2018-12-10

Designing Cybersecurity Competitions in the Cloud: A Framework and Feasibility Study

Chandler Ryan Newby
Brigham Young University - Provo

Designing Cybersecurity Competitions in the Cloud:

A Framework and Feasibility Study

Chandler Ryan Newby

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Dale C. Rowe, Chair
Barry M. Lunt
Derek L. Hansen

School of Technology

Brigham Young University

ABSTRACT

Designing Cybersecurity Competitions in the Cloud:
A Framework and Feasibility Study

Chandler Ryan Newby
School of Technology, BYU
Master of Science

Cybersecurity is an ever-expanding field. In order to stay current, training, development, and constant learning are necessary. One of these training methods has historically been competitions. Cybersecurity competitions provide a method for competitors to experience firsthand cybersecurity concepts and situations. These experiences can help build interest in, and improve skills in, cybersecurity.

While there are diverse types of cybersecurity competitions, most are run with on-premise hardware, often centralized at a specific location, and are usually limited in scope by available hardware. This research focuses on the possibility of running cybersecurity competitions, specifically CCDC style competitions, in a public cloud environment.

A framework for running cybersecurity competitions in general was developed and is presented in this research. The framework exists to assist those who are considering moving their competition to the cloud.

After the framework was completed, a CCDC style competition was developed and run entirely in a public cloud environment. This allowed for a test of the framework, as well as a comparison against traditional, on-premise hosting of a CCDC. The cloud-based CCDC created was significantly less expensive than running a comparable size competition in on-premise hardware. Performance problems—typically endemic in traditionally-hosted CCDCs—were virtually non-existent. Other benefits, as well as potential contraindications, are also discussed.

Another CCDC style competition, this one originally built for on-premise hardware, was then ported to the same public cloud provider. This porting process helped to further evaluate and enrich the framework. The porting process was successful, and data was added to the framework.

Keywords: cybersecurity, IT, cloud, virtualization, competition, CCDC

ACKNOWLEDGEMENTS

My family has been an integral part in completing my graduate degree and this thesis, both in support and in motivation: My wife Sydney has been a constant source of loving support and encouragement throughout my college education. She has cheered me on and buoyed me up at every step. Her help cannot be overstated. My 2-year-old daughter Julia many times saw me working and encouraged me forward by stating "Daddy work on thesis." My son Ryan was born during the final stretch of completing my thesis. And my parents, Van and Bethanie Newby who instilled in me a love of learning that has motivated and pushed me forward throughout my life. To all my family: thank you.

Thank you to my graduate chair, Dr. Dale Rowe for supporting and mentoring me throughout my studies, and for allowing me to compete in such a plethora of CCDCs throughout my time in school. Those competitions helped form the basis for my research.

Finally, thanks be to the Lord, whose guiding influence throughout my life has helped make me into who I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Background and Problem Statement

Cybersecurity is a rapidly expanding field. Unfortunately, the number of people qualified to work in the field and help defend our systems is not expanding quickly enough. There are many more job openings than there are individuals qualified to fill them (Culbertson, et al. 2017). With more and more security breaches happening all the time (Identity Theft Resource Center; CyberScout 2018), this dearth of trained professionals is more problematic than ever. One of the reasons for this shortage is the difficulty of training highly qualified individuals to do cybersecurity work. Job training and education can be difficult in any field, but it is especially onerous in information technology, and even more so in cybersecurity. Rowe, Ekstrom, and Lunt detailed a need for an "integrative and pervasive security theme" in general IT education, as well as "more advanced education in security topics" (Rowe, Ekstrom and Lunt, Cyber-Security, IAS and the Cyber Warrior 2012). Cybersecurity training requires not only traditional education, information gathering skills, and memorization, but also practical application of learned information in controlled, yet open environments. Unfortunately, it is this practical experience that is often the most difficult to gain.

Cybersecurity competitions are a good place for students to practice those necessary hands-on skills. They provide a fun atmosphere with new and exciting challenges that push students to learn and try new things. There are many different formats of cybersecurity

competitions that fall into different genres. There are competitions for defense (CCDC, or the Collegiate Cyber Defense Competition), offense (CPTC, or the Collegiate Penetration Testing Competition), mixed offence and defense (CTF, or Capture the Flag), and many others. One common theme throughout these competitions is the inclusion of hands-on, applied action by the competitors.

Hands-on, practical cybersecurity experiences are something that have historically been hard to provide, both for learning and competition environments. Students and competitors need to be able to try things without fear of causing significant downtime on a production network, or even of doing irreparable harm to systems (Rowe, Cunha and Cornel, A Highly Scalable and Reduced-Risk Approach to Learning Network Man-in-the-Middle (MITM) and Client-Side Exploitation (CSE) 2017). Allowing students of cybersecurity to practice or compete on a production network is dangerous at best. In cases where the only network available is a functioning production network, oftentimes individuals are forced to simply observe what is happening as opposed to interacting with the networks and systems. This type of interaction is insufficient. Another option is to use dedicated infrastructure. This solves the problem of negatively affecting production networks, however it is often quite expensive. This high cost makes many competitions and learning experiences that would otherwise be very beneficial cost prohibitive.

With the advent of virtualization technology, both of the above problems can be solved. Virtual machines and virtual networks allow all sorts of computer and network devices to be simulated without the need for additional hardware. With these advances in virtualization technology, hands-on cybersecurity training and competitions can be made available at low cost and with negligible impact on production networks.

Extending the benefit of virtualization to another level is cloud computing. Cloud computing couples the density and scalability of virtualization with the cost benefits of outsourcing (McKendrick 2014). With cloud computing, large simulated networks can be virtualized, experimented on, and torn down in an extremely small time period with no large upfront expenses and minimal operational costs. There are many different companies that provide cloud services at minimal cost. Some of the most notable examples are AWS (Amazon Web Services), Microsoft Azure, and Google Cloud Platform. Although these three cloud providers comprise more than 80% of the market share (Coles 2016), there are many other potential cloud vendors available to choose from.

Security training and competitions are different than many other common workloads and have different requirements. A technical framework for evaluating the requirements of cybersecurity trainings and competitions, specifically when run in the cloud, is needed.

The purpose of this research is to develop a framework for the requirements needed to successfully run hands-on cybersecurity training and/or competitions in the cloud. Once the framework is developed, it will be tested and expanded on by implementing a specific cybersecurity competition (CCDC) in a public cloud provider.

1.2 Research Questions and Objectives

There are two research questions that will be answered and two research objectives that will be met in this thesis:

Research Question 1 – Running a CCDC in on-premise hardware has been successful in the past. What requirements are needed to run such a competition in the cloud?

Research Question 2 – How do the costs and benefits of running a CCDC in a public cloud environment compare generally to running a similar style competition in on-premise hardware?

Research Objective 1 – Develop a technical framework and list of requirements for the specific workload of cybersecurity competitions in a public cloud environment.

Research Objective 2a – Create a CCDC entirely in a public cloud provider using the developed framework as a reference. Run the competition with multiple live teams and a live red team, simulating an official CCDC.

Research Objective 2b – Port an existing on-premise cybersecurity competition (preferably a CCDC) into a public cloud environment. Use the information gathered to enhance the framework.

1.3 Delimitations

This thesis will only examine public cloud providers in the context of cybersecurity hands-on training and competition workloads. No other cloud workloads will be considered. Additionally, only a single cloud provider (AWS) will be used for the prototype for this thesis. AWS was chosen due to familiarity with the platform. Exploring the feasibility of additional cloud providers could be done in future work. See (NTT Communications n.d.), (Garg, Versteeg and Buyya 2011), and (Li, et al. 2010) for research on choosing a provider. The prototype and port will only be used to test the efficacy of a cloud-based cybersecurity competition.

1.4 Summary of Methodology

The initial data for the framework and requirements list will be created by utilizing personal experience and knowledge of CCDCs, and then requesting and incorporating feedback from those familiar with CCDC, cybersecurity, and IT in general.

Once the framework and list of technical requirements is complete, a CCDC will be built in a public cloud environment. The steps required will include planning the competition details, creating the network backbone, creating team servers and scored services, automating the deployment of team environments, and final integration testing. Once the competition is complete, it will be run by individuals familiar with CCDCs.

After the first cloud-based CCDC is created and evaluated, another on-premise CCDC will be ported to a public cloud provider. This porting will be done in conjunction with the original authors of said on-premise competition. The framework and experience from the first CCDC will be utilized to complete the porting. Once the port is complete, the process will be evaluated and used to enhance the framework.

The final step in the methodology is the cost and benefit analysis. This will be performed by estimating the cost of running a CCDC similar in size to the cloud-based competition in on-premise hardware. This information will be compared with the costs incurred in running the competition in the cloud. Additional benefits from and problems with the cloud-based CCDC will be gathered in this step.

1.5 Thesis Layout

Chapter 2 – Literature Review: An overview of current literature on cybersecurity education and competitions; virtualization; cloud technologies; and the costs and difficulties of running cybersecurity competitions.

Chapter 3 – Methodology: The methods and procedures that will be used to meet the research objectives and answer the research questions.

Chapter 4 – Results of Framework Creation and Final Framework: The final resulting technical framework generated from following the methodology outlined in chapter 3. Will also include preliminary discussion on the results.

Chapter 5 – Results of CCDC Creation, Testing, and Porting: The results of implementing 1) a cybersecurity competition (CCDC) entirely on cloud-hosted infrastructure, 2) running said competition, and 3) porting an on-premise CCDC to a cloud provider.

Chapter 6 – Conclusions and Future Work: Analysis of both research questions and both research objectives, examination of possible future work, recommendations, and conclusions.

2 LITERATURE REVIEW

According to a report by Cybersecurity Ventures there will be a shortfall of cybersecurity professionals of around 1.5 million by 2019 (Morgan 2016). While some of this shortage is due to the exponential increase in need, a significant portion of the shortage is due to lack of good cybersecurity training at the university level. According to a report by CloudPassage, "not one of the top 10 U.S. computer science programs … requires a single cybersecurity course for graduation" (CloudPassage 2016).

Even if there was a great increase in the number of individuals interested in cybersecurity, an additional problem in the industry is finding ways to give those people relevant experience. Traditional teaching methods can be valuable, but hands-on experiences are vital: "Students studying topics in cybersecurity benefit from working with realistic training labs that test their knowledge of network security" (Stewart, Humphries and Andel 2009). Comer suggests that not only are lab environments useful for learning, they are "absolutely essential … because students learn by doing" (Comer 2002). Unfortunately, security training has traditionally cost a large amount of time and money. One of the major security training providers, SANS, typically charges more than $5000 for a single class (SANS Institute 2018). Obtaining the infrastructure for hands-on training can also be quite expensive. According to Kneale and Box, "provid[ing] 20 work areas so that each student has equal access to the equipment during class would cost about AU$160,000" (Kneale and Box 2003).

2.1 Cybersecurity Competitions

One common way to gain hands-on experience is via cybersecurity competitions. A recent report by Katzcy Consulting said: "The current and projected workforce needs must be met not only by training more cybersecurity personnel, but also by raising the bar on their skills, aptitude and ability to collaborate. Cybersecurity competitions can play a critical role in this mandate" (Katzcy Consulting 2016). In addition to being vital to learning, students often *enjoy* learning through competition. In Namin et al., the authors conclude that as long as students have a solid base understanding of the topic at hand, "… participants liked the competition-based learning incorporated into the workshop. The competition atmosphere stimulated their motivations to solve more challenges" (Namin, Aguirre-Muñoz and Jones 2016).

Katzcy further remarks: "Cyber competitions have been around for over two decades" (Katzcy Consulting 2016). Today there are diverse types of cybersecurity competitions, across many disciplines and skill levels. Some of these include CCDC (Collegiate Cyber Defense Challenge), US Cyber Challenge, CTF (Capture the Flag), CyberPatriot, CPTC (Collegiate Penetration Testing Competition), and many more. Each of these competition types provides different benefits and comes with different complications. While the focus of this research is limited to CCDC style competitions, it is useful to understand a few of these different competitions and how they are implemented.

