Travaux Dirigés (partie 3 sur 3)

Le traitement numérique des images

Extrait du référentiel : BTS Systèmes Numériques option A (Informatique et Réseaux)		Niveau(x)
S4. Développement logiciel S4.1. Principes de base	Représentation et codage des informations : bases de calcul (2,10,16), types scalaires, réels, caractères, etc.	4
S7. Réseaux, télécommunications et modes de transmissions S7.2.Concepts fondamentaux des réseaux	Modèle en couches et protocoles de l'Internet : IP, ICMP, ARP, UDP, TCP, etc.	3

Objectifs du TD :

- Configuration et analyse du réseau :
  - protocoles : ICMP et ARP
  - classfull et classless
  - IP privée et publique
  - calcul IP
  - commande : tracert
- Du texte « caché » dans des images (la suite)

Support d'activité :

- Logiciel : EditHexa
- Fichiers : Image.bmp, Image modifiée.bmp et MV_XP_WampServer.vxd
- Internet
- Ce document au format PDF

CONFIGURATION ET ANALYSE DU RÉSEAU

Vous allez utiliser une application codée en PHP. Comme vous l'avez vu au cours du premier semestre une application **PHP** s'exécute **coté serveur**.

Dans un premier temps, vous allez donc vérifier la connectivité avec le serveur puis analyser la configuration du réseau.

Question 1

*Vérifiez à l'aide d'un test d'écho de niveau 3 (protocole ICMP) , la connectivité avec le serveur « **SRV-nano-snir** » dont l'adresse **IP** est **172.16.8.124** ? Indiquez la façon dont vous avez procédé et le résultat obtenu.*

……...

……...

……...

Si le test est concluant, vous poursuivez à la question 2. Dans le cas contraire, vous appelez le professeur !

Question 2

*Donnez l'adresse **IP** et le **masque** de votre machine (en notation **décimale pointée**).*

……...

Question 3

*Écrivez votre réponse à la question 2 en notation « **CIDR** ».*

……...

Question 4

*L'adresse IP de votre machine est-elle **privée** ou **publique** ? Expliquez les deux termes.*

……...

……...

……...

……...

……...

Question 5

*D'après la **RFC 1918**, la configuration réseau de votre machine (réponse à la question 3) est-elle en « **classfull** » ou en « **classless** » ? Expliquez.*

……..

……..

……..

……..

Question 6

*Calculez **l'adresse réseau** et **l'adresse de diffusion** (broadcast) de votre machine.*

……..

……..

……..

……..

Question 7

Donnez la plage d'adresse IP disponible du réseau et le nombre d'hôtes maximum.

……..

……..

……..

Question 8

La configuration du réseau vous semble t-elle optimisée (du point de vue logique) ? Justifiez votre réponse.

……..

……..

……..

Question 9

Votre machine est-elle dans le même réseau que « SRV-nano-snir » ? Justifez votre réponse.

……..

……..

……..

Question 10

*Retrouvez l'adresse **IP** correspondante à une machine appartenant à votre réseau dont l'adresse **MAC** est « **b8:27:eb:fb:dd:7c** ». Indiquez la façon dont vous avez procédé et le résultat obtenu.*

……...

……...

……...

Question 11

L'adresse IP de « SRV-nano-snir » est-elle privée ou publique ?

……...

Question 12

Retrouvez l'adresse physique (MAC) de « SRV-nano-snir ». Indiquez la façon dont vous avez procédé et le résultat obtenu.

……...

……...

Question 13

*Vérifiez à l'aide d'un test d'écho de niveau 3, la connectivité avec l'adresse IP **212.27.48.10** ?*

Si le test est concluant, vous poursuivez à la question 14. Dans le cas contraire vous appelez le professeur !

Question 14

L'adresse IP testée à la question précédente est-elle privée ou publique ?

……...

Question 15

*Déterminez et expliquez (succinctement) l'itinéraire entre votre machine et l'IP **212.27.48.10**. Indiquez la façon dont vous avez procédé et le résultat obtenu.*

……...

