## 8.4 Praktische Aufgaben

Der derzeitig immer noch gültige Versuchsaufbau ist in Abbildung 71 dargestellt.

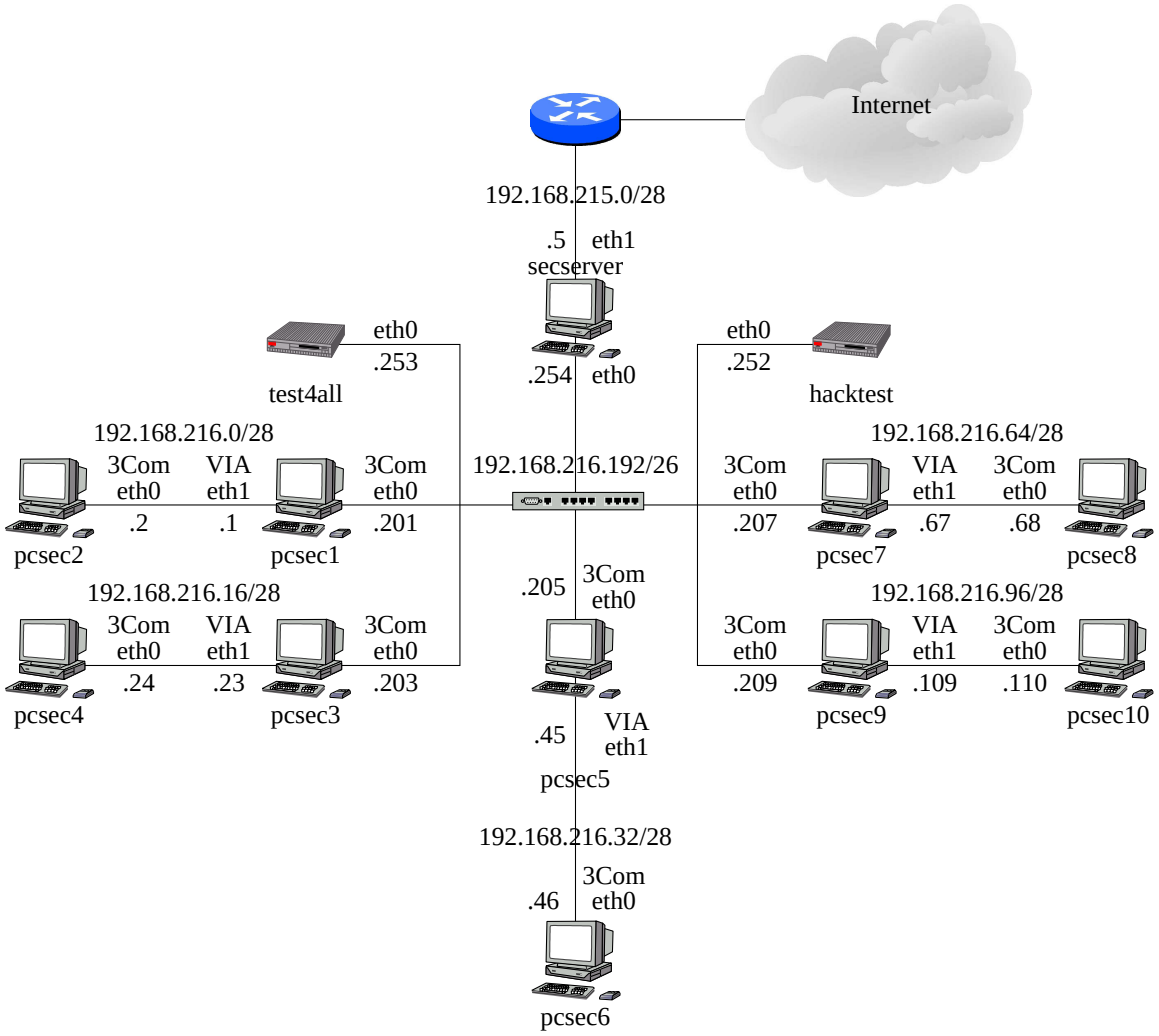


Abbildung 71: Der Versuchsaufbau für die weiteren Versuche des Praktikums

Für alle folgenden Aufgaben ist eine schriftliche Ausarbeitung zu erstellen, die folgendes beinhaltet:

- Angabe aller gemachten Konfigurationsänderungen mit Begründung.
- Angabe aller gemachten Tests mit Aufführung der bekommenen Meldungen inklusive Interpretation.

- Auflistung der erzeugten Logeinträge mit den Erklärungen, welche Schlüsse daraus abzuleiten sind.

Diese Ausarbeitung ist zum nächsten Termin per Mail an die E-Mail Adresse `secp@nm.informatik.uni-muenchen.de` zu schicken. Als SMTP-Gateway kann der `secserver` verwendet werden. Ein Tip ist das Führen eines Logfiles, so daß alle gemachten Änderungen mitprotokolliert werden.

### 8.4.1 Squid

1. Deaktivieren Sie alle Paketfilterregeln und sorgen Sie dafür, daß alle Dienste auf allen Interfaces hören und von allen Adressen aus dem Netz `192.168.216.0/24` erreichbar und nutzbar sind.

2. Installieren Sie Squid von der SuSE 8.0 Distribution.

3. Tragen Sie in der `/etc/resolv.conf` die `192.168.216.254` als einzigen Nameserver ein.

4. Nun folgt die Squidkonfiguration. **Überprüfen Sie alle logischen Teilschritte mittels in der Vorbereitung beschriebener Tests und der Logfileeinträge. Gehen Sie immer schrittweise vor!**

   - Setzen Sie den Squidport auf 8888. ICP Port ist standardmäßig 3130.
   - Setzen Sie die bei einer Fehlermeldung angezeigte Mailadresse auf den `root`-Account Ihres Rechners.
   - Es sind nur Domains aus `.de` und `.org` erlaubt.
   - Bei allen Definitionen der übergeordneten Proxies soll dafür gesorgt werden, daß
     - kein von diesem Cache geholtes Object gespeichert wird.
     - keine ICP-Anfrage erfolgt.

