# Script de configuration d'Azure AD pour la sécurité du courrier électronique de Cisco

## Contenu



## Introduction

Ce document fournit un script qui peut être exécuté d'un environnement UNIX/Linux pour simplifier le processus utilisé pour créer un certificat auto-signé et des étapes exigées de Microsoft Azure si nécessaire pour configurer la sécurité du courrier électronique de Cisco. Ce script peut être utilisé connecteur pour de boîte aux lettres LDAP automatique de correction (MARS), de Microsoft Office 365, ou analyseur de menace de Cisco pour le bureau 365. Ce script est indépendant et peut être utilisé avec toutes les versions d'AsyncOS pour l'appliance de sécurité du courrier électronique (ESA).

> **Note**: Cet article est un preuve-de-concept et si comme exemple base. Tandis que ces étapes ont été avec succès testées, cet article est destiné principalement pour des buts de démonstration et d'illustration. Les scripts personnalisés sont en dehors de la portée et de la prise en charge de Cisco. Le centre d'assistance technique Cisco (TAC) n'écrira pas, mettra à jour, ou dépannera les scripts externes à tout moment. Avant que vous tentiez et construisiez tous les scripts, assurez-vous que vous avez la connaissance de script quand vous construisez le script final.

> **Note**: Cisco TAC et le support de Cisco ne sont pas autorisés à dépanner des questions de côté client avec l'AD de Microsoft Exchange, de Microsoft Azure, ou le bureau 365.

## Conditions préalables

### Exigences

Cisco recommande que vous lisiez et comprenniez Comment-[à des configurations configuriez d'Azure AD et de bureau 365 boîte aux lettres pour l'ESA](#).

### [Composants utilisés](#)

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Pour le but et l'exécution de ce script, c'est dans la supposition que vous avez OpenSSL avez

installé. De votre demande terminale, exécutez-vous **qui openssl** ou **version d'openssl** afin de vérifier l'installation.

Afin de cet article, le script s'appellera et sera exécuté comme *my_azure.sh*. Sentez-vous libre de nommer le script comme vous souhaitez.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande.

# Script de configuration d'Azure AD pour la sécurité du courrier électronique de Cisco

D'un hôte externe (UNIX/Linux), créez un script et copiez et collez ce texte :

```
effacez
########################################################################################## d'écho «
      my_azure.sh par Robert Sherwin (robsherw@cisco.com) ©2018 Cisco. : |:. : |:.
Utilisant l'openssl, ce script créera un certificat auto-signé pour que vous utilisiez dedans
commande pour se terminer la configuration de configurations de boîte aux lettres pour la
sécurité du courrier électronique de Cisco.
Répondez s'il vous plaît aux demandes suivantes :
