PLTW COMPUTER SCIENCE

Activity 1.2.2

Algorithms and APIs: Hack Attack

goals

- Identify ethical considerations that impact all users
- Learn how application programming interfaces (API) connect different computing devices
- Apply algorithms to automate attempts to discover a password
- Develop an app as part of a pair programming collaboration

description of app

Add to an existing app using APIs and algorithms to automate a brute force attack that will attempt to identify another team's numeric password.

Essential Questions

1. What does it mean to "hack"?
2. How can algorithms automate processes for both good and bad purposes?
3. Why are APIs such an essential tool in computer science today?

essential Concepts

- Cybersecurity
- Algorithms to Automate Processes
- Application Programming Interfaces (API)
- Incrementing Counts

Resources

Shell122HackAttack app

Flowchart Infographic

API Reference Sheet

Application Programming Interface

Application Programming Interfaces (API) connect many different types of devices that access the internet. An API specifies how software components should interact, and an understanding of this type of program is essential for development of web applications. To gain insights into cybersecurity issues, you will look at how information is exchanged on the internet through API endpoints on a server that stores data.

When designing applications to use an API, developers need to test the security of data. To test the security of an API endpoint, you will design and use a brute-force hacking algorithm with the explicit permission of the other teams in your class and your teacher. A brute-force attack will try every possible password combination on another team's app to gain their API password. This type of hacking to find and remove weaknesses is referred to as "white hat hacking".

PLTW DEVELOPER'S JOURNAL

Record information about when hacking is acceptable and when it is not. Hacking is using an electronic device to get access to data in a system beyond what your permissions normally allow. In this activity, you have permission to test specific API endpoints created in your class to see whether you can gain their passwords. Research and discuss the following:

- Would it be okay to…
  - Work on this outside of the class period?
  - Try to gain the password of an API endpoint that was not created during your class period?
  - Use an algorithm you developed for this activity in a different app, website, or class?
  - Tell people you are a hacker?
  - Test the security of a site without the permission of the owners of that site?
  - Use what you have learned to protect your development ideas from hacking?

Power in Social Apps

Many social apps and games use servers that offer a web API endpoint for communication with a client app. A user's device can send instructions to the server and get data from the server by sending requests to various API endpoints. These endpoints are specific URL locations that allow a device to interact with a web server to modify or collect data from the API.

Other programs may be able to access these API endpoints, so the data passed is not always secure. Whenever you post or reply on social media, or follow a result from a browser search, you are using APIs and **algorithms**. All these things then exist on the internet and may be seen by others, including future employers, teachers, and family.

API Hack Attack

PLTW provides a web service to test APIs at **https://ics-api.pltw.org** . The site is designed for learning about APIs and for use only within the scope of educational learning under a PLTW teacher.

You are going to set up a Hurl.it account on a PLTW server and practice using the API endpoint of that account. You will first access the API through a browser on a desktop or laptop computer, and then you will access that endpoint through an app on your Android device.

You will design an algorithm for an app to gain another team's server password. Gaining someone's password using algorithms is considered hacking and should not be done outside this specific context.

Hacking is using an electronic device to get access to data in a system beyond what your permissions normally allow. In this case, your teacher is giving you permission to test the security of a specific API, to gain access to only the accounts of your classmates.

Important: Using algorithms to gain information such as passwords without explicit permission is unethical and illegal.

