KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Andrius Kulbis

PASKIRSTYTO PRIEVADŲ SKENAVIMO ATAKŲ APTIKIMAS TINKLO SRAUTUOSE

Baigiamasis magistro darbas

Vadovas
Doc. dr. Rimantas Kavaliūnas

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

TVIRTINU
Katedros vedėjas
(parašas)Prof. dr. Algimantas Venčkauskas
(data)

PASKIRSTYTO PRIEVADŲ SKENAVIMO ATAKŲ APTIKIMAS TINKLO SRAUTUOSE

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

Vadovas
(parašas) Doc. dr. Rimantas Kavaliūnas
(data)

Recenzentas
(parašas) Doc. dr. Gytis Vilutis
(data)

Projektą atliko
(parašas) Andrius Kulbis
(data)

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS

(Fakultetas)

(Studento vardas, pavardė)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Paskirstyto prievadų skenavimo atakų aptikimas tinklo srautuose“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ m. _____ ____ d.

Kaunas

Patvirtinu, kad mano **Andriaus Kulbio** baigiamasis projektas tema „Paskirstyto prievadų skenavimo atakų aptikimas tinklo srautuose“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

_____ _____

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

Kulbis, A. Paskirstyto prievadų skenavimo atakų aptikimas tinklo srautuose. Magistro baigiamasis projektas / vadovas doc. dr. Rimantas Kavaliūnas; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Kaunas, 2015. 42 psl.

SANTRAUKA

Kompiuterinės bei ryšio technologijos labai palengvina bendravimą bei bendradarbiavimą, tačiau kartu jos atneša ir vis daugiau naujų grėsmių tiek pavieniams vartotojams namuose, tiek įmonių ar įstaigų darbuotojams.

Šiame magistriniame darbe nagrinėjami kompiuterių tinklo prievadų skenavimo tipai ir vykdymo metodikos, įrankiai ir literatūroje aprašomi metodai, šioms atakoms tinklo duomenų srautuose aptikti. Tolesniam tyrimui buvo pasirinktos paskirstyto prievadų skenavimo atakos, kuomet aibė šaltinių atlieka skenavimą nukreiptą prieš aibę įrenginių, veikiančių tinkle.

Buvo sudarytas metodas šio tipo atakų aptikimui, kuris remiasi plačiai naudojamo vieno šaltinio atliekamo skenavimo aptikimo algoritmu ieškant pėdsakų NetFlow tinklo srautų įrašuose, tam kad sumažinti, informacijos, kurią reikia analizuoti kiekį ir aprėpti visame tinkle vykstančius įvykius.

Atlikus eksperimentą nustatyta, kad siūlomas metodas efektyviau naudoja tinklo srauto analizei naudojamo įrenginio resursus, bei tiksliau aptinka ir identifikuoja paskirstytas tinklo prievadų skenavimo atakas lyginant su Snort įsilaužimų aptikimo sistema.

Kulbis, A. Distributed port scan detection in network traffic. Master thesis / research supervisor Assoc. Prof. Dr. Rimantas Kavaliūnas; Department of Computer Science, Faculty of Informatics, Kaunas University of Technology.

Kaunas, 2015. 42 p.

SUMMARY

Computer and communications technology greatly facilitates communication and cooperation, and together they bring more and more new threats to both individual home users and corporate bodies or employees.

This paper focuses on network port scanning types and techniques and detection methods for this kind of network attacks. Further investigation has been distributed for the selected port scanning attack, where multiple sources are scanning multiple host in order to find security gaps or weak spots in the network.

A method for detecting this type of attack, which is based on the widely used single source scan detection algorithm using NetFlow network flow records as a data source, in order to reduce the amount of the information you need to analyze and coverage events taking place throughout the entire network.

The experiment showed that the proposed method is more efficient with the use of device resources for network traffic analysis and has a better accuracy for detecting and identifying a distributed network port scanning attacks compared with the Snort intrusion detection system.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

IPT – Interneto paslaugų tiekėjas.

TCP *(Transmission Control Protocol)* – tai vienas iš pagrindinių protokolų, esančių Internetinių protokolų rinkinyje.

IP *(Internet protocol address)* – kompiuterio identifikatorius IP tinkluose.

ĮVADAS

Informacinės technologijos įgauna vis didesnę reikšmę žmonių kasdieniniame gyvenime. Kompiuterinės bei ryšio technologijos labai palengvina bendravimą bei bendradarbiavimą, tačiau kartu jos atneša ir vis daugiau naujų grėsmių tiek pavieniams vartotojams namuose, tiek įmonių ar įstaigų darbuotojams. Saugumas - tęstinas veikimo būdas, o ne vienkartinis rezultatas. Saugumui užtikrinti turi būti nuolat naudojamos administracinės, techninės bei programinės priemonės. Kadangi kasdien vis daugiau įrenginių yra prijungiama prie pasaulinio kompiuterių tinklo, piktavališkų kėslų turintiems asmenims atsiranda daugiau taikinių kenkėjiškiems veiksmams atlikti. Dėl šios priežasties būtina pasirūpinti tinklo saugumu ir stebėjimu.

Darbo problematika ir aktualumas

Piktavalis, prieš atlikdamas ataką nukreiptą į kompiuterių tinklą dažniausiai atlieką kokios nors formos informacijos rinkimą stengdamasis nustatyti silpnąsias taikinio ar tinklo įrenginio vietas [33]. Vienas iš galimų informacijos rinkimo būdų yra tinklo prievadų skenavimas. Skenavimo metu yra siunčiamos prisijungimo užklausos į prievadus įrenginyje ir remiantis gautu atsakymu daromos išvados apie įrenginyje veikiančias paslaugas, operacinę sistemą ir pan. Panjawi nustatė, kad maždaug po 50 % visų tinkle vykusių skenavimų į tinklą buvo nukreipta ataka [42]. Aptikus prievadų skenavimą ir turint omenyje, kad tai gali būti žvalgyba prieš planuojamą ataką, tinklo administratoriai gauna šansą nustatyti, kokiomis silpnosiomis vietomis gali būti pasinaudota. Taip pat, būti budriems, žinant, kad gali įvykti ataka nukreipta į tinklą ar įrenginį.

Piktavaliai, atsižvelgdami į didėjantį prievadų skenavimo atakų aptikimo metodų spektrą ir efektyvumą turi rasti naujų būdų, kaip kuo labiau užmaskuoti savo atliekamus veiksmus nuo aptikimo.

Vienas iš tokių metodų yra paskirstytas tinklo prievadų skenavimas. Pasinaudojus šiuo metodu, tinklo skenavimo procesas yra padalinamas į N dalių ir šios dalys paskirstomos aibei šaltinių S, kurie gali atlikti skenavimą. Šie šaltiniai gali būti atakos organizatoriaus valdomi įrenginiai arba kompiuteriai – zombiai, užkrėsti žalinga programine įranga, kuriuos galima valdyti nuotoliniu būdu. Taip pat gali būti naudojamas IP adresų klastojimas organizatoriaus tapatybei nuslėpti. Taikinio pusėje tokia ataka bus matoma kaip prisijungimo užklausos ateinančios iš daugybės skirtingų šaltinių. Jei kiekvienam šaltiniui bus paskirtas nedidelis kiekis taikinių, kuriuos reikia skenuoti, paskirstytas prievadų skenavimas gali likti visiškai nepastebėtas tinklo sraute. Iš to, kiek pastangų piktavalis įdeda tam, kad liktų nepastebėtas, galime spręsti apie jo aukštą kvalifikuotumą ir didesnę grėsmę, keliamą tinklui [38].

Šio magistrinio darbo sritis yra kompiuterių tinklai. Darbo objektas – paskirstytos kompiuterių tinklo prievadų skenavimo atakos ir metodai jų aptikimui. Yra pasiūlyta nemažai metodų tokių

atakų aptikimui naudojant gilią tinklu perduodamų paketų analizę [11][30][33][35]. Tačiau praktiškai jie beveik nenaudojami, nes tokiai analizei realiu laiku reikalingi labai dideli kompiuteriniai resursai.

Šiame darbe nagrinėjami egzistuojantys prievadų skenavimo tipai, jų atlikimo bei aptikimo metodai ir įrankiai ir siūlomas mažiau kompiuterinių resursų reikalaujantis paskirstyto prievadų skenavimo aptikimo metodas, kuris vietoj gilios tinklu perduodamų paketų analizės naudoja maršrutizatorių registruojamus išvestinius duomenis apie duomenų srautus.

Darbo tikslas ir uždaviniai

Magistrinio darbo tikslas – pasiūlyti paskirstyto prievadų skenavimo atakų aptikimo metodą naudojant NetFlow įrašus apie tinklu perduodamus srautus ir ištirti pasiūlyto metodo efektyvumą. Šiam tikslui pasiekti buvo iškelti tokie uždaviniai:

- Išanalizuoti tinklo prievadų skenavimo atakų ypatybes;
- Išanalizuoti tinklo stebėjimo, skenavimo aptikimo metodų ir įrankių privalumus ir trūkumus;
- Sudaryti paskirstyto prievadų skenavimo atakų aptikimo metodą remiantis gerosiomis egzistuojančių aptikimo metodų praktikomis;
- Eksperimentiškai palyginti sudaryto metodo galimybes ir efektyvumą su pasirinktu klasikiniu prievadų skenavimo aptikimo įrankiu.

Darbo rezultatai ir jų svarba

Darbo metu pasiūlytas ir eksperimentiškai ištirtas paskirstyto prievadų skenavimo atakų aptikimo metodas, palyginant jį su klasikinių įrankių tinklo saugumo ir įsilaužimų aptikimo srityje Snort. Pasiūlytas metodas aptinka šias atakas greičiau ir tiksliau, naudojant mažiau įrenginio resursų

Darbo struktūra

Pirmame šio darbo skyriuje yra apžvelgiami tinklo prievadų skenavimo tipai bei jų atlikimo metodikos. Taip pat aprašomi ir palyginami tinklo stebėjimo metodai, įrankiai prievadų skenavimo aptikimui bei literatūroje rasti paskirstyto skenavimo aptikimo metodai ir jų trūkumai. Antrame skyriuje pateikiamas sudaryto metodo pasirinkto tipo paskirstyto prievadų skenavimo atakų aptikimui aprašymas. Trečiame skyriuje pateikti eksperimento, atlikto siekiant patikrinti sudaryto metodo efektyvumui aprašymas bei rezultatai. Ketvirtame skyriuje pateikiamos šio baigiamojo darbo rezultatus apibendrinančios išvados.

1. PROBLEMINĖS SRITIES ANALIZĖ

Tinklo prievadų skenavimas yra ataka, kurios metu siunčiamos užklausos į vieną ar daugiau įrenginio tinklo prievadų. Šiais veiksmais siekiama nustatyti atvirus prievadus ir pasinaudoti žinomomis saugumo spragomis paslaugose, kurios naudojasi aptiktais atvirais prievadais duomenų priėmimui [1]. Kartais šią metodiką naudoja sistemos administratoriai tinklo tyrimui, saugumo spragų tinkle paieškai, tačiau dažniausiai prievadų skenavimą atlieka piktavaliai vartotojai ieškantys silpnųjų vietų tinkle [2]. Prievadų skenavimas taip pat gali būti atliekamas norint užversti įsilaužimo aptikimo sistemas klaidingais pranešimais ir taip atitraukti sistemos administratoriaus dėmesį [11].

Aplikacijos, esančios skirtinguose įrenginiuose komunikuoja tarpusavyje per internetą nurodydamos duomenų priėmimui paskirto prievado numerį. Standartiškai yra nustatyti 65536 TCP ir 65536 UDP [3] prievadai paslaugų tarpusavio komunikavimui, kurie yra suskirstyti į tris grupes [4]:

- Gerai žinomi prievadai (0 – 1023)
- Užregistruoti prievadai (1023 – 49151)
- Dinaminiai ir privatūs prievadai (49152 – 65535)

Paprastai prievadų skenavimas nepadaro jokios tiesioginės žalos, tačiau padeda piktavaliui aptikti silpnąsias vietas, į kurias gali būti nukreiptos atakos ateityje. Prievadų skenavimas susideda iš užklausos į prievadą siuntimo ir atsakymo iš su juo susietos aplikacijos gavimo. Remiantis atsakymu galima spręsti ar prievadas yra atviras ir naudojamas [5]. Skenavimas dažniausiai vykdomas į TCP prievadus, kadangi vykdant informacijos perdavimą, įrenginiai susijungia tiesiogiai ir išlaiko šį susijungimą viso informacijos perdavimo metu, taip suteikdami daug grįžtamojo ryšio informacijos [6].

Šio darbo metu bus nagrinėjamas TCP prievadų skenavimas.

1.1. Tinklo prievadų skenavimo tipai

TCP prievadų skenavimas pagrįstas TCP sujungimo tarp dviejų tinklo mazgų proceso šalutiniais efektais, kurie išgaunami naudojant nestandartinę sujungimo prašymo procedūrą ar netipinius TCP vėliavėlių rinkinius. Egzistuoja daug įvairių prievadų skenavimo tipų, tačiau galima išskirti keletą pagrindinių, dažniausiai sutinkamų, leidžiančių greitai ir nepastebimai (išskyrus *„TCP Connect“)* atlikti prievadų skenavimą [7][19]:

- ***„TCP Connect“*** **skenavimas** – pats paprasčiausias skenavimo tipas, kurio metu yra siunčiamas paketas su SYN vėliavėle, gaunamas atsakymas yra paketas su SYN/ACK vėliavėlėmis pranešantis apie susijungimo leidimą ar patvirtinimą, kurį piktavalis užbaigia paketu su ACK vėliavėle. Šis skenavimo tipas lengvai aptinkamas ir filtruojamas, kadangi sistemos – taikinio įvykių žurnaluose fiksuojami prisijungimo ir paslaugų, kurios priėmė susijungimą klaidų pranešimai. Šis prievadų skenavimo tipas yra laikomas greičiausiu [8].

