KAUNO KOLEGIJA

TECHNOLOGIJŲ FAKULTETAS

INFORMATIKOS IR MEDIJŲ TECHNOLOGIJŲ KATEDRA

Ainoras Petraška

SIMULIACINĖ APLINKA AKTYVIOS KIBERNETINĖS ŽVALGYBOS 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
Prievadai (angl. *Port*)	Prievadai yra numeruoti ryšio taškai, per kuriuos kompiuteris gali bendrauti su kitais įrenginiais lokaliame ar viešame tinkle.	(Pittman, Jason, 2023)
Kibernetinė žvalgyba (angl. *cyber reconnaissance*)	Tai procesas, kurio metu kibernetiniai operatoriai renka informaciją apie kompiuterines sistemas, tinklos ar kitus elektroninius išteklius, siekdami parengti potencialias atakas.	(Mazurczyk, Caviglione 2021)
Enumeracija (angl. *Enumeration*)	Tai yra procesas, kuriuo identifikuojami ir išvardinami visi galimi objektai, reikšmės ar kategorijos. Šis terminas dažnai naudojamas kibernetinio saugumo srityje.	(Agghey, Mwinuka, Pandhare, Dida, Ndibwile, 2023)
Pažeidimų skenavimas (angl. *Vulnerability scanning*)	Tai yra procesas, kurio kibernetiniai saugumo specialistai arba įrankiai sistemingai tikrina kompiuterines sistemas ar tinklas, siekdami nustatyti jų pažeidimus ar spragas.	(Jinfeng, 2020)
Prievadų skenavimas (angl. *Port Scanning*)	Tai yra procesas, kurio metu kibernetiniai operatoriai bando nustatyti, kokie tinklo prievadai yra atviri.	(Pittman, Jason ,2023)
Simuliacija (angl. *Simulation*)	Tai yra procesas arba veiksmas, kuriuo modeliuojama realaus pasaulio situacija, sistema ar procesas, tam kad būtų galima tyrinėti jo veikimą.	(Scherb, Heitz, Grimberg, Grieder, Maurer, 2023)
Kibernetinė sauga (angl. *Cyber security*)	Tai yra priemonės, procedūros ir technologijos, skirtos apsaugoti kompiuterines sistemas, tinklus ir programas nuo kibernetinių grėsmių, atakų ir neigiamų poveikių.	(Aktuğ, Yilmaz, Akin, 2023)
Operacinės sistemos (angl. *Operating system*)	Tai yra programinė įranga, kuri baldo ir koordinuoja kompiuterio ar kitų įrenginių veikimą. Ji suteikia sąsają tarp vartotojo ir kompiuterio aparatūros, bei programinės įrangos.	(AL-Khorazmi, 2023)
Nmap (angl. *Network mapper*)	Galinga atviro kodo įrankio programa skirta tinklo skenavimui ir tinklo įrenginių aptikimui, bei jų audito atlikimui.	(„*Nmap*“ dokumentacija)
„*Wireshark*“ Ir „*Tshark*“	Tai yra tinklo analizės įrankiai, kurie leidžia stebėti ir analizuot tinklo duomenų srautą realiu laiku arba išsaugoti jį tolimesniam tyrimui.	(„*Wireshark*“ dokumentacija)
Pažeidimų srautas (angl. *Incident stream*)	Tai yra terminas apibūdinantis nuolatinius įvykius arba incidentus, susijusius su saugumo pažeidimais ar anomalijomis.	(Naseer, Naseer, Ahmad, Maynard, Masood Siddiqui, 2023)
„*GNS3*” (angl. *Graphical network simulator – 3*)	Tai yra atviro kodo, grafinę sąsaja paremtas tinklo simuliacijos įrankis, naudojamas kurti ir testuoti įvairias tinklo topologijas, bei konfigūracijas.	(„*GNS3*” dokumentacija)
„*VMware*“	Tai yra virtualizacijos įrankis, skirtas tiek asmeniniam, tiek verslo naudojimui.	(„*VMware*“ dokumentacija)

SANTRAUKA

Autorius Ainoras Petraška. *Simuliacinė aplinka aktyvios kibernetinės žvalgybos tyrimams*. Baigiamasis darbas. Vadovas Paulius Baltrušaitis. Kauno kolegija, Technologijų fakultetas, Informatikos ir medijų technologijų katedra. Kaunas, 2024, 60 psl.

Reikšminiai žodžiai: Tinklų naudojimas, kibernetinė žvalgyba, srauto generavimas, srauto analizė, grėsmių identifikavimas, srauto pažeidimai, simuliacinė aplinka.

Šiuolaikinėje informacinių technologijų eroje, vis didėja skaitmeninės paslaugos ir tinklų naudojimo mastas. Simuliacinė aplinka aktyvios kibernetinės žvalgybos tyrimams tampa svarbiu aspektu, norint užtikrinti tinklo, programų ar sistemų saugumą. Efektyvi simuliacinė aplinka aktyvios kibernetinės žvalgybos tyrimams sistema leidžia greitai identifikuoti potencialias grėsmes ir užkirsti kelia joms. Todėl ši tema yra svarbi norint užtikrinti saugesnes ir patikimesnes interneto aplinkas. Baigiamojo darbo tikslas – Išnagrinėti ir patobulinti esamus srauto generavimo ir analizės metodus ir rasti būdą, kaip efektyviau reaguoti į srauto pažeidimus. Tyrimo metu naudojami įvairūs eksperimentiniai ir analitiniai metodai, siekiant gauti patikimus rezultatus, kurių pagalba lengviau suprasime pažeidimų srauto generavimo ir analizes procesų esmę. Baigiamojo darbo problema - yra pastovus trūkumas efektyvių pažeidimų srauto analizės ir generavimo būdų sistemose. Todėl yra svarbu tobulinti metodus, kuriais reaguojame ir atpažįstame nusikalstama kibernetinę veiklą. Apibendrinant, šis baigiamasis darbas suteikia įžvalgą į pažeidimų srauto generavimo ir analizės svarbą. Išnagrinėjus tyrimo rezultatus, galima drąsiai teigti, kad pažeidimų srauto valdymas yra būtinas, norint užtikrinti tinklų saugumą ir stabilumą. Tai yra svarbi tema visai visuomenei.

SUMMARY

Author Ainoras Petraška. *Simulation Environment for Research of Active Reconnaissance*. Graduation Thesis. Supervisor Paulius Baltrušaitis. Kauno kolegija HEI, Faculty of Technologies, Department of Informatics and Media Technologies. Kaunas, 2024, 60 pages.

Keywords: Network usage, active reconnaissance, traffic generation, traffic analysis, identification of potential threats, traffic violations, simulation environment.

In the modern era of information technology, the scale of digital services and network usage is continuously increasing. A simulation environment for research of active reconnaissance becomes a crucial aspect in ensuring the security of networks, applications and/or systems. An effective breach traffic generation and analysis systems allows for the rapid identification of potential threats and preemptive action against them. Therefore, this topic is essential for ensuring safer and more reliable internet environments. The objective of this thesis is to examine and improve existing methods of traffic generation and analysis, and to find more effective ways to respond to the traffic breach. Various experimental and analytical methods are employed during the research to obtain reliable results, which will facilitate a better understanding of the essence of traffic breach generation and analysis processes. The problem addressed in this thesis is the persistent lack of effective methods for traffic breach analysis and generation systems. Therefore, it is crucial to enhance the methods by which we respond to and identify criminal cyber activities. In summary, this thesis provides insight into the importance of breach traffic generation and analysis. Upon reviewing the research results, it can be confidently stated that managing breach traffic is essential for ensuring the security and stability of networks. This is a significant topic for society as a whole.

ĮVADAS

Simuliacinė aplinka aktyvios kibernetinės žvalgybos tyrimams tampa vis svarbesniu kibernetinės saugos aspektu, nes kibernetinis nusikalstamumas visad kyla, dėl pastovaus kibernetinių nusikaltėlių tobulėjimo. Ši analizė leidžia suprasti naujas grėsmes, spragas ir tendencijas, o tai padės užkirsti kelią ateityje kylančioms infrastruktūros, bei duomenų saugumo pažeidimams. Generuojant ir analizuojant pažeidimų srautus galima įgyvendinti efektyvias strategijas, kurios ateityje užkirs kelią kibernetinio saugumo spragoms. Taigi, pažeidimų srauto generavimas, bei analizė yra esminis įrankis padedantis kovoti prieš kibernetinius nusikaltimus.

Darbo problema – pastovus trūkumas kibernetinių pažeidimų duomenų, kad būtų galima pritaikyti mašininį mokymąsi efektyviam pažeidimų identifikavimui ir prevencijai. Todėl yra svarbu tobulinti metodus, kuriais generuojame ir analizuojame pažeidimų srautą.

Darbo objektas – Aktyviosios kibernetinės žvalgybos simuliacija.

Darbo tikslas – simuliacinės aplinkos sukūrimas aktyvios kibernetinės žvalgybos duomenų generavimui ir analizei.

Darbo uždaviniai:

1. Išanalizuoti kibernetinių žvalgybinių atakų procesu ir naudojamas priemones.
2. Įvertinti virtualių aplinkų galimybes, šiuos procesu simuliuoti ir rinkti duomenis.
3. Suprojektuoti simuliacinę platformą aktyviosios kibernetinės žvalgybos simuliavimui ir duomenų srauto fiksavimui.
4. Platformą išbandyti atliekant aktyviosios kibernetinės žvalgybos simuliacijas.

Darbo metodai – nagrinėjant šią temą atliksiu išsamią mokslinės literatūros analizę apie pažeidimų srauto generavimą ir analizę. Tai apims platų mokslinių straipsnių, knygų, konferencinių pranešimų ir kitų šaltinių analizę.

Darbo struktūra apims šiuos elementus:

1. Įvadas – pristato darbo temą, tikslus ir svarbiausias idėjas
2. Analitinė dalis – ši dalis skirta duomenų ir mokslinės literatūros analizei, bei interpretacijai.
3. Projektavimas – Šioje dalyje buvo suprojektuota simuliacinė aplinka atlikti kibernetiniai žvalgybai.
4. Simuliacija – tai yra kūrybinis procesas, kuriuo modeliuojama tam tikra veikla kompiuteriniu būdu.
5. Išvados – tai pagrindinės mintys ir rezultatai, kuriuos padariau remdamasis savo analize ir tyrimu.

1. ANALITINĖ DALIS

Pažeidimų srauto analizė ir generavimas yra vieni iš esminių kibernetinės žvalgybos procesų etapų. Šiame skyriuje aptarsime kas yra kibernetinė žvalgyba ir kaip pasinaudojant aktyvia ir pasyvia žvalgyba sugeneruojamas ir analizuojamas pažeidimų srautas, bei kaip gauti duomenys interpretuojami, kokiais įrankiais naudojamasi, kaip aptikti „Žvalgus“ ir užkirsti kelią jų veiksmams.

1.1. Kibernetinė žvalgyba

Kibernetinė žvalgyba – procesas, kurio pagalba yra stebimas ir analizuojamas internetinėje erdvėje vykstantis informacijos srautas siekiant aptikti galimas tinklo silpnybes ar spragas kuriomis galėtų pasinaudoti kenkėjai. Šis procesas apima duomenų rinkimą, analizę, informaciją, vertinimą ir sprendimų priėmimą siekiant įsilaužti į sistemas, tinklus ir kitus su internetu susijusius išteklius. Pagrindinis kibernetinės žvalgybos tikslas yra paveikti tinklą arba asmenį neigiamai, siekiant pažeisti žmogaus ar įmonės duomenų saugumą, prieinamumą, vientisumą ir konfidencialumą. Tačiau svarbu paminėti, kad kibernetinė žvalgyba gali būti naudojama ir gerais tikslais, jei tai atliekama savo tinkle, įmonėje, infrastruktūroje siekiant išsiaiškinti savo tinklo spragas ir trūkumus, siekiant apsisaugoti nuo nusikalstamos kibernetinės veiklos. Kibernetinė žvalgyba yra skirstoma į pasyviąja ir aktyviąją (Mazurczyk, Caviglione, 2021).

1.1.1. Pasyvi žvalgyba

Pasyvi žvalgyba – duomenų rinkimas, stebėjimas be įsikišimo į srautą. Įrankiai kaip „*Wireshark*“ arba „*tcpdump*“ padeda stebėti tinklo srautą. Remiantis gautais duomenimis galima identifikuoti pasikeitimus, pastebėti spragas ir pagal tai pradėti spręsti kokie pirmi žingsniai bus įsilaužiant. Naudojant pasyviąją kibernetinę žvalgybą galima gauti informaciją apie tinklų architektūrą ir tai panaudoti įsilaužiant. Taip pat kibernetiniai nusikaltėliai gali panaudoti pasyviąja žvalgyba siekdami išvengti aptikimo, nes priešingai nei aktyvioji žvalgyba, naudojant tam tikrus pasyvius metodus nepaliekama pėdsakų. Vis dėl to svarbu paminėti, kad kibernetinė žvalgyba gali būti naudinga netik kibernetiniams nusikaltėliams, bet ir geros valios žmonėm, siekiantiems užtikrinti kibernetinį saugumą. Tad svarbu, kad organizacijos naudotų veiksmingas saugumo strategijas ir apsisaugotų nuo bet kokių potencialių grėsmių.

1.1.2. Aktyvi žvalgyba

Aktyvioji žvalgyba – yra procesas, kurio metu įgyvendinami tiesioginiai veiksmai siekiant įsilaužti į tinklą ir greitai sukelti grėsmę asmenims ar organizacijomis. Priešingai pasyviajai žvalgybai, aktyvioji žvalgyba įsikiša tiesiogiai į kibernetinę erdvę. Aktyvioji žvalgyba yra labai svarbi kibernetinio saugumo strategijos dalis. Keletas būdų kaip aktyvioji žvalgyba gali pakenkti žmogui ar organizacijai:

- Privatumo pažeidimas – gali sukelti žalą asmeninei laisvei ir saugumui, surinkus duomenis apie asmenį ar organizaciją, kurios prieštarauja privatumo principams.
- Neteisėti stebėjimai ir atakos – kibernetinę žvalgybą galima panaudoti neteisėtiems stebėjimams ar atakoms, kurios vėliau gali sukelti ar destabilizuoti sistemą bei jos veiklą savo puolimais.
- Sukčiavimai ir įsilaužimai – kenkėjai pasinaudoję aktyviąja žvalgyba gali išgauti informacija iš asmenų ir tai vėliau panaudoti kaip savo tikslams.
- Duomenų manipuliavimas – aktyviosios žvalgybos pagalba, žvalgai gali manipuliuoti svarbius duomenis, juos pakeisti, ištrinti, kas galiausiai gali turėti labai daugi neigiamų padarinių.
- Sistemos pažeidžiamumo išnaudojimas – galima išsiaiškinti sistemų versijų spragas ar silpnybes ir jas naudoti, kas galiausiai leidžia įsibrauti į tinklą ir atlikti kenksmingus veiksmus.

Svarbu paminėti, kad aktyvioji kibernetinė žvalgyba tinkle palieka pėdsakus, naudojant tinkamomis priemonėmis galima laiku jai užkirsti tam kelią.

1.1.3. Apibendrinimas

Kibernetinė žvalgyba tai yra procesas, kuriuo stebimas ir analizuojamas internetinėje erdvėje vykstantis informacijos srautas. Kibernetinės žvalgybos tikslas – aptikti galimas tinklo silpnybes ar spragas, kurias gali panaudoti kenkėjai ir taip paveikti tinklą ar asmenį neigiamai pažeisdami duomenų konfidencialumą, vientisumą ir prieinamumą.

Kibernetinė žvalgyba skirstoma į pasyviąja ir aktyviąją. Pasyvioji žvalgyba apima duomenų stebėjimą ir rinkimą be įsikišimo į srautą. O aktyvioji žvalgyba, priešingai pasyviajai, įsikiša tiesiogiai į kibernetinę erdvę. Aktyvioji žvalgyba palieka pėdsakus, naudojant tinkamas priemones galima užkirsti tam kelią.

1.2. Kibernetinio saugumo žvalgai

Kibernetinio saugumo žvalgai – tai yra asmenys arba grupės, kurios dažnai naudoja savo žinias siekiant nelegaliai gauti duomenis iš kitų asmenų ar organizacijų. Juos galima pavadinti kibernetiniais nusikaltėliais. Žvalgų dažnai naudojamos technikos yra:

1. Enumeracija (angl. *Enumeration*) – kenkėjai dažnai naudoja tokius įrankius kaip „*SMBenum*“, „*SNMPwalk*“, kad gautų svarbius duomenis kaip vartotojų prisijungimus, paskyras (Mwinuka, Pandhare, Dida ir Ndibwile, 2021).
2. Pažeidžiamumų skenavimas (angl. *Vulnerability scanning*) – tokių įrankių kaip „*OpenVAS*“ ir „*Nessus*“ pagalba kenkėjai aptinka spragas programinėse įrangose ir nustatymuose. Šis etapas yra labai svarbus norint surasti silpnybes, kurios galėtų būti išnaudojamos (Jinfeng, 2020).
3. Prievadų skenavimas (angl. *Port Scanning*) – pasinaudojus įrankiais „*Nmap*“, „*Masscan*“, „*ZMap*“ galima skenuoti norimą tinklą, taikinį. Tada išsiaiškinti atvirus prievadus ir kokias paslaugas jie naudoja. Ši informacija padeda žvalgui suprasti tinklą ir apsispręsti kaip bandyti įsilaužti.

1.2.1. Enumeracija

Viena iš technikų, kuria gali pasinaudoti kibernetiniai žvalgai yra enumeracija. Tai yra procesas, kuriuo stengiamasi surinkti informacija apie sistemas, jų komponentus ir jų konfigūracijas. Tai apima tokias veiklas, kaip pažeidžiamų prievadų nustatymą, operacinių sistemų ypatybes, failų ar direktorijų sąrašus, peržiūrą.

Enumeracija dažniausiai būna pirmas kibernetinių nusikaltėlių žingsnis, nes tai jiems leidžia greitai ir lengvai surinkti daug tikslios informacijos apie tinklą ir nustatyti jo spragas, bei atakos taikinius. Svarbu paminėti, kad tuo galima pasinaudoti ir etiniai hakeriai. Gali atlikti pažeidimų skenavimą, aplikacijų testavimą ir netgi patys bandyti įsilaužti, kad ateityje užkirstų kelią potencialiems įsilaužimams, taip labiau sustiprinant tinklo apsaugą.

Keli plačiai naudojami enumeracijos įrankiai yra „*SMBenum*“, „*SNMPwalk*“. Šių įrankių paskirtis yra panaši, abu skirti išrinkti informaciją iš sistemų. „*SMBenum*“ naudojamas sistemoms kurios naudoja SMB (angl. *Server Message Block*) protokolą ir leidžia gauti spausdinimo paslaugų sąrašus ir grupių naudotojų duomenis. O „*SNMPwalk*“ naudojamas išgauti sistemų informaciją apie įrenginių konfigūraciją, prieinamumą, tinklo srautą ar kitą svarbia tinklo veiklos informaciją. Šie įrankiai gali būti naudojami tiek teisėtai siekiant atlikti sistemų administravimą ir stebėjimą, tiek neteisėtai, atliekant kibernetinius išpuolius.

