University of Kentucky
UKnowledge

Information Science Faculty Publications | Information Science

4-20-2015

Teaching Cybersecurity Using the Cloud

Khaled Salah
Khalifa University of Science, Technology and Research (KUSTAR), Abu Dhabi

Mohammad Hammoud
Carnegie Mellon University, Qatar

Sherali Zeadally
University of Kentucky, szeadally@uky.edu

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Follow this and additional works at: https://uknowledge.uky.edu/slis_facpub

 Part of the Data Storage Systems Commons, Digital Communications and Networking Commons, and the Information Security Commons

Repository Citation

Salah, Khaled; Hammoud, Mohammad; and Zeadally, Sherali, "Teaching Cybersecurity Using the Cloud" (2015). *Information Science Faculty Publications*. 50.
https://uknowledge.uky.edu/slis_facpub/50

This Article is brought to you for free and open access by the Information Science at UKnowledge. It has been accepted for inclusion in Information Science Faculty Publications by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

Teaching Cybersecurity Using the Cloud

Notes/Citation Information
Published in *IEEE Transactions on Learning Technologies*, v. 8, no. 4, p. 383-392.

© 2015 IEEE

The copyright holder has granted the permission for posting the article here.

Digital Object Identifier (DOI)
https://doi.org/10.1109/TLT.2015.2424692

This article is available at UKnowledge: https://uknowledge.uky.edu/slis_facpub/50

Teaching Cybersecurity Using the Cloud

Khaled Salah, *Senior Member, IEEE*, Mohammad Hammoud, *Member, IEEE*, and Sherali Zeadally, *Senior Member, IEEE*

Abstract—Cloud computing platforms can be highly attractive to conduct course assignments and empower students with valuable and indispensable hands-on experience. In particular, the cloud can offer teaching staff and students (whether local or remote) on-demand, elastic, dedicated, isolated, (virtually) unlimited, and easily configurable virtual machines. As such, employing cloud-based laboratories can have clear advantages over using classical ones, which impose major hindrances against fulfilling pedagogical objectives and do not scale well when the number of students and distant university campuses grows up. We show how the cloud paradigm can be leveraged to teach a cybersecurity course. Specifically, we share our experience when using cloud computing to teach a senior course on cybersecurity across two campuses via a virtual classroom equipped with live audio and video. Furthermore, based on this teaching experience, we propose guidelines that can be applied to teach similar computer science and engineering courses. We demonstrate how cloud-based laboratory exercises can greatly help students in acquiring crucial cybersecurity skills as well as cloud computing ones, which are in high demand nowadays. The cloud we used for this course was the Amazon Web Services (AWS) public cloud. However, our presented use cases and approaches are equally applicable to other available cloud platforms such as Rackspace and Google Compute Engine, among others.

Index Terms—Cybersecurity, network security, computer security, education, cloud computing, Amazon AWS

1 Introduction

Cybersecurity has become one of the emerging areas of paramount importance to industry, government, and society. Since the last few years, various governmental agencies have started investing heavily in IT and cybersecurity education and training [14], [31]. The demand for highly trained cybersecurity professionals is skyrocketing and will likely continue the same way for many years to come. As a result, teaching of cybersecurity in academia has become increasingly vital and is currently playing a major role in addressing the shortage of skilled workforce in this area.

Effective pedagogy and delivery of cybersecurity material require not only theoretical learning outcomes, but more importantly practical and useful hands-on experience. For instance, in industry, students are not only expected to conceive the theories behind any emerging cybersecurity problem but further to demonstrate the ability of *applying* such theories and, consequently, design, develop, and implement innovative pertaining solutions. To provide students with such a capability, we promote using hands-on, cloud-based cybersecurity laboratories, which involve common security tools, packages, and software that are becoming applicable in cloud computing environments. This will enable students (local and remote) to gain invaluable skills, which are currently in high demand; especially that the cloud model is rapidly gaining acceptance and becoming the paradigm of choice for industry.

In general, designing cybersecurity laboratories requires specific features and has to follow certain guidelines as described in [1], [5], [27], [68]. To summarize, these features and guidelines entail that: (1) the lab machines must have connectivity to the Internet so as to download requisite tools and access online information, (2) the lab machines must be isolated from campus network(s), (3) the lab networking environment should be as realistic as possible and capable of carrying out most of the popular and known cybersecurity exercises in the literature, (4) the lab should be set up in a way that makes it easy to manage, allocate, and scale resources for different assignments with non-uniform complexities, (5) the lab should be manned with a sufficient number of IT staff and technicians to provide adequate maintenance and prompt troubleshooting, and (6) the lab should be distributed in nature and equally accessible to on- and off-campus students.

Besides, offering practical cybersecurity training for local and remote students poses some major challenges. Specifically, in a traditional classroom setting, a cybersecurity lab can be established with a number of machines equipped with relevant security tools and software packages. These machines are typically interconnected via a network that is isolated from the production network(s), either physically or by means of a firewall. With the presence of local and remote students taking the *same* course across multiple university campuses, we need to replicate such a setting. Coordinating, managing and maintaining the same setting across campuses for students taking the same cybersecurity course are usually costly and not viable.

Alongside, with classical lab settings, configuring, installing, scheduling and managing lab equipment and workstations might easily become a burden on lab technicians and instructors. Other than the anticipated costs associated with

- *K. Salah is with the Department of Electrical and Computer Engineering, Khalifa University of Science, Technology and Research (KUSTAR), UAE. E-mail: Khaled.salah@kustar.ac.ae.*
- *M. Hammoud is with the School of Computer Science, Carnegie Mellon University (CMU), Qatar. E-mail: mhhammou@qatar.cmu.edu.*
- *S. Zeadally is with the College of Communication and Information, University of Kentucky, Lexington, KY 40506-0224. E-mail: szeadally@uky.edu.*

Manuscript received 13 Nov. 2014; revised 14 Mar. 2015; accepted 15 Apr. 2015. Date of publication 20 Apr. 2015; date of current version 11 Dec. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TLT.2015.2424692

1939-1382 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the operation and maintenance of lab machines, a large percentage of an instructor's time is often spent on troubleshooting issues and solving (benign and malicious) problems initiated by students. This typically prevents the instructor from focusing on the pedagogical aspects of her/his course. In addition, students can easily get frustrated for being constantly distracted with configuration and installation errors. In practice, scheduling fair and effective lab usage for *large* and *dispersed* classes can cause a major obstacle, especially prior to deadlines when students rush to finish their assignments. In short, with classical lab settings, hands-on assignments can easily get diverted from meaningful, focused and enjoyable exercises to superfluous setup, installation, recovery, management and scheduling chores for lab technicians, instructors and students alike.