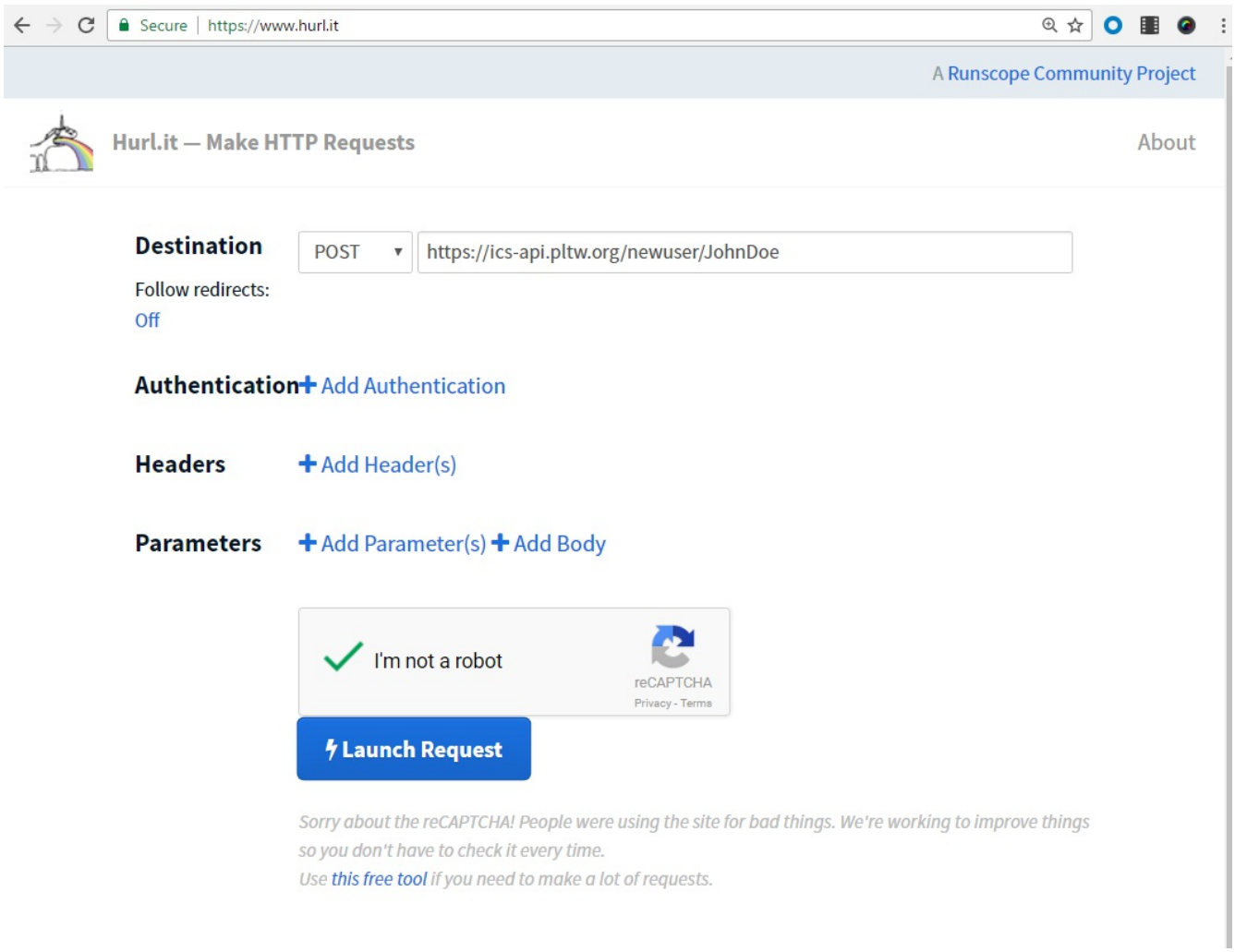
Setting Up an API

Hurl.it is a web service that lets you access API endpoints directly. You can use these endpoints to add the functionality to access information in servers. Hurl.it provides a way to test the processes for the programs you are writing by making and handling HTTP requests at these endpoints.

When you create your account using the provided link, all the data that is transferred through the web is stored on a server at PLTW. This lets you test posting and retrieving data in your apps. "GET" and "POST" are the two most common methods for making HTTP requests. In the first example you see, you will post to the server (POST), retrieve information from the server (GET), and then vote (through POST) to "like" posts that are made.

At the https://ics-api.pltw.org web service, PLTW created an API endpoint for you to learn and practice with. The endpoints to get, post, and vote are documented in a Reference Sheet you can access using the QR code on your PLTW Developer's Journal cover. Using the provided API, you can create your own set of endpoints under a username you create.

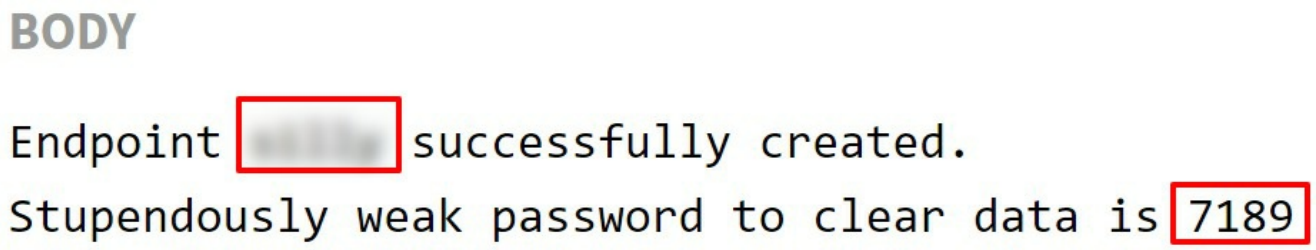
1. Form teams as directed by your teacher.
2. Open a browser window and navigate to: **https://www.hurl.it**
3. Set the Destination to POST and enter: "https://ics-api.pltw.org/newuser"
4. Create a professional API username for your team. Replace the username "JohnDoe" in the sample address "https://ics-api.pltw.org/newuser/JohnDoe" with your username.



