بسمه تعالی

گزارش تحلیلی بدافزار

Android.Trojan-Spy.Buhsam.A

فهرست مطالب

Contents

۱- مقدمه

امروزه با گسترش فناوری، دیجیتالی و سایبری شدن سیستم‌ها، حملات آن‌ها نیز به همان شکل شده‌اند و بدلیل اینکه بسیاری از سیستم‌ها علاوه برا اطلاعات شخصی یک فرد اطلاعات یک مجموعه از افراد مثلا یک سازمان مهم مالی و یا بانکی را دارند این مسئله اهمیت بیشتری پیدا می‌کند. یکی از این حملات که سیستم‌ها و اطلاعات آن‌ها را تهدید می‌کند، بدفزارها هستند. بدافزار در واقع نرم‌افزاری است که برخلاف یک نرم‌افزار عادی که نیاز ما را برطرف می‌کند، جهت تخریب یا سوءاستفاده از کاربران نوشته شده است. بدافزار (Malware) یک اصطلاح جامع و فراگیر است که به هر نرم‌افزاری اطلاق می‌شود که عمداً برای انجام اعمال غیرمجاز و مضر ایجاد شده است. بسیاری بر این عقیده هستند که یک بدافزار حتما باید بر روی سیستم شما در قالب یک فایل دانلود شود تا بتواند کارش را انجام دهد. بدافزارها از انواع روش‌ها و تکنیک‌های مختلف برای اجرا خود استفاده می‌کنند. برخی از بدافزارها از اجرای نرم‌افزارها یا عملیات‌های خاصی برروی سیستم جلوگیری می‌کنند، بعضی از آن‌ها سیستم شما را به عنوان یک سیستم قربانی برای سوءاستفاده و انجام عملیات تخریبی برروی سیستم‌های دیگر استفاده می‌کنند. برخی از بدافزارها نیز وجود دارند که صرفا برای جمع آوری اطلاعات شخصی کاربران طراحی شده‌اند، برای مثال اطلاعات مربوط به کارت شناسایی، شماره حساب‌های‌بانکی، رمزهای عبور و نام‌های کاربری و ... را جمع آوری و برای نویسنده آن بدافزار ارسال می‌کنند. بدافزارهایی که برای جمع‌آوری اطلاعات شخصی طراحی شده‌اند، ممکن است باعث تخریب در سیستم قربانی نیز شوند و این همان دلیلی است که شناسایی و حذف این بدافزارها را از سیستم‌عامل کاربر را به امری حیاتی تبدیل کرده‌است.

اخیرا تیم تحقیقاتی G DATA جاسوس افزاری کشف کرده‌اند که پیام‌های پیام‌رسان WhatsApp تحت تاثیر قرار می‌دهد، که در این گزارش اطلاعات یافت شده از این بدافزار تحلیل خواهد شد.

۲- بررسی بدافزار Android.Trojan-Spy.Buhsam.A

این بدافزار جدید که سیستم‌عامل اندروید را تحت تاثیر خود قرار می‌دهد از ویژگی‌های جاسوسی زیادی برخوردار است، مانند سرقت تاریخچه مرورگر، تصاویر، پایگاه داده پیام‌رسان WhatsApp و بسیاری از ویژگی‌های دیگر. مشخص نیست که این بدافزار دقیقاً چه هدفی را دنبال می‌کند. با این حال ویژگی‌های اصلی این بدافزار در این گزارش بررسی می‌شود.

این بدافزار با کلاس MainActivity.class اجرا می‌شود و سرویس OwnMe.class را فراخوانی می‌کند. کد اصلی این بدافزار در گیت‌هاب در صفحه‌ی کاربری earthshakira وجود دارد که در صورت نیاز می‌توان آن را مشاهده کرد.

کلاس OwnMe.class یک سرویس است. سرویس یکی از عناصر اصلی است که می‌تواند در توسعه یک برنامه اندرویدی بکار گرفته شود. این عنصر معمولا وظیفه انجام فعالیت‌های سنگین که در پس زمینه و مستقل از برنامه انجام می‌شود را دارد. در صورت اجرای تابع ( )startService، تابع ()onStartCommand فراخوانی می‌شود.

