

[British Columbia Institute of Technology]

# [Data Exfiltration & Android Security Suite]

---

*[Major Project Proposal]*



Name: Jivanjot S. Brar

Student ID:	A00774427
-------------	-----------

Instructor:	Aman Abdulla
-------------	--------------

COMP8045:	Major Project
-----------	---------------

Date:	March 5, 2015
-------	---------------

BCIT Bachelor of Technology Program  
*Network Security & Administration*

# CONTENTS

---

<b>1 </b>	<b><i>CURRICULUM VITAE</i></b>	<b>4</b>
<b>I.</b>	<b>FORMAL EDUCATION</b>	<b>4</b>
<b>II.</b>	<b>WORK EXPERIENCE</b>	<b>5</b>
<b>III.</b>	<b>AREAS OF SPECIALIZATION</b>	<b>6</b>
<b>2 </b>	<b><i>PROJECT INFORMATION</i></b>	<b>7</b>
•	Data Exfiltration Tool	7
•	Android Security Tools	7
<b>I.</b>	<b>BACKGROUND INFORMATION</b>	<b>9</b>
•	Android Security Suite	9
•	Data Exfiltration Tool	10
<b>II.</b>	<b>ASSUMPTIONS</b>	<b>11</b>
•	Android Security Suite	11
•	Data Exfiltration Tool	11
<b>III.</b>	<b>SCOPE</b>	<b>12</b>
•	Android Security Suite	12
•	Data Exfiltration Tool	14
<b>IV.</b>	<b>INNOVATION</b>	<b>15</b>
•	Android Security Suite	15
•	Data Exfiltration Tool	15
<b>V.</b>	<b>TECHNICAL CHALLENGES</b>	<b>17</b>
•	Android Security Suite	17
•	Data Exfiltration Tool	17
<b>VI.</b>	<b>METHODOLOGY</b>	<b>19</b>
•	Android Security Suite	19
•	Data Exfiltration Tool	21
<b>VII.</b>	<b>TECHNOLOGIES USED</b>	<b>23</b>
•	Android Security Suite	23
•	Data Exfiltration Tool	23

<b>VIII. TESTING PLAN .....</b>	<b>24</b>
• Android Security Suite .....	24
• Data Exfiltration Tool.....	26
<b>IX. SCHEDULED ESTIMATES.....</b>	<b>28</b>
• Data Exfiltration Tool.....	30
• Android Security Suite .....	30
<b>X. DELIVERABLES .....</b>	<b>32</b>
<b>3  REFERENCES .....</b>	<b>33</b>
<b>4  CHANGE LOG.....</b>	<b>34</b>

# 1 | CURRICULUM VITAE

## 1. FORMAL EDUCATION

My formal education has been fragmented between countries. My early education took place in India and my secondary and post-secondary education took place in Canada. Upon completing my grade 12, I decided to enrol at BCIT with the intention of completing the Bachelor of Technology Program.

2013 – 2015

**British Columbia Institute of Technology**

Bachelor of Technology Program

- Network Security & Administration

2010 – 2012

**British Columbia Institute of Technology**

Diploma of Technology Program

- Technical Programming Option

2005 – 2010

**Sands Secondary School**

High School Diploma (Grade 8 – 12)

## II. WORK EXPERIENCE

My work experience in the field of Computer Systems has been very little; therefore I decided to go to school in order to further my education in the field of technology. It is also the reason why I have designed a practicum that draws from my interest in programming, network security and mobile development and provide me with some useful experience in each individual field.

2014 - present

### **DNN Corp.**

Software Support Analyst

- Providing answers to clients by Analyzing and identifying software problems; researching solutions; guiding clients through a series of actions over the telephone or email to help resolve issues or help install and configure software.
- Work continuously on a task until completion (or escalation of ticket to Tier 2 or Tier 3 for further assistance).
- Prioritising and managing many open cases at one time.

2013 - 2013

### **XModus Software Inc.**

Contract Web Developer

- Converted PSD files into HTML/CSS pages
- Created interactive plugins such as sliders, slideshows using JavaScript
- Created store locator plugin using JavaScript and PHP.
- Provided HTML/CSS/JavaScript bug fixes

### III. AREAS OF SPECIALIZATION

During my education at BCIT, I have learned numerous things. I have always had a passion for programming and always enjoyed learning about new techniques and exploits' hackers or security experts used to compromise a network or a system. Before my education at BCIT, I learned everything from trial and error, but going through Bachelors of Technology program has really polished me as a programmer and gave me a whole new set of skills and knowledge.