5. You will need to verify that you are not a robot by responding appropriately to the prompts.
6. Select **Launch Request**.

   Note: You may have to try several usernames before you find an available username.

7. Check the bottom of the web page to verify that the API user endpoint was successfully created.
   If it was, you will see the endpoint account name and a four-digit password that was generated for your API:



PLTW DEVELOPER'S JOURNAL Record the username endpoint and the four-digit password.

Important: You can POST messages, see messages, and vote on those messages without a password. However, you cannot clear all data from your team's API endpoint without the correct password.

What Does an API do?

The server where you access the API endpoints stores a list of strings for each account. Your class will practice *posting*, *incremental voting*, and *getting* information from API endpoints, just as web browsers do. Pay attention to any patterns you see, because this information will help you later in knowing how to access data through an API.

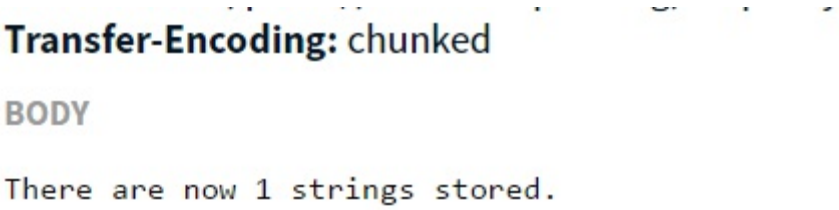
8. Share your team's URL username with the class as instructed by your teacher.

   Important: The API web service stores and presents information that is input or retrieved for individual user accounts. Each time an app uses the POST method, it adds or changes information on the account's server.

You already added *a user* with POST; now you will add *a string* to the specific user endpoint that you created. After using a few POST methods, you will get information from the server using the GET method to retrieve and present the information to a user.

9. Use the POST method.
   - Post a string containing something you have learned in CSE to another team's URL.
   - Use the example below to help with how to use the POST method. A common error is to leave "newuser" in the URL.

10. To verify that the POST request was made, check the bottom of the web page. The server should send a response to let you know whether you successfully connected with the endpoint to post a message.



