Sintetični simulator na osnovi raziskovalnih podatkovnih zbirk za kibernetsko varnost

Damijan Novak, Jani Dugonik

Inštitut za informatiko, Univerza v Mariboru, Koroška Cesta 46, 2000 Maribor, Slovenija
damijan.novak@um.si, jani.dugonik@um.si

Synthetic Simulator Based on Research Datasets for Cybersecurity

Abstract. *One of the challenges in the cybersecurity domain is the static nature and absence of up-to-date datasets, which significantly limit the ability of artificial intelligence techniques to learn, understand, and protect against the latest cyberattacks (i.e., the ability to simulate and analyze modern attack patterns is vital). This article proposes a synthetic simulator that integrates existing cybersecurity research datasets with dynamic agent-based environments inspired by real-time strategy games. Experimental validation using a ransomware dataset demonstrates that data-driven agents achieve 42% higher success rates than baseline random approaches, proving the practical value of transforming static research data into dynamic, trainable cybersecurity simulation platforms for enhanced threat detection and response capabilities.*

1 Uvod

V medijih pogosto zasledimo novice o vdorih v informacijske sisteme, kraji osebnih podatkov ali zlorabi digitalnih identitet. V preteklosti se je kibernetska varnost (angl. *cybersecurity*) pogosto enačila z varnostjo informacijsko-komunikacijske tehnologije (IKT), kar pa ne zajema vseh vidikov sodobnih groženj. Danes se kibernetska varnost ob prodoru umetne inteligence (angl. *Artificial Intelligence*, AI) sooča s preporodom, ki prinaša nove priložnosti za zaščito, hkrati pa tudi številne izzive [1], med katerimi so tako tehnični kot organizacijski in človeški dejavniki.

Preden se spustimo na tehnični nivo izzivov, je smiselno podati delovno definicijo kibernetske varnosti, saj splošna, enotna definicija (zaenkrat) ne obstaja [2]. V priročniku kibernetske varnosti Urada vlade Republike Slovenije za informacijsko varnost [3] je kibernetska varnost opisana kot praksa zaščite omrežij, sistemov, podatkov in informacijskih tehnologij pred zlorabami (npr. nepooblaščen dostop) ter vključuje tehnološke, procesne in človeške dejavnike skupaj s politiko (angl. *policy*). Zaradi vpliva teh dejavnikov, kibernetske varnosti ni mogoče enačiti zgolj z zaščito IKT.

Na področju kibernetske varnosti lahko najdemo področja, kot so *ribarjenja* (angl. *phishing*), napade onemogočanja, kot je na primer *napad s porazdeljeno zavrnitvijo storitve* (angl. *denial-of-service attack*) z omrežji botov (tj., omrežja okuženih naprav), okužbe računalnikov (npr. izsiljevalsko programje), vdore v strežnike, izgube gesel, itd.

V raziskavi se osredotočamo na podatkovne zbirke *izsiljevalske programske opreme* (angl. *ransomware*), ki šifrira podatke žrtve z namenom izsiljevanja. Zaradi hitrega razvoja predstavlja velik izziv za raziskovalce. Različni napadi zahtevajo različne pristope, ki so danes predvsem usmerjeni v umetno inteligenco, zlasti v inteligentne agente in agentno modeliranje, kjer se napadalci in branilci obravnavajo kot avtonomni agenti s specifičnimi strategijami in cilji [4].

Osnovni cilj prispevka je prikazati, kako lahko statične zbirke podatkov uporabimo kot osnovo za razvoj dinamičnih (sintetičnih) simulatorjev. Ti omogočajo izvajanje simulacijskih scenarijev, kot je na primer usklajen napad več napadalcev na varnostni sistem, pri čemer lahko analiziramo metrike, kot so uspešnost napada, obseg škode in odzivnost obrambnih mehanizmov. Kot drugi cilj prispevka pa je prikaz tega, kako lahko v teh simulatorjih integracija značilk iz statičnih podatkovnih zbirk izboljša uspešnost inteligentnih agentov v primerjavi z osnovnimi naključnimi pristopi.