۲-۱- onStartCommand

ابتدا یک پیغام Toast نمایش داده می‌شود که نوشته شده است Service started بنابراین ما فرض بر عادی بودن استفاده برنامه می‌کنیم و متوجه اینکه این سرویس برای فعالیت‌های مشروع نبوده نخواهیم شد و به عبارتی شک نخواهیم کرد. مجرمان همیشه سعی دارد تا اقدامات خود را تا حد امکان مخفی کنند و عادی جلوه دهند. در ادامه برنامه متغیرهای زیادی چون upLoadServerUri، android_id، username، JSON objects ping و handshake تعریف می‌کند. شی handshake شامل اطلاعاتی با فیلد خالی باتری و cpu است که هنوز مقداری نگرفته است و ظاهرا در ادامه نیز مقداری نخواهد گرفت. بنابراین تصور می‌شود که این بدافزار هنوز در حال توسعه است و به معنای استفاده آنطور که باید باشد فعال نیست. هنگامی که اینکار انجام می‌شود برنامه تابع startExploit( ) را فراخوانی می‌کند.

۲-۲- startExploit & connectWebSocket

اگر بدافزار قصد اتصال به اینترنت را داشته باشد با از تابع ( )connectWebSocket استفاده کند. در صورت در دسترس بودن اینترنت بدافزار ارتباطی با آدرس ws://ipofthec۲:۸۰۸۰ خواهد گرفت. هنگامی که یک پیام از سرور دریافت شود تابع ( )onMessage نیز اجرا می‌شود و پیام را دریافت می‌کند. سپس برنامه متن دریافت شده که بصورت شی JSON می‌باشد را به پارامترهای موجود درون تابع نگاشت می‌کند. اگر این عملیات با لغو شود در پاسخ به سرور مقدار Null و یا خالی ارسال می‌شود.

۳- فعالیت‌های اصلی بدافزار Android.Trojan-Spy.Buhsam.A

طریقه انجام فعالیت‌های این بدافزار بدین صورت است که با دریافت پیام‌هایی از سمت سرور مرکزی که حاوی عبارات خاص هست فعال خواهد شد. بدین معنی که یک عبارت یا دریافت کرده و سپس کد مربوط به آن را در سیستم‌عامل فراخوانی می‌کند. نتیجه را گرفته و سرور مرکزی ارسال می‌کند . در ادامه به بررسی انواع این عبارات می‌پردازیم.

- **Screenshot**: با دریافت این عبارت Screenshot برنامه فعلا فعالیت خاصی انجام نمی‌دهد و ظاهرا متغیرهایی را بصورت خالی قرار می‌دهد. اما این تابع درون برنامه وجود دارد و مشخص است که این بدافزار هنوز درحال توسعه است.
- **Whatsapp**: اگر پیام از سمت سرور مرکزی حاوی عبارت whatsapp بود. برنامه تابع uploadWhatsApp را فراخوانی می‌کند. این تابع نیز همانطوری که اشاره شد به اقدام سرقت پایگاه داده این پیام‌رسان خواهد کرد و برای اینکار از Query زیر نیز استفاده خواهد کرد که بخشی از اطلاعات را مجزا به سرور مرکزی ارسال کند.

http://ipofthec۲/db/upload_whatsapp.php?username=%E۲٪۸۰٪۹D+username+%E۲٪۸۰٪۹D&device_id=%E۲٪۸۰٪۹D+android_id

