KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ INŽINERIJA (P170B400)

Application Layer (L7) Firewall component

Naudojoto vadovas

Atliko ELITNET1:

Kazimieras Buškus, IFF-6/9 gr.

Šarūnas Andrijauskas, IFF-6/7 gr.

Robertas Strazdauskas, IFF-6/7 gr.

Tomas Jurevič, IFF-6/1 gr.

Priėmė:

lekt. Andrej Ušaniov

TURINYS

1. Įvadas 3

1.1. Papildoma informacija 3

2. Programos funkcijos ir diegimas 3

2.1. Pagrindinės funkcijos 3

2.2. Diegimas 3

2.3. Vartotojo sąsaja 5

# Įvadas

Šis programinės įrangos produktas yra viena dalių, turinčių funkcionuoti bendroje L7 ugniasienės platformoje, tačiau jį galima naudoti ir kaip atskirą programą. Tai nėra plačiajai vartotojų auditorijai skirta programinė įranga, todėl daroma prielaida, jog šio dokumento skaitytojas – sistemos administratorius, testuotojas ar kitos srities IT profesionalas.

## Papildoma informacija

Neatlikus papildomų, šiame dokumente neaprašytų pasirengimo darbų ir sistemų sinchronizacijos, analizatorių galima naudoti vienos (lokalios) tinklo sąsajos plokštės stebėjimui ir pranešimams apie galimą ataką gauti. Daroma prielaida, jog sistemoje (konkrečiai šiam dokumentui – Ubuntu 17.10) jau įdiegta *Python 3.6* versija ir jo modulių tvarkyklė *pip3*.

Programa konfigūruojama per *yaml* formato failus.

Kadangi tam tikri programos naudojami paketai funkcionuoja tik linux platformoje, dokumente bus pateikti įdiegimo šablonai būtent šio tipo operacinei sistemai (konkrečiai Ubuntu 17.10).

Naudojami sutrumpinimai:

* IP – internet protocol address
* ML – machine learning
* HRPI – HTTP GET requests per second
* GUI – graphical user interface

# Programos funkcijos ir diegimas

## Pagrindinės funkcijos

Programa atlieka veiksmus, leidžiančius:

* įspėti apie ataką paleidžiant nurodytą *skriptą*
* stebėti tinklo būseną grafinėje sąsajoje

Programa iš užklausų ir IP adresų skaičiaus, sužinomų išanalizavus į tinklo plokštę ateinančius paketus, pasisirstymo entropiją išgauna tam tikrus parametrus, kurie ML algoritmu kas 0.1 sekunės klasifikuoja tinklo būseną kaip ataką arba ne. Tinklo būsenos charakteristikos ir jų įvertinimas saugojami duomenų bazėje, istorinį ir realaus laiko grafikus galima matyti grafinėje vartotojo sąsajoje (naršyklėse Chrome (version 66) bei Mozilla naršyklės (version 60))

## Diegimas

Pasirengimui programinės įrangos vartojimui etapai yra šie:

* programinio kodo atsisiuntimas
* *dependencies* instaliavimas
* programos paleidimas

Programą galima atsisiųsti iš repozitorijos adresu: https://github.com/sarand2/ELITNET1.

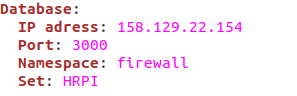
Atsisiuntus programinį kodą į norimą aplanką, reikia instaliuoti jai reikalingas bibliotekas ir modulius. Ubuntu platformoje tą galima padaryti per pip modulių tvarkyklę. Visi reikalavimai programai surašyti faile *requirements.txt*. Pavyzdys:



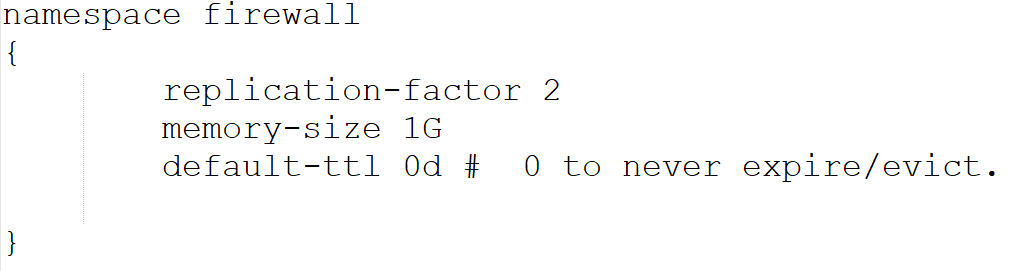
Pav 1. Paketų diegimas

Jeigu visi paketai instaliuojasi sėkmingai, galima tęsti diegimą, jei ne – programa gali funkciuonuoti neteisingai arba iš viso neveikti.

Kadangi analizatorius naudoja *Aerospike* duomenų bazę charakteristikų saugojimui, reikia lokaliai įsidiegti arba naudoti turimą duomenų bazės serverį. Informaciją apie instaliaciją Ubuntu galite rasti čia: <https://www.aerospike.com/docs/operations/install/linux/ubuntu>

Jeigu serveris veikia kitame kompiuteryje, programos *configuration* aplanko *scriptConfig.yaml* faile galima nurodyti prisijungimo prie duomenų bazės informaciją:

Pav 2. Duomenų bazės konfigūracija



Pav . Aerospike konfigūracija

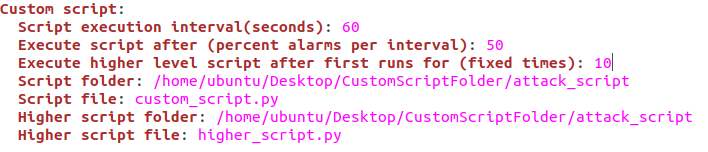
Analizatorius naudoja įvardintą *Aerospike* duomenų bazės *namespace* su tam tikra konfigūracija. Aukščiau parodyta informacija yra iš */etc/aerospike/aerospike.conf* aplanko aplinkoje, naudotoje programos testavimui. Čia nurodyta bazinė *namespace* informacija, ją galima keisti/papildyti pritaikant atitinkamai platformai.

Programinis kodas tinkle būsenai nustatyti *“sniffina”* paketus naudodamas *scapy* biblioteką, kuri įdiegiama kartu su kitais reikalingais paketais iš *requirements.txt* failo. Kad analizatorius veiktų teisingai, reikia nurodyti virtualią arba fizinę sąsają, iš kur bus imami paketai. Tai galima padaryti *scriptConfig.yaml* faile, nurodant atitinkamos sąsajos pavadinimą:

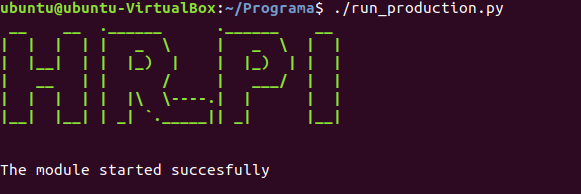


Pav 4. Sąsajos konfigūracija

Analizatorius suteikia galimybę paleisti pasirinktą *skriptą,* kai tinklo būsena tam tikrą laiką signalizuoja ataką. Nustatymus, kada tiksliai vykdyti, kur *skriptai* yra kompiuteryje galite rasti *scriptConfig.yaml.*



Pav 5. Skripto konfigūracija

Programos paleidimas vykdomas iš katalogo, kur yra programinis kodas, arba nurodant absoliutų kelią. Pavyzdys:

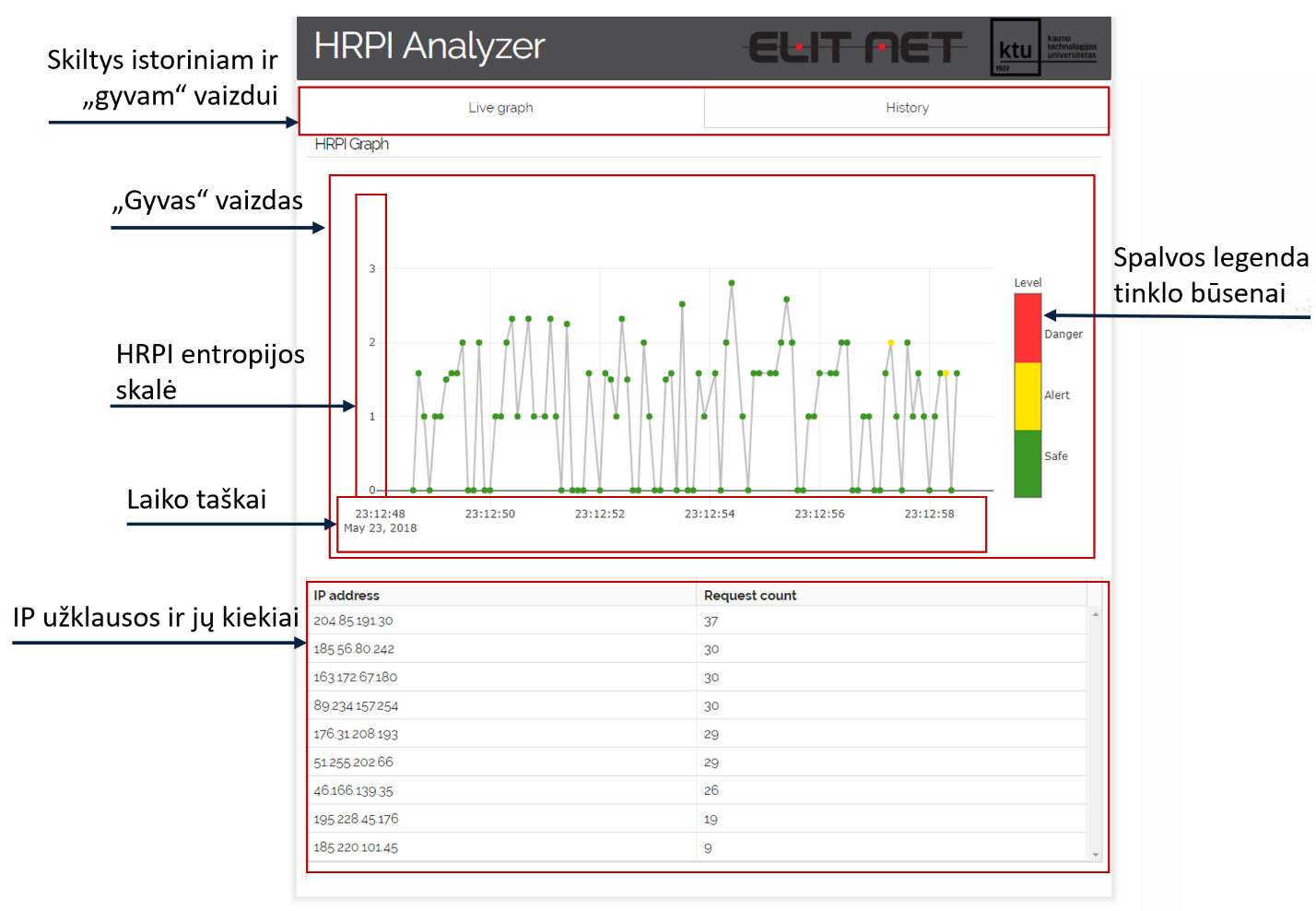
Pav 6. Programos paleidimas

Informacija apie programos veikimą galima stebėti *info.log* bei *error.log* failuose, esančiuose programos pagrindinio scripto *run\_production.py* aplanke.