1.2.2. Pažeidžiamumų skenavimas

Pažeidžiamumų skenavimas tai procedūra, kuria kibernetiniai nusikaltėliai siekia nustatyti potencialius saugomo pažeidžiamumus kompiuteriniuose tinkluose ar sistemose. Šio proceso metu nusikaltėliai naudoja specializuotus įrankius ar programinę įrangą siekdami rasti pažeidžiamus taškus. Jei tai pavyksta, tai galima panaudoti planuojant įsilaužimą ir vykdyti kibernetines vagystes, kurios gali pažeisti duomenų konfidencialumą.

Keli plačiai pažeidžiamumų skenavimo įrankiai yra „*OpenVAS*“ ir „*Nessus*“. Abu įrankiai yra panašūs, pagrindinė jų paskirtis – aptikti pažeidžiamumus. Įrankis „*OpenVAS*“ yra populiarus dėl savo atviro kodo pobūdžio, kuris leidžia bendruomenei prisidėti prie vystymosi ir atnaujinimo. O „*Nessus*“ yra plačiai naudojamas, nes turi plačia pažeidžiamumų duomenų bazę ir gali atlikti gilias skenavimo procedūras, taip pat suteikia išsamias ataskaitas, bei rekomendacijas dėl pažeidimų pašalinimo. Šie įrankiai yra naudingi tiek etiniams hakeriams, tiek kibernetiniams nusikaltėliams.

1.2.3. Prievadų skenavimas

Trečias būdas dažnai naudojamas kibernetiniai žvalgybai yra – prievadų skenavimas. Šio proceso metu dažnai naudojami įrankiai toki kaip: „*Nmap*“, „*Mascan*“ ir „*Zmap*“. Prievadų skenavimo tikslas yra atrasti atvirus tinklo prievadus ar sistemas. Jei prievadai yra atviri, dažniausiai yra pažeidžiami, kuo ir pasinaudoja įsibrovėliai (Pittman, 2023).

Skenavimo metodai:

TCP Scan – tai vienas iš paprasčiausių būdų nustatyti ar tam tikras prievadas yra atviras ar ne. Svarbu stebėti sąsajų būsenos laukus, kurie nurodo sąsajos būseną. Pavyzdžiui, jei kontrolinis laukas „*S*“ (angl. *Synchronizing*) tai rodo, kad pradedamas naujas ryšys, o jeigu „*A*“ (angl. *Acknowledgment*) tai rodo, kad tinklas atsako į SIN paketą. Atliekant šį skenavimą, kompiuteris bando užmegzti TCP ryšį su kiekvienu prievadu, jei pavyksta reiškia, kad prievadas yra atviras.

UDP Scan – UDP skenavimas naudoja UDP (angl. *User Datagram Protocol*) paketus norint nustatyti ar prievadas yra atviras. Taip pat svarbu stebėti atsakymo būsenos laukus, kurie nurodo kaip serveris reaguoja į skenavimo paketus. Pavyzdžiui, jei gaunamas „*ICMP Host Unreachable*“ pranešimas, galima suprasti, kad iškyla problemų su pasiekiamumu, jei gaunamas „*ICMP Port Unreachable*“ galima manyti, kad prievadas yra uždaras.

SYN Scan – SYN skenavimas yra subtilesnis būdas norint nustatyti ar tam tikras prievadas yra atviras. Šis skenavimo būdas dažnai yra naudojamas norint nustatyti ar konkretus TCP prievadas yra atviras. Jis yra daug greitesnis, kadangi bandoma užmegzti ryši tik su vienu prievadu, Taip pat šis būdas yra efektyvus, nes nereikia užmegzti pilno ryšio, o pakanka tik pradinio prisijungimo.

Skenavimo metu kompiuteris išsiunčia SYN (angl. *synchronize*) paketą į tam tikrą prievadą ir laukia atsakymo. Jei atsakymas gaunamas su SYN/ACK (angl. *synchronize/acknowledge*) paketu, tai rodo, kad prievadas yra atviras. Jei gaunamas RST (angl. *reset*) paketas, tai rodo, kad prievadas yra uždaras.

FIN Scan – šis skenavimo metodas yra pagrįstas FIN (angl. *finish*) TCP paketais. Jis yra siunčiamas tada, kai klientas nori užbaigti ryšį su serveriu. Šio paketo pagalba galima išbandyti ar serveris tinkamai reaguoja į ryšio baigimo užklausas ir ar tinkamai veikia ugniasienės taisyklės. Skenuojantis kompiuteris siunčia FIN paketą i tikslinį prievadą. Jei gaunamas RST paketas, tai reiškia, kad prievadas yra uždarytas. O jei negaunamas atsakymas tai reiškia, kad prievadas yra atviras.

ACK Scan – ACK skenavimas naudoja TCP ACK (angl. *acknowledge*) paketus nustatymui. ACK paketai padeda užtikrinti patikimą duomenų perdavimą tarp kliento ir serverio. Gaunant ACK patvirtinimą žinoma, kad duomenys buvo gauti be jokių problemų. Šiuo skenavimo būdu galima patvirtinti, kad serveris tinkamai reaguoja į patvirtinimus. Panašiai kaip ir su FIN Paketais, jei gaunamas RST paketas, tai rodo, kad prievadas yra uždaras. O jei negaunamas atsakymas, tada reiškia, kad prievadas yra uždaras.

NULL Scan – NULL skenavimo metodas yra paremtas tuščiu TCP paketu, kuriame nėra jokių nustatymų, tai leidžia tyrinėti kaip serveris reaguoja į tokį paketą. Šį paketą kai kurios IDS (intruzijų aptikimo sistemos) gali aptikti kaip potencialų įsilaužimą ir neleisti vykdyti šio skenavimo. Jei išsiuntus NULL paketą gaunamas atsakymas „*ICMP Port Unreachable*“ arba „*TCP RST*“ tai reiškia, kad prievadas yra uždarytas. O jeigu gaunamas „*No response*“ tai gali rodyti, kad prievadas yra atviras.

XMAS Scan – šis skenavimo metodas siunčia TCP paketą, kuriame yra nustatytos visos TCP žymės – FIN, PSH ir URG. Šis būdas dažniausiai naudojamas norint nustatyti ar konkretus TCP prievadas yra atviras, uždaras ar filtruojamas, tai vienas iš būdų išbandyti saugumo spragas arba tinklo konfigūraciją. Kaip ir NULL paketas, šis paketas gali būti aptiktas aptikimo sistemų ir blokuojamas. Jei atgal gaunamas RST paketas, tai rodo, kad prievadas yra uždaras, jei negaunama jokio atsakymo, tada reiškia, kad prievadas yra atviras.

1.2.4. Apibendrinimas

Kibernetinio saugumo žvalgymas yra procesas, kuriuo asmenys ar grupės siekia gauti nelegalią informaciją iš kitų asmenų ar organizacijų. Pagrindinės technikos, kurias jie naudoja yra: Enumeracija, pažeidžiamumų skenavimas ir prievadų skenavimas.

Enumeracija yra informacijos surinkimo procesas apie sistemas, jų komponentus ir konfigūracijas. Šis žingsnis dažnai būna atliekamas pirmas, kadangi galima lengvai ir greitai susirinkti daug informacijos apie tinklą ir nustatyti jo spragas, bei pirmus atakos taikinius.

Pažeidžiamumų skenavimas yra procedūra, kurią kibernetiniai nusikaltėliai naudoja norėdami nustatyti galimus pažeidžiamumus kompiuteriniuose tinkluose. Jie panaudoja specializuotus įrankius ir programinę įranga siekiant atrasti pažeidžiamus tinklo taškus.

Prievadų skenavimas yra dar vienas iš būdas, kuris yra dažnai naudojamas kibernetiniai žvalgybai. Tai leidžia efektyviai aptikti atvirus tinklo prievadus ar sistemas. Specializuoti įrankiai palengvina ši procesą ir taip pat leidžia pasirinkti skirtingus metodus kaip tai atlikti. Keli skenavimo metodai yra TCP, UDP, SYN, FIN, ACK ir NULL skenavimai.

Kibernetinio saugumo žvalgymas yra svarbus procesas, norint sužinoti tinklo silpnybes ir spragas. Tačiau tuo gali pasinaudoti tiek etiniai hakeriai, tiek kibernetiniai nusikaltėliai, tad svarbu kad organizacija būtų sąmoninga apie savo tinklo pažeidžiamumus ir imtųsi priemonių siekiant apsaugoti savo tinklą.

1.3. Kibernetinės žvalgybos teisės ir aktų svarba

Kibernetinė žvalgyba yra esminė šiuolaikinio saugomo dalis, ji yra skirta identifikuoti ir apsaugoti asmenis ar organizacijas nuo kibernetinių grėsmių. Aiškūs reisiniai rėmai ir reguliavimas yra svarbu, kad būtų užtikrinta teisėta ir proporcinga veikla, kuri apsaugotų asmenų ir organizacijų privatumą ir užkirstų kelią piktnaudžiavimui. Teisės aktai nustato taisykles ir apibrėžia atsakomybę už neteisėta veiklą. Tai padeda kurti ir sustiprinti saugią, bei patikimą kibernetinę erdvę.

1.3.1. Kibernetinės žvalgybos teisės ir aktai

Kibernetinė žvalgybos teisės ir aktai priklausomai nuo šalies gali daug kur skirtis, nes visos šalys turi skirtingas politikas ir įstatymus reguliuojančius kibernetinės žvalgybos privatumą. Kibernetinės žvalgybos teisės aktai yra svarbūs reguliuojant ir užtikrinant tinkamą kibernetinės erdvės veiklą bei saugumą. Tai apima kelis esminius aspektus. Pirma, tai nustato griežtus reikalavimus apsaugai ir aiškiai apibrėžia kaip turi būti saugoma, renkama ar naudojama asmeninė informacija. Antra, tai teisinis pagrindas turi būti tvirtai įtvirtintas, kad kibernetinė žvalgyba būtų vykdoma tik su reikiamais teisėsaugos įgaliojimais ar leidimais. Trečia, kai kurie aktai taip pat numato priemones kibernetinių atakų prevencijai ir aptikimui, reikalaujant, kad viešosios valstybės įmonės ar organizacijos įsidiegtų tam tikras priemones pažeidimų aptikimui. Galiausiai, aktai padeda apsaugoti intelektinę nuosavybę nuo kibernetinės saugos žvalgybos, bei pavogimo. Valstybės

įstatymų ir aktų nesilaikymas gali sukelti daug blogų padarinių ir už tai gali grėsti baudžiamoji atsakomybė ar teisiniai ginčai (Raul, 2019).

1.3.2. Apibendrinimas

Kibernetinės žvalgybos teisių ir aktų svarba negali būti pervertinta. Tai padeda užtikrinti saugią kibernetinę erdvę, bei padeda apsaugoti piliečių bei organizacijų teises nuo kibernetinių nusikaltimų. Kibernetinės žvalgybos teisės ir aktai skirtingose šalyse gali labai skirtis, nes kiekviena šalis turi savo įstatymus, politikas ir teisės aktus, reguliuojančius kibernetinės žvalgybos privatumą. Nepaisant šių skirtumų, svarbu atsižvelgti į šalies teisės aktus ir tinkamai užtikrinti kibernetinės erdvės veiklą ir saugumą. Nesilaikant valstybės įstatymų ir aktų gali kilti rimtų pasekmių įskaitant baudžiamąją atsakomybę ar teisinius ginčus. Todėl svarbu, kad organizacijos ir asmenys, būtų tinkamai informuoti apie savo šalies kibernetinio saugumo teises ir aktus.

1.4. Naudotos platformos ir įrankiai

Siekiant taisyklingai susigeneruoti ir išsianalizuoti pažeidimų srautą buvo naudojamos „*GNS3*" ir „*VMware*" platformos kartu su šiais įrankiais:

1. „*Kali Linux*" – „*VMware*" sukurta virtuali mašina su įvairiais įrankiais.
2. „*Nmap*" – Tinklo skeneris.
3. „*Wireshark"* – paketų analizavimas, tinklo srauto stebėjimas.
4. „*Tshark*" – Paketų kopijavimas.
5. „*Ubuntu*" – „*VMware*" sukurta virtuali mašina naudojama tinklo skenavimui ir analizavimui.
6. „*MS Windows 10*" – „*VMware*" sukurta virtuali mašina naudojama kaip taikinys.

1.4.1. „*GNS3*" Simuliacinė aplinka skirta duomenų generavimui

Platforma „*GNS3* (angl. *Graphical Network Simulator 3*)" – atviro kodo simuliacijos įrankis. Jis leido lengvai kurti kompleksinius kompiuterinius tinklus virtualioje aplinkoje ir juos testuoti. Šis įrankis yra plačiai naudojamas tarp informatikos specialistų norint išsitestuoti naujus tinklo dizainus ar konfigūracijas, bei veikimą nenaudojant fizinės tinklo įrangos. Pagrindinės „*GNS3*" funkcijos ir savybės yra:

- Tinklo topologijos kūrimas – „*GNS3*“ vartotojams leidžia lengvai kurti, testuoti ir simuliuoti įvairias tinklo topologijas, jas testuoti naudojant įvairias virtualias mašinas, komutatorius ir maršrutizatorius, bei kitus įrenginius.
- Virtualių įrenginių palaikymas – „*GNS3*“ palaiko daug skirtingų operacinių sistemų. Leidžia įtraukti virtualius maršrutizatorius, komutatorius, bei pačias virtualias mašinas kaip „*Linux*“ ar „*MS Windows 10*“ į tinklo simuliacijas.
- Realaus laiko stebėjimas ir analizė – „*GNS3*“ leidžia vartotojams stebėti virtualių įrenginių veikimą realiu laiku, jas konfigūruoti, testuoti ir vertinti jų veikimą, bei efektyvumą.
- Išplėstinė konfigūracija bei integracija – „*GNS3*“ leidžia naudoti įvairias išplėstines funkcijas ir įrankius tokias kaip „*Wireshark*“ ar „*Docker*“ kas leidžia sukurti daug sudėtingesnes tinklo topologijas ar aplinkas.

„*GNS3*“ yra galinga ir pažangi atviro kodo platforma, kuri puikiai tinka simuliacijom, testavimui, bei mokinimuisi IT specialistams („*GNS3*“ dokumentacija).

1.4.2. „*VMware*“ virtualizacijos platforma

Platforma „*VMware*“ yra viena iš pažangiausių virtualizavimo platformų. Ji leidžia vartotojams ar organizacijoms kurti virtualias mašinas tokias kaip „*Linux*“ savo kompiuteryje ar serveryje. Ši virtualizacijos sistema suteikia didelį lankstumą ir efektyvumą organizacijoms, bei asmenims, leidžianti kurti ir valdyti virtualias mašinas. Ši platforma puikiai tinka simuliuoti kibernetinę žvalgybą saugumo tyrimams („*VMware*“ dokumentacija).

1.4.3. Naudingi įrankiai simuliacijoje

Šioje simuliacijoje buvo panaudoti toki įrankiai kaip „*Wireshark*“, „*Tshark*“, „*Nmap*“. Šie įrankiai yra tobuli norint išsitestuoti ir geriau suprasti sukurtą tinklą. Platus funkcijų pasirinkimas leidžia laisvai skenuoti norimus tinklo aspektus, gautus rezultatus išsisaugoti atskirame CSV faile ir greitai, bei efektyviai juos išsianalizuoti. Šių įrankių pagalba buvo sukurtas pažeidimų srautas. Pagrindinės šių įrankių funkcijos:

„*Nmap* (angl. *Network Mapper*)“ įrankis. Tai yra galingas atviro kodo įrankis. Kuris yra naudojamas skenuoti tinklą. Šis įrankis administratoriams ir saugumo specialistams leidžia lengvai ištirti tinklą, aptikti daug įvairios informacijos, identifikuoti įrenginius bei įvertinti tinklo saugumą. Kelios iš svarbiausių „*Nmap*“ savybių:

- Prievadų skenavimas.

- Operacinių sistemų aptikimas.
- Paslaugų identifikavimas.
- Skriptų variklis.
- Skirtingi skenavimo variantai.

„*Nmap*“ įrankis turi ir daugiau naudingų funkcijų, jas galima rasti „*Nmap*“ dokumentacijoje. Pagrindinė simuliacijai naudojama „*Nmap*“ funkcija buvo prievadų skenavimas. Buvo naudojami septyni skenavimo būdai: „TCP scan“, „UDP scan“, „SYN scan“, „FIN scan“, „ACK scan“, „NULL scan“ ir „XMAS scan“. Šių skenavimo būdų savybes buvo aprašytos analizės skyriuje „1.2.3. Prievadų skenavimas“ („*Nmap*“ dokumentacija).

„*Wireshark*“ įrankis. Tai yra labai populiarus atviro kodo tinklo analizės įrankis, kuris leidžia stebėti ir analizuoti tinklo srautą realiu laiku. Šio įrankio pagalba galima detaliai tirti visus praeinančius paketus, iš jų išsirinkti daug svarbios informacijos, tokios kaip paketo šaltinį, paskirties vietą, protokolą ir t.t. „*Wireshark*“ leidžia gautus rezultatus filtruoti taip, kad būtų atvaizduojami tik aktuali informacija – tai leidžia efektyviai dirbti. Šis įrankis turi grafinę vartotojo sąsają taip palengvindamas darbą kiekvienam savo vartotojui („*Wireshark*“ dokumentacija).

„*Tshark*“ įrankis. Šis įrankis yra skirtas analizuoti ir kompiuterinius tinklus. Šis įrankis yra „*Wireshark*“ komandinės eilutės versija, tai leidžia naudoti tas pačias funkcijas kaip ir „*Wireshark*“ tačiau be grafinės sąsajos. Tai yra labai naudinga naudojantis Linux sistemomis.

„*Kali Linux*“ operacinė sistema tai yra atviro kodo „*Linux*“ distribucija, kurioje buvo sudiegta daug įvairių įrankių skirtų kibernetinio saugumo specialistams, bei tinklo testuotojams. Vienas iš pagrindinių „*Kali Linux*“ tikslų yra suteikti vartotojams platų pasirinkimą tinklo testavimui, saugumui ir atkūrimui. Ši operacinė sistema yra paremta „*Debian Linux*“ sistema, todėl paveldi labai daug jo funkcijų (Tigner, Wimmer ir Rebman, 2021).

„*Ubuntu*“ yra dar viena atviro kodo operacinė sistema. Ši operacinė sistema pasižymi paprastu naudojimu ir dažnai yra renkamasi pradedančiųjų naudotojų. „*Ubuntu*“ naudoja savo tvarkyklę, vadinamą „apt“, tai leidžia lengvai valdyti programas, jas atnaujinti, instaliuoti. Ši operacinė distribucija yra naudojama tiek komerciniam naudojimui, tiek asmeniniam („*Ubuntu*“ dokumentacija).