In this paper, we argue that the cloud computing model can satisfy and alleviate most (if not all) of the requirements and problems introduced by classical lab settings. The cloud provides on-demand, elastic, dedicated, isolated, scalable, (virtually) unlimited, easily configurable, and equally accessible virtual machines (VMs) for all students whether they are at local or remote sites. We describe how the cloud with such characteristics can address many of the aforementioned challenges caused by classical lab settings. We share our experience on harnessing the power of the cloud to enable an effective delivery of a cybersecurity course with a strong hands-on component. We used Amazon Web Services (AWS) [40] as our cloud platform. AWS offers VMs (or *instances* in AWS parlance) as infrastructure-as-a-service (IaaS). IaaS is the foundation of all cloud services, whereby it allows provisioning and controlling fundamental computing resources and software, including varied CPU, memory and disk capacities as well as arbitrary OS, libraries and applications needed for practical exercises on cybersecurity. Other cloud service models include platform-as-a-service (PaaS) and software-as-a-service (SaaS), which provide *only* middleware and application services that are primarily managed and controlled by cloud providers. Thus, with PaaS and SaaS, users (e.g., instructors and students) cannot have fine-grained control or administrative access over OSes, libraries and network services, which are critical to carry out any useful and practical cybersecurity laboratory. To this end, we note that although we adopt AWS as a cloud platform, our presented use cases and approaches can be easily applied to other IaaS cloud platforms such as Rackspace [44], Google Compute Engine [12], GoGrid [13], HP Cloud [8], and Salesforce [47], among others.

Our one-semester US style cybersecurity course was taught in the spring of 2013 for 18 senior undergraduate students across two campuses (six at one campus and 12 at the other) via a virtual classroom equipped with live audio and video. The goal of the course was to expose students to modern network security issues, protocols and technologies as well as to various means of evaluating and analyzing security solutions and countermeasures. The learning outcomes of the course were set as follows: after finishing the course, students should be able to (1) assess popular network security vulnerabilities and threats, (2) analyze network authentication techniques, (3) apply cryptography and hashing algorithms to secure network protocols and devices, (4) compare and evaluate techniques and technologies related to network firewalls, intrusion detection systems, and wireless security, (5) and identify present and future trends in network security.

The rest of the paper is organized as follows. Section 2 briefly discusses related work and the contributions of this paper. Section 3 provides a brief overview of the AWS cloud environment and the concepts of Amazon AMI images and EC2 instances. Section 4 presents three different approaches for managing student accounts on the AWS cloud. Section 5 highlights and summarizes key advantages of using the cloud for teaching cybersecurity. Section 6 describes a list of cybersecurity hands-on laboratories which can be carried out on the cloud. Section 7 presents and discusses some of the challenges and limitations for using cloud-based cybersecurity labs, and proposes various ways to address these limitations. Course assessment and student feedback are presented in Section 8. Finally, we summarize our main results in Section 9.

2 Related Work and Contributions of This Paper

2.1 Related Work

In the literature, labs for cybersecurity training exist in different forms. One form is the traditional lab setting with multiple interconnected physical machines, network devices and security appliances (e.g., firewall and intrusion detection systems). Some of these lab platforms and testbeds are described in [1], [5], [32]. A major benefit of such types of labs is the ability to provide realistic experience with actual hardware equipment. In [62], a remote access to a traditional security lab is provided to students with the capability of remotely allocating, configuring, parameterizing, and managing physical machines and devices. However, as discussed in Section 5, such a classical approach has many associated overhead costs and clear disadvantages pertaining to setup, configuration, installation, scheduling and management of equipment.

Another common approach utilized for providing practical cybersecurity training is the use of virtual technology to establish remotely-accessible security labs. Access to a virtual lab from outside a university campus is typically done via VPN connections (which need to be setup by students). Among the popular virtual security labs are the Virtual Information Assurance Laboratory (VITAL) [60], DeterLab [11], [25] and Tele-Lab [65]. These virtual labs can facilitate cybersecurity experiments, whereby students can configure a number of networked virtual machines and embark on security offense and defense exercises. The use of the virtualization technology to construct virtual security labs is further described in [21], [22], [23], [24], [25], [26], [27]. As opposed to using cloud-based platforms, virtual lab environments and platforms are not scalable and do not provide open and easily accessible resources to students and instructors. More precisely, they offer limited centralized access to instructors for performing remote activities such as management, monitoring, troubleshooting and grading. In contrast, with a public cloud such as AWS, there is no need for VPN connections because *root* accesses to EC2 instances are always granted, scalability and elasticity are ensured, and access to sites which are usually blocked by

governments and universities but required for cybersecurity training, are provided, to mention a few (see Section 5 for more information on this).

NETinVM, a standalone virtualized environment, is yet another approach that is adopted for cybersecurity training [39]. NETinVM runs multiple virtual machines on a single Linux host, thereby utilizing the concept of User-Mode Linux (UML). Unfortunately, running experiments on many VMs within NETinVM with reasonable performance requires students to own high-speed and large-memory laptops or desktops. In addition, unlike the cloud, NETinVM does not offer remote access control to instructors so as to troubleshoot, check, and grade students' course work. As such, this makes NETinVM a poor fit for instructors.

A combination of public and private clouds (i.e., a *hybrid* cloud) can also be used for conducting effective cybersecurity laboratories. For instance, the authors in [7] demonstrated how public and private cloud platforms such as that of AWS and GENI can be leveraged for specifically designing a course on cloud computing and offering cloud-based education in general. The authors described useful exercises targeted at developing essential cloud skills. Alternatively, a private (on-campus) cloud platform can be built and exploited using open-source and publically available deployment and management software such as OpenStack [36] and CloudStack [9]. Such a platform can be a viable option for delivering cloud-based education as opposed to relying solely on public or even hybrid clouds. Except for the IT equipment cost and maintenance, this option can provide features like scalability, on-demand usage, availability, virtualization, resource sharing, and elasticity that are common on public clouds. Moreover, the option effectively addresses some of the issues related to privacy, security, and data or vendor lock-in, which can be of a major concern to some academic institutions.

To this end, we note that there are several other works that discuss the critical role that cloud computing can play in education, e-learning and distance learning [2], [4], [16], [19], [22], [38], [61], [63], [70]. Most of these works describe how the cloud technology can be a key enabler in education, highlighting the significant benefits that cloud computing can offer to institutions, instructors, and students. In [2], the authors characterized and compared qualitatively the various commercial cloud platforms from Amazon, Microsoft, Google, IBM, and HP, which can be considered for higher education. In [16], the authors briefly described how the cloud technology can help teaching computer science; namely, courses related to web design, database management systems, parallel and distributed computing, and image processing. The authors in [19] discussed how the AWS cloud can be used to teach parallel programming and cluster computing. Finally, in [63], cybersecurity games were developed using the AWS cloud platform to provide students with skills mostly related to network scanning and discovery.

2.2 Contributions of This Paper

This paper is a *major extension* of our short five-page preliminary version which appeared recently in [46]. In this revised and enhanced version, we elaborate on the descriptions of the AWS infrastructure and its suitability for academic hands-on course assignments. We offer guidance to instructors on managing student accounts using AWS. We discuss three possible approaches for pursuing such requisite step, *centralized*, *distributed* and *distributed with consolidated billing* management approaches. Moreover, in this extended version, we compare and contrast cloud-based lab cybersecurity exercises versus traditional ones. Furthermore, we present alternative cybersecurity lab exercises (with relevant references) which can be carried out on the cloud. The sections on related work and limitations of cloud-based labs have been greatly expanded as well. An assessment section, which presents a student opinion survey has also been added. Lastly, issues related to cost, ethical considerations in pursuing cloud-based labs, and possible enhancements for future course offerings have been also discussed.

