



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

Zero Trust principų taikymas debesų kompiuterijos saugoje **Implementation of Zero Trust Principles in Cloud Security**

Baigiamasis bakalauro darbas

Atliko: Karolis Živulskas

VU el. p.: karolis.zivulskas@mif.stud.vu.lt

Vadovas: lektorius Tomas Poškevičius

Vilnius
2024

TURINYS

SANTRAUKA	5
SUMMARY	6
1. ĮVADAS.....	7
1.1. Tikslas.....	8
1.2. Uždaviniai.....	8
2. ANALITINĖ DALIS	9
2.1. Zero Trust.....	9
2.1.1. Zero Trust koncepcija	9
2.1.2. Zero Trust privalumai bei iššūkiai.....	10
2.2. Debesų kompiuterijos sauga.....	10
2.2.1. Debesų kompiuterijos saugos svarba.....	10
2.2.2. Duomenų šifravimas	11
2.2.3. Tapatybės ir prieigos valdymas	11
2.2.4. Infrastruktūros saugumo stiprinimas	11
2.2.5. Atsakomybių dalijimosi modelio taikymas.....	12
2.3. Debesų kompiuterijos sauga AWS debesyje.....	13
2.3.1. AWS Shield	13
2.3.2. AWS WAF	13
2.3.3. AWS CloudFront	13
2.3.4. AWS Security Hub	14
2.3.5. AWS IAM	14
2.3.6. AWS GuardDuty	14
2.3.7. AWS Inspector	15
2.3.8. AWS Verified Access.....	15
2.3.9. AWS VPN	15
2.3.10. AWS KMS.....	15
2.3.11. AWS CloudTrail.....	15
2.3.12. AWS Certificate Manager	16
2.3.13. AWS Detective.....	16
2.3.14. AWS CloudWatch	16
2.3.15. AWS Macie	16
2.3.16. AWS Config.....	17
3. METODINĖ DALIS.....	18
3.1. Aplinkos kūrimas.....	18
3.1.1. Pasirinkti įrankiai	18

3.1.2.	Tinklo dalies kūrimas.....	18
3.1.3.	Prieigų ir teisių dalies konfigūravimas.....	18
3.1.4.	Duomenų bazės kūrimas.....	18
3.1.5.	Duomenų talpyklos kūrimas.....	18
3.1.6.	Serverių kūrimas.....	19
3.2.	Duomenys bei programa	19
3.3.	Zero Trust taikymas AWS aplinkai	19
3.3.1.	Išplėstinė priežiūrų ir teisių peržiūra.....	19
3.3.2.	Dirbtiniu intelektu grįstų saugumo įrankių taikymas	19
3.3.3.	Bendrinės AWS ugniasienės konfigūravimas	19
3.3.4.	Sertifikatų ir raktų valdymas AWS įrankiuose	20
3.3.5.	CloudTrail ir CloudWatch pritaikymas paskyros stebėjimui.....	20
3.3.6.	Centralizuoto stebėjimo centro sukūrimas.....	20
3.3.7.	Atsarginių kopijų valdymas	20
3.3.8.	Tinklo segmentacija.....	20
3.3.9.	Turinio transliavimo tinklas.....	20
3.4.	Sistemos veikimo bei saugumo tikrinimas.....	21
3.4.1.	Prisijungimas su nutekėjusiais duomenimis.....	21
3.4.2.	Sistemos trikdymo ataka	21
3.4.3.	DNS užklausos iš ribojamų geolokacijų bei IP adresų	21
3.4.4.	RDS duomenų bazės pasiekiamumas iš išorės	21
3.4.5.	Programinės sąsajos prieinamumo tikrinimas	21
3.4.6.	Perimti SSH raktai ir prisijungimas prie AWS EC2	21
3.4.7.	Bandymas kompromituoti užklausų skirstytuvą	21
3.4.8.	Dirbtiniu intelektu grįstų įrankių radinių vertinimas.....	21
4.	ĮGYVENDINIMO DALIS	22
4.1.	Aplinkos kūrimas.....	22
4.1.1.	Pasirinktų įrankių paruošimas	22
4.1.2.	Tinklo dalies sukūrimas	22
4.1.3.	Prieigų ir teisių dalies konfigūravimas.....	23
4.1.4.	Duomenų bazės kūrimas	23
4.1.5.	Duomenų talpyklos kūrimas.....	23
4.1.6.	Serverio kūrimas	24
4.2.	Duomenys bei programa	24
4.2.1.	Programos įrašymas	24
4.2.2.	Duomenų generavimas bei talpinimas.....	25
4.3.	Zero Trust taikymas AWS aplinkai	26
4.3.1.	Išplėstinė prieigų ir teisių peržiūra	26

4.3.2.	Dirbtiniu intelektu grįstų saugumo įrankių taikymas	26
4.3.3.	Bendrinės AWS ugniasienės konfigūravimas	27
4.3.4.	Sertifikatų ir raktų valdymas AWS įrankiuose	27
4.3.5.	CloudTrail ir CloudWatch pritaikymas paskyros stebėjimui.....	27
4.3.6.	Centralizuoto stebėjimo centro sukūrimas	28
4.3.7.	Atsarginių kopijų valdymas	28
4.3.8.	Tinklo segmentacija.....	28
4.3.9.	Turinio transliavimo tinklas.....	29
4.4.	Sistemos veikimo bei saugumo tikrinimas.....	29
4.4.1.	Prisijungimas su nutekėjusiais duomenimis.....	29
4.4.2.	Sistemos trikdymo ataka	30
4.4.3.	DNS užklausos iš ribojamų geolokacijų bei IP adresų	30
4.4.4.	RDS duomenų bazės pasiekiamumas iš išorės	31
4.4.5.	Programinės sąsajos prieinamumo tikrinimas	31
4.4.6.	Perimti SSH raktai ir prisijungimas prie AWS EC2	32
4.4.7.	Bandymas kompromituoti užklausų skirstytuvą	32
4.4.8.	Dirbtiniu intelektu grįstų įrankių radinių vertinimas.....	32
5.	REZULTATAI IR APIBENDRINIMAS	34
	LITERATŪRA.....	36
1	priedas.....	37
2	priedas.....	38
3	priedas.....	39
4	priedas.....	42
5	priedas.....	43
6	priedas.....	46
7	priedas.....	50
8	priedas.....	52
9	priedas.....	54
10	priedas.....	57
11	priedas.....	59
12	priedas.....	61
13	priedas.....	62
14	priedas.....	63
15	priedas.....	65
16	priedas.....	66
17	priedas.....	68
18	priedas.....	71
19	priedas.....	73

20 priedas	74
21 priedas	75
22 priedas	76
23 priedas	77
24 priedas	78
25 priedas	80
26 priedas	81
27 priedas	83
28 priedas	84
29 priedas	85
30 priedas	86

SANTRAUKA

Pagrindinis baigiamojo darbo tikslas – sukurti visapusiškai veikiančią ir Zero Trust koncepciją idėjiškai atitinkančią saugumo strategiją AWS debesijoje kuriamoms sistemoms. Darbo metu buvo sprendžiami šie uždaviniai: išanalizuoti AWS debesijos teikiamus saugos sprendimus bei Zero Trust pritaikomumą debesų kompiuterijoje, suprojektuoti saugią AWS debesijos infrastruktūrą, atitinkančią Zero Trust koncepciją, sukurti Zero Trust koncepciją atitinkančius praktinius debesų kompiuterijos saugos sprendimus bei infrastruktūrą AWS debesijoje, patikrinti sukurtos saugumo strategijos koncepciją atitinkančios infrastruktūros saugumą imituojant galimas grėsmes. Sukurta saugumo strategija, atitinkanti Zero Trust koncepciją bei apimanti visą sistemą holistiškai. Išbandytas sukurtos sistemos taikymas AWS infrastruktūroje realizuojant tinklo vidinę bei išorinę saugą, tinklo mikrosegmentaciją, atsarginių kopijų šifravimą bei konfigūravimą, dirbtinio intelekto taikymą grėsmių aptikimui.

Raktiniai žodžiai: debesija, sauga, nulinis, pasitikėjimas.

SUMMARY

The main objective of the thesis is to develop a fully operational security strategy for systems built on the AWS cloud that is conceptually consistent with the Zero Trust concept. The thesis addressed the following tasks: to analyse the security solutions provided by the AWS cloud and the applicability of Zero Trust in cloud computing, to design a secure infrastructure for the AWS cloud complying with the Zero Trust concept, to develop practical cloud computing security solutions and infrastructure for the AWS cloud complying with the Zero Trust concept, and to test the security of the infrastructure for the developed infrastructure complying with the Zero Trust concept by simulating possible threats. A security strategy has been developed that is consistent with the Zero Trust concept and covers the whole system holistically. The application of the developed system to the AWS infrastructure is tested by implementing internal and external network security, network microsegmentation, backup encryption and configuration, and the application of artificial intelligence for threat detection.

Keywords: cloud, security, zero, trust.

1. ĮVADAS

Šioje dalyje pateikiami baigiamojo bakalauro darbo tikslai, uždaviniai, teorinės medžiagos, reikalingos uždavinių išpildymui, analizė bei įgyvendinimas. Baigiamąjį darbą sudaro aštuonios dalys: įvadas, analitinė dalis, metodinė dalis, įgyvendinimo dalis, rezultatai ir apibendrinimas, išvados, literatūros sąrašas, priedai.

Pasirinkta bakalauro darbo tema yra glaudžiai susijusi su modernių laikų diktuojamais įvykiais bei aplinkybėmis. Debesų kompiuterija yra itin plačiai paplitusi, todėl net archajišką architektūrą taikančios įmonės stengiasi įgyvendinti hibridinės debesijos taikymą. Taip nutinka dėl to, jog įmonėms vis mažiau apsimoka kurti savo fizinius duomenų centrus bei juos išlaikyti nei mokėti viešosios debesijos tiekėjams (AWS, Azure, Google Cloud). Dalis saugumo atsakomybių tenka debesijos platformai, o kita dalis naudotojui. Siekiant, kad būtų užtikrintas pilnavertiškas sistemos saugumas, reikia taikyti nulinio pasitikėjimo arba kitaip Zero Trust koncepciją.

Taip pat vis daugiau įmonių bei darbuotojų renka darbą nuotoliu. Tai atspindi darbo etikos pažangą, tačiau sukuria ir papildomų iššūkių. Atsiranda tikimybė, jog darbuotojo naudojamas tinklas gali būti nesaugus. Jei ne itin apsaugotame tinkle naudojami keli prietaisai, užkrėsti asmeniniai prietaisai gali tapti tiltu į darbui naudojamą įrenginį. Taip iškyla grėsmė įmonės sistemoms. Tinkama sauga gali apsaugoti debesijoje esančią sistemą ir nuo prisijungimų su neteisėtai perimtais prisijungimo duomenimis. Svarbu paminėti, jog debesijoje yra sunkiau atpažinti ar naudotojas yra tai, kuo prisistato. Todėl yra itin svarbu taikyti nuolatinį naudotojų veiksmų stebėjimą bei įtartinų atvejų paiešką. Taip galima užkirsti kelią sistemos sutrikdymui ar sunaikinimui.

Kitas svarbus aspektas yra hibridinė Rusijos agresija prieš Lietuvą. Itin reikalinga stiprinti Lietuvos įmonių bei institucijų debesijos infrastruktūros saugą, kadangi priešišškai nusiteikę kibernetiniai nusikaltėliai bando pakenkti. Vienas iš kertinių aspektų yra duomenų saugumas bei gebėjimas juos atkurti. Pritaikant Zero Trust koncepcijos idėjas galima apsaugoti programas bei duomenis nuo sistemos trikdymo atakų, duomenų perėmimo bei kitų užpuolimų.

Ne mažiau svarbus aspektas yra ir individualūs debesijos vartotojai. Debesų kompiuterija nepatyrusiems naudotojams sukelia išpūdį, jog viskas yra ganėtinai paprasta ir saugu. Reikia pabrėžti, jog pagal atsakomybių dalijimosi principą debesijų tiekėjai rūpinasi tik dalimi saugos. Tai reiškia, jog kitą saugumo įgyvendinimo dalį naudotojas turi atlikti pats.

Zero Trust koncepcija nėra konkrečių žingsnių ar veiksmų seka, kurią įgyvendinus, garantuojamas sistemos saugumas. Į šią sąvoką vertėtų žvelgti kaip į nuolatinį postūmį gerinti saugumą. Tai reiškia, jog sistema elementariai niekada nėra pakankamai saugi, o jos viduje gali būti įsibrovėlių. Taip yra sukuriamas postūmis labiau saugoti savo programas, duomenis bei infrastruktūros resursus.

Darbe pasirinkta atlikti tyrimą AWS debesijos platformoje todėl, jog AWS šiuo metu yra labiausiai paplitusi viešoji debesija ir yra lyderė rinkoje. Svarbu paminėti, jog AWS debesija neturi sukurto Zero Trust įgyvendinimo sprendimo ar instrukcijų, kaip tai padaryti. Todėl tyrimo metu pritaikyta saugumo sprendimų visuma kuo labiau atsižvelgiant į Zero Trust koncepciją, teigiančią, jog sistema yra nuolat nesaugi. Įgyvendinus Zero Trust principų pritaikymą kuriamame

saugumo mechanizme, atlikti sukurtos AWS aplinkos testavimai. Bandymai atlikti imituojant įvairaus tipo nelaimes ar įsilaužimo bandymus.

1.1. Tikslas

Šio baigiamojo bakalauro darbo tikslas yra sukurti Zero Trust koncepciją atitinkančią saugumo strategiją bei ją pritaikyti AWS debesijoje sukurta infrastruktūrai pilnai ją apsaugant iš naudotojo pusės.

1.2. Uždaviniai

Bakalauro baigiamajam darbui išsikelti uždaviniai:

1. Išanalizuoti AWS debesijos teikiamus saugos sprendimus bei Zero Trust pritaikomumą debesų kompiuterijoje.
2. Suprojektuoti saugią AWS debesijos infrastruktūrą, atitinkančią Zero Trust koncepciją.
3. Sukurti Zero Trust koncepciją atitinkančius praktinius debesų kompiuterijos saugos sprendimus bei infrastruktūrą AWS debesijoje.
4. Patikrinti sukurtos Zero Trust koncepciją atitinkančios infrastruktūros saugumą imituojant galimas grėsmes.

2. ANALITINĖ DALIS

Baigiamojo bakalauro darbo metu buvo nagrinėjama Zero Trust saugos koncepcija bei debesų kompiuterijos sauga. Taip pat analizuotas Zero Trust principų taikymas debesų kompiuterijos saugoje. Šioje dalyje aprašyti ir AWS debesijoje taikomi saugumo įrankiai bei jų savybės.

2.1. Zero Trust

Šiame skyrelyje pateikiama Zero Trust saugumo koncepcija bei jos ypatumai.

2.1.1. Zero Trust koncepcija

Zero Trust yra metodologija, kuri išvertus į lietuvių kalbą, reiškia pasitikėjimo nebuvimą arba kitaip „nulinį pasitikėjimą“. Ši koncepcija nesilaiko griežtai suformuluotų taisyklių bei neturi savo išskirtinių įrankių. Tai skatina žvelgti į šią koncepciją kaip į strategiją, o ne išbaigtą sprendimą ar konkrečią technologiją.

Nulinis pasitikėjimas reiškia, jog visi tinkle dalyvaujantys komponentai (naudotojai, prietaisai, programinės sąsajos) yra laikomi nepatikimais. Sistema, kurioje yra pritaikyta Zero Trust saugumo strategija, prieš atliekant kažkokius veiksmus, visuomet reikalauja tapatybės patvirtinimo. Sistemos naudotojams bei programinėms sąsajoms yra paskiriamos minimalios teisės, neviršijančios jų realaus poreikio. Pavyzdžiui, programinei sąsajai dėl duomenų apdorojimo reikia nuskaityti duomenis iš konkrečios duomenų talpyklos. Tokiu atveju programinės sąsajos teisės apims tik nuskaitymo operacijas toje duomenų talpykloje. Jeigu prieigos teisės šiuo atveju apimtų daugiau veiksmų ar daugiau duomenų talpyklų, toks teisių priskyrimas jau nebeatitiktų Zero Trust strategijos.

Nulinio pasitikėjimo strategiją galima vadinti globalaus amžiaus strategija. Ji ypatingai pritaikyta šiuolaikiniam verslui, atsižvelgiant į darbuotojų saugumą nuotolinėse, hibridinėse ar pilnai debesijos pagrindu veikiančiose darbo aplinkose, siekiant apsaugoti nuo sparčiai plintančios kenkėjiškos programinės įrangos bei įsilaužimų į sistemas. Zero Trust modelį apibūdina šie teiginiai [11]:

- Niekuo nepasitikėk nepriklausomai nuo to, kur jis yra. Visada patikrink keletą kartų.
- Mažiausios būtinos teisės: prieigos valdymas grindžiamas „būtina žinoti“ principu ir yra griežtai vykdomas.
- Visi resursai turi būti pasiekiami saugiai nepriklausomai nuo vietos.
- Dinaminis vartotojų, aplikacijų ir įrenginių reitingavimas, vertės nustatymas.
- Kontroliuok ir analizuok visą duomenų srautą.

Tradiciniai saugumo modeliai dažnai yra grindžiami idėja, kad vidaus tinklo elementai yra saugūs. Zero Trust strategija prieštarauja šiai nuostatai ir remiasi principu, kad net ir vidiniai tinklo elementai nėra saugūs. Pagrindiniai metodai, leidžiantys įgyvendinti nulinio pasitikėjimo strategiją sistemose yra dinaminis prieigos teisių taikymas, autentifikavimo ir autorizavimo procedūros, nuolatinis tinkle dalyvaujančių naudotojų, prietaisų bei programinių sąsajų veiksmų stebėjimas ir saugojimas, kelių veiksmų autentifikavimo priemonių naudojimas bei atnaujintų programų versijų naudojimas.

2.1.2. Zero Trust privalumai bei iššūkiai

Šiame skyrelyje pateikiu Zero Trust modelio privalumus bei iššūkius (žr. 2.1 Lentelė).

2.1 Lentelė. Zero Trust modelio privalumai bei iššūkiai.

Privalumai	Iššūkiai
Zero Trust koncepcija turi gerą suderinamumą su debesija.	Perėjimas prie Zero Trust gali užtrukti (resursų reikalaujantis pritaikymas).
Zero Trust modelis itin palengvina nuotolinį tinklo naudojimą (nuotolinis darbas).	Galimas naudotojų nepasitenkinimas.
Centralizuoti įrankiai patogiau leidžia stebėti tinklo būseną bei naudotojų veiksmus.	Pasenusias programas dėl nesuderinamumo gali tekti atnaujinti skiriant nemažus finansinius išteklius.
Mažesnės galimybės duomenų nutekinimui.	Didėjantys kaštai saugumo sričiai.

Zero Trust modelio įgyvendinimas yra ganėtinai sudėtingas ir laiko reikalaujantis procesas. Šis sudėtingumas kyla iš to, jog yra reikalinga pilna sistemų inventorizacija. Visi prietaisai, naudotojai bei programinės sąsajos turi būti žinomi. Tuomet pagal reikiamą prieigos lygį nustatomos teisės kiekvienam naudotojui ar sąsajai. Kitas kertinis iššūkis įgyvendinant Zero Trust strategiją yra sistemų suderinamumas. Yra tikimybių, jog sistema turėtų būti atnaujinama tinkamam komponentų tarpusavio suderinamumui. Tai gali sugeneruoti papildomus finansinius kaštus bei užtrukti papildomai laiko.

Nepaisant kylančių iššūkių įgyvendinant Zero Trust modelį, nauda yra neabejotina. Šis modelis sukuria sistemos kontrolės mechanizmus, leidžiančius stebėti tinkle vykstančius procesus bei konkrečių prietaisų ar naudotojų atliktus veiksmus. Tai ženkliai pagerina duomenų saugumo situaciją [9]. Taip pat įvykus kritinei situacijai, yra lengviau imtis tolesnių veiksmų dėl nuolatinio tinklo stebėjimo. Nepaisant nemažų finansinių kaštų Zero Trust strategijos pritaikymui, šis procesas potencialiai apsaugo nuo daug didesnės finansinės naštos duomenų nutekėjimo atveju.

2.2. Debesų kompiuterijos sauga

Šiame skyrelyje apžvelgiamos pagrindinės gerosios debesų kompiuterijos saugumo praktikos ir principai.

2.2.1. Debesų kompiuterijos saugos svarba

Debesų kompiuterijos sauga apima teises, technologijas bei praktikas, kuriomis siekiama apsaugoti debesijoje esančią infrastruktūrą bei duomenis. Prieš pradėdant detalesnę debesijos saugos komponentų analizę svarbu paminėti, jog debesų kompiuterijos saugoje yra kelios suinteresuotos šalys. Viena šalis yra debesijos platformos kūrėjai, o kita – šios platformos naudotojai. Tam tikrais atvejais galime turėti dvi šalis viename, kai privataus debesies kūrėjai yra ir jo naudotojai. Visgi, dažnesnė forma yra, kai šalys yra atskiros (viešojo debesies kūrėjai AWS, Microsoft Azure, Google Cloud Platform ir kt. bei naudotojai). Tokiu atveju pilnavertei debesijos saugai reikia abiejų šalių pastangų [10]. Tinkamai įgyvendintos saugumo praktikos apsaugo nuo kylančių rizikų:

- Duomenų nutekinimo.
- Prieigos, kuri neturėtų būti įgalinta, suteikimo.

- Duomenų praradimo.
- Bendro naudojimo infrastruktūros pažeidžiamumo.
- Netinkamo debesijos resursų konfigūravimo.
- Netinkamo šifravimo taikymo.

Šias rizikas galima eliminuoti tinkamai pritaikius saugumo praktikas tiek viešojo debesies infrastruktūrai (platformos kūrėjo atsakomybė), tiek debesijoje naudojamiems komponentams (naudotojas). Kituose skyreliuose išskiriamos pagrindinės saugos praktikos, darančios įtaką bendrai debesijos būklei.

2.2.2. Duomenų šifravimas

Vienas iš svarbiausių debesijos saugumo aspektų yra tinkamas duomenų šifravimas tiek ramybės, tiek perdavimo metu. Tai garantuoja, jog prieigą prie duomenų turi tik įgalinti naudotojai.

Duomenų šifravimui galima naudoti tiek simetrinius, tiek asimetrinius raktus, sertifikatus bei tokenus priklausomai nuo poreikių ir debesijos paslaugų [3]. Taip pat svarbu paminėti, jog šifravimo raktai turi būti saugomi su ypatinga priežiūra.

Įprastai viešuosiuose debesyse duomenų šifravimas yra platformos kūrėjų prerogatyva. Netgi siūlant tokius įrankius kaip valdomos duomenų bazių sistemos, platforma pasirūpina tinkamu duomenų šifravimu ir rakto saugojimu. Naudotojo atsakomybė tokiu atveju yra užtikrinti duomenų saugumą perdavimo metu. Tačiau priklausomai nuo pasirinkimo, naudotojas gali būti atsakingas už šifravimo raktus ir pats.

