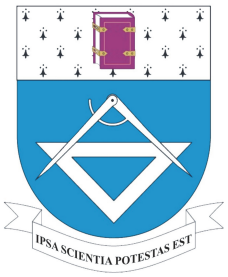


UNIVERSITATEA TEHNICĂ
”GHEORGHE ASACHI”
DIN IAŞI

CONTRIBUȚII PRIVIND SECURITATEA CIBERNETICĂ ÎN CLOUD COMPUTING

Florea Răzvan
Conducător de doctorat: Prof. Univ. Dr. Inf. Mitică Craus

Iaşi, 2025

CONTRIBUȚII PRIVIND SECURITATEA CIBERNETICĂ ÎN CLOUD COMPUTING

Florea Răzvan
Calculatoare și Tehnologia Informației

Conducător de doctorat: **Prof. Univ. Dr. Inf. Mitică Craus**

Abstract

Software Defined Networking (SDN) reprezintă o abordare revoluționară a managementului rețelelor, care separă procesul de management de fluxurile de transmitere în arhitectura rețelei. Spre deosebire de rețelele tradiționale, în care fiecare dispozitiv de rețea ia decizii independent cu privire la traficul de date, SDN centralizează controlul, permițând administratorilor de rețea să gestioneze serviciile de rețea printr-un controler centralizat. Prin acest controler se obține o viziune globală a rețelei și poate ajusta dinamic fluxurile de trafic, optimiza performanța și îmbunătăți implementarea aplicațiilor și serviciilor. Programabilitatea SDN permite gestionarea automatizată și eficientă a rețelei și capacitatea de a se adapta rapid la condițiile și cerințele în schimbare ale rețelei.

În domeniul securității cibernetice, SDN prezintă atât oportunități, cât și provocări. Pe de o parte, controlul centralizat și programabilitatea SDN pot îmbunătăți semnificativ securitatea rețelei. Administratorii pot implementa politici de securitate cuprinzătoare în întreaga rețea, pot răspunde rapid la amenințări și pot izola segmentele compromise, reducând răspândirea activităților malițioase. Natura dinamică a SDN permite detectarea și atenuarea amenințărilor în timp real, îmbunătățind reziliența generală a rețelei. Pe de altă parte, centralizarea controlului introduce un punct unic de eșec și o țintă atrăgătoare pentru atacatorii cibernetici. Compromiterea controlerului SDN ar putea permite unui atacator să preia controlul întregii rețele, necesitând măsuri de securitate robuste pentru a proteja controlerul și pentru a asigura integritatea și disponibilitatea infrastructurii SDN.

În acest studiu, prezentăm o examinare detaliată a implementării cadrului de Security Enhanced Cloud Software Defined Networking, în conjuncție cu Snort, un sistem open-source de detectare a intruziunilor, pentru a întări mediile cloud împotriva atacurilor de tip Distributed Denial of Service (DDoS). SEC-SDN îmbunătățește arhitectura SDN tradițională cu caracteristici avansate de securitate, facilitând managementul dinamic, în timp real, al traficului și atenuarea amenințărilor.

Integrarea Snort cu SEC-SDN conferă sistemului capacități avansate de monitorizare, permițând detectarea tiparelor de trafic malițios și aplicarea imediată a politicilor

de securitate pentru interceptarea și neutralizarea amenințărilor. Această abordare dublă nu doar atenuează impactul atacurilor DDoS, ci și menține integritatea și disponibilitatea serviciilor cloud. Studiul evaluează eficacitatea acestui cadru de securitate integrat printr-o serie de scenarii simulate de atacuri DDoS într-o rețea bazată pe cloud. Rezultatele demonstrează o îmbunătățire substanțială a timpilor de detectare și răspuns la amenințări, reducând semnificativ susceptibilitatea mediului cloud la incidente DDoS.

Descoperirile subliniază potențialul combinării SEC-SDN cu sisteme de detectare a intruziunilor, cum ar fi Snort, pentru a stabili o poziție de securitate rezilientă și adaptivă, asigurând disponibilitatea și fiabilitatea continuă a serviciilor cloud în contextul amenințărilor cibernetice în creștere. Acest studiu aduce o contribuție semnificativă la discursul privind securitatea cloud, propunând o soluție scalabilă și eficientă pentru una dintre cele mai urgente provocări din securitatea rețelelor contemporane.

Termeni cheie: *Software Defined Network, Security Enhanced Cloud - Software Defined Networking, securitate cloud, mitigarea amenințărilor în timp real, teoria jocurilor, atacuri DDoS*.

CUPRINS

CAPITOLUL 1

Introducere

1.1 Motivație

Scrierea unei lucrări despre securitatea cibernetică în sistemele cloud utilizând cadrul SEC-SDN și aplicarea teoriei jocurilor reprezintă pentru mine o călătorie profund personală și extrem de importantă. Aceasta nu este doar o abordare academică, ci o misiune pe care o consider esențială pentru viitorul lumii noastre digitale. Pe măsură ce devenim din ce în ce mai dependenți de cloud computing pentru stocarea datelor personale și funcționarea infrastructurii critice, securitatea acestor sisteme devine nu doar o provocare tehnică, ci și o necesitate socială.

Cloud-ul oferă beneficii incredibile, precum scalabilitatea, eficiența și accesibilitatea, dar introduce și vulnerabilități semnificative. Aceste vulnerabilități sunt adesea complexe, dinamice și dificil de anticipat, ceea ce face ca măsurile tradiționale de securitate să fie insuficiente. Aici intervine interesul meu pentru cadrul SEC-SDN. Utilizând rețele definite prin software (SDN), putem crea o arhitectură de securitate mai flexibilă și mai receptivă, capabilă să se adapteze în timp real la amenințările emergente. Acest cadru reprezintă o frontieră nouă în securitatea cibernetică, având potențialul de a revoluționa protecția sistemelor cloud.

Totuși, cred că implementarea unor soluții tehnice avansate nu este suficientă. Securitatea este, în esență, un joc strategic între apărători și atacatori. De aceea, aplicarea teoriei jocurilor în acest cadru mă pasionează atât de mult. Teoria jocurilor ne permite să modelăm interacțiunile dintre actorii rău intenționați și sistemele de securitate ca o serie de mișcări strategice și contra-mișcări. Înțelegând aceste dinamici, putem concepe protocoale de securitate care nu doar reacționează la atacuri, ci le anticipează și le neutralizează înainte de a deveni o amenințare reală. Scopul este să fim mereu cu un pas înaintea celor care încearcă să exploateze vulnerabilitățile în scopuri proprii.

Această lucrare reprezintă culminarea pasiunii mele pentru securitatea cibernetică, a dedicării mele pentru avansarea acestui domeniu și a dorinței mele de a proteja lumea digitală de care depindem cu toții. Sunt condus de convingerea că putem și trebuie să îmbunătățim securitatea sistemelor cloud, iar această cercetare este modul meu de a contribui la acest obiectiv. Prin explorarea intersecției dintre cadrul SEC-SDN și

teoria jocurilor, nu doar că extindem limitele posibilului în securitatea cibernetică, ci contribuim la construirea unui viitor mai sigur și mai protejat pentru toți.

1.2 Problema cercetării

Creșterea exponențială a cloud computing-ului a revoluționat modul în care sunt stocate, procesate și accesate datele, oferind o scalabilitate și o flexibilitate fără precedent. Cu toate acestea, această expansiune rapidă a introdus și noi provocări de securitate, în special sub forma unor atacuri cibernetice sofisticate care vizează infrastructura cloud. Una dintre cele mai mari amenințări este reprezentată de atacurile Low-rate Distributed Denial of Service (LDDoS) și DDoS, concepute pentru a degrada subtil performanța serviciilor cloud fără a declanșa mecanismele tradiționale de detectare DDoS. Natura insidioasă a acestor atacuri le face deosebit de dificil de detectat și de atenuat, punând în pericol disponibilitatea și fiabilitatea serviciilor cloud.

Pentru a răspunde acestei probleme, cadrul SEC-SDN (Security-Enhanced Cloud folosind Software-Defined Networking) a apărut ca o soluție promițătoare. SEC-SDN valorifică flexibilitatea SDN pentru a ajusta dinamic politicile și configurațiile de securitate în timp real, oferind un răspuns mai rapid și mai eficient la amenințările în evoluție. Cu toate acestea, eficacitatea cadrului SEC-SDN depinde de capacitatea sa de a anticipa și contracara strategiile atacatorilor, în special în contextul atacurilor DDoS. Aici intervine importanța aplicării teoriei jocurilor.

Cum poate fi îmbunătățit cadrul SEC-SDN folosind teoria jocurilor pentru a detecta, atenua și preveni eficient atacurile DDoS în mediile cloud?

1.3 Obiective

Obiectivul principal al acestei teze de doctorat este de a explora și dezvolta soluții inovatoare pentru îmbunătățirea securității cibernetice în sistemele cloud prin integrarea cadrului SEC-SDN cu teoria jocurilor.

1.4 Organizarea tezei

Următoarele capitole ale tezei stabilesc subiectul și dezvoltă temele de cercetare menționate mai sus, după cum urmează:

- **Capitolul 2:** *Fundamente și lucrări conexe.* Se prezintă metode de securitate în cloud computing, teoria jocurilor în securitatea cibernetică și o prezentare generală a SDN.
- **Capitolul 3:** *Modele bazate pe teoria jocurilor pentru securitatea rețelelor.* Se

analizează tehnici defensive utilizând teoria jocurilor și implementarea algoritmică a jocului Stackelberg.

- **Capitolul 4:** *Îmbunătățirea securității rețelelor prin integrarea SEC-SDN.* Se prezintă cadrul SEC-SDN și abordările bazate pe teoria jocurilor pentru apărarea împotriva atacurilor DDoS.
- **Capitolul 5:** *Rezultate și analiză.* Se prezintă rezultatele și analiza performanței cadrului propus într-un mediu cloud real.
- **Capitolul 6:** *Concluzii.* Se rezumă contribuțiile și limitările acestei cercetări, precum și direcții viitoare de explorare.

CAPITOLUL 2

Fundamente și Lucrări Corelate

2.1 Fundamentele Cloud Computing

Cloud computing, ca un paradigm transformator în domeniul tehnologiei informației, oferă avantaje substanțiale în ceea ce privește scalabilitatea, flexibilitatea și potențialul de inovație. Cu toate acestea, introduce și provocări semnificative legate de securitate, necesitând o înțelegere cuprinzătoare a măsurilor de securitate existente și a vulnerabilităților specifice mediilor cloud. Această secțiune oferă o prezentare detaliată a măsurilor actuale de securitate în cloud computing, clasificându-le în trei domenii principale: securitatea fizică, securitatea rețelei și securitatea informației. Fiecare dintre aceste domenii abordează diferite aspecte ale protejării infrastructurilor cloud și a datelor acestora.

2.1.1 Măsuri Actuale de Securitate în Cloud Computing

Securitatea în mediile cloud este stratificată și include controale de securitate fizică, a rețelei și a informației:

- **Securitatea Fizică:** Se referă la protecția infrastructurii fizice, cum ar fi centrele de date, prin locații securizate, întăriri structurale, controale de acces pe mai multe niveluri, monitorizare constantă și măsuri de control ambiental. Centrele de date moderne pun accent pe reziliență, utilizând sisteme de rezervă și locații distribuite geografic pentru a asigura continuitatea serviciilor în cazul unor perturbări fizice.
- **Securitatea Rețelei:** Include mecanisme precum firewall-uri, protocoale de criptare și sisteme de detectare a intruziunilor (IDS), menite să prevină accesul neautorizat și atacurile asupra rețelei. Aceste măsuri asigură protecția infrastructurii cloud împotriva diferitelor amenințări la nivel de rețea.
- **Securitatea Informației:** Se concentrează pe protecția datelor prin mecanisme precum autentificarea, controalele de acces și criptarea. Acest strat de securitate este proiectat pentru a asigura confidențialitatea, integritatea și disponibilitatea datelor stocate și procesate în mediile cloud.

În ciuda acestor cadre de securitate cuprinzătoare, mediile cloud rămân vulnerabile din cauza complexității lor și a factorului uman. Riscurile asociate cu multi-tenancy, posibilele configurări greșite și amenințările în continuă evoluție, care pot depăși măsurile de apărare existente, subliniază necesitatea unei îmbunătățiri continue a securității. Cercetările viitoare trebuie să prioritizeze strategii de securitate adaptive și predictive, cu accent pe colaborarea dintre furnizorii de cloud, utilizatori și experți în securitate. În special, integrarea învățării automate și a inteligenței artificiale pentru securitate predictivă, alături de respectarea standardelor internaționale, este considerată crucială pentru îmbunătățirea securității mediilor cloud.

2.1.2 Provocări în Securizarea Rețelelor Software Defined Networks (SDN) în Mediile Cloud

Această secțiune examinează, de asemenea, provocări specifice, cum ar fi atacurile de tip Low-rate Distributed Denial of Service (LDDoS) în rețelele Software Defined Networks (SDN) din mediile cloud. Soluția propusă implică un cadru unificat de detectare, care utilizează capacitățile de management centralizat ale SDN pentru a detecta și atenua astfel de amenințări în mod eficient. Capacitatea SDN de a centraliza managementul rețelei oferă un avantaj semnificativ în abordarea acestor provocări de securitate prin configurarea dinamică a măsurilor de apărare și creșterea vizibilității asupra întregii rețele.

2.1.3 Modelele Cloud Computing: Cloud Privat, Public și Hibrid

Cloud computing poate fi clasificat în trei modele de implementare, fiecare având caracteristici, avantaje și provocări distincte:

- **Cloud Privat:** Un model în care infrastructura cloud este dedicată unei singure organizații, oferind un control sporit, securitate îmbunătățită și capacități de personalizare. Cloud-urile private permit implementarea unor protocoale de securitate personalizate și respectarea cerințelor de conformitate, fiind deosebit de valoroase în sectoare cu reglementări stricte, cum ar fi sănătatea și finanțele [1,2]. Cu toate acestea, costurile asociate cu investițiile în infrastructură și întreținerea acesteia, alături de riscul de dependență de un furnizor unic, reprezintă provocări semnificative [5].

- **Cloud Public:** Implică servicii cloud furnizate prin internet, disponibile pentru orice utilizator dispus să le achiziționeze sau să le utilizeze. Principalele avantaje ale acestui model includ costurile reduse, scalabilitatea și accesibilitatea [3]. Totuși, există îngrijorări privind multi-tenancy, resursele partajate și dependența de tehnologii proprietare [6]. Tendințele actuale în dezvoltarea cloud-ului public

se concentrează pe computing-ul serverless, integrarea inteligenței artificiale și sustenabilitate [4].

- **Cloud Hibrid:** Combină infrastructurile cloud private și publice, oferind flexibilitate, structuri de cost optimizate și capacități îmbunătățite de conformitate. Cloud-urile hibride permit alocarea strategică a resurselor, valorificând beneficiile de cost ale cloud-ului public, menținând în același timp securitatea oferită de cloud-ul privat [1, 5]. Cu toate acestea, gestionarea acestui mediu dual implică complexități legate de integrare, coerența măsurilor de securitate și administrarea costurilor.

2.1.4 Servicii și Roluri în Cloud Computing

Pe baza Arhitecturii de Referință NIST pentru Cloud Computing, ecosistemul cloud implică mai mulți actori cheie: Consumatorii de Cloud, Furnizorii, Auditorii, Brokerii și Transportatorii [4]. Fiecare actor joacă un rol esențial în ciclul de viață al serviciilor cloud, de la aprovizionare la gestionare, audit și interconectivitate, subliniind importanța unei abordări coordonate a securității în cloud.

2.1.5 Soluții Tradiționale și Avansate de Securitate în Cloud Computing

Această secțiune face, de asemenea, distincția între soluțiile tradiționale și cele avansate de securitate aplicabile în mediile cloud:

- **Soluții Tradiționale de Securitate:** Includ firewall-uri, sisteme de detectare și prevenire a intruziunilor (IDS), instrumente anti-malware, criptare, controale de acces, VPN-uri și audituri periodice [8]. Aceste măsuri fundamentale oferă prima linie de apărare împotriva amenințărilor obișnuite la securitatea cloud.

- **Soluții Avansate de Securitate:** Abordează complexitățile unice ale mediilor cloud prin mecanisme precum Autentificarea Multi-Factor (MFA), Sisteme de Management al Informațiilor de Securitate și Evenimentelor (SIEM), detectarea anomaliilor bazată pe AI, modelele de securitate Zero Trust, Prevenirea Pierderii Datelor (DLP), micro-segmentarea, Brokerii de Securitate pentru Acces în Cloud (CASB) și securitatea containerelor [9]. Aceste soluții avansate reprezintă o strategie de apărare mai proactivă și conștientă de context.