- ***„TCP SYN“* skenavimas** – dar vadinamas „pusiau – atviru“ skenavimu, kadangi nėra sudaromas pilnas TCP susijungimas [8]. Skenavimo metu piktavalis siunčia paketą su nustatyta SYN vėliavėle į taikinį ir laukia atsakymo. Jei prievadas yra atviras, taikinys gražina paketą su SYN/ACK vėliavėlėmis. Jei prievadas nėra atviras, taikinys gražina paketą su RST vėliavėle. Jei nesulaukiama jokio atsako iš prievado, gali būti, kad jis yra apsaugotas ugniasienės. Šio tipo skenavimai sunkiau aptinkami, kadangi dažniausiai nebūna tinkamai sukonfigūruotas sistemos – taikinio įvykių žurnalo vedimas. Tai yra pats populiariausias prievadų skenavimo tipas [2][11].
- ***„TCP FIN“* skenavimas –** šio tipo prievadų skenavimas dažniausiai lieka nepastebėtas ugniasienių, paketų filtrų ir skenavimą aptinkančių programų. Atakuojanti sistema siunčia paketą su FIN vėliavėle į taikinį. Neaktyvūs prievadai į šį paketą atsako paketu su RST vėliavėle, tuo tarpu aktyvūs prievadai ignoruoja paketą. Piktavalis tokiu būdu gali nustatyti prievadų būsenas: kurie atsakė – yra neaktyvūs, kurie neatsakė – yra atviri [7].
- ***„TCP ACK“* skenavimas –** šio tipo prievadų skenavimas nėra skirtas nustatyti ar prievadas yra aktyvūs ar neaktyvus. Jis yra naudojamas išsiaiškinti, kurie prievadai yra apsaugoti ugniasienės taisyklių. ACK skenavimo paketas turi tik ACK vėliavėlę. Skenuojant tiek aktyvūs, tiek neaktyvūs prievadai atsako paketu su RST vėliavėle. Iš to galima spręsti, kad jie nėra filtruojami ugniasienės taisyklių. Prievadai, kurie negrąžina atsakymo yra filtruojami [9].
- ***„TCP XMAS“, „TCP NULL“* skenavimas –** šio tipo skenavimai savo scenarijumi ekvivalentūs *„TCP FIN“* skenavimui, tačiau skiriasi nustatytos TCP vėliavėlės. XMAS skenavimo metu yra nustatomos FIN, PSH ir URG vėliavėlės, tuo tarpu NULL skenavimo metu nėra nustatomos jokios vėliavėlės.

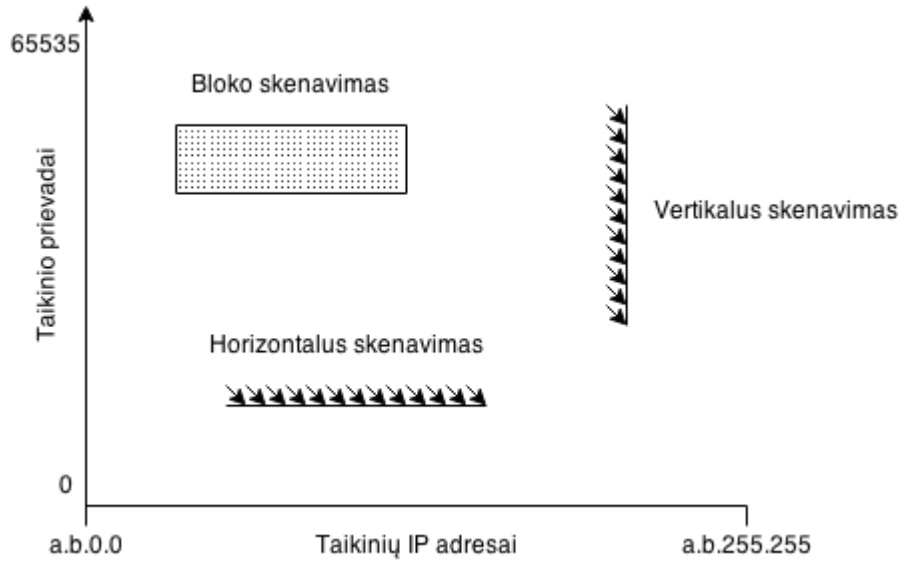
Remiantis surinkta informacija atlikus prievadų skenavimą, įsilaužėlis gali planuoti tolesnius veiksmus:

- įvertinti ruošiamos atakos perspektyvumą,
- pasirinkti atakai tinkamą objektą,
- parinkti tolesnį atakos kelią ar pan.

Šiame darbe sudaryto paskirstyto prievadų skenavimo atakų aptikimo metodas skirtas aptikti *„TCP SYN“* prievadų skenavimą. Kaip parodė [2] tyrimas, net 75 % prievadų skenavimo atakų yra būtent šio tipo.

1.2. Tinklo prievadų skenavimo metodai

Tinklo prievadų skenavimo metodai gali būti suskirstyti į tris kategorijas, pagal tai, kokius IP adresus ir prievadus skenavimas apima [10].



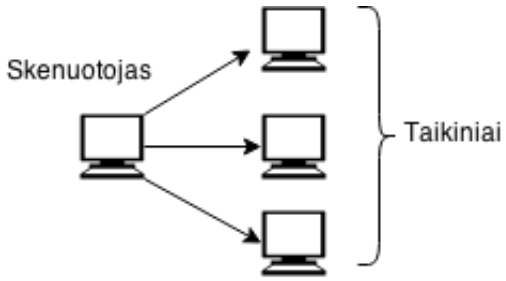
1 pav. Prievadų skenavimo metodikų schema [36]

- **Vertikalus prievadų skenavimas** – prievadų skenavimas, kurio metu yra siunčiamos užklausos į aibę prievadų esančių viename taikinyje (serveryje ar kompiuteryje). Vertikalus prievadų skenavimas būdingas piktavaliams, renkantiems informaciją apie konkretų taikinį. Šis metodas yra lengviausiai aptinkamas, kadangi aptikimo mechanizmas yra reikalingas tik konkrečiame įrenginyje, kuris yra skenuojamas [5].
- **Horizontalus prievadų skenavimas –** prievadų skenavimas, kurio metu yra siunčiamos užklausos į tą patį prievadą. Skirtingai nei vertikalaus skenavimo atveju, šio tipo atakos taikiniais tampa aibė įrenginių, ieškant juose konkrečios spragos, žinomos piktavaliui, kuria pasižymi programinė įranga besiklausanti skenuojamo prievado [36]. Tai viena iš dažniausiai sutinkamų prievadų skenavimo metodikų [11].
- **Blokinis prievadų skenavimas –** prievadų skenavimas, kurio metu yra apjungiamas dvi anksčiau minėtos metodologijos. Atliekant blokinį prievadų skenavimą užklausos siunčiamos į aibę prievadų esančių aibėje įrenginių. Šiuo metodu galima gauti sąrašą galimų taikinių pažeidžiamumu išnaudojimui.

Remiantis tuo, kaip yra atliekamas prievadų skenavimas, galima išskirti dvi didesnes skenavimo praktikų grupes [5]: atliekamas iš vieno šaltinio ir paskirstytas prievadų skenavimas.

1.2.1. Vieno šaltinio atliekamas prievadų skenavimas

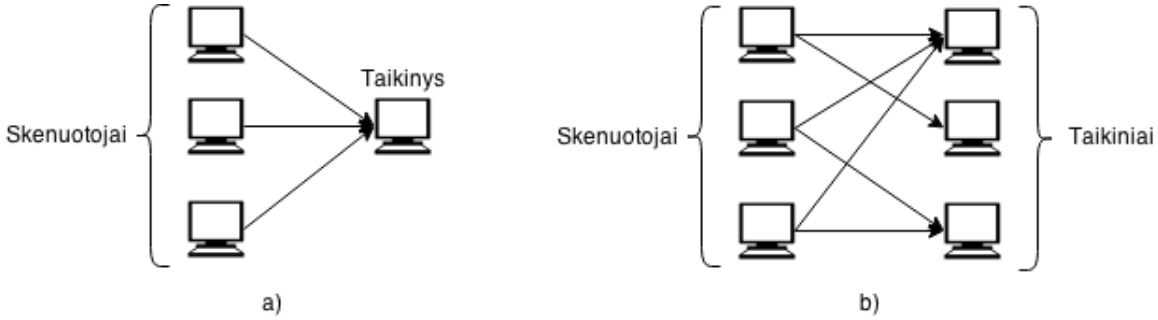
Skenavimą atlieka piktavalis naudodamasis vienu įrenginiu, kurį galima identifikuoti vienu IP adresu. Skenavimas atliekamas taikant vieną iš anksčiau apžvelgtų trijų skenavimo metodikų: vertikalų, horizontalų arba bloko skenavimą.



2 pav. Vieno šaltinio atliekamo skenavimo schema

1.2.2. Paskirstytas prievadų skenavimas

[26] pateikia sąvoką „paskirstytas informacijos rinkimas“. Straipsnyje pažymima, kad toks informacijos rinkimo būdas yra atliekamas naudojantis *„daug su vienu“* (3 pav. A) arba *„daug su daug“* (3 pav. B) modeliu, siekiant surinkti informaciją apie įrenginį ar tinklą, į kurį nusitaikyta. Piktavalis panaudoja keletą šaltinių informacijos rinkimui, paskirstant jiems užduotis atsitiktiniu, netiesiniu būdu. Šiuo atveju netiesiškumas pasireiškia IP adresų ir prievadų porų atsitiktinių paskirstymu.



3 pav. Paskirstyto skenavimo schema

[27] apibūdina paskirstytą ataką, kaip kelių pakopų veiksmus naudojantis lygiagrečiomis sesijomis, kur veiksmų paskirstymas tarp sesijų užmaskuoja vieningą atakos prigimtį arba padeda ją atlikti greičiau. [28] paskirstytą ataką apibūdina kaip aibę IP adresų siekiančių vieno tikslo.

Puikus paskirstytos tinklo atakos pavyzdys yra paskirstytas prievadų skenavimas. [29] apibrėžia paskirstytą prievadų skenavimo ataką kaip atliekamą kelių šaltinių ir nutaikytą į prievadų aibę įrenginiuose, priklausančiuose tam pačiam potinkliui (/24) vienos valandos bėgyje. Gates [33] pasiūlė paskirstytą skenavimą apibrėžti taip: aibė prievadų skenavimų iš skirtingų šaltinių, kuriuos koordinuoja vienas atakos organizatorius. [30] IP adresai kurie atlieka skenavimą ir yra arti vienas kito adresų erdvėje yra traktuojami, kaip dalyvaujantys paskirstytame skenavime. Ši prielaida daroma siekiant aptikti paskirstytą skenavimą, kuris yra atliekamas piktavalio, turinčio aibę IP adresų iš vieno IPT.

Visi skyrelyje paminėti paskirstyto skenavimo apibrėžimai išskiria keletą pagrindinių charakteristikų, kuriomis pasižymi šio tipo atakos:

- Bendras tikslas

- Užduočių paskirstymas tarp skirtingų šaltinių tikslui pasiekti

Šiame darbe bus naudojamas paskirstytos atakos apibrėžimas, suformuluotas remiantis [33] ir [30]: aibė skenavimų, kuriuos atlieka skirtingi šaltiniai, valdomi vieno organizatoriaus ir turintys IP adresus, esančius arti vienas kito adresų erdvėje.

1.3. Tinklo srauto stebėjimo metodai

Tinklo stebėjimas – gebėjimas kaupti ir analizuoti duomenis apie tinklo srautus [25]. Išskiriami du pagrindiniai stebėjimo principai:

Aktyvus – naudojamas kokybinių tinklo charakteristikų nustatymui, pasitelkiant nesudėtingą programinę įrangą (*ping, traceroute, netperf*). Šio metodo trūkumas yra matavimo metu siunčiami šalutiniai duomenys [25].

Pasyvus – suteikia išsamią informaciją apie tinklo srautus. Stebėjimui atlikti naudojama specializuota tinklo įranga arba prie tinklo prijungtas pasiklausymo įrenginys, stebėjimo metu nėra siunčiama šalutinė informacija, galinti sutrikdyti tinklo veiklą [25].

Šiame darbe keliamiems uždaviniams tinkami tik pasyvūs tinklo stebėjimo metodai, nedarantys įtakos tinklo srautui.

1.3.1. NetFlow

Netflow srautas yra apibūdinamas kaip vienkrypčių paketų aibė, siunčiama to paties siuntėjo tam pačiam gavėjui. Siuntėjas ir gavėjas identifikuojami pagal tinklo sluoksnio IP adresus ir transporto sluoksnio siuntėjo ir gavėjo prievadų adresus. Visą paketų srautą NetFlow registruoja vienu įrašų, kurio struktūra parodyta lentelėje 1. Skirtingi tinklo paketai priklauso tam pačiam srautui, jei sutampa septyni raktiniai srauto laukai. Lentelėje 1 jie yra paryškinti.

