KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Gintautas Beržunskis

SAUGAUS NUOTOLINIO MARŠRUTO PARINKTUVŲ VALDYMO TINKLO TYRIMAS

Baigiamasis magistro darbas

Vadovas
lekt. dr. Dangis Rimkus

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

SAUGAUS NUOTOLINIO MARŠRUTO PARINKTUVŲ VALDYMO TINKLO TYRIMAS

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) lekt. dr. Dangis Rimkus
(data)

Recenzentas
(parašas) Doc. dr. Jevgenijus Toldinas
(data)

Projektą atliko
(parašas) Gintautas Beržunskis
(data)

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Gintautas Beržunskis

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga, M4096M21

(Studijų programos pavadinimas, kodas)

„Saugaus nuotolinio maršrutizatorių valdymo tinklo tyrimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 16 m. gegužės 18 d.

Kaunas

Patvirtinu, kad mano Gintauto Beržunskio baigiamasis projektas tema „Saugaus nuotolinio maršruto parinktuvų valdymo tinklo tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

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

Beržunskis, Gintautas. „Saugaus nuotolinio maršruto parinktuvų valdymo tinklo tyrimas“. Magistro baigiamasis projektas / vadovas lekt. dr. Dangis Rimkus; Kauno technologijos universitetas, Informatikos fakultetas, Kompiuterių katedra.

Kaunas, 2016. 51 p.

SANTRAUKA

Darbe pateikiamas saugaus nuotolinio maršruto parinktuvų prisijungimo prie valdymo tinklo sprendimas, kurio pagrindinis tikslas - automatiškai ir saugiai sujungti maršruto parinktuvus į vieningą valdymo tinklą.

Atlikta probleminės srities analizė, tai yra, išanalizuoti konkurentų sprendimai naudojami valdymo tinkluose, taip pat atlikta valdymo protokolų analizė. Analizuojant esamus valdymo protokolus buvo atsižvelgiama į klientų keliamus reikalavimus kuriamam saugiam valdymo tinklui. Atlikus analizę buvo nuspręsta naudoti JSON-RPC protokolą kartu su OpenVPN. JSON-RPC valdymo protokolas pasirinktas dėl suteikiamos galimybės vykdyti komandas maršruto parinktuve, kurių nereikia iš anksto aprašyti. OpenVPN pasirinktas dėl duomenų, siunčiamų tarp serverio ir maršruto parinktuvo šifravimo galimybės bei dėl tiesioginio maršrutų parinktuvo pasiekiamumo, nes dauguma maršruto parinktuvų neturi išorinio IP adreso.

Projektuojant sistemą buvo nuspręsta, kad maršruto parinktuvai privalo prie valdymo tinklo prisijungti automatiškai. Tam tikslui buvo suprojektuotas autentifikavimo servisas, prie kurio maršruto parinktuvai jungiasi tam, kad gautų sertifikatus ir konfigūraciją. Valdymo tinklas buvo suprojektuotas taip, kad prie jo galėtų prisijungti tik patvirtinti maršrutizatoriai, turintys sertifikatus.

Kuriant sistemą buvo realizuotas automatinis maršruto parinktuvų prisijungimas prie OpenVPN autentifikavimo tunelio, kuriame jie yra įdentifikuojami. Vėliau maršruto parinktuvai yra priskiriami jiems priklausančiai įmonei, jiems sugeneruojami sertifikatai su konfigūracija ir perduodami į kiekvieną maršruto parinktuvą. Sertifikatų generavimo programa parašyta C programavimo kalba, naudojant SSL biblioteką. Sertifikatų generavimo užduotys yra rikiuojamos į eilę, kad nebūtų per daug apkraunamas serverio procesorius. Užduotys sistemoje yra vykdomos po kelias iš karto, tam buvo realizuotas kelių gijų palaikymas. Pagrindinė programos dalis stebi visas gijas ir skirsto joms naujas užduotis. Maršruto parinktuvui gavus naują konfigūraciją ir sertifikatus yra vykdomas jo prisijungimas prie kliento saugaus valdymo tinklo.

Atliekant sistemos tyrimą buvo nustatyta, kiek laiko maršruto parinktuvas užtrunka prisijungdamas prie valdymo tinklo pirmą kartą. Tyrimo metu buvo prijungiamas skirtingas maršruto parinktuvų kiekis ir analizuojama, kokią įtaką jis daro prisijungimo laikui. Taip pat buvo simuliuojama DDoS ataka prieš serverį ir tiriamas jos vaidmuo serverio veikimo bei naujų maršruto parinktuvų prisijungimo trukmės kitimo kontekste.

Beržunskis, Gintautas. Secure Remote Router Management Network Research: *Master's* thesis in Informatics Engineering supervisor lect. prof. Dangis Rimkus. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: 07T Informatics Engineering

Key words: software protection, illegal use, functionality, cryptography

Kaunas, 2016. 51 p.

SUMMARY

This paper explores means for a safe remote router connection to the network management solution, with the main goal to automatically and securely connect routers to a unified network management system.

A problem domain analysis was performed during which the competitors' solutions being used for network management along with a management protocols were assessed. During the study of existing management protocols the customer requirements for the safe management of the network have also been taken into account. After analysis it was decided to use JSON-RPC protocol on top of OpenVPN based secure tunnels. JSON-RPC management protocol was selected for providing the ability to execute commands on a router that do not need to be defined in advance. OpenVPN was selected for it's ability to encrypt data sent between the server and the router as well as a way to have an access to the router through a secure tunnel, because most routers do not have an external IP address.

In the process of designing the system it was decided that the routers are required to connect to the management network automatically. To achieve this kind of functionality an authentication service was created to act as an initialization vector for new devices. A "clean" router would first connect to this service in order to obtain custom certificates and configuration for OpenVPN tunnel. The main network management system has been designed in a way that only authorized router would be able to connect using custom certificates for its data tunnel encryption.

An automatic router connection to the OpenVPN authentication tunnel has been implemented. At this stage routers are identified and classified as belonging to a particular customer after which the certificates are generated and together with configuration uploaded to each device. Certificate generation program is written in the C programming language using the SSL library. All the certificate generation tasks are being queued up to be executed in controlled fashion as to not take up all the server resources. A multithreading support was implemented for the program to be able to execute multiple tasks at once. The main program thread monitors all worker threads and distributes new tasks to them. After receiving new configuration and certificates router connects to the network management system using a safe encrypted tunnel.

During the testing phase it was measured how long it takes for the router to connect to the network management system for the first time and what impact a different number of devices have on that time. A simulated DDoS attack was performed against the server to compare device connection time change against the baseline.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

OpenVPN – atvirojo kodo projektas, kuriantis virtualaus privataus tinklo (VPN) programinį sprendimą, skirtą koduojamu ryšiu tarpusavyje sujungti du nutolusius įrenginius.

4G arba LTE – tai mobiliojo ryšio technologija, pažangesnė už 3G ir GSM technologiją.

TR-069 – vienas iš tinklo protokolų, skirtas tinkle veikiantiems įrenginiams stebėti ir valdyti

SNMP – vienas iš tinklo protokolų, skirtas tinkle veikiantiems įrenginiams stebėti ir valdyti.

JSON-RPC – vienas iš tinklo protokolų, skirtas tinkle veikiantiems įrenginiams stebėti ir valdyti.

IT – informacinės technologijos.

ICMP – interneto kontrolės žinučių protokolas.

PING – įrankis, naudojamas patikrinti ar kompiuteris yra pasiekiamas per IP tinklą.

DDOS – yra būdas kenkti serveriui, kad jis nepajėgtų aptarnauti vartotojų užklausų.

ĮVADAS

Atliekamas darbas priklauso informatikos studijų programai, informacijos ir informacinių technologijų saugos specializacijai.

Norint įmonei didinti maršruto parinktuvų pardavimų kiekius, neužtenka pasiūlyti vis naujesnių, greitesnių maršruto parinktuvų. Klientai pirkdami maršruto parinktuvus dažnai atkreipia dėmesį į tai, kaip lengvą ir patogu bus juos valdyti, bei administruoti kelis šimtus ar net kelis tūkstančius. Kaip pavyzdį pateiksiu LTE maršruto parinktuvus, naudojamus vėjo jėgainėse. Įmonės tikslas, kuo mažesnėmis sąnaudomis administruoti visus maršruto parinktuvus. Nesant vieningam saugiam valdymo tinklui, atsiranda nemažai išlaidų, kai norima konfigūruoti maršruto parinktuvus. Tinklo administratoriui reikia vykti į vietą, kur yra maršruto parinktuvas. Problemos nebūtų, jei įmonės turėtų tik kelis maršruto parinktuvus, bet kai maršruto parinktuvų yra keli šimtai ar net keli tūkstančiai, tai įmonėms kainuoja nemažai kaštu.

Iš čia atsiranda ir problemos įmonėms, kaip užtikrinti, patogų maršrutizatorių valdymą:

- kaip stebėti maršruto parinktuvų veikimą,
- kaip centralizuotai valdyti maršruto parinktuvus,
- kaip sutrumpinti administratoriaus darbą prižiūrint maršruto parinktuvus,
- kaip prijungti maršruto parinktuvą į vieningą tinklą.

Norint įmonėms sumažinti administravimo kaštus, kurios perka didelius kiekius maršruto parinktuvų, yra reikalingas saugus valdymo tinklas. Įmonės nusipirkusios didelį kiekį maršruto parinktuvų, taip pat gauna ir prisijungimą prie UAB „Teltonika“ įmonės serverio, saugaus nuotolinių maršruto parinktuvų valdymo tinklo. Iš kur jie gali saugiai valdyti savo maršruto parinktuvus. Įjungus maršrutizatorių jis pats susijungia su autentifikavimo serveriu, kur jam yra sugeneruojami sertifikatai, ir tada jungiasi prie įmonės saugaus valdymo tinklo.

Darbo problematika ir aktualumas

Saugus nuotolinių maršruto parinktuvų automatinis prisijungimas prie valdymo tinklo. Automatinis OpenVPN sertifikatų perdavimas nutolusiems maršrutų parinktuvams.

Pagrindinis darbo tikslas

Realizuoti saugų nuotolinių maršruto parinktuvų valdymo tinklą, prie kurio maršruto parinktuvai prisijungia automatiškai.

Tikslui pasiekti iškelti uždaviniai:

- Atlikti literatūros analizę dėl esamų valdymo tinklo protokolų,
- Sukurti sertifikatų perdavimo metodą,
- Realizuoti sertifikatų perdavimo metodą,
- Atlikti eksperimentinius tyrimus prijungiant maršruto parinktuvus ir DDOS atakas prieš serverį.

Darbo rezultatai ir jų svarba

Įvertinus analizės dalyje gautus rezultatus, buvo sukurtas ir realizuotas saugus nuotolinio maršruto parinktuvų valdymo tinklas. Realizuota sistema palengvina maršruto parinktuvų administravimą ir taupo įmonių kaštus. Realizuotas metodas, kuris naudojamas maršruto parinktuvų prisijungimui prie saugaus valdymo tinklo yra saugus ir pilnai automatizuotas. Realizuoto metodo dėka maršruto parinktuvų, prijungimas prie įmonės valdymo tinklo trunka palyginus mažai laiko ir visa tai įvyksta automatiškai, be administratorių konfigūravimo. Pagal tyrimo metu gautus rezultatus, nustatyta kokią įtaką sistemos veikimui turi DDOS ataka prieš serverį.

Darbo struktūra

Šis darbas buvo suskirstytas į kelis etapus, kur kiekvienas etapas atitinką šio dokumento skyrių:

- Pirmas skyrius – probleminės srities analizės skyrius. Šiame skyriuje atlikta analizė į esamus problemos sprendimų variantus. Esamų valdymo tinklų protokolų analizė, bei palyginimas su klientų norima sistema. Išanalizuoti metodai, naudojami perduoti sertifikatus.
- Antras skyrius – sprendimas. Šiame skyriuje aprašoma, kaip turėtų veikti sistema ir atskiros jos dalys. Aprašytos ir pavaizduotos sistemos veikimo diagramos, bei kuriamos sistemos konfigūracija. Taip pat šioje dalyje yra pavaizduotas ir aprašytas, kuriamos sistemos modelis.
- Trečias skyrius – realizacija. Šiame skyriuje aprašoma, kaip realizuota sistema. Taip pat yra pavaizduotas sistemos modelis ir aprašytas pilnas sistemos veikimas.
- Ketvirtas skyrius – tyrimas. Šiame skyriuje atlikti tyrimai, kiek trunka maršruto parinktuvų prisijungimas prie saugaus valdymo tinklo. Taip pat buvo atliktas tyrimas, kaip maršrutizatoriams jungiantiems prie serverio, turi įtakos DDOS ataką vykdoma prieš serverį. Atliktas siunčiamų paketų analizė tarp maršrutizatoriaus ir valdymo tinklo.
- Penktas skyrius – rezultatų apibendrinimas ir išvados. Šiame skyriuje pateiktas tikslų, užsibrėžtų uždavinių įgyvendinimo rezultatai ir gautos atlikto darbo išvados.
- Šeštas skyrius – atliekant darbą, naudotos literatūros sąrašui pateikti.
- Septintas skyrius – skyrius skirtas priedams.

1. PROBLEMINĖS SRITIES ANALIZĖ

Kuo toliau tuo vis daugiau atsiranda įrenginių, kuriems reikalingas internetas. Dažnai internetas reikalingas ne pačio įrenginio veikimui, o stebėjimui, kaip jis veikia. Pavyzdžiu galėtų būti vėjo jėgainės, kurioms internetas nereikalingas, bet internetas reikalingas jėgainių veikimui stebėti. Problema išsprendžiama montuojant LTE maršruto parinktuvą. Esant dideliam kiekiui maršruto parinktuvų, juos tampa sunku suvaldyti, administruoti ar pajunginėti į vieningą valdymo tinklą. Todėl įmonėms reikalingas sprendimas, kaip kuo paprasčiau sujungti šiuos maršruto parinktuvus į vieningą valdymo tinklą, tuo pačiu užtikrinant, kad tinklas yra saugus.

1.1. Analizės tikslas

Analizės tikslas yra išanalizuoti kitų įmonių siūlomus sprendimus, kaip prijungti maršruto parinktuvams į saugų valdymo tinklą, rasti siūlomo sprendimo geriausias ir blogiausias savybes. Atlikus analizę sukurti sistemą, kuri bus pranašesnė už kitas sistemas ir tuo pačiu atitiktų įmonės, kuriai ir bus kuriama ši sistema, keliamus reikalavimus.

1.2. Tyrimo objektas, sritis ir problema

Tyrimo sritis yra informacinių technologijų sauga. Sprendžiama problema – saugus nuotolinių maršruto parinktuvų prijungimas prie valdymo tinklo. Tyrimo dalyje bus atlikti sukurtos sistemos tyrimai, kur bus pavaizduotos sistemos stipriosios ir silpnosios pusės.

1.3. Tyrimo objekto naudotojų analizė

Atlikus įmonės, kuriai kuriamas produktas ir jos klientų, kurie naudosis šia sistema analizę, buvo sudaryti funkciniai reikalavimai, kurie buvo reikalingi kuriamai sistemai. Nefunkciniai sistemos reikalavimai yra svarbūs, nes į juos bus atsižvelgiama kuriant saugaus nuotolinio maršruto parinktuvo valdymo tinklo modelį. Pagal surinktus įmonės ir jos klientų nefunkcinius reikalavimus bus toliau lyginamos dabar naudojamos technologijos ir metodai.

Nefunkciniai reikalavimai kuriamai sistemai:

- Turi būti galimybė ne tik pranešinėti maršruto parinktuvui apie savo statusą, bet ir sistemai užsiklausti.
- Dažniausiai maršruto parinktuvai būna ne viename tinkle ir neturi išorinio IP adreso.
- Turi būti galimybė pasiekti maršruto parinktuvo interneto sąsają per saugų maršruto parinktuvų valdymo tinklą.
- Turi būti galimybė pasiekti maršruto parinktuvo terminalą per saugų maršruto parinktuvų valdymo tinklą.

1.4. Esamų problemos sprendimo metodų analizė

Išanalizuotos kelios pagrindinės įmonės konkurentės, kurios taip pat gaminą LTE maršruto parinktuvus arba tik siūlo jų valdymo tinklą. Įmonės, kurios siūlo tik maršruto parinktuvų valdymą ir jų negamina, savo sprendimą paremia standartiniais protokolais, tokias kaip TR-069, SNMP ir kt. Visi protokolai, skirti valdyti maršruto parinktuvams, yra panašus tiek savo galimybėmis, tiek saugumu. Dažniausiai, norint užtikrinti saugų duomenų siuntimą, naudojama HTTPS arba SSL. Standartiniai protokolai yra gana populiarūs, dėl savo suderinamumo su skirtingais įrenginiais ir yra labiau skirti ne tinklui valdyti, o patiems įrenginiams stebėti. Tarp įmonių siūlančių tik maršrutizatorių valdymą labiausiai paplitęs „Friendly technologies“ [1] siūlomas sprendimas, kuris paremtas TR-069 protokolu, kuris yra suderinamas su dauguma tinklo maršrutizatorių.

Tarp įmonių, kurios gamina maršrutizatorius, dažniausiai būną naudojami nestandartiniai sprendimai. Tai yra keli į vieną sprendimą sujungti protokolai:

- OpenVPN kartu su TR-069,
- OpenVPN kartu su SNMP,
- OpenVPN kartu su SSH.

Tokie sprendimai dažnai ir naudojami klientų, kurie perka didelius kiekius maršruto parinktuvų. Tokios sistemos yra geresnės, nes jos būna pritaikytos būtent tiems gaminiams. Sistemos siūlomos įvairiais variantais: galima naudoti maršruto parinktuvą kūrėjų serveriuose, taip pat galima išsipirkti serverį ir naudoti savo.