2.1.6 Identificarea Vulnerabilităților și Evoluția Atacurilor DDoS

Mediile cloud, în ciuda beneficiilor lor, nu sunt imune la vulnerabilități, cum ar fi brește de date, erorile umane, amenințările interne și atacurile DDoS. Modelul de amenințări STRIDE și clasificările OWASP oferă o abordare structurată pentru înțelegerea acestor

riscuri [10]. În special, evoluția atacurilor DDoS, de la atacuri simple și previzibile la asalturi sofisticate, multi-vectoriale, facilitate de botnet-uri și AI, ridică preocupări serioase [11].

Bibliografie

[1] Li D, Yan L, Song Y, Li S, Liang H. Network computer security and protection measures based on information security risk in cloud computing environment. In: 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA); 2021. p. 424-427. doi: 10.1109/AEECA52519.2021.9574194.

[2] Xiao G. Research on cyberspace security system based on cloud computing environment. In: 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE); 2020. p. 425-430. doi: 10.1109/ICISCAE51034.2020.9236841.

[3] Li S, Dang F, Yang Y, Liu H, Song Y. Research on computer network security protection system based on level protection in cloud computing environment. In: 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA); 2021. p. 428-431. doi: 10.1109/AEECA52519.2021.9574216.

[4] Bhajantri LB, Mujawar T. A survey of cloud computing security challenges, issues and their countermeasures. In: 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC); 2019. p. 376-380. doi: 10.1109/I-SMAC47947.2019.9032545.

[5] Kavitha T, Hemalatha S, Saravanan TM, Singh AK, Alam MI, Warshi S. Survey on cloud computing security and scheduling. In: 2022 International Conference on Computer Communication and Informatics (ICCCI); 2022. p. 1-4. doi: 10.1109/ICCCI54379.2022.9740932.

[6] Tadeo DAG, John SF, Bhaumik A, Neware R, Yamsani N, Kapila D. Empirical analysis of security enabled cloud computing strategy using artificial intelligence. In: 2021 International Conference on Computing Sciences (ICCS); 2021. p. 83-85. doi: 10.1109/ICCS54944.2021.00024.

[7] National Institute of Standards and Technology. NIST cloud computing reference architecture [Internet]. 2021 [cited 2023 Aug 8]. Available from: https://www.nist.gov/publications/nist-cloud-computing-reference-architecture

[8] Abdulsalam YS, Hedabou M. Security and privacy in cloud computing: technical review. Future Internet. 2022;14(11). doi: 10.3390/fi14010011.

[9] Tahirkheli AI, Shiraz M, Hayat B, Idrees M, Sajid A, Ullah R, et al. A survey on modern cloud computing security over smart city networks: threats, vulnerabilities, consequences, countermeasures, and challenges. Electronics. 2021;10(1811). doi: 10.3390/electronics10151811.

[10] Mishra P, Pilli ES, Varadharajan V, Tupakula U. Intrusion detection techniques in a cloud environment: a survey. J Netw Comput Appl. 2017;77:18–47.

[11] Durkota K, Lisy V, Bosansky B, Kiekintveld C. Approximate solutions for attack graph games with imperfect information. In: Proceedings of the Lecture Notes in Computer Science book series; 2015. p. 1–7.

CAPITOLUL 3

Modele de Teorie a Jocurilor pentru Securitatea Rețelelor

3.1 Introducere în Teoria Jocurilor în Securitatea Cibernetică

Integrarea teoriei jocurilor în securitatea cibernetică oferă metode inovatoare pentru abordarea amenințărilor avansate și în continuă evoluție, în special prin prisma Decepției Defensive (DD). Măsurile de securitate tradiționale, precum controlul accesului și detectarea intruziunilor, devin insuficiente împotriva atacatorilor sofisticați, care pot ocoli sau evita aceste mecanisme. În acest context, decepția apare ca o strategie unică menită să distorsioneze percepția atacatorului și să îl determine să ia decizii suboptimale, acționând astfel împotriva propriilor sale obiective.

Cercetarea privind Decepția Defensivă (DD) este profund influențată de teoria jocurilor, unde interacțiunile dintre atacatori și apărători sunt modelate ca jocuri strategice. În aceste modele, apărătorii utilizează tactici decepționiste pentru a induce în eroare atacatorii, determinându-i să aleagă strategii de atac ineficiente sau inferioare. Mai multe studii au folosit modele bazate pe teoria jocurilor pentru a optimiza strategiile de decepție, evidențiind interacțiunea dintre măsurile defensive și tacticile adversariale.

Studii de referință din literatura de specialitate oferă clasificări cuprinzătoare ale tehnicilor DD, furnizând perspective valoroase asupra diferitelor strategii și a eficienței acestora. De exemplu, Rowe și Rrushi [1] clasifică metodele DD în impersonare, întârzieri, falsuri, camuflaj, scuze false și inginerie socială. Ei analizează detectabilitatea și eficiența acestor tehnici, însă oferă doar o prezentare sumară a aplicațiilor teoretice bazate pe jocuri. În schimb, Han et al. [2] propun o clasificare mai structurată, concentrându-se pe obiective, entitățile implicate, straturile de implementare și implementarea strategică. Deși analiza lor este detaliată, discuția despre decepția bazată pe teoria jocurilor rămâne limitată. Pawlick et al. [3] explorează mai profund metodele bazate pe teoria jocurilor, clasificând decepția în șase tipuri principale, precum perturbarea, apărarea cu ținte în mișcare, ofuscarea, amestecarea, honey-X și angajarea atacatorului, subliniind importanța teoriei jocurilor în strategiile defensive.

Decepția defensivă (DD) își are rădăcinile în practicile militare, unde decepția este utilizată pentru a induce adversarii în eroare cu privire la intențiile, capacitățile și strategiile proprii [4]. În securitatea cibernetică, aceasta se traduce prin utilizarea

tehnicilor de decepție pentru a manipula percepțiile atacatorilor, determinându-i astfel să ia decizii suboptimale care favorizează apărătorul. Conform lui Almeshekah și Spafford [5], decepția cibernetică implică acțiuni intenționate menite să inducă în eroare atacatorii, forțându-i să acționeze în moduri care îmbunătățesc securitatea sistemului.

Literatura de specialitate discută diverse taxonomii pentru DD, concentrându-se pe factori precum tehnici, autenticitatea obiectelor, rezultatele așteptate, obiectivele generale și natura proactivă sau pasivă a decepției [6]. Tehnicile de decepție frecvent utilizate includ mascarea, reambalarea, orbirea, imitarea, inventarea, momeala și capcanele. Aceste tehnici se inspiră din strategii militare precum ascunderea, camuflajul și dezinformarea, adaptate pentru contextul securității cibernetice cu scopul de a deruta și induce în eroare atacatorii.

Principiile de proiectare ale DD se concentrează pe identificarea tipului de atacator ce urmează a fi indus în eroare, determinarea momentului optim pentru implementarea decepției și alegerea tehnicilor adecvate de decepție. Aceste principii vizează țintirea eficientă a atacatorilor potriviți la momentul potrivit, utilizând metode adecvate pentru maximizarea eficacității defensive, minimizând în același timp costurile. DD oferă avantaje semnificative, precum eficiența costurilor, completarea altor măsuri de apărare, versatilitate la diferite niveluri ale sistemului și eficacitate împotriva amenințărilor automate. Totuși, prezintă și unele dezavantaje, printre care dificultatea de a evalua corect intențiile atacatorilor, riscul de a deruta utilizatorii legitimi, necesitatea unei întrețineri continue și creșterea costurilor odată cu sofisticarea atacatorilor. De asemenea, DD împărtășește similarități cu alte strategii defensive, precum Apărarea cu Ținte în Mișcare (MTD) și ofuscarea, dar se distinge prin utilizarea informațiilor false pentru a manipula percepțiile atacatorilor.

3.2 Tehnici de Decepție Defensivă bazate pe Teoria Jocurilor

Teoria jocurilor este utilizată frecvent pentru a modela scenarii de securitate cibernetică, concentrându-se pe interacțiunile strategice dintre atacatori și apărători. Principalele elemente ale modelelor bazate pe teoria jocurilor în securitatea cibernetică includ rolurile jucătorilor (atacatori și apărători), informația din joc (completă vs. incompletă, perfectă vs. imperfectă), observabilitatea, strategiile (pure vs. mixte), incertitudinea recompenselor, considerentele de utilitate, structura jocului (jocuri complete vs. sub-jocuri) și dinamica jocului (static vs. dinamic). Aceste elemente sunt esențiale pentru construirea unor modele eficiente care să surprindă complexitatea mediilor reale de securitate cibernetică.

Mai multe tipuri de jocuri sunt utilizate frecvent în strategiile de decepție defensivă:

- **Jocuri Bayesiene**: Aceste jocuri abordează informația incompletă, în care jucătorii

pot să nu cunoască caracteristicile adversarilor lor, dar au distribuții de probabilitate subiective care reflectă convingerile lor [4].

- **Jocuri Stackelberg**: Un joc ierarhic în care un jucător (liderul) face prima mutare, iar celălalt (urmăritorul) răspunde în consecință, oferind liderului un avantaj strategic prin influențarea acțiunilor urmăritorului [4].

- **Jocuri de Semnalizare**: Un subset al jocurilor Bayesiene, în care un jucător trimite un semnal informațional către altul. Costuri sunt asociate cu semnalele înșelătoare, iar rezultatul jocului depinde de interpretarea semnalului de către destinatar [8].

- **Jocuri Stochastice**: Jocuri multi-etapă în care starea sistemului evoluează probabilistic în funcție de acțiunile jucătorilor. Aceste jocuri iau în considerare atât factori stocastici interni, cât și externi, în determinarea tranzițiilor de stare și a recompenselor.

- **Procese Decizionale Markoviene Parțial Observabile (POMDPs)**: Acestea implică luarea deciziilor în condiții de incertitudine, în care un agent încearcă să maximizeze recompensele fără a avea cunoștințe complete despre starea sistemului. Acest model este deosebit de relevant în scenariile în care apărătorii trebuie să deducă acțiunile atacatorilor pe baza observațiilor limitate.

Teoria jocurilor oferă un cadru matematic puternic pentru modelarea și analiza procesului decizional strategic în securitatea cibernetică. Prin utilizarea modelelor bazate pe teoria jocurilor, apărătorii pot manipula strategic percepțiile atacatorilor, selecta strategii defensive optime și determina momentele și metodele adecvate pentru implementarea decepției. Aplicarea teoriei jocurilor în decepția defensivă îmbunătățește eficacitatea măsurilor de securitate cibernetică prin introducerea impredictibilității și manipulării cognitive în strategiile defensive.

3.3 Jocul Stackelberg pentru Strategii Defensive

Cadrul graficului de atac este conceput pentru a colecta informații despre o rețea, inclusiv topologia sa, vulnerabilitățile, configurația și conectivitatea, pentru a genera un grafic de atac. Modelul a fost propus, iar rezultatele au fost prezentate în [9].

Accesul la rețea pentru o sursă și o destinație este necesar pentru a obține detalii despre rețea. Acest aspect poate fi modelat într-o matrice de accesibilitate, care este o matrice bidimensională, având ca primă dimensiune IP-ul sursă și ca a doua dimensiune IP-ul destinație. Matricea de accesibilitate oferă o perspectivă directă asupra punctelor slabe ale sistemului și arată cum un atacator ar putea exploata anumite căi.

Ca o specializare a graficului generic de atac, această matrice reprezintă o metodă mai simplificată de exprimare a căilor într-o structură matricială. Rândurile și coloanele reprezintă stările sau condițiile sistemului, iar intrările demonstrează dacă o stare poate duce la alta.

Puterea matricei de accesibilitate constă în forma sa concisă. Graficele de atac convenționale pot deveni destul de dificil de gestionat, iar situația se înrăutățește odată cu proliferarea vulnerabilităților și a căilor de atac. Pe de altă parte, reprezentarea matricială surprinde aceste informații într-un mod clar și structurat. În această matrice, fiecare element exprimă un indicator binar: dacă exploatarea într-o stare poate sau nu să conducă la o altă stare.

În plus, matricile de accesibilitate sunt compatibile cu anumite funcții matematice care oferă informații suplimentare. De exemplu, pătratul matricei generează o nouă matrice ce reprezintă căi pentru două etape. Dacă acest proces de ridicare la putere continuă, vor fi evidențiate căi mai lungi, permițând astfel analistului să navigheze toate posibilitățile de atac multi-etapă. Această tehnică oferă o soluție mai versatilă pentru explorarea sistemelor complexe care includ mai multe vulnerabilități și interconexiuni. Pe baza matricei de accesibilitate, analiștii pot identifica punctele cheie ale stărilor sistemului ce sunt traversate de diferite căi de atac. Aceste stări sunt adesea corelate cu vulnerabilități semnificative. Prin reducerea acestor vulnerabilități, riscul general al sistemului poate fi redus semnificativ. În concluzie, matricea ajută la identificarea amenințărilor care trebuie abordate cu prioritate, astfel încât resursele să fie aplicate asupra vulnerabilităților cele mai urgente.

Tabelul 3.1 prezintă definiția rețelei de accesibilitate; în acest tabel, considerăm o VM (mașină virtuală) ca fiind o entitate găzduită pe un server, având un IP alocat și servicii care rulează pe ea, permițând interacțiunea cu alte servicii pe diferite VM-uri.

Table 3.1 Matricea de accesibilitate.

			Destinații		
			VM_x IP_x	VM_y IP_y	VM_z IP_z
Surse	VM_a	IP_a	$Service_m$		$Service_o$
	VM_b	IP_b		$Service_n$	

Rețelele mari, care conțin diverse platforme, sisteme de operare și tipuri diferite de conectivitate, prezintă inevitabil probleme de securitate care pot trece neobservate de administratorii de securitate.

Modelul nostru este un model de joc Stackelberg, unde apărătorul face prima mutare în joc, urmat de atacator. Utilizarea honeypots ajută apărătorii să-și securizeze

rețeaua și creează un mediu complex pentru atacatori. Interacțiunea dintre graficele de atac și modelul de joc Stackelberg îmbină teoria jocurilor cu securitatea cibernetică, oferind astfel un cadru strategic pentru analiza și contracararea amenințărilor potențiale. Jocul Stackelberg a fost aplicat în diverse contexte de securitate încă de la prima sa publicare: un apărător (în general considerat lider) face prima mutare, iar atacatorul (urmăritorul) răspunde la această mutare.

Pentru ușurința manipulării, considerăm echivalente două gazde care rulează aceleași servicii și au conectivitate similară.

Notări:

N - rețeaua cu servicii/gazde;

s - un serviciu/gazdă;

T - tipuri de servicii/gazde;

$T = [t_1, t_2, t_3, ., t_n]$, unde t reprezintă un tip de serviciu/gazdă;

$S = [s_1, s_2, s_3, ., s_n]$ – gazde/servicii de tip t;

$H = [h_1, h_2, h_3, ., h_n]$ - honeypots de tip t în rețeaua N;

$A = [a_1, a_2, a_3, ., a_n]$ - acțiunile atacatorului;

$D = [d_1, d_2, d_3, ., d_n]$ - acțiunile apărătorului.

Prin plasarea unor honeypots în rețea, mai mulți apărători pot fi simulați. Rețeaua va fi reprezentată astfel:

$$d(t) = h(t) + s(t). \tag{3.1}$$

Probabilitatea ca un atacator să aleagă un honeypot crește odată cu numărul de honeypots implementate în rețea. O alertă ar trebui generată pentru apărător dacă atacatorul reușește să compromită un honeypot, iar atacul ar trebui oprit înainte de a cauza indisponibilitatea rețelei.

Totul se reduce la costuri și probabilități de succes, astfel că definim:

$c(s)$ - costul serviciului sau gazdei, care poate fi duplicat prin honeypot;

$l(s)$ – pierderea de serviciu și date;

$c(a)$ - costul atacatorului;

$c(d)$ - costul apărătorului;

$p(a)$ - probabilitatea de succes a atacatorului;

$resp(a)$ - reacția atacatorului la mutarea apărătorului;

$resp(d)$ - răspunsul apărătorului la acțiunea atacatorului.

Honeypots sunt gazde false care imită gazdele reale și rulează servicii identice. Scopul utilizării honeypots este de a introduce complexitate în mediu, astfel încât apărătorul să câștige timp și să obțină mai multe informații despre un atac. Lucrarea noastră se concentrează pe primul jucător din joc, apărătorul. Costul instalării și întreținerii unui honeypot depinde de tipul gazdei reale.

