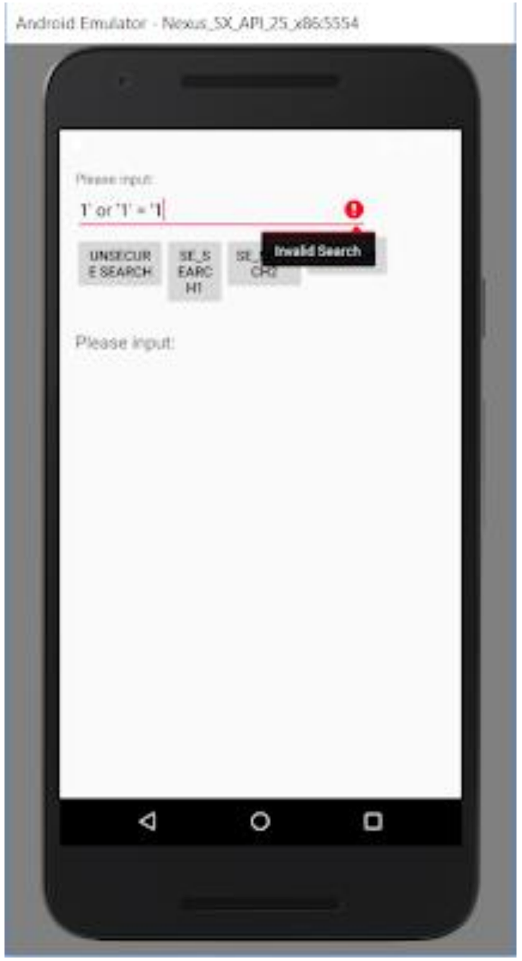
Secure Mobile Software Development Modules

Introduction

- Many Android smartphones compromised because users download malicious software disguised as legitimate apps
- Malware vulnerabilities can lead to:
  - Stolen credit card numbers, financial loss
  - Stealing user's contacts, confidential information
- Frequently, unsafe programming practices by software developers expose vulnerabilities and back doors that hackers/malware can exploit
- Examples:
  - Attacker can send invalid input to your app, causing confidential information leakage

Secure Mobile Software Development (SMSD)

- **Goal:** Teach mobile (Android) developers about backdoors, reduce vulnerabilities in shipped code
- Hackers generally attack Android devices more than iOS
- SMSD: Android Plug-In my collaborators and I have developed:
  - Alerts Android coder about vulnerabilities in their code
  - Hands-on, engaging labs to instill concepts, principles


Android Emulator - Nexus_5X_API_25_x86:5554

Please input:

1' or '1' = '1

Invalid Search

SMSD: 8 Modules

- **M0: Getting started**
- **M1: Data sanitization for input validation**
- **M2: Data sanitization for output encoding**
- M3: SQL injections
- **M4: Data protection**
- M5: Secure inter-process communication (IPC)
- M6: Secure mobile databases
- M7: Unintended data leakage
- M8: Access control

- **Lab:** Go through M0, M1, M2 and M4 + fill out a survey
- **My thought process:** SMSD modules more useful for you, easier than research papers

M1: Data Sanitization for Input Validation

- Malicious inputs can:
  - Leak confidential information to the attacker
  - Lead to system crashes
  - Cause malicious database manipulation, corrupt database
- Countermeasure strategies:
  - **White list valid inputs:**
    1. Use regular expression to check whether an input is of the authorized type, rejects everything else
       - E.g. if a date is expected, Regular expression determines if input is valid date
    2. If input is from a fixed set of limited options, use a drop-down menu or radio button
  - **Black list invalid inputs:**
    1. Build blacklist of known common attack characters and patterns (‘, <script>)
    2. Compare input to blacklist entries