Lentelė 1. NetFlow paketo struktūra [25]

Turinys	*Ilgis*	*Paaiškinimas*
srcaddr	**4**	**Siuntėjo IP adresas**
dstaddr	**4**	**Gavėjo IP adresas**
nexthop	4	Tolimesnio maršrutizatoriaus IP adresas
input	**2**	**Priimančios sąsajos SNMP indeksas**
output	2	Išsiunčiančios sąsajos SNMP indeksas
dPkts	4	Paketų skaičius sraute
dOctets	4	Oktetų (baitų) skaičius sraute
first	4	Sistemos laikas srauto pradžioje
last	4	Sistemos laikas srauto pabaigoje
srcport	**2**	**Siuntėjo prievado numeris**

Turinys	*Ilgis*	*Paaiškinimas*
dstport	**2**	**Gavėjo prievado numeris, arba ICMP tipas ir kodas**
pad1	1	Nenaudojama, užpildoma nuliais
tcp_flags	1	TCP vėliavėlių suma panaudojant loginį „arba“.
prot	**1**	**Protokolo numeris**
tos	**1**	**IP paketo TOS reikšmė**
src_as	2	Siuntėjo autonominės sistemos numeris
dst_as	2	Gavėjo autonominės sistemos numeris
src_mask	1	Siuntėjo tinklo kaukės ilgis
dst_mask	1	Gavėjo tinklo kaukės ilgis
pad2	2	Nenaudojama, užpildoma nuliais

Naujas NetFlow įrašas sukuriamas maršrutizatoriui gavus pirmąjį neatitinkantį jau registruotiems srautams paketą. Netflow įrašas taip pat turi papildomų laukų, saugančių naudingą informaciją: paketų ir baitų skaičius sraute, srauto pradžios ir pabaigos laiko žymės, maršruto informacija – kito šuolio adresas, šaltinio adresas.

Maršrutizavimo įrenginys tikrina NetFlow įrašų saugyklą kartą per sekundę ir pažymį srautą pasibaigusiu jei:

- Persiuntimas yra baigtas (gauta TCP FIN arba RST vėliavėlė);
- Saugykla užsipildė;
- Srautas yra nebeaktyvus bent 15 sekundžių;
- Srautas tęsiasi jau 30 minučių;

Pasibaigusių srautų įrašai yra sugrupuojami į NetFlow eksportavimo duomenų sekas, kurias maršrutizatorius periodiškai siunčia surinkėjams [23].

NetFlow eksportavimo mechanizmas leidžia eksportuoti tik reikalingus srauto įrašo laukus, todėl yra sumažinamas saugojimo vietos poreikis Siunčiant tik reikalingą informaciją taip pat sumažinamas tinklo apkrovimas [24]. Mažesnis informacijos kiekis, kurį reikia saugoti, leidžia sukaupti ilgesnio periodo tinklo srautų informaciją, bei sumažinti reikalingų skaičiavimo resursų poreikį jos analizei.

NetFlow tinklo srautų informacija gali būti panaudota [37]:

- Identifikuoti aplikacijas ir protokolus, kurie naudoja tinklo srautą ir panaudoti šia informaciją problemoms tinkle spręsti;
- Nustatyti, kada tinkle vyko incidentai ir kokią įtaką tinklo darbui jie turėjo;
- Aptikti tinklo veikimo sutrikimus sukeliančias priežastis stebint anomalijas duomenų srautų charakteristikose;

- Pateikti duomenis valstybės įgaliotoms institucijoms apie tam tikrų tinklo mazgų inicijuotus sujungimus pagal Elektroninių ryšių įstatymą [40].

1.3.2. SNMP

SNMP – bendra tinklo stebėjimo kalba, integruota į didžiąją dalį tinklo įrenginių. Tai standartinis programinio lygio protokolas, kuris leidžia valdymo stočiai – programinei įrangai, kuri surenka SNMP informaciją – užklausti agentus, veikiančius tinklo įrenginiuose apie jų būsenos kintamuosius [17]. Kokia informaciją pateiks agentai, priklauso nuo įrenginio tipo. Jei agentas veikia serveryje, jis gali pranešti apie serverio procesoriaus ir atminties panaudojimą. Jei agentas veikia maršrutizatoriuje, galima sužinoti sąsajų būseną ir panaudojimą, perspėjimus apie perkrovą, maršrutizatoriaus techninius rodmenis – ar veikia aušintuvas [18].

Visi SNMP palaikantys įrenginiai savyje turi specifinį failą – valdymo informacijos bazę (MIB). MIB apibrėžia kokią informaciją galima gauti iš įrenginio, o SNMP yra protokolas skirtas tai informacijai gauti. SNMP suteikia galimybe tinklo administratoriams valdyti skirtingų gamintojų įrenginius, naudojantis vienu įrankiu [17].

Nors protokolas teikia puikią, apibendrintą statistiką, tačiau juo negalima gauti detalizuotos informacijos, reikalingos išsiaiškinti daugumos tinkle kylančių problemų. Pavyzdžiui, nors SNMP gali pranešti apie didelį apkrovimą maršrutizatoriaus interneto sąsajoje, jis negali detalizuoti, koks srautas naudoja sąsajos pralaidumą arba kas yra srauto šaltinis. Todėl naudojantis SNMP galima nustatyti problemos tinkle atsiradimą, bet negalima sužinoti tos problemos priežasties [18].

1.3.3. LIBPCAP

LIBPCAP yra paketų stebėjimui naudojama biblioteka, kuri suteikia aukšto lygio sąsają su paketų stebėjimo mechanizmais ir leidžia pasiekti visus paketus priimamus iš tinklo, net ir tuos, kurie neskirti stebinčiam įrenginiui. LIBPCAP biblioteką naudoja tcpdump, ethereal, Snort ir kitos populiarios tinklo srautų analizės programos detaliai paketų analizei [24].

Šis tinklo stebėjimo metodas iš apžvelgtų yra informatyviausias, nes gaunama ne tik tarnybinė informacija apie paketus: tinklo adresai, prievadai, bet ir perduodamos informacijos turinys. Naudojant tokią informaciją galima aptikti piktavališką elgseną tinkle ne tik pagal duomenų ar paketų kiekius, tačiau ir pagal jų turinį. Tačiau reikia saugoti ir apdoroti didžiulius informacijos kiekius, kas didelės spartos tinkle gali būti sudėtingas ar net neįmanomas uždavinys. Taip pat saugoti ar stebėti paketų pernešamą informaciją gali būti neteisėta vartotojų privatumo atžvilgiu [38].

1.3.4. Tinklo srauto stebėjimo metodų apibendrinimas

Taigi, skenavimo aptikimo įrankiai ir metodai gali remtis NetFlow arba LIBPCAP surinkta informacija apie tinklo srautus, tačiau SNMP iš principo šiam darbui netinka. Tuo tarpu NetFlow ir LIBPCAP tinklo stebėjimo metodai tinka šiam darbui nagrinėjamai situacijai, tačiau turi esminių skirtumų, kurie apibendrinti lentelėje 2.

Lentelė 2. NetFlow ir LIBPCAP tinklo stebėjimo metodų lyginamoji lentelė

Charakteristika	LIBPCAP	NetFlow	
Informatyvumas	Pilnas - duomenų mainų kopija	Dalinis - agreguota informacija iš pagrindinių IP ir TCP antraščių laukų	
Tipinė srauto stebėjimo vieta	Lokaliame tinkle: galinis tinklo mazgas ar komutatoriau sąsaja naudojant "port mirroring"	Globaliame tinkle: magistralės ar prieigos maršrutizatorius	
Stebėjimas realiu laiku	Taip	Min 1s vėlinimas	
Reikalingi resursai duomenų kaupimui	Didžiuliai	Vidutinis	180 MB NetFlow srautų faile saugoma informacija apie 69.7 GB tinklo paketų duomenis
Reikalingi resursai duomenų analizei	Didžiuliai	Vidutiniai	
Tipiniai stebėjimo objektai	Konkreti, kompiuteryje veikianti, aplikacija	Tinklo būsena, anomalijos, sujungimai tarp duotų tinklo mazgų ir prievadų	

NetFlow labiau orientuotas į viso tinklo stebėjimą, anomalijų jame radimą, susijungimų tarp tinklo mazgų ir prievadų stebėjimą. LIBPCAP padaro duomenų mainų kopiją suteikdama pilną informaciją apie paketą detaliai analizei atlikti, tačiau šiame darbe nagrinėjamai situacijai pilnai pakanka analizuoti NetFlow teikiamų IP ir TCP antraščių laukus, kurie reikalauja daug mažiau resursų duomenų kaupimui ir analizei. Pavyzdžiui, NetFlow srautų faile, kurio dydis 180 MB yra saugoma informacija, apie tinklo paketus, kurie srauto duomenų kopijos saugojimo atveju užimtų 69.7 GB. NetFlow šiuo konkrečiu atveju reikalauja 387 kartus mažiau vietos duomenims saugoti.

Toliau šiame darbe nagrinėjamai situacijai bus naudojamas NetFlow tinklo stebėjimo metodas.

1.4. Tinklo skenavimo įrankių apžvalga

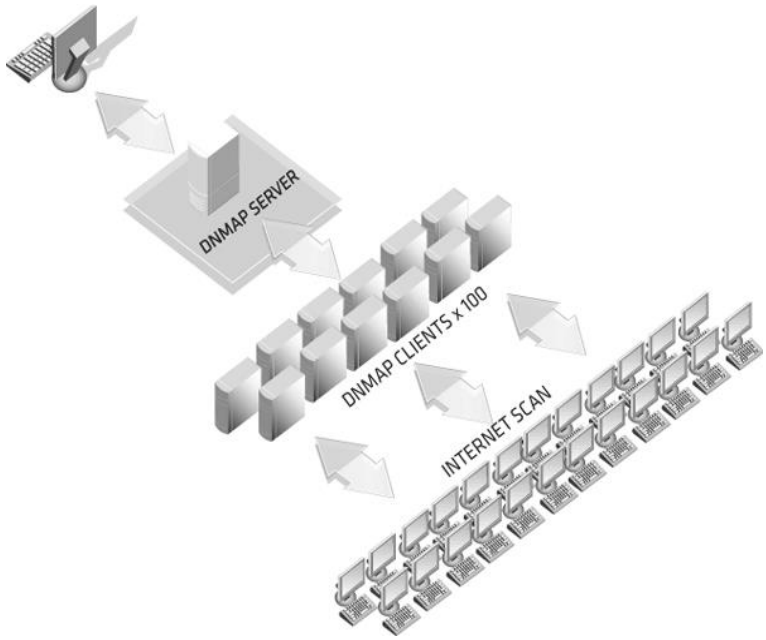
Yra daug įrankiu, kurie gali būti panaudoti sistemų silpnosioms vietoms rasti ir galimiems atakų scenarijams numatyti. Piktavaliai kuria specialius įrankius neteisėtiems veiksmams atlikti, tačiau juose naudojami metodai atkartojami ir legaliuose įrankiuose, kurie skirti tinklo ir jame veikiančių paslaugų saugumo spragoms aptikti. Eksperimentiniam tyrimui šiame darbe bus naudojamas vienas iš tokių įrankių.

1.4.1. Nmap

Geriausiai žinomas, detaliausiai aprašytas ir plačiausiai naudojamas yra atviro kodo įrankis tinklo tyrimui ir saugumo auditui Nmap, sukurtas Fyodor Yarochkin [12]. Nmap buvo kurtas didelių tinklų skenavimui, tačiau puikiai veikia ir prieš vieną taikinį. Įrankis naudoja įvairias skenavimo metodikas ir jų kombinacijas, tam kad nustatyti kokie įrenginiai veikia tinkle, kokios paslaugos juose paleistos, operacinių sistemų versijas, kokie paketų filtrai ir ugniasienės yra naudojamos ir daugybę kitokių charakteristikų. Nmap taip pat gali būti panaudotas sistemų administratorių užduotims, tokioms kaip: tinklo įrenginių inventorizavimui, įrenginių ir paslaugų pasiekiamumo stebėjimui [13]. Nmap turi scenarijų variklį, kuris leidžia vartotojams rašyti ir dalintis skenavimo scenarijais darbo su tinklu užduotims automatizuoti. Šie scenarijai yra vykdomi lygiagrečiai, todėl pasižymi greičiu ir efektyvumu. Vienas iš tokių scenarijų yra Seb Garcia sukurtas įrankis DNmap [14].

1.4.2. DNmap

DNmap yra Nmap skenavimo užduočių paskirstymo po aibę klientų karkasas. Įrankis perskaito failą su išsaugotomis Nmap skenavimo komandomis ir siunčia jas į prisijungusius klientus. Karkasas naudoja kliento – serverio architektūrą (4 pav.).



4 pav. DNmap architektūros schema [22]

Visa užduočių paskirstymo ir statistikos fiksavimo logika yra serveryje. Klientai atlikę užduotis rezultatus siunčia į serverį. DNmap puikiai tinka kai norima skenuoti didelį taikinių kiekį, kadangi skenavimą galima padalinti į keletą mažesnių dalių, iš kurių kiekviena bus atliekama atskiro kliento lygiagrečiai, taip paspartinant viso proceso eigą ir užmaskuojant vieningą atakos prigimtį. Vykdant skenavimą naudojantis šiuo įrankiu, taikinio pusėje bus matoma daug mažų skenavimų iš skirtingų šaltinių [14].

1.4.3. ZMap

ZMap yra atviro kodo skenavimo įrankis skirtas plataus masto Interneto skenavimui [15]. Kūrėjai nurodo, jog naudojant šį įrankį vidutinės klasės kompiuteryje be jokios papildomos aparatūrinės įrangos, viešąją IPv4 tinklo adresų erdvę galima nuskenuoti per 45 minutes, kai tuo tarpu naudojant anksčiau aprašytą įrankį Nmap tai užtruktų savaitę. Tokią spartą ZMap pasiekia, kadangi kitaip nei Nmap, po paketo išsiuntimo nelaukia atsakymo, o priima visus atsakymus, kurie ateina skenavimo metu. Taip pat įrankis į kiekvieną taikinį siunčia fiksuotą paketų skaičių (pagal nutylėjimą – 1), neatsižvelgdamas į įvykusias susijungimo pauzes (*angl. "timeout").*

1.4.4. Masscan

Masscan dar vienas atviro kodo įrankis, kuris pasižymi didesne sparta nei ZMap ir laikomas greičiausiu skenavimo įrankiu. Šiame įrankyje taip pat, kaip ir ZMap naudojamos atskiros procesų gijos paketų siuntimui ir atsakymų priėmimui. Abu skenavimo įrankiai – Zmap ir Masscan – pasižymi galimybe atlikti paskirstytą skenavimą, vykdomą kelių šaltinių [16].