Pagrindiniai įmonės konkurentai, kurie taip pat siūlo industrinius LTE maršruto parinktuvus, yra įmonė „Connel“, kuri siūlo „Rseenet“ [2] sprendimą. Sprendimas paremtas SNMP protokolu. Norint pridėti įrenginį, reikia maršruto parinktuvo duomenis suvesti į sistemą ir į maršruto parinktuvą. Sistema pagal sukonfigūruotą laiką užsiklausia maršruto parinktuvų apie jų statusą. Didžiausias sprendimo trūkumas yra tai, kad įrenginiai turi būti tame pačiame tinkle kartu su serveriu arba turėti išorinį IP adresą, kad serveris galėtų užklausti informacijos iš maršrutizatoriaus.

1.5. Esamų valdymo tinklo protokolų analizė

Šioje dalyje bus palyginami protokolai, kurie dažniausiai naudojami maršruto parinktuvų valdymo tinkle. Egzistuojančiuose maršruto parinktuvų valdymo ir stebėjimo tinklo modeliuose dažniausiai naudojami protokolai:

- TR-069
- SNMPv3
- JSON-RPC

1.5.1. TR-069

TR-069 – tai techninė specifikacija, aprašanti abonentinės įrangos valdymo per internetą protokolą. Dažnai naudojamas maršruto parinktuvų valdymo protokolas ir yra stipriai paplitęs tarp sistemų skirtų maršruto parinktuvų stebėjimui ir valdymui. Visi duomenys tarp serverio ir įrenginio perduodami XML forma. Duomenys siunčiami per HTTP arba jei norima užtikrinti, kad siunčiami duomenys būtų saugūs galima naudoti HTTPS protokolą [3].

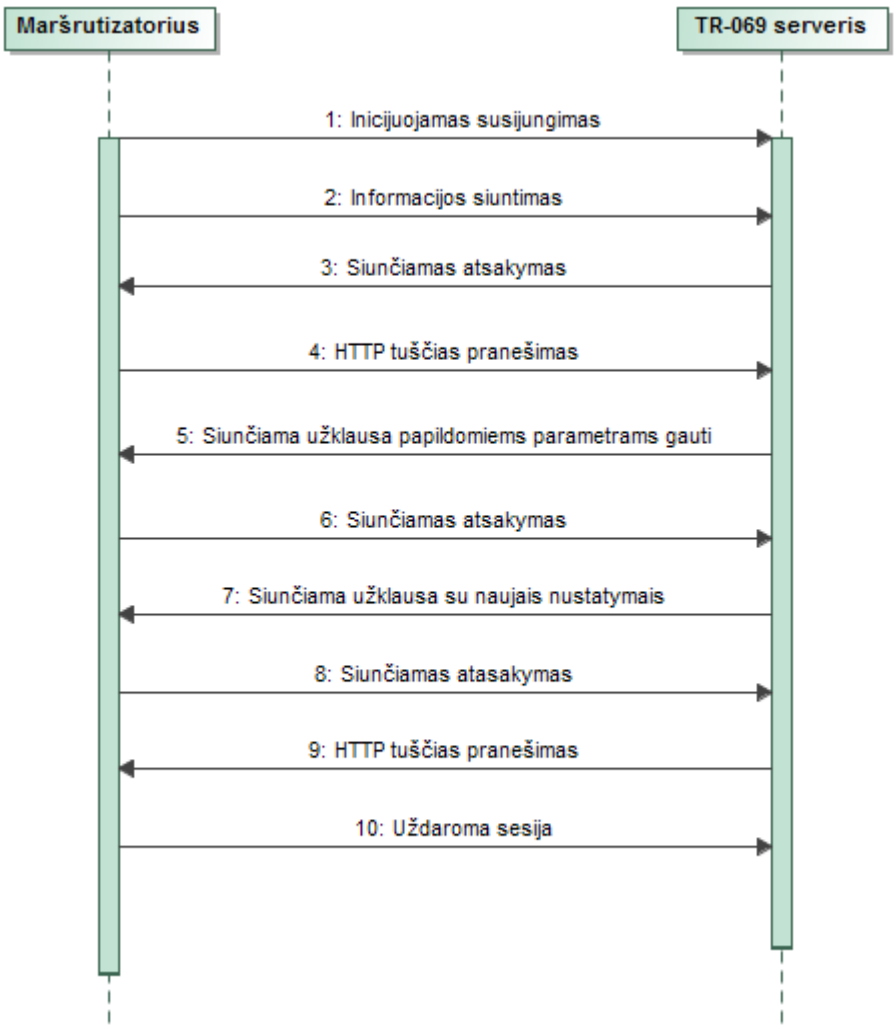
Pagrindinės TR-069 metodo galimybės yra gauti informaciją iš maršruto parinktuvų ir keisti nustatymus. Taip pat yra galimybė ir keisti tam tikrus parametrus maršruto parinktuve. Kai kurie įrenginiai palaiko programinės įrangos perrašymą iš serverio ir „log“ atsiuntimą iš maršruto parinktuvo.

TR-069 metodo trūkumas yra tas, kad maršruto parinktuvas pats kreipiasi į serverį pagal sukonfigūruotą laiką. Serveris nevaldo gaunamų užklausų skaičiaus. Norint atlikti veiksmus su maršruto parinktuvu, yra du variantai kaip tai gali atlikti serveris:

1. Jei įrenginys turi išorinį IP adresą arba serveris yra tame pačiame tinkle, kaip maršruto parinktuvas, tai serveris gali nusiųsti signalą per HTTP arba HTTPS protokolą maršruto parinktuvui. Gavus signalą, maršruto parinktuvas kreipiasi į serverį ir gauna nurodymus.
2. Jei įrenginys neturi išorinio IP adreso, tai nurodymus iš serverio gauna sekančios informacijos siuntimo eigoje.

TR-069 konfigūracija jungimuisi prie serverio yra nesudėtinga, tai yra reikia užpildyti serverio adresą, prievadą, kelią ir prisijungimo duomenis. Maršruto parinktuvo prisijungimą prie serverio inicijuoja pats įrenginys. Tai yra maršruto parinktuvui įsijungus ir turint interneto ryšį įrenginys siunčia informaciją apie savo būseną. Toliau informaciją apie save įrenginys į serverį siunčia periodiškai pagal sukonfigūruotą laiką.

Serveris su maršruto parinktuvu sesiją sudaro tik tam momentui, kai įrenginys siunčia informaciją į serverį, vėliau sesija nutraukiama. Informacijos siuntimas į serverį, per HTTP protokolą pavaizduotas 1.1 pav. TR-069 sesijos pavyzdys Paveikslėlyje matyti kaip maršruto parinktuvas sudaro sesiją su serveriu. Kai sesija sudaroma ir naudojamas HTTPS protokolas pirmiausiai vyksta SSL raktų apsikeitimas, jei naudojamas HTTP protokolas, šis žingsnis praleidžiamas. Toliau serveris užsiprašo informacijos iš maršruto parinktuvo, atsakydamas įrenginys gražina visą reikiamą informaciją. Jei serveryje buvo pakeista kažkokia maršruto parinktuvo parametro reikšmė, toliau serveris siunčia pranešimą kokį parametrą pakeisti ir naujo parametro reikšmę. Maršruto parinktuvas atsako apie pakeistos reikšmės statusą. Serveris nusiunčia tuščią pranešimą jeigu nebėra užduočių ir maršruto parinktuvas nutraukia sesiją.



1.1 pav. TR-069 sesijos pavyzdys

TR-069 protokolas turi nemažai galimybių ir funkcijų. Bet šis protokolas turi didelį trūkumą, kai maršruto parinktuvas neturi išorinio IP adreso arba nėra tame pačiame tinkle kaip serveris. Tokiu atveju serveris negali tiesiogiai siųsti užduočių į maršruto parinktuvą, o turi laukti kol maršruto parinktuvas pagal sukonfigūruota laiką praneš apie savo statusą.

TR-069 metodo trūkumai:

- Iš anksto apibrėžtos komandos. Tai yra iš anksto žinomos komandos, kokia informacija galima gauti iš įrenginio ir kokią informaciją galima nusiųsti į įrenginį.
- Naujų komandų siuntimas. Turi būti atnaujinta ar perrašyta programinė įranga maršruto parinktuve norint atlikti pakeitimus.
- Negalima pasiekti įrenginio interneto sąsajos.
- Negalima pasiekti įrenginio komandinės eilutės.
- Pavėluotas informacijos pranešimas. Tai yra dažniausiai serveriai priima informaciją iš gaminių kas tris minutes. Tai apie atsijungusį įrenginį sužinosime gana vėlai.

- Galime nesužinoti, kad gaminys buvo praradęs interneto ryšį.
- Prijungus daug maršruto parinktuvų prie vieno serverio, net to nenorėdami galime padaryti DDOS ataką į savo serverį. Tai yra serveris nevaldo informacijos paėmimo iš maršruto parinktuvo.

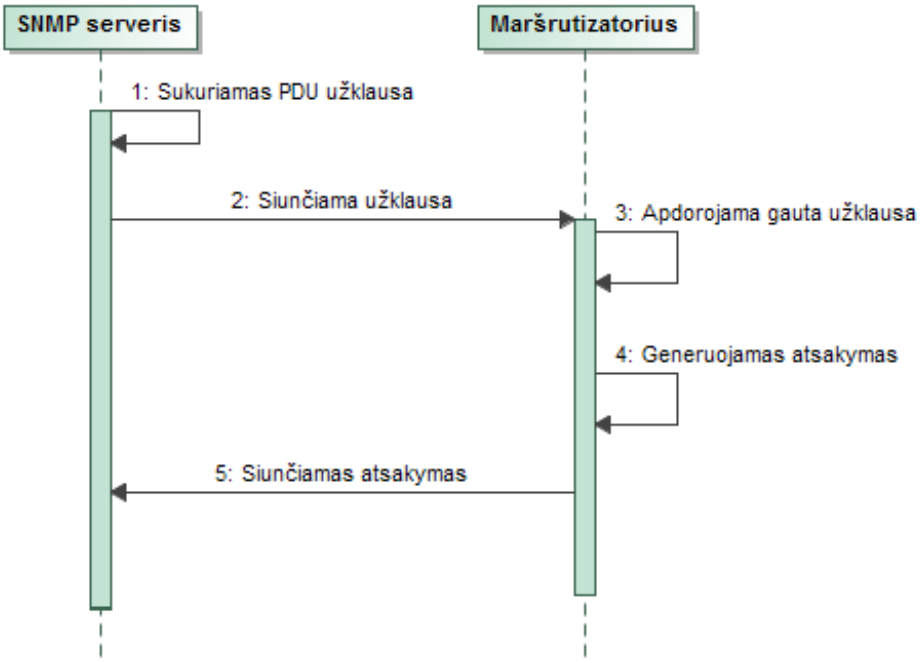
1.5.2. SNMPv3

SNMP – vienas iš tinklo protokolų, skirtų maršruto parinktuvams tinkle stebėti ir valdyti [4]. Vertimas pažodžiui – „Paprastas Tinklo Stebėjimo Protokolas“. Priešingai, nei teigia protokolo pavadinimas, jis yra bene sudėtingiausias iš standartizuotų protokolų, turintis kelis abstrakcijos lygmenis. SNMP veikia ne tik TCP/IP tinkluose, skirtingai nuo daugumos dabar naudojamų.

SNMP protokolas yra populiarus tarp įmonių, kurioms reikia suvaldyti vidinių maršruto parinktuvų tinklą. Taip pat dažnai naudojamas tarp ugdymo įstaigų. Šis protokolas yra tinkamas kai maršruto parinktuvai tiesiogiai pasiekiami iš valdymo stoties ir yra viename tinkle su serveriu. SNMP protokolas naudojamas ne tik gauti informacijai iš maršruto parinktuvo, bet ir jį valdyti. Tiesa, valdymo galimybė atsirado tik nuo trečiosios SNMP versijos.

SNMP protokole [5] nuo v3 versijos atsiradusi valdymo galimybės, bei SSL sertifikatų naudojimo pasirinkimas. Iki v3 versijos buvo tik informacijos gavimo galimybė iš maršruto parinktuvo. Norint gauti maršruto parinktuvo būseną arba atlikti pakeitimus, serveris tiesiogiai siunčia užklausą į maršruto parinktuvą.

Duomenų gavimo per SNMP protokolą pavyzdys pateiktas 1.2 pav. SNMP informacijos gavimo procesas



1.2 pav. SNMP informacijos gavimo procesas.

1.5.3. JSON-RPC

JSON-RPC protokolas sudarytas iš dviejų dalių, tai yra JSON protokolo ir RPC. JSON [6] yra standartizuotas duomenų apsikeitimo formatas. Šis formatas yra lengvai suprantamas vartotojams, ir gana plačiai paplitęs tarp programuotojų nes suderinamas su įvairiomis programavimo kalbomis. RPC - nuotolinių užduočių vykdymui skirtas protokolas [7]. Dažnai naudojamas tarp serverio ir kliento. Apjungti šie du protokolai sudaro vieną JSON-RPC [8] protokolą, kuris yra lengvai suprantamas ir vykdomas nepriklausomai nuo pasirinktos programavimo kalbos.

JSON-RPC protokolo pavyzdys pavaizduotas 1.3 pav. JSON-RPC užklausos siuntimas Paveikslėlyje pateiktas JSON-RPC formato pavyzdys, su užklausos siuntimu į maršruto parinktuvą. Pats protokolas susideda iš objektų ir masyvų, kas leidžia vykdyti kelias komandas iš karto ir komandai perduoti kelis parametrus. Pateiktos komandos pavyzdys, kai serveris maršruto parinktuvo užsiklausia dviejų parametrų, prie kokio operatoriaus maršruto parinktuvas prisijungęs ir koks signalo stiprumas.

```
{
        "jsonrpc": "2.0",
        "id": 1,
        "method": "call",
        "params": ["001e421344eb@05744102", "file", "exec",
        {
                "command":"gsmctl",
                "params":[„-o“, „-q“]
        }]
}
```

1.3 pav. JSON-RPC užklausos siuntimas

Pateiktame paveikslėlyje 1.4 pav. JSON-RPC atsakymo gavimas iš maršrutizatoriaus. yra pateiktas gautas atsakymas iš maršruto parinktuvo. Kadangi JSON formatas yra lengvai suprantamas ir skaitomas, tai nesunku suprasti anksčiau vykdytos komandos rezultatą. Tokiu pat principu galima vykdyti ir kitas komandas maršruto parinktuve ir jos neturi būti iš anksto aprašytos.

```
{
        "jsonrpc":"2.0",
        "id":1,
        "result":[0,
        {
                "code":0,
                "stdout":"-77\nLT BITE GSM\n"
        }]
}
```

1.4 pav. JSON-RPC atsakymo gavimas iš maršrutizatoriaus.

1.6. Protokolų palyginimas

Vykdant ši uždavinį lyginami egzistuojantys protokolai su klientų norima sistema. Lyginant metodus, kaip vienas iš svarbių kriterijų buvo, kad įrenginį prie sistemos būtų galima prijungti kuo paprasčiau. Tai yra, kad sistemos administratoriams užtektų pajungti gaminį į tinklą, o tada jau konfigūruoti per serverį. Protokolų palyginimas pavaizduotas 1-1 Lentelė. Protokolų palyginimų lentelė

Kitas svarbus kriterijus buvo tai, kad sistema turėtų galimybę pati užklausti informacijos apie maršruto parinktuvo būseną. Toks sprendimas buvo pasirinktas tam, kad serveris galėtų valdyti savo apkrovas. Jei maršruto parinktuvai patys pranešinėja apie savo būseną, tai gaunasi, kaip serverio DDOS ataka. Vienu metu pranešant visiems įrenginiams apie savo būseną, gali neužtekti tinklo pralaidumo arba serveris gali nepajėgti laiku apdoroti visas gautas užklausas. Jei serveris pats užklausia informacijos įrenginių apie jų būseną, tai galima nesunkiai valdyti serverio apkrovimus rikiuojant gaminius į eilę. Atnaujinant vienu metu tiek gaminių, kiek serverio apkrova leidžia tuo laiko momentu.

1-1 Lentelė. Protokolų palyginimų lentelė

Palyginimo kriterijus	**TR-069**	**SNMPv3**	**JSON-RPC**	**Klientų norima sistema**
Pirmasis maršrutizatoriaus prisijungimas prie sistemos	Reikalingas konfigūravimas	Reikalingas konfigūravimas	Nereikalingas	Automatinis arba minimalus konfigūravimas
Maršrutizatoriaus būsena	Maršrutizatorius praneša automatiškai	Reikalingas užsiklausimas	Reikalingas užsiklausimas	Reikalingas užsiklausimas
Išjungto maršrutizatoriaus aptikimas	Pagal konfigūraciją	Pagal konfigūraciją	Pagal konfigūraciją	Pagal konfigūraciją
Duomenų kiekis būsenai pranešti.	Didelis	Vidutinis	Mažas	Kuo mažesnis
Išorinis IP nutolusiam gaminiui pasiekti	Be išorinio IP ribotas funkcionalumas	Būtinas	Būtinas	Nebūtinas
Galimybė vykdyti komandas maršrutizatoriuje	Iš anksto aprašytas maršrutizatoriaus programinėje įrangoje.	Iš anksto aprašytas maršrutizatoriaus programinėje įrangoje.	Galimos visos komandos.	Galimos visos komandos.
Galimybė pasiekti maršrutizatoriaus web sąsaja.	Nėra	Nėra	Nėra	Yra
Galimybė pasiekti maršrutizatoriaus terminalą	Nėra	Nėra	Nėra	Yra
Šifravimo galimybė	HTTPS	SSL	HTTPS	Būtinas šifravimas

Atlikus protokolų analizę, nebuvo nei vieno protokolo, kuris atitiktų klientų keliamus reikalavimus. Kadangi klientai norėjo, kad maršruto parinktuvas galėtų būti tiesiogiai pasiekiamas, neturint išorinio IP adreso įrenginyje. Labiausiai kliento norus atitiko JSON-RPC protokolas, dėl galimybės vykdyti maršruto parinktuve komandas iš serverio, kurios iš anksto nebuvo aprašytos sistemoje.

