Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Αυτοματοποιημένη Ανάπτυξη Υποδομών Εκπαίδευσης Κυβερνοασφάλειας στο Υπολογιστικό Νέφος

Διπλωματική Εργασία

του

Ξενοφώντα Ι. Δρακωτού

Επιβλέποντες: Καθηγητής Γεώργιος Καμπουράκης
Δρ. Παναγιώτης Τριμίντζιος, Ερευνητής, ENISA

Εξεταστική επιτροπή: Καθηγητής Γεώργιος Καμπουράκης
Δρ. Παναγιώτης Τριμίντζιος, Ερευνητής, ENISA
Δρ. Μάριος Αναγνωστόπουλος, Ερευνητής, NTNU

1/2021

UNIVERSITY OF THE
AEGEAN

Department of Information and Communication Systems Engineering

MSc STUDIES

INFORMATION AND COMMUNICATION SYSTEMS SECURITY

Automated Cloud Cyber Range Deployments

MASTER THESIS
of
Xenophon I. Drakotos

Supervisors: Professor Georgios Kampourakis
Dr. Panagiotis Trimintzios, Researcher, ENISA

Committee: Professor Georgios Kampourakis
Dr. Panagiotis Trimintzios, Researcher, ENISA
Dr. Marios Anagnostopoulos, Researcher, NTNU

1/2021

Contents

List of Figures

List of Abbreviations

API Application Programming Interface

AWS Amazon Web Services

CaaS Communications as a Service

CLR Common Language Runtime

CPU Central Processing Unit

CSP Cloud Service Provider

DaaS Data as a Service

DDoS Distributed Denial of Service

DevOps Development and Operations

DNS Domain Name Service

EaaS, XaaS Everything as a service

ECSO European Cyber Security Organization

ENISA European Union Agency for Cybersecurity

FaaS Function as a service

FXML EFF-ects eXtended Markup Language

GENI Global Environment for Network Innovations

GUI Graphical User Interface

IaaS Infrastructure as a Service

HPC High-Performance Computing

ICT Information and Communication Technology

IT Information Technology

MaaS Monitoring as a Service

MBaaS Mobile "back-end" as a service

MS Microsoft

MVC Model View Controller

NaaS Network as a Service

NICE National Initiative for Cybersecurity Education

NCR US National Cyber Range

NDA Non-Disclosure Agreement

NIC Network Interface Card

NSG Network Security Group

NIS Network and Information Security

NISD Network and Information Security Directive

NIST National Institute of Standards and Technology

OS Operating System

OT Operational Technology

PaaS Platform as a Service

POM Project Object Model

QoS Quality of Service

RaaS Recovery as a Service

RAM Random Access Memory

Range Cyber Range

RLMS Range Learning Management System

R&D Research and Development

SaaS Software as a Service

SLA Service Level Agreement

SOC Security Operations Centre

STaaS Storage as a Service

UI User Interface

URL Uniform Resource Locator

VM Virtual Machine

vNet virtual Network

VPN Virtual Private Network

XSS Cross-Site Scripting

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές, για την εποικοδομητική συνεργασία και την συμβολή τους στο τελικό αποτέλεσμα.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υπομονή που επέδειξαν, την στήριξη και την κατανόηση τους, καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Με την ταχεία ανάπτυξη της τεχνολογίας σε κάθε επιμέρους τομέα (τηλεπικοινωνίες, υλικό, λογισμικό κλπ.), τα υπολογιστικά συστήματα έχουν αποτελέσει ένα αναπόσπαστο κομμάτι για τη λειτουργία των περισσότερων οργανισμών και εταιριών.

Επιπλέον, ένας μεγάλος αριθμός σύγχρονων, έξυπνων συσκευών έχουν συνδεθεί στο δίκτυο, προσθέτοντας πολυπλοκότητα και πρόσθετα, πιθανώς εκμεταλλεύσιμες ευπάθειες.

Συνεπώς, τα προβλήματα ασφάλειας των υπολογιστικών και επικοινωνιακών συστημάτων αποτελούν τις σημαντικότερες απειλές που αντιμετωπίζουμε σήμερα. Αυτές οι απειλές μας προκαλούν να ανακαλύψουμε νέες και αποδοτικότερες τεχνικές για την εκτίμηση των τρεχόντων και μελλοντικών αναγκών με στόχο την προστασία της υποδομής ενός οργανισμού από εξωτερικές απειλές.

Μία μορφή προληπτικής άμυνας επιτυγχάνεται μέσω της εκπαίδευσης των επαγγελματιών στον τομέα της κυβερνοασφάλειας με σκοπό τη βελτίωση και ενημέρωση τους πάνω σε σύγχρονα θέματα κυβερνοασφάλειας και σε μοντέρνες απειλές. Με αυτόν τον τρόπο θα βελτιώσουν τις τεχνικές τους δεξιότητες και θα είναι σε θέση να προστατεύσουν την υποδομή τους από κακόβουλους χρήστες και εξωτερικούς κινδύνους.

Μία δεύτερη μορφή άμυνας αποτελεί η αύξηση της επίγνωσης στον τομέα της κυβερνοασφάλειας σε μη επαγγελματίες του τομέα και στο ευρύ κοινό. Αυτό μπορεί να επιτευχθεί με την εκτέλεση ασκήσεων κυβερνοασφάλειας και τη διεξαγωγή εκπαιδευτικών προγραμμάτων, με σκοπό την αύξηση της τεχνικής γνώσης και του επιπέδου ετοιμότητας.

Αυτές οι εκπαιδευτικές διαδικασίες απαιτούν ειδικές υποδομές για τη ρεαλιστική προσομοίωση και εκτίμηση των μοντέρνων υποδομών των οργανισμών.

Οι ασκήσεις για ανάπτυξη κυβερνο-ικανοτήτων (Cyber Ranges) είναι πλατφόρμες, οι οποίες εκμεταλλεύονται τις νέες τεχνολογίες (όπως την εικονικοποίηση) και παρέχουν ρεαλιστική, υψηλής πιστότητας, προσομοίωση ενός περιβάλλοντος στο οποίο μπορούμε να διεξάγουμε πειράματα και να εκτιμήσουμε το συνολικό επίπεδο ασφάλειας του.

Παρουσιάζουμε την εργασία των Maximilian Frank, et al., “Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education” [1] και περιγράφουμε το προτεινόμενο μοντέλο κύκλου ζωής, έχοντας ως στόχο το σχεδιασμό και την υλοποίηση πειραματικών υποδομών (testbeds) κυβερνοασφάλειας.

Συγκεκριμένα, βασιζόμαστε στο μοντέλο των συγγραφέων και προτείνουμε μία εμπλουτισμένη επέκταση της, εκμεταλλευόμενοι σύγχρονες τεχνολογίες όπως τους περιέκτες (Containerization) και το υπολογιστικό νέφος με σκοπό να ωφεληθούμε από τα πλεονεκτήματα που αυτά προσφέρουν. Χρησιμοποιούμε το προτεινόμενο μοντέλο κύκλου

ζωής με σκοπό να υλοποιήσουμε έναν αυτοματοποιημένο τρόπο για την ανάπτυξη εκπαιδευτικών Cyber Ranges απευθείας στο υπολογιστικό νέφος και να παρέχουμε την πλήρη υπηρεσία στον τελικό χρήστη με έναν απλό τρόπο.

Το αποτέλεσμα της παρούσας διπλωματικής είναι μία πλήρης εφαρμογή λογισμικού, απευθυνόμενη στον τελικό χρήστη, η οποία μπορεί είτε να προσαρμοστεί, να εμπλουτιστεί με το κατάλληλο περιεχόμενο και να χρησιμοποιηθεί με εκπαιδευτικό σκοπό ή να αποτελέσει την βάση για περεταίρω ανάπτυξη για μελλοντικές πρωτοβουλίες σχετικά με την αύξηση της επίγνωσης στον τομέα της κυβερνοασφάλειας.

Λέξεις κλειδιά: Κυβερνοασφάλεια, εκπαίδευση, επίγνωση, Ασκήσεις για ανάπτυξη κυβερνο-ικανοτήτων, Υπολογιστικό Νέφος, Αυτοματοποίηση

Abstract

With the rapid development of technology in every sector (communications, hardware, software, etc.), information systems have become an important infrastructure for the operation of most modern organizations and enterprises.

Furthermore, a large number of contemporary, intelligent devices are also being connected to the network, adding complexity and additional, possible exploitable, points.

Consequently, information security issues have also become one of the major security threats faced. This new situation challenges for newer, more efficient techniques for assessing current and future needs and protecting an organization's core infrastructure from external threats.

One form of defence is through training cybersecurity professionals, aiming at the improvement of understanding of the well-known and of newer types of threats, thus improving their technical skills and gain the ability to be protected from them.

A second form of defence is to increase cybersecurity awareness on non-security professionals/specialists and the general public. This can be achieved by executing cybersecurity exercises and by conducting training programs, aiming at increasing technical background knowledge and readiness.

As a result, those educational procedures require specific infrastructure and testbeds for the realistic simulation and assessment of modern organizations.

Cybersecurity Ranges are platforms that take advantage of modern technologies such as virtualization and provide a high fidelity and realistic simulation, experimental, environment, which enables one to test and assess the overall security of the simulated information systems.

First off, we present the research work of Maximilian Frank, et al., "Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education" [1] and we describe the proposed life cycle model, to design and implement cybersecurity testbeds.

Based on the work of the authors, we propose the extension of it, utilizing modern technologies such as Containerization and Cloud Computing, as to benefit from their advantageous characteristics. We, therefore, utilize the proposed life cycle design, in order to create an automated way to deploy educational cyber ranges directly at the cloud and offer the service to the end-user, in a hassle free-manner.

The result of this study is a complete end user software application that can either be customised, enriched and be used for educational reasons or as a baseline for further development for future initiatives towards cybersecurity awareness.

Keywords: Cybersecurity, Training, Awareness, Cyber Ranges, Cloud Computing, Automation

1 Introduction

Recent security reports and incidents reveal what is already suspected, a huge increase not only at the severity of security threats, but at their complexity as well. Each new attack is more organized, more sophisticated, uses in depth knowledge of technologies and automations that were created to add value and not to destroy. Multiple malicious tools are developed and sold online, giving the chance to adversaries to gain access to restricted resources and usually, make profit of it.

1.1 EU Policy Context

Consequently, we must develop countermeasures in all possible ways. One of them relies at increasing cybersecurity awareness in the public, as to be prepared to repel possible adversary attacks. One way to achieve this is to conduct cybersecurity exercises. Cyber incident exercises help organizations to estimate their level of resiliency and practice their response and internal/national procedures at a safe environment. Another affected component is the cultivation of a culture of learning at the participants and therefore, maximize their readiness and effectiveness during an incident. To maintain and manage cybersecurity exercises, the Cyber Range concept is proposed.

Additionally, according to EU legislation, as part of the EU Cybersecurity strategy the European Commission proposed the EU Network and Information Security directive. The **NIS Directive** [2] is the first piece of EU-wide cybersecurity legislation. The goal is to enhance cybersecurity across the EU. The NIS directive was adopted in 2016 and subsequently, because it is an EU directive, every EU member state has started to adopt national legislation, which follows or 'transposes' the directive.

According to **NISD** [2], Article 7:

"Each Member State shall adopt a national strategy on the security of network and information systems defining the strategic objectives and appropriate policy and regulatory measures with a view to achieving and maintaining a high level of security of network and information systems and covering at least the sectors referred to in Annex II and the services referred to in Annex III. The national strategy on the security of network and information systems shall address, in particular, the following issues:"

A. the objectives and priorities of the national strategy on the security of network and information systems

B. a governance framework to achieve the objectives and priorities of the national strategy on the security of network and information systems, including roles and responsibilities of the government bodies and the other relevant actors

C. the identification of measures relating to preparedness, response and recovery, including cooperation between the public and private sectors

D. an indication of the education, awareness-raising and training programmes relating to the national strategy on the security of network and information systems

E. an indication of the research and development plans relating to the national strategy on the security of network and information systems

F. a risk assessment plan to identify risks

G. a list of the various actors involved in the implementation of the national strategy on the security of network and information systems

ENISA [3] has been active in a number of NIS areas, such as the cyber exercises. The following are some of the cyber exercising activities where ENISA is involved [4]:

- Cyber Europe programme
- Cyber Exercise Platform (CEP)
- Trainings and studies
- Other cyber exercises supported

In addition to this, according to the **EU Cybersecurity Act** [5], Article 6 (Capacity Building):

ENISA shall assist:

- Member States by regularly organising the cybersecurity exercises at Union level referred to in Article 7(5) on at least a biennial basis and by making policy recommendations based on the evaluation process of the exercises and lessons learned from them
- Relevant public bodies by offering trainings regarding cybersecurity, where appropriate in cooperation with stakeholders

Therefore, the EU member states should conduct cybersecurity exercises at Union level.

1.2 Why Cyber Ranges

From systems and security viewpoint, cyber networks are non-deterministic, complex systems. To address this, one must develop foundational research protocols enabling them to reproduce large scale environments for conducting cyber experiments, which will provide them with deep understanding of a system's security overall status. A tenant of this approach is to create a test environment and make cybersecurity tests on it, revealing further weaknesses or generally allowing us to have an overall security estimation of an infrastructure.

Such large infrastructure environments have been created and called **Cyber Testbeds** or **Cyber Ranges**. Those include national efforts, such as the US National Cyber Range (NCR), which can represent a variety of complex, heterogeneous systems and offers processes to take a cyber experiment from inception to final analysis.

In cyberspace, maintaining a high level of situation awareness is required for supporting decision making. Therefore, a tool such as the Cyber Range is necessary for the security assessment of a complex system. This tool enables us to develop appropriate training processes for developing or evaluating cybersecurity awareness in target audiences. The target audience can vary from students to professionals and specialists.

This thesis starting point is the work of Maximilian Frank et al. [1], which introduces a new design life cycle for testbeds and it is combined with personal knowledge and working experience for utilizing current, up to date, technologies. Our purpose is to extend this model to utilize the cloud and more specifically Microsoft (MS) Azure, providing automation procedures and a complete application to the end-user (trainee) who wishes to acquire in depth knowledge of the offered challenges.

1.3 Thesis Structure

The following chapters introduce some core components required for our work implementation.

This thesis is organized as follows: Some introductory points are given in **chapter 1**. In **chapter 2** and **chapter 3**, we introduce vital technology achievements and basic information for **Cyber Ranges** and **Cloud Computing**. In **chapter 4**, we present our basis of work, key components and the provided testbed design life cycle. In **chapter 5** we introduce our cloud-based design model, we explain the necessity of our work and we validate its life cycle steps, as authors also do. In **chapter 6** we further analyse the implementation requirements and provide a more in detail description of the developed application and its life cycle flow. A model usage example is given in **chapter 7** and a brief description of the produced code and of the back-end view is given in **chapter 8**. Lastly, **chapter 9** presents our conclusions and future work.

1.4 Contribution

The contribution of this work relates to the three entities involved.

The system manager will no longer be worried about the service scalability and will dedicate less time and effort to create, maintain and delete the required cloud resources.

The trainer's work, having to do with managing the current/future challenges is minimal and straightforward.

The trainees are offered an easy to use, all-in-one application, enabling them to quickly deploy the challenges they wish to interact with.

2 Cyber Ranges

We have already referred to cyber ranges (or simply "ranges") as a tool to simulate a complete organization infrastructure that can be used for cyber training and cyber technology development. There is also the possibility to mimic large, complex networks to improve the realism and quality of training. In this section, we will examine more thoroughly the cyber ranges and their specific components and capabilities.

2.1 Definition - Terminology

According to **NIST** [6]:

“Cyber ranges are interactive, simulated representations of an organization’s local network, system, tools, and applications that are connected to a simulated Internet level environment. They provide a safe, legal environment to gain hands-on cyber skills and a secure environment for product development and security posture testing.

A cyber range may include actual hardware and software or may be a combination of actual and virtual components. Ranges may be interoperable with other cyber range environments. The Internet level piece of the range environment includes not only simulated traffic, but also replicates network services such as webpages, browsers, and email as needed by the customer.”

Cyber Ranges can:

- Provide performance-based learning and assessment
- Provide a simulated environment where teams can work together to improve teamwork and team capabilities
- Provide real-time feedback
- Simulate on-the-job experience
- Provide an environment where new ideas can be tested, and teams can work to solve complex cyber problems

The **European Cyber Security Organization** [7] identifies the below list of target users and target entities of a cyber range [8]:

Target Users:

- Corporates (private and government)
- Strategic decision makers (private and government)
- Security professionals

- Military agencies and CNOs
- Security Operations Centres (SOCs)
- Educators
- Students
- Researchers
- Event organisers

Use Cases:

- **Security Testing:** Along with security research, this is the most traditional use case of cyber ranges where system and application simulations are tested and security attacks are carried out against them, in a controlled way, to identify potential vulnerabilities before deployment and use.
- **Security Research**: Cyber ranges are a fundamental means to carry out security research across a wide range of security domains. By their very nature, cyber ranges are themselves being developed by researchers around the world in order to research new attack detection and mitigation methods, malware emulation, and much more.
- **Competence Building:** The majority of security training today is done through online and face to face training courses. In both cases, most of the learning occurs through listening to videos or live lectures, and through reading notes or slides. Relatively little time is spent on hands on learning. The use of cyber ranges changes that as it can provide a convenient and more cost-effective way of delivering hands on training, as well as the associated training assessment and certification. According to Gartner, by 2022, 15% of large enterprises will be using cyber ranges to develop the skills of their security teams, up from less than 1% today [9].
- **Security Education:** Security education specifically refers to academia as opposed to the lifelong learning and training that professionals undergo after they leave university. One of the recurring complaints from industry is the lack of hands on experience by young graduates. The root cause of such a gap is the cost and complexity of providing students with hands on experience, while at the same time not diluting the educational value of university degrees. Universities around the world have begun looking at cyber ranges as a means of improving teaching and learning.
- **Development of Cyber Capabilities:** Cyber capabilities are the resources and assets available to a state to resist or project influence through cyberspace. At a human level, cyber capabilities coincide with the competences of security professionals across a wide range of cyber attack and defense domains. In such context, cyber ranges are part of a country's cyber capabilities and can be used for developing the capabilities of security professionals, for the research and development of cyber tools and other assets, and for the continuous delivery of cyber exercises to test those cyber capabilities. Specifically, cyber ranges allow a country to carry out cyber capability development at a whole different scale

and level of efficiency. Also, within the context of cyber capability development, cyber ranges can be used to organise large scale cyber exercises involving hundreds to thousands of people.

- **Development of Cyber Resilience:** Cyber resilience refers to the capability of an organisation to respond and be able to sustain a security incident or cyber attack while maintaining its ability to deliver its core business services. NIST defines cyber resilience as "the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that include cyber resources" [10]. Gartner defines it as "...the degree of adaptiveness and responsiveness to a threat to or failure of digital business ecosystems" [11]. Overall, cyber resilience applies to any process, system, business and organisation where there is a reliance on IT, OT, IoT which pretty much covers the majority of organisations in a nation, including critical infrastructure.

  In the context of cyber resilience, cyber exercises provide opportunities for organisations to demonstrate critical capabilities and reveal how effectively they integrate people, processes, and technology to protect their critical information, services, and assets.

  Cyber exercises can be divided into 'Capture The Flag' (CTF) and live-fire exercises. CTF are usually organised in attack and defence style where individuals or teams have to find and fix vulnerabilities in their own systems, while simultaneously attacking systems that belong to other participants. Live-fire cyber exercises enable teams to train cyber professionals to detect and mitigate large scale cyber attacks, while being constantly attacked by a "red team" of hackers.

  Cyber exercises provide the opportunity to test an organisation's capability to handle complex cyber incidents involving several organisations at the same time, thus simulating the interaction with subcontractors, service providers, customers, etc. upon which modern organisations depend. Cyber exercises also enable organisations to find gaps and areas for development in their processes, procedures, and technologies. By addressing the findings from exercises, organisations can greatly enhance their cyber resilience against modern cyber attacks.

- **Competence Assessment:** Competence is a set of attributes such as knowledge, skills and abilities required to successfully perform specific tasks. As the security skills gap increases, organisations need an efficient way of assessing and selecting the right personnel. Using cyber ranges can allow organisations to perform competence assessment beyond the traditional tests, based on multiple choice questions or theoretical simulations. Cyber ranges allow the assessment to be practical and based on the successful completion of practical tasks and/or on the observation of user behaviour and choices made in the execution of practical tasks or assignments.

- **Recruitment:** As cyber ranges are used for competence assessment, it is also to be expected that they will change hiring practices allowing organisations to better identify, validate and hire suitable candidates. This application is highly dependent on the development of the national and international competence

frameworks currently being developed around the world.

- **Digital Dexterity:** Digital dexterity, as defined by Gartner, is “the ability and desire to exploit existing and emerging technologies for business outcomes” [9]. A colourful and simplified, yet effective, way of describing the use case of cyber ranges in relation to digital dexterity is to think of them as a development environment on steroids. Traditional software development methodologies and security best practices recommend the use of different environments such as developing, staging and production. With the ongoing digital transformation and the requirements to support multiple communication and business challenges, organisations are being challenged to project those very same traditional best practices across different channels while at the same time supporting faster development lifecycles.
- **National and International Cybersecurity Competitions:** More and more countries are organising national cybersecurity competitions and participating in international ones as a way of discovering new cybersecurity talents and to help fill the security skills gap. Such competitions are typically delivered as CTFs involving a combination of practical challenges. Cyber ranges are changing the way such competitions have been organised allowing for more large scale events and more realistic simulations. Notable examples include the European Cyber Security Challenge organised by ENISA [12], the Word Skills [13], and the CyberStars competition [14].

2.2 Features and Components

According to the **National Initiative for Cybersecurity Education (NICE)** [15], conventional education and training models are insufficient to fill the cybersecurity skills gap. As per **NICE Cyber Range Guide** [16]:

"Cyber ranges provide enabling technology to operationalize, predict, and monitor the training and performance of cybersecurity professionals. Cyber ranges instil confidence in cybersecurity workforce seekers and cybersecurity workforce employers that training will predict job success. Below, the critical features of cyber ranges are being identified as catalysts in closing the cybersecurity workforce skills gap, including technical components, realism & fidelity, accessibility & usability, scalability & elasticity, and curriculum & learning outcomes."

Cyber ranges comprise many parts, but the essential core technological components include [16]:

2.2.1 Range Learning Management System

A central feature for many cyber ranges is the range learning management system. As the name suggests, a range learning management system (RLMS) contains the standard features of an LMS and the unique characteristics of a cyber range.

The **NICE** diagram depicted in figure 2.1 [15] illustrates both the technical components of a range combined with several RLMS features.