- **Browserhistory**: اگر پیام حاوی عبارت Browserhistory بود، پیغام پاسخ به سرور شی JSON را با استفاده از تابع getHistory() پر خواهد کرد. این تابع یک مقدار رشته‌ای از id،عنوان، زمان، آدرس، نشان‌شده‌ها بازمی‌گرداند. با توجه به اینکه اشاره شد این بدافزار در حال توسعه است در این قسمت فعلا تنها سایت‌های نشانشده‌ی کاربر را برمی‌گرداند.
- **Contacts**: اگر پیام حاوی عبارت Contacts بود، پیغام پاسخ به سرور شی JSON را با استفاده از تابع getContacts() پر خواهد کرد. این تابع می‌تواند تمام اطلاعات مخاطبین تلفن را بخواند و به صورت رشته متشکل از متغیر _id ، display_name و شماره همراه ارسال کند.
- **Calllog**: اگر پیام حاوی عبارت Calllog بود، پیغام پاسخ به سرور شی JSON را با استفاده از تابع getCallLogs ( ) پر خواهد کرد. اگر برنامه مجوز android.permission.READ_CALL_LOG را نداشته باشد تابع مقدار «بدون مجوز» را بر می‌گرداند. در غیر اینصورت در قالب یک پیام رشته ای متشکل از number،name، type ، date، duration این تابع مقدار لازم را برمی‌گراند تا بصورت شی JSON نگاشت شود.
- **Fetch**: اگر پیام حاوی عبارت fetch بود، پیغام پاسخ به سرور شی JSON را با استفاده از تابع getBase۶۴(v۸.get("path")) پر خواهد کرد.

getBase۶۴ یک پارامتر از جنس Bitmap فراهم می‌کند که یک مسیر فایل عکس را نشان خواهد داد. اگر این طول این تصویر بزرگتر ۴۸۰ بود متناسب است در غیر اینصورت عملیات فشرده‌سازی انجام خواهد شد. در نتیجه این تابع یک مقدار رشته‌ای برای نگاشت به شی JSON که به فرمت base۶۴ است برخواهدگرداند.

- **Gallery**: اگر پیام حاوی عبارت Gallery باشد. یک رشته از تابع v۴.get(v۵).toString() برای نگاشت به شی JSON برگردانده می‌شود. این رشته حاوی مسیر، نام پوشه‌های کارت حافظه است. که بصورت المان‌های id و android_id که اطلاعات صفحه جاری است و total که اطلاعات کل است می‌باشد.

تابع mWebSocketClient.send() نیز اطلاعات را به اتصالات WebSocket ارسال می‌کند. این فرآیند آنقدر تکرار می‌شود تا حلقه به حداکثر ظرفیت خود برسد.

- **Camera**: اگر پیام حاوی عبارت Camera باشد تابع openCameraVideo() فراخوانی می‌شود که هنگام فراخوانی پارامترهای نوع دوربین (جلو یا عقب) و تعداد فریم عنوان خواهد شد.

اگر نسخه کیت توسعه کمتر از ۲۱ بود سپس دوربین خاص خود باز می‌شود و عکس می‌گیرد (این عکس به سرور منتقل نمی‌شود) در غیر اینصورت تابع takePictureR() فراخوانی می‌شود که اساساً وظیفه گرفتن عکس برای کیت توسعه نسخه‌های بالاتر از ۲۱ می‌باشد. این عکس بصورت base۶۴ کدگذاری خواهد شد و درون یک فایل JSON به WebSocket ارسال می‌شود.

- **UpdateBattery**: این عبارت موجب می‌شود تا تابع مربوط به آن مقدار باطری دستگاه و مقدار ظرفیت درحال استفاده CPU را برگرداند. این تابع هنوز بطور کامل پیاده‌سازی نشده و فعال نمی‌باشد.
- **No command provided**: در صورتی که هیچ دستوری وجود نداشته باشد یک شی JSON ساخته شده و عبارت error و no command found در آن قرار می‌گیرد و به WebSocket ارسال می‌شود.

۴- ماندگاری

در این بدافزار کلاس BootCompletedIntentReceiver.java که مشتق شده از کلاس BroadCastReceiver وظیفه ماندگاری و همیشه برقرار نگه داشتن این بدافزار را دارد. بطوریکه با فراخوانی تابع onReceive() بررسی می‌کند که intent دریافتی برابر با android.intent.action.BOOT_COMPLETED است یا خیر، اگر بود سرویس OwnMe.class را مجدداً فراخوانی می‌کند ؛ به عبارت دیگر بدافزار کلاسی که مسئول فعالیت‌های خرابکارانه است را در هنگام روشن شدن گوشی فعال می‌کند.

