Les failles exploitées par les spammeurs

Du côté de l'administrateur

Nicolas Bareil

28 mars 2006

Plan

Réseau de bots

Qu'est-ce qu'un bot ?

Un bot est une machine infectée et contrôlée par un pirate.

L'union fait la force

Un bot isolé est inutile, mais en posséder plusieurs milliers offre :

- Attaques DoS,
- Robots spammeurs,
- Scans de réseau,
- Récolte de données,
- Une bonne monnaie d'échange ;

Fonctionnement d'un bot

Capacité d'un bot :

- Quelques commandes internes,
- Possibilité de se mettre à jour automatiquement,
- Controle du bot via IRC,
- Écrit *from scratch* ou développé à partir de *framework* existant ;

Récolte d'informations

Les données récupérées peuvent être :

- Adresses mail
  - Carnet d'adresses,
  - Recherche dans le disque dur,
- Liste de fichiers présents sur le disque,
- Mots de passe, clefs privées, etc.
- Topologie réseau ;

Et les mails eux-même afin de les renvoyer modifier.

Mais. . .

Difficile d'écrire un moteur SMTP afin d'envoyer du spam :

- Une RFC longue et piégeuse,
- Des montagnes de détails ;

Mais rater 20% des envois est négligeable.

Détection possible de bots

Toutes ces erreurs d'implémentation pourront nous être utiles !

Mais...

Difficile d'écrire un moteur SMTP afin d'envoyer du spam :

- Une RFC longue et piégeuse,
- Des montagnes de détails ;

Mais rater 20% des envois est négligeable.

Détection possible de bots

Toutes ces erreurs d'implémentation pourront nous être utiles !

Mais. . .

- Obsolescence rapide
  - Détection par les anti-virus,
  - Code désuet ou buggé,
  - Réinstallation de la machine par l'utilisateur ;
- Vol du botnet par un autre pirate ;

Utilisation de serveurs dédiés

Location de serveurs dédiés au spamming :

- Officiellement des serveurs de mailing `opt-in`,
- Utilisant de vrais serveurs `SMTP`

En théorie

Une solution très efficace.

Utilisation de serveurs dédiés

Location de serveurs dédiés au spamming :

- Officiellement des serveurs de mailing `opt-in`,
- Utilisant de vrais serveurs `SMTP`

En théorie

Une solution très efficace.

Architecture coûteuse

Solution élégante, mais coûteuse :

- Bande passante importante,
  - Connectivité payée au volume ;
- Puissance de calcul pour paralléliser au maximum,
- Paiement du matériel et de la salle d'hébergement ;

Très peu viable

- Durée de vie très courte,
  - À la première plainte au fournisseur de connectivité ;
- Architecture centralisée, peu flexible,
- Non anonyme, possibilité de poursuites judiciaires ;

Aussi rapide à tomber qu'à être monté

Une fois l'adresse du serveur *blacklistée*, l'efficacité devient nulle.

OpenCGI

Utilisation de script mal programmé permettant l'envoi de mail en masse.

- Application vulnérable,
- Fonction « Send this link » sans bridage,
- Plateformes de test
  - Ni protégé,
  - Ni monitoré,
  - Exemple : site de démonstration de webmail ;

Open HTTP Proxy

La requête `CONNECT` permet de se connecter à une machine et à un port précis.

- Implémentée sur tous les proxys pour gérer le `SSL`,
- Non limitative : c'est un vrai tunnel ;

Un proxy ouvert ?

Les spammeurs l'utilisent alors pour faire du `SMTP`.

Exemple de spams au travers un proxy HTTP

CONNECT vers un serveur SMTP

```
CONNECT smtp.bigcorp.com :25\r\n
\r\n
EHLO my.hostname.com
MAIL FROM :<>
RCPT TO :<sales@poorcorp.com>
DATA
...
.
```

Exemple de spams au travers un proxy HTTP

CONNECT vers un serveur SMTP

```
CONNECT smtp.bigcorp.com :25\r\n
\r\n
EHLO my.hostname.com
MAIL FROM :<>
RCPT TO :<sales@poorcorp.com>
DATA
...
.
```

OpenProxy

Encore d'autres méthodes...