##########################################################################################
"
si quel openssl >/dev/null ; puis
    contrôle d'openssl d'écho « passé : l'openssl est installé ! » et version d'openssl
autrement
    écho « vous ne semblez pas faire installer l'openssl. » sortie de &&
fi

écho «
Veuillez écrire un nom pour votre CERT : "
lisez le my_cert

tandis que [- f $my_cert.key] ;
faites
    fichier d'écho le « existe, satisfait écrivent un nom pour votre CERT : le » && a lu le
my_cert
fait

écho «
Merci.  Les fichiers qui seront générés pour votre CERT sont : "

crt=$my_cert.crt
key=$my_cert.key
pem=$my_cert.pem

écho $crt
écho $key
écho $pem
"" d'écho

tandis que vrai ; faites
    lu - p « êtes-vous prêt à poursuivre et générer ces fichiers pour votre configuration ? $
(tput yn de smso) (y/n)$ (tput sgr0) »
    cas $yn dedans
```

```
        [Yy] *) req -x509 -sha256 d'openssl - Noeuds - jours 1825 - newkey rsa:2048 - keyout
$key - $crt
openssl RSA - dans $key - $key
cat $key $crt > $pem

"" d'écho
base64Thumbprint=`openssl x509 - der d'outform - dans $crt | dgst d'openssl - binaire -sha1 |
openssl base64`
base64Value=`openssl x509 - der d'outform - dans $crt | openssl base64 - Un `
python de `de keyid= - uuid d'importation c « ; print(uuid.uuid4())"`
écho «
################################################################################
Ensuite, $ (smul de tput) copy$ (rmul de tput) le suivant à Azure pour votre manifeste :
################################################################################
"
« d'écho \ « keyCredentials \ » : [
{
\ « customKeyIdentifier \ » : \"$base64Thumbprint\",
\ « keyId \ » : \ « $keyid \ »,
\ « type \ » : \"AsymmetricX509Cert\",
\ « utilisation \ » : \ « vérifiez \ »,
\ « valeur \ » : \"$base64Value\"
}
], »
écho «
################################################################################
Puis $ (smul de tput) complete$ (rmul de tput) la configuration azurée pour obtenir le client
ID$ (tput sgr0) $ (smso de tput) et le locataire ID$ (tput sgr0) $ (smso de tput).
################################################################################
"
faites écho « ceci est le $ (smso de tput) Thumbprint$ (tput sgr0) pour votre configuration ESA
: $base64Thumbprint"
faites écho « ceci est le certificat $ (smso de tput) Key$ privé (tput sgr0) pour votre
configuration ESA : $pem
« ; rupture ; ;
        [Nn] *) sortie ; ;
        *) l'écho « répondent s'il vous plaît oui ou non » ; ;
    esac
fait
tandis que vrai ; faites
    lu - p « souhaitez-vous examiner ce certificat en détail ? $ (tput yn de smso) (y/n)$ (tput
sgr0) »
    cas $yn dedans
        [Yy] *) openssl x509 - dans $crt - texte ; écho «
Merci ! » rupture de && ; ;
        [Nn] *) l'écho « vous remercient ! » sortie de && ; ;
        *) l'écho « répondent s'il vous plaît oui ou non » ; ;
    esac
fait
```

