**ROMÂNIA**

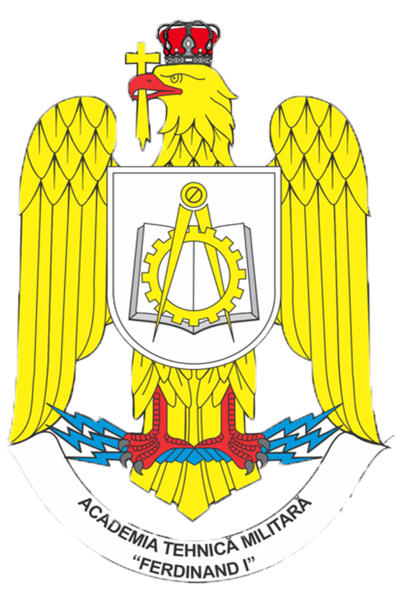
**MINISTERUL APĂRĂRII NAȚIONALE**

**ACADEMIA TEHNICĂ MILITARĂ "FERDINAND I "**

**FACULTATEA DE SISTEME INFORMATICE ȘI**

**SECURITATE CIBERNETICĂ**

**Ingineria și Securitatea Sistemelor Informatice Militare**



**PLATFORMĂ DE TIP CTF UTILIZÂND UN CLUSTER KUBERNETES**

|  |  |
| --- | --- |
| **Coordonator**  Col. prof. dr. ing.  ION BICA | **Absolvent**  Sd. Sg. Maj. Bănică Liviu-Marian |
|  | Conţine \_\_\_\_\_\_\_\_\_\_\_ file  Inventariat sub nr. \_\_\_\_\_\_  Poziţia din indicator: \_\_\_\_  Termen de păstrare: \_\_\_\_\_ |

Abstract

Currently, cybersecurity is a topic of great importance for society. With the increase in the use of technology and the digitization of many activities, the number of cyber threats to which individuals and organizations are exposed has increased. In addition, the COVID-19 pandemic has accelerated the digitization of many processes, further increasing the risk of cyberattacks. One of the main causes of these problems is the lack of cyber security awareness, as many people and organizations do not have a proper understanding of cyber threats and the measures, they must take to protect themselves. An INTERPOL study revealed that in the first four months of 2020, the private sector had approximately 907,000 spam messages, 737 incidents, and 48,000 malicious URLs related to COVID-19. One solution to protect information systems is to have professionals trained in the area. To train them, one method is the use of gamification, interactive activities, and simulations, which allows for improving the performance of the students, the commitment, and the motivation of learning [1 ,2]. In the area of computer security, a common hands-on learning method is Capture the Flag (CTF) competitions, which consist of teams competing to see who can solve the most security problems within a certain time limit [ 3]. As a solution to the lack of cybersecurity knowledge, this thesis proposes the development of a Capture the Flag platform for training one’s cybersecurity skills by practice hands-on. The objective is to provide a tool that enhances the education of future professionals in this field and covers the existing demand for this type of specialist. The platform is developed in from two major component, a web application and a Kubernetes cluster. The first component was developed in Python Flask and uses APIs like Kubernetes API, Helm API for proper handling cluster scope or namespace scope events and helm charts. The second is Minikube which is a lightweight Kubernetes distribution made for local deploymnent, later on this thesis aims to provide ways to develop the solution on production grade Kubernetes distribution. Other components integrated are: ChartMuseum, Docker registry, PostgreSQL database, Kaniko. Several tests were performed which resulted in the compatibility of all the as well as their effectiveness. Tests played a significant role in detecting and resolving bugs and research. Many of the problems identified were simple to solve and focused on the fusion side of the components.

Rezumat

În prezent, securitatea cibernetică este un subiect de mare importanță pentru societate. Odată cu creșterea utilizării tehnologiei și digitalizarea multor activități, numărul amenințărilor cibernetice la care sunt expuse persoanele și organizațiile a crescut. În plus, pandemia COVID-19 a accelerat digitalizarea multor procese, crescând și mai mult riscul de atacuri cibernetice. Unul dintre principalele cauze ale acestor probleme este lipsa conștientizării în ceea ce privește securitatea cibernetică, cu alte cuvinte multe persoane și organizații nu au o înțelegere adecvată a amenințărilor cibernetice și a măsurilor pe care trebuie să le ia pentru a se proteja. Un studiu realizat de către INTERPOL a relevat că, în primele patru luni ale anului 2020, sectorul privat a înregistrat aproximativ 907.000 de mesaje de tip spam, 737 de incidente și 48.000 de URL-uri malitioase legate doar de COVID-19. Una dintre soluțiile pentru protejarea sistemelor de informații este de a avea profesioniști pregătiți în acest domeniu. Pentru a îi instrui o metodă este utilizarea activităților interactive și a simulărilor, ceea ce rezultă în îmbunătățirea performanței studenților, implicarea și motivația în procesul de învățare[1,2]. În domeniul securității cibernetice, o metodă comună de învățare practică este reprezentată de competițiile "Capture the Flag" (CTF), care constau în echipe care concurează cu scopul a rezolva cele mai multe probleme de securitate într-un anumit interval de timp[3]. Ca soluție a problemei conștientizării pericolului unui atac cibernetic, această lucrare propune dezvoltarea unei platforme CTF pentru formarea abilităților de securitate cibernetică prin practică. Obiectivul este de a oferi o unealtă care să îmbunătățească educația viitorilor profesioniști în acest domeniu și să acopere cererea existentă pentru acest tip de specialiști. Platforma implică două componente majore: o aplicație web și un cluster Kubernetes. Prima componentă a fost dezvoltată în Python Flask și utilizează API-uri precum Kubernetes API, Helm API pentru gestionarea corespunzătoare a evenimentelor, la nivel de cluster sau namespace, și pentru gestionarea și prelucrarea Helm Chart-urilor. A doua componentă este Minikube, o distribuție Kubernetes de mici dimensiuni și concepută pentru dezvoltarea locală, ulterior lucrarea propune implementări soluției peste o distribuție Kubernetes de producție. Alte componente relevante sunt: ChartMuseum, Docker Registry, PostgreSQL, Cert-Manager. Au fost efectuate mai multe teste care au condus la compatibilitatea tuturor componentelor și la eficacitatea lor. Testele au jucat un rol semnificativ în detectarea și rezolvarea bug-urilor și cercetare. Multe dintre problemele identificate au fost ușor de rezolvat și s-au concentrat pe partea de integrare a componentelor.