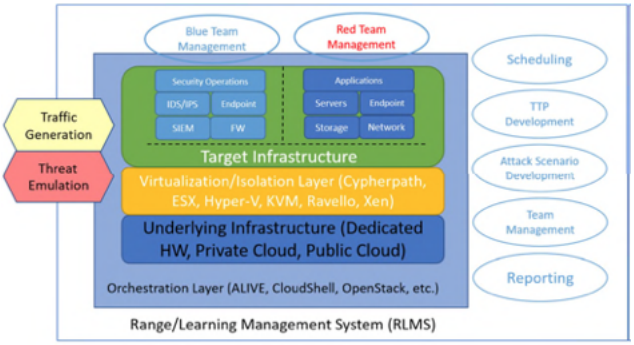

Figure 2.1: RLMS

2.2.2 Orchestration Layer

Taking input from the RLMS, the orchestration layer pulls together all the technology or service components of the cyber range. Many cyber ranges use an in house developed orchestration layer. Some ranges utilize a commercial product for this layer. The orchestration layer can provide "the special sauce" of cyber ranges because it facilitates the meshing of the underlying infrastructure, virtualization or isolation layer, and the target infrastructure. This layer also enables dynamic cyber range extensibility that supports public cloud, private cloud, and dedicated hard wire infrastructures.

2.2.3 Underlying Infrastructure

All cyber ranges are on top of an infrastructure of network, servers, and storage. Some dedicated ranges are built directly on top of physical infrastructure (switches, routers, firewalls, endpoints, etc.) in a rack, though this is typically expensive and not scalable.

For scalability, cost, and extensibility reasons, many range providers are shifting to software-defined virtual infrastructure. Infrastructure drives the realism or fidelity of the cyber range. In addition, a determining factor around infrastructure selection and use centres on how much legacy hardware or software must the cyber range support to meet the client's use cases. In addition, and though not exactly part of the underlying infrastructure, many cyber range employ use cases that require traffic generation and attack emulation.

2.2.4 Virtualization Layer

Most cyber ranges look to some level of virtualization to shrink the physical footprint. Here are two general approaches: hypervisor-based solutions and software defined infrastructure. Regardless of the virtualization approach, the level of disintermediation between underlying physical infrastructure and target infrastructure affects the realism of the cyber range due to unwanted and unpredictable jitter and latency. On the other hand, economically viable cyber ranges would not be possible without this virtualization layer. It also acts as a firewall between the target infrastructure (with associated attack vectors) and the underlying infrastructure (dedicated, public cloud, private cloud).

2.2.5 Target Infrastructure

The target infrastructure is the simulated environment in which students train. Based on the use case, the target infrastructure can in some cases match the student's real-world IT and security infrastructure. Advanced cyber ranges contain profiles of commercially available servers, storage, endpoints, applications, and firewalls. Based on student interaction, the RLMS will generate scripts to instruct the orchestration layer to create the target infrastructure. These scripts might include client-specific configuration information including IP Address ranges, routing information, server stacks, and endpoint software.

2.3 Realism & Fidelity

The accuracy with which the cyber range represents the real world is important to developing predictive operational and learning outcomes. A high-fidelity simulation does not always mean a real-world simulation. In general, emulation may create a more realistic environment with high fidelity, but simulation is often a more practical option. In other words, a balance must be found among three competing interests: cost, practicality, and reality.

2.4 Accessibility & Usability

Another central question around the capabilities of a cyber range depends on how users access the features of or gain access to the activities of the range. Accessibility and usability can largely be divided into two categories: location and sophistication.

- **Location**: The answer to this question centers largely on whether the deployed range platform is either an on-premises or cloud-based solution. Users, instructors and range owners must all understand how and under what circumstances they can access the range technology and applications.
- **Sophistication**: The question of accessibility also requires analysis relative to the sophistication of the users. Cyber range owners must understand the amount of effort necessary relative to installation, use, and implementation. Operators, trainers and faculty members must understand the modules, levels and tools within each platform or system.

2.5 Scalability & Elasticity

Scalability refers to the ability of the cyber range to support the target population of the system. Elasticity refers to the time required to increase capacity to support additional users.

Ideally, a range is able to simultaneously support its entire potential user population and can increase capacity to support additional users immediately (or nearly so) upon request. Cyber ranges that rely on local hardware infrastructure are limited by the amount of RAM and hard drive space supported on the available hardware.

These ranges can only scale to the point where local resources are exhausted, and as a result, they tend to be very inelastic; increasing capacity to support users beyond the provisioned capacity requires purchase and configuration of new hardware and software. This can take weeks or months. Public cloud-based ranges should generally scale extremely well because they can leverage additional cloud provider systems upon request. They can also be very elastic if they heavily leverage automation and rely on the underlying public-cloud infrastructure to support system provisioning for additional users.

2.6 Functionalities vs Use Cases

As observed from figure 2.2, the **European Cyber Security Organisation** research [8] offers a table which compares cyber range functionalities vis-a-vis the different use cases. Each cyber range capability is marked as (D) Desirable.

Functionality	Cyber Range Use Cases									
	Security Testing	Security Research	Competence Building	Security Education	Development of Cyber Capabilities	Development of Cyber Resilience	Competence Assessment	Recruitment	Digital Dexterity	National Cyber Security Competitions
Orchestration			D	D	D	D	D	D	D	D
Internet Services Simulation						D				
Attack Simulation	D	D	D	D	D	D	D			D
User Activity Simulation		D			D	D				
Competency Management			D	D	D	D	D	D		D
Scenarios and Content Development			D	D	D	D	D	D		D
Data Collection and Analysis		D	D	D	D	D		D		D
Scoring and Reporting			D	D	D	D	D	D		D
Instructor Tools			D	D	D	D	D			

Figure 2.2: Functionalities vs Use Cases

It is important to highlight that each cyber range, regardless of the offered functionalities, could potentially be used for a wide range of different use cases. The difference between cyber ranges lies primarily in the amount of work required for each cyber range to deliver specific use cases. For instance, when addressing the cybersecurity training use case, while a native orchestration functionality is desirable, one could equally adopt a cyber range which does not support orchestration, where the orchestration is substituted by manual effort or by adaption of the cyber

range usage workflow. Finally, it could be argued that a modern powerful laptop containing a virtualised environment could be considered an extreme example of a cyber range, yet one that is focused on the delivery of training and education activities to a small group of users. Ultimately, the choice of cyber range should be based on the intended use cases.

2.7 Types of Cyber Ranges

Cyber ranges have developed into a variety of types with each of them holding a variety of the features and capabilities. In general, **NICE** [15] outlines that there are four main types of cyber ranges: simulations, overlay, emulation, and hybrid. Though the differences may appear insignificant, these differences become important when matching the type of cyber range to the use case of an individual or an organization.

2.7.1 Simulation Ranges

Simulations are the cyber range of choice for most environments. The concept behind simulation is recreating identical systems/network environment based on the behaviour of real components. Simulations run in virtual instances, in a way just like the virtual machines (VMs) and components and represent any kind of device, such as VM servers, network, and storage.

VM templates are standardized and thus, somewhat limited in how closely they simulate a real IT infrastructure. Consequently, those are quick to create and use, and as the used infrastructure matches the existing one, the higher the fidelity of the exercise. So, the granularity with which the simulation can match the target environment infrastructure is directly proportional to the successful simulation exercise outcome. For this reason, cyber ranges should require a strong orchestration layer.

Additionally, a simulation environment can be reconfigured easily, and one can use generic resources to create it. But the primary downside of a simulated network is the unpredictable and unrealistic latency and jitter of network performance.

2.7.2 Overlay Ranges

"Overlay ranges are cyber ranges running on top of real networks, servers and storage" [16].

Overlay cyber ranges have a significant fidelity advantage over simulation ranges, but they come at a considerable cost of hardware and the cost of potential compromise of the underlying network infrastructure. Typically, overlay networks are set up as global testbeds, one of the largest being the Global Environment for Network Innovations (GENI) [17], sponsored by the National Science Foundation.

2.7.3 Emulation Ranges

Emulation is running the cyber range on dedicated network infrastructure, mapping as built network/server/storage infrastructure onto physical infrastructure:

a physical infrastructure that becomes the cyber range. An emulation provides closed-network experiences with multiple interconnected environments. Emulation includes traffic generation that emulates numerous protocols, source patterns, traffic flows, attacks, and underlying internet connectivity. When done right, emulation creates true-to-life experiences, rather than pre-programmed actions and response. A key differentiator for accurate emulation has URLs that resolve to the cyber range's DNS and virtualized Internet IP addresses using real-world geo-IP addresses. The US National Cyber Range (NCR) is probably the most significant emulation initiative.

2.7.4 Hybrid Ranges

As the name suggests, hybrid ranges emerge from a customized combination of any of the above types.

2.7.5 Updated Taxonomy

Muhammad Mudassar Yamin et al. [18] studied the concept, architectures, capabilities, tools and evaluation criteria of Cyber Ranges and developed the taxonomy illustrated in figure 2.3:

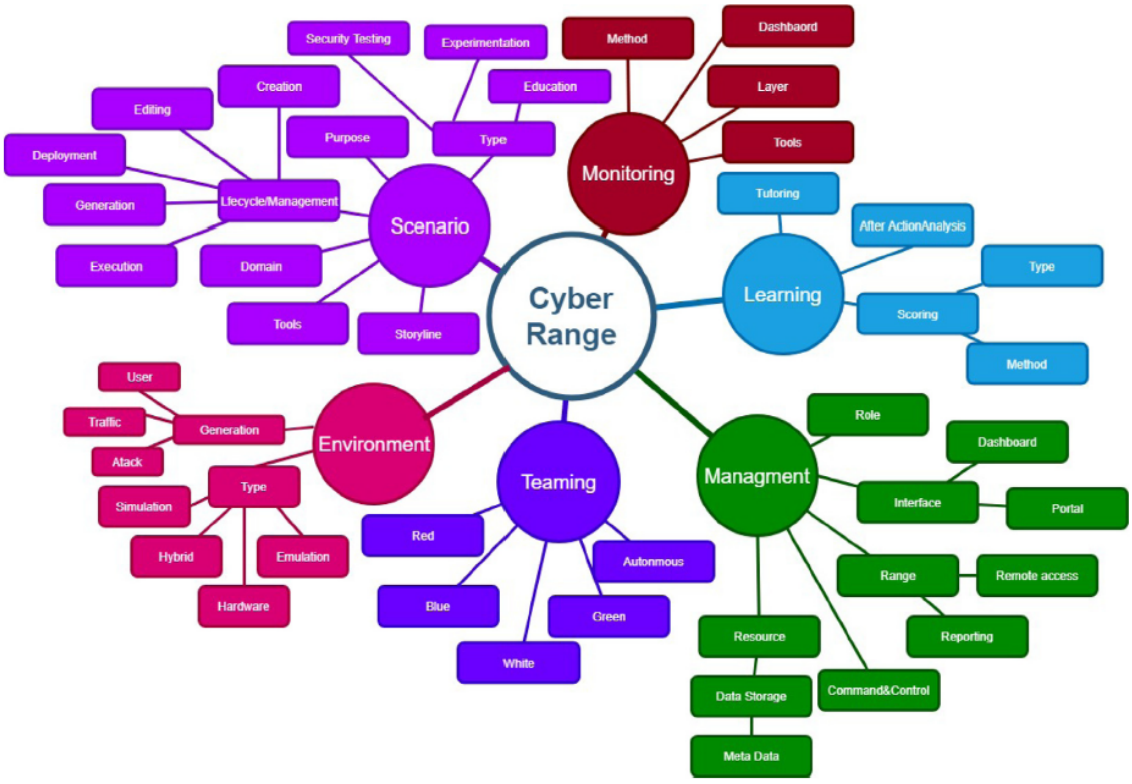

Figure 2.3: Updated Taxonomy

The proposed taxonomy presents the functionality of cyber ranges and security test beds based upon the research's collection of available bibliography.

2.8 Cyber Range Technologies

When talking about cyber range technologies, the focus of the discussion shifts to virtualisation since it is the only technology that allows the creation of cost effective and efficient simulation for most modern technologies. But despite all efforts, not everything can be simulated by using virtualisation technologies and some parts of a simulation environment may indeed require physical components. Most use cases of a cyber range can be achieved through virtualisation and with that in mind, according to **ECSO** [7], cyber ranges can be broadly divided into two types, based on the main technologies used to develop them:

1. **Conventional Cyber Ranges** – They are based on conventional virtualisation
2. **Cloud-Based Cyber Ranges** – They rely on cloud technology

As cloud technologies and conventional virtualisation continue to evolve, a third type of hybrid cyber ranges will also begin to develop, based on the use of the different technologies.

2.8.1 Conventional Virtualisation

Figure 2.4 [8] illustrates the basic types of conventional virtualisation, including traditional hypervisor-based virtualisation and container technology. Many cyber ranges rely on one or the other, or a combination of the two.

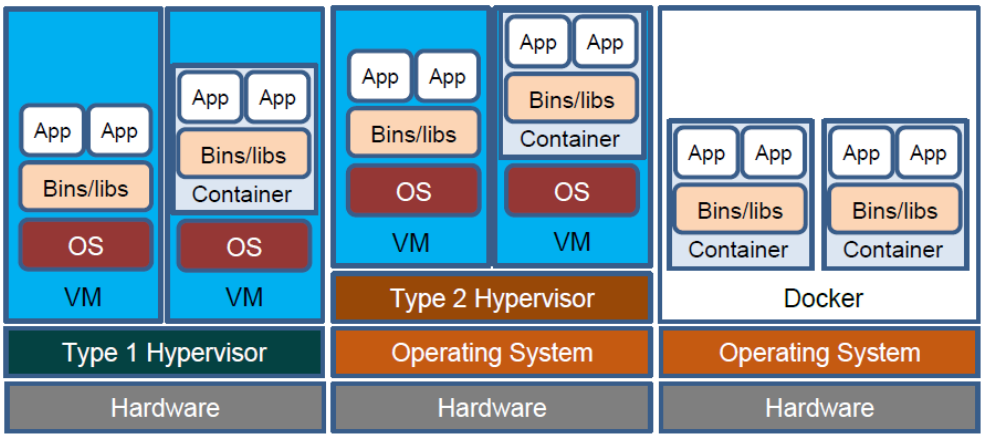

Figure 2.4: Conventional Virtualisation

It is worth noting that in the figure, the container technology is represented by Docker whereas no specific examples are illustrated with regards to traditional virtualisation. This is because, unlike the traditional virtualisation characterised by many technological flavours, the container technology is currently dominated by the Docker technology.

A. **Traditional Virtualisation**

Traditionally, virtualisation is achieved through the creation of a VM, which is a software programme simulating the behaviour of a physical computer or similar component. Many multiple VMs can run on real hardware and an

extra software layer is used, called hypervisor, which ensures that VMs do not interfere with each other and that each has access to the physical resources it needs. There are two types of hypervisors [19]:

- Type 1 or "bare-metal" hypervisors interact with the underlying physical resources, replacing the traditional operating system altogether. They most commonly appear in virtual server scenarios and therefore for the development of data centres.
- Type 2 hypervisors run as an application on an existing operating system (OS). They are most commonly used on endpoint devices to run alternative operating systems and carry a performance overhead because they must use the host OS to access and coordinate the underlying hardware resources.

Sample hypervisors are listed in the table in figure 2.5:

	Type 1	Type 2
Commercial	• VMware's ESXi (data center-focused) • Microsoft Hyper-V • XenServer, now known as Citrix Hypervisor • IBM z/VM	• VMware Workstation (Player or pro) • Parallels
Opensource	• KVM (kernel-based virtual machine)	• VirtualBox • QEMU

Figure 2.5: Sample Hypervisors

Several cyber ranges exist which have been built on traditional virtualisation, especially those cyber ranges that were built before the widespread of cloud technology. The majority of such cyber ranges have been built upon commercial virtualisation providers (for the most part VMware), which accounts for large investment costs and large running costs associated to the annual licensing of the virtualisation software. Traditional virtualisation does not have an advanced level of orchestration, so the cyber range provider would need to develop on top of it.

As the main use case for traditional virtualisation is to have servers running and not to have a large number of both servers and clients dynamically boot, shutdown, delete, configure on demand etc., cyber ranges built on traditional virtualisation are not characterised by a strong level of orchestration, but cyber ranges built on traditional virtualisation benefit from a great level of flexibility and control, which can include support for different capabilities and most importantly the control over the cyber range data and information.

B. **Container Technology**

The other traditional approach to virtualisation, one which has witnessed huge technology developments in recent years, is the one based on containerisation technology. Unlike conventional virtualisation where each VM runs its own OS, containers share a machine's operating system kernel. Specifically, container technology has been developed to facilitate and improve portability of applications across different computing environments by bundling the application

code together with the related configuration files, libraries, and dependencies required for it to run.

Container technology offers cyber range vendors the ability to run multiple containers on the same VM. Container technologies also provide orchestration capabilities able to pre-configure complex networks and inter-container communications making them more cost effective. However, because of their very nature, container technologies do not simulate a real system, but rather a stripped-down version of an environment designed for the sole purpose of running an application. The scope of simulation offered by containers is therefore limited and when it comes to security, both container and virtualisation technologies have inherent issues.

Gartner predicts that by 2022, more than 75% of global organisations will be running containerised applications in production, compared to fewer than 30% today [20]. In the coming years, it is safe to assume that cyber ranges will be requiring an increasing capability of simulating containers.

2.8.2 Cloud Virtualisation

With conventional virtualisation, resources are not efficiently used and in fact it is not natively possible to tap into unused physical resources that span across an entire infrastructure. While building on conventional virtualisation, cloud virtualisation abstracts the underlying physical resources across an entire infrastructure (RAM, disk space, network etc.) and makes them available transparently to the hypervisor to be able to better and more fully utilise all available resources. As a result, the most immediate advantage of cloud virtualisation is cost reductions by increased efficiency.

Another great advantage that comes with cloud virtualisation is its native orchestration capability to automate and facilitate the workflow management of VMs.

The inherent support for dynamic configurations, increased efficiency and scalability makes cloud technology a natural choice of implementation for modern cyber ranges. Three types of cloud based cyber ranges exist which are related to the same three types of cloud deployments normally used for standard business applications.

Public Cloud-based Ranges

In a public cloud environment, the cloud services are open to the public for subscription, which allows users to access services without having to worry about running or maintaining the cloud infrastructure, which is done by the cloud provider. Examples of public cloud providers include Amazon Web Services (AWS), Microsoft Azure, Google and Rackspace.

While easier to set up and operate, public cloud-based cyber ranges do offer several drawbacks which one must take into consideration, such as:

- The first one is the inherent lack of control over the data that flows through.
- Another main disadvantage is the lower level of flexibility required by the cyber range vendor in order to develop and configure bespoke scenarios and to offer

some of the cyber range capabilities.

- Limitations imposed by the public cloud provider to deliver simulated attack scenarios including but not limited to DDoS attacks or any other disruptive type of attacks, which may affect the public cloud infrastructure. For this reason, the majority of available public cloud cyber ranges are mostly focused on the training and education use cases.
- Charging issues as resources must be managed correctly.

Private Cloud-based Ranges

Private Clouds are created and maintained by a private organisation for its own private use or to offer services to its clients. Private cloud cyber ranges, along with cyber ranges based on conventional technology, may be a more suitable choice where privacy and confidentiality are mandatory requirements, such as for government or military applications. Like cyber ranges based on conventional technology, private cloud cyber ranges also have great flexibility in the development of additional cyber range capabilities, beyond the core ones, either natively through the vendor's own development work or through the integration of third-party systems and applications.

Cyber ranges built on private clouds can rely on both commercial and open source solutions. The most common open source technologies used for building private clouds include Openstack [21] and OpenNebula [22].

Hybrid Cloud-based Ranges

A hybrid cloud is an infrastructure that includes links between the private cloud, managed by the organisation, and at least one public cloud, managed by a third party, e.g., AWS or Microsoft. Hybrid clouds offer stronger controls, especially with regards to the security of critical data, assets, and operations while at the same time leveraging the natural scalability of the public cloud infrastructure. Cyber ranges built on hybrid cloud technology can apply better control to the sensitive data associated to the range.

2.8.3 Inter-Cyber Range Communication

Initiatives are being brought forward which bring together multiple cyber ranges in a way to increase or improve simulation capabilities as well as other capabilities beyond what can be offered by a single cyber range.

Federation of Cyber Ranges

In information technology, a federation is a group of computing or network providers agreeing upon standards of operation in a collective fashion. In relation to cyber ranges, standards of operation include scenario description language, description of cyber range capabilities, and request and provision of cyber range services within the federation. With regards to scenarios, for instance, it may be possible to use a common way of describing them across different cyber ranges, allowing each cyber

range to implement and deliver them in their own specific way. The concept of federation is based on the assumption that it is highly complex and costly for a specific cyber range to be able to provide all the required capabilities and functionalities and it is therefore, conceivable that multiple cyber ranges, each with its area of specialisation, could work together to offer end users the ability to achieve multiple use cases and different types of scenarios.

It is important to note that federation does not imply integration, which instead requires that two or more cyber ranges must be able to communicate with one another in order to deliver a scenario. Cyber ranges in a federation may well be able to communicate. However, integration is not a requirement for federation to exist. There are notable examples of cyber range federations being developed in Europe by the European Defence Agency (EDA) [23], and the EU funded project ECHO [24]. The CyberSec4Europe [25] project will demonstrate requirements, specifications and use cases for federation of cyber ranges during 2020 and 2021.

Integration of Cyber Ranges

Compared to the concept of federation, integration does require cyber ranges to talk to one another. Cyber range integration means a group of two or more cyber ranges which can communicate with one another to deliver a simulation environment spread across the cyber ranges. Integration between cyber ranges is usually achieved through traditional integration methods, such as VPN tunnels. Using such technologies requires that the integrated IP address spaces from across the different cyber ranges are different for enabling cyber ranges to transfer data between each other.

In other words, integration requires discipline in planning the technical network environment. Integration of cyber ranges carries more technical and logistical challenges than a federation of cyber ranges. The technical challenges are related to the ability to orchestrate cyber ranges running on potentially different technologies, along with issues of IP addressing, Internet bandwidth and more. Also, scenario design must take into consideration the different cyber range technologies and potential dependencies.

2.8.4 Cyber Range Delivery Models

Having defined what a cyber range is, the question is "how can one have access to one"? Cyber ranges can be broadly accessed in one of two possible ways, which are:

- **Cyber Range as a Service**: In this model, the cyber range is owned and managed by a cyber range provider. They make it available to third parties with charging models based on the cyber range provider, the specific functionalities and capabilities and, ultimately, the services offered. Two main types of cyber range as a service exist:
  - Online – The cyber range is accessed remotely by the client.
  - Physical – The cyber range is hosted at a physical location that the client needs to visit in order to use it, usually for training.

