TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Sander Plukš 206070 IAAB

Küberkaitse kursuste automaatne juurutamine pilveteenuste platvormil

Bakalaureusetöö

Juhendaja: Siim Vene
Magistrikraad
Kaasjuhendaja: Randel Raidmets
Magistrikraad

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Sander Plukš

13.05.2024

Annotatsioon

Käesoleva bakalaureusetöö käigus uuriti erinevaid infrastruktuur kui kood tarkvarasid, et leida ettevõttele parim tööriist küberkaitse kursuste automaatseks juurutamiseks pilveteenustes. Tööriista kasutatakse ettevõtte loodud automatiseerimisplatvormiga vLM-Next, et juurutada küberkaitse kursuste süsteeme.

Pilveteenuste platvormiks valiti Amazon Web Services, kus uuriti saadaolevaid teenuseid süsteemide juurutamiseks pilveplatvormil. Teenuste põhjal koostati kavand, mida oleks võimalik kasutada prototüübi loomisel.

Automatiseerimistööriistade analüüsiga võrreldi esialgu populaarsemaid tarkvarasid, mille tulemusel testiti kolme tööriistaga infrastruktuuri keskkonna ja virtuaalmasinate pilve juurutamist. Testimise käigus osutus valituks Ansible, mis vastas seatud nõuetele.

Prototüübi loomisel kasutati Ansible tarkvara ning ettevõtte vLM-Next platvormi, et viia läbi privaatse virtuaalse pilve ning küberkaitse kursuse süsteemide juurutamine ja seadistamine. Tulemusena kinnitati, et prototüüp vastab nõuetele ning töö eesmärk sai saavutatud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 6 peatükki, 4 joonist, 1 tabelit.

Abstract

Automatic Deployment of Cybersecurity Courses on a Cloud Service Platform

This bachelor's thesis explores popular Infrastructure as Code (IaC) tools to determine the optimal solution for a company to automate deployment of cybersecurity courses on a cloud service platform. Research focuses on selecting a tool that could integrate with the company's existing automation platform vLM-Next to deploy cybersecurity course systems.

Amazon Web Services (AWS) was chosen as the cloud service platform. Study involved analyzing services offered by AWS that could be utilized for deploying systems on the platform. Based on these services, a draft was created that could be used for developing a prototype.

Analysis was conducted with examining the most popular IaC tools, which led to comparing three tools – Ansible, Terraform and AWS Cloudformation. Further analysis continued with testing these tools for deploying infrastructure environments and customizing instances on AWS. Ansible was selected as the preferred tool as it met the established requirements.

Prototype was developed using Ansible software together with the company's orchestration tool, vLM-Next, to facilitate the deployment and configuration of a virtual private cloud and cybersecurity course systems. The successful outcome confirmed that the prototype met all requirements and achieved the objectives of the thesis, demonstrating efficiency of using Ansible for automation within a cloud environment.

The thesis is in Estonian and contains 25 pages of text, 6 chapters, 4 figures, 1 tables.

Lühendite ja mõistete sõnastik

ACL	*Access Control List*, kasutajaile ja/või protsessidele kinnistatud pääsuõiguste andmestik
AD	*Active Directory*, kataloogiteenus Microsofti serverites kasutajakontode haldamiseks
AMI	*Amazon Machine Image*, Amazon kettapildifail või mall
ASA(v)	*Cisco Adaptive Security (Virtual) Appliance*, tulemüür (virtuaalsetele) keskkondadele
AWS	*Amazon Web Services*, Amazon pilveplatvorm
CRP	*Cyber Range* Platform, küberharjutusplatvorm
CTF	*Capture the Flag*, küberkaitse kursuse tüüp, kus ülesande lahendamiseks otsitakse süsteemist virtuaalset lippu
DNS	*Domain Name System*, domeeninimede süsteem
EC2	*Amazon Elastic Compute Cloud*, virtuaalmasinate loomise teenus
HCL	*Hashicorp Configuration Language*, Hashicorp konfiguratsioonikeel
IaC	*Infrastructure as Code*, infrastruktuuri ressursside haldamise ja juurutamise protsess
IAM	*Amazon Identity and Access Management*, kasutajate ja reeglite kogumiku pääsuhaldus
ISA	*Integrated Scoring and Awareness* Tool, platvorm, mis pakub kasutajatele visuaalset harjutuse- või kursuse keskkonda
IP	*Internet Protocol*, internetiprotokoll
SaaS	*Software as a Service*, tarkvara teenusena
SSH	*Secure Shell*, võrguprotokoll turvaliseks võrguteenuste opereerimiseks turvamata võrgu kaudu
vLM	*Virtual Lab* Manager, platvorm mängustsenaariumi süsteemide automaatseks seadistamiseks ja juurutamiseks
VNC	*Virtual Network Computing*, graafiline töölaua jagamissüsteem
VPN	*Virtual Private Network*, virtuaalne privaatvõrk
YAML	*Yet another markup language*, andmete jadastuse keel

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Küberjulgeolek ja valmidus reageerida järjest sagenevatele küberrünnakutele on reaalsus, millega ühiskond, ettevõtted ja riigiasutused peavad kohanema. See on valdkond kuhu tuleb aktiivselt panustada tänapäeva infoajastul ning seetõttu on autori ajendiks käesolev teema, mida teostab ta ettevõttes CybExer Technologies.

Ettevõtte peamiseks tooteks on küberharjutusplatvorm, mida on võimalik kasutada mitmeotstarbeliselt. Platvormi erinevate kasutusvõimaluste hulka kuuluvad nii testkeskkondade loomine, ülimalt realistlikud õppused- ja kursused kui ka spetsiifilistele kliendisoovidele vastavad lahendused. Väljatoodud näited vajavad optimeeritud ressursside kasutust ja automatiseerimist, olenemata sellest, kas harjutusplatvorm asub pilves või on selleks kohapealne infrastruktuur. Eeltoodud näidete efektiivseks teostamiseks on firma tehnoloogiapinus mitmeid väljaarendatud tööriistu, mille abil on läbi viidud hulga kursuseid ning treeninguid.

Ettevõttes käimasoleva automatiseerimistööriista uue väljalaske raames on jõutud punkti, kus tekib uusi võimalusi kasutada toote potentsiaali lahenduste juurutamiseks pilveplatvormidele. Käsitletav probleem seisneb aga ühtse lähenemise puudumises, millega pilveteenuste platvormile automaatselt erinevat tüüpi sisu paigaldada.

Lõputöö eesmärgiks on leida ettevõtte automatiseerimisplatvormile infrastruktuur kui kood tarkvara, millega juurutada küberkaitse kursuseid pilveteenuste platvormile. Eesmärgi täitmiseks analüüsitakse olemasolevat olukorda, defineeritakse nõuded loodavale prototüübile, võrreldakse erinevaid automatiseerimistööriistu ning luuakse funktsionaalne prototüüp.

2 Olemasoleva olukorra kirjeldus ja nõuded

Käesolevas peatükis kirjeldatakse olemasolevat olukorda ja nõudeid, mis on seatud ettevõtte poolt antud tööle. Tutvustatakse ettevõtte küberharjutusplatvormi ülesehitust ja arhitektuuri. Arvesse võetakse CybExer Technologies poolt loodud platvorme, et analüüsi käigus leitud automatiseerimistarkvara sobituks kirjeldatud toodetega, võimaldades sujuvat integratsiooni ja efektiivset kasutust ka edaspidi.

2.1 Ettevõttest CybExer Technologies

CybExer Technologies OÜ on NATO auhinnaga pärjatud Eesti küberturvalisuse ettevõte. Ettevõttel on laiaulatuslikud kogemused kõrgetasemeliste küberturvalisuse koolitusplatvormide pakkumisel ja hooldamisel, keskendudes eelkõige kübervõimekuse arendamisele. Ettevõtte pakutav küberharjutusplatvorm on võtmetähtsusega küberturvalisuse koolituste ja õppuste edukaks läbiviimiseks, suunatuna paljudele tavakasutajatele, tehnilistele kasutajatele ja strateegilise taseme juhtidele [1].

Ettevõtte tegutseb globaalses küberturvalisuse valdkonnas, kus valitseb nõudlus kvaliteetsete ja kõrge võimekusega praktiliste lahenduste järele.

Peamised tegevusvaldkonnad on järgnevad [1]:

- küberharjutusväljakute tehnoloogia arendamine ja müük;
- kübertreeningute ja -õppuste korraldamine;
- testkeskkondade arendamine;
- küberturvalisusega seotud lisateenuste müük

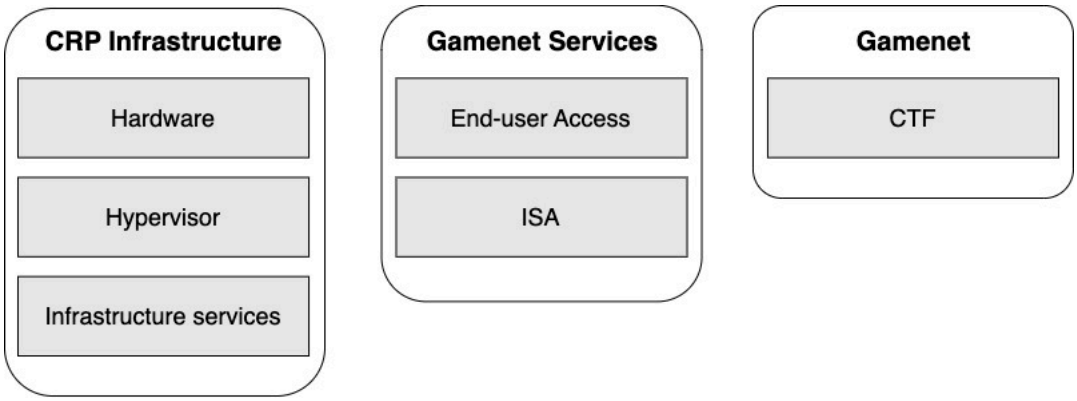
2.2 CybExer Technologies küberharjutusplatvorm

Küberharjutusplatvorm sisaldab mitmeid teenuseid, mis aitavad toimida sel eesmärgipäraselt ning luua sinna õppusteks, harjutusteks, treeninguteks ja kursusteks

sisu. Selle taustal töötab põhjalikult seotud komplekt erinevaid tarkvarasid, et platvorm võimaldaks sellist laadi eesmärke üldse pakkuda. Järgnevalt on kirjeldatud CybExer Technologies küberharjutusplatvormi toimimist riistvaralise infrastruktuuri peal.

2.2.1 Küberharjutusplatvormi ülesehitus

Küberharjutusplatvormi ülesehituse võib jagada kolmeks segmendiks (Joonis 1). Esimeseks platvormi alusinfrastruktuuriks olevad teenused administraatoritele, kui ka harjutuste või treeningute sisu operaatoritele. Teiseks *gamenet* (edaspidi sisuvõrk) teenused, mis on seotud lõppkasutaja ligipääsu ning platvormidega, kus osalejad ülesandeid lahendavad jpm teevad. Kolmandaks sisuvõrk, mis sisaldab emuleeritud keskkondi platvormil, et jooksutada erinevaid harjutusi [2]. Sisuks nimetatakse kõike, mis asub sisuvõrgu tsoonis ja töö raames on selleks küberkaitse kursus.



Joonis 1. Küberharjutusplatvormi ülesehitus.

2.2.2 Cyber Range Platform (CRP) Infrastruktuur

CRP infrastruktuuri segmenti kuulub küberharjutusplatvormi vundament, mis majutab enda alla riistvara – serverid, kommutaatorid, andmetalletusseadmed. Selle peal asub hüperviisor, mis võimaldab virtualiseerimist. Selline variant on kasutusel kohapealse infrastruktuuri lahendusena.

Ettevõttel on mitmeid küberharjutusplatvorme ning neid kutsutakse *tenant*'iteks (platvormi alamkeskkondadeks). Alamkeskkondi võib vastavalt riistvaralisele ressursile ühe infrastruktuuri peal olla mitu. See võimaldab ettevõttel pakkuda *Software as a Service* (SaaS) lahendust, mis päästab kliendi riistvaralistest kohustustest ning lubab platvormi kiire kasutusmugavuse. Eksisteerib ka hübriidlahendus, kus on kasutusel kohapealne

infrastruktuur, kui ka vSphere pilveplatvormi variant [2]. See tähendab, et keskkonda juurutatud lõppkasutajate sisu jagatakse ära mõlema ressursi vahel.

2.2.3 *Gamenet-services* ja *Gamenets*

Sisuvõrgu teenuste tsoon väljendab süsteeme, mis on administraatorite haldusalas, kuid kasutatakse ka lõppkasutaja poolt. See hõlmab erinevaid teenuseid, mis seadistatakse küberkaitseharjutusteks või kursusteks, näiteks punktide jagamine, stsenaariumi juhendid, kommunikeerimisvahendid.

Sisuvõrk on lõppkasutajale suunatud stsenaariumi sisu. Selleks on tavaliselt kogumik virtuaalmasinaid, mis on omavahel seotud ja moodustavad terviku. Stsenaariumi järgi erinevad emuleeritud süsteemid tehniliselt päris palju olenevalt ülesandest ning sisuvõrgu tüübist. Käesolevas töös vaadeldakse ainult *Capture the Flag* (CTF) tüüpi kursuseid.

2.2.4 Küberharjutusplatvormi võrk

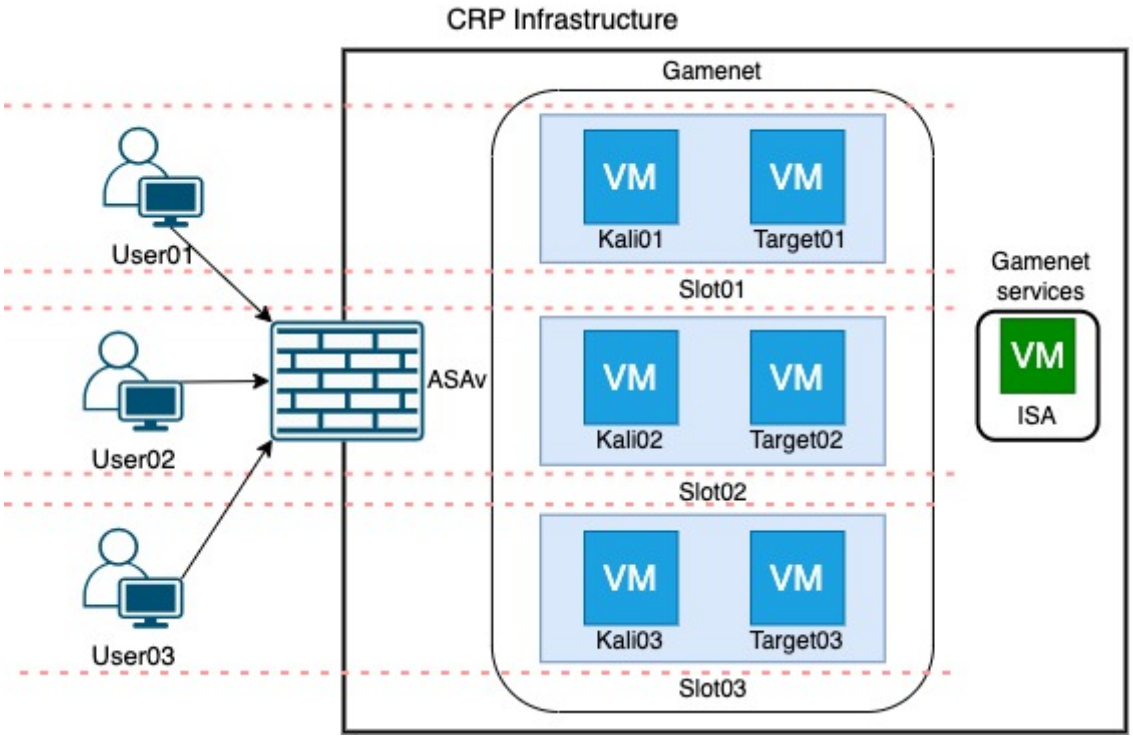
Võrk on kõige keerulisem osa platvormi infrastruktuurist, sest stsenaariumi sisu juurutatakse dünaamiliselt ja see nõuab ühendusi, mis suudavad käsitleda muutuvat olekut. Enamik põhilisi võrguteenuseid seoses marsruutimise ja tulemüüridega on virtualiseeritud, mis võimaldab neid dünaamiliselt seadistada [2].