Cuprins

[Abstract 2](#_Toc138698729)

[Rezumat 3](#_Toc138698730)

[Cuprins 4](#_Toc138698731)

[Introducere 5](#_Toc138698732)

[1.1 Importanța temei 6](#_Toc138698733)

[1.2 Motivația și scopul proiectului 6](#_Toc138698734)

[1.3 Rezumatul lucrării pe capitole 7](#_Toc138698735)

[Spațiul cibernetic 8](#_Toc138698736)

[2.1 Definiții 8](#_Toc138698737)

[2.2 Abordări existente: 10](#_Toc138698738)

[3.1 Kubernetes 13](#_Toc138698739)

[3.1.1 Tehnologii 14](#_Toc138698740)

[3.2 Minikube 14](#_Toc138698741)

[3.3 Docker 14](#_Toc138698742)

[3.4 Helm 15](#_Toc138698743)

[3.5 kubectl 15](#_Toc138698744)

[3.6 Chartmuseum 15](#_Toc138698745)

[3.7 PostgreSQL 15](#_Toc138698746)

[3.8 Cert-manager 16](#_Toc138698747)

[3.9 Docker Registry 16](#_Toc138698748)

[3.10 Flask 16](#_Toc138698749)

[Arhitectura sistemului 17](#_Toc138698750)

[4.1 Arhitectura cluster-ului Kubernetes 17](#_Toc138698751)

[4.2 Digrame cazuri utilizare 18](#_Toc138698752)

[4.2.1Diagrama din perspectiva utilizatorului admin 19](#_Toc138698753)

[4.2.2 Diagrama din perspectiva utilizatorului student 20](#_Toc138698754)

[4.2.3 Diagrama din perspectiva sistemului 21](#_Toc138698755)

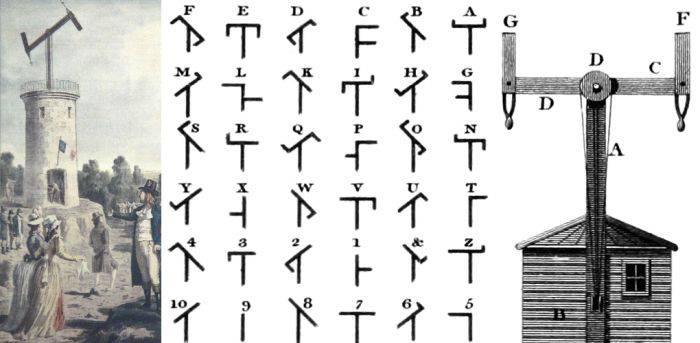
[4.3 Diagrama de activitate 22](#_Toc138698756)

[4.4 Diagrama de secvență 23](#_Toc138698757)

[Referințe 24](#_Toc138698758)

|  |  |  |
| --- | --- | --- |
| Capitolul 1  Introducere | |  |
|  |  | |

În 1834, doi frați francezi, François și Joseph Blanc, au compromis sistemul „semafor" al bursei din Bordeaux pentru a obține avantaje financiare. Planul lor a implicat mituirea unui operator pentru a transmite informații eronate către următorul turn și utilizarea unui intermediar pentru a traduce și oferii informațiile fraților Blanc. Acesta introducea un set de coduri în mesajele transmise de autorități, „deghizate” sub o forma unor erori, ele defapt conțineau informații importante despre acțiunile bursei. Mecanismul sistemului semafor, în cazul în care s-au realizat erori de transmisie, impunea operatorului să incifreze și o erată alături de eroare, mai apoi fiind transmis la turnul următor. Ultima stație avea și scopul de a repara eroarea, aici intervenind ultima piesă a planului fraților Blanc, un complice cu un telescop cu rolul de a citii, traduce și transmite eroarea primită de ultimul turn celor doi frați. Înșelătoria a ținut doi ani, intervenind autoritățile, astfel extrapolând pentru a satisface contextul lucrării, aceștia pot fii considerați ca fiind personalul instruit. Pe scurt, aceștia au reușit să extragă informații vitale bursei franceze prin intermediul acestui atac „cibernetic” prin exploatarea vulnerabilităților sistemului semafor considerat sigur la inaugurare.



Figură 1.1 Imagine reprezentativă pentru sistemul „semafor”

Evenimentul anterior prezentat oferă o perspectivă asupra modului în care oamenii pot utiliza tehnologia pentru a-și atinge obiectivele, chiar și prin metode necinstite. De asemenea, acesta subliniază importanța instruirii și a conștientizării riscurilor cibernetice, pentru a preveni astfel de atacuri.În aceeași manieră, numărul de soluții software prezente este intr-o continuă, rapidă creștere, iar securitatea și mentenanța lor este primordială, astfel nevoia educației în domeniul securității cibernetice și instruirea în condiții simulate cât mai aproape de un scenariu real al unui atac cibernetic este necesară, indiferent că este vorba de o instituție de stat sau de o companie privată plecând de la premiza că nici una nici alta nu este imună. Cu alte cuvinte, apariția unei soluții, fie ea în Internet sau în rețeaua restrânsă a unei instituții, pune o „țintă pe spatele” acesteia și nu poate fii „ștearsă”,ci doar prevenită într-o manieră proactivă.

## 1.1 **Importanța temei**

În contextul actual al securității cibernetice, importanța unei platforme CTF pe infrastructură Kubernetes devine tot mai evidentă. Amenințările cibernetice evoluează rapid, iar specialiștii în securitatea cibernetică trebuie să fie pregătiți să facă față acestora. Prin utilizarea unei platforme CTF bazate pe Kubernetes, studenții și profesioniștii în securitatea cibernetică pot simula și evalua diverse scenarii de atac și vulnerabilități într-un mediu controlat și securizat.

Un studiu NIST[5], CR-urile sunt reprezentări interactive și simulate ale rețelei locale, sistemului, instrumentelor și aplicațiilor unei organizații, care sunt conectate la un mediu simulat. Ele furnizează un mediu sigur și legal, folosit pentru dobândirea ablităților practice în domeniul securității cibernetice și un mediu securizat pentru dezvoltarea de produse și testarea posturii de securitate. Un cyber range poate include atât componente hardware, cât și software reale sau virtuale[6].

Soluția își propune să ofere un cadru sigur, structurat și intuitiv pentru un student și totodată să satisfacă cerințele unui autor de exerciții CTF prin flexibilitate, autorizare utilizatorilor și grupurilor, automatizarea procesului de creare a unui CTF și izolarea a mediilor de lucru prin creearea de namespace-uri unice.