Struktura članka je naslednja. V drugem poglavju najprej na kratko opišemo teoretično ozadje raziskovalnih področij kibernetske varnosti in njeno povezavo z AI. V tretjem poglavju je predstavljena arhitektura sintetičnega simulatorja ter logistična regresija kot metoda za odločanje znotraj inteligentnega napadalnega agenta. V četrtem poglavju sledi opis eksperimenta. Peto poglavje je namenjeno predstavitvi rezultatov ter krajši diskusiji. V šestem poglavju sledi zaključek in predlogi za prihodnje raziskave.

2 Teoretično ozadje povezave umetne inteligence in kibernetske varnosti

Uporaba AI v kibernetski varnosti se je razvila od osnovne avtomatizacije do kompleksnih algoritmov, ki zaznavajo in preprečujejo grožnje v realnem času. Sprva so raziskave obravnavale preproste *sisteme za zaznavanje vdorov* (angl. *Intrusion Detection Systems*, znane pod kratico IDS), z napredkom AI pa so se razvila podpodročja, kot je strojno učenje [5], ki omogoča prilagajanje novim grožnjam in odkrivanje neznanih napadov.

Sodobni algoritmi strojnega učenja običajno potrebujejo velike količine kakovostnih podatkov za namen treninga in validacije [6]. Sintetični simulatorji lahko pri tem pomagajo, saj omogočajo funkcionalnost dinamičnega okolja, kjer se algoritmi učijo prilagajati

spreminjajočim se pogojem in interakcijam. To je še posebej pomembno za napredne algoritme iz veje okrepitvenega učenja, kjer je ključnega pomena simulacija realističnih scenarijev, osnovanih na kvalitetnih sintetičnih podatkih za optimalno učenje modelov [7].

V praksi, lahko k problemu izsiljevalske programske opreme pristopimo preko statične ali pa dinamične analize [8]. Pri statični analizi se preverja struktura in koda programske opreme (npr. katere programske inštrukcije se izvršujejo), brez da bi jo izvršili. Dinamična analiza pa poteka tako, da v varovanem okolju izvršimo izsiljevalsko programsko opremo in opazujemo njeno delovanje ter učinke.

V zadnjih letih se uveljavlja uporaba generativnih modelov, zlasti *generativnih nasprotniških mrež* (angl. *Generative Adversarial Networks*), za ustvarjanje sintetičnih podatkov napadov [9]. Ti pristopi omogočajo generiranje realističnih vzorcev za učenje modelov zaznavanja groženj, a se soočajo z izzivi kakovosti, ohranjanja značilnosti napadov in tveganja prekomernega prileganja. Naš pristop namesto ustvarjanja podatkov povsem na novo, integrira značilke obstoječih zbirk v dinamično simulacijsko okolje, kar zagotavlja večjo sledljivost, prilagodljivost in varnost.

3 Sintetični simulator za kibernetsko varnost

V nadaljevanju predstavljamo koncept in arhitekturo sintetičnega simulatorja, ki omogoča prehod iz statičnih raziskovalnih podatkovnih zbirk v dinamično simulacijsko okolje. Tak pristop je ključen za naslavljanje omejitev tradicionalnih metod, saj omogoča eksperimentiranje z realističnimi scenariji, preverjanje hipotez in ocenjevanje učinkovitosti različnih strategij v varnem, nadzorovanem okolju. Poglavje se osredotoča na opis arhitekturnih slojev, integracijo podatkovnih značilk ter odločanje agentov, kar predstavlja temelj za nadaljnje empirične analize.

