GenCyber 2021 Student Manual

Mckenzie Mack

Time	Day 1	Day 2	Day 3	Day 4	Day 5
8-8:50	Welcome, Introductions, Pre-survey, Icebreaker – "Fact or Fiction"– Morgan	Game/Activity – Cyber ball – Pair 2 teams – Wilson / Morgan	Game/Activity Poor Me – Pair 2 teams – Wilson / Morgan	Game/Activity Spaghetti Tower – Wilson / Morgan	Game/Activity Cyber Bingo – Pair 2 teams – Wilson / Morgan
9-9:50	L – Intro to Cyber Security - Chang A – Trashketball - Mckenzie	L – OS Overview - David A – Terminal Commands Resource Utilization - Ally	L – Applications, Password Cracker - Chang A – Profile, Crack Password - Kirsten	L – Databases, CIA - Xie A – Install / Configure MySQL - Kirsten	L – Review Security Layers, CIA, F10P Relationships - Chang
10-10:50	L – Physical Security - Chang A – Scavenger Hunt - Ally	L – NMAP Network Scanner - David A -Practice NMAP - Ally	L – Basic Programming Concepts - Xie A – Write Simple Program - Mckenzie	L – Create Simple DB, Python Program to Connect - David A – Create DB Tables, Write Program to Read/Write Data - Ally	Project Activity – Finish Presentation
11-11:50	L – Explore RPi - Chang A – Assemble Raspberry Pi - Kirsten	Game/Activity - Spheros – Wilson	Mechatronics Lab Tour	High Performance Computing Lab Tour	Speaker Dr. Campbell –Women in Cybersecurity
12-1	Lunch	Lunch	Lunch Panel Discussion	Lunch Student Outreach	Lunch w/ Parents UTC Career Counseling
1-1:50	L – Network topology, infrastructure - Xie A – Paper Airplanes Message Routing - Mckenzie	L – Wireshark Packet Analyzer - Xie A – Practice Wireshark - Kirsten	L – Python Programming, Cryptography / Encryption - David A – Write Caesar Cipher - Ally	L – Attach Camera to RPi, OpenCV - David A – Capture Image, Apply Filters, Security - Mckenzie	Project Presentation – 15 minutes per team
2-2:50	L – Terminal commands - Xie A – Explore the network and find classmates - Ally	Game/Activity – Spheros – Wilson	Game/Activity – Spheros – Wilson	Game/Activity – Spheros – Wilson	Project Presentation – 15 minutes per team
3-3:50	Introduction to Project – Act it out, Tic Tok video,	Game/Activity – Spheros – Wilson	Game/Activity – Spheros – Wilson	Game/Activity – Spheros – Wilson	Post-survey, Wrap-up

	Poster – Wilson/Morgan				
4-4:50	Project – Activity checkpoints Wilson/Morgan	Project – Activity checkpoints Wilson/Morgan	Project – Activity checkpoints Wilson/Morgan	Project – Activity checkpoints Wilson/Morgan	

Table of Contents

Day 1

On Day 1, you will kick off the camp by getting to know a little bit about the other students that will join you on your GenCyber journey. This will include some introductions along with forming groups of four. You will work with the same group throughout the week. Once you have your group, you and your fellow group members can play a game of **Icebreaker - Fact or Fiction** to see how much you really know about each other.

After that, you will move on to the first lesson of the day - **Intro to Cybersecurity**. In this lesson, you will take a quick look at the five layers of security along with some key principles of cybersecurity that you are going to revisit in each of the layers throughout the camp. If you can't remember all of the terms, don't worry, you will get a chance to go back over them as you try to win a game of **Trashketball**.

In the second lesson, **Physical Security**, you will take a closer look at cybersecurity in the physical layer. Then, you will work with your team to spot some physical security measures in the world around you in the **Physical Security Scavenger Hunt**.

Next, you get to use some of that awesome equipment that you were given! First, you will learn a little bit about the Raspberry Pi in the **Raspberry Pi** lesson. Next comes the fun part – setting up your Raspberry Pi in the **Raspberry Pi Setup** activity. After this activity, your little computer will be ready to go!

After lunch, you will explore cybersecurity at the next layer in **Communication in the Network Layer**. Once you have the basics of a network down, you and your fellow classmates can build your own "network" to send messages in **Paper Airplanes Message Routing**.

You will continue to explore the network layer in the last lesson of the day, **Using Terminal Commands in the Network**. In this lesson, you will learn some commands that you can use to find out more about the network that your Pi is connected to. Then, you will get a chance to actually use these commands in **Exploring the Network: Find Your Classmates**.

Finally, you will end the day by starting your group project. This is how you will end each day besides the last day of the camp.

Ice Breaker - Fact or Fiction?

Learning Objectives/Outcomes

- Students and staff will get to know one another playing a fun entertaining game of fact or fiction.

Game / Activity Length of Completion: maximum of an hour

Game / Activity Mode: Group

Game / Activity Description

Students and staff write two true and one false statements about themselves. Others in the group will have to guess which is the false statement.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

You will write three statements on your index card with two being true and one false that help us get to know you. After everyone has completed their card, each member will take turns to introduce themselves and read their statements. The other participants will guess which statement is false with a poll raising their hands for #1, #2, or #3.

Trashketball and the First Ten Principles

Learning Objectives/Outcomes

- Describe the ten principles of cybersecurity
- Explore how people are the weakest link in any security system

Activity Length of Completion: ~25 mins.

Activity Mode: Group

Activity Description

Students will demonstrate their understanding of the F10P through a competitive game of Trashketball.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. In this activity, you will review the first ten principles of cybersecurity through a game of Trashketball! Here are the rules:
    a. The first team will be tossed a ball and read the definition of one of the principles of cybersecurity. It is their job to guess which term the definition belongs to. The person who catches the ball is in charge of answering for their team, but they can consult other team members if they do not know the answer. Each team will have 15 seconds to answer.
    b. If at any point a player catches the ball and they have already been a spokesperson, they should give the ball to another person on the team that has not gotten a chance to answer. Once everyone has guessed, it does not matter who answers.
    c. If the team guesses the term correctly, they get 1 point. The person who answered for the team then has a chance to throw the ball into the basket. If the ball makes it in the basket, the team gets another point. The spokesperson then tosses the ball to the other team.
    d. If the ball misses the basket, the spokesperson still throws it to the other team. The process is then repeated.

e. If the team guesses incorrectly, the other team has a chance to steal. The ball should be tossed to the other team and whoever catches the ball acts as the team's spokesperson. If the second team also guesses incorrectly, no points are awarded. The instructor reveals the term and the game continues with the next definition.
f. The game continues until all terms have been guessed. Whichever team has the most points at the end wins and each member of the winning team will receive 4 points.

2. When the game is finished, discuss the following questions with your pair of teams:
   a. Do you see any of the ten principles as more important than the others? If so, why?
   b. Why is it important for each member of your team to understand the ten principles?

Physical Security Scavenger Hunt

Learning Objectives/Outcomes

- Familiarize yourself with physical security countermeasures
- Identify physical security compliance issues
- Find solutions for physical security compliance issues

Activity Length of Completion: ~30 mins.

Activity Mode: Group

Activity Description

During this activity you will get more familiar with physical security countermeasures through a scavenger hunt with your group of 4 and discuss physical security with everyone.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. Stay in your groups - everyone needs their own pencil and paper (you may use this paper).
2. There will be various tags on different physical security components placed around the designated area. There will be 8 tags that will have a symbol and a number.
3. Find these tags and write down the following information next to the corresponding symbol:
    a. The number
    b. The physical security component

    Then take a selfie with all of the team members and the tag! We will collect these later.
4. As you search, identify any physical security compliance issues and write them down.
5. Once you have found all of the elements or once 15 minutes has elapsed (late return will result in disqualification), return to the room and upload all of your photos via the provided Google Form. Time stops once your submission is complete. Points will be awarded to the three teams who found and identified the most symbols the fastest.
6. Join for discussion:
    a. What answers did you get for the physical security tags?

b. What physical security compliance issues did you find?
c. How could these physical security compliance issues be resolved?

Symbol	Riddle	Number	Component
	You can find me on a door Or a chest of hidden treasure If you hear a click when you turn me It certainly brings you pleasure		
	Sometimes to get inside a door You can give a couple knocks Sometimes you will swipe a card So the door you need unlocks		
	You can find this in your home Or really everywhere It sometimes has a lock But it's not a piece of hair It is often made of wood Although it is not a chair A closet sometimes has one But it's not something you wear		
	You may see a blaze And you will know to go But if you do not see it I will let you know		
	This might be open when it's hot And be closed if it has rained If you look on some old churches This has glass that has been stained		
	I cannot see the future I hang on to the past You can view me in the present If you find the display fast		
	Stop, drop, and roll If you ever catch alight If the fire isn't on you Use me to put up a fight		
	I'm a common building item As I am found in every room Flick a switch, my filament glows I dispel darkness and gloom		

Raspberry Pi Setup

Learning Objectives/Outcomes

- Assemble Raspberry Pi
- Install Raspberry Pi OS

Activity Length of Completion: ~25 mins.

Activity Mode: Individual

Activity Description

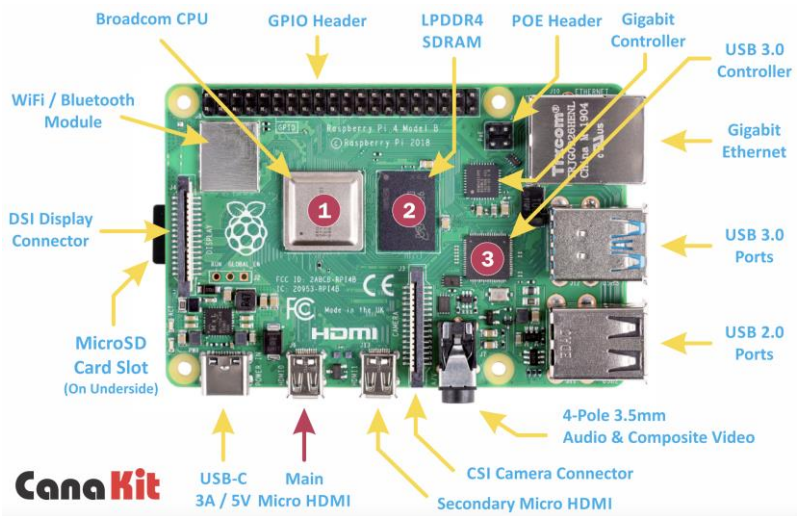
In this activity you will assemble your Raspberry Pi using the provided CanaKit and install the OS.

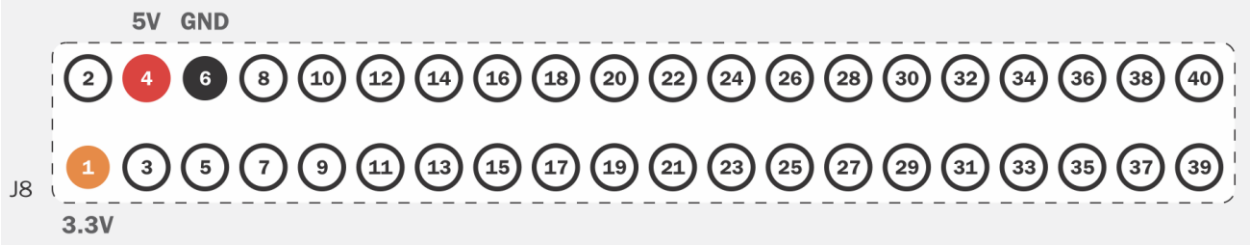
Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. Open your CanaKit and lay out all of the components.
2. Locate the Raspberry Pi and the heat sinks (these are small ridged aluminum pieces). Peel the blue plastic layer off of the glue on the heat sinks and stick the large square one to the (1) spot, stick the rectangular one to the (2) spot, and stick the small square one to the (3) spot.





3. Locate the CanaKit case and disassemble the 3 parts so that you have the bottom, sides, and top.
4. Set the Raspberry Pi into the base of the case and then put the sides onto the base.
5. Locate the CanaKit fan and clip it into the inside of the lid of the case with the label facing towards you. Connect the red and black wires to the corresponding spots on the GPIO Header. You can now snap the lid on.
6. Locate the MicroSD. This MicroSD is preloaded with NOOBS version 3.1.0 or later. Insert this MicroSD card into the MicroSD Card slot on the bottom side of the Raspberry Pi. Be gentle yet firm and make sure that you have the card oriented properly (label facing down).
7. Connect the keyboard, mouse, and monitor to the Raspberry Pi. Attach the PiSwitch to the power supply.
8. After all connections have been made, plug in the power cable.
9. Follow all of the on-screen instructions. The download will take about 15-20 minutes.
    a. Hint: The default username is “pi” and the password is “raspberry”.
10. Connect to the class network. Credentials will be provided.
11. Once configuration is complete, reboot your Raspberry Pi.
12. When your Raspberry Pi reboots, open a terminal window and execute the following commands:
    a. $ **sudo apt update**
    b. $ **sudo apt upgrade**
    c. $ **sudo apt autoremove**

d. $ **sudo apt-get install cmake**

13. You are now ready to go!

Paper Airplanes Message Routing

Learning Objectives/Outcomes

- Discuss how communication works in the Network Layer
- Describe the role of protocols in communication
- Explore the functions of routers and switches

Activity Length of Completion: ~30 mins.

Activity Mode: Group

Activity Description

Through this activity, students will use paper airplanes to simulate how packets travel across a network, as well as how routers and packet switches direct a message on its path.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Process Isolation, Resource Encapsulation, Simplicity, Modularization

Activity Instructions

1. Place the source address given to you somewhere where the other campers can easily see it.
2. On one of the paper airplanes, write down your address as the source address but do not write a destination address. Add a short message as well. While you wait, think of another student that you would like to send the airplane to.
3. Then, on one of your other paper airplanes, write down your address as the source address and another camper's address as the destination address of the airplane. Be sure to label which is the source and which is the destination. After that, add a short message to the airplane.
4. Repeat step 4 for your third airplane.
5. When everyone is finished with their airplanes, gather in a circle. Start with the first airplane that you filled out (the one without a destination address). Try your best to throw your paper airplanes to its intended destination. If the plane does not reach its destination, try to work with the other students to route it to the right person.
6. Once everyone's first plane has reached its destination, repeat step 5 for your second and third airplanes. Wai
7. When all the airplanes have finished flying and have reached their targets, answer the following questions:

a. How was the information written on the paper airplanes valuable in the activity? What role would this information play in network communication?
b. Did all of the planes reach their intended destination without any assistance from other campers? If not, how does the help of your fellow campers to route the message to its destination relate to what was discussed in the lesson?
c. Could you identify some security concerns within the activity?
d. What security measures could you implement to ensure that the paper airplanes maintain a high level of confidentiality, integrity, and availability along their route?