2.1.1 Capture the Flag (CTF)

Capture the Flag competitions are often structured as a jeopardy board of questions, although they can also be attack-defense based. They have various categories of problems, such as cryptography, packet analysis, steganography, and binary exploitation. Each category has a

handful of challenges worth different amounts of points. Each solved challenge grants the solver points. The person with the most points at the end of the competition wins. Capture the flag competitions can be somewhat useful educational experiences. However, they often have a high knowledge barrier to entry that prevents some from being able to fully participate (Mansurov 2016). Some well-known capture the flag events include the DefCon CTF (vulc@n of DDTek n.d.), PlainCTF, hosted by the Plaid Parliament of Pwning (Plaid Parliament of Pwning n.d.), and PicoCTF, also hosted by the Plaid Parliament of Pwning.

2.1.2 US Cyber Challenge

The US Cyber Challenge isn't specifically a single competition. It's an initiative by the US government to help "connect America's best and brightest to the cybersecurity industry" (Evans n.d.). The Cyber Challenge organization holds multiple events that help qualify people to attend Cyber Camps. One of these events is called a Cyber Quest. The US Cyber Challenge competitions are specifically targeted towards high-school, undergraduate, and post-graduate students.

2.1.3 Collegiate Cyber Defense Challenge (CCDC)

A CCDC event is a national competition aimed at college students. It is a defensive competition where students are assigned to manage an insecure network representing a small business or government office and asked to understand and secure the network while maintaining operational functions. CCDC was created at the University of Texas at San Antonio in 2006 when members of The Center for Infrastructure Assurance and Security (CIAS) held a workshop to "discuss the possibility of establishing a national collegiate cybersecurity competition" (White and Williams 2005). In a CCDC, students put into practice comprehensive and diverse security

and system administration skills. They are asked to defend an insecure network while a red team of professional penetration testers actively attempt to break into their network. The CCDC requires a robust skillset and is representative of a real-world environment with artificially accelerated cybersecurity threats and management requirements. For this reason, it is the focus of this research.

2.1.4 CyberPatriot

CyberPatriot is a competition very much like CCDC. Its main difference is that it is geared towards high school students. In CyberPatriot competitions, students download an operating system image that has been purposely made vulnerable. On their own time, they open the image and attempt to find and fix as many vulnerabilities as possible while keeping the service available. A CyberPatriot scoring engine keeps track of how often the service goes down (CyberPatriot n.d.). The CyberPatriot is set up so that teams can participate whenever they have time. They just need to have a computer capable of running the provided image. Although this competition is technically similar to CCDC, it doesn't fit this research as well. Because the images are run on the teams' computers, there isn't a good opportunity to study running them in the cloud.

2.1.5 Collegiate Penetration Testing Competition

The Collegiate Penetration Testing Competition (CPTC) is a new type of competition started in 2015 at the Rochester Institute of Technology. It is another collegiate competition with roots similar to CCDC. The difference is that CPTC is an offensive, penetration testing competition where competitors "use their technical knowledge and skills to identify weaknesses in a simulated corporate environment without impacting the operations of simulated business

activities." The goal of this type of competition "is to model a real-life penetration test as closely as possible in a competition environment" (RIT n.d.).

Table 2-1 - Competition Type Comparison

Competition Type	Structure	Skills Targeted	Age Group Targeted
CTF	Varied. Can be jeopardy style or attack/defense	Varied. Typically includes binary exploitation, reverse engineering, network analysis, and others	Different competitions for all age groups
US Cyber Challenge	Events are often used as qualifies to attend a Cyber Camp	Varied	High school, undergraduate, and post-graduate students
CCDC	Teams of 4 to 8 defend a mock network against live attackers	System Administration, Network Defense, Incident Response	College
CyberPatriot	Teams work to identify and fix vulnerabilities in a downloaded VM image	Vulnerability Identification, System Administration	High school
CPTC	Teams work to attack and infiltrate a mock company, then create a professional report	Penetration Testing, Report Writing	College

As stated previously, this research will focus primarily on CCDCs, although the findings could be applied to other types of competitions once their hardware and infrastructure requirements are understood.

2.2 Costs and Difficulties of Running Competitions

Regardless of the competition type, there will be some set of requirements for hardware and/or software. Each type of competition, and even different iterations of a single competition, often have costs associated with them. “These costs generally involve (1) hardware costs for hosting the competition, (2) the human resource expense required to administer the competition, and (3) the availability of and/or investment associated with competition material for the particular event” (Taylor, et al. 2017). These costs can be significant. The Carnegie Mellon University sponsored PicoCTF, designed and built by “CMU's four-time DefCon ‘World Series of Hacking’ champion hacking team, the Plaid Parliament of Pwning (PPP) … costs tens of thousands of dollars, to pay for the game development, problem development and hosting services” (Carnegie Mellon University Crowdfunding 2018). Dan Manson, Professor and Department Chair in Computer Information Systems at Cal Poly Pomona./Co-Chair NICE Working Group Competitions SubGroup said, “Until you’ve done one, you don’t know how critical and how difficult it is getting the competition environment working. You need a willing partner, usually a college or university, to host and deliver the resource-intensive tasks of providing working computers, a functional wireless network, enough bandwidth from the ISP, and so on” (Katzcy Consulting 2016).

2.3 CCDC Costs

The majority of this research will be concerned with hardware costs specific to building and running CCDCs. These costs are explored in the next few sections.

2.3.1 Hardware/Operational Costs

In order to run a CCDC, a significant amount of hardware resources is required. Speaking about the Southeast Collegiate Cyber Defense Competition (a regional qualifying event for the national CCDC), Whitman and Mattord said: "The most challenging part of hosting the SECCDC is collecting sufficient resources" (Whitman and Mattord 2008). A CCDC should emulate a small business or government network with its multiple servers, networks, and services. Typical CCDC's have 2-3 systems per competitor, including network devices. For a team of eight people (the standard CCDC team size), that is 16-24 servers, multiple network devices, and other miscellaneous equipment. Multiplying that by the number of teams in a competition leads to potentially hundreds of devices to manage. If each of these devices were purchased individually, the costs of running a full scale CCDC would be astronomical. To alleviate some of these costs, virtualization technology is typically used, often with on-premise hardware that may be dedicated to the purpose, or temporarily assigned.

Beyond the servers used in a CCDC, each competitor needs access to a workstation to compete on. This is done differently in different competitions. The two most common methods for providing workstations for competitors are:

1. Provide each competitor with a dedicated laptop to use. This can be good because it ensures that each person on each team has the same resources; no one has an unfair advantage. However, it adds additional cost and maintenance requirements on the competition organizers.

2. Have competitors bring their own laptops and connect to virtualized workstations. This method can be much less expensive; virtualized workstations are cheaper than purchasing laptops. However, it can limit the competitors because they are forced to do all their work

on a remote system. It also leaves room for cheating or unfair advantages such as when some competitors have more powerful hardware available to them or potentially "pre-stage" beneficial programs and other materials on their personal laptops. Pre-staging is already difficult to detect and enforce, having competitors bring their own machines makes it even harder.

Another cost associated with operating a CCDC (as well as with other cybersecurity competitions) is the network fabric. All of the servers and competition equipment needs to be connected to the network, requiring switches, routers, cables, etc. The competitors also need to each be connected to the network, although "some of this cost ... may be mitigated by employing wireless networking, which is not limited by network cables or the number of ports on network switches" (Taylor, et al. 2017).

Power consumption costs should also be considered. While there are many methods for estimating power consumption of servers, it can be difficult to gather exact data without recording it during an actual CCDC. Educated estimates can be made based on the likely amount of hardware needed for an event, along with estimated power consumption for typical servers and average price for electricity. Data points used for the estimation in this thesis are gathered from "Calculating Space and Power Density Requirements for Data Centers" (Rasmussen 2013) and "Electric Power Monthly with Data for April 2018" (U.S. Department of Energy 2018).

Fortunately, there are multiple technologies and systems that can be utilized to reduce these hardware/operational costs potentially drastically, some of which are: virtualization technology and public cloud computing.

2.3.2 Human Costs

Depending on the format, it can often take hundreds of hours to plan, design, and build a large cybersecurity competition. As an example: The DEFCON CTF is the largest CTF competition in the world (Korber 2013). Every 3-5 years, a new team will take over creating and hosting it. From 2013 to 2017, it was hosted by a group known as Legitimate Business Syndicate (LegitBS 2018). The Legitimate Business Syndicate published a writeup about the process of being selected as hosts, as well as the process of building and running the top CTF competition in the world. In their description, they mention needing a full team of experts, all with highly specialized knowledge (Genovese 2017). These experts all put in many hours, days, weeks, months, and sometimes even years to create a world class competition. For example, see Lightning's writeup of cLEMENCy (Businessman 2017). Lots of this time was dedicated to the infrastructure planning and development required to host the competition.

2.4 Virtualization

Virtualization is the ability to emulate all the typical pieces of hardware a computer normally uses in software and run other operating systems and programs in that emulated environment (VMWare 2018). These pieces of virtual hardware may include, among other things, CPUs, storage devices, network devices, and virtual memory. The virtual hardware is controlled by software called a hypervisor, whose job it is to allocate, manage, and safely segment these resources. This hypervisor software typically runs directly on bare-metal systems (common bare-metal hypervisors include VMware ESXi, Microsoft Hyper-V, and Citrix XenServer), but there are also hypervisor applications that run on host operating systems (examples include VMware Workstation, Oracle VM VirtualBox, and Parallels for Mac). These are more suited to testing, development, and single-use virtualization, whereas the bare-metal

hypervisors are often used in large scale environments where a large number of virtualized systems are needed.

Advances in virtualization technology have made simulated environments increasingly realistic and performant, as well as more isolated and secure. When properly tuned, and in very specific environments, virtualized environments can sometimes even offer slightly *better* performance than native hardware (Simons, DeMattia and Chaubal 2016). In addition, virtualized environments are typically assumed to be completely isolated from the host system they run on, which greatly increases the security of the applications and OSes that run on the system. This idea has been around since the conception of virtualization and is still applicable today (Garfinkel and Warfield 2007, Madnick and Donovan 1973).

Virtualization can also allow many different platforms to be experimented on without the cost of purchasing each one (Spanbauer 2006). With these advances in virtualization technology, many companies are turning to virtualization to increase compute density and bring down costs (CDW n.d.).

While virtualization offers some absolute benefits, not all problems can be solved with virtualization. Most (see (Simons, DeMattia and Chaubal 2016) for a counterexample) virtualized environments incur a non-trivial amount of overhead when compared to running the same applications in bare-metal (Chen, et al. 2015). This limitation should be considered, along with the benefits gained by virtualizing, when deciding whether or not to use virtualized environments. For example, in a cybersecurity competition that involves attacking physical hardware (see (Halderman, et al. 2008) for an example), virtualized systems could have a negative impact on performance. However, a competition that required using a disparate

collection of operating systems and configurations (such as a CCDC) would most likely benefit greatly from virtualization.

2.5 Cloud

One area where virtualization technology is being heavily used is in the cloud. Companies such as AWS (Amazon Web Services), Microsoft Azure, and GCP (Google Cloud Platform) together operate millions of servers in hundreds of data centers around the world. They can provide near-real-time access to virtually limitless amounts of computational power, storage, and bandwidth (Amazon Web Services 2016). When a user requests resources, the cloud provider is able to provision and provide said resources, incrementally billing the user until they no longer need the resources and deprovision them. In public cloud environments, the capital expenditures, as well as much of the operational expenditures, of running servers is removed and replaced with ongoing subscription-based billing. The management of things such as physical locations, server hardware, network equipment, cooling, and power are all offloaded to the cloud provider.

