Ecki's Place

IPcop v1.3 und v1.4

Der Fernwartungszugang

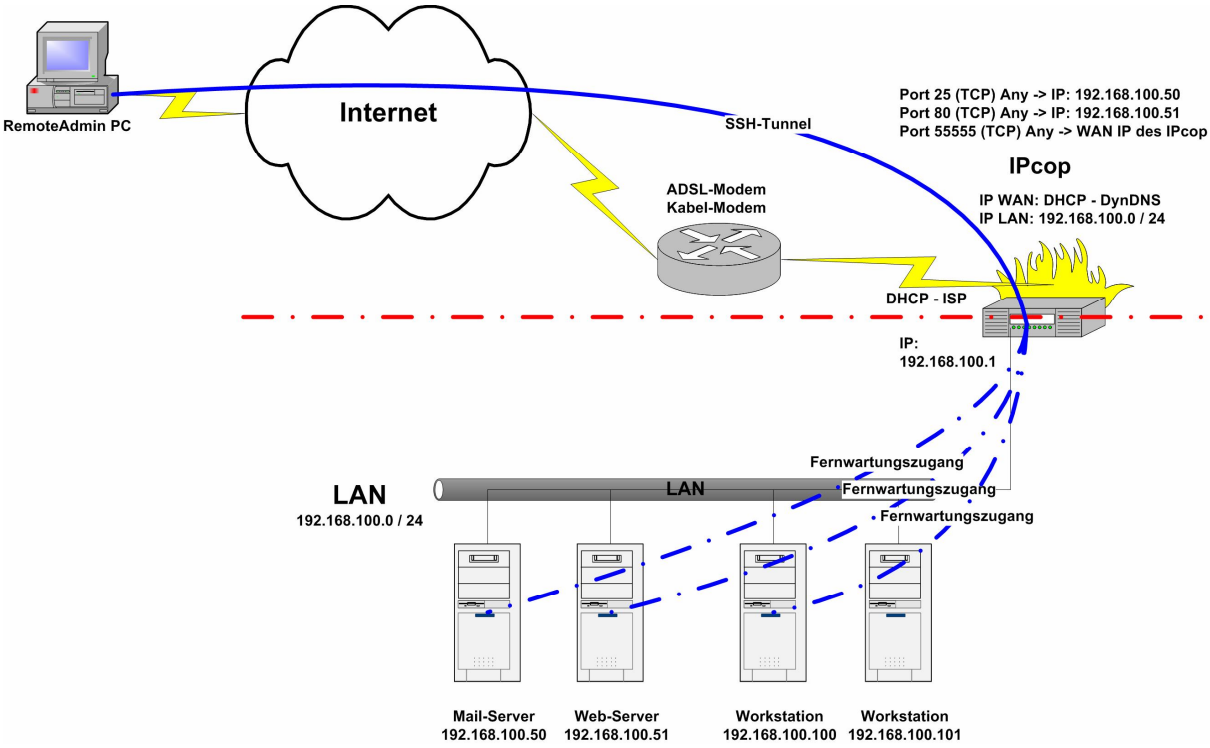
Grundsätzliches

Dieses Tutorial setzt eine Grundkonfiguration wie in dem Tutorial zum Basissetup des IPcop voraus. Die IP-Adressen müssen gegebenenfalls an die lokalen Vorgaben angepasst werden.
Es werden die Einstellungen besprochen, die nötig sind, um den IPcop vom Internet her warten und konfigurieren zu können. Es wird dabei besonderer Wert auf Sicherheit gelegt. Des weiteren wird gezeigt, wie ohne VPN sicher auf LAN-Ressourcen zugegriffen werden kann.

Vorbereitung

1. Grundkonfiguration des IPcop nach einem der folgenden Tutorials:
   http://www.gutzeit.ch/ipcop/pdf/gr_konf_adsl_dyn.pdf
   http://www.gutzeit.ch/ipcop/pdf/gr_konf_adsl_fix.pdf
   http://www.gutzeit.ch/ipcop/pdf/gr_konf_kabel_dyn_fix.pdf

Beispiel eines Fernwartungszugangs



Was ist ein Fernwartungszugang?

Ein Fernwartungszugang im eigentlichen Sinne ermöglicht es einem Benutzer, von einem definierten, bzw. je nach Konfiguration auch beliebigen Internetzugang auf die Konfigurationsschnittstellen des IPcop zugreifen zu können. An Schnittstellen bietet der IPcop sowohl einen SSH-Server (keinen Client) als auch das Webfrontend, sowohl mit, als auch ohne SSL-Verschlüsselung an. Des Weiteren lässt sich natürlich auch über VPN auf den IPcop zugreifen.

