AVIS D'EXPERT
MARS 2015

IBANKING :

QUAND LES MALWARES CONTOURNENT L'OTP DE FACEBOOK

Par **François Normand**, expert en cybersécurité chez LEXSI

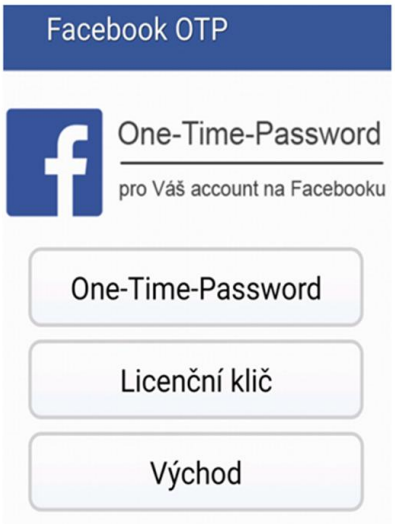
Introduction

Ibanking est un malware bancaire pour la plateforme Android. Il dispose de fonctionnalités classiques comme l'interception / le transfert d'appel et de SMS, le vol d'informations sur l'équipement, et bien d'autres.

Ce malware a impacté de nombreuses banques à l'étranger en 2014. La version analysée pour cette publication est dédiée à la récupération de mots de passe à usage unique (OTP) afin de contourner la double authentification de Facebook [1].

Le condensat SHA256 de la souche étudiée est : 526ad1aada48fb5ed434443503d05e1ff86dbbbedd9d2bab5d568e502c11cb80.

Ce malware se fait passer pour un générateur d'OTP pour Facebook comme le montre les impressions d'écran ci-dessous.



Anti-émulateur

Lors du lancement de l'application dans un émulateur Android, l'application se termine sans lancement d'activité ni de message d'erreur.
Des constantes propres à l'émulateur Android et au «Bouncer » [2] de Google sont présentes dans le code afin de complexifier l'analyse dynamique.

```
if((this.getResources().getString(0x7F070003).equals("1")) && ((v0_1.equals("000000000000000"))
        || (this.c().startsWith("1555521")) || (this.d().equals("Android")) || (this.e().equals(
       "89014103211118510720")))) {
    Process.killProcess(Process.myPid());
}
```

La signification des constantes sont les suivantes :

- « 00000000000000000 » : l'identifiant de l'équipement par défaut pour l'émulateur
- « 1555521 » : les premiers caractères du numéro de téléphone pour l'émulateur et le Bouncer
- « Android » : l'opérateur téléphonique par défaut pour l'émulateur
- « 89014103211118510720 » : le numéro de série de l'équipement par défaut pour l'émulateur et le Bouncer

Cette technique a été détectée pour la première fois dans le malware Android OBAD.

Déchiffrement des configurations

Le malware analysé dispose de deux configurations, l'une globale pour le malware et l'autre pour les cibles.

Déchiffrement de la configuration du malware
Pour la configuration du malware, un algorithme de chiffrement développé par Adobe a été utilisé. Celui-ci est à l'origine utilisé pour chiffrer les portions de code des programmes de police Type 1 [3].

```
public static String a(String arg10) {
    String v3 = cD.b(arg10);
    long v0 = Long.parseLong(AService.a.getString(0x7F07000E));
    StringBuffer v4 = new StringBuffer("");
    int v2;
    for(v2 = 0; v2 < v3.length(); ++v2) {
        short v5 = ((short)v3.charAt(v2));
        v4.append(((char)(((int)((((long)v5)) ^ v0 >> 8)))));
        v0 = ((v0 + (((long)v5))) * 52845 + 22719) % 65536;
    }

    String v0_1 = v4.toString();
    if(v0_1.startsWith("011")) {
        v0_1 = "+" + v0_1.substring(3);
    }

    return v0_1;
}
```

L'implémentation de l'algorithme de déchiffrement permet de récupérer les données techniques inhérentes au malware.

Ces résultats mettent en avant :

- une liste de C&C : (robrato.net / guniches.net / utosedi.net / izbura.net / aftyshev.com / opleton.net /echenry.net / eznasne.net / robrato.net / rylnymi.com)
- une partie de l'arborescence du C&C :
  - /iBanking/sms/index.php
  - /iBanking/sms/saveSMS.php
  - /iBanking/sms/sync.php
  - /iBanking/getList.php
  - /iBanking/sendFile.php
  - /iBanking/sms/ping.php
  - /iBanking/checkUrl.php
- le numéro de téléphone de l'attaquant : 01179858114235
- un botID : 310
- une chaîne de caractères énigmatique : NYY2XXXXXXX602 (extrait de la chaîne originale)