……...

……...

……...

……...

DU TEXTE « CACHÉ » DANS DES IMAGES (LA SUITE)

Une autre technique permettant de cacher du texte dans une image repose sur la méthode dite d'usage de bits de poids faible (LSB) d'une image. Cette méthode consiste à modifier le bit de poids faible des pixels codant l'image.

Question 1

Comparez « visuellement » les deux images : ***Image.bmp*** *et* ***Image modifiée.bmp*** *et notez vos remarques.*

……...

Question 2

Vous vous en doutez, une chaîne de caractères a été dissimulée dans le fichier « Image modifiée.bmp ». Mais quelle est cette chaîne de caractères ?

Vous pouvez bien entendu utiliser l'éditeur hexadécimale pour essayer de retrouver la chaîne de caractères dissimulée comme vous l'avez fait avec la méthode précédente (partie 2/3).

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	42	4D	1A	50	02	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	13	01	00	00	B7	00	00	00	01	00	18	00	00	00
00000020	00	00	E4	4F	02	00	74	12	00	00	74	12	00	00	00	00
00000030	00	00	00	00	00	00	CB	CE	C5	CC	CF	C6	CC	CF	C6	CD
00000040	D0	C7	CE	D1	C8	CF	D2	C9	CF	D2	C9	D0	D3	CA	D0	D3
00000050	CA	D0	D3	CA	D1	D4	CB	D2	D5	CC	D2	D5	CC	D3	D6	CD
00000060	D4	D7	CE	D6	D7	CE	D8	D5	D0	D9	D5	D0	D9	D5	D0	D9
00000070	D5	D0	D8	D5	D0	D8	D5	D0	D8	D5	D0	D8	D5	D0	D7	D7
00000080	D1	D8	D8	D2	D6	D8	D2	D6	D8	D2	D7	D9	D3	D7	D9	D3
00000090	D7	D9	D3	D8	DA	D4	D9	DB	D5	D9	DB	D5	D9	DB	D5	D9
000000A0	DB	D5	D9	DB	D5	D9	DB	D5	D9	DB	D5	D9	DB	D5	D9	DB
000000B0	D5	D9	DB	D5	D9	DB	D5	DA	DC	D6	DA	DC	D6	DA	DC	D6
000000C0	DB	DD	D7	DB	DD	D7	DD	DD	D7	DD	DD	D7	DE	DE	D8	DE
000000D0	DE	D8	DE	DE	D8	DF	DF	D9	DF	DF	D9	DF	DF	D9	DD	DD
000000E0	D7	DD	DD	D7	DE	DE	D8	DE	DE	D8	DE	DE	D8	DF	DF	D9
000000F0	DF	DF	D9	DD	DE	DA	DD	DE	DC	DB	DE	DC	DB	DE	DC	DB
00000100	DE	DC	DB	DE	DC	DB	DE	DC	DB	DE	DC	DB	DE	DC	DD	E0
00000110	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE
00000120	DD	E0	DE	DB	E0	DE	DC	E1	DF	DB	E2	DF	DB	E2	DF	DB
00000130	E2	DF	DC	E1	DF	DC	E1	DF	DC	E1	DF	DC	E1	DF	DF	E2
00000140	E0	DF	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0
00000150	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1
00000160	E2	E0	E1	E2	E0	E1	E2	E0	DF	E2	E0	DF	E2	E0	DD	E2
00000170	E0	DD	E2	E0	DD	E2	E0	DD	E2	E0	DC	E3	E0	DC	E3	E0
00000180	DC	E3	E0	DC	E3	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	DF
00000190	E2	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	E0	E3
000001A0	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1
000001B0	E0	E3	E1	E0	E3	E1	E1	E2	E0	E1	E2	E0	E1	E2	E0	E2
000001C0	E3	E1	E2	E3	E1	E2	E3	E1	E3	E4	E2	E3	E4	E2	E1	E2
000001D0	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0
000001E0	E1	E2	E0	E1	E2	E0	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE
000001F0	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1