## 1.2 **Motivația și scopul proiectului**

Motivația proiectului a fost determinată de necesitatea unei soluții avansate și eficiente pentru instruirea în securitatea cibernetică, care să integreze tehnologii moderne și să ofere un mediu sigur și scalabil. Există o cerere crescută pentru platforme CTF care permit studenților și profesioniștilor în securitatea cibernetică să se antreneze în simulări de atacuri și să își dezvolte abilitățile practice. Cu toate acestea, există o lipsă de soluții robuste care să combine avantajele Kubernetes și framework-urilor web.

Având în vedere creșterea interesului și a amenințărilor din domeniul securității cibernetice, acest proiect reprezintă o contribuție semnificativă la cercetarea și dezvoltarea soluțiilor software.

Prin utilizarea tehnologiei Kubernetes, se pune accent pe scalabilitate, securitate și administrare optimizată a resurselor, toate acestea fiind deosebit de relevante în contextul soluțiilor software cu un nivel ridicat de risc de penetrare. Astfel obiectivul lucrării este reprezentat de dezvoltarea plaformei și de cercetarea uneltelor folosite, a câtorva alte unelte open-source relevante, care, în contextul temei lucrării și a soluției propuse, ar reprezenta o îmbunătățire semnificativă.

## 1.3 Rezumatul lucrării pe capitole

Lucrarea este compusă din șapte capitole. În primul capitol se realizează o introducere în domeniul securității cibernetice cu ajutorul unor statistici oficiale astfel este adus în prim-plan importanța și actualitatea temei abordate atât în spațiul academic, la nivelul corporațiilor cât și în cel al instituțiilor statelor. Tot aici se regăsesc și obiectivele lucrării de față.

În cel de al doilea capitol sunt prezentate câteva abordări existente de platforme CTF dezvoltate pe infrastructură Kubernetes și alte soluții existente relevante.

În capitolul trei sunt descrise tehnologiile folosite și cele cercetate, cele din urmă fiind considerate cu adevărat o îmbunătățire a platformei de față. Aici se regăsesc informații în detaliu legate de utilizabilitatea uneltelor, specifiicații tehnice extrase din documentația oficială, cât și concluzii personale legate de importanța acestora.

Cel de al patrulea capitol expune arhitectura întregii platforme, diagrame relevante înțelegerii și vizualizării soluției propuse cât și ce a componentelor din care este formată, realizările lucrării și cerințe non-funcționale.

Capitolul cinci sunt prezentate etapele parcurse în vederea configurării, instalării și dezvoltării platformei astfel: detalierea specificațiilor hardware, instalarea și configurarea Minikube, detalierea YAML-urilor și Helm Chart-urilor aferente componentelor integrate, detalierea exercițiilor CTF care se regăsesc pe platformă, dezvoltarea aplicației web.

Capitolul șase prezintă procesele de testare folosite, astfel aici este prezentată testarea modulelor gradual, de la funcții simple la întreaga platformă. Tot aici se regăsesc erori întâmpinate în timpul dezvoltării soluției și rezolvările acestora.

În capitolul șapte este concludată lucrarea printr-un scurt rezumat și prin secțiune unde enunțiez propuneri de direcții ulterioare în dezvoltarea soluției și cercetării.

|  |  |
| --- | --- |
| Capitolul 2 |  |
| Spațiul cibernetic |  |

Utilizarea termenului de spațiu cibernetic, sau cyberspace, a devenit o platitudine cotidiană. Numărul utilizatorilor de platforme digitale la nivel mondial acoperă peste patru miliarde de oameni, aproximativ 60% din populația globului[8]. Nu este imposibil de imaginat o lume complet digitalizată în orizontul următorilor 25-30 de ani și, în niciun caz, scenariul nu mai pare desprins dintr-o lucrare science fiction.

O caracteristică importantă a spațiului cibernetic este constituită de universalitatea sa, în prezent, toate aspectele vieții noastre sunt dominate de cyber. Fenomenul poate fi observat și în societățile cunoscute anterior ca fiind „țări în curs de dezvoltare”, care intră în sfera cibernetică datorită dezvoltării extrem de rapide a rețelelor de telefonie wireless. Dar mișcarea este și mai evidentă în societățile dezvoltate, unde acoperirea cibernetică este mult mai semnificativă și mult mai rapidă. Într-adevăr, nu putem reduce cyber space-ul doar la internet, deoarece toate rețelele interconectate, publice sau private, indiferent de mijloacele de interconexiune ale acestora (fire, fibre, unde de proximitate, unde provenite de la sateliți etc.) alcătuiesc spațul cibernetic.

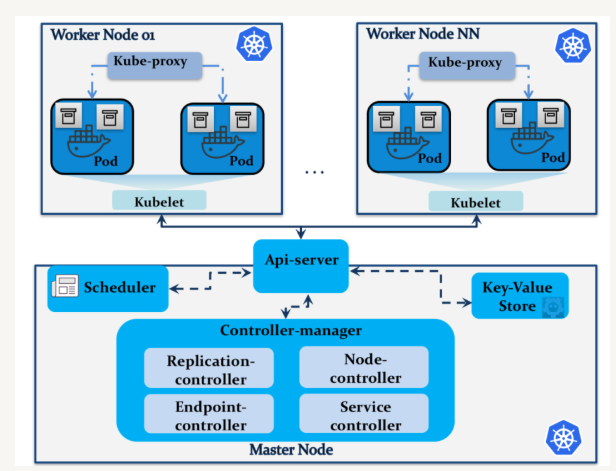
Un studiu al celor de la Cybersecurity Ventures prezice ca până în 2025 vor fii peste trei milioane de poziții vacante, o creștere semnificativă față de un milion de poziții vacante cum a fost raportat în 2014. O mare parte a problemei cu care se confruntă industria noastră este deficitul de competențe avansate care pot fi învățate doar din experiența din lumea reală.

2.1 Definiții

O platformă CTF este un termen general folosit pentru a descrie un sistem care permite utilizatorilor să învețe cum să atace și să se apere în spațiul cibernetic prin intermediul unor exerciții și challenge-uri ce imită scenarii reale. Cei care învață își pot îmbunătăți eficient cunoștințele despre atacuri și apărări, precum și abilitățile de răspuns la incidente[8].

Folosiți interschimbabil soluției cyber range" provine dintr-o formulare militară care descrie un spațiu în care trupele sunt trimise pentru a-și îmbunătăți abilitățile de luptă prin intermediul unor exerciții realiste și bine planificate, care pot implica interacțiuni cu arme și muniții, tancuri, avioane de război, nave de război și altele. În acest fel, un soldat poate să se antreneze într-un mediu care replică condițiile de luptă. Similar, un mediu cyber range oferă personalului din domeniul rețelelor și tehnologiei IT o platformă realistă pentru antrenament. În ceea ce privește scenariile de atac și apărare în cadrul rețelelor, acestea transpun situațiile reale într-un laborator, astfel încât să reproducă cât mai fidel o luptă reală. Antrenamentul într-o se concentrează pe evaluarea situațiilor și aplicarea corectă a configurațiilor în scenarii specifice de atac din viața reală.