Exploring the Network: Find Your Classmates

Learning Objectives/Outcomes

Explore how commands are used to communicate and identify nodes on the network.

Activity Length of Completion: ~25 mins.

Activity Mode: Individual

Activity Description

Through this activity, students will use their knowledge of network commands to explore their local network and to try to find other classmates.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Process Isolation, Resource Encapsulation, Simplicity, Modularization

Activity Instructions

Beginner:

1. Log in to your Raspberry Pi and open a new Terminal window.
2. Identify the IP address of your raspberry pi using **ifconfig** and record it in the space below.
   My IP address: _____
3. Next, try to guess the IP address of your other classmates using **ping -c 4** followed by the target IP address. To do this, only alter the last digit in the IP address that you found above (For example, if your IP address is 192.168.1.27, change the 27 to another number between 1 and 254). If your ping is successful, log the IP address in the space below.
   Other IP addresses: _____
   _____
4. Repeat Step 3 for around 10 minutes or until you have identified a few candidate IP addresses for your classmates' Pis.
5. Consult with your other classmates to see if you successfully identified any of their IP addresses.
6. Spend the next few minutes answering the following questions:
   a. How did you use the results of the ping to verify that a certain IP address was inactive?
   b. Were you able to successfully ping other IP addresses on the network that did not belong to one of your classmates? If so, what devices might these other IP addresses belong to?

c. What is the importance of knowing another device's IP address?
d. Why might a network administrator want to keep the IP addresses of devices on the network private?
e. Discuss your answers to the above questions with your classmates.

Intermediate:

1. Safiya just got a job as a member of the IT team here and her boss asked her to log some information about the two networks that you are using to store in their records. Unfortunately, Safiya was so nervous about starting her new job that she mixed up her notes and can't remember which characteristics belong to which network!

Characteristics:

- This network has the higher link quality of the two networks (bit rate: _____)
- The access point for this network is F4:2E:7F:03:B0:54
- The IP address of 10.199.10.203 could be found on this network
- More Rx packets have been sent across this network (# of Rx packets: _____)
- The IP address 10.199.11.215 could be found on this network
- The access point for this network is F4:2E:7F:03:B0:53
- This network has the lower link quality of the two networks (bit rate: _____)
- The broadcast address for this network is 10.199.10.255
- The IP address 10.199.11.227 could be found on this network
- More Tx packets have been sent across this network (# of Tx packets: _____)
- The broadcast address for this network is 10.199.11.255
- Fewer Rx packets have been sent across this network (# of Rx packets: _____)
- The IP address 10.199.10.235 could be found on this network
- Fewer Tx packets have been sent across this network (# of Tx packets: _____)

Use **ifconfig**, **iwconfig**, and what you have learned about networks to figure out which of the characteristics above belong to which network to help Safiya out. Then, write the characteristics in the column of the network that they belong to.

Note: If there is a blank space in parentheses at the end of a characteristic, be sure to fill in that space in your table with the corresponding value. For example, if gencyber2021-1 has the higher link quality and its link quality is 35/70, you would write, "This network has the higher link quality (35/70)," in your table.

Network name:_____	Network name: _____

2. When your table is finished, show your completed table to the instructor.
3. Answer the following question:
    a. For most of the characteristics, you had to be logged in to that network in order to view that characteristic. Why do you think administrators hide information from those that are not included in the network?

Day 2

Woohoo, it's Day 2! This morning, you will start off with a game of **Cyber Ball**. Next, you will dive deeper into the security layers as you examine security at the operating systems level in **Operating Systems: An Overview**. Along with this lesson is the activity titled **Terminal Commands and Resource Utilization**. In this activity, you will use some terminal commands to find out more about some tasks that are being performed by your Pi's operating system in the background.

Afterwards, you will go back up to the network layer to learn about the valuable tool NMAP in the **NMAP** lesson. Then, you will use NMAP for yourself in **NMAP Practice**.

Before having lunch, you will have your first of many chances to have fun with some Spheros in **Cyber Robotics – Spheros** (refer to **Cyber Robotics – Spheros**).

You will finish up your lessons for the day with **Wireshark**, an overview of a useful packet analyzing tool. You will also gain some first-hand experience with Wireshark in the **Wireshark Activity**.

Finally, you will end the day with another round of **Cyber Robotics – Spheros** (**refer to Cyber Robotics – Spheros**) and adding what you learned today to your group project.

Cyber Ball

Learning Objectives/Outcomes

Students will work sharpen their soft skills of collaboration, critical thinking, and innovation, as well as cyber security principle of abstraction and simplicity.
Collaboration: The students will work interdependently as part of a collaborative group in order to complete the task.
Critical Thinking: The students will exhibit problem solving skills in order to solve the challenge.
Innovation: Groups will innovate using an iterative approach where they surface and test hypothesis in order to "fail forward" or learn from their mistakes.
Abstraction and Simplicity: Students will test different approaches to solving the challenge, however the challenge is designed to reward the simplest and most efficient.

Game / Activity Length of Completion:

Introduction: 5 Minutes
Game Play: 30-45 Minutes
Debrief: 15 Minutes

Game / Activity Mode: Group

Game / Activity Description

Students are arranged into two teams of 5-15 students a team.
There are 3 rounds of game play.
Each team is given two buckets. They need to transfer balls from one bucket to another. The team that transfers the most balls from one bucket to the other wins.
The directions given to students are thorough yet vague. They are told just enough to solve the problem as a group, yet drawing from their problem solving skills to complete the task.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

The rules are straightforward. To score a point:

- Everyone is part of one team.
- Each ball must have air-time.
- Each ball must be touched at least once by every team member.
- Balls can't be passed to your direct neighbor to your immediate left or right.
- Each ball must return to the same person who introduced it into the system.

Google Slides Presentation: (Student and Instructor instructions & debrief)
https://docs.google.com/presentation/d/1PD9ey95MD0WyPgZ65VdEkBYjBZBw81E_UJEMyOUqclw/edit?usp=sharing

Terminal Commands and Resource Utilization

Learning Objectives/Outcomes

- Explore how commands can be used to gather information about running processes, memory usage, and storage utilization.
- Describe how security breaches could be detected through the command line.

Activity Length of Completion: ~30 mins.

Activity Mode: Individual

Activity Description

In this activity, students will use the commands discussed in the lesson along with a script file to identify the different characteristics of running processes as well as a system's memory usage.

Applicable Cyber Security Principles (highlight all applicable)

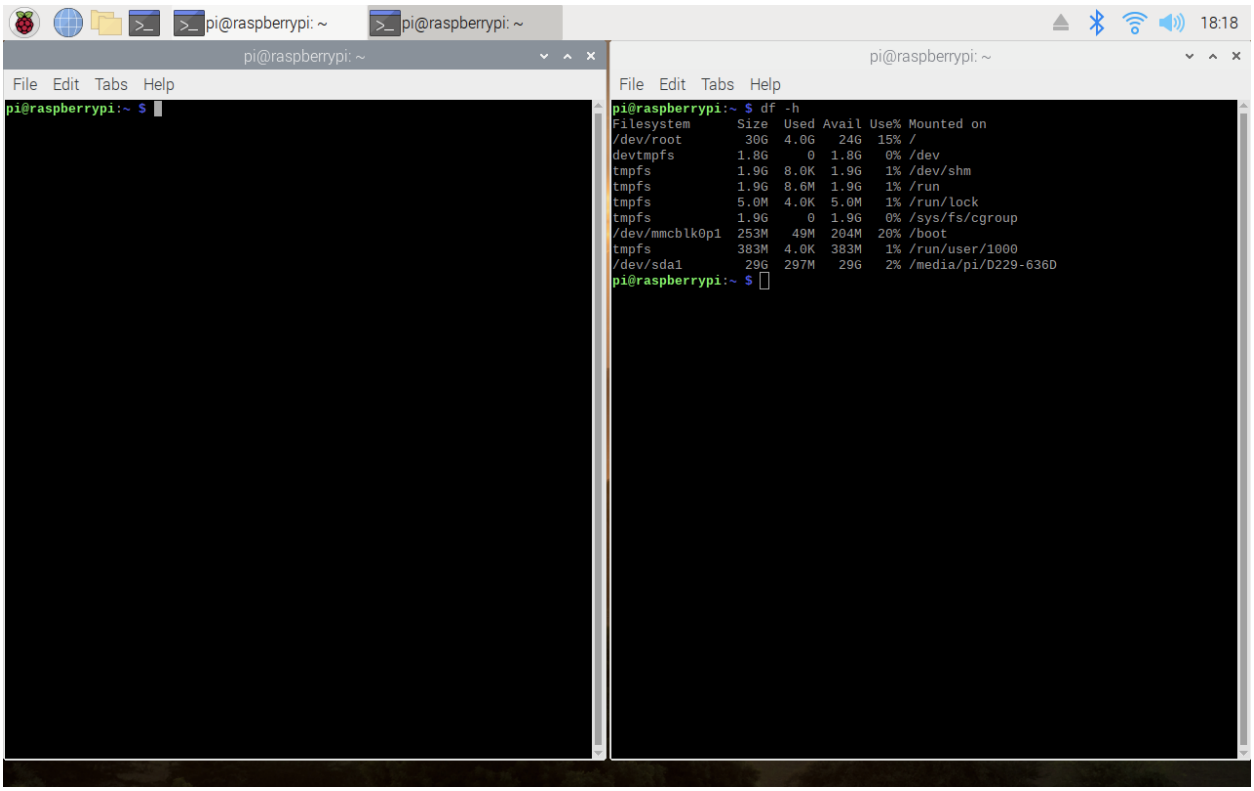
Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Process Isolation, Resource Encapsulation, Simplicity, Modularization

Activity Instructions

Beginner:

1. Log in to your Raspberry Pi and open a new Terminal window.
2. Type **df -h** into the command prompt and press **Enter** to see current disk usage for your system.

   Note: Please do not close out of this window as you will need the results of the above command later on.
3. While keeping the current window open, click the Terminal icon in the top left of the screen to open another Terminal window. Arrange your screen similar to the following image:



4. In the new Terminal window, type **top** and press **Enter**.

   Please read before continuing***: In the next step, you will run a python script that spawns multiple processes and uses these processes to create files. While this file is running, click back over to the Terminal running top. Look for a line in the top list with the command **python**. When you find the command **python** in the list, type **q** to freeze the data and terminate top. (Hint: top moves pretty fast, so try to move quickly!) Also, if for some reason you need to repeat step 5, you will need to delete the files that you created in step 5 before rerunning the program again or you will get an error.

5. With the top command still running, return to the original Terminal window. Make sure that you have navigated your command line to the folder where the file processSpawn.py is located. Once you have done so, type **python processSpawn.py** and press **Enter**. Write the information that you find about the process in the spaces below:

   PID: _____

   PR: _____

   VIRT: _____

   %CPU: _____

   %MEM: _____

6. Once you have collected the information above, close the window that was running top and return to the original Terminal window.

7. Type **df -h** once again and press **Enter**. Keep this window open for step 8.
8. Answer the following questions:
    a. Did any of the values listed in the results of **df -h** change? Which filesystems had the highest percent of disk space usage? Is this to be expected?
    b. How would the values that you collected in steps 2 and 7 be valuable when analyzing a system's storage utilization?
    c. How would the values that you collected in steps 2, 5, and 7 be valuable when trying to detect security breaches like malware running on the system?
    d. What are some strategies that could be used to protect the OS (and ultimately the entire computer system) from security breaches like the ones mentioned above?

Intermediate:

1. You should have been given a booklet titled *File Permissions on the Stardust Comet*. This book works like the choose-your-own-adventure books that you may have read before. The book will have several questions. Depending on how you answer those questions, you will flip to different pages in the book. Work through the booklet until you reach the last page.
2. Once you reach the end of the book, answer the following questions:
    a. In the booklet, you read what a disaster it can be if file permissions are not assigned correctly. If you were tasked with managing file permissions for a company here on Earth, how would you go about it? Try to create a plan for managing file permissions that would be easy for another person to use.

Advanced:

1. AFK Entertainment is one of the top-selling virtual-reality game development companies at the moment and you happen to work at the IT Help Desk of their San Francisco office. Below are some help requests from employees that you received this morning. For each request, write how you would solve the issue using the commands that you learned about in the lesson (free, df, ps, and top).

    Request 1:

    Sorry to bother you,

    This is probably a silly question. I am one of the program developers for the new *Zombie's Crypt* game and I am about to download a huge batch of source code. I want to make sure that I have enough room in memory first. The problem is that I am not used to working on this OS and I cannot figure out how to see how much memory I have left. Do you know how?

    Your response:

_____
_____
_____
_____
_____

Request 2:

Hi there!

This is Cassandra. I'm one of the script writers on the fourth floor development team and I have noticed that my computer has been running pretty slow the last couple of days. The only programs that I usually run are Microsoft Word, Spotify, and my email which shouldn't take up a lot of power. What do you think I should do?

Your response:

_____
_____
_____
_____
_____

Request 3:

Bill here from the Billing Department. I'm starting to get notifications from my computer saying that I am almost out of disk space. I would make some room, but I don't know what is taking up so much space and I want to make sure that I don't delete anything important! Can you help?

Your response:

_____
_____
_____
_____
_____

Request 4:

Hello,

My name is Mallory and I am a new intern here at AFK. I don't know if this is me being paranoid, but I think someone may be spying on my computer. Sometimes I see programs or files that I never downloaded or created appear and disappear on my computer. The same thing happens with my company email, too! Occasionally I see sent emails that I don't remember sending. This may just be anxiety from starting a new job, but is there anything that I can do to make sure that no one has installed something on my computer?

Your response:

_____
_____
_____
_____
_____

Answer the following questions:

a. Throughout the camp so far, we have talked about how cybersecurity must be implemented at each layer of the "cybersecurity onion." Do you think that any of the solutions to the help requests will require adding more security precautions to other layers besides the operating system layer? If so, which help requests? What security precautions would you add to the other layers?

NMAP Practice

Learning Objectives/Outcomes

- Demonstrate how to use hacking tools to extract information from the network.
- Explore how hackers could use the information gathered from nmap to execute an attack.

Activity Length of Completion: ~25 mins.