- Méthode `HTTP POST`,
- Proxy Socks ;

Serveur SMTP OpenRelay

SMTP Openrelay

Un serveur SMTP ouvert est un serveur qui transmet les mails de n'importe qui vers n'importe où, souvent à cause d'une erreur dans la configuration.

C'est l'environnement le plus favorable pour un spammeur car :

- Stable,
- Discret,
- Efficace : comprends en intégralité le protocole SMTP,
- Décentralisé : la perte d'un noeud n'est pas critique ;

Cela peut arriver à tout le monde

Certains groupes de pirates modifient la configuration de serveurs compromis afin de les transformer en relais ouvert.

Plan

Objectif des spammeurs

- Récupérer des adresses électroniques
- Une adresse (vérifiée) valide se revend plus cher,
- Exploiter d'autres clients ;

Solutions ?

Deux solutions pour arriver à ses fins :

- Pousser l'utilisateur à faire une action,
- Éxecuter du code dans son dos ;

HTM Hell

Un mail au format `HTML` est un vecteur parfait :

- Liens, images, *frames*. . .
- Éxecution de code Javascript possible,
- Complexe à interpréter : vulnérabilités nombreuses ;

Solutions ?

Deux solutions pour arriver à ses fins :

- Pousser l'utilisateur à faire une action,
- Éxecuter du code dans son dos ;

HTM Hell

Un mail au format `HTML` est un vecteur parfait :

- Liens, images, *frames*. . .
- Éxecution de code Javascript possible,
- Complexe à interpréter : vulnérabilités nombreuses ;

Image cachée

L'affichage d'image lors de la lecture d'un mail est pratique. . . pour le spammeur :

- Téléchargement automatique,
- Pas forcément visible (carré transparent d'1x1 pixel),
- Permet d'obtenir beaucoup d'informations :
  - Navigateur,
  - Système d'exploitation,
  - Langue ;

Personnalisation de l'adresse de l'image

Chaque URL est unique et spécifique à une adresse mail, s'il y a un hit, c'est que l'adresse est valide.

Image cachée

L'affichage d'image lors de la lecture d'un mail est pratique. . . pour le spammeur :

- Téléchargement automatique,
- Pas forcément visible (carré transparent d'1x1 pixel),
- Permet d'obtenir beaucoup d'informations :
  - Navigateur,
  - Système d'exploitation,
  - Langue ;

Personnalisation de l'adresse de l'image

Chaque URL est unique et spécifique à une adresse mail, s'il y a un hit, c'est que l'adresse est valide.

Même chose pour les liens

Les liens sont également trompeurs :

- Véritable lien différent de celui affiché,
- Adresse personnalisée,
- Technique d'obfuscation d'URL ;

