KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Mindaugas Zmitrulevičius

Apsaugos nuo klaviatūros sekimo programų tobulinimas

Baigiamasis magistro darbas

Vadovas
doc. dr. J. Čeponis

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Apsaugos nuo klaviatūros sekimo programų tobulinimas

Baigiamasis magistro darbas
Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) doc. dr. J. Čeponis
(data)

Recenzentas
(parašas) dr. Audronė Janavičiūtė
(data)

Projektą atliko
(parašas) M. Zmitrulevičius
(data)

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Mindaugas Zmitrulevičiaus

(Studento vardas, pavardė)

Informatikos ir informacinių technologijų sauga, 621E10003

(Studijų programos pavadinimas, kodas)

„Apsaugos nuo klaviatūros sekimo programų tobulinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 17 m. gegužės 22 d.

Kaunas

Patvirtinu, kad mano, **Mindaugo Zmitrulevičiaus,** baigiamasis projektas tema „Apsaugos nuo klaviatūros sekimo programų tobulinimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka) *(parašas)*

Zmitrulevičius, M. „Apsaugos nuo klaviatūros sekimo programų tobulinimas“. Magistro baigiamasis projektas / vadovas doc. dr. Jonas Čeponis; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Raktažodžiai: klaviatūros sekimas, taikomas lygmuo, branduolio lygmuo, sauga.

Kaunas, 2017. 52 p.

SANTRAUKA

Klaviatūros sekimo programos - tai programinė arba techninė įranga, kuri renka informaciją apie kompiuterio vartotojo klavišų paspaudimus ir siunčia surinktą informaciją piktavaliui. Nors ši problema žinoma labai seniai, klaviatūros sekimo programos šiais laikais yra daug didesnė problema nei anksčiau, nes šios programos sparčiai tobulėja bei plinta internetu. Šios piktavališkos programos tampa įvairesnės, rafinuotesnės, sunkiau aptinkamos. Norint apsisaugoti nuo įsilaužėlių naudojamos antivirusinės programos. Tačiau, kaip ir apsisaugojimo priemonės nuo įsilaužėlių, taip ir piktavalių sukurtos programos ar įtaisai, padedantys sekti klaviatūra atliekamus veiksmus, tobulėja. Todėl antivirusinės programos ne visada padeda apsisaugoti nuo įsilaužėlių.

Dėl šios priežasties baigiamojo darbo tikslas yra patobulinti ir pagerinti vartotojų klaviatūra įvedamų duomenų apsaugą. Sukurtas metodas privalo apsaugoti vartotojo konfidencialią informaciją. Kadangi programinis klaviatūros sekimas vykdomas taikomajame bei branduolio lygmenyje − realizuotas sprendimas turi padėti apsisaugoti nuo abiejuose lygmenyse veikiančių klaviatūros sekimo programų.

Galutinis baigiamojo darbo rezultatas − apsaugos nuo klaviatūros sekimo programų tobulinimas. Atlikus tyrimus yra nustatomas realizuotų apsaugos nuo klaviatūros sekimo programų metodų veiksmingumas, piktavališkų programų aptikimo tikslumas, CPU apkrovos padidėjimas.

Zmitrulevičius, M. "*Improving Protection against Keyboard Tracking Applications"*: *Master's* thesis / supervisor doc. dr. Jonas Čeponis. Department of Computer Science, Faculty of Informatics, Kaunas University of Technology.

Research area and field:

Key words: keylogger, security, kernel level, application layer.

Kaunas, 2017. 52 p.

SUMMARY

Keyloggers – this is software or hardware that collects information about computers user's keystrokes and sends the gathered information to hacker. The problem is very old, the keyboard tracking program nowadays is much larger problem than earlier. Malicious programs is evolving rapidly and spreading online. This malicious program becomes more diverse, sophisticated and difficult to detect. To prevent from hackers most of users use antivirus programs. However, as well as prevention measures against hackers, new malicious and devices to track of actions performed by the keyboard is created every day.

For this reason, the final aim is to improve improve the user input data keyboard protection. A method is required to protect the user's confidential information. The software keyboard tracking is carried out at application and kernel level, realized the solution to help protect themselves from the two levels of the active keyboard tracking software.

The final result is to improve protection against keyloggers programs. The experiments are designed to set the protection on the keyboard tracking program effectiveness methods, malicious programs detection accuracy, CPU load increase.

Turinys

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

FTP (angl. *File Transfer Protocol*, „Failų Perdavimo Protokolas“) – standartas failų perdavimui.

IDS (angl. *Intrusion detection system)* – atakų atpažinimo sistema.

OS – operacinė Sistema, speciali programinė įranga, užtikrinanti vartotojo sąsają ir kompiuterio techninės įrangos taikomųjų programų bei duomenų valdymą. Moderniausios operacinės sistemos sudaro galimybę dirbti daugeliui vartotojų vienu metu daugialypėje aplinkoje, užtikrina bylų (failų) apsaugą, turi daug kitų naudingų savybių. Dauguma operacinių sistemų yra pirma programinė įranga, kurią pradeda vykdyti įjungtas kompiuteris.

BIOS – *Basic Input/Output System* bazinė įvesties/išvesties sistema – tai sisteminė programa, kuri saugoma kompiuterio pastovios atminties (informacija nedingsta išjungus elektros maitinimą) mikroschemoje.

IRQ – kompiuteryje, pertraukimų prašymas (ar yra aparatūros signalas siunčiamas į procesorių, kuris laikinai sustabdo vykdomą programą ir speciali programa *Londonderry*, įterpties prižiūrėtojas, paleidžiamas jos vietoj.

Tty (angl. *Teleprinter*) – elektromechaninis teletaipas.

Rootkits – slapto tipo kenkėjiška programa, sukurta pasislėpti kompiuteriuose ir bandanti išvengti antivirusinių programų nustatymo. Ji leidžia prisijungti prie kompiuterio administratoriaus teisėmis ir suteikia prieigą prie jūsų asmeninių duomenų. *"Rootkit'ai"* dažniausiai naudojami su kitais kenkėjais, kad juos paslėptų nuo vartotojo ir apsaugos sprendimų.

CPU – funkcinis vienetas, kurį sudaro vienas arba keli procesoriai ir jų vidinė atmintis.

LKM (angl. *loadable kernel module*) – užkraunamas branduolio modulis.

SSH (angl. *Secure shell*) – tai tinklo protokolas, aprašantis apsaugotą kliento prisijungimą prie serverio aplinkos (*shell*) ir komandų vykdymą. Standartinis ssh prievadas (portas) yra 22 (TCP).

ĮVADAS

Šiuolaikinėje visuomenėje telefonai ir kompiuteriai yra neatsiejama žmogaus gyvenimo dalis. Šiuos įrenginius mes naudojame įvairiais tikslais. Naudojant kompiuterio ar telefono klaviatūrą mes dažnai įvedinėjame konfidencialią informaciją. Neteisingai prižiūrimas kompiuteris ar telefonas yra patrauklūs įsilaužėliams. Norint to išvengti reikalingos papildomos priemonės apsaugančios nuo įsilaužėlių.

Norint apsisaugoti nuo įsilaužėlių naudojamos antivirusinės programos. Tačiau, kaip ir apsisaugojimo priemonės nuo įsilaužėlių, taip ir piktavalių sukurtos programos ar įtaisai, padedantys sekti klaviatūra atliekamus veiksmus, tobulėja. Todėl antivirusinės programos ne visada padeda apsisaugoti nuo įsilaužėlių [1].

Klaviatūros sekimo programos – tai programinė arba techninė įranga, kuri renka informaciją apie kompiuterio vartotojo klavišų paspaudimus ir siunčia surinktą informaciją piktavaliui. Nors ši problema žinoma labai seniai, klaviatūros sekimo programos šiais laikais yra daug didesnė problema nei anksčiau, nes šios programos sparčiai tobulėja ir plinta internetu. Šios piktaviliškos programos tampa įvairesnės, rafinuotesnės, sunkiau aptinkamos [2].

Tyrimo sritis – nelegalus vartotojų informacijos perėmimas. Tyrimo objektas – klaviatūra įvedamos informacijos sekimas ir perėmimas.

Darbo tikslas ir uždaviniai:

Tikslas: pagerinti vartotojų klaviatūra įvedamų duomenų apsaugą.

Uždaviniai:

- išanalizuoti klaviatūros sekimo programų veikimo principus;
- ištirti klaviatūros sekimo pasirinktose OS galimybes;
- išanalizuoti egzistuojančias apsaugas nuo klaviatūros sekimo;
- pasiūlyti metodą padedantį apsisaugoti nuo klaviatūros sekimo;
- praktiškai realizuoti pasiūlytą metodą ir atlikti šio metodo analizę.

Darbo struktūra:

- analizės dalyje išanalizuoti klaviatūros sekimo programos principus ir apsaugas nuo sekimo;
- antroje darbo dalyje pasiūlomas ir realizuojamas metodas, leisiantis vartotojui vesti duomenis virtualia klaviatūra, kuri skirsis nuo realios, todėl sekimo programa negaus tikslių duomenų;
- trečiojoje darbo dalyje bus praktiškai realizuotas pasiūlytas metodas. Bus išanalizuoti populiarių OS klaviatūros programų sekimo principai, pasiūlytas apsaugos nuo klaviatūros sekimo modelis;
- paskutinė darbo dalis pateikia bendras išvadas.

1. KLAVIATŪRA ĮVEDAMOS INFORMACIJOS APSAUGOS ESAMOS SITUACIJOS ANALIZĖ

Klaviatūros sekimo programos paprastai yra skirstomos i dvi pagrindines kategorijas: techninė įranga ir programinė įranga [3]. Techninė klaviatūros sekimo įranga - maži prietaisai kurie lengvai gali būti įdėti tarp kompiuterio klaviatūros ir kompiuterio prievado − tokio kaip PS/2 arba USB. Programinės įrangos klaviatūros sekimas − tai programos kurios veikia kompiuterio viduje [4].

Klaviatūros sekimas aparatiniame lygmenyje gali būti užmaskuotas kaip dalis klaviatūros kabelio arba įmontuotas kaip klaviatūros komponentas. Juos dažnai sunku pastebėti bei jo veikla gali būti sunkiai pastebima ir įsilaužimo aptikimo sistemoms [5]. Aparatiniai klaviatūros sekimo įrankiai taip pat gali būti žalingesni nei programiniai, nes jie gali rinkti informaciją apie paspaustus klavišus nuo to momento, kai kompiuteris yra įjungtas, o taip veikiant gali gauti ir BIOS slaptažodžius [6].

1.1. pav. Klaviatūros sekimas vyksta dviejuose lygmenyse

Programinės įrangos klaviatūros paspaudimų registravimo programos gali būti lengvai sukuriamos ir platinamos piktavalių. Kadangi klaviatūros sekimo programos gali būti pridėtos į atsisiunčiamus failus su kita programine įranga − ji dažnai gali likti nepastebima [7]. Yra dviejų tipų programinės įrangos klaviatūros sekimas: branduolio lygmens ir taikymo lygmens klaviatūros paspaudimų registravimo programos, kurios naudojamos atsižvelgiant į tai, kuriame lygmenyje veikia piktavališka programa.

1.1. Programinis klaviatūros sekimas

Yra dviejų tipų programinės įrangos klaviatūros sekimas: branduolio lygmens ir taikymo lygmens klaviatūros paspaudimų registravimo programos, kurių pasirinkimas priklauso nuo to, kuriame lygmenyje veikia piktavališka programa. Taikomojo lygmens klaviatūros paspaudimų registravimo programos paprastai naudoja aukšto lygio sistemos funkcijas klavišų paspaudimams stebėti. Pavyzdžiui “Windows” sistemos *SetWindowsHookEx* funkcija gali įdiegti gaudyklės (angl.

hook) procedūrą į gaudyklės grandinę tam tikroms klaviatūros procedūroms stebėti, bei *GetAsyncKeyState* funkcija gali nustatyti ar klavišas buvo paspaustas arba atleistas, kada funkcija iškviečiama po paskutinio klavišo paspaudimo [3]. Taikomojo lygmens klaviatūros sekimo programos yra lengvai sukuriamos, jas ganėtinai paprasta aptikti. Parašo ir gaudyklės principu veikiančios programos gali efektyviai aptikti taikomojo lygmens klaviatūros sekimo programas.

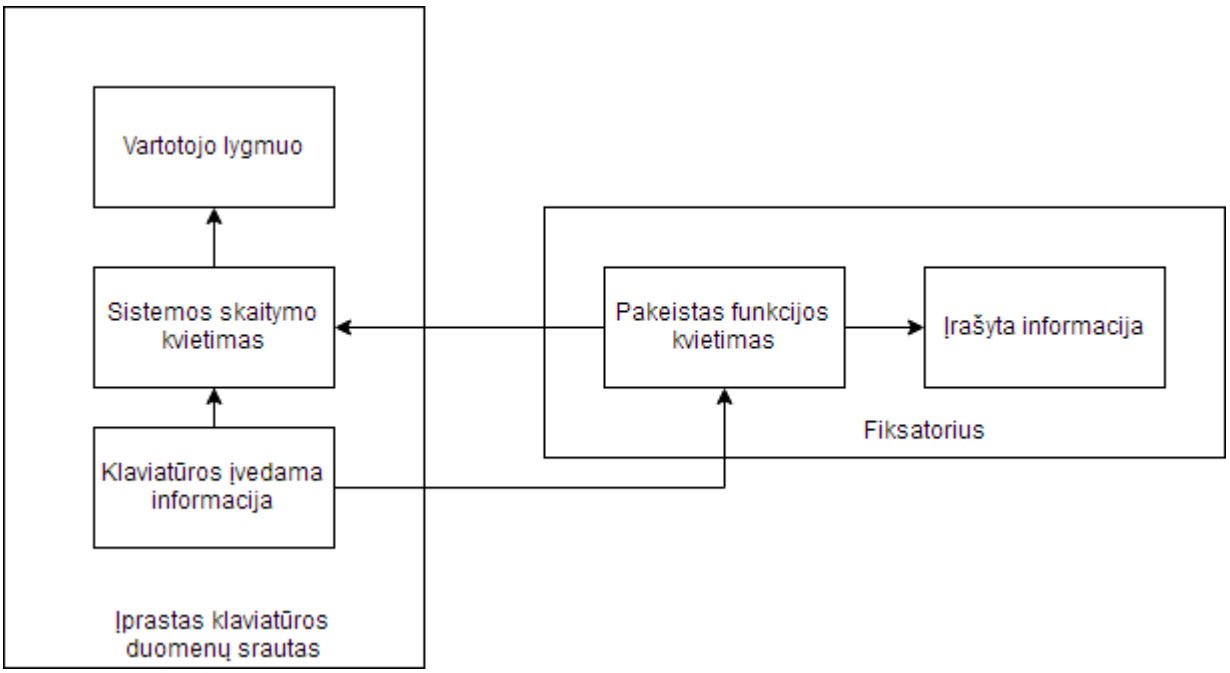
Tai programos sukurtos veikti aukos kompiuterio OS. Programinis klaviatūros sekimas skirstomas į:

- **prižiūryklės** (angl. *Hypervisor-based*): klaviatūros sekimo programa teoriškai gali būti kenkėjiškoje prižiūryklėje po operacine sistema [8].
- **branduolio lygmens** (angl. *Kernel-based*): programa mašinoje paima šaknines teises, kad paslėptų save OS ir perima klavišo paspaudimus kurie keliauja pro branduolį [9]. Šią kenkėjišką programą sunku sukurti, tačiau sunku ir aptikti.
- **taikomųjų programų sąsajos** (angl. *API-based*): klaviatūros sekimo programa prisikabina prie klaviatūros taikomųjų programų sąsajos veikiančioje programoje **(angl.** *Application***)** [10]. Klaviatūros sekimo programa registruoja kiekvieną klavišo paspaudimą, kaip tai būtų normali programa, o ne kenkėjiška programa.
- **formos perėmimo metodas** (angl. *Form grabbing based*): Formos perėmimo klaviatūros sekimo programa saugo veiksmus atliktus internete, saugo naršyklės veiksmus [11]. Ši kenkėjiška programa išsaugo formą prieš išsiunčiant ją internetu.
- **naudojantis atminties įterpimą** (angl. *Memory injection based*): Atminties įterpimą naudojančios piktavalės programos saugo funkcijų įrašus įspėjant atminties lenteles susijusias su naršymu ir kitomis sistemos funkcijomis [12].
- **paketų analizavimas** (angl. *Packet analyzers*): Į tai įtraukta tinklo paketų sugavimas susietas su HTTP POST veiksmais kad gauti neužšifruotus slaptažodžius. HTTPS tai apsunkina.
- **paremata Javascript** (angl. *Javascript-based*)**:** kenkėjiška kodas yra įdedamas į tikslinį puslapį ir klauso, klavišų įvykių, tokių kaip *onKeyUp()* [13]. Kodas gali būti įskiepijamas daugybe skirtingų būdų įterptinių komandų atakos**(**angl. Cross-site scripting (XSS)**),** žmogus naršyklėje (angl. *Man-in-the-browser* (MITB, MitB, MIB, MiB)), žmogus viduryje (angl. *man-in-the-middle***)** [14].
- **nuotolinės prieigos šnipinėjimo programos** (angl. *Remote access software keyloggers*): Tai yra vietinės programos naudojamos šnipinėjimui (su pridėta funkcija prisijungti nuotoliniu būdu).

