Learning Database Security with Hands-on Mobile Labs

Kai Qian[1],Dan Lo[1],Hossain Shahriar[2],Lei Li[2]
[1] Computer Science Department
[2] Information Technology Department
Kennesaw State University
Kennesaw, GA, USA
kqian,dlo2,hshahria,lli13@kennesaw.edu

Fan Wu
Computer Science Department
Tuskegee University
Tuskegee, AL, USA
fwu@tuskegee.edu

Prabir Bhattacharya
Computer Science Department
Morgan State University
Baltimore, MD, USA
prabir.bhattacharya@morgan.edu

***Abstract*—As mobile computing is becoming more and more popular, the security threats to mobile applications are simultaneously increasing explosively. Most malicious activities hack the user's private information, such as contact and location information, hijack the user's transactions and communications, and exploit the confidential enterprise data stored in mobile databases or in cache on mobile devices. Database security is one of the most important security areas to be addressed. Many schools are integrating database security topics into database and cybersecurity education. This paper addresses the needs for pedagogical learning materials for database security education and the challenges of building database security capacity through effective, engaging, and investigative learning approaches, through transferrable and integratable mobile-based learning modules with hands-on companion labs based on the OWASP recommendations, such as input validation, data encryption, data sharing, auditing, and others. The primary goal of this learning approach is to create a motivating learning environment that encourages and engages all students in database security concepts and practices learning. The preliminary feedback from students was positive. Students gained hands-on real world learning experiences on Mobile Database Security (MDS) with Android mobile devices, which also greatly promoted students' self-efficacy and confidence in their mobile security learning.**

KeyWords—Mobile Security, Computer science, learning, Database security

I. INTRODUCTION

More and more institutions started offering cybersecurity courses on information security. The Information Assurance and Security (IAS) Knowledge Area (KA) has been added to the Body of Knowledge in recognition of the world's reliance on information technology and its critical role for computer science education in ACM/IEEE CS2013 [20]. IAS KA prepares the workforce with knowledge, skills, and abilities to protect information and information systems and ensures Confidentiality, Integrity, and Availability (CIA) in computing. The importance of security concepts and topics has emerged as a core requirement in the computer science (CS) discipline. Despite the great need of such professionals, many CS programs have not offered security education due to the lack of security expertise and lab resources. Even for many of these CS programs with IAS courses, they still remain in traditional in-class teaching or seminar type class, conducted without hands-on labs to practice. The challenges in teaching the principles and practice of secure software development in a dedicated course or integration into multiple relevant courses include: scarce learning materials and hands-on practice labs; dedicated staff and faculty in this field; difficult choices in teaching formats and setting.

On the other side, smart mobile devices are becoming much more powerful and popular, and also less expensive than before that almost every student has his/her own mobile smartphone or tablet. As mobile computing is becoming more and more popular, the security threats to mobile application are simultaneously increasing explosively. Most malicious activities not only hack the user's private information, such as contact and location information or hijack the user's transactions and communications, but also hack the organization data by exploiting the confidential data with BYOD [1-3]. There is an urgent demand for security workforce, which has resulted in a shortage of cybersecurity professionals [7]. We need to promote mobile security education and meet the emerging industry and education needs. However, mobile security is a relatively weak area in most schools' computing curriculum, and there is a shortage of effective mobile-security learning materials and hands-on laboratory platform and resources. Database security is one of the most important security areas to be addressed. The database is the most popular target for malicious attacks. More and more IT educators realized the importance of database security education in their program curriculums. Many schools are integrating database security topics into database and cybersecurity education [4-5]. The three most crucial components of database security are Database CIA components, where Confidentiality addresses the protecting of information from disclosure to unauthorized parties (who has access to data) by data encryption, SSL, permissions, access control; Integrity focuses on protecting information from being tampered by unauthorized parties (what data can be changed) by message hashing, signature, etc.; Availability ensures that authorized parties should be able to access the information when needed even during DDOS.

