CloudWhip: A Tool for Provisioning Cyber Security Labs in the Amazon Cloud

A. Kevin Amorin, B. Shekar NH, and C. Leena AlAufi
College of Computer and Information Science, Northeastern University, Boston, MA, USA

Abstract— *Many traditional techniques of teaching cybersecurity lack realistic environments to gain practical experience. In this paper we present CloudWhip, an open source framework to assist educators with the creation of security labs on Amazon Cloud Services. CloudWhip is developed to be accessible to even those people new to IaaS. We have successfully implemented various network security labs over a three year period in the cloud and our results suggest that the application of cloud computing in cybersecurity education not only saves costs, but also relieves the educational institutions of the burden of handling and maintaining complex IT Infrastructure. Cloud also better emulates managed IT service environments which is essential for SCADA security education. Our lab modules have initiated interest among students and spurred other faculty to conduct numerous security projects using Cloud services.*

Keywords: Cyber Security Labs, Cloud Computing in Education, Amazon Web Services(AWS), SCADA Security Labs, CloudWhip.

1. Introduction

As the number of organizations reporting data breaches in 2013 has increased 30% over 2012[1], the number of attacks continue to rise at a similar rate (about 47k security incidents in 2013[2]). The demand for security professionals continue to increase to handle this threat. According to the International Information Systems Security Certification Consortium (ISC)2 more than 300,000 additional trained cybersecurity professionals are required in 2014[3] to meet the growing demand. This workforce gap has encouraged various government and private organizations to help fund programs designed to train security professionals in higher education. However to apply these core security concepts in industry students need to *"practice the science and the art of computer security"*[4] and many institutions fall short in crossing this chasm between textbook and practical learning.

To bridge the gap in hands-on training, institutions must invest a significant amount in hardware computing resources and SCADA devices. Even with the resources, faculty are tasked with creating challenging and engaging lab exercises using advanced security tools. Unfortunately, during lab exercises students using these programs can inadvertently attack public network computers that are not part of the target environment. Therefore, these exercises require precautions and an appropriate level of isolation from the main university network to avoid collateral damage. This can have extreme side effects if SCADA production environment were affected inadvertently. Needing these isolated clusters, requires the university to devote more resources to maintain and firewall these environments.

The solution to isolating these security labs would be using virtual machines on a different campus LAN network as described in [5], [6] but, as noted above, the main drawback of these architectures are that they require additional resources, time and management. Moreover the scalability and flexibility in such a framework is constrained by the available budget from the university. One other alternative is utilizing a service provider for computing resources. In fact, the use of public cloud computing can present a flexible and cost effective solutions to address these concerns. These services can scale to fit any class load and be customized with policies to allow varying degrees of access to the students. Additionally, the infrastructure services are built to provide redundancy, including backup and storage which prevents downtime or data-loss due to equipment failure. Furthermore, online access and remote access requirements are built into the cloud platform as a requirement.

Cloud computing generally consists of either or a combination of these three main service models - Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). It is beyond the scope of this paper to go into each in detail, but IaaS in this context is where the Cloud Service Providers(CSP) can allow the educator to run virtual machines within the service provider's infrastructure. IaaS provides the option to choose the amount of Disk, Security, CPU, and Bandwidth resources you would like to consume. It also provides the ability to configure these resources to create a very specific custom network environment. One such IaaS provider is Amazon Web Services (AWS).

In this paper we propose an open source framework for deploying security lab environments on Amazon's AWS Cloud Services. The goal of this framework is to be able to implement an existing information security lab in a cloud with minimal knowledge of IaaS. This framework allows instructors to include cloud concepts into the lab or abstract them away if it is beyond the scope of the project. If included, students will be able to control all aspects of the computing platform including provisioning, configuration,

security, termination, monitoring and alerting. Our survey results show that the majority of students have never before had access to manage cloud computing resources. After these labs, if given the choice, most will opt to use similar IaaS features in future security projects. Thereby increasing their knowledge of IaaS along with basic security functionality.

What follows is a review of the related work in Section 2. We then discuss the tools and technologies used to create the lab modules in Section 3. Section 4 illustrates three lab modules that we have implemented in our course work. Next, Section 5 presents the results of the survey conducted and impact of the course modules and environment on student interest and the ease of usage for both students and faculty. Finally, Section 6 concludes this paper and presents future work.

