VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Veronika Fišarová

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOSTNÍ CVIČENÍ NA PLATFORMĚ OPENSTACK
CYBER EXERCISES ON THE OPENSTACK PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Veronika Fišarová

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Tomáš Lieskovan

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Veronika Fišarová ***ID:*** 211784
Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Bezpečnostní cvičení na platformě OpenStack

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je seznámit se s platformami pro provádění bezpečnostních cvičení (tzv. cyber range). V teoretické části budou porovnány platformy a virtualizační nástroje vhodné pro bezpečnostní cvičení. Teoretickou část zaměřte primárně na virtualizační platformu OpenStack a cyber-range platformu KYPO včetně možností instalace. Na experimentálním pracovišti zprovozněte platformy OpenStack a KYPO. V tomto prostředí zprovozněte vzorový scénář k demonstraci funkčnosti.

DOPORUČENÁ LITERATURA:

[1] VIGNA, Giovanni. Teaching network security through live exercises. In: Security education and critical infrastructures. Springer, Boston, MA, 2003. p. 3-18.

[2] SIMPSON, Michael T.; BACKMAN, Kent; CORLEY, James. Hands-on ethical hacking and network defense. Cengage Learning, 2010.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: Ing. Tomáš Lieskovan

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Cílem bakalářské práce je seznámení s platformami, které se používají pro tvorbu cyber rangů. Hlavní část práce je zaměřena na cloudovou platformu OpenStack a její využití při tvorbě cyber range. Rozebrány jsou možnosti instalace OpenStacku a je vybrána instalace, která je vhodná pro nasazení projektu KYPO. Poslední část práce se věnuje analýze projektu KYPO a instalaci OpenStacku na virtuální stanici. Následuje nasazení platformy KYPO na vhodně upravený OpenStack a zprovoznění trénovacího scénáře.

KLÍČOVÁ SLOVA

cyber range, blue team, red team, capture the flag, OpenStack, KYPO projekt, virtualizace, Kolla-Ansible, Docker, Ansible, Vagrant

ABSTRACT

The goal of this bachelor thesis is to get acquainted with the platforms that are used to create the cyber ranges. The main part of thesis is focused on the cloud platform named OpenStack and its possibilites in creation of cyber ranges. The OpenStack installation options are compared and the selection of the instalation that is suitable for the KYPO project is made. The last part is devoted to the analysis of the KYPO project and the actual installation of OpenStack to the virtual station followed by the deployment of the KYPO platform on the modified OpenStack cloud and the commissioning of a training scenario.

KEYWORDS

cyber range, blue team, red team, capture the flag, OpenStack, the KYPO project, virtualisation, Kolla-Ansible, Docker, Ansible, Vagrant

FIŠAROVÁ, Veronika. *Bezpečnostní cvičení na platformě OpenStack*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 71 s. Bakalářská práce. Vedoucí práce: Ing. Tomáš Lieskovan

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Veronika Fišarová
VUT ID autora:	211784
Typ práce:	Bakalářská práce
Akademický rok:	2020/21
Téma závěrečné práce:	Bezpečnostní cvičení na platformě OpenStack

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu bakalářské práce panu Ing. Tomáši Lieskovanovi za odborné vedení a flexibilní přístup při řešení potíží a zároveň děkuji KYPO týmu Masarykovy univerzity za ochotný přístup při zodpovídání dotazů a nesrovnalostí.

Obsah

Úvod **12**

1 Bezpečnostní cvičení **14**
1.1 Typy bezpečnostních cvičení 14
1.2 Cyber range 15
1.3 Trénovací scénáře 15
1.3.1 Red team/Blue team 15
1.3.2 Capture the Flag (CTF) 15
1.4 Účastnické role 16

2 Cyber range platformy **17**
2.1 SecDevOps@Cuse CyberRange 17
2.1.1 Amazon Web Services (AWS) 17
2.2 CyTrONE 19
2.3 Alpaca 21
2.4 Security Scenario Generator (SecGen) 23
2.5 CTFd 24

3 OpenStack **26**
3.1 Architektura 26
3.2 Metody nasazení 28
3.2.1 DevStack 28
3.2.2 OpenStack-Ansible 29
3.2.3 OpenStack-Fuel 30
3.2.4 OpenStack-TripleO 31
3.2.5 OpenStack-RDO Packstack 32

4 Analýza projektu Kybernetický polygon **34**
4.1 Architektura 34
4.2 Shrnutí projektu KYPO 36
4.2.1 Související cyber range 36

5 Praktická část **38**
5.1 Původní nasazení platformy OpenStack 38
5.2 Konečné nasazení platformy OpenStack 39
5.2.1 Vyvtoření externí sítě 42
5.3 Nasazení platformy KYPO 43
5.3.1 Nasazení základní infrastruktury 44

5.3.2 Nasazení KYPO cyber range 48
5.4 Ovládání platformy KYPO 51
5.4.1 Zprovoznění ukázkového scénáře 53
5.4.2 Import vlastního sandboxu 58

Závěr **61**

Literatura **63**

Seznam symbolů a zkratek **69**

A Průběh cvičení ukázkového scénáře **71**

Seznam obrázků

2.1 Zjednodušená architektura SecDevOps@Cuse CyberRange, převzato z [11] 18
2.2 Architektura CyTrONE, převzato z [14] 20
2.3 Mřížka zranitelnosti, převzato z [21] 22
2.4 Postup vytvoření cyber range pomocí systému Alpaca, převzato z [21] . . . 22
3.1 Klíčové komponenty OpenStacku, převzato z [37] 26
4.1 Architektura KYPO cyber range se dvěma sandboxy, převzato z [41] 35
5.1 Základní infrastruktura KYPO 44
5.2 Základní infrastruktura KYPO v prostředí Horizon 47
5.3 Úplná infrastruktura platformy KYPO a OpenStacku 50
5.4 Portál KYPO platformy z pohledu hlavního uživatele 51
5.5 Topologie ukázkového scénáře zobrazená v prostředí Horizon 55
5.6 Topologie ukázkového scénáře na platformě KYPO 56
5.7 Přístup do instance sandboxu pomocí Spice konzole 57
5.8 Časový graf jednotlivých úkolů ukázkového scénáře 58

Seznam tabulek

2.1 Shrnutí projektu SecDevOps@Cuse CyberRange [11] 19
2.2 Shrnutí projektu CyTrONE [14, 16] 21
2.3 Shrnutí projektu Alpaca [21] 23
2.4 Shrnutí projektu SecGen [22, 23] 24
2.5 Shrnutí projektu CTFd [24] 25
3.1 Shrnutí metody nasazení DevStack [29, 30] 28
3.2 Shrnutí metody nasazení OpenStack-Ansible [31] 30
3.3 Shrnutí metody nasazení OpenStack-Fuel [32] 31
3.4 Shrnutí metody nasazení OpenStack-TripleO [33] 32
3.5 Shrnutí metody nasazení OpenStack RDO Packstack [34] 33
5.1 Parametry virtuálních stanic 39
5.2 Vyhrazené prostředky pro OpenStack instance 45
5.3 Bezpečnostní pravidla pro instance základní infrastruktury 46
5.4 Import definice sandboxu ukázkového scénáře 54
5.5 Import definice vlastního sandboxu 60

Seznam výpisů

5.1 Instalace balíčků a vytvoření virtuálního prostředí 40
5.2 Příprava konfiguračních souborů Ansible a Kolla-Ansible 40
5.3 Úprava souboru multinode 41
5.4 Úprava instalačního souboru globals.yml 41
5.5 Zahájení nasazení OpenStacku 42
5.6 Vytvoření externí sítě public 43
5.7 Splnění prerekvizit pro instalaci základní infrastruktury 45
5.8 Spuštění virtuálního prostředí 45
5.9 Upravený konfigurační soubor source openstack-defaults.sh 45
5.10 Vytvoření základní infrastruktury 46
5.11 Obsah souboru local-demo-extra-vars.yml 49
5.12 Vytvoření a spuštění virtuálního stroje s platformou KYPO 50
5.13 Upravené parametry pro GitLab repozitář 59
5.14 Spuštění nástroje Vagrant s upravenými GitLab parametry 60

Úvod

Komunikační a informační systémy (ICT systémy) se prudce rozšiřují a de facto každá infrastruktura je určitým způsobem připojena k veřejné síti - Internetu, a může se tedy stát přístupnou i mimo oprávněný kybernetický prostor. Vývoj těchto systémů dosáhl takové úrovně, že rezignace jedince nebo společnosti na komunikační a informační systémy by znemožnila její chod a způsobila izolaci od okolního světa. Páteřní entity státu, které jsou důležité pro jeho chod a rozvoj, jako ekonomika, zdravotnictví, školství, průmysl, obrana aj., jsou vzájemně propojené s mnoha dalšími entitami prostřednictvím ICT a jejich izolace nebo částečné přerušení není možné, a pokud nastane, negativně ovlivní fungování celé společnosti. Proto je nutné se zaměřit na zabezpečení ICT systémů, na kterých pracují zásadní entity, a u kterých byla bezpečnost nedostatečná vzhledem k jejich obrovskému a rychlému rozvoji [1].

Odolnost ICT systémů byla četně zkoušena v první třetině roku 2020 v rámci nástupu pandemie COVID-19. Jak už bylo několikrát historií potvrzeno, útočníci k provedení útoků využívají důležité události a krizové situace, například volby a pandemie. Toto tvrzení dokazuje hlášení, které uvedl Secretary General Jürgen Stock ve výkazu Interpolu z dubna roku 2020 (plný dokument dostupný ke stažení v literatuře [2]), cituji „Kyberzločinci vyvíjejí a zesilují své útoky alarmujícím tempem a využívají strach a nejistotu způsobenou nestabilní sociální a ekonomickou situací vyvolanou pandemií COVID-19.“[1] Příkladem této skutečnosti jsou kybernetické útoky na české nemocnice, které dokázaly, jak taková událost může ovlivnit normální provoz celé organizace. Ať už je důvod podobných kybernetických útoků jakýkoliv, ohrožují důvěru v bezpečnost společnosti i zdraví lidí. Bezpečnostní opatření ICT systémů je řízeno normami a jejich dodržování kontrolují určené orgány. Kombinace správné implementace těchto norem s pravidelnou údržbou ICT systémů snižuje riziko realizace těchto útoků, ale není možné ho zcela eliminovat. Proto je důležité, aby odborníci aktivně se podílející na bezpečnosti systémů, a aktiv do nich zahrnutých, byli řádně připraveni na situace, kdy se objeví příznaky nadcházejícího útoku, případně již probíhajícího a dokázali na ně bez prodlevy reagovat. Na tyto situace je obtížné se teoreticky připravit, a proto je v posledních dvou dekádách kladen velký důraz na praktickou přípravu, která je realizovaná pomocí bezpečnostních cvičení a vytváření cyber range prostředí. Státy si uvědomily, jak je praktická příprava proti kybernetickým útokům důležitá a k vytvoření vhodného prostředí rezervují finanční prostředky a často se na jejich výstavbě podílí národní obrana.

[1] Přeloženo z anglického originálu: “Cybercriminals are developing and boosting their attacks at an alarming pace, exploiting the fear and uncertainty caused by the unstable social and economic situation created by COVID-19.”

V bakalářské práci bude čtenář seznámen s rozdělením bezpečnostních cvičení, používanými typy scénářů a účastnickými rolemi. Další část je zaměřena na používané cyber range platformy. Vybráno je 5 druhů platforem tak, aby byly ukázány různé přístupy k vytvoření cyber range.

Dále je představena open-source cloudová platforma OpenStack a možnosti nasazení této platformy podle jejího účelu a nabízených služeb.

Poslední teoretická část rozebírá projekt KYPO (Kybernetický polygon), který je možné nasadit na vhodně přizpůsobenou instalaci OpenStacku.

Následuje úplný popis způsobu řešení práce - výběr metody nasazení OpenStacku, vlastní nasazení Openstacku, nasazení platformy KYPO a zprovoznění ukázkového scénáře pro demonstraci funkčnosti.

1 Bezpečnostní cvičení

Bezpečnostní cvičení představuje nenahraditelný pohled na zabezpečení konkrétní infrastruktury proti možným kybernetickým útokům. Testování probíhá v uzavřeném prostředí, kde nehrozí riziko ovlivnění reálné infrastruktury, cvičení probíhá bez porušení legislativních předpisů. Výsledek takového cvičení může odhalit slabiny v aktuální implementaci zabezpečovacích systémů a umožnit jejich zesílení.

1.1 Typy bezpečnostních cvičení

Bezpečnostní cvičení simulují rozličné krizové situace, na jejichž řešení se podílejí specialisté z více oborů. Cvičení probíhají na mezinárodní úrovni a dosahují i strategického měřítka. NÚKIB (Národní úřad pro kybernetickou a informační bezpečnost) typizuje bezpečnostní cvičení následovně:

- technická cvičení,
- table-top cvičení,
- procesní cvičení,
- komunikační cvičení,
- hybridní cvičení.

Technická cvičení se odvíjejí od individuálně připravených cyber ranges, ve kterých je možné simulovat reálně vedené útoky a umožňují odborníkům se připravit na detekci a obranu proti útokům ve skutečné infrastruktuře.

Table-top cvičení jsou netechnickým typem cvičení určených zejména pro střední a vyšší management, který primárně analyzuje předložený krizový scénář a vytváří plán na jeho efektivní splnění.

V **procesních cvičeních** je důraz kladen na danou organizaci jako celek při reakci na kybernetický útok. Jedná se o komplexní cvičení náročné na plánování a exekuci.

Komunikační cvičení mají za cíl ověřit dostupnost komunikačních kanálů v krizové situaci, kdy je klíčová bezpečná a rychlá komunikace mezi státními a soukromými subjekty při snížené použitelnosti komunikačních zdrojů a sítí.

Hybridní cvičení kombinuje výše zmíněné typy, jedná se o nejvíce komplexní typ cvičení, kde se prolínají odvětví, které se problematiky kybernetické bezpečnosti jakkoliv týkají (například právní a mediální). Tento typ cvičení pořádá ve spolupráci s Českou republikou Severoatlantická aliance NATO (North Atlantic Treaty Organization) jako součást programu Cooperative Cyber Defence Centre of Excellence [4, 5].

1.2 Cyber range

Cyber range je podle NIST (National Institute of Standards and Technology) definován jako realistická simulace systémů, aplikací a zařízeních v trénovacím prostředí. Výsledná simulace může být kombinací hardwaru a softwaru, případně skutečných a virtuálních prvků [3]. Prostředí cyber range nemusí sloužit pouze k testování odolnosti infrastruktury proti určitým typům útoků, ale i například pro vývoj nových technologií a výzkum.

1.3 Trénovací scénáře

V této podkapitole jsou představeny dvě nejčastější formy trénovacích scénářů bezpečnostních cvičení používaných v prostředí cyber rangů, popsán je jejich princip a obecné rozdělení účastníků do rolí.

1.3.1 Red team/Blue team

V tomto typu trénovacího scénáře se účastníci rozdělí do dvou týmů – red a blue, které představují soupeřící týmy. Red team znázorňuje útočící tým a snaží se získat přístup k chráněnému aktivu nacházejícímu se na zařízení blue týmu, případně získat přihlašovací údaje k privilegovanému účtu apod. Cílem blue týmu je zachytit a monitorovat podezřelou síťovou komunikaci a použít vhodné metody ke zmatení, zpomalení, případně k úplnému odvrácení hrozby útočícího týmu. Chráněné aktivum je předem určeno a známo oběma týmům. Scénář se většinou skládá z přípravné fáze, kde se každý z týmů seznámí se systémem a samotným ovládáním cyber range, připraví své zařízení a nástroje, které budou využívat k útoku respektive obraně, a z fáze exekuční. Scénář má stanovená pravidla, která se již odvíjejí od individuálnosti každého cvičení. Red team může být zaměřen na penetrační testování, sociální inženýrství, odhalení slabin (tzv. exploitů), black-box testování a podobně. Cílem defenzivního týmu je primárně zajištění bezpečnosti celé infrastruktury, včasné odhalení bezpečnostního incidentu a digitální forenzní analýza [7].

Trénovací scénář red team/blue team je také nazýván kybernetickou bojovou hrou, jelikož jsou často účastníci motivováni přirozenou rivalitou. Podrobnější popis poznatků tohoto typu trénovacího scénáře v literatuře [8].

1.3.2 Capture the Flag (CTF)

CTF je souhrn bodovaných úkolů s rostoucí obtížností, jedná se spíše o individuální trénovací scénář, ale úkoly mohou vyžadovat i skupinovou spolupráci. Každý úkol se skládá z popisu úkolu, bodového hodnocení a případně přiloženého souboru, který je nutný k jeho splnění. Účastníci cvičení úkoly postupně plní a získávají body, v nouzi mohou využít nápovědy, za kterou část získaných bodů zpravidla ztratí. Splnění úkolu sestává z objevení

takzvané "flagy", kterou účastníci doplňují do předem připravených polí ke kontrole správnosti. CTF soutěže často slouží k procvičení dovedností v časovém limitu. Úkoly jsou zaměřeny na různá odvětví kryptologie – kryptoanalýzu, steganografii, luštění šifer, lámání hashů a jiné. Výhodou CTF cvičení je jejich volitelná obtížnost, trénovat své dovednosti mohou začátečníci i odborníci [9].

1.4 Účastnické role

Bezpečnostní cvičení typu Red team/Blue team se zpravidla skládá z více účastnických skupin, než z pouze popsaných dvou v podkapitole 1.3.1. Následující rozdělení je nejvíce používáno vzhledem k aktuálnímu pojetí bezpečnostních cvičení:

- Red team,
- Blue team,
- White team,
- Green team,
- Yellow team,
- Purple team.

Red team představuje útočníky, kteří odhalují slabiny blue teamu. V rámci efektivity cvičení red team musí dodržovat určitá pravidla, nesmí například kompromitovat servisní infrastrukturu a musí se držet předem stanoveného scénáře.

Blue team tvoří účastníci, kteří jsou odpovědní za zabezpečení zranitelných míst sítě a monitorování podezřelé komunikace, obdobně jako red team musí dodržovat stanovená pravidla a lokální etický kodex.

White team je role skládající se z organizátorů a instruktorů příslušného cvičení, hlavním úkolem této skupiny je dohled na dodržování pravidel, případně poskytnutí rady a průběžné analyzování získaných dat během cvičení.

Green team je skupina zodpovědná za přípravu pracoviště – konfigurace virtuálních strojů, sítě a v případě poruchy zabezpečí jejich opravu.

Yellow team obdobně jako red team, ohrožuje bezpečnost infrastruktury, jedná se o nepřímo oddělenou skupinu, která instaluje podezřelý software nebo stahuje infikované soubory. Tento tým spolupracuje s organizátory, aby podpořili průběh scénáře.

Purple team je spojení red a blue týmů, tyto dva týmy pracují společně a zvyšují efektivnost celého cvičení poskytnutím zpětné vazby na aktuálně provedené úkony.

V některých cvičících scénářích se uvádí i grey team, který generuje normální síťový provoz, může se jednat o automatizovaný skript nebo reálné osoby. Motivace zahrnutí tohoto týmu do cvičení je zjistit, jak případné útoky a jejich šetření omezuje normální provoz sítě nebo služby [6, 10].

2 Cyber range platformy

V této kapitole jsou představeny platformy, které umožňují tvorbu cyber range. Platformy byly vybrány takovým způsobem, aby byly představeny různé způsoby vytvoření cyber range, podporující alespoň jeden z výše uvedených trénovacích scénářů, a které nejsou v čase psaní práce zastaralé.

2.1 SecDevOps@Cuse CyberRange

Projekt CyberRange se označuje jako první open-source *blueprint* pro bezpečnostní cvičení na světě. Projekt byl zveřejněn v září roku 2019, dodatečné verze nejsou, ale v době psaní práce jsou přidávány další zranitelné instance, poslední aktualizace proběhla v srpnu roku 2020. Projekt obsahuje soubor zranitelných systémů a volně šiřitelných nástrojů pro penetrační testování, analýzu malware, forenzní a reverzní inženýrství a další. Projekt zahrnuje oslabené operační systémy, aplikace a souhrn open-source nástrojů pro jejich detekci nebo zneužití slabin. Projekt je k dispozici na platformě GitHub společně s průvodcem instalací [11].

