Challenge Insomni'Hack 2008

Société SCRT - Préverenges - Suisse

Solution des épreuves

proposées par Bruno Kerouanton

http://bruno.kerouanton.net

Introduction

La société SCRT (www.scrt.ch) basée à Préverenges (Suisse) a organisé le 8 février 2008 un concours nocturne destiné aux personnes s'intéressant à la sécurité des systèmes d'information.
Durant toute la nuit, les personnes inscrites ont pu se confronter à différentes épreuves mettant en oeuvre des problèmes de sécurité informatique variés.

J'ai à cette occasion préparé une « épreuve à tiroirs ». Le présent document donne les réponses à cette épreuve.

Comme pour tous les concours en sécurité informatique (et en Perl !), « *There is more than one way to do it* », la solution donnée ci-dessous n'est donc qu'indicative, et vous pouvez avoir trouvé (... ou pas !) la solution d'une autre manière, comme j'ai pu le constater en faisant bêta-tester le challenge auprès de l'équipe de SCRT organisant le concours. Vous pouvez par exemple utiliser d'autres outils parce que vous les maîtrisez mieux que ceux que j'utilise, ce n'est pas un problème ! Chaque approche est bonne du moment que vous arriviez au résultat demandé.

Si vous n'avez pas participé à la nuit InsomniHack, ou que vous voulez vous amuser à refaire les épreuves, vous trouverez sur mon site web (http://bruno.kerouanton.net) en y faisant une recherche sur "insomni'hack" :

- le fichier que j'ai proposé pour le challenge,
- le présent document présentant les solutions

Bonne lecture !

Bruno Kerouanton

Table des matières

Contexte

Chaque candidat reçoit[1] :

- un fichier d'environ 64Ko intitulé "challenge-insomnihack", et aucune instruction. Il doit se débrouiller pour savoir quoi en faire.

- une collection d'outils (basée sur la collection µTools) permettant de tout résoudre; reste à savoir lesquels utiliser, et comment s'en servir...

Bien que représenté par un seul petit fichier; cette épreuve est en fait composée de mini-challenges indépendants, chacun ayant un rapport avec une spécialité différente :

1. forensics, recherche d'informations
2. win32.exe, épreuve sur un exécutable Win32
3. crypto, épreuve mettant en oeuvre de la cryptographie
4. reverse, épreuve de désassemblage
5. oldschool. l'épreuve "subsidiaire", pour les anciens programmeurs sur 8bits !

Le niveau des épreuves est variable, mais celles-ci sont généralement assez simples car le challenge ne dure que quelques heures. La diversité permet aux candidats débutants de trouver un ou deux résultats, tandis que les meilleurs pourront creuser un peu plus avec les autres mini-épreuves.

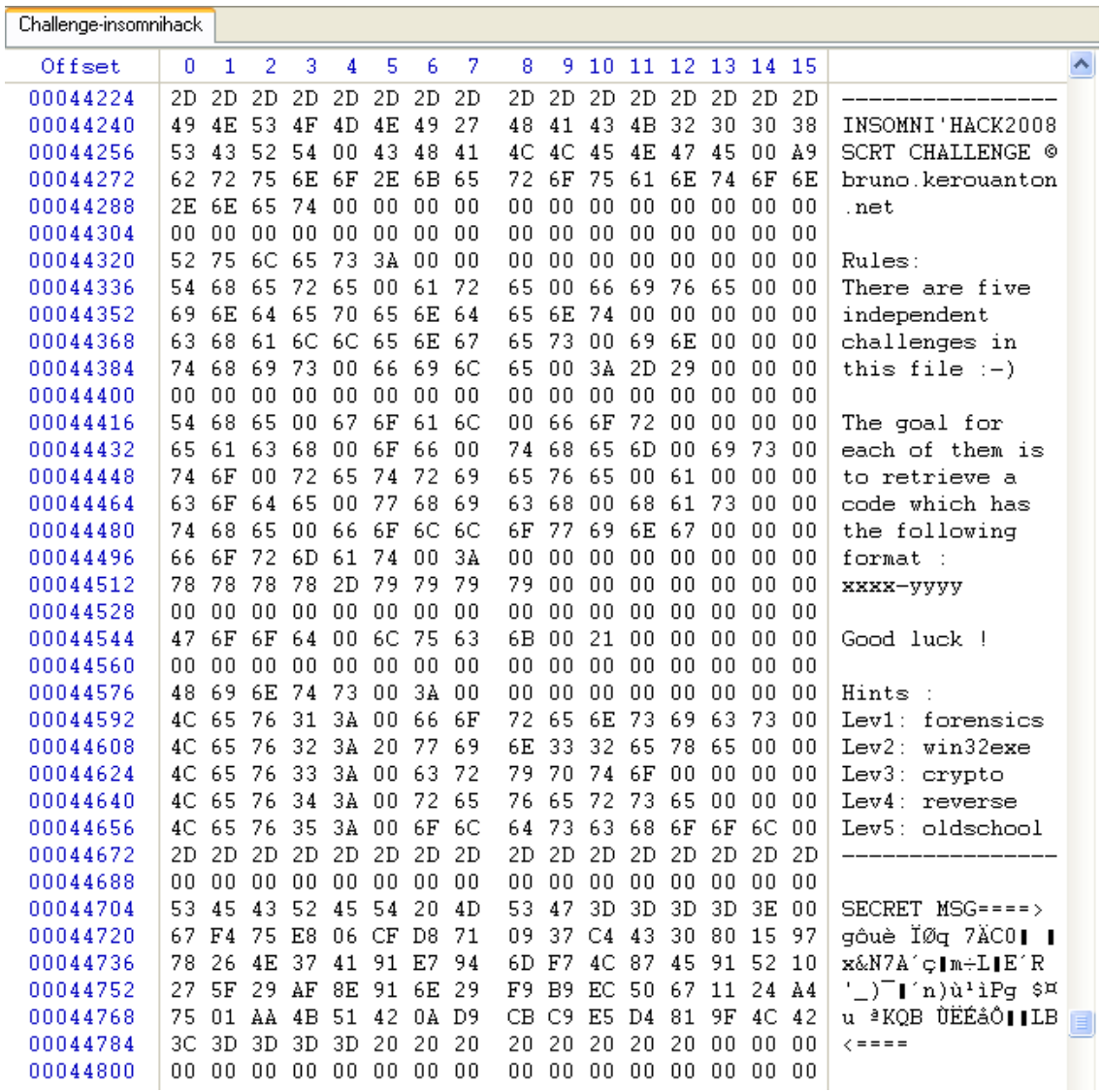
1 Fichiers téléchargeables via http://bruno.kerouanton.net

1. Démarrage de l'épreuve : forensics... savoir ce qu'il faut faire !

On doit tout d'abord déterminer la nature du fichier qu'il possède, et ce qu'il faut faire.

1.1 Trouver le mode d'emploi

Vu la taille très réduite du fichier, un simple drag-n-drop dans un éditeur texte ou hexadécimal quelconque tel que *FileAlyzer*, *WinHex* ou *PsPad* (mais pas avec *notepad.exe* !) permettra de trouver des choses intéressantes, telles que le "mode d'emploi" du challenge situé à l'offset 44224 :