Private companies that provide such services are CybExer [26] and Cisco [27].

- **On-Premise Cyber Range**: An on-premise cyber range is physically deployed on-premise at an organisation. This is usually the most expensive cyber range option as it requires a larger upfront capital investment associated to the cyber range hardware and software. The on-premise option, while definitely more expensive, is better suited to meet the security requirements of an organisation which can better exercise control over the data associated to the use of the cyber range.

How a cyber range can be accessed is irrespective of the technology used for its implementation. While many people can naturally associate the Cyber Range as a Service delivery mode to the use cloud technology (as SaaS concept), there also exists cyber ranges which are accessible as a service, but are not developed using cloud technology or available online.

As cyber range technologies mature, mixed-mode cyber ranges will become more mainstream and begin to appear on the market, providing some of the functionalities on premise within the organisation while leaving other functionalities available as a service (mostly online but also from a physical site of the cyber range provider).

3 Cloud Computing

We live in the age of cloud computing. It offers many things, such as agility, lower cost, and better access to resources on a global scale. The benefits of cloud based data are still being discovered as new technologies emerging day to day, making use of cloud resources and services. Data centers and static computer rooms start to seem outdated and complex applications are deployed at the cloud directly, keeping their business continuity characteristics and utilizing special services (such as DevOps). In our lifetime, we have seen the progression from 1.44Mb floppy disc to 128Gb USB storage devices and beyond. Below, we analyse cloud computing, as to further understand why it is a trend and why we want it to be a key component of our work.

3.1 Definition

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. The term is generally used to describe data centres available to many users over the Internet [28]. Large clouds, predominant today, often have functions distributed over multiple locations from central servers.

As per **NIST** definition [28]:

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.”

3.2 Essential Characteristics

Cloud computing exhibits the following key characteristics:

- **Agility** for organizations may be improved, as cloud computing may increase users’ flexibility
- **Cost reductions**: A public-cloud delivery model converts capital expenditures (e.g., buying servers) to operational expenditure [29]. This lowers barriers to entry, as infrastructure is typically provided by a third party and need not be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is "fine-grained", with usage-based billing options (such as MS Azure Pay-As-You-Go). Of course, in-house IT skills are required for implementation of projects that use cloud computing.

- **Device and location independence** [30] enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect to it from anywhere [31].
- **Maintenance of cloud computing applications is easier**, because they do not need to be installed on each user's computer and can be accessed from different places.
- **Multitenancy** enables sharing of resources and costs across a large pool of users thus allowing for:
  - **Centralization of infrastructure** in locations with lower costs
  - **Peak-load capacity increases**
  - **Utilisation and efficiency improvements** for systems that are often only 10–20% utilised [32] [33].
- **Performance** is monitored by IT experts from the service provider, and consistent and loosely coupled architectures are constructed using web services as the system interface [31] [34].
- **Productivity** may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer [35].
- **Availability** improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery [36].
- **Scalability and elasticity** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time [37] [38] (Note, the VM start-up time varies by VM type, location, OS and cloud providers [37]), without users having to engineer for peak loads [39] [40] [41]. This gives the ability to scale up when the usage need increases or down if resources are not being used [42]. Emerging approaches for managing elasticity include the use of machine learning techniques to propose efficient elasticity models [43].
- **Security** can improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because service providers are able to devote resources to solving security issues that many customers cannot afford to tackle or which they lack the technical skills to address [44]. However, the complexity of security is greatly increased when data is distributed over a wider area or over a greater number of devices, as well as in multi-tenant systems shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

The **National Institute of Standards and Technology's** definition of cloud computing identifies "five essential characteristics":

- **On-demand self-service**. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- **Broad network access**. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- **Resource pooling**. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
- **Rapid elasticity**. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- **Measured service**. Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

3.3 Service Models

Though service-oriented architecture advocates "Everything as a service" (with the acronyms EaaS or XaaS) [45], cloud-computing providers offer their "services" according to different models, of which the three standard models per NIST are **Infrastructure as a Service (IaaS)**, **Platform as a Service (PaaS)**, and **Software as a Service (SaaS)** [46]. These models offer increasing abstraction; they are thus often portrayed as layers in a stack: infrastructure-, platform- and software-as-a-service, but these need not be related. For example, one can provide SaaS implemented on physical machines (bare metal), without using underlying PaaS or IaaS layers, and conversely one can run a program on IaaS and access it directly, without wrapping it as SaaS.

The following sections present some more information about the basic Cloud Service Models [47].

3.3.1 Infrastructure as a service (IaaS)

According to **NIST** [28]:

"**Infrastructure as a Service (IaaS)**: The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls)."

In (Iaas) the cloud service provider provides a set of virtualized computing resources like CPU, Memory, OS, and Application Software etc in the cloud. IaaS uses virtualization technology to convert physical resources into logical resources that can be dynamically provisioned and released by customers as needed.

Some of the major companies offering infrastructure as a service include Rackspace Cloud Servers, Google, Amazon EC2, IBM, and Verizon.

Benefits of IaaS Solutions

- Reduces cost of capital expenditures
- Users pay for the service they want
- Access to enterprise-grade IT resources and infrastructure
- Users can scale up and scale down the resources based on their requirements at any time

3.3.2 Platform as a service (PaaS)

The **NIST**'s definition of cloud computing defines **Platform as a Service (PaaS)** as [28]:

"The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment."

This is more advanced type of cloud computing service. In PaaS, a cloud service provider offers, runs and maintains both system software (i.e., the operating system) and other computing resources. PaaS services include design, development and hosting of applications. Other services include collaboration, DB integration, security, web service integration, scaling etc.

Users don't need to worry about having their own hardware and software resources or hire experts for management of these resources. This scheme provides flexibility in installing software on system; scalability is another advantage of the PaaS. A downfall of the PaaS is the lack of interoperability and portability among providers.

Consumers purchase access to the platforms, enabling them to deploy their own software and applications in the cloud.

Examples of PaaS solutions include Rackspace Cloud Sites, Salesforce.com's Force.com and Google App Engine, Microsoft Azure.

Benefits of PaaS Solutions

- Community – Most of the time, many people are involved in building cloud applications in PaaS environments. This creates a strong supportive community that can help your development team along the way
- No more upgrades – Companies are not required to update or upgrade the infrastructure software. Instead, the PaaS provider handles all upgrades, patches and routine software maintenance
- Lower cost – Companies face lower risk since they do not have to make upfront investment in hardware and software
- Simplified deployment – The development team can concentrate on developing the cloud application without having to worry about the testing and deployment infrastructure

3.3.3 Software as a service (SaaS)

The **NIST**'s definition of cloud computing defines **Software as a Service (SaaS)** as [28]:

"The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings."

In this model, cloud service providers are responsible for running and maintaining application software, operating system and other resources. SaaS model appears to the customer as a web-based application interface where internet is used to deliver services that are accessed using a web-browser. The hosted applications like Gmail and Google Docs can be accesses through different devices like smart phones and laptops etc.

Unlike traditional software, SaaS has the advantage that the customer does not need to buy licences, install, upgrade, maintain or run software on his own computer [48]. It has also other advantages such as multitenant efficiency, configurability and scalability [49].

Benefits of SaaS Solutions

- Rapid Scalability

- Accessibility from any location with Internet
- Eliminates infrastructure concerns
- Custom levels of service offerings
- Bundled maintenance and Support

Figure 3.1 illustrates all three cloud service models.

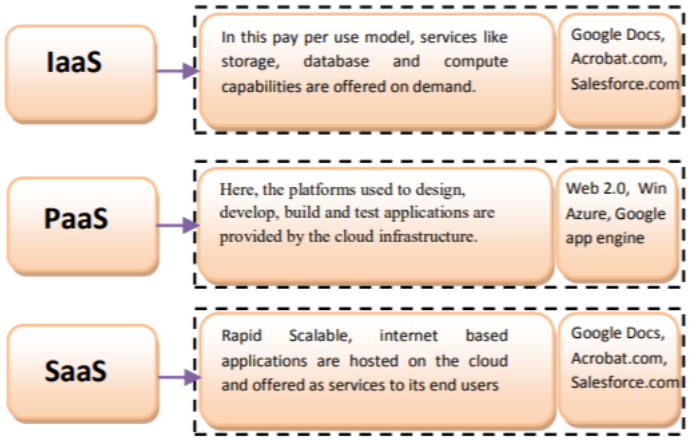

Figure 3.1: Cloud Service Models

3.3.4 Additional Emerging Models

Year to year, newer emerging models appear in the market. Some of those are:

- **Recovery as a Service (RaaS)**: Recovery as a Service (RaaS) solutions helps companies to replace their backup, archiving, disaster recovery and business continuity solutions in a single, integrated platform. RaaS providers help companies recover entire data centers, servers (OS, applications, configuration and data), and database files. RaaS helps business establishments in reducing the impact of downtime in case of disasters or like situations [47].

  RaaS is also called as DRaaS (Disaster Recovery as a Service). Example of companies doing RaaS is WindStream Business, Geminare etc.

  Benefits of RaaS Solutions

  - Prevent temporary or permanent loss of critical company data.
  - Prevent permanent loss of physical infrastructure, including IT infrastructure.
  - Cost-effective way of recovering data.
  - Enable faster recovery while maintaining accuracy.
  - Offer greater flexibility on the type of backup required (either primary or secondary backup).

Businesses can benefit from cloud services by improving efficiency and reducing costs. Based on their priorities different companies can adopt various cloud services, business processes and areas of expertise. Careful planning and preparation should be adopted in case with an IT project before switching to cloud services. Lastly, we have to mention that when reading most of the bibliography, RaaS is not always included in cloud basic service models.

- **Mobile "back-end" as a service (MBaaS):** In this model, also known as back-end as a service (BaaS), web app and mobile app developers are provided with a way to link their applications to cloud storage and cloud computing services with application programming interfaces (APIs) exposed to their applications and custom software development kits (SDKs). Services include user management, push notifications, integration with social networking services [50] and more [51].
- **Serverless computing:** It is a cloud computing code execution model in which the cloud provider fully manages starting and stopping VMs as necessary to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request, rather than per VM, per hour [52]. Despite the name, it does not actually involve running code without servers [52]. Serverless computing is so named because the business or person that owns the system does not have to purchase, rent or provision servers or VMs for the back-end code to run on [51].
- **Function as a service (FaaS):** It is a service-hosted remote procedure call that leverages serverless computing to enable the deployment of individual functions in the cloud that run in response to events [53]. FaaS is included under the broader term serverless computing, but the terms may also be used interchangeably [54]. FaaS is an extremely recent development in the cloud services pyramid, first made available to the world in 2014, followed by IBM Cloud Functions, AWS Lambda, Google Cloud Functions and Microsoft Azure Functions. On the open source side of FaaS, there is IBM/Apache's OpenWhisk (launched in 2016) and, more recently, Oracle Cloud Fn (2017), both of which are available for public use [51].
- **Storage as a Service (STaaS):** It is a business model in which a large company rents space in their storage infrastructure to a smaller company or individual. The economy of scale in the service provider's infrastructure theoretically allows them to provide storage much more cost-effectively than most individuals or corporations can provide their own storage when the total cost of ownership is considered. STaaS is generally seen as a good alternative for a small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage infrastructure.
- **Communications as a Service (CaaS):** It is an outsourced enterprise communications solution that can be leased from a single vendor. Such communications can include voice over IP (VoIP or Internet telephony), instant messaging (IM), collaboration and video conference applications using fixed and mobile devices. The CaaS vendor is responsible for all hardware and software management and offers guaranteed Quality of Service (QoS). CaaS allows businesses to selectively deploy communications devices and modes on a pay-as-you-go,

as-needed basis.

- **Network as a Service (NaaS):** A framework that integrates current cloud computing offerings with direct, yet secure, client access to the network infrastructure. NaaS is a new cloud computing model in which the clients have access to additional computing resources collocated with switches and routers. NaaS can include flexible and extended Virtual Private Network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusion detection and prevention, Wide Area Network (WAN), content monitoring and filtering, and antivirus.

- **Monitoring as a Service (MaaS):** A framework that facilitates the deployment of monitoring functionalities for various other services and applications within the cloud. The most common application for MaaS is online state monitoring, which continuously tracks certain states of applications, networks, systems, instances or any element that may be deployable within the cloud. MaaS makes it easier for users to deploy state monitoring at different levels of Cloud services.

- **Data as a Service (DaaS):** In this model, data is readily accessible through a Cloud-based platform. Data is supplied "on-demand" via cloud platforms (as opposed to the traditional, on-premise models in which the data remains in the customer's hands) and the vendor provides the tools that make it easier to access and explore.

  DaaS provides a dynamic infrastructure for delivering information on demand to users, regardless of their geographical location or organizational separation and, in the process, presents solution providers with a number of significant opportunities. DaaS eliminates redundancy and reduces associated expenditures by accommodating vital data in a single location, allowing data use and/or modification by multiple users via a single update point.

  Typical business applications include customer relationship management (CRM), enterprise resource planning (ERP), e-commerce and supply chain systems and, more recently, Big Data analytics.

  Some of the best-known enterprise-level providers are Oracle's Data Cloud, Amazon DynamoDB, Microsoft SQL Database (formerly known as SQL Azure) and Google Cloud's Datastore.

  For open source projects, Apache Cassandra, CockroachDB or CouchDB will almost certainly catch your eye.

3.4 Deployment Models

There are three basic deployment models for cloud utilization.

As per **NIST** [28], the Cloud Deployment Models are:

- **Private cloud:** The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It

may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- **Community cloud:** The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- **Public cloud:** The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
- **Hybrid cloud:** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

All three, cloud deployment models are depicted in figure 3.2:

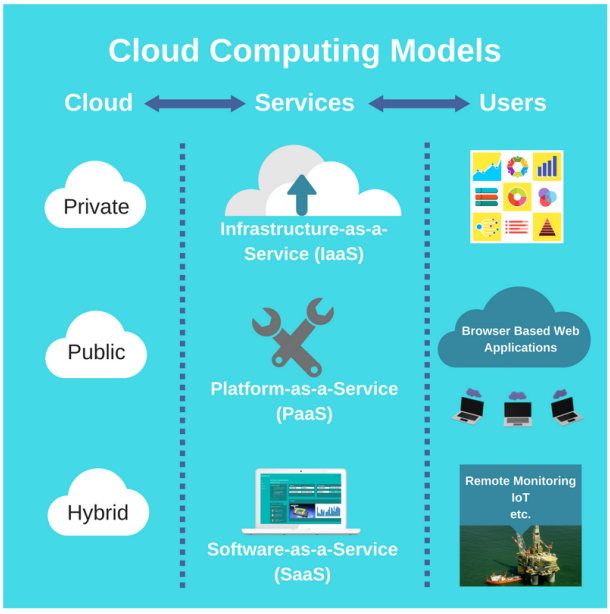

Figure 3.2: Cloud Computing Models

Undertaking a private cloud project requires significant engagement to virtualize the business environment, and requires the organization to re-evaluate decisions about existing resources. It can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers [55] are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures.

Technically, there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon Web Services (AWS), IBM Cloud, Oracle, Microsoft, Google, and Alibaba own and operate the infrastructure at their data center and access is generally via the Internet.

Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources [46]. Gartner defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers [56]. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

There are also some newer emerging deployments, such as the following:

- **Distributed cloud:** A cloud computing platform can be assembled from a distributed set of machines in different locations, connected to a single network or hub service. It is possible to distinguish between two types of distributed clouds: public-resource computing and volunteer cloud:
  - Public-resource computing—This type of distributed cloud results from an expansive definition of cloud computing, because they are more akin to distributed computing than cloud computing.
  - Volunteer cloud—Volunteer cloud computing is characterized as the intersection of public-resource computing and cloud computing, where a cloud computing infrastructure is built using volunteered resources. It can also be called peer-to-peer clouds, or ad-hoc clouds.
- **Multicloud:** is the use of multiple cloud computing services in a single heterogeneous architecture to reduce reliance on single vendors, increase flexibility through choice, mitigate against disasters, etc. It differs from hybrid cloud in that it refers to multiple cloud services, rather than multiple deployment modes (public, private, legacy) [57] [58] [59].
- **Poly cloud:** refers to the use of multiple public clouds for the purpose of leveraging specific services that each provider offers. It differs from multicloud in that it is not designed to increase flexibility or mitigate against failures but is rather used to allow an organization to achieve more that could be done with a single provider [60].
- **Big Data cloud:** The issues of transferring large amounts of data to the cloud as well as data security once the data is in the cloud initially hampered adoption of cloud for big data, but now that much data originates in the cloud and with the advent of bare-metal servers, the cloud has become [61] a solution for use cases including business analytics and geospatial analysis [62].
- **HPC cloud:** HPC cloud refers to the use of cloud computing services and infrastructure to execute high-performance computing (HPC) applications [63].

These applications consume considerable amount of computing power and memory and are traditionally executed on clusters of computers.

Clouds may be limited to a single organization (enterprise clouds [64] [65]), or be available to many organizations (public cloud).

Cloud computing relies on sharing of resources to achieve coherence and economies of scale.

Advocates of public and hybrid clouds note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand [65] [66] [67], providing the burst computing capability: high computing power at certain periods of peak demand [68].

Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models [69].

The availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture and autonomic and utility computing has led to growth in cloud computing [70] [71] [72]. By 2019, Linux was the most widely used operating system, including in Microsoft's offerings and is thus described as dominant [73]. The Cloud Service Provider (CSP) will screen, keep up and gather data about the firewalls, intrusion identification or/and counteractive action frameworks and information stream inside the network [74].

3.5 Security and Privacy

Cloud computing poses privacy concerns because the service provider can access the data that is in the cloud at any time. It could accidentally or deliberately alter or delete information [75]. Many cloud providers can share information with third parties if necessary for purposes of law and order without a warrant. That is permitted in their privacy policies, which users must agree to before they start using cloud services. Solutions to privacy include policy and legislation as well as end-users' choices for how data is stored [75]. Users can encrypt data that is processed or stored within the cloud to prevent unauthorized access [75] [76]. Identity management systems can also provide practical solutions to privacy concerns in cloud computing. These systems distinguish between authorized and unauthorized users and determine the amount of data that is accessible to each entity [77]. The systems work by creating and describing identities, recording activities, and getting rid of unused identities.

According to the **Cloud Security Alliance** [78], the top eleven threats in the cloud, ranked in order of significance are [79]:

1. Data Breaches.
2. Misconfiguration and Inadequate Change Control.

3. Lack of Cloud Security Architecture and Strategy.
4. Insufficient Identity, Credential, Access and Key Management.
5. Account Hijacking.
6. Insider Threat.
7. Insecure Interfaces and APIs.
8. Weak Control Plane.
9. Metastructure and Applistructure Failures.
10. Limited Cloud Usage Visibility.
11. Abuse and Nefarious Use of Cloud Services.

In a cloud provider platform being shared by different users, there may be a possibility that information belonging to different customers resides on the same data server. Because data from hundreds or thousands of companies can be stored on large cloud servers, hackers can theoretically gain control of huge stores of information through a single attack. Some examples include the Dropbox security breach, and iCloud 2014 leak [80]. Dropbox had been breached in October 2014, having over 7 million of its users passwords stolen by hackers in an effort to get monetary value from it by Bitcoins (BTC). By having these passwords, they are able to read private data as well as have this data be indexed by search engines (making the information public) [80].

There is the problem of legal ownership of the data (if a user stores some data in the cloud, can the cloud provider profit from it?). Many Terms of Service agreements are silent on the question of ownership [81]. Physical control of the computer equipment (private cloud) is more secure than having the equipment off-site and under someone else's control (public cloud). This delivers great incentive to public cloud computing service providers to prioritize building and maintaining strong management of secure services [82].

Some small businesses that don't have expertise in IT security could find that it's more secure for them to use a public cloud. There is the risk that end users do not understand the issues involved when signing on to a cloud service (persons sometimes don't read the many pages of the terms of service agreement, and just click "Accept" without reading). Fundamentally, private cloud is seen as more secure with higher levels of control for the owner, however public cloud is seen to be more flexible and requires less time and money investment from the user [83].

3.6 Limitations and Disadvantages

In cloud computing, the control of the back end infrastructure is limited to the cloud vendor only. Cloud providers often decide on the management policies, which moderate what the cloud users are able to do with their deployment [84]. Cloud users are also limited to the control and management of their applications, data and services [85]. This includes data caps, which are placed on cloud users by the cloud

vendor allocating a certain amount of bandwidth for each customer and are often shared among other cloud users [85].

Privacy and confidentiality are big concerns in some activities. For instance, sworn translators working under the stipulations of an NDA, might face problems regarding sensitive data that are not encrypted [86].

Cloud computing is beneficial to many enterprises. it lowers costs and allows them to focus on competence instead of on matters of IT and infrastructure. Nevertheless, cloud computing has proven to have some limitations and disadvantages, especially for smaller business operations, particularly regarding security and downtime. Technical outages are inevitable and occur sometimes when cloud service providers (CSPs) become overwhelmed in the process of serving their clients. This may result in temporary business suspension. Since this technology's systems rely on the Internet, an individual cannot access their applications, server or data from the cloud during an outage [87]. However, many large enterprises maintain at least two internet providers, using different entry points into their workplaces, some even use 4G as a third fallback.

3.7 Emerging trends

Cloud computing is still a subject of research [88]. A driving factor in the evolution of cloud computing has been chief technology officers seeking to minimize risk of internal outages and mitigate the complexity of housing network and computing hardware in-house [89]. Major cloud technology companies invest billions of dollars per year in cloud Research and Development. For example, in 2011, Microsoft committed 90% of its $9.6 billion R&D budget to its cloud [90]. Research by investment bank Centaur Partners in late 2015 forecasted that SaaS revenue would grow from $13.5 billion in 2011 to $32.8 billion in 2016 [91].

3.8 Digital forensics in the cloud

The issue of carrying out investigations where the cloud storage devices cannot be physically accessed has generated a number of changes to the way that digital evidence is located and collected [92]. New process models have been developed to formalize collection [93].

4 Basis of Work

Our work starting point is the research work **"Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education"** [1]. Within this research, the authors recognize the progress within the academic and research fields, but also identify the lack of simple documentation for beginners, as often only the source code is given. Additionally, those systems tend to be complex, so their setup can become time consuming and cumbersome. To fill this gap, the authors make an assessment of typical functionality and development methodologies for cybersecurity testbeds.

The typical cybersecurity model followed consists of a security challenge, i.e. a task or activity to be solved by a participant and a participant (challenger) that is able to work on the challenge and solve it.