Tam, kad išspręsti išorinio IP adreso problemą buvo nuspręsta panaudoti OpenVPN protokolą. Jo pagalba gaminiai būtų tiesiogiai pasiekiami iš serverio. OpenVPN tunelis [9] yra šifruotas, tad nebereikėtų naudoti JSON-RPC su HTTPS. Taip pat atliktas ir palyginimas tarp JSON-RPC su HTTPS ir JSON-RPC be HTTPS per OpenVPN tunelį, kuris pavaizduotas 1-2 lentelė.

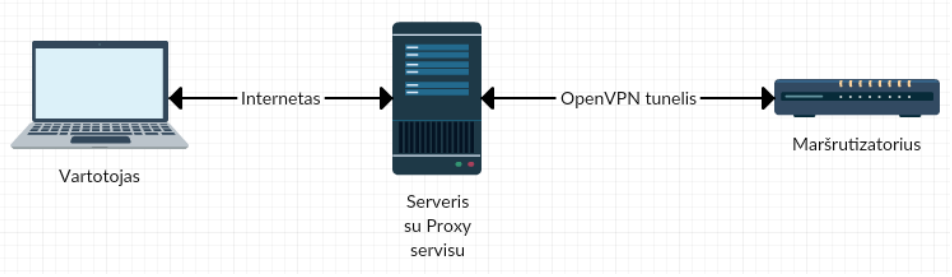
1-2 lentelė. JSON-RPC palyginimas

Palyginimo kriterijus	**JSON-RPC su HTTPS**	**JSON-RPC be HTTPS per OpenVPN**	**Klientų norima sistema**
Reikalingas papildomas konfigūravimas dėl saugumo	Ne	Taip	Kuo mažiau ir paprasčiau konfigūruoti
Maršrutizatoriaus pasiekimas iš serverio	Galima jei maršrutizatorius turi išorini IP	Galima ir be gaminio išorinio IP	Gaminio pasiekimas be išorinio IP

Naudojant JSON-RPC su HTTPS, maršruto parinktuve nereikalingas joks papildomas konfigūravimas. Tokia sistema turi didelį pranašumą, nes norint valdyti kaip pavyzdys 100 gaminių tai nesudėtingai galima daryti, nes nereikia papildomai perdavinėti jokių sertifikatų ar panašiai.

Naudojant JSON-RPC be HTTPS per OpenVPN tunelį, maršruto parinktuve reikalingas papildomas konfigūravimas. Norint sujungti maršrutizatorių su valdymo serveriu, į maršruto parinktuvą reikia suvesti serverio duomenis, įkelti sertifikatus ir kt. Norint valdyti kaip pavyzdys 100 gaminių tai kainuoja papildomai resursų, nes kiekvieną įrenginį reikia konfigūruoti.

Iki pilno klientų norimo funkcionalumo reikalingas maršruto parinktuvo interneto sąsajos pasiekimas, šiam tikslui bus panaudotas „Proxy“ serveris, sistemos modelis pavaizduotas 1.5 pav. „Proxy“ sistemos modelis



1.5 pav. „Proxy“ sistemos modelis

1.7. Protokolų palyginimo išvados

Atlikus protokolų analizę, nebuvo nei vieno protokolo, kuris atitiktų klientų keliamus reikalavimus. Kadangi klientai norėjo, kad maršruto parinktuvą galėtų tiesiogiai pasiekti iš serverio, šiai problemai išspręsti bus panaudotas OpenVPN tunelis. Naudojant OpenVPN tunelį, maršruto parinktuvas būtų tiesiogiai pasiekiamas per vidinį OpenVPN tinklą iš serverio. Maršruto parinktuvui nebereikalingas išorinis IP adresas. Tinkamiausias protokolas, gaminio valdymui yra JSON-RPC, dėl galimybės vykdyti maršruto parinktuve komandas iš serverio, kurios iš anksto nebuvo aprašytos maršruto parinktuve.

1.8. Sertifikatų perdavimo metodai

Saugaus nuotolinių maršrutizatorių valdymo tinkle, bus naudojamas OpenVPN. Šiam protokolui reikalingi sertifikatai. Atlikta analizė, dėl metodų, kaip sertifikatai perduodami iš sertifikatų generavimo sistemos į įrenginį, kuriame jie reikalingi.

Sertifikatų perdavimo metodai naudojami perduodant sertifikatus iš vienos sistemos į kitą:

- FTP – standartinis failų perdavimo protokolas. Kai kuriose maršrutizatoriuose yra galimybė įkelti sertifikatus, suvedus FTP serverio duomenis ir nurodžius kokius sertifikatus atsisiųsti. Slaptažodžiai ir duomenys tarp serverio ir FTP kliento siunčiami nešifruoti, tad galimas duomenų perėmimas.
- FTPS – standartinis failų perdavimo protokolas, kaip ir prieš tai minėtas, bet kartu naudojamas su SSL. Duomenys tarp maršrutizatoriaus ir serverio yra šifruoti.
- HTTP – bene paprasčiausias ir dažniausiai naudojamas perdavimo metodas, kai yra prisijungiama prie maršrutizatoriaus interneto sąsajos ir sertifikatai sukeliami per naršyklę. Toks sertifikatų įkėlimo metodas yra dažniausiai naudojamas, bet yra nesaugus, nes duomenys tarp maršrutizatoriaus ir vartotojo kompiuterio yra nešifruoti.
- HTTPS – toks pat protokolas kaip ir HTTP tik tiek, kad duomenys tarp kompiuterio ir maršrutizatoriaus yra šifruojami. Dažnai naudojamas tarp IT administratorių, kai atliekami maršrutizatoriaus konfigūravimai ar sertifikatų kėlimai į maršrutizatorių.
- Laikmenos – retai pasitaikantis, bet ganėtinai populiarus tarp industrinių maršrutizatorių. IT administratoriai prijungia laikmeną tiesiai prie maršrutizatoriaus, o konfigūruojant maršrutizatorių pasirenka, kad įkelti sertifikatus iš prijungtos laikmenos.

Kadangi dauguma maršrutizatorių valdymo sistemų naudoja HTTPS, tai tokios sistemos nesusiduria su problema, kaip perduoti sertifikatus į gaminį. Tokios sistemos, kurios paremtos sertifikatais, dažniausias sprendimas yra į maršruto parinktuvą įkelti sertifikatus rankiniu būdu.

Serveryje sugeneruojami sertifikatai maršruto parinktuvams, parsisiunčiami į kompiuterį ir tada įkeliami į maršrutizatorių. Klientų vienas iš pagrindinių reikalavimų buvo kuo paprastesnis maršruto parinktuvų prijungimas į valdymo tinklą. Šiam tikslui reikalingas metodas, kuris automatiškai visa tai atliktų už vartotoją.

Atliekant sertifikatų perdavimo metodų analizę buvo analizuojama variantai, kaip ta tinkamai atlikti. Vienas iš variantų, tai yra kai maršruto parinktuvai gaminami ir į juos su programine įranga taip pat būtų įrašomi ir kliento OpenVPN sertifikatai. Bet šis metodas yra nepatikimas, nes gamybos procese, kai įrašoma programinė įranga į maršruto parinktuvą, jau turi būti žinoma, kuriam klientui bus siunčiami gaminiai. Tai stipriai apsunkina ir išbrangina gamybos procesą, tad šis metodas yra netinkamas.

Antras variantas, kad į gaminius butų įrašomas standartinė programinė įranga. Visi gaminiai pasileidę, susigeneruoja unikalų vartotojo vardą ir slaptažodį pagal įrenginio unikalų numerį ir fizinį tinklo plokštės adresą. Toliau maršruto parinktuvai jungiasi prie autentifikavimo serverio, kur parsisiunčia sertifikatus tolesniam jungimuisi prie kliento OpenVPN serverio. Sertifikatų parsisiuntimui iš serverio reikia užtikrinti, kad ryšys tarp serverio ir maršruto parinktuvo būtų šifruotas. Sertifikatų parsiuntimui būtų galima panaudoti HTTPS protokolą, bet tokiu atveju yra galimybė, kad bus apsimesta, kitu įrenginiu. Tad serveriui reikia papildomai patikrinti, kad maršrutų parinktuvas, kuris jungiasi prie autentifikavimo serverio yra tikras ir neapsimeta kitu įrenginiu. Tad norint užtikrinti, kad gaminys tikras, maršruto parinktuvui prisijungus prie serverio ir sudarius OpenVPN tunelį, serveris turi atlikti papildomus patikrinimus. Kai maršruto parinktuvas prisijungus prie serverio, serveris siunčia papildomas užklausas į įrenginį per JSON-RPC protokolą, nuskaito įrenginio vidinės atminties duomenis, taip užtikrinant, kad maršruto parinktuvas yra tikras.

Išanalizavus įvairius variantus prieita prie išvados, kad maršruto parinktuvui įsijungus jis jungiasi su standartiniai OpenVPN sertifikatais prie serverio. Maršruto parinktuvo papildomam patikrinimui serveris prisijungia prie gaminio, patikrina papildomus maršruto parinktuvo duomenis. Jei visi duomenys teisingi autentifikavimo serveris, sugeneruoja sertifikatus maršruto parinktuvui. Sugeneruotus sertifikatus serveris ir naują konfigūraciją perduoda į įrenginį šifruotu OpenVPN tuneliu, taip užtikrinant, kad niekas daugiau nenuskaitys duomenų. Gavus sertifikatus maršruto parinktuvas jungiasi prie kliento saugaus nuotolinio valdymo tinklo, iš kur visi gaminiai yra valdomi ir konfigūruojami.

2. SPRENDIMAS

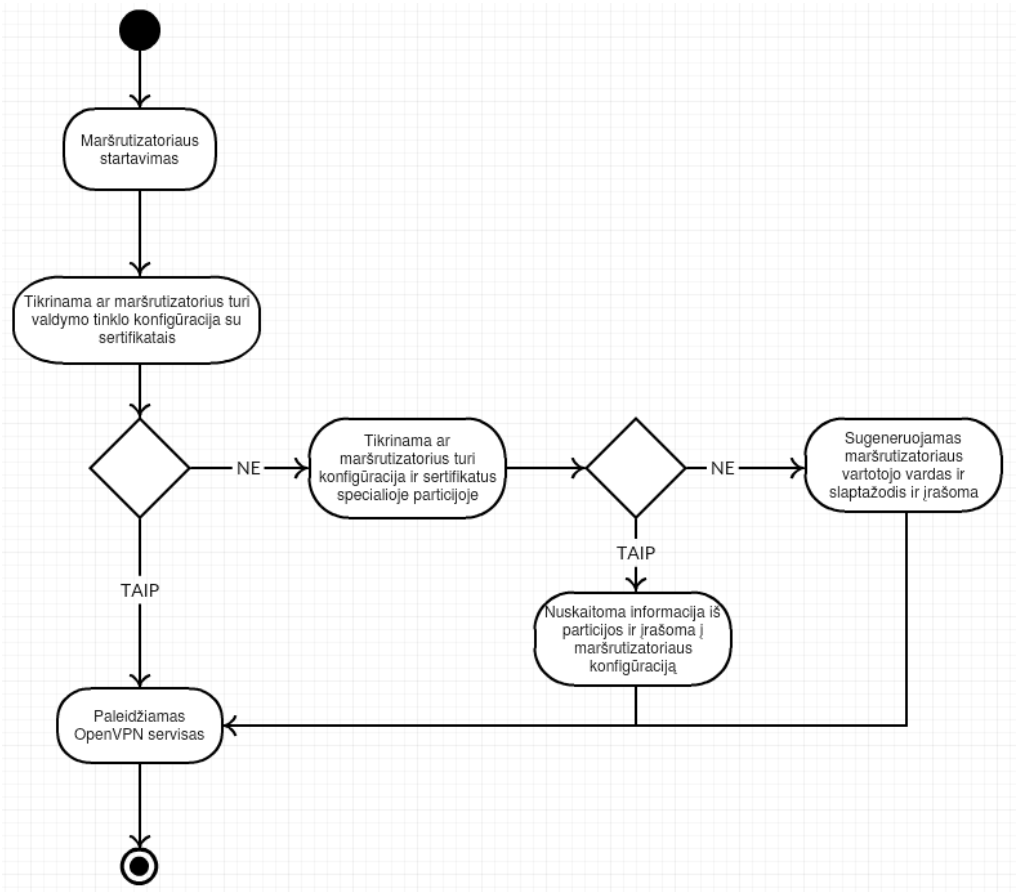
Sprendimo įgyvendinimo etape turime pilnai suprojektuoti ir sumodeliuoti, kaip turi veikti saugus nuotolinių maršrutizatorių valdymo tinklo modelis. Pagal anksčiau atliktos analizės išvadas bus projektuojamas saugus nuotolinių maršrutizatorių valdymo tinklo modelis. Saugų maršruto parinktuvų valdymo tinklo modelį sudarys kelios pagrindinės dalys:

- Maršruto parinktuvas
- Autentifikavimo servisas
- Valdymo tinklo servisas

2.1. Maršruto parinktuvo sistema

Pagal anksčiau atliktą analizę buvo išsiaiškinta, kad maršrutizatoriaus pajungimas prie saugaus valdymo tinklo turi būti kuo paprastesnis, ir kuo mažiau trunkantis procesas. Norint padaryti, kad maršrutizatorius atliktų veiksmus automatiškai, yra reikalingi maršrutizatoriaus programinės įrangos pakeitimai. Į maršrutizatoriaus programinę įrangą buvo įdėta OpenVPN konfigūracija, kartu su sertifikatais.

Maršruto parinktuvo veikimo diagrama pavaizduota 2.1 pav. Maršruto parinktuvo veikimo diagrama



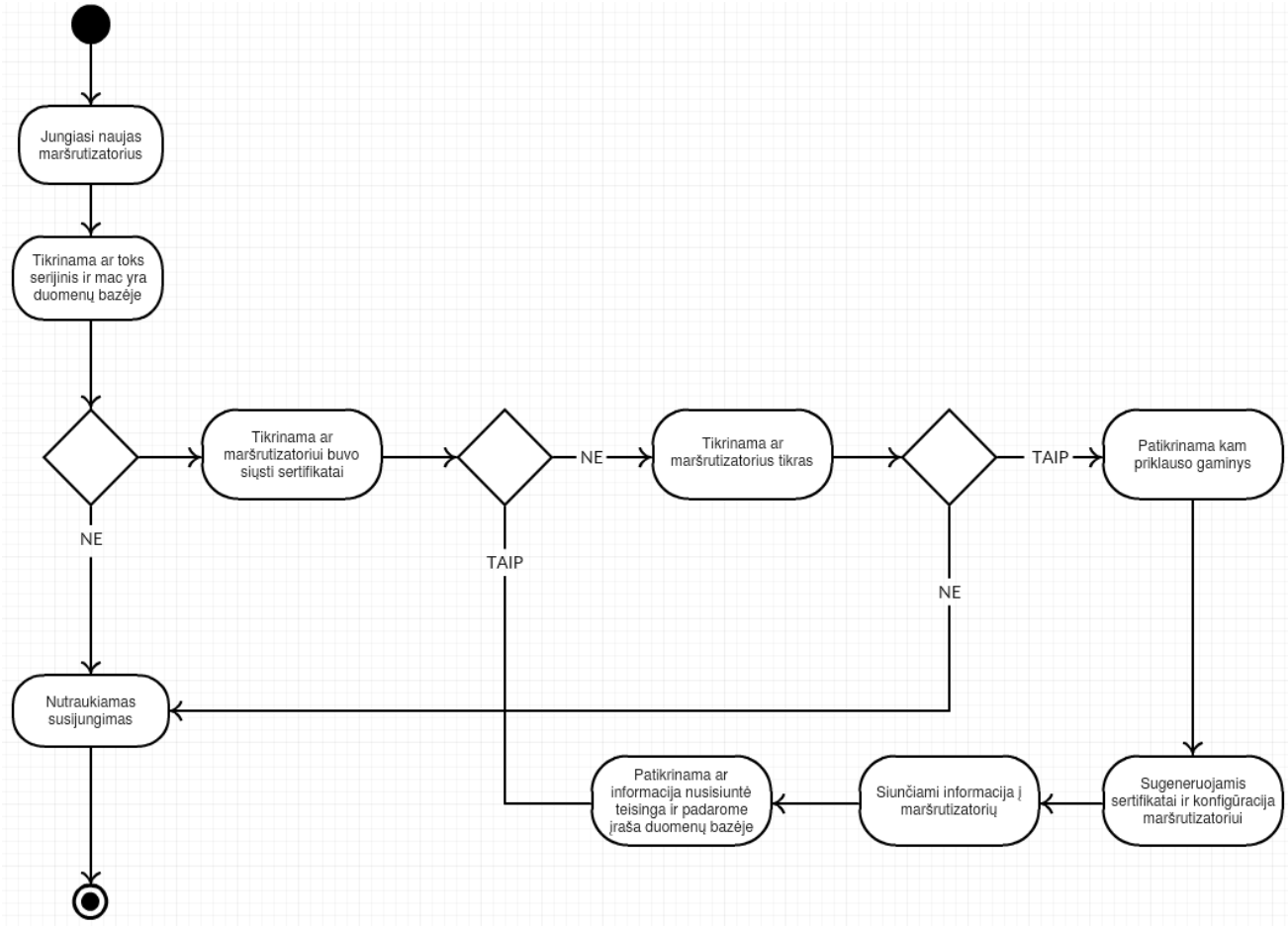
2.1 pav. Maršruto parinktuvo veikimo diagrama

Diagramoje matyti, kad maršruto parinktuvui įsijungus yra tikrinama ar įrenginys turi nuotolinio valdymo tinklo konfigūraciją ir sertifikatus. Jei maršruto parinktuvas reikalingą konfigūraciją turi, paleidžiamas OpenVPN servisas. Jei konfigūracijos nėra, tikrinama maršruto parinktuvo vidinė atminties speciali vieta, kurioje duomenys nėra perrašomi po programinės įrangos perrašymo ar maršruto parinktuvo atstatymo į gamyklinius parametrus. Jeigu konfigūracija yra specialioje atminties dalyje, tai konfigūracija ir sertifikatai nuskaitomi ir įrašomi į maršruto parinktuvo konfigūraciją. Jei specialioje atminties vietoje konfigūracijos nėra, vadinasi maršruto parinktuvas įsijungė pirmą kartą arba dar nebuvo prisijungęs prie autentifikavimo serverio. Tokiu atveju sugeneruojamas prisijungimo vardas ir slaptažodis pagal maršruto parinktuvo unikalų numerį ir fizinį tinklo plokštės adresą. Atlikus visus tikrinimus ir konfigūravimus, paskutiniame žingsnyje yra paleidžiamas OpenVPN servisas.

