BEACON Labs: Designing Hands-on Lab Modules with Adversarial Thinking for Cybersecurity Education

Jordan Whyte
Computer Science Department
Boise State University
jordanwhyte@u.boisestate.edu
0000-0003-0663-4584

Gaby G. Dagher
Computer Science Department
Boise State University
gabydagher@boisestate.edu
0000-0001-7837-182X

Sara Hagenah
Curriculum, Instruction, and Foundational Studies
Boise State University
sarahagenah@boisestate.edu
0000-0002-9892-8653

***Abstract*—Cybersecurity is an interdisciplinary field that is concerned with protecting digital assets from cyber-attacks aiming to illegally access sensitive information in order to tamper and disrupt systems and processes. Producing cybersecurity materials that are vertically-aligned is highly desired, given the shortage of cybersecurity educators and the dynamic and evolving nature of cybersecurity. More specifically, universities must do more to help fill the huge cybersecurity workforce shortage and address the lack of materials centered around adversarial thinking. In this paper, we propose a four-step process to turn a recent cybersecurity paper into a hands-on lab that utilizes game theory to promote adversarial thinking and show a case study where this process was used. The four-step process explains how papers are chosen, their research replicated, the production of lab materials, and complementary materials for students to work from. The case study demonstrates this process in practice and explains how game theory is incorporated into the lab.**

Keywords—cybersecurity education, game theory, adversarial thinking

I. INTRODUCTION

The need for cybersecurity experts has been growing fast in the U.S. job market [1] [2]. There is a lack in hands-on materials consistent with industry standards for both teachers and students. Designing labs with the goal of having students carry out the attack instead of just reading about the attack provides students with authentic experience that are responsive to current industry needs and sets them up to be responsive to future cybersecurity job needs.

Cybersecurity labs allow students to learn about vulnerabilities in systems that can be exploited. By practicing to think like someone trying to attack a computer or network system, students can then learn to defend against such attacks. This is referred to as adversarial thinking. Adversarial thinking has been previously explored through students completing labs in roles as the attacker or the defender. While prior research [3] has explored the use of game theory as a method to teach adversarial thinking, the implementation of it into a cybersecurity lab has not been explored.

In this paper, we introduce BEACON Labs (cyBersEcurity eduCAtiON labs), a set of hands-on cybersecurity labs that lead students through the process of exploiting a vulnerability and challenging them to think critically, while utilizing game theory to teach them about adversarial thinking. Our approach allows students to recognize adversarial thinking working on a smaller scale and then translate it to the larger implications of the lab as a whole. Additionally, we hope the proposed framework for creating labs that include game theory will be used by other instructors to create more hands-on cybersecurity labs. We will make the BEACON materials publicly available through a dedicated webpage for easy access and download. We will also work with the Cybersecurity Labs and Resource Knowledgebase (CLARK) project team to make our materials available to the public through the CLARK digital library.

The BEACON labs are constructed based on vulnerabilities or defense against vulnerabilities that were published within the last five years. By using current papers we are able to construct labs that are from recent research and not outdated, which gives the students applicable experience when they enter the work force.

This paper aims to demonstrate the methodology of creating a cybersecurity BEACON lab by first finding modern papers with recent vulnerabilities, completing the attack or defense discussed in said paper, then reducing each part to meaningful tasks, one of which will have an emphasis on game theory to teach students adversarial thinking. This paper will then explore an example of a lab we have constructed recently and how our methodology was followed to see its completion.

II. RELATED WORK

A Suite of Instructional Laboratories for Computer SEcurity EDucation (SEED) is a common resource used for teaching hands-on cybersecurity [4]. SEED Labs allow for students to learn about certain attacks and vulnerabilities within computer systems or networks. The SEED labs teach adversarial thinking through having students complete attacks in a controlled environment as an attacker. However, the concept of teaching adversarial thinking through game theory is not used. Through the construction of our hands-on labs, we aim to teach similarly to the SEED labs, but by using vulnerabilities that have been discovered in recently published papers.

Similar projects such as the National Initiative for Cybersecurity Education (NICE) challenge create real-world scenarios that teach students to handle issues that could arise within a company and how to solve them [5]. We have taken influence from this approach within the game theory part of our labs by designing real-world scenarios students will have to analyze. However, the NICE challenge's more technical approach is substituted for students predicting how an attacker would act via game theory.