I am currently nearing the end of my Bachelors of Technology degree at BCIT; with a dual option in Technical Programming and Network Security. It is with these specializations and my interest in development that I have designed a practicum that draws from each area of expertise and fully make use of what I have learned during my time with the program.

## 2 | PROJECT INFORMATION

This practicum is focused on two applications: 1) Data Exfiltration Tool and 2) Android Security Tools. Details of each applications are described below:

- **Data Exfiltration Tool**

The primary portion of this application will be to create prototype desktop applications that will highlight various portions of the main application and then a final application that will utilize all the prototypes to combine and create a fully functional Data Exfiltration Tool. The main functionality of this tool will be to run on a target desktop machine and send the contents of the system in an encrypted format to the controller machine. For the purposes of this practicum this application will be targeted towards Linux machines. This application will use the common encryption tools such as Triple DES and also some advanced information hiding techniques using QR Codes and Steganography ( Akshay Choche and Hamid R. Arabnia) & (Somdip Dey, Kalyan Mondal, Joyshree Nath, Asoke Nath, 2012).

- **Android Security Tools**

The primary focus of this application is an exploration of Android OS with respect to network penetration testing security suite. The main functionality of this suite for the purpose of the practicum will be to provide users with few common security tools that can be used to compromise system; gather information on the network; or craft and send custom packets to analyze their response to the packets. Throughout my exploration of this practicum, I will attempt to implement the following 4 major features and 1 optional feature in the Security Suite:

- ARP Spoofer – used for spoofing the router and (single or multiple) victim machines on a network, for the purpose of tapping on the network traffic between the target machine and the router.
- DNS Spoofer – used for intercepting DNS requests from victim machines and feeding them false information ultimately redirecting them to a fake server.
- TCPCDump – used to capture network packets both on mobile and wireless network.
- Packet Crafter – send custom packet to a destination device and capture the response packets.

- Network Scanner – scans the network and returns with a list of available hosts on a network, list open ports on a target host (OPTIONAL).



## 1. BACKGROUND INFORMATION

### • Android Security Suite

A penetration test, or the short form pentest, is an attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data. (Stephen Northcutt, Jerry Shenk, Dave Shackleford, Tim Rosenberg, Raul Siles, and Steve Mancini;, 2006)

The process involves identifying the target systems and the goal, then reviewing the information available and underlying available means to attain the goal. A penetration test target may be a white box or black box. A penetration test can help determine whether a system is vulnerable to attacks.

Penetration tests are valuable for several reasons:

- 1) Determining the feasibility of a particular set of attack vectors.
- 2) Accessing the magnitude of potential business and operational impacts of successful attacks.
- 3) Testing the ability of network defenders to successfully detect and respond to the attacks.
- 4) Identifying vulnerabilities that may be difficult or impossible to detect with automated network or vulnerability scanning software.

Network Security experts such as penetration testers use variety of security tools to perform penetration testing on a Network. Some of these tools are specialized OS Distributions, which are geared towards performing penetration testing. Distributions typically contain pre-packaged and pre-configured set of tools (Kali Linux Tool Listing, n.d.). This is useful because penetration tester does not have to hunt down a tool when it is required. Popular examples of OS distributions are Kali Linux, Pentoo, and WHAX etc. Some popular examples of software tools include Metasploit, nmap, w3af and many more.

There are various types of exploits or attacks that can be used by the penetration testers and these attacks are divided into two categories: Active and Passive attacks. Passive attacks are when network intruder intercepts data traveling though the network, and Active attacks is in which an intruders initiates commands to disrupt the network's normal operation.

Types of attacks include: (Network security, 2014)

- Passive
  - Wiretapping / Packet Sniffing
  - Port Scanning
  - Idle Scan
- Active

- Denial-Of-Service (DOS) attack
- Spoofing
- Man In The Middle
- ARP Poisoning
- Buffer Overflow
- Stack Overflow
- SQL Injections
- Brute Force Password Guessing

Above mentioned OS distributions, software's are fully developed for desktops or laptops, however they are in the beginning phase for Android and have only been partially developed/experimented with on Android Platform.

### • **Data Exfiltration Tool**

Data exfiltration is the unauthorized copying, transfer or retrieval of data from a computer or server. Data exfiltration is a malicious activity performed through various different techniques. Such a transfer may be manual and carried out by someone with physical access to a computer or it may be automated and carried out by cybercriminals through malicious programming over a network. Data exfiltration is also known as data extrusion, data exportation or data theft.

Data exfiltration is primarily a security breach that occurs when an individual's or organization's data is illegally copied. Generally, data exfiltration's are targeted attacks where the hacker's primary intent is to find and copy specific data from the target machine. Once attackers acquire the stolen information, the impact to any organization or individual may include sabotage, data theft and damage to brand image and reputation.