2.2.3. Tapatybės ir prieigos valdymas

Tapatybės ir prieigos valdymas (IAM) yra kertinis mechanizmas debesijos saugoje. Jis leidžia sukurti taisykles, įgalinančias arba draudžiančias prieigą prie konkrečių duomenų, paslaugų ar infrastruktūros resursų. Taip galima išskirstyti prieigos teises individualiai kiekvienam naudotojui ar programinei sąsajai.

Naudotojų autentifikavimui reikia naudoti slaptažodžius, biometrinius duomenis ar kelių žingsnių autentifikavimą (MFA). Kiekvienam naudotojui teisės turi būti priskirtos remiantis nuo rolės priklausančia prieigos kontrole (RBAC). Tai reiškia, jog naudotojas turi priskirtą rolę, įgalintą atlikti konkrečius veiksmus sistemoje priklausomai nuo poreikio.

Taip pat galima kurti ir naudotojų grupes. Šis tapatybės ir prieigos valdymo vienetas taip pat apibrėžia prieigos teises, tačiau į jį gali būti priskirtas ne vienas naudotojas.

Tinkamai sukonfiguruota IAM dalis viešajame debesyje leidžia atsekti naudotojų veiksmus. Iš esmės, kiekvienas veiksmas yra įrašomas į žurnalą, administruojamą platformos kūrėjų. Tuomet pagal datą, naudotoją bei veiksmo tipą galima atsekti, kas ir kada atliko konkrečius veiksmus debesyje.

2.2.4. Infrastruktūros saugumo stiprinimas

Debesų kompiuterijos saugoje yra ypatingai svarbu nuolat stiprinti esamos infrastruktūros saugumą. Tai apima periodinį konfigūracijų peržiūrėjimą bei atnaujinimą, išvengiant saugumo spragų. Dažnas infrastruktūros peržiūrėjimas leidžia atsisakyti nereikalingų prievadų, servisų ar

funkcijų, kurie nebėra naudojami. Dėl kompiuterinių resursų, kurie naudojami platformos palaikymui ir paslaugų teikimui, atsakingi debesijos platformos kūrėjai.

Taip pat svarbu periodiškai naujinti debesijos programinę įrangą bei naudoti automatizavimo įrankius konfigūracijos standartų bei taisyklių užtikrinimui. Vis dėlto, naudotojas atsako už tai, ką daro debesijos platformos jam suteiktuose resursuose. Vadinasi, naudotojas turi uždaryti nereikalingus prievadus, atnaujinti savo naudojamą operacinę sistemą.

2.2.5. Atsakomybių dalijimosi modelio taikymas

Debesų kompiuterijoje sauga yra klausimas, liečiantis tiek platformos naudotoją, tiek platformą. Šis mechanizmas yra vadinamas atsakomybių pasidalijimo modeliu. Priklausomai nuo debesijoje naudojamo įrankio, santykis tarp naudotojo ir platformos atsakomybių kinta. Įprastai platforma yra atsakinga už fizinius resursus bei jų priežiūrą, o naudotojas už tai, ką daro tų resursų izoliuotoje aplinkoje. Tačiau viešųjų debesų platformos turi daugybę valdomų įrankių. Tokiu atveju platformos atsakomybės išauga, o naudotojo sumenksta. Šis santykis kinta priklausomai nuo atsakomybių pasidalijimo modelio rūšies (žr. 2.2 Lentelė).

2.2 Lentelė. Viešojo debesies ir naudotojo atsakomybių pasidalijimas.

Atsakomybių pasidalijimo modelis	Infrastruktūros kaip paslaugos (IaaS)	Platformos kaip paslaugos (PaaS)	Programos kaip paslaugos (SaaS)
Duomenų apsauga	Naudotojas yra atsakingas už duomenų apsaugą pats, tiek ramybės būsenoje, tiek perdavimo metu.	Naudotojai turi užtikrinti, jog duomenys būtų saugūs perdavimo metu. Duomenų sauga ramybės būsenoje rūpinasi debesijos kūrėjai.	Debesijos kūrėjai atsakingi už duomenų saugą.
Tinklo sauga	Naudotojas atsakingas už tinklo segmentavimą, ugniasienes, stebėjimo sistemas.	Atsakomybė yra abiejų šalių. Debesijos kūrėjai įgyvendina saugumo parametrus, naudotojas atsako dėl atidarytų prievadų ar nesaugaus kodo rašymo (pvz, Elastic Beanstalk).	Debesijos kūrėjai atsakingi, bet naudotojas gali reguliuoti prieigą kūrėjų sukurtu keliu.
Tapatybių valdymas	Naudotojais valdo savo tapatybes bei prieigas.	Bendra atsakomybė.	Sprendimą įgyvendina debesijos platforma. Naudotojas gali koreguoti.
Fizinė sauga	Užtikrina debesijos kūrėjai.	Užtikrina debesijos kūrėjai.	Užtikrina debesijos kūrėjai.
Autentifikavimas	Naudotojai turi įgyvendinti autentifikavimosi mechanizmus.	Bendra atsakomybė. Tapatybės patvirtinimas suteikiamas iš platformos pusės, bet autentifikacija reguliuojama pačio naudotojo.	Autentifikavimas prižiūrimas debesų platformos, naudotojas gali tik keisti siūlomus nustatymus pagal savo poreikius.

Iš esmės, infrastruktūros kaip paslaugos modelis suteikia naudotojui daug laisvės. Galima pasirinkti saugumo technologijas savo nuožiūra ir įgyvendinti norimu mastu. Tačiau tai perkelia daugiau atsakomybių naudotojui. Viešajame debesyje atsakomybės dažnai yra dalijamos abiem šalims, kūrėjui ir naudotojui. Tokia metodologija, platforma kaip paslauga, ypatingai išplito, nes naudotojui sumenksta atsakomybių, bet vis dar yra pakankamai lankstumo. Šiame modelyje sauga negali egzistuoti be vienos iš šalių. Tuo tarpu programos kaip paslaugos modelis leidžia vartotojui naudotis reikalingu įrankiu, rūpinantis tik įrankio viduje esančiomis konfigūracijomis.

2.3. Debesų kompiuterijos sauga AWS debesyje

Šiame skyrelyje analizuojami AWS viešajame debesyje siūlomi saugumo įrankiai bei jų galimybės.

2.3.1. AWS Shield

AWS Shield turi ganėtinai siaurą, bet svarbią funkciją. Tai yra apsauga nuo sistemų trikdymo (DDoS) atakų. Šis įrankis yra valdomas, todėl naudotojui nereikia rūpintis dėl specifinių konfigūracijų.

Shield automatiškai aptinka beprasidedančias sistemos trikdymo atakas bei jas užkardo. Papildomai apsaugai galima integruoti AWS Shield su AWS siūloma ugniasiene WAF, sustiprinant programų saugumą.

Šis įrankis siūlo apsaugą tinklo (L3), duomenų perdavimo (L4) bei programos (L7) lygmenyse [2].

2.3.2. AWS WAF

AWS WAF yra Amazon siūloma ugniasienė. Šis įrankis padeda apsaugoti internete veikiančias programas bei programines sąsajas, užblokuojant tam tikras tinklo užklausas.

Viena iš pagrindinių AWS WAF funkcijų yra tinklo užklausų iš botų ribojimas. Tam naudotojas gali sukurti specifines taisykles pagal savo poreikius. Taip pat ši ugniasienė saugo nuo kibernetinių atakų, pavyzdžiui, SQL injekcijų [6].

Šis įrankis leidžia sukurti taisykles specifiniams IP adresams. Galima sudaryti tiek juoduosius (angl. blacklist), tiek leistinus (angl. whitelist) IP adresų sąrašus. Taip pat galima parinkti, koks atsakas bus siunčiamas atgal, gavus tinklo užklausą iš konkretaus IP adreso.

2.3.3. AWS CloudFront

Amazon CloudFront veikia kaip du įrankiai viename. Viena įrankio savybė yra turinio transliavimas tinkle, paskirstant jį įvairiose geografinėse pasaulio vietose esančiuose serveriuose. Ši technologija reiškia, jog priklausomai nuo užklausos siuntėjo fizinės lokacijos, parenkama arčiausiai esanti turinio saugykla. Toks veikimo principas paspartina turinio teikimą. Skirtingoms geografinėms lokacijoms parenkami skirtinguose geografiniuose regionuose esantys AWS infrastruktūros serveriai.

Taip pat CloudFront, kartu su AWS Shield, suteikia naudotojui apsaugą nuo sistemos trikdymo atakų. Svarbu paminėti ir tai, jog CloudFront šifruoja tinklo užklausas, taip dar labiau padidinant saugumą.

2.3.4. AWS Security Hub

AWS Security Hub yra centralizuotas įrankis, leidžiantis vienoje vietoje stebėti AWS paskyrų saugumą bei automatizuoti saugos patikrinimus. Šis įrankis naudoja grafikus informacijos apie saugumo situaciją bei atitikimą nustatytiems standartams vizualiam pateikimui.

Viena pagrindinių šio įrankių funkcijų yra surinkti bei atvaizduoti įvairius saugos radinius, įžvalgas ar vertinimus vienoje vietoje. Šis ypatumas leidžia centralizuotai viename pasirinktame regione naudotojui matyti bei valdyti surinktą informaciją iš kelių skirtingų AWS regionų.

AWS Config įrankis yra privalomas šio komponento veikimui. Tai reiškia, jog savo paskyroje naudotojui būtina įjungti AWS Config. Taip yra dėl to, jog Security Hub pasitelkia AWS Config funkcijas saugos patikrų atlikimui.

Šis resursas turi automatinę integraciją su AWS GuardDuty. Pastarasis komponentas siunčia savuosius įvykius į centralizuotąjį Security Hub ASFF (*AWS Security Finding Format*) formatu [7]. Rakiniai dažniausiai išsiunčiami per penkias minutes.

Security Hub yra puikiai integruojamas su trečiųjų šalių įrankiais. Integracija galima tiek radinių į Security Hub siuntimui (Aqua, 3CoreSec), tiek radinių iš Security Hub gavimui (Atlassian, Fortinet).

2.3.5. AWS IAM

AWS IAM yra tapatybių valdymo įrankis Amazon debesijoje. Tai yra vienas kertinių resursų, kadangi nuo jo priklauso visos prieigos teisės. Šiuo įrankiu galima kurti naudotojus, naudotojų grupes, roles bei taisykles.

AWS taisyklės apibrėžia įvairius veiksmus, kuriuos galima atlikti įvairiais AWS įrankiais [8]. Tuomet taisyklės gali būti priskiriamos prie tapatybių (naudotojas, grupė arba rolė). Geroji praktika yra taisyklės priskirti rolei, kuri turi konkrečią funkciją. Tokia rolė gali būti panaudota AWS IAM sukurtų naudotojų.

Šiame įrankyje yra itin svarbu suteikti minimalias reikalingas teises tapatybių subjektams, kadangi tinkamai paskirstytos teisės ženkliai sumažina saugumo spragų rizikas.

2.3.6. AWS GuardDuty

AWS GuardDuty yra grėsmių atpažinimo įrankis, grįstas dirbtiniu intelektu. Jis susideda iš kelių svarbių komponentų: mašininio mokymosi algoritmų, grėsmių analitikos bei naudojamų trečiųjų šalių įrankių. Pagrindinis GuardDuty tikslas yra apsaugoti naudotojo valdomas AWS paskyras, jose esančius duomenis bei programas ir jų komponentus.

GuardDuty vykdo nepertraukiamą AWS paskyrų stebėjimą. Šiais stebėjimais yra ieškoma kenkėjiškos veiklos požymių, o stebėjimų duomenys yra panaudojami detalioms saugumo ataskaitoms, kuriais remiantis galima atitaisyti žalą arba sukurti papildomus saugumo garantus.

Šis įrankis gali analizuoti įvairius duomenis:

- AWS CloudTrail įvykių įrašus (angl. „logs“) – neįprastų programinės sąsajos užklausų, resursų sukūrimo ar ištrynimo aptikimui.
- AWS VPC tinklo užklausų įrašus – neįprastų IP adresų ar vidinių tinklo užklausų pastebėjimui.
- DNS įvykių įrašus – užkoduotos informacijos siuntimo DNS užklausomis aptikimui.

GuardDuty teikia radinius (angl. „findings“) apie aukščiau minimus duomenų šaltinius. Kiekvienas radinys yra vertinamas pagal jo pavojingumo lygį (aukštas, vidutinis arba žemas).

2.3.7. AWS Inspector

AWS Inspector yra dar vienas automatizuotas debesijos saugos įrankis. Jis pristatomas kaip pažeidžiamumų ar rizikų valdymo resursas. Inspector pagrindinė funkcija yra tikrinti kompiuterinius resursus dėl programinės įrangos saugumo ar tinklo nustatymų spragų.

Patikrinimų atlikimui šis įrankis naudoja AWS EC2 virtualias mašinas, AWS ECR konteinerių šablonus arba AWS Lambda funkcijas. Kai Inspector aptinka tinklo konfigūravimo spragą arba programinės įrangos pažeidžiamumą, yra sukuriamas radinys.

Radiniuose yra įvertinamas kiekvienos saugumo spragos lygis. Taip pat nurodomas paveiktas resursas, aptikta saugumo spraga bei pažeidžiamumo sprendimo būdas.

2.3.8. AWS Verified Access

AWS Verified Access įrankis suteikia galimybę saugiai pasiekti programas be VPN (virtualus privatus tinklas) konfigūravimo. Šio įrankio pagalba naudotojas gali sukurti saugumo taisykles, kuriomis remiantis nusprendžiama, ar programos vartotojai gali naudotis programa. Tai reiškia, jog naudotojai, neatitinkantys nustatytų taisyklių, neturės prieigos.

2.3.9. AWS VPN

AWS VPN yra Amazon siūlomas virtualaus privataus tinklo sprendimas. Šis įrankis automatiškai prisitaiko prie prisijungimų skaičiaus, todėl naudotojui nereikia galvoti apie išplečiamumo konfigūracijas.

AWS siūlo kelių tipų VPN. Vienas iš jų yra AWS Client VPN, o kitas AWS Site-to-Site VPN. Pastarasis yra skirtas saugiam prisijungimui iš duomenų centro ar įmonės ofiso į AWS debesiją. AWS Client VPN yra skirtas nuotoliu dirbančių žmonių apsaugotam resursų pasiekimui tinkle tiek įmonės fizinėse lokacijose, tiek AWS debesijoje.

2.3.10. AWS KMS

AWS KMS yra valdomas raktų saugojimo įrankis. Šis komponentas supaprastina naujų kriptografinių raktų kūrimą ar jų valdymą. KMS raktų saugojimui naudoja kompiuterinės įrangos saugos modulius (HSM) pagal „FIPS 140-2 Cryptographic Module Validation Program“ programą [1].

AWS KMS lengvai integruojasi su kitais duomenis šifruojančiais AWS resursais. Taip pat šis įrankis veikia kartu su CloudTrail tam, jog būtų stebimas raktų panaudojimas ar keitimas. Naudotojas gali kurti tiek simetrinius, tiek asimetrinius raktus.

2.3.11. AWS CloudTrail

AWS CloudTrail veikia kaip auditavimo žurnalas. Šiame įrankyje galima atsekti kiekvieną AWS paskyroje atliktą veiksmą. Paieškos palengvinimui yra taikomi įvykių tipų, laiko intervalo bei naudotojo, atlikusio veiksmą, pasirinkimai.

CloudTrail renka duomenis iš visų AWS valdymo mechanizmų (grafinės sąsajos, programinės sąsajos, komandinės eilutės) [5]. Šis įrankis yra aktyvuojamas sukūrus Amazon paskyrą.

2.3.12. AWS Certificate Manager

AWS Certificate Manager yra naudojamas viešų bei privačių SSL bei TLS sertifikatų kūrimui bei priežiūrai. Tai sumažina naudotojo atsakomybę sertifikatų priežiūroje, kadangi sertifikatų pirkimu, aktyvavimu bei naujinimu rūpinasi Amazon Certificate Manager.

Naudotojas taip pat gali pasirinkti ir savo sertifikatus, tačiau jiems negalimas automatinis atnaujinimas Certificate Manager sistemoje. AWS Certificate Manager sertifikatai gali būti naudojami vienam, keliems ar „wildcard“ (sertifikatas, saugantis visus domeno subdomenus) domenų vardams.

2.3.13. AWS Detective

AWS Detective yra įrankis, ieškantis saugumo problemų priežasčių. Naudotojui tai palengvina paieškas, kadangi Detective naudoja mašininio mokymo algoritmus bei grafikus. Šis įrankis automatiškai renka duomenis bei įrašus iš kitų AWS resursų (CloudTrail įrašai, VPC tinklo įrašai). Tai padeda greičiau surasti saugumo rizikos ar jau egzistuojančios problemos kilmę. Tuomet galima tinkamai pritaikyti saugumo rizikos šalinimo strategiją.

2.3.14. AWS CloudWatch

AWS CloudWatch yra Amazon resursų ir naudotojo programų stebėjimo įrankis. Jis pats realiuoju laiku matuoja įvairių Amazon įrankių būklę bei agreguoja stebėjimo rezultatus. CloudWatch numatytieji nustatymai sukuria stebėjimo grafikus bei įvykių grupes kiekvienam naudojamam resursui.

Stebėjimo grafikai vizualiai pateikia būseną apie pasirinktą AWS resursą. Juose galima keisti laiko intervalą, grafiko tipą bei pasirinkti konkrečius resursus pagal pavadinimus.

Taip pat šis įrankis turi perspėjimų (angl. „alerts“) sistemą. Vartotojas pasirenka naudojamą resursą, matuojamą savybę bei jos ribines vertes. Pasiėkus ribinę vertę CloudWatch išsiunčia pranešimą.

2.3.15. AWS Macie

AWS Macie yra mašininio mokymo įrankis, aptinkantis jautrią informaciją. Iš esmės, tai yra duomenų nuskaitymo resursas, ieškantis saugumo spragų bei naudojantis šablonų atitikimo mechanizmu. Tai reiškia, jog Macie įrankis apmokomas atskirti jautrią informaciją (adresai, banko kortelės duomenys, asmens kodai, slaptažodžiai) pagal įvairius tokių duomenų pavyzdžius.

Dažniausiai duomenys yra laikomi AWS S3 talpykloje. Tuomet Macie ieško jautrios informacijos. Apie duomenų saugos spragas įrankis praneša generuodamas radinius. Kiekvienam radiniui yra priskiriamas pažeidžiamumo laipsnis. Galima sukurti automatizuotas taisykles tokių pažeidžiamumų sprendimui.

2.3.16. AWS Config

AWS Config yra įrankis, centralizuojantis paskyroje naudojamų AWS resursų konfigūracijas. AWS Config galime pamatyti, kaip resursai yra susiję tarpusavyje, kaip jie yra sukonfigūruoti bei kaip buvo konfigūruoti praeityje. Tai leidžia stebėti resursų tarpusavio sąveikas bei konfigūracijų pokyčius laike [4].

3. METODINĖ DALIS

Šioje dalyje aprašomas kurtas metodas bei jo seka.

3.1. Aplinkos kūrimas

Šiame skyrelyje aprašytas pirminės aplinkos kūrimas bei jam naudoti įrankiai. Toks aplinkos sukūrimas apibūdina mažiau apsaugotos aplinkos kūrimą.

3.1.1. Pasirinkti įrankiai

Tyrimo dalies aplinkos sukūrimui pasirinkta AWS debesija. Vienas iš esminių argumentų, lėmusių tokį pasirinkimą, yra gana stipriai akcentuojama saugumo sritis bei daugybė egzistuojančių saugumo įrankių. AWS periodiškai kuria naujus produktus, orientuotus į sistemų saugą, bei atnaušina jau esamus.

Taip pat AWS debesija kuria mokomuosius straipsnius tiek apie savo kuriamus įrankius, tiek apie įvairius saugumo aspektus (įskaitant ir Zero Trust taikymą).

Dalį AWS debesijos infrastruktūros resursų nuspręsta kurti Terraform kodo įrankiu, dalį nuspręsta kurti pasitelkiant grafinę AWS paskyros sąsają.

3.1.2. Tinklo dalies kūrimas

Tinklo dalies kūrimui panaudoti oficialūs Terraform moduliai, leidžiantys koncentruočiau aprašyti AWS VPC (izoliuotos tinklo dalies AWS debesijoje) resursą. AWS VPC tinklo konfigūravimas apima potinklį aprašymą, tinklo lentelį (angl. route table), NAT Gateway ir Internet Gateway kūrimą, IP adresų priskyrimą bei AWS VPC lygmens ugniasienės (angl. security group) taisyklių reguliavimą.

3.1.3. Prieigų ir teisių dalies konfigūravimas

Prieigos ir teisės sukurtos AWS IAM rolių bei taisyklių pagalba. Kiekvienam kuriamam komponentui priskiriamos tik funkcijos atlikimui reikalingos prieigos. Terraform įrankiu kai kuriems resursams automatiškai parenkami ir sukuriami teisių nustatymai.

3.1.4. Duomenų bazės kūrimas

Duomenų bazei panaudota AWS RDS duomenų bazių valdymo sistema. Pasirinkta AWS RDS PostgreSQL duomenų bazė. Šiai duomenų bazei pritaikomi AWS KMS numatytieji šifravimo raktai.

Atskirai yra kurtos bei konfigūruotos AWS RDS duomenų bazės ugniasienės taisyklės, leidžiančios užklausas iš konkrečių IP adresų ar kitų ugniasienės taisyklių grupių. Taip pat sukurti AWS RDS duomenų bazės parametrai.

3.1.5. Duomenų talpyklos kūrimas

Duomenų saugojimui sukurta AWS S3 talpykla. Šiam įrankiui nustatyta galimybė užklausas siųsti bei priimti per programinę sąsają. Taip pat sukonfigūruota AWS S3 ugniasienė.

3.1.6. Serverių kūrimas

Tyrimo programos paleidimui panaudotas AWS EKS Kubernetes klasteris su AWS EC2 virtualiomis mašinomis kaip mazgais. Mazgams parinkta Linux Ubuntu operacinė sistema. Prieigai prie AWS EKS klasterio pridėta *kubeconfig* konfigūracija. Taip pat palikta galimybė naudoti Systems Manager prisijungimą AWS debesijoje. EKS valdomajam bei vykdomiesiems mazgams priskirtos reikalingos IAM priegios. Šio žingsnio metu taip pat sukurta AWS EC2 virtuali mašina, skirta prisijungimui prie AWS RDS PostgreSQL duomenų bazės.

3.2. Duomenys bei programa

Dėl tyrimo pobūdžio jame panaudoti jokio aktualumo ir vertės neturintys duomenys (angl. dummy data). Dalis duomenų yra parinkta iš viešai prieinamų šaltinių, dalis duomenų sugeneruoti tyrimo autoriaus.

