VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORACINIŲ SISTEMŲ KATEDRA

Nikita Malyškinas

„MEDAUS PUODYNĖS“ KONCEPCIJOS PRITAIKYMAS DAIKTŲ IR PASLAUGŲ INTERNETUI
ADAPTATION OF „HONEYPOT“ CONCEPTION TO INTERNET OF THINGS AND SERVICES

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga, 6211BX014

Informacijos ir informacinių technologijų saugos specializacija

Informatikos inžinerija

Vilnius, 2020

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacijos ir informacinių technologijų saugos** programos magistro baigiamasis darbas

Pavadinimas	**„Medaus puodynės'' koncepcijos pritaikymas daiktų ir paslaugų internete.**
Autorius	**Nikita Malyškinas**
Vadovas	**Nikolaj Goranin**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamas "medaus puodynės" veikimo principas ir šios koncepcijos pritaikymas daiktų ir paslaugų internete. Tyrimo tikslas - pritaikyti „medaus puodynės" koncepciją daiktų ir paslaugų interneto įrenginiams. Darbą sudaro kelios dalys: įvadas, susijusios literatūros apžvalga, siūlomo metodo aprašymas,siūlomo metodo realizacija, siūlomo metodo testavimas. Susijusios literatūros apžvalgoje buvo analizuojami esamos "medaus puodynės" skirtos daiktų ir paslaugų interneto įrenginiams. Atlikus literatūros analizę buvo pasiūlytas metodas, kaip realus daiktų ir paslaugų interneto įrenginys gali būti paverčiamas "medaus puodyne". Buvo atlikti bandymai ir išsiaiškinta ar sukurta "medaus puodynė" atitinka keliamus reikalavimus.

Prasminiai žodžiai: Daiktų ir paslaugų internetas, medaus puodynė, virtualizacija

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Information and Information Technologies Security** study programme Master Graduation Thesis

Title	**"Honeypot" concept adaptation to internet of things and services**
Author	**Nikita Malyškinas**
Academic supervisor	**Nikolaj Goranin**

Thesis language: Lithuanian

Annotation

The final master's thesis examines the principle of operation of the "honeypot" and the application of this concept to the internet of things and services. The aim of the study is to apply the concept of "honeypot" to internet of things and services. The work consists of several parts: introduction, review of related literature, description of the proposed method, implementation of the proposed method, testing of the proposed method. A review of the relevant literature analyzed existing "honey pots" for IoT devices. After literature review a method has been proposed for how a real internet of things and services device can be turned into a "honey pot". Tests have been carried out to verify that created "honey pot" meets all requirements.

Keywords: Honeypot, internet of things, virtualization

Turinys

Paveikslų sąrašas

Lentelių sąrašas

Santrumpų aiškinamasis žodynas

IP – Internet protocol

WPAN – Wireless personal area network

API – Application programming interface

DDoS – Distributed denial of service

IT – Informacinės technologijos

SNMP – Simple network management protocol

SCADA – Supervisory control and data acquisition

MIPS – Microprocessor without interlocked pipelined stages

1. Įvadas

Daiktų internetas – tai tarptinklinė elektroninių įrenginių sąveika, kuri leidžia rinkti ir dalyti informaciją. Daiktų internetas leidžia aptikti ir manipuliuoti įtaisais per internetą, taip pat leidžia „daiktams“ patiems spręsti, kokius veiksmus atlikti, remiantis kitų „daiktų“ atsiųsta informacija. Toks valdymo būdas yra daug efektyvesnis, nes kompiuteris gali išanalizuoti duomenis daug greičiau ir priimti geresnius sprendimus, nei žmogus. Daiktų internetas suteikia galimybę integruoti kompiuterines sistemas į fizinį pasaulį. Kompiuteris jau dirba ne tik su virtualia aplinka, bet gali gauti informacijos ir iš fizinės aplinkos be žmogaus įsikišimo. Žurnalas „Forbes“ daiktų internetą pavadino nauja revoliucija, kuri iš esmės pakeis ne tik verslą, bet ir mūsų gyvenimo stilių. Jau dabar naujos kartos automobiliai turi interneto jungtį, kuri leidžia jiems ne tik apskaičiuoti optimaliausią maršrutą, bet ir pranešti kitiems, ta pačia linkme važiuojantiems automobiliams, apie pavojus kelyje arba oro sąlygas. „Išmanūs“ namai gali būti prijungiami prie jūsų išmaniojo telefono ir, suskambus žadintuvui, kavos aparatas pradės gaminti kavą. Ekspertų iš „International Data Company“ kompanijos manymu, 2025 metais daiktų internetą sudarys beveik 50 milijardų įtaisų (International Data Company [IDC], 2019).

Tačiau naujos technologijos suteikia ne tik naujų patogumų, bet ir naujų problemų. Daugėjant daiktų interneto įrenginių, daugėja ir asmenų, bandančių į juos įsilaužti. Įsilaužimas į daiktų interneto įrenginį yra daug pavojingesnis negu į kompiuterį. Iš jūsų kompiuterio gali būti pavogta informacija ir jūs galite patirti finansinių nuostolių, tačiau įjungus namo šildymo sistemą visu pajėgumu gali įvykti sprogimas ir šiuo atveju gali būti prarasta gyvybė. Daiktų internetas yra naudojamas ne tik paprastų vartotojų, daiktų internetą aktyvai pritaiko ir korporacijos. Daiktų internetas aktyvai naudojamas energetikos srityje. Jutikliais yra stebimi ir valdomi elektros tinklai. Logistikos srityje su jutikliais yra sekami kroviniai. Tokių infrastruktūrų pažeidimas gali atnešti ne tik didelių finansinių problemų kompanijoms, bet ir didelių nuostolių vartotojams.

Dabar daug investuojama į daiktų interneto saugumą. Yra kuriami nauji įrankiai įsilaužimui aptikti. Tačiau, kaip pastebėta darbe „Kibernetinio saugumo kūrimas integruotose daiktų interneto įrenginiuose“ (Usmonov et al., 2017), dabar yra sukurta labai daug skirtingų įrenginių, kuriems reikia atskirai kurti saugos mechanizmus. Vienas iš perspektyviausių įrankių yra medaus puodynės koncepcija. Ji suteikia galimybę ne tik aptikti ir sustabdyti įsilaužėlį, bet ir pateikia ataskaitą, kaip tai buvo padaryta. Tai ypač svarbu, siekiant greitai įgyvendinti naujus saugos mechanizmus greitai besivystančioje daiktų interneto rinkoje.

Tyrimo objektas

Tyrimo objektas – „medaus puodynės“ koncepcijos pritaikymas daiktų internete.

Darbo tikslas ir uždaviniai

Pagrindinis darbo tikslas yra pritaikyti „medaus puodynės“ koncepciją komerciniams daiktų ir paslaugų interneto įrenginiams.

Pagrindiniai uždaviniai:

- Išanalizuoti „medaus puodynės“ veikimo principus.
- Pasiūlyti „medaus puodynės“ koncepciją, skirtą daiktų ir paslaugų internetui.
- Atlikti „medaus puodynės“ koncepcijos realizaciją ir testavimą.

Temos naujumas

Pastarąjį dešimtmetį pradėjo sparčiai vystytis daiktų internetas. Tai lėmė du faktoriai: pingantys elektronikos įrenginiai ir komunikacijos technologijų tobulėjimas. Šiandien mes turime labai sparčiai auganti daiktų internetą. Sensoriniai kompiuteriniai įrenginiai su prieiga prie interneto supa mus. Jie yra naudojami namuose, viešajame transporte, gamyklose, net valdant miestą yra naudojami įvairūs sensoriniai įrenginiai. Tai suteikia patogumo ir supaprastina mūsų gyvenimą. Įvairūs įrenginiai ar net gamyklos gali būti valdomos iš bet kurio pasaulio taško, kur yra internetas. Tačiau su inovacijomis ateina ir grėsmė. Šiandienai dar nėra sukurta pakankamai saugos mechanizmų daiktų interneto įrengiamas apsaugoti. Kol kas egzistuoja tik mėgėjiškos ir universitetuose tyrimams sukurtos „medaus puodynės“, tačiau jų funkcionalumo nepakanka apsaugoti daiktų interneto įrenginius.

Temos aktualumas

Daiktų internetas yra sparčiai besivystanti sritis, kuri prijungia vis daugiau ir daugiau įrenginių prie interneto. Ne visi įrenginiai yra suprojektuoti prisijungimui prie interneto, todėl reikalingi papildomi saugos mechanizmai, kad galima būtų juos apsaugoti nuo kibernetinių atakų. Atsižvelgiant į tai, kad vis daugiau pramonės įmonių pradeda naudoti veiklos/gamybos procesuose išmaniuosius sensorius, sujungiant

juos į kompiuterinius tinklus, kyla didelis pavojus, kad į juos bus įsilaužta. Užvaldžius tokią sistemą, atsiranda galimybė neigiamai daryti įtaką vidiniams procesams, galima sukelti fizinį pavojų dirbantiems asmenims ir vartotojams. Vienas iš žinomų atvejų buvo „Stuxnet“ kenksmingas programinis kodas, kuris užvaldė „Siemens“ valdiklius, valdančius urano gamyklą, ir pakeitė jų konfigūraciją, tokiu būdu sugadindamas įrangą. Aukų nebuvo, tačiau buvo padaryta milžiniška turtinė žala ir, remiantis ekspertų nuomone, sustabdė Irano branduolinę programą 15-ai metų. Daiktų interneto įrenginių saugumo užtikrinimas išlieka aktuali problema. Užvaldžius tokį įrenginį galima ne tik pavogti asmeninę informaciją ir padaryti turtinę žalą, bet ir sukelti pramoninę katastrofą, kuri gali nusinešti ne vieną gyvybę, ir padaryti esminę žalą aplinkai ir gamtai.

Tyrimo metodai

Atlikti esamų „medaus puodynių“, pritaikytų daiktų ir paslaugų internetui, analizę naudojant lyginamosios, vertinimo ir gretinimo analizės metodus, bibliotekinio tyrimo metodus. Pagrindinėms išvadoms išvesti naudojamas loginės indukcijos metodas.

Mokslinė darbo vertė

Šio mokslinio darbo metu bus ištirtos daiktų interneto įrenginio pažeidžiamumas, atlikta jo analizė ir sukurtas įrankis, gebantis stebėti ir padėti išvengti šių pažeidžiamumo atsiradimo.

Darbo struktūra

Magistro baigiamasis darbas yra sudarytas iš 5 skyrių:

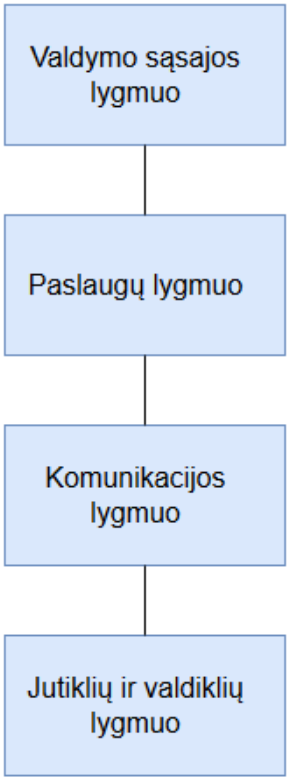
1. Įvadas – šiame skyriuje aprašomas darbo aktualumas ir naujumas.
2. Literatūros apžvalga – šiame skyriuje atliekama su magistro darbo tema susijusios literatūros analizė, apžvelgiami esamų metodų trūkumai.
3. Siūlomo metodo aprašymas – šiame skyriuje aprašomas siūlomas metodas.
4. Siūlomo metodo realizacija ir testavimas – šiame skyriuje aprašomas metodo įgyvendinimas ir testavimo rezultatai.
5. Apibendrinimas ir išvados – šiame skyriuje atliekama magistro darbo analizė, padaromos išvados.

