Kenkėjiškų programų aptikimo gerinimas taikant kelių klasių gerybinės programinės įrangos analizę

Juozapas Rokas Čypas, Viktor Medvedev, Juozas Dautartas

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Duomenų mokslo ir skaitmeninių technologijų institutas,
Akademijos g. 4, LT-08412, Vilnius, Lietuva
juozapas.cypas@mif.stud.vu.lt

Santrauka. Šiame tyrime siūloma metodika, apimanti gerybinės ir kenkėjiškos programinės įrangos kategorizavimą, siekiant pagerinti kenkėjiškų programų aptikimo tikslumą. Siekiama išanalizuoti, kaip skirtingų programų požymiai padeda atskirti kenkėjišką programinę įrangą nuo gerybinės. Metodika grindžiama statinės analizės duomenimis ir šiuolaikiškais duomenų apdorojimo bei vizualizavimo metodais.

Raktiniai žodžiai: kibernetinis saugumas, gerybinė programinė įranga, kenkėjiškos programinės įrangos aptikimas, programinės įrangos klasifikavimas, gilieji neuroniniai tinklai, mašininis mokymasis.

1 Įvadas

Šiandieninėje skaitmeninėje erdvėje kibernetinio saugumo svarba yra kaip niekad aktuali. Kibernetinių atakų metu kasmet padaroma vis didesnė žala, kuri pastaruoju metu savo dydžiu užima trečios didžiausios pasaulio ekonomikos vietą po JAV ir Kinijos [1]. Į globalius karinius konfliktus įsitraukusios šalys skiria milžiniškas lėšas kibernetiniam karui. Kenksminga programinė įranga (angl. *malware*) technologinio vystymosi akivaizdoje sparčiai tobulėja, tad kibernetinių atakų atpažinimas ir prevencija yra itin svarbus iššūkis. Praktikoje naudojami mašininio mokymosi metodai norint išmokyti modelius atpažinti ir užkirsti kelią kibernetinėms grėsmėms įrenginiuose ar tinkluose.

2 Kenkėjiškų programų analizės metodai

Kenkėjiškos programos, kurios dažniau šiomis dienomis vadinamos virusais, yra analizuojamos dviem metodikom: statine ir dinamine analize. Stati-

DOI: https://doi.org/10.15388/LMITT.2025.3

nės analizės metu tyrėjai gilinasi į tokias failo savybes kaip sekcijų kiekis, failų tipas, kompiliatorius, esami žodžių junginiai faile (angl. *strings*), *Windows API* kvietimai ir pan. Šios analizės metu failas nėra paleidžiamas, siekiama patikrinti jo išorines savybes jo neatidarant, nėra gaunama informacija apie įtartiną failo funkcionalumą. Nors šis analizės būdas turi savo privalumų, jį gana lengva apeiti modifikuojant failo savybes. Todėl siekiant susidaryti pilnesnį vaizdą apie failo funkcionalumą, taip pat yra taikoma dinaminė analizė, kurios metu failas yra paleidžiamas izoliuotoje aplinkoje (angl. *sandbox*) siekiant išnagrinėti paleidžiamo proceso savybes ir veiksmus. Šios analizės metu galime pastebėti tokius dalykus kaip sisteminius kvietimus (angl. *system calls*), į kokius IP adresus bando kreiptis procesas ir pan. Tačiau siekiant atpažinti dar nematytas virusų grėsmes dažnai yra naudojami mašininio mokymosi arba giliojo mokymosi metodai, kuriems apmokyti naudojami statinės ir dinaminės analizės metu surinkti duomenys. Tačiau duomenų rinkiniai, kurie naudojami šių modelių mokymui, neretai būna nesubalansuoti ir kenksmingos programinės įrangos duomenų imtis būna gerokai didesnė nei gerybinės (nekenksmingos, nekenkėjiškos) programinės įrangos (angl. *goodware, benign)*. Kenkėjiškos programinės įrangos kūrėjai dažnai taiko naujas metodikas bei keičia savo įrankių funkcionalumą, todėl duomenų rinkiniai nespėja pakankamai dažnai atsinaujinti ir prisitaikyti prie pokyčių. Tuo tarpu, nekenksmingos programinės įrangos kūrėjai esminio funkcionalumo nekeičia, nes jų tikslai nėra išvengti aptikimo ar apgauti antivirusines sistemas. Todėl šio tyrimo metu bus nagrinėjami nekenksmingos programinės įrangos bruožai, suskirstant nekenksmingą programinę įrangą į kelias dažniausiai pasitaikančias grupes, aprėpiančias dažniausiai naudojamą funkcionalumą. Tyrimo metu taip pat bus bandoma palyginti, kaip šių programų bruožai atsiskiria nuo kenksmingos programinės įrangos. Šis tyrimas yra grindžiamas statinės failų analizės metodu, nes šis metodas įprastai yra pirmas antivirusinių programų atliekamas žingsnis norint nustatyti, ar failas yra pavojingas. Statinė analizė duoda greitus rezultatus, tačiau ją galima apeiti manipuliuojant failo metaduomenimis. Dinaminė analizė, nors ir tikslesnė, reikalauja *sandbox* aplinkos bei daugiau resursų, be to, ne visus failus gali pavykti atidaryti *sandbox* aplinkoje.