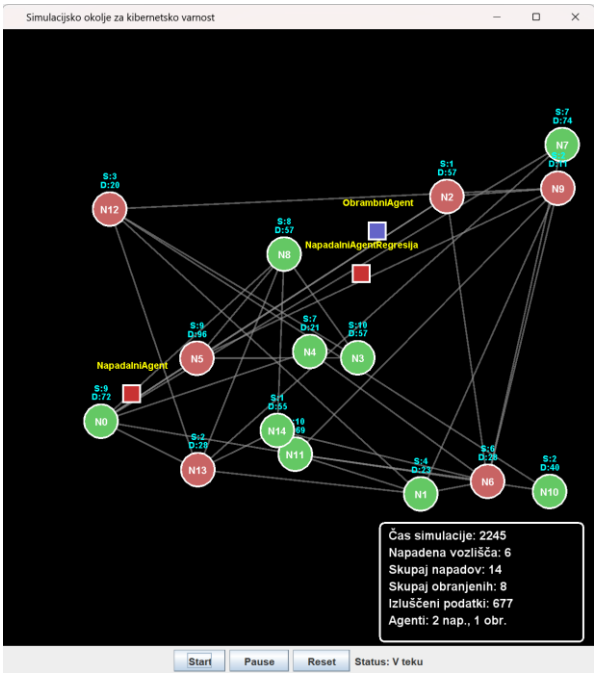
3.1 Arhitektura sintetičnega simulatorja

Simulator je sestavljen iz več komponent, ki skupaj omogočajo izvajanje dinamičnih simulacij kibernetskih napadov (slika 1). Arhitektura je razdeljena na štiri glavne sloje, ki so opisani v nadaljevanju.

1. Sloj za upravljanje podatkov omogoča nalaganje in obdelavo podatkovnih zbirk (tj., datoteke tipa csv). Sistem samodejno prepozna tipe atributov, izračuna normalizacijske vrednosti (min/max) in omogoča ročno filtriranje po značilkah. Implementirani so nastavki, da se več značilk iz podatkovnih zbirk lahko združi v enoten vir podatkov za simulacijo.

2. Simulacijsko okolje modelira omrežje kot graf vozlišč, kjer vsako vozlišče predstavlja sistem z določeno varnostno stopnjo (1-10) (oznaka S na Sliki 1) in vrednostjo podatkov (1-100) (oznaka D na Sliki 1). Vrednost podatkov se določi na podlagi značilk iz podatkovne zbirke *prenosljivih izvršljivih datotek* (angl. *Portable Executable*, PE), pri čemer se strukturne značilke kot so velikost kode, skupna velikost struktur in prisotnost določenih sekcij preslikajo v oceno pomembnosti vozlišča. Ta pristop omogoča direktno preslikavo statičnih značilk iz raziskovalnih podatkovnih zbirk v dinamične atribute simulacijskega omrežja. Če podatki iz zbirke niso na voljo, se uporabi naključna vrednost. Ta vrednost služi kot kazalnik pomembnosti vozlišča in vpliva na odločitve napadalcev ter prioritete branilcev. Agenti (napadalci in branilci) se premikajo med vozlišči, izvajajo napade ali obrambne ukrepe, medtem ko se stanje simulacije sproti beleži za namene raznih analiz.

3. Arhitektura agentov omogoča enostavno dodajanje novih tipov. Napadalni agenti izbirajo cilje glede na nizko varnostno stopnjo in visoko vrednost podatkov, njihovo vedenje pa temelji na hevristikah, inteligentnih rešitvah iz AI ali, kot v našem primeru, na logistični regresiji, ki ocenjuje verjetnost uspešnega napada na posamezno vozlišče. Logistična regresija je bila izbrana zaradi interpretativnosti in nizke računske zahtevnosti. V prihodnje načrtujemo vključitev dodatnih algoritmov, trenutno pa se osredotočamo na integracijo podatkovnih zbirk. Obrambni agenti patruljirajo po omrežju po vnaprej določenih pravilih, arhitektura pa omogoča nadgradnjo obeh vrst agentov z bolj prilagodljivimi modeli.



Slika 1. Primer delovanja simulacijskega okolja za kibernetsko varnost

4. Vizualizacija in nadzor vključuje grafični vmesnik, ki prikazuje omrežje, gibanje agentov in ključne metrike (npr. število napadenih ter ubranjenih vozlišč). Uporabnik lahko nadzoruje potek simulacije (začetek, pavza, ponastavitev) ter izvozi rezultate za nadaljnjo analizo. Povečanje števila vozlišč omogoča modeliranje

bolj kompleksnih in realističnih omrežnih topologij, vendar hkrati povečuje računsko zahtevnost simulacije (npr. večje število interakcij med agenti in daljši čas izvajanja). Arhitektura simulatorja je zasnovana tako, da podpira skaliranje na večje omrežje, pri čemer je možno dodajati tudi nove tipe agentov, če scenarij to zahteva.