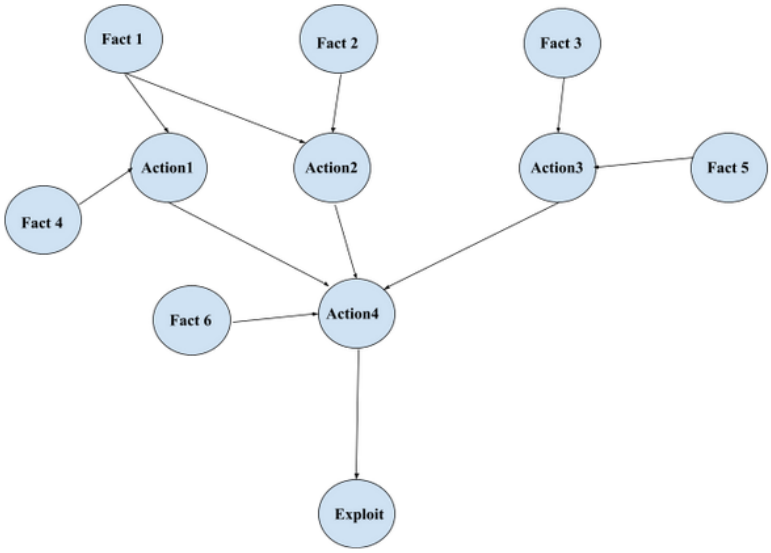


Fig. 3.1 Grafic de atac.

Un grafic de atac descrie relațiile dintre vulnerabilitățile exploatabile pe fiecare nod. În [10], nodurile sunt utilizate pentru a reprezenta stările unui sistem, inclusiv gazde, privilegii și vulnerabilități. Diferite grafice de atac pot fi generate în funcție de reprezentarea nodurilor și muchiilor. Figura 3.1 arată că pentru fiecare fapt adevărat, există o acțiune asociată, având o probabilitate și un cost ce depind de tipul gazdei atacate.

3.3.1 Implementarea Algoritmică a Soluțiilor bazate pe Teoria Jocurilor

Pentru a reduce costurile de protecție, acțiunile a ale atacatorului și d ale apărătorului trebuie să fie opuse ca performanță, deoarece performanța costului trebuie să crească pentru atacator și să scadă pentru apărător. Câștigul G pentru atacator și apărător va fi $G^a(a,d)$ și $G^d(a,d)$, ceea ce va conduce la un joc cu sumă nulă:

$$G^a(a,d) = -G^d(a,d). \tag{3.2}$$

Fiind lider în acest joc, apărătorul poate lua decizii și poate seta costurile primul. Echilibrul Stackelberg (*SE*) este atins atunci când acțiunea apărătorului d este luată în considerare de atacator în formularea răspunsului său a.

Propunem **Jocul Stackelberg pentru Cursul Acțiunii (CoASG)**, utilizând costurile pentru fiecare parte și procesul jocului.

Pentru a modela deciziile și acțiunile atacatorului, pe lângă informațiile colectate despre rețea, se iau în considerare următoarele aspecte:

1. Atacatorul poate încheia atacul în orice moment.

2. Când interacționează cu un honeypot, probabilitatea de succes este

$$p_1 = 1 - \sum_{t \in T} \frac{a_t - h_t}{a_t}. \tag{3.3}$$

3. Când nu există interacțiune cu un honeypot și nu are succes, probabilitatea este

$$p_3 = 0. \tag{3.4}$$

Utilizând algoritmul Backward Induction, putem găsi un Echilibru Stackelberg. Pot exista mai multe echilibre SE, dar alegem SE cu cel mai mic cost pentru fiecare parte.

Faza 1. (a) Atacatorul ia în considerare acțiunea apărătorului *d* și răspunde în consecință:

$$\textit{resp}(d) = \arg\max_a G^a(a, d). \tag{3.5}$$

(b) Cel mai bun Curs al Acțiunii ales de atacator, dacă există multiple SE, va avea cel mai mic cost:

$$\textit{resp}_0(d) = \arg\min_{\textit{resp}(d)} \left(\sum_{i=1}^{n} c(a)_i \textit{resp}(d)_i \right). \tag{3.6}$$

Faza 2. (a) Din perspectiva apărătorului, Cursul Acțiunii este ales după răspunsul atacatorului pentru a maximiza câștigul:

$$\textit{resp}(a) = \arg\max_d G^d(\textit{resp}_0(d), d). \tag{3.7}$$

(b) Alegerea SE cu cel mai mic cost oferă noul Curs al Acțiunii:

$$\textit{resp}_0(a) = \arg\min_{\textit{resp}(a)} \left(\sum_{i=1}^{n} c(d)_i \textit{resp}(a)_i \right). \tag{3.8}$$

Costul apărătorului pentru adăugarea unui honeypot $h(t)$ în rețea este:

$$c_d(h) = \gamma l_d(h), \tag{3.9}$$

unde $\gamma \in \mathbb{R}^+$ este un parametru ajustabil.

În [11], se afirmă că securizarea unei infrastructuri informatice este extrem de costisitoare și există o nevoie clară de dezvoltare a unui sistem automatizat de suport decizional pentru a asista un administrator de securitate în configurarea apărării rețelei și detectarea atacurilor. Au fost utilizate honeypots pentru a întări rețeaua și a acționa ca momeli pentru intruși.

Metoda noastră propune definirea costului fiecărui nod din graficul de atac ca

un raport între costul nodului și numărul de vulnerabilități coexistente pe acel nod. Costul primului nod este distribuit către nodurile următoare conform unor reguli specifice până la atingerea nodului țintă ce trebuie apărat. Abordarea noastră adoptă Procesul Decizional Markovian Parțial Observabil (POMDP) pentru a concentra strategia apărătorului asupra înșelării atacatorului și maximizării câștigului său. Deoarece atacatorul poate obține unele informații despre rețea în timpul recunoașterii, presupunem că fiecare jucător poate observa doar o stare parțială. Prin urmare, propunem o soluție pentru Jocul Stocastic Parțial Observabil utilizând POMDP.

Notări:

V - reprezintă o vulnerabilitate, unde $V = [V_1, V_2, V_3, \ldots, V_n]$;

F - reprezintă toate stările din spațiul acțiunilor, unde $F = [f_1, f_2, f_3, \ldots, f_n]$;

O^a - reprezintă observațiile atacatorului, unde $O^a = [o_1^a, o_2^a, o_3^a, \ldots, o_n^a]$;

O^d - reprezintă observațiile apărătorului, unde $O^d = [o_1^d, o_2^d, o_3^d, \ldots, o_n^d]$;

B - reprezintă stările de credință, unde $B = [b_1, b_2, b_3, \ldots, b_n]$.

O vulnerabilitate este reprezentată de un nod *V*, iar muchia care leagă două noduri V_1 și V_2 indică faptul că a doua vulnerabilitate, reprezentată de nodul V_2, poate fi exploatată doar dacă prima vulnerabilitate V_1 este exploatată. Atacatorul decide ce nod să atace prin exploatarea vulnerabilității, iar apărătorul decide câte honeypots să implementeze și unde de-a lungul muchiilor grafului.

În acest stadiu, introducem O^a ca setul de observații al atacatorului și O^d ca setul de observații al apărătorului:

$$Pr(o_n^a, o_n^d, f_n \mid f_1, o_1^a, o_1^d). \tag{3.10}$$

$G^a(f, d, a)$ reprezintă câștigul atacatorului în starea *f*, astfel încât putem presupune că:

$$V(b_n) = \max_{a_n}[G^a(a_n, f_n) + \sum_{f \in F} \tau(f_1, d, f_n)V(f_1)]. \tag{3.11}$$

Din studiul [5], putem observa diferența dintre metodele prezentate, subliniind că Procesul Decizional Markovian (MDP) poate fi aplicat într-o stare vizibilă, pe când modelul nostru utilizează POMDP pentru o stare invizibilă, unde un adversar trebuie să observe, să colecteze informații și apoi să acționeze.

3.4 Simularea și Analiza Modelului Bazat pe Teoria Jocurilor

Să considerăm câteva notații care ne vor ajuta să înțelegem modelul. Pe fiecare gazdă rulează mai multe servicii, care pot fi reale (λ_{real}) sau honeypot ($\lambda_{honeypot}$). În modelul nostru, am luat în considerare mai multe gazde, care pot fi privite ca servere, pe care sunt configurate și rulează servicii precum servicii web, baze de date, servicii de găzduire

fișiere etc. Printre serviciile reale, există honeypots care servesc drept momeală pentru atacatori.

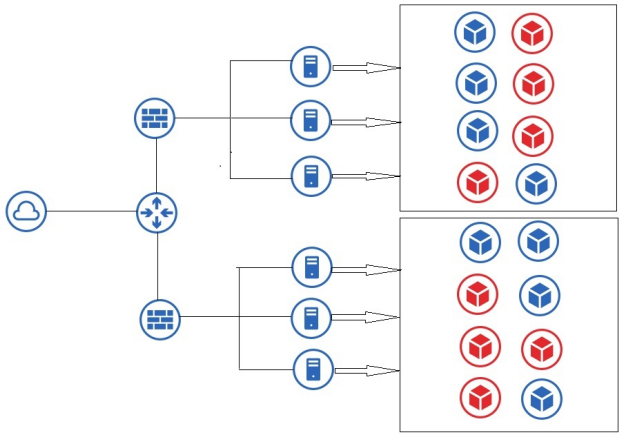


Fig. 3.2 Reprezentarea modelului sistemului

Modelul este prezentat în Figura 3.2, unde accesul la cloud (exterior) se face printr-un router, firewall-urile protejează rețeaua internă prin reguli aplicate, iar pe fiecare mașină fizică rulează mai multe mașini virtuale (gazde). Aceste gazde pot fi reale sau honeypots, însă mirroring-ul este realizat pentru cele de mare interes.

În sistemul nostru, am folosit notații pentru a descrie serviciile, care pot fi în trei stări la un moment dat:

Starea 1. Serviciul (λ) este activ/deschis (λ_{real}).

Starea 2. Serviciul este un honeypot $(\lambda_{honeypot})$.

Starea 3. Serviciul este închis (λ_{closed}).

Mulțimea serviciilor din fiecare stare pentru o gazdă sau un server poate fi notată astfel:

$$\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}, \tag{3.12}$$

dar din cauza honeypots, serviciile devin:

$$\Lambda = \{\lambda_{11}, \lambda_{12}, ..., \lambda_{1n}\}. \tag{3.13}$$

În sistemul nostru, există mai multe servicii care pot fi în 4 stări:

Starea 1.λ_{10} serviciul este închis.

Starea 2.λ_{11} serviciul este deschis.

Starea 3.λ_{20} utilizatorul din rețea nu poate accesa serviciul.

Starea 4.λ_{21} utilizatorul din rețea poate accesa serviciul.

Pentru a defini o strategie simplă, să considerăm că un serviciu poate fi real, honeypot sau închis, atunci din (3.13) putem decide când atacatorul poate accesa serviciile:

$$\lambda = \{\lambda_{111}, \lambda_{121}, ..., \lambda_{1n1}\} \tag{3.14}$$

și când atacatorul nu poate accesa serviciile:

$$\lambda = \{\lambda_{110}, \lambda_{120}, ..., \lambda_{1n0}\} \tag{3.15}$$

Din (3.14) și (3.15) rezultă că strategia simplă α este:

$$\alpha = \{\lambda_{111}, \lambda_{110}\} \tag{3.16}$$

Pentru a calcula câștigurile atacatorului și ale apărătorului, trebuie să definim următorii parametri cu condițiile:

$G^d > 0$;

C^a este costul atacatorului și are relația $G^d \geq C^a > 0$;

$G^h > 0$ este câștigul honeypotului;

$\gamma \geq 1$ este factorul de daune al atacatorului;

$\eta \geq 1$ este factorul de înșelăciune al utilizării unui honeypot.

Putem calcula câștigurile pentru două cazuri:

Cazul 1. Apărătorul oferă un serviciu real pe o gazdă, iar atacatorul îl poate accesa, atunci câștigurile sunt:

$$G^a = \gamma G^d - C^a \tag{3.17}$$

și

$$G^d = -\gamma G^a. \tag{3.18}$$

Cazul 2. Apărătorul oferă un serviciu fals, iar câștigurile sunt:

$$G^a = -\eta G^h - C^a \tag{3.19}$$

și

$$G^d = \eta G^h. \tag{3.20}$$

Având un sistem cu s servicii/gazde, rezultă matricea câștigurilor.

Table 3.2 Matricea câștigurilor.

		Atacator		Apărător	
		λ_{21}	λ_{20}	λ_{21}	λ_{20}
λ_{110}	λ_{11}	(-$\gamma G^d/s, \gamma G^d/s - C^a$)	(0,0)	(G^d, G^d)	(0,0)
	λ_{10}	(0, -C^a)	(0,0)	(-G^d, $-G^d$)	(0,0)
λ_{111}	λ_{11}	($\eta G^h/s, -\eta G^d/s - C^a$)	(0,0)	(0,-G^d)	(0,0)
	λ_{10}	(0,-C^a)	(0,0)	(0,-G^d)	(0,0)

Folosind matricea de recompense descrisă în Tabelul 3.2, putem afirma că avem o matrice $m * m$ $X = (x_{ij})_{mm}$, unde x_{ij} este reprezentată de recompensa jucătorului de rând atunci când este aleasă strategia i^{th}, iar jucătorul de coloană alege strategia j^{th}. Putem scrie matricea de recompense generală astfel:

$$\begin{bmatrix} -\gamma G^d/s & \gamma G^d/s - C^a & 0 & 0 \\ 0 & -C^a & 0 & 0 \\ \eta G^h/s & -\eta G^d/s - C^a & 0 & 0 \\ 0 & 0, -C^a & 0 & 0 \end{bmatrix}. \tag{3.21}$$

3.5 Evaluarea sistemului

Pentru simulare, ne concentrăm pe jocul dintre atacator și apărător, utilizând Gambit V15.1.1, care este o bibliotecă de software și instrumente de teorie a jocurilor pentru analiza jocurilor extensive și strategice finite [6], precum și MATLAB R2021b v9.11.0.

Parametrii utilizați pentru rularea simulării sunt definiți în Tabelul 3.3.

Table 3.3 Parametrii simulării.

Parametru	Valoare	Observație
G^d	50	recompensa apărătorului
C^a	40	costul atacatorului
G^h	40	recompensa honeypot-ului
γ	2	factor de daună
η	1	factor de inducere în eroare
N_1	1	rețea cu 1 serviciu / gazdă
N_{10}	10	rețea cu 10 servicii / gazde
N_{50}	50	rețea cu 50 servicii / gazde
N_{100}	100	rețea cu 100 servicii / gazde

Utilizând simularea Gambit pentru 1 serviciu/gazdă și aplicând matricea de câștiguri Tabelul 3.2, observăm că atacatorul are un avantaj real, sugerând că apărătorul poate suferi pierderi semnificative (Figura 3.3).

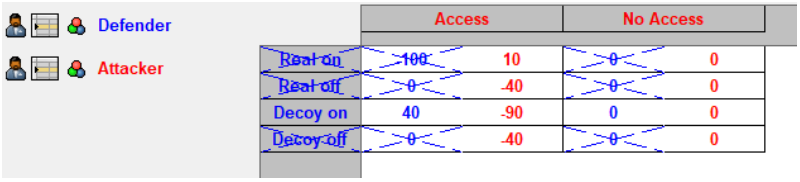


Fig. 3.3 Rezultatele dominanței pentru N_1.

Observăm din Figurile 3.4, 3.5, 3.6 că, atunci când numărul de servere sau gazde crește și resursele disponibile pentru honeypots sunt utilizabile, rezultatele arată că dominația se transferă către partea apărătorului, iar atacatorul nu mai poate accesa rețeaua cu ușurință.

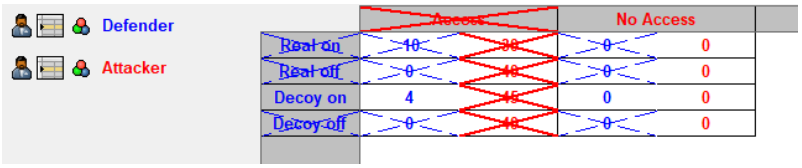


Fig. 3.4 Rezultatele dominației pentru N_{10}.

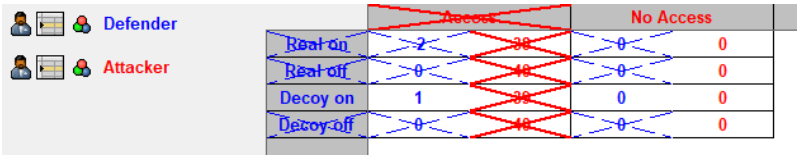


Fig. 3.5 Rezultatele dominației pentru N_{50}.

