** **

**PROJECT PROPOSAL**

ANDROID NETWORK SNIFFER

Bachelor of Computer Science (Digital Systems Security)

**Team SSP19/2B**

*Kan Kar Jun Alvin 5709659*

*Liao Weisheng Wilson 6212591*

*Nyein Soe 5710959*

**Supervisor**

*Mr. Premarajan P*

**Project Website**

[*https://android-network-sniffer.webnode.com/*](https://android-network-sniffer.webnode.com/)

**Contents**

1. Android Network Sniffer 3

1.1. Objective 3

1.2. Introduction 3

2. Network Sniffer 4

2.1. How does it work 4

2.2. What are the goals 5

2.3. Different usage of this sniffer tools 7

2.4. Different sniffer tools 7

2.5. How it can be used to analyze network 8

2.6. What are the key aspects of a sniffer application 9

3. How does it differ from other existing network sniffing tools 10

3.1. WireShark 10

3.2. PRTG 11

3.3. Packet Capture 12

3.4. Network Minor 13

4. Research Conclusions 14

5. Implementation(proposal) 15

5.1. Purpose 15

5.2. Functionalities 16

5.3. Refined Functionalities 17

5.4. Use Case Diagram 17

6. Project Timeline 18

7. Roles and Responsibilities 19

8. Team Members Profiles 19

9. References 20

1. **Android Network Sniffer**

**1.1. Objective**

The objective of this project is to create a network sniffer in an Android OS, where the program is able to listen to the network communication and break down the packets to readable format. For instance, a WEP program crack can be implemented to provide a more robust implementation.

**1.2. Introduction**

In general, network sniffing involves using sniffer tools that enable real-time monitoring and analysis of data packets flowing over computer networks. It can be a hardware device or a separate software program or a combination of both. It is also called as packet sniffing, snoop, network probe, packet analyzer, network analyzer, or protocol analyzer. Basically, it examines traffic on the network and takes snapshot copies of the packet data.

**2. Network sniffer**

**2.1. How does it work**

Packet sniffers work by intercepting and logging network traffic that they can 'see' via the wired or wireless network interface that the packet [sniffing software](https://www.lifewire.com/introduction-to-packet-sniffing-2486803) has access to on its host computer.

On a wired network, what can be captured depends on the structure of the network. A packet sniffer might be able to see traffic on an entire network or only a certain segment of it, depending on how the [network switches](https://www.lifewire.com/definition-of-network-switch-817588) are configured, placed, etc. On wireless networks, packet sniffers can usually only capture one channel at a time unless the host computer has multiple wireless interfaces that allow for multichannel capture.

Once the raw packet data is captured, the packet sniffing software must analyze it and present it in human-readable form so that the person using the packet sniffing software can make sense of it. The person analyzing the data can view details of the 'conversation' happening between two or more nodes on the network. Network technicians can use this information to determine where a fault lies, such as determining which device failed to respond to a network request.

Hackers can use sniffers to eavesdrop on unencrypted data in the packets to see what information is being exchanged between two parties. They can also capture information such as passwords and authentication tokens (if they are sent in the clear). Hackers can also capture packets for later playback in replay, man-in-the-middle, and packet injection attacks that some systems may be vulnerable to.

In addition, all networks use “packets” to send data. The size of a data file may be quite huge and sending it as a single packet would strain the network and cause congestion. The integrity of the data may also be affected. Hence, whenever a user sends a file or an email it is broken up into smaller parts or packets and then sent to the destination. Each data packet includes:

1. Destination address
2. Number of packets
3. Reassembly order
4. Source address

Once the data packet gets to its destination, all the footers and headers are removed, and the data is reconstituted. Every network and computer has a filter that discards any packets that aren’t addressed to it.

In Network Sniffing, the Packet sniffers intercept and log the network traffic through the packet sniffing software. Based on the capability of the software, a packet sniffer may be able to access traffic on the complete network or just a part of the network.

The packet sniffing software then analyses and converts it into a user-friendly form. This enables the hacker to function as a man in the middle and access the details of the communication between two parties/organizations. Unencrypted data is very vulnerable, and hackers can gain access to passwords, authentication tokens and other credentials. Hackers use the captured packets for man-in-the-middle and packet injection attacks.

Network sniffing, like SSL sniffing is widely used by hackers, but it can also be used for network troubleshooting. Our product includes this and other security options to ensure the safety of your information, so contact us today to learn more.

**2.2. What are the goals**

Typical use of network sniffer is to analyze network traffic and bandwidth utilization, so that underlying troubles in the network can be identified. There are, however, two directional usages of sniffer which have coexisted since it was first produced:

Positive usage of a sniffer is also its regular usage, which has as its objective the desire to maintain the network and keep it working normally.

