Correction du TP milxc et sécurité des échanges par mail

Question 1 : Avec l'architecture actuelle, donnez 2 exemples d'échange de mail frauduleux, essayez de les mettre en place.

Cette question a pour but de faire réfléchir les élèves aux différents problèmes que peuvent engendrer les échanges par mails. Il n'y a pas de réponse type. La réponse qui peut sembler la plus évidente est l'usurpation d'identité. Pour ce faire il suffit de suivre l'exemple suivant :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.isp-a.milxc 587
220 mi-isp-a-infra ESMTP Postfix (Debian/GNU)
HELO infra.isp-a.milxc
250 mi-isp-a-infra
MAIL FROM:<admin@target.milxc>
250 2.1.0 Ok
RCPT TO:<user@gcorp.milxc>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject : Usurpation d'identité
On a bien Hacker qui se fait passer pour Admin auprès de User
.
250 2.0.0 Ok: queued as 15AAC86A64
QUIT
221 2.0.0 Bye
```

Il faut ensuite regarder les mails de User et vérifier si il a bien reçu ce mail.

Cette question est un question d'introduction qui en théorie devrait être traitée en commun avant que les élèves commencent à réellement se plonger dans le TP. Elle permet les premières manipulation de **mi-lxc** de mieux connaitre l'architecture et les manipulations qu'ils seront amenés à faire tout le long du TP.

Question 2 : Dans la configuration actuelle, cette séparation des tâches est-elle mise en place sur tous les serveurs mails ? Mettez en place cette configuration dans le cas où elle n'est pas encore appliquée.

Dans la configuration fournie, les fichiers `/etc/postfix/master.cf` de **isp-a-infra** et de **gcorp-infra** auront bien la ligne `submission inet n       -       y       -       -       smtpd` décommentée. Mais ca ne sera pas le cas pour **target-dmz**. Les élèves doivent dans cette question apprendre à trouver les fichiers de configuration de postfix (les fichiers `main.cf` et `master.cf`). Ils doivent aussi tester différents échanges de mails pour se rendre compte qu'ils ne peuvent pas se connecter au port 587 de **target-dmz**. Une fois qu'ils auront remarqué que le port est bloqué sur cette machine uniquement ils peuvent comparer les fichiers des différents serveur mail pour remarquer que une des seules différence à ce moment de l'énoncé réside dans cette ligne qui est resté commenté pour la machine **target-dmz**. Pour mettre en place cette séparation des services, il suffit de décommenter la ligne et ensuite de lancer la command `postfix reload`.

Question 3 : En vous aidant de la documentation de postfix, quelle(s) paramètre(s) peuvent être utilisé(s) pour appliquer cette restriction ? Cette restriction est-elle actuellement appliquée sur les serveurs mail ?

Cette restriction n'est pas appliqué sur les serveurs mails. On peut simplement le vérifier en se connectant depuis extérieur sur le port 587 d'un serveur mail externe :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.target.milxc 587
220 mi-target-dmz ESMTP Postfix (Debian/GNU)
HELO smtp.target.milxc
250 mi-target-dmz
MAIL FROM:<hacker@isp-a.milxc>
250 2.1.0 Ok
RCPT TO:<admin@target.milxc>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject : Connexion frauduleuse au port 587
On a bien Hacker qui se connecte au port 587 du serveur SMTP de target.
.
250 2.0.0 Ok: queued as 17BBC89A24
QUIT
221 2.0.0 Bye
```

L'exemple ci dessus peut être répété à tous les serveurs de mail.

On peut s'aider du lien en annexe pour trouver que le paramètre permettant de vérifier des informations sur la personne qui envoie le mail est `smtp_client_restrictions`. Il faut maintenant regarder dans les paramètre possible de ce dernier pour trouver le paramètre permettant de vérifier l'adresse IP du client. On trouve : `permit_mynetworks` qui visiblement est l'option qui convient le mieux à notre problème. Il convient donc de rajouter dans le fichier `/etc/postfix/master.cf` la ligne suivante :

```
submission inet n       -       y       -       -       smtpd
  -o smtpd_client_restrictions=$mua_client_restrictions
```