The work is partially supported by the U.S. National Science Foundation under awards: 1244697, 1438858, 1663350, and U.S. Department of Homeland Security Scientific Leadership Award grant number: 2012-ST-062-000055.

The threats to data confidentiality may come from SQL injection into the SQL database and any other data leakages due to the lack of input validation and weak data leak protection. Integrity failure is often caused by unauthorized access without enforced access control and authentication or the lack of least privilege policy with excessive privileges and permissions granted. Weak availability will make database go down unexpectedly. Backup the data at periodic intervals to ensure data recovery in case of application issues. Databases should be secured against security vulnerabilities and malware [6-10].

To achieve broader database security education, we proposed a set of portable, modular, and easy-to-adopt MDS learning modules based on Android SQLite database, which is widely used not only in Android, but also in iOS. PC and mobile SQLite databases face most of the common database security threats that other commercial databases are also facing. The new learning modules include principles of emerging database security, companion labs with inexpensive portable mobile-based labs, and well-designed projects that provide students with a hands-on learning experience.

The Android SQLite based learning modules for database security will create an engaging and motivating learning environment for database security concepts and practices. This approach provides students with hands-on laboratory practices on real-world mobile database security. The laboratory consists of multiple modules covering major database vulnerabilities. Each topic consists of a series of progressive sub-labs: a pre-lab, lab activities, and a student add-on post-lab.

The developed labs can easily be integrated into many courses in CS and IT curricula as an individual practice lab and can also be used in a dedicated mobile security course. The labs can be carried out anywhere and anytime by students in the classroom or outside of the classroom.

The selected labs have been implemented in various computing courses, such as database, mobile software development, and mobile app & security to enhance the security learning in the discipline. The preliminary results from student performance evaluations and their feedback showed that this approach can increase their learning confidence in the state-of-the-art technology and promote their self-efficacy. Many students showed their creativity with their new findings and new solutions to protect mobile devices and mobile apps. Students appreciated the hands-on learning experiences. Also, all of the labs are affordable and easily adopted with the open source Android Studio and smartphones and tablets.

To meet the need for IAS education, especially for database security education, we propose an authentic pedagogically-effective immerging learning approach through portable real-world-scenario hands-on mobile based labs. This approach aims to increase student self-efficacy and engage and motivate them in learning. The portable lab platform is affordable, easily adoptable, sustainable, and accessible anywhere and anytime. This learning approach has multiple effects that can not only foster the database security education in CS, but also enhance students' Java mobile programming skills. The lab components can be integrated into individual CS courses or used in a dedicated IAS or mobile security course. The popularity of smart devices and portable labs makes it affordable and accessible anywhere and anytime. The short learning curve with open source Android Java and lab integrality also makes this approach feasible for easy adoption and sustainable for faculty and helps to build up their capacity in IAS.

II. RELATED WORK

Many efforts have been made to enhance the secure software development education in recent years. Boston university, Eastern Illinois University, Stevens Institute of Technology, UT Dallas, and many other schools have offered database security since 2011 [11-15]. However, these efforts focus primarily on lectures and few of them offer hands-on labs in desktop and server environments. They also offer no insight into the security vulnerability of mobile software applications.

Recently, NSF has supported a number of research projects for mobile security research for education [16-19], which mainly focus on the mobile threat analysis, rather than the learning approach and hands-on learning materials for mitigation, solution, and countermeasures for database vulnerability threats. Some NSF projects have used mobile device based labware to teach mobile security [21] and had positive impacts on student learning.

Our project is designed to broaden MDS education with a set of innovative MDS learning modules, which is portable, modular, integratable, and easy-to-adopt with hands-on learning environment. All the labs offer insight into the database security vulnerability and threats. Also, these labs provide the security countermeasures for learning enhancement.

