

Mobile Device Penetration Testing Framework and Platform for the Mobile Device Security Course

Suyash Jadhav*, Tae Oh*, Young Ho Kim**, Joeng Nyeo Kim**

*Dept. of Information Sciences and Technologies, Dept. of Computing Security, Rochester Institute of Technology,
152 Lomb Memorial Dr, Rochester, NY, USA

**Cyber Security System Research Dept., Electronics and Telecommunication Research Institute,
218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, KOREA

thoics@rit.edu, ssj8127@rit.edu, wtowto@etri.re.kr, jnkim@etri.re.kr

Abstract— The authors have developing mobile device evaluation and testing platform to evaluate the mobile malware. Using the platform, the authors have created several courses in mobile device security. One of the important requirements is to provide students with a safe and sandboxed environment for malware analysis. Other features include tool enhanced lab environment, updated malware repository, log collection and exact assistance. Java based client-server application have been created to serve these requirements. Also a framework to perform mobile malware analysis and mobile penetration testing is proposed and implemented under this research work. Paper focuses on analysing requirements for such coursework to perform mobile malware analysis and mobile application penetration testing. Paper also gives details about the tools created and framework implemented to successfully teach Advanced Mobile Device Security course and perform interactive lab exercises..

Keywords— Mobile Malware Analysis Framework, Mobile Device Security, Mobile Penetration Testing, Mobile Application Vulnerabilities, Mobile Application Security Testing Framework, Mobile Malware Repository

I. INTRODUCTION

Number of mobile device users has been increasing significantly in last couple of years, and mobile applications are becoming integral tools for daily life. Therefore, protecting data used by mobile application has been becoming a critically important. At enterprise level, BYOD policies allow employees to connect their mobile devices to the enterprise networks. This allows an opportunity for hackers to penetrate into the network, and recent survey indicated in increased attacks using mobile malwares exponentially. So, it is important for security professionals to understand the security for both mobile device and application levels. As a part of research, several efforts have been taken to create an architecture of a runtime malware analysis framework and enhanced mobile penetration testing framework.

II. BACKGROUND

Recently, mobile device security has been becoming an emerging field of research and the types of mobile devices are divide into two applications, enterprise application and personal application. In the general, mobile device security focuses on Mobile Device Management (MDM), device level security, storage security, transport layer security, and mobile device application security. The authors considered all of those focused points for designing a course materials and the detailed requirement analysis was performed for lab platform design.

As for the analysis result, the following requirements were derived:

- a. Perform easy maintain and updates for the malware repository,
- b. Provide a proper AVD (android virtual device image) configuration
- c. Provide a safe lab environment,
- d. Easy and exact assistance to student,
- e. Provide features of log collection and data collection for grading and trouble shooting purposes.
- f. Provide detailed malware analysis lab exercises,
- g. Create penetration testing framework and lab exercises to analyse OWASP Mobile top 10 vulnerabilities.
- h. Create test bed for learning mobile penetration testing.

A detailed survey and evaluation were performed to select best open source tools for the mobile device security curriculum. One of the best suitable operating system chosen was Santoku-Linux, which includes the tools related for mobile forensics, malware analysis, mobile application security and development tools [8].

III. MALWARE ANALYSIS

Malware analysis is a core part of the mobile device security curriculum. Several mobile malware analysis techniques are considered for designing the labs for this coursework and those techniques are static analysis, dynamic analysis, network analysis, and user intent and geographical location of the servers for finding outlier.

A. Malware Analysis Techniques

1) Static analysis: Static analysis utilizes code reverse engineering techniques, and explores the malware code. Static analysis labs offer students with a program level understanding and application features of malware. This analysis also provides understanding of permission characteristics and exploiting malware characteristics. Reverse engineering tools used in analysis are Apktool, Dex2Jar. Static analysis also reveals functional capabilities of malware and possible family to which the malware belongs. Considerable number of malware families are made available to students [1].