1. Capturing packets;
2. Recording and analyzing traffic;
3. Decrypting packets and displaying in clear text;
4. Converting data to readable format;
5. Showing relevant information like IP, protocol, host or server name and so on.

Not all packet sniffing software product have the same functions; some sniffers can analyze hundreds of protocols whereas others can only deal with one or two. The most common protocols analyzed by sniffer are TCP/IP, IPX, DECNet-Ordinarily, a sniffer is used as assistant tool of the network engineer for monitoring and analyzing a network, detecting intrusion, controlling traffic or supervising network activity. IT should be noted that such features may also be utilized by hackers as a snooping tool to break into other computers.

Negative usage of a sniffer is well known as its harms to network security:

1. Catching password, which is the main reason for most illegal uses of sniffing tool;
2. Capturing special and private information of transactions, like username, credit ID, account, and password;
3. Recording email or instant message and resuming its content;
4. Some Sniffers can even modify target the computer's information and damage the system;
5. Interrupting the security of a network or to gain higher level authority.

With more and more hackers using of packet sniffers, it has become one of the most important tools in the defense of cyber-attacks and cyber-crime.

*Monitoring network usage* – Packet sniffers are great at monitoring the network usage at any given time, helping Network Managers identify whether a particular network is normal or congested. Also, making it possible to identify bottlenecks within the network and identify and improve the performance with infrastructure upgrades.

*Identifying problems* – As mentioned earlier, packet sniffers can identify network-related issues. This is possible because a packet sniffer can analyze the conversation between two or more nodes in a network. So, in the event of a network error, the information captured by the packet sniffer can be used to identify the erroneous packets and pinpoint the node that failed to answer the request(s). Making it easy to identify faulty devices within the network in an efficient manner and providing the ability to take swift corrective actions.

*Detecting security loopholes* – A disturbing fact about packet sniffers is their ability to work as spying tools. They also help the good guys, such as your Network Manager, by testing the vulnerabilities of a network. Once these vulnerabilities are detected, it is easier to remove the loopholes thus preventing the possibilities of hacking attempts.

**2.3. Different usage of this sniffer tools**

Based on our research, we have found several packet/network sniffing tools which are currently being used and they are as mentioned below.

* SolarWinds Packet Analysis Bundle
* WireShark
* PRTG Network Monitor
* Steel Central Packet Analyzer
* Tcpdump
* Network Miner
* Kismet
* Fiddler
* EtherApe
* Packet Capture

Each type of tools has their own functionalities and to use some of them, the user will have to pay in order to gain access to the full feature of the respective tools.

**2.4. Different sniffer tools**

Based on our research, we have found several types of sniffers and they are as mentioned below.

* *ARP Sniffers:* In ARP sniffing, the network traffic is not sent to the hosts, but it is sent to the ARP cache of both network hosts, which is then forwarded to the network administrator. ARP Sniffers sniff the data when it is in the ARP cache. The ARP sniffer is popular among hackers, as the data captured in the cache allows them to create a map of the IP addresses and their associated MAC addresses. This map allows the hackers to conduct packet-spoofing attacks, search for router-based vulnerabilities and ARP poisoning attacks.
* *IP Sniffers:* IP Sniffers sniff all the data that corresponds with a specific IP address filter. This allows capturing of specific data packets for analysis and diagnosis. This method is also popular among hackers, who use it for stealing data and also for stealing the TCP session. They also use this to create fake TCP sessions, act as a man-in-the-middle and unleash MitM attacks.
* *MAC sniffers:* MAC sniffers sniff all the data that corresponds with a specific MAC address filter.
* *LAN sniffer*s: These are deployed on internal LANs and they have the capability to scan the complete IP range.
* *Protocol sniffers:* These are used to sniff data that are related to the network protocols that are used on the network.
* *Web password sniffers:* Hackers steal http sessions parse them to acquire login credentials, user IDs and passwords. While many websites protect their external facing web pages/websites with [SSL](https://securebox.comodo.com/ssl-sniffing/network-sniffing/key5sk1=fbdf6f7639540dba2706b006f150b765c0a209ec), they do not use SSL or use lesser secure encryption for their internal web pages. Hackers can exploit this vulnerability.

**2.5. How it can be used to analyze network**

For instance, SolarWinds is used to monitor network performance and have an overall view of what's going on in your network. What this means, more plainly, is it pays mind to more of the pure motility of the network. Transmission speeds and rates, packet transmission reliability, and even comes pre-configured with a wide variety of visual aids and sharp looking charts to make irregularities easier to spot.

Its counterpart, the Network Analyzer, again with a self-explanatory name, is more focused on the traffic itself. While the Performance Monitor is focused more on the overall view of the network's performance, the Network Analyzer is paying a lot more attention to the network on a more granular level.

In particular this part of the program ferrets out the [bandwidth hogs](http://www.pcwdld.com/bandwidth-monitors-best-free-network-tools) and anomalies, sorted by merit of users, protocols, or applications.

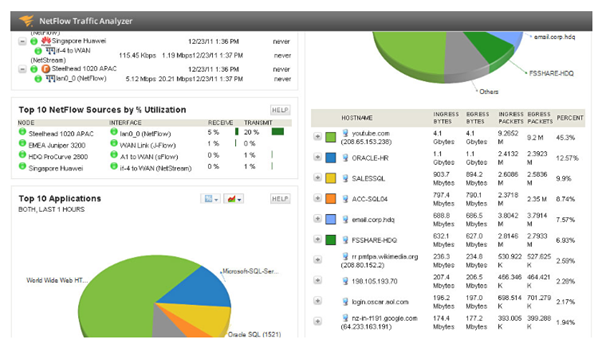


Figure 2.5.1. Example output of using SolarWinds

**2.6. What are the key aspects of a sniffer application?**

Some of the key aspects being used in network sniffing includes, PCAP, WinPCAP. These programs are used as an application programming interface (API) for packet-capturing. Sniffers also need to place the computer's network card in *promiscuous mode* to receive all the data that passes by, not just packets addressed to it.

**3. How does it differ from other existing network sniffing tools**

**3.1. WireShark**

WireShark is a windows-based network sniffing tool that show all the network information of the network the device is currently connected to. WireShark is able to provide details like the resolved addresses, protocol hierarchy status, connections, endpoints, packet lengths and the IO graphs.

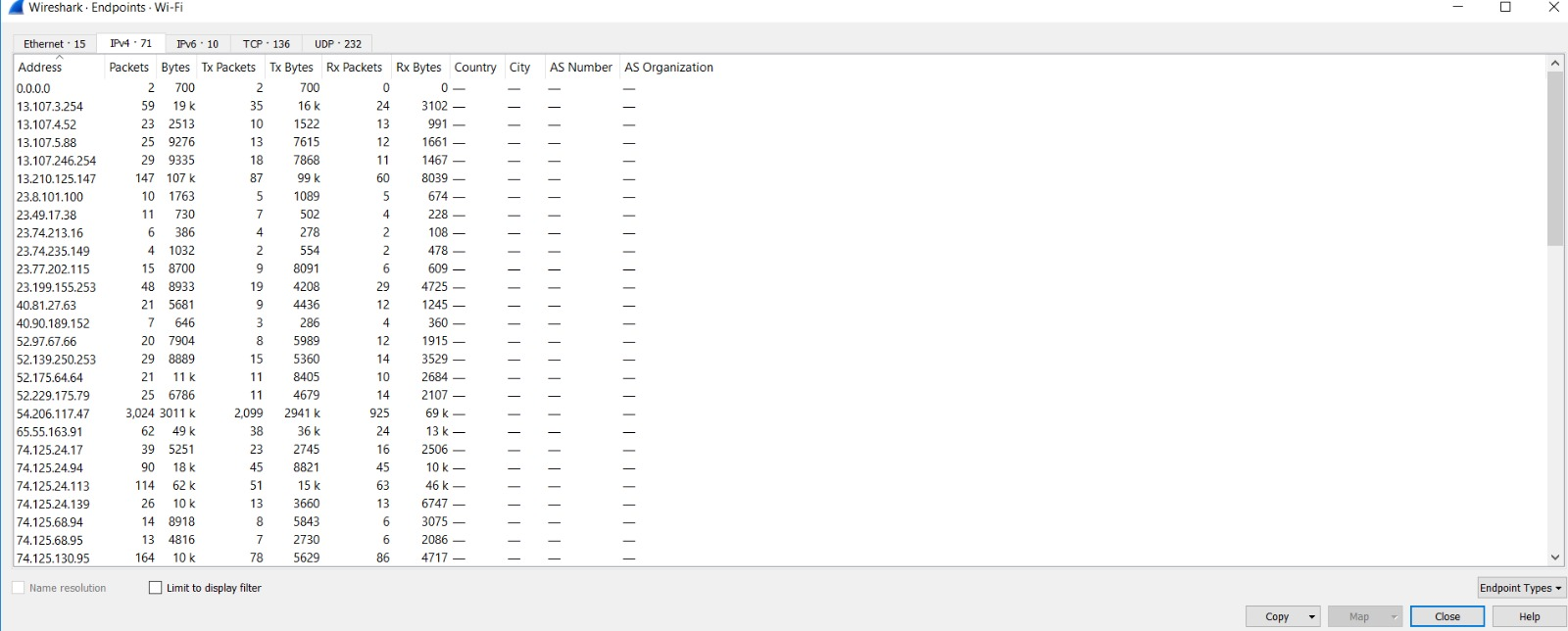


Figure 3.1.1. Features of WireShark

**3.2. PRTG**

PRTG is an android based app that provides information on the devices and services that are connected to the network. These includes information on servers or FTP services that are on the network. Example of these information are the ping, model number, serial number of the devices, and the number of files on the FTP server.