Sertifikatų saugojimas specialioje vidinės atminties dalyje naudotas tam, kad išspręsti problemą, kai įrenginiui yra perrašoma programinė įranga ar maršruto parinktuvas atstatomas į gamyklinius parametrus. Jei nebūtų išspręsta ši problema reikėtų padaryti, kad maršrutizatorius į autentifikavimo serverį jungtųsi ne vieną kartą. Tad reikėtų siųsti tuos pačius sertifikatus kelis kartus, kas yra nesaugu arba būtų apkraunamas autentifikavimo serveris, dėl naujų sertifikatų generavimo.

2.2. Autentifikavimo servisas

Autentifikavimo servisas – sistemos dalis kurioje turi vykti sertifikatų, kurie bus naudojami jungtis prie saugaus valdymo tinklo, generavimas maršruto parinktuvams. Autentifikavimo serviso veikimo diagrama pavaizduota 2.2 pav. Autentifikavimo serviso veikimo diagrama

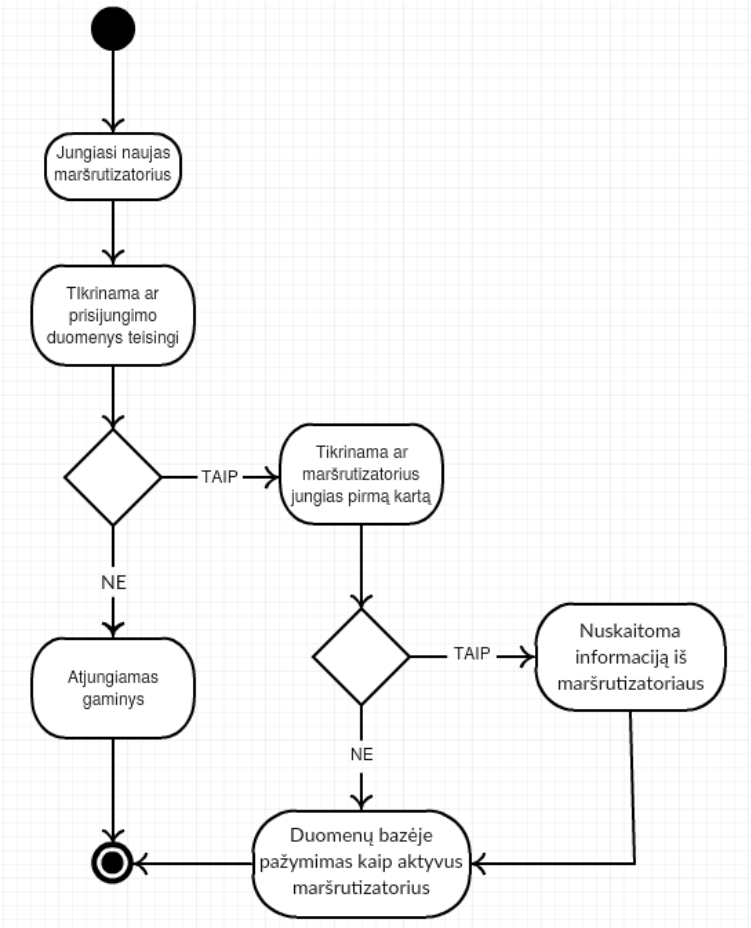


2.2 pav. Autentifikavimo serviso veikimo diagrama

Prie autentifikavimo serviso jungiantis naujam maršruto parinktuvui, įvyksta įrenginio patikrinimas ar maršruto parinktuvo unikalus numeris ir fizinės tinklo plokštės adresas yra duomenų bazėje. Jei toks maršruto parinktuvas duomenų bazėje neegzistuoja, jis nedelsiant atjungiamas nuo serverio. Jei toks maršruto parinktuvas yra, tada tikrinama ar toks įrenginys jau buvo prisijungęs ir ar jam yra sugeneruoti ir išsiųsti sertifikatai su konfigūracija. Toks sistemos tikrinimas turi būti padarytas tam, kad sistema antrą kartą negeneruotų ir neišsiųstų sertifikatų, taip padidinamas sistemos saugumas. Jei gaminys naujas, tada serveris siunčia įvairias užklausas maršruto parinktuvui, kad įsitikinti ar įrenginys tikras ir prisistato tuo kuom iš tikrųjų ir yra. Jei maršruto parinktuvas tikras, tada vyksta patikrinimas, kuriai įmonei jis priklauso, tam kad būtų sugeneruojami būtent tos įmonės OpenVPN sertifikatai maršruto parinktuvui. Sugeneruoti sertifikatai ir konfigūracija siunčiami į maršruto parinktuvą ir taip pat sertifikatai įrašomi į vidinę atminties vietą maršrutų parinktuve. Toliau patikrinama ar informacija nusiųsta į maršruto parinktuvą nesugadinta ir teisinga, jei viskas gerai, duomenų bazėje padaromas įrašas apie naują maršruto parinktuvą ir pažymima, kad būtent tam įrenginiui jau yra sugeneruoti ir nusiųsti sertifikatai.

2.3. Saugus valdymo tinklo servisas

Saugaus valdymo tinklo servisas – sistemos dalis, kurioje maršruto parinktuvai yra sujungti į vieną saugų OpenVPN tinklą. Prie valdymo tinklo jungiasi jau patikrinti maršrutų parinktuvai ir su sugeneruotais sertifikatais. Saugaus valdymo tinklo serviso veikimo diagrama pavaizduota 2.3 pav. Saugaus valdymo tinklo serviso



2.3 pav. Saugaus valdymo tinklo serviso

Jungiantis gaminiui prie sistemos, sistema patikrina prisijungimo duomenis, patikrina sertifikatus ir ar sertifikatai galiojantys ir kt. Jei duomenys klaidingi, serveris maršruto parinktuvui neleidžia prisijungti ir atmeta prisijungimo užklausas. Jei duomenys teisingi, tikrinama ar gaminys prie valdymo tinklo prisijungė pirmą kartą. Jei maršruto parinktuvas prie valdymo tinklo prisijungė nebe pirmą kartą, tai toks gaminys duomenų bazėje pažymimas kaip aktyvus ir toliau paliekamas susijungimas su maršruto parinktuvu. Jei įrenginys prie sistemos prisijungė pirmą kartą, tai prieš pažymint gaminį kaip aktyvų iš maršrutizatoriaus nuskaitoma papildoma informacija, tokia kaip techninės įrangos versija, gaminio kodas ir kita panaši informacija, kuri nekintanti, arba kintanti retai.

2.4. Papildomi servisai

Papildomi servisai reikalingi pilnam sistemos funkcionalumui, kurio pageidavo klientai, tai butų „Proxy“ serveris. Jis yra reikalingas tam, kad vartotojas galėtų pasiekti maršruto parinktuvo interneto sąsają. Tiesiogiai klientai negali pasiekti maršruto parinktuvo interneto sąsajos, nes dauguma įrenginių neturi išorinio IP adreso arba nėra tame pačiame tinkle su kliento kompiuteriu. Tam tikslui bus į serverį įdiegiamas „Proxy“ servisas, per kurį klientai pasieks maršruto parinktuvo interneto sąsają.

2.5. OpenVPN konfigūracija

Saugiame valdymo tinkle yra naudojami dviejų tipų OpenVPN tuneliai. Tai yra autentifikavimo serveris su OpenVPN tuneliu, kuriame naudojamas vartotojo vardo ir slaptažodis prisijungimui. Antrasis OpenVPN tunelis yra kliento saugaus valdymo tinklo prie kurio jungiasi maršruto parinktuvai.

2.5.1. OpenVPN konfigūracija autentifikavimo servise

OpenVPN serverio konfigūracija yra nustatyta, kad maršruto parinktuvai susijungimui su serveriu naudoja vartotojo vardą ir slaptažodį, kurie prisijungimo metu yra perduodami į programą ir tikrinami duomenų bazėje. Programa gavusi duomenis iš OpenVPN serviso, patikrina ar prisijungimo duomenys atitinka formatą, jei viskas gerai, toliau duomenys yra tikrinami duomenų bazėje. Patikrinus duomenis, programa į OpenVPN servisą gražina ar prijungti tokį įrenginį, ar ne. Jei visi duomenys buvo teisingi ir įrenginys buvo rastas duomenų bazėje, tai su tokiu maršruto parinktuvu sudaromas susijungimas per OpenVPN tunelį. Maršruto parinktuvui prisijungus prie autentifikavimo OpenVPN serviso, toliau yra kviečiama programa, kurioje yra tikrinamas įrenginys ir generuojami sertifikatai maršruto parinktuvui.

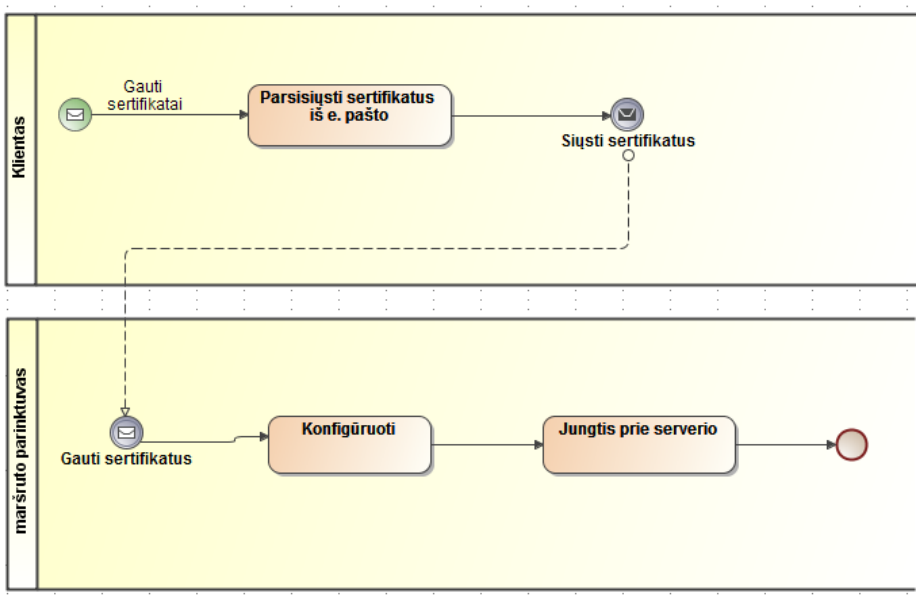
2.5.2. OpenVPN konfigūracija valdymo tinkle

OpenVPN saugaus maršruto parinktuvų tinklo konfigūracijoje yra naudojami sertifikatai. OpenVPN tuneliui šifruoti yra naudojamas asimetrinis RSA šifravimas. RSA rakto ilgis naudojamas tuneliui yra 2048 bitų ilgio. Toks rakto ilgis pasirinktas dėl optimalaus duomenų kiekio ir laiko sunaudojimo, tarp serverio ir maršruto parinktuvo. Generuojami Diffie-Hellman parametrai, gali trukti iki kelių minučių, nes pasirinkto rakto dydis yra 2048 bitų, todėl šių parametrų generavimas yra padaromas iš anksto, kai į sistemą yra pridedama nauja įmonė. Visam vienos įmonės valdymo tinklui yra naudojamas tas pats Diffie-Hellman. Taip pat atsižvelgiant į įmonės poreikius šis parametras kas kiek laiko bus pergeneruojamas iš naujo. Kadangi pasirinktas RSA rakto ilgis yra 2048 bitų ilgio,

raktas yra skaitomas saugiu, ir jį nulaužti šiuolaikinėmis technologijomis kainuotų daugiau, nei būtų gauta naudos. Beje sistemoje yra numatyta, kad bus sertifikatų pergeneravimo algoritmas, kuris prasiuntus tam tikrą duomenų kiekį arba buvus aktyviu tam tikrą laiką maršruto parinktuvui sugeneruos naujus sertifikatus ir perduos į maršruto parinktuvą. Toks naujų sertifikatų generavimas būtų galimas kaip pavyzdys, kas metus.

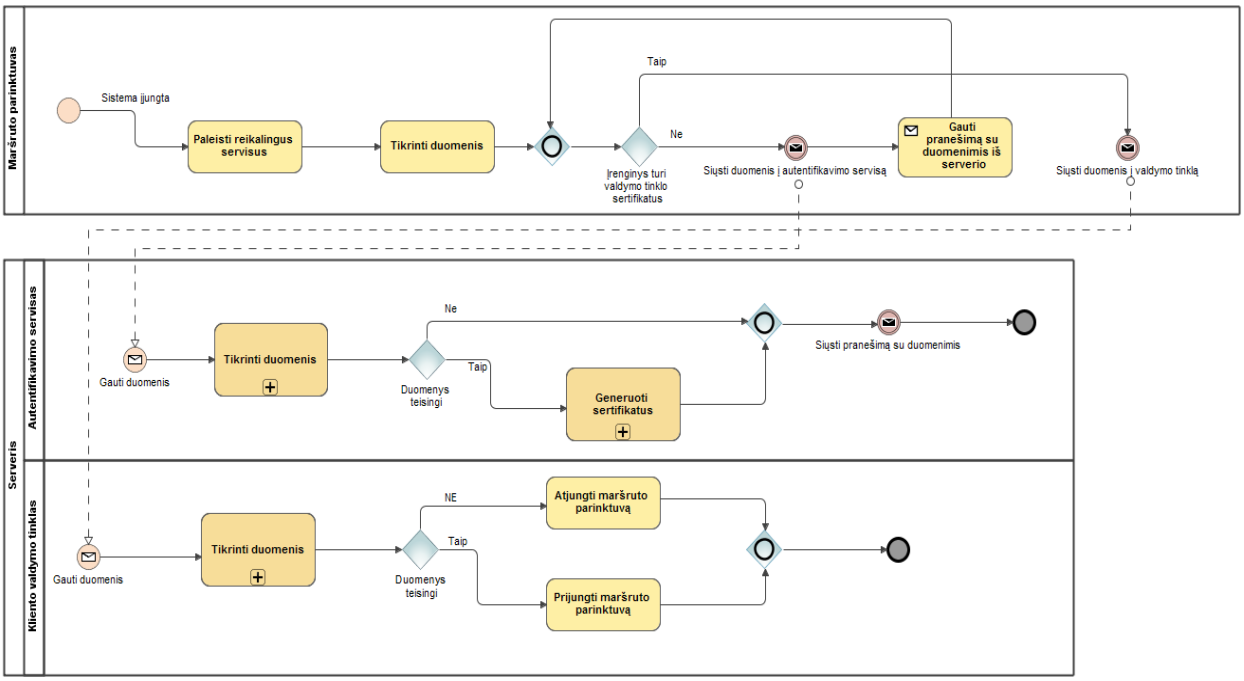
2.6. Sistemos modelis

Dabar naudojamas maršruto parinktuvų prijungimui prie kliento valdymo tinklo naudojamas modelis pavaizduotas 2.4 pav. Esamas sistemos modelis. Jame pavaizduota, kaip vartotojai rankiniu būtų konfigūruoja maršruto parinktuvą, ir įkelia sertifikatu. Prieš konfigūruojant maršruto parinktuvą jungimuisi prie saugaus kliento valdymo tinklo, sistemos administratorius atsiunčia sertifikatus į kliento elektroninį paštą. Šiame modelyje dar nėra pavaizduotas administratoriaus darbas kurį turi atlikti administratorius norit sugeneruoti sertifikatus ir išsiųsti klientui.



2.4 pav. Esamas sistemos modelis

Įdiegus naująją sistemą būtų atliekamas automatinis maršruto parinktuvo prisijungimas prie saugaus valdymo tinklo. Nebereikėtų administratoriaus darbo, kuris turi sugeneruoti sertifikatus maršruto parinktuvams ir juos perduoti klientams. Naujoji sistema visa tai atliks automatiškai. Naujasis sistemos modelis pavaizduotas 2.5 pav. Kuriamos sistemos veikimo modelis Prieš jungiantis prie saugaus valdymo tinklo, pirmiausiai būtų sudaromas pirminis OpenVPN tunelis tarp maršruto parinktuvo ir sertifikatų generavimo serverio. Prisijungus prie serverio, būtų pirmiausiai patikrinamas maršruto parinktuvas, ir jei įrenginys tinkamas, jam būtų sugeneruojami sertifikatai ir konfigūracija, kurie vėliau per OpenVPN tunelį nusiunčiami į maršruto parinktuvą. Saugiai perduoti sertifikatus tarp serverio ir maršruto parinktuvo yra sudaromas OpenVPN tunelis, kad duomenys būtų šifruoti.