Dauguma šių kenkėjiškų programų nėra sustabdomos HTTPS šifravimo, kadangi tai apsaugo tik informacijos apsikeitimą tarp kompiuterių. Tačiau šitos programos egzistuoja nuosavame kompiuteryje, tame kuris prijungtas prie klaviatūros.

1.1.1. Branduolio lygmens klaviatūros sekimo programos

Branduolio lygmens klaviatūros sekimo programos veikia OS branduolio lygmenyje ir renką klavišų informaciją tiesiai iš klaviatūros [15]. Bendrais bruožais, branduolio lygio klaviatūros sekimo programos veikia pakeičiant klaviatūros tvarkyklės funkcijos kvietimą sava funkcija, kuri įdeda fiksavimo modulį kaip pavaizduota pav 1.2. Šis registravimo modulis yra užprogramuotas fiksuoti visas klavišų reikšmes perduodamas per modifikuotą funkciją. Toks fiksavimo mechanizmas padaro branduolio lygmens klaviatūros sekimo programą nematoma esamoms klaviatūros sekimo apsaugoms. Branduolio lygio klaviatūros sekimo aptikimas bei prevencija vis dar yra sudėtinga užduotis [5] [6].



1.2. pav. Branduolio lygmenyje vykdomas klaviatūros sekimas

1.1.2. Nuotolinio prisijungimo klaviatūros sekimo programos

Tai yra lokalaus klaviatūros sekimo veikimo programa su pridėta funkcija, kuri sugeba nuotoliniu būdų prisijungti prie duomenų kurie įrašyti lokaliai. Nuotolinis prisijungimas gali būti atliekamas tokiais metodais:

- Įrašai yra įkeliami į svetainę, duomenų bazę, ar FTP serverį.
- Informacija periodiškai persiunčiama į iš anksto nustatytą el. paštą.
- Duomenys perduodami į bevieliu tinklu prijungtą sistemą.

- Programa suteikia prieigą nuotoliniu būdu prie lokaliai veikiančios mašinos, per Internetą ar vietinį tinklą, informacija saugoma aukos kompiuteryje, kad panorėjus programišius prie jos galėtų prisijungti.

Nuotolinė prieiga prie surinktų klaviatūros sekimo duomenų padeda apsisaugoti piktavaliams, todėl tampa sunku juos nustatyti, aptikti. Atsiranda galimybė sėdint namuose šnipinėti savo aukas.

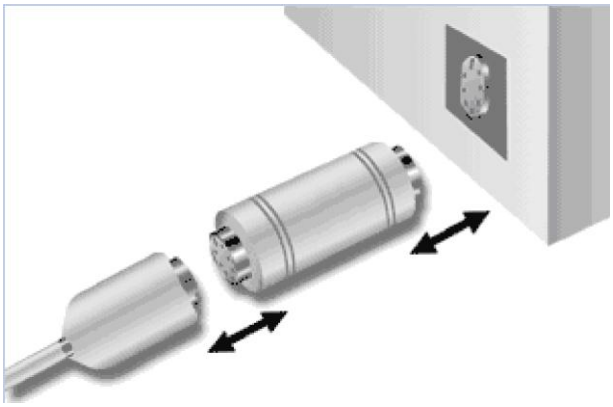
1.1.3. Papildomas programinio klaviatūros sekimo funkcijos

Šnipinėjimo programos gali būti papildytos funkcijomis, kad kenkėjiška programa nepasikliautų tik klaviatūros paspaustais mygtukai. Kai kurios iš šių funkcijų susideda iš:

- Trumpalaikės atminties sekimas**.** Tai būdas gauti informaciją kuri buvo nukopijuota. Ši informacija gali būti gauta šnipinėjimo programos.
- Ekrano šnipinėjimas. Yra daromos ekrano nuotraukos, kad gauti gauti informaciją paremtą grafikos pagrindu [16]. Aplikacijos su galimybę atlikti ekrano sekimo funkciją, gali fotografuoti ekraną, daryti viso ekrano, tik vienos aplikacijos, arba tik vaizdą esantį aplink kursorių. Nuotraukos gali būti daromos periodiškai arba priklausomai nuo vartotojo elgesio. Tai padeda nugalėti interneto svetainėje naudojamas papildomas klaviatūras.
- Programinis teksto sugavimas. Kai kurios „The Microsoft Windows“ aplikacijos leidžia programom prašyti teksto reikšmės. Tai reiškia, kad galima sugauti tekstą, net jei jis yra užmaskuotas po slaptažodžio kauke [17].
- Įrašymas kiekvienos programos/aplanko/atidaryto lango taip pat ekrano nuotraukos kiekvienos atidaryto puslapio.
- Duomenų įrašymas apie kiekvieną paiešką, susirašinėjimus, FTP atsisiuntimus, bei kita Interneto naudojimą.

1.1.4. Aparatiniame lygmenyje vykdomas klaviatūros sekimas

Techniniai įrangai veikiant klaviatūros sekimas vykdomas pagalbinį prietaisą įterpiant tarp klaviatūros jungties ir kompiuterio prievado pav 1.3. Klaviatūros sekimo įrenginys dažniausiai būna panašus į universalią jungtį arba į bet kokią kitą jungtį naudojamą kompiuteryje. Kad auka nesuprastų jog šis įrenginys yra naudojamas šnipinėjimui [18]. Šis įrenginys turi vidinę atmintį, kurioje ir yra saugoma informaciją apie atliktus veiksmus naudojant klaviatūrą.



1.3. pav. Klaviatūros sekimo įrenginio įterpimas

Aparatiniam lygmenyje vykdomos klaviatūros sekimas skirstomas į:

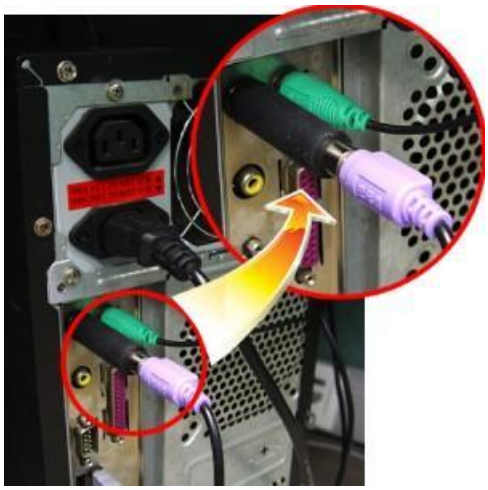
- **įprastinis aparatinis klaviatūros sekimas (angl.** *Firmware-based***)**: Naudojamas surinkti informaciją apie klaviatūros klavišų paspaudimus. Informaciją apie vykdomą veiklą yra saugojama įrenginio vidinėje atmintyje [19].;
- **klaviatūros aparatūra (angl.** *Keyboard hardware***):** Aparatiniame lygmenyje vykdomas klaviatūros sekimas yra skirtas rinkti informaciją apie mygtukų paspaudimų reikšmes, tai vykdoma tarp klaviatūros ir kompiuterio, dažniausiai klaviatūros laido jungtyje. Kad sumažinti susekamumo lygį klaviatūros sekimo prietaisai gali būti sumontuoti pačioje klaviatūroje. Informacija apie gautus duomenis yra saugoma prietaiso vidinėje atmintyje kuri gali būti atidaryta įvedus slaptą raktą.;
- **bevielis klaviatūros šnipinėjimas (angl.** *Wireless keyboard sniffers***)**: surenka informacijos paketus perduodamus iš bevielės klaviatūros. Ir tuomet bandoma nulaužti šifravimo raktą naudojamą apsaugoti bevielį komunikavimą tarp dviejų įrenginių [20].;
- **mikro programa (angl.** *Microprogram***)**- kompiuterio išvedimo įvedimo sistema, kuri paprastai yra atsakinga už klaviatūros veiksmus gali būti suprogramuota, kad rinktų informaciją apie klaviatūros klavišų paspaudimus.;
- **klaviatūros perdengimas (angl.** *Keyboard overlays***)** – netikra klaviatūra yra uždedama ant tikros taip kad kiekvienas mygtuko paspaudimas būtų fiksuojamas ir taip pat tikroji klaviatūra reaguotų į mygtuko paspaudimus [21].;
- **akustinis klaviatūros sekimas (angl.** *Acoustic keyloggers***)**: Akustinį šnipinėjimą galima atlikti pasiklausant klaviatūra spaudžiamų klavišų garso, kadangi kiekvienas klavišas skleidžia skirtingą signalą [22]. Taip pat analizuojamas dažnumas tarp klaviatūros mygtukų paspaudimų [23].;
- **elektromagnetinės emisijos (angl.** *Electromagnetic emissions***)**: galima sužinoti laidu prijungtos klaviatūros elektromagnetinę emisija esant iki 20 metrų atstumu nuo jos, nereikia

būti prie jos prisijungus. Naudojant plačiajuostį imtuvą paversti į tam tikrą dažnį gautą emisiją išspinduliuotą iš klaviatūros [24].;

- **optinis sekimas (angl.** *Optical surveillance***)**: vykdomas kameros pagalba stebėti klaviatūra atliktus veiksmus [25].;
- **fiziniai įkalčiai (angl.** *Physical evidence***)**: klaviatūroms skirtoms įvesti tik slaptažodį ant mygtukų kurie bus paspausti liks pirštų antspaudai. Sužinojus klavišus galima vykdyti perrinkimo ataką.;
- **išmaniųjų telefonų sensorių (angl.** *Smartphone sensors***)**: išmanųjį telefoną padėjus ant to pačio stalo kaip klaviatūra galima akselometro pagalba nustatyti vibracijas sukurtas spaudinėjant klavišus. Tuomet šitas vibracijas paversti į perskaitomus žodžius su 80% tikslumu [26].

1.1.5. Klaviatūros sekimo programų konstrukcija

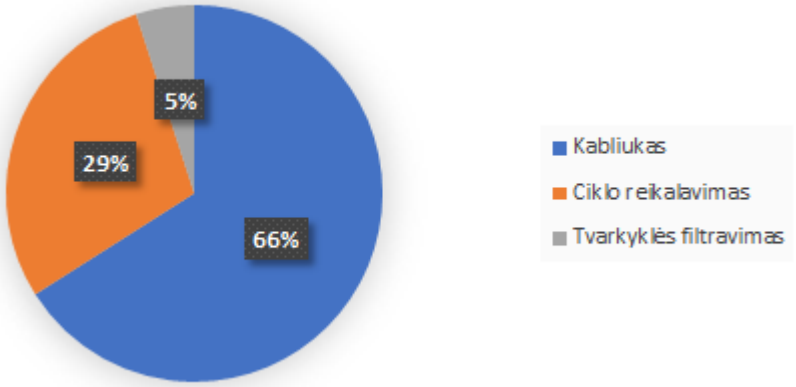
Pagrindinė klaviatūros sekimo programos idėja yra įsiterpti į bet kokią grandinę kaip pavaizduota pav. 1.4 kuri sudaryta iš momento kai klaviatūros mygtukas yra paspausta ir kada informacija yra parodoma ekrane. Tai galima pasiekti įvairių programų pagalba ar įrenginių.
Populiariausi metodai sukurti klaviatūros sekimo programas:



1.4. pav. Prijungtas aparatinis klaviatūros sekimas

- Sistemos kabliukas kuris perima pranešimus apie tai kuris klavišas buvo paspaustas (dažniausiai rašomas C programavimo kalboje);
- Ciklinis klaviatūros prašymas apie klaviatūros veikimas (dažniausiai rašomas pasinaudojant *Visual Basic*, kartais *Borland Delphi*);
- Naudojant filtravimo tvarkyklę (reikalauja specialių žinių, rašomas C kalboje).

Skirtingų klaviatūros sekimo programų pasiskirstymas pavaizduotas pav. 1.5.



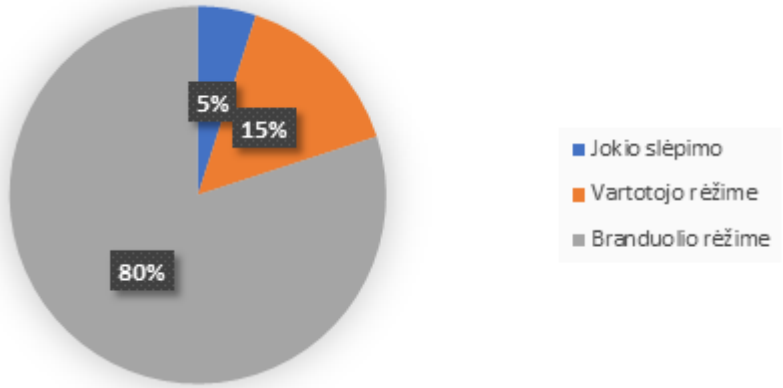
1.5. pav. Klaviatūros sekimo programų pasiskirstymas

Labiausiai paplitusios klaviatūros sekimo priemonės naudoja kabliuko principą.

Klaviatūros sekimo programos užmaskuoja savo failus, kad apsaugoti failus, nuo jų radimo bei antivirusinių programų. Ši paslėpimo technologija vadinama kompiuterių programų rinkiniu. Yra dvi pagrindinės kategorijos naudojamos klaviatūros sekimo programų:

- Vartotojo rėžimo maskavimas;
- Branduolio rėžimo maskavimas.

Pasiskirstymas pagal maskavimą pateikta diagram pav. 1.6.:



1.6. pav. Klaviatūros sekimo programų pasiskirstymas pagal slėpimą

Diagramoje matyti, kad labiausiai paplitęs klaviatūros šnipinėjimo pasislėpimo būdas yra branduolio rėžime.

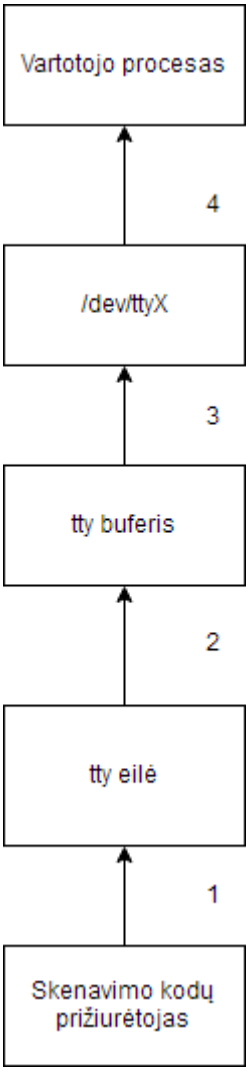
1.1.6. "Linux" klaviatūros sekimo modelis

Klaviatūros tvarkyklė skirtingose OS branduoliuose turi panašią struktūrą ir veiklą [4]. Jie paprastai susideda iš daug sudedamųjų dalių, atliekančių įvairias operacijas su klavišų duomenimis Šiame skyriuje išanalizavome "Linux" branduolio klaviatūros sekimo modelį detaliau, dėl to, kad "Linux" sistema yra atviro kodo.