Typical design considerations for cybersecurity testbeds are presented. Figure 4.1 shows a schematic example of a layout of a cybersecurity testbed [1]:

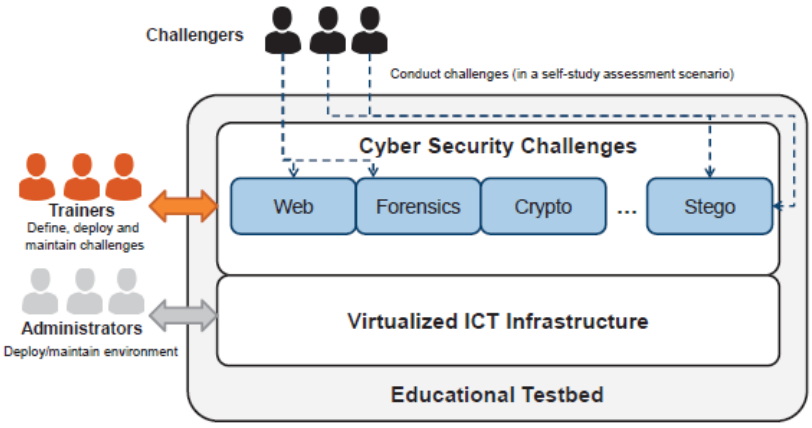

Figure 4.1: Layout of a Cybersecurity Testbed

On top of virtual ICT infrastructure, security challenges of different categories are provided. The challenges can have different contents depending on the course goals and the target audience. Challenges are typically defined and implemented in a collaboration between administrative staff and trainers, as a trainer has the responsibility of designing/implementing of the challenge, while the administrator is responsible for making it available through the offered service. Challengers can conduct security challenges. In a self-assessment case study, for example, challengers would be able to start and conduct security challenges themselves, while in supervised trainings, trainers might start the security challenges for challengers.

Security Challenges: The analysis of software systems for security vulnerabilities or the examination of data or programs and how to gamify them into a sort of puzzle

in order to teach specific security principles in a practical and playful way.

Challenge types: the most common types are the following:

- **Web:** It is related to websites or webapps. The challenger has to identify and exploit vulnerabilities.
- **Forensics:** A challenger is tasked with finding certain information, hidden in a data set.
- **Crypto:** The challenger learns important concepts and protocols of cryptography.
- **Reversing:** Reverse engineering (binary file analysis).
- **Exploitation:** Finding and exploiting vulnerabilities.
- **Stego:** Stenography is the art of hiding data, even encrypted.

Providing security Challenges: Looking at the different types of challenges available, one can understand that they require some sort of server, a specific OS or access to specific files. If one wishes to provide security challenges to a user, they also have to provide the appropriate infrastructure. There are multiple different approaches, with which this can be achieved, such as replicating the servers and providing challengers access to them, which is not really cost efficient and might also compromise servers. Therefore, user and server isolation need to be implemented and servers might also have to be reset to a previous state after a challenger solved the challenge. Some possibilities and their underlying technologies are briefly explained below:

- **Security challenges as a local application:** It provides security challenges locally by utilizing a challenger's own system utilizing desktop virtualization technology. This approach has the advantage of requiring very little resources on the provider site apart from hosting copies of the interface and images. This reduces costs, but challenges requiring more complex and resource intensive server setups might be too much for the hardware of challengers.
- **Security challenges as a service:** There is also the possibility to provide challenges as a service that can be started on demand. This way the challenger does not need to provide a strong personal computer to run challenge servers on a VM.

Afterwards, the authors describe a life cycle model, to design and implement cybersecurity testbeds. They also validate this model using open source technology. The design life cycle for testbeds is shown in figure 4.2 [1]:

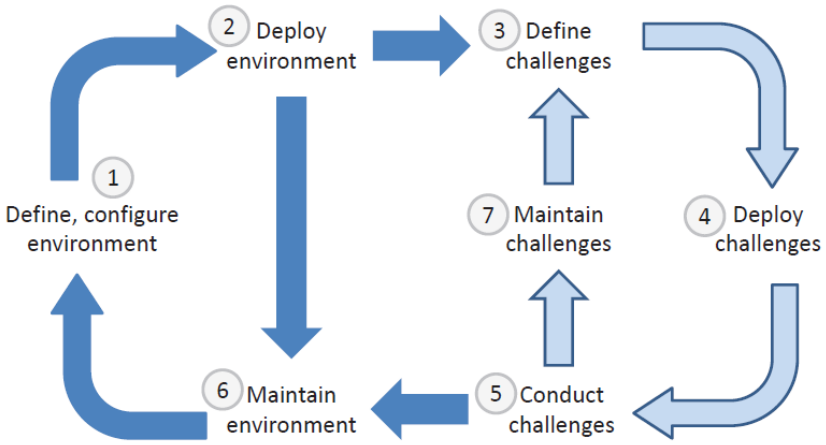

Figure 4.2: Testbed Design Life Cycle

As observed from the figure, the cycle is divided into two parts: The first part (dark blue) contains all activities around setting up the main environment (e.g., defining the hardware, virtual infrastructure and networks). The second part (light blue) focuses on setting up and maintaining the challenges on top of this infrastructure (OS, software or applications).

The use of local or cloud environment is questioned as well as the use of open source or commercial software and the scalability or complexity of the project.

Lastly, the authors provide a case study on how a cybersecurity testbed can be defined from scratch and present a running example of a web challenge (SQL Injection).

5 Design of a Cloud Based Model for Cyber Ranges

The work described at the previous section is very insightful and comprehensive, but it focuses mostly on physical hosts, which can be a restraining factor to multiple and/or concurrent deployments. Additionally, a respectful cloud service provider can guarantee scalability and availability needs. In our work, we build an end user application, utilizing a respectful cloud service provider (MS Azure) and automation processes for the sake of proceeding with the creation of the required infrastructure and deliver the final environment to the trainee, without the need of an overseeing administrator.

The purpose of this work is to utilize the above-mentioned life cycle design, as to create an automated way to deploy cyber-ranges directly at the cloud (MS Azure) and offer the service to the end user in a hassle free manner.

A generic description of the implementation life cycle follows:

A. There is a set of preconfigured components at the cloud:
    a. For infrastructure: VMs (windows/linux), disk snapshots, applications built within containers, etc.
    b. For challenges: vulnerable or misconfigured, containerized applications or systems. Simple web servers can provide additional functionality (i.e. provide information or data sets to the end user or validate results/end user inputs), if needed.

B. The deployment of vulnerabilities and its components is automatic, according to user preference (there is a basic user interface)

C. The user will have the ability to reset/recreate a course, if needed

D. Multiple deployments could take place simultaneously, providing a one-to-one approach. Otherwise, the deployment set could be a training source for a group of users, e.g., catch the flag sessions, depending on the aims of each training course. Note that:
    a. The VMs are fully scalable, as per MS Azure datacentre restrictions.
    b. Other technologies can be utilized as well, such as Azure Kubernetes Service (AKS).

E. One VM can host one or several vulnerable applications, according to the deployment approach. This is subject to each application/environment designer's needs and targets.

F. After each training course fulfils its purpose, an automated procedure will clean up all the cloud resources used.

Following the authors' steps, we will describe all the life cycle steps below.

1. **Define, configure environment:** the testbed creates an environment that fulfils:

   - **Automated deployment of environment and challenges:** The VMs/challenges are automatically deployed and the challenges are pre-built in. In this way i) the trainee does not need a third party to deploy the infrastructure for them and ii) the challenge deployment time is minimum.
   - **Reuse of challenges:** We did not implement this step in our case, but it is a feature that can be added easily to the main application.
   - **Maintain low cost of platform:** The automation that creates and maintains the required infrastructure only for the needed time frame, ensures that the pay-as-you-go model keeps the resources cost to the minimum, as all resources are automatically releases when the challenges end and all users quit the application. We understand that there might be a respectable, relevant cost involved, as we have to use commercial services (cloud service provider), but in this way:
     - One does not need to acquire and/or maintain large IT infrastructure on premises.
     - One does not need to commit specialized personnel for the infrastructure maintenance.
     - They can utilize maximum scalability and availability, ensuring that the training sessions will always be available, even with a large number of participants.

     Additionally, large educational organizations can make special arrangements with the cloud service providers, thus ensuring lower usage cost.

   - **Keep high availability of platform:** As mentioned previously, high availability is a characteristic that our CSP can provide us with, as SLAs ensure the 99.99% availability of our resources (under some components restrictions) [94]. We can also use other cloud native mechanisms such as Availability Zones [95] and Azure Site Recovery [96], utilize Traffic Manager Profiles [97], etc.
   - **Enable challenge as a service:** The platform enables the end users to select the challenges in which they want to participate. A future addition could be the trainer to be able to select challenges for all users. The main application core can be the same, but another UI must be created.

2. **Deploy Environment:** We maintain a core VM on which all our challenges are installed. The hard disk of this machine is kept as a snapshot at the cloud. Using this snapshot, one can deploy as many identical VMs as they want of any (data centre available) size.
   Furthermore, one can preserve many different snapshots and use the one they wish at each case, but this approach is different from the one we designed.

In any case, the designer/developer can follow any direction they want to and produce a more appropriate version of the offered service.
For the trainer to create/add a new challenge, e.g., a vulnerable application, they need to create a VM using the latest hard disk snapshot and then deploy the initial state of it, within the VM, in a dockerized form. Then they just save the new snapshot (ideally using a new name) and point the main application to use this snapshot (to be more accurate this configuration change is within the VM creation PowerShell script). Lastly, a minor addition must be made at the relevant application util file, in which the new application's info will appear (such as name, port, etc.). Again, this work can split to include the infrastructure administrator as well.
Note that even the trainer does not need to have a VM locally, as this is deployed to the cloud environment directly.

3. **Define Challenges:** the designed testbed can support a variety of challenges. A classification of the challenges could be applied in the future, if needed such as 'easy', 'medium', 'hard'. Another classification that one can perform could involve the target audience level. For example, the challenges that target school students should be different from challenges that target business specialists or security experts.
   The trainer could also inject some introductive information or even a small training session before the challenge appears.

4. **Deploy Challenges:** The deployment of the challenges is fully automated in our case. When the end users navigate between our main application screens, they are called to select one or more of the available vulnerable applications that they want to interact with (second screen).
   When the users selects those, followed by a valid username and presses the "next" button, the application back-end will perform the following actions:
   - Collect and validate the user's choices and username entered field.
   - An overlaid window is displayed until the VMs deployments and vulnerable applications initialization takes place.
   - If all previous checks are successful, the method "calculateVirtualMachines" is called for the calculation of the required resources (VMs) that need to be created at MS Azure.
   - The result from previous step is returned and according to it, all the required VMs are created at MS Azure (using Powershell and jPowershell) using a Java Task approach and the selected containers are started (using Java Secure Channel / JSch). For each VM object that needs to be built, the "CreateNewAzureVM" method is called.
   - The software waits for some time for each VM to be created and then performs a check if all the vulnerable applications selected by the user are started successfully, using the "startVulnerabilities" method.
   - When the previous task finishes successfully, then the main application

navigates the user to the third screen in which they are presented with the deployed applications' information.

5. **Conduct Challenges:** The end user is presented with the required application info and they can use a web browser to navigate to the given URL for joining the training session. Figures 5.1 and 5.2 demonstrate example screenshots of the two deployed applications.

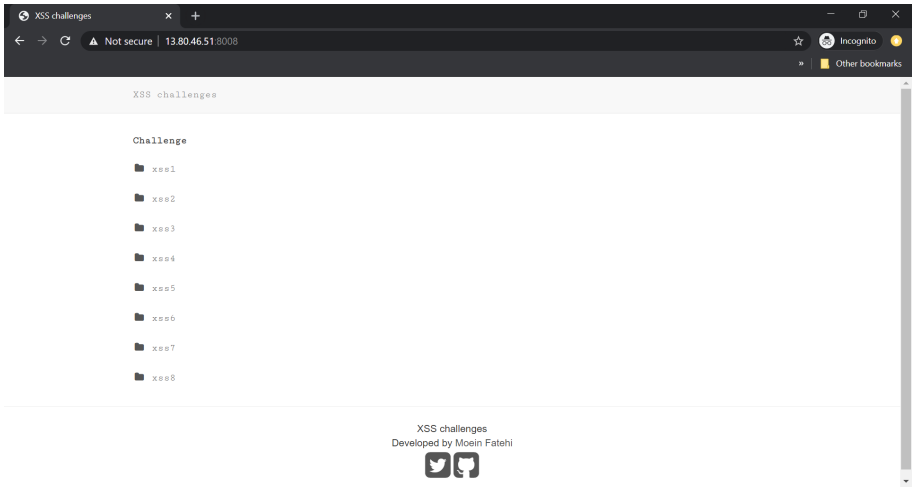

Figure 5.1: XSS Challenge example

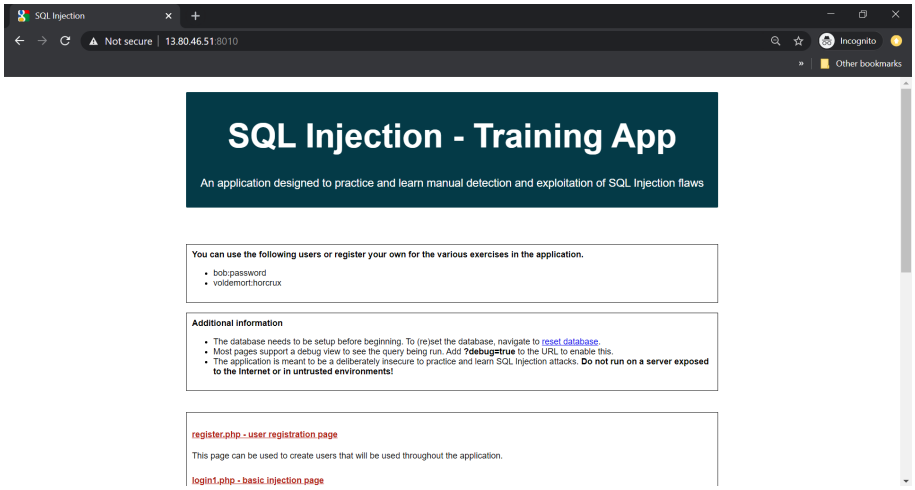

Figure 5.2: SQL Injection Challenge example

6. **Maintain Environment:** As mentioned earlier, the application also automates the clean-up of the allocated resources for minimizing the related cost. When the end user finishes the training and presses the button "Finished", as depicted at figure 5.3, the application back-end performs the following actions:

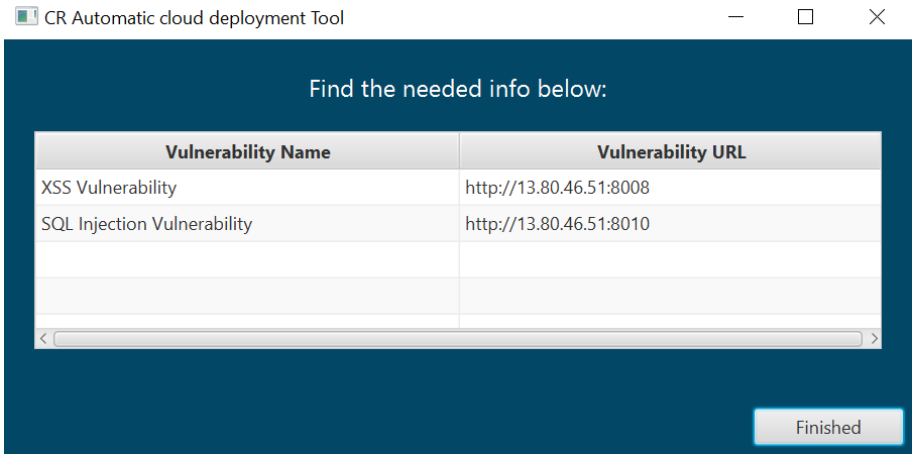

Figure 5.3: Main Application Screen

- The method "finished" is called, which it will firstly present an overlaid confirmation window on the finished action, as the deletion process cannot be stopped or undone.
- If confirmed, the resources will be released as per below procedure.
- The list of the current utilized VMs will populate.
- Using a Java Task approach the "clearResources" method is called.
- This method first shuts down the VM(s) involved, by securely connecting to it (utilizing JSch), and then proceeds with the full deletion of each of the VMs previously allocated resources (VM, disk, NIC, Public IP Address), utilizing Powershell and jPowershell within Java Tasks.
- Again, an overlaid window is displayed until the VMs resources release Java Task is finished. Lastly, if the Tasks finishes successfully, then the main application navigates the user to the next screen.

After those steps, all the previously allocated MS Azure resources are fully deleted, and thus are stopped from adding cost to the respective MS Azure subscription. The platform administrator can then confirm those actions at the MS Azure portal, within the specific Resource Group.

Related to maintaining the environment components such as the OS, from the up-to-date viewpoint, this has clearly to do with the designer's choices. The current implementation uses a pre-defined snapshot of an Ubuntu Server as a hosting VM. VM's OS upgrading and upgrading need an admin to perform those actions using the steps below:

- Create a new VM from the snapshot in use.
- Make all needed updates/upgrades, such as:
  - Download package information or their dependencies from all configured sources using the **sudo apt-get update** command.

- Update all the packages presently installed in our Linux system to their latest versions and install or remove packages as needed, in order to complete the upgrade using the **sudo apt-get upgrade** and **sudo apt-get dist-upgrade** commands.

- Save the snapshot either by overwriting or renaming the old one. Using the latter way ensures safe reverting back to the previous snapshot in case of any kind of failure/malfunction, but needs also a snapshot name change at the VM creation PowerShell script.

7. **Maintain Challenges:** Each challenge instance starts after an end user selects it to be deployed and ends when the application is terminated. In this way, we ensure that challenges are created at demand and end when they fulfil their purpose, avoiding orphan instances and wasting resources.
   Once more, related to the internal challenge components update/upgrade, e.g., Challenge web application version, it is again at the trainer's hands (following a similar procedure, as described above) and this is because the trainer can use specific components versions, on purpose, because they wish:
   - To simulate a specific, existing environment.
   - To reproduce a specific environment vulnerability set.
   - To make use of a vulnerable version application itself as part of the challenge, for example use a known web server or SQL database vulnerability.

5.1 Benefits vs Constraints

The benefits of our implementation are numerous and can be summarised below:

A. We provide a complete end user application; the end user can work from the beginning without the need of any third party.

B. The application and its code are easy to configure, deploy, use and maintain.

C. The cloud-based approach that we followed ensures advantages such as:
   i. All the application required components are located at the cloud environment; the end user needs only a personal computer for running the application.
   ii. No local ICT large infrastructure and data centers are needed.
   iii. Agility of the implementation.
   iv. Performance & Scalability; we can scale up or down the utilised resources, e.g., VM size, and handle the increasing load.
   v. High Availability as we utilise the relevant cloud services.
   vi. Business continuity: Backup and Disaster Recovery services, if needed.

vii. Security: The environment's attack surface is small. The VMs that are created do have restricted (IP based filtered) access. Further cloud mechanisms can be added, if needed.

viii. Cost Reduction: The charges for the cloud components are running only when a training session takes place.

ix. Device independence: The application is written in Java, so most devices can run it (portability).

x. Maintenance: The admin can monitor all cloud components through the MS Azure portal. Further automations (scripts) can be developed for additional utilities such as a clean up script, in case something went wrong due to a network communication error.

D. Modern and agile technologies utilization such as docker, providing:

i. Lightweight technology.

ii. Portability between different platforms/clouds.

iii. Efficiency by using fewer resources.

iv. Agility.

v. Improved security by isolating applications from the host system and from each other.

vi. Faster app start-up and easier scaling.

vii. Flexibility.

viii. Easier management.

E. The produced implementation and application are easy to use:

i. It makes infrastructure administrator's life easier, by not having to handle large local IT resources and having an overall view on the status of the resources.

ii. The trainer can handle the adding/reconfiguring of the challenges easily, by following the relevant guidelines.

iii. The end user (trainee) is trouble free, as they just run a Java application and uses the information provided by both the trainer and the application itself.

Constraints of the implementation are also identified and can be summarised below:

A. Limited accessibility to cloud backend.

B. IT experts are also necessary with the purpose of deploying an overall complete and safe application.

C. Security: as it provided from the cloud service provider.

D. Data Privacy and confidentiality: Even if data and communications are encrypted and SLAs are in place, there is always a lack of confidence to the cloud service provider.

E. Bandwidth: Despite the fact that all resources are located at the cloud environment, a minimum bandwidth is required, which in turn depends to the number of concurrent, active users.

F. Outage Periods: MS Azure provides high availability but not 100% as we understand. This is something out of the organizer's control.

5.2 Contribution

The contribution of this work will be initially for the system manager, who will only have to monitor the automated procedures health status, instead of creating and deleting the required cloud resources. Additionally, the system manager does not have to worry about the required resources, as those will always be adequate, up to the cloud service provider's capabilities.

Then, we can identify the contribution for the trainer who will only have to manage and maintain the current/future vulnerable systems or applications, according to the training needs. The trainer can design and deploy a variety of training fields, taking into consideration the target audience.

Lastly, the trainees participating will be able to easily select the training fields (vulnerability types) that they wish to attend and those will be automatically deployed on the cloud environment, without the need of any third party. Furthermore, our work will provide the end user with a simplified UI that could be extended to offer options such as to reset/recreate a course, if required. Upon completion of the course, the whole infrastructure components will be deleted, as to release all the utilized resources. Note that just deleting a VM will not release all the VM resources, consequently additional work is done in this field as well.

6 Model Implementation

6.1 Tools

There are multiple layers at the application architecture. The code is written in a way that can allow further development and be straightforwardly expanded in any way.

First of all, the main UI application is built using JavaFX, a modern programming approach which can be run in a vast range of devices. The UI comprises of four application screens, along with the relevant controllers.

A description of the application lifecycle is shown in figure 6.1:

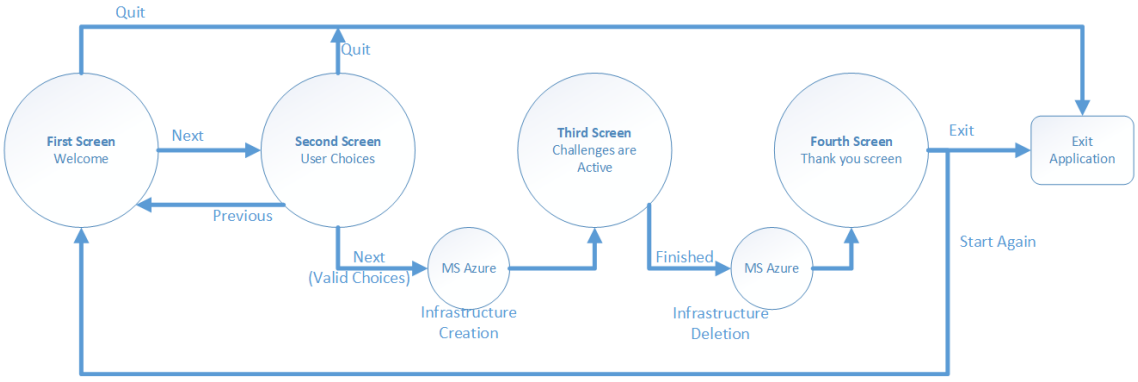

Figure 6.1: Main Application Life Cycle

For our implementation, we use MS Azure as our cloud service provider. Additionally, we use Docker container framework for the vulnerable applications deployment.