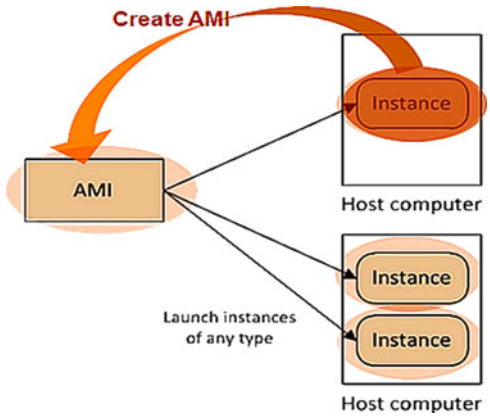

Fig. 1. Amazon AMIs and Instances.

3 The AWS Cloud: A Brief Overview

Amazon Web Services is a collection of remote computing services, which can be delivered over the Internet by Amazon.com. These services make up together one of the most popular cloud computing platforms nowadays. AWS offers infrastructure-as-a-service in which system resources can be provided to users in terms of Elastic Compute Cloud (EC2) *instances*. EC2 instances are basically compute units or virtual machines that can run any software application. They are offered in different sizes (or *types*) varying from "t1.micro" with one core and 613 MB of memory to "h1.4xlarge" with 16 cores and 60 GB of memory [49]. Instances of any type are launched from Amazon Machine Images (AMIs). An AMI is a template that contains a preconfigured OS, software packages, tools and libraries of a user choice. As shown in Fig. 1, using an AMI, one or many instances can be started. Amazon provides different types of AMIs which could be Windows or Linux based.

One key feature of AWS is that users can create their own AMIs. This is indeed a great advantage for instructors and students alike. Specifically, for a particular laboratory, which might require a specific OS and software packages, an instructor can simply start with a base AMI, which already includes a bare OS like Windows or Linux, launch a respective instance, install the needed software, and carry on tests to ensure that all the installed software work properly. Afterwards, the instructor can create a new AMI off this instance and make it available to students. When students launch this AMI, there will be no need for them

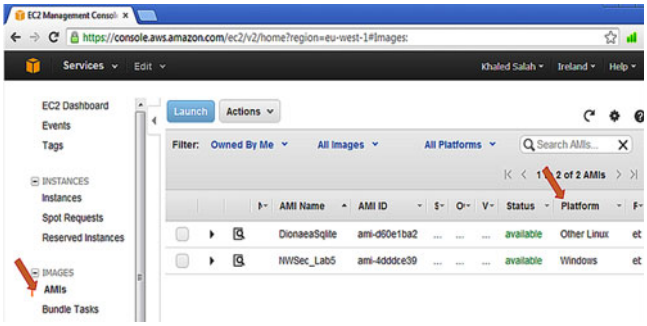

Fig. 2. A snapshot of AWS dashboard showing two AMIs of Linux and Windows.

to perform any type of installation, configuration, or troubleshooting. It is almost guaranteed that all the installed software will work as expected. As such, instructors and students will not be distracted by setup and management activities, and can primarily focus on pursuing lab exercises which predominantly satisfy course objectives.

To demonstrate the concept of AMIs and EC2 instances, Fig. 2 depicts a snapshot of the web-based AWS management dashboard (known as AWS Management Console) which involves two AMIs, "DionaeaSqlite" and "NWSec_lab5". The "DionaeaSqlite" AMI includes the Linux OS and the Dionaea honeypot. This can be used by students to launch Dionaea instances (see Fig. 3). In contrast, the "NWSec_Lab5" AMI incorporates the Windows OS and some preconfigured security tools which can be used to carry out labs 4 and 5 in our cybersecurity course (see Section 6 for details on all our labs). Fig. 3 shows the provisioning of various types of EC2 instances (e.g., Snort, Nessus, and Dionaea), using our "DionaeaSqlite" and "NWSec_lab5" AMIs.

To connect to Windows instances with "Administrator" login privileges, users can use the remote desktop Windows utility or type "mstsc" at a Windows command prompt. On the other hand, to connect as root to Linux instances, the "PuTTY" utility [43] or the JAVA SSH facilitated by the AWS Management Console can be utilized. In general, a student can connect to a Linux instance from a secure shell using an SSH command line as follows:

$ssh -i keypair.pem root@aws_ec2_domain_name

where keypair.pem is a file which gets generated when creating the correct EC2 instance. The file contains the RSA private keys which are required to launch a secure SSH session. This file should be stored securely by the

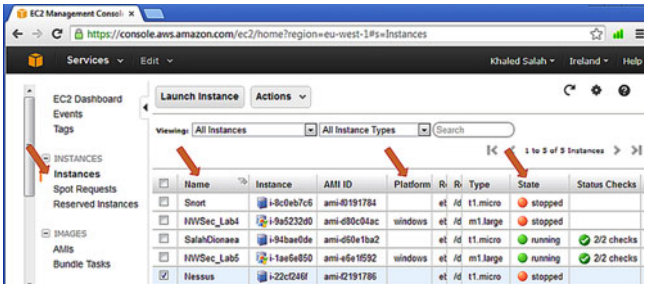

Fig. 3. A snapshot of AWS dashboard showing stopped and running instances.

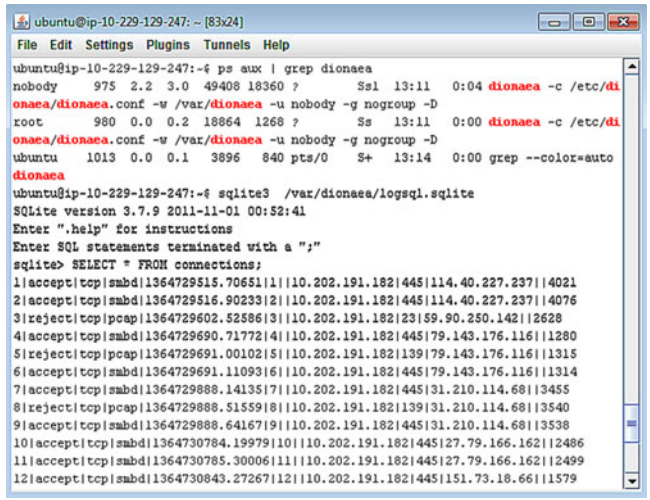

Fig. 4. SSH client screen connected to a Linux instance running Dionaea and Sqlite.

student. The domain name of the EC2 instance is given in aws_ec2_domain_name. Fig. 4 shows a snapshot of an SSH client connected to a running Dionaea instance. The snapshot verifies that Dionaea is running using the "ps" Linux command. The snapshot also depicts the network connections collected by Dionaea. These connections are stored in Sqlite database and could be displayed using SQL commands as demonstrated in the figure.