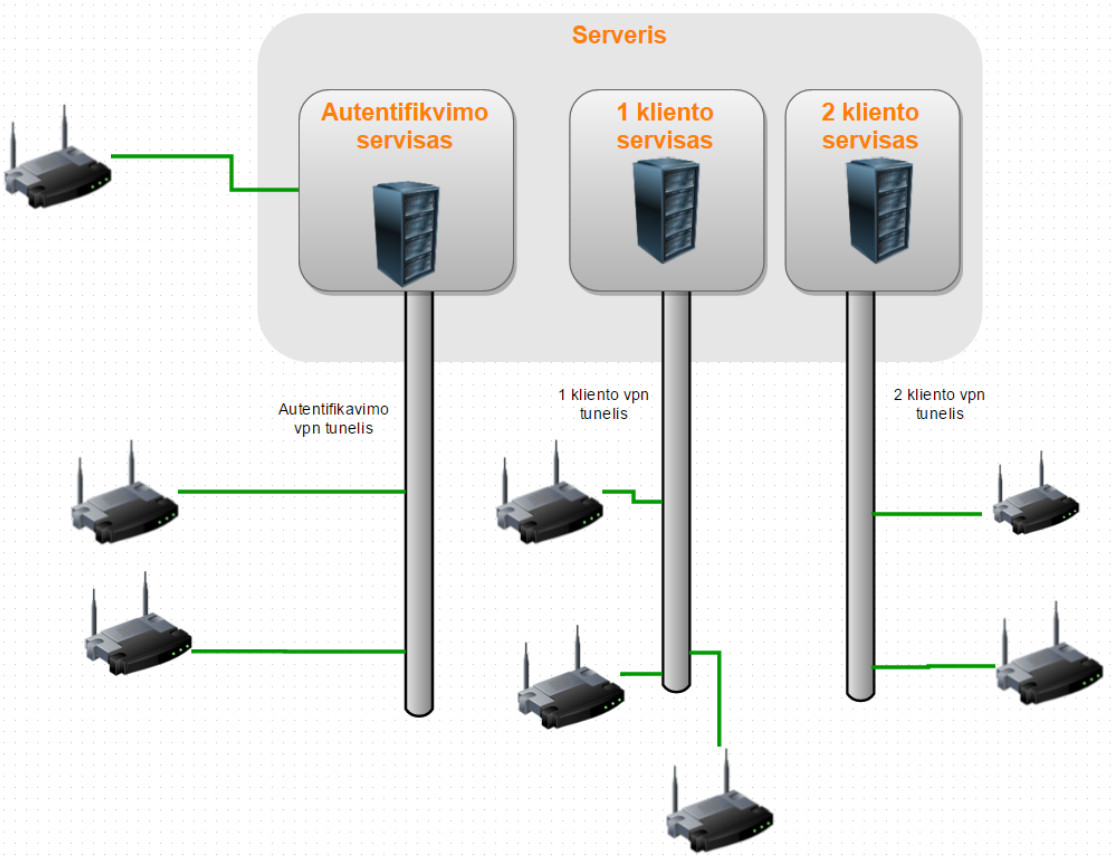
2.5 pav. Kuriamos sistemos veikimo modelis

Susijungus maršruto parinktuvui su serveriu per OpenVPN tunelį, OpenVPN servisas praneša sertifikatų generavimo aplikacijai apie naujai prisijungusį įrenginį. Aplikacija apdoroja gautus duomenis iš OpenVPN serviso ir duoda užklausą į sertifikatų generavimo programą. Sertifikatų generavimo programa gautas užduotis rikiuoja į eilę, ir jas vykdo eilės tvarka. Užduotys programoje vykdomos atsižvelgiant į serverio apkrovas. Jei serverio apkrova nedidelė ir serveris turi laisvų resursų, užduotys vykdomos ne po vieną, o kelias iš karto. Tai yra vienu metu gali būti generuojami sertifikatai keliems maršruto parinktuvams iš karto. Jei serveris apkrautas ir laisvų resursų nėra tai užduotys laukia eilėje, kol serveryje atlaisvės resursų, arba programa baigs anksčiau pradėta užduotį. Programai apdorojant gautą užduotį pirmiausiai prisijungiama prie maršruto parinktuvo ir nuskaitomi duomenys iš gaminio vidinės atminties. Taip patikrinamas ar gaminys tikras ir nebandoma apsimesti kitu gaminiu. Atlikus patikrinimą ir aptikus, kad gaminys netinkamas, jis tuoj pat atjungiamas nuo OpenVPN serverio ir duomenų bazėje pažymima, kad būtent šito gaminio nebeprileisti per OpenVPN tunelį prie autentifikavimo serviso. Jei maršruto parinktuvas tinkamas toliau atliekami sertifikatų generavimai. Po kiekvienos atliktos užduoties duomenų bazėje pažymimas įrašas apie atliktus veiksmus. Į duomenų bazę įrašai daromi tam, kad nutrūkus ryšiui tarp serverio ir įrenginio, nebūtų atliekamas tas pats darbas antrą kartą. Tai pat užtikrinama, kad tie patys sertifikatai nebus siunčiami du ar daugiau kartų. Viską sugeneravus ir nusiuntus į maršruto parinktuvą, įrenginys atjungiamas nuo serverio, ir toliau gaminys jungiasi prie kliento saugaus valdymo tinklo per OpenVPN tunelį.

3. REALIZACIJA

Realizuota sisteminė maršruto parinktuvo susijungimo dalis su saugiu nuotolinių maršrutizatorių valdymo tinklu. Realizacijoje buvo išspręsta problema, kaip saugiai perduoti sertifikatus maršruto parinktuvui, kai norima prie sistemos prijungti didelį kiekį įrenginių. Realizacijoje buvo įgyvendintas OpenVPN automatinis sertifikatų generavimas ir perdavimas į maršruto parinktuvą. Sertifikatai yra generuojami priklausomai nuo to, kuriai įmonei maršruto parinktuvas priklauso. Realizacijos metu įgyvendinta sistema užtikrina, kad maršruto parinktuvas saugiai susijungia su kliento OpenVPN tuneliu, be vartotojo.

Realizuota sistema pavaizduota 3.1 pav. Realizuota sistema, kuriame matyti, kad viename fiziniame serveryje veikia keli OpenVPN tuneliai. Pirmas OpenVPN tunelis yra autentifikavimo servisas, naudojamas maršruto parinktuvams, kurie neturi sertifikatų ir jungiasi pirmą kartą. Kiti OpenVPN tuneliai skirti klientų valdymo tinklams, prie kurių jungiasi maršruto parinktuvai turėdami sertifikatus.



3.1 pav. Realizuota sistema

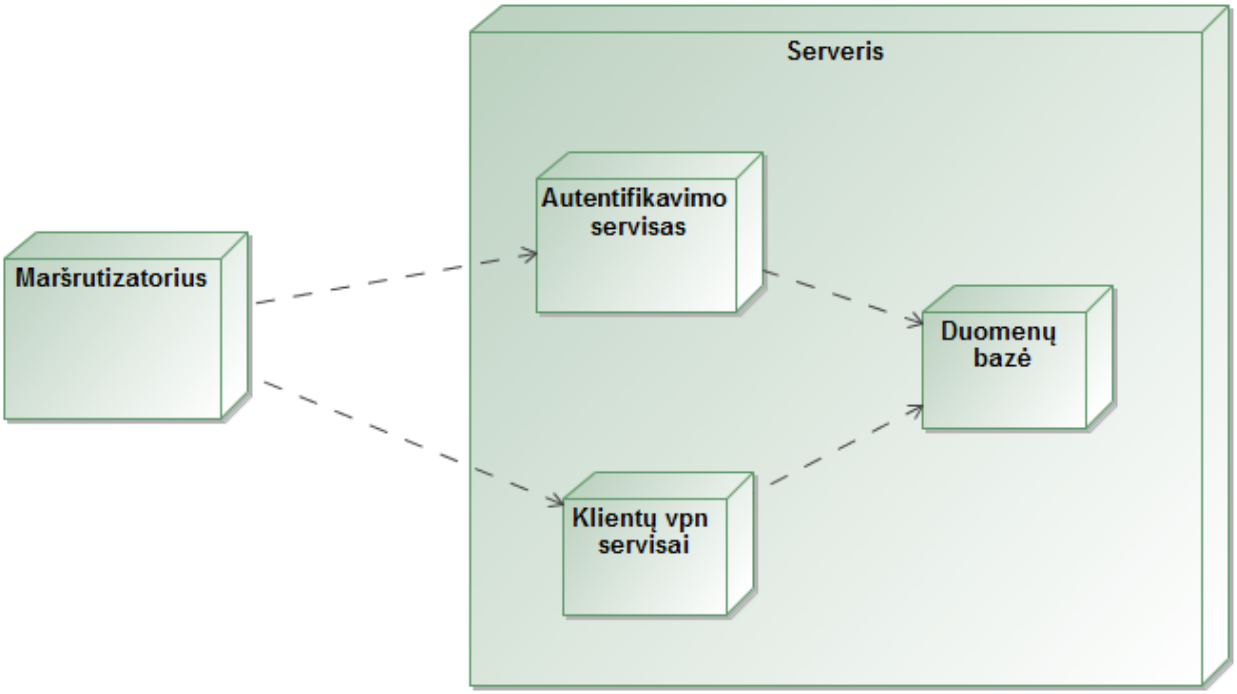
Prie kliento valdymo tinklo jungiasi, tik tam klientui priklausantys maršruto parinktuvai. Atskirų klientų maršrutų parinktuvai jungiasi prie atskirų valdymo tinklų, jie yra atskirti tam, kad nebūtų jokių galimybių pasiekti kito kliento įrenginių. Šis sprendimas buvo priimtas tam, kad esant poreikiui,

kliento valdymo tinklą galima sukonfigūruoti taip, kad kiekvieną maršruto parinktuvą galėtų pasiekti iš kito maršruto parinktuvo, esančio tame pačiame valdymo tinkle.

Realizuota sistema yra ganėtinai lanksti. Esant poreikiui arba dėl didelio maršruto parinktuvų kiekio, kliento OpenVPN tunelį galima iškelti į kitą fizinį serverį. Tarp sertifikatų generavimo serverio ir kliento serverio reikalingas tik OpenVPN tunelis, kuriuo bus galima perduoti sertifikatus iš vieno serverio į kitą.

3.1. Realizuotos sistemos komponentinis modelis

Sistemoje išskirtos dvi fizinės dalys, maršrutizatorius ir serveris, kurios pavaizduotos 3.2 pav. Sistemos komponentinis modelis



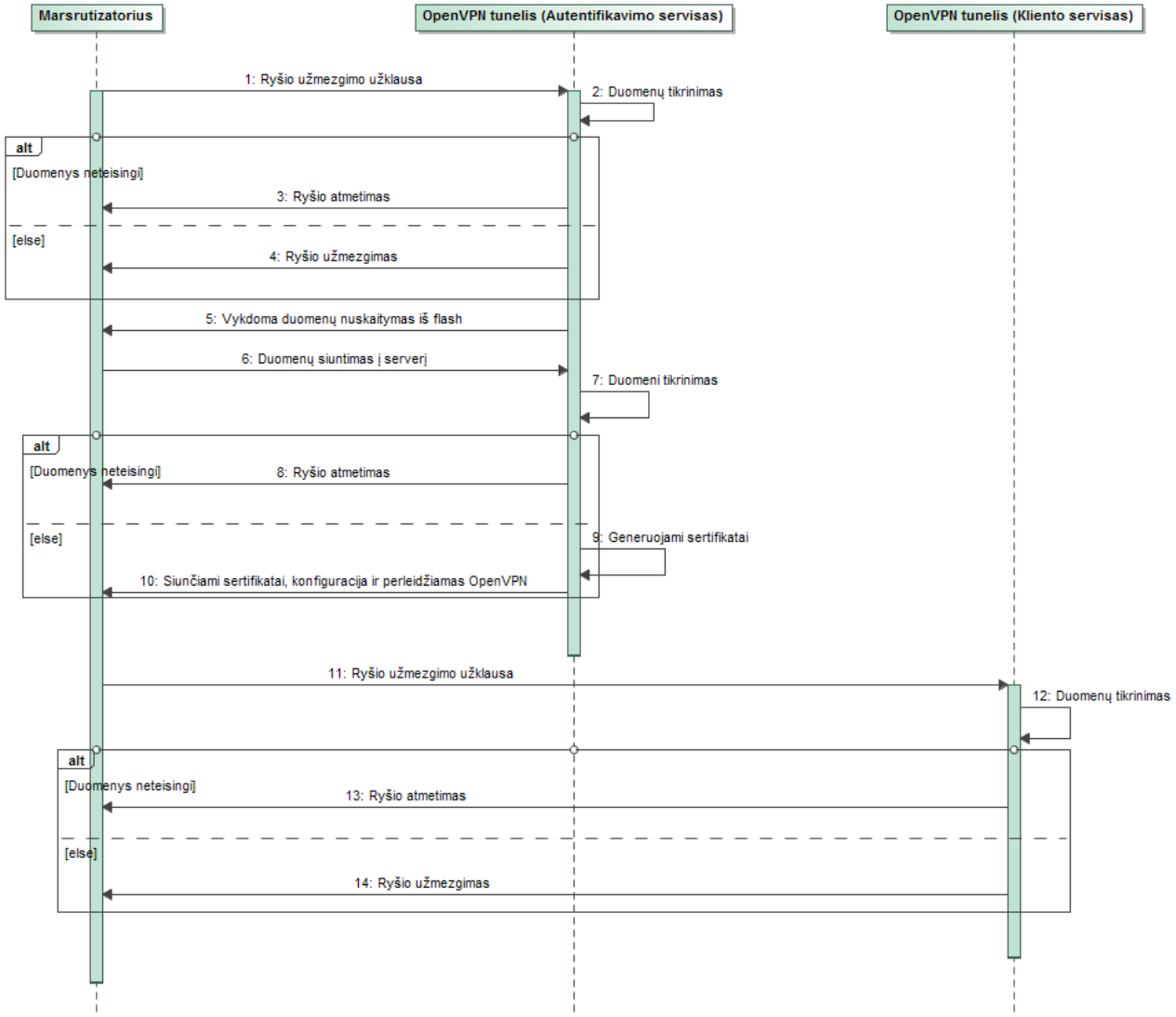
3.2 pav. Sistemos komponentinis modelis

Serveris sudarytas dar iš atskirų dalių:

- Pirmoji dalis – duomenų bazė, kurioje yra duomenys apie maršruto parinktuvus.
- Antroji dalis – OpenVPN tunelis, autentifikavimo servisas, prie kurio maršruto parinktuvas jungiasi dar neturėdamas kliento konfigūracijos.
- Trečioji dalis – klientų tuneliai, prie kurių maršrutizatoriai jungiasi, kai turi reikalingus sertifikatus ir konfigūraciją.

3.2. Realizuotos sistemos veikimo principas

Sistemos veikimo principas pavaizduotas 3.3 pav. Sistemos veikimo principas Modelyje pavaizduotos pagrindinės sistemos dalys ir tai kas vyksta iki tol, kol maršruto parinktuvai susijungia su kliento valdymo serveriu. Modelyje pavaizduotos tik pagrindinės dalys ir veiksmai, kurie nėra smulkiau aprašyti. Kaip pavyzdys, nėra aprašyta, kaip vyksta duomenų patikrinimas, nes tai yra ganėtinai aiškūs veiksmai.



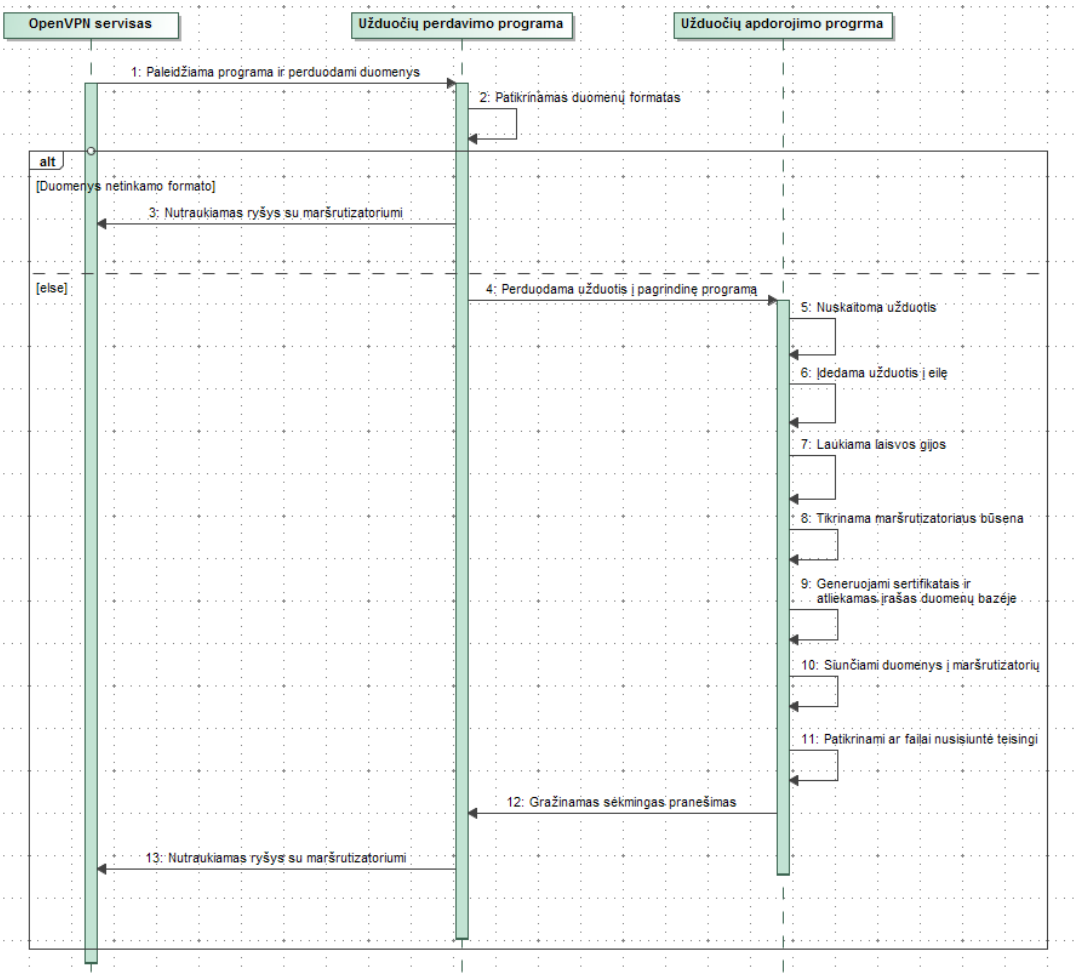
3.3 pav. Sistemos veikimo principas

Maršruto parinktuvo susijungimas su kliento OpenVPN tuneliu, pirmiausiai prasideda maršruto parinktuvui kreipiantis į autentifikavimo servisą. Prisijungimo metu maršruto parinktuvas jungdamasis prie serverio perduoda savo prisijungimo duomenis tai yra vartotojo vardas ir slaptažodis. Kad prie OpenVPN tunelio neprisijungtų bet kas, tai vartotojo vardas ir slaptažodis yra naudojamas maršruto parinktuvo unikalus numeris ir fizinės tinklo plokštės adresas, kurie yra unikalūs. Atėjus tokiai užklausai į serverį, duomenys patikrinami duomenų bazėje. Jei toks įrenginys yra duomenų bazėje ir

įrenginys yra priskirtas įmonei ir tam įrenginiui dar nėra sugeneruoti sertifikatai, ryšys su maršruto parinktuvu sudaromas. Kad sistema įsitikintu, jog įrenginys yra tikras ir nebandoma apeiti sistemos, serveris nuskaito tam tikrus duomenis iš maršruto parinktuvo vidinės atminties. Nuskaityti duomenys sulyginami su pateiktais duomenimis ir duomenų baze. Jei visi duomenys teisingi maršruto parinktuvui sugeneruojami konfigūraciniai failai, sertifikatai ir pasirašomi tos įmonės, kuriai maršruto parinktuvas priklauso. Sertifikatai ir nauja konfigūracija nusiunčiama į maršruto parinktuvą, patikrinami failų kontrolinės sumos. Jei viskas gerai, perleidžiamas OpenVPN servisas maršruto parinktuve. Maršruto parinktuvas su nauja konfigūracija ir sertifikatais jungiasi prie kliento OpenVPN tunelio, tai yra jau prie klientui priklausančio saugaus valdymo serverio. Sėkmingai susijungusius įrenginius sistemoje galima matyti kaip aktyvius, kuriuos galima konfigūruoti ir stebėti.

