IDS/IPS technologijomis grįsto mobiliųjų įrenginių atakų prevencijos metodo sukūrimas ir tyrimas

Mantas Bacevičius

Kauno Technologijos Universitetas, Informatikos fakultetas,
Studentų g. 50, Kaunas
mantas.bacevicius@ktu.edu

Santrauka. Spartus išmaniųjų įrenginių skaičiaus ir juose saugomų duomenų kiekio ir jautrumo augimas lemia taip pat augančias ir duomenų saugumo rizikas. Šias rizikas siekia sumažinti operacinių sistemų kūrėjai, periodiškai išleisdami saugos atnaujinimus, tačiau yra nustatyta, kad pavojingiausios aplikacijos gali būti įdiegiamos kartu su šiais operacinės sistemos ar saugos OTA (angl. *over-the-air*) atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos. Taip pat apsaugos priemonės yra taikomos ir šiems mobiliems įrenginiams pritaikytoms infrastruktūroms – elektroninėms aplikacijų parduotuvėms, tačiau 67% kenkėjiškų aplikacijų vartotojus pasiekia būtent per jas. Siekiant atliepti saugumo rizikų mažinimo poreikį, šiame darbe yra pasiūlytas naujas IDS/IPS technologijomis grįsto mobiliųjų įrenginių atakų prevencijos metodas, paruošta eksperimentinė metodo realizacija ir atliktas šio metodo tyrimas.

Raktiniai žodžiai: išmanusis įrenginys, Android, IDS/IPS sistema, kenkėjiškas srautas, apsaugos metodas.

1 Įvadas

Nuolat didėjant prie interneto prijungtų įrenginių skaičiui, auga ir kibernetinių atakų, nukreiptų prieš šiuos įrenginius, pavojus. Tarptautinių kibernetinės saugos specialistų teigimu, potencialiai pavojingos aplikacijos gali būti įdiegiamos kartu su operacinės sistemos ar saugos OTA atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos [1]. Taip pat populiariais atakos vektoriais tampa sparčiai rinkoje platinamos aplikacijos [2], kurios gali pasirodyti nežalingos [3] [4], tačiau vykdo duomenų vagystės, šnipinėjimo bei kitas veiklas. Tai kelia pavojų ne tik plačioms vartotojų grupėms bet ir medicinos [5], informacinių technologijų ir pramonės [6] verslo sektoriams. Tai vyksta todėl, kad neretai organizacijose yra priimtina BYOD (angl. *Bring your own device*) politika. Tai atveria kelius

Copyright © Mantas Bacevičius, 2022. Published by Vilnius University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
DOI: https://doi.org/10.15388/LMITT.2022.1

kenkėjiškoms programoms plisti į organizacijos viduje esančius įrenginius ir tinklus. Kibernetiniai incidentai, orientuoti į mobiliuosius įrenginius kelia vis didesnį susirūpinimą mobiliųjų įrenginių ir juose saugomų duomenų saugumu, o kibernetinės apsaugos priemonėmis, kuria suteikia aplikacijų ar operacinių sistemų kūrėjai, negalima. Šiuo metu vis efektyvus mobiliųjų įrenginių tinklų apsaugojimas vis dar kelia daug iššūkių. Dėl šios priežasties būtina sukurti metodus, leidžiančius apsaugoti mobiliųjų išmanių įrenginių tinklo traktą nuo tinklo bei mobiliųjų įrenginių pažeidžiamumų ir atakų [7]. Šiame darbe yra pateikiamas apsaugos nuo tinklo atakų prieš mobiliuosius įrenginius metodas. Šis metodas leidžia vykdyti tinklo srauto analizę mobiliajame įrenginyje, naudojant mažus sisteminių resursų kiekius, kurie mobiliuosiuose įrenginiuose yra riboti. Taip pat šiame darbe yra pateikiami metodo eksperimentinės realizacijos bandymų realiomis ir sintetinėmis sąlygomis rezultatai.