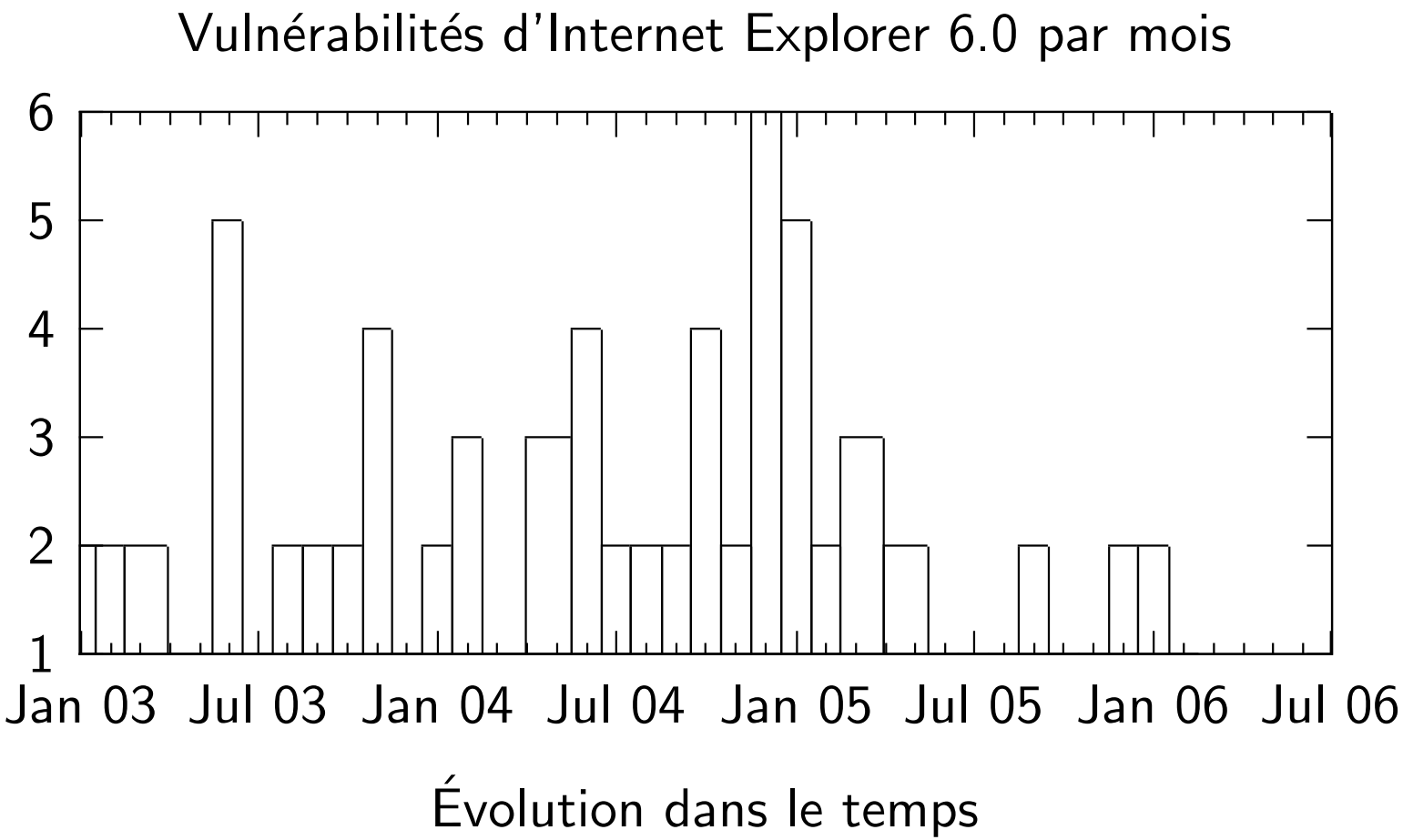
Sans compter les liens de désabonnements qui ne servent qu'à valider une adresse électronique !

Javascript & ActiveX

Toujours les mêmes à profiter des technologies :

- Exploitation de vulnérabilité des machines virtuelles,
- Lecture ou écriture de cookies,
- Envoi de données ;

Un navigateur ≃ passoire



Le pire. . .

Le pire, c'est que. . .

Ce ne sont que des vulnérabilités publiques !

Le cas Windows Meta File

Par exemple, la vulnérabilité « `WMF` » a été découverte par hasard sur un site pornographique.

Plan

Les bots

Caractéristiques des bots

- Implémentation minimaliste du protocole `SMTP`,
  - Cherchent à envoyer le plus de mail le plus vite possible,
  - Aucune gestion d'erreur
- Aucun état (pas de mémoire),
- Les bots viennent d'adresses résidentielles ;

Serveurs dédiés et ouverts

Les serveurs dédiés

- Comportement normal,
- Adresse fixe,
- Éphémère ;

Serveur ouvert

- Adresse résidentielle ;

OpenCGI et OpenProxy

OpenProxy :

- Très mauvais client SMTP ;

OpenCGI :

- Pourrait être détecté si `identd` existait encore ;
- Ajout d'en-têtes par l'hébergeur :

Exemple de spam

```
X-AntiAbuse: Primary Hostname - x.y.net
X-AntiAbuse: Originator/Caller UID/GID - [99 500]
X-AntiAbuse: Sender Address Domain - x.y.net
X-Source-Args: /usr/apache/bin/httpd -DSSL
X-Source-Dir: t.org:/public_html/contents/OldFIles
```

Plan

Session SMTP

```
220 mx.serveur.org ESMTP
HELO client.pays.fr
250 serveur.serveur.org Hello client.pays.fr [192.168.88.10]
MAIL FROM:<alice@paysdesmerveilles.org>
250 OK
RCPT TO:<bob@serveur.org>
250 Accepted
```

Suite...