Toliau pateikiama apžvelgtų skenavimo įrankių charakteristikų lyginamoji lentelė. Šie prievadų skenavimo įrankiai naudojami tiek sistemos administratorių, tiek piktavalių.

Lentelė 3. Esamų skenavimo įrankių palyginimas

Palyginimo kriterijus	**Nmap**	**DNmap**	**ZMap**	**Masscan**
Skenavimo metodas	Įvairūs	Įvairūs	TCP SYN	TCP SYN
Paskirstyto skenavimo galimybė	Nėra	Yra	Yra	Yra
Greitis	-	-	iki 10Gbps	Iki 10Gbps

Šiame darbe eksperimentui reikalingų, paskirstyto prievadų skenavimo duomenų pavyzdžių generavimui buvo pasirinkta naudoti DNmap, dėl savo paprastumo, bei lankstumo, kurį suteikia Nmap. Lyginant su ZMap ir Masscan paskirstyto skenavimo realizavimas yra paprastenis naudojant DNmap.

1.5. Įrankių, skirtų skenavimų aptikimui apžvalga

Šiame skyrelyje aprašomi praktikoje plačiausiai naudojami, jau tapę klasikiniais, įrankiai prievadų skenavimo aptikimui. 1.6. skyrelyje bus nagrinėjami literatūroje siūlomi metodai, kurie yra efektyvūs tam tikrų skenavimų tipų aptikimui ir gali būti naudojami autonomiškai arba kaip klasikinių įrankių papildiniai.

Įsilaužimų aptikimo sistema (IDS) yra programinė įranga, kuri stebi tinklo srautą ieškodama nuokrypių nuo normalios tinklo elgsenos, saugumo politikos pažeidimų ir informuoja apie tai sistemos administratorių [19]. Pagal informacijos šaltinį, IDS yra skirstomos į [20]:

- ***Veikiančias įrenginyje*** – analizuoja procesų identifikatorius ir sisteminius kreipinius, daugiausia susijusius su operacinės sistemos informacija.
- ***Veikiančias tinkle*** – analizuoja su tinklu susijusią informaciją – srauto apimtį, IP adresus, paslaugų prievadus, protokolų panaudojimą, paketų turinį.

Pagal tai, kokio tipo analizę vykdo, šios sistemos skirstomos į [21]:

- ***Įsilaužimo pėdsakų paieška pagrįstos*** – analizuojamoje informacijoje ieško gerai žinomų įsilaužimų pėdsakų iš turimos įsilaužimų charakteristikų duomenų bazės, tinka aptikti žinomas atakas.
- ***Anomalijų aptikimu pagrįstas*** – bando nustatyti normalų sistemos ar tinklo veikimo modelį ir vėliau praneša apie bet kokius nuokrypius nuo normų. Naudojamos aptikti naujus, dar neužfiksuotus įsilaužimų pėdsakų duomenų bazėje, įvykius. Nepaisant neatitikimų ar trūkumų duomenų bazėje, anomalijų aptikimu pagrįstos sistemos dažniausiai pateikia daugiau klaidingai rezultatų.

Šiame darbe įsilaužimo aptikimo sistemos bus analizuojamos tik skenavimų aptikimo požiūriu. Dažniausiai įsilaužimų aptikimo sistemos įtarimą apie prievadų skenavimą detaliau tikrina analizuodamos susietus paketus tam tikrame laiko lange [2].

1.5.1. Snort

Snort yra atviro kodo tinklo įsilaužimų aptikimo sistema, pagrįsta įsilaužimo pėdsakų paieška naudojantis iš anksto nustatytomis taisyklėmis, ir galinti realiu laiku tirti IP tinklo srautą. [34]. Šios taisyklės gali būti sudaromas pasinaudojant Snort suteikiama taisyklių sintakse tinklo srautui apibūdinti ir vėliau panaudotos tiriant tinklo paketus. Snort gali veikti trimis skirtingais režimais: tinklo įsilaužimų aptikimo, paketų sekimo ir paketų registravimo. Tinklo stebėjimui Snort naudoja LIBPCAP biblioteką.

Snort turi papildomų komponentų, kurie leidžia apdoroti paketus prieš jiems pasiekiant taisyklių apdorojimo variklį, vadinamų išankstiniais procesoriais. Vienas iš galimų šių komponentų panaudojimo atvejų yra prievadų skenavimo aptikimas. Snort įsilaužimų aptikimo sistemos modulis skenavimų aptikimui buvo sukurtas aptikti būtent įvairius prievadų skenavimo atvejus, kuriuos gali vykdyti Nmap įrankis [39]. Snort gali aptikti keturių skirtingų atlikimo tipų prievadų skenavimą [39]:

- Prievadų skenavimas – skenuojamas vienas ar keli prievadai viename įrenginyje
- Paskirstytas prievadų skenavimas – skenuojamas vienas ar keli prievadai viename įrenginyje, tačiau skenavimą atlieka keli skirtingi šaltiniai
- Sutrikdantis prievadų skenavimas – skenuojamas vienas ar keli prievadai viename įrenginyje, atliekamas kelių skirtingų šaltinių, tačiau į kiekvieną prievadą bandoma prisijungti keletą kartų
- Blokinis prievadų skenavimas – skenuojamas keletas prievadų aibėje įrenginių

Skenavimas aptinkamas remiantis neteisingomis TCP paketo vėliavėlių kombinacijomis, arba pastebėjus per didelį prisijungimų skaičių į skirtingus prievadus arba IP adresus. Pagal nutylėjimą Snort praneša jei aptinka paketus su SYN vėliavėle nusiųstus į 5 skirtingus IP adresus arba 20 skirtingų prievadų 60 sekundžių laikotarpyje. Šio kintamųjų vertės gali būti nustatomos konfigūracijoje. Piktavalis gali likti nepastebėtas, jei atliks skenavimą pakankamai lėtai [33].

Snort skenavimo atakų aptikimas paremtas trijų išankstiniu procesorių veikimu, kurie turi būti aktyvuoti kartu, norint aptikti prievadų skenavimą: Frag3, Stream5 ir sfPorscan. Kiekvienas iš jų atlieka papildomą kiekvieno tinklo paketo apdorojimą skenavimo aptikimui, todėl išauga įrenginio skaičiavimo resursų panaudojimas ir apdorojimo laikas.

Išanalizavus šios įsilaužimų aptikimo sistemos techninę dokumentaciją buvo nustatyta, kad Snort nesugeba aptikti *„daug į daug“* tipo paskirstyto prievadų skenavimo atakų, kurių metų daug šaltinių skenuoja po vieną ar keletą tinklo įrenginių, todėl kad per nustatytą laiko langą registruojama mažiau kreipinių į unikalius IP adresus tinkle, ateinančių iš skirtingu šaltinių, negu Snort jautrumo lygis. Skenavimą aptinkančio išankstinio paketų procesoriaus pagalba galima padidinti Snort jautrumą prievadų skenavimo atakoms, tačiau tuomet saugomas didesnis tinklo paketų laiko langas, o tai reikalauja daugiau įrenginio atminties ir procesoriaus laiko, be to išauga neteisingai identifikuotų atakų pranešimų skaičius.

1.5.2. Bro

Bro yra atviro kodo įsilaužimų aptikimo sistema atliekanti pasyvų tinklo srauto stebėjimą, naudojantis LIBPCAP srauto stebėjimo metodu, ieškant įsilaužimo pėdsakų. Sistema buvo kurta siekiant atlikti tinklo, pasižyminčio didele sparta, stebėjimui realiu laiku.

Bro stebi tinklą pasinaudodamas įvykių varikliu, tam kad sugeneruoti įvairius įvykius, paremtus interpretuojamu tinklo srautu. Tuomet šie įvykiai yra apdorojami naudojantis vartotojo suteiktais scenarijais tinklo saugumo analizei.

Bro sugrupuoja paketus panašiu principu kaip NetFlow, kuomet paketai turintys vienodus parametrus – šaltinio IP adresas, šaltinio prievadas, taikinio prievadas – identifikuojami, kaip priklausantys vienam susijungimui. Jei atlikus paketo analizę yra iššaukiamas vienas iš įvykių – *susijungimas sudarytas, bandymas prisijungti, susijungimas atmestas* – paketų grupė yra apdorojama algoritmo, aprašyto [32]. Jei

paketo šaltinis peržengia nustatytą leidžiamą nesėkmingų susijungimų ribą, jis laikomas atliekančiu skenavimą.

1.6. Literatūroje aprašytų paskirstyto skenavimo aptikimo metodų apžvalga

Šiame skyrelyje bus apžvelgti paskirstyto skenavimo aptikimo metodai rasti nagrinėjant mokslinę literatūrą. Pagal naudojamus įrankius ir problemos sprendimo būdus jie gali būti suskirstyti į klases: algoritminiai, grupavimo ir vizualiniai [5].

1.6.1. Grupavimu paremti aptikimo metodai

[11] autoriai sukūrė metodą slapto (*angl. stealthy*) ir paskirstyto prievadų skenavimo aptikimui, tačiau jis taip pat tinka ir DoS atakų ir įrenginių bei tinklo konfigūracijos klaidų aptikimui. Pirmiausiai paketai yra apdorojami pasinaudojant Snort įsilaužimo aptikimo sistemos papildiniu tinklo anomalijų aptikimui Spade, kuris pažymi kiekvieną paketą kaip normalų arba įtartiną. Tuomet visi paketai pažymėti kaip įtartini yra perduodami į koreliacijos variklį Spice, kur jie yra patalpinami grafe, taip kad paketai, kurie labiausiai panašūs vienas į kitą yra sugrupuojami kartu. Šis aptikimo metodas netinka paskirstytų skenavimų, kuomet daug šaltinių skenuoja daug taikinių, aptikimui. Metodas nustatytą laiko periodą saugo šias grupes, todėl pasirinkus per didelį periodą gali tekti atlikinėti paieškas dideliame grafe, o tai atsiliepia išaugusiu skaičiavimo resursų poreikiu. Spice papildinio realizacijos rasti nepavyko. Spade papildinys jau ilga laiką nėra prižiūrimas ir neveikia su naujomis Snort versijomis.

[35] Whyte savo darbe suprojektavo ir realizavo prievadų skenavimo aptikimo priemonių rinkinį, kuris leidžia aptikti trijų skirtingų tipų prievadų skenavimą dideliame tinkle: skenavimą tinklo viduje, lokalaus tinklo atliekamą skenavimą į išorinį tinklą ir išorinio tinklo atliekamą lokalaus tinklo skenavimą.

.

Skenavimui iš išorinio tinklo aptikti panaudojamas demaskavimo žemėlapių sudarymas. Metodas reikalauja mokymosi laikotarpio, kurio metu yra stebima tinklo elgsena, sekant kaip įrenginių prievadai atsako į prisijungimą. Jei prievadas atsako į prisijungimą, jis įtraukiamas į tinkle veikiančių paslaugų žemėlapį. Jei prievadas neatsako, jis įtraukiamas į *„tamsiųjų prievadų“* sąrašą, kurie neturėtų sulaukti prisijungimo užklausų. Mokymosi laikotarpio trukmė priklauso nuo tinklo dydžio ir aktyvių įrenginių tinkle skaičiaus. Pasibaigus mokymosi laikotarpiui, apie visas tinkle naujai atsiradusias paslaugas informuojamas tinklo administratorius, o žemėlapį reikia atnaujinti.

Metodas tinka visų tipų prievadų skenavimo atakoms aptikti, įskaitant ir *„daug su vienu“* ir *„daug su daug“* tipo paskirstyto prievadų skenavimo atakas. Tinklo paketų analizės metu yra užfiksuojami prisijungimo mėginimai į *„tamsiuosius prievadus“*, kurie neturėtų sulaukti prisijungimo, todėl skenavimas gali būti aptiktas vos po vieno mėginimo. Prisijungimo užklausos nukreiptos į tinkle naujai paleistas

paslaugas yra identifikuojamos kaip skenavimas, kol šios paslaugos neįtraukiamos į žemėlapį. Skenavimai į prievadus, kurie yra įtraukti į tinkle teikiamų paslaugų žemėlapį, nėra aptinkami.

Apžvelgtas metodas reikalauja mokymosi laikotarpio ir nuolatinio veikiančių paslaugų žemėlapio atnaujinimo, kitaip yra generuojami klaidinti aptikto skenavimo pranešimai. Taip pat nėra aptinkami skenavimai į tinkle veikiančias, žinomas paslaugas. Autorius aprašo metodo realizaciją kaip Bro įsilaužimų aptikimo sistemos politikų rinkinį, tačiau paviešintos realizacijos versijos rasti nepavyko.

1.6.2. Algoritminiai aptikimo metodai

[32] autoriai naudoją algoritmą, kuriam yra reikalinga informacija apie įrenginius ir paslaugas, kurios yra pasiekiamos tinkle. Tuomet kiekvienai naujai prisijungimo užklausai yra atliekamas nuoseklus hipotezių tikrinimas, norint nustatyti ar šaltinis atlieka skenavimą. Yra daroma prielaida, kad paprasti vartotojai žino apie veikiančias paslaugas adresu, kuriuo jie kreipiasi ir yra mažesnė tikimybė, kad jie kreipsis į uždarus prievadus, nei tie, kurie atlieka skenavimą. Šiam algoritmui užtenka 5 mėginimų prisijungti į skirtingus adresus, kad būtų identifikuotas skenavimas. Autoriai pabrėžia, kad jų sukurtas algoritmas skirtas aptikti pavienius, skenavimą atliekančius, šaltinius, tačiau norint aptikti paskirstyto skenavimo atakas, reikėtų jį praplėsti.