Cloud computing has taken off in the past few years. AWS (Amazon Web Services) posted $3.53 billion in revenue in just the fourth quarter of 2016 (Novet 2017). Many different studies, surveys, and projections have indicated that cloud computing adoption rates will continue to increase, and that cloud computing will be a major part of most IT organizations. Gartner projected that the public cloud market will hit $186.4 billion in 2018 (Moore and van der Meulen 2018).

Table 2-2 - Worldwide Public Cloud Service Revenue Forecast

	2017	**2018**	**2019**	**2020**	**2021**
Cloud Business Process Services (BPaaS)	42.6	46.4	50.1	54.1	58.4
Cloud Application Infrastructure Services (PaaS)	11.9	15	18.6	22.7	27.3
Cloud Application Services (SaaS)	60.2	73.6	87.2	101.9	117.1
Cloud Management and Security Services	8.7	10.5	12.3	14.1	16.1
Cloud System Infrastructure Services (IaaS)	30	40.8	52.9	67.4	83.5
Total Market	**153.5**	**186.4**	**221.1**	**260.2**	**302.5**

In billions of USD. From (Forni and van der Meulen 2016)

In a recent report, McAfee found that "97% of organizations use cloud services (public, private, or a combination of both), up from 93% one year ago" (McAfee 2018). According to Hofmann, cloud computing allows businesses to scale up quickly and remove many operational and capital expenditures (Hofmann and Woods 2010). In many cases, cloud computing is a big win for businesses.

3 METHODOLOGY

The methodology for the thesis is laid out. The process for developing the framework and list of requirements is presented first. Next, the methods for creating and running the cloud-based CCDC are put forth. Then the process for analyzing the required costs is explained. Finally, the process of porting an existing, on-premise CCDC to a public cloud is described.

3.1 RO 1 and RQ 1: Develop Framework and List of Requirements

The goal of this research objective was to develop a comprehensive framework and list of requirements for running cybersecurity competitions in a cloud environment. The process involved creating an initial framework, then identifying and working with subject matter experts (SMEs) in CCDCs, cybersecurity, public cloud, and IT in general to refine the framework. The SMEs were identified and selected based on previous experience with them and their work. They each responded positively when asked if they'd be willing to provide feedback. Feedback from these SMEs was gathered and incorporated into the framework. The final framework will be presented in chapter 4.

3.1.1 Create Initial Framework

The initial framework was created based on personal knowledge and experience. After gathering additional information during the literature review, a framework was created as seemed appropriate. At a minimum, it included:

1. A discussion on the desired outcome(s) of the competition.
2. Technical considerations
3. Operational considerations

While creating the framework, work was being done within a cloud provider to implement the competition discussed in research objective 2. This experience, along with personal experience in cybersecurity competitions, helped shape the initial framework creation.

Once the data gathering was complete, requirements were derived by aggregating a list of technologies used and requirements needed for a successful CCDC. This list was abstracted to a list of general technical requirements. For example:

Table 3-1 - Example of Technology to Technical Requirement Mapping

Technology	**Technical requirement**
Palo Alto Firewall	Firewall at the network perimeter
BIND DNS Server	Ability to authoritatively host a DNS server for a domain

After the technical requirements were established, they, along with other requirements gathered from the previous steps, were compiled into a framework of suggested features, requirements, and functionality. These formed suggestions for the things that are needed in a cloud provider in order to successfully run a cybersecurity competition. Once the initial framework was complete, the process of gathering feedback commenced.

3.1.2 Request Feedback

The framework and suggested requirements were sent to the people mentioned in the previous step. Their feedback was analyzed and incorporated into the final requirements/framework document.

3.2 Research Objective 2a: A Cloud-Based CCDC

This research objective was to build and run a full CCDC in a public cloud provider and document the results. This involved two major steps: creating the competition and running the competition. The framework developed in research objective 1 was both used and expanded upon while the competition was being developed.

3.2.1 Building the CCDC

The process of building the CCDC consisted of planning, designing the network layout, creating the server templates, automating the deployment process, and testing the entire competition setup.

3.2.1.1 Plan the Competition

Before a single host was deployed or line of code written, a plan was developed for the entire competition. This included the list of scored services and hosts to run them on, the network layout including how competitors and red team (The "red team" is the group of volunteers who act as the attackers. They are actively trying to break into the competitors' systems.) members will connect to the competition environment, and the competition schedule and timeline. One of the components of this step was a diagram of the proposed network design. This was essential in

determining where hosts were to be placed, where traffic was to enter and exit the network(s), and where services were scored from.

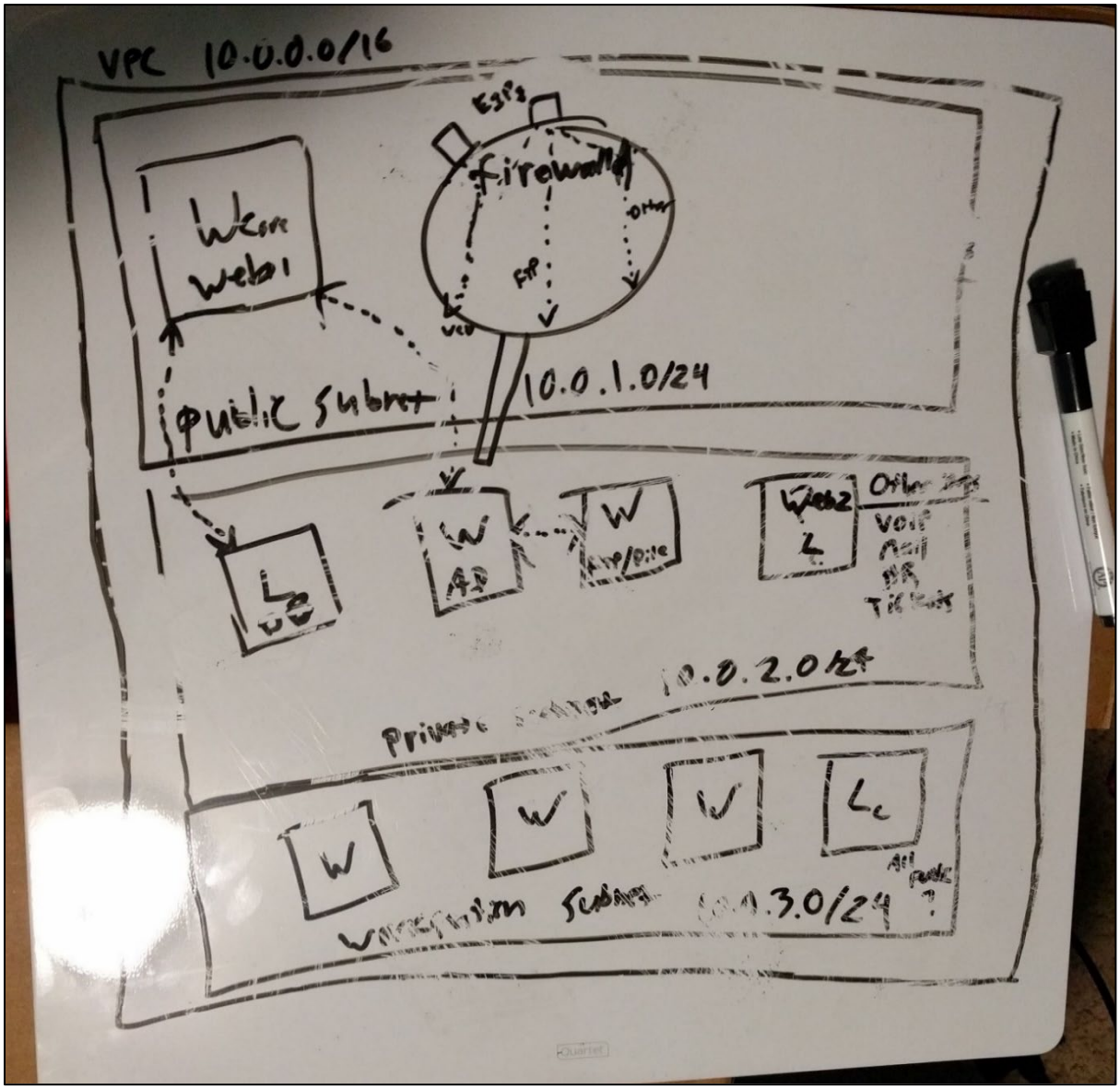

Figure 3-1 - Proposed Network Diagram Mockup

This competition was to be a practice scenario for the students in the Fall 2017 CCDC Prep class at BYU. In order to meet the objectives for the class, the competition had to be similar

to what the students would experience in their upcoming competitions. These competitions typically have a relatively basic network with only a few subnets, about 75% modern/common operating systems and 25% older/legacy/uncommon operating systems, and around 2-3 systems per competitor (including network devices). The competition plan was created with these requirements in mind. Once it was finished, the plans were sanity checked with individuals who have competed in CCDCs in the past, as well as with Dr. Dale Rowe, the BYU CCDC coach at the time and Professor of the Fall 2017 CCDC Prep class. Based on the collective CCDC experiences of these individuals, the competition described in the plan appeared similar in nature to official competitions hosted by the CIAS.

3.2.1.2 Create the Network Layout

Once the plans were in place, the next step was to create the network layout. The competition needed to be able to support a variable number of teams (4-10), so the design had to be modular. Each team setup had its own independent network layout. This differed from the original idea of creating a single network backbone for all teams to connect in to. Because of the way the public cloud provider's environments are set up, it was simpler to house each team inside of a dedicated network "container," as opposed to having a single network backbone connecting teams. An analogous idea in traditional on-premise hardware would be individual, identical, air-gapped networks all connected to the internet, as opposed to multiple subnets or VLANSs connected together and split into separate teams via a centralized router.

Each team's network layout included all infrastructure necessary to allow communication between the teams and the internet, between the red team and the teams, and between the competitors and their environments.

3.2.1.3 Create the Servers

Each server was then created and carefully documented. The servers were configured with whatever operating systems and software configurations were needed to fulfil the competition plan. While the server was being created, two types of documentation were created, both of which are available in the appendix.

First, a 'final-state' document was created. This describes the desired state of the server when it is fully deployed. This means that if things change multiple times throughout the configuration, only the final state will be recorded. This documentation exists as code definitions of the servers. It was originally planned to be salt state files but was later changed to be a collection of PowerShell, Bash, and shell scripts. Chapter 5 explores the reasons for this change.

The other type of documentation was an implementation guide for any unusual, difficult, or non-standard configurations. Specifically, anything that could not be defined in code. This allows for the process to be repeated in the future, and by different personnel.

3.2.1.4 Automation

Once all the servers were in their final state, the final-state scripts were used to fully automate the team environment creation. This automation created all the required network devices and settings, security policies, and servers. The networking and security devices were simply standard devices provided by the cloud provider. The servers themselves were based on base images from the cloud provider and automatically configured until they perfectly matched the desired final state. It was possible to deploy an entire team environment to a completely ready state with a single template.

The main automation tool used was AWS CloudFormation along with various Python, PowerShell, and Bash scripts. CloudFormation was chosen because it is native to AWS, where the competition was being developed. To ease development of CloudFormation templates, the Python library Troposphere (troposphere 2018) was used. Troposphere is a Python library that generates a CloudFormation template given a series of commands. This process will be examined in more detail in chapter 5.

3.2.1.5 Testing the Deployment

During the creation of the various pieces of the competition, servers and automation were tested regularly. During each work session, the previous session's hosts were deployed into the cloud environment. Any changes were recorded and integrated back into the deployment code. At the end of a work session, the entire infrastructure was torn down to save operational costs. Once the entire setup was complete, an additional set of tests was run, with additional validation to ensure that all components worked correctly during the competition. Even with this extensive testing, there were still some failures during the competition, a fact that will be explored during later analysis chapters.

