VPNs

Σενάρια χρήσης Εικονικών Ιδιωτικών Δικτύων (VPNs) που βασίζονται στο πρωτόκολλο SSL

Παναγιώτης Πλέσσας
ΜΕ / 07072

Τμήμα Ψηφιακών Συστημάτων
Πανεπιστήμιο Πειραιώς

Περιεχόμενα

1. Εισαγωγή

Το Internet αποτελεί αναμφισβήτητα ένα από τα μεγαλύτερα τεχνολογικά επιτεύγματα του αιώνα. Ξεκινώντας ως ένα απλό δίκτυο που συνέδεε υπολογιστές κρατικών Υπηρεσιών ή Πανεπιστημίων στις Ηνωμένες Πολιτείες τώρα πλέον είναι το μεγαλύτερο δίκτυο πληροφοριών, διασκέδασης και επικοινωνίας του πλανήτη. Τα τελευταία χρόνια εξελίσσεται και μεταλλάσσεται σε χώρο εμπορικής δραστηριότητας με τους δικούς του νόμους, περιορισμούς και προϋποθέσεις.

Το Internet στο χώρο αυτό επεκτείνεται σε δύο κυρίως επίπεδα: το δημόσιο (public level) και το ιδιωτικό (private level). Το πρώτο είναι εδώ και χρόνια σε ανάπτυξη και αφορά κυρίως εφαρμογές ηλεκτρονικού εμπορίου (e-commerce) δηλ. την παροχή υπηρεσιών ή την πώληση αγαθών. Το δεύτερο αναπτύσσεται ραγδαία τελευταίως και έχει να κάνει με τη χρήση του Διαδικτύου από μεγάλες επιχειρήσεις σαν μέσο μετάδοσης των στοιχείων και πληροφοριών τους (data) που είναι απαραίτητο να μεταφερθούν γρήγορα, σίγουρα και χωρίς υποκλοπές.

1.1. Ορισμός

Η παγκοσμιοποίηση της αγοράς και η εξέλιξη της τεχνολογίας ανάγκασε πολλές επιχειρήσεις να αλλάξουν τον τρόπο εργασίας τους. Οι περισσότερες επιχειρήσεις σήμερα έχουν ως κύριο μέλημα την γεωγραφική τους επέκταση και σε άλλες περιοχές πέρα από την έδρα τους. Οι επιχειρήσεις αυτές πρέπει πλέον να διατηρούν υποκαταστήματα σε πολλά σημεία της ίδιας χώρας ή ακόμα και του εξωτερικού, να έχουν εργαζόμενους που ταξιδεύουν, να μοιράζουν στοιχεία τους σε πελάτες και προμηθευτές.

Μέχρι και αρκετά πρόσφατα αυτό είχε ως επακόλουθο την χρήση μισθωμένων γραμμών (leased lines) με σκοπό την δημιουργία WAN (wide

area network). Αυτές οι μισθωμένες γραμμές είχαν εύρος (bandwidth) από απλή ISDN (**I**ntegrated **S**ervices **D**igital **N**etwork, 128 Kbps) ως και OC3 (**O**ptical **C**arrier-3, 155 Mbps) και παρείχαν στις εταιρίες την δυνατότητα να μεγαλώσουν το ιδιωτικό δίκτυό τους πέρα από μία μέση μικρή γεωγραφική περιοχή. Ένα WAN έχει προφανή πλεονεκτήματα, εν συγκρίσει με ένα δημόσιο δίκτυο, όπως το Internet, όσον αφορά την αποτελεσματικότητα, την ασφάλεια, την εγκυρότητα και τις επιδόσεις. Αλλά η διατήρηση ενός WAN, ιδιαίτερα όταν χρησιμοποιούνται μισθωμένες γραμμές, αποτελεί μεγάλο έξοδο, το οποίο σταδιακά αυξάνεται όσο μεγαλώνει η απόσταση των γραφείων της επιχείρησης. Έτσι το κόστος για τη συντήρηση τέτοιων γραμμών ήταν υπερβολικά υψηλό, το δίκτυο φορτωνόταν πάρα πολύ και επίσης το πρόβλημα της αποκοπής και μη λειτουργίας των υποκαταστημάτων αν κάτι συνέβαινε στα κεντρικά, πάντα υπήρχε.

Καθώς η δημοτικότητα του διαδικτύου μεγάλωνε οι εταιρίες αναζητούσαν να πετύχουν γρήγορο, ασφαλή και έγκυρη επικοινωνία μεταξύ των γραφείων τους σε οποιοδήποτε σημείο και αν βρίσκονται αυτά. Για να καλύψουν λοιπόν τις ανάγκες των εργαζομένων τους για επικοινωνία, αρκετές επιχειρήσεις άρχισαν να δημιουργούν τα δικά τους εικονικά ιδιωτικά δίκτυα, τα **Virtual Private Networks ,VPN** για να προσαρμοστούν στις ανάγκες των απομακρυσμένων υπαλλήλων και γραφείων.

Θα μπορούσαμε να ορίσουμε ένα Virtual Private Network ως εξής

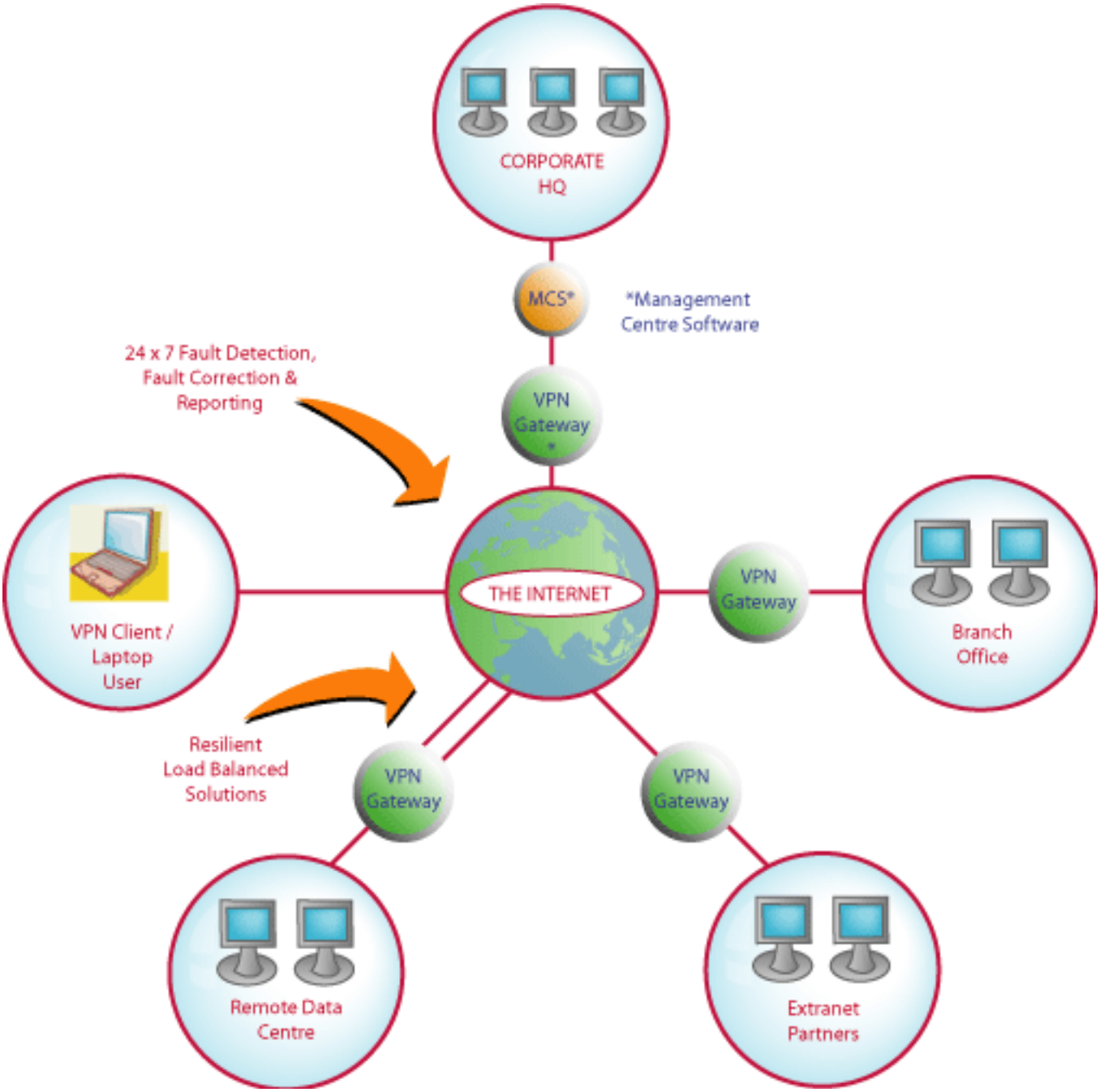
*Ένα **Virtual Private Network (VPN)** είναι ένα προσωπικό δίκτυο, το οποίο χρησιμοποιεί ένα πιο ευρύ (δημόσιο) δίκτυο, όπως είναι το Internet, προκειμένου να επικοινωνεί με άλλα sites ή απομακρυσμένα δίκτυα. Αντί της χρήσης μίας μοναδικά αφιερωμένης γι' αυτό τον σκοπό, πραγματικής σύνδεσης, όπως η μισθωμένη γραμμή (leased line), το VPN χρησιμοποιεί εικονικές συνδέσεις δρομολογημένες μέσω του διαδικτύου, από το ιδιωτικό δίκτυο της εταιρίας προς την απομακρυσμένη σελίδα ή εργαζόμενο.*

Όπως παρατηρούμε η φράση ***Virtual Private Network*** απαρτίζεται από τους όρους virtual, private, network. Η σημασιολογία τους φαίνεται παρακάτω.

Virtual: Ο όρος virtual σημαίνει εικονικό δηλαδή κάτι μη πραγματικό. Έχει δοθεί αυτός ο όρος επειδή αντίθετα με τις ευθείες γραμμές όπου χρησιμοποιούνται μόνιμες συνδέσεις μεταξύ των σημείων, εδώ η σύνδεση δημιουργείται μόνο για το χρόνο που απαιτείται για την εκτέλεση της εργασίας και κατόπιν διακόπτεται αφήνοντας το δίκτυο και τον εξοπλισμό ελεύθερο για άλλη χρήση. Επίσης ο όρος σημαίνει λογική και όχι φυσική δομή όπως για παράδειγμα στα LANs. Το δίκτυο υφίσταται, μεταβάλλεται, τροποποιείται ανάλογα με το σημείο και το χρόνο που γίνεται η σύνδεση χρησιμοποιώντας εξωτερικό εξοπλισμό (του ISP) και όχι κατά ανάγκη της ίδιας της εταιρίας.

Private: Ο όρος «Private» σημαίνει ότι δημιουργείται μια προσωπική-ιδιωτική σύνδεση μεταξύ δύο σημείων παρ'όλο που χρησιμοποιείται το κοινό τηλεφωνικό δίκτυο ή που συνταξιδεύουν παράλληλα και άλλα δεδομένα. Επίσης σημαίνει ασφάλεια και προστασία από κάθε λογής υποκλοπή αφού όλα τα δεδομένα θεωρούνται σημαντικά και απόρρητα.

Network: Ένα δίκτυο αποτελείται από δύο ή περισσότερες συσκευές που μπορούν ελεύθερα και ηλεκτρονικά να επικοινωνήσουν η μια με την άλλη μέσω των καλωδίων ή ασύρματα. Ένα VPN είναι ένα δίκτυο. Μπορεί να διαβιβάσει πληροφορίες πέρα από μεγάλες αποστάσεις αποτελεσματικά και αποδοτικά.


