Traffic Filtering and Intrusion Detection Using Snort on pfSense Firewalls with Random Forest Classification,” in *2023 46th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, Jul. 2023, pp. 101–104. doi: 10.1109/TSP59544.2023.10197784.

[60] I. Wickramasinghe and H. Kalutarage, “Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation,” *Soft comput*, vol. 25, no. 3, pp. 2277–2293, Feb. 2021, doi: 10.1007/s00500-020-05297-6.

[61] A.-D. Tudosi, A. Graur, D. G. Balan, and A. D. Potorac, “Research on Security Weakness Using Penetration Testing in a Distributed Firewall,” *Sensors*, vol. 23, no. 5, p. 2683, Mar. 2023, doi: 10.3390/s23052683.

[62] A.-D. Tudosi, A. Graur, D. G. Balan, and A. D. Potorac, “An Email Classification Framework for Phishing Detection in Virtualized Network Environments,” in *2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, IEEE, Sep. 2023, pp. 1–5. doi: 10.1109/RoEduNet60162.2023.10274915.

[63] Andrei-Daniel Tudosi, “A Python-based Approach for Monitoring and Troubleshooting Snort IDS in Distributed Firewall Environments,” in *14th International Conference on Advanced Scientific Computing - ICASC 2023*, Cluj-Napoca, Oct. 2023, pp. 1–5. doi: 10.1109/ICASC58845.2023.10328028.

[64] Andrei-Daniel Tudosi, Adrian Graur, Doru Balan, and Alin Potorac, “Automatic Directory Service Integration in Distributed Firewall Resources: A Study of Scripting and LDAP Integration with pfSense,” in *International Conference on e-Health and Bioengineering - EHB 2023 - 11-th Edition*, Bucharest, Nov. 2023.