

VPN, dead peer detection und dynamische DSL-Zugänge

Immer wieder taucht im Zusammenhang dieser drei Begriffe die Frage auf, warum korrekt eingerichtet VPN Tunnel trotzdem nicht ordnungsgemäß wieder aufgebaut werden. Dazu muss man sich mit der Vorgehensweise beim Start von IPsec und der Zusammenarbeit mit dynamischen IP-Adressen, wie sie typischerweise bei Einwahlzugängen vergeben werden, beschäftigen.

Beim Start der VPN-Module wird zunächst einmal überprüft, ob der eigene Name eine IP-Adresse oder ein Domain-Name ist.

Im letzteren Fall wird durch eine DNS-Abfrage die IP-Adresse ermittelt. Ist die eigene (dynamische) IP-Adresse nicht identisch mit der durch Namensauflösung ermittelten IP kann IPsec nicht starten: es kann sich ja niemand zu diesem Host verbinden (ipsec__plutorun 022 "Name": we have no ipsecN interface for either end of this connection) und macht deshalb keinen Sinn. Die Gegenseite eines definierten Tunnels bekommt diese IP-Adresse auch nur durch eine DNS-Abfrage heraus und versucht dann zu einer "falschen" IP-Adresse zu verbinden. Bei einer statisch vergebenen IP-Adresse existiert diese Fehlermöglichkeit schon mal gar nicht.

Im zweiten Schritt werden die in der ipsec.conf hinterlegten Tunnelbeschreibungen ausgelesen und gestartet. Auch hier gilt: statische IP-Adressen der Tunnel-Partner sind unproblematisch. Steht in der ipsec.conf für einen gegebenen Tunnel ein Name in der Form partner.dyndns.org muss auch hier zuerst eine DNS-Abfrage Klarheit über dessen IP-Adresse erbringen.

Wenn bis hierher alles in Ordnung war, wird der Tunnel (einfach gesprochen) in Betrieb genommen und die Aushandlung der Sicherheitsparameter läuft durch - Tunnel steht!

Jetzt kommt die DPD (= dead peer detection) ins Spiel. Erste Voraussetzung ist, dass beide Seiten die DPD unterstützen müssen. Spielt da einer nicht mit, hat die DPD für beide Seiten keinen Wert. Der Tunnel kommt gar nicht erst zustande, wenn nur ein Partner DPD aktiviert hat. Sinn und Zweck der DPD ist, einen Mechanismus bereitzustellen, der Aufschluss darüber gibt, ob der Partner eines VPN-Tunnels noch reagiert und wie im anderen Fall darauf zu reagieren ist. Andere Firewallhersteller kochen manchmal ihr eigenes Süppchen oder loten die Grenzen der DPD-Definition aus, so dass nicht immer eine Einigung zustande kommt. Und manche Firewalls unterstützen auch gar kein DPD. Um das herauszufinden, hilft nur das Lesen der Dokumentation. In solchen Fällen hilft dann nur das Abschalten von DPD auf dem IPCop.

Folgendes Szenario:

Der VPN-Tunnel steht und ist funktionstüchtig, es gehen aber momentan keine Pakete über diese Verbindung. Zur Überprüfung, ob der Partner noch reagiert sind folgende Werte in der ipsec.conf pro Verbindung definierbar, z.B.

- a) dpddelay=30
- b) dpdtimeout=120
- c) dpdaction=hold

dpddelay=30 bedeutet, dass alle 30 Sekunden ein Paket zum Partner geschickt wird (R_U_THERE, was frei übersetzt "bist du noch da" bedeutet). Normalerweise schickt der