2) Dynamic analysis: Dynamic analysis utilizes multiple tools to monitor the activities within the device. By observing the generated data from the tools, malware behaviour can be evaluated. This analysis is much faster and dynamically elastic technique when analyzing large number of malware and can be automated and deployed in cloud-based frameworks. Dynamic analysis reveals details about functional call by mobile malware and other system level activity calls related to malicious activities. Battery and network usage also helps in predicting malicious behaviour of application [2]. Stack activity of the malicious application is analysed as well. Tools utilized in dynamic analysis lab exercises are Android Device Monitor, Logcat, Dumpsys and Strace. Those tools create log files, which gives detailed information about the activities performed by malware.

3) Network level analysis: Network level analysis involves understanding of network protocols used by the malware to send data to remote servers. Malware utilizes http, https and ftp protocol but there have been families of malware using SMTP to compromise user private data. Network level malware analysis labs help students to understand network traffic characteristics of malware and provide understanding to effectively setup network Intrusion Detection System (IDS) in the future. Tools used for capturing the device traffic were TCPdump and Wireshark, and the captured .pcap files were analysed using WireShark.

4) User intents and geographical location based anomaly detection: An attempt has been made to understand behavioural intent and normal characteristics of user to define normal activities. If any malware gets installed on the user device, the server

location involved in malicious activities will be detected as outlier. Labs have been designed with a specific malware where normal New York localized user suddenly start sending data to server in Ukraine, which is unexpected and need to be analyzed as sudden outlier. Also enterprise mobile device suddenly starts communicating to one of the blacklisted ISP networks like 'Beyond The Network America' it might trigger an alarm for security operation centre.

B. Necessary Requirements For The Malware Analysis Lab Environment

1) Maintain updated malware repository: It is important to provide students with easy to access malware repository using categories and relevant malware samples. Using malware repository reduces risk for students in downloading unknown and potentially dangerous malware and allow faculty to control malware repository.

2) Sandboxed environment: Students must analyze malware in a secured and isolated environment by avoiding spread of malware to institution's network.

3) Analysis of platform aware malware: Some intelligent malware requires actual phone to analyze their behaviours, and students need to have access to mobile devices to perform the analysis.

4) Exact assistance: During malware analysis, students go through many steps and could get into issue. So the logging of all performed commands must be available.

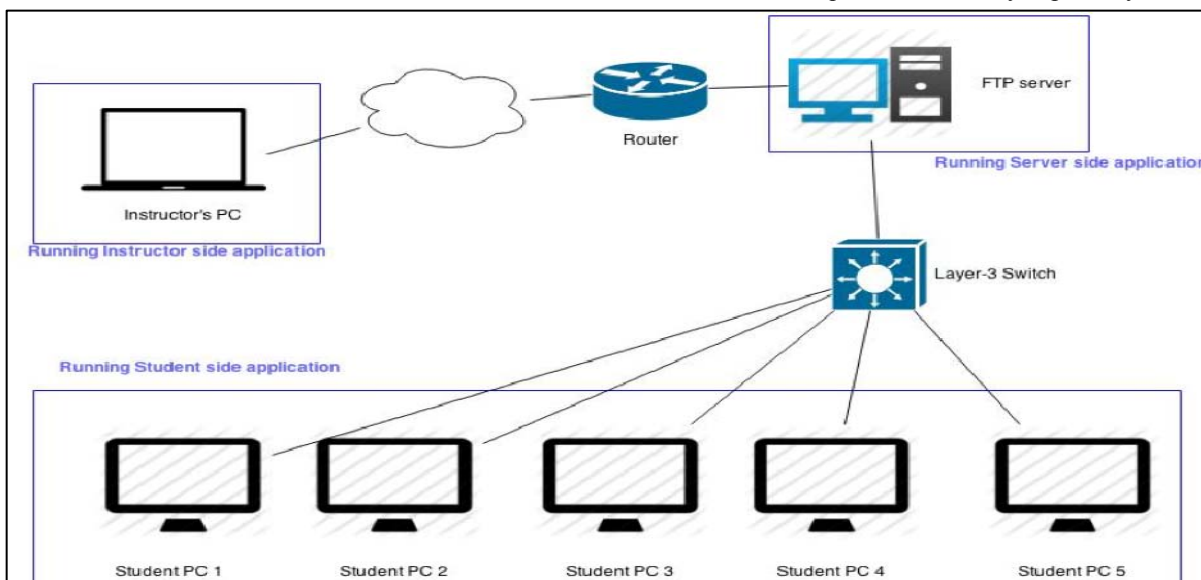
5) Log collection for grading labs: Performing of the lab exercise and successful completion of predefined checkpoints need to be reported to faculty using the logging method which results in time saving and faster grading.