Attackers use a mix of legitimate and malicious tools and techniques in order to extract specific data from the target machine. One of the most common techniques used for data exfiltration is the use of backdoors.

Backdoors have built-in upload and download functions and are commonly installed in target systems. Backdoors can upload collected files and commonly use ports like 80 and 443 (for HTTP or HTTPS) and port 53 (for DNS) to hide their traffic. One of the drawbacks for backdoors is their ability to transfer small amount of data in order to stay undetected. (TrendLabs)

## II. ASSUMPTIONS

There are few assumptions that development of these applications are depended on. I have listed these assumptions below.

- **Android Security Suite**

- 1) Development of all these application in the suite is based on single and most important assumption that all these Android Applications will run in a similar manner as they do on a Linux machine. The basis of this assumption is that Android Kernel is based on one of the Linux Kernels. Android Operating System uses an underline Linux Kernel in order to access and utilize its hardware components. (Android Operating System)
- 2) The application will require root access to the device for proper execution. This assumption is derived from the similar behaviour on a Linux machine where application would require a root access in order to access the network card capabilities for packet capture and custom packet crafting purposes.
- 3) Android device will support frameworks for Packet Sniffing and Packet Injecting.
- 4) The application will be able to capture packets on a Wi-Fi Network.
- 5) The application will be able to capture packets on a Device Data Network.
- 6) The application will be able to inject packets on a Wi-Fi Network.
- 7) The application will be able to inject packets on a Device Data Network.

- **Data Exfiltration Tool**

- 1) One of the benefits of this tool is its ability to encode large amounts of data, which could be single or multiple files. The assumption here is that QR codes have the ability to store large amounts of data, and for the purposes of this project, this application will be able to encode one or more files into single QR codes.
- 2) Second feature of this exfiltration tool is the encryption of the QR code generated for the data copied from the target machine. This feature is depended on the ability to convert the QR code into bytes and then encrypt the bytes.
- 3) Most important part also included ability to decrypt the encrypted bytes and regenerate the QR code.
- 4) Another assumption here is that decrypted and regenerated QR code will be an exact copy of the original and therefore it will be possible retrieve the file(s) encoded into the QR code.

### III. SCOPE

- **Android Security Suite**

This application consists of 5 sub-applications: Network Scanner, TCPDump Packet Scanner, Packet Crafter, ARP Spoofer and DNS Spoofer. I have scoped the project down to the basic execution of the 4 sub-applications and main wrapper application. The fifth application Network Scanner has been marked Optional in order to reduce the scope of this practicum. The optional application will only be included in the final product if the project is about to finish ahead of schedule. However if the project reaches or exceeds the scheduled dates, the optional application will be skipped. In this section I have described features of individual sub-applications.

1. TCPDump Packet Capture (AndroShark)
  - a. Allows user to capture and filter packets over the network.
  - b. Displays the captured packet and their detailed information.
  - c. Reads captured packet information from a PCAP file.
  - d. Writes captured packet information in a PCAP file.
2. Packet Crafter
  - a. Tool for auditing Firewalls and networks.
  - b. Create and Send custom TCP packets
  - c. Create and Send custom UDP packets
  - d. Create and Send customer ICMP packets
  - e. Captures response packets
3. ARP Spoofer
  - a. Sends packets to one Host (Default Gateway) and one Target Machine
  - b. Sends packets to one Host (Default Gateway) and many Target Machines
  - c. Combines with TCPDump to capture the packets.
  - d. Combines with TCPDump to Filter incoming packets
4. DNS Spoofer
  - a. Combines with ARP Spoofer to capture packets over the LAN.
  - b. Filters incoming DNS Inquiry packets with specific URLs or Addresses
  - c. Filters all incoming DNS Inquiry packets
  - d. Creates custom DNS response packets
5. Network Scanner (Optional)
  - a. Host Discovery – Identifying hosts on a network. For Example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.

- b. Port Scanning – Enumerating the open ports on target hosts.

- **Data Exfiltration Tool**

Data Exfiltration tool is a backdoor application that runs on the target machine and has two main purposes 1) Receive commands from the controller/hacker machine, execute those commands and collect the results, 2) send the collected information in a covert and encrypted format back to the controller machine.

For the purposes of this project I will be focusing on the sending the data in an encrypted format back to the controller machine. The data will be encrypted and then embedded into an image before sending it to the controller machine. This application will first encode the data to create a QR code. The application will then take the QR image and convert into bytes and encrypt those bytes using some of the popular encryption mechanism such as TrippleDES. The cypher text received from encrypting the QR image will then be embedded into a host Image using stenography. This image will then be sent to the controller machine.