Partner daraufhin ein Paket zurück (R_U_THERE_ACK, soll heissen: "ja, ich bin da"). Dann ist alles gut. Wenn aber die Antwort unterbleibt, wartet unsere Seite max. 120 Sekunden (=dpdtimeout) ob doch noch eine Bestätigung erfolgt. Wenn nicht, wird der Tunnel als abgebrochen betrachtet und alle damit zusammenhängenden Parameter (z.B. eroute, SA, etc.) verworfen (dpdaction=clear). Das ist sinnvoll wenn es sich z.B. um einen Roadwarrior handelt, bei dem die DSL-Verbindung abgebrochen ist. Geht man von der Annahme aus, dass die andere Seite sich nur temporär nicht meldet, setzt man dpdaction=hold. Das wiederum macht nur Sinn, wenn der Partner eine statische IP-Adresse besitzt. Eine dynamisch zugeordnete IP-Adresse des Partners wird sich mit hoher Wahrscheinlichkeit nach einer Neueinwahl geändert haben. Deshalb sollte für dynamische Adressen der Eintrag dpdaction=restart drinstehen.

Spätestens jetzt wird klar, dass jeder Partner eines Tunnels derselben Problematik unterworfen ist. Dreh- und Angelpunkt ist die Namensauflösung beim Start der IPsec-Module. Hat sich im Betrieb die IP-Adresse eines Partners geändert, merkt IPsec davon zuerst mal nichts. Ausgenommen sind statische IP-Adressen. Aber die sind ja immer gleich und machen deshalb auch kein Problem.

Mein Lösungsansatz:

1.) Es muß ein Skript her, dass in regelmässigen Abständen überprüft, ob sich eine der IP-Adressen der Tunnelpartner geändert hat. Ein solches Skript findet Ihr unter

<http://www.compass-host.de/ipcop/vpn-watch.sh>

Funktionsweise des Skripts:

Zuerst werden die Tunnelinformationen ausgelesen (es werden nur Net-To-Net Verbindungen überwacht. Roadwarrior werden meiner Erfahrung nach zuverlässig über die DPD abgehandelt).

Für jede Verbindung ruft sich das Skript selbst nochmals auf und arbeitet in einer Endlosschleife alle 2 Minuten die Überprüfung der IP-Adressen ab. Wird dabei eine geänderte IP-Adresse entdeckt (egal ob die eigene oder die des Partners), erfolgt ein Neustart der betreffenden Verbindung (es erfolgt keine vollständige Neuinitialisierung des VPN! Nur die betreffende Verbindung wird neu gestartet). Im Protokoll erfolgt bei Entdeckung einer geänderten IP-Adresse ein Eintrag.

Wer möchte, kann selbstverständlich die Variable CHECK_INTERVAL in Zeile 45 im Skript anpassen. Der Wert

```
CHECK_INTERVAL='120'
```

definiert, dass nach jeder Schleife 120 Sekunden bis zur nächsten Überprüfung gewartet wird. Wer den Wert z.B. auf 60 Sekunden reduziert, lässt die Verbindungen einmal pro Minute überprüfen. Da es nach einem Leitungsabbruch bei einem Partner 2-3 Minuten dauert, ehe der wieder online ist und seine dynamische IP-Adresse aktualisiert hat, ist 2 Minuten meines Erachtens ein guter Wert.

Um das Protokoll nicht übermäßig zu füllen, ist im Skript in der Zeile 207 die Ausgabe auskommentiert:

```
#          log "IP used in tunnel = $TUNIP, IP by DNS = $REMOTEIP"
```

Mehr Infos im Protokoll bekommt derjenige der das erste Zeichen in dieser Zeile entfernt. Mit jedem Durchlauf des Skripts erfolgt dann einen Eintrag im Protokoll.

Installation des Skripts:

- Kopiere die Skript-Datei nach /usr/local/bin
- Skript als ausführbar kennzeichnen ("chmod +755 vpn-watch.sh")

Starten des Skripts:

```
"/usr/local/bin/vpn-watch.sh start"
```

Für jede gestarte Verbindungsüberwachung wird einmal der Buchstabe "S" ausgegeben.

Stoppen des Skripts:

```
"/usr/local/bin/vpn-watch.sh stop"
```

Für jede (beendete) Verbindungsüberwachung wird einmal der Buchstabe "T" ausgegeben.

Statusinfo mit

```
"/usr/local/bin/vpn-watch.sh status"
```

Für jede Verbindung wird die Zahl der Neustarts ausgegeben.