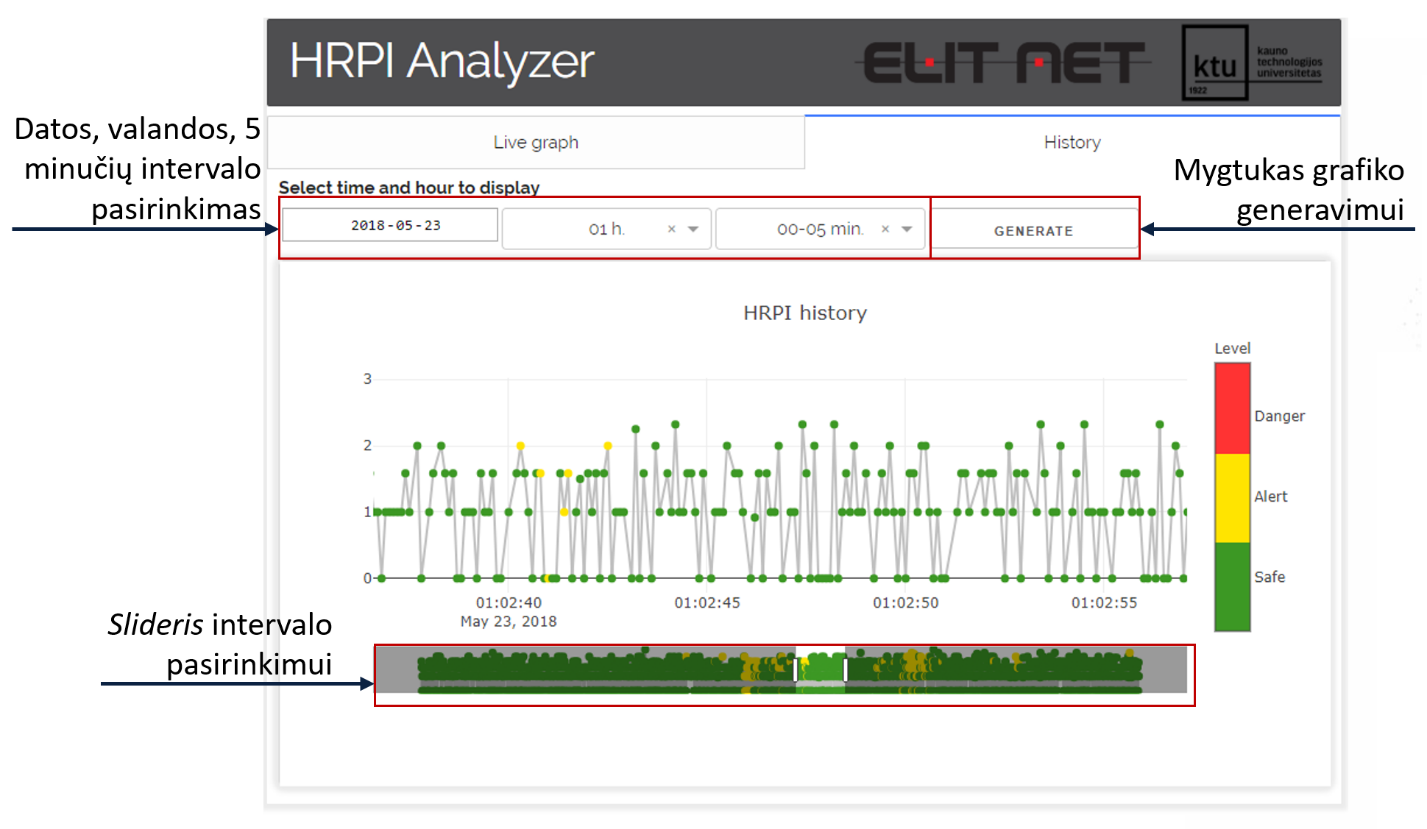
Vartotojo sąsaja įdiegimo nereikalauja, programa paleidžia GUI *web app* ant *localhost:3000*, ji gali būti pasiekiama per internetą atlikus *port forward*.

## Vartotojo sąsaja

Šio produkto vartotojo sąsaja yra tik papildomas programinės įrangos kūrimo proceso produktas, ji skirta patogiau stebėti, ar programa teisingai funkcionuoja realiu laiku. Vartotojo sąsają sudaro 2 sekcijos: realaus laiko grafikas, rodantis dabartines tinklo *traffic* charakteristikas, ir istorijos sekcija, rodanti pasirinkto intervalo informaciją.



Pav 7. Programos GUI



Pav 8. Programos GUI (2)