Potrebno je še omeniti, da gre za prvo iteracijo simulatorja. Ta še ne predstavlja popolne kopije (ena proti ena) izsiljevalske programske opreme v realnem okolju, temveč njen čim boljši približek, zasnovan na razpoložljivih podatkovnih zbirkah.

3.2 Uporaba logistične regresije kot metode delovanja napadalnega agenta

Za napadalne agente je implementirana logistična regresija, ki ocenjuje verjetnost uspešnega napada na posamezno vozlišče. Model je treniran na statičnih podatkovnih zbirkah, uteži pa se ob inicializaciji naložijo iz konfiguracijske datoteke. Med simulacijo agent uporablja cenitveno komponento, ki na podlagi linearne kombinacije značilk in uteži izračuna verjetnost napada (pretvorjeno s sigmoidno funkcijo). Na podlagi teh ocen agent izbira cilje, pri čemer dodatni dejavniki, kot so predhodni napadi ali nedavni obiski, vplivajo na končno odločitev.

Značilke, ki jih model uporablja, so trenutno v kodi ročno zapisane. V prihodnosti je tudi dodatno predvidena razširitev sistema, ki bo omogočala samodejno izbiro značilk na podlagi konfiguracije ali analize pomembnosti.

4 Opis eksperimenta

Za izvedbo eksperimenta smo uporabili osebni računalnik z Intelovim procesorjem i7-9700, 32 GB pomnilnika, ter naloženim operacijskim sistemom Windows 11 Pro. Simulator je bil v celoti razvit v okviru raziskave in implementiran v programskem jeziku Java (različica 22) z uporabo razvojnega okolja IntelliJ IDEA 2024.2.2 (Community Edition). Arhitektura in funkcionalnosti so zasnovane namensko za prikaz integracije raziskovalnih podatkovnih zbirk v dinamično simulacijsko okolje.

Tekom eksperimentalnega dela je bila kot vir podatkov uporabljena podatkovna zbirka (leto 2024), imenovana »*Ransomware Combined Structural Feature Dataset*« (slov. *zbirka kombiniranih strukturnih značilk izsiljevalske programske opreme*) [10], ki spada na področje raziskav izsiljevalske programske opreme. Zbirka vsebuje 2.675 vzorcev binarnih izvršljivih datotek (2.157 v učni in validacijski množici ter 518 v testni), pri čemer vključuje več družin izsiljevalske programske opreme (25 v učni množici, 15 v testni), kar zagotavlja reprezentativnost in možnost preverjanja generalizacije modela. V okviru našega dela smo uporabili datoteko header.csv, ki vsebuje 80 strukturnih značilk iz zaglavij PE datotek, kot so velikosti pomnilniških rezervacij, parametri zagona in informacije o strukturi datoteke. To zbirko smo izbrali, ker omogoča enostavno preslikavo značilk v simulacijsko okolje in uporabo v regresijskem modelu. V praksi se za raziskave in učenje modelov običajno uporabljajo javno dostopne raziskovalne podatkovne zbirke, saj neposredno zbiranje in izvajanje izsiljevalske kode predstavlja varnostna tveganja in pravne omejitve.

Zavedamo se, da obstajajo tudi druge zbirke, kot je MLRan [11] (vedenjski podatki, 4.800 vzorcev, 64 družin), ki temelji na dinamični analizi in je primerna za raziskave vedenjskih značilk. Vendar naš cilj ni bil analiza dinamičnega vedenja, temveč prikaz izvedljivosti integracije statičnih značilk v sintetični simulator, zato smo izbrali zbirko, ki omogoča neposredno uporabo strukturnih značilk brez potrebe po kompleksni dinamični analizi.

Te značilke se uporabljajo v regresijskem modelu za napoved verjetnosti ogroženosti vozlišča. Model temelji na logistični regresiji z utežmi, pridobljenimi med učenjem in shranjenimi v datoteki lr_weights.json. Izbrane so značilke z največjimi absolutnimi utežmi, med njimi LoaderFlags, DllCharacteristics, SizeOfCode, EXCEPTION_VA in RESOURCE_VA.