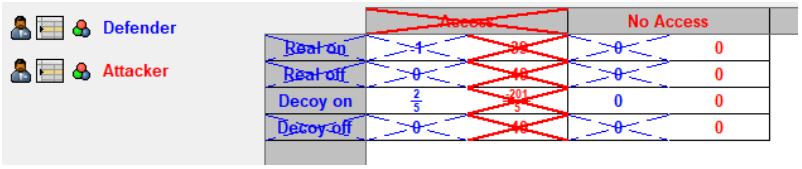
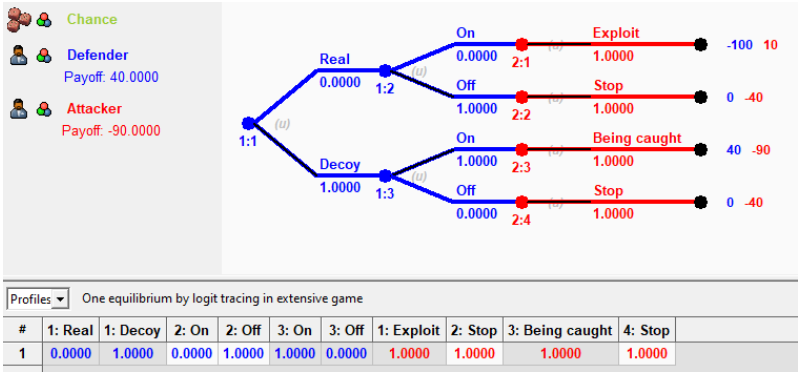


Fig. 3.6 Rezultatele dominației pentru N_{100}.

Observăm că, pe măsură ce numărul de servicii/gazde crește, câștigul apărătorului scade, fiind influențat de costurile honeypots-urilor. De asemenea, câștigul atacatorului scade în același timp în care câștigul apărătorului crește, iar costul atacării rețelei crește rapid până la punctul în care devine foarte riscant să fie prins sau consumă prea mult timp.

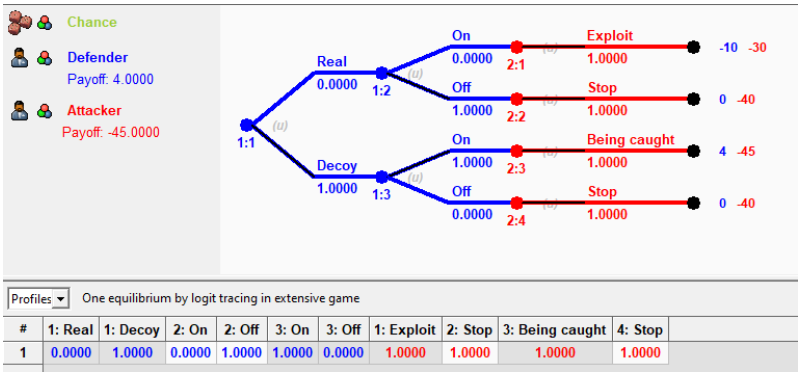
Într-o rețea mare pot exista peste 1000 de gazde sau servicii, dar aplicând metoda noastră de plasare a honeypots-urilor, putem concluziona că 50 de honeypots sunt suficiente pentru o strategie de apărare. Nu este necesar să implementăm mai multe honeypots, iar toate datele colectate pot fi utilizate pentru a atenua infiltrarea atacatorului în sistem. Scopul experimentului a fost de a demonstra că utilizarea honeypots poate crește securitatea unui sistem, dar întrebarea era când un administrator de sistem poate fi sigur că numărul de noduri false îl poate ajuta să protejeze rețeaua cu costuri mai mici.

Utilizând jocul extensiv (arborescent) din Gambit v15.1.1, am introdus valorile obținute în Figurile 3.3, 3.4, 3.5, 3.6 și apoi am calculat Echilibrul Nash. Rezultatele prezentate în Figurile 3.7, 3.8, 3.9, 3.10 demonstrează că metoda noastră propusă oferă un avantaj considerabil apărătorului. Honeypots-urile reduc dominația atacatorului și cresc șansele ca apărătorul să observe și să oprească acțiunile acestuia.



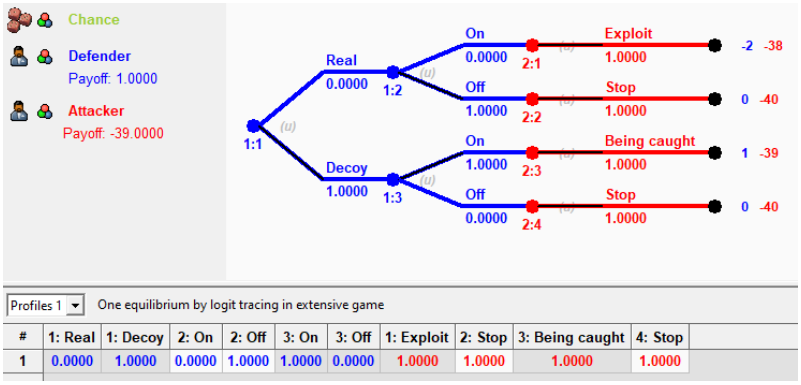
#	1: Real	1: Decoy	2: On	2: Off	3: On	3: Off	1: Exploit	2: Stop	3: Being caught	4: Stop
1	0.0000	1.0000	0.0000	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000

Fig. 3.7 Rezultatele dominației pentru N_1.



#	1: Real	1: Decoy	2: On	2: Off	3: On	3: Off	1: Exploit	2: Stop	3: Being caught	4: Stop
1	0.0000	1.0000	0.0000	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000

Fig. 3.8 Rezultatele dominației pentru N_{10}.



#	1: Real	1: Decoy	2: On	2: Off	3: On	3: Off	1: Exploit	2: Stop	3: Being caught	4: Stop
1	0.0000	1.0000	0.0000	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000

Fig. 3.9 Rezultatele dominației pentru N_{50}.

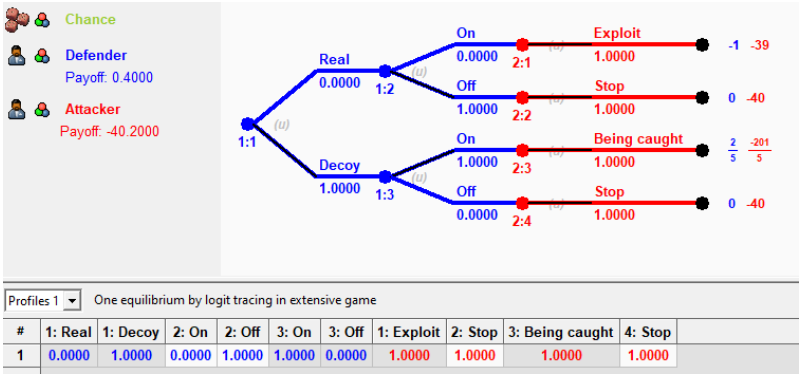


Fig. 3.10 Rezultatele dominației pentru N_{100}.

Echilibrul Nash este strategia cu o probabilitate mare pe care atacatorul o va lua în considerare, iar apărătorul încearcă să rezolve jocul imperfect cu informații limitate. Apărătorul plasează honeypots în rețea, iar atacatorul încearcă să le evite pentru a nu fi detectat. Ambii jucători suportă anumite costuri: apărătorul pentru plasarea honeypots-urilor și atacatorul pentru evitarea acestora și accesarea gazdelor reale. Am descoperit că jocul poate ajunge într-o fază de stagnare atunci când răspunsurile atacatorului și apărătorului $resp(d)$ și $resp(a)$ sunt egale cu 0. Acest lucru înseamnă că ambii jucători nu mai fac mișcări și rămân în modul inactiv.

Comparând algoritmul nostru cu o politică fixă de alocare a honeypots-urilor, observăm că cea din urmă nu recunoaște dinamica jocului și nu este flexibilă, deoarece toate informațiile sunt colectate la începutul jocului.

Acum, utilizând valorile din Tabelul 3.3, putem calcula strategia optimă folosind (3.21)

$$\begin{bmatrix} -100 & 10 & 0 & 0 \\ 0 & -40 & 0 & 0 \\ 40 & -90 & 0 & 0 \\ 0 & -40 & 0 & 0 \end{bmatrix}$$

Pentru a rezolva această matrice de câștiguri și a utiliza algoritmul dezvoltat în [7], trebuie să transformăm toate intrările în valori non-negative prin adăugarea unui număr fix adecvat. Adăugând 110, obținem noua matrice:

$$\begin{bmatrix} 10 & 120 & 110 & 110 \\ 110 & 70 & 110 & 110 \\ 150 & 20 & 110 & 110 \\ 110 & 70 & 110 & 110 \end{bmatrix}$$

Pentru a calcula în Matlab strategia optimă, traducem funcția de programare liniară în formatul:

$$min(x_1 + x_2 + x_3 + x_4)$$

,

-10 $x_1 - 110x_2 - 150x_3 - 110x_4 \leq -1, -120x_1 - 70x_2 - 20x_3 - 70x_4 \leq -1, -110x_1 - 110x_2 - 110x_3 - 110x_4 \leq -1, -110x_1 - 110x_2 - 110x_3 - 110x_4 \leq -1.$

Pentru a găsi soluția, putem utiliza funcția *linprog* din pachetul de optimizare din Matlab astfel:

c=[1,1,1,1];
a=[-10,-110,-150,-110;-120,-70,-20,-70;-110,-110,-110,-110;-110,-110,-110,-110];
b=[-1,-1,-1,-1];
lb=[0,0,0,0].

Aplicând formula *x = linprog (c,a,b,[],[],lb)*, aflăm că soluția optimă este:

$$x_r = (0.0032, 0, 0, 0.0088). \tag{3.22}$$

Pentru cel de-al doilea program liniar, avem transcrierea ecuației (3.25) și:

$$\min(-y_1 - y_2 - y_3 - y_4)$$

10 $y_1 + 120y_2 + 110y_3 + 110y_4 \leq 1, 110y_1 + 70y_2 + 110y_3 + 110y_4 \leq 1, 150y_1 + 20y_2 + 110y_3 + 110y_4 \leq 1, 110y_1 + 70y_2 + 110y_3 + 110y_4 \leq 1.$

c=[-1,-1,-1,-1];
a=[10,120,110,110;110,70,110,110;150,20,110,110;110,70,110,110];
b=[1,1,1,1];
lb=[0,0,0,0].

Astfel, după aplicarea în Matlab a comenzii *y = linprog (c,a,b,[],[],lb)*, obținem:

$$y_r = (0.004, 0.008, 0, 0). \tag{3.23}$$

După rularea funcției *linprog* în Matlab pentru 1, 10, 50 și 100 servicii/gazde în mediul nostru, utilizând metoda descrisă anterior, calculăm matricea de plată. Rezultatele simulării Matlab, prezentate în Figura 3.11, arată că, pe măsură ce N crește, costurile pentru ambii jucători cresc semnificativ.

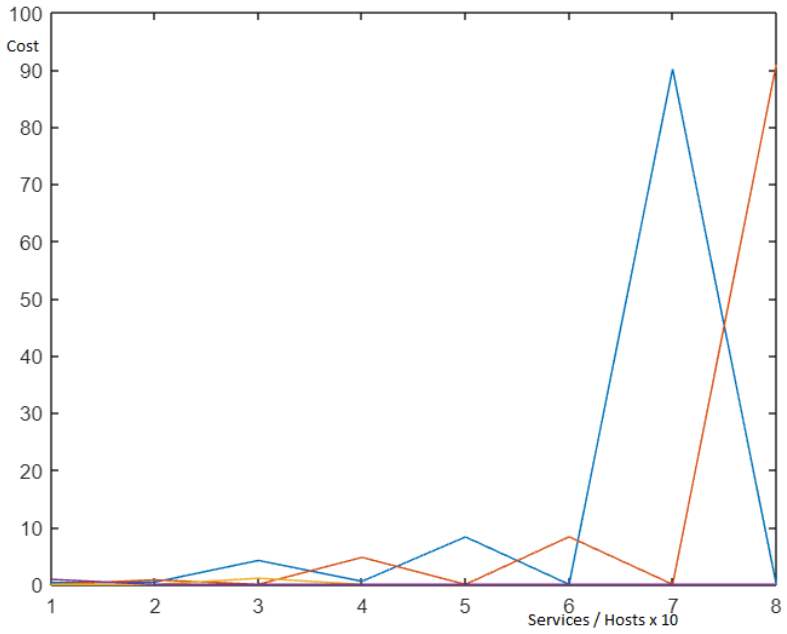


Fig. 3.11 Rezultatele matricei programării liniare.

Dacă folosim mai mult de 50 de honeypot-uri, costurile pentru ambii jucători cresc, iar probabilitatea ca atacatorul să fie prins devine extrem de mare. Prin urmare, o strategie de apărare cu doar 50 de honeypot-uri poate fi aplicată cu succes.

Bibliografie

[1] Rowe NC, Rrushi J. Introduction to Cyberdeception. Heidelberg: Springer Nat.; 2016.

[2] Han X, Kheir N, Balzarotti D. Deception techniques in computer security: A research perspective. ACM Comput Surveys. 2018 Jul;51(4):1–36.

[3] Pawlick J, Colbert E, Zhu Q. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. ACM Comput Surveys. 2019;52(4):1–28.

[4] Almeshekah MH, Spafford EH. Planning and integrating deception into computer security defenses. In: Proceedings of the New Security Paradigms Workshop; 2014. p. 127–138.

[5] Sharp WL. Military deception. Washington, DC: Joint War Fighting Center, Doctrine Educ. Group; 2006. Report No.: 3-13.4.

[6] Takabi H, Jafarian JH. Insider threat mitigation using moving target defense and deception. In: Proceedings of the International Workshop on Managing Insider Security Threats; 2017. p. 93–96.

[7] Nan S, Brahma S, Kamhoua CA, Njilla LL. On Development of a Game-Theoretic Model for Deception-Based Security. In: Kamhoua CA, editor. Game Theory and Machine Learning for Cyber Security. Hoboken, NJ: Wiley; 2020. p. 123–140.

[8] Anwar AH, Kamhoua C. Game theory on attack graph for cyber deception. In: Proceedings of the International Conference on Decision and Game Theory for Security; 2020. p. 445–456.

[9] Florea R, Craus M. A Game-Theoretic Approach for Network Security Using Honeypots. Future Internet. 2022;14(12):362. Available from: https://doi.org/10.3390/fi14120362

[10] Lallie HS, Debattista K, Bal J. A review of attack graph and attack tree visual syntax in cyber security. Comput Sci Rev. 2020;35:100219.

[11] Durkota K, Lisy V, Bosansky B, Kiekintveld C. Approximate solutions for attack graph games with imperfect information. In: Proceedings of the Lecture Notes in Computer Science book series; 2015. p. 1–7.

[12] Zeng J, Wu S, Chen Y, Zeng R, Wu C. Survey of attack graph analysis methods from the perspective of data and knowledge processing. Secur Commun Netw. 2019.

[13] McKelvey RD, McLennan AM, Turocy TL. Gambit: Software Tools for Game Theory, Version 15.1.1. Available from: http://www.gambit-project.org

[14] Karel Durkota; Viliam Lisý; Christopher Kiekintveld; Branislav Bošanský; Michal Pechoucek *Optimal Network Security Hardening Using Attack Graph Games*, Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, **2015**

CAPITOLUL 4

Îmbunătățirea securității rețelelor prin integrarea framework-ului SEC-SDN

4.1 Abordări bazate pe teoria jocurilor în medii SDN pentru apărarea împotriva atacurilor DDoS

Integrarea teoriei jocurilor în mediile Software-Defined Networking (SDN) oferă un mecanism robust pentru îmbunătățirea apărării împotriva atacurilor de tip Distributed Denial of Service (DDoS) prin utilizarea unor strategii dinamice și adaptive. Framework-ul SEC-SDN poate fi consolidat semnificativ prin aplicarea conceptelor teoretice ale jocurilor, cum ar fi Jocul Stackelberg al cursului de acțiune (CoASG) și Echilibrul Stackelberg (SE), care modelează interacțiunea dintre un atacator (urmăritor) și un apărător (lider) [9]. Această abordare permite apărătorului să anticipeze strategiile posibile ale atacatorului și să ajusteze mecanismele de apărare, cum ar fi pragurile de detectare a atacurilor DDoS și acțiunile de atenuare, în mod dinamic.

Modelarea strategiilor atacatorului și apărătorului: Framework-ul CoASG poziționează apărătorul ca lider și atacatorul ca urmăritor. Apărătorul trebuie să prevadă strategiile potențiale ale atacatorului și să își ajusteze mecanismele de apărare în consecință. Implementarea Echilibrului Stackelberg (SE) implică optimizarea strategiei apărătorului, luând în considerare cele mai bune răspunsuri ale atacatorului. Această optimizare se realizează prin ajustarea dinamică a parametrilor din Framework-ul SEC-SDN, cum ar fi mesajele Packet-In și pragurile de utilizare a procesorului, pe baza comportamentului anticipat al atacatorului.