Session SMTP, suite et fin

```
DATA
354 Enter message, ending with "." on a line by itself
From: alice@paysdesmerveilles.org
To: bob@serveur.org
Subject: vaisselle

I'm not a spam!
.
250 OK id=1CIASN-0000X0-IH
QUIT
221 mx.server.org closing connection
```

Quelques règles imposées par la RFC

Obligations de l'administrateur :

- L'adresse `postmaster` doit accepter **tous** les mails,
- On ne peut rejeter un client qu'après l'étape `RCPT TO`,
- Si possible, ne JAMAIS envoyer de *bounce* ;

L'intérêt de bloquer dès la session SMTP

Ce qu'en pense les standards

La RFC 2505 part du principe qu'un message est accepté ou bloqué lors de la session `SMTP` ou par le mail user agent.

Ainsi

- Pas de copie locale du message,
- Nous ne prenons aucune responsabilité,
- Nous ne bouncerons pas !

Exemple de spamming par bounce

`ezmlm` est écrit de façon à bouncer tout message qu'il ne comprendrait pas avec le message original dans son intégralité.

- Profitait de la réputation du serveur (adresse IP whitelistée),
- Utilisait pleinement les ressources du serveur ;

$\Longrightarrow$ Nécessite de patcher le code !

Ajout de temps mort

Les mailers mal programmés envoient les mails en rafale, sans se soucier du protocole.

Or, Un client `SMTP` doit attendre l'`HELO`/`EHLO` du serveur, dans la limite de cinq minutes.

Introduction d'un délai avant le `HELO`

$\Longrightarrow$ Rejet des clients envoyant des données avant nous

Ajout de temps mort

Les mailers mal programmés envoient les mails en rafale, sans se soucier du protocole.

Or, Un client `SMTP` doit attendre l'`HELO`/`EHLO` du serveur, dans la limite de cinq minutes.

Introduction d'un délai avant le `HELO`

$\Longrightarrow$ Rejet des clients envoyant des données avant nous

Bon usage des délais

Avantages

- Très efficace contre les bots ou virus,
- Opération presque gratuite pour le système,
- Aucun effet de bord
  - À condition d'avoir un délai raisonnable,
  - Tenir compte des *callouts* ;

Meilleure façon de l'utiliser

Mettre des délais en fonction du client :

- Adresses résidentielles,
- Classe de continents spammeurs ;

Bon usage des délais

Avantages

- Très efficace contre les bots ou virus,
- Opération presque gratuite pour le système,
- Aucun effet de bord
  - À condition d'avoir un délai raisonnable,
  - Tenir compte des *callouts* ;

Meilleure façon de l'utiliser

Mettre des délais en fonction du client :

- Adresses résidentielles,
- Classe de continents spammeurs ;

Greylisting

Il y a plusieurs codes d'erreur SMTP :

- Permanentes, ne jamais réessayer !
- Temporaires, réessayer plus tard ;

Les robots sont fainéants

Les robots ne regardent pas les codes d'erreur et passent tout de suite à l'adresse suivante, ne réessayant jamais.