Za namen primerjave delovanja in učinkovitosti agentov sta implementirana dva agenta z različnimi strategijami odločanja, in sicer, naključni agent ter napadalni agent z uporabo logistične regresije. Naključni agent predstavlja osnovni primerjalnik, ki izvaja izključno stohastične odločitve brez uporabe podatkovne zbirke. Pri poskusu napada na vozlišče se uporablja naključno določena osnovna verjetnost uspeha, ki smo jo določili na zaprtem intervalu razpona 20-70 %. Vključena pa je tudi preprosta strategija izogibanja zadnjim trem obiskanim vozliščem (tj., da se preprečuje takojšnja povratna gibanja).

Napadalni agent z uporabo logistične regresije (NapadalniAgentRegresija) implementira hibridni pristop odločanja, kjer 85 % odločitev temelji na logistični regresiji z uporabo PE značilk, preostalih 15 % pa predstavljajo naključne izbire za zagotavljanje nepredvidljivosti. Agent uporablja enako osnovno verjetnost uspešnosti napadov ([20, 70] %) kot naključni agent, vendar jo prilagaja na podlagi logističnega rezultata, z dodatno modifikacijo ±20 %. Pri izbiri ciljev kombinira regresijski rezultat z mrežnimi faktorji (varnostna raven vozlišča, vrednost podatkov, nedavni obiski, povezanost), pri čemer izbere vozlišče z najboljšim kombiniranim rezultatom. Oba agenta delujeta v enakih simulacijskih pogojih z istimi začetnimi parametri, kar omogoča neposredno primerjavo vpliva vključevanja podatkovnih zbirk na uspešnost napada.

Simulator tudi omogoča agentom, ki uporabljajo podatkovne zbirke, izločanje značilk, ki bi lahko umetno izboljšale napovedno moč modela (npr. ID, GR, FamilyID). Te so obravnavane kot "*značilke uhajanja*" (angl. *leakage features*) in se izključijo iz obravnave pri zbiranju vseh dostopnih značilk iz podatkovne baze.

Za statistično zanesljivost rezultatov se izvede 50 neodvisnih scenarijev za vsakega agenta, pri čemer vsak scenarij obsega 200 iteracij z možnostjo gibanja agenta med povezanimi vozlišči. Agenti se premikajo z optimizirano hitro strategijo gibanja, kjer se celotna pot do ciljnega vozlišča izračuna vnaprej in izvede v

maksimalno petih korakih, kar zagotavlja odzivnost simulacije brez vpliva na logiko odločanja.

Vsi scenariji uporabljajo mrežno topologijo s 15 vozlišči in naključno generirano povezanostjo (dve do štiri povezave na vozlišče), pri čemer se za vsak scenarij ustvari novo omrežje z različnimi parametri (varnostna raven 1–10, vrednost podatkov 1–100). Agenti začnejo na vozlišču 0. Vsak scenarij beleži metriko uspešnih in neuspešnih poskusov napadov, na podlagi česar se izračunajo povprečna uspešnost, standardni odklon in razpon uspešnosti za statistično analizo.

Namen eksperimenta je preveriti, ali podatkovne zbirke PE datotek izboljšajo uspešnost napadalnih agentov v primerjavi z naključnim odločanjem.

5 Rezultati in diskusija

V tabeli 1 so predstavljeni rezultati eksperimenta.

Tabela 1. Rezultati eksperimentalne primerjave obeh agentov

Metrika	Naključni agent	Napadalni agent	Δ
Povprečna uspešnost	47,99 %	68,23 %	+20,24 %
Razpon uspešnosti	30,0 – 66,7 %	50,00 – 93,75 %	-
Standardni odklon	9,17 %	10,51 %	-
Skupaj poskusov	1.110	1.079	-
Razmerje uspeha	523:587 (0,89:1)	721:358 (2:1)	1:2

Napadalni agent je dosegel povprečno uspešnost 68,23 % v primerjavi z 47,99 % naključnega agenta, kar predstavlja statistično značilno izboljšanje za 20,24 odstotnih točk oziroma 42 % relativno povečanje učinkovitosti. Še bolj izrazita je razlika v razmerju uspešnih proti neuspešnim napadom, kjer napadalni agent dosega dva uspešna napada na vsakega neuspešnega (2:1), medtem ko naključni agent dosega manj kot en uspešen napad na vsakega neuspešnega (0,89:1).

