

Quel est un fichier Pac de proxy et y a-t-il des exemples ?

Contenu

[Question :](#)

Question :

Quel est un fichier Pac de proxy, et y a-t-il des exemples ?

Environnement : Appliance de sécurité Web de Cisco

Remarque: Cet article de la base de connaissances met en référence le logiciel qui n'est pas mis à jour ou est pris en charge par Cisco. Les informations sont données comme courtoisie pour votre commodité. Pour davantage d'assistance, contactez s'il vous plaît le fournisseur de logiciels.

Lien de référence : [Javascript ou fichiers d'exemple d'Automatique-proxy de JScript](#)

Javascript ou fichiers d'exemple d'Automatique-proxy de JScript

Les scripts suivants fournissent des exemples de la façon dont un fichier « .pac » pourrait être utilisé pour spécifier un URL d'automatique-proxy. Pour utiliser ces fonctions, vous devez changer les noms, les numéros de port, et les adresses IP de proxy.

1. Les fonctions d'**isInNet**, **isResolvable**, et de **dnsResolve** questionnent un serveur DNS
2. Les références aux objets, aux propriétés, ou aux méthodes de modèle objet font échouer le fichier de configuration automatique de proxy silencieusement. Par exemple, les références **window.open (...)**, l'alerte (...), et le mot de passe (...) tout font échouer le fichier de configuration automatique de proxy sur l'Internet Explorer.

Exemple 1 : Les hôtes locaux connectent direct, tous les autres se connectent par l'intermédiaire du proxy

Les contrôles de fonctionnement suivants pour voir si l'adresse Internet est un hôte local, et si elle est, si la connexion est directe. Si l'adresse Internet n'est pas un hôte local, la connexion est par le proxy (proxy).

fonction FindProxyForURL (URL, hôte)

```
{
si (isPlainHostName(host))
retournez « DIRIGENT » ;
autrement
renvoyez le « PROXY proxy:80" ;
}
```

Les contrôles de fonctionnement d'**isPlainHostName** pour voir s'il y a des points dans l'adresse Internet. Si oui, il renvoie faux ; autrement, la fonction renvoie vrai.

Note: Pour une liaison directe au web server local, un domaine ou une correspondance de sous-réseau pourrait également être nécessaire. Non toutes les demandes aux web server locaux sont faites utilisant juste des adresses Internet

Exemple 2 : Les hôtes à l'intérieur du Pare-feu connectent les serveurs directs et externes locaux se connectent par l'intermédiaire du proxy

Les contrôles de fonctionnement suivants pour voir si l'hôte est une adresse Internet « ordinaire » (signifiant le nom de domaine n'est pas inclus) ou une partie d'un domaine particulier (.company.com) mais l'adresse Internet n'est pas WWW ou maison.

fonction FindProxyForURL (URL, hôte)

```
{
si ((isPlainHostName(host) ||
dnsDomainIs (hôte, « .company.com »)) &&
! && de localHostOrDomainIs (hôte, « www.company.com ")
! localHostOrDoaminIs (hôte, « home.company.com »))
retournez « DIRIGENT » ;
autrement
renvoyez le « PROXY proxy:80" ;
}
```

Note:

- La fonction de **localHostOrDomainIs** est exécutée seulement pour l'URLs dans le domaine local.
- **Les dnsDomainIs** fonctionnent des retours vrais si le domaine de l'adresse Internet apparie le domaine donné.

Exemple 3 : Si l'hôte est résoluble, connectez direct. Autrement connectez utilisant un proxy.

La fonction suivante demande au serveur DNS d'essayer de résoudre l'adresse Internet passée à elle. S'il peut, alors une liaison directe est faite. S'il ne peut pas, le rapport est établi par l'intermédiaire du proxy. C'est utile quand un serveur DNS interne est utilisé pour résoudre toutes les adresses Internet internes.

fonction FindProxyForURL (URL, hôte)

```
{
si (isResolvable(host))
retournez « DIRIGENT » ;
autrement
renvoyez le « PROXY proxy:80" ;
}
```

Voir la note sur la fonction isResolvable en haut de la page.

Exemple 4 : Si l'hôte est dans le sous-réseau spécifié, connectez direct. Autrement connectez utilisant un proxy.

La fonction suivante compare un modèle et un masque donnés d'adresse IP à l'adresse Internet. C'est utile si certains hôtes dans un sous-réseau sont connectés directement et d'autres sont connectés utilisant un proxy.

```
fonction FindProxyForURL (URL, hôte)
{
si (isInNet (hôte, "999.99.9.9", "255.0.255.0"))
retournez « DIRIGENT » ;
autrement
renvoyez le « PROXY proxy:80" ;
}
```

Voir la note sur la fonction d'**isInNet** en haut de la page.

La fonction d'**isInNet** (hôte, modèle, masque) renvoie vrai si l'adresse IP d'hôte apparie le modèle spécifié. Le masque indique quelle partie de l'adresse IP à apparier (255=match, 0=ignore).

Exemple 5 : Déterminez le type de connexion basé sur le domaine d'hôte

La fonction suivante spécifie une liaison directe si l'hôte est local. Si l'hôte n'est pas local, cette fonction détermine quel proxy utiliser a basé sur le domaine d'hôte. C'est utile si le nom de domaine d'hôte est l'un des critères pour la sélection de proxy.

```
fonction FindProxyForURL (URL, hôte)
{
si (isPlainHostName(host))
retournez « DIRIGENT » ;
autrement si (shExpMatch (hôte, « *.com »))
renvoyez le « PROXY comproxy:80" ;
autrement si (shExpMatch (hôte, « *.edu »))
renvoyez le « PROXY eduproxy:80" ;
autrement
renvoyez le « proxy de PROXY » ;
}
```

La fonction de **shExpMatch** (streptocoque, shexp) renvoie vrai si le streptocoque apparie le shexp utilisant des modèles d'expression de shell.