III. LEARNING MODULE DESIGN

The leaning modules are designed in a such way that they can not only be used for MDS but also be used for general database security education. Each module consists of a series of progressive sub-labs: a pre-lab, hands-on lab activities, and a student add-on post-lab. The pre-lab activities include learning objectives, overview, ethicial issues, targeting courses. Lab activities include hands-on practices. Post-lab activities include review questions and answers, assignments, and projects.

The pre-lab introduces the principles of a specific database security subject. We are developing eight hands-on mobile MDS learning modules with the secure database concepts and practice guidance to identify the database security flaws on the following eight subjects:

1. Data sanitization for input validation
2. Protection for data at rest(encryption)
3. Protection for data on motion(data sync)

4. Protection on data sharing(Content Provider)

5. Permissions

6. Data access control

7. Auditing(syslog)

8. Backup and recovery

The above mobile database security principles can also be applied to all kinds of database, and each of them is overviewed in their corresponding pre-lab. The pre-lab presents learning concepts and tutorials on the subject to prepare students with fundamental background knowledge on the subject.

The lab activity itself provides students with step-by-step interactive hands-on practice, which engages and motivates students in building mobile apps or security tools on their own mobile devices and boost their positive emotions through mastery experiences. Students learn how to avoid and fix the known flaws and mitigate such security risk (practical aspect). There is no better way to learn database security than actually seeing the flaws and the security consequences through real practice. We expect students will get a good understanding of the security risks and vulnerabilities and form basic ideas of secure programming to reduce such vulnerabilities.

Working on Android software with practical experiences will increase student learning interest because students are able to see their applications running in real time. This provides students with a sense of accomplishment and real world relevance, which will promote their programming confidence and self-efficacy.

The post-lab provides an add-on lab where students can share their own new analysis and defensive development on the topics. These post-labs can help students to build self-efficacy and confidence and promote their creativity in their own abilities through project experiences.

Portable and modular designed MDS learning labs are easy to be adopted and integrated into lower division computing courses, such as lower division Java computer programming courses, and upper division computing courses, such as mobile app development, database, information security, and others.

IV. SAMPLE MDS LEARNING MODULES

In this section, we illustrate the MDS learning modules and labs with the brief descriptions on two selected subjects to show how a module is organized and implemented.

A. Secure mobile database for content provider

1) Pre-Lab Activities

- Learning objective:
- Understand the concept of content provider
- Understand malicious input (SQL injection) for content provider vulnerability
- Migitate the risks

2) Overview

A content provider is an Android building component responsible for providing access sharing to data stored in a repository which may bring vulnerability. The stored data can be queried by other android apps (client). Both, the provider and client applications run on the same Android device. Fig. 1 illustrates that a content provider stores data in SQLite, Files and Internet and provides create(C), retrieve(R), update(U), delete(D) interfaces to client applications (App1, App2, Malware). Here, malware applications are intending to exploit content provider vulnerabilities by providing malicious inputs.

