MISC 11

LES TESTS D'INTRUSION EXTERNES

TESTS D'INTRUSION RESEAU, SYSTEME, APPLICATIFS

PATRICK CHAMBET
http://www.chambet.com

Les tests d'intrusion sont un sujet récurrent de la sécurité informatique : ils existent en théorie depuis toujours mais ne cessent en fait d'évoluer, en fonction des nouvelles familles de vulnérabilités, mais aussi en fonction des modes. Ils sont maintenant bien entrés dans les mœurs, aussi bien du côté des prestataires en sécurité informatique que des clients. Cependant, et même si tout le monde en propose désormais, il existe différents types de tests d'intrusion (voir le premier article de ce dossier) et également plusieurs niveaux de tests, avec des qualités très variables.

Nous allons étudier dans cet article les différentes phases d'un test d'intrusion technique externe, et nous montrerons à chaque fois quelques exemples tirés de prestations réelles, anonymisées, que nous avons réalisées.

Définition

Si vous avez lu le dossier de ce numéro depuis le début, vous savez déjà ce que sont les tests d'intrusion. En deux mots, un test d'intrusion consiste à se mettre dans la peau d'un attaquant externe et banalisé, disposant d'un certain degré de compétences, de ressources, de temps et de motivation pour pénétrer un système cible. En revanche, l'attaquant ne dispose au départ d'aucune connaissance particulière de la cible et commence donc « en aveugle ».

Nous n'aborderons pas dans cet article le thème du social engineering, et ce pour plusieurs raisons. Tout d'abord, des règles d'éthique et de déontologie couramment pratiquées parmi les professionnels des tests d'intrusion (voir charte de la FPTI [1]), nous interdisent de recourir à de tels procédés, car ils peuvent être sources de dérapages. Ensuite, ce genre de procédés est par trop facile : il existera toujours un moyen humain permettant d'outrepasser un dispositif technique. De nombreuses entreprises se vantant de procéder à des tests d'intrusion avec un taux de réussite de 100%, voire 110%, utilisent en fait du social engineering, ce qui facilite effectivement grandement les choses. Toute politique de sécurité qui se respecte comporte une grande partie de mesures organisationnelles, permettant de gérer les risques humains.

Nous nous focaliserons donc uniquement sur les tests d'intrusion techniques externes. En ce qui concerne les tests d'intrusion internes, la méthodologie reste identique mais le nombre de vulnérabilités potentielles est souvent plus important et les techniques d'attaque plus nombreuses (voir article dans ce dossier).

Ne se substituant pas aux audits (organisationnels et techniques) mais venant en complément, les tests d'intrusion constituent la mesure préventive ultime pour identifier les faiblesses et les vulnérabilités d'un système afin de les corriger, et ce, avant qu'un acte de malveillance réel, externe ou interne, n'ait lieu.

Le but des tests d'intrusion est donc la fourniture d'une série de recommandations pour corriger les vulnérabilités détectées et améliorer le niveau de sécurité de l'ensemble du système cible. Ces recommandations peuvent s'appliquer à :

- la conception de l'architecture,
- la robustesse des composants,
- la configuration des systèmes,
- les procédures d'exploitation et d'administration.

Concernant ce dernier point, un test d'intrusion peut servir également à évaluer la qualité de la surveillance réalisée par les administrateurs :

- Ont-ils été alertés à un moment ou à un autre, et si oui, comment ?
- Qu'ont-ils « vu » et comment ont-ils interprété les différentes phases des tests ?
- Comment ont-ils réagi ?

Prérequis

Quelques prérequis sont nécessaires pour se lancer dans les tests d'intrusion. Une bonne connaissance des réseaux est indispensable : principes de routage, protocoles IP (la lecture des RFC constitue pratiquement un passage obligé), architectures, etc... Une connaissance approfondie des systèmes cibles est également nécessaire (*NIX, Windows, Mac, …), ainsi qu'une bonne pratique des architectures applicatives et des applications actuelles (« 3-tier », serveurs Web, middlewares, bases de données, …).

Ensuite, il est nécessaire de se constituer une plate-forme d'intrusion, comprenant différents systèmes, réels ou virtuels (il est plus facile d'attaquer un Windows depuis un autre Windows par exemple) et une palette d'outils. Voici par exemple quelques outils que nous utilisons fréquemment : hping, nmap, netcat, un navigateur Web, un proxy HTTP intrusif, openssl, whisker, Brutus, L0phtcrack, des scripts Perl, un compilateur C pour les outils « maison », une base de vulnérabilités et d'« exploits » personnelle, etc… Des outils de tests automatisés, comme l'excellent Nessus par exemple, pourront être utilisés en complément, au cas où nous aurions oublié quelque chose.

Nous montrerons l'utilisation de certains de ces outils dans la suite de cet article. Mais il ne faut pas oublier le « nez » du testeur (l'instinct, diront certains), son expérience, son imagination et son inventivité. Il arrive souvent qu'une configuration non standard produise des effets imprévus. Une capacité à anticiper plusieurs coups d'avance, comme aux échecs, permet parfois de pénétrer un système encore plus en profondeur que ce qu'il semble possible de prime abord.

Et bien sûr, il vous faut… une autorisation de tests (voir l'article sur le droit et les tests d'intrusion, dans ce dossier) ! Nous voilà parés pour commencer nos tests.

DEROULEMENT DES TESTS

Les tests d'intrusion consistent en une démarche itérative comportant en général les phases suivantes :

1. Recueil d'informations sur la cible
2. Détection des systèmes et des services, cartographie
3. Recherche et exploitation de vulnérabilités réseau
4. Recherche et exploitation de vulnérabilités système
5. Recherche et exploitation de vulnérabilités applicatives
6. Progression

A l'issue de la dernière phase, on reboucle sur la phase 2 autant de fois que nécessaire, lorsque de nouveaux systèmes ont été détectés, ou lorsqu'il est possible d'effectuer un ou des rebonds de plus en plus profonds, par exemple.

Ce schéma de tests se distingue toutefois d'une attaque logique réelle [2] par les éléments suivants :

- Il n'y a pas de phase d'implantation de charge utile (rootkit, cheval de Troie, code hostile, bombe logique, etc…). Sur autorisation du client, on peut uploader certains outils sur un serveur compromis.
- A moins d'un accord du client, on ne perturbe pas le fonctionnement de la cible (donc pas de déni de service sans prévenir, par exemple, même pour faciliter une autre attaque).
- A moins d'une autorisation expresse du client, on ne modifie aucune donnée sur la cible, et en particulier, on n'efface pas nos traces dans les logs. Un simple recueil de preuves (copies d'écran par exemple) est effectué pour la rédaction du rapport final de la prestation.

Passons maintenant en revue chacune des phases ci-dessus.

RECUEIL D'INFORMATIONS SUR LA CIBLE

En général, le client ne nous communique qu'une liste d'adresses IP à tester, ou une adresse de sous-réseau (cas du test « en aveugle », le plus fréquent). Supposons par exemple que l'adresse IP qui nous a été communiquée par notre client *Yoopi, l'entreprise heureuse,* est l'adresse `123.123.123.123`. A nous de trouver les informations complémentaires nécessaires à nos tests. Pour cela, il existe de nombreux moyens.