6 Zaključek

Dokazana učinkovitost pristopa nakazuje, da lahko statične podatkovne zbirke služijo kot dragocen vir znanja za dinamične simulacijske sisteme na področju kibernetske varnosti.

V prihodnje bo simulator deležen številnih nadgradenj, kot je na primer bolj robusten sistem izbire značilk, ki ga želimo iz ročnega prenesti v avtomatizirane strojne variante preko uporabe metod, kot so *samodejno učenje značilk* (angl. *feature learning*), *izbira značilk na podlagi pomembnosti* (angl. *feature importance ranking*) ter z evolucijskimi algoritmi za optimizacijo izbora. Cilj je zmanjšati odvisnost od ročnega izbiranja značilk ter povečati učinkovitost in ponovljivost analiz. Nadgradnje bodo zajele tudi simulacijski pogon in mehanike, vključno z razširitvijo na druge tipe napadov, naprednejše upravljanje scenarijev ter shranjevanje zgodovine in ponovni prikaz (angl. *replay*) simulacij.

Zahvala

Raziskovalni program št. P2-0057 je sofinancirala Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije iz državnega proračuna.

Literatura

[1] A. H. Salem, S. M. Azzam, O. E. Emam, A. A. Abohany: Advancing cybersecurity: a comprehensive review of AI-driven detection techniques, Journal of Big Data, zv. 11, št. 1, 105, 2024.

[2] D. Craigen, N. Diakun-Thibault, R. Purse: Defining cybersecurity, Technology innovation management review, zv. 4, št. 10, 2014.

[3] Priročnik kibernetske varnosti urada Republike Slovenije za informacijsko varnost, https://www.gov.si/assets/vladne-sluzbe/URSIV/Datoteke/Prirocnik-kibernetske-varnosti.pdf

[4] G. A. Francia III III, X. P. Francia, C. Bridges: Agent-based modeling of entity behavior in cybersecurity, In Advances in cybersecurity management, Cham: Springer International Publishing, str. 3-18, 2021.

[5] D. Ucci, L. Aniello, R. Baldoni: Survey of machine learning techniques for malware analysis, Computers & Security, zv. 81, str. 123-147, 2019.

[6] N. Gupta, S. Mujumdar, H. Patel, S. Masuda, N. Panwar, S. Bandyopadhyay, et al.: Data quality for machine learning tasks, In Proceedings of the 27th ACM SIGKDD Conference on knowledge discovery & data mining, str. 4040-4041, avgust 2021.

[7] Q. Zhang, X. Fang, K. Xu, W. Zhao, H. Li, D. Zhao: High-quality Synthetic Data is Efficient for Model-based Offline Reinforcement Learning. In 2024 IJCNN, str. 1-7, IEEE, junij 2024.

[8] C. C. Moreira, D. C. Moreira, C. Sales Jr.: A comprehensive analysis combining structural features for detection of new ransomware families. Journal of Information Security and Applications, zv. 81, 103716, 2024.

[9] G. Agrawal, A. Kaur, S. Myneni: A Review of Generative Models in Generating Synthetic Attack Data for Cybersecurity, *Electronics*, zv. 13, št. 2, 322, 2024.

[10] C. C. Moreira, D. C. Moreira in C. Sales Jr., "Ransomware Combined Structural Feature Dataset", Mendeley Data, V1, 2024. doi: 10.17632/yzhcvn7sj5.1.

[11] F. C. Onwuegbuche, A. Olaoluwa, A. D. Jurcut, L. Pasquale: MLRan: A Behavioural Dataset for Ransomware Analysis and Detection, arXiv preprint arXiv:2505.18613, 2025.