TANET Margaux
BTS SIO 1ère année

04/10/2022

TP noté « image et sécurité informatique » / Stéganographie

SOMMAIRE

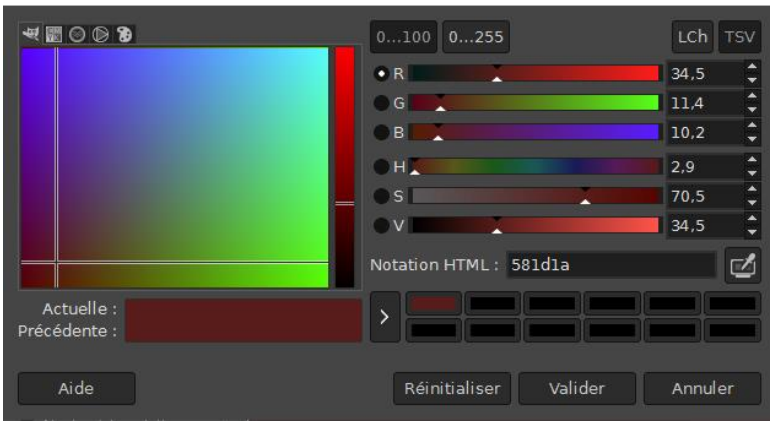
Pré-requis :

Codage des couleurs :

Couleur	Rouge	Vert	Bleu	Code HTML
Noir	0	0	0	#000000
Blanc	255	255	255	#FFFFFF
Gris	même valeur pour les 3			
Rouge	255	0	0	#FF0000
Jaune	255	255	0	#FFFF00
Vert	0	255	0	#00FF00
Cyan	0	255	255	#00FFFF
Bleu	0	0	255	#0000FF
Magenta	255	0	255	#FF00FF

TANET Margaux
BTS SIO 1ère année

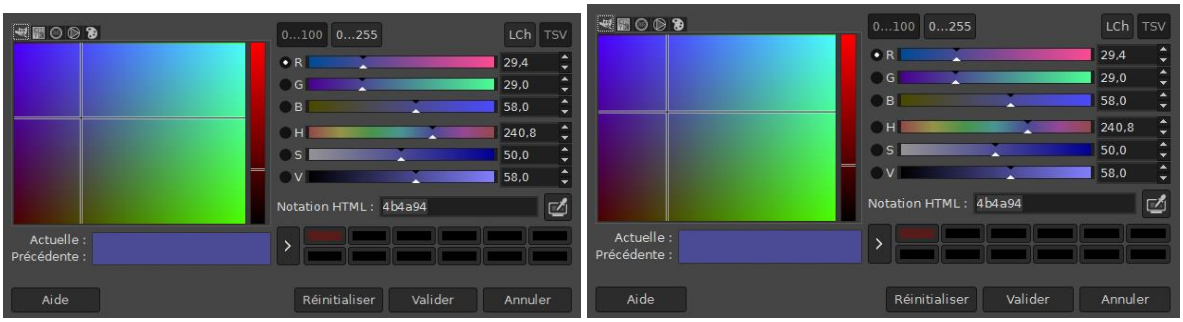
Couleur d'un pixel



Le code héxadécimal pour cette couleur est 581d1a.

Description du procédé stéganographique

1.

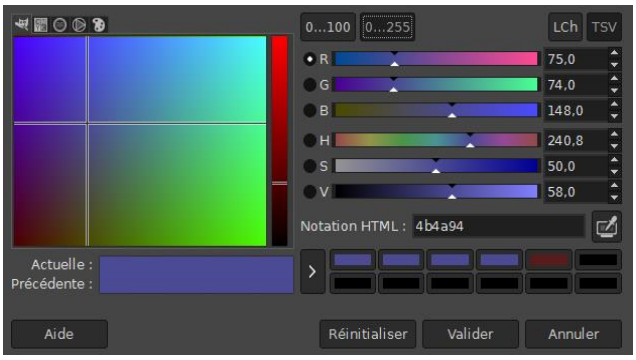


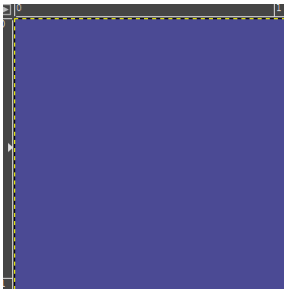
Grâce à l'outil pipette nous pouvons constater que les deux points sont de la même couleur en effet, il apparait « 4b4a94 ».