3.3. Autentifikavimo serviso programa

Autentifikavimo programos veiklos diagrama pavaizduotas 3.4 pav. Autentifikavimo programos



3.4 pav. Autentifikavimo programos veiklos diagrama

Autentifikavimo servise veikianti programa yra kaip procesas, kuris sukasi amžinai. Tai yra padaryta dėl to, kad sistema galėtų valdyti kiek vienu metu yra generuojama sertifikatų. Toks valdymas buvo būtinas, nes pirminėje programos versijoje buvo padaryta klaidų, kai nebuvo padarytas valdymas ir prisijungus keliems maršruto parinktuvams vienu metu, serveris tiek apsikraudavo, kad ilgą laiką nebeatsakinėdavo į užklausas. Taip pat, kad taupyti serverio resursus ir atlikti užduotis kuo greičiau, buvo nuspręsta naudoti C programavimo kalbą, kuri veikia žymiai greičiau, nei kitos programavimo kalbos. Į programą buvo įtraukta SSL biblioteka, kas leido generuoti sertifikatus pačiai programai, ir taip sutaupyti resursų nekviečiant „OpenSSL“ programos per komandinę eilutę. Pagrindinė programa – programa atliekanti darbus, kai gauna užduočių į savo užduočių sąrašą

Autentifikavimo programos veiklos diagramoje pavaizduotos trys pagrindinės dalys:

- OpenVPN servisas,
- Programa, kuri apdoroja užduotis,
- Programa, kuri yra tarpininkas tarp OpenVPN serviso ir užduočių apdorojimo programos.

Prisijungus įrenginiui prie OpenVPN serviso, yra iškviečiamas programa, kuriai OpenVPN servisas perduoda informaciją apie prisijungusį maršruto parinktuvą. Programa yra tiesiog tarpininkas tarp OpenVPN serviso ir pagrindinės programos, kuri gavus informaciją iš OpenVPN patikrina ar duomenys tinkamo formato. Programa suformuoja užduotį ir įrašo į pagrindinės programos „socket“. Pagrindinė programa, kuri ir atlieka visas užduotis pagrinde visą laiką miega, kol nesulaukia signalo, kad į programos „socket“ kažkas buvo įrašyta. Sulaukus signalo, programa nuskaito gautą užduotį, ir informaciją apie maršruto parinktuvą. Toliau jungiamasi prie duomenų bazės ir pasiimama papildoma informacija apie maršruto parinktuvą. Pirmiausiai patikrinama, ar toks maršruto parinktuvas jau buvo prisijungęs ir jei taip kokie veiksmai buvo atlikti su juo. Jei maršruto parinktuvas prisijungė pirmą kartą, įrenginiui priskiriamas kliento OpenVPN tunelio IP adresas, ir tai įrašoma duomenų bazėje. Toliau generuojami konfigūraciniai failai, ir padaromas įrašas duomenų bazėje. Vėliau generuojami sertifikatai ir tai pat apie atliktą veiksmą padaromas įrašas duomenų bazėje. Toliau konfigūracija ir sertifikatai siunčiami į maršruto parinktuvą per autentifikavimo serviso OpenVPN tunelį. Siuntimas per OpenVPN tunelį užtikrina, kad sertifikatai bus saugiai nusiunčiami į maršruto parinktuvą, nes duomenys siunčiami šifruotu tuneliu. Jei viskas nusisiuntė sėkmingai ir kontrolinės sumos sutapo, duomenų bazėje padaromas įrašas apie nusiųstus sertifikatus, kad antrą kartą nebūtų siunčiami. Jei siunčiant sertifikatus nutrūko ryšys tarp serverio ir maršruto parinktuvo, arba kontrolinės sumos nesutapo. Tokie sertifikatai serveryje ištrinami ir paskelbiami negaliojančiais. Jei viskas nusisiuntė sėkmingai, serveris perleidžia maršruto parinktuvo OpenVPN servisą. Po perleisto serviso maršruto parinktuvas jungiasi prie kliento valdymo tinklo per saugų OpenVPN tunelį su savo sertifikatais.

4. TYRIMAS

Tyrimo metu buvo bandoma aptikti serverio trūkumus, kuriais pasinaudojus būtų galima sutrikdyti sistemos darbą. Tyrimo metu naudotas įmonės bandymų serveris, su kuriuo ir buvo atliekami visi tyrimo darbai. Tyrimo metu bus atlikti šie tyrimai:

- Kiek laiko trunka pirmas maršrutizatoriaus prisijungimas prie valdymo tinklo. Tai yra kai vienu metu jungiant skirtingą maršrutizatorių kiekį pirmą kartą prie serverio ir jiems turi būti sugeneruojami sertifikatai.
- Kaip naudojami serverio resursai, kai maršrutizatoriai jungiasi pirmą kartą ir turi būti atliekamas sertifikatų generavimas.
- Kaip serverio apkrovos reaguoja į skirtingą maršrutizatorių prisijungimo kiekį vienu metu po sistemos įsijungimo.
- Kaip serveris reaguoja į DDOS ataką, kai jis yra vykdomas iš išorės.

Tyrimo metu naudota įranga:

- LTE maršrutizatoriai, „RUT950“ su „Linux“ operacine sistema.
- Serveris su „Linux Centos 7“ operacine sistema.

Serverio informacija:

- Procesorius - Intel(R) Pentium(R) 4 CPU 3.06GHz
- Operatyvioji atminti – 2GB
- Tinklo pralaidumas – 100Mbp/s

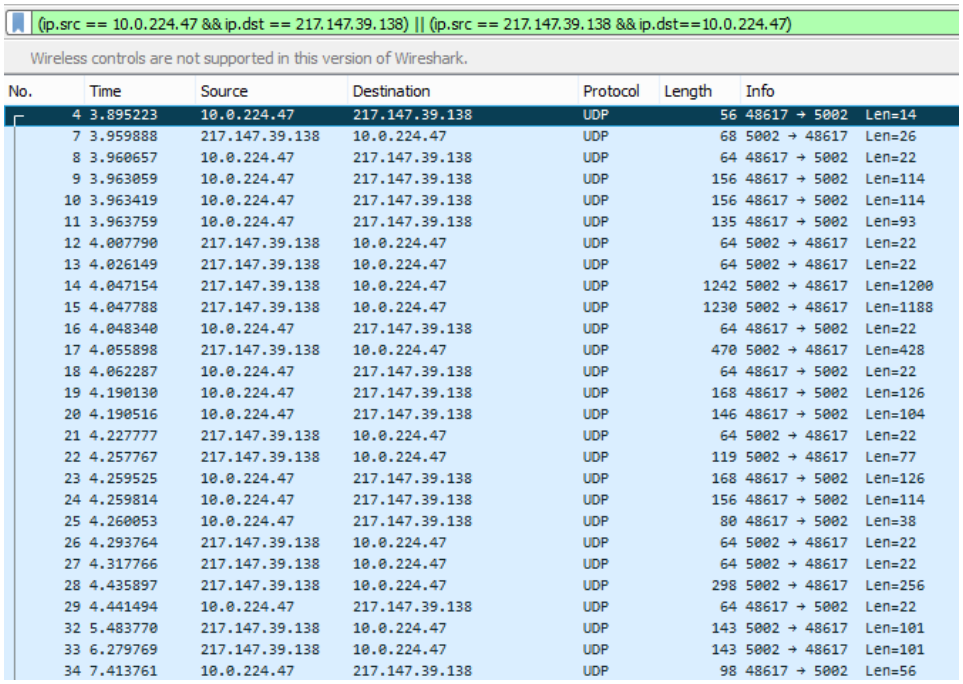
4.1. Maršrutizatorių prisijungimo prie serverio

Atlikti kelių tipų tyrimai, maršruto parinktuvų prisijungimui prie serverio:

- Kiek laiko trunka pirmas prisijungimas prie kliento valdymo tinklo neturint sertifikatų. Tai yra buvo atlikti matavimai kiek laiko trunka maršruto parinktuvui prisijungti prie autentifikavimo serverio, gauti sertifikatus, konfigūraciją ir pradėti jungtis prie kliento valdymo tinklo.
- Kiek laiko trunka maršruto parinktuvui prisijungti nebe pirmą kartą prie kliento valdymo tinklo, kai maršruto parinktuvas pasileidžia.
- Išanalizuoti tinkle siunčiamus paketus ir įsitikinti, kad duomenys kurie siunčiami tinkle yra šifruoti ir toks sertifikatų perdavimo metodas yra saugus. Toks tyrimas įrodytu, kad ir pašaliniams asmenims perimus tinkle siunčiamus duomenis, nebūtų galimybės jų suprasti.

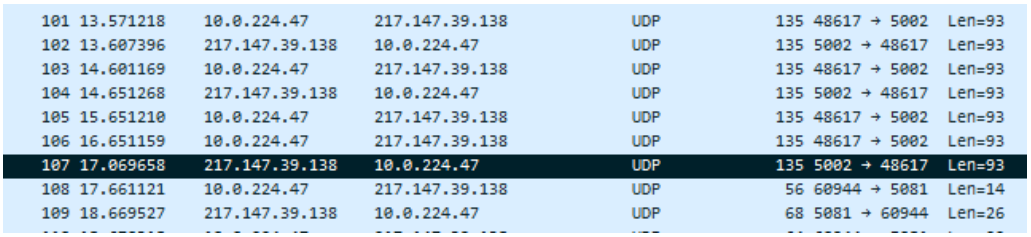
4.1.1. Pirmas maršruto parinktuvo prisijungimas prie saugaus valdymo tinklo

Atliekant tyrimą buvo skanuojamas tinklas su „Wireshark“, buvo renkami paketai ir analizuojami laikai pavyzdys pateiktas 4.1 pav. Surinktų paketų su „Wireshark“



4.1 pav. Surinktų paketų su „Wireshark“ pradžia

Paveikslėlyje matyti iškarpa iš „Wireshark“ surinktų paketų, pažymėtas paketas yra pirma užklausa siunčiama iš maršruto parinktuvo į autentifikavimo serverį. Laikas kurį maršruto parinktuvas užtrukdavo kol gauna sertifikatus ir jungiasi prie valdymo tinklo, buvo skaičiuojamas nuo pirmo paketo siunčiamo iš maršruto parinktuvo į autentifikavimo serverį iki paskutinio paketo, pavaizduotas 4.2 pav. Paskutinis paketas paryškintas paketas. Tai yra paskutinis paketas, nes po to maršruto parinktuvas kreipiasi į tą patį serverį, tik į kitą prievado adresą, tai kreipimasis vyko į saugų kliento valdymo tinklą.



4.2 pav. Paskutinis paketas

Toliau 4-1 Lentelė. Autentifikavimo serverio užduočių vykdymas po vieną. pateikti duomenys, kuriuose matyti kiek laiko trunka maršruto parinktuvui gauti sertifikatus iš autentifikavimo

serverio. Visi bandymai buvo atlikti po 5 kartus. Lentelėje pateikta kiek maršrutizatorių jungdavosi prie serverio vienu metu, kiek vidutiniškai laiko trukdavo aptarnauti visus maršrutizatorius ir įvykdyti visas užduotis ir paskutiniame stulpelyje paskaičiuotas laikas kiek vidutiniškai laiko trukdavo vieno maršrutizatoriaus aptarnavimas sistemoje. Bandymo metu buvo atlikti apribojimai, kad serveris vienu metu gali vykdyti vieną užduotį. Tai yra vienu metu aptarnauja tik vieną maršruto parinktuvą, kiti įrenginiai stovi eilėje ir laukia kol bus atlaisvinti programos resursai.

4-1 Lentelė. Autentifikavimo serverio užduočių vykdymas po vieną.

Maršruto parinktuvų skaičius	Vidutinis laikas visoms užduotims	Vidutinis laikas vienam maršruto parinktuvui
1	~15 s	~15 s
2	~31 s	~16 s
3	~46 s	~15 s
4	~65 s	~16 s
5	~78 s	~16 s
6	~ 96 s	~16 s
7	~115 s	~16 s
8	~134 s	~17 s
9	~154 s	~17 s
10	~173 s	~17 s

Tyrimo metu gauti duomenys, kurie buvo surašyti į lentelę, vėliau pavaizduoti grafiškai. Lentelėje ir grafike yra pavaizduoti vidutiniai laikai, nes tyrimo metu bandymas buvo kartojamas 5 kartus ir į lentelę rašomas tik vidutis laikas. Tai yra 4.3 pav. Serverio užduočių vykdymo laikas matyti, kad vidutinis visų maršruto parinktuvų aptarnavimo laikas serveryje yra tiesinė priklausomybė, kai serveris maršruto parinktuvų užduotis atlieka po vieną ir likę maršruto parinktuvai laukia eilėje. Tai ir yra matyti grafike, kad kuo daugiau maršruto parinktuvų jungiasi į sistemą vienu metu, tuo daugiau laiko trunka aptarnauti ir įvykdyti visų maršruto parinktuvų užduotis. Vidutinis laikas vienam maršrutizatoriui nežymiai didėja, daugėjant maršruto parinktuvų skaičiui, kurie yra programos eilėje ir laukia kol bus atliekamos užduotys.

4.3 pav. Serverio užduočių vykdymo laikas

Taip pat tyrimo metu buvo atliktas bandymas, kad vienu metu serveris atlieka dvi užduotis. Tai yra vienu metu serveris aptarnauja du maršruto parinktuvus, kiti įrenginiai laukia eilėje kol bus atlaisvinti programos resursai ir maršruto parinktuvai bus aptarnauti. Toliau 4-2 Lentelė. Autentifikavimo serverio užduočių vykdymas po dvi. pateikti duomenys, kuriuose matyti kiek laiko trunka maršruto parinktuvui gauti sertifikatus iš autentifikavimo serverio. Visi bandymai buvo atlikti po 5 kartus. Lentelėje pateikta kiek maršrutizatorių jungdavosi prie serverio vienu metu, kiek vidutiniškai laiko trukdavo aptarnauti visus maršrutizatorius ir įvykdyti visas užduotis ir paskutiniame stulpelyje paskaičiuotas laikas kiek vidutiniškai laiko trukdavo vieno maršrutizatoriaus aptarnavimas sistemoje.

4-2 Lentelė. Autentifikavimo serverio užduočių vykdymas po dvi.

Maršruto parinktuvų skaičius	**Vidutinis bandymų laikas**	**Vidutinis laikas vienam maršruto parinktuvui**
1	~15 s	~15 s
2	~16 s	~16 s
3	~33 s	~11 s
4	~33 s	~8 s
5	~49 s	~10 s
6	~50 s	~8 s
7	~66 s	~9 s
8	~66 s	~8 s
9	~83 s	~9 s
10	~83 s	~8 s

Tyrimo metu gauti duomenys, kurie buvo surašyti į lentelę, vėliau pavaizduoti grafiškai. Lentelėje ir grafike yra pavaizduoti vidutiniai laikai, nes tyrimo metu bandymas buvo kartojamas 5 kartus ir į lentelę rašomas tik vidutinis laikas. Tai yra 4.3 pav. Serverio užduočių vykdymo laikas matyti, kad vidutinis visų maršruto parinktuvų aptarnavimo laikas serveryje nėra tiesinė priklausomybė, kai serveris maršruto parinktuvų užduotis atlieka po dvi vienu metu ir likę maršruto parinktuvai laukia eilėje. Kadangi serveris aptarnauja maršruto parinktuvus po du vienu metu, tai grafike matoma, kad serveriui aptarnauti kaip pavyzdys penkis arba šešis maršruto parinktuvus kainuoja tiek pat laiko.

4.4 pav. Serverio užduočių vykdymo laikas, kai užduotys vykdomos po dvi

Šioje dalyje atliktas tyrimas parodė, kad serveriui atliekant kelias užduotis iš karto maršruto parinktuvai yra aptarnaujami greičiau, tai yra serveris gali generuoti sertifikatus keliems maršruto parinktuvams vienu metu. Tokiu būdu yra pagreitinamas maršruto parinktuvų prisijungimas prie serverio, kai maršruto parinktuvai jungiasi pirmą kartą. Tolimesniuose tyrimuose bus atliekamas tyrimas, kaip serverio apkrovai turi įtakos skirtingas maršrutizatorių aptarnavimas vienu metu, kai užduotis atliekamos po vieną ir ne tik.