Additionally, the NICE challenge presents the NICE framework. The framework describes three building blocks to be used as guidelines to enhance comprehension. By creating statements about the knowledge and skills required to complete a task, it is clear for both a student and instructor what must be accomplished for success [6]. We accomplish this through our learning objectives associated with each lab created. By requiring students to be knowledgeable about the paper's research, we are basing the lab upon and develop the skills to complete the research outlined, they will be able to complete the lab. We outline the knowledge and skills the students will use or learn at the beginning of the lab so both instructors and students comprehend what the outcome should be.

Adversarial thinking is often referred to as the idea of "how to think like a hacker." By teaching students to think like an adversary through carrying attacks assists them in preventing vulnerabilities in their future code. The application of game theory to teach adversarial thinking has not been fully explored, but prior research has been done that argues the development of strategic reasoning to predict an adversary's actions can be achieved through basic game theory concepts [3].

III. METHODOLOGY

In this section, we explain our approach for creating our labs through a four-step process, as shown in Fig. 1.

A. *Step 1: Identify Research Paper*

When constructing a BEACON lab, we first start by finding a recent security research paper that discusses a new vulnerability, or a new defense against a previously known vulnerability. We want to use recently found exploits so that students are learning where the field of cybersecurity currently is and where it is going. The research paper must be modern enough to be relevant to students, but not be too advanced that it requires every term the paper references to be explained. Striking this balance between the two is integral to constructing a lab that will hold the student's attention and be interesting to complete. Additionally, the paper must use software that is open source and can run on a virtual machine. When having students complete the lab we will present them with a lab document and a virtual machine that will contain the required tools to complete it, thus using software that is accessible to all is imperative.

B. *Step 2: Replicate Attack/Mitigation*

Once a paper is selected, we will replicate the actions taken by its writers. Our selection process requires the software used in the paper is open source, thus allowing us to complete the attack or defense previously done.

The attack (or defense) discussed in the research paper is usually executed using an Ubuntu virtual machine. The software and process of completing the attack is taken note of for when the lab document is created in step 3. If there is code required, that is saved for step 3 as well. Skeletons of the code used can be included for the students to complete without giving them the finished product. The code will have specific pieces removed so students will have to learn about the attack or defense to complete that task.

If replicating the research proves to be unattainable, we contact the researchers to ask how they completed certain steps. If even with their guidance we are still unable to complete what was done in the paper, we return to step one and find another paper.

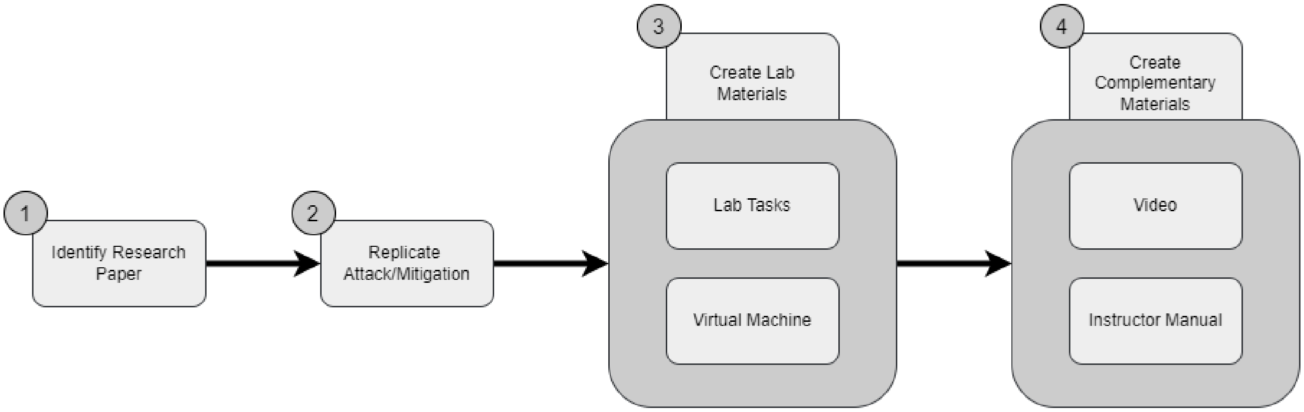

Fig. 1. Major steps for BEACON labs design

C. Step 3: Create Lab Materials

After the paper's research has been replicated, we begin to construct a lab guide for the students. This requires parsing each part of the attack or mitigation into steps for students to carry out. Additionally, a virtual machine with all required software and relevant files must be made. By giving students a lab guide and a virtual machine with all required software to complete the lab downloaded, students only need the two files to start working.

1) Lab Tasks: The process of completing the attack or defense detailed in the paper must be parsed into individual tasks for the students to complete. The labs are designed with adversarial thinking in mind. We aim to have the students try to execute the attack from the attacker's perspective. If the research paper we are basing the lab off of includes a defense against this attack, the students will implement the attack and then the defense.