6.1.1 Why use MS Azure Cloud Services Provider

As already reported in the previous section, a reputable Cloud Service Provider can ensure many critical benefits, while we do not have to purchase, maintain or upgrade any kind of hardware or special designed software.

Major Characteristics such as:

- Agility
- Flexibility
- Cost reduction
- Device and location independence
- Low Maintenance

- Multitenancy
- Increased Performance, monitored by IT experts from the service provider
- High Availability
- Scalability and elasticity
- Security at acceptable level

make the use of any CSP very attractive. On the other hand, if security is of utmost importance, e.g., Military, sensitive data, then the use of a local data centre or a private cloud solution should be considered as the most appropriate one.

6.1.2 Why Use Docker Containers

In this work, we utilize containerization and more specifically Docker micro services, for additional flexibility and isolation and following the technology trend of this new era. Below, we introduce some of this modern technology's basic characteristics and benefits.

6.1.2.1 Short Description of the Technology

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings [98].

The key word here is "isolated." Isolation means speed—containers are smaller entities than VMs so they can be deployed much faster. Isolation means responsive—start-up times are short. Isolation means versatility—containers are portable between different platforms and cloud vendors.

At figure 6.2, the containers technology architecture is illustrated.

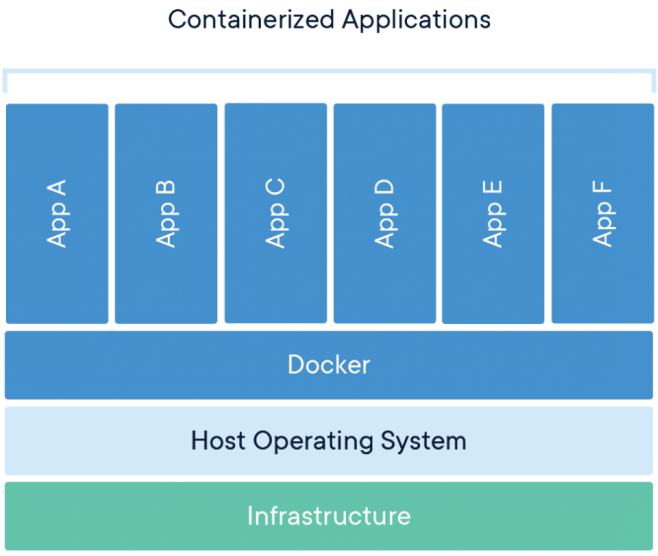

Figure 6.2: Containers Technology Architecture

6.1.2.2 Technology Benefits

Containerization of applications brings many benefits, including the following [99]:

- Portability between different platforms and clouds — it's truly write once, run anywhere.
- Efficiency through using far fewer resources than VMs and delivering higher utilization of compute resources.
- Agility that allows developers to integrate with their existing DevOps environment.
- Higher speed in the delivery of enhancements. Containerizing monolithic applications using microservices helps development teams create functionality with its own life cycle and scaling policies.
- Improved security by isolating applications from the host system and from each other.
- Faster app start-up and easier scaling.
- Flexibility to on virtualized infrastructures or on bare metal servers.
- Easier management since install, upgrade, and rollback processes are built into the Kubernetes platform.

6.1.3 Technologies Used

We investigated several approaches in order to select the appropriate technologies to be utilized for the project implementation. Java is one of the most popular programming languages and PowerShell provide us with unique interaction with MS Azure cloud services provider. The technologies currently used are the following:

- **Java [100]:** for the main functionality and the User Interface, using certain, required libraries (JSch, JPowershell, etc.)
  - Main application UI was created with JavaFX [101]. The code was organized to separate application screens along with the relevant controllers, handling the user input and providing them with all necessary info. JavaFX controller works based on MVC (Model-View-Controller) JavaFX MVC can be achieved by FXML (EFF-ects eXtended Markup Language). FXML is an XML based language used to develop the GUI for JavaFX applications as in the HTML. FXML can be used to build an entire GUI application scene or part of a GUI application scene. This FXML allows developers for separate User Interface logic from the business logic. If suppose User Interface in your JavaFX application, then no need to compile the application even if we have done some changes to the application. If we want, we can edit the FXML in the editor and re-run the app.
- **Apache Maven:** For making the build process easy, we used Maven. Apache Maven [102] is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.
- **PowerShell:** For cloud implementation (scripts and functions in order to create/configure/delete resources). PowerShell is a cross-platform task automation and configuration management framework, consisting of a command-line shell and scripting language. Unlike most shells, which accept and return text, PowerShell [103] is built on top of the .NET Common Language Runtime (CLR), and accepts and returns .NET objects. This fundamental change brings entirely new tools and methods for automation. To handle the needed integration with MS Azure, we use Azure Powershell. The latter is a set of cmdlets for managing Azure resources directly from the PowerShell command line. Azure PowerShell is designed to make it easy to learn and get started with, but provides powerful features for automation. Written in .NET Standard, Azure PowerShell works with PowerShell 5.1 on Windows, and PowerShell 6.x and higher on all platforms [104].
- **Docker Container Framework [98]:** The vulnerable applications are created as docker containers. As already pointed out, a container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

6.1.3.1 Third Party Libraries

We have utilized two external libraries for ensuring the integration between Java and powershell (jPowershell) and the connection to our infrastructure and pass the required commands (JSch).

jPowershell [105]**:** It is a simple Java API that allows to interact with PowerShell console. The user can use it to execute one or multiple commands using the same

PowerShell session or even execute a PowerShell Script.

This library was used for making the UI Java code to place calls to predefined MS Azure powershell scripts (passing the required parameters too), as to interact with MS Azure.

JSch - Java Secure Channel [106]**:** A pure Java implementation of SSH2.

JSch allows one to connect to a sshd server and use port forwarding, X11 forwarding, file transfer, etc. Also, one can integrate its functionality into their own Java programs. JSch is licensed under BSD style license. As it is well-known, SSH provides support for secure remote login, secure file transfer, and secure TCP/IP and X11 forwarding. It can automatically encrypt, authenticate, and compress transmitted data.

JSch is in pure Java, but it depends on JavaTM Cryptography Extension (JCE).

In our work, this library was used for the secure connection between the UI Java code (front-end) and the MS Azure infrastructure (VM(s) – back-end), passing the required commands.

6.2 Vulnerable Applications Used

In our case, we use as examples of vulnerable applications, two ready-to-go educational vulnerable web applications. Those have the benefit of a rapid, easy and controllable way of deployment through the relevant git repository. In order to add a new (to our environment) application, we only have to follow the developer's guidelines to deploy the application and just add the exposed port in order to provide this information to the end user.

After the application deployment, we have to save/update our MS Azure initial snapshot, which is the original one that is replicated and lastly add the application info to the relevant java util file. The application will then pick all the required information and present it to the end user. No other actions are needed for the system/platform administrator, which makes the platform very easy to use and flexible.

Below we will refer to the applications that are used to demonstrate our application, but we will not analyse the apps themselves in depth as this is out of scope of our work.

The two web applications that were chosen for the initial application demo creation are the following:

1. Vulnerability Type: Cross-Site Scripting (XSS)
   Developer: Moein Fatehi
   URL: https://github.com/moeinfatehi/XSS-challenges
   Command: docker run -d -p 8008:80 moeinfatehi/xss_vulnerability_challenges
   Info: This project is for Educational purpose only. This repository is a Dockerized php application containing some XSS vulnerability challenges. The ideas behind challenges are:
   - Javascript validation bypass

- html entities bypass
- WAF bypass
- Black-list validation bypass
- Basic XSS validation bypass
- Double encode bypass of WAF to exploit XSS
- Exploiting XSS by bypassing escape characters

2. Vulnerability Type: SQL injection
   Developer: Riyaz Walikar
   URL: https://github.com/riyazwalikar/sql-injection-training-app
   Command: -
   Info: A simple PHP application that can be used to demonstrate and train participants to detect and exploit SQL Injection vulnerabilities. The PHP code is extremely primitive, but clearly demonstrates the vulnerability and can be used to teach the various kinds of SQL injection in a hands on class.

Additionally, we inserted two 'dummy' applications ("Dummy Vulnerability 01" and "Dummy Vulnerability 02") only for those to show at the application's user selection choices.

We should note here that any kind of application that is/can be containerized (pretty much the majority of the applications) is able to be added to our main application, providing the user with multiple kind of challenges. Custom applications are welcome, as the application container deployment is automated, especially with the use of a repository, such as Git. In addition, if an application cannot be containerized, this can be deployed as another pre-configured VM, if needed. It is at the hands of the designer and developer to research for the specific project needs and target and adapt the project to them.

Lastly, a file repository is needed for sharing the required, per challenge, data (i.e. files, data sets, etc.). We can fulfil this requirement using one of the below suggestions:

- Use a one-to-one (per challenger) perspective, creating for example a user related hosting service (i.e. a web server like nginx or apache).
- Create a more generic service, such as a private file sharing server, say ownCloud or nextCloud and provide the data access info to the end users.
- Use a native cloud file service such as Azure Files, OneDrive, etc.

In our case, the file server is not required for the two demo web applications, but we have deployed the ownCloud cloud application, at a MS Azure server, for future integration.

6.3 Connectivity Matters / Network and User Isolation

A logical thought is how to protect our deployed infrastructure from unauthorized access. For this reason, we utilized two MS Azure features:

i **Network Security Group**: We can use an Azure network security group to filter network traffic to and from Azure resources in an Azure virtual network. A network security group contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, we can specify source and destination IPs, port, and protocol.
In our case, we have created a new network security group, named **"Cyber-Range-NSG"** and we applied security rules that allow network access only from specific IP addresses. In this way, we protect our infrastructure (current and future VMs) at the network layer. The rules can be modified according to the different case scenarios. For example, if all the users are within a class room, we can allow access from the public IP address of the provider. Note that any newly created VM from our application are inheriting this network security group.
Lastly, the NSG can apply only to a Virtual Network. **Azure Virtual Network (VNet)** is the fundamental building block for a private network in Azure. VNet enables many types of Azure resources, such as Azure VMs, to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that someone would operate in their own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. We won't give more information at this section, as it is not necessary. Our virtual network is named **"Cyber-Range-vNET"**. An example of the specific virtual network and network security group are shown in figures 6.3 and 6.4.

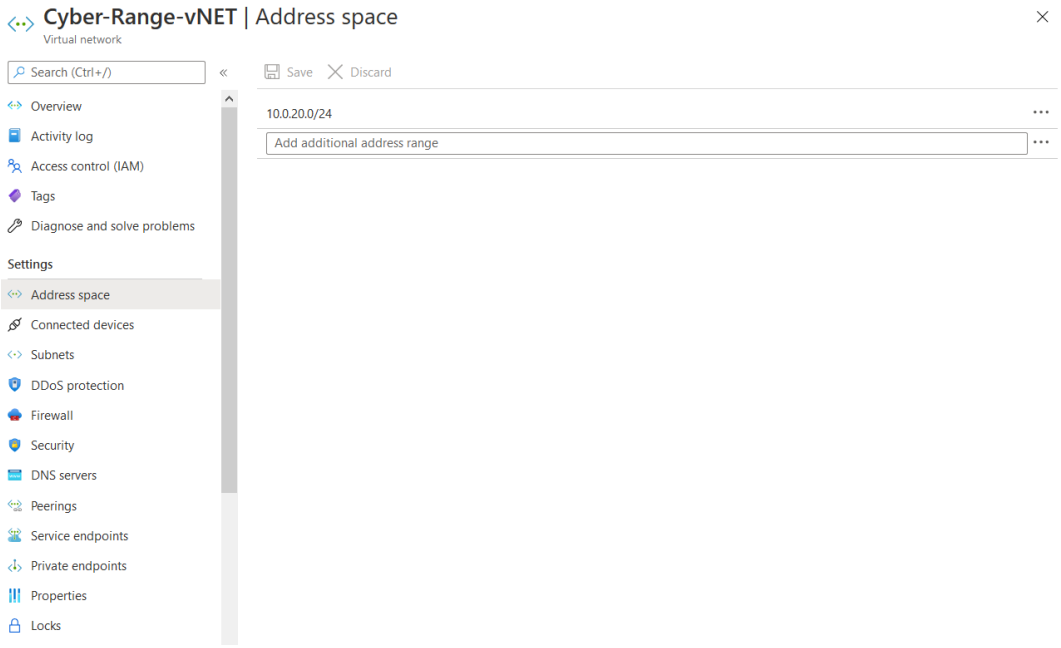

Figure 6.3: Virtual Network

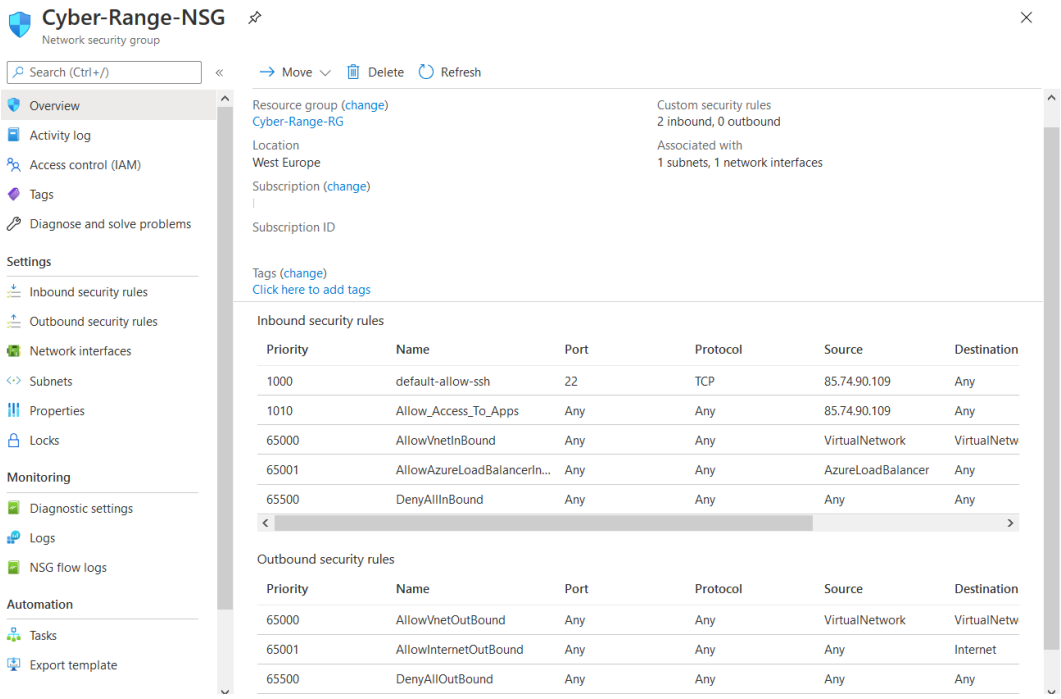

Figure 6.4: vNET Network Security Group

ii **Service Principal Account**: As an additional measure, for the communication between our local powershell scripts and MS Azure, we utilized a dedicated **Service Principal Account**, which access rights were restricted only to the specific Resource Group. In this way, we secure the access to our remaining infrastructure as well.

Automated tools that use Azure services should always have restricted permissions. Instead of having applications sign in as a fully privileged user, Azure

offers service principals.

An Azure service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources. This access is restricted by the roles assigned to the service principal, giving one control over which resources can be accessed and at which level. For security reasons, it is always recommended to use service principals with automated tools rather than allowing them to log in with a user identity.

6.4 Cyber Ranges Automatic Deployment Application

The following sections:

- Give a description of the core classes and scripts of our application.
- Describe the application from an end-user viewpoint, presenting what the user sees and interacts with.
- Examine the application code more in depth, for those who want to obtain a more detailed view of the application and/or wish to use or extend it.

6.4.1 Objects/Classes Description and operation

6.4.1.1 SSHConnection

This class utilizes the library JCraft JSch and its main purpose is to create secure shell connections to MS Azure VMs, in order to pass the required commands. In our case, the VMs are only linux Ubuntu releases (Ubuntu 18.04 LTS), so commands have also to be run in superuser mode. The SSHConnection objects needs to be initiated with the fields host (VM's public IP address), usr (username) and pass (password).

```
public SSHConnection(String host, String usr, String pass)
```

To connect to a remote VM and execute commands to it, we need fist to create an SSHConnection object and then use its method runCmd, which can be shown below:

```
public void runCmd(String[] commands, Boolean SuperUser)
```

This method takes as input a String table of the commands to be run on the server and a Boolean value declaring if the commands need to be run as superuser (root) or not.

6.4.1.2 Vulnerability Object

This object represents a vulnerability. In our environment the latter is a vulnerable application/program that will run on one (or more) VMs and will give the opportunity to the user to be familiarized and trained at each field.

Its variables represent the application's unique code, name, port that application runs at and the container name in which it is built in.

```
public class Vulnerability

   private Integer code;
   private String name;
   private Integer port;
   private String container;
```

6.4.1.3 VirtualMachine Object

This object represents a VM. Its variables represent the VM's name, public IP address, the MS Azure resource group it belongs to and a list of vulnerabilities that are to be tied to it.

```
public class VirtualMachine

   private String name;
   private String ip;
   private String resourceGroup;
   private List<Vulnerability> vulnerabilityList;
```

6.4.1.4 VirtualMachineService Class

This class can be characterized as the "heart" of our key operations, as it provides all the required functionality and communication with all the relevant components.

```
public class VirtualMachineService
```

6.4.1.4.1 Required Virtual Machine Calculation

For our code to work properly, we need a method that is in charge of calculating the number of the VMs to be deployed to the MS Azure cloud environment. The method "calculateVirtualMachines" accepts as input the vulnerable applications that are to be deployed and the username that the user entered, decides how many VMs will be created and returns a list with the newly created VMs.

```
public static List<VirtualMachine>
   calculateVirtualMachines(List<Vulnerability> vulnerabilityList,
   String username)
```

In current approach, we use a one to one deployment, meaning that we deploy one VM per user. This can be achieved due to the nature of the containerization, as we can host multiple, even similar, applications within the same host. Taking into

account the MS Azure VM sizing, we can always change the size of the deployed VM, as to cover further needs in any type of resource.

This class can be changed accordingly to the nature of the needs. For example, someone could follow another approach such as hosting one vulnerable application per host VM or select the number of deployed vulnerabilities according to the port number being used.

Despite this, the class returns a list of VM objects, as to make such changes easier to handle. Also, other classes handle data as a list of VM objects too.

6.4.1.4.2 Virtual Machine Creation at MS Azure

The method CreateNewAzureVM is responsible for deploying a new VM at MS Azure. The function's input fields are the MS Azure Resource Group that the VM will belong to and a VirtualMachine object.

```
public static void CreateNewAzureVM(String ResourceGroup,
   VirtualMachine vm)
```

The method uses the library jPowershell, to invoke the powershell script named "Create-new-VM-from-Snapshot-v3.ps1" with specific parameters, which are the MS Azure resource group that the VM will belong to and the name of the VM that will be created.

The response (output) of the script, which includes the output from MS Azure, is being further checked, as well as the received, assigned public IP address.

6.4.1.4.3 Start Vulnerabilities at Remote VM

The "startVulnerabilities" function's aim is to connect to a remote MS Azure VM through SSH and start the necessary containers, through the relevant commands, as for the vulnerable applications to start. The function's input is a VirtualMachine object and in case of a successful operation, the method returns true, otherwise it returns false.

```
public static boolean startVulnerabilities(VirtualMachine vm)
```

At first, we perform a check on the received VM public IP, using the method "isValidIP" and we only continue if the public IP address has an acceptable form. To pass the commands to the remote VM, we use the external library JSch, through our Java object "SSHConnection". The code also features a retry mode (five retries with 20 sec between each of them), as to overcome any temporary connection errors.

The commands are predefined and start the needed containers, according to user's choices made at the front-end (UI).

6.4.1.4.4 Remote Virtual Machine Shut Down

The method "ShutDownAzureVM" handles the safe shut down of a MS Azure VM. The function's input is a VirtualMachine object and in case of a successful operation,

the method returns true, otherwise it returns false.

```
public static boolean ShutDownAzureVM(VirtualMachine vm)
```

To pass the commands to the remote VM, we use the external library JSch, through our Java object "SSHConnection". The code also features a retry mode (five retries with 20 sec between each of them), as to overcome any temporary connection errors.

6.4.1.4.5 Remote Virtual Machine Deletion

Similarly to the VM creation function, we have created the remote VM deletion function, named "DeleteAzureVM". The function's input is a VirtualMachine object and in case of a successful operation, the method returns true, otherwise it returns false.

```
public static boolean DeleteAzureVM(VirtualMachine vm)
```

The method uses the library jPowershell, to invoke the powershell script named "ConnectToAzureAndDeleteVM-v3.ps1" with specific parameters, which are the MS Azure resource group that the VM belongs to and the name of the VM that will be deleted.

The response (output) of the script, which includes the output from MS Azure, is being further checked using the method "getVMdeletionResults". If there is a successful deletion of the resources, the method returns the value true, otherwise it returns false.

We have to note here that simply deleting a remote VM from MS Azure does not completely delete all the related allocated resources, but just the VM itself. Other resources such as the VM's disk, network interface and public IP address have also to be deleted, in order first to have a clear environment and second, to avoid the cost charged for those components. Our powershell script covers this case and all allocated resources are being released.

6.4.1.4.6 Clearing all MS Azure Resources

This is a combinational method which takes as input a list of VM objects, which represent the list of MS Azure deployed VMs and is responsible for clearing/deleting all the allocated resources. This operation is split in two parts for each VM:

- VM Shut Down: We call the method "ShutDownAzureVM" for shutting down the VM.
- VM Resources Deletion: We call the method "DeleteAzureVM" so as to delete the VM, along with all its components.

```
public static void clearResources(List<VirtualMachine>
   virtualMachines)
```

The operation continues for all VMs of the input list (multiple VMs can be handled).

6.4.1.4.7 Handle Virtual Machine Creation Result

When we create a new VM at MS Azure cloud, using the appropriate powershell script, we get a result response from MS Azure itself. This output contains some codes that need to be examined, as to define whether the operation was successful or not. The method "getVMCreationResults" does exactly this, by examining the input response.

```
public static String getVMCreationResults(String scriptOutput)
```

The method also returns the newly allocated public IP address of the VM. If the operation did not go as scheduled, for any reason, it returns null.