“Linux” sistemoje branduolio lygmens klaviatūros sekimas paprastai atliekamas kaip užkraunamasis branduolio modulis (angl. *loadable kernel module* (LKM)), kuris gali būti lanksčiai užkrautas į branduolį, LKM pasiekti gali prieiti prie aparatinės įrangos prietaisų, ir kontroliuoti ar keistis duomenis kiek reikalinga. Klaviatūros sekimo moduliai gali būti patalpinti į įvairius klaviatūros tvarkyklės komponentus kad perimti jos duomenis, perduoti jį vartotojo lygmeniui, ir tuomet išsaugoti juos į failą, ar persiųsti juos tinklu.

“Linux” branduolio klaviatūros sekimo programos veikia vienu iš dviejų metodų: registruojant tvarkyklės pertraukimus arba pakeičiant vartotojo funkcijas. Pirmas metodas grindžiamas “Intel” architektūra, kurioje nutraukiamas prašymas (IRQ) konkretaus klaviatūros valdiklio skaičiaus, abiejuose klaviatūros klavišų duomenys ir registruotas statusas yra apsikeičiamas per fiksuotus prievadus − 0x60 ir 0x64. Klaviatūros modulis fiksuoja pertraukymų prašymų tvarkyklę, kad perskaityti klaviatūros duomenis ir jos statusą. Tačiau šis metodas nėra patogus dėl savo platformos priklausomybės. Antrasis metodas paprasčiausiai modifikuoja esamas klaviatūros tvarkyklės funkcijas įterpdamas kenkėjišką kodą, todėl šis metodas labiau bendro pobūdžio.

Pav. 1.7. vaizduoja bendrą “Linux” klaviatūros tvarkyklės modelį. Kuomet klaviatūros mygtukas yra paspaudžiamas, klaviatūros lustas sugeneruoja iki šešių skenavimo kodai. Šie neperskaitomi skenavimo kodai yra perduodami į klaviatūros tvarkyklę, tuomet apdorojami ir konvertuojami į klavišų paspaudimų ir atleidimų seriją, priklausančius nuo skanavimo kodu vertimų lentelės manipuliavimo komponentų Pav. 1.7. pažymėta nr 1.



1.7. pav. Bendras Linux klaviatūros tvarkyklės modelis

Kuomet klavišų veiksmai yra sukonvertuojami i klavišų simbolius pagal tai ką jie atitinka vertimų lentelėje, jie patenka tty buferio eilę Pav. 1.7. pažymėta nr. 2**.** Kaip pagrindinis „Linux“ tty šerdies komponentas, tty branduolys garantuoja stabilumą generuojant klavišų kombinacijas, kurios yra perskaitomi simboliai. Klavišų kombinacijos gali skirtis skirtingoms klaviatūros tipams, bet dauguma tvarkyklių gali apimti pagrindines klavišų kombinacijas, kad standartinės klaviatūros veiktų tinkamai.

Galiausiai, tty buferio susijusi funkcija periodiškai kviečiama, tam, kad gauti simbolius iš tty buferio, ir įdėti juos tty skaitymo eilę Pav. 1.7. pažymėta nr. 3. Paprastai vartotojo lygio programas naudoja *read()* sistemos kvietimą, tam kad gauti laukiamų ženklų Pav. 1.7. pažymėta nr. 4**.**

1.1.7. Klaviatūros sekimo programų klavišo reikšmių fiksavimo būdai

Klaviatūros sekimo programos gali pakeisti klaviatūros tvarkyklės funkcijas, ir fiksuoti klavišų reikšmes trimis skirtingais būdais.

- **Perimant tvarkymo skenavimo kodus ar eilės klavišų kombinacijas**

Klaviatūros sekimo programos gali pakeisti *handle_scancodes()* ir *put_queue()* funkcijas, kad tiesiai fiksuotų skanavimo kodus. Šie pakeitimai yra neįprasti. Todėl, kad sugauti skanavimo kodai yra neįskaitomi, taip pat šis modulis priklausomas nuo platformos, todėl neįmanoma fiksuoti nuotolinės sesijos metu įvestų simbolių.

- **Renkant duomenis gaunančiajame buferyje**

Sekimo programos gali pakeisti žemo lygmens tty tvarkyklės funkciją *receive_buf(),* kad gauti klavišų kombinacijas. Klaviatūros sekimo programos pakeičia šias funkcijas nukreipdami *ldisc.receive_buf* į naują funkciją kurioje įdėtas fiksavimo modulis. Nauja funkcija gali fiksuoti visus perskaitomus simbolius tty buferyje ir fiksuoti nuotolinės sesijos metu paspaustus klaviatūros klavišus. Kadangi tty buferis yra bendras daugeliui platformų [5] [27], šis metodas yra efektyviausias ir populiariausias fiksavimo mechanizmas naudojamas klaviatūros sekimo programose.

- **Iš dalies pakeisti buferinę funkcija arba sistemos skaitymo kvietimą.**

Kiekvieną kartą, kuomet procesas skaito simbolį iš tty šerdies naudodamasis *sys_read()* funkciją, kviečiama funkcija *tty_read().* Programiškai gali pakeisti šias dvi funkcijas savo modifikuotomis. Kadangi *read_tty()* ir *sys_read()* yra aktyviai naudojamos įvairių procesų, jų pakeitimai žymiai sulėtintų sistemą, tokiu atveju jas aptikti būtų lengva.

Apibendrinant, fiksuoti tty buferyje yra veiksmingiausias iš šių trijų metodų. To priežastis yra tai, kad įsibrovėlis tiesiogiai prieina prie klavišų kombinacijos skaitymo palikdamas kelius akivaizdžius pėdsakus sistemoje. Todėl mūsų aptikimo sistema orientuota aptikti šį metodą naudojančias klaviatūros paspaudimų registravimo programas. Ši sistema taip pat taikoma, klaviatūros sekimo programom kurios naudoja kitus klavišu fiksavimo metodus, nors reikalingi keli plėtiniai.

1.2. Klaviatūros šnipinėjimo programų privalumai bei trūkumai

Šnipinėjimo programos gali veikti trijuose skirtinguose lygmenyse - tai branduolio lygmenyje, taikomajame lygmenyje bei aparatiniame lygmenyje. Kiekviename lygmenyje klaviatūros sekimas atliekamas skirtingai, todėl kiekvieno jų aptikimas, sukūrimas bei funkcionalumas skirtingas. Šioje lentelėje 1.1. palyginami skirtinguose lygmenyse veikiančios klaviatūros sekimo programos.

1.1. lentelė Klaviatūros šnipinėjimo programų palyginimas

	Branduolyje veikianti klaviatūros sekimo programa	Taikomajame lygmenyje veikianti klaviatūros sekimo programa	Aparatiniame lygmenyje veikianti klaviatūros sekimo programa
Sukūrimo sudėtingumas	Sunku	Lengva	Vidutinis
Prieiga įdiegimui	Nuotolinė	Nuotolinė	Fizinė. Reikalinga papildoma įranga kurią reikia prijungti prie aukos kompiuterio.
Aptikimas	Sudėtingas	Lengvas	Lengvas (jungiamos papildomos priemonės)
Klavišų perėmimas	Perėmimas vyksta kompiuterio žemiausiame lygmenyje - branduolyje. Negali perimti automatiškai užbaigiamų slaptažodžių, gali prieiti prie visos informacijos įvedamos klaviatūra ir keliaujančios per OS.	Perėmimas vyksta aplikaciniame lygmenyje. Galimas klaidingų klavišų perėmimas.	Gaunama informacija apie klavišų paspaudimus prieš tai kai ji pasiekė įrenginį.
Prieiga duomenų perėmimui	Nuotolinė, duomenys gali būti persiųsti tinklu.	Nuotolinė, duomenys gali būti persiųsti tinklu.	Gali saugoti tik ribotą kiekį duomenų. Reikalinga fizinė prieiga prie įrenginio. Negalima nuotolinė prieiga.

Palyginus šiuos tris klaviatūros šnipinėjimo programų tipus, yra aišku, kad lengviausia apsisaugoti yra nuo aparatiniame lygmenyje vykdomo klaviatūros sekimo. Sunkiausia aptikti yra branduolio lygmenyje vykdomą klaviatūros sekimą, tačiau norint sukurti tokio lygio programą reikia daug žinių. Todėl pats populiariausias yra taikomojo lygmens klaviatūros sekimas, nesunku sukurti, lengva platinti, nereikia fizinės prieigos.

1.3. Apsisaugojimas nuo šnipinėjimo programų

Yra daugybė būdų kaip šnipinėjimo programos renka informaciją todėl reikia daug įvairių technikų kaip aptikti bei apsisaugoti nuo šių kenkėjiškų programų. Pavyzdžiui klaviatūrą veikianti ekrane yra yra veiksmingas būdas apsisaugoti nuo aparatiniame lygmenyje veikiančių šnipinėjimo programų. Prieš šnipinėjimą (*anti-spyware*) sukurtos aplikacijos kurios gali išjungti kablio principu veikiančias klaviatūros sekimo programas, bus neaktyvios prieš branduolyje veikiančias kenkėjiškas programas.

Taip pat šnipinėjimo programą galima atnaujinti, to pagalba galima sumažinti susekamumo lygį, ir kenkėjiška programa gali toliau sėkmingai veikti aukos kompiuteryje. Apsisaugojimo būdų yra keletas:

- Programos aptinkančios šnipinėjimo programa;
- Diskas ar USB atmintinė kurioje yra veikianti OS;
- Tinklo monitoriai;
- Automatiškai formas užpildančios programos;
- Vienkartiniai slaptažodžiai;
- Apsaugos ženklai;
- Ekrano klaviatūros;
- Klaviatūros mygtukų paspaudimų programos;
- Kalbos atpažinimas;
- Rašymo ranka ir pelės judesių atpažinimas;
- Macro plėstuvai/įrašytuvai;
- Branduolio lygmenyje veikiančių klaviatūros sekimo programų aptikimas.

Anti šnipinėjimas (angl. *anti keylogger*) tai programos dalis sukurta aptikti klaviatūros sekimo programas jūsų įrenginyje. Paprastai lygina visus failus jūsų kompiuteryje su kenkėjiškų programų duombaze, ieškodami sutapimų [28]. Šios programos buvo sukurtos aptikti klaviatūros sekimo programas, todėl jos turi daugiau potencialo nei antivirusinės programos. Kadangi kai kurios anti-virusinės programos šnipinėjimo programą laiko tinkamai veikiančios programos dalimi.

Perkraunant kompiuterį naudojant diską ar USB atmintinę kurioje yra OS kurią galima paleisti kompiuteryje. Tai galimas variantas apsisaugant nuo programinio klaviatūros sekimo, jei diskas ar

USB yra švarus, jame nėra kenkėjiškų programų. Tačiau įjungiant skirtingą operacinę sistemą negalima apsisaugoti nuo aparatiniame lygmenyje ar BIOS veikiančioms šnipinėjimo sistemoms.

Tinklo monitoriai kitaip žinomos kaip atvirkštinės ugniasienės gali būti naudojamas perspėti vartotoją kad aplikacija bando užmegzti ryšį su internetu. Tai suteikia šansą vartotojui neleisti išsiųsti informacijos kurią gavo kenkėjiška programa. [29]

Automatiškai formas užpildančios programos gali padėti išvengti konfidencialios informacijos atskleidimo, panaikindami reikalavimą klaviatūra suvesti reikalingą informaciją. Šios programos yra sukurtos tinklo naršyklėms, kad jos užpildytu reikiamus laukus naudodamos vartotojo turimą informaciją. Programoje yra išsaugoma konfidenciali informacija kaip kredito kortelių duomenys, prisijungimo slaptažodžiai ir kiekvieną kartą jungiantis jie yra suvedami automatiškai. Tačiau tai sukelia kitą grėsmę, asmuo turinti fizinį prisijungimą prie įrenginio gali įdiegti programą, kad gauti šią informaciją iš OS ar tuomet kai ji yra perduodama tinklu.

Vien kartiniai slaptažodžiai yra saugūs kadangi jie tampa nebegaliojantys kuomet yra panaudojami. [30] Tačiau, jeigu programišiai turi nuotolinę prieigą prie kompiuterio jie gali palaukti kol slaptažodis bus įvestas ir atlikti norimus veiksmus.

Išmaniųjų kortelių ar kitų apsaugos ženklų naudojimas gali padidinti apsaugą prieš pakartojimo atakas, tai padidina apsaugą nuo klaviatūros šnipinėjimo, kadangi tuomet reikalingas slaptažodis bei apsaugos ženklas tuoks kaip išmani kortelė [31]. Todėl turimos informacijos apie klaviatūros klavišų paspaudimus, pelės veiksmų, kopijuotos informacijos, ekrano vaizdo neužteks norint gauti prieigą prie apsaugų resursų. Išmaniųjų kortelių skaitytuvai ir prie jų esančios klaviatūros įvesti PIN kodą gali būti pažeidžiamos. [4]

Daugelis ekrano klaviatūrų siunčia normalios klaviatūros žinutes papildomai programai. Kiekviena programiškai veikianti klaviatūros sekimo programa gali saugoti įvestus simbolius. [32] Taip pat klaviatūros sekimo programos gali fotografuoti ekrane atliekamus veiksmus.

Klaviatūros mygtukų paspaudimų programos mėgina apgauti klaviatūros sekimo programas siųsdami atsitiktina tvarka paspaustų klavišų rinkinius. [33] Todėl programišiui sunkiau atrinkti teisingą informaciją.

Panašiai kaip ekrano klaviatūros, kalbėjimo pavertimas į tekstą programos gali būti naudojamos prieš šnipinėjimo programas [34]. Jeigu nėra įtraukta klaviatūros mygtukų paspaudimų ar veiksmų su kursorium. Silpniausia šio metodo dalis kaip bus persiunčiama gauta informacija.

Daug planšetinių kompiuterių gali paversti liečiamame ekrane atliktus veiksmus, rašymą, simboliais, taip pat tai galima atlikti kursoriaus veiksmais, naudojant papildomas programas.

Didelio kiekio programų pagalba, atrodantį bereikšmį tekstą galima paversti prasmingu. Kaip daroma telefone naudojant žodžio spėjimo funkcijas. Didžiausia šio būdo silpnybė kad programos

siunčia mygtukų paspaudimus tiesiai į taikinio programą. Tačiau to galima išvengti siunčiant kursoriaus paspaudimus į neatsakančią vietą aukos programoje, siunčiant nereikšmingą tekstą.

Kol klaviatūros sekimo programų elgesys yra nepastebimas, efektyvus sprendimas šiai problemai yra sudėtingas [35]. Dažnas sprendimas yra naudoti *anti-rootkits* tam, kad aptikti kenkėjiškus klavišų perėmimus branduolio lygmenyje. *Rootkits* − tai programos sukurtos sugadinti teisėtą OS kontroliavimą. Kai kurios branduolio lygmenyje veikiančios klaviatūros sekimo programos naudoja rootkits tam, kad įterpti sistemos kvietimus. *Anti-rootkit* paprastai lieka atmintyje ir nuskaito visus procesus, modulius, paslaugas, ir užkraunamos tvarkyklės tam, kad rasti įtartiną veiklą [36]. Kadangi branduolio lygmens klaviatūros sekimo programos dažnai gali efektyviai paslėpti savo elgesį, jų neaptinka *anti-rootkits*. Be to, *anti-rootkits* gali generuoti didelę apkrova, skenuojant daug žemo lygio sistemos komponentų. Mūsų sistema naudoja dinaminio žymėjimo analizės techniką ir išlaiko dėmesį į klavišų duomenis, todėl jis gali būti tikslesnis ir efektyvesnis [37].

