

Configuration de PIX 5.0.x : TACACS+ et RADIUS

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Authentification contre l'autorisation](#)

[Sur ce que l'utilisateur voit avec l'Authentification/Autorisation activée](#)

[Configurations de serveur sécurisé utilisées pour tous les scénarios](#)

[Configuration de serveur TACACS de Cisco Secure UNIX](#)

[Configuration du serveur RADIUS de Cisco Secure UNIX](#)

[RAYON Cisco Secure de Windows 2.x](#)

[EasyACS TACACS+](#)

[2.x Cisco Secure TACACS+](#)

[Configuration du serveur Livingston RADIUS](#)

[Configuration du serveur Merit RADIUS](#)

[Étapes de débogage](#)

[Diagramme du réseau](#)

[Exemples de debug d'authentification des exemples de debug de PIX](#)

[Sortant](#)

[D'arrivée](#)

[Debug PIX - Bonne authentification - TACACS+](#)

[Debug PIX - Authentification erronée \(nom d'utilisateur ou mot de passe\) - TACACS+](#)

[PIX mettent au point - Ne peut cingler le serveur, aucune réponse - TACACS+](#)

[Debug PIX - Incapable de cingler le serveur - TACACS+](#)

[Debug PIX - Bonne authentification - RAYON](#)

[Debug PIX - Authentification erronée \(nom d'utilisateur ou mot de passe\) - RAYON](#)

[Debug de ping - Peut cingler le serveur, le démon vers le bas - RAYON](#)

[Debug PIX - Incapable de cingler le serveur ou de les introduire/non-concordance de client - RAYON](#)

[Ajoutez l'autorisation](#)

[Exemples de debug d'authentification et d'autorisation de PIX](#)

[Debug PIX - Bonne authentification et autorisation réussie - TACACS+](#)

[Debug PIX - Bonne authentification, autorisation défailante - TACACS+](#)

[Ajoutez la gestion des comptes](#)

[TACACS+](#)

[RAYON](#)

[Utilisation de excepté commande](#)

[Maximum-sessions et utilisateurs connectés de visionnement](#)

[Authentification et activation sur le PIX lui-même](#)

[Authentification sur la console série](#)

[Changez la demande que les utilisateurs voient](#)

[Personnalisez les utilisateurs de message voient sur le succès/panne](#)

[Inactif et temporisations absolues de Par-utilisateur](#)

[HTTP virtuel](#)

[Diagramme sortant de HTTP virtuel](#)

[HTTP virtuel de configuration PIX sortant](#)

[Telnet virtuel](#)

[Diagramme d'arrivée de telnet virtuel](#)

[Configuration Virtual Telnet PIX d'arrivée](#)

[Telnet virtuel de configuration utilisateur de serveur TACACS+ d'arrivée](#)

[Telnet virtuel de debug PIX d'arrivée](#)

[Telnet virtuel sortant](#)

[Configuration Virtual Telnet PIX sortante](#)

[Telnet virtuel de debug PIX sortant](#)

[Déconnexion virtuelle de Telnet](#)

[Autorisation sur le port](#)

[Configuration PIX](#)

[Configuration du serveur de logiciel gratuit TACACS+](#)

[Debug sur le PIX](#)

[AAA expliquant le trafic autre que le HTTP, le FTP, et le telnet](#)

[Informations connexes](#)

Introduction

L'authentification de RAYON et TACACS+ peut être faite pour le FTP, le telnet, et les connexions HTTP. L'authentification pour d'autres protocoles TCP moins communs peut habituellement être faite fonctionner.

L'autorisation TACACS+ est prise en charge. L'autorisation RADIUS n'est pas. Les changements de l'Authentification, autorisation et comptabilité (AAA) PIX 5.0 au-dessus de la version préalable incluent l'AAA expliquant le trafic autre que le HTTP, le FTP, et le telnet.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

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

Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

Authentification contre l'autorisation

- L'authentification est qui l'utilisateur est.
- Est l'autorisation ce que l'utilisateur peut faire.
- L'authentification *est* valide sans autorisation.
- L'autorisation est *non valide* sans authentification.

Comme exemple, supposez vous avez cent utilisateurs intérieurs et vous voulez seulement que six de ces utilisateurs puisse faire le FTP, le telnet, ou le HTTP en dehors du réseau. Dites le PIX d'authentifier le trafic sortant et de donner à chacun des six utilisateurs des id sur le serveur sécurisé TACACS+/RADIUS. Avec l'*authentification* simple, ces six utilisateurs peuvent être authentifiés avec le nom d'utilisateur et mot de passe, puis sortent. Les quatre-vingt-quatorze autres utilisateurs ne peuvent pas sortir. Le PIX incite des utilisateurs pour le nom d'utilisateur/mot de passe, puis passe leur nom d'utilisateur et mot de passe au serveur sécurisé TACACS+/RADIUS. Selon la réponse, il ouvre ou refuse la connexion. Ces six utilisateurs peuvent faire le FTP, le telnet, ou le HTTP.

D'autre part, assumez *un de* ces trois utilisateurs, « Terry, » n'est pas à sont de confiance. Vous voudriez permettre à Terry pour faire le FTP, mais pas le HTTP ou le telnet à l'extérieur. Ceci vous signifie le besoin d'ajouter l'*autorisation*. C'est-à-dire, autorisant *ce que les* utilisateurs peuvent faire en plus d'authentifier *qui* ils sont. Quand vous ajoutez l'*autorisation* au PIX, le PIX d'abord envoie le nom d'utilisateur et mot de passe de Terry au serveur sécurisé, alors envoie à une demande d'autorisation indiquant au serveur sécurisé ce que la « *commande* » Terry essaye de faire. Avec la configuration du serveur correctement, Terry peut être permis au « FTP 1.2.3.4 » mais est refusé la capacité au « HTTP » ou au « telnet » n'importe où.

Sur ce que l'utilisateur voit avec l'Authentification/Autorisation activée

Quand vous essayez d'aller de l'intérieur à l'extérieur (ou vice versa) avec l'authentification/autorisation en fonction :

- **Telnet** - L'utilisateur voit un affichage de l'invite nom d'utilisateur, suivi d'une demande pour le mot de passe. Si l'authentification (et l'autorisation) est réussie au PIX/serveur, l'utilisateur est incité pour le nom d'utilisateur et mot de passe par la destination host au-delà.
- **FTP** - L'utilisateur voit une invite de nom d'utilisateur être soulevé. Les besoins de l'utilisateur d'écrire « local_username@remote_username » pour le nom d'utilisateur et le « local_password@remote_password » pour le mot de passe. Le PIX envoie le « local_username » et le « local_password » au serveur de sécurité local, et si l'authentification (et l'autorisation) est réussie au PIX/serveur, le « remote_username » et le « remote_password » sont passés au serveur FTP de destination au-delà.
- **HTTP** - Une fenêtre affichée dans le navigateur qui demande le nom d'utilisateur et mot de passe. Si l'authentification (et l'autorisation) est réussie, l'utilisateur arrive au site Web de destination au-delà. Maintenez dans l'esprit que les **navigateurs cachent des noms**

d'utilisateur et mot de passe. S'il s'avère que le PIX devrait chronométrer une connexion HTTP mais ne fait pas ainsi, il est probable que la ré-authentification réellement ait lieu avec le navigateur « tir » le nom d'utilisateur en cache et le mot de passe au PIX, qui puis en avant ceci au serveur d'authentification. Le Syslog et/ou le serveur PIX mettent au point afficheront ce phénomène. Si le telnet et le FTP semblent fonctionner normalement, mais les connexions HTTP ne font pas, c'est pourquoi.

[Configurations de serveur sécurisé utilisées pour tous les scénarios](#)

[Configuration de serveur TACACS de Cisco Secure UNIX](#)

Assurez-vous que vous avez l'adresse IP PIX ou le nom de domaine complet et la clé dans le fichier CSU.cfg.

```
user = ddunlap {
password = clear "rtp"
default service = permit
}
```

```
user = can_only_do_telnet {
password = clear "telnetonly"
service = shell {
cmd = telnet {
permit .*
}
}
}
```

```
user = can_only_do_ftp {
password = clear "ftponly"
service = shell {
cmd = ftp {
permit .*
}
}
}
```

```
user = httponly {
password = clear "httponly"
service = shell {
cmd = http {
permit .*
}
}
}
```

[Configuration du serveur RADIUS de Cisco Secure UNIX](#)

Employez l'interface utilisateur graphique (GUI) pour ajouter l'IP PIX et la clé à la liste de serveur d'accès à distance (NAS).

```
user=adminuser {
radius=Cisco {
check_items= {
2="all"
}
}
```

```
reply_attributes= {  
6=6  
}  
}
```

[RAYON Cisco Secure de Windows 2.x](#)

Suivez ces étapes :

1. Obtenez un mot de passe dans le partie Interface graphique d'installation des utilisateurs.
2. De la section GUI de Group Setup, placez l'attribut 6 (type de service) pour ouvrir une session ou administratif.
3. Ajoutez l'IP PIX dans le GUI de configuration de NAS.