Orchestrarea în mediul cyber range și în tehnologiile de virtualizare se referă la configurarea automată a sistemelor informatice și a software-ului. Aceasta implică gestionarea fluxurilor de lucru automate, inclusiv configurarea unui număr mare de mașini virtuale sau containere Docker, crearea, modificarea și ștergerea acestora. Orchestrarea este o componentă integrată în mediile cyber range, care rămâne în mare parte invizibilă pentru utilizatori. Absența capacităților de orchestrare într-un mediu cyber range ar necesita eforturi manuale suplimentare și, implicit, costuri suplimentare pentru utilizatorii platformei.

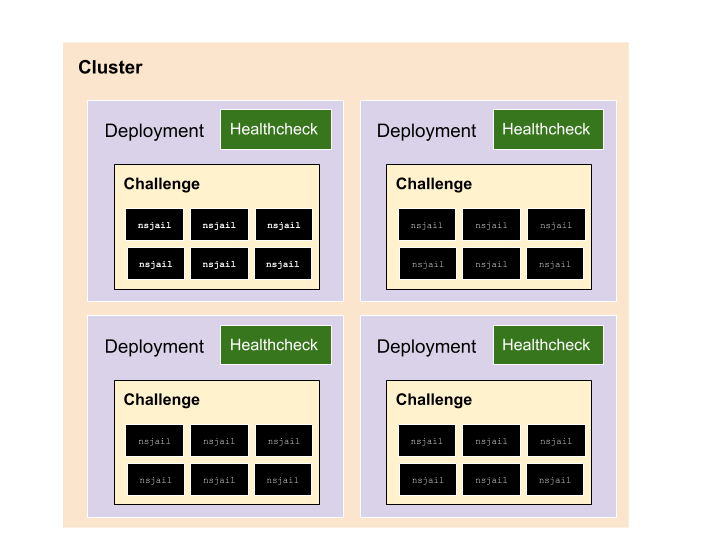


Figură 2.1 Arhitectură Cluster Kubernetes cu un Master Node și două Worker Nodes

2.2 Abordări existente:

Ultimii ani au raportat o cerere în creștere de cyber range-uri cu utilizabilitate extinsă, nu doar pentru pregătirea profesioniștilor, ci și pentru creșterea gradului de conștientizare în domeniul securității cibernetice în rândul publicului larg[9]. Astfel, mai jos sunt prezentate câteva soluții software dezvoltate deja:

* **kCTF** – este o platformă CTF, open source, complexă, dezvoltată peste Kubernetes ce folosește nsjail pentru izolarea mediilor de lucru al jucătorilor, Docker și Kubernetes. Un cluster kCTF este reprezentat de mai multe challenge-uri, fiecare challenge fiind un deployment în Kubernetes. Fiecare deployment conține in componența sa două containere: unul ce realizează healtcheck-uri ale deployment-ului, iar celălalt conține challenge-ul în sine. Un aspect remarcabil constă în crearea unui nou mediu de lucru, pod nsjail, cu fiecare conexiune TCP reușită, astfel se realizeză izolarea participanților[10].



Figură 2.2 Arhitectură kCTF

VEZI LA IMAGINE SI LA TABELUL DE IMAGINI

* **Hack The Box** – utilizează scenarii preluate din lumea reală și furnizează un mediu de simulare care permite participanților să se angajeze. Aceștia oferă o soluție cloud ce rulează 24/7 și inovează prin un tip exercițiu, denumit mașini, la care un utilizator se poate conecta folosind VPN.
* **CTFd** – este o platformă open source cu o documentație concisă. Structura platformei este intuitivă și modular. În aceeași manieră, aplicația oferă numeroase opțiuni de extindere a funcționalității, cum ar fi configurarea unui server SMTP, Swagger UI, autentificare folosind SSO(Single Sign On).
* **CyberTalents** – această platformă este gazda a peste 100 de competiții CTF. Aceasta nu este open source și nu oferă suport pentru exerciții attack and defense.
* **Root the Box** – Aceasta nu este întocmai o platformă CTF, mai mult un motor de scoring ce poate fi implementat folosind Docker și gazduită pe AKS.
* **FacebookCTF** – platforma dezvoltată de Facebook a devenit open source recent și poate găzdui două categorii de CTF-uri Jeopardy și King of the hill. Această platformă dispune de o interfață complexă, captivantă și inovatoare din multe puncte de vedere, un alt aspect inovator este modul de alegere al exercițiilor, acestea fiind sunt marcate ca puncte pe harta lumii.
* **Cyberskyline** – platformă completă, dar este disponibilă doar companiilor private, iar testarea unui demo se face prin comandă.

Mai jos este ilustrat un tabel care oferă o imagine de ansamblu asupra cerințelor funcționale ale fiecărei platfome:

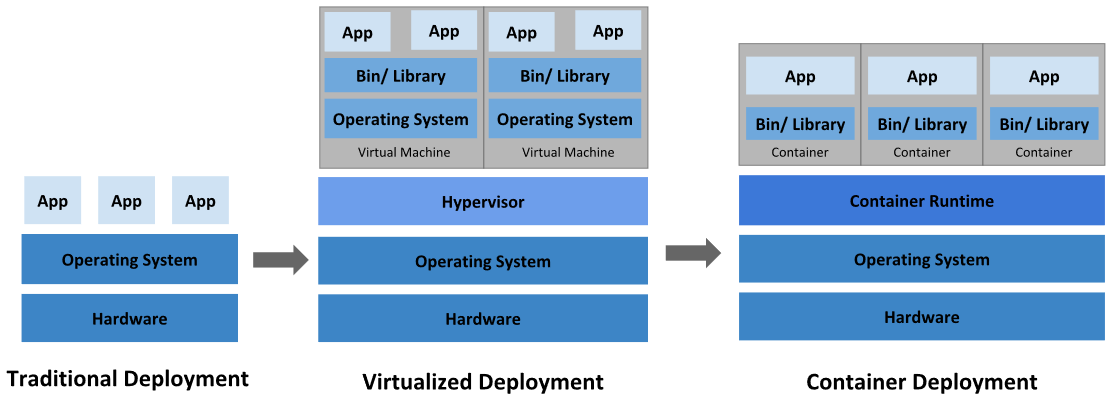
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | kCTF | HTB | CTFd | Cyber  Talents | RTB | Facebook | Cyber  skyline |
| Free | X |  | X |  | X | X |  |
| Scoring | X | X | X | X | X | X | X |
| Cloud  Hosting | X | X | X | X | X | X | X |
| Docker | X | X | X | X | X | X | X |
| Jeopardy | X | X | X | X |  | X | X |
| Attack-Defense |  | X |  |  |  |  | X |

Tabel 0.1 Tabel reprezentativ

|  |  |
| --- | --- |
| Capitolul 3  Tehnologii folosite |  |

În acest capitol vor fi prezentate tehnologiile folosite dezvoltării soluției începând cu tehnologiile folosite pentru organizarea și structurarea cluster-ului Kubernetes, mai apoi continuând cu cele utilizate la dezvoltarea aplicației web.