Exemple 6 : Déterminez le type de connexion basé sur le protocole étant utilisé

La fonction suivante extrait le protocole étant utilisé et fait une sélection de proxy en conséquence. Si aucune correspondance n'est faite sur le protocole, alors une liaison directe est faite. C'est utile si le protocole étant utilisé est l'un des critères pour la sélection de proxy.

```
fonction FindProxyForURL (URL, hôte)
{
si (url.substring(0, 5) HTTP de == « : ") {
renvoyez le « PROXY proxy:80" ;
}
autrement si (url.substring(0, 4) FTP de == « : ") {
renvoyez le « PROXY fproxy:80" ;
}
autrement si (url.substring(0, 7) Gopher de == « : ") {
renvoyez le « gproxy de PROXY » ;
}
```

```

}
autrement si (url.substring(0, 6) https de == « : ") {
renvoyez le « PROXY secproxy:8080" ;
}
autrement {
retournez « DIRIGENT » ;
}
}
}

```

La fonction de sous-chaîne extrait le nombre spécifié de caractères d'une chaîne.

Exemple 7 : Déterminez le paramètre de proxy en vérifiant pour voir si l'adresse Internet apparie l'adresse IP

La fonction suivante transforme une sélection de proxy en traduisant l'adresse Internet en adresse IP et en la comparant à une chaîne spécifiée.

```

fonction FindProxyForURL (URL, hôte)
{
si == "999.99.99.999" {/= http://proxy (de dnsResolve(host)
renvoyez le « PROXY secproxy:8080" ;
}
autrement {
renvoyez le « PROXY proxy:80" ;
}
}
}

```

Voir la note sur la fonction de **dnsResolve** en haut de la page.

Exemple 8 : Si l'IP d'hôte apparie l'IP spécifié, connectez par l'intermédiaire du proxy, connectent autrement direct

La fonction suivante est une autre manière de faire une sélection de proxy basée sur spécifier une adresse IP. Cet exemple, à la différence de l'exemple 7, emploie l'appel de fonction pour obtenir explicitement l'adresse IP numérique (l'exemple 7 emploie la fonction de **dnsResolve** pour traduire l'adresse Internet en adresse IP numérique).

```

fonction FindProxyForURL (URL, hôte)
{
si (== "999.99.999.99" de myIpAddress()) {
renvoyez le « PROXY proxy:80" ;
}
autrement {
retournez « DIRIGENT » ;
}
}
}

```

Les **myIpAddress** fonctionnent des retours l'adresse IP (dans le format d'entier-point) de l'hôte que le navigateur s'exécute en fonction.

Exemple 9 : S'il y a des points dans l'adresse Internet, connectez utilisant un proxy. Autrement, connectez direct.

Les contrôles de fonctionnement suivants pour voir combien de points sont dans l'adresse Internet. S'il y a des points dans l'adresse Internet, établissez un rapport par l'intermédiaire du proxy. S'il n'y a aucun point dans l'adresse Internet, faites une liaison directe. C'est une autre manière de déterminer des types de connexion basés sur des caractéristiques d'adresse Internet.

fonction FindProxyForURL (URL, hôte)

```
{
si (dnsDomainLevels(host) > 0) { //si le nombre de points dans l'hôte > 0
renvoyez le « PROXY proxy:80" ;
}
retournez « DIRIGENT » ;
}
```

Les **dnsDomainLevels** fonctionnent des retours par entier égal au nombre de points dans l'adresse Internet.

Exemple 10 : Spécifiez les jours de la semaine pour se connecter par l'intermédiaire du proxy, d'autres jours connectent direct

La fonction suivante détermine le type de connexion en spécifiant les jours de la semaine qui sont appropriés pour un proxy. Les jours qui ne tombent pas entre ces paramètres utilisent une liaison directe. Cette fonction pourrait être utile dans les situations où vous pourriez vouloir utiliser un proxy quand le trafic est lourd et permettre une liaison directe quand le trafic est lumière.

fonction FindProxyForURL (URL, hôte)

```
{
si (weekdayRange (« L'ÉPOUSEZ », « SAT », « GMT »))
renvoyez le « PROXY proxy:80" ;
autrement
retournez « DIRIGENT » ;
}
```

La fonction **weekdayRange(day1 [,day2] [, GMT])** retourne si l'heure système en cours fait partie de la marge spécifiée par les paramètres day1, day2, et GMT. Seulement le premier paramètre est exigé. Le paramètre GMT présume que les valeurs temporelles sont à l'heure l'heure de Greenwich plutôt que le fuseau horaire heure locale.

Exemple 11 : Équilibrage de charge avec une fonction simple de calcul

fonction FindProxyForURL (URL, hôte) {

renvoyez le randomProxy() ;

}

randomProxy() de fonction

{

commutez (Math.floor (Math.random () *2))

{

affaire 0 : renvoyez le « PROXY 1.1.1.1:3128 ; PROXY 2.2.2.2 ; PROXY DIRECT ; "

affaire 1 : renvoyez le « PROXY 2.2.2.2:3128 ; PROXY 1.1.1.1 ; PROXY DIRECT ; "

}

}

Vous verrez que les cas implémentent également un Basculement au cas où un proxy ne serait pas disponible.