Dynamically Generated Virtual Systems for Cyber Security Education

Steven Morgan and Brent Lagesse
University of Washington Bothell, Bothell, WA, USA
stevem24@uw.edu
lagesse@uw.edu

Abstract: Cyber Security classes are not offered in many colleges and fewer high schools, the main reason is that there are not enough resources in terms of both expertise and equipment for them to start a Cyber Security education program. We are seeking to reduce the overhead cost for offering such classes by developing a system that is cloud-based which will enable students to participate in labs from their school while only needing an internet connection and low-end computers. We are in the process of developing such a cloud system at the University of Washington Bothell and the tool that will generate a working network on the fly with either a probabilistic model or the specific instructions from the user via a network description language. These dynamically generated systems will then be instantiated on the cloud system along with progress monitoring applications that will determine when the student has completed the scenario. Since all of the heavy processing is done on the cloud-side, the school offering such a class only needs to have very basic, low-cost hardware to connect to the remote system, deploy the scenario, and participate in the scenario. We are currently working with the Yakama Nation to deploy a lab for their students to utilize our system. We will initially begin by holding cyber security workshops for the students and then, pending positive results, move to a more regular class-like format, these classes will be designed around teaching through interaction and hands on experience rather than the formal lecture style to help cultivate the ability to think outside the box which is much needed from todays security.

Keywords: cloud computing, education, cyber security, probabilistic model, dynamic, machine generation, virtual machines, vmware, workshops,

Introduction

Networking and security courses rely on using existing hardware to demonstrate how a network works. Many schools have the hardware in use and existing networks but generally cannot be used to test security due to privacy concerns, potential damage to the network and machines, and disrupting existing cyber security mechanisms. Network emulation tools are useful, but they do not give the full depth needed for students to learn how to use industry tools to monitor and test the network, on the account of their lack of fidelity to operational systems.

Most classes cover theory and give examples of how networks are laid out securely and what tools will show the vulnerabilities that exist. A common tool like Nmap might be used by an instructor during class to do a port scan as a demo for the students of what a scan looks like; however, there are few classes that comprehensively teach cyber security in a hands-on and realistic environment. This is a result of the fact that many schools do not have the budget or staff to support a full time class in cyber security.

It is the goal of our work to provide schools with a system for creating low-cost cyber security courses that do not require teachers to be cyber security experts (though some knowledge is required). These requirements are essential for introducing cyber security at a younger age, particularly in the K-12 curriculum. We are approaching this by dynamically generating scenarios in a cloud environment based on near-English descriptions of the scenario. These scenario descriptions will be included as part of a curriculum that is distributed with the software, but they can be extended if a teacher deems it beneficial. The school utilizing this system will only need to provide low-end computers for the student since nearly all of the processing is done on the cloud backend. The cloud system can be provided by a traditional cloud provider or as a private cloud hosted by a local organization or University, and as they becoming cheaper and more widely available

schools could be able to setup their own cloud and not rely on other organizations. (W. I. Bullers06)

Lab Generation

The software we are designing will be able to create networks with a simple script. These networks will be created based on probabilistic models that enable the creation of unique and realistic network systems. Figure 1 demonstrates the process for generating a lab scenario. These models will contain the specifications of how firewalls, workstations, and servers are connected, what OS they are running, and what software is running on the machines. The models will give the user the ability to generate any type of network with very little effort on their part while enabling them to make as many specifications as needed. An exit condition is also included in the script. These exit conditions are the goal of a given scenario. When the student meets the exit condition they pass the scenario.