[EasyACS TACACS+](#)

La documentation d'EasyACS décrit l'installation.

1. Dans la section de groupe, **exécutif de shell de clic** (pour donner des privilèges EXEC).
2. Pour ajouter l'autorisation au PIX, le clic **refusent des commandes IOS inégalées** au bas de l'installation de groupe.
3. **Nouvelle commande d'Add/Edit** choisi pour chaque commande que vous souhaitez permettre (par exemple, telnet).
4. Si vous voulez permettre le telnet aux sites spécifiques, écrivez l'IP dans l'argument section sous la forme la « autorisation #.#.#.# ». Pour permettre le telnet à tous les sites, le clic **permettent tous les arguments non listés**.
5. Cliquez sur Finish la **commande de retouche**.
6. Exécutez les étapes 1 à 5 pour chacune des commandes permises (par exemple, telnet, HTTP, ou FTP).
7. Ajoutez l'IP PIX dans la section GUI de configuration de NAS.

[2.x Cisco Secure TACACS+](#)

L'utilisateur obtient un mot de passe dans le partie Interface graphique d'installation des utilisateurs.

1. Dans la section de groupe, **exécutif de shell de clic** (pour donner des privilèges EXEC).
2. Pour ajouter l'autorisation au PIX, le clic **refusent des commandes IOS inégalées** au bas de l'installation de groupe.
3. **Nouvelle commande d'Add/Edit** choisi pour chaque commande que vous voulez permettre (par exemple, telnet).
4. Si vous voulez permettre le telnet aux sites spécifiques, écrivez l'IP d'autorisation dans l'argument rectangle (par exemple, « autorisation 1.2.3.4"). Pour permettre le telnet à tous les sites, le clic **permettent tous les arguments non listés**.
5. **Commande de retouche de finition de clic**.
6. Exécutez les étapes précédentes pour chacune des commandes permises (par exemple, telnet, HTTP et/ou FTP).
7. Ajoutez l'IP PIX dans la section GUI de configuration de NAS.

[Configuration du serveur Livingston RADIUS](#)

Ajoutez l'IP PIX et la clé aux clients classent.

```
adminuser Password="all"  
User-Service-Type = Shell-User
```

Configuration du serveur Merit RADIUS

Ajoutez l'IP PIX et la clé aux clients classent.

```
adminuser Password="all"  
Service-Type = Shell-User key = "cisco"
```

```
user = adminuser {  
login = cleartext "all"  
default service = permit  
}
```

```
user = can_only_do_telnet {  
login = cleartext "telnetonly"  
cmd = telnet {  
permit .*  
}  
}
```

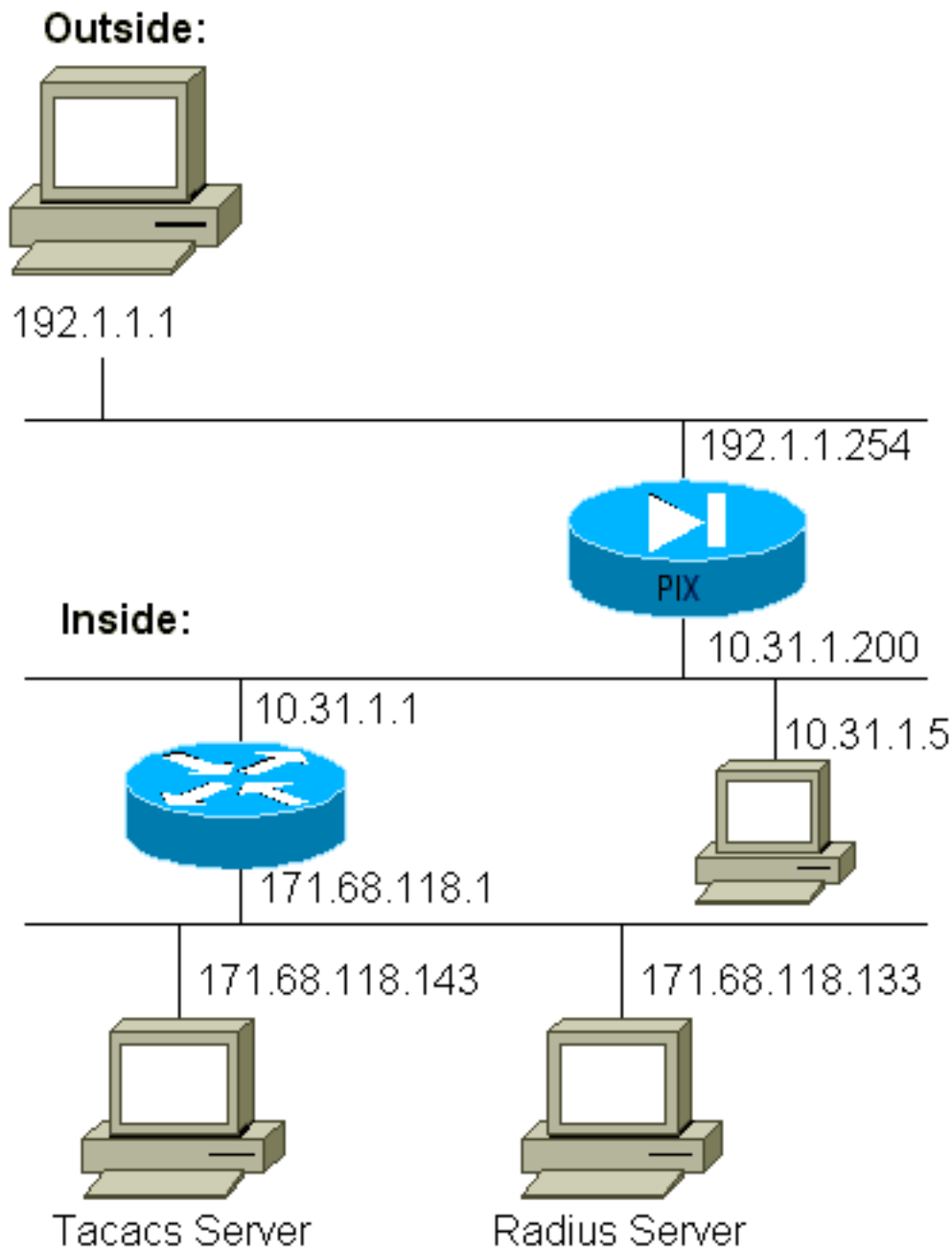
```
user = httponly {  
login = cleartext "httponly"  
cmd = http {  
permit .*  
}  
}
```

```
user = can_only_do_ftp {  
login = cleartext "ftponly"  
cmd = ftp {  
permit .*  
}  
}
```

Étapes de débogage

- Assurez-vous que les configurations PIX fonctionnent avant que vous ajoutiez l'AAA. Si vous ne pouvez pas passer le trafic avant d'instituer l'authentification et l'autorisation, vous ne pourrez pas faire tellement après.
- Enable ouvrant une session le PIX La commande de **logging console debugging** *ne devrait pas* être utilisée sur un système chargé lourd. La commande de **logging buffered debugging** peut être utilisée. La sortie du **show logging** ou des commandes de **se connecter** peut être envoyée à un serveur de Syslog et être examinée.
- Assurez-vous que le débogage est allumé pour les serveurs TACACS+ ou de RAYON. Tous les serveurs ont cette option.

