A Secured, Update Oriented Malware Analysis Framework And Tool Enhanced Mobile Penetration Testing Framework Designed for Advance Mobile Security Coursework

ABSTRACT

Research work was aligned to design an Advance Mobile Device Security coursework and assistive tools for such course. It is a necessary requirement to provide students with a safe and sandboxed environment for malware analysis. Other necessities includes tool enhanced lab environment, updated malware repository, log collection and exact assistance. Java based client-server application have been created to serve these requirements. As well as a framework to perform mobile malware analysis and mobile penetration testing is proposed and implemented under this research work. Paper focuses on the analyzing the 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.

1. BACKGROUND

Mobile device security is new, upcoming and important field of research. Mobile devices are categorized based on their use likely as enterprise asset or personal device. In the generalized view mobile device security focuses on Mobile Device Management (MDM) solutions, device level security, secured storage, transport layer security, and mobile application penetration testing.

While designing a coursework, which could accommodate all these aspects of the mobile device security a detailed requirement analysis was performed. The main requirements analyzed were as follows- a. Updated and easy to maintain malware repository, b. Providing right AVD (android virtual device image) configuration, c. Tools enhanced and safe lab environment, d. Easy and exact assistance to student, e. Log collection and data collection for grading, f. Detailed malware analysis lab exercises, g. Penetration testing framework and lab exercises to analyze OWASP Mobile top 10 vulnerabilities, h. Test bed for learning mobile penetration testing.

A detailed survey was performed to utilize the currently available Open Source tools in the mobile device security analysis. One of the best suitable operating system chosen was Santoku-Linux, which is enhanced with the mobile forensics, malware analysis and mobile application security testing and development tools [8].

2. MALWARE ANALYSIS

Malware analysis is the core part of the mobile device security curriculum. Few distinct mobile malware analysis techniques are used for designing the labs in this coursework. Mobile malware analysis techniques implemented are a. Static analysis, b. Dynamic analysis, c. Network analysis, d. User intent and geographical location of the servers for finding outlier.

2.1 Malware Analysis Techniques

Static analysis

Static analysis gives student more program level understanding and application features of the malware. It gives understanding of permission characteristics and exploit utilization by malware. Reverse engineering tools used are Apktool, Dex2Jar. Static analysis also reveals the capabilities of the malware and possible family to which the malware belongs. Considerable numbers of family based categorized android malware are made available to students [1].

Dynamic analysis

Dynamic analysis is much faster and dynamically elastic technique when analyzing large number of malware. Dynamic analysis 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 behavior of application [2]. Stack activity of the malicious application is analyzed. Tools utilized in dynamic analysis lab exercises are Android Device Monitor and Logcat, which are available in android ADT bundle.

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 helps student to understand what network traffic characteristics are observes from the mobile malware and utilizing this knowledge to effectively setup network IDS in future. Tools used for capturing the device traffic were TCPdump and .pcap files were analyzed using WireShark.

User intents and geographical location based anomaly detection

An attempt has been made to understand the intent of the user and the normal characteristics of the 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 expected 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 center.

2.2 Necessary Requirements For The Malware Analysis Lab Environment

Maintain updated malware repository- Providing students with easy to access malware repository. Providing more categorized and relevant malware samples. Reducing the risk for student in downloading unknown and potentially dangerous malware. Allowing faculty to control the malware repository.