Challenge-insomnihack

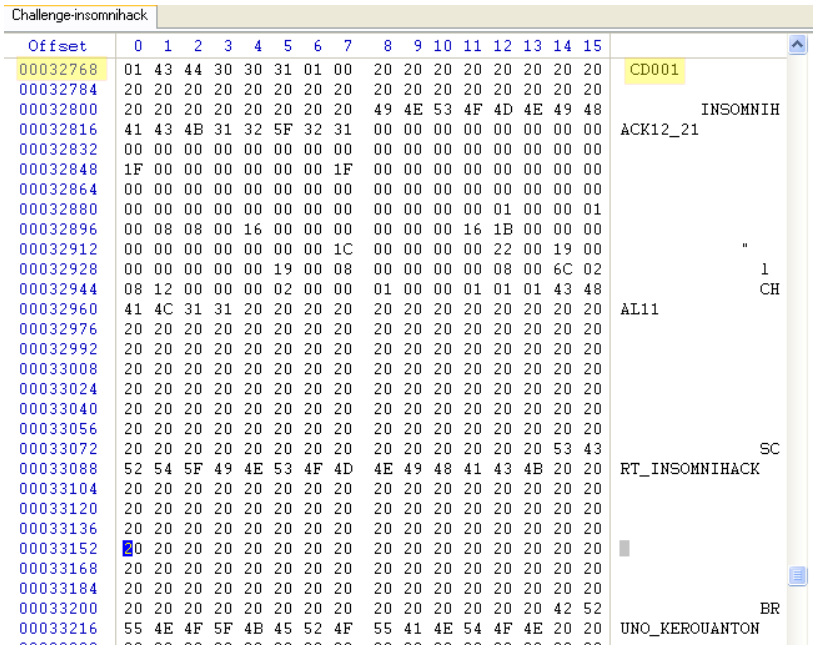
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00044224	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	----------
00044240	49	4E	53	4F	4D	4E	49	27	48	41	43	4B	32	30	30	38	INSOMNI'HACK2008
00044256	53	43	52	54	00	43	48	41	4C	4C	45	4E	47	45	00	A9	SCRT CHALLENGE ©
00044272	62	72	75	6E	6F	2E	6B	65	72	6F	75	61	6E	74	6F	6E	bruno.kerouanton
00044288	2E	6E	65	74	00	00	00	00	00	00	00	00	00	00	00	00	.net
00044304	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00044320	52	75	6C	65	73	3A	00	00	00	00	00	00	00	00	00	00	Rules:
00044336	54	68	65	72	65	00	61	72	65	00	66	69	76	65	00	00	There are five
00044352	69	6E	64	65	70	65	6E	64	65	6E	74	00	00	00	00	00	independent
00044368	63	68	61	6C	6C	65	6E	67	65	73	00	69	6E	00	00	00	challenges in
00044384	74	68	69	73	00	66	69	6C	65	00	3A	2D	29	00	00	00	this file :-)
00044400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00044416	54	68	65	00	67	6F	61	6C	00	66	6F	72	00	00	00	00	The goal for
00044432	65	61	63	68	00	6F	66	00	74	68	65	6D	00	69	73	00	each of them is
00044448	74	6F	00	72	65	74	72	69	65	76	65	00	61	00	00	00	to retrieve a
00044464	63	6F	64	65	00	77	68	69	63	68	00	68	61	73	00	00	code which has
00044480	74	68	65	00	66	6F	6C	6C	6F	77	69	6E	67	00	00	00	the following
00044496	66	6F	72	6D	61	74	00	3A	00	00	00	00	00	00	00	00	format :
00044512	78	78	78	78	2D	79	79	79	79	00	00	00	00	00	00	00	xxxx-yyyy
00044528	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00044544	47	6F	6F	64	00	6C	75	63	6B	00	21	00	00	00	00	00	Good luck !
00044560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00044576	48	69	6E	74	73	00	3A	00	00	00	00	00	00	00	00	00	Hints :
00044592	4C	65	76	31	3A	00	66	6F	72	65	6E	73	69	63	73	00	Lev1: forensics
00044608	4C	65	76	32	3A	20	77	69	6E	33	32	65	78	65	00	00	Lev2: win32exe
00044624	4C	65	76	33	3A	00	63	72	79	70	74	6F	00	00	00	00	Lev3: crypto
00044640	4C	65	76	34	3A	00	72	65	76	65	72	73	65	00	00	00	Lev4: reverse
00044656	4C	65	76	35	3A	00	6F	6C	64	73	63	68	6F	6F	6C	00	Lev5: oldschool
00044672	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	----------
00044688	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00044704	53	45	43	52	45	54	20	4D	53	47	3D	3D	3D	3D	3E	00	SECRET MSG====>
00044720	67	F4	75	E8	06	CF	D8	71	09	37	C4	43	30	80	15	97	gôuè ÏØq 7ÄC0▌ ▌
00044736	78	26	4E	37	41	91	E7	94	6D	F7	4C	87	45	91	52	10	x&N7A´ç▌m÷L▌E´R
00044752	27	5F	29	AF	8E	91	6E	29	F9	B9	EC	50	67	11	24	A4	'_)¯▌´n)ù¹ìPg $¤
00044768	75	01	AA	4B	51	42	0A	D9	CB	C9	E5	D4	81	9F	4C	42	u ªKQB ÙËÉåÔ▌▌LB
00044784	3C	3D	3D	3D	3D	20	20	20	20	20	20	20	20	00	00	00	<====
00044800	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

On notera :

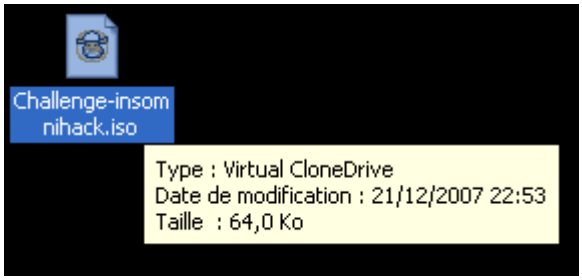
- qu'il faut trouver un numéro de série sous la forme **xxxx-yyyy** pour remporter chacune des épreuve,
- qu'il y a cinq épreuves indépendantes mettant en oeuvre des compétences différentes,
- Si on est observateur, on remarquera également un "messsage secret" situé à la suite ! Ce message sera utile pour l'épreuve crypto. J'ai volontairement indiqué où il se trouvait pour ne pas faire perdre trop de temps aux candidats, la nuit étant longue !

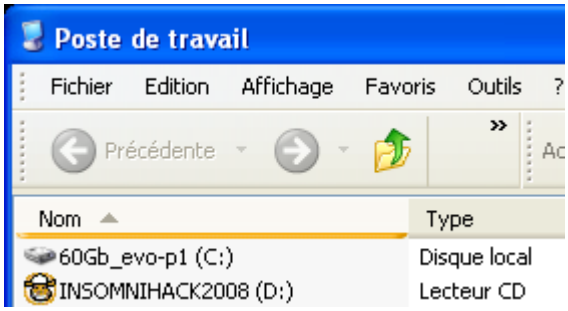
1.2 Trouver le format du fichier

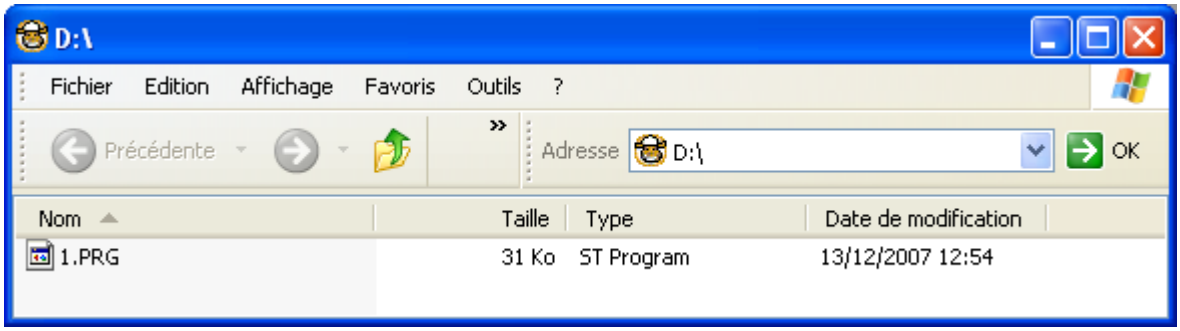
L'utilisation d'un éditeur hexadécimal, *FileAlyzer* par exemple, devrait montrer qu'il s'agit d'une image ISO, comme on le voit bien à l'offset singulier 32768 !



Après avoir renommé l'extension du fichier en .iso, on pourra monter l'image ISO via un outil spécialisé (VMware, Virtual Clone Drive, etc.) si on le souhaite, puis afficher son contenu :

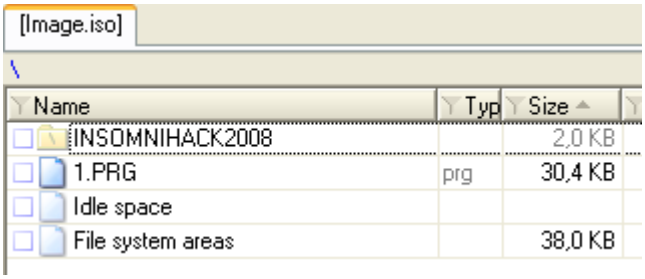






Le contenu révèle un fichier 1.PRG de 31Ko, que l'on pourra sauvegarder pour plus tard (voir la partie concernant l'épreuve "Oldschool").