3.2.2 Running the Competition

The CCDC created above was then run for students in the BYU Information Technology major in the Fall 2017 semester. Dr. Dale Rowe helped organize the competition as a part of his CCDC Prep class. Students were asked to participate as a practice for future competitions they would be competing in. Once a final tally of students was taken, the appropriate number of team environments was created from the automation templates. These team environments then

underwent a basic level of manual connectivity testing to ensure everything was working as expected. Once testing was complete, the competition began.

Although the original estimate was to have a total of three teams, at the last minute, a fourth team was added. Because of the scalable, on-demand nature of the cloud, as well as the fully automated deployment methodology developed in the competition creation process, this additional team was created with minimal extra overhead.

Throughout the competition, technical support was provided for the infrastructure, assisting in cases where the competition environment itself stopped working. Because of the nature of the competition, most teams encountered some form of problem. As mentioned in chapter 2, a CCDC involves unsecured, under-secured, and often misconfigured systems in a non-ideal environment that is under active attack by dedicated, persistent attackers. The purpose was not to help solve problems the students needed to solve themselves, but to make sure the competition environment remained operational and supportive of the experience. This included ensuring that the environment did not interfere with intentional problems created as part of the scenario.

After the competition was over, the students were asked to, on an optional volunteer basis, send feedback on their competition experience. This feedback was informal. It was simply a way to gauge, in a "thumbs up/thumbs down" way, if the competition was successful and if/how it mirrored other competitions these students had participated in.

3.3 Research Question 2: Analysis of Advantages and Disadvantages

Once the CCDC was over, data had been collected on the monetary and time costs of both creating and running a CCDC in a public cloud environment. This data was compared with

estimates for creating and running a traditional CCDC in on-premise hardware. For the purposes of this research question, simple educated estimates for a CCDC run in on-premise hardware were enough. A more exact comparison can be pursued in future work.

3.3.1 Cloud-based CCDC Cost Analysis Process

The costs of creating and then running the CCDC were recorded throughout the entire process using AWS's billing management tools. The free usage tier (AWS supplies every new account with one year of free trial usage) was utilized first, followed by an AWS education credit. All charges above and beyond that were paid at standard rates. Although the cost was low, AWS keeps meticulous records, and the cost details of the cloud CCDC were readily available.

This information was then compared to estimates generated for a similar competition hosted in on-premise hardware. Exact values are difficult to gauge exactly but the estimates gathered should be enough.

3.3.2 Other Analysis

The other, non-monetary results of the cloud hosted CCDC were compared to traditional CCDC experiences to determine other benefits gained and/or potential problems encountered in the cloud model. This was done by gathering anecdotal evidence from the competition participants, many of whom had participated in traditional CCDCs in the past. For this other analysis, there wasn't specific metrics in mind; the objective was simply to find and state differences between the two competitions to use as potential jumping off points for future research. For further exploration on these potential jumping off points, see chapter 6.

3.4 Research Objective 2b: Porting an On-Premise CCDC to the Cloud

In addition to a new CCDC created entirely in the cloud, another CCDC was "ported" to the cloud. A CCDC originally built for on-premise hardware was selected to be migrated into a public cloud provider. The framework was used as reference and to list out anything to watch out for when using the cloud provider. The process of porting also helped to improve on the framework. AWS was again utilized for the porting process.

3.4.1 Selecting an On-Premise Competition to Port

While the specific competition chosen wasn't as important as the fact that it was created for on-premise hardware, care was still given to select a viable competition. In order to make the process go as smooth as possible, and to focus the evaluation on cloud aspects of the process, the specifics of operating system configuration and competition scenario were a low priority. In addition, having the original competition's creator available to help with the porting process was an important aspect in choosing a competition.

3.4.2 Porting the Network and Machines

The chosen competition was similar in size to the competition created for research objective 2a. The network consisted of a simple flat subnet with a few machines connected to it. The networking equipment wasn't controlled by the competitors in the on-premise competition. All of this made porting the CCDC network straightforward. The machines were simply ported into a default subnet and connected to the internet via an internet gateway.

Porting the machines into the competition environment required a fair amount of conversion and preparation. The machines started out as VMs loaded into VMWare Workstation

on a standard PC. The process for importing working machine images into AWS, along with code snippets, is:

1. Upload the disk image to an AWS S3 bucket. (S3, or Simple Storage Service is AWS's object storage system. A bucket is a logical grouping of objects.) Prepare the disk image for conversion.

   For example: `aws s3 cp disk_image.ova s3://vm-import-bucket/`

2. Issue a request to AWS to convert the disk image into an AMI (Amazon Machine Image).

   ```
   aws ec2 import-image --description <imageDescription> --
   license-type <licenseValue> --disk-containers
   file://<diskContainersFile.json>
   ```

3. Using the AMI as a template, launch an instance into the default subnet. This was done from the AWS console.

4. Connect to the new host using the remote access credentials set before uploading, via either SSH or RDP (AWS Documentation 2018).

Once the AMIs were created, launching them into the default subnet was as simple as launching any other standard cloud image. The images were already configured with all the (mis)configurations, software, and potential vulnerabilities needed for the CCDC. They didn't need an additional onboarding script to prep them.

4 RESULTS OF FRAMEWORK CREATION AND FINAL FRAMEWORK

The framework creation process is discussed, including initial brainstorming, feedback gathering, feedback integration, and the final framework layout. The final framework is presented.

4.1 Initial Framework Brainstorming

As described in chapter 3, the initial framework was created entirely based on personal knowledge and experience. First, different categories to be covered in the framework were determined. The initial draft consisted of a brief introduction section along with three main categories: 1) desired learning/competition outcomes, 2) Minimum technical requirements for the public cloud provider, and 3) Operational considerations.

After research objective 2a (create and run a CCDC entirely in the cloud) was completed, more content was added to the framework. It still consisted of the three major categories mentioned above, but more material, gathered while working extensively in a public cloud provider, was added. After research objective 2b (port an existing CCDC to a public cloud environment) was completed, an additional section on porting virtual images to the cloud was added. Once the initial framework draft was updated to include all the things learned during the research objective methodology phases, it looked like this (only the section headings of the initial draft are included):

Choosing an Environment:
A Framework for Running Cybersecurity Competitions
in a Public Cloud Environment

1. Intro
2. Desired learning/competition outcomes:
    a. Potential outcomes for the competition
3. Minimum technical requirements for the public cloud provider:
    a. Platform
    b. Compute
    c. Network
    d. Connectivity to the environment
    e. Storage
    f. Legal/Regulatory
4. Operational considerations:
    a. Choosing between a public or private competition
    b. Pros and cons of each
    c. Competition timeframe and estimation of operational expenses
    d. Management console scoping.
    e. PaaS vs IaaS
    f. Determine how the competition will be created, porting or natively creating images in the cloud provider

Figure 4-1 - Framework Initial Draft Sections

4.2 Request and Integrate External Feedback

Once the initial framework was complete, the process of gathering feedback began. The framework and suggested requirements were sent to 21 SMEs with varying backgrounds in IT, cybersecurity, public cloud infrastructure, and CCDCs. Of the 21 people the framework was sent to, 10 responded and provided feedback. Their backgrounds were as follows. Some of the individuals fit into more than one category and have been counted more than once.

Table 4-1 - Breakdown of Respondents' Backgrounds

Background	Number of responses	Percentage of responses
IT Professional	4	40%
Cybersecurity Professional	4	40%
Public Cloud Expert	1	10%
CCDC Competitor	6	60%

Their feedback was analyzed and incorporated into the final requirements/framework document. Most of the feedback included suggestions on framework content to add or change. In addition, there was some feedback on grammar and organization. A selection of the feedback is presented below.

- "I only had one change I would suggest, which is the inclusion of Terms with Cost…. For Cloud providers, cost is rarely a simple number, but rather includes specific terms or conditions."

- "Another restriction on porting VMs is kernel version, at least in AWS (can't import something that's super old or brand new)."
- "The only addition or thing I thought was missing was mention of patch levels or possible staged vulnerabilities (as used in CCDC). … For example, I believe there were multiple versions of Windows supported (8, 10, maybe professional vs. home), but not necessarily multiple patch levels. It may be worth mentioning as some competitions need vulnerable systems."

4.3 Final Framework

After all the feedback was gathered and integrated, the final framework was assembled. The knowledge gained during the creation process of the CCDCs used in research objectives 2a and 2b was integrated into the final framework along with the external feedback. The final version of the framework, entitled **Cybersecurity Competitions in the Cloud: A Framework for Running Cybersecurity Competitions in a Public Cloud Environment**, is presented in its entirety starting with section 4.3.1 and ending with section 4.3.6.

4.3.1 Intro

This document describes a framework detailing the technical requirements for building and running a cybersecurity competition in a public cloud environment. It outlines the desired learning/competition outcomes, the required infrastructure, and any operational concerns. It *does* list out general recommendations and specific technical requirements. It *does not* list specific implementations of those technical requirements. Examples are clearly labeled as such. The resulting framework can be used to evaluate multiple potential environments for hosting a

cybersecurity competition. Because it is specific technology agnostic, it is intended to remain relevant after specific technologies have been superseded by newer technologies.

4.3.2 Desired Learning/Competition Outcomes

For any cybersecurity competition, the desired outcome(s) must first be determined. Potential outcomes include (but are not limited to):

1. Developing skills in securing host configurations
2. Evaluating collaboration under stress
3. Evaluating an individual's problem-solving ability
4. Recruiting more participants into the field
5. Illustrating how specific techniques fit into a realistic scenario

4.3.3 Competition Settings

Cybersecurity competitions can be held in different settings, for different reasons. Some of these include:

1. Lab style teaching/classroom experience for a group of students (e.g. A college pentesting course with a hacking lab)
2. General educational experience for interested parties (e.g. A college club holding an open CTF with the goal of teaching participants)
3. Friendly competition (e.g. A CTF competition at a cybersecurity conference)
4. Competition among candidates to determine the most skilled (e.g. A king-of-the-hill competition to determine job candidate's skill levels)

5. Competitions meant to test/improve soft skills as well as technical skills. These soft skills could include working well under stress, communicating technical requirements with non-technical stakeholders, working well in a team, or a myriad of other soft skills

4.3.4 Cloud Provider Selection Criteria

The competition needs a place to run. There are multiple things to consider when choosing a public cloud provider. Some of these include:

1. Cost and terms of service
2. Availability of resources
3. Ease of scaling in case the competition parameters include adding additional resources during the competition
4. Knowledge of the specific cloud provider, including any limits on specific types of resources
5. Functionality to support the technical needs presented elsewhere in the framework
6. Legal/Regulatory concerns: Security competitions often include performing malicious behavior
    a. Ensure the cloud provider's requirements are met for whatever level of malicious activity will be occurring
    b. If the cloud provider employs active countermeasures to detected attacks, ensure that these countermeasures won't interfere with the competition, or are disabled by the provider
    c. If needed, check for potential legal concerns such as limitations on specific types of attacks or restricted encryption methods

4.3.5 Minimum Technical Requirements for the Public Cloud Provider

1. Compute – Operating systems, machine images, etc. Almost all competitions will require some form of host(s)
    a. Choice of operating system
    b. Full control of operating system, i.e. root or Administrator level access
    c. Standard operating systems available, for example: modern and legacy versions of Windows and multiple flavors of Linux
2. Network – Connection options *inside* the competition
    a. Hosts and other endpoints in the competition must be able to communicate with each other
    b. Standard protocols such as TCP/UDP, IP, HTTP, and other application protocols, should be supported
    c. Configuration and management should be available as far down the OSI model as possible to provide more flexibility in competition methods. At least as far down as layer 3 (IP)
3. Connectivity to the environment – Connection options *into* and *out of* the competition
    a. The environment should support remote connections of multiple teams/individuals. These connections should all have similar bandwidth and latency to provide as level a playing field as possible. If using a globally distributed public cloud provider, physical location of competition infrastructure may need to be considered

b. This remote connectivity infrastructure should be independent of the competition infrastructure to provide isolation

4. Storage – Files and/or objects required for the competition need to be available to participants

    a. Space: Enough space to store all required file/object resources needed for the competition

    b. Bandwidth: Enough bandwidth to support simultaneous downloads at a reasonable rate

    c. Uploads: If required by the competition format, a place for participants to upload files

5. Legal/Regulatory – Security competitions often include performing malicious behavior

    a. The cloud provider should have well documented policies and procedures surrounding their requirements for whatever level of malicious activity will be occurring

    b. If needed, check for potential legal requirements

4.3.6 Operational Considerations

1. Decide if the competition will be publicly accessible via the internet or internal only. Both methods may be acceptable, and each has pros and cons. Table 4-2 below lists some of the operational concerns to consider when deciding if the competition network will be public on the internet or private only, requiring a VPN.