4.1.2. Maršruto parinktuvo prisijungimas prie valdymo tinklo

Taip pat buvo atliktas tyrimas, kai skirtingas maršruto parinktuvų skaičius vienu metu jungiasi prie saugaus valdymo tinklo nebe pirmą kartą. Tai yra maršruto parinktuvai prie serverio jungiasi jau turėdami sertifikatus. Tyrimo rezultatai pavaizduoti 4-3 Lentelė. Maršruto parinktuvų jungimasis prie

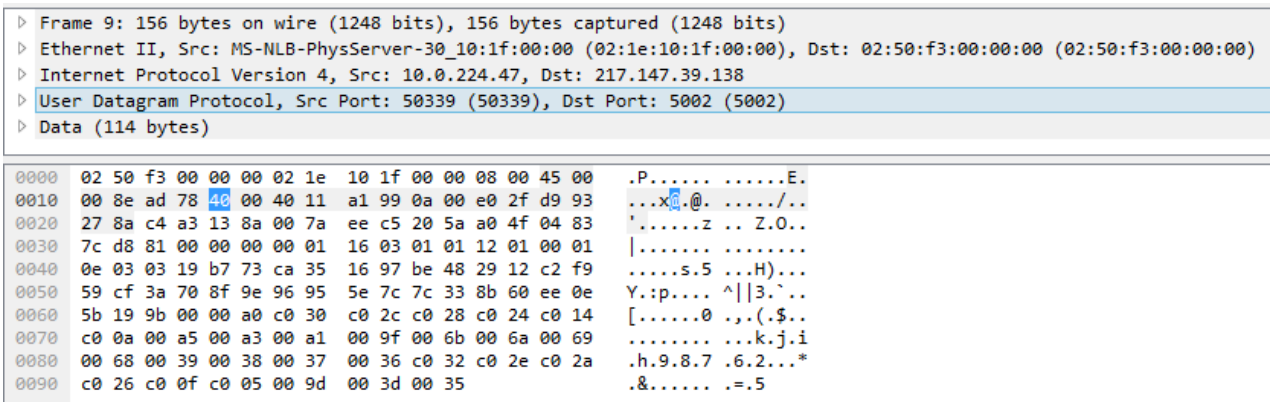
valdymo tinklo Joje matyti, kad maršruto parinktuvų jungimuisi prie saugaus valdymo tinklo neturi įtakos skirtingas maršruto parinktuvų skaičius, tai yra kai vienu metu jungiasi skirtingas maršruto parinktuvų skaičius.

4-3 Lentelė. Maršruto parinktuvų jungimasis prie valdymo tinklo

Maršruto parinktuvų skaičius	Vidutinis bandymų laikas	Vidutinis laikas vienam maršruto parinktuvui
1	~4 s	~4 s
2	~4 s	~4 s
3	~4 s	~4 s
4	~4 s	~4 s
5	~4 s	~4 s
6	~4 s	~4 s
7	~4 s	~4 s
8	~4 s	~4 s
9	~4 s	~4 s
10	~4 s	~4 s

4.1.3. Siunčiamų paketų analizė

Tinkle siunčiamus paketus tarp maršruto parinktuvo ir serverio buvo surenkami su „Wireshark“ įrankiu. Buvo išanalizuoti visi paketai tiek susijungimo, tiek duomenų siuntimo tarp maršruto parinktuvo ir autentifikavimo serverio. Prisijungimo metu siunčiamas vartotojo vardas ir slaptažodis yra taip pat šifruoti, jokie duomenys tarp serverio ir maršruto parinktuvo nesiunčiami atviru tekstu. Vienas iš siunčiamų paketų iš maršruto parinktuvo į serverį pavaizduotas 4.5 pav. Siunčiamas paketas



4.5 pav. Siunčiamas paketas

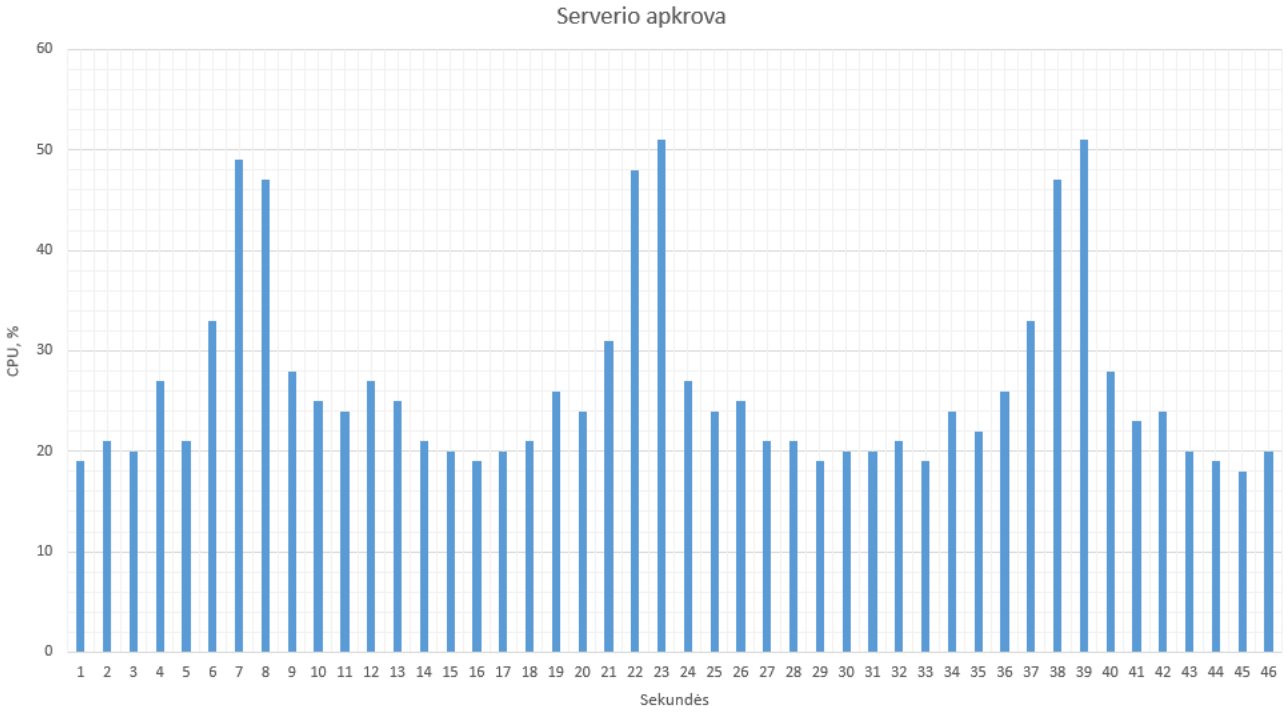
Paveikslėlyje matyti, kad siunčiamo paketo duomenys yra šifruoti. Kad ir perėmus paketą, kuris iš maršrutizatoriaus siunčiamas į autentifikavimo serverį, negalima perskaityti paketo turinio. Tai yra nėra galimybės sužinoti vartotojo vardo ir slaptažodžio.

4.2. Serverio apkrova jungiantis maršrutizatoriui pirmą kartą.

Šioje dalyje bus atliekamas tyrimas, kaip autentifikavimo serverio darbui, turi įtakos skirtingas maršruto parinktuvų kiekis, jungiamas prie serverio, ir generuojami sertifikatai su reikalinga konfigūracija.

4.2.1. Serverio procesoriaus apkrova vykdant po vieną užduotį.

Pirmo tyrimo metu, prie autentifikavimo serverio buvo jungiami trys maršruto parinktuvai vienu metu. Autentifikavimo serverio pagrindinė programa, buvo apribota, kad maršruto parinktuvų užduotis atliktų po vieną, tai yra kiti maršruto parinktuvai stovi eilėje ir laukia kol bus aptarnauti. Jungiant maršruto parinktuvus prie autentifikavimo serverio, buvo stebima autentifikavimo serverio procesoriaus apkrova. 4.6 pav. Serverio procesoriaus apkrova pavaizduotas serverio procesoriaus apkrova procentais, tam tikru laiko momentu.



4.6 pav. Serverio procesoriaus apkrova

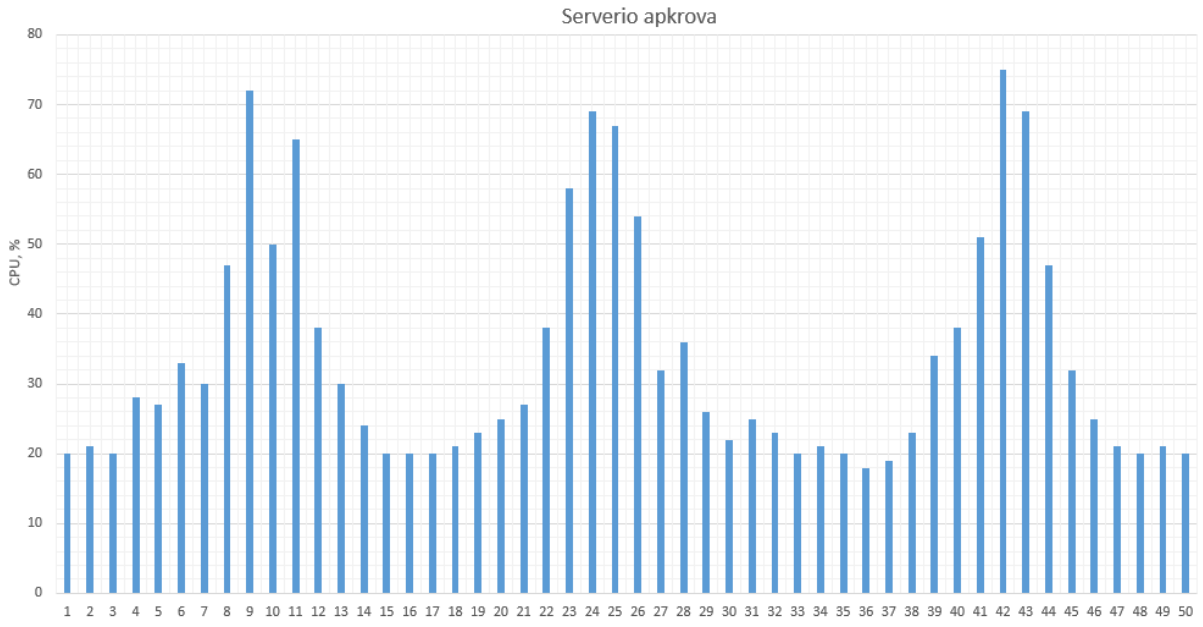
Tyrimo metu prie autentifikavimo serverio buvo jungiami trys maršruto parinktuvai vienu metu. Autentifikavimo serverio vidutinė normali apkrova yra nuo 18 iki 23 procentų. Grafike, nuo

pirmos iki 5 sekundės vyksta maršruto parinktuvų prisijungimas prie autentifikavimo serverio, ir duomenų gavimas iš pirmojo maršruto parinktuvo ir duomenų apdorojimas. Nuo 6 iki 8 sekundės vyksta sertifikatų generavimas maršruto parinktuvui. Grafike matyti, kad procesoriaus apkrova tuo laiko momentu kai generuojami sertifikatai labiausiai padidėja net iki 50 procentų. Nuo 9 iki 15 sekundės vyksta duomenų siuntimas iš autentifikavimo serverio į maršruto parinktuvą, taip pat tuo laiko momentu patikrinami ir nusiųsti duomenys, bei atliekamos papildomos komandos maršruto parinktuve. Toliau grafike matyti likusių dviejų maršruto parinktuvų aptarnavimas. Vaizdas grafike panašus ir toliau, kai vyksta sertifikatų generavimas likusiems maršruto parinktuvams, procesoriaus apkrova padidėja, baigus sertifikatų generavimą apkrova sumažėja.

4.2.2. Serverio procesoriaus apkrova vykdant po dvi užduotis.

Autentifikavimo serverio pagrindinėje programoje buvo atliktas apribojimas, kad vienu metu serveris aptarnautų po du maršruto parinktuvus vienu metu. Tai yra bandymo metu buvo jungiami šeši maršruto parinktuvai vienu metu, kurie buvo aptarnaujami po du. Serverio vidutinė normali apkrova yra nuo 18 iki 23 procentų. Grafike 4.7 pav. Serverio procesoriaus apkrova pavaizduota autentifikavimo serverio procesoriaus apkrova, kai serveris aptarnauja po du maršruto parinktuvus vienu metu.

.



4.7 pav. Serverio procesoriaus apkrova

Nuo pirmos iki 7 sekundės vyko maršrutizatorių prisijungimas prie serverio, ir dviejų maršruto parinktuvų duomenų nuskaitymas ir apdorojimas. Nuo 8 iki 11 sekundės vyko sertifikatų generavimas, tuo laiko momentu matyti didžiausia procesoriaus apkrova serveryje. Nuo 12 iki 16 sekundės vyko duomenų persiuntimas į maršruto parinktuvus, duomenų sutikrinimas ir papildomų

komandų įvykdymas. Toliau grafikas sistemingai kartojasi, nes baigus pirmąsias užduotis serveris pradeda aptarnauti sekančius du maršruto parinktuvus.

4.2.3. Serverio procesoriaus apkrovų išvados.

Tyrimo metu paaiškėjo, kad maršruto parinktuvui generuoti sertifikatus serveryje nėra daug kainuojantis procesas serveriui. Tai pat prieita prie išvados, kad vienu metu galima aptarnauti bent po du maršruto parinktuvus vienu metu serveryje. Taip pat didėjant maršruto parinktuvų skaičiui, kuriems generuojami sertifikatai, taupant jų prisijungimo laiką, būtų galima iš pagrindinės programos, pastoviai stebėti sistemos procesoriaus apkrovas. Serveris galėtų ne tik vienu metu aptarnauti po du maršruto parinktuvus, bet ir užduotis atliekant lygiagrečiai, tai yra kol apdorojami maršruto parinktuvo duomenys kitas procesas generuoja maršruto parinktuvui sertifikatus. Atlikus tokius pakeitimus programoje tas leistų bent jau padvigubinti, o gal net ir daugiau kartų padidinti maršruto parinktuvų aptarnavimų skaičių vienu metu.

4.3. Serverio apkrovų tyrimas po serverio įsijungimo.

Stebint serverio apkrovas po serverio pasileidimo buvo pastebėta, kad pasileidžiant serverio programoms ir klientų valdymo tinklams su OpenVPN, serverio pradžioje stipriai pakyla serverio apkrova. Tai pat serverio apkrovai turi įtakos vienu metu jungiantis didelis kiekis maršruto parinktuvų prie klientų valdymo tinklų. Po tokių pastebėjimų buvo prieita prie išvados, kad autentifikavimo serveris būtų paleidžiamas praėjus dviems minutėms po serverio pasileidimo. Tai yra per toki laiko momentą pasileidžia visos reikalingos serverio programos, visi klientų valdymo tinklai su maršruto parinktuvais. Tai pat stabilizuojasi serverio procesoriaus apkrova, ir tiktai praėjus dviems minutėms paleidžiamas serverio autentifikavimo servisas ir atidaromas prievadas, suteikiant galimybę naujiems maršruto parinktuvams jungtis prie autentifikavimo serviso.

4.4. Serverio DDOS atakos tyrimas

Serverio atakos tyrime bus ištirta, kaip serveris reaguoja į DDOS atakas. Tai yra tyrimo metu, bus analizuojama, kaip DDOS ataką turės įtakos maršruto parinktuvų prijungimui prie sistemos, kaip reaguos jau esami maršruto parinktuvai į vykstančią ataką.

Atliekant DDOS ataką prieš valdymo tinklo serverį, buvo panaudoti du vienodi kompiuteriai, kurių parametrai:

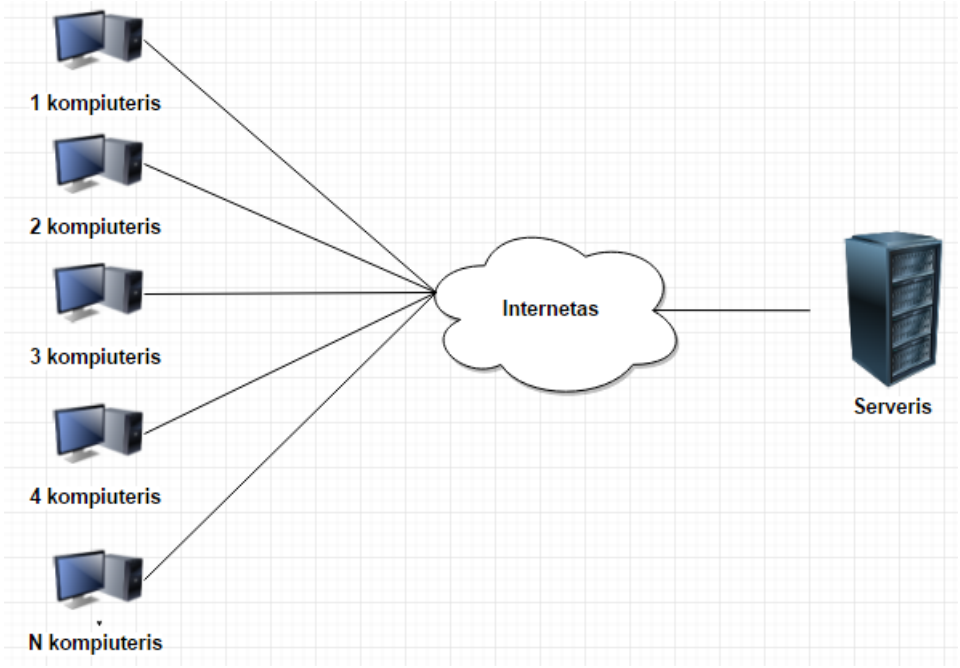
- Procesorius Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz
- Operatyvioji atmintis – 8GB

- Tinklo pralaidumas – 100Mbp/s
- Operacinė sistema – Linux

DDOS atakos simuliavimui buvo panauda „PING“ komanda su papildomais parametrais. Pilna komanda: „sudo ping -c 1000000 -s 65400 -l 10 -i 0.00001 -W 0.1 –q 217.147.39.138“. Komandos parametrai:

- „-c“ – parametras aprašantis kiek užklausų išsiųsti,
- „-s“ – parametras, kuriuo nurodomas siunčiamo paketo dydis baitais,
- „-l“ – parametras, kuriuo nurodomas, kiek maksimaliai paketų išsiųsti kol nesulaukiamas atsakymas. Naudojant šį parametrą su didesne reikšme nei 3, reikia vykdyti komanda „root“ vartotoju,
- „-i“ – parametras, kuriuo nurodoma, kas kiek sekundžių siųsti paketus,
- „-W“ – parametras, kuriuo nurodoma, kiek laiko laukiama atsakymo iš serverio sekundėmis,
- „-q“ – parametras, kuris nurodo, kad nebūtų spausdinamas komandos išvesties tekstas, kas leidžia pagreitinti programos veikimą.

