

**Inhalt:**

Proxy Auto  
Konfiguration PAC

- ▷ automatische Proxy Konfiguration in Internet Browsern
- ▷ Zentrale Änderung an Proxy Konfiguration
- ▷ Fallback auf direkten Internetzugang

**Aufgabe:**

Ein Netzwerk von unterschiedlichen Workstations (Linux / Windows) mit verschiedensten Browsern (Netscape / Internet Explorer / Firefox / Opera ) mit einer einheitlichen, zentral managebaren Internet Proxy Konfiguration zu versorgen.

**Umgebung:**

- Proxy (squid)
- Webserver (apache)
- DNS (bind)

**Ansätze:**

1. Nutzung der Netscape PAC (automatic proxy configuration)
2. Implementierung eines transparent Proxy für HTTP im Datenstrom

**Lösung 1: Nutzung der Netscape PAC**

Netscape hat mit der PAC einen Mechanismus geschaffen, der es erlaubt, zentral auf einem Webserver ein Konfigurationsskript für Browser zu hinterlegen das eine Entscheidung über die Nutzung von Internet Proxy Servern (z.B. squid) treffen kann.

Dokument von Netscape zu PAC:

<http://wp.netscape.com/eng/mozilla/2.0/relnotes/demo/proxy-live.html>

<http://developer.netscape.com/docs/manuals/proxy/adminux/autoconf.htm>

Die Dokumente beschreibt neben der Nutzung eines Proxy Mechanismus auch den Fallback auf einen direkten Internet Zugang.

In den Browsereinstellungen ist die automatische Proxy Übernahme zu aktivieren und als URL ein Pfad zu einem Webserver (z.B. Apache) anzugeben, der eine \*.pac Datei anbietet.

**Lösung 2: transparenter Proxy**

Durch die Verwendung eines LINUX Routers mit iptables, kann eine transparente Integration eines Proxy Servers für z.B. alle HTTP Zugriffe (TCP Port 80) realisiert werden. Am Client müsste keine Änderung passieren, allerdings muss der LINUX Router als das default Gateway für alle Clients eingetragen sein und evt. das vorhandene default Gateway ersetzen. Im Falle eines DSL Routers muss zusätzlich Sorge getragen werden, dass im Netzwerk keine ICMP Redirects erzeugt werden, wenn der DSL Router im gleichen Subnetz wie der Linux Router steht. In diesem Dokument wird auf diese Lösung verzichtet.

**Ziel: Zentrale Änderung der Proxy Konfiguration**

Beiden Lösungen ist gemein, dass nach der Integration nur noch an einer Stelle für alle Workstations die Proxy Konfiguration zu ändern ist und danach die Änderungen durch den Auto Config Mechanismus, bzw. Transparent auf alle Browser durchgereicht werden.





## Output

### Umsetzung

Es wird die Umsetzung der Lösung 1 beschrieben, da hierfür minimale Änderungen an einer bestehenden Umgebung notwendig sind, insbesondere ist der Zeitaufwand zur Implementierung minimal.

Eine proxy.pac, der Dateiname ist egal, wie Sie zum Einsatz kommen kann, regelt den lokalen Zugriff auf alle Domains des Kunden ohne Proxy, sowie alle Remote Zugriffe mit Proxy. Als Sicherheits-Fallback wurde der Direct Mode konfiguriert, falls der Proxy ausfallen sollte.

Die Endung pac ist wegen des MIME Types notwendig.

Damit ist die schnellstmögliche Lösung zur Integration eines Proxys umgesetzt.

```
# cat proxy.pac
```

```
function FindProxyForURL( url, host)
{
  if (isPlainHostName(host) ||
      dnsDomainIs(host, ".domain1.de") ||
      dnsDomainIs(host, ".domain2.de") ||
      isInNet(host, "192.168.0.0", "255.255.255.0") ||
      isInNet(host, "192.168.1.0", "255.255.255.0")) ||
      return "DIRECT";
  else
    return "PROXY 192.168.1.5:8080;" +
           "DIRECT";
}
```

Der Webserver der die proxy.pac zur Verfügung stellt muss dabei folgenden MIME Type verwenden:

```
application/x-ns-proxy-autoconfig pac
```

bei Apache wird das in der mime.types festgelegt und ist Standard.

Erweiterungen:

- mehrere Proxy Server einbinden, falls einer ausfällt, hierzu muss nur ein weiterer String PROXY IP:Port; in der proxy.pac angegeben werden.

- Proxy nur für bestimmte Protokolle (z.B. http) hierzu kann vor Rückgabe des PROXY Strings noch die URL mit Regular Expressions untersucht werden (shExpMatch)