TANET Margaux
BTS SIO 1ère année

2.

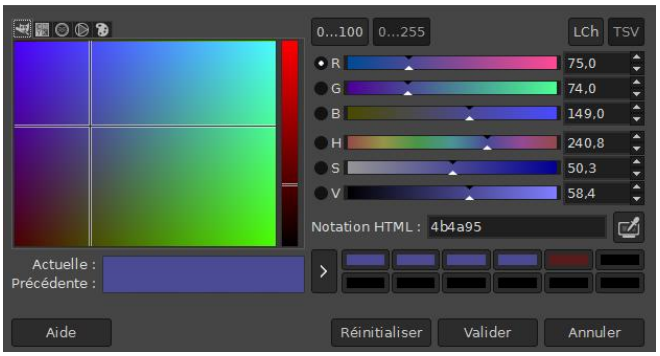
Voici le paramétrage avant :

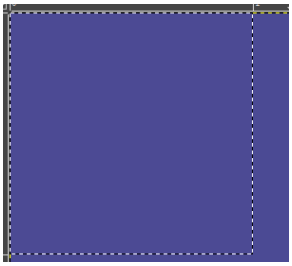




Pixel avant l'application de changement de couleur

Situation après le paramétrage (ajout B : +1) :

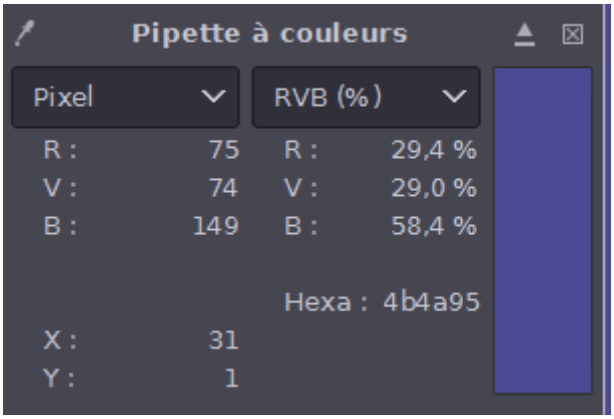




3. Non il n'y a aucune différence de couleur avec le pixel voisin, il nous est de toute façon impossible de le voir à l'œil nu (sauf dans le cas d'un gros zoom effectué).

TANET Margaux
BTS SIO 1ère année

Retrouver un message



En cliquant sur la première ligne horizontale sur la couleur bleue on voit :

Sur les 8 premiers pixels on remarque que « B » change soit en 148 soit en 149.

149 étant en hexadécimal donne en binaire 10100100<u>1</u> il se termine par un 1 c'est pour cela qu'on lui attribue le 1 pour la suite. C'est la valeur appelé bit de point faible.

A l'inverse 148 en binaire donne 10100100<u>0</u> se terminant par un 0 on lui attribue 0. C'est la valeur appelé bit de point faible.

Sur la première ligne on remarque : 148 148 148 148 148 149 148 148 ce qui correspond en binaire a 00000100. Il faut donc le convertir en décimal cela donne 4. La longueur du message sera de 4 caractères et le nombre de pixels sera de 32 pixels. 8 bits représentent 1 octet. En effet, 4 (longueur) X 8=32 bits/ pixels.

Donc pour la ligne du dessous il faut faire la même analyse/méthode que la ligne que nous venons de voir au-dessus. Nous pouvons utiliser un convertisseur pour convertir du binaire en décimal. Ensuite se référer au code ASCII (norme informatique pour le codage des caractères) et trouver le nombre décimal qui correspond au code binaire.

La table de code ASCII :

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

- Cela donne pour le premier paquet de 8 bits :

148 149 148 149 148 149 148 148 => 01010100

En décimale cela donne 84 et en code ASCII T.

- Sur les 8 prochains pixels :

148 149 148 148 148 148 149 148 => 01000010

En décimale 66 qui correspond à B en code ASCII.