Activity Mode: Individual

Activity Description

In this activity, students will practice utilizing the commands introduced in the NMAP Lesson to find out more about the other hosts on the network.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

Beginner:

1. Log in to your Raspberry Pi and open a new Terminal window.
2. Type **sudo apt install nmap** and press **Enter**. This will install nmap on your Pi.
3. Type **y** when prompted and press **Enter**.
4. Type **ifconfig** and press **Enter** to find your IP address. Write your address in the space below.
    a. My IP address: _____
5. Type **sudo nmap -sP** followed by the range for your network (To get the range, take the first three numbers of your IP address and replace the last number with 0/24. For example, if your IP address is 192.161.3.4, the command will be **sudo nmap -sP 192.161.3.0/24**) Press **Enter**.
6. Choose an IP address from the results of step 5. Type **sudo nmap -O** followed by the chosen IP address. Press **Enter** to run the command. This command may take a couple of minutes to run. Record your results in the spaces below.
    a. IP address: _____
    b. Number of closed or filtered ports: _____
    c. Number of open ports: _____
    d. Device type: _____
    e. OS running: _____

7. If the address used in step 6 had one or more open ports, type **sudo nmap -sV** followed by the chosen IP address to see if you learn more about the services of these ports. Press **Enter**. This command may also take a couple of minutes. Then, record the results below.
    a. Port number, service and version: _____
    _____
    _____
8. Repeat steps 6-7 for a couple of the other IP addresses listed in the results of the command that you ran in step 5. You don't have to record these results, but if you wish to do so because they were more interesting than the results that you wrote above, you can record the new results in the spaces below.
    a. IP address: _____
    b. Number of closed or filtered ports: _____
    c. Number of open ports: _____
    d. Device type: _____
    e. OS running: _____
    f. Port service and version: _____
    _____
9. Answer the following questions:
    a. Were you able to find out any details about the operating systems and services of the other hosts on your network? Why might this information be important to a malicious user trying to attack the network?
    b. What are some ways that you could use the commands above along with other nmap commands to better secure the network?
    c. Are there any other nmap flags that you would want to try using in the future? (Remember: all flags can be seen by typing **nmap -h** and pressing **Enter**)
10. Discuss your answers with the other students around you.

Intermediate:

1. Combine nmap with the traceroute command using the following command (replace **<host>** with facebook.com, twitter.com, etc.):

    sudo nmap --traceroute --script traceroute-geolocation <host>

    Not only does this command show the IP address at each hop on the route, but it also shows the geographic location of each host. Run the command several times with website URLs that you know. Each time you run the command, it may take ~3 minutes to complete.

2. Write down any IP addresses that show up multiple times in the results along with their location and internet service provider (ISP):

IP address	Location	ISP

3. Answer the following questions:
    a. Did any of the IP addresses belong to the same location? If so, what does this indicate?
    b. How could a malicious user use the command above in an attack on a network?
    c. In some cases, users can fake the geolocation of their IP address using a service, such as a virtual private network. What are some reasons why a user may hide the real location of their system?

Advanced:

1. You are now a part of the network management team for the network that you are connected to. Congrats! Your coworker Natalia has asked you to do a port vulnerability analysis of all the systems connected to the network (do not worry about actually analyzing all of the computers, just analyze enough to get a good amount of practice). First, use **ifconfig** to find your IP address. Record it in the space below:

    My IP address: _____

2. Use **nmap** followed by the IP address range of your network (for example, 192.168.1.0/24) to find all active computers on the network along with the open ports for each system.
3. If your initial scan from step 2 did not contain any open ports, use **sudo nmap -sU <target>** where **<target>** is replaced by one of the IP addresses connected to the network. This command does a UDP scan of the ports.
4. Using the results from steps 2 and 3, make a list of IP addresses with open ports and circle whether or not you think these ports should be open, closed, or filtered. You may want to do

some research on each of the open ports to see if there have been any cyberattacks in the past using that port. After you decide whether the port should be closed or not, write a short explanation for why you chose that option.

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

IP address: _____

Port #: _____ Circle one of the following: Open/Closed/Filtered

Why? _____

5. Answer the following questions:
    a. What factors did you consider when deciding which ports stayed opened and which ports should be closed or filtered? Was it difficult to decide whether to open, close, or filter some of the ports? If so, why?
    b. Earlier in the week, you learned about the principle of minimization. How does this activity relate to minimization?

Cyber Robotics - Spheros

Learning Objectives/Outcomes

1. Students will demonstrate collaboration, communication, and critical thinking skills.
2. Students will demonstrate knowledge of a network topology through their physical sphero course.
3. Students will practice the engineering design process including identify the need, research the problem, develop possible solutions, select the most promising solution, construct a prototype, test and evaluate the prototype, communicate the **design**, and redesign.

Game / Activity Length of Completion:

8 Hours:
Understand the problem and protoype design: 1 Hour
Course Build: 2 Hours
Speed/Time/Distance Lesson: 1 Hour
Program spheros to drive the course: 2 Hour
Re-Design: 1 Hour
Re-Test: 1 Hour

Game / Activity Mode: Group

Game / Activity Description

The Sphero robotics course will take place over several sessions where students will collaborate to solve three main challenges: 1. course design 2. traffic management/traffic flow 3. re-routing/containerization/isolation/patching.

Course Design: Students will have 9 tiles to construct a sphero maze, where each sphero ball is used to represent the flow of packets of internet traffic between routers,

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

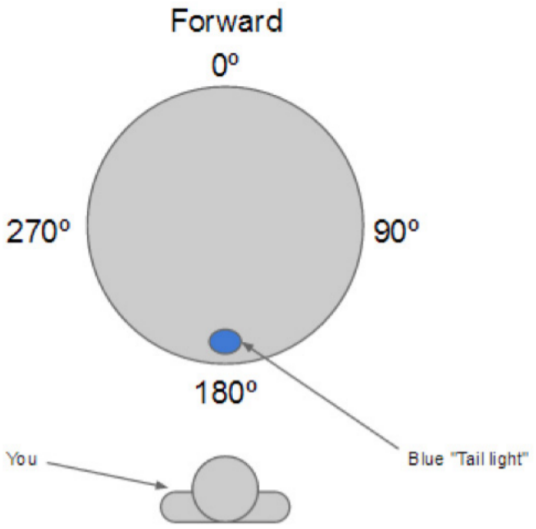
https://docs.google.com/document/d/1rLhVsHjZXbeDsxI3QpZChFf_wsA4lZk1SECg76o1HWo/edit?usp=sharing

Sphero Guide

What is a Sphero?
A Sphero is a robot ball with several features that can be controlled through mobile apps, including computer programs that the students build.
The main features are:
• Rolling. The Sphero can roll at a given speed and heading for a given amount of time.
• Colors. The Sphero can light up in any color.
• Bluetooth. Sphero connects to devices such an iPads, iPhones, and Android phones and tablets through wireless Bluetooth connections. This allows the Sphero to be controlled by a number of apps.

There are 4 education related apps available to control Sphero. Each of these is available for free from app stores such as iTunes and Google Play.
• Sphero. This is the main Sphero app used for firmware updates and general driving.
• Draw and Drive. Allows you to draw a shape with multiple colors and have Sphero roll in that shape and color.
• MacroLab. Creates simple programs (“macros”) that are a series of instructions for the Sphero through an easy-to-use graphical user interface.
• OrbBasic. Creates more complex programs using a text-based programming language.

Heading and Aiming
One of the things that makes Sphero so unique is that its heading is relative to the user, not relative to the ball. This makes the Sphero much easier to get to go where you want it to go.



The diagram shows how the heading works. Note that only 90 degree increments are shown in the diagram, but you can specify the heading down to 1 degree. Each time the Sphero is turned on, it needs to be "aimed," which means setting the direction that will be used for a heading of 0 degrees. This is accomplished with Sphero's "taillight". The taillight is a blue, light inside the Sphero. Each Sphero app has a button that lets you set the taillight, which looks like this:

To use this button, tap and hold on it, and then slowly move your finger around the circle. You will see the blue taillight rotate. When it is pointing directly at you (in other words, directly away from the direction you want the Sphero to roll for heading of 0 degrees), remove your finger. The student guides for all of the MacroLab lessons lead you through how to do this. For an interactive introduction on how to aiming, use the Sphero app.

Lesson 1: Speed, Time, Distance

You are going to be using Sphero to figure out how time, speed, and distance relate to each other. Sphero can be programmed to roll at a certain speed for a certain amount of time. You are going to be creating programs to do just that, and then measure how far it goes. By changing the speed that Sphero rolls at and the time it spends rolling, you will be able to learn about how time, speed, and distance all

relate. The Macrolab commands you are going to use in this lesson are:

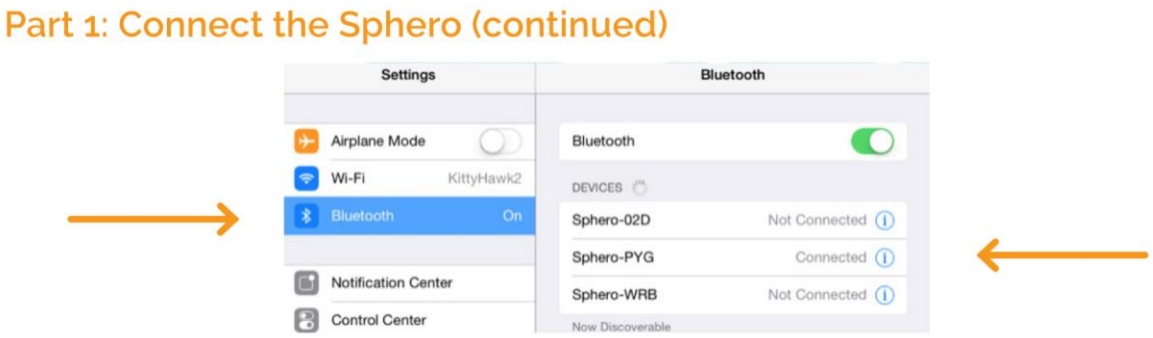

Roll – Makes Sphero roll at a given speed and heading.

Stop – Makes Sphero stop rolling immediately.

First you have to connect Sphero to the iPad (Part 1), then aim it (Part 2), and then there are two experiments (Part 3 and Part 4), and finally a challenge to see what you've learned (Part 4).

What can we figure out about time, speed, and distance? If something goes at a certain speed for a certain amount of time, we know that it will go a certain distance. But how are these related? If an object moves faster, we know it will move farther, but do we know how much farther? Same for if it moves for a longer period of time. The great thing about Sphero is that we can set its speed and we can set how long it should roll for. So then we just need to measure the distance. We can do that by marking the starting and ending points with masking tape, and measuring the distance between them. If we want to see how the distance has changed when we change the time or the speed, all we have to do is divide the new distance by the original distance. You'll do this math on your worksheet.

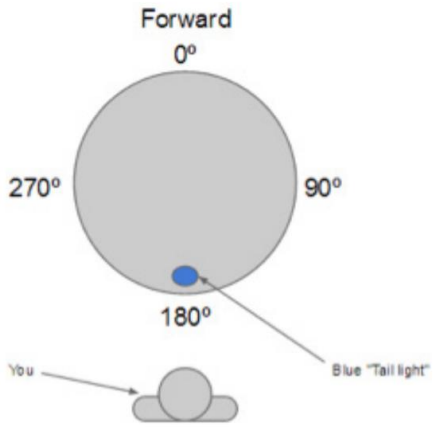
Part 1: Connect the Sphero First thing you need to do is to connect the iPad to Sphero. Here's how: 1. Pick up Sphero from its charging station and tap it twice on the logo to wake it up. You may have to tap it hard. It will start flashing colors when it is awakened out of its "sleep" state. 2. On your device, make sure Bluetooth is enabled. From the home page, click on Settings at the bottom. Then choose Bluetooth. 3. You will be shown a list of Spheros. Connect to the appropriate Sphero by tapping it. You can tell which Sphero is which by the names, which relate to the colors the ball is flashing. For example, if it flashes purple, then yellow, then green, then that is ball PYG. Select the one you want. Once successfully connected, it will say "Connected".



Part 2: Aiming Sphero

Sphero has a direction built into it that it thinks of as "straight ahead." This is called the orientation. The first thing we want to do is to aim the Sphero so that the orientation is on the path we want it to go.

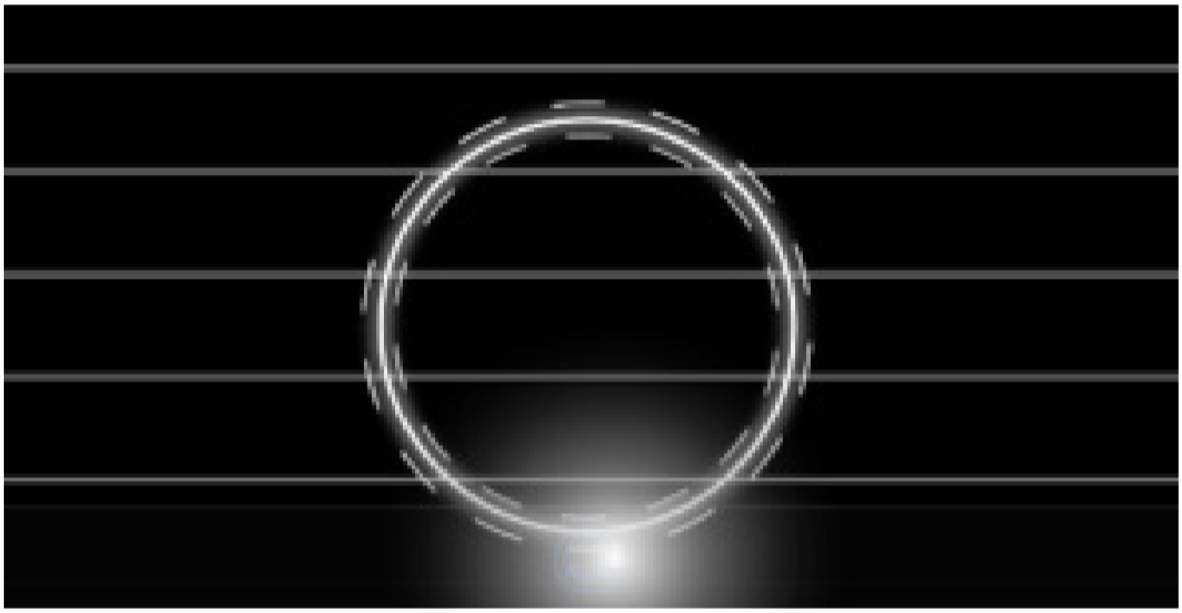
Each Sphero has a blue light inside of it called the “taillight”, which is always on the exact opposite side of the straight ahead direction. You are going to set the taillight so that it’s pointing right at you when you look down the path you want Sphero to go. Then, when it goes straight ahead, it will be on that path.



Follow these steps to aim the Sphero:

1. Go to the home screen and open MacroLab.
2. Have one of you hold the Sphero and stand at the beginning of the path you will use for your experiments.
3. Now, you will aim the Sphero in that direction. Have a second member of the group use the iPad. In MacroLab, you will see a circle with two arrows at the bottom center of the screen. Tap on it and hold it.