2 Susijusios literatūros analizė

2.1 Daiktų ir paslaugų interneto sąvoka ir veikimo principas

Remiantis IEEE organizacijos apibrėžimu (Minerva et al. 2015), daiktų internetas gali būti apibūdinamas kaip kompiuterinis tinklas, apjungiantis unikalius identifikatorius turinčius elektroninius įrenginius, gebančius perduoti duomenys per kompiuterinį tinklą kitiems įrenginiams, gauti duomenys iš kitų įrenginių per kompiuterinį tinklą ir gautų duomenų pagrindu atlikti reikiamus veiksmus be žmogaus įsikišimo. Kaip suprantame iš aprašymo, daiktų internetas susideda iš tarpusavyje kompiuteriniu tinklu sujungtų išmaniųjų įrenginių, gebančių sąveikauti tarpusavyje be tarpininko. Daiktų interneto įrenginiai neturi vienodos architektūros. Pagrindinis reikalavimas yra gebėjimas sąveikauti tarpusavyje. Daiktų interneto įrenginiu gali būti tiek vaizdo kamera, kuri pastebėjusi judėjimą išsiunčia pranešimą vartotojui, tiek paprastas naminis maršruto parinktuvas (Sharma, Shamkuwar and Sing, 2018).

Remiantis Kinijos Foshan universitete atlikta analize (Zhong, Zhu and Huang, 2016) daiktų interneto architektūra yra suskirstoma į 4 grupes, kaip parodyta 2.1 pav. jutiklių ir valdiklių lygmuo, komunikacijos lygmuo, paslaugų lygmuo ir valdymo sąsajos lygmuo. Jutiklių ir valdiklių lygmenyje yra įvairus jutikliai ir valdikliai. Šiuo metu įvairus „protingi“ jutikliai sugeba rinkti informaciją ir keistis ja su kitais įrenginiais.



2.1 pav. „Daiktų interneto“ modelio schema

Komunikacijos lygmenyje yra perduodama informaciją iš jutiklio į serverį, kuriame ji yra kaupiama. Daiktų interneto įrenginiai gali naudoti visus komunikavimo protokolus ir technologijas. Buitiniai jutikliai ir išmanieji daiktai paprastai naudoja 3G, 4G ir bevielius tinklus, kadangi jų vartotojai savo namuose turi prieigą prie globalinio tinklo per juos, ir tai yra patogiausias būdas daiktams ir jutikliams pasiekti serverį ir vartotoją. Pramoninėje aplinkoje jutikliai duomenų siuntimui naudoja WPAN tinklus.

Paslaugų lygmuo suteikia funkcionalumą daiktų internetui. Šiame lygmenyje sujungiamos programos ir jutikliai . Visi surinkti duomenys yra apdorojami ir pritaikomi kliento poreikiams. Tokiu būdu vieno jutiklio duomenys gali dalyvauti daugybėje programų , kas sumažina infrastruktūros kainą, nes nereikia kiekvienai programai įdiegti atskiro jutiklio ir supaprastina infrastruktūros architektūrą. Šiame lygmenyje galima surasti sekančius komponentus:

1) Paslaugos aptikimas –surandamas jau egzistuojantis jutiklis , kuris gali suteikti vartotojui reikalingą informaciją.
2) Paslaugos struktūra – paieškos metu yra surandami ir apjungiami jutikliai , tam kad išgauti kuo daugiau vartotojui naudingos informacijos.
3) Pasitikėjimo valdymas – nustatomi patikimi jutikliai , kurių informacija yra patikima. .
4) Paslaugos API – API padeda lengviau integruoti egzistuojančias paslaugas į programinę įrangą. .

Valdymo sąsajos lygmuo yra skirtas patogiam „daiktų“ infrastruktūros valdymui. Vienas iš populiariausių protokolų „daiktų“ valdymui yra „Universal Plug and Play“, kuris suteikia galimybę valdyti jutiklius bei pasiekti įvairias paslaugas.

2.2 Daiktų ir paslaugų interneto įrenginių pažeidžiamumas

Viena iš didžiausia IT saugumo problemų šiai dienai yra daiktų internetas. Ši sritis sparčiai vystosi ir atsiduria padidintoje saugumo rizikos zonoje. Pagrindinės priežastys, kodėl šios srities saugumui turi būti skiriama ypatingas dėmesys yra:

- Daiktų interneto įrenginiai yra prijungiami prie interneto. Tokių būdų juos gali pasiekti ne tik vartotojai, bet ir „hakeriai“.

- Daiktų interneto įrenginiai būna prijungti prie interneto 24 valandas per para ir 7 dienas per savaitę. Tokių būdu prie šių įrenginių galima jungti bet kuriuo paros metu.

- Daiktų interneto įrenginiai ne visada randasi tame pačiame tinkle su vartotojais. Vartotojas gali prie įrenginio jungti iš skirtingų tinklų su skirtingais IP adresais, kas apsunkina galimybę atskirti vartotojo prisijungimą nuo įsilaužėlio prisijungimo.

- Daiktų internetas susideda iš skirtingų įrenginių, kurie naudoja skirtingus komunikavimo protokolus, skirtingas operacines sistemas ir platformas, ir kiekvienas komunikavimo protokolas, operacinė sistema ir platforma turi savo silpnąsias vietas. Paveikus vieną įrenginį, galima paveikti visą tinklą.
- Daiktų interneto įrenginiai reikalauja apsaugos ne tik iš programinio kodo pusės, bet ir fizinės pusės.

Apžvelgsime dažniausiai įsilaužėlių naudojamus daiktų interneto silpnąsias puses. Bangaloro technikos institutas atliko analizę (Mendez, Papapanagiotou and Yang, 2017) ir išsiaiškino, kokios yra galimos kibernetinės atakos prieš daiktų interneto įrenginius.

Vartotojo sąsaja yra viena iš pažeidžiamiausių daiktų interneto infrastruktūros dalis. Tai yra todėl, kad vartotojo sąsaja yra dažniausiai talpinama žiniatinklyje, kas leidžia prie jos netrukdomai prieiti bet kam. Vartotojai dažnai naudoja silpnus ir lengvai atspėjamus slaptažodžius. Atakuojantis asmuo, turintis laisvą priėjimą prie vartotojo sąsajos, gali neribotai daug kartų bandyti atspėti slaptažodį arba atlikti SQL injekciją ir nutekinti visų vartotojų slaptažodžius.

Tinklas irgi yra pažeidžiama daiktų interneto dalis. J. Ahamed ir A. V. Rajan savo daiktų interneto pažeidžiamumo apžvalgoje (Ahamed & Rajan, 2016) nurodo, kad didžiausias tinklo ir komunikacijos trūkumas daiktų interneto įrenginiuose yra šifravimas. Komunikacijos kanalai yra nešifruojami ir visi vartotojo slaptažodžiai ir duomenys yra neapsaugoti. Nusikaltėliai gali pasinaudoti šiuo trūkumu ir atlikti pasyvią ataką. Stebėti viską, kas yra išsiunčiama ir atsiunčiama į įrenginį ir taip gauti jiems reikalingą informaciją.

Neatnaujinta programinė įranga irgi gali tapti įsilaužymo priežastimi. Dauguma „skylių“ programinėje įrangoje yra surandamos saugos specialistų ir pašalinamos atnaujinant programinę įrangą. Tačiau vartotojai ne visada laiku atsinaujina įrenginio programinę įrangą, įsidiegia jos paskutinę versiją. Tai suteikia „hakeriui“ galimybę lengvai užvaldyti įrenginius. Jis patikrina kokia programinės įrangos versija yra įdegta ir suranda internete saugos specialistų ataskaitose paminėtas šios programinės įrangos saugumo spragas. Toliau atakuojantis asmuo vykdo ataką.

Arizonos universitete buvo atlikta daiktų interneto pažeidžiamumo analizė (Williams et al. 2017), kurios metu buvo tikrinami išmanūs televizoriai, spausdintuvai, IP vaizdo stebėjimo kameros. Iš

156 680 testuotų įrenginių 20 237 įrenginiai arba 12% buvo pripažinti kaip pažeidžiami. Tarp IP vaizdo stebėjimo kamerų dažniausias aptinkama spraga buvo „MiniUPnP“ tinklo aptikimo ir komunikacijos protokolų sena versija. Ši saugumo spraga atakuojančiam asmeniui duotų galimybę prijungti prie „botnet“ tinklo ir atlikti „DDoS“ ataką. Didelė saugumo spraga buvo NAT-PMP protokolas. Jis leidžia įsilaužėliui gauti daugiau informacijos apie tinklą ir įsilaužti į jį. Vidutinio lygio pažeidžiamumas buvo rastas IP vaizdo stebėjimo kamerose, kur naudojamas nešifruotą telnet protokolą naudojantis serveris. Tokių būdų atakuojantis asmuo gali atlikti tarpininko (angl. man-in-the-middle) ataką ir pavogti vartotojo prisijungimo duomenis.

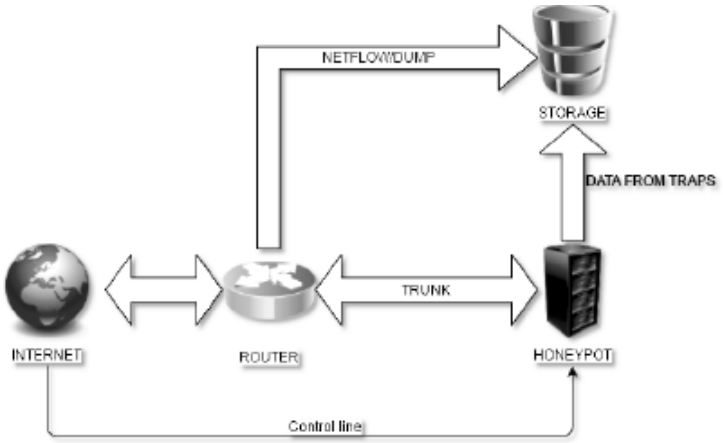
Išmaniuosiuose televizoriuose aptiktas kritinis pažeidžiamumas - SNMP protokolas. Šis protokolas leidžia serveriui rinkti informaciją apie įrenginius tinkle ir, modifikuojant šią informaciją, pakeisti įrenginių būseną, valdyti juos. Panaudojus šį protokolą, atakuojantis asmuo gali lengvai perimti įrenginio valdymą. Tokią pačią saugumo spragą turėjo prijungti prie kompiuterinio tinklo spausdintuvai. Juos irgi galima užvaldyti naudojant SNMP protokolą.

SCADA sistemos irgi yra pažeidžiamos. Tai yra gerai aprašyta Arizonos universitete atliktoje analizėje (Samtani et al., 2016), kurios metu buvo ištirti 587 158 SCADA sistemai priklausančių įrenginių. Iš 587 158 įrenginių 37 854 arba 6% turėjo saugumo spragas. Daugiausia kritinių spragų buvo susiję su buferio perpildymu, kuris pažeidžia valdiklį ir infrastruktūrą. Remiantis tuo pačiu dokumentu, atakuojantis ne visada galės pasinaudoti valdiklių pažeidžiamumu, nes tai reikalauja specifinių žinių. Dažniausiai atakuojantys asmenys analizuoja siunčiamus duomenis ir perima vartotojų prisijungimo duomenis, nes duomenys, kuriuos siunčia valdiklis serveriui arba serveris valdikliui, nėra šifruojami. Taip pattyrimo metu buvo aptikti keli valdikliai, kurie leisdavo naudotojui juo stabdyti arba paleisti be prisijungimo.

