Techniques d'intrusion dans les réseaux By Shark

De plus en plus d'attaques sont répertoriées par les agences officielles, ont voit surtout de plus e plus de personnes qui vont en prison, mais avez vous que la plupart des actes de piratages n'auront pas suite, car les entreprises à part un firewall et un antivirus n'ont pas de moyen pour lutter contre les "crasheurs"; nous allons aborder ici les bases qui permettent de comprendre comment font ces pirates...

Sommaire

1. Introduction

La connaissance des stratégies, méthodes et techniques employées par les pirates informatiques est de même primordiale pour la compréhension de leurs comportements, et l.élaboration de lignes de

défense adaptées : quelque soit la sophistication ou le raffinement des techniques utilisées, l.expérience nous fait dégager des lignes directrices quant au déroulement de telles attaques. Nous allons ici décrire d.un point de vue technique les différentes phases de l.attaque1 d.un système d.information par un agresseur situé sur Internet : de l.identification de la cible à l.exploitation puis à la progression sur le système compromis. 2. Cadre de la présentation Les menaces potentielles sur un système d.information sont multiformes et difficilement identifiables en raison de la variété des agresseurs, des cibles possibles et des moyens d.attaques. Nous allons dans la suite considérer le cas d.un agresseur externe à l.organisation cible, quelles que soient ses motivations et sa nature. En première approche, les possibilités de compromission du système visé se résument à la capacité de cet agresseur à se créer une porte d.entrée : - Compromission directe des accès extérieurs, tels que : accès Internet, système téléphonique (PABX), X.25 ou lignes dédiées vers des prestataires ou partenaires. - Injection de pièges (chevaux de Troie, virus) - Social engineering (pour obtenir indirectement ou faciliter un des points précédents), détournement de personnel Le présent document s.intéressera au cas des attaques directes contre l.accès Internet du système d.information cible. Notons finalement que même si le siège des accès Internet n.est pas la meilleur méthode pour obtenir une intrusion réussie (X.25 et le scanning téléphonique sont beaucoup plus probants), c.est du moins la plus facile à mettre en .uvre (multiplicité des moyens d.accès, quasi gratuité, .) et celle pour laquelle le plus d.information est disponible.
1

Hors attaques de type déni de service.

3. Identification de la cible

Dans une première phase, l.agresseur identifie la cible dans son environnement (Internet dans notre cas). En utilisant des informations publiques pour la plupart, il découvre les localisations et interactions de l.organisation avec le reste d.Internet, afin d.arriver à une liste exhaustive des accès extérieurs et donc des portes d.entrées potentielles à forcer. Il dispose à la suite de cette phase d.un début de cartographie des environs de la cible sur le réseau Internet, des partenaires et prestataires ayant potentiellement des liens physiques et logiques avec elle, de données brutes telles que des noms d.employés ou numéros de téléphones, qui seront potentiellement utiles pour des tentatives de social engineering ou de wardialing. En fonction des motivations et des résultats attendus, cette phase peut être d.extrêmement

rapide à assez longue, étant donné que son exhaustivité déterminera en grande partie le succès de l.intrusion, en permettant de centrer directement l.attaque sur une porte dérobée ou en donnant des alternatives à l.attaque d.une ligne de défenses efficaces. Dans certains cas, cette phase n.est même pas présente : il est fréquent que l.organisation agressée ne soit pas la motivation d.une attaque. De nombreux pirates balaient des plages d.adresses ou des pays entiers pour trouver des machines vulnérables qu.ils pourront compromettre, le plus souvent pour servir comme point de départ pour d.autres attaques. La cible elle-même est ainsi toute machine située dans le bloc balayé, quel que soit son propriétaire. La phase de recherche et d.identification de la cible est donc ici quasiment inexistante, elle se résume au balayage d.adresses IP ou d.entrées DNS. Etant donné que la majorité des informations recueillies lors de cette phase d.identification sont publiques (le pirate essaie d.éviter de « toucher » directement la cible), elle est assez difficile, voire impossible à détecter. Seules certaines techniques, en particulier les balayages DNS peuvent être visibles par les administrateurs sécurité ou les mécanismes de détection **d.intrusion**.

3.1 DNS, Domain Name System

Le DNS est la base d.informations la plus évidente pour la localisation d.une cible. Afin de connaître les noms de domaines possédés par la victime, notre agresseur peut procéder par essais ou aller consulter les registres publiques des noms de domaines (whois). L.interrogation des registres whois2 se fait directement par nom de domaine ou par mots clef. Le résultat obtenu est la liste des domaines correspondants aux mots clefs entrés, ainsi que toutes les informations associées : entre autres, noms des contacts techniques, administratifs, adresses postales, téléphones, et adresses IP des serveurs DNS. L.agresseur peut ensuite utiliser ces renseignements pour étendre sa recherche en

interrogeant par exemple le registre avec comme mot clef le nom du contact technique. Il obtient ainsi une liste déjà importante de cibles potentiellement utiles pour son intrusion. Fort de ces informations, il est ensuite utile d.évaluer l.étendue des domaines, en utilisant une fois de plus les services du DNS. Nous pouvons ainsi déterminer si les serveurs DNS sont hébergés chez des prestataires ou directement chez la victime. Si tel est le cas, ce prestataire peut lui aussi devenir une cible, en cas d.attaque directe infructueuse. Nous pouvons aussi demander une liste de tous les couples nom/adresse IP enregistrés dans le DNS pour un domaine donné. Cette fonctionnalité, dite AXFR, est nécessaire pour le transfert

de zones entre serveurs de noms. Elle est très intéressante pour un agresseur et est donc communément restreinte aux seuls serveurs autorisés. Le DNS nous permet finalement d.obtenir les adresses des machines courantes, telles que www.x.com, ftp.x.com, mail.x.com ainsi que les adresses des serveurs de messagerie externes

(entrées MX, mail-exchanger, des DNS). Un dernier type de requête, dite HINFO (host info), donne des renseignements sur la machine cible, tels que : modèle, système d.exploitation. Les entrées HINFO des tables DNS, pour des raisons évidentes de confidentialité et de sécurité ne sont maintenant quasiment jamais renseignées. http://whois.networksolutions.com/cgi-bin/whois/whois pour les domaines TLD .com/.net/.org, les sites des organismes chargés de gérer les noms de domaines nationaux autrement.
2

3.2 Bases d.adresses IP

Nous avons vu comment le DNS procure à un pirate des informations essentielles pour la préparation de son attaque. Le DNS n.est cependant pas exhaustif ; étant donné que chaque machine, donc chaque adresse IP, est potentiellement une cible, notre agresseur va étendre sa connaissance de quelques adresses IP à des blocs entiers d.adresses au moyen des bases d.adresses que sont le RIPE, l.ARIN et l.APNIC. Ces bases sont nécessaires au bon fonctionnement d.Internet car elles détaillent aussi bien les organismes propriétaires, personnels à contacter que les étendues et les routages associés aux netblocks, ces fameux pans d.adresses IP réservées sur Internet. L.agresseur identifie donc les différentes adresses qu.il a recueillies avec DNS grâce aux moteurs d.interrogation des bases d.IP3. Cette identification lui permet de savoir à quel netblock appartient une adresse IP donnée, et donc l.organisation propriétaire de l.adresse. Il élargit donc son champ d.attaque aux adresses non inscrites dans le DNS. Inversement, il va corréler le résultat de ses recherches en interrogeant les bases d.adresses avec comme critère de recherche le nom de la cible, des contacts techniques, .

3.3 Bounces de découverte

La plupart des serveurs de messagerie (SMTP et News) rajoutent aux messages qu.ils véhiculent des informations de diagnostic, masquées ensuite par le logiciel client de l.utilisateur. Ainsi, les en-têtes des e-mails comportent non seulement expéditeur, destinataire, mais aussi tout le cheminement du message au travers des serveurs de messagerie. Typiquement, chaque serveur rajoute au message une en-tête From : décrivant le serveur (adresse IP locale, identification du logiciel et numéro de version) et l.heure de passage. Il en est de même pour les messages NNTP tels que ceux postés sur Usenet. Une technique classique de découverte consiste à envoyer un e-mail de reconnaissance à une adresse invalide du domaine cible. L.e-mail est très souvent retourné à son expéditeur avec un message d.erreur, d.où le nom de bounce. L.examen des en-têtes présents dans le mail retourné va donc donner au pirate des informations importantes comme les versions des serveurs de messagerie externes, les adresses, noms et versions des serveurs de messagerie internes. Ces renseignements seront utiles par la suite lors des attaques et de la progression dans le système d.information compromis. Received: from relais.x.com ([1.2.3.4]) by smtp.Secway.com (smap) with SMTP id QAA21414 for

; Fri, 11 Aug 2000 16:20:06 0400 Received: from 192.1.1.20 by relais.x.com (InterScan E-Mail VirusWall NT); Fri, 18 Aug 2000 18:11:44 0200 (Paris, Madrid (heure d'été)) Received: by

mail.x.com(Lotus SMTP MTA SMTP v4.6 (462.2 9-3-1997)) id C1256938.0043AF89 ; Fri, 11 Aug 2000 14:19:18 0200 Les bases sont séparées géographiquement en: RIPE (Europe): http://www.ripe.net, ARIN (Amérique du Nord/Sud/Centrale, Caraibes, Afrique sous saharienne): http://www.arin.net, APNIC (Asie, Pacifique): http://www.apnic.net
3

Le social-engineering constitue aussi un moyen direct d.utiliser ces informations comme moyen de compromission, en appuyant la tentative de social-engineering par des informations internes découvertes par ce biais. Notons finalement que le problème de la sécurité des applicatifs clients est de plus en plus préoccupant, notamment en raison de la complexité de nombreuses applications telles que les navigateurs Web ou les logiciels clients de messageries. Les attaques dirigées sur le client (en compromettant un serveur et modifiant son comportement, pour exploiter un bug du client par exemple) vont en se répandant et la connaissance par le pirate des serveurs, logiciels utilisés, . peut faciliter grandement une intrusion. Si le bounce ne fonctionne pas (cas des bounces détruits sans retour à l.expéditeur d.origine), le pirate dissimule ceci sous forme d.un e-mail de demande d.information et attend la réponse légitime.

3.4 Moteurs de recherche

L.agresseur ne se limite pas aux seules informations techniques afin de préparer son attaque. L.interrogation des moteurs de recherche Web classiques, tels qu.Altavista, Google, Yahoo, peut révéler de nombreuses informations : - noms d.employés, ayant par exemple postés dans des mailing-lists ou adresses IP d.employés accédant à distance au réseau interne (VPN) - liens hypertextes à partir d.autres sites vers le site cible - communiqués de presse, détaillant des partenariats, rapprochements, . D.autre part, des moteurs plus spécifiques comme DejaNews révèlent en plus potentiellement des éléments techniques intéressants : l.analyse des en-têtes de messages envoyés sur Usenet permet de déterminer le serveur de News utilisé par l.organisation cible, les versions des logiciels clients utilisés, . Nous rejoignons ici la technique de « bounce » de découverte décrite précédemment.

3.5 Outils réseaux de diagnostic

De nombreux outils de diagnostic réseaux permettent de recueillir des informations utiles pour la phase d.identification de la cible. Ainsi, le programme traceroute fournit-il la liste de tous les systèmes (routeurs) entre la source du traceroute et la machine destination. Ces données, à recouper avec celles issues des registres, permettent de recadrer l.attaque vers un prestataire vulnérable, afin par exemple de prendre le contrôle du tuyau utilisé par la cible. Traceroute et ses dérivés laissent cependant des traces dans les systèmes de détection **d.intrusion** (ils utilisent couramment des paquets UDP en incrémentant le numéro de port à chaque saut, la réponse est un paquet ICMP), et les informations peuvent être obtenues par d.autres moyens publics. Leur utilisation dans cette phase est donc superflue et risque d.alerter prématurément les administrateurs de la victime.

3.6 Social-engineering

Même si dans les sociétés ayant déployé une politique de sécurité adéquat les tentatives de social-engineering directes ont peu de chance d.aboutir, une demande d.information bénigne auprès du service de support informatique est très souvent acceptée. Notre agresseur, se faisant passer pour un utilisateur du système, peut ainsi demander les adresses des proxy, des serveurs de messagerie internes, ou des serveurs Web internes, en étoffant sa crédibilité grâce par exemple aux informations recueillies avec les méthodes vues précédemment.

4. Scanning

La phase de recherche environnementale terminée, et la liste des cibles établies, le pirate va s.employer à balayer le réseau cible afin d.en obtenir une topologie détaillée, aussi bien d.un niveau réseau qu.applicatif. L.intérêt de cette phase, qui touche directement les systèmes cibles et est donc la première partie décelable d.une tentative **d.intrusion**, est de trouver un ou plusieurs systèmes « exploitables », c.est-à-dire qu.il sera facile au pirate de compromettre. N.oublions pas aussi le caractère opportuniste d.une attaque : il suffit potentiellement d.une seule vulnérabilité pour réussir

une intrusion sur un système d.information ; c.est aussi la grande différence avec le travail des administrateurs systèmes ou des équipes d.audit, qui doivent identifier et corriger toutes les failles. Le pirate peut donc pour des raisons de furtivité arrêter cette phase de scanning dès qu.il trouve un système vulnérable, le facteur chance est dans ce cas prépondérant.

4.1 Rappel sur les protocoles TCP/IP

TCP/IP est la famille de protocoles utilisés pour le transport logique des données sur Internet, et maintenant sur la plupart des réseaux locaux. Ses mécanismes, conçus il y a plus d.une vingtaine d.années, ont été étudiés pour permettre une facilité d.implémentation et d.utilisation, et souffrent de nombreux problèmes de sécurités inhérents.

4.1.1 IP

Rappelons les formats des paquets IP. IP est le protocole correspondant à la couche « Réseau » du modèle OSI (bien qu.IP soit antérieur à ce modèle). C.est donc IP qui s.occupe de la notion d.adressage, ici sous forme d.adresse IP sous 4 octets, souvent représentées par la notation quadruplet a.b.c.d . IP est responsable du routage des paquets de leur source à la destination ; un routage adaptable au vol est d.ailleurs une des problématiques à l.origine d.Arpa, le précurseur d.Internet. IP s.occupe aussi de la fragmentation des paquets, c.est-à-dire leur découpage en plus petits paquets afin de s.adapter aux contraintes des couches inférieures (physiques). La taille maximum des données que peut véhiculer un support physique donné est exprimée par son MTU (Maximum Transmit Unit) . IP découpe donc les données (fragmentation) pour les adapter au MTU du lien dans le cas d.une transmission, et les regroupe (défragmentation) avant de les remonter aux couches supérieures dans le cas d.une réception de paquet. Finalement, IP veille à ce qu.il n.y ait pas de bouclages de paquets, ou de paquets éternels en attribuant à chacun un Time-to-Live (durée de vie), qui donne le nombre d.interfaces (ou de systèmes, suivant les implémentations) que le paquet peut traverser avant d.être détruit. Une telle destruction de paquet, lorsque son TTL arrive à 0, génère un message ICMP « Time Exceeded in Transit » qui est renvoyé à l.émetteur du paquet. Les paquets IP sont constitués de la façon suivante : (bits) 0 4 8 12 16 18 20 24 28 31 Version IHL Type de service Longueur totale Identification Flags Offset TTL Protocole Checksum en-tête Adresse source Adresse destination Options (si présentes) Padding Données … (par exemple, paquet TCP ou UDP) En-tête IP (IP header) Flags : IP_RF 0x8000 reserved fragment flag IP_DF 0x4000 dont fragment flag IP_MF 0x2000 more fragments flag IP_OFFMASK 0x1fff mask for fragmenting bits Les données véhiculées par un paquet IP ne sont pas des données brutes de communication d.applications, mais des données d.autres protocoles comme TCP ou UDP, dit de transport, encapsulées dans le paquet IP. C.est le principe fondamental de l.encapsulation/décapsulation du modèle en couches. Chaque couche rajoute ses données de contrôle dans une en-tête qui lui

est propre, les données véhiculées (qui contiennent elles-mêmes des en-têtes) lui sont complètement opaques.

4.1.2 TCP

TCP est le protocole IP correspondant au protocole connecté du modèle OSI. TCP s.occupe donc d.assurer un transport fiable d.un service source à un service destination, identifiés tous deux par un numéro dit numéro de port (entre 0 et 65535), en établissant une connexion logique et une vérification de transmission des données, avec retransmission en cas d.erreur (PAR, Positive Acknowledgment with Retransmission). L.établissement d.une connexion TCP s.effectue par un mécanisme dit de 3-way handshake. La machine source A désirant établir une connexion envoie un paquet TCP contenant le flag SYN et un numéro de séquence SN=x. La machine recevant la connexion B l.accepte ou non, suivant qu.il y ait ou non un processus prêt à répondre à la demande sur le port destination de la connexion. Dans le cas positif, B répond par SYN ACK, SN=y, et AN=x 1 (numéro d.acquittement). Sinon, A répond par un paquet contenant le flag d.interruption brutale RST. Lorsque A reçoit le paquet SYN ACK, A émet un paquet contenant le flag ACK et un acquittement AN=y 1. La connexion est alors établie, le transfert des données peut alors commencer. Les acquittements suivants doivent être effectués pour tous les octets transférés.

Format d'un paquet TCP bits 0 4 8 12 16 20 24 28 31 Port source Port destination Numéro de séquence Numéro d'acknowledgment Offset Reservé Flags Window Checksum Pointeur urgent Options Padding Données Flags : TH_SYN 0x02 Bit de Synchronisation TH_RST 0x04 Réinitialisation (Reset), interruption brutale de connexion TH_PUSH 0x08 Force le 'push' (délivre les données à l'application) TH_ACK 0x10 Acknowledgment TH_URG 0x20 Données urgentes 4.1.3 UDP UDP est le protocole sans connexion de la famille IP. Comme dans TCP, une notion de port4 source et de port destination est utilisée pour multiplexer/démultiplexer les transferts de données il est à noter qu.un port UDP n.a aucune relation avec un port TCP, même si les deux couvrent la même notion.
4

entre les applications d.émission et de réception. Dans le cas d.UDP, aucun établissement de connexion préliminaire n.est effectué, et aucun contrôle du transfert n.est directement assuré. Dans le cas de l.envoi d.un paquet vers un port non bindé5, c.est-à-dire un port sur lequel

aucune application n.écoute, un message ICMP « Port Unreachable » est renvoyé à l.expéditeur.

4.2 Recherche des machines actives : scan ICMP et TCP
Le pirate balaie les adresses IP trouvées dans la phase d.identification et, par adresse, envoie un packet ICMP « Echo Request ». Les machines actives (up) répondent par un paquet ICMP « Echo Reply ». Le protocole de messages et diagnostics ICMP est cependant souvent filtré, en raison des nombreuses fonctionnalités risquées qu.il présente (par exemple, ICMP « Redirect » permettant de reconfigurer une route). L.agresseur peut donc utiliser en alternative une méthode dite de ACK-scan. Il envoie un paquet TCP vers un service quelconque, contenant le bit ACK positionné dans les flags. Ce paquet ne correspondant à aucune connexion établie, la machine distante, si elle est active, répond par un paquet TCP contenant le flag RST de réinitialisation de connexion. Etant donné que certains filtres IP (firewalls) annoncent le refus des paquets par un message ICMP, le scan ICMP permet donc aussi de faire une découverte rudimentaire des règles de filtrage IP.

4.3 Port scanning TCP et UDP
L.idée de ce type de scan, le plus important et le plus répandu, est d.obtenir la liste des services en écoute sur les machines distantes. Ces services sont susceptibles d.être fournis par des applicatifs contenant des bugs de sécurité, tels que les bugs d.implémentation que nous décrirons dans la phase suivante. L.agresseur envoie donc, par IP (active) et par port intéressant, des paquets qui, suivant la réponse qu.ils génèrent, permettront de dire si le port est ouvert (en écoute) ou non. L.envoi d.un paquet vers un port donné est appelé probe vers ce port. La découverte et la publication d.un bug sur telle ou telle application serveur publique, comme récemment en janvier 2001 le serveur de noms DNS bind d.ISC entraîne très souvent des milliers de scans de pirates pour les ports correspondants aux applicatifs incriminés, le port 53 (TCP et UDP) dans le cas de bind. Les ports scannés sont donc principalement les ports associés à des services contenant des vulnérabilités connues, et non des ports libres. En particulier, les scans visent la plupart du temps les ports inférieurs à 1024, historiquement réservés aux services publics tels que DNS (53, en TCP et UDP), http (80 en TCP), portmap (111 en TCP), POP3 (110 en TCP). Le scanning de ports étant assez bruyant au niveau des enregistrements (logs) des firewalls ou des systèmes de détection **d.intrusion**, de nombreuses variations existent pour assurer la furtivité des scans. Nous distinguons en particulier :

4.3.1.1 TCP connect scan
Ce type de scan, le plus simple à implémenter car il ne nécessite aucune manipulation de

paquet, tente d.établir une connexion légitime avec le port visé de la machine distante. C.est aussi le scan le plus visible. venant de la primitive C bind() de l.API des sockets BSD, qui permet de « nommer » une socket et donc de s.attacher à un port donner dans le cas TCP/UDP.
5

4.3.1.2 UDP scan

L.unique technique de scanning UDP consiste à envoyer un paquet au port UDP visé et attendre un éventuel message de réponse ICMP « Port Unreachable ». Si ce message est reçu, le port est fermé ou filtré. Cependant, le scan UDP est peu fiable car il est nécessaire que l.ICMP soit autorisé en sortie. D.autre part, de nombreuses stacks IP empêchent l.envoi trop rapide de réponses ICMP (au maximum 80 messages toutes les 4 secondes sous Linux par exemple, technique dite de throttle).

4.3.1.3 TCP SYN Scan

Le SYN Scan consiste à réaliser seulement une partie du 3-way handshake de l.établissement d.une connexion TCP, d.où son autre nom de half-open scan. L.agresseur envoie un paquet contenant le flag SYN au port à tester de la machine cible. Si celle-ci répond par un paquet contenant les flags SYN ACK, alors le port est ouvert et un service y est disponible. Sinon (cas où la cible répond par RST), le port est fermé. Une fois l.état du port déterminé par la réponse de la cible, l.agresseur ne finalise pas l.établissement de la connexion par un ACK mais détruit la connexion au moyen d.un RST. La connexion n.apparaît donc pas dans les logs des services de surveillance orientés connexion, tels que les tcpwrappers. Un tel scan est par contre détectable assez facilement par un système de détection **d.intrusion** en surveillant les paquets SYN.

4.3.1.4 TCP FIN Scan, NULL Scan, XMAS Scan6

D.après le RFC 793, la couche TCP/IP d.un système d.exploitation doit répondre par RST à tout paquet adressé à un port fermé, tandis qu.elle doit ignorer et ne pas répondre aux paquets non SYN adressés aux ports ouverts. L.idée des scans FIN, NULL et XMAS est d.utiliser des paquets contenant respectivement le flag FIN, aucun flag, les flags FIN URG et PUSH pour vérifier cette condition. Cependant quelques systèmes ne suivent pas les spécifications posées dans le RFC, et ces scans s.avèrent inopérants sur les plate-formes Microsoft (Windows NT et 9x), Cisco IOS, BSDI, HP/UX, SGI Irix. Ce type de scan est très difficile à détecter pour un IDS réseau, car il lui faudrait mémoriser toutes les connexions afin de déterminer si le FIN correspond à une connexion réellement établie ou non. Les IDS réseau reconnaissent les scans FIN en remarquant les rafales de paquets FIN vers différents ports en un laps de temps très courts. Insérer un délai suffisant entre les probes permet d.éviter la détection de ces scans.

4.3.1.5 FTP bounce scanning

Le protocole FTP pose une communication à deux canaux pour le transfert de fichiers: une communication vers le port TCP 21 du serveur est établie par le client et sert de canal pour les commandes, un autre canal est établi à partir du port source 20 vers un port libre aléatoire côté client pour le transfert des données (listing des fichiers, envoi ou réception de fichiers) proprement dites. Ce mécanisme, outre les autres failles inhérentes de FTP (à savoir, passage en clair des identifiants et mots de passe), présente un risque de sécurité important car l.IP et le port vers lesquels le serveur effectuera la connexion de retour pour l.envoi des données sont spécifiés par le client, au moyen d.une commande PORT x,x,x,x,y,y (où x,x,x,x est l.adresse IP, octets séparés par des virgules et y,y sont les deux octets du numéro de port). Le client est donc libre de spécifier n.importe quelle IP et n.importe quel port. Si aucune vérification n.est effectuée côté serveur, celui-ci ira établir une connexion vers une machine tiers. Un agresseur peut ainsi effectuer un scan de ports au moyen de la commande PORT, répétée autant de fois que Description précise dans le manuel du scanner de ports nmap par Fyodor, http://www.insecure.org/

nécessaire. Les messages d.erreurs renvoyés par le serveur FTP indiquent que le port spécifié n.est pas ouvert ou est filtré. Un transfert effectué sans erreur indique au contraire que le port est ouvert. Même si la plupart des implémentations actuelles de serveurs FTP interdisent de telles commandes PORT, de nombreuses machines sur Internet contiennent encore des serveurs FTP affectés qui peuvent servir de relais pour des scans, sans même que l.agresseur ait besoin d.en prendre le contrôle (en se connectant au serveur en ftp anonyme, par exemple). En effet, la machine source du scan sera vue par la cible comme étant le serveur FTP. D.autre part, dans le cas de réseaux protégés par un filtre, la présence d.un serveur FTP vulnérable à une telle attaque peut être utilisée pour

passer le filtre avec les règles relatives à la machine hébergeant le serveur FTP.

4.3.2 Techniques avancées de scanning

Voyons ici quelques techniques mettant en défaut certains filtres IP (firewalls) et IDS.

4.3.2.1 Ports sources spéciaux

Comme nous venons de voir, le protocole FTP s.attend à ce que toute communication de données soit faite avec comme port source au niveau serveur le port 20, et comme destination un port quelconque côté client. Ceci est souvent pris en compte dans les règles de firewall en incluant une règle, extrêmement dangereuse, acceptant toute connexion entrante sur le réseau dont le port source est 20. Ce type de règle est à proscrire; il est très facile de choisir le port source d.une connexion et un firewall arborant une telle règle est équivalent à pas de firewall au niveau TCP. Des outils7 permettent même de positionner le port source de n.importe quelle

connexion à une valeur donnée, sans besoin de modifier les logiciels eux-mêmes. Un autre port source souvent accepté en entrée, cette fois-ci pour UDP, est le port 53, qui correspond à la source d.une requête DNS. De nombreuses configurations de firewalls incluent par facilité des règles telles que « accepter tout paquet vers notre serveur DNS ayant comme port source 53 », exposant ainsi le serveur entier à des attaques sur des services UDP autres que DNS.

4.3.2.2 Rebonds au travers de proxies applicatifs

De nombreuses organisations disposent de proxies applicatifs dans des buts de sécurité (socks, firewalls applicatifs tels que NAI Gauntlet) ou d.optimisation du trafic Internet (proxy-cache http). Ces proxies, lorsqu.ils sont mal configurés, peuvent servir de rebonds à un utilisateur extérieur pour attaquer le réseau interne ou un autre réseau de façon anonyme. Les proxies les plus recherchés par les pirates sont les proxies http8 et Wingate, souvent utilisables sans authentification.

4.3.2.3 Fragmentation IP

Une des fonctionnalités rendues par IP est la fragmentation des paquets pour les adapter aux contraintes de taille du support physique, le lien. Ainsi, il est possible de découper les paquets de telle façon que les informations nécessaires au firewall pour décider de l.acceptation ou non du paquet soient distribuées sur plusieurs fragments, en coupant par exemple l.en-tête TCP en deux. Les alternatives possibles pour le firewall sont alors l.acceptation/le refus sans examen du paquet ou l.attente des autres fragments pour pouvoir défragmenter le paquet et ainsi disposer de l.en-tête complet pour faire la décision de passage. Dans de nombreux cas (en particulier dans les filtres IP simples des routeurs, sur les réseaux à grand débit), le coût de cette nmap (http://www.insecure.org/) et ADMfzap (ftp://adm.freelsd.net/pub/ADM/) Les caches http tels que Microsoft Proxy Server, Squid, permettent aussi d.émettre des connexions non http, notamment au moyen des méthodes POST et CONNECT.
7 8

défragmentation est trop important et seuls les fragments d.offset 0 (censés contenir les informations de décision) sont inspectés, les autres sont par défaut acceptés.

4.3.2.4 Sémantique des fragments spéciaux

L.offset des fragments est en fait le rang (offset) du premier octet de données du fragment dans le paquet final réassemblé : En jouant sur l.offset, nous pouvons créer des fragments chevauchant les précédents et réécrivant des données d.autres fragments, telles que les en-têtes TCP ou UDP contenues dans le paquet IP. Comme précédemment, le filtre doit disposer de l.intégralité des fragments pour pouvoir appliquer les règles au paquet. De plus, le filtre ne sait pas à priori comment va réagir la machine destination à cette réécriture des paquets par fragments se chevauchant : ce

comportement dépend en effet de la stack IP de la machine distante, et quasiment tous les cas existent (réécriture des données en se conformant aux derniers fragments reçus, abandon des fragments se chevauchant, abandon pur et simple du paquet, .). Ce problème appartient à une classe plus importante dite de sémantique de la stack IP (impossibilité de prévoir la réaction de la stack IP à certains stimuli, sans en connaître l.implémentation), et a été popularisé comme moyen de

contourner certains IDS et filtres IP. Des outils9 sont disponibles pour implémenter ces attaques.

4.3.2.5 Bugs des stacks IP ou de firewalling, exemples

Linux IPCHAINS10 La couche de firewalling des noyaux Linux < 2.2.11 contient un bug découvert en juillet 1999 par Thomas Lopatic de Data Protect AG, permettant, sous certaines conditions, de procéder à la réécriture du port dans l.en-tête TCP ou UDP du paquet. Il est ainsi possible de faire accepter toute connexion vers un port arbitraire d.une machine protégée par un firewall Linux affecté et pour laquelle il existe au moins un port non filtré. La vulnérabilité est dans le fait que le noyau considère un fragment d.offset égal à 0 trop petit pour contenir toute l.en-tête de transport (TCP ou UDP) comme un fragment d.offset supérieur à 0. Le scénario d.attaque est alors le suivant : Fragrouter, http://www.anzen.com/research/nidsbench/ 10
http://www.dataprotect.com/ipchains/
9

En-tête IP (0) A B C D 0 20 40 60 20 80 Offset de fragmentation En-tête IP (0) A En-tête IP (20) B En-tête IP (40) C En-tête IP (60) D

20 20 1. L.agresseur lance un fragment vers la machine protégée contenant le flag IP_MF, un offset de 0 et une en-tête TCP avec un port non filtré, 2. Il lance ensuite un fragment d.offset 0 d.une longueur de 4 octets, contenant le port filtré à contacter. En vertu du bug, ce fragment sera traité comme un fragment d.offset supérieur à 0 et les données viendront recouvrir le port non filtré du premier fragment. 3. Il termine le paquet par un fragment sans flag IP_MF et contenant le reste des données du paquet.

Inspection des réponses de serveurs FTP protégés

Afin de permettre à des clients externes de se connecter à un serveur FTP protégé par un firewall, de nombreux firewalls proposent une solution d.inspection du contenu de chaque paquet de réponse du serveur. Lorsqu.un tel paquet commence par la chaîne .227 . (correspondant à une acceptation de la commande PASV d.ouverture d.un canal de données symétrique à la commande PORT), le firewall extrait l.adresse IP et le numéro de port spécifié dans ce paquet et ouvre un « trou » temporaire dans le firewall pour laisser passer cette connexion du client vers l.adresse en question, normalement le serveur FTP lui-même. En forçant le serveur FTP à répondre par une chaîne .227 . adéquate, il est possible d.ouvrir une connexion TCP vers n.importe quel port de n.importe quelle machine protégée. Cette réponse peut être obtenue en envoyant une commande similaire à .xxx227 Entering Passive Mode (192,168,1,10,192,3). où a,b,c,d est l.adresse IP de la cible et e,f le port désiré. Le serveur répondra par un message d.erreur similaire à : « Unknown command : xxx227 Entering Passive Mode (192,168,1,10,192,3) ». En jouant sur l.option MSS de la connexion TCP, qui spécifie la taille maximum des segments de données transmis, il est facile de couper le flux pour que 227 soit au début d.un paquet. De nombreux firewalls IP sont affectés par cette vulnérabilité, découverte en février 2000 par John Mc Donald de Data Protect AG11 : Checkpoint FW1 4.0 et 3.0, Cisco PIX 5.0 et 4.x, Linux ipchains, .

4.3.3 Autres techniques de découverte de la topologie

4.3.3.1 Détection des systèmes d.exploitation

Le nombre de stacks IP actuellement disponibles implique autant de comportements différents pour le traitement des paquets spéciaux, des erreurs, . Cette constatation est la base de l.identification distante des systèmes d.exploitation, compte tenu de leurs réactions à certains paquets de test : réaction à l.envoi d.un paquet FIN (Microsoft Windows répond par RST alors que le RFC 793 recommande de ne pas répondre), incrément des numéros de séquences initiaux, . La granularité obtenue actuellement dans certains scanners est importante, allant même jusqu.à l.identification des patchs installés en plus de la version du système d.exploitation.

4.3.3.2 Firewalking

Cette technique, dérivée de traceroute, consiste à envoyer des paquets IP dont on incrémente successivement le TTL. Les réponses reçues nous donnent toutes les passerelles entre notre source et la machine destination. Le firewalking consiste aussi à faire varier les protocoles véhiculés par les paquets IP de test pour être par exemple acceptés par un firewall, la décision d.acceptation se faisant

souvent sur le protocole de transport (TCP ou UDP vers le port donné d.une IP donnée). 11http://www.dataprotect.com/,

http://www.securityfocus.com/frames/?content=/vdb/bottom.html?vid=979 Le pirate est ainsi capable d.obtenir une cartographie précise d.une zone protégée par un firewall, et de déterminer les règles d.accès de ce firewall.

4.3.3.3 Analyse applicative des ports ouverts

Afin d.optimiser les chances de succès de l.intrusion, l.agresseur détermine les versions précises des services ouverts qu.il a pu trouver grâce au scanning. Pour se faire, il corrèle les résultats de la découverte du système d.exploitation (méthodes précédentes) avec des analyses niveau application des bannières présentes sur les ports ouverts. En effet, les services répondent souvent aux clients qui se connectent par une bannière d.identification, donnant type et version du service. Par exemple, la connexion au port 21 d.une machine donne le résultat suivant : [root@secway /] telnet alyssa 21 Trying 192.168.1.18... Connected to alyssa. Escape character is '^]'. 220 alyssa FTP server (SunOS 4.1) ready. La bannière affichée indique non seulement un serveur FTP, mais aussi la version du système d.exploitation (ici, une machine Sparc SunOS 4.1). De la même manière, si aucune bannière n.est affichée, nous pouvons déduire ces informations de la réponse du serveur à une requête classique. [root@secway /]# telnet hera 8123 Trying 192.168.1.14... Connected to hera. Escape character is '^]'. GET / HTTP/1.0 HTTP/1.1 200 OK Date: Tue, 06 Mar 2001 18:02:21 GMT Server: Apache/1.3.12 Ben-SSL/1.40 (Unix) PHP/4.0.1pl2 FrontPage/4.0.4.3 Last-Modified: Mon, 20 Sep 1999 15:25:36 GMT ETag: "555cf-f08-37e651f0"

Accept-Ranges: bytes Content-Length: 3848 Connection: close !content-type:! text/html Le port 8123 du serveur hera est dans ce cas un serveur Web Apache 1.3.12 sur une machine Unix. Il dispose de plus des extensions Frontpage et SSL. La technique de mail-bounce décrite dans la phase de recherche environnementale peut aussi être reprise pour obtenir les versions des serveurs Mail utilisés. Certains services ont pour fonction première de fournir des informations. C.est le cas de SNMP, finger. Un agresseur peut obtenir des MIB standards d.un serveur SNMP (souvent accessibles en lecture seule avec des noms de communautés par défaut tels que « public », « cisco ») des données comme : le nombre et la nature des interfaces réseaux de l.équipement, la version du système, le nombre d.utilisateurs connectés, les adresses ARP des cartes, .

5. Exploitation

Le pirate a collecté suffisamment d.information pour être capable de déterminer le point le plus faible du réseau, celui sur lequel l.attaque sera la plus discrète et la plus optimale possible. La phase d.exploitation est le tournant de l.intrusion, celle à partir de laquelle l.agresseur ne peut plus reculer : autant la phase de scanning peut-elle être considérée comme une simple « recherche d.information », l.exploitation montre un engagement de l.agresseur dans une action offensive hostile. Cette phase est généralement très brève car elle se résume à l.utilisation d.un bug pour prendre le contrôle d.un système. Le pirate passe ensuite dans une phase de progression si le système compromis lui convient (d.un point de vue emplacement par exemple) pour mener à bien son intrusion, ou sinon recommence la phase d.exploitation sur un autre système : c.est dans cette phase que le caractère opportuniste de l.attaque dont nous avons parlé précédemment est le mieux mis en lumière. Détaillons maintenant les principales méthodes pour gagner l.accès à un système cible. Ces attaques peuvent être menées contre des services filtrés en utilisant les mêmes techniques de contournement que celles vues précédemment pour le scanning.

5.1 Classe des attaques par usurpation d.identité

L.authentification, en tant que reconnaissance fiable d.un utilisateur par certains critères secrets tels que les mots de passe, est la principale défense pour protéger l.accès à un système. C.est donc aussi la porte d.entrée la plus souvent forcée, les systèmes d.authentification déployés actuellement utilisant des concepts beaucoup trop faibles pour les techniques d.attaques modernes.

5.1.1 Découverte d.informations d.authentification par essais

La technique la plus classique d.usurpation d.identité est celle dite de « brute force » des

informations d.authentification d.utilisateurs donnés. Le pirate essaie des combinaisons de logins/mots de passe (dans le cas d.un système d.authentification par mot de passe) jusqu.à trouver un couple valide. Cette méthode, certaine mais délicate car facilement identifiable et potentiellement très longue, peut être facilitée par de nombreux moyens : les informations découvertes dans la phase d.identification de la cible ou de scanning, telles que les noms d.utilisateurs ou e-mails internes valides peuvent par exemple donner des logins valides. Les mots de passe à essayer sont tirés d.un dictionnaire, dérivés du login ou dérivés de dictionnaires adaptés (par exemple bâtis à partir de tous les mots présents sur le site Web de la cible).

5.1.2 Abus de relations de confiance

Un système d.information est inévitablement construit autour de relations de confiance qu.ont les différents éléments entre eux. Typiquement, un serveur d.authentification centralisé est par excellence l.élément de confiance (on dit « trusté ») du système d.information. Il a été historiquement très commun d.établir des relations de confiance entre machines pour les authentifications auprès de services comme les r- services BSD (rlogin/rsh/rexec). De même, l.authentification auprès de NFS est faite par la machine client : pouvoir changer son identifiant Unix sur la machine client revient à pouvoir accéder aux fichiers NFS des autres utilisateurs, le contrôle de l.UID étant effectué côté client. Nous mettons ainsi en évidence une méthode d.usurpation d.identité mettant en jeu les relations de confiance entre machines. L.exemple le plus connu12 de ce type d.attaques est celui du TCP Spoofing contre les serveurs rsh. Les serveurs rsh permettent une authentification par relations de confiance entre machines sur le modèle suivant : un client situé sur une machine A peut accéder sans demande de mot de passe à la machine serveur rsh B si celle-ci contient un fichier de configuration ($HOME/.rhosts) établissant une relation de confiance avec A. rsh étant implémenté au dessus de TCP, qui se charge de la négociation d.une véritable connexion logique entre les deux machines, ceci n.aurait pas été un véritable problème sans certains défauts de conception des stacks IP en ce qui concerne la génération des numéros de séquence initiaux (ISN) TCP. Rappelons que la machine B, en réponse à une demande de connexion de A, renvoie à A un paquet contenant un acquittement du numéro de séquence envoyé par A, ainsi qu.un autre numéro de séquence y, supposé aléatoire. A va alors répondre en acquittant y, et la connexion est ainsi établie. Dans le cas normal, le fait que seul A connaît le numéro de séquence y retourné par B garanti que c.est bien A à l.autre bout du flux. Cependant, la quasi-totalité des stacks IP d.il y a une dizaine d.années incrémentaient les ISN par pas de 64k. Il était donc facile d.immobiliser A pour ne pas qu.elle réponde (déni de service sur A), de prévoir le numéro y retourné par B, l.acquitter à la place de A et ainsi établir une connexion fictive « à l.aveugle »

entre A et B. Le pirate exécute par ce biais des commandes telles que echo >> ~/.rhosts lui permettant d.accéder à la cible par rsh, directement de sa machine d.attaque. Des attaques similaires sont possibles sur NFS, encore plus faciles au niveau transport car le protocole utilisé est UDP (donc pas d.établissement de connexion logique). Le pirate réalise une opération de montage fictive du partage NFS visé vers un système trusté non actif, ouvre un fichier (par exemple un ~/.rhosts) et y écrit ce qu.il souhaite. Dans ce cas, le seul facteur à maîtriser est le filehandle NFS (numéro de fichier NFS attribué aléatoirement à l.ouverture du fichier). Finalement, bien que cela soit fortement déconseillé, DNS est souvent utilisé pour la construction de relations de confiance : au lieu de préciser l.adresse IP, seul identifiant unique d.une machine, il est fréquent d.utiliser le nom DNS de la machine trustée. De la même manière, la compromission du serveur DNS (compromission directe ou pollution des caches DNS, prédiction des identifiants de requêtes) entraîne la compromission de l.autre serveur.

5.1.3 Interception d'informations d'authentification

Nous avons vu que toute machine « proche » topologiquement de la cible est elle-même une cible potentiellement intéressante. En effet, prendre le contrôle d.une des machines du prestataire Internet permet par exemple d.inspecter le trafic réseau de l.organisation, de l.enregistrer et souvent d.intercepter des informations d.authentification (mots de passe) circulant en clair sur le réseau. Même si la mode est au chiffrement des données pour leur assurer confidentialité et intégrité, de

nombreux protocoles véhiculent toujours leurs données, dont l.authentification, en clair. C.est le cas des services de récupération de courrier POP, IMAP, des services FTP, telnet, rlogin et de nombreux autres. De nombreux outils, appelés sniffers, sont disponibles pour capturer du trafic réseau à des fins d.analyse et de diagnostic. Certains sont spécifiquement écrits pour capturer des mots de passe : pcs13, dsniff14. Ces sniffers dédiés aux pirates peuvent être équipés de fonctionnalités évoluées comme le chiffrement des données capturées, le renvoi en temps réel de ces données vers d.autres serveurs, .. car popularisé par le fameux pirate Kevin Mitnick. 13 PCS par Halflife, http://packetstorm.securify.com/Exploit_Code_Archive/pcs.tgz 14 DSNIFF par Dug Song, http://www.m!onkey!.org/~dugsong/dsniff/
12

L.exemple suivant montre un fichier produit par le sniffer pcs. On remarque qu.une connexion ftp entre les machines intra et gaia a été interceptée et enregistrée. Nous pouvons y voir le login (bob) et le mot de passe (m0nP4sS) de l.utilisateur qui s.est connecté. --PATH: intra.Secway-int.net(2611) => gaia.Secway-int.net(ftp)

DATE: Wed Mar 7 03:55:25 2001 USER bob PASS m0nP4sS SYST PASV LIST CWD /tmp PASV LIST QUIT [CLOSED] La démocratisation des solutions de chiffrement telles que SSH (remplacement de telnet) ou SSL (http et beaucoup d.autres protocoles) a poussé les pirates a trouver des attaques plus subtiles pour intercepter les informations d.authentification. Aussi, l.attaque la plus fréquente15 dans le cadre de l.interception de données chiffrées est l.attaque dite man-in-the-middle. Nombre de solutions de chiffrement telles que SSH ou SSL reposent sur un chiffrement asymétrique, à base de clef publique/clef privée pour échanger une clef symétrique dite de session. Un client A qui souhaite se connecter au serveur B lui envoie sa clef publique, et réciproquement. Un pirate s.étant introduit entre A et B peut intercepter la connexion, se faire passer pour B et envoyer sa propre clef publique à A. Celui-ci dialoguera avec le pirate, pensant dialoguer avec B. En retour, le pirate dialoguera avec B et enverra les réponses de B à A. Cette attaque est possible si l.utilisateur ne vérifie pas que la clef publique qu.il a reçue est bien celle de B (cette distribution et certification des clefs est la problématique que tentent de régler les PKI).

5.1.4 Détournement de flux existants

Une variation de l.attaque précédente consiste à se placer sur le chemin d.une connexion entre A et B et y insérer des données qui seront reconnues comme valides venant de l.autre par la machine qui les reçoit. Cette possibilité est cependant subordonnée à une vulnérabilité dans le système de chiffrement et de vérification, telle que celle découverte en 1998 dans SSH16. Cette méthode est évidemment aussi applicable, et encore plus facilement, aux flux non chiffrés grâce à des outils comme Hunt17. Le pirate a même la possibilité de prendre le contrôle complet du flux existant, par une technique dite de Hijacking, où l.agresseur vient se synchroniser sur la connexion existante en reprenant ses paramètres (paramètres TCP tels que AN/SN/Window/.) et en la coupant du client légitime. Le « cassage » du chiffrement est hors de portée de la plupart des pirates, sauf cas précis où les informations de chiffrement ont été exposées (compromission des clefs privées ou des certificats).
15

16 17

ssh insertion attack, Core SDI, http://www.core-sdi.com/soft/ssh/ssh-advisory.txt Hunt 1.5 par Pavel Krauz, http://lin.fsid.cvut.cz/~kra/index.html#HUNT

5.2 Exploitation de fautes d.implémentation

La grande majorité des incidents récents sont à déplorer du fait de bugs dans les logiciels permettant d.en prendre le contrôle à distance, ou d.élever ses privilèges en local. Les fautes

d.implémentation ont toujours été un des problèmes majeurs de la sécurité, car impossibles à éradiquer complètement. Souvent, seuls des correctifs venant du constructeur (patchs) sont en mesure de résoudre le problème. Les fautes d.implémentation sont tellement importantes et

courantes que de nombreux pirates analysent eux-même les codes sources de services réseaux largement utilisés, ou désassemblent les applications commerciales afin d.y trouver des erreurs d.implémentation qu.ils pourront utiliser pour manipuler le serveur distant. Il en résulte qu.un nombre non négligeable de programmes destinés à tirer parti de ces failles (appelés exploits ou warez) est rendu public chaque semaine, et qu.un « marché » parallèle est organisé entre groupes pirates pour la diffusion restreinte d.exploits non publics donc plus efficaces (0-day). La découverte et la correction de ce type de failles se fait le plus souvent par leur découverte par des particuliers amateurs ainsi que dans les laboratoires de recherche de sociétés de sécurité, ou par la découverte d.exploits circulant dans les milieux undergrounds18. Les applications développées en interne, telles que les scripts CGI, sont aussi susceptibles d.être affectées par de tels bugs, et l.analyse empirique de ces programmes est souvent l.alternative la plus facile qu.il reste au pirate quand les failles connues sont corrigées.

5.2.1 Buffer overflows

La famille des buffer overflows a été décrite comme « faille de la décennie », et l.une des premières utilisations connues de ces bugs a été le fameux « Morris Worm » qui paralysa Internet fin 1988. C.est en fait une famille de failles, due à une erreur de code extrêmement commune consistant à ne pas vérifier la taille de zones mémoire lors de la copie/lecture/. de ces zones. En effet, de nombreux langages comme le C, pour des raisons d.optimisation et de souplesse, ne vérifient pas les tailles des zones mémoires et laissent l.utilisateur libre d.accéder à des octets hors des zones allouées : le programmeur se doit de vérifier lui-même qu.il ne tente pas de lire ou d.écrire à des adresses non attribuées. Les possibilités d.exploitation de ce problème sont nombreuses. En particulier, la technique des buffer overflows la plus répandue consiste à utiliser la non-vérification des tailles de chaînes de caractères d.un tableau à un autre pour écraser les octets qui suivent le tableau de destination en mémoire. Le pirate, en exploitant une telle vulnérabilité, peut écraser des données intéressantes en mémoire, telles que d.autres variables (un UID Unix sauvegardé, un nom de fichier, .) et détourner le flux normal du programme. Une variable couramment écrasée pour exploiter un programme vulnérable est l.adresse de retour des fonctions. Sur la plupart des

architectures et des compilateurs19, lors de l.appel à une fonction, l.adresse de retour de cette fonction est stockée en mémoire dans la pile d.exécution du programme. Modifier cette valeur en mémoire permet donc de contrôler où le programme sera aiguillé une fois la fonction terminée : typiquement, le pirate détourne le flux du programme vers une zone mémoire contenant un morceau de code dit shellcode, qui exécute une commande. De telles failles dans les services réseaux tels que les serveurs de mail ou les serveurs FTP/HTTP/. sont découvertes régulièrement et permettent de prendre le contrôle de la machine affectée, en détournant le flux du serveur et lui faisant exécuter par exemple un shell.

à noter à ce propos la conférence de Marcus J. Ranum (NFR) au salon BlackHat 2000 : « Full disclosure and Open Source » http://www.blackhat.com/html/bh-multi-media-archives.html 19 des attaques de type buffers overflows ont été démontrées pour quasiment toutes les platesformes actuelles.
18

Un des exemples les plus récents de buffer overflow a été celui découvert par Covert Labs de Network Associates20 dans le serveur de noms Bind 8.2 d.ISC, permettant de prendre le contrôle d.une machine par l.intermédiaire de Bind, et pour lequel des exploits circulent.

5.2.2 Attaques format string

Une nouvelle classe de failles récemment découverte (fin 1999) impliquent la modification d.octets en mémoire par le biais de chaînes de formatage. Lorsqu.un programme écrit en C génère un message contenant des variables (par exemple un message d.erreur destiné à être affiché ou enregistré dans un fichier), il formate l.apparence de ce message par des caractères spéciaux (formateurs) tels que %s, %d, .. Certains de ces caractères spéciaux (%n) permettent d.écrire en mémoire. Si l.utilisateur peut injecter des formateurs dans le message, il peut construire un message

qui ira réécrire l.adresse de retour (comme dans le cas précédent des buffer overflows) ou toute autre zone mémoire. Ce type d.attaques implique la modification de variables en mémoire et est donc souvent assimilé à la famille des buffer overflows.

5.2.3 Métacaractères shell et caractères spéciaux

De nombreux scripts CGI écrits en Perl ou en script shell Unix font appels à des commandes externes Unix, comme par exemple /usr/bin/sendmail pour envoyer un mail. La ligne de commande est potentiellement construite avec des paramètres rentrés par l.utilisateur dans le formulaire. Si aucune vérification de cette entrée n.est effectuée (telle que, filtrer les caractères non alphabétiques), des caractères spéciaux dits métacaractères peuvent être injectés pour exécuter des commandes. Ces caractères spéciaux sont en particulier les séparateurs Unix tels que ;, &, |, ( ), `, 0xff, .. Par extension, ce type de faille affecte aussi les pages faisant appels à des bases de données.

L.insertion de caractères spéciaux SQL permet d.imbriquer une requête SQL dans la requête construite à partir de l.entrée utilisateur : L.instruction SQL SELECT * FROM users WHERE user='$form.user' AND password='$form.password' devient SELECT * FROM users WHERE user='' ; INSERT INTO users VALUES ('hacked','hacked') ;' AND password=''

5.2.4 Virus, chevaux de Troie

Dans le cas d.une attaque directe inefficace, le pirate envoie un virus ou cheval de Troie à une des adresses e-mail découvertes dans la phase de recherche environnementale. Cette attaque nécessite l.acceptation du message par les filtres antivirus et l.exécution du programme par la victime (même si certains clients mails comme Outlook ont eu des bugs permettant l.exécution automatique de scripts à la réception du message ou à son ouverture). L.agresseur, pour plus de fiabilité, développe son propre virus ou cheval de Troie. Il ne sera ainsi pas reconnu par les antivirus à base de signatures. Le cheval de Troie récupère les mots de passe et les envoie au pirate, ou établit une connexion à travers le firewall vers une machine extérieure qu.il contrôle (technique de reverse-shell). Un des exemples récents les plus connus de telles attaques est la compromission de l.intranet Microsoft par une variante du virus QaZ, rendue publique en octobre 2000.
20

http://www.pgp.com/research/covert/advisories/047.asp

5.3 Elévation des privilèges

L.accès initial à une machine compromise par un pirate est souvent un accès non privilégié. Il s.attache alors à élever ses privilèges vers les droits root sous Unix ou Administrateur sous Windows NT. Les techniques employées pour ce faire sont les mêmes que celles utilisées précédemment pour gagner le premier accès. La méthode la plus courante est l.utilisation d.un buffer overflow sur un programme local setuid root sous Unix, ou des permissions déficientes dans les fichiers (C:) et la base des registres sous Windows NT.

6. Progression

Le pirate doit dans un premier temps nettoyer toute trace de son intrusion puis ensuite s.installer pour revenir plus facilement sur le système compromis. Une fois la passerelle vers le système d.information pénétré établie, il progresse rapidement ou lentement, furtivement, pour atteindre le but qui le motive : vol d.informations, destruction de systèmes de production, .. 6.1 Inspection du système compromis Les solutions de sécurisation des fichiers systèmes et journaux sont maintenant faciles à déployer, et de nombreuses organisations ont recours à des systèmes de détection d.intrusion dits « host-based » pour stopper à temps les pirates.