The receiver will need to know the mechanism used to embed information into the image and will need to know the appropriate keys and encryption mechanism to decrypt the information extracted from the image to redraw the QR code and then retrieve the actual data by decoding the QR code.

In this project I will first start by creating prototype applications for highlighting 4 aspects of this application:

- 1) Generating QR code for the selected Data & Retrieving the data from the generated QR Code,
- 2) Converting the QR code into Bytes, encrypting the bytes & Redrawing the QR code after decrypting the bytes,
- 3) Embedding the Bytes into an image & Retrieving the bytes from the image,
- 4) Sending & Receiving the image with embedded information.

In the end I will combine all the prototypes to create 1) a receiver application that will run on the controller machine & 2) exfiltration application that will run on the target machine.

This application will be developed using Java and will primarily focus on Linux systems.

## IV. INNOVATION

### • **Android Security Suite**

The applications developed in this project are used as penetration testing tools designed to test strengths of network defense tools such as firewalls etc. Penetration testing tools help determine the weak points of the network and also help testers determine the feasibility and a magnitude of an attack on the network. The tools discussed in this proposal are currently part of Operating System such as Kali and can also be manually installed in Linux OS.

My Innovation for this application involves introduction of few penetration testing tools in a mobile platform specifically Android. A penetration testing framework contains many tools, for this project I have focussed on some of the common tools such as sniffers, crafters and scanners.

Android currently lacks frameworks for developing low level networking tools; this project will help me lay down a framework for low level networking tools for Android and will allow me to pave a path for creating a penetration testing or pentesting suite for Android devices.

In this project, some of the innovative tools that I am focusing on that either have not been developed for Android device or I am including some addition functionality include:

- ARP Spoofer – This application already exist for an Android, however I am including an additional functionality which allows more than one machine over the network be targeted to this attack.
- DNS Spoofer – This application has not been developed for Android Platform and will be an extension of the ARP Spoofer.
- Packet Crafter – This application is similar to HPing or NPing, however this application also does not exist for an Android Platform.

### • **Data Exfiltration Tool**

Data Exfiltration tool is a backdoor application that runs on the target machine and its main purpose is to receive and execute command and the send collected information to the controller machine. For this application the innovation lies in the transfer of information back to the controller machine. There are two innovate components of this application:

- 1) The use of QR codes by the backdoor to encode large amounts of data and transfer it to the controller machine. The typical method of transferring data is to embed small portions of data in a TCP/IP packet and send it over port

80 (HTTP) or 443 (HTTPS), or embed the data in DNS packet and send it over port 53 (DNS).

- 2) The encryption of the QR code and the use of steganography to hide the data before transfer. The typical method is to encrypt the data directly using encryption mechanism of choice and then transfer the encrypted data through appropriate protocols.

Both of these components add unique features to this Data Exfiltration tool. This Data Exfiltration tool is able to send large amounts of data through the use of QR codes and the transfer of the data is undetected as the information is being sent in an image and the person monitor the traffic will only see an image being sent and will not be able to retrieve any information hidden inside due the use of steganography and encryption of the QR code.



## V. TECHNICAL CHALLENGES

This project allows me to utilize and build upon everything that I have learned in all the Network Security courses.

- **Android Security Suite**

Even though Android's open platform allow us to build these pentesting tools, lack of any previous projects present me with a few extremely difficult technical challenges. Challenges for this application include but aren't limited to following:

- Learning how to develop an Android Application. I have no prior experience for building Android application therefore this aspect of the project is going to be one of the major challenges that I will be facing while working on this project.
- Second most challenging aspect of the project is locating & using appropriate Android Libraries to achieve access to low level resources such as the network card & network stack etc. These libraries are crucial for this project and if no Java Networking Libraries can be located for accessing low level Android resources, project will require following a difficult path for using Native C or C++ code for achieving this functionality.
- If the project does rely on using Native C or C++ code, this will raise another challenge for learning how to develop Android application using Android NDK and running native code in Android.

Above are some of the most difficult challenges that are presented by this application, however overcoming these challenges will provide me with immense experience in Android Application development and also in Network Security application development.

- **Data Exfiltration Tool**

For this project innovation lies in two areas 1) embedding large amounts of data in a QR code, 2) encryption & retrieval of content from the QR code. Even though QR codes are extremely popular, they have not been utilized to the same extent as this project requires. Currently, QR codes are used to encode small amounts of data such as small messages, page URLs etc.

Some of the major technical challenges that exist include the following:

- Ability to encode multiple files into a QR code.
- Ability to decode and retrieve files from the QR codes.
- Encrypting the QR codes
- Decrypting and retrieving the data from the QR codes.