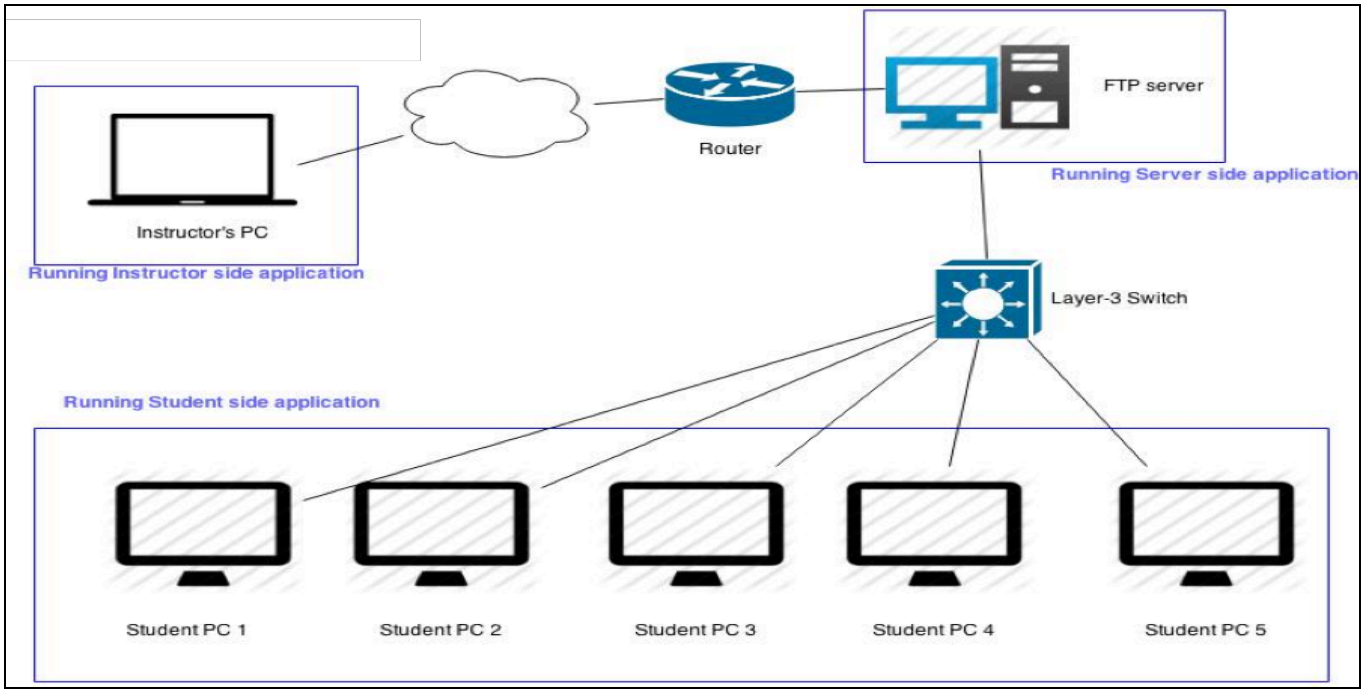

Figure 1: Malware Repository and Log Collection Platform Architecture Overview

Sandboxed environment- Students analyze these malware in a secured and isolated environment. Avoiding spread of the malware to institution's production network.

Analysis of platform aware malware- Some intelligent malware requires actual phone to analyze their behavior and students need to be provided with required devices to perform the malware analysis.

Exact assistance- During malware analysis students will go through lots of steps and might get stuck. So the logging of all performed commands must be available.

Log collection for grading labs- Abstract of the lab exercise and successful completion of predefined checkpoints need to be reported to faculty, results in time saving and faster grading.

Easy and user-friendly faculty side interface- Proposed system must have easy to use application interface on the faculty side.

2.3 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.

The figure 1 shown above gives architecture details for the implemented platform. The platform has following sub parts-

Student side JAVA application functionalities

- Secure Login
- Download emulator
- Backup emulator
- Command terminal to give ADB commands
- Download malware
- Record lab implementation video and screenshots on emulator

FTP server functionality

- Access control
- Repository sync
- Collect the log data
- Take backups of the emulator configurations periodically
- Allow 24x7 remote access and accessible over internet

Instructor side JAVA application functionalities

- Emulator configuration repository maintenance
- Malware repository maintenance
- Add-Delete user
- Pull the students collected log
- Pull the students emulator configuration file

2.4 Advantages Of The Created Platform

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

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

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.

