Question 3

a) IMAGES

b)

Voici la formule de l’entropie qui donne la complexité en nombre de bits

En prenant comme hypothèse que la longueur maximale du mot de passe est de 8 caractères voici la complexité selon les alphabètes :

|  |  |  |
| --- | --- | --- |
| **Alphabète** | **Formule** | **Entropie (bits)** |
| [a-zA-Z] |  |  |
| [a-zA-Z0-9] |  |  |
| Table ascii |  |  |

\*Il s’agit d’une valeur théorique car il y a des caractères qui sont impossibles de taper sur le clavier. Nous utilisons 256 pour démontrer que plus le nombre est grand plus l’entropie augmente.

c)

En analysant le tableau ci-haut, il est facile de remarquer que plus l’alphabète est diversifié, plus l’entropie maximale augmente.

d)

En regardant les résultats de John the ripper, on s’aperçoit que les mots de passe n’ont pas un dictionnaire élargie, qu’ils sont courts et qu’ils ne sont pas formé de façon aléatoire.

Voici 3 critères qui augmente l’entropie d’un mot de passe :

1. Diversité du dictionnaire.
2. Longueur du mot de passe.
3. Mot de passe aléatoire.

Question 6

1. Donnez la séquence exacte de caractères à entrer. Expliquez brièvement comment votre « hack » fonctionne.

Avec exactement 60 caractères aléatoires dans le champ username l’entré au site est toujours garantie. Il en est de même avec 100, 140 et 180 caractères aléatoires. Notre hack fonctionne en écrasant la valeur du nom d’utilisateur par une valeur arbitraire et en écrasant la valeur du mot de passe par le caractère de fin de ligne. Ainsi lorsque l’on se connecte avec par exemple 60 caractères, le nouveau nom d’utilisateur "root" est remplacé par les 20 derniers caractères entrés et le mot de passe est remplacé par “\0“ qui est ajouté lorsque l’on tente de se connecter. Comme le montre le tableau suivant, les 40 premiers caractères servent à écraser les tableaux user\_name et password.

Voici comment est représenté la mémoire du programme.

|  |  |
| --- | --- |
| Emplacement mémoire | Données |
| [0-19] | user\_name[20] |
| [20-39] | password[20] |
| [40-59] | users[0][0] ("root") |
| [60-79] | users[0][1] ("98765") |
| [80-99] | users[1][0] ("moi") |
| [100-119] | users[1][1] ("allo") |
| [120-139] | users[2][0] ("abc") |
| [140-159] | users[2][1] ("motdepasse") |
| [160-179] | users[3][0] ("") |
| [180-199] | users[3][1] ("") |

1. Que faudrait-il changer dans le programme pour enlever ce problème de sécurité?

Premièrement en mettant un nombre de caractère maximum dans le champ username et password. Il serait aussi possible d’ajouter une variable canarie. Par exemple, en ajoutant une chaine de caractère directement après la déclaration du password et en vérifiant l’intégrité de cette variable avant chaque tentative de connexion.