National College of Ireland

Configuration Manual

MSc Research Project

MSCCYB

Irfan Nayeem

Student ID: x20252391

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Irfan Nayeem

Student ID: x20252391

Programme: MSCCYB **Year:** 2022-2023

Module: Research and computing

Lecturer: Imran Khan

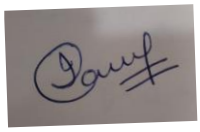
Submission Due Date: 15/12/2022

Project Title: Malware detection for URL security

Word Count: 1736 **Page Count:** 13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:



Date: 15/12/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Irfan Nayeem
Student ID: x20252391

The following steps will illustrate how to install the application on mobile devices run on the local machine.

Equipment's:

Compiling and organization of selected smart contracts in to separate files was conducted utilizing letest Andriod studio 3.11.17, java 11, Kali mint 20. The andriod studio install on kali mint with hardware: Asus Notebook pc, Processor Intel® Core™ i5-7200 CPU @ 2.60Hz, 2592 MHz, 4 Core(s), 8 Logical Processor(s) with 12GiB RAM.

1 Dataset:

This section provides information on the origin of the highlighted URLs. A significant dataset consisting of around 4.5 million URLs has been compiled by a number of machine learning researchers and web sources, including PishTank, OpenPish, BlockList, and Yahoo's directory listing, amongst others. In addition to sorting URLs into categories, information has to be collected so that we can assess whether or not they are secure or dangerous. As a consequence of this, they create software that is able to extract characteristics directly from the URL.

2 Installation:

Android studio: since the application is built and runs-on android studio environment so to test the application you must have the latest android studio, either on kali or windows.

Step 1: download the android studio.

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-2021.3.1.17-windows.exe Recommended	912 MiB	dd176791e15e921d4a3b3c9a251c61e5cfd28d75588fd717971dfbac030cd497
Windows (64-bit)	android-studio-2021.3.1.17-windows.zip No .exe installer	915 MiB	bdce14643efee37a4d892994b3329496460062f9c65ed870ff61a80267cb206a
Mac (64-bit)	android-studio-2021.3.1.17-mac.dmg	1000 MiB	4e10799559efc3445d61fb12bbf68e0a9801607a6114c6783bb26a93784d3150
Mac (64-bit, ARM)	android-studio-2021.3.1.17-mac_arm.dmg	989 MiB	0adbbddfa1e0e52e7bf21a5b560f60f8982ef82c0677db2d2ff7a2bd73ab156f
Linux (64-bit)	android-studio-2021.3.1.17-linux.tar.gz	937 MiB	89adb0ce0ffa46b7894e7bfedb142b1f5d52c43c171e6a6cb9a95a49f77756ca
Chrome OS	android-studio-2021.3.1.17-cros.deb	742 MiB	4d0c442d806fa8651c8e1baade6586c70aa46a61790aac0e91dfb4d5be7a7213

Figure1: Install any according to your operating system
Source: https://developer.android.com/studio

After downloading unzip, the downloaded file by name Android Studio

Try to open in terminal and with root privileges
Open the file with these commands

root@I-know-nothing:/home/cybertom/Desktop/Andriod# ls
android-studio android-studio-2021.3.1.17-linux.tar.gz

root@I-know-nothing:/home/cybertom/Desktop/Andriod# cd android-studio

root@I-know-nothing:/home/cybertom/Desktop/Andriod/android-studio# ls

bin build.txt Install-Linux-tar.txt jre lib license LICENSE.txt NOTICE.txt plugins product-info.json

root@I-know-nothing:/home/cybertom/Desktop/Andriod/android-studio# cd bin

root@I-know-nothing:/home/cybertom/Desktop/Andriod/android-studio/bin# ls

appletviewer.policy format.sh game-tools.sh icons.db inspect.sh lldb ltedit.sh profiler.sh restart.py studio.png studio.svg

brokenPlugins.db fsnotifier helpers idea.properties libdbm64.so log.xml printenv.py remote-

dev-server.sh studio64.vmoptions studio.sh

root@I-know-nothing:/home/cybertom/Desktop/Android/android-studio/bin# ./studio.sh