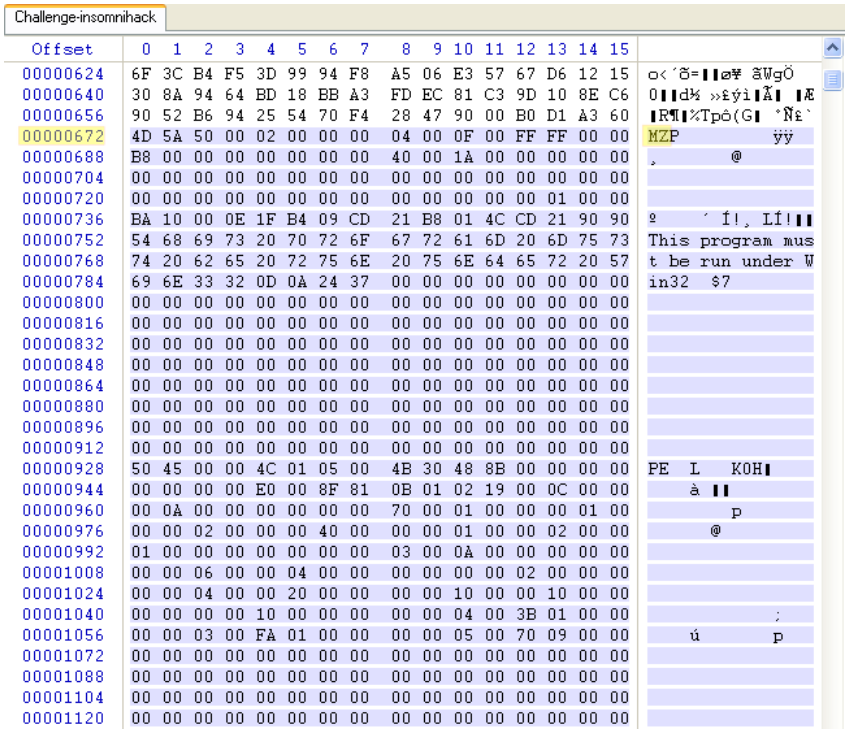
L'image ISO ne contient rien de plus en apparence, l'accès au répertoire INSOMNIHACK2008 étant rendu difficile car j'ai corrompu volontairement certains octets ;)



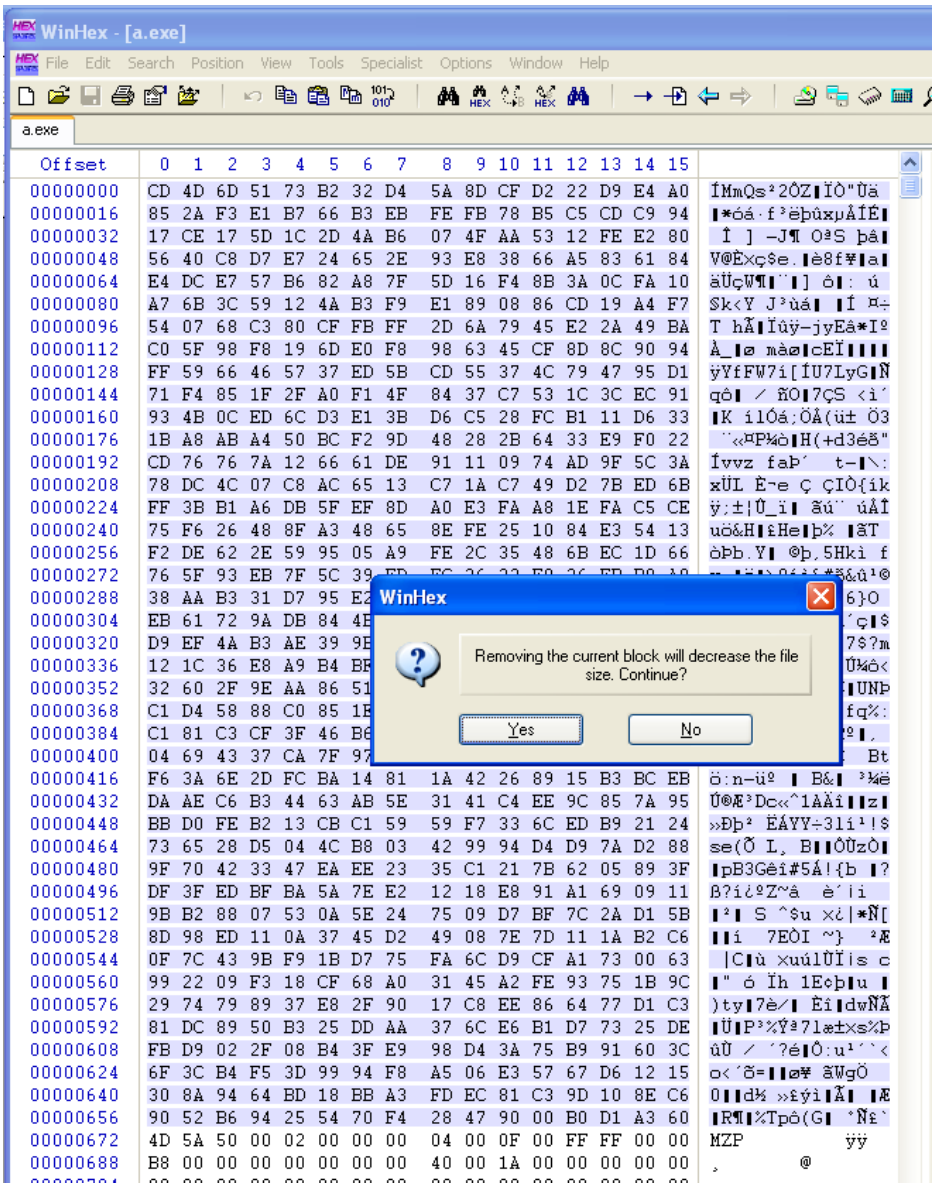
2. Mini-épreuve : win32.exe

L'épreuve de reverse n'en est pas vraiment une, car je n'ai pas eu le temps de m'y atteler. Il s'agit plus d'une recherche forensics sur un .exe !

L'utilisateur averti aura remarqué à l'offset 672 une signature "MZ", caractéristique des programmes exécutables au format PE :

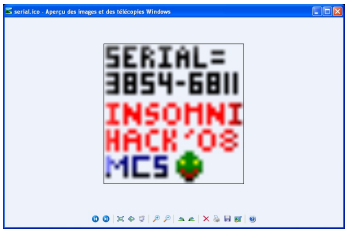


Simplement avec un éditeur texte ou hexadécimal standard, on pourra alors extraire le binaire en supprimant tout ce qui est au dessus du "MZ", et sauvegarder le résultat dans un fichier ".exe", cela fonctionne bien et c'est rapide ! Il n'est même pas nécessaire de supprimer les données qui traînent à la suite. Exemple avec WinHex :



En fait, le programme ne fait absolument rien, ou du moins ne contient pas d'épreuve de reverse... (il s'agit en réalité du programme LoadDll.exe" fourni avec OllyDbg). Le numéro de série se trouve en réalité dans l'icône du programme !

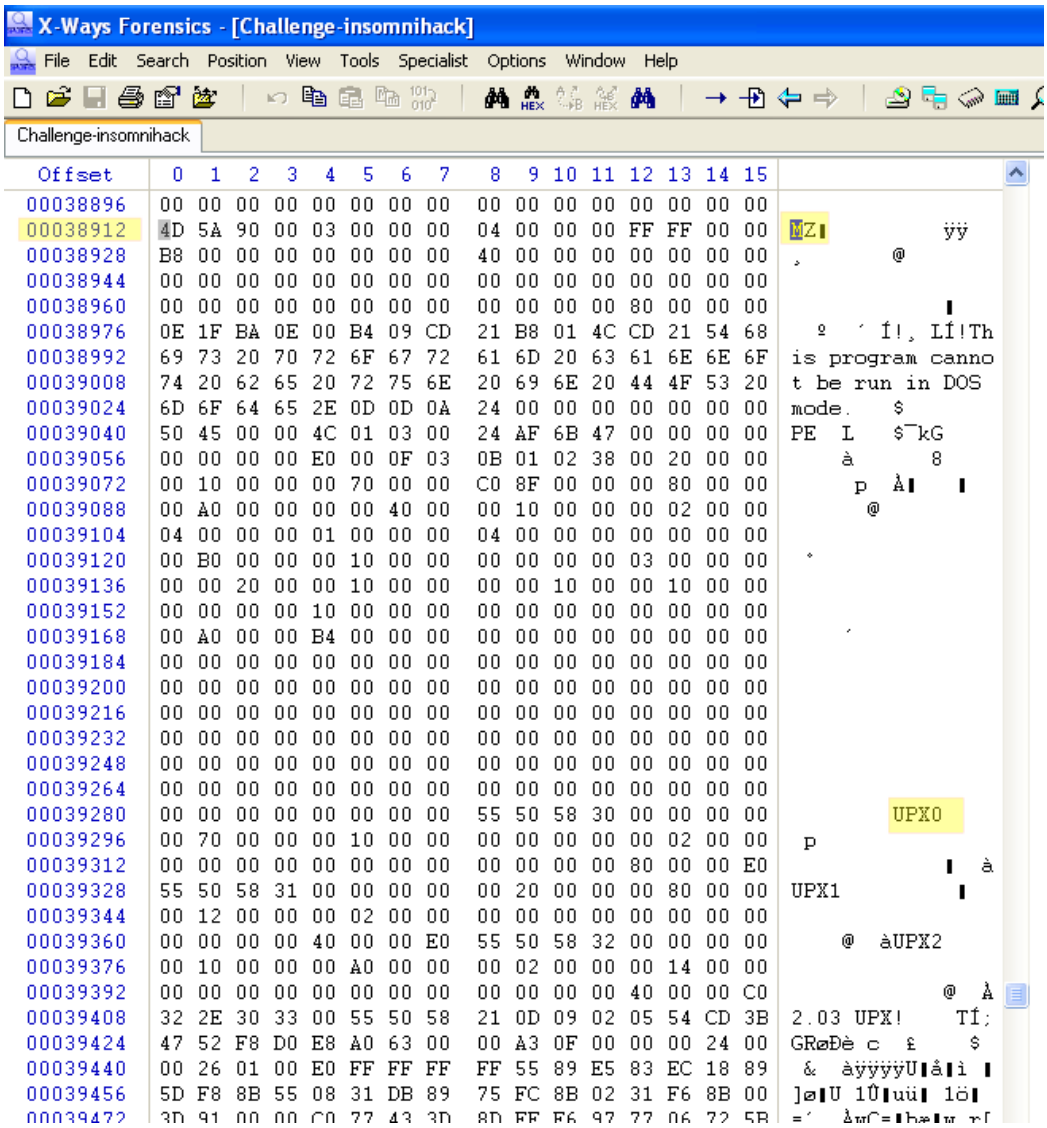
En zoomant avec la molette, on arrive mieux à lire le tout !