Küberharjutusplatvormi võrgu saab jaotada kahte segmenti, CRP infrastruktuur ning sisuvõrk. Platvorm võimaldab kasutada simuleeritud internetti sisuvõrgu keskkonnas, et ühendada erinevate stsenaariumite süsteeme internetiga. Liiklus ja juurdepääs segmentidele või nende vahel on kontrollitud *Access Control List*'i (ACL) ehk pääsupiiramisloendiga [2].

2.2.5 Lõppkasutaja ligipääs küberharjutusplatvormile

Küberharjutusplatvormis kasutatakse kaugjuurdepääsuks *Cisco Secure Client* virtuaalse privaatvõrgu tunneli (VPN) tarkvara. Kasutaja roll ja juurdepääsuõigused platvormile on määratud vastavalt Microsoft *Active Directory* (AD) serveris konfigureeritud poliitikatele [2]. Kasutajale määratud grupp oleneb sisust, mis on juurutatud ettevalmistatud keskkonda. Näiteks küberharjutuse puhul määratakse kasutajale sinise meeskonna grupp või treeningu puhul treeningu grupp.

Lõppkasutaja sisselogimine platvormile toimub läbi juhendi VPNi abil kasutajatunnuse ja parooliga. Ühendutakse läbi ASA tulemüüri, mis autendib kasutaja ja jagab vastava IP aadressi. Olenevalt rollidest pääsetakse ligi nii teenustele, kuhu õigused lubavad, kui ka emuleeritud sisule, mis keskkonnas asub (Joonis 2).



Joonis 2. Kasutaja ligipääs.

2.3 Lõputööga seotud ettevõtte tehnoloogiapinu platvormid

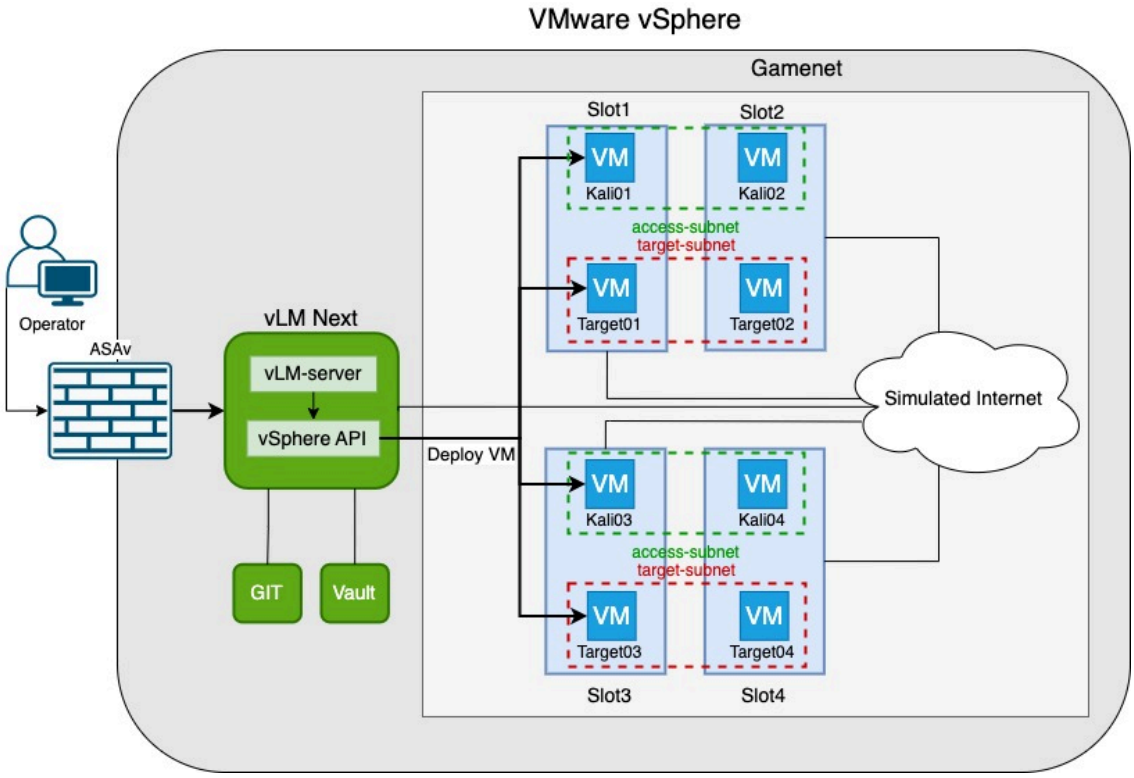
Lõputöös käsitletakse kahte põhilist platvormi, mis on CybExer Technologies poolt arendatud. Mõlemal on erinev eesmärk toetamaks küberkaitse kursuste läbiviimist.

2.3.1 Virtual Lab Manager-Next

Virtual Lab Manager (vLM) on autoriõigustega kaitstud tarkvara ja üks mitmest alamsüsteemist ning on üks osa CybExeri küberharjutusplatvormist. See on oluline veebirakendus ja keskne osa, mis pakub operaatoritele automaatlahendust mängustsenaariumi süsteemide automaatsel juurutamisel ja haldamisel. Täpsemalt on tööriist mõeldud sisuvõrgu haldamiseks ja seadistamiseks infrastruktuuri keskkonda. Sellega on võimalik luua realistlikke keskkondi harjutusteks, koolitusteks ja testimiseks.

vLM on paljuski varasemalt sõltunud VMware spetsiifilisest infrastruktuurist, mida kasutatakse läbi vSphere rakendusliidese. VMware pakub virtualiseerimise ja andmetöötluse tarkvara, pakkudes paindlikkust, mastabeeritavust ja turvalisust [3]. Sisuvõrgu süsteemide juurutamisel virtualiseerimiskeskkonda on selle haldamine piirdunud erinevate skriptidega. Sellisel viisil emuleeritud sisu haldamine tekitas omajagu probleeme, näiteks puudulikud logiväljundid, pikad ja keerulised skriptid ning pikk juurutusprotsess. Probleemide eemaldamiseks ja protsessi parandamiseks on arendatud toote uus versioon.

CybExeri uueks väljalaskeks on vLM-Next, mis on orienteeritud toetama mitmeid erinevaid virtualiseerimis- ja pilveplatvorme. Selline lähenemine võimaldab kasutada erinevaid teenusepakkujaid, millega välditakse sõltuvust ühest infrastruktuuri platvormist. vLM-Next koosneb mitmest erinevast komponendist ning suhtleb erinevate teenustega tagataustal (Joonis 3). Joonisel on välja toodud olemasolev VMware vSphere lahendus.



Joonis 3. vLM-Next orkestraator.

Operaatoritele avaneb graafiline kasutajaliides, kus saab seadistada sisuvõrgu süsteemide definitsioone automaatseks juurutamiseks erinevatel platvormidel. Taustal teeb põhilise

töö ära vLM-Server, millel on otseintegratsioon VMware vSphere'iga läbi vSphere rakendusliidese. Edasiminek on võimekus jooksutada Docker konteinereid, mille abil saab integreerida tarkvarasid sisuvõrgu seadistamiseks. vLM Nextiga on veel lõimitud Gitea koodihoidla ja HashiCorp Vault paroolihaldur, mis tagavad vastavalt süsteemide versioonihalduse ning paroolide või võtmete turvalise hoiustamise.

Lihtsustades toimib vLM Next *inventory* serverina, ehk kohana, kus kirjeldatakse keskkonna sisu ressursse. Keskkonna sisuna seadistatakse näiteks sisuvõrgu alamvõrgud, ühendused simuleeritud internetiga ning virtuaalmasinad ise piltlikult kirjeldatud pesadesse. Seejärel on mõeldud konteineris töötav automatiseerimistarkvara juurutama kirjeldatud olekut.

2.3.2 Integrated Scoring and Awareness Tool

Integrated Scoring and Awareness Tool (ISA) on CybExer Technologies arendatud tarkvara, mis pakub peaaegu reaalajas punktide andmist, olukorrateadlikuse visualiseerimist, ülesannete lahendamist, raporteerimist, ning vastuste esitamist [2].

Platvorm on mõeldud eelkõige esimese kontaktpunkti tööriistaks, kus osalejad kursusega algust teevad. Kasutajateks võivad olla küberharjutuse erinevad meeskonnad, kellel on ühised vaated, kui ka üksikisikud, kes lahendavad ülesandeid individuaalselt. Ülesannete lahendamiseks luuakse ISA veebirakenduses harjutusmoodul.

2.4 Capture the Flag

Capture the Flag võistlused küberkaitses koosnevad erinevatest alamtüüpi ülesannetest, näiteks pöördkonstrueerimisest, veebitehnoloogiatest, krüptograafiast või arvutikriminalistikast jne [4]. Tüüpiliselt peavad osalejad leidma arvutisüsteemist turvanõrkusi või lahendama kindla ülesande, et leida *Flag* ehk lipp. Selleks võib olla peidetud informatsioon või andmed, mis esitatakse vastusena. Selliste ülesannete lahendamine arendab ja kinnistab praktilisi oskusi, kui ka teadmisi ning tõstab osalejate enesekindlust küberturvalisuses [5].

2.4.1 Tehniline kirjeldus

Lõpplahenduses käsitletav CTF kursus, mida prototüübi käigus juurutama hakatakse, koosneb staatilistest süsteemidest, kuhu on peidetud tagaotsitavad lipud. Lipud on

paigaldatud Docker konteineritesse, mis omakorda sisaldavad veebilehti, logifaile või võrgupakette, mida ülesannete lahendamiseks analüüsida. Konteinerid asuvad Linux distributsiooniga virtuaalmasinas, mida kutsutakse *target*'iks (edaspidi sihtmärk).

Lahendamiseks on osalejatel Kali Linux süsteemid. Kali Linux on avatud lähtekoodiga Debianil põhinev Linuxi distributsioon, mis on mõeldud läbistustestimiseks ja turvaauditite läbiviimiseks [6].

Kõigile osalejatel on määratud eraldiseivad virtuaalkeskkonnad, kust pääsetakse ligi vaid oma ressurssidele ja internetile. Keskkonnad on üksteisest eraldatud ja piiratud virtuaalsete alamvõrkude ning tulemüüri reeglite abil, et vältida ligipääsu mõne teise osaleja süsteemi.

2.5 Nõuded

Nõuded prototüübile on kirjeldatud käesolevas peatükis, arvestades eelmainitud tehnoloogiapinu ja küberharjutusplatvormi tausta. Nõuded on jagatud kaheks osaks. Funktsionaalsed nõuded määratlevad, mida süsteem peab tegema kasutaja jaoks ja mittefunktsionaalsed nõuded keskenduvad süsteemi toimimisele ja kvaliteedile.

Prototüübi koostamisel ja testimisel peab arvestama, et valitud automatiseerimistarkvara sobiks ettevõtte automatiseerimisplatvormiga vLM-Next, mis lubab lõimida automaatset süsteemide juurutamist pilveplatvormidel. Pilveplatvormiks on ettevõtte nõuete poolt määratud AWS.

2.5.1 Funktsionaalsed nõuded

- **Integratsioon kasutusel olevate teenustega**: Automatiseerimistarkvara peab lõimuma efektiivselt ettevõttes kasutusel olevate teenustega, nagu näiteks vLM-Next, tagades nendega ühilduvuse ja toetades protsesside automatiseerimist ning lihtsat liidestamist praeguse platvormi infrastruktuuriga.

- **Automaatne ja ühtne juurutamine**: Tööriistaga peab olema võimalik automaatselt juurutada ja seadistada CTF kursust, tagades kiire ja efektiivse paigaldusprotsessi.

- **Süsteemide terviklikkus**: Tööriist peaks suutma automaatselt seadistada süsteeme nendesse lisatarkvara paigaldamata. See tähendab, et süsteemid ei vaja näiteks agente, et juhtserveriga suhelda. Sellega tagatakse, et juurutatud süsteemid on esitletud sellisel kujul nagu need eksisteeriksid päriselus.

- **Juurutusprotsessi jälgimine**: Võimalust jälgida tööriista juurutusprotsessi reaalajas, sealhulgas katkestada protsessi või kasutada logisid tõrkeotsinguks.

2.5.2 Mittefunktsionaalsed nõuded

- **Mastabeeritavus**: Prototüüp peab võimaldama ressursside dünaamilist hallatavust ning kohandamist minimaalsete seadistusmuutustega, tagades ressursside optimeeritud kasutuse muutuvale nõudlusele.

- **Kasutatavus ja kohandatavus**: Prototüüp peab jälgima IaC (*Infrastructure as Code*) parimaid praktikaid, et seda saaks hõlpsasti kohandada erinevate süsteemide juurutamiseks.

- **Töökindlus**: Prototüüp peab olema efektiivne, lihtsasti uuendatav ja hooldatav, et tagada pikaajaline töökindlus.

3 Pilveteenuste platvorm

Pilveteenuste valdkonna hüppeline areng soodustab mitmekesist valikuvõimalust organisatsioonidele ja üksikisikutele. Sektoris käib tihe heitlus, millest annab aimu pidev teenuste loomine ning täiendamine. Pilveteenuste platvormide pakkujatena kerkivad esile tuntud nimed tehnoloogiasektoris, nagu näiteks Amazon, Microsoft, Google ja Oracle [7], kes kõik pakuvad IT-infrastruktuuride ehitamiseks olulisi teenuseid.

AWS on tuntud innovaatilise liidrina ja valdab suurt turuosa [7], samuti on CybExer Technologies kasutanud AWS'i peamise pilveteenuse platvormina. Loogilise lahendusena on prototüübi loomisel otsus kasutada AWS pilveteenuseid, sest ettevõtte teenused asuvad osaliselt sellel platvormil. Lisaks sellele on tehnilisel personalil kogemus AWSi haldamisel, mistõttu valmiva prototüübi kasutuselevõtuks ning kohandamiseks muudeks eesmärkideks ei vaja personal koolitamist ega õppimisaega.

3.1 AWS keskkond

AWS pakub laia infrastruktuuri teenuste valikut nagu privaatset virtuaalset pilve, virtuaalmasinate ja virtuaalvõrkude loomist [8]. Kursuse keskkond pilveteenuses peab jälgima sarnaseid põhimõtteid riistvaralise keskkonnaga, et keskkonna pesade eraldatud säiliks. Selleks tuleb arvestada küberharjutusplatvormi arhitektuuri ja kursuse juurutamise tehniliste nõutega.

Mitmed AWS pilveteenused võiksid potentsiaalselt sobida prototüübi loomiseks. Teenuste otsimisel on lähtutud CTF kirjeldusest. Pilveteenuste valiku saavutamiseks saab osaliselt järgida teadmisi, kuidas on varasemalt kursuseid juurutatud ja seadistatud. Järgmisena vaadeldakse võimalikke AWS teenuseid, mis aitavad täita eesmärki küberkaitse kursuste juurutamiseks pilveteenustesse.

3.1.1 Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2) pakub nõudluspõhist ja mastabeeritavat ressursside kasutust. EC2 abil on võimalik seadistada ja käivitada virtuaalseid servereid,

turvagruppe, lisada neid alamvõrkudesse ja hallata nende salvestusruumi [9]. EC2 lubab valida mitmesugust tüüpi *instance*'ite ehk virtuaalmasinate vahel, millele seadistada vastavalt vajadusele protsessorite arvu, mälu või salvestusruumi.

Virtuaalmasin luuakse, kas kasutaja kohandatud või Amazoni enda poolt pakutavate Amazon Machine Image'ite (AMI) põhjal. Need on mallid, kus on seadistatud operatsioonisüsteem ja tarkvara, mis määravad kasutaja töökeskkonna. Mallide valik on suur ning kasutajad saavad leida endale sobiva Amazon'i valikust või luua enda eelistuste kohaselt vastav mall.

3.1.2 Elastic Container Service

Elastic Container Service on täielikult hallatav konteineri orkestreerimisteenus, mis võimaldab juurutada, hallata ja mastabeerida konteineripõhiseid rakendusi. Teenus on lõimitud kolmanda osapoolte tööriistadega, nagu Elastic Container Registry ja Docker [10]. ECS võimaldab keskenduda mitmesugustele rakenduste ehitamisele, lihtsast veebilehest kuni mikroteenusteni välja, jättes vahelt ära keskkonna haldamise. Teenus käivitab konteinereid kasutades EC2'te või AWS Fargate'i.