1.4.4. Apibendrinimas

Šiame skyriuje buvo aptarta naudotų platformų ir įrankių įvairovė skirta kibernetinės žvalgybos simuliacijoms ir saugumo tyrimams atlikti. Pagrindinės naudotos platformos buvo „*GNS3*“ ir „*VMware*“, kurios suteikė galimybė kontroliuoti ir valdyti virtualias tinklo aplinkas, bei įrenginius. Įvairių įrankių, tokių kaip „*Kali Linux*“, „*Ubuntu*“, „*Nmap*“, „*Wireshark*“, „*Tshark*“ ir „*MS Windows*“ naudojamas leido efektyviai atlikti tinklo skenavimus, paketų analizė ir įvairius saugumo testus.

„*GNS3*“ leidžia kurti sudėtingas tinklo topologijas ir jas stebėti realiu laiku virtualioje aplinkoje, taip pat topologijoje esančius įrenginius konfigūruoti, testuoti ir stebėti jų veikimą. „*VMware*“ platforma suteikia lankstumą ir efektyvumą kuriant ir valdant virtualias mašinas, kurių pagalba buvo galima naudoti įvairius įrankius, kurių pagrindinė paskirtis buvo tinklo srauto analizė. Šie įrankiai ir platformos buvo esminiai norint simuliuoti kibernetinės žvalgybos veiklą.

2. PROJEKTAVIMAS

Šioje dalyje aprašoma sistemos architektūra, aprašomas simuliuojamo tinklo topologijos sudarymas, reikalavimai aparatūrai, realizacijai. Buvo aprašytos objekto funkcijos, jis apibūdinamas, jo paskirtis, bei reikalavimai saugumui.

2.1. Projektuojamo objekto apibūdinimas

Šiame darbe buvo projektuojamas tinklas, kurio pagalba buvo galima lengvai simuliuoti tinklo skenavimus, atlikti bandymus, testavimus ir taip sugeneruoti pažeidimų srautą. Gautus rezultatus išanalizuoti, taip surasti tinklo pažeidžiamumus ir šia gautą informaciją panaudoti ateityje, užkertant kelią kibernetiniams nusikaltėliams.

2.2. Projektuojamo objekto paskirtis

Šio projekto paskirtis – užkirsti kelią kibernetiniams nusikaltimams analizuojant sugeneruotą pažeidimų srautą. Šio projekto pagalba buvo galima atlikti įvairius testavimus, išbandyti skirtingus įsilaužimo metodus ir iš jų gauti rezultatus, kuriuos vėliau bus galima panaudoti mašininiam mokymui, ar simuliaciją naudoti kaip virtualią laboratoriją įvairiems bandymams.

2.3. Projektuojamo objekto funkcijos

Projektuojamo objekto funkcijos:

1. Tinklo skenavimas įvairiais metodais. Naudojama „*Kali Linux*“ operacinė sistema, kurioje yra daug įvairių tinklo skenavimo įrankių, jų pagalba buvo nuskaitomi prievadai, aptinkamos pažeidimų vietos, spragos.
2. Tinklo duomenų rinkimas. Naudojantis įrankiais „*Wireshark*“ ir „*Tshark*“ buvo gauta daug informacijos apie tinklo srautą, siunčiamus ir gaunamus paketus.
3. Duomenų analizė ir virtualizacija siekiant aptikti neįprastus ar potencialius pavojingus srautus ar srauto anomalijas.
4. Sukurtos tinklo topologijos simuliavimas.

2.4. Reikalavimai projektuojamo objekto posistemei

Atsižvelgiant į tai, kad viskas buvo vykdoma virtualioje aplinkoje, svarbu užtikrinti, kad informacinės posistemės buvo suderintos su šia aplinka. Būtina užtikrinti, kad visi duomenys būtų apsaugoti nuo netikėtų duomenų praradimų, todėl buvo svarbu kurti atsargines kopijas, kad praradus duomenis, juos būtų galima atkurti. Taip pat svarbu, kad duomenys būtų greitai ir efektyviai analizuojami ir gaunami, kad būtų galima greitai ir efektyviai reaguoti į kibernetinius srautus.

2.5. Reikalavimai saugumui

Generuojant ir analizuojant pažeidimų srautus, svarbu atsižvelgti į saugumą. Tai apima duomenų saugumą, sistemos atkūrimo galimybes, apsaugą nuo kenkėjiško kodo.

Buvo užtikrintas duomenų konfidencialumas, vientisumas ir prieinamumas. Būtina užtikrinti, kad visi surinkti duomenys buvo saugomi ir nepasiekiami neautorizuotiems asmenims, tai apima duomenų šifravimą ir galimybę kontroliuoti prieigą prie jų.

Svarbu turėti sistemos atkūrimo galimybes, kad būtų galima atkurti sistemą po įvairių bandymų, bei testavimo, kad prieš tai atlikti bandymai neįtakotų naujo bandymo rezultatų. Taip pat svarbu turėti duomenų atkūrimo būdus, kad nebūtų netyčia prarasti visi duomenys.

Svarbu apsaugoti ir savo kompiuterį nuo kenkėjiškų programų, kodo, kad atliekami bandymai nebūtų įtakojami ar duomenys pasisavinami pašalinių.

Taigi, viską apibendrindamas galiu drąsiai teigti, kad saugumo užtikrinimui būtina užtikrinti generuojamų ir analizuojamų duomenų konfidencialumą, vientisumą ir prieinamumą.

2.6. Reikalavimai realizacijai

Realizuojant šį projektą, svarbu užtikrinti, kad kompiuteris, kuriame kuriamas tinklas testavimams atlikti atitiktų aparatūrinės posistemės reikalavimus. Tinklas buvo sukurtas taip, kad būtų galima lengvai konfigūruoti virtualią aplinką ir tinklo skenavimus, tai gali apimti virtualių mašinų valdymą, bei konfigūravimą. Svarbu, kad kompiuteris, kuriame atliekamos simuliacijos tūrėtų pakankamai fizinių resursų, kad viskas vyktų sklandžiai ir efektyviai. Reikia užtikrinti, kad analizuojami duomenys buvo efektyviai surenkami duomenų analizei ir tinkamai saugomi. Taip pat tinklas buvo sukurtas taip, kad būtų užtikrintas jo pasiekiamumas ir patikimumas. Naudinga jei tinklą galima stebėti ir konfigūruoti realiu laiku.

3. PROJEKTINĖ DALIS

Šioje dalyje pateikiama sistema, kokias platformas ji naudoja, jų integracija. buvo sukurta tinklo topologija, jos paruošimas, bei programų sudiegimas. Aprašoma kaip buvo paruoštos virtualios mašinos simuliacijos naudojimui, įrankių įdiegimas ir jų konfigūracija, kad jos galėtų laisvai komunikuoti tarpusavyje.

3.1. Sistemos architektūra

Ši architektūra rėmėsi dviem pagrindiniais elementais – „*GNS3*“ ir „*VMware*“ platformomis. „*GNS3*“ aplinka buvo naudojama kurti ir moduliuoti tinklus. Ji suteikė galimybę konfigūruoti tinklo įrenginius. „*VMware*“ – virtualizuoti serverius ir kompiuterius, šios virtualios mašinos buvo naudojamos kaip sistemos dalys, kuriose buvo vykdomos kibernetinės žvalgybos operacijos. Šiame projekte buvo svarbus integracijos taškas, kuris leido „*GNS3*“ aplinkai lengvai bendrauti su „*VMware*“ aplinkos virtualiomis mašinomis. Ši architektūra buvo grindžiama esamų platformų funkcionalumu, siekiant užtikrinti efektyvią kibernetinės žvalgybos simuliaciją. Sistemos architektūros funkcijos:

1. „*GNS3*“ ir „*VMware*“ integracija – „*GNS3*“ buvo naudojamas simuliuoti tinklo įrenginius ir jų sąveiką, o „*VMware*“ buvo skirta virtualizuoti serverius ir jų kompiuterius.
2. Tinklo modeliavimo aplinka – „*GNS3*“ suteikė aplinką, kurią buvo galima lengvai kurti ir konfigūruoti, keičiant topologijos išdėstymą, bei keičiant įrenginių nustatymus ir kt. Svarbu paminėti, kad ši aplinka leido testuoti ir simuliuoti kibernetinę žvalgybą skirtinguose tinklo elementuose.
3. Virtualios mašinos „*VMware*“ aplinkoje – „*VMware*“ platforma leido lengvai kurti ir valdyti virtualias mašinas, susidiegti norimas operacines sistemas, kurios atitiko poreikius. Šios virtualios mašinos buvo naudojamos kaip pagrindinės sistemos dalys, kurių pagalba buvo atliekama kibernetinę žvalgyba ir testuojamas tinklas.
4. Integracijos taškai – šie taškai buvo reikalingi tam, kad „*GNS3*“ sukurti tinklo modeliai lengvai bendrautų su „*VMware*“ virtualiomis mašinomis.

3.2. Simuliuojamos tinklo topologijos sukūrimas

Buvo sukurta simuliuojama ir konfigūruojama tinklo topologija, kuri buvo naudojama kibernetinės žvalgybos tinklų simuliavimui. Tai svarbus žingsnis. Leidžiantis atlikti išsamesnę kibernetinės žvalgybos analizė ir testavimą.

3.2.1. Tinklo modelių kūrimas „*GNS3*“ aplinkoje

Pasinaudojus „*GNS3*“ platforma, buvo sukurta virtuali tinklo topologija, kuri atitiko realaus pasaulio infrastruktūros elementus. Pati topologija buvo kuriama ganėtinai paprasta, buvo naudojamas vienas maršrutizatorius, keli serveriai ir kelios darbo vietos, šių objektų pagalba buvo galima atlikti tinklo skenavimus ir bandymus. Kiekvienas įrenginys buvo konfigūruojamas atitinkamai, kad galėtų komunikuoti tarpusavyje.

3.2.2. Virtualių mašinų kūrimas „*VMware*“ aplinkoje

„*VMware*“ platformoje buvo sukuriamos virtualios mašinos su „*Kali Linux*“ ir „*Ubuntu*“ operacinėmis sistemomis. Šios operacinės sistemos svarbios, nes jos yra pritaikytos tinklo skenavimui ir stebėjimui. Jas buvo lengva valdyti. Vos tik įsidiegus virtualias mašinas, jos iškart turėjo daug naudingų tinklo skenavimo įrankių, tokių kaip „*Nmap*“, „*Wireshark*“ ir t.t. Svarbu paminėti, kad trūkstamus įrankius į šias operacines sistemas buvo lengva susidiegti ir pradėti naudoti. Be šių operacinių sistemų, bei įrankių skenuoti tinklą būtų sudėtinga ir būtų buvę sunkiau sugeneruoti ir išanalizuoti pažeidimų srautą.

3.2.3. Integracija su „*GNS3*“ ir „*VMware*“

Buvo svarbu užtikrinti, kad simuliacijoje sukurtos virtualios mašinos su „*VMware*“ ir tinklo įrenginiai sukurti su „*GNS3*“ gali tarpusavyje lengvai sąveikauti ir komunikuoti, leisdami vykdyti kibernetinę žvalgybą, jos operacijas, bei testavimus tinklo topologijoje. Buvo atliekamas paprastas testavimas, siekiant įsitikinti, kad įrenginiai esantys tinklo topologijoje gali susisiekti. Buvo konfigūruojami tinklo adapteriai, kad viskas sklandžiai susijungtų į tinklą ir veiktų tarpusavyje.

3.2.4. Apibendrinimas

Viską apibendrinant, buvo sukurta tinklo topologija pasinaudojant „*VMware*“ ir „*GNS3*“ platformomis. Buvo užtikrinta šių platformų integracija, tai yra, kad platformos galėtų lengvai sąveikauti tarpusavyje. Su „*VMware*“ pagalba buvo sukurtos virtualios mašinos su „*Kali Linux*“ ir „*Ubuntu*“ virtualiomis mašinomis, kad būtų galima lengviau atlikti kibernetinę žvalgyba ir generuoti, bei analizuoti pažeidimų srautą sukurtoje topologijoje.

3.3. Reikalavimai aparatūrai

Šiam projekto vykdymui buvo reikalingas ganėtinai galingas kompiuteris, buvo svarbu, kad kompiuteris atitiktų šiuos reikalavimus:

1. Kompiuterio procesorius – turėjo būti pajėgus vykdyti virtualizacijos operacijas efektyviai, kad būtų užtikrintas efektyvus tinklo skenavimas ir duomenų analizavimo veiksmingumas.
2. Didelis atminties kiekis – buvo naudingas didesnis kiekis RAM atminties, kad viskas veiktų sklandžiai ir būtų galima vienu metu įsijungti kelias virtualias mašinas.
3. Tinkama saugykla – buvo naudinga turėti daugiau vietos duomenims saugoti, taip pat užtikrinti aukštą įrašymo greitį, kad duomenų operacijos būtų efektyvios.
4. Geras tinklo ryšys – kompiuteris turėjo užtikrinti greitą ir stabilų tinklo ryšį, kad būtų galima nenutrūkstamai atlikti bandymus, bei testavimus.

Svarbu buvo įsitikinti, kad virtualizuota aplinka buvo tinkamai sukonfigūruota ir optimaliai išnaudojo kompiuterio resursus. Patogu buvo jei vienu metu buvo galima įsijungti kelias virtualias mašinas, kad būtų galima stebėti jų tarpusavio veikimą. Šiuo atveju vienu metu buvo įjungtos „*Kali Linux“,* „*MS windows 10*“ ir „*Ubuntu*“ virtualios mašinos.

3.4. „*GNS3*“ ir „*VMware*“ virtualios platformos įdiegimas ir parengimas

Prieš pradedant simuliacijas, buvo svarbu įsitikinti, kad „*GNS3*“ ir „*VMware*“ platformos buvo tinkamai įdiegtos ir parengtos naudojimui. „*GNS3*“ ir „*VMware*“ platformos buvo įdiegtos ir paruoštos asmeniniame kompiuteryje, kuris buvo naudojamas kibernetinės žvalgybos simuliavimui ir tinklo kūrimui. Tai užtikrinau įvykdydamas toliau pateiktus žingsnius.

3.4.1. „*VMware*“ įdiegimas

„*VMware*“ platforma buvo parsisiųsta iš oficialios svetainės. Svetainėje buvo pasirinkta „*VMware*“ versija pagal operacinę sistemą ir kompiuterio pajėgumą, buvo atsisiųsta tvarkyklė ir atidaryta.

„*VMware*“ tvarkyklės langas (3.1 pav.).

3.1 pav. „*VMware*“ tvarkyklė

Buvo praeita visi jos žingsniai, tai yra, buvo pasirinkta reikalingi įrašymo nustatymai, atsižvelgiant į asmeninio kompiuterio pajėgumą, operacinę sistemą ir kitus svarbius aspektus. Tada buvo palaukiama kol „*VMware*“ platforma įsirašė. Tada buvo galima pradėti naudoti „*VMware*“ platformą.

3.4.2. GNS3 įdiegimas

„*GNS3*“ platforma buvo parsisiųsta iš oficialios svetainės. Svetainėje buvo pasirinkta „*GNS3*“ versija pagal operacinę sistemą ir kompiuterio pajėgumą, buvo atsisiųsta tvarkyklė ir atidaryta.

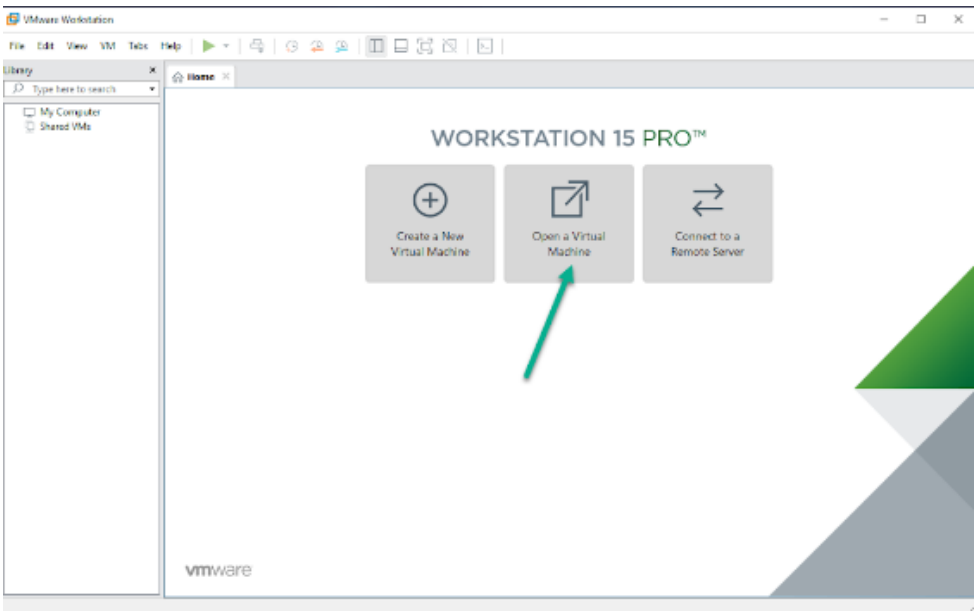
„*GNS3*“ tvarkyklės langas (3.1 pav.).

3.2 pav. „*GNS3*“ tvarkyklė

Buvo praeita visi jos žingsniai, tai yra, buvo pasirinkta reikalingi įrašymo nustatymai, atsižvelgiant į asmeninio kompiuterio pajėgumą, operacinę sistemą ir kitus svarbius aspektus. Tada buvo palaukiama kol „*GNS3*“ platforma įsirašė. Tada buvo galima pradėti naudoti „*GNS3*“ platformą.

3.4.3. „*GNS3 VM*“ Įdiegimas

Įsirašius „*GNS3*” ir „*VMware*“ platformas svarbu įsirašyti ir „*GNS3 VM*“, kad „*GNS3*“ platforma palaikytų „*VMware*“ platformą ir galėtų tarpusavyje sąveikauti be problemų. Norėdami tai padaryti, pradžioje parsisiunčiame „*GNS3 VM*“ failą iš oficialios „*GNS3*“ svetainės. Šis failas buvo parsisiųstas „*.rar*“ tipo archyvas, kurį išsiarchyvuojame. Tai atlikę atsidarome „*VMware*“ programą ir pasirenkame „*Open a Virtual Machine*“ kaip parodyta 3.3 paveiksle, kur nurodomas atitinkamas mygtukas.



3.3 pav. „*VMware*“ programos langas virtualiai mašinai pridėti

Paspaudus „*Open a Virtual Machine*“ pasirenkame direktoriją, kurioje buvome išsiarchyvavę „ar*GNS3 VM*“ failą ir pasirenkame jį. Tada spaudžiame „*Import*“ ir palaukiame, kol „*VMware*“ įkels „*GNS3 VM*“. „*GNS3 VM*“ integravimas atliktas, galima pradėti naudotis „*GNS3*“ ir „*VMware*“ platformomis.

