KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Saulius Grušnys

DDoS atakų tyrimas paketiniuose tinkluose

Magistro darbas

Darbo vadovas
lekt. dr. I. Lagzdinytė

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Saulius Grušnys

DDoS atakų tyrimas paketiniuose tinkluose

Magistro darbas

Recenzentas
doc. dr. T. Adomkus
2010-05-26

Vadovas
lekt. dr. I.Lagzdinytė
2010-05-26

Atliko
IFN 8/3 gr. stud.
Saulius Grušnys
2010-05-26

Kaunas, 2010

Turinys

Summary

Defending against Distributed Denial of Service (DDoS) attacks is one of the most important tasks to ensure service availability. At the same time it is one of the most challenging tasks because it requires complex and efficient methods to correctly identify and stop such kind of attacks. There are number of methods available to identify DDoS attacks. Some of the methods are based on single packet or connection; others evaluate packets according to all the traffic available at particular time. There is a need to identify what method or methods should be used under particular circumstances. In this work a software system is developed, which implements some of the available methods to detect DDoS attacks and creates firewall rules to stop the traffic from the hosts suspected to be participating in the attack. Implemented methods include Change Point Approach, Covariance model and Passive Measurement based Heuristics. The system enables to analyze characteristics of implemented DDoS identification methods and evaluate their efficiency in different conditions, distinguish legitimate and attacking traffic and block traffic from attacking packets.

Įvadas

Atsisakymo aptarnauti (DoS) atakos – tai tokia atakų rūšis, kuomet sutrikdomas paslaugos tiekimas taip, kad ja nebegalėtų pasinaudoti vartotojai, kuriems paslauga yra teikiama. Paskirstyta DoS ataka (toliau DDoS) – tai tokia ataka, kuri įvykdoma užtvindant auką dideliu kiekiu paketų, ateinančių iš didelio kiekio skirtingų atakos šaltinių [1]. Dažniausiai šiam tikslui pasiekti naudojami trojos arkliai, kitokia piktybine programine įranga užkrėsti nieko neįtariančių vartotojų kompiuteriai, įvairios nulaužtos tarnybinės stotys. DDoS ataka taip pat gali būti vykdoma siunčiant daugiau tarnybinės stoties resursų reikalaujančias užklausas, panaudojant mažesnį skaičių atakos šaltinių.

DDoS atakos gali būti dviejų tipų: 1) kai sudaromi nepilni susijungimai, arba siunčiami klaidingi paketai, siekiant pasinaudojant pažeidžiamumais išnaudoti atakuojamo serverio resursus; 2) kai sudaromi teisingi susijungimai, o rezultatas pasiekiamas panaudojant tokį užklausų kiekį, kurio atakuojama sistema nesugeba apdoroti.

Pirmojo tipo atakų metu atakuotojams reikia mažiau atakos šaltinių, tačiau įvairios užtvindymo atakos yra lengviau aptinkamos.

Jei DDoS atakos metu siunčiamos teisingos užklausos, įvairios taikomojo lygio užkardos ar įsilaužimo aptikimo, ar kitos sistemos, ieškančios žinomų atakų šablonų analizuojamame sraute, tokių užklausų nelaiko atakomis. Jei atakoje dalyvaujančių mazgų skaičius pakankamai didelis, visi serverio resursai bus skirti apdoroti atakuojančių mazgų užklausoms, o teisėtų vartotojų užklausos gali būti arba išvis neapdorojamos, arba apdorojamos per nepriimtinai ilgą laiką. Tokias atakas aptikti yra sudėtinga.

DDoS atakos taip pat gali būti klasifikuojamos pagal paketų siuntimo pobūdį: 1) tiesioginės atakos; 2) atspindžio atakos [2].

Tiesioginių atakų atveju visi paketai siunčiami iš atakuojančių mazgų tiesiogiai aukai. Daugeliu atveju naudojami suklastoti IP adresai, todėl ataka tampa dar efektyvesnė, nes auka pakartotinai siunčia paketus su atsakymais į užklausas neegzistuojantiems adresams.

Naudojant atspindžio atakas siunčiami paketai įvairiems tinklo mazgams, nurodant aukos adresą kaip šaltinio adresą. Auka užtvindoma atsakymais, ateinančiais iš serverių, gavusių suklastotas užklausas. Tokias atakas labai sunku stabdyti, nes srautas ateina iš teisėtų adresų, kurie gali būti tiesiogiai reikalingi paslaugoms aukos serveryje teikti.

Norint užkirsti DDoS atakai kelią nuo pat jos pradžios, atakos aptikimas turi būti kaip įmanoma greitesnis ir efektyvesnis [3]. DDoS atakos aptikimas taip pat yra labai komplikuotas, nes labai panašus srautas gali būti sugeneruotas ir padidėjus teisėtų vartotojų skaičiui. Kita DDoS atakų aptikimo problema yra ta, kad atsiunčiamo srauto apimtis gali būti tokia didelė,kad aukos įranga

išvedama iš rikiuotės vien bandydama apdoroti įeinančius paketus. Dėl šios priežasties atakos aptikimo metodai turi būti labai efektyvūs ir atlikti kuo mažiau papildomų skaičiavimų.

Kovojant su DDoS atakomis kyla sekančios problemos:

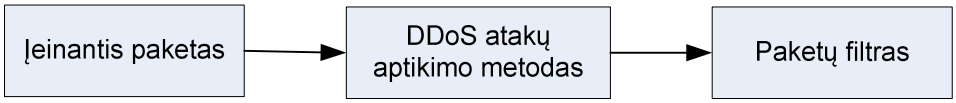
- DDoS atakos aptikimas - uždavinys, sprendžiantis kaip stebint visą srautą atpažinti, kad yra vykdoma DDoS ataka. Kaip jau buvo minėta, atskirai paėmus kiekvieną užklausą ji gali būti teisinga ir dauguma tradicinių sistemų tokių atakų neaptinka.
- DDoS atakos sustabdymas - tiek paslaugas tiekiančios tarnybinės stotys, tiek interneto paslaugų tiekiejų naudojami maršrutizatoriai bei ugniasienės, turi ribotus atminties ir skaičiavimo resursus, todėl netgi aptikus DDoS atakas, gali nepavykti užkirsti joms kelio arba ataka gali pavykti vien dėl šios įrangos apkrovimo. Kuomet maršrutizatorius visą laiką persiuntinėja DDoS atakoje dalyvaujančius paketus, kai kurie paketai dėl ribotų resursų gali būti tiesiog ignoruojami. Tarp ignuoruoajmų paketų gali patekti ir teisėtų srautų paketai. Taip pat ir ugniasienės neturėdamos pakankamai resursų gali pradėti atmetinėti visus įeinančius paketus, taip užkirsdamos kelią teisėtiems paketams pasiekti paslaugos tarnybinę stotį.

Šiame darbe nagrinėjami jau egzistuojantys DDoS atakų aptikimo ir sustabdymo metodai, praktiškai tiriamos jų charakteristikos. Taip pat nagrinėjami keli ugniasienių optimizavimo metodai, kurie vėliau pritaikyti praktinėje realizacijoje. Pagrindinis darbo tikslas – sukurti sistemą, kuri efektyviai atpažintų DDoS atakas analizuojamame paketų sraute ir jas sustabdytų.

Darbo tema publikuotas straipsnis konferencijoje „IT2010“.

1 DDoS atakų ir jų aptikimo metodų analizė

DDoS atakas aptinkančios ir stabančios sistemos pagrindiniai veiksmai pateikiami 1 paveiksle:



1 Pav. Pagrindiniai DDoS atakų aptikimo ir sustabdymo sistemos veiksmai

DDoS atakos aptikimyas vyksta įeinantį paketą patikrinant tam tikru atakos aptikimo metodu. Pagal atakos aptikimo metodo rezultatus priimamas sprendimas, ar prasidėjo DDoS ataka, ar ne. Jei nustatoma, kad vyksta ataka, paketas turi būti blokuojamas. Jei nustatoma, kad paketą reikia blokuoti, turi būti sukuriama ir įterpiama taisyklė į sistemos paketų filtro taisyklių sąrašą.

Jei taisyklės į paketų filtrą dedamos be jokios tvarkos, gali nutikti taip, kad įeinantys paketai pirmiausia bus tikrinami pagal tas taisykles, kurios nėra svarbios atakoje dalyvaujančių srautų blokavimui ir bus gaištamas laikas. Todėl naujų taisyklių kūrimas turi būti vykdomas taip, kad dažniausiai pasitaikančius paketus apdorojančios taisyklės būtų taisyklių sąrašo priekyje.

Kituose skyriuose bus nagrinėjami jau sukurti DDoS atakų aptikimo metodai, paketų blokavimo metodai, taisyklių išdėstymo atmintyje būdai. Taip pat bus analizuojamas paketų filtras, kuris bus naudojamas sistemos realizacijoj.

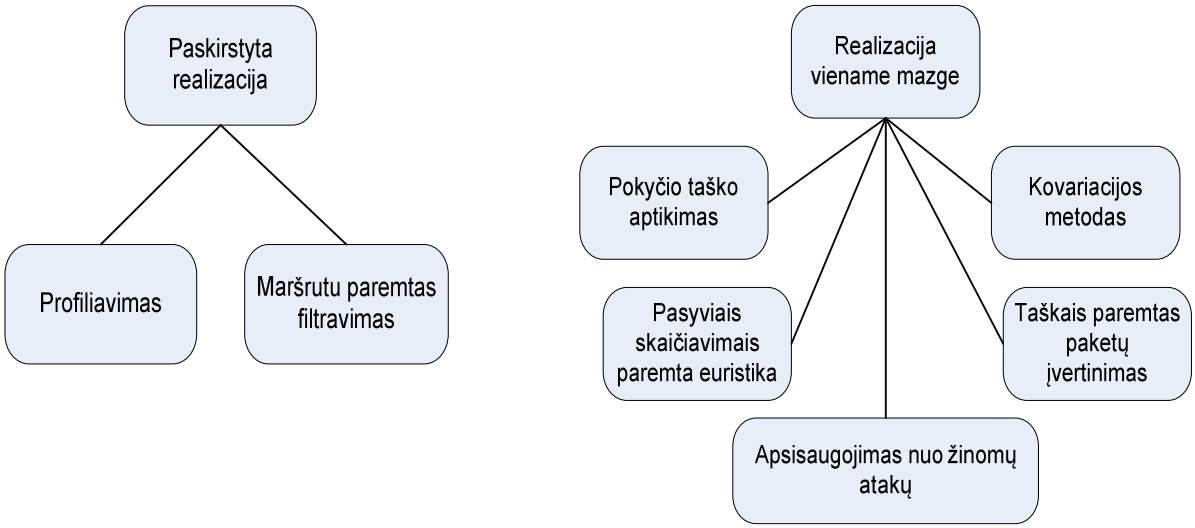
1.1 Egzistuojantys DDoS atakų aptikimo metodai

Yra sukurta keletas metodų, siekiančių aptikti DDoS atakas. Sukurti DDoS atakų aptikimo metodai gali būti klasifikuojami pagal tai, kaip jie išdėstomi tinklo infrastruktūroje. 1 pav. pateikta egzistuojančių DDoS atakų aptikimo metodų klasifikacija.

Pirmosios grupės metodai (2 pav. (a)) siūlo priemones DDoS atakų sustabdymui globaliu mastu [2,4]. Šių metodų idėja yra tokia, kad yra tam tikras skaičius sistemų, aptinkančių DDoS atakas. Šios sistemos išdėstytos daugelyje skirtingų vietų tiekėjo tinkle arba visame internete. Kai įtariama, kad prasidėjo ataka, sistemos susisiekia tarpusavyje ir apsikeičia informacija, ar kitos sistemos sraute pastebėjo ką nors įtartino. Jei ataka patvirtinama keliose srautą stebinčiose sistemose, registruojama ataka, o maršrutizatoriams nurodoma blokuoti paketus, siunčiamus iš atakuojančių mazgų [4].

Pagrindinis tokių metodų privalumas yra tas, kad sistemos gali gauti daugiau informacijos apie aplinką, komunikuodamos viena su kita. Kitas privalumas yra tas, kad srautas, pereinantis per kiekvieną sistemą yra mažesnis. Tokių metodų trūkumas yra sudėtingas jų įgyvendinimas, nes srauto stebėjimas daugelyje skirtingų vietų reikalauja didelių lėšų.

Kita metodų grupė (Pav. 1 (a)) skirta DDoS atakų aptikimui atakuojamame mazge [1, 3, 5-9]. Daugumos šių metodų veikimo principas paremtas srauto stebėjimo tam tikrą laiko tarpą, kuomet srauto apkrova yra įprasta ir tiksliai žinoma, kad nėra vykdomos jokios atakos. Stebint įvairius srauto parametrus sukuriami profiliai, atitinkantys normalią apkrovą.



(a) (b)

2 pav. DdoS atakų aptikimo metodų klasifikacija. (a) Paskirstyti metodai, (b) Viename mazge veikiantys metodai

Kuomet prasideda ataka, atsiranda skirtumai tarp stebimų srauto parametrų reikšmių ir reikšmių, įrašytų srauto profiliuose. Tokiu būdu galima identifikuoti prasidėjusią ataką.

Pagrindinis tokių metodų privalumas yra tai, kad jie veikia viename mazge. Kitas privalumas yra tas, kad slenkstinės metoduose naudojamos reikšmės gali būti koreguojamos atsižvelgiant į srauto, pasiekiančio tą mazgą, charakteristikas. Pagrindinė tokių metodų silpnybė yra ta, kad viename mazge veikianti sistem gali būti lengviau užtvindoma dideliu paketų srautu.

1.1.1 Bendri DDos atakų aptikimo metodų požymiai

Daugumos jau egzistuojančių DDoS atakų aptikimo algoritmų veikimas pagrįstas statistine paketų analize. Tam tikrą laiką stebimi paketai ir sudaromi įvairiais statistiniais parametrais paremti srauto šablonai. Jei nagrinėjamas srautas neatitinka sudarytų šablonų – įtariama, kad pradėta DDoS ataka. Pagrindiniai skirtumai tarp nagrinėtų algoritmų yra:

1. kaupiamų duomenų saugojimo būdai ir stebimi skirtingi srauto parametrai;
2. arba srauto šablonai sudaromi pagal skirtingas matematines formules.

Algoritmų efektyvumas vertinamas pagal du parametrus:

1) Kokia dalis vykdomų DDoS atakų atpažįstama;
2) Kiek pasitaiko netikro pavojaus atvejų taikant konkretų algoritmą.

Ne visuose literatūros šaltiniuose pateikiami siūlomų metodų tyrimai, todėl šioje darbo dalyje į šiuos parametrus nebus atsižvelgta. Prireikus, šie parametrai bus nustatyti tyrimų metu.

1.1.2 Profiliavimas

Šis metodas buvo kuriamas DDoS atakų aptikimui didelių interneto paslaugų tiekėjų (IPT) tinkluose. Stebimas srautas visuose IPT priklausančiuose maršrutizatoriuose ir sudaromi srauto profiliai. Per tam tikrą laiko tarpą sudaromi normalaus srauto šablonai. Kuomet keliaujantis per tam tikrą maršrutizatorių srautas neatitinka sudarytų šablonų, maršrutizatorius tampa "įtarus". Šio metodo pagrindinės problemos yra šios:

- Kaip išvengti netikrų pavojų atvejų, kuomet maršrutizatoriams nematytas intensyvus teisėtas srautas gali būti pripažintas kaip DDoS ataka.
- Ar tokia realizacija nesukels galimybių vykdyti naujas atakas?

Siūloma, kad maršrutizatoriai komunikuotų tarpusavyje apie įtartinus srautus. Tuomet tik didesnio masto atakos, kurių srautas keliauja per IPT tinklą būtų aptiktos. Tai sumažintų netikro pavojaus atvejų skaičių.

Profiliavimas atliekamas maršrutizatoriuose nustatant tam tikrą konstantą Θ. Ši konstanta yra dalis visos maršrutizatoriaus atmintes. Jei išeinantiems susijungimams į tam tikrą mazgą skirtos atminties dalis pasiekia Θ, srautas į tą mazgą pradedamas profiliuoti. Toks mazgas laikomas populiariu. Stebimi parametrai, naudojami srauto profiliavimui:

- Bendras baitų, siunčiamų stebimam mazgui, kiekis.
- Potinklių, iš kurių siunčiamas srautas stebimam mazgui, pasiskirstymas.
- Paketų, siunčiamų stebimam mazgui, dydžių pasiskirstymo stebėjimas.

Šio metodo privalumas yra tas, kad dirbama su mažesne dalimi srauto ir sprendimą apie ataką galima priimti remiantis IPT tinklo srauto kontekstu.

Šio metodo trūkumas yra tas, kad jeigu DDoS atakos srautas pereina per IPT maršrutizatorius daug kartų, jis gali įgauti teisėto srauto profilį.

1.1.3 Pokyčio taško aptikimas

Stebimas paketų pasiskirstymas pagal tam tikrus parametrus (pvz. paketų vidurkio, naujų susijungimų kiekio, paketų iš naujų adresų kiekio) per tam tikrą laiką. Medžiui formuoti naudojama iš maršrutizatorių gaunama informacija. Imamas stebimam mazgui skirtų paketų skaičiaus vidurkis tam tikrais laiko intervalais. Skaičiuojamas paketų vidurkis per visą stebėjimo laiką, vidurkį per tam tikrą intervalą dauginant iš to intervalo eilės numerio. Tokiu būdu sudaromas vidurkis, daugiau priklausantis nuo naujesnio srauto. Prasidėjus DDoS atakai, atsiranda paketų vidurkio nuokrypiai

nuo apskaičiuoto vidurkio.

Šio metodo trūkumas yra toks, kad nustačius per ilgą intervalą gali išaugti atakos užregistravimo laikas, nes ataka gali būti paskelbta tik praėjus keliems nustatytos trukmės laiko intervalamas.

Pagrindinis šio metodo privalumas yra toks, kad atakos aptikimas vykdomas prieš ataką buvusio srauto kontekste. Taip pat naudojant šį metodą galima tiksliai nustatyti atakos pradžios laiką.

1.1.4 Kovariacijos analizės metodas

Išrenkami stebimi srauto parametrai. Pagrindiniai stebimi parametrai:

- TCP SYN (susijungimo užmezgimo) paketų skaičius per tam tikrą laiko intervalą.
- Analizuojamo paketo šaltinio IP adresas.
- Analizuojamo paketo TTL (time to live – mazgų skaičius, per kuriuos gali pereit paketas, kol pasiekia adresatą) reikšmė.
- Apskaičiuojamas RTT (vidutinis paketo kelionės laikas) iki siuntėjo. Šio parametro stebėjimas gali paspartinti aptikimą DDoS atakų, komet naudojami suklastoti siuntėjo IP adresai.

Parametrai registruojami ir skaičiuojama jų koreliacija ir kovariacija. Normalių srautų parametrų koreliacija skirsis nuo tos srautų koreliacijos, kuomet vykdoma DDoS ataka.