Tyrime naudojama programa taip pat yra realybėje nenaudojama bei jokios jautrios informacijos neapdorojanti programa. Ši programa skirta saugumo testavimams. Šis sprendimas pasirinktas dėl to, jog tyrimo metu bandyta įvairiais būdais apeiti AWS debesijoje taikomus standartus. Kadangi lygintos dvi skirtingos aplinkos (viena apsaugota tik numatytais parametrais, o kita pritaikant Zero Trust koncepciją), nebuvo rizikuota intelektinės nuosavybės vertę turinčiais duomenimis ar programa.

3.3. Zero Trust taikymas AWS aplinkai

Šiame skyrelyje aprašyta Zero Trust strategijos taikymo numatytajai AWS aplinkai metodologija.

3.3.1. Išplėstinė priežiūra ir teisių peržiūra

Paleidus testavimui skirtą programą bei duomenų bazę užpildžius testavimo duomenimis, peržiūrėti jau sukurti IAM AWS įgaliojimai. Kiekvienos taisyklės perteklinės teisės pašalintos, jog nebūtų palikta saugumo spragų.

3.3.2. Dirbtiniu intelektu grįstų saugumo įrankių taikymas

Pritaikyti tokie įrankiai kaip AWS Detective bei AWS Inspector. Šie įrankiai skirti esamų bei potencialių saugumo spragų identifikavimui. Įrankių funkcionalumu peržiūrėti virtualios mašinos tinklo nustatymai. Taip pat patikrintos naudojamos programinės įrangos versijos (ar nėra pasenusių).

Duomenų talpykloje laikomiems duomenims pritaikytas AWS Macie. Atliktas saugomų duomenų nuskaitymas, ieškant jautrios informacijos (asmens kodai, banko sąskaitos numeriai, slaptažodžiai).

3.3.3. Bendrinės AWS ugniasienės konfigūravimas

AWS WAF sukonfigūruota AWS paskyros lygmens ugniasienė. Šioje ugniasienėje pasirinkta apsauga nuo botų. Sudarytas patikimų bei nepatikimų IP adresų sąrašas.

Taip pat aktyvuotas AWS Shield. Šis įrankis pritaikytas apsaugai nuo sistemos trikdymo atakų (DDoS). Dėl efektyvesnio šio įrankio veikimo, jis integruotas su AWS WAF.

3.3.4. Sertifikatų ir raktų valdymas AWS įrankiuose

Naudojimui reikalingi sertifikatai sukurti AWS Certificate Manager įrankyje. Pasirinktas šių sertifikatų valdymas AWS debesijos pusėje. Taip užtikrintas automatinis atnaujinimas.

Šifravimo raktai yra naudojami ir saugomi AWS KMS įrankyje. Šių AWS debesijos įrankių naudojimas sumažina saugumo spragas, kadangi AWS prisiima visas saugos atsakomybes.

3.3.5. CloudTrail ir CloudWatch pritaikymas paskyros stebėjimui

CloudTrail sukurtos įvykių grupės, renkančios informaciją apie AWS paskyroje atliekamus veiksmus. Atskirti konkrečiai tyrimui naudojami resursai.

Taip pat sukurti CloudWatch perspėjimai CloudTrail įvykiams. Tai perspėja vartotoją apie veiksmus, kurie galimai nėra pageidaujami. CloudWatch įrankyje sukurtas stebėjimo grafikas, vizualiai pateikiantis, kiek įtartinų veiksmų pastebėta.

3.3.6. Centralizuoto stebėjimo centro sukūrimas

Patogesniam saugumo situacijos valdymui aktyvuotas AWS Security Hub kartu su AWS Config. Sukonfigūruoti automatiniai patikrinimai bei centralizuotas radinių atvaizdavimas. Taip pat integruoti trečiųjų šalių saugumo įrankiai.

3.3.7. Atsarginių kopijų valdymas

Atsarginės kopijos yra itin svarbi sistemos nepertraukiamo veikimo dalis. Nustatytas atsarginių kopijų kūrimo mechanizmas. Pritaikyti šifravimo raktai, užtikrinantys, jog atsarginės kopijos nebus panaudojamos be tinkamo rakto.

3.3.8. Tinklo segmentacija

Tinklo segmentacija yra tinklo išskaidymas į mažas dalis, kurios tarpusavyje gali būti sujungtos tik pritaikant saugumo garantijas. Tai užtikrina sistemos komponentų saugumą, kadangi patekus į vieną segmentą nebūtinai pavyks pereiti į kitą segmentą.

Sukurta tinklo išskaidymą atitinkanti metodologija. Tinklo segmentacijai panaudotos lokalaus lygio ugniasienės (angl. „security group“) bei privataus tinklo prieigos ugniasienės (angl. „network access control list“).

3.3.9. Turinio transliavimo tinklas

Turinio transliavimo tinklas (angl. „content delivery network“) apibūdina turinio pateikimą vartotojams iš skirtingų paskirstytų vietovių. Tai sumažina transliavimo trukmę, kadangi vartotoją turinys arba jo kopija (angl. „cache“) pasiekia greičiau dėl trumpesnio atstumo.

Panaudotas AWS Cloudfront turinio transliavimo tinklas, kuris ne tik pagerina turinio pasiekiamumą, bet ir prisideda prie puslapio apsaugojimo nuo sistemos trikdymo atakos. Atliktas programos srauto nukreipimas į AWS Cloudfront vietoje nukreipimo į užklausų paskirstymo komponentą, taip apsaugant IP adresus.

3.4. Sistemos veikimo bei saugumo tikrinimas

Šiame skyrelyje atlikti sistemos saugumo testavimai. Tam pasitelkti įvairūs problemines situacijas simuliuojantys įvykiai (kibernetinės atakos, neapdairios konfigūravimo klaidos, nutekėję prisijungimo duomenys).

3.4.1. Prisijungimas su nutekėjusiais duomenimis

Šiame patikrinime bandyta prisijungti prie AWS paskyros su nutekėjusiais duomenimis. Prisijungimas vykdytas naudojant kitą IP adresą. Pateikus prisijungimo užklausas, analizuoti AWS saugumo įrankių duomenys. Patikrintas CloudWatch perspėjimas.

3.4.2. Sistemos trikdymo ataka

Šiame patikrinime imituota sistemos trikdymo ataka pasitelkiant automatinį didelio užklausų kiekio siuntimą. Stebėta, kaip AWS WAF bei AWS Shield geba minimizuoti kibernetinės atakos riziką.

3.4.3. DNS užklausos iš ribojamų geolokacijų bei IP adresų

Šiame patikrinime pagal AWS WAF ugniasienėje sukonfigūruotas taisykles patikrintas užklausų į subdomeną blokavimas priklausomai nuo geolokacijos bei atrinktų IP adresų. Stebėta, koks atsakymas gaunamas kreipiantis į subdomeną bei kaip reaguoja AWS WAF ugniasienė.

3.4.4. RDS duomenų bazės pasiekiamumas iš išorės

Šiame patikrinime bandyta pasiekti AWS RDS duomenų bazę iš išorės. Šia tyrimo dalimi siekta patikrinti, ar užtenka nutekėjusių prisijungimo duomenų sėkmingam prisijungimui prie duomenų bazės, kuomet ji yra apsaugota lokalaus lygio ugniasiene (angl. „security group“).

3.4.5. Programinės sąsajos prieinamumo tikrinimas

Šiame patikrinime siųstos programinės sąsajos užklausos į įvairius AWS S3 resursus, bandant išgauti atsaką. Tikrinta AWS S3 talpyklos programinė sąsaja bei jos saugumas.

3.4.6. Perimti SSH raktai ir prisijungimas prie AWS EC2

Šiame patikrinime imituotas SSH raktų nutekėjimas bei bandymas su šiais raktais prisijungti iš išorėje tinklo esančios virtualios mašinos. Patikrinta lokalaus lygio ugniasienė.

3.4.7. Bandymas kompromituoti užklausų skirstytuvą

Šiame bandyme atliktas užklausų siuntimas į subdomeną, tikintis išgaut skirtingų klasterio mazgų IP adresus dėl užklausų skirstymo principo. Ši metodologija leidžia surasti saugumo spragas, kuomet klasterio mazgai yra lengvai aptinkami bei bandomi perimti.

3.4.8. Dirbtiniu intelektu grįstų įrankių radinių vertinimas

Šioje dalyje įvertinti AWS dirbtiniu intelektu grįstų įrankių radiniai bei pasiūlymai. Tai leidžia suprasti, kiek saugiai, anot dirbtinio intelekto, sukurta aplinka.

4. ĮGYVENDINIMO DALIS

Šioje dalyje pateikiama baigiamojo darbo tyrimo įgyvendinimo dalis. Aprašomi atlikti žingsniai bei jų seka.

4.1. Aplinkos kūrimas

Šioje tyrimo dalyje buvo atliktas AWS aplinkos paruošimas. Taip pat paruošti tyrimo vykdymui reikalingi įrankiai.

4.1.1. Pasirinktų įrankių paruošimas

Šiame žingsnyje buvo sukurta AWS debesijos paskyra bei aktyvuoti specifiniai resursai iš vartotojo pusės. Taip pat įrašyti Terraform, EMS Data Generator bei Lens įrankiai. Atlikti veiksmai:

1. AWS debesijos platformoje sukurta paskyra.
2. AWS paskyros nustatymuose iš vartotojo pusės aktyvuoti to reikalaujantys resursai (AWS Config, AWS Security Hub). Aktyvavimas reikalingas tam, kad naudotojas galėtų pagal savo poreikius konfigūruoti šiuos įrankius.
3. Paruošti Terraform paleidimo failai, skirti AWS resursų valdymui. Sukonfigūruotas AWS infrastruktūros būsenos (angl. „state“) saugojimas AWS S3 talpykloje.
4. Įrašytas bei paruoštas Kubernetes klasterio valdymo įrankis Lens. Šis įrankis pateikia grafinę sąsają Kubernetes klasterio valdymui (žr.). Tai reiškia, jog naudotojui nereikia vykdyti tiesioginių komandų kreipiantis į Kubernetes programinę sąsają.
5. Įrašytas EMS Data Generator įrankis. Šia programa atliktas AWS RDS duomenų bazės užpildymas testavimo duomenimis. Pasirinktas atsitiktinis bandymo duomenų generavimas, kadangi duomenų pobūdis nėra aktualus tyrimo eigai.

4.1.2. Tinklo dalies sukūrimas

Šiame žingsnyje sukurti pagrindiniai AWS debesijos tinklo komponentai (žr. 2 Priedas). Atlikti šie veiksmai:

1. Terraform HCL kalba aprašytas AWS VPC privataus tinklo resursas. Sukonfigūruoti privataus tinklo potinkliai. Pasirinkta naudoti trijų pasiekiamumo zonų potinklius. Taip pat sukurtos potinklių žymos, reikalingos Kubernetes klasterio komunikacijai. Atskirai sukonfigūruoti duomenų bazės, AWS EKS Kubernetes klasterio bei užklausų skirstytuvo (angl. „load balancer“) potinkliai. Terraform įrankiu sukurtas aprašytas AWS VPC debesijos resursas.
2. Aktyvuoti NAT Gateway bei Internet Gateway AWS VPC privačiame tinkle esančių resursų komunikacijai su išoriniais komponentais. Internet Gateway reikalingas, jog viešai pasiekiamuose potinkliuose esantys komponentai galėtų priimti užklausas iš interneto. NAT Gateway reikalingas, jog privačiuose potinkliuose esantys komponentai galėtų pasiekti už AWS VPC tinklo esančius resursus.
3. Išskirti AWS EKS klasterio mazgams reikalingi viešieji IP adresai.

4. Sukurtos AWS VPC lygmens ugniasienės bei jų taisyklės, leidžiančios ribotą užklausų siuntimą į privataus tinklo resursą (nustatyti patikimi IP adresai).

4.1.3. Prieigų ir teisių dalies konfigūravimas

Šiame žingsnyje aprašytas prieigų ir teisių konfigūravimas naudotojams bei programinėms sąsajoms (žr. 1 Priedas). Atlikti šie veiksmai:

1. Sukurta AWS IAM debesijos rolė, leidžianti Terraform įrankiui valdyti (kurti, pakeisti, ištrinti) resursus. Nesuteikus Terraform įrankiui tinkamų teisių, neįmanoma sukurti resursų iš Terraform paleidžiamųjų failų AWS debesijoje.
2. Aprašytos specifinio pobūdžio IAM taisyklės, leidžiančios kitiems AWS resursams atlikti tik jiems svarbią užduotį.
3. Daliai resursų pritaikytos kartu su AWS resursais kuriamos numatytosios prieigų teisės (AWS Inspector, AWS Detector).
4. Sukurtos AWS IAM rolės AWS CloudTrail, AWS GuardDuty bei AWS Macie komponentams. Šie resursai turi bendrauti su kitais (AWS CloudWatch ar AWS S3), todėl rolėmis apibrėžta ši galimybė.

4.1.4. Duomenų bazės kūrimas

Šiame žingsnyje aprašytas AWS RDS duomenų bazės konfigūravimas bei kūrimas (žr. 3 Priedas). Atlikti šie veiksmai:

1. Terraform įrankiu aprašyta kuriama AWS RDS PostgreSQL duomenų bazė. Pasirinktas duomenų bazės tipas, nusakantis kompiuterinių resursų galingumą. Sukonfigūruota *t3.medium* tipo duomenų bazė, turinti bendrinio naudojimo SSD tipo talpyklą.
2. Sukonfigūruota duomenų bazės parametrų grupė, apsprendžianti duomenų bazės savybes. Šioje dalyje pridėta galimybė saugoti prisijungimus prie duomenų bazės kaip įvykius.
3. Sukurtas AWS KMS duomenų bazės šifravimo raktas. Nustatytas rakto valdymas AWS debesijos pusėje. Šis raktų saugojimo mechanizmas perkelia atsakomybę AWS platformai, todėl naudotojui nereikia rūpintis kaip asmeniniame prietaise tinkamai apsaugoti šifravimo raktus.
4. AWS RDS lygmeniu sukonfigūruotos ugniasienės taisyklės specifiniams IP adresams bei prievadams. Sukurtos ugniasienės taisyklės, leidžiančios duomenų bazės potinklių ir AWS EKS potinklių komunikaciją.

4.1.5. Duomenų talpyklos kūrimas

Šiame žingsnyje sukurta AWS S3 duomenų talpykla (žr. 4 Priedas). Atlikti šie veiksmai:

1. Terraform įrankiu aprašyta standartinio tipo AWS S3 duomenų talpykla.
2. Aktyvuotas talpyklos pasiekiamumas per programinę sąsają.
3. Sukurtos AWS S3 lygmens ugniasienės taisyklės, išskiriant patikimus bei nepatikimus IP adresus.
4. Sukurtas AWS S3 leidžiamų užklausų sąrašas.

5. Sukonfigūruotas talpyklos versijavimas. Tai apsaugo talpinamus duomenis nuo praradimo ar netyčinio pakeitimo.

4.1.6. Serverio kūrimas

Šiame žingsnyje sukurtas bei sukonfigūruotas AWS EKS Kubernetes klasteris. Kaip mazgus pasirinkta naudoti AWS EC2 virtualias mašinas. Taip pat sukurta prisijungimui prie duomenų bazės skirta virtuali mašina (žr. 5,6 Priedai). Atlikti šie veiksmai:

1. Terraform įrankiu aprašytas AWS EKS klasteris, naudojantis Linux Ubuntu *t2.medium* tipo virtualias mašinas kaip mazgus.
2. Sukonfigūruotas laikino panaudojimo (išnykstančios po virtualios mašinos išjungimo ar sustabdymo) talpyklų kūrimas klasterio mazgams.
3. Sukurtos teisės tiek valdančiajam mazgui, tiek vykdančiųjų mazgų grupei, leidžiančios atlikti AWS paskyros pokyčius virtualioms mašinoms (sustabdyti, sukurti, išjungti).
4. Sukurta bei priskirta išplečiamumo grupė (angl. autoscaling group). Šis komponentas reikalingas tam, jog esant dideliame apkrovimui būtų sukurti papildomi vykduojantieji mazgai.
5. AWS EKS Kubernetes klasteris pridėtas į Lens valdymo įrankį. Prisijungimas prie EKS klasterio atliktas komanda `aws eks update-kubeconfig --region eu-west-1 --name prod-zero-trust-cluster`.
6. Į AWS EKS Kubernetes klasterį įrašyti DNS bei tinklo plėtiniai (coreDNS, kube-proxy, VPC-CNI). Šie plėtiniai užtikrina sklandžią Kubernetes klasterio komunikaciją.
7. Sukonfigūruotas nginx valdiklis, naudojamas užklausių į programas skirstymui. Šis valdiklis yra saugesnis nei alternatyvūs, kadangi užtikrina saugumą ketvirtame (transporto) bei septintame (programų) OSI tinklo sluoksniuose.
8. Taip pat sukonfigūruotas ir į AWS EKS klasterį įrašytas sertifikatų valdiklis, atsakingas už sukurtų sertifikatų priskyrimą bei valdymą.
9. Papildomai sukonfigūruota bei sukurta AWS EC2 virtuali mašina, naudota prisijungimui prie AWS RDS PostgreSQL duomenų bazės.

4.2. Duomenys bei programa

Šioje tyrimo dalyje atliktas duomenų generavimas bei saugojimas kurtose talpyklose. Taip pat virtualioje mašinoje įrašyta tyrimui skirta programa.

4.2.1. Programos įrašymas

Šiame žingsnyje į AWS EKS Kubernetes klasterį įrašyta tyrimui naudota programa. Parinkta minimalistinė programa, neturinti daug funkcionalumo, kadangi tyrimo metu tikrinami saugos parametrai.

Pasirinkta programa „whoami“, spausdinanti operacinės sistemos, iš kurios atliekama užklausa, informaciją. Pradžioje buvo atliktas testavimas paleidžiant programą lokaliai. Taip pat

AWS Route53 domenų valdymo įrankyje sukurta zero-trust-karolis.com DNS zona. Atlikti šie žingsniai (žr. 7,8 Priedai):

1. YAML kalba sukurtas „whoami“ programos paleidimo failas (angl. „manifest“). Šios programos veikimas patikrintas lokaliai kompiuterio naršyklėje (žr. 4.2.1.2 pav.).
2. YAML kalba aprašyti programos tinklo komponentų (Kubernetes Service ir Kubernetes Ingress) paleidimo failai. Programa ir tinklo komponentai įrašyti į Kubernetes klasterį naudojant komandą `kubectl apply -f .` YAML paleidžiamųjų failu aplanke. Komponentai patikrinti Lens įrankyje grafiniu būdu (žr. 4.2.1.1 pav.).
3. Aktyvuotas zero-trust-karolis.com domenas (žr. 4.2.1.3 pav.).
4. Domenui zero-trust-karolis.com DNS A įrašu sukurtas subdomenas whoami.zero-trust-karolis.com, kuris tinklo užklausas nukreipia į tinklo užklausų skirstytuvą (žr. 4.2.1.3 pav.).
5. Paleidžiamuoju YAML failu subdomenui sukurtas sertifikatas. Šį veiksmą atliko sertifikatų valdiklis.
6. DNS zonoje sukurtas įrašas, reikalingas SSL sertifikato validavimui.
7. Adresu whoami.zero-trust-karolis.com patikrintas programos bei sertifikato veikimas (žr. 4.2.1.4 pav.).

4.2.2. Duomenų generavimas bei talpinimas

Šiame žingsnyje dalis duomenų buvo generuoti automatinio būdu, dalis pakoreguota rankiniu būdu. Kadangi naudojama PostgreSQL duomenų bazė, pasirinktas jų numatytasis duomenų generavimo įrankis EMS Data Generator. Šiam bandymui parinkti neaktualūs duomenys, neturintys jokios realios jautrios informacijos. Tyrime naudojami duomenys patalpinti AWS RDS PostgreSQL duomenų bazėje bei AWS S3 talpykloje. Atlikti šie veiksmai:

1. Patikrintas prisijungimas prie duomenų bazės naudojant `psql -h prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com -p 5432 -U postgres zerotrust` komandą, kurioje nurodyti AWS RDS duomenų bazės adresas, prievadas, besijungiančiojo naudotojo vardas ir PostgreSQL naudojama schema.
2. Tam, kad duomenų automatinis generavimas būtų įvykdytas korektiškai, rankiniu būdu sukurta automatiniam generavimui skirta schema bei lentelė.
3. EMS Data Generator įrankyje parinkti atsitiktinio generavimo nustatymai AWS RDS duomenų bazės duomenims.
4. Vykdam automatinį EMS Data Generator užklausų paleidimą duomenų bazė užpildyta skaitiniais (INTEGER) bei tekstiniais (VARCHAR(255)) duomenimis (žr. 4.2.2.1 pav.).
5. Duomenys, talpinti į AWS S3 talpyklą, taip pat generuoti automatinio būdu. Šiuos duomenis sudaro vardo, pavardės, gimimo datos, asmens kodo, banko sąskaitos IBAN bei turimų santaupų sąskaitoje informacija (žr. 4.2.2.2 pav.). Toks duomenų formatas pasirinktas dėl poreikio turėti panašius į jautrią informaciją duomenis (asmens kodas, banko sąskaita).
6. Sugeneruoti duomenys patalpinti į AWS S3 talpyklą kaip .csv failas.

4.3. Zero Trust taikymas AWS aplinkai

Šioje tyrimo dalyje atliktas sukurtos AWS sistemos saugumo gerinimas pritaikant Zero Trust koncepciją.

4.3.1. Išplėstinė prieigų ir teisių peržiūra

Šioje tyrimo dalyje atlikta nuodugni IAM prieigų ir teisių peržiūra (žr. 11 Priedas). Atlikti šie veiksmai:

1. Peržiūrėtos bei pakoreguotos IAM teisės ir prieigos remiantis mažiausio reikalingumo principu (angl. „least privilege“).
2. Panaikintos tiesiogiai su naudotojais susietos teisės (angl. „attached“).
3. Sukurtas naujas bandymams skirtas naudotojas (žr. 4.3.1.1 pav.).
4. Sukurta IAM naudotojų grupių bei rolių sistema. Reikalingos IAM teisės priskirtos sukurto IAM rolėms. Tuomet naudotojai priskirti prie IAM naudotojų grupių, turinčių teisę naudotis konkrečiomis IAM rolėmis. Naujas naudotojas priskirtas prie naudotojų grupės (žr. 4.3.1.2 pav.).
5. Aktyvuota kelių patvirtinimų autentifikacija, kuomet prisijungiant prie paskyros naudotojo prašoma patvirtinti savo tapatybę dar vienu pasirinktu būdu (SMS žinute ar mobilia programėle) (žr. 4.3.1.3 pav., 4.3.1.4 pav.).

