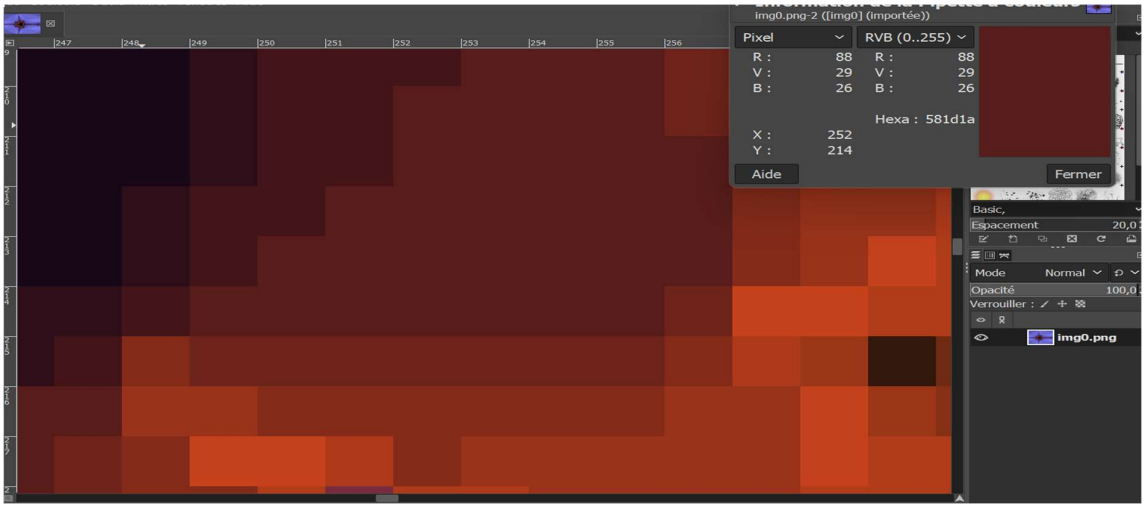
Cybersécurité des services informatiques

Professeurs : Mr JOBARD

TP N°1 : Images et sécurité informatique / Stéganographie

❖ Couleur d'un pixel

1°) A l'aide de GIM, la couleur du pixel de coordonnées (252,214) de l'image : imag0.png est une sombre teinte de couleur rouge.
2°) Le code hexadécimal, utilisé pour la couleur en HTML est : 581d1a



❖ Description du procédé stéganographique

1°) Apres une vérifications j'ai pu parvenir à conclue que les deux points de cordonnées (0,0) et (0,1), de l'image img0.png sont exactement de la même couleur.