Le numéro de série pour cette épreuve est donc 3854-6811 !

3. Mini-épreuve : Crypto

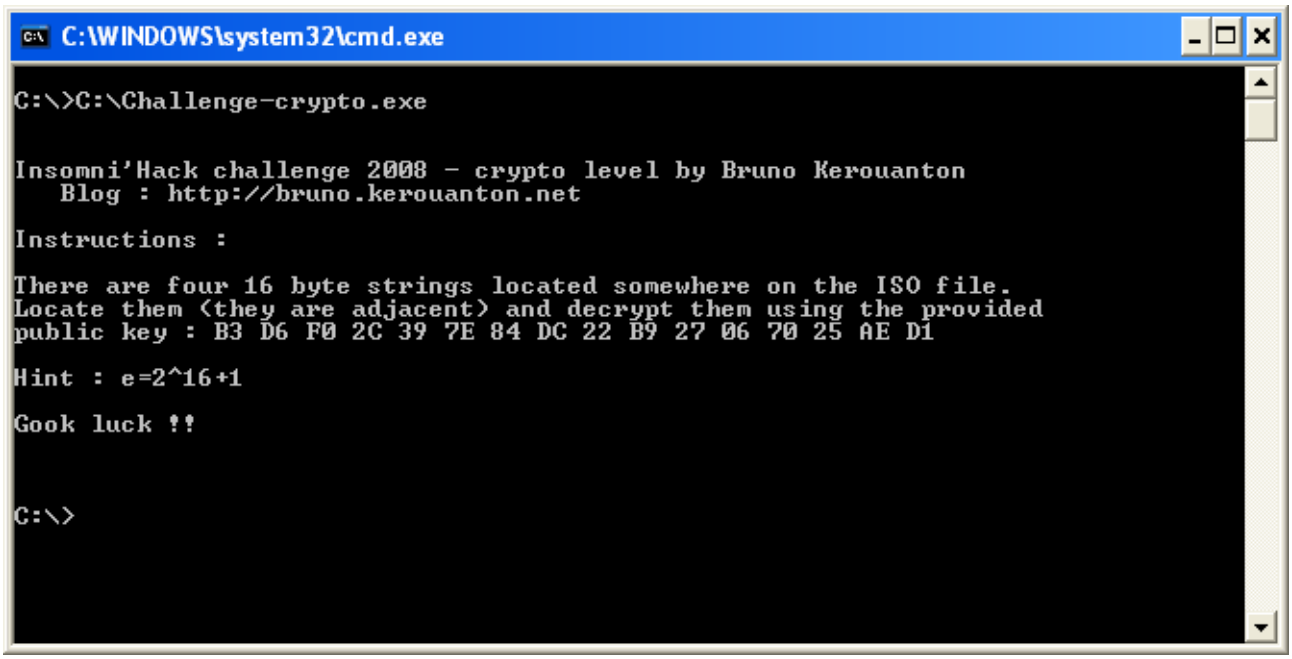
Le fichier challenge contient un second programme au format PE, situé un peu après le milieu du fichier (offset 38912). On voit qu'il est compacté au format UPX :



Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00038896	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00038912	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ▌ ÿÿ
00038928	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸ @
00038944	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00038960	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00	▌
00038976	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	º ´ Í!¸ LÍ!Th
00038992	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00039008	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00039024	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode. $
00039040	50	45	00	00	4C	01	03	00	24	AF	6B	47	00	00	00	00	PE L $¯kG
00039056	00	00	00	00	E0	00	0F	03	0B	01	02	38	00	20	00	00	à 8
00039072	00	10	00	00	00	70	00	00	C0	8F	00	00	00	80	00	00	p À▌ ▌
00039088	00	A0	00	00	00	00	40	00	00	10	00	00	00	02	00	00	@
00039104	04	00	00	00	01	00	00	00	04	00	00	00	00	00	00	00	
00039120	00	B0	00	00	00	10	00	00	00	00	00	00	03	00	00	00	°
00039136	00	00	20	00	00	10	00	00	00	00	10	00	00	10	00	00	
00039152	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	
00039168	00	A0	00	00	B4	00	00	00	00	00	00	00	00	00	00	00	´
00039184	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039216	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039232	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039248	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039264	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00039280	00	00	00	00	00	00	00	00	55	50	58	30	00	00	00	00	UPX0
00039296	00	70	00	00	00	10	00	00	00	00	00	00	00	02	00	00	p
00039312	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	E0	▌ à
00039328	55	50	58	31	00	00	00	00	00	20	00	00	00	80	00	00	UPX1 ▌
00039344	00	12	00	00	00	02	00	00	00	00	00	00	00	00	00	00	
00039360	00	00	00	00	40	00	00	E0	55	50	58	32	00	00	00	00	@ àUPX2
00039376	00	10	00	00	00	A0	00	00	00	02	00	00	00	14	00	00	
00039392	00	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0 @ À
00039408	32	2E	30	33	00	55	50	58	21	0D	09	02	05	54	CD	3B	2.03 UPX! TÍ;
00039424	47	52	F8	D0	E8	A0	63	00	00	A3	0F	00	00	00	24	00	GRøÐè c £ $
00039440	00	26	01	00	E0	FF	FF	FF	FF	55	89	E5	83	EC	18	89	& àÿÿÿÿU▌å▌ì ▌
00039456	5D	F8	8B	55	08	31	DB	89	75	FC	8B	02	31	F6	8B	00	]ø▌U 1Û▌uü▌ 1ö▌
00039472	3D	91	00	00	C0	77	43	3D	8D	FF	F6	97	77	06	72	5B	=´ ÀwC=▌þ▌w r[

On peut l'extraire une fois de plus en supprimant tout ce qui est situé au-dessus de la signature "MZ", et en sauvegardant le résultat dans un fichier avec l'extension .exe

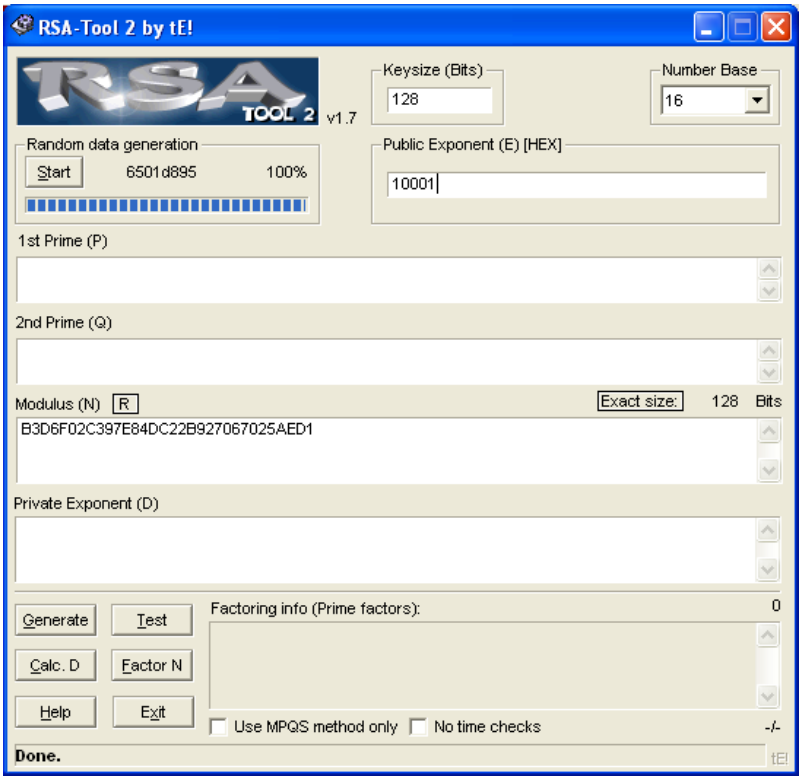
Celui-ci, une fois exécuté, donne les instructions et indices pour faire cette épreuve.