- Aus Sicherheitsgründen scheidet die unverschlüsselte Variante über „http://“ für den Fernwartungszugang generell aus, da hier das Admin-Passwort im Klartext übertragen wird.
- Auch der SSL-geschützte Zugang über „https://“ ist nur eingeschränkt zu empfehlen. Da der TCP-Port 445 zur Firewall hin geöffnet werden muss, sind BruteForce-Angriffe auf das Admin-Passwort und Angriffe auf den Apache Web-Server möglich. Des weiteren sperren einige Provider den Port 445 um ihre Kunden vor ihrer eigenen Unwissenheit/Dummheit zu schützen (viele Analog-, ISDN-, ADSL- und Kabelmodem-Nutzer neigen dazu, ihre Festplatten zum Internet hin freizugeben, indem sie den „Client für Microsoft-Netzwerke“ und die „Datei- und Druckerfreigabe“ auch auf ihrem Internet-Interface aktiviert haben). Wie man auf dem Apache Web-Server den SSL-Port ändert, wird in einem der folgenden Absätze gezeigt.
- SSH und VPN stellen sehr sichere Methoden für den Fernwartungszugang dar. Da das Thema VPN in einem eigenen Tutorial behandelt wird, beschränke ich mich hier auf den Zugriff per SSH.

Wofür brauche ich einen Fernwartungszugang

Nicht immer steht der IPcop unter dem eigenen Schreibtisch, sondern befindet sich manchmal x Kilometer entfernt in einer Filiale, oder die Firewall zuhause braucht dringend eine Konfigurationsänderung, man selber sitzt aber im Büro, oder ein guter Freund (aber leider ein DAU;-) will unbedingt noch heute sein neues Filesharingtool ausprobieren, dazu müssen aber Ports auf der Firewall geöffnet werden und der Freund wohnt in einer anderen Stadt, oder man möchte ganz einfach einen geschützten Zugang zum Heimnetzwerk aufbauen, um entweder Wartungsarbeiten an der LAN-Infrastruktur vornehmen zu können, oder um die neuen Mails abzufragen, oder …

Um dies alles zu ermöglichen gibt es mehrere Möglichkeiten.

1. Man richtet einen Fernwartungszugang per SSL ein. Damit ist jedoch kein LAN-Zugriff möglich.

2. Man baut ein VPN mit dem IPcop auf.

3. Man benutzt den integrierten SSH-Server für all diese Aufgaben.

Die Lösungsansätze

SSL, und wie ich den SSL-Port des Apache Web-Servers ändere

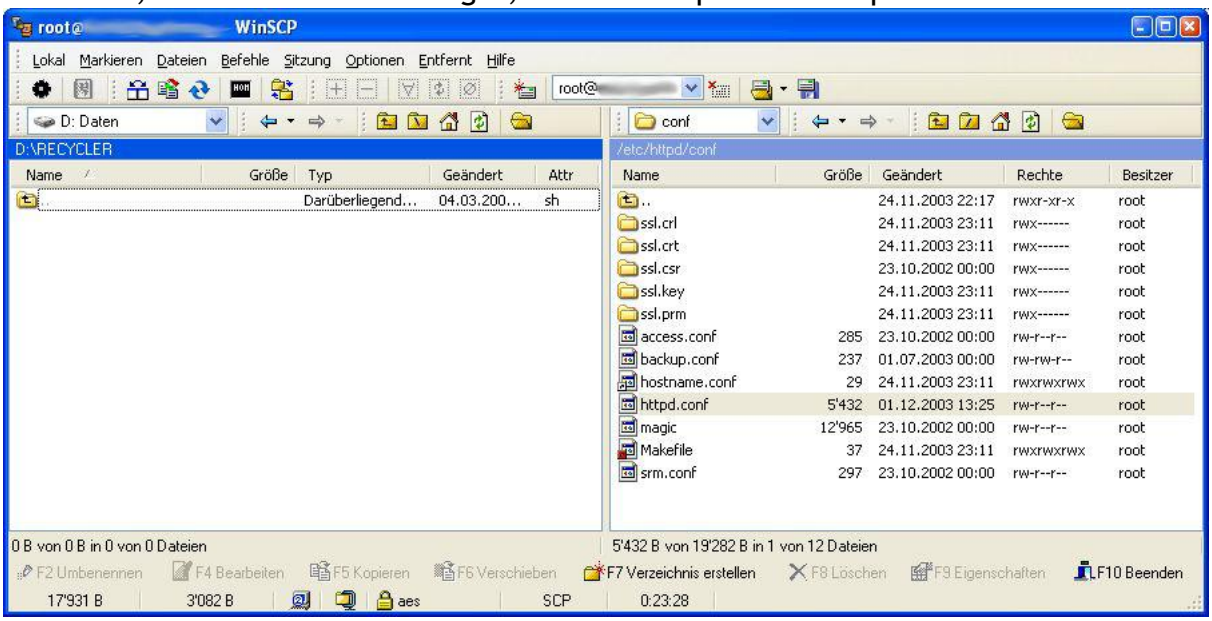
Die Sicherheitsprobleme bei dieser Art des Zugangs habe ich schon im vorangegangenen Absatz beschrieben. Um trotzdem ein wenig mehr Sicherheit zu erreichen, oder um Providersperren zu umgehen, empfehle ich, den SSL-Port des Apache Web-Servers auf einen so genannten „High Port“ zu verlegen.

Bevor jetzt ein Aufschrei wegen „Security by Obscurity“ (Sicherheit durch Unklarheit) kommt: Ich weiss, das dies keine wirkliche Sicherheit schafft, es verhindert jedoch trotzdem zu 99%, dass Scriptkiddies den offenen Port finden. Eine 5-monatige Analyse meiner SNORT-Logfiles hat ergeben, dass in diesem Zeitraum ca. 4500 Port-Scans auf den Standardports (1-1024) stattgefunden haben, Verbindungsversuche auf einem „High Port“, den ich für SSH verwende jedoch ausschliesslich von mir gemacht wurden.