     Alle übergeordneten Proxies sind als `parent` zu betrachten.
   - Implementieren Sie ein Proxychaining, daß hier beispielhaft für `pcsec4` und `pcsec3` dargestellt ist. Bitte transferieren Sie die hier gemachten Angaben auf Ihre spezielle Situation:
     - Proxychain für `pcsec4`:
       * `pcsec4.secp.nm.informatik.uni-muenchen.de` und `pcsec3.secp.nm.informatik.uni-muenchen.de` sind lokal aufzulösen. Hierzu ist keine Userauthentisierung nötig.
       * Schicken Sie den Rest an den in der Hirarchie über Ihnen stehenden Cache `pcsec3 192.168.216.23`.

    - Erlauben Sie nur Ihren Rechner und den `pcsec3`, Ihren Cache zu verwenden.
    - Laden Sie sich aus dem Internet den aktuellen Squid-Quellcode herunter und entpacken Sie ihn. Installieren Sie das Authentisierungsprogramm `ncsa_auth`, erzeugen Sie ein Authentisierungsfile mit `htpasswd` und aktivieren Sie die Authentisierung in der `/etc/squid.conf`.
    - Erlauben Sie alle URLs in der Domain `secp.nm.informatik.uni-muenchen.de` ohne Authentisierung.
    - Verlangen Sie für alle restlichen Verbindungen in den erlaubten Domains `.de` und `.org` Authentisierung.
  - Proxychain für `pcsec3`:
    - `pcsec4.secp.nm.informatik.uni-muenchen.de` und `pcsec3.secp.nm.informatik.uni-muenchen.de` bzw. `pcsec3-switch.secp.nm.informatik.uni-muenchen.de` sind lokal aufzulösen. Hierzu ist keine Userauthentisierung nötig.
    - Schicken Sie den Rest an den in der Hirarchie über Ihnen stehenden Cache `secserver 192.168.216.254` HTTP Port 3128, ICP Port 3130. Dabei müssen Sie bedenken, dass zumindest der `secserver` für den Zugriff auf die von Ihnen direkt aufzulösenden Domains erlaubt sein muß.
    - Sorgen Sie dafür, daß `pcsec1`, `pcsec5`, `pcsec7` und `pcsec9` nur die Webseiten `pcsec4.secp.nm.informatik.uni-muenchen.de` und `pcsec3.secp.nm.informatik.uni-muenchen.de` bzw. `pcsec3-switch.secp.nm.informatik.uni-muenchen.de` über Ihren Cache erreichen können.
    - Sorgen Sie dafür, daß `pcsec4` und Ihr Rechner alle erlaubten Domains ohne Userauthentisierung erreichen können.

### 8.4.2 FWTK

1. Deaktivieren Sie FTP und Telnet in der `/etc/inetd.conf`.
2. Kompilieren und Installieren Sie das FWTK nach Anleitung.
3. Aktivieren Sie den Authentisierungsserver `authsrv` auf Port 7777 über die `/etc/inetd.conf` und die `/etc/services`.
4. Legen Sie mit dem Programm `authsrv` Testuser (`testsec1`, `testsec2`, Passwort analog zur User-ID) an.
5. Aktivieren Sie den `http-gw` auf Port 8282.

**SOCKS-Client**

In einigen Browsern, z.B. Netscape, ist bereits der SOCKS-Support eingetragen, so daß sie ohne Anpassungen mit einem SOCKS-Server zusammenarbeiten können. Viele andere Programme sind aber nicht von vornherein SOCKS-fähig. Es gibt zwei Möglichkeiten, einen Client SOCKS-fähig zu machen. Die Erste ist, das Programm neu zu übersetzen. Da diese Methode sehr umständlich ist, wird sie nur verwendet, wenn der benötigte Client nicht mit der zweiten Methode socksifiziert werden kann. Die zweite Möglichkeit ist, einen dynamisch gelinkten Client davon zu überzeugen, vor allen anderen Libraries eine SOCKS-Library zu laden, die die Netzwerkaufrufe abfängt und socksifiziert. Beim Dante-Client wird hierzu das bereits fertige Programm `/usr/bin/socksify` mitgeliefert, das beim Clientstart die `libdsocks.so` lädt und über die Environment-Variable `LD_PRELOAD` die SOCKS-Libraries permanent vor die eigentlichen Netzwerkcalls setzt.
Der FTP-Client wird dann z.B. so aufgerufen:

```
socksify ftp
```

Wurde die SOCKS-Library nicht in der Environment-Variable `LD_PRELOAD` vor die eigentlichen Aufrufe gestellt, so muß jede Verbindung gesondert socksifiziert werden.
Möchte man nun den Dante-Client konfigurieren, so geschieht das über das Konfigurationsfile `/etc/socks.conf`.
Hier ein Konfigurationsbeispiel für den Dante-Client in `socks.conf` [Bart 01]:

```
logoutput:       syslog
resolveprotocol: udp    # default

# (1) direkte Verbindung zu den Nameservern
# route {
#        from: 192.168.1.0/24
#        to: 193.101.111.0/24 port = domain
#        via: direct
# }

# (2) loopback = hostinterne Verbindungen
route {
        from: 0.0.0.0/0
        to: 127.0.0.0/8
        via: direct
        command: connect udpassociate
        # everything but bind, bind confuses us.
}

# (3) Internes Netzwerk
route {
```