6) Easy and user-friendly faculty side interface: The faculty should have user-friendly Graphic User Interface (GUI) to manage the malware repository, collect the logs from each student, control the virtual malware platform.

C. Malware Repository And Log Collection Platform Design And Details

To server the requirement of easy repository maintenance

and log



collection a platform has been proposed and implemented using java application and client-server based design methodology. Refer to Figure 1 for overview of the architecture and Figure 2 for class diagram of developed application.

1) Student side JAVA application functionalities:

Secure Login: It is important for the malware repository to be restricted and accessible through secured login. The malware repository has an admin access to instructor, and students require login credential to access resources. Also instructor creates course specific access control for students.

Download emulator specification: Android emulator specifications are provided for successful implementation of lab exercises. It's found that many malware target specific Android platform, for analysis of such malware it is important to configure emulator with specific configurations. Considering this fact, the platform has facility to download configuration file to perform specific lab exercises.

Download malware: Students get access to malware repository resources allocated to specific course. This allows

followed during installations and changes made in device configurations. So server this purpose student can send his recorded logs and modified device configuration to central server.

2) Central server functionality:

Access control: The central server is programmed with strong access control over data resources. User gets an access to specific resources allocated to him/her by the instructor.

Repository sync: The server keeps sync between newly log user and update in data resources. Any new files or changes to existing files are immediately pushed to user side, helping user to get the latest updates.

3) Instructor side JAVA application functionalities

Malware repository maintenance: Instructor is provided with functionality to update, add and delete any content on the malware repository. Instructor is expected to keep the