[33] Gates pateikia piktavalio modelį, sudarytą pagal informaciją, kurią jis nori išgauti. Kiekvienas piktavalis yra susiejamas su atitinkamu skenavimo pėdsakų šablonu. Šie modeliai vėliau naudojami paskirstyto skenavimo aptikimui, naudojantis sudarytu algoritmu, kuris yra paremtas aibės padengimo problemą sprendimo paieška – ieškoma mažiausio skaičiaus pavienių skenavimų, kurie kartu padengia didžiausią dalį skenuojamo tinklo. Autorius įvertina modelį ir skenavimo aptikimo mechanizmo veiksmingumą atlikdamas eksperimentus su keliais skirtingais duomenų rinkiniais.

Metodas aptinka tik horizontalų prievadų skenavimą, kuomet yra skenuojamas vienas prievadas aibėje įrenginių. Šiam paskirstyto prievadų skenavimo aptikimo metodui reikalingas atskiras vieno šaltinio atliekamų skenavimų aptikimo įrankis, kurio surinkti duomenys vėliau panaudojami, kaip įvesties duomenų rinkiniai paskirstyto skenavimo aptikimo algoritme, todėl paskirstytų skenavimų aptikimo efektyvumas priklauso nuo pasirinkto vieno šaltinio atliekamų skenavimų aptikimo įrankio efektyvumo.

Tam, kad paskirstytas skenavimas būtų aptiktas, jis turi padengi nustatytą minimalią dalį stebimo tinklo. Tarkime jei buvo nustatyta, jog skenavimas turi padengti 70 % tinklo, o piktavalis, atlikdamas ataką padengia tik 65 % tinklo, jo koordinuoti veiksmai lieka nepastebėti. [33] aprašytų eksperimentų metu, šis kintamasis buvo nustatytas 95 %.

Šis skenavimo aptikimo metodas nėra veikiantis realiu laiku, ir gali aptikti skenavimą tik jam pasibaigus.

1.6.3. Vizualiniai aptikimo metodai

[31] Conti ir Abdullah pabandė aptikti paskirstytas skenavimo atakas vizualinėmis priemonėmis atvaizduodami tinklo srautą, tačiau neparodė, kaip atrodytų paskirstytas skenavimas normaliame tinklo sraute naudojantis jų įrankiu. Šis metodas reikalauja žmogaus, kuris analizuotų pateiktą informaciją, ieškotų joje atakų šablonų, todėl nepasižymi veikimu realiu laiku ir aptikimo automatizuotumu.

1.6.4. Literatūroje aprašytų paskirstyto skenavimo aptikimo metodų apibendrinimas

Lentelė 4. Paskirstyto skenavimo atakų aptikimo metodų palyginimo lentelė

Klasė	Nuoroda	Veikia realiu laiku	Aptikimo efektyvumas, %	Neteisingai aptikti, %	Analizuojamos informacijos tipas
Grupavimas	[11]	Ne	97.5	-	Paketų lygio
	[35]	Taip	99	0.5	Paketų lygio
Algoritminis	[32]	Ne	98.3	0.5	Paketų lygio
	[33]	Taip	96.3	2.5	Paketų lygio
Vizualinis	[31]	Ne	-	-	Paketų lygio

Lentelėje 3 apibendrintos kelios svarbios apžvelgtų paskirstyto prievadų skenavimo aptikimų metodų charakteristikos. Visi metodai pasižymi aukštu skenavimo aptikimo efektyvumu bei žemu klaidingai aptiktų skenavimų kiekiu, išskyrus [31], kuris tiesiogiai aptikimo neatlieka. Nei vienas iš apžvelgtų aptikimo metodų nėra paremtas tinklo srautų informacijos analize prievadų skenavimo aptikimui. [32][35] yra realizuoti kaip Bro įsilaužimų aptikimo sistemos politikų rinkiniai, tačiau [35] realizacijos, kuri pasirodė gana sudėtinga, rasti nepavyko. Taip pat neminima [33] realizacija, kuri taip pat gana sudėtinga. [11] realizuotas kaip Snort papildinys, tačiau jau ilgą laiką nėra palaikomas ir neveikia su naujausiomis šios įsilaužimų aptikimo sistemos versijomis.

1.7. Probleminės srities analizės rezultatai

Atlikus probleminės srities analizę nustatytą, kad paskirstytų skenavimo atakų, kurių metu skirtingi šaltiniai skenuoja aibę prievadų skirtinguose įrenginiuose (3 pav. B), aptikimas nėra plačiai nagrinėjamas.

[32] metodas reikalauja papildomo įrankio vieno šaltinio vykdomam skenavimui aptikti, kuris vėliau panaudojamas, kaip sukurto algoritmo įvesties duomenys, bei didelės dalies tinklo, padengto skenavimu, tam kad pasiekti maksimalų aptikimo efektyvumą. [35] aprašytas metodas reikalauja mokymosi laikotarpio, kurio metu stebint tinklą yra sudaromas veikiančių paslaugų žemėlapis. Kiekviena naujai paleista paslauga, sulaukianti prisijungimų ir neįtraukta į žemėlapį traktuojama kaip skenavimas. Nėra

aptinkami skenavimai į paslaugas, kurios yra įtrauktos į veikiančių paslaugų žemėlapį. Autorius mini metodo realizaciją, tačiau ji nėra viešai prieinama. [11] metodas realizuotas Snort įsilaužimų aptikimo sistemos papildinių pavidalu, tačiau šie papildiniai jau senai nebepalaikomi ir neveikia su naujausiomis Snort versijomis.

Remiantis kelių skirtingų autorių pateiktais paskirstyto prievadų skenavimo apibrėžimais, buvo sudarytas apibrėžimas tinkamiausias šio darbo kontekstui: aibė skenavimų, kuriuos atlieka skirtingi šaltiniai, valdomi vieno organizatoriaus ir turintys IP adresus, esančius arti vienas kito adresų erdvėje.

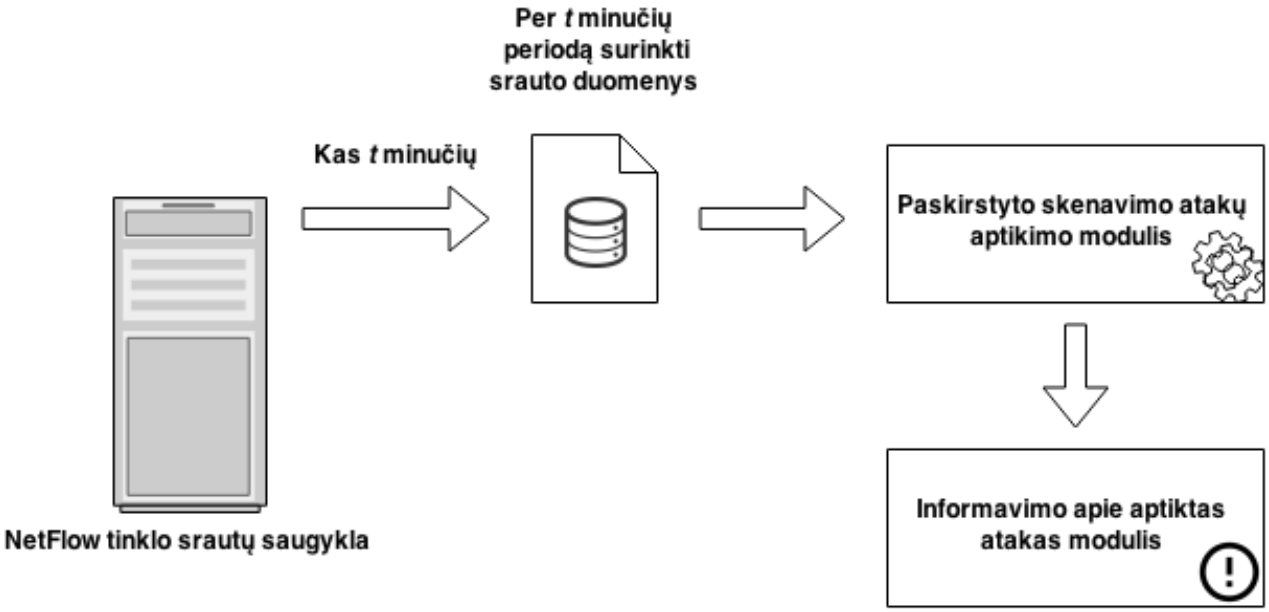
Taip pat analizuojant dokumentaciją buvo pastebėtas šio tipo skenavimų aptikimo trūkumas įsilaužimų aptikimo sistemoje Snort. Toliau šiame darbe bus pasiūlytas minėto tipo paskirstyto prievadų skenavimo atakų aptikimo metodas, analizuojant surinktą NetFlow tinklo srautų informaciją ir jo efektyvumas palygintas su skenavimų aptikimu naudojant Snort įsilaužimų aptikimo sistemą.

2. PROJEKTINĖ DALIS

Probleminės srities analizės metu buvo nustatytos gairės tolesnei šio darbo eigai. Siekiama aptikti paskirstyto skenavimo atakas, kurias atlieka keletas skirtingų šaltinių, skenuodami vieną prievadą aibėje įrenginių, kur kiekvienas šaltinis siunčia užklausas į vieną ar daugiau taikinių. Tam, kad pasiekti šį tikslą siūlomas naujas tokio tipo atakų aptikimo metodas.

2.1. Paskirstyto prievadų skenavimo aptikimo sistemos prototipas

Paskirstyto prievadų skenavimo atakų aptikimui darbe buvo suprojektuotas aptikimo sistemos prototipas, kuris pavaizduotas 5 paveiksle.



5 pav. Skenavimo atakų aptikimo sistemos prototipas

Sistema susideda iš trijų pagrindinių komponentų:

- NetFlow tinklo srautų informacijos saugykla – joje saugomi surinkti srautų įrašų duomenys;
- Paskirstyto prievadų skenavimo atakų aptikimo modulis – jame realizuotas srauto informacijos analizės ir atakų aptikimo algoritmas;
- Informavimo modulis – pranešimams apie aptiktas prievadų skenavimas.

Surinktų srautų analizė vykdoma periodiškai, kas *t* minučių. Kintamasis *t* gali būti laisvai pasirenkamas, tačiau reikia atsižvelgti į keletą aspektų:

- Tam, kad nuslėpti atliekamą skenavimą, piktavalis gali nustatyti laiko intervalą tarp atskirų paketų ir atlikti lėtą prievadų skenavimą. Nustačius per mažą stebimą tinklo informacijos laiko langą, toks lėtas skenavimas gali neviršyti skenavimo aptikimo algoritmo nurodytų ribinių verčių ir likti nepastebėtas.

- Dideliame tinkle per laiko vienetą gali būti surenkamas didelis kiekis NetFlow srauto įrašų, kurie yra siunčiami į srauto saugyklą periodiškai. Nors, kaip buvo minėta 1.3.4., NetFlow formatu saugoma tinklo srautų įrašų informacija efektyviai naudoja saugojimo vietą, stebint ilgą tinklo informacijos periodą, gali susidaryti didelis informacijos, kurią reikia apdoroti, kiekis.

2.2. Paskirstyto prievadų skenavimo aptikimo algoritmas

Analizuojant literatūroje minimus prievadų skenavimo aptikimo metodus, buvo paminėtas algoritmas, kuris iš esmės nėra skirtas šio tipo atakų aptikimui, vadinamas *TRW* (*angl. Threshold Random Walk*), kuris laikomas vienu moderniausiu prievadų skenavimo aptikimo algoritmų aprašytų literatūroje, skirtų vieno šaltinio atliekamam skenavimui aptikti [32]. Šiame darbe parskirstytam skenavimui aptikti naudojamo algoritmo pagrindu pasirinktas būtent *TRW* algoritmas.

Šio algoritmo esmė yra dviejų hipotezių, H_0 *ir* H_1 įvertinimas, čia H_0 - prisijungimas yra atliekamas paprasto vartotojo, H_1 – prisijungimas yra atliekamas skenuotojo. Prisijungimo užklausą siunčiantį šaltinį įvardijus kaip *r,* o prisijungimo į unikalų lokalų įrenginį įvykio baigtį įvardijus Y_i , kur $Y_i = 0$, jei prisijungimas pavyko, $Y_i = 1$, jei prisijungimas nepavyko, čia $i = 1, 2, 3,..$ nurodo unikalaus taikinio, į kurį kreipiasi šaltinis, eilės numerį. Kiekvieno naujo prisijungimo šaltiniui apskaičiuojama santykinė tikimybė, kad jis yra paprastas vartotojas arba skenuotojas [32]:

$$\Lambda(r) = \prod_{i=1}^{n} \left(\frac{P_r[Y_i|H_1]}{P_r[Y_i|H_0]} \right)$$

čia Y_i yra įvykių rinkinys apie prisijungti bandantį šaltinį *r* ir kiek kartų jis bandė jungtis į lokalų tinklą *n.* Su kiekvienu nepavykusiu prisijungimu didėja tikimybė, nurodanti, kad vartotojas atlieka skenavimą. Jei šaltinis neperžengia nustatytų ribų, kada jis identifikuojamas kaip paprastas vartotojas ar kaip skenuotojas, algoritmas nepriima jokio sprendimo apie šį šaltinį. Šiuos šaltinius neperžengusius ribos galime panaudoti paskirstyto skenavimo paieškai, nes jie galimai žino apie aptikimo algoritmą ir bando jo išvengti.