4 Managing Student Accounts on the AWS Cloud

A key concern with conducting laboratories on the AWS cloud is the creation and management of accounts for students so that they can write and execute code using EC2 instances. At present (February, 2014), managing student accounts is not a straightforward task on the AWS cloud. The challenge stems from the fact that AWS does not yet offer a fine-grained access control for student accounts. Currently, there are three main approaches available, namely (1) *Centralized Management*, (2) *Distributed Management*, and (3) *Distributed Management with Consolidated Billing*, each with its advantages and disadvantages as described next.

4.1 Centralized Management

In this approach, one account (i.e., the *master* account) is set up by an instructor using her/his own credit card. The instructor can add any credit dollar amount she/he has to this account, including grants which can be obtained from the Amazon AWS Education Team. Under this master account, the instructor can create student accounts with different usernames and passwords using the AWS Identify and Access Management (IAM) web service. IAM allows the creation of user groups with specific access privileges. Two groups can be created: *Student Group* (for student users) and *Administrator Group* (for the instructor). The access control policy for the Student Group can be restricted for EC2 management by removing the option "ec2:TerminateInstances" (as shown in Fig. 5). This prevents any student from terminating EC2 instances of other students.

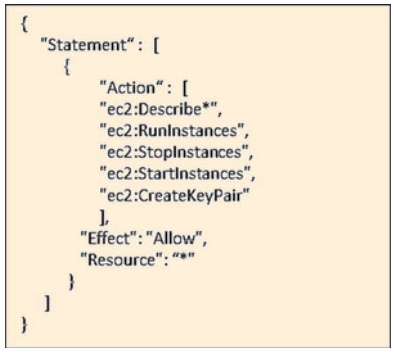

Fig. 5. Access control options for student group.

Nonetheless, with the above access control option, students and instructors can observe all AMI's and EC2 instances of each other. In particular, students can run, stop, and start instances that they have not created, but they cannot terminate them. Clearly, allowing students to inspect and stop instances of each other is a main limitation of this option. Yet, although students can start and run each other instances, they cannot (in principle) login into such instances if they have not created them. As explained in Section 3, to login into an EC2 instance, a user has to use the keypair.pem file which contains the RSA private keys that were generated at the time of creating the instance.

In contrast, a clear advantage of the centralized approach is that the instructor has visibility and full control over all students' activities. Thus, the instructor can easily monitor, debug, and troubleshoot, and (most importantly) verify and grade the work of students. Furthermore, the instructor does not need to make the AMI publicly available, but only privately accessible within the centralized account space (since all AMIs are visible by all users).

4.2 Distributed Management

As described in the previous subsection, one main disadvantage of the centralized management approach is that students can see other students' activities, with control privileges to stop and run their instances. To overcome this limitation, students can be asked to create their own separate accounts using their credit cards.[1] This way, activities of students are not visible to each other; yet every student will pay using her/his own credit card. Also, students are liable for any wrongdoing or overpayment. Thus, this might be an attractive approach as long as each student has a credit card and she/he can use it to open an AWS account. The drawback of this approach is that instructors have to make AMIs publicly available to students and *all* other AWS users. Furthermore, unlike the centralized approach, instructors cannot see students' accounts, and they need to ask students to share their individual usernames and passwords with them so that they can troubleshoot, debug (if necessary), and grade assignments. Lastly, instructors have no control over students' credits, and students cannot share credits among each other (*typically, students exhibit non-uniform usage of credits!*).

1. Amazon grants a credit of 100USD for each student, which can then be added to her/his AWS account.

4.3 Distributed Management with Consolidated Billing

This approach is similar to the distributed management one where each student creates her/his own separate account using her/his own credit card. The difference is that all credits can be accumulated into an instructor account, which can be set up for *consolidated billing* as described in [51]. Afterwards, each student account can be linked to the instructor account to use the credits. This way, any charges made by any student account will be billed to the instructor account. Sharing credits is desirable especially that students do not usually consume credits equally. A caveat, however, is that the instructor needs to constantly monitor the credit consumption of each student so as to avert any misuse of credits. With the default consolidated billing, a bill is generated at the end of every month. For this, the instructor can set billing alerts so that she/he can receive an email if consolidated charges exceed a certain amount. Alternatively, the instructor can configure programmatic billing access in which the monitoring of charges can be done on a daily basis as described in [52].

To this end, we note that for our cybersecurity course, we adopted the first approach. This was mainly because it is easy to apply, and it was the first time for us to pursue cloud-based labs. Now that we have gained more experience and have a better understanding about the different ways of managing student accounts, we are planning to apply the third approach. This is primarily due to its ability to keep student activities hidden from each other and allow instructors to effectively monitor and control student credit usages.

5 Main Advantages for Using the Cloud

We now summarize and highlight the advantages of using the cloud for teaching cybersecurity. A key advantage is the ability to create and use preconfigured AMIs with required security tools and software packages. This advantage was noted briefly in Section 3, and we discuss it further here. Some cybersecurity tools, such as ZeNmap [69], Cain&Abel [6], Wireshark [66], NetCat [28], Openssl [35], and JohntheRipper [20], to mention a few, are easy to install and configure. However, many other tools like Snort [53], Nessus [18], Nexpose [45], Maltego [37], NetWitness [29], and MetaSploit [24], among others, are known for being cumbersome to set up and configure. For instance, Snort (an open source and popular network intrusion detection and prevention system) requires installing and configuring the following supporting applications and software packages: PHP [40], Libcap [57], MySQL [26], Apache Webserver [41], Daemonlogger [10], BASE [3], and many others [54]. For students who are not very familiar with Linux, installing Snort can be an overwhelming task. In contrast, by using the cloud, an instructor can simply develop an AMI that has Snort properly installed *once* on it, and make it available to students (*even for many course offerings*). Subsequently, the students only need to launch instances from this AMI and thereafter directly perform Snort activities. It is worth noting that AMIs which include Snort and Nessus are already available at the Amazon public community. Thus, what remains is simply to download those AMIs and verify that they work properly before making them accessible to students.

Fig. 6. A browser output for www.oxid.it from a local machine (Left) and from an Amazon EC2 instance (Right).

There are many other key benefits when using the cloud infrastructure for pursuing cybersecurity exercises. First, students can rapidly and elastically allocate resources and launch instances. If instances fail, students can quickly start new ones. This is a desirable feature, especially when running security tools and software which often crash and/or cause problems to OSes. Second, as a byproduct, instances running on the cloud do not affect campus production network(s). Therefore, cybersecurity exercises are conducted by students in a safe and *contained* environment. Third, governments and universities often block unwanted websites which are known to contain malicious contents. Examples of these sites are www.oxid.it and www.milw0rm.com. When teaching a cybersecurity course, students need access to these sites so that they can either download tools or familiarize themselves with their contents (to broaden their knowledge). In many countries (e.g., UAE), these aforementioned sites are not reachable (see Fig. 6 Left). However, on the cloud, and as depicted in Fig. 6 Right, these web sites are available for students!. Fourth, instructors can troubleshoot and grade students' work remotely and effectively. Fifth, institutions can avert operating physical laboratories, which usually entail a great deal of administrative costs, overheads, setups, and maintenance. Sixth, students can have *root* accesses to a wide range of *cheap*[2] resources which can be typically provisioned within seconds. Finally, the problem of students competing for limited lab resources during peak periods (e.g., when a deadline approaches) will be entirely eliminated. As a result, students can work on their assignments in accordance to their own schedules from anywhere and at any time.