> **Conseil** : Une fois que vous avez écrit le script, écrivez le **<script_name> du chmod u+x** afin de rendre le script exécutable.

Un exemple complet du script dans l'action devrait avoir comme conséquence :

```
my_host$ ./my_azure
################################################################################
      my_azure.sh par Robert Sherwin (robsherw@cisco.com) ©2018 Cisco. : |:. : |:.
Utilisant l'openssl, ce script créera un certificat auto-signé pour que vous utilisiez dedans
commande pour se terminer la configuration de configurations de boîte aux lettres pour la
sécurité du courrier électronique de Cisco.
```

```
Répondez s'il vous plaît aux demandes suivantes :
################################################################################

contrôle d'openssl passé : l'openssl est installé !
LibreSSL 2.2.7

Veuillez écrire un nom pour votre CERT :
technote_example

Merci.  Les fichiers qui seront générés pour votre CERT sont :
technote_example.crt
technote_example.key
technote_example.pem

Êtes-vous prêt à poursuivre et générer ces fichiers pour votre configuration ? (y/n) y
Générer une clé privée de 2048 bits RSA
.......................................................... +++
........................................ +++
écrivant la nouvelle clé privée à « technote_example.key »
-----
Vous êtes sur le point d'être invité à écrire les informations qui seront incorporées
dans votre demande de certificat.
Ce que vous êtes sur le point d'entrer dans est ce qui s'appelle un nom unique ou un DN.
Il y a tout à fait quelques champs mais vous pouvez laisser un certain blanc
Pour quelques champs il y aura une valeur par défaut,
Si vous entrez « .  », le champ sera blanc de gauche.
-----
Nom du pays (2 marquent avec des lettres le code) [] : LES USA
Nom d'état ou de province (nom complet) [] : La Caroline du Nord
Nom de localité (par exemple, ville) [] : RTP
Nom d'organisation (par exemple, société) [] : Cisco
Nom d'unité organisationnelle (par exemple, section) [] : Exemple Service.
Nom commun (par exemple, entièrement - nom d'hôte qualifié) [] : example.local
Adresse e-mail [] : joe.user@example.local
inscription de la clé RSA



########################################################################
Ensuite, copiez le suivant sur Azure pour votre manifeste :
########################################################################

« keyCredentials » : [
{
« customKeyIdentifier » : « wWHhkWEfuhDHTXPzzmHoSEnjbNM= »,
« keyId » : "338836b8-fc8d-4e1b-9a3f-b252f8368d34",
« type » : "AsymmetricX509Cert",
« utilisation » : « Vérifiez »,
« valeur » :
"MIIDtDCCApwCCQDV3bbiHmaN2jANBgkqhkiG9w0BAQsFADCBmzELMAkGA1UEBhMCVVMxFzAVBgNVBAgMDk5vcnRoIENhcm9
saW5hMQwwCgYDVQQHDANSVFAxDjAMBgNVBAoMBUNpc2NvMRYwFAYDVQQLDA1FeGFtcGxlIERlcHQuMRYwFAYDVQQDDA1leGF
tcGxlLmxvY2FsMSUwIwYJKoZIhvcNAQkBFhZqb2UudXNlckBleGFtcGxlLmxvY2FsMB4XDTE4MTAxODAyMDA0OVoXDTIzMTA
xNzAyMDA0OVowgZsxCzAJBgNVBAYTAlVTMRcwFQYDVQQIDA5Ob3J0aCBDYXJvbGluYTEMMAoGA1UEBwwDUlRQMQ4wDAYDVQQ
KDAVDaXNjbzEWMBQGA1UECwwNRXhhbXBsZSBEZXB0LjEWMBQGA1UEAwwNZXhhbXBsZS5sb2NhbDElMCMGCSqGSIb3DQEJARY
Wam9lLnVzZXJAZXhhbXBsZS5sb2NhbDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKlYmW7DN+AxcZQcpc8hZhm
v9yqMHul2cjV3G088mkGtRZU5KUVNKZZSmMlny3lOKg6cTu4Ez4UuigzC/2JXEf3+wOj9YChK92bEYWjYsKeZtbIoqYRfHE+
Sk+bsJb5GpizXgPcYZGje81ecgamhDrg7NZrthPTSKa4ZxmYwpQl6xGDrMipolGoENf+eyNCo5VyAXlxuYH8m6t0GdPw+VKH
J7k+4wI9KTUw4LABoOWs8hUnDi0yz2k9mqNvTG+u75EUUMgcTWc/ISsXjC8kpbOsxteZIiU4xUvqNd1t96iccjad19n61Jds
wGX+CC1Pl+ZZMk8/IQEPtbPqs/4p3cmECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAQqq7ixBbtfhorrWk73uCoYUPRqWZLKH
lgs1UpEnmPjvLZiImY+O6kiR9icDVjFD47AW+0vYg3pHt6pKWl7TUZpilz4hNp0oYc/qjd6aCA8B2KMmbfh2DVhmpYWW8P7w
bNP/im3114F/zJvBVnHjeaY9KsuTUU54Wb8VX2FFX40/YFm/HTHrXcHHyWy5XBU9MFVmEu+Yv6JIxCaEgj5J7jV4qGQM++fn
+EpRPkVHn844Hzgxm40bRW747rjGuyKss+E2tjWJT6OmDJ4ruHCFdvhkZvvzVJyVn0PVN+cwoJ0gLM7p2oa7J3IdNZ3p2CMX
vFdZsRiFFUpBIbK3VYlFRrg=="
```

```
}
],

################################################################################
Terminez-vous alors la configuration azurée pour obtenir l'ID de client et l'ID de locataire.
################################################################################

C'est le Thumbprint pour votre configuration ESA : wWHhkWEfuhDHTXPzzmHoSEnjbNM=
C'est la clé privée de certificat pour votre configuration ESA : technote_example.pem
```