1.4. Analizės išvados

Apžvelgus visus nagrinėtus klaviatūros sekimo būdus gautos tokias išvadas:

1. Išanalizavus pagrindinius klaviatūros sekimo programų veikimo principus, įsitikinta, kad klaviatūros sekimo programos labai pavojingos ir lengvai plintančios. Taip pat įsitikinta, kad paprastam vartotojui gali pridaryti labai daug žalos − tiek finansinės, tiek moralinės, o moralinė žąla gali būti neįkainojamas turtas.
2. Išanalizuotos kompiuterių vartotojų informacijos perėmimo galimybės, kurios leidžia nesunkiai perimti vartotojo konfidencialią informaciją. Įsitikinta jog yra daugybė įvairių perėmimo būdų, ir apsisaugoti nuo jų tampa vis sudėtingiau.
3. Palygintos klaviatūros sekimo programas, veikiančias branduolio lygmenyje, taikomajame lygmenyje, bei aparatiniame lygmeny. Pastebėta, kad branduolio ir taikomojo lygmens klaviatūros sekimo programos yra efektyvesnės nei aparatinio lygmens klaviatūros sekimas. Taikomajame bei branduolio lygmenyje veikiančios klaviatūros sekimo programos turi daugiau funkcionalumo, yra sunkiau aptinkamos.
4. Išanalizavus apsisaugojimo bei klaviatūros sekimo programų aptikima ir apsisaugojimą nuo jų pastebėta, kad dauguma priemonių aptikimui naudoja tik parašo tipo aptikimą. Jis nėra pakankamai efektyvus kai klaviatūros sekimo programa yra nauja ir jos duomenys nėra įtraukti į duombazę. Tuomet klaviatūros sekimo programa nėra aptinkama.
5. Apžvelgus aptikimo bei apsisaugojimo priemones, nustatyta, kad būtina suprojektuoti ir patobulinti priemones padedančias apsisaugoti nuo klaviatūros sekimo. Kadangi taikomajame bei branduolio lygmenyje klaviatūros sekimas yra didelis pavojus.

2. KLAVIATŪRA ĮVEDAMOS INFORMACIJOS PROGRAMINĖS APSAUGOS TOBULINIMO PROJEKTAS

Yra dviejų tipų programinės įrangos klaviatūros sekimo programos: branduolio lygmens ir taikomojo lygmens klaviatūros paspaudimų registravimo programos, priklausomai nuo to kuriame lygmenyje veikia programa. Taikomojo lygmens klaviatūros paspaudimų registravimo programos paprastai naudoja aukšto lygio sistemos funkcijas stebėti klavišų paspaudimus. Pavyzdžiui, "Windows" sistemos, `SetWindowsHookEx()` funkcija gali įdiegti gaudyklės (angl. *hook*) procedūrą į gaudyklės grandinę stebėti tam tikras klaviatūros procedūras, bei GetAsyncKeyState funkcija gali nustatyti, ar klavišas buvo paspaustas arba atleistas, kuomet funkcija iškviečiama po paskutinio klavišo paspaudimo. Taikomojo lygmens klaviatūros sekimo programos yra lengvai sukuriamos, bet ir aptinkamos jos ganėtinai lengviau. Pėdsako pagrindu ir gaudyklės pagrindu technika gali efektyviai aptikti taikomojo lygmens klaviatūros sekimo programas [3].

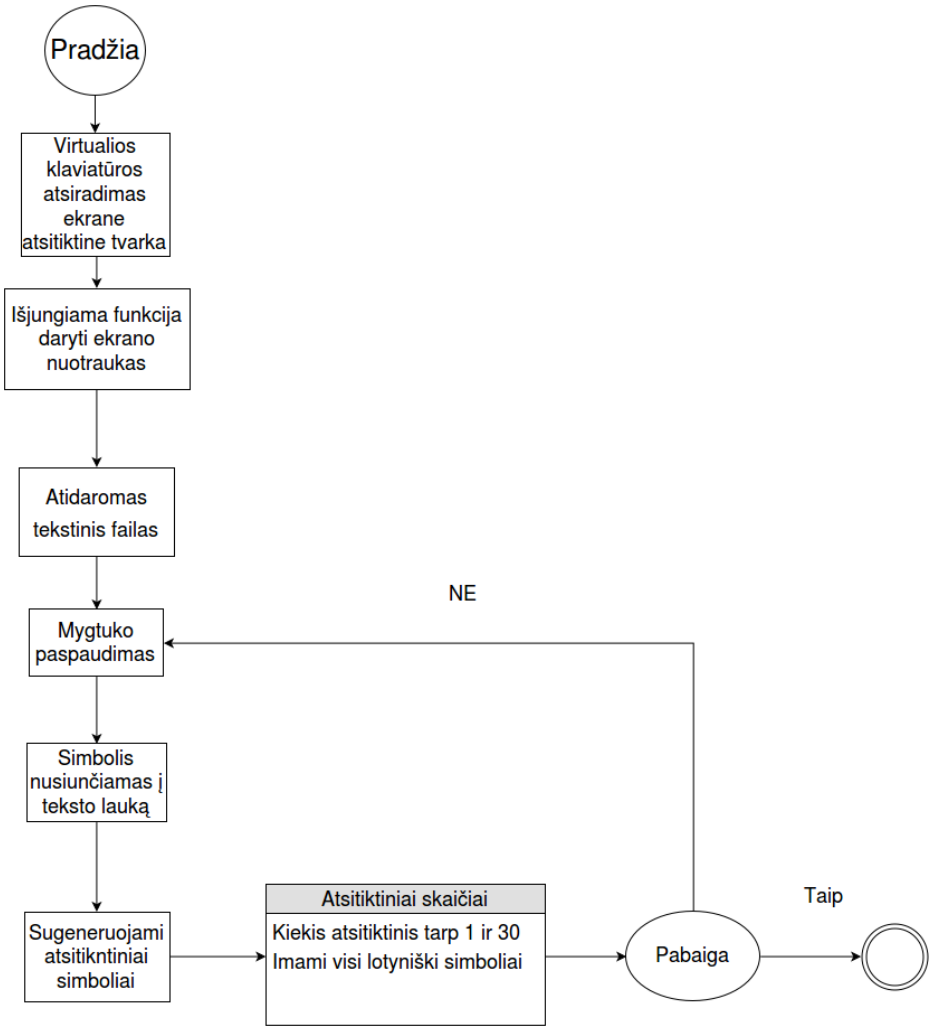
Pėdsako pagrindu: remiantis klaviatūros sekimo programų elgesiu, ši technika stebi failus, dinamiškai sujungtas bibliotekas arba registrų įrašus kurie yra pakeisti arba įdėti piktavališkai į sistemą. Be to, tikrinama visos programos pagal pėdsakus ir tikrinama ar nėra sutapimų duombazėje, pagal tai aptinkamos klaviatūros sekimo programos. Ši technika naudojama keliose komercinėse programose, tokiose kaip "WEBROOT" + "Prevx" [38] ir "TrendMicro" [39] , bet ji nėra efektyvi aptinkant klaviatūros sekimo programas kurių pėdsakai nėra užfiksuoti duombazėje. Gaudyklės pagrindu šis metodas turi pranašumą naudodamas `SetWindowsHookEx()` funkciją stebėti klaviatūros būseną prieš ir po, perduodant klavišų duomenis tarp dviejų gaudyklių procedūrų. Todėl klaviatūros sekimo programos perimančios duomenis tarp gaudyklės procedūrų gali būti aptiktos. Lyginant su pėdsako pagrindu veikiančiais klaviatūros sekimo aptikimo metodais, ši technika yra efektyvesnė ir plačiau naudojama. Pavyzdžiui HookFinder [40] technika buvo išplėsta aptikti gaudyklės pagrindo veikiančias kenkėjiškas programas. Kai kurios komercinės programos taip pat naudoja šią technika, tokios kaip "VICE" [41], "Microsoft Antispyware", "Ad-Aware" [42].

Klaviatūros sekimas yra vienas klastingiausių pavojų vartotojo asmeninei informacijai. Slaptažodžiai, banko duomenys ir kreditinės kortelės gali būti perimtos pasinaudojant klaviatūros sekimo programomis. Ši grėsmė labai sparčiai auga. Ne taip kaip žvejojimo atakos − šias atakas sunkiau susekti. Paprasti vartotojai pasitiki savo kompiuterio sistemomis, antivirusinėmis programomis, bei ugniasienėmis. Dėl šios priežasties jie tampa lengvais taikiniais.

2.1. Klaviatūros apsauga nuo šnipinėjimo taikomajame lygmenyje

Iki šiol yra atlikta keletas tyrimų norint išspręsti problemą dėl virtualių klaviatūrų koordinačių sekimo bei fotografavimo. Vienas iš pasiūlytų apsisaugojimo metodų buvo: kursorius pasirenka tam

tikrą simbolį ir tuomet jis yra pakeičiamas specialiu simboliu. Taip pat kuomet yra paspaudžiamas klaviatūros klavišas visi klaviatūroje esantys klavišai yra pakeičiami atsitiktinais simboliais, kad ekrano nuotraukoje nesimatytų kokie ištikrųjų simboliai tai yra. Kitas pasiūlytas metodas − spalvotos klaviatūros sukūrimas. Skaičiai bei raidės klaviatūroje yra pažymėtos skirtingomis spalvomis. Kiekvieną kartą paspaudus mygtuką klaviatūroje esantys simboliai yra išdėstomi nauja atsitiktine tvarka. Prieš paspaudžiant klavišą reikia įsiminti norimo mygtuko vietą klaviatūroje ir prieš tai paspausti “Hide Keys” klavišą. Tai paslėps visus mygtukus ir jie bus rodomi tušti. Tuomet vartotojui reikės paspausti norimą mygtuką. Kad vartotojas nepamirštų mygtuko vietos − naudojamos spalvos. Šių metodų problema yra ta, kad tai apsaugotų tik nuo slaptažodžių nužiūrėjimo. Kadangi visų klaviatūroje esančių simbolių vietos žmogus nespėtų atsiminti. Bet nuo ekrano vaizdo fotografavimo tai neapsaugo.

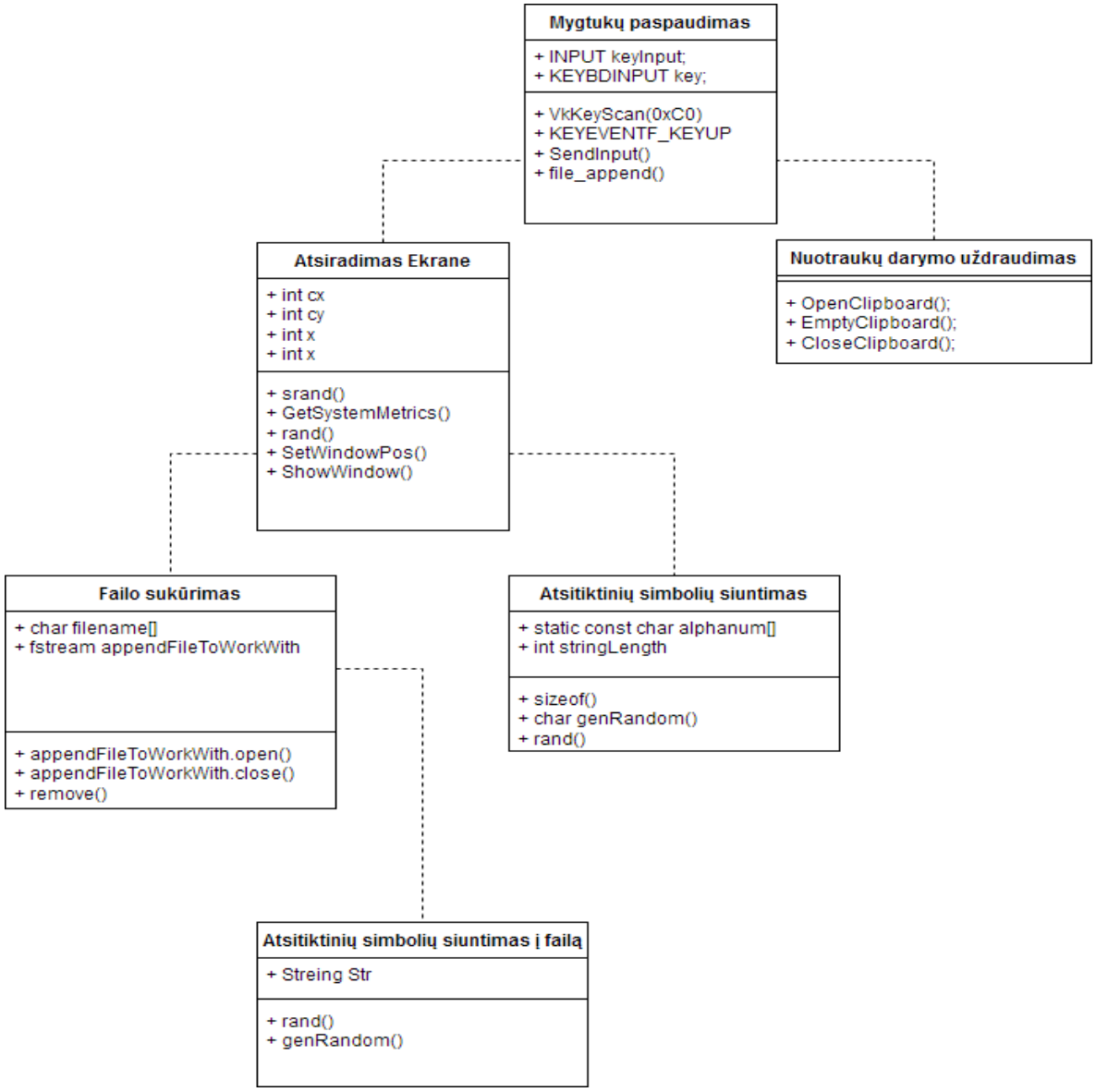


2.1. pav. Taikomojo lygmens klaviatūros apsaugos modelis

Norint apsisaugoti nuo virtualios klaviatūros sekimo sukursime virtualią klaviatūrą bei ją pritaikysime kelioms “Windows” operacinei sistemai. Apsaugos modelis pavaizduotas pav. 2.1.

2.1.1. Virtuali klaviatūra

Kadangi ne tik kompiuteriai, bet ir telefonai tampa neatsiejama mūsų gyvenimo dalis – dauguma mūsų įvedinėja konfidencialią informaciją pasinaudojant mobiliaisiais telefonais, jungdamiesi prie bankų, elektroninių paštų ir kitur, kur reikalingi slaptažodžiai.



2.2. pav. Sistemos klasių diagrama

Piktavaliai įvairiais būdais bando ją perimti. Išnagrinėję įvarius apsisaugojimo būdus nusprendėme sukurti priemonę padedančią apsisaugoti nuo klaviatūros šnipinėjimo. Pav. 2.2 pavaizdavome sistemos modelio klasių diagramą.