- Sur les 8 prochains pixels :

148 148 149 148 148 148 148 148 => 00100000

En décimale 32 et représente un espace en code ASCII

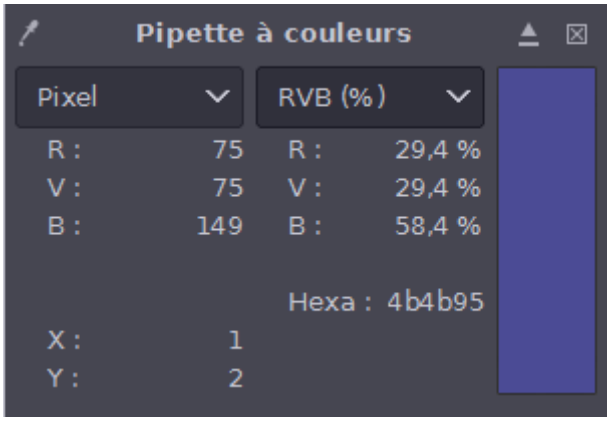
- Sur les 8 prochains pixels :

148 148 149 148 148 148 148 149 => 00100001

En décimale 33 et en code ASCII un point d'exclamation « ! »

Le message à la fin donne « TB ! »

Choix du format de sauvegarde du fichier

2. 

On remarque que 149 où qu'on clique, s'affiche. La donnée 148 qui s'affichait avant ne s'affiche plus.

3. Le fichier PNG a une taille plus grande (48 ko) que le format JPG (28ko).

4. GIF est possible aussi comme format en effet il compresse énormément et il divise la taille de fichier par deux (25 ko). Mais le format JPG fonctionne aussi bien.

TANET Margaux
BTS SIO 1ère année

Vers l'infini est au-delà

La stéganographie a pour but de transmettre un message de manière inaperçue au sein d'un autre message. La stéganographie s'utilise généralement et le plus souvent maintenant pour des cyberattaques et des diffusions de malwares mais reste exploitable dans de nombreux domaines. Il est facile de cacher des fichiers secrets dans des images BMP, GIF, JPEG, PNG, fichiers audio/vidéos ou fichiers WAV.

En 2016, Sundown, connu pour distribuer des chevaux de Troie à distance via des mails de phising (technique frauduleuse destinée à leurrer l'internaute pour l'inciter à communiquer des données sous diverses formes), a été mise à jour et a découvert qu'il utilisait la stéganographie pour masquer le code d'exploitation.

En aout 2017, le ransomware SyncCrypt récupère une image JPG dans lequel se trouve un fichier zip à la fin de l'image. Ce zip contient les composants qui infecteront l'ordinateur par la suite.

On remarque que c'est déjà un mode de propagation des malwares qui évite la détection des antivirus.

Des chercheurs de FortiGuard ont observé une hausse des échantillons de logiciels malveillants utilisant la stéganographie qui dissimuler des charges malveillantes dans les mèmes transmis sur les réseaux.

Pour se protéger de ces attaques cela devient de plus en plus compliqué, les ransomwares et les malwares deviennent de plus en plus innovants. Il faut donc adopter des technologies modernes de protection entre autre de type endpoint. Un endpoint consiste à protéger les points d'accès réseau informatique impliquant les ordinateurs, les téléphones, etc. Pour le cas des entreprises, il serait préférable pour ces employés, de les mettre au courant que les fichiers images peuvent aussi contenir des codes malveillants.

Le site Emisisoft a mis en place certains logiciels pour bloquer certains programmes qui tenteraient de nous connecter à un site internet malveillant même si le site masque de la stéganographie. De même pour un fichier contenant du code malveillant ainsi que l'exécution de fichier.

https://blog.emsisoft.com/fr/10637/les-merveilles-et-horreurs-de-la-steganographie-numerique/

De même, McAfee renforce sa sécurité de ses mécanismes de fournitures et de distributions de logiciels utilisés justement pour la protection de ces attaques.

Pour retenir les grandes lignes de cette activité :

La stéganographie est l'étude des procédés de dissimulation d'une information dans une autre. Une couleur peut être codée en utilisant un format RVB indiquant la quantité de rouge, vert et bleu (qu'il faut mélanger pour obtenir la couleur souhaitée).

La quantité de chaque couleur peut être exprimée par un nombre compris entre 0 et 255**.** La représentation binaire de ces nombres est 0 et 1**.** Chaque chiffre 0 ou 1 est appelé un bit et une suite de 8 bits est appelée un octet.

Une image est composée d'une série de points de couleur appelés pixels. Si l'on modifie seulement la valeur du bit de poids faible (le bit le plus à droite de l'octet) de chaque composant, l'impact sur le code sera faible. Quand on modifie un pixel la modification est invisible à l'œil nu. Ce type de technique peut être utilisé pour la diffusion de malwares par exemple.