Measuring System Utilization During A SlowLoris Laboratory Exercise

Taylor Lynch, Dr. Jason M. Pittman

Introduction

Creating hands-on exercises for cybersecurity education is useful to understand both offensive and defensive tactics. However, laboratories do not consider the system utilization needed to conduct such exercises. Often laboratories underestimate or overestimate the amount of CPU, memory, disk, and network resources that are required to run hands-on labs. As a result, either less students can participate if utilization is underestimated, or it is not cost-effective if resources are overestimated.

Unfortunately, while the literature contains a plethora of research asserting a variety of laboratory infrastructure and content designs, the existing research does not consider the system utilization needed to conduct hands-on learning scenarios.

Method

To address the gap in the literature with respect to providing accurate laboratory utilization information, we conducted an experiment to collect CPU, memory, disk, and network resources during a common cybersecurity education lab exercise. For this work we selected the denial-of-service lab used by Damon et al (2012). We developed two instruments to collect data: a HTTP attack tool based on the SlowLoris vulnerability as well as a program to collect system utilization values during the exercise.

The experimental environment consisted of two virtual machines, both running Ubuntu linux. One system- the attacker- ran the *nycticebus* attack instrument and the *utilization tool zero* (UTZ) data collection instrument. The other system- the target- ran a simple web server and the UTZ instrument. Both systems were allocated a single virtual CPU, a single network interface, 20GB disk, and 4GB RAM.

Procedure

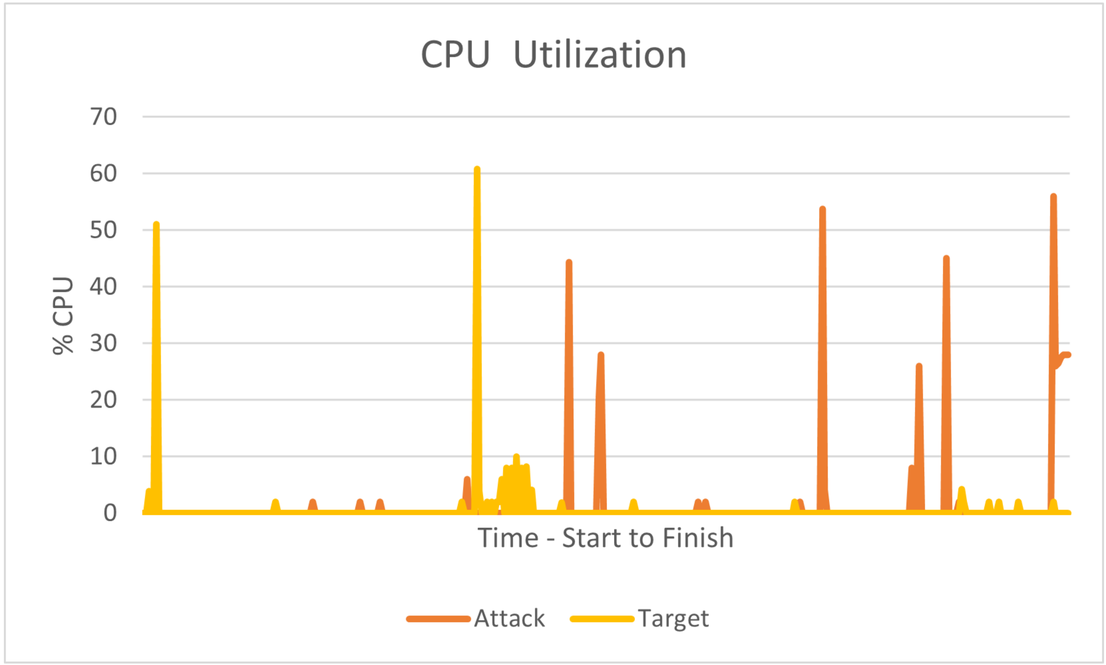
Preparation

1. Power on virtual machines
2. Check network connectivity to have a smooth experiment
3. Check *UTZ* and *nycticebus* configurations
4. Verify the Apache webserver is running

Execution

1. Start the packet captures
2. Start the *utz* instrument
3. Start the attack instrument
4. Confirm the target is unavailable
5. Stop attack
6. Stop *utz*
7. Stop the packet captures
8. Rename all log files

Results



	Min CPU	Max CPU	RAM	Network
Attacker	2%	56%	0.001	16KB
Target	1.90%	60.80%	0.14	18KB

Conclusions

As you can see from the results of the experiment the attack machine use roughly three times as much CPU on average, but the maximum amount on both machines was roughly the same. Meanwhile the target machine used approximately 100x more RAM. This means that the attack machine requires less RAM than the target machine, but the target machine needs just as much or even slightly more CPU. The network traffic was similar between the machines with about 17Kbytes of data sent/received, this means if you have 20Kbytes of data for this experiment then 3Kbytes of data are not being used.

Recommendations

As a next step similar research is necessary for the most common popular types of experiments done in cybersecurity education laboratories. Such work might be beneficial by adding to the body of cybersecurity education literature the optimal resources needed to conduct these hands-on exercises. Consequently, this will create a more accurate baseline for labs so there is less waste of infrastructure resources and could possibly expand the scale of learning for students. Logically, another next step might be to create an inventory of sorts to rank order laboratory exercises evidenced in existing literature.

References

Research funded, in part, by the URCW Office at High Point University

URCW

High Point University