When you found the studio sh file open with linux execution command ./studio sh

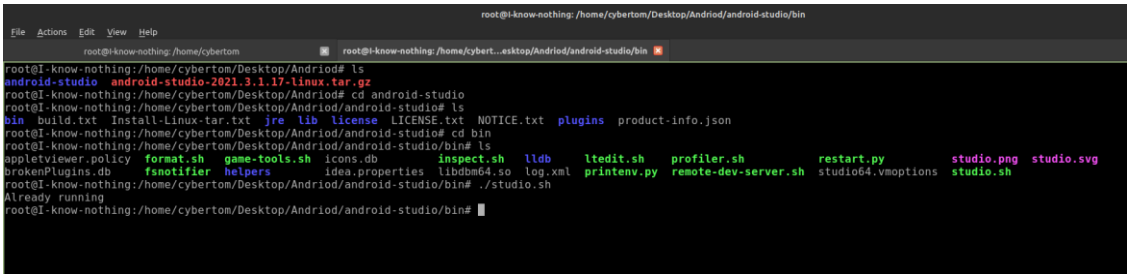


Figure2: When you execute the studio sh file it will appear to the install section

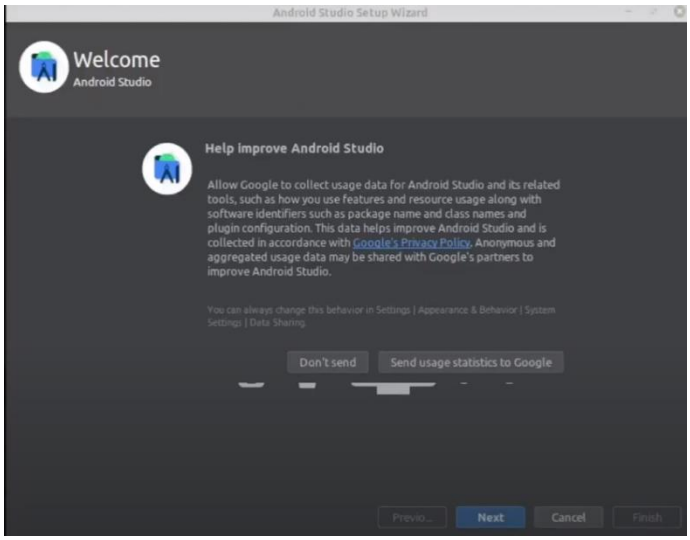


Figure3: Select next and installation process will begin

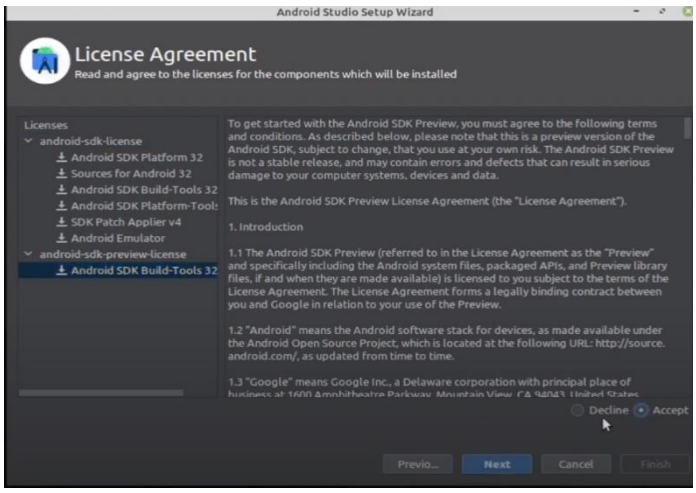


Figure4: Accept the conditions and select next

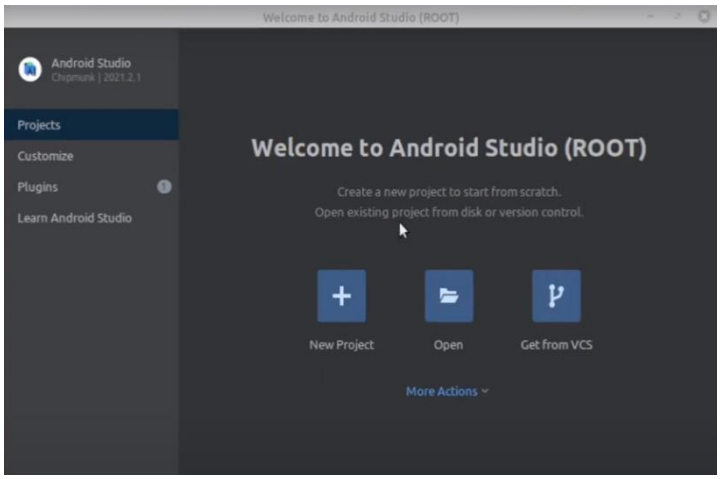


Figure5: Successful installed the android studio now you can add a new project

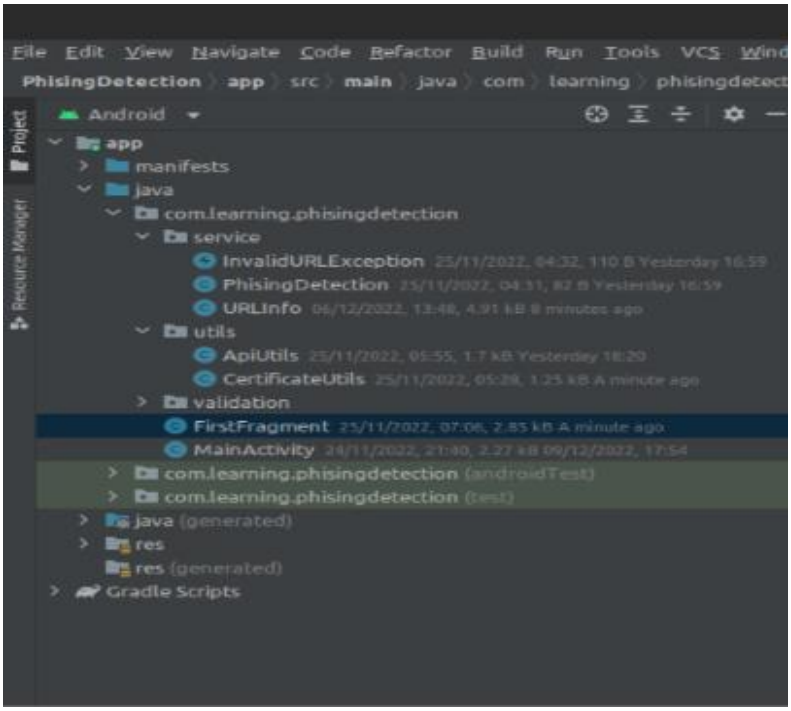


Figure6: import the project

Step 2: After successful installation extract the zip file in a folder

Step 3: run the android studio and wait for the landing page of environment to be load it may take time to load for the first time depends upon the configuration of the system

Step 4: once the android studio is started go to file > new > Import project

Step 5: select the folder and file where you extract the file:

Step 6: it will take little time to load and build up the file

Step 7: once project is loaded in android studio then it will appear like this on the left side of the windows.

Step 8: Although code is fully functional but for a good practice try to run to check any errors before running on emulator

