ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Protection des protocoles

www.ofppt.info

OFPPT

DIRECTION **R**ECHERCHE ET **I**NGENIERIE DE **F**ORMATION

SECTEUR NTIC

Sommaire

1. Introduction

2. Description d'attaques sur différents protocoles et

Introduction

Ce chapitre décrit les failles intrinsèques de différents protocoles. Intrinsèques par le fait qu'elles ne sont pas liées à une faille applicative du client ou du serveur gérant ce protocole, mais plutôt à sa conception. Nous présenterons aussi la manière de s'en protéger.

2.1. Dynamic Host Configuration Protocol – DHCP

Le protocole DHCP est utilisé pour délivrer dynamiquement une adresse IP unique pour chaque machine le demandant sur le réseau interne. En clair, si un client interne veut obtenir une adresse IP pour bénéficier des services réseau, il envoie un message DHCP à tout le réseau (broadcast) pour trouver le serveur DHCP. Le serveur DHCP répondra en lui envoyant tous les paramètres de configuration réseau.
Ce service permet «d'alléger» la gestion du réseau en évitant d'avoir des configurations statiques à maintenir sur chaque machine. Malheureusement, le protocole DHCP comporte diverses failles que nous allons vous présenter.

2.1.1. Attaque par épuisement de ressources

Comme il l'a été décrit, un serveur DHCP possède un stock d'adresses IP qu'il distribue aux différents clients. Ce stock est bien sûr limité. Il y aura seulement un nombre défini de clients
pouvant disposer des différentes adresses IP en même temps. Si le serveur est bien administré avec une liste «fermée» de correspondances entre adresses MAC et IP aucune attaque par épuisement n'est possible.
Si le service est mal administré ; c'est à dire que les correspondances entre adresses MAC et IP se font dynamiquement à partir d'une plage d'adresses IP vacantes, le scénario suivant est possible.
Si un pirate génère un grand nombre de requêtes DHCP semblant venir d'un grand nombre de clients différents, le serveur épuisera vite son stock d'adresses. Les «vrais» clients ne pourront donc plus obtenir d'adresse IP : le trafic réseau sera paralysé.

2.1.2. 9.1.2. Faux serveurs DHCP

Cette attaque vient en complément de la première. Si un pirate a réussi à saturer un serveur DHCP par épuisement de ressources, il peut très bien en activer un autre à la place. Ainsi il pourra ainsi contrôler tout le trafic réseau.

2.1.3. 9.1.3. Comment s'en protéger ?

Chaque fois que c'est possible, il faut limiter le service DHCP à une liste «fermée» de correspondances d'adresses MAC et IP. De cette façon toute requête «étrangère» à cette liste est systématiquement rejetée.

• SousWindows, remplissez les champs de l'option `Réservations` dans le programme de configuration du serveur DHCP

• Sous Linux, éditez le fichier `/etc/dhcpd.conf` sur le serveur DHCP. Par exemple, pour un
client `toto` avec l'adresse MAC `00:C0:34:45:56:67` à laquelle on fait correspondre
S'il est impossible d'établir une liste «fermée», segmentez votre réseau en sous-réseaux et attribuez-leur chacun un serveur DHCP. Ces serveurs seront indépendants les uns des autres. Enfin, les nouvelles versions du protocole DHCP permettent l'utilisation de mécanismes d'authentification plus stricts. Assurez vous que vos serveurs utilisent ces versions de protocoles (Voir RFC3118 (http://www.faqs.org/rfcs/rfc3118.html)).

2.2. Domain Name Service – DNS

Le protocole DNS assure la correspondance entre le nom d'une machine et son adresse IP. Un serveur DNS est en écoute par défaut sur le UDP port 53. Les attaques décrites ici concernent les faiblesses du protocole DNS.

2.2.1. Le DNS ID spoofing

C'est la première attaque que nous allons décrire. Elle aboutit à un détournement de flux entre deux machines à l'avantage du pirate.