2.1.2. Virtualios klaviatūros atsiradimas ekrane atsitiktine tvarka

Norint apsaugoti vartotojus pritaikysime sistemą, kuri kiekvieną kartą virtualios klaviatūros vietą parinks atsitiktine tvarka. Taip pat tai bus daroma kiekvieną kartą kai vartotojas paspaus klaviatūros klavišą, kad piktavalis negalėtų nustatyti koks simbolis buvo paspaustas pasinaudodamas

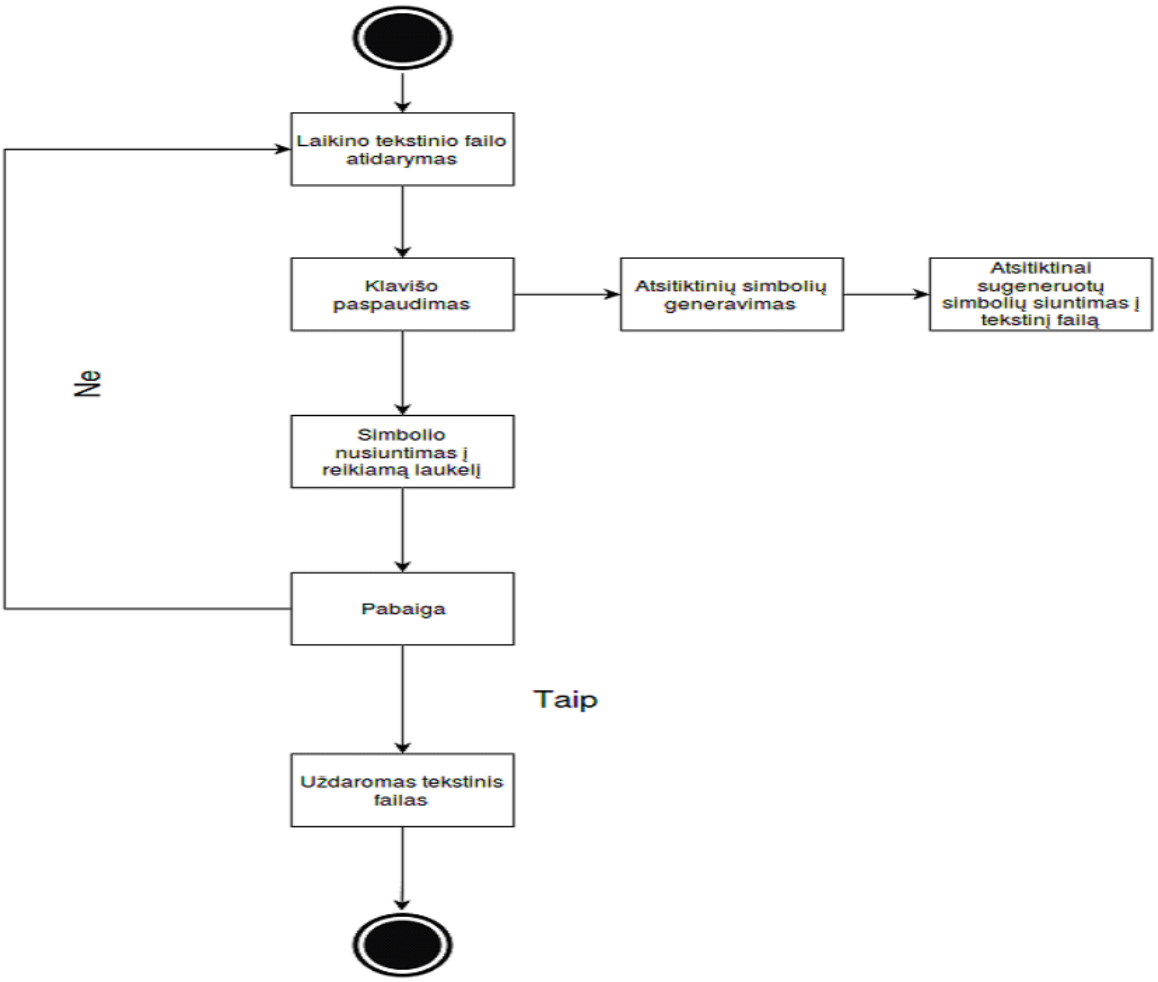
x bei y koordinatėmis ekrane.

2.1.3. Ekrano vaizdo fotografavimo uždraudimas

Išnagrinėję klaviatūros sekimo priemones sužinojome, kad šios programos dažnai daro ekrano vaizdo nuotraukas - viso ekrano nuotrauką, tik aktyvių programų nuotraukas, kursoriaus pozicijos nuotraukas. Taip piktavaliai gali nustatyti kur ir koks tekstas buvo vedamas. Todėl sukurtoje sistemoje uždraudžiame daryti ekrano nuotraukas.

2.1.4. Atsitiktinių simbolių siuntimas

Išanalizavę klaviatūros sekimo programų veikimo principus išsiaiškinome, kad taikomajame lygmenyje veikiančios klaviatūros sekimo programos fiksuoja kiekvieną gautą klavišo signalą. Sistemoje pritaikomas atsitiktinių simbolių sekos generavimo algoritmas pav. 2.3. Sekos dydis taip pat yra atsitiktinio dydžio. Taip piktavaliui yra sunku atskirti kur yra tikri simboliai kur yra generuoti, dėl šios priežasties negalima atstatyti tikro slaptažodžio ar kitos konfidencialios informacijos.



2.3. pav. Atsitiktinių simbolių generavimo bei siuntimo modelis

2.2.Apsauga nuo branduolio lygmenyje veikiančių klaviatūros sekimo programų

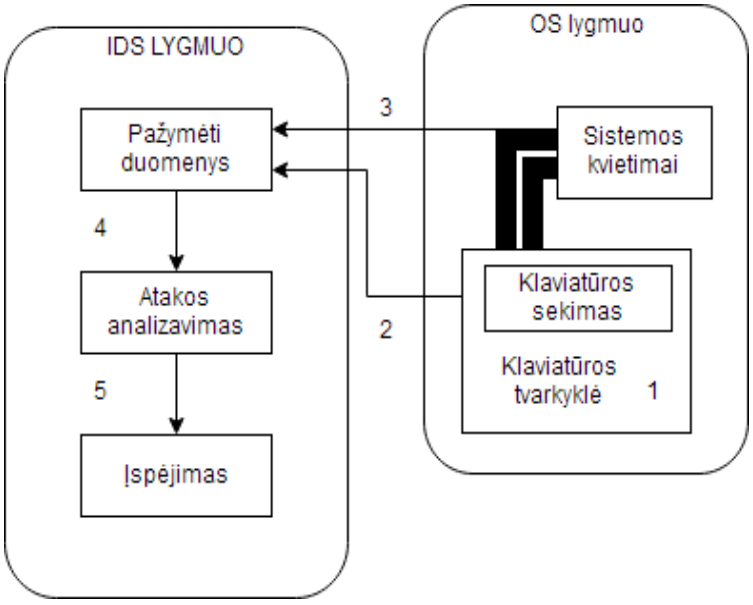
Sukūrėme sistemos prototipą, kuri atlieka įsilaužimo aptikimo sistemos funkcijas ir palaiko dinaminio žymėjimo analizę [43]. Patikrinamas šio prototipo efektyvumą naudojant „Linux“ branduolyje veikiančią klaviatūros sekimo programą pavadinimu “Vlogger”. Tyrimo rezultatai parodė, kad siūloma sistema gali tiksliai aptikti branduolio lygmenyje veikiančią klaviatūros sekimo programos veiklą ir nustatyti jos šaltinį.

Mūsų sistema naudoja įsilaužimo aptikimo sistemą(IDS) tam, kad paženklinti, stebėti ir išnagrinėti klaviatūros duomenis tvarkyklės lygmenyje. Pažymėti duomenys - tai klaviatūros duomenų turinys bei jos atminties adresas. Kuomet branduolio lygmens klaviatūros sekimo programa taiko fiksavimo modulį, kad nustatyti pažymėtus klaviatūros duomenis, šis kenkėjiškas elgesys bus aptiktas ir išanalizuotas naudojantis mūsų moduliu realizuotu įsilaužimo aptikimo sistemoje.

- Sisteminio kvietimo sužadinimas: parašome naują sistemos kvietimą, kuris veikia kaip sukėlėjas, kad praeiti klavišų duomenų srauto kontrolę iš OS branduolio į IDS. Klaviatūros duomenys yra pažymėti ir stebimi įsilaužimo aptikimo sistemos prieš tai, kai ji patenka į piktavališką programą. Sukėlėjas veikia gerai, kadangi OS sistemos kvietimų struktūra yra bendrinė ir primityvi, jie yra modifikuojami.
- Dinaminis duomenų žymėjimas: mes pasinaudojome IDS funkcionalumu tam, kad dinamiškai žymėti ir stebėti klavišų duomenų srautą. Daug IDS sugeba stebėti branduolio lygio duomenų srautą ir jį keisti tiek kiek reikia - taip pritaikomas klaviatūros duomenų žymėjimas.

2.2.1. Branduolio lygmenyje veikiančios apsaugos nuo klaviatūros sekimo sudarymas

Kaip parodyta pav. 2.4**,** visa konstrukcija susideda iš dviejų funkcionalumo lygių: OS funkcijų ir IDS lygio funkcijų. Pagrindiniai aptikimo komponentai yra IDS lygmenyje, OS lygmuo atsakingas už klavišų duomenų perdavimą IDS lygmeniui.



2.4. pav. Branduolio lygmenyje veikianti klaviatūros šnipinėjimo aptikimo sistema

2.2.2. OS lygmuo

OS lygmuo perduoda informaciją susijusią su klavišų duomenimis IDS lygmeniui, kad jie būtų pažymėti bei analizuojami. Susijusi informacija tai duomenų turinys ir korespondento atminties adresas, įvykio operacija. Duomenų turinys yra klavišų duomenys. Tam, kad įrašyti klavišų duomenis, piktavališkos programos naudoja tam tikras konkrečias operacijas duomenims perduoti. Dėl šios priežasties reikia stebėti šių operacijų veiksmus.

2.2.3. IDS lygmuo

IDS veikia kaip analizės komponentas, jame vykdomi du veiksmai su klavišų duomenimis − žymėjimas bei analizavimas.

2.2.4. Žymėjimas

Remiantis OS lygio būsenos keitimu, žymintis komponentas žymi klavišų duomenis ir įdeda pažymėtus duomenis į žymėtų duomenų lentelę. IDS lygmuo struktūrizuoja žymėtus duomenis lentelėje remiantis *passed_data***.** Lentelėje kiekvienoje eilutėje yra duomenys ir jos adresas, ir atitinkama PID žyma. Kai priskiriama vieno bito reikšmė, žyma yra pasirenkama nustatant, kad duomenys buvo pažymėti. Pažymėtas komponentas apibūdina gautus duomenis tam, kad identifikuoti originalius duomenis. Toliau aptariame procedūras.

2.2.5. Analizavimas

Analizavimo komponentas pirmiausia patikrina ar įtariami duomenys yra pažymėti. Jeigu nepažymėti − mes ignoruojama duomenis kol jie priklauso teisėtam rašymo procesui. Jeigu pažymėti

− mums reikia patikrint ar jie yra naudojami kenkėjiškų rašymo opearcijų. Tikrinimo komponentas patikrina PID reikšmes gautas iš *passed_data* ir iš pažymėtų duomenų lentelės. Po palyginimo, jeigu PID reikšmės nesutampa, reiškia šios rašymo operacijos yra atliktos kenkėjiško branduolio fiksavimo proceso. Tuomet IDS lygmuo išveda įspėjimą. Įspėjime įtraukti kenkėjiško saugojimo proceso PID klavišo duomenys ir korespondento atminties adresas. Kitu atveju, jeigu PID reikšmė sutampa, tai reiškia, kad rašymo operacija priklauso vartotojo lygmens procesui. Tuomet IDS lygmuo tiesiog išsiunčia įspėjimą. Šio įspėjimo turinys yra panašus kaip ir pavojaus signalo, tačiau šis pranešimas turi vartotojo lygmens aplikacijos PID.

2.2.6. Lankstumas

Su nedideliais pakeitimais sistema gali aptikti kitokio tipo branduolio klaviatūros sekimo programas. Be *tty-buffed-based* klaviatūros sekimo programos yra du kiti metodai pavogti klavišų duomenis. Pirmas būdas per skenuotus kodus arba eilių derinius, antras būdas skaitant buferio funkcijas ir skaitymo sistemos skambučius.

2.3.Projekto išvados

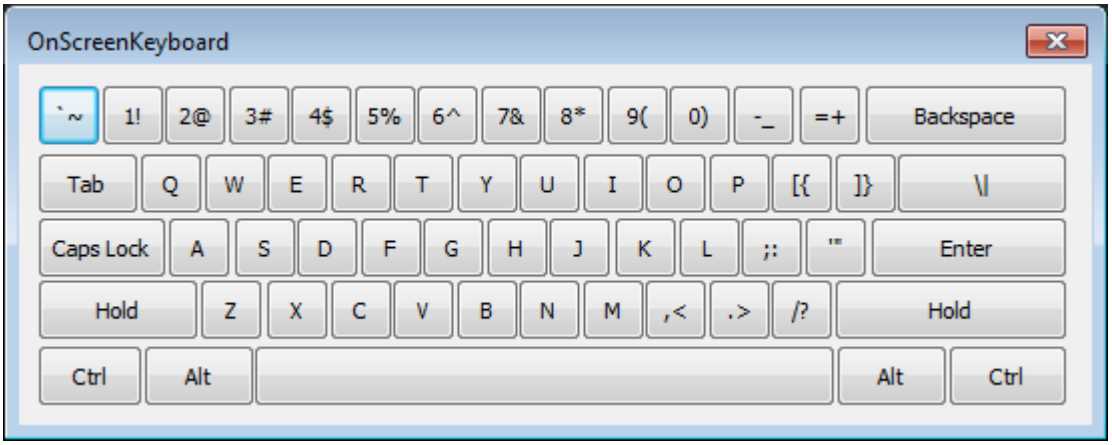
1. Šiuo metu sukurtos ir naudojamos apsisaugojimo priemonės ne visada gali apsaugoti nuo visų klaviatūros sekimo pavojų. Šiame darbe suprojektuota virtuali klaviatūra. Naudojant virtualią klaviatūrą apsisaugojama nuo aparatinio lygmens klaviatūros sekimo.
2. Kuriama virtuali klaviatūra turi kelias papildomas funkcijas, kurios skirtos apsisaugoti nuo klaviatūros sekimo. Kadangi nemažai klaviatūros sekimo programų gali daryti ekrano nuotraukas, perimti klavišų paspaudimus, įjungus kuriamos virtualios klaviatūros aplikaciją bus uždraudžiama daryti ekrano nuotraukas.
3. Piktavaliai gali bandyti nustatyti paspaustus klavišus ekrane pagal jų išsidėstymą. Šiame darbe siūloma funkcija, kuri priverčia virtualią klaviatūrą kiekvieno paleidimo metu ekrane atsirasti atsitiktinėje vietoje. Taip piktavalis negali nustatyti kur ir koks simbolis buvo įvedamas ekrane pagal x ir y koordinates.
4. Išanalizavus klaviatūros sekimo priemones nustatyta, kad dauguma programų, fiksuodamos paspaustus klavišus, nežino kur jis buvo vedamas. Dėl šios priežasties siūloma funkcija, kuri sugeneruoja simbolių masyvą atsitiktine tvarka ir siunčia juos į tekstinę bylą. Sukurta tekstinė byla vėliau turėtų būti ištrinama.
5. Išanalizavus branduolyje veikiančias klaviatūros sekimo programas, nustatyta, kad apsisaugoti nuo jų yra tikrai sudėtinga, kadangi jos veikia žemiausiame mašinos lygmenyje ir pastebėti jas sudėtingą. Dėl šios priežasties pasiūlytas branduolio lygmenyje veikiantis dinaminio žymėjimo metodas padedantis aptikti branduolio lygmenyje vykstantį klaviatūros sekimą.

3. KLAVIATŪRA ĮVEDAMOS INFORMACIJOS PROGRMINĖS APSAUGOS REALIZACIJA

Darbo metu sukurta virtuali klaviatūra, kurios klavišus spaudžiame pasinaudojant kursoriaus pagalba, taip išvengiame naudojimosi fizine klaviatūra ir apsisaugojame nuo aparatinio lygio klaviatūros sekimo. Tačiau tai nepadeda pilnai apsisaugoti, kadangi šiais laikais dauguma mobiliųjų įrenginių tokių kaip išmanieji telefonai naudoja lietimui jautrius ekranus ir juose jau yra virtualios klaviatūros, kurias gali stebėti įsibrovėlis fiksuodamas ekrane paspaustos vietos koordinates ir daryti ekrano nuotraukas. Jie tampa vis labiau patrauklūs piktavaliams. Kadangi jie naudojami kasdien įvairiose viešose vietose − tampa lengviau pažeidžiami.

3.1.Virtualios klaviatūros sukūrimas

Virtualiai klaviatūra pav.3.1. sukurta pasinaudojant C++ programavimo kalba ir pritaikyta “Windows” OS, kadangi tai populiariausia OS personaliniuose kompiuteriuose. Kuriant Virtualią klaviatūrą naudojomės MFC C++ programavimo kalbos biblioteka (angl. *The Microsoft Foundation Class (MFC) Library*) [44]. Kadangi tai žemo lygio programavimo kalba turinti daug funkcionalumo ir pritaikymo galimybių.



3.1. pav. Virtuali klaviatūra

3.1.1. Virtualios klaviatūros atsiradimas ekrane atsitiktine tvarka

Pasinaudoję „cstdlib“ C++ programavimo kalbos biblioteka pritaikėme programavimo funkcijas, kad jos nustatytų esamo ekrano dydį tam, kad klaviatūra neatsirastų už jo ribų. Tuomet pritaikėme algoritmą, kad kiekvieną kartą įjungus programą virtuali klaviatūra atsirandi kitoje vietoje pav. 3.2, kad piktavalis negalėtų jos aptikti.