3.4.4. Apibendrinimas

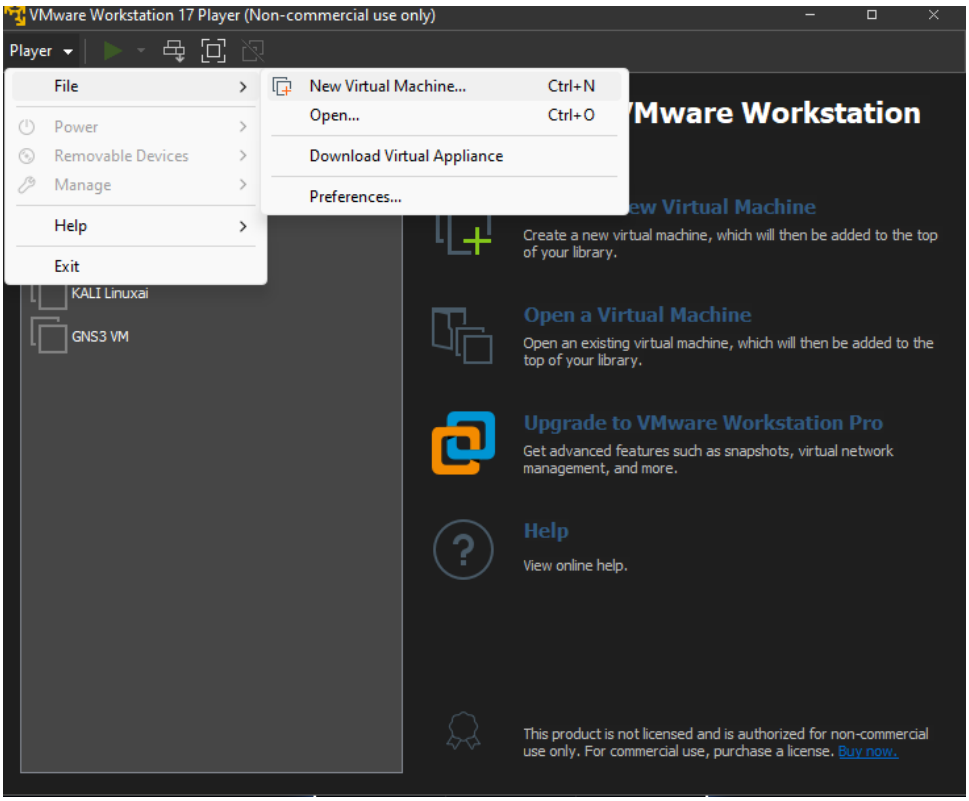
Į asmeninį kompiuterį buvo įdiegta „*GNS3*“ ir „*VMware*“ platformos, kuriose buvo galima atlikt įvairius tinklo testavimus ir bandymus. Į „*GNS3*“ pridėjau „*GNS3 VM*“, kad „*GNS3* “ platforma palaikytu „*VMware*“ virtualias mašinas.

3.5. Serverių įdiegimas

Šiam projektui buvo įdiegti trys virtualūs serveriai „*VMware*“ platformoje. Šie serveriai naudojo skirtingas operacines sistemas – „*Kali Linux*“, „*Ubuntu*“, „*MS Windows*“.

3.5.1. Naujos virtualios mašinos pridėjimas prie „*VMware*“

Prieš įdiegiant virtualią mašiną, „*VMware*“ platformoje turime ją pridėti. Tai buvo padaryta atsidarius „*VMware*“ programą ir paspaudus „*New Virtual Machine...*“ žiūrėti paveikslą (3.4).



3.4 pav. „*VMware*“ naujos virtualios mašinos pridėjimas

Tada atsidaro naujas langas, kuriame buvo pasirinkta operacinės sistemos ISO atvaizdas, toliau buvo pasirinkta operacinė sistema, jei ji nebuvo automatiškai aptikta. Tada buvo sukurta būsimos virtualios mašinos pavadinimą, pasirinkta virtualios mašinos vietą diske, kur ji bus įrašoma ir spaudžiame „*Next*“, tada buvo pasirinkta kietojo disko talpą ir būdas, kaip jis bus saugojamas, vėl spaudžiame „*Next*“ ir galiausiai peržiūrėję virtualios mašinos parametrus spaudžiame „*Finish*“. Tada reikia palaukti, kol virtuali mašina bus integruota į „*VMware*“.

3.5.2. „*Kali Linux*“ įdiegimas

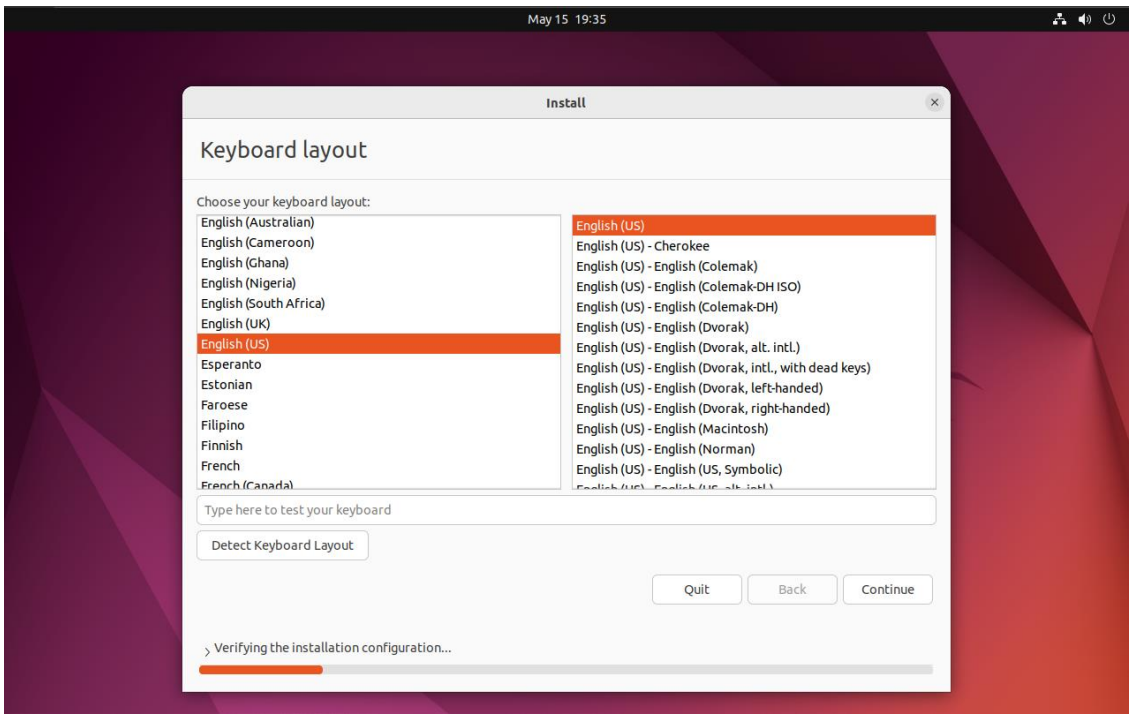
Pirma buvo įdiegta „*Kali Linux*“ operacinė sistema „*VMware*“ platformoje. Pradžiai buvo parsisiųstas „*Kali Linux*“ ISO failas iš oficialios svetainės. Svetainėje buvo pasirinkta „*Installer Images*“ tada buvo pasirinkta reikiama versija ir ji parsisiųsta. Toliau buvo pridėta virtuali mašina prie „*VMware*“ – žiūrėti skyrių „3.5.1 Naujos virtualios mašinos pridėjimas prie „*VMware*““. Tai padarius, buvo įsijungta pridėta virtualią mašiną. Atsidariusiame „*BIOS mode*“ buvo pasirinkta „*Install*“ pasirinkimas, žiūrėti 3.6 paveiksle.

3.5 pav. „*Kali Linux*“ BIOS mode.

Šis pasirinkimas įrašo „*Kali Linux*“ be grafinės sąsajos. Buvo praeiti visi „*Kali Linux*“ įrašymo procesai ir buvo įrašoma „*Kali Linux*“ operacinė sistema. Tai padarius operacinė sistema persikrovė ir priklausomai nuo pasirinkimų įrašymo procesuose įsijungė „*Kali Linux*“ darbalaukis. „*Kali Linux*“ operacinė sistema buvo paruošta naudojimui.

3.5.3. „*Ubuntu*“ įdiegimas

Antra buvo įdiegta „*Ubuntu*“ operacinė sistema „*VMware*“ platformoje. Pradžiai buvo parsisiųstas „*Ubuntu*“ ISO failas iš oficialios svetainės. Svetainėje buvo pasirinkta reikiamą versiją ir parsisiųsta. Tada buvo pridedama virtuali mašina prie „*VMware*“ – žiūrėti skyrių „3.5.1 Naujos virtualios mašinos pridėjimas prie „*VMware*““. Tai padarius, buvo įsijungta pridėta virtuali mašina. Atsidariusiame „BIOS mode“ buvo pasirinkta „*Graphical Install*“ pasirinkimas, žiūrėti 3.7 paveiksle.

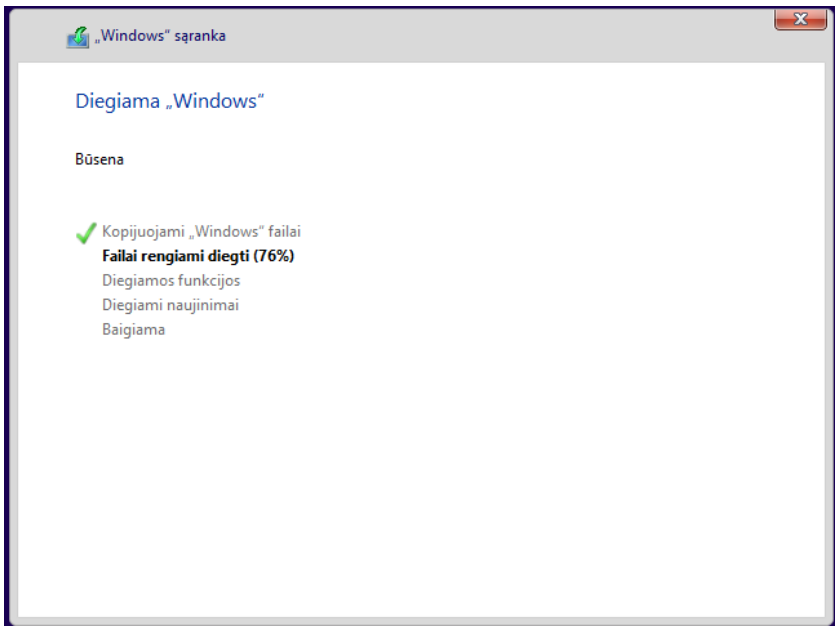


3.6 pav. „*Ubuntu*“ operacinės sistemos diegimas

Šis pasirinkimas įrašo „*Ubuntu*“ su grafine sąsaja. Buvo Praeiti visi „*Ubuntu*“ įrašymo procesai ir įrašomas „*Ubuntu*“. Tai padarius operacinė sistema persikrovė ir priklausomai nuo pasirinkimų įrašymo procesuose įsijungė „*Ubuntu*“ darbalaukis. „*Ubuntu*“ operacinė sistema buvo paruošta naudojimui.

3.5.4. „*MS Windows 10*“ įdiegimas

Galiausiai buvo įdiegta „*MS windows 10*“ operacinė sistema „*VMware*“ platformoje. Pradžiai buvo parsisiųstas „*MS Windows 10*“ ISO failas iš oficialios svetainės. Svetainėje buvo pasirinkta reikiama versiją ir parsisiųsta. Tada buvo pridėta virtuali mašina prie „*VMware*“ – žiūrėti skyrių „3.5.1 Naujos virtualios mašinos pridėjimas prie „*VMware*““. Tai padarius buvo įsijungta pridėta virtuali mašina. Įsijungus virtualią mašiną automatiškai pradėjo diegtis „*MS Windows 10”*. Įdiegimo sąranką galime matyti 3.8 paveiksle. Po greito operacinės sistemos susidiegimo įsijungė sąranka, kurioje reikėjo pasirinkti įvairius windows nustatymus, tokius kaip kalbą, regioną ir t.t.



3.7 pav. „*MS Windows* 10“ sąranka

Praėjus visus „*MS Windows 10*“ sąrankos procesus buvo baigta įrašinėti „*MS Windows 10*“. Tai padarius operacinė sistema persikrovė ir įsijungė Windows darbalaukis. „*MS Windows 10*“ operacinė sistema buvo paruošta naudojimui.

3.5.5. Apibendrinimas

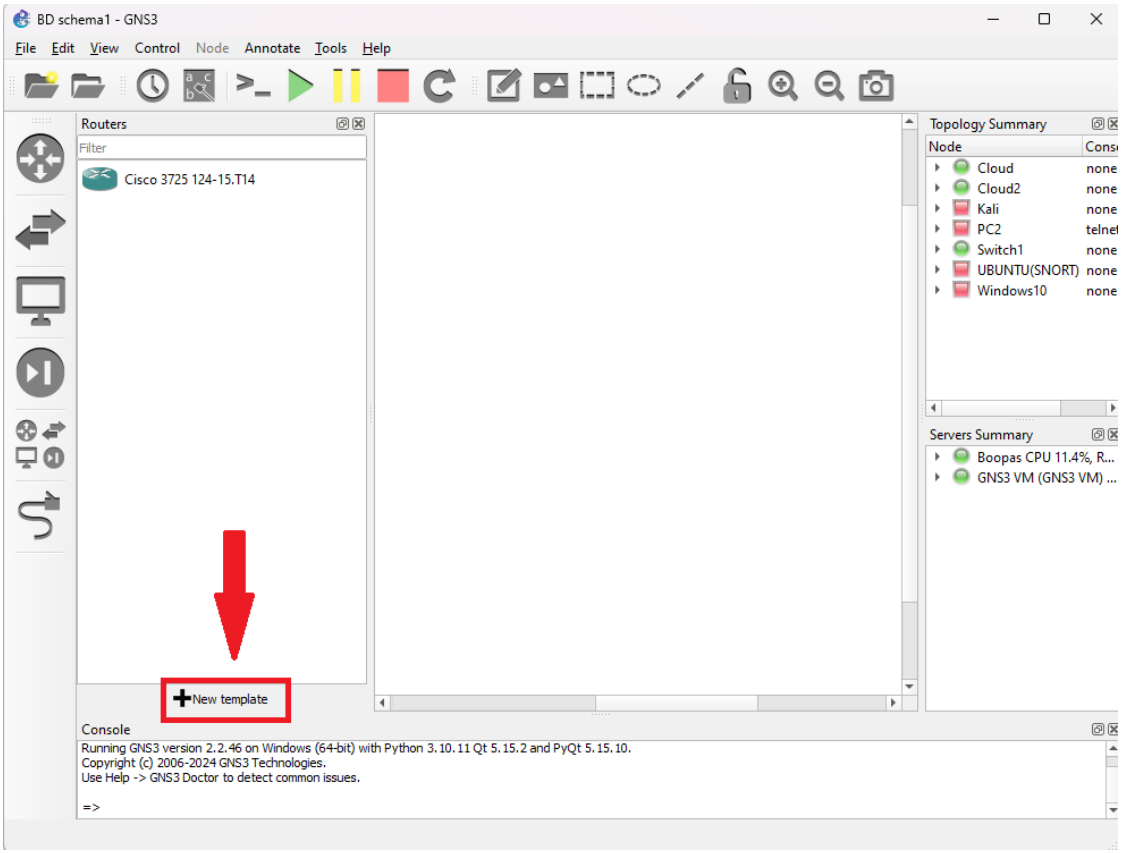
Taigi, buvo sudiegti trys serveriai su skirtingomis operacinėmis sistemomis – „*MS windows 10*”, „*Ubuntu*“, „*Kali Linux*“. Visi serveriai buvo sudiegti ir virtualizuojami „*VMware*“ platformos pagalba. Visos operacinės sistemos buvo sudiegtos su grafine sąsaja, kad būtų paprasčiau naudotis šiomis virtualiomis sistemomis.

3.6. Topologijos sukūrimas „*GNS3*“ aplinkoje

Prieš pradedant kurti topologiją, pradžiai buvo svarbu integruoti sudiegtas virtualias mašinas į „*GNS3*“. Tai padaryti buvo nesunku, kadangi „*GNS3*“ palaiko „*VMware*“, užtenka praeiti kelis žingsnius. Taip pat svarbu susipažinti su „*GNS3*“ aplinka. Tai padarius pradėjau kurti topologiją.

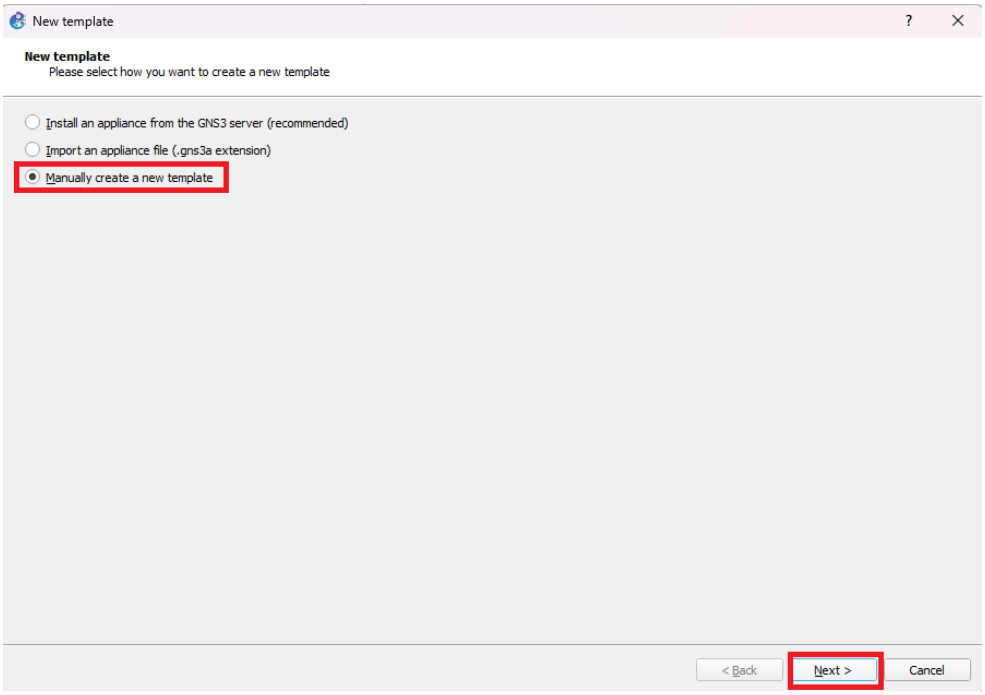
3.6.1. VMware virtualių mašinų pridėjimas į „*GNS3*“ platformą

Pradžiai buvo įsijungiama „*GNS3*“ programą, atidaromas bet kuris skirtukas dešinėje ir apačioje spaudžiame „*New template*“ (3.9 paveikslas).