4. A white circle will appear. Move your finger slightly to rotate the insides of the Sphero. You will see a blue light inside the ball. Move it around until the blue light is directly facing the person holding the Sphero. This is the “taillight”, and shows the direction opposite where the Sphero will move when moving straight ahead.



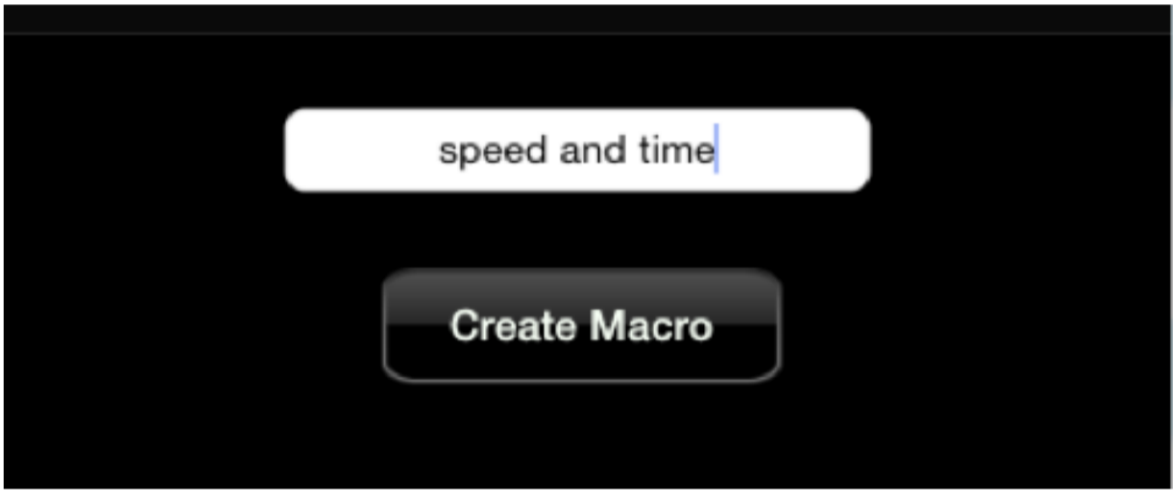
Important: For these experiments, the Sphero will travel a long distance, so be sure to aim the Sphero as accurately as you can to keep it on track. You can also re-aim Sphero anytime!

Part 3: Time and Distance

Now that we have Sphero going in the right direction, follow these steps for the first experiment:

1. Tap the + button at the bottom to create a new macro.

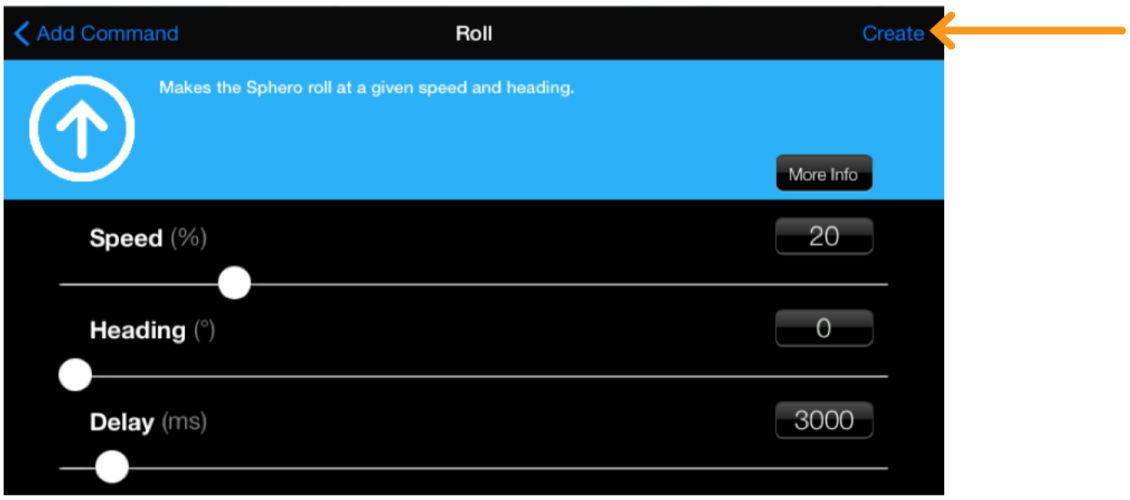
2. Where it says Macro Name, call type speed and time. Click Create Macro.



3. Add a command by tapping the + button at the bottom right.

4. Tap on Roll, the first command in the list

5. Change the Speed to 20 and the Delay to 3000. It may be easier to use the keyboard than try to get the right values with the slider. Leave the Heading at zero. Click the Create button up top.

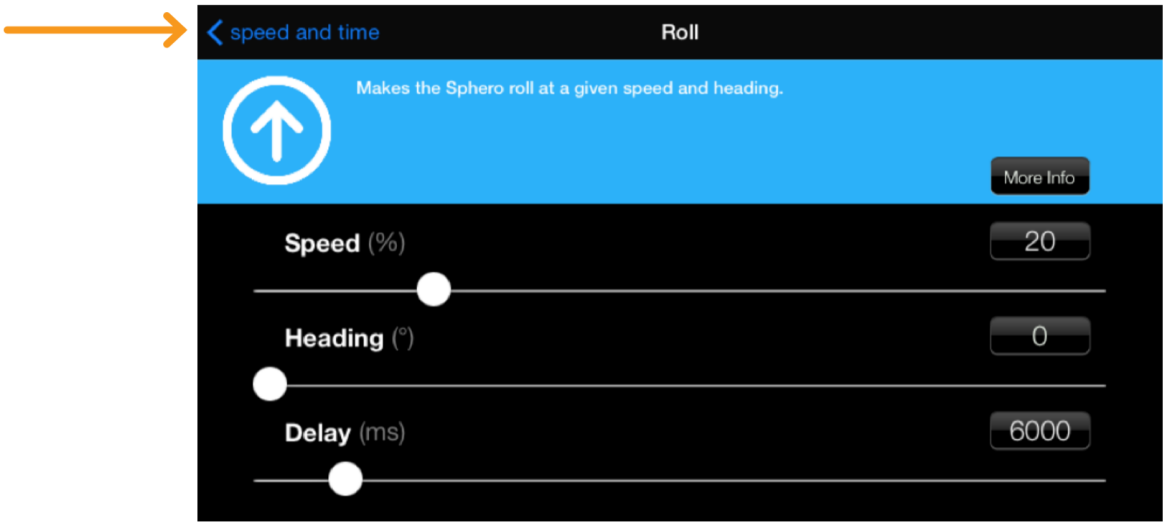


6. Add a new command using the plus key at the bottom.

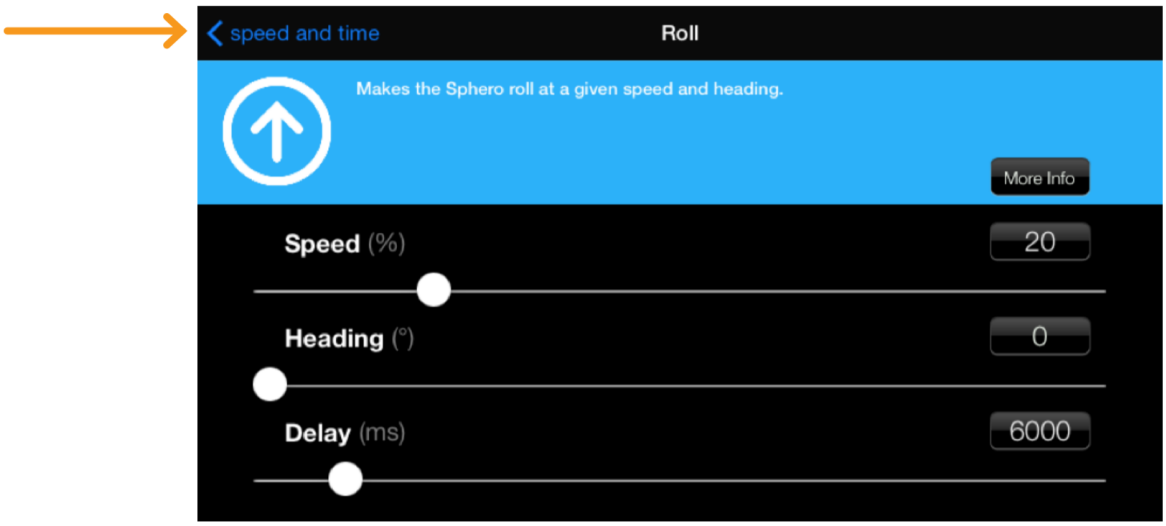
7. Choose Stop. This will stop Sphero from rolling immediately.

8. Move the bar all the way to the right to create a Delay of 255 and tap Create.

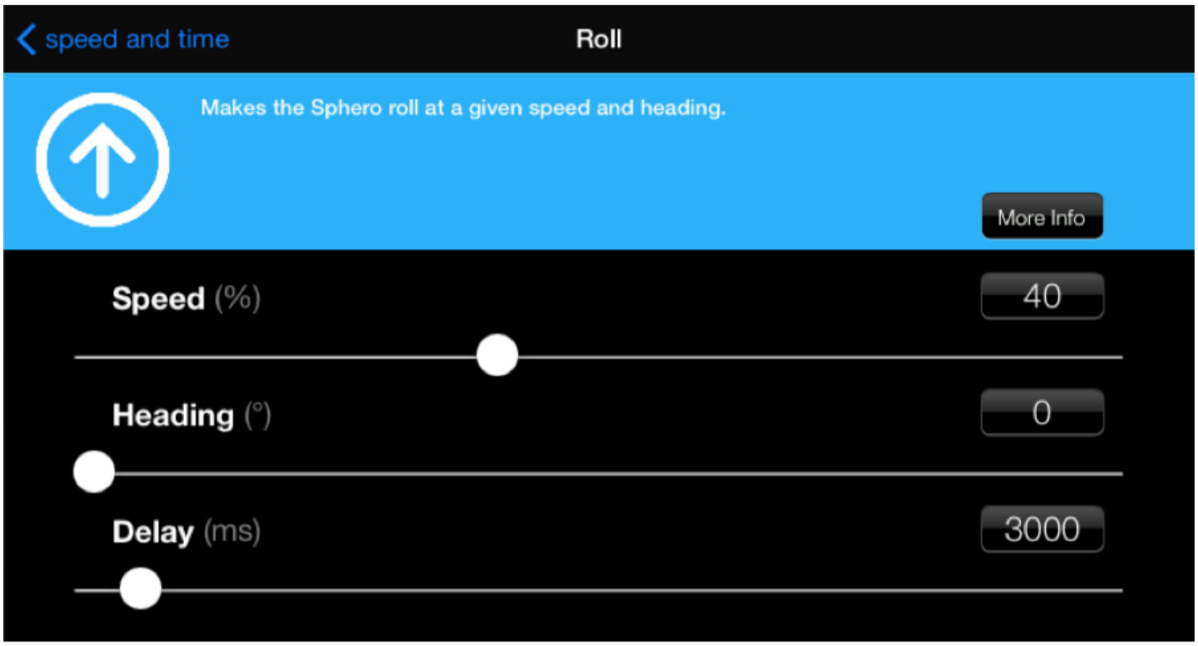
9. You've now written your first program! It tells Sphero to move at 20% speed for 3000 milliseconds, which means 3 seconds, and then stop. You will see a Roll command and a Stop Command. Click the Play button on the bottom.



10. The Sphero will roll slowly for 3 seconds. Now that we have Sphero going in the right direction, follow these steps for the first experiment: 1. Put a small piece of masking tape on the floor. Place the Sphero on top. 2. On the device, tap Play. The ball will roll for 3 seconds. (If it doesn't roll the path that you want, you can aim Sphero again to be more accurate.) 3. With your tape measure, measure how far it traveled. Write the answer on your worksheet. 4. Now tap on the Roll line and change the delay to 6000. This will make it roll for 6 seconds. (To figure out how many seconds, divide by 1000.) Tap speed and time when you are done.



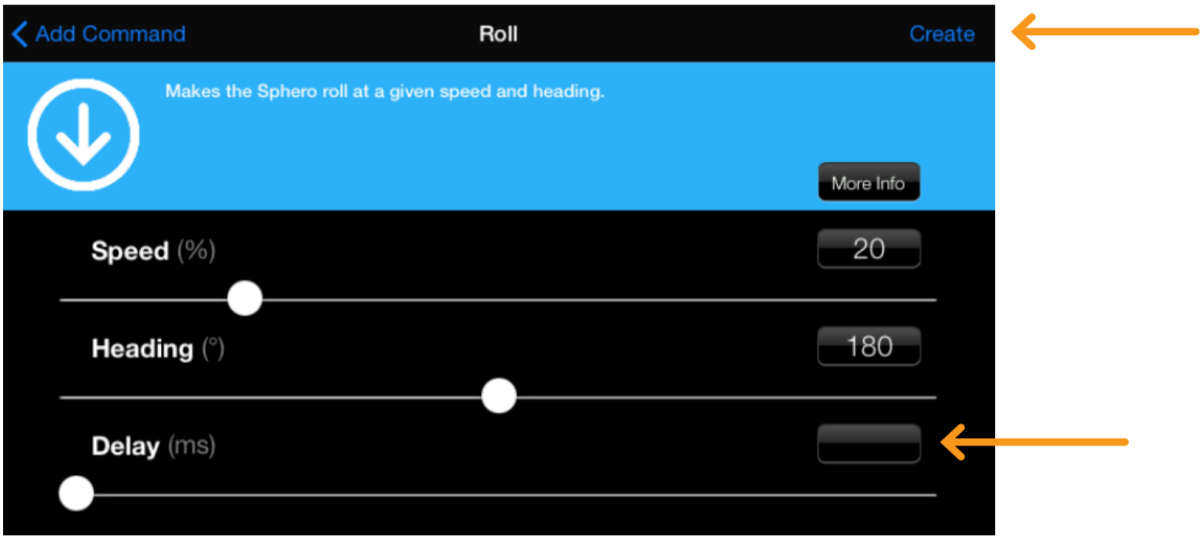
5. Put Sphero back on the tape and tap Play on the device. Measure the distance and write it on your worksheet.



