KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ INŽINERIJA (P170B400)

Application Layer (L7) Firewall component

Projekto apibendrinimas

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. Bendra informacija 3

2. Trečias sprintas 3

2.1. Pagrindinės užduotys 3

3. Testavimas 3

3.1. Naudojama programinė įranga 3

3.2. Testavimo planas bei rezultatai 4

4. Išvados 5

# Bendra informacija

* Tvarkaraštis:

Semestro projektas: antradienis 15:30

Programų sistemų inžinerija: nelyg. sav. trečiadienis 11:00

* Programos *wiki:* https://github.com/sarand2/ELITNET1/wiki/
* Programos kūrimo backlog‘as:

**Email**: gaming\_fun@gmail.com

**Password**: labasrytas

* Komandos narių pareigos pagal atliekamus darbus:

Robertas Strazdauskas: GUI kūrimas ir duomenų bazė

Kazimieras Buškus: HRPI logika ir multiproceso įgyvendinimas

Šarūnas Andrijauskas: DDoS atakų generavimas ir multiproceso įgyvendinimas

Tomas Jurevič: paketų iš tinklo plokštės rinkimas

* Programos logika remiasi šiuo moksliniu straipsniu. [1]. Tongguang Ni, Xiaoqing Gu, Hongyuan Wang, and Yu Li, “Real-Time Detection of Application-Layer DDoS Attack Using Time Series Analysis,” Journal of Control Science and Engineering, vol. 2013, Article ID 821315, 6 pages, 2013. doi:10.1155/2013/821315

# Išvados įvykdžius projektą

Bendradarbiaujančios įmonės „Elitnet“ pasiūlytas ugniasienės komponento kūrimas iš pat pradžių pasirodė nelengva užduotis, o kūrimo stadijoje ši tendencija tik ryškėjo. Nepaisant to, metodas, aprašytas moksliniame straipsnyje, buvo įgyvendintas, pasiekti panašūs rezultatai.

Priklausomai nuo to, ką kūrė, komandos nariai susipažino su statistinės analizės metodais (*Kalman filter*), *machine learning* algoritmo *Support vector machine* specifika, nereliacine duomenų baze *Aerospike,* taip pat *dashboard* kūrimo karkasu *plot.ly*. Projekte buvo naudojama išskirtinai *Python* programavimo kalba dėl jos lankstumo bei galimybės patogiai dirbti su *machine* *learning* moduliais *linux* aplinkoje, todėl kiekvienas komandos narys stipriai patobulino šios kalbos žinias, geriau susipažino su Ubuntu OS. Bendra išvada tokia: kuriant matematiniais modeliais grįstas programas daugiausia laiko sugaištama bandant teisingai inplementuoti aprašytus algoritmus. Taip atsitinka todėl, jog mokslininkai pateikia tik pseudokodus ir aprašymus, kurie reikalauja gilesnių matematikos ir statistikos žinių. Visi kiti reikalavimai buvo įgyvendinti įsisavinant karkaso, duomenų bazės projektavimo ir panaudojimo specifiką, tai buvo labiau esančių metodų pritaikymo programos specifikai uždavinys.

Programos inžinerijos procesų prasme, projektas ėjosi sklandžiai dėl pakankamai gerai suplanuotų sprintų bei darbo pasiskirstymo. Agile metodologija leido daryti kompromisus tarp iteracijų, nespėjant atlikti ar papildant užduotis bei reikalavimus projekto vykdymo metu. Dėl to, jog komandos nariai iš karto tiksliai identifikavo savo stipriąsias puses ir darbų pasiskirstymas nesukėlė didelių problemų, dauguma iteracijų praėjo sklandžiai, dauguma reikalavimų buvo įvykdyti.

Tai, jog įmonės programuotojai perėmė kai kuriuos sudarytus algoritmus, bei tai, kad tinklo būsenos (DDoS ir normalaus traffic) klasifikavimo teisingas pataikymų skaičius nepriklausomuose eksperimentuose buvo virš 90 %, laikome projekta sėkmingu.