2. Related Work

2.1 In-house Computer Security Labs

In-house computer security labs are those which demand the physical presence of students on campus and hence pose a challenge in current higher education environments. Many in-house security labs such as in [7], illustrate the difficulty in deploying such infrastructure in campus labs. These labs require physical isolation from the main campus network which is time consuming to install and configure and also requires additional resources and maintenance. Another difficulty the authors discuss is maintaining the state of the lab machines throughout the coursework. The execution of lab steps changes the state of the target host and it is not a trivial matter to revert the systems back to their initial state manually.

Another approach illustrated in the NetSecLab[6], consisted of several team machines, victim machines and traffic generator machines. The Traffic generator used a set of scripts to emulate a realistic environment. Such emulators will be restricted to generate traffic based on the pre-configured parameters and thus can only provide pseudo realistic environment. Due to the number of different components and the complexity of the environment, the initial provisioning and maintenance of this lab requires IT to dedicate resources for a significant period of time. If this lab is provisioned in the institution this would require support staff and computing resources. Additionally, scaling a complex support infrastructure with class size will also require scaling the support staff hours.

2.2 Virtual Lab Environments

Efforts have been made to isolate and decentralize virtual lab environment such as in [8]. The authors presents a security lab framework, where pre-configured images of virtual systems are distributed to the students and installed on student's personal computers which provides mobility and flexibility for students while maintaining the state of the system by instructors, as discussed earlier. If the state changes and the system can no longer be utilised for a lab, the initial state can be reverted to through the initial image of the virtual machine created. However, it is hard to conduct labs which require collaboration among students using this system and, this framework is not suitable for a dynamic lab modules. This is because a small change or update in the initial image of the virtual machine by the instructor requires redistribution of the entire image. The uncertainty of a student's personal resources adds a challenge to debugging during lab exercises.

Other virtual lab environments include [5]. They present a distributed virtual laboratory architecture based on the Tele-Lab framework using resources from two different universities with similar course structures. Though these exercises in some way provide realistic implementation for students, there exists scalability issues and collaboration among universities is usually difficult as there is no standard architecture for network and security.

In general virtual environment labs address the issues of mobility, flexibility and maintenance to a certain extent but share the same issue of scaling physical infrastructure as in in-house security labs, adding further cost and time to already overextended in-house staff and infrastructure.

2.3 Private Cloud

Private cloud computing model offers the same basic features as a cloud service but this infrastructure is implemented within the university firewalls, which offers better control over user data and an option to move away from proprietary vendor lock in. In [9] authors used Tele-Lab environments with a middleware layer integrating OpenNebula taking advantage of the cloud framework functionality. Xu and et al., in [10] presented their Cloud based lab called V-Lab which provides a contained experimental environment for hands-on experiments. Building on a private cloud allows educators to utilize existing in-house hardware with the abstraction flexibility of virtualization. Though, image management and debugging become easier compared to a VM decentralization method, this requires the necessary computing, memory and network resources owned and operated by the university to meet the lab's scalability requirements.

2.4 Cloud vs Dedicated Servers

Several research studies like [11] and [12] suggest that the use of cloud computing by educational institutions benefits students by raising their computing resource accessibility irrespective of location, increases availability and mobility. For faculty, it enables them to create custom images for a specific course and share the same infrastructure for different courses if necessary. For administration, cloud computing standardizes application and processes, lightens the burden of software version control and maintenance, optimizes

resource allocation and brings greater visualization. Importantly, it can be cost effective as it saves money on underutilized computing resources, software licensing, and IT staff time. The setup of lab infrastructure in public clouds can be done in few minutes and there exists no downtime during scaling hardware resources. In-house labs or a private cloud, setting up the infrastructure to a working state demands the Instructor or teaching assistant to have a great knowledge of infrastructure management, which is not simple and can be time consuming. Instead, using our proposed framework (CloudWhip), the Instructor can leverage the infrastructure management to the Cloud Service Providers and spend more time on designing the lab modules.

Through our analysis of above mentioned lab environments we observe, that security lab environments are usually designed in isolated network spaces with limitations related to hardware resources and maintenance during scaling along with access restrictions depending on resource availability. Our approach address these issues, while presenting a solution with effective provisioning, as well as a mobile and scalable infrastructure on a Cloud.