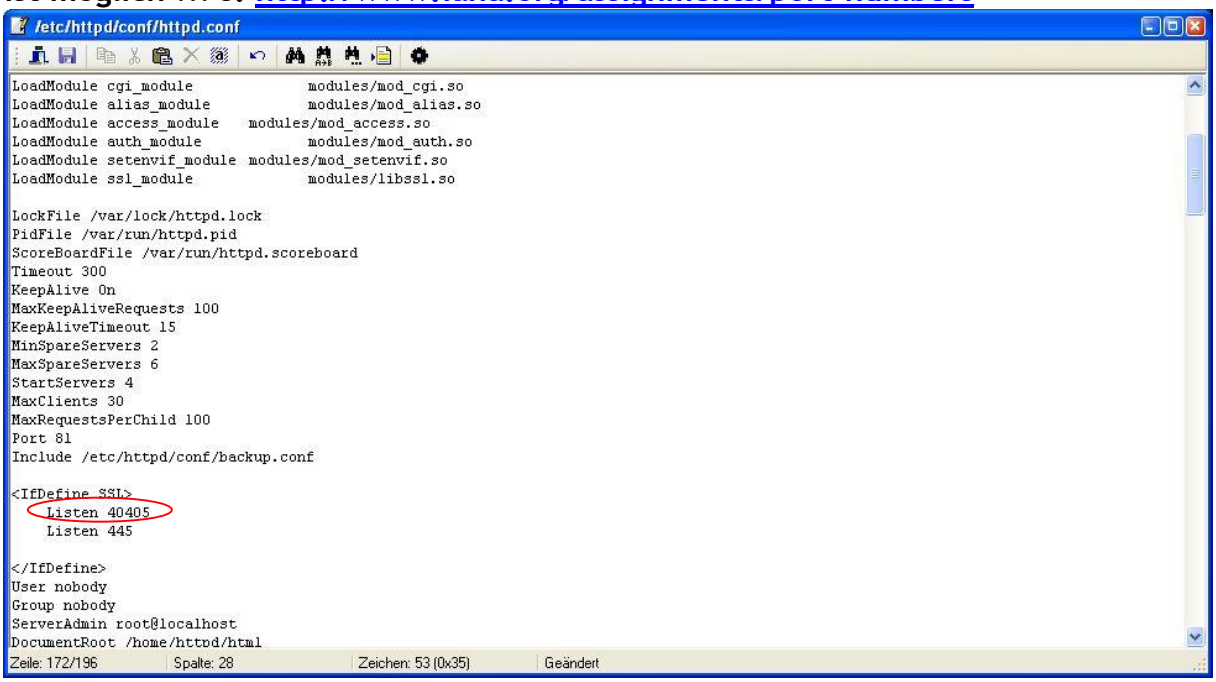
Zurück zum Thema.
Für die folgenden Änderungen benötigen wir entweder einen SSH-Client wie z. B. PuTTY, oder noch komfortabler WinSCP
Downloadlinks: http://www.gutzeit.ch/ipcop/tools.shtml

In diesem Tutorial verwende ich WinSCP, da es sich auch problemlos von jedem Linux-Neuling bedienen lässt.

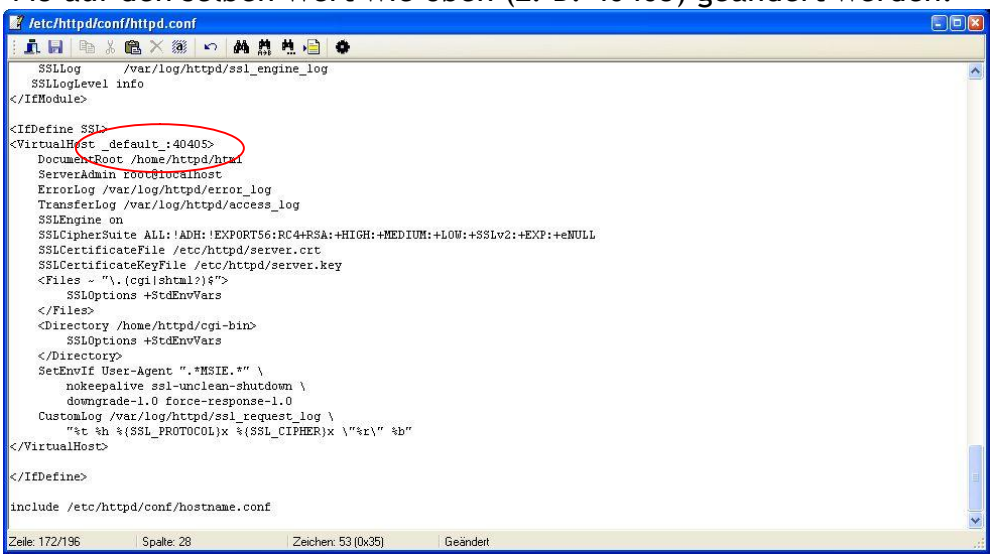
Die Datei, die es zu bearbeiten gilt, ist /etc/httpd/conf/httpd.conf



Mit einem Rechtsklick -> „Bearbeiten“, öffnet sich die Datei im Editor. Hier finden sich 2 Stellen, an denen der SSL-Port geändert werden muss. Die erste Stelle befindet sich ziemlich am Anfang der Datei. Hier kann, zusätzlich zum schon vorhandenen Eintrag „Listen 445“ ein neuer Eintrag mit dem gewünschten „high Port“, z. B. 40405 hinzugefügt werden. **!!! Praktisch jeder Port oberhalb von 1024 ist möglich !!! s.** http://www.iana.org/assignments/port-numbers



Die zweite Fundstelle befindet sich fast am Ende der Datei. Hier muss der Eintrag 445 auf den selben Wert wie oben (z. B. 40405) geändert werden.