Fonctionnement du Greylisting

Le Greylisting maintient une table de sessions :

-  Lorsqu'un serveur inconnu se présente, on refuse temporairement son mail et on démarre le timer,
- Tant que le timer n'est pas terminé, on refuse le message,
- Maintenant, on peut l'accepter ;

Fonctionnement du Greylisting

Le Greylisting maintient une table de sessions :

-  Lorsqu'un serveur inconnu se présente, on refuse temporairement son mail et on démarre le timer,
-  Tant que le timer n'est pas terminé, on refuse le message,
- Maintenant, on peut l'accepter ;

Fonctionnement du Greylisting

Le Greylisting maintient une table de sessions :

-  Lorsqu'un serveur inconnu se présente, on refuse temporairement son mail et on démarre le timer,
-  Tant que le timer n'est pas terminé, on refuse le message,
-  Maintenant, on peut l'accepter ;

Efficacité

À l'heure actuelle

Méthode efficace :

- Économique,
- Facile à mettre en œuvre ;

Mais. . .

- Entraine des latences importantes
  $\Longrightarrow$ les commerciaux n'aiment pas attendre,
- Les spammeurs vont s'adapter,
- Repose sur le respect des standards : tous les MTA ne sont pas égaux face aux politiques de *retries* ;

Plan

Prologue de la session

Commande HELO/EHLO

Envoyée par le client, elle sert à présenter son nom pleinement qualifié.

Exemple d'un EHLO valide

```
220 mx.serveur.org ESMTP
EHLO moi.client.fr
250-mx.serveur.org Hello moi.client.fr
```

Usurpation du nom présenté

Comportement étrange des moteurs SMTP des virus ou vers : ils ne savent pas donner un nom correct.

$\Longrightarrow$ Utilisation du nom du serveur (pour contourner des filtres ?)

Exemples

```
HELO mx.serveur.org
HELO 234.32.12.3
HELO [234.32.12.3]
HELO one.of.the.reverse.dns
```

Usurpation du nom présenté

Comportement étrange des moteurs SMTP des virus ou vers : ils ne savent pas donner un nom correct.

$\Longrightarrow$ Utilisation du nom du serveur (pour contourner des filtres ?)

Exemples

```
HELO mx.serveur.org
HELO 234.32.12.3
HELO [234.32.12.3]
HELO one.of.the.reverse.dns
```

Usurpation du nom présenté

Comportement étrange des moteurs SMTP des virus ou vers : ils ne savent pas donner un nom correct.

$\Longrightarrow$ Utilisation du nom du serveur (pour contourner des filtres ?)

Exemples

```
HELO mx.serveur.org
HELO 234.32.12.3
HELO [234.32.12.3]
HELO one.of.the.reverse.dns
```

Usurpation du nom présenté

Comportement étrange des moteurs SMTP des virus ou vers : ils ne savent pas donner un nom correct.

$\Longrightarrow$ Utilisation du nom du serveur (pour contourner des filtres ?)

Exemples

```
HELO mx.serveur.org
HELO 234.32.12.3
HELO [234.32.12.3]
HELO one.of.the.reverse.dns
```