Our goal is to create tasks that require the students to think critically about what the next step is and not to just explain what they must do to complete the lab. A balance must be struck between explaining the concepts students are unaware of without just giving answers to each step. With this balance, students will be able to learn new cybersecurity concepts but still problem-solve parts of the lab themselves.

If there is a small number of actions taken as discovered in step 2, additional cybersecurity concepts that are relevant to the paper can be made into tasks such as internet transmission protocols.

Once the lab document has been complete, we can look to include game theory into one of the tasks. This could either be weaving it into one of the tasks to lead students to the correct answer if they solve the game theory table correctly, or setting up a scenario where the attack or defense would be used in the real world. The example shown in this paper uses a game theory table that demonstrates how an attacker and a defender (server) would have to think if this attack were to be used. BEACON labs in general use different concepts of game theory. The type of game theory used in the example in this paper is backwards induction and imperfect information. Completion of the game theory table correctly is required for students to gain full points on the lab. Since adversarial thinking is integral to solving the game theory table students are required to think how an attacker would to succeed.

Students will submit a lab report to their instructor when finished with the lab. Each task reminds them to document what actions they took and to detail their thinking. Screenshots of the task being carried out are encouraged for students to include.

2) Virtual Machine: The virtual machines we construct for the labs are always an Ubuntu distribution. Making a specific virtual machine for each lab ensures that the vulnerability used in the lab is not patched. Additionally, it ensures that all students are working in the same environment for easier troubleshooting.

All software used to complete the attack or mitigation are downloaded and configured on the virtual machine. This is beneficial to the students so they do not have to find it online or spend time setting it up.

D. Step 4: Create Complementary Materials

When a lab is complete and ready for students to use, complementary materials will then be created. This consists of creating a video and instructor manual. The video will serve to teach students about the game theory step or a hard part of the lab, as well as cater to visual learners. The instructor's manual is self explanatory as it serves to teach instructors about what the lab comprises.

1) Videos: Every lab has an accompanying video. These videos are five to ten minutes long and explain either what we believe is the most difficult part of the lab, or the game theory portion. Since game theory is often a new concept to students studying in the cybersecurity field, an explanation of how to complete that task is beneficial. Alternatively, if a step of the lab is more difficult or harder to understand than others, a video will explain the process to complete it without giving students a direct answer to the problem. We may create more than one video for a lab if necessary.

2) Instructor's Manual: An instructor's manual is also created after the video. This manual will explain what the students will do at each step and what problems they may encounter. It will also explain to instructors how to grade each section through criteria of what makes an answer correct. We hope that by creating a document detailing the lab from the perspective of teaching it, instructors will better enable their students' success.

IV. CASE STUDY: EVILSCOUT LAB

EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi was published in March of 2020 in Institute of Electrical and Electronics Engineer's (IEEE) Transactions on Network and Service Management journal [7]. By creating a legitimate connection to an access point, an attacker can learn the mac address and name of the network. With this information they can create a duplicate network to capture traffic known as an Evil Twin. To a client the difference between the legitimate access point and the Evil Twin is imperceptible, thus allowing for the attacker to capture sensitive data. We used this paper to create a lab through the process detailed above.

A. Identify Research Paper

We gathered papers from reputable journals that have been released recently. Papers from UNIX and Advanced Computing Systems Professional and Technical Association (USENIX) and IEEE are top contenders for our selection, but papers from other popular journals were gathered as well.

After narrowing the quantity of papers to two, we had to decide between BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy Networks from USENIX [8], and EvilScout from IEEE. EvilScout was chosen after deliberation as the tools to implement it were explored in a physical and virtual environment. The paper detailed its steps to use the program Mininet-wifi to implement EvilScout in addition to a physical network. Since it was already implemented in a virtual space for the paper, we were sure that our implementation and creation of the lab would be possible. BlueShield however was only implemented physically. While there are likely tools available that would allow for its implementation virtually, EvilScout gave us the liberty of knowing what we were doing had been done before. Additionally, this would allow us to contact the authors with specific questions about replicating their process.

B. Replicate EvilScout Mitigation

Replicating the paper's research was trivial as the only software that was used was Mininet-wifi. Using this program, we could create a virtual network topology complete with a controller, access points, and users. The topology used within Mininet-wifi was detailed in the paper along with the pseudocode of EvilScout, allowing us to implement a similar but simpler topology to test upon.