Fargate on serverita lahendus, mille eest tasumine toimub vastavalt kasutusele [10]. Fargate hoolitseb virtuaalmasinate klastrite seadistamise, haldamise ja mastabeerimise eest, et jooksutada konteinereid. Valides EC2 teenuse ECS käivitustüübiks, tuleb valida virtuaalmasinate arv, nende tüüp ja salvestusruum, võimaldades kasutajale suuremat kontrolli infrastruktuuri haldamiseks.

3.1.3 Virtual Private Cloud

Amazon VPC võimaldab kasutajatel käivitada AWS ressursse isoleeritud võrgus. Võrku määratletakse konfiguratsioonidega, seadistades IP-aadresside vahemikke, marsruutimistabeleid, alamvõrke või internetilüüsi, mis lubab luua terviklikus võrgus eraldiseisvaid võrgusegmente [11].

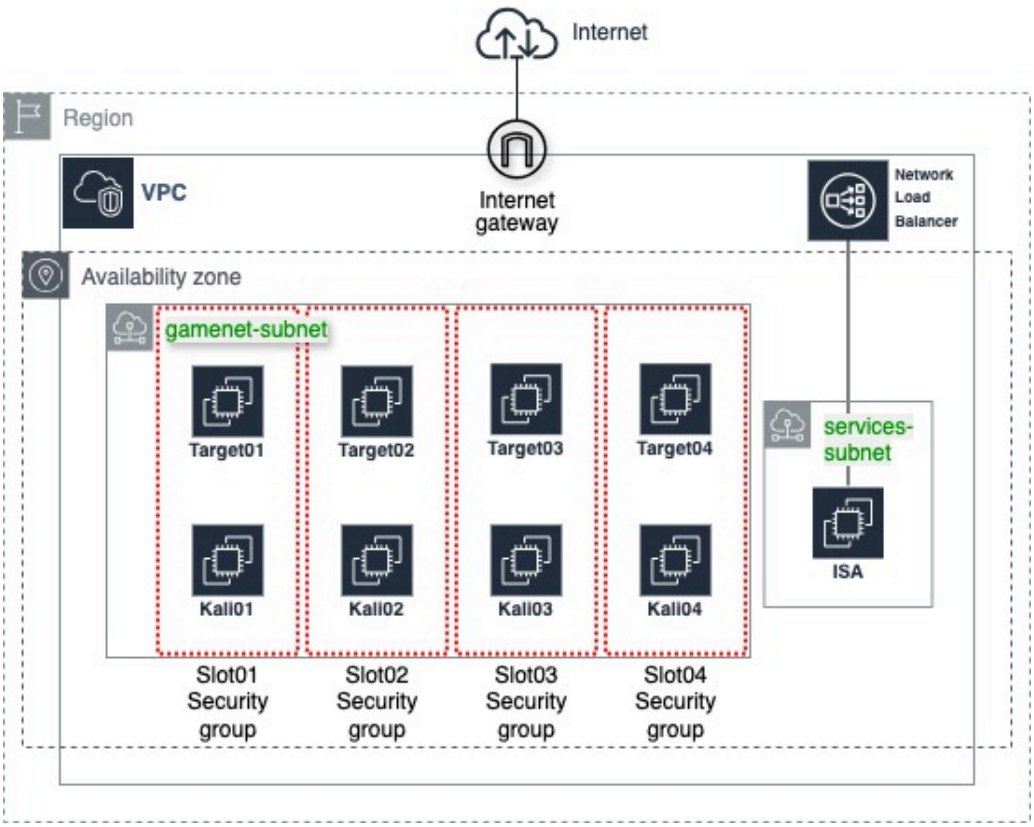
3.1.4 Järeldused

VPC on üks AWS teenustest, mida läheb kindlasti privaatse virtuaalse keskkonna loomiseks vaja ning alternatiivi sellele teenusele töö raames pole.

Põhjused, miks valida EC2 teenus, on seotud nii nõuete, kui ka kulude optimeerimisega. Kuna konteinerid pole mõeldud hoiustama graafilise liidesega virtuaalmasinaid jääb ECS Kali süsteemi valikust välja. Sihtmärk masinate ülesanded jooksevad küll konteinerites, kuid ECSi poleks otstarbekas kasutada. Kali ja sihtmärk masinad võiksid olla ühe teenuse peal, kuna see hoiab keskkonna ülesehituse järjepidevust. See lihtsustab süsteemide haldamist, kui mõlemad on virtuaalmasinad ning need on koodis sarnaselt kirjeldatud. ECS Fargate pakub serverivaba mugavust, kuid on AWS hinnakalkulaatori järgi umbes kolm korda kallim, kui EC2. Võrreldes omavahel EC2 ja ECS teenuseid ning kuidas juurutada CTF kursusele omaseid Kali ja sihtmärk masinaid, saab valituks EC2.

3.2 AWS keskkonna kavand

Küberkaitse kursuste juurutamiseks pilveteenuste keskkonda tuleb saada üldpilt komponentidest, mis annavad aimu, kuidas väljavalitud teenused on omavahel ühendatud. VPC ja EC2 on eelkirjeldatud teenustest kasutusele võetud kavandi loomisel. Loodud kavand (Joonis 4) visualiseerib AWS pilveteenuste keskkonda ning teenuste kasutust CTF kursuse raames.



Joonis 4. Pilveteenuste keskkonna kavand.

VPC on oluline lüli, mis annab võimaluse turvalisel viisil seadistada virtuaalset keskkonda AWS pilves, kus juurutada ja hallata rakendusi või süsteeme. Väline punktiirjoon kirjeldab regiooni maailmas, kus VPC asub, mis annab võimaluse majutada infrastruktuuri kliendile sobivas asukohas.

AWS regioonidele omaselt on neil mitu *availiability zone*'i või käideldavustsooni. Iga tsoon on isoleeritud, kuid omavahel ühendatud, juhul kui mõni tsoon lakkaks töötamast. Kavand on töös koostatud ühe tsooniga, sest kõrgkäideldavus pole osalejate süsteemide puhul esmatähtis. Privaatne pilv on ühenduses internetiga internetilüüsi abil.

Pilves on kujutatud tavapärased *access* ja *target* alamvõrgud ühena ning sellest on loodud sisuvõrgu alamvõrk. Niisiis eksisteerib kaks alamvõrku, *gamenet-subnet* – sisuvõrgu süsteemid, milleks on osalejate süsteemid Kali ning sihtmärk masinate näol, kust tõmmatakse või leitakse vajalikud andmed ülesannete lahendamiseks. Teiseks *services-subnet* – ISA platvorm kursusele ligipääsemiseks. Võrguliikluse koormusjaotur haldab ISA platvormile sissetulevat võrguliiklust, sest platvormi kasutavad kõik osalejad ülesannete nägemiseks, vastuste sisestamiseks ja tulemuste jälgimiseks.

Osalejate süsteemid eraldatakse turvagruppidega, et tekitada igaühele oma keskkond pesade näol. Grupid piiravad ligipääsu teiste kasutajate keskkondadesse, nii et igaühel on juurdepääs vaid ettenähtud Kali ja sihtmärk masinale.

Kavandis kirjeldatud AWS teenustega on võimalik juurutada ettevõttes loodud sisuvõrgu süsteeme CTF küberkaitse kursuse näol. Peatükis visualiseeritud kavandi põhjal toimub prototüübi loomine väljavalitud sobivaima automatiseerimistööriistaga.

4 Automatiseerimistarkvara analüüs

Infrastruktuur kui kood tööriistad pakuvad võimalust automatiseerida infrastruktuuri juurutamist ja haldamist, vähendades samal ajal manuaalseid protsesse ja vigade tekkimist. Gartneri 2021 aasta küsitluses 80% vastanutest, kes olid infrastruktuuri operatsioonide juhid, väitsid, et automatiseerimine on parim taktika kulude optimeerimiseks [12].

CybExer Technologies juurutab ja haldab iga kuu mitmeid erinevaid infrastruktuure mitmesuguste eesmärkide nimel. Automatiseerimine on sellisel juhul hädavajalik. Tänu automatiseerimisele suudetakse kokku hoida oluliselt kulusid aja või töötajate arvu pealt. Samal ajal võimendades projektide arvu, mida suudetakse pakkuda. Automatiseerimistööriist ei aita juurutada ainult kahe süsteemiga kursuseid, vaid ehitada sisuvõrke mitmete virtuaalmasinate segmentidega. Sisuvõrgu põhjal on võimalik korraldada sadade osalejatega üritusi, harjutusi või kursuseid. Seetõttu mängib automatiseerimistarkvara valik olulist rolli, millega pilveteenustes süsteeme juurutada ja neid hallata.

Tööriista valimine toimub kirjalike materjalide põhjal, võrreldes infrastruktuur kui kood tarkvarasid. Sellele järgnes tööriistade testimine AWS keskkonnas, et välja selgitada, kuidas tarkvarad päriselt toimivad, mille põhjal tehti reaalseid järeldusi tööriista kindlaks määramisel.

4.1 Tarkvarade üldine kirjeldus

Infrastruktuur kui kood tarkvarade valik on suur ning erinev paljude tegurite poolest. Nende hulgas on tööriistad nagu Ansible, AWS Cloudformation, Chef, Puppet, Terraform, Saltstack jne [13]. Automatiseerimistööriista valikul lähtutakse püstitatud nõuetest ning uuritakse tööriistade omadusi.

Kategooriliselt jaotuvad eelmainitud IaC tarkvarad konfiguratsioonihaldus või orkestraator tüüpideks vastavalt nende eesmärkidele, kas tööriist on mõeldud

infrastruktuuri haldama ja juurutama või süsteeme seadistama. Tegelikkuses ei kuulu tihtipeale aga need tarkvarad ühte kindlasse kategooriasse ja võivad hõlmata mõlemat tüüpi funktsioone.

Lisaks sellele on tööriistad ülesehituselt erinevad ning varieeruvad agentprogrammidega või agentideta tarkvaradeks. Ehk kas automatiseerimistööriistad peavad planeeritud süsteemide seadistamiseks paigaldama lisatarkvara, ehk agente, või teevad nad seda näiteks läbi *Secure Shell*'i (SSH).

Kirjelduskeeled, mida tööriistad pakuvad on kas imperatiivsed – kirjeldavad samm sammu haaval, kuidas saavutada lõpliku olekut, või deklaratiivsed – kirjeldavad lõpliku olekut, kuid IaC tööriist vastutab, kuidas seda olekut saavutada. Tarkvarad on tasuta kasutamiseks ning kõik peale Cloudformation'i on avatud lähtekoodiga. Tarkvaradel on ka tasulised versioonid, kuid antud töö raames neid vaja ei lähe.

4.1.1 Võrdlus

Võrdlusse on võetud automatiseerimistööriistad, mis on Google otsingumootori trendide seas kõige populaarsemad (vt Lisa 2), ning mis on välja toodud Gartner kogukonna küsitluses [14]. Võrdluse eesmärgiks on välja valida sobivad tööriistad, mis võiksid sobida prototüübi loomiseks [15].

Tabel 1. Automatiseerimistööriistade võrdlus.

Tööriist	**Tüüp**	**Kirjelduskeel**	**Lisatarkvara**	**AWS tugi**
Ansible	Konf. haldus	Imperatiivne	Ei	Jah
Cloudformation	Orkestraator	Deklaratiivne	Ei	Ainult AWS
Chef	Konf. haldus	Imperatiivne	Jah	Jah
Puppet	Konf. haldus	Deklaratiivne	Jah	Jah
Terraform	Orkestraator	Deklaratiivne	Ei	Jah

Tööriistadena on kolm konfiguratsioonihalduse ja kaks orkestraatori tüüpi tarkvara. Konfiguratsioonihaldustööriistade eelis peitub süsteemide või teenuste kirjeldamises

üksikasjalikult, et hooldada soovitud olekut ajas ning viia läbi seadistusmuudatusi üle mitme keskkonna. Orkestreerimise pluss on erinevas suuruses infrastruktuuri, töövoogude koordineerimise ja teostamise protsess ressursside vahel [16]. Orkestreerimistööriistade puhul on nende miinuseks süsteemide astmeline uuendamine. Kui süsteem tundub töötavat terviklikuna ning alusseadistus masina puhul ei muutu, siis orkestraator ei tuvasta, et masina peal olev muudatus on ootel ning ei uuenda midagi.

Ansible ja Chef omavad imperatiivset kirjelduskeelt, mis annab suurema kontrolli süsteemide loomisel ning juurutamisel, kui deklaratiivne keel. Samuti soodustab selline viis kirjutatud koodi terviklikku nägemist, mis annab kasutajale detailsema ülevaate, kuidas lõplik olek saavutatakse. Teisest küljest, mida pikem on kood, seda keerulisemaks selle lugemine võib osutuda ning vigade tekkimise oht tõusta. Deklaratiivse koodi kirjutamine on tihtipeale lihtsam ning muudab koodi loetavuse kergemaks. Lisaks sellele peegeldab kirjutatud koodi olek ka juurutatud infrastruktuuri viimatist olekut ning see muudab koodi hooldatavuse ajas lihtsamaks [15]. CTF kursuse süsteemide seadistamiseks peab siiski kaaluma ka detailsemat imperatiivset keelekasutust, sest tihtipeale on süsteemi juurutamisel oluline ka põhjalik tarkvara, mis masinasse paigaldatakse.

Lisatarkvara vajavad kaks tööriista. Chefi ja Puppeti puhul on vaja mõlemal tarkvaral juhtserverit ning hallatavatesse masinatesse agente, mis võtavad regulaarselt juhtserveriga ühendust, et seadistusmuudatusi avastada ja rakendada. Ülejäänud tarkvarad midagi lisaks paigaldama ei pea ning seetõttu Chef ja Puppet ei vasta süsteemi terviklikkuse nõudele.

Kõik tööriistad toetavad AWS pilveteenuseid omal moel, kas teekide või moodulite abil, kasutades AWS rakendusliidest või tarkvaraarenduskomplekti. Jättes välja nõuded lõputööle näitab see, et tegelikult kõik välja toodud tööriistad on võimelised AWS teenustega suhtlema.

Jättes võrdlusest välja lisatarkvara vajavad infrastruktuur kui kood tööriistad jäävad valikusse Ansible, AWS Cloudformation ning Terraform. Vaadates tööriistade populaarsust Google otsingumootoris (vt Lisa 2), võib väita, et kahe aasta jooksul on Terraform ja Ansible teiste seas kõige tuntumad. Kolmandale kohale on tõusnud

Cloudformation. Kuna kirjaliku analüüsi põhjal ei saa kindlalt väita, milline tarkvara on sobivaim, tuleb need ükshaaval üle vaadata ning viia läbi mõned tehnilised testid.

4.2 Ansible

Ansible on avatud lähtekoodiga automatiseerimise tarkvara, mis automatiseerib ressursside ettevalmistamist, konfigurationsioonihaldust, rakenduste juurutamist, orkestreerimist ja paljusid muid protsesse [17]. Tarkvara on loodud aastal 2012 tasuta kasutamiseks, 2015. aastal omandati see Red Hat'i poolt. Avatud lähtekoodi tõttu on tarkvara parendamisse kaasatud toote enda kogukond.

Automatiseerimiseks kasutab Ansible mooduleid. Need on väiksed programmid, mis käivitavad kasutaja kirjeldatud ülesandeid. Moodulid käivitatakse tarkvara poolt üle SSH ning eemaldatakse, kui töö on tehtud. Kavandatud on need idempotentsetena ning muudatusi tehakse süsteemides vaid siis, kui see on vajalik [18].

Ansible kasutab imperatiivset lähenemist keelega YAML (*Yet another markup language*), millega kirjeldatakse olekut, mida tarkvara rakendab. Ansible *Playbook* koosneb ühest või rohkemast *Play*'st, *Play* omakorda sisaldab ühe rakenduse või tarkvara koodi (rolli), ja igat tegevust koodis nimetatakse *task*'iks ehk tegevuseks [19].

Automatiseerimistarkvara väljastab käsureal reaalajas väljundeid täitmisel olevatest tegevustest, ning seda on võimalik lihtsasti katkestada näiteks klahvidega CTRL+C. See jätab juurutamisprotsessi katki kohast, kust katkestamissignaal anti.

4.3 Terraform

HashiCorp Terraform on 2014. aastal loodud infrastruktuuri kui kood tööriist, mis võimaldab määratleda ressursse inimloetavates konfiguratsioonifailides ja pakub versioonihaldust, taaskasutatavust ning jagatavust. Järjepidev töövoog tagab infrastruktuuri juurutamise ja haldamise kogu selle elutsükli jooksul [20].

