הנדסה לאחור

מטלה 1

Agent Smith

התייחסות היסטורית ורקע כללי

באפריל 2019, חברת Checkpoint גילתה תוכנה זדונית (באנגלית: malware) הנקראת "הסוכן סמית" (באנגלית: Agent Smith). ביכולת malware זו, להתפשט ממחשב אל מחשב ולשכפל את עצמה. על תוכנה זו "להרעיל" את המחשב הראשוני שיחל את התפרצות הווירוס על ידי קובץ או מסמך מסוים.

התוכנה הזדונית Agent smith מנצלת את המחשבים שהיא מדביקה על ידי פרסומות שצפייה בהם מזכה את הצופה בכסף בכך שהיא מעבירה כסף זה אל התוקפים ויוזמי התוכנה הזדונית, כלומר - הווירוס מדביק את המחשב הנגוע במספר רב של פרסומות שהכסף על הופעתן מועבר לתוקפים.

זוהי malware מודולרית שמנצלת חולשות רבות במערכת ההפעלה אנדרואיד בכך שהיא מחליפה אפליקציות לגיטימיות המותקנות בה באפליקציות הנראות כמוהן ולמעשה תוכנות זדוניות.

הסוכן סמית אינו גונב מידע, אלא האפליקציות המתחזות שהוזכרו לעיל, מציגות מספר רב של פרסומות למשתמש וגונבות את הקרדיט שלכאורה מגיע לו עבור הצפייה אל התוקפים שיזמו את התוכנה הזדונית Agent Smith. תוקפים אלו, יכולים להשתמש ב-Agent Smith כדי לגרום נזק כבד ביותר לסמארטפון של קורבן מסוים.

נציין שה-malware שנקרא Agent Smith הדביק מספר עצום של מכשירים ברחבי העולם. המדינה שבה יש את מספר ההדבקות הרב ביותר היא הודו. מחקר שבוצע בחברת Checkpoint מצביע על שלפחות כ-15 מיליון מכשירים נגועים ב-Agent Smith בהודו.

כשחברת checkpoint חקרה את agent smith היא גילתה תכונות ייחודיות לה אשר גרמו לחוקרי התוכנה מהחברה להאמין שהיא מצאה malware חדשה מסוגה בתחילת התפשטותה. בנוסף, החוקרים גילו שזוהי copycat malware, כלומר – תוכנה זדונית המתחזה לאפליקציה לגיטימית כדי להונות את הקורבן.

לאחר חקירה לעומק, התברר כי השימוש הראשון של נוזקה זו כנראה היה בפיתוח על-ידי ארגון הממוקמת בסין. ההפצה העיקרית של התקיפה הייתה על-ידי משתמשים שהורידו אפליקציות תמימות לכאורה מה-play store כמו אפליקציה 9apps שמאוד פופולארית באזור אסיה. ל- agent smith אין מכניזם הפצה פורמאלי שמדביק את המשתמשים. מתקפה זו לא כוללת דירוג בחנות האפליקציות של ה-play store.

Country	Total Devices	Total Infection Event Count
India	15,230,123	2,017,873,249
Bangladesh	2,539,913	208,026,886
Pakistan	1,686,216	94,296,907
Indonesia	572,025	67,685,983
Nepal	469,274	44,961,341
US	302,852	19,327,093
Nigeria	287,167	21,278,498
Hungary	282,826	7,856,064
Saudi Arabia	245,698	18,616,259
Myanmar	234,338	9,729,572

רשימה של מדינות שבהן קיימת/הייתה תופעה של Agent Smith ממוינת לפי

פירוט טכנולוגי כללי

ה-payload של Agent Smith מוריד קבצים זדוניים, משתמש ב-C&C, חולשות ופרצות של המערכת ומציג פרסומות לקורבנות שהמערכת שלהן נדבקה.

כאמור לעיל, הווירוס יכול להיות מורד מחנות אפליקציות רשמית ולא רשמית.

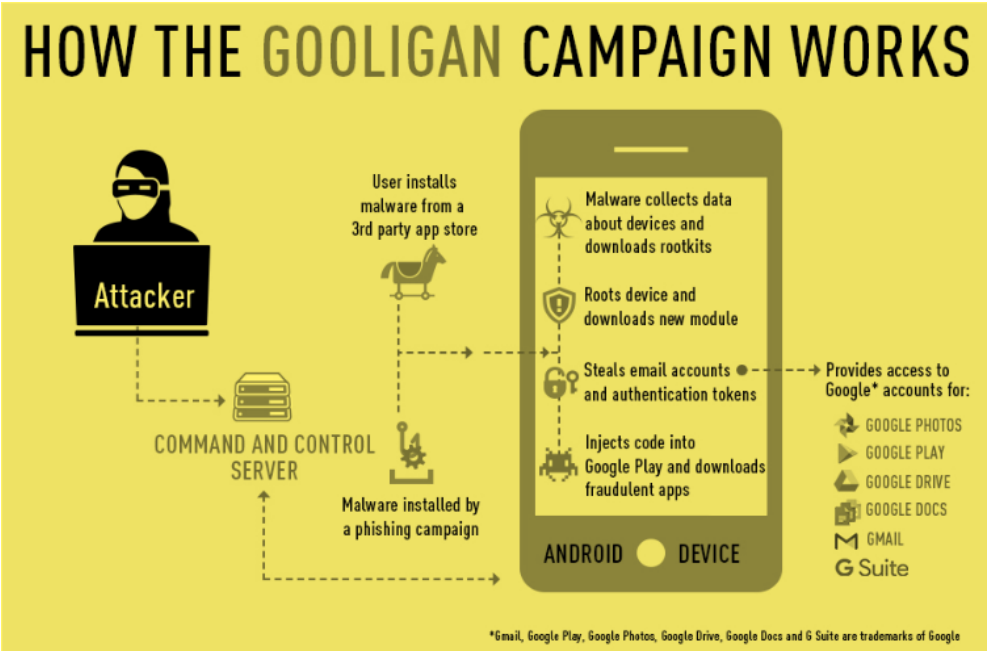
הווירוס מפעיל את הקובץ שאחראי ליצירת קשר עם שרת ה-C&C כדי להוריד את הרשימה של התוכנות הזדוניות שבעזרתן הוא מבצע את פעולותיו.