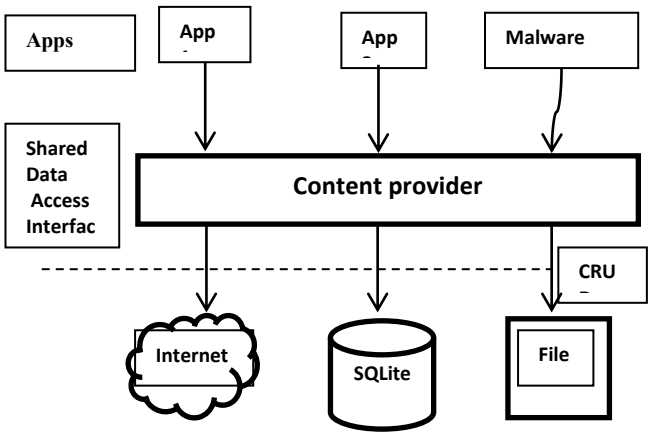


Fig. 1. Content provider and client applications

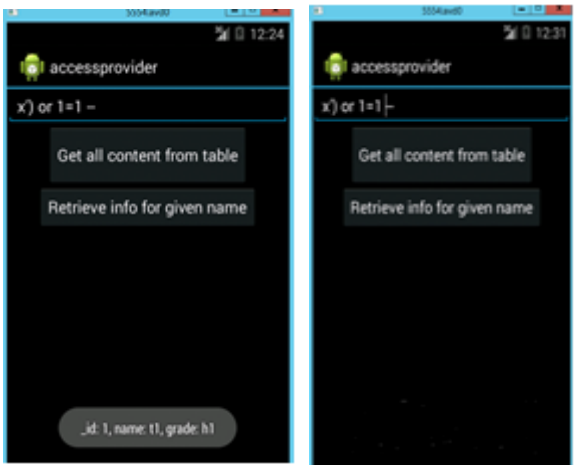
3) Lab activities:

Table 1 below shows a code snippet of a vulnerable content provider class (StudentsProvider) implementing query() interface, which takes five parameters. The Uri represents a database table. The projection indicates the list of columns to return, selection represents the where condition. The arguments accepted by the query() method is directly passed to the query invocation API qb.query(). Thus, an attacker may provide malicious inputs to exploit them. For example, it is possible to provide a list of arbitrary columns. Another example is to provide a tautology attack input through selection argument.

TABLE 1. VULNERABLE CODE IN CONTENTPROVIDER QUERY INTERFACE

```
public class StudentsProvider extends ContentProvider {
    public Cursor query(Uri uri, String[] projection, String selection,
String[] selectionArgs, String sortOrder) {
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
        qb.setTables(STUDENTS_TABLE_NAME);
        … … …
        Cursor c = qb.query(db,projection,selection,selectionArgs,null,
null,sortOrder); //vulnerable line
        … … …
        return c;  }
    … … … }
```

Fig. 2 (A) shows a screenshot of exploiting the vulnerability by providing tautology attack input in the input filed (x') or 1=1 --). The input is passed to the selection argument without filtering for SQL injection. The toast message shows an example of data retrieved from the table. (_id:1, name:t1, grade:h1).



(A) (B)

Fig. 2. (a): Exploiting content provider vulnerability, (b) Screenshot after fixing vulnerability

TABLE 2. FIXING OF VULNERABILITY IN CONTENTPROVIDER QUERY INTERFACE

```
public class StudentsProvider extends ContentProvider {
    … … …
    public Cursor query(Uri contentUri, String[] projection, String
selection, String[] selectionArgs, String sortOrder) {
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
        qb.setTables(STUDENTS_TABLE_NAME);
        if (selection.contains("*") || selection.contains(";") ||
selection.contains("'") || selection.contains("\"")) {
  //SQL injection
          return null;
        }
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
        qb.setTables (STUDENTS_TABLE_NAME);
        … … …
        Cursor c = qb.query (db,projection,selection, selectionArgs,
null, null, sortOrder);   //vulnerability fixed
        … … …
        return c;
    } }
```

Table 2 shows fixing of the vulnerability by performing additional checking on selection parameter content. The if statement is checking for malicious characters that could be part of SQL injection, such as *, ;, ', and ". If any character is found, no result is returned. Fig. 2 (B) shows a screenshot of the fixed code, where applying tautology attack input leads to no results as toast message.

B. *Protection for data at rest (public encryption)*

1) Pre-Lab Overview

Learn database encryption to protect sensitive data. You can use Android cipher API to encrypt and decrypt the SQLite database with public key encryption or use SQLite extension SQLChiper to encrypt the database. We introduce the first approach here.

2) Learning Objectives

- Learn how to use RSA to encrypt and decrypt sensitive data in the database
- Implement an Android program that encrypt and decrypt data using RSA in SQLite database.

