

Was ist eine Proxy Pac-Datei und gibt es Beispiele?

Inhalt

[Frage:](#)

Frage:

Was ist eine Proxy Pac-Datei, und gibt es Beispiele?

Umgebung: Cisco Web Security Appliance

Hinweis: Dieser Knowledge Base-Artikel bezieht sich auf Software, die nicht von Cisco verwaltet oder unterstützt wird. Die Informationen werden Ihnen zu Ihrer Zufriedenheit zur Verfügung gestellt. Wenden Sie sich für weitere Unterstützung an den Softwareanbieter.

Referenzlink: [JavaScript- oder JScript-Beispieldateien für Auto-Proxy](#)

JavaScript- oder JScript-Beispieldateien für Auto-Proxy

Die folgenden Skripte enthalten Beispiele dafür, wie eine **.pac**-Datei verwendet werden kann, um eine automatische Proxy-URL anzugeben. Um diese Funktionen zu verwenden, müssen Sie die Proxynamen, Portnummern und IP-Adressen ändern.

1. Das **istInNet**, **istAuflösbar**, und **dnsResolve**-Funktionen fragen einen DNS-Server ab
2. Verweise auf Objekte, Eigenschaften oder Methoden des Objektmodells führen dazu, dass die automatische Konfigurationsdatei des Proxys im Hintergrund fehlschlägt. Beispielsweise führen die Verweise **window.open(..)**, **alert(...)** und **password(..)** dazu, dass die automatische Konfigurationsdatei des Proxys im Internet Explorer fehlschlägt.

Beispiel 1: Die lokalen Hosts stellen eine direkte Verbindung her, alle anderen stellen eine Verbindung über den Proxy her.

Die folgende Funktion überprüft, ob der Hostname ein lokaler Host ist und ob die Verbindung direkt ist. Wenn der Hostname kein lokaler Host ist, wird die Verbindung über den Proxy (Proxy) hergestellt.

Funktion FindProxyForURL(url, host)

```
{  
if (isPlainHostName(host))  
"DIRECT" zurückgeben;  
andere  
return "PROXY proxy:80";  
}
```

Die `isPlainHostName`-Funktion überprüft, ob der Hostname Punkte enthält. Wenn dies der Fall ist, wird `false` zurückgegeben. Andernfalls gibt die Funktion `true` zurück.

Hinweis: Für eine direkte Verbindung zum lokalen Webserver kann auch eine Domain- oder Subnet-Übereinstimmung erforderlich sein. Nicht alle Anfragen an lokale Webserver erfolgen über Hostnamen

Beispiel 2: Die Hosts innerhalb der Firewall stellen eine direkte Verbindung her, die externen lokalen Server verbinden sich über den Proxy.

Die folgende Funktion überprüft, ob der Host entweder ein "einfacher" Hostname (d. h. der Domänenname ist nicht enthalten) oder ein Teil einer bestimmten Domäne (`.company.com`) ist, der Hostname jedoch weder `www` noch `home` ist.

Funktion `FindProxyForURL(url, host)`

```
{
if (isPlainHostName(host) |
dnsDomainIs(host, ".company.com") &&
!localHostOrDomainIs(host, "www.company.com") &&
!localHostOrDomainIs(host, "home.company.com"))
"DIRECT" zurückgeben;
andere
return "PROXY proxy:80";
}
```

Hinweis:

- Die `localHostOrDomainIs`-Funktion wird nur für URLs in der lokalen Domäne ausgeführt.
- Die `dnsDomainIs`-Funktion gibt `true` zurück, wenn die Domäne des Hostnamens mit der angegebenen Domäne übereinstimmt.

Beispiel 3: Wenn der Host auflösbar ist, schließen Sie direkt an. Andernfalls stellen Sie eine Verbindung über einen Proxy her.