Kovariacija skaičiuojama pagal išraišką:

$$cov\,xy = \frac{1}{N}\sum x_i y_i - \overline{xy}\ , \qquad (1)$$

kur: N – matavimų kiekis; x_i - pirmojo kintamojo, i-tajame matavime reikšmė; y_i - antrojo kintamojo i-tajame matavime reikšmė, i kinta nuo 1 iki N;

Šis metodas efektyvus tada, kai reikia aptikti TCP SYN užtvindymo atakas, nes SYN ir FIN vėliavėles turinčių paketų skaičius smarkiai pakinta ir praranda priklausomybę vienas nuo kito. Šį metodą taip pat galima pritaikyti aptikti kitokio tipo atakas, atrenkant skirtingas srauto parametrų poras ir identifikuojant kovariacijos pasikeitimus tarp tų reikšmių vykstant konkrečiai atakai.

Šio metodo privalumas yra toks, kad atakos aptikimas gali būti skelbiamas iš karto, kuomet nustatytas paketų, turinčių stebimus parametrus kiekis analizuojamame sraute. Taip pat šį metodą galima pritaikyti aptikti įvairių tipų atakas.

Šio metodo trūkumas yra toks, kad jo rezultatai labiau priklauso nuo nominalių srauto parametro reikšmių.

1.1.5 Apsisaugojimas nuo žinomų DDoS atakų

Vykstant žinomai DDoS atakai, siunčiamos jau žinomos valdymo komandos į apkrėstus kompiuterius. Aptikus tokias komandas siunčiamuose paketuose galima arba užblokuoti siunčiamų komandų srautą į aukų kompiuterius, arba iškart užblokuoti įeinantį srautą iš aukų kompiuterių.

Naudojant šį metodą sistemoje yra galimybė teikti paslaugas net ir vykstant atakai prieš tam tikrus servisus, tačiau jis yra sunkiau plečiamas ir neapsaugo nuo naujų atakų.

1.1.6 Pasyviais skaičiavimais paremta euristika

Šis DDoS atakų aptikimo metodas remiasi įeinančio ir išeinančio srauto į konkretų mazgą santykio įvertinimu. TCP srauto atveju skaičiuojamas santykis tarp įeinančių SYN (susijungimo užmezgimo) paketų per sekundę ir išeinančių FIN paketų per sekundę. Šiuo atveju stebimas TCP srauto simetriškumas. Tvindant mazgą SYN paketais, siunčiamas tik vienas SYN paketas. Tokiu atveju mazgas išsiunčia SYN/ACK ir kadangi negauna daugiau jokių paketų tam tikrą laika savo resursu naudoja pusiau atidarytai sesijai laikyti. Tokios atakos atveju santykis tarp įeinančių SYN ir išeinančių FIN paketų smarkiai išaugtų ir tai leistų pakankamai efektyviai identifikuoti ataką.

UDP ir ICMP protokolų atveju sunkiau nustatyti simetriškumą, todėl tam tikrą laiką reikėtų pasyviai stebėti srautą ir suskaičiuoti santykius tarp įeinančių ir išeinančių srautų. Jei vykdoma užtvindymo (flooding) ataka, įeinantis srautas turėtų smarkiai padidėti, taip sukeldamas santykio tarp įeinančių ir išeinančių srautų nuokrypį nuo nustatyto santykio per tam tikrą laiką.

Pagrindinis šio metodo trūkumas yra tas, kad kai kuriais atvejais DDoS ataka vykdoma sudarant susijungimus laikantis protokolų nustatytų taisyklių, o atakos rezultatas pasiekiamas panaudojant didelį skaičių besikreipiančių mazgų. Tokių srautų šiuo metodu gali nepavykti tinkamai identifikuoti.

Šio metodo privalumas yra tas, kad atliekant pradinį parametrų nustatymą, neapkraunama tinklo įranga, nes nereikia vykdyti paketų analizės realiu laiku. Paketai gali būti surenkami į failus ir analizuojami tuomet, kai įranga yra mažiau apkrauta.

1.1.7 Taškais paremtas paketų įvertinimas

Šio metodo esmė – kiekvienam paketui skiriamas taškų skaičius atsižvelgiant į tam tikrus parametrus. Pagal surinktą taškų skaičių, sprendžiama ar persiųsti paketą toliau, ar jį atmesti. Šis metodas labai panašus į jau paplitusį metodą nepageidaujamų internetinių laiškų (*angl.* SPAM) atpažinimui.

Šis metodas naudoja Bejeso (*angl.* Bayesian) sąlygines tikimybes nustatyti, ar įtartinas paketas priklauso teisėtam srautui. Taškų skaičiavimui naudojami trys etapai:

1) Atakos aptikimas ir aukos identifikavimas atliekant keturių srauto parametrų statistinę analizę (paketai per sekundę, bitai per sekundę, aktyvių susijungimų

skaičius, naujai įvykstančių susijungimų dažnis). Šių parametrų reikšmės lyginamos su nustatytomis nominaliomis reikšmėmis. DDoS atakų aptikimą atliekantys serveriai surenka informaciją iš maršrutizatorių, siekdami nustatyti, ar yra vykdoma ataka.

2) Suteikiamas taškų skaičius kiekvienam paketui, kuris yra skirtas aukai. Taškai skiriami sudarant srauto profilį ir palyginant jį su nominaliu profiliu. Profiliai apskaičiuojami taikant sąlygines tikimybes ir įrašomi į taškų sąrašus. Skaičiuojant šiuo metodu, teisėtiems paketams bus priskiriamas didesnis taškų skaičius dėl jų (didesnė tikimybė, kad įtartinas paketas yra teisėtas). Tokiu būdu atskiriamas teisėtas srautas nuo neteisėto, atakos vykdymo metu.
3) Atmetami tie paketai, kurie viršyja dinamiškai nustatomą slenkstinę reikšmę. Slenkstinė reikšmė nustatoma atsižvelgiant į taškų skaičiaus pasiskirstymą paketuose ir į paketų, skirtų aukai kiekį.

Nominalios reikšmės surenkamos ir nustatomos tuo metu, kai laikoma, kad tinkle nėra vykdomos DDoS atakos.

Šis metodas gali veikti patikimai gerai sureguliavus ir išanalizavus srauto parametrus stebimoje sistemoje. Tokiu būdu pagal sudarytus profilius būtų tiksliai skiriami taškai paketams ir vyktų efektyvus paketų atmetimas, tačiau šio metodo realizacija yra sudėtinga ir reikalauja daug papildomų žingsnių apdorojant paketus.

1.1.8 DDoS atakų aptikimo metodų palyginimas

Taikant jau egzistuojančius atakų aptikimo metodus reikia nustatyti parametrus, pagal kuriuos būtų galima tuos metodus įvertinti. Labai svarbu, kad taikomas metodas veiktų greitai ir būtų lengvai pritaikomas prie konkrečios situacijos. 1 lentelėje, ~~kurioje aukščiau nagrinėti~~ atakų aptikimo metodai palyginti pagal galimybę dirbti su įvairiais srauto parametrais, galimybę koreguoti metodo reikšmes pagal konkrečias srauto sąlygas, galimybę realizacijai viename mazge arba paskirstytu būdu.

1 lentelė. DDoS atakų aptikimo metodų palyginimas

Pavadinimas	Įvairių srauto parametrų palaikymas	Reikšmių koregavimas atsižvelgiant į sąlygas	Realizacija galiniame taške	Paskirstyta realizacija
Profiliavimas	+	+	-	+
Pokyčio taško aptikimas	-	+	+	-
Apsisaugojimas nuo žinomų atakų	-	-	+	+
Kovariacijos metods	+	+	+	-

Pasyviais skaičiavimais paremta euristika	+	+	+	-
Taškais paremtas paketų įvertinimas	-	+	+	-

Pagal lentelėje pateiktus duomenis matome, kad iš viename mazge realizuojamų metodų geriausias charakteristikas turi pokyčio taško aptikimo, kovariacijos, pasyviais skaičiavimais paremtos euristikos ir taškais paremto paketų įvertinimo metodai. Renkantis metodą realizacijai dar bus atsižvelgiama į reikalginų operacijų skaičių paketo įvertinimui.

1.2 DDoS atakų nutraukimo metodai

Stabdant DDoS atakas dažniausiai naudojamas atakoje dalyvaujančių paketų atmetimas. DDoS atakas aptinkančios sistemos identifikuoja srautą kaip DDoS ataką ir perduoda informaciją ugniasienėms ar maršrutizatoriams. Siuntėjo srautas gali būti blokuojamas:

- Interneto paslaugų tiekėjo maršrutizatoriuje, esančiame arčiausiai siuntėjo.
- Interneto paslaugų tiekėjo maršrutizatoriuje, esančiame arčiausiai aukos.
- Atakuojamoje tarnybinėje stotyje įdiegtoje ugniasienėje.

Paketų atmetimas IPT maršrutizatoriuose, esančiuose arčiausiai atakuojančio adreso yra efektyvesnis tuomet, kai DDoS ataka vykdoma siunčiant labai didelius duomenų paketų kiekius. Taip yra todėl, kad srautas auką pasiekia iš įvairių interneto taškų, todėl pasiskirsto per daugelį IPT maršrutizatorių. Kiekvienas maršrutizatorius atskirai turi apdoroti mažesnį paketų srautą. Jei visas srautas pasiektų auką, visi serverio resursai gali būti išnaudojami atmetinėjant atakuojančius paketus ir DDoS ataka būtų įgyvendinta. Jei DDoS ataka vykdoma pateikiant sąlyginai nedidelį kiekį, daug serverio resursų reikalaujančių užklausų, nuo jos gali apsisaugoti ir auka, pati atmesdama kaip DDoS ataką identifikuotus srautus.

Nurodymus atmesti vienus ar kitus paketus, maršrutizatoriai gali gauti iš anksčiau nagrinėtais metodais veikiančių DDoS atakų aptikimo sistemų.

1.2.1 Paketų atmetimas atkuriant kelią

Taikant šį metodą, interneto paslaugų tiekėjo įranga turi palaikyti paketų žymėjimą. Paketų žymėjimas atliekamas siunčiant kontrolinį paketą iki atsitiktinai pasirinkto siuntėjo, kurio srautas keliauja per tą maršrutizatorių ir kiekvienam maršrutizatoriui pridedant žymą apie save į tą paketą. Paketų žymėjimas gali būti atliekamas ir pareikalavus aukai. Jei auka nustato, kad tam tikri adresai dalyvauja vykdant DDoS ataką ir jei yra gauti žymėti paketai su keliais iki tų adresų, siunčiamos komandos maršrutizatoriams nurodant pilną kelią, kurį reikia užblokuoti. Tokiu būdu kiekvienas maršrutizatorius persiunčia šią užklausą kitiems maršrutizatoriams, kurie įtraukti į žymėtą paketą.

Tokiu būdu blokavimas atliekamas arčiausiai atakas atliekančių kompiuterių, taip sumažinant srautą visame tinkle.

Šio metodo trūkumas yra tas, kad reikalingos papildomos investicijos, nes dauguma šiuo metu naudojamos įrangos nepalaiko paketų žymėjimo. Dėl tos pačios priežasties šis metodas dažniausiai gali būti taikomas tik interneto paslaugų tiekėjo tinklo ribose.

Šio metodo privalumas – greitesnis suklastotų siuntėjo adresų aptikimas, jei tokie adresai naudojami vykdant ataką. Klastojant siuntėjo adresą gali būti panaudotas adresas, kuris tuo metu yra nepasiekiamas. Jei pasirenkama siųsti kontrolinį paketą į neegzistuojantį adresą, toks paketas nepasiekia siuntėjo ir visas srautas iš to siuntėjo iškart gali būti atmestas.

1.2.2 Aukos apsisaugojimas nuo žinomų atakų

Kartais siekiant greitesnio rezultato, ar turint mažiau resursų, pasirenkama įvykdyti DDoS ataką panaudojant žinomas atakas. Tokios atakos gali būti:

- Nepilni TCP susijungimai, kuomet atsiunčiamas tik SYN paketas.
- Didelių ICMP paketų atsiuntimas.
- Didelis susijungimų skaičius iš vieno mazgo.
- Daug resursų reikalaujančių užklausų siuntimas.

Nuo tokių atakų galima apsaugoti ir pačias atakuojamas tarnybines stotis. Galimi apsisaugojimo būdai:

- Įsilaužimo aptikimo sistemų naudojimas. Įdiegus IDS, galima stebėti įeinančius srautus ir imtis atitinkamų priemonių. Pvz. IDS gali nustatyti siunčiamus žinomų atakų paketus, bei didelį susijungimų skaičių iš vieno mazgo. Tokiu atveju galima tiesiog blokuoti įeinantį srautą iš tų mazgų.
- Laiko, kuomet laukiama ACK paketo, sutrumpinimas. Tokiu atveju serverio resursai, bus trumpesnį laiką išskirti nereikalingam susijungimui. Šis būdas tinkamas apsisaugojimui nuo nepilnų TCP susijungimų.
- IDS galėtų aptikti pakartotinius bandymus iš to paties adreso užmegsti pusiau atidarytas sesijas. Tinkamai sureguliavus slenkstines reikšmes tokie adresai galėtų būti užblokuojami tam tikram laikui.

1.2.3 Juodosios skylės metodas

Kritiniais atvejais DDoS ataka gali būti labai intensyvi ir gali neužtekti turimų resursų atskirti, kuris srautas yra teisėtas, o kuris kenkėjiškas. Tokiu atveju gali būti pasirinkta atmesti visą srautą, pagal kurį vykdoma ataka. Jei tarnybinėje stotyje teikiamos kelios paslaugos, o ataka

vykdoma tik prieš vieną iš jų (Pvz. POP3S, SSH, HTTP serveris atakuojamas HTTP užklausomis), gali būti priimtina tam tikram laikui blokuoti visą HTTP srautą į tą tarnybinę stotį.

Kai kuriais atvejais interneto paslaugų tiekėjas, matydamas, kad įranga nebesugeba apdoroti atakos srauto, gali pradėti blokuoti visus įeinančius paketus, skirtus aukai. Tokiu būdu sumažinama įtaka kitiems interneto paslaugų tiekėjo vartotojams, prieš kuriuos tiesioginė ataka nėra vykdoma.

1.2.4 Registravimas

Atmetant paketus reikia užregistruoti kuo daugiau detalių apie kiekvieną mazgą, dalyvaujantį atakoje. DDoS ataka gali sukelti finansinių nuostolių, todėl atliekant tyrimą gali prireikti informacijos apie atakoje dalyvavusius mazgus. Turint atakoje dalyvavusių mazgų sąrašą taip pat galima tokia informacija pasidalinti su kitais tiekėjais bei įspėti kompiuterių sąvininkus apie jiems priklausančių mazgų dalyvavimą atakoje. Sutvarkius problemas atakuojančiuose mazguose, jie nebebus naudojami ateityje vykstančiose atakose.

1.2.5 DDoS atakų nutraukimo metodų palyginimas

Atakų nutraukimo metodai turi apsaugoti auką nuo vykstančios atakos, suteikti sąlygas kai kuriems vartotojams pasinaudoti sistema, užkirsti kelią atakai iš to paties šaltinio ateityje. Atakų nutraukimo metodų palyginimas pagal šiuos parametrus pateikiamas 2 lentelėje.

2 lentelė. DDoS atakų nutraukimo metodų palyginimas

DDoS atakų nutraukimo metodas	Atakos sustabdymas	Galimybė pasinaudoti paslauga	Naujos atakos išvengimas
Paketų atmetimas atkuriant kelią	+	+	+
Aukos apsaugojimas nuo žinomų atakų	+	+	+
Juodosios skylės metodas	+	-	+
Registravimas	-	+	+

Pagal lentelės rezultatus matome, kad tik naudojant pirmuosius du metodus galima ir sustabdyti ataką, ir suteikti galimybę pasinaudoti paslauga. Juodosios skylės metodu galima galima atmesti srautą, tačiau paslauga nebeteikiama. Panaudojant registravimą ataka nesustabdoma, tačiau surenkama informacija efektyviam atakų sustabdymui ateityje. Šio darbo metu bus taikomas aukos apsaugojimo nuo žinomų atakų metodas kartu su atakų aptikimo metodų duomenimis. Realioje sistemoje turėtų būti ir juodosios skylės bei registravimo metodai, tačiau tyrimui jie įtakos neturi.

1.3 Ugniasienės

Siekiant efektyviai apsisaugoti nuo DDoS atakų, ugniasienės turi sugebėti apdoroti didelius paketų srautus. Standartinės ugniasienės labiau orientuotos į suteikiamą funkcionalumą, o ne į kokybę. Šioje dalyje bus apžvelgiami keli būdai, kaip pagreitinti paketų apdorojimą ugniasienėse. Taip pat bus apžvelgiama OpenBSD operacinėje sistemoje naudojama ugniasienė PF. Ši ugniasienė bus nagrinėjama dėl to, kad tai viena populiariausių ir efektyviausių UNIX šeimos operacinėms sistemoms sukurtų ugniasienių.

1.3.1 PF paketų filtras

OpenBSD operacinėje sistemoje naudojama ugniasienė PF – galingas ir daugelyje BSD sistemų naudojamas įrankis. Daugelis internete paslaugas teikiančių įmonių naudoja šį paketų filtrą dėl jo gerai ištobulinto paketų apdorjimo mechanizmo bei paprasto taisyklių sudarymo.

PF yra būsenomis paremtas paketų filtras, veikiantis operacinės sistemos branduolyje. Tikrina kiekvieną įeinantį ir išeinantį paketą. Paketo rezultatas būna vienas iš šių sprendimų:

- Praleisti paketą jį modifikuojant arba nemodifikuojant;
- Atmesti paketą neatliekan jokių kitų veiksmų;
- Atmesti paketą, siunčiant atsakymą (pavyzdžiui TCP paketą su nustatyta RST vėliavėle).

Pats filtras sudarytas iš dviejų bazinių elementų:

- Filtravimo taisyklių.
- Būsenų lentelės.

Kiekvienas paketas tikrinamas pagal sudarytą taisyklių rinkinį. Taisyklių rinkinį sudaro į dinaminį sąrašą sujungtos taisyklės. Kiekviena taisyklė turi aibę parametrų, kurie tikrinami pagal tą taisyklę. Kiekviena taisyklė taip pat turi nustatytą veiksmą, kuris atliekamas tuo atveju, jei paketas atitinka tasyklę.

Paketas tikrinamas pagal visas taisykles. Taikomas veiksmas pagal tą taisyklę, kuri tiko paskutinė. Taip yra padaryta todėl, kad būtų paprastesnis taisyklių rašymas. Iš pradžių galima apibrėžti platesnes taisykles (pvz. viską drausti), o vėliau siauresnes (praleisti srautą į tam tikrus prievadus). Panaudojus taisyklėje raktinį žodį *quick*, tos taisyklės veiksmas taikomas iškart, jei tik paketas atitinka tą taisyklę.