Le script vous incitera à examiner le certificat en détail. Écrivez **y** ou **n** afin de remplir le script.

```
Souhaitez-vous examiner ce certificat en détail ? (y/n) y
Certificat :
    Données :
        Version : 1 (0x0)
        Numéro de série : 1541067458222060693 8 (0xd5ddb6e21e668dda)
    Algorithme de signature : sha256WithRSAEncryption
        Émetteur : C=US, ST= la Caroline du Nord, L=RTP, O=Cisco, service d'OU=Example,
CN=example.local/emailAddress= joe.user@example.local
        Validité
            Pas avant : 20h18 GMT du 18 octobre 02:00:49
            Pas ensuite : 20h23 GMT du 17 octobre 02:00:49
        Objet : C=US, ST= la Caroline du Nord, L=RTP, O=Cisco, service d'OU=Example,
CN=example.local/emailAddress= joe.user@example.local
        L'information principale publique soumise :
            Algorithme de clé publique : rsaEncryption
                Clé publique : (bit 2048)
                Module :
                    00:a9:58:99:6e:c3:37:e0:31:71:94:1c:a5:cf:21 :
                    66:19:af:f7:2a:8c:1e:e9:76:72:35:77:1b:4f:3c :
                    9a:41:ad:45:95:39:29:45:4d:29:96:52:98:c9:67 :
                    cb:79:4e:2a:0e:9c:4e:ee:04:cf:85:2e:8a:0c:c2 :
                    ff:62:57:11:fd:fe:c0:e8:fd:60:28:4a:f7:66:c4 :
                    61:68:d8:b0:a7:99:b5:b2:28:a9:84:5f:1c:4f:92 :
                    93:e6:ec:25:be:46:a6:2c:d7:80:f7:18:64:68:de :
                    f3:57:9c:81:a9:a1:0e:b8:3b:35:9a:ed:84:f4:d2 :
                    29:ae:19:c6:66:30:a5:09:7a:c4:60:eb:32:2a:68 :
                    94:6a:04:35:ff:9e:c8:d0:a8:e5:5c:80:5e:5c:6e :
                    60:7f:26:ea:dd:06:74:fc:3e:54:a1:c9:ee:4f:b8 :
                    c0:8f:4a:4d:4c:38:2c:00:68:39:6b:3c:85:49:c3 :
                    8b:4c:b3:da:4f:66:a8:db:d3:1b:eb:bb:e4:45:14 :
                    32:07:13:59:cf:c8:4a:c5:e3:0b:c9:29:6c:eb:31 :
                    b5:e6:48:89:4e:31:52:fa:8d:77:5b:7d:ea:27:1c :
                    8d:a7:75:f6:7e:b5:25:db:30:19:7f:82:0b:53:e5 :
                    f9:96:4c:93:cf:c8:40:43:ed:6c:fa:ac:ff:8a:77 :
                    72:61
                Exposant : 65537 (0x10001)
    Algorithme de signature : sha256WithRSAEncryption
         42:aa:bb:8b:10:5b:b5:f8:68:ae:b5:a4:ef:7b:82:a1:85:0f :
         46:a5:99:2c:a1:e5:82:cd:54:a4:49:e6:3e:3b:cb:66:22:26 :
         63:e3:ba:92:24:7d:89:c0:d5:8c:50:f8:ec:05:be:d2:f6:20 :
         de:91:ed:ea:92:96:97:b4:d4:66:98:a5:cf:88:4d:a7:4a:18 :
         73:fa:a3:77:a6:82:03:c0:76:28:c9:9b:7e:1d:83:56:19:a9 :
