Charles MacDuff

Étienne Gélinas-Gagnon

|  |  |  |
| --- | --- | --- |
| **Partie Passive** | | |
| **Épreuve** | **Flags** | **Justification** |
| 1.A Débutant | FLAG-1188694300 | Dans la trace A, au paquet 5, dans la couche HTTP du paquet, après le ce que tu cherche. |
| 1.A Débutant | FLAG-753829651886 | Dans Chalenge A2, paquet # 81 c’est le mot de passe écrit en clearText |
| 1.A Débutant | FLAG-777777777777 | Dans Chalenge A3, follow TCP Stream c’est le mot de passe écrit en clearText |
| 1.A Débutant | FLAG-754300003327188 | Dans Chalenge A3, follow TCP Stream c’est le mot de passe écrit en clearText |
| 1.B Intermédiaire | FLAG123 | Dans le Challenge B1 paquet #4 le mot de passe est hasher. Après dé-hasage on obtient le flag. |
| 1.B Intermédiaire | FLAG-987656789 | Dans le Challenge B2 paquet #3. Dans le protocole http. |
| 1.C Crypto hard | FLAG-2167 | Dans le Challenge C, j’ai écouté la vidéo pour comprendre ce qu’il fallait faire et après j’ai convertis le binaire en ascii et rajouter un 0 au début pour avoir 72 digits. |
| 1.D HTTPS | FLAG-8888332751198565 | Dans Challenge D1, follow TCP Stream on le trouve sous FLAG no.1 |
| 1.D HTTPS | FLAG-99365826666577 | Dans Challenge D2, après l’ouverture du fichier compressedSecret.zip avec le mot de passe TigerSuperPassword. |
| 1.D HTTPS | FLAG-111068952415 | Dans Challenge D3, après avoir appliqué la clé PEM sur la trace on voit apparaitre le paquet 46 qui contient le flag. |
| 1.E Wireless | FLAG-6565465654 | Dans Challenge E, avec le mot de passe trouvé dans l’indice nous avons trouvé le flag en entrant ce mot de passe dans Challenge E |
| 1.F ??? | FLAG-147896325 | Dans Challenge F, en regardant dans le dns on voyait 1 caractère par trace. |
| 1.G Crypto easy | FLAG-339951423718 | En décryptant avec le Caesar Cipher |
| 1.H Code | FLAG246813579 | Dans le manifeste de l’application, à la 4e ligne de code. |
| 1.H Code | FLAG-7412369851 | Dans le fichier original\META-INF\CERT.RSA à la ligne 6, 9 et 18 |
| 1.H Code | FLAG-775839667 | Dans la classe SECRET.java à la ligne 13 |
| 1.I Hashing | FLAG-43245198 | En roulant une script qui générait des Flags et qui les comparais au Hash. |

|  |  |  |
| --- | --- | --- |
| **Partie Active** | | |
| **Épreuve** | **Flags** | **Justification** |
| 1.a | FLAG-25111 | Pinger tout le monde et attendre les réponses. Script a.py |
| 1.b | FLAG-5684211967 | Rouler script python b.py et avec la réponse dans wireshark on a trouvé le flag dans un paquet ICMP. |
| 1.c | FLAG-123452 | Rouler le script c.py, on dit qu’on est 192.168.1.111 et après on initie la connexion avec 192.168.1.254 et il nous envoie le flag dans une requête ICMP. |
| 1.d | FLAG-958712242 | En répondant aux requêtes DHCP Discover et Request, on peut apercevoir dans la requêtes DNS qui suit une requête au site web www.FLAG-958712242.com . L'IP 192.168.1.8 à été utilisé (Choisi de façon aléatoire) dans le Offert. Le flag est visible dans la requête DNS comme étant le site web à consulter -> www.FLAG-958712242.com |
| 1.e | FLAG-2569111 | On a premièrement roulé et enregistrer la trace du script e.py (qui consiste à faire des requêtes ARP). Ensuite, nous avons exécuter le script e2.py (qui consiste à obtenir les réponse ARP) pour obtenir le flag. |
| 1.f | FLAG-664499228877 | Envoie d'une requête ARP à 192.168.1.5 ensuite on répond avec du DNS pour finalement répondre avec une requête TPC et le flag se trouve dans la réponse |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |