Lesson 4: Entry Door Security Breach (Phase 4)

Paragraph 1: The First Security Breach (User Input)
It started with a rumor. Noah wasn't paying much attention at first—just grabbing his books from his locker—when he overheard a group of students whispering excitedly. "Dude, I swear, I just swiped my ID, and the teacher's door unlocked." Another student scoffed. "No way. It probably just glitched." The first one insisted, "I'm telling you, it worked! A bunch of us got in this morning."

Noah's brow furrowed. That wasn't supposed to happen. The entry doors were restricted to teachers and staff—students weren't even supposed to have the option to scan their IDs. Yet, somehow, a random group of students had walked right through. Minutes later, a hall monitor stopped a few of them near the faculty lounge, demanding to know how they had gotten in. One of them shrugged. "The door just opened. We didn't do anything."

But Noah wasn't buying it. If the system was malfunctioning, it would have been a school-wide issue. But from what he was hearing, only certain students were getting through. Something was off, and Noah intended to find out what.

Paragraph 2: The Access Logs Don't Add Up (String Variable)
Noah didn't waste time. Between classes, he slipped into the computer lab and pulled up the school's access control system. If students were getting through a restricted door, the logs would show who scanned in and when. He expected to see only teacher and staff names. Instead, what he found made his stomach twist. There were student names mixed in with the staff. That wasn't possible.

Each time someone scanned their ID, the system recorded their name and role as a string variable—a piece of text data that classified them as Student, Teacher, or Admin. But in today's logs, some entries didn't match real student accounts. Instead of "Student," their roles were blank or labeled as "Authorized." Noah leaned closer. Someone had altered how the system identified users. Normally, the system would store an entry like this:

```python
userRole = "Student"
```

But these logs showed something different—the role had been changed:

```python
userRole = "Authorized"
```

Noah’s heart pounded. Someone had bypassed the student restrictions. And he had a good guess who was behind it.

Paragraph 3: Olivia Puts Noah on the Spot (Conditional Statement)
Noah was still scanning through the access logs, trying to figure out how someone had altered the system, when a voice interrupted his thoughts. “Noah Anderson, report to the office.” He froze. The office? Why?

When he arrived, the principal, Mr. Lawson, was standing with Olivia, who looked way too pleased with herself. “Mr. Anderson,” the principal said, rubbing his temples. “Your friend here tells me you’re good with computers. We’re having an issue with the entry doors allowing students through when they shouldn’t. IT is busy, and we need this fixed before it causes bigger problems.” Noah shot Olivia a look. She gave a sweet, innocent smile. “You’re always fixing things,” she said, nudging him. “Might as well make it official.”

Noah sighed, stepping toward the computer. He wasn’t sure if he should be annoyed or flattered. As he clicked into the system, his suspicions were confirmed. Someone had tampered with the conditional statements controlling access. In programming, a conditional statement determines what happens based on a rule. The entry doors should have been locked unless the user’s role was "Teacher" or "Admin." But the code had been altered:

```python
if userRole == "Authorized":

    unlockDoor()
```

Instead of checking if the person was actually a teacher, it now granted access to anyone marked as "Authorized"—a category that shouldn’t even exist. Noah exhaled. Glitch was getting better at covering his tracks.

But before he could dig deeper, Mr. Lawson clapped him on the shoulder. “Just make sure this doesn’t happen again. Faculty has a meeting later, and we don’t want students wandering in.” Olivia smirked. “Also, they ordered pizza, and they’re keeping it locked away so students don’t raid it.” Noah snorted. “So it’s about pizza security now?” “Hey,” Olivia shrugged. “Teachers take their food seriously.”

Noah shook his head. This wasn’t just about pizza—it was about control. And Glitch had too much of it.

Paragraph 4: Tracking the Manipulated Access (For Loop & Counter Variable)
Fixing the code was one thing—but Noah needed to know how many times this had happened and who was involved. If Glitch had changed the system, there had to be a pattern. He opened the access logs again and started writing a for loop to scan through the entries. A for loop is a command that repeats a set of instructions a specific number of times. In this case, Noah used it to search through every log entry for the past week, looking for students marked as "Authorized."

```python
for entry in accessLogs:

    if entry.userRole == "Authorized":

        print(entry.userName, entry.timeStamp)
```

A list of names popped up on the screen. Noah frowned. This wasn't random. The same names appeared over and over—certain students had been scanning in multiple times per day.To confirm the pattern, Noah added a counter variable, which would track how many times each student gained unauthorized access.

```python
accessCount = {}


for entry in accessLogs:

    if entry.userRole == "Authorized":

        if entry.userName not in accessCount:

            accessCount[entry.userName] = 1

        else:

            accessCount[entry.userName] += 1
```

The numbers were worse than he expected. Some students had entered through teacher-only doors over 20 times in the past two days.

Olivia peeked over his shoulder. "That's… not good," she said. "No kidding." Noah exhaled. "Someone gave them access, and they figured it out fast." Olivia pointed at one name in the list. "Wait. I know him. He was bragging yesterday about sneaking into the teachers' lounge."

Noah looked back at the screen, his mind working. If they could find out how the students were learning about the hack, they might be able to trace it back to Glitch.

For the first time, they had a lead.