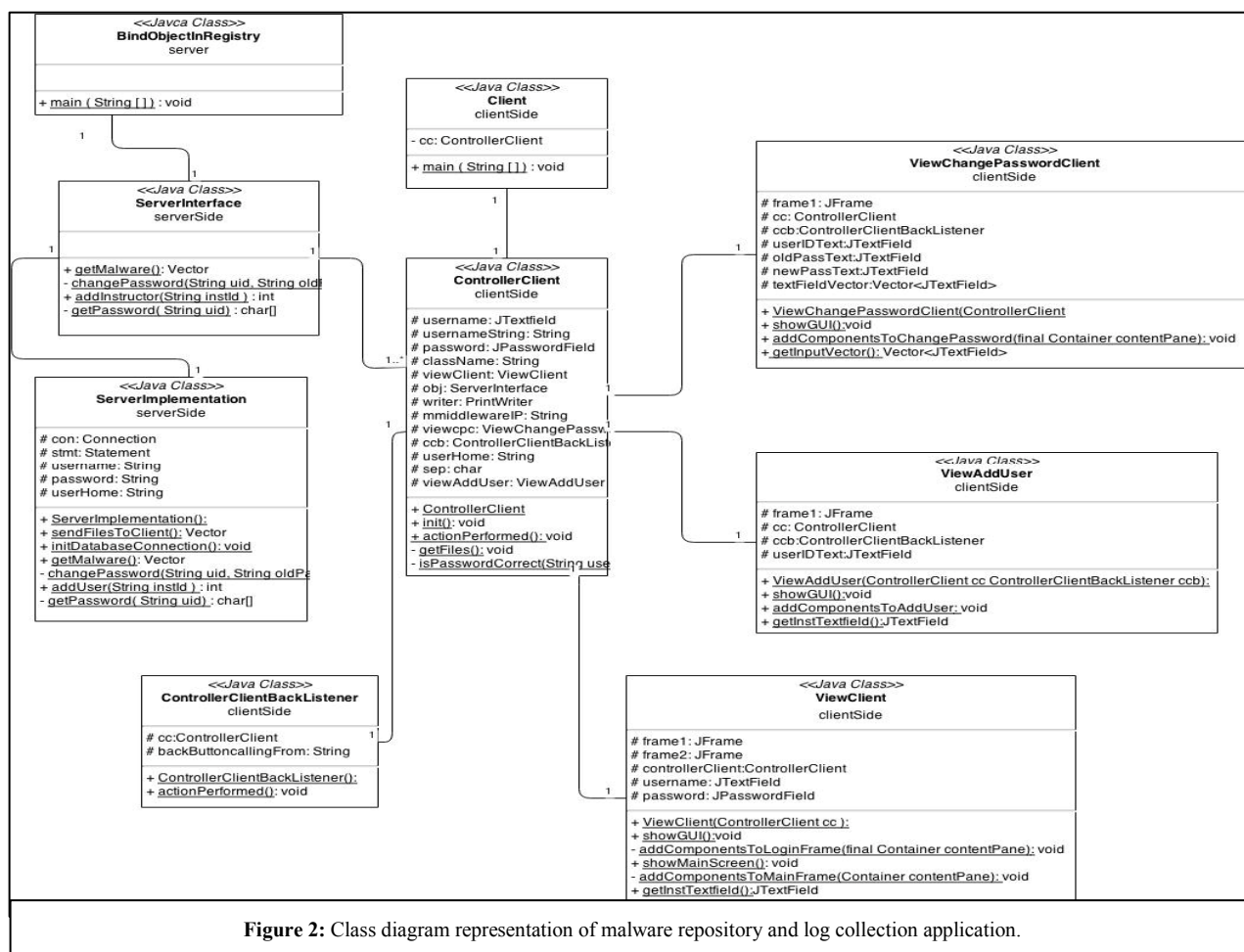


Figure 2: Class diagram representation of malware repository and log collection application.

student to get all the data and malware samples required for performing malware analysis at one place.

Upload logs and modified device configuration: For the exact assistance from instructor, one needs knowledge of steps

repository updated with latest malwares and analysis technique documentations.

Add-Delete user: Instructor can create any new group of users or any individual user with access to different resources available in malware repository. Instructor is given complete administrative access and can add other instructors.

Pull the student's collected logs and configuration files: Instructor can pull data collected from individual student, for assistive purpose.

D. Advantages Of The Created Platform

1) Central server: Centralized server allows flexibility to offer this course from multiple locations across the world and resource maintenance is easy.

2) Remote access and repository maintenance from instructor application: Faculty/Instructor can change the repository any time from anywhere, which gives a very good control over the malware repository update.

3) Access control: Strong access control techniques are implemented, which restricts the unauthorized access to student data, and unintended distribution of mobile malware can be restricted.

4) Network traffic analysis collection: Central server gives storage space for the network traffic captures, which allows analysis at any user intended time. The lab environment is secured by using Deep Freeze configured OS, and it avoids need of external storage to store .pcap files.

5) Test data on SDCard image: Students are provided with dummy personal data, and this data is provided in SDCard image, which can be mounted in emulator.

6) Emulator configurations: Students are provided with configuration files for emulator, which allows them to perform malware analysis in required manner.

IV. PENETRATION TESTING LAB ENVIRONMENT AND EXERCISES

Penetration testing of mobile application is new field and very few resources are available on the web. Research involves interactive and detailed lab exercise creation, which will train students to latest mobile application penetration testing techniques. Labs are designed to find the OWASP Mobile Top 10 Risks [3][4].

E. Proposed Lab Environment Framework Component Diagram

The Figure 3 shows the diagrammatic view of tools and other components used in designing mobile penetration lab environment. The framework is designed with an aim to provide tools enhanced lab environment for finding OWASP mobile top 10 vulnerabilities. The tools used provide detailed analysis and exploit performing capabilities to students.