Diagramme du réseau



Configuration PIX

```

pix-5# write terminal nameif ethernet0 outside security0
nameif ethernet1 inside security100 enable password
8Rv2YjIyt7RRXU24 encrypted passwd 2KFQbnNIdI.2KYOU
encrypted hostname pixfirewall fixup protocol ftp 21
fixup protocol http 80 fixup protocol smtp 25 fixup
protocol h323 1720 fixup protocol rsh 514 fixup protocol
sqlnet 1521 names name 1.1.1.1 abcd name 1.1.1.2
a123456789 name 1.1.1.3 a123456789123456 pager lines 24
logging timestamp no logging standby logging console
debugging no logging monitor logging buffered debugging
no logging trap logging facility 20 logging queue 512
interface ethernet0 auto interface ethernet1 auto mtu
outside 1500 mtu inside 1500 ip address outside
192.1.1.254 255.255.255.0 ip address inside 10.31.1.200
255.255.255.0 no failover failover timeout 0:00:00
failover ip address outside 0.0.0.0 failover ip address
inside 0.0.0.0 arp timeout 14400 global (outside) 1
192.1.1.10-192.1.1.20 netmask 255.255.255.0 static
(inside,outside) 192.1.1.25 171.68.118.143 netmask
255.255.255.255 0 0 static (inside,outside) 192.1.1.30

```

```
10.31.1.5 netmask 255.255.255.255 0 0 conduit permit tcp
any any conduit permit icmp any any conduit permit udp
any any no rip outside passive no rip outside default no
rip inside passive no rip inside default route inside
171.68.118.0 255.255.255.0 10.31.1.1 1 timeout xlate
3:00:00 conn 1:00:00 half-closed 0:10:00 udp 0:02:00
timeout rpc 0:10:00 h323 0:05:00 timeout uauth 0:00:00
absolute aaa-server TACACS+ protocol tacacs+ aaa-server
RADIUS protocol radius aaa-server AuthInbound protocol
tacacs+ aaa-server AuthInbound (inside) host
171.68.118.143 cisco timeout 5 aaa-server AuthOutbound
protocol radius aaa-server AuthOutbound (inside) host
171.68.118.133 cisco timeout 5 aaa authentication telnet
outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
aaa authentication telnet inbound 0.0.0.0 0.0.0.0
0.0.0.0 0.0.0.0 AuthInbound aaa authentication http
outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
aaa authentication http inbound 0.0.0.0 0.0.0.0 0.0.0.0
0.0.0.0 AuthInbound aaa authentication ftp outbound
0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound aaa
authentication ftp inbound 0.0.0.0 0.0.0.0 0.0.0.0
0.0.0.0 AuthInbound no snmp-server location no snmp-
server contact snmp-server community public no snmp-
server enable traps telnet timeout 5 terminal width 80
Cryptochecksum:fef4bfc9801d7692dce0cf227fe7859b : end
```

Exemples de debug d'authentification des exemples de debug de PIXAuthentification de PIX

Dans ces derniers mettez au point les exemples :

Sortant

L'utilisateur intérieur aux initiés de 10.31.1.5 trafiquent à 192.1.1.1 extérieur et sont authentifiés par TACACS+. La liste « AuthOutbound » de serveur d'utilisations du trafic sortant qui inclut le serveur 171.68.118.133 de RAYON.

D'arrivée

L'utilisateur externe aux initiés de 192.1.1.1 trafiquent à 10.31.1.5 intérieur (192.1.1.30) et sont authentifiés par TACACS. Liste d'arrivée « AuthInbound » de serveur d'utilisations du trafic qui inclut le serveur TACACS 171.68.118.143).

Debug PIX - Bonne authentification - TACACS+

Cet exemple affiche qu'un PIX met au point avec la bonne authentification :

```
pixfirewall# 109001: Auth start for user "???" from 192.1.1.1/13155
to 10.31.1.5/23
109011: Authen Session Start: user 'pixuser', sid 6
109005: Authentication succeeded for user 'pixuser' from 10.31.1.5/23
to 192.1.1.1/13155
109012: Authen Session End: user 'pixuser', Sid 6, elapsed 1 seconds
302001: Built inbound TCP connection 6 for faddr 192.1.1.1/13155
gaddr 192.1.1.30/23 laddr 10.31.1.5/23 (pixuser)
```


[Debug PIX - Authentification erronée \(nom d'utilisateur ou mot de passe\) - TACACS+](#)

Cet exemple affiche que PIX mettent au point avec l'authentification erronée (nom d'utilisateur ou mot de passe). L'utilisateur voit quatre positionnements de nom d'utilisateur/mot de passe et erreur du message « : nombre maximum d'essais dépassés. »

```
pixfirewall# 109001: Auth start for user '???' from 192.1.1.1/13157
to 10.31.1.5/23
109006: Authentication failed for user '' from 10.31.1.5/23
to 192.1.1.1/13157
```

[PIX mettent au point - Ne peut cingler le serveur, aucune réponse - TACACS+](#)

Cet exemple affiche que PIX mettent au point où le serveur peut être cinglé mais ne parle pas au PIX. L'utilisateur voit le nom d'utilisateur une fois, mais PIX ne demande jamais un mot de passe (c'est sur le telnet). L'utilisateur voit la « erreur : Nombre maximum d'essais dépassés. »

```
Auth start for user '???' from 192.1.1.1/13159 to
10.31.1.5/23
pixfirewall# 109002: Auth from 10.31.1.5/23 to 192.1.1.1/13159
failed (server 171.68.118.143 failed)
109002: Auth from 10.31.1.5/23 to 192.1.1.1/13159 failed
(server 171.68.118.143 failed)
109002: Auth from 10.31.1.5/23 to 192.1.1.1/13159 failed
(server 171.68.118.143 failed)
109006: Authentication failed for user '' from 10.31.1.5/23
to 192.1.1.1/13159
```

[Debug PIX - Incapable de cingler le serveur - TACACS+](#)

Cet exemple affiche qu'un PIX met au point où le serveur n'est pas pingable. L'utilisateur voit le nom d'utilisateur une fois, mais le PIX ne demande jamais un mot de passe (c'est sur le telnet). Ces messages sont affichés : « Délai d'attente au serveur TACACS+ » et à la « erreur : Nombre maximum d'essais dépassés » (nous avons permuté dedans un serveur faux dans la configuration).

```
109001: Auth start for user '???' from 192.1.1.1/13158
to 10.31.1.5/23
109002: Auth from 10.31.1.5/23 to 192.1.1.1/13158 failed
(server 171.68.118.143 failed)
109002: Auth from 10.31.1.5/23 to 192.1.1.1/13158 failed
(server 171.68.118.143 failed)
109002: Auth from 10.31.1.5/23 to 192.1.1.1/13158 failed
(server 171.68.118.143 failed)
109006: Authentication failed for user '' from 10.31.1.5/23
to 192.1.1.1/13158
```

[Debug PIX - Bonne authentification - RAYON](#)

Cet exemple affiche qu'un PIX met au point avec la bonne authentification :

```
109001: Auth start for user '???' from 10.31.1.5/11074
to 192.1.1.1/23
109011: Authen Session Start: user 'pixuser', Sid 7
109005: Authentication succeeded for user 'pixuser'
from 10.31.1.5/11074 to 192.1.1.1/23
109012: Authen Session End: user 'pixuser', Sid 7,
elapsed 1 seconds
302001: Built outbound TCP connection 7 for faddr 192.1.1.1/23
```

```
gaddr 192.1.1.30/11074 laddr 10.31.1.5/11074 (pixuser)
```

[Debug PIX - Authentification erronée \(nom d'utilisateur ou mot de passe\) - RAYON](#)

Cet exemple affiche qu'un PIX met au point avec l'authentification erronée (nom d'utilisateur ou mot de passe). L'utilisateur voit une demande pour le nom d'utilisateur et mot de passe. L'utilisateur a trois occasions pour l'entrée réussie de nom d'utilisateur/mot de passe.

```
- 'Error: max number of tries exceeded'
pixfirewall# 109001: Auth start for user '???' from
 192.1.1.1/13157 to 10.31.1.5/23
109001: Auth start for user '???' from 10.31.1.5/11075
  to 192.1.1.1/23
109002: Auth from 10.31.1.5/11075 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109002: Auth from 10.31.1.5/11075 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109002: Auth from 10.31.1.5/11075 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109006: Authentication failed for user '' from 10.31.1.5/11075
  to 192.1.1.1/23
```

[Debug de ping - Peut cingler le serveur, le démon vers le bas - RAYON](#)

Cet exemple affiche qu'un PIX met au point où le serveur fait l'objet d'un ping, mais le démon est vers le bas et ne communiquera pas avec le PIX. L'utilisateur voit que le nom d'utilisateur, le mot de passe, et serveur de RAYON des messages le « ont manqué » et « erreur : Nombre maximum d'essais dépassés. »

```
pixfirewall# 109001: Auth start for user '???'
  from 10.31.1.5/11076 to 192.1.1.1/23
109002: Auth from 10.31.1.5/11076 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109002: Auth from 10.31.1.5/11076 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109002: Auth from 10.31.1.5/11076 to 192.1.1.1/23 failed
 (server 171.68.118.133 failed)
109006: Authentication failed for user '' from 10.31.1.5/11076
  to 192.1.1.1/23
```