3.2. pav. Pozicijos pasikeitimas

- duodama pradinė reikšmė *rand()* funkcijai;
- nustatomas ekrano dydis: cx tai x koordinatė, cy tai y koordinatė;
- parenkamos atsitiktinės x ir y koordinatės pasinaudojant *rand()* funkciją;
- nustatoma lango pozicija.

```
BOOL COnScreenKeyboardDlg::OnInitDialog()
{
srand(time(0));
int cx = GetSystemMetrics(SM_CXVIRTUALSCREEN);
int cy = GetSystemMetrics(SM_CYVIRTUALSCREEN);
int x = rand() % cx + 1;
int y = rand() % cy + 1;
SetWindowPos(&wndBottom, x, y, 0, 0,    SWP_NOSIZE);
ShowWindow(SW_SHOW);
return TRUE;
}
```

3.3. pav. Virtualios klaviatūros atsitiktinis atsiradimas ekrane

Pasinaudojant šia funkcija virtuali klaviatūra kiekvieną kartą ekrane atsiranda atsitiktine tvarka. Taip neleidžia nustatyti koks klavišas buvo paspaustas ekrane.

3.1.2. Ekrano vaizdo fotografavimo uždraudimas

Jei sistema gauna signalą, kad paspaustas mygtukas darantis ekrano nuotraukas:

- *OpenClipboard()* funkcijos pagalba atidaroma iškarpinė (angl. *clipboard*);
- *EmptyClipboard()* tuomet ištrinamas iškarpinės turinys;
- *CloseClipboard()* iškarpinė uždaroma.

Taip įsibrovėlis nebegali daryti ekrano nuotraukų, kadangi sistema gauna signalą, kad buvo paspaustas ekrano vaizdo fotografavimo klavišas. Tai padarome pasinaudojant funkcija *(GetAsyncKeyState(VK_SNAPSHOT))*.

```
BOOL CDialog::PreTranslateMessage(MSG *pMsg)
{
    if (GetAsyncKeyState(VK_SNAPSHOT))
    {
      OpenClipboard();
      EmptyClipboard();
      CloseClipboard();
    }
    return 0;
}
```

3.4. pav. Ekrano nuotraukų darymo uždraudimas

Tuomet iškarpinė atidaroma, ištrinama ir uždaroma − taip ji lieka tuščia ir padaryta ekrano vaizdo nuotrauka dingsta. Todėl nustatyti kuri vieta ekrane buvo paspausta − nebeįmanoma.

3.1.3. Atsitiktinių simbolių siuntimas

Norint suklaidinti piktavalius, kiekvieną kartą kuomet yra paspaudžiamas klavišas virtualioje klaviatūroje nusiunčiami atsitiktinių simbolių masyvas naudojant *srand(time(0))* bei *rand()* funkcijas, kurių pagalba sudaromas masyvas. Jis siunčiamas į tekstinį failą. Taip yra suklaidinimas piktavalis kadangi nebeįmanoma atskirti kur kurie simboliai buvo vedami.

```
void file_append()
{
    srand(time(0));
    char filename[] = "C:/Users/A/Desktop/temp.txt";
    fstream appendFileToWorkWith;
    appendFileToWorkWith.open(filename,   std::fstream::in   |
std::fstream::out | std::fstream::app);
    if (!appendFileToWorkWith)
    {
      string Str;
      for (unsigned int i = 0; i < (rand() % 30 + 1); ++i)
```

```
            {
                 Str += genRandom();
            }
            appendFileToWorkWith.open(filename,     fstream::in     |
     fstream::out | fstream::trunc);
            appendFileToWorkWith << Str;
            appendFileToWorkWith.close();
          }
          else
          {
            string Str;
            for (unsigned int i = 0; i < (rand() % 30 + 1); ++i)
            {
                 Str += genRandom();
            }
            appendFileToWorkWith << Str;
            appendFileToWorkWith.close();
            cout << "\n";
          }
     }
```

3.5. pav. Atsitiktinių simbolių siuntimas į failą

- Darbalaukyje sukuriama tekstinė byla "*temp.txt*";
- Kiekviena kart paspudus simbolį papildomas atsitiktina simbolių masyvu;
- Simboliai generuojami nuo vieno iki trisdešimties simbolių;

```
void COnScreenKeyboardDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
      CAboutDlg dlgAbout;
      dlgAbout.DoModal();
    }
    else
    {
      CDialog::OnSysCommand(nID, lParam);
    }
    remove("C:/Users/A/Desktop/temp.txt");
}
```

3.6. pav. Atsitiktinių simbolių failo ištrinimas

Kuomet programa išjungiama, byla su atsitiktinių simbolių masyvu, gautu kuomet spaudžiami klavišai virtualioje klaviatūroje, ištrinama.

3.2.Apsauga nuo branduolio lygmenyje veikiančių klaviatūros sekimo programų

Kuomet klavišų duomenys yra įvesti į klaviatūros tvarkyklę (pav 2.4. nr. 1), OS lygmens trigeris yra aktyvuojamas tam, kad perduoti klavišų informaciją iš OS lygmens į IDS lygmenį. IDS

lygmenyje perduodant duomenis jie yra pažymimi. Duomenys pažymimi norint stebėti duomenų sklidimą (pav 2.4. nr.2). Jeigu su pažymėtais duomenimis yra atliekamos įtartinos operacijos, tokios kaip kopijavimas, perkėlimas, rašymas − OS lygmuo išsiųs duomenis (pav 2.4. nr.3) į IDS lygmenį analizavimui (pav 2.4. nr. 4). Remiantis pažymėtų duomenų įrašais − IDS atakos analizės komponentai perspėja apie įtartinas operacijas (pav 2.4. nr. 5).

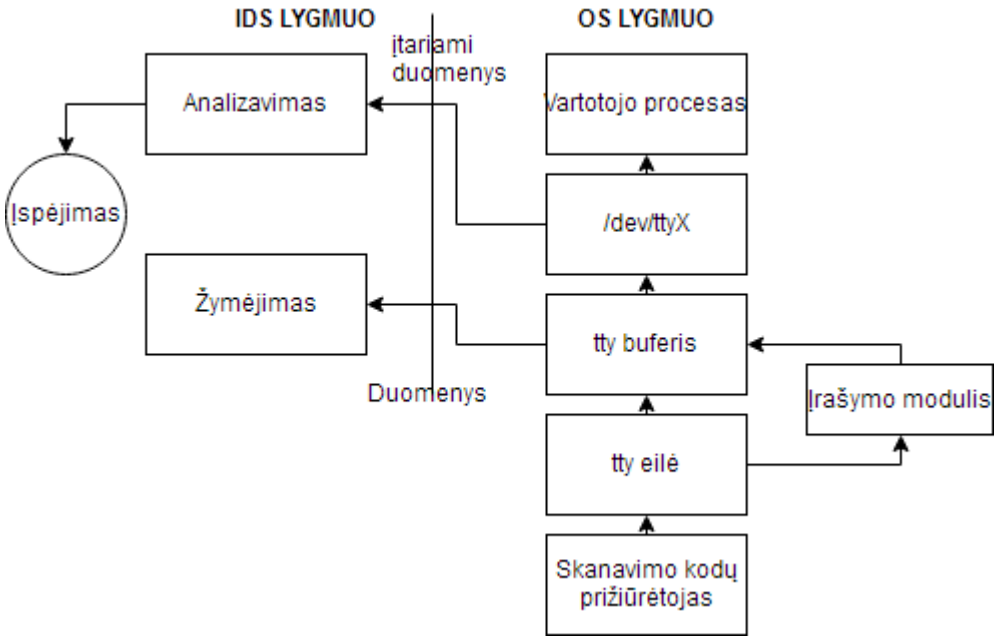
3.2.1. OS lygmuo

Tty buferis yra buferio simbolis, kurio dydis yra 512 baitų. Šis buferis yra dalis tty prietaiso. Abiejuose "Linux" branduoliuose (2.4 ir 2.6) tty įrenginys atstovauja *tty_struct* struktūrą. Mes pridedame trigerį į *n_tty_receive_buf()* funkciją norint gauti klavišų duomenis. Ši funkcija remiasi terminalo tvarkykle tam, kad saugotų klavišų duomenis į tty buferį. *tty_struct* funkcijoje galime perimti klavišų duomenis ir korespondento atminties adresą tty buferyje [45]. Perduoti duomenis į IDS lygmenis žymėjimui mes naudojome naują sistemos kvietimą, kuris iškviečiamas naudojant trigerį.

```
struct passed_data
{
dt_content;
dt_addr;
pro_id;
Is_written;
};
```

3.7. pav. Duomenų perdavimas tarp dviejų lygmenų

Pav. 3.7. vaizduoja duomenų perdavimą iš OS lygmens i IDS lygmenį. Struktūroje *passed_data***,** naudojam *is_written* žymą bei esamo proceso ID (PID), tam kad stebėti operacijas atliktas su duomenimis. Žyma *is_written* nustatoma tada, kai duomenys paliečiami rašymo operacijos. Tam, kad gautume PID mes naudojame current → pid**,** o vietoj sistemos kvietimo *getpid()*. PID nurodo procesą kuris iškviečia *n_tty_receive_buffer()* funkciją, tam kad išsaugoti klavišų duomenis į tty buferį. Šis PID yra toks pat kaip vartotojo lygmens proceso PID, kuris kontroliuoja klavišų duomenis iš skanavimo kodo prižiūrėjimo į vartojo lygmenį.



3.8. pav. Sistema su įrašymo moduliu OS lygmenyje ir analizavimo moduliu IDS lygmenyje

IDS gali stebėti ne tik klaviatūros duomenis, bet ir kitų rūšių − pavyzdžiui duomenis iš kitų įrenginių, tinklų, ar vartotojo lygio programas. Todėl mūsų sistemoje klaviatūros duomenys turi būti siunčiami IDS lygmenyje tam tikru būdu. "Linux" branduolyje, IRQ [46] ir registrai yra naudojami keistis duomenimis.

- IRQ "Linux" branduolyje IRQ gali būti naudojamas sinchronizuoti įvykius tarp branduolio komponentų. Mūsų sistemoje klavišo duomenų parengimas tty bufferyje laikomas įvykiu. Kuomet šis įvykis nutinka mes naudojame IRQ trigerį tam, kad kad siųsti duomenis. Šio IRQ pasirinkimas turėtų ne konfliktuoti su kitais naudotais IRQ. Pavyzdžiui IRQ 0x80 reguliariuose sistemos kvietimuose. Sistemoje mes naudojame IRQ 0x82 dėl jos prieinamumo.
- Registrai naudojami saugoti duomenims, kuriais keičiamasi branduolio komponentuose. Pavyzdžiui saugoti sistemos kvietimo parametrus (šie registrai "Linux" branduolyje naudojami didėjančia tvarka): *%eax, %ebx, %ecx, %edx, %esi, %edi, ir %ebp*. Kadangi IDS reikia užfiksuoti duomenų apsikeitimą per sistemos kvietimus, todėl priskiriamas konkretus registras šiam tikslui. Dėl riboto registrų skaičiaus 80x86 architektūroje priskyrimas turi tenkinti dvi sąlygas [32]: kiekvienas parametras negali būti ilgesnis nei registro ilgis (32 bitai 80x86 procesoriuje) ir parametrų skaičius negali viršyti šešių. Mūsų sistemoje kvietimų skaičius pereina per `%eax`, kol kiti `passed_data` elementai yra saugomi *%ebx, %ecx, %edx, %esi, ir %edi*. Kaip parodyta pav. 3.3**.** įtartini duomenys yra perduodant iš OS lygmens į IDS lygmenį. Įtartini duomenys gali būti naudojami atminties operacijose arba rašymo operacijose. Toliau mes aprašome du scenarijus detaliau.

Pirmiausia, klavišų duomenys tty buferyje gali būti naudojami kito branduolio funkcijos kvietimui. Šių funkcijų kvietimai gali kopijuoti arba judinti klavišų duomenis į kitą atminties vietą naudojant primityvias branduolio dideles funkcijas − tokias kaip **memcpy(), *memmove()*. Tam, kad atlaisvinti šias išskirtas branduolio atminties vietas funkcija *kfree()* yra iškviečiama. Tarkime, kad klaviatūros paspaudimų registravimo programos žino atminties vietą ir gali įrašyti duomenis. Tokiu atveju mes turime kontroliuoti duomenų sklidimą. Norint stebėti duomenis naudojame trigerį, kuris perduoda su juo susijusią informacija IDS lygmeniui analizuoti. Šios funkcijos gali būti naudojamos kitų procesų, todėl IDS lygis turi patikrinti ar duomenys buvo pažymėti anksčiau. Tam, kad padėti IDS lygmeniui stebėti duomenų sklidimą, mes turime nurodyti trijų atminties funkcijų adresus. Tam, kad atskirti tris atminties funkcijas naudojame funkcijos identifikavimo numerį. Tokiu būdu šaltinio adresas ir funkcijų indikatorių numeris yra pridedamas į *passed_data*.

Antra, kad įrašyti klavišų duomenis, klaviatūros sekimo programos gali naudoti vieną iš dviejų rašymo funkcijų: *tty_core_write()* ir *sys_write()*. Formuotojas naudojamas įrašyti pažymėtus duomenis į tty prietaisus, kurie vėliau naudojami į failų sistemą arba tinklą I/O. Perimti rašymo operacijas, kurios atliekamos su pažymėtais duomenimis, naudojame trigerį, kad persiųsti su pažymėtais duomenimis susijusią informaciją į IDS norint ją išanalizuoti. Tam, kad padėti IDS lygmeniui diferencijuoti įtariamus duomenis nuo orginalių klavišų duomenų kurie turi būti pažymėti, *is_written* žyma yra nustatoma *passed_data*.

```
asmlinkage int sys_passing(data, addr){
passing(data, addr, current_id, is_written);
}
asmlinkage int sys_propgt(data, addr, sr_addr, m_id){
propgt(data,addr,current_id,sr_addr,m_id);
}
_sysKBCall(type, name, data, addr, id, arg4, arg5){\
long __res; \ __asm__ volatile ("int $0x82"\
: "=a" (__res)\
: "" (__NR_##name), "b" (data), \
"c" (addr), "d" (id), "S" (arg4), "D" (arg5)); \
}
```

3.9. pav. OS lygio aukšto lygio trigerio funkcijos apibūdinimas

Pav 3.9. vaizduoja OS trigerio aukšto lygio funkcijos apibūdinimą. Du nauji sistemos kvietimai *sys_passing()* ir *sys_propgt()* primityviai iškviečia programą *_sysKBCall()*. Funkcija *sys_passing()* yra

aktyvuojama remiantis dviem įvykiais: pasirengimas klavišų duomenų tty buferyje ir įvykus rašymo operacijai atliekamai su įtariamais duomenimis. Funkcija *sys_propgt()* yra vykdoma remiantis atminties operacija iš padaugintų duomenų. Šaltinio adresas ir funkcijos identifikavimo numeris yra reprezentuojamas *sr_addr ir m_id***.**

3.2.2. IDS lygmuo

Žymėjimas - pirmiausia, žymėjimo komponentai patikrina passed_data gautus per trigerį sys_passing. Šie duomenys turi būti pažymėti jeigu jų nenaudojo rašymo operacijos. Tuomet žymėjimo komponentas patikrina įtartinus duomenis gautus per trigerį `sys_propgt`. Remiantis šaltinio adresu, mes patvirtiname, ar duomenys pažymėti, ar ne. Šie duomenys yra ignoruojami jeigu šaltinio adresas nėra pažymėtų duomenų lentelėje. Jeigu šaltinio adresas yra lentelėje − žymėjimo komponentas atskiria gautus duomenis remiantis funkcijos rodikliais. Funkcijai **memcpy()*, gauti duomenys yra pažymėti ir išsaugoti pažymėtų duomenų lentelėje. Funkcijai **memmove()* duomenys naujoje atminties vietoje yra pažymėti ir seni duomenys senoje atminties vietoje atžymimi. Galiausiai, *kfree()* funkcijai mums reikia atžymėti pažymėtus duomenis ištrinant žymą pažymėtų duomenų lentelėje.