LES BASES WHOIS

La base Whois RIPE répertorie tous les sous-réseaux et leurs propriétaires respectifs. Cela permet dans un premier temps de commencer par vérifier que les adresses IP communiquées correspondent bien au client qui a signé l'autorisation de tests, ce qui est fortement conseillé ! Pour interroger la base RIPE, vous pouvez utiliser l'URL suivante :

http://www.ripe.net/db/whois/whois.html

Il est intéressant de noter que ces recherches sont totalement furtives du point de vue de la cible : nous ne faisons aucune requête directe vers les adresses du client.

Vérifions que l'adresse 123.123.123.123 appartient bien à Yoopi :

```
inetnum:      123.123.123.96- 123.123.123.127
netname:      Yoopi
descr:        Yoopi is a great boite located in France
country:      FR
mnt-by:       PISTOLET-FR-MNT
changed:      je-gere@pistolet.net 20010802
changed:      tuveuxmesdoigts@pistolet.net 20010924
source:       RIPE

person:       Monsieur JOYEUX
address:      YOOPI
address:      4 RUE DU SOURIRE
address:      13010 MARSEILLE
phone:        +33 04 91 10 12 43
fax-no:       +33 04 91 10 13 04
e-mail:       joyeux@yoopi.com
admin-c:      MF2912-RIPE

person:       Jules-Edouard Gère
address:      PISTOLET – Super provider, surtout en Suisse
address:      60 rue des Colts
address:      75012 Paris - France
phone:        +33 01 70 99 58 99
fax-no:       +33 01 70 99 57 97
e-mail:       je-gere@pistolet.net
tech-c:       MP17435-RIPE
remarks:      For any complaint, please mail to "fischy@pistolet.net"
```

Nous identifions bien le client Yoopi, le sous-réseau qui lui est attribué (123.123.123.96/27), ainsi que d'autres éléments : l'hébergeur Pistolet, ainsi que les noms des contacts administratif (un employé de Yoopi) et technique (un employé de Pistolet). Ces informations peuvent être intéressantes pour l'attaquant, d'autant plus qu'il est possible de faire des recherches complémentaires dans les bases Whois sur ces éléments. En particulier, nous identifions un nom de domaine : `yoopi.com`. Une recherche dans les bases Whois sur ce domaine donne :

```
Domain Name: YOOPI.COM

Registrant:
YOOPI SA (NKZMKZBJDE)
   4 RUE DU SOURIRE
   MARSEILLE, FR 13010
   FR
```

```
   Administrative Contact:
      Monsieur Joyeux                          joyeux@yoopi.com
      Yoopi SA
      4 Rue du Sourire
      Marseille, FR 75010
      FR
      +33 04 91 10 12 43 fax: +33 04 91 10 13 04

   Technical Contact:
      J-E GERE                                 hostmaster@PISTOLET.NET
      PISTOLET France
      60, Rue des Colts
      Paris, FR 75012
      FR
      +33 01 70 99 58 99 fax: +33 01 70 99 57 97

   Domain servers in listed order:

   NS0.PISTOLET.COM           111.222.1.2
   NS1.PISTOLET.COM           111.222.1.3
```

Nous retrouvons les mêmes informations et vérifions bien la cohérence des enregistrements. Nous obtenons également les adresses des serveurs DNS du domaine yoopi.com (NS0.PISTOLET.COM et NS1.PISTOLET.COM), ce qui va nous permettre de passer à l'étape suivante.

LES BASES DNS

Si les DNS identifiés sont mutualisés, nous pouvons faire des requêtes sur ceux-ci en demeurant encore relativement furtifs du point de vue de la cible. Et si les DNS sont hébergés chez le client, on récupère parfois certains enregistrements de l'adressage interne… L'idéal est de récupérer l'intégralité des enregistrements DNS concernant le domaine yoopi.com, en effectuant un « transfert de zone » (AXFR) :

```
# host -l yoopi.com ns0.pistolet.net

yoopi.com SOA ns0.pistolet.net. dnsadmin.pistolet.net.
yoopi.com name server nsO.pistolet.net.
yoopi.com name server ns1.pistolet.net.
yoopi.com mail is handled by 15 hermes.yoopi.com.
joe.yoopi.com has address 123.123.123.101
jack.yoopi.com has address 123.123.123.102
william.yoopi.com has address 123.123.123.103
averell.yoopi.com has address 123.123.123.104
hermes.yoopi.com has address 123.123.123.122
www.yoopi.com has address 123.123.123.123
ftp.yoopi.com has address 123.123.123.124
```

Tous les serveurs enregistrés sont ainsi révélés : le serveur de mail, le serveur FTP, le serveur Web, ainsi que 4 serveurs dont la fonction sera à identifier.

Mais le transfert de zone est rarement autorisé, vu le nombre d'informations qu'il révèle. Si le transfert de zone ne fonctionne pas, essayons de récupérer certains enregistrements, en spécifiant des types de serveurs, par exemple :

```
# host –t MX yoopi.com ns0.pistolet.net

yoopi.com mail is handled by 10 hermes.yoopi.com.
```

La requête ci-dessus a permis d'identifier `hermes` comme serveur de messagerie. Vous pouvez utiliser d'autres types (NS, SOA, etc…) pour obtenir des informations supplémentaires.

Une autre méthode consiste à faire une recherche de type « dictionnaire » dans les bases DNS, afin d'identifier des enregistrements supplémentaires. Ainsi, dans l'exemple ci-dessus, la recherche de `ftp.yoopi.com` aurait renvoyé l'adresse 123.123.123.122. L'outil `dnsdigger` par

exemple permet d'automatiser cela, en partant d'un dictionnaire de noms courants (www, w3, www2, www3, etc…).

Enfin, un reverse lookup permet de partir des adresses IP et de rechercher les FQDN qui leur sont associés :

```
123.123.123.96    host.96.123.123.123.rev.pistolet.net.
123.123.123.97    host.97.123.123.123.rev.pistolet.net.
123.123.123.98    host.98.123.123.123.rev.pistolet.net.
(…)
123.123.123.127   host.127.123.123.123.rev.pistolet.net.
```

Dans le cas présent, des enregistrements génériques inverses sont définis dans les bases DNS, ce qui ne nous donne pas d'information supplémentaire.

MOTEURS DE RECHERCHE

Les moteurs de recherche permettent également d'obtenir de nombreux renseignements sur la cible, maintenant que nous connaissons le nom du client, le nom de certains employés, de certains serveurs, etc… Une recherche de type Google, y compris sur le site Web de Yoopi, permet d'obtenir le nom d'autres employés, de clients et de partenaires de Yoopi, etc… Des recherches dans les newsgroups, sur les forums, à partir des adresses e-mail des employés de Yoopi, fournissent également à un attaquant des informations précieuses, qui pourraient d'ailleurs être utilisées en social engineering. Il arrive couramment qu'un administrateur pose des questions techniques dans un newsgroup à propos de problèmes qu'il rencontre sur tel ou tel équipement, renseignant du même coup un attaquant sur le type de matériel utilisé dans l'entreprise et sur le niveau de compétence et de motivation de l'administrateur actuel. Toutes ces informations sont extrêmement précieuses, Sun Tzu l'avait compris bien avant l'invention de l'ordinateur... De plus, le fait de savoir qui correspond avec qui et à propos de quels sujets peut fournir des renseignements à un concurrent, par exemple. Mais nous abordons là les thèmes de l'intelligence économique et de la guerre de l'information [3], qui sont en-dehors du sujet de cet article.