Toks DDOS atakos tipas vadinamas „Ping of death“ [10] atakos modelis pavaizduotas 4.8 pav. DDOS atakAtakuojantys kompiuteriai generuoja didelį kiekį užklausų į serverį, su padidintu duomenų kiekiu. Tai yra iš vieno kompiuterio, vykdant vieną komandą yra išsiunčiama apie 140 paketų per vieną sekundę, kur vieno paketo dydis yra 65,4 kilobaito. Tai sugeneruojamas bendras srautas yra apytiksliai apie 69.6Mbp/s.



4.8 pav. DDOS atakos modelis

4.4.1. Serverio pasiekiamumas, vykstant DDOS atakai

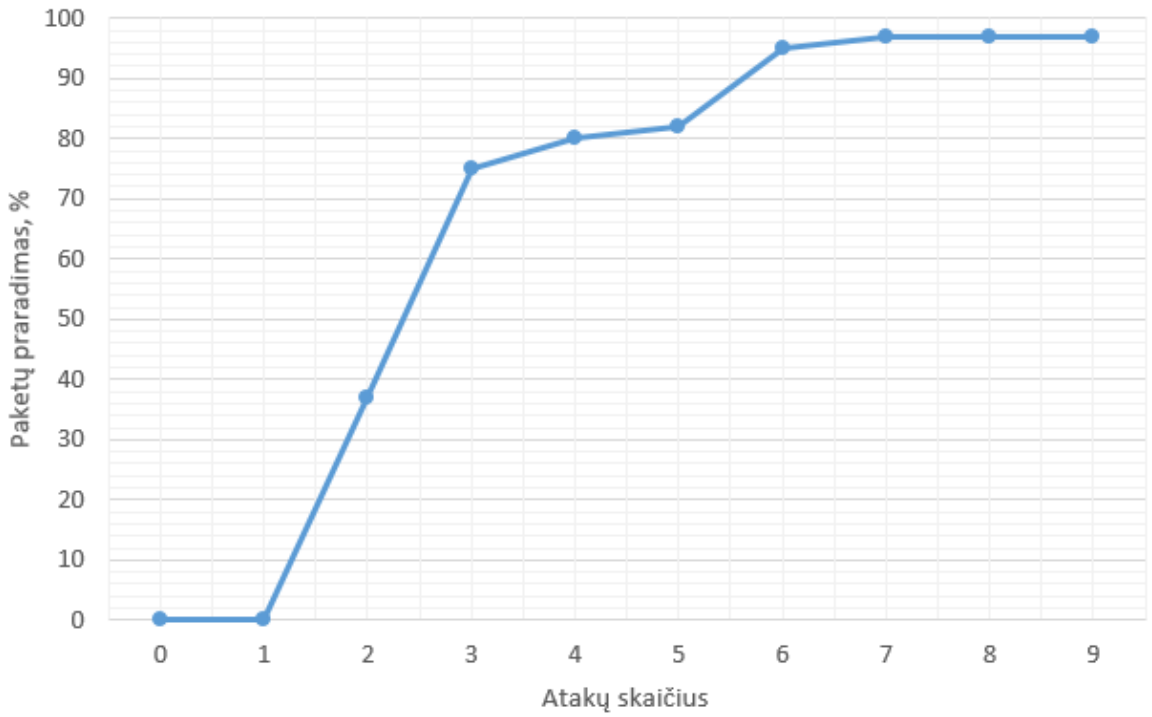
Simuliuojant DDOS ataką prieš serverį buvo bandoma užteršti visą interneto pralaidumą į valdymo tinklo serverį, taip sutrikdant valdymo tinklo pasiekiamumą. DDOS atakai naudotas prieš tai minėtas atakos tipas. Serverio pasiekiamumą buvo bandoma nustatyti iš neatakuojančio kompiuterio vykdant standartinę „PING“ komandą, tikrinant paketų vėlavimo laiką ir paketų praradimo procentą. Bandymo rezultatai pavaizduoti 4-4 Lentelė. Serverio pasiekiamumas DDOS atakos metu

4-4 Lentelė. Serverio pasiekiamumas DDOS atakos metu

Bandymo Nr.	Vykdomų DDOS atakų skaičius	„Ping“ paketo vėlavimas	Paketų praradimo procentas
1	0	4-5 ms	0 %
2	1	70-78 ms	0 %
3	2	173-196 ms	37 %
4	3	226-250 ms	75 %
5	4	251-259 ms	80 %
6	5	252-259 ms	82 %
7	6	252-258 ms	95 %
8	7	252-259 ms	97 %
9	8	252-259 ms	97 %
10	9	252-259 ms	97 %

Aukščiau pavaizduotoje lentelėje yra aprašyti bandymai, kiek bandymo metu yra vykdomų atakų į serverį ir kaip tai atsiliepia serverio pasiekiamumui. Pirmo bandymo metu nebuvo daroma atakų į serverį, serverio paketų vėlavimas 4-5 ms, paketų praradimas 0%. Vykdant antrą bandymą, kurio metu buvo atliekama viena ataka į valdymo tinklo serverį, paketų praradimo procentas buvo 0, bet atsirado nemažas paketų vėlavimas, tai yra 70-78 ms. Nuo trečio bandymo, kai buvo atliekamos dvi atakos pradėjo stipriai vėluoti paketai. Tai pat buvo pasiekta tinklo pralaidumo riba, nuo kurios pradėjo paketai nebepasiekti serverio ir atsirado paketų praradimas. Tolimesnėse atakose, paketo vėlavimas tik didėjo, tuo pačiu didėjo ir paketų praradimo procentas. Nuo septynių atakų prieš serverį, buvo pasiektas maksimalus paketų praradimo procentas, tai yra 97%. Tolimesnėse atakose paketų praradimo procentas nedidėjo. Norint dar labiau padidinti paketų praradimo procentą turėjo būti daromos atakos daugiau, nei iš dviejų kompiuterių.

Paketų praradimas pavaizduotas grafike 4.9 pav. Paketų praradimas priklausomai nuo atakų skaičiaus Grafike matyti kaip stipriai pirmosios atakos įtakoja serverio pasiekiamumą.



4.9 pav. Paketų praradimas priklausomai nuo atakų skaičiaus

4.4.2. Maršrutizatorių pridėjimas, kai vyksta DDOS ataka.

Bandymo metu, iš dviejų kompiuterių buvo vykdoma anksčiau aprašyta DDOS ataka ir bandoma prie sistemos pirmą kartą prijungti 10 maršruto parinktuvų vienu metu. Serveryje buvo atliktas apribojimas, kad serveris aptarnauja tik du maršruto parinktuvus vienu metu. Atakos metu surinkti duomenys aprašyti 4-5 Lentelė. Maršruto parinktuvų pridėjimas prie sistemos, vykstant DDOS atakai

4-5 Lentelė. Maršruto parinktuvų pridėjimas prie sistemos, vykstant DDOS atakai

Bandymo Nr.	**Maršrutizatorių skaičius**	**Vykdomų DDOS atakų skaičius**	**Laikas kol prisijungė visi maršrutizatoriai**
1	10	1	~83s
2	10	2	~107s
3	10	3	~9 min
4	10	4	~20 min
5	10	5	–
6	10	6	–

Iš anksčiau atliktų bandymų žinoma, kad serveriui atliekant po dvi užduotis vienu metu dešimt

maršruto parinktuvų aptarnauja per 83 sekundes. Vykdant vieną DDOS ataką prieš serverį prisijungimo laikas nepadidėjo. Vykdant dvi atakas prieš serverį buvo prarandami paketai, kas įtakojo ilgesnį maršruto parinktuvų aptarnavimą autentifikavimo serveryje. Nuo trijų atakų prieš serverį prasidėjo ilgas maršruto parinktuvų aptarnavimas sistemoje, nes dažnai būdavo prarandami siunčiami paketai. Taip pat dar padidinus atakų skaičių prieš serverį, maršruto parinktuvai pradėdavo atsijunginėti nuo serverio. Pasiekus 5 atakas prieš serverį, prie serverio nebebuvo galimybės pridėti maršruto parinktuvo, nes tik prisijungus maršruto parinktuvui prie serverio, serveris siųsdavo papildomas užklausas į maršruto parinktuvą, ir nesulaukus atsakymo, įrenginys atsijungdavo nuo serverio, dėl didelio paketų praradimo.

4.4.3. Maršruto parinktuvų pasiekiamumas iš serverio, kai vyksta DDOS ataka.

Paskutinio tyrimo metu, buvo atliktas tyrimas, kurio metu buvo ištirta ir išsiaiškinta, kaip jau esami maršruto parinktuvai reaguoja į vykstančią DDOS ataką prieš maršruto parinktuvų valdymo tinklą. Tai yra buvo bandoma pasiekti maršruto parinktuvą iš valdymo tinklo. Taip pat buvo stebima kiek reikia apkrauti valdymo tinklą, kad maršruto parinktuvai pradėtų atsijunginėti nuo valdymo tinklo. Bandymo metu gauti rezultatai surašyti į lentelę ir pavaizduoti 4-6 Lentelė. Maršrutizatorių pasiekimas iš serverio, vykstant DDOS atakai

4-6 Lentelė. Maršrutizatorių pasiekimas iš serverio, vykstant DDOS atakai

Bandymo Nr.	**Vykdomų DDOS atakų skaičius**	**„Ping“ paketo vėlavimas**	**Paketų praradimo procentas**
1	0	95 – 120 ms	0%
2	1	106-261 ms	0 %
3	2	208-906 ms	16 %
4	3	774-1722 ms	33 %
5	4	800-3716 ms	41 %
6	5	858-3714 ms	50 %
7	6	902-3718 ms	63 %
8	7	–	–
9	8	–	–

Lentelėje surašytuose duomenyse matoma, kad normaliame sistemos veikime, iš valdymo tinklo serverio vykdant „PING“ komandą į maršruto parinktuvą, per OpenVPN tunelį paketų praradimas lygus nuliui, o paketo vėlavimo laikas nuo 95 ms iki 120 ms. Vykdant vieną ataką

paketai tarp serverio ir maršruto parinktuvo nebuvo prarandami, tik padidėjo paketo vėlavimo laikas iki 261ms. Iki šešių atakų prieš serverį, maryti kad paketų praradimo procentas didėja, kaip ir paketo vėlavimo laikas didėja. Vykdant nuo septynių atakų prieš serverį, bandymo metu buvau atjungtas nuo valdymo tinklo serverio, ir nebebuvo galimybė išmatuoti prarandamą paketų procentą, bei paketų vėlavimo laiką.

4.5. Tyrimo išvados

Tyrimo metu analizuojant siunčiamus paketus tarp serverio ir maršruto parinktuvo, nustatyta, kad siunčiami prisijungimo duomenys ir vėlesni paketai yra šifruoti, kas leidžia duomenis tarp serverio ir maršruto parinktuvo perduoti saugiai. Atlikus tyrimą nustatyta, kad perduoti sertifikatus per OpenVPN tunelį yra saugu, nes duomenys siunčiami tarp maršruto parinktuvo ir serverio yra šifruoti.

Tyrimo metu nustatyta, kad maršruto parinktuvui pirmą kartą prisijungti prie valdymo tinklo nereikia daug laiko. Norint užtikrinti nepertraukiamą sistemos darbą, serveryje aptarnaujamus maršruto parinktuvus būtina rikiuoti į eilę, kad nebūtų per daug apkraunamas serveris. Jungiant keletą maršruto parinktuvų vienu metu, prijungimą galima pagreitinti serveriui aptarnaujant po du maršruto parinktuvus vienu metu. Tobulinant sistemą, ir norint dar labiau pagreitinti kelių maršruto parinktuvų prisijungimą pirmą kartą reikėtų programoje realizuoti procesoriaus apkrovų stebėjimą. Toks procesoriaus stebėjimas leistu vienu metu atlikti dar daugiau užduočių, tai yra kai keli procesai siunčia duomenis į maršruto parinktuvą ir procesorius apkrautas minimaliai, kiti procesai tuo metu galėtų generuoti sertifikatus.

Tyrimo metu nustatyta, kad jungiantis dešimt maršruto parinktuvų prie valdymo tinklo vienu metu nebe pirmą kartą, nėra jokio serverio prijungimo užvėlavimo, kas leidžia aptarnauti visus maršruto parinktuvus vienu metu.

Tiriant DDOS atakos įtaka serveriui paaiškėjo, kad atakos vykdomos prieš serverį turi didelę įtaka jau prijungtiems įrenginiams, kurie pradeda atsijunginėti nuo sistemos ir taip pat mažėja galimybė pridėti naujus įrenginius prie sistemos vykstant atakai. Norint padidinti serverio atsparumą nuo DDOS atakų reikalingos papildomos apsaugos serveryje. Tai yra reikia išjungti serverio atsakinėjimą į ICMP paketus, tai pat reikėtų realizuoti paketų atmetimą, kai į serverį siunčiamas didelis užklausų kiekis iš to paties IP adreso.

5. DARBO REZULTATAI IR IŠVADOS

Darbo rezultatai:

1. Analizės metu buvo nagrinėjamos jau esamos konkurentų sistemos. Taip pat analizės metu buvo išanalizuoti jau esami valdymo tinklo sprendimai, bei siūlomi standartiniai protokolai. Atliekant analizę buvo atkreipiamas dėmesys, ko labiau nori klientai iš būsimos sistemos. Kadangi klientų noras buvo tiesiogiai pasiekti gaminį iš saugaus valdymo tinklo, tad buvo nuspręsta naudoti OpenVPN tunelius tarp valdymo tinklo ir maršruto parinktuvo.
2. Saugiam maršruto parinktuvo ir valdymo tinklo darbui užtikrinti buvo reikalinga saugiai ir automatiškai perduoti sertifikatus iš sertifikatų generavimo serverio į maršruto parinktuvą. Tokiam sprendimui buvo nuspręsta naudoti pirminį OpenVPN autentifikavimo serverį iš kurio sertifikatai saugiai, šifruotu tuneliu, siunčiami į maršruto parinktuvą. Prie pirminio autentifikavimo serverio maršrutizatoriai jungiasi su prisijungimo vardu ir slaptažodžiu.
3. Autentifikavimo serveriui apsaugoti, nuo per didelės procesoriaus apkrovos, kai jungiasi dideli kiekiai maršruto parinktuvų vienu metu, buvo realizuotas užduočių statymas į eilę. Norint pagreitinti maršruto parinktuvų prijungimą buvo realizuotas kelių užduočių vykdymas vienu metu, tam panaudojant „multithreading“.
4. Realizuotas saugus valdymo tinklas, pagal iškeltus klientų reikalavimus. Realizuotas automatinis maršruto parinktuvų susijungimas su saugiu valdymo tinklu, prieš tai maršruto parinktuvui susijungiant su sertifikatų generavimo servisu ir parsisiunčiant sertifikatus.
5. Tyrimo metu buvo įsitikinta, kad visi duomenys tarp serverio ir maršruto parinktuvų siunčiami šifruotu tuneliu. Taip pat tyrimo metu ištirti laikai, reikalingi maršruto parinktuvams prisijungti prie serverio, esant skirtingai serverio tinklo apkrovai, bei skirtingam maršrutizatorių kiekiui besijungiančiam prie serverio vienu metu.

Išvados:

1. Sukurta sistema veikia ne tik su „UAB Teltonika“ įmonės LTE maršrutizatoriais, bet ir su automobiliniais „RUT850“ maršrutizatoriais, nes palaiko reikalingus protokolus.
2. Sukurtą sistemą galima pritaikyti ir kitų įmonių LTE maršruto parinktuvams. Svarbu, kad maršrutizatoriai palaikytų OpenVPN, JSON-RPC protokolus.
3. Yra galimybė padidinti sukurtos sistemos atsparumą DDOS atakoms, išjungiant PING komandos atsakymą iš serverio ir panaudojant ugniasienės taisykles, kurios ribotų didelį kiekį užklausų iš vieno IP adreso.
4. Sukurta sistema yra viename serveryje, bet taip pat yra galimybė autentifikavimo servisą ir klientų valdymo tinklų servisus iškelti į atskirus fizinius serverius.

LITERATŪRA

[1] „TR-069 friendly-tech,“ [Tinkle]. Available: http://www.friendly-tech.com/. [Kreiptasi 17 03 2016].

[2] „R-SEENET,“ [Tinkle]. Available: http://www.bb-smartcellular.cz/r-seenet-78616645/. [Kreiptasi 15 04 2016].

[3] „TR-069,“ [Tinkle]. Available: https://www.broadband-forum.org/technical/download/TR-069.pdf. [Kreiptasi 17 03 2016].

[4] „SNMP protokolas,“ [Tinkle]. Available: http://www.tcpipguide.com/free/t_SNMPProtocolBasicRequestResponseInformationPollUsi.htm. [Kreiptasi 14 02 2016].

[5] B. Forouzan, „SNMP,“ įtraukta *Data communications and networking*, New York, McGraw-Hill, 2007, pp. 891-897.

[6] JSON. [Tinkle]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf. [Kreiptasi 21 03 2016].

[7] „RPC,“ [Tinkle]. Available: http://www.cs.cf.ac.uk/Dave/C/node33.html. [Kreiptasi 07 03 2016].

[8] „JSON-RPC,“ [Tinkle]. Available: http://www.jsonrpc.org/specification. [Kreiptasi 12 03 2016].

[9] „OpenVPN,“ [Tinkle]. Available: https://openvpn.net/index.php/open-source/documentation.html. [Kreiptasi 02 04 2016].

[10] J. ". Erickson, įtraukta *Hacking: the art of exploitation Second Edition*, San Francisco, William Pollock, 2008, pp. 256-257.

PRIEDAI