PF ugniasienė gali filtruoti ne tik pavienius paketus, bet ir visus susijungimus. Tam tikslui naudojama būsenų lentelė. Prieš tikrinant paketą pagal nustatytas taisykles, patikrinama, ar paketas

nepriklauso kuriam nors susijungimui, įrašytam į būsenų lentelę. Jeigu nustatoma, kad paketas yra jau egzistuojančio susijungimo dalis, jis yra praleidžiamas netikrinant jo pagal taisykles. Tokiu būdu išvengiama pakartotinio paketų tikrinimo.

Būsenos saugomos kaip įrašai AVL medyje. AVL medis – tai subalansuotas dvejetainis medis. Tokia duomenų struktūra suteikia greitą paiešką ir gerai veikia su didesnės apimties duomenimis. Netgi blogiausiu atveju užtikrinama tokia pati veikimo sparta O(log *n*).

PF naudojami tam tikri taisyklių optimizavimo metodai. Jeigu grupė taisyklių naudoja bendrą parametrą (pvz. nurodytas siuntėjo IP adresas) ir paketas neatitinka to parametro, kuomet tikrinama pirma tos grupės taisyklė, visos kitos taisyklės iš tos grupės yra praleidžiamos. Kuomet taisyklės įkraunamos, operacinės sistemos branduolys patikrina visas taisykles ir sudaro praleidimo žingsnius. Kiekvienos taisyklės kiekvienam parametrui nustatoma rodyklė į kitą taisyklę, turinčią kitokią to parametro reikšmę.

Taip sutvarkytos ugniasienės veikimo greitis priklauso nuo pačių taisyklių. Blogiausiu atveju visi praleidimo žingsniai nurodo į kitą iš eilės einančią taisyklę ir paketas tikrinamas iš eilės pagal visas taisykles. Tačiau net ir blogiausiu atveju veikimo greitis nesuprastėja, lyginant su neoptimizuota versija.

Labai svarbu atkreipti dėmesį į tai, kad PF taisyklės atmintyje išdėliojamos stengiantis išlaikyti pradinę taisyklių surašymo tvarką konfigūracijos faile. Dėl šios priežasties visos taisyklės, įterpiamos į PF taisyklių rinkinį aptikus DDoS atakas turės būti įterpiamos kuo arčiau taisyklių rinkinio pradžios. Tokiu būdu vykstant atakai kiekvienas įeinantis paketas bus tikrinamas pagal tas taisykles, kurios reikalingos atakoje dalyvaujančių mazgų blokavimui.

1.3.2 Ugniasienių taisyklių valdymas

1.3.2.1 Dinaminis taisyklių pertvarkymas

Ugniasienės taisyklės dažniausiai saugomos kaip sutvarkytas sąrašas sudarytas iš *n* paketų filtravimo taisyklių. Kiekvienas paketas iš eilės tikrinamas pagal kiekvieną iš tų taisyklių, kol randama tinkama taisyklė. Jei nerandama tinkama taisyklė, pritaikomas standartinis veiksmas (rekomenduojama paketo atmetimas). Galima teigti, kad paketo apdorojimo laikas priklauso nuo taisyklių skaičiaus, nes blogiausiu atveju paketas gali būti patikrintas pagal visas taisykles.

Dinaminio taisyklių pertvarkymo metodo esmė – sumažinti taisyklių, pagal kurias tikrinamas paketas, skaičių. Tokiu atveju paketui turi būti pritaikytas pagal pirmą atitikusią taisyklę priklausantis veiksmas iškart, kai randama pirmoji atitikusi taisyklė. Kiekvienai taisyklei skaičiuojama, kiek kartų tikrinti paketai atitiko tą taisyklę. Kuo didesnis atitikusių paketų skaičius –

tuo aukščiau sąraše taisyklė perstumiama. Atlikus perstatymus tos taisyklės, kurios dažniausiai atitinka keliauja į sąrašo viršų ir dauguma paketų tikrinami tik pagal vieną taisyklę.

Vykstant DDoS atakai, taip veikiančios ugniasienės taisykles būtų galima sudaryti taip:

- Visus paketus, kurie pripažinti teisėtais, praleisti.
- Visus kitus paketus – atmesti.

Kiekvienas DDoS atakoje dalyvaujantis paketas atitiktų atmetimo taisyklę taip vis didindamas aititikusių taisyklę paketų skaičių. Per tam tikrą laiką taisyklė pakiltų į patį sąrašo viršų ir visi paketai, pripažinti, kaip dalyvaujantys DDoS atakoje būtų atmetami atlikus tik vieną patikrinimą.

1.3.2.2 Duomenų struktūrų panaudojimas taisyklėms saugoti

Dar vienas būdas, kaip paspartinti paketų patikrinimą ugniasienėje – naudoti optimalesnes duomenų struktūras taisyklėms saugoti. Ugniasienės taisykles galima dėti į medžių tipų duomenų struktūras, jas rikiuojant pagal tam tikrus požymius. Tokiu atveju bet kokios taisyklės laikas patikrinimo laikas sutrumpėja ir visų paketų aptarnavimo laikai pasidaro panašūs.

Tobulesnių duomenų struktūrų panaudojimo privalumai:

- Netiesioginė paketų apdorojimo priklausomybė nuo taisyklių skaičius (dažniausiai logaritminė).
- Greitas naujų taisyklių įterpimas.
- Dauguma paketų apdorojami beveik vienodu greičiu.

Šio metodo trūkumas – nėra galimybės išskirti tam tikro tipo paketų, kurie būtų apdorojami su didesniu prioritetu, o tai yra reikalinga, kuomet stengiamasi sustabdyti DDoS atakas.

1.4 Skyriaus apibendrinimas

Šiame skyriuje išnagrinėti pagrindiniai DDoS atakų tipai, aptartos svarbiausios charakeristikos, jų daroma žala. Išnagrinėti jau sukurti metodai DDoS atakų aptikimui. Atlikus analizę nustatyta, kad efektyviausia DDoS atakų aptikimui naudoti kelis skirtingus mazgus, nes tai padeda paskirstyti srautą ir sumažinti vieno mazgo atskyrimą. Taip pat nustatyta, kad visi nagrinėti metodai remiasi statistine srauto analize, tačiau išsiskiria stebimais parametrais ir gautos informacijos apdorojimo būdais.

Darbe nuspręsta išsamiau tirti DDoS atakų aptikimo metodus veikiančius viename mazge. Paskirstyti metodai nebus tiriami, nes jų realizavimui ir testavimui reikia sudėtingesnės

architektūros. Nagrinėjamiems metodams iškelti pagrindiniai kriterijai: 1) algoritmo paprastumas ir minimalus operacijų skaičius; 2) galimybė identifikuoti užtvindymo atakos tipą; 3) greita reakcija į prasidėjusią ataką. Pagal iškeltus kriterijus pasirinkta, realizuoti pokyčio taško aptikimo, kovariacijos ir pasyviais skaičiavimais paremtos euristikos DDoS aptikimo metodus.

Išnagrinėti DDoS atakų sustabdymo būdai, leidžiantys tinkamai atlikti nepageidaujamo didelės apimties srauto blokavimą. Nustatyta, kad norint efektyviai blokuoti nepageidaujamą srautą reikalinga turėti efektyviai veikiančias ugniasienes bei DDoS atakų aptikimo sistemas. Tokiu atveju blokuojamas srautas iš atakoje dalyvaujančių adresų. Kritiniais atvejais gali būti blokuojamas visas srautas į atakuojamą paslaugą, arba apskritai visas srautas į atakuojamą tarnybinę stotį.

Aptartas OpenBSD operacinėje sistemoje naudojamos ugniasienės PF veikimo principas bei išnagrinėti du ugniasienių taisyklių išdėstymo atmintyje metodai, leidžiantys ugniasienės taisykles išdėstyti taip, kad nebūtų gaištamas laikas tikrinant atakoje dalyvaujančius paketus pagal taisykles, kurios nėra reikalingos jiems blokuoti.

Tolimesniuose darbo etapuose numatoma suprojektuoti ir pokyčio taško aptikimo, kovariacijos ir pasyviais skaičiavimais paremtos euristikos DDoS atakų aptikimo metodus bei praktiškai ištirti jų charakteristikas. Kuriama DDoS atakų aptikimo sistema, turėtų leisti taikyti pasirinktą DDoS atakų identifikavimo metodą. Kuriamą sistemą numatoma integruoti su PF ugniasiene siekiant taisykles atmintyje išdėstyti taip, kad teisėtus srautus praleidžiančios ir atakas blokuojančios taisyklės būtų tikrinamos anksčiau negu visos kitos atmintyje esančios ugniasienės taisyklės.

DDoS atakų aptikimo metodų tyrimo metu nerasta metodų, kurie bandytų spręsti atakoje dalyvaujančių ir teisėtų srautų atskyrimo problemas. Kadangi tai yra svarbus faktorius, siekiant bent iš dalies užtikrinti teikiamų paslaugų kokybę net ir vykstant atakai, numatoma pasiūlyti patobulinimus jau egzistuojantiems DDoS atakų aptikimo metodams, kurie padėtų atskirti atakoje dalyvaujančius ir teisėtus srautus bei sumažinti blokuojamų teisėtų paketų skaičių.

2 Projektinė dalis

2.1 Sistemos reikalavimų specifikacija

2.1.1 Funkciniai reikalavimai

Kuriamai sistemai keliami reikalavimai, paremti ankstesniojo skyriaus išvadomis. Toliau pateikiamas sąrašas savybių, kuriomis turi pasižymėti kuriama sistema:

- Aptikti DDoS atakas;
- Reaguoti į DDoS atakas;
- Sugebėti apdoroti kuo didesnius duomenų srautus;
- Pateikti vartotojui visą reikiamą informaciją apie sistemos būseną;
- Leisti vartotojui valdyti DDoS atakų aptikimo būdus;
- Palikti vartotojui galimybę ar reaguoti į ataką, ar ne;
- Registruoti DDoS atakoje dalyvaujančius mazgus;
- Atskirti atakoje dalyvaujančius srautus nuo teisėtų srautų.

Sistemą sudarantys komponentai:

- DDoS atakų aptikimo komponentas;
- Atakoje dalyvaujančių adresų aptikimo komponentas;
- Ugniasienės taisyklių valdymo komponentas;
- Sistemos valdymo programa.

Sistemos atliekamos funkcijos :

- Kuriama sistema turi sugebėti aptikti DDoS atakas, naudodama pasirinktu kelis pasirinktus DDoS atakų aptikimo metodus, iš pirmojoje dalyje aptartų metodų;
- Pagal pasirinktą metodą turi būti apdorojami įeinantys paketai;
- Aptikus, kad pradėta vykdyti DDoS ataka, kiekvienas adresas, kuris pripažintas kaip dalyvaujantis atakoje, yra blokuojamas;
- Visos naujai įdedamos taisyklės turi būti optimizuojamos, kad greičiau būtų apdorojami konkretūs paketai;
- Vartotojas gali reguliuoti, ar taisyklės optimizuojamos automatiškai po kiekvieno

būsenos pakitimo, ar po atskiro kreipinio iš vartotojo lygmens programos;

- Apdorojant įeinančius paketus pagal ugniasienės taisykles, kinta paketų, praleistų arba atmestų pagal tam tikrą taisyklę, kiekis, todėl reikia užtikrinti, kad taisyklių sąrašo pradžioje būtų tos taisyklės, pagal kurias turi būti blokuojami atakoje dalyvaujantys paketai.

Valdymo programa, naudojama DDoS atakų aptikimo metodui išrinkti, sistemos būsenos informacijai nuskaityti, aktyvuoti ugniasienės taisyklių optimizavimą. Nuskaitant sistemos būseną, pateikiama tokia informacija:

- Susijungimai, pripažinti kaip dalyvaujantys DDoS atakoje. Grąžinamas siuntėjo adresas, gavėjo portas ir laikas, kuomet buvo užregistruotas susijungimas;
- Kiekis susijungimų, pripažintų teisėtais;
- Vidinė pasirinkto paketų apdorojimo metodo informacija.

Dar viena svarbi valdymo programos funkcija – sistemos parametrų apkrovos stebėjimas, bei perėjimo į „juodos skylės“ režimą galimybė. Tai yra žinant, prieš kokią paslaugą vykdoma ataka bei pasiekus nustatytas resursų apkrovos ribas, įdedama taisyklė blokuoti visą srautą į atakuojamą paslaugą. Tokiu būdu būtų siekiama užtikrinti, kad kitos sistemoje teikiamos paslaugos liktų pasiekiamos atakos metu. Sprendimas gali būti priimamas automatiškai, arba inicijuojamas vartotojo.

2.1.2 Nefunkciniai reikalavimai

Kuriamai sistemai keliami tokie nefunkciniai reikalavimai:

- DDoS atakų aptikimo metodai turi būti realizuoti naudojant priemones, kurių pagalba metodus būtų galima taikyti keliose operacinėse sistemose.
- Kiekvienas DDoS atakų aptikimo metodas turi būti realizuojamas kaip atskira funkcija, kad esant poreikiui jį būtų galima iškelti į atskirą programą.

2.1.3 Reikalavimai programos kodui

Laikomasi OpenBSD nurodyto branduolio programavimo stiliaus (iš „Kernel source file style guide (KNF)“).

Rašant programos kodą laikomasi bendrų C kalbos programavimo principų, kur įmanoma stengiamasi panaudoti jau sukurtas sąsajas ar bibliotekas.

2.1.4 Sistemos taikymo sritis ir apribojimai

Sistema skirta DDoS atakų aptikimui ir sustabdymui galiniame taške. Naudojami tie DDoS atakų aptikimo metodai, kurie nereikalauja duomenų iš kelių maršrutizatorių. Daroma prielaida, kad visi maršrutizatoriai, esantys kelyje nuo teisėto vartotojo iki mazgo, kuriame bus įdiegta paslauga, veikia tinkamai DDoS atakos metu, t.y. ataka daro įtaką tik galinio mazgo teikiamų paslaugų prieinamumui.

Sistemos kūrimo ir testavimo metu taip pat priimama, kad yra viršutinė riba įeinančių paketų kiekiui, kurį gali apdoroti operacinė sistema. Visi sukurtos sistemos testavimai bus atliekami prieš tai nustačius šią ribą.

Paketų atmetimui naudojamas paketų blokavimas galiniame taške. Blokuojami visi paketai, kurių siuntėjai įtariami dalyvavimu atakoje.

Kuriama sistema bus orientuota pirmiausiai darbui su OpenBSD operacine sistema. Srautų blokavimui bus naudojamos PF valdymo komandos. Realizuojant DDoS atakų aptikimo metodus paketų, atsiunčiamų į tinklo sąsają nuskaitymui, bus naudojama pcap biblioteka. Dėl šių priežasčių DDoS atakų aptikimo mechanizmą, realizuotą vartotojo lygyje bus galima pritaikyti kitoms operacinėms sistemoms.

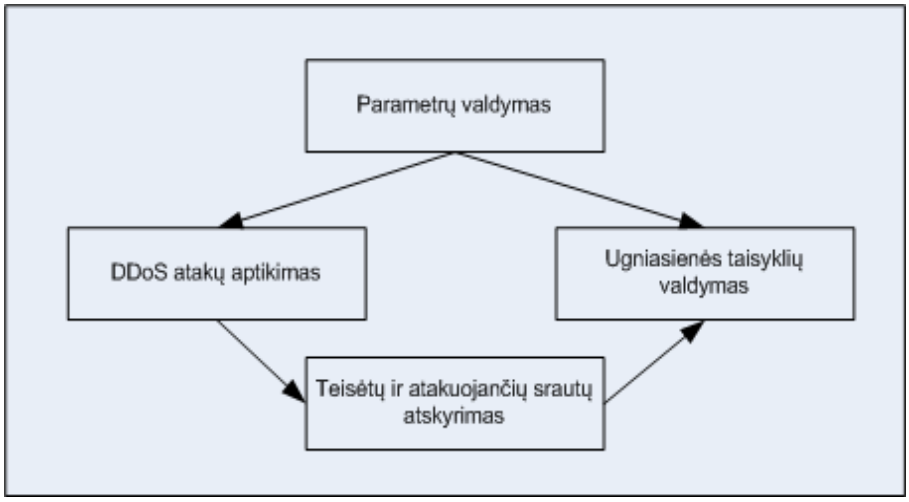
Sistema bus skirta tik ketvirtąją IP protokolo versiją naudojantiems tinklams. Šis apribojimas pirmiausiai kyla teisėtų ir atakoje dalyvaujančių srautų atskyrimo metode, nes didžioji dalis visų galimų IP adresų gali būti saugoma operatyviojoje atmintyje. Šeštosios versijos IP protokolo visų adresų net ir su nemažu persidengimu operatyviojoje atmintyje išdėstyti nėra galimybės.

2.2 Kuriamos sistemos architektūra

Kuriamą sistemą sudarys tokie tarpusavyje sąveikaujantys komponentai:

- DDoS atakų aptikimo
- Atakos šaltinių identifikavimo
- Ugniasienės taisyklių kūrimo
- Sistemos parametrų valdymo

Toliau pateikiama sistemos architektūros grafinė schema:



3 pav. Sistemos architektūros grafinė schema

Šioje schemoje pavaizduoti pagrindiniai komponentai ir numatomi duomenų srautai tarp jų Vartotojo nustatyti parametrai įtakos atakų aptikimo metodų ir taisyklių kūrimo komponentų veikimą. Pagal analizuotus paketus atakų aptikimo metoduose ir tiesioginių bei atakoje dalyvaujančių srautų atskyrimo modulį bus siunčiama informacija taisyklių kūrimo moduliui. Kuriama sistema veiks kaip atskiras procesas operacinėje sistemoje, todėl paketai, patenkantys atakų aptikimo metodams, pirmiausia bus apdoroti operacinės sistemos branduolio ir branduolio lygmenyje veikiančios ugniasienės.

2.2.1 DDoS atakų aptikimo komponentas

DDoS atakų aptikimo komponentas programos atmintyje saugo konkrečiam veikiančiam metodui reikalingas duomenų struktūras ir kitą vidinę informaciją. Kiekvienas į sistemą patekęs paketas perduodamas šiam komponentui ir jame yra apdorojamas. Apdorojant kiekvieną paketą, kinta su konkrečiu algoritmu susiję parametrai. Kuomet pasiekiamos tam tikros reikšmės, arba fiksuojami staigūs parametrų pokyčiai – galima nustatyti, kad prasidėjo DDoS ataka.

2.2.2 Atakos šaltinių identifikavimo komponentas

Vykstant DDoS atakai, gali atsirasti poreikis užblokuoti srautą, siunčiamą iš atakoje dalyvaujančių mazgų. Blokuoti visą įeinantį srautą nėra optimalu, nes nukenčia teisėtų vartotojų siunčiami paketai. Literatūros analizės metu nebuvo rasta metodų, kurie mėgintų spręsti atakoje dalyvaujančių ir teisėtų srautų atskyrimo problemą, todėl šioje dalyje bei realizacijoje numatoma suprojektuoti, realizuoti ir ištestuoti tokį metodą.

Teisėtų ir atakoje dalyvaujančių srautų atskyrimą autorius siūlo organizuoti principu, panašiu į nepageidaujamų reklaminių laiškų (SPAM) blokavimo būdą. Sistemai veikiant ir atliekant stebėjimus, tam tikrą laiko tarpą bus registruojami IP adresai, iš kurių gaunami paketai. Kuomet

vienas iš atakų aptikimo metodų fiksuoja, kad prasidėjo ataka, visi nauji IP adresai, kurie prieš tai nebuvo užregistruoti, traktuojami, kaip dalyvaujantys atakoje ir yra įtraukiami į blokuojamų adresų sąrašą. Srautai iš adresų, kurie buvo užregistruoti iki tada, kol ataka nebuvo prasidėjusi, yra praleidžiami.

Reikalavimai šio metodo realizacijai:

- Greitas adresų apdorojimas – reikia, kad adreso tikrinimas ir naujo adreso registravimas būtų atliekamas panaudojant minimalų operacijų skaičių.
- Adresų registravimas – reikia galimybės užregistruoti adresą, dalyvavusį atakoje.

2.2.3 Taisyklių valdymo komponentas

Identifikavus DDoS atakos pradžią bei atakoje dalyvaujančius mazgus, sistema turi tikrinti, ar įjungta srautų blokavimo funkcija. Jei reikia blokuoti srautus, šaltinių adresai nustatomi pagal atakos šaltinių identifikavimo modulio informaciją. Srautų blokavimas vyksta įdedant atitinkamas PF taisykles. Taisyklės turi būti įdedamos tokia tvarka, kad atakoje dalyvaujantys srautai būtų apdorojami prieš visas kitas taisykles.

Realizacijos etape reikalinga PF ugniasienės taisyklių rinkinį sudaryti taip, kad naujų taisyklių, susijusių su DDoS atakoje dalyvaujančių srautų blokavimu įterpimas būtų kuo greitesnis, o naujos taisyklės automatiškai atsidurtų taisyklių sąrašo viršuje.

2.2.4 Sistemos parametrų valdymo komponentas

Sistemos parametrų valdymo programa realizuojama kaip vartotojo lygio (*angl.* userland*)* programa, kurioje pagal įvestas komandas suformuojami duomenys ir iškviečiamos atitinkamą atakų aptikimo metodą realizuojančios funkcijos.

Valdymo programoje nustatomos slenkstinės kiekvieno metodo reikšmės. Tikrinant šias reikšmes su kiekvieno metodo vidinėmis reikšmėmis, priimamas sprendimas, ar vyksta DDoS ataka, ar ne. Pagal sistemos parametrus taip pat priimamas sprendimas, ar kreiptis į taisyklių valdymo komponentą, ar ne.

2.3 Duomenų srautų modelis

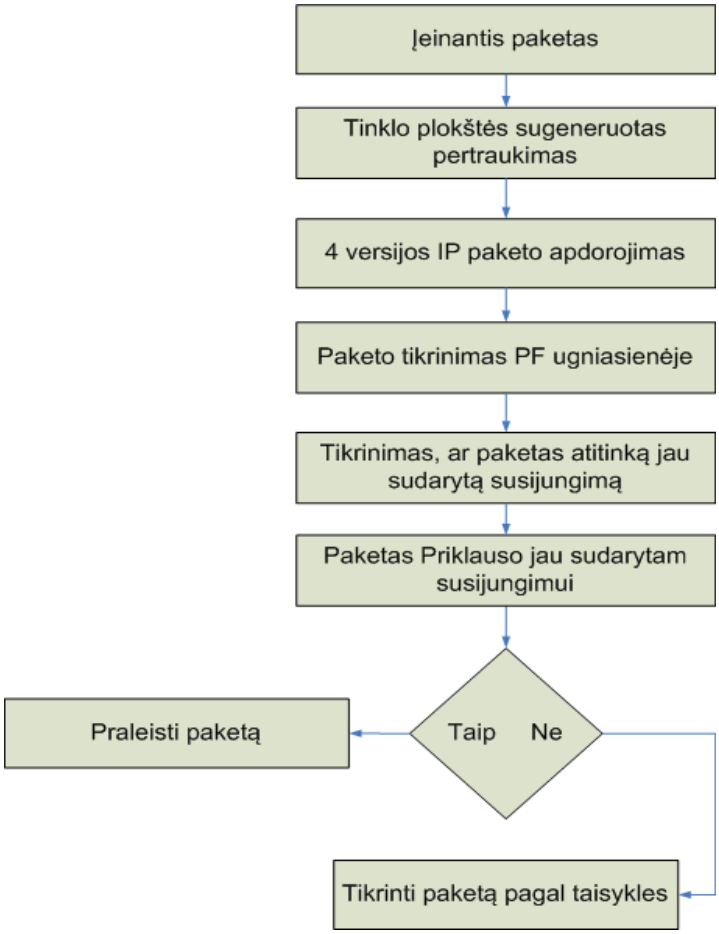
2.3.1 Paketo kelias sistemos branduolyje

Atėjus paketui, valdymas perduodamas į branduolio funkciją `ipintr`. Joje paketas yra apdorojamas. Vienas iš žingsnių – paketo perėjimas per ugniasienę. Kviečiama funkcija `pf_test()`, atlikti ugniasienėje numatytiems veiksmams su tuo paketu.

Perdavus valdymą ugniasienei, tikrinama, ar tuo paketu pradedamas naujas susijungimas, ar ateinantis paketas priklauso jau sudarytam susijungimui. Tam naudojama funkcija *pf_test_state*. Šiame žingsnyje paketas tikrinamas su kiekvienu būsenų sąraše esančiu įrašu. Jei nustatoma, kad paketas priklauso jau egzistuojančiam susijungimui, tolimesnis paketo tikrinimas nutraukiamas ir paketas praleidžiamas. Jei nustatoma, kad paketas nepriklauso jokiam anksčiau sudarytam susijungimui, paketas apdorojamas pagal sudarytą taisyklių sąrašą.

Atlikus paketo patikrinimą ugniasienėje, jei pektas praleidžiamas, valdymas perduodamas tolimesnėms funkcijoms ir paketas pasiekia vartotojo lygmenyje veikiančius procesus.

4 pav. pateikiamas paketo kelias nuo tinklo interfeiso iki paketą atitikusioje ugniasienės taisyklėje aprašyto veiksmo pritaikymo tam paketui.



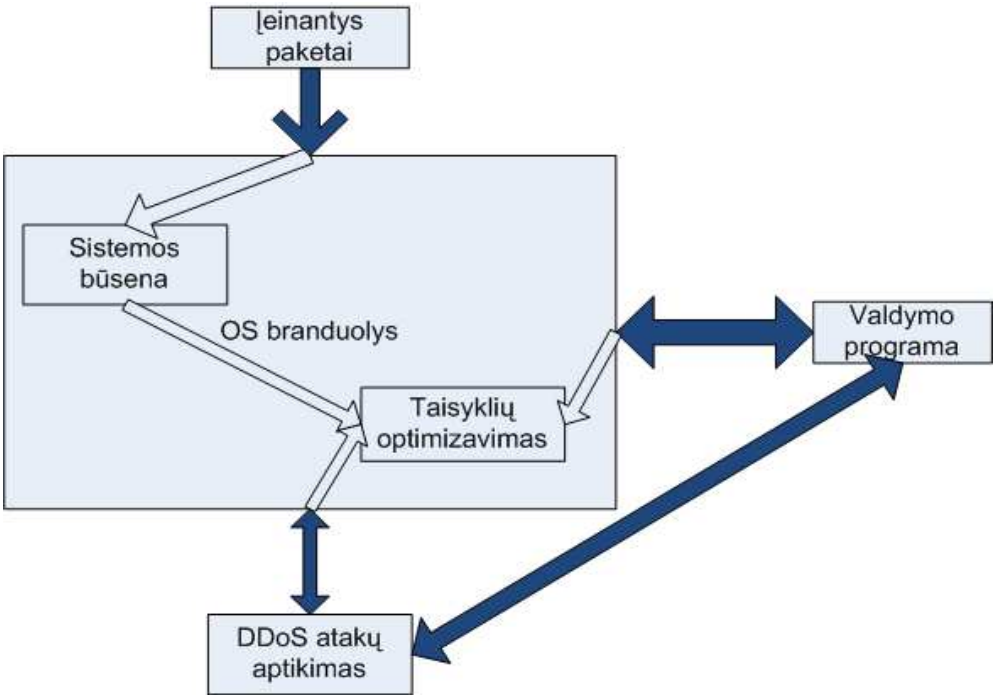
4 pav. Paketo kelio branduolyje grafinė schema

4 pav. pateiktoje sutrumpintoje paketo kelio branduolyje grafinėje schemoje pavaizduoti tik tie žingsniai, kurie yra svarbūs paketų filtravimui ir DDoS atakų aptikimui.

2.3.2 Grafinės duomenų apsikeitimo tarp komponentų schemos

5 pav. pateiktoje schemoje pavaizduota grafinė duomenų apsikeitimo tarp sistemos komponentų schema, kuomet DDoS atakų aptikimo komponentas realizuojamas vartotojo

lygmenyje veikiančiame procese:



5 pav. Duomenų srautų schema, kuomet DDoS aptikimas realizuotas vartotojo lygyje

Šiuo atveju skiriasi duomenų perdavimas DDoS atakų valdymo komponentui. Įeinantys paketai skaitomi pasinaudojant operacinės sistemos pateikiama sąsaja (OpenBSD naudojamas BPF pseudo įrenginys), tačiau į BPF pakliūva visi pagal taisykles praleisti paketai, įskaitant ir tuos, kurie priklauso jau sudarytiems susijungimams. Apdorojant didelius srautus paketų branduolio lygyje, galima nebenagrinėti paketų, kurie priklauso jau sudarytiems susijungimams. Realizavus DDoS atakų aptikimo komponentą vartotojo lygyje reikia arba tikrinti visus paketus, arba atskirai realizuoti susijungimų būsenų sekimą. Vartotojo programą pasiekia tik tie paketai, kuriuos praleidžia ugniasienė.

2.4 Algoritmų aprašymas

2.4.1 DDoS atakų aptikimą realizuojantys algoritmai

Atsižvelgiant į analizės dalyje atliktą DDoS atakų aptikimo metodų palyginimą, realizuojami šie atakų aptikimo algoritmai:

- Pokyčio taško aptikimas;
- Kovariacinis analizės metodas;
- Pasyviais skaičiavimais paremta euristika.

Toliau bus pateikiamas kiekvieno metodo veikimo aprašymas, bei grafinė kiekvieno algoritmo schema.

2.4.1.1 Pokyčio taško aptikimas

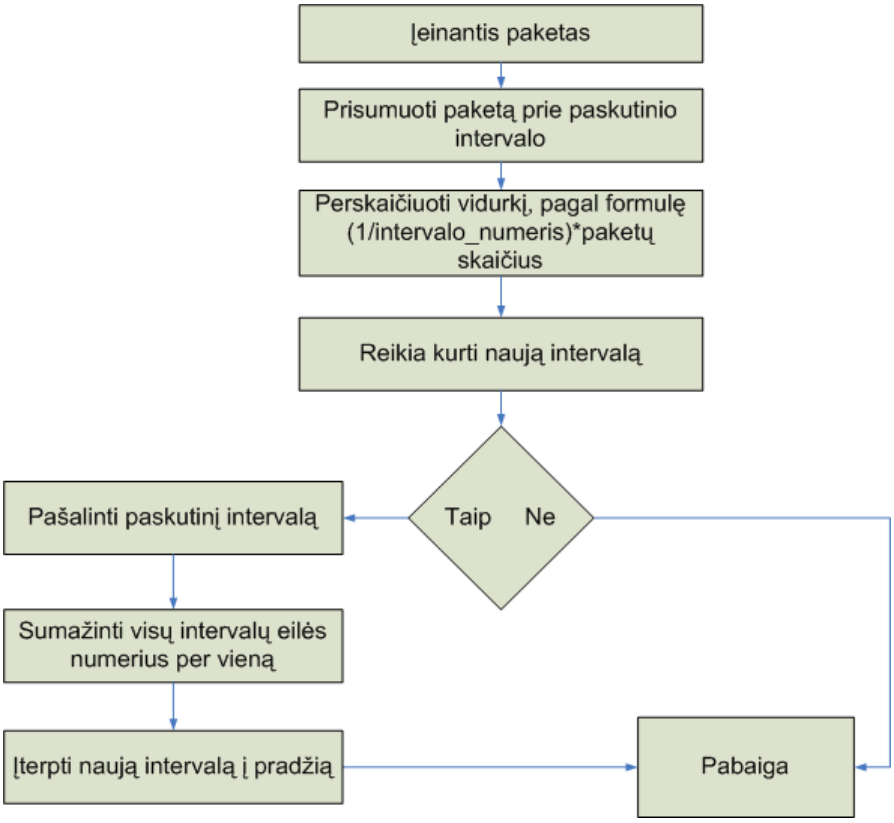
Šis algoritmas paremtas kaupiamosios sumos skaičiavimu. Kaupiamoji suma skaičiuojama pagal tokiąveiksmų seką:

1. Fiksuojamas paketų skaičius per nurodytą intervalą.
2. Nustatomas fiksuotas intervalų skaičius ir stebimas vidutinis paketų skaičius per kiekvieną intervalą.
3. Pradedama nuo 0 ir kiekvienam intervalui kaupiamoji suma skaičiuojama prie sukauptos sumos pridedant nagrinėjamo intervalo paketų skaičių ir atimant paketų skaičiaus vidurkį.
4. Paskutiniame intervale kaupiamoji suma turi gautis 0.
5. Vertinami įvairūs parametrai (pvz. Skirtumas tarp maksimalios ir minimalios paketų skaičiaus reikšmės tarp stebimų intervalų).

Prasidėjus DDoS atakai, atsiranda paketų vidurkio nuokrypiai nuo apskaičiuoto vidurkio, viršyjantys nustatyta maksimalią leistiną nuokrypio reikšmę.

Kaupiamosios sumos panaudojimas leidžia įvertinti konkrečiu laiko momentu nagrinėjamus paketus užduoto laiko intervalo kontekste.

Metodo grafinė schema pateikta 6 pav.



6 pav. Pokyčio taško aptikimo algoritmo schema

2.4.1.2 Kovariacinis analizės metodas

Kovariacija – vienas iš koeficientų, reikalingas įvertinti, ar 2 kintamieji tarpusavyje yra susiję, ar ne. Kai cov(x, y) = 0, ryšio tarp dviejų kintamųjų nėra. Kai cov(x, y) < 0, susiję priešinga priklausomybe (vienam didėjant, kitas mažėja). Kai cov(x, y) > 0, abu kintamieji kinta ta pačia

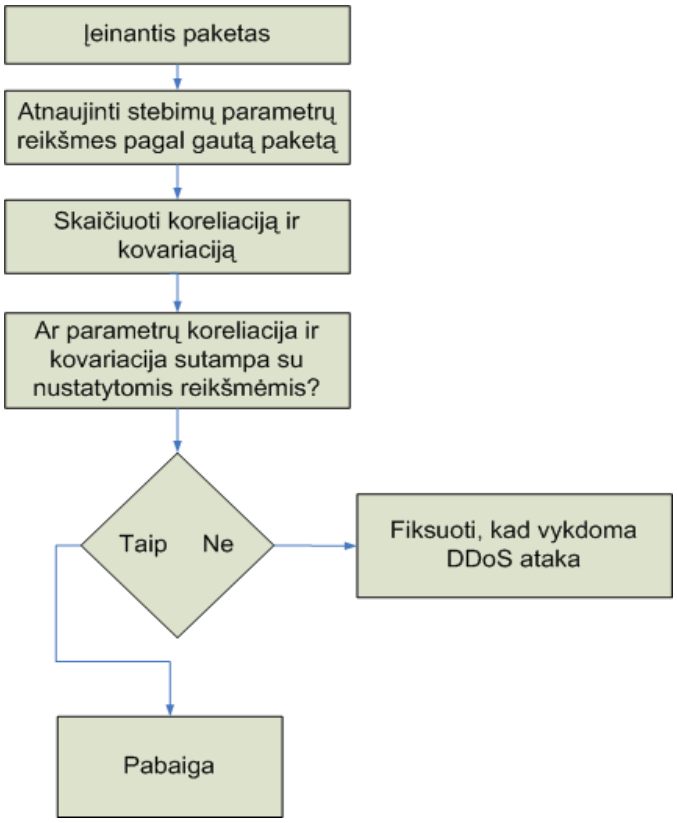
kryptimi.

Išrenkami stebimi srauto parametrai. Pagrindiniai stebimi parametrai:

- TCP SYN (susijungimo užmezgimo) paketų skaičius per tam tikrą laiko intervalą;
- Analizuojamo paketo TTL (time to live – mazgų skaičius, per kuriuos gali pereiti paketas, kol pasiekia adresatą) reikšmė.

Parametrai registruojami ir skaičiuojama jų koreliacija. Normalių srautų parametrų koreliacija skirsis nuo tos srautų koreliacijos, kuomet vykdoma DDoS ataka.

7 pav. pateikta metodo grafinė schema:

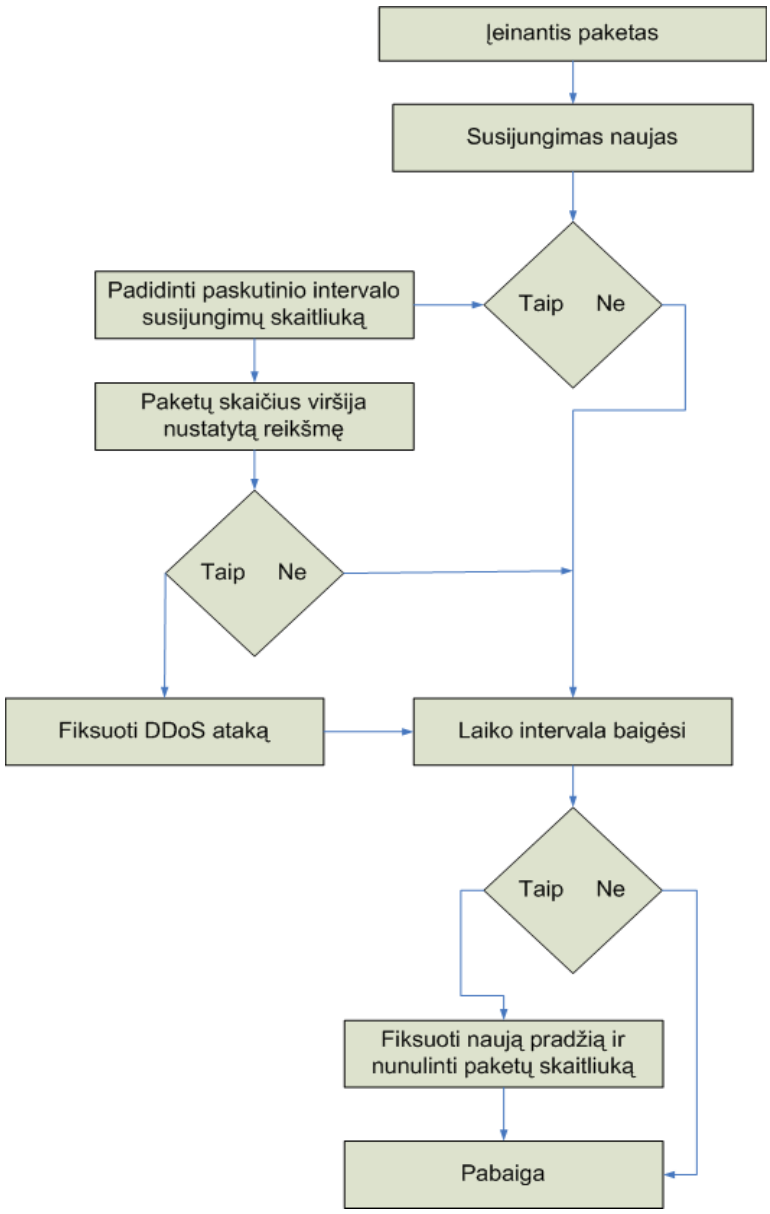


7 pav. Kovariacinės analizės metodo algoritmo schema

2.4.1.3 Pasyviais skaičiavimais paremta euristika

Šio metodo veikimo metu, skaičiuojamas naujų susijungimų skaičius per nustatytą laiko intervalą esant normaliam srautui. Prasidėjus atakai, naujų susijungimų skaičius per nustatytą intervalą viršys iš anksto nustatytą reikšmę.