Nach dem Abspeichern der Datei fehlt jetzt nur noch ein Neustart des Apache Web-Servers, um die geänderten Einstellungen zu aktivieren.

Dies erreichen wir mit folgenden Befehlen an der Kommandozeile, welche aus WinSCP heraus mit „Ctrl + T“ aufgerufen werden kann:

Bei IPcop 1.3.x: „/usr/sbin/killall httpd“ und anschliessend „/usr/sbin/httpd -DSSL“
Bei IPcop 1.4.x reicht: „killall httpd“ und anschliessend „httpd“

Beim anschliessenden Test sollt nun lokal eine Verbindung mit https://192.168.100.1:40405 möglich sein.

Um es ganz richtig zu machen, muss noch die Datei /home/httpd/cgi-bin/portfw.cgi via WinSCP angepasst werden. Dazu muss in der Zeile
„my @tcp_reserved = (81,222,445)“ noch der neue Port hinzugefügt werden.

Wie der Fernwartungszugang konfiguriert werden muss, um auch vom Internet her auf den IPcop zugreifen zu können, wird im Abschnitt „Zugriff vom Internet auf den IPcop“ exemplarisch für SSH beschrieben. Um den Zugriff direkt über HTTPS zu erhalten, muss natürlich die Portnummer auf den benutzten Wert geändert werden.

Bevor diese Zugangsart jedoch eingesetzt wird, bitte ich alle Leser, auch den Abschnitt zum Thema SSH genau zu lesen. Mit den dort vorgestellten Methoden können die meisten Sicherheitsprobleme der oben gezeigten Lösung vermieden werden.

VPN, der komplette Netzzugang
Da es sich bei dem Thema VPN um ein recht komplexes Gebiet handelt, werde ich es in diesem Tutorial nicht im Detail behandeln. Wer sich mit dieser Art des Netzzugangs beschäftigen möchte, dem empfehle ich mein VPN-Tutorial.

SSH, ein fast universelles Tool

Mein absoluter Favorit für den Fernwartungszugriff ist SSH (**S**ecure **SH**ell). SSH ist das schweizer Taschenmesser für den Zugriff auf einen beliebigen Remote-PC, vorausgesetzt, auf diesem läuft ein SSH-Server.

Was genau ist eigentlich SSH?

SSH ist eine Protokollsuite für den Remotezugriff auf entfernte Maschinen und kann damit u. a. als Ersatz für das unsichere „Telnet“ angesehen werden. Mit SSH können sowohl Kommandos auf der entfernten Maschine ausgeführt werden, als auch Dateien dorthin, oder von dort, übertragen werden, wodurch es auch zu einem sicheren Ersatz für „FTP“ wird. Die Sicherheit wird erreicht durch eine hochwertige Verschlüsselung aller Datenübertragungen (3DES, Blowfish,…) und starke Authentifizierung (RSA, DSA) zu Beginn einer Session. Zusätzlich können über eine SSH-Verbindung beliebige Protokolle getunnelt werden, was es für unsere Zwecke geradezu prädestiniert. SSH existiert in zwei Versionen, SSH1 und SSH2. Kurz gesagt, handelt es sich bei SSH2 um eine Neuimplementation von SSH1 (welches nicht mehr weiterentwickelt wird), wobei die Sicherheit, Performance und Portabilität verbessert wurde. Es sollte daher immer SSH2 verwendet werden, wenn eine Verbindung zum IPcop aufgebaut wird.
Weitere Informationen zu SSH sind z. B. hier zu finden:
http://www-user.tu-chemnitz.de/~hot/ssh/ (deutsch)
http://www.employees.org/~satch/ssh/faq/ (englisch)

Was kann man nun mit SSH machen?

„Ja, aber da lande ich ja auf der Kommandozeile und ich kenne mich mit Linux nicht aus. Ausserdem hat der IPcop ja so ein schönes Webfrontend das ich auch nutzen will.“ werden Einige jetzt schreien.