2.3 „Medau puodynės“ koncepcija daiktų internete

„Medaus puodynė“ - tai įrankis, gebantis aptikti įsilaužėlį ir surinkti informaciją apie tai, kaip tai buvo padaryta. Šiuolaikiniame besivystančių technologijų pasaulyje šis įrankis tampa itin svarbus. Jis padeda ne tik apsaugoti skaitmeninę infrastruktūrą nuo įsilaižymo, bet ir užfiksuoti įsilaužymo eigą. Tokiu būdu įgalinama įsilaužymo analizė ir „spragų“ taisymas. Panaudojant „medaus puodynės“ galimybes galima padaryti kompiuterinę infrastruktūrą saugesne. Daiktų internetui saugumo lygio padidinimas yra aktualus. Dėl mažo galingumo išmanūs įrenginiai ne visada gali turėti savyje antivirusines programas ir analizuoti įeinantį tinklo srautą. Šiuo atveju „medaus puodynės“ koncepcija puikiai tiktų sprendžiant saugumo padidinimo klausimus. „Medaus puodynė“ tampa masalu tinkle, kuris patraukia įsilaužėlių dėmesį ir atitinkamai nukreipia jį nuo pagrindinės sistemos.

Straipsnyje „Medaus puodynės“ kūrimas ir diegimas“ (Egupov, Zareshin, Yadikin and Silnov, 2017) yra paaiškinamas jos veikimo pricipas. Į kompiuterinį tinklą yra įdiegiamas kompiuteris. Jis yra izoliuojamas nuo kitų kompiuterių, esančių tinkle, ir jo saugumo standartai yra sumažinami. Tokie kompiuteriai tinkle tampa patrauklūs atakuojančiam asmeniu, nes į jį galima lengvai įsilaužyti. Tačiau įsilaužęs asmuo iš šio kompiuterio niekur daugiau negali patekti, nes kompiuteris yra izoliuotas nuo kitų. Kiekvienas prisijungimas prie šio kompiuterio yra traktuojamas kaip įsilaužymas ir IP adresas, iš kurio buvo įsilaužta, iškart blokuojamas. Ši schema yra parodyta 2.2 pav.



2.2 pav. „Medaus puodynės“ veikimo principas

„Medaus puodynės“ yra naudojamos dviem tikslams: tyrimams ir IT infrastruktūrų saugumui. Moksliniuose tyrimuose „medaus puodynės“ yra naudojamos tam, kad surinkti kuo daugiau informacijos apie įsilaužėlių naudojamus įsilaužymo metodus ir įrankius. Šio tipo „medaus puodynės“ nesuteikia didelio apsaugos lygio, nes pagrindinis jų tikslas yra būti masalu ir privilioti kuo daugiau „hakerių“, kad galima būtų surinkti kuo daugiau informacijos apie juos ir jų įvykdytus įsilaužimus.

IT infrastruktūrai apsaugoti naudojamos „medaus puodynės“ turi mažiau funkcionalumo nei tyrimams skirtos „medaus puodynės“, tačiau su mažesniu funkcionalumu atsiranda mažiau pažeidžiamumo. Šio tipo „medaus puodynės“ suteikia mažiau duomenų apie įsilaužėlį ir įsilaužymo metodus ir įrankius, tačiau tokio tipo „medaus puodynėse“ yra naudojama daugiau saugos mechanizmų. „Medaus puodynės“ minimaliai sąveikauja su įsilaužėliais ir praneša kompiuterinio tinklo administratoriui apie prisijungimą prie įrenginio, tuo pačiu metu užlaikydama įsilaužėlį imituojant pasiustų veiksmų apdorojimą ir duoda laiko saugos specialistams sureaguoti.

„Medaus puodynės“ koncepcija daiktų interneto įrenginiuose veikia panašiu principu. Šis principas yra gerai aprašytas darbe „ZigBee Honeypot įvertinimas daiktų interneto atakoms tirti“ (Dowling, Schukat and Melvin, 2017). Į įrenginį yra įdiegiamas „medaus puodynė“. Ši programa pradeda stebėti visus portus ir prisijungimus. Esant nesankcionuotam prisijungimui, programa paleidžia emuliatorių, netikrą

komandinę eilutę. Atakuojantis mano, kad jis prisijungė prie valdiklio ir paleidžia savo kenksmingą programinį kodą. „Medaus puodynė“ įrašo visus atakuojančio veiksmus ir siunčia įrašą kompiuterinio tinklo administratoriui. Tokių būdų yra ne tik pagaunamas atakuojantis asmuo, bet ir aptinkama silpna infrastruktūros vieta.

Remiantis I. Mokube ir M. Adamso darbu „Medaus puodynės: koncepcija, požiūriai ir iššūkiai“ (Mokube and Adams, 2007). Medaus puodynės turi trys sąveikos lygius: žemas sąveikos lygis, vidutinis sąveikos lygis ir aukštas sąveikos lygis.

Žemo sąveikos lygio medau puodynės gali imituoti tik programas . Šios medaus puodynės geba kopijuoti esamą IT infrastruktūrą ir sukurti netikrus kompiuterius su nenaudojamais IP adresais. Šis metodas yra saugiausias, nes naudojami nesudėtingi mechanizmai, o prieigos prie operacinės sistemos atakuojantis neturi. Kadangi atakuojančiam yra pasiūlomas ribotas veiksmų pasirinkimas, atakų žurnalizavimas irgi yra limituotas.

Vidutinio sąveikos lygio „medaus puodynės“ neimituoja operacinės sistemos, tačiau gali imituoti sudėtingesnes programas. Nors tikimybė surasti pažeidimą „medaus puodynėje“ išauga, tačiau atakuojančiam asmeniui sudaroma tikroviškesnė iliuzija, kad jis atakuoja tikrą sistemą. Kadangi programos yra kompleksiškesnės, vykdomas detalesnis atakų žurnalizavimas, kas gali padėti geriau suprasti, kaip buvo atakuojama ir kokiomis spragomis buvo pasinaudota.

Aukšto sąveikos lygio „medaus puodynės“ yra kompleksiškiausias „medaus puodynės“ tipas. Šio tipo „medaus puodynės“ naudoja įrenginio operacinę sistemą ir viskas yra leidžiama. Yra surenkama daugiausia informacijos apie ataką ir atakuojantį, nes jis niekaip nėra stabdomas arba ribojamas. Tai yra pavojingiausias metodas, nes neteisingai sukonfigūravus įrenginį arba prieiga prie įrenginio, atakuojantis gali be kliūčių patekti į IT infrastruktūrą ir kitus įrenginius esančius infrastruktūroje.

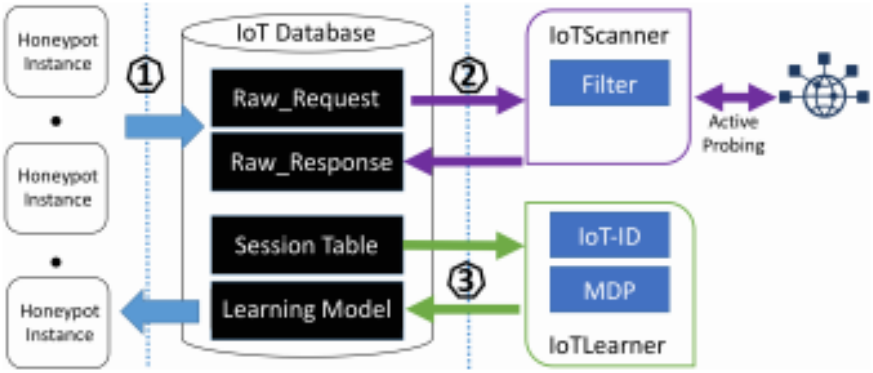
Kol kas rinkoje neegzistuoja „medaus puodynės“ programos skirtos daiktų internetui. Tačiau galima rasti mėgėjiškų projektų, kurie pradeda pritaikyti šią koncepciją daiktu interneto įrenginiams. Vienas iš tokių yra „Telnet IoT HoneyPot“ žemo sąveikos lygio „medaus puodynė“, paremta kliento/serverio komunikacija. Ši „medaus puodynės“ programinė įranga buvo sukurta gaudyti „botnet“ tipo kenkėjiška kodą, kuris šiuo metu infekuoja daiktų interneto įrenginius. Ši programinė įranga simuliuoja nesaugų telnet protokolą su populiariausiais prisijungimo slaptažodžiais, tačiau nesimuliuoja daiktų interneto įrenginio, nes buvo sukurta su tikslu analizuoti veikiančias „botnet“ atakas. Ši „medaus puodynė“ gali simuliuoti visus telnet sesijos tipus, kurie bandys užmėgsti ryšį su juo ir bandys jį užkrėsti. Klientas sąveikauja su įsilaužėliu ir priima visas jo komandas ir failus, kuriuos jis įkelią į įrenginį, ir siunčia serveriui. Serveris priima visas komandas, kenksmingą programinį kodą ir analizuoja jį. Tokio tipo sąveika apsaugo infrastruktūrą nuo „medaus puodynės“ pažeidimo ir vartotojo teisių eskalavimo, nes klientas atlieka fasado vaidmenį ir nėra

sujungtas su infrastruktūra, o serveris visus ateinančius duomenys traktuoja kaip pavojingus ir analizuoja juos „smėlio dėžėje“.

„IoTPOT“ yra vidutinės sąveikos lygio „medaus puodynė“ (Pa et al. 2016), kuri gali simuliuoti programas naudojančias telnet komunikacijos protokolą įvairiuose įrenginiuose su įvairiais CPĮ architektūrą. Ši „medaus puodynė“ irgi buvo sukurta vykdyti tyrimams. „IoTPOT“ susideda iš dviejų dalių. Pirma dalis yra išorinė sąsaja, su kuria sąveikauja „hakeris“. Prisijungus prie jos, įsilaužėlis mato įprastą telenet komandinę eilutę, gali peržiūrėti vykstančius procesus, kurie irgi yra simuliuojami, gali panaudoti komandas arba įkelti virusą. Visa informacija ir kenksmingas programinis kodas yra siunčiami į antrą, vidinę dalį, dar vadinama „IoTBOX“. Šioje dalyje yra simuliuojami įvairus daiktų interneto įrenginiai su įvairiais CPĮ ir stebima, kaip atsiųstas kenksmingas programinis kodas paveiks sistemą. Ši „medaus puodynė“ gali paveldėti įvairius nustatymus iš telnet protokolo, tokius, kaip atakuojančio kliento nustatymai, realistiškas pasveikinimo žinutes. Taip pat, gali leisti prisijungti be iš anksto nustatyto slaptažodžio ir vartotojo vardo, tokiu būdu suteikiant šansą išanalizuoti, kokius vartotojų vardus ir slaptažodžius naudoja įsilaužėliai. Tačiau svarbiausiais privalumas yra tai, kad „IoTPOT“ palaiko įvairias CPĮ architektūras, kas leidžia aptikti įvairaus tipo kenksmingą programinį kodą. Autentifikavimo metu ši „medaus puodynė“ gali: atmesti bet kokius prisijungimus, tam kad galima būtų stebėti kokius metodus naudoja įsilaužymui, leisti visiems prisijungti, tam kad galima būtų gauti kenksmingo programinio kodo pavyzdžių ir išanalizuoti, kokius veiksmus atlieka įsilaužėliai po įsilaužymo arba prisijungimą tik su tam tikrais duomenimis.

Ši „medaus puodynė“ palaiko kelias „Linux“ operacines sistemas, skirtas įterptinoms elektroninėms sistemoms. Ir kai „hakeris“ paleidžia komandą, ji yra simuliuojama „IoTBOX“ terpėje ant kelių skirtingų sistemų su skirtinga CPĮ architektūra. Po komandos apdorojimo, rezultatas yra siunčiamas atgal į „IoTPOT“ ir atvaizduojamas įsilaužėliui. Tokiu būdų galima ištirti įrenginius su įvairia architektūra ir sudaryti iliuziją, kad įsilaužėlis dirba su tikru daiktų interneto įrenginiu.