[Debug PIX - Incapable de cingler le serveur ou de les introduire/non-concordance de client - RAYON](#)

Cet exemple chausse un PIX mettent au point où le serveur n'est pas pingable ou il y a une clé/non-concordance de client. L'utilisateur voit le nom d'utilisateur, le mot de passe, et délai d'attente des messages le « au serveur de RAYON » et à la « erreur : Nombre maximum d'essais dépassés » (un serveur faux a été permuté dedans la configuration).

```
109001: Auth start for user '???' from 10.31.1.5/11077
  to 192.1.1.1/23
109002: Auth from 10.31.1.5/11077 to 192.1.1.1/23 failed
 (server 100.100.100.100 failed)
109002: Auth from 10.31.1.5/11077 to 192.1.1.1/23 failed
 (server 100.100.100.100 failed)
109002: Auth from 10.31.1.5/11077 to 192.1.1.1/23 failed
 (server 100.100.100.100 failed)
109006: Authentication failed for user '' from 10.31.1.5/11077
  to 192.1.1.1/23
```

[Ajoutez l'autorisation](#)

Si vous décidez d'ajouter l'autorisation, vous aurez besoin de l'autorisation pour le même intervalle source et de destination (puisque l'autorisation est non valide sans authentification) :

```
aaa authorization telnet inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound aaa authorization
HTTP inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound aaa authorization ftp inbound 0.0.0.0
0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound
```

Notez que l'autorisation n'est pas ajoutée pour « sortant » parce que le trafic sortant est authentifié avec le RAYON, et l'autorisation RADIUS est non valide.

[Exemples de debug d'authentification et d'autorisation de PIX](#)

[Debug PIX - Bonne authentification et autorisation réussie - TACACS+](#)

Cet exemple affiche qu'un PIX met au point avec la bonne authentification et l'autorisation réussie :

```
109011: Authen Session Start: user 'pixuser', Sid 8
109007: Authorization permitted for user 'pixuser'
      from 192.1.1.1/13160 to 10.31.1.5/23
109012: Authen Session End: user 'pixuser', Sid 8,
      elapsed 1 seconds
302001: Built inbound TCP connection 8 for faddr 192.1.1.1/13160
      gaddr 192.1.1.30/23 laddr 10.31.1.5/23 (pixuser)
```

[Debug PIX - Bonne authentification, autorisation défailante - TACACS+](#)

Cet exemple affiche qu'un PIX met au point avec la bonne authentification mais avec l'autorisation défailante. Ici l'utilisateur `erreur` voit également message « : Autorisation refusée. »

```
109001: Auth start for user '???' from 192.1.1.1/13162
      to 10.31.1.5/23
109011: Authen Session Start: user 'userhttp', Sid 10
109005: Authentication succeeded for user 'userhttp'
      from 10.31.1.5/23 to 192.1.1.1/13162
109008: Authorization denied for user 'userhttp'
      from 10.31.1.5/23 to 192.1.1.1/13162
109012: Authen Session End: user 'userhttp', Sid 10,
      elapsed 1 seconds
302010: 0 in use, 2 most used
```

[Ajoutez la gestion des comptes](#)

[TACACS+](#)

```
aaa accounting any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound
```

Débuggez l'aspect les mêmes si la comptabilité est "Marche/Arrêt". Cependant, au moment du « a construit, » enregistrement des comptes de « début » a est envoyé. Au moment de la « désinstallation, » l'enregistrement des comptes de « arrêt » a est envoyé.

Les enregistrements des comptes TACACS+ ressemblent à cette sortie (ceux-ci sont de NT Cisco Secure, par conséquent du format délimité par une virgule) :

```
04/26/2000,01:31:22,pixuser,Default Group,192.1.1.1,
start,,,,,,0x2a,,PIX,10.31.1.200,telnet,6,
```

```
Login,1,,,1,,,,,,,,,,,,,local_ip=10.31.1.5 foreign_ip=192.1.1.1,  
,,,,,,,,,,,,,,,zekie,,,,,,,,^  
04/26/2000,01:31:26,pixuser,Default Group,192.1.1.1,stop,4,  
,36,82,,,0x2a,,,PIX,10.31.1.200,telnet,6,  
Login,1,,,1,,,,,,,,,,,,,local_ip=10.31.1.5 foreign_ip=192.1.1. 1,  
,,,,,,,,,,,,,zekie,,,,,,,,
```

RAYON

```
aaa accounting any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
```

Débuggez les aspects les mêmes si la comptabilité est "Marche/Arrêt". Cependant, au moment du « a construit, » enregistrement des comptes de « début » a est envoyé. Au moment de la « désinstallation, » l'enregistrement des comptes de « arrêt » a est envoyé.

Les enregistrements des comptes de RAYON ressemblent à cette sortie (ceux-ci sont de Cisco Secure UNIX ; ceux dans le NT Cisco Secure peuvent être délimités par une virgule à la place) :

```
radrecv: Request from host a1f01c8 code=4, id=18, length=65  
Acct-Status-Type = Start  
Client-Id = 10.31.1.200  
Login-Host = 10.31.1.5  
Login-TCP-Port = 23  
Acct-Session-Id = "0x0000002f"  
User-Name = "pixuser"  
Sending Accounting Ack of id 18 to a1f01c8 (10.31.1.200)  
radrecv: Request from host a1f01c8 code=4, id=19, length=83  
Acct-Status-Type = Stop  
Client-Id = 10.31.1.200  
Login-Host = 10.31.1.5  
Login-TCP-Port = 23  
Acct-Session-Id = "0x0000002f"  
Username = "pixuser"  
Acct-Session-Time = 7
```

Utilisation de excepté commande

Dans notre réseau, si nous décidons qu'une source et/ou une destination particulières n'a pas besoin d'authentification, d'autorisation, ou de comptabilité, nous pouvons faire n'importe quoi de pareil sorti :

```
aaa authentication except inbound 192.1.1.1 255.255.255.255 0.0.0.0 0.0.0.0 AuthInbound
```

Si vous êtes « sauf » une case de l'authentification et avez l'autorisation en fonction, vous devez également excepter la case de l'autorisation.

Maximum-sessions et utilisateurs connectés de visionnement

Quelques serveurs TACACS+ et RADIUS ont des caractéristiques de « maximum-session » ou de « affichage des utilisateurs connectés ». La capacité de faire des maximum-sessions ou des utilisateurs connectés de contrôle dépend des enregistrements des comptes. Quand il y a un enregistrement de « début » de comptabilité généré mais aucun enregistrement de « arrêt », le serveur TACACS+ ou de RAYON suppose que la personne est encore ouverte une session (a une session par le PIX).

Ceci fonctionne bien pour le telnet et les connexions FTP en raison de la nature des connexions.

Ceci ne fonctionne pas bien pour le HTTP en raison de la nature de la connexion. Dans cet exemple de sortie, une configuration réseau différente est utilisée, mais les concepts sont identiques.

Les telnets d'utilisateur par le PIX, authentifiant sur le chemin :

```
(pix) 109001: Auth start for user '???' from 171.68.118.100/1200
to 9.9.9.25 /23
(pix) 109011: Authen Session Start: user 'cse', Sid 3
(pix) 109005: Authentication succeeded for user 'cse'
from 171.68.118.100/12 00 to 9.9.9.25/23
(pix) 302001: Built TCP connection 5 for faddr 9.9.9.25/23
gaddr 9.9.9.10/12 00 laddr 171.68.118.100/1200 (cse)
(server start account) Sun Nov 8 16:31:10 1998
rtp-pinecone.rtp.cisco.com cse
PIX 171.68.118.100 start task_id=0x3 foreign_ip=9.9.9.25
local_ip=171.68.118.100 cmd=telnet
```

Puisque le serveur n'a vu un enregistrement de « début » mais aucun enregistrement de « arrêt » (en ce moment), le serveur prouve que l'utilisateur de « telnet » est ouvert une session. Si l'utilisateur tente une autre connexion qui exige l'authentification (peut-être d'un autre PC) et si des maximum-sessions est placées à "1" sur le serveur pour cet utilisateur (assumant le serveur prend en charge des maximum-sessions), la connexion est refusée par le serveur.

