Pourquoi la sécurité est un échec (et comment y remédier)

Nicolas Ruff
nicolas.ruff(@)eads.net

EADS Innovation Works

« The whole IT security industry is an accident - an artifact of how the computer industry developed. » – Bruce Schneier [1,2]

1 Introduction

La sécurité « informatique » est un domaine qui remonte quasiment à l'invention du transistor, si l'on inclut dans ce terme la sécurité de tout système automatisé, comme les centraux téléphoniques. Mais depuis au moins 10 ans, l'apparition d'Internet a provoqué une croissance exponentielle de ce secteur auparavant chasse gardée d'une élite souvent souterraine. Avec l'entrée des marchands dans le temple, la sécurité se résume aujourd'hui le plus souvent à l'acquisition de produits, voire de certificat(ion)s. Alors que les produits de *Data Loss Prevention* (DLP) semblent faire le *buzz* en 2009, les plus anciens se souviendront avec amusement de toutes les innovations qui étaient censées apporter la *sécurité par la technologie*. Parmi les stars déchues des conférences et des salons, devenues aujourd'hui *so last year*, on peut citer : le contrôle d'intégrité des fichiers, les *honeypots*, les IDS et toutes leurs déclinaisons, voire les antivirus et les *firewalls*... Les tentatives pour imposer la *sécurité par le papier* semblent également vouées à l'échec : le standard PCI-DSS n'empêche toujours pas les numéros de carte bleue de disparaître par milliers [3]. Enfin la *sécurité par l'éducation* en est toujours au point mort. Il est même raisonnable de considérer qu'elle a régressé avec l'irruption du « Web 2.0 », particulièrement des réseaux sociaux. Ce sombre tableau étant dressé, il sera difficile d'adresser tous les axes d'amélioration potentiels en seulement quelques dizaines de pages. C'est pourquoi cet article tentera de répondre à une question apparemment simple : « pourquoi votre réseau interne est toujours vulnérable en 2009, malgré tout l'argent investi dans la sécurité ». Et par corollaire : « comment mieux dépenser cet argent pour rendre le réseau plus sûr ». Ne pouvant pas non plus couvrir tous les domaines de l'informatique, cette présentation filera l'exemple d'un réseau bureautique exploité sous Microsoft Windows (exemple susceptible d'intéresser le public le plus large). Toutefois les problèmes et les méthodes de résolution évoqués

restent largement applicables à d'autres domaines, tels que la sécurité des applications Web, etc.

2 Les péchés capitaux de la sécurité

2.1 Le biais de perception

Il existe un biais majeur de perception dans le domaine de la sécurité, à savoir que les seules menaces dangereuses sont celles « dont on parle », ce qui occulte la majorité des menaces effectives comme :

1. Les attaques qui ne sont pas massivement exploitées « dans la nature ». Malheureusement, la majorité des failles Office découvertes ces dernières années ont d'abord été utilisées dans des attaques très ciblées.
2. Les attaques pour lesquelles aucun produit de protection n'existe - ce qui ne veut pas dire qu'il est impossible de s'en protéger, mais que personne ne gagne assez d'argent à préconiser l'activation d'une « simple » option de sécurité.
3. Les attaques trop anciennes pour intéresser les journalistes et les organisateurs de conférences. Typiquement, l'attaque en relais SMB (Credential Reflection) qui date d'avant les années 2000 (mais reste toujours très efficace).
4. Les attaques trop compliquées, qui ne trouvent pas d'écho dans la presse, même dite « informatique ».
5. Paradoxalement, les attaques trop simples n'intéressent également pas grand monde, alors que ce sont parfois les plus dangereuses ! Dans cette catégorie, on peut citer le « Cross-Site Request Forgery » (CSRF ou XSRF).

Je qualifierai par la suite toutes ces attaques de « menace invisible ».

See no evil Le premier biais peut être entendu sous cette forme : « cette faille n'est pas critique, car il n'existe pas de code d'exploitation sur Milw0rm, PacketStorm ou dans Metasploit ». Tous ceux qui ont un peu trop tardé à installer le correctif MS08-067 [4] ont probablement compris (à leurs dépends) qu'une faille réellement exploitable (Exploitability Index à 1 selon Microsoft) est toujours exploitée, même si le code n'est pas rendu public. C'est le cas par exemple pour la faille MS05-043 [5], seulement disponible dans le produit Immunity CANVAS, même 4 ans après sa sortie. D'autre part, les personnes qui publient des codes d'exploitation à titre gracieux semblent rarement concernées par les problématiques d'entreprise – c'est pourquoi on trouve peu de codes pour des failles RPC nécessitant une authentification par exemple. Pourtant ces failles sont redoutables au sein d'un domaine Windows, et les

correctifs publiés pour ces failles devraient se voir assignés une priorité de déploiement très élevée.

Do no evil Le deuxième biais s'exprime de la façon suivante : « je suis protégé contre l'écoute du trafic, car j'ai activé l'option anti-empoisonnement de cache ARP dans mon routeur ». Malheureusement l'empoisonnement de cache ARP, c'est très « 90's » comme méthode de redirection de trafic. Pour adresser le problème dans sa globalité, il faudrait également prendre en compte l'intégrité du fichier HOSTS, la réponse aux messages ICMP « Redirect », les faux serveurs DHCP, la modification de la configuration DNS par un malware, l'enregistrement du nom WPAD, l'empoisonnement de nom NetBIOS, les mises à jour DNS non sécurisées... c'est rapidement l'escalade.

2.2 La qualité de l'information disponible

Le biais précédent est alimenté par « l'industrie de la sécurité », qui regroupe pêle-mêle (à mon sens) des gens d'horizons aussi divers que les consultants, les auditeurs (liés à une norme ou une certification), les vendeurs de produits, les orateurs des conférences (et le « star system » afférent), les journalistes et les « blogueurs », ainsi que les « chercheurs ». Bien sûr, il y a des « bons » et des « mauvais » dans chacune de ces catégories. Mais plusieurs années d'expérience m'ont laissé la très nette impression que la seconde catégorie est très fortement représentée – principalement à cause du fait qu'il n'existe aucun diplôme ni aucune certification qui permette de s'assurer qu'un individu est bien ce qu'il prétend être. Le système est essentiellement basé sur la réputation et la reconnaissance par les pairs. Mais comme tous les systèmes à base de réputation (ex. Digg [6]), la quantité ne fait pas la qualité. C'est malheureusement un travers qu'on retrouve également dans la presse spécialisée, avec des arguments tels que « 80% des entreprises utilisent le produit A », sous-entendant que ce produit est un bon choix même s'il n'est pas bon – car il vaut mieux avoir tort avec les autres que raison tout seul. Des initiatives de certification de personnes existent. CISSP ou ISO 27000 sont les plus connues. Mais ces initiatives me semblent plus proches d'une activité commerciale que d'une réelle sélection des compétences. Le CISSP, c'est comme le bac – à force d'essayer on finit toujours par l'avoir !

Tout conseil n'est pas bon à prendre « L'explosion » du marché de la sécurité se traduit dans les faits par une prolifération de produits et de services. Malgré des métiers très différents, on constate que les consultants en sécurité, les auditeurs et les vendeurs de produits poursuivent des buts assez similaires, car ils tirent tous les 3 des revenus du consensus qu'ils ont contribué à installer sur le sujet de la sécurité.

Quant aux soi-disant « chercheurs », ils sont bien souvent à mettre dans le même sac, la quasi-totalité d'entre eux travaillant pour des sociétés de conseil, d'audit ou de conception de produits ...