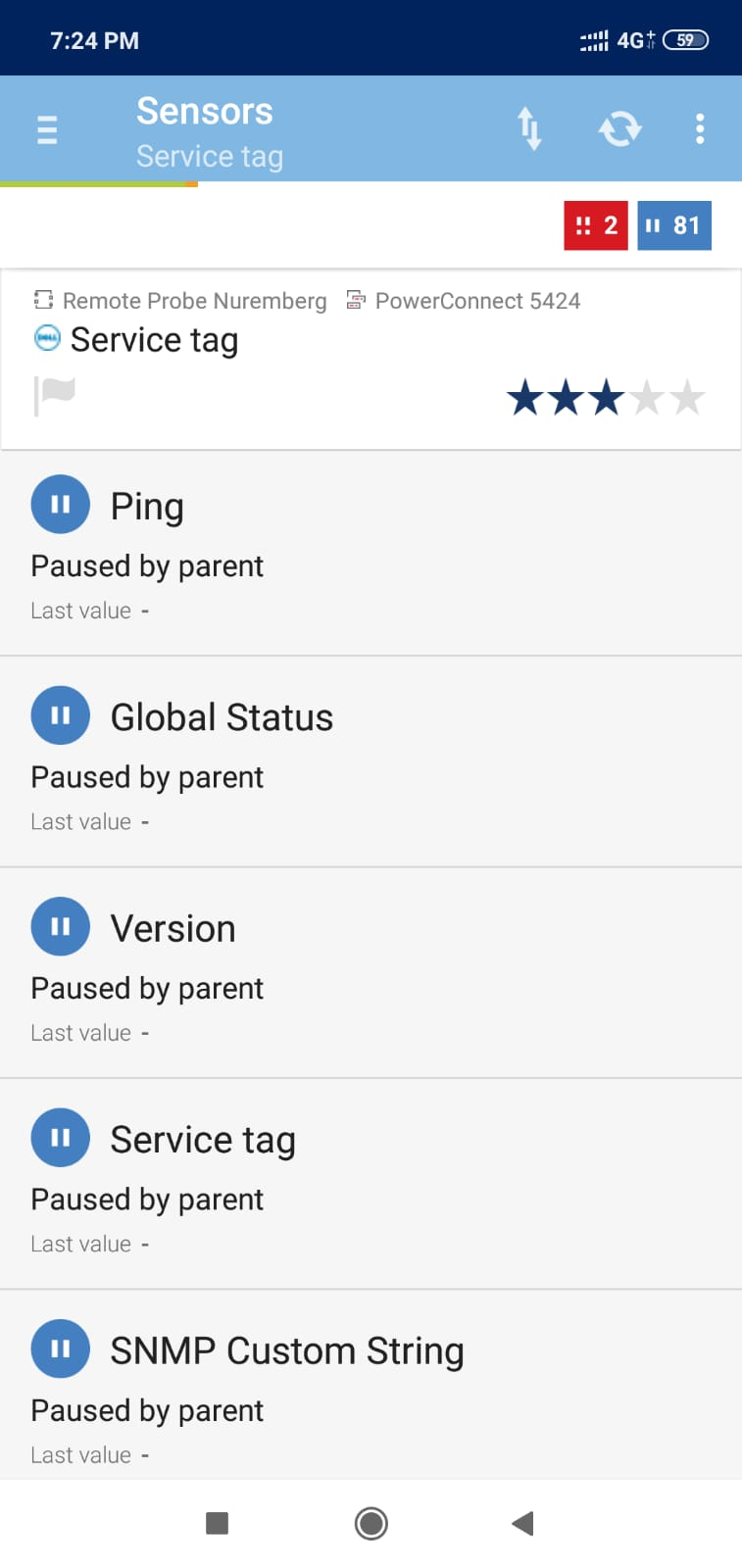


Figure 3.2.1. Features of PRTG

**3.3. Packet Capture**

Packet capture is an android based app that provides information on the usage of the network by the various apps and processes on the device. This information includes the security standards, MAC address, as well as the server details.