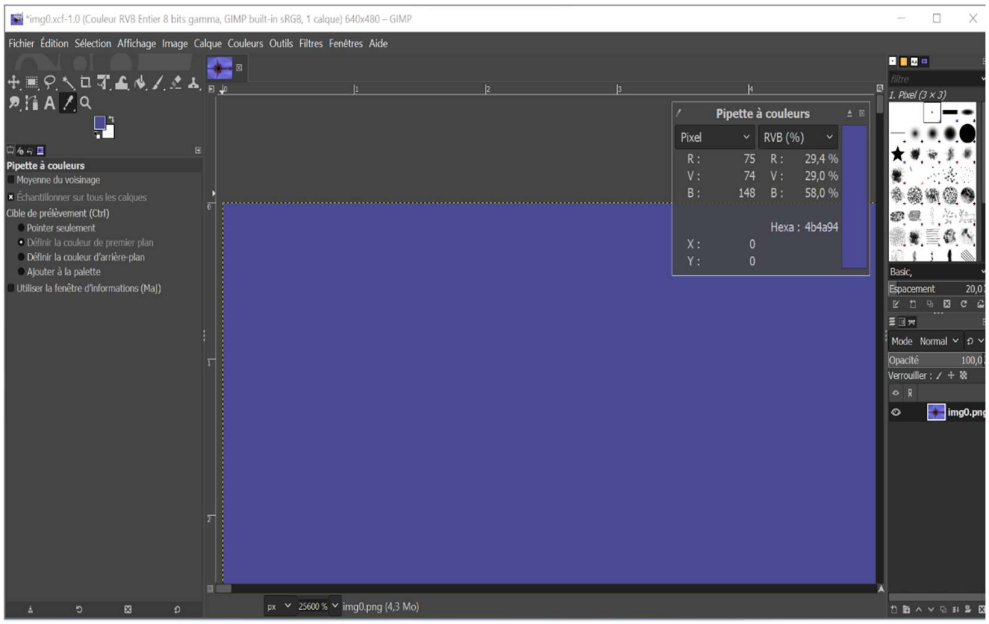


Image img0.png /Pixel de coordonnées (0, 0)

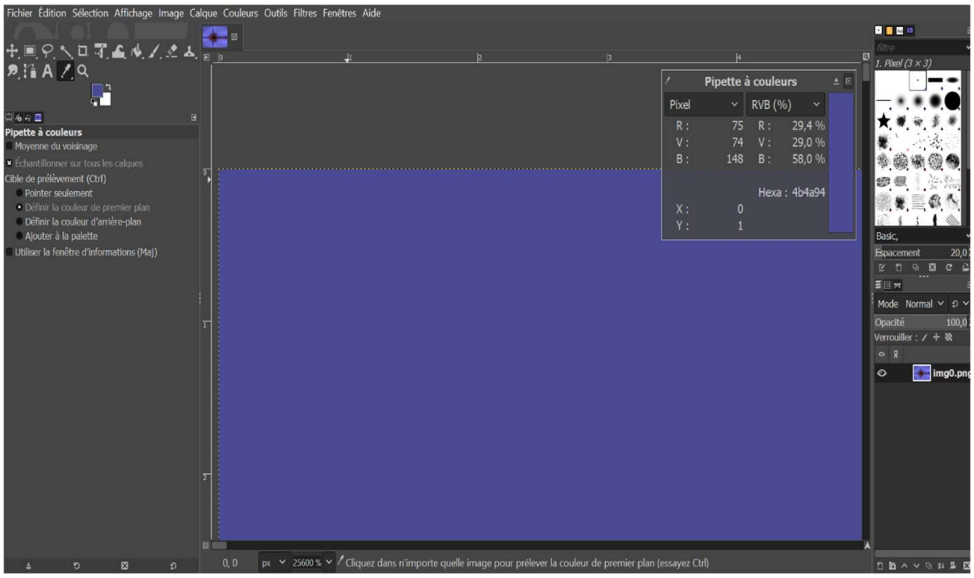
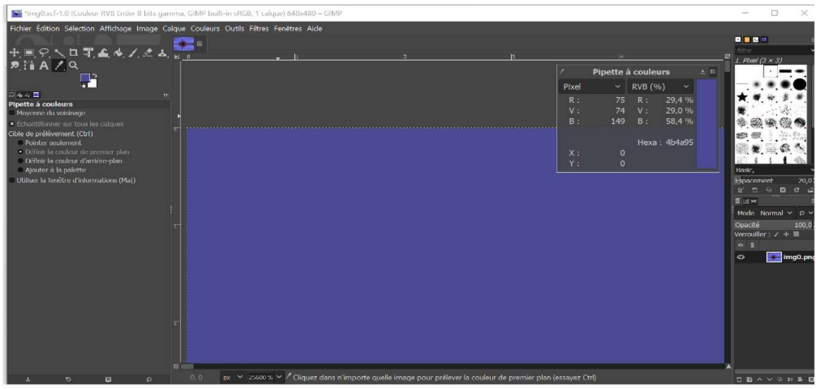