Značná část nástrojů pro detekci tvoří projekt Detection Lab vydaný v roce 2017, detailnější popis projektu v literatuře [20], celý projekt je přístupný na platformě GitHub [19].

2.1.1 Amazon Web Services (AWS)

Projekt zajišťuje automatizované vytváření scénářů na cloudu AWS. Pro přístup k AMI (Amazon Machine Images) je nutná registrace na cloudové platformě AWS, k funkčnosti stačí trial účet, nicméně některé služby jsou zpoplatněné způsobem *pay as you go*, zpoplatnění závisí zejména na počtu instancí a zdrojům jím přiděleným.

AWS uživatelům poskytuje neustálý přístup k uloženým datům, výpočetním zdrojům a více než 175 nástrojům pro síťové služby, pro vývojáře aplikací, analýzu dat a dalších. Kompletní seznam služeb a nástrojů v literatuře [12].

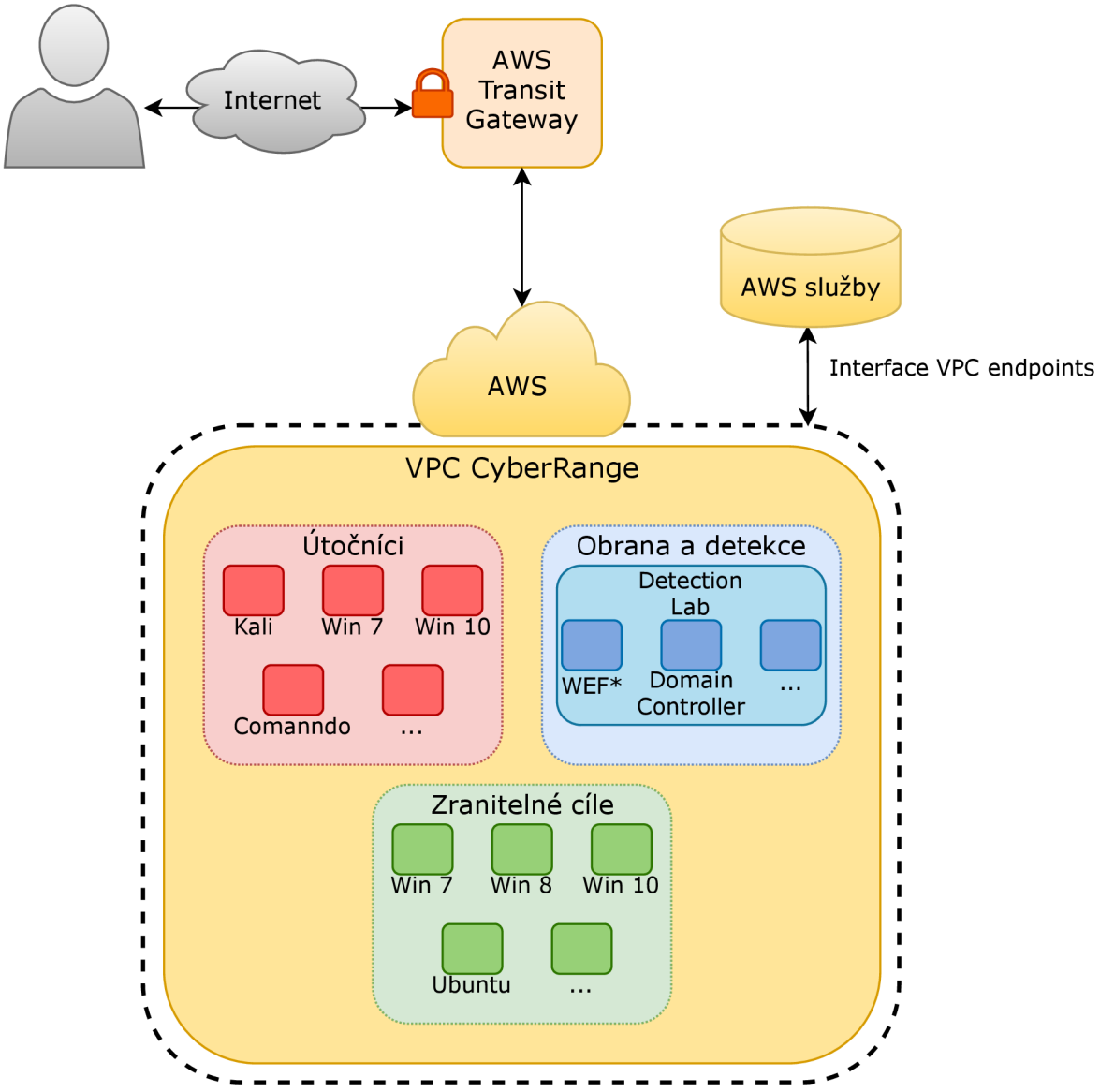
Služby podporující vývoj cyber range platformy jsou zejména tyto:

- Amazon Virtual Private Cloud (VPC),
- AWS Transit Gateway,
- Interface VPC endpoints.

VPC umožňuje vytvořit logicky oddělené sekce AWS cloudu a sestavit mezi nimi virtuální sítě. AWS Transit Gateway je služba, pomocí které se více VPC jednotek, případně výstupní body vlastní sítě, připojí na jedinou výstupní bránu. Interface VPC endpoints umožňuje bezpečně připojit konkrétní VPC jednotky k AWS službám, například službu

Amazon EC2, která poskytuje přerozdělování výpočetních zdrojů mezi VPC jednotkami [13, 18].

Obrázek 2.1 znázorňuje zjednodušenou architekturu SecDevOps@Cuse cyber range s použitím služeb AWS.



Obr. 2.1: Zjednodušená architektura SecDevOps@Cuse CyberRange, převzato z [11]

Shrnutí informací a požadavků

Následující tabulka 2.1 shrnuje informace o projektu.

Tab. 2.1: Shrnutí projektu SecDevOps@Cuse CyberRange [11]

Hardwarové požadavky	Žádné, platforma je přístupná přes AWS cloud
Softwarové požadavky	Terraform, SSH, Git, AWS Command Line Interface
Poslední aktualizace projektu	16.8.2020
Přístupnost	Registrace na AWS, cena se odvíjí od množství instancí a přidělených výpočetních zdrojů a použitých doplňujících služeb v rámci AWS
Dostupné scénáře	Nejsou
Vytvoření vlastního scénáře	Částečné lze z výběru 30 AMI, další jsou doplňovány
Dokumentace	Omezená na GitHubu, dostupné jsou video ukázky

2.2 CyTrONE

CyTrONE, celým názvem Cybersecurity Training and Operation Network Enviroment, je framework vyvinutý japonskou společností Japan Advanced Institute of Science and Technology. Framework umožňuje automaticky generovat trénovací scénáře pomocí sdružení několika open-source nástrojů. CyTrONE projekt následuje všechna doporučení NIST pro vývoj prostředí pro bezpečnostní cvičení[2]. První verze byla na platformě GitHub vydána v květnu roku 2017 a poslední (aktuální) verze byla vydána v červenci 2019, celý framework je open-source [15].

Architektura frameworku je složena z těchto hlavních modulů:

- GUI (Grafické uživatelské rozhraní),
- Trénovací databáze,
- Modul trénovací management,
- Modul CyRIS,
- Modul CyLMS.

GUI umožňuje jednoduchou interakci mezi uživatelem a nastavením cyber range. GUI organizátora postupně provede nastavením konkrétního trénovacího scénáře – o jaký typ scénáře se jedná, na co má být zaměřen a jeho obtížností. GUI umožňuje i konfigurovat konkrétní stanice – nastavovat jejich parametry jako například hostname, IP adresu, specifikace mezilehlých bodů virtuální sítě a další. GUI také vizualizuje celý cyber range, tedy uživatel má představu o počtu stanic a zda jsou aktivní (barevné odlišení). Snímek

[2]Dostupné zde: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf`

GUI aktivního cyber range je možné vidět v literatuře [14] na straně 47. Pokud uživatel nemůže grafické rozhraní použít, je možné celý framework upravit přes UI (User Interface) a CLI (Command Line Interface).

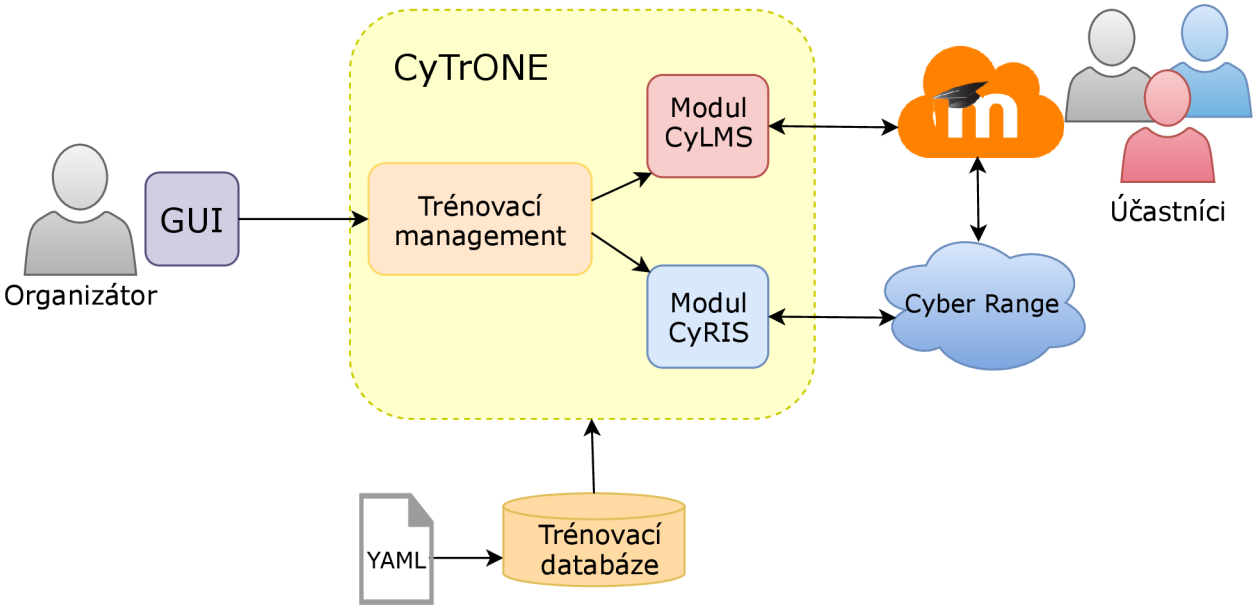
Trénovací databáze obsahuje veškerý popis trénovacích scénářů společně s požadavky na jejich zprovoznění. K charakteristice trénovacího scénáře se používá textově založený formát YAML (YAML Ain't Markup Language), jeho úprava je tedy možná přes textový editor.

Modul trénovací management slouží pro koordinaci celého frameworku, zprostředkovává veškerou komunikaci mezi GUI a frameworkem a zároveň komunikaci mezi frameworkem a dalšími moduly. Implementace umožňuje běh více vláken, a tedy součinnost více uživatelů.

Modul CyRIS je odpovědný za správu cyber range – vytvoření virtuálních strojů, nainstalování scénáře a konfiguraci entit cyber range (hostname, IP adresa, síťové připojení aj.), které jsou specifikovány v trénovací databázi v YAML souboru, a za samotné spuštění cyber range.

Modul CyLMS slouží k prezentaci cvičení účastníkům, je založen na systému LSM (Learning Management System). Pomocí Moodle systému účastníci s cyber rangem interagují, zadávají odpovědi, mohou si případně zobrazit nápovědu a systém automaticky udělí body. CyLMS modul pracuje se statistickými daty jako správnost mezikroků, délka trvání cvičení, počet použitých nápověd aj., vyhodnocení se potom účastníkům zobrazí v systému Moodle [14, 16].

Zjednodušenou architekturu frameworku CyTrONE znázorňuje obrázek 2.2.



Obr. 2.2: Architektura CyTrONE, převzato z [14]

Shrnutí informací a požadavků

Na stroji musí být operační systém Ubuntu nebo CentOS s balíčky Python alespoň verze 2.7, knihovna PyYAML, knihovna PassLib a nainstalovaný systém Moodle [16].

Následující tabulka 2.2 shrnuje informace o projektu.

Tab. 2.2: Shrnutí projektu CyTrONE [14, 16]

Hardwarové požadavky	Dvoujádrové CPU, 8GB RAM
Softwarové požadavky	Moodle, Python 2.7 a vyšší, knihovny PyYAML a PassLib, operační systémy Ubuntu nebo CentOS
Poslední aktualizace projektu	24.7.2019
Přístupnost	Open-source
Dostupné scénáře	Jeden CTF scénář a jeden kvíz inspirovaný NIST Technical Guide to Information Security Testing and Assessment, dostupné z [17]
Vytvoření vlastního scénáře	Uživatel si pro vlastní potřebu upraví předdefinované scénáře pro prostředí Ubuntu 16, CentOS 7 nebo Windows 7
Dokumentace	Obšírná

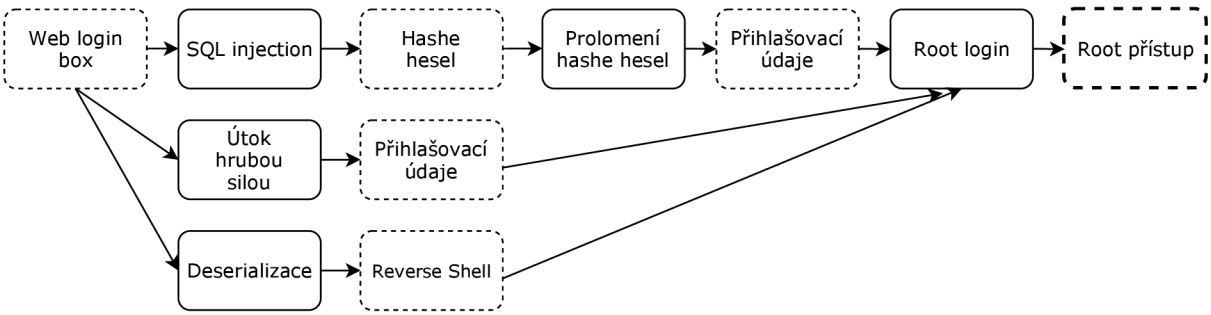
2.3 Alpaca

Cyber range Alpaca byl vydán jako open-source projekt roku 2019, naposledy byl aktualizován v dubnu roku 2019. Projekt Alpaca sám sebe zařazuje mezi dynamické cyber range, který podle uživatelem předem nastavených omezení vygeneruje trénovací prostředí, kde je možné zadaný úkol splnit různými způsoby. Tento systém umožňuje trénujícím bezpečnostní slabinu využít vlastními metodami. Systém Alpaca lze rozdělit do těchto logických částí:

- databáze zranitelností,
- plánovací stroj,
- generování cyber range.

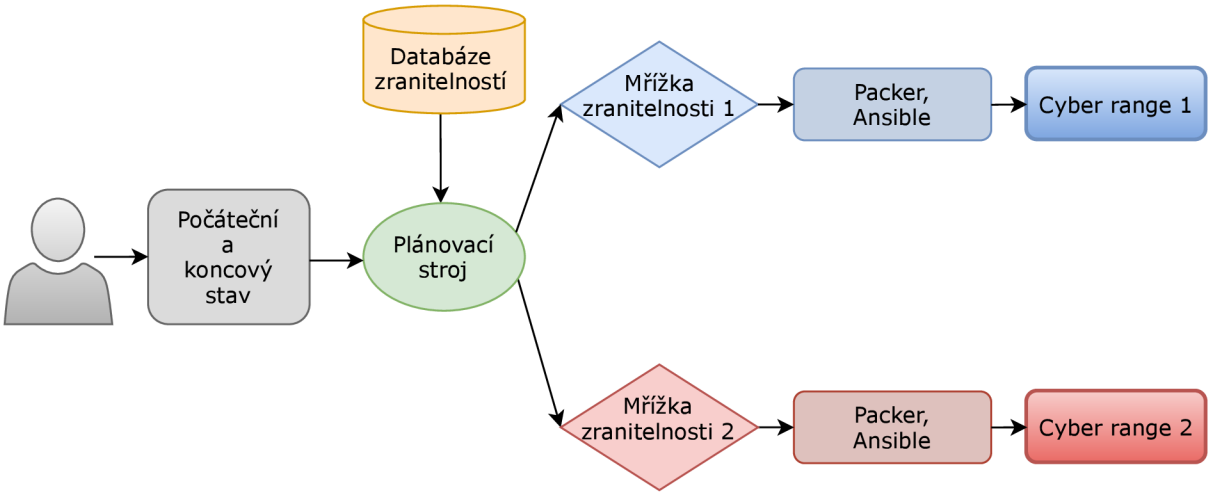
Databáze zranitelností obsahuje seznam zranitelností a k nim podmínkám, které musí být pro realizaci útoku na tuto zranitelnost splněny. Značným důvodem pro vedení databáze je možnost koexistence několika zranitelností zároveň, které vyžadují například dvě různé verze knihoven. Podle údajů v databázi zranitelností potom plánovací stroj nevytvoří pouze jednu instanci cyber range, ale instance dvě, které budou obsahovat rozdílné verze knihoven.

Plánovací stroj na základě počátečního stavu a koncového stavu nalezne všechny možné cesty, kterými lze konečného stavu dosáhnout. Generování možných cest může být upraveno v konfiguračních souborech plánovacího stroje. Plánovací stroj je implementován v jazyce Prolog a jeho úprava vyžaduje znalost tohoto jazyka. Tato část systému Alpaca také zajišťuje, že všechny způsoby řešení jsou kompatibilní, tedy není možné, že jeden způsob řešení potlačí jiný. Obrázek 2.3 znázorňuje tvorbu mřížky zranitelností[3], na obrázku je vidět, že lze úspěšně získat administrátorský přístup několika cestami. Plně ohraničené kolonky představují zranitelnosti, přerušované znázorňují stavy [21].



Obr. 2.3: Mřížka zranitelnosti, převzato z [21]

Všechny konfigurační soubory jsou předány nástrojům Packer a Ansible, pomocí kterých je vygenerován virtuální stroj s příslušným cyber rangem, který je spuštěn virtualizačním nástrojem VirtualBox. Postup vytvoření cyber range znázorňuje obrázek 2.4.



Obr. 2.4: Postup vytvoření cyber range pomocí systému Alpaca, převzato z [21]

[3]V anglické literatuře se uvádí fráze *vulnerability lattice*, překlad je zvolen jako mřížka zranitelností.

Shrnutí informací a požadavků

Následující tabulka 2.3 shrnuje informace o projektu.

Tab. 2.3: Shrnutí projektu Alpaca [21]

Hardwarové požadavky	Nejsou specifikovány
Softwarové požadavky	Nástroje SWI-Prolog, Graphviz, Packer a Ansible
Poslední aktualizace projektu	24.7.2019
Přístupnost	Open-source
Dostupné scénáře	Získání administrátorského přístupu pomocí 21 implementovaných zranitelností
Vytvoření vlastního scénáře	Není přesně specifikováno
Dokumentace	Dostačující, dostupný je pouze popis funkce systému a jeho částí

Nástroj SWI-Prolog je open source implementace jazyka Prolog. Nástroj Graphviz je svobodný software pro kreslení grafů zadaných ve formátu DOT (graph description language). Nástroj Packer slouží pro automatizované vytváření virtuálních strojů ve formátech EC2 (Elastic Compute Cloud), VMDK (VMware Virtual Disks), VMX (Virtual Machine Configuration) nebo OVF (Open Virtualization Format). Ansible je nástroj umožňující automatizovat správu softwaru.