Aptikimo metodo sudarymui išskiriami du galimi *„daug su daug“* tipo paskirstyto prievadų skenavimo atvejai:

- Skenuoja šaltiniai, turintys artimus IP adresus, priklausančius tam pačiam potinkliui *L.* Galime daryti prielaidą, kad tinklas, iš kurio atliekamas skenavimas nėra tinkamai apsaugotas nuo IP adresų klastojimo, o atakos organizatorius tuo naudojasi, slėpdamas savo tapatybę. Taikinys į užklausas atsako suklastotam šaltinio adresui, o ne paketo siuntėjui. Jei suklastotas adresas priklauso tam pačiam lokaliam tinklui, atakos organizatorius gali sužinoti atsakymą iš taikinio sekdamas tinklo paketus [41].

  Kita prielaida, kad piktavalis gali turėti nuosavą tinklo segmentą arba prieigą prie IPT tinklo segmento, kuriame išoriniai IP adresai skirstomi DHCP serverio ir buvo gauti skirtingi IP adresai kelioms įrenginio ar įrenginių tinklo sąsajoms, kurios panaudojamos skenavimui.

Išvardintais atvejais skenavimą atliekantys IP adresai tinklo adresų erdvėje bus arti vienas kito. [30] tyrimas parodė, kad skenavimą atliekančius šaltinius, turinčius IP adresus arti vienas kito, galima sugrupuoti naudojantis pakankamai mažu grupės dydžių ≤ 1024, tam kad į grupes apjungti didžiąją jų dalį. Šis grupės dydis prilygsta IP adresų potinkliui su kauke *255.255.252.0.*

- Skenuoja šaltiniai priklausantys virusais užkrėstų kompiuterių zombių tinklams. Šiuo atveju IP adresai yra pasiskirstę visoje IP adresų erdvėje tarpusavyje neturintys aiškios koreliacijos. Šio atvejo darbe nenagrinėjame.

TRW algoritmas iš esmės pritaikytas NetFlow srauto įrašų analizei, kadangi realizacija, naudojama Bro įsilaužimų aptikimo sistemoje analizuoja duomenis, savo struktūra labai primenančius NetFlow srauto įrašą [29].

Remiantis srauto įrašo TCP vėliavėlių reikšmėmis, reikia įvardinti galimas susijungimo būsenas, pagal kurias algoritmas atpažins skenuotojo ir paprasto vartotojo atliekamus prisijungimus. Kadangi šiame darbe bandome aptikti *„TCP SYN“* tipo skenavimą, mus domina lentelėje 5 nurodytos vėliavėlių kombinacijos. Visos kitos TCP vėliavėlių kombinacijos bus laikomos paprasto vartotojo požymiu.

Lentelė 5. TCP SYN skenavimo būsenos

Susijungimo būsena	Apibūdinimas
S	Paketas su SYN vėliavėle, reiškiantis naują prisijungimą, kuris negavo jokio atsakymo
SR	Paketas su SYN vėliavėle, į kurį buvo atsakyta paketu su RST vėliavėle – prievadas uždaras
SAR	Paketas su SYN vėliavėle, į kurį buvo atsakyta paketu su SYN/ACK vėliavėlėmis. Susijungimo iniciatorius nutraukė susijungimą RST paketų – prievadas atviras

Remiantis visomis padarytomis prielaidomis, siūloma papildyti TRW algoritmo veikimą. Siūlomo paskirstyto prievadų skenavimo aptikimo algoritmo žingsniai:

1. Paimama *t* trukmės NetFlow informacija iš saugyklos.
2. TRW algoritmas įvertina kiekvieną srauto įrašą, kaip atliekantį skenavimą arba paprastą vartotoją.
3. Gauname aibę šaltinių, identifikuotų kaip skenuotojai, atliekantys skenavimą iš vieno šaltinio.
4. Kiekvienam iš šių šaltinių paskaičiuojame potinklio, kuriam jis priklauso, adresą. Skaičiavimas atliekamas remiantis proceso pradžioje nustatyta artimų IP adresų grupės dydžio reikšme *L*.
5. Atliekame 4 žingsnyje nurodytą procesą visiems šaltiniams, kurie nebuvo identifikuoti, kaip atliekantys skenavimą, tačiau nebuvo pridėti į paprastų vartotojų sąrašą.

6. Išmetame visas grupes, kurios turi tik po vieną joms priskirtą šaltinį. Likusioms grupėms paskaičiuojame vidutinę grėsmės koeficientą, kuris lygus visų grupės narių tikimybių, kad jie atlieka skenavimo ataką, sumos vidurkiui. Jei ši reikšmė viršija nustatytą ribą, visa grupė identifikuojama kaip paskirstyto skenavimo šaltinis.

2.3. Siūlomo metodo apibendrinimas

Skyrelyje aprašytas siūlomas metodas bus paremtas gerai žinomu ir pripažintu vieno šaltinio prievadų skenavimo aptikimo algoritmu, kuriam užtenka 4 mėginimų prisijungti į uždarus prievadus skenavimo faktui nustatyti. Dar vienas algoritmo pliusas, kad jis veikia sistemoje, kurioje analizuojami duomenys struktūra panašūs į NetFlow srauto įrašus. Skenavimą atliekantys šaltiniai bus grupuojami, su šaltiniais, kurie galimai atliko skenavimą, bet neviršijo aptikimo ribos į grupės, pagal tai kaip arti vienas kito adresų erdvėje jie yra. Grupės, kurių skenavimo tikimybės vidurkis viršys nustatytą ribą nariai bus laikomi paskirstyto prievadų skenavimo šaltiniais.

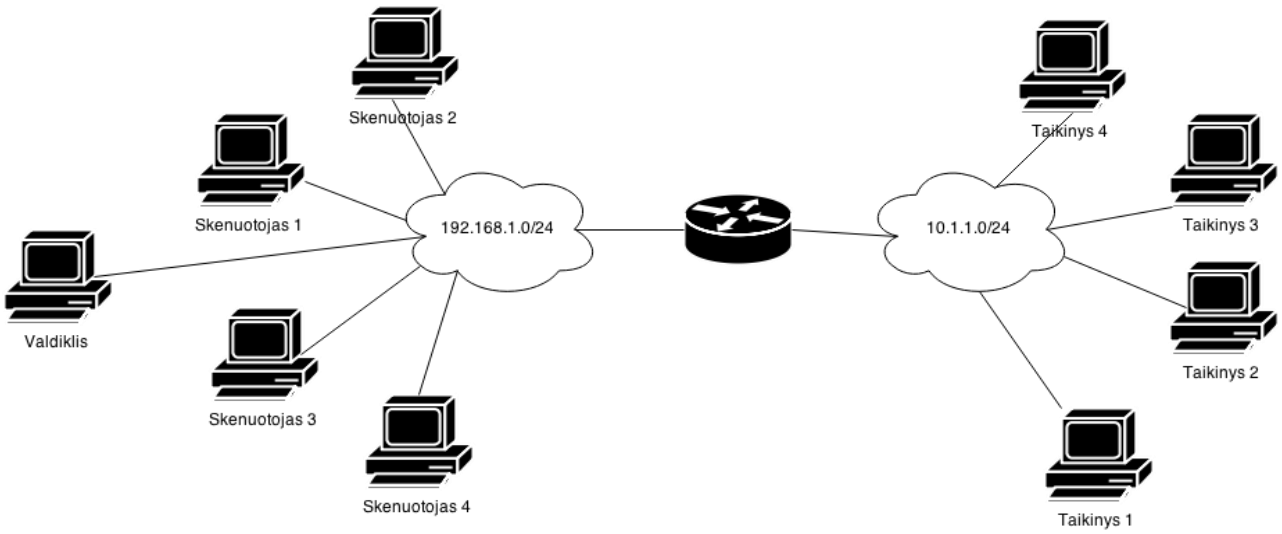
3. TYRIMAS

Šioje darbo dalyje eksperimentiškai palygintos sudaryto paskirstyto prievadų skenavimo aptikimo metodo galimybės ir efektyvumas su klasikiniu prievadų skenavimo aptikimo įrankiu, kuriuo buvo pasirinkta Snort įsilaužimų aptikimo sistema. Eksperimentui buvo naudojamas sintetinis tinklo srautas, registruotas virtualiame tinkle, CAIDA 2015 metų pasyvaus tinklo stebėjimo anonimizuoti duomenų rinkiniai [44], bei realaus KTU tinklo srauto duomenys.

3.1. Eksperimento atlikimo darbo vieta

Paskirstyto skenavimo atakos atlikimui buvo sukurtas eksperimentinis virtualus tinklas, kurio principinė schema matoma 6 paveiksle. Du tinklai tarpusavyje sujungti per maršrutizatorių, kurį eksperimente atstoja virtuali mašina su Linux operacine sistema, sukonfigūruota praleisti iš tinklo A *(192.168.1.0/24)* į tinklą B *(10.1.1.0/24)* nukreiptus paketus. Maršrutizatoriuje įdiegtas Snort įsilaužimų aptikimo sistema. Viso eksperimento metu buvo naudojami numatytieji Snort nustatymai, išskyrus prievadų skenavimo aptikimo modulį.

Sukurtas metodas aptinka paskirstyto skenavimo atakas, vykdomas šaltinų, kurie yra arti vienas kito IP adresų erdvėje – priklauso tam pačiam potinkliui. Eksperimento metu ataką atliekantys šaltiniai buvo sujungti į vieną potinklį su IP adreso kaukės bitų skaičiumi $L = 24$.



6 pav. Eksperimentui naudotos tinklo topologijos schema.

Kadangi Snort neturi priemonių NetFlow tinklo srauto įrašams analizuoti, eksperimento metu buvo pasinaudota tcpdump tinklo paketų stebėjimo įrankiu naudojančiu LIBPCAP biblioteką tinklo duomenų mainų kopijai faile išsaugoti. Šis failas buvo konvertuotas į NetFlow formatą, kurį gali analizuoti sudarytas aptikimo metodas. Tuo tarpu Snort analizuoja to paties tinklo srauto duomenų kopijos failą.

Skenavimo vykdymui panaudotas įrankis DNmap, aprašytas 1.4.2. skyrelyje, kuris įdiegtas aibėje įrenginių. Šiems įrenginiams komandas siunčia kompiuteris - valdiklis.

Paskirstyto prievadų skenavimo atakų aptikimo algoritmas buvo realizuotas PHP programavimo kalba, jos vykdymui pasinaudojant Facebook HHVM [45].

3.2. Paskirstyto prievadų skenavimo aptikimo efektyvumo tyrimas

Siekiant įvertinti šiame darbe pasiūlyto metodo efektyvumą buvo atlikta eilė įvairaus tipo paskirstyto prievadų skenavimo atakų, o aptikimo rezultatai palyginti su rezultatais gautais bandant šias atakas aptikti Snort įsilaužimų aptikimo sistemos pagalba. Buvo vykdomi tokie scenarijai:

1. Skenavimą vykdo keletas šaltinių. Atliekamas *„daug su vienu“* tipo skenavimas – skenuojama aibė prievadų viename taikinyje.
2. Skenavimą vykdo keletas šaltinių. Skenuojamas vienas prievadas aibėje taikinių. Vienas šaltinis skenuoja 4 taikinius.
3. Skenavimą vykdo keletas šaltinių. Skenuojamas visas potinklis dalimis, dalis paskirstant šaltiniams. Kreipiamasi į vieną prievadą.

Skenavimų aptikimui buvo naudojami standartiniai Snort skenavimų aptikimo modulio nustatymai, nurodyti 6 lentelėje. 7 lentelėje matome vykdytų paskirstytų skenavimo atakų aptikimo rezultatus. Eksperimente iš viso buvo skenuojami 24 taikiniai, tačiau šią užduotį atliko 6 šaltiniai skenuodami po 4 taikinius.