6.4.1.4.8 Handle Virtual Machine Deletion Result

In a similar way to creation procedure, when a VM is deleted at MS Azure cloud, using the appropriate powershell script, we get a result response from MS Azure itself. This output contains some codes that need to be examined, so as to define whether the operation was successful or not. The method "getVMDeletionResults" does exactly this, by examining the input response.

```
public static boolean getVMdeletionResults(String scriptOutput)
```

The method also returns true if the operation did go as scheduled, otherwise it returns false.

6.4.1.4.9 Printing Utility

For troubleshooting/debugging purposes, we have implemented the method "printAllVMs", which takes as input a list with VM objects and prints all the involved information.

```
public static void printAllVMs(List<VirtualMachine>
   runningVirtualMachines)
```

6.4.1.4.10 Check IP Address Validity

An additional method named "isValidIP" is used for checking the validity of an IP address. If the input variable is a valid IP address, the method returns true, otherwise it returns false.

```
public static boolean isValidIP (String ip)
```

6.4.1.5 Powershell Scripts

At previous steps we referred to two PowerShell scripts that are being called from the main JavaFX application. One of those handles the remote creation of the required VMs at MS Azure and the second one handles the remote deletion of existing VMs and their components from MS Azure.

6.4.1.5.1 Virtual Machine Creation PowerShell Script

We use the PowerShell script named “Create-new-VM-from-Snapshot-v3.ps1” with the below parameters as input:

- The MS Azure Resource Group that the VM will belong to.
- The name of the VM that will be created (in our case the VM to be created is named after the username field followed by a small addition from our side).

The script has also a flag (switch to be more accurate) named "Wait", that can be enabled if one wishes to wait until the script finishes.

The script, in short, performs the following actions:

i Creates the required connection components (such as the used account and MS Azure tenant info).

ii Copies the predefined snapshot to a new, managed, hard disk object, using the new VM name.

iii Sets some required components such as the disk size and the location (Data Centre Region) of the VM, the VM’s Virtual Network (vNET) and size.

iv Retrieves a new, dynamic IP address from the cloud provider and assigns it to a new network interface controller (NIC).

v Creates the new VM, using the configuration built at previous steps.

vi Retrieves the new VM’s public IP address.

The response (output) of the script, which includes the output from MS Azure and the VM’s public IP address, is then passed to the main application, which should examine it for possible errors.

6.4.1.5.2 Virtual Machine Deletion PowerShell Script

We use the PowerShell script named “ConnectToAzureAndDeleteVM-v3.ps1” with the below parameters as input:

- The MS Azure Resource Group that the VM belongs to.
- The name of the VM that will be deleted.

This script also includes the Wait flag, that is enabled, waiting until the script finishes.

The script, in short, performs the following actions:

- Creates the required connection components such as the used account and MS Azure tenant info,
- Connects to MS Azure and retrieves the VM's resources, namely VM, disk, NIC, Public IP address.
- Deletes all the resources and releases the dynamic, public IP address.

The response (output) of the script is again passed to the main application, which should examine it for possible errors/failures.

7 Model Usage Example

The application is created utilizing a front-end User Interface, written in JavaFX as a Maven project. The source code is organized per screen into FXML files along with the respective Java controller files.

JavaFX controller works based on MVC (Model-View-Controller). JavaFX MVC can be achieved by FXML (EFF-ects eXtended Markup Language). FXML is an XML based language used to develop the GUIs, for JavaFX applications as in the HTML. FXML can be used to build an entire GUI application scene or part of a GUI application scene.

Each screen that the user interacts with, is related with a Java controller code file that includes the source code that handles the user input, the operations instructed, etc.

More specifically in our case, we present four screens to the end user, which are described briefly at the sections below. An extensive description on back-end operation procedures will follow.

7.1 First Screen - Welcome

Figure 7.1 depicts the first screen that the user interacts with. It contains just some info about the application itself and provides the user with two simple options:

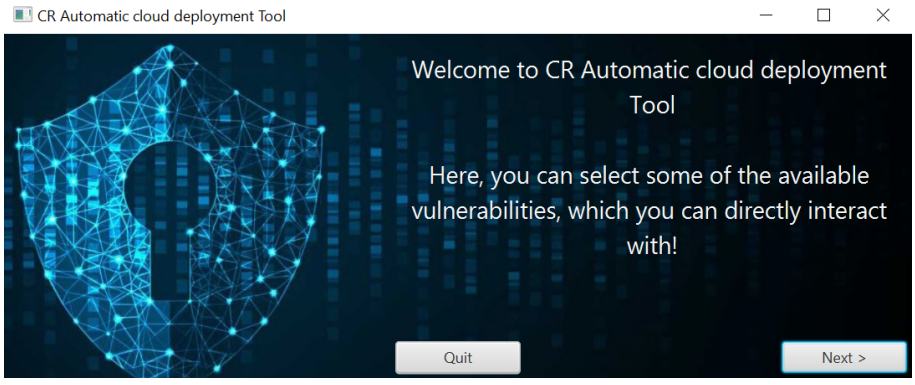

Figure 7.1: First Screen

- "Next": This option will navigate the user to the second application screen.
- "Quit": This option will present a confirmation window for the user confirmed exit, as depicted in figure 7.2, and if the user selects "Yes", the application will terminate.

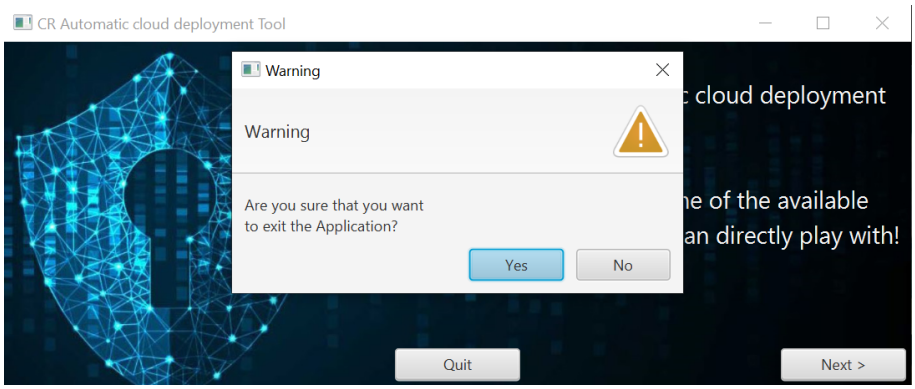

Figure 7.2: Quit Option

7.2 Second Screen – User Options

At the second screen illustrated in figure 7.3, the end user is being presented with a checkbox list with all the available vulnerable applications. This is a dynamically loaded list, meaning that no additional work is needed to handle this window, if additional vulnerable applications are added to the application. The user can select one or more from the challenges (i.e. vulnerable applications) that they wish the main application to create for him, as to be familiarized with.

Figure 7.3 depicts the current vulnerable application options, where there is an also prompt for the user to enter a username that will separate them from other users that simultaneously use the application. The user can also see the requirements for the entering field and it is “five to 10 letters (no blanks)”. This is, again, something very easy for a developer to change, according to the needs of the organizer or event, as the username validity check is structured in a method named “isValidUsername”.

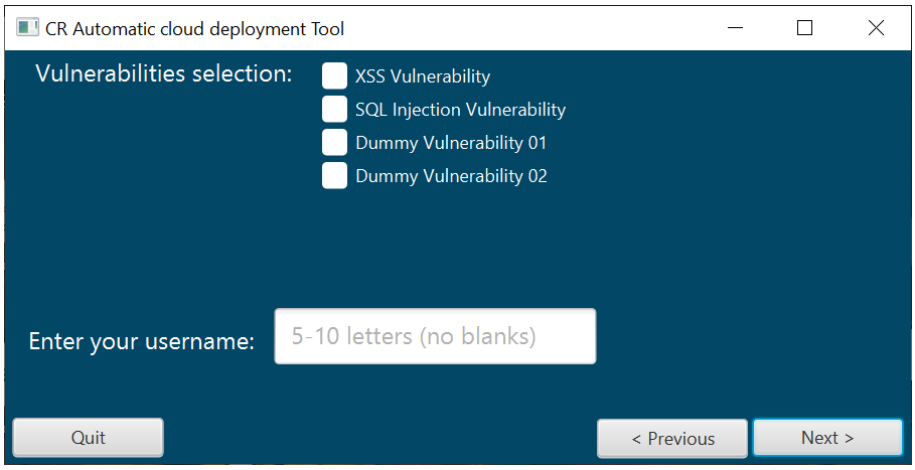

Figure 7.3: Second Screen

Note that if the user:

- Does not select at least one vulnerable application from the list or
- Does not enter a valid user name or

- A combination of the above

Then they will get an alert window, informing them on the error occurred and how he can rectify it. This situation is given in figure 7.4.

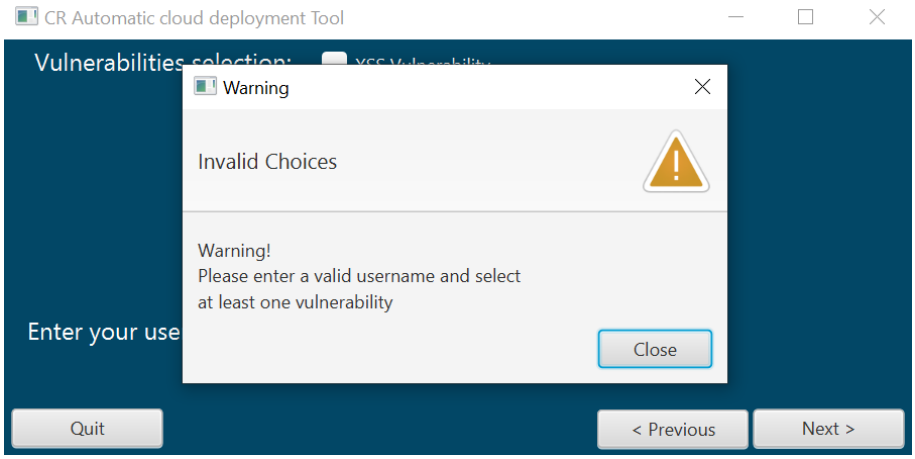

Figure 7.4: Invalid Options

Lastly, the user will see three buttons at the lower side of the window:

- "Next": this option will validate the user-entered options and according to them, the application will create the entire required infrastructure at the MS Azure cloud environment, it will deploy the vulnerable applications and then it will navigate the user to the third application screen. While the user waits for the MS infrastructure to be created and the applications to start, they face a "Please wait. . . " window, as to ensure them that the application is still running. This situation is given in figure 7.5.

Figure 7.5: Wait Screen

- "Previous": This option will navigate the user to the first (initial) screen.
- "Quit": It will present a confirmation window for the user confirmed exit and if the user selects "Yes", the application will terminate.

Figure 7.6 depicts a typical selection and a typical username for demonstration purposes:

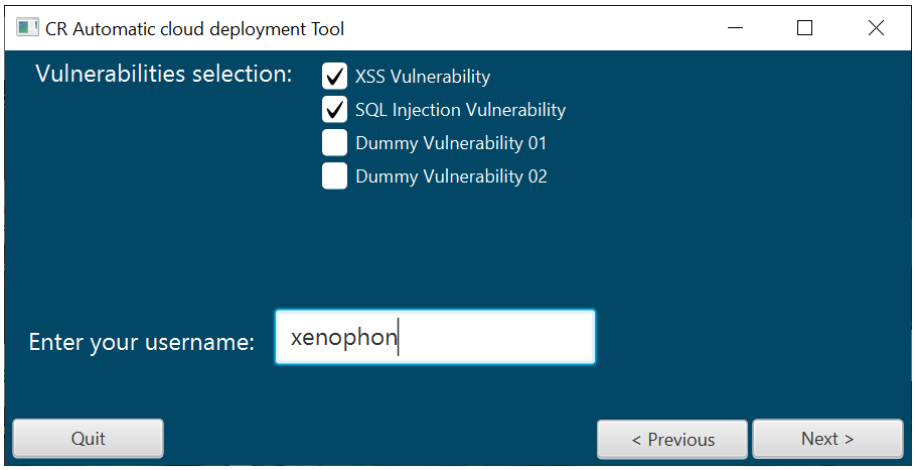

Figure 7.6: Typical Entries

7.3 Third Screen – Vulnerable Applications Info

As the required infrastructure is successfully deployed at MS Azure and the relevant vulnerable applications have been deployed too, the user is presented with a table which contains, for each selected vulnerable application:

- **Vulnerability Name**: The name of the vulnerable application deployed, as presented at the second screen.
- **Vulnerability URL**: The challenge (vulnerable application) URL. The end user can use this URL and a web browser in order to navigate to the vulnerable application. There they can follow the guidelines of the trainer or the application itself. In this way they will be able to be familiarized, exploit any vulnerabilities available, etc.

The outcome of the previous example is shown in figure 7.7:

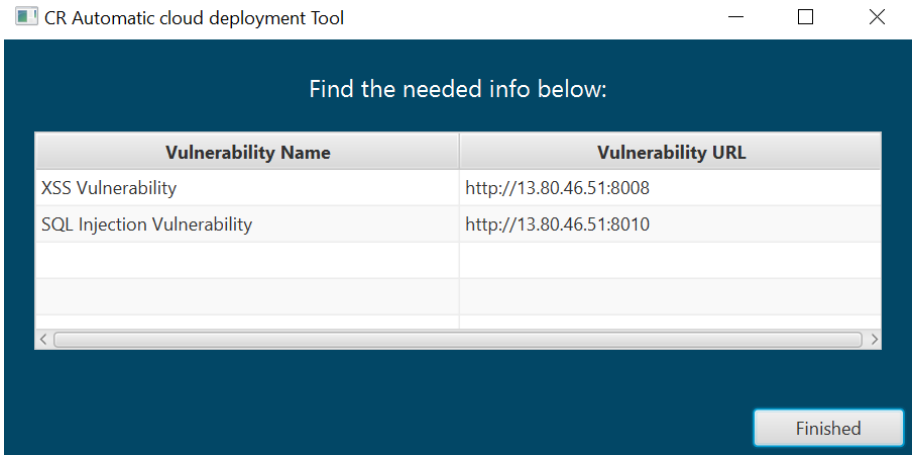

Figure 7.7: Challenges URLs

The result (screenshots) when the end user uses a web browser and navigates to the vulnerable applications, using the URLs is demonstrated in figures 7.8, 7.9, 7.10, 7.11, 7.12 and 7.13.

Vulnerable Application 1 – XSS Vulnerability

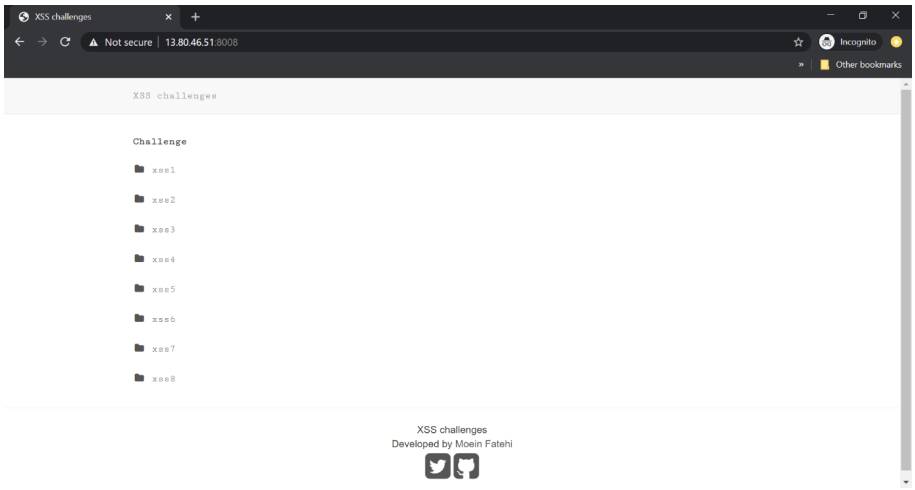

Figure 7.8: XSS Challenge Page 1

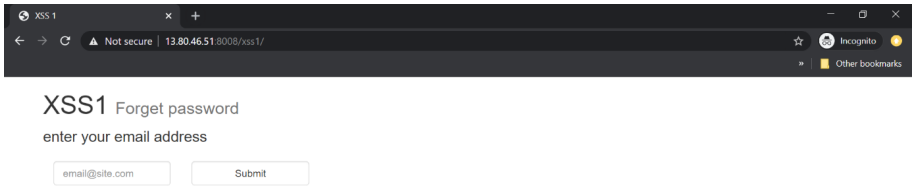

Figure 7.9: XSS Challenge Page 2

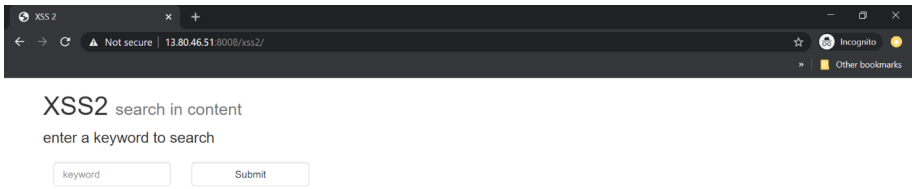

Figure 7.10: XSS Challenge Page 3

Vulnerable Application 2 – SQL Injection

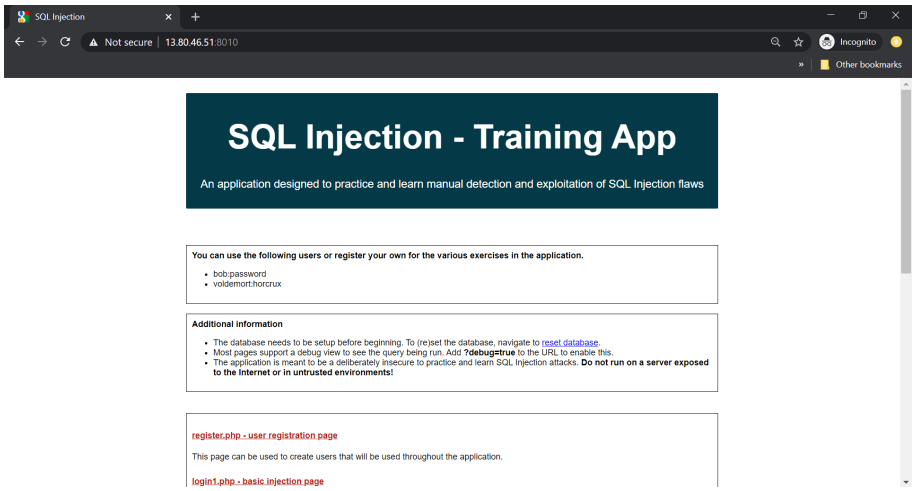

Figure 7.11: SQL Injection Challenge Page 1

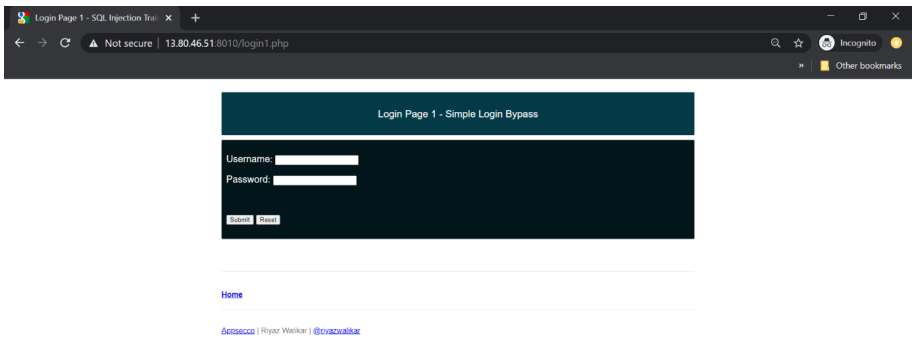

Figure 7.12: SQL Injection Challenge Page 2

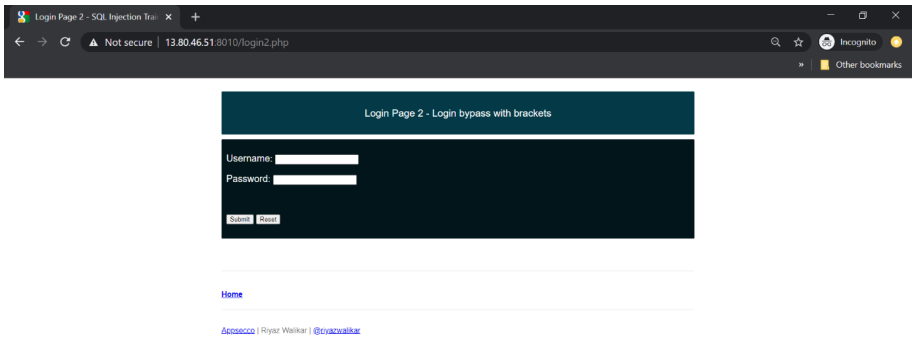

Figure 7.13: SQL Injection Challenge Page 3

Note that, as the application has now fulfilled the user side requirements, there is no any other option available than the “Finished” button. This is illustrated in figure 7.14.

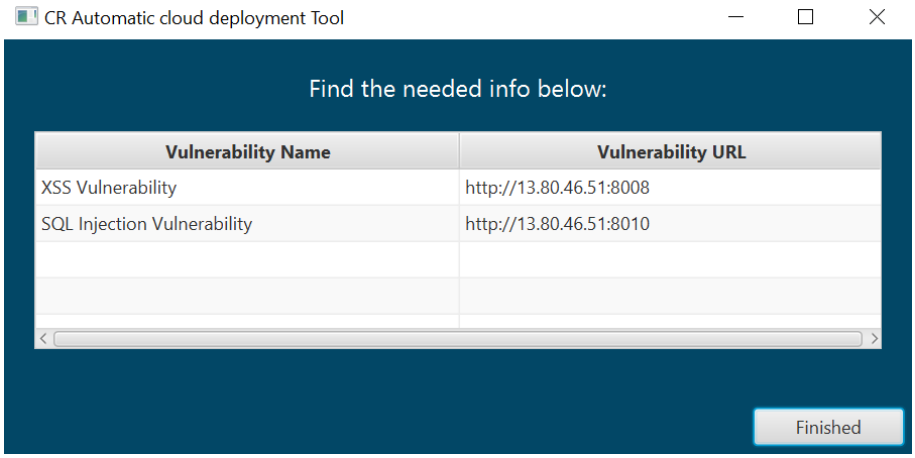

Figure 7.14: Challenges URLs

- "Finished": the application will clear the entire, previously deployed, infrastructure at the MS Azure cloud environment and afterwards it will navigate the user to the fourth application screen. A confirmation window will enforce the user to confirm his intention to terminate the application, as depicted in figure 7.15.