4.3.2. Dirbtiniu intelektu grįstų saugumo įrankių taikymas

Šiame tyrimo žingsnyje atliktas dirbtiniu intelektu grįstų saugos įrankių konfigūravimas. Šie įrankiai skirti esamų bei potencialių saugumo spragų identifikavimui. Sukonfigūruoti AWS Detective, AWS Inspector, AWS GuardDuty bei AWS Macie resursai (žr. 9 Priedas). Atlikti šie žingsniai:

1. Sukonfigūruotas AWS Detective. Matuotas IAM rolių ir naudotojų kreipimosi į programinę sąsają aktyvumas. Taip pat stebėta vykdoma komunikacija su AWS EC2 virtualiomis mašinomis (Kubernetes klasterio mazgai bei prisijungimo prie duomenų bazės virtuali mašina).
2. Sukonfigūruotas AWS Inspector (žr. 4.3.2.1 pav., 4.3.2.2 pav.). Patikrinti potencialūs kompiuterinių resursų pažeidžiamumai. Atrasti pažeidžiamumai, susiję su AWS EC2 virtualios mašinos šablono programine įranga.
3. Aktyvuotas AWS GuardDuty. Šiuo įrankiu sukonfigūruotas AWS S3 talpyklos, AWS RDS duomenų bazės bei AWS EKS Kubernetes klasterio saugojimas. Atlikti patikrinimai parodė, jog į vieną iš virtualių mašinų buvo bandyta prisijungti iš Kinijos naudojant SSH protokolą. Radinys AWS GuardDuty detaliai aprašytas bei pasiūlytas sprendimo būdas.
4. Sukonfigūruotas AWS Macie (žr. 4.3.2.3 pav.). Patikrinti AWS S3 talpyklos resursai, kuriuose patalpinti bandymui sugeneruoti duomenys. Panašūs į jautrią informaciją duomenys buvo įvertinti kaip jautri, tačiau ne per jautri informacija.

4.3.3. Bendrinės AWS ugniasienės konfigūravimas

Šioje dalyje atliktas AWS WAF programos lygmens ugniasienės konfigūravimas. Atlikti šie veiksmai:

1. Aktyvuotas AWS WAF ugniasienės resursas.
2. Sukonfigūruota programos lygmens apsauga nuo botų. Taip programa apsaugota nuo per didelio kompiuterinių resursų naudojimo ar sistemos trikdymo.
3. Pritaikytas užklausų blokavimas, kuomet negalima identifikuoti siuntėjo (Tor naršyklės naudojimas, VPN) (žr. 4.3.3.4 pav.).
4. Aktyvuota ugniasienės taisyklė, besiremianti AWS platformos surinkta informacija apie nepatikimus IP adresus. Žemos reputacijos IP adresai tokiu atveju blokuojami.
5. Sukurtas rankinis blokuotinų IP adresų sąrašas (žr. 4.3.3.1 pav.).
6. Sukonfigūruota ugniasienės taisyklė, blokuojanti užklausas iš specifinių valstybių (Kinija, Rusija, Singapūras) ir rankinio IP adresų sąrašo (žr. 4.3.3.2 pav., žr. 4.3.3.3 pav.).

4.3.4. Sertifikatų ir raktų valdymas AWS įrankiuose

Šiame žingsnyje įgyvendintas šifravimo raktų bei domeno sertifikato saugojimas AWS debesijos resursuose AWS Certificate Manager bei AWS KMS (žr. 10 Priedas). Atlikti šie veiksmai:

1. Programos domenui naudojamas SSL sertifikatas sukurtas AWS Certificate Manager (žr. 4.3.4.1 pav., 4.3.4.2 pav.).
2. Sukonfigūruotas automatinis SSL sertifikato atnaujinimas whoami.zero-trust-karolis.com domenui.
3. AWS KMS sukurtas šifravimo raktas, reikalingas AWS RDS duomenų bazės duomenims šifruoti (žr. 4.3.4.3 pav., 4.3.4.4 pav.).
4. Sukonfigūruotas AWS KMS šifravimo raktas, skirtas įslaptinti AWS CloudTrail radinius.

4.3.5. CloudTrail ir CloudWatch pritaikymas paskyros stebėjimui

Šiame žingsnyje praplėstas numatytasis CloudTrail bei CloudWatch funkcionalumas. CloudTrail resursas pasitelktas veiksmų AWS paskyroje stebėjimui, o CloudWatch resursas panaudotas surinktų duomenų apie AWS resursus stebėjimui ir vizualizavimui (žr. 14 Priedas). Atlikti šie veiksmai:

1. Sukonfigūruota AWS CloudTrail įvykių grupė, stebinti tik tyrime naudojamus komponentus (žr. 4.3.5.1 pav.).
2. Sukonfigūruoti CloudWatch perspėjimai (angl. „alerts“) praeitame veiksmo aprašytos CloudTrail grupės įvykiams.
3. Sukurtas stebėjimo grafikas CloudTrail įvykiams, leidžiantis vizualiai stebėti aptiktus įtartinus veiksmus AWS paskyroje.

4. CloudWatch įrankiu sukurti stebėjimo grafikai tyrime naudojamiems resursams (AWS RDS, AWS S3, AWS EC2), matuojantys šių įrankių technines charakteristikas (žr. 4.3.5.2 pav.).

4.3.6. Centralizuoto stebėjimo centro sukūrimas

Šiame žingsnyje sukonfigūruotas centralizuotas saugumo stebėjimo įrankis, kuriame gauta informacija iš įvairių skirtingų šaltinių. AWS Security Hub panaudotas kaip kertinis šio metodo resursas. Atlikti šie veiksmai:

1. Paruošti AWS SecurityHub bei AWS Config įrankiai (žr. 4.3.6.1 pav.). Sukonfigūruotas įvykių siuntimas iš AWS Config į AWS S3 talpyklą bei SecurityHub (žr. 4.3.6.2 pav.).
2. Sukonfigūruoti patikrinimai pagal saugumo standartus (CIS AWS Foundations Benchmark, PCI DSS v3.2.1 ir NIST 800-53).
3. Panaudota integracija su trečiųjų šalių saugumo įrankiu 3CORESec.
4. Sukurtas radinių bei patarimų grafikų ir įvykių patogus atvaizdavimas.

4.3.7. Atsarginių kopijų valdymas

Šiame žingsnyje sukurtas atsarginių kopijų kūrimo bei saugojimo mechanizmas. Taip pat įgyvendintas atsarginių kopijų šifravimas, užtikrinantis duomenų saugumą net ir perėmimo metu. Atlikti šie veiksmai:

1. Sukonfigūruotas atsarginių kopijų kasdienis kūrimas AWS RDS duomenų bazei.
2. Nustatytas AWS RDS duomenų bazės atsarginės kopijos saugojimo laikas (trys dienos).
3. Sukurta AWS S3 talpyklos atsarginė kopija (žr. 4.3.7.1 pav.).
4. Sukonfigūruotas AWS EC2 virtualių mašinų automatinis atsarginių kopijų kūrimas (žr. 4.3.7.2 pav.).
5. Sukurtas AWS KMS šifravimo raktas, panaudotas atsarginių kopijų duomenų įslaptinimui.

4.3.8. Tinklo segmentacija

Šioje dalyje užtikrinta tinklo segmentacija, arba kitaip tinklo išskaidymas. Tinklo segmentacijai panaudotos AWS resursų lygmens ugniasienės. Atlikti šie veiksmai:

1. AWS VPC privataus tinklo potinkliams sukonfigūruotos tinklo prieigos ugniasienės. Taip sukurtas segmentas, izoliuojantis potinklius.
2. AWS EKS Kubernetes klasteriui bei AWS RDS duomenų bazei įvykdyta segmentacija lokalaus lygio ugniasienėmis. Sukonfigūruotas tinklo užklauso praleidimas tik tuo atveju, jei užklausa praleidžia ir kito resurso ugniasienė.
3. Sukonfigūruoti AWS S3 talpyklos patikimų IP adresų sąrašai. AWS S3 talpykla priima užklauso programine sąsaja tik iš patikimų IP adresų.

4.3.9. Turinio transliavimo tinklas

Šioje dalyje įgyvendintas turinio transliavimo tinklas. Sukurta CloudFront distribucija. Taip pat įvykdytas užklausų kreipimas iš subdomeno į CloudFront. Atlikti šie veiksmai:

1. Sukurta CloudFront distribucija turinio transliavimui (žr. 4.3.9.1 pav.).
2. Sukonfigūruotas DNS A įrašas užklausų nukreipimui į CloudFront (žr. 4.3.9.2 pav.).
3. Nustatytas užklausų kreipimas į užklausų skirstytuvą (angl. „load balancer“).

4.4. Sistemos veikimo bei saugumo tikrinimas

Šioje tyrimo dalyje atlikti sistemos saugumo testavimai. Simuliuoti panašūs į realius scenarijai, kuomet išbandomas sistemos saugumas. Sistemos saugumas patikrintas atliekant kibernetines atakas (užkoduotos DNS užklauskos siuntimas, sistemos trikdymo ataka), bandant pasiekti AWS infrastruktūrą su naudotojo prisijungimais bei tikrinant programinės sąsajos prieinamumą (AWS S3).

Daliai bandymų atlikti pakartotiniai testavimai be sukurtos Zero Trust koncepciją atitinkančios saugumo strategijos.

4.4.1. Prisijungimas su nutekėjusiais duomenimis

Šioje tyrimo dalyje buvo panaudoti papildomai sukurti naudotojo prisijungimo prie AWS paskyros duomenys. Situacija imituotas atvejis, kuomet darbuotojas atsainiai rūpinasi savo duomenų sauga. Dėl taikomos Zero Trust koncepcijos, naudotojas privalėjo sukurti kelių patvirtinimų prisijungimą (žr. 17 Priedas). Atlikti šie veiksmai:

1. Sukurtas naujas naudotojas bei jo prisijungimo duomenys.
2. Šis naudotojas priskirtas prie jam sukurtos naudotojų grupės.
3. Aktyvuotas kelių patvirtinimų prisijungimas (MFA).
4. Kitu prietaisu bandyta prisijungti prie AWS paskyros (žr. 4.4.1.1 pav., 4.4.1.2 pav.).
5. Prisijungimas nepavyko nepaisant teisingų prisijungimo duomenų dėl kelių patvirtinimų autentifikacijos (žr. 4.4.1.3 pav.).
6. AWS CloudTrail gauti pranešimai apie bandymą jungtis iš naujo IP adreso nepateikiant teisingo MFA autentifikavimo.

Taip pat bandyta prisijungti su nutekėjusiais duomenimis prie AWS paskyros, kuomet sistemos apsauga neatitinka darbo metu sukurtos saugumo strategijos (žr. 26 Priedas). Atlikti šie veiksmai:

1. Bandyta prisijungti tuo pačiu asmeniniu prietaisu prie AWS paskyros (žr. 4.4.1.4 pav.).
2. Prisijungimas pavyko, kadangi prisijungimo duomenys buvo vienintelis saugos barjeras (žr. 4.4.1.5 pav.).
3. Bandyta išplėsti perimto naudotojo IAM teisės, jog būtų padaryta daugiau žalos.
4. Naudotojas, kurio prisijungimais įsilaužta, neturėjo tam teisių (žr. 4.4.1.6 pav.).
5. Numatytoji praktika yra neskirti administratoriaus prieigų jokiame naudotojui, tačiau perimtos paskyros esamų prieigų gali pakakti žalos darymui.

4.4.2. Sistemos trikdymo ataka

Šiame žingsnyje atliktas mažo intensyvumo sistemos trikdymo imitavimas. Užklausų siuntimui panaudotas Apache Bench įrankis (žr. 18 Priedas). Atlikti šie veiksmai:

1. Įrašytas Apache Bench įrankis.
2. Kelis kartus komanda `ab -n 5000 -c 100 https://whoami.zero-trust-karolis.com/` imituota sistemos trikdymo ataka. Pirmas parametras nusakė siunčiamų užklausų kiekį, o antras parametras nusakė lygiagretinimo laipsnį (kiek vienu metu išsiunčiama užklausų).
3. Užklausų siuntimo metu, tikrintas programos veikimas. Trikdžių nepastebėta.
4. AWS WAF blokavo beveik visas užklausas didžiąją jų dalį klasifikuojant kaip įterptinių instrukcijų atakas (angl. „Cross site scripting“) (žr. 4.4.2.1 pav., 4.4.2.2 pav., 4.4.2.3 pav.).

Taip pat bandyta trikdyti sistemą šia ataka, kuomet subdomenas neatitinka sukurtos saugumo strategijos (žr. 26 Priedas). Atlikti šie veiksmai:

1. DNS zonoje sukurtas A įrašas srauto nukreipimui tiesiai į užklausų skirstytuvą (žr. 4.4.2.4 pav.).
2. Komanda `ab -n 5000 -c 100 -s 300 https://whoami.zero-trust-karolis.com/` imituota sistemos trikdymo ataka neapsaugotai sistemai. Pridėtas parametras, nusakantis maksimalų laukimo laiką prieš nutraukiant susijungimą.
3. Užklausų siuntimo metu Google Chrome naršyklėje tikrintas programos atsakymo laikas. Atakos metu atsakymo greitis sulėtėjo nuo 60 ms iki maždaug 200 ms (žr. 4.4.2.5 pav.).
4. Pastebėta, jog prastos užklauskos užkemša tinklą, todėl net ir itin minimalistiškos programos atsakas ilgėja.

4.4.3. DNS užklauskos iš ribojamų geolokacijų bei IP adresų

Šioje tyrimo dalyje atliktas užklausų siuntimas į subdomeną `whoami.zero-trust-karolis.com` iš apribotos geografinės vietovės bei apriboto IP adreso. Patikrintas programos pasiekimas per ugniasienę. Atlikti šie veiksmai:

1. Įrankiu Proxyium, suteikiančiu galimybę pasinaudoti Singapūro geografinė vietovė, siųsta užklausa į subdomeną (žr. 4.4.3.1 pav.).
2. Ugniasienė nepriėmė užklauskos dėl to, jog užklausa siunčiama iš geografiškai apribotos valstybės (žr. 4.4.3.2 pav.).
3. Asmeniniu kompiuteriu išsiųsta užklausa į programos subdomeną.
4. Ugniasienė atmetė užklauską, kadangi asmeninio kompiuterio IP adresas įtrauktas į blokuotinų adresų sąrašą (žr. 4.4.3.3 pav.).

Taip pat bandyta pasiekti subdomeną iš Singapūro geografinės vietovės, kuomet subdomenas neatitinka sukurtos saugumo strategijos. Atlikti šie veiksmai:

1. Dar kartą įrankiu Proxyium, suteikiančiu galimybę pasinaudoti Singapūro geografinė vietovė, siųsta užklausa į subdomeną (žr. 4.4.3.4 pav.).
2. Užklausa buvo priimta, kadangi nėra jokių ugniasienės taisyklių (žr. 4.4.3.5 pav.).

3. Tokiu būdu programa prarado filtrą siunčiamoms užklausoms.

4.4.4. RDS duomenų bazės pasiekiamumas iš išorės

Šioje tyrimo dalyje bandyta pasiekti AWS RDS duomenų bazę iš išorės. Šioje situacijoje imituotas duomenų bazės prisijungimo duomenų nutekėjimas bei bandymas šiuos duomenis išnaudoti. Atlikti šie veiksmai:

1. Pridėtas virtualios mašinos IP į lokali ugniasienės taisyklę (žr. 4.4.4.1 pav.).
2. Naudojant `psql` komandą bandyta prisijungti prie duomenų bazės iš virtualios mašinos. Prisijungimas pavyko, kadangi lokali ugniasienė leidžia nurodytos virtualios mašinos IP adresą (žr. 4.4.4.2 pav.).
3. Analogiškas prisijungimas atliktas asmeniniame įrenginyje. Prisijungimas atmestas, kadangi lokali ugniasienė, nepaisant teisingų prisijungimo duomenų, nelaiko IP adresą patikimu (žr. 4.4.4.3 pav.).

Taip pat bandyta pasiekti AWS RDS duomenų bazę iš išorės, kuomet ji neatitinka sukurtos saugumo strategijos. Atlikti šie veiksmai:

1. Naudojant `psql` komandą su perimtais duomenų bazės prisijungimo duomenimis vykdytas prisijungimas. Prisijungimas iš asmeninio prietaiso pavyko (žr. 4.4.4.4 pav.).
2. Tai atvėrė kelią komandų vykdymui duomenų bazėje, kas leido pasiekti joje esančius duomenis (4.4.4.5 pav.).

4.4.5. Programinės sąsajos prieinamumo tikrinimas

Šiame žingsnyje patikrintas programinių sąsajų prieinamumas. Taip bandyta atrasti saugumo spragą, leidžiančią pasinaudoti programine sąsaja (žr. 21 Priedas). Atlikti šie veiksmai:

1. Naudojant `curl` komandą tikrintas AWS S3 talpyklos programinės sąsajos pasiekiamumas.
2. Gautas rezultatas „Access denied“, reiškiantis apsaugotą programinę sąsają.
3. Naudojant tą pačią komandą kreiptasi į AWS EC2 privatų adresą. Kadangi prietaisas, iš kurio kreiptasi, nėra tinklo dalis, užklausos buvo atmestos.
4. Rezultatas patvirtino, jog lokali ugniasienės svarbios ne tik tiesiogiai programos vykdyme dalyvaujantiems komponentams, bet ir atskirai veikiančioms talpykloms.

Taip pat bandyta pasiekti AWS S3 talpyklą iš išorės, kuomet ji neatitinka sukurtos saugumo strategijos. Atlikti šie veiksmai:

1. Naudojant `curl` komandą tikrintas AWS S3 talpyklos programinės sąsajos pasiekiamumas.
2. Gauta išvestis apie S3 talpyklą, jos konfigūracijas bei joje esančių objektų informacija (žr. 29 Priedas).
3. Pabandyta pasiekti bandymui sukurtus duomenis, tačiau to padaryti nepavyko (žr. 4.4.5.1 pav.). Numatytoji apsauga, jei nenurodyta kitaip, neleidžia bet kam iš išorės nuskaityti failų.

4. Rezultatas rodė, jog failai nebus pasisavinti ar pakeisti, tačiau apie juos galima surinkti daug informacijos, kuri būtų panaudota tolesniam saugumo spragų ieškojimui.

4.4.6. Perimti SSH raktai ir prisijungimas prie AWS EC2

Šioje tyrimo dalyje imituojama situacija, kai yra perimami SSH raktai, skirti prisijungimui prie AWS EC2 virtualios mašinos. Buvo atlikti šie veiksmai:

1. Sukurti SSH raktai prisijungimui prie virtualios mašinos.
2. Raktai įrašyti į asmeninį prietaisą.
3. Bandyta jungtis prie virtualios mašinos naudojant perimtus SSH raktus.
4. Prisijungimas nepavyko dėl lokalsienės veikimo (žr. 4.4.6.1 pav.).

Taip pat bandyta pasiekti AWS EC2 virtualią mašiną iš išorės su perimtais SSH raktais, kuomet ji neatitinka sukurtos saugumo strategijos. Atlikti šie veiksmai:

1. Su jau įrašytais SSH raktais bandyta prisijungti prie virtualios mašinos.
2. Prisijungimas pavyko dėl to, jog numatytuoju nepatyrusių naudotojų atveju virtualioms mašinoms gan dažnai yra paliekama atvira komunikacija su SSH.
3. Viduje EC2 buvo galima vykdyti komandas (žr. 4.4.6.2 pav.).

4.4.7. Bandymas kompromituoti užklausų skirstytuvą

Šiame bandyme siekta išgauti skirtingų klasterio mazgų adresus dėl veikiančio užklausų skirstymo principo. Atlikti šie veiksmai:

1. Siųsta `ping` komandos užklausa į `whoami.zero-trust-karolis.com` (žr. 4.4.7.1 pav.).
2. Komandos išvestis gražino CloudFront distribucijos adresą. Užklausų skirstytuvą paslėptas po šiuo adresu.
3. Pakartotinė užklausa gražina tą patį adresą.
4. Išgauti klasterio mazgų adresų nepavyko.

Šiame bandyme atliktas užklausų siuntimas į subdomeną, tikintis išgauti skirtingų klasterio mazgų IP adresus dėl užklausų skirstymo principo. Ši metodologija leidžia surasti saugumo spragas, kuomet klasterio mazgai yra lengvai aptinkami bei bandomi perimti.

4.4.8. Dirbtiniu intelektu grįstų įrankių radinių vertinimas

Šioje tyrimo dalyje įvertintos sukonfigūruotų dirbtinio intelekto įrankių surastos saugumo spragos bei pasiūlymai. Atlikti šie veiksmai:

1. Patikrintos SecurityHub rastos grėsmės. Rastos kelios aukštos grėsmės Python kalbos bibliotekos numatytuoju būdu įrašytos į AWS EC2 virtualias mašinas (žr. 4.4.8.1 pav.). Taip pat nustatyta, jog AWS IAM kuriant slaptažodžius nėra privalu naudoti simbolius bei skaičius, todėl ši rizika priklauso nuo naudotojų sąmoningumo (žr. 4.4.8.2 pav.).
2. AWS Detective stebėta, kaip kito AWS programinės sąsajos užklauskos. Tai leido pastebėti, kuris naudotojas ar rolė atliko daugiau užklausų AWS viduje (žr. 4.4.8.3 pav.). Labiausiai naudojama rolė buvo ta, kuria buvo valdoma AWS grafinė sąsaja.

Taip pat daug užklausų atliko Kubernetes klasterio mazgai. Taip pat stebėtas duomenų srautas AWS EC2 virtualiose mašinose (žr. 4.4.8.4 pav.).

3. AWS Inspector atrasti panašūs Python bibliotekų pažeidžiamumai. Tačiau šis įrankis leido aptikti ir gilesnių programinės įrangos numatytųjų saugumo spragų (žr. 4.4.8.5 pav.).
4. AWS Macie įvertino bandymui skirtus duomenis, kaip jautrius, bet nepakankamai, jog reiktų dėl to rūpintis (žr. 4.4.8.6 pav.). Tokį mažą rizikos įvertį galėjo sukelti mažas informacijos įrašų skaičius arba tai, jog informacija buvo pasikartojanti.