By having a user take the access point's name and hardware address and then broadcast that at a higher range, we had successfully created an Evil Twin. Then, by implementing the pseudocode in python and analyzing packets sent wirelessly on all channels with scapy, a popular packet manipulation tool for python, we had successfully implemented EvilScout.

C. Create EvilScout Lab materials

We created four tasks for the EvilScout lab. Task One: Creating the Network with Mininet-wifi, Task Two: Evil Twin, Task Three: EvilScout, and Task Four: Adversarial Thinking in Games With Imperfect Information. This section will focus on the creation of the second and the fourth tasks.

Once all tasks are created, we can finalize the virtual machine that will be given to students. By creating the lab tasks first, we can take an inventory of all necessary files or software that the virtual machine should contain.

1) Lab Tasks: The first task has the students create a network topology, including a controller, switch, host, and access point. Additionally, two legitimate clients that are attempting to use the network are present with a third client that will act as the attacker. With this topology, students can move to step two where they will set up the attacker as an Evil Twin.

Before having the students alter the attacker to be an Evil Twin, we explain some useful commands within Mininet-wifi such as opening a new terminal for running commands on specific nodes in the topology. Giving students instructions on how to use Mininet-wifi instead of having them figure it out for themselves is necessary as it is a program they are unlikely to be familiar with.

To create an Evil Twin, students must have the attacker copy the name of the network and it's hardware address. In the task, we explain to the students how to scan for networks over the terminal to see these specifications, but we do not give the commands for how to change the attacker's network interface. This is done to incentivize students to seek how to do this on their own.

Students are then asked to scan for available networks with another client while the Evil Twin is running on the same channel as the legitimate access point and while it is on a different channel. This has the students go through the steps of setting up the Evil Twin again with different specifications. Clients scanning for available networks while the Evil Twin is on the same channel should yield only one network to connect to, and if it runs on a different channel, two networks will be available. We expect students to figure out this difference for themselves and include why this happens in their lab report. The second task finishes by having students connect a legitimate client to the Evil Twin to setup for the next task where EvilScout is implemented. The second task walks them through what they should do but still requires students to figure out what to do on their own as well as create their own conclusions to include in the lab report.

The fourth task has students answering questions about the game theory table in Fig. 2. The explanation for this table included in the lab guide proposes a scenario where a server exists that is vulnerable to the Evil Twin attack. Player one is the attacker and player two is the defender (server). The attacker must make two decisions before the server is able to act: choosing to attack or do nothing and completing a long attack versus a short attack. The dotted line indicates that the server does not know what actions the attacker will take, thus they must make their decision to implement EvilScout or do nothing with the imperfect information of just viewing the game theory table. The numbers at the end of each line indicate the utility that is gained by each party if the game were to end in that scenario with the bottom number corresponding to the server and the top number corresponding to the attacker.

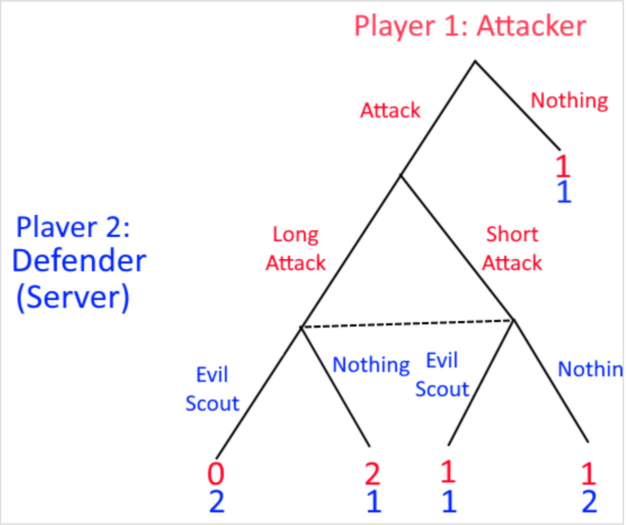

Fig. 2. Game theory table presented in the fourth task

TABLE I. EVILSCOUT LAB TASK 2 RUBRIC

	Full Credit	**Partial Credit**	**No Credit**
Task 2	Correct list of commands used to change network name and hardware address	Some of the correct commands are listed for changing network name and hardware address	No correct commands are listed
	Show the evil twin running on both: the same channel as the legitimate access point and a different channel	Show the evil twin either running on the same channel or on a different channel	Incorrect or not enough information showing evil twin on same or different channel
	An image showing that a client is connected to the evil twin via tcpdump	N/A	No image is showing that the client is connected to the evil twin is provided

Students are asked the following questions:

1. Should player one always choose to attack? Why?
2. When attacking, should a short or long attack be used?
3. Knowing that the server is unaware of the attack being short or long, should player 2 implement EvilScout or do nothing?

