Scripting an Automated Score and Message Board; Cybersecurity Competitive Labs as a Service (CLaaS)

Nicholas Hempenius and Te-Shun Chou

Department of Technology Systems
East Carolina University
Greenville, North Carolina 27858
hempeniusn15@students.ecu.edu
chout@ecu.edu

Background

This research discusses an Automated Score and Message Board (ASMB) system used in a Competitive Labs-as-a-Service (CLaaS) education platform. The CLaaS platform provides students with identical simulated networks containing multiple Virtual Machine(VM) servers. The student networks are interconnected in a larger self-contained virtualized environment. In CLaaS Students harden their own VM servers and attack others. The ASMB displays and updates student scores when servers are successfully hardened or attacked. CLaaS encourages a high level of interaction between students and the ASMB facilitates competitive learning among students.

Introduction

This presentation describes the methods and technologies behind the ASMB used in the CLaaS system. The inherent nature of cybersecurity encourages bending rules. This meant security needed to be considered when designing the ASMB. The ASMB was designed as a five step process

1. **Collection:** Where and how to securely collect scoring metrics?
2. **Transportation**: Where and how to securely transport the metrics?
3. **Storage:** Where and how to securely store the metrics?
4. **Analysis:** Where and how to securely analyze the metrics?
5. **Display:** Where and how to display the results?

Collection

The purpose of this step was to create a flow of data to be analyzed. Custom scoring agents were created to run on each VM server. The agents run checks in periodic intervals, collecting information from data points determined by lab objectives and monitor for objective completion. The scripts run only in memory as Terminate Stay Resident programs and are hidden from even privileged users. Figure 1. below is an example of suedo code that might appear in the agent scripts.

```
1  Check-Lab1-Objective1:
2  FIREWALL = (Check if the firweall is on)
3  if (FIREWALL = on)
4   then
5     Lab1-Objective1 = Finished
6     Send check results
7   else
8     Lab1-Objective1 = Not Finished
9     Send check results
10 end
```

Figure 1. Example suedo code. If Lab1-Objective one is "Turn on Firewall." Check if the firewall is turned on.

Transportation

The second step addresses a common tradeoff in cybersecurity between performance and security. Rsyslog w/TLS was chosen as the transport mechanism, it provides the equivalent security of other protocols such as SSH, however it faster, and requires less network bandwidth. For further integrity, the Rsyslog server resides behind a firewall. The Rsyslog traffic is sent over a network separate from where students would perform their lab tasks, as shown in Figure2.

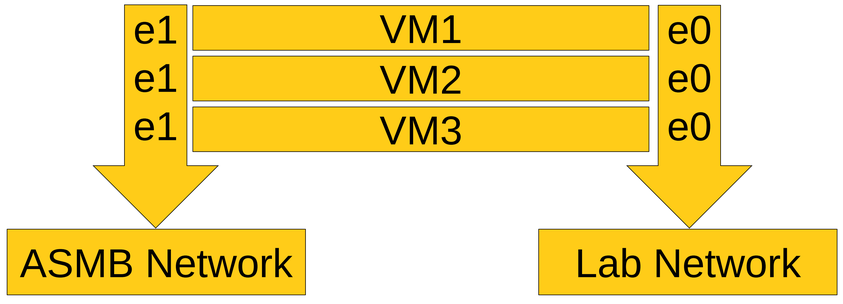

Figure 2. Each VM with a score agent has two network connections, the ASMB network is where score data is transmitted and the Lab network is where Lab tasks are performed.

Storage

In step three, the score data needs to be stored in a secure location, and in a common, format. The score format needed to provide five key pieces of information, the student ID number (SID), lab number(LN), objective number (ON), objective state (OS), and if it is attack or defend data(A/D) data. Figure 3 illustrates our chosen format. For each VM, there is a corresponding log file holding their raw score data. Access controls to the data files provide sufficient data integrity.

```
172.20.2.3:L3_1-0:L3-2_1:L3-3_255:L3-4_255
172.20.2.3:10.10.101.252-L3-1_255:10.10.101.252-L3-2_254:10.10.101.252-L3-3_254:10.10.101.252-L3-4_254
172.20.2.3:10.10.102.252-L3-1_0:10.10.102.252-L3-2_254:10.10.102.252-L3-3_254:10.10.102.252-L3-4_254:
172.20.2.3:10.10.103.252-L3-1_255:10.10.103.252-L3-2_254:10.10.103.252-L3-3_254:10.10.103.252-L3-4_254
172.20.2.3:10.10.104.252-L3-1_255:10.10.104.252-L3-2_254:10.10.104.252-L3-3_254:10.10.104.252-L3-4_254
172.20.2.3:10.10.105.252-L3-1_255:10.10.105.252-L3-2_254:10.10.105.252-L3-3_254:10.10.105.252-L3-4_254
172.20.2.3:10.10.106.252-L3-1_255:10.10.106.252-L3-2_254:10.10.106.252-L3-3_254:10.10.106.252-L3-4_254
172.20.2.3:10.10.107.252-L3-1_255:10.10.107.252-L3-2_254:10.10.107.252-L3-3_254:10.10.107.252-L3-4_254
172.20.2.3:10.10.108.252-L3-1_255:10.10.108.252-L3-2_254:10.10.108.252-L3-3_254:10.10.108.252-L3-4_254
172.20.2.3:10.10.109.252-L3-1_255:10.10.109.252-L3-2_254:10.10.109.252-L3-3_254:10.10.109.252-L3-4_254
172.20.2.3:10.10.110.252-L3-1_255:10.10.110.252-L3-2_254:10.10.110.252-L3-3_254:10.10.110.252-L3-4_254
```