CORPORATE HQ
MCS*
*Management Centre Software
24 x 7 Fault Detection, Fault Correction & Reporting
VPN Gateway
THE INTERNET
VPN Client / Laptop User
VPN Gateway
Branch Office
Resilient Load Balanced Solutions
VPN Gateway
VPN Gateway
Remote Data Centre
Extranet Partners

1.2. Κύρια σημεία και πλεονεκτήματα των VPNs

Τα στοιχεία που θα πρέπει απαραίτητα να ενσωματώνει ένα καλοσχεδιασμένο VPN συνοψίζονται ως εξής:

• **Χαμηλό κόστος :** Οι μισθωμένες γραμμές (leased lines) T1 (1.5 Mbps) και T3 (45 Mbps) απαιτούν μεγάλο μηνιαίο πάγιο και χρέωση ανάλογα με την απόσταση των συνδεδεμένων σημείων καθώς επίσης απαιτούνται επιπλέον χρήματα για κάθε Permanent Virtual Circuit (PVC) που δημιουργείται. Ειδικά στην περίπτωση που η απόσταση είναι μεγάλη το κόστος είναι απαγορευτικό. Αντίθετα γραμμές με τις ίδιες ταχύτητες σε τοπικό παροχέα Internet (ISP) στοιχίζουν πολύ λιγότερο ή μπορούν και να αποφθεχθούν αφού η διασύνδεση μπορεί να γίνει από παντού με μια απλή σύνδεση και όλα τα πλεονεκτήματα των ανωτέρω.

• **Ευκαμψία (Flexibility) :** Στα παραδοσιακά δίκτυα έπρεπε να υπάρχει συμβατός εξοπλισμός που να υποστηρίζει όλα τα περιφερειακά γραφεία ή τους απομακρυσμένους κλάδους της επιχειρήσεις . Στα VPNs δεν υπάρχει περιορισμός ή προβλήματα ασυμβατότητας εξοπλισμού αφού απλά και μόνο η σύνδεση με έναν ISP αρκεί για την επικοινωνία.

• **Επεκτασιμότητα (scalability) :** α) Η χρήση του Internet ως μέσο μετάδοσης προσφέρει απεριόριστη γεωγραφική επέκταση. Πολύ εύκολα από οποιοδήποτε μέρος του κόσμου ανά πάσα στιγμή πελάτες, προμηθευτές ή άνθρωποι της επιχείρησης συνδέονται μεταξύ τους. β) Οι συνδέσεις αυτές είναι εύκολα αναβαθμίσιμες ανάλογα με τις απαιτήσεις χωρίς όμως και υποχρεωτική αναβάθμιση του εξοπλισμού σε κάθε σημείο (point) αφού αλλάζει μόνο το είδος της σύνδεσης με τον ISP.

- **Ασφάλεια (Security) :** Ένα από τα πιο σημαντικά στοιχεία για την λειτουργία ενός VPN είναι η ασφάλεια του. Τα VPN παρέχουν αυξημένη ασφάλεια λόγω των πρωτοκόλλων tunneling και ασφάλειας που χρησιμοποιούνται. Για τη διατήρηση της ασφάλειας κατά την πρόσβαση των χρηστών στο VPN, χρησιμοποιούνται γνωστά μέσα, όπως Firewalls και κρυπτογράφηση δεδομένων, καθώς και άλλα μέσα, όπως για παράδειγμα το πρωτόκολλο IPSec και AAA Servers.

Το πρωτόκολλο **IPSec** [**I**nternet **P**rotocol **S**ecurity] παρέχει βελτιωμένες μεθόδους ασφάλειας, όπως για παράδειγμα καλύτερους αλγορίθμους κρυπτογράφησης και πιο εύχρηστη πιστοποίηση χρηστών. Μόνο τα συστήματα τα οποία υποστηρίζουν το πρωτόκολλο IPSec μπορούν να εκμεταλλευτούν τις λειτουργίες του, ενώ απαιτείται και η χρήση κοινών firewalls και ρυθμίσεων ασφάλειας δικτύου. Το πρωτόκολλο IPSec μπορεί να κρυπτογραφήσει δεδομένα μεταξύ διαφόρων τερματικών.

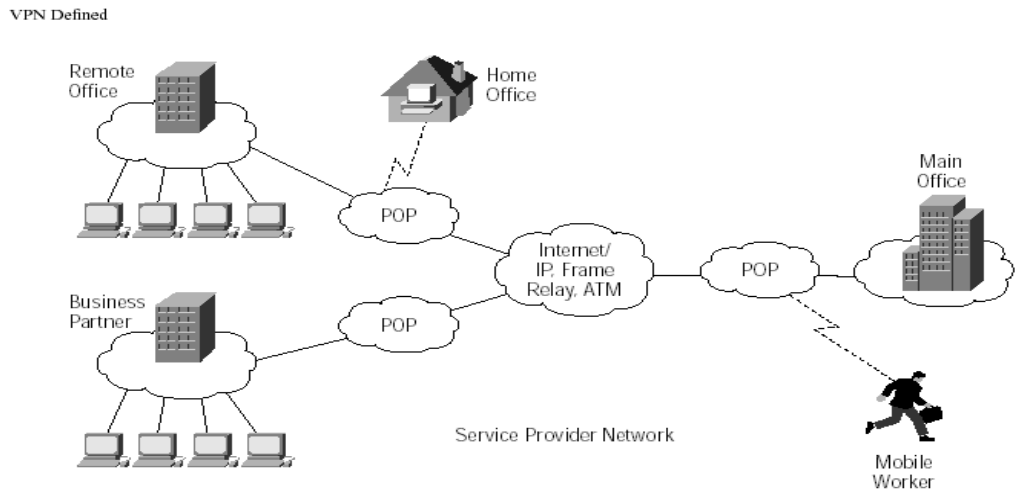
Τέλος οι AAA Servers είναι servers οι οποίοι παρέχουν επιπλέον προστασία κατά τη σύνδεση εργαζομένων σε κάποιο VPN. Μόλις κάποιος χρήστης συνδεθεί σε ένα VPN για να ανοίξει κάποια **session**, δημιουργείται μια «αίτηση» η οποία ελέγχει: ποιος είναι ο χρήστης (**Authentication**), τι πρόσβαση έχει (**Authorization**) και τέλος τι λειτουργίες πραγματοποιεί (**Accounting**). Όλα τα παραπάνω αναλύονται παρακάτω.

- **Διαχείριση (Management) :** Εύκολη και συγκεντρωτική διαχείριση του δικτύου διότι από ένα σημείο ελέγχονται IP addressing, πολιτικές πρόσβασης χρηστών, ασφάλεια και άλλες συναφείς εργασίες.

1.3. Τεχνολογίες VPN

Υπάρχουν διάφορες τεχνολογίες VPN οι οποίες θα μελετηθούν και θα εξεταστούν στα πλαίσια της πτυχιακής. Αναφέρω επιγραμματικά τις παρακάτω:

- IPsec με κρυπτογράφηση. Το IPSec - Internet Protocol Security είναι ένα σύνολο από πρωτόκολλα για την ασφάλεια επικοινωνίας με το πρωτόκολλο IP το οποίο κρυπτογραφεί και πιστοποιεί κάθε πακέτο IP σε μια ροή δεδομένων. Η κρυπτογράφηση γίνεται είτε με πιστοποιητικά είτε με κάποιο κοινό κωδικό. Το IPsec περιγραφεται σε πολλα RFCs, 2401, 2406, 2407, 2408, και 2409.
- IPsec πάνω από L2TP (που περιγράφεται στο RFC 3193). Το Layer 2 Tunneling Protocol (L2TP) ένα πρωτόκολλο δημιουργίας σήραγγας – tunnel για την υποστήριξη εικονικών δικτύων. Δε προσφέρει ασφάλεια και κρυπτογράφηση από μόνο του και βασίζεται σε άλλα πρωτόκολλα κρυπτογράφησης (IPSec).
- Εικονικά δίκτυα SSL 3.0 ή TLS. Κλασικό παράδειγμα τέτοιου συστήματος είναι το OPENVPN.



Σχήμα 1.1: Τοπολογία VPN

1.4. Υπηρεσίες VPN

Τα δίκτυα λειτουργούν είτε στο στρώμα-2 είτε στο στρώμα-3 του προτύπου διασύνδεσης ανοικτών συστημάτων (OSI).

1.4.1 Τα στρώματα του προτύπου OSI

Τα στρώματα του προτύπου OSI είναι τα ακόλουθα:

Μοντέλο OSI			
	Μονάδα δεδομένων	Επίπεδο	Λειτουργία
Λογισμικό	Δεδομένα	7. Εφαρμογών	Παρέχεται στις εφαρμογές πρόσβαση στο δίκτυο
		6. Παρουσίασης	Αναπαράσταση δεδομένων και κρυπτογράφηση
		5. Συνόδου	Έλεγχος του διαλόγου μεταξύ των άκρων της επικοινωνίας
	Πακέτο	4. Μεταφοράς	Αξιόπιστη επικοινωνία από άκρο σε άκρο
Υλικό	Πακέτο	3. Δικτύου	Καθορισμός διαδρομών και λογικών διευθύνσεων των κόμβων στα πλαίσια ενός διαδικτύου
	Πλαίσιο	2. Ζεύξης δεδομένων	Φυσική διευθυνσιοδότηση (MAC & LLC)
	Bit	1. Φυσικό	Δυαδική μετάδοση σήματος μέσω του φυσικού μέσου

Φυσικό επίπεδο (Physical Layer) – Στρώμα 1ο

Το φυσικό επίπεδο καθορίζει όλες εκείνες τις φυσικές, ηλεκτρικές και λειτουργικές δυνατότητες μιας διασύνδεσης. Στο φυσικό επίπεδο ανήκουν τα μέσα μετάδοσης όπως οι οπτικές ίνες, τα συνεστραμμένα ζεύγη ή οι ασύρματες ζεύξεις. Οι δυνατότητες του φυσικού επιπέδου ρυθμίζουν συχνά και τις επιλογές που κάνουν οι σχεδιαστές συστημάτων στα υπερκείμενα επίπεδα. Από τη μορφή τους σε φυσικό επίπεδο τα δίκτυα χαρακτηρίζονται σε τοπικά (Local Area Network), μητροπολιτικά (Metropolitan Area Network) και ευρείας περιοχής (Wide Area Network). Το φυσικό ασχολείται λοιπόν με τη μετάδοση ακατέργαστων bits σε ένα κανάλι επικοινωνίας . Τα θέματα σχεδίασης έχουν να κάνουν με τη διασφάλιση ότι , όταν η μία πλευρά στέλνει ένα bit 1 , αυτό λαμβάνεται από την άλλη πλευρά ως bit 1 και όχι ως bit 0 . Τα θέματα σχεδίασης εδώ , στην πλειοψηφία τους ασχολούνται με μηχανικές , ηλεκτρικές και διαδικασιακές διασυνδέσεις καθώς και με το φυσικό μέσο μετάδοσης , το οποίο βρίσκεται κάτω από φυσικό επίπεδο .

Τα χαρακτηριστικά του layer - 1 είναι λοιπόν

- Επίπεδο μεταγωγής, ρυθμός μετάδοσης
- Διπλής κατεύθυνσης
- Καθορίζει τον τύπο των μέσων μεταφοράς και του υλικού διασύνδεσης

Στην περίπτωση του Ethernet τα στοιχεία του φυσικού επιπέδου μπορούν να είναι:

- 10BaseT (UTP, 10Base2 (thinnet), 10Base5 (thicknet), 10BaseF (fiber)
- 10, 100Mbps

Στην περίπτωση του ATM τα στοιχεία του φυσικού επιπέδου μπορούν να είναι:

- Fiber, SONET, Cat 5 UTP
- 45, 100, 155, 622Mbps; 2.5Gbps

Επίπεδο Σύνδεσης Δεδομένων (Data Link Layer) – Στρώμα 2ο

Το επίπεδο σύνδεσης δεδομένων είναι υπεύθυνο για την αξιοπιστία στη μεταφορά της πληροφορίας μέσω του φυσικού καναλιού. Στο επίπεδο αυτό γίνεται η διαμόρφωση των δεδομένων σε ομάδες ανάλογα με το μέσο που χρησιμοποιείται και το πρωτόκολλο που υιοθετείται. Παρέχεται επίσης ένα επιπλέον επίπεδο ελέγχου σφαλμάτων διαχείρισης της διασύνδεσης μεταξύ των υπολογιστικών συστημάτων και του ελέγχου ροής της πληροφορίας.

Η κύρια αποστολή του επιπέδου σύνδεσης δεδομένων (data link layer) είναι να μετασχηματίσει το ακατέργαστο μέσο μετάδοσης σε μια γραμμή που εμφανίζεται ελεύθερη από σφάλματα μετάδοσης στο επίπεδο δικτύου. Ο σκοπός αυτός επιτυγχάνεται με τη διάσπαση των δεδομένων εισόδου από τον αποστολέα σε πλαίσια δεδομένων (data frames) μετάδοση αυτών με μια σειρά & έπειτα επεξεργασία των πλαισίων επιβεβαίωσης λήψης (acknowledgement frames) που επιστρέφονται από τον αποδέκτη. Εφόσον το φυσικό επίπεδο απλώς αποδέχεται και μεταδίδει ένα συρμό bits, χωρίς να νοιάζεται για το νόημα και τη δομή, η δημιουργία και η αναγνώριση των ορίων των πλαισίων εξαρτάται πλέον από το επίπεδο σύνδεσης δεδομένων. Αυτή μπορεί να επιτευχθεί με την επισύναψη ειδικών ακολουθιών bits στην αρχή και στο τέλος των πλαισίων.

Στο επίπεδο αυτό λοιπόν υιοθετείται η φυσική διεύθυνση σε αντιδιαστολή με το επίπεδο δικτύου που τα δεδομένα δρομολογούνται προς τη λογική διεύθυνση. Η φυσική διεύθυνση προσδιορίζεται ως ένα μοναδικό αναγνωριστικό που φέρει η κάρτα δικτύου του υπολογιστικού συστήματος.

Τα χαρακτηριστικά του layer - 2 είναι λοιπόν

- Κατάτμηση των ψηφιο-πακέτων (packet streams) σε πλαίσια πληροφορίας (frames)
- Έλεγχος λάθους μέσω του αθροίσματος των ψηφίων που διαβιβάζονται (Checksum)
- Εγγυάται την παράδοση των πλαισίων πληροφορίας (frames)

Επίπεδο Δικτύου (Network Layer) – Στρώμα 3ο

Το επίπεδο δικτύου παρέχει τη δρομολόγηση της πληροφορίας μεταξύ των διασυνδεδεμένων υπολογιστικών συστημάτων. Τα πρωτόκολλα αυτού του επιπέδου επιτρέπουν την ταυτόχρονη ανταλλαγή πληροφορίας ανάμεσα σε πολλαπλές υπολογιστικές μονάδες. Αυτό επιτυγχάνεται με την επιπλέον πολυπλεξία της πληροφορίας και την υιοθέτηση λογικών διευθύνσεων στη θέση των φυσικών διευθύνσεων που επιβάλλει το δικτυακό υλικό των υπολογιστικών συστημάτων. Τα πρωτόκολλα που συναντώνται στο επίπεδο δικτύου είναι κυρίως δρομολόγησης αλλά και εξακρίβωσης των διευθύνσεων των υπολογιστικών συστημάτων που συμμετέχουν σε μία ανταλλαγή. Έτσι λοιπόν εάν στο υποδίκτυο είναι παρόντα πολλά πακέτα την ίδια χρονική στιγμή, θα εμπλακεί το ένα στην διαδρομή του άλλου, δημιουργώντας συμφόρηση (bottleneck). Ο έλεγχος μιας τέτοιας συμφόρησης επίσης ανήκει στις αρμοδιότητες του επιπέδου δικτύου .

Παραδείγματα τέτοιων πρωτοκόλλων είναι το Internet routing protocol (IP), Open Shortest Path First (OSPF) που δημιουργήθηκε για τη διασύνδεση μεταξύ κόμβων στο Διαδίκτυο και το Address Resolution Protocol (ARP) που χρησιμοποιείται στην αντιστοίχιση των λογικών διευθύνσεων (IP-Address) με φυσικές (MAC-Address).

Τα χαρακτηριστικά του layer - 3 είναι λοιπόν

- Δρομολογεί επαναληπτικά δεδομένα
- Με απευθείας σύνδεση των συστημάτων: X25
- Χωρίς απευθείας σύνδεση των συστημάτων: IP

Επίπεδο Μεταφοράς (Transport Layer) – Στρώμα 4ο

Το επίπεδο μεταφοράς είναι υπεύθυνο για την αξιόπιστη μεταφορά των δεδομένων, καθώς ευθύνεται για τον έλεγχο ροής της πληροφορίας ώστε να μην παρατηρούνται φαινόμενα υπερφόρτωσης σε κάποιο από τα συστήματα που συμμετέχουν, την πολυπλεξία στην ανταλλαγή δεδομένων διαφορετικών εφαρμογών μέσα από το ίδιο μέσο, τη δημιουργία εικονικών συνδέσεων σημείου προς σημείο (virtual-circuit) και το συγχρονισμό στη λήψη των δεδομένων ώστε να επιτυγχάνεται η παρουσία της πληροφορίας με τη σωστή σειρά. Ακόμη, ενσωματώνει μηχανισμούς ελέγχου σφαλμάτων της πληροφορίας και μηχανισμούς επανεκπομπής της με σκοπό τη σωστή μεταφορά της.

Έτσι λοιπόν εάν η σύνδεση μεταφοράς απαιτεί υψηλό ρυθμό εξυπηρέτησης (throughput) , το επίπεδο μεταφοράς μπορεί να δημιουργήσει πολλαπλές συνδέσεις δικτύου , μοιράζοντας τα δεδομένα ανάμεσα στις συνδέσεις δικτύου για να βελτιώσει το ρυθμό εξυπηρέτησης . Από την άλλη πλευρά εάν η δημιουργία ή η συντήρηση μιας σύνδεσης δικτύου είναι ακριβή , το επίπεδο μεταφοράς μπορεί να πολυπλέκει πολλές συνδέσεις μεταφοράς στην ίδια σύνδεση δικτύου για να ελαττώσει το κόστος . Σε όλες τις περιπτώσεις το επίπεδο μεταφοράς χρειάζεται πάντα για να κάνει την πολυπλεξία διάφανη στο επίπεδο συνόδου. Ορισμένα από τα πρωτόκολλα που υλοποιούν το επίπεδο μεταφοράς είναι το Transmission Control Protocol - TCP που αναλαμβάνει την ορθή και συγχρονισμένη ανταλλαγή δεδομένων στα δίκτυα TCP/IP και το Name Binding Protocol - NBP στα AppleTalk δίκτυα.

Τα χαρακτηριστικά του layer - 4 είναι λοιπόν

- Από σημείο σε σημείο αξιοπιστία στην παράδοση δεδομένων.
- Διαχωρισμός της πληροφορίας σε ψηφιο-πακέτα και δημιουργία ομάδας ψηφιο-πακέτων (chunks)
- Σε περίπτωση μη απευθείας διασύνδεση των συστημάτων δημιουργεί συγχρονισμό στην σειρά με την οποία λαμβάνονται τα δεδομένα
- Με απευθείας (εικονική) σύνδεση (σημείο σε σημείο): TCP
  Χωρίς σύνδεση μεταξύ των συστημάτων: UDP

Επίπεδο Συνόδου (Session Layer) – Στρώμα 5ο

Το επίπεδο συνόδου είναι υπεύθυνο για τη δημιουργία, διατήρηση και λήξη συνόδων μεταξύ δύο διασυνδεδεμένων υπολογιστικών συστημάτων. Οι σύνοδοι αυτοί αποτελούνται από απαιτήσεις και ανταποκρίσεις μεταξύ των εφαρμογών των διασυνδεδεμένων συστημάτων. Όταν δύο υπολογιστικά συστήματα επιθυμούν να εκκινήσουν μια σύνοδο (session) για την ανταλλαγή δεδομένων τότε αυτό το επίπεδο αναλαμβάνει την εκκίνηση της συνόδου, την κατεύθυνση και το ρυθμό ροής των δεδομένων όπως επίσης και τον έλεγχο για επανεκπομπή χαμένων ή κατεστραμμένων δεδομένων που παραλήφθηκαν.

Το επίπεδο συνόδου (session layer) επιτρέπει στους χρήστες διαφορετικών μηχανημάτων να εγκαθιστούν συνόδους (sessions) μεταξύ τους. Μία σύνοδος επιτρέπει μια συνήθη μεταφορά δεδομένων, όπως και το επίπεδο μεταφοράς, αλλά παρέχει και μερικές πρόσθετες υπηρεσίες που είναι χρήσιμες σε πολλές εφαρμογές . Μία σύνοδος μπορεί να χρησιμοποιηθεί για να επιτρέψει τη σύνδεση ενός χρήστη σ' ένα απομακρυσμένο σύστημα καταμερισμού χρόνου (time sharing) ή για να μεταφέρει ένα αρχείο μεταξύ δύο μηχανών. Ορισμένα από τα πρωτόκολλα αυτού του επιπέδου είναι το AppleTalk και το Session Control Protocol (SCP).

Τα χαρακτηριστικά του layer - 5 είναι λοιπόν

- Επιπλέον δυνατότητες εκσφαλμάτωσης και αξιοπιστία στη μετάδοση
- Συνήθως δεν είναι ευδιάκριτο, διότι η λειτουργικότητα του μεταβιβάζεται στα γειτονικά του επίπεδα

Επίπεδο Παρουσίασης (Presentation Layer) – Στρώμα 6ο

Το επίπεδο παρουσίασης παρέχει μια σειρά από συναρτήσεις και μηχανισμούς για την κωδικοποίηση και τη μετατροπή των δεδομένων που μετέπειτα θα διαβιβαστούν στο χρήστη μέσω του επιπέδου εφαρμογής και της εφαρμογής που χρησιμοποιεί. Αυτό το επίπεδο εξασφαλίζει ότι η πληροφορία που διαμορφώθηκε σε ένα υπολογιστικό σύστημα θα είναι αναγνώσιμη από κάποιο άλλο που θα την παραλάβει. Η κύρια επομένως λειτουργία του επιπέδου αυτού είναι η ομοιόμορφη, βάσει προτύπων, αναπαράσταση των δεδομένων, η μετατροπή των δεδομένων ανάμεσα σε διαφορετικά πρότυπα, η συμπίεση/αποσυμπίεση των δεδομένων & κρυπτογράφηση όπου απαιτείται.

Έτσι λοιπόν το επίπεδο παρουσίασης (presentation layer) εκτελεί συγκεκριμένες λειτουργίες οι οποίες ζητούνται αρκετά συχνά από τους χρήστες, για να εξασφαλίζουν την εύρεση μιας γενικής λύσης γι' αυτούς, ώστε να μην αφήνεται ο κάθε χρήστης να λύνει τα προβλήματα μόνος του. Συγκεκριμένα, ενώ όλα τα κατώτερα επίπεδα ενδιαφέρονται μόνο για την αξιόπιστη μετακίνηση bits από το ένα μέρος στο άλλο, το επίπεδο παρουσίασης καταπιάνεται με το συντακτικό και τη σημασιολογία των πληροφοριών που μεταδίδονται. Ένα τυπικό παράδειγμα υπηρεσίας παρουσίασης είναι η κωδικοποίηση δεδομένων σε ένα κώδικα που συμφωνήθηκε στη διαδρομή. Επίσης το επίπεδο παρουσίασης ενδιαφέρεται και για άλλα θέματα όπως η αναπαράσταση πληροφοριών.

Για παράδειγμα η συμπίεση των δεδομένων χρησιμοποιείται για να ελαττώσει τον αριθμό των bits που πρόκειται να μεταδοθούν και συχνά απαιτείται κρυπτογράφηση για να εξασφαλιστεί η μυστικότητα (privacy) και η γνησιότητα (authentication) της πληροφορίας. Το επίπεδο είναι υπεύθυνο και για την αναπαράσταση των χαρακτήρων στα πρότυπα ASCII, EBCDIC, UTF, των εικόνων σε GIF, JPEG, TIFF, των βίντεο σε MPEG, κλπ.