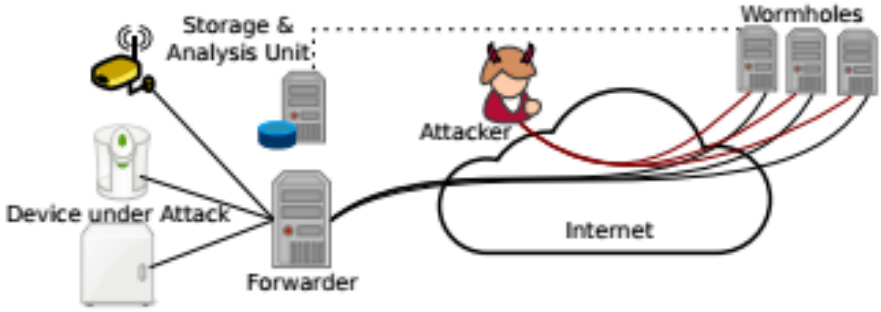
Viena iš pažangiausių sukurtų “medaus puodynių” šiai dienai yra “IoTCandyJar” (Luo et al. 2017). Ši “medaus puodynė” buvo sukurta remiantis idėja, kad, dėl didelio skaičiaus įrenginių daiktų internete, neįmanoma sukurti kiekvienam įrenginio tipui aukšto sąveikos lygio “medaus puodynės”, o žemo sąveikos lygio “medaus puodynės” nesuteikia reikiamos duomenų kokybės ir geba tik analizuoti atskirus atakų tipus. Buvo sukurta intelektualios sąveikos “medaus puodynė”, kuri geba mokytis ir simuliuoti įvairiausius įrenginius. Pagrindinis šios „medaus puodynės“ tikslas yra išmokti teisingai sąveikauti su atakuojančiu asmeniu ir išprovokuoti jį įvykdyti realią ataką. “IoTCandyJar” veikimas yra parodytas 2.3 pav.



2.3 pav. “IoTCandyJar” veikimo principas

Visa informacija, kuri yra gaunama sąveikaujant su atakuojančiu asmeniu, yra saugoma “IoT Database” duomenų bazėje. Šioje duomenų bazėje saugomi atsakymai į tam tikras įsilaužėlio užklausas. Jeigu ateina nauja užklausa, kurios nėra duomenų bazėje, „medaus puodynė“ išsaugo ją, kaip naują užklausą “Raw_Request” lentelėje ir ieško internete, koks įrenginys naudoja tokias užklausas ir kokius atsakymus jis turi siųsti. Visi surasti atsakymai yra išsaugomi “Raw_Response” lentelėje ir vienas atsakymas yra atvaizduojamas atakuojančiam. Sesijos pabaigoje, panaudojant kompiuterio mokymosi algoritmus, yra daroma analizė, kokios užklausos ir atsakymai buvo sėkmingiausi ir jos rezultatai yra išsaugomi. Tokiu būdu pastoviai yra kuriamos naujų įrenginių simuliacijos ir sistema nuolat mokosi kaip pergudrauti “hakerį” ir paskatinti jį , įvykdyti ataką.

Iki šiol buvo aprašytos „medaus puodynės“, kurios simuliavo įrenginius. “SIPHON” yra „medaus puodynė“ (Guarnizo et al. 2017), kuri naudoja realius daiktų interneto įrenginius. Veikimo principas parodytas 2.4 pav.



2.4 pav. “SIPHON” veikimo principas

Šioje „medaus puodynėje“ yra naudojami tikri įrenginiai. Jų neprijungė tiesai prie interneto, bet panaudojo „wormholes“ tunelį, kad galima būtų įrašinėti visus atakuojančio asmens veiksmus. Visi duomenys iš

„wormholes“ yra siunčiami panaudojus telnet protokolą, tokiu būdų yra sudaroma iliuzija, kad atakuojantis tiesiogiai kontaktuoja su daiktų interneto įrenginiu. „Wormholes“ yra reikalingi interneto srauto fiksavimui ir analizei. „Medaus puodynė“ buvo įdiegta į įrenginius ir sekė, kokius veiksmus atlieka atakuojantis asmuo.

Kaip rašoma „Atakų ICS tinkle sekimas remiantis „Honeypot“ (Abe, Tanaka, Uchida and Horata, 2017) „medaus puodynė“ negali suteikti visiškos apsaugos nuo kenksmino programinio kodo, tačiau ji suteikia galimybę aptikti kenksmingus procesus kompiuteryje ir juos sustabdyti. Ši koncepcija yra geriausias variantas daiktų internetui, nes nereikalauja daug resursų veikimui. Tai yra ypač svarbu jutikliuose, kur energijos resursai yra riboti.

2.3 Antro skyriaus apibendrinimas ir pagrindiniai rezultatai

Šiame skyriuje buvo išanalizuotas daiktų ir paslaugų interneto veikimas, pagrindinės atakos prieš daiktų ir paslaugų internetą ir „medaus puodynių“ pritaikymas identifikuojant grėsmes ir apsaugant daiktų ir paslaugų interneto įrenginius. Pagrindiniai šio skyriaus rezultatas yra atlikta analizė. Buvo suprasti „medaus puodynių“ daiktų ir paslaugų internete veikimo principai ir pagrindiniai jų trūkumai.

2.4 Išvados

Atlikus literatūros analizę, galima padaryti keletas išvadų. Daiktų ir paslaugų internetas reikalauja daug dėmesio saugos srityje. Tokie įrenginiai yra nuolat prijungti prie interneto ir dažniausiai juos gali pasiekti bet kuris interneto vartotojas. Atliktos analizės parodo, kad galima nesunkiai užvaldyti įrenginius. Turi būti sukurti mechanizmai, kurie padės nustatyti pažeidžiamas vietas įrenginiuose. „Medaus puodynės“ puikiai susidorojo su šia užduotimi, tačiau ši koncepcija yra naudojama tik daiktų ir paslaugų interneto tyrimams. Dėl daiktų ir paslaugų interneto įrenginių įvairoves yra labai sudėtinga sukurti universalią „medaus puodynę“ tinkančia visiems įrenginiams, tačiau šia linkme judama. Egzistuojančios „medaus puodynės“ koncepcijos turi būti patobulintos, kad jas galima būtų pritaikyti daiktų ir paslaugų interneto apsaugai nuo įsilaužymo. Turi būti sukurtas pilnavertis tokio tipo saugos mechanizmo sprendimas.

3 „Medaus puodynės“ koncepcijos pritaikymas daiktų ir paslaugų internetui

3.1 Siūlomas metodas

Šio magistrinio darbo tikslas yra sukurti „medaus puodynę“ pritaikyta komerciniams daiktų interneto įrenginiams. Tyrinėjant literatūrą apie „medaus puodynes“ ir šios koncepcijos pritaikymą daiktų ir paslaugų internete, buvo stengiamasi suformuoti metodą, kuris leistų pilnai imituoti daiktų ir paslaugų interneto įrenginį. Siūlomas metodas daiktų interneto įrenginio „medaus puodynės“ sukūrimui gali būti padalintas į dvi dalys:

- Virtualaus įrenginio sukūrimas
- Virtualaus įrenginio paleidimas

Siūlomas metodas susideda iš 6 žingsnių, kurie yra pavaizduoti 3.1 pav. Realaus įrenginio programinės aparatinės įrangos gavimo žingsnyje yra pasirenkamas daiktų ir paslaugų interneto įrenginys, kuris bus imituojamas ir gaunama jo programinė aparatinė įranga (angl. firmware) binarinio failo pavidalu. Šiame faile yra užkoduota failinė sistema ir programos.

Kadangi norima simuliuoti pilną įrenginio veikimą, programinės įrangos dekodavimo žingsnyje, iš įrenginio binarinio failo yra ištraukiama įrenginio failinė sistema. Tai padės suprasti daiktų ir paslaugų interneto įrenginio programinės įrangos veikimo principą ir kuo tiksliau imituoti įrenginio veikimą virtualioje aplinkoje.

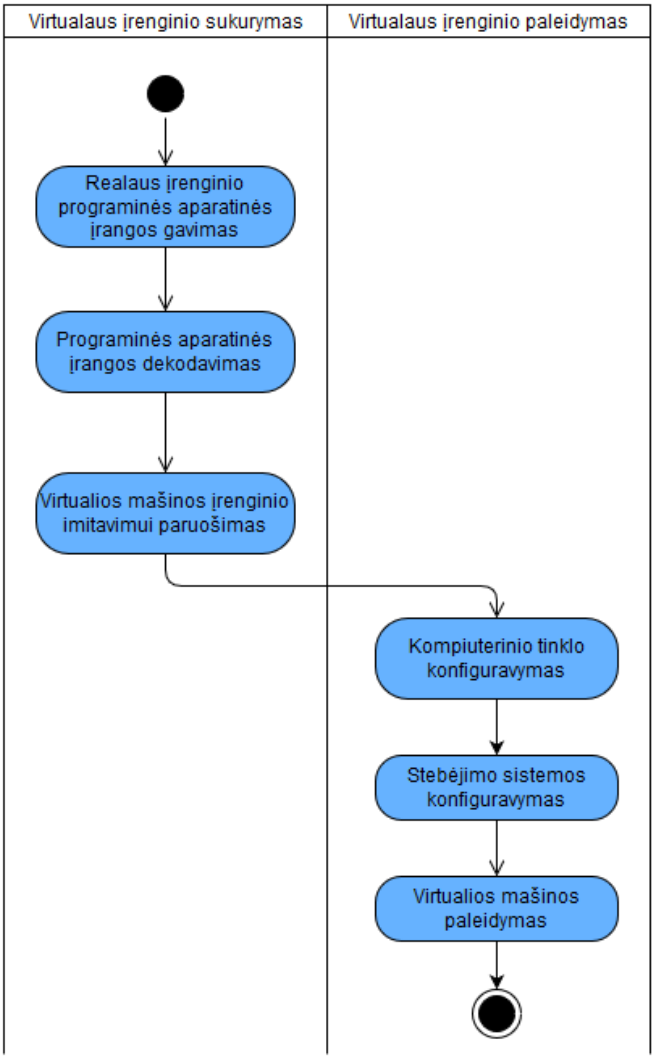
Virtualios mašinos daiktų ir paslaugų interneto įrenginio imitavimui paruošimo žingsnyje yra atliekama virtualios mašinos paruošimas. Šiame žingsnyje ištraukta failinė sistema, turi būti paruošta virtualizacijai. Taip pat yra sukuriamas tinkamas operacinės sistemos branduolys.

Kompiuterinio tinklo konfigūravimo žingsnyje yra atliekama lokalaus kompiuterinio tinklo ir tinklo įrenginių konfigūravimas, nes norima, kad sukurta „medaus puodynė“ būtų pasiekiama iš lokalaus tinklo. Šiame žingsnyje yra sukuriami tinklo įrenginiai, kurie leidžia imituoti atskirą lokalų tinklą virtualioje mašinoje ir prie jo yra prijungiama sukurta „medaus puodynė“.

Tinklo srauto stebėjimo sistemos konfigūravimo žingsnyje prie sukurtų tinklo įrenginių yra prijungiamos tinklo srauto stebėjimo sistemos. Kadangi norima žinoti, kada yra prisijungema prie „medaus puodynės“ ir iš kokio IP adreso, atitinkamai yra sukonfigūruojama tinklo srauto stebėjimo sistema ir paskutiniame, šeštame žingsnyje yra paleidžiama sukurta virtuali mašina.