2.4 Security Scenario Generator (SecGen)

SecGen je open source software dostupný na platformě GitHub, který vytváří komplexní virtuální stroje obsahující náhodné scénáře. V souboru XML (eXtensible Markup Language) se určí podrobnosti scénáře jako název zranitelnosti, CVE označení, uživatelé, obtížnost, generování nápověd apod. SecGen na základně této konfigurace náhodně vygeneruje scénář a pomocí nástrojů Puppet a Vagrant, který pomocí virtualizace vytvoří příslušné virtuální stroje spustitelné ve VirtualBoxu.

SecGen je navrhnutý tak, aby byl jednoduše rozšiřitelný pomocí modulů. Každý modul definuje část konfigurace celého scénáře – nastavení operačního systému, zranitelnosti, použitých služeb, konfigurace síťových rozhraní a konfigurace softwaru třetích stran. Moduly mohou být uživatelem přidávány a odebírány [22]. Projekt je přístupný na GitHubu [23].

Shrnutí informací a požadavků

Následující tabulka 2.4 shrnuje informace o projektu.

Tab. 2.4: Shrnutí projektu SecGen [22, 23]

Hardwarové požadavky	Nejsou specifikovány
Softwarové požadavky	Doporučen je operační systém Ubuntu, ale je možné použít i Windows a MacOS, OS (operační systém) musí disponovat jazykem Ruby a nástroji Vagrant, Puppet, Packer, ImageMagick, VirtualBox
Poslední aktualizace projektu	rok 2020, projekt je pravidelně aktualizován
Přístupnost	Open-source
Dostupné scénáře	Jsou dostupné předdefinované scénáře
Vytvoření vlastního scénáře	Jsou generovány ze šablon, které si uživatel vhodně upraví
Dokumentace	Dostačující pro instalaci i tvorbu scénářů

2.5 CTFd

CTFd je framwork pro vytváření cvičení typu Capture the Flag. Framework je velmi jednoduchý a nenabízí vetší konfiguraci kromě přizpůsobení GUI a tvorbu vlastních úkolů. CTFd je zejména soutěžní platforma, která sleduje postup jednotlivců i skupin pomocí bodového hodnocení, které ke každému úkolu může přiřadit organizátor. K využívání frameworku je potřebný software Docker, který slouží k izolaci aplikací (tzv. odlehčená virtualizace). Soutěžící k frameworku přistupují přes webovou stránku. Pokud organizátor nemá k dispozici webový server, je možné framework hostovat na cloudovém úložišti, tato služba je placená. Účastníci cvičení se k přístupu do CTFd musí registrovat, k úkolům se potom mohou zpětně vrátit. Framework je přístupný na GitHubu [25].

Shrnutí informací a požadavků

Následující tabulka 2.5 shrnuje informace o projektu [24].

Tab. 2.5: Shrnutí projektu CTFd [24]

Hardwarové požadavky	Nejsou specifikovány
Softwarové požadavky	Nástroj Docker
Poslední aktualizace projektu	22.9.2020
Přístupnost	Open-source
Dostupné scénáře	Je k dispozici demo scénář
Vytvoření vlastního scénáře	Jednoduchá tvorba vlastních úkolů, jiné přizpůsobení systém nepodporuje
Dokumentace	Dostačující vzhledem k složitosti frameworku

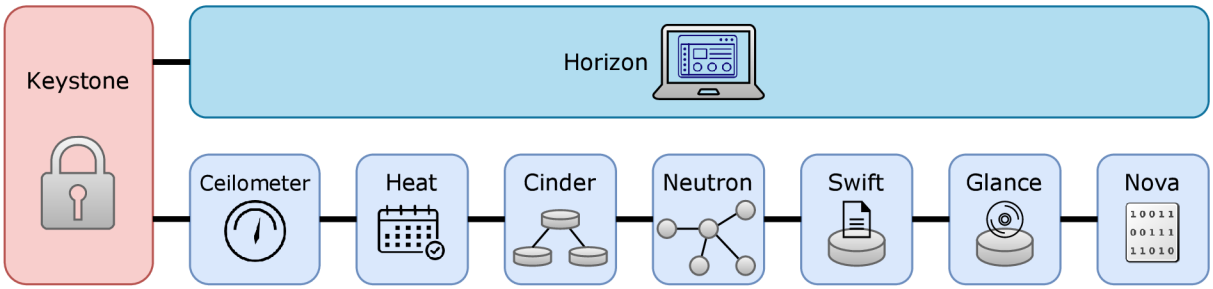
3 OpenStack

OpenStack je soubor volně dostupných nástrojů, které umožňují vytvořit cloudovou platformu nabízející výpočetní a paměťové služby pomocí virtualizace. První verze byla vydána v říjnu roku 2010. Projekt vznikl na podnět společností Rackspace Technology a NASA (National Aeronautics and Space Administration). Zpočátku byl financován neziskovou organizací OpenStack Foundation. V dnešní době na vývoj OpenStacku přispívá více než 500 společností, mezi které patří i světoví giganti jako IBM, VMware, RedHat, Cisco, Dell a další [26].

OpenStack se řadí do distribučního modelu IaaS (Infrastructure as a Service), což je typ modelu, kdy služba uživatelům propůjčuje své výpočetní a paměťové prostředky, které uživatel může využít dle své potřeby – pro běh operačních systémů, webových serverů nebo pro vývoj aplikací, bez nutnosti fyzické správy těchto prostředků a přerozdělování zdrojů při změně užívání. Pokud uživatel potřebuje zvýšit výpočetní nebo paměťové nároky, OpenStack dynamicky tyto prostředky přidělí, uživatelům tímto například odpadá nutnost koupit další hardware pro zvýšení výpočetní a paměťové kapacity při rozrůstající se infrastruktuře.

3.1 Architektura

Architektura OpenStacku se skládá z mnoha propojených komponent, z nichž je devět klíčových, poskytujících konkrétní služby, jsou jimi Horizon, Keystone, Ceilometer, Heat, Cinder, Neutron, Swift, Glance a Nova [37]. Vztah komponent je na obrázku 3.1, pod ním následuje stručný popis každé z nich.



Obr. 3.1: Klíčové komponenty OpenStacku, převzato z [37]

Horizon je webové grafické rozhraní, které usnadňuje uživatelům přístup ke službám a jejím konfiguracím. Přes Horizon lze konfigurovat většinu služeb, které OpenStack nabízí bez nutnosti instalovat další software nebo nástroje kromě webového prohlížeče. V případě, že uživatel nechce nebo nemůže webové rozhraní používat, konfigurace je možná přes CLI.

Keystone je komponenta, která slouží pro autentizaci, autorizaci uživatelů a kontrolu servisní politiky.

Autentizace probíhá důkazem znalostí (uživatelské jméno a heslo), autentizace může být i externí, pomocí propojení s účty třetích stran.

Autorizace slouží ke kontrole, zda má uživatel oprávnění používat určitou službu, případně v jaké míře. Keystone také umožňuje udělovat role uživatelům, slučovat je do skupin a spravovat jejich kompetenci pro používání konkrétních služeb.

Servisní politika upravuje předpisy a oprávnění uživatelů tak, aby nedocházelo k porušování legislativních předpisů země nebo interním předpisům společnosti, ve které je OpenStack používán.

Glance poskytuje službu pro vytváření, ukládání a spouštění virtuálních diskových obrazů a metadat k nim přiřazených. Uživatel může nastavit virtuální jednotku podobným způsobem jako u lokálního softwaru pro správu virtuálních strojů jako například VMware a VirtualBox. Virtuální jednotky mohou být uloženy v lokálním souborovém systému, úložišti Swift nebo na externím cloudu jako například AWS.

Nova je komponenta zodpovědná za poskytnutí výpočetních zdrojů ke zpracování žádostí uživatele. Skládá se z několika běžících démonů na linuxovém serveru a výpočetní zdroje dynamicky přiděluje z množiny virtuálních strojů, které pracují nad fyzickými servery a přímo výpočetními jednotkami, uživatel má pod kontrolou, jaké zdroje chce konkrétní instanci přidělit a Nova jejich přidělení poskytne. Tato komponenta je zásadní pro poskytování téměř všech služeb.

Neutron je síťová komponenta poskytující službu NaaS (Networking as a Service), která je zodpovědná za spojení mezi uživatelem a konkrétní službou a mezi službami samotnými. Zároveň uživatelům poskytuje službu pro vytvoření vlastní síťové infrastruktury.

Swift slouží jako úložiště uživatelských dat jakéhokoliv formátu, která jsou uložena jako objektový soubor. Uživatel je schopen přes Horizon nebo CLI vytvořit kontejnery, do kterých se poté data vkládají. Swift podporuje ve spojení s komponentou Keystone autentizaci a autorizaci uživatelů, nastavení přístupových práv pro individuální kontejnery, sám o sobě poskytuje automatizované mazání dat, zálohování a archivaci, tvorbu dočasných URL (Uniform Resource Locator) odkazů pro dočasný přístup a jiné.

Cinder je úložiště dat, které data ukládá ve formě bloků s pevnou velikostí. Cinder nekomunikuje přímo s komponentou Nova, slouží zejména jako připojitelné úložiště.

Ceilometer slouží k monitorování uživatelem využitých služeb a celkovou dobu jejich používání. Na základě těchto údajů je potom možné poskytnuté služby fakturovat.

Heat slouží k orchestraci služeb OpenStacku a jejich integraci. Více informací o všech dostupných komponentách a jejich instalaci v literatuře [36].

3.2 Metody nasazení

Manuální instalace a konfigurace komponent je velmi časově náročná a pro uživatele, kteří se s OpenStackem nesetkali, obtížná. Existují nástroje, které usnadňují nasazení OpenStacku do vlastní infrastruktury. Důležité je, aby si uživatel uvědomil, pro jaký účel chce OpenStack využívat, a k tomu přizpůsobil výběr metody nasazení, který bude nejvíce vyhovovat jeho účelům. V následujících podkapitolách jsou popsány nástroje, které nasazení OpenStacku usnadňují, kompletní seznam nástrojů pro nasazení je uveden v literatuře [38].

3.2.1 DevStack

DevStack je soubor skriptů a nástrojů pro rychlé nasazení klíčových komponent uvedených v podkapitole 3.1. Zdrojové soubory pro instalaci jsou přístupné na platformě GitHub [29]. DevStack je typ instalace *all-in-one*, při instalaci se zároveň konfiguruje základní vztah mezi klíčovými komponentami. Tato distribuce je tedy vhodná pro uživatele, kteří s OpenStackem začínají, protože nemusejí porozumět všem komponentám, vztahu mezi nimi a ručně je konfigurovat. Nicméně je důležité porozumět skriptům, které konfiguraci automatizují a pozměňují síťové nastavení a případně skripty upravit. DevStack podporuje plugin, který umožňuje doinstalovat další komponenty OpenStacku [30].

Následující tabulka 3.1 shrnuje informace o metodě nasazení DevStack.

Tab. 3.1: Shrnutí metody nasazení DevStack [29, 30]

Hardwarové požadavky	Dvoujádrové CPU, 8GB RAM
Softwarové požadavky, nástroje třetích stran	Operační systémy Ubuntu, CentOS, Fedora nebo OpenSUSE, vždy aktuální verze nebo jedna nižší
Použití	Pouze pro testovací a vzdělávací účely
Update modelu a dodatečné přidání nových komponent	Manuální, pro konfiguraci je nutné systém zastavit
Automatizace instalace	Není
Náročnost instalace	All-in-one, nenáročná
Dostupná podpora vendora	Žádná

3.2.2 OpenStack-Ansible

OpenStack-Ansible je způsob instalace pomocí Ansible playbooks[4], které komponenty nasazují pomocí LXC (Linux Containers) kontejnerů[5], které zajišťují izolaci mezi komponentami a jejich službami. Instalace se skládá z pěti hlavních kroků:

- příprava zaváděcí stanice,
- příprava cílové stanice,
- konfigurace nástroje Ansible,
- spuštění Ansible-playbooks,
- ověření instalace.

Zaváděcí stanice disponuje Ansible nástrojem a řídí celou instalaci na cílovou stanici. Zaváděcí stanice musí mít připojení k internetu pro přístupu k repozitářům s instalačními soubory a zároveň musí být zprovozněn SSH přístup na GitLab repozitář projektu OpenStack-Ansible.

Cílová stanice je stanice, na kterou nástroj Ansible instaluje OpenStack služby a klíčové komponenty. Aby instalace byla umožněna, je nutné zprovoznit mezi zaváděcí a cílovou stanicí SSH spojení pomocí klíčového managementu. Cílová stanice není zpravidla pouze jedna, ale několik stanic rozdělených do skupin, kde každá obstarává jednu OpenStack službu. Rozdělení bývá zpravidla logické (nezáleží na fyzickém rozložení cílových stanic). Logického rozdělení je docíleno pomocí nástroje LVM (Logical Volume Manager), který jediné fyzické zařízení rozdělí do několika logických částí a každou z nich operační systém spravuje jako samostatnou fyzickou jednotku.

Konfigurační soubory nástroje Ansible musí být upraveny tak, aby odpovídaly nasazení na příslušné cílové stanice. Obecně si uživatel v tomto kroku musí uvědomit, k jakému účelu bude OpenStack využíván a podle toho upravit konfigurační soubory nástroje Ansible. Zejména se jedná o konfiguraci síťových rozhraní a virtuálních sítí, úpravu seznamu obsahující cílové stanice a služby, které na ně Ansible bude instalovat, správu hesel k příslušným službám a další.

Ansible-playbooks spustí automatizovaný proces, kde Ansible připravuje cílové stanice k vlastní instalaci OpenStack komponent a poskytujících služeb a následně je nainstaluje podle konfiguračních souborů vytvořených v předchozím kroku.

V posledním kroku se ověří funkčnost základních služeb. Zkouška funkčnosti probíhá pouze na cílové stanici. Ověřuje se zejména funkčnost základních příkazů a uživatelské prostředí Horizon přístupné přes webový prohlížeč.

Výhoda tohoto způsobu instalace je oddělení individuálních komponent v LXC kontejnerech, úprava nebo update jedné služby neovlivní běh služby jiné, což značně usnadňuje

[4]Ansible playbooks jsou soubory, kde uživatel konfiguruje nasazení softwaru a orchestraci složek, jsou napsány v jazyce YAML.

[5]LXC je virtualizace podporující běh několika linuxových operačních systémů s použitím jednoho kernelu.

správu. Veškerá správa probíhá přes nástroj Ansible, který se stará o vzájemné propojení jednotlivých komponent a služeb, které nabízejí [31].

Následující tabulka 3.2 shrnuje informace o metodě nasazení OpenStack-Ansible.

Tab. 3.2: Shrnutí metody nasazení OpenStack-Ansible [31]

Hardwarové požadavky	Liší se od způsobu použití a množství použitých komponent, více zde [39]
Softwarové požadavky, nástroje třetích stran	Debian, Ubuntu, CentOS, SSH, Network Time Protocol, Python min. 3.6, en_US.UTF-8
Použití	Produkční účely
Update modelu a dodatečné přidání nových komponent	Obojí je možné přes Ansible
Automatizace instalace	Ansible
Náročnost instalace	Náročnější, nutné je porozumění nástroje Ansible
Dostupná podpora	Není

3.2.3 OpenStack-Fuel

Fuel je open-source nástroj, který zjednodušuje a urychluje prvotní nasazení OpenStacku na zařízení a zjednodušuje jeho budoucí správu. Nasazení zajišťuje open-source cloudová distribuce Mirantis, která nabízí distribuci OpenStacku na obrazovém disku určeném pro virtualizaci na hypervizoru VMware vSphere.

Architektura nástroje Fuel se skládá ze dvou částí – Fuel Master Node a Fuel Slave Nodes. Fuel Master Node je server s předinstalovaným Fuel nástrojem, který zajišťuje konfiguraci, síťovou správu a PXE (Preboot Execution Environment) spuštění Fuel Slave Nodes. Fuel Slave Node je komponenta nabízející konkrétní OpenStack službu, každá Slave Node komponenta je řízena Master Slave Node. Tato architektura je vhodná pro dynamický růst prostředí bez nutnosti pozměnit již stávající infrastrukturu. Instalace probíhá pomocí Mirantis skriptů, které nainstalují nástroj Fuel do virtuálního prostředí VMare vSphere. Před instalací uživatel modifikuje konfigurační soubor, kde je specifikováno, kolik má být vytvořeno Slave Nodes, přiřazení výpočetních jednotek, paměti, uživatelské údaje a podobně. Po instalaci je přes jednotku Master Node zpřístupněno webové GUI, ve kterém je možné vytvořit vlastní OpenStack infrastrukturu. Hardwarové a softwerové požadavky jsou specifikovány v literatuře [32], požadavky se liší v závislosti na službách, jakými má infrastruktura disponovat.

Následující tabulka 3.3 shrnuje informace o metodě nasazení OpenStack-Fuel.

Tab. 3.3: Shrnutí metody nasazení OpenStack-Fuel [32]

Hardwarové požadavky	Čtyřjádrové CPU, 4GB RAM, 10Gbit síťový port, 20 GB volného místa pro Master Node, 20GB pro každou další Slave Node
Softwarové požadavky, nástroje třetích stran	64bit OS, VMware vSphere
Použití	Produkční účely
Update modelu a dodatečné přidání nových komponent	Obojí možné přes CLI.
Automatizace instalace	Není
Náročnost instalace	Náročnější
Dostupná podpora vendora	Mirantis

3.2.4 OpenStack-TripleO

TripleO znamená OpenStack on OpenStack, jedná se o způsob nasazení OpenStacku na fyzický hardware nebo virtuální stanici, kde řídící roli instalace a následného managementu zaujímá samotný OpenStack. Řídící část se nazývá UnderCloud a OverCloud se nazývá část, která může být nasazována jako produkční nebo testovací prostředí.

Instalace se skládá ze tří hlavních kroků:

- příprava zaváděcí stanice,
- příprava cílové stanice,
- konfigurace nástroje Ansible,
- spuštění Ansible-playbooks,
- ověření instalace.

Pro tento způsob nasazení OpenStacku je potřeba mít 3 fyzické nebo virtuální stanice s operačními systémy RHEL 7 nebo CentOS 8. Na řídící stanici musí být vytvořen obyčejný uživatel patřící do sudo skupiny. UnderCloud je OpenStack s pevně vytvořenými a nakonfigurovanými komponentami zprostředkovávající služby pro nasazení do vlastní infrastruktury.

UnderCloud se nainstaluje z dostupných repozitářů Epel a Dealorean Deps, přístup k těmto repozitářům je třeba na stanici povolit. Instalace UnderCloudu probíhá přes automatický skript, na konci instalačního procesu je vytvořena virtuální jednotka přístupná přes SSH, ve vytvořených konfiguračních souborech se nacházejí hesla pro jednotlivé služby UnderCloudu.

Instalace OverCloudu probíhá pomocí UnderCloudu úpravou konfiguračních souborů, před instalací je nutné si uvědomit, která ze dvou stanic určených pro OverCloud bude

sloužit jako výpočetní jednotka, a která jako správce služeb. Podrobnější průvodce instalací v literatuře [33].

Hardwarové požadavky se liší podle toho, zda uživatel použije nasazení na virtuální stanici nebo na fyzické zařízení.

Následující tabulka 3.4 shrnuje informace o metodě nasazení OpenStack-TripleO.

Tab. 3.4: Shrnutí metody nasazení OpenStack-TripleO [33]

Hardwarové požadavky	Liší se v závislosti na nasazení na fyzický hardware nebo pomocí virtualizace, podrobně v instalačním manuálu
Softwarové požadavky, nástroje třetích stran	RHEL x84_64, CentOS x84_64
Použití	Produkční i testovací účely
Update modelu a dodatečné přidání nových komponent	Pouze celé nasazení
Automatizace instalace	Částečná
Náročnost instalace	Náročná
Dostupná podpora vendora	RedHat

3.2.5 OpenStack-RDO Packstack

PackStack je nástroj sloužící pro *all-in-one* instalaci OpenStacku na jediné fyzické nebo virtuální zařízení. K instalaci PackStacku je využit nástroj Puppet[6], který automaticky nakonfiguruje klíčové OpenStack komponenty, takže je uživatel nemusí konfigurovat ručně při instalaci. Z tohoto důvodu je PackStack vhodný pro obecné použití OpenStack služeb, protože umožňuje pouze základní konfiguraci komponent. Pro nasazení do již vytvořené infrastruktury, případně pro speciální účely, je doporučeno použít jiný způsob nasazení, který umožňuje komponenty nakonfigurovat vyhovujícím způsobem pro vlastní účely [34].