להלן רשימה באנגלית של יכולות הווירוס:

- It utilizes the Janus vulnerability to inject the “boot” module into the repacked application. After the next run of the infected app, the “boot” module will run the “patch” module, which hooks the methods from known ad SDKs to its own implementation.

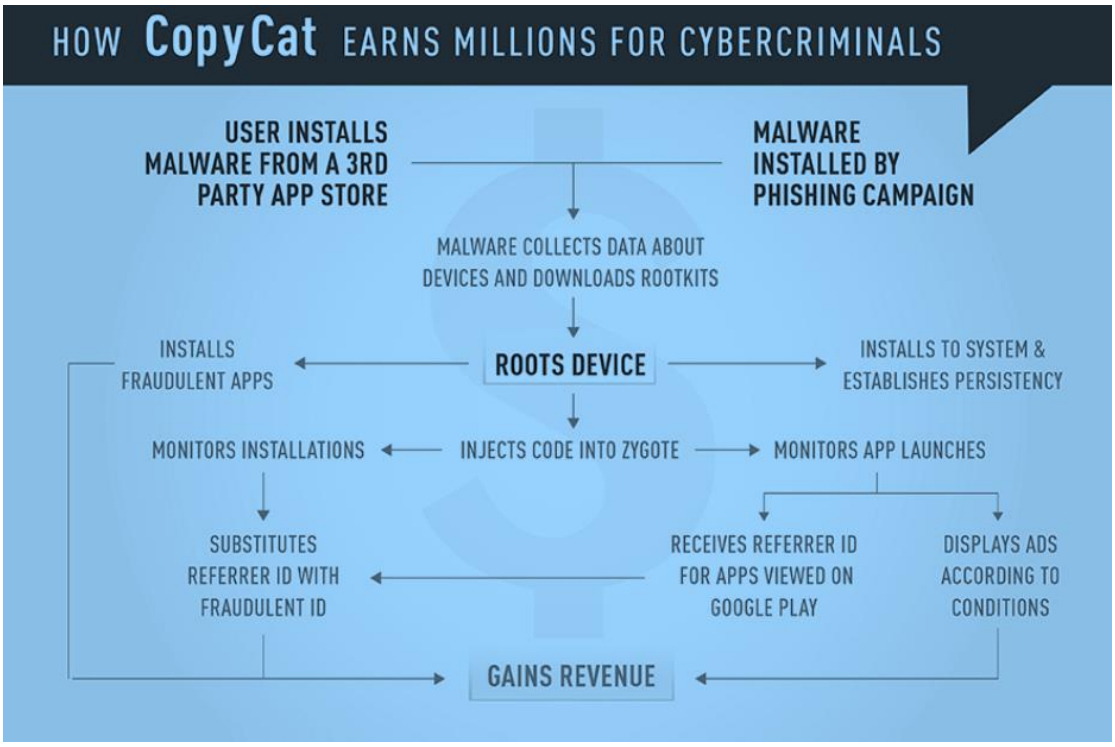
- It exploits a series of ‘Bundle’ vulnerabilities to install applications without the victim knowing.

- The 'AD' payload will display ads to the victims.
- the malware currently uses its broad access to the device’s resources to show fraudulent ads for financial gain. This activity resembles previous campaigns such as Gooligan, Hummingbad and CopyCat malware and can infect all smartphones updated beyond even Android v.7.



