Vilniaus universitetas

Mašininiu mokymusi grindžiami metodai apgaulingoms ir obfuskuotoms kenkėjiškoms programoms generuoti stiprinant kibernetinį saugumą

Doktorantūros pradžios/pabaigos metai: 2023–2027
Studijų metai: 2024–2025 pirmas pusmetis

Doktorantas: Juozas Dautartas
Vadovas: Dr. Viktor Medvedev

Studijų planas ir jo vykdymo suvestinė

Vilniaus universitetas

Studijų metai	Egzaminai [1]	
	Planas	Įvykdyta
I (2023/2024)	3	3
II (2024/2025)	**1**	**1**
III (2025/2026)		
IV (2026/2027)		
Iš viso:	4	4

Studijų metai	Dalyvavimas konferencijose				Publikacijos					
	Tarptautinėse [2]		Nacionalinėse [3]		Su citav. rodikliu [4]			Be citav. rodiklio [5]		
	Planas	Įvykdyta	Planas	Įvykdyta	Planas	Įvykdyta [6]	Būklė [7]	Planas	Įvykdyta [6]	Būklė [7]
I (2023/2024)				1						
II (2024/2025)			**1**	**1**						
III (2025/2026)	1				1			1		
IV (2026/2027)	1				1					
Iš viso:	2		1	2	2			1		

Ataskaitinio pusmečio darbo planas ir jo vykdymo suvestinė

Vilniaus universitetas

Egzaminai 2024/2025 (I pusmetis)		
Planas	**Įvykdyta**	**Būklė**
Fundamentalieji informatikos ir informatikos inžinerijos metodai 2025 m. 1 ketvirtis.	Informatikos ir informatikos inžinerijos tyrimo metodai ir metodika 2025 m. sausio 28 d.	Išlaikytas

Dalyvavimas konferencijose 2024/2025 (I pusmetis)		
Planas	**Įvykdyta**	**Konferencijos tipas**
-	DAMSS: 15th conference on data analysis methods for software systems, Druskininkai, Lithuania, November 28-30, 2024 **Red team tactics against malware detection using adversarial attacks.** Dautartas, Juozas; Budžys, Arnoldas; Jomantas, Haroldas; Kurasova, Olga; Medvedev, Viktor.	Nacionalinė

Publikacijos 2024/2025 (I pusmetis)			
Planas	**Įvykdyta**	**Būklė**	**Publikacijos tipas**

Tyrimų objektas

Mašininio mokymosi algoritmai C2 sistemų agentams (kenkėjiškoms programoms) generuoti ir obfuskuoti.

Tikslas

Pasiūlyti ir ištirti naują kenkėjiškos programinės įrangos generavimo metodą, skirtą veiksmingai išnaudoti mašininiu mokymusi pagrįstų kenkėjiškų programų aptikimo sistemų pažeidžiamumą, siekiant padidinti kibernetinių grėsmių atsparumą ir pagerinti bendrą jų aptikimo tikslumą ir patikimumą.

Uždaviniai

- Atlikti išsamią analitinės literatūros apžvalgą, siekiant nustatyti esamus mašininio mokymosi metodus ir būdus, skirtus kenkėjiškoms programoms aptikti, klasifikuoti ir generuoti kenkėjiškas programas.
- Analizuoti antivirusinių ir galinių įrenginių aptikimo ir reagavimo (angl. Endpoint Detection and Responce, EDR) sistemų kenkėjiškos programinės įrangos aptikimo mechanizmus, siekiant suprasti kenkėjiškos programinės įrangos klasifikavimo ypatybes.
- Įvertinti ir išskirti pagrindinius kenkėjiškos programinės įrangos požymius, turinčius įtakos jų aptikimui, siekiant pagerinti kenksmingų programų klasifikavimo tikslumą.
- Sukurti naują arba modifikuoti esamą kenkėjiškų programų (angl. Command and Control framework agents) generavimo metodą, kuriuo siekiama sukurti klaidinančius ir sunkiai aptinkamus C2 sistemų agentų pavyzdžius, galinčius klaidinti mašininiu mokymusi pagrįstus kenkėjiškų programų aptikimo modelius.
- Įvertinti sugeneruotus kenkėjiškų programų pavyzdžius, naudojant įvairius kenkėjiškų programų aptikimo sprendimus ir mašininiu mokymusi grindžiamas kibernetinio saugumo sistemas.
- Ištirti esamas ir pasiūlyti naują strategiją, kaip padidinti mašininio mokymosi modelių atsparumą priešiškoms atakoms.

Per pusmetį gauti rezultatai

- Atnaujinta mokslinės literatūros apžvalga, kurioje nagrinėjami įvairūs kenkėjiškos programinės įrangos aptikimo ir obfuskavimo metodai.
- Tęsiami tyrimo darbai, susiję su mašininio mokymosi metodų taikymu kenkėjiškos programinės įrangos aptikimui ir klasifikavimui, bei priešiškomis atakomis prieš šiuos modelius.
- Išlaikytas „Informatikos ir informatikos inžinerijos tyrimo metodai ir metodika“ dalyko egzaminas; ir dalyvauta nacionalinėje konferencijoje „DAMSS: 15th conference on data analysis methods for software systems“.
- Pasiūlytas metodas kenkėjiškų programų obfuskavimui panaudojant dirbtinio neuroninio tinklo architektūrą.
- Dalyvauta tarptautinėse kibernetinio saugumo pratybose „Gintarinė migla 2024“ raudonosios komandos sudėtyje. Šių pratybų metu buvo gilinamos ir praktiškai pritaikomos žinios susijusios su atliekamų tyrimų tematika ir disertacijoje nagrinėjama problema.