Τα χαρακτηριστικά του layer - 6 είναι λοιπόν

- Συμπίεση δεδομένων
- Κρυπτογράφηση
- Κωδικοποίηση χαρακτήρων

Επίπεδο Εφαρμογής (Application Layer) – Στρώμα 7ο

Το επίπεδο εφαρμογής περιέχει όλες τις απαραίτητες δομές για την υποστήριξη της εφαρμογής με την οποία έρχεται σε επαφή ο τελικός χρήστης. Περιέχει δηλαδή τις ρουτίνες και τους μηχανισμούς διεπαφής προγραμματιστικά (Application Programming Interface-API) μεταξύ της υποδομής του υπολογιστικού συστήματος (λειτουργικό σύστημα, οδηγοί δικτυακού υλικού, οδηγοί πρωτοκόλλων) και της εφαρμογής που αναπτύσσεται.

Άρα το επίπεδο εφαρμογής (application layer) περιέχει μια ποικιλία πρωτοκόλλων που χρειάζονται συχνά. Για παράδειγμα μπορούμε να αναφέρουμε το λογισμικό των νοητών τερματικών δικτύων (network virtual terminals) ή την εφαρμογή της μεταφοράς αρχείων. Στην τελευταία περίπτωση διαφορετικά συστήματα αρχείων έχουν διαφορετικούς μεθόδους καθορισμού ονομασίας, διαφορετικούς τρόπους αναπαράστασης των γραμμών κειμένου και ούτω καθεξής. Η μεταφορά ενός αρχείου μεταξύ δύο διαφορετικών συστημάτων απαιτεί αντιμετώπιση αυτών και άλλων μη συμβατών καταστάσεων. Η εργασία αυτή επίσης ανήκει στο επίπεδο εφαρμογής, όπως επίσης και το ηλεκτρονικό ταχυδρομείο, η εμφάνιση καταλόγων αρχείων και διάφορες άλλες ειδικού και γενικού σκοπού ευκολίες .

Τα χαρακτηριστικά του layer - 7 είναι λοιπόν

- Λειτουργίες στο επίπεδο του χρήστη και διασύνδεσης με εφαρμογές (API).
- FTP
- Telnet, Authentication
- SMTP
- NFS, DNS κλπ

Έτσι λοιπόν μια αντιστοίχιση του OSI με το TCP/IP πρότυπο διαδικτύωσης φαινεται παρακάτω.

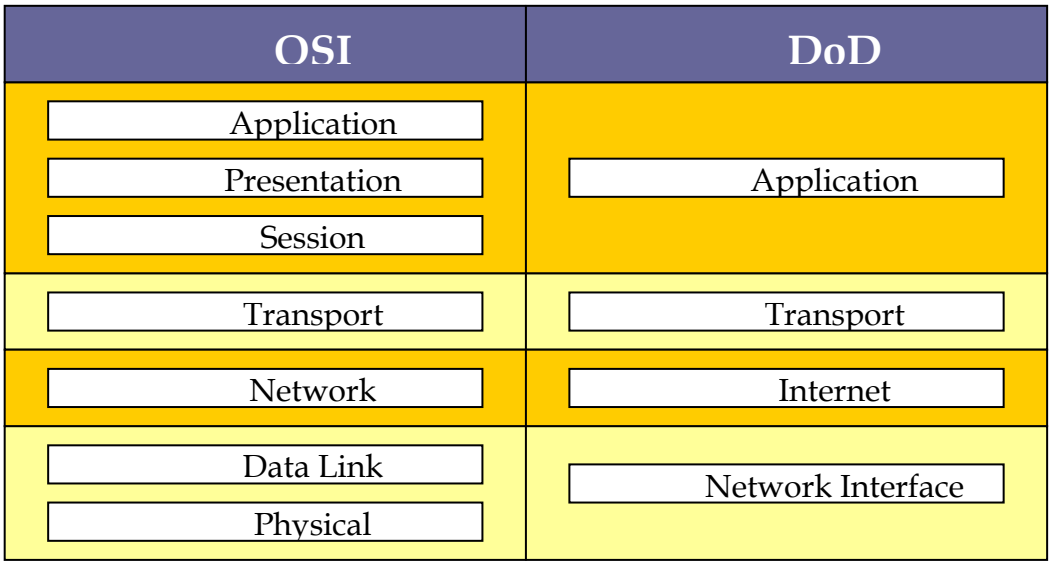
Το **επίπεδο Πρόσβασης Δικτύου** (Network Interface) εκτελεί τις λειτουργίες που στο OSI αντιστοιχούν στο επίπεδο Γραμμής Δεδομένων (Data Link) και το Φυσικό επίπεδο (Physical).

Το **επίπεδο Διαδικτύου** (Internet) αντιστοιχεί στο επίπεδο Δικτύου (Network) του OSI.

Το **επίπεδο Μεταφοράς** (Transport) αντιστοιχεί στο επίπεδο Μεταφοράς (Transport) του OSI.

Το **επίπεδο Εφαρμογής** (Application) εκτελεί τις λειτουργίες των επιπέδων Συνόδου (Session), Παρουσίασης (Presentation) και Εφαρμογής (Application) του OSI.

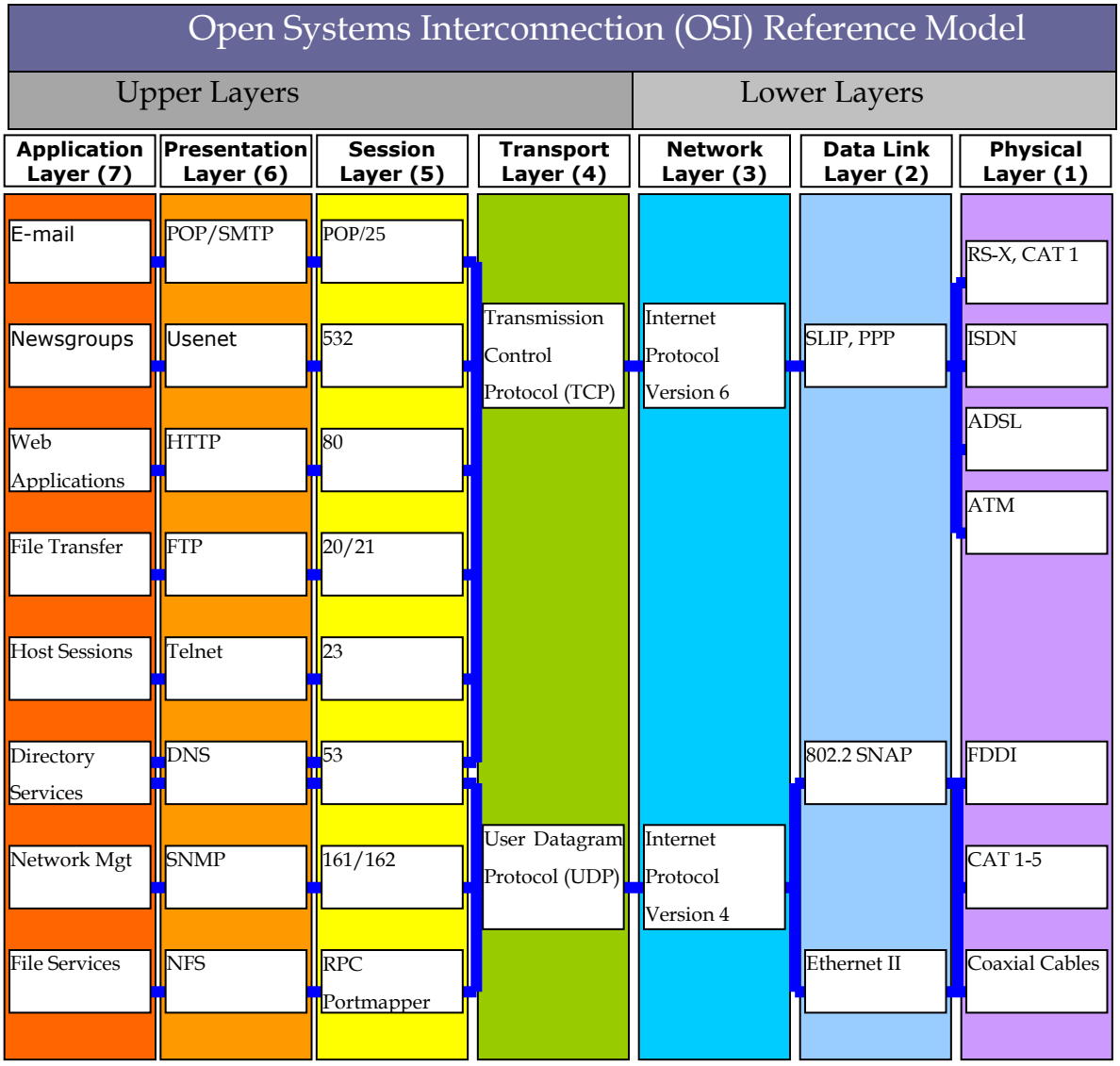
Παρακάτω υπάρχει η γραφική αντιστοίχηση των δύο προτύπων όπου επιχειρείται μια αναγωγή των υπηρεσιών & των πρωτοκόλλων του προτύπου TCP/IP στο OSI.



Αντιστοίχιση των επιπέδων του OSI και του TCP/IP

Το IP είναι πρωτόκολλο τρίτου επιπέδου και χρησιμοποιείται για διασύνδεση ηλεκτρονικών υπολογιστών που μπορούν να ανήκουν στο ίδιο ή σε διαφορετικά δίκτυα.

Η μετάδοση στο IP γίνεται με την τεχνική των πακέτων (datagrams). Το κάθε πακέτο του IP φθάνει στον παραλήπτη διασχίζοντας ένα ή περισσότερα διασυνδεδεμένα δίκτυα IP, χωρίς να εξαρτάται από άλλα προηγούμενα ή επόμενα πακέτα διατηρώντας έτσι την αυτονομία του μέσα στο δίκτυο.



Αναγωγή του TCP/IP μοντέλου σε OSI και κάποια ενδεικτικά πρωτόκολλα που αντιστοιχούν σε κάθε επίπεδο

Το IP ως πρωτόκολλο τρίτου επιπέδου δεν ασχολείται με τις φυσικές συνδέσεις ή τον έλεγχο των ενδιάμεσων ζεύξεων μεταξύ των κόμβων του δικτύου (που είναι αρμοδιότητα άλλων πρωτοκολλων χαμηλότερων επιπέδων όπως Ethernet, Frame Relay, PPP, κλπ). Στην ουσία ασχολείται με την διευθυνσιοδότηση, τον κατακερματισμό (fragmentation) μεγάλων πακέτων και την επανασυγκόλληση τους.

Το πρωτόκολλο IP δεν θεωρείται αξιόπιστο καθώς δεν εξασφαλίζει την απ' άκρου εις άκρο ακεραιότητα των δεδομένων μέσω κάποιων τεχνικών επανεκπομπής, ελέγχου ροής κλπ. Οι λειτουργίες αυτές επιτυγχάνονται με το πρωτόκολλο TCP που είναι στο αμέσως ανώτερο επίπεδο. Το IP δεν απαιτεί την αποκατάσταση σύνδεσης μεταξύ δύο σημείων πριν την αναταλλαγή δεδομένων και γι' αυτό χαρακτηρίζεται ως χωρίς σύνδεση (connectionless).

Το IP παραλαμβάνει τα δεδομένα από το ανώτερο επίπεδο σε πακέτα με μέγιστο μέγεθος 64 Kbyte και διαιρεί το κάθε πακέτο σε περισσότερα τμήματα (fragments) (αν είναι απαραίτητο) και τα μεταδίδει μέσω του δικτύου. Ο κατακερματισμός αυτός γίνεται στις περιπτώσεις που τα πακέτα IP πρέπει να περάσουν από δίκτυα που έχουν περιορισμό στο μέγιστο μέγεθος πλαισίου (frame). Το Ethernet για παράδειγμα μπορεί να χειριστεί πλαίσια μεγέθους 64 έως 1500 Byte. Το έργο του κατακερματισμού μπορεί να γίνει από οποιαδήποτε ενδιάμεση συσκευή (π.χ. δρομολογητή) του δικτύου, ενώ η επανασυγκόλληση των IP πακέτων γίνεται από τον τελικό παραλήπτη.

Στο χειρισμό του πρωτοκόλλου IP συμμετέχουν μόνο οι δύο ακραίοι υπολογιστικοί σταθμοί και οι ενδιάμεσοι δρομολογητές. Την δρομολόγηση του IP πρωτοκόλλου αναλαμβάνουν οι δρομολογητές οι οποίοι γνωρίζουν την τοπολογία του δικτύου και διαθέτουν κατάλληλους πίνακες δρομολόγησης. Έτσι οι χρήστες αρκεί να γνωρίζουν μόνο την τελική διεύθυνση του αποδέκτη ώστε να δρομολογηθεί κατάλληλα το μήνυμά τους.

1.4.2 Κατηγορίες VPN βάσει του προτύπου OSI

Τα Εικονικά Ιδιωτικά Δίκτυα κατηγοριοποιούνται με διάφορους τρόπους, ανάλογα με την οπτική γωνία που τα εξετάζει κανείς. Με βάση την αντιστοιχία τους με τα επίπεδα του μοντέλου αναφοράς OSI, τα Εικονικά Ιδιωτικά Δίκτυα κατηγοριοποιούνται ως εξής:

- Data link layer VPNs (layer 2)
- Network layer VPNs (layer 3)
- Application layer VPNs (layer 7)

Data Link Layer VPNs (Στρώμα 2)

Στα data link layer VPNs, δυο ιδιωτικά δίκτυα συνδέονται στο Layer-2 του OSI model χρησιμοποιώντας πρωτόκολλα όπως Frame Relay ή ATM. Παρόλο που αυτοί οι μηχανισμοί παρέχουν έναν αρκετά βολικό τρόπο για την δημιουργία VPNs είναι πολύ συχνά ακριβοί, διότι χρειάζονται μισθωμένες (Layer 2) γραμμές.

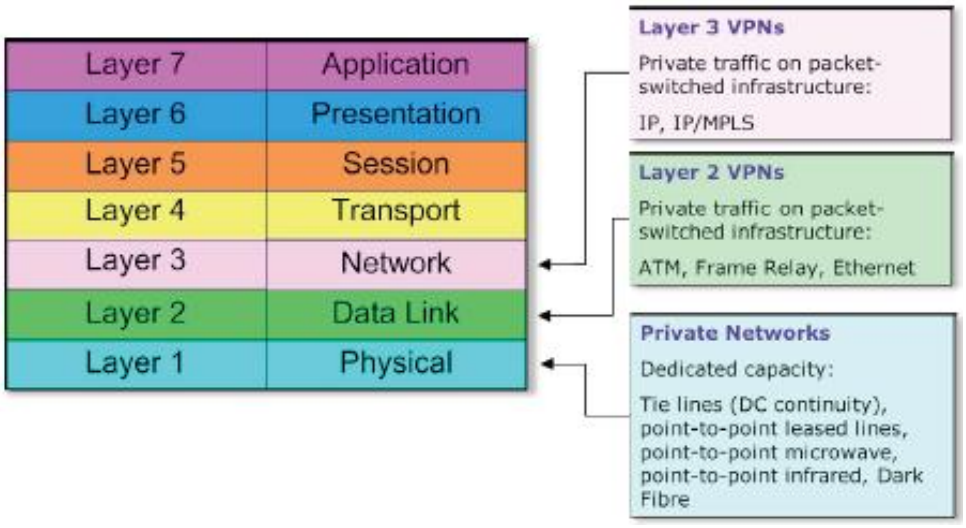
Ακόμα τα Frame Relay και ATM πρωτόκολλα εγγενώς δεν παρέχουν μηχανισμούς κρυπτογράφησης. Επιτρέπουν μόνο να γίνεται διαχωρισμός στη κίνηση βασισμένος σε ποιο layer2 connection ανήκει αυτή. Επομένως εάν χρειαζόμαστε περαιτέρω ασφάλεια είναι σημαντικό να υπάρξει κάποιο είδος μηχανισμού κρυπτογράφησης σε ισχύ.

Δύο από τα ευρύτατα χρησιμοποιημένα πρωτόκολλα για τη δημιουργία ενός layer-2 VPN μέσω του Διαδικτύου είναι και το point-to-point tunneling protocol (PPTP).

Network Layer VPNs (Στρώμα 3)

Τα Network Layer VPNs δημιουργούνται χρησιμοποιώντας το Layer-3 tunneling και τεχνικές κρυπτογράφησης. Ένα παράδειγμα είναι η χρήση του IPSec tunneling και πρωτοκόλλου κρυπτογράφησης για την δημιουργία VPNs. Άλλα παραδείγματα είναι τα πρωτόκολλα GRE και L2TP. Ακόμα είναι ενδιαφέρον να σημειωθεί ότι αν και το L2TP tunnels layer-2 κίνηση χρησιμοποιεί το layer-3 (το στρώμα IP) για να το κάνει αυτό. Επομένως, για αυτόν το λόγο είναι ταξινομημένο ως Network Layer VPN.
Σε αυτό το layer ανήκουν τα VPN που δομούνται πάνω σε IP δίκτυα και χρησιμοποιούν το πρωτόκολλο IPSec, καθώς και τα VPN που δομούνται πάνω σε MPLS δίκτυα.

Έτσι λοιπόν το Network Layer παρέχει μια πολύ κατάλληλη θέση για να χρησιμοποιηθεί κρυπτογράφηση. Το επίπεδο δικτύου είναι αρκετά χαμηλά στο πρότυπο διασύνδεσης OSI για να παρέχει seamless connectivity σε όλες τις εφαρμογές που τρέχουν πάνω από αυτό και ταυτόχρονα αρκετά υψηλά ώστε να επιτρέψει τον κατάλληλο βαθμό λεπτομέρειας για την κυκλοφορία που πρέπει να είναι μέρος του VPN βασισμένο στην εκτενή IP addressing αρχιτεκτονική που βρίσκεται σε ισχύ. Λόγω της φυσικής του θέσης η Cisco εστιάζει την κρυπτογράφηση στο network layer ως κυρίου μηχανισμού για την δημιουργία VPNs.

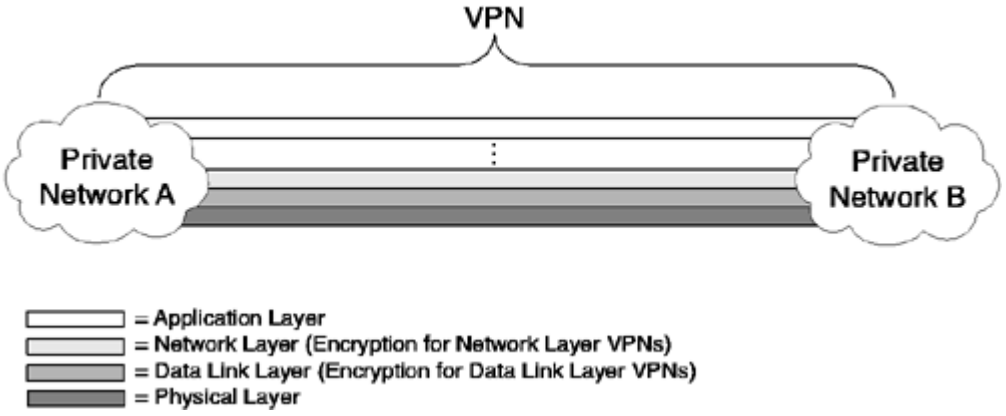


Application Layer VPNs (Στρώμα 7)

Το Application Layer VPN δημιουργείται για να λειτουργήσει συγκεκριμένα με ορισμένες εφαρμογές. Ένα πολύ καλό παράδειγμα τέτοιων είναι τα VPNs βασισμένα στο SSL. Το SSL παρέχει την κρυπτογράφηση μεταξύ των web browsers και των servers που τρέχουν το SSL. Ένα άλλο καλό παράδειγμα application layer VPN είναι και το SSH. Το SSH ωθείται ως μηχανισμός για τις κρυπτογραφημένες και ασφαλείς συνόδους σύνδεσης (login sessions) στις διάφορες συσκευές δικτύων. Ακόμα το SSH μπορεί να κρυπτογραφήσει και να δημιουργήσει έτσι VPNs για άλλα πρωτόκολλα του application layer, όπως το FTP και το HTTP.

Ένα από τα κύρια μειονεκτήματα των application layer VPNs είναι ότι συχνά ο χρήστης πρέπει να εκτελέσει την εφαρμογή που θα επιτρέψει στις συσκευές να δημιουργήσουν το VPN. Ακόμα με τις νέες υπηρεσίες και τις αντίστοιχες εφαρμογές που προστίθενται χρειάζεται περαιτέρω υποστήριξη να αναπτυχθεί για αυτές.

Αυτό είναι αντίθετο από τα network layer και link layer VPNs, τα οποία παρέχουν seamless VPN connectivity για όλες τις εφαρμογές αφότου το βασικό VPN έχει εγκατασταθεί. Έτσι λοιπόν ενώ τα layer-2 VPN χρησιμοποιούν layer-2 πλαίσιο όπως το Ethernet, τα layer-3 VPN χρησιμοποιούν πακέτα στρώματος δικτύου όπως το IP.



1.4.3 Τύποι Εικονικών Ιδιωτικών Δικτύων (VPN)

Υπάρχουν κυρίως δύο διαδεδομένοι τύποι εικονικών ιδιωτικών δικτύων.

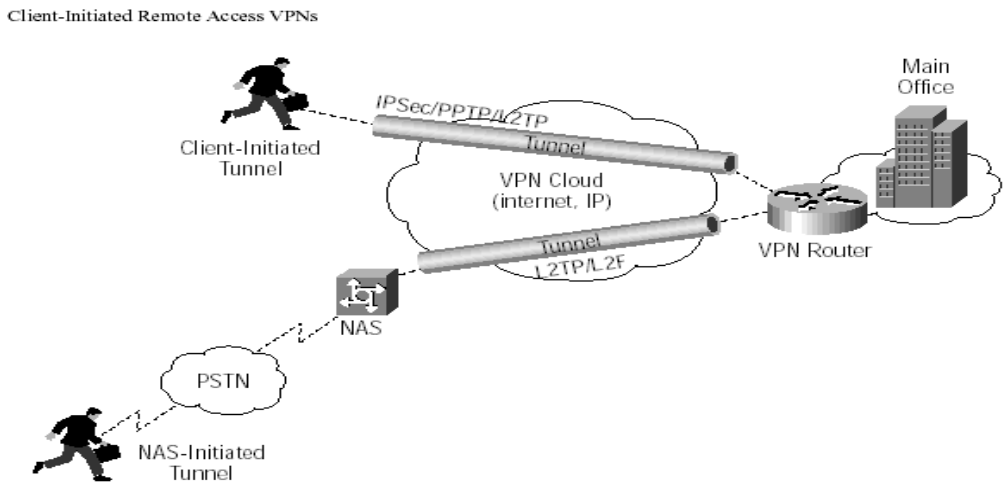
- **Remote – Access (VPDN)**
- **Site-to-Site**

1. **Remote – Access:** Επίσης αποκαλείται και **virtual private dial-up network [VPDN]**. Αυτό αποτελεί μία σύνδεση από τον χρήστη προς το τοπικό δίκτυο, που χρησιμοποιείται από μια εταιρία των οποίων οι υπάλληλοι χρειάζεται να συνδεθούν στο ιδιωτικό δίκτυο από διάφορες απομακρυσμένες τοποθεσίες.

   Για τη δημιουργία ενός Dial-up VPN απαιτείται ένας **N**etwork **A**ccess **S**erver **NAS** (Εξυπηρετητή Δικτυακής Πρόσβασης) μέσω του οποίου τα μέλη της επιχείρησης θα αποκτήσουν πρόσβαση στο δίκτυό της. Τον Network Access Server τον παρέχει μια εταιρεία **E**nterprise **S**ervice **P**rovider **ESP**, η οποία εγκαθιστά και το απαραίτητο λογισμικό (software) για την πρόσβαση των εργαζομένων στο NAS.

   Οι telecommuters μπορούν να καλέσουν στην συνέχεια, μέσω τηλεφώνου, έναν αριθμό χωρίς χρέωση, για να επικοινωνήσουν με τον Εξυπηρετητής Δικτυακής Πρόσβασης (NAS) και στην συνέχεια χρησιμοποιούν το λογισμικό εικονικών ιδιωτικών δικτύων που τους χορηγήθηκε (VPN client software) για να αποκτήσουν πρόσβαση στο εταιρικό δίκτυο.

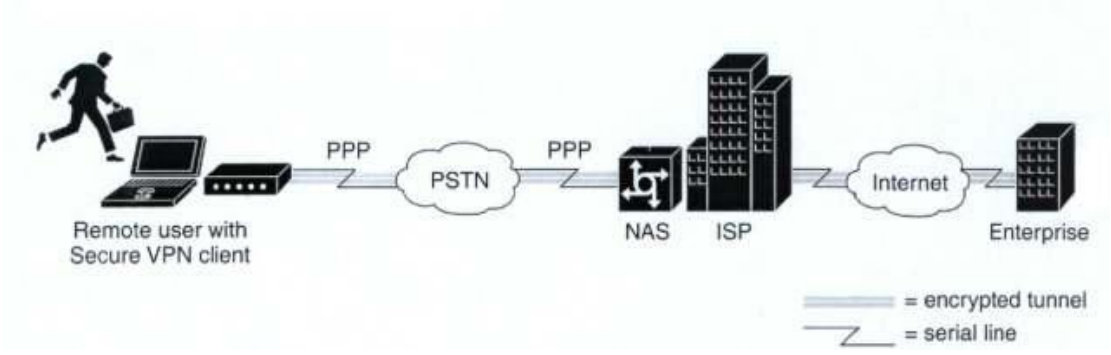
   Ένα καλό παράδειγμα εταιρίας που χρειάζεται ένα remote-access VPN θα ήταν μια μεγάλη φίρμα με εκατοντάδες πωλητές σε μια περιοχή. Ακόμα επιτρέπουν ασφαλή, κωδικοποιημένες συνδέσεις μεταξύ του ιδιωτικού δικτύου μιας επιχείρησης με τους απομακρυσμένους χρήστες, διαμέσου κάποιου τρίτου παροχέα δικτυακών υπηρεσιών.



Σχήμα 1.2: Remote Access VPN

Τα remote access VPNs χωρίζονται σε δύο κατηγορίες: στα **client-initiated** και στα **network access server** (NAS) **initiated**.

(1.α) Client initiated: Στην περίπτωση αυτή, απομακρυσμένοι χρήστες χρησιμοποιούν client εφαρμογές για να δημιουργήσουν κρυπτογραφημένα IP tunnels, μέσω του διαμοιραζόμενου δικτύου ενός ISP, προς το δίκτυο κάποιας εταιρείας.

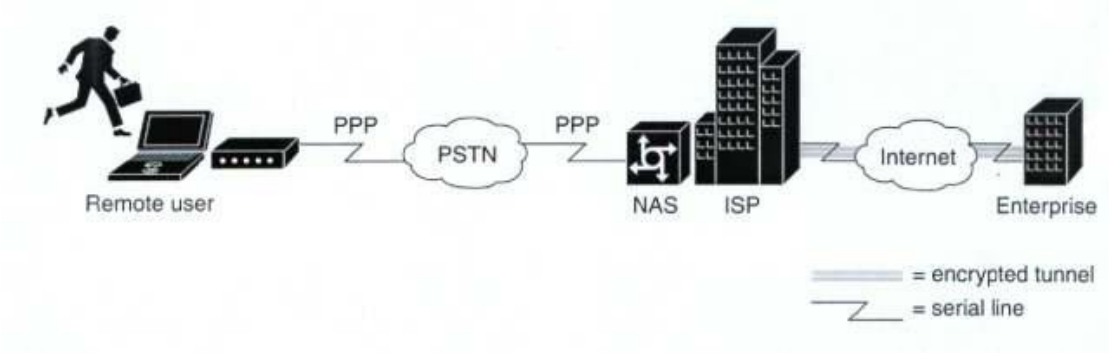


Σχήμα 1.3: Client-initiated remote access VPN

Στο παραπάνω σχήμα 1.3 φαίνεται ένα client-initiated remote access VPN. Η client εφαρμογή δημιουργεί μία PPP σύνδεση με το NAS του ISP και στη συνέχεια σχηματίζεται ένα κρυπτογραφημένο tunnel μέσω του δημόσιου τηλεφωνικού δικτύου.

(1.β) NAS-initiated: Στην περίπτωση αυτή οι απομακρυσμένοι χρήστες κάνουν μία κλήση στο NAS του ISP και αυτό δημιουργεί ένα κρυπτογραφημένο tunnel με το VPN της εταιρείας. Τα NAS-initiated VPN δίνουν τη δυνατότητα στους χρήστες να συνδεθούν σε διάφορα δίκτυα χρησιμοποιώντας πολλαπλά tunnels, ενώ η client εφαρμογή δεν χρειάζεται να έχει λογισμικό για τη δημιουργία tunnels. Το αρνητικό στην περίπτωση αυτή είναι ότι η σύνδεση μεταξύ του χρήστη και του ISP δεν είναι κρυπτογραφημένη και άρα στηρίζεται στο PSTN, το οποίο δυστυχώς δεν παρέχει καμία ασφάλεια.

Το διάγραμμα ενός τέτοιου VPN φαίνεται στο Σχήμα 1.4 .

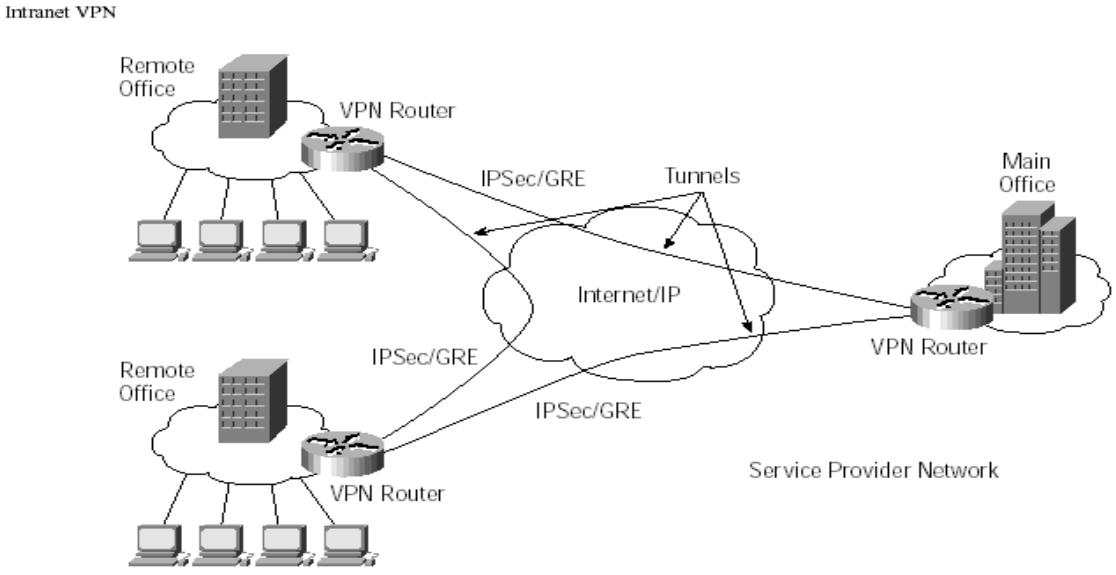


Σχήμα 1.4: NAS-initiated remote access VPN

2. **Site-to-Site :** Μέσω της χρήσης αποκλειστικά αφιερωμένου εξοπλισμού και μεγάλου βαθμού κωδικοποίησης, μία εταιρία μπορεί να συνδέσει πολλαπλές σταθερές σελίδες πάνω σε δημόσιο δίκτυο, όπως το internet. Η σύνδεση των εργαζομένων γίνεται πάλι μέσω ειδικού λογισμικού και εξοπλισμού, ενώ για την ασφάλεια των δεδομένων χρησιμοποιείται κρυπτογράφηση (encryption), όπως αυτή που συνιστάται και στα κοινά sites.

Το **Site-to-site VPN** χωρίζεται σε δύο κατηγορίες ανάλογα με το τύπο που μπορεί να είναι:

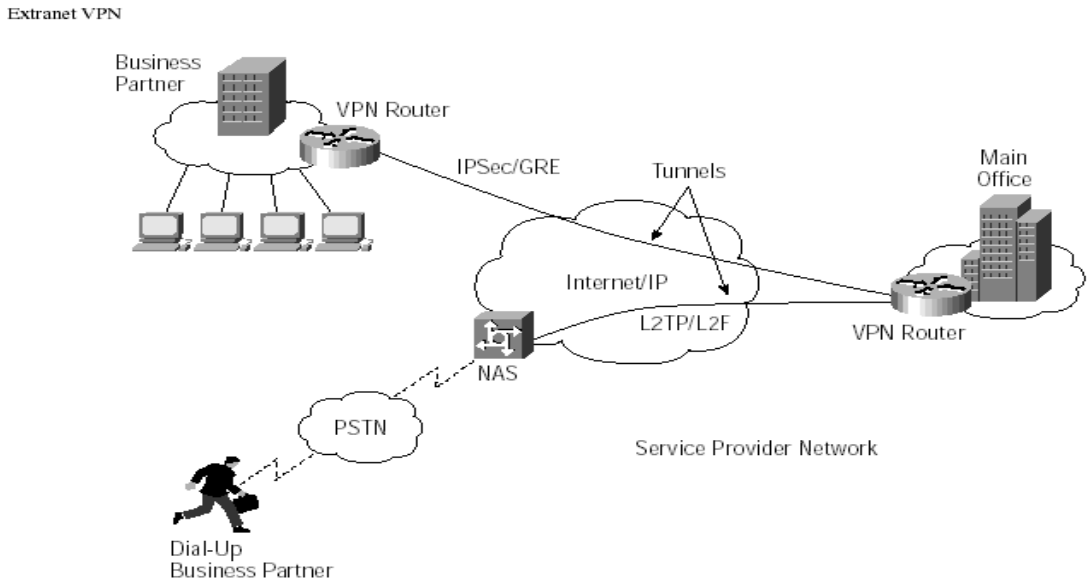
(2.α) Intranet-Based : Αν μια επιχείρηση έχει μία ή παραπάνω τοποθεσίες που θα ήθελε να ενώσει σε ένα μόνο ιδιωτικό δίκτυο, μπορεί να δημιουργήσει ένα VPN βασισμένο σε intranet, έτσι ώστε να ενώσει το ένα τοπικό δίκτυο με το άλλο (LAN to LAN). Τα Intranet VPNs επιτρέπουν στα ιδιωτικά δίκτυα να εκτείνονται πάνω στο Internet ή σε άλλα δημόσια δίκτυα με έναν ασφαλή τρόπο. Τα Intranet VPNs συχνά αναφέρονται σαν site-to-site ή LAN-to-LAN VPNs. Ένα intranet είναι ένα δίκτυο εργασίας το οποίο είναι εσωτερικό σε κάποια εταιρεία και παρέχει τις πιο πρόσφατες πληροφορίες και υπηρεσίες σε όλους τους υπαλλήλουςτης εταιρείας που συνδέονται σε αυτό. Τα Intranets προσφέρουν ένα κοινό, interface που είναι ανεξάρτητο πλατφόρμας, το οποίο είναι λιγότερο ακριβό στην υλοποίηση από μία client/server εφαρμογή. Επιπλέον τα Intranets αυξάνουν την παραγωγικότητα των υπαλλήλων επιτρέποντας μία αξιόπιστη σύνδεση σε πληροφορίες συνεχής ροής. Στο Σχήμα 1.5 φαίνεται μία Intranet VPN τοπολογία.