Ajustarea dinamică a pragurilor: Analiza bazată pe teoria jocurilor este utilizată pentru a adapta pragurile de detectare a atacurilor DDoS în Framework-ul SEC-SDN. Aceste praguri sunt flexibile și sunt ajustate în timp real, în funcție de condițiile rețelei și strategiile posibile ale atacatorului.

Optimizarea strategiilor de apărare: Utilizarea unui algoritm de inducție inversă permite determinarea Echilibrului Stackelberg, care ajută la optimizarea sistematică a strategiilor de apărare din Framework-ul SEC-SDN. Aceasta implică evaluarea acțiunilor defensive, cum ar fi blocarea adreselor IP suspecte sau ajustarea regulilor de

flux, pentru a minimiza daunele potențiale ale atacurilor DDoS, luând în considerare costurile asociate cu alarmele false și utilizarea resurselor.

Simulare și evaluare: Înainte de implementare, diferite scenarii de atac trebuie simulate împotriva framework-ului SEC-SDN îmbunătățit pentru a evalua eficacitatea acestuia în atenuarea atacurilor DDoS în timp real. Măsurile bazate pe teoria jocurilor, cum ar fi ratele de succes în prevenirea atacurilor, eficiența utilizării resurselor și acuratețea detecției, sunt utilizate pentru a evalua performanța framework-ului.

Integrarea matematică a teoriei jocurilor în SDN: Pentru a integra abordările bazate pe teoria jocurilor în SDN, se propune un model matematic care aliniază mecanismele strategice de apărare cu configurațiile dinamice ale SDN. Să considerăm că $G_d(a, d)$ reprezintă câștigul apărătorului în modelul bazat pe teoria jocurilor, unde a și d sunt acțiunile atacatorului și apărătorului, respectiv. Având în vedere capacitatea SDN de a reconfigura dinamic resursele rețelei, nivelul general de securitate S al unei rețele bazate pe SDN poate fi reprezentat ca o funcție a câștigului apărătorului și a stării de configurare a rețelei C. Funcția $S(C, d)$, care denotă nivelul de securitate al rețelei, poate fi definită astfel:

$$S(C, d) = max_{d \in D} G_d(resp_0(d), d) \cdot f(C), \tag{4.1}$$

unde:

- $G_d(resp_0(d), d)$ reprezintă câștigul apărătorului la Echilibrul Subjocului Perfect (SPE), caracterizat de costul minim.
- $resp_0(d)$ este funcția de răspuns optim al apărătorului, dat fiind strategiile adoptate de atacator.
- $f(C)$ este o funcție care măsoară eficiența stării de configurare a rețelei C în îmbunătățirea securității rețelei.
- D este setul de acțiuni posibile disponibile apărătorului în paradigma SDN.

Funcția $f(C)$ ilustrează capacitatea SDN de a se adapta la amenințări prin modificarea configurațiilor sale, detaliată prin metrici specifice, cum ar fi reducerea suprafeței de atac, creșterea ratei de detectare sau minimizarea impactului atacurilor reușite.

4.2 Tehnologia Software-Defined Networking

SDN reprezintă o schimbare de paradigmă în proiectarea rețelelor prin separarea funcțiilor de control și de redirecționare a traficului în dispozitivele de rețea. Spre deosebire de rețelele tradiționale, unde switch-urile și routerele gestionează atât controlul, cât și

redirecționarea datelor, SDN separă aceste funcții, permițând o administrare mai flexibilă și eficientă a rețelei prin programare software [4]. Arhitectura SDN este alcătuită din trei planuri: planul aplicației, planul de control și planul de date.

Stratul Aplicației include operațiuni de rețea, securitate, echilibrare a încărcării și gestionarea performanței, implementate sub formă de aplicații software. *Stratul de Control* constă în controlere care orchestrează resursele și gestionează topologia rețelei, oferind o interfață programabilă pentru aplicații. *Stratul Infrastructurii* cuprinde switch-uri SDN și protocoale precum OpenFlow, responsabile pentru procesarea datelor, redirecționare și colectarea informațiilor despre starea rețelei.

4.3 OpenFlow

În Framework-ul arhitecturii SDN, un comutator OpenFlow este esențial, fiind alcătuit, de obicei, dintr-un tabel de fluxuri, un canal securizat și protocolul OpenFlow. Spre deosebire de comutatoarele tradiționale, cele OpenFlow sunt dedicate exclusiv sarcinilor de redirecționare a datelor. În contextul SDN, OpenvSwitch (OVS) este utilizat pe scară largă, oferind un comutator virtual bazat pe software, compatibil cu mai multe platforme și care suportă protocolul OpenFlow [4]. OVS funcționează prin interacțiunea cu interfețele de rețea fizice și virtuale, asigurând o redirecționare fluentă a datelor prin intermediul tabelului său de fluxuri.

Când un pachet este recepționat, comutatorul OVS verifică tabelul de fluxuri pentru a găsi o potrivire. Dacă există o potrivire, pachetul este procesat conform instrucțiunilor din tabelul de fluxuri; în caz contrar, acesta este transmis către controler pentru procesare suplimentară. Controlerul determină apoi intrarea corespunzătoare în tabelul de fluxuri și actualizează comutatorul OVS, asigurând o adaptare continuă la condițiile rețelei.

4.4 Controlerul Software-Defined Networking

Controlerul SDN acționează ca un creier central al rețelei, gestionând fluxurile de trafic și politicile dintr-un punct centralizat. Acesta separă planul de control de planul de date, permițând o administrare flexibilă și scalabilă a rețelei [10]. Poziționarea eficientă a controlerului este esențială pentru a minimiza latența, a asigura toleranța la erori și a optimiza performanța rețelei. Proiecte precum Ryu oferă un sistem de operare SDN cu o API bine proiectată pentru gestionarea traficului de rețea, suportând OpenFlow și permițând controlul avansat al traficului și planificarea acestuia [12].

4.5 Sistemul de Detecție a Intruziunilor SNORT

Sistemele de detecție a intruziunilor (IDS), precum Snort, joacă un rol crucial în securitatea rețelelor în Framework-ul arhitecturilor SDN, cum ar fi SEC-SDN. Integrarea Snort IDS presupune generarea automată a regulilor, reducând astfel erorile umane și consumul de resurse, îmbunătățind totodată detecția amenințărilor [13]. Pentru a crește acuratețea detecției și a reduce alarmele false, sunt utilizate rețele neuronale profunde (DNN) pentru detecția colaborativă, formând o strategie de detecție în două straturi [15].

Framework-ul SEC-SDN utilizează, de asemenea, managementul elastic al instanțelor Snort, optimizând distribuția traficului de rețea și timpii de răspuns. Prin integrarea principiilor teoriei controlului, cum ar fi controlerele Proporțional-Integral (PI) și Proporțional-Integral-Derivativ (PID), sistemul ajustează dinamic încărcătura instanțelor Snort pe baza condițiilor în timp real, menținând performanțe ridicate ale detecției [14]. Mai mult, integrarea tehnologiei blockchain susține o rețea colaborativă de detecție a intruziunilor, asigurând partajarea securizată a alertelor și semnăturilor IDS între nodurile distribuite Snort, sporind astfel capacitățile de detecție a amenințărilor [15].

4.6 Framework-ul SEC-SDN

4.6.1 Lucrări conexe

Complexitatea și amploarea tot mai mare a rețelelor moderne impun abordări inovatoare pentru detectarea și răspunsul eficient la amenințări. Arhitecturile tradiționale de rețea sunt adesea limitate în capacitatea lor de a se adapta la noile amenințări de securitate. În acest context, controlul centralizat și infrastructura programabilă oferite de Software-Defined Networking (SDN) constituie un cadru promițător pentru abordarea acestor provocări. Această cercetare explorează integrarea SDN cu strategiile bazate pe teoria jocurilor pentru dezvoltarea unui cadru de securitate a rețelei mai eficient și mai receptiv.

Apariția tehnologiei de virtualizare a transformat semnificativ dezvoltarea rețelelor, necesitând o arhitectură nouă care separă funcțiile de control și de redirecționare [9, 10]. SDN reprezintă acest nou model, structurat în trei straturi distincte: stratul de infrastructură, stratul de control și stratul de aplicații. Prin separarea planului de control de planul de date, SDN oferă o gestionare îmbunătățită, scalabilitate, programabilitate și performanță superioară arhitecturilor tradiționale de rețea [11]. Controlerul SDN menține o imagine de ansamblu asupra activităților din rețea, permițând o gestionare dinamică și un răspuns eficient la amenințări.

Cercetări anterioare [9] au explorat o abordare bazată pe teoria jocurilor pentru securitatea rețelelor, utilizând honeypots și echilibrul Stackelberg (SE) pentru mode-

larea interacțiunilor dintre atacatori și apărători. Perspectivele strategice oferite de teoria jocurilor constituie un cadru solid pentru ghidarea proceselor decizionale în rețelele compatibile cu SDN. Integrarea strategiilor bazate pe teoria jocurilor cu capacitățile SDN permite dezvoltarea unor mecanisme de securitate sofisticate și adaptive. Această sinergie permite ajustări în timp real ale posturii de apărare a rețelei pe baza amenințărilor emergente, sporind astfel rezistența și robustețea sistemului.

Cercetarea se concentrează pe protejarea platformelor cloud OpenStack împotriva atacurilor de tip Distributed Denial of Service (DDoS) de intensitate redusă, folosind SDN. Aceste atacuri sunt deosebit de dificil de detectat din cauza subtilității lor și a complexității mediului cloud. Framework-ul propus utilizează controlul centralizat oferit de SDN pentru a gestiona traficul de rețea mai eficient și include trei componente de bază: preprocesarea achiziției de date, detectarea atacurilor și mecanismele de apărare [3]. Acesta folosește OpenVSwitch pentru monitorizarea fluxurilor de trafic și detectarea potențialului trafic malițios pe baza unor caracteristici specifice. Odată detectat, controlerul central implementează politici pentru blocarea traficului malițios, menținând în același timp disponibilitatea serviciului.

Au fost propuse mai multe cadre de securitate SDN pentru combaterea atacurilor DDoS, fiecare adoptând metode diferite de detectare și atenuare. De exemplu, Dao et al. [5] propun o metodă de monitorizare a pachetelor pe baza adreselor IP pentru detectarea atacurilor DDoS, însă aceasta implică un consum ridicat de resurse. Mousavi et al. [6] utilizează entropia pentru detectarea atacurilor, dar această metodă poate fi ocolită prin modificarea adreselor IP de către atacatori. Dharma et al. [7] introduc un colector de fluxuri pentru filtrarea traficului malițios, în timp ce Shoeb et al. [8] utilizează un sistem bazat pe niveluri de încredere pentru protejarea planurilor de control și de date în perioadele de vârf.

Framework-ul SEC-SDN se remarcă prin concentrarea sa asupra detectării și atenuării atacurilor DDoS de tip flooding printr-o analiză detaliată a statisticilor controlerului. Acesta introduce un mecanism dinamic pentru ajustarea pragurilor mesajelor Packet-In, permițând detectarea și atenuarea precisă a atacurilor fără a supraîncărca resursele rețelei. SEC-SDN face, de asemenea, diferențierea între sursa atacurilor provenite de la gazde sau de la comutatoare, permițând o atenuare direcționată care conservă lățimea de bandă și menține calitatea serviciilor în timpul unui atac. Această abordare nuanțată și adaptivă oferă îmbunătățiri semnificative față de cadrele anterioare, prin optimizarea utilizării resurselor și creșterea rezilienței rețelei împotriva amenințărilor DDoS sofisticate.

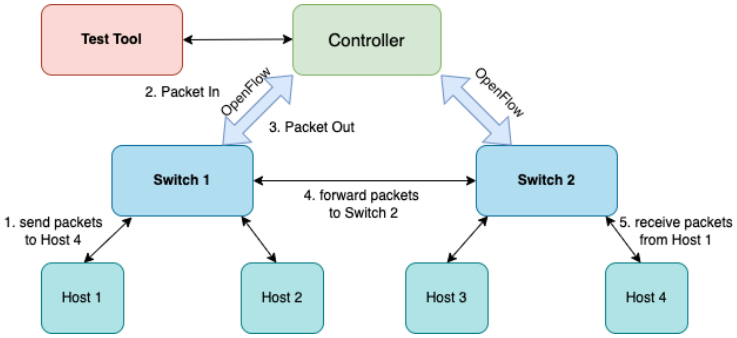


Fig. 4.1 Arhitectura SDN.

4.6.2 Framework-ul propus SEC-SDN

În studiul nostru, presupunem că un actor malițios poate compromite fie o gazdă, fie un comutator SDN pentru a inunda cu trafic o altă gazdă sau controlerul. Dacă o gazdă sau un comutator este infectat cu malware, acesta poate participa la atacuri active sau pasive. Într-un atac pasiv, dispozitivul infectat pare să funcționeze normal, dar colectează informații în secret. Într-un atac activ, dispozitivul perturbă rețeaua, de exemplu, prin lansarea unor atacuri de tip flooding.

În timpul unui atac de tip Denial of Service (DoS), o gazdă compromisă poate inunda comutatorul cu pachete, reducând performanța acestuia și trimițând solicitări false către controler. În mod similar, un comutator compromis ar putea ataca direct controlerul, epuizându-i resursele și împiedicându-l să gestioneze solicitările legitime. Pentru a aborda aceste amenințări, propunem Framework-ul SEC-SDN pentru a apăra împotriva atacurilor de tip Distributed Denial of Service (DDoS).

Deși OpenFlow oferă atât moduri proactive, cât și reactive, studiul nostru se concentrează exclusiv pe aplicațiile reactive.

De exemplu, să considerăm transmiterea datelor de la $Host1$ la $Host4$, așa cum este ilustrat în Figura 4.4. Când $Host1$ trimite un pachet către $Switch1$, comutatorul examinează tabelul său de fluxuri pentru a determina ruta. Dacă nu există o regulă corespunzătoare pentru pachet, $Switch1$ generează o solicitare care încapsulează informațiile pachetului și o transmite către controler. Această solicitare, cunoscută sub denumirea de mesaj $Packet - In$, determină controlerul să evalueze politicile sale și să răspundă cu un mesaj $Packet - Out$ înapoi către comutator. Mesajul $Packet - Out$ conține instrucțiuni pentru gestionarea noului pachet, de obicei prin adăugarea unei noi intrări în tabelul de fluxuri al comutatorului.

Dacă politica de rutare a aplicației SDN imită operațiunile tradiționale de rețea, controlerul ar putea instrui comutatorul să difuzeze pachetul pentru a localiza destinația. Tabelul de fluxuri al fiecărui comutator este actualizat cu noi intrări pe măsură ce răspund la difuzare. Odată ce $Switch1$ primește un răspuns de la $Host4$ prin inter-

mediul $Switch2$, acesta redirecționează pachetul prin portul corect conform tabelului său de fluxuri. Primul pachet parcurge cinci pași până la destinație, însă pachetele ulterioare de la $Host1$ la $Host4$ urmează doar trei pași, ghidate de intrarea de flux predefinită în comutator. Acest proces permite gazdei client să faciliteze interacțiunea dintre planul de control și planul de date, în ciuda incapacității sale de a comunica direct cu planul de control. În scenariile în care există mai multe comutatoare între sursă și destinație, un nou pachet declanșează mai multe mesaje $Packet - In$ în rețea, pe măsură ce fiecare comutator intermediar contactează controlerul pentru instrucțiuni de rutare.

SEC-SDN este un cadru SDN conceput pentru a detecta atacurile de tip flooding DDoS prin analiza statisticilor colectate de controler. Acesta include funcții esențiale de rutare care gestionează pachetele într-un mod similar cu un router tradițional. Scopul nostru este de a dezvolta un model capabil să identifice cu precizie sursa unui atac, fie că provine de la o gazdă sau un comutator, și să se asigure că strategia de atenuare nu consumă excesiv resursele rețelei, cum ar fi lățimea de bandă.

În SEC-SDN, mecanismul de detectare se concentrează în principal pe mesajele $Packet - In$, evaluând atât volumul mesajelor $Packet - In$ primite, cât și capacitatea de procesare a pachetelor de către controler.