2. Atakų prevencijos metodo realizacija

2.1. Realizacijos architektūra

Prototipo realizacija yra paremta Android operacinei sistemai skirta aplikacija. Aplikacija sukuria izoliuotą aplinką naudojant *chroot* konteinerį. *Chroot* konteineryje yra įdiegiama pagalbinė *Linux* operacinė sistema. Dėl lengvo pritaikymo, prieinamumo ir reikiamos minimalios papildomos konfigūracijos, buvo pasirinkta *Arch Linux* distribucija. *Chroot* konteineryje įdiegus pagalbinę operacinę sistemą, yra diegiama pasirinktina IDS/IPS sistema. Prototipo realizacijos metu buvo pasirinkta IDS/IPS sistema – „Suricata“. Svarbu pažymėti, kad konteinerio kūrimo, pagalbinės operacinės sistemos diegimo, IDS/IPS sistemos diegimo ir naudojimo metu veiksmus vykdančiai aplikacijai turi būti suteiktos „administratoriaus“ teisės, dėl šių priežasčių: (1) *Chroot* konteinerio naudojami failai yra diegiami aplikacijos konteinerio viduje tam, kad pašalinus aplikaciją, būtų pašalinti ir konteinerio viduje veikiančios pagalbinės operacinės sistemos failai, taip atlaisvinant mobiliojo įrenginio išorinės atminties talpyklos resursus; (2) Pagalbinėje operacinėje sistemoje veikiančios IDS/IPS sistemos paleidimui reikalingos administratoriaus teisės. Šios teisės leidžia IDS/IPS sistemai prieiti prie tinklo trakto.

Konteinerio kūrimo, pagalbinės operacinės sistemos, IDS/IPS sistemos diegimo veiksmams atlikti aplikacija naudoja virtualų terminalą.

2.2. Tinklo srauto perėmimo ir persiuntimo IDS/IPS procesams realizacija

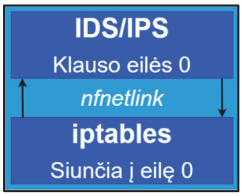
Tinklo srauto valdymui yra panaudotas pagrindinėje operacinės sistemos branduolyje veikiantis tinklo srauto maršrutizavimo karkasas *Netfilter*. Šis karkasas vykdo tinklo srauto maršrutizavimo funkcijas bei leidžia programiškai prisijungti ir stebėti su tinklo srautu susijusius įvykius. Šio karkaso veikimo koregavimui atlikti yra naudojamas pagalbinės operacinės sistemos vartotojo erdvėje veikianti tinklo srauto maršrutizavimo karkaso manipuliavimo programa *iptables*, leidžianti valdyti maršrutizavimo karkaso veiklą, naudojant konfigūruojamus taisyklių rinkinius, kurie yra taikomi taisyklėse nurodytiems tinklo srautams.

iptables palaiko taikinius *NFQUEUE*, kurie leidžia siųsti paketus į vartotojo erdvę, kur jie gali būti valdomi ir siunčiami atgal į operacinės sistemos branduolį, arba pašalinti. *NFQUEUE* eilėse saugomus paketus IDS/IPS sistemos gali perimti naudojant *Netlink* – duomenų schemomis orientuotą pranešimų siuntimo sistemą, leidžiančią perduoti pranešimus iš branduolio erdvės į vartotojo erdvę ir atvirkščiai. *Netlink* sąsaja perduodami tinklo paketai yra enkapsuliuojami į *Netlink* paketus, kurio struktūra yra pateikta 1 lentelėje.