POST and GET

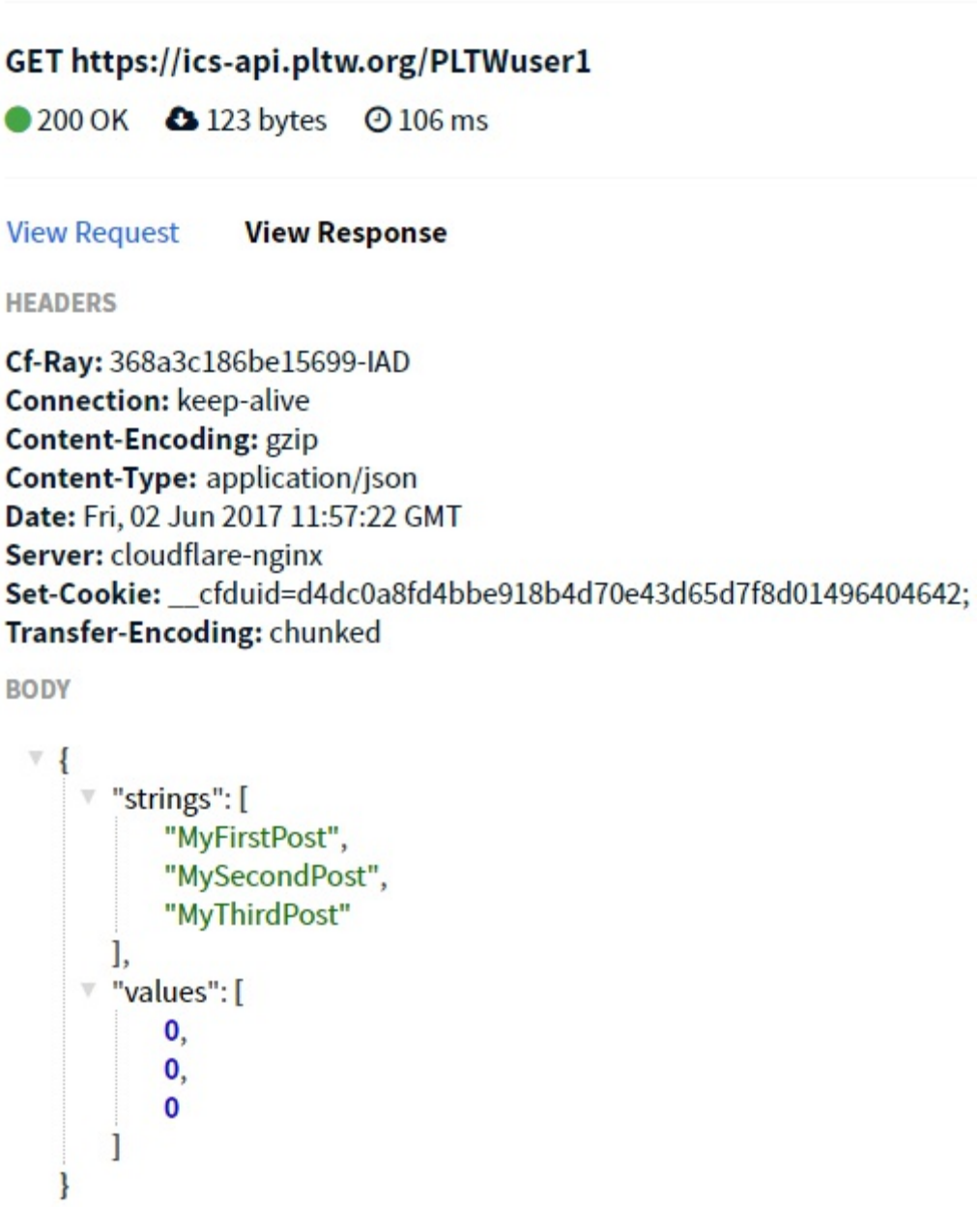
Refer to your downloadable resources for this material. Interactive content may not be	Refer to your downloadable resources for this material. Interactive content may not be

available in the PDF edition of this course.	available in the PDF edition of this course.

To see what the other teams have posted to your team's URL, make a request for all the strings that have been submitted. Instead of posting a new user or posting a string, you will retrieve the information that is stored there. Therefore, you will need to change your method from POST to GET, so the API will give you all the information stored at the user endpoint.

11. Make a GET request of another team's API server endpoint.

    If the GET request is successful, you will see information about where the information is coming from and the strings and values stored at that endpoint. It will look something like this:



Voting Post

As shown in the previous image, when you get the information stored at the user endpoint, you see the statements posted by others and that each string has a value. These values are in the same order that you see the comments.

The values can be incremented, much as you did with incrementing variables in loops to increase the value of a variable. This sort of incrementing is similar to how people add “likes” to things in social media. They identify a string and increment the value of that string post by liking it, creating a new total number of likes on a single string post.

All teams should have statements on their API user endpoint from other teams in the class. Each person in the team will vote on his or her favorite learning statement by incrementing the value

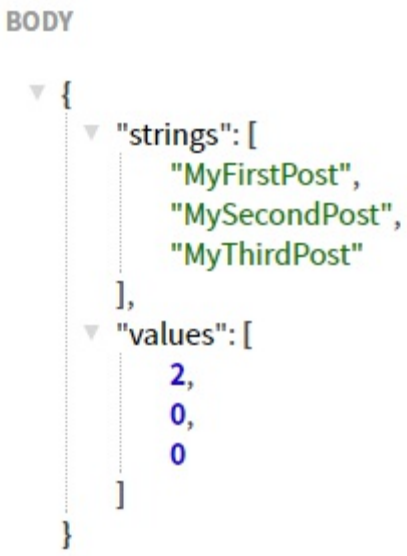
associated with that string.

Using the Index to Vote

Look at all the posts and count from the top down. What place is your favorite statement in? That is the index number of that statement. In Hurl.it, the index starts at 1. The “values” are the count for that post.

12. Use the example above to help you increment by 1 your favorite string post. You can do this by changing the number at the end of the https location to the index number of your favorite string post.