Σχήμα 1.5: Intranet VPN

(2.β) Extranet Based : Όταν μια επιχείρηση έχει στενές σχέσεις με μία άλλη (π.χ. αν είναι συνεργάτες ή προμηθευτής ή σημαντικός πελάτης), μπορεί να χτιστεί ένα Εικονικό Ιδιωτικό Δίκτυο βασισμένο σε extranet για να γίνει σύνδεση LAN to LAN. Αυτό επιτρέπει όλες τις εταιρίες που θα δημιουργήσουν μεταξύ τους ένα extranet VPN να μοιραστούν το ίδιο εικονικό περιβάλλον. Τα Extranet VPNs επιτρέπουν ασφαλείς συνδέσεις μεταξύ των συνεταίρων των επιχειρήσεων και των πελατών. Τα Extranet VPNs αποτελούν μια επέκταση των Intranet VPNs με την προσθήκη firewalls για την προστασία του εσωτερικού δικτύου.

Τα extranets διαφέρουν από τα intranets στο ότι επιτρέπουν πρόσβαση και σε χρήστες που δεν είναι υπάλληλοι της εταιρείας είτε μέσω χρήσης του HTTP πρωτοκόλλουειτε μέσω κάποιου πρωτοκόλλου στο οποίο θα συμφωνούν οι συμβαλλόμενοι στην επικοινωνία φορείς. Χρησιμοποιώντας ψηφιακά πιστοποιητικά, οι πελάτες δημιουργούν μέσω Internet, ασφαλή tunnels προς το δίκτυο μίας εταιρείας. Κάθε πελάτης λαμβάνει από την Αρχή Πιστοποίησης (Certification Authority – CA) ένα ψηφιακό πιστοποιητικό το οποίο χρησιμοποιείται για πιστοποίηση από τον CA server της εταιρείας.



Σχήμα 1.6: Extranet VPN

Στο Σχήμα 1.6 φαίνεται μία Extranet VPN τοπολογία.

Σε όποια μορφή και αν συναντήσουμε ένα VPN, η αρχή λειτουργίας του παραμένει η ίδια. Κάθε απομακρυσμένο μέλος μιας εταιρείας, μπορεί μέσω του ίδιου του Internet να επικοινωνήσει με το προσωπικό δίκτυο της εταιρείας με ασφάλεια και αξιοπιστία. Όπως συμβαίνει και με ένα απλό τοπικό δίκτυο (LAN), ένα VPN μπορεί να μεγαλώσει αρκετά εύκολα, προκειμένου να μπορεί να εξυπηρετήσει περισσότερους εργαζόμενους, γεγονός το οποίο αποτελεί ένα σημαντικό πλεονέκτημα των VPNs. Ένα VPN μπορεί να επεκταθεί και να μεγαλώσει και σε άλλες περιοχές, χωρίς να υπάρχει κάποιο μεγάλο κόστος, σε αντίθεση με κάποια μισθωμένη γραμμή, όπου το κόστος είναι υψηλότερο όσο μεγαλώνει η απόσταση.

1.5. Ασφάλεια των VPN

Το πιο σημαντικό στοιχείο σε ένα VPN είναι η ασφάλεια. Αυτή επιτυγχάνεται με γνωστά μέσα όπως

- Τείχος προστασίας (firewall),
- AAA Servers,
- Έλεγχος αυθεντικότητας του χρήστη (authentication),
- Κρυπτογράφηση δεδομένων (encryption),
- Το πρωτόκολλο IPsec (Internet Protocol Security),
- Tunneling

Τείχος Προστασίας (Firewalls) – Το firewall (ή πύρινο τοίχος) αποτελεί έναν σημαντικό φύλακα ανάμεσα στο ιδιωτικό δίκτυο και το Internet. Τα firewalls μπορούν να ρυθμιστούν έτσι ώστε να περιορίζουν τον αριθμό των ανοιχτών θυρών, τι είδος πακέτων επιτρέπεται να περάσουν στο LAN, ακόμα και το ποια θα είναι τα επιτρεπόμενα πρωτόκολλα. Θα πρέπει να υπάρχει τοποθετημένο ένα καλό firewall πριν μπει σε λειτουργία ένα Εικονικό Ιδιωτικό Δίκτυο (VPN). Ένα firewall μπορεί επίσης να χρησιμοποιηθεί για να τερματίσει μια περίοδο ενός Εικονικού Ιδιωτικού Δικτύου.

AAA Servers – Οι εξυπηρέτες AAA (authentication, authorization and accounting) χρησιμοποιούνται για ακόμα πιο ασφαλή πρόσβαση σε ένα απομακρυσμένο περιβάλλον VPN. Όταν λαμβάνεται μια αίτηση για δημιουργία νέας περιόδου σύνδεσης από κάποιον πελάτη μέσω τηλεφώνου, η αίτηση πηγαίνει από έναν proxy (διαμεσολαβητή) στον εξυπηρέτη AAA. Ο AAA αμέσως μετά κάνει τον ακόλουθο έλεγχο:

- Ταυτότητας (authentication)
- Δικαιωμάτων (authorization)
- Πραγματικών ενεργειών (accounting)

Οι πληροφορίες περί των πραγματικών ενεργειών του χρήστη ενός AAA είναι πολύ χρήσιμες ειδικά για την καταγραφεί των κινήσεων του πελάτη για λόγους ελέγχου, χρέωσης αλλά και για λόγους παιδαγωγικούς.

Αυθεντικοποίηση (Authentication) - Η αυθεντικοποίηση χρησιμοποιείται στα VPNs για να εξασφαλίσει ότι τα επικοινωνούντα μέρη ανταλλάσουν πληροφορίες με το σωστό χρήστη ή μηχάνημα. Τα περισσότερα συστήματα αυθεντικοποίησης που χρησιμοποιούνται βασίζονται στο σύστημα διαμοίρασης κλειδιού (shared key). Αυτά τα κλειδιά περνούν από ένα αλγόριθμο κωδικοποίησης σύνοψης (one way hash algorithm) και προκύπτει μία τιμή σύνοψης (hash value) η οποία αποστέλλεται στον αποδέκτη. Ακόμα εκτος από pre-shared key υπάρχουν και άλλοι τρόποι αυθεντικοποίησης όπως certificate-based και username/password authentication. Η χρησιμοποίηση του pre-shared key είναι η ευκολότερη με την certificate-based να είναι η πιο ανθεκτική και πλούσια σε χαρακτηριστικά.

Ο αποδέκτης που έχει στην κατοχή του τον αλγόριθμο κωδικοποίησης θα παράγει μία τιμή σύνοψης (hash value) και θα την συγκρίνει με αυτήν που στάλθηκε από τον αποστολέα. Το αποτέλεσμα της κωδικοποίησης σύνοψης (hash value) το οποίο στάλθηκε μέσω του διαδικτύου δεν έχει αξία για τον πιθανό εξωτερικό παρατηρητή οπότε διασφαλίζεται η αυθεντικοποίηση της επικοινωνίας. Η αυθεντικοποίηση τυπικά γίνεται στην αρχή της συνεδρίας και ύστερα σε τυχαίες χρονικές στιγμές κατά την διάρκεια της συνεδρίας για να εξασφαλιστεί ότι κάποιος εισβολέας δεν εισχώρησε στην επικοινωνία.
Ένα τυπικό παράδειγμα μεθόδου αυθεντικοποίησης είναι το CHAP (Challenge Handshake Authentication Protocol) για το οποίο εκτενής ανάλυση γίνεται παρακάτω. Πιο αναλυτικά για την διαδικασία του authentication θα αναφερθούμε παρακάτω στην κρυπτογράφηση δεδομένων και συγκεκριμένα εκεί όπου μας ενδιαφέρει ιδιαιτέρως δηλαδή στην public key encryption (κρυπτογράφηση δημοσιου κλειδιού).

Κρυπτογράφηση δεδομένων (Encryption) - Με αυτήν την διαδικασία παίρνουμε όλα τα δεδομένα που ένας υπολογιστής στέλνει σε έναν άλλον, κωδικοποιώντας τα σε μια φόρμα που μόνο ο άλλος υπολογιστής θα είναι σε θέση να αποκωδικοποιήσει. Τα περισσότερα Συστήματα Απόκρυψης Κώδικα για Υπολογιστή (computer encryption systems) ανήκουν σε μια από τις δύο παρακάτω κατηγορίες:

- Symmetric-key encryption
- Public-key encryption

(α) Στην **Κρυπτογράφηση Συμμετρικού Κλειδιού (Symmetric-key encryption)**, κάθε υπολογιστής έχει ένα κρυφό κλειδί (κώδικα) το οποίο μπορεί να χρησιμοποιήσει για να αποκρύψει ένα πακέτο πληροφοριών πριν σταλεί μέσω δικτύου σε κάποιον άλλο υπολογιστή. Το Symmetric-key προϋποθέτει ότι υπάρχει γνώση για το ποιοι υπολογιστές θα «συνομιλήσουν» έτσι ώστε να μπορέσεις να εγκαταστήσεις το κλειδί στον καθένα.

Η Symmetric-key απόκρυψη δεδομένων είναι συνήθως σαν ένας κρυφός κώδικας που και οι δύο υπολογιστές θα πρέπει να γνωρίζουν, για να μπορέσουν να αποκωδικοποιήσουν τις πληροφορίες. Ο κώδικας παρέχει το κλειδί για την αποκωδικοποίηση του μηνύματος. Είναι σαν να δημιουργούμε ένα κωδικοποιημένο μήνυμα για να αποσταλεί σε έναν φίλο, στο οποίο κάθε γράμμα είναι αντικατεστημένο με το γράμμα που βρίσκετε 2 θέσεις πιο πριν στο αλφάβητο. Έτσι το "A" γίνεται "C," και το "B" γίνεται "D", κ.ο.κ.. Βέβαια εμείς έχουμε πληροφορήσει τον φίλο μας πως να αποκωδικοποιήσει το γράμμα. Έτσι αυτός όταν παίρνει το γράμμα μπορεί και το διαβάζει. Για τους υπόλοιπους το περιεχόμενο του γράμματος δεν είναι κατανοητό.

Το βασικό πλεονέκτημα των συμμετρικών αλγορίθμων είναι ότι οι χρήστες δεν καταλαβαίνουν κάποια σημαντική χρονική καθυστέρηση λόγω της κωδικοποίησης / αποκωδικοποίησης, αρκεί να διατηρείται μυστικό το κλειδί κωδικοποίησης. Η ασφάλεια των συμμετρικών κρυπτογραφικών συστημάτων

εξαρτάται από την προφύλαξη της μυστικότητας του κλειδιού. Ένα παράδειγμα τέτοιας κωδικοποίησης ιδιωτικού κλειδιού που χρησιμοποιούν τα συστήματα Unix είναι το DES (Data Encryption Standard). Άλλοι γνωστοί αλγόριθμοι είναι οι Triple-DES, IDEA, RC2, RC4 και AES. Όταν χρησιμοποιείται η κωδικοποίηση ιδιωτικού κλειδιού υπάρχει το εξής πρόβλημα «όλα τα επικοινωνούντα μέρη πρέπει να γνωρίζουν το μυστικό κοινό κλειδί». Το πρόβλημα οξύνεται όταν έχουμε ένα μεγάλο δίκτυο με πολλούς χρήστες.