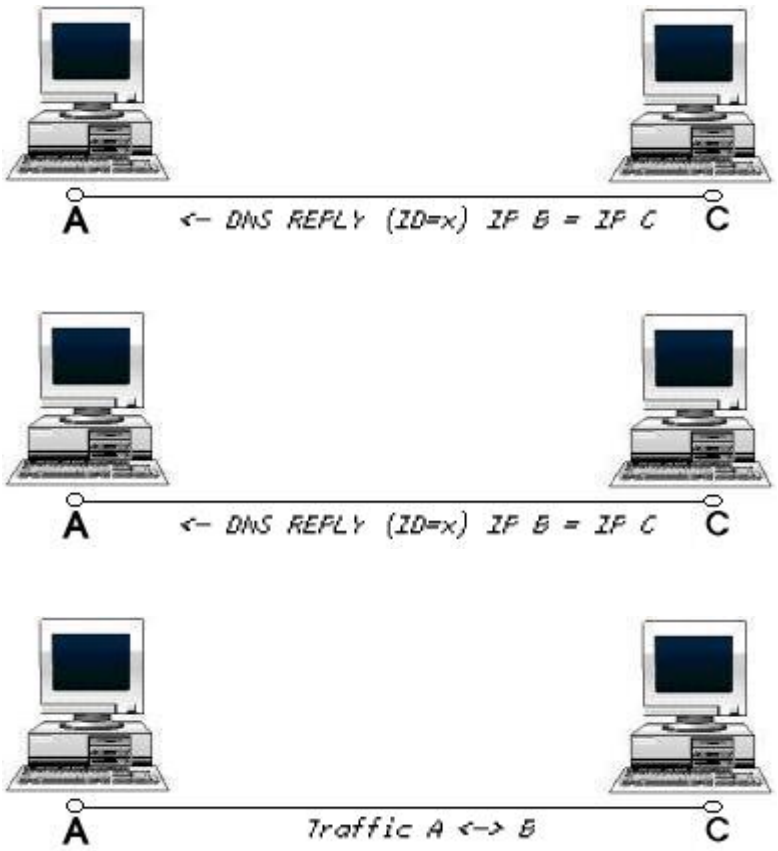
Imaginons qu'un client A veuille établir une connexion avec une machine B. La machine A connaît le nom de la machine B mais pas son adresse IP, ce qui lui empêche pouvoir communiquer avec.

La machine A va donc envoyer une requête au serveur DNS du réseau de B pour connaître l'adresse IP de B, cette requête sera identifiée par un numéro d'identification (ID).

Le serveur répond à cette requête en fournissant l'adresse IP de B et en utilisant le même numéro d'ID. Ce numéro a une valeur comprise entre 0 et 65535.

Le DNS ID spoofing a pour but de d'envoyer une fausse réponse à une requête DNS avant le serveur DNS. De cette façon, le pirate peut rediriger vers lui le trafic à destination d'une machine qu'il l'intéresse.

Dans notre exemple, un pirate C doit répondre à A avant le serveur DNS (D) du réseau de B. Ainsi, il envoie à A son adresse IP associée au nom de la machine B. A communiquera alors avec le pirate C au lieu de la machine B.
Illustration :