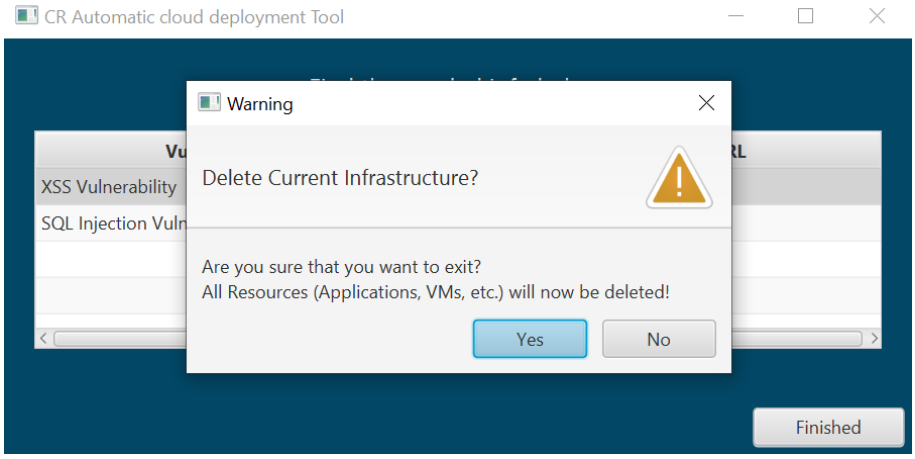

Figure 7.15: Finished Button Pressed URLs

After the user presses "Yes" and while the user waits for the various components to stop and the MS infrastructure to be cleared, the user faces again, a "Please wait. . . " window, as to ensure him that the application is still running. This is illustrated in figure 7.16.

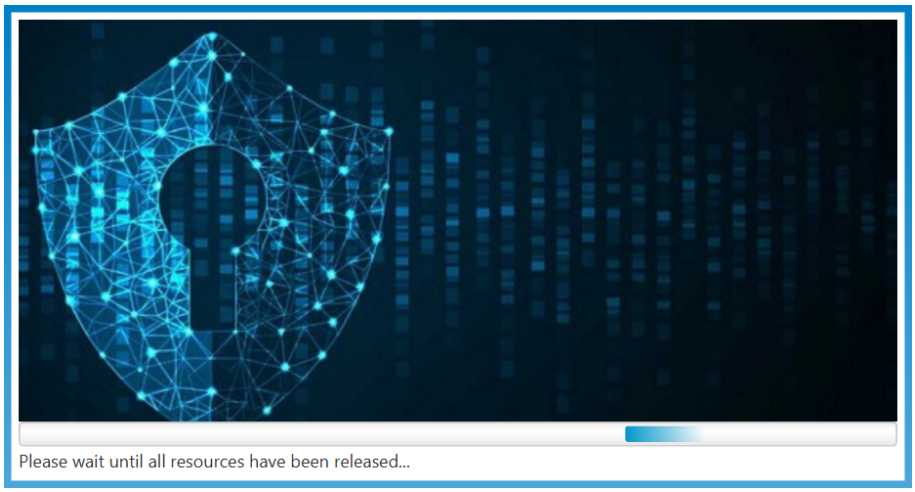

Figure 7.16: Please Wait Screen

7.4 Fourth Screen - End

When the MS Azure resources are being released, the users will face a final screen, which only contains a general “Thank you” message, as depicted in figure 7.17 and provides them with two options:

- “Start Again”: This option will navigate the user back to the first/initial application screen, in case he wants to start from the beginning.
- “Exit”: This option will present a confirmation window for the user confirmed exit, as depicted in figure 7.18 and if the user selects “Yes”, the application will terminate.

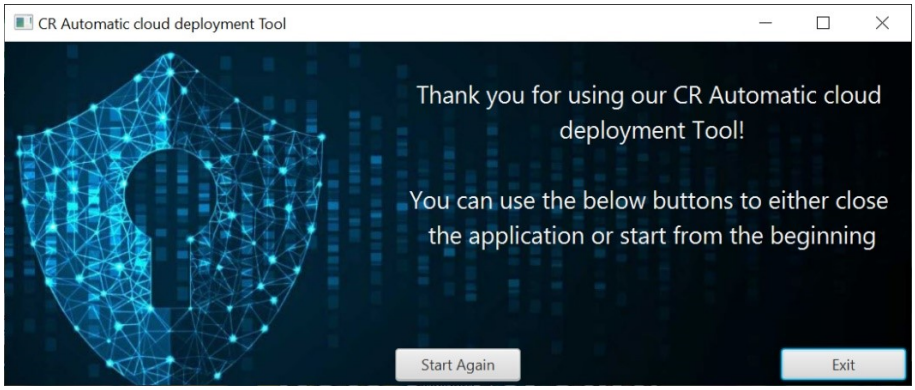

Figure 7.17: Fourth Screen

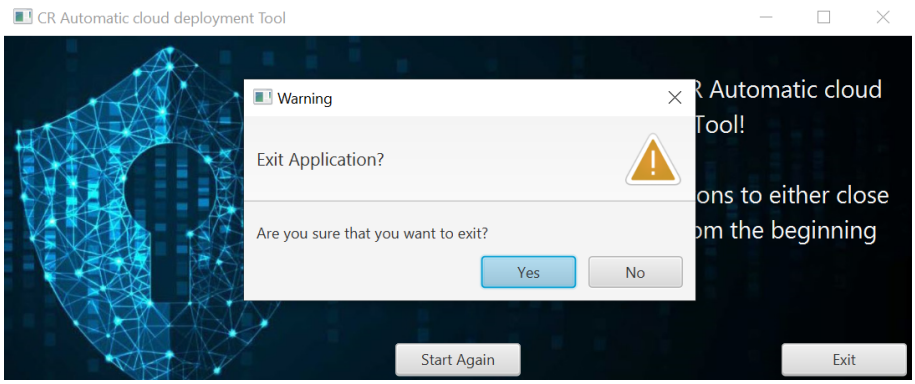

Figure 7.18: Exit Screen

8 Application Back-End Description & Flow

As we already mentioned in **chapter 6**, each application screen is connected with the relevant controller. At this part of our work we will describe the functionality of the four controllers in more detail. The names of the controllers are similar to the respective FXML files.

8.1 First Screen Controller

The first screen provides only some informational text to the user, so the source code is simple.

```
public class FirstScreenController
```

The button “Next” calls the method “switchToSecondScreen”, which navigates the user to the second screen.

```
private void switchToSecondScreen()
```

The button “Quit” will call the method “quit” which creates an overlay window asking for confirmation on the exit action (“Are you sure that you want to exit the Application?”). If the end user presses the “No” button, then the confirmation window will close and the application will remain active. Alternatively, if the “Yes” button is pressed, then the confirmation window will close and the main application will terminate too.

```
public void quit(ActionEvent actionEvent)
```

8.2 Second Screen Controller

The second screen provides most of the building functionality of the main application, as it makes calls towards the “VirtualMachineService” class.

```
public class SecondScreenController implements Initializable
```

This class implements controller initialization interface. As per Java documentation, this interface has been superseded by automatic injection of location and resources properties into the controller. FXMLLoader will automatically call any suitably annotated no-arg initialize() method defined by the controller. This means that when

the second controller source code runs, the method “initialize” is called to initialize a controller after its root element has been completely processed. In our case, more specifically, this method loads the vulnerable applications information data into the view of the end user in a list of clickable check boxes.

Consequently, the end users can make their choices on the vulnerable applications check boxes, by ticking them directly and enter a valid username, according to the directions presented.

The button “Next” calls the method “getUserResponse”:

```
public void getUserResponse(ActionEvent actionEvent)
```

, which performs the following actions:

- We collect and validate user’s choices and username entered field. With the assistance of the method:

  ```
  public boolean isValidUsername(String enteredString)
  ```

  we perform the username check. Additionally, if the user:

  - did not select at least one vulnerable application from the list or
  - did not enter a valid user name or
  - we have a combination of the above

  , then a Java native alert window appears, declaring the exact nature of the problem and prompting the user to make acceptable choices.

- If all previous checks are successful, we call the method “calculateVirtualMachines” (of the VirtualMachineService class):

  ```
  VirtualMachineService.calculateVirtualMachines(selectedVuln,
     userName)
  ```

  This step is needed for calculating the required resources (VMs) that need to be created at MS Azure. As a result, a list of VM objects is returned,

- The next step is to read the result from previous step (calculated VM list) and according to it, to create all the required VMs at MS Azure and start the selected containers (vulnerable applications). As this consecutive operation would “freeze” the user interface and confuse the end user if we follow a traditional implementation, we choose to implement this component as a Java Task which runs at a separate, new Thread.

  Tasks are used to implement the logic of work that needs to be done on a background thread. First, we need to extend the Task class. The implementation of

the Task class must override the call method to do the background work and return the result.

The call method is invoked on the background thread, therefore this method can only manipulate states that are safe to read and write from a background thread.

The Task we refer to is the following:

```
Task<Void> createVMsTask = new Task<Void>()
```

As noted before, the call method is invoked:

```
protected Void call()
```

For each Virtual Machine object that needs to be built, the “CreateNewAzureVM” method (of the VirtualMachineService class) is called.

```
VirtualMachineService.CreateNewAzureVM(usedResourceGroup,vm)
```

This method calls a PowerShell script named ’Create-new-VM-from-Snapshot-v3.ps1’, which connects to the MS Azure for creating the required VMs.

Afterwards, the software waits for some time for each VM to be created, it performs a check if all the VMs are created successfully and then it starts the vulnerable applications selected by the user by using the “startVulnerabilities” method (of the VirtualMachineService class).

```
public static boolean startVulnerabilities(VirtualMachine vm)
```

After the Task is defined, we have to call it in a separate Thread, as to avoid freezing of the UI. This is done using the following line:

```
new Thread(createVMsTask).start();
```

Furthermore, in order to give the user an essence of the background work’s processing, an “On Progress” Splash Stage (window) is presented, and while the background operations are running, the user only sees the window given in figure 8.1.

```
Stage splashStage = new Stage();
```

Figure 8.1: Please Wait Screen

This overlaid window is displayed until the VM creation Java Task is finished and it is implemented through the following method, which is defined at the "Util" java file.

```
public static void showSplash(final Stage initStage,
                    Task<?> task,
                    InitCompletionHandler
                       initCompletionHandler, String text)
```

- Lastly, if the Task finishes successfully, then the main application navigates the user to the third screen presented in figure 8.5.

The button "Quit" will call the method "quit" which creates an overlay window asking for confirmation on the exit action ("Are you sure that you want to exit the Application?"). If the end user presses the "No" button, then the confirmation window will close and the application will remain active. Alternatively, if the "Yes" button is pressed, then the confirmation window will close and the main application will terminate too.

```
public void quit(ActionEvent actionEvent)
```

At this point, figure 8.2 demonstrates the infrastructure administrator's view at MS Azure portal, where we can recognize the newly created VM, named "xenophon-vm1":

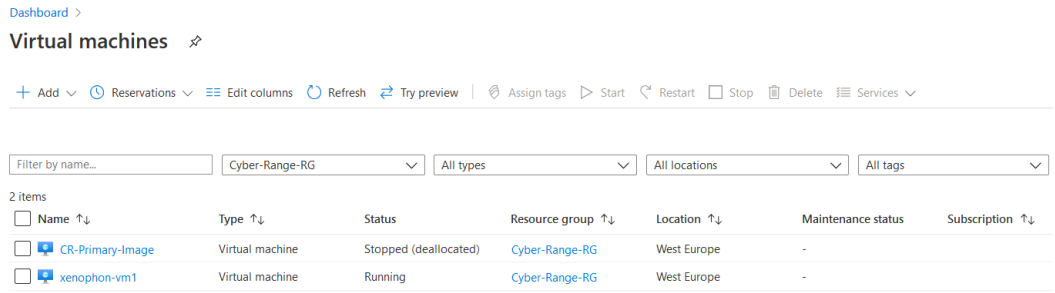

Figure 8.2: MS Azure Portal - Virtual Machines

In figure 8.3, we can also verify that the components used are the predefined ones (Resource Group, Size, Virtual Network, OS Disk Name, Location):

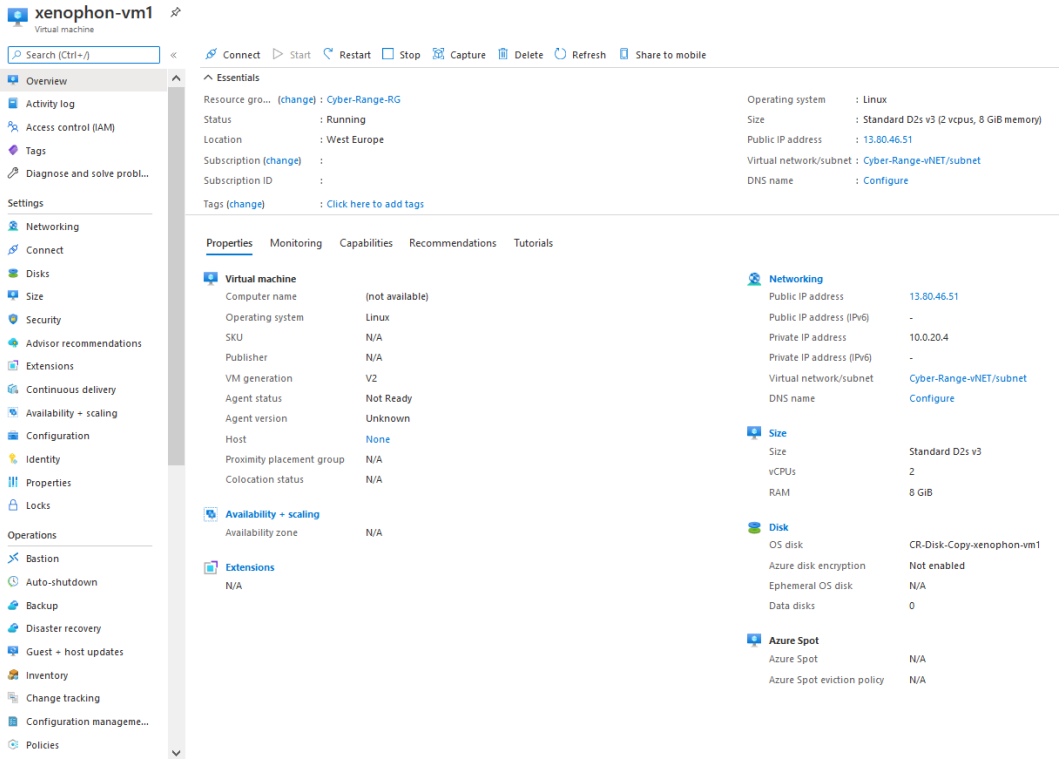

Figure 8.3: MS Azure Portal - Virtual Machine Details

In figure 8.4, we can also see that the new VM inherited the default network's Network Security Group, which applies preconfigured network rules to the VM:

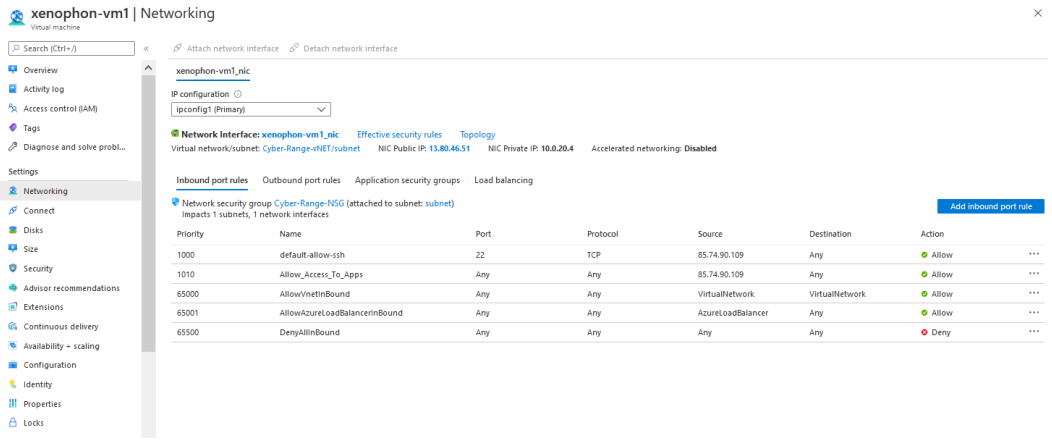

Figure 8.4: MS Azure Portal - VM Network Properties

8.3 Third Screen Controller

As at previous screens/steps the required infrastructure is created at MS Azure and the respective docker containers (vulnerable applications) are started, the third screen of the application provides the end user with the relevant information.

```
public class ThirdScreenController implements Initializable
```

This class also implements controller initialization interface. The method “initialize” is called to initialize the controller.

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

The initialization method loads the vulnerable applications information, which is necessary for the end user, into a table view as depicted in figure 8.5. In this table, the user is presented with the all vulnerable applications’ Name and Access URL.

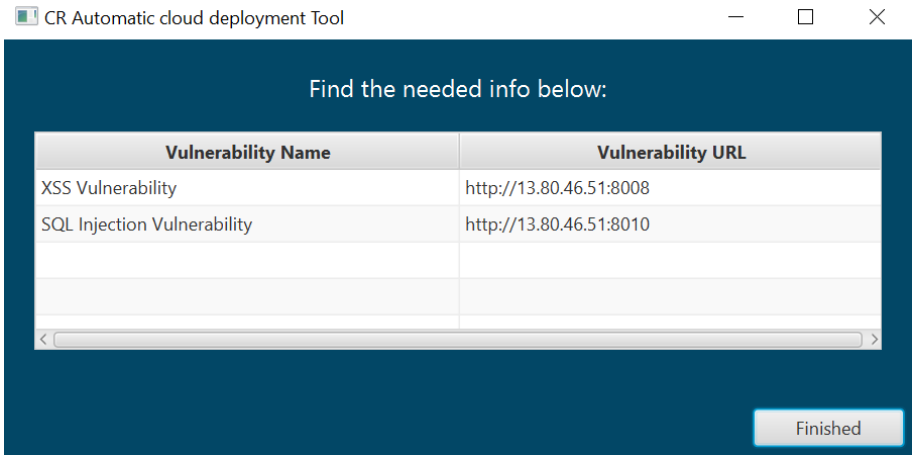

Figure 8.5: Challenges URLs

At current application screen, only the “Finished” button is available to the end user. This assumes that the user has finished their work with the application. Pressing the “Finished” button, the application will call the method “finished”:

```
public void finished(ActionEvent actionEvent)
```

, which creates an overlay window asking for confirmation on the finished action (“Are you sure that you want to exit? All Resources (Applications, VMs, etc. will now be deleted!)”). If the end user presses the “No” button, then the confirmation window will close and the application will remain active, maintaining the created infrastructure. Alternatively, if the “Yes” button is pressed, then the confirmation window will close and the resources will be released as per below procedure:

- The list of the current utilized VMs will populate.
- In a similar way as implemented at the previous controller, a Java Task will run, as not to freeze the user interface. The Task we refer to, is the following:

  ```
  Task<Void> clearResourcesTask = new Task<Void>()
  ```

  As noted before, the call method is invoked:

  ```
  protected Void call()
  ```

  Afterwards, the “clearResources” method (of the VirtualMachineService class) is called.

  ```
  VirtualMachineService.clearResources(virtualMachines);
  ```

  Recall that the above-mentioned method triggers a two-step procedure for each VM involved:

  - Shut down the remote VM, by using the method "ShutDownAzureVM" of the VirtualMachineService class.

    ```
    public static boolean ShutDownAzureVM(VirtualMachine vm)
    ```

  - Full deletion of the VM allocated resources (VM, disk, NIC, Public IP Address), by using the method "DeleteAzureVM" of the VirtualMachineService class.

    ```
    public static boolean DeleteAzureVM(VirtualMachine vm)
    ```

    This method calls a PowerShell script named ’ConnectToAzureAndDeleteVM-

v3.ps1', which connects to the MS Azure in order to delete the relevant VMs' components.

After the Task is defined, we have to call it in a separate Thread, as to avoid freezing of the UI. This is done using the following line:

```
new Thread(clearResourcesTask).start();
```

As in previous section, a "Please wait until all resources have been released. . . " Splash Stage (window) is presented and while the background operations are running, the user only sees the window given in figure 8.6.

```
Stage splashStage = new Stage();
```

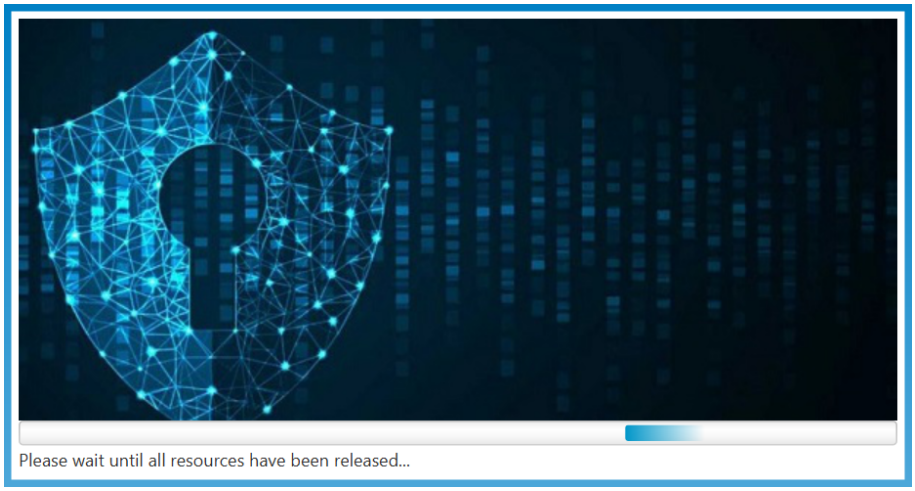

Figure 8.6: Please Wait Screen

Again, this overlaid window is displayed until the VM resources release Java Task is finished.

- Lastly, if the Task finishes successfully, then the main application navigates the user to the fourth screen shown in figure 8.7.