Terraform töötab kahe sisendallika põhjal. Esimeseks on kasutaja kirjeldatud sisend, milliseid ressursse luuakse ja teine sisend koosneb andmetest, mis annavad teavet hetke infrastruktuuri seadistusest. Tarkvara kasutab seda sisendit, võrdleb olemasoleva infrastruktuuri olekuga ja seadistab oleku viisil, mis kõrvaldab erinevused [21].

Deklaratiivne lähenemine on võimalik *Hashicorp Configuration Language* (HCL) abil, millega soovitud infrastruktuuri olekut kirjeldada. Terraform suhtleb infrastruktuuri majutajate või pilveteenuste pakkujatega läbi pistikprogrammide. Peamisteks komponentideks on *providers* – pakkujad ja *provisioners* – seadistajad. Pakkujad suhtlevad pilveteenuse pakkujatega ning loovad, haldavad ja kustutavad ressursse [21]. Loodud ressursside konfigureerimiseks kasutatakse seadistajaid viimase võimalusena toimingute sooritamiseks.

Terraform pakub enne ressursside loomist planeerimisvaadet, mis käivatatakse *plan* käsuga, see kuvab detailsed andmed loodavatest ressurssidest. Samuti kuvatakse oleku juurutamise ajal käsureal väljundit toimuva kohta, kuid pole nii detailne, kui Ansible väljund. Katkestamissignaali saab saata samuti sarnaselt Ansible'le.

4.4 Cloudformation

AWS Cloudformation on 2011. aastal loodud teenus, mis kasutab mallifaile AWS ressursside seadistamise automatiseerimiseks. Vahendit võib kirjeldada kui infrastruktuuri automatiseerimise või IaC tööriistana, kuna see võimaldab mitmesuguste AWS teenuste seadistamist ja juurutamist [22].

Teenus võimaldab ka käsurea põhist lähenemist, mis lubab luua skripte automaatseks ressursside haldamiseks ning seadistamiseks Cloudformation'iga. Sellist lähenemist saab teostada AWS CLI kaudu, mis pakub laiemat käsurea haldamise võimalust paljudele AWS teenustele.

Cloudformation mallid on deklaratiivsed ja kasutavad JSON või YAML formaati, mis salvestatakse lokaalsesse masinasse, AWS S3 salvestusteenusesse või Git koodihoidlasse. Seejärel luuakse Cloudformation'iga *Stack*, mis on ühe mallifaili põhjal põhinev ressursside kogum, mida tarkvara juurutab.

4.5 Testimine

Tehnilised testid teostati kolme tööriistaga, mis teoreetiliste andmete põhjal valikutest esile tõusid – Ansible, AWS Cloudformation ja Terraform. Testide loomiseks võeti aluseks AWS keskkonna peatükis loodud kavand (vt Joonis 4, lk 21). Kavandi põhjal on tähtsamad tegevused VPC ja virtuaalmasinate loomine, mis võiksid peegeldada

automatiseerimistööriistade kasutust kursuse juurutamisel. Lisaks sellele, annab testimine aimu tööriistade kasutusmugavusest, lihtsusest ning kasutaja õppimiskõverast. Tehniliste testide eesmärk on välja selgitada parim tööriist prototüübi loomiseks ja vastamine nõuetele.

4.5.1 Testimise keskkond ja kirjeldus

Testimine viiakse läbi lokaalses arvutis, seetõttu sõltub Ansible ja Terraformi juurutamise kiirus ka süsteemis saadaolevatest ressurssidest. Näiteks protsessori jõudlusest ja mälu mahust. Kuna suur osa töötlemisest tehakse Cloudformationi puhul AWSi peal, siis lokaalsed ressursid erilist rolli ei mängi. Sellegipoolest võib kõiki tööriistu mõjutada internetikiirus ning latentsus.

Tööriistade testimisel on lähtutud nende dokumentatsioonidest ning üritatud kasutada nende parimaid praktikaid. Samuti on tegevused teostatud võimalikult sarnaselt, et saavutada samasugune lõplik olek. Olekute saavutamiseks on näidisandmed järgmised: AWS Stockholmi regioon, VPC loomine, millega kaasnevad internetilüüs, alamvõrk 10.80.64.0/18, marsruutimistabeli IPv4 ja IPv6 liikluse suunamine ja turvagrupp.

Testimise käigus mõõdeti ka juurutamiskiirust mitmel katsel ning selle põhjal arvutati keskmine aeg. Esimene test on VPC loomine eelkirjeldatud näidisandmetega. Teine test on EC2 Ubuntu virtuaalmasina juurutamine ja sidumine esimeses testis loodud VPC alamvõrgu ja turvagrupiga. Kolmanda testiga on katsetatud virtuaalmasina seadistamist Dockeri Nginx konteineriga.

4.5.2 Testide järeldused ja nõuetele vastamine

Terraform ja Cloudformation on orkestreerimise tüüpi tarkvarad ja neil on eelis infrastruktuuride juurutamises. Ansible on konfiguratsioonide haldamiseks parim ning läbis kolmanda testi kõige edukamalt. Tegelikkuses on kõikide tarkvarade piirid hägused ja kõigiga oli võimalik läbida kolm testi vaatamata nende erinevustele.

Ansible tarkvaraga on võimalik peale seadistamise luua ka infrastruktuuri juurutamise koodi, mis vastab parimatele praktikatele. Tarkvara kasutamisel peab kirjutama vastavalt ressursside juurutamise koodile ka ressursside hävitamise koodi. Seda on võimalik teha sildistamise abil, mille järgi ressursse unikaalselt nimetada [23]. Terraformi eelis on oleku salvestamine ning võrdlemine kasutaja sisendiga, mis seetõttu ei vaja lisa koodi

kirjutamist, et ressursse hävitada. AWS Cloudformation hoiab ressursside juurutamist pinudena, mis näitab tulemusi ja sündmusi sellest protsessist. Pinu malli põhjal toimub ka kõigi nende ressursside kustutamine, mis on juurutamise mallis kirjeldatud.

Terraform ja Cloudformation juurutavad infrastruktuure kergemini, kui Ansible, aga virtuaalmasinate seadistamises jäävad need suuresti siiski viimasele alla. Kuigi Ansible võib olla robustne ja vajada üksikasjalike kirjeldusi, kuidas teatud ressursside olekut saavutada, on see tööriist siiski ainsana võrdlemisi efektiivne juurutama infrastruktuuri ja seadistama virtuaalmasina tarkvara. Erinevus on sedavõrd suur, et kasutatavuse ja kohandatavuse nõudele vastab Ansible kõige rohkem.

Kombinatsioon Terraform'ist ja Ansible'st või Clouformation'ist ja Ansible'st on võimalus, mis võiks ka teoreetiliselt sobida, kuid läheks vastuollu automaatse ja ühtse juurutamise nõudega. Potentsiaalselt on võimalik luua baasmallid virtuaalmasinatest Terraform'i või Cloudformation'iga ja juurutada need kursuse raames pesadesse, kuid detailset seadistamist nendega teha ei saa. Ühe tööriista kasutamine annab eelise, et kood on unifitseeritud ja sellel on ühtne lähenemine. Seda enam, et infrastruktuuri osa on väiksem, kui sisuvõrgu juurutamine, muudab see protsessi lihtsamaks, kasutades ühte infrastruktuur kui kood tarkvara.

Testide keskmiste aegade mõõtmistest selgus, et tegelikult on kõik tööriistad suhteliselt võrdsed ja erinesid vaid mõne sekundi jagu. Eelduste kohaselt oli esimeses kahes testis Cloudformation kõige kiirem, mis osutus tõeseks, kuna tegemist on AWSi enda tööriistaga. Ansible oli kõige kiirem kolmandas testis, mis oligi pigem sellele suunatud. Selgus siiski, et Ansible ei jäänud teistest tööriistadest infrastruktuuri juurutamisel maha ja oli isegi Terraformist kiirem esimese testi puhul. Sellegipoolest ei ole aegade mõõtmises kindlat eelist ühelgi tööriistal, sest erinevused olid liiga väikesed.

Vaadeldes testimise käigus selgunud järeldusi, vastab Ansible tööriist nõuetele märksa rohkem, kui teised. Arvestades kasutusel olevate teenuste integratsiooni nõuet on otsuse tegemine Ansible kasuks veelgi soodsam. Kuna tarkvara on juba ettevõttes kasutusel ning võrdluses olnud tööriistad Ansible ees märgatavat eelist analüüsi tulemusena ei paku, langeb automatiseerimistarkvara valik pilveteenustesse juurutamisel Ansible kasuks.

5 Prototüüp

Prototüüp rakendatakse Ansible automatiseerimistarkvaraga, mis vastas analüüsi käigus töös kirjeldatud nõuetele. Prototüübi arendamisel on oluline asjaolu, et vLM-Next kasutab DevOps osakonna loodud Ansible *inventory* pistikprogrammi ja sellel on võimekus genereerida Ansible jaoks sobivat ressursside loendit.

Kontseptsiooni loomisel on kasutatud AWS kavandit (Joonis 4, lk 21), mis kirjeldab keskkonda, kuhu on võimalik juurutada töös kirjeldatud CTF küberkaitse kursust. Selle alla ei kuulu ISA rakenduse juurutamine AWSi ega harjutusmooduli loomine ISA veebirakenduses. Prototüübi valmimiseks valitakse kaks sisuvõrgu pesa, kuhu paigaldada CTF kursus. Sellest piisab, et tõestada kontseptsiooni toimimist ning potentsiaalset dünaamilisust.

Prototüübi loomisel on infrastruktuuri ja sisuvõrgu osa jagatud kaheks. Põhjus seisneb selles et, infrastruktuuri loomine on ettevõtte protsesside järgi administraatorite tegevus ja jääb DevOps osakonna ülesandeks. Sisuvõrgu juurutamine infrastruktuuri keskkonda käib omakorda vLM-Next'iga, mille eesmärk ongi luua, arendada ja hallata sisuvõrke.

Kontseptsiooni tõendamiseks kasutatakse AWS keskkonna infrastruktuuri ning vLM-Nexti veebirakenduse genereeritud Ansible *inventory*'it. Viimase funktsionaalsuse toimimiseks on vLM-Next'il API pääse, mis genereeritakse sisuvõrgu juurutusprotsessis. Juurutusprotsessi loomiseks on aga vaja defineerida nii infrastruktuuri keskkond, kui ka sisuvõrk. See võimaldab teha kõik juurutusprotsessi tegevused lokaalses masinas. Selline viis on mugav Ansible koodi katsetamiseks ja testimiseks, kasutades ettevõtte loodud platvormi uut versiooni.

5.1 Ansible kasutamine

Ansible paigaldamisel lokaalsesse arvutisse on kasutatud isoleeritud virtuaalset Pythoni keskkonda *virtualenv* [24], et alustada projekti loomisel tühja keskkonnaga. Lisaks sellele on kasutatud *virtualenvwrapper*'it [25], mis laiendab *virtualenv* virtuaalselt keskkonda võimekusega luua ja kustutada mitmeid keskkondi. Paigaldatud on Ansible 2.16.6 ja

Python 3.10.13 versioon. Samuti sõltub Ansible *requirmenets.txt/yml* failis olevatest teekidest, mis võimaldavad suhelda erinevate moodulite ja AWS rakendusliidesega.

Group_vars kaustas on defineeritud põhilised muutujad, mida läheb projekti käigus vaja. Näiteks *aws_region* – kasutatav regioon, *aws_account_id* – AWS kasutaja id, vLM-Next, Vault, IAM muutujad jne. Prototüübi loomisel on seatud projekti eesliiteks *a12*, mis on defineeritud muutujaga *aws_prefix*. Selle järgi sildistatakse teenused, mida projekti raames kasutatakse.

Juurkaustas olevad tegevused infrastruktuuri ja sisu juurutamiseks on kirjeldatud *roles* all, iga roll täidab oma ülesannet. Kokku on üheksa rolli, millest viis on seotud virtuaalmasinate juurutamise ja sisu paigaldamisega. Kaustas *plugins* eksisteerib ka pistikprogramm *vlm_inventory*, mis on koodihoidlast välja jäetud. vLM-Next ühenduse infot käsitleb *inventory.yml* fail.

5.2 Virtuaalse pilve infrastruktuur

Infrastruktuuri juurutusprotsessis on kasutatud Ansible *playbook*'e, mida käivitatakse lokaalses masinas käsurealt ning mille põhirõhk on AWS keskkonna ettevalmistamisel. Pilvekeskkonna ettevalmistamiseks on kirjutatud neli rolli, mille abil luua AWS infrastruktuur.

5.2.1 Rollide tegevused

Create_keys rolli nimi kirjeldab rolli sisu peaaegu ära – see loob ja salvestab RSA võtmepaari, kui neid juba AWS'is ei eksisteeri. Neid võtmeid kasutatakse hiljem Ansible poolt ühenduse loomiseks ja EC2 virtuaalmasinate seadistamiseks.

Setup_vpc rakendab virtuaalse privaatse pilve nimega *a12_vpc* Stockholmi regiooni, määrates sellele IP-aadresside ploki 10.80.0.0 võrgumaskiga /16. Teise tegevusena luuakse internetilüüs ning registreeritakse tulemus. Seejärel kasutatakse AWSi poolt määratud IPv6-aadresside plokki ja salvestatakse muutujana mällu. Samuti registreeritakse saadaolevad AWSi käideldavustsoonid muutujatesse. Tsükliga läbitakse alamvõrkude seadistamise tegevus, kus kasutatakse eelnevaid muutujaid ja registreeritakse ka see tulemus. Peale seda päritakse vaikimisi marsruutimistabel, et seadistada väljaminev võrguliiklus internetilüüsiga. Viimasena käesolevas rollis luuakse

teenuste ja sisuvõrgu alamvõrk, kus lubatakse erinevad protokollid nagu *Virtual Network Computing* (VNC), *Remote Desktop Protocol* (RDP) ja SSH. Võrgud on seotud, kuna osalejad ühenduvad ISA's asuvate konsoolivalikute kaudu sisuvõrgu süsteemidega.

Vlm_iam roll on seotud vLM-Next veebirakendusega ning loob selle jaoks piiratud õigustega kasutaja. Esimese tegevusena kopeeritakse reeglite kogumiku mall AWSi ning luuakse AWS *Identity and Access Management* (IAM) tavade kogumik. Reeglite kogumik kujutab endast laia ligipääsu EC2 ressurssidele. Seejärel luuakse selle kogumikuga vLM-Next teenuse jaoks kasutaja ja prinditakse ligipääsuks võtmed. Võtmed lisatakse veebirakendusega seotud paroolihaldurisse.

Ctf_sec_grp on sisuvõrgu pesade tegemiseks mõeldud roll, mis loob pesadele vastavad turvagrupid. Esialgu päritakse VPC info, mis registreeritakse muutujasse ning seejärel kasutatakse neid turvagruppide loomisel. Grupid luuakse *a12-s001-sg* vormingus vastavalt pesa numbrile ning lubavad ligipääsu ettevõtte IP-aadressilt, kaughaldus protokollidelt ning pesade-sisest võrguliiklust.

Selle protsessiga on juurutatud AWS infrastruktuur. VPC loomise tegevuste väljund on saadaval Lisas 3, IAM kasutaja loomine Lisas 4. Turvagruppide väljund Lisas 5.

5.3 vLM-Next'iga sisuvõrgu seadistamine

Sisuvõrgu defineerimiseks kasutatakse vLM-Next platvormi, et näidata sisu juurutamist platvormi genereeritud Ansible *inventory*'ga. Selline võimekus on loodud *vlm_inventory* skriptiga. Skript on loodud Pythonis ettevõtte DevOps osakonna inseneri poolt ning selle eesmärgiks on lugeda vLM-Nextis defineeritud ressursse. Eelduseks on vaja platvormile seadistada ja defineerida AWS platvorm, keskkond, süsteemide definitsioonid ja moodulid.

5.3.1 AWS Platvormi ja sisu defineerimine

Esimeseks sammuks vLM-Nextis tuleb defineerida platvorm, mida hakatakse kasutama infrastruktuuri keskkonnana sisu juurutamiseks. Selleks on prototüübi puhul AWS. Teiseks tuleb seadistada keskkond, mis piiritleb mitmesse pesasse on võimalik sisu juurutada, keskkonna alamvõrgud ning muutujad nagu *aws_prefix* ja pesade nimede vormingu.