Algorithm 1 Failo išsaugojimas neurono tinklo parametruose

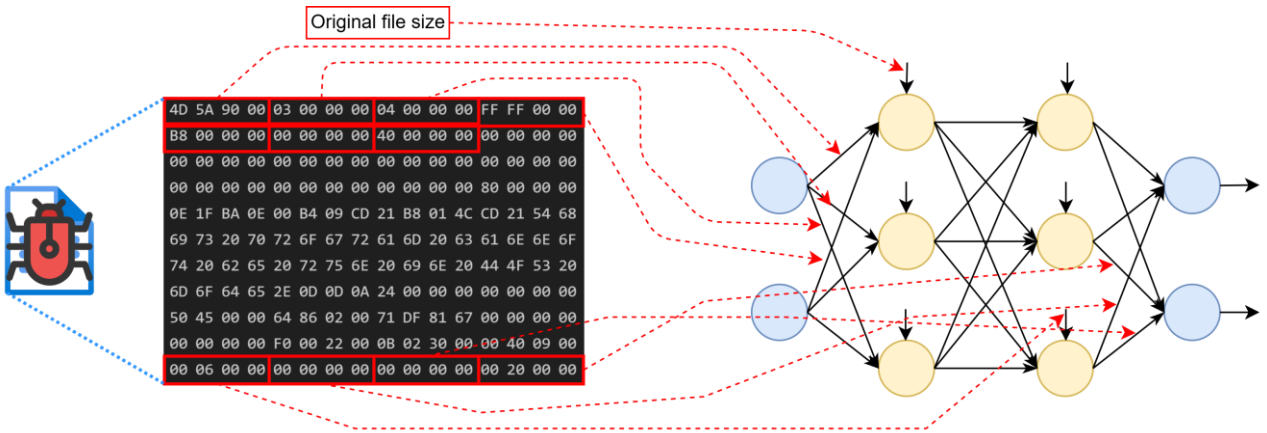
Ivestis: *inputFile* – kelias iki failo, *outputModelPath* – modelio išvesties kelias
Išvestis: Sukurtas modelio failas, kuriame patalpinti failo duomenys

(1) Perskaitome *inputFile* kaip baitu masyva: $file\_data$
(2) Paimame failo baitu kieki: $file\_size = length(file\_data)$
(3) Paruošiame baitu masyva *combined_data*, kurio dydis yra $file\_size + 4$
(4) Irašome $file\_size$ i pirmuosius keturis baitus little-endian formatu
(5) Užpildome likusia masyvo dali $file\_data$
(6) Paverčiame *combined_data* i $float32$ tensoriu: *data_tensor*
(7) Paleidžiame funkcija $make\_model\_for\_size(file\_size + 4)$, kad sukurtume tinkla su pakankamu parametru kiekiu
(8) $flat\_params \leftarrow flatten\_parameters(model)$
(9) Nukopijuojame *data_tensor* reikšmes i $flat\_params[0..(file\_size + 3)]$
(10) $unflatten\_parameters(model, flat\_params)$ – atstatome parametrus i modeli
(11) Iš $model.state\_dict()$ išsaugome su visais jo svoriais ir pridedame *__layer_sizes__* rakta, kad galėtume atkurti tinklo struktūra
(12) Galiausiai išsaugome viska kaip *outputModelPath* faila

Algorithm 2 Failo atkūrimas iš neurono tinklo parametru

Ivestis: *inputModelPath* – modelio failas, *outputFile* – failas, i kuri išsaugosime atkurtus duomenis
Išvestis: Originalus failas atkurti iš neurono tinklo
(1) $state\_dict \leftarrow torch.load(inputModelPath)$
(2) Iš *state_dict* paimame *__layer_sizes__*, kad sužinotume sluoksniu struktūra: $layer\_sizes \leftarrow state\_dict[\_\_layer\_sizes\_\_]$. Tada pašaliname ši rakta iš *state_dict*.
(3) Sukuriame nauja $MultiLayerPEStorage(layer\_sizes)$ objekta: *model*
(4) $model.load\_state\_dict(state\_dict)$ – atkuriame modelio parametrus
(5) $flat\_params \leftarrow flatten\_parameters(model)$
(6) Pirmieji keturi $flat\_params$ elementai nurodo failo dydi (baitais): $file\_size \leftarrow int\_from\_4bytes(flat\_params[0..3])$
(7) Sekančius $file\_size$ elementu ($flat\_params[4..(4 + file\_size - 1)]$) suprantame kaip failo turini
(8) Parašome atstatytus baitus i *outputFile*

Vilniaus universitetas



Kito pusmečio darbo planas

- Sudalyvauti tarptautinėje doktorantų vasaros mokykloje Italijoje, Advanced Course on Data Science & Machine Learning, birželio 9 – 13 d., 2025 m, (An Interdisciplinary Residential Course: from Generative AI to AI Agents).
- Tęsti mašininio mokymosi metodų tyrimą, skirtą kenkėjiškos programinės įrangos obfuskavimui.
- Toliau vystyti pasiūlytą mašininio mokymosi metodą, kenkėjiškos programinės įrangos obfuskavimui.
- Atlikti pradinius empirinius tyrimus, siekiant pritaikyti sudarytus arba identifikuotus metodus praktinių uždavinių sprendimui.
- Publikacijos rašymas mokslo leidiniui, turinčiam cituojamumo rodiklį Clarivate Analytics Web of Science duomenų bazėje.

KLAUSIMAI

Juozas Dautartas
juozas.dautartas@mif.stud.vu.lt