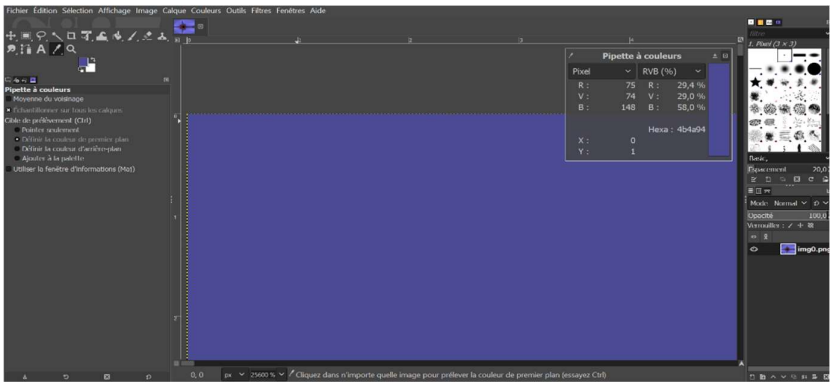
Image img0.png /Pixel de coordonnées (0, 1)

2°) j'ai modifié la couleur du pixel de coordonnées (0,0) en ajoutant 1 à la composante bleue de sa couleur, en suivant les consignes.

3°) Apres la modification, je ne vois pas à l'œil nu la différence de couleur avec le pixel voisin par compte en utilisant l'outil pipette à couleur y'a un changement de couleur.



Pixel de coordonnées (0,0) après modification, en utilisant l'outil pipette à couleur



Pixel voisin de coordonnées (0,1)

❖ Retrouver un message

J'ai utilisé Gimp pour retrouver le nombre l du message dissimulé, en commençant d'abord par noter les valeurs de composantes bleues. Puis je détermine les valeurs de leur bit de poids faible. Tout cela est résumé dans le tableau ci-dessous

Coordonnées pixel	Valeurs de composantes bleues	Valeurs de bit de poids faible
(0,0)	148	0
(1,0)	148	0
(2,0)	148	0
(3,0)	148	0
(4,0)	148	0
(5,0)	149	1
(6,0)	148	0
(7,0)	148	0

Le nombre l du message dissimulé : ***l*** = 00000100= 4

Pour trouver les codes binaires des caractères cachés et révéler le message, trouvons d'abord :

Le nombre de pixels dissimulant QUI est donc égal à 8xl d'où 32 .

3°) Le code binaire des caractères cachés ?

Je fais le même procédé pour trouver les codes binaires des caractères caches, en notant les valeurs de composantes bleues et leurs bits de poids faibles. Or je connais le nombre de pixels dissimulés

Coordonnées pixel	Valeurs de composantes bleues	Valeurs de bit de poids faible
(0,1)	148	0
(1,1)	149	1
(2,1)	148	0
(3,1)	148	1
(4,1)	148	0
(5,1)	149	1
(6,1)	148	0
(7,1)	148	0
(8,1)	148	0
(9,1)	141	1
(10,1)	148	0
(11,1)	148	0
(12,1)	148	0
(13,1)	148	0
(14,1)	149	1
(15,1)	148	0
(16,1)	148	0
(17,1)	148	0
(18,0)	149	1
(19,1)	148	0
(20,1)	148	0
(21,1)	148	0

(22,1)	148	0
(23,1)	148	0
(24,1)	148	0
(25,1)	148	0
(26,1)	149	1
(27,1)	148	0
(28,1)	148	0
(29,1)	148	0
(30,1)	148	0
(31,1)	149	1
(32,1)	148	0

- Le code binaire des caractères cachés est : 01010100010000100010000000010000010

4°) ET enfin je vais trouver le message dissimulé en convertissant le code binaire en texte. Pour cela je vais utiliser PREPOSTSEO (traducteurs binaire).

01010100010000100010000000010000010 === » TB !

- Le message dissimulé est : TB !