3.1 Kubernetes



Figură 3.1 De la metodele tradiționale de dezvoltare la Kubernetes

Kubernetes este un sistem de orchestrare open-source, folosit pentru management, plasarea, scalarea și rutarea containerelor. Platforma Docker oferă un mediu Kubernetes securizat și foarte performant, acesta este potrivit pentru atingerea unor scopuri comune în dezvoltarea software, și oferă maximă flexibilitate. Atunci când se construiește o arhitectură bazată pe microservicii este necesară agregarea tuturor microserviciilor. Kubernetes oferă numeroase abstractizări și API-uri pentru a permite acest lucru[11]. Departamentele de informatică și știința calculatoarelor ale universităților din Oslo, respectiv Turin au concludat cercetarea “A formal Model of the Kubernetes Container Framework”[12] ce are ca scop primar să ajute dezvoltatorul să contureze o strategie de implementare a sistemelor bazate pe microservicii, cum este și soluția de față. Aceștia compară diferite strategii de implementare și testează modelul pe infrastructura HPC4AI[13] din Turin.

## 3.1.1 Tehnologii

* Pod – reprezintă cea mai mică unitate care poate fi lansată peste Kubernetes, aceasta fiind compusă din unul sau mai multe containere.
* Deployment – reprezintă o resursă care definește cum trebuie să fie create și rulate unul sau mai multe pod-uri pentru o aplicație specifică. Aceasta include specificații precum: numărul de replici, imaginea folosită, strategii de scalare.
* Serviciu – reprezintă calea prin care un deployment poate interacționa cu alte servicii și microservicii din interiorul sau din afara clusterului. Serviciile pot fi de trei tipuri: NodePort, ClusterIP și LoadBalancer. În cadrul aplicației au fost folosite primele două.
* Ingress – reprezintă resursa Kubernetes care permite gestionarea traficului de intrare. Există și Egress care permite gestionarea traficului de ieșire.
* Secret – reprezintă obiectul care permite stocarea și gestionarea informațiilor sensibile cum ar fi parole, certificate, token-uri de autentificare.
* Namespace – reprezintă un mecanism care are rolul de a izola resurse și grupuri de resurse, practic fiecare namespace reprezintă un “cluster” în cluster-ul în care sunt definite.

3.2 Minikube

Minikube este o unealtă open-source care permite rularea locală a unui cluster Kubernetes cu un singur nod. Acesta se integrează bine cu instrumentele de dezvoltare populare cum ar fi Helm, Docker, kubectl. Alte unelte populare open source care permit rularea Kubernetes sunt: Microk8s, k3s, kind. Alegerea Minikube ca distribuție Kubernetes a fost facută din două considerente și anume lipsa experienței ca dezvoltator DevOps și gama largă de unelte open source intregabile cu acesta.

3.3 Docker

Această unealtă abstractizează infrastructura la nivel de aplicaţie spre deosebire de aplicaţiile monolit care necesitau virtualizarea. De asemenea, Docker implică un consum mai mic de resurse de memorie şi spaţiu in cloud. Un container Docker este un format standard în care sunt împachetate toate codurile și dependențele unei aplicații, pentru rularea rapidă și fiabilă a acesteia în mai multe medii de calcul. Acestea au devenit din ce în ce mai populare pe măsură ce organizațiile trec la dezvoltarea cloud-native și a mediilor hibride multicloud.

3.4 Helm

Tehnologia Helm a fost de asemenea folosită în cadrul proiectului, ea este o unealtă care automatizează crearea, împachetarea și configurarea resurselor Kubernetes prin combinarea lor într-un singur pachet, numit Helm Chart. Un release reprezintă o instanță a unui Helm Chart ce rulează în Kubernetes. Acestuia îi este atribuit, printre altele, și o versiune care este incrementată automat la fiecare upgrade și decrementată la fiecare rollback, astfel Helm ajută și la gestionarea versiunii microserviciilor. În cadrul aplicației a fost folosită versiunea trei Helm, aceasta fiind cea mai recentă[14].

3.5 kubectl

Utilitarul kubectl permite dezvoltatorilor să comunice cu clusterele Kubernetes și să execute diverse operații, cum ar fi crearea, actualizarea și ștergerea resurselor Kubernetes. Acesta este un utilitar în linie de comandă și oferă o metodă de a controla și gestiona containerele, podurile, serviciile, volumele și multe alte componente ale unui cluster Kubernetes. Mai mult, cu ajutorul kubectl s-au realizat procesele de depanare de erori ale soluției prin acțiuni preimplementate cum ar fii *logs* sau *describe* și prin stabilirea unei comunicării SSH între pod-ul pe care rulează aplicația și terminalul mașinii host.

3.6 Chartmuseum

Open source, această unealtă scrisă în limbajul de programare Go și este un server ce oferă Helm chart-urilor un loc unde pot fi stocate și gestionate privat, un repository. Helm Chart-urile din cadrul aplicației sunt administrate prin interogări de tip GET, POST, DELETE către API-ul Chartmuseum[15].

3.7 PostgreSQL

Cunoscută și ca Postgres, acesta este un sistem de gestionare a bazelor de date ce suportă atât interogări în limbajul SQL cât și NoSQL, în cadrul aplicației este creată o bază de date relațională.

3.8 Cert-manager

Acesta este un set de unelte care adaugă certificate și emitenți de certificate ca și CRDs Kubernetes, astfel este simplificat procesul de obținere, reînnoire și folosire a acestora și se asigură traficul HTTPS pe tot ciclul de viață a soluției.

3.9 Docker Registry

Docker Registry este un sistem de gestionare și depozitare a imaginilor Docker. Oferă control asupra imaginilor Docker folosite în fluxul aplicației prin versionare cu ajutorul tag-urilor, suportă trafic TLS și este open source.