Énigme de la chaîne de caractères
Les autres composants de la configuration ayant déjà fait l'objet d'une analyse, attardons-nous sur cette fameuse chaîne de caractères.
On retrouve dans la fonction « onReceive » de la classe «SMSReciever » une comparaison entre une variable « v4 » et une référence vers la chaîne en question.

```
if(!v4.equals(new String(cD.a(this.a.getString(0x7F07000F))))) {
```

Précédemment, la variable « v4 » a été initialisée avec une chaîne de caractères provenant du corps de message d'un SMS reçu.

```
v4 = v3.getDisplayMessageBody().toString();
```

Dans le cas où la condition n'est pas prise, le malware enregistre dans sa base de données le numéro de téléphone ayant émis le message :

```
new StringBuilder("new_tel: ").append(v5).toString();
ek.a = v5;
new StringBuilder("smsParser.tel1 - ").append(ek.a).toString();
AService.a();  //  fonction d'ajout du numéro dans la base de données
SmsReciever.a(arg13);
this.abortBroadcast();
dS.a();  // fonction d'envoi du SMS de confirmation
```

La dernière fonction « a() » envoie une réponse à l'émetteur lui indiquant que la prise de contrôle du téléphone est effectuée.

```
public static void a() {
    try {
        SmsManager.getDefault().sendTextMessage(ek.a, null, "Control number change to " + ek.a,
                null, null);
```

Le mystère de la chaîne de caractères est maintenant levé puisque celle-ci est en réalité le message permettant d'activer la prise de contrôle à distance du téléphone.

Déchiffrement de la configuration liée aux sites ciblés
Dans cette partie, l'algorithme utilisé est de l'AES 128. Le vecteur d'initialisation, la clé privée et le mode utilisé sont déclarés au niveau du constructeur de la classe.

```
private String a;
private Cipher a;
private IvParameterSpec a;
private SecretKeySpec a;
private String b;

public cD() {
    super();
    this.a = "exs4Ts*D0T69O^MW";
    this.b = "o&2V&TZ3g*6U9hIq";
    this.a = new IvParameterSpec(this.a.getBytes());
    this.a = new SecretKeySpec(this.b.getBytes(), "AES");
    try {
        this.a = Cipher.getInstance("AES/CBC/NoPadding");
    }
    catch(NoSuchPaddingException v0) {
        v0.printStackTrace();
    }
    catch(NoSuchAlgorithmException v0_1) {
        v0_1.printStackTrace();
    }
```

Voici le résultat de la configuration après son déchiffrement :

```
<string name="template44_head">Facebook OTP</string>
<string name="template44_one_time_password">One-Time-Password</string>
<string name="template44_exit">Východ</string>
<string name="template44_license_code">Licenční klíč</string>
<string name="template44_about_programs">O programu</string>
<string name="template44_generate">Generovat</string>
```

À la lecture de celle-ci, on constate que la communauté tchèque de Facebook est ciblée par cette souche.
Le contenu de cette configuration correspond aux chaines de caractères qui seront utilisées pour construire « l'Activité » de l'application malveillante, dont le résultat graphique est dans l'introduction.

Conclusion

La technique employée par les auteurs de ce malware n'est pas nouvelle. Cependant, une méthode à la base utilisée pour le secteur bancaire est désormais exploitée afin d'accéder à des données personnelles sur le réseau social Facebook et potentiellement d'autres dans les souches à venir.

[1] https://www.facebook.com/notes/facebook-engineering/introducing-login-approvals/10150172618258920
[2] http://googlemobile.blogspot.fr/2012/02/android-and-security.html
[3] https://partners.adobe.com/public/developer/en/font/T1_SPEC.PDF

À propos de LEXSI :

15 ans d'histoire ont fait de LEXSI la première société de service spécialisée en cybersécurité en France. Avec le plus important CERT indépendant d'Europe et une base de threat intelligence faisant référence sur le marché, LEXSI veille sur la sécurité de plus de 500 clients dans le monde.

Le portfolio de LEXSI compte des services d'audit, de conseil, d'incident response et de formation, ainsi qu'une offre logicielle aboutie, essentiellement disponible en SaaS. Ces produits s'articulent naturellement autour d'une démarche de défense structurée faite d'anticipation et de riposte.

LEXSI en quelques chiffres :

- 22M€ de chiffre d'affaire
- Croissance à 2 chiffres depuis plus de 5 ans
- 200+ experts
- 4 pôles : audit, conseil, formation, CERT
- 1er CERT privé d'Europe
- 15 ans d'expérience

Contacts presse :

OXYGEN
Claire du Boislouveau
01 41 11 37 85
claireb@oxygen-rp.com

LEXSI
Anne Bigel
01 73 30 17 08
abigel@lexsi.com

***