La vérité révélée Il est amusant de constater combien les 3 catégories susnommées s'illustrent par un discours péremptoire, à la limite de la révélation mystique. Le cas des vendeurs de produits est particulièrement caricatural, avec les fameux « notre produit est sûr » ou « nous utilisons AES-256 ». Les consultants utilisent plus volontiers « les Chinois ont l'œil sur votre réseau ». Quant aux chercheurs, c'est plutôt « mon attaque est la plus dangereuse de toutes ». Ce discours est malheureusement encouragé par une baisse générale du niveau de compréhension des clients, sur laquelle je reviendrai plus loin. On rencontre souvent des consultants trop techniques, qui connaissent et utilisent (parfois avec succès) des recettes et/ou des outils. Untel ne jure que par l'empoisonnement de cache ARP, tandis qu'un autre commence par envoyer des chevaux de Troie par email à tous les employés de la société. En résumé, « pour qui a un marteau, tout problème ressemble à un clou ». A l'inverse, on rencontre également des consultants trop théoriques, qui modélisent et quantifient à outrance. Le problème de l'évaluation des risques et des impacts est un grand classique, mais n'a jamais été résolu avec courage, les différents « coefficients » étant souvent assignés au hasard (par manque d'expérience technique ou par impossibilité d'évaluation objective). Les risques les plus difficiles à traiter sont souvent minorés (consciemment ou inconsciemment) ou acceptés sans aucune justification. Mais ces deux familles de praticiens sont le plus souvent victimes du biais de perception initial, et n'apportent jamais de réponse aux attaques « invisibles », les premiers parce qu'ils ne savent pas les mettre en œuvre, les deuxièmes parce qu'elles sortent du modèle.

Internet n'oublie rien L'essentiel de l'information disponible sur Internet est parfois simplement périmée, mais le plus souvent fausse, voire limite dangereuse d'utilisation, et ce pour plusieurs raisons : Internet n'oublie rien, mais l'information se périme très vite – particulièrement dans les domaines des techniques d'attaque ou de la cryptographie. Il suffit de relire « Hacking Exposed », première édition pour s'en rendre compte. L'autre exemple est celui de l'article « Smashing The Stack For Fun And For Profit » publié dans Phrack numéro 49 [7]. On voit encore passer des questions sur cet article dans les forums, alors que les exemples de code donnés ne fonctionnent plus depuis longtemps. Même Red Hat 9.0 implémentait une randomisation (minimale) de la pile ! La majorité du contenu indexé dans les moteurs de recherche provient de blogs et de forums. Or le niveau est en moyenne faible, voire très faible - il suffit d'avoir fréquenté des forums de support utilisateur pour

s'en convaincre. Chacun obtient sa minute de gloire en publiant une réponse ou un billet – peu importe que l'information soit vraie ou fausse du moment qu'elle écrase l'interlocuteur par un semblant de technicité. Ce qui contribue ensuite à propager des contre-vérités ... et à noyer le peu d'information pertinente disponible par ailleurs. Un point particulièrement sensible concerne les exemples de code disponibles, qui sont notoirement bogués mais malgré tout copiés/collés à outrance. Ceci est applicable à tous les forums de support pour développeurs débutants et auto-formés, comme les forums dédiés au langage PHP[1]. A titre anecdotique, on pourra noter également que les implémentations de référence pour le futur algorithme SHA-3 étaient également boguées[8]...

Toute recherche donne un résultat C'est un biais connu dans la recherche scientifique : toute personne qui s'investit dans un sujet de recherche finit par trouver un résultat (quel qu'il soit) afin de justifier l'argent dépensé. Et va ensuite générer de nombreuses études corollaires, qui finissent par créer un nouveau « domaine » de recherche à part entière, drainant toujours plus de financement. Sans pour autant approcher la vérité de quelque manière que ce soit (cf. [9]) ! La recherche en sécurité informatique n'échappe pas à ce biais, surtout que les conférences de « hackers » jouent beaucoup plus sur le « star system » que les conférences de l'IEEE ... Les « chercheurs » vont donc artificiellement amplifier l'impact de leurs travaux, et les rejouer à outrance pour en augmenter la visibilité. Ce qui a fini par alimenter cette image de « Security Circus » [10] que donne notre industrie. Parmi les exemples récents, on peut citer l'intervention de Kris Kaspersky à HITB 2008 sur l'exploitabilité des bogues CPU (qui n'en a démontré aucun), ou l'intervention de Shawn Embleton & Sherri Sparks à BlackHat 2008 sur les rootkits SMM qui présentait le défaut... de n'avoir aucune solution pour injecter du code en mode SMM ! (Ce dernier point ayant été résolu par Loïc Duflot postérieurement à leur présentation, heureusement pour eux). Pour résumer, tant que l'attaque est entre les mains des « chercheurs », et la défense entre les mains de vendeurs de produits, le niveau n'est pas prêt de s'améliorer ...

2.3 Le niveau général

Face aux risques de désinformation et de mauvaise perception des risques évoqués précédemment, force est de constater que le niveau moyen ne s'améliore pas de l'autre côté. Peu de gens sont aujourd'hui capables, ou disposent du temps nécessaire, pour

[1] Ceci ne fait toutefois qu'ajouter à la longue liste de mes griefs contre ce langage, qui fût la pire régression de ces 10 dernières années pour la sécurité d'Internet...

qualifier la pertinence d'un risque ou d'une information. Cet état de fait ne résulte certainement pas d'un seul facteur, mais plutôt d'un ensemble de forces contraires : la financiarisation des activités industrielles, la dévaluation des connaissances techniques, la dilution des responsabilités, voire la corruption... Il est triste de constater que dans les (nombreuses) filières de formation à la sécurité qui se sont créées ces dernières années, beaucoup d'élèves souhaitent directement passer par la case « RSSI » pour disposer des avantages matériels de la fonction, sans avoir aucune expérience ni connaissance dans le domaine de l'informatique. Et les entreprises, parfois légalement tenues de disposer d'un responsable sécurité (cf. dispositions Bâle 2 ou CNIL), y trouvent des éléments malléables et peu coûteux, qui acceptent d'endosser les risques sans disposer d'aucun moyen d'action.

La conspiration des mauvais Un fait se dessine dans les lignes précédentes : l'essentiel des contributeurs au domaine de la sécurité informatique sont mauvais ou ne savent pas de quoi ils parlent. Des théories économiques prétendent que les mauvais finissent par disparaître, car le marché sait faire la différence. Mais en pratique, leur prolifération semble au contraire rampante dans l'industrie (ce qui confirme que les théories économiques sont elles-mêmes affectées par le biais de la recherche, et majoritairement fumeuses). Par le jeu de la cooptation (je t'invite dans ma conférence et je parle dans la tienne, tu me cites dans ton papier et je t'écris une bonne revue), de l'effet bisounours, ou de tout autre nom qu'on donnera à cet instinct de survie, les mauvais finissent par tisser des réseaux beaucoup plus solides et plus étendus, qui leur évitent de courir le risque d'être démasqués un jour.