3. Tools and Technologies Used

3.1 AWS and AWS Education Grant

Amazon Web Services(AWS)[13], is a collection of IT infrastructure or Cloud Computing services. These services include global computing, storage, database, analytics, application, and deployment to foster organizations scale applications and computing resources on demand at lower IT costs. All the lab modules mentioned in this work were built on AWS services.

AWS in Education is a program that assists educators, academic researchers, and students by providing free usage credits to utilize the on-demand infrastructure of the Amazon Web Services to teach advanced courses, tackle research endeavors, and explore new projects. We have received a grant each of the past three years which helped us provide the labs to the students without any cost to the university. We found the grant application process to be fairly simple and it is available online at [14] for any institution.

Once the grant was approved, the Instructor has the option to receive the AWS credits on his account, or provide it directly to each student in the form of a credit code. The credit code would allow students to manage their own usage, however it does require the student to sign up for an AWS account with a credit card. For these security labs we choose the single central account model to avoid any account provisioning issues. Access to instances is authorized through the Instructor's AWS account by creating accounts in the AWS Identity and Access Management(IAM)[15] service.

3.2 AWS CLI and Boto

Amazon AWS Command Line interface (CLI) allows the user to automate and control multiple AWS services via simple to use tools. Boto is an AWS Software Development(SDK) Kit for Python. It provides Application Programming Interface(API) to many AWS services which eases the process of scripting and automation. The documentation for the AWS CLI and Boto can be found at [16] and [17] respectively. Section 4 illustrates to how we used these tools in our lab environment on Amazon Cloud Services.

3.3 AMIs and EBS

An Amazon Machine Image (AMI) is a template that provides the requisite information (Operating System and applications) to launch an instance. The advantage of creating such a template is that it can be used to launch any number of instances assuring idempotence in the initial state of the virtual machines and also include launch permissions that control the instance, thus easing user and access management in a large deployment. We can also configure an AMI to use an Elastic Block Store (EBS) which allows you to create storage volumes acting like an external block device. Customized AMIs can either be created from scratch or use one of the Amazon provided images as a base to install the required application on top of it. The process of creating your own AMI depends on the root storage of the device - it can either be an Amazon EBS-backed AMI or an Instance store-backed AMI. The steps to create each type can be found at [18][19] and [20][21] respectively.

4. Design and Implementation of Lab Modules

We designed three labs modules for our Network Security Course on AWS. In this section we will walkthrough the steps used to design and implement these labs.

4.1 Lab 1: Gaining Access to OS & Application

The first lab was designed to give an hands-on experience with attacking a target computer. [22] defines first three phases of the attack architecture as Reconnaissance, Scanning and Gaining Access to OS & Application. For diverse exposure in operating systems and applications we build 3 customized AMIs for the lab. The configuration for these AMIs are as shown in Table 1 and the entire architecture for the lab environment is as shown in Figure 1. Every student was assigned to a Point Of Delivery(POD) consisting three systems; an attacker system (Kali Linux) and two victim machines (Windows 2008, CentOS). To access the POD, students would use VNC client such as TightVNC[23] to connect to the X Windows GUI of the attacker system. The VNC port on the attacker system was the only item accessible to external users.

All of the PODs were placed under one large subnet (172.16.0.0/20) and an additional subnet (172.16.255.0/24) acted as a Demilitarized Zone(DMZ) Network. The DMZ consisted two instances running a web application and

MySQL-Server, emulating a Multi-Tier Architecture[24]. Kali Linux was chosen as the attacker system because, this distribution is packed with a wealth of pre-configured security tools such as Metasploit, Nmap and other open source penetration testing tools. Also note that in this architecture, only Kali Linux had a public IP assigned to it so students can reach the system remotely and firewall rules were applied to these subnets such that outbound attack traffic from this system was contained within its own subnet. In case of more granular isolation requirement, each POD can be configured to reside on its own subnet as shown in Figure 2. This configuration requires creation and configuration of more subnets.