Reverse DNS

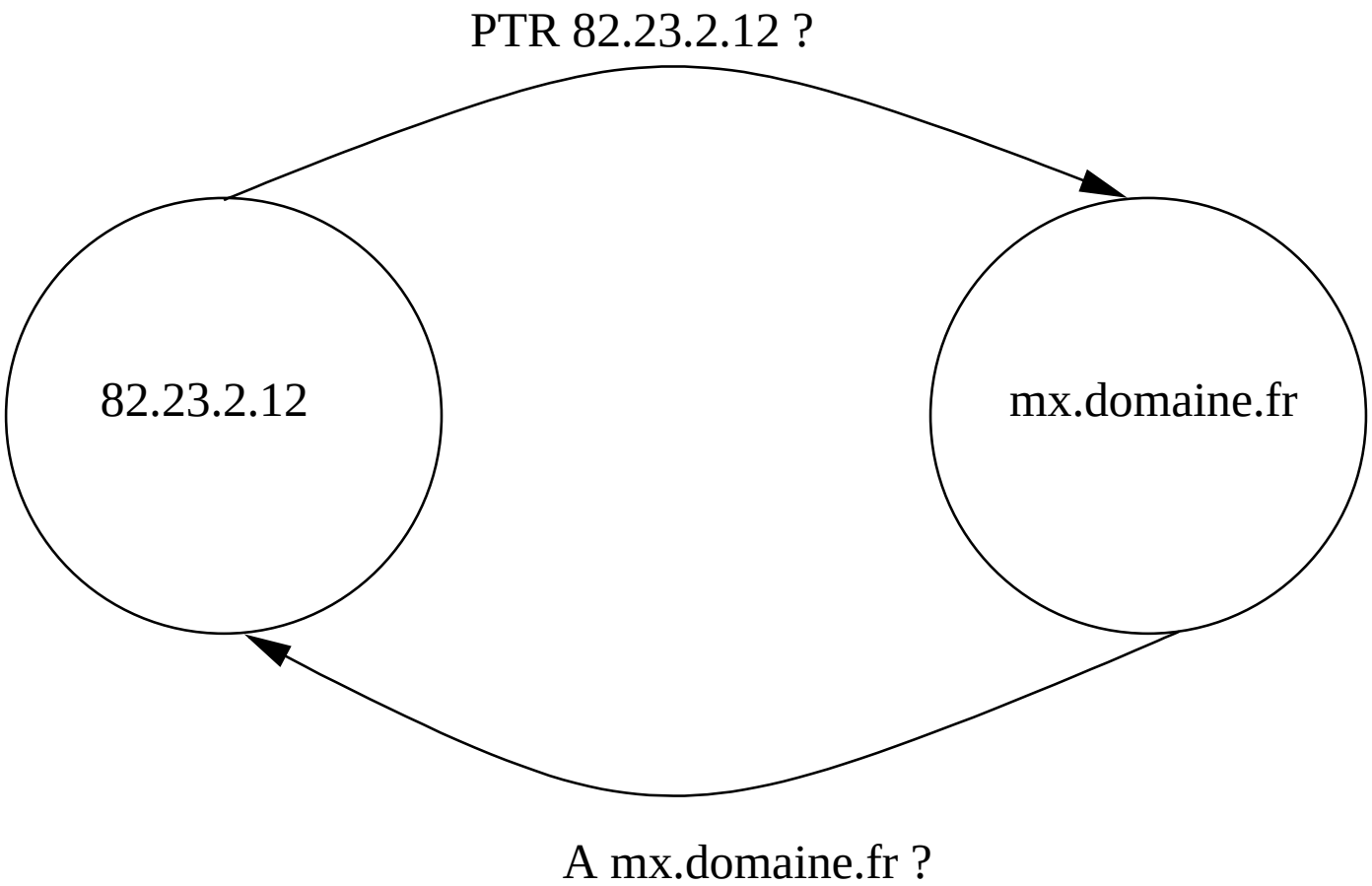


FIG.: Vérification du reverse DNS

Syntaxe incorrecte

Le nom donné doit être un FQDN donc :

- Comporter au moins un point,
- Être un des *reverse dns* de l'adresse IP

Exemples

```
HELO localhost
HELO 234.32.12.3
HELO labo-pc4231
```

Syntaxe incorrecte

Le nom donné doit être un FQDN donc :

- Comporter au moins un point,
- Être un des *reverse dns* de l'adresse IP

Exemples

```
HELO localhost
HELO 234.32.12.3
HELO labo-pc4231
```

Syntaxe incorrecte

Le nom donné doit être un FQDN donc :

- Comporter au moins un point,
- Être un des *reverse dns* de l'adresse IP

Exemples

```
HELO localhost
HELO 234.32.12.3
HELO labo-pc4231
```

Usurpation d'identité

"How could somebody steal my identity when I still haven't figured out who I am?"

Empêcher l'usurpation d'identité

Plusieurs techniques existent :

- Sender Policy Framework (SPF),
- SenderID, SPF vu par Microsoft,
- DomainKeys, poussé par Yahoo ! ;

Sender Policy Framework & SenderID

Récupération de l'émetteur

- Pour SPF : enveloppe,
- Pour SenderID : adresse dans le message ;

Vérification

À partir du domaine de l'émetteur, on vérifie qu'on est contacté par un serveur autorisé :

- Une liste de tous les serveurs autorisés est créée,
- Stockée dans une entrée `TXT` du DNS ;

DomainKeys

Utilisation de la cryptographie asymétrique :

- Publication de la clef publique des serveurs SMTP,
  - Dans un champ `TXT` de la zone DNS ;
- Signature de chaque message sortant du serveur
  - Ajout d'un en-tête dans le message ;

À la réception d'un message, on vérifie la signature.

