

Rapport de projet long
Attaques sur un AR Drone 2.0 Parrot

Kevyn LEDIEU, Ferréol PENNEL, Alexis PERNOT

27 février 2019

Table des matières

1	Introduction	2
2	Matériel et Méthode	3
	2.1 Matériel : AR DRONE 2.0	3
	2.2 Méthode	3
3	Hypothèse de travail	4
4	Présentation vulgarisée du travail	5
	4.1 Premiers pas avec le drone	5
	4.2 Prise de contrôle du drone	5
	4.3 Injection de commandes	5
	4.4 Virus sur le drone	5
	4.5 Conclusion	5
5	Présentation scientifique du travail	7
	5.1 Introduction	7
	5.2 Prise de contrôle du drone par désauthentification	7
	5.3 Injection de paquets	7
	5.4 Exploitation d'une connexion Telnet	7
6	Tutoriel	8
	6.1 Initialisation de l'application	8
	6.2 Menu principal	8
	6.3 Prise de contrôle du drone	9
	6.4 Injection de commandes	10
	6.5 Dépose de virus	11
7	Conclusion	12

1 Introduction

De plus en plus présents autour de nous, les drones de loisirs représentent à la fois des opportunités technologiques et des risques de sécurité. Régulièrement, nous pouvons observer des exemples de drones ayant perturbé le trafic aérien par leur présence aux abords d'un aéroport. Ainsi un premier risque de sécurité qu'ils représentent est leur intégration dans l'espace aérien, espace que ces nouveaux aéronefs partagent avec de nombreux autres de toutes les tailles. Aussi des nouvelles règles sont à l'étude afin de réglementer ces activités de loisirs. Toutefois, même une fois réglementée et contrôlée, l'activité des drones de loisir présente un second risque de sécurité lié non plus à la gestion du drone par l'opérateur mais au drone lui-même. En effet, dans la course à l'innovation dans ce domaine porteur qu'est le drone de loisir, les entreprises négligent potentiellement l'aspect sécurité hardware et logicielle de leurs drones. Aussi, ceux-ci peuvent présenter de nombreuses vulnérabilités permettant à un attaquant extérieur de potentiellement prendre le contrôle du drone. Dans ce contexte, nous avons décidé d'étudier un drone grand public proposé par un des leaders du marché, la société *Parrot*. Nous étudierons les différentes vulnérabilités potentiellement présentes sur ce drone et mettront en oeuvre un scénario d'attaque sous forme de tutoriel.

2 Matériel et Méthode

2.1 Matériel : AR Drone 2.0

Nous allons étudier un AR DRONE 2.0. C'est un hélicoptère quadrirotor pilotable via une liaison WiFi au travers d'une application disponible sous iOS, Android, Linux ou Windows. C'est un drone civil principalement destiné au divertissement. Il est équipé de :

- un *processeur ARM Cortex A8 32 bits cadencé à 1 GHz*
- *1 Go de RAM DDR2 cadencée à 200 MHz*
- un *système d'exploitation Linux 2.6.32*
- un *module WiFi b/g/n*
- un *accéléromètre 3 axes*
- un *gyroscope 3 axes*
- un *capteur de pression*
- un *magnétomètre 3 axes*
- des *capteurs de proximité à ultrasons*
- une *caméra vericale QVGA*
- un *port USB 2.0*

2.2 Méthode

3 Hypothèse de travail

Pour ce projet, nous supposons que le drone utilisé est un ARDrone 2.0 qui n'a pas subi de modifications logicielles et qui est donc dans un état identique à sa sortie d'usine. Il dispose ainsi uniquement des protections éventuelles prévues par le constructeur Parrot. Nous supposons dans le cadre de ce projet que nous sommes dans la position de l'attaquant et que l'ARDrone 2.0 est connecté et contrôlé par un client légitime.

4 Présentation vulgarisée du travail

4.1 Premiers pas avec le drone

Ce projet est centré sur la découverte de vulnérabilités sur l'ARDrone 2.0. Une première mise sous tension du drone nous a permis de découvrir que celui-ci propose un réseau Wifi ouvert, donc non protégé, permettant de se connecter à celui-ci et de le contrôler. Cette observation a grandement orienté notre travail. En effet, nous nous attendions à trouver un réseau Wifi sécurisé qu'il aurait été nécessaire de contourner ou de pénétrer afin d'avoir accès au drone. Cependant, le réseau ouvert nous permet en tant qu'attaquant de nous connecter directement au réseau Wifi créé par le drone. L'ARDrone 2.0 supporte la connexion de plusieurs clients simultanés à son réseau Wifi, toutefois seul le premier client connecté au réseau est maître du drone et a la possibilité de le contrôler. En supposant qu'un client légitime est connecté au drone, la question est : que peut faire l'attaquant ?