PackStack umožňuje nainstalovat aktuálně udržované verze OpenStacku, v čase psaní práce jsou to verze Stein, Train a Ussuri[7], podrobný seznam verzí a jejich aktualizací je k dispozici v literatuře [35].

Během instalace je vytvářen textový soubor `packstack-answers`, kde se nachází všechna použitá konfigurace použitá při instalaci. Pokud je instalace úspěšně dokončena, je vytvořen soubor `keystonerc_admin`, kde se nacházejí přihlašovací údaje do služby Horizon a IP adresa, na které je možné se do služby přihlásit.

[6]Podobně jako Ansible, nástroj Puppet slouží k orchestraci a automatizaci softwaru.

[7]Nejnovější verze Victoria není v dokumentaci zmíněna, proto zde není uvedena

Následující tabulka 3.5 shrnuje informace o metodě nasazení OpenStack RDO PackStack.

Tab. 3.5: Shrnutí metody nasazení OpenStack RDO Packstack [34]

Hardwarové požadavky	Doporučeno je 16GB RAM
Softwarové požadavky, nástroje třetích stran	RHEL x84_64 a jiné RHEL linuxové distribuce, CentOS 7 nebo CentOS 8 podle požadované verze OpenStacku
Použití	Pouze testovací a vzdělávací účely
Update modelu a dodatečné přidání nových komponent	Pouze celé nasazení pomocí CLI
Automatizace instalace	Puppet
Náročnost instalace	All-in-one, nenáročná
Dostupná podpora vendora	RedHat

4 Analýza projektu Kybernetický polygon

Kybernetický polygon (KYPO) je projekt, který vznikal od roku 2013 v prostředí CERIT Science Part Masarykovy univerzity ve spolupráci s Ministerstvem vnitra České republiky, jeho cílem je vytvořit kontrolované, monitorované, virtualizované a škálovatelné prostředí přizpůsobené pro bezpečnostní cvičení a výzkum v oblasti kybernetické bezpečnosti. Po dokončení svého vývoje v roce 2015 se KYPO projekt stal unikátně technicky a programově vybaveným prostředím v České republice [40].

V následující podkapitole jsou popsány klíčové části architektury.

4.1 Architektura

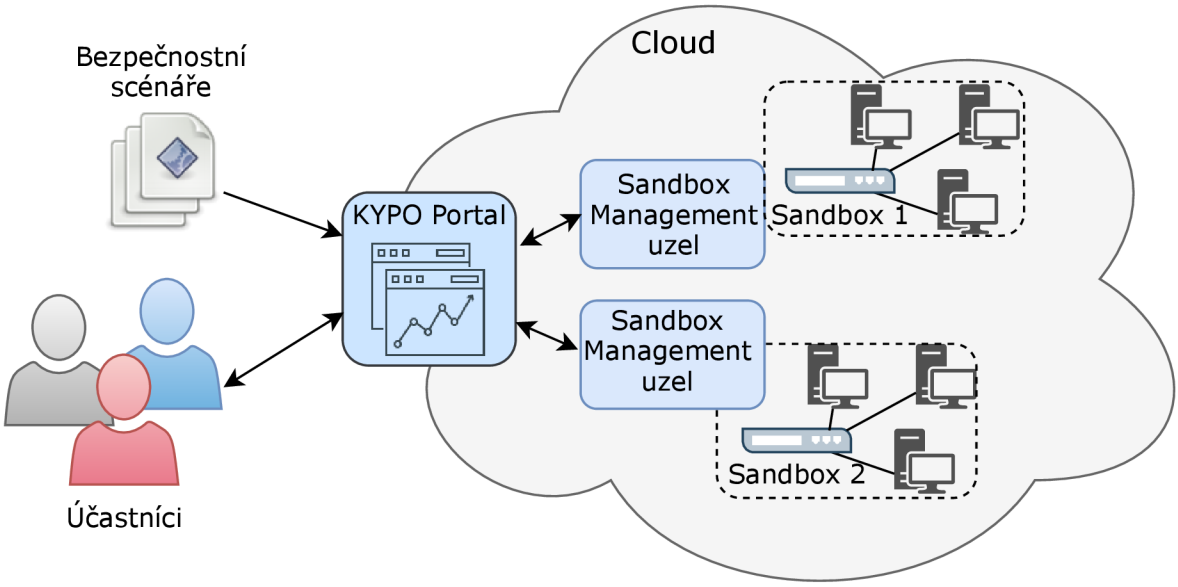
Architektura KYPO cyber range je založena na následování těchto podmínek:

- flexibilita,
- škálovatelnost,
- nákladově efektivní,
- isolace,
- monitorování,
- jednoduchý přístup,
- open-source.

Splnění těchto podmínek je zajištěno sestavením KYPO projektu v cloudovém prostředí. Modularita KYPO projektu zajišťuje, že je možné projekt nasadit na různé cloudové platformy, jakými jsou OpenStack nebo OpenNebula[8].

Vysokoúrovňová architektura zjednodušeného cyber range se dvěma sandboxy je znázorněna na obrázku 4.1. Pod ním následuje stručný popis jednotlivých částí [42].

[8]KYPO projekt byl zprvu vyvíjen na platformě OpenNebula, od roku 2017 byla vyvíjena adaptace na OpenStack cloud.



Obr. 4.1: Architektura KYPO cyber range se dvěma sandboxy, převzato z [41]

Sandbox je izolovaná část cyber range skládající se z množiny virtuálních strojů, virtuální síťové topologie a konfigurovatelných zařízení. Sandboxy poskytují uživatelům klíčové funkce k uskutečnění požadovaného experimentu.

Sandbox management uzel je komponenta zodpovědná za vytvoření a ovládání jednotlivých sandboxů. Při vytváření sandboxu tato komponenta koordinuje veškerou síťovou infrastrukturu sandboxu pomocí pokročilého ovladače, který poskytuje nezávislost na skutečné síťové a spojové vrstvě, a tím umožňuje uživatelům dynamicky vytvořit svou vlastní síťovou infrastrukturu s libovolným adresním prostorem. Pro udržení konektivity se zařízeními, které nejsou virtualizované, bylo vytvořeno zařízení založené na Raspberry PI platformě, které toto spojení umožňuje pomocí automatického zprovoznění VPN (Virtual Private Network) tunelu do jednotlivých sandboxů cyber range, ihned po startu celého systému.

KYPO Portal je webové GUI, přes které účastníci interagují s vytvořenými sandboxy. GUI je intuitivní a kromě samotného ovládání cyber range zprostředkovává účastníkům přístup ke zdrojům cyber range, dostupným službám a nástrojům, nebo přehled síťové topologie, vykreslení grafů a další. Portál se může lišit vzhledem k roli, jakou účastník má. Rozdílné GUI může mít příslušník blue, green týmů nebo white týmu, který sleduje celkový rozvoj konkrétního scénáře a analyzuje jej.

Účastníci k samotným sandboxům přistupují přes připravené webové rozhraní KYPO portálu. Každý scénář má vlastní předdefinované uživatelské prostředí, které obsahuje nezbytné prvky k obsloužení sandboxu s příslušným scénářem. Účastníci mohou k ce-

lému cyber rangi přistoupit i vzdáleně bez nutnosti instalovat webové rozšíření jako Java nebo Flash.

Bezpečnostní scénáře reprezentují množství konfiguračních souborů, které definují samotné cvičení. Jedná se zejména o soubory ve formátu JSON (JavaScript Object Notation), kde se nacházejí nezbytné podrobnosti o účastnických rolích, cílech specifického týmu, instrukcích k samotnému scénáři a konfiguraci celé struktury sandboxu společně se všemi stanicemi, síťovými prvky a dalšími entitami, které má sandbox obsahovat [42].

4.2 Shrnutí projektu KYPO

Prostředí Kybernetického polygonu je unikátní prostředí sloužící k výzkumu nových bezpečnostních metod, potenciálních hrozeb a jejich dopadu na národní kybernetickou infrastrukturu. KYPO cyber range ve spolupráci s NÚKIB a Ministerstvem vnitra České republiky pravidelně pořádá bezpečnostní cvičení Cyber Czech, které svým zaměřením spadá do typu technických cvičení (podkapitola 1.1), které se soustředí na ochranu integrovaného záchranného systému. Cvičení probíhá na mezinárodní úrovni a jako aktivní účastníci se v něm podílí týmy ze zahraničí. V posledním uskutečněném cvičení, v roce 2018, se účastnily země Estonsko, Izrael, Chorvatsko a Jižní Korea. Cvičení se jako pozorovatelné účastnili odborníci ze zemí - Maďarsko, Nizozemí, Polsko, Velká Británie a Spojené státy americké.

KYPO cyber range je zpřístupněn studentům bezpečnostních oborů v rámci studia na Masarykově univerzitě i pro odbornou veřejnost v rámci školících programů [43]. Na podzim roku 2020 byl KYPO projekt zpřístupněn pod open-source licencí.

4.2.1 Související cyber range

Porovnání projektu KYPO s podobnými kybernetickými polygony na mezinárodní úrovni není jednoduché, protože podrobnosti o způsobu provedení cyber rangů, které umožňují vytvořit prostředí schopné provádět bezpečnostní cvičení v takovém měřítku, jako právě KYPO, jsou z velké části klasifikována, neboť slouží zejména k ochraně národní bezpečnosti a často využívají komunikační infrastrukturu národní obrany [40].

Pro porovnání se zmíněnými platformami z kapitoly 2 je možné porovnání pouze s platformou SeCDevOps@Cuse Cyber Range, jelikož je stavěna na cloudu AWS a následuje většinu zmíněných předpokladů pro stavbu cyber range (flexibilita, škálovatelnost, jednoduchý přístup atd.). Nicméně Cyber Range je dostupný pouze přes spojení s AWS službami na privátní cloud, který SeCDevOps@Cuse poskytuje. KYPO projekt umožňuje nasazení do vlastního prostředí bez nutnosti využívat například právě externí cloud, a tím dává prostor pro vytvoření de facto jakéhokoliv scénáře dle vlastní potřeby.

Pokud je srovnání zaměřeno na cyber range, které svým provedením dokáží hostit cvičení na podobné úrovni jako KYPO projekt, vhodné je zmínit Estonian Cyber Range, který poskytuje své služby pro pořádání dvou mezinárodních NATO cvičení Locked Shields a Cyber Coalition. Estonian Cyber Range je založen na využití hypervizoru VMware vSphere, ve kterém je simulována celá herní infrastruktura pro všechny typy týmů a účastníkům je zajištěn vzdálený přístup pomocí VPN [40].

Podrobný průzkum dostupných cyber rangů, které jsou využívány pro cvičení zaměřené na ochranu státních komunikačních infrastruktur je dostupný v literatuře [44]. Průzkum je publikován v roce 2013, tudíž seznam popsaných cyber rangů z pohledu dnešní doby nebude kompletní, nicméně jedná se o ojedinělý neklasifikovaný dokument, který nabízí podrobnější popis cyber rangů a základní principy jejich funkcionality.

5 Praktická část

Kapitola se zabývá rozborem úvah a celkovému popisu postupu pro úspěšné zprovoznění KYPO platformy, včetně instalace platformy OpenStack. Nasazení cloudové platformy OpenStack a nasazení platformy KYPO vychází zejména z literatury [47] a [45].

Jak bylo uvedeno v kapitole 4, platforma KYPO je postavena na platformu OpenStack, díky čemuž je dosaženo velkým možnostem využití cyber range pro různorodá bezpečnostní cvičení a v případě nutnosti jednoduchému rozšíření o výpočetní a paměťové prostředky.

Projekt KYPO k funkčnosti potřebuje několik klíčových OpenStack komponent - **Nova**, **Neutron** s podporou plovoucích adres[9], **Keystone**, **Horizon**, **Heat** a **Placement**. Služba Placement není v kapitole 3.1 zmíněna, ale pro zprovoznění KYPO platformy je nutná. Služba Placement slouží k mapování využitých zdrojů a dat sdílených mezi více službami a poskytuje jej podle potřeby [45]. Pro komunikaci mezi uživatelem a vytvořenými sandboxy (sandbox viz podkapitola 4) je využita tzv. Spice konzole, která je jednou ze služeb nabízených komponentou Nova a umožňuje plně interaktivní přístup pomocí protokolu SPICE (Simple Protocol for Independent Computing Environments).

5.1 Původní nasazení platformy OpenStack

Nasazení platformy OpenStack je komplexní záležitost a metody nasazení se mohou vzájemně velmi odlišovat podle toho, k jakému účelu bude OpenStack použit. I přesto, že veškeré metody nasazení OpenStacku mají stejný cíl, a to usnadnit instalaci a propojení vybraných komponent do plně funkční cloudové platformy, může nevhodně zvolená metoda nasazení ztížit, ne-li znemožnit případně potřebnou úpravu komponent a jejich služeb v pozdější době.

Pro metodu nasazení byla původně zvolena metoda nasazení pomocí RDO Packstack, která nasazení OpenStacku automatizuje pomocí tzv. Puppet modulů na CentOS 8 a obsahuje všechny potřebné komponenty, které jsou nutné pro KYPO. Nasazení je na konkrétní stanici přizpůsobeno pomocí konfiguračního souboru, tzv. `answer-file`, ve kterém se volí potřebné komponenty a zejména se konfiguruje komponenta Neutron, aby nasazení sedělo na konkrétní síťové rozhraní. Soubor je poté před instalací načten, a podle něho Puppet jednotlivé OpenStack komponenty instaluje a upravuje.

Nasazení typu Packstack ovšem nepodporuje instalaci komponenty Nova s podporou Spice konzole, ta by tedy musela být dodatečně nainstalována. CentOS 8 nenabízí jednotný balík pro podporu Spice konzole a pro CentOS 8 nejsou k dispozici nástroje, které by podporovaly použití Spice konzole, která je kompatibilní s komponentou Nova.

[9] V angličtině se jedná o název floating IP address, tyto adresy slouží pro přístup k instancím z externích sítí. Fungují na principu NAT (Network Address Translation).

Obecně použití metody nasazení Packstack není vhodné, pokud by bylo třeba komponenty upravit na již nasazeném OpenStacku, z toho důvodu, že Packstack neumožňuje konfigurovat komponenty jednotlivě. Pokud by bylo nutné provést nějakou změnu, instalace všech komponent by musela být provedena znovu pomocí upraveného konfiguračního souboru `answer-file`. Toto je nevhodné hlavně z toho důvodu, že projekt KYPO je stále velmi čerstvý. Jako open-source vyšel v poslední třetině roku 2020, a tedy se jeho požadavky na OpenStack mohou měnit. Samotný tým KYPO změnil své doporučení nasazovat KYPO platformu na CentOS a nyní doporučuje použít Ubuntu 20.04. Dalším důvodem je, že OS CentOS končí na konci roku 2021 podpora a bude uvedena nová distribuce CentOS Stream, která plánuje mít zrychlený vývoj, a tedy není tuto novou distribuci vhodné použít do produkčního prostředí [46].

Z těchto výše uvedených důvodů bylo od původního rozhodnutí použít nasazení na CentOS 8 s využitím metody Packstack opuštěno a bylo zvoleno nasazení na Ubuntu 20.04 pomocí metody OpenStack-Ansible, konkrétně Kolla-Ansible, viz literatura [47].

5.2 Konečné nasazení platformy OpenStack

Metoda nasazení Kolla-Ansible umožňuje OpenStack nasadit ve dvou módech, v tzv. módu *all-in-one* a módu *multinode*. V módu *all-in-one* jsou všechny prostředky a jednotlivé komponenty dostupné z jedné stanice. V módu *multinode* je nasazení rozděleno alespoň na dvě stanice nazývaných *controller* a *compute*. Výhoda nasazení v módu *multinode* je hlavně v jednoduchém přidání dalších výpočetních nebo paměťových prostředků, neboť v případě selhání výpočetní stanice nemusí být celé nasazení znehodnoceno.

Cyber range jsou obecně velmi dynamické a pro zajištění co nejkvalitnějších bezpečnostních cvičení pro libovolný počet hráčů je nutné výpočetní a paměťové prostředky, případně poskytované služby, dynamicky upravovat, proto je volba módu *multinode* z tohoto hlediska nejvhodnější.

Pro praktickou implementaci jsou vytvořeny 3 virtuální stanice, které jsou fyzicky umístěné na virtuálním stroji v laboratoři na Ústavu telekomunikací. Přístup k těmto stanicím je zajištěn pomocí VPN a vzdálené plochy s využitím protokolu xRDP (Remote Desktop Protocol). Parametry virtuálních stanic jsou obsaženy v tabulce 5.1.

Tab. 5.1: Parametry virtuálních stanic

Název stanice	Operační systém	Počet CPU	RAM	Disk	Síťové rozhraní	IP adresa
Kypo	Ubuntu 20.04	4	12,5GB	160GB	2	192.168.1.120
Controller	Ubuntu 20.04	4	8GB	100GB	2	192.168.1.121
Compute	Ubuntu 20.04	8	32GB	400GB	2	192.168.1.122

Na stanici Controller a Compute jsou přidány záznamy do souboru `/etc/hosts` pro překlad doménového jména *control* a *compute* s odpovídající IP adresou.

Všechny tři stanice mají přes druhé fyzické rozhraní přístup do sítě `172.24.0.0./24`, která bude sloužit pro přidělování plovoucích adres vytvořeným instancím.

Konfigurace a nasazení OpenStacku probíhá na stanici Controller. Před zahájením nasazení je nutné nainstalovat potřebné balíčky *python3*, rozhraní *libffi*, kompilátor *gcc* a knihovnu *libssl* pro podporu protokolů SSL a TLS. Instalaci je doporučeno provádět ve virtuálním prostředí pro zamezení konfliktu mezi systémovými balíčky a závislostmi, a těmi potřebných pro instalaci. Pro vytvoření virtuálního prostředí slouží nástroj *python3-venv*. Ve virtuálním prostředí se následně nainstaluje nástroj *pip* a samotný balík nástrojů a kontejnerů Kolla-Ansible. Instalaci závislostí a vytvoření virtuálního prostředí zobrazuje výpis 5.1.

Výpis 5.1: Instalace balíčků a vytvoření virtuálního prostředí

```
user@controller:~$ sudo apt-get update
user@controller:~$ sudo apt-get install python3-dev libffi-
   dev gcc libssl-dev python3-venv
user@controller:~$ python3 -m venv /home/user/ostack
user@controller:~$ source /home/user/ostack/bin/activate
(ostack)user@controller:~$ pip install -U pip
(ostack)user@controller:~$ pip install 'ansible<2.10'
(ostack)user@controller:~$ pip install kolla-ansible
```

Dále je potřeba zkopírovat soubory `globals.yml` a `passwords.yml` do vytvořeného adresáře `/etc/kolla`. Do aktuálního adresáře se také zkopíruje konfigurační soubor pro instalaci OpenStacku v módu *multinode*. Provedené kroky zobrazuje výpis 5.2.

Výpis 5.2: Příprava konfiguračních souborů Ansible a Kolla-Ansible

```
(ostack)user@controller:~$ sudo mkdir -p /etc/kolla
(ostack)user@controller:~$ cp -r /path/to/venv/share/kolla-
   ansible/etc_examples/kolla/ * /etc/kolla
(ostack)user@controller:~$ cp /path/to/venv/share/kolla-
   ansible/ansible/inventory/multinode.yml
```

Nyní se může přejít k úpravě souboru `multinode.yml`, kde se specifikují informace o stanicích, na které má být OpenStack nasazen. Místo IP adres jsou zadány doménová jména *control* a *compute*. Hvězdička ve výpisu značí heslo uživatele *user* na dané stanici. Hodnota parametru `ansible_become` vyjadřuje, zda je potřeba pro administrátorská práva použít příkaz `sudo`. Upravený soubor zobrazuje výpis 5.3

Výpis 5.3: Úprava souboru multinode

```
[control]
192.168.1.121
ansible_become_user=user ansible_password= * ansible_become=
   true
[network]
192.168.1.121
[compute]
192.168.1.122
ansible_become_user=user ansible_password= * ansible_become=
   true
[monitoring]
192.168.1.121
[storage]
192.168.1.122
[deployment]
localhost
ansible_connection=local become=true
```

Dostupnost stanic lze zkontrolovat pomocí příkazu `ansible -i multinode all -m ping`.