Avec la variable `mua_client_restrictions` qui est définie dans le fichier `/etc/postfix/main.cf`

```
mua_client_restrictions = permit_mynetworks, reject
```

D'après la documentation, `permit_mynetworks` vérifie l'entrée de `mynetworks`. Il convient de bien définir ce champ avec les bonnes valeurs. Il faut donc remplacer les valeurs par :

```
mynetworks = 127.0.0.0/8, XXX.XXX.X.0/16
```

On pourrait simplement avoir toutes les informations réunies dans le fichier `/etc/postfix/master.cf` :

```
submission inet n       -       y       -       -       smtpd
  -o smtpd_client_restrictions=permit_mynetworks, reject
```

Ces manipulations doivent être effectuées sur les 3 serveurs mails. Il faut aussi penser à lancer la commande `postfix reload` pour prendre en compte les modifications. Certaines modifications peuvent ne pas être à jour même après un `reload`, on peut dans ce cas faire `postfix stop && postfix start` pour complètement redémarrer postfix.

Question 4 : Maintenant que vous avez appliqué cette restriction, quelle erreur obtenez vous lorsque vous essayez de reproduire le comportement interdit ? Où les messages d'erreurs sont-ils stockés sur le serveur de mail ?

Pour obtenir le message d'erreur il suffit d'essayer de faire ce que nous venons d'interdire :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.target.milxc 587
220 mi-target-dmz ESMTP Postfix (Debian/GNU)
HELO smtp.target.milxc
250 mi-target-dmz
MAIL FROM:<hacker@isp-a.milxc>
250 2.1.0 Ok
RCPT TO:<admin@target.milxc>
554 5.7.1 <unknown[100.120.0.4]>: Client host rejected: Access denied
```

Nous avons donc comme erreur : `554 5.7.1 <unknown[100.120.0.4]>: Client host rejected: Access denied`

Il faut maintenant chercher où sont stocké les logs pour un serveur géré par postfix. Ils sont stockés dans le fichier `/var/log/mail.log`. On trouve le message d'erreur :

```
NOQUEUE: reject: RCPT from unknown[100.120.0.4]: 554 5.7.1
<unknown[100.120.0.4]>: Client host rejected: Access denied; from=<hacker@isp-
a.milxc> to=<admin@target.milxc> proto=SMTP helo=<smtp.target.milxc>
```

Cette question permet dans un premier temps de vérifier que la configuration rentrée par les élèves précédemment est correcte en leur faisant tester. Elle leur permet aussi de trouver le fichier ou sont stockés les logs des serveurs de mail ce qui pourra leur être utile dans la suite du TP.

Question 5 : Avec la configuration établie aux questions précédentes, que peut faire un utilisateur final de malveillant en communiquant sur le port 587 ?

Cette question a pour but de rappeler aux élèves que lorsque l'on est en charge de la sécurité d'une architecture, il faut garder à l'esprit que l'on ne peut pas contrôler toutes les actions des utilisateurs dont on a la charge. Il convient donc de ne pas les considérer comme étant sans danger. Un utilisateur final peut être malveillant ou il peut se faire pirater son ordinateur.

Il est attendu à cette réponse qu'un utilisateur peut toujours usurper une adresse mail d'un serveur externe.

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.isp-a.milxc 587
220 mi-isp-a-infra ESMTP Postfix (Debian/GNU)
HELO infra.isp-a.milxc
250 mi-isp-a-infra
MAIL FROM:<admin@target.milxc>
250 2.1.0 Ok
RCPT TO:<user@gcorp.milxc>
250 2.1.5 Ok
```

```
DATA
354 End data with <CR><LF>.<CR><LF>
Subject : Usurpation d'identité
On a bien Hacker qui se fait passer pour Admin auprès de User
.
250 2.0.0 Ok: queued as 1AA6887C78
QUIT
221 2.0.0 Bye
```

Question 6 : Proposer et implémenter une solution pour empêcher cette action.