2.) Aufruf des Skripts bei Neustart des IPCops

Dazu die Datei rc.local im Verzeichnis /etc/rc.d mit einem Editor aufrufen und die Zeile

```
/usr/local/bin/vpn-watch.sh start
```

ergänzen.

3.) Da die übliche Zwangstrennung einer DSL-Verbindung i.d.R. nicht dann erfolgt, wenn wir es uns wünschen, machen wir aus der Not eine Tugend und trennen die Leitung unter unserer Kontrolle. Dazu teilen wir dem System in der cron-Tabelle mit, dass

- um 03:00 Uhr nachts das Skript beendet wird
- um 03:01 Uhr nachts die Leitung getrennt wird
- um 03:02 Uhr nachts die Leitung neu verbunden werden soll
- um 03:10 Uhr nachts das Skript wieder gestartet wird

Die Uhrzeit hierfür setzt jeder nach eigener Wahl. Im Wissen um die Verzögerung beim Aktualisieren der dynamischen IP-Adresse empfiehlt es sich bei den Einstellungen zum VPN eine Verzögerung im Bereich von mind. 2 Minuten zu wählen. Dann sollte der DYNDNS-Eintrag aktualisiert worden sein.

Aufruf der cron-Table zum Editieren:

```
"fcrontab -e"
```

und Eintragen der Zeilen:

```
00 03 * * * /usr/local/bin/vpn-watch.sh stop
01 03 * * * /etc/rc.d/rc.red stop
02 03 * * * /etc/rc.d/rc.red start
10 03 * * * /usr/local/bin/vpn-watch.sh start
30 03 * * * /etc/rc.d/ipsec restart
```

Die letzte Zeile startet alle IPSec-Module um 03:30 Uhr neu. Leider kommt es immer wieder vor, dass ein Teil der Verbindungen erst nach einem Neustart aller IPSec-Module (auf beiden Cops) wieder vollständig funktioniert. Teilweise wird auf der einen Seite die VPN-Verbindung als vollständig und funktional geführt, während die andere Seite „behauptet“, dieser Tunnel wäre „Beendet“. Als pragmatische Abhilfe wird deshalb das IPSec komplett neu gestartet.

Ich habe auf den von mir betreuten Cops diese Einstellungen auf allen Cops durchgeführt. Auch nach Tagen und Wochen stehen die VPN-Tunnel nach spätestens 2 Minuten wieder zur Verfügung ohne dass ein manueller Eingriff notwendig ist.

4.) Hilfreich, aber nicht notwendig ist eine Ergänzung des Web-GUI im IPCop, die es erlaubt im Protokoll die Eintragungen herauszufiltern, die nur das Skript betreffen. So behält man die Übersicht welche Verbindung wann neu gestartet wurde. Dazu holt man sich die Datei log.dat aus dem Verzeichnis /home/httpd/cgi-bin/logs.cgi und ergänzt wie folgt:

Der Bereich (ab Zeile 50)

```
my %sections = (
    'ipcop' => '(ipcop)',
    'red' => '(red.*|kernel:
usb.*|pppd\[.*\]|chat\[.*\]|pppoe\[.*\]|pptp\[.*\]|pppoa\[.*\]|pppoa3\[.*\]
|pppoeci\[.*\]|ipppd|ipppd\[.*\]|kernel: ippp\d|kernel:
isdn.*|ibod\[.*\]|kernel: eth.*|dhcpcd\[.*\]|modem_run.*)',
    'dns' => '(dnsmasq)\[.*\]',
    'dhcp' => '(dhcpd)',
    'cron' => '(fcron)\[.*\]',
    'ntp' => '(ntpd|ntpdate)\[.*\]',
    'ssh' => '(sshd(?:\[.*\])?\[.*\])',
    'auth' => '(\\w+(pam_unix)\[.*\])',
    'kernel' => '(kernel)',
    'ipsec' => '(ipsec_\\w_|pluto)\[.*\]',
    'snort' => '(snort)',
    'installpackage' => '(installpackage)\[.*\]'
);
```