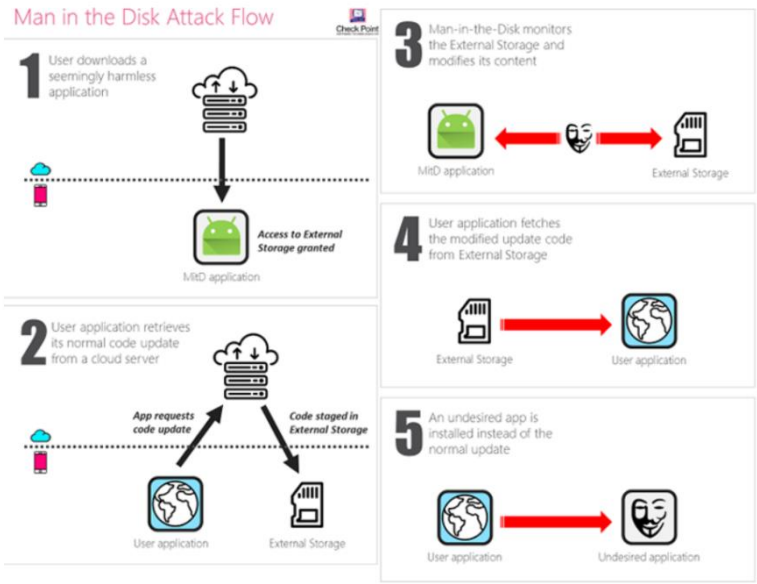
שיטת
ההדבקה של
המכשיר

הנוזקה Agent Smith מתחזה לאפליקציה לגיטימית (למשל – אפליקציה לאנשי קשר או משחק כלשהו) ובכך מצליחה להתפשט בחנויות אפליקציות צד שלישי (לא פורמליות). נוזקה זו מנצלת מספר חולשות מוכרות שקיימות במערכת ההפעלה android הכוללת את חולשת ה-Janus flaw ואת חולשת ה-Man in the Disk כדי "להזריק" קוד זדוני אל תו קבצי ה-APK של האפליקציות הלגיטימיות שמותקנות במכשיר המשתמש, לאחר מכן הקוד הזדוני שהוזרק מתחיל בפעולות כמו התקנה מחדש או עדכון האפליקציות הלגיטימיות לאפליקציות הזדוניות שמתחזות אליהן מבלי פעולה ישירה כלשהי של המשתמש. לכל הידוע, לא היה שימוש במודולים שהיו שייכים לנוזקות אחרות, את סקריפט הנוזקה נציג בהמשך .

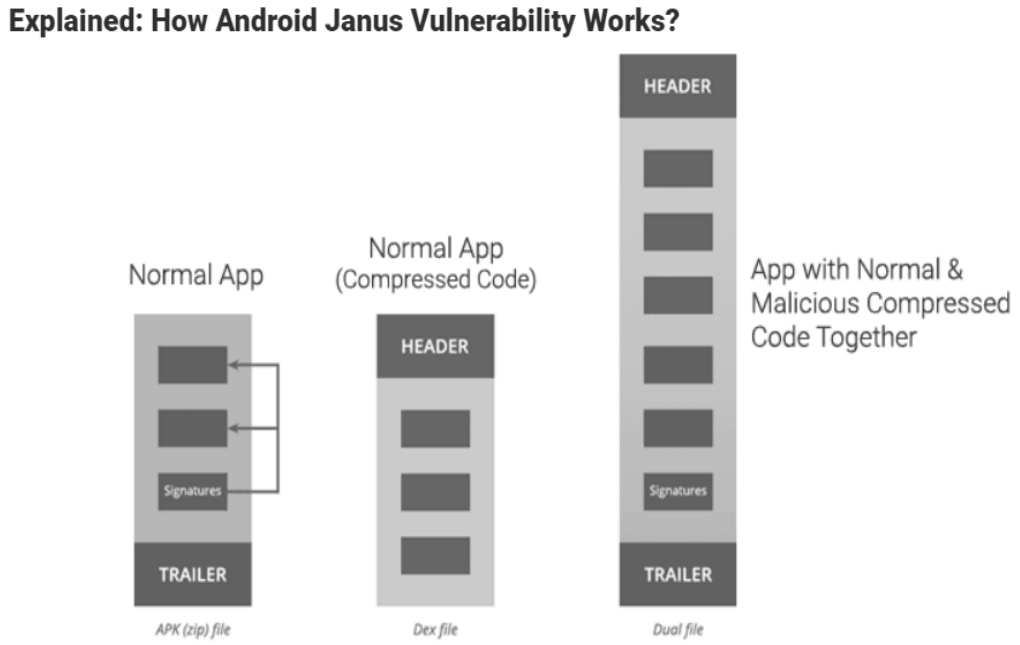


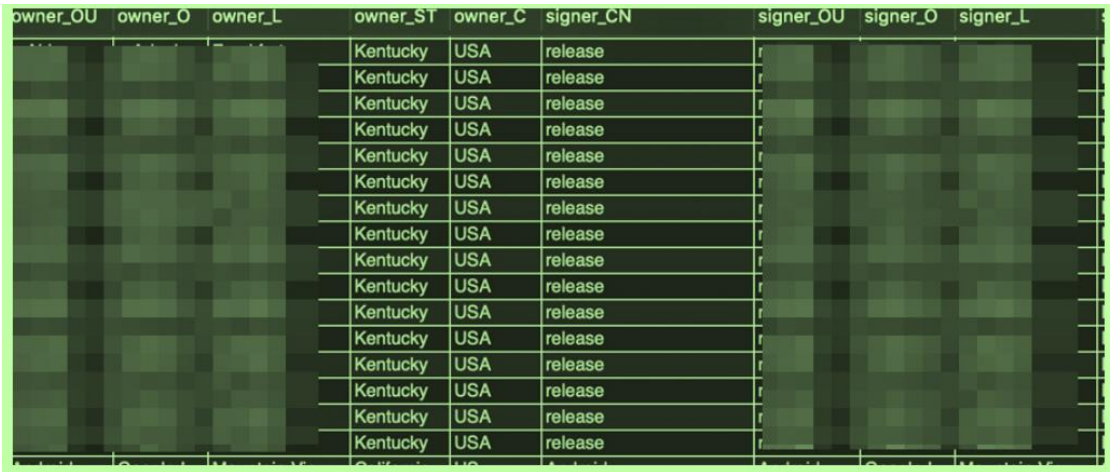
כאן ניתן לראות כיצד אפליקציות זדוניות מתחזות לאפליקציות לגיטימיות ומפעילות את הנוזקות השונות