Definiții ale parametrilor SEC-SDN:

i: Numerele porturilor controlerului, unde $i = 1, 2, 3, \ldots, n$;

j: Numerele porturilor comutatorului, unde $j = 1, 2, 3, \ldots, m$;

T: Perioada de resetare a pragului;

t: Perioada de monitorizare a contoarelor;

t': Intervalul de timp pentru compararea pragului și a contoarelor în timpul unui atac DDoS;

t_h: Timpul înainte ca comanda de eliminare să fie ștearsă din comutator;

t_d: Timpul înainte ca comanda de eliminare să fie eliminată dintr-un port specific;

γ_{ij}: Numărul de mesaje $Packet - In$ primite de la portul i al controlerului și portul j al comutatorului într-o perioadă t definită;

α_i: Pragul de detecție DDoS la portul i al controlerului;

α'_i: Pragul pentru portul i al controlerului;

β'_i: Numărul de mesaje $Packet - In$ primite pe un port în t' în timpul unui atac;

σ: Utilizarea CPU-ului;

ω: Numărul maxim de mesaje $Packet - In$ într-un interval t când $\sigma\%$ este utilizarea CPU-ului;

λ_i: Numărul de mesaje $Packet - In$ primite pe portul i în timpul T calculat;

τ_i: Pragul candidat pentru portul i al controlerului.

Înainte de implementarea SEC-SDN într-un mediu SDN, administratorii trebuie să configureze pragurile sau parametrii prestabiliți, precum t și ω. Aceste valori de-

pind de performanța hardware-ului și de cerințele specifice ale utilizatorului. Pentru rețele care gestionează volume mari de trafic, valori mai mari pentru ω și σ pot fi necesare. Odată configurată inițial, SEC-SDN poate ajusta independent pragurile pentru detectarea atacurilor DDoS pe baza statisticilor rețelei.

Pe lângă funcțiile sale de rutare, SEC-SDN include în principal funcții pentru actualizarea pragurilor, monitorizare, detectare și apărare:

- **Funcția de actualizare a pragurilor:** Stabilește pragul pentru funcția de monitorizare, calculat pe baza datelor colectate.
- **Funcția de monitorizare:** Colectează statistici și compară contoarele în timp real cu pragul stabilit, alertând fie funcția de actualizare a pragurilor în condiții normale, fie funcția de detectare în cazul anomaliilor.
- **Funcția de detectare:** Evaluează dacă are loc un atac DDoS și activează funcția de apărare pentru a atenua traficul malițios.
- **Funcția de apărare:** Inițiază măsuri de contracarare în timpul unui atac DDoS și resetează pragul după finalizarea atacului.

4.6.3 Funcția de stabilire a pragului

SEC-SDN utilizează diverse contoare pentru a distinge între traficul normal și cel suspect. Pentru a diferenția creșterile legitime de trafic de atacuri, SEC-SDN ajustează pragurile pe baza datelor istorice. De exemplu, dacă un port de comutator înregistrează o creștere neașteptată a solicitărilor, dar acestea rămân în intervalul normal, SEC-SDN actualizează pragul portului corespunzător, evitând astfel clasificarea eronată a acestui trafic ca atac.

În condiții normale, sistemul monitorizează numărul de mesaje $Packet - In$ primite de fiecare port al controlerului i utilizând două metrici: β_i și α_i. La fiecare interval t, β_i este comparat cu α_i, iar α_i este utilizat pentru a stabili un prag potențial τ_i la fiecare interval T. Aceste contoare nu sunt utilizate în timpul atacurilor DDoS, deoarece datele colectate în astfel de incidente nu reflectă cu exactitate comportamentul tipic al rețelei.

În timpul unui atac DDoS, sistemul folosește o altă metrică, β_i', pentru a număra mesajele $Packet - In$ pentru fiecare port al controlerului i, comparând această valoare cu α_i' la fiecare interval ajustat t' pentru a determina dacă atacul s-a încheiat. Deși solicitările malițioase nu sunt procesate, controlerul continuă să monitorizeze interfața sudică care conectează comutatorul la controler.

După ce atacul se termină, pragul α_i este resetat la valoarea sa implicită, reluând procesul de actualizare a pragului. Această resetare este necesară, deoarece datele colectate în timpul atacurilor DDoS nu sunt fiabile pentru referințe viitoare.

4.6.4 Funcția de monitorizare

Funcția de monitorizare reprezintă componenta centrală a SEC-SDN, colectând continuu statistici de rețea și comparându-le cu pragurile predefinite. Această comparație continuă ajută la determinarea necesității activării altor module. În mod specific, SEC-SDN monitorizează fiecare port de pe comutatoarele conectate direct la controler. Funcția de monitorizare urmărește numărul de mesaje $Packet - In$ primite într-o anumită perioadă pentru a evalua starea rețelei. Dacă rețeaua este stabilă și lipsită de fluctuații neobișnuite de trafic, funcția operează fără a activa alte module SEC-SDN. Cu toate acestea, dacă detectează trafic anormal, declanșează fie funcția de actualizare a pragului, fie funcția de detectare a DDoS.

Pentru a realiza acest lucru, se stabilește o sesiune de ascultare pe conexiunea dintre comutator și controler, permițând analiza tuturor mesajelor $Packet - In$ primite. Aceste mesaje furnizează, de asemenea, informații despre sursă, cum ar fi ID-ul comutatorului și ID-ul portului, care sunt esențiale pentru identificarea dispozitivelor din rețea și localizarea atacurilor DDoS folosind funcția de detectare DDoS. În timpul fazei de monitorizare, SEC-SDN evaluează și înregistrează numărul de mesaje $Packet - In$ primite la fiecare t secunde prin fiecare port al controlerului $(i = 1, 2, \ldots, n)$ și port al switch-ului $(j = 1, 2, \ldots, m)$, stocând aceste date în contorul γ_{ij}, conform următoarei ecuații:

$$\gamma_{ij}(t) = \sum Packet_{ij}(t). \tag{4.2}$$

Numărul total de mesaje $Packet - In$ primite de la un port individual al controlerului, notat cu β_i, este comparat periodic cu pragul său respectiv α_i la intervale de t secunde. Această evaluare este esențială pentru a discerne starea actuală a rețelei, diferențiind între un comportament normal și unul potențial anormal. Calculul lui β_i este dat de următoarea ecuație:

$$\beta_i(t) = \sum_{j=1}^{m} \gamma_{ij}(t). \tag{4.3}$$

Punctul de referință pentru un atac DDoS este definit de valoarea numerică ω, care reprezintă capacitatea de performanță a controlerului. ω este determinată de numărul de mesaje $Packet$ care trebuie primite într-un interval de timp t pentru a crește utilizarea procesorului controlerului la un prag specificat, $\sigma\%$. Calculul lui ω este exprimat astfel:

$$\omega(t) = \sum_{i=1}^{n} \beta_i(t). \tag{4.4}$$

Este crucial să se definească valoarea lui ω înainte de implementarea SEC-SDN, deoarece aceasta depinde de caracteristicile software specifice și de capacitățile hard-

ware ale controlerului. Dacă valoarea unei variabile β_i depășește α_i, dar suma totală a valorilor β rămâne sub ω, SEC-SDN tratează acest eveniment ca o creștere bruscă de trafic, nu ca un atac DDoS, și procedează la actualizarea pragului. Această distincție este esențială pentru funcționarea corectă și eficientă a sistemului.

Această condiție permite sistemului să distingă între creșterile temporare de trafic și atacurile DDoS reale, menținând astfel stabilitatea și performanța rețelei.

$$\text{Dacă } \beta_i(t) > \alpha_i(t)$$

$$\text{și } \sum_{i=1}^{n} \beta_i(t) \leq \omega(t)$$

$$\text{atunci } \alpha_i(t) = \tau_i(T). \tag{4.5}$$

Prin urmare, α_i va fi actualizat pentru a se alinia la τ_i, permițând variații acceptabile în rețea. Funcția de actualizare a pragului ajustează α_i pe baza datelor istorice, având un interval de reîmprospătare T semnificativ mai mare decât perioada de monitorizare t. La fiecare T secunde, este generată o nouă listă de praguri candidate τ_i pentru fiecare port al controlerului, derivată din α_i. Valoarea implicită a lui α_i este calculată prin împărțirea egală a lui ω la numărul de porturi ale controlerului n, conform formulei:

$$\text{valoarea implicită a lui } \alpha_i(t) = \frac{\omega(t)}{n}. \tag{4.6}$$

În consecință, α_i indică limita superioară a pachetelor permise pe un anumit port al controlerului în fiecare interval t. Acesta acționează ca o limită critică ce diferențiază operațiunile normale de activitățile anormale. Pragul candidat τ_i poate fi determinat utilizând următoarea ecuație:

$$\tau_i(T) = \left[\omega(t) \cdot \frac{\lambda_i(T)}{\sum_i^n \lambda_i(T)} \right]. \tag{4.7}$$

Când funcția de monitorizare identifică faptul că numărul de solicitări de la un comutator depășește limita permisă, aceasta activează funcția de detectare DDoS pentru o analiză și inspecție mai detaliată.

4.6.5 Funcția de detectare a atacurilor DDoS

În timpul etapei de apărare împotriva atacurilor DDoS, SEC-SDN aplică măsuri diferite în funcție de sursa atacului, fie că acesta este inițiat de o gazdă sau de un comutator, conform rezultatelor funcției de detectare DoS.

Atac din partea unui host:

1. **Politica de eliminare:** SEC-SDN instalează o politică de eliminare pe comutator, care blochează exclusiv fluxurile provenite de la gazda atacatoare pe portul sursă al comutatorului. Aceasta este o metodă de a elimina pachetele suspecte venite de pe acel port.

2. **Modificarea fluxului:** Această măsură diferă de faza de detectare prin două aspecte principale:

   - Se aplică doar portului comutatorului conectat la gazda atacatoare.
   - **Timp de expirare inert:** Dacă niciun pachet nu se potrivește cu această regulă într-un interval consecutiv de t_d secunde, politica de eliminare este eliminată automat.

Atac din partea unui switch:

1. **Deconectare:** Dacă atacul provine direct de la un comutator și acesta devine incontrolabil, una dintre cele mai evidente contramăsuri este deconectarea acestuia de la controler. Dificultatea constă în restabilirea conexiunii în siguranță.

2. **Monitorizare pasivă fără procesare:** Chiar dacă SEC-SDN continuă să numere mesajele $Packet - In$ venite de la comutatorul infectat, acesta nu procesează acele mesaje. Acestea sunt stocate în β_i' (conform ecuației 4.3), economisind astfel resursele controlerului în timp ce situația este monitorizată.

3. **Compararea pragurilor:** Valoarea β_i' este comparată periodic cu un prag implicit α_i' la fiecare t' secunde pentru a determina dacă atacul s-a încheiat.

4. **Restaurare:** Când β_i' scade la α_i' sau sub acest prag, SEC-SDN consideră că atacul s-a încheiat și inițiază actualizarea pragurilor, resetând valoarea α_i la setul de valori implicite. Prin reînceperea de la zero", acest proces asigură că setările noi nu sunt influențate de cele anterioare, care ar fi putut fi corupte în timpul atacului DDoS. Ambele metode se opresc independent atunci când rețeaua revine la starea normală, iar pragurile sunt resetate după atac. Prin acest design, SEC-SDN contribuie la gestionarea atacurilor DDoS și la restaurarea configurațiilor rețelei fără nicio intervenție ulterioară. Cu alte cuvinte, SEC-SDN conferă rețelei proprietăți automate de auto-vindecare în cazul unui atac DDoS.

4.7 Implementarea și evaluarea framework-ului SEC-SDN

4.7.1 Modelarea strategiilor atacatorului și apărătorului

Vedem atacurile de tip Distributed Denial of Service (DDoS) ca pe un joc evolutiv între atacator și administratorul rețelei. Scopul atacatorului este de a perturba infrastructura

critică prin inundarea acesteia cu trafic excesiv, utilizând boți situați atât în interiorul, cât și în exteriorul rețelei țintă. Deși un sistem de detectare a intruziunilor (IDS) poate identifica și bloca unii dintre acești boți prin potrivirea semnăturilor, se recunoaște că botnet-urile DDoS moderne operează cu un grad ridicat de discreție.

Propunem un model bazat pe teoria jocurilor, destinat să detecteze complet botnet-ul și să aplice limitări asupra traficului provenit din surse malițioase. Acest model utilizează conceptul de recompensă și pedeapsă, o strategie frecvent utilizată în teoria jocurilor pentru a încuraja cooperarea între participanți. În acest context, mecanismul este adaptat pentru a stimula conformitatea și a descuraja acțiunile malițioase. Prin impunerea unor penalizări agenților sau utilizatorilor care manifestă un comportament dăunător, modelul urmărește menținerea unui mediu de rețea echilibrat și securizat.

Framework-ul nostru este conceput ca un joc dinamic multi-jucător, în care un singur administrator de rețea (reprezentat în acest context de controlerul SDN) se confruntă cu mai mulți adversari, inclusiv atacatori. Tacticile administratorului sunt implementate prin aplicarea regulilor OpenFlow.

Forma extinsă a arborelui de decizie pentru acest scenariu este prezentată în Figura 4.2, conform matricei de recompense calculate în lucrarea [9].

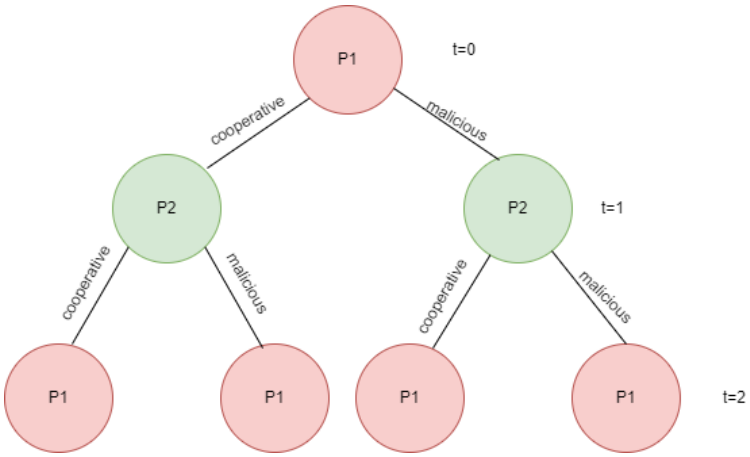


Fig. 4.2 Joc dinamic.

4.7.2 Optimizarea strategiilor de apărare în SEC-SDN

În acest scenariu B simbolizează lățimea de bandă. Prin analiza matricei de recompense, putem determina lățimea de bandă medie la finalul perioadei $t = 2$, care este $\frac{B}{2}$, cu condiția ca utilizatorul să colaboreze pe parcursul perioadelor $t = \{0, 1\}$. Pe de altă parte, dacă $Atacatorul$ adoptă un comportament malițios, acesta obține inițial o lățime de bandă de:

$$\frac{3B}{4} \text{ la } t = 1. \tag{4.8}$$

Ca răspuns, administratorul $Apărătorul$ impune o penalizare la $t = 1$, ceea ce duce la o reducere a lățimii de bandă (B_r) la:

$$B_r = \frac{B}{5}, \tag{4.9}$$

pentru perioada următoare. Aceasta rezultă într-o lățime de bandă medie (B_a) de:

$$B_a = \frac{1}{2} \times (\frac{3B}{4} + \frac{B}{5}) = 0.475B. \tag{4.10}$$

care este inferioară valorii de $0.75B$ care ar fi fost atinsă dacă $Atacatorul$ ar fi continuat să colaboreze. Măsurile punitive ale administratorului ca răspuns la un atac sunt reprezentate de traseul marcat cu roșu. Pe termen lung, adversarul este motivat să mențină un comportament non-malițios dacă este implementat un mecanism de limitare a ratei, conform descrierii din *Algoritmul 1*.

Algorithm 1 Dynamic Game Between Network Administrator and Attackers

Initialize game tree with players P_1 (attacker) and P_2 (administrator)
$B \leftarrow$ Initial bandwidth
for $t = 0$ to End **do**
    if P_1 cooperates **then**
        $B_i \leftarrow \frac{3B}{4}$ ▷ Bandwidth allocated if P_1 cooperates at $t = 1$
        Update P_2's strategy to maintain normal flow
    else ▷ P_1 defects
        $B_i \leftarrow \frac{3B}{4}$ at $t = 1$
        P_2 punishes P_1 at $t = 1$, resulting in $B_r = \frac{B}{5}$
        $B_a \leftarrow \frac{1}{2} \times \left(\frac{3B}{4} + \frac{B}{5}\right)$
    end if
    if punishment period is over **then**
        Reset B_i to default rate for cooperative behavior
    end if
end for
Evaluate long-term outcome for P_1 and adjust strategy accordingly

Fig. 4.3 Algoritmul 1: Joc dinamic între administratorul rețelei și atacatori.