- Encoding the encrypted QR Codes into a host image.
- Retrieving the embedded in data from the host image.

Above are some of the main features of this application, where the innovation for application lies. These innovative features are also the main areas where the major technical challenges exist.

## VI. METHODOLOGY

### • Android Security Suite

The exploration of this Security Suite application can be divided up into several discrete modules, which can then be integrated. The following sections will briefly outline each module and its function.

#### 1. Packet Capturing

Capturing network packets is an integral part of this application and is used by all 5 sub-applications. This module in application will be responsible for capturing any and all network packets that arrive at the network card of the device. It will then filter the packets based on the sub-application it's incorporated into. The difference here will be the packet filter applied by the sub-application. For example DNS Spoofer will capture all packets however it will only display any incoming DNS packets. Similarly TCPDump will allow user to apply a custom filter which will then be passed to this module and this module will only return the packets that matches the specified filter by the sub-application.

What this means is that this functionality will only be developed once and will then be incorporated into each sub-application and therefore reducing the duplicate code and also reducing the effort to test and fix any bugs.

#### 2. Packet Crafting

One of the other main features of this application is the ability to create and send custom packets over the network. This ability create custom packets is used by all the sub-applications except TCPDump. This feature involves creating custom packets for different protocols such as IP, TCP, UDP, ICMP, DNS, and ARP etc. The purpose of this feature is to allow the application to create custom packets in order to retrieve information about the network.

This module will work alongside with Packet Capturing module, in order to capture the response from the target machines over the network, or to respond to incoming packet from target machine to feed false information. For example DNS Spoofer captures incoming DNS request packets from machine(s) over the network and then sends a false DNS response packet in order to redirect the traffic to intended destination. Similarly Network Scanner sends various TCP, UDP, ICMP packets in order to get more information about the network such as Machine IP addresses, open TCP & UDP ports, services running over the ports of different machines over the network etc.

### 3. Packet Traversing

The application has the functionality to create & send custom packets and also capture incoming packets. This module will further allow the application to parse the information from the captured incoming packets and display the information for the user in a readable format. This module will be used by Network Scanner and TCPDump. TCPDump would parse the captured packet to display the user common information such as Source IP & port, destination IP & port, and packet data. Network scanner parses information to retrieve information such as identifying host machines over a network, identifying open ports on a target host, determining the operating system of the target hosts on the network, applications that are running on the target hosts on the network etc.

### 4. Prototyping

Given the exploratory nature of this practicum, I will be going with an incremental software prototyping approach to development. Small-scale mock-ups of each part of the application outlined above will be developed following an iterative modification process until the prototype evolves to meet the application requirements.

For instance, the first task would be to determine whether assumptions are valid as the completion of the project depends entirely on their validity. This test would require creating a simple packet capturing application in Android and using a stock (non-rooted) device to test the application & similarly also using a rooted device to run the application. If the application runs on the non-rooted device this will determine that there will be no need for a device to be rooted and the application will run on all Android devices. If the application only runs on the rooted device, the application will only be targeted for rooted devices and will also help me determine whether this project can be finished. This will then expand into developing basic functionality of different sub-applications. However, because of the exploratory nature of this practicum, a rigid development and integration plan is not feasible. This is why I will be focusing on creating iterative prototypes, slowly adding features to each iteration until a base level of functionality has been achieved.

- **Data Exfiltration Tool**

Similar to the Android Security Suite, the development of this application can be divided into several discrete modules or prototype application, which then can be integrated together to make two separate applications 1) the controller/receiver application, 2) the backdoor data exfiltration application.

- 1) QR Codes

Encoding and generating QR Codes is the integral part of this application. This module in the application will be responsible for retrieving the requested information, encoding the gathered information to generate a QR code. This application will be developed as a separating working prototype and will only be focused on generating a QR code with fixed amount of data and then retrieving the data back from the QR code.

- 2) QR Encryption

One of the other main features of this application is ability to encrypt the generated QR codes in order to ensure that any unauthorized receiver cannot retrieve the contents encoded into the QR code. Similar to the QR code application, this module will be developed as a separate working prototype application which will take a QR Code as an input and provide the encrypted information back. The application will then take encrypted information as an input, decrypt it and then redraw the original QR code. Whether the application returned the correct QR code back, can be checked using a QR code scanner and compare the content decoded from the QR code.

- 3) Steganography

Other main feature of this application is the ability to embed specific content into a Host image file. For the purposes of this application this module will take two inputs 1) Encrypted QR information and 2) Host Image file. This module will then determine if the provided image file is large enough to embed the provided Encrypted QR code. If it is large enough it will embed the information and create a new file image file, if the image file is not large enough it will request input of new image file with required file size. This feature will also be developed as a separate prototype application.