4.2 Prise de contrôle du drone

Une première idée est la prise de contrôle du drone par l'attaquant. Une fois connecté au réseau du drone, l'attaquant n'a qu'à déconnecter l'utilisateur légitime pour devenir le maître du drone. Ainsi en envoyant des messages au drone demandant la déconnexion du client légitime, l'attaquant est capable de reprendre le contrôle du drone une fois celui-ci déconnecté.

4.3 Injection de commandes

Dans ce cas, le client légitime reste maître du drone. Toutefois l'attaquant se fait passer pour le client légitime et envoie des messages au drone. Il peut ainsi lui envoyer des instructions que le client légitime n'a jamais envoyées. Par exemple, l'attaquant peut envoyer au drone l'instruction d'atterrir à la place du client légitime.

4.4 Virus sur le drone

Ce troisième point exploite une faille importante du drone. En effet, celui-ci offre à tout utilisateur connecté à son réseau Wifi la possibilité d'accéder directement au système d'exploitation du drone en ayant tous les droits sur celui-ci et sans authentification et protection. Ainsi, en tant qu'attaquant connecté au réseau Wifi du drone, nous utilisons cet accès au drone pour déposer sur celui-ci un script. Ce script est chargé de copier une image sur toute clé USB connectée au drone ceci afin de démontrer le potentiel de l'attaque. Il est en effet aisé de diffuser n'importe quel virus qui se diffuse par clé USB grâce au drone et ceci sans difficultés particulières.

4.5 Conclusion

Ces trois attaques différentes permettent de démontrer les vulnérabilités majeures que présente l'ARDrone 2.0. Une des failles principales de celui-ci est son réseau Wifi non protégé. La protection de celui-ci par du WPA2 permettrait de rendre ces trois différentes attaques beaucoup plus complexes car l'attaquant aurait dans un premier temps besoin de casser le réseau Wifi. De plus, le drone offre des accès non protégés et privilégiés qui

sont une faille majeure et permettent à un attaquant une prise de contrôle complète sur le drone.

5 Présentation scientifique du travail

5.1 Introduction

5.2 Prise de contrôle du drone par désauthentification

5.3 Injection de paquets

5.4 Exploitation d'une connexion Telnet

6 Tutoriel

Le tutoriel du projet se présente sous la forme d’une application Linux développée en Python permettant d’exploiter les trois types d’attaques présentées précédemment. Disponible à l’adresse suivante : <https://github.com/ferreolpennel/Krok mou>, cette application de démonstration est utilisable par tout utilisateur satisfaisant les pré-requis à son installation. L’application, appelée KROKMOU, est dédiée à l’ARDrone 2.0 et ne permet des attaques que contre ce type de drone et ce à des fins de démonstration uniquement. Elle permet ainsi de prendre le contrôle d’un drone à la place d’un utilisateur légitime déjà connecté au drone, d’envoyer des commandes pirates au drone sans déconnecter l’utilisateur légitime de celui-ci et de déposer un virus de démonstration sur le drone. Elles illustre ainsi les attaques présentées précédemment et met en relief les failles correspondantes sur ce type de drone.

6.1 Initialisation de l’application

L’application permet de sélectionner l’interface Wifi à utiliser pour se connecter au drone. Elle réalise ensuite un scan des réseaux Wifi alentours et affiche ensuite uniquement les réseaux Wifi de drone Parrot. Il suffit à l’attaquant de sélectionner le drone auquel il veut se connecter puis l’application configure l’interface Wifi pour se connecter au drone.



FIGURE 1 – Initialisation de l’application

6.2 Menu principal

Le menu principal de l’application permet de sélectionner une des trois attaques afin de la réaliser sur le drone auquel l’attaquant s’est connecté précédemment.



FIGURE 2 – Menu principal de l'application

6.3 Prise de contrôle du drone

Cette option du menu permet à l'attaquant de prendre le contrôle du drone à la place de l'utilisateur légitime grâce à l'attaque décrite précédemment dans ce rapport. Le contrôle du drone se réalise au travers du navigateur Web et d'un serveur **Node.js** issu d'un dépôt Github (<https://github.com/functino/drone-browser>).