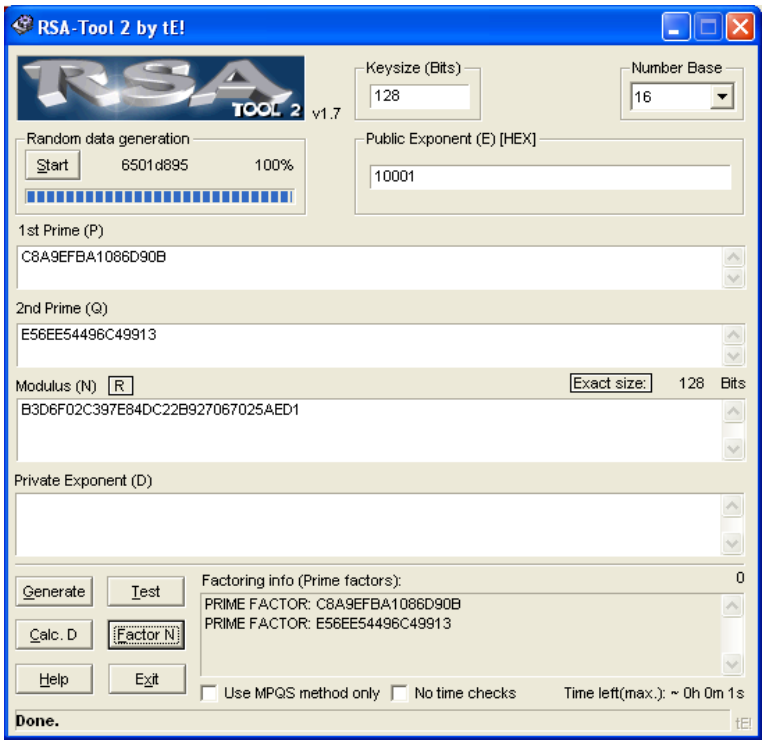
Une fois le programme lancé, on y découvre une clef publique n = 0xB3D6F02C397E84DC22B927067025AED1, ainsi qu'une information utile : e=2^16+1, soit 0x10001 en hexadécimal.

Le fait de voir 0x10001 doit titiller un peu ceux qui ont fait du RSA car c'est une valeur d'exponentiation très fréquemment employée. L'épreuve consiste donc à déchiffrer des chaînes de caractères chiffrées en RSA.

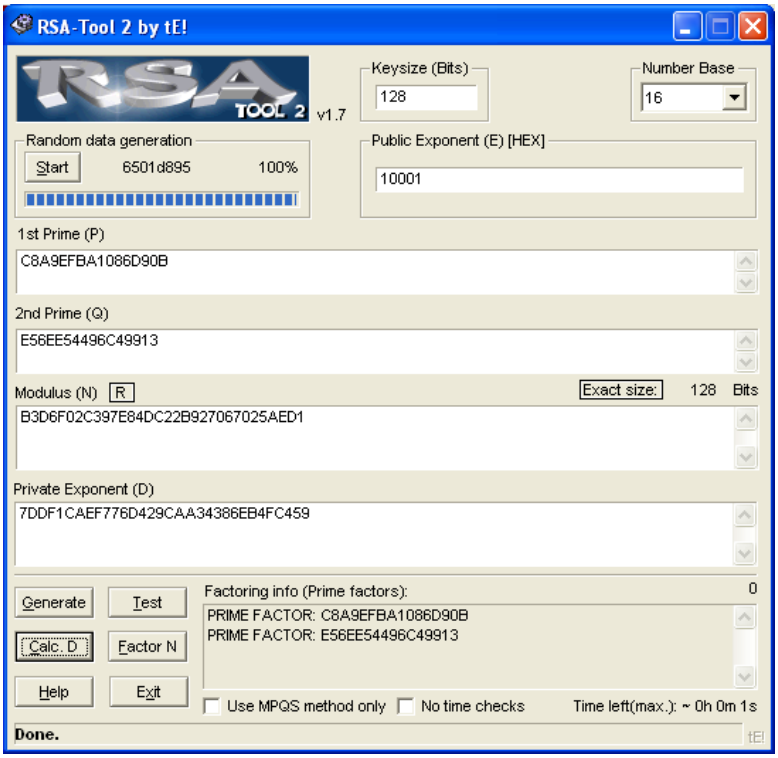
Afin de s'en assurer, lançons *RSA-Tool 2 by Te!* puis entrons les valeurs e et n dans les champs correspondants, puis vérifions la taille de la clef publique en cliquant sur "Exact size" : On est rassurés, celle-ci fait bien 128 bits, donc ce sera rapide !



Il nous faut trouver la clef privée d, qui ne figure bien entendu pas dans le fichier du challenge...
Factorisons N pour retrouver les deux facteurs premiers P et Q. Pour cela, cliquer sur "Factor N" et attendre quelques secondes :



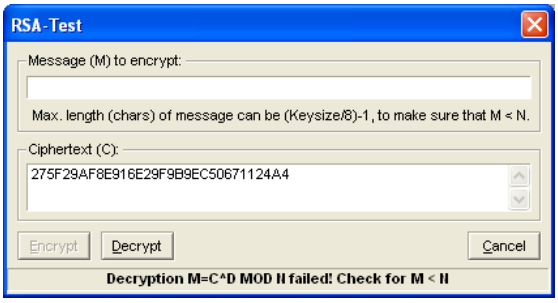
Ensuite cliquer sur "Calc. D" permet d'obtenir la fameuse clef privée d=0x7DDF1CAEF776D429CAA34386EB4FC459 avec laquelle on va pouvoir déchiffrer les messages.



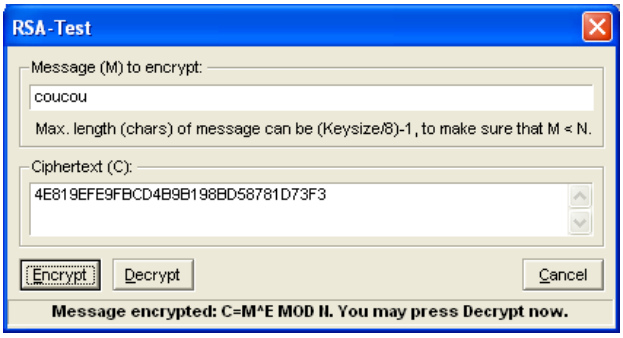
Ceux-ci sont présentés ainsi :

67F475E806CFD8710937C44330801597
78264E374191E7946DF74C8745915210
275F29AF8E916E29F9B9EC50671124A4
7501AA4B51420AD9CBC9E5D4819F4C42

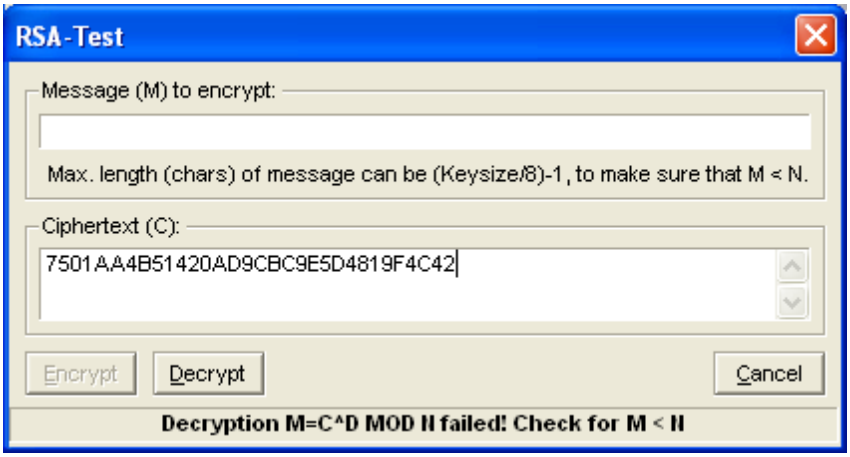
Lancer l'interface permettant de chiffrer et déchiffrer en cliquant sur "Test":