6 Cloud-Based Cybersecurity Laboratories

In this section, we describe eight of the cloud-based laboratories that were assigned to our senior students as part of the coursework. These labs complement in-class theoretical material and aim to increase student interests as well as enhance their learning experience. The main goal of the labs is to allow students to acquire a deeper understanding of various real-world cybersecurity threats and how to mitigate them using different security software, tools, and appliances. In most of these labs, students were asked to submit reports with adequate snapshots of their work. In several occasions, the teaching staff had to login into students' instances, troubleshoot configurations and grade their work after submissions.

AWS uses a *pay-per-use* model for cloud resources. In order to cut down AWS charges, we asked the students (for most of our labs unless explicitly specified) to use EC2 instances of type "t1.micro". The t1.micro EC2 instances are indeed available for free. Nevertheless, each such instance has low compute power with a 32-bit single core processor, 613 MB of memory and no local storage.

Ethical considerations were also considered and integrated in the design of all our cybersecurity labs. The cloud was used as an alternative platform to carry out exercises with the primary objective of building and developing essential skills in the area, and not to launch attacks, host malware and viruses, or perform unethical conducts of any type. We did not host any vulnerable EC2 instance on the cloud. On the contrary, we strongly emphasized to students to pursue their work ethically and professionally in accordance to: (1) the code of ethics and standards laid out by ACM and IEEE [33], [34], (2) the AWS security polices [50], and (3) our university security policies. Students were continuously asked to carefully self-study, follow and abide by such guidelines and policies. In all our assignments and most of our lectures, we reminded them to adhere to rules and avoid engaging in unethical and unauthorized activities (e.g., attacking and stopping each other EC2 instances, or using the cloud platform to do penetration tests, network reconnaissance, phishing, and eavesdropping).

The following is a list of the eight assigned labs with adequate description and necessary guidelines and references.

- *Packet sniffing*. Tools like Wireshark [66] and TCPDump [56] were used to sniff packets and analyze network traffic generated by a web browser and network utilities such as FTP, ICMP Ping, and Traceroute [58]. The commands to generate this traffic were executed from a Windows and a Linux EC2 instances targeting remote sites, including catless.ncl.ac.uk, hack.me, www.crackmes.de and crackme.cenzic.com. For the site crackme.cenzic.com, students were asked to capture packets containing usernames and passwords. Students were also asked to use the Cain&Abel tool [6] to *easily* filter and capture (i.e., without examining the header fields of TCP packets as is the case with Wireshark) the usernames and passwords sent via the HTTP protocol.
- *Network footprinting and port scanning*. Tools like ZeNmap [69], Whois [64], Maltego [37], Hping [15], and Traceroute [58] were installed on a Linux AMI and made available to students so as to carry out exercises related to footprinting and gathering information about email headers and IP addresses. Also sites like geektools.com, geopiptool.com, shodanhq.com and dnsstruff.com were used to gather more accurate indications about the originating IP addresses and their geographical locations, the ISP provider contact information, and the routes to the originating IP addresses, to mention a few.
- *Vulnerability assessment and penetration testing*. A Windows-based AMI with Nessus [18], Nexpose [45],

2. We note that we had a grant from Amazon of 1,800 USD (100 USD per student), and by the end of the semester we only consumed approximately a total of 300 USD (including usage by the teaching staff). This is a tremendous saving and is substantially lower than the cost of maintaining a *small* classical lab setting.

and Metasploit [24] was created and made available to students as well. Nexpose and Nessus were used to perform vulnerability assessment against another instance running the Dionaea honeypot [42], which contains simulated vulnerabilities. Students had to list and discuss the vulnerabilities reported by Nexpose and Nessus in the Dionaea instance. To gain exploitation skills on a realistic setting, simulated vulnerabilities of Dionaea will not work properly; hence, we had to set up a Windows local machine which has a Mircrosoft IIS FTP service with a buffer overflow vulnerability and leverage it remotely from EC2 instances. The students were asked to use the Metasploit framework which is preinstalled in the Windows AMI to exploit the FTP service vulnerability and run a Windows reverse command line.

- *Backdoor establishment.* To show how sophisticated data exfiltration is performed by an attacker in which network firewalls and IDS/IPS are bypassed, we asked students to use Netcat [28] and establish a backdoor shell via DNS port 53 to bypass firewall rules. For most firewalls, port 53 is always left open for DNS protocol communications. The backdoor was created between an EC2 instance and every student's local machine.
- *Firewalls-EC2, Windows, and Linux.* Students were asked to experiment with configuring an Amazon EC2 firewall through setting and testing simple rules (for inbound and outbound traffic) to a group of EC2 instances [48]. Students were also asked to experiment with a Windows firewall utility via "$netsh advfirewall" [17] and with Linux IPTables to add, delete and test the activation of simple firewall rules. Simple rules were set for passing and denying FTP and HTTP traffic. Lastly, students were also asked to examine the impact of changing the order of rules in the firewall rulebase so as to demonstrate that the first match counts. Each student had to use two machines: one EC2 t1.micro instance with firewall setup and FTP and HTTP services, and another client machine (which could be a local PC, a laptop or an EC2 t1.micro instance) to send HTTP and FTP requests.
- *Snort NIDS.* To acquire skills on network intrusion detection systems, we prepared a Linux-based AMI with the widely used and open source Snort NIDS [53] and made it available to students. Subsequently, students were asked to experiment with Snort and get familiar with its various components, including rulebase, rule structure, MySQL database, log files, and sniffing and alert features. Students were also asked to use Nessus [18], Nmap [30] and Ping to trigger Snort alerting and logging features relevant to network scanning and DDoS attacks.
- *Dionaea honeypot.* A Linux AMI with Dioneae honeypot [42] which encompasses numerous simulated vulnerabilities was made available for students to experiment with as well. The aim of this lab was to teach students skills related to tracking attackers and analyzing network forensics. Students were requested to use (from other EC2 instances) the Ping and the Nmap utilities [30] to target a Dionaea instance and force it to log events. Subsequently, they were asked to query, inspect, interpret and analyze the collected logs and data stored in the Dionaea's Sqlite database.
- *OpenSSL.* To complement the theoretical component of cryptography in the course, students were requested to launch an AMI with preinstalled OpenSSL [35]. OpenSSL implements the basic cryptographic functions. As such, we asked students to perform a variety of tasks using OpenSSL, including creating symmetric and asymmetric keys, producing X.509 certificates, and encrypting, decrypting, and hashing files with different modes and options.

To this end, we note that instructors can assign various other labs (e.g., the ones discussed in [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]), which are also important to cybersecurity training. We note that the majority of the cybersecurity labs which we have seen can be easily enhanced and carried out on the cloud. Among these labs are OS hardening, password cracking, dictionary and rainbow attacks, denial of service attacks, detecting spoofing, buffer overflow vulnerabilities, SQL injection, cross-site scripting, fuzzying, web security and vulnerabilities, Rootkit detection, malware analysis and software cracking, and forensics of captured disk images, among others.