- 4) Integration

Finally at this steps all the features developed in the form of prototype applications above steps will be integrated into two separate applications. One application will be developed for the controller machine from where the commands will be sent and the responses will be received, information will be extracted from

the host image, then decrypted to retrieve the original QR code and finally content will be decoded from the QR code.

Second application will be the backdoor which will receive commands from the controller machine, execute them, encode the data in a QR code, encrypt the QR code and finally embed the encrypted QR in the Host image before sending the image to the controller machine. Commands sent from the controller machine will be predefined and will include some common commands such as directory listing, file content etc. Completion of this step will provide the final application.

## VII. TECHNOLOGIES USED

There are various technologies that are going to be used for this project both at the software level for development and hardware level for testing. In this section I will be listing all the technologies that I am going to be using for this project.

- **Android Security Suite**

- 1) Operating Systems
  - a. Windows 8.1
  - b. Fedora Linux
- 2) Development Tools
  - a. Android Studio
  - b. Eclipse with Android Developer Tools
  - c. Bash Shell
- 3) Hardware Testing Tools
  - a. Nexus 9 Tablet
  - b. Nexus 5 Phone
  - c. Linux Machine
- 4) Software Testing Tools
  - a. Android Emulators
  - b. Wireshark
- 5) Programming Languages
  - a. Java with Android SDK

- **Data Exfiltration Tool**

- 1) Operating Systems
  - a. Fedora Linux
- 2) Development Tools
  - a. Eclipse
  - b. Bash Shell
  - c. Vim
- 3) Hardware Testing Tools
  - a. Linux Machine
  - b. VMware Player
- 4) Software Testing tools
  - a. Wireshark
- 5) Programming Languages
  - a. Java

## VIII. TESTING PLAN

The testing format in the Final Report will be similar to the format defined in this section, however two more columns will be added in the report which will include detailed information of the actual results and whether the test passed or failed.

### • Android Security Suite

The Security Suite application consists of 5 sub-applications. The testing plan for this practicum project includes a set of test cases for each of the sub-applications. In this section I have included set of example test cases for each of the sub-applications. The test cases include a test description, expected result, actual result and whether a test passed or failed. Final Report will include many more test cases that go through the all the features of each sub-application.

#### 1) Network Scanner (Optional)

Below are few examples of test cases for the Network Scanner application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Get a list of open TCP Ports on a Target Machine	The scanner will return port numbers that are open on the target machine. These ports only responded to TCP packets.
2	Get a list of open UDP Ports on a Target Machine	The scanner will return port numbers that are open on the target machine. These ports only responded to UDP packets.

#### 2) TCPDump Packet Capture (AndroShark)

Below are few examples of test cases for the TCPDump application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Capture all TCP packets	The application captures and displays all TCP packets incoming or outgoing
2	Capture all UDP packets	The application captures and displays all UDP packets incoming or outgoing



## 3) Packet Crafter

#	Test Case	Expected Results
1	Create a TCP packet for a target machine on port 80	The application creates and sends TCP packets for a specified target host with source port being 80. Since number of packets are not specified packets are sent every seconds until the process is manually killed.
2	Create a UDP data packet for a target machine on port 1024	The application creates and sends UDP packets for a specified target host with source port being 1024. Since number of packets are not specified packets are sent every seconds until the process is manually killed.

## 4) ARP Spoofer

Below are few examples of test cases for the ARP Spoofer application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Send ARP packets to a single Target Machine	The application creates and sends an ARP packet to the target machine every second. The application will includes its own IP as the source IP in order to trick the target machine into thinking the device is the router
2	Send ARP packets to multiple Target Machine	The application creates and sends an ARP packet to the all the target machines every second. The application will includes its own IP as the source IP in order to trick the target machines into thinking the device is the router

### 5) DNS Spoofer

Below are few examples of test cases for the DNS Spoofer application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Detect DNS request packets	The application captures all DNS packets and filters only DNS request packets
2	Send DNS response packets	The application parses DNS request packets and send a response packet back to the source machine

### • Data Exfiltration Tool

The Data Exfiltration tool consists of two application 1) the controller, 2) the backdoor. The testing plan for this practicum project includes a set of test cases for each of the application. In this section I have included set of example test cases for each of the application. The test cases include a test description, expected result, actual result and where a test passed or failed. Final Report will include many more test cases that go through all the features of each application.

Testing will be done for each of the prototypes as they are developed, however this section will only include the test cases and test results for the final application.