Išsaugojant PID pažymėtus duomenis duomenų lentelėje yra susietas su vartotojo lygmens programomis tam, kad leistų analizuoti ir kontroliuoti klavišų duomenų srautą. Žymėjimo proceso metu atminties operacijų funkcijos **memcpy()*ir **memremove()* gali būti iškviestos procesų susietų su PID. Šie procesai skiriasi nuo rašymo operacijų procesų klaviatūros sekimo programose.

Aukšto lygio funkcijos trigerio aprašymas IDS lygmenyje yra pavaizduotas priede nr. 1. Manoma, kad 253 ir 254 yra sistemos trigerių skambučių numeriai *sys_passing()* ir *sys_porpgt()*, per prižiūrėtoją *handle_keyboard_syscalls()*, IDS lygmuo aktyvuoja žymėjimo ir analizavimo komponentus šiem numeriam.

3.2.3. Lankstumas

Reguliavimo skenavimo kodai arba eilių kombinacijos (angl. *Handling scancodes or queuing combinations*): Stebėti ir pakeisti šias funkcijas po to, kai *handle_scancode()* funkcija yra atlikta, skanavimo kodai ir klavišo duomenys yra pažymėti ir stebimi. Norint stebėti duomenis mums reikia realizuot trigerį viduje *handle_scancode()* . Panašiai atakos gali būti aptiktos remiantis rašymo operacijomis, kurios atliekamos su skanavimo kodų ir klavišo paspaudimų žymomis pažymint duomenis.

Skaitymo buferis ir skaitomi sistemos kvietimai (angl. *Reading buffer ir reading system call*)**:** Klaviatūros sekimo programos gali bandyti perimti ir įrašyti *sys_read()* sistemos kvietimą ir *read()* funkcijos išvestį. Tam, kad aptikti neteisėtas operacijas, mums reikia pažymėti įvesties duomenis prieš

tai kai jie yra naudojami. Klaviatūros duomenys yra funkcijos įvesties duomenys. Mums taip pat reikia patikrinti, ar pažymėti duomenys nėra naudojami rašymo operacijos.

3.3.Išvados

1. Sukuriama virtuali klaviatūra, kuria bus vedama konfidenciali informacija. Virtuali klaviatūra apsaugo ne nuo visų klaviatūros sekimo problemų, todėl pridedame kelias papildomas saugumo funkcijas.
2. Tai ekrano vaizdo nuotraukų darymo blokavimas, kadangi nemažai naujų kenkėjiškų programų gali tai daryti ir pagal tai nustatyti kokie klavišai buvo paspausti.
3. Kiekvieną kartą virtualios klaviatūros programos įjungimo metu ji atsiranda atsitiktinai parinktoje vietoje tam, kad piktavalis negalėtų nustatyti kurie simboliai buvo paspausti pagal x ir y koordinates.
4. Pritaikomas algoritmas siųsti atsitiktinių simbolių masyvą į kitą failą, taip suklaidinant piktavalį ir nepaliekant galimybės nustatyti koks klaviatūros simbolis buvo paspaustas.
5. Branduolio lygmenyje panaudojant IDS ir dinaminio žymėjimo metodą, sukuriama sistema, padedanti aptikti piktavališkus veiksmus su klaviatūros tvarkykle. Sistema pritaikoma “Linux” operacinei sistemai, kadangi ji yra atvirojo kodo ir žinoma kaip veikia jos branduolys.
6. Dinaminio žymėjimo pagalba pažymime klaviatūros įvestus duomenis ir stebime ar jos sraute nebuvo atlikta jokių pakeitimų viso klaviatūros duomenų keliavimo metu.

4. KLAVIATŪRA ĮVEDAMOS INFORMACIJOS APSAUGOS TOBULINIMO EKSPERIMENTINIS TYRIMAS

Tyrimo metu yra tikrinama kaip pasikeitė klavišų aptikimas naudojant apsaugos sistemą. Apsaugos nuo klaviatūros sekimo sistemos įvertinimo metu bus analizuojama: kaip pagerėjo aptikimo tikslumas, ar įgyvendintos saugumo sritys padeda apsisaugoti nuo klaviatūros sekimo, kaip padidėjo CPU apkrova naudojant mūsų klaviatūros sekimo programų aptikimo sistemą. Taip pat patikrinta kaip pagerėjo klaviatūros duomenų apsauga naudojant mūsų apsaugos sistemą.

4.1.Aptikimo tikslumas

Patikriname mūsų prototipo efektyvumą tokioje aplinkoje, kurioje jau yra veikianti branduolio klaviatūros sekimo programa “Vlogger” ir 11 nepiktavališkų aplikacijų. Vlogger yra pažangi Linux branduolio klavišų stebėjimo programa, kuri gali stebėti administratoriaus ir paprasto vartotojo klavišus, konsolės pagalba gali stebėti vietines ir nuotolines sesijas (*telnet, ssh*). Lyginant su kitomis žinomomomis branduolio lygmenyje veikiančiomis klaviatūros sekimo programomis, tokiomis kaip: ttyrpld [47] arba Sebek [48]**,** “Vlogger” . “Vlogger” yra daugiau bendrinė, galingesnė ir sunkiau aptinkama. Geranoriškos programos esančios darbinėje aplinkoje − trys interneto naršyklės (Konqueror, Opera, FireFox), keturi teksto redaktoriai (Vim, Nedit, KWord bei OpenOffice) ir keturios žinučių siuntimo programėlės (Gaim, Skype, aMSN ir Kopete). Jos visos sukelia įvairią klaviatūros sąveiką. Mes naudojame šias programas tikrinant ar klavišai nėra netinkamai klasifikuojami kaip branduolio klaviatūros sekimo programų veikla.

Pirmiausia patikriname, kad be mūsų aptikimo sistemos originalus Argos IDS negali aptikti “Vlogger”, nepaisant ar kitos programos veikia ar ne. Šio rezultato tikimąsi, kadangi Argos skirta aptikti kankėjišką duomenų panaudojimą − tokį kaip šuolius, funkcijos adresus ir instrukcijas, bet ne aptikti klavišų vagystes. Tuomet mes įvertiname mūsų sistemos aptikimo tikslumą pagal tris kriterijus: tik veikiant “Vlogger”, tik su geranoriškom programom, bei veikiant geranoriškom ir “Vlogger” programom. Įvertinimas pateiktas lentelėje 4.1.

4.1. lentelė Klaviatūros sekimo aptikimas branduolio lygmenyje

Scenarijus	Klaviatūros sekimo programų aptikimas	Neatitikimai
Veikiant tik “Vlogger”	100%	0%
Veikiant tik geranoriškom programom	N/A	0%
“Vlogger” ir geranoriškos programos	100%	0%

Tik su "Vlogger": kuomet veikia tik "Vlogger" "Linux" sistemoje − mūsų aptikimo sistema visada gali aptikti „Vlogger" egzistavimą, kuris piktavališkai kopijuoja klaviatūros tvarkyklės duomenis iš tty buferio ir rašo juos į žurnalo failą. Mes pažymime įspėjimus "Vlogger" saugojime, kuomet klaviatūros mygtukai nuspausti. Kiekvienam veiklos fiksavimui mes generuojame įspėjimo įrašą kuris turi "Vlogger" branduolio proceso ID, buferio turinį, ir susijusios atminties adresą. Norint papildomai įvertinti mūsų sistemos aptikimo pajėgumus, taip pat įvykdome įvairius nuotolinius SSH prisijungimus prie mūsų sistemos. SSH klientas prisijungia prie SSH serverio per pseudo tty prižiūrėtoją, kuris veikia kaip tarpininkas tarp nuotolinės aparatinės įrangos prietaisų ir tikros tty tvarkyklės. Tokiu atveju, nuotolinės sesijos klavišai yra perduodami tty buferiui ir juos gali užfiksuoti „Vlogger". Vis dėlto, mūsų aptikimo sistema gali aptikti visą "Vlogger" veiklą SSH sesijoje.

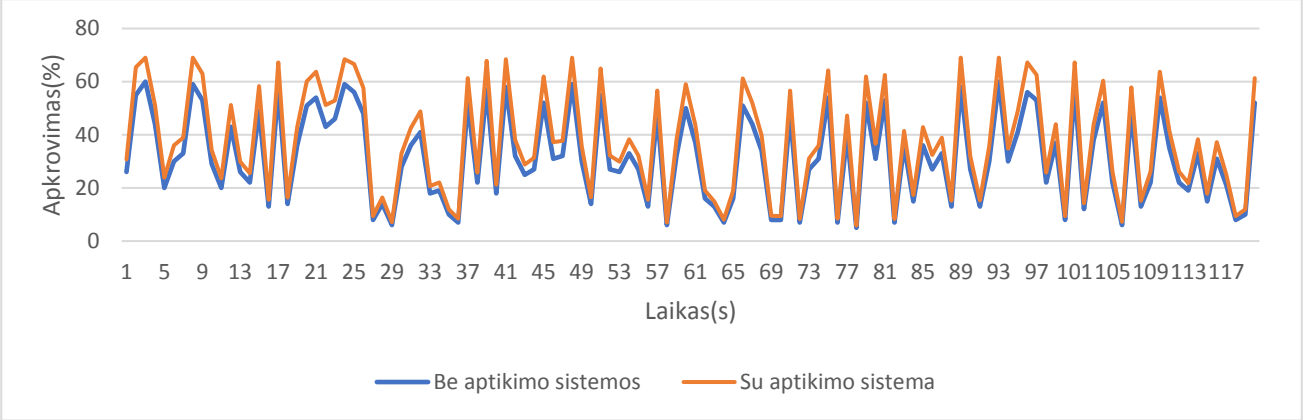
Tik su geranoriškomis programomis: kuomet "Vlogger" neaktyvuotas ir tik geranoriškos programos veikia „Linux" sistemoje, galimos žemo lygio operacijos su klavišų duomenimis, kurias sukelia šios aplikacijos ir jas visada galima teisingai identifikuoti. Atlikus daugybę bandymų, paleidus 11 geranoriškų programų su skirtingais klaviatūros aktyvumais, mūsų aptikimo sistema neteikia jokių įspėjimų apie netinkamą veiklą.

Su "Vlogger" ir geranoriškom programom: kuomet "Vlogger" yra aktyvuota ir paleistos geranoriškos programos „Linux" sistemoje − mūsų aptikimo sistema visada gali aptikti "Vlogger" neteisėtą klavišų fiksavimą ir pateikti pranešimą. Įtartini veiksmai su klavišų duomenimis visada gali būti aptikti naudojantis mūsų aptikimo sistemą ir suskirstyti į įspėjimus ir pranešimus.

4.2.CPU išnaudojimas

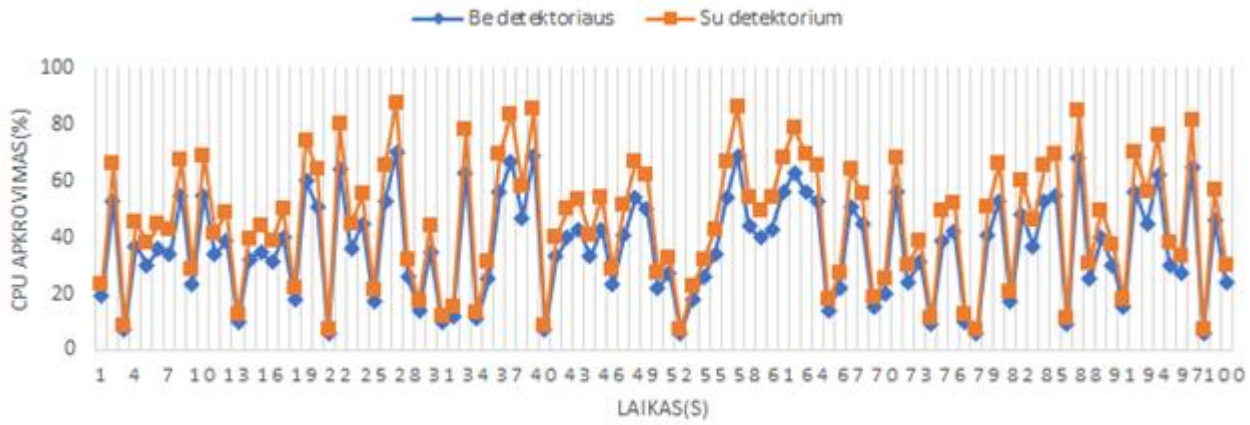
Argos veikimas [43], parinkus didžiausio greičio konfiguraciją, yra lėtesnis nei vanilla QEMU [49]. Šis skaičius beveik nepakinta kuomet taikoma mūsų aptikimo Sistema. Tai vyksta dėl dviejų priežasčių. Pirmiausia, klavišų duomenys užima mažai vietos, todėl mūsų aptikimo sistema gali efektyviai perduoti, žymėti ir atžymėti duomenis. Taip pat, klavišų paspaudimų dažnis yra ganėtinai žemas, taigi, bendras aptikimo uždavinys nėra sunkus. Atlikto tyrimo metu nustatyta, kad vidutiniškai vartotojas suveda 33 žodžius per minutę [50].

Tam, kad turėtume aiškius rezultatus apie mūsų aptikimo sistemos veikimą su Argos sistema, mes įvertiname branduolio lygmens CPU apkrovimą įjungimo metu ir veikimo metu. Paleidimas vidutiniškai užtrunka 120 sekundžių, pav. 4.1. pavaizduota kaip apkrautas CPU paleidimo metu kiekviena sekundę, viena kreivė žymi paleidimą su mūsų aptikimo sistema kita be mūsų aptikimo sistemos. Matyti, kad CPU apkrova padidėja, padidėjimo vidurkis - 18%.



4.1. pav. CPU apkrovimas OS krovimo metu

Veikimo metu, pamatuojamas CPU apkrovimas, kuomet „KidLogger“ ir geranoriškos programos yra įjungtos. Sistema laikoma įjungta 5 minutes ir CPU apkrova tuo metu matuojama. Pav. 4.2. atvaizduoja CPU apkrovimą su arba be aptikimo sistemos.



4.2. pav. Darbo metu CPU apkrovimas

CPU skirtomo vidurkis padidėjo iki 24% kuomet aptikimo sistema yra įjungta.

4.3.Klaviatūros sekimo programų suklaidinimas

Realizavę apsaugos nuo klaviatūros sekimo patobulinimus patikrinome juos su trijomis klaviatūros sekimo programomis:

- Actual Keylogger;
- G³ iSam;
- “KidLogger”.

4.2. lentelė Klaviatūros sekimas be apsaugos sistemos

	Actual Keylogger	G³ iSam	KidLogger
Virtualios klaviatūros vietos nustatymas	Taip	Taip	Taip
Ekrano nuotraukų darymas	Taip	Taip	Taip
Simbolių atrinkimas	Taip	Taip	Taip

4.2. Lentelėje pateikti klaviatūros sekimo rezultatai ir galimybės be mūsų apsaugos priemonės. Be apsaugos sistemos klaviatūros sekimo programos gali atlikti šias funkcijas:

- Klaviatūros sekimo programa G³ iSam [51] turi galimybę užfiksuoti klavišus bei daryti ekrano nuotraukas;
- Spyrix Free Keylogger [52] yra bandomoji versija Spyrix Personal Monitor programos. Ji gali fiksuoti klavišų paspaudimus, kopijuoti ir daryti ekrano nuotraukas;
- KidLogger [53] − tai absoliučiai nemokama ir atviro kodo vartotojo stebėjimo programa. Gali ne tik fiksuoti klavišus, kopijuoti, daryti ekrano nuotraukas, bet ir įrašinėti USB / failų ir aplankalų naudojimą, mikrofono garsą.

4.3. lentelė Klaviatūros sekimo galimybės naudojant mūsų apsisaugojimo priemones

	Actual Keylogger	G³ iSam	KidLogger
Virtualios klaviatūros vietos nustatymas	Ne	Ne	Ne
Ekrano nuotraukų darymas	Ne	Ne	Ne
Simbolių atrinkimas	Ne	Ne	Ne

Atliktas tyrimas parodė, kad sistema gali padėti apsisaugoti nuo klaviatūros sekimo taikomajame lygmenyje, tai pateikta 4.3. lentelėje. Šnipinėjimo programos negali daryti ekrano nuotraukų, nustatyti virtualios klaviatūros vietos ekrane ir atrinkti simbolių, dėl papildomai gaunamų

sugeneruotų simbolių masyvų.