In the first phase of this lab, students were allowed to conduct reconnaissance on the network and identify the target systems within the subnet assigned to them. The second involved students performing intense network scan using Zenmap[25] to determine the services that were running on the target systems exploring for any vulnerable application using Nessus[26] in Kali Linux. The final phase of the lab was to use the knowledge gained from the first two phases and try to gain access to OS and applications running on these target systems using the tool Armitage[27]. Amazon Windows AMIs are patched with latest Microsoft security updates and older non-patched versions are not available. Due to the up to date security patches, it is difficult to a student to use common Windows OS exploits available in Armitage. In an effort to in-secure the OS, we tried to remove patches from the default Windows AMIs (2003, 2008). This ended up being counter productive as two issues occurred; first the uninstaller crashed on a number of security patches and failed to back out the change, and second the patches that were removed semi-often caused instability in the OS leading to kernel lockups. For this reason, we focused the attacks in the lab on the applications installed vs. the OS itself. We believe focusing on the application also represents the shift to APT style attacks which have increased in the past decade[28] since Blaster Worm[29]. For this lab we installed a vulnerable Oracle MySQL application, and students exploited the application using the *mysql_payload*[30] module found in Metasploit for UDF payload execution vulnerability.

We included two bonus question for the lab; the first was to brute force *ssh* and gain access to CentOS system, and the second was to exploit a vulnerable e-commerce site in the DMZ and dump all the credit card information stored in a MySQL database. At the end of the lab students were asked to submit a short report on their findings and how they can defend against each phases of the attack architecture covered in this lab.

To implement the lab infrastructure, we first configured the VPC and Subnets using the AWS Console VPC Wizard tool. A AWS security group was created that allowed only the required inbound and outbound traffic to carry out lab exercises, allowing us to contain the attack traffic within the internal lab environment. Finally we associated the subnet with Internet Gateway in the route table console. This enabled any explicitly allowed network traffic to flow out of the VPC to the general internet. A step-by-step guide to manually set up your VPC and subnets can be found here[31]. This entire provisioning and configuration process is very well documented and the AWS console has number of wizards to walk you through the process. Once the VPC and subnet were configured, we utilized the AWS CLI and developed a script to deploy instances in our subnets according to the architecture shown in Figure 1. The script was rewritten to be much more flexible and formed the basis for the CloudWhip tool.

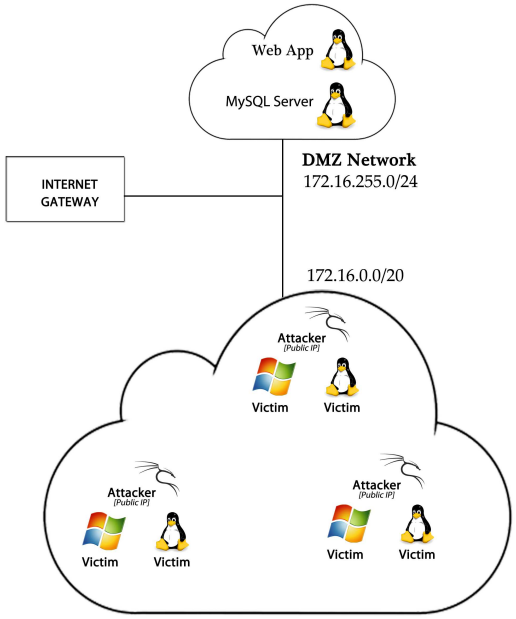

Fig. 1: Lab 1 - Architecture with PODs in Same Subnet

Table 1: Configuration Details of Customizes AMIs.

Operating System	Packages Installed and Additional Configurations
Kali Linux	openSSH, VNC Server, Nmap, Nessus, Metasploit, Armitage
Windows 2k3 R2	mysql (Oracle 5.5.9), Enabled File and Print server roles and removed security updates
CentOS	dovecot, apache web server

4.2 Lab 2: AWS Services and Snort IDS

In this lab students were introduced to AWS Cloud services to deploy and run Snort[32], an Intrusion Detection System. Here we utilized the Identity and Access Management(IAM)[15] service on AWS to create multiple users and manage permissions through Role Based Access