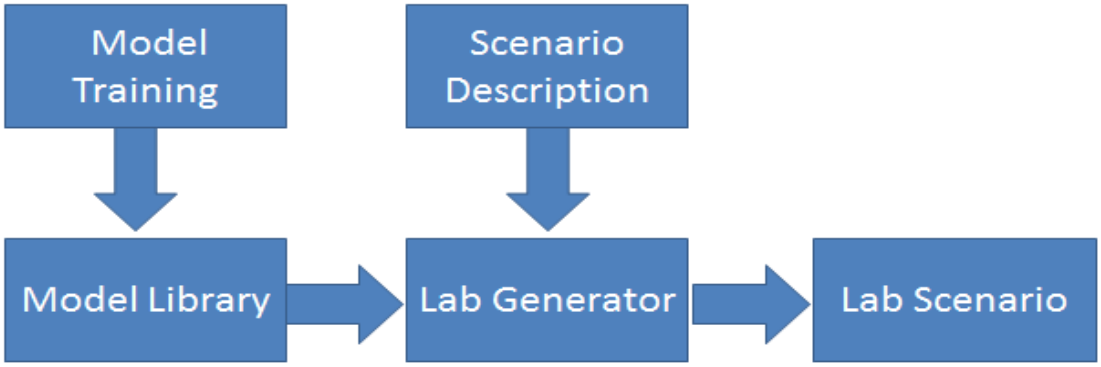

Figure 1: Scenario Generation

The description that the user will input looks much like an English sentence; it's designed for anyone to use without any special training required. In order to generate a network that looks like what would be found in the University of Washington, the user would simply type "Create a system like UW with 100 machines." The program will look through this using regular expressions for the information it needs, in this case the first expression it will look for is in the structure of "System <type> with <size> machines." Any other words it encounters that don't match will be skipped, giving it the ability to process an instruction with a wide range of variance in the words used. However users may try typing something different, say they would type "a network like UW containing 100 Machines." Now this poses some problems the machine will not be looking for the words network or containing, as this project moves on the plan is to augment our techniques with work in natural language processing. This way the network descriptive language will be very natural and easy to learn and remember, the language will have three parts, first is what the declaration of the network setup, the second part allows the user to describe what features each machine has. For instance "half the machines run Windows, all machines have 2 GHz 1 core processors" will make half the machines with windows as the OS and all the machines have 2 GHz processors. Each command is separated by a comma and the parts are separated by a period. The third is the exit goal which will give the condition of when the lab is complete, "all windows machines are updated." will set the goal for the lab so the student will need to update all the machines to complete the lab.

These networks are generated from a model library. The model library contains probabilistic models that have been trained from real world systems, so that our tool can generate realistic systems with the same statistical properties as those that the system trained on. The model library also supports models that are deterministic; however, they are significantly less interesting for teaching and research than the probabilistically created scenarios. These scenarios are then deployed on a cloud system such as the one shown in Figure 2.

To monitor the students' progress each machine will have a monitor program that will communicate with the controller to receive instructions, report latest changes and to report on the configuration of software and settings. The machines will report back to the control machine that the instructor will be running via a second network that will allow the machines to communicate without creating network traffic that might interfere with network related tasks, like packet sniffing or IDS operations. The program will keep the controller informed by reporting status but also with a heartbeat that will allow the instructor to see if the machine is rebooting or crashed.

Teaching students about security and networking will be made easier with hands on experience with access to actual computers on a closed network without needing to worry about damaging a real network. Furthermore, since the networks are probabilistically generated, students can perform the same exercises repeatedly until they learn the concepts without the concern that they memorize the scenario instead of actually learning how to perform the educational tasks. This will promote the ability to quickly assess the situation when presented a problem, which will be of great need in the real world where the networks are not all setup by some template. (A. Conklin06)

By using a generated system during the learning phase the students will not become dependent on a fixed system, rather it will enable them to learn the system they are giving quickly. This skill will be needed for anyone who is going to become a security consultant where they may not work on the same network every week, they will need to be able to look at each network as a unique system with its own problems and strengths without allowing their experiences from previous jobs to cloud their view of the system at hand.