    If your increment is successful, you will see a message like this:



13. When time is up for voting, get all the strings and their corresponding values from your team’s API. Which string got the most votes?

14. Now it is time to clear your API endpoint. Use the example below as a guide, substituting your own username and password. Because you are modifying the data at the endpoint, use the POST method instead of the GET method in the Destination box.

    If the password was successfully reset, you receive a message like:

```
BODY

Data for endpoint JohnDoe has been cleared.
```

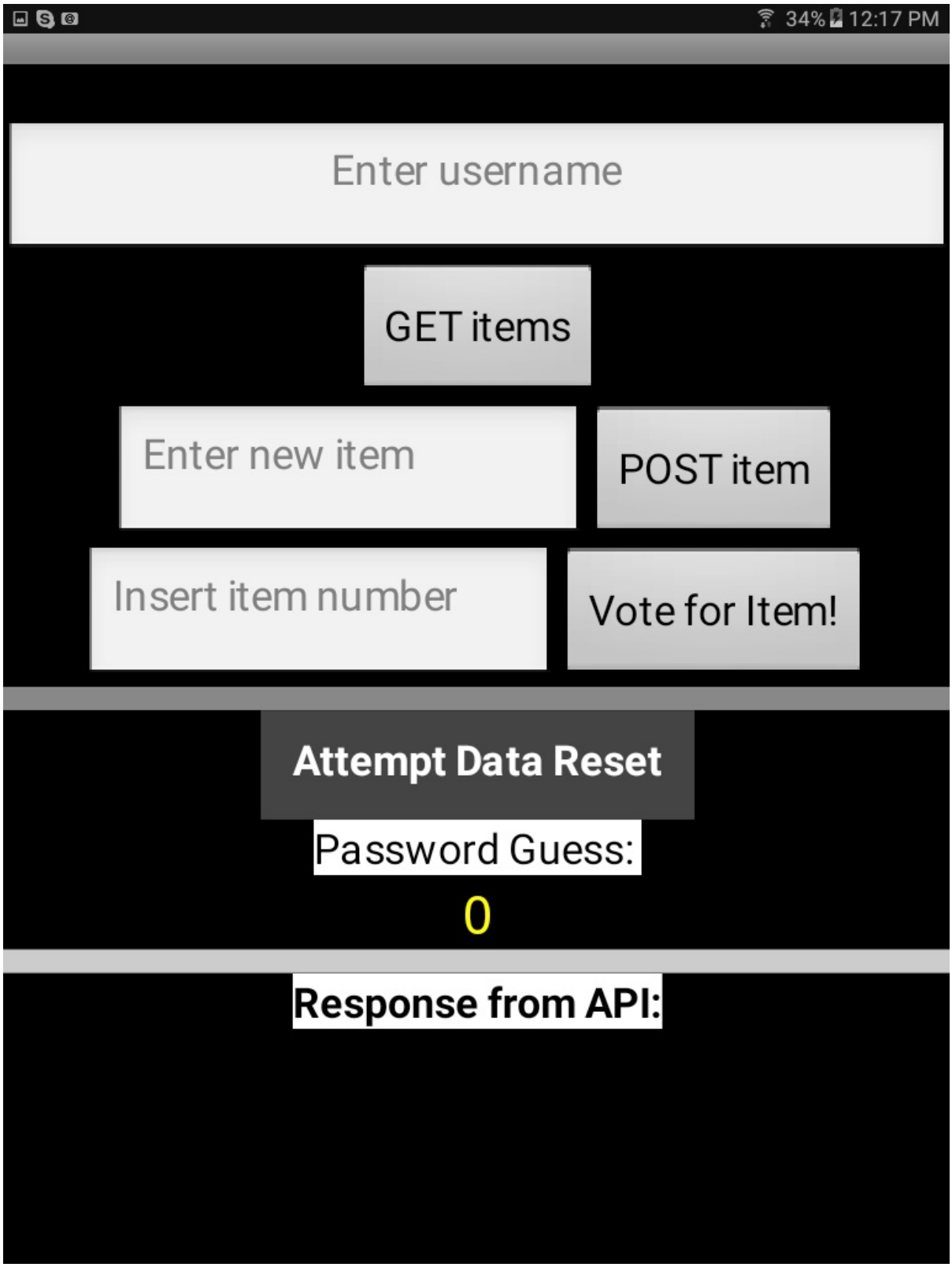
Starter Hack Attack App

Now that you have spent some time posting, incrementing, and getting used to an API, you are going to take what you have learned to create an algorithm that will do a brute-force hack on another team's API endpoint to discover their password.