Solutions limitées

Mesures insuffisantes

- Empêche l'usurpation,
- Les spammeurs ont vite adopté SPF,
- Peu applicable en général
  - Il faut forcer les utilisateurs à utiliser son serveur !

Conclusion

Ces technologies réduiront juste les dommages collatéraux.

Solutions limitées

Mesures insuffisantes

- Empêche l'usurpation,
- Les spammeurs ont vite adopté SPF,
- Peu applicable en général
  - Il faut forcer les utilisateurs à utiliser son serveur !

Conclusion

Ces technologies réduiront juste les dommages collatéraux.

Vérification des deux parties

Adresses valides ?

En pratique, il est impossible de vérifier qu'une adresse est valide.
$\Longrightarrow$ `VRFY` ou équivalent sont désactivés

Avec un peu de ruse. . . si !

Lors de la réception d'un mail, on va simuler la livraison d'un mail pour l'adresse de l'expéditeur.

```
while (1);
```

Mais c'est une boucle infinie ! ? !

Vérification des deux parties

Adresses valides ?

En pratique, il est impossible de vérifier qu'une adresse est valide.
$\Longrightarrow$ `VRFY` ou équivalent sont désactivés

Avec un peu de ruse. . . si !

Lors de la réception d'un mail, on va simuler la livraison d'un mail pour l'adresse de l'expéditeur.

`while (1);`

Mais c'est une boucle infinie ! ? !

Un callout en action

```
220 mx.serveur.fr ESMTP
EHLO moi.chezmoi.fr
250-mx.serveur.fr Hello moi.chezmoi.fr
250 ...
MAIL FROM:<>        adresse forcément valide !
250 OK
RCPT TO:<la@serveur.fr>
250 Accepted
RCPT TO:<pasla@serveur.fr>
550 unknown user
QUIT
221 mx.serveur.fr closing connection
```

DNSBL, RBL, etc.

DNS Blacklist ?

Stockage d'adresses IP dans un serveur DNS.

- IP ayant déjà spammé,
- IP d'une liste blanche,
- IP dans une plage d'adresses résidentielles ;

Alléchant ?

- Économique : une question/réponse DNS
  - À coupler avec un serveur de cache local,
  - Être sûr de ne pas tomber en timeout,

Mais comment choisir les listes ?

Éthique

Il faut impérativement connaître :

- Règles d'entrée de la liste,
- Règles de sortie,

Suivre son évolution

Lorsqu'une liste s'arrête, elle blackliste le monde entier afin que personne ne l'utilise. . . En théorie !

Mais comment choisir les listes ?

Éthique

Il faut impérativement connaître :

- Règles d'entrée de la liste,
- Règles de sortie,

Suivre son évolution

Lorsqu'une liste s'arrête, elle blackliste le monde entier afin que personne ne l'utilise. . . En théorie !

Plan

Vérification syntaxique

Comme pour le `SMTP`, peu de respect des standards

- Header manquant,
  $\Longrightarrow$ `MessageID` est un cas intéressant
- Mauvais encodage,
- Syntaxe incorrecte ;

Filtrage par mots clefs

Mots clefs

Langage des spammeurs :

$\Longrightarrow$ viagra, sexe, drogue, argent, casino, etc.

Calcul d'un score à partir de ces mots clefs (bonus/malus)

Spamassassin

Bien plus qu'une simple liste de mots :

- Analyse d'URL,
- Transformation d'alphabet,
- Vérification de syntaxe des mails,
- Tests réseaux,

$\Longrightarrow$ Usine à gaz ;

Limites du filtrage

Les spammeurs s'adaptent vite pour éviter les règles :

- Écriture avec un alphabet différent (viagra $\longrightarrow$ v14gr4),
- Modification des en-têtes des mails,

Veille des règles publiques

On estime qu'une règle Spamassassin postée perd 80% de son efficacité après sa publication.