7 Limitations for Using the Cloud and Recommendations to Address These Limitations

In this section, we present some challenges and limitations for using the cloud to teach cybersecurity. We also suggest, whenever possible, ways to address such limitations. Cloud-based teaching requires a learning curve for both instructors and students. From our experience, we found that the learning curve is reasonable especially given that Amazon provides ample videos and webinars for creating Linux and Windows based AMIs and provisioning EC2 instances [59]. A limitation to note, however, is that not every cybersecurity lab is amenable to be carried out using the AWS cloud (or other clouds in general). For instance, certain exercises, which involve disk and smartphone forensics, require the disk and smartphone devices to be directly connected through write blockers to a USB interface of a local machine. To date, cloud-based forensics of such devices via remote network connectivity do not exist, but may emerge in the near future. Another example is one relevant to cracking wireless security protocols; namely, WEP and WPA. As an alternative to using the cloud, students can be asked to perform cracking exercises through setting access points (APs) using their home wireless routers (or smart phones), and perform experiments using bootable Backtrack CDs from their laptops or home desktops.

Network security exercises involving eavesdropping and man-in-the-middle (MITM) attacks are not applicable on the cloud as well. The main reason has to do with network virtualization in which the hypervisor masks out the MAC addresses of the running EC2 instances. Fig. 7 shows two snapshots when running Cain&Abel sniffer to scan and reveal the MAC addresses of hosts within the local network.

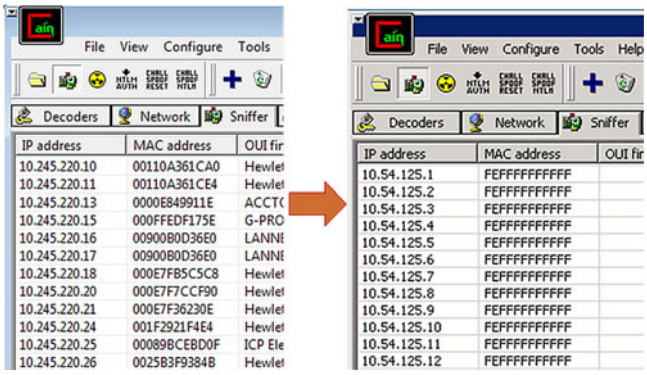

Fig. 7. A snapshot of Cain&Abel sniffer output from a local physical machine (Left) and from Amazon EC2 instance (Right).

We ran the sniffer on a local physical machine and on the AWS cloud. On a local machine connected to a LAN, the sniffer is able to reveal the MAC addresses of the machines within the respective subnet (see Fig. 7 Left). However, the MAC addresses of running EC2 instances within an AWS subnet become meaningless (see Fig. 7 Right). These MAC addresses are needed to perform ARP poisoning so as to pursue a successful MITM attack. To overcome this limitation, instructors can ask students to simply run Cain&Abel from their laptops through their home networks.

Another limitation has to do with managing student accounts. As discussed in Section 4, Amazon AWS does not yet offer a fine-grained access control for managing EC2 accounts. In our first offering of the course we adopted the centralized management approach (see Section 4). Although we gained full monitoring and control over all students' EC2 instances, students were given unauthorized access to each other instances (with permissions of stopping and running instances). This concern has been communicated to AWS people and no solution has been offered yet. As a substitute, in the future offering of the course, we will use the distributed management approach, whereby each student can get a separate account. This approach can be attractive, yet requires each student to open her/his account using a credit card. Besides, available credits cannot be shared among students, and instructors need to make AMIs publically available. In short, there is no single satisfying account management approach thus far. However, we envision that this limitation will be overcome soon, especially with the rapid advances occurring in the cloud computing domain. In the meantime, each instructor needs to weigh the pros and cons of each management approach and selects the one which suits her/him the best.

8 Course Assessment By Students

In this section we report on a survey we ran by our students about the course towards the end of the semester. Some questions asked in the survey as well as overall scores collected and averaged across our two campuses are illustrated in Table 1 (each score is between 1.0 and 5.0). As shown, the average scores reflect that the students were highly satisfied by the course.

More precisely, students agreed that course prerequisites, which include computer networks and operating systems, are appropriate. Nonetheless, some students wished they had more Linux experience. Students did not give a high average score for the textbook [55]. It is generally known that a good textbook on cybersecurity is not available as to date. To fill this critical gap, we incorporated in the course notes and slides various *modern* cybersecurity material, especially those related to our lab exercises. In future course offerings, we will enrich our slides and notes with more information and adopt extra textbooks such as the recently published one by Wu and Irwin [67].

TABLE 1
Course Assessment Results Given by Students with Average Values: 5 = Strong Agree, 4 = Agree, 3 = Neutral, 2 = Disagree and 1 = Strongly Disagree

Question	Local Site (12 students)	Remote Site (six students)
Were course prerequisites helpful?	4.6	4.75
Was the textbook useful?	4.8	3.75
Did the teaching methods help in achieving the specified learning outcomes?	5	4.75
Was the workload of the course appropriate?	4.8	4.75
Did the lab exercises help in gaining new skills?	5	4.75
Did the course spark your interest in the subject?	5	4.75

As shown in Table 1 also, the scores marked by the students from both campuses concerning our teaching methods and course delivery were quite satisfactory. In order to avoid leaving our distant students at a disadvantage, we traveled and delivered five lectures at the remote campus. This improved the interactions between us and the remote students tremendously. Of course, all the remaining lectures were delivered via a virtual classroom setting with live audio and video. Phone and Skype calls had also served us very well over the whole semester. To this end, a field trip to aeCERT facility in Dubai was made, wherein all students from both campuses got together. In summary, the trip was highly motivational and enriched considerably the students' learning experience. Specifically, the students gained a first-hand exposure and deeper insights into cybersecurity activities and operations at the Security Operation Center and the Cyber Crime Forensic Laboratory of aeCERT.

With respect to the lab exercises, the students also gave high scores (as shown in Table 1). Both remote and local students found the lab exercises very helpful. Interestingly, they perceived the overall lab load appropriately. In particular, they provided many positive written comments not shown in the table. For instance, in some written comments, they stated that using the cloud to perform lab exercises was greatly enriching, at least from two perspectives: (1) developing mastery in many hacking and cybersecurity concepts, and (2) gaining various cloud computing skills, which are in high demand nowadays. Several students commented that carrying out lab exercises on the cloud was convenient because they were able to work on them in accordance to their own schedules and from different sites. All students indicated that little time was wasted on configuring and installing software packages and security tools.

Finally, many students expressed that the course sparked their interests and made them think seriously about pursuing careers or graduate degrees in the field of cybersecurity. Two of them commented that the course was one of the best courses they have ever taken.