On cherche maintenant à empêcher qu'un de nos utilisateurs utilise une adresse mail externe pour communiquer depuis notre sous-réseau. On va donc fouiller dans la doc pour trouver un paramètre permettant de vérifier des informations sur l'adresse mail d'envoie. Ce paramètre est `smtpd_sender_restrictions`. Une des solutions que l'on peut adopter est d'autoriser seulement les adresses mail comportant le nom de domaine adéquat. Plusieurs options sont possibles on peut utiliser le paramètre `reject_unlisted_sender` par exemple. Ce paramètre regarde différents champs dont le champs `mydestination`. Il faut alors compléter ce champs pour n'autoriser que le nom de domaine désiré.

Les modifications à apporter dans le fichier `/etc/postfix/master.cf` sont les suivantes :

```
submission inet n       -       y       -       -       smtpd
  -o smtpd_client_restrictions=$mua_client_restrictions
  -o smtpd_sender_restrictions=$mua_sender_restrictions
```

On va donc définir la variable `mua_sender_restrictions` dans le fichier `/etc/postfix/main.cf` :

```
mydestination = *****.milxc, $myhostname, hash:/etc/postfix/access
mua_sender_restrictions = reject_unlisted_sender
```

Comme dit précédemment, `reject_unlisted_sender` regarde en particulier dans `mydestination`. On a donc rajouté une entrée avec le fichier `/etc/postfix/access` suivant :

```
*****.milxc OK
```

Il faut penser à faire la commande `postmap /etc/postfix/access` sinon le fichier ne sera pas reconnu.

Cette question est un peu compliqué car l'ajout du fichier `/etc/postfix/access` n'est pas évident à deviner même en ayant compris tout le reste avant. Un indice ou un changement de sujet pour mieux faire comprendre cette idée aux étudiants serait la bienvenue.

Encore une fois, `postfix reload` ou `postfix stop && postfix start`.

SPF

Question 7 : Afficher la politique SPF de gcorp.milxc et expliquez. Vous pouvez utiliser `dig` pour vous aider.

Le branche **gcorp** contrairement à la branche **isp-a** et **target** possède déjà une politique SPF. Pour la récupérer, il faut exécuter la commande suivante :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@mi-isp-a-hacker$ dig +short gcorp.milxc TXT
"v=spf1 mx -all"
```

On a donc le résultat suivant : `v=spf1 mx -all` On utilise la version **1** de SPF. Les personnes pouvant envoyer des mails pour le nom de domaine **gcorp.milxc** sont les personnes ayant l'adresse IP du serveur de mail de gcorp.

On peut récupérer son adresse IP avec les commande suivante :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@mi-isp-a-hacker$ dig +short gcorp.milxc MX
10 smtp.gcorp.milxc.
root@mi-isp-a-hacker$ dig +short smtp.gcorp.milxc
dmz.gcorp.milxc.
100.84.1.2
```

On peut aussi répondre à cette question en allant fouiller dans la configuration DNS de **gcorp-infra** :

```
root@buster$ ./mi-lxc.py attach gcorp-infra
root@mi-gcorp-infra$ cat /etc/nsd/gcorp.milxc.zone
$TTL    86400
$ORIGIN gcorp.milxc.
@  1D  IN  SOA ns.gcorp.milxc. hostmaster.gcorp.milxc. (
                  2002022401 ; serial
                  3H ; refresh
                  15 ; retry
                  1w ; expire
                  3h ; nxdomain ttl
                 )
       IN NS       ns.gcorp.milxc.
       IN MX    10 smtp.gcorp.milxc.
       IN TXT   "v=spf1 mx -all"
ns     IN A     100.84.1.2
ns     IN AAAA  2001:db8:84:1::2
dmz    IN A     100.84.1.2
dmz    IN AAAA  2001:db8:84:1::2
smtp   IN CNAME     dmz
imap   IN CNAME     dmz
```

Le fichier du TP pourrait être légèrement différent de ce dernier.

Cette question permet aux étudiants de commencer à comprendre SPF et à regarder comment on trouve une politique et comment elles peuvent être formées. Ce TP n'est pas sur les différentes fonctionnalités de SPF qui sont plus nombreuses que ce que l'on va utiliser dans la suite mais c'est déjà une bonne première approche pour comprendre en quoi ca peut servir.

Question 8 : Montrez qu'un utilisateur extérieur peut toujours usurper l'adresse mail de **gcorp-user**.