La course en avant Enfin, le domaine de la sécurité (et des TIC) en général est très clairement un marché de renouvellement dans lequel on capitalise peu. Chaque nouvelle version ou technologie reproduit les erreurs du passé, parfois les aggrave. Quelques buzzwords comme « Web 2.0 » ou « Cloud Computing » devraient immédiatement faire jaillir des images précises dans la tête du lecteur. Vous étiez satisfaits de Windows XP ? Vous aviez des certificats signés par MD5 ? Et bien dansez maintenant ! Cet état d'esprit atteint son paroxysme aujourd'hui, de nombreux logiciels ne fonctionnant plus qu'en mode « connecté à Internet », et sautant allègrement de version majeure d'une année sur l'autre. Mais comment gérer un antivirus pour lequel plusieurs dizaines de mises à jour sont publiées chaque jour (avec le risque de faux positif que cela comporte) ? Comment faire confiance à un HIPS censé détecter les attaques « 0day », s'il est doit être mis à jour régulièrement pour cela ? Comment gérer les correctifs de sécurité qui ne sont disponibles que pour une version majeure supérieure d'un logiciel (ex. Acrobat Reader) ?Comment gérer les mises à jour de sécurité de plusieurs

dizaines de méga-octets, qui nécessitent les droits administrateur sur le poste local (ex. QuickTime, Java) ? Si le modèle de la mise à jour automatique améliore sensiblement le niveau de sécurité des réseaux non administrés, où tout le monde est administrateur (c'est-à-dire essentiellement les réseaux de particuliers ou de professions libérales), ce nouveau modèle de support n'aide pas les entreprises à maitriser leur niveau de sécurité. Sans parler des systèmes industriels qui intègrent du logiciel « COTS[2] », tels que panneaux d'affichage, vitrines interactives, équipements de contrôle d'accès, etc. Et il n'existe plus aujourd'hui de réseau déconnecté des menaces provenant d'Internet, comme l'a démontré l'épisode du ver « Conficker » (cf. [11] aux USA ou [12] en France). Le problème de fond, c'est que l'innovation provient essentiellement du marché « grand public » et contraint les entreprises à adopter à marche forcée des modèles logiciels complètement orthogonaux aux règles de sécurité les plus élémentaires... et parfois les plus règlementaires !

3 Exemple : qu'est-ce qu'un bon mot de passe ?

3.1 Contexte

Le choix d'un bon mot de passe fait partie des sujets les plus discutés dans le domaine de la sécurité. Voici un florilège de ce qu'on peut lire ou entendre sur le sujet :

- « Les puissances de calcul augmentent, il faut passer les longueurs minimales de 7 à 10 caractères. »
- « Il faut changer son mot de passe tous les 90 jours. »
- « Un bon mot de passe mélange majuscules, minuscules et caractères spéciaux. »
- « Il faut générer ses mots de passe aléatoirement avec un outil de gestion adapté. »

On comprend bien d'où viennent ces principes, puisqu'ils sont pour la plupart applicables par des options de l'interface Windows (principe de disponibilité de l'option, en application du biais de perception vu précédemment) ou par des outils sur étagère. Mais dans le cas pratique des réseaux Windows, ces principes sont globalement faux. Pour bien appréhender la nature d'un bon mot de passe, il est nécessaire d'embrasser sa dérivation, son stockage, et son utilisation du mot de passe. Les détails ayant été présentés par Aurélien Bordes lors de SSTIC 2007 [13], la description qui en est faite ci-dessous ne donnera que les grandes lignes.

[2] Commercial Off The Shelf

3.2 Dérivation

Les systèmes modernes (ce qui remonte au moins aux années 90) ne stockent pas les mots de passe en clair. Ils utilisent des algorithmes de dérivation à sens unique permettant de générer un hash normalement très coûteux à inverser. En environnement Windows, si la méthode de condensation « LM » est toujours active (c'est le cas par défaut avant Vista), les mots de passe sont mis en majuscules, puis séparés en 2 blocs de 7 caractères servant de clé DES pour chiffrer une chaine fixe. En conséquence, la méthode « LM » ne supporte pas les mots de passe de plus de 14 caractères. La méthode de condensation « NTLM » est quant à elle un simple algorithme MD4 appliqué au mot de passe en version Unicode, limité à 255 caractères. Il n'existe pas de méthode spécifique à Kerberos, le hash NTLM étant réutilisé.

3.3 Stockage

Les hash sont stockés dans la base d'authentification (base SAM pour un compte local, annuaire Active Directory pour un compte de domaine). On notera que dans les deux cas, il n'existe pas de « sel ». Un mot de passe donné génère toujours la même paire de hash, ce qui permet la génération de tables pré-calculées (dites « Rainbow Tables »). Windows propose également une méthode de stockage dite « réversible », non utilisée par défaut, qui consiste à stocker le mot de passe chiffré par une clé symétrique (donc trivialement inversible). Bien qu'il n'existe pas de « preuve de concept » publique permettant de retrouver le mot de passe en clair à partir d'une base de comptes, cette dernière option est évidemment extrêmement dangereuse à activer. Elle est toutefois nécessaire si un serveur Windows doit supporter des protocoles d'authentification non LM/NTLM (ex. authentification HTTP « Digest »).

3.4 Utilisation

Lorsqu'un client souhaite s'authentifier auprès d'une ressource réseau, il négocie avec le serveur un protocole d'authentification. Le serveur peut lui demander d'envoyer son mot de passe en clair (protocole utilisé par « Windows for Workgroup » et « Samba 1.0 »), mais tous les clients Windows refusent depuis longtemps - dans la configuration par défaut - de répondre à une telle demande. Les autres protocoles possibles sont LM, NTLM, NTLM2 ou NTLMv2. Tous ces protocoles sont basés sur un défi/réponse : le client prouve qu'il connait le mot de passe en chiffrant un élément aléatoire envoyé par le serveur (défi). La clé de chiffrement utilisée est soit le hash LM, soit le hash NTLM, soit les deux (en fonction du protocole négocié). Le serveur peut vérifier la réponse retournée, connaissant exclusivement ces hash. On notera qu'il n'est pas

possible d'utiliser exclusivement le protocole Kerberos sur un réseau Windows, au moins l'une des méthodes précédentes doit rester disponible. Le cas classique de « non utilisation » de Kerberos concerne l'accès à des ressources par adresse IP, pour lesquelles le « Principal Name » (nécessaire à l'obtention d'un ticket de service) ne peut pas être déterminé. Par ailleurs Kerberos peut aussi faire l'objet d'attaques, telles que l'usurpation du KDC, qui ont été décrites par ailleurs [14] mais pour lesquelles il n'existe pas d'implémentation « sur étagère ».

3.5 Attaques possibles

Une fois ces préliminaires techniques maitrisés, voici les véritables règles qui régissent la sécurité d'un mot de passe et de l'authentification Windows en général :

1. La compromission du hash LM implique la compromission du mot de passe en quelques minutes, par le biais de tables pré-calculées. Ceci est totalement indépendant de la longueur du mot de passe et du jeu de caractères utilisé (sauf quelques caractères particuliers qui empêchent la génération du hash LM). On notera toutefois que le hash LM n'est généré que pour des mots de passe de moins de 15 caractères.
2. Les hash LM et NTLM peuvent être compromis dans la base de comptes (fichier SAM pour les comptes locaux, annuaire Active Directory pour les comptes de domaine). La compromission de la base de comptes en « live » nécessite de disposer d'un compte administrateur (administrateur local dans le premier cas, administrateur de domaine dans le deuxième cas). Mais il est également possible de compromettre ces fichiers par un accès physique « offline » au disque dur, ou depuis une sauvegarde.
3. Les hash LM et NTLM peuvent également être compromis dans la session d'un utilisateur par toute application qui s'y exécute, et ce quels que soient les droits de l'utilisateur sur son poste. En effet Windows dispose d'une fonction de « SSO » automatique dans le domaine, qui nécessite que les hash LM et NTLM soient conservés en mémoire pendant toute la durée de la session. Ceci inclut également les hash des comptes qui ont été utilisés pour monter des lecteurs réseau, si ces montages s'effectuent sous des comptes différents. Enfin par extension, la compromission d'un serveur de type « Terminal Server » ou « Citrix » permet de récupérer les hash de toutes les sessions utilisateur actives, si l'attaquant dispose d'un compte administrateur sur ce serveur.
4. Les hash LM et NTLM sont les « secrets » qui permettent d'authentifier l'utilisateur. Ce qui signifie que le mot de passe en clair ne sert à rien, sauf lorsqu'il est

demandé explicitement par le protocole utilisé (ex. connexion RDP). Par corollaire, quelle que soit la complexité du mot de passe, la compromission du hash NTLM d'un utilisateur permet d'effectuer toutes les opérations que l'utilisateur peut normalement effectuer (accès aux partages réseau, exécution d'outils d'administration, etc.).