Před zahájením nasazení je nutné upravit konfigurační soubor `globals.yml` tak, aby nasazený OpenStack obsahoval všechny požadavky pro projekt KYPO a odpovídal konkrétní infrastruktuře. Konfigurační soubor je obsáhlý a důležité jej pročíst celý a upravit potřebné části, které by nevyhovovaly nasazení do naší infrastruktury. Upravené části souboru jsou zobrazeny ve výpisu 5.4.

Výpis 5.4: Úprava instalačního souboru globals.yml

```
kolla_base_distro: "ubuntu"
network_interface: "ens192"
neutron_external_interface: "ens160"
kolla_internal_vip_address: "192.168.1.123"
neutron_plugin_agent: "linuxbridge"
nova_console: "spice"
```

Parametr `kolla_base_distro` označuje linuxovou distribuci, kromě Ubuntu je možné OpenStack nainstalovat na distribuce CentOS, RHEL a Debian.

Parametr `network_interface` označuje fyzické síťové rozhraní pro provoz mezi jednotlivými stanicemi.

Parametr `neutron_external_interface` slouží pro vytváření externích (veřejných) sítí. Toto rozhraní by mělo být nakonfigurováno bez IP adresy, aby instance z vnitřní sítě

měly přístup do externích sítí a případně do internetu.

Parametr `kolla_internal_vip_address` označuje IP adresu dedikovanou pro správu mezi stanicemi Controller a Compute, případně dalšími. Tato IP adresa musí být staická a je nutné, aby pocházela z rozsahu, ke kterému je přiřazen parametr `network_interface`. Na tuto adresu se také budou připojovat zařízení vyžadující některou ze služeb OpenStacku.

Parametr `neutron_plugin_agent` identifikuje agenta pro síťovou správu fyzických a virtuálních rozhraní, přes které se vnitřní instance mohou připojit k externí síti a k internetu.

Parametr `nova_console` pouze označuje způsob ovládání vnitřních instancí. Projekt KYPO umožňuje použít pouze typ konzole Spice[10].

Po úpravě a uložení souboru `globals.yml` už stačí pouze spustit Ansible playbooky, viz výpis 5.5. Pokud některý úkol vrátí hodnotu `failed=1` je nutné ho vyřešit před spuštěním dalšího playbooku.

Výpis 5.5: Zahájení nasazení OpenStacku

```
(ostack)user@controller:~$ kolla-ansible -i ./multinode
   bootstrap-servers
(ostack)user@controller:~$ kolla-ansible -i ./multinode
   prechecks
(ostack)user@controller:~$ kolla-ansible -i ./multinode
   deploy
```

Pokud poslední playbook skončí s návratovou hodnotou `failed=0`, OpenStack je úspěšně nasazen a pro používání stačí pouze vygenerovat soubor `admin-openrc.sh`, kde jsou vypsány přístupové údaje a adresy služeb OpenStacku, toho se dosáhne příkazem `kolla-ansible post-deploy`. Pro používání OpenStack příkazů je nutné tento soubor připojit jako zdroj příkazem `source`.

5.2.1 Vyvtoření externí sítě

Pro potřeby platformy KYPO je vytvořena externí síť `public` s podsítí `172.24.0.0/24`, pomocí které je umožněno spojení s budoucími stanicemi KYPO cyber range. Vytvoření externí sítě je zobrazeno ve výpisu 5.6.

[10]Podle aktuální KYPO dokumentace je možné od května 2021 používat i typ konzole noVNC

Výpis 5.6: Vytvoření externí sítě public

```
user@controller:~$ network create --provider-network-type=
   flat --share --provider-physical-network physnet1 --
   external public
user@controller:~$ openstack subnet create --subnet-range
   172.24.0.0/24 --allocation-pool start=172.24.0.2,end
   =172.24.0.100 --gateway 172.24.0.1 --network public public
   -subnet
```

Parametr `physnet1` označuje spojení s fyzickým rozhraním `ens160`. Název se může lišit v závislosti na použitém OS a distribuci. Ten je možné zkontrolovat v konfiguraci komponenty Neutron v souboru s názvem `linuxbridge_agent.ini`. Rozhraní `physnet1` musí být připojeno k fyzickému rozhraní `ens160` tak, jak bylo zadáno v konfiguračním souboru `globals.yml`. Je vhodné stejný soubor zkontrolovat i na stanici Compute, obsah musí být totožný. Výchozí brána `172.24.0.1` umožňuje připojení k internetu.

5.3 Nasazení platformy KYPO

Nasazení platformy KYPO se skládá ze dvou částí:

- instalace základní infrastruktury,
- nasazení KYPO cyber range.

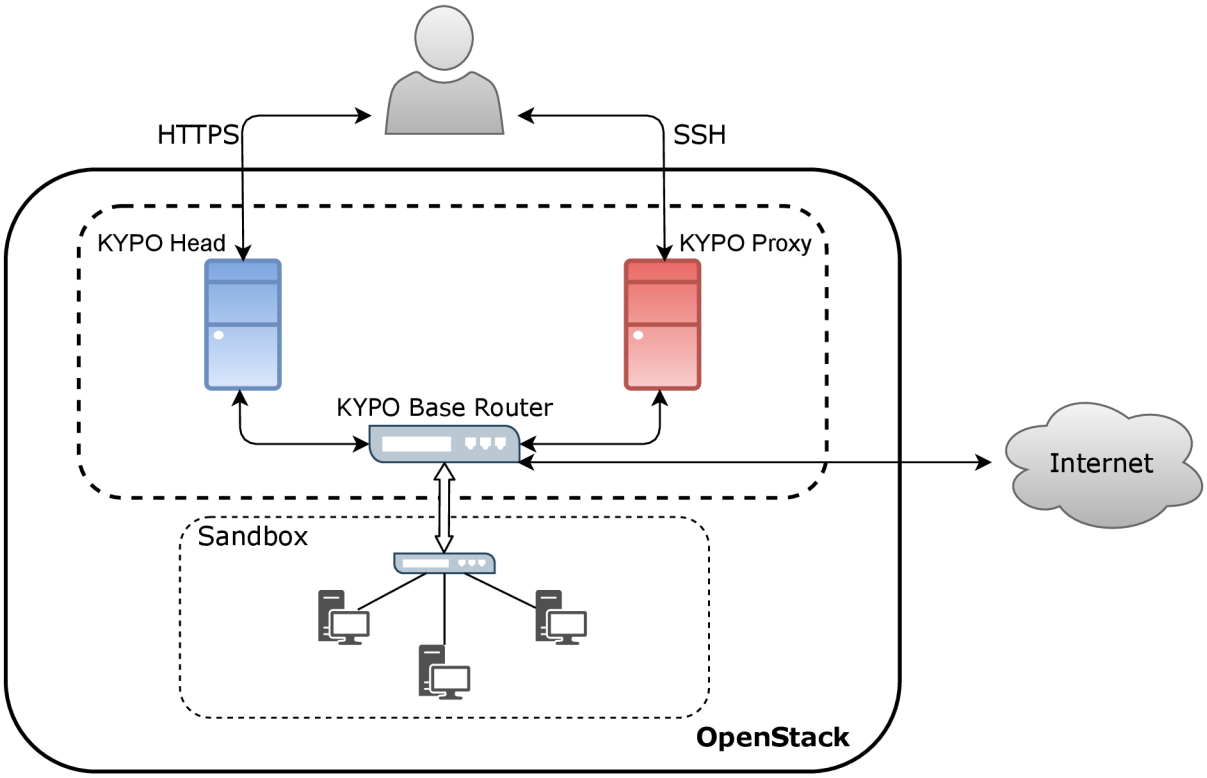
Instalace základní infrastruktury se skládá z vytvoření instancí KYPO Head a KYPO Proxy a interní sítě Kypo Base Network, do které obě instance spadají. Interní síť je přes výchozí bránu spojena s externí sítí `public`, čímž je zajištěn vzdálený přístup k instancím a konektivita do internetu. Základní infrastruktura platformy KYPO je znázorněna na obrázku 5.1.

KYPO Head je stanice, která je kontrolní stanicí celého cyber range a zprostředkovává všechny potřebné služby pro jeho fungování. **KYPO Proxy** je stanice zprostředkovávající SSH komunikaci mezi uživatelem a konkrétními sandboxy (viz kapitola 4). **KYPO Base Network** je interní síť, která zajišťuje konektivitu mezi vnitřními a vnějšími stanicemi, včetně spojení se samotnými sandboxy.

Nasazení KYPO cyber range zahrnuje vytvoření autentizačního serveru OIDC[11] pro správu uživatelů přistupujících na cyber range a konfiguraci backendu a frontendu samotného portálu KYPO.

Nasazení celé platformy KYPO následuje oficiální návod dostupného v literatuře [45], nicméně každá instalace může být specifická podle systému, na který je KYPO platforma nasazována, provedené příkazy jsou tedy upraveny tak, aby vyhovovaly hostující stanici

[11] OpenID Connect je autentizační metoda ověření uživatelů, kteří přistupují na server přes HTTPS spojení



Obr. 5.1: Základní infrastruktura KYPO

Kypo s OS Ubuntu 20.04 (viz tabulka 5.1), případné další odklony od oficiálního návodu jsou vždy upřesněny.

5.3.1 Nasazení základní infrastruktury

Instalace základní infrastruktury probíhá na stanici Kypo, na kterou se musí dodatečně nainstalovat balíčky *python3*, *pipenv*, *openssh-clients* a *jq*.

Před zahájením konfigurace a instalace je na OpenStack nahrán obraz OS Ubuntu 18 Bionic[12], který bude sloužit jako OS pro stanice KYPO Head a KYPO Proxy. Obraz je nahrán s názvem `ubuntu-bionic-x86_64` a musí mít přiřazen parametr *public*.

Pro instalaci základní infrastruktury je nutné poskytnout přístup ke konzolové správě platformy OpenStack, k tomu slouží tzv. *application credentials*, které jsou v podobě spustitelného skriptu `app-cred-kypo-openrc.sh` vygenerovány v prostředí Horizon a větvi admin s parametrem *unrestricted*. Následně jsou staženy konfigurační a instalační soubory z GitLab repozitáře projektu KYPO[13].

Provedené kroky zobrazuje výpis příkazů 5.7.

[12]Dostupný na adrese: `https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img`

[13]Dostupného na adrese: `https://gitlab.ics.muni.cz/muni-kypo-crp/devops/kypo-crp-openstack-base`

Výpis 5.7: Splnění prerekvizit pro instalaci základní infrastruktury

```
user@kypo:~$ sudo apt-get install python3-pip openssh-clients
    jq
user@kypo:~$ pip3 install pipenv
user@kypo:~$ source app-cred-kypo-openrc.sh
user@kypo:~$ git clone https://gitlab.ics.muni.cz/muni-kypo-
   crp/devops/kypo-crp-openstack-base.git
user@kypo:~$ cd kypo-crp-openstack-base
```

Ve složce `kypo-crp-openstack-base` je následně spuštěno virtuální prostředí pomocí *Pipenv shell* viz. výpis 5.8.

Výpis 5.8: Spuštění virtuálního prostředí

```
user@kypo:~/kypo-crp-openstack-base$ pipenv install
user@kypo:~/kypo-crp-openstack-base$ pipenv shell
```

Spuštěné virtuální prostředí se nesmí před dokončením instalace základní infrastruktury opustit.

Pro správnou instalaci základní infrastruktury je nutné upravit konfigurační soubor `openstack-defaults.sh` tak, aby odpovídaly konkrétní platformě OpenStack. Upravený soubor je zobrazen ve výpisu 5.9. Parametry `KYPO_HEAD_FLAVOR` a `KYPO_PROXY_FLAVOR` vyjadřují paměťové a výpočetní prostředky alokované dané instanci, tzv. *flavor*. Vyhrazené prostředky obsahuje tabulka 5.2, poslední flavor bude využit později při tvorbě scénářů. Uložený soubor je nastaven jako zdroj příkazem `source openstack-defaults.sh`.

Výpis 5.9: Upravený konfigurační soubor source openstack-defaults.sh

```
export KYPO_HEAD_FLAVOR="standard.large"
export KYPO_HEAD_IMAGE="ubuntu-bionic-x86_64"
export KYPO_HEAD_USER="ubuntu"
export KYPO_PROXY_FLAVOR="standard.medium"
export KYPO_PROXY_IMAGE="ubuntu-bionic-x86_64"
export KYPO_PROXY_USER="ubuntu"
```

Tab. 5.2: Vyhrazené prostředky pro OpenStack instance

Název *flavor*	Počet CPU	RAM	Disk
standard.large	4	8GB	80GB
standard.medium	2	4GB	80GB
csirtmu.tiny1x2	1	2GB	20GB

Nyní je možné přistoupit k instalaci základní infrastruktury. Při instalaci jsou vytvořeny stanice KYPO Head a KYPO Proxy, jsou jim přiřazeny plovoucí IP adresy pro přístup z externí sítě `public` a jsou vytvořeny soubory `admin_kypo-base-key.key` pro vzdálený bezheslový SSH přístup k oběma stanicím, viz výpis 5.10. Prvním příkazem jsou rezervovány plovoucí IP adresy pro obě instance ze sítě `public` a proběhne generování klíčového páru pro SSH přístup. Druhým příkazem se již vytvoří vnitřní síť KYPO Base, bezpečnostní skupiny a jednotlivé stanice. Bezpečnostní skupiny fungují jako firewall na principu přidávání pravidel (obdobně jako *iptables*). Bezpečnostní skupiny, které KYPO vytvoří, dědí pravidla z defaultní OpenStack bezpečnostní skupiny `default`. Defaultní skupiny jsou ve skutečnosti dvě, přičemž je jedna uživateli skrytá. V instalačních souborech základní infrastruktury je zadán název skupiny, ze které se dědí, tedy `default`, tímto vzniká kolize a OpenStack vytvoření nových bezpečnostních skupin nepřipustí. Je tedy nutné všechny parametry s názvem `default` nahradit jednoznačným ID označením bezpečnostní skupiny, konkrétně v souborech `/heat/kypo-base-remote-security-groups.yaml` a `/heat/kypo-base-security-groups.yml`.

Výpis 5.10: Vytvoření základní infrastruktury

```
user@kypo:~/kypo-crp-openstack-base$ ./bootstrap.sh public
user@kypo:~/kypo-crp-openstack-base$ ./create-base.sh
```

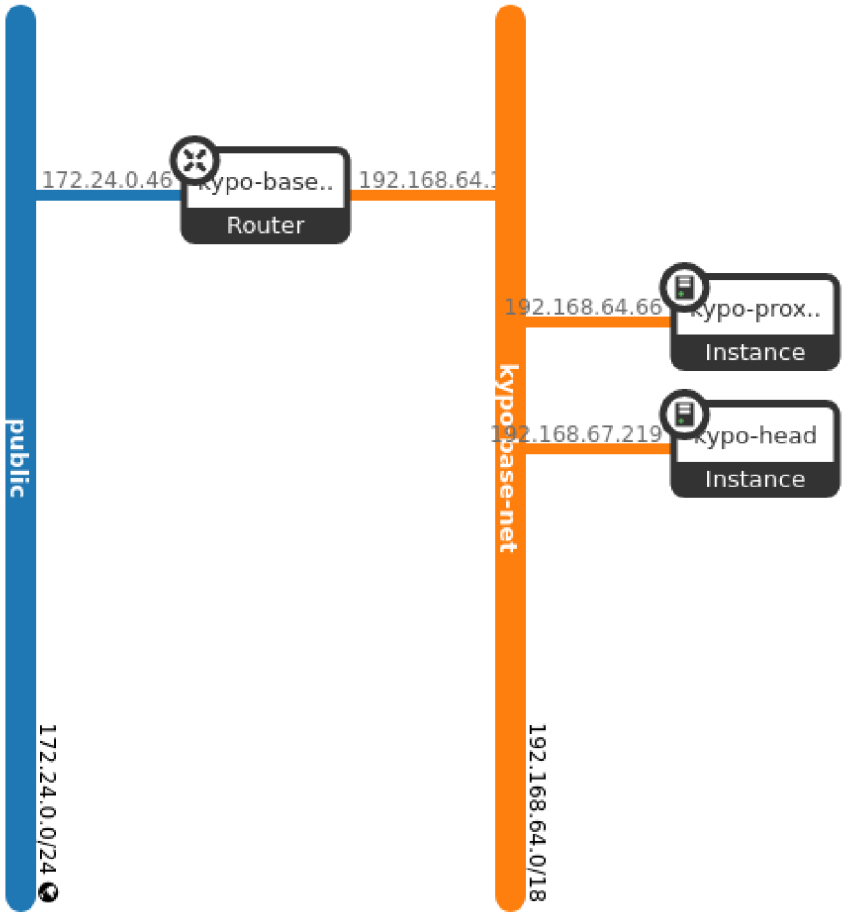
Nově vytvořené bezpečnostní skupiny `kypo-base-head-sg` a `kypo-base-proxy-sg` s povolením pro spojení přes protokoly SSH, HTTP a HTTPS. Podle instalačního návodu KYPO je přidána výjimka i pro protokol ICMP, v tomto případě se tak nestalo, proto bylo nutné bezpečnostní pravidlo pro protokol ICMP přidat dodatečně v prostředí Horizon. V opačném případě by nebylo možné se ke stanicím připojit a instalace by nemohla pokračovat. Tabulka 5.3 obsahuje všechna pravidla instancí `kypo-base-head-sg` a `kypo-base-proxy-sg`, které je nutné mít povolené.

Tab. 5.3: Bezpečnostní pravidla pro instance základní infrastruktury

Směr	IP protokol	Port
Příchozí	ICMP	Vše
Příchozí	SSH	22
Příchozí	TCP	443
Příchozí	TCP	8443
Odchozí	Vše	Vše

Pro kontrolu správnosti instalace stanic KYPO Head a KYPO Proxy a vytvořené nové sítě je spuštěn skript `ansible-check-base.sh`, pomocí kterého je ověřena dostupnost stanic a vzdálený přístup na obě stanice. Skript musí mít návratovou hodnotu `failed=0`.

Úspěšně vytvořená základní infrastruktura je znázorněna na obrázku 5.2, jedná se o znázornění síťové topologie, které vykresluje prostředí Horizon.



Obr. 5.2: Základní infrastruktura KYPO v prostředí Horizon

5.3.2 Nasazení KYPO cyber range

Tato část popisuje nasazení samotné platformy KYPO, tedy té části, ke které uživatelé přistupují, a přes kterou je spravován celý cyber range.

Nasazení KYPO je uskutečněno zejména pomocí nástroje Ansible, který na libovolnou stanici, ke které je možné přistoupit, instaluje a konfiguruje potřebné nástroje, spravuje uživatele, síťovou správu a zejména spojení s OpenStack cloudem. KYPO je možné nasadit dvěma způsoby, a to na stanici `kypo-head`, která běží jako instance na OpenStacku, nebo na lokální stroj jako virtualizovaný OS. Výběr způsobu záleží na několika faktorech. Zejména na zhodnocení metod přístupu k cyber rangi, pokud bude řídící stanice KYPO hostována na OpenStack cloudu, je k ní možné přistoupit z jakékoliv stanice v lokální síti (nebo pomocí VPN), která má zároveň přístup k OpenStacku[14]. Pokud bude KYPO lokálně virtualizovaný, je nutné na KYPO platformu přistoupit přímo z hostující stanice nebo provést dodatečnou síťovou správu pro vzdálený přístup k virtualizované stanici.