F. Test Bed

OWASP mobile security project has provided an excellent android penetration test bed. This test bed is utilized to write

lab exercises to teach students mobile penetration testing techniques. Lab exercise uses two applications 1. Herd Financial and 2. Four Goats first one is a banking application and the second one is social networking application. Used test bed has an independently running server, which handles all the server application, web services and databases. The test bed is meant to have OWASP mobile top 10 vulnerabilities. Labs are designed to utilize many other tools to find these vulnerabilities [5][9].

G. Techniques Proposed In Performing Mobile Penetration Testing

1) Static analysis: Different programming level securitybugs can be easily found by performing application reverse engineering. Misconfigurations during database creation like setting MODE_WORLD_READABLE to 1 or misconfiguring Content Providers can be easily detected during code analysis. Also use of any vulnerable API and libraries can be noted in code analysis [7].

2) Port scanning: Port scanning is a formal method to analyse the device features and services running on the device. NMAP is used to design a port scanning lab exercises. OS finger printing and possible attack prone services are found in such exercise.

3) Finding IPC based attack surfaces: Drozer is used to find possible vulnerable IPC. And other exploits can be carried out using Drozer. Labs are designed to utilize other capabilities of Drozer framework like SQL injections [6].

4) Proxy based attack for MITM: Burp Suite proxy is used to exploit the transport layer flaws and perform man in the middle attack. The Herd Financial test bed has transport layer security flaws. Labs provide exploitation technique to exploit transport layer flaws and potential business logic flaws. Also private information can be found through exploring the sent data through application due to insecure transmission of data[10].

5) Dynamic analysis for invalid input failure: Client side input validation is required. Test bed allows SQL injection and reveals the stored information as the client side input validation is not performed properly.

6) Logcat analysis for data leakage through log generation: Application developers many times log some critical error, which gives more knowledge about the possible attacks surfaces.

7) Insecure data storage analysis: Insecure data storage can be a critical issue for mobile application, which stores credit card or financial data. Content provider miss configuration and insecure storage of encryption key may lead to such flaws.

8) Detailed vulnerability reporting and report generation: Some labs are dedicatedly designed to teach students about writing a professional mobile application auditing report.

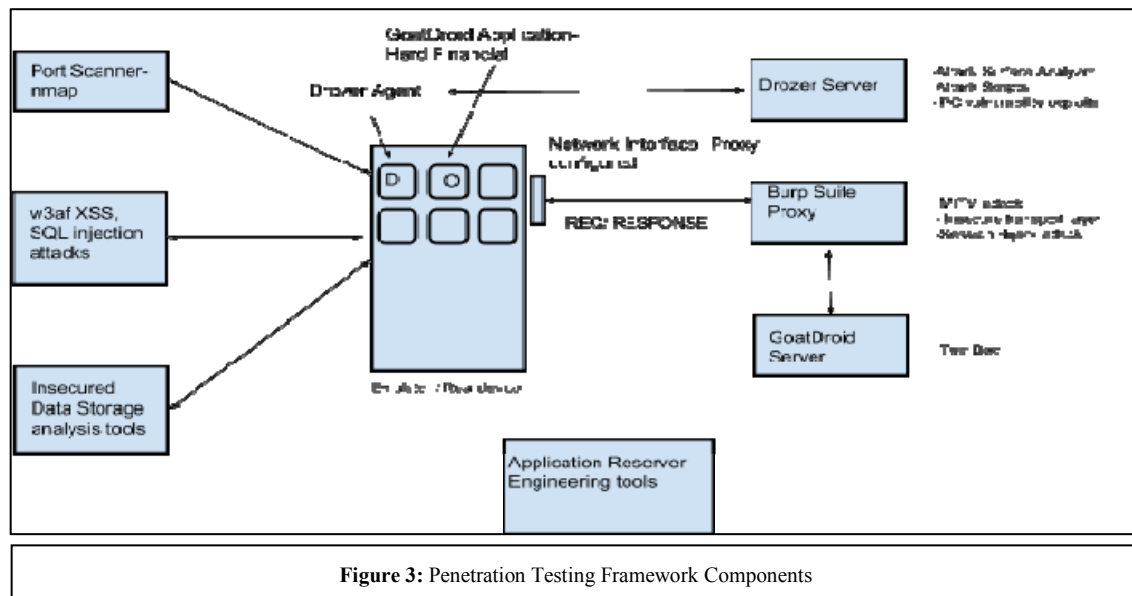


Figure 3: Penetration Testing Framework Components

9) Exact analysis of vulnerabilities and recommendation on them are required to be documented.

V. OWASP MOBILE TOP 10 VULNERABILITY TESTING

OWASP project has special chapter for mobile vulnerabilities. Detailed survey from OWASP for finding leading mobile application vulnerabilities resulted in listing out top 10 mobile application flaws. Following is the listing of OWASP mobile top 10 and methodology to test these security flaws.

A. Weak Server Side Control

Weak server side control is more over a focus of web application security and web service security. Flaws in server side control can be found with general web application security testing technique. To exploit business logic flaws try to perform MITM attacks and perform fuzzy testing on web service.

B. Insecure Data Storage

In mobile application insecure data storage can occur due to lack of awareness about security in developers. Insecure data storage flaws can be found with reverse engineering or code analysis. Data storage options available in Android are shared preferences, internal storage, external storage, SQLite database, network storage. To find out possible insecure data storage, look for use of storage modes `MODE_WORLD_READABLE` and in some cases `MODE_WORLD_WRITEABLE`. This mode allows insecure read of data and insecure modification to data.

C. Insufficient Transport Layer Protection

Flaws in transport layer protection allows attacker to capture user critical data over network. These attacks involve traffic capture using malicious network nodes. To find out application flaws in these category look for use of unsecure communication protocols like HTTP, also check for mechanism to update stored certificates. Updates to stored certificate from untrusted sources are also a problem. Use

proxy server to perform MITM attack and look for possible data leakage.

D. Unintended Data Leakage

Many time developers unknowingly reveal sensitive data through log files or through error messages. In android Logcat utility is used for system log generation and it is seen that on application crash or error many time sensitive argument data and static variables are printed in error log messages. Look for log collection and error messages for any unintended data leakage.

E. Poor Authorization and Authentication

It is common practice to create multiple activities within one application. But many times access to resources held by sensitive activities is not properly authorized. Authentication activity can be bypassed to access critically sensitive data like banking information and credit card data. Check for publicly exported sensitive activities.

F. Broken Cryptography

Broken cryptography is due to poor keys management or due to use of deprecated cryptographic algorithms. This allows attacker to decrypt the data captured over the network. Also it allows data retrieval by physically accessing the phone, avoiding purpose of protecting against physical theft.

G. Client Side Injection

Mobile devices uses SQLite databases and it is important to perform input validation before performing any query using user given inputs. SQL injection is possible and can be exploited very easily in mobile devices. This exploit reveals all the stored personal and critical data. Also searching history, transactions and other private data can be breached. To find this flaws perform code analysis and look for insufficient input validations. Perform random SQL injections and providing unexpected data inputs.

H. Security Decisions Via Untrusted Inputs

This flaw allows unauthorized user to gain access to critical data, this occur due to logical flaw in programming. This is

similar to privilege escalation. In this flaw application uses untrusted inputs to take critical authentication and authorization decision. For example providing bank details with account number as only input, this allows malicious users to enter account number for other users to steal banking data. This flaw can be found by fuzzy testing, penetration testing and code review.

I. Improper Session Handling

Improper session handling or broken session allows attacker to steal session of legitimate user any perform malicious activity or steal user data. This is a result of either unsecured management of session keys or predictable session keys. This allows attacker to steal user session id from his mobile device or simple predict the session key and use that session to pretend being legitimate user. To avoid this flaw look for mechanism use to handle session keys, also look for repetition or predictability in session keys. One can use Burp proxy to modify any session related data.

