Secure & Intelligent Smartphone App

Dr. John Yoon
Mercy College
Situation

- Mobile devices as
  - Emerging user-interfaces to remote big data servers
  - More users using mobile devices to access servers
  - More attacks on mobile devices

- Smartphone apps are portable
  - Even so, authentication and authorization should not be portable

- Data transmission increases
  - If any deep learning is needed
  - Data analytics may be available on remote servers
Data transmission between mobile devices and remote servers

- User authentication
  - Determined on a local device only, or from a remote server?

- Apps authorization
  - Made by a local device decision, by a remote server?

- Outcomes of mobile apps
  - Stored on a local device, or posted to remote servers?

- Logs of app activities
  - Once attacked, one of the important evidence pieces is missing
  - Saved on local device, or posted to remote servers?
Data Used in Mobile Apps

- Data available in the local device, e.g., SQLite3
- Data remotely transmitted from webserver
  - Back-end database, MySQL
  - In the middle, Apache server running PHP
Data Securely Used in Mobile Apps

- Data available in the local device, e.g., SQLite3
  - Access Control on UI
- Data remotely transmitted from webserver
  - Control on JSON
Concept Proving

Quest Pool
Login ID
Password
Role as defined, if needed
SIGN IN

Pick a Subject
Admin  Self Assess  Report

Which of the following is not true for Python functions?

click here to choose your answer:

Python by Questions
00: 08: 36

Grading
More Questions
Video Lectures  Web Lectures

print() is a built-in function
a function can be defined
a function can be called to executed
a function can take no parameter
a function can take several parameter
a function can return no value
a function can return several values
Selecting Questions

Questions
- Selection by combining 1) Randomization, 2) Progressive promotion, and 3) User’s level and performance

User’s performance
- Recorded and stored in another database

Selections
- Ex) for three levels of difficulties in chapters, 1, 2, 3, ..., gradual upgrading and progressing lessons.
- Ex) Consider the below:

<table>
<thead>
<tr>
<th>Level</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chap 1</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>Chap 1</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>A</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>Chap 1</td>
<td></td>
</tr>
<tr>
<td>Chap 2</td>
<td></td>
</tr>
<tr>
<td>Chap 3</td>
<td></td>
</tr>
<tr>
<td>Chat 4</td>
<td></td>
</tr>
<tr>
<td>Chap 5</td>
<td></td>
</tr>
</tbody>
</table>

sequence
<table>
<thead>
<tr>
<th>Level</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chap 1</td>
</tr>
<tr>
<td>Chap 2</td>
</tr>
<tr>
<td>Chap 3</td>
</tr>
<tr>
<td>Chat 4</td>
</tr>
<tr>
<td>Chap 5</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Level</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Chap 1</td>
</tr>
<tr>
<td>Level</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Sequence</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
</tbody>
</table>
Access Control

- Determined by user’s credentials (provided from the remote database)

- Authentication
  - User’s login and password
  - UI button to enable/disable

- Authorization
  - Authorize a user to access an App of questions
  - Ex) Authorize John to access the questions on Python
  - UI options to reconstruct for a user to pick
Access Control on UI

Using **Interface Builder** Outlet/Action in Swift

```swift
@IBOutlet var btnDBAccess: UIButton!
@IBAction func btnDBAccess(_ sender: Any) {
    if idPassVerificier(id, password) {
        btnDBAccess.isEnabled = true
    } else {
        btnDBAccess.isEnabled = false
    }
}
```
Student management from a class teacher

List of authorized classes

Class performance analytics

Login
Password
Role

Quest Pool

Admin
Self Assess
Report
UI Pickerview Created on Users
Addition Study from Youtube

Python

Python by Questions

What is the top container when tkinter is imported?

tkinter.Tk()

Grading

More Questions

Video Lectures

Web Lectures

Reference:
- Conditional Statement
  https://www.youtube.com/watch?v=TNDUFws8ic&tl=38s

- Graphical User Interfaces
  https://www.youtube.com/watch?v=SiV9X-0jw-M&tl=222s
What is the top container when tkinter is imported?

```python
tkinter.Tk()
```
Traditional Approach

- Cryptographic Approach
  - Encrypt JSON data
  - Works well for each single transaction
  - Once data compromised, data confidentiality is losing and spoofing is still possible
Data Spoofing is Worse in

- Data associated with long-duration data transmission
  - Multiple data download and upload
  - Multiple mobile devices and servers
  - Data is linearly incremented

- Proposed Approach
  - Block-chaining of JSON
  - Hashing blocks in chaining
Databases

AppJSONreceiver.swift

AppJSONsender.php

shh = hash(JSON1+KEY)

WEB SERVER

(JSON1, JSON2, ..., JSON_n)

Data to send

(Cloud) Databases

AppJSONreceiver.swift

Data to send

WEB SERVER

(JSON1, JSON2, ..., JSON_n)

AppJSONsender.php

shh = hash(JSON1+KEY)
CHECK: \( \text{shh} \neq \text{hash} (\text{JSON1+KEY}) \)

\( \text{chh} = \text{hash} (\text{shh+"ACK"}) \)
**CHECK:** \( \text{shh} \neq \text{hash(JSON1+KEY)} \)

\( \text{chh} = \text{hash}(\text{shh}+"\text{ACK}\") \)

**CHECK:** \( \text{chh} \neq \text{hash}(\text{shh}+"\text{ACK}\") \)

\( \text{shh2} = \text{hash}(\text{JSON2}+ \text{chh}) \)
**CHECK:** $\text{shh} = \text{hash}(\text{JSON1}+\text{KEY})$

$\text{chh} = \text{hash}($\text{shh}"ACK")$

**CHECK:** $\text{chh} = \text{hash}($\text{shh}+"ACK")$

$\text{shh2} = \text{hash}(\text{JSON2}+\text{chh})$

**CHECK:** $\text{shh2} = \text{hash}(\text{JSON2}+\text{chh})$

$\text{chh2} = \text{hash}($\text{shh2}"ACK")$

**CHECK:** $\text{shh2} = \text{hash}(\text{JSON2}+\text{chh})$

Data to send
Concluding Remarks

- Secure mobile apps can be implemented
- Authentication and authorization are controlled and managed by a backend server
- Data can be transferred securely between mobile devices and servers

- Cybersecurity classes apps are developed
  - Question-driven student learning
  - Questing-driven student performance analysis