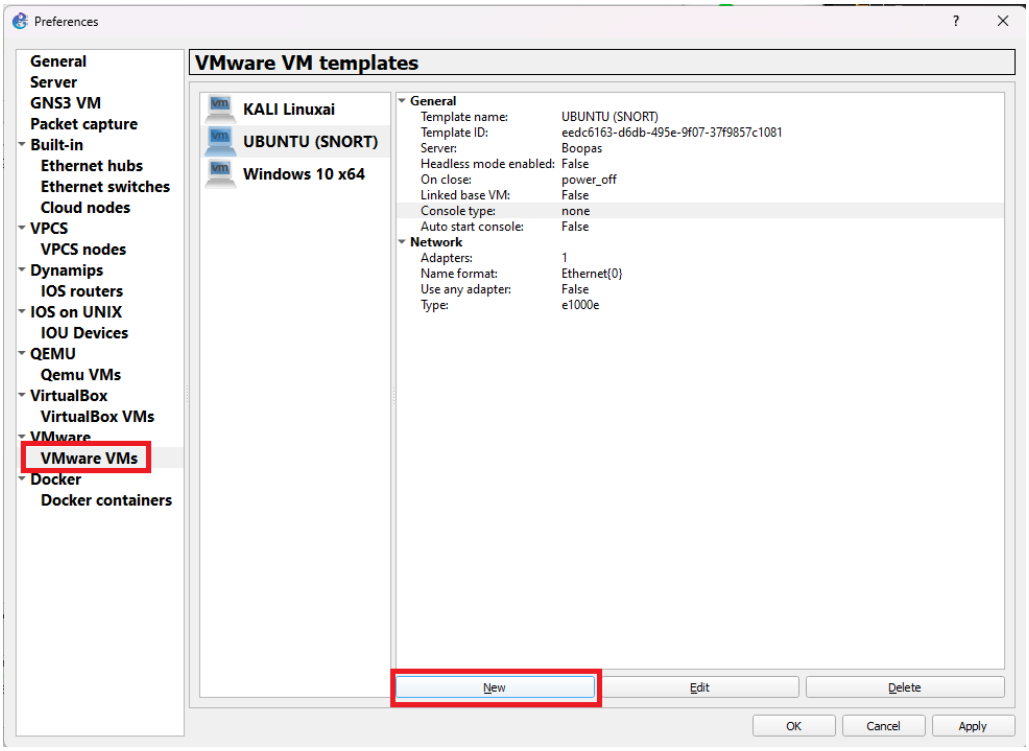
3.8 pav. „*GNS3*“ „*New template*“ mygtukas

Tada atsidariusiame „*New template*“ lange buvo uždedamos varnelės ties „*Manually create a new template*“ ir spaudžiame „*Next*“ žiūrėti 3.10 pav.



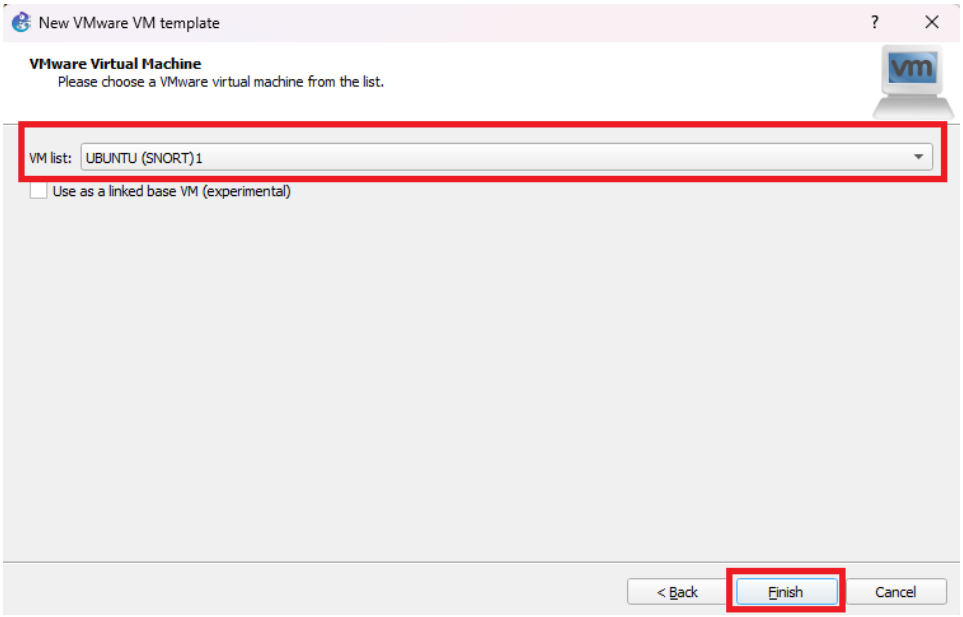
3.9 pav. „New Template“ langas

Toliau Atsidarė naujas „*Preferences*“ langas, jame buvo pasirinkta „*VMware VMs*“ skirtukas ir apačioje spaudžiama „*New*“ žiūrėti 3.11 pav.



3.10 pav. „*GNS3*“ „*Preferences*“ langas

Tada Atsidaro naujas „*New VMware VM template*“ langas, buvo uždedamos varnelės ant „*Run this VMware VM on my local computer*“ ir vėl spaudžiama „*Next*“. Sekančiame lange buvo pasirinkta norima virtuali mašina, kurią buvo pridėta į „*GNS3*“ platformą. Pasirinkę norimą virtualią mašiną, buvo spaudžiama „*Finish*“ žiūrėti 3.12 paveiksle.



3.11 pav. „*New VMware VM Template*“ langas

Tada „*VMware*“ virtuali mašina buvo integruota „*GNS3*“ platformoje, ją buvo galima pradėti naudoti.

3.6.2. Susipažinimas su „*GNS3*“ grafine sąsają

Prieš pradėdamas naudotis „*GNS3*“ platforma, buvo svarbu suprasti jos grafinę sąsają. Šis skyrius trumpai paaiškins svarbiausias grafinės sąsajos funkcijas. Kad būtų galima lengviau suprasti paveikslėli „pav. 3.13 „*GNS3*“ grafinė sąsają“ buvo sužymėtas skirtingomis spalvomis (Juoda, geltona, raudona, mėlyna, pilka).

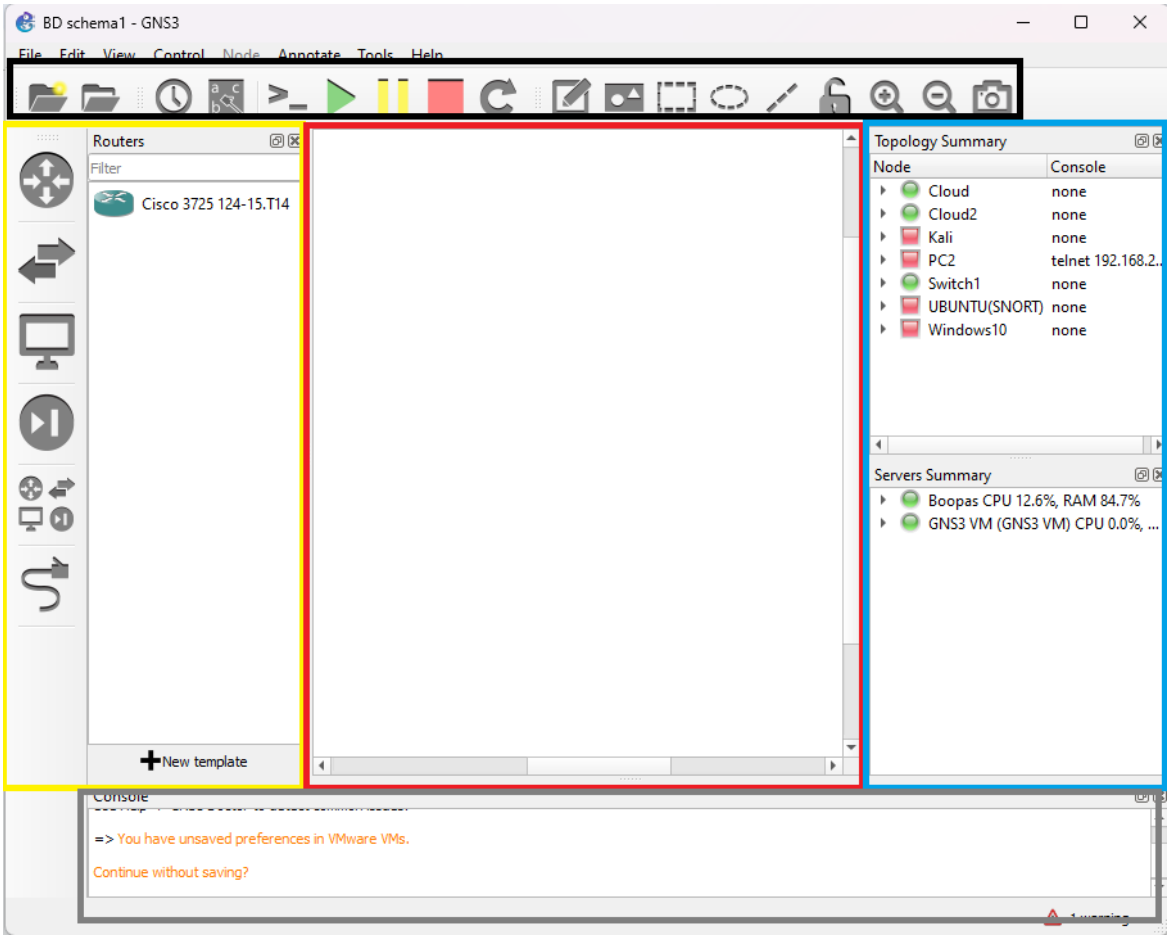
Juoda spalva – pažymėtas įrankių meniu, kuriuo galima lengvai ir patogiai valdyti virtualias mašinas. Jas įjungti, išjungti, perkrauti ir t.t.

Geltona spalva – buvo pažymėtas virtualių įrenginių meniu. Šio meniu pagalba galima susirasti reikiamus virtualius renginius ir juos pridėti į topologiją.

Raudona spalva – buvo pažymėtas laukas, kuriame buvo kuriama topologija. Šioje vietoje galima pridėti, ištrinti ar redaguoti įrenginius, siekiant sukurti norimas topologijas.

Mėlyna spalva – buvo pažymėta vieta, kur rodomi prie topologijos pridėti virtualūs įrenginiai, taip pat rodoma jų būsena, kiek kompiuterio resursų jie naudoja.

Pilka spalva – buvo pažymėta „*GNS3*“ konsolė. Šioje dalyje pridedant, startuojant ar atliekant kitus veiksmus, buvo rašomi įvairūs pranešimai, toki kaip klaidos, informacija ar kt.

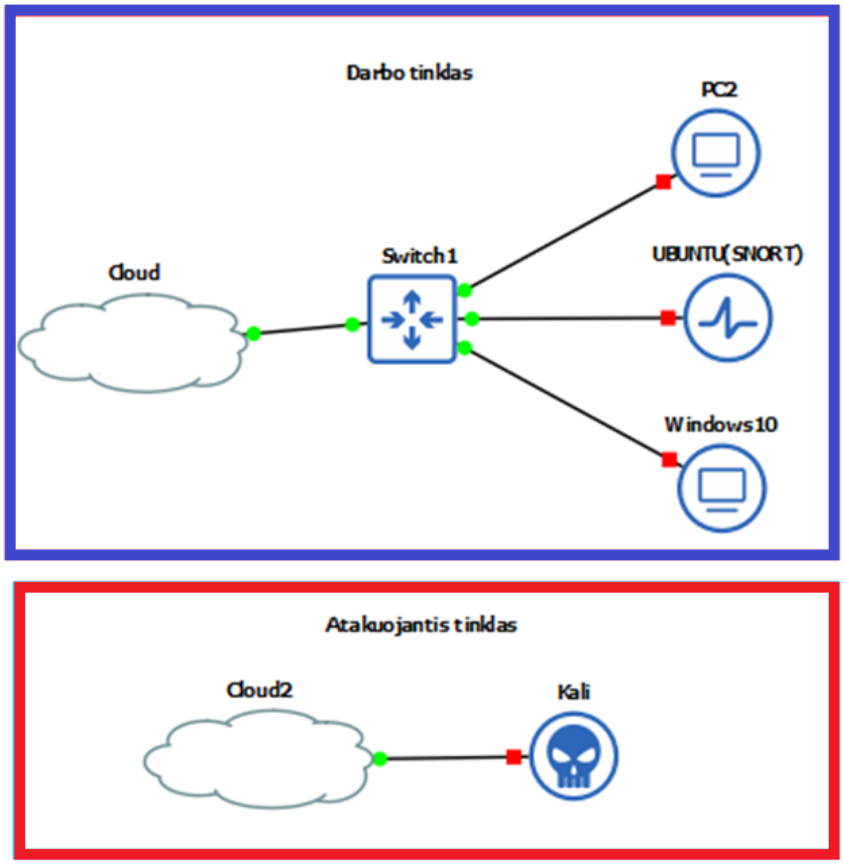


3.12 pav. „*GNS3*“ grafinė sąsaja

Susipažinęs su „*GNS3*“ grafine sąsaja, buvo galima pradėti kurti topologiją ir atlikti kibernetinės žvalgybos simuliaciją siekiant sugeneruoti pažeidimų srautą ir jį analizuoti.

3.6.3. Topologijos sukūrimas

Toliau buvo sukurta topologija naudojantis „*GNS3*” grafine sąsaja. Prieš kurdamas topologiją buvo atsižvelgta į jos projektavimą, kam ji buvo naudojama, kiek kokių virtualių įrenginių jai reikės. Į topologijos lauką buvo įdėtos trys virtualios mašinos („*Ubuntu*“, „*Kali Linux*“ ir „*MS Windows 10*“), Taip pat vienas GNS3 terminalą („*PC2“*). Buvo pridėtas vienas maršrutizatorius ir du debesys („*Could* ir „*Cloud2*“). Visi šie elementai buvo sujungti į du tinklus: vieną „Darbo tinklą“ jis buvo pažymėtas Mėlyna spalva, ir antrą „Atakuojantis tinklas“ jis buvo pažymėtas raudona spalva – žiūrėti paveikslėli „3.14 pav. Kuriamo tinklo topologija“.



3.13 pav. Kuriamo tinklo topologija

Topologija buvo sukurta. Tačiau prieš pradedant dirbti su ja, reikėjo dar sukonfigūruoti interneto adapterius ir įrašyti į virtualias mašinas trūkstamą programinę įrangą.

3.6.4. Apibendrinimas

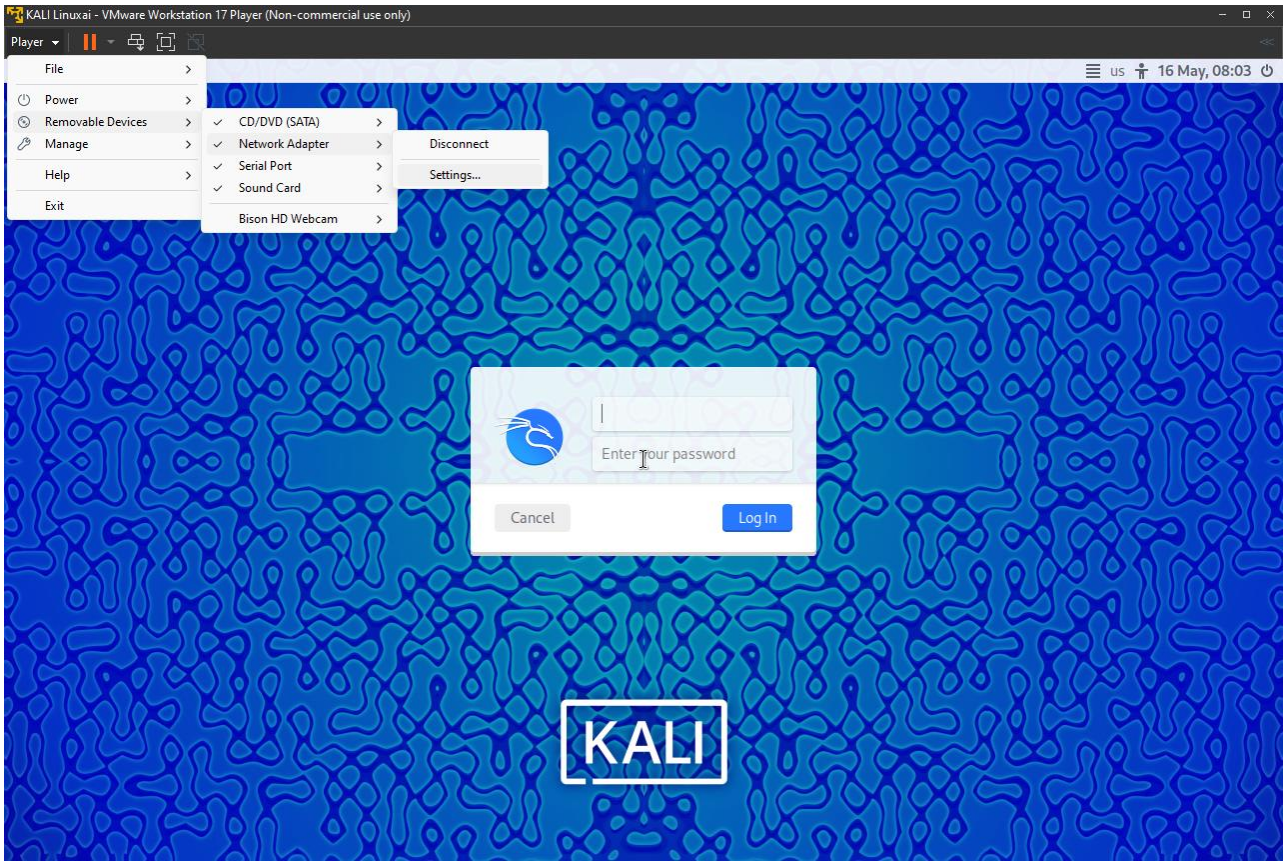
Į „*GNS3*“ virtualią aplinka buvo integruota „*VMware*“ ir sukurtos virtualios mašinos, kad galėčiau jas laisvai ir patogiai naudoti tinklo topologijoje. Buvo Susipažinta su „*GNS3*“ grafinės sąsajos svarbiausiomis funkcijomis, kad galėčiau efektyviau atlikti kibernetinės žvalgybos simuliacijas ir bandymus. Buvo sukurta tinklo topologija, ją sudarė du tinklai: „Darbo tinklas“ ir „Atakuojantis tinklas“.

3.7. Sistemos konfigūravimas

Norint, kad veiktų topologijos sukurtas tinklas ir virtualios mašinos prisijungtų prie interneto, reikėjo sukonfigūruoti virtualių mašinų interneto adapterius.

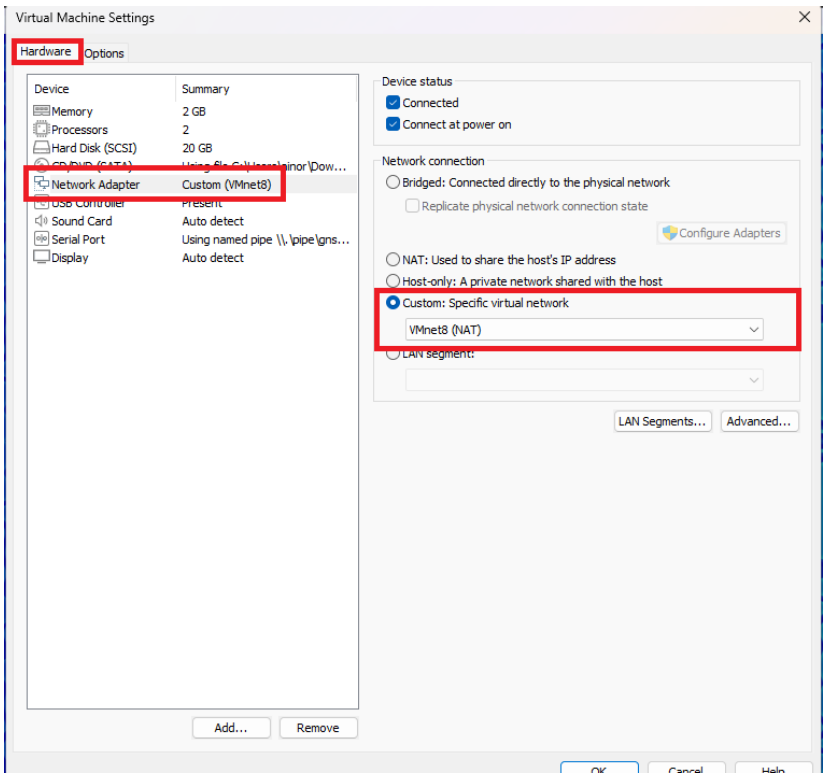
3.7.1. Virtualių mašinų interneto adapterių konfigūracija.