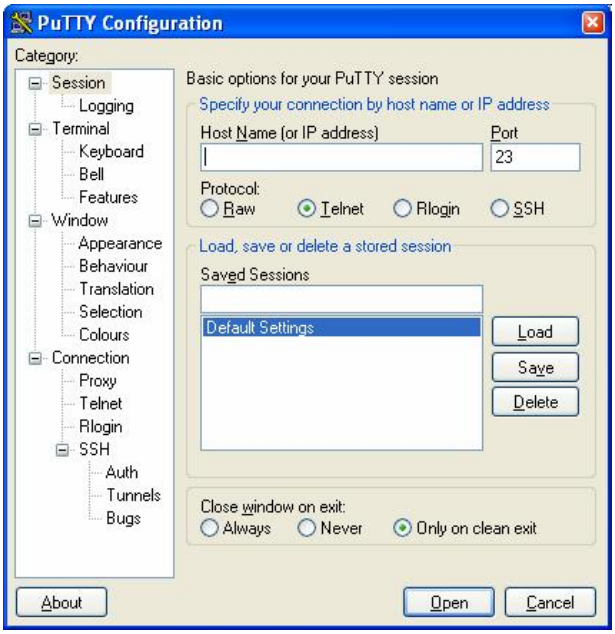
Ein Ausflug auf die Kommandozeile ist nur in ganz wenigen Situationen wirklich nötig. Das in einem der vorherigen Abschnitte vorgestellte Tool „WinSCP“ funktioniert mit Hilfe der SSH-Protokollsuite. Damit lassen sich Dateien sehr komfortabel in Windowsmanier bearbeiten. Zusätzlich liesse sich der IPcop damit sogar als privater und sicherer „FTP-Server“ (eigentlich SCP-Server, **S**ecure **CoP**y, als Alternative für FTP) verwenden. Für diesen Einsatzzweck wurde der IPcop jedoch nicht entwickelt.
Noch mächtiger und vielfältiger ist das Tool PuTTY, welches ebenfalls schon weiter oben angesprochen wurde. Obwohl es als einfacher Telnet-/SSH-Client für Windows daher kommt, schlummern unter der Haube gewaltige Möglichkeiten. Mit diesem Tool ist es möglich, die Fähigkeiten von SSH fast vollständig auszuschöpfen. Neben der Funktion als simpler SSH1/2-Client, erlaubt es auch die Konfiguration von SSH-Tunnel, um auf Dienste auf dem IPcop, z. B. HTTP(S), bzw. im LAN, z. B. (RDP, VNC, pcAnywhere, etc.) zuzugreifen.

Diese Fähigkeiten und die sich daraus ergebenden Möglichkeiten, möchte ich im Folgenden näher erläutern.

PuTTY

Der SSH1/2-Client PuTTY bietet primär die Möglichkeit, sich von einem beliebigen Internetzugang per SSH mit dem IPcop zu verbinden. Die dazu nötigen Schritte sind leicht erklärt. Nach dem Download der „putty.exe“ kann diese direkt per Doppelklick gestartet werden. Eine Installation ist nicht nötig.

Das Startfenster



Für die Grundkonfiguration müssen nur folgende Felder ausgefüllt werden:

Host Name:	Ein passender Eintrag ist root@name.dyndns.org, oder root@192.168.100.1, je nach dem, über welches Interface die Verbindung aufgebaut werden soll. Hiermit wird gleich der Username „root“ übergeben, so dass anschliessend nur noch das Passwort eingetippt werden muss.
Protocol:	Muss auf „SSH“ gewechselt werden, da der IPcop kein Telnet kann.
Port:	ist beim IPcop standardmässig 222
Saved Sessions:	Hier kann ein „sprechender“ Name für die Session eingegeben werden.

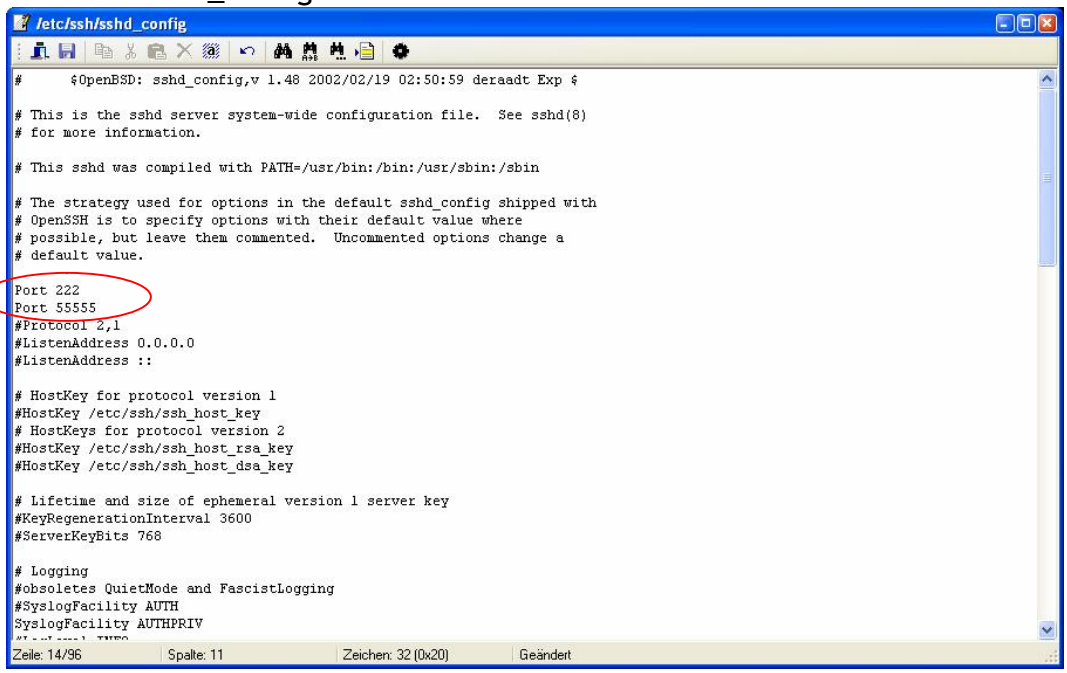
Mit „Save“ wird diese Konfiguration für die spätere Verwendung gespeichert.