9 Conclusion

In this paper we illustrated how the cloud computing platform can be harnessed to teach academic courses. Specifically, we described our experience in using the AWS cloud for teaching a one-semester course on cybersecurity for senior undergraduate students across two campuses via a virtual classroom setting. The paper offered valuable guidelines on how the cloud infrastructure and services can be leveraged to pursue popular cybersecurity laboratories. It has been noted that, except for few labs, which require local physical hardware, the majority of hands-on cybersecurity lab exercises can be carried out effectively on the cloud.

To summarize, using the cloud for conducting cybersecurity assignments: (1) allows instructors to easily create *only once* and re-use *over many times* preconfigured AMIs with necessary security tools and software packages, (2) enables students to rapidly and elastically allocate resources and provision EC2 instances, (3) provides a *contained* and *safe* environment, whereby instances cannot affect production networks whatsoever, (4) makes websites (which are typically blocked by governments and universities, yet are essential for teaching cybersecurity) available, (5) grants instructors the ability to monitor, troubleshoot and grade students' work remotely and directly within their instances, and (6) allows precluding costly operations, maintenance and scheduling of physical laboratories, to mention a few. We note that the cost for conducting all our cybersecurity labs on the AWS cloud was low. The total charges were approximately 300 USD for 18 students, including usage and access by the teaching staff.

At the end of the semester, we conducted a student opinion survey about the course. The overall feedback we received was highly encouraging. In particular, students expressed that the cloud-based labs were highly educational and motivational, wherein they were able to gain substantial cybersecurity skills as well as cloud computing management (related to managing and using the cloud resources and services) hands-on experience. Such skills and experience can prove to be extremely valuable for their future careers, especially that both are currently in high demand. Besides, the teaching staff who ran the course found that cloud-based labs are tremendously relaxing, rewarding, convenient and time-saving. The results obtained helped us to focus on improving the overall quality of the course and provided the teaching staff with more time to enhance the content of the course material (e.g., slides, notes and teaching methods). Throughout the course, configuration and setup problems were rarely reported by the students. The teaching staff noticed that the students invested the majority of their time on pursuing the actual assignment tasks which align exactly with course objectives.

For future course offerings, we will continue to improve our set of labs and course material. Among the extra exercises that we might incorporate are the ones that relate to network security (e.g., eavesdropping, fuzzying, and real-time social media and network forensics).

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments, which helped us to improve the content, quality, and presentation of this paper. Khaled Salah is the corresponding author.

References

[1] R. T. Abler, D. Contis, J. B. Grizzard, and H. L. Owen, "Georgia tech information security center hands-on network security laboratory," *IEEE Trans. Edu.*, vol. 49, no. 1, pp. 82–87, Feb. 2006.
[2] F. A. Alshuwaier, A. A. Alshwaier, and A. M. Areshey, "Applications of cloud computing in education," in *Proc. 8th Int. Conf.Comput. Netw. Technol.*, 2012, pp. 26–33.
[3] BASE. (2014) [Online]. Available: http://sourceforge.net/projects/secureideas/
[4] R. Boyatt and J. Sinclair, "Meeting learners needs inside the educational cloud," *Int. J. Learn. Technol.*, vol. 8, no. 1, pp. 61–85, 2013.
[5] J. C. Brustoloni, "Laboratory experiments for network security instruction," *J. Educational Resources Comput.*, vol. 6, no. 4, p. 5, 2006.
[6] Cain and Abel. (2014). [Online]. Available: http://www.oxid.it/cain.html/
[7] P. Calyam, S. Seetharam, and R. B. Antequera, "Geni laboratory exercises development for a cloud computing course," in *Proc. 3rd GENI Res. Educational Exp. Workshop*, 2014, pp. 19–24.
[8] HP Cloud. (2014) [Online]. Available: http://www.hpcloud.com/
[9] CloudStack. (2015) [Online]. Available: http://cloudstack.apache.org/
[10] Daemonlogger. (2014) [Online]. Available: http://sourceforge.net/projects/daemonlogger/files/?source=navbar
[11] DeterLab. (2014) [Online]. Available: http://deter-project.org and http://education.deter-lab.net
[12] Google Compute Engine. (2014) [Online]. Available: https://cloud.google.com/products/compute-engine/
[13] GoGrid. (2014) [Online]. Available: http://www.gogrid.com/2014
[14] HM UK Government. (2013) *A Strong Britain in an Age of Uncertainty: The National Security Strategy* [Online]. Available: http://www.official-documents.gov.uk/
[15] Hping. (2014) [Online]. Available: http://www.hping.org/
[16] L. Huang and Y. Yang, "Facilitating education using cloud computing infrastructure," *J. Comput. Sci. Colleges*, vol. 28, no. 4, pp. 19–25, 2013.
[17] Microsoft Inc. (2014). Netsh commands for windows firewall with advanced security [Online]. Available: http://technet.microsoft.com/en-us/library/cc771920(v=ws.10).aspx
[18] Tenable Network Security Inc. (2014). Nessus vulnerability scanner [Online]. Available: http://www.tenable.com/products/nessus
[19] C. Ivica, J. T. Riley, and C. Shubert, "Starhpcteaching parallel programming within elastic compute cloud," in *Proc. 31st Int. Conf. Inf. Technol. Interfaces*, 2009, pp. 353–356.
[20] JohntheRipper. (2014) [Online]. Available: http://www.openwall.com/john/, 2014.
[21] H. A. Lahoud and X. Tang, "Information security labs in IDS/IPS for distance education," in *Proc. 7th Conf. Inf. Technol. Edu.*, 2006, pp. 47–52.
[22] M. Masud, X. Huang, and J. Yong, "Cloud computing for higher education: A roadmap," in *Proc. IEEE 16th Int. Conf. Comput. Supported Cooperative Work Des.*, 2012, pp. 552–557.
[23] J. Scambray, S. McClure, and G. Kurtz, *Hacking Exposed 7: Network Security Secrets and Solutions*, 7th ed. New York, NY, USA: McGraw-Hill, 2012.
[24] Metasploit. (2014) [Online]. Available: http://www.metasploit.com/
[25] J. Mirkovic and T. Benzel, "Teaching cybersecurity with deterlab," *IEEE Security Privacy*, vol. 10, no. 1, pp. 73–76, Jan./Feb. 2012.
[26] MySQL. (2014) [Online]. Available: http://www.mysql.com/
[27] K. Nance, B. Hay, R. Dodge, A. Seazzu, and S. Burd, "Virtual laboratory environments: Methodologies for educating cybersecurity researchers," *Methodological Innovations Online*, vol. 4, no. 3, pp. 3–14, 2009.
[28] The GNU Netcat. (2014) [Online]. Available: http://netcat.sourceforge.net/
[29] RSA NetWitness. (2014) [Online]. Available: http://www.emc.com/security/rsa-netwitness.htm