L'utilisateur va en fonction de pair avec le telnet ou les activités en FTP sur l'hôte de cible, puis les sorties (passe 10 minutes là) :

```
(pix) 302002: Teardown TCP connection 5 faddr
9.9.9.25/80 gaddr 9.9.9.10/128 1
laddr 171.68.118.100/1281 duration 0:00:00 bytes 1907 (cse)
(server stop account) Sun Nov 8 16:41:17 1998
rtp-pinecone.rtp.cisco.com cse
PIX 171.68.118.100 stop task_id=0x3 foreign_ip=9.9.9.25
local_ip=171.68.118.100 cmd=telnet elapsed_time=5
bytes_in=98 bytes_out=36
```

Si l'uauth est 0 (authentifiez chaque fois) ou plus (authentifiez une fois et pas de nouveau au cours de la période uauth), un enregistrement des comptes est coupé pour chaque site accédé à.

Le HTTP fonctionne différemment en raison de la nature du protocole. Cette sortie affiche un exemple de HTTP :

L'utilisateur parcourt de 171.68.118.100 à 9.9.9.25 par le PIX :

```
(pix) 109001: Auth start for user '???' from 171.68.118.100/1281
to 9.9.9.25 /80
(pix) 109011: Authen Session Start: user 'cse', Sid 5
(pix) 109005: Authentication succeeded for user 'cse'
from 171.68.118.100/12 81 to 9.9.9.25/80
(pix) 302001: Built TCP connection 5 for faddr 9.9.9.25/80
gaddr 9.9.9.10/12 81 laddr 171.68.118.100/1281 (cse)
(server start account) Sun Nov 8 16:35:34 1998
rtp-pinecone.rtp.cisco.com cse
PIX 171.68.118.100 start task_id=0x9 foreign_ip=9.9.9.25
local_ip=171.68.118.100 cmd=http
(pix) 302002: Teardown TCP connection 5 faddr 9.9.9.25/80
gaddr 9.9.9.10/128 1 laddr 171.68.118.100/1281 duration
0:00:00 bytes 1907 (cse)
(server stop account) Sun Nov 8 16:35.35 1998
rtp-pinecone.rtp.cisco .com cse PIX 171.68.118.100
stop task_id=0x9 foreign_ip =9.9.9.25
```

```
local_ip=171.68.118.100 cmd=http elapsed_time=0
bytes_in=1907 bytes_out=223
```

L'utilisateur lit la page Web téléchargée.

L'enregistrement de début signalé à 16:35:34, et l'enregistrement d'arrêt signalé à 16:35:35. Ce téléchargement a pris une seconde (c'est-à-dire, il y avait moins d'une seconde entre le début et l'enregistrement d'arrêt). L'utilisateur est-il encore ouvert une session au site Web et à la connexion encore ouverts quand ils lisent la page Web ? Non. Les maximum-sessions ou l'affichage des utilisateurs connectés fonctionneront-ils ici ? Non, parce que le temps de connexion (le temps entre « construit » et la « désinstallation ») dans le HTTP est trop court.

L'enregistrement de « début » et de « arrêt » est fraction de seconde. Il n'y aura pas un enregistrement de « début » sans enregistrement de « arrêt », puisque les enregistrements se produisent pratiquement au même instant. Il y aura toujours « début » et « arrêtez » l'enregistrement envoyé au serveur pour chaque transaction, si l'auth est placé pour 0 ou quelque chose plus grande. Cependant, les maximum-sessions et l'affichage des utilisateurs connectés ne fonctionnent pas en raison de la nature de la connexion HTTP.

Authentification et activation sur le PIX lui-même

La discussion précédente a décrit authentifier le trafic de telnet (et HTTP, FTP) *par le* PIX. Nous nous assurons le telnet aux travaux PIX *sans* authentification en fonction :

```
telnet 10.31.1.5 255.255.255.255 passwd ww
aaa authentication telnet console AuthInbound
```

Quand le telnet d'utilisateurs au PIX, ils sont incités pour le mot de passe de telnet (**ww**). Alors le PIX demande également le TACACS+ (dans ce cas, puisque la liste de serveur de « AuthInbound » est utilisée) ou nom d'utilisateur RADIUS et mot de passe. Si le serveur est en panne, vous pouvez entrer dans le PIX en écrivant le **pix** pour le nom d'utilisateur, et alors le mot de passe d'enable (**mot de passe d'enable quoi que**) pour accéder.

Avec cette commande :

```
aaa authentication enable console AuthInbound
```

l'utilisateur est incité pour un nom d'utilisateur et mot de passe, qui est envoyé serveur TACACS (dans ce cas, puisque la liste de serveur au de « AuthInbound » est utilisée, la demande va au serveur TACACS) ou de RAYON. Puisque le paquet d'authentification pour l'enable est identique que le paquet d'authentification pour la procédure de connexion, si l'utilisateur peut ouvrir une session au PIX avec TACACS ou RAYON, ils peuvent activer par TACACS ou RAYON avec le même nom d'utilisateur/mot de passe. Ce problème a été assigné l'ID de bogue Cisco [CSCdm47044](#) (clients [enregistrés](#) seulement).

Authentification sur la console série

La commande d'**AuthInbound de console série d'authentification d'AAA** exige de la vérification d'authentification afin d'accéder à la console série du PIX.

Quand l'utilisateur exécute des commandes de configuration de la console, des messages de Syslog sont coupés (assumant le PIX est configuré pour envoyer le Syslog au niveau de débogage à un hôte de Syslog). C'est un exemple de ce qui est affiché sur le serveur de Syslog :

```
logmsg: pri 245, flags 0, from [10.31.1.200.2.2], msg Nov 01 1999
03:21:14: %PIX-5-111008: User 'pixuser' executed the 'logging' command.
```

[Changez la demande que les utilisateurs voient](#)

Si vous avez la commande de l'authentique-**demande PIX_PIX_PIX**, les utilisateurs qui passent par le PIX voient cet ordre :

```
PIX_PIX_PIX [at which point one would enter the username]
Password:[at which point one would enter the password]
```

Sur l'arrivée à la zone de destination finale, le « nom d'utilisateur : » et « mot de passe : la » demande est affichée. Cette demande affecte seulement des utilisateurs allant *par le* PIX, pas au PIX.

Remarque: Il n'y a aucun enregistrement des comptes coupé pour l'accès au PIX.

[Personnalisez les utilisateurs de message voient sur le succès/panne](#)

Si vous avez les commandes :

```
auth-prompt accept "GOOD_AUTH" auth-prompt reject "BAD_AUTH"
```

les utilisateurs voient cet ordre sur procédure de connexion défectueuse/réussie par le PIX :

```
PIX_PIX_PIX
Username: asjdkl
Password:
"BAD_AUTH"
"PIX_PIX_PIX"
Username: cse
Password:
"GOOD_AUTH"
```

[Inactif et temporisations absolues de Par-utilisateur](#)

L'inactif et les limites absolues d'UAUTH peuvent être envoyés vers le bas du serveur TACACS+ sur une base par utilisateur. Si tous les utilisateurs dans votre réseau doivent avoir le même « uauth de délai d'attente, » n'implémentez pas ceci ! Mais si vous avez besoin du par-utilisateur différent d'uauths, continuez à lire.

Dans cet exemple, la commande de l'**uauth 3:00:00 de délai d'attente** est utilisée. Une fois qu'une personne authentifie, ils ne doivent pas authentifier à nouveau pendant trois heures. Cependant, si vous installez un utilisateur avec ce profil et avez l'*autorisation d'AAA* TACACS en fonction dans le PIX, l'inactif et les temporisations absolues dans le profil utilisateur ignorent l'uauth de délai d'attente dans le PIX pour cet utilisateur. Ceci ne signifie pas que la session de telnet par le PIX est déconnectée après l'inactif/temporisation absolue. Il contrôle juste si la ré-authentification a lieu.

Ce profil provient le logiciel gratuit TACACS+ :

```
user = timeout {
default service = permit
login = cleartext "timeout"
```

```
service = exec {  
  timeout = 2  
  idletime = 1  
}  
}
```

Après authentification, exécutez une commande d'**uauth d'exposition** sur le PIX :

```
pix-5# show uauth Current Most Seen Authenticated Users 1 1 Authen In Progress 0 1 user  
'timeout' at 10.31.1.5, authorized to: port 11.11.11.15/telnet absolute timeout: 0:02:00  
inactivity timeout: 0:01:00
```

Après que l'utilisateur repose l'inactif pour une minute, le débogage sur le PIX affiche :

```
109012: Authen Session End: user 'timeout', Sid 19, elapsed 91 seconds
```

L'utilisateur doit authentifier à nouveau quand il revient au même hôte de cible ou à un différent hôte.