On a vu à la question précédente que une politique SPF était bien implémenté pour **gcorp.milxc**. Pour autant, cette dernière n'est pas activée et correctement configurée avec postfix.

Pour vérifier cela, il suffit de reproduire les exemples précédent mais cette fois ci en usurpant l'adresse mail de **gcorp-user**.

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.isp-a.milxc 587
220 mi-isp-a-infra ESMTP Postfix (Debian/GNU)
HELO infra.isp-a.milxc
250 mi-isp-a-infra
MAIL FROM:<user@gcorp.milxc>
250 2.1.0 Ok
RCPT TO:<admin@target.milxc>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject : Usurpation d'identité
On a bien Hacker qui se fait passer pour User auprès d'Admin
.
250 2.0.0 Ok: queued as A8EFA87C6E
QUIT
221 2.0.0 Bye
```

Il convient ensuite de vérifier que le mail a bien été reçu.

Question 9 : En vous aidant de `man` (ou autres sources) effectuez les manipulations pour activer SPF avec postfix.

On propose aux étudiants de regarder le manuel de **policyd-spf** car on trouve toutes les informations utiles pour l'implémenter directement sur postfix.

```
root@buster$ ./mi-lxc.py attach gcorp-infra
root@mi-gcorp-infra$ man policyd-spf
...
POSTFIX INTEGRATION
        1. Add the following to /etc/postfix/master.cf:

               policyd-spf  unix  -       n       n       -       0       spawn
                   user=policyd-spf argv=/usr/bin/policyd-spf

           NOTE: Check the path to both the installed Python interpreter and
                 policyd-spf.  These vary from system to system.  To use non-
default
                 settings, you must also add the config file (see above and
                 policyd-spf.conf(5) for details).  Python and Python 3 versions
                 prior to 3.3 are not supported.

        2. Configure the Postfix policy service in /etc/postfix/main.cf:

               smtpd_recipient_restrictions =
                   ...
                   reject_unauth_destination
                   check_policy_service unix:private/policyd-spf
                   ...
               policyd-spf_time_limit = 3600

           NOTE:  Specify check_policy_service AFTER reject_unauth_destination
or
                  else your system can become an open relay.

        3. Reload Postfix.
```

Normalement, rajouter cette configuration à notre configuration actuel est suffisant pour activer SPF. Pour autant, dans notre cas on prefere explicitement autoriser ou interdire dans la configuration. Ainsi on a modifié le fichier `/etc/postfix/master.cf` comme suit :

```
smtp      inet  n       -       y       -       -       smtpd
  -o smtpd_relay_restrictions=$mx_relay_restrictions
policyd-spf  unix  -       n       n       -       0       spawn
  user=policyd-spf argv=/usr/bin/policyd-spf
```

Avec ces modifications dans le ficher `/etc/postfix/main.cf`

```
policyd-spf_time_limit = 3600
mx_relay_restrictions = reject_unauth_destination,
                        check_policy_service unix:private/policyd-spf
```

Question 10: Avez vous pensé à rajouter la politique SPF dans la configuration DNS ? Si non, vous pouvez trouver le fichier à modifier dans `/etc/nsd/*.zone` dans les serveurs de mail.

Les modifications précédentes n'ont aucun effet si il n'y a aucune politique SPF à appliquer. Pour cette question, il suffit de rajouter la politique SPF dans la configuration DNS.

```
       IN TXT   "v=spf1 mx -all"
```

Cette ligne doit être ajoutée dans les fichiers `/etc/nsd/target.milxc.zone` et `/etc/nsd/isp-a.milxc.zone`.

Question 11 : Pouvez vous toujours usurper l'adresse de target-admin ou de gcorp-user avec isp-a-hacker ? Quel est le message obtenu pour signaler ce refus ?

Normalement, si la configuration est bonne, les adresses mails ne peuvent plus être usurpée. Si on effectue les commandes suivantes :

```
root@buster$ ./mi-lxc.py attach isp-a-hacker
root@isp-a-hacker$ nc smtp.isp-a.milxc 587
220 mi-isp-a-infra ESMTP Postfix (Debian/GNU)
HELO infra.isp-a.milxc
250 mi-isp-a-infra
MAIL FROM:<admin@target.milxc>
250 2.1.0 Ok
RCPT TO:<user@gcorp.milxc>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
.
250 2.0.0 Ok: queued as 51D8487C6F
QUIT
221 2.0.0 Bye
```

On obtient dans le fichier `/var/log/mail.log` de **gcorp-infra** :