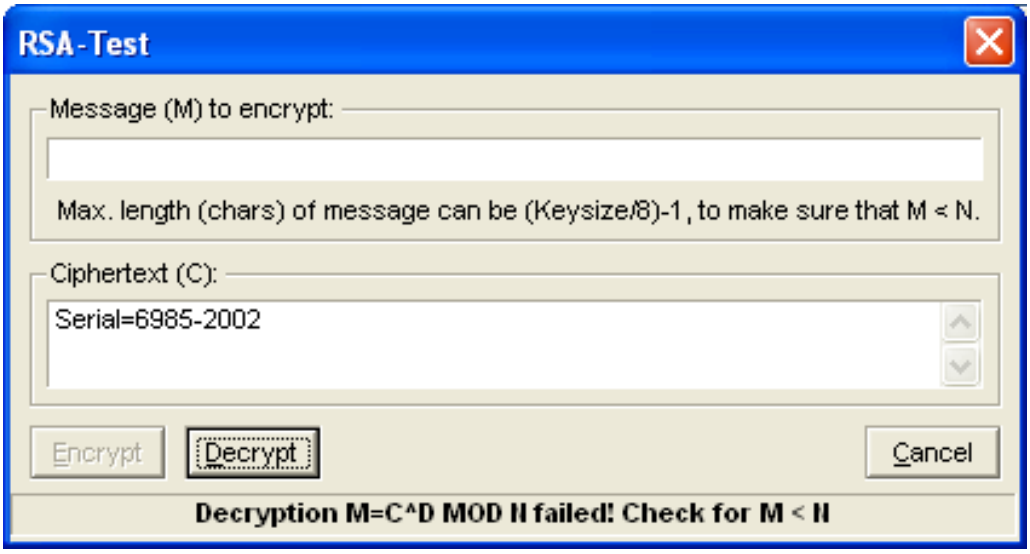


Il y a un petit bug dans l'outil puisqu'il faut faire un premier chiffrement avant de pouvoir déchiffrer. Entrer n'importe auoi dans la zone du haut, puis faire "Encrypt" :



Maintenant, le bouton "Decrypt" est dégrisé. Copier une par une dans la zone du bas les chaînes hexa trouvées dans le fichier; puis cliquer sur "Decrypt" pour obtenir chacune des phrases et le sérial demandé :





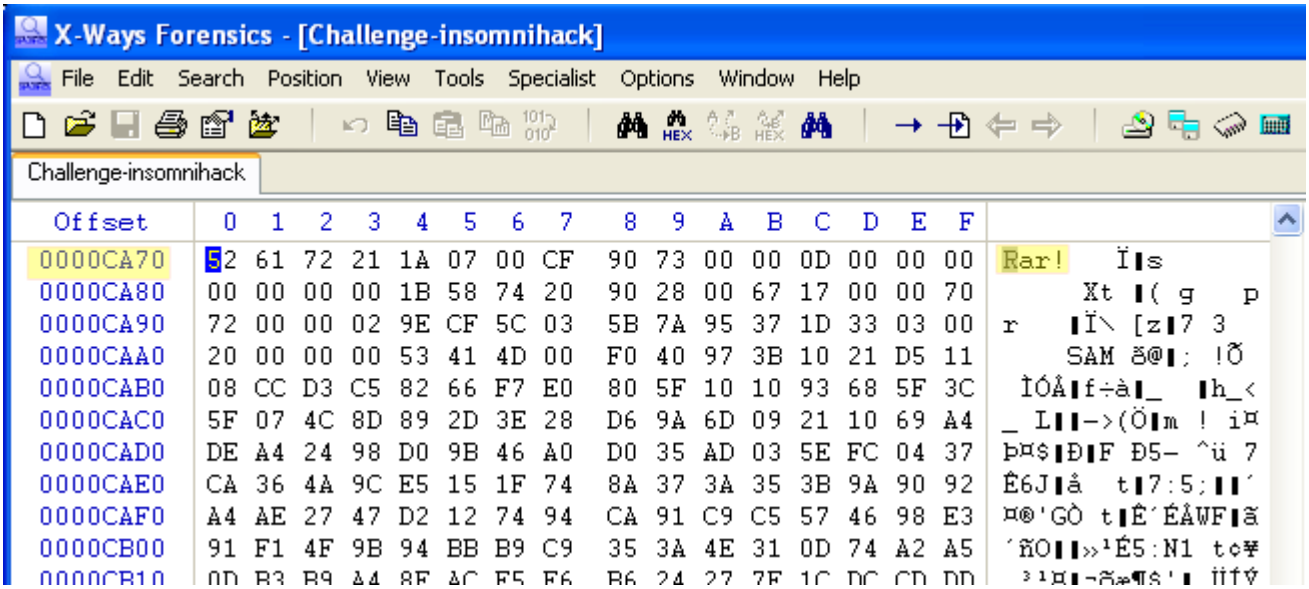
En reprenant toutes les chaînes déchiffrées, on obtient ainsi le message suivant :

Congratulations	67F475E806CFD8710937C44330801597
You found the	78264E374191E7946DF74C8745915210
InsomniHack2008	275F29AF8E916E29F9B9EC50671124A4
Serial=6985-2002	7501AA4B51420AD9CBC9E5D4819F4C42

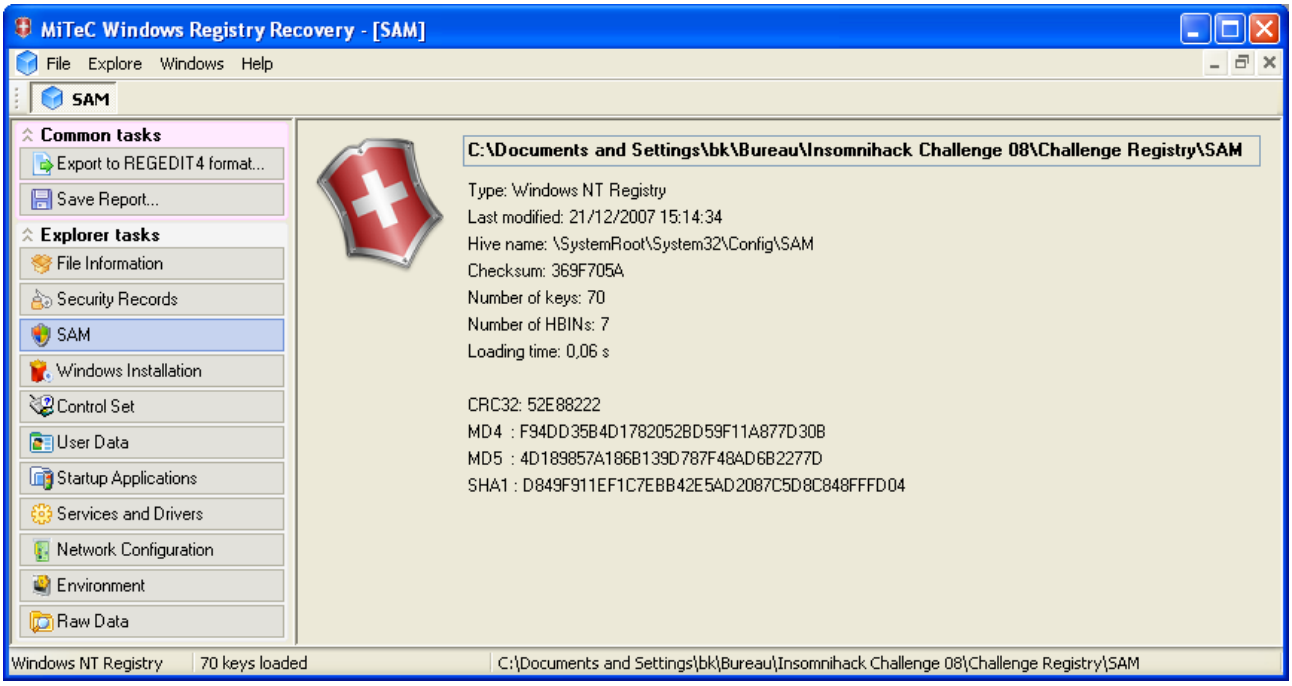
Le numéro de série à trouver était donc : 6985-2002

4. Mini-épreuve : Forensics

En recherchant les fichiers effacés à l'aide d'un outil de récupération de fichiers (ou bien manuellement), on finir par retrouver un fichier compacté au format RAR à l'offset 0CA70 :