5. REZULTATAI IR APIBENDRINIMAS

Darbo metu sukurta visapusiškai veikianti ir į nulinio pasitikėjimo principus atsižvelgianti saugumo strategija, pritaikyta AWS debesijai. Suprojektuotas atsparios sistemos modelis, kuomet atsarginės kopijos kuriamos aprėpiant tiek programos, tiek duomenų bazės, tiek talpyklos lygmenis. Įgyvendintas atsarginių kopijų šifravimas, jog duomenys nebūtų panaudojami perėmimo metu. Suprojektuota keliais sluoksniais veikianti paskyros ugniasienė. Sukonfigūruota programos lygmens AWS WAF ugniasienė ir jos taisyklės, o AWS resursų lygmenyje įgyvendintos lokalaus lygio ugniasienės. Lokalių ugniasienių integravimu sukurta tinklo mikrosegmentacija, kai tinklo srautas iš vieno komponento į kitą net ir viduje sistemos patenka tik patikrintas papildomų taisyklių. Taip pat sukurtas paskyros prisijungimų stebėjimo mechanizmas, kuomet neįvykdžius kelių patvirtinimų autentifikacijos, sukuriama peržiūrimi įvykiai. Suprojektuotas centralizuotas AWS infrastruktūros stebėjimo centras, panaudojant AWS Config, AWS SecurityHub, AWS GuardDuty. Į šį stebėjimo centrą integruoti sukonfigūruoti dirbtinio intelekto ir mašininio mokymo pagrindais veikiantys įrankiai, ieškantys saugumo spragų esamoje sistemoje. Tai yra daroma atsižvelgiant į programinę įrangą, tinklo srautus bei užklausas, konfigūravimo klaidas ir AWS programinės sąsajos panaudojimą. Šis visą sistemą apimantis saugumo sprendimas atitinka Zero Trust koncepciją – sistema nuolat laikoma nepatikima ir pastoviai ieškoma, kaip pagerinti esamą saugumo situaciją. Zero Trust neturi aiškių išmatuojamų tikslų, todėl tai yra labiau vizija. Galima teigti, jog šiame baigiamajame darbe įgyvendintas saugumo sprendimas, kuris saugumo sferoje eina koja kojon kartu su nulinio pasitikėjimo saugos debesijoje vizija. Tai galima patvirtinti ir testavimo scenarijais bei jų rezultatais tiek apsaugotai aplinkai, tiek neapsaugotai aplinkai (žr. 2.2 Lentelė).

2.2 Lentelė. Zero Trust modelio privalumai bei iššūkiai.

Testavimo scenarijus	Apsaugota aplinka	Neapsaugota aplinka
Perimti SSH raktai	Nepavyko prisijungti. Tinkamai sukonfigūruotos lokalios ugniasienės.	Pavyko prisijungti.
S3 programinės sąsajos tikrinimas	Nepavyko prisijungti.	Dalinai (gauta informacija apie talpykloje esančius objektus, bet nenuskaitytas pats failas)
RDS pasiekiamumas iš išorės su duomenų bazės prisijungimais	Nepavyko prisijungti. Tinkamai sukonfigūruotos lokalios ugniasienės.	Pavyko prisijungti.
DNS užklausų siuntimo į subdomeną ribojimas pagal geografines vietas ar IP sąrašus	Užklausos atmestos.	Nėra galimybės blokuoti tokių užklausų.
Sistemos trikdymo ataka	Didžioji dalis užklausų blokuotos. Neturėjo įtakos pasiekiamumui.	Priimta didžioji dalis užklausų. Sukeltas laikinas sistemos trikdymas arba sulėtinimas.
Prisijungimas prie AWS paskyros su nutekintais duomenimis	Nepavyko prisijungti dėl kelių patvirtinimų autentifikavimo.	Pavyko prisijungti.

Testavimo scenarijų rezultatuose išryškėjo aiškus infrastruktūros, apsaugotos šio baigiamojo darbo metu kurta sistema, pranašumas. Rezultatai parodė, jog saugumo strategija, koreliuojanti su nulinio pasitikėjimo koncepcija, net ir po žmogiškosios klaidos išliko saugi. Tuo tarpu neapsaugotoje infrastruktūroje bandymų metu žmogaus klaidos nulėmė dar didesnę duomenų nutekėjimą, intelektinės nuosavybės praradimą ar sistemos sutrikdymą.

Šio darbo metu nustatyta, jog lokalaus lygmens ugniasienės yra kertinis faktorius apsaugant savo debesijos infrastruktūrą. Ši koncepcija ne tik leidžia išskaidyti tinklą į mažesnius, saugesnius segmentus, bet ir užtikrina AWS resursų apsaugą (EC2, RDS). Įvertinta, jog nesant korektiškam lokalių ugniasienių tinklui, per daug saugos atsakomybių lieka sistemos kūrėjams ar partneriams, kadangi prisijungimo duomenys asmeninių prietaisų pagalba gali būti prarandami. Eksperimentų rezultatai parodė, jog tinkamai sukonfigūruota programos lygmens ugniasienė WAF kartu su turinio paskirstymo tinklu CloudFront ne tik visapusiškai apsaugo DNS zoną, bet net ir pagreitina programos veikimą. Nustatyta, jog dirbtiniu intelektu grįsti įrankiai tiksliai atranda saugumo pažeidžiamumus, todėl šio komponento nenaudojimas atima iš sistemos kūrėjų „akis“ tokių pažeidžiamumų matymui.

Rekomenduojama taikyti šį saugumo modelį visapusiškai, atsižvelgiant į nulinio pasitikėjimo koncepciją. Pilnam sistemos saugumui būtina integruoti programos lygmens ugniasienes kartu su lokalaus (infrastruktūros) lygio ugniasienėmis. Toks šių komponentų derinimas užtikrina, jog sistema nenukentės tiek nuo išorinių atakų (bandymo paveikti DNS zoną), tiek nuo pandymo prasiskverbti į infrastruktūros lygmenį (SSH prievado atakos). Rekomenduojama įgyvendinti atsargines kopijas ne tik duomenų bazei, bet ir duomenų talpyklai bei serveriams. Kuo dažnesnės kopijos bus, tuo mažesnis šansas, jog bus prarasti duomenys. Itin svarbu prie jau minėtų saugumo parametrų pridėti ir dirbtinio intelekto įrankius. Šie įrankiai pastebės programinės įrangos bibliotekų pažeidžiamumus bei stebės sistemą dėl anomalių tinklo ar programinės sąsajos reiškinių.

LITERATŪRA

- [1] Amazon Web Services. (2024). *AWS Key Management Service*. Nuskaityta iš Developer Guide: <https://docs.aws.amazon.com/kms/latest/developerguide/overview.html>
- [2] Amazon Web Services. (2024). *AWS Shield Standard*. Nuskaityta iš How Shield and Shield Advanced work?: <https://docs.aws.amazon.com/waf/latest/developerguide/ddos-standard-summary.html>
- [3] Amazon Web Services. (2024). *Data Encryption*. Nuskaityta iš Introduction to AWS Security: <https://docs.aws.amazon.com/whitepapers/latest/introduction-aws-security/data-encryption.html>
- [4] Amazon Web Services. (2024). *How AWS Config Works*. Nuskaityta iš Developer Guide: <https://docs.aws.amazon.com/config/latest/developerguide/how-does-config-work.html>
- [5] Amazon Web Services. (2024). *How CloudTrail works*. Nuskaityta iš AWS CloudTrail: <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/how-cloudtrail-works.html>
- [6] Amazon Web Services. (2024). *Using AWS WAF protections*. Nuskaityta iš Developer Guide: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-web-aws-waf.html>
- [7] Amazon Web Services. (2024). *What is AWS Security Hub?* Nuskaityta iš AWS Security Hub: <https://docs.aws.amazon.com/securityhub/latest/userguide/what-is-securityhub.html>
- [8] Amazon Web Services. (2024). *What is IAM?* Nuskaityta iš AWS Identity and Access Management: <https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>
- [9] Gilman, E., & Barth, D. (2017). *Zero Trust Networks: Building Secure Systems in Untrusted Networks*. O'Reilly Media.
- [10] Google Cloud. (2024). *What is Cloud Security?* Nuskaityta iš Google Cloud: <https://cloud.google.com/learn/what-is-cloud-security>
- [11] Networks, S. M. (2024). *Zero Trust: kas tai?* Vilnius.

1 PRIEDAS

Šiame priede pateikta AWS infrastruktūros prieigų ir teisių kodo dalis.

```
data "aws_caller_identity" "this" {}

data "aws_iam_policy_document" "instance_assume_role_policy" {
  statement {
    actions = ["sts:AssumeRole"]

    principals {
      type       = "AWS"
      identifiers = ["arn:aws:iam::${data.aws_caller_identity.this.account_id}:root"]
    }
  }
}

resource "aws_iam_role" "this" {
  name           = "${var.env}_terraform_role"
  description    = "Role to be assumed for creating infrastructure in ${var.env}
enviroment."
  assume_role_policy = data.aws_iam_policy_document.instance_assume_role_policy.json
}

resource "aws_iam_role_policy_attachment" "this" {
  role       = aws_iam_role.this.name
  policy_arn = "arn:aws:iam::aws:policy/AdministratorAccess"
}

data "aws_iam_policy_document" "this" {
  version = "2012-10-17"

  statement {
    sid      = "AllowAssumeRole"
    effect   = "Allow"
    actions  = ["sts:AssumeRole"]
    resources = [aws_iam_role.this.arn]
  }
}

resource "aws_iam_policy" "this" {
  name           = "Allow_assume_terraform_role_${var.env}"
  description    = "Allow useres in group to assume role."
  policy         = data.aws_iam_policy_document.this.json
}

resource "aws_iam_group" "this" {
  name = "Terraform_admins_${var.env}"
}

resource "aws_iam_group_policy_attachment" "this" {
  group       = aws_iam_group.this.name
  policy_arn = aws_iam_policy.this.arn
}
```

2 PRIEDAS

Šiame priede pateikta AWS infrastruktūros tinklo kodo dalis.

```
locals {
  cluster_name = "${var.env}-${var.eks_cluster_name}"
}
module "vpc" {
  source          = "terraform-aws-modules/vpc/aws"
  version        = "5.8.0"
  name           = "${local.cluster_name}-vpc"
  cidr           = var.vpc_cird
  enable_nat_gateway = true
  single_nat_gateway = false
  one_nat_gateway_per_az = true
  azs            = ["${var.aws_region}a", "${var.aws_region}b",
"${var.aws_region}c"]
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24"]
  public_subnets  = ["10.0.101.0/24", "10.0.102.0/24",
"10.0.103.0/24"]
  database_subnets = ["10.0.201.0/24", "10.0.202.0/24",
"10.0.203.0/24"]
  enable_dns_hostnames = true
  enable_dns_support   = true
  enable_flow_log      = true
  create_flow_log_cloudwatch_iam_role = true
  create_flow_log_cloudwatch_log_group = true

  public_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/elb"                      = 1
  }

  private_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/internal-elb"             = 1
  }

  database_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/internal-elb"             = 1
  }

  vpc_tags = {
    Name = "VPC-${var.env}"
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
  }
}
```

3 PRIEDAS

Šiame priede pateikta duomenų bazės kūrimo kodo dalis.

```
locals {
  vpc_id           = data.tfe_outputs.vpc.values.vpc_id
  private_subnets = data.tfe_outputs.vpc.values.vpc_private_subnets
  public_subnets  = data.tfe_outputs.vpc.values.vpc_public_subnets
  database_subnets = data.tfe_outputs.vpc.values.vpc_database_subnets
  eks_name         = data.tfe_outputs.eks.values.eks_name
  eks_sg           = data.tfe_outputs.eks.values.eks_sg
}

data "tfe_outputs" "vpc" {
  organization = "BTI"
  workspace    = "01-vpc-zero-trust"
}

data "tfe_outputs" "eks" {
  organization = "BTI"
  workspace    = "02-eks-zero-trust"
}

module "rds" {
  source           = "../03-rds-zero-trust-module"
  env              = "prod"
  vpc_id           = local.vpc_id
  database_subnets = local.public_subnets
  eks_sg_id        = local.eks_sg
  rds_family        = "postgres14"
  rds_engine        = "postgres"
  rds_engine_version = "14.12"
  rds_instance_class = "db.t3.medium"
  rds_db_name        = var.rds_master_db_name
  rds_db_username    = var.rds_master_username
  rds_db_password    = var.rds_master_password
  manage_master_user_password = false
}

locals {
  rds_name = "${var.env}-${var.rds_engine}"
}

resource "aws_db_parameter_group" "this" {
  name     = "${local.rds_name}-param-group"
  family   = var.rds_family
  parameter {
    name = "log_connections"
    value = "1"
  }
}

resource "aws_db_subnet_group" "this" {
  name       = "${local.rds_name}-subnet-group"
  subnet_ids = var.database_subnets
}

resource "aws_db_instance" "this" {
  identifier           = "${local.rds_name}-rds-instance"
  allocated_storage    = var.rds_storage_size
  engine               = var.rds_engine
  engine_version        = var.rds_engine_version
  instance_class        = var.rds_instance_class
  db_name               = var.rds_db_name
  username              = var.rds_db_username
}
```

```

    password                                = var.manage_master_user_password == false ?
var.rds_db_password : null
    manage_master_user_password            = var.manage_master_user_password == true ?
var.manage_master_user_password : null
    master_user_secret_kms_key_id          = var.manage_master_user_password == true ?
aws_kms_key.this[0].key_id : null
    skip_final_snapshot                    = true
    parameter_group_name                    = aws_db_parameter_group.this.name
    db_subnet_group_name                    = aws_db_subnet_group.this.name
    vpc_security_group_ids                  = [aws_security_group.this.id]
    publicly_accessible                      = true
    storage_encrypted                       = true
}

resource "aws_kms_key" "this" {
    count      = var.manage_master_user_password ? 1 : 0
    description = "${local.rds_name}-kms-key"
}

resource "aws_security_group" "this" {
    name      = "${local.rds_name}-rds-sg"
    description = "Allow all inbound traffic rds to eks"
    vpc_id    = var.vpc_id
}

resource "aws_vpc_security_group_ingress_rule" "eks_rds_ingress_5432" {
    description      = "Ingress from EKS to RDS"
    from_port        = var.rds_port
    to_port          = var.rds_port
    ip_protocol       = "tcp"
    security_group_id = var.eks_sg_id
    referenced_security_group_id = aws_security_group.this.id
    tags = {
        Name = "eks_rds_ingress_5432"
    }
}

resource "aws_vpc_security_group_egress_rule" "eks_rds_egress_5432" {
    description      = "Egress from EKS to RDS"
    from_port        = var.rds_port
    to_port          = var.rds_port
    ip_protocol       = "tcp"
    security_group_id = var.eks_sg_id
    referenced_security_group_id = aws_security_group.this.id
    tags = {
        Name = "eks_rds_egress_5432"
    }
}

resource "aws_vpc_security_group_egress_rule" "rds_eks_egress_5432" {
    description      = "Egress from EKS to RDS"
    from_port        = var.rds_port
    to_port          = var.rds_port
    ip_protocol       = "tcp"
    security_group_id = aws_security_group.this.id
    referenced_security_group_id = var.eks_sg_id
    tags = {
        Name = "rds_eks_egress_5432"
    }
}

resource "aws_vpc_security_group_ingress_rule" "rds_eks_ingress_5432" {
    description      = "Ingress from RDS to EKS"
    from_port        = var.rds_port
    to_port          = var.rds_port
    ip_protocol       = "tcp"
    security_group_id = aws_security_group.this.id

```

```
referenced_security_group_id = var.eks_sg_id
tags = {
    Name = "rds_eks_ingress_5432"
}
}
```

4 PRIEDAS

Šiame priede pateikta duomenų talpyklos AWS S3 kodo dalis.

```
resource "aws_s3_bucket" "project_bucket" {
  bucket = "zero-trust-data"
}
resource "aws_s3_bucket_acl" "project_bucket" {
  bucket      = aws_s3_bucket.project_bucket.id
  acl         = "public-read"
  depends_on = [aws_s3_bucket_ownership_controls.s3_bucket_acl_ownership]
}
resource "aws_s3_bucket_versioning" "project_bucket" {
  bucket = aws_s3_bucket.project_bucket.id
  versioning_configuration {
    status = "Enabled"
  }
}
resource "aws_s3_bucket_cors_configuration" "project_bucket" {
  bucket = aws_s3_bucket.project_bucket.id
  cors_rule {
    allowed_headers = ["*"]
    allowed_methods = ["PUT", "POST"]
    allowed_origins = ["*"]
    expose_headers  = ["ETag"]
    max_age_seconds = 3000
  }
}
resource "aws_s3_bucket_server_side_encryption_configuration" "project_bucket" {
  bucket = aws_s3_bucket.project_bucket.bucket
  rule {
    apply_server_side_encryption_by_default {
      sse_algorithm = "AES256"
    }
  }
}
resource "aws_s3_bucket_public_access_block" "zero_trust" {
  bucket = aws_s3_bucket.project_bucket.id

  block_public_acls       = false
  block_public_policy     = false
  ignore_public_acls     = false
  restrict_public_buckets = false
}
resource "aws_s3_bucket_ownership_controls" "s3_bucket_acl_ownership" {
  bucket = aws_s3_bucket.project_bucket.id
  rule {
    object_ownership = "BucketOwnerPreferred"
  }
  depends_on = [aws_s3_bucket_public_access_block.zero_trust]
}
```

5 PRIEDAS

Šiame priede pateikta AWS EKS serverio kūrimo kodo dalis.

```
locals {
  vpc_id          = data.tfe_outputs.vpc.values.vpc_id
  private_subnets = data.tfe_outputs.vpc.values.vpc_private_subnets
  public_subnets  = data.tfe_outputs.vpc.values.vpc_public_subnets
  database_subnets = data.tfe_outputs.vpc.values.vpc_database_subnets
  subnets         = concat(local.private_subnets)
}

data "tfe_outputs" "vpc" {
  organization = "BTB"
  workspace    = "01-vpc-zero-trust"
}

module "eks" {
  source      = "../02-eks-zero-trust-module"
  eks_version = "1.27"
  env         = "prod"
  eks_name    = "zero-trust-cluster"
  subnet_ids  = local.private_subnets
  node_groups = {
    general = {
      capacity_type = "ON_DEMAND"
      instance_types = ["t3.medium"]
      scaling_config = {
        desired_size = 2
        max_size      = 3
        min_size      = 1
      }
    }
  }
}

output "asg" {
  value      = module.eks.asg
  sensitive = true
}

resource "aws_iam_role" "eks" {
  name = "${var.env}-${var.eks_name}-eks-cluster"

  assume_role_policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
POLICY
}

resource "aws_iam_role_policy_attachment" "eks" {
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
  role       = aws_iam_role.eks.name
}
```

```

resource "aws_eks_cluster" "this" {
  name      = "${var.env}-${var.eks_name}"
  version   = var.eks_version
  role_arn  = aws_iam_role.eks.arn

  vpc_config {
    endpoint_private_access = false
    endpoint_public_access  = true

    subnet_ids = var.subnet_ids
  }

  depends_on = [aws_iam_role_policy_attachment.eks]
}

resource "aws_iam_role" "nodes" {
  name = "${var.env}-${var.eks_name}-eks-nodes"

  assume_role_policy = jsonencode({
    Statement = [{
      Action = "sts:AssumeRole"
      Effect = "Allow"
      Principal = {
        Service = "ec2.amazonaws.com"
      }
    }]
    Version = "2012-10-17"
  })
}

resource "aws_iam_role_policy_attachment" "nodes" {
  for_each = var.node_iam_policies

  policy_arn = each.value
  role       = aws_iam_role.nodes.name
}

resource "aws_eks_node_group" "this" {
  for_each = var.node_groups

  cluster_name    = aws_eks_cluster.this.name
  node_group_name = each.key
  node_role_arn   = aws_iam_role.nodes.arn

  subnet_ids = var.subnet_ids

  capacity_type = each.value.capacity_type
  instance_types = each.value.instance_types

  scaling_config {
    desired_size = each.value.scaling_config.desired_size
    max_size     = each.value.scaling_config.max_size
    min_size     = each.value.scaling_config.min_size
  }

  update_config {
    max_unavailable = 1
  }

  labels = {
    role = each.key
  }
  tags = {
    "key"           = "Name"
    "value"         = "${var.env}-cluster-${each.key}-node"
    "propagate_at_launch" = true
  }
}

```

```

    depends_on = [
        aws_iam_role_policy_attachment.nodes
    ]
}

resource "aws_autoscaling_group_tag" "this" {
    for_each = aws_eks_node_group.this

    autoscaling_group_name = each.value.resources[0].autoscaling_groups[0].name

    tag {
        key            = "Name"
        value          = "${var.env}-cluster-${each.key}-node"
        propagate_at_launch = true
    }
    depends_on = [
        aws_eks_node_group.this
    ]
}

data "tls_certificate" "this" {
    count = var.enable_irsa ? 1 : 0

    url = aws_eks_cluster.this.identity[0].oidc[0].issuer
}

resource "aws_iam_openid_connect_provider" "this" {
    count = var.enable_irsa ? 1 : 0

    client_id_list = ["sts.amazonaws.com"]
    thumbprint_list = [data.tls_certificate.this[0].certificates[0].sha1_fingerprint]
    url            = aws_eks_cluster.this.identity[0].oidc[0].issuer
}

```

6 PRIEDAS

Šiame priede pateikta Kubernetes klasterio konfigūracijos kodo dalis.

```
locals {
  eks_name = data.tfe_outputs.eks.values.eks_name
  eks_sg   = data.tfe_outputs.eks.values.eks_sg
  eks_oidc_provider_url = data.tfe_outputs.eks.values.eks_oidc_provider_url
  openid_provider_arn   = data.tfe_outputs.eks.values.openid_provider_arn
  vpc_private_subnets  = data.tfe_outputs.vpc.values.vpc_private_subnets
  vpc_id                = data.tfe_outputs.vpc.values.vpc_id
}

data "tfe_outputs" "vpc" {
  organization = "BTT"
  workspace    = "01-vpc-zero-trust"
}

data "tfe_outputs" "eks" {
  organization = "BTT"
  workspace    = "02-eks-zero-trust"
}

data "tfe_outputs" "rds" {
  organization = "BTT"
  workspace    = "03-rds-zero-trust"
}

module "vpc-cni" {
  source          = "../05-k8s-zero-trust-module/VPC-cni"
  env             = var.env
  eks_name        = local.eks_name
  openid_provider_arn = local.openid_provider_arn
  addon_vpc_cni    = true
}

module "kube-proxy" {
  source          = "../05-k8s-zero-trust-module/kube-proxy"
  env             = var.env
  eks_name        = local.eks_name
  openid_provider_arn = local.openid_provider_arn
  addon_kube_proxy = true
}

module "core-dns" {
  source          = "../05-k8s-zero-trust-module/core-dns"
  env             = var.env
  eks_name        = local.eks_name
  openid_provider_arn = local.openid_provider_arn
  addon_coredns    = true
}

module "autoscaler" {
  source          = "../05-k8s-zero-trust-module/Autoscaler"
  env             = var.env
  eks_name        = local.eks_name
  cluster_autoscaler_helm_version = "9.28.0"
  openid_provider_arn = local.openid_provider_arn
  depends_on = [
    module.kube-proxy,
    module.core-dns,
    module.vpc-cni
  ]
}

module "cert-manager" {
  source          = "../05-k8s-zero-trust-module/Cert-manager"
  env             = var.env
  eks_name        = local.eks_name
```

```

    enable_issuer_cloudflare      = true
    cluster_certmanager_helm_version = "1.11.1"
    openid_provider_arn           = local.openid_provider_arn
    email                         = "karolis.zivulskas@bttcloud.com"
    aws_region                    = "eu-west-1"
    depends_on = [
        module.autoscaler
    ]
}

module "ebs-driver" {
    source      = "../05-k8s-zero-trust-module/EBS-driver"
    env         = var.env
    eks_name    = local.eks_name
    openid_provider_arn = local.openid_provider_arn
    depends_on = [
        module.autoscaler
    ]
}

module "nginx-controller" {
    source = "../05-k8s-zero-trust-module/Nginx-ingress-controller"
    env    = var.env
    eks_name    = local.eks_name
    openid_provider_arn = local.openid_provider_arn
    nginx_ingress_helm_version = "4.6.0"
    nginx_values    = file("${path.module}/nginx/values.yaml")
    depends_on = [
        module.autoscaler
    ]
}

data "aws_eks_cluster" "this" {
    name = var.eks_name
}

resource "helm_release" "nginx" {
    name      = "nginx"
    repository = "https://kubernetes.github.io/ingress-nginx"
    chart     = "ingress-nginx"
    version   = var.nginx_ingress_helm_version
    namespace = "ingress-nginx"
    create_namespace = true
    values = [
        var.nginx_values
    ]
    depends_on = [
        data.aws_eks_cluster.this
    ]
}

data "aws_iam_openid_connect_provider" "this" {
    arn = var.openid_provider_arn
}

resource "aws_iam_role" "cert_manager" {
    name      = "${var.eks_name}-cert-manager-role"
    assume_role_policy = <<-EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Principal": {

```

```

        "Federated": "${data.aws_iam_openid_connect_provider.this.arn}"
      },
      "Condition": {
        "StringEquals": {
          "${data.aws_iam_openid_connect_provider.this.url}:sub":
"system:serviceaccount:cert-manager:cert-manager"
        }
      }
    }
  ]
}
EOF
}

resource "aws_iam_role_policy" "cert_manager" {
  name = "${var.eks_name}-cert-manager-policy"
  role = aws_iam_role.cert_manager.id

  policy = <<-EOF
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": "route53:GetChange",
        "Resource": "arn:aws:route53::change/*"
      },
      {
        "Effect": "Allow",
        "Action": [
          "route53:ChangeResourceRecordSets",
          "route53:ListResourceRecordSets"
        ],
        "Resource": "arn:aws:route53::hostedzone/*"
      },
      {
        "Effect": "Allow",
        "Action": [
          "route53:ListHostedZonesByName",
          "route53:ListHostedZones"
        ],
        "Resource": "*"
      }
    ]
  }
  EOF
}

resource "helm_release" "cert-manager" {
  name           = "cert-manager"
  repository     = "https://charts.jetstack.io"
  chart         = "cert-manager"
  version       = var.cluster_certmanager_helm_version
  namespace     = "cert-manager"
  create_namespace = true
  values = [
    templatefile("${path.module}/tpl_values.yaml", {})
  ]
  set {
    name  = "serviceAccount.create"
    value = true
  }
  set {
    name  = "serviceAccount.annotations.eks\\.amazonaws\\.com/role-arn"
    value = aws_iam_role.cert_manager.arn
  }
  set {

```

```

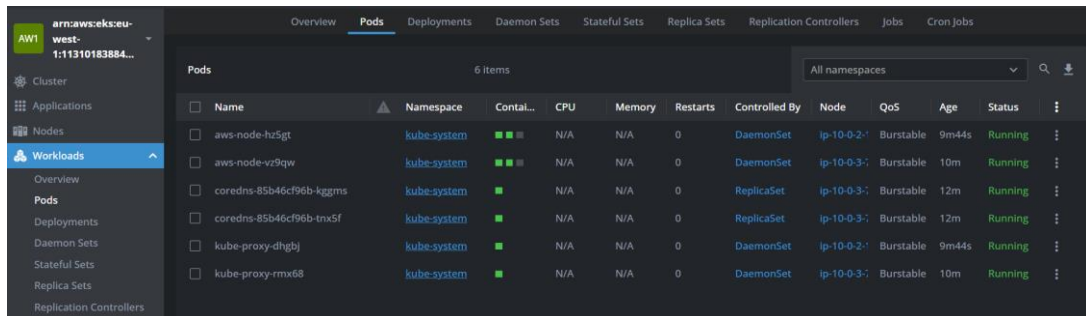
    name = "serviceAccount.name"
    value = "cert-manager"
  }
  set {
    name = "installCRDs"
    value = true
  }
  set {
    name = "autoDiscovery.clusterName"
    value = var.eks_name
  }
}

resource "kubectrl_manifest" "issuer_route53" {
  yaml_body = <<YAML
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    # Email address used for ACME registration
    email: ${var.email}
    privateKeySecretRef:
      name: letsencrypt-private-key
    solvers:
      # An empty 'selector' means that this solver matches all domains
      - selector: {}
        dns01:
          route53:
            region: eu-west-1
YAML
  validate_schema = false
  server_side_apply = true
  depends_on = [
    helm_release.cert-manager
  ]
}

```

7 PRIEDAS

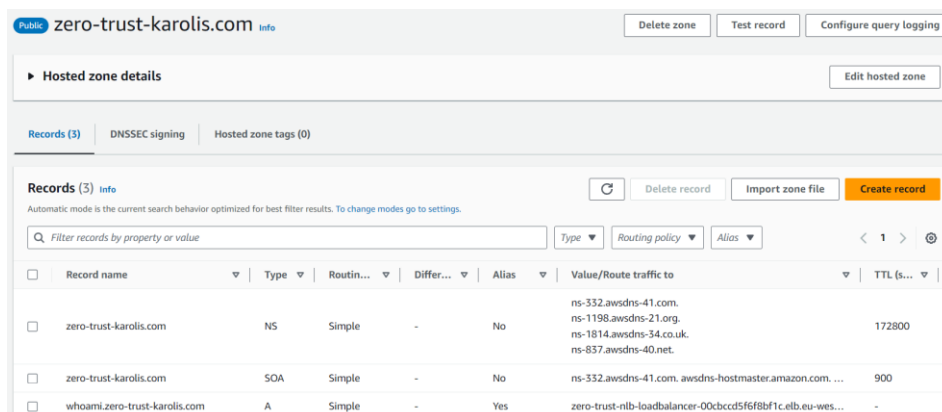
Šiame priede pateikta papildoma vizualinė medžiaga programos paleidimo bei duomenų sukūrimo ir talpinimo daliai.



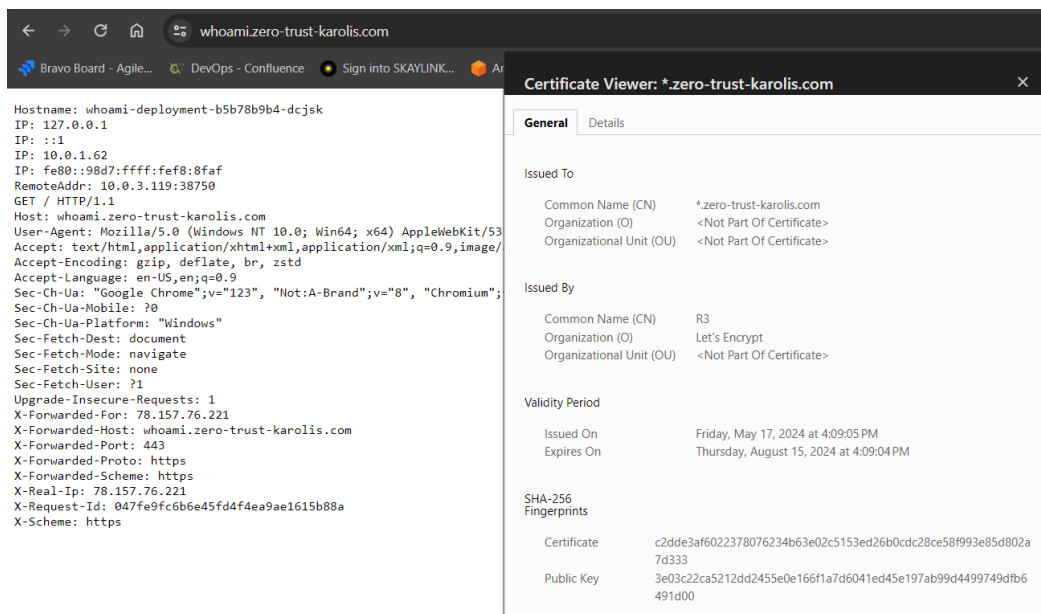
4.2.1.1 pav. Kubernetes klasterio valdymas Lens įrankio grafiniėje sąsajoje.



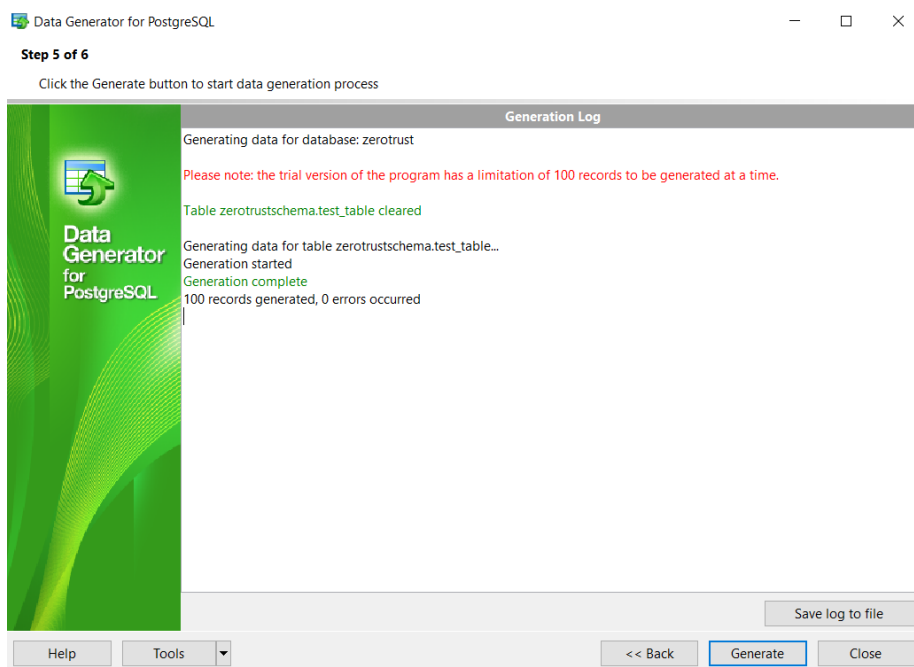
4.2.1.2 pav. Programos paleidimas lokaliai naršyklėje.



4.2.1.3 pav. Domenui zero-trust-karolis.com sukurta DNS zona. Vaizde matomas ir DNS A įrašas, kreipiantis užklausas į užklausų skirstytuvą.



4.2.1.4 pav. Patikrintas domeno ir sertifikato veikimas whoami.zero-trust-karolis.com.



4.2.2.1 pav. Duomenų generavimas EMS Data Generator įrankiu duomenų bazei RDS.

Dorothy	Walker	1983-12-21	87890123456	IS140159260076545510730339	23945.78
Joseph	Hall	1971-03-07	98901234567	BG80BNBG96611020345678	17893.12
Sandra	Allen	1995-06-19	9012345678	HR1210010051863000160	8476.23
Mark	Young	1987-01-11	10123456780	RO49AAAA1B31007593840000	9345.50
Ashley	Hernández	1979-10-18	21234567891	BH67BMAG00001299123456	7654.45
Eric	King	1984-05-05	32345678902	SA4420000001234567891234	13459.00
Kimberly	Wright	1982-11-09	43456789023	KZ176010251000042993	9876.45
Joshua	Lopez	1986-03-18	54567890134	CR72012300000123456789	17589.12
Donna	Scott	1981-09-22	65678901245	QA58DOHB00001234567890	9324.56
Brian	Green	1978-02-16	76789012356	AL47212110090000000235698741	11578.34
Laura	Adams	1993-11-13	87890123467	PK36SCBL0000001123456702	15678.22
Kevin	Baker	1976-07-26	98901234578	AZ21NABZ00000000137010001944	9432.78
Helen	Nelson	1989-06-02	9012345679	IL620108000000099999999	11230.00
Edward	Carter	1983-12-25	10123456781	KW81CBKU000000000001234560101	15487.67
Emily	Mitchell	1991-04-14	21234567892	MK07200002785123453	12945.50
Scott	Perez	1988-10-19	32345678903	LB62099900000001001901229114	17856.12
Laura	Roberts	1975-05-07	43456789034	FO6264600001631634	10459.23
Deborah	Turner	1980-12-11	54567890145	GL8964710001000206	8934.45
Jason	Phillips	1987-06-15	65678901256	DO28BAGR00000001212453611324	12345.00

4.2.2.2 pav. Atsitiktinių duomenų sugeneruotas failas S3 talpyklai.

8 PRIEDAS

Šiame priede pateikiamas programos paleidimo kodas.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whoami-deployment
  namespace: whoami
spec:
  replicas: 1
  selector:
    matchLabels:
      app: whoami
  template:
    metadata:
      labels:
        app: whoami
    spec:
      containers:
        - name: whoami-container
          image: containous/whoami
---
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: whoami-certificate
  namespace: whoami
spec:
  secretName: whoami-certificate
  commonName: "*.zero-trust-karolis.com"
  dnsNames:
    - "*.zero-trust-karolis.com"
    - "zero-trust-karolis.com"
  issuerRef:
    name: letsencrypt-prod
    kind: ClusterIssuer

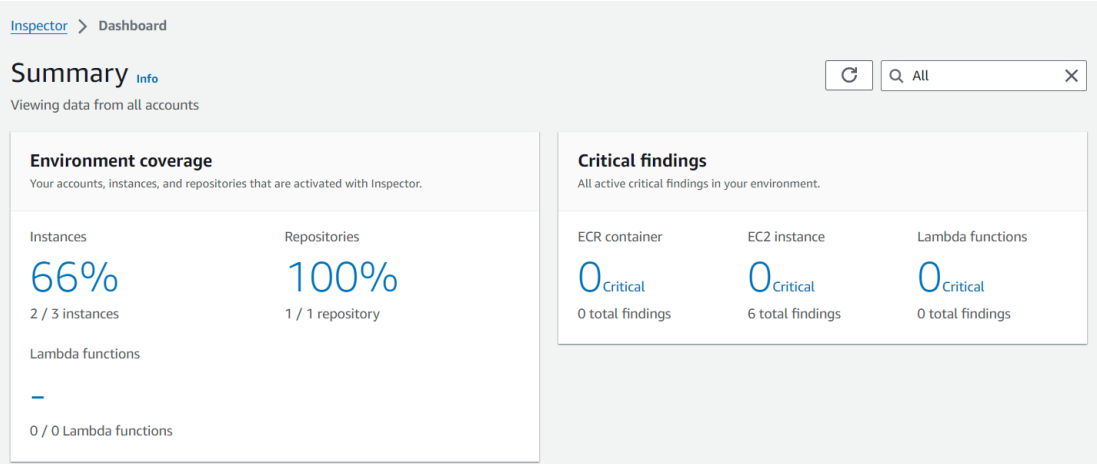
apiVersion: v1
kind: Service
metadata:
  name: whoami-service
  namespace : whoami
spec:
  ports:
    - name: https
      targetPort: 80
      port: 80
  selector:
    app: whoami

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: whoami-ingress
  namespace: whoami
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  ingressClassName: nginx
  tls:
    - hosts:
        - whoami.zero-trust-karolis.com
      secretName: whoami-certificate
  rules:
```

```
- host: whoami.zero-trust-karolis.com
http:
  paths:
    - path: /
      pathType: Prefix
      backend:
        service:
          name: whoami-service
          port:
            number: 80
```

9 PRIEDAS

Šiame priede pateikiama vaizdinė medžiaga dirbtiniu intelektu grįstų saugumo įrankių konfigūravimo daliai. Taip pat pateikta Macie taisyklių valdymo kodo dalis JSON kalba.



4.3.2.1 pav. Sukonfigūruotas Inspector įrankis.

All3

Scanning2

Not scanning1

Instances (3)Info

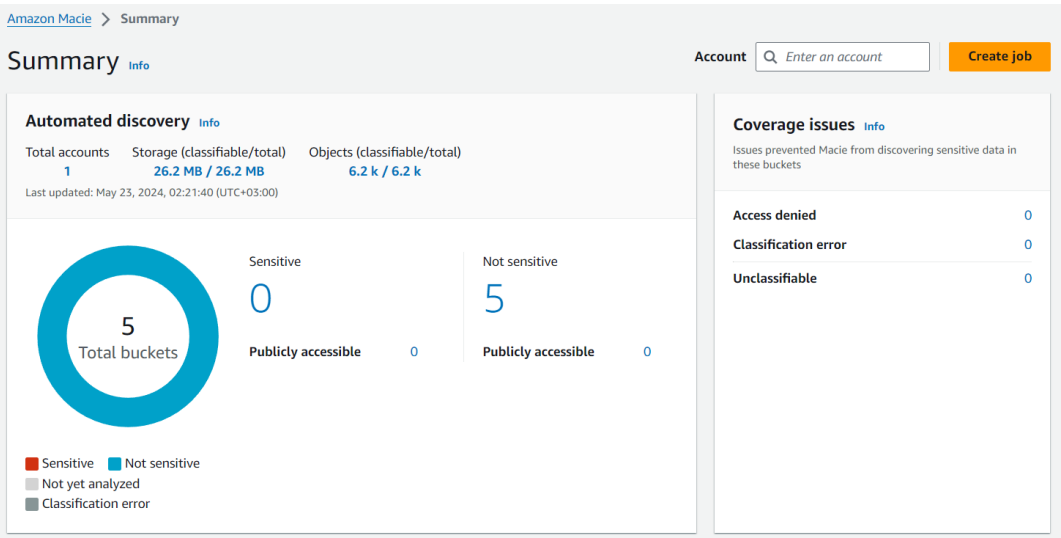
QAdd filter

Resource type EQUALS Amazon EC2 InstanceXClear filters

<1>⚙

EC2 insta...	EC2 insta...	Account	AMI	Operating...	Last scan...	Status	Monitore...
i-078b47b...	Name:jumphost	113101838846	ami-0607...	UNKNOWN	-	Unmanaged EC2 instance	Agent-based
i-0b1d839...	aws:autoscaling:groupName...	113101838846	ami-090ff...	LINUX	May 18, 2...	Actively monitoring	Agent-based
i-0ee56e3...	aws:autoscaling:groupName...	113101838846	ami-090ff...	LINUX	May 18, 2...	Actively monitoring	Agent-based

4.3.2.2 pav. Inspector įrankis aptinka egzistuojančias EC2 mašinas bei jas tikrina.



4.3.2.3 pav. Sukonfigūruotas Macie jautrios informacijos aptikimo įrankis.

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "Deny non-HTTPS access",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::zero-trust-data/*",
    "Condition": {
      "Bool": {
        "aws:SecureTransport": "false"
      }
    }
  },
  {
    "Sid": "Deny incorrect encryption header. This is optional",
    "Effect": "Deny",
    "Principal": {
      "Service": "macie.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::zero-trust-data/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:eu-west-1:113101838846:key/7286b527-a169-4f11-827d-a0d6e9b55ade"
      }
    }
  },
  {
    "Sid": "Deny unencrypted object uploads. This is optional",
    "Effect": "Deny",
    "Principal": {
      "Service": "macie.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::zero-trust-data/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "aws:kms"
      }
    }
  },
  {
    "Sid": "Allow Macie to upload objects to the bucket",
    "Effect": "Allow",
    "Principal": {
      "Service": "macie.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::zero-trust-data/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "113101838846"
      },
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:macie2:eu-west-1:113101838846:export-configuration:*",
          "arn:aws:macie2:eu-west-1:113101838846:classification-job/*"
        ]
      }
    }
  },
  {
    "Sid": "Allow Macie to use the getBucketLocation operation",
    "Effect": "Allow",
    "Principal": {
      "Service": "macie.amazonaws.com"
    }
  }
]

```

```

    },
    "Action": "s3:GetBucketLocation",
    "Resource": "arn:aws:s3:::zero-trust-data",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "113101838846"
      },
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:macie2:eu-west-1:113101838846:export-configuration:*",
          "arn:aws:macie2:eu-west-1:113101838846:classification-job/*"
        ]
      }
    }
  }
]
}

```

10 PRIEDAS

Šiame priede pateikiama vaizdinė informacija raktų bei sertifikatų kūrimui ir valdymui. Taip pat pateikta kodo dalis KMS rakto naudojimo teisėms JSON kalba.

Certificate status

Identifier

8ca8771f-2500-4cd8-81a9-fb682296910a

Status

 Issued

ARN

arn:aws:acm:us-east-1:113101838846:certificate/8ca8771f-2500-4cd8-81a9-fb682296910a

Type

Amazon Issued

Domains (2)

Create records in Route 53

Export to CSV

< 1 >

Domain	Status	Renewal status	Type	CNAME name	CNAME value
whoami.zero-trust-karolis.com	 Success	-	CNAME	_57c7ed3b2e85c86b1242e920466683c2.whoami.zero-trust-karolis.com.	 _eac63e81cevalidations.a
zero-trust-karolis.com	 Success	-	CNAME	_39229138dd5fff57cf36bf52467c045a.zero-trust-karolis.com.	 _46c75d9fetvalidations.a

4.3.4.1 pav. Sukurtas sertifikatas CloudFront turinio transliavimo įrankiui.

In use	Serial number
Yes	0b:78:02:22:7c:d6:45:d9:2e:0b:87:00:ba:76:54:3d
Domain name	Public key info
whoami.zero-trust-karolis.com	RSA 2048
Number of additional names	Signature algorithm
1	SHA-256 with RSA

4.3.4.2 pav. Programos subdomeno sertifikato informacija.

Key type [Help me choose](#)

☐ Symmetric
A single key used for encrypting and decrypting data or generating and verifying HMAC codes

☒ Asymmetric
A public and private key pair used for encrypting and decrypting data or signing and verifying messages

Key usage [Help me choose](#)

☒ Encrypt and decrypt
Use the key only to encrypt and decrypt data.

☐ Sign and verify
Key pairs for digital signing
Uses the private key for signing and the public key for verification.

Key spec [Help me choose](#)
Choose the type of key material in your signing key.

☐ RSA_2048

☐ RSA_3072

☒ RSA_4096

4.3.4.3 pav. KMS šifravimo rakto kūrimas.