HTTP virtuel

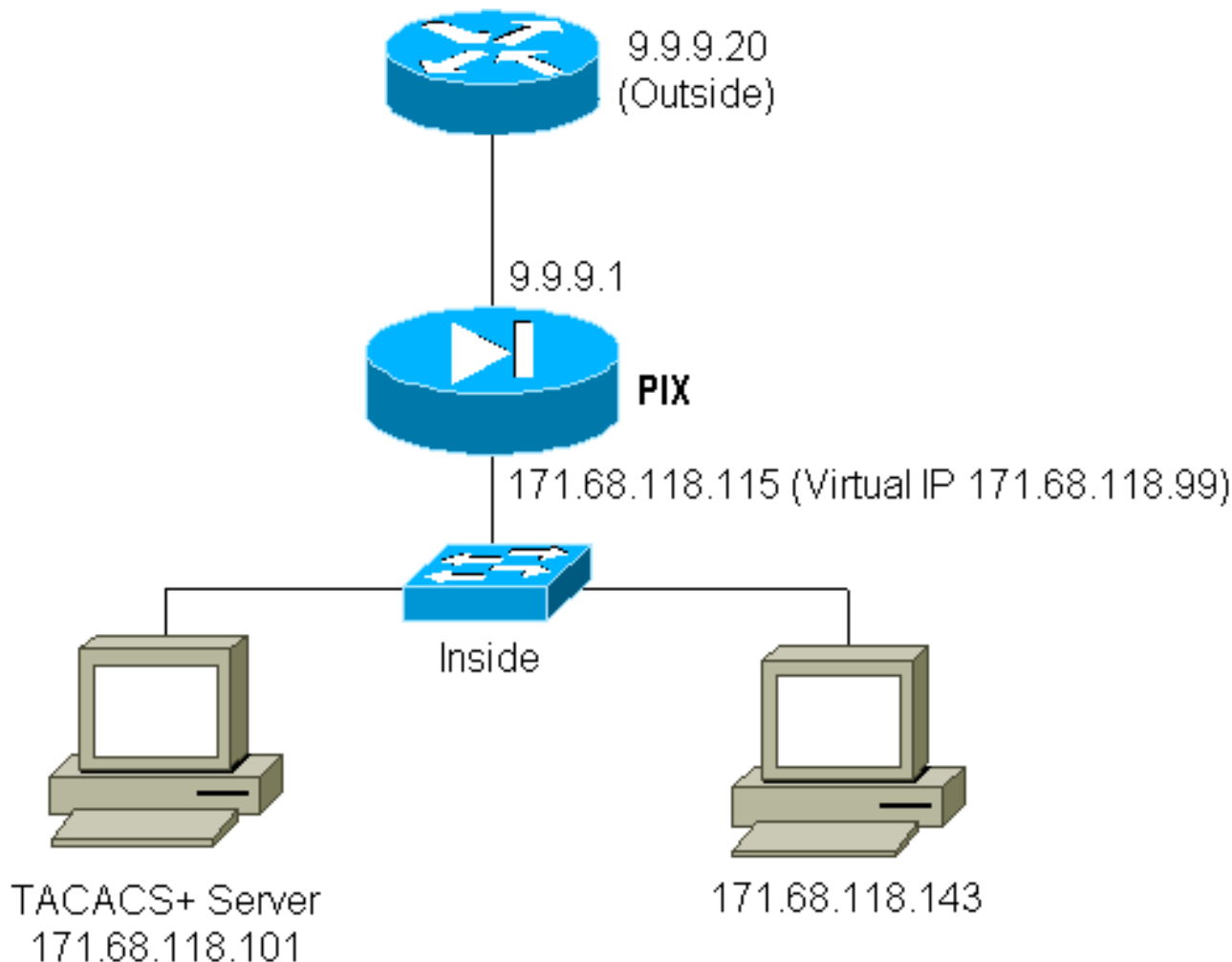
Si l'authentification est exigée sur des sites en dehors du PIX, aussi bien que sur le PIX lui-même, on peut parfois observer le comportement du navigateur peu commun puisque les navigateurs cachent le nom d'utilisateur et mot de passe.

Pour éviter ceci, vous pouvez implémenter le HTTP virtuel en ajoutant une adresse [RFC 1918](#) (une adresse qui est unroutable sur l'Internet, mais valide et seul pour le PIX à l'intérieur du réseau) à la configuration PIX utilisant cette commande :

```
virtual http #.#.#.# [warn]
```

Quand l'utilisateur essaye d'aller en dehors du PIX, l'authentification est exigée. Si le paramètre d'avertissement est présent, l'utilisateur reçoit un message de réorientation. L'authentification est bonne pour la durée dans l'uauth. Comme indiqué dans la documentation, ne placez pas la durée de commande d'**uauth de délai d'attente aux** secondes 0 avec le HTTP virtuel. Ceci empêche des connexions HTTP au vrai web server.

Diagramme sortant de HTTP virtuel



HTTP virtuel de configuration PIX sortant

```
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
global (outside) 1 9.9.9.5-9.9.9.9 netmask 255.0.0.0
timeout uauth 01:00:00
aaa-server TACACS+ protocol tacacs+
aaa-server AuthOutbound protocol tacacs+
aaa-server AuthOutbound (inside) host 171.68.118.101 cisco timeout 10
aaa authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
virtual http 171.68.118.99
auth-prompt THIS_IS_PIX_5
```

Telnet virtuel

Il est possible de configurer le PIX pour authentifier tout le trafic en entrée et en sortie, mais ce n'est pas une bonne idée de faire ainsi. C'est parce que quelques protocoles, tels que la « messagerie, » ne sont pas facilement authentifiés. Quand un essai de serveur de messagerie et de client à communiquer par le PIX quand tout le trafic par le PIX est authentifié, Syslog PIX pour les messages unauthentifiable d'exposition de protocoles comme :

```
109001: Auth start for user '???' from 9.9.9.10/11094
to 171.68.118.106/25
109009: Authorization denied from 171.68.118.106/49 to
9.9.9.10/11094 (not authenticated)
```

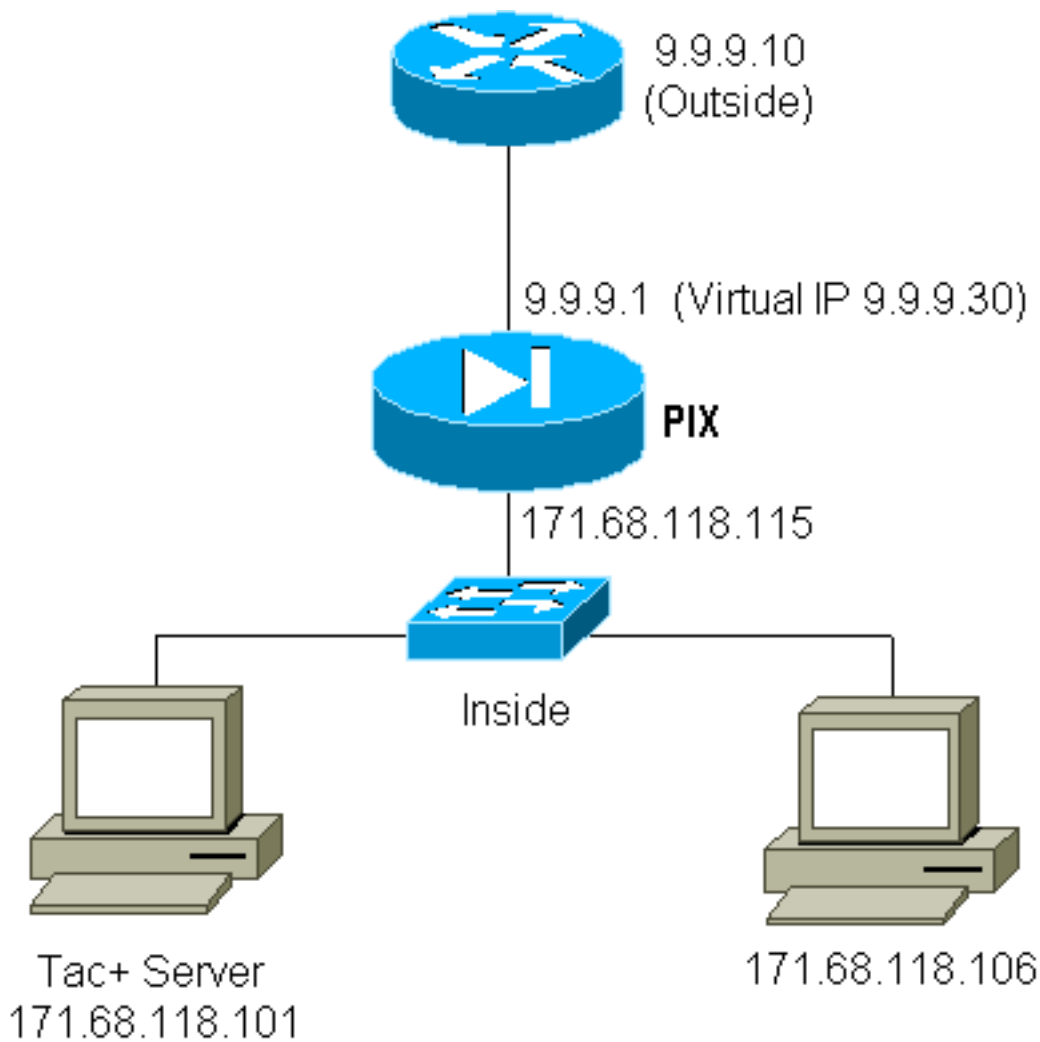
Puisque la messagerie et quelques autres services ne sont pas assez interactifs pour authentifier, une solution est d'utiliser **à moins que** commande pour l'authentification/autorisation (authentifiez

tous excepté la source/destination du serveur de messagerie/du client).