1 lentelė *Netlink* paketo struktūra

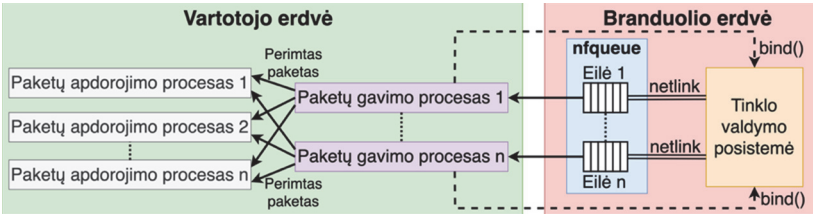
Bitų poslinkis	**0-15**	**16-32**
0	Paketo ilgis	
32	Tipas	Vėliavos
64	Sekos numeris	
96	Proceso identifikatorius (PID)	
128	Duomenys	

Netlink leidžia naudoti iki 32 ryšio magistralių operacinės sistemos branduolio erdvėje. Realizacijos atveju, ryšio magistralė yra naudojama vienos branduolio posistemės. Paketų perdavimo tarp IDS/IPS sistemos ir *iptables* schema yra pavaizduota 1 paveiksle.



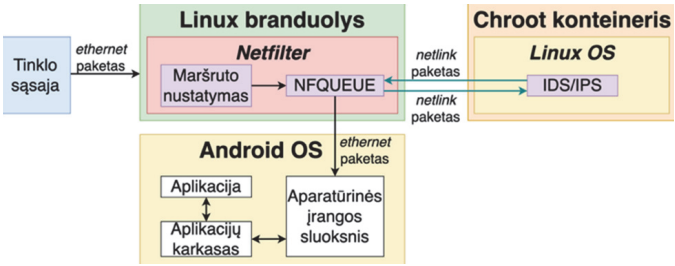
1 pav. Paketų perdavimo tarp IDS/IPS sistemos ir *iptables*

Tinklo srauto maršrutizavimo karkasas palaiko *Unicast* ir *Multicast* tipų komunikacijas, kurias palaiko ir IDS/IPS sistemos. Prototipo realizacijoje naudojama *Multicast* tipo komunikacija, kuri leidžia paketų gavimo procedūrų sudaromas apkrovas išskaidyti per keletą procesų, veikiančių skirtingose gijose. Priklausomai nuo konkrečios IDS/IPS sistemos realizacijos, paketai iš gavimo procesų gali būti perduodami asinchroniškai veikiantiems paketų apdorojimo procesams, kurių įprastai veikia bent du kart daugiau, nei yra veikiančių paketų gavimo procesų dėl didesnių skaičiavimo resursų kaštų. Paketų perdavimo, tarp vartotojo erdvės ir branduolio erdvės, procesų schema pateikta 2 paveiksle.



2 pav. Paketų perdavimas vartotojo erdvėje, naudojant *multicast* tipo komunikacijas

Pasitelkus *Netlink* sąsajos teikiamą tinklo maršrutizavimą į *NFQUEUE* eiles, iš kurių IDS/IPS sistemos perima paketus iš branduolio erdvės bei paketų grąžinimą įrenginio branduoliui tolimesniam Android OS apdorojimui, buvo pasiektas eksperimentinės realizacijos architektūrinis išdėstymas, pateiktas 3 paveiksle.