6. Do this one more time with a delay of 9000 (9 seconds). Measure the distance and write it on your worksheet.

7. Either by hand or with a calculator (you can use the one on the iPad), divide the 6 second distance by the 3 second distance. Also divide 9 second distance by the 3 second distance. Write these numbers on the worksheet. (You only have to write one digit after the decimal point – that will be good enough.) What do you notice about the distances and the time it took for Sphero to go those distances? Discuss this as a class. Part 4: Speed and Distance In our first experiment, we had Sphero move at the same

speed, but we changed how long it was moving for. This time, we will have it move for the same amount of time, but we will change how fast it moves. Follow these steps: 1. Tap on Roll and change the delay time back to 3 seconds (a value of 3000.) Tap Play, and measure how far it goes. It should be about the same as the first time you measured it. Write it on your worksheet. 2. Now change the speed to 40%. Again, using the keyboard might be easier than the slider. Play, and measure how far it goes. Write it on your worksheet. Part 4: Speed and Distance (continued) 3. Now change the speed to 60%. Play, and measure how far it goes. Write it on your worksheet. 4. Either by hand or with a calculator, divide the 40% distance by the 20% distance. Also divide the 60% distance by the 20% distance. Write these numbers on your worksheet. (You only have to write one digit after the decimal point – that will be good enough.) What do you notice about the distances and the speed the Sphero used to go those distances? Discuss this as a class. Part 5: Challenge For our challenge, we will have Sphero move a distance out, and then you have to figure out how to move it back at a given speed to have it stop about where it started. Follow these steps: 1. Tap the Roll line and change the speed to 40% and the delay to 5000. This will make it roll at 40% for 5 seconds. Tap the Back button to get back to the speed and time screen. 2. Tap the Add button at the bottom of the screen. Tap Roll. Set the speed to 20% and the heading to 180. This will turn Sphero around 180 degrees and start rolling back at 20% speed. 3. For the delay, put in a number that you think will bring it back to where it started. Tap the Create button up top. 4. Add another Stop command with a Delay of zero.



5. Tap the Play button. Does the Sphero come back about to where it started? If not, adjust the delay number on the second Roll by tapping it. It won't come back exactly, but it should come within a few inches. (If your surface is slippery, it might only be within about a foot.) What number did you get? Why do you think that's the correct number?

Worksheet:

Names:

Part 3 - Time and Distance:
How far did the Sphero travel for Speed:
20% Time: 3000 (3 seconds):
How far did the Sphero travel for Speed: 20% Time: 6000 (6 seconds):
How far did the Sphero travel for Speed: 20% Time: 9000 (9 seconds):
What is the 6 second answer divided by the 3 second answer:
What is the 9 second answer divided by the 3 second answer:

Part 4 - Speed and Distance: How far did the Sphero travel for Speed: 20% Time: 3000 (3 seconds):
How far did the Sphero travel for Speed: 40% Time: 3000 (6 seconds):
How far did the Sphero travel for Speed: 60% Time: 3000 (9 seconds):
What is the 40% answer divided by the 20% answer:
What is the 60% answer divided by the 20% answer:
Part 5 - Challenge: What dalay value did you use in the challenge:

Wireshark Activity

Learning Objectives/Outcomes

1. Install Wireshark to complete the activity and for potential future use.
2. Demonstrate some basic functions with Wireshark.

Activity Length of Completion: ~25 mins

Activity Mode: Group

Activity Description

This activity will allow the students to get a basic understanding of Wireshark's interface and approach it in a hands-on fashion.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. Download Wireshark on your Raspberry Pi.
    a. Open the Terminal and enter the following
        i. $ **sudo apt-get install wireshark**
    b. When prompted, press "Y' then Enter.
    c. When prompted, select "Yes" using arrow keys and the Enter button
        i. If you accidentally select "No" use the following command to fix it
            1. $ **sudo dpkg-reconfigure wireshark-common**
    d. Enter the following command to add your default user to the group.
        i. $ **sudo usermod -a -G wireshark pi**
    e. Restart your Pi to initialize the changes.
        i. $ **sudo reboot**
2. Relaunch the terminal.
3. Launch Wireshark with the command
    a. $ **wireshark**
4. Enter into "wlan0" from the capture screen. This will automatically start a capture.

5. Open another terminal window and ping the IP address of one of your group members.
6. Watch Wireshark pick up network traffic. Soon, you should get pinged. You will notice when you do!
7. Use the information that you see to identify the student pinging you. Here is a spot for some notes if that helps you.
    a. _____
    _____
    _____
8. Now take a few minutes to browse the web. Visit some of these websites:
    a. utc.edu
    b. gen-cyber.com
    c. nsf.gov
    d. Nsa.gov
9. Explore the traffic that Wireshark has picked up. Take some notes about what you see.
    a. _____
    _____
    _____
    _____
10. When instructed, end the capture by pressing the stop button  at the top left. Also ensure that you have stopped pinging in the command line by pressing ctrl+c.
11. Discussion:
    a. How many packets did you capture?
    b. Are you surprised by how much traffic there is on a network?
    c. What could you see about the websites that you visited?
    d. How could this technology be used by hackers?
    e. How could you protect a network from this type of attack?

Day 3

It's Day 3, which means you are halfway there! This day begins with a round of **Pour Me**. Up next is a topic that you are probably already familiar with – applications and social media. You will explore how these topics relate to cybersecurity in **Applications, Social Media, Multimedia, Password Cracker**. Then, you will get to play the role of a hacker **in Password Cracking Exercise**.

After you finish your super-secret hacking mission, you will be introduced to some vital programming concepts in the **Basic Programming Concepts** lesson. You will practice using these concepts as you write "programs" in the **Writing Simple Programs** activity. Right before lunch, you will take a tour of the Mechatronics Lab. Then, you will get a chance to ask some of the RAs any questions that you have about cybersecurity or college in a panel discussion during lunch!

Lunch will be followed by **Python Programming and Cryptography**, a lesson in which you will learn about the amazing programming language of Python and cryptography, a concept utilized throughout cybersecurity. You will become more familiar with the concept of cryptography as you write your own encrypting and decrypting program in **Write the Caesar Cipher**.

Like yesterday, you will end the day with another round of **Cyber Robotics – Spheros** (**refer to Cyber Robotics – Spheros**) and working on your group project.

Pour Me

Learning Objectives/Outcomes

1. Students will demonstrate collaboration, communication, and critical thinking skills.

Game / Activity Length of Completion: 1 hour

Game / Activity Mode: Group

Game / Activity Description

Groups of 4 students are each given a rope that is attached to an elastic circle. There is a 5 gallon bucket that is full of 50 plastic balls. The object of the game is to work as a team to transfer the balls from one of the buckets to another bucket without spilling any balls. The team that does it in the least amount of time wins the challenge.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

Working as a team, you will decide how to transfer all 50 balls from one bucket to another by holding the ends of the rope. If you spill any balls you must start over from the beginning. The team that does it in the least amount of time will win this challenge. Any questions?

Password Cracking Exercise

Learning Objectives/Outcomes

- Gain more experience working in the command-line environment.
- Take user information and intuitively add password options to a password list.
- Implement a dictionary attack in an effort to access a Secure Shell Protocol from another server.

Activity Length of Completion: ~30 mins.

Activity Mode: Individual

Activity Description

In this activity you will use some password-cracking software in the command-line in order to implement a dictionary attack and access the Secure Shell Protocol of a preconfigured server.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. Open the Terminal
2. Install libssh
    a. $ **sudo apt-get install libssh-dev**
    b. $ **y**
3. Install Hydra
    a. $ **wget https://github.com/vanhauser-thc/thc-hydra/archive/refs/tags/v9.2.tar.gz**
    b. $ **tar -xzf v9.2.tar.gz**
    c. $ **cd thc-hydra-9.2/**
    d. $ **./configure**
    e. $ **make**
    f. $ **sudo make install**
    g. $ **cd ..**
4. Install BruteDum
    a. $ **git clone https://github.com/GitHackTools/BruteDum**
    b. $ **cd BruteDum**
5. Download a password dictionary
    a. $ **wget https://raw.githubusercontent.com/berzerk0/Probable-Wordlists/master/Real-Passwords/Top207-probable-v2.txt**

**(Note: This link will only work during the camp)

6. Ensure that BruteDum and the password dictionary are installed
    a. $ **ls**
        i. You should see brutedum.py README.md Top207-probable-v2.txt
7. Use the provided user information to add some password guesses to the password dictionary
    a. $ **echo "***yourPasswordGuess***" >> Top207-probable-v2.txt**
        i. Note: leave the quotation marks, change the italics.
    b. Repeat the above command for as many guesses as you would like to add
8. Once you have added all of the guesses that you would like to add, execute BruteDum by using the following commands
    a. $ **python3 brutedum.py**
    b. $ *given_IP_address*
    c. $ **y**
    d. $ **4**
    e. $ **2**
    f. $ **n**
    g. $ **gracehopper**
    h. $ *file path of text document w/ passwords*
    i. $ **y**
    j. If you have successfully included the password in the password dictionary the program will display the login and password
        i. login: _____
        ii. password: _____
    k. Add some more guesses to the password dictionary if you do not successfully identify the password, then repeat the process
    l. Tell the program that you would not like to continue once you successfully receive the password
9. Once you have exited BruteDum and have the login and password, execute the following commands
    a. $ **ssh gracehopper@***given_ip_address*
    b. Enter the password when prompted
    c. Enter "yes" if the terminal asks you a yes or no question at this point
10. You should be in now! To confirm that you have made it to the right place, do the following:
    a. $ **ls**
    b. $ **more W***[hit tab then enter]*
11. The first three people to successfully complete step 10 will receive points!
12. Discussion questions
    a. Did you successfully guess the password?
    b. What did you find once you got in?
    c. How could Grace improve her password?

Writing Simple Programs

Learning Objectives/Outcomes

- Demonstrate how to create a program that handles several different conditions through the use of if statements and else statements
- Explore how to use while loops or for loops within a program to repeat a specific function or functions a number of times

Activity Length of Completion: ~30 mins.

Activity Mode: Group/Individual

Activity Description

In this activity, students will practice using the programming components that they learned about in the Basic Programming Concepts lesson to build programs of their own.

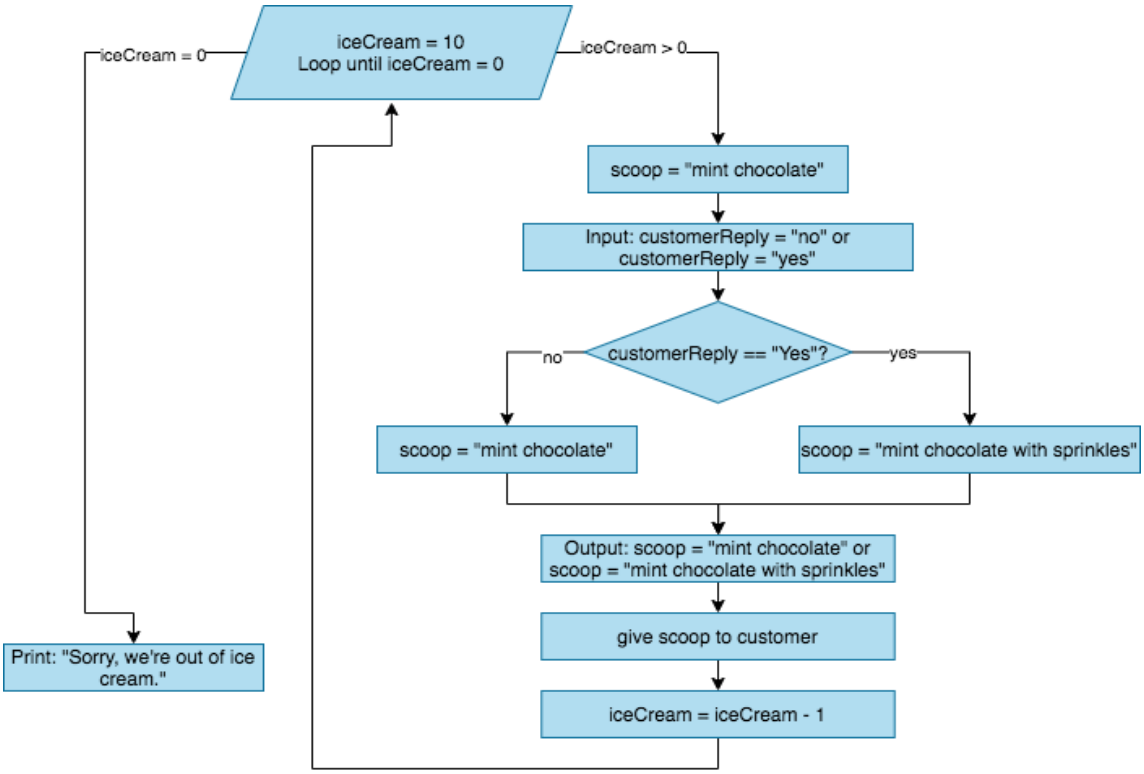
Applicable Cyber Security Principles (highlight all applicable)

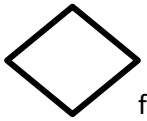
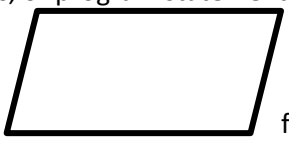
Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions (for Students)

Beginner:

In this activity, you will work together in groups of four to create a series of diagrams that represent programs that you would write in code. Structure your diagrams similar to the one below:



To make sure that everyone can easily understand each other's diagrams, use  for inputs, variables, or program statements that are not if statements or a loop. Use  for if statements and  for for loops. Be sure to draw your diagrams on the paper that has been given to you large enough for other students to be able to read from across the room.

1. **For this question, you will work together as a class along with the instructor.** Create a program that tells the user whether or not they can cross the street at an intersection. The input of this program is the crosswalk light across the street. In your diagram, do not put a specific string as the input, just create a variable that will hold the input. If the light shows a little man, then let the user know that they can go. If the light shows a hand with a countdown, then let the user know that they can go but that they have to hurry. Otherwise, let the user know that they will have to wait.
2. **For this question, you will work together in your groups.** Let us say that the user is playing a game of ring toss. In this case, the input is whether or not the ring landed successfully on the top of the bottle (Again, create a variable to hold the input value but do not use any specific value for the input). **While one variable is required, you can add as many as you like**. If the ring landed correctly, let the user know that they have gotten 20 points. Otherwise, let the user

know that they missed. The user should get three tries (In other words, the program should perform the process three times).

3. Once everyone has finished questions 1 and 2, share your solutions with the other groups.
4. Answer the following questions individually:
    a. Were you surprised by some of the solutions of the other students? In what ways were the other solutions different from the solutions developed by your group?
    b. Consider if the scenario of step 1 was implemented at a real crosswalk. If a malicious user were to somehow manipulate the program to tell the user to go when the crosswalk light was not a man or a hand with a countdown, what consequences would this have?
    c. Can you think of possible ways to secure your program so that the situation described in question b does not happen?

Intermediate:

1. **For this question, work on your own**. Write a program that lets a user know whether or not they have gotten a Yahtzee. In other words, take in the values of five dice as input and check to see if all five values are the same (five ones, five twos, etc.). If all five values are the same, let the user know that they have gotten a Yahtzee. Otherwise, let the user know that they did not have luck this time. For this program, do not worry about putting specific values for the inputs. Just use '?' for each value (ex: input1 = ?).
2. When you have finished writing the program, answer the following question:
    a. Using five variables to store five input values can make the program a bit more complicated than it needs to be. Can you think of a way to simplify the way that these values are stored? If you do not know an answer, you may use the Internet or the instructor for help.