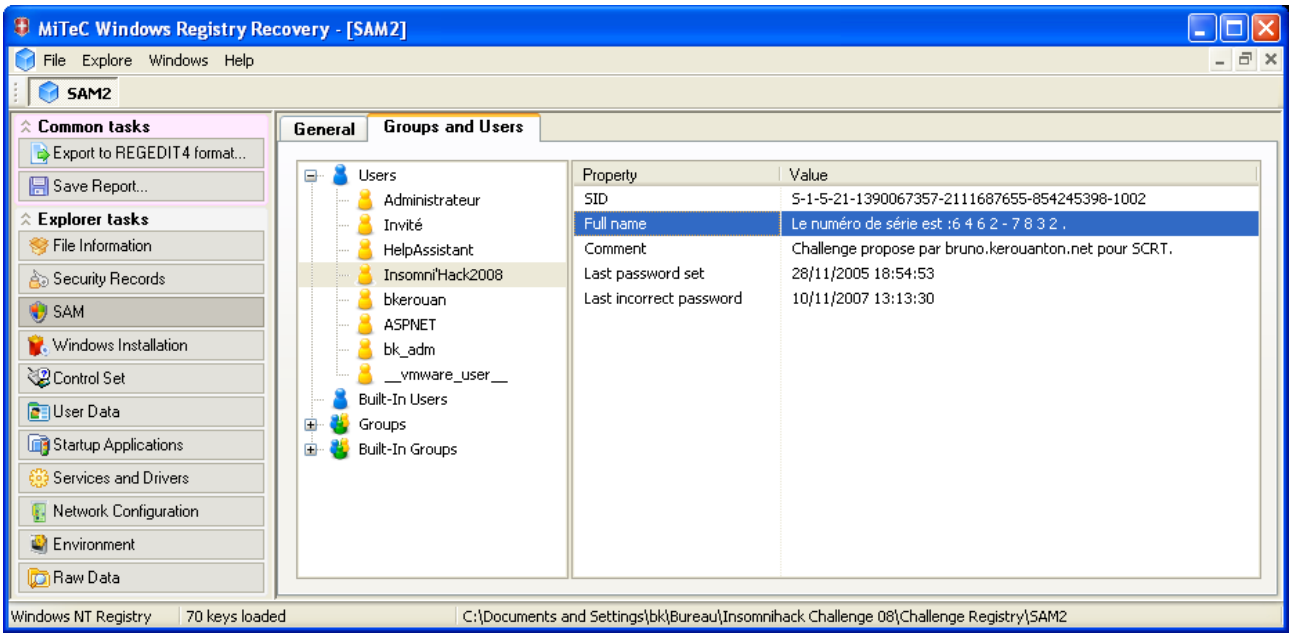


Une fois extrait, on tombe sur un fichier SAM de base de registre, que l'on pourra par exemple ouvrir avec MiTeC Windows Registry Recovery fourni dans la collection µTools :



En naviguant dans la liste des utilisateurs, on en remarque un qui a un nom éloquent :

"Insomni'Hack2008" ! En allant voir les détails de l'utilisateur, on récupère le numéro de série.

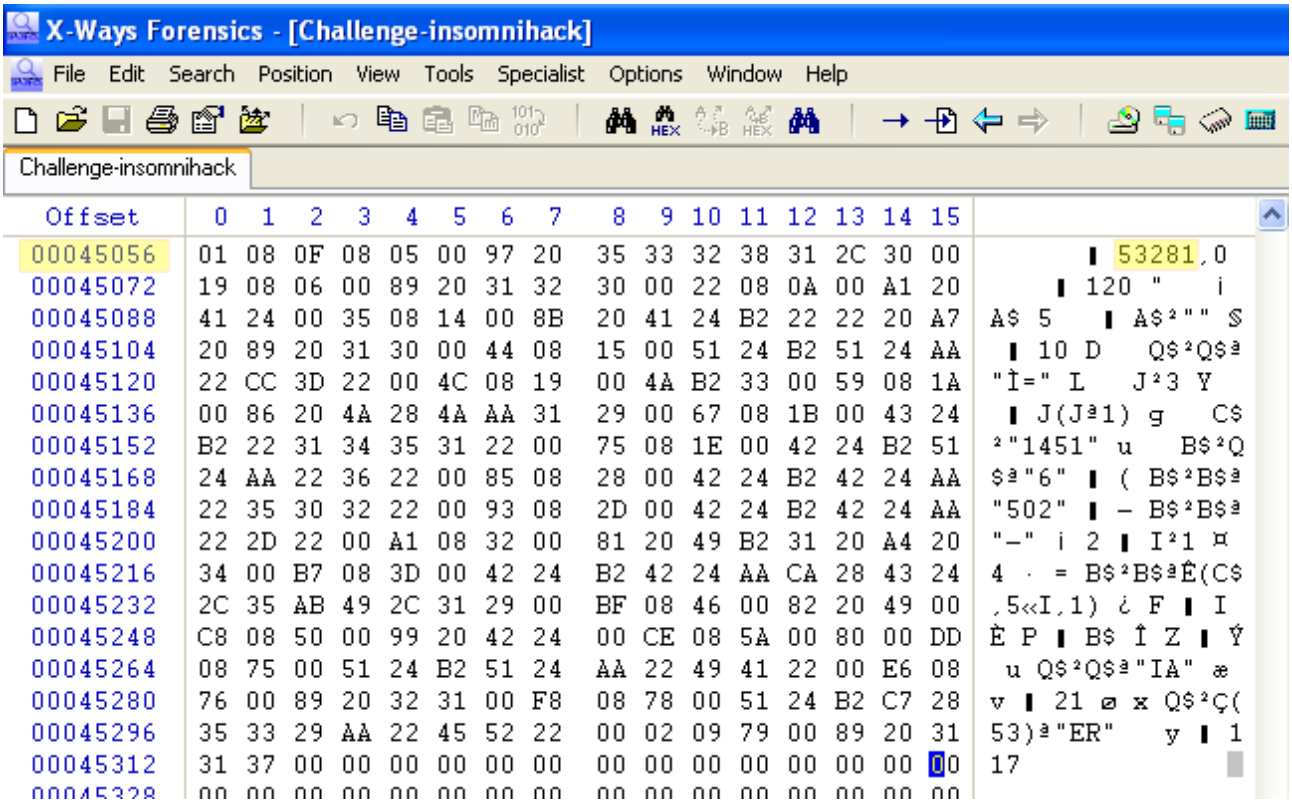


Le numéro de série pour cette épreuve est donc : 6462-7832.

5. Mini-épreuve : Oldschool

Pour commencer, il faut récupérer le fichier "1.prg" retrouvé en montant l'image ISO, ou par un autre biais (récupération de fichiers, etc...). Il se trouve à l'offset 45056.

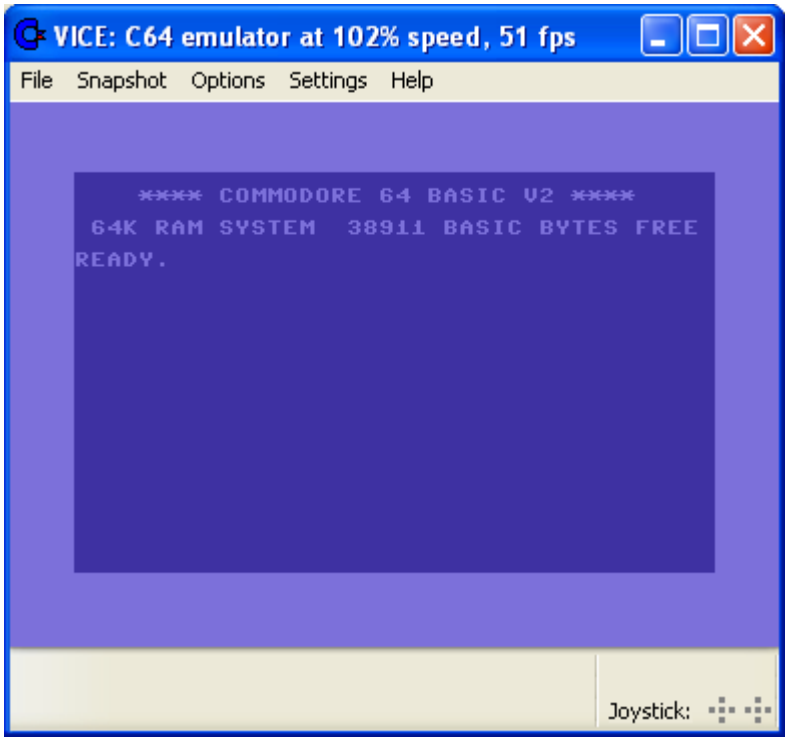
En regardant le contenu, on remarque la valeur 53281, bien connue des anciens programmeurs C64. Il s'agit de l'adresse mémoire pour le registre contrôlant la couleur de la bordure d'écran.