7286b527-a169-4f11-827d-a0d6e9b55ade Key actions ▼ Edit

General configuration

Alias zero-kms-trust-key	Status Enabled	Creation date May 18, 2024 18:42 GMT+3
ARN arn:aws:kms:eu-west-1:113101838846:key/7286b527-a169-4f11-827d-a0d6e9b55ade	Description -	Regionality Single Region

Key policy

Cryptographic configuration

Tags

Key rotation

Aliases

Cryptographic configuration

Key Type Symmetric	Origin AWS KMS	Key Spec ⓘ SYMMETRIC_DEFAULT	Key Usage Encrypt and decrypt
-----------------------	-------------------	---------------------------------	----------------------------------

4.3.4.4 pav. KMS rakto valdymas AWS pusėje.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::113101838846:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::113101838846:user/karolis.zivulskas@bttcloud.com"
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms>Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
      ],
      "Resource": "*"
    }
  ]
}
```

11 PRIEDAS

Šiame priede pateikiama vaizdinė informacija tyrimo daliai apie išplėstinę prieigų ir teisių peržiūrą.

User name

Zero-trust-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

 **Do you want to provide console access to a person?**
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications. To do so, you must be signed in to the Identity Center console using the credentials of the management account in AWS Organizations. If you aren't the management account owner, contact the owner to perform this task.

Console password

☐ Autogenerated password
You can view the password after you create the user.

☒ Custom password
Enter a custom password for the user.

Zerotrust123!

☒ Show password

4.3.1.1 pav. Papildomo naudotojo sukūrimas.

Users (1) Permissions Access Advisor				
Users in this group (1) An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.				
Search < 1 > ⚙				
☐	User name 	▲	Groups	Last activity ▼
☐	Zero-trust-user		1	None
				Creation time ▼
				4 days ago

4.3.1.2 pav. Papildomos naudotojų grupės sukūrimas.

Authenticator app
A virtual MFA device is an application running on your device that you can configure by scanning a QR code.

- 1 Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer.
[See a list of compatible applications](#)
- 2 
Open your authenticator app, choose **Show QR code** on this page, then use the app to scan the code. Alternatively, you can type a secret key.
[Show secret key](#)
- 3 Type two consecutive MFA codes below
Enter a code from your virtual app below

Wait 30 seconds, and enter a second code entry.

4.3.1.3 pav. Kelių patvirtinimų autentifikavimo pridėjimas papildoma naudotojui.

Multi-factor authentication (MFA) (1) Remove Resync Assign MFA device			
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more			
	Type	Identifier	Created on
	Virtual	arn:aws:iam::113101838846:mfa/zero-trust-user	Now

4.3.1.4 pav. Kelių patvirtinimų autentifikavimo valdymas.

12 PRIEDAS

Šiame priede pateikta vaizdinė informacija, susijusi su atsarginių kopijų konfigūravimu AWS S3 bei AWS EC2 resursams.

[AWS Backup](#) > [Backup plans](#) > [zero-trust-s3-backup-plan](#) > zero-trust-s3-backup-rule

zero-trust-s3-backup-rule

DeleteEdit

Summary

Backup rule name zero-trust-s3-backup-rule	Frequency Daily At 12:30 AM Europe/Vilnius (UTC+03:00)	Start within 8 hours	Complete within 7 days
Transition to cold storage Not enabled	Archive Amazon EBS snapshots Not enabled	Total retention period 1 week	Backup vault Default
Continuous backup Disabled	Tags added to recovery points - optional -		

4.3.7.1 pav. AWS S3 atsarginės kopijos sukonfigūruota taisyklė.

policy-0fefbf325a769aa2d

EnableDisableModifyDelete

Policy ID policy-0fefbf325a769aa2d	Description zero-trust-policy-ec2	Resource type Instance	Resource location cloud
Policy type EBS snapshot policy	Policy state Enabled	IAM role arn:aws:iam::113101838846:role/service-role/AWSDataLifecycleManagerDefaultRole	Date created Thu May 23 2024 03:18:58 GMT+0300 (Eastern European Summer Time)
Date modified Thu May 23 2024 03:18:58 GMT+0300 (Eastern European Summer Time)	Target instances with these tags CreatedBy:Karolis_Zivulskas	Exclude snapshot of root volume No	Exclude data volumes with these tags -

SchedulesMonitoringTags

Schedule

Schedule details

Frequency

Every 24 hours starting at 09:00 UTC.

Variable tags

instance-id:\$(instance-id), timestamp:\$(timestamp)

Retain rule

A maximum of 2 will be retained. The oldest snapshot retained will be <= 2 days old.

4.3.7.2 pav. AWS EC2 atsarginės kopijos konfigūravimas.

61

13 PRIEDAS

Šiame priede pateikta vaizdinė dalis centralizuoto stebėjimo įrankio kūrimo procesui.

Security standards

Info

Track your cloud security posture with a summary security score and standard security scores. This widget always shows complete, unfiltered data.

Security score

The security score has been requested, the score will be available within 30 minutes

Standard	Passed	Failed	Score ▲
AWS Foundational Security Best Practices v1.0.0	0	0	0%
CIS AWS Foundations Benchmark v1.2.0	0	0	0%
NIST Special Publication 800-53 Revision 5	0	0	0%
PCI DSS v3.2.1	0	0	0%

4.3.6.1 pav. SecurityHub sukonfigūruoti saugumo standartai.

Data governance

IAM role for AWS Config

☒ Use an existing AWS Config service-linked role
Service-linked roles are predefined and include all the permissions that AWS Config requires to call other AWS services.

☐ Choose a role from your account
Choose an IAM role from one of your pre-existing roles and permission policies.

Delivery method

Amazon S3 bucket

☒ Create a bucket

☐ Choose a bucket from your account

☐ Choose a bucket from another account

Ensure appropriate permissions are available in this S3 bucket's policy. [Learn more](#)

S3 Bucket name (required)

karolis-config-talpykla

Prefix (optional)

/AWSLogs/113101838846/Config/
eu-west-1

Amazon SNS topic

☐ Stream configuration changes and notifications to an Amazon SNS topic.
If you choose email as the notification endpoint for your SNS topic, this can cause a high volume of email. [Learn more](#)

4.3.6.2 pav. Config konfigūravimas bei radinių talpyklos kūrimas.

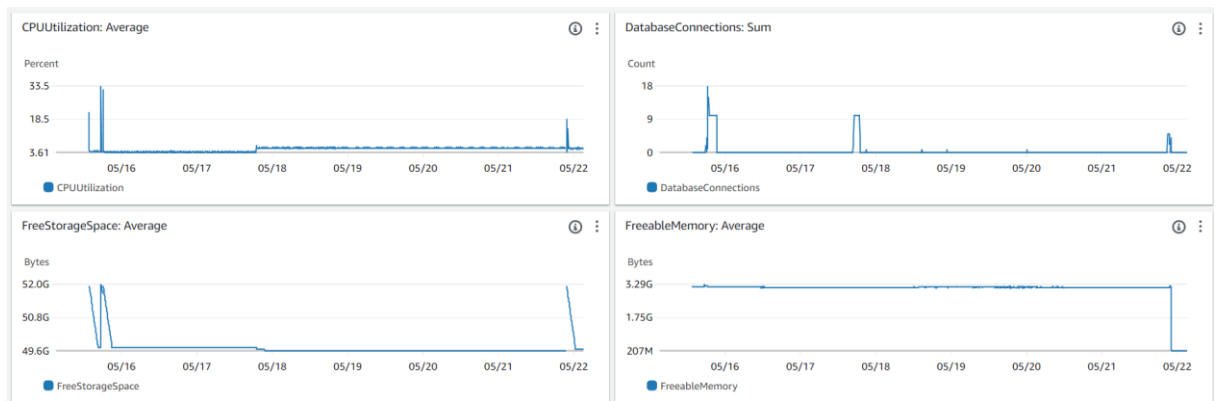
14 PRIEDAS

Šiame priede pateikiama CloudWatch stebėjimo įrankio panaudojimo CloudTrail įvykių tikrinimui kodo dalis JSON kalba. Taip pat pateikta vaizdinė medžiaga apie CloudTrail įvykių sekos kūrimą bei CloudWatch resursų stebėjimą.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailCreateLogStream20141110",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream"
      ],
      "Resource": [
        "arn:aws:logs:eu-west-1:113101838846:log-group:aws-cloudtrail-logs-113101838846-f3270fb5:log-stream:113101838846_CloudTrail_eu-west-1*"
      ]
    },
    {
      "Sid": "AWSCloudTrailPutLogEvents20141101",
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:eu-west-1:113101838846:log-group:aws-cloudtrail-logs-113101838846-f3270fb5:log-stream:113101838846_CloudTrail_eu-west-1*"
      ]
    }
  ]
}
```

General details				Edit
Trail logging ✔ Logging	Trail log location zero-trust-cloudtrail-logs/AWSLogs/113101838846	Log file validation Enabled	SNS notification delivery Disabled	
Trail name zero-trust-events	Last log file delivered -	Last file validation delivered -	Last SNS notification -	
Multi-region trail Yes	Log file SSE-KMS encryption Enabled			
Apply trail to my organization Not enabled	AWS KMS key arn:aws:kms:eu-west-1:113101838846:key/35e13a3c-56a0-4648-b566-a81d483734de			
	AWS KMS key alias zero-trust-cloudtrail-key			

4.3.5.1 pav. Sukurta CloudTrail įvykių seka į S3 talpyklą.



4.3.5.2 pav. Resursų stebėjimas CloudWatch komponente.

15 PRIEDAS

Šiame priede pateikiama vaizdinė medžiaga CloudFront transliavimo tinklo sukūrimui bei konfigūravimui.

Details

Distribution domain name

 d10hy8ye9v07mt.cloudfront.net

ARN


arn:aws:cloudfront::113101838846:distribution/E2TAORN6X9YI
OY

Last modified

May 18, 2024 at 1:03:20 PM UTC

Settings

Edit

Description

-

Price class

Use only North America and Europe

Supported HTTP versions

HTTP/3, HTTP/2, HTTP/1.1, HTTP/1.0

Alternate domain names

whoami.zero-trust-karolis.com

Custom SSL certificate

 whoami.zero-trust-karolis.com 

Security policy

TLShv1.2_2021

Standard logging

On

Cookie logging

On

Default root object

-

4.3.9.1 pav. CloudFront distribucijos kūrimas.

Record name [Info](#)

whoami

.zero-trust-karolis.com

Record type [Info](#)

A – Routes traffic to an IPv4 address and some AWS resources

Keep blank to create a record for the root domain.

☒ Alias

Route traffic to [Info](#)

Alias to CloudFront distribution

US East (N. Virginia)

An alias to a CloudFront distribution and another record in the same hosted zone are global and available only in US East (N. Virginia).

Q d10hy8ye9v07mt.cloudfront.net

X

Routing policy [Info](#)

Simple routing

Evaluate target health

☐ No

4.3.9.2 pav. CloudFront DNS A įrašo sukūrimas subdomenui.

16 PRIEDAS

Šiame priede pateikiama vaizdinė AWS WAF ugniasienės konfigūracijos medžiaga.

[AWS WAF](#) > [IP sets](#) > zero-trust-mano-ip

zero-trust-mano-ip

Info Edit

Name zero-trust-mano-ip	Region Global (CloudFront)
Description	IP version IPV4

IP addresses (1) Delete Add IP address

< **1** > ⚙

☐ IP addresses

☐ 78.157.76.221/32

4.3.3.1 pav. WAF ugniasienės nepatikimų IP adresų sąrašo sudarymas.

zero-trust-waf-taisykle

Edit Delete

Details JSON

Rule

Rule name zero-trust-waf-taisykle	Type Regular rule	Region Global (CloudFront)
--------------------------------------	----------------------	-------------------------------

If a request matches at least one of the statements (OR)

Statement 1

Request option
Originates from countries

Country
China - CN, Russian Federation - RU, Singapore - SG

4.3.3.2 pav. WAF ugniasienės taisyklė su geografinės lokacijos ribojimais.

OR

Statement 2

IP set name
zero-trust-mano-ip

IP set description
None

IP set ARN
[arn:aws:wafv2:us-east-1:113101838846:global/ipset/zero-trust-mano-ip/3f5e8a8c-beb3-4d09-8f12-bf5da6045618](#)

IP address version
IPV4

IP addresses
78.157.76.221/32

4.3.3.3 pav. WAF ugniasienės taisyklė su IP adresų ribojimais.

Rules (4) Edit Delete Add rules ▼			
If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.			
☐	Name	Capacity	Action
☐	AWS-AWSManagedRulesAmazonIpReputationList	25	Use rule actions
☐	AWS-AWSManagedRulesKnownBadInputsRuleSet	200	Use rule actions
☐	AWS-AWSManagedRulesAnonymousIpList	50	Use rule actions
☐	AWS-AWSManagedRulesCommonRuleSet	700	Use rule actions

4.3.3.4 pav. WAF ugniasienei sukonfigūruotos taisyklės.

17 PRIEDAS

Šiame priede pateiktas bandymas prisijunti prie AWS paskyros su nutekintais duomenimis. Taip pat pateiktas įvykis iš CloudTrail apie nepavykusį bandymą autentifikuotis JSON kalba.

Sign in as IAM user

Account ID (12 digits) or account alias

113101838846

IAM user name

Zero-trust-user

Password

.....

☐ Remember this account

Sign in

4.4.1.1 pav. Bandymas prisijungti naudojant prisijungimo duomenis.

Multi-factor Authentication

Enter an MFA code to complete sign-in.

MFA Code:

645275

Submit

[Cancel](#)

4.4.1.2 pav. Bandymas atspėti kelių patvirtinimų autentifikavimo kodą.

Your authentication information is incorrect. Please try again.

Sign in as IAM user

Account ID (12 digits) or account alias

113101838846

IAM user name

Zero-trust-user

Password

☐ Remember this account

Sign in

4.4.1.3 pav. Nepavykęs prisijungimas.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDARUVLFDH7LYNPJ2TDE",
    "accountId": "113101838846",
    "accessKeyId": "",
    "userName": "Zero-trust-user"
  },
  "eventTime": "2024-05-22T23:35:37Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "78.157.76.221",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/123.0.0.0 Safari/537.36",
  "errorMessage": "Failed authentication",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "LoginTo": "https://eu-west-
1.console.aws.amazon.com/console/home?hashArgs=%23&isauthcode=true&region=eu-west-
1&state=hashArgsFromTB_eu-west-1_057512d7f1a67d0e",
    "MobileVersion": "No",
    "MFAUsed": "Yes"
  },
  "eventID": "47e3aa76-0d6f-457a-8229-1baae617457b",
  "readOnly": false,
  "eventType": "AwsConsoleSignIn",
  "managementEvent": true,
  "recipientAccountId": "113101838846",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
```

```
    "cipherSuite": "TLS_AES_128_GCM_SHA256",  
    "clientProvidedHostHeader": "eu-west-1.signin.aws.amazon.com"  
  }  
}
```

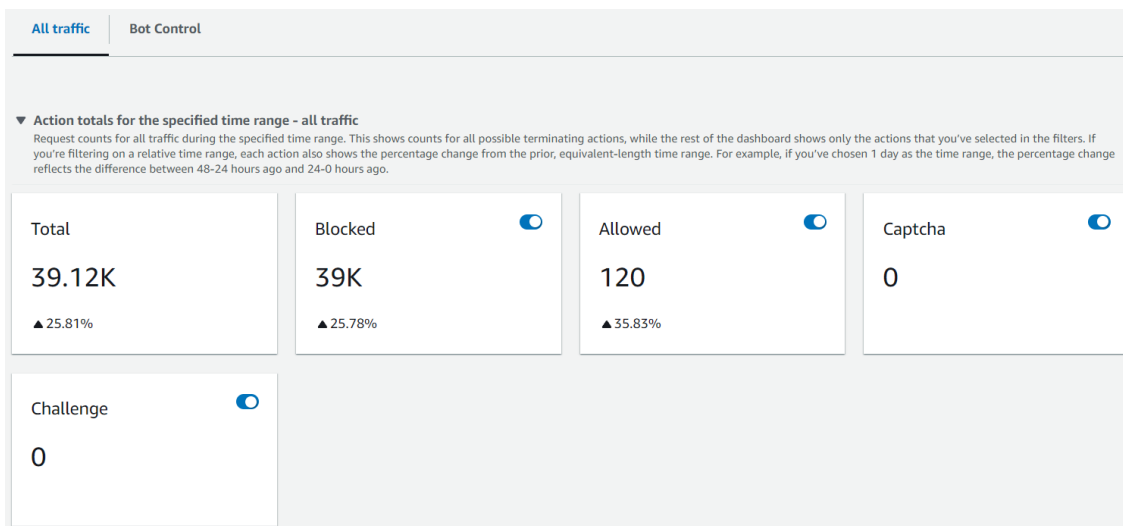
18 PRIEDAS

Šiame priede pateikta Apache Bench įrankio išvestis simuliuojant sistemos trikdymo ataką. Taip pat pateikta vaizdinė medžiaga apie tai, kaip WAF ugniasienė susitvarkė su siunčiamomis užklausomis.

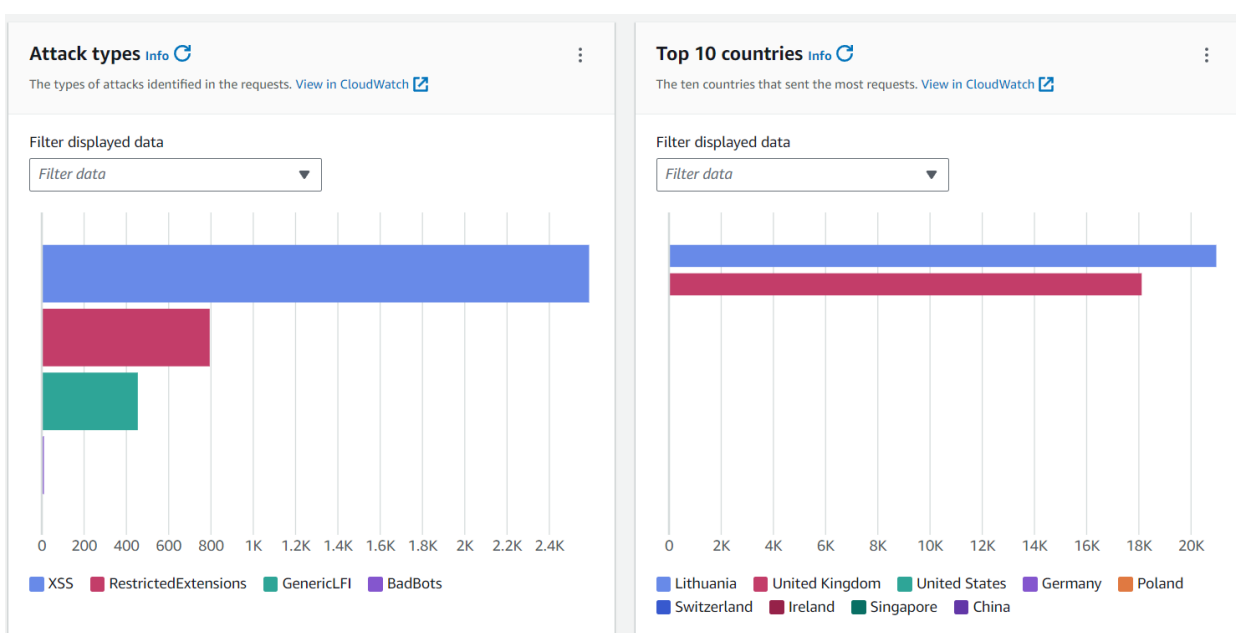
```
karolis@BTTLT117:/mnt/c/Users/karolis.zivulskas/Desktop$ ab -n 5000 -c 100
https://whoami.zero-trust-karolis.com/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking whoami.zero-trust-karolis.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests
Server Software:      CloudFront
Server Hostname:      whoami.zero-trust-karolis.com
Server Port:          443
SSL/TLS Protocol:     TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128
Server Temp Key:      X25519 253 bits
TLS Server Name:      whoami.zero-trust-karolis.com
Document Path:        /
Document Length:      919 bytes
Concurrency Level:    100
Time taken for tests:  43.388 seconds
Complete requests:    5000
Failed requests:      0
Non-2xx responses:    5000
Total transferred:    7425000 bytes
HTML transferred:     4595000 bytes
Requests per second:  115.24 [#/sec] (mean)
Time per request:     867.770 [ms] (mean)
Time per request:     8.678 [ms] (mean, across all concurrent requests)
Transfer rate:        167.12 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:     30    477 1274.8      45   15760
Processing:   10     15    6.5       13     66
Waiting:     10     14    5.3       13     65
Total:       42    492 1274.5      60   15772

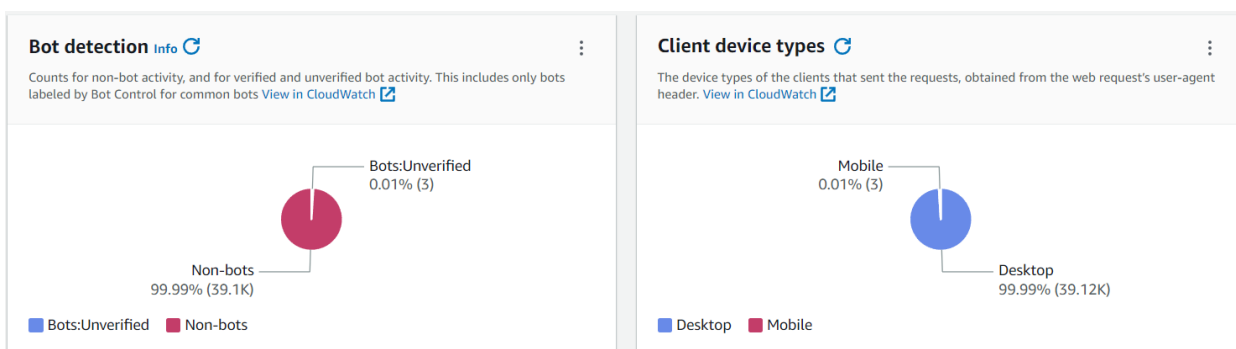
Percentage of the requests served within a certain time (ms)
 50%    60
 66%    77
 75%   102
 80%   1065
 90%   1137
 95%   3170
 98%   3230
 99%   7229
100%  15772 (longest request)
```



4.4.2.1 pav. Informacija apie užklausų blokavimą AWS WAF ugniasienėje.



4.4.2.2 pav. Informacija apie užklausų skirstymą pagal atakos tipą bei užklauskos kilmės šalį.



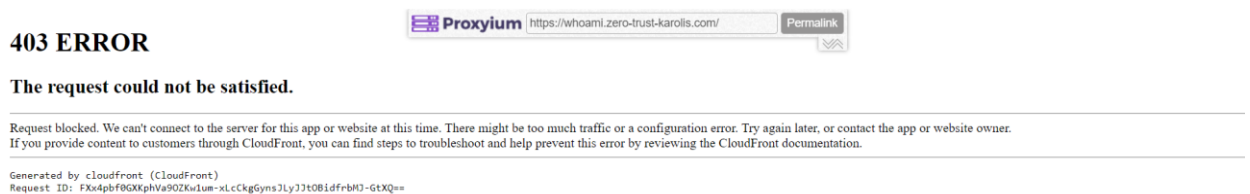
4.4.2.3 pav. Informacija apie botus bei prietaiso, siunčiančio užklausą, tipą.

19 PRIEDAS

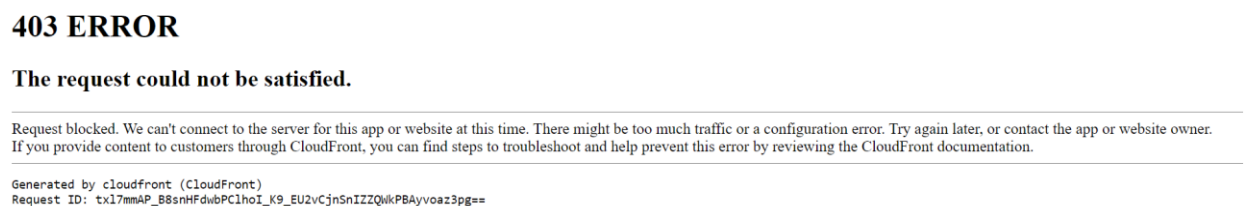
Šiame priede pateikiama vaizdinė medžiaga užklausų siuntimui iš apribotos geografinės lokacijos ar apriboto IP adresų sąrašo.



4.4.3.1 pav. Panaudotas Proxyium internetinis įrankis, suteikiantis galimybę naudotis Singapūro geografinė vietoje.



4.4.3.2 pav. Užklausa į whoami.zero-trust-karolis.com blokuota.



4.4.3.3 pav. Užklausa iš asmeninio kompiuterio blokuota.

20 PRIEDAS

Šiame priede pateikiama vaizdinė medžiaga apie RDS pasiekimo iš išorės tyrimo dalį.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-002dbc16af1b60768	PostgreSQL	TCP	5432	Custom	Q	Ingress from RDS to EKS Delete
sgr-030346c7dc63bf20c	PostgreSQL	TCP	5432	Custom	Q sg-0a2f5411f08db8a4e X	jump host Delete
					10.0.101.171/32 X	

Add rule

Cancel Preview changes Save rules

4.4.4.1 pav. Lokalios ugniasienės taisyklė.