Advanced:

1. **For this question, work on your own**. Write a program that simulates the classic game Rock, Paper, Scissors. Take in two player's inputs. The possible inputs are "rock," "paper," or "scissors." For the inputs, just use '?' for the value of each input, do not worry about picking specific values (ex: val1 = ?). Examine the inputs and let the players know who won or if the game was a draw. Remember, the possible outcomes are:
    - rock + paper = paper
    - rock + scissors = rock
    - rock + rock = draw
    - paper + scissors = scissors
    - paper + paper = draw
    - scissors + scissors = draw

    Note: Even if the two inputs are in a different order, they will still produce the same result.

2. When you have finished writing the program, answer the following questions
    a. How did you approach writing this program?
    b. What would you say was the most challenging part of creating this program?

c. Do you feel like you are more confident in translating a concept into a program than you were before this activity?

Write the Caesar Cipher

Learning Objectives/Outcomes

- Demonstrate how to implement simple programming concepts in a Python program, including variables, if statements, and for loops.
- Use the Caesar cipher algorithm to encrypt and decrypt confidential data.

Activity Length of Completion: 30 mins.

Activity Mode: Individual

Activity Description

In this activity, students will write a program in Python that utilizes basic programming structures as well as the Caesar cipher to encrypt and decrypt chosen messages.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

Beginner:

1. On your Raspberry Pi, click on the raspberry icon at the top left corner of the screen.
2. Scroll over **Programming**, then click on **Thonny Python IDE**.
3. Click on **New Project** to create a new Python file.
4. Click Save, then save the file as **caesarName** where **Name** is your name. Press **Enter**.
5. In the new python file, you will write a program that encrypts and decrypts a message using the Caesar cipher. To start, you will need to create variables to hold the shifted alphabet, plaintext, and ciphertext. Type the following lines:

```
enPlain = "this is a super secret message"

enCipher = ""

dePlain = ""
```

```
alpha = "abcdefghijklmnopqrstuvwxyz"

newalpha = ""
```

enPlain holds the plaintext while **enCipher** will hold the ciphertext after encryption. **dePlain** will hold the plaintext after decryption and **alpha** holds the normal alphabet that you will use to create the shifted alphabet. This shifted alphabet will be stored in **newalpha**. Press **Enter**.

6. A variable to hold the shift value will also need to be created in the event the programmer wants to change the shift value. To make things simple, name this variable **shift** and assign it the value of **3**. Press **Enter**.
7. Now, you can move on to creating the shifted alphabet that will be used to encrypt the message. Type the following line:

```
for i in range(len(alpha) - shift):
```

This line creates a for loop that will loop from 0 to the length of **alpha** minus the value of **shift**. The reason that you do not loop all the way through **alpha** is because if you tried to get the letter three spaces to the right of 'x', you would get an index error since the index does not loop around back to the beginning of **alpha**. Press **Enter** to enter the indented region of the for loop.

8. Fill in the for loop that you created in step 7 with a program statement that fills **newalpha** with the shifted alphabet. Choose which statement to use for this below (Note: **newalpha += …** is the same as writing **newalpha = newalpha + …**):
    a. `newalpha += alpha[i]`
    b. `newalpha += alpha[i + shift]`
    c. `newalpha += i`

    If you chose b as your answer, congratulations, you are correct! This statement takes the character 3 spaces to the right of the current character in **alpha** and adds it to the end of the **newalpha** string. If you chose another option, take a moment to think about why these other options might be incorrect. Then, type option b. into your code. Press **Enter**, then **Backspace** to exit the for loop.

9. Remember, you did not loop all the way through **alpha** in the for loop. So, some letters from the alphabet are missing in **newalpha**. Which of the statements below would correctly add the missing letters to **newalpha**?
    a. `newalpha += alpha`
    b. `newalpha += alpha[shift]`
    c. `newalpha += alpha[shift:]`
    d. `newalpha += alpha[0:shift]`

    The correct answer to this question is d! Since you only included characters three spaces to the right in **newalpha** in the for loop above, letters a, b, and c were left out since they are the first three letters. So, they need to be added to the end of **newalpha** to complete the shifted alphabet. Option d does this by adding the characters in **alpha** from index 0 to the index before

the value of **shift** (In this case, it will be from index 0 to index 2) to the end of **newalpha**. Type option d, then press Enter to move to the next line.

10. Now that you have the shifted alphabet, you are ready to start encrypting! To do this, you will want to loop through all of the letters in the plaintext and change each letter to its encrypted counterpart. Begin by writing a program statement that creates a for loop to loop from 0 to the end of **enPlain**. Use **i** to keep track of the indices within the loop (Hint: use the for loop that you typed in step 7 as a template.). Once you have finished writing the statement, press **Enter**.
11. You want to make sure that you preserve the spaces in the ciphertext so that they will still appear in the decrypted plaintext. In the indented region of the for loop that you just created, type the following line then press **Enter**:

```
if enPlain[i] == " ":
```

This if statement checks whether or not the character at the current index is a space. If it is a space, you want to add the space to the end of the ciphertext (Remember: you are using **enCipher** to hold the value of the encrypted ciphertext). Which program statement should be added to the body?

a. `enCipher += enPlain[i]`
b. `enCipher += enPlain[i+shift]`
c. `enCipher += enPlain`

And the answer is...a! Since the if statement confirms that **enPlain[i]** is a space, all you have to do is add **enPlain[i]** to the end of **enCipher**. Type option a and press **Enter,** then **Backspace.**

12. If the character at the current index is not a space, you want to encrypt the letter using the **newalpha** alphabet that you created earlier. Type the following lines in your python file:

```
else:
        for j in range(len(alpha)):
```

The else statement tells the computer to only run the following commands if the current character is not a space. The next statement is an inner for loop that you will use to compare the current letter in **enPlain** with each letter in **alpha**. Press **Enter** to enter the body of the inner loop.

13. You want to find which index in **alpha** corresponds to the letter at the current index in **enPlain**. For example, if the letter in **enPlain** is 'a', the corresponding index in **alpha** would be 0. What kind of statement could you use to compare letters in **alpha** to a letter in **enPlain**?

a. **for loop**
b. **if statement**
c. **print statement**

Hopefully, you picked option b, because that is the correct answer. Write an if statement that checks if the letter at the current index in **enPlain** is equal to the letter at the current index in

alpha (Remember: use **i** for indexing **enPlain** and use **j** for indexing **alpha**). Once you are happy with the if statement, press **Enter** to enter the body of the if statement.

14. If the two letters are equal, you want to switch the letter in **enPlain** to the correct encrypted letter in the **newalpha** alphabet. One way to do this is by changing the plaintext letter to the letter in **newalpha** with the same index as the one in **alpha**. For example, if the plaintext letter is 'a', the **alpha** index is 0. So, you would replace 'a' with the letter at index 0 in **newalpha**, which is 'd'. Which of the following statements performs this task?
    a. `enCipher += newalpha[j]`
    b. `enCipher += enPlain[j]`
    c. `enCipher += alpha[j]`

    In this case, the answer is a. This line takes the character at index j in **newalpha** and adds it to the ciphertext string. Type option a and press **Enter**.

15. Once you have encrypted the letter, it would be unnecessary to keep looping through newalpha. Type **break** to create a break in the loop and to return to the outer loop. Press **Enter** and then **Backspace** four times to exit out of all loops.
16. At this point, your program should be able to take in a string and output the encrypted version of it. To double check this, write three print statements: one to print the shifted alphabet, one to print the plaintext before encryption, and one to print the ciphertext after encryption.
17. Press the **Run** button at the top to run your code. Your output should look similar to the output below:

    ```
    defghijklmnopqrstuvwxyzabc

    this is a super secret message

    wklv lv d vxshu vhfuhw phvvdjh
    ```

    If anything in your output is off, reexamine your code to see if there are any errors.
18. Now, you can move on to decrypting the ciphertext that you just created. The good news is that the process of decryption is virtually the same as encryption, so you can reuse some code! Copy all lines of code from the second for loop to the break statement and paste it to the end of your code. Then, make the following changes to the copied code:
    a. replace all instances of **enPlain** in original code with **enCipher** in copied code (This way, each character of **enCipher** is looped through instead of **enPlain**.)
    b. replace all instances of **enCipher** in original code with **dePlain** in copied code (This way, each decrypted character is added to **dePlain** instead of **enCipher**.)
    c. replace all instances of **alpha** in original code with **newalpha** in copied code (This way, each character in **enCipher** is checked against the letters in **newalpha**.)
    d. replace all instances of **newalpha** in original code with **alpha** in copied code (This way, each character in **enCipher** is replaced by the corresponding letter of the normal alphabet.)

Take a moment to read through the copied code to ensure that you understand how the code takes in **enCipher** as input, decrypts the string, and stores the decrypted plaintext in **dePlain**. Once you are satisfied, move on to the next line.

19. Finally, use a print statement to print the decrypted plaintext to the screen.
20. Run your program one more time to make sure it outputs the correct plaintext. Congratulations, you have written your first program!
21. Once you have finished, answer the following questions:
    a. There is almost always more than one way to write a program. Is there any way that you would code this program differently?
    b. How could this program be used to protect certain sensitive pieces of information, such as passwords, emails, etc.? Answer this question in terms of confidentiality, integrity, and availability.
    c. What are some possible weaknesses of this program? How could you modify this program to create a more secure encryption algorithm?

Intermediate:

1. The Zig Zag cipher is another simple encryption algorithm. Without using the Internet, try to figure out how the Zig Zag cipher works by reading through the code below:

main.py

```
plain = "coffee and cakes"
cipher = ""

print("Plaintext: " + plain)
for i in range(len(plain)):
    if i % 2 != 0:
        if plain[i] == " ":
            cipher += "z"
        else:
            cipher += plain[i]
for i in range(len(plain)):
    if i % 2 == 0:
        if plain[i] == " ":
            cipher += "z"
        else:
            cipher += plain[i]
print("Ciphertext: " + cipher)
```

The output of the above program is:

```
Plaintext: coffee and cakes
Ciphertext: ofeadckscfeznzae
```

Here are some hints for reading the program:

- "total += 5" is the same as writing, "total = total + 5"
- The % is known as the **modulo symbol**. It produces the remainder after two numbers are divided. For example, 9 % 2 = 1 since 9 / 2 = 4 with a remainder of 1.
- Sometimes, it is helpful to walk through each line of the program on paper to better understand how the program works as a whole.

2. Once you fully understand how the Zig Zag cipher and the program works, answer the following questions:
    a. If you had to describe the Zig Zag cipher in one sentence to another student, what would you say?
    b. If you did not already know the plaintext, how would you go about decrypting the encrypted message? See if you can come up with a formula for decryption. **Note**: The decryption formula does not have to be formatted like a math formula. Think of it more like a recipe with a series of steps.

Advanced:

1. While encryption is important, the process of decryption is equally as important. Say that you come across a piece of ciphertext. You know that it was encrypted using a Caesar cipher, but you do not know the key that was used. You could try all 25 different keys, or you could find the most frequently used letters in the ciphertext and match those to the most frequently used letters in the alphabet. For example, if 'y' is used the most times in the ciphertext and 'e' is the most popular letter in the alphabet, then the key is most likely 20 (Since 'e' is 20 spaces away from 'y'). This is known as **frequency analysis**.

    Write a program that performs a frequency analysis on the following message:

    vs lbh frr guvf pbatenghyngvbaf lbh unir penpxrq gur pbqr

    In other words, write a program that:
    - takes the ciphertext as input
    - finds how many times each letter in the alphabet is used in the ciphertext

    Then, use that information to guess the key used to encrypt the message. Remember, the most frequently used letters in the English alphabet are E, A, R, I, and O. To make sure that your guess for the key is correct, you can write a program to decrypt the message or decrypt the message by hand on paper. Once you are sure that your guess is correct, write your answer for the key value in the space below:

    Key value: _____

2. Answer the following questions:
    a. Would you say that you were able to find the key and decrypt the message faster using your program than you would have by trying all of the keys?
    b. Would you add anything to the program to simplify the decryption process?
    c. Why is it important to always keep the key secret no matter what encryption algorithm is used?

Day 4

Day 4 and you are almost there! First, you will show off your construction skills in **Spaghetti Tower.** After that, you will look at cybersecurity in the data layer in **Databases and CIA**. You will get a chance to create your own databases in **MySQL Setup and CIA**.

You will then combine your knowledge of Python programming with what you just learned about databases in **Database Tables and Python**. Once you finish this lesson, you will have a chance to write a program that can access a database in the **Database Tables and Python Exercise**. Next, you guessed it, it's back to the Spheros (**refer to Cyber Robotics – Spheros**). Before lunch, you will get a chance to tour the High Performance Computing Lab. During lunch, you will hear from UTC's Student Outreach program.

Once lunch is over, you will look at how cameras and filters are utilized in cybersecurity in **Cameras and OpenCV**. Following the lesson, you will use your programming skills to capture some pictures with your Pi camera and apply some filters with OpenCV in the activity **Images, Filters, and Cybersecurity**.

Lastly, you will finish working with the Spheros (**refer to Cyber Robotics – Spheros**) and add what you learned today to your group project.

Spaghetti Tower Challenge

Learning Objectives/Outcomes

1. Students will demonstrate collaboration, communication, and critical thinking skills.
2. Students will practice the engineering design process including identify the need, research the problem, develop possible solutions, select the most promising solution, construct a prototype, test and evaluate the prototype, communicate the **design**, and redesign.

Game / Activity Length of Completion:

2 hours

Game / Activity Mode: Group

Game / Activity Description

The students will be given a package of uncooked spaghetti noodles and a package of marshmallows. The teams are to build the tallest free standing spaghetti tower using only the spaghetti noodles and marshmallows. Limitations: The towers must start from the floor. No supports or props are to be used.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

Build the tallest freestanding tower using only the supplies given to you; uncooked spaghetti and marshmallows. The tower is only allowed to touch the floor, but nothing else may be used for support, and it cannot touch anything other than the floor. Build your tower by placing it on the table cloths to keep your area clean.

MySQL Setup and CIA

Learning Objectives/Outcomes

1. Set up MySQL/MariaDB and configure a user
2. Understand security implications in regards to databases

Activity Length of Completion: ~25 mins.