Weiter sinnvolle Anpassungen können unter „Window“ gemacht werden, indem der Wert „Lines of scrollback“ erhöht wird, bei mir z. B. auf 2000. Das bedeutet, dass man in der Lage ist, im Terminalfenster 2000 Zeilen zurückzuscrollen, bevor sie aus dem Puffer gelöscht werden.

Nun sind wir in der Lage, per SSH auf den IPcop zu verbinden. Bei einem Test aus dem LAN sollte, nach der Eingabe des Passworts, eine Kommandozeile angezeigt werden. Mit „Ctrl + D“ kann die Session jederzeit beendet werden.

Zugriff vom Internet auf den IPcop

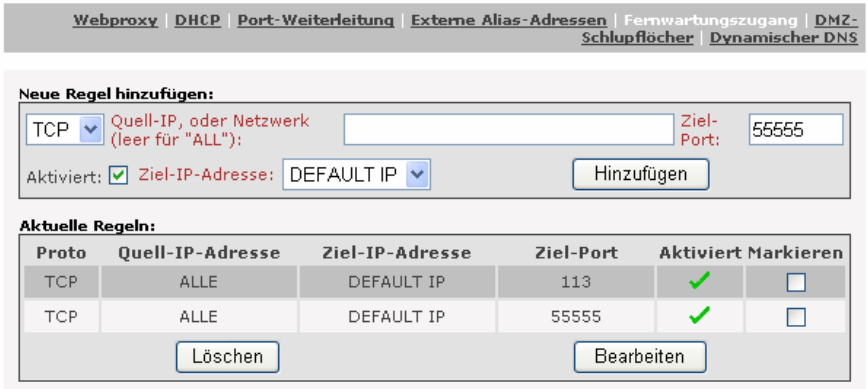
Um auch vom Internet her per SSH auf den IPcop zugreifen zu können, muss ein Fernwartungszugang eingerichtet werden.
Auch bei SSH ziehe ich es vor, nicht den IPcop Standard-Port „222“ zu verwenden, sondern lasse meinen SSH-Server auf einem beliebigen „high Port“ lauschen. Diese Änderung ist leicht gemacht. Mit WinSCP auf den IPcop verbinden und die Datei /etc/ssh/sshd_config zum Bearbeiten öffnen.



Unter der Zeile in der „Port 222“ steht, eine weitere Zeile mit „Port xxxxx“ einfügen und die Datei anschliessend speichern. Nach einem Neustart des SSH-Serverdienstes mit folgendem Kommando, lauscht der SSH-Server nun auf beiden Ports.

/usr/local/bin/restartssh

Im Webfrontend fehlt nun noch folgender Eintrag, um den Zugriff auch vom Internet her zu gestatten. (v1.4 zu finden unter „Firewall“ -> „Externer Zugang“)

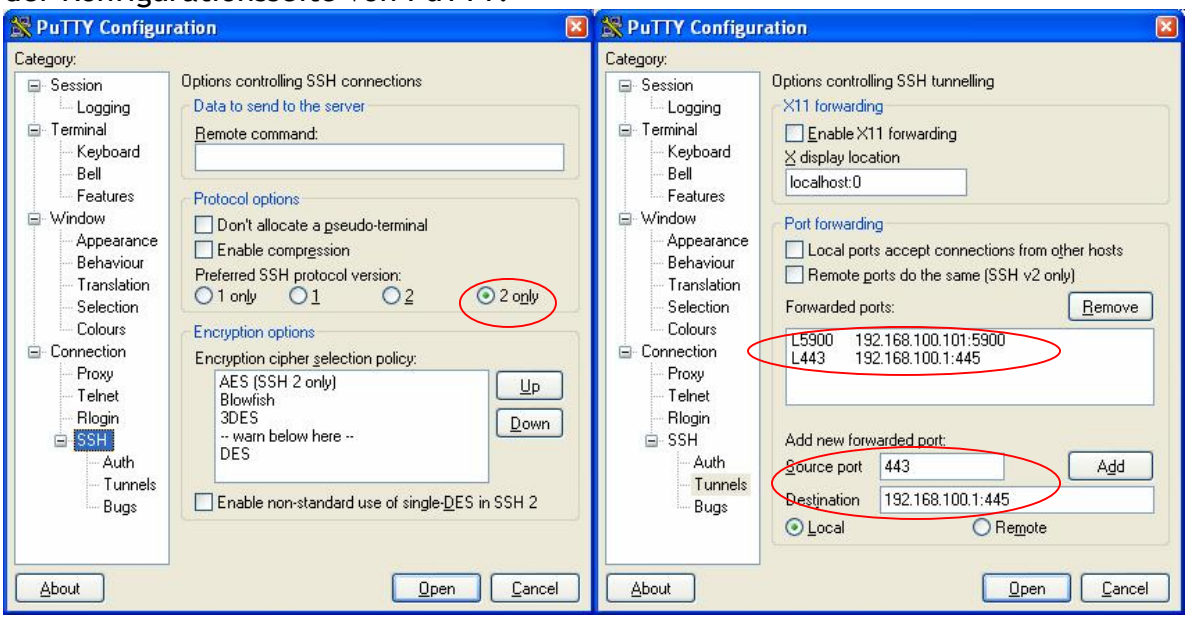


Nun steht dem Zugriff per SSH auch vom Internet her nichts mehr im Wege.