Pour notre exemple, nous découvrons par exemple dans des logs publics que la machine 123.123.123.102 a été utilisée comme serveur CStrike il y a peu... Il est probable qu'elle ait été placée à l'époque devant le firewall de Yoopi, et cela se reproduira peut-être un jour. Nous avons aussi rencontré par exemple un poste de travail servant de client Kazaa et placé devant le firewall… Ce poste sans défense renfermait des informations intéressantes pour la suite des tests.

DETECTION DES SYSTEMES ET DES SERVICES, CARTOGRAPHIE

Nous allons maintenant établir la cartographie de la plate-forme cible. A ce stade, nous ne sommes plus furtifs, car nous allons effectuer des requêtes directement sur les systèmes cibles.

ROUTAGE

Pour déterminer les machines situées entre les serveurs cibles et nous, nous allons étudier le routage des paquets échangés. Pour cela, commençons par utiliser l'outil `traceroute` :

```
# traceroute 123.123.123.122
traceroute to hermes.youpi.com (123.123.123.122), 30 hops max, 38 byte packets
 1  POS-0-0.NRAUB101.Charlebourg.raei.francetelecom.net (194.51.159.25)  50.294 ms
49.270 ms  50.796 ms
 2  P12-1.ntaub201.Aubervilliers.francetelecom.net (193.251.126.230)  53.401 ms  52.496
ms  56.621 ms
 3  193.251.126.54 (193.251.126.54)  50.680 ms  50.413 ms  50.804 ms
(…)
12  ge0-0.gw1.lnd8.gbb.uk.uu.net (158.43.188.25)  59.541 ms  106.676 ms  116.761 ms
```

```
13  yoopi-gw.customer.PISTOLET.NET (111.222.10.10)  113.736 ms  60.858 ms  59.871 ms
14  * * *
15  * * *
```

Le traceroute ci-dessus est un traceroute UDP (méthode par défaut du traceroute Unix). Pour effectuer un traceroute ICMP, utilisez `traceroute –I`. Sur une plate-forme Windows, `tracert` utilise le protocole ICMP.

Dans notre cas, ni le traceroute UDP ni le traceroute ICMP ne permet d'atteindre le serveur cible (le serveur SMTP). Il semble qu'un firewall soit placé devant la cible et filtre nos paquets. Utilisons alors un traceroute TCP sur un port TCP ouvert. Il s'agit de la technique dite du « firewalking » :

```
# hping -S -p 25 123.123.123.122 -t 15
HPING 123.123.123.122 (eth0 123.123.123.122): S set, 40 headers + 0 data bytes
len=46 ip=123.123.123.122 ttl=103 DF id=32734 sport=25 flags=SA seq=0 win=8576 rtt=198.2
ms
```

Le serveur SMTP a bien reçu notre paquet SYN, et répond avec un paquet SYN-ACK (voir ci-dessous le paragraphe sur le scanning). Le serveur SMTP est donc bien situé au hop 15.

```
# hping -S -p 25 123.123.123.122 -t 13
HPING 123.123.123.122 (eth0 123.123.123.122): S set, 40 headers + 0 data bytes
TTL 0 during transit from ip=111.222.10.10 name= yoopi-gw.customer.PISTOLET.NET
```

Cette fois, le TTL ne permettant d'atteindre que le routeur Internet situé au hop 13, c'est bien celui-ci qui répond, avec un « TTL expired ».

De la même façon, essayons d'identifier l'équipement situé au hop 14 :

```
# hping -S -p 25 123.123.123.122 -t 14
HPING 123.123.123.122 (eth0 123.123.123.122): S set, 40 headers + 0 data bytes
TTL 0 during transit from ip=123.123.123.98 name=UNKNOWN
```

Bingo, il semble que la machine 123.123.123.98, qui a répondu, soit le firewall recherché, situé entre le routeur externe et le serveur SMTP.

En utilisant des techniques identiques et en recoupant avec les scans (voir plus loin), il est possible d'identifier l'adresse interne du routeur Internet (123.123.123.97), ainsi que, souvent, l'adresse interne du firewall (123.123.123.101). Les adresses 123.123.123.99 et 123.123.123.100 sont des adresses de sous-réseaux.

ROUTAGE SMTP

L'envoi d'un mail à une adresse inexistante du domaine provoque le retour d'un mail d'erreur contenant des informations intéressantes dans ses en-têtes.

- Informations sur le routage SMTP sortant :

```
Received: from hermes.yoopi.com (hermes.yoopi.com [123.123.123.122]) by edelweb.fr
(nospam/1.7); Tue, 28 Oct 2003 12:24:05 +0100 (MET)
Received: from 172.26.0.46 by hermes.youpi.com (InterScan E-Mail VirusWall NT); Tue, 28
Oct 2003 11:20:48 -0000
```

- Informations sur le routage SMTP entrant :

```
Received: from hermes.yoopi.com ([123.123.123.122]) by exch.yoopi.net with Microsoft
SMTPSVC(5.0.2195.4905);
         Wed, 29 Oct 2003 12:47:19 +0000
Received: from 212.234.46.16 by hermes.yoopi.com (InterScan E-Mail VirusWall NT); Wed, 29
Oct 2003 12:45:23 -0000
Received: from toto (localhost.edelweb.fr [127.0.0.1]) by edelweb.fr with SMTP id
NAA14304 for <pere-noel@yoopi.com>; Wed, 29 Oct 2003 13:48:30 +0100 (MET)
```

Nous en déduisons que le système de messagerie comprend une passerelle InterScan VirusWall NT sur `hermes.yoopi.com`, ainsi qu'un serveur Exchange 2000 SP3 (identifié par la version de son service SMTP : 5.0.2195.4905) sur le serveur interne `exch.yoopi.net`, ayant comme adresse interne 172.26.0.46.

De plus, la passerelle anti-virus laisse passer les exécutables en pièces attachées... Encore de précieuses informations pour la suite.

WEB BUG ET SERVEUR HOSTILE

Cette étape ne devra se faire qu'avec l'accord explicite du client, et en collaboration avec lui. Elle consiste à envoyer un message contenant un Web bug (une image invisible provoquant une connexion HTTP vers l'extérieur) ou une pièce attachée hostile, et/ou à lui demander de se connecter sur un serveur hostile spécialement préparé dans ce but, hébergé en général chez le testeur. Ces opérations permettent de recueillir d'avantage d'informations concernant l'architecture de la plate-forme du client (notamment l'adresse de son proxy HTTP).

Dans notre exemple, nous mettons par exemple en évidence le fait que le serveur `averell.yoopi.com` est le proxy sortant de Yoopi.

EXPLOITATION DES EN-TETES

En effectuant certaines requêtes HTTP sur les serveurs Web, par exemple, il est possible de provoquer le renvoi de certains en-têtes HTTP.

- Sur Apache :

```
HTTP/1.1 301 Moved Permanently
Date: Mon, 17 Nov 2003 11:39:22 GMT
Server: Apache/1.3.29 (Win32)
Location: http://srv-web/toto/
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

- Sur IIS :

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: https://172.26.0.40/Default.htm
Date: Mon, 17 Nov 2003 15:05:59 GMT
Content-Type: text/html
Accept-Ranges: bytes
Content-Length: 1704
```

Ces requêtes ont permis de révéler le nom de machine ou l'adresse interne des serveurs Web. Ces données sont intéressantes, car elles renseignent sur le plan d'adressage interne et sur l'architecture du réseau interne de la cible, permettant de ce fait de mener des attaques ultérieures basées sur ces éléments.