#### 1) The Controller

Below are few examples of test cases for the controller application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Send a command to the backdoor	The backdoor running on the victim machine receives the command
2	Receives the response from the backdoor	The image is sent from the victim machine is saved in the file system, also response packets sent from the victim machine are captured by Wireshark
3	Able to decode the content from the QR code – For example, results for “ls /bin” command	Compare the content decoded from the QR code, match the content sent by the backdoor. For example the content decoded from the QR code matches the

	directory listing for the bin directory.
--	--

## 2) The Backdoor


















Below are few examples of test cases for the backdoor application. The final report will contain many more test cases.

#	Test Case	Expected Results
1	Receives commands from the backdoor	Wireshark captures a packet received from the controller machine, also the backdoor application is able to parse the received packet and display the command.
2	Executes the commands received from the controller	The backdoor application lists the results of the command.
3	Create a QR code for the results of the command executed	The application creates and save a QR code image in the file system.

## IX. SCHEDULED ESTIMATES















In this section I have included a Gantt chart describing the various activities/tasks and allocated duration for each of the tasks. This project uses an incremental software prototyping approach to development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. I have allotted a time frame of 188 days to the completion of this project. Each day consists of minimum of 3 hrs of works which adds up to 564 hours in total.

The Gantt chart below is divided based on each application and sub-application and various tasks for each sub-application. Upon completion of all the tasks for the application, a complete Use case testing is done on the entire sub-application.

	 Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors ▾
0		▾ Major Project	188 days	Mon 6/1/15	Sat 12/5/15	
1		▾ Android Security Suite	111 days	Mon 6/1/15	Sat 9/19/15	
2		Research	10 days	Mon 6/1/15	Wed 6/10/15	
3		▸ TCPDump Packet Capture	36 days	Thu 6/11/15	Thu 7/16/15	2
11		▸ Packet Crafter	26 days	Fri 7/17/15	Tue 8/11/15	10
17		▸ ARP Spoofer	21 days	Wed 8/12/15	Tue 9/1/15	16
22		▸ DNS Spoofer	18 days	Wed 9/2/15	Sat 9/19/15	21
27		▾ Data Exfiltration Tool	34 days	Sun 9/20/15	Fri 10/23/15	26
28		▸ QR Codes	6 days	Sun 9/20/15	Fri 9/25/15	26
31		▸ QR Encryption	8 days	Sat 9/26/15	Sat 10/3/15	30
34		▸ Steganography	8 days	Sun 10/4/15	Sun 10/11/15	33
37		▸ Integration	6 days	Mon 10/12/15	Sat 10/17/15	36
40		Testing	6 days	Sun 10/18/15	Fri 10/23/15	39
41		Final Report	15 days	Sat 10/24/15	Sat 11/7/15	40
42		▾ User Manual	28 days	Sun 11/8/15	Sat 12/5/15	41
43						
44		▸ Network Scanner (Optional)	28 days	Sun 11/8/15	Sat 12/5/15	














## • Data Exfiltration Tool

27		▸ Data Exfiltration Tool	34 days	Sun 9/20/15	Fri 10/23/15	26	
28		▸ QR Codes	6 days	Sun 9/20/15	Fri 9/25/15	26	
29		Generating QR Codes	4 days	Sun 9/20/15	Wed 9/23/15	26	
30		Retrieving Content	2 days	Thu 9/24/15	Fri 9/25/15	29	
31		▸ QR Encryption	8 days	Sat 9/26/15	Sat 10/3/15	30	
32		Encryption	4 days	Sat 9/26/15	Tue 9/29/15	30	
33		Decryption	4 days	Wed 9/30/15	Sat 10/3/15	32	
34		▸ Steganography	8 days	Sun 10/4/15	Sun 10/11/15	33	
35		Embedding Data in an Image	4 days	Sun 10/4/15	Wed 10/7/15	33	
36		Retrieving data from the Image	4 days	Thu 10/8/15	Sun 10/11/15	35	
37		▸ Integration	6 days	Mon 10/12/15	Sat 10/17/15	36	
38		Controller Application	3 days	Mon 10/12/15	Wed 10/14/15	36	
39		Backdoor Application	3 days	Thu 10/15/15	Sat 10/17/15	38	
40		Testing	6 days	Sun 10/18/15	Fri 10/23/15	39	