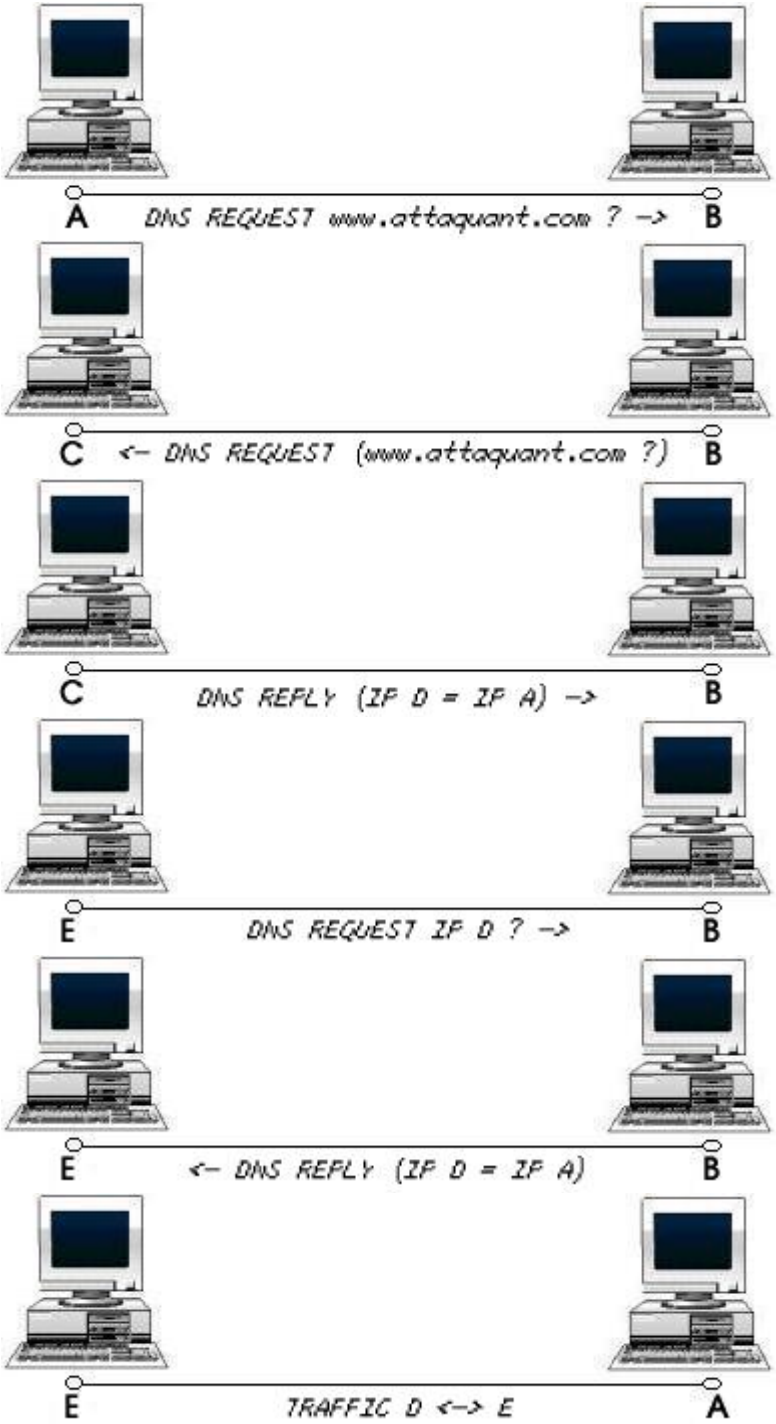
Néanmoins, pour implémenter cette attaque, le pirate doit connaître l' ID de requête DNS. Pour cela, il peut utiliser un sniffer s'il est sur le même réseau, soit prédire les numeros d'ID par l'envoi de plusieurs requêtes et l'analyse des réponses.

2.2.2. Le DNS cache poisoning

Le principe de cette attaque est très similaire à celui de l'ARP-Poisoining. Pour gagner du temps dans la gestion des requêtes, le serveur DNS possède un cache temporaire contenant les correspondances adresses IP - noms de machine. En effet, un serveur DNS n'a que la table de correspondance des machines du réseau sur lequel il a autorité. Pour des machines distantes, il doit interroger d'autres serveurs DNS. Pour éviter de les interroger à chaque requête, il garde en mémoire (dans un cache), le résultat des précédentes requêtes.
L'objectif du pirate est d'empoisonner ce cache avec de fausses informations. Pour cela, il doit avoir un nom de domaine sous contrôle et son serveur DNS.

Imaginons qu'un pirate (A) possède le nom de domaine `attaquant.com`, et son serveur DNS (C) et qu'il veuille empoisonner le cache du serveur DNS (B) du réseau `cible.net`.
Le pirate envoie une requête au serveur DNS (B) du réseau `cible.net` demandant la résolution du nom de domaine `attaquant.com`.

Le serveur DNS (B) de `cible.net` va donc envoyer une requête sur le serveur DNS (C) de l'attaquant (c'est lui qui a autorité sur le domaine `attaquant.com`). Celui-ci répondra et joindra des informations additionnelles falsifiées par le pirate (un nom de machine (D) associé à l'adresse IP (A) du pirate). Ces informations seront mises en cache sur le serveur DNS (B) de `cible.net`. Si un client quelconque (E) demande l'adresse IP pour le nom de la machine (D), il recevra l'adresse du pirate (A) en retour.