Bayes & la classification automatique

Apprentissage

En fournissant un nombre conséquent de messages déjà triés (spam ou non), on va pouvoir attribuer une probabilité de spam à un mot.

À la réception d'un message, on somme les probabilités de chaque mots présent pour déterminer si cela est un spam.

Contre attaque des spammeurs

Pour affaiblir les probabilités, les spammeurs envoient des messages avec des mots aléatoires à la fin de leurs messages.

Bayes & la classification automatique

Apprentissage

En fournissant un nombre conséquent de messages déjà triés (spam ou non), on va pouvoir attribuer une probabilité de spam à un mot.

À la réception d'un message, on somme les probabilités de chaque mots présent pour déterminer si cela est un spam.

Contre attaque des spammeurs

Pour affaiblir les probabilités, les spammeurs envoient des messages avec des mots aléatoires à la fin de leurs messages.

Meilleure solution ?

Sûrement la solution la plus efficace !
$\Longrightarrow$ Intégrée partout : Thunderbird, Outlook, Evolution...
Mais :

- Nécessite un long apprentissage pour être efficace,
- Entretien constant,
- Individuel,
- Gourmand en espace disque et accès I/O ;

Hashage des messages

Hashage

Une fonction de hashage prend une suite d'octets de taille variable et la transforme en un condensé d'une taille fixée.

$\Longrightarrow$ Modifier un bit du message modifie totalement le hash

$\Longrightarrow$ On suppose la très faible probalité d'une collision

Publication des condensés de spam

À la réception de chaque message, on calcul son condensé et on regarde sur un serveur si ce hash est connu pour être du spam.

Parade contre ces hashes

Adaptation rapide

- Personnalisation des messages :
  - Ajout du nom de la personne,
  - Modification des dates ;
- Suite de caractères aléatoires à la fin du message ;

Conséquence

Ces services n'ont plus d'efficacité !

Plan

Que peuvent faire les ISP ?

Problématiques

- Difficultés techniques :
  - ACL très coûteuses et peu maintenables,
  - Passage à l'échelle ;
- Publicité,
- Liaison payée au volume

Solutions (?)

- Proposer des relais `SMTP` stables
  $\Longrightarrow$ Afin de limiter l'installation de serveur SMTP
- Le filtrage du port 25 n'est pas une solution
  $\Longrightarrow$ Mettre de la QoS sur le port 25 en sortie ?

Que peuvent faire les serveurs d'entreprise ?

Si moyens suffisants :

- Clusters de serveurs utilisant Spamassassin,
- Implémenter toutes les vérifications syntaxiques,
  - Les communications inter-entreprises utilisent des vrais `MTAs`,
- Utilisation de liste noire et blanche ;

Le *greylisting* est une option si les utilisateurs peuvent accepter un retard lors du premier échange.

Important

- Filtrez le port 25 en sortie !
- Surveillez le nombre d'envois ;

Que peuvent faire les serveurs d'entreprise ?

Si moyens suffisants :

- Clusters de serveurs utilisant Spamassassin,
- Implémenter toutes les vérifications syntaxiques,
  - Les communications inter-entreprises utilisent des vrais `MTAs`,
- Utilisation de liste noire et blanche ;

Le *greylisting* est une option si les utilisateurs peuvent accepter un retard lors du premier échange.

Important

- Filtrez le port 25 en sortie !
- Surveillez le nombre d'envois ;

Que peuvent faire les particuliers ?

Deux solutions efficaces à 99,99% :

- Spamassassin,
- Filtre Bayesien ;

En plus d'un firewall et un anti-virus à jour.

En entrée et en sortie

Personne n'est à l'abri

Vous devez contrôler tout ce que vous recevez et ce que vous émettez !

Soyez exigent

- Contrôle syntaxique,
- Surveillance statistiques,
- Utilisation de Spamassassin ;

Imposez des quotas !

Questions ?

Questions ou remarques ?

- nbareil@mouarf.org
- http://mouarf.org/