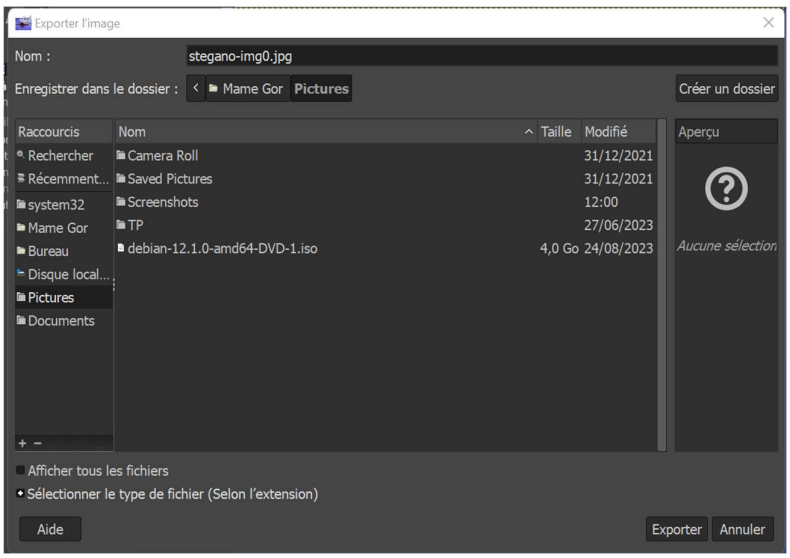
❖ Dissimuler un message

À vous de dissimuler maintenant !

J'ai caché le message « SLT ! » dans l'image : stegano-img0. En premier lieu J'ai codé le texte en binaire. Puis j'ai modifié l'image de façon suivante **: le bit de poids faible du i^e octet est égal à la valeur du i^e bit du message**. Ensuite j'ai envoyé l'image à mon ami pour qu'il trouve le message.

❖ Choix du format de sauvegarde du fichier

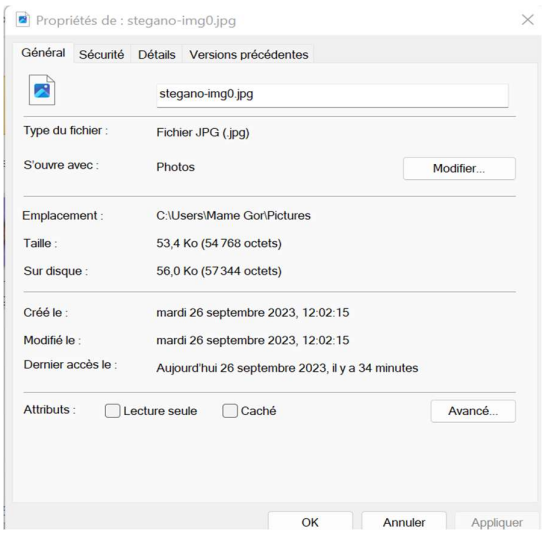
1°) j'ai repris l'image(stegano-img0.png) que j'avais extrait le message qui y était dissimulé, en utilisant Gimp. Puis je l'enregistre sur disque au format jpg, en laissant inchangés les paramètres par défaut.

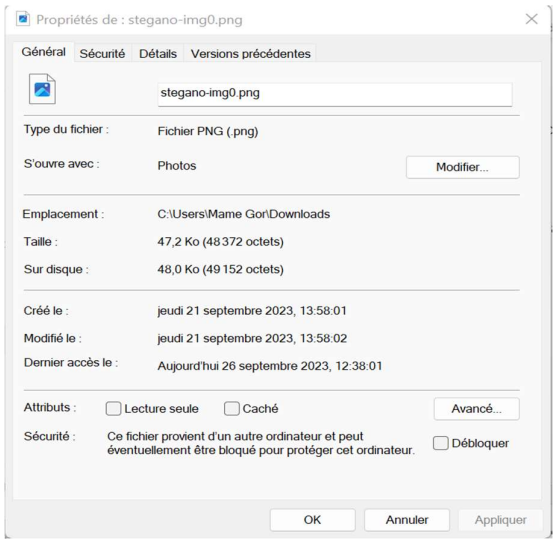


2°) Par la suite je vais charger l'image(stegano-img0.png) au format jpg avec Gimp et tenter de retrouver l'information dissimulée.
Après la modification de l'extension(format) png de l'image en jpg, je constate que :

- La composante bleue de la couleur des pixels est différent (ligne d'ordonnée 1) par exemple au coordonnées (0,1) la valeur bleue est égal à 149 au lieu de 148.
- Je n'arrive pas à trouver le message d'avant.