SCANS DE PORTS

Le scan de ports est la méthode de découverte la moins discrète, mais elle est le plus souvent indispensable. Elle consiste à balayer la cible afin de déterminer quels sont les ports ouverts (en écoute) pour chaque protocole supporté sur les machines. L'art du scan de ports mériterait un article complet à lui tout seul (voir par exemple [4]). En effet, il existe un grand nombre de méthodes pour mener à bien cette opération, qui semble pourtant évidente, et le succès de l'opération dépend en fait d'un grand nombre de facteurs.

Rappels simplifiés :

Lors de l'établissement d'une connexion TCP (« handshake »), plusieurs paquets sont échangés entre le client et le serveur :

1 – Le client envoie un paquet SYN sur un port TCP de la cible.

2 – Si la cible est inexistante ou située derrière un firewall qui droppe les paquets, aucun paquet n'est reçu en retour.

3 – Si la machine est accessible mais le port est fermé, la cible renvoie un paquet RST-ACK, ce qui met fin à la négociation. Si le port et ouvert, la cible renvoie un paquet SYN-ACK et attend.

4 – Le client renvoie alors un paquet ACK : la connexion TCP est alors établie.

En UDP, le concept de connexion n'existe pas. Lorsque le client envoie un paquet sur un port UDP :

1 – Si la cible est inexistante ou située derrière un firewall qui droppe les paquets, aucun paquet n'est reçu en échange.

2 – Si la machine est accessible mais le port est fermé, la cible renvoie un paquet ICMP « Port Unreachable ».

3 – Si le port est ouvert, la machine ne renvoie aucun paquet ICMP.

On voit donc que dans le cas de l'UDP, si aucun paquet ICMP n'est reçu, cela peut vouloir dire soit que le port est inexistant, soit qu'il est filtré, soit qu'il est ouvert, d'où ambiguïté. Des tests manuels complémentaires (envoi de paquets UDP « applicatifs » pour provoquer une réponse UDP, par exemple) sont nécessaires pour lever cette ambiguïté.

Nous allons utiliser ici une méthode classique de scan TCP : le « half-open » scan ou SYN scan. Nous supposerons qu'il n'y a de perte de paquets nulle part entre le réseau de test et la cible (ce qu'il convient toujours de vérifier, et ce, pendant toute la durée des scans). Le SYN scan consiste à envoyer uniquement un paquet SYN à la cible, et à ne jamais envoyer de ACK final. C'est pourquoi on appelle ce type de scan « half open ». C'est un scan relativement discret, car il ne laisse pas de trace dans les logs applicatifs, contrairement au « connect scan », qui établit complètement la connexion TCP.

Utilisons par exemple nmap sur le firewall de Yoopi. La nouvelle option `-sV` de nmap permet d'effectuer une identification du service tournant sur un port ouvert :

```
# nmap -vv -sS -sV -P0 -p 1-65535 -oN nmap-TCP-result.txt 123.123.123.98

nmap 3.48 scan initiated Mon Nov 24 11:45:25 2003 as: nmap -vv -sS -sV -P0 -p 1-65535 -oN
nmap-TCP-result.txt 123.123.123.98
Interesting ports on 123.123.123.98:
(The 65534 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE VERSION
264/tcp   open  bgmp?
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi
:
SF-Port264-TCP:V=3.48%D=11/24%Time=3FC21A15%r(GenericLines,4,"Q\0\0\0")%r(
SF:Help,4,"Q\0\0\0");
```

Nmap a trouvé un port TCP ouvert : le port 264. Au cours de l'identification du service tournant sur le port, la cible a renvoyé 4 octets : un « Q » suivi de 4 octets nuls (\0). Cela correspond en fait au service FW1_Topo de Firewall-1.

En utilisant nmap sur tout le sous-réseau, nous détectons bien le serveur de mail et le serveur Web parmi les autres adresses IP :

```
Interesting ports on 123.123.123.122:
PORT    STATE    SERVICE
25/tcp  open     smtp

Interesting ports on 123.123.123.123:
PORT    STATE    SERVICE
80/tcp  open     http
```

IDENTIFICATION DES SYSTEMES

Nmap possède deux options, une documentée (`-O`) et une non documentée (`--osscan_guess`), permettant d'effectuer une identification de l'OS cible par fingerprinting. Reprenons l'exemple du routeur externe de Yoopi :

```
# nmap -vv -sS -sV -P0 -O --osscan_guess -p 1-65535 123.123.123.97

nmap 3.48 scan initiated Mon Nov 24 11:44:56 2003 as: nmap -vv -sS -sV -P0 -O --
osscan_guess -p 1-65535 123.123.123.97
Interesting ports on 123.123.123.97:
(The 65529 ports scanned but not shown below are in state: closed)
PORT     STATE    SERVICE      VERSION
79/tcp   open     finger?
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
1433/tcp filtered ms-sql-s
3306/tcp filtered mysql
Device type: broadband router|terminal server|router
Running: Cisco IOS 12.X|11.X, Cisco embedded
OS details: Cisco 827 aDSL router running IOS 112.2(11), Cisco AS5200 terminal server,
Cisco 2501/5260/5300 terminal server IOS 11.3
.6(T1), Cisco IOS 11.3 - 12.0(11)
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=320C%IPID=Z%TS=U)
T1(Resp=Y%DF=N%W=1020%ACK=S++%Flags=AS%Ops=M)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=N%W=1020%ACK=S++%Flags=AS%Ops=M)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(Resp=N)

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=12812 (Worthy challenge)
TCP ISN Seq. Numbers: B33B4044 B33EE686 B3427722 B3468B82 B34A488D
IPID Sequence Generation: All zeros
```

Nmap identifie ici avec certitude un routeur Cisco (IOS 12.x). Nous reviendrons sur les ports filtrés sur le routeur plus loin.

En plus de nmap, il existe d'autres outils d'identification, comme Xprobe, par exemple. Pour identifier notre firewall, nous pouvons utiliser l'outil ike-scan, qui va faire du fingerprinting sur le service ISAKMP tournant sur le port UDP 500 du firewall (utilisé pour les connexions VPN) :

```
# ike-scan -v -o 123.123.123.98
Starting ike-scan 1.2 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
123.123.123.98   IKE Main Mode Handshake returned (1 transforms)
---     Removing host entry 1 (123.123.123.98) - Received 84 bytes

IKE Backoff Patterns:

IP Address      No.     Recv time               Delta Time
123.123.123.98   1       1070528489.900139       0.000000
123.123.123.98   2       1070528491.928154       2.028015
123.123.123.98   3       1070528494.010194       2.082040
123.123.123.98   4       1070528495.957137       1.946943
123.123.123.98   5       1070528497.964220       2.007083
123.123.123.98   6       1070528499.974061       2.009841
123.123.123.98   7       1070528501.996129       2.022068
123.123.123.98   8       1070528505.993926       3.997797
```