3.10 Flask

Flask este un framework Python utilizat pentru dezvoltarea aplicațiilor web. Acesta este clasificat ca un micro-framework, datorită componentelor sale minimale și ușor de utilizat. Flask este cunoscut pentru simplitatea sa, dar și pentru flexibilitatea pe care o oferă dezvoltatorilor în construirea aplicațiilor web, chiar și în cazul proiectelor cu cerințe complexe. Mai mult, Flask oferă și o bibliotecă integrată bazată pe Jinja2 pentru gestionarea template-urilor în aplicațiile web. Aceasta permite dezvoltatorilor să împartă și să refolosească componente de interfață între diferite pagini, ceea ce îmbunătățește lizibilitatea și flexibilitatea codului creat.

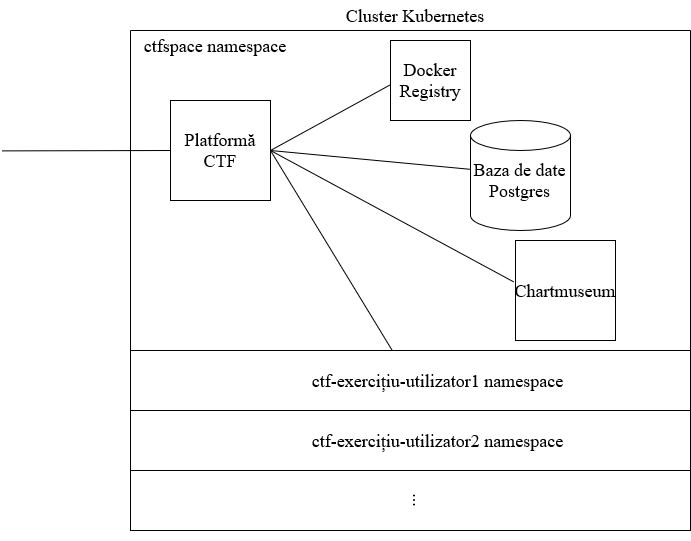
|  |
| --- |
| Capitolul 4 |
| Arhitectura sistemului |

Conform rezumatului lucrării pe capitole, scopul capitolului ce urmează este de a contura o imagine de ansamblu în ceea ce privește infrastructura peste care a fost dezvoltată platforma web, prezentarea fluxului aplicației prin diagrame UML și enunțarea cerințelor funcționale și non-funcționale îndeplinite de aceasta, folosind tehnica de prioritizare MoSCoW.

Contextul lucrării implică dezvoltarea unui mediu de rețea sintetic care să simuleze un context real de dezvoltare, bazat pe comportamentul unor componente reale ale unei rețele. Într-un mediu de simulare tipic dezvoltat în cloud, containerele replică un server specific, rețea și stocare a unei anumite infrastructuri IT.

## 4.1 Arhitectura cluster-ului Kubernetes

Mașina virtuală pe care s-a dezvoltat cluster-ul are în componența sa o imagine docker ce rulează minikube. Comunicarea cu platforma CTF s-a realizat prin Ingress, iar comunicarea între componentele cluster-ului se realizează prin asignat automat sau kubeDNS. Mai mult, componentele interne nu sunt vizibile din afara cluster-ului și nici nu permit comunicare cu acesta, o caracteristică ce aduce un plus la capitolul securitatea sistemului. Pod-urile ce rulează containere vulnerabile, menite pentru a fi atacate pentru că doar astfel se poate rezolva exercițiul, rulează în spații de nume diferite, unice între ele și nu permit comunicarea bidirecțională cu spațiul în care rulează platforma.

 *Figură 1 reprezentativă cu elementele componente*

## 4.2 Digrame cazuri utilizare

Pentru a putea reprezenta grafic, m-am folosit de UML(Unified Modeling Language) care este un limbaj standard pentru descrierea de modele și specificații software. Dezvoltat de Object Management Group(OMG), acest limbaj este dezvoltat pentru arhitecți și dezvoltatori de sisteme IT și nu numai. Diagramele UML sunt clasificate astel:

* De structură
* De comportament
* De interacțiune

Diagramele de cazuri de utilizare, sunt un tip diagrame UML(Unified Modeling Language) dedicate descrierii funcțiilor sau serviciilor oferite de un sistem.

### 4.2.1 **Diagrama din perspectiva utilizatorului admin**

Utilizatorul admin reprezintă entitatea care deține sistemul, acesta are acces la intreaga infrastructură, poate creea alți utilizatori cu drepturi de admin asupra platformei și utilizatori normali(denumiți student), creează și instalează exerciții CTF peste cluster. În aceeași manieră, el poate creea grupuri pentru o gestionare mai ușoară a autorizării asupra unui anumit exercițiu,

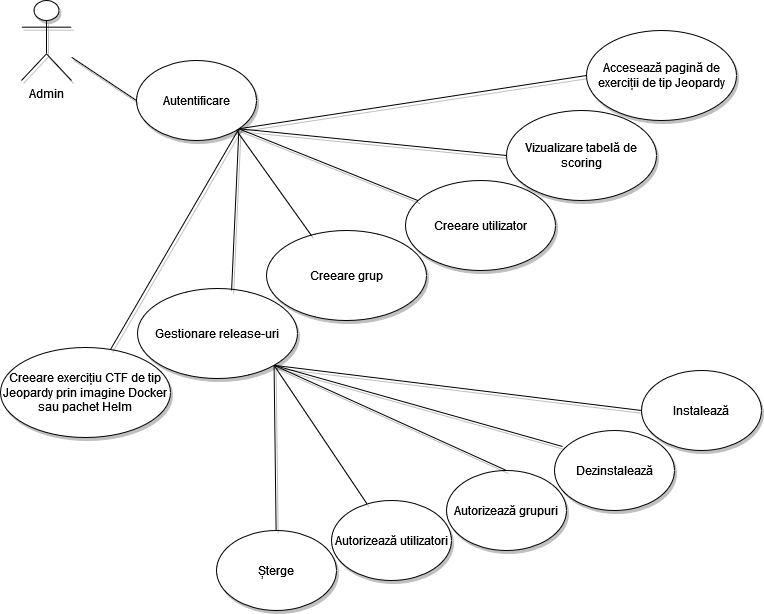


Figure 2 Figură reprezentativă pentru perspectiva administratorului de sistem

### 4.2.2 **Diagrama din perspectiva utilizatorului student**

Utilizatorii student reprezintă utilizatorul țintă alaturi de utilizatorii cu drepturi de admin, acesta poate să se autentifice cu credențialele primite de la un utilizator cu drepturi de admin, având posibilitatea în pagina de profil să și le personalizeze. El poate accesa doar exercițiile la care este autorizat, astfel s-a vrut să se implementeze ideea de mediu privat de învățare.

