Privater Schlüsseldienst

VON JÜRGEN DANKOWEIT

Authentifizierungen mit Username und Paßwort sind jedem Administrator und Benutzer bekannt. In Situationen, in denen es nicht möglich ist, ein Paßwort zu übertragen, bedient man sich zum Verschlüsseln von Daten und für die Authentifizierung der PSK-Methode.

Kommunizieren Programme oder Systeme im privaten Kontext miteinander, wird bei der Authentifizierung oft mit symmetrischer Kryptographie gearbeitet. Ein geeignetes Verfahren ist PSK (Private Shared Key oder Pre Shared Key), das beispielsweise in Wireless LANs im Modus WPA/PSK (Wireless Protected Access with Private Shared Keys) anzutreffen ist. Bei diesem Verfahren werden keine Benutzernamen und Paßwörter ausgetauscht, sondern es wird mit gemeinsamen Schlüsseln gearbeitet. Wenn der Client mit einem Server kommuniziert, werden die Daten entweder blockweise oder als Datenstrom über PSK verschlüsselt. Da der Schlüssel dem Client und Server bekannt ist, lassen sich die Informationen wieder entschlüsseln, was eine Art der Authentifizierung ist. Der Client ist dann also authentifiziert, weil die Schlüssel identisch sind und er ist daher auch autorisiert, weiterhin Daten zu senden.

Schlüssel

Dieses Verfahren ist schnell und einfach konfiguriert und für Netzwerke im privaten Kontext sicherlich ausreichend. Unbedingt sollte darauf geachtet werden, daß der Schlüssel so komplex wie möglich gewählt wird und es wurde bereits mit Hilfe der Rechenleistung mehrerer NVidia-Grafikkarten PSK entschlüsselt, weshalb dieses Verfahren nicht mehr in Unternehmen angewendet werden sollte – aber WLAN ist in Unternehmen ja grundsätzlich verpönt.

Bild 1 zeigt die Funktionsweise anhand des für den *bind* wichtigen Steuerprogramms *rndc*.

rndc steuert den DNS-Daemon bind. Damit läßt er sich unter anderem anhalten, neu starten oder seine Konfiguration prüfen. Damit nicht jeder Benutzer im Intranet Zugriff auf den DNS-Daemon bekommt und Schaden anrichten kann, wird die Verbindung über PSK verschlüsselt. Der Client hat somit nur dann die Möglichkeit, mit bind zu kommunizieren, wenn er den gleichen Schlüssel besitzt, was bedeutet, daß nur dann der Client authentifiziert ist. Das Paket *bind* enthält neben dem eigentlichen

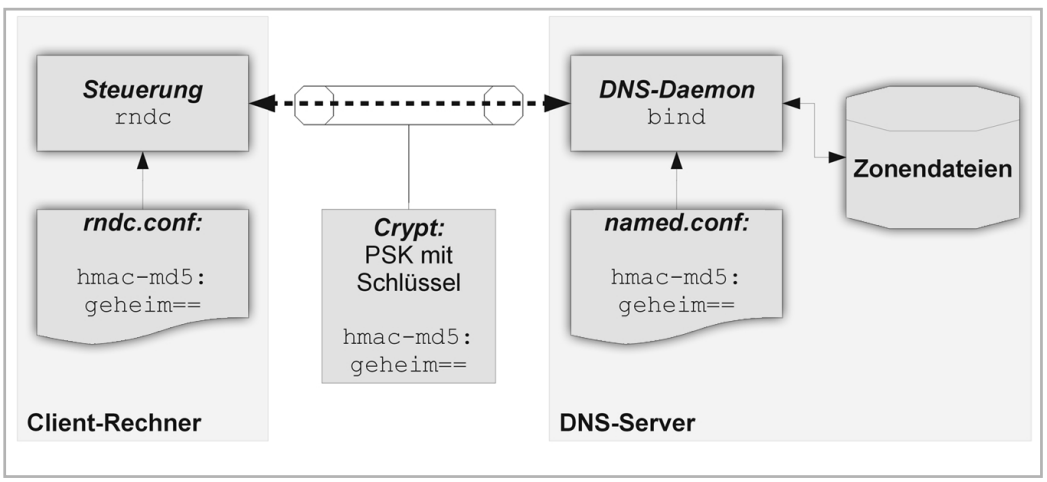

Bild 1: Funktionsdiagramm zu PSK mit rndc

Daemon-Prozeß *named* diverse Tools. Zwei davon sind für die Konfiguration wichtig: *rndc*(8) und *rndc-confgen*(8).
Weil sich die folgenden Schritte nur als root-User ausführen lassen, ist es wichtig, sehr sorgfältig zu arbeiten!
Zunächst legt man eine Basiskonfiguration an, auf der die weiteren Optionen aufbauen:

```
# rndc-confgen -a
# cat /etc/named/rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "schluessel==";
};
```

Der zweite Schritt ist die vollständige Konfiguration des Programms *rndc*:

```
# cat /etc/namedb/rndc.conf
options {
  default-server  dns_router;
  default-key  dns_router_key;
};

server dns_router {
  key  dns_router_key;
  addresses {
    dns.intranet.net;
  };
};

key dns_router_key {
  algorithm hmac-md5;
  secret "schluessel==";
};
```

Zuerst werden die Standardeinträge definiert. Sie gelten, wenn *rndc* ohne weitere Optionen aufgerufen wird, in diesem Beispiel sind das *dns_router* und *dns_router_key*. Der Eintrag *dns_router* verweist auf den Abschnitt *server dns_router*. Hier wird schließlich festgelegt, welcher Schlüssel gelten soll und auf welcher Adresse der DNS-Daemon kommuniziert.
In der Konfiguration */etc/named/named.conf* des DNS-Daemons ist nur eine Ergänzung notwendig:

```
# include "rndc.key";
```

Achtung FreeBSD-Administratoren, die den DNS-Daemon in einer Jail betreiben: Die Basiskonfiguration sollte nicht in der Jail vorgenommen werden, sondern auf dem Host. Die Datei */etc/namedb/rndc.key* kopiert man vom Host in das Verzeichnis */etc/namedb/* in der Jail und konfiguriert anschließend den DNS weiter wie beschrieben.
Um den DNS-Daemon vom Arbeitsplatz des Administrators aus zu steuern, muß die Konfigurationsdatei *rndc.conf* in das Verzeichnis */etc/namedb* des betreffenden Rechners kopiert werden.
Zum Abschluß wird getestet, ob die Kommunikation zwischen *rndc* und dem DNS sauber funktioniert:

```
$ rndc status
number of zones: 13
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

Bei eventuellen Schwierigkeiten sollte man sich die Protokolldatei */var/log/messages* genauer ansehen, in der alle Probleme dokumentiert sind. ◆