5. Il n'est pas possible d'utiliser exclusivement le protocole Kerberos sur un réseau Windows. Mais même si c'était le cas, la réutilisation du hash NTLM expose le protocole Kerberos aux mêmes attaques (sous réserve de disposer des outils ad-hoc).
6. Si une séquence de défi /réponse est capturée sur le réseau, il est aussi difficile de remonter aux hash LM/NTLM utilisés que de remonter au mot de passe utilisé.

L'écoute passive du réseau est donc le seul scénario où la désactivation du protocole LM et le choix d'un mot de passe complexe vont rendre le travail de l'attaquant plus difficile.

3.6 Remarque sur les comptes de domaine

On notera qu'un attaquant ayant accès physique à un contrôleur de domaine sans forcément être administrateur de domaine (ex. sites distants dans une grosse société, disposant chacun d'un contrôleur de domaine local en réplication) peut théoriquement créer un nouveau compte administrateur ou modifier le mot de passe d'un compte administrateur de domaine existant par accès direct au moteur Active Directory ou aux fichiers stockant le contenu de l'annuaire. Il n'existe qu'une seule « preuve de concept » publique de cette attaque [15]. Microsoft (ou ses clients) a toutefois jugé le risque suffisamment important pour faire apparaitre la notion de « Read-Only Domain Controller » (RODC) dans Windows 2008. Les modifications apportées à ces contrôleurs ne sont jamais répliquées dans le domaine.

3.7 Conclusion

Les conclusions précédentes sont issues d'une analyse technique de l'authentification dans les environnements Windows. Elles sont donc totalement reproductibles et irréfutables. Malgré tout, on continue à trouver une majorité d'analyses et d'outils sur Internet prétendant améliorer le niveau de sécurité en générant des mots de passe « complexes », en renommant le compte « administrateur » local, ou en recommandant le verrouillage de comptes définitif après 3 essais infructueux. La majorité des entreprises imposant aujourd'hui une certaine complexité sur les mots de passe, voire une certaine forme d'analyse automatique des journaux système, l'attaque par

dictionnaire sur tous les comptes de domaine appartient à un passé lointain. Les techniques d'attaque exploitant les failles conceptuelles des protocoles LM et NTLM sont beaucoup plus efficaces et discrètes. Pendant longtemps il était difficile de trouver des outils publics dépassant le stade de la « preuve de concept » pour ces attaques. Mais les travaux d'Aurélien Bordes, Hernan Ochoa [16] et du projet Metasploit [17] ont largement contribué à industrialiser ces attaques.

4 Exemple : test d'intrusion

4.1 Contexte

Après avoir présenté les concepts théoriques autour des attaques « fatales » contre l'authentification Windows, voyons comment tout ceci s'articule dans le cadre d'un test d'intrusion au sein d'un réseau d'entreprise « classique ». En général, la littérature sur le sujet commence toujours par un avertissement à la valeur légale douteuse : « toute ces techniques sont à utiliser exclusivement contre des réseaux dont vous avez la responsabilité ».

4.2 Quelques idées reçues sur le test d'intrusion

La qualité du test est corrélé au prix des outils utilisés FAUX

Cette idée fait évidemment le bonheur des vendeurs. Elle est issue du fait que les outils circulent essentiellement dans le monde anglo-saxon, où les budgets et les pratiques d'achat sont très différents du contexte français. Il faut savoir que les services gouvernementaux américains peuvent acheter au prix le plus bas jamais pratiqué pour un client privé, ce qui tire les prix vers le haut. Sans parler des incitations fortes que sont les contraintes réglementaires et législatives à utiliser certains types d'outils (ex. déploiement des correctifs de sécurité, archivage des traces, etc.). Bien évidemment la quantité d'attaques disponibles dans l'outil, et la qualité de leur implémentation, va jouer sur les résultats obtenus. Ces données sont proportionnelles à l'effort, donc au coût. Mais on peut opposer que :

- Plus une attaque est complexe, moins elle sera fiable de toute façon. C'est le cas de la majorité des attaques de type « Heap Overflow » ou « faille noyau » aujourd'hui.
- A contrario, une attaque simple est probablement disponible « sur étagère », ou facile à ré-implémenter par l'auditeur en quelques lignes de script.
- Enfin les techniques les plus efficaces sont celles qui n'utilisent aucune attaque, mais exclusivement des outils « légitimes ».

Ce dernier point est essentiel. Dès qu'un outil est labellisé « offensif », il est immédiatement ajouté dans toutes les bases de signatures existantes. C'est le cas par exemple de l'outil NetCat, alors que cet outil se contente de se connecter à un port ou d'en ouvrir un sur la machine locale (opération reproductible en quelques lignes de script). Ainsi il est beaucoup plus discret et efficace d'utiliser la commande `net user /domain` pour énumérer les utilisateurs, que des outils intrusifs (gratuits ou payants).

Un test est plus efficace avec un « 0day » FAUX

Certains vendeurs présentent les attaques « 0day » comme un argument jouant en faveur de leurs produits (ou de leurs services), que ce soit du côté offensif (produits d'intrusion) que du côté défensif (produits IDS/IPS). Non seulement cette idée est fausse, mais de plus elle est nocive. La recherche de failles dans une application particulière, afin d'évaluer la qualité générale de celle-ci, peut effectivement faire l'objet de prestations particulières. Mais dans le cadre général de la protection d'une entreprise, il est évident que des failles seront trouvées tôt ou tard dans les logiciels utilisés, y compris Windows. Pas besoin d'exhiber un « 0day » dans Acrobat Reader pour savoir qu'il faut au minimum désactiver le support de JavaScript dans ce logiciel ! La « bonne » question à se poser est plutôt celle de la gestion des correctifs et de la défense en profondeur (je reviendrai sur ces thèmes dans la partie dédiée à la prévention). Quant à la présence d'attaques inconnues dans les produits IDS/IPS, cette idée est particulièrement nocive pour l'industrie de la sécurité puisqu'elle s'apparente à du racket ...

Le test mesure la compétence de l'auditeur VRAI

Si le test d'intrusion se doit d'être reproductible, il ne pourra jamais être automatisé (comme le laissent pourtant croire de nombreux vendeurs dont les produits sont censés assurer une conformité réglementaire aux obligations de tests réguliers). L'auditeur se doit d'égaler la créativité des attaquants qui sévissent actuellement sur Internet, et dont le foisonnement d'idées n'est plus à démontrer. En fonction du temps imparti, des conditions locales rencontrées et de ses compétences (voire de celles de son équipe), l'auditeur devra faire des choix dans les tests réalisés. La qualité d'un bon rapport est de démontrer des scénarios réalistes, d'être adapté au contexte de l'entreprise auditée, de prendre du recul sur les résultats techniques, et de proposer des solutions applicables ... De telle sorte que le prochain test (dans le cadre d'audits réguliers) soit une nouvelle source d'émerveillement et pas un exercice de routine.