Die folgende Funktion fordert den DNS-Server auf, zu versuchen, den an ihn übergebenen Hostnamen aufzulösen. Ist dies möglich, wird eine direkte Verbindung hergestellt. Ist dies nicht der Fall, wird die Verbindung über den Proxy hergestellt. Dies ist nützlich, wenn ein interner DNS-Server zum Auflösen aller internen Hostnamen verwendet wird.

Funktion `FindProxyForURL(url, host)`

```
{
if (isResolvable(host))
"DIRECT" zurückgeben;
andere
return "PROXY proxy:80";
}
```

Siehe Hinweis zur Funktion `isResolvable` (Auflösbar) oben auf der Seite.

Beispiel 4: Wenn sich der Host im angegebenen Subnetz befindet, stellen Sie eine direkte Verbindung her. Andernfalls stellen Sie eine Verbindung über einen Proxy her.

Die folgende Funktion vergleicht ein angegebenes IP-Adressmuster und eine IP-Maske mit dem Hostnamen. Dies ist nützlich, wenn bestimmte Hosts in einem Subnetz direkt verbunden werden sollen und andere über einen Proxy verbunden werden sollen.

```
Funktion FindProxyForURL(url, host)
{
  if (isInNet(host, "999.99.9.9", "255.0.255.0")
  "DIRECT" zurückgeben;
  andere
  return "PROXY proxy:80";
}
```

Siehe Hinweis auf der Funktion **isInNet** oben auf der Seite.

Die Funktion **isInNet** (Host, Muster, Maske) gibt true zurück, wenn die Host-IP-Adresse dem angegebenen Muster entspricht. Die Maske gibt an, welcher Teil der IP-Adresse übereinstimmt (255=Match, 0=Ignore).

Beispiel 5: Bestimmen des Verbindungstyps anhand der Hostdomäne

Die folgende Funktion gibt eine direkte Verbindung an, wenn der Host lokal ist. Wenn der Host nicht lokal ist, bestimmt diese Funktion anhand der Hostdomäne, welcher Proxy verwendet werden soll. Dies ist nützlich, wenn der Hostdomänenname eines der Kriterien für die Proxy-Auswahl ist.

```
Funktion FindProxyForURL(url, host)
{
  if (isPlainHostName(host))
  "DIRECT" zurückgeben;
  else if (shExpMatch(host, "*.com")
  return "PROXY comproxy:80";
  else if (shExpMatch(host, "*.edu")
  return "PROXY eduproxy:80";
  andere
  return "PROXY proxy";
}
```

Die Funktion **shExpMatch**(str, shexp) gibt true zurück, wenn str mit Shell-Ausdrucksmustern mit dem shexp übereinstimmt.

Beispiel 6: Bestimmen des Verbindungstyps anhand des verwendeten Protokolls

Die folgende Funktion extrahiert das verwendete Protokoll und macht eine Proxy-Auswahl entsprechend. Wenn im Protokoll keine Übereinstimmung gefunden wird, wird eine direkte Verbindung hergestellt. Dies ist nützlich, wenn das verwendete Protokoll eines der Kriterien für die Proxy-Auswahl ist.

```
Funktion FindProxyForURL(url, host)
{
  if (url.substring(0, 5) == "http:") {
  return "PROXY proxy:80";
  }
  else if (url.substring(0, 4) == "ftp:") {
  return "PROXY fproxy:80";
  }
```

```

}
else if (url.substring(0, 7) == "gopher:") {
return "PROXY gproxy";
}
else if (url.substring(0, 6) == "https:") {
return "PROXY secproxy:8080";
}
else {
"DIRECT" zurückgeben;
}
}
}

```

Die Unterzeichenfolgenfunktion extrahiert die angegebene Anzahl von Zeichen aus einer Zeichenfolge.

Beispiel 7: Bestimmen Sie die Proxy-Einstellung, indem Sie prüfen, ob der Hostname mit der IP-Adresse übereinstimmt.