8 pav. pateikta metodo grafinė schema:



8 pav. Pasyviais skaičiavimais paremtos euristikos algoritmo schema

2.4.2 Teisėtų ir atakoje dalyvaujančių srautų atskyrimo algoritmas

Dar vienas sistemoje naudojamas algoritmas naudojamas teisėtų ir atakoje dalyvaujančių srautų atskyrimui. Šis metodas yra sukurtas šio darbo metu kaip patobulinimas jau esantiems atakų aptikimo metodams.

Vykdant atakoje dalyvaujančių ir teisėtų srautų identifikavimą, tikrinama, ar vyksta ataka. Jei ataka vyksta ir mazgas nėra užregisruotas anksčiau, adresas registruojamas, kaip dalyvaujantis atakoje. Jei ataka nevyksta, adresas registruojamas į nedalyvaujančių atakoje mazgų sąrašą.

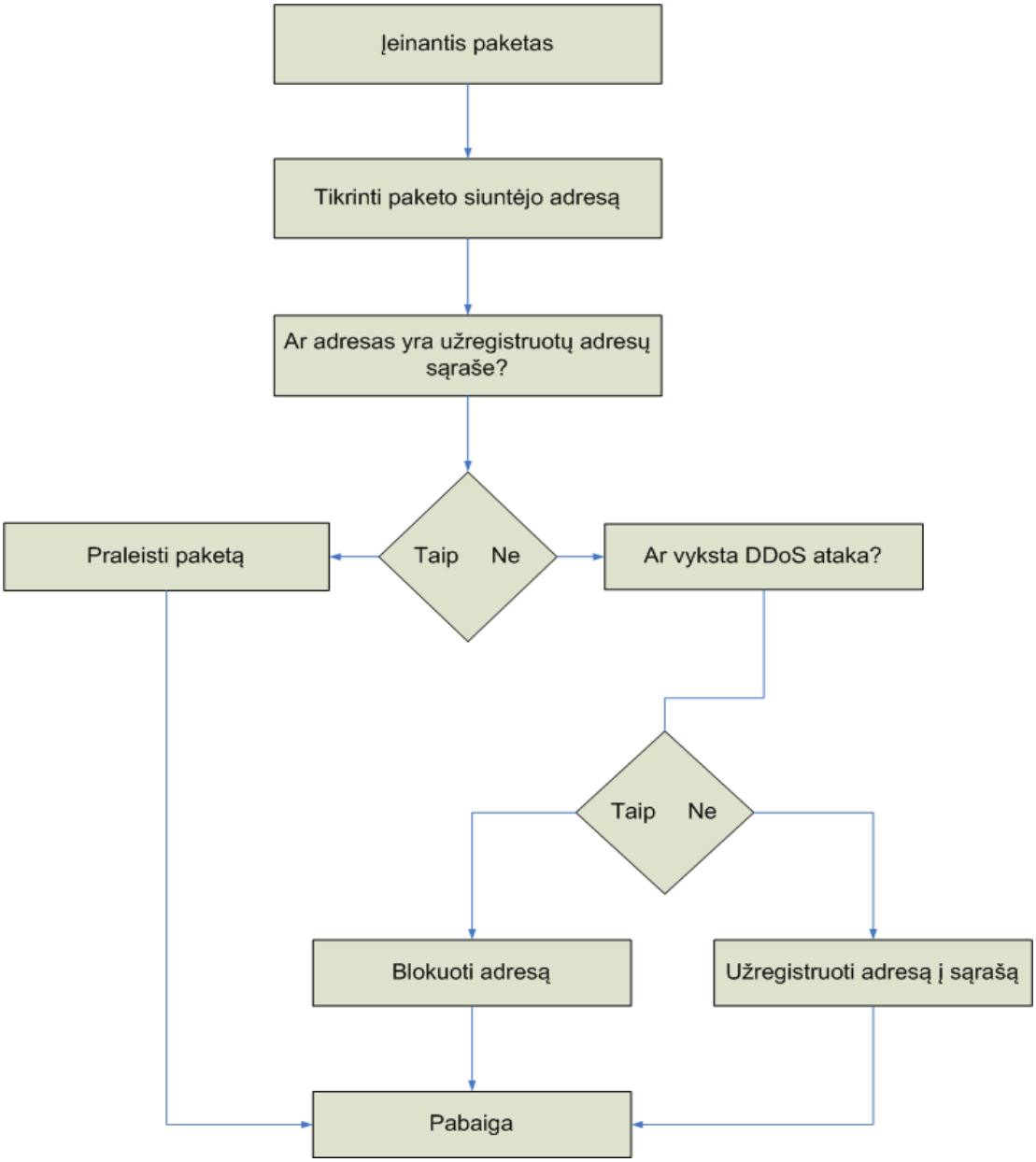
Pagrindinis šio metodo trūkumas yra tas, kad jeigu atakuojantys mazgai siunčia paketus dar prieš aptinkant ataką, jų adresai gali būti įtraukiami į teisėtų paslaugos gavėjų adresų sąrašą ir būti praleidžiami netgi vykstant atakai. Šio darbo metu realizuojamas metodas visus naujus adresus, iš kurių kreipiamasi į atakuojamą adresą po atakos paskelbimo, įtraukia į blokuojamų adresų sąrašą,

jei jie nebuvo kreipęsi seniau.

Dar viena problema iškyla tada, kai atakoje dalyvauja virusais užkrėsti ar kitaip nuotoliniu būdu valdomi tiesėtų paslaugos gavėjų kompiuteriai, kurių adresai jau yra užregistruoti teisėtų paslaugos gavėjų sąraše.

Norint išvengti teisėtų adresų įtraukimo į blokuojamų adresų sąrašą reikėtų sistemai kurį laiką veikti realiomis sąlygomis be DDoS atakų ir registruoti visus IP adresus, iš kurių kreipiamasi. Tokiu būdu į sąrašą pakliūtų dauguma vartotojų, kuriems paslauga yra tikrai reikalinga ir atakos metu jie nepatektų į blokuojamų adresų sąrašą.

9 pav. pateikta metodo grafinė schema:



9 pav. Teisėtų ir atakoje dalyvaujančių srautų atskyrimas

Šį metodą galima taikyti adresų registravimui kai vykdoma bet kokio tipo ataka. Daugiausiai

naudos šio metodo panaudojimas suteikia tada, kai ataka vykdoma siunčiant teisėtas užklausas, tačiau naudojant didelį skaičių atakuojančių mazgų siekiant išvesti iš rikiuotės atakuojamą serverį. Jei vykdomos žinomos atakos, šio metodo taikyti nebūtina, nes atakuojančius mazgus galima aptikti analizuojant konkrečią ataką atpažįstančio atakos aptikimo metodo duomenis. Pavyzdžiui SYN užtvindymo atveju galima tiesiog registruoti siuntėjus, kurie per nustatytą laiką neatsiunčia FIN paketo. Jei ataka vykdoma siunčiant didelį kiekį žinomas atakas išnaudojančių paketų (komandos išnaudojančios klaidas protokolouose ar servisų realizacijose), tokius paketus gali aptikti ir blokuoti IDS, arba IDS naudojamą paketų analizavimo mechanizmą galima nesunkiai įdiegti į šiame darbe kuriamą sistemą.

2.5 Testavimo planas

Testavimas atliekamas simuliuojant tinklą su skirtingų adresų klasėmis. Tam, kad būtų lengviau pasiekti atakuojamos sistemos naudojamų resursų ribas simuliuojant atakas vietiniame tinkle, atakuojamai sistemai išskiriami mažesni CPU ir operatyviosios atminties resursai, nei įprastuose šiuolaikiniuose serveriuose.

Testuojant siekiama patikrinti, kaip kinta sistemos apkrovos parametrai, vykdant tokią pačią ataką, tačiau keičiant sistemos nustatymus:

- Naudojant skirtingus DDoS atakų aptikimo metodus;
- Pritaikius srautų blokavimą.

Numatomi testavimo etapai:

- Testuojant skirtingus DDoS atakų aptikimo metodus siekiama nustatyti, kokį didžiausią srautą sistema sugeba apdoroti, bei kokią papildomą apkrovą sistemai prideda vieno ar kito atakos aptikimo metodo naudojimas.
- Testuojant DDoS atakoje dalyvaujančių adresų registravimo ir blokavimo komponentus atliekamas testas, ar atakos metu atsakoma į užklausą, kuri turėtų būti pripažinta kaip teisėta.
- Testuojamas srautų blokavimo modulio veikimas. Bus tikrinama, ar taisyklės įdedamos teisinga tvarka. Tai yra, ar atakoje dalyvaujantys paketai tikrinami pagal taisykles, esančias taisyklių sąrašo pradžioje.

2.6 Skyriaus apibendrinimas

Šiame skyriuje nustatyti funkciniai ir nefunkciniai reikalavimai numatomai kurti sistemai, sudarytos duomenų srautų diagramos, sukurta bendra sistemos architektūra, sudarytos sistemai

aktualių funkcijų algoritmų schemos.

Sudarytos ir pateiktos atakų aptikimo kovariacijos metodu, atakų aptikimo pokyčio taško aptikimo metodu, atakų aptikimo pasyviais skaičaivimais paremtos euristikos metodu, teisėtų ir atakuojančių srautų atskyrimo bei paketo kelio operacinėje sistemoje ir ugniasienėje grafinės schemos.

Projektavimo etape apgalvotas, specifikuotas ir grafiškai pavaizduotas teisėtų ir atakoje dalyvaujančių adresų atskyrimo algoritmas.

Sudarytas sistemos testavimo planas pagal kurį numatoma kurti ir testuoti DDoS atakų aptikimo ir blokavimo sistemą.

3 DDoS atakų aptikimo ir sustabdymo sistemos realizacija ir testavimas

Siekiant išanalizuoti ir įvertinti atakų aptikimo metodus, analizės dalyje pasirinkti realizavimui atakų aptikimo metodai buvo suprogramuoti ir integruoti į vieną testavimo sistemą, remiantis antroje darbo dalyje parenktu sistemos projektu.

Testavimo sistema suprogramuota naudojant PCAP biblioteką ir C programavimo kalbą. Panaudotos universalios paketų antraščių struktūros, todėl kodas susikompiliuoja naudojant įvairių operacinių sistemų kompiliatorius.

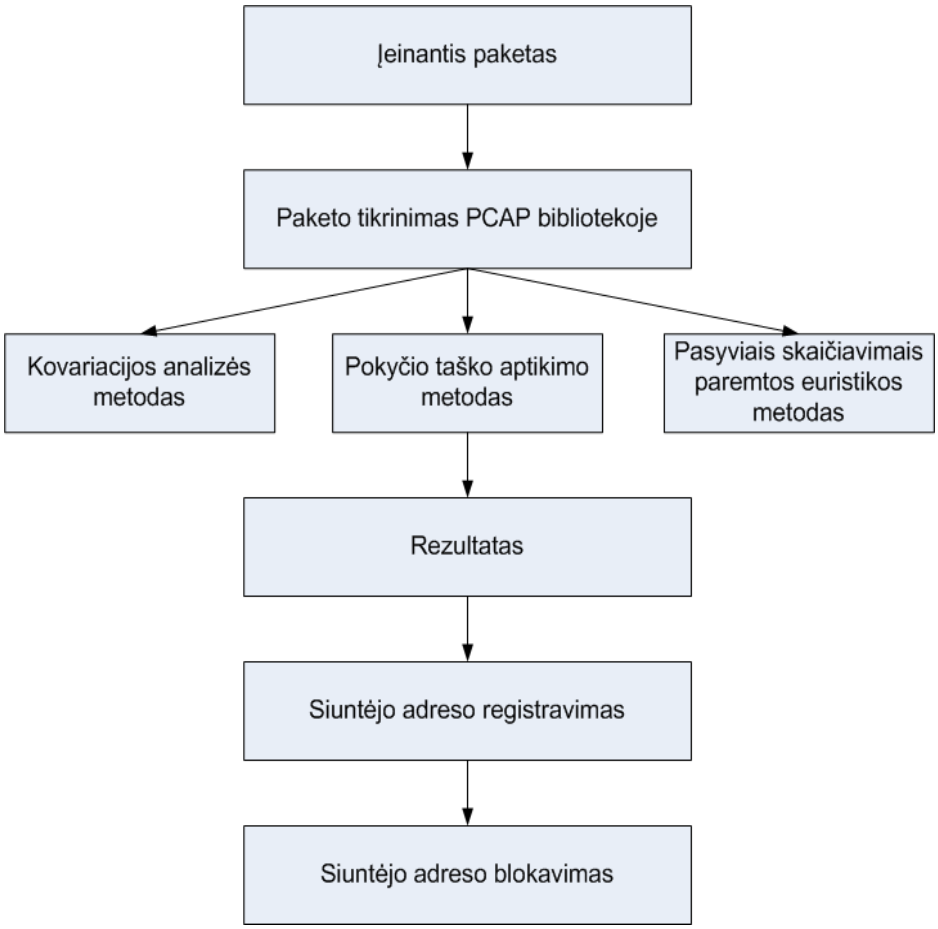
Kiekvieno metodo realizacija parašyta kaip atskiras modelis, nepriklausantis nuo kitų metodų. Kiekvienas paketas gali būti perduodamas bet kuriems iš pasirinktų metodų.

Patikrinus paketą, sistemos kintamieji atnaujinami ir pagal naujas reikšmes priimamas sprendimas, ar skelbti, kad vykdoma ataka, ar ne. Jeigu vyksta ataka, reikia priimti sprendimą, ar blokuoti siuntėjo adresą. Paketo siuntėjo adresas perduodamas adresų atskyrimo moduliui ir nustatoma, ar reikia blokuoti adresą, ar ne. Jei nustatoma, kad adresą reikia blokuoti, kviečiamas srautų blokavimo modulis ir įdedama taisyklė, pagal kurią visi paketai, ateinantys iš siuntėjo adreso yra blokuojami.

Kiekvienas įeinantis paketas apdorojamas operacinės sistemos branduolyje ir perduodamas atitinkamam vartotojo lygmens procesui. Kuomet paketas perduodamas sukurtai sistemai, jis patikrinamas pagal PCAP bibliotekos filtrą ir nustatoma, ar reikia paketą toliau apdoroti atakos aptikimui. Testavimo metu filtras buvo nustatytas praleisti tik įeinančius tcp paketus į 80 prievadą, taip pat visų tipų ICMP paketus. Sukurta sistema tikrina savo vidinę būseną ir vartotojo nustatytus parametrus. Atlikus patikrinimus nusprendžiama, kurį DDoS atakos aptikimo modulį iškviesti. Sistema gali būti nustatyta tikrinti kiekvieną paketą visuose moduliuose.

Ataka gali būti registruojama arba atsižvelgiant tik į vieno metodo rezultatus, arba naudojant visų trijų metodų rezlutatus. PCAP bibliotekos panaudojims taip pat labai palengvina testavimą, nes galima panaudoti filtrus, kurių pagalbą į testuojamą sistemą patenka tik tie paketai, kurie atitinka nustatytus parametrus.

10 pav. pavaizduota realizuotos sistemos grafinė schema.



10 pav. Grafinis sistemos vaizdas

Pagrindinis sukurtos sistemos privalumas yra tas, kad joje yra sujungti keli skirtingi DDoS atakų aptikimo metodai. Kiekvienas metodas turi savo privalumų ir trūkumų apdorodamas tam tikro tipo atakas. Sistema gali būti pritaikyta naudoti labiausiai tinkantį metodą, atsižvelgiant į tai, kokio tipo ataka vykdoma. Sistema taip pat yra lengvai plečiama dėl savo modulinės struktūros.

Dar vienas sistemos privalumas yra tas, kad sistema gali ne tik aptikti ir įspėti apie vykstančias atakas, bet ir užregistruoti atakoje dalyvaujančius adresus, arba netgi blokuoti srautus iš tų adresų, kurie pripažįstami dalyvaujančiais atakoje.

3.1 Sistemos komponentų realizacija

3.1.1 DDoS atakų aptikimo metodai

Sukurtoje sistemoje buvo realizuoti 3 DdoS atakų aptikimo metodai:

- Pokyčio taško aptikimo;
- Kovariacijos;
- Pasyviais skaičiavimais paremtos euristikos.

Kiekvienas metodas realizuotas kaip atskira funkcija, todėl kiekvieną iš jų galima laisvai iškelti į kitas programas. Pagrindinė programa PCAP bibliotekos pagalba priima reikiamą paketą ir pagal nurodytus parametrus perduoda jį apdoroti atitinkantį metodą realizuojančiai funkcijai. Kiekvienas metodas apdorojęs paketą atnaujina savo vidinę būseną ir globalius visos sistemos parametrus.

3.1.2 Teisėtų ir atakoje dalyvaujančių srautų atskyrimas

Teisėtiems ir atakoje dalyvaujanties srautams atskirti pasirinktas kreipimųsi iš kiekvieno adreso registravimo metodas. Sukuriamas masyvas, kuriame saugojama reikšmė, ar buvo kreipimasis iš tikrinamo adreso, ar ne. Šiam tikslui buvo sukurtas masyvas, sudarytas iš vieno baito dydžio elementų. Masyvo įrašų skaičius yra tokio dydžio, kiek skirtingų galimų IP adresų galima užregistruoti. Kadangi IP adreso ilgis yra 32 bitai, o iš viso yra 4294967295 galimos IP adresų kombinacijos, turėtų būti sukuriamas tokio dydžio masyvas. Šiame masyve elemento indeksas atitinka IP adresą, todėl IP adresų registravimas ir tikrinimas vyksta itin greitai, nes nereikia gaišti laiko ieškant adreso tam tikrose duomenų struktūrose, o elementas pasiekiamas pritaikius adresų aritmetikos veiksmą.