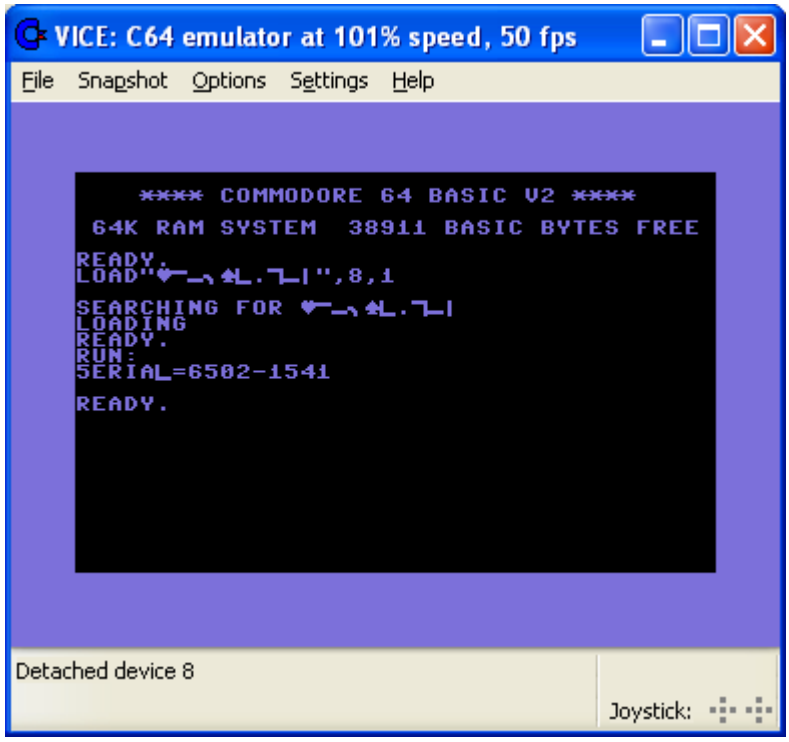


Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00045056	01	08	0F	08	05	00	97	20	35	33	32	38	31	2C	30	00
00045072	19	08	06	00	89	20	31	32	30	00	22	08	0A	00	A1	20
00045088	41	24	00	35	08	14	00	8B	20	41	24	B2	22	22	20	A7
00045104	20	89	20	31	30	00	44	08	15	00	51	24	B2	51	24	AA
00045120	22	CC	3D	22	00	4C	08	19	00	4A	B2	33	00	59	08	1A
00045136	00	86	20	4A	28	4A	AA	31	29	00	67	08	1B	00	43	24
00045152	B2	22	31	34	35	31	22	00	75	08	1E	00	42	24	B2	51
00045168	24	AA	22	36	22	00	85	08	28	00	42	24	B2	42	24	AA
00045184	22	35	30	32	22	00	93	08	2D	00	42	24	B2	42	24	AA
00045200	22	2D	22	00	A1	08	32	00	81	20	49	B2	31	20	A4	20
00045216	34	00	B7	08	3D	00	42	24	B2	42	24	AA	CA	28	43	24
00045232	2C	35	AB	49	2C	31	29	00	BF	08	46	00	82	20	49	00
00045248	C8	08	50	00	99	20	42	24	00	CE	08	5A	00	80	00	DD
00045264	08	75	00	51	24	B2	51	24	AA	22	49	41	22	00	E6	08
00045280	76	00	89	20	32	31	00	F8	08	78	00	51	24	B2	C7	28
00045296	35	33	29	AA	22	45	52	22	00	02	09	79	00	89	20	31
00045312	31	37	00	00	00	00	00	00	00	00	00	00	00	00	00	00

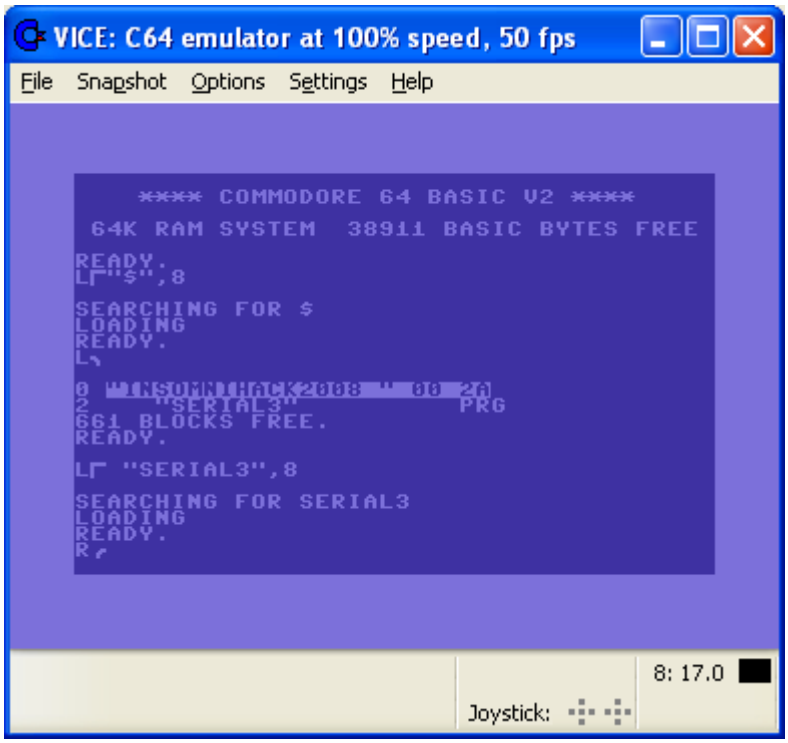
On récupère donc en fait un binaire qui n'est ni plus ni moins qu'un programme BASIC pour C64 ! Le listing est très simple (voir plus bas), mais a le mérite d'être légèrement obfusqué pour ne pas laisser le numéro de série directement accessible via une recherche texte brute, et de plus n'est pas un simple fichier texte mais un basic tokenisé ce qui rend le tout peu clair !

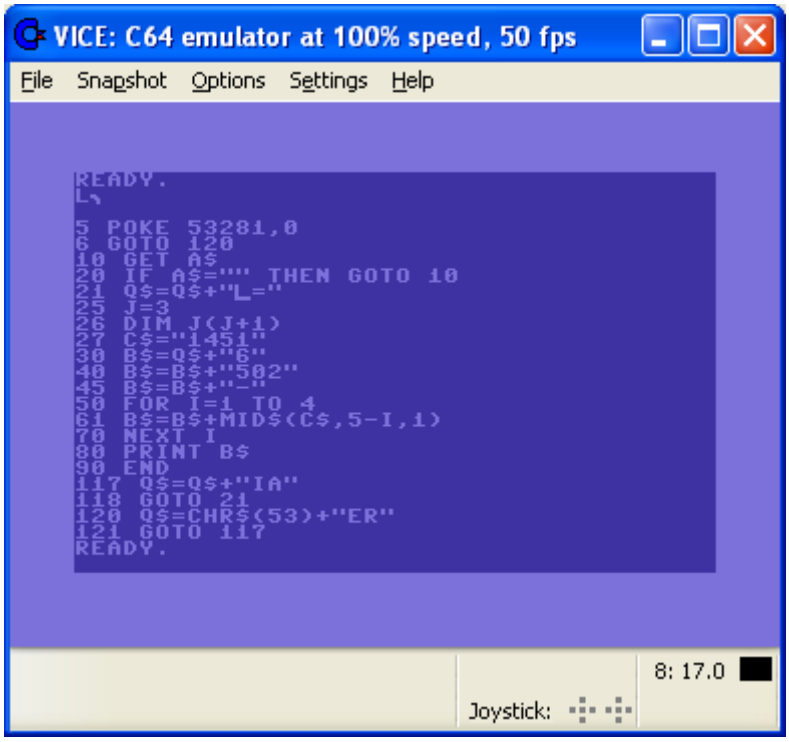
Pour obtenir le numéro de série, il faut tout simplement le lancer en drag-n-drop dans l'émulateur VICE64 (x64.exe) fourni dans la collection d'outils !





Pour information, voici le listing associé, très simple et écrit en 2mn :





Le numéro de série pour cette épreuve est donc 6502-1541[2] !

2 Et un petit bonus pour ceux et celles qui sauront m'en dire plus sur le choix de ces valeurs !!!

Remerciements

J'espère que tout cela vous a bien amusé, et je remercie les organisateurs de la nuit Insomni'Hack et plus spécialement Paul Such, fondateur de la société SCRT sans qui InsomniHack n'existerait pas.

Je remercie également les participants du Challenge pour leur motivation, ainsi que les personnes qui ont lu ce document jusqu'au bout pour leur intérêt !

Merci au passage pour les auteurs de quelques bons logiciels qui m'ont permis de mettre sur pied cette épreuve :

- Stefan Fleischmann : X-Ways Forensics / WinHex, très bons éditeurs hexa.
- L'équipe de Vice-C64, une suite d'émulateurs Commodure OpenSource très sympa.
- Jan Fiala : PsPad, un éditeur texte super
- Mark Russinovich (SysInternals), Nir Sofer (Nirsoft), MiTec, Piriform, et tous les autres développeurs de petits outils géniaux qui font désormais partie de ma collection µTools dont je ne me sépare plus !
- Et une mention spéciale à Ifak Guilfanov pour IDA Pro et son décompilateur Hex-Rays, et la FRET (French Reverse Engineering Team) même si je n'ai tout compte fait pas pu intégrer d'épreuve de reverse...

Je remercie enfin les enseignants du Mastère SSI Supélec / ENST-Bretagne qui m'ont supporté pendant un an en 2006, car ils m'ont permis durant cette période d'apprendre ou de revoir plein de bonnes choses notamment en cryptographie et en programmation.

A bientôt,

Bruno Kerouanton

CISSP, Mastère SSI Supélec/ENST-B, 27001 Lead Auditor

Lisez, réagissez sur ***http://bruno.kerouanton.net/blog***