Table 4-2 - Pros and Cons of Public vs. Private Competitions

Operational Concern	*Competition operational mode*	
	Public	**Private**
Connectivity *into* the environment	Simple: connections are made using configurations provided by the cloud provider	Potentially complex: May require special routing (e.g. VPNs, ACLs, custom NAT)
Connectivity *within* the environment	Partially difficult: Externally facing addresses may not be known until after some infrastructure is built	Simple: All addresses can be predetermined and preassigned and all components can use RFC 1918 internal addresses
Cost (only differences are listed)	More expensive: If domains or public IP addresses are needed, they may need to be bought	Less expensive: Internal only domains and IPs can be used for free
Real world approximation	Close: Most online organizations have a public facing presence	Less realistic, depending on competition: If the competition is simulating a real-world network with a public web presence, this method would be less true to the simulation
Ease of setup	Simpler: Can utilize standard configurations in many places	Potentially more difficult: standard configs need to be modified to work with private IP addresses
Locking down access	Difficult: Preventing access from unwanted parties is much more difficult if the competition is running on public infrastructure	Simple: The internal-only network can be isolated from other networks and access can be strictly controlled
Security of competition materials	Difficult: The competition applications or cloud access management tools must be relied upon to keep any confidential data in the competition secure and keep competition data from transiting untrusted networks without suitable encryption	Simple: All confidential data is stored internally and can only be accessed by appropriate individuals

2. Identify if the competition will include any sensitive or proprietary information. If it will, consider how communications will be encrypted to ensure data remains confidential.
3. Determine the competition timeframe and estimate operational expenses as much as possible in advance.

a. Public cloud providers list their costs for various types of infrastructure. If it is known how long the competition will run for, along with what pieces of infrastructure will be needed, it should be relatively simple to get a rough estimate of how much the competition will cost.
b. Once this amount is known, confirm that it fits expectations and projected budget.

4. All cloud providers provide some method of controlling resources that are deployed (i.e. access to a control plane). This control plane provides a unique view into the competition environment and may be another valid target during the competition. The decision should be made as to whether or not the control plane will be "in scope" in the competition, as there are just as many misconfigurations and vulnerabilities possible in cloud control planes as there are in traditional running systems. See http://flaws.cloud (Piper n.d.) for an example. A competition focused purely on the control plane would be advantageous.
5. In conjunction with the above, most cloud providers provide PaaS (platform as a service) services in addition to just IaaS (infrastructure as a service). The minimum technical requirements in the section above can be implemented in either PaaS or IaaS configurations. As an example of the difference, if a MySQL database was required for the competition, either a virtual machine could be created and MySQL installed and managed manually (IaaS), or a managed database could be created straight in the public cloud provider (PaaS). The IaaS solution allows more customization, which can be useful for teaching. The PaaS solution is usually more robust, simpler, and (not always, depending on size) cheaper.
6. Determine how the competition will be created. Many cybersecurity competitions already exist in some form or another as templates created in a traditional on-premise

virtualization environment. These pre-built competitions can be "ported" into a public cloud environment to take advantage of the benefits of public cloud. However, sometimes it is advantageous to create the competition directly in the public cloud environment.

a. Porting
    i. Porting the competition means that the same image that existed in the on-premise setup is available in the cloud. However, not all cloud providers import virtual machines *exactly* as provided. For example, AWS does allow machines created in a separate hypervisor such as VMWare to be imported, but the machine is slightly modified during the conversion process. Some modifications include: the IP address is reset to allow communication when the machine is launched, native AWS management tools are installed, and the hosts file is modified, among other things.
    ii. Porting can be less work if competition images already exist and don't have to be recreated from scratch in the cloud.
    iii. Porting allows new operating systems that don't have a corresponding image in the cloud provider to be run.
    iv. Keep in mind that not all operating systems, kernel versions, and hardware configurations are supported in all public cloud providers.
b. Natively creating images in the cloud provider
    i. Native images are built to work in the cloud provider. They often are more compatible and natively supported.
    ii. Native images can be more optimized because they have been tuned to work in the specific cloud provide

5 RESULTS OF CCDC CREATION, TESTING, AND PORTING

The results of the CCDC creation process are presented, including the initial plans, network design, server design, and final version. The results of running said CCDC are also presented. Comparisons with other CCDCs are made using personal experience, cost estimates, and feedback from competition participants. The ported CCDC is examined and the results of the porting operation are presented.

5.1 Creating the CCDC

As presented in section 3.2 above, the CCDC was built over the course of a few months and went through various iterations and versions before a final version was produced. The process was iterative: each day, the current build version of the environment was deployed, added to, tested, documented, and destroyed. One of the benefits of this method was that the environment was only active during the time it was being developed, which meant that that was the only time charges were accruing. Another benefit is that it forced the competition to be built in a way that would allow for a completely automated deployment.

While the initial plan for the CCDC process had included using salt (an open source remote execution and configuration management tool) to manage the final state configuration of each of the servers, this plan never developed beyond the initial proof of concept stage. Salt's master/minion architecture made it difficult to automate the deployment method used during the

iterative development process. Either the master would have to be part of the automated deployment, requiring that the competition network be configured to support that, or the master would have to be permanently available outside of each of the competition environments, and the new minion keys would have to be reconfigured on the master every time the environment was redeployed. This process would have caused significant overhead during the development process. Neither of these options was desired so salt was abandoned early in development. In the end, simple AWS user data scripts, including PowerShell, Bash, and shell scripts, were used to automate all operating system deployments. This process will be reviewed later in this chapter.

As the CCDC development progressed, more testing and less development was included in the process. Some work sessions were dedicated entirely to testing a fully deployed environment, without any additional features being added, or testing the deployment process itself. This testing helped to fine tune the deployment process and make it as reliable and efficient as possible.

5.1.1 Network Design

The network design went through multiple iterations before finally ending on a simple two-tier DMZ/internal network with two additions: First, an additional subnet to simulate workstation connections was added. Second, an additional interface was created on the DMZ side of the firewall/NAT device. This interface was given a second public IP address and was set up as a 1:1 NAT for one of the internal servers. Other than adding an additional interface to the correct host and subnet, the additional configuration and routing for the 1:1 NAT was done entirely on the host level.

Part of the reason for this design was the constraints encountered by building in a cloud provider. AWS provides a plethora of network configuration options and services, but nothing focused on lower-level networking. For example, it is impossible to specify static ARP entries on AWS VPC routers. This fits well with AWS's design guidelines, but it makes it difficult if a project is dependent on manipulating lower-level networking infrastructure. Luckily, for the objective of building a CCDC, this constraint didn't present a critical problem.

The final network diagram looked like this:

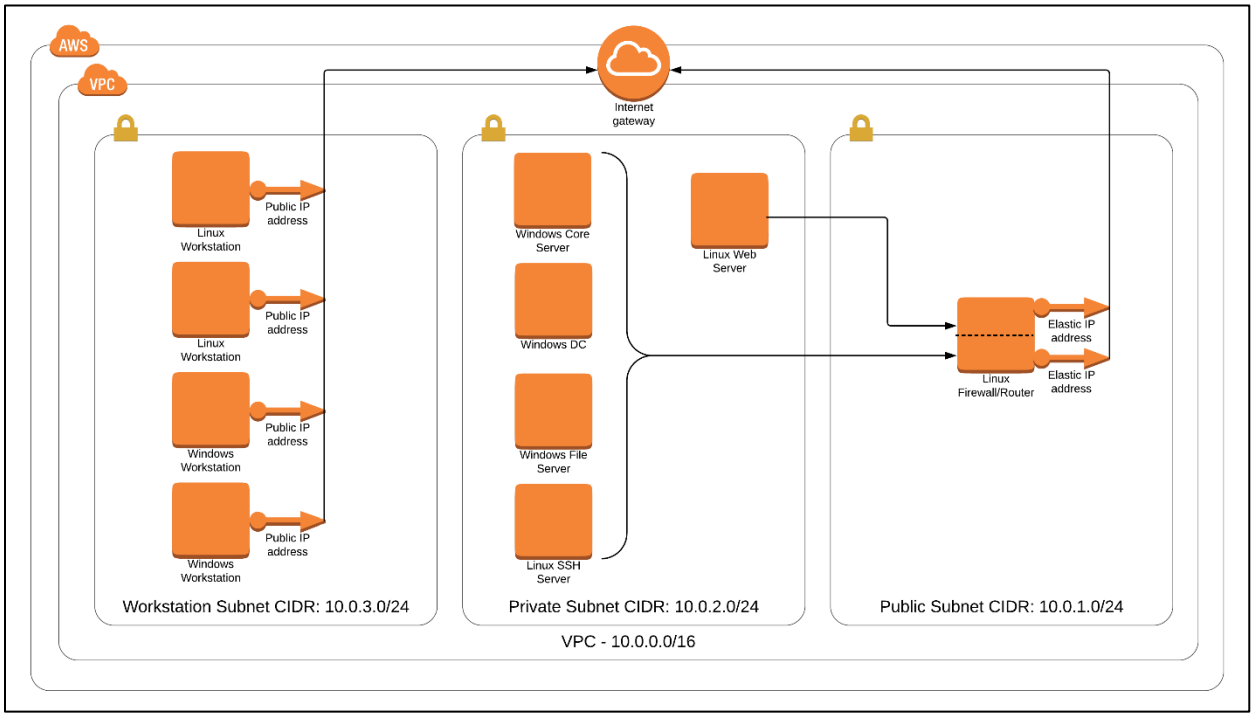

Figure 5-1 - Final Network Diagram Showing Instances and Default Routes

Once the final network design was decided on, the process of automating the deployment was begun. AWS networks belong in VPC (Virtual Private Cloud) instances. Each VPC is an entirely segregated, independent network. This segregation leant itself very well to a CCDC

design: In a CCDC, each team should have an identical copy of a network with identical servers. The more similar the team environments are to each other when starting, the better, as this helps prevent any one team from having an unfair advantage or handicap. The team network was designed and configured on an individual level—in a single VPC—and then cloned for each team in the competition. The VPC network's CIDR range was `10.0.0.0/16`.

After the VPC was created, subnets were added within the VPC's CIDR range. Three subnets were created: a workstation subnet to simulate internal users on a corporate network (`10.0.3.0/24`), a public subnet representing a DMZ (`10.0.1.0/24`), and a private subnet containing servers not meant to be exposed to the internet (`10.0.2.0/24`). The subnets were also given default routes: a concept that is implemented differently in AWS than in traditional on-premise infrastructure. AWS provides a routing table object for each subnet that can be configured via the AWS console. The workstation and public subnets simply used the main VPC gateway as their default route, while the private subnet used the firewall/NAT server in the DMZ subnet as its default route. This design allowed the internal servers to be NAT'd through the firewall/DMZ host, which fit with the network design. While the implementation details of the VPC, subnets, gateway, and route tables were different than what is typically seen in on-premise hardware, the test CCDC network was still able to be built as designed and provided a useful competition environment.