אופן פעולת ה-Man in the Disk



מספר ה-CVE של הקורבן הוא: CVE-2017-13156. זוהי חולשה קריטית של מערכת האנדרואיד שברשות הקורבנות המאפשרת לתוקפים לשנות את הקוד של האפליקציות מבלי להשפיע על החתימה שלהן. החולשה מתבטאת בין היתר בקובץ שיכול להיות בו-זמנית קובץ APK וקובץ DEX תקינים. לחולשה זו קוראים חולשת Janus. חולשה זו קיימת מצאת מערכת הפעלה זו.





owner_OU	owner_O	owner_L	owner_ST	owner_C	signer_CN	signer_OU	signer_O	signer_L
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			
			Kentucky	USA	release			

Figure 1. Signer and owner listed in the certificate content of Agent Smith-infected apps are exactly the same.

All of the apps' malicious versions occupied more OS space and contained *android.support,multidex.MultiDexApplication*, allowing the attacker to use the automatic tool to repackage its codes to malicious ones. Trend Micro has notified Google of this; Google already removed the remaining apps at the time of publishing.

פירוט טכנולוגי פרטני על הנוזקה ומערך ההפצה

נוזקת הסוכן סמית' אינה כוללת hooking אבל כן יש בה שימוש בהזרקת קוד.

לאחר שמכשיר הקורבן "הורעל" על-ידי אפליקציה זדונית, היא מזריקה לו קוד זדוני לתוך קבצי ה-APK (קבצי ההתקנה) של האפליקציות הלגיטימיות שהאפליקציה הזדונית שמוזכרת לעיל מתחזה אליהן ולאחר מכן מתקינה ומעדכנת את אותן אפליקציות לגיטימיות אל אפליקציות זדוניות מבלי ידיעתו של הקורבן.

לסוכן סמית יש מבנה מודולרי וקבוע של המודלים שלהלן:

- Loader
- Core
- Boot
- Patch
- AdSDK
- Updater

לסוכן סמית יש 3 שלבים להדבקה / הרעלה:

<u>שלב ראשון:</u>

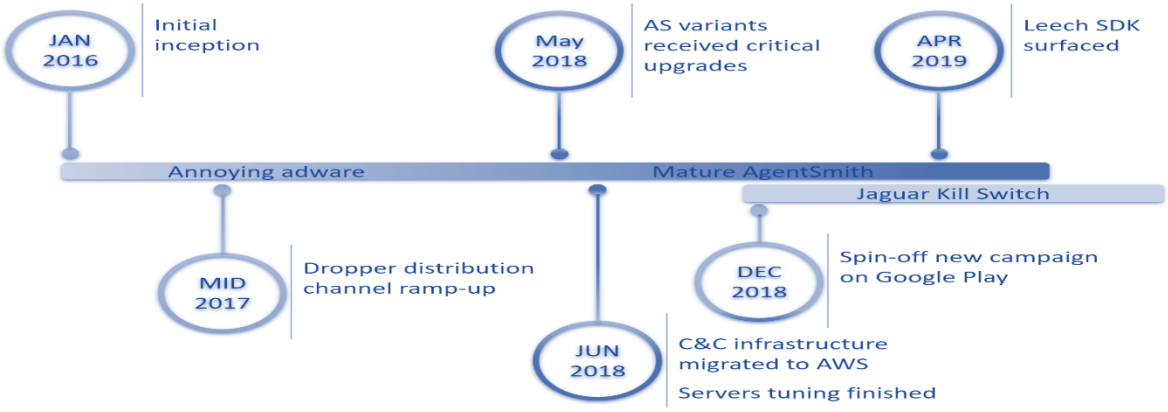
התוקף מפרסם את עצמו בפרסומת בחנות אפליקציות צד שלישי למשל ApkMirror ופרסומת מוצג דבר הנועד לפתות את הקורבן כמו "משחק חינם". לאחר שהקורבן יוריד את האפליקציה שהוצעה לו בפרסומת, האפליקציה שהותקנה תבדוק האם אפליקציות פופולאריות כמו WhatsApp מותקנות במכשיר ששמן מופיעה ברשימה מוכנה מראש אצל התוקף. אם אפליקציה כלשהו מהרשימה של התוקף נמצאת את המכשיר של הקורבן, אז הסוכן סמית' יתקוף אפליקציה זו.

<u>שלב שני:</u>

לאחר שהאפליקציה הזדונית הותקנה במכשיר היא באופן אוטומטי מצפינה payload זדוני לצורתו המקורית שהיא קובץ APK אשר משמש כליבה של מהלך התקיפה של הסוכן סמית'.