         61:65:bc:3f:bc:1b:34:ff:e2:9b:7d:75:e0:5f:f3:26:f0:55 :
         9c:78:de:69:8f:4a:b2:e4:d4:53:9e:16:6f:c5:57:d8:51:57 :
         e3:4f:d8:16:6f:c7:4c:7a:d7:70:71:f2:5b:2e:57:05:4f:4c :
         15:59:84:bb:e6:2f:e8:92:31:09:a1:20:8f:92:7b:8d:5e:2a :
         19:03:3e:f9:f9:fe:12:94:4f:91:51:e7:f3:8e:07:ce:0c:66 :
         e3:46:d1:5b:be:3b:ae:31:ae:c8:ab:2c:f8:4d:ad:8d:62:53 :
         e8:e9:83:27:8a:ee:1c:21:5d:be:19:19:be:fc:d5:27:25:67 :
         d0:f5:4d:f9:cc:28:27:48:0b:33:ba:76:a1:ae:c9:dc:87:4d :
         67:7a:76:08:c5:ef:15:d6:6c:46:21:45:52:90:48:6c:ad:d5 :
```

```
        62:51:51:ae
-----COMMENCEZ LE CERTIFICAT-----
MIIDtDCCApwCCQDV3bbiHmaN2jANBgkqhkiG9w0BAQsFADCBmzELMAkGA1UEBhMC
VVMxFzAVBgNVBAgMDk5vcnRoIENhcm9saW5hMQwwCgYDVQQHDANSVFAxDjAMBgNV
BAoMBUNpc2NvMRYwFAYDVQQLDA1FeGFtcGxlIERlcHQuMRYwFAYDVQQDDA1leGFt
cGxlLmxvY2FsMSUwIwYJKoZIhvcNAQkBFhZqb2UudXNlckBleGFtcGxlLmxvY2Fs
MB4XDTE4MTAxODAyMDA0OVoXDTIzMTAxNzAyMDA0OVowgZsxCzAJBgNVBAYTAlVT
MRcwFQYDVQQIDA5Ob3J0aCBDYXJvbGluYTEMMAoGA1UEBwwDUlRQMQ4wDAYDVQQK
DAVDaXNjbzEWMBQGA1UECwwNRXhhbXBsZSBEZXB0LjEWMBQGA1UEAwwNZXhhbXBs
ZS5sb2NhbDElMCMGCSqGSIb3DQEJARYWam9lLnVzZXJAZXhhbXBsZS5sb2NhbDCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKlYmW7DN+AxcZQcpc8hZhmv
9yqMHul2cjV3G088mkGtRZU5KUVNKZZSmMlny3lOKg6cTu4Ez4UuigzC/2JXEf3+
wOj9YChK92bEYWjYsKeZtbIoqYRfHE+Sk+bsJb5GpizXgPcYZGje81ecgamhDrg7
NZrthPTSKa4ZxmYwpQl6xGDrMipolGoENf+eyNCo5VyAXlxuYH8m6t0GdPw+VKHJ
7k+4wI9KTUw4LABoOWs8hUnDi0yz2k9mqNvTG+u75EUUMgcTWc/ISsXjC8kpbOsx
teZIiU4xUvqNd1t96iccjad19n61JdswGX+CC1Pl+ZZMk8/IQEPtbPqs/4p3cmEC
AwEAATANBgkqhkiG9w0BAQsFAAOCAQEAQqq7ixBbtfhorrWk73uCoYUPRqWZLKHl
gs1UpEnmPjvLZiImY+O6kiR9icDVjFD47AW+0vYg3pHt6pKWl7TUZpilz4hNp0oY
c/qjd6aCA8B2KMmbfh2DVhmpYWW8P7wbNP/im3114F/zJvBVnHjeaY9KsuTUU54W
b8VX2FFX40/YFm/HTHrXcHHyWy5XBU9MFVmEu+Yv6JIxCaEgj5J7jV4qGQM++fn+
EpRPkVHn844Hzgxm40bRW747rjGuyKss+E2tjWJT6OmDJ4ruHCFdvhkZvvzVJyVn
0PVN+cwoJ0gLM7p2oa7J3IdNZ3p2CMXvFdZsRiFFUpBIbK3VYlFRrg==
-----CERTIFICAT D'EXTRÉMITÉ-----

Merci !
```

Àce moment, vous avez trois fichiers : .crt, .key, et .pem.

Utilisez les *keyCredentials* sortis comme instruit, et copiez la sortie sur Azure quand vous installez l'enregistrement d'app. *Le Thumbprint* a sorti et la *clé privée de certificat* (.pem) sont nécessaire quand vous exécutez les étapes de configuration sur la sécurité du courrier électronique de Cisco.

## Informations connexes

- Appliance de sécurité du courrier électronique de Cisco - Guides d'utilisateur
- Support et documentation techniques - Cisco Systems