Control(RBAC) system. In the first part of the lab, students login to the AWS Management Console using the credentials emailed to them and launch an existing customized AMI, which is a Linux distribution with Snort pre-installed in it. They also create and apply a new security group while initializing the instance. In this case the security group is wide open to all traffic from any source. This was done to allow the snort instance to get an uncensored view of incomming traffic. In most cases 10-15 minutes after an instance is launched it will start receiving incoming unsolicited requests from scanning systems. These requests are a mix of other AWS instances and external compromised hosts and will generate IDS alerts allowing students to experience a realistic attack traffic environment. Also students are able to experiment with the snort sensor signatures at greater depth. This flexibility would not have been possible with a virtual machine running on student's laptop or virtual machine hosted on our college without significant IT configuration. The rest of the lab focused on configuring Snort sensor and creating rules to alert to various scenarios such as a ssh connection to a particular system, alerting when a particular URL is accessed from the internal network and others. Students used BASE, which is one of the GUI for Snort IDS, to manage and visualize the alert notifications. Instructors can also incorporate a SCADA honeypot as explained in [33] which could use snort alerts to capture packets that match any known SCADA attack profiles. Later this packet capture can to used to replay the attack in a SCADA lab environment. Students can then dissect the attacks and discuss the various appropriate defenses. Optionally students could create and test IPS rules to block these specific attacks vectors and apply them to the SCADA honeypot.

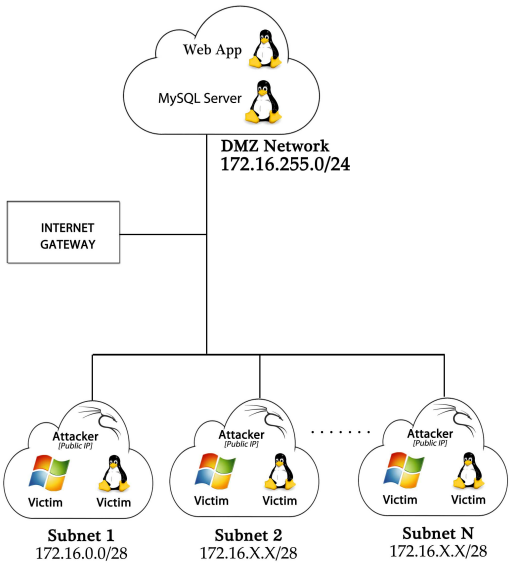

Fig. 2: Lab 1 - Architecture with PODs in its Own Subnet

4.3 Lab 3: Online Brute Force Attack

Verizon Data Breach Report[2] shows 76% of network intrusions in 2013 exploited weak or stolen user credentials, that is by far the largest attack vector. In this lab students performed an online brute force attack from their local computer against a web target hosted in AWS cloud. Each student was assigned an Amazon virtual machine running a web server configured with a basic HTTP authentication for "secret" URLs. Each student had previously in the course installed a local virtual machine of Kali Linux on their personal computers. The first phase of the lab was reconnaissance, where students gathered about 200 user email information associated with target web application using an open source tool *theharvester*[34] from their local VM. Later the students used a password dictionary containing 10k most common passwords[35] and the *hydra-gtk*[36] network logon cracker to brute force a user account gathered in the reconnaissance phase. Once they were able to logon as a valid user, the web page provided them instructions for bonus question. The bonus question comprised of an additional secured URL with a different username and a password generated from a larger phpBB dictionary. This dictionary, which is publicly available at [37], contains 184k clear text passwords from users of phpBB.com. This site was compromised and the MD5 hashes were posted to pastebin, and later were brute forced by [38] and others and made available during DefCon17. A similar attack scenario can be crafted as a lab module to gain access to a publicly facing control system with admin privileges in a SCADA environment, as illustrated in [39]

For this lab the Instructor used the community Ubuntu AMI, installed open source Nginx web server and configured HTTP authentication on specified URLs. A set of well known weak user credentials were used for this purpose so that the students will be able to brute force accounts using the common password dictionary. The goal of the lab was to show level of difficulty in online brute force attacks based on password complexity.

5. Survey and Results

The above mentioned lab modules were implemented on Amazon Cloud Services and used in our Network Security Practices coursework over a period of three years (6 classes: 3 online and 3 on-campus). At the end of the semester during Spring 2014, to evaluate the effectiveness of our Lab modules and the Cloud environment, we conducted an online survey and the results are as follows.