<u>שלב שלישי:</u>

האפליקציה הזדונית מובילה מהלך תקיפה אל מול כל אפליקציה לגיטימית שמותקנת במכשיר הקורבן ומופיעה ברשימת האפליקציות שאצל התוקף.



פעולות המבנה המודולרי של הסוכן סמית':

Loader:

האפליקציה הזדונית מכילה בתוכה מודול הנקרא Loader שמטרתו היחידה היא לפענח, לחלץ ולהריץ את המודול שנקרא Core.

Core:

לאחר ביצוע פועלת ה-Loader, מודול ה-Core מתקשר עם שרת ה-C&C של התוקף כדי להשיג רשימה של אפליקציות תקיפה פופולאריות. אם ה-Core מוצא התאמה המותקנת במכשיר הקורבן, אז מודול הקרנל מנסה להעתיק את ה-APK אל מכשיר הקורבן באמצעות חולשת האנדרואיד שציינו לעיל הנקראת Janus או על-ידי קימפול מחדש של ה-APK עם payload זדוני.

Boot Module:

מודול זה כלול ב-charger הזדוני אשר כבר חלק מהאפליקציה המקורית ועובד ממש כמו מודול של charging. מודול זה מחלץ ומפעיל את ה-payload הזדוני, קורא למודול ה-Patch כאשר הקורבן מריץ את האפליקציה ששונתה.

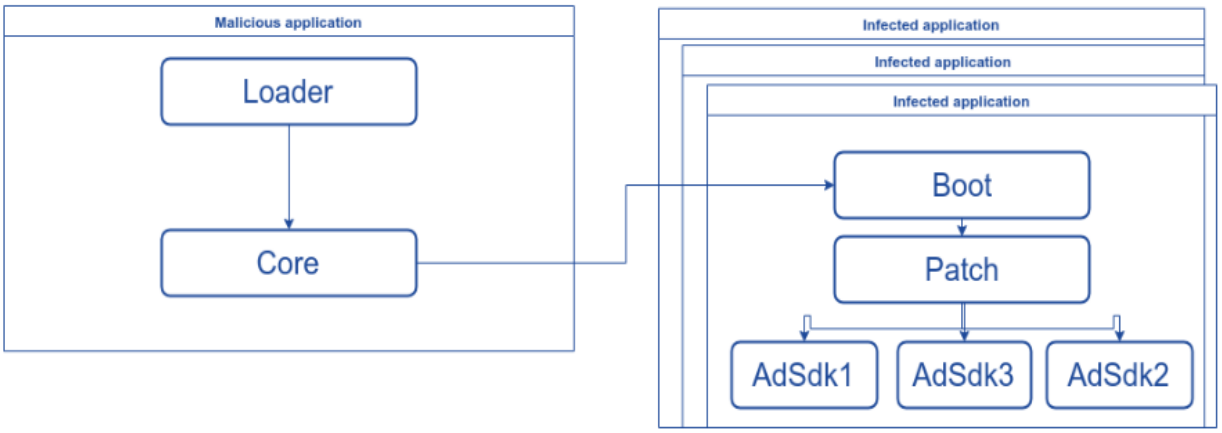
Patch:

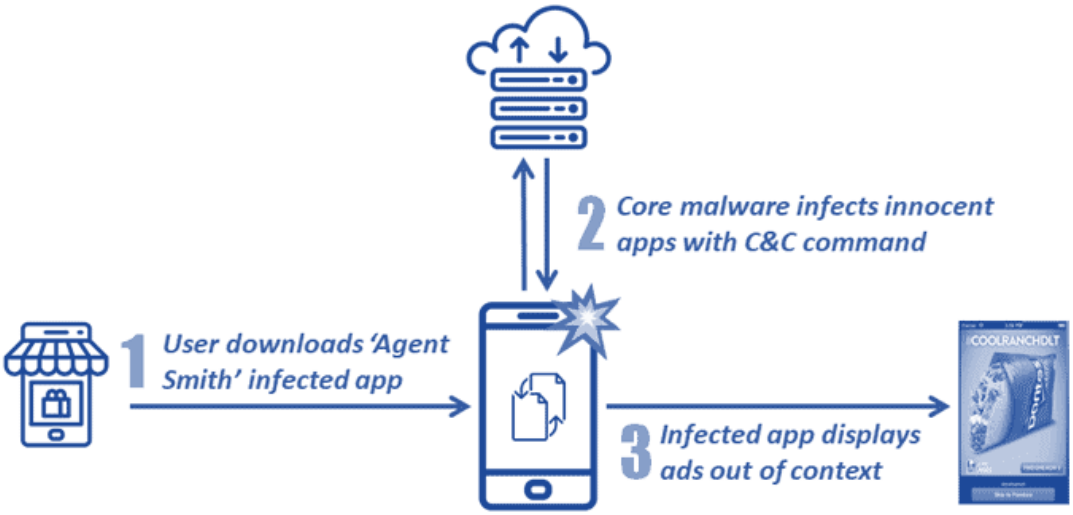
מודול זה עוצב כדי למנוע מאפליקציות להתעדכן עדכון לגיטימי (עדכון פורמלי אמיתי), מכיוון שאם עדכון כלשהו יותקן, אז כל רכיב זדוני שייושם לאפליקציה יוסר.