Seejärel luuakse moodul, kuhu sisestada mooduli nimi. vLM-Next loob samanimelise Git koodihoidla ja harjutuse tüübi. Moodulisse lisatakse veel õiged alamvõrgud ning muutujad, mis viitavad Vault'is asuvatele privaatsetele andmetele ja AWS väärtustele. Lisaks sellele lisatakse süsteemide definitsioonid, mis tuleb aga enne ära teha, kui mooduli seadistamise saab lõpuni viia. Süsteemi defineerimisel kirjeldatakse vajalikud andmed. AWS AMI puhul on selleks vabalt valitud süsteemi nimi, mis loob ka eraldi koodihoidla, virtuaalmasina riistvaralise seadistuse, ligipääsetava kasutaja ja rakendusliidese valiku.

Süsteemi defineerimisel luuakse sellest koodihoidlasse Ansible'le sobiva struktuuriga kataloogid koos defineeritud andmetega. vLM-Next veebirakendusel on sisse ehitatud tekstiredaktor ja funktsionaalsus avada, luua, kustutada ning muuta faile ja nende sisu. Samuti neid Gitea'sse üles laadida, tõmmata ja avaldada. Definitsioone saab seejärel seadistada moodulitesse, kus saab need lisada kirjeldatud alamvõrkudesse, muuta virtuaalmasinale jaotatud ressursse või omakorda korrigeerida definitsiooni moodulispetsiifilisteks.

Viimase asjana viiakse läbi sisu juurutamine ja luuakse projekti juurutamise protsess. Selle loomisel valitakse õige moodul, versioon ja keskkond, kuhu moodul juurutatakse. Valikute tegemisel otsustatakse mitut keskkonna pesa kasutaja soovib kasutada ning milline on esimene pesa, kuhu sisu juurutatakse. Protsessi all saab veelkord üle vaadata kõik ressursid ning vLM-Next'i loodud Ansible *inventory*. Juurutamisprotsessil on tegelikult kolm režiimi, mida käivitada – *deploy*, *redeploy* ja *undeploy*, kuid see viiakse läbi lokaalses masinas.

5.3.2 vLM-Next *inventory* ja rollid

vLM-Nextis kõige eelneva seadistamise tulemusena väljastatakse juurutusprotsessi algatades vLM-Next *inventory*, et juurutada seal defineeritud süsteeme. Pärida on seda võimalik ka lokaalses masinas *ansible-inventory --list* käsuga (vt Lisa 6), kui *inventory.yml* on loodud õigete parameetritega. Siinkohal kasutatakse *main.yml* failis väljatoodud Ansible rolle.

Configure_vars on loodud, et pärida informatsiooni AWS konto, kui ka SSH kohta, mis on kirjeldatud Vault'is. SSH võti salvestatakse ajutiselt lokaalsesse arvutisse, et luua ühendus virtuaalmasinate juurutamiseks.

Deploy roll koosneb mitmest failist ja nendega juhitakse tegevusi olenevalt lisamuutujatest, mida kasutatakse käsureal *playbooki* jooksutamisel. Lisamuutujate kasutamine käsureal on mõeldud juurutusprotsessi lihtsustamiseks. Neid on võimalik kasutada *-e=deploy_mode=undeploy/redeploy* variantidega, mis kustutavad või juurutavad uuesti ressursse.

Connection roll sarnaselt nimele, seab ette vajalikud parameetrid ja ootab, et luua ühendus masinaga. Eelkirjeldatud rollide olemasolul on võimalik nendega juurutada CTF küberkaitse kursust.

5.3.3 CTF kursuse juurutamine

CTF virtuaalmasinate loomisel on kasutatud *a12-ami-target* ja *a12-ami-kali* definitsioone, et luua kohandatud AMI'id ehk tõmmised AWS'is. Vastasel korral võtab süsteemide juurutamine pesadesse liiga palju aega, sest virtuaalmasinad on tühjad ning kohandamata. AMI'de loomise väljundid leiab Lisast 7 ja 8.

AMI'de identifikaatoreid saab kasutada uute süsteemi definitsioonide loomisel. Uued definitsioonid lisatakse samamoodi vLM-Next moodulisse, kuid seadistatakse need sisuvõrgu alamvõrku. Seejärel kasutatakse jällegi käsurida, et juurutusprotsess lõpuni viia ja küberkaitse kursuse süsteemid pesadesse juurutada (vt Lisa 9).

Kursuse süsteeme on võimalik juurutada dünaamiliselt kasutades regulaaravaldisi, mille abil saab täpsustada käsureal mustritega soovituid masinaid *ansible-playbook -l '~.*01.s00[0-9].*' main.yml*. Sellise käsuga on võimalik juurutada *inventory* loendis saadaolevaid masinaid üheksasse pesasse.

5.4 Järeldused

Prototüübi loomisel oli eesmärgiks kasutada väljavalitud automatiseerimistööriista, millega juurutada küberkaitse kursuseid AWS pilveteenusesse. Valminud prototüüp täidab sõnastatud eesmärki ning vastab ka peatükis 2.5 kirjeldatud funktsionaalsetele nõuetele. Lahendus sobib kokku ettevõttes eksisteerivate platvormide ja tööriistadega, hoides järjepidevust. Tööriistaga on võimalik automaatselt juurutada CTF kursust paigaldamata süsteemidesse lisatarkvara, näidates protsessist reaalajas väljundit.

Loodud lahendus vastab ka mittefunktsionaalsetele nõuetele, mis lubab ressursside dünaamilist hallatavust, võimaldades valida spetsiifilisi süsteeme regulaaravaldiste mustritega. Kood jälgib infrastruktuur kui kood parimaid praktikaid [26], nagu näiteks rollide eraldatus, salajaste muutujate Vault'is hoidmine, tühimike ja plokkide kasutamine jne. Lisaks sellele on koodi lihtne uuendada arvestades rollide organiseeritust ja seda on kerge kasutada lisamuutujate tõttu käsureal. Prototüübis kasutatud kood on kättesaadav https://github.com/sapluk-ttu/Ansible-cloud-deploy lingilt.

5.4.1 Edasiarendused

Edasiarendusteks on prototüübi raames mõned valikud. Üheks võimaluseks on juurutada ISA platvorm Ansible *playbook*'idega samasse virtuaalsesse pilve. Seejärel on võimalik lisada käesolev CTF harjutusmoodul ISA veebirakendusse. Samuti juurutusprotsessi lõpuni viimine vLM-Nextis, lõimides prototüübi käigus loodud rolle ja nende koodi. Lisaks sellele saab Ansible'ga automatiseerida kohandatud virtuaalmasinast AMI loomise tegevust.

6 Kokkuvõte

Käesoleva töö eesmärgiks oli leida ettevõtte automatiseerimisplatvormile vLM-Next infrastruktuur kui kood tarkvara, mille abil juurutada küberkaitse kursuseid pilveteenuste platvormil. Analüüsi käigus selgus mitu tarkvara, mis võiksid selle eesmärgi täitmiseks sobida. Valiku tegemisel järgiti defineeritud funktsionaalseid ja mittefunktsionaalseid nõudeid ja võrreldi peamiselt kolme sobivaimat automatiseerimistarkvara, et luua toimiv prototüüp.

Pilveteenuste platvormina kasutati AWS'i, kus autor uuris seal pakutavaid teenuseid, mis sobiksid küberkaitse kursuse juurutamiseks. Selle tulemusena koostati kavand, milline võiks välja näha toimiv CTF kursuse lahendus pilves. Samuti analüüsis töö autor populaarseid ja soovitatud infrastruktuur kui kood tarkvarasid, mille tulemusel liiguti edasi Ansible, Cloudformation ja Terraform tööriistadega. Teoreetilise võrdluse põhjal kolme hulgast aga ei selgunud kindlat tööriista, millega edasi minna. Seetõttu teostati tehniline testimine, et välja selgitada nõuetele vastav tööriist prototüübi loomiseks, milleks osutus Ansible.

Töö tulemusena töötas autor välja Ansible rollid AWS infrastruktuuri keskkonna ja küberkaitse kursuse juurutamiseks pilves. Infrastruktuuri juurutamise viis autor läbi lokaalses masinas kasutades Ansible'd. Sisuvõrgu loomiseks kasutati vLM-Next platvormi, et defineerida olemasolevat infrastruktuuri keskkonda ja planeeritavaid kursuse süsteeme. Selle tulemi järel väljastas vLM-Next Ansible *inventory*, mida rakendati, et viia lõpuni küberkaitse kursuse juurutamine lokaalses arvutis. Prototüübi tulemusena täideti töö eesmärk ja saavutati sellele seatud.

Kasutatud kirjandus

[1] CybExer Technologies OÜ, “Majandusaasta aruanne.“ Accessed: 07 Märts 2024. [Online]. Available: https://ariregister.rik.ee/est/company/14013437/file/9010485052

[2] CybExer Technologies OÜ. Internal Documentation. Tallinn, 2024.

[3] R. Bastiaansen ja R. Lanigan, “VMware.“ Accessed: 07 Märts 2024. [Online]. Available: https://www.techtarget.com/searchvmware/definition/VMware

[4] “Capture the flag competition.“ Accessed: 09 Märts 2024. [Online]. Available: https://buildyourfuture.withgoogle.com/events/ctf

[5] K. Leune ja S. J. Petrilli, “Using Capture-the-Flag to Enhance the Effectiveness of Cybersecurity Education.“ Accessed: 09 Märts 2024. [Online]. Available: https://research.leune.org/assets/p47-leune.pdf

[6] “What is Kali Linux?“ Accessed: 20 April 2024. [Online]. Available: https://www.kali.org/docs/introduction/what-is-kali-linux/

[7] D. Wright, D. Smith, K. Ji, M. A. Borrega, A. Galimberti ja S. Bauman, “Magic Quadrant for Strategic Cloud Platform Services.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-2FTDYPQN&ct=231204&st=sb&trk=article-ssr-frontend-pulse_little-text-block

[8] A. Wittig ja M. Wittig, “Amazon Web Services in Action, Third Edition.“ Accessed: 10 Mai 2024. [Online]. Available: https://books.google.ee/books?id=joK3EAAAQBAJ&lpg=PA1&dq=aws%20virtual%20machine&lr&pg=PR5#v=onepage&q=aws%20virtual%20machine&f=false

[9] “Amazon EC2 User Guide.“ Accessed: 10 Mai 2024. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[10] “Amazon ECS Developer Guide.“ Accessed: 10 Mai 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html

[11] “Amazon VPC User Guide“ Accessed: 10 Mai 2024. [Online]. Available: docs.aws.amazon.com

[12] M. Rimol, “4 Predictions for I&O Leaders on the Path to Digital Infrastructure.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.gartner.com/en/articles/4-predictions-for-i-o-leaders-on-the-path-to-digital-infrastructure

[13] F. Pialoux, “Best Infrastructure as Code Tools (IaC).“ Accessed: 10 Mai 2024. [Online]. Available: https://bluelight.co/blog/best-infrastructure-as-code-tools

[14] “Infrastructure as Code Tool Usage Poll.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.gartner.com/peer-community/poll/infrastructure-code-iac-tool-organization-use

[15] Y. Brikman, “Why we use Terraform and not Chef, Puppet, Ansible, SaltStack, or CloudFormation.“ Accessed: 10 Mai 2024. [Online]. Available:

https://lsi.vc.ehu.eus/pablogn/docencia/AS/Act7%20Admin.%20centralizada/Terraform%20Chef%20Puppet%20Ansible%20Salt.pdf

[16] “How do you choose between configuration management and orchestration tools?“ Accessed: 10 Mai 2024. [Online]. Available: https://www.linkedin.com/advice/3/how-do-you-choose-between-configuration-management

[17] “Ansible.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.ansible.com/

[18] “How Ansible Works.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.ansible.com/how-ansible-works/

[19] “What is an Ansible module—and how does it work?“ Accessed: 10 Mai 2024. [Online]. Available: https://www.redhat.com/en/topics/automation/what-is-an-ansible-module

[20] “What is Terraform?“ Accessed: 10 Mai 2024. [Online]. Available: https://developer.hashicorp.com/terraform/intro

[21] D. Harrington, “What is Terraform: Everything You Need to Know.“ Accessed: 10 Mai 2024. [Online]. Available: https://www.varonis.com/blog/what-is-terraform

[22] “What Is AWS CloudFormation?“ Accessed: 10 Mai 2024.. [Online]. Available: https://www.contino.io/insights/aws-cloudformation

[23] “A simple approach to delete AWS resources with Ansible.“ Accessed: 10 Mai 2024. [Online]. Available: https://nleiva.medium.com/a-simple-approach-to-delete-aws-resources-with-ansible-b31c796fa16d

[24] “virtualenv.“ Accessed: 10 Mai 2024. [Online]. Available: https://pypi.org/project/virtualenv/

[25] “virtualenvwrapper.“ Accessed: 10 Mai 2024. [Online]. Available: https://virtualenvwrapper.readthedocs.io/en/latest/

[26] “Best Practices.“ Accessed: 10 Mai 2024. [Online]. Available: https://docs.ansible.com/ansible/2.8/user_guide/playbooks_best_practices.html

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

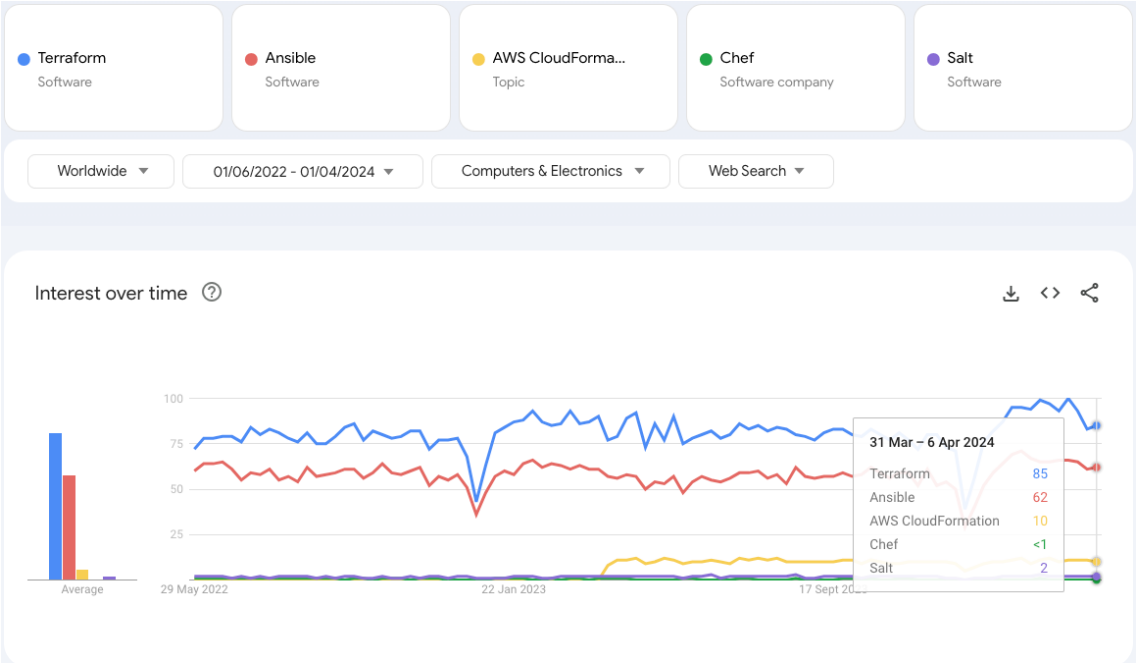
Mina, Sander Plukš

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Küberkaitse kursuste automaatne juurutamine pilveteenuste platvormil", mille juhendaja on Siim Vene
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

13.05.2024

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Google Trends graafik



Lisa 3 – Ansible väljund VPC tegevustest

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
playbook vpc.yml -t vpc

PLAY [localhost] ***************************************************************

TASK [setup_vpc : Create a new VPC] ********************************************
Wednesday 08 May 2024  17:50:16 +0300 (0:00:00.023)       0:00:00.023 ***
changed: [localhost]

TASK [setup_vpc : Create Internet Gateway for VPC] *****************************
Wednesday 08 May 2024  17:50:19 +0300 (0:00:03.090)       0:00:03.114 ***
changed: [localhost]

TASK [setup_vpc : Define allocated ipv6 block] *********************************
Wednesday 08 May 2024  17:50:23 +0300 (0:00:03.265)       0:00:06.380 ***
ok: [localhost]

TASK [setup_vpc : Get available AZs] ********************************************
Wednesday 08 May 2024  17:50:23 +0300 (0:00:00.087)       0:00:06.467 ***
ok: [localhost]

TASK [setup_vpc : Create subnets] ***********************************************
Wednesday 08 May 2024  17:50:26 +0300 (0:00:03.196)       0:00:09.663 ***
changed: [localhost] => (item)
changed: [localhost] => (item)

TASK [setup_vpc : Get default route table] **************************************
Wednesday 08 May 2024  17:50:36 +0300 (0:00:10.209)       0:00:19.872 ***
ok: [localhost]

TASK [setup_vpc : Configure VPC default route table] ****************************
Wednesday 08 May 2024  17:50:39 +0300 (0:00:03.308)       0:00:23.180 ***
changed: [localhost]

TASK [setup_vpc : Create default services security group] ***********************
Wednesday 08 May 2024  17:50:50 +0300 (0:00:10.914)       0:00:34.095 ***
changed: [localhost]

TASK [setup_vpc : Create gamenet security group] ********************************
Wednesday 08 May 2024  17:50:57 +0300 (0:00:06.638)       0:00:40.734 ***
changed: [localhost]

TASK [create_keys : Create a new EC2 key pair] **********************************
Wednesday 08 May 2024  17:51:03 +0300 (0:00:06.003)       0:00:46.738 ***
ok: [localhost]

TASK [create_keys : Save the key pair] ******************************************
Wednesday 08 May 2024  17:51:04 +0300 (0:00:01.321)       0:00:48.059 ***
skipping: [localhost]

PLAY [Setup AWS CTF slot VPC security groups] ***********************************
```

```
PLAY RECAP *********************************************************************
localhost                  : ok=10   changed=6    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0

Wednesday 08 May 2024  17:51:04 +0300 (0:00:00.044)       0:00:48.104 ***
===============================================================================
setup_vpc : Configure VPC default route table ------------------------------ 10.91s
setup_vpc : Create subnets ------------------------------------------------- 10.21s
setup_vpc : Create default services security group ------------------------- 6.64s
setup_vpc : Create gamenet security group ---------------------------------- 6.00s
setup_vpc : Get default route table ---------------------------------------- 3.31s
setup_vpc : Create Internet Gateway for VPC -------------------------------- 3.27s
setup_vpc : Get available AZs ---------------------------------------------- 3.20s
setup_vpc : Create a new VPC ----------------------------------------------- 3.09s
create_keys : Create a new EC2 key pair ------------------------------------ 1.32s
setup_vpc : Define allocated ipv6 block ------------------------------------ 0.09s
create_keys : Save the key pair -------------------------------------------- 0.04s
Playbook run took 0 days, 0 hours, 0 minutes, 48 seconds
```

Lisa 4 – Ansible väljund IAM kasutaja loomisest

```
TASK [vlm_iam : Create IAM policy for i1 VLM] *********************************
Thursday 09 May 2024  10:41:27 +0300 (0:00:00.043)       0:00:18.405 ***
changed: [localhost]

TASK [vlm_iam : Create AWS user for VLM] **************************************
Thursday 09 May 2024  10:41:32 +0300 (0:00:05.400)       0:00:23.806 ***
[DEPRECATION WARNING]: The 'iam_user' return key is deprecated and will be replaced by 'user'.
Both values are returned for now. This feature will
be removed from amazon.aws in a release after 2024-05-01. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
changed: [localhost]

TASK [vlm_iam : Get AWS user access key info] *********************************
Thursday 09 May 2024  10:41:35 +0300 (0:00:02.662)       0:00:26.469 ***
ok: [localhost]

TASK [vlm_iam : Create a new access key for the VLM user] *********************
Thursday 09 May 2024  10:41:36 +0300 (0:00:01.677)       0:00:28.146 ***
changed: [localhost]

TASK [vlm_iam : Print keys] ***************************************************
Thursday 09 May 2024  10:41:38 +0300 (0:00:01.782)       0:00:29.928 ***
ok: [localhost] =>
  msg:
    access_key:
      access_key_id: ***************
      create_date: '2024-05-09T07:41:38+00:00'
      status: Active
      user_name: i1.vlm.a12.service
    access_key_id: ***************
    changed: true
    failed: false
    secret_access_key: ****************************

PLAY [Setup AWS CTF slot VPC security groups] *********************************

PLAY RECAP ********************************************************************
localhost                  : ok=15   changed=3    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0

Thursday 09 May 2024  10:41:38 +0300 (0:00:00.065)       0:00:29.994 ***
===============================================================================
vlm_iam : Create IAM policy for i1 VLM ----------------------------------- 5.40s
setup_vpc : Create subnets ----------------------------------------------- 3.68s
setup_vpc : Create default services security group ----------------------- 2.89s
vlm_iam : Create AWS user for VLM ---------------------------------------- 2.66s
setup_vpc : Create gamenet security group -------------------------------- 2.21s
setup_vpc : Create a new VPC --------------------------------------------- 1.96s
vlm_iam : Create a new access key for the VLM user ----------------------- 1.78s
setup_vpc : Configure VPC default route table ---------------------------- 1.71s
vlm_iam : Get AWS user access key info ----------------------------------- 1.68s
create_keys : Create a new EC2 key pair ---------------------------------- 1.66s
setup_vpc : Create Internet Gateway for VPC ------------------------------ 1.47s
```

```
setup_vpc : Get default route table ----------------------------------------- 1.42s
setup_vpc : Get available AZs ----------------------------------------------- 1.29s
vlm_iam : Print keys -------------------------------------------------------- 0.07s
create_keys : Save the key pair --------------------------------------------- 0.04s
setup_vpc : Define allocated ipv6 block ------------------------------------- 0.04s
```

Lisa 5 – Ansible väljund turvagruppide loomisest

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
playbook vpc.yml -t sec

PLAY [localhost] ***************************************************************

PLAY [Setup AWS CTF slot VPC security groups] **********************************

TASK [ctf_sec_grp : amazon.aws.ec2_vpc_net_info] *******************************
Wednesday 08 May 2024  18:06:42 +0300 (0:00:00.047)       0:00:00.047 ***
ok: [localhost]

TASK [ctf_sec_grp : Create slot security groups] *******************************
Wednesday 08 May 2024  18:06:44 +0300 (0:00:01.631)       0:00:01.678 ***
changed: [localhost] => (item=1)
changed: [localhost] => (item=2)

PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

Wednesday 08 May 2024  18:06:57 +0300 (0:00:12.458)       0:00:14.137 ***
===============================================================================
ctf_sec_grp : Create slot security groups ---------------------------------- 12.46s
ctf_sec_grp : amazon.aws.ec2_vpc_net_info ---------------------------------- 1.63s
Playbook run took 0 days, 0 hours, 0 minutes, 14 seconds
```

Lisa 6 – Ansible inventory päring vLM-Next'ist

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
inventory --list
{
    "_meta": {
        "hostvars": {
            "a12-ami-kali": {
                "ansible_become": true,
                "ansible_become_method": "sudo",
                "ansible_become_user": "root",
                "ansible_host": "{{ connection_ip | default(omit) }}",
                "ansible_shell_type": "sh",
                "ansible_user": "kali",
                "aws_account_id": ************,
                "aws_ami_name": "kali-last-snapshot-amd64-*",
                "aws_ami_owners": "************",
                "aws_ec2_vol_size": "50",
                "aws_ec2_vol_type": "gp2",
                "aws_instance_type": "m5.large",
                "aws_key_name": "{{ aws_prefix }}-kp",
                "aws_prefix": "a12",
                "aws_region": "eu-north-1",
                "aws_security_group": [
                    "{{ aws_prefix }}-services-01-sg"
                ],
                "access_ipv4": "**********/**",
                "access_ipv6": "**********/**",
                "cid": 1,
                "cpus": 2,
                "crp_id": 1,
                "deploy_api": "aws_ec2",
                "disk_size": 50,
                "eid": 12,
                "env_id": 12,
                "env_networks": {
                    "shared": {
                        "0": {
                            "ipv4": "10.80.64.0",
                            "ipv4_mask": "255.255.192.0",
                            "ipv6": null,
                            "network_name": "a12-gamenet-01-subnet",
                            "network_type": "SHARED",
                            "type": "segment"
                        },
                        "1": {
                            "ipv4": "10.80.0.0",
                            "ipv4_mask": "255.255.240.0",
                            "ipv6": null,
                            "network_name": "a12-services-01-subnet",
                            "network_type": "SHARED",
                            "type": "segment"
                        }
                    }
```

```
    },
    "infra_id": 1,
    "infra_prefix": "i1",
    "module_name": "A12-CTF",
    "nics": [
        {
            "addresses": [],
            "entity_type": "SEGMENT",
            "label": "nic-1",
            "module_network_name": "services-01",
            "network_name": "a12-services-01-subnet",
            "network_type": "SHARED",
            "primary": true
        }
    ],
    "project_name": "sander-testing",
    "ram": 4096,
    "regenerate_access_keys": false,
    "route53_zone": "*************",
    "slot_count": 2,
    "slot_id_2d": "{{ '%02d' | format(slot_id) }}",
    "slot_id_3d": "{{ '%03d' | format(slot_id) }}",
    "transport_protocol": "ssh",
    "vault_aws_access_key": "***/****:*******",
    "vault_aws_secret_key": "***/****:*******",
    "vault_aws_ssh_priv_key": "***/****:*******",
    "vlm_next": true,
    "vlm_source_ip": [
        "**********/**",
        "**********/**"
    ],
    "vpc_cidr_block": "10.80.0.0/16",
    "vpc_subnets": [
        {
            "ipv4_subnet": "10.80.0.0/20",
            "name": "services-01"
        },
        {
            "ipv4_subnet": "10.80.64.0/18",
            "name": "gamenet-01"
        }
    ]
},
"a12-ami-target": {
    "ansible_become": true,
    "ansible_become_method": "sudo",
    "ansible_become_user": "root",
    "ansible_host": "{{ connection_ip | default(omit) }}",
    "ansible_shell_type": "sh",
    "ansible_user": "kali",
    "aws_account_id": *************,
    "aws_ami_name": "kali-last-snapshot-amd64-*",
    "aws_ami_owners": "*************",
    "aws_ec2_vol_size": "50",
    "aws_ec2_vol_type": "gp2",
    "aws_instance_type": "m5.large",
    "aws_key_name": "{{ aws_prefix }}-kp",
    "aws_prefix": "a12",
    "aws_region": "eu-north-1",
    "aws_security_group": [
        "{{ aws_prefix }}-services-01-sg"
```

```
"access_ipv4": "**********/**",
"access_ipv6": "**********/**",
"cid": 1,
"cpus": 2,
"crp_id": 1,
"deploy_api": "aws_ec2",
"disk_size": 50,
"eid": 12,
"env_id": 12,
"env_networks": {
    "shared": {
        "0": {
            "ipv4": "10.80.64.0",
            "ipv4_mask": "255.255.192.0",
            "ipv6": null,
            "network_name": "a12-gamenet-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        },
        "1": {
            "ipv4": "10.80.0.0",
            "ipv4_mask": "255.255.240.0",
            "ipv6": null,
            "network_name": "a12-services-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        }
    }
},
"infra_id": 1,
"infra_prefix": "i1",
"module_name": "A12-CTF",
"nics": [
    {
        "addresses": [],
        "entity_type": "SEGMENT",
        "label": "nic-1",
        "module_network_name": "services-01",
        "network_name": "a12-services-01-subnet",
        "network_type": "SHARED",
        "primary": true
    }
],
"project_name": "sander-testing",
"ram": 4096,
"regenerate_access_keys": false,
"route53_zone": "*************",
"slot_count": 2,
"slot_id_2d": "{{ '%02d' | format(slot_id) }}",
"slot_id_3d": "{{ '%03d' | format(slot_id) }}",
"transport_protocol": "ssh",
"vault_aws_access_key": "***/****:*******",
"vault_aws_secret_key": "***/****:*******",
"vault_aws_ssh_priv_key": "***/****:*******",
"vlm_next": true,
"vlm_source_ip": [
    "**********/**",
    "**********/**"
],
"vpc_cidr_block": "10.80.0.0/16",
"vpc_subnets": [
```

```
        {
            "ipv4_subnet": "10.80.0.0/20",
            "name": "services-01"
        },
        {
            "ipv4_subnet": "10.80.64.0/18",
            "name": "gamenet-01"
        }
    ]
},
"kali01.s001.a12.crp.sh": {
    "ansible_become": true,
    "ansible_become_method": "sudo",
    "ansible_become_user": "root",
    "ansible_host": "{{ connection_ip | default(omit) }}",
    "ansible_shell_type": "sh",
    "ansible_user": "kali",
    "aws_account_id": ************,
    "aws_ec2_vol_size": "50",
    "aws_ec2_vol_type": "gp2",
    "aws_image_id": "ami-071aead8ba82810fd",
    "aws_instance_type": "t3.medium",
    "aws_key_name": "{{ aws_prefix }}-kp",
    "aws_prefix": "a12",
    "aws_region": "eu-north-1",
    "aws_security_group": [
        "{{ aws_prefix }}-gamenet-01-sg",
        "{{ aws_prefix }}-s{{ slot_id_3d }}-sg"
    ],
    "access_ipv4": "**********/**",
    "access_ipv6": "**********/**",
    "cid": 1,
    "cpus": 2,
    "crp_id": 1,
    "deploy_api": "aws_ec2",
    "disk_size": 50,
    "eid": 12,
    "env_id": 12,
    "env_networks": {
        "shared": {
            "0": {
                "ipv4": "10.80.64.0",
                "ipv4_mask": "255.255.192.0",
                "ipv6": null,
                "network_name": "a12-gamenet-01-subnet",
                "network_type": "SHARED",
                "type": "segment"
            },
            "1": {
                "ipv4": "10.80.0.0",
                "ipv4_mask": "255.255.240.0",
                "ipv6": null,
                "network_name": "a12-services-01-subnet",
                "network_type": "SHARED",
                "type": "segment"
            }
        }
    },
    "infra_id": 1,
    "infra_prefix": "i1",
    "module_name": "A12-CTF",
```

```
    "nics": [
        {
            "addresses": [
                {
                    "dhcp": true,
                    "stack": "ipv4"
                }
            ],
            "entity_type": "SEGMENT",
            "label": "nic-1",
            "module_network_name": "gamenet-01",
            "network_name": "a12-gamenet-01-subnet",
            "network_type": "SHARED",
            "primary": true
        }
    ],
    "project_name": "sander-testing",
    "ram": 4096,
    "regenerate_access_keys": false,
    "route53_zone": "*************",
    "sid": 1,
    "slot_count": 2,
    "slot_id": 1,
    "slot_id_2d": "{{ '%02d' | format(slot_id) }}",
    "slot_id_3d": "{{ '%03d' | format(slot_id) }}",
    "slot_networks": {
        "dedicated": {}
    },
    "transport_protocol": "ssh",
    "vault_aws_access_key": "***/****:*******",
    "vault_aws_secret_key": "***/****:*******",
    "vault_aws_ssh_priv_key": "***/****:*******",
    "vlm_next": true,
    "vlm_source_ip": [
        "**********/**",
        "**********/**"
    ],
    "vpc_cidr_block": "10.80.0.0/16",
    "vpc_subnets": [
        {
            "ipv4_subnet": "10.80.0.0/20",
            "name": "services-01"
        },
        {
            "ipv4_subnet": "10.80.64.0/18",
            "name": "gamenet-01"
        }
    ]
},
"kali01.s002.a12.crp.sh": {
    "ansible_become": true,
    "ansible_become_method": "sudo",
    "ansible_become_user": "root",
    "ansible_host": "{{ connection_ip | default(omit) }}",
    "ansible_shell_type": "sh",
    "ansible_user": "kali",
    "aws_account_id": *************,
    "aws_ec2_vol_size": "50",
    "aws_ec2_vol_type": "gp2",
    "aws_image_id": "ami-071aead8ba82810fd",
    "aws_instance_type": "t3.medium",
```