Dalším faktorem jsou paměťové a zejména výpočetní prostředky. Ve zdejším provedení má OpenStack předem určené prostředky a spuštění další instance by prostředky odebralo[15], které by mohly být použity pro trénovací scénář. Nasazení pomocí virtualizovaného OS je také mírně jednodušší, protože obsahuje z větší části již nakonfigurovaný Ansible s vytvořenými certifikáty pro HTTPS komunikaci s KYPO portálem a autentizačním serverem OIDC, což snižuje riziko zadání chybných údajů. Pro infrastrukturu, která je použita v této práci, byl vybrán způsob nasazení lokálně na virtualizovaný OS, a to hlavně z důvodu uvolnění výpočetních prostředků OpenStacku. Po vybrání této metody je instance `kypo-head` odstraněna ze základní infrastruktury, protože ji zastoupí stanice Kypo a virtualizovaný OS.

Nasazení znovu probíhá ze stanice Kypo, na které je nutné nainstalovat nástroje Ansible, *python3-passlib* a *bcrypt*. Pro správu a běh virtuálního stroje je potřeba VirtualBox a nástroj Vagrant. Nástroj Vagrant slouží pro automatické spouštění a správu přenosných virtuálních strojů, které mohou být ovládány například z CLI hostujícího OS. Pro běh virtuálního OS jsou potřeba 4 jednotky CPU a 8GB RAM, tedy stejné jako by bylo nutné vyhradit pro OpenStack instanci, ale využijí se prostředky stanice Kypo. Zdrojové soubory pro nasazení platformy KYPO se nacházejí v GitLab repozitáři Masarykovy univerzity[16].

Pro nasazení je nutné poskytnout údaje pro přístup ke službám OpenStacku a pro přístup na instanci `kypo-proxy-jump`. Je potřeba zajistit ID a heslo k tzv. *application credentials*, které slouží k autentizaci pro používání OpenStack služeb, použity mohou být

[14]Konkrétně k rozhraní, na kterém se nachází management uzel, v tomto případě se jedná o adresu 192.168.1.123

[15]Paměťové a výpočetní prostředky, které má instance vyhrazeny udává předem definovaný *flavor*, viz tabulka 5.2.

[16]Dostupného na adrese: `https://gitlab.ics.muni.cz/muni-kypo-crp/devops/kypo-crp-deployment`

ty údaje, které byly vygenerovány dříve pro instalaci základní infrastruktury. Dále je nutné zajistit vzdálený přístup k vnitřní instanci `kypo-proxy-jump` pomocí plovoucí IP adresy a klíče. Klíč může být použit ten, který= byl vygenerován při instalaci základní infrastruktury nebo může být znovu stažen, OpenStack uchovává všechny vygenerované klíče k příslušným instancím. Klíč je nutné zakódovat nástrojem bcrypt pomocí příkazu `base64`. Údaje se uloží do souboru `local-demo-extra-vars.yml`, který zobrazuje výpis 5.11. Zakódovaný klíč pro vzdálený přístup se uloží do souboru `local-demo-secrets.yml`, ve kterém se také nachází zakódovaný soukromý a veřejný klíč certifikátu pro zabezpečený přístup na KYPO platformu.

Parametr `kypo_crp_dns` slouží pro označení DNS (Domain Name System) serverů, na které budou přicházet požadavky od vnitřních KYPO instancí, v tomto případě má veřejná síť `public` přístup do internetu, tudíž je možné použít veřejné DNS servery.

Výpis 5.11: Obsah souboru local-demo-extra-vars.yml

```
# The URL of OpenStack Identity service API.
kypo_crp_os_auth_url: 192.168.1.123:5000
# The ID of application credentials to authenticate at the
   OpenStack cloud platform.
kypo_crp_os_application_credential_id:
   a817375ce1944833a28273abcf8fb807
# The secret string of Application Credential ID.
kypo_crp_os_application_credential_secret: <heslo>
# The KYPO Jump host IP address or hostname.
kypo_crp_proxy_host: 172.24.0.75
# The name of the user on the KYPO Jump host.
kypo_crp_proxy_user: ubuntu
kypo_crp_dns:
    - 8.8.8.8
    - 8.8.4.4
```

Po uložení souboru je přistoupeno k vytvoření virtuálního stroje. Údaje pro jeho vytvoření se nachází v souboru `Vagrantfile`, podle kterého Vagrant virtuální stanici vytvoří, nachází se v něm údaje o názvu stanice, síťových rozhraní, IP adrese a odkaz na zdroj obrazu stanice.

Nástroj Vagrant je spuštěn s odkazem na soubory `local-demo-extra-vars.yml` a `local-demos-ecrets.yml`, které předá nástroji Ansible pro automatizaci instalace frontendu a backendu KYPO platformy a autentizačního serveru OIDC. Nasazení je spuštěno příkazy ve výpisu 5.12.

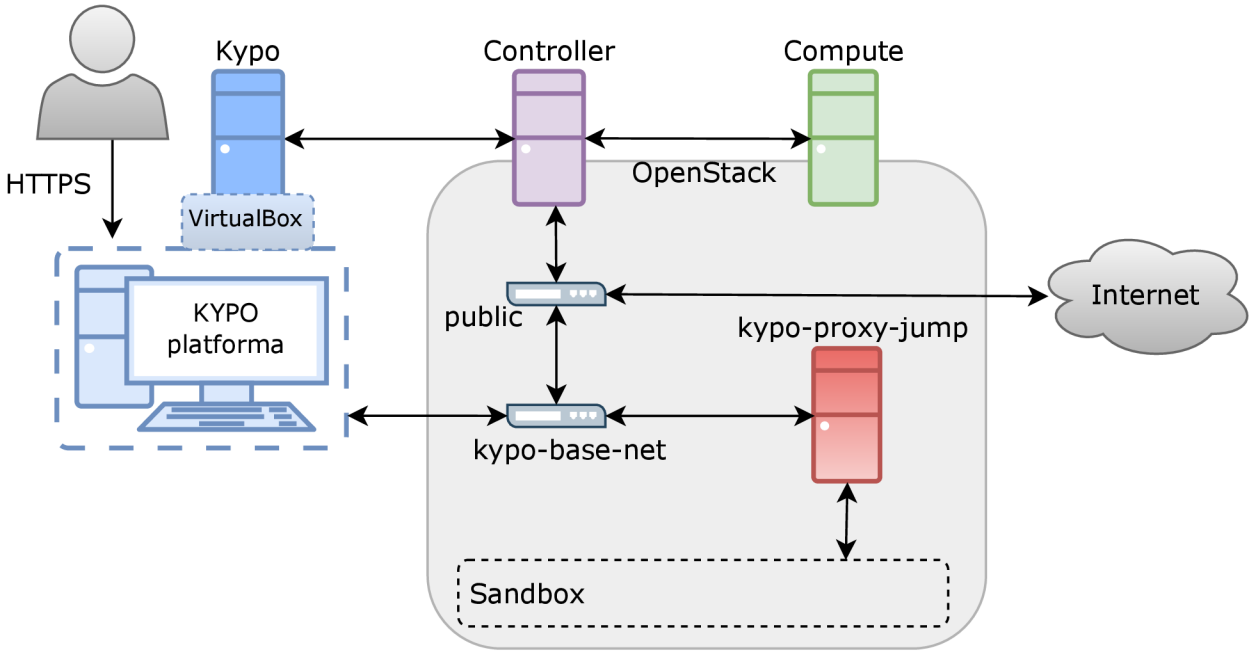
Výpis 5.12: Vytvoření a spuštění virtuálního stroje s platformou KYPO

```
user@kypo:~/kypo-crp-deployment$ vagrant box update
user@kypo:~/kypo-crp-deployment$ EXTRA_VARS=./local-demo-
   extra-vars.yml,./local-demo-secrets.yml vagrant up
```

Po zavedení samotné Kypo stanice do VirtualBoxu je do konzole vypisován průběh operací, které provádí Ansible. Jedná se o totožné kroky, které by byly provedeny v případě, že by nasazení probíhalo na stanici `kypo-head` na OpenStacku. Pokud by se během některého z kroků vyskytla chyba, jedná se pravděpodobně o chybně zadané údaje v předaných souborech, při řešení chyb je důležité vycházet z chybových hlášení vypsaných do konzole nástrojem Ansible.

Virtuální stanice je označena jako *run always*, tedy že je neustále spuštěna. Pokud by z nějakého důvodu došlo k přerušení běhu virtuální stanice, například z důvodu restartu hostitele, je možné stanici znovu zapnout příkazem `vagrant up` s příslušnými parametry. Pokud dojde k přerušení během nasazení, tak je nutné virtuální stroj spustit znovu příkazem `vagrant up --provision`, Ansible tak může zkontrolovat veškeré provedené kroky a některé případně provést znovu, tento krok se také musí provést, pokud dojde ke změně některých parametrů (například klíče pro SSH přístup). Práce s Vagrantem musí být vždy prováděna ze složky `kypo-crp-deployment`.

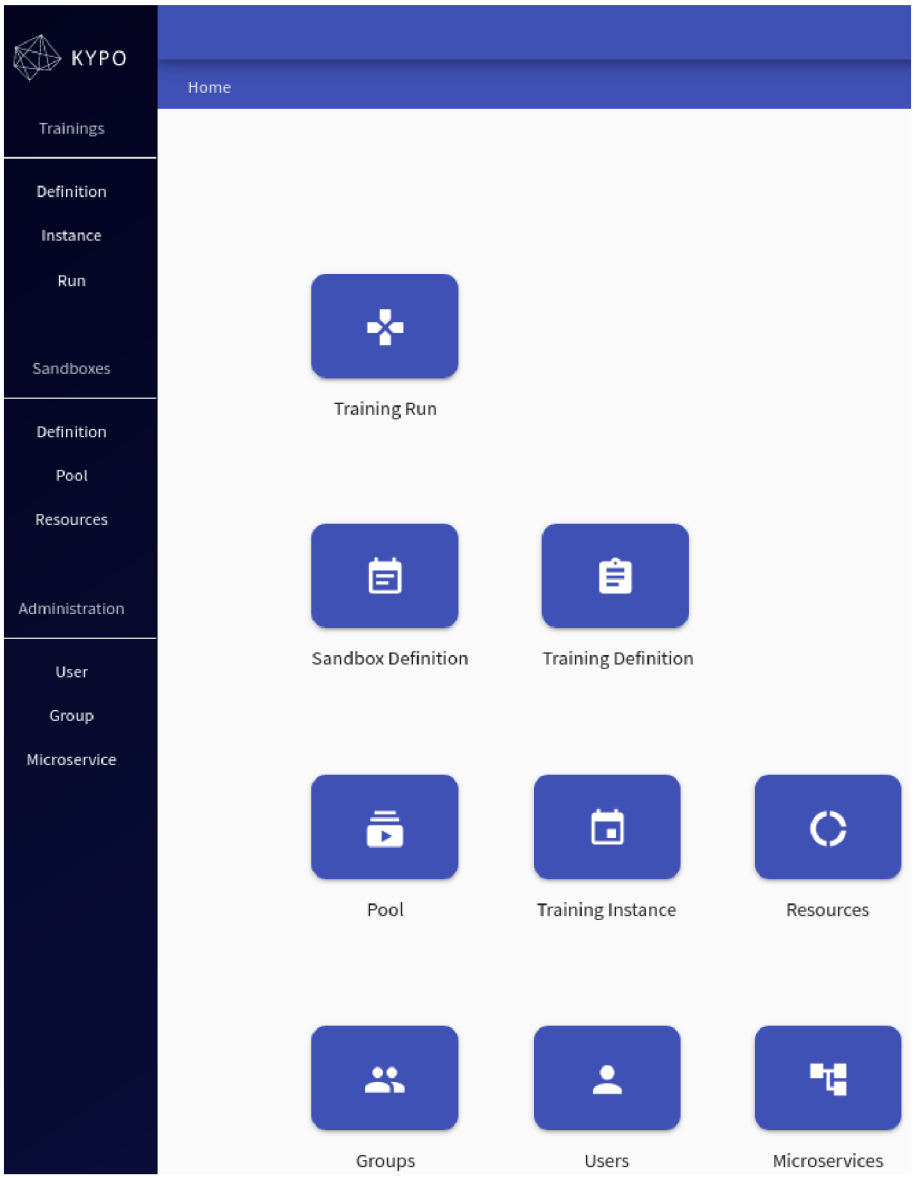
Pokud po dokončení nasazení Ansible vrátí hodnotu `failed=0` platforma KYPO je připravena k používání na IP adrese `172.19.0.22`. Konečná infrastruktura je znázorněna na obrázku 5.3.



Obr. 5.3: Úplná infrastruktura platformy KYPO a OpenStacku

5.4 Ovládání platformy KYPO

Ve výchozím nastavení je vytvořen jeden hlavní uživatel s plnými právy `kypo-admin` pro ovládání KYPO platformy a tvorbu nových uživatelů a tři obyčejní uživatelé pro přístup k trénovacím scénářům `kypo-user`, `john.doe` a `jane.doe`. Výchozí heslo je pro všechny stejné `password`. Autentizace na KYPO je vždy přes OIDC server, kam je uživatel pro přihlášení přesměrován. Noví uživatelé a správa účtů je prováděna hlavním uživatelem na adrese `https://172.19.0.22:8443/csirtmu-dummy-issuer-server`. Domovská strana portálu KYPO platformy z pohledu hlavního uživatele je zobrazena na obrázku 5.4



Obr. 5.4: Portál KYPO platformy z pohledu hlavního uživatele

Sekce **Training Definition** je určena pro vytváření a úpravám trénovacích scénářů z pohledu cvičícího. Je zde upravován text scénáře, kterým je cvičící prováděn, zadávány flagy, otázky, jejich ohodnocení a nápovědy. Celý trénovací scénář může být také importován v souboru typu JSON. V sekci **Training Instance** jsou zobrazeny všechny možné *pooly*, které jsou dostupné, alespoň jeden musí být scénáři přiřazen (viz níže), dále je zde definována doba, po kterou je cvičení zpřístupněno a určen přístupový kód, který cvičící musí zadat pro přístup k trénovacímu scénáři. V sekci **Training Run** je zadáván přístupový kód a poté zobrazen obsah samotného scénáře, případně výsledky provedeného cvičení. Toto je jediná sekce, kterou vidí obyčejní uživatelé, tedy cvičící po přihlášení se do Portálu.

Sekce **Sandbox Definition** slouží pro definici sandboxů. Definice jsou importovány z GitLab repozitáře, který je specifikován před nasazení KYPO platformy v souboru `local-demo-extra-vars.yml`, nyní je možné čerpat pouze z jednoho repozitáře, který je ve výchozím stavu nastaven na interní repozitář MUNI projektu KYPO, v tomto nasazení tak bylo ponecháno. Sandbox definice zpravidla obsahuje hlavní soubor `sandbox.yml`, kde jsou specifikovány veškeré instance a sítě, které budou v OpenStacku vytvořeny, stejně tak prostředky (*flavors* a diskové obrazy), které jsou pro vytvoření konkrétní instance potřeba. Sekce **Pool** slouží pro vytvoření tzv. *poolů*. Jeden *pool* značí jednu definici sandboxu alokovanou na OpenStacku. Alokací jednoho *poolu* nastane vyčerpání prostředků OpenStacku definovaných pro jeden sandbox. Dojde tedy k vytvoření všech instancí, sítí a síťových prvků, které jsou vázané na konkrétní trénovací scénář, tento krok procesu celé alokace *poolu* je označována jako *OpenStack Stage*. V dalším kroku jsou jednotlivé stanice upravovány, aby vyhovovaly prerekvizitám konkrétního trénovacího scénáře, tento krok je nazýván jako *Ansible Stage*. Průběh celé alokace je pomocí Spice konzole z instance `kypo-proxy-head` uživateli vypisován, pokud tedy nastane během nějakého kroku chyba, lze ji dohledat. Alokace postupně prochází stavy *In Queue*, *Running*, *Finished* nebo *Failed*. Nepovedená alokace se musí zrušit a spustit celá znovu. Alokováno může být více *poolů*, platí, že pro jeden aktivní trénovací scénář, tedy pro jednoho aktivního hráče, musí být alokován jeden *pool*. Pro více hráčů musí být alokováno odpovídající množství *poolů*, s tím samozřejmě souvisí i použití dalších výpočetních a paměťových prostředků. Jeden sandbox se tedy může skládat z více identických *poolů*. Sekce **Resources** zobrazuje vyčerpané prostředky z celkových možných, které OpenStack může alokovat. Údaje jsou přímo přebrány z prostředí Horizon. Dále je zde seznam všech dostupných obrazových disků poskytovaných službou Heat.

Sekce **Administration** slouží ke správě uživatelů, přidělování rolí a správě uživatelských skupin. V sekci **Microservices** je možné registrovat nové služby do KYPO platformy. Například službu Elasticsearch pro vyhledávání a filtraci dat na webových stránkách.

Detailnější zobrazení jednotlivých sekcí je možné nalézt v literatuře [45] v kapitole

User Guide.

5.4.1 Zprovoznění ukázkového scénáře

KYPO nabízí jeden ukázkový trénovací scénář typu CTF, který demonstruje schopnosti KYPO platformy a možnosti ovládání platformy zejména pro hlavní uživatele a organizátory cvičení. Tento scénář tedy neslouží pro trénovací účely, cílem tohoto scénáře je připojit se přímo do instance sandboxu z externí sítě pomocí SSH připojení.

V této kapitole je popsán postup zprovoznění ukázkového scénáře[17].

Import trénovací definice

Trénovací definice je soubor ve formátu JSON, podle kterého jsou cvičící prováděni. Postup importu trénovací definice je následující:

1. Přihlásit se ke KYPO platformě za hlavního uživatele `kypo-admin`,
2. v sekci *Training* zvolit *Definition*,
3. zvolit volbu *Upload*,
4. vybrat soubor `training.json`.

Po nahrání definice je možné ji upravit již v platformě KYPO. V trénovací definici je také možné vybraným uživatelům udělit roli organizátorů cvičení, které následně mají plné právo cvičení upravovat a případně spouštět.

Definice sandboxu

Sandbox určuje topologii, která je vytvořena v OpenStacku pro pro daný scénář. Definice se importují z přístupného GitLab repozitáře, který je specifikován před nasazením celé KYPO platformy. Sandbox ukázkového scénáře má následující strukturu.

V souboru `sandbox.yml` se nacházejí veškeré informace o instancích a sítích, které mají být vytvořeny, tedy název instance, flavor, zařazení instance do sítě, obrazový disk, uživatelé. Pro síť je specifikován název a IP adresa, podsíť a alokační velikost IP adres. Postup vytvoření sandboxu je následující:

1. V sekci *Sandboxes* zvolit *Definition*,
2. zvolit *Create*,
3. pole vyplnit dle tabulky 5.4.

Pokud je spojení s repozitářem úspěšné, sandbox je připraven k alokaci.