```
ubuntu@ip-10-0-101-171:~$ psql -h prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com -p 5432 -U postgres zerotrust
Password for user postgres:
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 14.12)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.
```

zerotrust=>

i-078b47b7352bd39c8 (jump host)
PublicIPs: 54.171.49.47 PrivateIPs: 10.0.101.171

4.4.4.2 pav. Pavykęs prisijungimas iš virtualios mašinos, kurią leidžia ugniasienė.

```
karolis@BTTLT117:~$ psql -h prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com -p 5432 -U postgres zerotrust
psql: error: connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (10.0.103.68), port 5432 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (205.251.192.186), port 5432 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (205.251.194.60), port 5432 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (205.251.197.227), port 5432 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (205.251.199.100), port 5432 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
connection to server at "prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com" (2600:9000:5305:e300::1), port 5432 failed: Network is unreachable
Is the server running on that host and accepting TCP/IP connections?
karolis@BTTLT117:~$
```

4.4.4.3 pav. Nepavykęs prisijungimas iš asmeninio prietaiso.

21 PRIEDAS

Šiame priede pateikiamas S3 talpyklos atsakas į bandymą nelegaliai pasiekti talpyklos programinę sąsają.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error><Code>AccessDenied</Code><Message>Access
Denied</Message><RequestId>BAGEG59E8J12VNC4</RequestId><HostId>Fpk3ulJtuDY694ES8QfpWR1
mk2YOWarYIuBOXVCZL2cz83Mug3J3JJ7yZXxv7sxcqTUqQKiUqLg=</HostId></Error>
```

22 PRIEDAS

Šiame priede pateikiamas bandymo jungtis prie AWS EC2 rezultatas, kuomet naudojami neteisėtai perimti SSH raktai.

```
karolis@BTTLT117:~$ ssh ubuntu@ec2-52-19-103-163.eu-west-1.compute.amazonaws.com -i .ssh/zero-trust-jumphost-key.pem
ssh: connect to host ec2-52-19-103-163.eu-west-1.compute.amazonaws.com port 22: Connection timed out
karolis@BTTLT117:~$ █
```

4.4.6.1 pav. Nepavykęs prisijungimas prie EC2 virtualios mašinos.

23 PRIEDAS

Šiame priede pateiktas bandymas iš užklausų skirstytuvo išgauti klasterio mazgų adresus.

```
karolis@BTTLT117:/mnt/c/Users/karolis.zivulskas/Desktop$ ping whoami.zero-trust-karolis.com
PING whoami.zero-trust-karolis.com (18.66.233.4) 56(84) bytes of data.
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=1 ttl=246 time=10.3 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=2 ttl=246 time=10.3 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=3 ttl=246 time=9.90 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=4 ttl=246 time=10.5 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=5 ttl=246 time=10.6 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=6 ttl=246 time=10.4 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=7 ttl=246 time=10.8 ms
^C
--- whoami.zero-trust-karolis.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 9.899/10.395/10.825/0.270 ms
karolis@BTTLT117:/mnt/c/Users/karolis.zivulskas/Desktop$ ping whoami.zero-trust-karolis.com
PING whoami.zero-trust-karolis.com (18.66.233.4) 56(84) bytes of data.
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=1 ttl=246 time=10.1 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=2 ttl=246 time=10.7 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=3 ttl=246 time=9.58 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=4 ttl=246 time=10.1 ms
64 bytes from server-18-66-233-4.waw51.r.cloudfront.net (18.66.233.4): icmp_seq=5 ttl=246 time=10.1 ms
```

4.4.7.1 pav. Ping komandos užklausos į whoami.zero-trust-karolis.com subdomeną, siekiant pamatyti skirtingus mazgų adresus.

24 PRIEDAS

Šiame priede pateikta vaizdinė bandymo dalis, kurioje analizuoti sukonfigūrotų dirbtinio intelekto įrankių pastebėjimai bei patarimai.

☐	Finding	Severity	Workflow status	Record State	Region	Account Id	Company	Product	Resource
☐	CVE-2023-6597 - python3-libs, python3	HIGH	NEW	ACTIVE	eu-west-1	113101838846	Amazon	Inspector	EC2 Instance i-0b1d839e0b0cdf34a
☐	CVE-2023-6597 - python3-libs, python3	HIGH	NEW	ACTIVE	eu-west-1	113101838846	Amazon	Inspector	EC2 Instance i-0ee56e308866b83d2
☐	A Kubernetes API commonly used in Discovery tactics invoked from a known malicious IP address.	MEDIUM	NEW	ACTIVE	eu-west-1	113101838846	Amazon	GuardDuty	AwsEksCluster prod-zero-trust-cluster
☐	A Kubernetes API commonly used in Discovery tactics invoked from a known malicious IP address.	MEDIUM	NEW	ACTIVE	eu-west-1	113101838846	Amazon	GuardDuty	AwsEksCluster prod-zero-trust-cluster

4.4.8.1 pav. Grėsmės, susijusios su Python bibliotekomis.

☐	Ensure IAM password policy requires at least one number	MEDIUM	NEW	ACTIVE	eu-west-1	113101838846	AWS	Security Hub	Account 113101838846
☐	Ensure IAM password policy requires at least one symbol	MEDIUM	NEW	ACTIVE	eu-west-1	113101838846	AWS	Security Hub	Account 113101838846
☐	Ensure IAM password policy requires at least one uppercase letter	MEDIUM	NEW	ACTIVE	eu-west-1	113101838846	AWS	Security Hub	Account 113101838846

4.4.8.2 pav. Grėsmės, susijusios su IAM slaptažodžių kūrimo reikalavimais.

Roles and users with the most API call volume Info							Export
< 1 2 3 >							
Principal (role or user)	AWS account	Trend (7 days)	Success	Failure	Total		
AWSReservedSSO_AdministratorAccess_ec3914d90436bdc2 AWS role	113101838846		2,723	45	2,768		
prod-zero-trust-cluster-eks-nodes AWS role	113101838846		2,083	2	2,085		
AWSServiceRoleForAmazonEKSNodegroup AWS role	113101838846		1,728	0	1,728		
prod-zero-trust-cluster-autoscaler AWS role	113101838846		1,430	0	1,430		
AWSServiceRoleForConfig AWS role	113101838846		1,277	113	1,390		
AWSServiceRoleForSecurityHub AWS role	113101838846		1,013	20	1,033		
AWSServiceRoleForAutoScaling AWS role	113101838846		737	0	737		

4.4.8.3 pav. Detective informacija apie naudotojų bei rolių atliekamas užklausas AWS viduje.

EC2 instances with the most traffic volume Info							Export
< 1 >							
EC2 instance	AWS account	Trend (7 days)	Bytes in	Bytes out	Total		
i-0ee56e308866b83d2	113101838846		649 MB	267 MB	916 MB		
i-0b1d839e0b0cdf34a	113101838846		501 MB	165 MB	666 MB		
i-078b47b7352bd39c8	113101838846		12.8 MB	967 kB	13.8 MB		
i-0701ae5b3eb8e83cd	113101838846		746 kB	477 kB	1.22 MB		

4.4.8.4 pav. Detective informacija apie užklausų srautus virtualiose mašinose.

Finding status

Active

Filter criteria

Q

Add filter

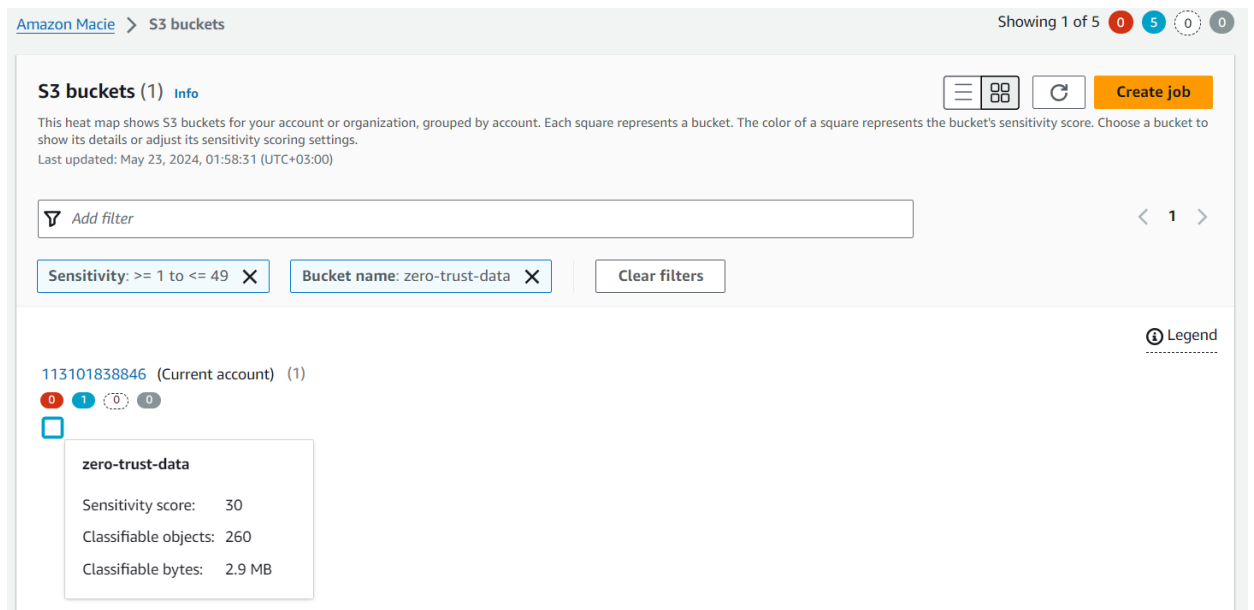
Resource type: AWS EC2 Instance

Clear filters

< 1 >

	Severity	Title	Impacted resource	Type	Age	Status
☐	High	CVE-2023-6597 - python3-libs, python3	i-0ee56e308866b83d2	Package Vulnerability	4 days	Active
☐	High	CVE-2023-6597 - python3-libs, python3	i-0b1d839e0b0cdf34a	Package Vulnerability	4 days	Active
☐	Medium	CVE-2024-26583 - kernel-headers, kernel-devel and 1 more	i-0ee56e308866b83d2	Package Vulnerability	a day	Active
☐	Medium	CVE-2024-26584 - kernel-headers, kernel-devel and 1 more	i-0ee56e308866b83d2	Package Vulnerability	a day	Active
☐	Medium	CVE-2024-26583 - kernel-headers, kernel-devel and 1 more	i-0b1d839e0b0cdf34a	Package Vulnerability	a day	Active
☐	Medium	CVE-2024-26584 - kernel-headers, kernel-devel and 1 more	i-0b1d839e0b0cdf34a	Package Vulnerability	a day	Active

4.4.8.5 pav. Inspector rasti pažeidžiamumai bei jų rizikos lygiai.



4.4.8.6 pav. Macie įvertintas testavimo duomenų jautrumas.

25 PRIEDAS

Šioje dalyje pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą AWS paskyros prisijungimą.

Sign in as IAM user

Account ID (12 digits) or account alias

113446850438

IAM user name

Zero-trust-user

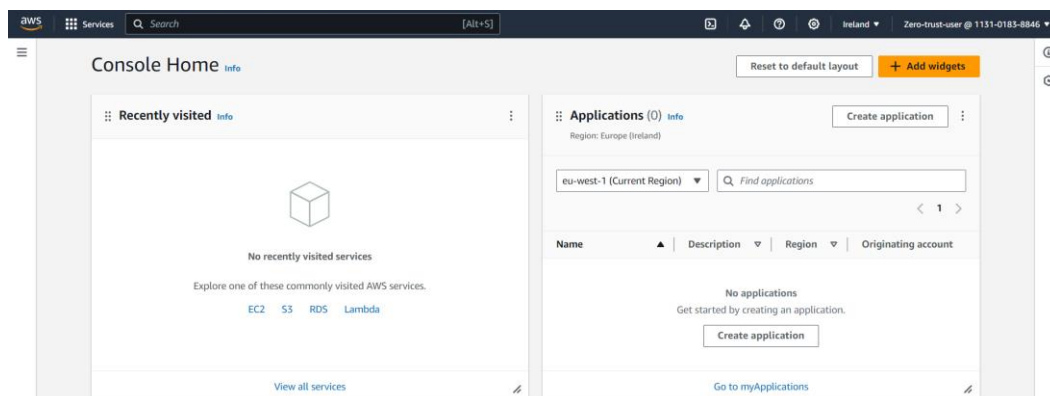
Password

.....

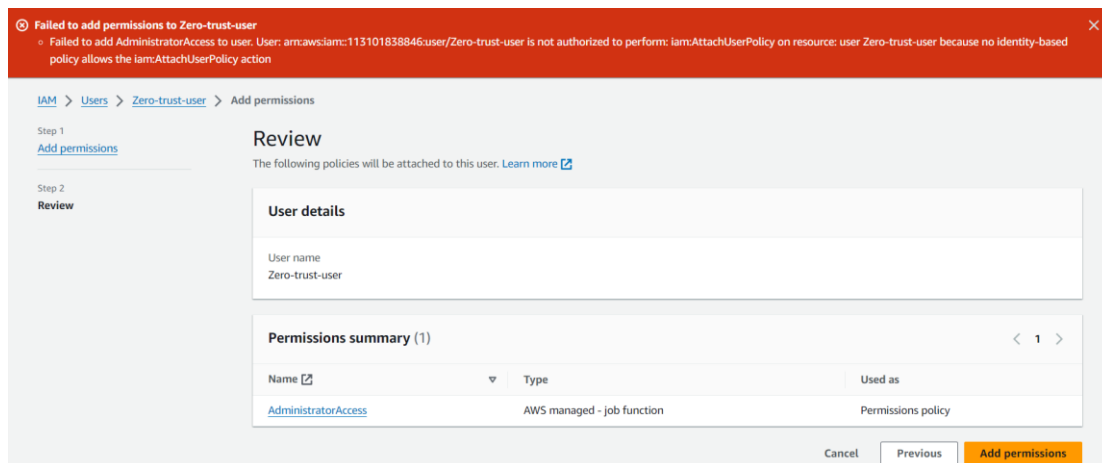
☐ Remember this account

Sign in

4.4.1.4 pav. Prisijungimo bandymas.



4.4.1.5 pav. Pavykęs prisijungimas.



4.4.1.6 pav. Nepavykęs IAM prieigų teisių keitimas.

26 PRIEDAS

Šioje dalyje pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą subdomeną. Taip pat pateikta sistemos trikdymo atakos komandos išvestis.

The screenshot shows the AWS Route 53 console. The 'Records (5)' tab is selected. A table lists DNS records. The record 'whoami.zero-tr...' is highlighted. To the right, the details for this record are shown: Record name is 'whoami.zero-trust-karolis.com', Record type is 'A', Value is 'zero-trust-nlb-loadbalancer-00cbcc5f6f8bf1c.elb.eu-west-1.amazonaws.com', Alias is 'Yes', TTL (seconds) is '-', and Routing policy is 'Simple'.

Record name	Type	Routing policy	Alias	Value/Route traffic
zero-trust-karol...	NS	Simple	No	ns-332.awsdns-41.c ns-1198.awsdns-21 ns-1814.awsdns-34 ns-837.awsdns-40.r
zero-trust-karol...	SOA	Simple	No	ns-332.awsdns-41.c
_39229138dd5...	CNAME	Simple	No	_46c75d9fe970f34f
whoami.zero-tr...	A	Simple	Yes	zero-trust-nlb-load
_57c7ed3b2e8...	CNAME	Simple	No	_eac63e81ce8b6d8

4.4.2.4 pav. Subdomeno kreipimas tiesiai į užklausų skirstytuvą DNS A įrašu.

The left part of the screenshot shows a terminal window with the output of a curl command. The output includes the host 'whoami.zero-trust-karolis.com' and the IP address '10.0.1.62'. The right part shows the Chrome DevTools Network tab with a single request to 'https://whoami.zero-trust-karolis.com/'. The request is successful (Status: 200 OK) and the response is '10.0.1.62'.

4.4.2.5 pav. Programa simuliutos atakos metu suletėjo.

```
karolis@BTTLT117:~$ ab -n 5000 -c 100 -s 300 https://whoami.zero-trust-karolis.com/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking whoami.zero-trust-karolis.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests
Server Software:
Server Hostname:      whoami.zero-trust-karolis.com
Server Port:          443
SSL/TLS Protocol:     TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128
Server Temp Key:      X25519 253 bits
TLS Server Name:      whoami.zero-trust-karolis.com
Document Path:        /
Document Length:      484 bytes
Concurrency Level:     100
Time taken for tests:  156.781 seconds
Complete requests:     5000
Failed requests:       0
```

```

Total transferred:      3425000 bytes
HTML transferred:      2420000 bytes
Requests per second:    31.89 [#/sec] (mean)
Time per request:       3135.620 [ms] (mean)
Time per request:       31.356 [ms] (mean, across all concurrent requests)
Transfer rate:          21.33 [Kbytes/sec] received

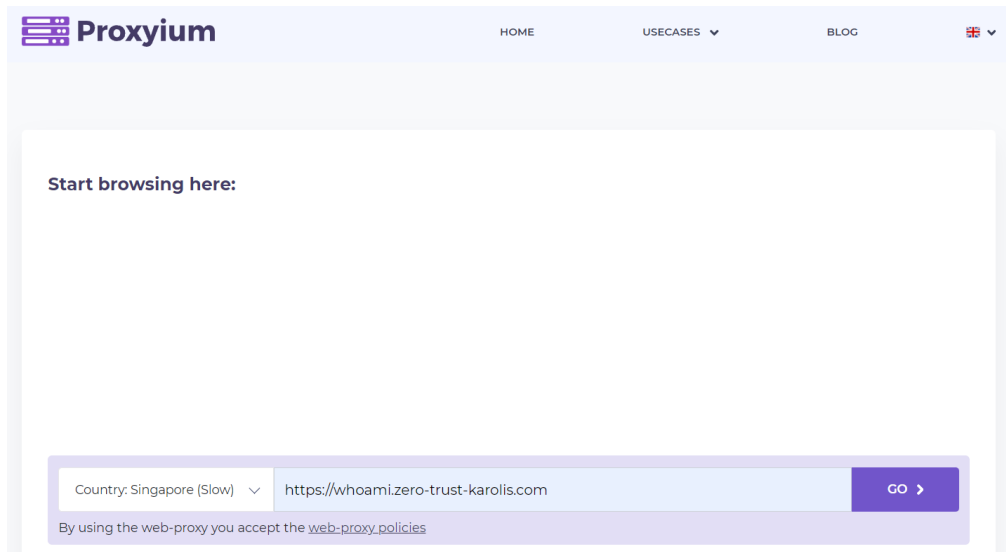
Connection Times (ms)
      min  mean[+/-sd] median   max
Connect:    143  2975 12770.2    364   66022
Processing:   45   152  108.3     117     810
Waiting:     45   151  108.3     117     810
Total:      192  3127 12762.7    510   66202

Percentage of the requests served within a certain time (ms)
 50%    510
 66%    572
 75%    605
 80%    635
 90%    789
 95%    971
 98%   65658
 99%   66054
100%   66202 (longest request)

```

27 PRIEDAS

Šiame priede pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą subdomeną dėl užklausų iš skirtingų geografinių lokacijų.



4.4.3.4 pav. Proxyium įrankis naudojamas Singapūro geografinės vietovės atkartojimui.



4.4.3.5 pav. Subdomenas pasiektas iš Singapūro geografinės vietovės.

28 PRIEDAS

Šioje dalyje pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą AWS RDS duomenų bazę.

```
ubuntu@ip-172-31-10-220:~$ psql -h prod-postgres-rds-instance.cdku8wrodvth.eu-west-1.rds.amazonaws.com -p 5432 -U postgres zerotrust
Password for user postgres:
psql (16.2 (Ubuntu 16.2-1ubuntu4), server 14.12)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, compression: off)
Type "help" for help.

zerotrust=>
```

4.4.4.4 pav. Pavykęs prisijungimas iš asmeninio prietaiso.

```
zerotrust=> \du
```

Role name	List of roles	Attributes
postgres	Create role, Create DB	+
	Password valid until infinity	
rds_ad	Cannot login	
rds_iam	Cannot login	
rds_password	Cannot login	
rds_replication	Cannot login	
rds_superuser	Cannot login	
rdsadmin	Superuser, Create role, Create DB, Replication, Bypass RLS+	
	Password valid until infinity	
rdsrepladmin	No inheritance, Cannot login, Replication	
rdstopmgr		

4.4.4.5 pav. Galėjimas vykdyti komandas viduje duomenų bazės.

Šioje dalyje pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą AWS S3 talpyklą. Taip pat pridėtas komandos `curl` atsakas į bandymą pasiekti testavimo duomenų failą S3 talpykloje.

4.4.5.1 pav. Pavykęs S3 talpyklos pasiekimas `curl` komanda.

85

30 PRIEDAS

Šioje dalyje pateikti vaizdiniai priedai, gauti testuojant sukurta saugumo sistema neapsaugotą AWS EC2 virtualią mašiną.

```
karolis@BTTLT117:~/ssh$ ssh ubuntu@ec2-52-19-103-163.eu-west-1.compute.amazonaws.com -i zero-trust-jumphost-key.pem
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu May 23 13:39:24 UTC 2024

System load:  0.08               Processes:    108
Usage of /:   27.6% of 6.71GB    Users logged in: 1
Memory usage: 24%               IPv4 address for enX0: 172.31.10.220
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
20 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu May 23 13:12:26 2024 from 18.202.216.52
ubuntu@ip-172-31-10-220:~$
```

4.4.6.2 pav. Pavykęs prisijungimas prie EC2 virtualios mašinos.