```
"aws_key_name": "{{ aws_prefix }}-kp",
"aws_prefix": "a12",
"aws_region": "eu-north-1",
"aws_security_group": [
    "{{ aws_prefix }}-gamenet-01-sg",
    "{{ aws_prefix }}-s{{ slot_id_3d }}-sg"
],
"access_ipv4": "**********/**",
"access_ipv6": "**********/**",
"cid": 1,
"cpus": 2,
"crp_id": 1,
"deploy_api": "aws_ec2",
"disk_size": 50,
"eid": 12,
"env_id": 12,
"env_networks": {
    "shared": {
        "0": {
            "ipv4": "10.80.64.0",
            "ipv4_mask": "255.255.192.0",
            "ipv6": null,
            "network_name": "a12-gamenet-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        },
        "1": {
            "ipv4": "10.80.0.0",
            "ipv4_mask": "255.255.240.0",
            "ipv6": null,
            "network_name": "a12-services-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        }
    }
},
"infra_id": 1,
"infra_prefix": "i1",
"module_name": "A12-CTF",
"nics": [
    {
        "addresses": [
            {
                "dhcp": true,
                "stack": "ipv4"
            }
        ],
        "entity_type": "SEGMENT",
        "label": "nic-1",
        "module_network_name": "gamenet-01",
        "network_name": "a12-gamenet-01-subnet",
        "network_type": "SHARED",
        "primary": true
    }
],
"project_name": "sander-testing",
"ram": 4096,
"regenerate_access_keys": false,
"route53_zone": "*************",
"sid": 2,
"slot_count": 2,
```

```
    "slot_id": 2,
    "slot_id_2d": "{{ '%02d' | format(slot_id) }}",
    "slot_id_3d": "{{ '%03d' | format(slot_id) }}",
    "slot_networks": {
        "dedicated": {}
    },
    "transport_protocol": "ssh",
    "vault_aws_access_key": "***/****:*******",
    "vault_aws_secret_key": "***/****:*******",
    "vault_aws_ssh_priv_key": "***/****:*******",
    "vlm_next": true,
    "vlm_source_ip": [
        "**********/**",
        "**********/**"
    ],
    "vpc_cidr_block": "10.80.0.0/16",
    "vpc_subnets": [
        {
            "ipv4_subnet": "10.80.0.0/20",
            "name": "services-01"
        },
        {
            "ipv4_subnet": "10.80.64.0/18",
            "name": "gamenet-01"
        }
    ]
},
"target01.s001.a12.crp.sh": {
    "ansible_become": true,
    "ansible_become_method": "sudo",
    "ansible_become_user": "root",
    "ansible_host": "{{ connection_ip | default(omit) }}",
    "ansible_shell_type": "sh",
    "ansible_user": "kali",
    "aws_account_id": ************,
    "aws_ec2_vol_size": "50",
    "aws_ec2_vol_type": "gp2",
    "aws_image_id": "ami-0e588f1a58a794f4c",
    "aws_instance_type": "t3.medium",
    "aws_key_name": "{{ aws_prefix }}-kp",
    "aws_prefix": "a12",
    "aws_region": "eu-north-1",
    "aws_security_group": [
        "{{ aws_prefix }}-gamenet-01-sg",
        "{{ aws_prefix }}-s{{ slot_id_3d }}-sg"
    ],
    "access_ipv4": "**********/**",
    "access_ipv6": "**********/**",
    "cid": 1,
    "cpus": 2,
    "crp_id": 1,
    "deploy_api": "aws_ec2",
    "disk_size": 50,
    "eid": 12,
    "env_id": 12,
    "env_networks": {
        "shared": {
            "0": {
                "ipv4": "10.80.64.0",
                "ipv4_mask": "255.255.192.0",
                "ipv6": null,
```

```
            "network_name": "a12-gamenet-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        },
        "1": {
            "ipv4": "10.80.0.0",
            "ipv4_mask": "255.255.240.0",
            "ipv6": null,
            "network_name": "a12-services-01-subnet",
            "network_type": "SHARED",
            "type": "segment"
        }
    }
},
"infra_id": 1,
"infra_prefix": "i1",
"module_name": "A12-CTF",
"nics": [
    {
        "addresses": [
            {
                "dhcp": true,
                "stack": "ipv4"
            }
        ],
        "entity_type": "SEGMENT",
        "label": "nic-1",
        "module_network_name": "gamenet-01",
        "network_name": "a12-gamenet-01-subnet",
        "network_type": "SHARED",
        "primary": true
    }
],
"project_name": "sander-testing",
"ram": 4096,
"regenerate_access_keys": false,
"route53_zone": "*************",
"sid": 1,
"slot_count": 2,
"slot_id": 1,
"slot_id_2d": "{{ '%02d' | format(slot_id) }}",
"slot_id_3d": "{{ '%03d' | format(slot_id) }}",
"slot_networks": {
    "dedicated": {}
},
"transport_protocol": "ssh",
"vault_aws_access_key": "***/****:*******",
"vault_aws_secret_key": "***/****:*******",
"vault_aws_ssh_priv_key": "***/****:*******",
"vlm_next": true,
"vlm_source_ip": [
    "**********/**",
    "**********/**"
],
"vpc_cidr_block": "10.80.0.0/16",
"vpc_subnets": [
    {
        "ipv4_subnet": "10.80.0.0/20",
        "name": "services-01"
    },
    {
```

```
            "ipv4_subnet": "10.80.64.0/18",
            "name": "gamenet-01"
        }
    ]
},
"target01.s002.a12.crp.sh": {
    "ansible_become": true,
    "ansible_become_method": "sudo",
    "ansible_become_user": "root",
    "ansible_host": "{{ connection_ip | default(omit) }}",
    "ansible_shell_type": "sh",
    "ansible_user": "kali",
    "aws_account_id": ************,
    "aws_ec2_vol_size": "50",
    "aws_ec2_vol_type": "gp2",
    "aws_image_id": "ami-0e588f1a58a794f4c",
    "aws_instance_type": "t3.medium",
    "aws_key_name": "{{ aws_prefix }}-kp",
    "aws_prefix": "a12",
    "aws_region": "eu-north-1",
    "aws_security_group": [
        "{{ aws_prefix }}-gamenet-01-sg",
        "{{ aws_prefix }}-s{{ slot_id_3d }}-sg"
    ],
    "access_ipv4": "**********/**",
    "access_ipv6": "**********/**",
    "cid": 1,
    "cpus": 2,
    "crp_id": 1,
    "deploy_api": "aws_ec2",
    "disk_size": 50,
    "eid": 12,
    "env_id": 12,
    "env_networks": {
        "shared": {
            "0": {
                "ipv4": "10.80.64.0",
                "ipv4_mask": "255.255.192.0",
                "ipv6": null,
                "network_name": "a12-gamenet-01-subnet",
                "network_type": "SHARED",
                "type": "segment"
            },
            "1": {
                "ipv4": "10.80.0.0",
                "ipv4_mask": "255.255.240.0",
                "ipv6": null,
                "network_name": "a12-services-01-subnet",
                "network_type": "SHARED",
                "type": "segment"
            }
        }
    },
    "infra_id": 1,
    "infra_prefix": "i1",
    "module_name": "A12-CTF",
    "nics": [
        {
            "addresses": [
                {
                    "dhcp": true,
```

```
                            "stack": "ipv4"
                        }
                    ],
                    "entity_type": "SEGMENT",
                    "label": "nic-1",
                    "module_network_name": "gamenet-01",
                    "network_name": "a12-gamenet-01-subnet",
                    "network_type": "SHARED",
                    "primary": true
                }
            ],
            "project_name": "sander-testing",
            "ram": 4096,
            "regenerate_access_keys": false,
            "route53_zone": "*************",
            "sid": 2,
            "slot_count": 2,
            "slot_id": 2,
            "slot_id_2d": "{{ '%02d' | format(slot_id) }}",
            "slot_id_3d": "{{ '%03d' | format(slot_id) }}",
            "slot_networks": {
                "dedicated": {}
            },
            "transport_protocol": "ssh",
            "vault_aws_access_key": "***/****:*******",
            "vault_aws_secret_key": "***/****:*******",
            "vault_aws_ssh_priv_key": "***/****:*******",
            "vlm_next": true,
            "vlm_source_ip": [
                "**********/**",
                "**********/**"
            ],
            "vpc_cidr_block": "10.80.0.0/16",
            "vpc_subnets": [
                {
                    "ipv4_subnet": "10.80.0.0/20",
                    "name": "services-01"
                },
                {
                    "ipv4_subnet": "10.80.64.0/18",
                    "name": "gamenet-01"
                }
            ]
        }
    }
},
"all": {
    "children": [
        "ungrouped",
        "environment",
        "ami_kali",
        "slot",
        "module"
    ]
},
"environment": {
    "children": [
        "slot1",
        "slot2",
        "slot3",
        "slot4",
```

```
            "slot5",
            "slot6",
            "slot7",
            "slot8",
            "slot9",
            "slot10"
        ],
        "hosts": [
            "a12-ami-kali",
            "a12-ami-target"
        ]
    },
    "module": {
        "hosts": [
            "a12-ami-kali",
            "a12-ami-target",
            "kali01.s001.a12.crp.sh",
            "kali01.s002.a12.crp.sh",
            "target01.s001.a12.crp.sh",
            "target01.s002.a12.crp.sh"
        ]
    },
    "slot": {
        "hosts": [
            "kali01.s001.a12.crp.sh",
            "kali01.s002.a12.crp.sh",
            "target01.s001.a12.crp.sh",
            "target01.s002.a12.crp.sh"
        ]
    },
    "slot1": {
        "hosts": [
            "kali01.s001.a12.crp.sh",
            "target01.s001.a12.crp.sh"
        ]
    },
    "slot2": {
        "hosts": [
            "kali01.s002.a12.crp.sh",
            "target01.s002.a12.crp.sh"
        ]
    }
}
```

Lisa 7 – Ansible väljund Kali AMI loomisest

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
playbook -l a12-ami-kali main.yml

PLAY [Deploy VM] ***********************************
Friday 10 May 2024  17:17:28 +0300 (0:00:00.106)       0:00:00.106

TASK [configure_vars : Configure AWS account] *********
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:29 +0300 (0:00:00.508)       0:00:00.615

TASK [configure_vars : Check if SSH key exists] *******
ok: [a12-ami-kali -> localhost]
Friday 10 May 2024  17:17:30 +0300 (0:00:01.332)       0:00:01.948

TASK [configure_vars : Get SSH key] ******************
skipping: [a12-ami-kali]
Friday 10 May 2024  17:17:30 +0300 (0:00:00.047)       0:00:01.995

TASK [configure_vars : Configure SSH key] *************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:30 +0300 (0:00:00.069)       0:00:02.064

TASK [deploy : Lookup instance] ***********************
ok: [a12-ami-kali -> localhost]
Friday 10 May 2024  17:17:33 +0300 (0:00:02.292)       0:00:04.357

TASK [deploy : Setting new_deploy status] *************
ok: [a12-ami-kali -> localhost]
Friday 10 May 2024  17:17:33 +0300 (0:00:00.087)       0:00:04.444

TASK [deploy : Undeploy instance] *********************
skipping: [a12-ami-kali]
Friday 10 May 2024  17:17:33 +0300 (0:00:00.054)       0:00:04.499

TASK [deploy : Deploy instance] ***********************
skipping: [a12-ami-kali]
Friday 10 May 2024  17:17:33 +0300 (0:00:00.058)       0:00:04.557

TASK [deploy : Start instance] ************************
skipping: [a12-ami-kali]
Friday 10 May 2024  17:17:33 +0300 (0:00:00.057)       0:00:04.615

TASK [connection : Set connection params] **************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:33 +0300 (0:00:00.081)       0:00:04.696

TASK [connection : Waiting 60 seconds for system to become reachable] **************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:36 +0300 (0:00:03.513)       0:00:08.210

TASK [aws_kali_gui : Update apt cache] *****************
ok: [a12-ami-kali]
```

```
Friday 10 May 2024  17:17:39 +0300 (0:00:02.391)       0:00:10.601

TASK [aws_kali_gui : Removing cloud managed hostname] **
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:41 +0300 (0:00:01.937)       0:00:12.539

TASK [aws_kali_gui : Install kali desktop environment] *
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:43 +0300 (0:00:01.888)       0:00:14.428

TASK [aws_kali_gui : Install kali desktop tools] *******
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:44 +0300 (0:00:01.777)       0:00:16.206

TASK [aws_kali_gui : Install VNC server] ***************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:47 +0300 (0:00:02.015)       0:00:18.221

TASK [aws_kali_gui : Enable password login] ************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:50 +0300 (0:00:03.059)       0:00:21.281

TASK [aws_kali_gui : Enable root login] ****************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:51 +0300 (0:00:01.332)       0:00:22.613

TASK [aws_kali_gui : Change root password] *************
changed: [a12-ami-kali]
Friday 10 May 2024  17:17:55 +0300 (0:00:04.268)       0:00:26.882

TASK [aws_kali_gui : Change kali password] *************
changed: [a12-ami-kali]
Friday 10 May 2024  17:17:57 +0300 (0:00:01.335)       0:00:28.218

TASK [aws_kali_gui : Set user VNC password] ************
included: /Users/sanderpluks/Work/crp-toolkit/automation-project-cloud/cloud-
deploy/roles/aws_kali_gui/tasks/vnc-passwd.yml for a12-ami-kali => (item={'username': 'root',
'home_dir': '/root', 'vnc_password': '**************'})
Friday 10 May 2024  17:17:57 +0300 (0:00:00.233)       0:00:28.451

TASK [aws_kali_gui : Create .vnc directory] ************
ok: [a12-ami-kali]
Friday 10 May 2024  17:17:59 +0300 (0:00:02.446)       0:00:30.898

TASK [aws_kali_gui : Set VNC password for VNC user] *****
ok: [a12-ami-kali]
Friday 10 May 2024  17:18:01 +0300 (0:00:01.942)       0:00:32.840

TASK [aws_kali_gui : Set correct permissions for VNC passwd file] ******************
ok: [a12-ami-kali]
Friday 10 May 2024  17:18:02 +0300 (0:00:01.200)       0:00:34.041

TASK [aws_kali_gui : Copy VNC session configuration] ****
ok: [a12-ami-kali] => (item={'username': 'root', 'home_dir': '/root'})
Friday 10 May 2024  17:18:04 +0300 (0:00:01.990)       0:00:36.031

TASK [aws_kali_gui : Copy root VNC service] ************
ok: [a12-ami-kali]
Friday 10 May 2024  17:18:06 +0300 (0:00:01.728)       0:00:37.760

TASK [aws_kali_gui : Start vnc service for root] ********
ok: [a12-ami-kali]
```

```
Friday 10 May 2024  17:18:11 +0300 (0:00:04.885)       0:00:42.645

TASK [aws_kali_gui : Set basic background] **************
changed: [a12-ami-kali]
Friday 10 May 2024  17:18:12 +0300 (0:00:00.888)       0:00:43.533

TASK [aws_kali_gui : Clean logs] ************************
changed: [a12-ami-kali]
Friday 10 May 2024  17:18:13 +0300 (0:00:01.059)       0:00:44.592

TASK [aws_kali_gui : Clean VNC logs] ********************
changed: [a12-ami-kali]
Friday 10 May 2024  17:18:14 +0300 (0:00:01.101)       0:00:45.694

TASK [aws_kali_gui : Reboot] ****************************
changed: [a12-ami-kali]

PLAY RECAP **********************************************
a12-ami-kali               : ok=27   changed=6    unreachable=0    failed=0    skipped=4
rescued=0    ignored=0

Friday 10 May 2024  17:18:48 +0300 (0:00:33.996)       0:01:19.691
===============================================================================
aws_kali_gui : Reboot ---------------------------- 34.00s
aws_kali_gui : Start vnc service for root --------- 4.89s
aws_kali_gui : Change root password --------------- 4.27s
connection : Waiting 60 seconds for system to become reachable -- 3.51s
aws_kali_gui : Enable password login -------------- 3.06s
aws_kali_gui : Create .vnc directory -------------- 2.45s
aws_kali_gui : Update apt cache ------------------- 2.39s
deploy : Lookup instance -------------------------- 2.29s
aws_kali_gui : Install VNC server ----------------- 2.01s
aws_kali_gui : Copy VNC session configuration ----- 1.99s
aws_kali_gui : Set VNC password for VNC user ------ 1.94s
aws_kali_gui : Removing cloud managed hostname ---- 1.94s
aws_kali_gui : Install kali desktop environment --- 1.89s
aws_kali_gui : Install kali desktop tools --------- 1.78s
aws_kali_gui : Copy root VNC service -------------- 1.73s
aws_kali_gui : Change kali password --------------- 1.34s
configure_vars : Check if SSH key exists ---------- 1.33s
aws_kali_gui : Enable root login ------------------ 1.33s
aws_kali_gui : Set correct permissions for VNC passwd file -- 1.20s
aws_kali_gui : Clean VNC logs --------------------- 1.10s
Playbook run took 0 days, 0 hours, 1 minutes, 19 seconds
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗)
```