Just viewing the table may not be sufficient to answer the questions. Therefore we provide an accompanying video explaining backwards induction, a method of solving game theory tables, in the complementary materials.

2) Virtual Machine: Once the tasks are written we know what must be included on the virtual machine given to students. In this case, only Mininet-wifi and a code skeleton used in task three must be included. The code skeleton requires students to fill in certain parts of a script that will analyze packets sent over a wireless connection and is used in the third lab task.

Some labs will see the students using multiple software, but in this example, only one is required.

D. Create EvilScout Complementary Materials

The EvilScout lab included a type of game theory that students are likely not familiar with. We seek to leverage this to have them think critically to solve it. The video accompanying this lab clues them into how they should think to solve it without giving the answer away directly.

The instructor's manual explains some hard parts of the lab that instructors should know to assist them in teaching students as well as using a rubric.

1) Video: The video created for this lab explains the concept of backwards induction. It uses a game theory table different to that used in the lab with altered utility values for each player to earn and a more linear style. The video explains how only by working backwards one could figure out what the outcome of the table would be.

Using backwards induction can assist the students in the lab by showing them that working backwards can help in solving the table used in the lab.

Since the game theory table used in the lab is unlikely to have been seen before by students, we felt a video explaining something that could help in solving it would be beneficial for those completing the lab.

2) Instructor's Manual: The instructor's manual is similar to the lab guides, but includes additional notes in the margins or additional sections that explain either solutions or how to teach certain parts of the lab. The second task for example includes the specific commands needed to alter the name of a network and the hardware address. Students are expected to figure this out on their own, but instructors should have access to what is required of the students to be able to push them in the correct direction. Additionally, it includes how having the Evil Twin on the same channel versus a different channel impacts a client viewing local networks.

The instructor's manual also includes a rubric for each task. This can be used as a guide for instructors to know what they are looking for on each task to assess its completeness. It is a suggestion for how grading should be done, but we emphasize that it should be up to the instructor's discretion. The rubric for the second lab task is in Table I.

V. CONCLUSION AND FUTURE WORK

In this paper, we describe a process to create hands-on cybersecurity lab materials with adversarial thinking using game theory. We propose a four-step process: (1) identify a research paper, (2) replicate attack/mitigation, (3) create lab materials, and (4) create complementary materials. Using this process, we show a case study of one of the BEACON labs created and how game theory is used to emphasize adversarial thinking. As a future work, we plan to integrate our lab materials into a cybersecurity curriculum and implement them in a classroom environment to assess the effectiveness concerning adversarial thinking.

REFERENCES

[1] K. O'Hara, "The future of cybersecurity jobs," *Monster*.

[2] J. Kauflin, "The fast-growing job with a huge skills gap: Cyber security," *Forbes*.

[3] S. T. Hamman, K. M. Hopkinson, R. L. Markham, A. M. Chaplik, and G. E. Metzler, "Teaching game theory to improve adversarial thinking in cybersecurity students," *IEEE Transactions on Education*, vol. 60, pp. 205–211, 2017. [Online]. Available: https://doi.org/10.1109/TE.2016.2636125

[4] W. Du, "Seed: Hands-on lab exercises for computer security education," *IEEE Security Privacy*, vol. 9, pp. 70–73, 2011. [Online]. Available: https://doi.org/10.1109/MSP.2011.139

[5] V. Nestler, T. Coulson, and J. D. Ashley, "The nice challenge project: Providing workforce experience before the workforce," *IEEE Security Privacy*, vol. 17, pp. 73–78, 2019. [Online]. Available: https://doi.org/10.1109/MSEC.2018.2888784

[6] R. Petersen, D. Santos, M. C. Smith, K. A. Wetzel, and G. Witte, "Workforce framework for cybersecurity (nice framework)," *NIST Special Publication*, 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-181r1

[7] P. Shrivastava, M. S. Jamal, and K. Kataoka, "Evilscout: Detection and mitigation of evil twin attack in sdn enabled wifi," *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, vol. 17, pp. 89–102, 2020. [Online]. Available: https://doi.org/10.1109/TNSM.2020.2972774

[8] J. Wu, Y. Nan, V. Kumar, M. Payer, and D. Xu, "Blueshield: Detecting spoofing attacks in bluetooth low energy networks," *USENIX International Symposium on Research in Attacks, Intrusions and Defenses*, vol. 23, pp. 397–411, 2020.