3) RSA cryptosystem and example

RSA is one of the first practicable public-key cryptosystems, which is widely used for secure data transmission where the encryption key is public and the decryption key is kept secret. In RSA, this asymmetry is based on the impractical factoring the product of two large prime numbers. Students learn the RSA encryption and decryption algorithms via a simple example via a simple example.

Fig. 3 shows the concept of the public encryption algorithm.

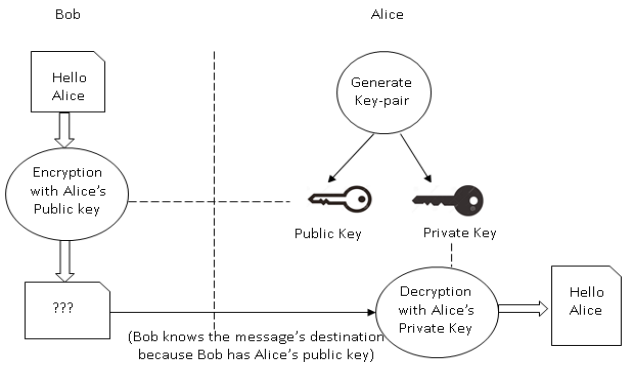


Fig. 3. Encryption with Public key and decryption with private key.

4) Lab activities

This lab shows how to protect SQLite database lab through encryption and password based authentication. If data is not stored in encrypted manner or protected with a password, then data can be read by an attacker. Table 3 is a vulnerable code snippet for a notebook application, where a user is able to enter information into notebook (stored as sqlite table) without any authentication. The MainActivity class invokes a new intent to facilitate storing user provided data into notebook database. The NoteActivity class invokes addNote() operation by storing the data without encryption.

TABLE 3. VULNERABLE CODE IN SQLITE DATABASE APPLICATION

```
public class MainActivity extends ListActivity {

  public void onCreate(Bundle savedInstanceState) {
     Button addNote=(Button)findViewById(R.id.addBtn);
     addNote.setOnClickListener(new View.OnClickListener(){
     public void onClick(View view) {
```

```
            Intent oNoteActivity = new Intent(getBaseContext(),
NoteActivity.class);
            startActivity(oNoteActivity);
        }
    });
}
public class NoteActivity extends Activity {
    private NotepadOperations notepadDBoperation;
    EditText name;
    EditText writing;
  public void saveNote(View view) {
    Notepad note =
notepadDBoperation.addNote(name.getText().toString(),
writing.getText().toString()); //insecure operation
    Intent oNoteActivity = new Intent(getBaseContext(),
MainActivity.class);
    startActivity(oNoteActivity);
  }
}
```

Assume that a sample of notebook entries (note1, note1, new123, Hello, test). Now, an attacker can obtain a copy of it. Fig. 4 shows a screenshot of the contents by hexdump tool.

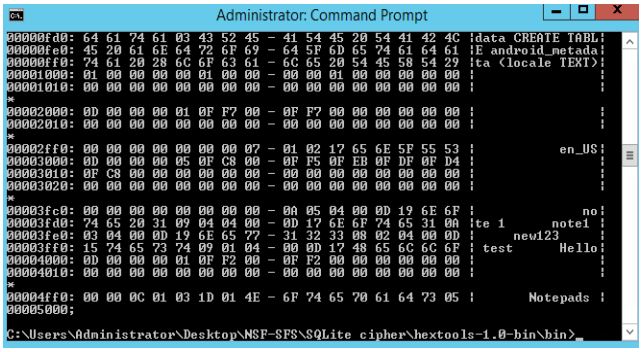


Fig. 4. Unencrypted SQLite database entries

To fix the issue, a new Activity can be added which will ask a user for password and then validate the provided password to grant access to the database for viewing or editing entries through note activity. Table 4 shows the user is validated through checkDBforSecret() method call. If the information matches, then a user is allowed to enter data.