95% of the students agreed that these lab modules aid in better understanding of concepts taught in the class room and 82% of them noticed that conducting labs on a cloud provided them mobility and flexibility in completing their lab exercises. The survey results suggest that our lab modules encouraged most of the class to use Cloud Services in their

Table 2: Survey Results

New Cloud Service users	47%
Difficulty level using AWS Services	Easy: 79%, Moderate: 16%, Hard: 5%
Had performance or access issues	21%
Prefer Cloud Services over VMs on Localhost or College Servers	82%

future security projects, the main reason being flexibility and scalability. Students also commented that they would use Cloud Services more often if it was free. In fact they can register to AWS Free Tier[40] which allows them to use most of the AWS Services for a year free of cost. Table 2 summarizes our survey results.

6. Conclusion and Future Work

In this paper we discussed various drawbacks of some of the traditional cybersecurity teaching methods and how educational institutions, students and faculty can benefit by implementing cyber security labs on a cloud instead. We presented to you three sample lab modules and also demonstrated how to build the environment using AWS-CLI. To further automate the process of deployment, we developed CloudWhip, a wrapper using AWS Boto API, which allows instructors to specify their requirements in a configuration file and deploy the entire lab environment including VPC, Subnets, Instances and Internet Gateway in the Amazon Cloud Services within minutes.

CloudWhip is used to automate the process of deploying and configuring the lab environments. It's goal is to take the time necessary to create a AWS security lab environment from hours to minutes in a simple extensible way. It is under development and is made available at `github.com/NUCyberEd/CloudWhip` under the *MIT License*. The labs discussed above used a very primitive version of CloudWhip. The tool was re-written to support a variety of lab architectures, not only the ones listed above. We would like to extend the CloudWhip project further to cover all the features on Amazon Web Services and provide more granular configuration of lab infrastructures.

We highly encourage course instructors to make use of this wrapper and provide us with reviews and suggestions and share the labs they created using our tool for improvement towards this project.

7. Acknowledgments

This work was possible because of the *AWS in Education Grant* program, which funded us to conduct the security labs on Amazon Cloud Services. We would also like to thank the students of IA5150 for their feedback and faculty and staff at College of Computer and Information Science, Northeastern University for their support.

References

[1] ITRC, "Identity Theft Resource Center - 2013 Breach List," http://goo.gl/ZVzu5k, Tech. Rep., 2013.

[2] Verizon, "2013 Data Breach Investigations Report," 2013.

[3] M. Suby, "The 2013 (ISC)2 Global Information Security Workforce Study," http://goo.gl/dfguAI, Tech. Rep., 2013.

[4] M. Bishop, "Education in information," *IEEE Concurrency*, vol. 1, pp. 4–8, October-Decmeber 2000.

[5] C. Willems, T. Klingbeil, L. Radvilavicius, A. Cenys, and C. Meinel, "A distributed virtual laboratory architecture for cybersecurity training," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, Dec 2011, pp. 408–415.

[6] C. P. Lee, A. S. Uluagac, G. S. Member, K. D. Fairbanks, J. A. Copeland, and L. Fellow, "The Design of NetSecLab : A Small Competition-Based Network Security Lab," vol. 54, no. 1, pp. 149–155, 2011.

[7] L. ben Othmane, V. Bhuse, and L. Lilien, "Incorporating lab experience into computer security courses," in *Computer and Information Technology (WCCIT), 2013 World Congress on*, June 2013, pp. 1–4.

[8] L.-C. Chen and L. Tao, "Teaching web security using portable virtual labs," in *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, July 2011, pp. 491–495.

[9] D. Moritz, C. Willems, M. Goderbauer, P. Moeller, and C. Meinel, "Enhancing a Virtual Security Lab with a Private Cloud Framework," pp. 314–320, August 2013.

[10] L. Xu, D. Huang, and W.-t. Tsai, "Cloud-Based Virtual Laboratory for Network," pp. 1–6, 2013.

[11] M. D. B. Chandra, Deka Ganesh, "Cost Benefit Analysis of Cloud Computing in Education," *2012 International Conference on Computing, Communication and Applications (ICCCA)*, pp. 1–6, Feb 2012.

[12] F. Abidi and V. Singh, "Cloud servers vs. dedicated servers; a survey," in *Innovation and Technology in Education (MITE), 2013 IEEE International Conference in MOOC*, Dec 2013, pp. 1–5.

[13] AWS, "Amazon Web Services (AWS) - Cloud Computing Services," http://aws.amazon.com/, [Online; Accessed 02-April-2014].