Arhitectura sistemului, bazată pe o platformă Software-Defined Networking (SDN) cu Ryu, utilizează API-uri sudice pentru a interacționa cu elementele din planul de date. Comutatoarele compatibile cu OpenFlow permit interacțiuni atât cu gazdele din rețea, cât și cu cele externe. Unele gazde generează trafic obișnuit către comutator, în timp ce altele fac parte dintr-un botnet DDoS.

Framework-ul controlerului SDN integrează mai multe componente, inclusiv un manager de topologie care supraveghează ajustările de topologie a rețelei și un modul de configurare a rețelei care menține starea curentă a acesteia.

Un sistem de detecție bazat pe semnături, implementat prin Snort, este utilizat pentru a monitoriza traficul rețelei, clasificându-l ca fiind malițios sau benign. Această informație este transmisă către serviciile de rețea și straturile de orchestrare prin API-uri REST nordice.

4.7.3 Implementarea SEC-SDN și algoritmul de limitare a lățimii de bandă

Ne concentrăm pe trei tipuri specifice de atacuri Distributed Denial of Service (DDoS): SYN-Flood, UDP Flood și ICMP Flood. Când Snort detectează un atac, mecanismul de apărare DDoS actualizează tabelul de fluxuri SDN pentru a impune o limitare a ratei traficului provenit de la sursa atacatoare către o destinație din rețeaua internă.

Figura 4.4 prezintă intrările dintr-un tabel de fluxuri SDN. Câmpul de potrivire identifică porturile de intrare și antetele pachetelor. În acest caz, Snort declanșează o alertă pentru adresa IP 10.0.0.1, marcată ca IP-ul atacatorului. Câmpul de instrucțiuni din tabelul de fluxuri este actualizat în consecință. Limitarea ratei, determinată de algoritm, este setată în subcâmpul Rate din câmpul Band. Conform Figurii 4.4, după expirarea perioadei de pedeapsă definită de administrator pentru această adresă IP, subcâmpul Rate revine la rata implicită de trafic. REST API asigură actualizarea continuă a acestor setări în controlerul ODL.

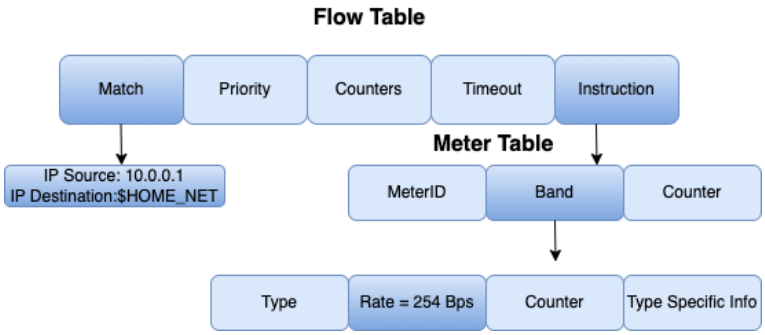


Fig. 4.4 Limitarea traficului în SDN.

Tabelul nu este, în mod implicit, o parte obligatorie a unui tabel de fluxuri. O gazdă care se comportă normal își menține lățimea de bandă obișnuită fără nicio restricție. Cu toate acestea, dacă o gazdă desfășoară acțiuni malițioase și un eveniment, cum ar fi o alertă Snort IDS, declanșează contorul cu ID 1, lățimea de bandă va fi modificată pe baza unei politici de limitare a ratei. Această modificare depinde de respectarea sau încălcarea politicii de către gazda malițioasă în perioada curentă și în cele următoare.

Bibliografie

[1] Neves, R., H., Silva, A.A.A., Gava, V., Azevedo, M., T., Sandoval, J. F.R., et.al - *DoS Attack on SDN: a study on control plane strategies in-band and out-of-band*, Springer Nature, **2022**.

[2] Ali Abedi, Andrew Heard, and Tim Brecht - *Conducting repeatable experiments and fair comparisons using 802.11n mimo networks.* **2015**.

[3] Razvan Florea; Mitica Craus - *Modeling an Enterprise Environment for Testing Openstack Cloud Platform against Low-Rate DDoS Attacks*, 26th International Conference on System Theory, Control and Computing (ICSTCC), **2022**

[4] Arbettu, R. K., Khondoker, R., Bayarou, K., and Weber, F. - *Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers.*, 17th International telecommunications network strategy and planning symposium (Networks) (pp. 37-44). IEEE. (PIC SandT), **2016**.

[5] N. N. Dao, J. Park, M. Park, and S. Cho, "A feasible method to combat against DDoS attack in SDN network," in Conference on Information Networking, pp. 309–311, **2015**

[6] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in Conference on Computing, Networking and Communications, pp. 77–81, **2017**

[7] N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung, and D. Choi, "Time-based DDoS detection and mitigation for SDN controller," in Network Operations and Mag. Symp., pp. 550–553, **2015**

[8] A. Shoeb and T. Chithralekha, "Resource management of switches and controller during saturation time to avoid DDoS in SDN," in IEEE Conference on Engineering and Technology, pp. 152–157, **2016**

[9] Florea, R.; Craus, M. A Game-Theoretic Approach for Network Security Using Honeypots. Future Internet **2022**, 14, 362. https://doi.org/10.3390/fi14120362

[10] Paliwal, M., Shrimankar, D., Tembhurne, O. Controllers in SDN: A Review Report. IEEE Access, 6, 36256-36270 **2018**

[11] Zhu, L., Karim, M., Sharif, K., Xu, C., Li, F., Du, X., Guizani, M. SDN Controllers. ACM Computing Surveys (CSUR), 53, 1 - 40. **2020**

[12] Das, T., Sridharan, V., Gurusamy, M. A Survey on Controller Placement in SDN. IEEE Communications Surveys and Tutorials, 22, 472-503 **2020**

[13] Jaw, E., Wang, X. A novel hybrid-based approach of snort automatic rule generator and security event correlation (SARG-SEC). PeerJ Computer Science, 8 **2022**

[14] Akhtar, N., Matta, I., Raza, A., Wang, Y. EL-SEC: ELastic management of security applications on virtualized infrastructure. IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 778-783 **2018**

[15] Ujjan, R., Pervez, Z., Dahal, K. Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN. 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 1-8 **2019**

CAPITOLUL 5

Rezultate și Analiză

5.1 Evaluarea rezilienței rețelei împotriva atacurilor DDoS

Am evaluat abordarea noastră prin două experimente utilizând un simulator de rețea și controlerul Ryu pe un sistem de operare Ubuntu 22.04. Primul experiment a folosit Algoritmul 1 pentru a atenua atacurile de tip ICMP flood. În al doilea experiment, am utilizat același algoritm pentru a atenua atacurile de tip TCP SYN flood și UDP flood într-o topologie de rețea fat-tree. Au fost alese diferite topologii pentru aceste experimente, în vederea evaluării aplicabilității generale a soluției noastre propuse.

5.1.1 Atacuri DDoS de tip ICMP Flood într-o topologie liniară

Am realizat inițial un experiment cu 50-500 gazde pentru a configura o topologie liniară în Mininet. Aceasta avea un singur strat de gazde conectate la un comutator, așa cum se vede în Figura 5.1. Acest experiment nu este o soluție universală, dar demonstrează că un trafic ICMP masiv cu dimensiuni mari ale pachetelor poate fi generat către o gazdă din rețea prin lansarea unei sesiuni shell pentru fiecare gazdă, utilizând Python multiprocessing. Captura pachetelor a fost redirecționată către un port dummy, iar atunci când sistemul de detecție a intruziunilor (IDS) a identificat o semnătură de atac ICMP flood, acesta a transmis informația către controlerul Ryu.

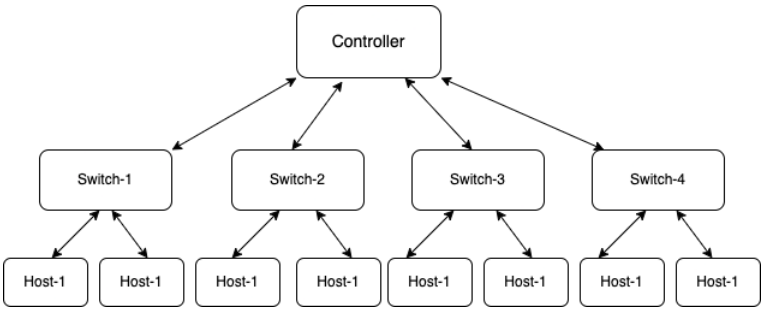


Fig. 5.1 Topologia de testare în simulare.

Aplicația Ryu pentru atenuarea atacurilor DDoS reduce progresiv rata traficului printr-un factor multiplicativ, δ, până când media traficului pe termen lung se aliniază

cu modelele normale de trafic ale unei gazde specificate. În acest studiu, factorul de amortizare δ a fost stabilit la 0.8. Spre deosebire de metodele care blochează complet traficul sau impun o limită statică a ratei, această tehnică reduce treptat debitul gazdelor atacatoare, menținând în același timp calitatea serviciului pentru utilizatorii legitimi.

Table 5.1 Parametrii algoritmului SEC-SDN utilizat în simulare

Gazde Atacatoare	Trafic ICMP Flood (Mb/s)	Trafic ICMP după Limitarea Ratei (Mb/s)
50	38.22	1.23
100	78.12	2.56
200	161.84	5.43
300	240.01	8.02
400	319.21	10.23
500	456.77	15.27

Tabelul prezintă impactul vârfurilor de trafic asupra țintei pentru 100 de gazde, indicând o rată a traficului de 78.12 Mbps în absența mecanismelor de atenuare DDoS. Când IDS activează mecanismul de limitare a ratei (RL), traficul scade semnificativ la 2.56 Mbps, reflectând un factor de reducere de 30. Pe măsură ce numărul gazdelor atacatoare crește de la 100 la 500, debitul traficului DDoS crește de la 78.12 Mbps la 456.77 Mbps, indicând o progresie liniară a traficului de atac. Algoritmul se ajustează dinamic la această creștere, reducând traficul permis pentru 500 de gazde la 15.27 Mbps. Analiza comparativă pentru 500 de gazde arată un factor de reducere de 29 între traficul de atac și traficul limitat de algoritm. Acest studiu confirmă eficacitatea abordării bazate pe teoria jocurilor în atenuarea atacurilor DDoS, demonstrând viabilitatea sa ca măsură de contracarare robustă în rețele extinse.

5.1.2 Atacuri DDoS utilizând TCP/UDP Floods în topologii de rețea Fat-Tree

Un procent semnificativ din atacurile cibernetice care vizează organizațiile sunt direcționate către serverele DNS, inundând sistemele țintă cu volume mari de pachete TCP sau UDP. Având în vedere utilizarea extinsă a topologiilor fat-tree în arhitecturile centrelor de date, studiul nostru evaluează performanța algoritmului nostru într-o topologie fat-tree configurată în Mininet, cu o adâncime și un fanout de 3. În acest experiment, factorul de amortizare δ a fost stabilit la 0.9.

În timpul experimentului, controlerul SDN a stabilit un prag pentru traficul permis TCP și UDP la 3.0 Mbps. Un atac DDoS de tip TCP SYN Flood a fost lansat pe o topologie de rețea formată din 64 de gazde. Impactul algoritmului de limitare a ratei, activat ca măsură de contracarare în urma alertelor IDS, este ilustrat cu roșu în Figura 5.2. Inițial, traficul DDoS a crescut la 156.33 Mbps, depășind semnificativ limita prestabilită. Intervenția controlerului SDN pentru atenuarea atacului a redus cu succes traficul

la 18.11 Mbps la $t = 10$ secunde. Volumul traficului s-a stabilizat în final la 3 Mbps la $t = 50$ secunde, aliniindu-se cu rata normală de trafic autorizată pentru această rețea.

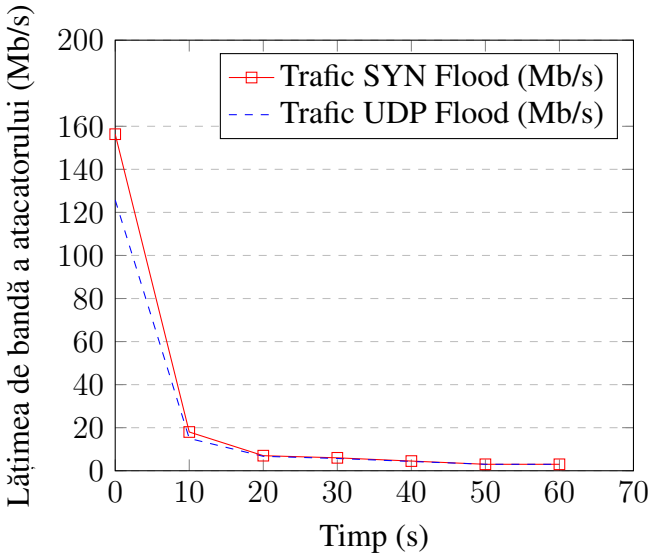


Fig. 5.2 Atenuarea atacurilor UDP și TCP Flood pe o topologie Fat-Tree

5.2 Performanța framework-ului SEC-SDN în arhitectura cloud OpenStack

5.2.1 Integrare și configurare

Integrarea framework-ului SEC-SDN cu OpenStack are scopul de a întări apărarea cloud-ului împotriva atacurilor Distributed Denial of Service (DDoS), utilizând măsuri avansate de securitate bazate pe rețele definite prin software (SDN). Obiectivul principal este asigurarea unei comunicări fluide între Framework-ul SEC-SDN și serviciul Neutron din OpenStack, facilitând monitorizarea eficientă și gestionarea traficului de rețea virtuală. Această integrare urmărește permiterea gestionării în timp real a politicilor de securitate și a răspunsului dinamic la amenințări în mediul cloud.

Integrarea necesită o instalare actualizată a OpenStack, în special versiunea *Caracal, OpenStack 2024.1*, care suportă API-ul Neutron. Framework-ul SEC-SDN, compatibil cu protocolul OpenFlow, trebuie să se integreze fără probleme cu OpenStack Neutron pentru gestionarea rețelei. Resursele hardware adecvate sunt esențiale pentru a susține sarcina suplimentară impusă de SEC-SDN, mai ales în timpul simulărilor de atac DDoS. În plus, infrastructura de rețea trebuie să fie capabilă să gestioneze volume mari de trafic și să ofere izolare pentru scenariile de testare.

Am utilizat o instalare OpenStack configurată cu servicii standard de calcul, stocare și rețea. Versiunea de OpenStack utilizată suportă cele mai noi funcționalități ale API-ului Neutron. Framework-ul SEC-SDN a fost instalat pe o mașină virtuală dedicată

în mediul cloud. Aceasta a fost configurată pentru a interacționa cu OpenStack Neutron și infrastructura fizică de rețea.

În OpenStack, am stabilit mai multe rețele de tip tenant, fiecare având parametri de securitate și tipare de trafic distincte, pentru a emula scenarii din lumea reală. Înainte de implementarea SEC-SDN, am evaluat performanța și metricele de securitate ale cloud-ului OpenStack în condiții normale și în timpul unor atacuri simulate de tip DDoS. Folosind instrumente de generare a traficului, am simulat diverse atacuri DDoS, cum ar fi SYN Flood, ICMP Flood și UDP Flood, vizând atât instanțele de mașini virtuale, cât și componentele rețelei din infrastructura OpenStack. La activarea framework-ului SEC-SDN, am măsurat parametri-cheie, inclusiv timpul necesar pentru detectarea atacurilor, eficiența strategiilor de atenuare și impactul asupra traficului legitim.

Am evaluat capacitatea framework-ului de a ajusta dinamic politicile de securitate în timp real ca răspuns la amenințările identificate. Aceasta a inclus configurarea adaptivă a regulilor de flux de rețea și politicilor de securitate administrate prin serviciul OpenStack Neutron.

5.2.2 Configurarea mediului de testare

Mediul de testare a fost proiectat meticulos pentru a reproduce o infrastructură cloud tipică, constând din 100 de gazde virtuale pe platforma OpenStack. Dintre acestea, 90 de gazde emulează operațiuni standard în cloud, rulând aplicații și servicii uzuale, în timp ce celelalte 10 gazde sunt desemnate ca atacatori. Aceste gazde atacatoare sunt utilizate pentru lansarea diverselor atacuri DDoS, în vederea evaluării mecanismelor de apărare ale rețelei.

Pentru testarea SEC-SDN în cloud, am utilizat un număr redus de mașini virtuale, menținând un raport de 1:50 comparativ cu simularea Mininet. Rezultatele acestui test bazat pe cloud au fost apoi comparate cu datele prezentate în Tabelul 5.1.