wird zu

```

my %sections = (
    'ipcop' => '(ipcop)',
    'red' => '(red.*|kernel:
usb.*|pppd\[.*\]|chat\[.*\]|pppoe\[.*\]|pptp\[.*\]|pppoa\[.*\]|pppoa3\[.*\]
|pppoeci\[.*\]|pppd|pppd\[.*\]|kernel: ippp\d|kernel:
isdn.*|ibod\[.*\]|kernel: eth.*|dhcpcd\[.*\]|modem_run.*)',
    'dns' => '(dnsmasq)\[.*\]',
    'dhcp' => '(dhcpd)',
    'cron' => '(fcron)\[.*\]',
    'ntp' => '(ntpd|ntpdate)\[.*\]',
    'ssh' => '(sshd(?:\[.*\])?\[.*\])',
    'auth' => '(\w+(pam_unix)\[.*\])',
    'kernel' => '(kernel)',
    'ipsec' => '(ipsec_\w_|pluto\[.*\])',
    'snort' => '(snort)',
    'vpn-watch' => '(vpn-watch 1.6.3)',
    'installpackage' => '(installpackage)\[.*\]'
);

```

Und gleich danach (ab Zeile 67)

```

my %trsections = (
    'ipcop' => 'IPCop',
    'red' => 'RED',
    'dns' => 'DNS',
    'dhcp' => "$Lang::tr{'dhcp server'}",
    'cron' => 'Cron',
    'ntp' => 'NTP',
    'ssh' => 'SSH',
    'auth' => "$Lang::tr{'loginlogout'}",
    'kernel' => "$Lang::tr{'kernel'}",
    'ipsec' => 'IPSec',
    'snort' => 'Snort',
    'installpackage' => "$Lang::tr{'update transcript'}"
);

```

so ergänzen:

```

my %trsections = (
    'ipcop' => 'IPCop',
    'red' => 'RED',
    'dns' => 'DNS',
    'dhcp' => "$Lang::tr{'dhcp server'}",
    'cron' => 'Cron',
    'ntp' => 'NTP',
    'ssh' => 'SSH',
    'auth' => "$Lang::tr{'loginlogout'}",
    'kernel' => "$Lang::tr{'kernel'}",
    'ipsec' => 'IPSec',
    'snort' => 'Snort',
    'vpn-watch' => 'VPN-Watch',
    'installpackage' => "$Lang::tr{'update transcript'}"
);

```

Damit gibt es im Web-GUI des Cops im Menü LOGs->System-Logdateien als neue Auswahl in der Drop-Down-Liste den Eintrag VPN-Watch. Hier noch ein kleines Beispiel für die dortige Ausgabe:

```
07:57:27 vpn-watch 1.6.3 (Tunnel1) IP used in tunnel = 84.57.131.177, IP by
DNS = 84.57.131.177
07:56:52 vpn-watch 1.6.3 (Tunnel2) left or right IP has changed: restarting
connection...
07:56:52 vpn-watch 1.6.3 (Tunnel2) Left IP new = 84.161.65.157, right IP
new = 84.182.135.154
07:56:52 vpn-watch 1.6.3 (Tunnel2) Left IP old = 84.161.65.157, right IP
old = 84.182.167.185
```

Letzte Änderung am:

25-04-2006:

- Prüfung ob Skript bereits gestartet
- Prüfung, ob stat. IPs verwendet werden; dann keine Namensauflösung per DNS durchführen
- Protokollmeldungen überarbeitet

 Zeile 207: protokolliert ermittelte IPs

 Zeilen 264-298: Meldungen abhängig vom Vergleichsergebnis

Im Normalfall sind diese Protokollmeldungen deaktiviert. Durch Entfernen des Kommentarzeichens (#) in der ersten Spalte werden diese wieder aktiviert. Je nachdem wie viele Tunnel überprüft werden, kann das zu einer signifikanten Vergrößerung der Protokolldatei /var/log/messages führen.