[14] AWS Grants, "AWS in Education (Grants)," http://aws.amazon.com/grants/, [Online; Accessed 31-March-2014].

[15] "AWS Identity and Access Management (IAM) in the Cloud," http://aws.amazon.com/iam/, [Online; Accessed 04-April-2014].

[16] AWS CLI, "AWS Command Line Interface," http://aws.amazon.com/cli/, [Online; Accessed 31-March-2014].

[17] AWS Boto, "AWS SDK for Python," http://aws.amazon.com/sdkforpython/, [Online; Accessed 01-April-2014].

[18] AMI Linux EBS, "Creating an Amazon EBS-Backed Linux AMI - Amazon Elastic Compute Cloud," http://goo.gl/u83ypX, [Online; Accessed 31-March-2014].

[19] AMI Win EBS, "Creating an Amazon EBS-Backed Windows AMI - Amazon Elastic Compute Cloud," http://goo.gl/lfwbR6, [Online; Accessed 31-March-2014].

[20] AMI Linux: Store-backed, "Creating an Instance Store-Backed Linux AMI - Amazon Elastic Compute Cloud," http://goo.gl/QcMAQS, [Online; Accessed 31-March-2014].

[21] AMI Win: Store-backed, "Creating an Instance Store-Backed Windows AMI - Amazon Elastic Compute Cloud," http://goo.gl/5sRxvZ, [Online; Accessed 31-March-2014].

[22] T. L. Edward Skoudis, *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, 2nd ed. Pearson Education, Inc., 2006.

[23] "TightVNC: VNC-Compatible Free Remote Control / Remote Desktop Software," http://www.tightvnc.com/, [Online; Accessed 04-April-2014].

[24] Oracle, "Application and Networking Architecture," http://goo.gl/JwS2O2, 2013, [Online; Accessed 18-April-2014].

[25] Zenmap, "Zenmap - Official cross-platform Nmap Security Scanner GUI," http://nmap.org/zenmap/, [Online; Accessed 04-April-2014].

[26] Tenable, "Nessus Vulnerability Scanner," http://goo.gl/sC3OZM, [Online; Accessed 04-April-2014].

[27] "Armitage Tutorial: Cyber Attack Management for Metasploit," http://goo.gl/SS3v0s, [Online; Accessed 04-April-2014].

[28] Mandiant, “Mandiant 2013 Threat Report,” http://goo.gl/lr5JxM, Tech. Rep., 2013.
[29] G. Keizer, “Blaster from the past: The worm that zapped XP 10 years ago - Computerworld,” http://goo.gl/ZBa8uH, [Online; Accessed 17-April-2014].
[30] Rapid7, “Oracle MySQL for Microsoft Windows Payload Execution,” http://goo.gl/JYEj0l, [Online; Accessed 15-April-2014].
[31] AmazonVPC, “Getting Started with Amazon VPC - Amazon Virtual Private Cloud,” http://goo.gl/xzKmHf, [Online; Accessed 15-April-2014].
[32] “Snort :: Home Page,” http://www.snort.org/, [Online; Accessed 04-April-2014].
[33] K. Wilhoit, “The SCADA That Didn’t Cry Wolf,” http://goo.gl/Amw1VG, Trend Micro Forward-Looking Threat Research Team, Tech. Rep. Part 2, 2013.
[34] Edge-Security, “theharvester - The Information Gathering Suite,” http://www.edge-security.com/theharvester.php, [Online; Accessed 17-April-2014].
[35] Hood3dRob1n, “10k Most Common,” http://goo.gl/2NO0X7, [Online; Accessed 17-April-2014].
[36] Van Hauser, “THC-HYDRA - fast and flexible network login hacker,” https://www.thc.org/thc-hydra/, [Online; Accessed 17-April-2014].
[37] SkullSecurity, “Passwords - SkullSecurity,” https://wiki.skullsecurity.org/Passwords, [Online; Accessed 18-April-2014].
[38] S. A. Matt Weir, “Cracking 400,000 Passwords,” http://goo.gl/0vRJp5, Tech. Rep., 2009.
[39] ICS-CERT, “Incident response activity,” http://goo.gl/gRsXtb, Tech. Rep., April 2014.
[40] AWS Free, “AWS Free Usage Tier,” http://aws.amazon.com/free/, [Online; Accessed 18-April-2014].