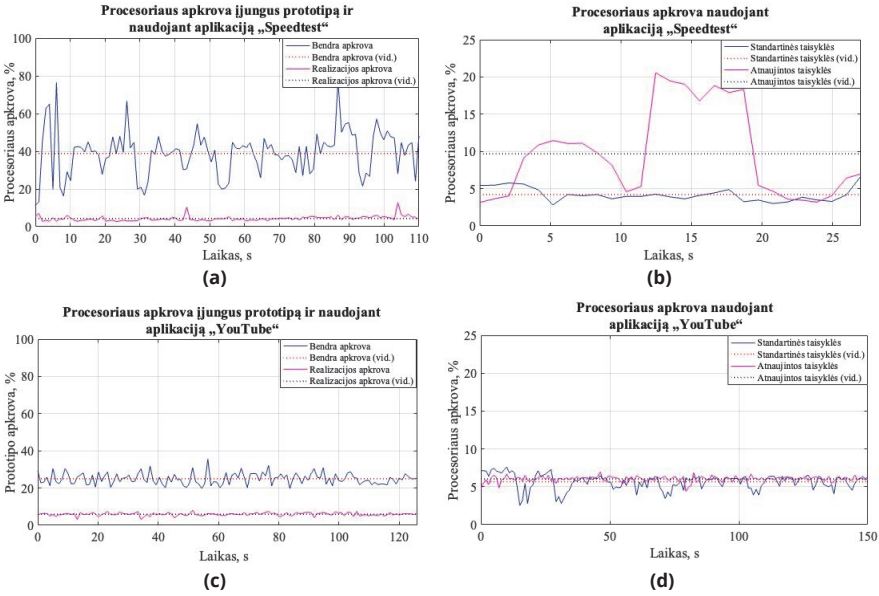


3 pav. IDS/IPS veikimo mobiliajame įrenginyje realizacijos architektūra

3. Tyrimai

3.1. Eksperimentinės realizacijos naudojamumo tyrimas

Eksperimentinės realizacijos naudojamumo tyrimui buvo atlikti keturi bandymai, kurių metu naudotos aplikacijos „YouTube“ ir „Speedtest“, siekiant atkartoti įprastą mobiliojo įrenginio naudojamą, paremtą skirtingais tinklo naudojimo profiliais. „YouTube“ aplikacija buvo pasirinkta dėl populiaraus aplikacijos naudojimo bei gebos kompensuoti tinklo pralaidumo nestabilumus, pasitelkiant buferizavimą. Aplikacija „Speedtest“ buvo pasirinkta dėl maksimalaus tinklo pralaidumo ir sisteminių resursų išnaudojimo. Tyrimo metu buvo atlikti atskiri bandymai, siekiant įvertinti eksperimentinės realizacijos generuojamos procesoriaus apkrovos ir realizacijoje naudojamų taisyklių kiekio poveikį naudojamumui. Bandymų rezultatai yra pateikti 4 paveiksle. Taip pat, tyrimų metu nustatytų metrikų paaiškinimai pateikti 2 lentelėje.



4 pav. Eksperimentinės realizacijos sugeneruojama sisteminių resursų apkrova, kartu naudojant ir aplikaciją „Speedtest“ (a), naudojant standartines ir atnaujintas taisykles (b) bei naudojant aplikaciją „Youtube“ (c), eksperimentinėje realizacijoje naudojant standartines ir atnaujintas taisykles (d)

2 lentelė. Tyrimų metu nustatytų metrikų paaiškinimai

Eil. Nr.	Žymėjimas	Paaiškinimas
1	$\overline{\mu_{e_a}}$	Realizacijos vidutinė procesoriaus apkrova,
2	β_a	Procesoriaus resursų vidutinis likutis
3	$max\ \mu_{b_a}$	Didžiausia fiksuota bendra procesoriaus apkrova
4	$max\ \mu_{e_a}$	Didžiausia fiksuota realizacijos generuota procesoriaus apkrova
5	σ_{b_a}	Bendros procesoriaus apkrovos standartinis nuokrypis
6	$\overline{\mu_{s_b}}$	Realizacijos, veikiančios su standartinėmis taisyklėmis, generuota vidutinė procesoriaus apkrova
7	$\overline{\mu_{a_b}}$	Realizacijos, veikiančios su atnaujintomis taisyklėmis, generuota vidutinė procesoriaus apkrova
8	$max\ \mu_{a_b}$	Didžiausia realizacijos generuota procesoriaus apkrova
9	σ_{a_b}	Procesoriaus apkrovos standartinis nuokrypis, naudojant realizaciją su atnaujintomis taisyklėmis
10	$\overline{\mu_{e_c}}$	Realizacijos vidutinė procesoriaus apkrova,
11	$max\ \mu_{b_c}$	Didžiausia fiksuota bendra procesoriaus apkrova
12	$\overline{\mu_{b_c}}$	Vidutinė bendroji procesoriaus apkrova
13	σ_c	Bendrosios apkrovos standartinis nuokrypis
14	$\overline{\mu_{s_d}}$	Realizacijos, veikiančios su standartinėmis taisyklėmis, generuota vidutinė procesoriaus apkrova
15	$\overline{\mu_{a_d}}$	Realizacijos, veikiančios su atnaujintomis taisyklėmis, generuota vidutinė procesoriaus apkrova
16	$\Delta\mu_{s_d a_d}$	Realizacijos veikimo su standartinėmis ir atnaujintomis taisyklėmis, sistemos apkrovos skirtumas
17	σ_{s_d}	Bendrosios apkrovos standartinis nuokrypis, naudojant realizaciją su standartinėmis taisyklėmis
18	σ_{a_d}	Bendrosios apkrovos standartinis nuokrypis, naudojant realizaciją su atnaujintomis taisyklėmis