Šis metodas savo pilnoj realizacijoj yra labiau teorinis, nes tokio dydžio masyvui saugoti reikia nemažai operatyviosios atminties ir 64 bitų procesoriaus bei 64 bitų platformai sukompiliuotos operacinės sistemos. Šis metodas taip pat negali dirbti su 6 versijos IP adresais.

Realizacijoje ir testavime naudojama 32 bitų operacinė sistema ir kintantys adresai iš fiksuoto potinklio, todėl į pradinę adreso dalį galima neatsižvelgti. Dėl šios priežasties prieš kreipiantis į adresų masyvą iš siuntejo adreso išmetamas pirmasis baitas, nes jis visuose paketuose sutampa. Tokiu būdu galima dirbti su žymiai mažesniu masyvu, kurį gali apdoroti ir 32 bitų operacinės sistemos. Toks masyvo sumažinimas gali būti naudojamas ir realiomis sąlygomis, tačiau reikia atkreipti dėmesį į tai, kad atsiranda galimybė praleisti dalį atakoje dalyvaujančių srautų ir užblokuoti dalį teisėtų srautų.

3.1.3 Paketų blokavimas

Realizuotas paketų blokavimas remiasi teisėtų ar atakoje dalyvaujančių srautų atskyrimo metodo duomenimis. Jei nustatoma, kad nagrinėjamas adresas dalyvauja atakoje, dedama PF ugniasienės taisyklė su parametru, nurodančiu, kad tolimesnis taisyklių apdorojimas pritaikius šą taisyklę neturi būti vykdomas. Taisyklė jungiama prie taisyklių grupės, kuri yra pačioje taisyklių sąrašo pradžioje. Tokiu būdu užtikrinama, kad atakoje dalyvaujančių srautų blokavimas būtų vykdomas pagal pirmas taisykles.

Šiam tikslui naudojamas PF ugniasienių taisyklių adresų lentelės. Lentelių panaudojimo idėja yra tokia, kad į taisyklių sąrašą įterpiama taisyklė, kurioje adresai srautų blokavimui imami iš nurodytos lentelės:

```
table <ddos> {}
block in quick proto tcp from <ddos> to port 80
```

Toliau, pagal oficialią OpenBSD pateikiamą dokumentaciją, adresų sąrašas gali būti pildomas trimis būdais:

- Panaudojant „load rule“ taisyklių konfigūracijos faile;
- Panaudojant pfctl komandą;
- Įrašant adresą į failą.

Pirmasis ir trečiasis būdai reikalauja failo modifikavimo ir viso taisyklių sąrašo atnaujinimo. Pagal šio darbo specifiką toks taisyklių įterpimas gali naudoti daug resursų, nes nuolat reikėtų kreiptis į taisyklių failą jį papildant ir dar kartą nuskaitant taisykles į atmintį. Realizacijoje buvo realizuotas adreso įterpimas į adresų lentelę iškviečiant `pfctl` komandą su atitinkamais parametrais. Tačiau siunčiant 4000 paketų per sekundę, sistema sugebėdavo įtraukti tik apie 200 naujų adresų per sekundę. Vykdant sisteminę komandą iš veikiančio proceso, iš disko skaitomas tos komandos paleidžiamasis failas ir kuriamas naujas procesas. Šiems veiksmams operacinė sistema sugaišta nemažai laiko, todėl naujų adresų įterpimas vyksta lėtai.

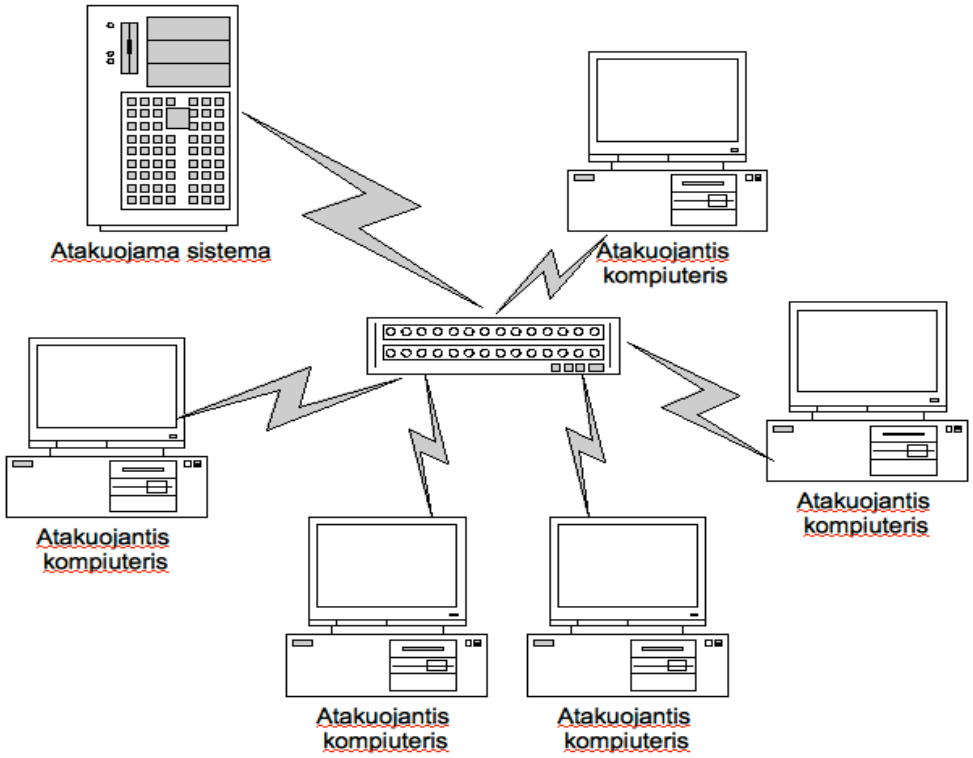
Tokia veikimo sparta nepriimtina atsižvelgiant į projekto dalyje iškeltus reikalavimus sistemos greitaveikai, todėl buvo nuspręsta išanalizuoti `pfctl` įrankio kodą ir tą kodo dalį, kuria naudojantis vykdomas adresų įterpimas į nurodytą lentelę, iškelti ir integruoti tiesiai į kuriamą sistemą. Atlikus šią sistemos modifikaciją bus atliekamas atskiras eksperimentas, siekiant nustatyti, kiek naujų adresų gali įterpti sistema tokiu būdu.

3.2 Sistemos eksperimentiniai tyrimai

3.2.1 Eksperimento aplinka

Darbo metu sukurta sistema buvo testuojama vidiniame tinkle naudojant specialiai testavimui sukurtą paketų generavimo įrankį. Srautų generavimui buvo planuojama naudoti bonesi arba hping2, tačiau šiuose įrankuose nebuvo reikiamo lankstumo išeities adresų atsitiktiniam generavimui bei lanksčiam intervalų tarp siunčiamų paketų valdymui. Buvo nuspręsta testavimui sukurti savo įrankį, turintį visas reikiamas funkcijas.

Paketai buvo generuojami naudojant atsitiktinius išeities adresus ir siunčiami atakuojamam serveriui. Eksperimento aplinkos schema pavaizduota 11 paveikslėlyje.



Pav. 11. Eksperimento tinklo toplogija

Sukurta sistema buvo instaliuota į serverį, turintį 1.8 GHz Intel Pentium 4 procesorių ir 1024 MB operatyviosios atminties. Serveryje įdiegta OpenBSD 4.5 operacinė sistema. PF ugniasienė sukonfigūruota taip, kad praleistų įeinančius ir išeinančius TCP susijungimus į visus testavime naudojamus prievadus.

Prieš pradedant testavimą su DDoS atakomis, sistema tam tikrą laiką buvo testuojama su normaliu apkrovimu. Normaliu apkrovimu buvo laikoma 15 HTTP GET užklausų per sekundę iš 50 skirtingų išeities IP adresų kurie kreipiasi per vieną minutę. Realizuotų metodų reikšmės, atspindinčios normalaus srauto parametrus išsaugotos tolimesniam naudojimui. Visi skaičiavimai atliekami naudojant per paskutines 10 sekundžių surinktus duomenis. Atmintyje saugoma 60 paskutinių intervalų. Tokio testavimo metu sistemoje saugoma paskutinių 10 minučių srauto būsena. Dėl to atakų aptikimas vykdomas paskutinių 10 minučių srauto kontekste, o aptikimo laikas turėtų būti ne daugiau kaip 10 sekundžių. Nominalios metodų reikšmės pateiktos 3 lentelėje.

Lentelė 3. Įprastinio srauto parametrai

Srauto parametrai	Reikšmės	DDoS identification methods		
		Pokyčio taško aptikimas	Kovariacijos metodas	Pasyviais skaičiavimais paremta euristika
http užklausų/įeinančių paketų skaičius per sekundę/klientų skaičius	15/40/50	2512.31	> 0	407
Paketų vidurkiai 10 sekundžių intervalais		350	SYN:350, FIN:350	-

Kovariacijos analizės metodo reikšmės šioje lentelėje nepateikiamos, nes esant įprastinei srauto apkrovai kovariacija tarp pasirinktų srauto parametrų (SYN ir FIN paketų) yra tegiama. Tai yra SYN ir FIN paketų skaičius priklauso tiesiogiai vienas nuo kito.

3.2.2 Sistemos įvertinimo kriterijai

Sistema įvertina kiekvieną metodą atskirai ir juos visus tris veikiančius kartu. Kadangi visi naudojami metodai priklauso nuo iš anksto nustatytų parametrų reikšmių ir nuo to, kas konkrečiomis sąlygomis laikoma normaliu apkrovimu, nuspręsta, kad nėra tikslinga skaičiuoti netikrų įspėjimų apie atakas arba neaptiktų atakų. Toks testavimas turėtų būti atliktas realiomis sąlygomis, pasirenkant nominalias reikšmes metodams. Testavimo metu buvo tiriama, kaip kiekvieno metodo naudojimas apkrauna sistemos procesorių, kintant įeinančių paketų skaičiui. Taip nustatytos metodų parametrų reikšmės testavimo sąlygomis.

Antra eksperimento dalis skirta realiai ištestuoti, kaip veikia teisėtų ir atakoje dalyvaujančių srautų atskyrimas. Įjungus blokavimo funkciją, kurį laiką siunčiami paketai, imituojantys teisėtas užklausas. Vėliau ~~bus~~ pradedama ataka. Srautas iš atakoje dalyvaujančių adresų turėtų būti blokuojamas, o teisėtas užklausas imitavęs srautas turėtų gauti atsakymus. Taip pat tikrinama, kaip galima „apgauti“ šią sistemą, palaipsniui didinant atsiunčiamų paketų skaičių ir bandant patekti į teisėtų srautų sąrašą dar prieš prasidedant atakai.

Trečiojoje eksperimento dalyje testuojami tokie sistemos parametrai:

- Kiek naujų adresų sistema gali įtraukti į blokuojamų adresų sąrašą per sekundę;
- Sistemos apkrova, įtraukiant naujus adresus;
- Užklausų vėlinimas kintant taisyklių skaičiui.

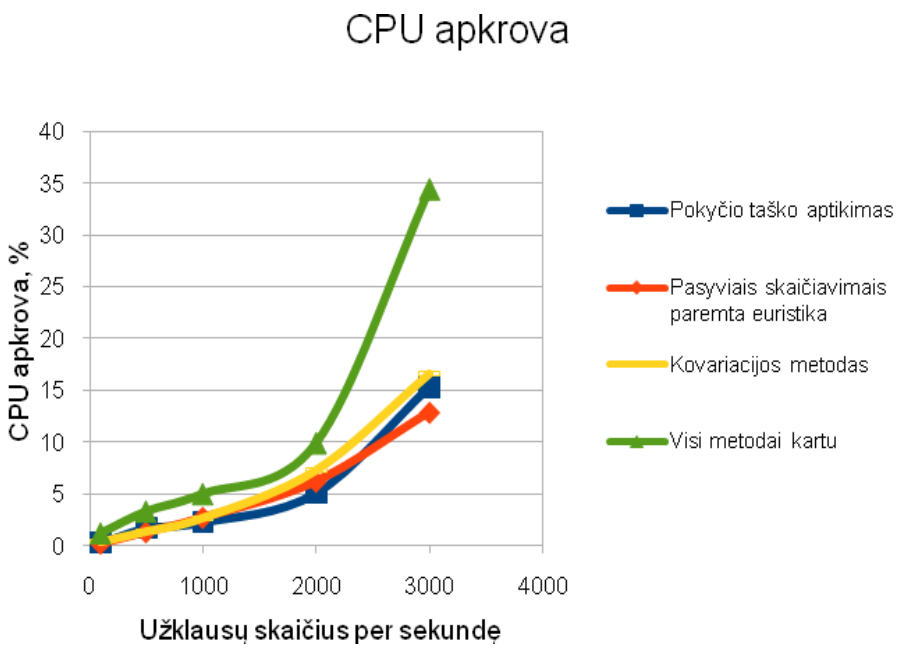
3.2.3 Eksperimento rezultatai

3.2.3.1 DDoS atakų aptikimo metodų įvertinimas

Eksperimentinis sistemos įvertinimas pradėtas maksimalaus įeinančių paketų per sekundę kiekio nustatymu. Buvo tiriama, kiek paketų per sekundę serverio operacinė sistema gali apdoroti nesutrikdydama kitų programų darbo. Eksperimento būdu nustatyta, kad sistema gali apdoroti 3000 užklausų per sekundę, atsakydama į kiekvieną iš jų. Kuomet įeinančių paketų skaičius per sekundę viršyja šią reikšmę, sistemoje susidaro tokia padėtis, kuomet branduolyje nuolat apdorojami paketai, o vartotojo procesai negauna procesoriaus laiko. Procesoriaus apkrovos testavimas tokiomis sąlygomis neturi jokios prasmės.

Sistema buvo testauojama, siunčiant 100, 500, 1000, 2000, 3000 TCP SYN paketų į 80 prievadą. TCP SYN užtvindymo ataka, tai tokia ataka, kuomet aukai siunčiami tik SYN paketai, o atsakymai iš aukos nėra apdorojami. Atakos tikslas – išnaudoti visus sistemos resursus, nes gavus SYN paketą sistema išskiria resursų naujo susijungimo sudarymui ir juos saugo tam tikrą laiką. Testavimo metu joks servisas nebuvo paleistas klausytis susijungimų į 80 prievadą. Tokiu būdu visa apkrova tenka testavimo sistemai, nes operacinė sistema į kiekvieną SYN paketą atsako RST paketu ir papildomų resursų neišnaudoja. Kiekvienas metodas buvo naudojamas atskirai siekiant nustatyti, kokia procesoriaus apkrova susidaro naudojant konkretų metodą. Sistema taip pat buvo testuojama, kuomet naudojami visi trys metodai.

Eksperimento rezultatai pateikti 12 paveikslėlyje.



Pav. 12. CPU išnaudojimas naudojant skirtingus aptikimo metodus

Iš pateiktų rezultatų matyti, kad procesoriaus apkrova ženkliai išauga padidėjus įeinančių paketų skaičiui ir naudojant visus tris aptikimo metodus kartu, tačiau suma yra mažesnė nei suma sistemos apkrovų reikšmių, kuomet naudojamas kiekvienas metodas atskirai. Taip pat matoma, kad naudojant pasikeitimų aptikimo ir kovariacijos metodus, apkrova didėja greičiau, kuomet didėja įeinančių paketų skaičius, o pasyviais skaičiavimais paremtos eurstikos metodas išlaiko pastovų ryšį su įeinančių paketų kiekiu. Tai rodo, kad įeinančių paketų kiekiui artėjant prie ribinio kiekio, kurį gali apdoroti operacinė sistema, sukurtos sistemos pridedama apkrova neviršyja 15% CPU, todėl sistema tinkama naudoti atakų aptikimui. Taip pat iš rezultatų matoma, kad sistemą galima konfigūruoti vykdyti DDoS atakų aptikimą naudojant kelis metodus kartu, siekiant gauti kuo tikslesnius rezultatus.

Tyrimo metu taip pat buvo renkamos metodų reikšmės, esant konkrečiai apkrovai. Gautos reikšmės pateikiamos 4 lentelėje.

4 Lentelė. Apskaičiuotos metodų reikšmės

Metodo pavadinimas	**100 paketų/s**	**500 paketų/s**	**1000 paketų/s**	**2000 paketų/s**	**3000 paketų/s**
Pokyčio taško aptikimas	2110	10354	19811	37806	59757
Pasyviais skaičiavimais paremta euristika	1000	5000	10000	20000	30000

Pokyčio taško aptikimo metodo reikšmės atspindi skirtumus tarp paketų skaičiaus esant normaliai apkrovai, ir paketų skaičiaus kuomet vykdoma ataka. Tetavimo metu nustatytos reikšmės gali būti naudojamos nustatant slenkstines reikšmes. Viršijus slenkstines reikšmes registruojama ataka.

Pasyviais skaičiavimais paremto metodo reikšmės tiesiogiai priklauso nuo eksperimento sąlygų. Kadangi naudojami intervalai po 10 sekundžių, metodo reikšmės gautos 10 kartų didesnės, nei paketų skaičius, siųstas per sekundę testavimo metu. Trumpinant intervalus galima aptikti trumpalaikius įeinančių paketų šuolius, o ilginant intervalus, trumpalaikiai šuoliai išlyginami ir tokiu būdu galima gauti mažiau klaidingų pranešimų apie atakas, tačiau ilgėja laikas nuo atakos pradžios iki jos užregistravimo.

Kovariacijos metodo reikšmės šioje lentelėje nepateikiamos, nes atliekant testavimą buvo stebima tik ar reikšmė teigiama, ar neigiama. Kadangi buvo testuojama tik su SYN užtvindymo ataka, šio metodo reikšmė visada būna neigiama.

3.2.3.2 Teisėtų ir atakoje dalyvaujančių srautų aptikimo įvertinimas

Šio eksperimento metu buvo nustatoma maksimali leistina vieno iš metodų reikšmė, kurią viršijus registruojama, kad vykdoma DDoS ataka. Siunčiant paketus buvo tikrinama, ar bus praleidžiama užklausa, kuri pagal srautų atskyrimo metodo aprašymą turėjo būti pripažinta, kaip teisėta. Eksperimento eiga:

- Eksperimentui buvo pasirinktas pasyviais skaičiavimais paremtos euristikos metodas;
- Parametruose nurodyta, kad ataka turi būti skelbiama, jei paketų skaičius per sekundę yra didesnis nei 500
- Siunčiamas srautas iš įvairių adresų, tačiau įeinančių paketų skaičius per sekundę palaikomas mažesnis nei 500. Kartu su šiuo srautu siunčiamos teisėtos užklausos iš pasirinkto adreso.
- Įeinančių paketų skaičius per sekundę padidintas iki 550
- Tikrinama, ar teisėtos užklausos gauna atsakymą

Įvykdžius visus aukščiau aprašytus testavimo žingsnius, įsitikinta, siunčiant teisėtą užklausą prasidėjus atakai, atsakymas gaunamas, o adresas, iš kurio srautas buvo siunčiamas iki prasidedant atakai yra įtraukiamas į neblokuojamų adresų sąrašą.