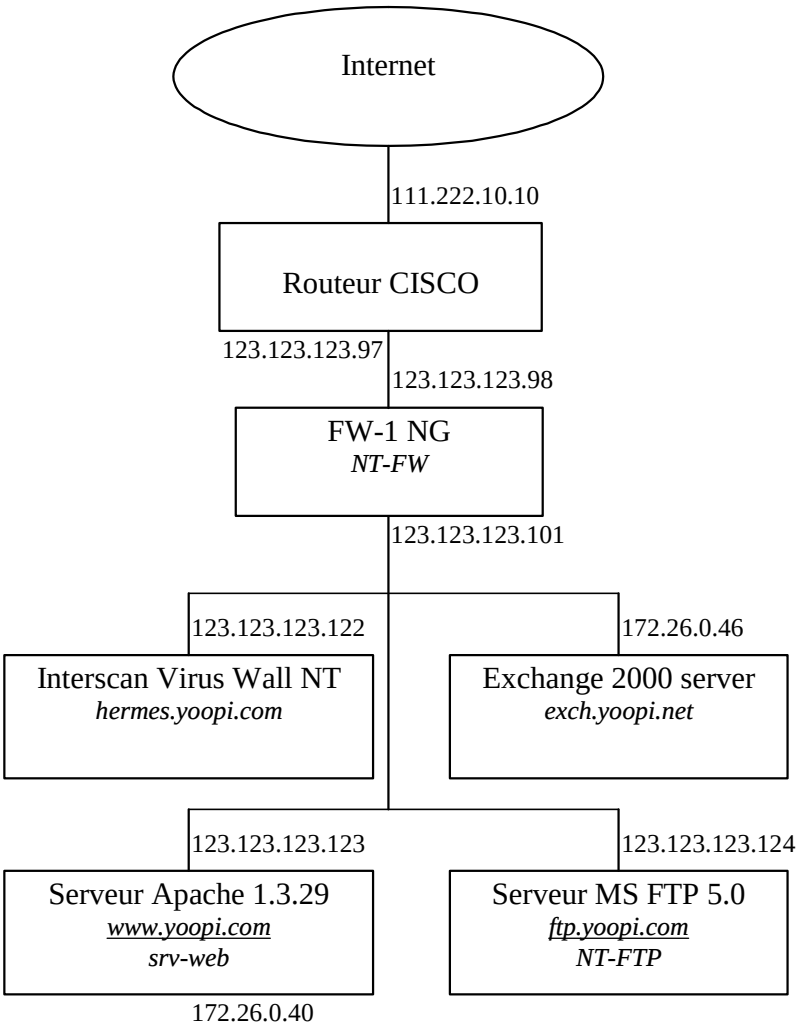
```
123.123.123.98   9       1070528510.012207       4.018281
123.123.123.98   10      1070528513.993244       3.981037
123.123.123.98   11      1070528518.013153       4.019909
123.123.123.98   12      1070528522.003191       3.990038
123.123.123.98   Implementation guess: Firewall-1 4.1/NG

Ending ike-scan 1.2: 1 hosts scanned.  1 returned handshake; 0 returned notify
```

Ike-scan identifie un Firewall-1 1.4.1/NG sur la machine 123.123.123.98. Un client Securemote nous permet de plus de nous connecter sur le port TCP 264 du Firewall-1. Nous obtenons ainsi la bannière suivante : `CN=NT-FW VPN Certificate,O=nt-fw..q5xqwz`

Cette chaîne indique le nom de la machine hébergeant le Firewall-1 : `NT-FW`.

Enfin, bien sûr, l'analyse des bannières renvoyées par certains démons permet d'identifier les services et les OS tournant sur les machines (serveurs Web, FTP, telnet, …).

L'architecture identifiée de Yoopi est donc la suivante :



Maintenant, soit vous êtes de la NSA ou du MOSAD, soit vous utilisez votre 0-day personnel pour Firewall-1, soit vous passez à la suite…

RECHERCHE ET EXPLOITATION DE VULNERABILITES RESEAU

REGLES DE FILTRAGE

Il est important de tenter d'identifier les règles de filtrage configurées sur le routeur et sur le firewall afin d'essayer ensuite de les outrepasser. Le résultat des scans du routeur et du firewall, ainsi que des machines situées derrière, indiquent quels sont les ports filtrés ou non filtrés. Il est possible d'en déduire les règles de filtrage définies au niveau des équipements.

Par exemple, les ports suivants ont été relevés par nmap sur le routeur :

```
79/tcp   open     finger?
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
1433/tcp filtered ms-sql-s
3306/tcp filtered mysql

67/udp   open  dhcpserver?
123/udp  open  ntp?
135/tcp  open  msrpc?
137/tcp  open  netbios-ns?
138/tcp  open  netbios-dgm?
161/udp  open  snmp?
1434/tcp open  ms-sql-m?
```

On peut donc en déduire, en TCP, qu'un service finger tourne sur le routeur, et qu'un certain nombre de ports TC,P considérés comme sensibles, car fréquemment utilisés par les vers récents (ports correspondant à MS RPC, NetBIOS, MS SQL Server et mySQL), sont filtrés.

En UDP, ce n'est plus aussi facile, car si aucun paquet ICMP « Port Unreachable » n'est reçu, cela peut vouloir dire soit que le port est inexistant, soit qu'il est filtré, soit qu'il est ouvert, d'où ambiguïté. Des tests manuels complémentaires (envoi de paquets UDP « applicatifs » pour provoquer une réponse UDP, par exemple) sont nécessaires pour lever cette ambiguïté. Ainsi, s'il est possible de vérifier que des services DHCP, NTP et SMTP écoutent effectivement sur les ports UDP 67, 123 et 161, il semble logique de penser que les ports UDP correspondant, là encore, à MS RPC, NetBIOS et SQL Server, soient en fait filtrés, pour les mêmes raisons que ci-dessus.

Au niveau du firewall, nous avons vu que les règles de filtrage en entrée doivent être sensiblement égales à :

Action	**Source**	**Destination**	**Service**
Permit	Any	hermes.yoopi.com	SMTP
Permit	Any	ftp.yoopi.com	FTP
Permit	Any	www.yoopi.com	HTTP
Permit	Any	FW-1	FW1_Topo
Permit	Any	FW-1	ISAKMP
Drop	Any	Any	Any

Il est parfois difficile de distinguer les règles de filtrage du routeur de celles du firewall. Un ensemble de règles globales, synthèse des règles définies sur les deux équipements, sera alors établi.

TENTATIVES D'OUTREPASSEMENT DES EQUIPEMENTS DE FILTRAGE

Remarque : A partir de ce stade, si des failles critiques sont découvertes, il est indispensable d'avertir immédiatement le client, car une attaque réelle pourrait survenir à tout moment.

Un très grand nombre de méthodes ont été inventées pour tenter d'outrepasser les règles de filtrage définies sur les routeurs et les firewalls. L'invention des attaquants et des testeurs semble sans limite. Mais au fur et à mesure que de nouvelles techniques apparaissent, les éditeurs de firewalls trouvent une parade qu'ils incluent à leur produit. Aujourd'hui, avec la plupart des bons firewall stateful du marché, il est extrêmement difficile de tromper les modules de filtrage.