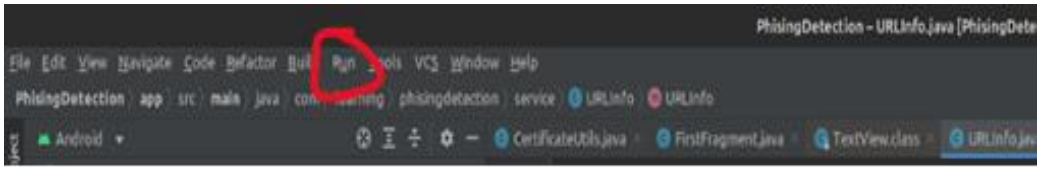


Figure7: Run the code

Step 9: run the emulator

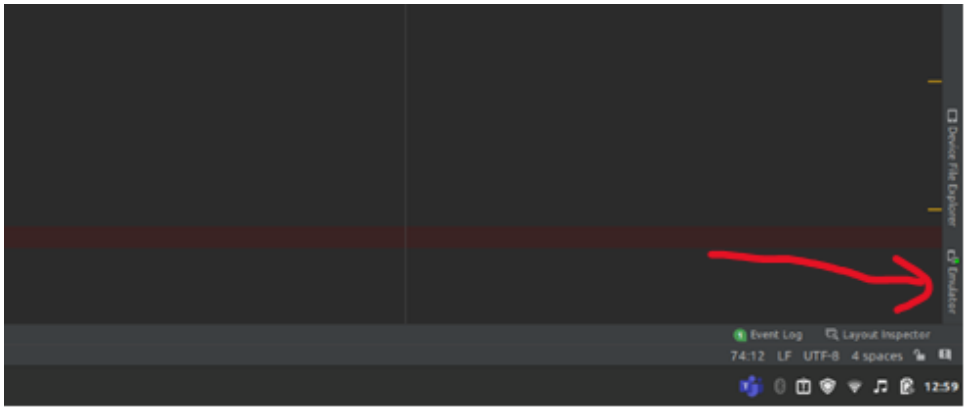


Figure8: position of emulator click to run it

Step 10: once you select it you will get the application open by name phishing detection.

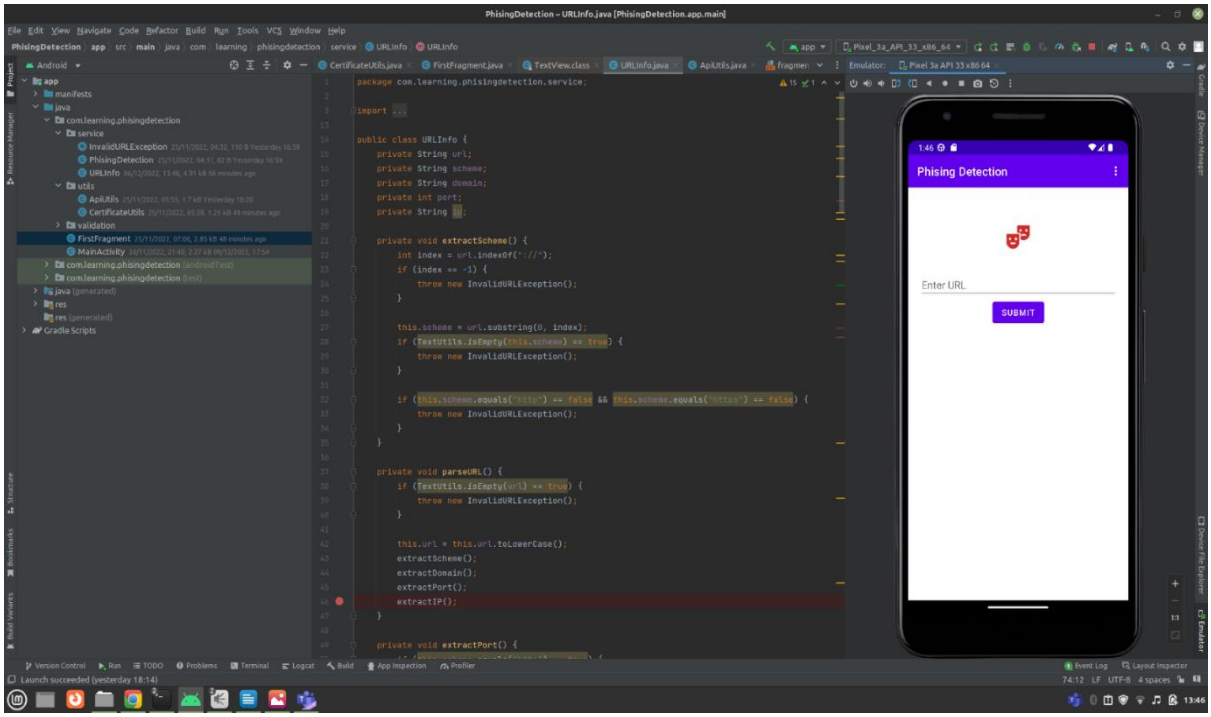


Figure9: This kind of interface you will have since the application requires internet to operate. Make sure you have a Wi-Fi connection for testing enter the URL for testing.