L.agresseur ayant gagné le contrôle d.une machine examine donc celle-ci afin d.y déceler tout d.abord la présence d.un administrateur système qui pourrait noter le comportement suspect du pirate. L.activité des utilisateurs eux-mêmes est aussi un bon moyen de se fondre dans le système pour ne pas être repéré. Il utilise donc les utilitaires Unix standards tels que who, whodo, last pour vérifier la fréquence de login des administrateurs, netstat pour inspecter les éventuelles connexions non répertoriées dans les fichiers journaux, .. Une fois le facteur humain éliminé, car c.est celui qui

peut réagir le plus rapidement et qu.il est impossible de modifier, il examine les processus actifs ainsi que le système de fichiers pour trouver d.éventuels systèmes de surveillance ou IDS. Il utilise pour ce faire les outils standards tels que ps, et examine les fichiers ouverts (avec l.utilitaire lsof) et les derniers fichiers modifiés du disque, probablement des fichiers journaux (/var/adm, /var/log, .). Il inspecte les fichiers d.exécution programmée tels que crontab ou at afin de déterminer la fréquence des backups et la potentialité que ses outils d.intrusion ou les fichiers journaux contenant des traces aient été sauvegardés dans les backups précédents. Il détermine alors si des traces de l.attaque ont été enregistrées et s.il lui est possible d.effacer ces traces sans en créer de nouvelles (cas des systèmes de surveillance des fichiers comme Tripwire21). Il inspecte aussi les configurations des systèmes de journaux tels que syslogd afin de voir une éventuelle copie automatique des journaux vers d.autres machines (loghost dans /etc/hosts sous Unix). Les sessions shell étant communément enregistrées pour des raisons de commodité (possibilité de revenir à des commandes antérieures), il vérifie aussi que le shell qu.il utilise n.enregistre pas d.historique (~/.bash_history, ~/.sh_history).

6.1.1 Nettoyage du système

Le pirate dispose alors d.une information précise sur les moyens de protection et de surveillance en place sur la machine. Il décide alors s.il lui est possible de nettoyer les traces qu.il a laissé. Ce nettoyage n.est en effet pas systématique car il peut introduire des traces beaucoup plus importantes, notamment au niveau des systèmes de surveillance de fichiers comme Tripwire. Il cherche cependant à nettoyer au maximum, en compromettant ces systèmes (modification des bases de référence Tripwire si celles-ci sont stockées sur un support modifiable, ou backdoor kernel comme nous le verrons dans la partie suivante). L.enregistrement de session suivant montre un pirate nettoyant les traces de ses tentatives de connexion sur le serveur ftp d.une machine Solaris qu.il a par la suite réussi à compromettre :
21

http://www.tripwire.com/

cd /var/adm # ls –l messages -rw-r--r-- 1 root other 18244 Jan 7 12:29 messages # tail –1 messages Jan 7 12:29:52 solserv ftpd[10216]: bcomora (bogus) LOGIN FAILED [from

192.168.38.6] # grep -v FAILED messages > .yo ; mv .yo messages Il nettoie ensuite l.espace disque libre où les données de ses fichiers effacés sont toujours présentes. Si jamais le pirate ne peut pas effacer ses traces, il surveille régulièrement la machine afin de déterminer si les administrateurs se sont aperçus de son intrusion et pouvoir l.évacuer rapidement dans le cas positif.

6.1.2 Pose de backdoors

L.accès à un système par exploitation d.un des services est une tâche potentiellement difficile et surtout bruyante. Le pirate se crée donc ensuite des portes détournées, dites backdoors, qui lui permettront de revenir plus facilement sur le système. Les backdoors peuvent être des plus simples (remplacement du programme de login par un programme disposant d.un mot de passe générique) aux plus complexes, mettant en jeu des modifications du système d.exploitation même, pouvant masquer jusqu.à la présence des fichiers et des processus, voire de certains types de paquets réseau. Les backdoors fréquemment utilisées sont : - L.ajout de programmes d.apparence innocents introduisant des flux furtifs (« covert channels ») dissimulant des commandes dans des flux standards (requêtes Web vers l.extérieur, requêtes DNS ou ICMP, .). - Des modifications de programmes tels que sshd, le serveur de connexions sécurisées chiffrées. Le pirate installe une version modifiée du programme sshd qui lui permet de revenir en se connectant sans trace, et avec une connexion chiffrée, évitant ainsi les IDS réseau situés sur d.autres machines. - L.insertion de modules kernels, normalement utilisés pour rajouter des pilotes périphériques au vol, permet de modifier le noyau du système d.exploitation. Le pirate a alors un contrôle complet sur le système, donc en particulier sur les logiciels de surveillance, car ils font tous appels au système d.exploitation, censé être fiable. Certaines backdoors kernel publiées récemment ont même la possibilité d.intercepter des commandes envoyées par le pirate dans du trafic réseau bénin et de

masquer ce trafic aux sniffers en ne leur remontant pas les paquets22. Sur les systèmes ne disposant pas de support des modules kernels, l.agresseur recompile un nouveau noyau (/boot.img, /vmunix) ou modifie le noyau au vol par accès direct à la mémoire au travers des fichiers /dev/mem, /dev/kmem ou /proc. plaguez, .Weakening the Linux kernel., http://phrack.infonexus.com/search.phtml?view&article=p52-18
22

6.2 Prise d.information, progression

Suivant ce que le pirate cherche, il examine le comportement des utilisateurs afin d.identifier ses prochaines cibles.

Il détermine le rôle de la machine sur laquelle il est par son nom, les fichiers qui y sont stockés, le nombre et le type des utilisateurs. Il trouve les utilisateurs à suivre en regardant les fichiers de mots de passe (/etc/passwd et /etc/group), où les attributions de chacun sont souvent détaillées (par exemple : groupe des développeurs, groupe marketing, .). Les fichiers de profil (/etc/profile, ~/.profile) des utilisateurs lui donnent des informations intéressantes, comme par exemple les variables d.environnement des programmes tels que CVS dans le cas de l.attaque d.un laboratoire de recherche. CVS ou ClearCase, largement utilisés pour le contrôle des versions de projets, positionnent des variables d.environnement sur le poste client donnant le nom des serveurs de développement où sont stockées les sources. L.intrus cherche donc ces variables s.il s.attaque à des secrets industriels tels que des codes sources. Il inspecte aussi les historiques des shells utilisateurs, pour déterminer vers quels serveurs se connectent les développeurs. Finalement, le pirate réitère autant de fois que nécessaire les phases d.exploitation et de progression. Le rapatriement des informations capturées se fait par l.intermédiaire de backdoors de type « covert channel » .

7. Conclusion

Nous avons vu le déroulement typique d.une attaque sur Internet : une phase de recherche environnementale, où l.agresseur identifie sa cible, une phase de scanning, permettant de trouver des services potentiellement exploitables, l'intrusion proprement dite puis finalement la progression dans le système compromis. Les deux dernières phases sont ensuite reproduites sur les systèmes internes, en se servant de la première machine capturée comme point de départ. Cet exemple de déroulement, que l.on peut qualifier de tactique empirique, est valable quelle que soit la motivation du pirate et ses compétences : seuls les moyens mis en oeuvre et le raffinement technique des méthodes vont varier. L.établissement de lignes de défense adaptées passe par la compréhension de ces mécanismes, en particulier en ce qui concerne la détection des intrusions et la récupération d.un système après un incident.Nous avons fait un pas en avant pour comprendre comment les pirates opéraient, il reste à voir pour très bientot l'utilisation d'outils spécifique afin de mener à bien cette intrusion