AdSDK:

מודול הפרסומות. זהו charger אשר מציג לקורבנות פרסומות ומעביר את הכסף עבור הצפייה בהן לתוקפים. מודול זה גם מתקין במכשיר הקורבן גם עוד תוכנת פרסומות זדונית.

לא הייתה שום תזכורת ל-DGA וגם לא שימוש ב-fileless malware בסוכן סמית'.





השימוש בתוכנות מערכת קיימות הוא על ידי השימוש באפליקציה מתוך רשימת האפליקציות שיכולה להיות כבר מותקנת במכשיר הקורבן, האפליקציה הזדונית מזריקה קוד זדוני כדי לקמפל מחדש את האפליקציה הלגיטימית.

אכן התגלה Kill Switch, במקום שיבוץ payload זדוני בפרסומות, התוקף מחליף זאת בגישת SDK.

במודול המסוכן שוכנת לוגיקת kill Switch אשר מחפשת את מלית המפתח "infect". כאשר מילת מפתח זו נמצאת, ה-SDK יחליף משרת פרסומות לגיטימי אל שולח payload זדוני. ולכן קמפיין תקיפה זה נקרא Jaguar Kill Switch. הקוד שמוצג למטה הוא כרגע קוד מופרד ורדום, כאשר בעתיד הוא יעורר על-ידי ה-SDK הזדוני בזמן הצגת הפרסומות. ככל הידוע לנו לעת עתה, אין כל מידע על הפרוטוקול שבו משתמש שרת ה-C&C של הסוכן סמית'.

השירותים העיקריים של הסוכן סמית' הם מבוססי ענן בעוד שהשגת הרשימה המעודכנת היא מושגת על-ידי תקשורת P2P.

ראיות מעידות שהסוכן סמית' הוא כרגע עובד מאחורי הקלעים בפרופיל נמוך ומגביר את החדירה שלו לחנות האפליקציות Google Play ומחכה לזמן הנכון כדי להתחיל בתקיפות.

```
package com.[redacted]network;

import com[redacted]ads.c;

public class d {
    public static final String a = "http://[redacted]:9010";
    public static final String b = "http://sdk.[redacted].com";
    public static final String c = "http://tt[redacted].net:8080";
    public static final String d = "/api/sdk.ad.requestRes";
    public static final String e = "/api/sdk.ad.requestAds";
    public static final String f = "/api/sdk.ad.uploadResult";
    public static final String g = "/api/sdk.ad.uploadAlphaData";

    public d() {
        super();
    }

    public static String a(String arg3) {
        String v0;
        if(c.a()) {
            v0 = "http://[redacted]:9010" + arg3;
        }
        else {
            StringBuilder v1 = new StringBuilder();
            v0 = "infect".equals("") ? "http://tt.[redacted].net:8080" : "http://sdk.[redacted].com";
            v0 = v1.append(v0).append(arg3).toString();
        }

        return v0;
    }
}
```

עד כמה שידוע, אין כל מכניזם כלשהו למניעת מתקפה שכזו, אך מומלץ מאוד למשתמשים לבדוק את המכשיר שלהם ולהסיר כל אפליקציה חשודה שהם מוצאים. בנוסף גם על מפתחי תוכנה ומערכות לזכור ליצור patches בכדי למנוע מהאפליקציות שהם מפתחים להינזק בידי מתקפות שכאלו ולתקן אותם, אצל המשתמשים שהורידו את האפליקציות שלהם. ישנה המלצה להוריד אנטי-וירוס למכשיר שיזהה malwares שכאלו. בנוסף, חנות האפליקציות Google Play מסירה אפליקציות חשודות.

ה-payload שמציג את הפרסומות:

ב-payload שהוזרק, המודול ממש את המתודה "callActivityOnCreate".

בכל זמן נתון, אפליקציה "שהורעלה" תיצור פעילות כלשהו, המתודה הנ"ל תיקרא והיא תקרא ל-"requestAd" מהקוד של הסוכן סמית'. לאחר מכן, הסוכן סמית' יחליף את הפעילות המקורית של האפליקציה בפעילות SDK אשר תראה באנר של פרסומת שנשלח מהשרת שמכיל את כל הפרסומות.

במקרה ואפליקציה שהורעלה אינה מצוינת ברשימת האפליקציות שהוזכרה לעיל, הסוכן סמית' פשוט יציג את הפרסומת על הפעילות שנטענה.