In addition to VPCs and subnets, EIP (Elastic IP) objects were utilized in AWS. In AWS, if a public IP address is requested for a server, a random IP is assigned from the pool of addresses owned by AWS. This address will stay connected to the instance only as long as the instance is running. If the instance is stopped from within the AWS console (powering off the server from inside the operating system is not enough), AWS will release the IP back to its pool.

When the instance is restarted, a new public IP will be assigned. An EIP is a public IP address dedicated to a single account that can be assigned and reassigned at will to different virtual servers within an AWS region. EIPs were used to ensure that, even if the instance was replaced (if the team requested a roll back, for example), it could be assigned the same public IP address(es). The firewall/NAT server in the DMZ subnet was assigned two EIPs, on two separate interfaces. More details on the resulting network configuration for this server are explored later in this chapter.

Using these AWS objects/constructs, including VPCs; subnets (while subnets aren't specific to AWS, this is referring to the AWS specific implementation of subnets inside a VPC); and EIPs, brought to light an AWS-specific challenge that had to be worked through: account limits. In a typical AWS account, there are certain limitations on how many resources of a specific type can be deployed. These limits are imposed on a per-account level "to help guarantee the availability of AWS resources, as well as to minimize billing risks for new customers." AWS does say that they will raise some account limits automatically while an account is in use, but for other limits, AWS support must be contacted to specifically request a limit increase (AWS 2018). Some of these limits include:

Table 5-1 - Default AWS Resource Limits

AWS Service	Limit
VPCs per region	5
EIPs per region	5
EC2 Instances per region	20

In the case of this CCDC, the initial plan was to have three teams of four people each. While the VPC limit wouldn't be a problem, if each team required 2-3 EIPs (an additional EIP slot was reserved for each team for troubleshooting during the competition), the EIP limit would be a problem. In this case, the problem was solved by running each team in a separate AWS region. Another option would have been to contact Amazon and request a limit increase on EIPs per region. However, at the final CCDC event, a 4th team was added, and so an additional network's worth of EIPs would be required. Using separate regions worked in this case, but each competition or event should be evaluated with regards to whatever limits are imposed by the chosen cloud provider.

5.1.2 Server Design

When designing the servers and workstations to be used in the CCDC, there were two methods initially considered: 1) creating virtual machines on a local computer, potentially with an ISO, and moving those VMs into AWS or 2) starting with stock images from AWS and running automated deployment code on them to achieve the final desired end state. After reviewing the different options, it was decided to use the second method: using stock images and running deployment scripts. Using this method improved compatibility. The choice was also simpler to make because the exact configuration of the servers had not yet been decided on. Which stock images were available then helped shape the types of servers that went into the competition. While the final servers changed a small amount from the initial plan, the main design stayed the same. The following table lists all the servers deployed for each of the teams, along with their operating system, function, subnet location, and instance size.

Table 5-2 - CCDC Servers and Functions

Server Name	Operating System	Function	Subnet Location	Instance Size
firewalld	CentOS 7	Firewall for connections coming into the internal subnet and NAT/PAT for connections leaving the internal subnet	Two interfaces in the DMZ subnet: one for NAT and generic incoming connections and one for the full 1:1 NAT to oldJoomla	t2.medium
oldJoomla	Ubuntu 12.04.2	An old (3+ years) version of Joomla running on an old (5+ years) version of Ubuntu. This simulated a legacy system/configuration that had not been updated in a long time	Internal (but with a 1:1 NAT'd public IP address routed through the firewalld server)	m3.medium
domainctl	Windows Server 2012 R2	Domain Controller	Internal	t2.large
storage	Windows Server 2016	Windows File Server	Internal	t2.large
servercore	Windows Server 2016 Core	Target for inject to migrate certain web server data	Internal	t2.large
freebsd	FreeBSD 10.4	SSH Server	Internal	t2.medium
debworkstation (x2)	Debian 8	Linux workstation	Workstation	t2.medium
2008workstation (x2)	Windows Server 2008 R2	Windows workstation	Workstation	t2.large

In order to allow for the automated deployment and testing of an entire environment, each of the servers had to be created in a fully automated method. This resulted in each server being deployed with an AWS user data script. These user data scripts are executed by the hosts when they first boot up. Once they've run once, they don't run again on future reboots. These

scripts were used to set up the system in the desired final state before control was given to the teams. The Linux servers used Bash scripts, the FreeBSD server used a shell script, and the Windows servers used PowerShell scripts. In addition, some of the Windows servers required additional PowerShell scripts that were downloaded and run via the initial user data script. All these scripts can be found in the appendix.

The resulting servers could all be created, configured, and fully deployed within 20 minutes. The majority of this time was spent in configuring and promoting the Windows 2012 R2 server to be a domain controller. The rest of the servers were either waiting for the domain control to finish (most of the Windows servers) or ran their automation scripts and completed well ahead of the domain controller.

5.1.3 Final Automation Design

As described in section 3.2.1 above, the final deployment was automated using AWS's CloudFormation service. The CloudFormation tool "allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts" (AWS n.d.). CloudFormation templates are written in JSON, which can be unwieldy when longer than a few lines. In order to help keep the template code itself understandable and manageable, a tool called Troposphere was used to programmatically generate the JSON files. Troposphere is a Python library for writing CloudFormation templates (troposphere 2018). The automation tasks, including network deployment, server deployment, and server configuration, were all added into a Python file that was then run to fully generate the CloudFormation template to be uploaded to AWS. The final Python file used to generate the

CloudFormation template was less than half the length of the final CloudFormation template itself.

5.2 Running the CCDC

Once the CCDC environment was completely developed and tested, it was deployed, and the competition was given to a group of students in the BYU Fall 2017 CCDC Prep class. Dr. Dale Rowe was the class instructor and helped to split the students into teams and run the competition from an administrative standpoint.

5.2.1 Successes, Failures, and Observations During the Competition

While not flawless, the competition event ran satisfactorily, especially when looking at the infrastructure (network and servers) specifically. One of the perennial problems with on-premise hosted CCDCs has often been performance. It can be difficult to accurately simulate the load a competition team will place on their environment, as well as the load the red team, scoring engine, and other connection attempts will add. Compounding the problem even further is the fact that load testing must be done on all teams' environments *simultaneously*, as that is how the competition itself will operate.

When running a CCDC in a public cloud environment, many of these problems are already solved by the public cloud provider. In the case of this specific competition, as long as each server was built on an instance with enough resources for that individual server's use case, the entire environment performed well. Server requirements and instance sizes were considered while building the environment, and performance was found to be adequate on each of the

servers after it was built. The entire environment, including all networking and servers, functioned well without any performance hiccups for the duration of the competition.

During this CCDC, there was a live red team attacking the competitors' networks and servers. Because the competition was hosted publicly on the internet, the red team didn't have to do any sort of special routing or VPN management. Instead, the red team was just able to attack the targets on public IP addresses. This allowed for simpler red team management.

Another success realized during the running of this competition was the simplicity of licensing for proprietary software. In a typical on-premise environment, running operating systems that require a commercial license can be difficult. Each company has different licensing requirements, some issuing licenses per machine, others per CPU. In order to be compliant with all licensing requirements, a detailed analysis of operating systems and license terms is needed. When operating in a public cloud environment and using the providers images, these licensing requirements are met automatically. The license costs of proprietary operating systems are simply added in to the infrastructure cost of running the system itself. In this way, an expensive single purchase license is converted to an inexpensive license rental. Because the CCDC was run over a short time period, this drastically decreased licensing costs. This and other cost factors are examined later.

In addition to the successes realized while running the competition, there were a few failures observed. The first of these failures concerns region instance limits and AMI mappings:

A total of four teams competed, one more than was originally anticipated. Two different factors combined to cause an unforeseen delay in the competition start time:

1. The limitations shown in Table 5-1 above meant that each region could only support a maximum of one team. The CCDC environment required 10 instances, and there was a limit of 20 instances per region. An increase was requested and granted for this limit, but it was unknown if it would take effect before the competition began. Each team only *required* 10 instances, but in order to support rolling back instances (both old and new instance running simultaneously for a few minutes) and the creation of support instances during the competition, a region limit of 20 was not enough for more than one team.
2. Each of the servers used in the competition environment was based off a single AMI which was built from a specific operating system/version. AMI IDs are specific to each region, which means that even though there may exist identical AMIs in separate regions, the IDs of those AMIs would be different across regions. This was planned for and a CloudFormation mapping was used. This mapped the current region an environment was being deployed into with the correct AMI ID for that region for each of the required servers in the competition. When a new team was added last minute, this mapping had to be created for a new region.

Creating the new AMI ID mapping for an additional region pushed back the start time of the competition by roughly 30 minutes.

Another failure that occurred during the competition was made apparent when multiple teams requested help with isolating issues they were having that prevented them from accessing one or more of their servers. During this time there was an active red team, so it was unknown whether the access problems were because of red team interference, errors on the part of the competitors, or competition infrastructure failures.

5.2.2 Feedback

After running the CCDC as described in the methodology in section 3.2.2, simple feedback was requested from the participants. There was no requirement to give feedback, and the questions were simply to gauge how well the environment functioned and how it compared to previous competitions the competitors had attended. Feedback was received from four participants (about 25%). Three of the four students had been to at least one CCDC in the past.

The responses were all positive when it came to infrastructure performance. The students that had been to a traditional CCDC in the past all mentioned that the infrastructure was faster than previous CCDCs they had attended. While these evaluations are subjective, and the analysis qualitative, the fact that all students mentioned specifically noticing and appreciating the competition performance is significant. Problems with underperforming infrastructure is a common occurrence at traditional competitions. The one student that had never competed in a CCDC also mentioned that the infrastructure performed very well.

For this competition, public IPv4 addresses were given to the internet facing public servers, as well as to the workstation machines. All the students said that they enjoyed using public IP addresses. They liked that it made it possible to check the status of their servers from outside their network and that it didn't require using a VPN to connect and see information. They also liked the fact that the public IP addresses and separate VPCs provided simple isolation between teams.

The feedback for comparisons to traditional CCDCs was also positive. They mentioned again that having public IP address made testing and validating that services were up easier. They also mentioned again that the competition ran quickly and without any performance hiccups. One observant student also mentioned that the scalability of the cloud environment

would be beneficial. They said: “A competition creator can create one network virtually (no need to buy hardware/wait for it to get delivered and set up), test it thoroughly, and then clone it once it's done to fit as many teams as will be competing.” This same benefit was discussed in the previous section.

5.2.3 Costs

The costs of developing the entire CCDC were covered by the AWS Free Tier and a small educational credit. Part of the reason for the limited charges is that servers were only spun up for a few hours at a time while work was being done. All changes were then integrated back into the deployment scripts and the instance was shut down. The charges were incredibly small: all together, they were **less than $5**.

When the time came to run the full CCDC, all required team environments were launched. This involved deploying the CloudFormation template in AWS once per team. The template deployed the required VPC and servers and ran the deployment scripts on the servers. For the duration of the competition, there were four teams competing with anywhere from 9 to 11 instances running at any one time. Instance count variance was due to teams periodically requesting that their servers be rolled back. A new server would be stood up and the old one taken down. This resulted in 36-44 servers running simultaneously. The competition ran for roughly four hours and the servers were left online for about one hour after the competition to allow teams to debrief and scrutinize their configurations outside of the pressure of the competition. The total operational cost (not including development time) at the end of the competition, after all resources had been destroyed, was **$33.06**.

Instance Type	Dec 15, 2017	Dec 16, 2017	Instance Type Total
Total cost ($)	2.90	30.16	33.06
t2.large ($)	2.78	22.35	25.13
t2.medium ($)	0.05	5.20	5.24
m3.medium ($)	0.07	2.43	2.49
t2.micro ($)		0.00	0.00
No Instance Type ($)	0.01	0.19	0.19

Figure 5-2 - AWS Cost Breakdown by Instance Type for Competition Days

These costs encompassed all infrastructure costs related to the CCDC including power, cooling, compute, storage, and network bandwidth for ~40 servers; public IP addressing and connectivity for 20 servers; and storage and cabling for all servers.