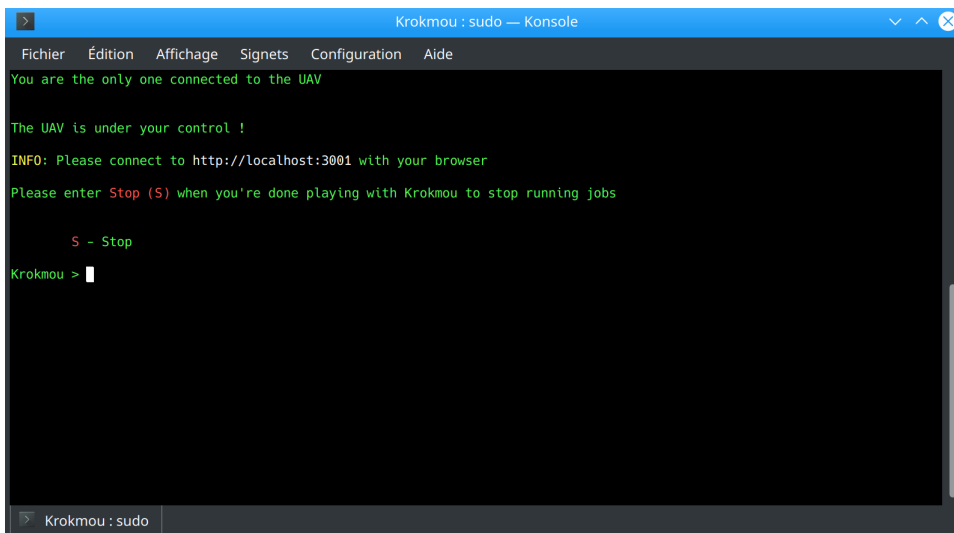


FIGURE 3 – Prise de contrôle du drone par l'application

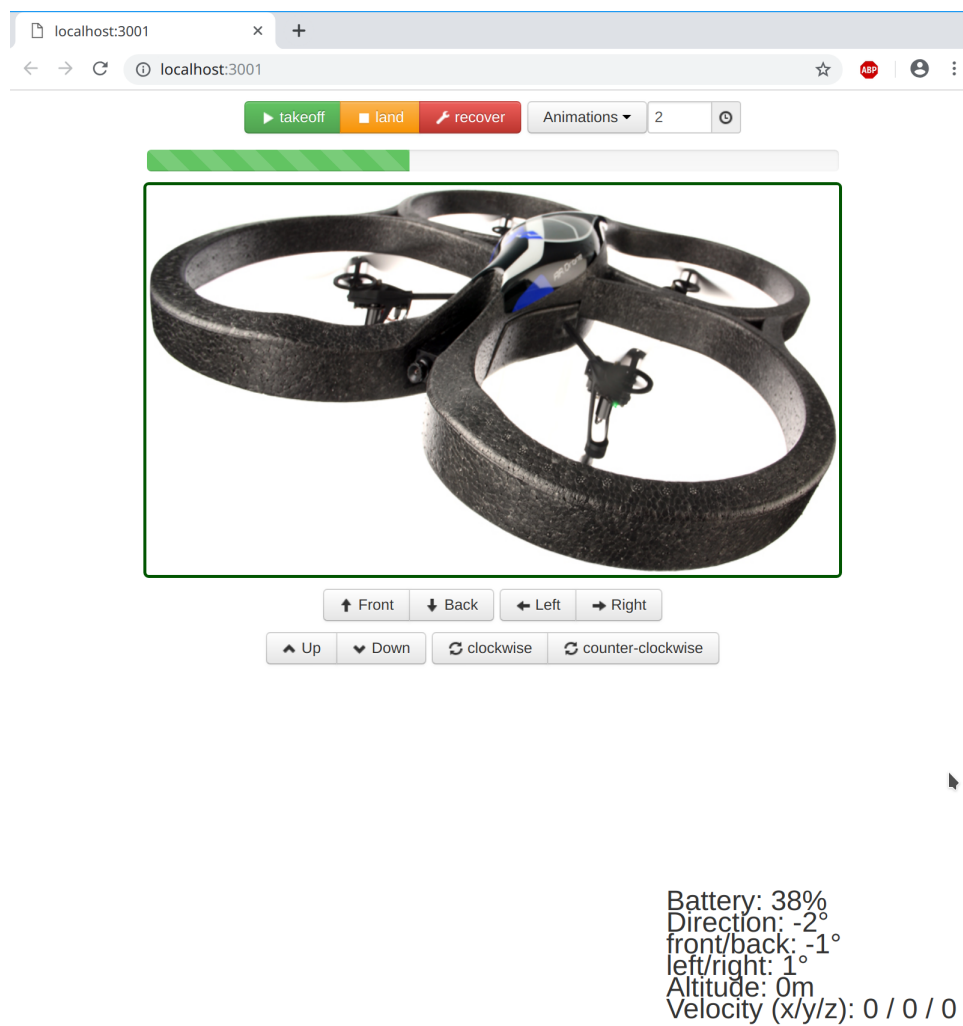


FIGURE 4 – Interface de contrôle web

6.4 Injection de commandes

Ce sous-menu permet à l'attaquant d'envoyer des instructions au drone sans déconnecter le client légitime de celui-ci à travers une injection de paquets.