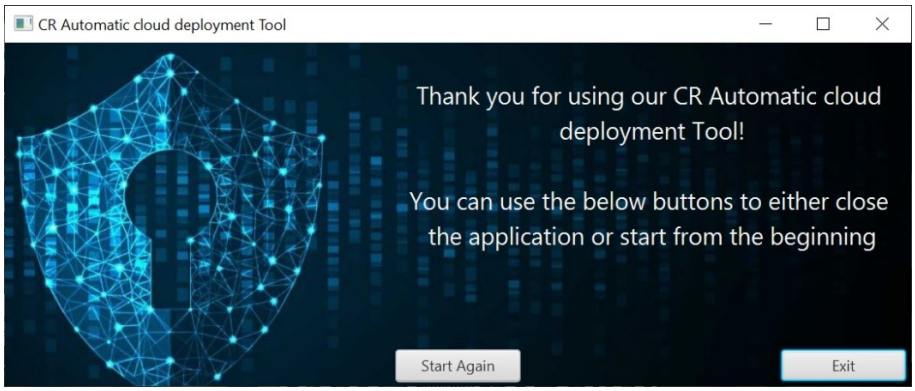

Figure 8.7: Fourth Screen

Again, figure 8.8 illustrates the infrastructure administrator's point of view at MS Azure portal, while the script has initiated the VM deletion process:

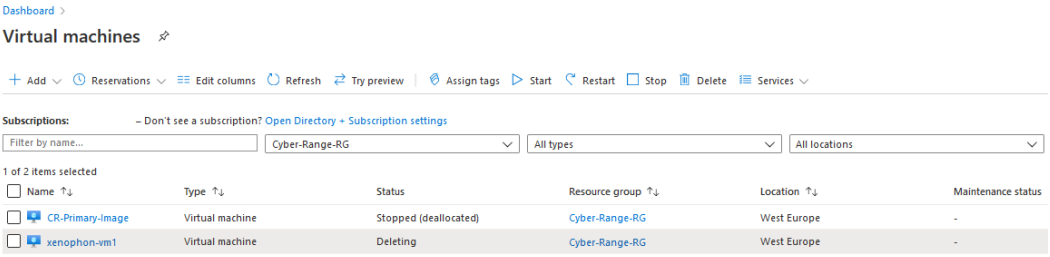

Figure 8.8: MS Portal - VM Deletion

After the process finishes, no signs of the VM resources appear at any resources view dashboard.

8.4 Fourth Screen Controller

This is the fourth and last screen of the application and only presents some post action, informational text to the user ("Thank you for using our CR Automatic cloud Deployment Tool"), so again the source code is simple. There is also additional text that prompts the user whether they wish to exit the application, by pressing the "Exit" button or they wish to start over, by pressing the button "Start Again".

```
public class FourthScreenController
```

The button "Start Again" calls the method "startAppAgain", which navigates the user to the first screen of the application, where they can start using the application from the beginning.

```
public void startAppAgain(ActionEvent actionEvent)
```

The button “Exit” will call the method “exitApp” which creates an overlay window asking for confirmation on the exit action (“Are you sure that you want to exit?”). If the end user presses the “No” button, then the confirmation window will close and the application will remain active. Alternatively, if the “Yes” button is pressed, then the confirmation window will close and the main application will terminate too.

```
public void exitApp(ActionEvent actionEvent)
```

9 Conclusions and Future Work

9.1 Conclusions

In this MSc thesis, we have presented the need for cybersecurity training and awareness. We introduced some basic information about Cyber Ranges and Cloud Computing, which are critical components for our work, and we described the generic design life cycle for testbeds presented by the research work authors, including typical design considerations for cybersecurity testbeds.

We presented the scope of our work that is, a design model which extends and validates the given life cycle model utilizing a cloud service provider, Microsoft Azure in our case and automation procedures. All in all, this work designed and implemented a fully automated application for the end users that provides them with a variety of types of security challenges, in order for them to assess their cybersecurity knowledge and training needs.

We detailed on the technologies used and on the ways we implemented our model and gave a high level description of the developed code. The end user model usage example and the back-end description were also described.

The contribution of this work is initially for the system manager/administrator, who will only have to monitor the automated procedures health status, instead of creating and deleting the required cloud resources. Additionally, the system manager does not have to worry about the required resources, as those will always be adequate, up to the cloud service provider's capabilities. No large local infrastructure is required. Furthermore, the admin can always monitor the application's result actions at Microsoft Azure portal, under the relevant Resource Group. With a quick view, they can understand the current status and identify potential problems. We also must mention that upon completion of a training course, the whole infrastructure components are being deleted, releasing the utilized resources and reducing the operational cost. Lastly, as we refer to a 'access restricted' environment, administrators can feel that their environment is secure from external threats.

Another contribution is for the trainer who will only have to manage and maintain the current/future challenges, according to the training needs. Precisely, the trainer can design and deploy challenges in a variety of training fields, taking into consideration the target audience.

Lastly, the contribution for the trainees who participate is providing them with an easy to use application, which enables them to select the training fields (challenges) that they wish to get into and those will be automatically deployed on the cloud environment, without the need of any third party.

The main focus of this thesis is not only the Cyber Range automatic deployment by means of cloud computing, but also the provision of administrator/trainer/trainee

with a user-friendly environment and expose newer technologies to a larger audience.

Of course, further study of each use case is needed, so as to customise the application to the required standards.

9.2 Future Work

This thesis presented an innovative model design and its implementation for automated cloud cyber range deployment.

The implementation though, is a pilot based one. Consequently, some improvements/optimizations or additional utilities for both the trainee and the trainer can be developed. For example, some cases might not be handled in a strict, professional manner (i.e. hardcoded sensitive info).

We conducted many experiments that proved the correct, concurrent, VMs creation for multiple users and the proper functionality of each of those.

The multiple VM creation is supported, and one just has to modify one method for producing the required VMs, making our application very agile.

The users could also have the ability to move from the third application screen, back to the second one, so as to modify their challenge choices.

During this work we also implemented some output logging at text files, but this was mainly for debugging purposes. This can be extended by creating a basic logging service which collects logs from all different logging procedures and either integrates them in one single file or processes it for further parsing and decision making.

10 Bibliography

[1] M. Frank, M. Leitner, and T. Pahi, "Design considerations for cyber security testbeds: A case study on a cyber security testbed for education," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2017, pp. 38–46.

[2] "NIS Directive," 2016. [Online]. Available: http://data.europa.eu/eli/dir/2016/1148/oj

[3] "The European Union Agency for Cybersecurity (ENISA)." [Online]. Available: https://www.enisa.europa.eu

[4] "Enisa supporting Cybersecurity Exercises." [Online]. Available: https://www.enisa.europa.eu/topics/cyber-exercises

[5] "EU Cybersecurity Act, REGULATION (EU) 2019/881 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 april 2019." [Online]. Available: https://eur-lex.europa.eu/eli/reg/2019/881/oj

[6] NIST, "Cyber ranges." [Online]. Available: https://www.nist.gov/system/files/documents/2018/02/13/cyber_ranges.pdf

[7] "European Cyber Security Organisation." [Online]. Available: https://www.ecs-org.eu/

[8] European Cyber Security Organisation, "Understanding Cyber Ranges: From Hype to Reality." [Online]. Available: https://www.ecs-org.eu/documents/uploads/understanding-cyber-ranges-from-hype-to-reality.pdf

[9] Gartner, "Digital Dexterity," *Gartner Digital Workplace Summit*, 2020. [Online]. Available: https://www.gart-ner.com/en/conferences/na/digital-workplace-us/featured-topics/digital-dexterity

[10] NIST, "Developing Cyber Resilient Systems: A Systems Security Engineering Approach," 2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft-fpd.pdf

[11] Gartner, "Organizational Resilience Is More Than Just the Latest Trend," 2018. [Online]. Available: https://www.gartner.com/en/documents/3875514/organizational-resilience-is-more-than-just-the-latest-t

[12] ENISA, "European Cyber Security Challenge," 2020. [Online]. Available: https://europeancybersecuritychal-lenge.eu

[13] WorldSkills, "WorldSkills," 2019. [Online]. Available: https://worldskills.org

[14] CyberStars, “Cyber Stars,” 2019. [Online]. Available: https://www.cyberstars.pro

[15] “US National Initiative for Cybersecurity Education.” [Online]. Available: https://www.nist.gov/itl/applied-cybersecurity/nice/about

[16] US National Initiative for Cybersecurity Education, “The Cyber Range: A Guide.” [Online]. Available: https://www.nist.gov/system/files/documents/2020/06/25/The%20Cyber%20Range%20-%20A%20Guide%20%28NIST-NICE%29%20%28Draft%29%20-%20062420_1315.pdf

[17] “Global Environment for Network Innovations (GENI).” [Online]. Available: http://www.geni.net/about-geni/what-is-geni/

[18] M. M. Yamin, B. Katt, and V. Gkioulos, “Cyber ranges and security testbeds: Scenarios, functions, tools and architecture,” *Computers & Security*, vol. 88, p. 101636, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404819301804

[19] IBM, “Virtualization,” 2019. [Online]. Available: https://www.ibm.com/cloud/learn/virtualization-a-complete-guide

[20] Gartner, “6 best practices for creating a container platform strategy,” 2019. [Online]. Available: https://www.gartner.com/smarterwithgartner/6-best-practices-for-creating-a-container-platform-strategy/

[21] “Openstack.” [Online]. Available: https://www.openstack.org/

[22] “Open Nebula.” [Online]. Available: https://opennebula.io/

[23] European Defence Agency, “Cyber Ranges Federation Project reaches new mile-stone,” 2018. [Online]. Available: https://www.eda.europa.eu/info-hub/press-centre/latest-news/2018/09/13/cyber-ranges-federation-project-reaches-new-milestone

[24] ECHO Project. [Online]. Available: https://www.echonetwork.eu

[25] CyberSec4Europe Project, “Cyber ranges federation project reaches new mile-stone.” [Online]. Available: https://cybersec4europe.eu/

[26] “CybExer.” [Online]. Available: https://cybexer.com/

[27] P. Qiu, “Cyber range service: A platform to experience the intelligent cyber security for the real world.” [Online]. Available: https://www.cisco.com/c/dam/global/en_hk/assets/event/cisco_connect_2015/pdf/4-3.pdf

[28] NIST, “The NIST Definition of Cloud Computing.” [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[29] “Recession Is Good For Cloud Computing – Microsoft Agrees,” 2012. [Online]. Available: http://www.cloudave.com/link/recession-is-good-for-cloud-computing-microsoft-agrees

[30] D. Farber, “CNET News: The new geek chic: Data centers,” 2008. [Online]. Available: http://news.cnet.com/8301-13953_3-9977049-80.html

[31] "Defining 'Cloud Services' and 'Cloud Computing'," 2008. [Online]. Available: https://web.archive.org/web/20100722074526/http://blogs.idc.com/ie/?p=190

[32] Business Week, "Jeff Bezos' Risky Bet." [Online]. Available: http://www.businessweek.com/magazine/content/06_46/b4009001.htm

[33] S. He, L. Guo, M. Ghanem, and Y. Guo, "Improving resource utilisation in the cloud environment using multivariate probabilistic models," in *2012 IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 574–581.

[34] Q. He, J. Han, Y. Yang, H. Jin, J. Schneider, and S. Versteeg, "Formulating cost-effective monitoring strategies for service-based systems," *IEEE Transactions on Software Engineering*, vol. 40, no. 5, pp. 461–482, 2014.

[35] Heather Smith, *Xero For Dummies*. John Wiley & Sons, 2013. [Online]. Available: https://books.google.com/books?id=drOF19aBKfgC&pg=PT37

[36] R. King, "Cloud Computing: Small Companies Take Flight," 2008. [Online]. Available: http://www.businessweek.com/technology/content/aug2008/tc2008083_619516.htm

[37] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud," in *2012 IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 423–430.

[38] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa, "Workload-Based Software Rejuvenation in Cloud Systems," *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1072–1085, 2013.

[39] Michael Kuperberg, Nikolas Herbst, et al., "Defining and Measuring Cloud Elasticity," 2011. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000023476

[40] "Economies of Cloud Scale Infrastructure," 2011. [Online]. Available: https://www.youtube.com/watch?v=nfDsY3f4nVI

[41] S. He, L. Guo, Y. Guo, C. Wu, M. Ghanem, and R. Han, "Elastic Application Container: A Lightweight Approach for Cloud Resource Provisioning," in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012, pp. 15–22.

[42] S. Marston, Z. Li, S. Bandyopadhyay, and A. Ghalsasi, "Cloud Computing - The Business Perspective," in *2011 44th Hawaii International Conference on System Sciences*, 2011, pp. 1–11.

[43] S. Nouri, H. Li, S. Venugopal, W. Guo, M. He, and W. Tian, "Autonomic decentralized elasticity based on a reinforcement learning controller for cloud applications," *Future Generation Computer Systems*, vol. 94, 12 2018.

[44] E. Mills, "Cloud computing security forecast: Clear skies," 2009. [Online]. Available: https://www.cnet.com/news/cloud-computing-security-forecast-clear-skies/

[45] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu, "Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends," in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 621–628.

[46] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce," 2011. [Online]. Available: https://doi.org/10.6028%2FNIST.SP.800-145

[47] A. Rashid and A. Chaturvedi, "Cloud Computing Characteristics and Services: A Brief Review," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, vol. 7, pp. 421–426, 02 2019.

[48] A. Joint and E. Baker, "Knowing the past to understand the present1 – issues in the contracting for cloud based services," *Computer Law & Security Review*, vol. 27, no. 4, pp. 407 – 415, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0267364911000689

[49] V. Gonçalves and P. Ballon, "Adding value to the network: Mobile operators' experiments with software-as-a-service and platform-as-a-service models," *Telematics and Informatics*, vol. 28, no. 1, pp. 12 – 21, 2011, mobile Service Architecture and Middleware. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585310000365

[50] M. Carney, "AnyPresence partners with Heroku to beef up its enterprise mBaaS offering," 2013. [Online]. Available: http://pandodaily.com/2013/06/24/anypresence-partners-with-heroku-to-beef-up-its-enterprise-mbaas-offering/

[51] "Wikipedia." [Online]. Available: https://en.wikipedia.org/

[52] R. Miller, "AWS Lambda Makes Serverless Applications A Reality," 2015. [Online]. Available: https://techcrunch.com/2015/11/24/aws-lamda-makes-serverless-applications-a-reality/

[53] "bliki: Serverless." [Online]. Available: https://www.martinfowler.com/bliki/Serverless.html

[54] P. Sbarski, "Serverless Architectures on AWS: With examples using AWS Lambda," 2017. [Online]. Available: https://www.amazon.com/Serverless-Architectures-AWS-examples-Lambda/dp/1617293822

[55] "Self-Run Private Cloud Computing Solution – GovConnection," 2014. [Online]. Available: http://www.govconnection.com/IPA/PM/Info/Cloud-Computing/Self-Run-Private-Cloud.htm

[56] T. Bittman, "Mind the Gap: Here Comes Hybrid Cloud – Thomas Bittman," 2015. [Online]. Available: http://blogs.gartner.com/thomas_bittman/2012/09/24/mind-the-gap-here-comes-hybrid-cloud/

[57] M. Rouse, "What is a multi-cloud strategy." [Online]. Available: http://searchcloudapplications.techtarget.com/definition/multi-cloud-strategy

[58] R. King, "Pivotal's head of products: We're moving to a multi-cloud world (ZDnet)." [Online]. Available: https://www.zdnet.com/pivotals-head-of-products-were-moving-to-a-multi-cloud-world-7000030737/

[59] "Multcloud manage multiple cloud accounts." [Online]. Available: http://www.groovypost.com/reviews/multcloud-manage-multiple-cloud-accounts/

[60] R. Gall, "Polycloud: a better alternative to cloud agnosticism," 2018. [Online]. Available: https://hub.packtpub.com/polycloud-a-better-alternative-to-cloud-agnosticism/

[61] L. Roh, "Is the Cloud Finally Ready for Big Data?" 2016. [Online]. Available: http://dataconomy.com/2016/08/cloud-ready-for-big-data/

[62] C. Yang, Q. Huang, Z. Li, K. Liu, and F. Hu, "Big Data and cloud computing: innovation opportunities and challenges," *International Journal of Digital Earth*, vol. 10, no. 1, pp. 13–53, 2017. [Online]. Available: https://doi.org/10.1080/17538947.2016.1239771

[63] M. A. S. Netto, R. N. Calheiros, E. R. Rodrigues, R. L. F. Cunha, and R. Buyya, "HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges," *ACM Comput. Surv.*, vol. 51, no. 1, Jan. 2018. [Online]. Available: https://doi.org/10.1145/3150224

[64] H. Wang, W. He, and F.-K. Wang, "Enterprise cloud service architectures," *Information Technology and Management. 13 (4): 445–454*, 2012. [Online]. Available: https://doi.org/10.1007%2Fs10799-012-0139-4

[65] "What is Cloud Computing?" *Amazon Web Services*. [Online]. Available: https://aws.amazon.com/what-is-cloud-computing/

[66] R. Baburajan, "The Rising Cloud Storage Market Opportunity Strengthens Vendors," 2011. [Online]. Available: http://it.tmcnet.com/channels/cloud-storage/articles/211183-rising-cloud-storage-market-opportunity-strengthens-vendors.htm

[67] K. Oestreich, "Converged Infrastructure," 2010. [Online]. Available: https://web.archive.org/web/20120113094920/http://www.thectoforum.com/content/converged-infrastructure-0

[68] T. Simpson and J. Novak, "Hands on Virtual Computing," 2017. [Online]. Available: http://www.thectoforum.com/content/converged-infrastructure-0

[69] "Where's The Rub: Cloud Computing's Hidden Costs," 2014. [Online]. Available: https://www.forbes.com/sites/centurylink/2014/02/27/wheres-the-rub-cloud-computings-hidden-costs/

[70] "Cloud Computing: Clash of the clouds," *The Economist*, 2009. [Online]. Available: http://www.economist.com/displaystory.cfm?story_id=14637206

[71] Gartner, "Gartner Says Cloud Computing Will Be As Influential As E-business." [Online]. Available: http://www.gartner.com/it/page.jsp?id=707508

[72] G. Gruman, "What cloud computing really means," 2008. [Online]. Available: http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031

[73] Steven J. Vaughan-Nichols, "Microsoft developer reveals Linux is now more used on Azure than Windows Server." [Online]. Available: https://www.zdnet.com/article/microsoft-developer-reveals-linux-is-now-more-used-on-azure-than-windows-server/

[74] G. Kumar, "A Review on Data Protection of Cloud Computing Security, Benefits, Risks and Suggestions," *United International Journal for Research & Technology*, 2019. [Online]. Available: https://uijrt.com/articles/v1i2/UIJRTV1I20004.pdf

[75] "Cloud Computing Privacy Concerns on Our Doorstep." [Online]. Available: http://cacm.acm.org/magazines/2011/1/103200-cloud-computing-privacy-concerns-on-our-doorstep/fulltext

[76] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Systems with Applications*, vol. 42, p. 7905–7916, 11 2015.

[77] "Identity and access management in cloud environment: Mechanisms and challenges." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098617316750

[78] "Cloud Security Alliance." [Online]. Available: https://cloudsecurityalliance.org/

[79] Cloud Security Alliance, "Top Threats to Cloud Computing: The Egregious 11," 2019. [Online]. Available: https://cloudsecurityalliance.org/artifacts/top-threats-egregious-11-deep-dive/

[80] "Google Drive, Dropbox, Box and iCloud Reach the Top 5 Cloud Storage Security Breaches List," 2015. [Online]. Available: https://web.archive.org/web/20151123032912/https://psg.hitachi-solutions.com/credeon/blog/google-drive-dropbox-box-and-icloud-reach-the-top-5-cloud-storage-security-breaches-lis

[81] M. Maltais, "Who owns your stuff in the cloud?" *Los Angeles Times*, 2012. [Online]. Available: http://articles.latimes.com/2012/apr/26/business/la-fi-tech-savvy-cloud-services-20120426

[82] "Security of virtualization, cloud computing divides IT and security pros," 2010. [Online]. Available: https://www.networkworld.com/article/2244954/virtualization/security-of-virtualization--cloud-computing-divides-it-and-security-pros.html

[83] "The Bumpy Road to Private Clouds." [Online]. Available: http://www.computerworld.com/article/2549867/data-center/the-bumpy-road-to-private-clouds.html

[84] "Disadvantages of Cloud Computing (Part 1) – Limited control and flexibility." [Online]. Available: https://cloudacademy.com/blog/disadvantages-of-cloud-computing/

[85] "The real limits of cloud computing." [Online]. Available: https://www.itworld.com/article/2726566/cloud-computing/the-real-limits-of-cloud-computing.html

[86] M. Karra, "Cloud solutions for translation, yes or no?" [Online]. Available: https://www.iapti.org/articles/art34-using-cloud-solutions-for-translation-yes-or-no.html

[87] L. Seltzer, "Your infrastructure's in the cloud and the Internet goes down. Now what?" [Online]. Available: https://www.zdnet.com/article/the-internet-is-down-what-next/

[88] D. Smith, "Hype Cycle for Cloud Computing, 2013." [Online]. Available: https://www.gartner.com/doc/2573318/hype-cycle-cloud-computing-

[89] "The evolution of Cloud Computing." [Online]. Available: https://web.archive.org/web/20170329121024/http://www.hello-cirro.co.uk/evolution-of-cloud-computing/

[90] "Microsoft Says to Spend 90% of R&D on Cloud Strategy." [Online]. Available: https://web.archive.org/web/20131018050315/http://cloudtimes.org/2011/04/12/microsoft-says-to-spend-90-of-rd-on-cloud-strategy/

[91] "Roundup of Cloud Computing Forecasts And Market Estimates, 2014," *Forbes*, 2014. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2014/03/14/roundup-of-cloud-computing-forecasts-and-market-estimates-2014/

[92] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics: An overview," 01 2011. [Online]. Available: https://www.researchgate.net/publication/229021339

[93] R, Adams, "The emergence of cloud storage and the need for a new digital forensic process model," 2013. [Online]. Available: http://researchrepository.murdoch.edu.au/id/eprint/19431/

[94] "Microsoft service level agreements for azure services." [Online]. Available: https://azure.microsoft.com/en-us/support/legal/sla/summary/

[95] "Microsoft azure availability zones." [Online]. Available: https://azure.microsoft.com/en-us/global-infrastructure/availability-zones

[96] "Microsoft azure backup and disaster recovery services." [Online]. Available: https://azure.microsoft.com/en-us/solutions/backup-and-disaster-recovery/#related-products

[97] "Microsoft azure traffic manage profiles." [Online]. Available: https://docs.microsoft.com/en-us/azure/traffic-manager/

[98] "Docker container framework." [Online]. Available: https://www.docker.com/

[99] IBM, “The Benefits of Containerization and What It Means for You.” [Online]. Available: https://www.ibm.com/cloud/blog/the-benefits-of-containerization-and-what-it-means-for-you

[100] “Oracle java.” [Online]. Available: https://www.java.com/en/

[101] “Openjfx.” [Online]. Available: https://openjfx.io/

[102] “Apache maven project.” [Online]. Available: https://maven.apache.org/

[103] “Microsoft powershell.” [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7

[104] “Microsoft powershell documentation.” [Online]. Available: https://docs.microsoft.com/en-us/powershell/azure/?view=azps-4.7.0

[105] “jpowershell library.” [Online]. Available: https://github.com/profesorfalken/jPowerShell

[106] “Java secure channel library.” [Online]. Available: http://www.jcraft.com/jsch/