A brute-force hack is an algorithm that tries every possible combination until it retrieves a working password. As you create the algorithm, remember that you must get the other team's password, not just reset the endpoint.

15. Download the **Hack Attack app**, which is already programmed to post strings, get strings, clear the endpoint, and increment the voting as you have already done in the web browser.
16. Use your endpoint in the Shell122*HackAttack*.*aia* app to post, get, increment, and reset, just as you did with the API in a web browser.

    You should see that it does not matter whether you are using a web browser or an app on a mobile device, POST, GET, and incrementing through POST methods work the same way.



AppChat Backlog

Some parts of the backlog have already been completed, so you are only addressing the last few as a team.

17. As you look through the blocks provided in the app, determine which of the following backlog items have been completed and which ones remain for your team.
    - ☐ Identify a specific API endpoint
    - ☐ POST to a specific API endpoint
    - ☐ Get information from a specific API endpoint
    - ☐ Vote on a certain string stored at a specific API endpoint
    - ☐ Input a number to clear all the information stored at a specific API endpoint
    - ☐ Generate numbers in a pattern automatically
    - ☐ Determine whether a particular number is correct based on the API feedback
    - ☐ Display a number based on the feedback of the API endpoint

The blocks in the following image are what you will see in AppChat. You do not need to modify these blocks at all. The example is there just to help you see how App Inventor can accomplish the same tasks as your web browser.

> Refer to your downloadable resources for this material. Interactive content may not be available in the PDF edition of this course.

These blocks access a user API endpoint, get all the strings from that user endpoint, POST new strings, vote on strings, and then reset the API endpoint for that user name.

For each of these, a user clicks a button, and the computer concatenates the URL, the username, and the unique text that tells the API endpoint what to do. As with the API in the web browser, a response is sent back to the app that the .*GotText* event handler returns to the user in the *APIResponseValue*.

PLTW DEVELOPER'S JOURNAL Review what is provided in the Hack-Attack app. Take notes to help remind you about how the app accesses the API through POST and GET methods.

18. Connect your Android device to the shell app and enter your API username on the tablet.
19. Try posting, voting, and clearing.
20. As you test the app, discuss the following questions with your team:
    - What responses come back from the API as you post, get, increment, and reset the password in the app? Have you seen those responses before?
    - When you reset your team's password in the web browser, what response do you get back from the API?
    - If you were to attempt to guess another team's password, without an algorithm, what would you systematically do to find the other team's password?
    - How long would this process take?
    - How would you know when you got the correct password for the other team's site/app?