Naudojant aplikaciją „Speedtest“, eksperimentinės realizacijos prototipas naudoja $\overline{\mu_{e_a}}$ = 4,4% procesoriaus resursų (4 pav. a). Taip pat lieka β_a = 95,6% skaičiavimo resursų kitiems sistemos procesams vykdyti. Be to, šio bandymo metu fiksuota didžiausia bendra sistemos apkrova $max\mu_{b_a}$ = 77,7%, nors maksimali realizacijos apkrova siekė $max\mu_{e_a}$ = 12,7 %. Šis pokytis tarp maksimalių bendrųjų apkrovų skirtumų ir maksimalios realizacijos apkrovos siejamas su IDS/IPS sistemos geba tinklo gavimo ir ap-

dorojimo procesus vykdyti lygiagrečiai. Buvo pastebėta, kad šio bandymo bendros apkrovos standartinis nuokrypis σ_{b_a} = 11,9 %.

Iš 4 paveikslo (b) dalies galima matyti, kad naudojant aplikaciją „Speedtest" su standartinių taisyklių rinkiniu veikianti IDS/IPS sistema „Suricata" vidutiniškai generavo $\overline{\mu_{s_b}}$ = 4,2% procesoriaus apkrovos, o su atnaujintų taisyklių rinkiniu $\overline{\mu_{a_b}}$ = 9,7%. Šių rodiklių skirtumas siejamas su didesniu atnaujintame rinkinyje esančių taisyklių skaičiumi. Dėl to grafike yra matomi keli procesoriaus apkrovos rodiklių šuoliai, kurie pasiekė $max\mu_{a_b}$ = 20,57 % ribą, o rodiklių standartinis nuokrypis buvo σ_{a_b} = 5,95 %. Dėl šių priežasčių galima teigti, kad IDS/IPS sistemai taikomų taisyklių skaičius turi tiesioginį poveikį tik esant didelei tinklo apkrovai.

Pagal rezultatus 4 paveikslo (c) dalyje, kuomet buvo naudota „YouTube" eksperimentinė realizacija generavo vidutiniškai $\overline{\mu_{e_c}}$ = 6,2% procesoriaus apkrovos. Fiksuota didžiausia bendra procesoriaus apkrova – $max\mu_{b_c}$ = 35,6%, o vidutinė bendroji procesoriaus apkrova – $\overline{\mu_{b_c}}$ = 25,0%. Taip pat verta pabrėžti, kad bendrosios apkrovos standartinis nuokrypis buvo lygus σ_c = 3,0 %, todėl galima teigti, kad eksperimentinė realizacija nesukelia didelių sisteminių resursų apkrovos šuolių, kurie turėtų įtakos sistemos naudojamumui.

Su standartinių taisyklių rinkiniu veikianti eksperimentinė realizacija vidutiniškai generavo $\overline{\mu_{s_d}}$ = 5,6% procesoriaus apkrovos, o su atnaujintų taisyklių rinkiniu $\overline{\mu_{a_d}}$ = 6,0% (4 pav. d). Tai yra $\Delta\mu_{s_d a_d}$ = 0,4% skirtumas, kuris gali būti siejamas su didesniu taisyklių skaičiumi atnaujintame rinkinyje. Eksperimentinės realizacijos veikimo su standartinėmis taisyklėmis procesoriaus apkrovos rodiklių gautas standartinis nuokrypis σ_{s_d} = 0,99 %, o su atnaujintomis taisyklėmis σ_{a_d} = 0,36 %.