[30] Nmap. (2014) [Online]. Available: http://nmap.org/
[31] US NSC. (2013). The comprehensive national cybersecurity initiative [Online]. Available: http://www.whitehouse.gov/cybersecurity/ comprehensive-national-cybersecurity-initiative
[32] RIT Golisano College of CIS. (2013). Networking and systems security laboratory website [Online]. Available: http://nssa.rit.edu/?q=node/52
[33] ACM Code of Ethics. (2014). [Online]. Available: http://www.acm.org/about/code-of-ethics
[34] IEEE Code of Ethics. (2014). [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html
[35] OpenSSL. (2014). [Online]. Available: http://www.openssl.org/
[36] OpenStack. (2015). [Online]. Available: https://www.openstack.org/
[37] Paterva.com. (2014). [Online]. Available: Maltego, http://www.paterva.com/web6/products/maltego.php
[38] H. Peng, "The application of cloud computing technology in distance education network," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2013, pp. 681–683.
[39] D. Perez and C. Perez. (2013). Netinvm: A tool for teaching and learning about systems, networks and security [Online]. Available: http://informatica.uv.es/ carlos/docencia/netinvm/netinvm.html
[40] PHP. (2014). [Online]. Available: http://php.net/
[41] Apache HTTP Server Project. (2014). [Online]. Available: http://httpd.apache.org/
[42] The Honeynet Project. (2014). Dionaea honeypot [Online]. Available: http://www.honeynet.org/project
[43] PuTTY. (2014). [Online]. Available: http://www.putty.org/
[44] Rackspace. (2014). [Online]. Available: http://www.rackspace.com/
[45] Rapid7.com. (2014). Vulnerability management software: Nexpose [Online]. Available: http://www.rapid7.com/products/nexpose/
[46] K. Salah, "Harnessing the cloud for teaching cybersecurity," in *Proc. ACM Tech. Symp. Comput. Sci. Edu.*, 2014, pp. 529–534.
[47] Salesforce. (2014). [Online]. Available: http://www.hpcloud.com/
[48] Amazon Web Services, *Amazon EC2 Security Groups*. (2013). [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html
[49] Amazon Web Services. (2013, Mar.). *Amazon Elastic Compute Cloud (EC2) User Guide* [Online]. Available: http://aws.amazon.com/documentation/ec2/
[50] Amazon Web Services. (2013). *AWS Security and Compliance Center* [Online]. Available: http://aws.amazon.com/documentation/ec2/
[51] Amazon Web Services. (2014). AWS account billing [Online]. Available: http://docs.aws.amazon.com/awsaccountbilling/latest/about/consolidatedbilling.html
[52] Amazon Web Services. (2014). Programmatic billing access," http://docs.aws.amazon.com/awsaccountbilling/latest/about/programaccess.html
[53] Snort. (2014). [Online]. Available: http://snort.org/
[54] Snort.org. (2014). Snort additional downloads: Add-ons and other cool projects [Online]. Available: http://www.snort.org/snort-downloads/additional-downloads#sguil
[55] William Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.
[56] Tcpdump. (2014). [Online]. Available: http://www.tcpdump.org/
[57] TCPdump and Libcap. (2014). [Online]. Available: http://www.tcpdump.org/
[58] Traceroute. (2014). [Online]. Available: http://www.traceroute.org/
[59] Amazon AWS Videos and Webinrars. (2013). [Online]. Available: http://aws. amazon.com/resources/webinars/
[60] Vital. (2014). Virtual lab [Online]. Available: https://vital.poly.edu/
[61] B. Wang and H. Xing, "The application of cloud computing in education informatization," in *Proc. Int. Conf. Comput. Sci. Service Syst.*, 2011, pp. 2673–2676.
[62] K. Webb, M. Hibler, R. Ricci, A. Clements, and J. Lepreau, "Implementing the emulab-planetlab portal: Experience and lessons learned," in *Proc. 1st Workshop Real, Large Distrib. Syst.*, 2004.
[63] R. Weiss, M. Locasto, J. Mache, and V. Nestler, "Teaching cybersecurity through games: A cloud-based approach," *J. Comput. Sci. Colleges*, vol. 29, no. 1, pp. 113–115, 2013.
[64] Whois. (2014). [Online]. Available: http://www.networksolutions.com/whois/index.jsp
[65] C. Willems, T. Klingbeil, L. Radvilavicius, A. Cenys, and C. Meinel, "A distributed virtual laboratory architecture for cybersecurity training," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, 2011, pp. 408–415.
[66] Wireshark. (2014). [Online]. Available: http://www.wireshark.org/
[67] C.-H. J. Wu and J. D. Irwin, *Introduction to Computer Networks and Cybersecurity*. Boca Raton, FL: CRC Press, 2013.
[68] T. A. Yang, K.-B. Yue, M. Liaw, G. Collins, J. T. Venkatraman, S. Achar, K. Sadasivam, and P. Chen, "Design of a distributed computer security lab," *J. Comput. Sci. Colleges*, vol. 20, no. 1, pp. 332–346, 2004.
[69] Zenmaps. (2014). [Online]. Available: http://nmap.org/zenmap/
[70] Q. Zhao, "Application study of online education platform based on cloud computing," in *Proc. 2nd Int. Conf. Consumer Electron., Commun. Netw.*, 2012, pp. 908–911.

Khaled Salah received the BS degree in computer engineering with a minor in computer science from Iowa State University, in 1990, the MS degree in computer systems engineering from Illinois Institute of Technology, in 1994, and the PhD degree in computer science from the same institution in 2000. He has more than 10 years of industrial experience in software and firmware development. He is an associate professor in the Electrical and Computer Engineering Department, Khalifa University of Science, Technology and Research (KUSTAR). He joined KUSTAR in August 2010. Prior to joining KUSTAR, he was an associate professor in the Department of Information and Computer Science, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He has been teaching graduate and undergraduate courses and has more than 100 publications in the areas of cloud computing, computer and network security, operating systems, computer networks, and performance evaluation. He is an Editorial Board member of a number of prestigious international journals including *IET Communications*, *IET Networks*, *Elsevier JNCA*, *Wiley IJNM*, *Wiley SCN*, and *J.UCS*. He received the Khalifa University Outstanding Research Award 2014/2015, KFUPM University Excellence in Research Award of 2008/09, and KFUPM Best Research Project Award of 2009/10, and also received the departmental awards for Distinguished Research and Teaching in prior years. He is a senior member of the IEEE.

Mohammad Hammoud received the PhD degree in computer science from the University of Pittsburgh, in 2010. He is a visiting assistant professor at Carnegie Mellon University, Qatar (CMU-Q). He has a broad interest in computer systems with an emphasis on computer architecture, distributed systems, cloud computing, and databases. For his PhD thesis, he focused on L2 cache design of multicore processors. After joining CMU-Q in 2011, he extended his work to cloud computing where he devised multiple MapReduce scheduling techniques and characterized task parallelism for improved Hadoop performance. Recently, he started exploring ways to offer a cloud support for emerging Big Graph applications and RDF systems. In addition to research, he teaches various courses at CMU-Q, including Distributed Systems, Computer Architecture, Cloud Computing, and Database Applications. He is a member of the IEEE.

Sherali Zeadally received the bachelor's and doctoral degrees, both in computer science, from the University of Cambridge, England, and the the University of Buckingham, England, respectively. He is an associate professor in the College of Communication and Information, University of Kentucky. He is a fellow of the British Computer Society and a fellow of the Institution of Engineering Technology, England. He is a senior member of the IEEE.