Poster: Shell We Play A Game? CTF-as-a-service for Security Education

Adam Doupé and Giovanni Vigna
Arizona State University and UC Santa Barbara
doupe@asu.edu, *vigna@cs.ucsb.edu*

I. Introduction

The United States is facing a cyber-security crisis. The supply-side of the cyber-security workforce is not keeping pace with demand: The 2015 (ISC)^2 Global Information Security Workforce Study predicts a shortfall of 1.5 million global information security jobs by 2020 [6]. The lack of qualified cyber-security workforce gives rise to high-profile security incidents, such as the recent Office of Personal Management data breach, where hackers stole 21 million personal files containing sensitive background check information [4]. In addition, attacks against the nation's critical infrastructure can have devastating effect that go well beyond the financial losses we are witnessing today.

The rise in the sophistication of the modern hacker—who waits patiently, quietly leveraging vulnerabilities on one system to compromise another, then slowly exfiltrating sensitive data—demands an equal rise in the skills of security professionals and security-minded developers. Therefore, we must train the next generation of security professionals who will secure the software systems that run companies, organizations, and the nation's critical infrastructure.

Security training requires that developers acquire both the skills necessary to find security vulnerabilities in software, as well as the skills to fix existing flawed software. The knowledge that comes from studying vulnerabilities and vulnerability patterns provides students with the hands-on expertise to complement the theoretical security skills of protection, detection, and response.

Live cyber-security exercises are an excellent tool to help teach and reinforce security concepts in students. In the traditional cyber-security exercise concept, also called *Capture The Flag* (CTF) competitions, the students attempt to discover one or more vulnerabilities in a piece of software (which the organizers created) and then prove that they found a vulnerability by crafting an exploit that takes advantage of the vulnerability, stealing a piece of information from the service (i.e., the flag). At the same time, the students develop patches and defense mechanisms to prevent the exploitation of the vulnerabilities. In this way, the students receive hands-on experience finding vulnerabilities, crafting exploits, and patching services. Previous research work on this topic has shown that not only do the students learn during the competitions, but they also experience significant learning in *preparing* for the competition [1].

Unfortunately, live attack-defense cyber-security competitions place a significant time and effort burden on the organizers, because they require a careful design of the infrastructure and a complex network configuration, including complex routing, network filters, and traffic anonymization[1]. In addition, the creation of vulnerable services requires a skill set that many security educators lack. This limits considerably the adoption of attack-defense live competitions in security curricula.

II. The iCTF Framework

In 2003, the first attack-defense educational competition, called the International Capture The Flag (iCTF), was started at UCSB. The competition was repeated every year (the most recent edition was in March 2017). Each year, the organizers experimented with various designs and approaches to the game [8], [1], [2], [7], [5].

After running the competition for 12 years, however, the organizers recognized that many of the game infrastructure components were reused year after year [9]. Therefore, to ease the burden on other CTF organizers, and to allow educators access to a CTF-like competition for their classroom, in August 2014 the UCSB SecLab released an open-source framework for hosting interactive CTF competitions [3]. By abstracting the common infrastructure (starting services, scoring, service checking, VM creation) and by defining a common interface to create services, the authors of the framework allowed anyone, with significant manual effort, to set up and host a CTF-like competition.

III. Creating CTFs in the Cloud

Even though the iCTF framework provides the components necessary to run a competition, their setup and configuration is far from trivial, and the technical barrier to adoption is still substantial.

Therefore, we decided to design a CTF-as-a-service system. The system, based on the iCTF framework, provides a way for educators to create and run attack-defense cyber-security exercises without needing to know all the details about setting up a complex infrastructure.

The external-facing website contains the functions that allow a user (here, a user is anyone who wants to create an on-demand security competition, which can include students,

[1]There are other security competition designs that are easier to deploy. For example, challenge-based security competition are relatively easy to set up and do not require a complex infrastructure.

educators, CTF teams, or other groups or organizations) to register for an account with the system.

After registration, a user can then create a new competition. At this point, the user is the administrator for the new competition, and can modify all settings. The administrator is able to either select intentionally-vulnerable services from a library of existing services, or write her own vulnerable services, which will then become a part of the library, and available to other educators. The administrator can then set various parameters for the competitions, such as the number of teams, the members of the teams, the game time/length, and so on. Finally, the administration provides credentials for a valid account on the Amazon Web Services (AWS)[2], which are used to instantiate the components necessary to run the competition.

As a first step, the system creates the *game master* component, whose job is to orchestrate the creation of the game infrastructure. Initially, the game master instantiates the *database* and *team services* components. The database stores all the information associated with the competition (e.g., the flags submitted, the status of the services for each round), while the team services component allows teams to interact with the game (e.g., to register users during the pre-competition phase, or to submit flags during the competition). The team services component is accessible through a RESTful API, which can be invoked using a Python module.

Once the competition is about to start, the master creates the virtual machines specific to the competition. These virtual machines include the *gamebot* (which advances the game and computes the team scores), the *scriptbot* (which tests the teams' services and gets and sets the game flags), the *team servers* (which are the team-specific hosts, installed with the vulnerable services), and a *router* that routes traffic between the various servers of the game, and makes sure that the teams are only able to access the proper ports and servers (and not other parts of the game infrastructure). Finally, a *scoreboard* component is created to provide feedback to the teams about their score and the status of their services.