Der SSH-Tunnel Teil 1

Mit den oben gemachten Einstellungen kann nun jedermann auf dem offenen Port 55555 versuchen, eine SSH-Session zu starten. Dass das Root-Passwort mit Bedacht zu wählen ist, sollte jedem klar sein. Mindestens 14 Zeichen, Gross-/Klein-Buchstaben, Zahlen und Sonderzeichen sollten darin schon vorkommen.
Des Weiteren ist es sehr zu empfehlen, wenn irgend möglich die „Quell-IP, oder Netzwerk“ zu definieren, um den Fernwartungszugang nicht dem ganzen Internet zugänglich zu machen, sondern z. B. auf das Netz des Arbeitgebers zu beschränken. DynDNS-Namen werden im Webfrontend leider nicht akzeptiert, sondern nur IP-Adressen oder Netzwerke der Art „123.123.123.0/24“.

Um nun einen Tunnel zum IPcop aufzubauen, über den das Webfrontend via HTTPS wie gewohnt zu bedienen ist, braucht es nur wenige zusätzliche Einstellungen auf der Konfigurationsseite von PuTTY.



Nach dem eintragen der Daten auf der Startseite (root@name.dyndns.org, SSH, Port 55555), sollte als erstes unter „SSH“ in den „Protocol options“ auf „2 only“ gewechselt werden, damit ausschliesslich SSH2 verwendet wird.
Anschliessend müssen unter „Tunnels“ noch ein oder mehrere Tunnel konfiguriert werden.
Der Zugriff auf das Webfrontend findet beim IPcop per Default auf dem Port 445 statt. Da es sich hierbei um eine HTTPS-Verbindung handelt, dessen Standard-Port eigentlich 443 ist, kann ich mit folgender Regel einen Tunnel für das Webfrontend des IPcop erstellen:

Source Port:	443 (sobald ich im Browser https:// tippe, wird immer ein Verbindungsaufbau auf dem Port 443 versucht)
Destination:	GREEN-IP-Adresse des IPcop, gefolgt von einem Doppelpunkt und dem gewünschten Ziel-Port. In unserem Fall 192.168.100.1:445

Mit „Add“ wird die Regel übernommen. Für die spätere Verwendung muss die Konfiguration jedoch noch zusätzlich auf der Startseite gespeichert werden.

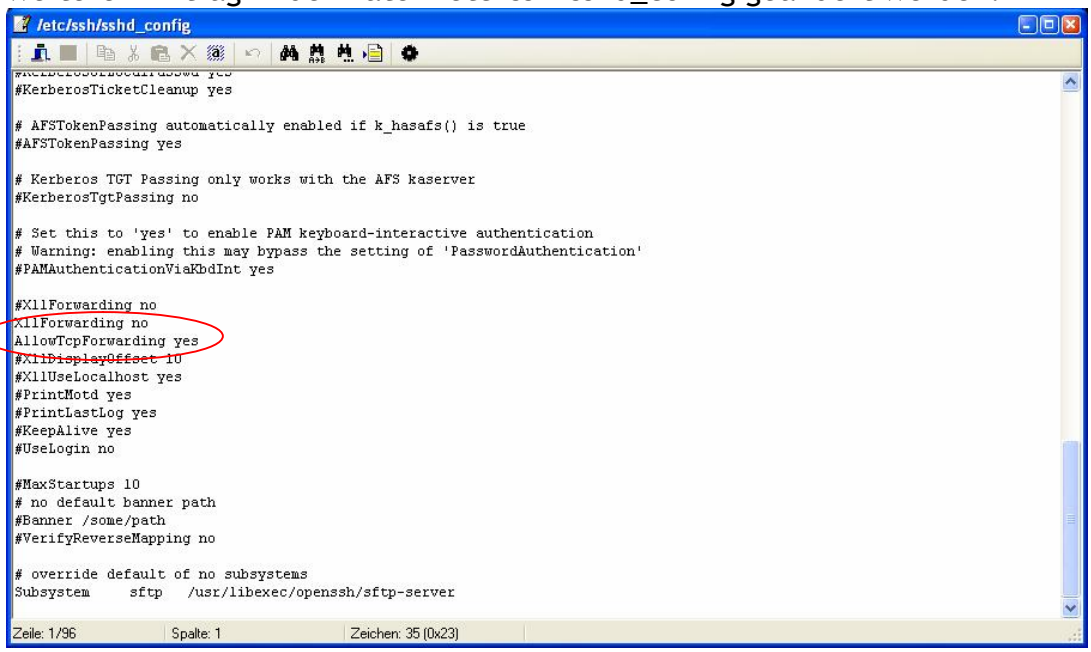
Wenn ich jetzt z. B. bei der Arbeit bin, und meinen IPcop zuhause konfigurieren möchte, starte ich die soeben gespeicherte Verbindung. Nun verbindet PuTTY über den Port 55555 per SSH auf das rote Interface des IPcop. Sobald die Verbindung steht, lauscht PuTTY auf dem lokalen PC am Port 443 und wartet auf eine Verbindung. Wenn ich jetzt auf meinem PC einen Browser starte und in der Adressleiste https://localhost eintippe, wird die Anfrage über den SSH-Tunnel an den Port 445 auf dem IPcop weitergeleitet. Hier nimmt der Webserver die Anfrage entgegen und sendet die Startseite des IPcop über den Tunnel zurück.