Virtualių mašinų interneto adapteriai buvo konfigūruojami per „*GNS3*“ įjungiant pasirinktą virtualią mašiną. Tada atsidarė „*VMware*“ virtualios mašinos langas, viršui dešinėje buvo spaudžiama „*Player*“, tada užvedama pelytė ant „*Removable Devices*“ toliau ant „*Network Adapter*“ ir galiausiai spaudžiama „*Settings*“, žiūrėti 3.15 paveiksle.



3.14 pav. Interneto adapterio konfigūravimo lango įjungimas

Paspaudus „*Settings*“ atsidaro „*Virtual Machine Settings*“ langas. „*Hardware*“ skirtuke buvo pasirinkta „*Network adapter*“ ir prie „*Network connection*“ buvo uždedama varnelė ant „*Custom: Specific virtual network*“. Tada apačioje buvo pasirinkta „*VMnet8 (NAT)*“ adapteris ir spaudžiama „*Ok*“. Visus pasirinkimus galima matyti pateiktame ekrano vaizde, 3.16 paveiksle.



3.15 pav. „*Virtual Machine Settings*“ langas

Tai padarius virtuali mašina prisijungė prie Interneto automatiškai, gali reikėti palaukti viena ar dvi minutes.

3.8. Įrankių įdiegimas

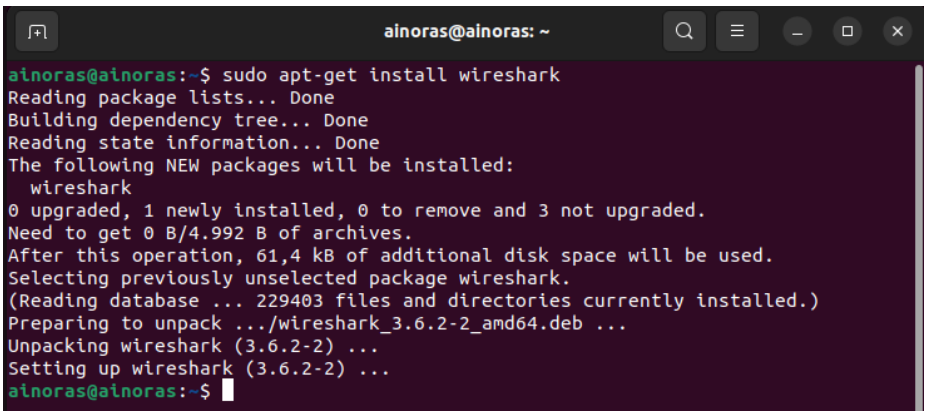
Simuliacijai atlikti, „*Ubuntu*“ virtualiai mašinai buvo reikalingi „*Wireshark*“ ir „*Tshark*“ įrankiai. Toliau nurodomos komandos, kuriu pagalba jie buvo sudiegti į „*Ubuntu*“ operacinę sistemą. Po sudiegimo buvo galima juos naudoti tinklo skenavimui.

3.8.1. „Wireshark“ ir „*Tshark*“ programų įdiegimas į „*Ubuntu“*

„*Wireshark*“ ir „*Tshark*“ įrankiai į operacinę sistemą buvo įdiegti per terminalą. Pirma buvo įjungta „*Ubuntu“* virtuali mašina per „*VMware*“ ir atidaromas terminalas. Siekiant įdiegti šias programas terminale buvo rašomos šios komandos:

„*Wireshark*“ įdiegimas (komanda matoma 3.17 pav.):

sudo apt-get install wireshark

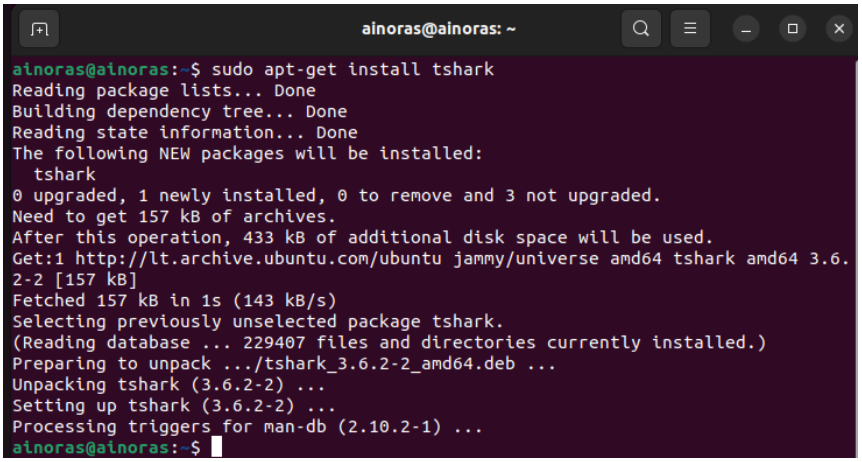


3.16 pav. „*Wireshark*“ įdiegimas

Įdiegiant „*Wireshark*“ programą gali paprašyti administratoriaus slaptažodžio, jį suvedus, programa buvo įdiegta. „*Wireshark*“ programa įdiegta.

„*Tshark*“ įdiegimas (komanda matoma 3.18 pav.):

Sudo apt-get install tshark



3.17 pav. „*Tshark*“ įdiegimas

Įdiegiant „*Tshark*“ programą gali paprašyti administratoriaus slaptažodžio, jį suvedus, programa buvo įdiegta. „*Tshark*“ programa įdiegta.

3.8.2. Apibendrinimas

Į „*Ubuntu*“ virtualią mašiną buvo įdiegti „*Wireshark*“ ir „*Tshark*“ programiniai įrankiai pasinaudojant „*Ubuntu*“ terminalu.

3.9. Išvados ir apibendrinimai

1. Sukurta sistemos architektūrą, kuri remiasi dviem pagrindiniais elementais – „*GNS3*“ ir „*VMware*“ platformomis.
2. Sukurta tinklo topologija pasinaudojant „*VMware*“ ir „*GNS3*“ platformomis. Buvo Užtikrinta platformų integracija. „*VMware*“ platforma palaikė virtualias mašinas naudojamas tinklo topologijoje, o „*GNS3*“ platforma buvo atsakinga už šių virtualių mašinų veikimą tarpusavyje ir komunikavimą.
3. Buvo Apsibrėžti reikalavimai aparatūrai. Išsiaiškinta, kad yra reikalingas ganėtinai galingas kompiuterio procesorius, didelis atminties kiekis, talpi saugykla ir geras tinklo ryšys norint atlikti kibernetinės žvalgybos simuliacijas nuosavame kompiuteryje.
4. Į kompiuterį įdiegta „*GNS3*“ ir „*VMware*“ platformos, jos buvo pasirengtos tolesniam naudojimui.
5. Buvo įdiegti trys serveriai su skirtingomis operacinėmis sistemomis – „*MS windows 10*“, „*Ubuntu*“, „*Kali Linux*“. Visi serveriai buvo virtualizuojami „*VMware*“ platformos pagalba.
6. Integruotos virtualios mašinos į „*GNS3*“ platformą. Buvo susipažinta su „*GNS3*“ grafine sąsaja, siekiant patogiai ir greitai ja naudotis.
7. Susikurta tinklo topologiją, kurioje buvo atliekama kibernetinės žvalgybos simuliacija.
8. Sukonfigūruotos virtualios mašinos, kad jos galėtų prisijungti į tinklą ir komunikuoti tarpusavyje.
9. Į virtualų serverį „*Ubuntu*“ buvo įdiegti „*Wireshark*“ ir „*Tshark*“ įrankiai, kad atlikdamas kibernetinę žvalgybą galėčiau atlikti tinklo srauto skenavimus ir gauti, bei išsisaugoti rezultatus.

4. EKSPERIMENTINĖ – PRAKTINĖ DALIS

Šioje dalyje, buvo pradedami eksperimentai. Buvo aprašyta, kaip buvo įsitikinta, kad simuliacinė aplinka yra paruošta naudojimui, sukuriami simuliacijos scenarijai lengvesniam simuliacijos įsivaizdavimui. Buvo pateikiamas programinių įrankių parengimas ir galiausiai simuliacijos atlikimas, duomenų užfiksavimas.

4.1. Simuliacinės aplinkos parengimas

Prieš pradėdamas skenavimus ir bandymus atliekant kibernetinę žvalgybos simuliaciją, buvo įsitikinta, kad visos virtualios mašinos įjungtos, ir gali komunikuoti tarpusavyje – nestringa ir yra užtikrintas kokybiškas veikimas. Siekiant tai užtikrinti, per „*GNS3*“ platformą buvo įjungtos visos trys „*VMware*“ virtualios mašinos – „*Ubuntu*“, „*Kali Linux*“ ir „*MS windows 10*“.

„*MS windows 10*“ virtualioje sistemoje buvo įsijungta Komandinę eilutė ir joje sužinotas serverio IP adresas. Buvo panaudota „*ping*“ komanda ir patikrinta ar „*MS Windows 10*“ mašina susisiekia su „*Ubuntu*“.

Komandinėje eilutėje buvo naudotos šios komandos:

ipconfig

ping 192.168.200.133

„*Ubuntu*“ virtualioje mašinoje buvo įjungti du terminalai. Viename terminale išsiaošlomta „*Ubuntu*“ IP adresas, kad galėčiau patikrinti „*Ubuntu*“ operacinės sistemos komunikacija su kitomis virtualiomis mašinomis. O kitame terminale buvo patikrinta ar „*Ubuntu*“ gali komunikuot su „*MS Windows 10*“ serveriu.

Terminaluose buvo naudotos šios komandos:

ifconfig

ping 192.168.200.134

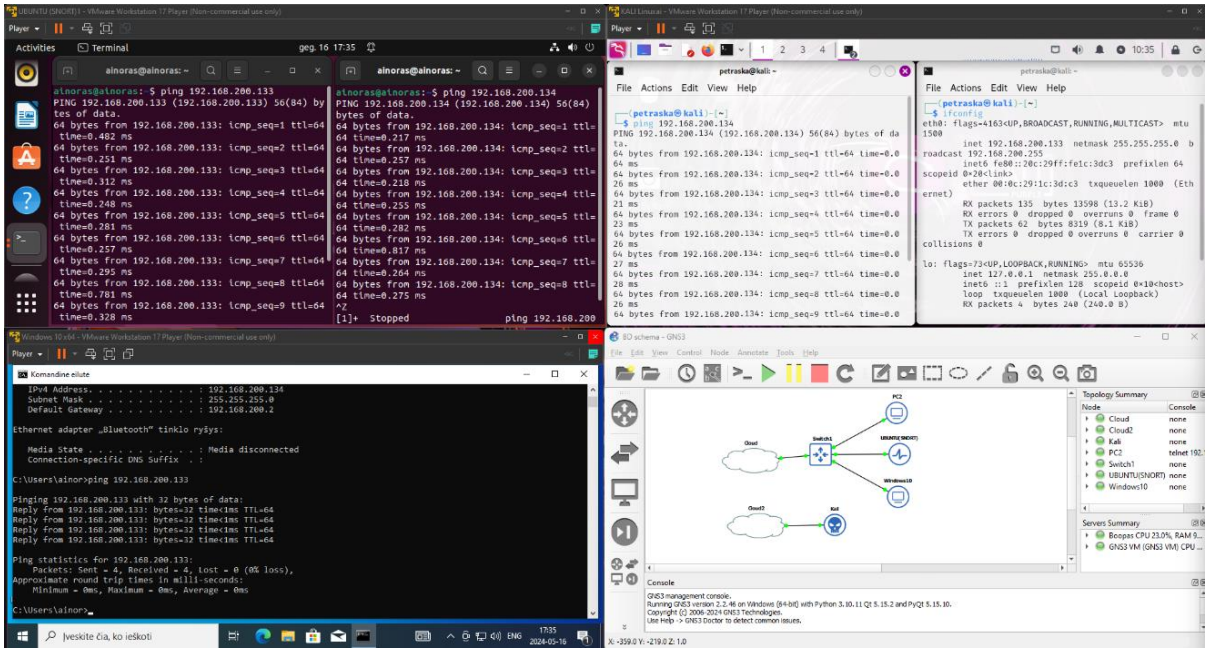
„*Kali Linux*“ virtualioje mašinoje buvo įjungti du terminalai. Abu terminalai buvo naudojami siekiant užtikrinti ryšį su „*MS Windows 10*“ ir „*Ubuntu*“ serveriais.

Terminaluose buvo naudotos šios komandos:

ping 192.168.200.133

ping 192.168.200.134

Tuo pačiu metu buvo stebima „*GNS3*“ grafinė sąsaja. „*Topology Summary“* skirtuke buvo galima pastebėti, jog visi virtualūs įrenginiai buvo taisyklingai įjungti, taip pat „*Console*“ skirtuke nebuvo jokių įspėjimų, klaidų. Viskas buvo sukonfigūruota taisyklingai, veikė be klaidų. Visos komandos ir visi patikrinimai matomi 4.1 paveiksle.



4.1 pav. Virtualių Mašinų komunikacijos užtikrinimas

Simuliacinė aplinka buvo paruošta kibernetinės žvalgybos simuliacijai, virtualios mašinos komunikavo tarpusavyje, viskas buvo sukonfigūruota taisyklingai.

4.2. Simuliacijos scenarijų parengimas

Prieš atliekant simuliacijas, buvo svarbu susikurti scenarijus, kad būtų galima lengviau įsivaizduoti atliekamą simuliacija.

4.2.1. Scenarijaus aprašymas

Buvo simuliuojamas kibernetinės žvalgybos scenarijus. Šiame scenarijuje buvo dvi komandos: „Atakuotojai“ ir „Darbuotojai“. „Atakuotojai“ buvo atsakingi už „Atakuojantis tinklas“ tinklą, o „Darbuotojai“ už „Darbo tinklas“ tinklą – žiūrėti „pav. 13 Kuriamo tinklo topologija“. Komandos „Atakuotojai“ tikslas buvo skenuoti „Darbuotojai“ tinklą ir išsiaiškinti visus tinklo atvirus prievadus, jie naudojosi „*Kali Linux*“ sistema ir įvairiais jos įrankiais siekdami išsiaiškinti kokie prievadai yra atviri. Komandos „Darbuotojai“ tikslas buvo apsisaugoti nuo įsilaužėlių ir apsaugoti tinklą. „Darbuotojai“ tinkle turėjo „*Ubuntu*“ operacinę sistemą, kurios paskirtis buvo stebėti ir

analizuoti tinklo srautą pasinaudojant „*Wireshark*“ ir „*Tshark*“ įrankiais. „Darbuotojai“ Pastebėję neautorizuotą prieigą prie tinklo išsianalizuoja pažeidimų srautą ir identifikuoja tinklo spragas, bei silpnybes, galiausiai imdamiesi veiksmų sustiprinti tinklo apsaugą.

4.2.2. Scenarijaus veiksmų eiga

Scenarijų veiksmų eiga, kurie buvo vykdomi atliekant simuliaciją:

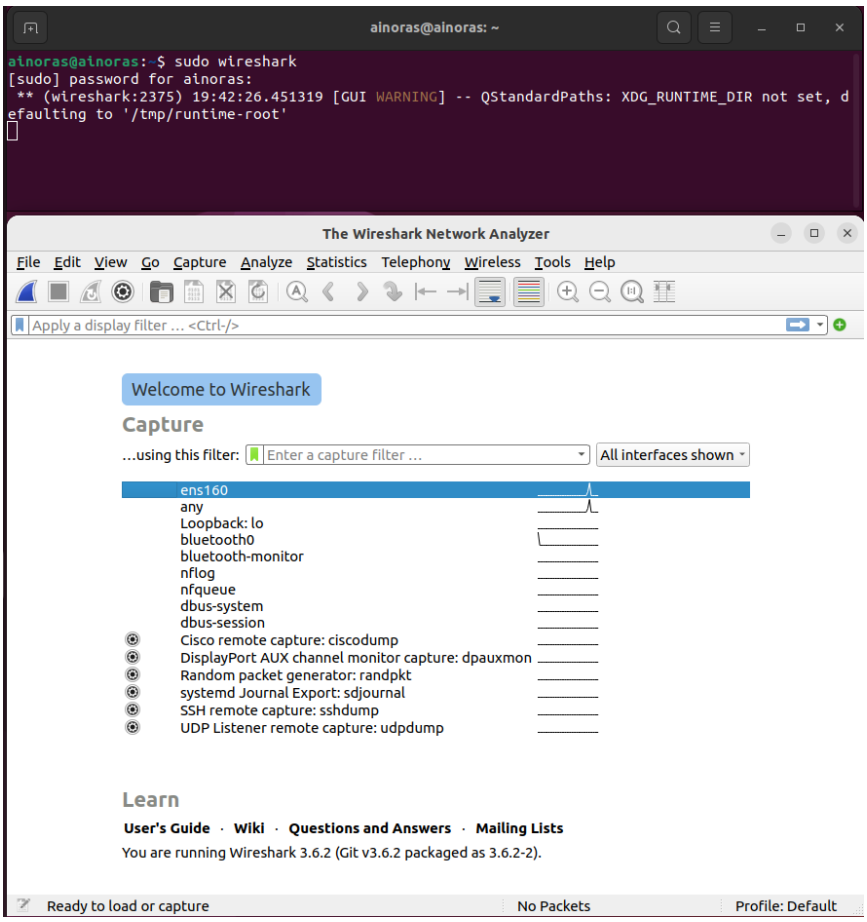
1. „Darbuotojų“ komanda stebi savo tinklo srautą su „*Wireshark*“ ir „*Tshark*“ įrankiais, siekdami aptikti srauto anomalijas.
2. „Atakuotojai“ pradeda prievadų skenavimo operacijas naudodami „*Nmap*“ įrankį, siekdami aptikti atvirus prievadus ir tinklo silpnybes, bei spragas.
3. „Darbuotojų“ komanda stebi ir fiksuoja tinklo srauto anomalijas, kurias atlieka „Atakuotojų“ komanda.
4. „Darbuotojų“ komanda reaguoja į įvykius, ir imasi konkrečių priemonių siekiant sustabdyti „Atakuotojų“ veiklą ir užkirsti kelią įsibrovimui.
5. Remiantis užfiksuotais tinklo srauto rezultatais, „Darbuotojai“ įdiegia patobulinimus tinklui ir sustiprina tinklo saugumą.