Adding up all the instance types used in the competition and compiling their CPU and memory requirements gives a total of 19 CPU cores and 58.75 GB of memory. Although a single, high end industry standard server could theoretically support all four teams simultaneously, the IO performance would take a large hit. While solid state drives would help, they are more expensive. Additionally, having two servers running simultaneously to provide high availability would be ideal. Altogether, $10000-$15000 worth of hardware would be needed

to run all 40 servers concurrently and with adequate performance for the competition to be lifelike. The hardware used for the competition could of course be put to other uses outside of the competition. If the $10000-$15000 cost mentioned above is spread out across the lifespan of the hardware, and the amount of time needed to create and run the competition is known, a cost can be estimated. The following assumptions will be made: A standard five-year (60 month) depreciation lifespan, the competition takes four months to create and run, and the hardware is fully utilized for the balance of its usable lifespan. This results in a 6.667% utilization rate, or **$667-$1000**, which is **20-30x** the entire infrastructure cost of running the competition in AWS.

Power costs for the servers must also be considered. Totaling all the time needed for creating the CCDC scenario, along with the time actually running the competition, and assuming things are shut down when organizers aren't working on the competition, the server hardware would likely be running for ~one month. Estimating two industry standard servers using 500 watts of energy each (Rasmussen 2013), and 10.5 cents per kilowatt-hour (U.S. Department of Energy 2018) the total power cost would be **$76.65**. While actual power costs would vary depending on the server footprint, preparation time, cost per kilowatt-hour, and other factors, this power estimate by itself is already more than **twice** as much as the entire infrastructure costs for the CCDC run entirely in AWS.

In addition to hardware and physical operating costs, licensing of proprietary software (Microsoft Windows, Microsoft Exchange, RedHat, or others) must also be considered. While there are many ways to correctly license these programs, they must be licensed to comply with the terms of service. Licenses for client versions of Microsoft Windows (Windows 7, Windows 10) can cost up to **$200**, while server licenses (Server 2012 R2, Server 2016) can cost over **$1200**. It is possible to host a CCDC entirely without commercial software, and thus avoid

paying licensing fees, but the experience would be sub-optimal. The commercial software used during this CCDC was licensed as part of the infrastructure costs and thus required no additional license fees.

5.3 Porting a CCDC

After building and running a CCDC entirely in a public cloud environment, another competition environment was selected and ported into AWS. For this objective, the focus was different. Instead of creating a full CCDC with the end goal of running a live competition, the objective was to execute and examine the porting process and record how well it worked and where it failed.

The competition selected to port had originally been built to run in on-premise hardware. It consisted of 5-7 servers all directly connected to a single network. This made the networking setup incredibly easy. Each host was simply connected to the default subnet and given a public IP address. The servers themselves included both Windows and Linux (CentOS, Ubuntu, and Fedora). The contents and configurations of the servers were not considered for this process, only the operating system and the ability to boot.

Porting the machines included following guidance provided by AWS to create an S3 storage bucket to store the VM images in and an IAM (Identify and Access Management is AWS's tool for managing who has access to resources) role, then allowing the AWS VM Import Service to assume that role. These steps were performed successfully and the S3 bucket and role were created and ready for the port.

The first step in porting the machines involved uploading the machine image to S3. A command was then issued to AWS to read the image file and begin converting it into an AWS

image. Other than an initial problem with inadequate bandwidth that was later solved, the upload process completed without incident. The conversion process was where the largest number of errors was observed. The first attempt at converting the machines into an AWS-compatible format failed on all servers with various errors about the disk image not being supported. After further research, the virtual machines were converted to .ova files via VMWare Workstation then uploaded and converted again. This time the conversion process completed successfully for all machines—other than a server running a previous version of Fedora—and AMIs were created. After examination, it was discovered that the machine running Fedora had a kernel that wasn't supported by AWS's VM import process.

Once the AMIs were created, they were launched in the same way as any other AWS image. After launching, the instances were examined to ensure that their state and configuration hadn't deviated from what they were before the import process. While all the critical attributes of the servers remained the same, there were some distinct observable differences. For example, on the Windows servers, AWS management tools were installed to C:\Program Files\Amazon. This folder also included logs about the instance's launch. In addition, the networking on all servers was reset to allow for a dynamic IP address. If a server had been configured with a static IP address on the primary interface, that address was replaced with a DHCP configuration. The hosts files were also modified.

```
[root@ip-172-31-8-158 ~]# cat /etc/sysconfig/network
# Automatically generated by the vm import process
NETWORKING=yes
[root@ip-172-31-8-158 ~]#
```

Figure 5-3 - The VM Import Process Reconfigured Some Networking Parameters

6 CONCLUSIONS AND FUTURE WORK

The research questions and objectives are re-examined and evaluated. Potential for future work is explored.

6.1 Analysis of Research Questions and Objectives

Each of the research questions and objectives will be restated and evaluated one at a time.

Research Question 1 – *Running a CCDC in on-premise hardware has been successful in the past. What requirements are needed to run such a competition in the cloud?*

This question is answered with the framework developed and presented in chapter 4. The framework was developed with data gathered while creating and running a successful CCDC entirely in a public cloud provider, as well as data/feedback gathered from multiple experts in IT, cybersecurity, public cloud, and CCDCs. The framework as presented is meant to be sufficient guidance to build and run a CCDC in a public cloud provider.

Research Question 2 – *How do the costs and benefits of running a CCDC in a public cloud environment compare generally to running a similar style competition in on-premise hardware?*

The biggest benefit realized when building and running a CCDC in the cloud was cost. CCDCs, as well as other cybersecurity competitions, are typically short-lived. CCDCs also require a large amount of resources—enough to support a full small business/government network for anywhere from 4 to 15 teams. The hardware, networking, and performance requirements of a CCDC are large, but the resources are only needed over a very small time window. This is a perfect use case for public cloud's pay-as-you-go model. When estimating the costs of running an on-premise CCDC, the calculated power bill alone for the servers required to run the competition was more than twice that of the entire cloud-based competition. This was before the required hardware was even considered. After factoring in hardware and licensing, this amount ballooned to more than 20 times the total costs of a similar competition in the cloud. While every competition has different requirements and costs can be offset in multiple ways (donated or legacy hardware, offsetting some costs with BYOD policies, using existing or volume licenses for proprietary software, among others), when starting from nothing, using public cloud infrastructure can be multiple orders of magnitude cheaper.

Another benefit of running the competition in the cloud was that performance monitoring and load testing were significantly easier. In a cloud environment, each server is guaranteed a minimum level of performance, regardless of how many servers are created. When using on-premise hardware, in addition to ensuring that each server has an adequate amount of resources to do its job, the entire system as a whole has to be continually monitored to ensure there are sufficient resources to run all the servers simultaneously. When running servers in the cloud, as long as each server had the appropriate amount of resources, the system as a whole functioned well. The cloud provider transparently handled all the load balancing and server deployment to provide each instance with its allotted resources.

While most of the results were positive, there were a few areas where a public cloud environment wasn't as versatile or useful as an on-premise solution. In most public cloud environments, there are limits associated with an account that limit the amount of certain types of resources that can be created. So even though it seems as though the cloud offers unlimited elasticity and scalability, there are artificial limits placed on how far an environment can actually scale. These limits can be raised, and most problems avoided, but it requires pre-planning and communicating with the cloud provider. In addition, not all limits can be raised high enough.

Another problem is one of intended use case. Most public cloud providers are set up to provide the most commonly used IT scenarios as easily configurable defaults, especially around networking. If a competition involves standard, simple networking, there typically aren't any problems. However, for competitions that center on complex, non-standard, or very low-level networking (OSI layer 1 or 2), or those where networking is the main focus of the competition, public cloud environments can be more difficult to use.

Research Objective 1 – *Develop a technical framework and list of requirements for the specific workload of cybersecurity competitions in a public cloud environment.*

See chapter 4, specifically section 4.3, and the answer to research question 1.

Research Objective 2a – *Create a CCDC entirely in a public cloud provider using the developed framework as a reference.*

See chapter 5 (sections 5.1 and 5.2) and the answer to research question 2. The CCDC was created and run with live competitors and a live red team.

Research Objective 2b – *Port an existing on-premise cybersecurity competition (preferably a CCDC) into a public cloud environment. Use the information gathered to enhance the framework.*

See chapter 5 (section 5.3) and the answer to research question 2. An existing CCDC was ported to a cloud provider. Once the process for converting virtual machine images built for on-premise use to cloud-compatible machines was understood, it was straightforward. There were a few soft limitations in the process (the VM had to be in a specific format before uploading), as well as one hard limitation (certain kernel versions couldn't be uploaded), but the process was generally smooth and was certainly successful.

The information gathered during the porting process was used to enhance the framework. Sections were added around utilizing preexisting competitions along with potential benefits and drawbacks of such an approach.

6.2 Future Work

The cybersecurity field is continuously evolving, and cybersecurity competitions are evolving with it. This research was focused primarily on CCDC style competitions. Other competition types could be explored, as different competition types typically have (sometimes wildly) different requirements. The framework itself could also be expanded on and broadened in scope to include additional comparisons.

6.2.1 Framework

The framework presented in chapter 4, while complete for its intended purpose, could be expanded to include additional considerations. For example, in its current state, the framework

gives general guidance on how to select a public cloud provider. There is no information given on how to compare or rate public cloud providers against each other. A numeric (or other) ranking system could be developed that would allow users of the framework to evaluate multiple public cloud providers against their specific competition's needs and workload. This would allow a user to find the most suitable public cloud provider.

Another way the framework could be extended is to consider different competition types specifically. Each type of cybersecurity competition has different technical requirements. Currently the framework lists general technical needs and expects the users to understand which of those needs apply to their competition type. The framework could be extended to include templates for several types of standard competitions and their requirements. Users of the framework would still need to understand their own workloads, but the framework could assist in enumerating additional things that may have been missed.

6.2.2 Competition Prototypes

The CCDC created during this research is available in the appendix. Due to the nature of CCDCs (In a standard CCDC, advanced knowledge of the competitions is prohibited), this specific competition is unable to be reused. However, with proper planning and execution, CCDCs could be built with reuse in mind. For example, students might be required to participate in a simplified version of a CCDC during a class. The competition could be reused across different sections of the class or, with modifications, across semesters. As another example, a CCDC could be built and reused as a training exercise in a corporate environment.

In this research, only a CCDC style competition was built and run to test the framework's feasibility. Moving beyond just CCDC style events, additional types of competitions such as

CTF or CPTC could be built and run with the framework as guidance. The resource requirements for each of these competition types would likely determine how well it would perform, as well as what benefits are gained and what problems are encountered, by moving to the cloud.

6.2.3 CCDC Build Process

The CCDC used in this research was built using publicly available server templates (AMIs) with custom scripts applied to fully configure the templates for the competition environment. Different methods of building the server images could be explored in future research. Some other methods include:

1. Using a configuration management tool that is cloud-aware (for example, salt-cloud) to automate the deployment of servers and network configurations. Salt-cloud would also automatically install the salt minion on each machine. This would allow the final state of the servers in the competition to be configured using salt states instead of custom scripts—a method much easier to maintain and develop. Another benefit would be the ability to run commands across system reboots, a problem that was encountered during this research.

2. Using cloud native images and manually configuring them, then snapshotting them and deploying the snapshots for each team. This process would still depend on the cloud provider having prebuilt images for the operating systems needed for the specific competition, but the development process would be simplified. Instead of requiring all configurations for the system to be done entirely from the command line in a single script, native OS tools could be used to develop the system to exactly what it should look like for the competition. Once the system is developed, a snapshot can be taken and then

repeatedly deployed for each team. This would still ensure that each team is using an exact copy of the original system. Any changes that need to be made on a per-team basis, such as adding team numbers to webpages or changing public IP address designations, could be done in a simple boot script injected via user data.

6.3 Recommendations and Conclusions

After building and running a CCDC in a public cloud environment, a few recommendations have surfaced that public cloud providers could implement to make transitioning a competition into their environment easier. These recommendations include:

1. Enable the use of lower-level networking primitives such as static ARP entries on routers
2. Allow (with appropriate warnings) the use of older, potentially vulnerable kernels and operating systems
3. Allow more provider-controlled infrastructure (such as network routers) to be replaced with user-controlled devices

This thesis and research have shown that running cybersecurity competitions in the cloud is not only feasible, but incredibly beneficial. It has also provided a framework that can be used to make informed decisions about moving specific competitions into the cloud. While not perfect for all competition scenarios, utilizing a public cloud model for some competitions—CCDCs in particular—can drastically decrease costs, increase performance, and simplify licensing. So long as the competition type is compatible, the cloud should be considered for use in cybersecurity competitions in the future.

REFERENCES

Amazon Web Services. *AWS re:Invent 2016: Tuesday Night Live with James Hamilton.* November 30, 2016. https://youtu.be/AyOAjFNPAbA?t=21m0s.

AWS. *AWS CloudFormation.* n.d. https://aws.amazon.com/cloudformation/ (accessed October 16, 2018).

AWS Documentation. *Importing a VM as an Image Using VM Import/Export.* 2018. https://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html (accessed 10 24, 2018).

AWS. *How do I manage my AWS service limits?* Edited by AWS. 5 21, 2018. https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/.

Bianchini, R., and R. Rajamony. "Power and energy management for server systems." *Computer* 37 (11 2004): 68-76.

Businessman, L. *cLEMENCy - Showing Mercy.* Edited by Legitimate Business Syndicate. 10 7, 2017. https://blog.legitbs.net/2017/10/clemency-showing-mercy.html.

Carnegie Mellon University Crowdfunding. *picoCTF 2018: a middle & high school cybersecurity competition.* 2018. https://crowdfunding.cmu.edu/project/9737.

CDW. "Server Virtualization: Decrease IT Cost and Data Center Space." techreport, CDW, n.d.

Chen, L., S. Patel, H. Shen, and Z. Zhou. "Profiling and understanding virtualization overhead in cloud." *Parallel Processing (ICPP), 2015 44th International Conference on.* 2015. 31-40.

CloudPassage. *CloudPassage Study Finds U.S. Universities Failing in Cybersecurity Education.* 4 7, 2016. https://www.cloudpassage.com/company/press-releases/cloudpassage-study-finds-u-s-universities-failing-cybersecurity-education/.

Coles, C. *Azure vs AWS vs Google Cloud Market Share.* 6 28, 2016. https://www.skyhighnetworks.com/cloud-security-blog/microsoft-azure-closes-iaas-adoption-gap-with-amazon-aws/.

Comer, D. "Position Paper: Networking Curricula and Laboratories." *Workshop on computer networking: Curriculum designs and educational challenges.* 2002. 21-22.

Corry, J. *INI professor Dr. Martin Carlisle to speak about picoCTF at Global CISO Executive Summit.* Edited by Carnegie Mellon University. 8 20, 2018. https://www.cmu.edu/ini/news/2018/martincarlisleciso.html.

Culbertson, D., D. Humphries, G. Ivy, J. Kolko, and V. Rodden. *Indeed Spotlight: The Global Cybersecurity Skills Gap.* 1 17, 2017. http://blog.indeed.com/2017/01/17/cybersecurity-skills-gap-report/?lang=en.

CyberPatriot. "CyberPatriot Competition Overview." n.d.

Evans, K. S. "US Cyber Challenge." n.d.

Forni, A., and R. van der Meulen. "Gartner Says Worldwide Public Cloud Services Market to Grow 17 Percent in 2016." Edited by Gartner. 9 2016.

Garfinkel, T., and A. Warfield. "What virtualization can do for security." *The USENIX Magazine* 32 (2007): 28-34.

Garg, S. K., S. Versteeg, and R. Buyya. "SMICloud: A Framework for Comparing and Ranking Cloud Services." *Proc. Fourth IEEE Int. Conf. Utility and Cloud Computing.* 2011. 210-218.

Genovese, V. *Building DEF CON CTF.* Edited by Legitimate Business Syndicate. 2017. https://blog.legitbs.net/search/label/Building%20DEF%20CON%20CTF.

Halderman, J. A., et al. "Lest We Remember: Cold Boot Attacks on Encryption Keys." *Proceedings of the 17th USENIX Security Symposium.* San Jose, 2008.

Hofmann, P., and D. Woods. "Cloud Computing: The Limits of Public Clouds for Business Applications." *IEEE Internet Computing* 14 (Nov 2010): 90-93.

Identity Theft Resource Center; CyberScout. *Annual Number of Data Breaches and Exposed Records in The United States from 2005 to 2018 (in Millions).* Edited by Statista The Statistics Portal. July 2018. https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/ (accessed November 26, 2018).

ISO. "Information technology -- Security techniques -- Code of practice for information security controls based on ISO/IEC 27002 for cloud services." Standard, International Organization for Standardization, Geneva, CH, 2015.

Katzcy Consulting. *Cybersecurity Games: Building Tomorrow's Workforce.* Tech. rep., Katzcy Consulting, Katzcy Consulting, 2016.

Kneale, B., and I. Box. "A virtual learning environment for real-world networking." *Information Science* 71 (2003).

Korber, S. *Cyberteams duke it out in the World Series of hacking.* Edited by CNBC. November 8, 2013. https://www.cnbc.com/2013/11/08/defcon-capture-the-flag-competition-is-only-for-top-hackers.html.

LegitBS. *Legitimate Business Syndicate.* 2018. https://legitbs.net/.

Li, A., X. Yang, S. Kandula, and M. Zhang. "CloudCmp: Comparing Public Cloud Providers." *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement.* New York, NY, USA: ACM, 2010. 1-14.

Madnick, S. E., and J. J. Donovan. "Application and analysis of the virtual machine approach to information system security and isolation." *Proceedings of the workshop on virtual computer systems.* 1973. 210-224.

Mansurov, A. "A CTF-Based Approach in Information Security Education: An Extracurricular Activity in Teaching Students at Altai State University, Russia." *Modern Applied Science* (Canadian Center of Science and Education) 10 (aug 2016): 159.

McAfee. *Navigating a Cloudy Sky: Practical Guidance and the State of Cloud Security.* Tech Report, McAfee, Santa Clara: McAfee, 2018.

McKendrick, J. *Cloud May Be The New Outsourcing, But The Same Due Diligence Must Apply.* 10 18, 2014. https://www.forbes.com/sites/joemckendrick/2014/10/18/cloud-may-be-the-new-outsourcing-but-the-same-due-diligence-must-apply/#6a251f310792.

Moore, S., and R. van der Meulen. *Gartner Forecasts Worldwide Public Cloud Revenue to Grow 21.4 Percent in 2018.* Edited by Gartner. 4 12, 2018. https://www.gartner.com/newsroom/id/3871416.

Morgan, S. C. "Cybersecurity Jobs Report." 2016.

Namin, A. S., Z. Aguirre-Muñoz, and K. S. Jones. "Teaching Cyber Security through Competition: An Experience Report about a Participatory Training Workshop." *International Conference on Computer Science Education Innovation \& Technology (CSEIT). Proceedings.* 2016. 98.

Novet, J. "AWS posts $3.53 billion in revenue in Q4 2016, up 47% from last year." Edited by VentureBeat. 2 2017.

NTT Communications. "An Evaluation Framework for Selecting an Enterprise Cloud Provider." techreport, NTT Communications, n.d.

Piper, S. *flAWS challenge.* n.d. http://flaws.cloud/ (accessed November 7, 2018).

Plaid Parliament of Pwning. "About Us." n.d.

Rasmussen, N. *Calculating Space and Power Density Requirements for Data Centers.* Tech. rep., APC, APC, 2013.

RIT. *Collegiate Pentesting Competition.* n.d. https://nationalcptc.org/.

Rowe, D. C., J. J. Ekstrom, and B. Lunt. "Cyber-Security, IAS and the Cyber Warrior." *The Colloquium for Information Systems Security Education (Lake Buena Vista, Fl, 2012).* 2012.

Rowe, D. C., S. Cunha, and C. Cornel. "A Highly Scalable and Reduced-Risk Approach to Learning Network Man-in-the-Middle (MITM) and Client-Side Exploitation (CSE)." *Journal for the Colloquium for Information Systems Security Education.* CISSE, 2017.

Ruiz-Alvarez, A., and M. Humphrey. "An Automated Approach to Cloud Storage Service Selection." *Proceedings of the 2Nd International Workshop on Scientific Cloud Computing.* New York, NY, USA: ACM, 2011. 39-48.

SANS Institute. *OnDemand: Courses & Prices.* Edited by SANS. 2018. https://www.sans.org/ondemand/courses/security.

Simons, J., E. DeMattia, and C. Chaubal. "Virtualizing HPC and Technical Computing with VMware vSphere." techreport, Case Study: Johns Hopkins University Applied Physics Laboratory, 2016.

Spanbauer, S. "Run Multiple Virtual PCs for Free." Edited by PC World. 5 2006.

Stadtmueller, L. "Tips for Choosing a Cloud Service Provider." 3 2012.

Stanton, B., M. Theofanos, and K. P. Joshi. *Framework for Cloud Usability.* techreport, NIST, National Institute of Standards and Technology (NIST), 2015.

Stewart, K. E., J. W. Humphries, and T. R. Andel. "Developing a Virtualization Platform for Courses in Networking, Systems Administration and Cyber Security Education." *Proceedings of the 2009 Spring Simulation Multiconference.* San Diego, CA, USA: Society for Computer Simulation International, 2009. 65:1--65:7.

Taylor, C., P. Arias, J. Klopchic, C. Matarazzo, and E. Dube. "CTF: State-of-the-Art and Building the Next Generation." *2017 USENIX Workshop on Advances in Security Education. https://www. usenix. org/conference/ase17/workshop-program/presentation/taylor.* Vancouver, BC, 2017.

troposphere. "README.rst." *troposphere.* July 1, 2018. https://github.com/cloudtools/troposphere (accessed October 16, 2018).

U.S. Department of Energy. *Electric Power Monthly with Data for April 2018.* Tech. rep., U.S. Energy Information Administration (EIA), Washington, DC: U.S. Energy Information Administration (EIA), 2018.

VMWare. *Virtualization Technology & Virtual Machine Software.* 2018. https://www.vmware.com/solutions/virtualization.html (accessed November 7, 2018).

Von Laszewski, G., J. Diaz, F. Wang, and G. C. Fox. "Comparison of multiple cloud frameworks." *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on.* 2012. 734-741.

vulc@n of DDTek. "A history of Capture the Flag at DEF CON." Edited by Defcon. n.d.

White, G. B., and D. Williams. "The collegiate cyber defense competition." *Proceedings of the 9th Colloquium for Information Systems Security Education.* 2005.

Whitman, M. E., and H. J. Mattord. "The Southeast Collegiate Cyber Defense Competition." *Proceedings of the 5th Annual Conference on Information Security Curriculum Development.* New York, NY, USA: ACM, 2008. 1-4.

APPENDIX: CCDC AUTOMATION CODE

All build scripts and other files referenced in this document are available in the following locations. The files hosted at each location are identical.

1. https://github.com/mew1033/Thesis-Code
2. https://ccdc-in-the-cloud-stuffs.s3.amazonaws.com/list.html
3. https://cloud-thesis-code.csnewby.com