Daiktų ir paslaugų interneto įrenginys bus imituojamas virtualioje aplinkoje. Tai padės efektyviau stebėti įrenginio būseną panaudojant įrankius, kuriuos būtų sunku įdiegti į mikroprocesorinį

įrenginį. Daugumą daiktų ir paslaugų interneto įrenginių veikia ant MIPS architektūros mikroprocesorių. Aparatinės įrangos virtualizacijos įrankis turi gebėti virtualizuoti būtent šio tipo mikroprocesorių aparatinę dalį.



3.1 Siūlomo metodo veiklos diagrama.

3.2 Naudojamų įrankių pasirinkimas ir analizė

Šiame darbe, renkantis įrankius, prioritetas buvo suteikiamas atviro kodo programiniai įrangai. Atliekant įvairios programinės įrangos analizę buvo atsižvelgta į kainą ir prioritetas buvo suteiktas nemokomiems atviro kodo įrankiams.

3.2.1 Operacinės sistemos pasirinkimas

Renkantis operacinę sistemą, kurioje bus paleidžiama „medaus puodynė“ buvo taikomi šie kriterijai:

- Darbo patirtis
- Konfigūravymo patogumas
- Kaina

3.1 lentelė. *Operacinių sistemų palyginimas.*

	Patirtis	Patogumas	Kaina
Windows	-	-	Mokoma
Linux	+	+	Nemokoma

Išanalizavus turimus variantus, buvo nuspręsta pasirinkti “Linux” operacinė sistema kaip operacinę sistemą, kurioje bus kuriama ir testuojama daiktų ir paslaugų interneto “medaus puodynė”. Tai lėmė keli faktoriai: daugiau patirties darbe su “Linux” operacinėmis sistemomis, nei su “Windows” operacine sistema. “Linux” operacinės sistemos yra lengviau konfigūruojamos. Didžiausią svorį turėjo kainosfaktorius kuris nulėmė “Linux” operacinės sistemos pasirinkimą.. Priešingai negu “Windows” operacinės sistemos, “Linux” operacinės sistemos yra atviro kodo ir nemokomos.

3.2.2 Virtualizacijos platformos pasirinkimas

Renkantis virtualizacijos platforma buvo keliami sekantis reikalavimai:

- Galimybė imituoti mikroprocesorinį įrenginį;
- Galimybė paleisti MIPS architektūros „Linux“ operacinę sistemą ir programas;
- Atskirai paleisti „Linux“ operacinės sistemos branduolį ir failinę sistemą.

3.2 lentelė. *Virtualizacijos platformų palyginimas.*

	MIPS architektūros palaikymas	Atskirai nurodyti branduolį ir failinę sistemą	Kaina
VMWare	-	+	Nemokama
VirtualBox	-	+	Nemokama
Qemu	+	+	Nemokama

Išanalizavus skirtingas virtualizavimo platformas, buvo pasirinkta naudoti „Qemu“ virtualizavimo platformą. Pagrindė priežastis, dėl kurios buvo priimtas toks sprendimas, yra galimybė imituoti MIPS architektūrą. Daiktų interneto įrenginys, iš kurio bus kuriama „medaus puodynė“ turi MIPS architektūra ir todėl virtualizacijos platforma turi turėti galimybė virtualizuoti tokio tipo „Linux“ operacinę ir failinę sistemas.

3.2.3 Tinklo srauto stebėjimo įrankio pasirinkimas

Renkantis tinklo srauto stebėjimo įrankį buvo taikomi šie kriterijai:

- Galimybė stebėti tinklo įrenginio ir žurnalizuoti tinklo srautą;
- Galimybė sužinoti IP adresą iš kurio atėjo užklausa;
- Galimybė atvaizduoti paketų turinį.

3.3 lentelė. *Tinklo srauto stebėjimo įrankių palyginimas*

	Tam tikro tinklo įrenginio stebėjimas	Suinčiančio IP adreso atvaizdavimas	Paketo turinio atvaizdavimas
iftop	+	+	-
Tcpdump	+	+	+
Wireshark	+	+	+

Išanalizavus tinklo srauto stebėjimo įrankius, buvo pasirinktas tcpdump įrankis. Jo privalumai yra sekantys: jis turi galimybė stebėti nurodyta tinklo įrenginys, atsekti iš kurio IP adreso buvo atsiusta užklausa ir jis jau yra įdiegtas į „Linux“ operacines sistemas.

3.2.4 „Linux“ operacinės sistemos branduolio kūrimo įrankio pasirinkimas

Renkantis „Linux“ operacinės sistemos branduolio kūrimo įrankį buvo keliami šie reikalavimai:

- Galimybė sukurti MIPS architektūros „Linux“ operacinės sistemos branduolį;
- Patogumas;
- Paprastumas;
- Patirtis;

3.4 lentelė. „Linux“ operacinės sistemos branduolio kūrimo įrankių paliginimas

	MIPS architektūros palaikymas	Patogumas	Patirtis	Kaina
Kbuild	+	-	-	Nemokomas
Buildroot	+	+	+	Nemokomas
Yocto Project	+	-	-	Nemokomas

Išanalizavus įrankius skirtus „Linux“ operacinės sistemos branduolio kūrimui, buvo apsistota ties „Buildroot“ įrankiu. Tai yra patogus įrankis, kuris gali greitai sukurti reikiamos konfigūracijos branduolį. Įrankis yra valdomas naudojant konfigūracinius failus.

3.3 Tyrimo atlikimo metodika

Šiame darbe bus taikomas empirinis kokybinis tyrimo metodas – eksperimentas. Eksperimento metu bus patikrinti ar virtualizuotas daiktų ir paslaugų interneto įrenginys gali veikti kaip „medaus puodynė“. Eksperimento metu įrenginys bus paleidžiamas lokaliame kompiuteriniame tinkle ir su nmap įrankio pagalba bus patikrinama ar įrenginys gali atsakinėti į siunčiamas užklausas ir žurnalizuoti visas jam siunčiamas užklausas. Taip pat bus patikrinama, kokią informaciją galima surinkti apie įrenginį pasinaudojus nmap įrankiu ir ar iš tos informacijos galima sužinoti, kad vietoj realaus daiktų ir paslaugų interneto įrenginio veikia „medaus puodynė“.

4. Metodas pritaikyti „medaus puodynės“ koncepciją daiktų ir paslaugų internetui.

4.1 Metodo realizacija

Šiame darbe daiktų interneto įrenginio virtualizavimui buvo pasirinktas „D-Link“ įmonės įrenginys „DSP-W215“. Tai yra išmanu kištukinis lizdas, kuri galima valdyti per mobiliąją aplikaciją. Šis įrenginys sugeba prisijungti prie bevielio tinklo ir palaiko IEEE 802.11n standartą (D-Link, 2015), kas reiškia kad jis gali prisijungti prie standartinio bevielio tinklo, kurį galima surasti pas eilinį vartotoją, o ne naudoja daiktų interneto įrenginiams skirto specifinio komunikacijos protokolo, ir tokio įrenginio aptikimas kompiuteriniame tinkle neturi sukelti įtarimų įsilaužėliams.

Dauguma daiktų interneto įrenginių, tarp kurių ir šiame darbe pasirinktas „D-Link“ įmonės įrenginys „DSP-W215“, kaip procesorinį įrenginį naudoja mikroprocesorių. Visa įrenginio programinė aparatinė įrangą yra sukuriama remiantis mikroprocesoriaus architektūra, todėl buvo svarbu surasti virtualizacijos platformą, kuri sugebėtų simuliuoti mikroprocesorinį įrenginį. Kaip virtualizacijos platforma, šiame eksperimente buvo naudojama „Qemu“ virtualizavimo platforma. „Qemu“ yra atviro kodo virtualizacijos platforma, gebanti virtualizuoti įvairius procesorinius įrenginius ir pilnas sistemas. „Qemu“ yra lengvai modifikuojama ir gali būti pritaikoma vartotojo poreikiams.

Taip pat šiame darbe bus naudojama „Buildroot“ atviro kodo programinė įranga, kuri yra skirta „Linux“ operacinės sistemos branduolio sukūrimui. Ši programinė įranga buvo pasirinkta dėl to, kad suteikia patogų būdą ne tik sukurti branduolį, bet ir pakeisti jo struktūrą, naudojamas bibliotekas, jo versiją. Su šia programine įranga, šiame darbe bus sukurtas „Linux“ operacinės sistemos branduolys, kuris bus pritaikytas imituojamam daiktų interneto įrenginiui.

4.1.1 Programinės aparatinės įrangos dekodavimas

Iš „D-link“ oficialaus tinklapio buvo parsisiųstas daiktų interneto įrenginio binarinis failas. Tyrimo metu buvo naudojamas išmanus kištukinis lizdas „DSP-W215“. Prieš paleidžiant daiktų ir paslaugų interneto įrenginio emuliaciją „Qemu“ virtualizacijos platformoje, gautas binarinis failas turi būti dekoduotas į failinę sistemą, iš kurios sekančiame etape bus sukurta ext2 tipo failinė sistema.

Programinės aparatinės įrangos dekodavimui bus naudojamas įrankių rinkinys programinės aparatinės įrangos modifikavimui (angl. firmware modification kit) iš „Googel Code“ (Google Code, n.d.). Šis rinkinys yra skirtas darbui su įterptinių sistemų ir daiktų interneto įrenginių programine įranga. Visas

rinkinys susideda iš kelių programinių įrangų, skirtų įvairiam darbui su programine aparatine įranga. Šiame darbe yra naudojamas programinės aparatinė įrangos dekodavimo skriptas „exteact-firmware.sh“.

Šis skriptas dekoduoja programinę aparatinę įrangą, kuri jam yra paduodama binarinio failo formatu. Pasinaudojant „binwalk“ programa jis sužino, kokio tipo failinė sistema yra užkoduota binariniame faile, nuo kokio baito prasideda failinė sistema. Su „binwalk“ pagalba yra sužinoma, kokio tipo baitų seka žodyje yra naudojama: kada reikšmingiausias baitais yra pirmas ar kada reikšmingiausias baitas yra paskutinis. Nuo baitų sekos priklausis kokio tipo „Linux“ operacinės sistemos branduolį naudosime kuriant virtualią mašiną. Taip pat, yra patikrinama kokio tipo kompresija yra naudojama: „gzip“, „xz“ ar „lzma“.

Gavus visą reikiamą informaciją, yra pradedamas failinės sistemos dekodavimas naudojant „Linux“ operacinės sistemos komanda dd. Ši komanda yra naudojama todėl, kad ji leidžia konvertuoti failą į kito tipo failą. Taip pat, ši komanda leidžia konvertuoti ne pilną failą, bet nurodžius baitų ribas. Iš ankščiau yra žinoma, nuo kokio baito binariniame faile prasideda failinė sistema ir su kokia kompresija buvo suarchyvuota failinė sistema. Pasinaudojus šia informacija, dd komandoje nurodome baitų tarpa kuriame yra failinė sistema ir failo formatą, kuris turi būti grąžintas po apdorojimo.

„Linux“ komanda dd mums grąžina suformuotą naują failą. Tai yra užarchyvuota failinė sistema. Sekantis žingsnis yra išarchyvuoti failinę sistemą. Kai buvo analizuojamas binarinis failas su „binwalk“ programa, buvo sužinota, kad failinės sistemos tipas yra squashfs. Failinės sistemos iš lzma archyvo ištraukimui yra naudojama „Unsquash“ programa. Su jos pagalba yra atkuriama pradinė failinė sistema, su visais daiktų interneto įrenginio „DSP-W215“ naudojamais failas.