להלן האינטגרציה של פרסומת ה-SDK:

```
Method v2 = Class.forName("android.app.ActivityThread").getDeclaredMethod("currentActivityThread");
v2.setAccessible(true);
Object v1 = v2.invoke(null);
Field v4 = v1.getClass().getDeclaredField("mInstrumentation");
v4.setAccessible(true);
v4.set(v1, new InstrumentationProxy(v4.get(v1)));
```

להלן הקוד שמחליף את הפעילות הלגיטימית של האפליקציה בפעילות זדונית של ה-SDK:

```
if("com.mxtech.videoplayer.ad".equals(v1)) {
    HookManager.BorrowOtherActivity("com.google.android.gms.ads.AdActivity");
}
else if("com.lenovo.anyshare.gps".equals(v1)) {
    v2 = 10000;
    HookManager.BorrowOtherActivity("com.google.android.gms.ads.AdActivity");
}
else if("com.whatsapp".equals(v1)) {
    HookManager.BorrowOtherActivity("com.whatsapp.voipcalling.VoipActivityV2");
}
```

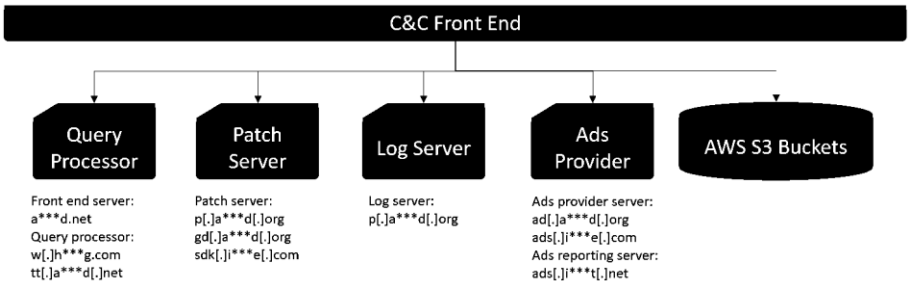
להלן הקוד שגורם ל-malware להראות פרסומות בכל פעילות שנטענת:

```
class ShowAdRunnable implements Runnable {
    ShowAdRunnable(StartAdBusiness arg1, com.hplaceads.business.StartAdBusiness$1 arg2) {
        this(arg1);
    }

    private ShowAdRunnable(StartAdBusiness arg1) {
        StartAdBusiness.this = arg1;
        super();
    }

    public void run() {
        try {
            AliUtil.sendAnalyticsCalculate("Cal_AdsSDKLisen", "StartAdBusiness", "Show Interstitial");
            AdService.showAd(new AdsConfig("interest"));
        }
        catch(Throwable v0) {
            AliUtil.sendAnalyticsError("error", v0);
        }
    }
}
```

אמנם העובדים מחברת checkpoint שחקרו את הסוכן סמית' לא איתרו את שרתי הפיקוד ולכן לא השיגו כל פירוט על התשתיות שלהן רק ידוע שהסוכן סמית' פותח באזור שטח סין, אך הם כן גילו משהו חשוב בנוגע לשרת ה-C&C.

מודול הליבה מתקשר לשרת ה-C&C ומנסה להשיג רשימה מעודכנת של האפליקציות אשר אותן הוא צריך לחפש. אם פעולה זו נכשלת, אזי – הוא משתמש ברשימת האפליקציות ברירת-המחדל שלהלן:

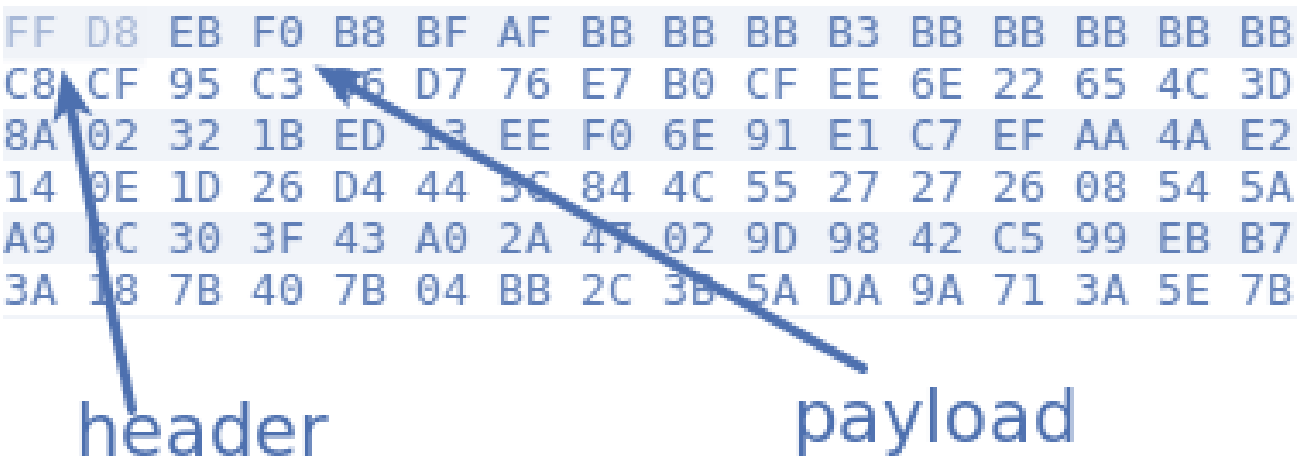
- whatsapp
- lenovo.anyshare.gps
- mxtech.videoplayer.ad
- jio.jioplay.tv
- jio.media.jiobeats
- jiochat.jiochatapp
- jio.join
- good.gamecollection
- opera.mini.native
- startv.hotstar
- meitu.beautyplusme
- domobile.applock
- touchtype.swiftkey
- flipkart.android
- cn.xender
- eterno
- truecaller



טכניקת האפלה:

נזכיר שמודול ה-loader מחלץ ומפעיל את מודול הליבה.