3°) Je vais comparer la taille des deux fichiers aux formats jpg et png.



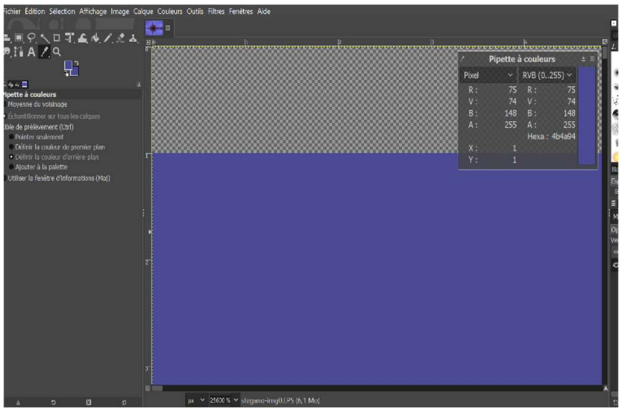


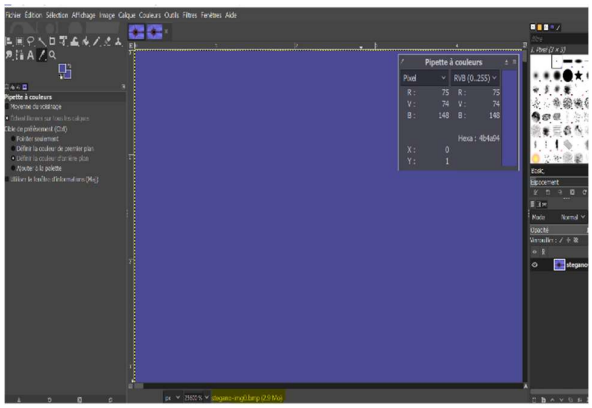
Fichier jpg **Fichier png**

- En fessant la comparaison de taille des deux fichiers je constate que la taille du fichier jpg (53,4 Ko) est supérieur à celle du fichier png (47,2 Ko).
- Je pense que l'extension d'un fichier joue à la taille du fichier.

4°) J'ai examiné d'autres formats possibles tels que :bmp, svg, tiff, psd, pdf .

Les formats qui conviennent pour notre procédé stéganographique sont : tiff, psd bmp





Format eps **Format bmp**

Conclusion

D’après mes recherches d’autres types de fichier sont utilisés pour faire stéganographié. Par exemples :

- Stéganographie de texte
- Stéganographie vidéo
- Stéganographie audio

« La stéganographie est intéressante dans le cadre de la cybersécurité, car les groupes de ransomwares et d'autres acteurs menaçants cachent souvent des informations pour attaquer une victime. Par exemple, ils peuvent cacher des données, dissimuler un outil malveillant ou envoyer des instructions à des serveurs de commande et de contrôle. Toutes ces informations peuvent se trouver dans des fichiers image, vidéo, audio ou texte et paraitre inoffensives. »

Exemples de stéganographie utilisée dans le cadre de cyberattaques

Skimming

« En 2020, la plateforme néerlandaise de sécurité du commerce électronique Sansec a publié des recherches qui ont démontré que des acteurs malveillants avaient intégré des programmes de skimming à des fichiers SVG (Scalable Vector Graphics) sur les pages de paiement de sites de commerce électronique. Les attaques impliquaient une charge utile malveillante dissimulée dans des images SVG ainsi qu'un décodeur caché séparément dans d'autres parties des pages Internet.

Les utilisateurs qui saisissaient leurs coordonnées sur les pages de paiement compromises ne remarquaient rien de suspect, car les images représentaient de simples logos d'entreprises bien connues. Étant donné que la charge utile était intégrée à une syntaxe d'élément SVG à première vue correcte, les programmes d'analyse de sécurité standard recherchant une syntaxe invalide ne détectaient pas l'activité malveillante. »

Source : https://www.kaspersky.fr