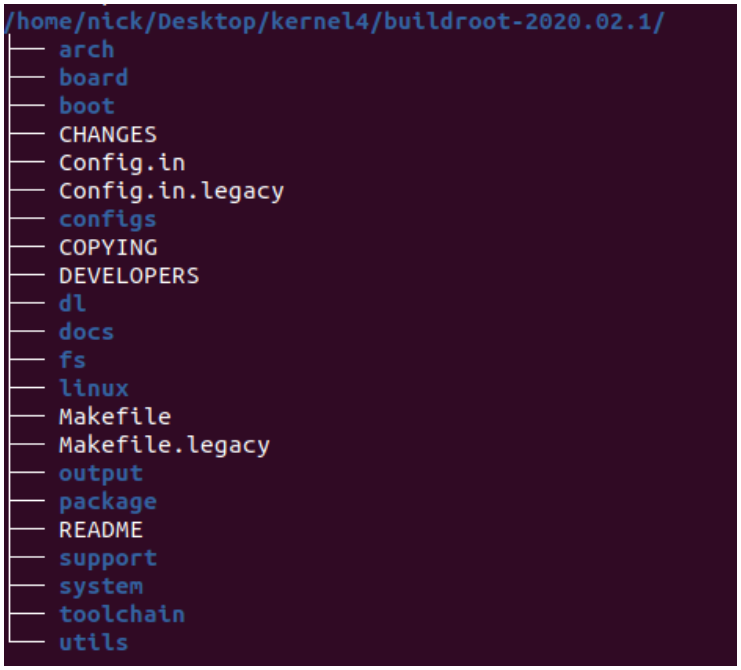
4.1.2 „Linux“ operacinės sistemos branduolio paruošimas

Dekodavus failinę sistemą reikia paruošti „Linux“ operacinę sistemą. Nėra galimybės naudoti įprastus „Linux“ operacinės sistemos branduolius, nes jie nėra pritaikyti veikimui ant įrenginių, naudojančių MIPS architektūrą, o šiame darbe tyrimams naudojamas daiktų interneto įrenginys naudoja MIPS architektūros procesorinį įrenginį. Tam, kad visos programos ir bibliotekos veiktų be trukdžių, reikia sukurti „Linux“ operacinės sistemos branduolį su reikiama architektūra, versija ir baitų seka.

„Linux“ operacinės sistemos branduolio kūrimui bus naudojama atviro kodo programinė įranga „Buildroot“. Ši programinė įranga leidžia greitai ir patogiai sukurti reikiamą „Linux“ operacinės sistemos branduolį. Taip pat vartotojui jau yra pasiūlyti paruošti branduolių konfigūraciniai failai.

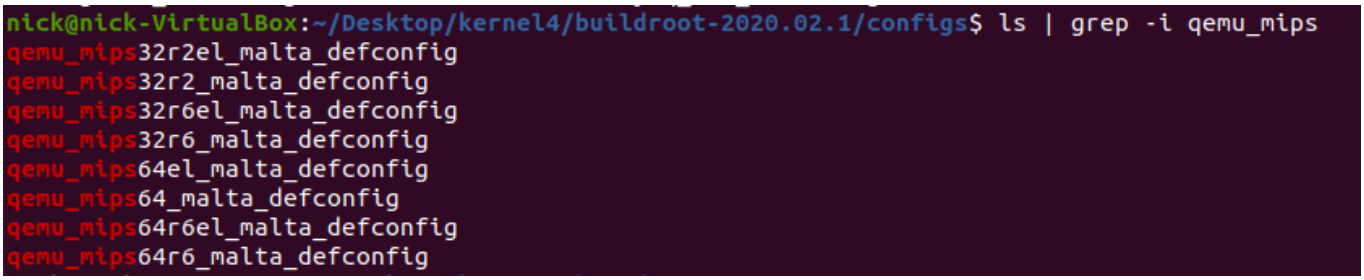
Iš oficialaus tinklapio, parsisiuntus ir išpakavus „Buildroot“, galima pamatyti 4.1 pav. parodyta aplankalų struktūrą. Patys svarbiausi aplankalai šiame kataloge yra:

- board – šiame aplankale yra saugomi konfigūraciniai failai, tokie kaip „Linux“ operacinės sistemos branduolio nustatymai. Šie konfigūraciniai failai yra skirti kiekvienai sistemai, kuriai „Buildroot“ gali sukurti „Linux“ operacinės sistemos branduolį.
- configs – šiame aplankale yra saugomi operacinių sistemų konfigūraciniai failai – defconfig. Nuo aukščiau aprašytų konfigūracinių failų jie skiriasi tuo, kad tai yra ne specifiniai, kiekvienai sistemai skirtos konfigūracijos, bet konfigūracijos, kurios skiriasi nuo konfigūracijų pagal nutylėjimą.
- Output/target – šiame aplankale yra saugomi sukurtos operacinės sistemos failai. Po to kai „Buildroot“ užbaigs operacinės sistemos surinkimą, ji išsaugos operacinės sistemos branduolį ir failinę sistemą šiame kataloge.



4.1 pav. „Buildroot“ aplanklų struktūra

4.2 pav. matoma, kad Config aplankale yra konfigūracijos skirtos „Qemu“ virtualizacijos platformai ir skirti MIPS tipo architektūrai.



4.2 Konfiguraciniai failai skirti „Qemu“ virtualizacijos platformai

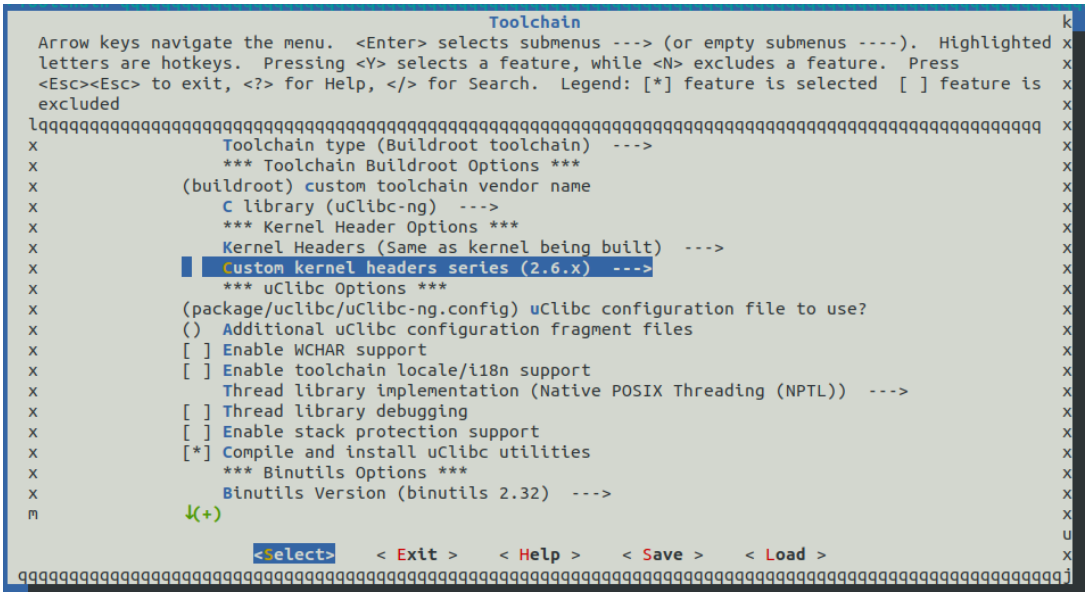
Turint daiktų interneto įrenginio failinę sistemą, galima lengvai nustatyti kokio tipo architektūrą jis naudoja ir kokio tipo „Linux“ operacinės sistemos branduolys turi būti sukurtas. Išanalizavus failinėje sistemoje esančias bibliotekas su komanda file, galima pamatyti, kad šios bibliotekos yra pritaikytos 32 bitų MIPS architektūrai, kuri naudoja žemesnio bito seką. Tai pavaizduota 4.3 pav. Taip pat, išanalizavus modulius, pastebėjome, kad jie yra pritaikyti „Linux“ versijai 2.6.32. Šios informacijos pakanka norint sukurti reikiamą „Linux“ operacinės sistemos branduolį. Config aplankale yra pasirenkamas „qemu_mips32r2el_malta_defconfig“. Pačiame konfiguraciniame faile yra pakeičiamos dvi eilutės: pirmą, kuri nurodo ar reikia naudoti 4.19 versijos branduolio antraštes ir antrą, kuri nurodo kokios versijos „Linux“ branduolys bus kuriamas.

```
nick@nick-VirtualBox:~/Desktop/dlink/fmk3/rootfs/lib$ file libgcc_s.so.1
libgcc_s.so.1: ELF 32-bit MSB shared object, MIPS, MIPS32 rel2 version 1 (SYSV), dynamically linked, no sectio
n header
```

4.3 pav. „Linux“ operacinės sistemos bibliotekos parametrai

Sekantis žingsnis - nustatyti, kokius įrankius naudos „Buildroot“, surinkinėdamas operacinės sistemos branduolį. Šiam tikslui panaudojama komanda „make menuconfig“.

Pasinaudojus šia komanda yra atidaroma KConfig vartotojo sąsaja, per kurią galima atlikti „Linux“ operacinės sistemos nustatymus ir kaip ši operacinė sistema bus surenkama. Šiame lange yra pasirenkama skiltis „Toolchain“. Šioje skiltyje yra nustatomi įrankiai ir parametrai, su kuriais bus sukuriama „Linux“ operacinė sistema. Analizuojant dekoduotą daiktų interneto failinę sistemą, aplankale, kuriame randasi operacinės sistemos bibliotekos, galima pastebėti, kad bibliotekos buvo sukurto naudojant uClibc kompiliatorių. Pasirinkama branduolio antraštės versija. Ji turi sutapti su kuriama „Linux“ operacinės sistemos branduolio versija, šiuo atveju tai bus 2.6.32. Konfigūravimo langas parodytas 4.4 pav.



4.4 pav. „Buildroot“ įrankių konfigūravimo langas

Atlikus visas reikiamas konfigūracijas yra paleidžiamas surinkimo procesas. Jam pasibaigus output/target aplankale gulės du failai. Vienas bus „Linux“ operacinės sistemos branduolys, o kitas bus failinė sistema. Kadangi, dekoduota failinė sistema jau yra, bus naudojamas tik operacinės sistemos branduolys. Užbaigus darbus su operacinės sistemos branduolio kūrimu, yra pereinama prie tinklo įrenginių konfigūravimo.

4.1.3 Tinklo įrenginių konfigūravimas ir stebėjimo sistemos prijungimas

Prieš paleidžiant „medaus puodynę“, ji turi būti prijungta prie kompiuterinio tinklo, kad ją galima būtų pasiekti iš išorės. Tam, kad virtuali mašina su virtualizuotu daiktų interneto įrenginiu galėtu būti pasiekiama per kompiuterinį tinklą, bus sukurti du tinklo įrenginiai: tinklo tiltas (angl. bridge) ir TAP tinklo įrenginys.