Extrait du code hexadécimal du fichier « Image.bmp »

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	42	4D	1A	50	02	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	13	01	00	00	B7	00	00	00	01	00	18	00	00	00
00000020	00	00	E4	4F	02	00	74	12	00	00	74	12	00	00	00	00
00000030	00	00	00	00	00	00	C9	CC	C7	CC	CD	C6	CD	CD	C4	CE
00000040	D0	C4	CD	D3	C8	CE	D1	CA	CF	D3	C9	D3	D1	C9	D1	D3
00000050	C9	D0	D1	CA	D1	D5	C9	D3	D5	CD	D1	D7	CC	D2	D4	CE
00000060	D4	D4	CD	D6	D5	CD	D9	D7	D0	DB	D5	D3	D9	D4	D0	DA
00000070	D4	D0	D9	D6	D0	D9	D5	D3	D8	D7	D1	DB	D4	D3	D5	D6
00000080	D3	DB	D9	D2	D4	DB	D1	D6	DA	D1	D7	DA	D2	D5	D8	D2
00000090	D4	D8	D3	DA	D8	D4	D8	DA	D4	D8	D9	D6	DB	D8	D5	DA
000000A0	D8	D5	D8	DA	D4	D8	D9	D6	D8	DB	D5	DA	DB	D7	D9	DB
000000B0	D5	D9	D9	D6	D8	DB	D5	DA	DE	D4	D9	DE	D5	D9	DC	D6
000000C0	D8	DC	D4	DB	DC	D7	DC	DE	D7	DE	DC	D5	DE	DE	D8	DE
000000D0	DE	D8	DE	DE	D8	DF	DF	D9	DF	DF	D9	DF	DF	D9	DD	DD
000000E0	D7	DD	DD	D7	DE	DE	D8	DE	DE	D8	DE	DE	D8	DF	DF	D9
000000F0	DF	DF	D9	DD	DE	DA	DD	DE	DC	DB	DE	DC	DB	DE	DC	DB
00000100	DE	DC	DB	DE	DC	DB	DE	DC	DB	DE	DC	DB	DE	DC	DD	E0
00000110	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE	DD	E0	DE
00000120	DD	E0	DE	DB	E0	DE	DC	E1	DF	DB	E2	DF	DB	E2	DF	DB
00000130	E2	DF	DC	E1	DF	DC	E1	DF	DC	E1	DF	DC	E1	DF	DF	E2
00000140	E0	DF	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0
00000150	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1
00000160	E2	E0	E1	E2	E0	E1	E2	E0	DF	E2	E0	DF	E2	E0	DD	E2
00000170	E0	DD	E2	E0	DD	E2	E0	DD	E2	E0	DC	E3	E0	DC	E3	E0
00000180	DC	E3	E0	DC	E3	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	DF
00000190	E2	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	DF	E2	E0	E0	E3
000001A0	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1	E0	E3	E1
000001B0	E0	E3	E1	E0	E3	E1	E1	E2	E0	E1	E2	E0	E1	E2	E0	E2
000001C0	E3	E1	E2	E3	E1	E2	E3	E1	E3	E4	E2	E3	E4	E2	E1	E2
000001D0	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0	E1	E2	E0
000001E0	E1	E2	E0	E1	E2	E0	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE
000001F0	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1	DF	DE	E1

Extrait du code hexadécimal du fichier « Image modifiée.bmp »

Question 3

*Comparez les en-têtes (**BITMAPFILEHEADER**) des deux fichiers et notez vos remarques.*

……..

Question 4

*Comparez les en-têtes de bitmap (**BITMAPINFOHEADER**) des deux fichiers et notez vos remarques.*

……

Question 5

Comparez les corps des images des deux fichiers et notez vos remarques.