[17] Dostupného na adrese: `https://gitlab.ics.muni.cz/muni-kypo-crp/prototypes-and-examples/sandbox-definitions/kypo-crp-demo-training/-/tree/master`

Tab. 5.4: Import definice sandboxu ukázkového scénáře

SSH adresa	git@git-internal-ssh:/repos/prototypes-and-examples/sandbox-definitions/kypo-crp-demo-training.git
revize	master

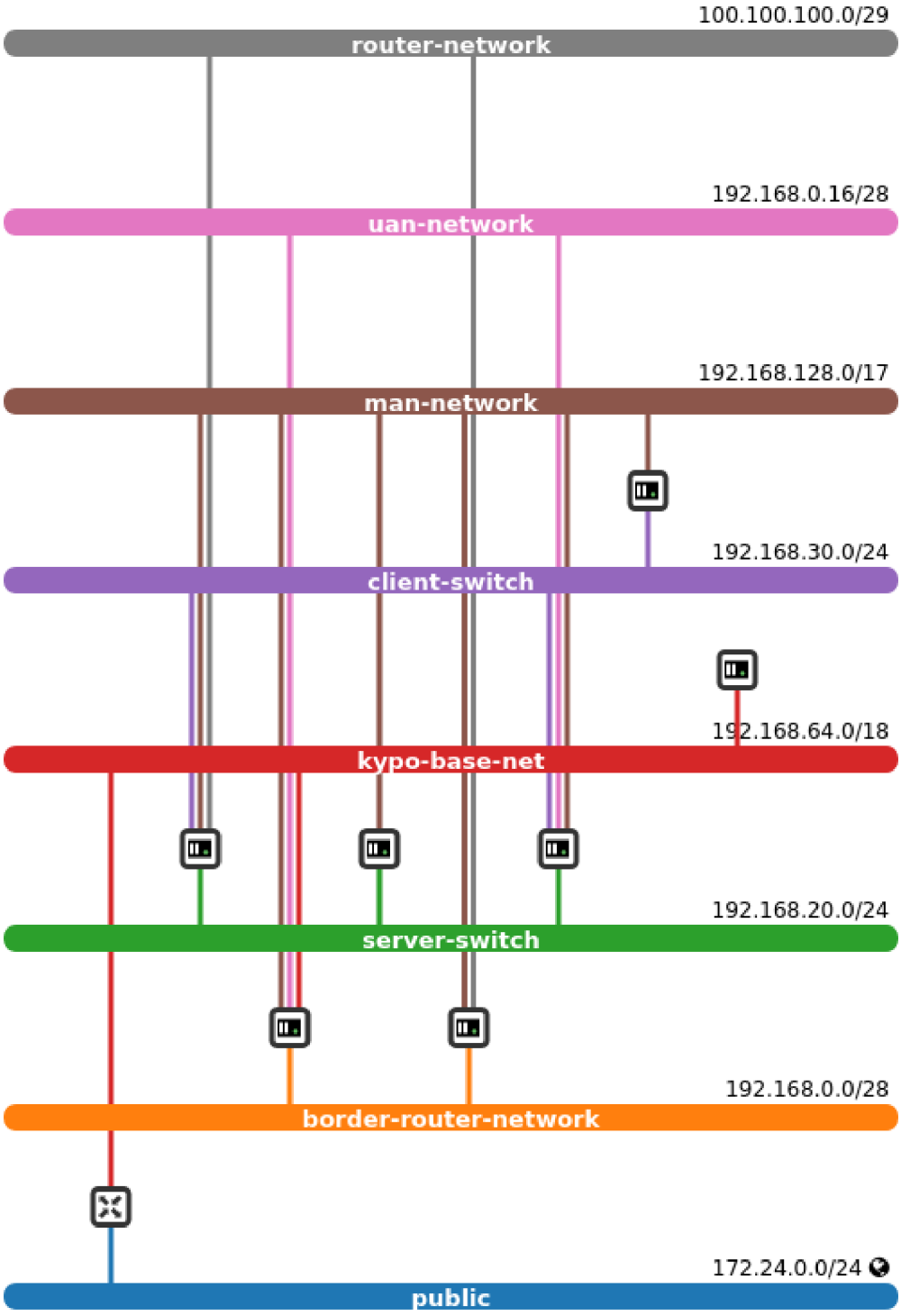
Alokace sandboxu

Samotná alokace sandboxu, přestože je automatizovaná, skládá se z mnoha kroků a jedná se o nejkritičtější část zprovoznění jakéhokoliv scénáře. Alokace se skládá ze tří částí, tzv. *stages*, těmi jsou OpenStack Stage, Anssible Networking Stage, Ansible User Stage.

Spuštění alokace sandboxu pro jeden *pool* je následující:

1. V sekci *Sandboxes* zvolit *Pool*,
2. zvolit *Create*.

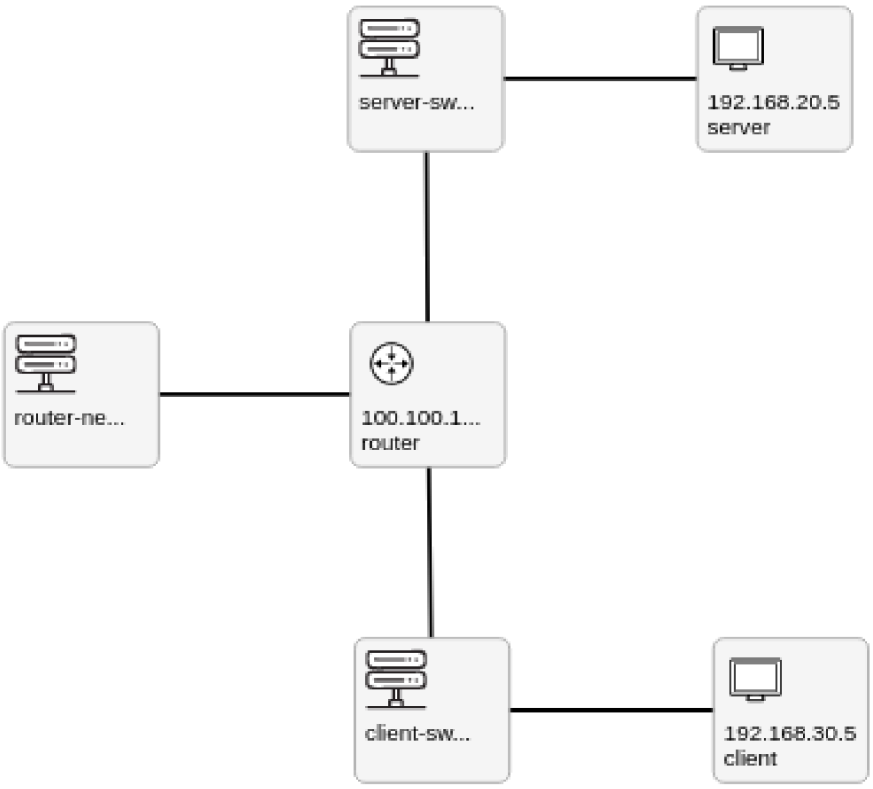
Alokace sandboxu probíhá přes instanci `kypo-proxy-jump` a celý průběh alokace ve formě výpisu Ansible *playbooků* je zobrazen pomocí Spice konzole. Očekávaným výsledkem je stav *Finnished*. Doba alokace se pohybuje kolem deseti minut. Po dokončení alokace je možné si vytvořenou topologii sandboxu prohlédnout přímo v prostředí Horizon, viz obrázek 5.5.



Obr. 5.5: Topologie ukázkového scénáře zobrazená v prostředí Horizon

Platforma KYPO také vytvoří uživatelsky přívětivější topologii, která obsahuje pouze části důležité pro samotný scénář, viz obrázek 5.6. Tato topologie je také k dispozici

při vypracovávání samotného scénáře cvičení.



Obr. 5.6: Topologie ukázkového scénáře na platformě KYPO

Spuštění scénáře

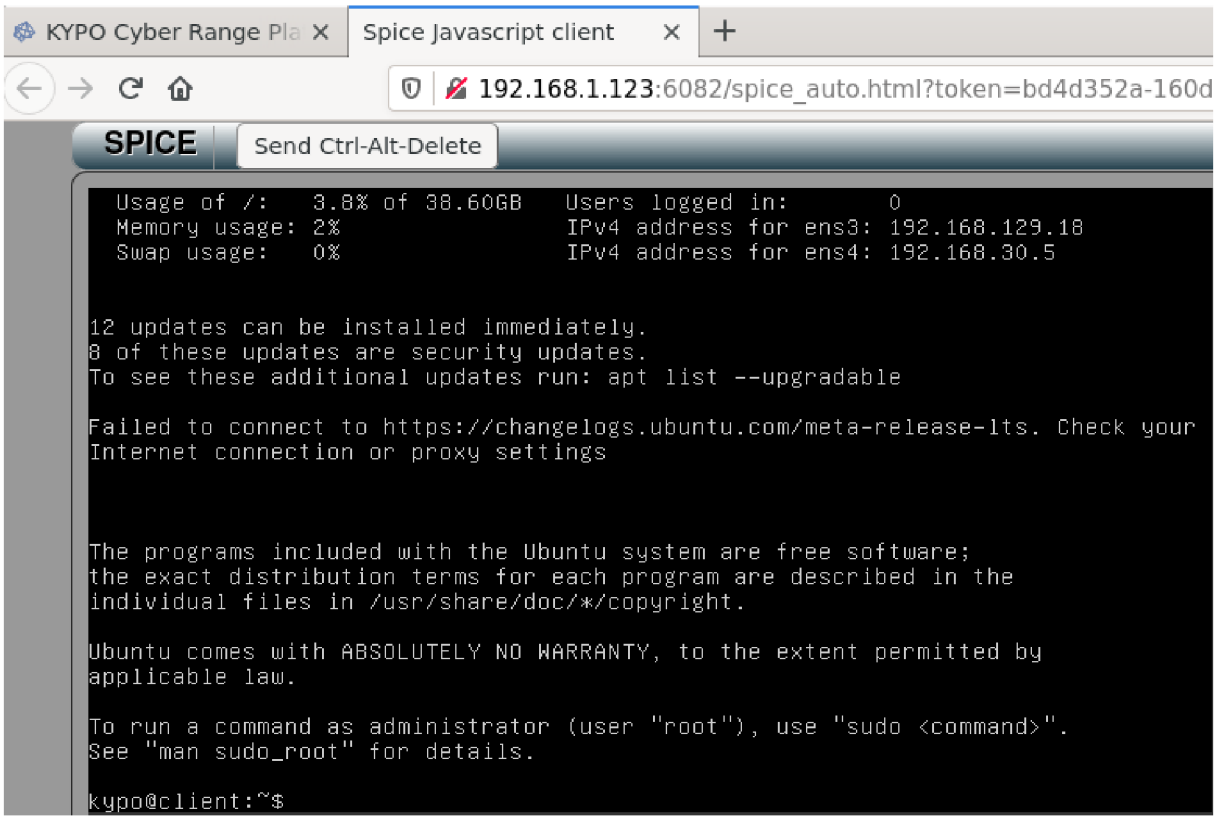
Před spuštěním scénáře je ještě nutné přiřadit trénovací definici příslušnému alokovanému sandboxu a specifikovat, kdy bude scénář dostupný. Obecně je možné vytvořit tolik trénovacích instancí, respektive scénářů cvičení, kolik je dostupných *poolů*. V tomto případě je dostupný jen jeden *pool*. Postup pro vytvoření instance scénáře je následující:

1. V sekci *Training* zvolit *Instances*,
2. zvolit *Create*,
3. vyplnit údaje o scénáři - název, čas začátku a konce cvičení, zvolit čtyřmístný PIN pro přístup do cvičení,
4. zvolit *Assign Pool* a vybrat příslušný alokovaný *pool*,
5. libovolně zvolit organizátory cvičení,
6. zvolit *Create*.

Scénář je nyní dostupný ke spuštění. S instancí se vygeneroval tzv. *Access Token*, který se skládá z vybraného PINu a dalších náhodně vygenerovaných čtyř číslic. Pro spuš-

tění cvičení je nutné zadat celý token. Samotné cvičení je následně spuštěno v sekci *Run*. Ukázka průběhu cvičení je zobrazena v příloze A.

Pro připojení přímo do instance sandboxu je využívána Spice konzole. Pro přístup k instanci z KYPO scénáře je vygenerována adresa, která umožní účastníkovi ovládat instanci přímo v OpenStacku. Ukázka vygenerované Spice konzole je na obrázku 5.7.



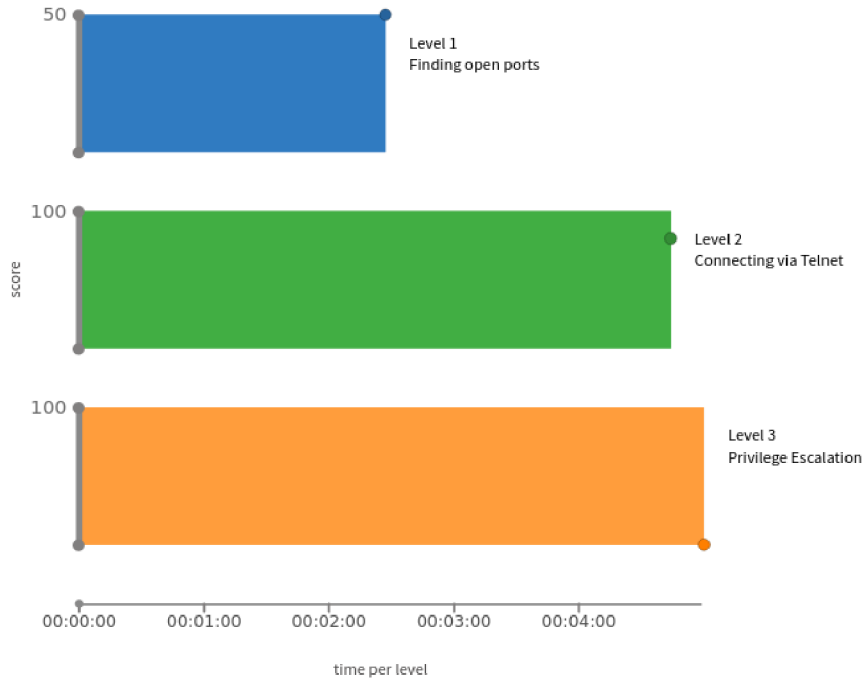
Obr. 5.7: Přístup do instance sandboxu pomocí Spice konzole

Po dokončení scénáře platforma KYPO vygeneruje v podobě grafů statistické údaje získané během cvičení. Organizátoři tak mohou hodnotit trvání určitých úkolů, nesprávné odpovědi, případně použité nápovědy. Tyto údaje je také možné sledovat v průběhu cvičení. Na obrázku 5.8 se nachází vygenerovaný graf jednotlivých úloh a jejich ohodnocení v závislosti na délce vypracování dané úlohy.

V případě, že by uživatel chtěl scénář spustit znovu, je nutné alokovat nový *pool* a k němu přiřadit trénovací definici. Opětovné spuštění stejného scénáře není možné z důvodů změn, které v instancích sandboxu proběhly v důsledku plnění cvičení.

Vyčištění sandboxu

Po dokončení cvičení je nejprve nutné odebrat *pool* z trénovací instance. Poté je možné v sekci *Sandboxes - Pool* konkrétní *pool* vymazat spuštěním žádosti na vymazání. Sandbox



Obr. 5.8: Časový graf jednotlivých úkolů ukázkového scénáře

po tomto kroku zůstává stále definovaný a lze jej vymazat až po odstranění všech *poolů*. Jiný postup není doporučován, i když je možné definici sandboxu vymazat i s alokovaným *poolem*, organizátor nad ním následně ztratí kontrolu a vymazání *poolu* je možné pouze přímo v OpenStacku.

5.4.2 Import vlastního sandboxu

Při nasazení KYPO platformy musí být dostupný právě jeden GitLab repozitář, ze kterého se importují sandboxové definice pro konkrétní cvičení. V ukázkovém scénáři byl použit interní GitLab repozitář, který KYPO tým poskytuje k demonstraci používání platformy a možností, které nabízí. Použití interního repozitáře je z hlediska plného využití KYPO platformy nevýhodné, jelikož neumožňuje importovat vlastní definice sandboxů, které jsou vytvořeny pro specifické cvičení, například v budoucnosti pro studenty. KYPO nabízí vlastní nástroj pro vytváření trénovacích scénářů zvaný Sandbox Creator, který sandboxy generuje cíleně pro použití na KYPO platformě, více v literatuře [48].

Jako testovací sandbox byl použit sandbox vytvořený v rámci bakalářské práce Virtualizační platforma pro bezpečnostní cvičení (literatura [49]), který byl nahrán na osobní

GitLab repozitář[18]. Adresa cesty k projektu se sandboxem musí být v tomto formátu `https://gitlab.com/skupina/podskupina/projekt`. V čase psaní této části práce[19] totiž KYPO neumí importovat projekty, které nejsou zanořené minimálně do skupiny a podskupiny.

Při první fázi alokace sandboxu je používán Ansible playbook, který kontroluje funkčnost spojení mezi stanicí Kypo, instancí `kypo-proxy-jump` a instancemi v sandboxu. Projekt s tímto playbookem musí být dostupný ve stejném repozitáři jako projekt se sandboxem, playbook je proto importován z externího KYPO repozitáře[20].

Údaje o GitLab repozitáři, ze kterého jsou sandboxy importovány, jsou zadávány před nasazením KYPO platformy v souboru `local-demo-extra-vars.yml`. KYPO umí pracovat pouze s GitLab repozitářem, ke kterému má uživatel plný přístup, je nutné zajistit tyto údaje:

- URL adresu serveru,
- URL adresu GitLab REST API serveru,
- SSH klíče pro bezheslový přístup,
- Access Token uživatele pro autentizaci na REST API server,
- SSH adresu k Ansible playbooku a název revize.

Upravené části souboru `local-demo-extra-vars.yml` zobrazuje výpis 5.13. Klíčový pár pro SSH přístup a Access Token jsou z výpisu vynechány. Veřejný a soukromý klíč musí být zakódovány příkazem `Base64`.

Výpis 5.13: Upravené parametry pro GitLab repozitář

```
## The optional Git repository settings.
#kypo_crp_git: '{␣kypo_crp_git_internal␣}'
kypo_crp_git:
 type: GITLAB
 server: gitlab.com
 rest_server_url: https://gitlab.com/api/v4/
 user: git
 private_key: <zakódovaný soukromý klíč>
 public_key: <zakódovaný veřejný klíč>
 access_token: <string tokenu>
 ansible_networking_url: https://gitlab.com/kypo-crp/sanbox-
    definitions/kypo-ansible-stage-one.git
 ansible_networking_rev: master
```

[18]Dostupný na adrese: `https://gitlab.com/kypo-crp/sanbox-definitions/small-sandbox-training.git`

[19]Duben 2021

[20]Ansible playbook je dostupný na adrese: `https://gitlab.ics.muni.cz/muni-kypo-crp/backend-python/ansible-networking-stage/kypo-ansible-stage-one`

Po uložení souboru je nutné spustit Vagrant s Ansible playbooky znovu. Toho je docíleno příkazem `vagrant up --provision`, bohužel tímto postupem nejsou restartovány Docker kontejnery, pomocí kterých jsou jednotlivé komponenty KYPO platformy nasazovány. Důsledkem toho byla při snaze o import sandboxu zobrazena chybová hláška *„The GIT user does not match the configured value for this instance: expected='ssh', actual=git“*. Problém byl vyřešen vypnutím virtuálního OS s KYPO platformou a následným zapnutím s upraveným souborem, viz výpis 5.14.

Výpis 5.14: Spuštění nástroje Vagrant s upravenými GitLab parametry

```
user@kypo:~/kypo-crp-deployment$ Vagrant halt
user@kypo:~/kypo-crp-deployment$ EXTRA_VARS=./local-demo-
   extra-vars.yml,./local-demo-secrets.yml vagrant up --
   provision
```

Kypo platforma je následně opět dostupná na adrese `172.19.0.22`. Postup importu sandboxu je totožný jako v případě importu ukázkového scénáře s tím rozdílem, že je zadána odlišná SSH adresa pro klonování projektu s vlastním sandboxem. Zadávané údaje pro import sandboxu obsahuje tabulka 5.5.

Tab. 5.5: Import definice vlastního sandboxu

SSH adresa	git@gitlab.com:kypo-crp/sanbox-definitions/small-sandbox-training.git
revize	master

Při importu sandboxu je zobrazena chybová hláška *„Not enough values to unpack (expected 2, got 1)“*. Dle informací od KYPO týmu tato chyba je pravděpodobně způsobena při pokusu o import sandboxu, který nemá vyhovující strukturu. Proto byl na osobní repozitář nahrán ukázkový scénář, který má správnou strukturu. V tomto případě se zobrazilo chybové hlášení *„Sandbox.yml 404: 404 Not Found“*, i přesto, že projekt je zanořen do požadovaného formátu cesty.

Shrnutí importu vlastních sandboxů

Použití vlastních sandboxových definic importovaných z veřejného repozitáře je dle KYPO dokumentace možné při vhodném přizpůsobení zdrojového souboru `local-demo-extra-vars.yml`. V tomto případě import nebyl úspěšný. Import z vlastních sandboxů není důkladně vyzkoušená služba, kterou KYPO nabízí. Při řešení potíží importu bylo komunikováno s KYPO týmem, který se snažil určit kořeny chyb, které vznikly. Ze závěru KYPO týmu vyplynulo, že se pravděpodobně jedná o chybu, která bude řešena. V čase psaní práce tedy není možné určit zdroj chyby nefunkčního importu a následného vyřešení.

Závěr

V bakalářské práci byly představeny typy bezpečnostních cvičení, na jejichž principu jsou v současnosti stavěny cyber range, které slouží k praktickému využití znalostí nejen pro ICT specialisty v bezpečném a izolovaném prostředí s využitím různých druhů trénovacích scénářů.

V navazující kapitole bylo vybráno pět platforem pro stavbu cyber range. Těchto pět platforem bylo vybráno, protože každá z nich se staví k přístupu vytvoření cyber range odlišným způsobem. Vyzdvižena je platforma SecDevOps@Cuse CyberRange, jelikož využívá cloud jako výpočetní a paměťový prostředek, což je založeno na podobném principu stavění cyber range jako KYPO platforma, na kterou je tato práce zaměřena.

SecDevOps@Cuse CyberRange využívá externí cloud se zpoplatněnými službami, platforma KYPO je založena na open-source cloudové platformě OpenStack, kterou je možné nasadit do vlastní infrastruktury. OpenStack využívá mnoho služeb, které nabízí jednotlivé komponenty a nejpodstatnější z nich byly představeny. Následoval popis metod nasazení OpenStacku a u každé z nich byla shrnuta vhodnost, či nevhodnost použití nasazení pro tuto práci.

Závěrem teoretické části byla provedena analýza platformy KYPO a na základě uvedených poznatků byla vybrána metoda nasazení OpenStacku tak, aby vyhovovala potřebám platformy KYPO.

Praktická část se skládá ze tří částí - nasazení OpenStacku, nasazení platformy KYPO a zprovoznění ukázkového scénáře. Po původně nevhodně zvolené metodě nasazení pomocí nástroje PackStack, byl OpenStack nasazen pomocí nástroje Kolla-Ansible, který se nyní jeví jako nejvhodnější metoda vzhledem k potřebám platformy KYPO. Kvůli změně metody nasazení z metody PackStack na Kolla-Ansible bylo nutné upravit použitou infrastrukturu z jedné stanice na celkem tři stanice.

Nasazení jednotlivých platforem OpenStack a KYPO není v zásadě komplikované, ale obě dvě spolu musí precizně kooperovat. Z tohoto důvodu musela být platforma OpenStack několikrát nasazena znovu, jelikož až při přípravách na nasazení platformy KYPO, v některých případech dokonce po jejím nasazení, bylo zjištěno, že došlo k nevhodné konfiguraci OpenStack komponent, které zamezovaly správné činnosti KYPO cyber range. Praktická část plně vycházela z oficiálního návodu pro nasazení a správu platformy KYPO, ale z toho důvodu, že je stále celý projekt velmi čerstvý, není možné zachytit veškeré nesoulady, ke kterým může v jednom z mnoha kroků dojít. V práci jsou poznatky, které vedly k funkčnímu nasazení, a nebyly objasněné v KYPO dokumentaci, uvedeny.

Nasazená KYPO platforma je plně funkční. Na platformě byl zprovozněn a vyzkoušen dostupný ukázkový scénář, který KYPO nabízí k demonstraci funkčnosti cyber range. KYPO platforma tréninkové scénáře importuje z GitLab repozitáře, který je specifikován před samotným nasazením platformy. Z důvodu budoucího použití vlastních tréninkových

scénářů byla KYPO platforma nasazena podruhé se specifikovaným vlastním GitLab repozitářem pro vyzkoušení importu vlastního scénáře, který ale nebyl úspěšný i za pomoci podpory KYPO týmu.

Projekt KYPO je jako platforma pro bezpečnostní cvičení jako open-source velmi čerstvá. Projekt je ojedinělý a perspektivní, je na něm stále pracováno a případné chyby nebo nesrovnalosti jsou dosud řešeny, proto je pro plně funkční nasazení KYPO platformy potřebná spolupráce s KYPO týmem.

Literatura

[1] KOLOUCH Jan, BAŠTA Pavel a kol. *CyberSecurity* . 1. vydání. Praha: CZ.NIC, z.s.p.o., 2019. 556 s. ISBN 978-80-88168-31-7.

[2] INTERPOL. *INTERPOL report shows alarming rate of cyberattacks during COVID-19* [online]. 2020 [cit. 2020-12-04]. Dostupné z : <https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19>.

[3] NIST. *Cyber Ranges* [online]. 2018 [cit. 2020-11-04]. Dostupné z : <https://www.nist.gov/system/files/documents/2018/02/13/cyber_ranges.pdf>.

[4] NÚKIB. *Typy cvičení* [online]. 2019 [cit. 2020-11-04]. Dostupné z: <https://www.nukib.cz/cs/kyberneticka-bezpecnost/cviceni/typy-cviceni/>.

[5] CCDCOE. *New Study on the Status of Cyber Security Organization in Czechia* [online]. 2019 [cit. 2020-11-04]. Dostupné z: <https://ccdcoe.org/news/2019/new-study-on-the-status-of-cyber-security-organization-in-czechia/>.

[6] VYKOPAL Jan, VIZVARY Martin, OSLEJSEK Radek, CELEDA Pavel, TOVARNAK Daniel. *Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range* [online]. Brno. 2017 [cit. 2020-11-04]. Dostupné z: <https://is.muni.cz/publication/1391675/2017-FIE-lessons-learned-exercises-cyber-range-paper.pdf>.

[7] CLIMER Siobhan, KHAN Mishaal. *Red Team Vs Blue Team: The Two Sides Of Cybersecurity: A Cybersecurity Report* [online]. Mindsight, 2020 [cit. 2020-11-04]. Dostupné z: <https://gomindsight.com/insights/blog/red-team-vs-blue-team/>.

[8] VIGNA Giovani. *Teaching network security through live exercises* [online]. In: Security education and critical infrastructures. p. 3-18. Boston, MA, 2003 [cit. 2020-11-04]. Dostupné z: <https://www.researchgate.net/publication/221211746_Teaching_Network_Security_Through_Live_Exercises>.

[9] DUBEY Siddhant. *An Introduction to Cybersecurity, Capture the Flag Contests, and Basic Security Concepts* [online]. Medium, 2019 [cit. 2020-11-04]. Dostupné z: <https://medium.com/better-programming/an-introduction-to-cybersecurity-capture-the-flag-contests-and-basic-security-concepts-80f3fbf62bbc>.

[10] REDSCAN. *What is purple teaming and how can it strengthen your cyber security?* [online]. 2018, poslední aktualizace 2020-10-09 [cit. 2020-11-04]. Dostupné z: <https://www.redscan.com/news/purple-teaming-can-strengthen-cyber-security/>.

[11] CAPPETTA. *CyberRange Overview* [online]. 2019, poslední aktualizace 2020-07-12 [cit. 2020-11-04]. Dostupné z: <https://github.com/secdevops-cuse/CyberRange/blob/master/README.md>.

[12] BRANDON John. *AWS: Your complete guide to Amazon Web Services and features* [online]. Techradar.pro, 2020 [cit. 2020-11-04]. Dostupné z: <https://www.techradar.com/news/aws#complete-list-of-amazon-web-services>.

[13] FORMENTO John, Jr., CERINI Adam. *What is a cyber range and how do you build one on AWS?* [online]. AWS, 2020 [cit. 2020-11-04]. Dostupné z: <https://aws.amazon.com/blogs/security/what-is-cyber-range-how-do-you-build-one-aws/>.

[14] BEURAN Razvan, TANG Dat Thanh, PHAM Cuong, CHINEN Ken-ichi, TAN Yasuo, SHINODA Yoichi. *Integrated Framework for Hands-on Cybersecurity Training: CyTrONE* [online]. In: Computers and Security, 2018 [cit. 2020-11-04]. Dostupné z: <https://www.researchgate.net/publication/325787301_Integrated_Framework_for_Hands-on_Cybersecurity_Training_CyTrONE>.

[15] CROND-JAIST. *CyTrONE: Integrated Cybersecurity Training Framework* [online]. 2017, poslední aktualizace 2020-03-13 [cit. 2020-11-04]. Dostupné z: <https://github.com/crond-jaist/cytrone/blob/master/README.md>.

[16] JAIST, Cyber Range Organization and Design. *User Guide* [online]. Japan. 2019 [cit. 2020-11-04]. Dostupné z: <https://github.com/crond-jaist/cytrone/releases/tag/1.1>.

[17] JAIST. *Achievements* [online]. 2018 cit. [2020-11-06]. Dostupné z: <https://www.jaist.ac.jp/misc/crond/achievements-en.html>.

[18] AWS. *Amazon Virtual Private Cloud User Guide: Interface VPC endpoints (AWS PrivateLink)* [online]. 2020 [cit. 2020-11-04]. Dostupné z: <https://docs.aws.amazon.com/vpc/latest/userguide/vpce-interface.html>.

[19] CLONG. *Detection Lab* [online]. 2017, poslední aktualizace 2020-10-01 [cit. 2020-11-04]. Dostupné z: <https://github.com/clong/DetectionLab/blob/master/README.md>.

[20] LONG Chris. *Introducing: Detection Lab* [online]. Medium, 2017 [cit. 2020-11-04]. Dostupné z: <https://medium.com/@clong/introducing-detection-lab-61db34bed6ae>.

[21] ECKROTH Joshua, CHEN Kim, GATEWOOD Heyley, BELMA Brandon. *ALPACA: Building Dynamic Cyber Ranges with Procedurally-Generated Vulnerability Lattices* [online]. DeLand, Florida, USA. 2019 [cit. 2020-11-04]. Dostupné z: <https://github.com/StetsonMathCS/alpaca/blob/master/publications/acmse-2019-alpaca.pdf>.

[22] SCHREUDERS Cliffe Z., SHAW Thomas, SHAN-A-KHUDA Mohammad, RAVICHANDRAN Gajendra, KEIGHLEY Jason. University of Birmingham. *Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events* [online]. 2017 [cit. 2020-11-04]. Dostupné z: <https://www.usenix.org/conference/ase17/workshop-program/presentation/schreuders>.

[23] CLIFFE. *Security Scenario Generator (SecGen)* [online]. 2014 , poslední aktualizace 2020-04-10 [cit. 2020-11-04]. Dostupné z: <https://github.com/cliffe/SecGen/blob/master/README.md>.

[24] CHUNG Kevin. *CTFd Documentation, Release 2.3.2* [online]. 2020-04-10 [cit. 2020-11-04]. Dostupné z: <https://docplayer.net/183691217-Ctfd-documentation-release-kevin-chung-apr-10-2020.html>.

[25] CTFD. *CTFd Documentation, Release 2.3.2* [online]. 2015 ,poslední aktualizace 2020-09-08 [cit. 2020-11-04]. Dostupné z: <https://github.com/CTFd/CTFd/blob/master/README.md>.

[26] BUSINESSWIRE. *OpenStack Launches as Independent Foundation, Begins Work Protecting, Empowering and Promoting OpenStack* [online]. 2012 [cit. 2020-11-04]. Dostupné z: <https://www.businesswire.com/news/home/20120919005997/en/OpenStack-Launches-Independent-Foundation-Begins-Work-Protecting>.

[27] CLOUDOLOGIC CONSULTING PVT. LTD. *Introduction to OpenStack, Why and deployment models of OpenStack* [online]. In: Issuu [cit. 2020-11-04]. Dostupné z: <https://issuu.com/cloudologic/docs/introduction_to_openstack__why_and_deployment_mode/s/10133378>.

[28] OPENSTACK. *The OpenStack Marketplace, Public Clouds* [online]. [cit. 2020-11-04]. Dostupné z: <https://www.openstack.org/marketplace/public-clouds/>.

[29] OPENSTACK. *DevStack* [online] 2017, poslední aktualizace 2020-06-18 [cit. 2020-11-04]. Dostupné z: <https://github.com/openstack/devstack/blob/master/README.rst>.

[30] OPENSTACK DEVSTACK TEAM. *DevStack Docs* [online]. Poslední aktualizace 2020-05-29 [cit. 2020-11-04]. Dostupné z: <https://docs.openstack.org//devstack/latest/doc-devstack.pdf>.

[31] OPENSTACK. *OpenStack-Ansible Deployment Guide* [online]. Poslední aktualizace 2020-08-18 [cit. 2020-11-04]. Dostupné z: <https://docs.openstack.org/project-deploy-guide/openstack-ansible/latest/index.html>.

[32] MIRANTIS, INC. *Fuel Installation Guide* [online]. ©2005-2020, poslední aktualizace 2018-03-27 [cit. 2020-11-04]. Dostupné z: <https://docs.mirantis.com/fuel-docs/mitaka/userdocs/fuel-install-guide.html>.

[33] OPENSTACK. *TripleO Deployment Guide* [online]. Poslední aktualizace 2019-08-16 [cit. 2020-11-04]. Dostupné z: <https://docs.openstack.org/project-deploy-guide/tripleo-docs/latest/index.html>.

[34] RDO. *Packstack: Create a proof of concept cloud* [online]. ©2020 [cit. 2020-11-04]. Dostupné z: <https://www.rdoproject.org/install/packstack/>.

[35] OPENSTACK. *OpenStack Releases* [online]. Poslední aktualizace 2020-11-03 [cit. 2020-11-04]. Dostupné z: <https://releases.openstack.org/index.html>.

[36] OPENSTACK. *Overview, OpenStack Components* [online]. [cit. 2020-11-04]. Dostupné z: <https://www.openstack.org/software/project-navigator/openstack-components#openstack-services>.

[37] OPENSTACK. *Introduction to OpenStack* [online]. [cit. 2020-11-04]. Dostupné z: <https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html>.

[38] OPENSTACK. *Overview, Deployment Tools* [online]. [cit. 2020-11-04]. Dostupné z: <https://www.openstack.org/software/project-navigator/deployment-tools>.

[39] OPENSTACK. *OpenStack-Ansible, Installation requirements and recommendations* [online]. Poslední aktualizace 2019-03-26 cit. [2020-11-06]. Dostupné z: <https://docs.openstack.org/project-deploy-guide/openstack-ansible/ocata/overview-requirements.html>.

[40] ČELEDA Pavel, ČEGAN Jakub, VYKOPAL Jan, TOVARŇÁK Daniel. Institute of Computer Science, Masaryk University, Brno. *KYPO - A Platform for Cyber Defence Exercises* In: M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence. [online]. 2015 [cit. 2020-11-08]. Dostupné z: <https://is.muni.cz/repo/1319597/2015-NATO-MSG-133-kypo-platform-cyber-defence-exercises-paper.pdf>.

[41] EICHLER Zdenek, Ošlejšek Radek, TOTH Dalibor. Institute of Computer Science, Masaryk University, Brno. *KYPO: A Tool for Collaborative Study of Cyberattacks in Safe Cloud Environment* [online]. 2015 [cit. 2020-11-08]. Dostupné z: <https://is.muni.cz/publication/1308417/full-paper.pdf>.

[42] VYKOPAL Jan, OŠLEJŠEK Radek, ČELEDA Pavel, VIZVÁRY Martin, TOVARŇÁK Daniel. Institute of Computer Science, Faculty of Informatics, Masaryk University, Brno. *KYPO Cyber Range: Design and Use Cases* [online]. 2017 [cit. 2020-11-09]. Dostupné z: <https://is.muni.cz/publication/1386573/2017-ICSOFT-kypo-cyber-range-design-paper.pdf>.

[43] CSIRT, MASARYKOVA UNIVERZITA. *Mezinárodní cvičení Cyber Czech 2018.* [online]. 2018-11-22 [cit. 2020-11-13]. Dostupné z: <https://csirt.muni.cz/about-us/news/cyberczech2018-international>.

[44] DAVIS Jon, MAGRATH Shane. *A Survey of Cyber Ranges and Testbeds* Cyber Electronic Warfare Division. [online]. 2013 [cit. 2020-11-20]. Dostupné z: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a594524.pdf>.

[45] MASARYK UNIVERZITY. *KYPO Cyber Range Platform* [online]. ©2020 [cit. 2020-11-15]. Dostupné z: <https://docs.crp.kypo.muni.cz/installation-guide/>.

[46] BOWEN Rich. *CentOS Project shifts focus to CentOS Stream* In: Blog.CentOS.org [online]. 2020-12-8 [cit. 2021-03-17]. Dostupné z: <https://blog.centos.org/2020/12/future-is-centos-stream/>.

[47] OPENSTACK. *Kolla-Ansible Deployment Guide* [online]. Poslední aktualizace: 2021-01-18 [cit. 2021-03-17]. Dostupné z: <https://docs.openstack.org/project-deploy-guide/kolla-ansible/victoria/index.html>.

[48] MASARYK UNIVERZITY. *Cyber Sandbox Creator* [online]. Poslední aktualizace: 2020 [cit. 2021-04-23]. Dostupné z: <https://gitlab.ics.muni.cz/muni-kypo-csc/cyber-sandbox-creator/-/wikis/home>.

[49] HORVÁTHOVÁ, Estera. *Virtualizační platforma pro bezpečnostní cvičení* Brno, 2021, 77 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Tomáš Lieskovan.

Seznam symbolů a zkratek

AMI	Amazon Machine Image
AWS	Amazon Web Services
CLI	Command Line Interface
CTF	Capture the Flag
DNS	Domain Name System
DOT	Graph Description Language
EC2	Elastic Compute Cloud
EdDSA	Edwards-curve Digital Signature Algorithm
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol
JSON	JavaScript Object Notation
KYPO	Kybernetický polygon
LMS	Learning Management System
LVM	Logical Volume Management
LXC	Linux Containers
NASA	National Aeronautics and Space Administration
NAT	Network Address Translation
NATO	North Atlantic Treaty Organization
NIST	Národní institut standardů a technologie
NÚKIB	Národní úřad pro kybernetickou a informační bezpečnost
OS	Operační systém

OVF	Open Virtualization Format
PXE	Preboot Execution Environment
RDP	Remote Desktop Protocol
RSA	Rivest Shamir Adleman
SSH	Secure Shell
SPICE	Simple Protocol for Independent Computing Environments
UI	User Interface
URL	Uniform Resource Locator
VMDK	VMware Virtual Disks
VMX	Virtual Machine Configuration
VPC	Virtual Private Cloud
VPN	Virtual Private Network
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

A Průběh cvičení ukázkového scénáře

1. Info — 2. Finding... — 3. Connect... — 4. Privile... — 5. Test Ex... — 6. Assessm...

Admin
kypo-admin

00:01:28

Finding open ports

Your goal is to get access to a **server**. You know that there is a **telnet** service running on the server but it is not running on the default port. Your first task is to find the **port** on which the telnet service is running. The flag is the port number.

Below are two options how to connect to the client from which you can connect to the server.

GUI access

1. In the topology overview, click the button in the top-right corner of the graph, then **`Expand All`**, **`client`** and **`Generate console URL`**. After a few moments, **`Open link`** next to the **`Generate console URL`** should appear.
2. Login using username **kypo** and password **kypo**.

SSH from local machine

1. Use the **`Get SSH Access`** button to download **`ssh-access.zip`**.
2. Extract the **`ssh-access.zip`** file to your **`~/.ssh/`** directory.

```
$ unzip ssh-access.zip -d ~/.ssh/
```

3. Execute the extracted source script in the current shell using the **`source`** command with the path to the KYPO proxy SSH private key. The source script that will set the **`ssh`** command and the KYPO proxy SSH private key, which is available from instance operator.

```
$ source ~/.ssh/pool-id-<pool_ID>-sandbox-id-<sbx_ID>-user-source.sh PATH_TO_KYPO_PROXY_PRIVATE_KEY
```

4. Connect to the client to **kypo** user.

```
$ ssh kypo@client
```


Hint 1 | Solution | Flag | Submit

Get SSH Access