Tinklo tiltas yra įrenginys, kuris veikia OSI modelio kanalo lygmenyje. Šis įrenginys yra skirtas apjungti skirtingus kompiuterinio tinklo segmentus į vieną kompiuterinį tinklą. Veikiant tinklo tiltui, paketai yra perduodami remiantis MAC adresu. TAP įrenginys yra virtuali tinklo tvarkyklė, skirtas imituoti lokalaus kompiuterinio tinklo sujungimą. TAP įrenginys veikia OSI modelio kanalo lygmenyje (Олифер и Олифер, 2020). Naudojant šituos du tinklo įrenginius virtualizuotas įrenginys yra prijungiamas prie pagrindinės virtualios mašinos kompiuterinio tinklo ir leidžia išeiti į išorę. Programini kodas, kuri buvo naudojamas tinklo įrenginių kūrimui parodytas apačioje

```
1 #!/bin/sh
2 BRIDGE=br0
3 NETWORK=10.0.3.0
4 NETMASK=255.255.255.0
5 GATEWAY=10.0.3.255
6 brctl addbr "$BRIDGE"
7 brctl stp "$BRIDGE" off
8 brctl setfd "$BRIDGE" 0
9 ifconfig "$BRIDGE" "$GATEWAY" netmask "$NETMASK" up
10 echo 1 | dd of=/proc/sys/net/ipv4/ip_forward > /dev/null
11 iptables -A POSTROUTING -s $NETWORK/$NETMASK -j MASQUERADE
12 iptables -A FORWARD -i $BRIDGE -o $BRIDGE -j ACCEPT
13 iptables -A FORWARD -s $NETWORK/$NETMASK -i $BRIDGE -j ACCEPT
14 iptables -A FORWARD -d $NETWORK/$NETMASK -o $BRIDGE -m state –state
RELATED,ESTABLISHED -j ACCEPT
15 iptables -A INPUT -i $BRIDGE -p tcp -m tcp --dport 53 -j ACCEPT
16 iptables -A INPUT -i $BRIDGE -p udp -m udp --dport 53 -j ACCEPT
17 if test "$1" ; then
18    ifconfig "$1" 0.0.0.0 up
19    brctl addif "$BRIDGE" "$1"
20 fi
```

Tinklo įrenginių kūrimo programinis kodas

Pasinaudojant komandą brctl yra sukuriamas tinklo tilto įrenginys pagrindinėje virtualioje mašinoje. Šiam įrenginiui yra priskiriamas IP adresas – 10.0.3.0. Tam, kad tinklo paketai galėtų būti nukreipiami į įrenginius, prijungtus prie sukurto tinklo tilto, pagrindinėje virtualioje mašinoje yra įjungiama

paketų persiuntimo funkcija (angl. packet forwarding). Toliau yra sukuriamas TAP tinklo įrenginys ir padaromas pavaldžiu tinklo tilto įrenginiui.

Tam, kad tinklo paketai galėtu nevaržomai keliauti tarp įrenginių, su iptable įrankiu yra nustatomos IP paketų filtravimo taisyklės. Yra nustatoma, kad visi paketai priklausantys subdomenui 10.0.3.0/24, būtų nukreipiami į tinklo tiltą br0. Taip pat šiems paketams leidžiama pasiekti visus išorinius resursus, esančius kompiuteriniame tinkle. 4.5 pav. yra parodyti su komanda ip gauta informacija apie sukonfigūruotus tinklo įrenginius. TAP tinklo įrenginys yra automatiškai sukuriamas paleidžiant nauja „medaus puodynę“. Tai padaroma panaudojant funkciją pademonstruota tinklo įrenginių kūrimo programinio kodo 17,18,19 ir 20 eilutėje.

```
5: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 26:ca:24:8e:a1:3c brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.255/24 brd 10.0.3.255 scope global br0
       valid_lft forever preferred_lft forever
    inet 169.254.93.9/16 brd 169.254.255.255 scope global noprefixroute br0
       valid_lft forever preferred_lft forever
    inet6 fe80::d9e3:8588:1938:b398/64 scope link
       valid_lft forever preferred_lft forever
6: tap0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master br0 state UNKNOWN group default qlen
 1000
    link/ether 26:ca:24:8e:a1:3c brd ff:ff:ff:ff:ff:ff
    inet 169.254.243.137/16 brd 169.254.255.255 scope global noprefixroute tap0
       valid_lft forever preferred_lft forever
    inet6 fe80::c5c1:fdfe:9fee:228a/64 scope link
       valid_lft forever preferred_lft forever
```

4.5 pav. Tinklo tiltas ir TAP tinklo sąsaja

Kaip tinklo srauto stebėjimo sistema, šiame darbe buvo pasirinktas tcpdump įrankis. Jis atlieka visas reikiamas funkcijas: stebi visą įeinantį ir išeinantį tinklo srautą, atvaizduoja IP adresus, iš kurių ir į kuriuos yra kreipiamasi. Šis įrankis pilnai tinka „medaus puodynės“ koncepcijos įgyvendinimui. Sukūrus br0 tinklo tiltą, yra paleidžiamas tcpdump įrankis, kuriam yra nurodoma stebėto br0 tinklo įrenginio.

4.1.4 Virtualios mašinos įrenginio imitavimui paruošimas ir paleidimas

Aukščiau aprašytuose poskyriuose buvo dekoduota failinė sistema, išanalizuota daiktų interneto įrenginio architektūra ir jo pagrindu sukurtas „Linux“ operacinės sistemos branduolys. Sekantis žingsnis yra ext2 failinės sistemos kūrimas, kuri bus paleidžiama kartu su „Linux“ operacinės sistemos branduoliu. Pasinaudojant failine sistema, kuri buvo dekoduota iš originalaus binarinio failo, sukuriama ext2 tipo failinė sistema, kuri bus naudojama virtualioje mašinoje. Šiam tikslui buvo parašytas bash skriptas, kuris yra parodytas žymiau.

```
#!bin/bash
1  tar czf rootfs.tar.gz -C rootfs .
2 dd if=/dev/zero bs=1M count=512 of=rootfs.img
3 mkfs.ext2 rootfs.img
4 mkdir ext2
5 mount rootfs.img ext2/
6 tar xf rootfs.tar.gz -C ext2/
7 umount ext2/
```

Failinės sistemos kūrimo bash skriptas

Pirmoje eilutėje komanda užpakuoja visus failus, esančiu rootfs aplankale į rootfs.tar.gz failą. Toliau yra sukuriamas 512 MB failas rootfs su img išplėtimu. Sekantis žingsnis - sukurti failinę sistemą ext2 praeitame žingsnyje sukurtame faile. Sukuriamas aplankalas, prie jo yra prijungiama sukurta failinė sistema rootfs.img. Šiame aplankale mes išpakuojam pirmame žingsnyje supakuotą failinę sistemą. Atjungiame ext2 aplankalą nuo failinės sistemos. Atlikus visus žingsnius yra sukuriama nauja failnė sistema su duomenimis iš testuojamo daiktų interneto įrenginio.

Turint visus reikiamus komponentus „Qemu“ virtualizacijos platformoje paleidžiama virtuali mašina. Komanda, su kuria yra paleidžiama virtuali mašina, yra parodyta žemiau.

```
sudo qemu-system-mips -M malta -kernel vmlinux -hda rootfs.img -append "root=/dev/hda" -
serial stdio -display none -nic tap,mac=52:54:00:AB:21:D7
```

„Linux“ komanda su kuria yra paleidžiama virtuali mašina

„Qemu“ virtualizacijos platformai nurodome su kokiais parametrais paleisti sukurta virtualią mašiną. Su komanda „-kernel“ yra nurodamas anksčiau sukurtas „Linux“ operacinės sistemos branduolys. Su komanda „-hda“ nurodoma, kad reikia prijungti sukurta image tipo failą prie virtualios mašinos kaip diską ir „-append“ paduodama komanda, kad reikia paleisti jį kaip failinę sistemą. „-serial“ nurodoma, kad kaip valdymo sąsaja bus naudojama komandinė eilute. Su komanda „-nic“ yra įjungiamas tinklo įrenginys TAP ir jam yra suteikiamas MAC adresas. Įsijungus „Qemu“ virtualiai mašinai, galime naršyti po failinę sistemą ir matyti, kokie procesai yra paleisti. Prisijungus prie virtualizuoto daiktų interneto įrenginio su komanda ip, suteikiam jam IP adresą. Ir pabandome jį pasiekti iš kito tinklo, ir kaip matome, viskas veikia.

4.1.5 „Medaus puodynės“ testavimas

Aukščiau aprašytuose poskyriuose iš daiktų interneto įrenginio buvo sukurtas virtualus daiktų interneto įrenginys, kuris bus testuojamas kaip „medaus puodynė“. Testavimas bus atliekamas paleidžiant nmap įrankį iš įvairių įrenginių ir stebint, ar tinklo srauto stebėjimo įrankis tcpdump užfiksuoja einantį tinklo srautą į „medaus puodynę“. 4.6pav. yra pateikta „medaus puodynės“ tinklo įrenginių konfigūracija.

```
/ # ip addr
1: lo: <LOOPBACK> mtu 65536 qdisc noop qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 52:54:00:ab:21:d7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.10/24 brd 10.0.3.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:feab:21d7/64 scope link
       valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
```

4.6 pav. „Medaus puodynės“ tinklo sąsaja

Pirmo tyrimo metu, pasinaudojus nmap įrankiu, buvo nuskanuota „medaus puodynė“. Skanavimas vyko iš kitos virtualios mašinos, kurios IP adresas yra 10.0.3.20 ir esančios tame pačiame lokaliame tinkle. 4.7 pav. matome, ką užfiksavo tinklo srauto stebėjimo sistema tcpdump. Iš tcpdump žurnalo įrašų galima matyti, kad iš virtualios mašinos 10.0.3.20 buvo išsiustas ICMP protokolo paketas, ir „medaus puodynė“, kurios IP adresas yra 10.0.3.10, atsakė į šitą paketą, taip pranešdama potencialiam atakuojančiam asmeniui, kad yra veikiantis įrenginys.

```
22:10:35.908426 IP (tos 0x0, ttl 64, id 52377, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.3.20 > 10.0.3.10: ICMP echo request, id 45, seq 2, length 64
        0x0000:  4500 0054 cc99 4000 4001 53f2 0a00 0314  E..T..@.@.S.....
        0x0010:  0a00 030a 0800 e4cb 002d 0002 5eca c6a3  .........-..^...
        0x0020:  0009 0c6c 77fa 84d0 0000 0000 77f2 2510  ...lw.......w.%.
        0x0030:  77f1 a4e0 77fa 84d0 77fb 3000 0000 0001  w...w...w.0.....
        0x0040:  77f8 2c5c 0000 0000 77f8 a17c 77f1 ec08  w.,\....w..|w...
        0x0050:  77f1 ec04                                w...
22:10:35.909172 IP (tos 0x0, ttl 64, id 48115, offset 0, flags [none], proto ICMP (1), length 84)
    10.0.3.10 > 10.0.3.20: ICMP echo reply, id 45, seq 2, length 64
        0x0000:  4500 0054 bbf3 0000 4001 a498 0a00 030a  E..T....@.......
        0x0010:  0a00 0314 0000 eccb 002d 0002 5eca c6a3  .........-..^...
        0x0020:  0009 0c6c 77fa 84d0 0000 0000 77f2 2510  ...lw.......w.%.
        0x0030:  77f1 a4e0 77fa 84d0 77fb 3000 0000 0001  w...w...w.0.....
        0x0040:  77f8 2c5c 0000 0000 77f8 a17c 77f1 ec08  w.,\....w..|w...
        0x0050:  77f1 ec04                                w...
```