Lentelė 6. Snort skenavimų aptikimo modulio konfigūracija

```
preprocessor sfportscan:
        proto  { all }
        sense_level { low }
        scan_type { all }
```

Lentelė 7. Paskirstyto skenavimo scenarijų suvestinė

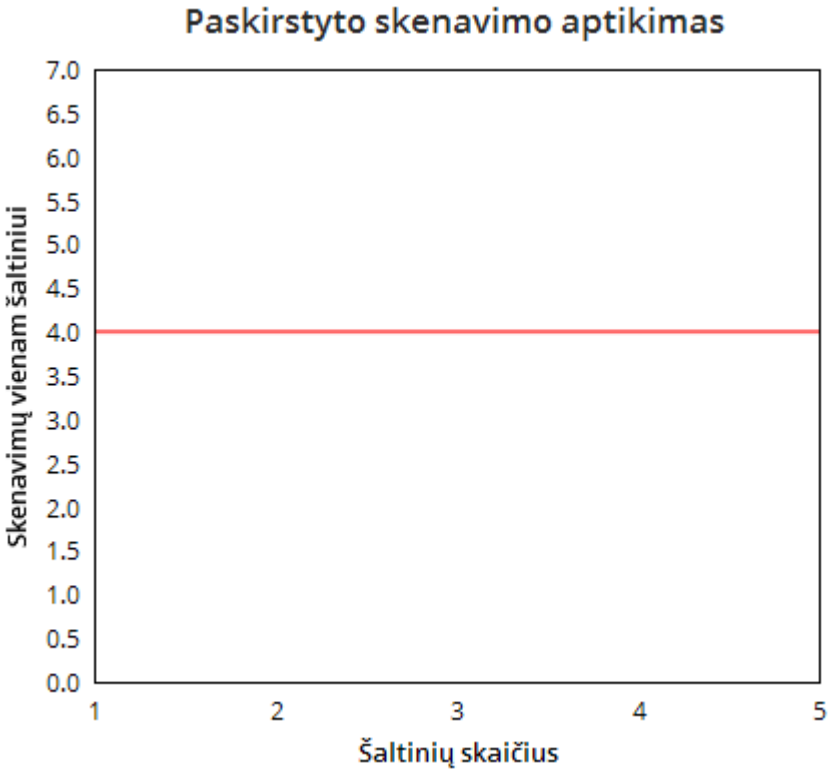
Scen. nr.	Skenavime dalyvaujančių šaltinių skaičius	Skenuojamų taikinių skaičius vienam šaltiniui	Skenuojamų prievadų skaičius	NetFlow analizės metodas	Snort
1	6	1	40	Aptiktas	Aptiktas
2	6	4	1	Aptiktas	Neaptiktas
3	6	30	1	Aptiktas	Aptiktas

Snort nesugebėjo susidoroti su antruoju skenavimo scenarijumi, kai aibė šaltinių skenavo po 4 taikinius. Kadangi Snort neaptiko šios atakos, galime daryti išvadą, kad Snort nekoreliuoja arti vienas kito IP adresų erdvėje esančių adresų. Padidinus skenuojamų taikinių skaičių iki 5, Snort aptiko skenavimą, tačiau pažymėjo jį ne kaip paskirstytą skenavimą, o atskirų šaltinių atliekamą skenavimą. Konkrečiu atveju buvo aptikti 6 atskiri vieno šaltinio atliekami skenavimai. Jei bent vienas iš paskirstytą prievadų skenavimą

atliekančių šaltinių neviršija šios ribos, jis lieka nepastebėtas. Snort praneša jei aptinka paketus su SYN vėliavėle nusiųstus į 5 skirtingus IP adresus arba 20 skirtingų prievadų 60 sekundžių laikotarpyje.

Snort sfPortscan modulis gali veikti skirtingo jautrumo lygiais. Kadangi veikiant numatytuoju, žemo jautrumo lygiu nepavyko aptikti prievadų skenavimo fakto (nr. 2 lentelėje 7), buvo nustatytas aukščiausias jautrumo lygis, tikintis, kad bus aptiktas anksčiau nepastebėtas skenavimas, kai skirtingi šaltiniai skenuoja po 4 taikinius. Dokumentacijoje perspėjama, kad nustačius aukščiausią jautrumo lygį išauga neteisingai identifikuotų atakų skaičius bei atminties, reikalingos paketų analizei, sąnaudos, kadangi paketų laiko langas, kurį stebi Snort išauga iki 600s [39]. Tai gali padėti aptikti ilgai besitęsiančius prievadų skenavimus, tačiau jautrumo padidinimas iki maksimalaus nepadėjo aptikti skenavimo, siunčiamo iš kelių šaltinių aibei taikinių.

Darbo metu sukurto metodo įvertinimui papildomai buvo pabandyta išsiaiškinti, kiek šaltinių turi atlikti skenavimą ir kiek taikinių mažiausiai turi būti skenuojama, kad ši veikla būtų pastebėta. Buvo pradėta nuo vieno šaltinio atliekamo skenavimo, tačiau šis scenarijus nėra nagrinėjamas pasiūlyto metodo. Toliau palaipsniui didinant skenavimą atliekančių šaltinių skaičių ir taikinių skaičių, buvo gauti rezultatai, parodyti 7 paveiksle.



7 pav. Prisijungimų kiekis reikalingas kiekvienam šaltiniui identifikuoti

Iš grafiko matyti, kad nepriklausomai nuo šaltinių skaičiaus šiam metodui reikia mažiausiai 4 prisijungimo mėginimų iš kiekvieno šaltinio norint aptikti paskirstytą skenavimą. Ši charakteristika paveldėta iš TRW algoritmo, kurio pagrindu paremtas sukurtas metodas. Mažinti šį slenkstį būtų galima,

tačiau tada labai padidėtų neteisingų atakos identifikavimų kiekis. Visgi metodas aptinka paskirstytas atakas greičiau, nei Snort, kuriam reikia bent 5 prisijungimo mėginimų. Be to Snort nesugeba identifikuoti vieningos *„daug su daug“* tipo paskirstyto prievadų skenavimo atakos prigimties, todėl šiuo atveju jis duoda netikslius įspėjimus tinklo administratoriams.

3.3. Kompiuterio resursų panaudojimo efektyvumo tyrimas

Siekiant ištirti disko vietos, reikalingos saugoti NetFlow tinklo srauto įrašų informaciją ir palyginti duomenis su pilnos duomenų apsikeitimo kopijos saugojimui reikalingais resursais buvo panaudoti 30 minučių KTU tinklo srauto duomenų rinkiniai. Statistiniai duomenys pateikti lentelėje 6.

Lentelė 8. Tinklo srauto duomenų srautų lyginamoji lentelė

Tinklo srauto dydis *(SD), GB*	**Paketų skaičius *(PS)*, milijonais**	**NetFlow įrašų skaičius *(NS)***	**NetFlow įrašų failo dydis *(ND), MB***	***SD / ND***	***PS / NS***
69,5	85	2814995	182	467	30
71	86	2664525	186	462	32
70	83,4	2763983	176	474	30
67,6	77,9	2482061	159	490	31
73,1	83,9	2904923	187	449	28
76	86	2951683	190	453	29
427,7	**502,2**	**16582170**	**1080**		

Vidutinis duomenų mainų tinkle sugeneruojamas srautas per 5 minutes nagrinėjamu laikotarpiu yra 71,2 GB. Tuo tarpu saugoti maršrutizatorių registruojamus išvestinius duomenis apie duomenų srautus NetFlow formatu vidutiniškai reikalauja 180 MB disko vietos. Tai yra vidutiniškai 465 kartus mažiau vietos, sunaudojamos to paties laiko intervalo tinklo duomenų srauto informacijai saugoti, lyginant su pilna duomenų kopija, kuri užimtų 71,2 GB.

Vidutinis, per 5 minutes registruojamų tinklo paketų skaičius duotuoju laiko momentu yra 83 000 000, kurie saugomi 2763695 NetFlow srauto įrašuose. Tai yra 30 kartų mažiau duomenų rinkinių, kuriuos reikėtų analizuoti.

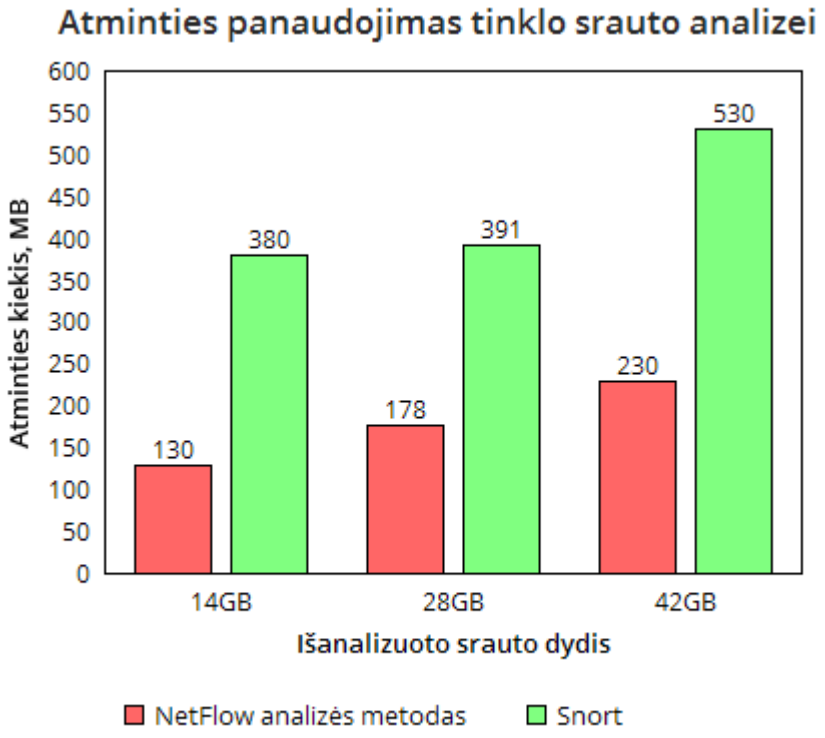
Palyginus NetFlow srauto įrašus ir pilną srauto duomenų kopiją matome didžiulį sutaupomos saugojimo vietos kiekį. Taip pat, kadangi NetFlow panašius paketus sujungia į vieną įrašą, gaunamas mažesnis įrašų skaičius, todėl mažėja analizuojamos informacijos imtis ir skaičiavimo resursų poreikis.

Norint palyginti įrenginio resursų poreikį prievadų skenavimo atakų aptikimui, naudojant sukurtą metodą ir Snort įsilaužimų aptikimo sistemą, buvo panaudoti 15 minučių laikotarpio CAIDA anonimizuoti duomenų rinkiniai. Resursų panaudojimo tyrimo procesas buvo vykdomas tris kartus, palaipsniui didinant informacijos, kurią reikia išanalizuoti, kiekį. Tyrimo metu buvo siekiama gauti ne kokybines, o kiekybines resursų panaudojimo charakteristikas. Pabrėžtina, kad šiuose duomenų rinkiniuose nėra saugomi pernešami duomenys, o tik paketų antraštės. Statistinė informacija apie naudotus duomenų rinkinius pateikta 7 lentelėje.

Lentelė 9. Naudotų tinklo srauto duomenų rinkinių informacija

Tinklo srauto dydis *(SD), GB*	**Paketų skaičius *(PS),* milijonais**	**NetFlow įrašų skaičius *(NS)***	**NetFlow įrašų failo dydis *(ND), MB***
14	18,6	2995067	155
14	19,2	3104763	160
14	20,3	3185782	161

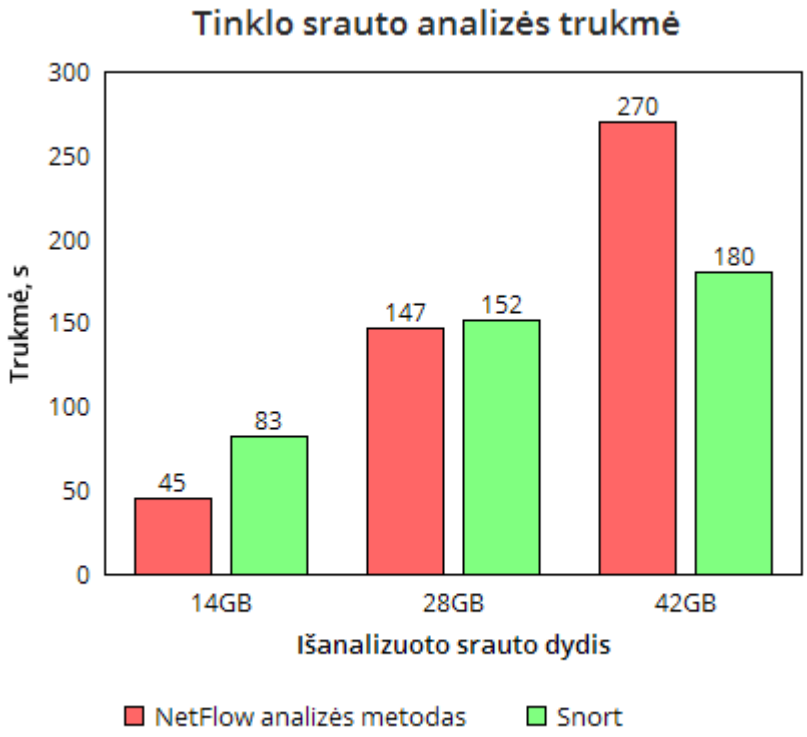
Pirmiausia buvo ištirtas reikalingos atminties kiekis analizuojant trijų skirtingų dydžių tinklo srauto duomenų rinkinius. 8 paveiksle galima pastebėti, kad visais trimis atvejais šiame darbe pasiūlytas metodas naudojo bent du kartus mažiau įrenginio atminties duomenų analizei nei Snort įsilaužimų aptikimo sistemą. Tokį rezultatą galima paaiškinti didesniu duomenų kiekių, kurį Snort turi išanalizuoti bei daugybe Snort modulių, kurie dalyvauja paketų informacijos apdorojime.



8 pav. Tinklo srauto analizei reikalingos operatyviosios atminties palyginimas

Tiriant tinklo srauto analizės trukmę buvo gauti rezultatai sukurto prievadų skenavimo aptikimo metodo nenaudai, matomi 9 paveiksle. Analizuojant sąlyginai mažą tinklo srauto informacijos kiekį, matomas pasiūlyto metodo pranašumas, tačiau informacijos kiekiui augant, Snort rodo geresnį rezultatą. Šiuos rezultatus labai paprasta paaiškinti: siūlomas aptikimo metodas buvo realizuotas naudojant PHP programavimo kalbą, kuri nėra skirta didelių informacijos kiekių apdorojimui ir nepasižymi našumu, kurį siūlo C programavimo kalba, kuria parašyta Snort įsilaužimų aptikimo sistema. Remiantis programavimo kalbų našumo palyginimu [45] galime daryti išvadą, jog realizavus aptikimo metodą

žemesnio lygio programavimo kalba, tokia kaip C++ ar C, tinklo srauto analizės trukmė pastebimai sutrumpėtų.



9 pav. Tinklo srauto analizės trukmės palyginimas

3.4. Tyrimo išvados

Eksperimentiškai ištyrus pasiūlyto skenavimų aptikimo metodo efektyvumą pastebėtas pagerėjimas aptinkant atakas turinčias paskirstytą prigimti, kai bandoma sukompromituoti ne vieną o kelis taikinius kuo mažiau patraukiant dėmesį. NetFlow tinklo srautų informacijos panaudojimas vietoj įprastinės gilios tinklo paketų analizės sumažino reikalingos tinklo srauto analizei įrenginio atminties kiekį. Norint pagerinti informacijos analizavimo greitį reikėtų metodą realizuoti žemesnio lygio į didelio duomenų kiekio apdorojimą orientuota programavimo kalba.