Lisa 8 – Ansible väljund Target AMI loomisest

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
playbook -l a12-ami-target main.yml

PLAY [Deploy VM] ***********************************
Friday 10 May 2024  18:07:39 +0300 (0:00:00.091)       0:00:00.091

TASK [configure_vars : Configure AWS account] *********
ok: [a12-ami-target]
Friday 10 May 2024  18:07:40 +0300 (0:00:00.350)       0:00:00.442

TASK [configure_vars : Check if SSH key exists] *******
ok: [a12-ami-target -> localhost]
Friday 10 May 2024  18:07:40 +0300 (0:00:00.754)       0:00:01.196

TASK [configure_vars : Get SSH key] ******************
skipping: [a12-ami-target]
Friday 10 May 2024  18:07:40 +0300 (0:00:00.039)       0:00:01.236

TASK [configure_vars : Configure SSH key] *************
ok: [a12-ami-target]
Friday 10 May 2024  18:07:40 +0300 (0:00:00.055)       0:00:01.291

TASK [deploy : Lookup instance] ***********************
ok: [a12-ami-target -> localhost]
Friday 10 May 2024  18:07:42 +0300 (0:00:01.710)       0:00:03.001

TASK [deploy : Setting new_deploy status] *************
ok: [a12-ami-target -> localhost]
Friday 10 May 2024  18:07:42 +0300 (0:00:00.062)       0:00:03.064

TASK [deploy : Undeploy instance] *********************
skipping: [a12-ami-target]
Friday 10 May 2024  18:07:42 +0300 (0:00:00.041)       0:00:03.105

TASK [deploy : Deploy instance] ***********************
skipping: [a12-ami-target]
Friday 10 May 2024  18:07:42 +0300 (0:00:00.041)       0:00:03.147

TASK [deploy : Start instance] ************************
skipping: [a12-ami-target]
Friday 10 May 2024  18:07:42 +0300 (0:00:00.050)       0:00:03.197

TASK [connection : Set connection params] **************
ok: [a12-ami-target]
Friday 10 May 2024  18:07:42 +0300 (0:00:00.072)       0:00:03.270

TASK [connection : Waiting 60 seconds for system to become reachable] **************
ok: [a12-ami-target]
Friday 10 May 2024  18:07:45 +0300 (0:00:02.363)       0:00:05.633

TASK [target01 : Install docker and necessary packages] *
ok: [a12-ami-target]
```

```
Friday 10 May 2024  18:07:47 +0300 (0:00:02.100)       0:00:07.733

TASK [target01 : Enable and start Docker service] *******
ok: [a12-ami-target]
Friday 10 May 2024  18:07:50 +0300 (0:00:02.985)       0:00:10.719

TASK [target01 : Download and install Docker Compose] ***
ok: [a12-ami-target]
Friday 10 May 2024  18:07:51 +0300 (0:00:01.624)       0:00:12.343

TASK [target01 : Copy hacking tasks archive to remote] **
ok: [a12-ami-target]
Friday 10 May 2024  18:07:53 +0300 (0:00:01.355)       0:00:13.699

TASK [target01 : Extract hacking tasks] *****************
changed: [a12-ami-target]
Friday 10 May 2024  18:07:58 +0300 (0:00:04.877)       0:00:18.576

TASK [target01 : Copy investigation tasks archive to remote] ***
changed: [a12-ami-target]
Friday 10 May 2024  18:07:59 +0300 (0:00:01.609)       0:00:20.186

TASK [target01 : Extract investigation tasks] ***********
changed: [a12-ami-target]
Friday 10 May 2024  18:08:01 +0300 (0:00:01.510)       0:00:21.696

TASK [target01 : Get list of docker-compose files] ******
ok: [a12-ami-target]
Friday 10 May 2024  18:08:02 +0300 (0:00:00.837)       0:00:22.533

TASK [target01 : Build CTF docker images] ****************
changed: [a12-ami-target] => (multiple items)

PLAY RECAP *****************************************
a12-ami-target             : ok=16   changed=4    unreachable=0    failed=0    skipped=4
rescued=0    ignored=0

Friday 10 May 2024  18:08:10 +0300 (0:00:08.767)       0:00:31.301
===============================================================================
target01 : Build CTF docker images -------------------- 8.77s
target01 : Extract hacking tasks ---------------------- 4.88s
target01 : Enable and start Docker service ------------ 2.99s
connection : Waiting 60 seconds for system to become reachable -- 2.36s
target01 : Install docker and necessary packages ----- 2.10s
deploy : Lookup instance ------------------------------ 1.71s
target01 : Download and install Docker Compose -------- 1.62s
target01 : Copy investigation tasks archive to remote - 1.61s
target01 : Extract investigation tasks ---------------- 1.51s
target01 : Copy hacking tasks archive to remote ------ 1.36s
target01 : Get list of docker-compose files ----------- 0.84s
configure_vars : Check if SSH key exists ------------- 0.75s
configure_vars : Configure AWS account --------------- 0.35s
connection : Set connection params ------------------- 0.07s
deploy : Setting new_deploy status ------------------- 0.06s
configure_vars : Configure SSH key ------------------- 0.06s
deploy : Start instance ------------------------------ 0.05s
deploy : Deploy instance ----------------------------- 0.04s
deploy : Undeploy instance --------------------------- 0.04s
configure_vars : Get SSH key ------------------------- 0.04s
Playbook run took 0 days, 0 hours, 0 minutes, 31 seconds
```

Lisa 9 – Ansible väljund küberkaitse kursuse masinate juurutamisest

```
(ansible-venv) ~/Work/crp-toolkit/automation-project-cloud/cloud-deploy (main ✗) ansible-
playbook -l '~.*01.s00[0-9].*' main.yml

PLAY [Deploy VM] ************
Friday 10 May 2024  19:10:11 +0300 (0:00:00.110)       0:00:00.110
Friday 10 May 2024  19:10:11 +0300 (0:00:00.015)       0:00:00.125
Friday 10 May 2024  19:10:11 +0300 (0:00:00.017)       0:00:00.143
Friday 10 May 2024  19:10:11 +0300 (0:00:00.018)       0:00:00.161

TASK [configure_vars : Configure AWS account] *************
ok: [kali01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:11 +0300 (0:00:00.488)       0:00:00.650
ok: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:11 +0300 (0:00:00.033)       0:00:00.683
ok: [kali01.s001.a12.crp.sh]
ok: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:11 +0300 (0:00:00.034)       0:00:00.718
Friday 10 May 2024  19:10:11 +0300 (0:00:00.024)       0:00:00.742

TASK [configure_vars : Check if SSH key exists] ***********
ok: [target01.s002.a12.crp.sh -> localhost]
ok: [target01.s001.a12.crp.sh -> localhost]
ok: [kali01.s002.a12.crp.sh -> localhost]
ok: [kali01.s001.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:13 +0300 (0:00:01.157)       0:00:01.900
Friday 10 May 2024  19:10:13 +0300 (0:00:00.017)       0:00:01.917
Friday 10 May 2024  19:10:13 +0300 (0:00:00.020)       0:00:01.938
Friday 10 May 2024  19:10:13 +0300 (0:00:00.026)       0:00:01.964

TASK [configure_vars : Get SSH key] ***********************
skipping: [kali01.s001.a12.crp.sh]
skipping: [kali01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:13 +0300 (0:00:00.027)       0:00:01.992
Friday 10 May 2024  19:10:13 +0300 (0:00:00.026)       0:00:02.018
skipping: [target01.s001.a12.crp.sh]
skipping: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:13 +0300 (0:00:00.022)       0:00:02.041
Friday 10 May 2024  19:10:13 +0300 (0:00:00.023)       0:00:02.065

TASK [configure_vars : Configure SSH key] *****************
ok: [kali01.s001.a12.crp.sh]
ok: [kali01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:13 +0300 (0:00:00.034)       0:00:02.099
ok: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:13 +0300 (0:00:00.029)       0:00:02.128
ok: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:13 +0300 (0:00:00.031)       0:00:02.160
Friday 10 May 2024  19:10:13 +0300 (0:00:00.026)       0:00:02.186

TASK [deploy : Lookup instance] ***************************
```

```
ok: [target01.s002.a12.crp.sh -> localhost]
ok: [target01.s001.a12.crp.sh -> localhost]
ok: [kali01.s001.a12.crp.sh -> localhost]
ok: [kali01.s002.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:16 +0300 (0:00:02.841)       0:00:05.028
Friday 10 May 2024  19:10:16 +0300 (0:00:00.020)       0:00:05.049
Friday 10 May 2024  19:10:16 +0300 (0:00:00.025)       0:00:05.075
Friday 10 May 2024  19:10:16 +0300 (0:00:00.026)       0:00:05.101

TASK [deploy : Setting new_deploy status] *****************
ok: [kali01.s001.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.038)       0:00:05.139
ok: [kali01.s002.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.038)       0:00:05.178
ok: [target01.s001.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.040)       0:00:05.219
ok: [target01.s002.a12.crp.sh -> localhost]

TASK [deploy : Undeploy instance] *************************
skipping: [kali01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.038)       0:00:05.258
Friday 10 May 2024  19:10:16 +0300 (0:00:00.032)       0:00:05.290
skipping: [kali01.s002.a12.crp.sh]
skipping: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.029)       0:00:05.320
Friday 10 May 2024  19:10:16 +0300 (0:00:00.027)       0:00:05.348

TASK [deploy : Deploy instance] ***************************

TASK [deploy : Undeploy instance] *************************
skipping: [target01.s002.a12.crp.sh]
included: /Users/sanderpluks/Work/crp-toolkit/automation-project-cloud/cloud-
deploy/roles/deploy/tasks/deploy.yml for kali01.s001.a12.crp.sh
Friday 10 May 2024  19:10:16 +0300 (0:00:00.070)       0:00:05.419
Friday 10 May 2024  19:10:16 +0300 (0:00:00.022)       0:00:05.441

TASK [deploy : Deploy instance] ***************************
included: /Users/sanderpluks/Work/crp-toolkit/automation-project-cloud/cloud-
deploy/roles/deploy/tasks/deploy.yml for kali01.s002.a12.crp.sh, target01.s001.a12.crp.sh
Friday 10 May 2024  19:10:16 +0300 (0:00:00.068)       0:00:05.510
Friday 10 May 2024  19:10:16 +0300 (0:00:00.020)       0:00:05.531

TASK [deploy : Lookup AMI] **
skipping: [kali01.s001.a12.crp.sh]

TASK [deploy : Deploy instance] ***************************
included: /Users/sanderpluks/Work/crp-toolkit/automation-project-cloud/cloud-
deploy/roles/deploy/tasks/deploy.yml for target01.s002.a12.crp.sh
Friday 10 May 2024  19:10:16 +0300 (0:00:00.098)       0:00:05.629
Friday 10 May 2024  19:10:16 +0300 (0:00:00.045)       0:00:05.675

TASK [deploy : Lookup AMI] **
skipping: [kali01.s002.a12.crp.sh]
skipping: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:16 +0300 (0:00:00.027)       0:00:05.702
Friday 10 May 2024  19:10:16 +0300 (0:00:00.030)       0:00:05.735

TASK [deploy : Source AMI id] *****************************
skipping: [kali01.s001.a12.crp.sh]

TASK [deploy : Lookup AMI] **
skipping: [target01.s002.a12.crp.sh]
```

```
Friday 10 May 2024  19:10:17 +0300 (0:00:00.040)       0:00:05.776
Friday 10 May 2024  19:10:17 +0300 (0:00:00.035)       0:00:05.811

TASK [deploy : Source AMI id] ***************************
skipping: [kali01.s002.a12.crp.sh]
skipping: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:17 +0300 (0:00:00.035)       0:00:05.847
Friday 10 May 2024  19:10:17 +0300 (0:00:00.037)       0:00:05.884

TASK [deploy : Source AMI id] ***************************
skipping: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:17 +0300 (0:00:00.042)       0:00:05.927

TASK [deploy : Get instance VPC subnet id] ****************
ok: [target01.s001.a12.crp.sh -> localhost]

TASK [deploy : Get instance VPC subnet id] ****************
ok: [target01.s002.a12.crp.sh -> localhost]

TASK [deploy : Get instance VPC subnet id] ****************
ok: [kali01.s001.a12.crp.sh -> localhost]

TASK [deploy : Get instance VPC subnet id] ****************
ok: [kali01.s002.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:23 +0300 (0:00:06.632)       0:00:12.559
Friday 10 May 2024  19:10:23 +0300 (0:00:00.108)       0:00:12.668
Friday 10 May 2024  19:10:24 +0300 (0:00:00.137)       0:00:12.805
Friday 10 May 2024  19:10:24 +0300 (0:00:00.152)       0:00:12.958

TASK [deploy : Deploy instance] ***************************
changed: [target01.s001.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:57 +0300 (0:00:33.503)       0:00:46.462

TASK [deploy : Start instance] ****************************
skipping: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:57 +0300 (0:00:00.082)       0:00:46.545

TASK [connection : Set connection params] *****************
ok: [target01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:57 +0300 (0:00:00.151)       0:00:46.697

TASK [deploy : Deploy instance] ***************************
changed: [kali01.s002.a12.crp.sh -> localhost]

TASK [deploy : Deploy instance] ***************************
changed: [kali01.s001.a12.crp.sh -> localhost]

TASK [deploy : Deploy instance] ***************************
changed: [target01.s002.a12.crp.sh -> localhost]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.109)       0:00:46.806
Friday 10 May 2024  19:10:58 +0300 (0:00:00.023)       0:00:46.830
Friday 10 May 2024  19:10:58 +0300 (0:00:00.031)       0:00:46.861

TASK [deploy : Start instance] ****************************
skipping: [kali01.s001.a12.crp.sh]
skipping: [kali01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.042)       0:00:46.904
skipping: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.038)       0:00:46.943
Friday 10 May 2024  19:10:58 +0300 (0:00:00.033)       0:00:46.976
```

```
TASK [connection : Set connection params] *****************
ok: [kali01.s001.a12.crp.sh]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.049)       0:00:47.026
ok: [kali01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.057)       0:00:47.083
ok: [target01.s002.a12.crp.sh]
Friday 10 May 2024  19:10:58 +0300 (0:00:00.034)       0:00:47.118

TASK [connection : Waiting 600 seconds for system to become reachable] ******************
ok: [target01.s001.a12.crp.sh]
ok: [target01.s002.a12.crp.sh]
ok: [kali01.s002.a12.crp.sh]
ok: [kali01.s001.a12.crp.sh]

PLAY RECAP ******************
kali01.s001.a12.crp.sh     : ok=10   changed=1    unreachable=0    failed=0    skipped=5
rescued=0    ignored=0
kali01.s002.a12.crp.sh     : ok=10   changed=1    unreachable=0    failed=0    skipped=5
rescued=0    ignored=0
target01.s001.a12.crp.sh   : ok=10   changed=1    unreachable=0    failed=0    skipped=5
rescued=0    ignored=0
target01.s002.a12.crp.sh   : ok=10   changed=1    unreachable=0    failed=0    skipped=5
rescued=0    ignored=0

Friday 10 May 2024  19:11:35 +0300 (0:00:37.167)       0:01:24.286
===============================================================================
connection : Waiting 600 seconds for system to become reachable - 37.37s
deploy : Deploy instance ----------------------------------------- 33.50s
deploy : Get instance VPC subnet id ------------------------------ 6.63s
deploy : Lookup instance ----------------------------------------- 2.93s
configure_vars : Check if SSH key exists ------------------------- 1.25s
configure_vars : Configure AWS account --------------------------- 0.54s
deploy : Deploy instance ----------------------------------------- 0.29s
connection : Set connection params ------------------------------- 0.27s
deploy : Deploy instance ----------------------------------------- 0.20s
deploy : Start instance ------------------------------------------ 0.18s
deploy : Undeploy instance --------------------------------------- 0.15s
deploy : Lookup AMI ---------------------------------------------- 0.12s
deploy : Setting new_deploy status ------------------------------- 0.11s
deploy : Deploy instance ----------------------------------------- 0.11s
configure_vars : Configure SSH key ------------------------------- 0.11s
configure_vars : Get SSH key ------------------------------------- 0.09s
deploy : Get instance VPC subnet id ------------------------------ 0.08s
deploy : Source AMI id ------------------------------------------- 0.07s
deploy : Source AMI id ------------------------------------------- 0.05s
deploy : Source AMI id ------------------------------------------- 0.04s
Playbook run took 0 days, 0 hours, 1 minutes, 24 seconds
```