3.2. Eksperimentinės realizacijos gebos fiksuoti kenkėjišką tinklo srautą tyrimas

Eksperimentinės realizacijos gebos fiksuoti kenkėjišką tinklo srautą tyrimui atlikti buvo naudota IDS/IPS sistema „Suricata", veikianti „pakartojimo" režimu bei du taisyklių rinkiniai. Standartinės taisyklės – taisyklių rinkinys, kuris įdiegiamas kartu su IDS/IPS sistema ir atnaujintos taisyklės – kurios yra įdiegiamos „suricata-update" [8] įrankio pagalba. Abu taisyklių rinkiniai buvo bandyti su tinklo srauto įrašais, saugomais „*.pcap" failuose, kuriuose yra išsaugotas kenkėjiškų programų tinklo srautas, klasifikuotas pagal kenkėjišką programą ir kenkėjiškos programos šeimą [9] . Eksperimentinės realiza-

cijos veikla buvo bandyta su p = 42 kenkėjiško tinklo srauto įrašais, kuriuos sudarė n = 3636640 tinklo paketų (žr. 3 lentelėje).

3 lentelė. Kenkėjiško tinklo srauto detekcijos, naudojant standartines ir atnaujintas taisykles

Eil. Nr., i	Kenkėjiškos programos pavadinimas	Paketų skaičius	Kenkėjiškos programos šeima	Detekcijų skaičius		Atmestų paketų dalis, μ_i	
				Atnaujintos taisyklės	Standartinės taisyklės	Atnaujintos taisyklės	Standartinės taisyklės
1	Dowgin	113005	Adware	1192	25	80,69%	6,52%
...	...	...		...	...	...	...
4	Gooligan	406956		485	0	67,82%	4,15%
...	...	...		...	...	...	...
10	Youmi	99524		1320	57	79,65%	7,21%
11	Charger	86855	Ransomware	1470	23	48,09%	4,13%
...	...	...		...	...	...	...
15	Pletor	59448		675	0	63,69%	0,22%
...	...	...		...	...	...	...
20	WannaLocker	82619		1425	61	50,90%	2,77%
21	AndroidDefender	98932	Scareware	2624	1925	62,27%	5,10%
...	...	...		...	...	...	
31	VirusShield	154540		974	60	54,47%	0,92%
32	Beanbot	18757	SMSMalware	328	8	60,45%	6,85%
...	...	...		...	...	...	...
35	fakemart	10688		184	0	37,57%	1,02%
...	...	...		...	...	...	...
42	zsone	25022		512	25	42,13%	1,35%

Naudojant standartines „Suricata“ IDS/IPS sistemos taisykles, „Gooligan“, „Pletor“ ir „fakemart“ kenkėjiškų programų, kurios priklauso 3-ims iš 4-rių bandytų kenkėjiškų programų šeimų, tinklo srautas nebuvo fiksuojamas kaip žalingas ir maža dalis tinklo srauto buvo atmesta (3 lentelė). Naudojant atnaujintas taisykles, visų bandytų kenkėjiškų programų tinklo srautas buvo fiksuotas kaip žalingus ir sugeneravo α = 44577 kenkėjiško tinklo srauto perspėjimo pranešimų. Tai yra Δpr = 1429% daugiau kenkėjiško tinklo srauto perspėjimo pranešimų, palyginus su IDS/IPS sistema, veikiančia

su standartinėmis taisyklėmis. Dėl šių priežasčių galima teigti, kad su IDS/IPS sistema „Suricata“ diegiamų standartinių taisyklių nepakanka pilnavertiškai kenkėjiškų programų generuojamo tinklo srauto detekcijai mobiliajame įrenginyje. Svarbu pabrėžti, kad IDS/IPS sistemai veikiant su atnaujintu taisyklių rinkiniu, ji atmetė $\overline{\mu_a}$ = 51,48% visų tinklo srauto įrašuose buvusių paketų.