These components are instantiated in AWS using the credentials provided by the user, and organized in a Virtual Private Cloud (VPC). A Virtual Private Cloud allow for the provisioning of a logically isolated section of the AWS cloud. Within the VPC the administrator has full control on the IP addresses of the servers and the routing of the traffic among them. As a result, the traffic of the competition is completely contained within the VPC network, and the attacks carried out in the infrastructure cannot affect external hosts.

Once the game network is successfully created, the administrator can, using the team services, give the teams access to their services, by publishing the SSH keys necessary to log into their respective servers. Finally, when the administrator deems it necessary, she can start the game. When a game starts, the master will configure the router so that the team servers can communicate with one another, allowing the teams to attack the services of other teams. During the competition, the router makes sure that the traffic used to set/get the service flags and determine the status of the service is "blended" with the traffic of the teams, to prevent selective filtering of the traffic.

After the game is over, the master is responsible for collecting any remaining data from the virtual machines and releasing those resources. Finally, the master produces a final report that allows for the *post mortem* analysis of the competition, and stores the complete traffic collected by the router during the game.

[2]Currently Amazon's AWS is the only cloud-based infrastructure supported. However, support for other infrastructure will come soon.

IV. Large-Scale Experience

We tested the scalability of our CTF-as-a-service approach, but running the iCTF on March 3^{rd}, 2017 exclusively in the cloud. This edition of the competition was open to all participants (and not only to educational competitions as it was for previous editions), and 315 teams registered for the competition. We developed 10 services for the competition, which lasted 24 hours (another change from previous editions, which typically lasted 8 hours).

The infrastructure was able to withstand 315 teams attacking each other and the competition did not suffer from any infrastructure problems until six hours from the end of the competition, when one of the teams used a series of scripts to perform a coordinate attack against another team, resulting in a denial-of-service attack against the infrastructure.

The cost of the 24 hours infrastructure was approximately 3,500 USD, and we received a generous sponsorship from Amazon to support the costs of the competition. However, smaller and shorter competitions should cost a fraction of the amount.

This experience proved that it is possible to leverage the on-demand computational paradigm of the cloud to support large-scale cyber-security competitions.

V. Shell We Play a Game?

Given the success of the competition, we decided to launch our CTF-as-a-service web site shellweplayagame.org as a joint project between UCSB and ASU, with the goal of supporting cyber-security education.

In addition, we have made the source code of the iCTF framework used for the creation of the CTF-as-a-service system available at https://github.com/ucsb-seclab/ictf-framework.

We hope that educators from organization of all kinds will use the service to introduce live attack-defense cyber-security competition as integral parts of their security curriculum.

VI. Acknowledgments

This work is based on research supported or sponsored by the National Science Foundation (NSF) under Award No. 1623246, and by the generous contributions of Amazon Web Services (special thanks to the AWS Security team).

REFERENCES

[1] N. Childers, B. Boe, L. Cavallaro, L. Cavedon, M. Cova, M. Egele, and G. Vigna. Organizing Large Scale Hacking Competitions. In *Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, Bonn, Germany, July 2010.

[2] A. Doupe, M. Egele, B. Caillat, G. Stringhini, G. Yakin, A. Zand, L. Cavedon, and G. Vigna. Hit 'em Where it Hurts: A Live Security Exercise on Cyber Situational Awareness. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Orlando, FL, December 2011.

[3] The iCTF Framework. https://github.com/ucsb-seclab/ictf-framework.

[4] R. Jalabi. OPM hack: 21 million people's personal information stolen, federal agency says. *The Guardian*, July 2015.

[5] Y. Shoshitaishvili, L. Invernizzi, A. Doupe, and G. Vigna. Do You Feel Lucky? A Large-Scale Analysis of Risk-Rewards Trade-Offs in Cyber Security. *ACM Symposium on Applied Computing*, March 2014.

[6] M. Suby. The 2015 (isc) 2 global information security workforce study. *Frost & Sullivan in partnership with Booz Allen Hamilton for ISC2*, 2015.

[7] K. Vamvoudakis, J. Hespanha, R. Kemmerer, and G. Vigna. Formulating Cyber-Security as Convex Optimization Problems. In *Control of Cyber-Physical Systems*, volume 449 of *Lecture Notes in Control and Information Sciences*, pages 85–100. Springer, July 2013.

[8] G. Vigna. Teaching Network Security Through Live Exercises. In C. Irvine and H. Armstrong, editors, *Proceedings of the Third Annual World Conference on Information Security Education (WISE)*, pages 3–18, Monterey, CA, June 2003. Kluwer Academic Publishers.

[9] G. Vigna, K. Borgolte, J. Corbetta, A. Doupe, Y. Fratantonio, L. Invernizzi, D. Kirat, and Y. Shoshitaishvili. Ten Years of iCTF: The Good, The Bad, and The Ugly. In *Proceedings of the USENIX Summit on Gaming, Games and Gamification in Security Education (3GSE)*, San Diego, CA, August 2014.