……

Ci-dessous vous trouverez le code PHP qui a permis de cacher le message dans l'image BMP.

```
<?php

 $message = $_POST['message'];//1
 $lien = $_POST['image'];//2
 $octet_decoupe = array();//déclaration d'un tableau
 $message.= chr(26);//3
 $f_image = fopen($lien,'r+b');//4
 fseek($f_image, 54);//5
 for($i=0;$i<strlen($message);$i++){
        $caractere = $message[$i];
        $valeur_octet = ord($caractere);//6
        $octet_binaire = decbin($valeur_octet);//7
        $octet_binaire = str_pad($octet_binaire,8,'0',STR_PAD_LEFT);//8
        $octet_decoupe = str_split($octet_binaire,2);//9
        foreach($octet_decoupe AS $partie_octet){
               $octet_image = fread($f_image,1);//10
               $octet_image = ord($octet_image);//11
               $octet_image -= $octet_image%4;//12
               $partie_octet = bindec($partie_octet);//13
               $octet_image += $partie_octet;
               fseek($f_image, -1, SEEK_CUR);//14
               fputs($f_image, chr($octet_image));//15
        }
 }
 fclose($f_image);//16
 ?>
```

Les commentaires (//numéro dans le code) devraient vous aider à comprendre le fonctionnement du code et donc la façon dont les bits ont été modifiés.

1 : on stocke dans la variable **$message** le texte que vous souhaitez dissimuler dans l'image, ici elle va contenir un texte envoyé depuis un formulaire par une méthode POST.

2 : on stocke dans la variable **$lien** le chemin et le nom de l'image qu'on souhaite utiliser pour y dissimuler notre message « secret », le format de l'image doit être Bitmap (bmp) ;

3 : on ajoute à la variable contenant le message, le caractère défini à la 26ième position du code ASCII, obtenu grâce à la fonction **chr()**. Si vous regardez dans une table ASCII, vous verrez que cela correspond au caractère EOF (End Of File), donc un caractère non imprimable. Il ne va donc, en théorie, jamais apparaître dans notre message à dissimuler. Ce caractère va nous servir uniquement à indiquer la fin du message ;

4 : on ouvre le fichier image, l'option « r+b » permet d'ouvrir un fichier binaire en mode « lecture/écriture » ;

5 : on place notre curseur au niveau du 54ième pixel, pour sauter les informations contenues dans les « headers » de notre image ;

6 : la fonction **ord()** permet de retourner le code ASCII d'un caractère ;

7 : on récupère les bits de poids faible sous forme binaire ;

8 : on ajoute un zéro (0) devant le nombre binaire récupéré si nécessaire ;

9 : on convertit la chaîne de caractères contenu dans la variable **$octet_binaire** en un tableau de longueur deux (2) ;

10 : on récupère un seul octet, sous forme de caractère ;

11 : on convertit l'octet récupéré en nombre grâce à sa valeur correspondante dans la table du code ASCII ;

12 : on rend les bits de poids faible égaux à zéro ;

13 : on reconvertit l'octet en base décimal (base 10) pour pouvoir faire une addition ;

14 : la fonction **fseek()** renvoie une position donnée par rapport à la position actuelle, ensuite on écrase l'octet suivant, qu'on ne veut pas encore modifier ;

15 : on écrit dans le fichier image, en écrasant l'octet suivant ;

16 : on ferme le fichier image.

Question 6

Essayez de retrouver la chaîne de caractères cachée (au moins les quatre premiers caractères).

……

……

……

// Appeler le professeur pour faire valider votre travail avant de poursuivre

Dissimuler un texte dans une image c'est bien mais il est tout aussi important de savoir déchiffrer un texte qui serait caché dans une image et si possible de façon « automatique » car vous vous en êtes aperçu avec la question précédente l'opération peut s'avérer laborieuse si elle est faite « à la main ».