Pour information, voici les principales techniques qui ont permis d'outrepasser les firewalls par le passé :

- Connexion avec des ports sources connus (DNS, HTTP, …)
- Attaques DNS (DNS spoofing, DNS poisoning, …)
- Utilisation d'options TCP non standard (exemple : flags SYN-FIN)
- IP spoofing
- Fragmentation de paquets (tiny fragments, fragment overlapping, …)
- TCP session hijacking
- ARP spoofing

Pour plus d'informations sur ces techniques, reportez-vous à MISC N° 0 [5].

Les techniques ci-dessus conservent tout leur intérêt une fois le firewall externe franchi : si on parvient à prendre la main sur une machine interne, ces techniques permettront d'avancer plus avant à l'intérieur du réseau, car elles sont extrêmement efficaces sur un LAN. Il est donc intéressant de les connaître.

Enfin, une autre méthode pour explorer le réseau interne est l'utilisation de proxies applicatifs mal configurés. Si le proxy HTTP sortant du client, par exemple, accepte les connexions depuis l'extérieur, vous pouvez vous connecter sur celui-ci et effectuer une requête vers les serveurs internes (nous avons rencontré cela il y a peu chez un client important…). Il est parfois possible, sur les serveurs Wingate, MS Proxy Server et Squid, d'effectuer des requêtes au moyen des méthodes GET et CONNECT, par exemple.

RECHERCHE ET EXPLOITATION DE VULNERABILITES SYSTEME

Nous incluons également ici certains services réseau de « bas niveau » et les protocoles les plus courants d'Internet (SMTP, POP3, HTTP, SSL, SSH, etc…).

Une fois les OS et services tournant sur les systèmes cibles parfaitement identifiés (voir « Identification des systèmes »), y compris leur version et niveau de patches exacts, il est possible de déterminer quelles sont les vulnérabilités potentielles dont ils risquent de souffrir. Nous pouvons également utiliser un scanner de vulnérabilités (comme Nessus par exemple). Ce type d'application est très utile mais a ses limites. En effet, un tel scanner peut remonter de fausses alertes (faux positifs) ou, inversement, ne pas détecter certaines vulnérabilités (faux négatifs, plus grave). Il pourra néanmoins compléter notre liste de vulnérabilités au cas ou nous en aurions oublié.

Les vulnérabilités système proviennent tout d'abord d'erreurs de programmation et d'implémentation. Depuis de nombreuses années, trois grandes familles de vulnérabilités de ce type ont été découvertes :

1. Les buffer overflows (stack overflows) [6]. Le ver de Morris utilisait déjà cette vulnérabilité pour se propager en 1988 !

2. Les format strings (1999) [7]
3. Les heap overflows (malloc() et double free()) (2001) [8]

La description de ces types de vulnérabilités et des méthodes d'exploitation associées a déjà été faite mainte fois, et vous pouvez vous reporter aux références pour plus d'informations. Vous remarquerez au passage que la contribution des Français est considérable (format strings ET heap overflows !). Et de nouvelles familles vont certainement être découvertes à l'avenir…

La récupération des « exploits » relatifs aux vulnérabilités détectées sur la cible peut se faire par différents moyens :

- Lecture des avertissements de sécurité (CERT, CVE, BID, etc…).

Ces publications expliquent en général sommairement la nature des vulnérabilités, et ne contiennent presque plus actuellement de code source d'exploitation. Il faut donc souvent, à partir des quelques informations techniques publiées, recréer complètement l'exploit.

- Suivi des listes de diffusion, newsgroups, channels IRC sur la sécurité.

La veille de ces sources, moins officielles, permet souvent de recueillir un grand nombre de renseignements précieux, le plus souvent techniques. Il arrive aussi que du code source, complet ou partiel, soit diffusé. Nous pouvons alors partir de cette base pour élaborer notre exploit.

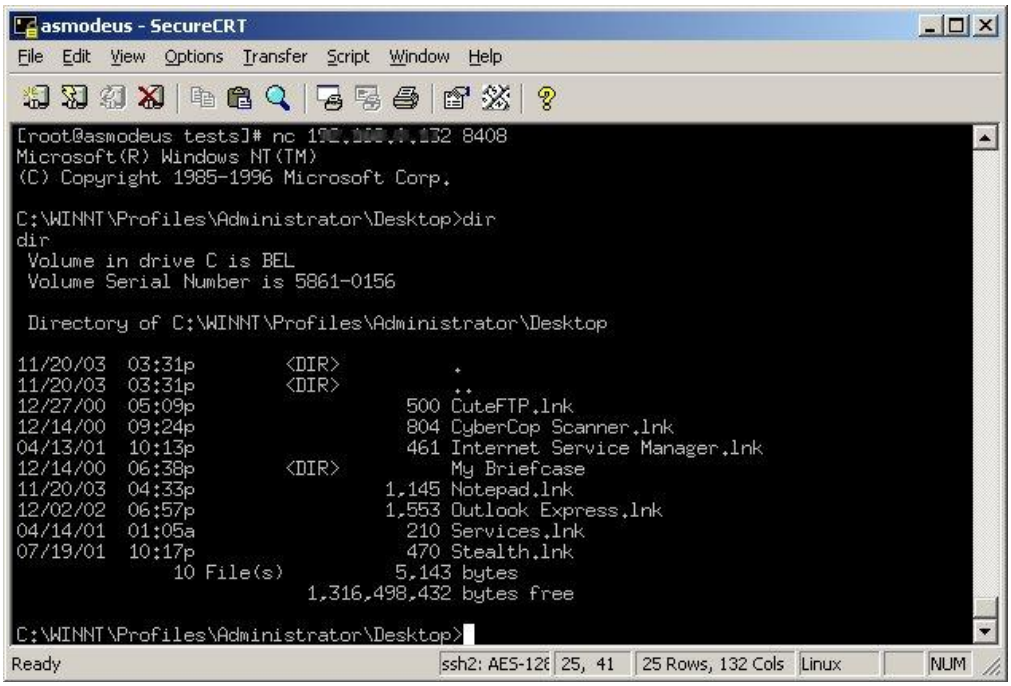
- Analyse des correctifs distribués.

Lorsque aucune information concernant une vulnérabilité n'a pu être récoltée, il reste la possibilité d'analyser le correctif pour en déduire la nature de l'attaque originale.

- Recherches personnelles.

Attention cependant, car la plupart des buffer overflows, par exemple, provoquent un déni de service du service visé, voire de l'OS dans son ensemble, si le shellcode ne produit pas l'effet escompté. Il est donc nécessaire de prévenir le client avant de tenter une exploitation de vulnérabilité qui risque de produire un déni de service sur un serveur critique ou qui prend du temps à redémarrer.

Par exemple, les vulnérabilités récentes « Apache Chunk Encoding » et « IIS WebDAV » provoquaient le plantage de serveur Web, voire un remote shell (à l'aide du shellcode adéquat) :



SSH et openssl ont aussi été des cibles privilégiées ces dernières années, et rien ne dit que ce soit terminé…

Que l'attaquant ait obtenu un shell ou qu'il puisse seulement exécuter des commandes sur le serveur distant, l'objectif de l'attaquant est maintenant d'uploader ses outils sur le serveur (sur autorisation du client uniquement), afin d'élever ses privilèges sur celui-ci et de passer root/Administrateur/SYSTEM, pour prendre le contrôle total du serveur. Il pourra alors faire ce qu'il veut avec les autres services tournant sur celui-ci, et éventuellement rebondir vers d'autres machines, de plus en plus en profondeur sur le réseau du client.