TABLE 4. FIXING OF VULNERABILITY IN SQLITE DATABASE APPLICATION

```
public class StartActivity extends Activity {
  … …
  protected void onCreate(Bundle savedInstanceState) {
  …       public void onClick(View view) {
          checkDBforSecret(); //password checking
        }
    });
  }
  private void checkDBforSecret(){
    String secret=mPassword.getText().toString();
    secretDBoperation = new SecretOperations(this);
    secretDBoperation.open();
    secretDBoperation.addSECRET("1234");
    boolean isAccess= secretDBoperation.IsSecretCorrect(secret);
    if(isAccess){
      Intent oNoteActivity = new Intent(getBaseContext(),
MainActivity.class);
      startActivity(oNoteActivity);
      Toast.makeText (getBaseContext(),"Access
Granted",Toast.LENGTH_LONG).show();
    }
    else{
      Toast.makeText(getBaseContext(),"Access
Denied",Toast.LENGTH_LONG).show();
    }
secretDBoperation.deleteSECRET((Secret)secretDBoperation.getAllS
s().get(0));
  }
}
```

The Fig. 5 shows a encrypted SQLite database is protected from the hexdump exploration.

Fig. 5. Encrypted SQLite database entries by hexdump

V. COURSE EXPERIENCE AND EVALUATION

We have implemented selected labs in various computing courses, such as Database, Mobile App Development, and other security relevant courses to enhance the security learning in the discipline. The preliminary results from student performance evaluations and their feedback showed that this approach can increase their learning confidence in the state-of-the-art technology and promote their self-efficacy. Many students showed their creativity with their new findings and new solutions to protect mobile devices and mobile apps. Students appreciated the hands-on learning experiences. Also, all labs are easily adopted with the open source Android Studio IDE and Android smartphones and tablets. We have conducted quantitative surveys with the following questionnaire in the spring 2016 database class, and the overall response from students was overwhelmingly positive.

Q1: I like being able to work on Android App development with this hands-on labware.

Q2: The real world mobile security problems in the labs such as SQL-Injection help me gain knowledge and experience on mobile security and understand better on cybersecurity

Q3: The Online hands-on labs help me learn and practice anywhere and anytime without space and time constraints.

Q4: The project helps me learn both mobile DB App developments with SQLite and DB security

There were 18 students in this survey and their feedback was very positive. Fig. 6 shows our survey

results. Overall, more than 85% students responded "agree" on all questions in the scale of: [Agree], [Neutral], [Disagree]. Most of students said they enjoyed the hands-on labs with Android in the course. Some students said in their comments:

1. *Best class, very helpful.*
2. *Overall, I think the labs were sufficient in educating us about the basic concepts of mobile app security.*
3. *This database class has been the best I have attended in KSU so far. Learnt a lot. Thanks a lot.*

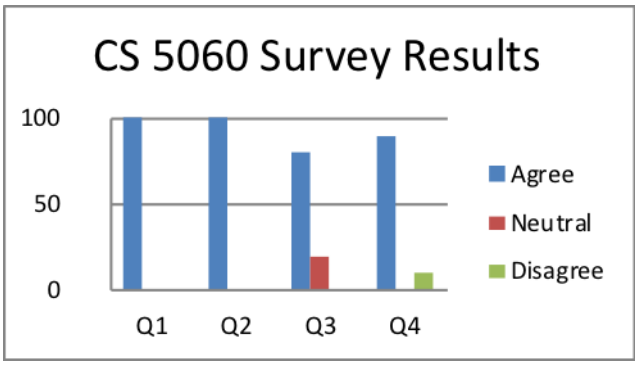


Fig. 6. Survey result of CS 5060 Database Design (Spring 2016)

VI. CONCLUSION

This paper presents an approach to immerse students in mobile authentic learning of database security through real-world hands-on experience. Some of the selected learning modules are being implemented in the Database, Mobile App Development, and Mobile Security classes. The preliminary feedback from students was very positive. Students have gained hands-on real world experiences in database security with Android mobile devices, which also greatly promoted students' self-efficacy and confidence in their database security learning.