Eksperimentas buvo vykdomas 32 bitų operacinėje sistemoje, turinčioje 1024 megabaitus operatyviosios atmintes, todėl visų 4 versijos IP adresų tokioje sistemoje neįmanoma išdėstyti atmintyje. Kadangi eksperimentas buvo vykdomas iš 10.0.0.0/8 potinklio, pirmasis adreso baitas gali būti nenagrinėjamas ir masyvui užtenka naudoti tik 16777216 elementų masyvą išlaikant rezultatų vientisumą. Naudojant tokio dydžio masyvą ir nenagrinėjant pirmojo adreso baito realiomis sąlygomis, adresai gali persidengti (pvz. adresas 87.135.14.182, 197.135.14.182 ir visi kiti adresai *.135.14.182 būtų registruojami tame pačiame elemente). Tokiu būdu atakuojančiam adresui didėja tikimybė būti pripažintam teisėtu.

Metodą taikant realiose sąlygose atitinkamai būtų nustatomos atakų aptikimo parametrų reikšmės, o sistema veikdama ilgesnį laiką surinktų tuos adresus, kurie ilgesnį laiką naudojasi paslauga prieš prasidedant atakai. Vykstant atakai srautas iš tokių adresų būtų priimamas ir taip bent iš dalies būtų užtikrinamas paslaugos prieinamumas teisėtiems vartotojams.

Eksperimento metu nustatyta, kad didinant įeinančių paketų skaičių neviršyjant nustatyto maksimalaus įeinančių paketų kiekio per nustatytą laiko intervalą, sistema nemažą dalį atakuojančių adresų pripažįsta kaip teisėtus ir jau užregistravus ataką, srautai iš šių adresų sėkmingai pasiekia sistemą.

3.2.3.3 Srautų blokavimo tyrimas

Tiriant srautų blokavimą svarbu išsiaiškinti, kaip papildomai naudojami sistemos resursai įdedant naujas taisykles į taisyklių sąrašą. Tiriant srautų blokavimą keičiamas užklausų skaičius per sekundę ir stebimi tokie parametrai:

- Kiek paketų per sekundę sistema gali apdoroti, kuomet įterpiami adresai į blokuojamų adresų sąrašą;
- Kiek pailgėja atsakymas į užklausą didėjant taisyklių skaičiui.

Atsakymo į užklausą laiko kitimo tyrimo eiga:

- Išmatuoti atsakymo į užklausą laiką normaliomis sąlygomis;
- Išmatuoti atsakymo į užklausą laiką vykstant atakai, bet neblokuojant srautų;
- Išmatuoti atsakymo į užklausą laiką vykstant atakai ir blokuojant srautus.

Atliekant tyrimą, kiek paketų sistema gali apdoroti, kuomet įterpiami adresai į blokuojamų srautų sąrašą, buvo siunčiami paketai atakuojamai sistemai, žinant, kiek tiksliai paketų per sekundę išsiunčia atakuojantys kompiuteriai kartu sudėjus. Sistemoje buvo matuojama, kiek paketų sistema gali apdoroti taikant visus tris DDoS atakų aptikimo metodus ir atliekant srautų blokavimą.

Sistemoje apdorojamų paketų kiekiai per sekundę, kintant įeinančių paketų skaičiui pateikiami 3 lentelėje:

3 Lentelė. Sistemoje apdorojamų paketų kiekiai

Šiunčiama paketų	**100 paketų/s**	**500 paketų/s**	**1000 paketų/s**	**2000 paketų/s**	**3000 paketų/s**
Apdorojama paketų vykdant blokavimą	100	500	1000	1939	2915
Apdorojama paketų nevykdant blokavimo	100	500	1000	2000	2952

Pagal lentelėje pateiktus rezultatus matome, kad įeinančių paketų skaičiui artėjant link paketų kiekio, kurį operacinė sistema gali apdoroti, ribos, sistemai nebeužtenka laiko apdoroti visus paketus ir sudėti adresus į taisyklių sąrašą, todėl apdorojamų paketų skaičius šiek tiek sumažėja.

Atliekant atsakymo į užklausą tyrimą buvo siunčiamos užklausos sistemai visais viršuje nurodytais atvejais ir matuojamas atsakymo į užklausą laikas. Tyrimas buvo vykdomas siunčiant 4000 TCP SYN paketų į 80 prievadą. Atsakymas į teisėtą užklausą buvo matuojamas siunčiant 100 teisėtų užklausų naudojant hping2 įrankį ir imant atsakymų laikų vidurkį. Nevykstant atakai vidutinis atsako į užklausą laikas yra 1 milisekundė.

Rezultatai pateikiami 4 lentelėje:

4 **Lentelė. Atsako į teisėtas užklausas laikai**

Šiunčiama paketų per sekundę	**100**	**500**	**1000**	**2000**	**3000**
Atsako laikas blokuojant srautus (ms)	1.2	1.6	1.9	2.9	3.7
Atsako laikas neblokuojant srautų (ms)	1.1	1.3	1.4	1.8	2.5

Pagal lentelėje pateiktus rezultatus matome, kad atsakymo į užklausas vėlinimas išauga ženkliai, kuomet įeinančių paketų kiekis artėja prie operacinės sistemos ribinio apdorojamų paketų kiekio. Taip pat matome, kad naudojant srautų blokavimą sistemos resursai apkraunami daugiau, įtraukiant naujus adresus į blokuojamų adresų sąrašą, todėl atsakymo į užklausą laikas taip pat pailgėja.

3.3 Skyriaus apibendrinimas

Atlikus eksperimentus nustatyta, kad visi trys darbe analizuoti ir realizuoti DDoS atakų aptikimo metodai gali efektyviai aptikti užtvindymo atakas, tačiau nei vienas iš metodų nėra skirtas atskirti teisėtus srautus nuo srautų, dalyvaujančių atakoje. Nei vienas iš metodų taip pat neturi galimybės sekti ar registruoti atakoje dalyvaujančių mazgų IP adresų.

Naudojant kovariacijos metodą galima aptikti pakitimus ryšyje tarp pasirinktų dviejų srauto parametrų. Norint šį metodą išnaudoti dar efektyviau reikia atlikti daugiau stebėjimų ir išsiaiškinti ryšius tarp įvairių srauto parametrų vykdant įvairias atakas. Nustačius ryšius, metodas gali būti pritaikyas stebėti daugiau parametrų ir aptikti įvairių rūšių atakas.

CPU apkrova naudojant bet kurį iš DDOS atakų aptikimo metodų yra maža, kuomet užklausų skaičius per sekundę neviršyja 1000. Apkrova ženkliai dideja, kuomet užklausų skaičius per sekundę yra didesnis nei 2000. Pasikeitimų aptikimais paremtas metodas veikia geriausiai, kuomet užklausų skaičius per sekundę yra intervale nuo 700 iki 2500. Kitomis sąlygomis pasyviais skaičiavimais paremtos euristikos metodas veikia efektyviau. CPU apkrova naudojant visus tris metodus kartu išauga sparčiai, didėjant užklausų skaičiui, tačiau apkrova yra mažesnė, nei visų trijų metodų naudojamų atskirai. Taip yra todėl, kad visi paketo priėmimo ir apdorojimo veiksmai

sistemoje atliekami nepriklausomai nuo to, koks metodas naudojamas.

Teisėtų ir atakoje dalyvaujančių srautų atskyrimo testavime patvirtinta, kad metodas veikia praktiškai ir atakos metu praleidžia srautus iš tų adresų, iš kurių buvo gauti paketai dar prieš prasidedant atakai. Taip pat nustatyta, kad metodą galima naudoti skiriant jam mažiau operatyviosios atminties, tačiau sumažinant tikslumą. Nustatyta, kad metodas veikia tiksliai tais atvejais, kuomet atakuojančių adresų kiekis auga greitai, arba atakoje dalyvauja visai nauji adresai, iš kurių srauto seniau nebuvo siunčiama.

Atliekant srautų blokavimo tyrimą ištirta:

- Tiriant, kiek paketų per sekundę sistema gali apdoroti, kintant įeinančių paketų kiekiui, nustatyta, kad prie mažesnių nei maksimalių operacinės sistemos apdorojamų apkrovų sistema sugeba apdoroti visus paketus. Nevykdant srautų blokavimo sistema nespėja apdoroti visų paketų tik tada, kai įeinančių paketų per sekundę kiekis artėja prie maksimalaus operacinės sistemos apdorojamų paketų skaičiaus per sekundę.
- Tiriant srautų blokavimo metu atsirandantį papildomą sistemos resursų apkrovimą, buvo stebima, kaip pailgėja teisėtų užklausų apdorojimo laikas, kuomet įterpinėjami nauji adresai į blokuojamų adresų sąrašą. Nustatyta, kad įeinančių paketų kiekiui pasiekus operacinės sistemos apdorojamų paketų ribą, atsako laikas į užklausą yra ilgesnis daugiau nei sekunde, negu nevykdant srautų blokavimo.
- Buvo ištirta ir nustatyta, kad įeinančių paketų kiekiui artejant link operacinės sistemos maksimalios apdorojamų paketų ribos, užklausos laikas pradeda ilgėti greičiau, nei esant mažoms sistemos apkrovoms. Taip pat nustatyta, kad taisyklių įterpimo metu sistemos resursai apkraunami daugiau ir tai taip pat įtakoja užklausos apdorojimo laiką.

Išvados

1. Atlikus egzistuojančių DDoS aptikimo metodų analizę nustatyta, kad metodai gali būti realizuoti paskirstytu būdu arba vienoje sistemoje, yra skirti tam tikro tipo atakų aptikimui ir yra efektyvūs esant tik tam tikromis sąlygomis. Nei vienas iš nagrinėtų DDoS atakų aptikimo metodų neturi galimybės atskirti teisėtus srautus nuo srautų, dalyvaujančių atakoje. bei sekti ar registruoti atakoje dalyvaujančių mazgų IP adresų.
2. Išnagrinėti DDoS atakų sustabdymo būdai, leidžiantys tinkamai atlikti nepageidaujamo didelės apimties srauto blokavimą. Nustatyta, kad norint efektyviai blokuoti nepageidaujamą srautą reikalinga turėti efektyviai veikiančias ugniasienes bei DDoS atakų aptikimo sistemas.
3. Išskirti pagrindiniai metodų efektyvumą nusakantys kriterijai. Pagal šiuos kriterijus nuspręsta plačiau tirti ir realizuoti pokyčio taško aptikimo, kovariacijos ir pasyviais skaičiavimais paremtos euristikos DDoS aptikimo metodus.
4. Aptartas OpenBSD operacinėje sistemoje naudojamos ugniasienės PF veikimo principas bei išnagrinėti ugniasienių taisyklių išdėstymo atmintyje metodai, leidžiantys pagreitinti duomenų srauto apdorojimą.
5. Suprojektuota ir realizuota DDoS atakų aptikimo ir blokavimo sistema DDoS atakų aptikimui naudojanti pokyčio taško aptikimo, kovariacijos ir pasyviais skaičiavimais paremtos euristikos DDoS atakų aptikimo metodus, o paketų iš atakoje dalyvaujančių mazgų atskyrimui ir blokavimui naudojantį autoriaus pasiūlytą teisėtų ir atakoje dalyvaujančių adresų atskyrimo algoritmą. Suprojektuotoje sistemoje panaudota optimizuota PF ugniasienė, leidžianti vykdyti teisėtus srautus praleidžiančių ir atakas blokuojančių taisyklių patikrinimą anksčiau negu visas kitas atmintyje esančios ugniasienės taisykles.
6. Sistemos įvertinimo etape nustatyta, kad sistema gali aptikti DDoS atakas naudodama bet kurį iš trijų realizuotų DDoS atakų aptikimo metodų ir sunaudoja nedidelę dalį sistemos resursų. Nustatyta, kad resursų išnaudojimas išauga tik tada, kai įeinančių paketų skaičius artėja prie maksimalaus paketų skaičiaus, kurį operacinė sistema gali apdoroti su turimais resursais.
7. Teisėtų ir atakoje dalyvaujančių srautų atskyrimo eksperimento metu nustatyta, kad pasiūlytas ir realizuotas teisėtų ir atakoje dalyvaujančių adresų atskyrimo algoritmas, leidžia atskirti atakoje dalyvaujančius ir teisėtus srautus bei sumažinti blokuojamų teisėtų paketų skaičių. Testuojant galimas algoritmo silpnąsias vietas nustatyta, kad jei paketai iš atakos

šaltinių pasiekia sistemą prieš užregistruojant DDoS atakos pradžią, paketai iš tų srautų praleidžiami ir toliau traktuojami kaip teisėti srautai. Metodas veikia tiksliai tais atvejais, kuomet atakuojančių adresų kiekis auga greitai, arba atakoje dalyvauja visai nauji adresai, iš kurių srauto seniau nebuvo siunčiama.

8. Tiriant suprojektuotos sistemos srautų blokavimo savybes, nustatyta, kad atsako į užklausą laikas ilgėja blokuojamų adresų įdėjimo į taisyklių sąrašą metu. Taip pat nustatyta, kad įeinančių paketų kiekiui artėjant prie operacinės sistemos apdorojamų paketų kiekio per sekundę ribos, sistema nebesugeba apdoroti visų įeinančių paketų.
9. Visi atlikti praktiniai tyrimai rodo, kad realizuota sistema atitinka iškeltus greitaveikos reikalavimus, o jos sukeliama papildoma apkrova sistemos resursams leidžia sistemai funkcionuoti net ir vykstant atakai. Kai kurie sistemos apribojimai gali būti pašalinti naudojant daugiau resursų, o dėl nesudėtingos architektūros gali būti įdiegiami nauji DDoS atakų aptikimo metodai.
10. Atlikti pokyčio taško aptikimo, kovariacijos ir pasyviais skaičiavimais paremtos euristikos DDoS aptikimo metodų CPU išnaudojimo praktiniai tyrimai rodo, kad pokyčio taško aptikimo metodas veikia geriausiai, kuomet užklausų skaičius per sekundę yra intervale nuo 700 iki 2500. Kitomis sąlygomis pasyviais skaičiavimais paremtos euristikos metodas veikia efektyviau. CPU apkrova naudojant visus tris metodus kartu išauga sparčiai, didėjant užklausų skaičiui, tačiau apkrova yra mažesnė, nei visų trijų metodų naudojamų atskirai. Taip yra todėl, kad visi paketo priėmimo ir apdorojimo veiksmai sistemoje atliekami nepriklausomai nuo to, koks metodas naudojamas.

Literatūros sąrašas

[1] **P. E. Ayres, H. Sun, H. J. Chao, Fellow, W. C. Lau**. ALPi: A DDoS Defense System for High-Speed Networks, *IEEE Journal on Selected Areas in Communications*, October 2006, Vol. 24, No. 10, pp. 1864–1876.

[2] **K.Park, H.Lee.** On the Effectiveness of Route-Based Packet Filtering for Distributed DDoS Attack in Power-Law Internets, *SIGCOMM Comput. Commun*. October 2001, Rev. 31, No. 4, pp. 15-26.

[3] **S. Jin D. S. Yeung.** A Covariance Analysis Model for DDoS Attack Detection, *IEEE International Communication Conference (ICC04)*, June 2004, Vol. 4, pp. 20-24.

[4] **K. K. K. Wang, R. K. C. Chang.** Engineering of a global defense infrastructure for DDoS attacks, *Proceedings of IEEE International Conference on Networking*: *ICON2002*, August 2002, pp. 419-427.

[5] **Y. Xiang, Y. Lin, W.L. Lei, S.J. Huang.** Detecting DDOS attack based on network self-similarity, *Communications, IEE Proceedings*, June 2004, Vol. 151, Issue: 3, pp. 292-295.

[6] **C. Siaterlis, B. Maglaris.** Detecting DDoS attacks with passive measurement based heuristics, *Proceedings of the Ninth International Symposium on Computers and Communications 2004*, 2004, Vol. 2, pp. 339-344.

[7] **L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred.** Statistical Approaches to DDoS Attack Detection and Response, *Proceedings of DARPA Information Survivability Conference and Exposition*, 2003, Vol. 1, pp. 303-314.

[8] **Y. Chen, K. Hwang, W. S. Ku.** Distributed Change-Point Detection of DDoS Attacks: Experimental Results on DETER Testbed, *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007*, August 2007, pp.7-7.

[9] **M. Lee, E. J. Kim, C. W. Lee.** A Source Identification Scheme against DDoS Attacks in Cluster Interconnects, *Proceedings of the 2004 International Conference on Parallel Processing Workshops*, August 2004, pp. 354 – 361.

[10] **A. Akella, A. Bharambe, M. Reiter, S. Seshan.** Detecting DDoS Attacks on ISP Networks, *ACM SIGMOD/PODS Workshop on management and processing of data streams (MPDS) FCRC*, 2003, P. 3.

[11] **R. K. C. Chang.** Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, *IEEE In Communications Magazine*, 2002, Vol. 40, No. 10, pp. 42-51.

[12] http://code.google.com/p/bonesi/

[13] **Hartmeier, D.** Design and Performance of the OpenBSD Stateful Packet Filter (pf), 2002

[14] ftp://ftp3.usa.openbsd.org/pub/OpenBSD/doc/pf-faq.txt

[15] http://www.linuxsecurity.com/content/view/121960/171/

Priedas Nr. 1

Analysis and Evaluation of Distributed Denial of Service Attacks identification methods

Saulius Grusnys[1], Ingrida Lagzdinyte[2]

[1]Kaunas University of Technology, Computer Networks Department, Studentu 50, Kaunas, Lithuania, saulius.grusnys@stud.ktu.lt

[2]Kaunas University of Technology, Computer Networks Department, Studentu 50, Kaunas, Lithuania, ingrida.lagzdinyte@ktu.lt

Abstract. Defending against Distributed Denial of Service (DDoS) attacks is one of the most important tasks to ensure service availability. At the same time it is one of the most challenging tasks because it requires complex and efficient methods to correctly identify and stop such kind of attacks. There are number of methods available to identify DDoS attacks. Some of the methods are based on single packet or connection; others evaluate packets according to all the traffic available at particular time. There is a need to identify what method or methods should be used under particular circumstances.

In this paper we present a software system, which implements some of the available methods to detect DDoS attacks and creates firewall rules to stop the traffic from the hosts suspected to be participating in the attack. Implemented methods include Change Point Approach, Covariance model and Passive Measurement based Heuristics.

The system enables to analyze characteristics of implemented DDoS identification methods and evaluate their efficiency in different conditions.

Keywords: Distributed Denial of Service, DDoS, DDoS identification methods.

Introduction

Distributed Denial of Service Attacks

Distributed Denial of Service (DDoS) is the kind of attacks that are performed in order to interrupt Internet services by flooding the victim with a high volume of malicious packets originating from many different sources [1]. There are two general types of DDoS attacks classified: a) direct attacks; b) reflector attacks [2].

In case of direct attacks all the packets are sent from attacking hosts directly to victim. Most of the time spoofed IP addresses are used, making the attack more effective as the victim tries to repeat reply packets to non-existing hosts.