3 Implementation

1) The click on **SUBMIT** button is recorded in **binding.buttonFirst()** in **onViewCreated()** of **FirstFragment**.

```
binding.buttonFirst.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        binding.output.setText("");
        String url = binding.url.getText().toString().toLowerCase();

        if (handler == null) {
            binding.output.setText("Failed to check");
            return;
        }
```

Figure10:

2) here first we check if the url is valid with an inbuilt Validation class function **Validations.validURL(url) == false**

```
51
52          if (handler == null) {
53              binding.output.setText("Failed to check");
54              return;
55          }
56
57          handler.post(() -> {
58              if (Validations.validURL(url) == false) {
59                  mainHandler.post(() -> {
60                      binding.output.setText("URL not exists");
61                  });
62
63                  return;
64              }
65
```

Figure11:

3) if the URL is incorrect then show "URL does not exist" and end the process (**return;**) else if url is valid then:
   a) First, we convert the string url collected from line20 (fragment_first.xml) into URL format by using line67 **IF** all the conditions of the URL Info class are met.

```
                binding.output.setText("URL not exists");
            });

            return;
        }

        try {
            URLInfo urlInfo = new URLInfo(url);
            String status =  urlInfo.getStatus();
            mainHandler.post(() -> {
            binding.output.setText(status);
            });
        } catch (Exception ex) {
            mainHandler.post(() -> {
                binding.output.setText("Invalid URL");
            });
        }
```

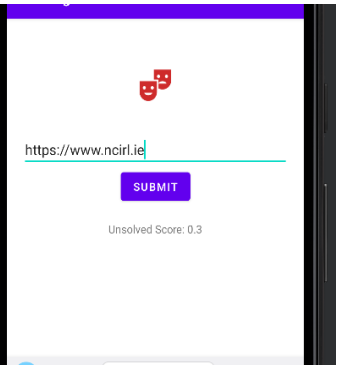


Figure12:

b) Conditions of URLInfo class line67 which are present in parseULR() function on line76 – URLInfo.java:

i) First, we check if the URL is empty then show error line38 else if it's not empty then we convert the url to LowerCase at line42 and then call the 4 main functions which disect the input url into scheme, domain, port and ip by using functions extractScheme(); extractDomain(); extractPort(); and extractIP();

```
private void parseURL() {
   if (TextUtils.isEmpty(url) == true) {
      throw new InvalidURLException();
   }

   this.url = this.url.toLowerCase();
   extractScheme();
   extractDomain();
   extractPort();
   extractIP();
}
```

ii) **extractScheme();** checks **indexOf("://")** line22, if :// is not found in the url then throw error else go ahead and check store the scheme line27. Again check if scheme is empty then throw error. Then check if scheme is equal to http or https line32. If none then throw error else scheme variable has the correct scheme.

```
private void extractScheme() {
int index = url.indexOf("://");
if (index == -1) {
   throw new InvalidURLException();
}
```

iii) **extractDomain();** this function extracts the domain part from the url

```
private void extractDomain() {
   String chkDomain = "";
   chkDomain = this.url.replace(this.scheme + "://", "");
   chkDomain = chkDomain.split("/")[0];
   chkDomain = chkDomain.split(":")[0];
   this.domain = chkDomain;
```

iv) **extractPort();** this function extracts the port in the url. If nothing is specified then it checks the scheme. If scheme is http, then assigns port as 80 and if https then assigns port as 443. If port is specified in the url then it collects the port and stores it in **port** variable.

```
private void extractPort() {
   if (this.scheme.equals("http") == true) {
      this.port = 80;
   } else if (this.scheme.equals("https") == true) {
      this.port = 443;
   }
```