## • Android Security Suite

1		▸ Android Security Suite	111 days	Mon 6/1/15	Sat 9/19/15		
2		Research	10 days	Mon 6/1/15	Wed 6/10/15		
3		▸ TCPDump Packet Capture	36 days	Thu 6/11/15	Thu 7/16/15	2	
4		Packet Capturing	5 days	Thu 6/11/15	Mon 6/15/15	2	
5		Filter Packets	4 days	Tue 6/16/15	Fri 6/19/15	4	
6		Displays Packet Summary	5 days	Sat 6/20/15	Wed 6/24/15	5	
7		Display Packet Details	6 days	Thu 6/25/15	Tue 6/30/15	6	
8		Write Captured Packets to a file	5 days	Wed 7/1/15	Sun 7/5/15	7	
9		Read & Display Packet from a file	5 days	Mon 7/6/15	Fri 7/10/15	8	
10		Testing	6 days	Sat 7/11/15	Thu 7/16/15	9	
11		▸ Packet Crafter	26 days	Fri 7/17/15	Tue 8/11/15	10	
12		Create and Send TCP Packets	6 days	Fri 7/17/15	Wed 7/22/15	10	
13		Create and Send UDP Packets	4 days	Thu 7/23/15	Sun 7/26/15	12	
14		Create and Send ICMP Packets	5 days	Mon 7/27/15	Fri 7/31/15	13	
15		Capture and Display Response Packets	5 days	Sat 8/1/15	Wed 8/5/15	14	
16		Testing	6 days	Thu 8/6/15	Tue 8/11/15	15	

		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors ▾	Resource Names
17			⚡ ARP Spoofer	21 days	Wed 8/12/15	Tue 9/1/15	16	
18			Send ARP packets to the Router and 1 Target	4 days	Wed 8/12/15	Sat 8/15/15	16	
19			Send ARP packets to the Router and Multiple Targets	5 days	Sun 8/16/15	Thu 8/20/15	18	
20			Execute TCP Dump to Capture Packets	6 days	Fri 8/21/15	Wed 8/26/15	19	
21			Testing	6 days	Thu 8/27/15	Tue 9/1/15	20	
22			⚡ DNS Spoofer	18 days	Wed 9/2/15	Sat 9/19/15	21	
23			Execute ARP Spoofer	4 days	Wed 9/2/15	Sat 9/5/15	21	
24			Capture and Filter DNS Packets	3 days	Sun 9/6/15	Tue 9/8/15	23	
25			Create and Send DNS Response Packets	5 days	Wed 9/9/15	Sun 9/13/15	24	
26			Testing	6 days	Mon 9/14/15	Sat 9/19/15	25	

## X. DELIVERABLES

The deliverables for this project will include but are not limited to the following items:

- Application Source Code
- Application Installation Instruction Guide
- Application APK file for quick installation
- User Manual
- Testing Documentation
- Test Supporting documents or scripts
- Project Proposal
- Project Final Report
- Subject Expert Approval Form

The deliverables however will not include any prototype applications that were developed in order to test a specific feature. The prototypes are developed to ensure that development of each feature possible before including that feature in the final integrated application.



## 3 | REFERENCES

- Akshay Choche and Hamid R. Arabnia. (n.d.). *A Methodology to Conceal QR Codes for Security Applications*.
- Android Operating System*. (n.d.). Retrieved March 2015, from Wikipedia: [http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)
- Kali Linux Tool Listing*. (n.d.). Retrieved March 2015, from Kali Linux: <http://tools.kali.org/tools-listing>
- Network security*. (2014, August 22). Retrieved from Wikipedia: [http://en.wikipedia.org/wiki/Network\\_security](http://en.wikipedia.org/wiki/Network_security)
- Somdip Dey, Kalyan Mondal, Joyshree Nath, Asoke Nath. (2012). *Advanced Steganography Algorithm Using Randomized Intermediate QR Host Embedded With Any Encrypted Secret Message: ASA\_QR Algorithm*. Retrieved from <http://www.mecs-press.org>
- Stephen Northcutt, Jerry Shenk, Dave Shackelford, Tim Rosenberg, Raul Siles, and Steve Mancini;. (2006, June). Retrieved March 2015, from SANS: <http://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635>
- TrendLabs. (n.d.). Retrieved from Trend Micro: [http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/how\\_do\\_threat\\_actors\\_steal\\_your\\_data.pdf](http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/how_do_threat_actors_steal_your_data.pdf)

## 4 | CHANGE LOG

This is a second version of the Project Proposal. In this version a lot of changes were made, more specifically a whole new application was added, alongside the application that was described in first version of the Proposal.

In this this version some of the concerns of the committee were addressed, specifically in the Innovation section of the proposal for Android Security Suite. The Innovative components of the applications were highlighted and the innovation behind them was described. Due to the addition of a whole new application, some minor functionality from the existing applications has been removed and one sub-application is now marked as “Optional”.

Apart from the application mentioned in the first version, a new application was included to the practicum which also includes some fairly difficult and innovative components. Full details about the application has also been included in this version.