9.2.3. Comment s'en protéger ?

Configurez votre serveur DNS pour qu'il ne résolve directement que les noms de machine du réseau sur lequel il a autorité.
Autorisez seulement des machines internes à demander la résolution de noms de domaines distants.
Mettez à jour ou changez les logiciels assurant le service DNS pour qu'ils vous protégent des attaques décrites précedemment.

2.2.3.

2.2.4.

2.3. FTP

FTP (File Transfert Protocol, en écoute par défaut sur les ports 20 et 21) est le service utilisé pour assurer le transfert de fichiers. Il y a deux types de serveurs FTP : les serveurs FTP avec authentification par mots de passe et les serveurs anonymes. Pour les premiers, le client désirant se connecter devra fournir un login accompagné d'un mot de passe pour authentification. Dans le cas du serveur FTP anonyme, tout le monde peut s'y connecter librement.
Le premier défaut du protocole FTP est de ne pas encrypter les mots de passe lors de leur transit sur le réseau. Les mots de passe associés aux logins circulent en clair à la merci des sniffers.

Voici l'exemple d'une interception par un sniffer d'une authentification FTP :

Le logiciel utilisé est tcpdump.

```
22 :10 :39.528557 192.168.1.3.1027 ¿ 192.168.1.4.ftp : P 1 :12(11) ack
47
win 5840 ¡nop,nop,timestamp 441749 100314¿ (DF) [tos 0x10]
0x0000 4510 003f 88d6 4000 4006 2e7b c0a8 0103 E.. ?..@.@.......
0x0010 c0a8 0104 0403 0015 e351 3262 8d6a dd80 .........Q2b.j..
0x0020 8018 16d0 68da 0000 0101 080a 0006 bd95 ....h...........
0x0030 0001 87da 5553 4552 2061 6c65 780d 0a00 ....0,1,0USER.alex...
22 :10 :57.746008 192.168.1.3.1027 ¿ 192.168.1.4.ftp : P 12 :23(11) ack
80
win 5840 ¡nop,nop,timestamp 443571 101048¿ (DF) [tos 0x10]
0x0000 4510 003f 88d8 4000 4006 2e79 c0a8 0103 E.. ?..@.@..y....
0x0010 c0a8 0104 0403 0015 e351 326d 8d6a dda1 .........Q2m.j..
0x0020 8018 16d0 5ba1 0000 0101 080a 0006 c4b3 ....[...........
0x0030 0001 8ab8 5041 5353 2074 6f74 6f0d 0a00 ....0,1,0PASS.toto...
```

On peut voir facilement que l'utilisateur `alex` a le mot de passe `toto`.

2.3.1. Le serveur FTP anonyme

Le serveur FTP anonyme pose de plus gros problèmes. Le premier est qu'une mauvaise gestion des droits d'accès peut s'avérer être une erreur fatale. Laisser trop de répertoires en droit d'écriture et/ou d'exécution est plus que dangereux pour la sûreté du système. Le pirate pourrait y installer ou y exécuter des codes malveillants lui permettant d'accroître son pouvoir sur la machine.

2.3.2. Boucing attack - Attaque par rebonds

Les serveurs FTP anonymes peuvent être sujets à des attaques par rebonds. Ces attaques consistent à utiliser un serveur FTP anonyme comme relais pour se connecter à d'autres serveurs FTP. Imaginons qu'un pirate se voit refuser l'accès par un serveur FTP dont l'accès est alloué à seulement un certain groupe d'adresses IP. Imaginons que le pirate ne fait pas partie de ce groupe, mais qu'un serveur FTP anonyme y appartienne. Le pirate peut très bien se connecter sur le serveur FTP anonyme, utiliser les commandes assurant la connexion sur le serveur FTP protégé et y récupérer des fichiers.