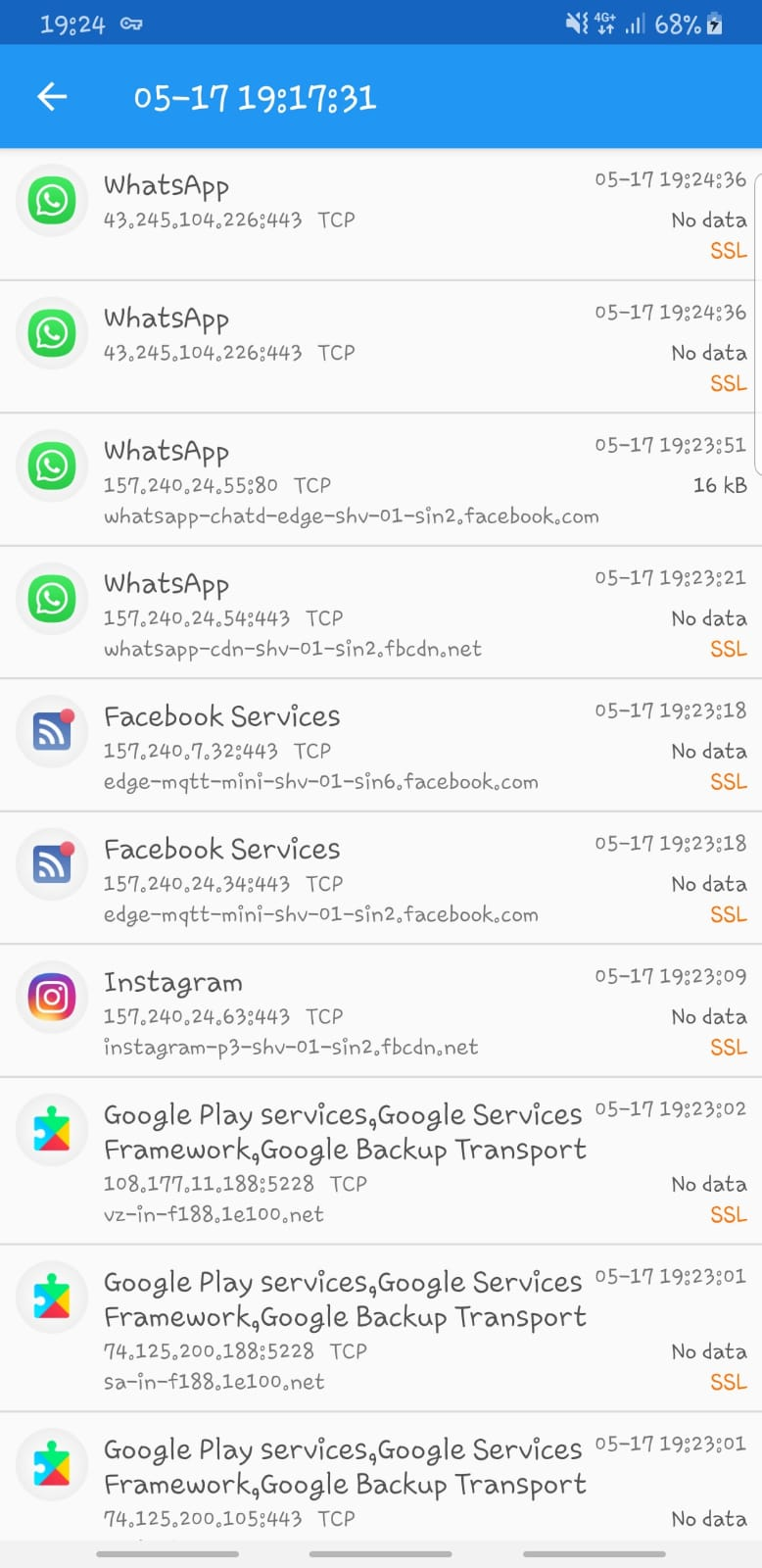


Figure 3.3.1. Features of Packet Capture

**3.4. Network Miner**

Network Miner is also a windows-based network sniffing tool that shows the packet information of the network that the device is connected to. It provides information like the packets sent, these packets show associated packets and if there are ongoing sessions as well as the packet size.