Gazdele obișnuite sunt configurate uniform pentru a susține operațiuni standard în cloud, fiecare fiind echipată cu 1 vCPU și 2GB RAM și distribuite în mai multe rețele de tip tenant, pentru a reflecta un mediu multi-tenant real. Pe de altă parte, gazdele atacatoare, deși au configurații hardware similare pentru a se integra fără probleme cu traficul normal, sunt echipate cu instrumente pentru generarea de trafic malițios, cum ar fi SYN Flood, ICMP Flood și UDP Flood. Această configurație este concepută meticulos pentru a simula un mediu realist în care atacurile provin din interior, replicând amenințările din interiorul organizațiilor sau sistemele compromise.

În domeniul detecției intruziunilor, Snort este utilizat ca sistem de detecție a intruziunilor (IDS) pentru a analiza traficul de rețea în căutarea semnelor de intruziune și activități anormale, în special cele provenite de la entități ostile. Funcționalitățile Snort sunt utilizate pentru a diferenția între volume mari de trafic obișnuit și potențiale

amenințări de securitate, asigurând că atacurile reale sunt detectate rapid fără a genera alarme false. În paralel cu Snort, monitorizarea și jurnalizarea traficului de rețea sunt realizate pentru a evalua metricele de performanță, cum ar fi volumul traficului, pierderile de pachete și timpii de răspuns. Aceste metrice sunt esențiale pentru evaluarea impactului atacurilor și eficienței strategiilor de atenuare implementate.

Atacurile DDoS sunt simulate la intervale controlate, vizând resurse specifice ale rețelei pentru a evalua reziliența infrastructurii. Răspunsul framework-ului SEC-SDN la aceste atacuri este analizat critic, concentrându-se pe viteza și eficiența detectării și atenuării facilitate de Snort. În plus, capacitatea framework-ului de a menține operațiunile normale pentru gazdele neimplicate în atacuri este testată, asigurând că sistemul poate izola și neutraliza amenințările fără a perturba activitățile obișnuite ale rețelei.

Snort este configurat pe o mașină virtuală dedicată în mediul OpenStack. Această VM, echipată cu 8GB RAM, 4 vCPU-uri și un hard disk de 100GB, funcționează ca server Snort, analizând traficul de rețea. Pentru testare, această configurație oferă resurse suficiente (CPU, memorie și spațiu pe disc) pentru a gestiona traficul de rețea anticipat și datele de jurnalizare. După instalarea Snort, acesta a fost configurat pentru a înțelege arhitectura rețelei și tipurile de trafic monitorizate. Acest proces a implicat configurarea regulilor adecvate pentru identificarea activităților suspecte. Regulile Snort trebuie actualizate periodic pentru a răspunde amenințărilor emergente.

Pentru a permite monitorizarea traficului de rețea în infrastructura OpenStack SDN, este esențial ca Snort să primească un flux duplicat al acestui trafic. Acest obiectiv a fost realizat prin implementarea port mirroring, cunoscut și sub denumirea de Switch Port Analyzer (SPAN), pe comutatoarele virtuale din mediul OpenStack. Componenta Neutron din OpenStack a fost configurată pentru a redirecționa o replică a traficului de rețea de la fiecare comutator virtual către mașina virtuală Snort, permițând astfel o analiză detaliată a traficului.

Framework-ul SEC-SDN automatizează întregul flux de lucru, de la detectarea amenințărilor până la atenuare. Prin integrarea Snort cu controlerul SDN și Neutron, Framework-ul asigură răspunsuri rapide și precise la amenințări, reducând astfel timpul de expunere la potențiale atacuri. Sistemul permite ajustări dinamice ale listei de blocare, pe măsură ce sunt identificate noi amenințări sau sunt corectate alarme false.

5.3 Eficacitatea strategiilor de apărare bazate pe teoria jocurilor și impactul SEC-SDN asupra responsivității rețelei

Pentru a evalua eficiența SEC-SDN, am efectuat teste de performanță asupra rețelei în timpul atacurilor de tip Denial of Service (DoS), comparând scenarii cu și fără implementarea SEC-SDN. În această configurație experimentală, gazda expeditor transmite

constant datagrame TCP către gazda receptor, în timp ce un atacator inițiază atacuri DoS la frecvențe variabile. Frecvența acestor atacuri a fost ajustată prin modificarea parametrului *time-to-transmit* (`'-t'`) în iPerf pentru traficul UDP. Parametrul `'-t'` determină durata transmiterii pachetelor către o destinație specificată, modulând astfel frecvența actualizărilor destinației.

Table 5.2 Frecvența atacurilor DDoS

Frecvență	Durata Transmiterii (secunde)	Numărul de mesaje Packet-In (mesaje/secundă)
Redusă	1	123
Medie	0.1	256
Ridicată	0.001	543

Inițierea unei solicitări *Packet-In* la întâlnirea unei noi destinații permite o modulare precisă a frecvenței atacurilor DDoS. Lățimea de bandă a traficului UDP este menținută constant la 10 Mbps. De exemplu, comanda pentru a executa un atac de frecvență redusă este `"iperf -c 10.0.1.100 -u -t 1 -b 10M"`. Framework-ul nostru experimental include trei frecvențe distincte de atac—redusă, medie și ridicată—pentru a emula atacuri DoS asupra infrastructurii SDN. Tabelul 5.2 oferă definițiile detaliate ale acestor frecvențe de atac, împreună cu numărul de mesaje asociat fiecărei frecvențe. Durata transmisiei este reglată prin parametrul `'-t'` din iPerf, iar numărul de mesaje *Packet-In* este mediat pe 10 iterații, măsurate la nivelul controlerului folosind Wireshark.

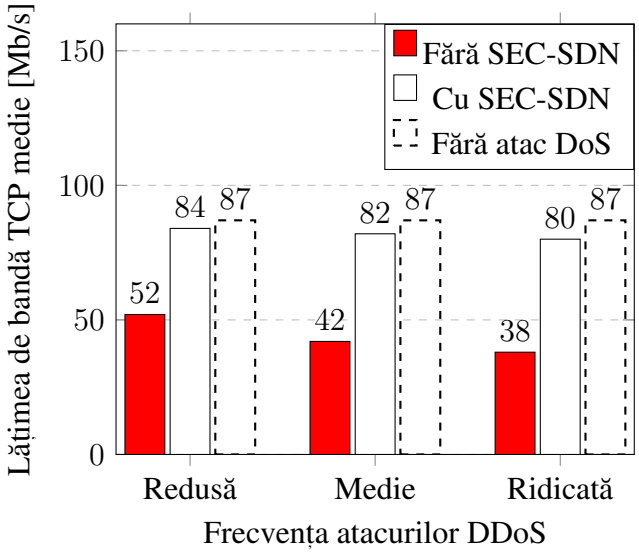


Fig. 5.3 Lățimea de bandă TCP sub atacuri DDoS în cloud-ul SEC-SDN

Pentru a evalua impactul atacurilor DDoS asupra lățimii de bandă disponibile, atât cu, cât și fără implementarea SEC-SDN, am utilizat iPerf pentru a genera trafic

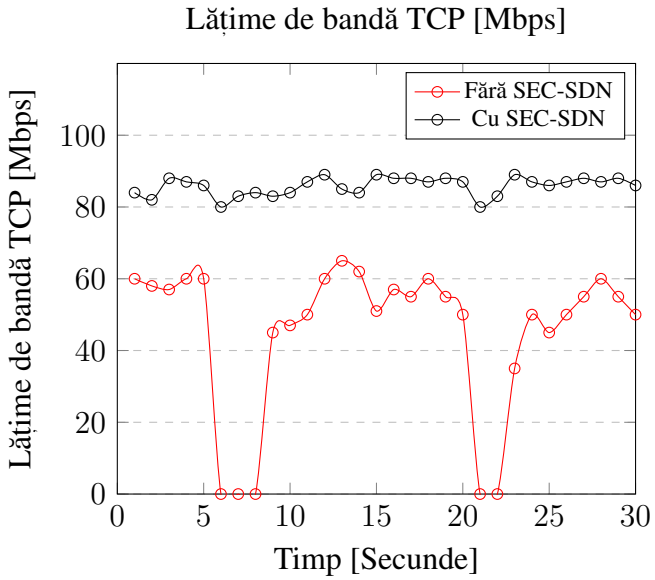


Fig. 5.4 Lățimea de bandă TCP disponibilă instantaneu sub atacuri DDoS rapide în cloud-ul SEC-SDN

TCP între expeditor și receptor, exploatând capacitatea iPerf de a produce fluxuri TCP la lățimea de bandă maximă posibilă. Un prag de 100 a fost stabilit pentru activarea mecanismelor de atenuare ale SEC-SDN la intensități scăzute, medii și ridicate ale atacurilor. Figura 5.3 ilustrează influența atacurilor DoS asupra lățimii de bandă TCP atunci când algoritmul SEC-SDN este activat, comparativ cu situația în care acesta nu este utilizat. Linia neagră din Figura 5.4 ilustrează faptul că implementarea SEC-SDN stabilizează eficient lățimea de bandă instantanee a TCP, asigurând un flux constant și neîntrerupt al traficului de rețea. Acest lucru indică faptul că intervenția SEC-SDN atenuează efectele negative ale atacurilor DoS, prevenind fluctuațiile semnificative ale lățimii de bandă TCP și menținând consistența performanței rețelei.

Implementarea algoritmului SEC-SDN arată că lățimea de bandă TCP rămâne constant ridicată, apropiindu-se de performanța observată în absența atacurilor, cu o retenție a lățimii de bandă între 96.5% și 91.95%. Acest fapt subliniază capacitatea SEC-SDN de a detecta și atenua eficient atacurile DDoS. În plus, validarea cu Wireshark confirmă că controlerul reușește să oprească recepția mesajelor *Packet-In* de la gazda malițioasă.

CAPITOLUL 6

Concluzii și Direcții Viitoare

6.1 Constatări Principale

Peisajul actual al cloud computing necesită cadre avansate și adaptabile de securitate pentru a contracara complexitatea și amploarea tot mai mare a amenințărilor cibernetice. Măsurile tradiționale de securitate devin din ce în ce mai insuficiente pe măsură ce organizațiile migrează servicii și date critice către infrastructuri cloud. Ca răspuns, cadrul *Security Enhanced Cloud Software-Defined Networking (SEC-SDN)* a apărut ca o soluție sofisticată pentru consolidarea securității mediilor cloud. SEC-SDN integrează beneficiile programabilității și gestionării centralizate oferite de Software-Defined Networking (SDN) cu module de securitate avansate, special concepute pentru aplicațiile cloud, având scopul de a asigura o strategie de securitate flexibilă și rezistentă, capabilă să se adapteze dinamic la amenințările emergente. Spre deosebire de aplicațiile SDN convenționale, SEC-SDN reprezintă o schimbare de paradigmă prin integrarea securității direct în nucleul managementului rețelei. Această abordare permite implementarea unor politici de securitate proactive și reactive, potrivite pentru medii cloud caracterizate prin multi-tenancy, virtualizare a resurselor și servicii distribuite.

Conceptul central al cadrului SEC-SDN este controlul centralizat asupra resurselor de rețea, îmbunătățit prin mecanisme suplimentare de securitate. Separarea planului de control de cel de date, specifică SDN, permite orchestrarea centralizată a funcțiilor de rețea, esențială pentru nevoile dinamice și scalabile ale mediilor cloud. Prin exploatarea programabilității SDN, SEC-SDN facilitează automatizarea sarcinilor de securitate, cum ar fi monitorizarea traficului, controlul accesului și detectarea amenințărilor, reducând astfel latența și permițând reacții în timp real la amenințările emergente. O capacitate notabilă a acestui cadru este integrarea sistemelor de detecție a intruziunilor (IDS), precum Snort, care sunt încorporate în infrastructura SDN, îmbunătățind detectarea amenințărilor cunoscute și necunoscute, inclusiv a exploit-urilor de tip zero-day.

6.2 Abordarea metodologică pentru evaluarea SEC-SDN

Evaluarea cadrului SEC-SDN a combinat analiza teoretică cu implementarea practică pentru a măsura indicatori-cheie de performanță, precum acuratețea detectării, timpul de răspuns, scalabilitatea sistemului și impactul asupra performanței generale a rețelei. Componenta teoretică a implicat o analiză amplă a literaturii existente privind securitatea în cloud, cu un accent deosebit pe tehnologiile SDN și IDS, în timp ce aspectul practic a fost testat într-un mediu controlat, prin simularea unor atacuri cibernetice pentru a evalua reziliența și adaptabilitatea SEC-SDN în diverse scenarii de amenințare. Această metodologie riguroasă oferă o înțelegere cuprinzătoare a capacităților și limitărilor cadrului SEC-SDN.

6.3 Contribuții la securitatea cloud

Cadrul SEC-SDN aduce îmbunătățiri semnificative în securitatea cloud prin creșterea vizibilității și controlului asupra rețelei, integrarea IDS pentru gestionarea proactivă a amenințărilor, îmbunătățirea rezilienței și adaptabilității, facilitarea scalabilității și oferirea unei abordări holistice asupra securității rețelei. Planul de control centralizat al SEC-SDN permite o vizibilitate unificată asupra rețelei și aplicarea dinamică a politicilor, esențiale pentru detectarea și răspunsul în timp real la anomalii. Integrarea IDS, precum Snort, cu capabilitățile programabile ale SDN permite implementarea și ajustarea dinamică a senzorilor IDS, îmbunătățind astfel precizia detectării și reducând alarmele false. Adaptabilitatea arhitecturii SEC-SDN, care permite reconfigurarea și implementarea contramăsurilor în timp real ca răspuns la amenințările detectate, crește reziliența rețelei atât împotriva amenințărilor cunoscute, cât și a celor necunoscute.

6.4 Limitări și provocări

În ciuda punctelor sale forte, SEC-SDN are anumite limitări comparativ cu cadrele de securitate bazate pe teoria jocurilor, în special în ceea ce privește capacitatea de a modela și anticipa comportamentul adversarilor și optimizarea resurselor. În timp ce cadrele bazate pe teoria jocurilor excelează în anticiparea comportamentului atacatorilor și optimizarea strategiilor defensive prin simulări, SEC-SDN rămâne preponderent reactiv, concentrându-se pe răspunsuri în timp real, mai degrabă decât pe măsuri preventive. În plus, natura centralizată a SEC-SDN introduce riscuri asociate cu punctele unice de eșec, care ar putea compromite securitatea rețelei în cazul unei compromiteri a controlerului central. Aceste limitări sugerează că, deși SEC-SDN oferă un nivel ridicat de securitate, există loc pentru îmbunătățiri prin adoptarea unor abordări mai descentralizate sau predictive.

6.5 Direcții viitoare de cercetare

Cercetările viitoare ar trebui să exploreze integrarea în SEC-SDN a tehnologiilor de machine learning și inteligență artificială (AI) pentru a îmbunătăți capacitățile de detectare, permițând identificarea mai eficientă a amenințărilor noi și emergente. De asemenea, integrarea tehnologiei blockchain ar putea consolida integritatea și autenticitatea datelor în cadrul SEC-SDN, sporind garanțiile de securitate. Alte provocări practice, precum interoperabilitatea cu sistemele de securitate existente, gestionarea alarmelor false și minimizarea impactului asupra performanței în medii la scară largă, necesită investigații suplimentare. Abordarea acestor provocări va fi esențială pentru adoptarea pe scară largă a SEC-SDN în infrastructurile cloud comerciale.

6.6 Concluzii

Cadrul SEC-SDN reprezintă un progres semnificativ în securitatea cloud, oferind o soluție scalabilă, eficientă și adaptabilă pentru protejarea mediilor cloud împotriva amenințărilor cibernetice sofisticate, inclusiv a atacurilor Distributed Denial of Service (DDoS). Prin integrarea flexibilității și controlului oferite de SDN cu mecanisme avansate de securitate, precum monitorizarea în timp real și integrarea IDS, SEC-SDN furnizează o strategie de apărare stratificată, care îmbunătățește atât securitatea, cât și performanța rețelei. Constatările studiului sugerează că SEC-SDN are potențialul de a deveni un standard în proiectarea și operarea infrastructurilor cloud securizate și reziliente, oferind atât perspective academice, cât și recomandări practice pentru specialiștii din industrie care doresc să își îmbunătățească capabilitățile de securitate în cloud.

LISTA PUBLICAȚIILOR

1. Florea Răzvan and Craus Mitică, ”**A Game-Theoretic Approach for Network Security Using Honeypots**”, *MDPI - Future Internet*, 2022 [Q2]

2. Florea Răzvan and Craus Mitică, ”**Modeling an Enterprise Environment for Testing Openstack Cloud Platform against Low-Rate DDoS Attacks**”, *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, 2022

3. Florea Răzvan and Craus Mitică, ”**Enhancing Network Security Through Integration of Game Theory in Software-Defined Networking Framework**”, *Springer Nature - International Journal of Information Security*, 2025 [Q2].

LISTA FIGURILOR

LISTA TABELELOR