Activity Mode: Individual

Activity Description

In this activity you will set up MariaDB, a fork of MySQL, on your Raspberry Pi and configure a new user. You will then discuss with your peers the necessary security considerations for database implementation.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

Beginner:

1. Launch the terminal on your Raspberry Pi
2. In general, type the bold exactly the way it is presented and change the italics to suit you personally.
3. Start with the following commands
    a. $ **sudo apt update**
    b. $ **sudo apt upgrade**
4. MariaDB is one of the most suitable databases to use on Raspberry Pi OS. MariaDB is a fork of MySQL. To install MariaDB, enter the following commands
    a. $ **sudo apt install mariadb-server**
    b. $ **sudo mysql_secure_installation**
        i. Inherently, you will not have a password and can just press enter to skip through the “Enter password” request.
        ii. When asked the yes/no questions, select no for each one except for “Reload privilege table now?”; select yes on this one.
5. Now you must configure MariaDB for use

   a. $ **sudo mysql -u root -p**
   b. Inherently, you will not have a password and can just press enter to skip through the "Enter password" request.
6. Create a user and set a password that is for only that user
   a. > **CREATE USER** *username***@localhost IDENTIFIED BY '***password***';**
   b.
```
MariaDB [(none)]> CREATE USER Meredith@localhost IDENTIFIED BY 'Grey';
```
   for example: Desired username^^^ Desired password^^^
   c. Confirm the creation of your user using the following command
      i. > **SELECT host, user FROM mysql.user;**
7. Grant all privileges on every database to your user
   a. > **GRANT ALL PRIVILEGES ON *.* TO '***username***'@'localhost';**
8. Create a test database to later confirm access to your new user and exit root access.
   a. > **CREATE DATABASE Test;**
   b. > **exit**
9. Log back in with your new user
   a. $ **sudo mysql -u** *username* **-p**
   b. When prompted, enter your chosen password
10. Confirm access to the Test database
   a. > **SHOW DATABASES;**
   b. As long as you see "Test" in the list, you are good to go! Exit MariaDB once more.
11. Discussion:
   a. Did you encounter any challenges during the set up of MariaDB? Can you foresee any future challenges?
   b. What security considerations for databases should we have based on previous discussions of Physical, Network, OS, and Application Layers?

Intermediate:

1. If you have completed the previous section you may skip to step 7, otherwise continue to step 2.
2. Launch the terminal on your Raspberry Pi
3. Start with the following commands
   a. $ **sudo apt update**
   b. $ **sudo apt upgrade**
4. MariaDB is one of the most suitable databases to use on Raspberry Pi OS. MariaDB is a fork of MySQL. To install MariaDB, enter the following commands
   a. $ **sudo apt install mariadb-server**
   b. $ **sudo mysql_secure_installation**
   c. $ **pip3 install mariadb**
5. Now you must configure MariaDB for use
   a. $ **sudo mysql -u root -p**
   b. Inherently, you will not have a password and can just press enter to skip through the "Enter password" request.
6. Create a user and set a password that is for only that user
   a. > **CREATE USER** *username***@localhost IDENTIFIED BY '***password***';**

b.
```
MariaDB [(none)]> CREATE USER Meredith@localhost IDENTIFIED BY 'Grey';
```
for example: Desired username^^^ Desired password^^^

c. Confirm the creation of your user using the following command
   i. > **SELECT host, user FROM mysql.user;**

7. Create a database to keep track of some business information.
   a. > **CREATE DATABASE Business;**
8. Create a few tables to organize data. Don't worry about understanding this too much; you will learn about tables in another lesson.
   a. > **USE Business;**
   b. > **CREATE TABLE Sales (DollarAmount INT PRIMARY KEY);**
   c. > **CREATE TABLE Schedule (ShiftTime VARCHAR(15) PRIMARY KEY);**
   d. > **CREATE TABLE ClientList (CustomerName VARCHAR(20) PRIMARY KEY);**
9. Exit MariaDB and log back in as root.
   a. > **exit**
   b. $ **sudo mysql -u root -p**
10. Create a few employees using the instructions in step 6. Be creative, but keep track of the information.
   a. Manager
      i. Username: _____
      ii. Password: _____
   b. Accountant
      i. Username: _____
      ii. Password: _____
   c. Sales crew
      i. Username: _____
      ii. Password: _____
11. Grant the appropriate levels of permissions to the appropriate users. Here are some specific options. You may use more than one option for your users. Some options may not be used.
   a. Global privileges - privileges to administer the database and manage user accounts, as well as privileges for all tables, functions, and procedures
      i. Option 1: > **GRANT ALL PRIVILEGES ON *.* TO '***username***'@'localhost';**
   b. Database privileges - privileges to create tables and functions, as well as privileges for all tables, functions, and procedures in the database
      i. Option 1: > **GRANT ALL PRIVILEGES ON Business.* TO '***username***'@'localhost';**
   c. Table privileges - privileges include the ability to select and change data in the table
      i. Option 1: > **GRANT ALL PRIVILEGES ON Business.Sales TO '***username***'@'localhost';**
      ii. Option 2: > **GRANT ALL PRIVILEGES ON Business.Schedule TO '***username***'@'localhost';**
      iii. Option 3: > **GRANT ALL PRIVILEGES ON Business.ClientList TO '***username***'@'localhost';**
12. If you would like to review your grant choices you may use the following command
   a. SHOW GRANTS FOR '*username***'@localhost;**

13. What grant options did you choose and why?

Advanced:

1. If you have completed a previous section you may skip to step 7, otherwise continue to step 2.
2. Launch the terminal on your Raspberry Pi
3. Start with the following commands
    a. $ **sudo apt update**
    b. $ **sudo apt upgrade**
4. MariaDB is one of the most suitable databases to use on Raspberry Pi OS. MariaDB is a fork of MySQL. To install MariaDB, enter the following commands
    a. $ **sudo apt install mariadb-server**
    b. $ **sudo mysql_secure_installation**
5. Now you must configure MariaDB for use
    a. $ **sudo mysql -u root -p**
    b. Inherently, you will not have a password and can just press enter to skip through the "Enter password" request.
6. Create a user and set a password that is for only that user
    a. > **CREATE USER** *username***@localhost IDENTIFIED BY '***password***;**
    b. 

    for example: Desired username^^^ Desired password^^^
    c. Confirm the creation of your user using the following command
        i. > **SELECT host, user FROM mysql.user;**
7. Exit MariaDB.
    a. > **exit**
8. Download the provided AcademicRecords.py file.
9. Open AcademicRecords.py in Thonny IDE and change the username and password to your own then resave the file.
10. Navigate to the folder housing the AcademicRecords.py file in your terminal using the **cd** and **ls** commands.
11. Once you are in the folder, run the following command:
    a. $ **python3 AcademicRecords.py**
    b. The above script creates and fills a database named AcademicRecords with three pre-populated tables: GradePointAverage, StudentInfo, and StudentSchools. These scripts rely on one another to receive information.
12. Evaluate the tables and try to delete all of the tables in the correct order. The command to delete a table is as follows:
    a. > **DROP TABLE** *TableName***;**
13. If at any point you want to restart you may run the python script again and it will regenerate everything to its original state.
14. Discussion:
    a. Did you successfully delete the tables in order?
    b. What was your strategy in figuring out the order in which you deleted the tables?

Database Tables and Python Exercise

Learning Objectives/Outcomes

1. Use the MariaDB Python connector to populate a table
2. View contents of a table on MariaDB

Activity Length of Completion: ~25 mins.

Activity Mode: Individual

Activity Description

Now that you have set up MariaDB you will learn how to put it to work by creating a contact book where you can store and view contact information.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers: Physical, Network, Operating System, Application, Data

CIA: Confidentiality, Integrity, Availability

F10P: Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

1. Launch the terminal on your Raspberry Pi
2. Log back in with your user
    a. $ **sudo mysql -u** *username* **-p**
    b. Replace *username* with your chosen username
    c. When prompted, enter your chosen password
3. Enter the following commands to create your Contact Book
    a. **>CREATE DATABASE ContactBook;**
    b. **>USE ContactBook;**
    c. **>CREATE TABLE Contacts (**
    ContactID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(15),
    LastName VARCHAR(20),
    PhoneNumber VARCHAR(11),
    EmailAddress VARCHAR(50));
4. Check to see if you have created your table properly
    a. **>DESCRIBE Contacts;**

b. Your results should look something like this

```
MariaDB [ContactBook]> DESCRIBE Contacts;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| ContactID    | int(11)     | NO   | PRI | NULL    | auto_increment |
| FirstName    | varchar(15) | YES  |     | NULL    |                |
| LastName     | varchar(20) | YES  |     | NULL    |                |
| PhoneNumber  | varchar(11) | YES  |     | NULL    |                |
| EmailAddress | varchar(50) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
5 rows in set (0.003 sec)
```

5. Open Thonny
   a. Hover your mouse over the Raspberry icon in the top left corner
   b. Hover your mouse over the Programming option
   c. Select Thonny Python IDE
6. Enter the following code into a new python program and name it ContactBook.py

```
# Module Imports
import mariadb
import sys

# Connect to the MariaDB Platform
try:
    conn = mariadb.connect(
        user="your_username",     # insert your username
        password="your_password", # insert your password
        host="localhost",
        port=3306,
        database="ContactBook")

except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
    sys.exit(1)

# Get Cursor
cur = conn.cursor()

# Prompt user input
firstName = input("Please enter the contact's first name:
")
lastName = input("Please enter the contact's last name: ")
phoneNumber = input("Please enter the contact's phone
number: ")
emailAddress = input("Please enter the contact's email
address: ")

# Populate table with the collected information
cur.execute(f"INSERT INTO Contacts (FirstName, LastName,
PhoneNumber, EmailAddress) VALUES ('{firstName}',
'{lastName}', '{phoneNumber}', '{emailAddress}')")
```

```
# Commit and close the connection
conn.commit()
conn.close()
```

7. Click “Run” at the top of the IDE. Enter contact information for three random contacts by running the file three times.
8. Return to your terminal. If MariaDB is no longer logged in, repeat the commands on steps 2a, 2b, and 3b.
9. Confirm that your contacts have populated the database using the following command
    a. >**SELECT * FROM Contacts;**
    b. Your results should look something like this

```
MariaDB [ContactBook]> SELECT * FROM Contacts;
+-----------+-----------+----------+-------------+-------------------------+
| ContactID | FirstName | LastName | PhoneNumber | EmailAddress            |
+-----------+-----------+----------+-------------+-------------------------+
|         3 | Derek     | Shepherd | 4234850777  | dereks@greysloan.org    |
|         4 | Alex      | Karev    | 4238992820  | alexk@greysloan.org     |
|         5 | Christina | Yang     | 4238943181  | christinay@greysloan.org |
+-----------+-----------+----------+-------------+-------------------------+
3 rows in set (0.001 sec)
```

10. Discuss the following with the group
    a. Do you feel like this is an efficient way to collect and store information?
    b. What could be done to improve this process?
    c. What else could you use a database like this for?

Images, Filters, and Cybersecurity

Learning Objectives/Outcomes

- Demonstrate how to take images and videos using Python scripts
- Explore implementing OpenCV functions to apply filters to photo and video files
- Explain how images and videos are used for cyber security

Activity Length of Completion: ~30 mins.

Activity Mode: Individual

Activity Description

In this activity, students will practice using the Raspberry Pi camera, PiCamera package, and time package to take images and videos on their Raspberry Pi. Students will also use functions in the OpenCV library to apply filters to these images and videos.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Activity Instructions

Beginner:

1. Open the Terminal and type **sudo raspi-config**.
2. In the menu, choose Option 3, then P1, then yes.
    a. Hit the Escape button to go back to the Terminal.
3. Type the following commands to install OpenCV:
    a. **sudo pip3 install opencv-python**
    b. **sudo apt-get update**
    c. **sudo apt-get install libatlas-base-dev**
    d. **sudo apt-get install libjasper-dev**
    e. **sudo apt-get install libqtgui4**
    f. **sudo apt-get install libqt4-test**
    g. **sudo pip3 install -U numpy**
    h. **sudo reboot**
4. Click on the raspberry icon in the top left corner of the screen. Scroll over **Programming**, then click on **Thonny Python IDE**.

5. Click the **New** button to create a new Python file and save it as **imgName** where **Name** is your name.
6. First, you will need to import the libraries and packages that you will use in the program. Type the following lines and press **Enter**:

```
from picamera import PiCamera
import cv2
import time
```

7. Now, you will use the code that you looked at in the lesson to take a picture. Type the following lines (Note: The code is the text in bold):

```
camera = PiCamera()
#creates camera object to control camera

camera.start_preview()
#starts camera preview before image capture

time.sleep(5)
#tells program to wait 5 seconds before executing the next line

camera.capture("/home/pi/Desktop/newPhoto.jpg")
#takes picture and saves it as newPhoto.jpg on the Desktop

camera.stop_preview()
#stops camera preview
```

   Ensure that you understand how the code above works, then press **Enter**.

8. Load the new picture and resize the picture to better fit the screen. To do this, type the following lines and press **Enter**:

```
source = cv2.imread("/home/pi/Desktop/newPhoto.jpg", -1)

source = cv2.resize(source, (0,0), fx = 0.5, fy = 0.5)
```

   The second size will show the photo at half of its original size.

9. You will probably want to see the image before the filter is applied. Write three lines that each perform one of the following operations:
   - use **`cv2.imshow(`*`windowName`*`,source)`** to show the picture to the user (replace ***`windowName`*** with what you would like to name the preview window. It must be a **`string`** and therefore your name must be in quotes).
   - use **`cv2.waitKey()`** to tell the program to leave the window open until the user presses any key on the keyboard (Hint: use 0 as the parameter)
   - use **`cv2.destroyAllWindows()`** to close the window that holds the image

10. Now, you can finally apply the filter! For this picture, you are going to apply a Gaussian blur. To do this, type the following line of code:

```
updatedSource = cv2.GaussianBlur(source, (5,5),
cv2.BORDER_DEFAULT)
```

The three parameters are:

- the image file path or variable that holds the path
- the size of the matrix used to apply the blur to the picture
- border type

Adjusting the size of the matrix can change the level of blurriness applied to the photo.

11. Before you show the filtered image, save it to a new file. Which of the following lines would save the new image to a file named, "updatedPhoto.jpg," on the Desktop?
    a. `cv2.imread("/home/pi/Desktop/updatedPhoto.jpg", updatedSource)`
    b. `cv2.imwrite("/home/pi/Desktop/updatedPhoto.jpg", updatedSource)`
    c. `cv2.imwrite("/home/pi/Desktop/updatedPhoto.jpg", source)`

    If you answered option b., congratulations, you are correct! Type option b. and press **Enter**.

12. Finally, display the new image to the screen. To do this, copy the three lines that you wrote in step 9 and paste them to the bottom of the file. Replace the second parameter of **`imshow()`** with **`updatedSource`** and save the file.

13. Press the **Run** button to run your program. You should see a preview of your camera, then the captured image. Press any key to close the image window. After this, you should see the image with the blur filter applied to it. Press any key to close the window.