4.7 pav. „TCPDump“ įrankio žurnalo įrašas

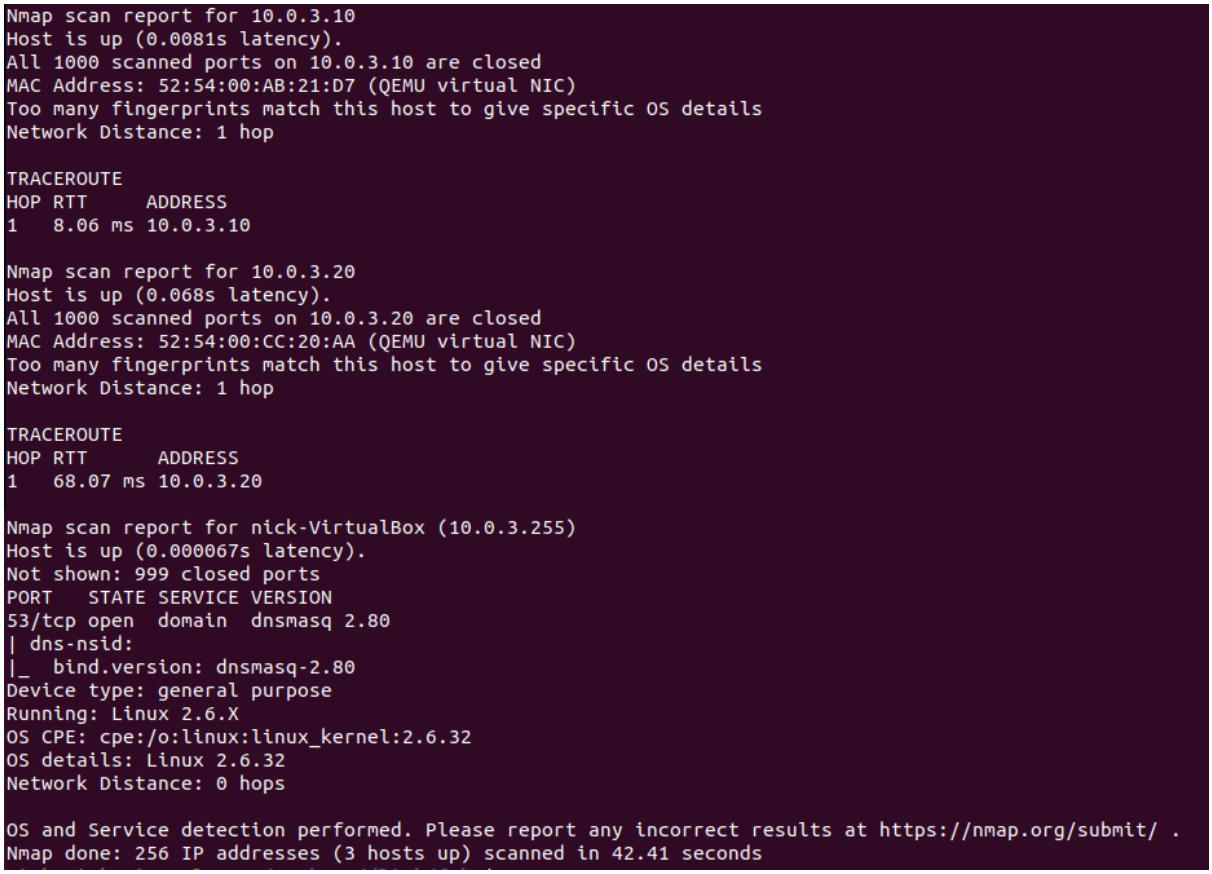
Antro tyrimo metu buvo pasiekti du tikslai vienu metu. Pirma tikslas buvo patikrinti ar sugeba tinklo srauto stebėjimo sistema „TCPDump“ užfiksuoti, kai yra daroma užklausa su kitu protokolu, negu su ICMP. Antras tikslas buvo patikrinti, ar gali viename tinkle veikti daugiau negu viena „medaus puodynė“.

Šiam tikslui buvo paleistos dvi „medaus puodynės“. Abiejose „medaus puodynėse“ veikė žiniatinklio serveris „Lighttpd“, kuris klausėsi 80 porto. Pasinaudojus „nmap“ įrankiu, buvo atliktas kompiuterinio tinklo segmento 10.0.3.0/24 skanavimas. Patikrinus tinklo srauto stebėjimo sistemą, galima pamatyti, kad buvo užfiksuotas kreipimasis ir atsakymas iš 80 porto, kur veikia paleistas žiniatinklio serveris. 4.8 pav. parodytas atsakymas iš žiniatinklio serverio į nmap įrankio užklausą. Matome, kad komunikacija vyko pasinaudojant TCP protokolu. Taip pat matome, kad tinklo srauto stebėjimo sistema užfiksavo atsakymus iš dviejų „medaus puodynių“. Tokių būdų matome „medaus puodynių“ skaičius tinkle yra apribojamas tik pačio IP adresų diapazonu. Kol skirtingoms „medaus puodynėm“ bus suteikiami skirtingi IP adresai, tol jos bus traktuojamos kaip skirtingi įrenginiai.

```
23:05:53.284308 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 44)
    10.0.3.20.80 > 10.0.3.255.46868: Flags [S.], cksum 0x5daa (correct), seq 958397221, ack 207498337, win 642
40, options [mss 1460], length 0
        0x0000:  4500 002c 0000 4000 4006 1fba 0a00 0314  E..,..@.@.......
        0x0010:  0a00 03ff 0050 b714 391f fb25 0c5e 2c61  .....P..9..%.^,a
        0x0020:  6012 faf0 5daa 0000 0204 05b4            `...]........
23:05:53.507850 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 44)
    10.0.3.10.80 > 10.0.3.255.46868: Flags [S.], cksum 0x82b7 (correct), seq 3028769466, ack 207498337, win 64
240, options [mss 1460], length 0
        0x0000:  4500 002c 0000 4000 4006 1fc4 0a00 030a  E..,..@.@.......
        0x0010:  0a00 03ff 0050 b714 b487 5aba 0c5e 2c61  .....P....Z..^,a
        0x0020:  6012 faf0 82b7 0000 0204 05b4            `...........
```

4.8 pav. Atsakymo į užklausą iš žiniatinklio serverio žurnalo įrašas

Trečiame tyrime buvo tikrinama, kokie duomenys gali būti surinkti apie veikiančias „medaus puodynes“. „Nmap“ įrankis suteikia galimybę surinkti tokius duomenys apie veikiančius įrengius, tokius kaip operacinė sistema, MAC adresai. Atlikus tokio tipo skanavimą gavome rezultatus, pavaizduotus 4.9 pav. Tyrimo metu buvo nuskanuojamos antrame tyrime paleistos dvi „medaus puodynės“. Iš rezultatų matoma, kad „medaus puodynės“ kaip savo operacinės sistemas, parodo „Linux“ operacinę sistemą, versija 2.6.32. Taip pat grąžina „medaus puodynės“ paleidimo metu suteikta MAC adresą, tačiau pažymi, kad tai yra „Qemu“ virtualus lokalaus kompiuterinio tinklo portas.



4.9 pav. „Medaus puodynių“ skanavimo rezultatai

Iš atliktų tyrimų matome, kad sukurtos „medaus puodynės“ atlieka savo funkcijas. Naudojant „nmap“ įrankį galima aptikti „medaus puodynes“ ir su jose veikiančiais žiniatinklio serveriais galima užmegzti komunikaciją. Aptiktas minusas yra virtualus lokalaus kompiuterinio tinklo portas, kuris yra pažymimas kaip virtualizuotas.

Išvados

- Atlikus literatūros analizę ir išanalizavus esamas „medaus puodynes“ buvo išsiaiškinta, kad daiktų ir paslaugų internetas reikalauja daug dėmesio saugos srityje. Tokie įrenginiai yra nuolat prijungti prie interneto ir dažniausiai juos gali pasiekti bet kuris interneto vartotojas. Atliktos analizės parodo, kad galima lengvai užvaldyti įrenginius. Todėl turi būti kuriami ir vystomi saugos mechanizmai pritaikyti daiktų ir paslaugų interneto įrenginiams.

- Atlikus pasiūlyto metodo realizaciją ir testavimą, buvo įsitikinta, kad metodas sugeba virtualizuoti realų daiktų ir paslaugų įrenginį ir, ant virtualaus įrenginio pagrindo, sukurti „medaus puodynę“. Visas tinklo srautas yra matomas, sekamas ir žurnalizuojamas.

- Pasiūlytas metodas įrodo, kad „medaus puodynės“, pritaikytos daiktų ir paslaugų internetui, sukūrimas nieko nekainuoja, nes buvo naudojami nemokomi atviro kodo programinė įranga ir įrankiai. Tokiu būdu, galima padidinti daiktų ir paslaugų interneto infrastruktūros saugumą, nepadidinus kaštų.

Visi šiame darbe keliami uždaviniai buvo įvykdyti. Buvo atlikta „medaus puodynės“ veikimo analizė. Remianti atlikta analize buvo pasiūlytas metodas, kuris padėtų pritaikyti „medaus puodynės“ koncepciją daiktų ir paslaugų interneto įrenginiams. Atlikus siūlomo metodo realizaciją ir testavimą buvo padaryta išvada, kad siūlomas metodas yra veikiantis. Su juo pagalba yra įmanoma virtualizuoti egzistuojanti daiktų ir paslaugų interneto įrenginį ir paversit jį „medaus puodyne“, kuri gebės stebėti ir žurnalizuoti tinklo srautą.

LITERATŪROS SĄRAŠAS

Abe S., Tanaka Y., Uchida Y., Horata S. 2017. Tracking Attack Sources based on Traceback Honeypot for ICS Network 2 . TRACEBACK HONEYPOT SYSTEM.

Ahamed J., Rajan A. V 2016. Internet of Things (IoT): Application Systems and Security Vulnerabilities, *5th International Conference on Electronic Devices, Systems and Applications*: 1–5.

Dowling S., Schukat M., Melvin H. 2017. A ZigBee honeypot to assess IoT cyberattack behaviour, *2017 28th Irish Signals and Systems Conference, ISSC 2017*: 0–5. DOI: 10.1109/ISSC.2017.7983603 .

Egupov A.A., Zareshin S. V, Yadikin I.M., Silnov D.S. 2017. Development and Implementation of a Honeypot- Trap, *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*: 382–385.

Guarnizo J., Tambe A., Bhunia S.S., Ochoa M., Tippenhauer N., Shabtai A., Elovici Y. 2017. SIPHON: Towards Scalable High-Interaction Physical Honeypots. DOI: 10.1145/3055186.3055192 .

Luo T., Xu Z., Jin X., Jia Y., Ouyang X. 2017. IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices, *Blackhat*Prieiga per internetą: https://www.blackhat.com/docs/us-17/thursday/us-17-Luo-Iotcandyjar-Towards-An-Intelligent-Interaction-Honeypot-For-IoT-Devices-wp.pdf.

Mendez D.M., Papapanagiotou I., Yang B. 2017. Internet of Things: Survey on Security and Privacy, (November)Prieiga per internetą: http://arxiv.org/abs/1707.01879.

Mokube I., Adams M. 2007. Honeypots: Concepts, Approaches, and Challenges, *Proceedings of the 45th annual southeast regional conference on - ACM-SE 45*: 321–326. DOI: http://dx.doi.org/10.1145/1233341.1233399 .

Pa Y.M.P., Suzuki S., Yoshioka K., Matsumoto T., Kasama T., Rossow C. 2016. IoTPOT: A Novel Honeypot for Revealing Current IoT Threats, *Journal of Information Processing* 24(3): 522–533. DOI: 10.2197/ipsjjip.24.522 .

Samtani S., Yu S., Zhu H., Patton M., Matherly J., Chen H. Identifying Supervisory Control and Data Acquisition (SCADA) Devices and their Vulnerabilities on the Internet of Things (IoT): A Text Mining Approach, : 1–11.

Usmonov B., Iskhakov A., Shelupanov A., Iskhakova A., Meshcheryakov R. 2017. The cybersecurity in development of IoT embedded technologies.

Williams R., McMahon E., Samtani S., Patton M., Chen H. 2017. Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach, *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*: 179–181. DOI: 10.1109/ISI.2017.8004904 .

Zhong C. Le, Zhu Z., Huang R.G. 2016. Study on the IOT architecture and gateway technology, *Proceedings - 14th International Symposium on Distributed Computing and Applications for Business, Engineering and Science, DCABES 2015*: 196–199. DOI: 10.1109/DCABES.2015.56 .

IDC: The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast. 2019 Prieiga per internetą: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

D-Link, 2015. mydlink Home Smart Plug DSP-W215. Datasheet

Google Code: Firmware Modification Kit (n.d). Prieiga per internetą https://code.google.com/archive/p/firmware-mod-kit/

Олифер Виктор, Олифер Наталья *Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание*. — СПб.: Питер, 2020. — 1008 с.: ил. — (Серия ≪Учебник для вузов≫).
ISBN 978-5-4461-1426-9