4. IŠVADOS

1. Sudėtingus vykdomos veiklos nuslėpimo metodus kuriantys ir taikantys kvalifikuoti ir suinteresuoti piktavaliai turi susilaukti ypatingo dėmesio, kaip keliantys didelę grėsmę, o prieš juos turi būti taikomos tokio pat aukšto lygio aptikimo priemonės.
2. Magistrinio darbo metu buvo išanalizuota paskirstyto prievadų skenavimo atakų aptikimo problema ir nustatyta, kad dažnai tokios atakos vykdomos neviršijant klasikinių prievadų skenavimo atakas aptinkančių įrankių, paremtų nesėkmingų prisijungimų per laiko vienetą aptikimu, jautrumo lygio ir lieka nepastebėtos.
3. Snort įsilaužimų aptikimo sistema naudodamasi specializuotais moduliais aptinka didelę dalį prievadų skenavimų, tačiau ieškant ilgai besitęsiančių atakų atlikti pilną duomenų srauto analizę darosi sudėtinga dėl didelio informacijos kiekio.
4. Darbo metu buvo pasiūlytas paskirstyto prievadų skenavimo, vykdomo IP adresų esančių arti vienas kito adresų erdvėje aptikimo metodas, paremtas plačiai naudojamu vieno šaltinio vykdomų skenavimų aptikimo algoritmu, atliekant NetFlow tinklo srauto charakteristikų informacijos analizę.
5. Eksperimentiškai palyginus aptikimo efektyvumą, paaiškėjo, kad metodas geba aptikti ir identifikuoti *„daug su daug“* tipo paskirstytą prievadų skenavimą pastebėjus mažiau bandymų nei Snort jautrumo lygis ir naudojant mažiau įrenginio atminties. Dėl netinkamai parinktos programavimo kalbos metodo realizacijai nepavyko pagerinti srauto analizės spartos.

5. LITERATŪRA

[1] RFC 2828 „Internet Security Glossary“. [Tinkle]. Pasiekiamas: http://tools.ietf.org/html/rfc2828. [Kreiptasi 20 Balandžio 2015].

[2] C. Bailey, L. Chris, R. E. Silenok, „Detection and Characterization of Port Scan Attacks“. [Tinkle]. Pasiekiamas: http://cseweb.ucsd.edu/~clbailey/PortScans.pdf . [Kreiptasi 20 Balandžio 2015].

[3] H. Singh, „Distributed Port Scanning Detection“, 2009. [Tinkle]. Pasiekiamas: http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1141&context=etd_projects. [Kreiptasi 20 Balandžio 2015].

[4] Prabhaker Mateti, „Port scanning“, 2012. [Tinkle]. Pasiekiamas: http://cecs.wright.edu/~pmateti/Courses/4420/Probing/index.html. [Kreiptasi 23 Balandžio 2015].

[5] M. H. Bhuyan1, D. K. Bhattacharyya, J. K. Kalita, „Surveying Port Scans and Their Detection Methodologies“, 2010. [Tinkle]. Pasiekiamas: http://www.cs.uccs.edu/~jkalita/papers/2011/BhuyanMonowarComputerJournal.pdf. [Kreiptasi 23 Balandžio 2015].

[6] Lawrence Abrams, „TCP and UDP Ports Explained“, [Tinkle]. Pasiekiamas: http://www.bleepingcomputer.com/tutorials/tcp-and-udp-ports-explained/. [Kreiptasi 23 Balandžio 2015].

[7] Roger Christopher , „Port Scanning Techniques and the Defense Against Them“, 2001. [Tinkle]. Pasiekamias: http://www.sans.org/reading-room/whitepapers/auditing/port-scanning-techniques-defense-70. [Kreiptasi 23 Balandžio 2015].

[8] Fyodor, „The Art of Port Scanning“, 1997. [Tinkle]. Pasiekiamas: http://hamsa.cs.northwestern.edu/media/readings/port_scanning.pdf. [Kreiptasi 23 Balandžio 2015].

[9] Nmap Reference Guide: Port Scanning Techniques. [Tinkle]. Pasiekiamas: http://nmap.org/book/man-port-scanning-techniques.html. [Kreiptasi 23 Balandžio 2015].

[10] Tariq Ahamad Ahanger, „Port Scan - A Security Concern“, 2014. [Tinkle]. Pasiekiamas: http://www.ijeit.com/Vol%203/Issue%2010/IJEIT1412201404_46.pdf. [Kreiptasi 23 Balandžio 2015].

[11] S. Staniford, J. A. Hoagland., J. M. McAlerney, „Practical Automated Detection of Stealthy Portscans“. [Tinkle]. Pasiekiams: http://cs.fit.edu/~pkc/id/related/staniford-jcs02.pdf. [Kreiptasi 24 Balandžio 2015].

[12] „Footprinting and Scanning“, 117 p., 118p. [Tinkle]. Pasiekiamas: http://ptgmedia.pearsoncmg.com/images/9780789735317/samplechapter/0789735318_CH03.pdf. [Kreiptasi 24 Balandžio 2015].

[13] Nmap documentation. [Tinkle]. Pasiekiamas: https://nmap.org/book/man.html. [Kreiptasi 10 Sausio 2015].

[14] DNmap documentation. [Tinkle]. Pasiekiamas: http://sourceforge.net/p/dnmap/wiki/Home/. [Kreiptasi 24 Balandžio 2015].

[15] Z. Durumeric, E. Wustrow, J. A. Halderman, „ZMap: Fast Internet-Wide Scanning and its Security Applications“, 2013. [Tinkle]. Pasiekiamas: https://zmap.io/paper.pdf. [Kreiptasi 24 Balandžio 2015].

[16] D. Myers, E. Foo, K. Radke, „ Internet-wide Scanning Taxonomy and Framework“, 2015. [Tinkle]. Pasiekiamas: http://crpit.com/confpapers/CRPITV161Myers.pdf. [Kreiptasi 26 Balandžio 2015].

[17] Chakchai So-In , „A Survey of Network Traffic Monitoring and Analysis Tools“. [Tinkle]. Pasiekiamas: http://www.cse.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors3.pdf. [Kreiptasi 26 Balandžio 2015].

[18] Network Instruments, „SNMP Monitoring: One Critical Component to Network Management“. [Tinkle]. Pasiekiamas: http://www.gordion.de/uploads/media/SNMP_WP.pdf. [Kreiptasi 29 Balandžio 2015].

[19] W. Fuertes, P. Zambrano, M. Sánchez, P. Gamboa, „Alternative Engine to Detect and Block Port Scan Attacks using Virtual Network Environments“, 2011. [Tinkle]. Pasiekiamas: http://paper.ijcsns.org/07_book/201111/20111103.pdf. [Kreiptasi 29 Balandžio 2015].

[20] SANS Institute, „Host- vs. Network-Based Intrusion Detection Systems“, 2005.[Tinkle]. Pasiekiamas: http://www.giac.org/paper/gsec/1377/host-vs-network-based-intrusion-detection-systems/102574. [Kreiptasi 29 Balandžio 2015].

[21] Monowar Hussain Bhuyan , D. K. Bhattacharyya, J. K. Kalita, „Survey on Incremental Approaches for Network Anomaly Detection“, 2011. [Tinkle]. Pasiekiamas: http://arxiv.org/ftp/arxiv/papers/1211/1211.4493.pdf. [Kreiptasi Gegužės 3 2015].

[22] http://www.tripwire.com/state-of-security/vulnerability-management/distributed-nmap-port-scanning-dnmap-megacluster/. [Kreiptasi Gegužės 3 2015].

[23] Cisco, „NetFlow Services Solutions Guide“. [Tinkle]. Pasiekiamas: http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.html. [Kreiptasi Gegužės 3 2015].

[24] RFC 3954 „Cisco Systems NetFlow Services Export Version 9“, 2004. [Tinkle]. Pasiekiamas: https://www.ietf.org/rfc/rfc3954.txt. [Kreiptasi Gegužės 3 2015].

[25] Vytautas Krakauskas, „Kompiuterių tinklo srautų anomalijų aptikimo metodai“, 2006. [Tinkle]. Pasiekiamas: http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2006~D_20060603_144221-31628/DS.005.0.02.ETD. [Kreiptasi Gegužės 3 2015].

[26] Phrack Magazine, „Distributed Information Gathering“, 1999. [Tinkle]. Pasiekiamas: http://phrack.org/issues/55/9.html. [Kreiptasi Gegužės 3 2015].

[27] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle, „A graph based intrusion detection system for large networks“. [Tinkle]. Pasiekiamas: http://seclab.cs.ucdavis.edu/papers/nissc96.pdf. [Kreiptasi Gegužės 3 2015].

[28] J. Green, D. Marchette, „Analysis Techniques for Detecting Coordinated Attacks and Probes“, 1999. [Tinkle]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.6598&rep=rep1&type=pdf. [Kreiptasi Gegužės 3 2015].

[29] V. Yegneswaran, P. Barford, J. Ullrich, „Internet Intrusions: Global Characteristics and Prevalence“, 2003. [Tinkle]. Pasiekiamas: http://pages.cs.wisc.edu/~pb/dshield_paper.pdf. [Kreiptasi Gegužės 12 2015].

[30] S. Robertson, E. V. Siegel, M. Miller, S. J. Stolfo, „Surveillance Detection in High Bandwidth Environments“, 2003. [Tinkle]. Pasiekiamas:

http://ids.cs.columbia.edu/sites/default/files/SD-DiscexIII.pdf. [Kreiptasi Gegužės 12 2015].

[31] G. Conti, K. Abdullah, „Passive Visual Fingerprinting of Network Attack Tools“, 2004. [Tinkle]. Pasiekiamas: http://www.rumint.org/gregconti/publications/20040617_VizSec_Fingerprinting.pdf. [Kreiptasi Gegužės 12 2015].

[32] J. Jung, V. Paxson, A. W. Berger, H. Balakrishnan, „Fast portscan detection using sequential hypothesis testing“, 2004. [Tinkle]. Pasiekiamas: http://nms.csail.mit.edu/papers/portscan-oakland04.pdf. [Kreiptasi Gegužės 12 2015].

[33] C. Gates, „Coordinated Scan Detection“, 2006. [Tinkle]. Pasiekiamas: https://www.isoc.org/isoc/conferences/ndss/09/pdf/09.pdf. [Kreiptasi Gegužės 14 2015].

[34] https://www.snort.org. [Kreiptasi Gegužės 14 2015].

[35] D. Whyte, „Network Scanning Detection Strategies for Enterprise Networks“, 2008. [Tinkle]. Pasiekiamas: https://www.ccsl.carleton.ca/people/theses/Whyte_PhD_Thesis_08.pdf.

[Kreiptasi Gegužės 15 2015].

[36] Himanshu Singh, „Distributed Port Scanning Detection“, 2009.[Tinkle]. Pasiekiamas: http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1141&context=etd_projects.

[Kreiptasi Gegužės 15 2015].

[37] Solarwinds, „NetFlow Tips and Tricks“. [Tinkle]. Pasiekiamas: http://cdn.swcdn.net/creative/v9.6/pdf/Whitepapers/netflow_tips_and_tricks_0613.pdf.

[Kreiptasi Gegužės 18 2015].

[38] C. Gates, J. McNutt, J. B. Kadane, M. Kellner, „Detecting Scans at the ISP Level“, 2006. [Tinkle]. Pasiekiamas: http://www.sei.cmu.edu/reports/06tr005.pdf. [Kreiptasi Gegužės 15 2015].

[39] sfPortscan manual, [Tinkle]. Pasiekiamas: http://manual.snort.org/node78.html. [Kreiptasi Gegužės 18 2015].

[40] „Elektroninių ryšių įstatymas“. [Tinkle]. Pasiekiamas: http://www3.lrs.lt/pls/inter3/dokpaieska.showdoc_l?p_id=463812. [Kreiptasi Gegužės 18 2015].

[41] Farha Ali, „IP Spoofing“. [Tinkle]. Pasiekiamas: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-4/104_ip-spoofing.html. [Kreiptasi 16 gegužės 2015]

[42] S. Panjwani; S. Tan, K. M. Jarrin, M. Cukier, „ An Experimental Evaluation to Determine if Port Scans Are Precursors to an Attack“, 2005. [Tinkle]. Pasiekiamas: https://sm.asisonline.org/ASIS%20SM%20Documents/port_scans0306.pdf. [Kreiptasi 17 gegužės 2015]

[43] V. Ališauskaitė, „ Kompiuterių tinklų saugos modelio sudarymas“, 2008. [Tinkle]. Pasiekiamas: http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2008~D_20080128_103419-62847/DS.005.0.01.ETD. [Kreiptasi 17 gegužės 2015]

[44] The CAIDA Anonymized Internet Traces 2015 Dataset. [Tinkle]. Pasiekiamas: http://www.caida.org/data/passive/passive_2015_dataset.xml. [Kreiptasi 18 gegužės 2015]

[45] The HipHop Virtual Machine. [Tinkle]. Pasiekiamas: http://hhvm.com/. [Kreiptasi 18 gegužės 2015]

[46] Ivan Zahariev, „C++ vs. Python vs. Perl vs. PHP performance benchmark“, 2010. [Tinkle]. Pasiekiamas: http://blog.famzah.net/2010/07/01/cpp-vs-python-vs-perl-vs-php-performance-benchmark/. [Kreiptasi 18 gegužės 2015]