S'il y a un besoin réel d'authentifier un certain type service peu commun, ceci peut être fait au moyen de la **commande telnet virtuelle**. Cette commande permet à l'authentification pour se produire à l'IP virtuel de telnet. Après cette authentification, le trafic pour le service peu commun peut aller au vrai serveur.

Dans cet exemple, nous voulons que le trafic du port TCP 49 découle de l'hôte 9.9.9.10 d'extérieur à l'hôte interne 171.68.118.106. Puisque ce trafic n'est pas vraiment authentifiable, nous installons un telnet virtuel. Pour le telnet virtuel d'arrivée, il doit y avoir une charge statique associée. Ici, 9.9.9.20 et 171.68.118.20 sont des adresses virtuelles.

Diagramme d'arrivée de telnet virtuel



Configuration Virtual Telnet PIX d'arrivée

```
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
static (inside,outside) 9.9.9.20 171.68.118.20 netmask 255.255.255.255 0 0
static (inside,outside) 9.9.9.30 171.68.118.106 netmask 255.255.255.255 0 0
conduit permit tcp host 9.9.9.20 eq telnet any
conduit permit tcp host 9.9.9.30 eq tacacs any
aaa-server TACACS+ protocol tacacs+
aaa-server AuthInbound protocol tacacs+
aaa-server AuthInbound (inside) host 171.68.118.101 cisco timeout 5
```

```
AAA authentication any inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound
virtual telnet 9.9.9.20
```

Telnet virtuel de configuration utilisateur de serveur TACACS+ d'arrivée

```
user = pinecone {
default service = permit
    login = cleartext "pinecone"
service = exec {
    timeout = 10
    idletime = 10
}
}
```

Telnet virtuel de debug PIX d'arrivée

L'utilisateur chez 9.9.9.10 doit d'abord authentifier par Telnetting à l'adresse de 9.9.9.20 sur le PIX :

```
pixfirewall# 109001: Auth start for user '???' from 9.9.9.10/11099
to 171.68.118.20/23
109011: Authen Session Start: user 'pinecone', Sid 13
109005: Authentication succeeded for user 'pinecone'
from 171.68.118.20/23 to 9.9.9.10/1470
```

Après l'authentification réussie, la commande **d'uth d'exposition** prouve que l'utilisateur a le « temps sur le mètre » :

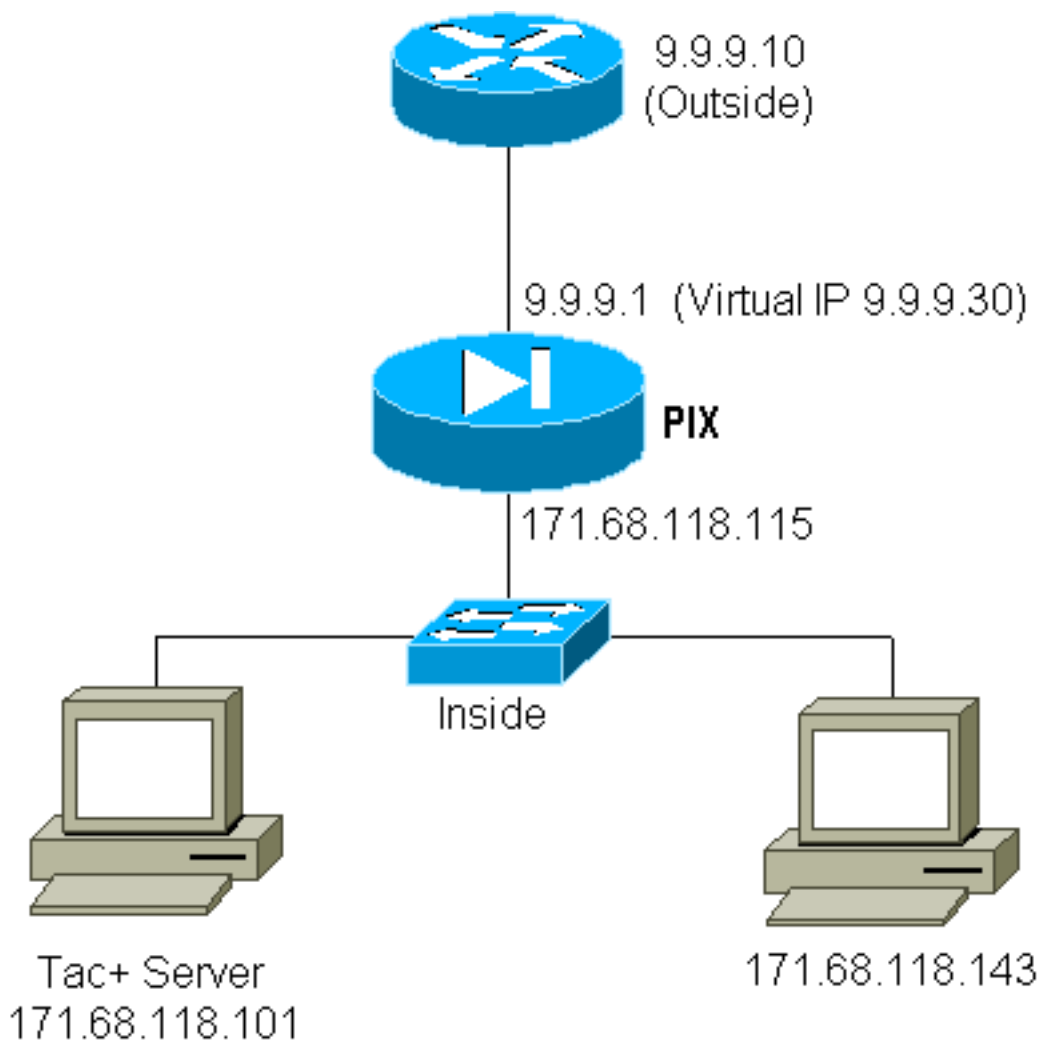
```
pixfirewall# show uauth Current Most Seen Authenticated Users 1 1 Authen In Progress 0 1 user
'pinecone' at 9.9.9.10, authenticated absolute timeout: 0:10:00 inactivity timeout: 0:10:00
```

Ici, le périphérique chez 9.9.9.10 veut envoyer le trafic TCP/49 au périphérique chez 171.68.118.106 :

```
pixfirewall# 109001: Auth start for user 'pinecone' from 9.9.9.10/11104
to 171.68.118.20/23
109011: Authen Session Start: user 'pinecone', Sid 14
109005: Authentication succeeded for user 'pinecone' from 171.68.118.20/23
to 9.9.9.10/1470
302001: Built TCP connection 23 for faddr 9.9.9.10/11104 gaddr 9.9.9.30/49
laddr 171.68.118.106/49 (pinecone)
302002: Teardown TCP connection 23 faddr 9.9.9.10/11104 gaddr 9.9.9.30/49
laddr 171.68.118.106/49 duration 0:00:10 bytes 179 (pinecone)
```

Telnet virtuel sortant

Puisqu'on permet le trafic sortant par défaut, pas statique est exigé pour l'usage du telnet virtuel sortant. Dans cet exemple, l'utilisateur intérieur aux telnets de 171.68.118.143 à 9.9.9.30 virtuel et authentifie. La connexion de telnet est immédiatement abandonnée. Une fois qu'authentifié, on permet le trafic TCP de 171.68.118.143 au serveur chez 9.9.9.10 :