```
mi-gcorp-infra policyd-spf[2013]: 550 5.7.23 Message rejected due to: SPF fail -
not authorized.
mi-gcorp-infra postfix/smtpd[2008]: NOQUEUE: reject: RCPT from
unknown[100.120.1.2]: 550 5.7.23 <user@gcorp.milxc>: Recipient address rejected:
Message rejected due to: SPF fail - not authorized.
```

Notre politique SPF est donc bien prise en compte et fonctionnelle.

Question 12 : Proposez divers scénarios à tester pour montrer que votre implémentation est bonne. Aidez-vous du fichier scénarii.txt et de la commande `./mi-lxc.py mail` pour vérifier votre configuration.

Cette question sert à conclure la séance de TP en mettant en commun les recherches des étudiants. Les plus avancés peuvent passer du temps à chercher tout ce qui a été empêcher avec la configuration et rajouter d'autres protections si ils le désirent. Les personnes n'étant pas encore arriver la peuvent donc obtenir la liste des scénarii sans impacter leur avancement, ils auront juste une liste de mails qui doivent être bloqué à la fin du TP et cela pourrait les débloquer ou les aider à mieux comprendre leur problème.

Voici une liste non exhaustive mais que nous pensons complète des différentes mails possible et attaques bloqué grâce aux différentes configurations mise en place tout au long de ce TP. Le format utilisé est celui du fichier `` `/mailscript/scenarii.json ``.

```
{
"mail1" : {
    "Server" : "smtp.isp-a.milxc",
    "ServerPort" : 587,
    "From" : "Hacker<hacker@isp-a.milxc>",
    "To" : "admin@target.milxc",
    "Subject" : "Test 1",
    "Body" : "Email usuel : Hacker vers Admin, doit réussir.",
    "Container" : "isp-a-hacker",
    "ShouldPass" : true
    },

"mail2" : {
    "Server" : "smtp.isp-a.milxc",
    "ServerPort" : 587,
    "From" : "Hacker<hacker@isp-a.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 2",
    "Body" : "Email usuel : Hacker vers User, doit réussir.",
    "Container" : "isp-a-hacker",
    "ShouldPass" : true
    },

"mail3" : {
    "Server" : "smtp.gcorp.milxc",
    "ServerPort" : 587,
    "From" : "User<user@gcorp.milxc>",
    "To" : "admin@target.milxc",
    "Subject" : "Test 3",
    "Body" : "Email usuel : User vers Admin, doit réussir",
    "Container" : "gcorp-user",
    "ShouldPass" : true
    },
```

```
"mail4" : {
    "Server" : "smtp.gcorp.milxc",
    "ServerPort" : 587,
    "From" : "User<user@gcorp.milxc>",
    "To" : "hacker@isp-a.milxc",
    "Subject" : "Test 4",
    "Body" : "Email usuel : User vers Hacker, doit réussir",
    "Container" : "gcorp-user",
    "ShouldPass" : true
    },

"mail5" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 587,
    "From" : "Admin<admin@target.milxc>",
    "To" : "hacker@isp-a.milxc",
    "Subject" : "Test 5",
    "Body" : "Email usuel : Admin vers Hacker, doit réussir",
    "Container" : "target-admin",
    "ShouldPass" : true
    },

"mail6" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 587,
    "From" : "Admin<admin@target.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 6",
    "Body" : "Email usuel : Admin vers User, doit réussir",
    "Container" : "target-admin",
    "ShouldPass" : true
    },

"mail7" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 587,
    "From" : "Admin<admin@target.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 1",
    "Body" : "Email malveillant : Hacker cherche à se faire passer pour Admin en
usurpant son adresse mail et en se connectant directement au serveur mail de
target, ne doit pas réussir",
    "Container" : "isp-a-hacker",
    "ShouldPass" : false
    },