3 Duomenų rinkinio paruošimas

Paprastai dauguma virusų klasifikavimo modelių yra mokomi remiantis viešai prieinamais duomenų rinkiniais, tokiais kaip EMBER [2] ar SOREL-20M [3].

Šiuose rinkiniuose gerybiniai failai traktuojami kaip viena klasė, toks supaprastinimas lemia klaidingai teigiamus (angl. *false positive*) rezultatus, kai virusai struktūriškai panašūs į tam tikrą nekenksmingos programinės įrangos grupę. Šiame tyrime siūlome naują požiūrį į kenkėjiškų programų aptikimą, analizuojant ne tik kenkėjiškas programas, bet ir įvairias gerybinės programinės įrangos kategorijas.

Šiame tyrime bus pasiūlytas duomenų rinkinio paruošimo metodas, pagrįstas nekenksmingos programinės įrangos kategorizavimu. Siūlomos kategorijos apima ofiso programas, sisteminius įrankius, žaidimus, vaizdo redagavimo programas ir kt., ir vieną kenkėjiškų programų kategoriją. Naudojant pasiūlytą metodą galima geriau suprasti gerybinės programinės įrangos ypatumus ir identifikuoti anomalijas, kurios gali sufleruoti kenkėjišką elgesį. Naudojant statinę analizę, iš kiekvieno programos failo galima išgauti daugiau nei 2000 požymių įskaitant failo antraštės (angl. *header*) informaciją, sekcijų pasiskirstymą, importuojamas funkcijas, simbolių eilutes, baitų pasiskirstymą. Toks didelis požymių skaičius kelia iššūkių tiek skaičiavimų efektyvumui, tiek modelių generalizacijai apdorojant nematytus duomenis. Siekiant sumažinti duomenų dimensiją ir išlaikyti svarbiausią informaciją, planuojama taikyti įvairius dimensijos mažinimo metodus, tokius kaip pagrindinių komponentų analizė (PCA), t-SNE, UMAP ir autoenkoderiai. Dimensijos mažinimas leidžia ne tik sumažinti duomenų požymių skaičių, bet ir vizualizuoti skirtingų programų klasių pasiskirstymą, leidžiantį pastebėti panašumus tarp tam tikrų kenkėjiškų ir gerybinių programų kategorijų.

4 Tyrimo metodika

Šio tyrimo metu bus tikrinama hipotezė ar pasiūlyta programų kategorizavimo strategija gali pagerinti kenkėjiškų programų aptikimo tikslumą. Yra tyrimų, kurie rodo, kad kai kurių kenkėjiškų programų struktūra bei atliekamos funkcijos, norėdamos išvengti aptikimo, siekia imituoti tam tikrų gerybinių programų požymius [4]. Tokie pastebėjimai pabrėžia būtinybę gilinti gerybinės programinės įrangos analizę ir įtraukti ją į kenkėjiškų programų aptikimo modelių mokymą.

Tyrimo metu bus siekiama apmokyti dirbtiniu intelektu pagrįstus modelius, pavyzdžiui, dirbtinį neuroninį tinklą. Kitas būdas yra pamėginti statinės analizės metu išgautus požymius transformuoti į vaizdus (pavyzdžiui, panaudojant GASF, GADF, MTF, GAFMAT metodus [5]) arba kenkėjiškos pro-

graminės įrangos dvejetainį kodą transformuojant į vaizdą [6]. Tokiais būdais gauti vaizdai gali būti naudojami apmokyti konvoliucinį neuroninį tinklą atskirti kenkėjišką įrangą nuo gerybinės.

Padėka: Finansavimą skyrė Lietuvos mokslo taryba (LMTLT), sutarties Nr. S-MIP-24-116.

Literatūra

[1] Morgan, S. (žiūrėta 2025 m. balandžio 10 d.). Cybercrime to cost the world 8 trillion annually in 2023. https://cybersecurityventures.com/cybercrime-to-cost-the-world-8-trillion-annually-in-2023/

[2] Roth, P., & Anderson, H. S. (2018). EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. https://doi.org/10.48550/arXiv.1804.04637

[3] Harang, R., & Rudd, E. M. (2020). SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection. https://doi.org/10.48550/arXiv.2012.07634

[4] Yin, H., Lou, B., & Reiher, P. (2023). A Method for Summarizing and Classifying Evasive Malware. (p. 455-470). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3607199.3607207

[5] Budžys, A., Kurasova, O., & Medvedev, V. (2024 m. rugpjūčio 29 d.). Deep Learning-Based Authentication for Insider Threat Detection in Critical Infrastructure. Artificial Intelligence Review, 57(10), 1-35. https://doi.org/10.1007/s10462-024-10893-1

[6] Meenpal, T., & Kumar, N. (2019). Texture-Based Malware Family Classification. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (p. 1-6). https://doi.org/10.1109/ICCCNT45670.2019.8944659