v) **extractIP();** this function is the one which is checking the url from its IP at line88. If the domain has an ip address, then it means the site is live and we get the host address from line89.

```
public void extractIP() {
    try {
        InetAddress address =  Inet4Address.getByName(this.domain);
        this.ip = address.getHostAddress();
    } catch (Exception e) {
        throw new InvalidURLException();
    }
}
```

c) Once all the above conditions are met then we call **urlInfo.getStatus();** on line68 and store its result in the variable **status.** This function is calculating the url phishing score based on the Decision Tree and stores the score in variable **total** line120 as follows:

i) **getPointUrlLength()** - calculates the length of the full url and returns the score accordingly.

```
public float getPointUrlLength(){
    float point;
    if (url.length() < 54){
        point = 0;
    }else if (url.length() < 75){
        point = 0.025f;
    }else{
        point = 0.05f;
    }
    return point;
```

ii) **getDotCount()** - counts the number of dots present in the url and returns the score.

```
public float getDotCount() {
    String[] domains = this.domain.split("\\.");
    float point = 0f;
    if (domains.length > 3) {
        point = 0.5f;
    } else if (domains.length == 3) {
        point = 0.25f;
    }
```

iii) **getCertificateCount()** - checks the validity of ssl certificate using **CertificateUtils** class line177, where we open a connection to capture the certificate in **Certificate** class variable. Next, we check if this received certificate equals **"X.509"** which is a standard to check non-phishing official certificates.

iv) **getDomainAge()** - checks the domain's age and scores it accordingly. Age is calculated from line147 where we call the getDomainInformation(url) function of the ApiUtils class. This function connects to api

```
private float getDomainAge() {
float point = 0;

String url = this.scheme + "://" + this.domain + ":" + this.port;
int age =  ApiUtils.getDomainInformation(url);
if (age > 365) {
   point = 0f;
} else if (age > 90) {
   point = 0.25f;
} else {
   point = 0.05f;
}
```

(https://ip2whois.com/developers-api) which is a free service from ip2whois.com for developers to integrate their services in the developer's application. From this api, once we get the response in JSON object format, we extract **domain_age** from it and return the result.

4) **getPointSuffix()** - this function checks for hyphen ("-") and returns the score accordingly.

```
private float getPointSuffix(){
   float point;
   if (url.contains("-")){
      point = 0.05f;
   }else{
      point = 0;
   }
   return point;
```

a) Once we have the total calculated, then we find out the appropriate message to be shown to the user depending on the total.

```
String result = "unsolved";
if (total < 0.1){
   result = "TrustWorthy";
}else if (total < 0.3){
   result = "Fairly Legitimate";
}else if (total < 0.5){
   result = "Unsolved";
}else if (total < 0.75){
   result = "Suspicious";
}else{
   result = "Phishy";
}
```

References

“Download Android Studio & App Tools,” Android Developers. [Online]. Available https://developer.android.com/studio. [Accessed: 13-Dec-2022].

“Download Android Studio & App Tools,” Android Developers. [Online]. Available https://developer.android.com/studio. [Accessed: 13-Dec-2022].

“URL: What is the URL - javatpoint,” www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/url. [Accessed: 13-Dec-2022].

user3183user3183 3, AlexAlex 64477 silver badges1313 bronze badges, and user23307user23307 6, “What exactly is the Sh Command?,” Super User, 01-Feb-1957.

[Online]. Available: https://superuser.com/questions/97614/what-exactly-is-the-sh-command. [Accessed: 14-Dec-2022].

Yuan, J., Zhou, S., Lin, L., Wang, F. and Cui, J., 2020. Black-box adversarial attacks against deep learning-based malware binaries detection with GAN. In ECAI 2020 (pp. 2536-2542).

IOS Press.https://ebooks.iospress.nl/volumearticle/55183

Mirai DDOS attack. Journal of Information Systems Engineering & Management, 3(3), p.19.https://www.jisem-journal.com/download/malware-detection-and-mitigation-techniques-lessons-learned-from-mirai-ddos-attack.pdf