Using the reflector attacks, requests with spoofed victim's source address are sent to lots of different servers. Victim gets flooded by the replies coming from those servers. It is very hard to stop such attacks as the traffic comes from legitimate servers which may be needed to provide services on the victim side.

Due to the fact that a DDoS attack has to be detected on-line, the detection of an attack should be as quick as possible in order to prevent attack from the very beginning [3]. The detection of a DDoS attack is also very complicated by the fact that it is very similar to the traffic generated by increased number of legitimate users. Another problem with DDoS attacks is that the equipment of the victim or victim's ISP can be taken down by the amount of traffic generated by attacking hosts. No methods for detecting an attack are effective in such case.

The detection of the attack can be more effective if it is performed on ISP routers as close to the attackers as possible. Each node takes the part of the attack load and, if detected effectively, blocks it.

DDoS Identification Methods

There are number of methods available which aim to detect DDoS attacks. Figure 1 presents the classification of available DDoS identification methods.

There first group of methods (see Figure 1 (a)) propose measures to stop DDoS attacks in global scale [2, 4]. The idea of such methods is that the number of systems detecting DDoS attacks is located in many different places on the Internet. When the attack is suspected, it is communicated with other system whether the traffic is suspicious to them or not. If the attack is confirmed, routers are instructed to block packets from attacking hosts [4].

The first advantage is that systems using those methods can get more information about the environment communicating with each other. Another one is that the amount of traffic that each system must process is smaller.

Disadvantage of such methods is that implementing them requires more cost. Besides it is more complicated to add traffic analyzing systems in various places of the network.

Another group of methods (see Figure 1 (b)) is available for detecting DDoS attacks on the victim side [1, 3, 5-9]. Most of these methods rely on monitoring traffic for some period of time when the traffic load is normal and no attacks are performed. Normal traffic profiles are created examining different traffic parameters in different ways.

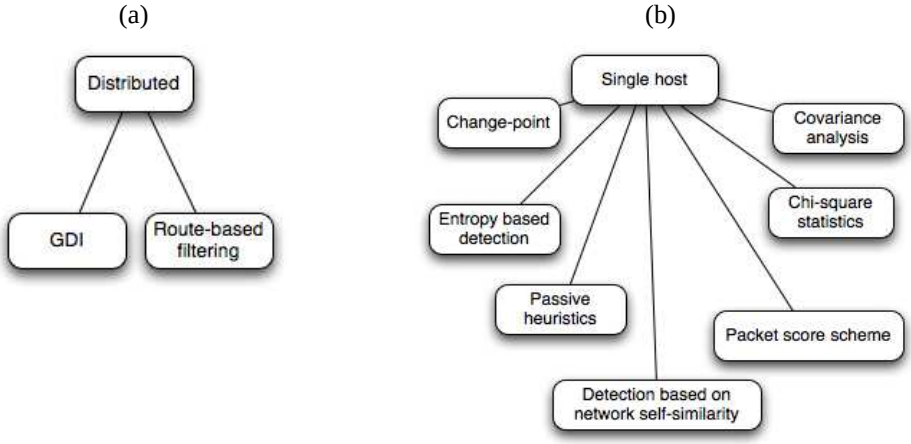


Figure 1. DDoS Identification Methods Classification. (a) Distributed methods, (b) Single host methods

When attack starts the difference between parameter values in profiles reflecting normal traffic and values reflecting current traffic occurs. In that way the system knows that the attack has been launched.

The main advantage of such methods is that they can be implemented and run on one host. The second advantage is that the threshold values used by the methods can be tuned according to the particular traffic that flows through monitoring host. The main disadvantage of such methods is that one system can easily be flooded by great amount of traffic coming from many different hosts.

Further in this paper we will focus on detecting DDoS attacks on a single host, saying that ISP and our hardware are able to cope with the load generated by the attack.

As single host DDoS detection methods operate differently, evaluate different parameters and are suitable to detect different kinds of DDoS attacks there is a need to identify what method or methods should be used under particular circumstances. In this article three methods will be evaluated: Change Point Approach, Covariance Model and Passive Measurement based Heuristics. We think that these methods can be effective and easily implemented in hardware level if necessary**.**

The rest of the paper is structured as follows: in the Section 2 the main characteristics of Change Point Approach, Covariance model and Passive Measurement based Heuristics methods are discussed. In Section 3 we describe the software that implements these methods and is used for their evaluation. Section 4 presents some experimental results. Finally, in the Section 5, the conclusions are made.

Change Point Approach, Covariance model and Passive Measurement based Heuristics methods

Change Point Approach, Covariance Model and Passive Measurement based Heuristics methods are presented and fully described in [3, 6, 8]. Here we will discuss only the most important aspects of these three methods.

Change Point Approach

Using the Change Point Detection algorithm, the number of packets arrived during certain period of time is measured. Then cumulative sum is calculated for predefined number of intervals. Drastic changes in cumulative sum values mean that the change in the state of the traffic has occurred [8]. In such way the beginning of the attack can be identified quickly and effectively even if the number of legitimate packets arriving has been high before starting the attack.

The algorithm used to calculate cumulative sum for any parameter can be illustrated by 5 steps: 1) define number of intervals and the length of the interval; 2) count the number of packets matching particular feature during each interval; 3) calculate average of packets in all the intervals; 4) starting with 0 count cumulative sum for all other intervals by adding the value of the earlier interval with the value of current interval and subtracting the average; 5) the change is called if the differences in cumulative sums between intervals exceed the maximum defined value.

The main drawback of this method is that the attack cannot be detected if the volume of traffic is increasing steadily while not triggering the maximum difference between cumulative sum values. The difference of the values in cumulative sum that triggers the attack alarm should be identified by monitoring the legitimate traffic for some period of time and by setting the maximum difference between cumulative sum values.

Covariance model

Covariance method used for identifying DDoS attacks is based on calculating covariance between two parameters in arriving packets [3]. When attack starts the covariance value should differ from the value calculated during normal load. Covariance between two parameters in N measurements can be calculated by expression (1):

$$cov\,xy = \frac{1}{N}\sum x_i y_i - \overline{xy}\,, \qquad (1)$$

where: N – is the number of measurements; x_i - is the value of the first variable in i-th measurement; y_i - is the value of the second variable in i-th measurement, i ranging from 1 to N;

This method is effective in detecting TCP SYN flooding attacks, as the number of packets containing SYN flags set and the number of packets containing FIN flag set becomes different and not related to each other. The tuning of this method involves identifying the pairs of parameters to monitor and the normal values for the selected pairs. To be able to detect other types of attacks, the relations between various traffic parameters should be identified. However this is not always possible, meaning that this method cannot be used in detecting all kinds of flooding attacks.

Passive Measurement based Heuristics

There is also a way to detect DDoS attacks by using heuristics [6]. This simple method calculates the number of packets received during predefined interval of time. During some period of time the average and maximum of packets received is determined. If the number of packets received exceeds the predefined maximum, flooding attack is detected.

The main advantage of this method is the short period of time needed to identify the attack. Quicker identification results in quicker reaction. However this method can produce a number of false positives if there are short high traffic peaks.

Software for DDoS Identification Methods Analysis and Evaluation

In order to analyze and evaluate Change Point Approach, Covariance Model and Passive Measurement based Heuristics methods they were implemented and integrated into one testing system.

The testing system was programmed using PCAP library interface and C programming language. The structures representing protocol headers were used from Linux include files so for now the system can only be compiled in Linux operating system. Each method was written as a separate module. Each of the function is called passing every received packet as a parameter. After the evaluations are made the internal variables of the system are updated and the decision is made whether to call an attack or not. Figure 2 illustrates the main architecture of implemented system.

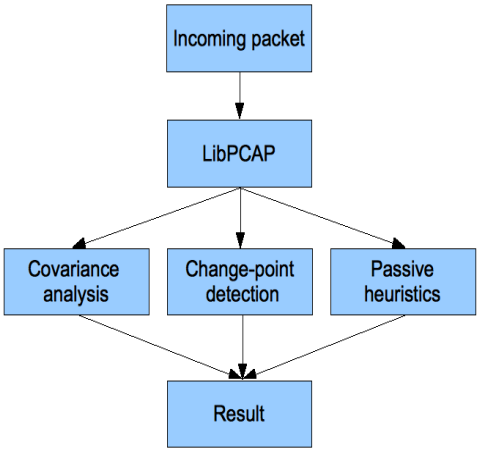


Figure 2. The graphical view of the system

Every incoming packet is processed by Linux kernel. After that the packet is passed to appropriate user level processes. Once the packet is passed to the process of our implemented system it gets evaluated against the LibPCAP filter to determine whether the packet should be further processed in order to detect an attack. Currently the filters are set to only allow packets arriving to TCP 80'th port and all types of ICMP packets. Our system then checks its internal state as well as the parameters set by the user and decides which DDoS identification module should be called. The system can be set to evaluate each packet using all available methods to get more reliable results.

Attack can be registered either using the results of one of the methods, or using the results of all three methods. Using PCAP library is also helpful in testing since it is much easier to set filters for incoming packets and work with the particular packets that are needed for evaluation.

The main strength of the developed system is that it combines a number of different DDoS identification methods. Each method has it's own strengths and weaknesses working with particular kind of attacks. The system can be adjusted to use the most suitable attack identification method according to the type of the attack being detected. The system is also made easily extendible due to its modular structure.

Performance measurement

Experiment environment

The system is tested in local area network using bonesi DDoS Botnet simulator [12]. Packets are generated using random source IP addresses in packets sent to the victim host. The experimental environment is illustrated in Figure 3.

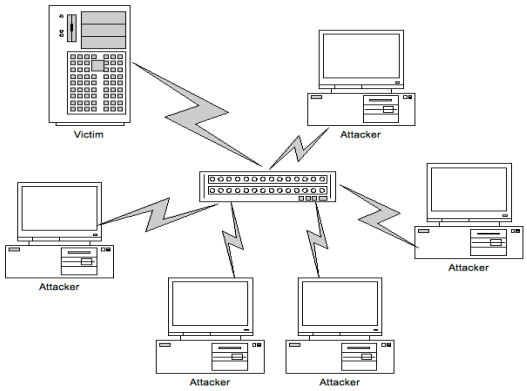


Figure 3. An experimental network topology

The server where the implemented system was installed has 1.8 GHZ Intel Pentium 4 CPU and 512 MB RAM. It runs on Linux 2.6.29.4 kernel. IPTABLES firewall is configured to allow all incoming and outgoing TCP traffic to and from the ports used in testing.

The system is run for some time under normal load. Normal load is considered 15 HTTP GET requests per second, using 50 different source IP addresses during one minute interval. Values reflecting normal load are recorded for future use. All the calculations are made using data collected in 10 seconds intervals. There are 60 intervals. As a result the system is checking its state in last 10 minutes, being able to detect attack in less than 10 seconds. The nominal values are presented in Table 1.

Table 1. Parameters of normal traffic

Traffic parameters	Value	DDoS identification methods		
		Change Point Approach	Covariance model	Passive Measurement based Heuristics
http requests/incoming packets per second/number of clients	15/40/50	2512.31	> 0	407
Packet averages in 10 seconds interval		350	SYN:350, FIN:350	-

Performance Evaluation Metrics

Our system evaluates each method separately and all of them as one. Since all the methods used depends on preset values for normal traffic we think there is no need to test the number of false positives or false negatives, since it would not reflect the actual environment where the system will be implemented. The methods will always give an alert if preset values are exceeded or not matched. Instead we will test the performance of each method by measuring CPU usage with different number of packets arriving. We will also determine the values for each method against those counts.

Experimental Results

The testing was started by identifying the maximum number of packets per second that operating system can handle. The experiments showed that it can process 3000 requests per second by replying to every one of them. Sending more requests brings operating system into state when kernel processes and user processes are not given any CPU time. In such case testing the CPU usage of the system is worthless.

The system was tested with 100, 500, 1000, 2000 and 3000 TCP SYN packets incoming to 80 port. TCP SYN flooding attack is a kind of attack when only the SYN packet is sent and no replies are expected from victim. The

purpose of attack is to run target system out of resources. No service was listening on 80 port, so all the load has been taken by the testing system. Each method has been launched separately to identify the CPU usage of the system. The system was also tested with all three methods activated. The results are provided in Figure 4.

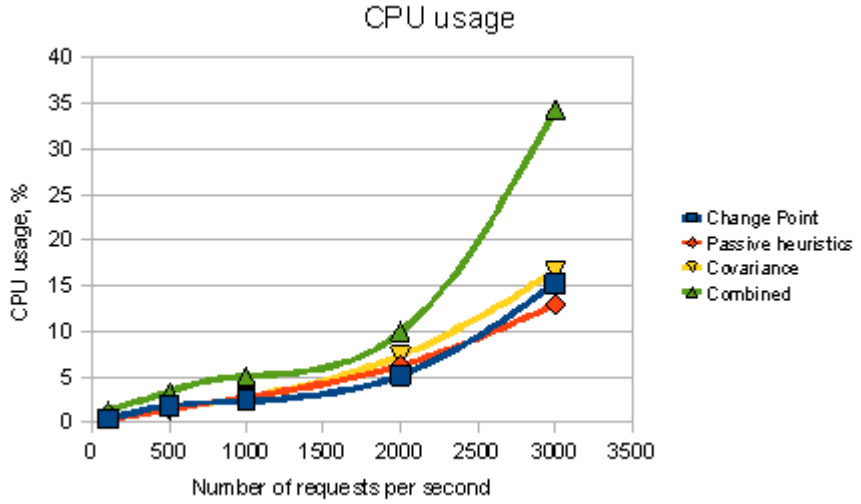


Figure 4. CPU usage in every DDoS identification method's case

As it can be seen in the chart, CPU usage of the system when all three methods are active rises significantly under higher loads. However it is slightly less then the sum of usages of all three methods separately. We can also see that CPU usage using Change-Point and Covariance methods grow faster when number of packets increases, while method based on Passive Heuristics maintains constant relation with the number of packets being processed.

During the testing values of the methods under the certain load were calculated. The values are presented in Table 2.

Table 2. Calculated values of Change Point and Passive Measurement based Heuristics methods

Method name	**100 packets/s**	**500 packets/s**	**1000 packets/s**	**2000 packets/s**	**3000 packets/s**
Change point	2110	10354	19811	37806	59757
Passive heuristics	1000	5000	10000	20000	30000

The values of Change Point method reflect the difference between the number of packets under normal load and the number of packets when the attack is launched. The values identified during the testing may be used to set the threshold value to detect increase of the number of packets. If we set the threshold to 10000, the attack can be called when system receives more then 500 packets per second.

The values of passive heuristics method directly depends on the values used in testing. Since we use the interval of 10 seconds in testing, the values we get are number of packets per second multiplied by 10. As it was discussed in previous chapter, the performance of this method is very high, but the values are static and do not depend on any traffic context. The values in the table can also be used as thresholds to detect when the peaks of traffic are exceeded. Choosing longer interval would produce approximate values and reduce the risk of getting false positives due to very short bursts of incoming traffic.

Values of Covariance method are not provided since in our testing they were either positive or negative and since the testing was made with SYN flooding attack, the value was always negative.

Conclusions

After running the tests we can see that all three methods can effectively detect flooding attacks. However these methods are not intended to distinct legitimate packets from the flow. They are also unable to track or log the IP addresses of the hosts participating in the attack.

The Covariance method is able to detect changes between relations in selected traffic parameters. To get the most from using this method, more observations should be performed in relations between various traffic parameters. Having the relations identified, the method can be adopted to monitor more parameters, making it possible to detect various kinds of flooding attacks.

Change-Point approach is able to detect the exact moment of increase or decrease of traffic, which provides quick detection of an attack. Using this method, the changes in values should be identified when the traffic gets lower than nominal traffic, because the change would also be registered and that can lead to a false positive.

Using Passive Heuristics method, attacks can be detected as soon as the last packet exceeding the threshold arrives. However that may produce a number of false positives in cases of short peaks in traffic.

The CPU usage of every DDoS identification method is low when number of requests does not exceed 1000 requests per second. It greatly increases when number of requests per second is greater than 2000. Change Point method produces the best CPU performance when number of requests per second is in interval from 700 from 2500. In other conditions Passive Heuristics method is more effective.

CPU usage of the system when all three methods are active rises significantly under higher loads. However it is slightly less then the sum of usages of all three methods separately.

Future work

According to the tests performed we can see that the system developed is still in a need of improvement. First of all the ability to distinct legitimate traffic from malicious traffic during the attack is necessary. After having such ability, the system would be able to block malicious hosts when the attack is detected.

The system can also be extended by implementing other available DDoS detection methods and by improving the currently implemented ones.

References

[1] **P. E. Ayres, H. Sun, H. J. Chao, Fellow, W. C. Lau.** ALPi: A DDoS Defense System for High-Speed Networks, *IEEE Journal on Selected Areas in Communications*, October 2006, Vol. 24, No. 10, pp. 1864–1876.

[2] **K.Park, H.Lee.** On the Effectiveness of Route-Based Packet Filtering for Distributed DDoS Attack in Power-Law Internets, *SIGCOMM Comput. Commun.,* October 2001, Rev. 31, No. 4, pp. 15-26.

[3] **S. Jin D. S. Yeung.** A Covariance Analysis Model for DDoS Attack Detection, *IEEE International Communication Conference (ICC04)*, June 2004, Vol. 4, pp. 20-24.

[4] **K. K. K. Wang, R. K. C. Chang.** Engineering of a global defense infrastructure for DDoS attacks, *Proceedings of IEEE International Conference on Networking: ICON2002*, August 2002, pp. 419-427.

[5] **Y. Xiang, Y. Lin, W.L. Lei, S.J. Huang.** Detecting DDOS attack based on network self-similarity, *Communications, IEE Proceedings*, June 2004, Vol. 151, Issue: 3, pp. 292-295.

[6] **C. Siaterlis, B. Maglaris.** Detecting DDoS attacks with passive measurement based heuristics, *Proceedings of the Ninth International Symposium on Computers and Communications 2004*, 2004, Vol. 2, pp. 339-344.

[7] **L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred.** Statistical Approaches to DDoS Attack Detection and Response, *Proceedings of DARPA Information Survivability Conference and Exposition*, 2003, Vol. 1, pp. 303-314.

[8] **Y. Chen, K. Hwang, W. S. Ku.** Distributed Change-Point Detection of DDoS Attacks: Experimental Results on DETER Testbed, *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007*, August 2007, pp.7-7.

[9] **M. Lee, E. J. Kim, C. W. Lee.** A Source Identification Scheme against DDoS Attacks in Cluster Interconnects, *Proceedings of the 2004 International Conference on Parallel Processing Workshops*, August 2004, pp. 354 – 361.

[10] **A. Akella, A. Bharambe, M. Reiter, S. Seshan.** Detecting DDoS Attacks on ISP Networks, *ACM SIGMOD/PODS Workshop on management and processing of data streams (MPDS) FCRC*, 2003, P. 3.

[11] **R. K. C. Chang.** Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, *IEEE In Communications Magazine*, 2002, Vol. 40, No. 10, pp. 42-51.

[12] http://code.google.com/p/bonesi/