ACKNOWLEDGMENT

The work is partially supported by the U.S. National Science Foundation under awards: 1244697, 1438858, 1663350, and U.S. Department of Homeland Security Scientific Leadership Award grant number: 2012-ST-062-000055. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and Department of Homeland Security.

REFERENCES

[1] P. Ghorbanzadeh, A survey of mobile database security threats and solutions for it, Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on, IEEE Xplore: 03 August 2010

[2] Database Security Report Q1 2016, http://www.informationsecuritybuzz.com/articles/database-security-report-q1-2016/

[3] WHITE PAPER Top Ten Database Security Threats: The Most Significant Risks of 2015 and How to Mitigate Themhttp://www.imperva.com/docs/WP_TopTen_Database_Threats.pdf

[4] L. Yang, Teaching database security and auditing, ACM DL, ACM SIGCSE 2009, http://dl.acm.org/citation.cfm?doid=1539024.1508954

[5] F. Li, NSF IUSE: Collaborative Project: Building Virtual Research, Interactive, Service, and Experiential Learning Modules for Cyber Security Education, Award Number:1431330

[6] WHITE PAPER Top Ten Database Security Threats: The Most Significant Risks of 2015 and How to Mitigate Themhttp://www.imperva.com/docs/WP_TopTen_Database_Threats.pdf

[7] Introduction to Database Security Issues Types of Security Database,http://www.academia.edu/6866589/Introduction_to_Database_Security_Issues_Types_of_Security_Database

[8] R. Maurer, Top Database Security Threats and How to Mitigate Them,https://www.shrm.org/hrdisciplines/safetysecurity/articles/pages/top-database-security-threats.aspx, 2015

[9] Database Security Report, A SPECIAL REPORT FROM THE EDITORS AT CYBERSECURITY VENTURES, http://cybersecurityventures.com/database-security-report-2016/

[10] Category:Sensitive Data Protection Vulnerability, https://www.owasp.org/index.php/Category:Sensitive_Data_Protection_Vulnerability

[11] J. Andrus, J. Nieh, Teaching Operating Systems Using Android, Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE 2012), 2012 (Best Paper Award)

[12] http://www.bu.edu/csmet/files/2011/08/674-A1.pdf

[13] http://castle.eiu.edu/pingliu/tec5363/Notes/tec5363Instructor_Bio.htm

[14] https://web.stevens.edu/academic_files/courses/syllabus/CS696syl.pdf

[15] http://www.utdallas.edu/~muratk/courses/dbsec16s-syllabus.pdf

[16] REU Site: Research on Security of Mobile Devices and Wireless Networks, https://www.nsf.gov/awardsearch/showAward?AWD_ID=1559652

[17] EDU: Collaborative: Enhancing Pervasive and Mobile Computing Security Education With Research Integration, https://nsf.gov/awardsearch/showAward?AWD_ID=1419280&HistoricalAwards=false

[18] Collaborative Project: Capacity Building in Mobile Security Through Curriculum and Faculty Development, https://www.nsf.gov/awardsearch/showAward?AWD_ID=1241651&HistoricalAwards=false

[19] EDU: Deploying and Evaluating Secure Programming Education in the IDE, https://www.nsf.gov/awardsearch/showAward?AWD_ID=1523041&HistoricalAwards=false

[20] ACM/IEEE-CS Computer Science Curricula 2013, https://www.acm.org/education/CS2013-final-report.pdferemy Andrus, Jason Nieh, Teaching Operating Systems Using Android, ACM SIGCSE 2012

[21] P. Bhattacharya, Li Yang, Minzhe Guo, Kai Qian, Ming Yang: Learning Mobile Security with Labware. IEEE Security & Privacy 12(1): 69-72 (2014)