14. Experiment with the numbers in the second parameter of the line that you wrote in step 6. Are any numbers not accepted? What happens when you increase the size of the matrix? What happens when you decrease its size?

15. Answer the questions below and discuss your answers with the other students around you.

    a. How do images and videos play an important role in Cyber Security?
    b. How can the filters that you used today and other filters, such as facial recognition, be used to enhance security?
    c. What other filters found in the OpenCV would you want to try using in the future?

Intermediate:

1. You are going to write a program that takes a video and applies an edge detection filter to it. Navigate to the Thonny Python IDE on your Raspberry Pi.

2. Press the **New** button and save it as **vidName** where **Name** is your name.

3. Import PiCamera, time, and cv2 using the following lines. Then, press **Enter**.

```
from picamera import PiCamera

import cv2
```

```
import time
```

4. Again, use the code that you were given in the lesson to record a 10 second video (The actual code is in bold):

```
camera = PiCamera()
#creates camera object to control camera

camera.start_preview()
#starts camera preview before video capture

camera.start_recording("/home/pi/Desktop/newVid.h264")
#above line starts recording video and saves it as newVid.h264 on
the Desktop

time.sleep(10)
#tells computer to wait 10 seconds before executing the next line

camera.stop_recording()
#stops video recording

camera.stop_preview()
#stops camera preview
```

5. Now, load the video to apply the filter to it. Which of the following lines would work best?
   a. `vid = cv2.VideoCapture("/home/pi/Desktop/newVid.h264")`
   b. `vid = cv2.vidread("/home/pi/Desktop/newVid.h264")`
   c. `vid = cv2.VideoCapture("/home/pi/Desktop/newVid.jpeg")`

   And the correct answer is….option a! Type option a and press **Enter**.

6. As you heard in the lesson, opening and viewing videos using OpenCV is a bit more complicated than the other code that you have looked at. Take a moment to read through the code below and make sure that you understand how it works. Once you understand each line, type the lines in bold into your program and press **Enter**.

```
while(vid.isOpened()):
#loops the following lines of code as long as the VideoCapture
has been initialized correctly

      ret, frame = vid.read()
#reads in the next frame of the video. ret is a boolean to show
whether or not there is another frame and frame is the next
frame.

      if ret == True:
#if there was another frame, the indented lines are executed

            edge = cv2.Canny(frame, 50, 200)
#applies a Canny edge detection filter to the frame. 50 and 200
are the low and high thresholds of filter intensity
```

```
            cv2.imshow('New Video', edge)
#shows the filtered video in a new window

            if cv2.waitKey(1) & 0xFF == ord('x'):
#displays each frame for 1 ms and executes the next line if the
user presses the 'x' key

                break
#breaks out of the while loop and closes the video

      else:
#if there are no frames left, the next line is executed

            break
#breaks out of the loop
```

7. Finally, close the video. Write two lines that each do one the following:
   - uses **`cv2.release()`** to release the video (Remember: vid holds the loaded video)
   - uses **`cv2.destroyAllWindows()`** to close the video window

8. Press the **Run** button to run your program. You should see a preview of the camera's view as the video is recording. Then, a window should pop up showing the video with an edge detection filter.

9. Looking back at the line, "**`edge = cv2.Canny(frame, 50, 200)`**," that you typed in step 15, the last two parameters of the Canny() function control the level of edge detection in the video. Change the values of these last two parameters until you feel that you have a good level of detail without making the video difficult to view. Then, write the values in the space below:

   low threshold:_____
   high threshold:_____

Advanced:

1. First, log in to MariaDB using the username and password that you have used in past activities.
2. Then, create a database called **FilteredImages**. Then, create a table called **Images. Images** should have three columns:
   - **ImgID** (auto increment primary key)
   - **ImgPath** that holds the path to the filtered images (varchar(50))
   - **Description** that holds a description of the image (varchar(255))

3. Once you have created the table, write a Python program called **nightVision.py** that does the following:
   - captures an image using the Pi's camera
   - displays the image to the user
   - applies a "night vision" filter to the new image. Note: The "night vision" effect can be achieved using the following two lines of code:

```
newImg = cv2.bitwise_not(originalImg)
```

```
newImg2 = cv2.cvtColor(newImg, cv2.COLOR_BGR2GRAY)
```

- displays the filtered image to the user
- prompts the user for a description of the image
- adds the filtered image along with the description to the **Images** table that you created in step 2

4. After you run the program, use the command prompt to double check the **Images** table that you created in step 2 to make sure that the images have been correctly stored.
5. Run your program several times to fill your table with a few different images.
6. Answer the following questions:
    a. Can you think of some real life scenarios where you would need to store captured images or videos in a database for security purposes?
    b. If the images stored in the database contain sensitive information (for example, a person's fingerprint or a copy of their social security card), what cybersecurity techniques could you use to make sure that the images could not be accessed by unauthorized users?

Day 5

Congratulations, you have made it to the last day of the GenCyber Summer Camp! Today might feel a little shorter than the rest. You will kick off the last day with a game of **Cyber Bingo** followed by a review of the security layers, CIA, and the ten principles of cybersecurity (**Review of the Week**). After the review, you will hear from Dr. Campbell in her talk, "Women in Cybersecurity." This will be followed by lunch, where you and your parents can chat with some of UTC's career counselors. Then, you will have a chance to add the finishing touches to your group project before you present. Finally, it is presentation time!

Data Table Cyber Bingo

Learning Objectives/Outcomes

Students will become familiar with the common vocabulary that will be used throughout the week by hearing the terms without the definitions.
Students will identify the vocabulary term by hearing the definition.

Game / Activity Length of Completion: 1 hour

Game / Activity Mode: Groups of 8

Game / Activity Description

The students will be placed in groups of 8, and be given a bingo card. Teams will rotate through 3 different bingo stations, each lasting 20 minutes each. The winners will be awarded prizes.

Applicable Cyber Security Principles (highlight all applicable)

Security Layers:	Physical, Network, Operating System, Application, Data
CIA:	Confidentiality, Integrity, Availability
F10P:	Abstraction, Data Hiding, Domain Separation, Layering, Least Privilege, Minimization, Modularity, Process Isolation, Resource Encapsulation, Simplicity

Game / Activity Student Instructions

The facilitator/teacher will provide the definition without revealing their vocabulary words, and you will circle the vocabulary words that match with the definitions. When you think you have collected all the terms either vertically or horizontally in rows or columns, you can shout “bingo.” The instructor will check that you got the correct answer. If so, you win!

Commands List

Terminal

1. **cd**: used to navigate to a specific folder or directory.
2. **df**: lists the amount of disk space used by the various file systems of a device.
    a. **-h**: output of df will be printed in a more human readable format.
3. **free**: displays the amount of used and available space in memory.
    a. **-h**: output of free will be printed in a more human readable format.
    b. **-t**: will print the total of each column of the output of free.
4. **ifconfig**: displays information about a computer's network interface, including IP address, subnet mask, etc.
5. **iwconfig**: displays information about a computer's wireless connection.
6. **ls**: lists all files and directories.
7. **mysql -u root -p**: used to access mysql with root privileges.
8. **mysql -u** *username* **-p**: used to access mysql under a certain user.
9. **nmap**: used to gain more knowledge about the hosts connected to a device's network.
    a. **-sP**: pings multiple hosts at once (ex: nmap -sP 10.161.219.0/24 pings 254 different hosts).
    b. **-O**: provides information about a host's operating system and its version.
    c. **-sV**: displays the service version of services used by host that are associated with open ports.
    d. **-p**: can be used to scan specific ports; ports can be specified by the port number or by the name of the port's service (ex: nmap -p ssh 1.121.163.13)
    e. **-h**: displays all flags available for nmap
    f. **--traceroute --script traceroute-geolocation**: similar to traceroute, but displays the geolocation information of each hop (ex: sudo nmap --traceroute --script traceroute-geolocation 1.192.161.24).
10. **ping**: sends a series of packets to another device to see if that device is online and connected to the same network.
    a. **- c**: can be used along with ping to specify the number of packets to be sent by the command (ex: ping -c 4 will send 4 packets).
11. **ps**: provides information about the active processes of a system.
    a. **-e**: can be used to view all processes on a system
    b. **-A**: produces the same result as ps -e
12. **python/python3**: runs a specified python program.
13. **raspi-config**: used to access the Pi Software Configuration Tool.
14. **sudo**: executes a command in superuser mode.
15. **top**: shows information about processes running on a system in real time.
16. **traceroute**: lists each intermediate device between a packet's source address and destination address.
17. **wget**: used to download files from webservers.
18. **wireshark**: opens Wireshark from the Terminal.



Wireshark

1. **http**: filters Wireshark results by the HTTP protocol.
2. **ip.dst==IP-address**: filters Wireshark results by a specific destination IP address (when using this search function, replace IP-address with the specified destination IP address).
3. **ip.src==IP-address**: filters Wireshark results by a specific source IP address (when using this search function, replace IP-address with the specified source IP address).

Python

1. **+=**: used to shorten the statement x = x + 4 (this statement would be shortened to x += 4).
2. **conn.close()**: closes the MariaDB connection.
3. **conn.commit()**: commits any pending transactions to a database.
4. **destroyAllWindows()**: closes window that is displaying an image to the viewer (ex: cv2.destroyAllWindows()).
5. **else**: section of code that is executed if the if statement above it is false.
    a. format: else:
        <statements to execute if if statement above is false>
6. **for**: section of code that is repeated a specified number of times.
    a. ex. format: for x in range(6):
        <statements to execute>
7. **if**: section of code that is executed if a condition or series of conditions are true.
    a. format: if <condition(s)>:
        <statements to execute if condition is true>
8. **import**: used to gain access to code in another module.
9. **imread()**: loads an image from a specific file (ex: cv2.imread("Desktop/me.jpg", -1)).
10. **imshow()**: displays an image to the user (format: cv2.imshow(title of window, image)).
11. **imwrite()**: saves any changes to an image (format: cv2.imwrite(path to output file, modified source image)).
12. **input**: used to obtain and store input from a user (ex: answer = input("What is your favorite color?")).
13. **print()**: function that prints a specified string or the value of a specified variable to the output (ex: print("This is a test.").
14. **release()**: closes a video (format: variableName.release()).
15. **resize()**: resizes an image (format: cv2.resize(width in pixels, height in pixels)).
16. **VideoCapture()**: loads a video for use (format: cv2.VideoCapture(file path)).
17. **waitKey()**: displays an image to the user for a certain amount of time or, if 0 is used as the parameter, the image will be displayed until the user hits any key (ex: cv2.waitKey(0)).
18. **x = y**: format used to define a variable (ex: ticketSales = 12500).

Databases

1. **CREATE DATABASE <databaseName>;**: creates a new database (when using this command, replace <databaseName> with the desired name of the database).

2. **CREATE TABLE <tableName> (<columnName> <dataType>,…);**: creates a new table with specified columns.
3. **CREATE USER** *username***@localhost IDENTIFIED BY '***password***';**: creates a mysql user with a specified username and password.
4. **DESCRIBE <tableName>;**: displays the setup of a specified table.
5. **exit**: exits mysql.
6. **GRANT ALL PRIVILEGES ON *.* TO '***username***'@'localhost';**: grants all privileges on every database to a user.
7. **GRANT ALL PRIVILEGES ON <databaseName>.<tableName> TO '***username***'@'localhost';**: grants all privileges to a specified user for a specified table.
8. **INSERT INTO TableName (ColumnName(s)) VALUES (Values);**: used to insert data into an existing table.
9. **SELECT * FROM <tableName>;**: used to filter the contents of a table based on a certain criteria (note: using '*' displays everything in a table, but '*' can be replaced with a certain criteria).
10. **SHOW DATABASES;**: shows the databases available to a specified user.
11. **USE <databaseName>;**: tells mysql to enter into a specific database.

Glossary

CIA

1. **availability**: authorized users have access to the systems and the resources they need.
2. **confidentiality**: data, objects and resources are protected from unauthorized viewing and other access.
3. **integrity**: data is protected from unauthorized changes to ensure that it is reliable and correct.

Ten Principles of Cybersecurity

1. **abstraction**: reduce components down to only the essential attributes and exclude any details that the user does not need to know.
2. **data hiding**: hide data that users do not need to access.
3. **domain separation**: separate data and processes into different domains; allows different security rules to be set for the different domains.
4. **layering**: security should be implemented in each layer of the system.
5. **least privilege**: assign each user the least amount of privilege that they need.
6. **minimization**: keep elements as small and simple as possible while still meeting the user's needs.
7. **modularity**: separate a program into interchangeable modules.
8. **process isolation**: separate each running process from one another to make sure that they don't interfere with each other.
9. **resource encapsulation**: place data and functions that act on that data into one component.
10. **simplicity**: ensure that security mechanisms are easy to follow and maintain.

Other Terms

1. **ARP**: short for Address Resolution Protocol; protocol used to translate IP addresses to MAC addresses for hosts that are on the same subnet.
2. **brute force attack**: trying every possible combination of words, numbers, and symbols to try and guess a user's password.
3. **dictionary attack**: trying to guess a user's password by entering words from a list.
4. **hash function**: function that takes in an input such as a string and produces a fixed-length output (known as the hash value). No two inputs should produce the same hash value.
5. **HTTP**: short for HyperText Transfer Protocol; protocol used to request and send Web pages between Web servers and clients.
6. **IP**: short for Internet Protocol; specifies format of packets sent between routers and hosts.
7. **nonvolatile**: memory that can retain information even if power is lost.
8. **paging**: memory management technique in which data is stored in secondary storage in fixed-sized blocks.
9. **phishing**: when an attacker sends a fake message to users in order to trick them into giving the attacker sensitive information (ex: an email saying that your bank account has been compromised and you have to click the link to re-enter your bank account information).

10. **rainbow table attack**: when an attacker tries to gain access by matching a password's hash to a hash stored in a dictionary of plaintext passwords and their corresponding hashes.
11. **secondary memory**: where data can be stored on a long-term basis; includes hard disk drives, floppy disks, etc.
12. **segmentation**: memory management technique in which data is stored in secondary storage in variable-sized blocks.
13. **shoulder surfing**: looking over a user's shoulder as they are using a computer to retrieve confidential information.
14. **social engineering**: manipulating people into giving up confidential information (phishing is a form of social engineering).
15. **SSH**: short for Secure Shell Protocol; protocol that allows two computers to securely communicate over an unsecure network.
16. **TCP**: short for Transmission Control Protocol; protocol used to transport messages across the application layer that provides guaranteed delivery and flow control.
17. **thread**: unit within a process that can be scheduled for execution; one process can have several threads.
18. **UDP**: short for User Datagram Protocol; protocol used to transport messages across the application layer that does not provide guaranteed delivery or flow control.