2.3.3. Comment s'en protéger ?

Installez un serveur FTP anonyme seulement en cas d'absolue nécessité. Si vous devez le faire, limitez au maximun les droits sur les différents répertoires et fichiers laissés au public.
Pour vous protéger des attaques par sniffer, je vous recommande d'utiliser SFTP (Secure FTP) pour vos transactions FTP. SFTP encryptera les échanges et les protégera ainsi des écoutes indiscrètes. Vous pouvez aussi utiliser des tunnels comme IPSec pour protéger vos connexions
Filtrez les accès (via un firewall) en allouant seulement l'accès à un certain groupe d'adresses IP (en évitant d'inclure des serveurs anonymes permettant de servir de relais).

2.4. HTTP

Un serveur HTTP est en écoute sur le port 80. Le protocole HTTP est sûrement le plus utilisé sur le web pour les pages html. Ce protocole ne comporte pas de failles intrinsèques majeures. Par contre, les applications assurant son traitement sont souvent bourrées de failles. Cela vient du fait que le web devient de plus en plus demandeur en terme de convivialité et cela génère une complexité plus grande des applications, d'où un risque de failles plus important. Nous allons décrire ces failles une à une.

2.4.1. Les serveurs trop bavards

Parfois, les bannières des serveurs web sont trop explicites. Exemple sur un serveur Apache :

```
[root@nowhere /root]# telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Date: Sun, 04 Jan 2004 15:07:06 GMT
Server: Apache/1.3.29 (Debian GNU/Linux)
Last-Modified: Sat, 24 Nov 2001 16:48:12 GMT
ETag: "17082-100e-3bffcf4c"
Accept-Ranges: bytes
Content-Length: 4110
Connection: close
Content-Type: text/html; charset=iso-8859-1
Connection closed by foreign host.
```

Lors de l'envoi de la commande **`HEAD / HTTP/1.0`**, trop d'informations sont données.
Les pages d'erreurs (404 : page non trouvée) peuvent aussi contenir des informations sur le système.

2.4.2. Vulnérabilités liées aux applications web

La complexité des serveurs ou des navigateurs (clients) web pose de gros problèmes de sécurité. Ces applications sont vulnérables à de nombreux bugs. Chaque application a son type de faille.
Netscape par exemple devient vulnérable lors du traitement de certaines chaînes de caractères. Cela peut permettre de remonter toute l'arborescence des fichiers du serveur. Les serveurs IIS peuvent renvoyer un shell système pour un envoi de commandes particulières.
Les langages comme Javascript, Perl, PHP, ASP pour la réalisation de scripts peuvent se rélèver dangereux. L'origine d'une faille dans une application web peut apparaître à cause de deux problèmes. Le premier est la fiabilité de la conception du script, le second est la fiabilité des fonctions utilisées. Si un script est mal conçu, il peut être la source de nombreuses failles. De même, si sa conception est bonne mais qu'il utilise des fonctions boguées, il peut se révéler encore plus dangereux.

2.4.3. Comment se protéger ?

Vérifiez que votre serveur web n'est pas trop bavard. Si c'est le cas, modifiez sa configuration pour qu'il se taise. Pour cela, consultez la documentation pour modifier le contenu des messages d'erreur ou de bienvenue.

Un serveur web ne devrait jamais être exécuté avec les droits administrateurs.
Mettez à jour les navigateurs et les serveurs pour prévoir d'éventuelles failles.
Lors du développement de scripts, prenez garde lors de la conception à la gestion des droits des utilisateurs pour son exécution. Informez-vous aussi sur les fonctions connues pour être «sensibles».
Les NIDS peuvent être une bonne parade contre les attaques reposant sur des failles logicielles. Ils permettent de détecter l'exécution de telles attaques
L'utilisation de SHTTP (Secure HTTP) est aussi une bonne parade contre les attaques HTTP.
Une bonne définition de SHTTP est donné par E.Rescorla et A. Schiffman :
"Le protocole SHTTP est une extension de HTTP qui fournit des services de sécurité, applicables indépendamment, qui permettent de garantir la confidentilité, l'authenticité/intégrité, et le non refus d'origine."
SSL ("Secure Socket Layer" pour Netscape) permet de protéger les transactions web, il peut être judicieux de l'utiliser.

Note d'attention particulière.

Pour approfondir le sujet....

Proposition de références utiles permettant d'approfondir le thème abordé

Sources de référence

Citer les auteurs et les sources de référence utilisées pour l'élaboration du support