Ci-dessous vous trouverez le code PHP qui permet de retrouver le message cachée dans l'image BMP.

```
<?php
 $lien = $_GET['image'];//1
 $tampon ="";//variable qui servira de tampon
 $message ="";//variable qui contientra le message récupéré
 $f_image = fopen($lien,'rb');//2
 fseek($f_image,54);//3
 while(!feof($f_image)){ //4
        $octet_image = fread($f_image,1);//5
        $octet_image = ord($octet_image);//6
        $bits_pf = $octet_image%4;//7
        $bits_pf = decbin($bits_pf);//8
        $bits_pf = str_pad($bits_pf,2,'0',STR_PAD_LEFT);//9
        $tampon.= $bits_pf;//10
        if(strlen($tampon) == 8){ //11
               $tampon = bindec($tampon);//12
               if($tampon == 26){
                      echo ('<h3><font color="red">'.$message.'</font></h3>')//13
                      return;
               }
               $message.= chr($tampon);//14
               $tampon ="";//15
        }
 ?>
```

1 : on crée une variable **$lien** qui contient le chemin et le nom du fichier image dont on souhaite récupérer (si il existe) le message caché ;

2 : on ouvre notre fichier image en mode « lecture » uniquement, pour cette opération (récupérer et afficher le message) il n'est pas nécessaire d'ouvrir le fichier en essayant de le modifier, d'où le mode « lecture »;

3 : on saute le « header » ;

4 : on crée une boucle « while » qui consistera à effectuer un certain nombre d'opérations de manière répétitive tant que l'image ne sera pas entièrement lu ;

5 : on lit un caractère du fichier image ;

6 : on retourne le code ASCII du caractère lu à l'instruction 5, c'est le rôle de la fonction **ord()** ;

7 : on stocke dans la variable **$bits_pf** le reste de la division de la valeur contenue dans la variable **$octet_image** par quatre, en d'autres termes on stocke la valeur du modulo ;

8 : on récupère les bits de poids faible sous forme binaire ;

9 : on ajoute un zéro si nécessaire ;

10 : on concatène les bits de poids faible trouvés à la valeur initiale stocké dans la variable **$tampon** ;

11 : on effectue une condition qui prendra effet si et seulement si la taille de la valeur contenu dans la variable tampon est égal à huit;

12 : on convertit en base décimale la valeur de notre variable tampon ;

13 : on affiche le message caché dans l'image ;

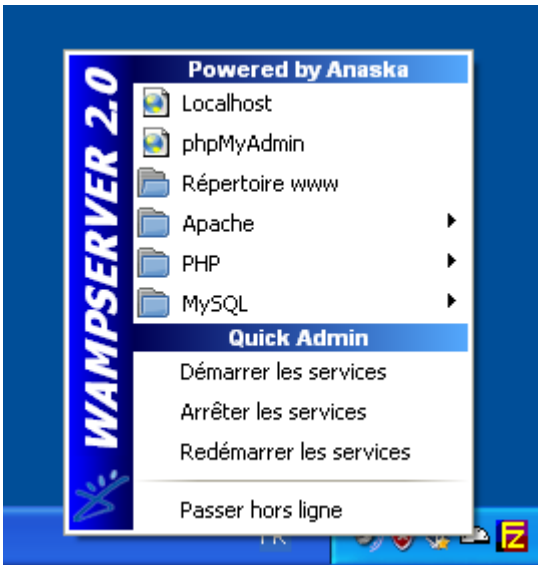
14 : si l'on n'est pas arrivé à la fin du message, donc le caractère renvoyé par la fonction **chr()** est différent du caractère EOF (rappelez-vous, c'est ce caractère qui nous permet de savoir si on a atteint la fin du message), on ajoute le caractère trouvé ;

15 : enfin on réinitialise le tampon.

« Il est temps de tester l'application PHP »

Vous utiliserez la machine virtuelle nommée « **MV_XP_WampServer** ».
Avant de lancez la machine virtuelle, pensez à configurer les paramètres, notamment les paramètres réseau et remplacez entre autre le premier octet de l'adresse physique de votre machine par le numéro de votre poste.

Après avoir lancé la VM, cliquez sur « WampServer » dans la barre des tâches (en bas à droite).



Puis choisissez « **Localhost** », cliquez sur « **Steganographie** » et testez l'application.