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

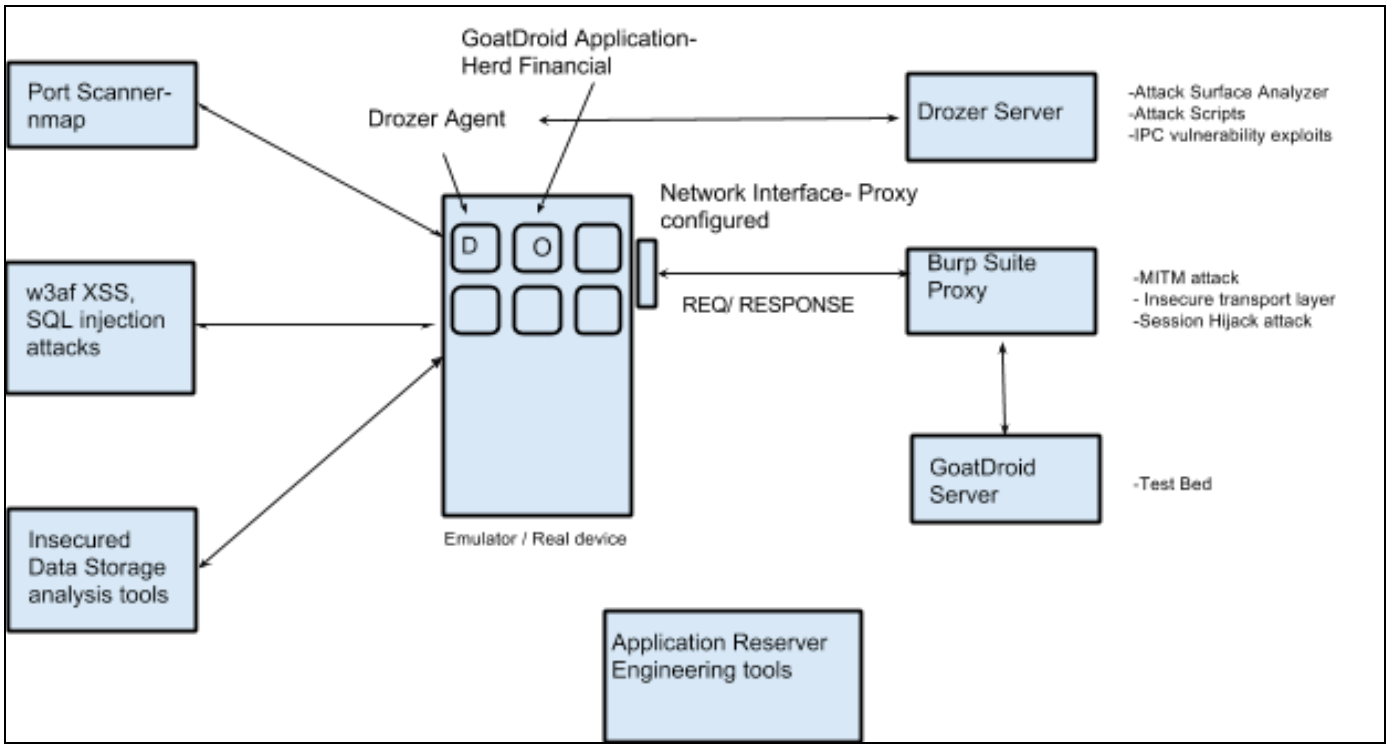

Figure 2: Penetration Testing Framework Components

Test Data on SDCard image- Students are provided with dummy personal data, this data is provided in SDCard image, which can be mounted in emulator.

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

3. PENETRATION TESTING LAB ENVIRONMENT AND EXERCISES

Penetration testing of mobile application is new filed 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].

3.1 Proposed Lab Environment Framework Component Diagram

The figure 2 shows the diagrammatic view of the 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.

3.2 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 handing 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].

3.3 Techniques Proposed In Performing Mobile Penetration Testing

Static Analysis

Different programming level security bugs 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].

Port scanning

Port scanning is a formal method to analyze 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.

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].

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 ahs been designed to exploit these as well as more information and potential business logic flaws can be found through exploring the sent data through the application [10].

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.

Logcat analysis for data leakage through log generation

Application developers many times log some critical error, which gives more knowledge about the possible attacks.

Insecure data storage analysis

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

Detailed vulnerability reporting and report generation

Some labs are dedicatedly designed to teach students about writing a professional mobile application auditing report. Exact analysis of vulnerabilities and recommendation on them are required to be documented.

4. 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.

5. 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.

6. REFERENCES

[1] William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri. 2011. A study of android application security. In Proceedings of the 20th USENIX conference on Security (SEC'11). USENIX Association, Berkeley, CA, USA, 21-21.

[2] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang. 2011. 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). ACM, New York, NY, USA, , Article 5 , 6 pages. DOI=10.1145/2070562.2070567 http://doi.acm.org.ezproxy.rit.edu/10.1145/2070562.2070567

[3] Introducing the Smartphone Penetration Testing Framework by Georgia Weidman, Retrieved June 2014, from BlackHat: https://media.blackhat.com/ad-12/Weidman/bh-ad-12-smartphone-penetration-Weidman-WP.pdf

[4] Top 10 Mobile Risk from OWASP Mobile Security Project Retrieved June 2014, from OWASP: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Risks

[5] OWASP GoatDroid test bed, Retrieved June 2014, from github.com/jackMannino: https://github.com/jackMannino/OWASP-GoatDroid-Project

[6] Drozer, Retrieved June 2014, from MWR INFOSECURITY: https://www.mwrinfosecurity.com/products/drozer/

[7] Android Security Overview, Retrieved June 2014, from Android Open Source Project: http://source.android.com/devices/tech/security/index.html

[8] Keith Makan, Scott Alexander-Bown *Android Security Cookbook.* Packt Publishing Ltd., Birmingham UK, 2013

[9] Santoku-Linux Features and OS Details, Retrieved June 2014, from VIAFORENSICS: https://santoku-linux.com/about-santoku

[10] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. 2012. 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). ACM, New York, NY, USA, 50-61. DOI=10.1145/2382196.2382205 http://doi.acm.org.ezproxy.rit.edu/10.1145/2382196.2382205