Configuration Virtual Telnet PIX sortante

```
ip address outside 9.9.9.1 255.255.255.0
ip address inside 171.68.118.115 255.255.255.0
global (outside) 1 9.9.9.5-9.9.9.9 netmask 255.0.0.0
timeout uauth 00:05:00
aaa-server TACACS+ protocol tacacs+
aaa-server AuthOutbound protocol tacacs+
aaa-server AuthOutbound (inside) host 171.68.118.101 cisco timeout 10
AAA authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
virtual telnet 9.9.9.30
```

Telnet virtuel de debug PIX sortant

```
109001: Auth start for user '???' from 171.68.118.143/1536
to 9.9.9.30/23
109011: Authen Session Start: user 'timeout_143', Sid 25
109005: Authentication succeeded for user 'timeout_143' from
171.68.118.143/1536 to 9.9.9.30/23
302001: Built TCP connection 46 for faddr 9.9.9.10/80 gaddr
9.9.9.30/1537 laddr 171.68.118.143/1537 (timeout_143)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302001: Built TCP connection 47 for faddr 9.9.9.10/80 gaddr
9.9.9.30/1538 laddr 171.68.118.143/1538 (timeout_143)
302002: Teardown TCP connection 46 faddr 9.9.9.10/80 gaddr
9.9.9.30/1537 laddr 171.68.118.143/1537 duration 0:00:03
bytes 625 (timeout_143)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302002: Teardown TCP connection 47 faddr 9.9.9.10/80 gaddr
```

```
9.9.9.30/1538 laddr 171.68. 118.143/1538 duration 0:00:01
bytes 2281 (timeout_143)
302009: 0 in use, 1 most used
```

Déconnexion virtuelle de Telnet

Quand les telnets d'utilisateur à l'IP virtuel de telnet, la commande d'**uauth d'exposition** affiche l'uauth.

Si l'utilisateur veut empêcher le trafic d'aller après que la session soit de finition (quand il y a temps laissé dans l'uauth), le telnet des besoins de l'utilisateur à l'IP virtuel de telnet de nouveau. Ceci bascule la session hors fonction.

Autorisation sur le port

Vous pouvez avoir besoin de l'autorisation sur une plage de port. Dans cet exemple, l'authentification était encore exigée pour tout sortant, mais seulement l'autorisation a été exigée pour des ports TCP 23-49.

Configuration PIX

```
AAA authentication any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound AAA authorization
tcp/23-49 outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthOutbound
```

Quand le telnet a été fait de 171.68.118.143 à 9.9.9.10, l'authentification et l'autorisation se sont produites parce que le port 23 de telnet est dans la plage 23-49.

Quand une session de HTTP est faite de 171.68.118.143 à 9.9.9.10, vous devez encore authentifier, mais le PIX ne demande pas au serveur TACACS+ d'autoriser le HTTP parce que 80 n'est pas dans la plage 23-49.

Configuration du serveur de logiciel gratuit TACACS+

```
user = telnetrange {
    login = cleartext "telnetrange"
    cmd = tcp/23-49 {
        permit 9.9.9.10
    }
}
```

Notez que le PIX envoie « cmd=tcp/23-49" et « cmd-arg=9.9.9.10" au serveur TACACS+.

Debug sur le PIX

```
109001: Auth start for user '???' from 171.68.118.143/1051
to 9.9.9.10/23
109011: Authen Session Start: user 'telnetrange', Sid 0
109005: Authentication succeeded for user 'telnetrange'
from 171.68.118.143/1051 to 9.9.9.10/23
109011: Authen Session Start: user 'telnetrange', Sid 0
109007: Authorization permitted for user 'telnetrange'
from 171.68.118.143/1051 to 9.9.9.10/23
302001: Built TCP connection 0 for faddr 9.9.9.10/23
gaddr 9.9.9.5/1051 laddr 171.68.118.143/1051 (telnetrange)
109001: Auth start for user '???' from 171.68.118.143/1105
```

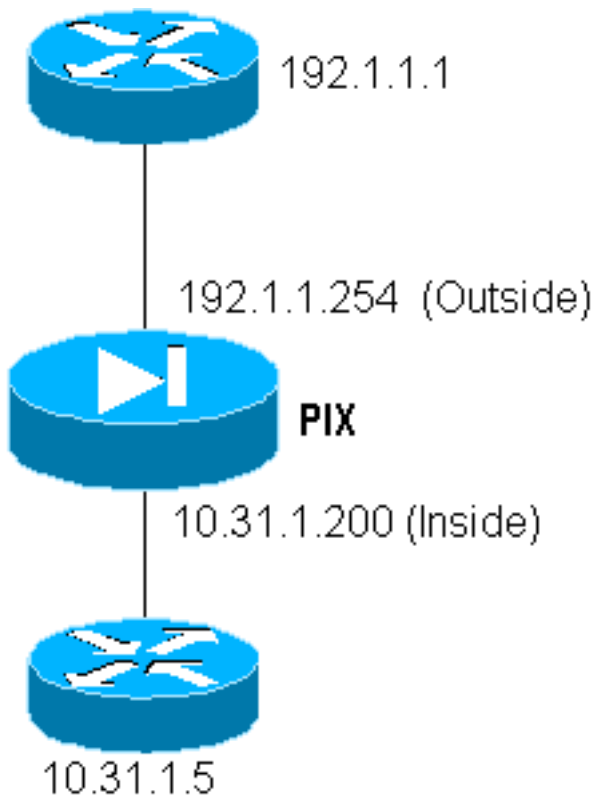
```

to 9.9.9.10/80
109001: Auth start for user '???' from 171.68.118.143/1110
to 9.9.9.10/80
109011: Authen Session Start: user 'telnetrange', Sid 1
109005: Authentication succeeded for user 'telnetrange'
from 171.68.118.143/1110 to 9.9.9.10/80
302001: Built TCP connection 1 for faddr 9.9.9.10/80 gaddr 9.9.9.5/1110
laddr 171.68.1 18.143/1110 (telnetrange)
302001: Built TCP connection 2 for faddr 9.9.9.10/80 gaddr 9.9.9.5/1111
laddr 171.68.1 18.143/1111 (telnetrange)
302002: Teardown TCP connection 1 faddr 9.9.9.10/80 gaddr 9.9.9.5/1110
laddr 171.68.11 8.143/1110 duration 0:00:08 bytes 338 (telnetrange)
304001: timeout_143@171.68.118.143 Accessed URL 9.9.9.10:/
302002: Teardown TCP connection 2 faddr 9.9.9.10/80 gaddr 9.9.9.5/1111
laddr 171.68.11 8.143/1111 duration 0:00:01 bytes 2329 (telnetrange)

```

AAA expliquant le trafic autre que le HTTP, le FTP, et le telnet

La version du logiciel PIX 5.0 change la fonctionnalité de comptabilité du trafic. Des enregistrements des comptes peuvent maintenant être coupés pour le trafic autre que le HTTP, le FTP, et le telnet, une fois que l'authentification est terminée.



La TFTP-copie un fichier du routeur extérieur (192.1.1.1) au routeur interne (10.31.1.5), ajoutent le telnet virtuel pour ouvrir un trou pour le processus TFTP :

```

virtual telnet 192.1.1.30 static (inside,outside) 192.1.1.30 10.31.1.5 netmask 255.255.255.255 0
0 conduit permit udp any any AAA authentication udp/69 inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
AuthInbound AAA authorization udp/69 inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound AAA
accounting udp/0 inbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound

```

Ensuite, le telnet du routeur extérieur chez 192.1.1.1 à IP virtuel 192.1.1.30 et authentifie à l'adresse virtuelle qui permet à l'UDP pour traverser le PIX. Dans cet exemple, le processus d'instantané de copy tftp a été commencé de l'externe vers interne :

```
302006: Teardown UDP connection for faddr 192.1.1.1/7680
      gaddr 192.1.1.30/69 laddr 10.31.1.5/69
```

Pour chaque **éclair de copy tftp** sur le PIX (il y avait de trois pendant cette copie IOS), un enregistrement des comptes est coupé et envoyé au serveur d'authentification. Être suit un exemple d'un enregistrement TACACS sur Windows Cisco Secure) :

```
Date, Time, Username, Group-Name, Caller-Id, Acct-Flags, elapsed_time,
  service, bytes_in, bytes_out, paks_in, paks_out,
  task_id, addr, NAS-Portname, NAS-IP-Address, cmd
04/28/2000, 03:08:26, pixuser, Default Group, 192.1.1.1, start, , , , , , , ,
0x3c, , PIX, 10.31.1.200, udp/69
```

[Informations connexes](#)

- [Référence des commandes PIX](#)
- [Page de support produit PIX](#)
- [Demandes de commentaires \(RFC\)](#)
- [Support technique - Cisco Systems](#)