"mail8" : {
    "Server" : "smtp.isp-a.milxc",
    "ServerPort" : 587,
    "From" : "Admin<admin@target.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 2",
    "Body" : "Email malveillant : Hacker cherche à se faire passer pour Admin en
usurpant son adresse mail et en se connectant à son propre serveur mail, ne doit
pas réussir",
    "Container" : "isp-a-hacker",
    "ShouldPass" : false
```

```
    },

"mail9" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 25,
    "From" : "Hacker<hacker@isp-a.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 3",
    "Body" : "Email malveillant : Hacker utilise le serveur mail de target comme
un open relay, ne doit pas réussir",
    "Container" : "isp-a-hacker",
    "ShouldPass" : false
    },

"mail10" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 25,
    "From" : "Hacker<hacker@isp-a.milxc>",
    "To" : "admin@target.milxc",
    "Subject" : "Test 4",
    "Body" : "Email malveillant : Hacker se connecte sur le port 25 pour envoyer
des mails en interne, ne doit pas réussir",
    "Container" : "isp-a-hacker",
    "ShouldPass" : false
    },

"mail11" : {
    "Server" : "smtp.target.milxc",
    "ServerPort" : 587,
    "From" : "Hacker<hacker@isp-a.milxc>",
    "To" : "user@gcorp.milxc",
    "Subject" : "Test 5",
    "Body" : "Email malveillant : Admin cherche à usurper l'adresse mail de
Hacker qui utilise un autre nom de domaine que celui du serveur utilisé, ne doit
pas réussir",
    "Container" : "target-admin",
    "ShouldPass" : false
    },

"mail12" : {
    "Server" : "smtp.isp-a.milxc",
    "ServerPort" : 587,
    "From" : "User<user@gcorp.milxc>",
    "To" : "admin@target.milxc",
    "Subject" : "Test 6",
    "Body" : "Email malveillant : Hacker cherche à se faire passer pour User en
usurpant son adresse mail et en se connectant à son propre serveur mail, ne doit
pas réussir",
    "Container" : "isp-a-hacker",
    "ShouldPass" : false
    }
}
```

Les mails 1 à 6 sont des emails usuels, ils doivent passer sinon il y a une mauvaise configuration dans un ou plusieurs serveurs de mail. Les autres mails sont des attaques, ces dernières sont décrites dans le corps de leur message. Les étudiants peuvent vérifier leurs différents scénarii en utilisant la commande `./mi-lxc.py mail`. Le résultat de cette commande est de la forme

suivante :

```
X NON RECU :
Server : smtp.isp-a.milxc
ServerPort : 587
From : Admin<admin@target.milxc>
To : user@gcorp.milxc
Subject : 4190972219814667_Test 2
Body : Email malveillant : Hacker cherche à se faire passer pour Admin en
usurpant son adresse mail et en se connectant à son propre serveur mail, ne doit
pas réussir
Container : isp-a-hacker
ShouldPass : False
----------------------
V NON RECU :
Server : smtp.target.milxc
ServerPort : 25
From : Hacker<hacker@isp-a.milxc>
To : user@gcorp.milxc
Subject : 5339224124655387_Test 3
Body : Email malveillant : Hacker utilise le serveur mail de target comme un open
relay, ne doit pas réussir
Container : isp-a-hacker
ShouldPass : False
----------------------
V RECU :
Server : smtp.target.milxc
ServerPort : 587
From : Admin<admin@target.milxc>
To : hacker@isp-a.milxc
Subject : 17132834916986484_Test 5
Body : Email usuel : Admin vers Hacker, doit réussir
Container : target-admin
ShouldPass : True
----------------------
X NON RECU :
Server : smtp.gcorp.milxc
ServerPort : 587
From : User<user@gcorp.milxc>
To : hacker@isp-a.milxc
Subject : 22456985223825343_Test 4
Body : Email usuel : User vers Hacker, doit réussir
Container : gcorp-user
ShouldPass : True
```

Pour comprendre ce résultat, voici le tableau suivant :

	Email usuels	**Emails Problématiques**
V	L'émail a été reçu	L'email n'a pas été reçu, il a été bloqué
X	L'email n'a pas été reçu, ce n'est pas normal	L'email a été reçu, ce n'est pas normal