4.3. Simuliacija

Susikūrus scenarijų, buvo pradėta jį įgyvendinti. Pirma buvo įsijungta „*Wireshark*“ programa, srauto stebėjimui ir pradedamas tinklo skenavimas.

4.3.1. „*Wireshark*“ pasiruošimas tinklo srauto stebėjimui

„*Ubuntu*“ operacinėje sistemoje buvo įjungta „*Wireshark*“ programa administratoriaus teisėmis. Tai buvo padaryta į terminalą įrašant komandą „*sudo wireshark*“ ir tada suvedus administratoriaus slaptažodį, buvo įsijungtas „*Wireshark*“ įrankis. Terminalo ir „*Wireshark*“ ekrano vaizdas pateikiamas 4.2 paveiksle.



4.2 pav. „*Wireshark*“ programos įjungimas

Tada buvo pasirinktas tinklo adapteris, kuris bus stebimas stebėsime, šiuo atveju buvo pasirinkta „*ens160*” interneto adapteris. Pasirinkus interneto adapterį buvo pradėtas tinklo srauto stebėjimas. „*Wireshark*“ įrankis buvo paruoštas tinklo srauto stebėjimui.

4.3.2. Prievadų skenavimas

„*Kali Linux*“ sistemoje buvo įjungtas terminalas ir prie jo prisijungta su „*root*“ teisėmis. Tai buvo padaryta įvedus komandą „*sudo su* – " komandą matoma 4.3 paveiksle.

4.3 pav. Prisijungimas „*root*“ teisėmis

Prisijungus „*root*“ teisėmis, buvo pradėti tinklo skenavimai su „*Nmap*“ įrankiu. Atlikti šių prievadų tipų skenavimai: TCP, UDP, SYN, FIN, ACK, NULL, XMAS.

TCP Skenavimui atlikti buvo naudojama ši komanda:

nmap -sT <taikinys>

4.4 pav. TCP skenavimo metodas

4.4 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas TCP skenavimas. Ekrano vaizde galima matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

UDP skenavimui atlikti, buvo naudojama ši komanda:

nmap -sU -p 1-10 <taikinys>

4.5 pav. UDP skenavimo metodas

4.5 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas UDP skenavimas. Ekrano vaizde galima matyti gautus rezultatus. UDP skenavimo metu buvo naudojama komanda su papildomu parametru. Buvo pridėtas „*-p 1-10*" parametras, nes skenavimas užtrunka ilgai, ir rezultatuose rodys visus prievadus, nesvarbu jie atviri ar uždari. Dėl šio parametro „*Nmap*" pagalba buvo nuskenuoti tik pirmi 10 prievadų. Išsiaiškinta, kad pirmi 10 prievadų yra uždari ir kokias paslaugas jie naudojo.

SYN skenavimui atlikti, buvo naudojama ši komanda:

nmap -sS <taikinys>

4.6 pav. SYN skenavimo metodas

4.6 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas SYN skenavimas. Ekrano vaizde galima matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

FIN skenavimui atlikti, buvo naudojama ši komanda:

nmap -sF <taikinys>

4.7 pav. FIN skenavimo metodas

4.7 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas FIN skenavimas. Ekrano vaizde galim matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

ACK skenavimui atlikti, buvo naudojama ši komanda:

nmap -sA <taikinys>

4.8 pav. ACK skenavimo metodas

4.8 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas ACK skenavimas. Ekrano vaizde galima matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

NULL skenavimui atlikti, buvo naudojama ši komanda:

nmap -sN <taikinys>

4.9 pav. NULL skenavimo metodas

4.9 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas NULL skenavimas. Ekrano vaizde galima matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

XMAS skenavimui atlikti, buvo naudojama ši komanda:

nmap -sX <taikinys>

4.10 pav. XMAS skenavimo metodas

4.10 paveiksle pateiktas ekrano vaizdas, kuomet buvo atliktas XMAS skenavimas. Ekrano vaizde galima matyti gautus rezultatus, atvirus prievadus ir jų naudojamas paslaugas.

4.3.3. Apibendrinimas

Atlikti septyni prievadų skenavimo metodai. Visi skenavimai atlikti taisyklingai ir gauti reikiami rezultatai. Skenavimų rezultatai nurodyti 1 lentelėje.

1 lentelė. Atlikti skenavimai

Eil. Nr.	Skenavimas	Skenavimo įgyvendinimas	Duomenų surinkimo įgyvendinimas	Pastabos
1.	nmap -sT	Taip	Taip	
2.	nmap -sU	Taip	Taip	Buvo skenuojami tik pirmi 10 prievadų.
3.	nmap -sS	Taip	Taip	
4.	nmap -sF	Taip	Taip	
5.	nmap -sA	Taip	Taip	
6.	nmap -sN	Taip	Taip	
7.	nmap -sX	Taip	Taip	

Buvo išsiaiškinta, kurie prievadai buvo atviri – 80 ir 5500. Taip pat buvo išsiaiškinta kokias paslaugas jie naudoja. Pavyko atlikti visus skenavimus. Kadangi UDP skenavimui buvo naudojama komanda su papildomu parametru, nuskenuoti buvo tik pirmi 10 prievadų.

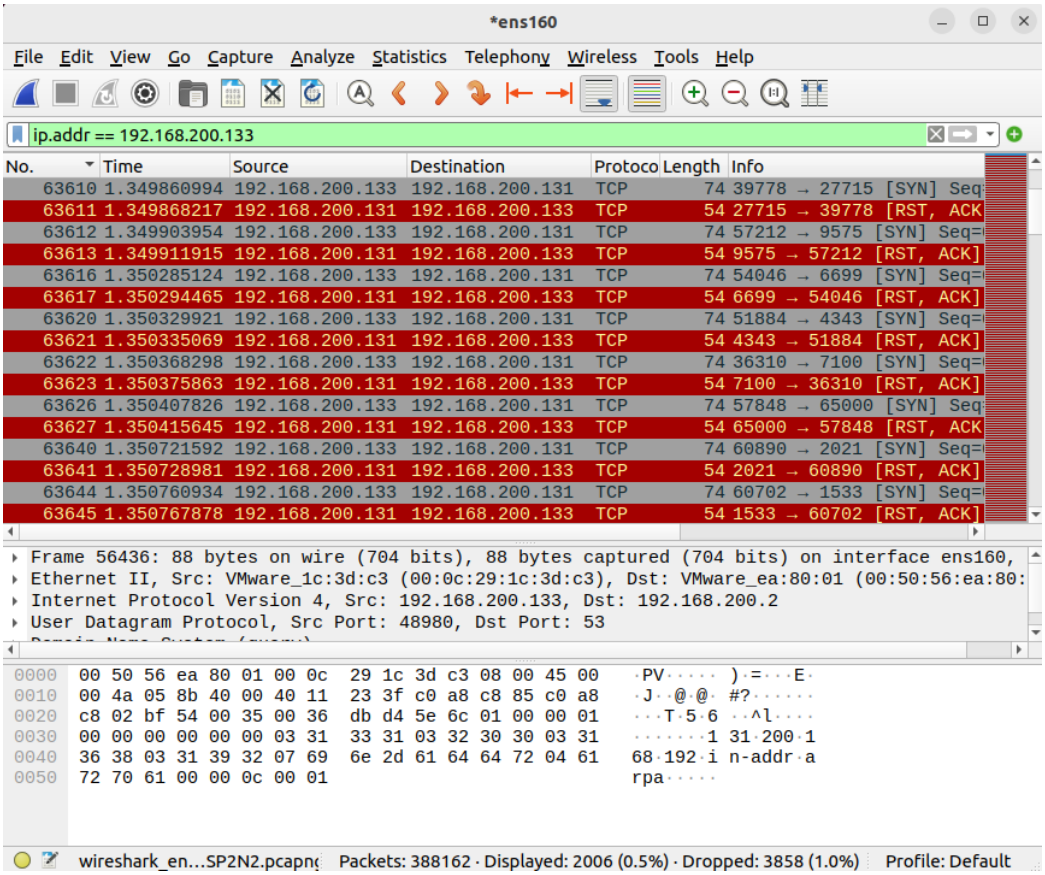
4.4. Duomenų stebėjimas ir fiksavimas

Atliekant skenavimus, tuo pačiu metu buvo atliktas duomenų fiksavimą „*Wireshark*“ pagalba. Gauti paketus buvo išsisaugoti „*Ubuntu*“ virtualioje mašinoje, tai leido paketus peržiūrėti ateityje.

4.4.1. Duomenų stebėjimas su Wireshark

Duomenų stebėjimui ir fiksavimui buvo naudojama „*Ubuntu*“ operacinėje sistemoje įjungta „*Wireshark*“ programa. Buvo naudotas „*ip.addr == 192.168.200.133” filtras,* kad „*Wireshark*“ išfiltruotų tik „*Kali Linux*“ sukuriamus paketus tinkle. „*Wireshark*“ buvo įjungta tuo metu, kai buvo atliekamas „*Nmap*“ skenavimas iš „*Kali Linux*“ operacinės sistemos.

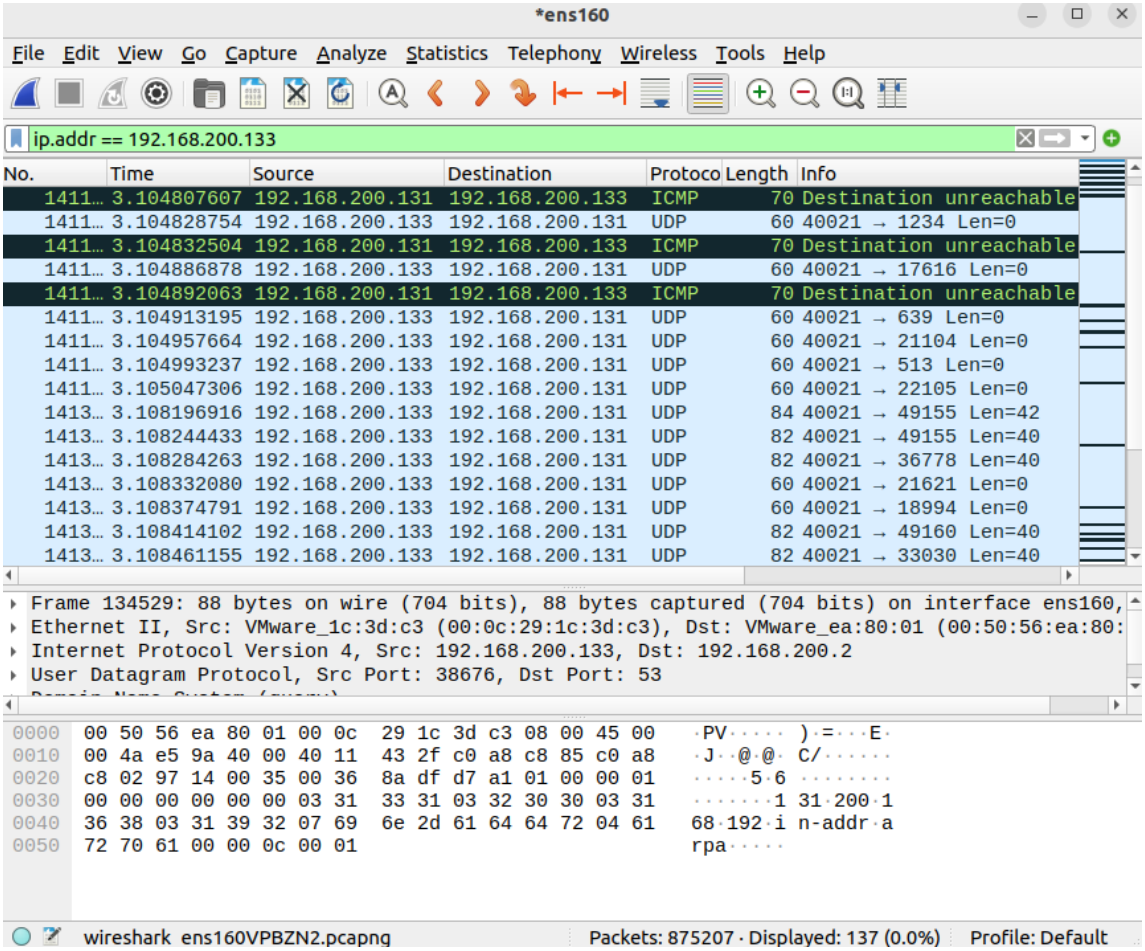
TCP skenavimo metu buvo aptikti paketai, kurie yra matomi 4.11 paveiksle.



4.11 pav. „*Wireshark*“ TCP skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matoma kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė SYN paketus, kas yra būdinga TCP skenavimui.

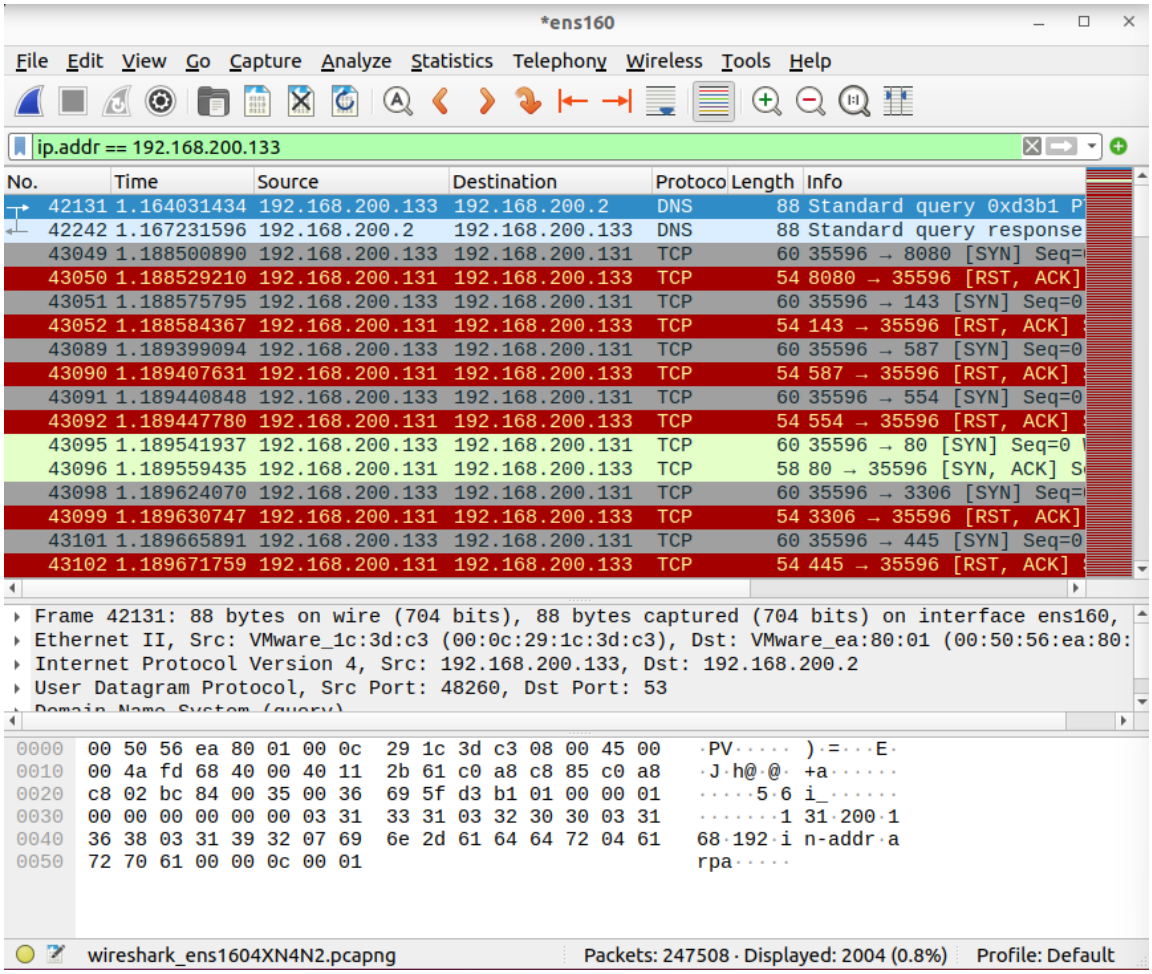
UDP skenavimo metu buvo aptikti paketai, kurie yra matomi 4.12 paveiksle.



4.12 pav. „*Wireshark*“ UDP skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matome kokiais paketais buvo keičiamasi tarp serverių.

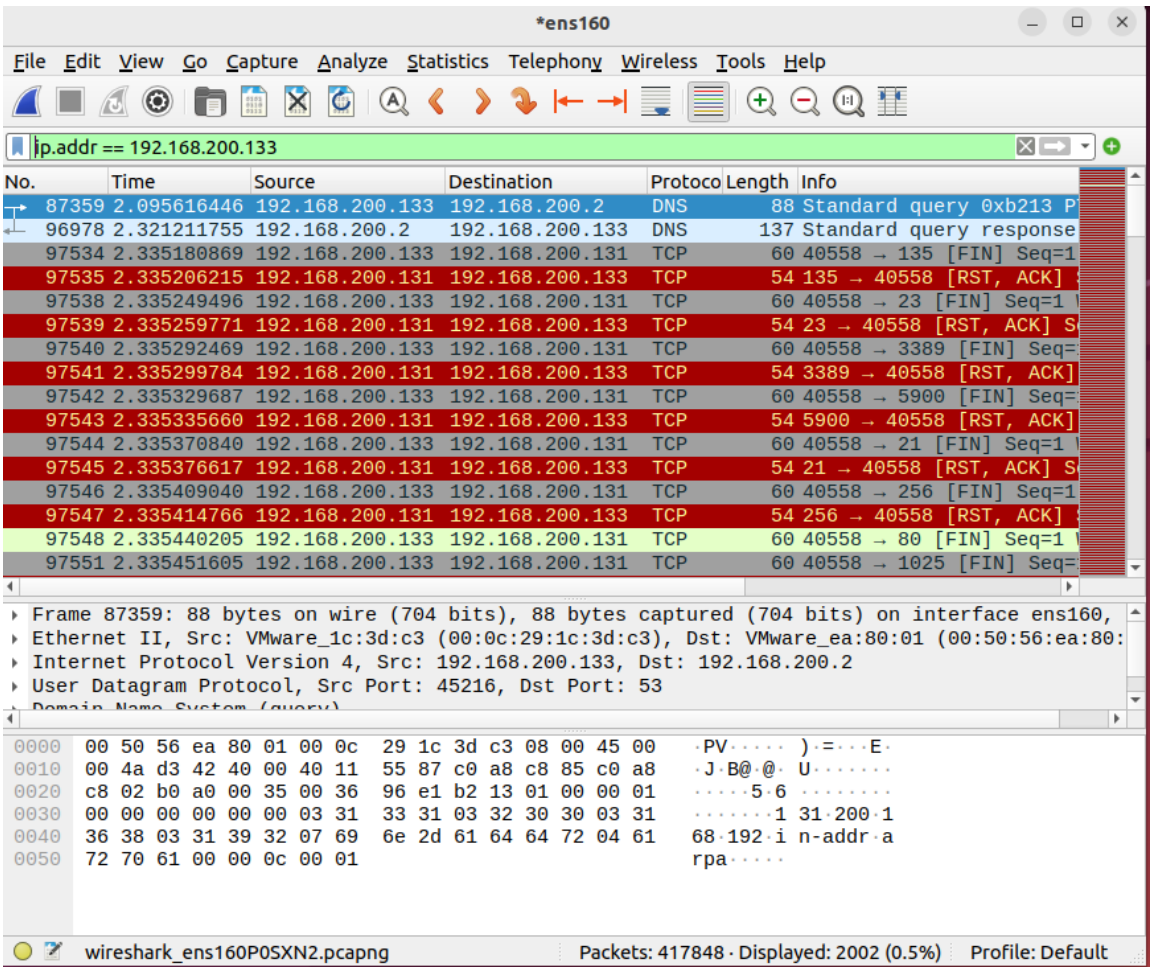
SYN skenavimo metu buvo aptikti paketai, kurie yra matomi 4.13 paveiksle.