Figure 3. Example of a single round of score data for student2, lab3, objectives1-4.

Analysis

Step four is where the raw score data is analyzed. The analysis happens as data is received by the Rsyslog server and is broken down into two major steps. The data is normalized and combined into a single file. Then, the SID, LN, ON, OS, and (A/D) are extracted, a database of score and message values is referenced, and the corresponding students score and message is updated. Figure 4 and 5 show how the data is interpreted.

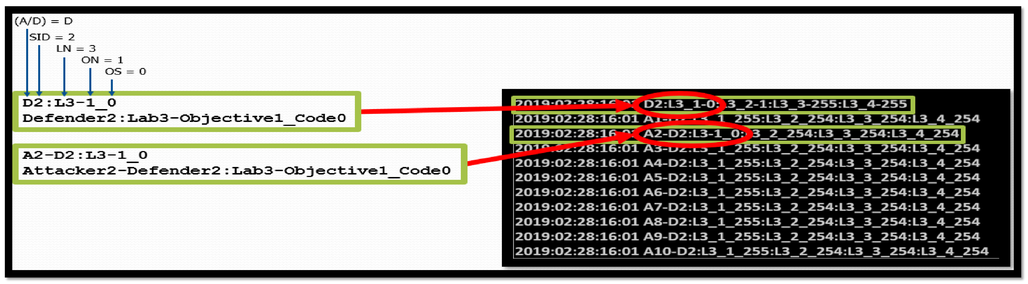

Figure 4. Score data analysis, extracting (A/D), SID, LN, ON, and OS.

Analysis Cont..

```
Student1 completes defender
Lab3 objective 1.
Log = D2:L3-1_0
D2:
L3-1_
0
```

defenderScores

Lab-Obj	0	1
3-1	1000	-1000
3-2	1200	-1200
3-3	1300	-1300
3-4	1400	-1400

D2:L3-1_0 = Student2 gets 1000 Points

```
Student1 completes Lab1
objective 1 against Student2.
Log = A1-D2:L1-1_0
A1-D2:
L1-1_
0
```

attackerScores

Lab-Obj	0	1
A1-1	1000	0
D1-1	-500	0
A1-2	1300	0
D1-2	-650	0

A1-D2:L1-1_0 = Student1 gets 1000 Points and Student2 loses 500

Figure 5. Score data analysis, finding score values with (A/D), SID, LN, ON, and OS values. The same method is used for finding message values.

Display

The final step display would be as simple as populating a table with the students username, score and message values stored in the database. As shown in Figure 6.

Name	Score	Message
Lisa	+9	You attacked Student3's Defender1.
Tim	2	You properly configured Defender2's firewall.
Jessy	-3	Student1 attacked your Defender1.

Figure 6. Example Scoreboard displaying student scores and messages.

Materials and Methods

Using PowerShell and Bash scripts, the agents are light weight and require minimal dependency's making them easy to install on new Lab VMs. Hidden agents, a stringent PFSense firewall, TLS encrypted traffic and a Security Enhanced Linux Rsyslog Server ensure that score data is handled securely at every stage of the process. Python scripts are used to analyze the data. Score and messages definitions and student score and message values are stored in a MongoDB Database, which is easily queried for display by a web application. A MEAN (MongoDB, Express, Angular, Node.js) web stack is used to display the values directly on a Score and Message Board in the CLaaS learning system GUI.

Acknowledgments

This research is made possible by the National Science Foundation grant 1723650. The authors are grateful to the support of the Department of Technology Systems in the College of Engineering an Technology at East Carolina University