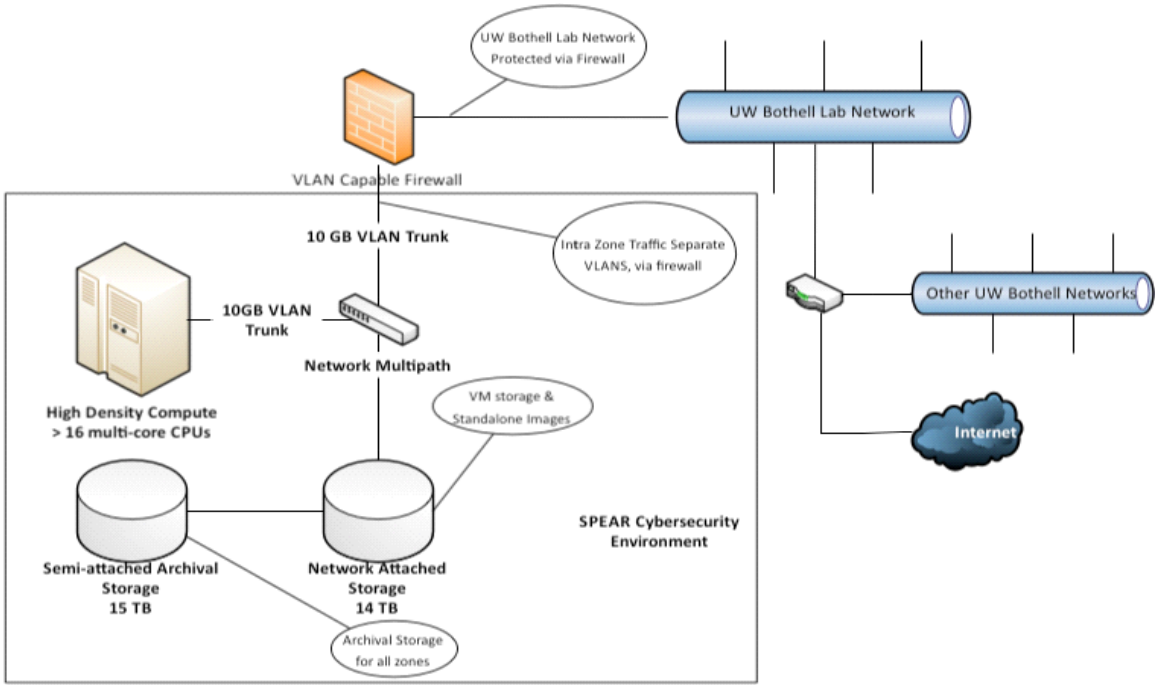

Figure 2: Cloud Architecture

Related works

At the University of Western Australia, they have made a network simulation to help students better understand how computers communicate by showing them the packets as they pass through the different levels of the OSI model, and as they get routed through the network. This program has a descriptive language

that allows the user to specify the network conditions like the cost between nodes, the program processes this input and generates the nodes on different threads. Their goal for this program is to allow the student to have a more graphical view of the network and to allow them to learn without needing to spend a long time setting up a network. (McDonald91)

With the idea of teaching students in a way that would make learning fun and enjoyable while giving them valuable skills in the growing field of cyber security. Students at MIT have developed a game of capture the flag that they use in teaching how cyber-attack and defense works. (Werther11) This game works by having a hidden file on a server and the team has to get in and take it, not only does it teach cyber security but also how to think outside of the box, to look at the system through the eyes of an attacker. While the main goal is to make the learning fun, it's also very important to teach the ethics of hacking to those who you give the skills (Mirkovic14); we need the people who are trying to learn how to hack to understand the potential damage that can be done if misused.

Future Work and Conclusion

We are deploying this tool at the University of Washington Bothell on our local private cloud system and enabling remote access from a laboratory at the Yakama Nation Library. We will be using this laboratory in conjunction with lectures to introduce students to cyber security concepts and help the students develop practical skills.

The goal of this project is to help the students of areas where there is a very low rate of success and the lack of ability to teach cyber security, this tool has many areas to grow and to provide students an area of research. Machine learning is another area, where we can feed in a model of an existing network and have the system learn about that network and prepare a probabilistic model from that. This tool can also be used for capture the flag events to allow students to hone skills of attack and defense, as well as management and response to cyber incidents. The skills they could learn will help them to look at the world with more cautious eyes, able to see and identify the security problems that exist around them, even if they don't become a programmer they will see the problems that can come from a web site that might limit them to an 8 character password or own that doesn't warn them when they type in a week password.

As we develop this system and the educational material, our goal is to make our systems available so that other organizations can use them easily. We are doing this by developing our system so that it can be deployed on low-cost hardware and only require access to low-end computers to access the scenarios. We also are designing it so that it is easy for teachers to use our teaching material and deploy scenarios for their students to work through. These materials could allow colleges to start offering a cyber-security courses for all their CS programs, or high schools could have classes that will give students an idea of what a career would require.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant no. DGE-1419313.

References

McDonald, C. (1991) *A Network Specification Language and Execution Environment for Undergraduate Teaching.* In Proceedings of the Twenty-Second SIGCSE Technical Symposium on Computer Science Education, 25–34. SIGCSE '91. New York, NY, USA: ACM.

Mirkovic, J, and Peterson, P. (2014) *Class Capture-the-Flag Exercises*, Usenix Summit on Gaming, Games, and Gamification in Security Education.

A. Conklin, "Cyber Defense Competitions and Information Security Education: An Active Learning Solution for a

Capstone Course," in Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. HICSS '06, 2006, vol. 9, p. 220b–220b.

Werther, J, Zhivich, M., Leek, T., and Zeldovich, N..(2011) *Experiences in Cyber Security Education: The MIT Lincoln Laboratory Capture-the-Flag Exercise.* In Proceedings of the 4th Conference on Cyber Security Experimentation and Test, 12–12. CSET'11. Berkeley, CA, USA: USENIX Association.

W. I. Bullers Jr., S. Burd, and A. F. Seazzu, "Virtual Machines - an Idea Whose Time Has Returned: Application to Network, Security, and Database Courses," in Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, New York, NY, USA, 2006, pp. 102–106.