Pour finir, les attaques en force brute sur les services réseau (telnet, SSH, HTTP Basic Authentication, SNMP, …) peuvent aussi être incluses dans les attaques système. Une attaque réussie de ce type, en cas de mot de passe trop faible et/ou présent dans un dictionnaire, peut permettre d'obtenir de manière très simple le contrôle de certains services ou du serveur dans son ensemble. Dans le cas de SNMP, si le nom de communauté est découvert, de nombreuses informations critiques sur le réseau interne peuvent être dévoilées (type d'équipement, interfaces, adresses IP, table de routage, connexions actives et ports utilisés, processus tournant sur l'équipement, utilisateurs logués, etc…).

Passons maintenant aux applications hébergées sur les systèmes cibles.

RECHERCHE ET EXPLOITATION DE VULNERABILITES APPLICATIVES

Par expérience, actuellement, ce sont les vulnérabilités de ce type qui constituent la majeure partie des vulnérabilités que nous découvrons et exploitons. Les vulnérabilités réseau sont rares avec les firewalls modernes, et même les vulnérabilités système sont bien moins nombreuses que les vulnérabilités applicatives.

La raison en est simple : les applications métier tournant sur les serveurs des clients ont été réalisées soit en interne, soit par un prestataire externe, et par des développeurs qui ont rarement de formation au développement sécurisé. En général, ils n'imaginent même pas combien il est facile d'exploiter leur code pour faire faire à leur application une tâche qui n'était

pas prévue. Il nous est couramment arrivé de voir la mâchoire du client et de ses développeurs dégringoler sur la table de réunion lors de la réunion de restitution des résultats de nos tests (voir dernier paragraphe). Une fois les plus émotifs réanimés, nous leur expliquons les grandes catégories de vulnérabilités applicatives.

Prenons l'exemple d'une application Web. Dans ce cas, un attaquant peut a priori envoyer tout ce qu'il veut au serveur Web sur les ports HTTP et/ou HTTPS. Le firewall, s'il ne possède pas de module applicatif de type reverse proxy, n'effectue aucun contrôle et ne voit rien de ce qui circule sur ces ports.

Remarque : contrairement à une idée reçue très répandue, l'utilisation de SSL ne suffit pas à protéger une application Web. Le chiffrement SSL (sans utilisation de certificats clients X.509) ne concerne que la confidentialité, et ne protège pas des **intrusions**.

Les différentes sortes d'attaques sur les applications Web sont les suivantes (pour plus d'informations, reportez-vous à [9]) :

- **Interprétation des URLs**

Il s'agit d'envoyer des URLs malformées au serveur Web afin de le faire accéder à des zones non autorisées du serveur, ou de lui faire exécuter des commandes non prévues. Il est intéressant aussi d'essayer d'afficher le code source des pages de scripts (ASP, JSP, PHP, …), qui peuvent révéler des mots de passe, par exemple.

- **Mauvais contrôle des données entrées par l'utilisateur**

L'utilisation de certains caractères spéciaux dans les saisies peut conduire à l'exécution de commandes hostiles sur le serveur.

- **Injection de code SQL**

L'injection SQL peut être une conséquence directe d'un mauvais contrôle des données entrées par l'utilisateur. En effet, les caractères « ' » et « ; » peuvent être utilisés pour enchaîner plusieurs requêtes SQL à la suite et ainsi provoquer l'affichage de certaines données, ou plus couramment le contournement de certaines étapes d'authentification ou l'insertion dans la base de données de données corrompues (nouveau login/mot de passe par exemple). Il est même possible de scanner les serveurs internes situés autour du serveur de base de données !

- **Attaques sur les identifiants de session**

Une attaque classique consiste à voler la session d'un utilisateur qui vient de s'authentifier sur un système en essayant de deviner la valeur de son identifiant de session. Si la valeur de celui-ci est découverte, par une étude statistique par exemple, un attaquant peut alors utiliser l'application Web en lieu et place de l'utilisateur légitime.

- **Cross Site Scripting**

Le principe du Cross Site Scripting (ou XSS) est d'attaquer les utilisateurs de l'application plutôt que l'application elle-même. Pour cela, l'attaquant provoque l'envoi à la victime, par le site Web légitime, d'une page hostile contenant des scripts hostiles ou des composants malveillants. Cette page est exécutée sur le poste de la victime, dans le contexte du site Web d'origine (Internet, sites de confiance, …), et dans le contexte de sécurité de l'utilisateur courant. Il est ainsi possible de récupérer l'identifiant de session d'un utilisateur, par exemple.

- **Autres attaques**

Par exemple :

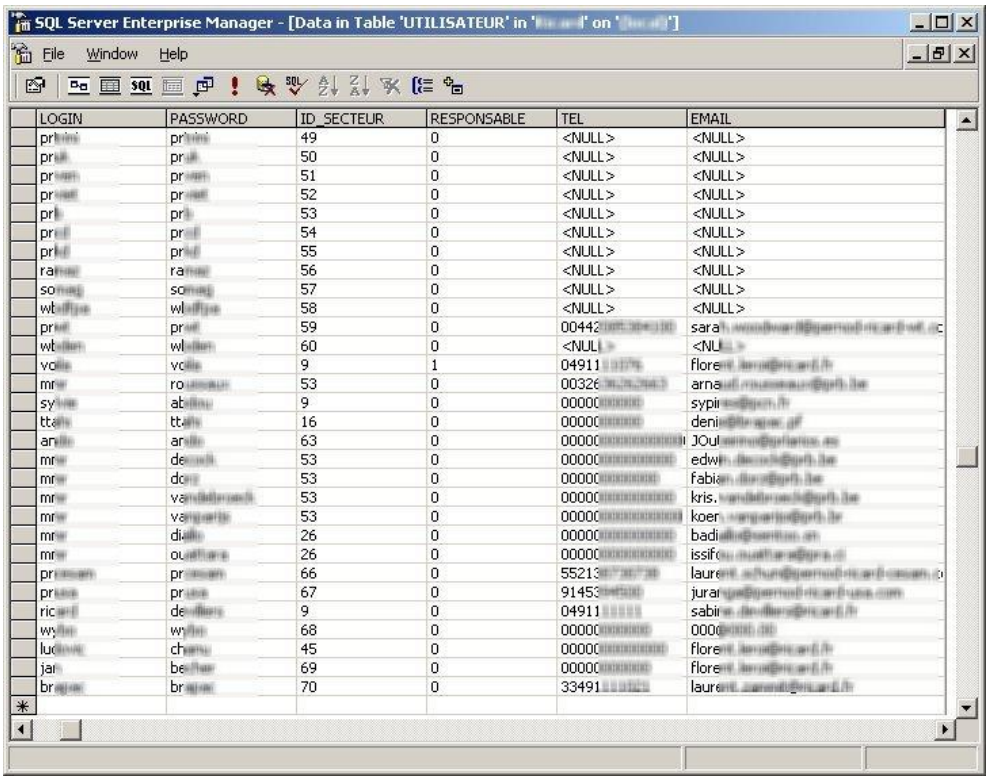
- Mécanismes d'authentification basés sur Java, JavaScript ou ActiveX
- Contrôle d'accès basé sur le header HTTP_REFERER
- Manque de ré-authentification
- Mauvaise gestion du contexte utilisateur
- Man in the middle

- Attaques côté client.