4.4.Eksperimentinio tyrimo išvados

1. Atliktas tyrimas, kurio metu nustatyta, jog naudojant branduolyje veikiančių klaviatūros šnipinėjimo programų aptikimo sistemą, padidėja CPU apkrovimas.
2. Išanalizavus realizuotų sistemų veikimo principus gauti rezultatai parodė, jog aptikimo sistema veikia ir geba aptikti sistemoje nesankcionuotą veiklą. Atskirti piktavališkus veiksmus nuo normalaus sistemos veikimo. Geba atskirti piktavališkas programas nuo geranoriškų programų.
3. Atlikto tyrimo metu nustatyta, kad sukurta apsauga nuo klaviatūros šnipinėjimo veikia. Sukurta virtualios klaviatūros sistema padeda apsisaugoti nuo klaviatūros sekimo. Ekrano nuotraukų uždraudimas veikia, piktavalis daryti ekrano nuotraukų nebegali, taip pat nebegali rinkti informacijos apie darbalaukį.
4. Išanalizavus realizuotą sistemą nustatyta, kad virtualios klaviatūros atsiradimas ekrane padeda apsisaugoti nuo klaviatūros sekimo programų, kadangi nežinoma kurioje vietoje ji atsiranda − kursoriaus paspaustos pozicijos ekrane nustatyti nebegalima.
5. Virtualios klaviatūros paspausti klavišai siunčia sugeneruotus masyvus į tekstinę bylą ir kenkėjiška veikla sutrikdoma. Nebeįmanoma atstatyti simbolių kurie buvo paspausti vartotojo ir tų, kurie sugenertuoti. Piktavalis negali perimti vedamos informacijos. Sumažėja galimybė nustatyti paspausto klavišo reikšmę.

5. KLAVIATŪRA ĮVEDAMOS INFORMACIJOS APSAUGOS TOBULINIMO IŠVADOS

Baigiamojo darbo tikslas yra patobulinti esamas apsaugas nuo klaviatūros sekimo programų. Sukurti du metodai padedantys apsisaugoti nuo klaviatūros sekimo programų. Vienas metodas pritaikytas “Windows”, kitas “Linux” OS.

Realizavus metodus patikrintas jų veikimas, ar tai padeda apsisaugoti nuo klaviatūros sekimo programų, ar realizuotas metodas sugeba aptikti branduolyje veikiančias klaviatūros sekimo programas.

1. Išanalizuotos esamos klaviatūros sekimo priemonės, jų veikimo principai. Įsitikinta jog klaviatūros sekimas gali būti labai didelė problema vartotojui. Išsiaiškinta kaip veikia šnipinėjimo programos, kokius metodus naudoja perimti informaciją, koki metodai naudojami paslėpti kenkėjiškas programas, kad būtų sunkiau aptikti.
2. Dabartinės apsisaugojimo ir aptikimo priemonės nėra tobulos, yra įvairių metodų klaviatūros sekimo programom tapti nepastebimoms bei neaptinkamoms. Taikomajame lygmenyje veikiančias klaviatūros sekimo programas lengviau aptikti nei branduolyje veikiančias kenkėjiškas programas.
3. Sudaryti du apsisaugojimo metodai nuo klaviatūros sekimo programų. Taikomajame lygmenyje realizuota sistema bei pritaikyta “Windows” OS, kadangi tai populiariausia sistema personaliniuose kompiuteriuose. Virtuali klaviatūra kurios pagalba galime vesti duomenis saugiai, kaip parodė mūsų tyrimo rezultatai.
4. Sukurtas prototipas aptikti branduolyje veikiančias klaviatūros sekimo programas. Jos pagalba žymime klavišų duomenis, taip juos stebime, kokie pakeitimai buvo padaryti, taip galima aptikti šnipinėjimo programas, taip pat, stebėti klavišų duomenų srautą galime nustatyti kurioje vietoje pakeitimai buvo padaryti.
5. Ištirtas apsisaugojimo metodas virtuali klaviatūra. Sistemoje įdiegtos šnipinėjimo programas. Patirkinę virtulią klaviatūrą nsutatėme, kad apsisaugojimo priemonė veikia piktavalis negali perimti informacijos naudodamas taikomojo lygmens klaviatūros sekimo programas. Negali daryti ekrano nuotraukų, nustatyti kurie simboliai yra tikri, bei nustatyti klaviatūros vietos ekrane.
6. Išanalizuotas klaviatūros sekimo programų aptikimo metodas – dinaminis žymėjimas. Sistemoje įdiegtos šnipinėjimo programos veikiančios branduolio lygmenyje. Įjungta dinaminio žymejimo metodo programa. Pastebėta, kad sukurtas metodas aptinka ir atskiria piktavališkas programas. Tačiau padidėja CPU apkrovimas, kadangi tikrinamos klaviatūros

tvarkyklės funkcijos.

7. Tyrimas parodė, kad naudojant dinaminio žymėjimo metodą naudojanti sistema, aptinka klaviatūros sekimo programas. Atskiria geranoriškų programų veiklą nuo piktavališkų. Taip pat nustato, grėsmės šaltinį.

BIBLIOGRAFIJA

[1] S. Sagiroglu and G. Canbek, "Keyloggers," in *IEEE Technology and Society Magazine*, 2009.

[2] A. Emigh, "The Crimeware Landscape: Malware, Phishing, Identity Theft and Beyond," in *A Joint Report of the US Department of Homeland Security — SRI International Identity Theft Technology Council, the Anti-Phishing Working Group*, 2006, 2006.

[3] S. Shetty, „Introduction to Spyware Keyloggers," 13 Balandis 2005. [Tinkle]. Available: https://www.symantec.com/connect/articles/introduction-spyware-keyloggers. [Kreiptasi 12 Gegužė 2017].

[4] Mercenary, "packetstormsecurity.com," 26 sausis 2002. [Online]. Available: https://packetstormsecurity.com/UNIX/security/kernel.keylogger.txt. [Accessed 12 Gegužės 2017].

[5] B. S. a. C. O. M. Xu, "Internet and Multimedia Systems and Applications (IMSA)," in *Copyright and Security (IMSA)*, Honolulu, 2005.

[6] C. E. F. a. D. H. G. K. Subramanyam, "Keyloggers: The Overlooked Threat to Computer Security," in *Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, Denison, 2003.

[7] A. M. T. B. H. L. Steven D. Gribble, "A Crawler-based Study of Spyware in the Web," in *Proceedings of the 13th Annual Network and Distributed Systems Security Symposiu*, SanDiego, 2006.

[8] C.Santwana, Dr. S.Magesh, K. Sai Aditya , "Hypervisor based Mitigation Technique for Keylogger Spyware Attacks," in *International Journal of Computer Science and Information Technologies*, 2015.

[9] Jesus Navarro, Enrique Naudon, Daniela Oliveira, "Bridging the Semantic Gap to Mitigate Kernel-level Keyloggers," in *Computer Science Department Bowdoin College*, Brunswick ME USA.

[10] J. Canavan, "The Evolution of Malicious IRC Bots," in *Symantec Security Response*, Dublin, 2005.

[11] Hemita Pathak, Apurva Pawar, Balaji Patil, "A Survey on Keylogger: A malicious Attack," in *International Journal of Advanced Research in Computer Engineering & Technology*, 2015, 2015.

[12] Sonal Shinde, Ujwala H. Wanaskar, "Keylogging: A Malicious Attack," in *International Journal of Advanced Research in Computer and Communication Engineering*, 2016.

[13] Adrian Schipor, Alexandru Maximciuc, Cristina Vatamanu , "Inside Netrepser – a JavaScript-based Targeted Attack," Bitdefender LABS, 2017.

[14] T. Spring, "threatpost.com," 6 Spalis 2016. [Online]. Available: https://threatpost.com/web-based-keylogger-used-to-steal-credit-card-data-from-popular-sites/121141/. [Accessed 12 Sausis 2017].

[15] Stefano Ortolani, Cristiano Giuffrida, Bruno Crispo, "Bait your Hook: a Novel Detection Technique for Keyloggers," International Workshop on Recent Advances in Intrusion Detection , Berlin, 2010.

[16] Nairit Adhikary, Rohit Shrivastava, Ashwani Kumar, Sunil Kumar Verma,

Monark Bag, Vrijendra Singh, "Battering Keyloggers and Screen Recording," I. J. Computer Network and Information Security, 2012.

[17] Microsoft, "microsoft.com," - - -. [Online]. Available: https://msdn.microsoft.com/en-us/library/bb761584(VS.85).aspx. [Accessed 12 Vasario 2017].

[18] T. Olzak, "Keystroke logging (keylogging)," Adventures in Security, 2008.

[19] K. Chen, "Reversing and Exploiting an Apple Firmware Update," in *Black Hat USA*, Las Vegas, 2009.

[20] M. Newlin, "github.com," 23 Vasaris 2016. [Online]. Available: https://github.com/BastilleResearch/mousejack/blob/master/doc/pdf/MouseJack-whitepaper-v1.1.pdf. [Accessed 1 Gruodzio 2016].

[21] J. Kirk, "Swedish police warn of tampered credit card terminals," CSO, -, 2008.

[22] A. Kelly, "Cracking Passwords using Keyboard," University of Edinburgh, Edinburgh, 2010.

[23] S. Yang, "Berkeley.edu," UCBerkeleyNews, 14 Rugsėjis 2005. [Online]. Available: http://www.berkeley.edu/news/media/releases/2005/09/14_key.shtml. [Accessed 4 Rugsejis 2016].

[24] M. Vuagnoux and S. Pasini, "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards," in *USENIX Security Symposium*, Montreal, 2009.

[25] A. f. e. a. a. touchscreens, "Skimming with ATM Cameras," in *IEEE*, Melaka, 2011.

[26] Philip Marquardt, Arunabh Verma, Henry Carter, Patrick Traynor, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards," in *CCS '11*, Illinois, 2011.

[27] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers, 2nd Edition, Oreilly & Associates Inc, 2005.

[28] Kalpa Vishnani, Alwyn Roshan Pais, and Radhesh Mohandas , "An In-Depth Analysis of the Epitome of Online Stealth: Keyloggers; and Their Countermeasures," in *Dept. of Computer Science & Engg, National Institute of Technology Karnataka, Surathkal, Srinivasnagar,* , Mangalore , 2011.

[29] N. S.-D. Ilya Mironov, "Cryptographic Reverse Firewalls," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Berlin, 2015.

[30] "Independent One-Time Passwords," in *Proceedings of the Fifth USENIX UNIX Security Symposium Salt Lake City*, Salt Lake City, 1995.

[31] K. Matsuura, "Security Tokens and Their Derivatives," in *In 7th International Conference of the Society for Computational Economics*, 2001, 2001.

[32] D. Bovet and M. Cesati, Understanding The Linux Kernel, Sebastopol, California: Oreilly & Associates Inc, 2005.

[33] D. Brumley, J. Newsome, D. Song, H. Wang, and S. Jha., "Towards automatic generation of vulnerability-based signatures," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06)*, Washington, 2006.

[34] André Gustavo Adami, "Automatic Speech Recognition: From the Beginning to the Portuguese," in *Universidade de Caxias do Sul, Centro de Computação e Tecnologia da Informação Rua Francisco Getúlio Vargas*, Caxias do Sul.

[35] D. Florencio and C. Herley. Klassp:, "Entering passwords on a spyware infected machine using a shared-secret proxy," in *ACSAC '06: Proceedings of the 22nd*

Annual Computer Security Applications Conference on Annual Computer Security Applications Conference, Washington, 2006.

[36] Donghai Tian, Xiaoqi Jia, Junhua Chen, Changzhen Hu, "An Online Approach for Kernel-level Keylogger," in *Preprint submitted to JISE*, 2016.

[37] James Newsome, Dawn Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software," in *School of Computer Science Carnegie Mellon University Pittsburgh*, Pittsburgh, 2004.

[38] Webroot, "webroot.com," webroot, 2004-2017. [Online]. Available: https://www.webroot.com. [Accessed 8 Kovas 2016].

[39] T. Micro, "trendmicro.eu," Trend Micro, [Online]. Available: http://www.trendmicro.eu/index.html. [Accessed 8 Kovas 2016].

[40] H. Yin, Z. Liang, and D. Song, "Hookfinder: Identifying and understanding malware hooking behavior.," in *NDSS - Proceedings of the 15th Annual Network and Distributed System Security Symposium*, San Diego, 2008.

[41] J. Butler and G. Hoglund, "VICE," in *Black Hat USA 2004 Conference*, Las Vegas, 2004.

[42] Adaware, "Adaware," Adaware, [Online]. Available: https://www.adaware.com/. [Accessed 16 Gruodis 2016].

[43] A. S. a. H. B. G. Portokalidis, "Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation.," in *EuroSys '06 Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006* , Belgium, 2006.

[44] Microsoft, "Microsoft.com," Microsoft, [Online]. Available: https://msdn.microsoft.com/en-us/library/d06h2x6e.aspx. [Accessed 12 Spalis 2016].

[45] A. Rubini and J. Corbet, Linux Device Drivers, 2nd Edition, Sebastopol, California: Oreilly & Associates Inc., 2001.

[46] Valentin Rothberg, "Interrupt Handling in Linux," Friedrich-Alexander-Universit Erlangen-Nurnberg, Dept. of Computer Science, Erlangen, 2015.

[47] http://ttyrpld.sourceforge.net/desc.php, "http://ttyrpld.sourceforge.net/desc.php," [Online]. Available: http://ttyrpld.sourceforge.net/desc.php. [Accessed 17 Sausis 2017].

[48] Sebek, "honeynet.org," [Online]. Available: http://www.honeynet.org/project/sebek. [Accessed 18 Lapkritis 2016].

[49] F. Bellard, "Qemu, a fast and portable dynamic translator," in *ATEC'05: Proceedings of the USENIX Annual Technical Conference 2005 on USENIX Annual Technical Conference*, Berkeley, 2005.

[50] C.-M. Karat, C. Halverson, D. Horn, and J. Karat., "Patterns of entry and correction in large vocabulary continuous speech recognition systems," in *CHI '99: Proceedings of the SIGCHI conference on Human factorsin computing systems*, New York, 1999.

[51] T. G. ®, "www.tripleg-3.com/," tripleg-3, 2007-2017 . [Online]. Available: http://www.tripleg-3.com/. [Accessed 12 Lapkritis 2016].

[52] Spyrix, "www.spyrix.com," spyrix, [Online]. Available: http://www.spyrix.com/spyrix-free-keylogger.php. [Accessed 7 Gruodis 2016].

[53] KidLogger, "http://kidlogger.net/," KidLogger, [Online]. Available: http://kidlogger.net/. [Accessed 28 Lapkritis 2016].

PRIEDAI

1 Priedas. Aukšto lygio funkcijos trigerio aprašymas IDS lygmenyje

```
handle_keyboard_syscalls(NR_sysalls){
switch (NR_syscalls)
case 253:
/*Passing: Tainting and Analyzing*/
/*Converting from a virtual to physical address*/
phyaddr = get_phy_page(env, env->regs[R_ECX]);
/*Receiving passed_data*/
. . .
if (!is_written)
/*Tainting the keystroke data*/
tag_set_keyboard(ECXTAG, phyaddr,. . .);
else/*Occurred writing operation*/
/*Analyzing tainted data */
if (map_istainted(phyaddr & PAGE_MASK)&. . .)
/* Data is tainted, checking the process ID*/
if (matchedPID())
notice(env, PID,. . .);
else
alert(env, ALERT_CI, PID,. . .);
/*Untainting*/
tag_clear(. . .);
else
/*Ignoring this data*/
case 254:/*Taint propagating*/
/*Receiving passed data*/
. . .
/*Checking tainted data*/
if (map_istainted(phyaddr & PAGE_MASK)& . . .)
if (m_id == memcpy)
/*Tainting the current data*/
else if (m_id == memove)
/*Tainting the new data, untainting the old one*/
else if (m_id == kfree)
/*Untainting the current data*/
else
/*Ignoring this data*/
default:
/* Notice error in passing */
}
```