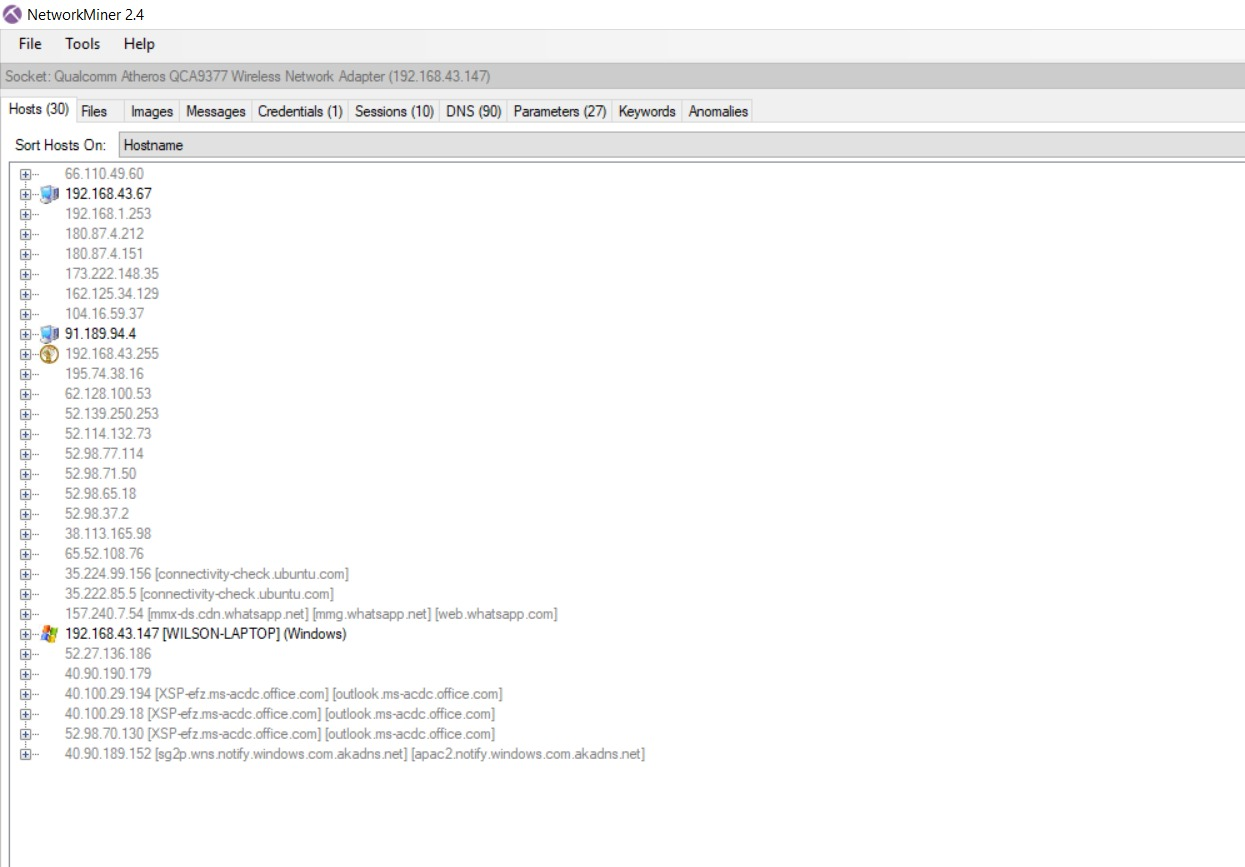


Figure 3.4.1. Features of Network Miner

**4. Research Conclusions**

After going through all the research, we have found out that there is a variety of network sniffers. Some of them have similar functions but some software provides better function and a more detailed information regarding the packets. In addition, different software are based on different OS/platform. Some of them are based on windows, some in IOS and last but not least, in androids. We have also learned that network sniffers are tools which intercept logs and analyse the traffic. For instance, as data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyses its content according to the appropriate RFC or other specifications.

From a search through the different types of network sniffers available in the market currently, the network sniffers used in android platform and Windows platform are mainly for high level usage, where the purpose of the application is to provide IT professionals real-time monitoring of the network. However, these applications usually provide detailed information only, it does not provide a more interactive view like diagrams and chart presentations.

Through our research, our goal is to create an application, aimed at developing a powerful android network sniffer that allows IT professionals to empower their android devices. However, we will add in more features like including interactive presentations in the graphical manner, as well as making the application more user friendly. The application should provide detailed information on the network, which IT professionals can make use of the information acquired to assist them in their daily work, or to help them work more efficiently.

**5. Implementation(proposal)**

**5.1. Purpose**

The purpose of the application is to develop an android network sniffer so that network administrators are able to bring around a portable android device and conduct network monitoring and troubleshooting.

Windows devices are usually more expensive and less portable as compared to android devices, since android devices powers our phones, which are our everyday device. By developing such an app, it empowers the android device to conduct higher level network monitoring.