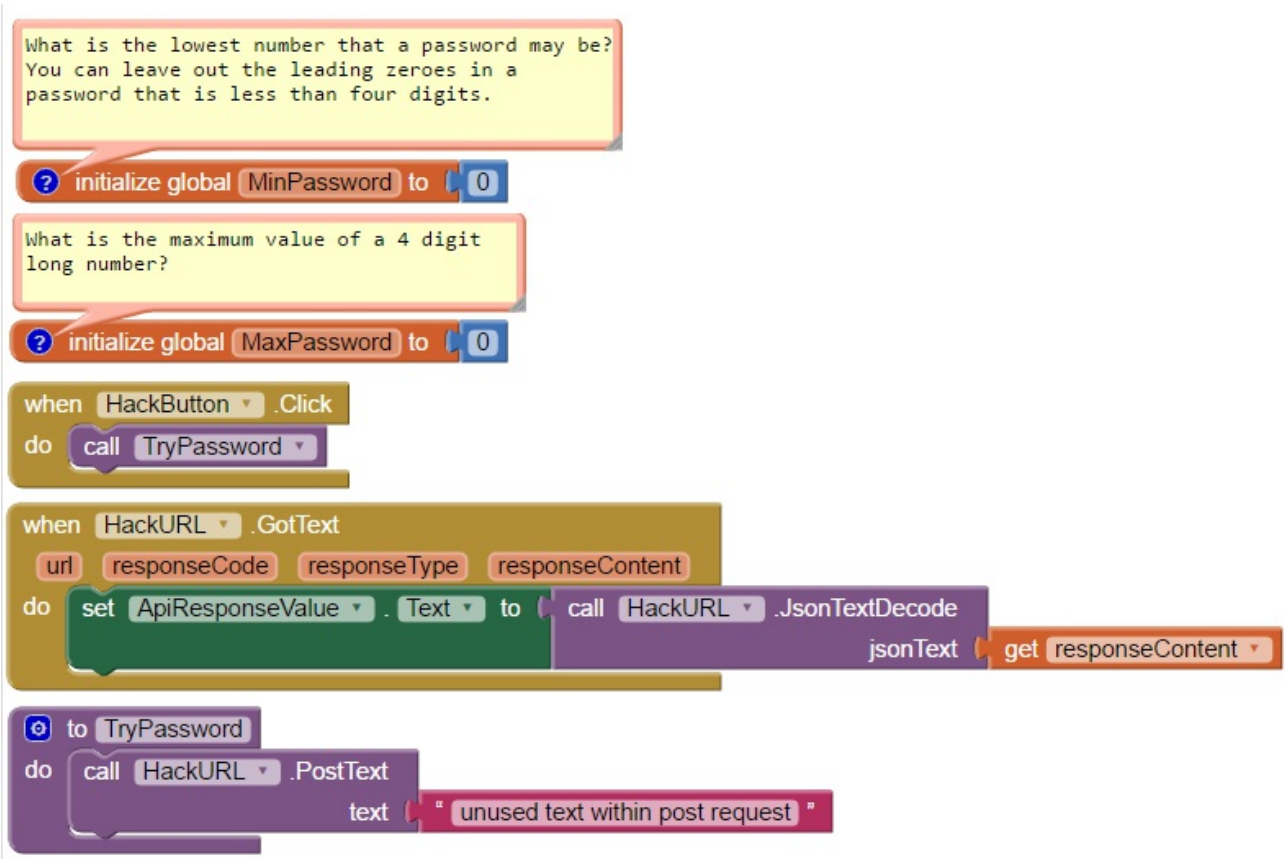
Developing an Algorithm

PLTW DEVELOPER'S JOURNAL Flowchart, diagram, or outline in natural language how your team could automate what the program should do to get the password, including the condition when you have discovered the password and the program can stop.

Note: Review the Flowchart Infographic for more information about flowcharting.

The algorithm you develop in the block-based programming viewer should systematically try all the passwords to get the other team's password.

21. Modify and add to only this portion of the code, based on the strategies you and your partner brainstorm.



22. Use the steps below to help make your algorithm. Review the use of conditional statements, incrementing variables, and loops to help create this app. Be sure to check errors if they arise.
    - Modify the global variables *MinPassword* and *MaxPassword* for the range of password possibilities. (Keep in mind you have more than one person and device on your team.)
    - Modify the *TryPassword* procedure block. It should have the procedures for accessing the API. (Compare to the code used to reset your own known password in the app.)
    - Modify the *HackURL.GotText* based on the other parts of the app as appropriate for what you are doing in the *TryPassword* procedure.
    - Use the *PasswordNumberGuess* to help you see variables that you are testing.
23. When you are successful, provide your block-based code along with the visual evidence of

resetting another team's password through the known password box in the user interface of your team's app.

Ethical Hacking

Sometimes it is important for companies to try to hack their own software to see where vulnerabilities exist. It is not okay for them to hack their competition to see where their vulnerabilities exist. Similarly, hacking outside of this activity, or of any user endpoints not specified by your teacher, is not okay. If you do not have permission for a specific hack, it is illegal and unethical.

24. Write an algorithm that will hack into the other team's API endpoint and get their password.

    Important: Clearing the other team's account is not enough. You must get the password for your team to win. Be aware that other teams might be listening as your team discusses algorithmic strategies.

    Hack Guidelines

    - Each student may make no more than 3000 total requests per day to the combination of all endpoints on the API.
        - You will need to show your final app(s) to your teacher to show that you did not violate this rule of the game.
    - Requests to the *reset* endpoint may only be made for a user your class created through the *newuser* endpoint during your class period.
    - Do not try this on other accounts; you have not been given permission to access any other endpoints.
    - Do not use the app you develop on anything other than the specific API endpoints directed by your teacher and only under your teacher's supervision.

Conclusion

1. Is the activity you just participated in ethical? Provide support for your claim.
2. Describe the exploit or idea that allowed you to identify another team's password. How could you modify your password to prevent a team from capturing your password so easily?

3. How did you interpret and respond to the **essential questions**? Capture your thoughts for future conversations.