4.3 Déroulement d'un test interne

Schéma général Le plan de bataille du test d'intrusion interne « classique » n'a pas évolué depuis les origines, seules les techniques mises en œuvre se sont adaptées aux évolutions des réseaux. Ce schéma est le suivant :

1. Elévation de privilèges sur la cible
2. Collecte d'information locale
3. Découverte de l'environnement
4. Exploitation d'une faille et compromission d'une cible

4.4 Élévation de privilèges sur la cible

Dans le cas où l'auditeur dispose d'un poste appartenant au réseau à auditer, il se doit d'abord de collecter toutes les traces locales possibles et imaginables. Et pour se faire, il est souvent nécessaire soit d'être « administrateur » local de la cible, soit de pouvoir accéder directement à la mémoire ou au disque dur de celle-ci. Parmi les 10 commandements de la sécurité, il en existe un qui ne s'est pas encore démenti : « si quelqu'un dispose d'un accès physique à votre machine, alors ce n'est plus votre machine ». Je ne vais pas détailler ici toutes les techniques disponibles, mais de ma propre expérience les techniques les plus courantes sont :

- L'amorçage sur support externe (ce qui inclut CD-ROM/DVD, disquettes USB, disques USB, disques FireWire, amorçage PXE, partition de « support » ou de « récupération » préinstallées par les constructeurs, etc.).
- Le démontage du disque dur (si celui-ci n'est ni chiffré, ni protégé par mot de passe ATA).
- L'utilisation du débogueur noyau Windows sur le port série.
- Les erreurs de configuration locale (principalement droits d'accès aux fichiers, répertoires, clés de base de registre, et services) – cette technique étant la seule à pouvoir être utilisée à distance, sans accès physique à la machine.

Ce qui n'exclut par l'extraction de données en mémoire physique par le port FireWire ou d'autres techniques tout aussi expérimentales, principalement destinées à « épater la galerie ».

4.5 Collecte d'information locale

Le contrôle total du système local étant obtenu, il reste à en extraire la substantifique moelle. De ce point de vue, le développement des techniques offensics (« offensive forensics ») est tout à fait appréciable. L'objet de cet article n'est pas

d'énumérer toutes les techniques à disposition de l'auditeur, qui ont déjà été largement documentées et dont le corpus s'enrichit chaque année. Il faut toutefois noter qu' la fin de cette étape - qui se déroule sans connexion au réseau, voire machine éteinte – l'auditeur dispose bien souvent d'un jeu de cibles sensibles et de mots de passe tels que le mot de passe « administrateur » local.

4.6 Découverte de l'environnement

Le terme « environnement » désigne au sens large : machines, comptes, services, topologie réseau, etc. En clair, toute partie de la cible susceptible de compromission. Tout l'art de cette étape est de minimiser le trafic « anormal » risquant d'alarmer les exploitants, voire de porter atteinte à l'intégrité des cibles. La furtivité fait rarement partie des objectifs d'un test d'intrusion, mais en pratique la principale cause de détection du test n'est pas la présence de sondes IDS, mais bien l'explosion du trafic réseau dû à l'utilisation irraisonnée de « scanneurs de vulnérabilités ».

4.7 Compromission d'une nouvelle cible

Entendons-nous bien : de nos jours, il est de plus en plus rare que cette « faille » soit un « buffer overflow » quelconque. Les systèmes d'exploitation intègrent nativement des défenses de plus en plus élaborées, la complexité des codes d'exploitation diminue d'autant leur fiabilité, le coût de développement d'un code d'exploitation devient prohibitif (ex. 3 semaines pour implémenter un « heap overflow » d'après la société Immunity, leader dans le domaine).

Par « faille », j'entends tout type de défaut de configuration facilement exploitable : mot de passe faible, partage réseau mal protégé, port ou sous-réseau non filtré par un pare-feu, etc.

Par expérience, la réutilisation du mot de passe « administrateur » local ou d'un compte de service sur une autre machine est la faille la plus prévalente pour se propager au sein d'un domaine Windows. Il s'agit d'une faille liée à l'effet de la duplication des postes depuis un « master » et à la nécessité (discutable) pour le personnel de support - ainsi que parfois certains scripts et services - d'avoir accès à ce mot de passe. Dans ce cas, le scénario de compromission est malheureusement assez simple et quasiment imparable :

1. Extraction des hash LM/NTLM du compte administrateur local depuis le poste local.
2. Identification des utilisateurs logués sur chaque poste du domaine avec la commande standard nbtstat.

3. Connexion aux postes d'administrateurs avec la combinaison des outils « pass the hash » et PsExec (un outil Microsoft, ex-SysInternals).
4. Extraction des hash du compte administrateur de domaine sur le poste compromis.
5. Connexion au contrôleur de domaine avec les hash précédents et création d'un nouveau compte administrateur de domaine.

Bien entendu il existe quelques variantes du scénario précédent, par exemple si les utilisateurs stockent leurs mots de passe dans des logiciels tels qu'Internet Explorer, FireFox, Putty, SecureCRT, KeePass, Excel, Notepad, ou autre ...

4.8 S'il n'y a pas de faille...

... c'est qu'il faut chercher plus fort

Un cas plus rare qu'on ne l'imagine, mais qui peut toutefois se rencontrer, est celui où il n'existe aucun compte privilégié[3] présent sur au moins 2 machines de l'entreprise avec le même mot de passe. Si aucun mot de passe n'a pu être collecté localement, sur des partages réseau « cachés » ou mal protégés, dans des sauvegardes moins bien protégées que les originaux, ou dans la description des comptes, c'est déjà que le réseau est plus solide que la moyenne. Pour aller plus loin, il est nécessaire d'aller chercher le mot de passe à la source – c'est-à-dire chez les utilisateurs.

Redirection de trafic La redirection de trafic est en général une technique de dernier ressort pour aller capturer un mot de passe utilisateur, car elle induit des effets potentiellement disruptifs sur le réseau. Pour se faire, l'auditeur dispose d'un large panel de possibilités.

– Le spoofing ARP

Très simple à réaliser (même des journalistes y arrivent), encore efficace en 2008 [18], cette technique est toutefois très connue, donc largement détectée par les IDS, et facile à bloquer par des options de configuration des routeurs. C'est loin d'être ma technique de prédilection.

– DHCP

Mettre en place un faux serveur DHCP est une solution simple et efficace, vu le nombre d'options de configuration qu'il est possible de fournir dans une réponse DHCP. C'est d'ailleurs une technique utilisée récemment par le virus « DNSChanger » [19]. Là encore cette technique est assez simple à bloquer par une bonne hygiène réseau ... sauf qu'en vertu du biais vu en introduction, presque personne ne met en place cette contre-mesure avant d'avoir été victime d'une attaque.

[3] On pense principalement au compte administrateur local, mais certains produits utilisent également un compte de domaine privilégié pour démarrer localement des services

– DNS

Les attaques sur les caches DNS sont bien connues, car elles (re)font régulièrement l'actualité. Mais ces attaques sont verbeuses et aléatoires. Une autre solution consiste à utiliser la fonction « mise à jour DNS dynamique » des serveurs Windows. Cette fonction n'est sûre que dans la configuration par défaut. Il existe plein d'autres configurations qui permettent à un client de créer des entrées DNS arbitraires, voire de remplacer des entrées existantes :

– Si la zone n'est pas intégrée à Active Directory mais gérée dans un fichier « plat », alors aucune ACL n'est applicable aux enregistrements.
– Si l'enregistrement des noms DNS est à la charge du serveur DHCP, et que celui-ci s'exécute sur la même machine que le serveur DNS, alors le service DHCP peut modifier n'importe quel enregistrement.
– Si l'enregistrement des noms DNS est à la charge du client, et que les mises à jour « non sécurisées » sont activées sur le serveur, alors tout client peut créer ou modifier des entrées DNS.