Die folgende Funktion führt eine Proxyauswahl durch, indem der Hostname in eine IP-Adresse übersetzt und mit einer angegebenen Zeichenfolge verglichen wird.

```

Funktion FindProxyForURL(url, host)
{
if (dnsResolve(host) == "999.99.99.999") { // = http://proxy
return "PROXY secproxy:8080";
}
else {
return "PROXY proxy:80";
}
}
}

```

Siehe Hinweis auf die Funktion **dnsResolve** oben auf der Seite.

Beispiel 9: Wenn die Host-IP mit der angegebenen IP übereinstimmt, verbinden Sie sie über den Proxy, ansonsten verbinden Sie direkt

Die folgende Funktion ist eine andere Möglichkeit, eine Proxyauswahl basierend auf der Angabe einer IP-Adresse vorzunehmen. Im Gegensatz zu Beispiel 7 verwendet dieses Beispiel den Funktionsaufruf, um explizit die numerische IP-Adresse abzurufen (Beispiel 7 verwendet die Funktion **dnsResolve**, um den Hostnamen in die numerische IP-Adresse zu übersetzen).

```

Funktion FindProxyForURL(url, host)
{
if (myIPAddress() == "999.99.999.99") {
return "PROXY proxy:80";
}
else {
"DIRECT" zurückgeben;
}
}
}

```

Die **myIpAddress**-Funktion gibt die IP-Adresse (im Integer-Punktformat) des Hosts zurück, auf dem der Browser ausgeführt wird.

Beispiel 9: Wenn der Hostname Punkte enthält, stellen Sie eine Verbindung mit einem Proxy her. Andernfalls direkte Verbindung herstellen.

Die folgende Funktion überprüft, wie viele Punkte sich im Hostnamen befinden. Wenn der Hostname Punkte enthält, stellen Sie eine Verbindung über den Proxy her. Wenn der Hostname keine Punkte enthält, stellen Sie eine direkte Verbindung her. Dies ist eine andere Methode, um Verbindungstypen anhand der Eigenschaften des Hostnamens zu bestimmen.

```
Funktion FindProxyForURL(url, host)
{
if (dnsDomainLevels(host) > 0) { // if the number of points in host > 0
return "PROXY proxy:80";
}
"DIRECT" zurückgeben;
}
```

Die `dnsDomainLevels`-Funktion gibt eine ganze Zahl zurück, die der Anzahl der Punkte im Hostnamen entspricht.

Beispiel 10: Wochentage für die Verbindung über Proxy festlegen, andere Tage direkte Verbindung

Die folgende Funktion bestimmt den Verbindungstyp, indem sie Wochentage angibt, die für einen Proxy geeignet sind. Tage, die nicht zwischen diesen Parametern liegen, verwenden eine direkte Verbindung. Diese Funktion kann in Situationen nützlich sein, in denen Sie bei starkem Datenverkehr einen Proxy verwenden und bei leichtem Datenverkehr eine direkte Verbindung zulassen möchten.

```
Funktion FindProxyForURL(url, host)
{
if (week dayRange("WED", "SAT", "GMT"))
return "PROXY proxy:80";
andere
"DIRECT" zurückgeben;
}
```

Die Funktion `weekRange(day1 [,day2] [,GMT])` gibt zurück, ob die aktuelle Systemzeit in den durch die Parameter `day1`, `day2` und `GMT` festgelegten Bereich fällt. Nur der erste Parameter ist erforderlich. Der `GMT`-Parameter setzt voraus, dass die Zeitwerte in der Greenwich Mean Time und nicht in der lokalen Zeitzone liegen.

Beispiel 11: Lastenausgleich mit einer einfachen Mathematikfunktion

```
function FindProxyForURL(url, host){
return randomProxy();
}
function randomProxy()
{
switch( Math.floor( Math.random() *2 ))
{
Fall 0: return "PROXY 1.1.1.1:3128; PROXY 2.2.2.2; PROXY DIRECT;";
```

```
Fall 1: return "PROXY 2.2.2.2:3128; PROXY 1.1.1.1; PROXY DIRECT;"
```

```
}
```

```
}
```

Sie sehen, dass die Fälle auch ein Failover implementieren, falls kein Proxy verfügbar ist.