Išvados

1. Atlikus literatūros analizę buvo nustatyta, kad pavojingiausios aplikacijos gali būti įdiegiamos kartu su operacinės sistemos ar saugos OTA atnaujinimais – apie 5% įrenginių gamintojų įdiegtų aplikacijų yra kenkėjiškos. Tai kelia pavojų ne tik plačioms vartotojų grupėms, bet ir medicinos, informacinių technologijų ir pramonės verslo sektoriams.
2. Šiame tyrime pasiūlytas naujas metodas, skirtas mobiliajame įrenginyje sustabdyti tinklo atakas. Šis metodas naudoja izoliuotą pagalbinę operacinę sistemą ir leidžia diegti bei vykdyti IDS/IPS sistemos procesus, nepriklausomai nuo įrenginio architektūros. Pagal pasiūlytą metodą paruošta eksperimentinė realizacija, gebanti izoliuotame konteineryje įdiegti *Unix* šeimos pagalbinę operacinę sistemą, IDS/IPS sistemą ir ją valdyti.
3. Nustatyta, kad standartinių, IDS/IPS sistemoje diegiamų, tinklo srauto patikros taisyklių rinkinių nepakanka, pilnavertiškai IDS/IPS sistemos veiklai ir apsaugai nuo kenkėjiško tinklo srauto. Pastebėta, kad eksperimentinės realizacijos veikimas neturi žymios įtakos įrenginio naudojamumui. Tyrimų metu buvo nustatyta, kad naudojant atnaujintas taisykles, realiomis sąlygomis, naudojamų resursų kiekis išaugo $\Delta\mu_{s_d a_d}$ =0,4%.
4. Testavimui panaudoti p = 42 kenkėjiško tinklo srauto įrašai, kuriuos sudarė n = 3636640 tinklo paketų. Eksperimentiniai rezultatai parodė, kad $\overline{\mu_a}$ = 51,48% paketų fiksavo kaip žalingus ir juos atmetė. Apie kenkėjišką tinklo srautą buvo sugeneruoti α = 44577 perspėjamieji pranešimai.

Literatūra

[1] J. Gamba, M. Rashed, A. Razaghpanah, J. Tapiador and N. Vallina-Rodriguez, „An Analysis of Pre-installed Android Software“, *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1039-1055, doi: 10.1109/SP40000.2020.00013.

[2] P. Kotzias, J. Caballero and L. Bilge, „How Did That Get In My Phone? Unwanted App Distribution on Android Devices,“ *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 53-69, doi: 10.1109/SP40001.2021.00041.

[3] B. Acohido, „Mobile threats are everywhere – here's what you can do", 2019, https://blog.avast.com/mobile-device-cyberattacks.
[4] Crowdstrike, „A COMPREHENSIVE REVIEW OF MOBILE MALWARE TRENDS," 2019, https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2019MobileThreatLandscape.pdf?lb_email=noEmail@noemail.com&utm_source=Marketo&utm_medium=Web&utm_campaign=Threat_Landscape_Mobile_Malware_2019.
[5] B. Gyunka, O. C. Abikoye, „THE IMPACT OF ANDROID MALWARE ON MOBILE HEALTH APPLICATIONS (mHealth Apps) SERVICES", *1st International Conference of the IEEE Nigeria Computer Chapter 2016 (IEEEnig ComputConf'16)*, 2016.
[6] Cybersecurity & Infrastructure Security Agency, „ICS Advisory (ICSA-18-256-01)", 2018, https://us-cert.cisa.gov/ics/advisories/ICSA-18-256-01.
[7] X. Su, M. Chauah, G. Tan, „Smartphone Dual Defense Protection Framework: Detecting Malicious Application in Android Markets", *2012 8th International Conference on Mobile Ad-hoc and Sensor Netowrks (MSN)*, 2012, pp. 153-160, doi: 10.1109/MSN.2012.43
[8] Open Information Security Foundation, „Github," https://github.com/OISF/suricata-update.
[9] A. H. Lashkari, A. F. A. Kadir, L. Taheri, A. A. Ghorbani, „Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification", *2018 International Carnahan Conference on Security Technology (ICCST)*, 2018, pp. 1-7, doi: 10.1109/CCST.2018.8585560.