Il est intéressant de constater que tous les scanneurs de vulnérabilités du marché que j'ai pu rencontrer ne testent pas ces configurations, pourtant très dangereuses.

– WPAD, ISATAP, et autres noms réservés

Certains noms DNS, s'ils existent, prennent un sens particulier pour Windows. WPAD est par exemple utilisé par Internet Explorer comme serveur mandataire (proxy) par défaut si l'option « détecter automatiquement les paramètres de connexion » est cochée (cas par défaut). Combinée avec la création d'enregistrements DNS arbitraires vue précédemment, cette technique d'attaque est particulièrement redoutable ... Elle a d'ailleurs défrayé la chronique en 2007, lorsque H. D. Moore (auteur du projet Metasploit) fit une présentation à Black Hat sur toutes les techniques d'intrusion n'utilisant aucune exploitation de faille binaire [20]. A noter que ce problème a finalement été adressé par le correctif MS09-008 [21].

– NetBIOS & WINS

En environnement Windows, DNS est loin d'être le seul protocole de résolution de noms. Les protocoles basés sur « NetBIOS » sont également critiques pour le bon fonctionnement du réseau. Ces protocoles sont nombreux et souvent mal compris. Pour faire simple, un client a le choix entre un broadcast UDP, la requête du Master Browser local, ou un serveur WINS. L'utilisation du broadcast pose évidemment un problème de sécurité, toute personne située dans le domaine de broadcast pouvant y répondre. Par conception du protocole, les noms NetBIOS sont limités à 15 caractères. Ce qui permet toutefois de voir parfois passer des trames de résolution en broadcast pour le nom `WWW.GOOGLE.FR` par exemple ...

– ICMP Redirect

L'horizon de la sécurité informatique ne se limite pas aux attaques découvertes ces 2 dernières années. En pratique, des « vieilleries » tels que les messages ICMP Redirect peuvent encore fonctionner. On imagine mal le nombre d'imprimantes ou de serveurs de stockage construits sur un Windows NT4 Embedded, par exemple ...

Capture des authentifiants L'intérêt de rediriger du trafic utilisateur est de pouvoir capturer des authentifiants dans le flux. Il existe plein de protocoles réseau qui vont alors permettre de capturer le mot de passe de l'utilisateur en clair (POP3, IMAP, FTP, HTTP - y compris l'authentification sur le proxy si applicable, etc.). Il existe également une méthode plus active, qui consiste à forcer un client à s'authentifier sur une ressource contrôlée par l'attaquant par le biais d'un lien spécialement conçu (`http://attaquant/` ou `\\attaquant\ressource`) et d'un serveur malveillant (le `module smb_sniffer` du projet Metasploit est fait pour cela). Il ne reste plus qu' diffuser le lien par email, par un partage réseau commun ou par la compromission d'un serveur Web Intranet par exemple.

Attaque en réflexion Dans tous les cas, et comme expliqué précédemment, la capture un défi/réponse LM ou NTLM est aussi dur à « casser » que ne l'est le mot de passe de l'utilisateur. Une légende dit que les mots de passe les plus courants sont « god », « sex » et « password ». En France, une statistique personnelle indique plutôt « soleil », « vacances » et « bonjour ». Toutefois la majorité des entreprises imposent un mot de passe complexe, ce qui rend les mots de passe précédents caducs (on ne se méfie jamais assez de « bonjour01 », ceci dit). Heureusement, il existe une technique tout à fait redoutable (et disponible sur étagère dans le projet Metasploit) : la réflexion des authentifiants. L'explication de cette technique tient en 5 lignes, une fois le protocole de défi/réponse bien compris :

1. Le client se connecte sur une ressource contrôlée par l'attaquant.
2. L'attaquant demande alors un « défi » au serveur qu'il souhaite compromettre.
3. L'attaquant transmet ce « défi » au client.
4. Le client retourne la « réponse » chiffrée par ses hash LM/NTLM.
5. L'attaquant transmet cette « réponse » au serveur.
6. Si le client fait partie des utilisateurs qui disposent d'un accès légitime au serveur, l'attaquant est alors authentifié auprès du serveur sous l'identité du client.

Cette attaque ne nécessite pas de « casser » les hash et de remonter au mot de passe en clair.

4.9 Efficacité des contre-mesures

En résumé, les étapes du test d'intrusion classique sont :

1. Récupération du mot de passe « administrateur » local
2. Identification du poste d'un administrateur de domaine (commande `nbtstat`)
3. Récupération des hash sur ce poste (commande `PsExec`)
4. Connexion au contrôleur de domaine

Passons maintenant en revue les produits de sécurité les plus courants et leur inefficacité dans cette situation :

– Antivirus

L'antivirus ne détecte que les exécutables réputés nuisibles. Il est donc inefficace contre la commande nbtstat ou les outils SysInternals (rachetés et distribués par Microsoft). Stricto sensu, le rôle de l'antivirus n'est pas de détecter les attaques réseau - même si de nombreux produits se vantent d'être « tout en un ».

– HIDS / HIPS

La détection de comportement anormal, telle qu'elle est souvent implémentée dans les produits, vise principalement à détecter :

- Les attaques binaires de type « Buffer Overflow » par des heuristiques de type « détection de shellcode ». Mais dans notre scénario, aucune faille de type « Buffer Overflow » n'est utilisée, car elles sont jugées « peu fiables ».
- Les comportements de malwares, tels que l'exécution de code depuis le répertoire « Temporary Internet Files » ou l'ajout de données dans la clé « Run ». Ceci est très loin des techniques utilisées lors d'une intrusion « manuelle ».
- Une mention particulière doit être apportée à la surveillance des API système. Sans trop rentrer dans les détails techniques, `WriteProcessMemory()` et `CreateRemoteThread()` sont souvent considérées comme des API dangereuses car pouvant permettre l'injection de code. Mais les « bons » outils d'extraction de hash (comme « pass the hash ») fonctionnent uniquement avec l'API `ReadProcessMemory()`, ce qui les rend stables et peu susceptibles d'être détectés.

– NIDS / NIPS

Les connexions effectuées par les outils mis en œuvre sont exclusivement basées sur les protocoles MS-RPC et SMB, qui sont toujours autorisés dans un contexte de domaine Windows. Lorsqu'un module pare-feu est intégré à ces produits, dans presque toutes les configurations rencontrées sur le terrain, l'ensemble des composants du domaine Windows (postes et serveurs) sont autorisés à communiquer entre eux avec les protocoles précédents. Les outils NIDS/NIPS cherchent principalement des « signatures » d'attaques connues (ex. scan NMAP, « ping of death », « NOP sled »

dans un shellcode, etc.). Dans ces conditions, ne pas utiliser d'outil réputé « offensif » garantit une indétectabilité quasi absolue, d'autant que le trafic de l'attaquant est noyé dans un volume considérable de trafic MS-RPC/SMB « légitime ».

– SIEM (Security Information & Event Management)

Les SIEM sont aussi efficaces que leurs « yeux », à savoir tous les logiciels dont ils agrègent l'information : antivirus, HIDS et NIDS. Toute attaque non détectée par ces sondes ne le sera pas plus par un SIEM. Le SIEM intègre également les journaux d'activité système (en général sur les serveurs, mais pas sur les postes de travail). Un attaquant malin, ayant opéré à distance depuis le poste d'un administrateur de domaine, ne laissera donc aucune trace suspecte dans le SIEM. Tout au plus pourra-t-on éventuellement tracer a posteriori la création d'un nouveau compte administrateur et l'usage qui en est fait ... si l'attaquant a choisi de poursuivre son test par cette méthode.