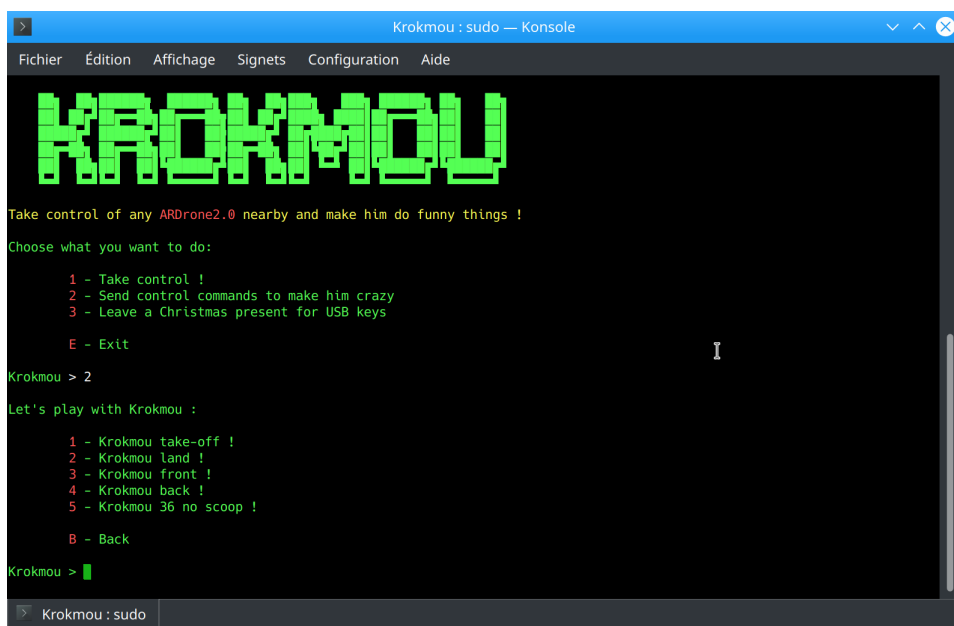


FIGURE 5 – Sous-menu d'injection de commandes

6.5 Dépose de virus

Cette option permet d'exploiter une vulnérabilité laissant à l'attaquant un contrôle total du drone. Dans le cadre de la démonstration, l'application dépose un virus et un fichier sélectionné par l'utilisateur sur le drone. Ce fichier sera copié sur toute clé USB qui sera connectée au drone qui sera par conséquent considérée comme infectée. Par défaut, l'application dépose le virus et une image sur le drone. C'est cette image qui sera copiée sur toute clé USB connectée au drone toujours à des fins de démonstration. Cependant, l'attaquant peut indiquer le chemin d'un fichier de son choix au moment où l'application lui propose afin de remplacer cette image.

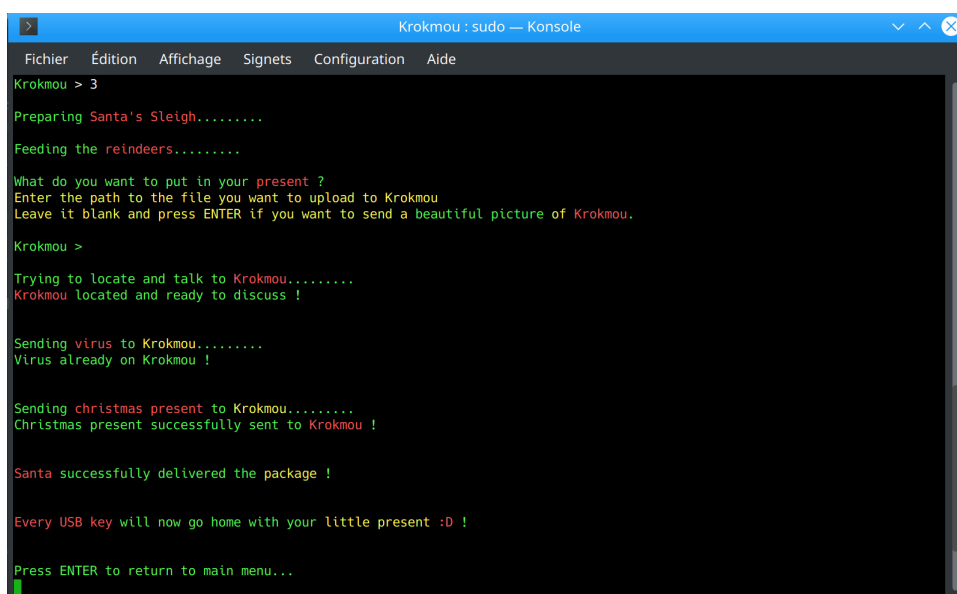


FIGURE 6 – Envoi du virus et du fichier sur le drone

7 Conclusion