Étude du Système de Communication de l’AR.Drone 2.0 Quad-copter

Abstract

Les drones ont commencé pour etre utilisés au profit des forces armées ou de sécurité d’un État, mais de plus en plus ont aussi des applications civiles comme la recherche, le cinema, et l'environnement.

C’est à cause de ces nouveaux marches qu'on a commencé à commercialiser les drones pour cibler différents activités…

Dans cet article on fait l’étude du système de communication de l’AR.Drone 2.0 Quad-Copter auquel suive l’étude de faiblesses dans le système et la démonstration de possibles scénarios d’attaque.

1 INTRODUCTION

L’AR.Drone est un drone Quad-Copter commandé à distance grâce à l’application AR.FreeFlight disponible pour les apparails mobiles de plateforme iOS et Android. La connexion c'est faite sur un réseau Wi-Fi non-sécurisé monté par le drone après son l’allumage, auquel tout appareil peut se connecter et envoyer des instructions et recevoir la télémetrie et le flux vidéo.

Même si l’AR.Drone 2.0 n’est pas un drone professionnel, ses caractéristiques physiques et facilité de control, grâce à l’application AR.FreeFlight font de lui un outil puissante. L’AR.DRONE offre aussi la possibilité, en utilisant son propre SDK, de développer des applications diverses ciblant la recherche, la sécurité, les jeux, la cinématographie, etc.

2 DÉVELOPPEMENT

2.1 Système de Communication

Une fois allumé, l'AR.Drone 2.0 mis en place un point d'access non-securisé appelé ardrone2\_\* ou \* est une séquence de números apparentement aléatoires. 090933. Déjà connecté, on est capable de faire un scan de ports ciblant l'adresse IP du drone (192.168.1.1), en utilisant l'application NMAP. Le résultats de cet scan nous montre que on a par defaut quelques ports ouverts :

C'est très important de remarquer que les services FTP et TELNET ne sont pas securisés (protegés par mot de passe) c'est suffi de se connecter à le réseaux proposé par le drone et initialiser ces services vers les ports (par defaut) de FTP et TELNET.

2.2 Système Interne

On peut voir que

2.3 Scenarios d’attaque

2.3.1 Reset du drone

Le processus programe.elf contrôle est le chargé de commander le drone. Si on termine ce processus tout le système de control se reinitialise et crash le drone.

On peut donner un scenario d'attaque très basique dans lequel on termine succesivement le processus controleur du drone.

A continuation on detail les pas a suivre pour effectuer ce type d'attaque

Faire un script qui cherche le PID du processus « program.elf » et le termine cycliquement.

Se connecter à le reseau non-securisé proposé par l'AR.Drone

Transferer ce script au drone (via FTP), lui donner des permises d'execution et lancer le script (via TELNET)

2.3.2 Déconnection du client

Un attaque un peu plus complexe sera de « voler » le control du AR.Drone au client principal.

2.4 Les normatives légales ciblant les drones en France

3 CONCLUSIONS/COMMENTAIRES

4 RÉFÉRENCES