5 Plus de sécurité – avec moins de complexité

5.1 Contexte

Protéger un réseau d'entreprise est une tâche plus difficile que de l'attaquer, compte-tenu des failles conceptuelles dans les systèmes Windows vues précédemment, du contexte historique de l'entreprise parfois lourd à porter (ex. telle application qui ne fonctionne qu'avec un compte « administrateur »), et des pratiques des administrateurs eux-mêmes (souvent difficiles à changer). Les seules questions à se poser sont :

– « Cette attaque est-elle techniquement possible ? » - ce qui signifie qu'elle sera probablement disponible « sur étagère » tôt ou tard - et il vaudra mieux à ce moment que ce soit dans une conférence de sécurité plutôt que dans un ver.

– « Comment puis-je mettre en œuvre une architecture capable de m'en protéger, indépendamment des détails d'implémentation de l'attaque ? ».

La tendance générale du marché est d'ajouter toujours plus de lourdeur et de complexité pour corriger des problèmes assez simples à la base, mais pris trop tard dans le cycle de déploiement des technologies. Pour prendre un exemple, je ne prêche pas pour les listes blanches d'exécutables (comme le fait Joanna Rutkowska), mais il est vrai que cette solution serait beaucoup plus efficace que de gérer d'énormes bases de signatures, avec une définition du code « malveillant » dépendante des éditeurs, et des risques de faux positifs - comme c'est le cas actuellement. L'effort d'adaptation initial du système d'information vers une sécurité « proactive » est important, mais sur le long terme il est évident que cela génère des économies durables, compte-tenu du coût des différents produits de sécurité nécessaires pour survivre aujourd'hui, des impacts de ces produits sur le support, et du coût des différentes crises (comme « Conficker »).

5.2 Vers une sécurité « proactive »

La probabilité qu'une faille critique soit trouvée dans Windows à court terme est de 1, puisque de telles failles sont publiées chaque 2ème mardi du mois. Et maintenant ? Les formats « .DOC » et « .PDF » font l'objet d'attaques récurrentes et parfois ciblées, sans correctif disponible chez l'éditeur ni signature dans l'antivirus. Faut-il pour autant bloquer d'autorité ces formats sur les passerelles Web et la messagerie ? De la même manière, il a été démontré par le passé que les ports « série[4] » et « FireWire » présentent un risque immédiat pour la sécurité des machines en cas d'accès physique. Mais il n'est pas besoin de comprendre en détail l'architecture des PC pour imaginer que le port « PCMCIA », voire les plus récents « ExpressCard » et « eSATA », pourraient également être utilisés comme vecteur d'accès... Sauf que personne n'a encore eu l'idée (la date de rédaction de cet article) de faire un « talk » ou un « white paper » sur le sujet. Face à toutes ces questions, la meilleure solution consiste toujours à adresser le problème à la racine. Mais il est souvent difficile de penser « out of the box » (c'est-à-dire hors des options activables par une ligne ou un clic) et d'adopter une stratégie globale cohérente, en évitant de laisser délibérément des maillons faibles, car « ça ne s'est pas vu ailleurs » ou « il n'existe pas de produit pour le faire ». La liste que je propose ci-dessous n'engage absolument pas ma responsabilité si vous vous faites pirater après l'avoir mise en œuvre. Mais il me semble néanmoins qu'elle constitue le socle de survie dont il est nécessaire de disposer pour couvrir l'état de l'art en matière d'intrusion ...

5.3 Ma liste todo

La « bonne » gestion des secrets Le risque principal est lié aux mots de passe, et les exigences de longueur et/ou complexité sont aujourd'hui inutiles comme on l'a démontré précédemment. L'essentiel est d'empêcher la compromission des éléments d'authentification (hash LM/NTLM) par tous les moyens, ce que couvrent toutes les rubriques suivantes.

– Les comptes locaux - et particulièrement le compte « administrateur » - doivent tous être désactivés.

Ceci impacte souvent le support bureautique. Il faut déjà noter que le compte « administrateur » local reste utilisable lorsque Windows est démarré en mode « sans échec ». Ensuite, des comptes de domaine nominatifs et révocables individuellement peuvent être utilisés pour les accès de maintenance, si nécessaire. Certains produits de prise en main à distance gèrent leur propre système d'authentification - il convient de s'en méfier, particulièrement du service VNC installé sur toutes les machines avec

[4] En cause : l'option F8 au démarrage de Windows permettant d'activer le débogueur noyau

le même mot de passe ! Dans le pire des cas, un compte local dont le mot de passe est généré à partir d'un secret et d'un élément lié au poste est préférable à un mot de passe identique partout !

– Il n'est pas recommandé d'utiliser des comptes de service autres que « SYSTEM », « LocalService » et « NetworkService ».

En effet, si un compte de service est utilisé, son mot de passe doit être stocké en clair dans une zone appelée « secrets de la LSA », ce qui ouvre un risque de compromission immédiat. Éventuellement, ces comptes de service peuvent être des comptes locaux, au mot de passe aléatoire, différent sur chaque poste, et géré par le service - ce que peu de produits savent faire malheureusement.

– Un système de gestion des mots de passe doit être proposé.

Pour les plus riches, cela consiste en une solution de SSO. Sinon un gestionnaire de mots de passe comme il en existe plein (KeePass, Password Safe, etc.), bien utilisé, remplace avantageusement la feuille Excel ou le fichier texte à plat utilisé par défaut !

– Les techniques d'accès physique au poste doivent être bloquées.

Les techniques présentées ici concernent :

– L'amorçage sur support externe : facile à bloquer dans les paramètres du BIOS.
– L'accès physique au disque : peut être bloqué par un chiffrement intégral ou un simple mot de passe ATA ... pourvu qu'il ne soit pas connu de l'utilisateur (dans le scénario de l'utilisateur malveillant).
– L'accès au débogueur noyau. Il n'existe pas d'autre technique que de désactiver le port série dans le BIOS. Fort heureusement, il y a de moins en moins de port série sur les machines récentes, et celui-ci est bien souvent émulé sur un bus USB qui n'est pas supporté par défaut par Windows XP.

Les « bonnes » options de sécurité dans Windows Il existe quelques options de sécurité facilement accessibles dans Windows qui permettent de bloquer une partie des techniques précédentes. Ces options sont accessibles par la stratégie de groupe « stratégie de sécurité », mais c'est le nom de la clé de base de registre sous-jacente qui est donnée ci-dessous (car il s'agit de l'information de référence).

– « NoLMHash » : permet de désactiver le stockage du hash LM au prochain changement de mot de passe.
– « NTLMAuthenticationLevel » : permet d'activer le support du protocole NTLMv2 uniquement, qui intègre une authentification mutuelle des extrémités (contrairement à ses prédécesseurs).
– « EnableSecuritySignature » / « RequireSecuritySignature » : permet d'activer la signature SMB, qui (si elle est rendue obligatoire) bloque les attaques en réflexion d'authentifiants.

Enfin le patch MS08-068 protège les systèmes contre la réflexion d'authentifiants dirigée contre eux-mêmes, en mémorisant la liste des derniers « défis » envoyés pour empêcher leur rejeu.

La gestion des périmètres d'administration

– Une mesure très efficace consiste à empêcher les communications réseau de client à client, et de n'autoriser que les connexions de client à serveur.

Ceci protège contre les risques de rebond présentés précédemment, et contre les attaques applicatives. En effet, il n'est pas rare de trouver sur les clients de nombreux ports ouverts par des applications dont la sécurité est mal maitrisée : installations sauvage d'applications Web sur EasyPHP, agents Symantec ou McAfee, composant ActiveSync, etc. Microsoft préconise l'écriture de règles pour le moteur IPSEC afin de réaliser cette isolation. On notera qu'il n'est absolument pas nécessaire d'utiliser IPSEC pour mettre en œuvre cette recommandation, seule la partie « pare-feu » du moteur IPSEC étant utilisée. Il est également possible de mettre en œuvre cette recommandation sur les routeurs en utilisant la fonction « Private VLAN » [22] par exemple. Les cibles les plus critiques sont les postes des administrateurs (aussi bien système, réseau qu'applications).

– Il est totalement inacceptable que les administrateurs de domaine travaillent au quotidien avec un compte administrateur de domaine.

La commande « RunAs » de Windows est effectivement assez contraignante à utiliser, du fait qu'elle oblige à saisir le mot de passe à chaque commande privilégiée. Mais il existe des solutions gratuites (telles que « SudoWin » [23] ou « Superior SU » [24]) pour automatiser ces tâches.

Protection des composants non maitrisés Le problème des logiciels tiers en « boite noire » se pose, car c'est par ces composants que proviennent la majorité des attaques sur Internet aujourd'hui. La situation n'est pas aussi inextricable qu'il n'y parait, car la plupart de ces logiciels sont configurables, même si la documentation sur le sujet est peu abondante. Voici quelques pistes pour les logiciels les plus courants.

– **Adobe Acrobat Reader :** Pour Acrobat 9.0, la configuration (et en particulier l'option `JSPrefs\bEnableJS` qui permet de désactiver JavaScript) se trouve dans la base de registre sous la clé `HKCU\Software\Adobe\Acrobat Reader\9.0`. Cette configuration est applicable par utilisateur. Il est également possible de désactiver des plugins dans le répertoire `C:\Program Files\Adobe Reader 8.0\Reader\plug_ins` ou `plug_ins3d`, mais ces modifications affecteront alors tous les utilisateurs.
– **Adobe Flash Player :** La configuration du lecteur Flash peut être modifiée globalement en éditant le fichier suivant : `C:\windows\system32\macromed\flash\mms.cfg`

– **Sun Java :** La configuration de la JVM Sun en version 1.6 peut être modifiée globalement en éditant le fichier `C:\Program Files\Java\jre6\lib\security\java.policy`
– **Office 2003** L'utilitaire MOICE (mis à disposition gratuitement par Microsoft 25) permet de contenir une éventuelle exécution de shellcode à l'ouverture d'un document Office malformé. En effet, le document est alors converti dans un environnement d'exécution extrêmement contrôlé. Pour la petite histoire, la même technique est utilisée par le navigateur Google Chrome pour le rendu des pages Web.

6 Conclusion

La sécurité fait partie des domaines où les résultats devraient primer sur les moyens. De nombreux problèmes pragmatiques, comme le vers Conficker qui s'amuse d'une simple clé de registre mal configurée, ne nécessite aucunement d'avoir le dernier matériel à la mode qui arrête tous les pirates et vous offre le luxe de faire le café pour vous. Il faut simplement de la rigueur, des administrateurs attentifs et curieux et quelques moyens de faire respecter les politiques de sécurité d'ores et déjà établies. La course à l'armement permet seulement d'afficher des étoiles sur son trois pièces, mais ne permet jamais, in fine, de sauver des vies. Dans une société où l'outil informatique est devenu la pierre angulaire, il semble inopportun de jouer la santé de votre entreprise sur des grigris l où de simples notions d'hygiène permettraient d'enrayer la globalité de la menace. Il est évident qu'aucun retour sur investissement (le sacro saint ROI) ne peut être évoqué dans notre matière, ce qui la rend parente pauvre, loin derrière la production ou le commercial. Ce qu'il faut cependant garder présent à l'esprit, c'est combien coûte aujourd'hui un retour à l'âge du papier-crayon. L où des moyens simples assurent, contre un peu de sueur, un état connu et maîtrisé, faut-il opposer des usines à gaz disponibles sur étagère ?Aujourd'hui, la sécurité n'est pas seulement une question de moyens financiers, c'est avant tout une question de bon sens, de compétence et de volonté réelle.

PS. Merci à mes relecteurs, Elvis Tombini et Jean-Philippe Gaulier, pour leurs contributions éclairées à ce vaste sujet.

Références

1. Bruce Schneier : The Death of the Security Industry, http://www.schneier.com/essay-196.html
2. Bruce Schneier : Do We Really Need a Security Industry ?, http://www.schneier.com/blog/archives/2007/05/do_we_really_ne.html
3. SANS ISC : Thoughts on the Best Western Compromise, http://isc.sans.org/diary.html?storyid=4928

4. MS08-067, Vulnerability in Server Service Could Allow Remote Code Execution, http://www.microsoft.com/technet/security/Bulletin/MS08-067.mspx
5. MS05-043, Vulnerability in Print Spooler Service Could Allow Remote Code Execution, http://www.microsoft.com/technet/security/bulletin/MS05-043.mspx
6. Wired : I bought votes on Digg, http://www.wired.com/techbiz/people/news/2007/03/72832
7. Phrack numéro 49 : Smashing The Stack For Fun And For Profit, http://www.phrack.com/issues.html?issue=49&id=14
8. Fortify, NIST SHA-3 Competition Security Audit Results, http://blog.fortify.com/repo/Fortify-SHA-3-Report.pdf
9. Slashdot : Why Most Published Research Findings Are False, http://science.slashdot.org/article.pl?sid=08/10/19/172254
10. Linus Torvalds : The Security Circus, http://article.gmane.org/gmane.linux.kernel/706950
11. Wired, Under Worm Assault, Military Bans Disks, USB Drives, http://blog.wired.com/defense/2008/11/army-bans-usb-d.html
12. Silicon, Le virus Conficker touche la Marine française et ses Rafales, http://www.silicon.fr/fr/news/2009/02/09/le_virus_conficker_touche_la_marine_francaise_et_leurs_rafales
13. Aurélien Bordes, SSTIC 2007, Secrets d'Authentification sous Windows, http://actes.sstic.org/SSTIC07/Authentification_Windows/
14. Emmanuel Bouillon, PacSec 2008, Gaining Access Through Kerberos, http://dragos.com/psj08/slides/psj08-en/PacSec08_Bouillon.ppt
15. RADPass, an offline Active Directory password remover, http://www.tbiro.com/projects/RADPass/
16. Hernan Ochoa, Pass The Hash Toolkit, http://oss.coresecurity.com/projects/pshtoolkit.htm
17. MS08-068 : Metasploit and SMB Relay, http://blog.metasploit.com/2008/11/ms08-067-metasploit-and-smb-relay.html
18. ARP Spoofing vs. BlackHat 2008, http://news.cnet.com/8301-1009_3-10010989-83.html
19. Virus « DNSChanger » (a.k.a. « Trojan.Flush »), http://www.symantec.com/security_response/writeup.jsp?docid=2008-120318-5914-99&tabid=2
20. H.D. Moore/Valsmith, Black Hat USA 2007, Tactical Exploitation, http://www.metasploit.com/data/confs/blackhat2007/tactical_blackhat2007.pdf
21. MS09-008, Vulnerabilities in DNS and WINS Server Could Allow Spoofing,http://www.microsoft.com/technet/security/Bulletin/MS09-008.mspx
22. Wikipedia, Private VLAN, http://en.wikipedia.org/wiki/Private_VLAN
23. SudoWin, http://sourceforge.net/projects/sudowin
24. Superior SU, http://www.stefan-kuhr.de/cms/index.php?option=com_content&view=article&id=62&Itemid=73
25. Release of Microsoft Office Isolated Conversion Environment (MOICE) and File Block Functionality for Microsoft Office http://www.microsoft.com/technet/security/advisory/937696.mspx