In a company, where the IT department are required to monitor the network quality and status, devices need to be deployed to complete this task. Deploying a Windows laptop is more expensive compared to an android device. Android devices are also more portable and hence developing this application builds the android device to become more cost effective than the Windows laptop.

**5.2. Functionalities**

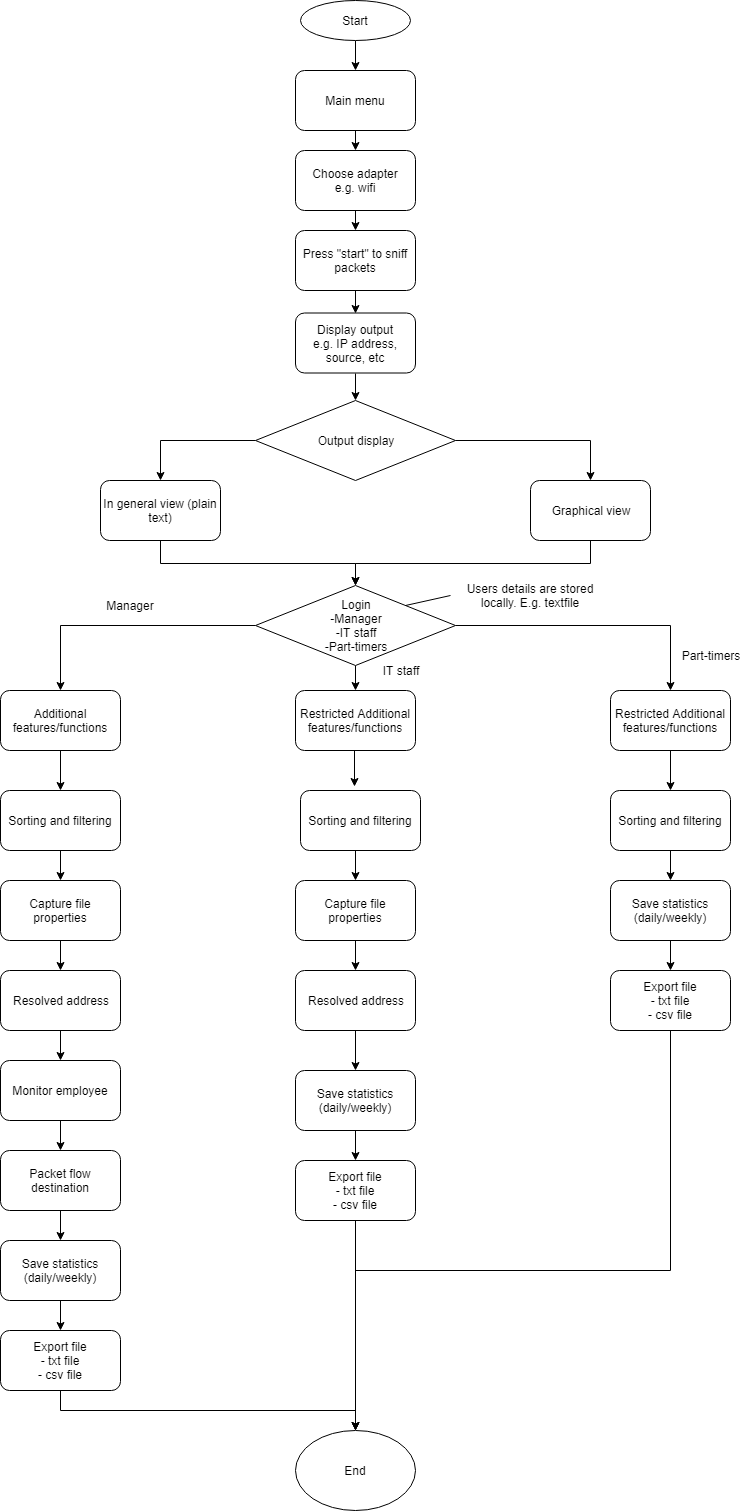


Figure 5.2.1. Program flow diagram

**5.3. Refined Functionalities**

**5.4. Use Case diagram**

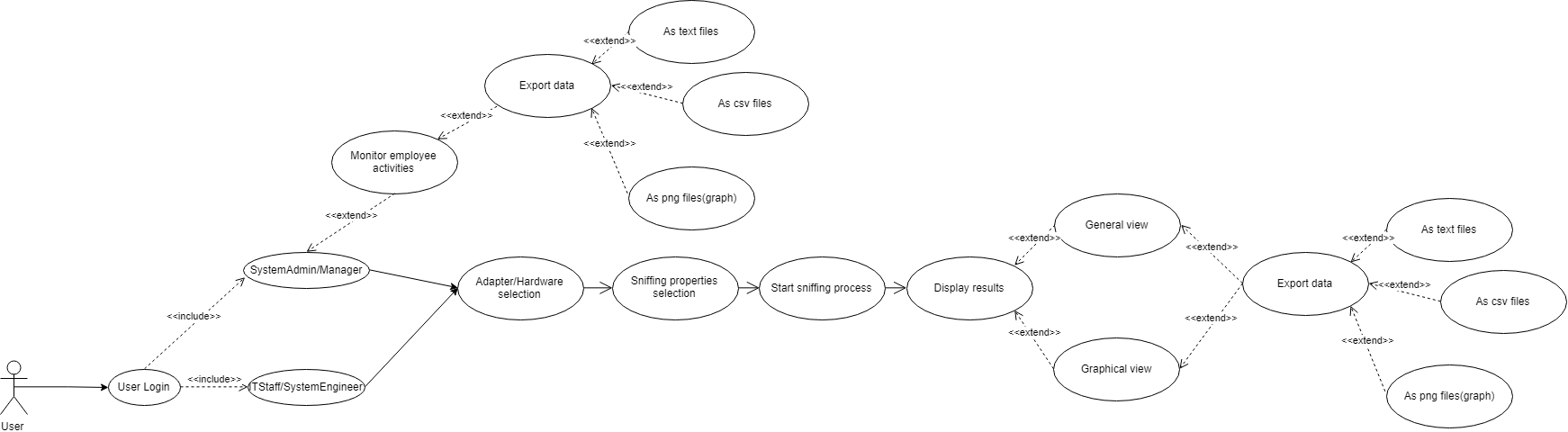
****

Figure 5.4.1. Use Case diagram

**6. Project Timeline**

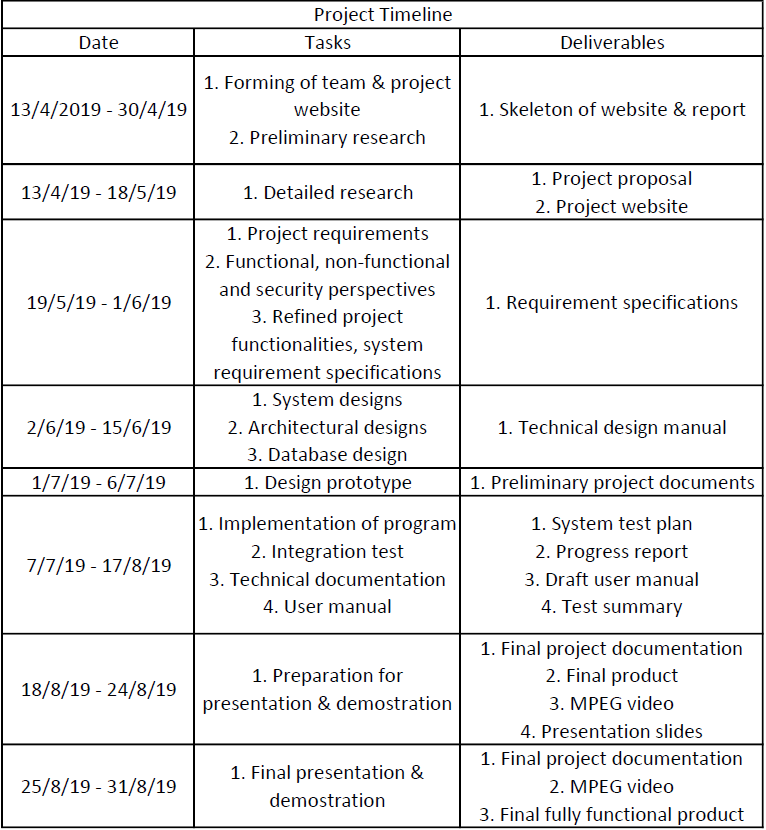
****

Table 6.1. Project Timeline

**7. Roles and Responsibilities**

|  |  |
| --- | --- |
| **Name** | **Roles/Responsibilities** |
| NYEIN SOE | Team Leader, Researcher, Implementation Designer, Tester |
| LIAO WEISHENG WILSON | Lead Developer, Implementation Designer, Tester |
| KAN KAR JUN ALVIN | Backup Developer, Implementation Designer, Planner, Technical Writer |

**8. Team Members Profiles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Number** | **Name** | **Phone Number** | **Email Address** |
| 5710959 | NYEIN SOE | 90721870 | nyeinsoe.wayne@gmail.com |
| 6212591 | LIAO WEISHENG WILSON | 97591961 | z.irashi.z@gmail.com |
| 5709659 | KAN KAR JUN ALVIN | 98346884 | xalvin.k@gmail.com |

**9. References**