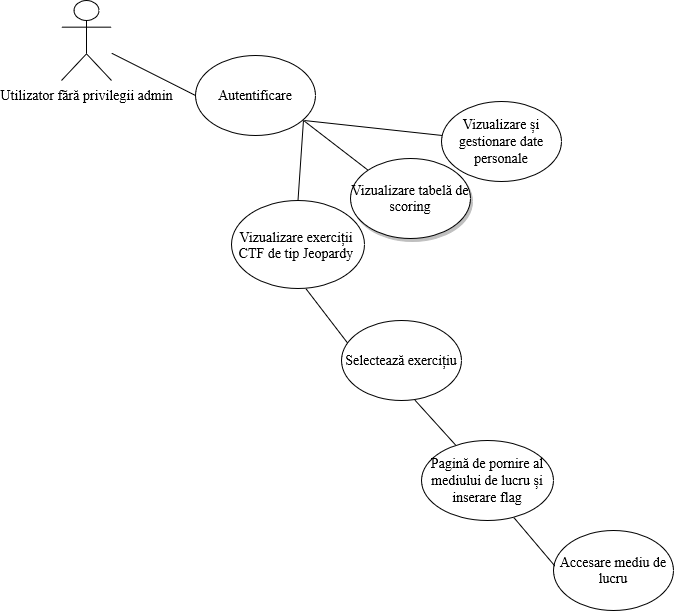


Figure 3 Figură reprezentativă pentru utilizatorul student

### 4.2.3 **Diagrama din perspectiva sistemului**

Sistemul oferă, pe lângă funcționalitățile ce se regăsesc mai sus, o gestionare a resurselor Kubernetes prin: templatizarea resurselor, creearea namespace-urilor unice, accesul și gestionarea resurselor din interiorul altor namespace-uri. Accesul la pod-uri și servicii din interiorul cluster-ului, adăugarea, vizualizarea și ștergerea au putut fi realizate doar dacă platforma, rulând de asemenea pe un pod, se dovedea a fi autentificată în fața API-ului Kubernetes prin RBAC(Role-Bassed Access Control).

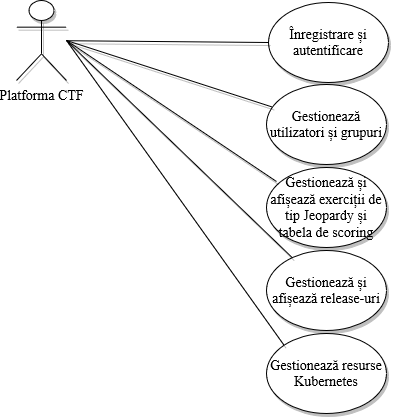


Figure 4 Figură reprezentativa pentru perspectiva sistemului

## 4.3 Diagrama de activitate

Diagrama de activităţi este o variantă a diagramei de stare şi este folosită pentru a modela dinamica fluxului sau a unei operaţii. Diagramele de activitati scot în evidenţă controlul execuţiei de la o activitate la alta.

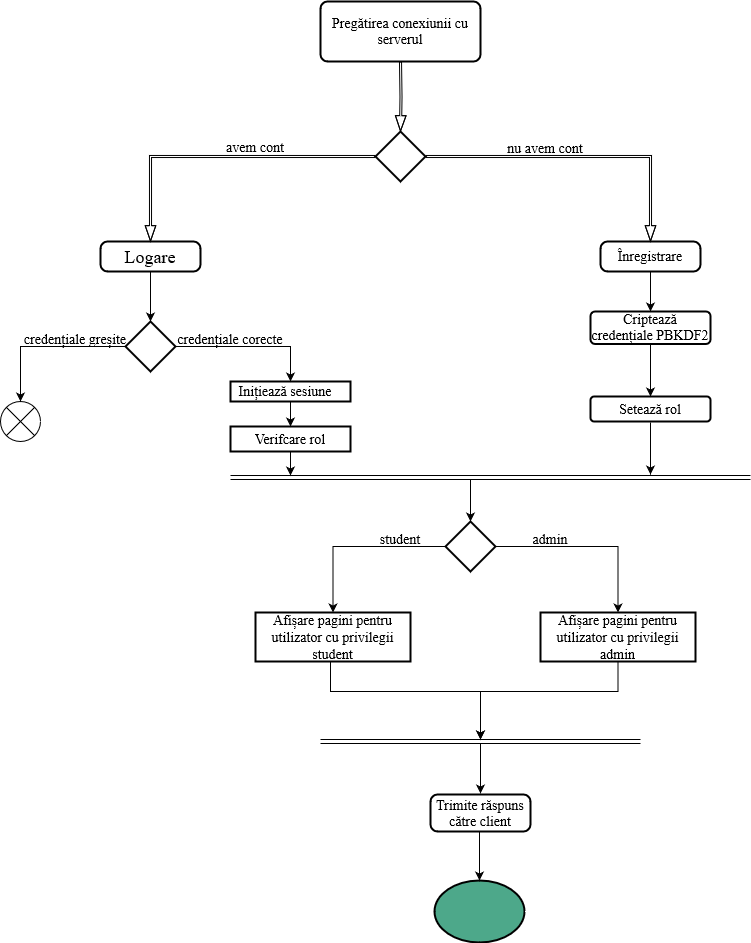
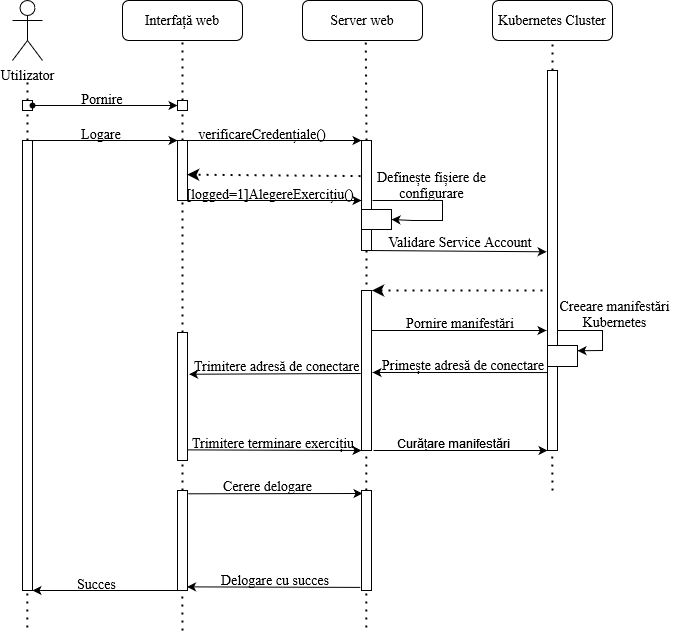


Figure 5 Figură reprezentativă pentru activitatea sistemului

## 4.4 Diagrama de secvență

Diagrama de secvență sau de colaborare are menirea să prezinte un caz de utilizare sau un scenariu particular dintr-un caz de utilizare. Perspectiva oferită de o astfel de diagramă aduce în prim plan, pe lângă scenariu, ciclul de viață al obiectelor create în timpul desfășurării evenimentului. Dimensiunea temporală este reprezentată prin linia verticală trasată sub numele obiectului. Astfel, elementele ce constituie o diagramă de secvență sunt: obiectele, stimuli ce reprezintă apeluri de funcții sau semnale și linia de viață a obiectului.