4.13 pav. „*Wireshark*“ SYN skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matoma kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė SYN paketus, kas yra būdinga SYN skenavimui.

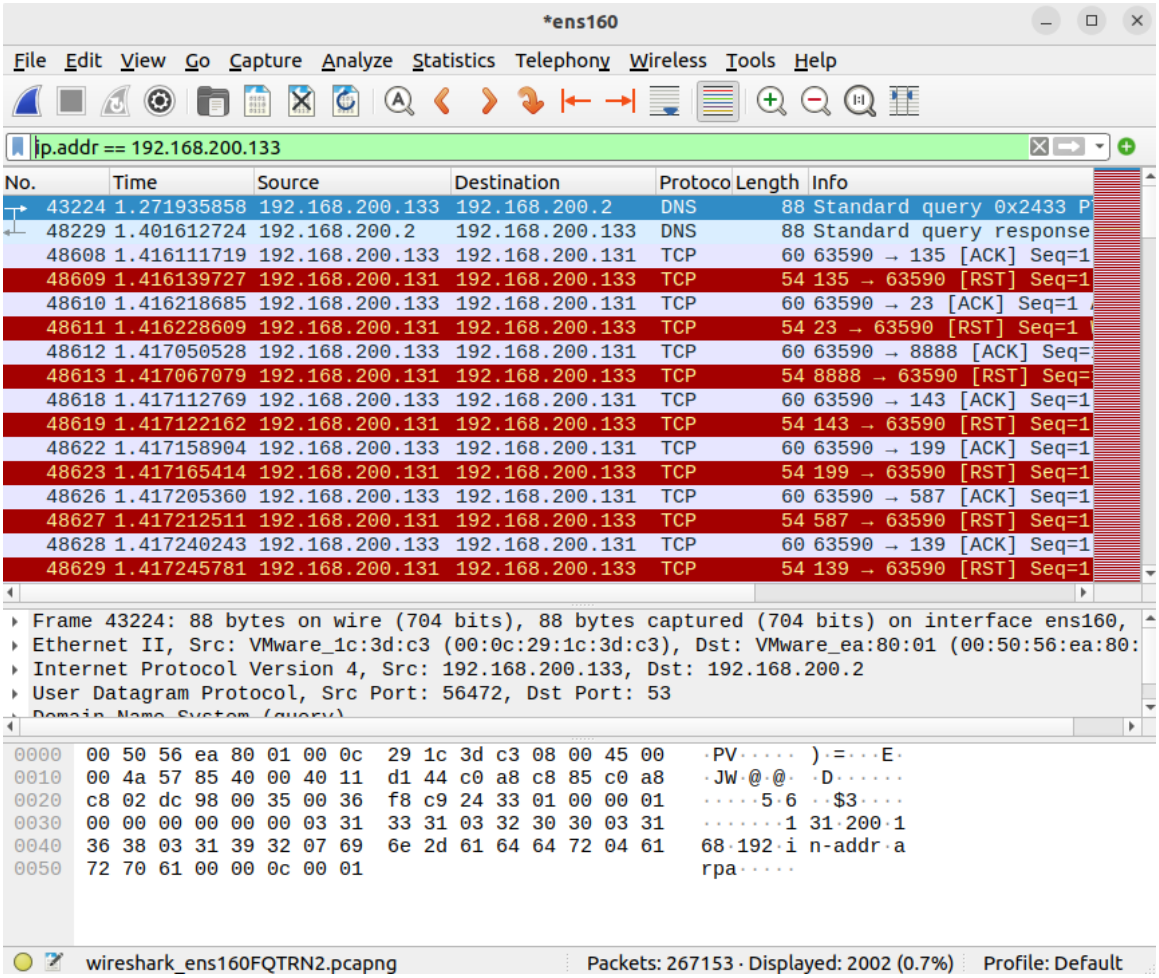
FIN skenavimo metu buvo aptikti paketai, kurie yra matomi 4.14 paveiksle.



4.14 pav. „*Wireshark*“ FIN skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matome kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė FIN paketus, kas yra būdinga FIN skenavimui.

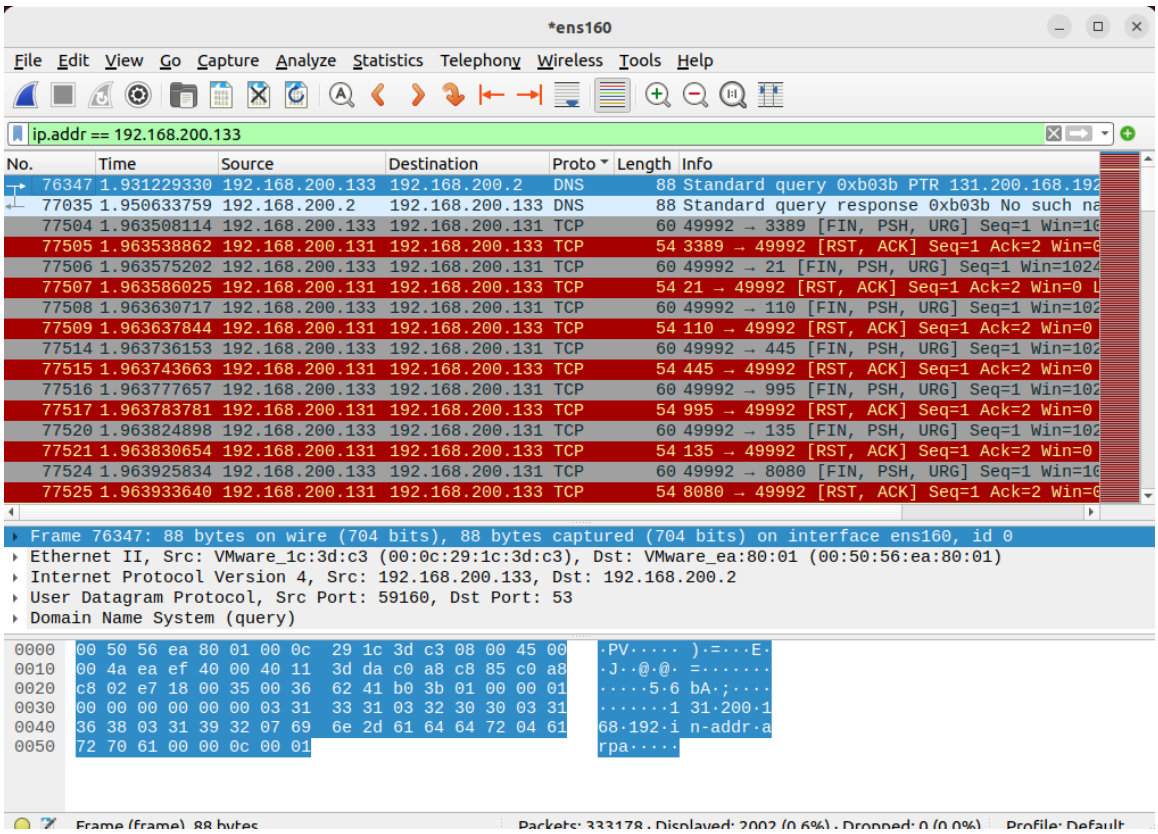
ACK skenavimo metu buvo aptikti paketai, kurie yra matomi 4.15 paveiksle.



4.15 pav. „*Wireshark*“ ACK skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matoma kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė ACK paketus, kas yra būdinga ACK skenavimui.

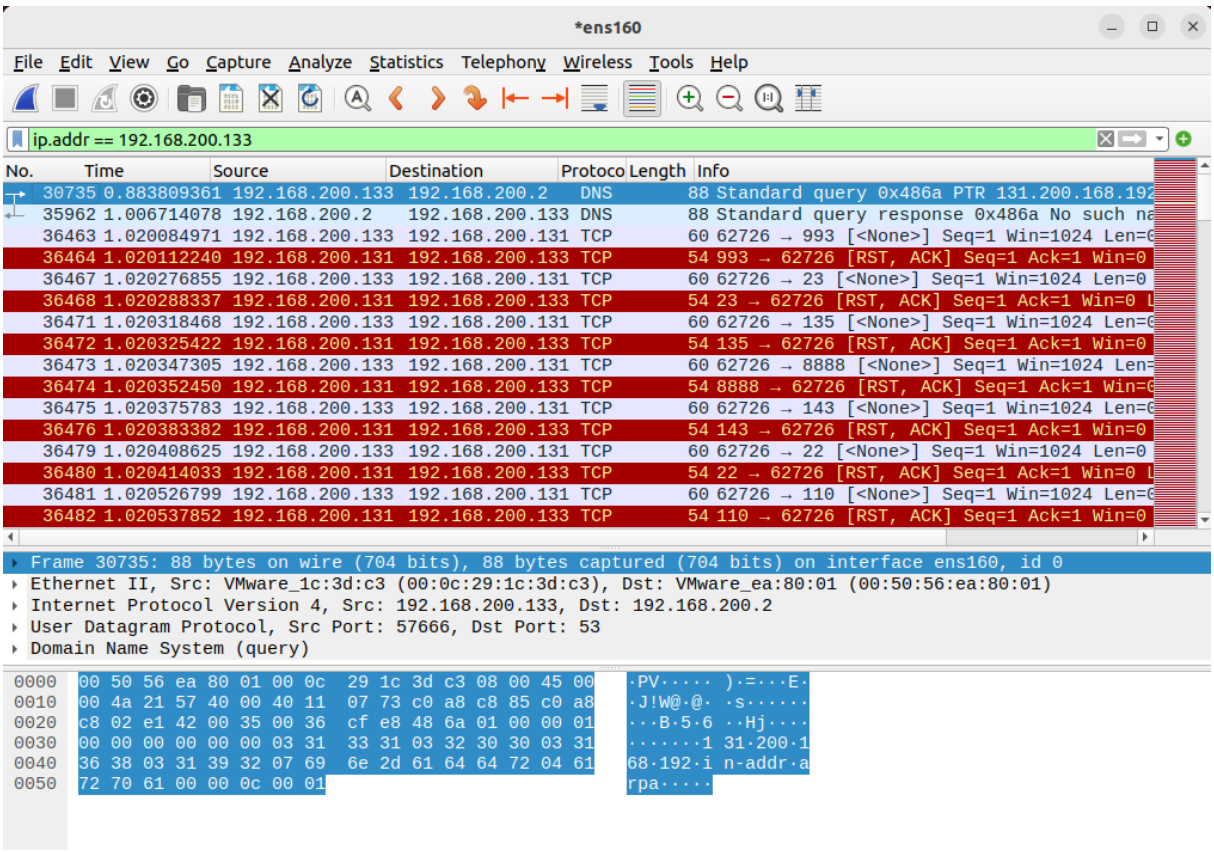
XMAS skenavimo metu buvo aptikti paketai, kurie yra matomi 4.16 paveiksle.



4.16 pav. „*Wireshark*“ XMAS skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matoma kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė FIN, PSH, URG paketus, kas yra būdinga XMAS skenavimui.

NULL skenavimo metu buvo aptikti paketai, kurie yra matomi 4.17 paveiksle.



4.17 pav. „*Wireshark*“ NULL skenavimo metu siunčiami paketai

Stebint tinklo srautą buvo aptikti paketai siunčiami iš 192.168.200.133, „*info*” skiltyje matoma kokiais paketais buvo keičiamasi tarp serverių. Šiuo atveju „*Kali Linux*“ skenavimo metu siuntė tuščius (angl. *none*) paketus, kas yra būdinga NULL skenavimui.

4.4.2. Apibendrinimas

Naudodamas „Wireshark“ programą, buvo aptiktos tinklo anomalijos, kurios buvo sukeltos kitos, „*Kali Linux*“, virtualios mašinos. Iš gautų duomenų galima atpažinti kokio tipo buvo atliekami skenavimai. Matomas puolėjo („*Kali Linux*“) IP adresas. Visų aptikimų rezultatai pateikiami 2 lentelėje.

2 lentelė. Aptikti skenavimai

Eil. Nr.	Skenavimas	Skenavimo aptikimas	Duomenų surinkimo įgyvendinimas
1.	TCP	Taip	Taip
2.	UDP	Taip	Taip
3.	SYN	Taip	Taip
4.	FIN	Taip	Taip

5.	ACK	Taip	Taip
6.	XMAS	Taip	Taip
7.	NULL	Taip	Taip

Pasinaudojus „*Wireshark*“ programa pavyko aptikti visų skenavimų būdus, jų siunčiami paketai, buvo užfiksuoti ir išsisaugoti „*Ubuntu*“ virtualioje mašinoje, tolesniam naudojimui.

4.5. Apibendrinimas ir išvados

1. Buvo sėkmingai atlikta kibernetinės žvalgybos simuliacija.
2. Prieš atlikdamas kibernetinės žvalgybos simuliaciją, buvo užtikrinta, kad visos virtualios sistemos gali tarpusavyje komunikuoti.
3. Prieš atlikdamas kibernetinės žvalgybos simuliaciją, buvo parengti ir aprašyti scenarijai.
4. Atlikta kibernetinės žvalgybos simuliaciją. Tai yra, iš „*Kali Linux*“ operacinės sistemos buvo atlikti tinklo skenavimus su „*Nmap*“ įrankiu ir tuo pačiu metu su „*Ubuntu*“ operacine sistema ir „*Wireshark*“ programine įranga, buvo aptiktos tinklo srauto anomalijas, kurios buvo sukeltos „*Nmap*“ skenavimų.
5. „*Wireshark*“ pagalba buvo užfiksuoti ir išsaugoti tinklo srauto anomalijų duomenys.

IŠVADOS

1. Atlikta kibernetinės žvalgybos ir jos atakų literatūrinę analizė ir išsiaiškintos pagrindines atakų rūšis, jų vykdymo procesai, etapai ir naudojamos priemonės. Tai leido suprasti kokios situacijos gali būti simuliuojamos simuliacinėse aplinkose.

2. Atlikus analizę įvertintos virtualios aplinkos ir nustatytos tinkamiausios platformos kibernetinės žvalgybos simuliavimui. Pasirinktos „*GNS3*“ ir „*VMware*“ platformos, kurios geriausiai atitiko projektuojamos aplinkos poreikius ir leido lengvai, bei efektyviai atlikti kibernetinių atakų scenarijų simuliavimą.

3. Suprojektuotas simuliacinis tinklas tenkinantis reikalavimus, poreikius ir tikslus, išsikeltus darbo pradžioje. Šis tinklas sukurtas atsižvelgiant į patikimumą, lankstumą ir duomenų generavimo, bei analizės galimybes.

4. Išbandyta sukurta simuliacija, kuri pasiteisino ir parodė savo veiksmingumą, bei efektyvumą atliekant aktyviosios kibernetinės žvalgybos bandymus ir testavimus. Jos pagalba buvo įgyvendintas kibernetinės žvalgybos scenarijus, leidęs įvertinti simuliacijos veiksmingumą.

LITERATŪRA IR KITI INFORMACIJOS ŠALTINIAI

1. Agghey, A. Z., Mwinuka, L. J., Pandhare, S. M., Dida, M. A., & Ndibwile, J. D. (2021). Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach. Symmetry. https://doi.org/10.3390/sym13112192
2. Al-Khorazmi, M. (2023). Network operating systems. Retrieved from https://uzresearchers.com/index.php/XAFT/article/view/863
3. Aslan, O., Aktug, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. MDPI Electronics. https://doi.org/10.3390/electronics12061333
4. Charles Raul, A. (2019). The privacy data protection and cybersecurity law review. Retrieved from https://datamatters.sidley.com/wp-content/uploads/sites/2/2019/11/The-Privacy-Data-Protection-and-Cybersecurity-Law-Review-Edition-6.pdf
5. GNS3. (n.d.). Documentation. Retrieved from https://docs.gns3.com/
6. Gottsegen, G. (2023). Cybersecurity review: A focus on machine learning. BuiltIn. Retrieved from https://builtin.com/artificial-intelligence/machine-learning-cybersecurity
7. Kumar, S. (2023). Anomaly detection algorithms in cybersecurity: A review. BuiltIn. Retrieved from https://builtin.com/machine-learning/anomaly-detection-algorithms
8. Li, J. (2020). Vulnerabilities Mapping based on OWASP-SANS: a Survey for Static Application Security Testing (SAST). https://doi.org/10.1038/joc.2020.45
9. Mazurczyk, W., & Caviglione, L. (2021). Cyber reconnaissance techniques. ACM Computing Surveys. https://doi.org/10.1145/3418293
10. Nasser, A., Nasser, H., Ahmad, A., Maynard, S. B., & Siddiqui, A. M. (2023). Moving towards agile cybersecurity incident response: A case study exploring the enabling role of big data analytics-embedded dynamic capabilities. https://doi.org/10.1016/j.jcm.2023.02.007
11. Pittman, J. M. (2023). Machine Learning and Port Scans: A Systematic Review. https://doi.org/10.1016/j.jcr.2023.01.005
12. Pykes, K. (2023). Classification versus clustering in machine learning. https://doi.org/10.1016/j.dsj.2023.03.002
13. Scherb, C., Heitz, L. B., Grieder, H., & Maurer, M. (2023). A cyberattack simulation for teaching cybersecurity. Retrieved from https://www.researchgate.net/profile/Christopher-Scherb/publication/370739115_A_Cyber-Attack_Simulation-for-Teaching_Cybersecurity/links/6478d9aa2cad460a1be92da0/A-Cyber-Attack-Simulation-for-Teaching-Cybersecurity.pdf

14. Shah, M., Ahmed, S., & Khan, H. (2019). Penetration Testing Active Reconnaissance Phase - Optimized Port Scanning With Nmap Tool. Retrieved from https://www.researchgate.net/publication/332106249_Penetration_Testing_Active_Reconnaissance_Phase_-Optimized_Port_Scanning_With_Nmap_Tool

15. Stanham, L. (2023). Cybersecurity 101: Machine learning in cybersecurity. Retrieved from https://www.crowdstrike.com/cybersecurity-101/machine-learning-cybersecurity/

16. Tigner, M., Wimmer, H., & Rebman, C. M. (2021). Analysis of Kali Linux Penetration Tools: A Survey of Hacking Tools. https://doi.org/10.1109/ACCESS.2021.3092092

17. Ubuntu. (n.d.). Documentation. Retrieved from https://help.ubuntu.com/

18. VMware. (n.d.). Documentation. Retrieved from https://docs.vmware.com/

19. Wireshark. (n.d.). Documentation. Retrieved from https://www.wireshark.org/docs/