Der SSH-Tunnel Teil 2

Über einen SSH-Tunnel kann aber nicht nur auf den IPcop zugegriffen werden, sondern man kann über diese Tunnel auch Zugriff auf Server und PCs im LAN oder in der DMZ bekommen. Sehr praktisch, um z. B. einen Server in der DMZ per RDP-Protokoll zu warten, ohne den RDP-Port 3389 auf der Firewall öffnen zu müssen, oder um einen PC im LAN per VNC zu bedienen, ohne den VNC-Port 5900 auf der Firewall zum LAN hin öffnen zu müssen.

Der einzige offene Port auf der Firewall muss für all diese Zwecke der SSH-Port 55555 sein. Alle anderen Dienste können darüber getunnelt werden. So schafft man eine möglichst geringe Angriffsfläche, ohne auf praktische Dienste verzichten zu müssen. Die Sicherheit steht und fällt jedoch mit der Sicherheit des SSH-Zugangs. Wenn diese durch fehlende Patches oder schwache Passwörter unterminiert wird, stellt diese Art des Zugriffs eine gewaltige Gefahr für das gesamte Netzwerk dar.

Um Zugriff auf PCs hinter dem IPcop zu bekommen, muss beim Ipcop v1.3 ein weiterer Eintrag in der Datei /etc/ssh/sshd_config geändert werden:



Der Wert von „AllowTcpForwarding“ muss von „no“ auf „yes“ geändert werden und der SSH-Serverdienst mit „/usr/local/bin/restartssh„ neu gestartet werden.

Beim Ipcop v1.4 geht dies wesentlich komfortabler über das Webinterface. Hier genügt es, auf der SSH-Konfigurationsseite das Häkchen bei „TCP-Weiterleitung zulassen“ zu setzen. SSH1 sollte aus Sicherheitsgründen deaktiviert bleiben.

Von jetzt an können Verbindungen zu beliebigen Servern oder PCs per SSH getunnelt werden.

VNC
Im Falle einer VNC-Verbindung auf einen PC im LAN wird eine PuTTY-Regel nach folgendem Muster benötigt:

Source Port:	5900 (der Standard-Port von VNC)
Destination:	LAN-IP-Adresse des PCs, gefolgt von einem Doppelpunkt und dem Ziel-Port. In unserem Fall z. B. 192.168.100.101:5900

Ein Aufruf des VNC-Viewers mit folgender Einstellung erlaubt dann den Zugriff per VNC auf den PC mit der IP 192.168.100.101.

Um per RDP-Protokoll auf einen W2k/W2k3-Server oder einen PC mit WinXP zuzugreifen, muss die PuTTY-Regel so aussehen:

Source Port: 3389 (der Standard-Port des RDP-Protokolls)

Destination: LAN-IP-Adresse des Servers/PCs, gefolgt von einem Doppelpunkt und dem Ziel-Port. In unserem Fall 192.168.100.50:3389

Ein Aufruf der Remotedesktopverbindung mit folgender Einstellung erlaubt dann den Zugriff per RDP-Protokoll auf den Server/PC mit der IP 192.168.100.50.

Bei der Benutzung von Windows XP als Client, lauert hier jedoch ein Fallstrick. Da Windows XP von Haus aus einen eigenen Terminalserver mitbringt, ist der Port 3389 lokal schon belegt. Die Regel muss daher leicht angepasst werden.

Als „Source Port“ muss hier ein beliebiger freier Port gewählt werden, z. B. 6666. Die Konfiguration der „Destination“ erfolgt dann, wie weiter oben beschrieben.

Ein Aufruf der Remotedesktopverbindung mit folgender Einstellung erlaubt dann den Zugriff von einem Windows XP-PC per RDP-Protokoll auf den Server/PC mit der IP 192.168.100.50.

Diese Liste liesse sich natürlich beliebig verlängern, aber ich denke, dass das Prinzip klar geworden sein sollte. Generell können alle TCP-Ports über einen SSH-Tunnel geschickt werden. Es ist auch möglich, eine PuTTY-Verbindung einzurichten, die mehrere Tunnel (z. B. HTTPS und VNC und RDP) parallel öffnet. Die gleichzeitige Benutzung mehrerer Tunnel parallel führte bei mir jedoch gelegentlich zu Stabilitätsproblemen von PuTTY.

Und jetzt?

- Wie wäre es mit dem Einrichten eines Web- oder Mail-Servers?
- Was ist eine DMZ und wofür brauche ich sie?
- Wie bringe ich meinen Webserver in der DMZ zum Laufen?

Also weiter geht's mit dem nächsten Tutorial.