Figură 6 Diagramă de secvență

## 4.5 Specificații tehnice

Modelul Waterfall a fost ales ca model de dezvoltare, deoarece parcurgerea secvențială a cerințelor oferă o direcție oarecum stabilă. În aceeași manieră, succesul unei sarcini îndeplinite ajută la creșterea stimei de sine și încrederii. Pe de altă parte, acest model are un dezavantaj major și anume sensul unidirecțional al dezvoltării, dat de complexitatea revenirii la versiuni anterioare.

În ceea ce privește cerințele funcționale și non-funcționale ale sistemului, s-a ales tehnica de prioritizare MoSCoW în faza de pre-dezvoltare, chiar dacă aceasta este în general folosită la dezvoltarea prin metodologia Agile. Fragmentarea cerințelor importante în cerințe mai simple și prioritizarea acestora în funcție de importanță a fost natural și sunt de părere că fiecare dezvoltator își structurează, în scris sau nu, o ordine a implementării, în funcție de capabilități și considerente personale. Acronimul “MoSCoW” reprezintă categoriile în care au fost împărțite acestea:

* M – Must Have, cerințele enunțate aici reprezintă caracteristici obligatorii ale sistemului.
* S – Should Have, lista de cerințe definite aici reprezintă o imbunătățire a sistemului rezultat în urma implementării soluției obligatorii.
* C – Could Have, aici au fost enunțate cerințele ce ar putea să fie, dar nu au prezentat o prioritate.
* W – Won’t Have, cerințele descrise aici vor fi implementate într-o versiune ulterioară

Astfel, printre realizările lucrării se pot enumera următoarele:

* Securitate, comunicarea componentelor se realizează prin HTTPS, iar credențialele utilizatorilor sunt criptate folosind PBKDF2-HMAC.
* Portabilitate, componentele soluției sunt definite în fișiere de configurare YAML și pachete Helm.
* Izolarea mediului de lucru, realizat prin definirea și creearea namespace-urilor unice pentru fiecare utilizator.
* Platforma CTF permite înregistrarea/logarea utilizatorilor, creeare grupuri, autorizare utilizator/grup pentru rezolvarea exercițiilor, derularea exercițiilor la nivel individual și scoring.
* Definirea și creearea exercițiilor CTF de tip Jeopardy.
* Docker registry și Chartmuseum, pentru stocarea și gestionarea privată a imaginilor Docker, respectiv Chart-urilor Helm.

În aceeași manieră, din punct de vedere al cerințelor non-funcționale:

* Disponibilitate ridicată (aplicația ca și serviciu trebuie să ofere funcționalitate și disponibilitate pentru o anumită perioadă de timp, chiar și în cazul unei defecțiuni)
* Scalabilitate (aplicația trebuie să suporte corect un volum mai mare de trafic, sau de a permite mărirea sau extinderea sa)
* Performanță (deoarece în proiect se prezintă un cyber-range ce se bazează pe containere și rulează diverse aplicații acesta trebuie să fie rapid și receptiv la comenzile date de către utilizatori)
* Aplicația trebuie sa poată fi folosită pentru implementarea cât mai multor tipuri de exerciții CTF (Attack-Defense, King of the Hill)

Referințe

[1] „Fact Sheet - NATO Cyber Defence (April 2021)”,[Online]Available: <https://www.nato.int/nato_static_fl2014/assets/pdf/2021/4/pdf/2104> factsheet- cyber-defence-en.pdf [Accesed 3-Apr-2023]

[2] cyberwiser.eu,[Online]Available:[https://www.cyberwiser.eu/content/what cyber- range](https://www.cyberwiser.eu/content/what%20cyber-%20%20%09range) [Accesed 10-Apr-2023]

[3] National Initiative for Cybersecurity Education (NICE), „The Cyber Range: A Guide”.

[4] „CyExec\_A\_High-Performance\_Container-Based\_Cyber\_Ra (January 2021) ”[Online]Available: [ResearchGate | Find and share research](https://www.researchgate.net/) [Accesed 5-Dec-2023]

[5] „The Cyber Range: A Guide - Guidance Document for the Use Cases, Features, and Types of Cyber Ranges in Cybersecurity Education, Certification and Training” - Prepared by the NationalInitiative for Cybersecurity Education (NICE) Cyber Range Project Team

[6] „Vasileios\_Linardos–MTE1819”

[Online]Available:<https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/13337/Vasileios_Linardos%20-%20MTE1819.pdf> [Accesed 20-Apr-2023]

[7] „What is a cyber range?”

[Online]Available:<https://cybersecurityguide.org/resources/cyber-ranges/> [Accesed 10-May-2023]

[8] [Online] Available:<https://www.statista.com/> [Accesed 5-Dec-2023]

[10] [Online] Available:<https://cve.report/software/google/kctf> [Accesed 15-Jan-2023]

[11] Hightower, K., Burns, B., Beda, J - “Kubernetes: Up and Running Dive into the Future of Infrastructure. O’Reilly (2017)”

[12] “A Formal Model of the Kubernetes Container Framework ?”[Online] Available:https://www.mn.uio.no/ifi/english/research/projects/adapt/news/uioresearchreport-496.pdf [Accesed 29-Apr-2023]

[13] “HPC4AI: an AI-on-demand federated platform endeavour” Aldinucci, M., Rabellino, S., Pironti, M., Spiga, F., Viviani, P., Drocco, M., Guer-zoni, M., Boella, G., Mellia, M., Margara, P., Drago, I., Marturano, R., Marchetto,G., Piccolo, E., Bagnasco, S., Lusso, S., Vallero, S., Attardi, G., Barchiesi, A., Colla, A., Galeazz

[14] [Online] Available:https://helm.sh/docs [Accesed 1-Mar-2023]

[15] [Online] Available:https://chartmuseum.com/docs/ [Accesed 5-Mar-2023]