כאשר מודול הליבה נמצא בתוך קובץ ה-APK, הוא מוצפן ומוסווה אל קובץ תמונה JPG, כאשר שני הבתים הראשונים הם בעצם ה-magic header של קבצי ה-JPG, בעוד ששאר ה-data מוצפן עם מצפין מסוג XOR.



פירוט כללי וטכני ביחס לקבוצה מאחורי הנוזקה

לאחר חקירה לעומק, התברר כי השימוש הראשון של נוזקה זו כנראה היה בפיתוח על-ידי חברה הממוקמת בסין. ההפצה העיקרית של התקיפה הייתה על-ידי משתמשים שהורידו אפליקציות תמימות לכאורה מה-play store כמו אפליקציה 9apps שמאוד פופולארית באזור אסיה.
ל-agent smith אין מכניזם הפצה פורמאלי שמדביק את המשתמשים. מתקפה זו לא כוללת דירוג בחנות האפליקציות של ה-play store.

אין כל מידע על הקבוצה עצמה שפיתחה את Agent Smith ולכן אין לנו אינפורמציה על עוד נוזקות שפותחו מגורם זה.

אמנם הסוכן סמית' היא נוזקה שפועלת באופן ייחודי לה, אך עם זאת די סטנדרטי בחלק מפעולותיה. בהסתכלות מהמאמרים עליה, לא נמצא עוד נוזקות המשתמשות במרכיבים טכניים שבהם הסוכן סמית' משתמש.

חלק מהשיטות שפועלות באופן סטנדרטי שהוזכרו לעיל, עובדות כמו נוזקות רבות אחרות. למשל – הטכניקה שבה הפרסומת הזדונית מציגה את עצמה לפרסומת עבור אפליקציה לגיטימית היא די נפוצה בקרב ה-malwares הרבים שנמצאים בחנויות אפליקציות צד שלישי.

סוף סיפור

אמנם התוקף שמאחורי הסוכן סמית' הרוויח כסף בצורה לא חוקית על-ידי נוזקה זאת, תוקף אחר יכול לעשות שימוש בה בצורה הרבה יותר קשה ומסוכנת לקורבנות. ישנן אינסוף דרכים שבהם תוקפים מסוג כזה יכולים לפגוע באבטחת מכשירי הקורבן למשל גניבת מידע ו/או זהות.

כרגע, אין מידע על כמה עלה פיתוח הסוכן סמית'.

ניתן לראות כי פיתוח הנוזקה היה בידי מגוון רחב של מפתחים שמבינים בעניין תהליך התקנת אפליקציות במכשירים השונים ובנוסף יודעים על החולשות שבתהליכים אלו. לא ידוע האם הנוזקה פותחה בידי מספר קטן או גדול של ארגונים שבהן יש מפתחים בעלי ניסיון.

מכיוון שידוע שהסוכן סמית' משתמש בחולשות החתימה של קבצי ההתקנה ה-APK, מומלץ למפתחים לממש בקבצי ההתקנה שהם יוצרים, את סכמת החתימה העדכנית ביותר הנקראת V2 בקובץ ההתקנה שלהם, בכדי למנוע מאפליקציות זדוניות לנצל את חולשת האנדרואיד Janus שצוינה לעיל.

לכן יש להיזהר מכל אפליקציה הנראית לנו במעט חשובה ואף לדווח עליה במקום שבו ניתן להוריד אותה.

רשימת מאמרים

https://research.checkpoint.com/agent-smith-a-new-species-of-mobile-malware/

https://blog.checkpoint.com/2017/07/06/how-the-copycat-malware-infected-android-devices-around-the-world/

https://blog.checkpoint.com/2016/07/01/from-hummingbad-to-worse-new-in-depth-details-and-analysis-of-the-hummingbad-andriod-malware-campaign/

https://blog.checkpoint.com/2016/11/30/1-million-google-accounts-breached-gooligan/

https://thehackernews.com/2018/08/man-in-the-disk-android-hack.html

https://thehackernews.com/2017/12/android-malware-signature.html

https://www.thehindu.com/sci-tech/technology/beware-of-agent-smith-an-android-phone-malware/article28759485.ece

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-13156

https://blog.trendmicro.com/trendlabs-security-intelligence/janus-android-app-signature-bypass-allows-attackers-modify-legitimate-apps/?_ga=2.39349283.1649811272.1570288589-1745146892.1570037896

https://searchsecurity.techtarget.com/definition/man-in-the-disk-MITD-attack

https://www.youtube.com/watch?v=BmAnm1QEq1k

https://www.bleepingcomputer.com/news/security/25-million-android-devices-infected-by-agent-smith-malware/

https://mobile.twitter.com/virqdroid/status/1149312490968440832

https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/AndroidOS_InfectionAds.HRXA

https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/agent-smith-malware-infecting-android-apps-devices-for-adware

https://www.zdnet.com/video/new-android-malware-replaces-legitimate-apps-with-ad-infested-doppelgangers/

https://securityaffairs.co/wordpress/88272/hacking/agent-smith-android-malware.html