Reprenons par exemple notre exemple de Yoopi, et lançons l'outil `whisker` sur le serveur Web www.yoopi.com. Whisker effectue une recherche des répertoires et fichiers existants sur le serveur Web, à l'aide d'une base de connaissance. Whisker détecte par exemple les répertoires suivants :

```
/include
/lib
/manual
/php
/cgi-bin
/cgi-bin/test-cgi
/admin
/yoopi
```

Le répertoire /yoopi contient les pages PHP de l'application Extranet de Yoopi. Le répertoire /admin est protégé par une authentification applicative. Mais le répertoire /php attire immédiatement notre attention : il semble que PHP soit installé en tant que CGI sur le serveur Apache. Grave erreur : cela permet de fournir n'importe quel chemin de fichier sur le serveur à PHP, qui va en afficher le contenu ! De plus, il est possible sous certaines conditions d'exécuter des commandes PHP sur le serveur. A l'aide de ces deux possibilités, il est par exemple possible de télécharger la base de comptes (SAM) du serveur, d'uploader des outils sur celui-ci, puis de récupérer par exemple la base de données contenant les comptes utilisateurs de l'application, y compris le compte d'administration.



Il nous suffit alors de nous connecter à l'URL http://www.yoopi.com/admin pour avoir le contrôle total de l'application Extranet et du serveur qui l'héberge.

Game over… Sauf si l'on veut aller encore plus profond, bien sûr !

PROGRESSION

Une fois le contrôle d'un équipement obtenu, le but du testeur sera de tenter d'effectuer un rebond vers un autre équipement afin de progresser dans la DMZ, voire sur le réseau interne du client.

Pour nous aider dans ces rebonds, nous pouvons utiliser toutes les informations disponibles sur le système local (celui dont nous avons le contrôle). Ainsi, sur un serveur Windows par exemple, il est possible de récupérer la Browse List (gérée par le service Browser), par exemple : celle-ci contient toutes les machines présentes dans le voisinage réseau du serveur. Si celui-ci se trouve dans un domaine, nous pouvons également identifier quelle est la machine qui est le PDC de ce domaine, afin de tenter de récupérer la base des utilisateurs du domaine.

Si aucune défense en profondeur n'est prévue (segmentation du réseau interne en zones de confiance, par exemple), nous nous retrouvons quasiment dans les conditions d'un test interne.

AUTRES ATTAQUES

D'autres attaques peuvent cibler des équipements de différentes natures, comme les PABX des entreprises, par exemple, mais aussi les postes de travail des utilisateurs.

LES PABX

Il apparaît que, de plus en plus, les PABX sont pilotés depuis des consoles d'administration tournant sur Windows. Nous avons rencontré des consoles d'administration faisant tourner un serveur IIS installé par défaut afin de permettre l'administration distante du PABX. Nous avons également rencontré une console d'administration installée sur un Windows 95 ayant la racine de son disque dur partagée pour tout le monde… Et il s'agissait là d'une configuration standard effectuée par l'installateur lui-même. Cela fait froid dans le dos, d'autant plus que le réseau sur lequel est installé le PABX et sa console est le plus souvent accessible depuis le reste du réseau de l'entreprise.

Bref, les installateurs de PABX, comme les développeurs, ne sont pas formés à la sécurité, et un audit et/ou un test du PABX et de son environnement réseau/système/applicatif semble indispensable actuellement.

LES POSTES DE TRAVAIL

Actuellement, vu la difficulté d'exploitation des infrastructures réseau, mais aussi des systèmes, la tendance actuelle des attaquants est de cibler le poste de travail de l'utilisateur et l'utilisateur lui-même. C'est en effet le maillon faible de la chaîne, celui qui est le moins formé à la sécurité.

Des tests d'intrusion spécifiques, sous forme de serveur Web hostile ou de mails hostiles, par exemple, permettent de tester la robustesse de la configuration des postes de travail, ainsi que les réactions des utilisateurs face à une page Web contenant des éléments dangereux ou face à un message comportant une pièce attachée inconnue.

En fonction des résultats, le client pourra ainsi faire évoluer la configuration des « masters » de ses postes de travail (Voir MISC No 1 : [10]) ainsi que sa politique de sécurité « humaine ».

DERNIERE ETAPE

Les tests d'intrusion se terminent par la rédaction d'un rapport détaillé. A la différence des tests automatisés dont le rapport est en général généré automatiquement également, il ne reste jamais

de faux positif dans un rapport de tests d'intrusion manuels.

Le rapport contient une description précise de la visibilité de la plate-forme du client vis-à-vis de l'extérieur, une liste des vulnérabilités identifiées et cataloguées par criticité, chacune accompagnée par les mesures à prendre pour la corriger. Enfin, les risques résiduels éventuels sont présentés et qualifiés.

Une réunion de restitution des résultats clôt en général la prestation et permet au client de demander des explications complémentaires concernant les points qu'il considère comme les plus importants.

LES TESTS RECURRENTS

L'intérêt des tests récurrents repose sur le fait que le niveau de sécurité d'une plate-forme est quelque chose qui évolue dans le temps. Un test d'intrusion en fait une évaluation à un moment donné. Mais lors des modifications et des mises à jour des systèmes et des applications, lors de l'ajout d'un nouveau serveur, lors de la découverte de nouvelles vulnérabilités, le niveau de sécurité de l'ensemble varie. Il est donc important d'évaluer régulièrement ce niveau afin d'en suivre l'évolution dans le temps et de toujours vérifier qu'il est supérieur au minimum admissible par le client.

CONCLUSION

Les tests d'intrusion sont donc bien nécessaires, notamment avant la mise en ligne d'une nouvelle plate-forme Internet, car ils sont le dernier moyen de s'assurer de son niveau de sécurité vis-à-vis de l'extérieur. Venant en complément des audits classiques, ils permettent notamment de vérifier que l'on n'a rien oublié.

Nous avons observé que les clients faisant procéder à des tests d'intrusion de manière régulière et mettant à chaque fois en oeuvre les recommandations qui en découlent présentent un niveau de sécurité croissant à chaque test, et finissent par atteindre un niveau de sécurité excellent, bien au-dessus de la moyenne du marché.

Patrick Chambet

http://www.chambet.com

Consultant Senior - Edelweb / Groupe ON-X

http://www.edelweb.fr - http://www.on-x.com

BIBLIOGRAPHIE

[1] Charte FTPI : http://www.freesec.net/pro_charte.htm

[2] Scénario d'attaque réelle : http://www.chambet.com/publications/SPIRAL-Scenario-catastrophe.pdf

[3] La guerre de l'information : http://www.miscmag.com/articles/index.php3?page=404

[4] The art of portscanning : http://www.insecure.org/nmap/nmap_doc.htm

[5] Les attaques externes : http://www.miscmag.com/articles/index.php3?page=106

[6] Les buffer overflows : http://www.phrack.org/phrack/49/p49-14

[7] Les format strings : http://www.security-labs.org/index.php3?page=601

[8] Les heap overflows : http://www.phrack.org/phrack/57/p57-0x08

[9] Les vulnérabilités applicatives : http://www.chambet.com/publications/sec-web-apps

[10] Durcissement d'Internet Explorer et Outlook Express : http://www.chambet.com/publications/ie-oe-security/index.html