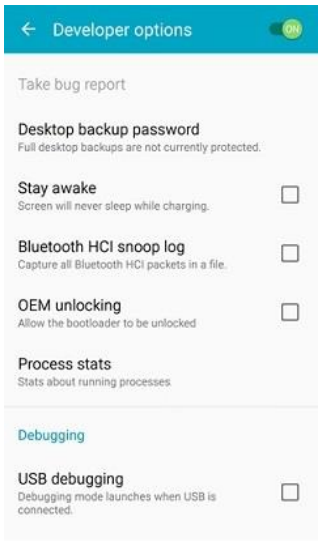
مقدار hash فایل :

SHA-۲۵۶
۴bed۸۹b۵۸c۲ecf۳۴۵۵۹۹۹dc۸۲۱۱c۸a۷e۰f۹e۸۹۵۰cb۹aa۸۳cd۸۲۵b۸۳۷۲b۱eaa۳d

۵- جمع بندی

شرکت G DATA در آخرین تحقیقات خود به بدفزار اندرویدی برخورده است که درحال توسعه است و اطلاعات زیادی از کاربر را می‌تواند به سرقت ببرد. فرمان سرقت این اطلاعات با ارسال یک پیام از سمت سرور مرکزی صورت می‌گیرد. بدافزارها همواره در حال پیشرفت هستند و در تلاشند منطعف‌تر شوند و همچنین مکانیزم‌های بیشتری جهت مقابله با مهندسی معکوس استفاده کنند.کاربر باید بکوشد تا با رعایت اقدمات امنیتی تهدیدها را به حداقل برساند. این نکات امنیتی که رعایت آنها بسیار اهمیت دارند عبارتند از :

- برای دستگاه همراه از احرازهویت با درجه امنیت بالا مانند PIN code و حسگر اثر انگشت استفاده شود.
- قابلیت USB Debugging، غیرفعال باشد، چرا که این مورد برای عیب‌یابی دستگاه و یا گوشی تلفن‌همراه است که خود می‌تواند دسترسی کاربر برای دور زدن احراز هویت را راحتتر نماید[۵].
- باید مطمئن شد که OEM Unlocking غیرفعال باشد[۵].



شکل ۱ غیر فعال بودن قابلیت USB Debugging و OEM unlocking

- در پیام‌ها، شبکه‌های اجتماعی و یا بطور کلی اینترنت در برروی لینک‌های ناآشنا کلیک نشود. گاهی این کلیک موجب نصب بدافزار و یا بازشدن port خواهد شد.
- باتوجه به تهدیدات مهندسی اجتماعی توصیه می‌شود که با آگاهی به افراد و مطلع نمودن آن‌ها از تهدیدها و روش‌های رایج، از دستگاه‌همراه کاربر و دارایی‌های آن که اطلاعات حساس کاربر نیز جزو آن است محافظت نمود.
- از دانلود برنامه‌های کاربردی از منابع ناشناس و غیرمعتبر خودداری شود. معتبرترین مکان برای دانلود برنامه‌های کاربردی برای سیستم‌عامل android، google play store و برای سیستم‌عامل ios، app store می‌باشد. بیشترین آلودگی‌های بدافزاری در پلتفرم اندروید از طریق منابع ثالث و متفرقه‌ی انتشار نرم‌افزار گسترش می‌یابند. منابعی مانند کانال‌های تلگرامی، انجمن‌های تلفن همراه، وبسایت‌ها و وبلاگ‌های متفرقه ....
- در هنگام نصب برنامه‌کاربردی، حتما به دسترسی‌هایی که برنامه‌کاربردی از سیستم‌عامل می‌گیرد دقت شود.

مراجع

[۱] https://twitter.com/LukasStefanko/status/۱۰۳۶۹۳۲۴۱۱۱۱۰۶۸۶۷۲۵

[۲] https://github.com/earthshakira/android_hack/tree/master/App/network/app/src/main/java/com/google/android/network

[۳] https://developer.android.com/guide/components/services

[۴] http://blog.teamtreehouse.com/an-introduction-to-websockets