ECOLE SUPERIEURE
D'INGENIEURS DE RENNES

Rendu CC SRIO

Noémie LE DET

6 janvier 2023

J'atteste que ce travail est original, qu'il indique de façon appropriée tous les emprunts, et qu'il fait référence de façon appropriée à chaque source utilisée.

1 Introduction

1.1 D'après vous, Debian est-il mieux sécurisé qu'un système Windows ?

Debian est mieux sécurisé que Windows de par le fait des mise à jour plus fréquente qui permettent une meilleure maintenance du système. De plus, c'est un système moins courant et donc moins susceptible d'être victime d'attaque.

1.2 D'après votre compréhension du business de SRIOEYE, quelle sont les deux principes de sécurité que vous jugez plus délicat ?

La **confidentialité** est un principe de sécurité délicat dans ce cas, car il faut sécuriser l'accès aux données pour que des personnes non autorisées n'y aient pas accès. De plus, le principe de **disponibilité** est également délicat de par le fait que les vidéos des caméras de surveillance doivent être disponible à tout moment. Le principe d'intégrité est moins délicat que les autres dans ce cas.

1.3 Si vous devez résumer en un mot la raison pour laquelle les caméras de SRIOEYE sont vulnérables, quel mot utiliserez-vous ?

Les caméras sont disponibles à cause de l'**"interconnexion"** des différents éléments.

1.4 Pour un équipement IOT comme une caméra connectée à faible puissance, quelles sont les conséquences sur les protocoles de sécurité ?

Si une caméra a une faible puissance, nous devons alors adapter le protocole de sécurité pour le rendre le moins énergivore possible. Ainsi, certains protocoles de sécurité ne pourront pas être utilisés.

1.5 Quelle peut être l'avantage d'un protocole comme LORA pour SRIOEYE ?

Le protocole LORA permet une transmission de données sur de longues distances et est peu énergivore. Ainsi, les caméras ayant une faible puissance peuvent l'utiliser. De plus, le chiffrement de bout en bout permet une protection efficace des données transmises.

2 Obtention de secret d'architecture

Grâce à l'ex copine d'un responsable du SI de SRIOEYE, vous arrivez à ruser et à obtenir des informations précieuses concernant les dossiers sensibles et l'architecture de certaines solutions.

2.1 Comment appelle-t-on cette stratégie façon d'extirper les informations sur une cible ?

Cette stratégie consiste à utiliser l'ingénierie sociale. Cette pratique consiste à user de manipulation pour obtenir des informations auxquelles nous ne devrions pas avoir accès.

3 Gestion de la base de donnée

À travers les informations collectées, vous déterminez que les caméras utilise une version modifiée du protocole WEP-512 (WEP-844) afin de pouvoir échanger avec la base de donnée.

3.1 Sachant qu'on sait que l'algorithme RC4 est utilisé, quelle est la longueur de la clé utilisée par les caméras ?

La clef a une longueur de 40 bits.

3.2 Si les caméras doivent changer des clés, chaque une minute et garder les anciennes clés de la journée pour un souci de reporting, quelle est la taille en MBytes que consommera le stockage des clés.

Il y a 1440 minutes dans une journée : 60 * 24. La clef est stockée sur 40 bits = 5 octets. Ainsi, il y aura 5* 1440 = 7200 octets. Ce qui fait 0,0072 MBytes.

3.3 Quelle est le nombre combinaisons pour une attaque de type force brute si on veut deviner la clé utilisée par les caméras ?

La clef est chiffrée en hexadécimale et il y a 10 caractères hexadécimaux sur cette clef donc : $16^{10} = 1.1^{12}$

3.4 Avec un CPU cadencé à 2.8 GHz par seconde, pourrais-je réussir à obtenir la clé utilisée par une caméra en moins d'une minute ? Sinon, quelle puissance de calcul me faudrait-il ?

Il faudrait réaliser 1.1^{12} calcul par secondes. Or le CPU ne peut en réaliser que 2.8^9 par seconde.

3.5 En hexadécimal, réaliser la conversion des chiffres ci-dessous (en expliquant votre cheminement) : (1044), (481), (844), (24)

$\frac{1044}{16} = 65$ avec reste 4 $\frac{65}{16} = 4$ avec reste 1 $\frac{4}{16} = 0$ avec reste 4 On a donc $1044 = 414$ en base hexadécimale

$\frac{481}{16} = 30$ avec reste 1 $\frac{30}{16} = 1$ avec reste 14 (E) $\frac{1}{16} = 0$ avec reste 1 On a donc $481 = 1E1$ en hexadécimal

$\frac{844}{16} = 52$ avec reste 12 (C) $\frac{52}{16} = 3$ avec reste 4 $\frac{3}{16} = 0$ avec reste 3 On a donc $844 = 34C$ en hexadécimal

$\frac{24}{16} = 1$ avec reste 8 $\frac{1}{16} = 0$ avec reste 1 On a donc $24 = 18$ en hexadécimal

3.6 Vous avez mis la main sur les bouts de code suivant de monitoring ci-dessous. Analyser et déterminer à quelle faille il s'expose avec un exemple concret. Corrigez-les.

Ces codes présentent un problème de **BufferOverflow**. Pour les corriger, il faut ajouter des **assert** pour empêcher l'écriture.

3.7 Quelles sont les techniques dans la littérature pour contrer le type de problème que présentaient les deux codes précédents.

Pour s'en protéger, il est possible d'utiliser le **kernel Adress Space Layout Randomization**, limiter les zones de mémoire avec les droits d'exécution et d'activer les **canaries**.