J. Lack of Binary Protection

Lack of binary protection will result in exposure of application code to attacker. Android SDK have functionality to obfuscate the code using ProGuard. Developers need to use code obfuscation to avoid reverse engineering of their applications. This allows developer to patch any security bugs before attacker. It helps in circumvents exploits attacker perform by understanding programming flaws.

VI. FUTURE WORK

Deploy McAfee EMM and design labs for mobile device administration

Enterprise uses many professional security monitoring and vulnerability solutions integrated in SIEM. It is a necessary task to teach students with such professional tools regarding mobile device security. Course involves installation process of McAfee EMM (Enterprise Mobility Management) solution, policy creation and compliance using McAfee EPO. These labs focuses on policy based mobile device management and incidence response methods on lost or stolen mobile devices.

Creating more advanced mobile penetration learning test bed

In the future this research team will focus more on creating advance mobile penetration test bed. Also will try to get actual world beta phase mobile applications for auditing purpose.

Designing more advanced malware detection techniques and lab exercises

Abnormal file permission change detection on rooted device- Sudden file permission changes on the rooted phone and suspicious file permissions can be considered as a trigger for detecting malicious activities on mobile device. Research team is focusing on creating lab exercises, which will involve more advanced and innovative malware detection techniques. Outlier detection using AI algorithms and heuristic based mobile malware analysis techniques are been studied.

Botnet beacon analysis

Botnet based malware are targeting Android devices on grater scale. Detection of such infected device and malicious application using the Botnet beacons in mobile devices is currently being researched. And successful detection techniques will be used for creating more advanced lab exercises in future.

Collection of malware responses by DNS faking

Creating a malware analysis environment in which DNS faking techniques and malware responses to such faked DNS resolutions will be captured.

VII. CONCLUSION

The implemented malware analysis framework and the malware repository applications help in creating tool enhanced and safe malware analysis lab environment. The labs created for mobile penetration testing carefully covers OWASP mobile top 10 vulnerabilities. Also the implemented mobile penetration-testing framework for performing mobile application penetration lab exercises gives more flexible and detailed auditing capabilities to the students.

This work was supported by the ICT R&D program of MSIP/IITP. [R0101-15-0195(10043959), Development of EAL 4 level military fusion security solution for protecting against unauthorized accesses and ensuring a trusted execution environment in mobile devices].

VIII. REFERENCES

- [1] William Enck, Damien Ocateau, Patrick McDaniel, and Swarat Chaudhuri, "A study of android application security," In Proceedings of the 20th USENIX conference on Security (SEC'11), p.21-21.
- [2] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang, "Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices," In Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), Article p.5-6.
- [3] (2014) BlackHat website. Introducing the Smartphone Penetration Testing Framework by Georgia Weidman. Available: <https://media.blackhat.com/ad-12/Weidman/bh-ad-12-smartphone-penetration-Weidman-WP.pdf>
- [4] (2014) OWASP website. Top 10 Mobile Risk from OWASP Mobile Security Project. Available: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#ab=Top_10_Mobile_Risks
- [5] (2014) GitHub website. OWASP GoatDroid test bed. Available: <https://github.com/jackMannino/OWASP-GoatDroid-Project>
- [6] (2014) MWR INFOSECURITY website, Drozer. Available: <https://www.mwrinfosecurity.com/products/drozer/>
- [7] (2014) Android Open Source Project website. Android Security Overview. Available: <http://source.android.com/devices/tech/security/index.html>
- [8] Keith Makan and Scott Alexander-Bown, *Android Security Cookbook*, Birmingham UK: Packt Publishing Ltd., 2013
- [9] (2014) VIAFORENSICS website. Santoku-Linux Features and OS Details. Available: <https://santoku-linux.com/about-santoku>
- [10] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith, "Why eve and mallory love android: an analysis of android SSL (in)security," In Proceedings of the 2012 ACM conference on Computer and communications security (CCS'12), p. 50-61.