Συμμετρικό Κρυπτοσύστημα λοιπόν είναι εκείνο το οποίο χρησιμοποιεί κατά την διαδικασία της κρυπτογράφησης αποκρυπτογράφησης ένα κοινό κλειδί (Σχ 1.7). Η ασφάλεια αυτών των αλγορίθμων βασίζεται στην μυστικότητα του κλειδιού. Τα συμμετρικά κρυπτοσυστήματα προϋποθέτουν την ανταλλαγή του κλειδιού μέσα από ένα ασφαλές κανάλι επικοινωνίας ή μέσα από την φυσική παρουσία των προσώπων. Αυτό το χαρακτηριστικό καθιστά δύσκολη την επικοινωνία μεταξύ απομακρυσμένων ατόμων.

Σχ. 1.7 – Συμμετρικό Κρυπτοσύστημα

(β) Από ιστορικής πλευράς, το πρόβλημα της διανομής των κλειδιών ήταν πάντα ο αδύναμος κρίκος των περισσότερων κρυπτοσυστημάτων. Άσχετα με το πόσο δυνατό ήταν το κρυπτοσύστημα, εάν ένας παρείσακτος μπορούσε να κλέψει το κλειδί το σύστημα ήταν άχρηστο. Εφόσον όλοι οι κρυπτολόγοι θεωρούσαν δεδομένο ότι τα κλειδιά κρυπτογράφησης και αποκρυπτογράφησης ήταν τα ίδια (ή θα προέκυπταν εύκολα το ένα από το άλλο) και το κλειδί έπρεπε να διανεμηθεί σ' όλους τους χρήστες του συστήματος, φαινόταν σαν να υπήρχε ένα εγγενές πρόβλημα: τα κλειδιά έπρεπε να προφυλαχθούν από κλοπή, αλλά έπρεπε επίσης και να διανεμηθούν, επομένως δεν μπορούσαν να κλειδωθούν στο θησαυροφυλάκιο μιας τράπεζας. Η **κρυπτογράφηση δημοσίου κλειδιού (Public Key Cryptography) ή ασύμμετρου κλειδιού (Asymmetric Cryptography)** επινοήθηκε στο τέλος της δεκαετίας του 1970 από τους Whitfield Diffie και Martin Hellman και παρέχει έναν εντελώς διαφορετικό μοντέλο διαχείρισης των κλειδιών κρυπτογράφησης. Στην συμμετρική κρυπτογραφία η εμπιστευτικότητα των δεδομένων μπορεί να εξασφαλιστεί, αλλά όχι και η ταυτοποίηση του αποστολέα και επιπλέον είναι πολύ δύσκολη η διανομή του μυστικού κλειδιού με ασφάλεια. Η βασική ιδέα

της κρυπτογράφησης δημοσίου κλειδιού είναι ότι ο αποστολέας και ο παραλήπτης δεν μοιράζονται ένα κοινό μυστικό κλειδί όπως στην περίπτωση της κρυπτογράφησης συμμετρικού κλειδιού, αλλά διαθέτουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες.

Η κρυπτογραφία λοιπόν δημόσιου κλειδιού πρόκειται για ένα εντελώς διαφορετικό μοντέλο διαχείρισης κρυπτογραφικών κλειδιών, όπου η βασική ιδέα είναι ότι ο αποστολέας και ο παραλήπτης δεν διαμοιράζονται ένα μυστικό κλειδί, αλλά αντιθέτως έχουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες.

Το κλειδί κρυπτογράφησης δημοσιεύεται στο ευρύ κοινό, ενώ αυτό της αποκρυπτογράφησης κρατείται μυστικό (ιδιωτικό κλειδί). Οποιοσδήποτε διαθέτει το δημόσιο κλειδί μπορεί να κρυπτογραφήσει το κείμενο αλλά όχι και να το αποκρυπτογραφήσει. Το βασικό προτέρημα αυτού του είδους της κρυπτογραφίας (και ο λόγος που αρχικά πολεμήθηκε η διανομή του) είναι ότι επιτρέπει στο ευρύ κοινό να ανταλλάσει μηνύματα με ασφαλή τρόπο. Μέχρι τώρα αυτό ήταν προνόμιο μόνο των κυβερνήσεων και των μεγάλων οργανισμών που είχαν την οικονομική ευχέρια να αναπτύξουν ασφαλή δίκτυα διανομής των μυστικών κλειδιών τους.

Συγκεκριμένα κάθε χρήστης διαθέτει δύο κλειδιά κρυπτογράφησης: το ένα ονομάζεται **ιδιωτικό κλειδί (private key)** και το άλλο **δημόσιο κλειδί (public key)**. Το ιδιωτικό κλειδί θα πρέπει ο κάθε χρήστης να το προφυλάσσει και να το κρατάει κρυφό, ενώ αντιθέτως το δημόσιο κλειδί θα πρέπει να το ανακοινώνει σε όλη την διαδικτυακή κοινότητα. Υπάρχουν δε και ειδικοί εξυπηρετητές δημοσίων κλειδιών (public key servers) στους οποίους μπορεί κανείς να απευθυνθεί για να βρει το δημόσιο κλειδί του χρήστη που τον ενδιαφέρει ή να ανεβάσει το δικό του δημόσιο κλειδί για να είναι διαθέσιμο στο κοινό.

Τα δύο αυτά κλειδιά (ιδιωτικό και δημόσιο) έχουν μαθηματική σχέση μεταξύ τους. Εάν το ένα χρησιμοποιηθεί για την κρυπτογράφηση κάποιου μηνύματος, τότε το άλλο χρησιμοποιείται για την αποκρυπτογράφηση αυτού.

Η επιτυχία αυτού του είδους κρυπτογραφικών αλγορίθμων βασίζεται στο γεγονός ότι η γνώση του δημόσιου κλειδιού κρυπτογράφησης δεν επιτρέπει με κανέναν τρόπο τον υπολογισμό του ιδιωτικού κλειδιού κρυπτογράφησης. Στην εικόνα (Σχ. 1.8) φαίνεται ένα δημόσιο κλειδί όπου τα 1024 bits του κλειδιού αναπαριστώνται ως μία ακολουθία αλφαριθμητικών χαρακτήρων

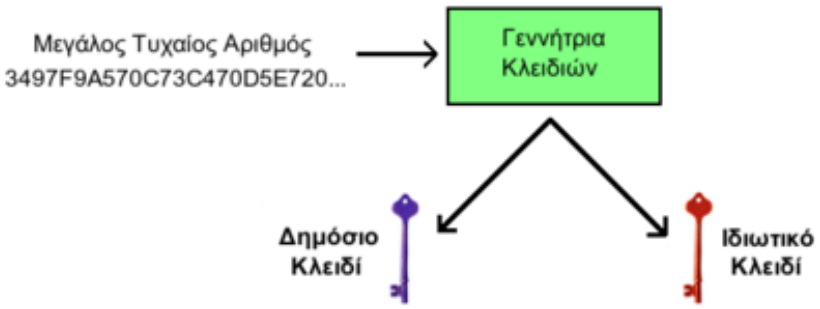
```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP Key Server 0.9.6

mQGiBDmXx8cRBADd0Dko7J7Gb5G/FINwO48AgrlYE87wCT5dlqSXl2uoDmR0/dKp
pJmvDeLQw+Z02yGx7TKf7PC5dfh61tIHyeI0SfCZVA5DtRDk3keNXy2WLnLMg2yS
J44JG3I/o1QKXL8PKD2bkv/vUl7gtXe3qa57oC+2ZmxzptnLeBh08QrWXwCg/8A2
L7WHGKbhYKCApM+0FJStYrcD/1iTYhNsiZnW2tc86e/uHiX8rg7tD7VGm/Wg2E4V
Hadsb4wMLhlfi/vCSEZmlH3hFxvGk6YCWXkdFxzqhq2ZJRWQStqZqjSPWTI3HvOK
Nzjq5DbRMQY2O025g1FyWB62ZDrUWXqzybi4okoEdXT/lQA7Xe95T8uy1zfTUmBg
ieTtA/0RU3MYboVOyDgGvJ7fVyFNdk8+v6Hzcn6EZMWYJifE5hw/tSLnRAXb7ejh
wsLDDCGsjloUj4TnMHH0LUTZSWbgDZwCF6tig3mbhk7YH21zWrVQwPidSpS5O3Oh
85V3nJx5r0406nM4N1cp46yKUMekE6nhubCpVme6o+f9sUMXkLQhR2VydCBLYXNw
ZXJzZW4gPGdla2FzQHdtkGF0YS5jb20+iQBLBBARAgALBQI5l8fHBAsDAgEACgkQ
Ls3rBj/p+6oQGwCgv5ML3xAatvJtY1mKnwz1SH2YbJ8AoPeBkUY73P+QDc5aFdHC
rCkobzlYuQQNBDmXx8cQEAD5GKB+WgZhekOQldwFbIeG7GHszUUfDtjgo3nGydx6
C6zkP+NGlLYwSlPXfAIWSIC1FeUpmamfB3TT/+OhxZYgTphluNgN7hBdq7YXHFHY
UMoiV0MpvpXoVis4eFwL2/hMTdXjqkbM+84X6CqlFGHjhKlP0YOEqHm274+nQ0YI
xswdd1ckOErixPDojhNnl06SE2H22+slDhf99pj3yHx5sHIdOHX79sFzxIMRJitD
YMPj6NYK/aEoJguuqa6zZQ+iAFMBoHzWq6MSHvoPKs4fdIRPyvMX86RA6dfSd7ZC
LQI2wSbLaF6dfJgkCo1+Le3kXXn11JJPmxiO/CqnS3wy9kJXtwh/CBdyorrWqULz
Bej5UxE5T7bxbrlLOCDaAadWoxTpj0BV89AHxstDqZSt90xkhkn4DIO9ZekX1KHT
UPj1WV/cdlJPPT2N286Z4VeSWc39uK50T8X8dryDxUcwYc58yWb/Ffm7/ZFexwGq
01uejaClcjrUGvC/RgBYK+X0iP1YTknbzSC0neSRBzZrM2w4DUUdD3yIsxx8Wy2O
9vPJI8BD8KVbGI2Ou1WMuF040zT9fBdXQ6MdGGzeMyEstSr/POGxKUAYEY18hKcK
ctaGxAMZyAcpesqVDNmWn6vQClCbAkbTCD1mpF1Bn5x8vYlLIhkmuquiXsNV6z3W
FwACAhAAumPF6HT301BhkfuMTVI2jFXUJ7prdWy4pwZArvdDVYQ35M8sG/ISJjwg
B2GdpK9i02B25Cen309snDSj/aJkz7PQgD8CyB1V0KiDFX+KxR8S0le2k1tdbiP3
wNYxw7MDiz757IY9/hmv6YDwSeS2sWnyjgfQXR5z2RBs4r+UZ8YWk8h4YQsLSL2B
Z/PImaiopMScDJy1m4AzasXNdyr1SywU4tlwOXZhZh6ccZB/z6nlZJgMnwfVw1fZ
8wyiTKGoyOp5eh9edDFwtcAVNHVoI0hj9kDfa5sfA9zcKfeLH7TouYLuMcdws9Uc
fNeJI6AgzUvGwzvG9HVuVGf7n5b1j9Kp+jcSjvWqpQX1/iDEN8KG6/yk5mtok4lv
JBieGM2SahP1cTk1P4kcrlwJ419EhKtC6xZS96J0+TmyLBTSrJHazWR+n/lQsXvz
g7wNUycZx+/v1QDGO3HyekqZR76BSMVrpNEYEWcBclTeJ6nBJCPTGxqvP4IQhzhJ
3znANVOsViJZ7rG3DrygE+8vQ32GjbBZzouN/gHxmSKOuvw6yjFdkiSMNNdlIeKm
05Z5aVSXPuE7+TfkbFGHy+GCxjsH6FNhU/8QKtr712NO9laDARGCVaOiVvBVevVO
PXajFSW19F7Rswmh7kD5jUEy9E7ANAa0YD5i07ee0wvmaGSwfKKIRgQYEQIABgUC
OZfHxwAKCRAuzesGP+n7qj2kAJ48xQqu8b8kzQHmmUvMFr8z+bfivQCgxXhBHUT1
LsvbbxbTJ/Da6n5YSYE=
=9v+B
-----END PGP PUBLIC KEY BLOCK-----
```

Σχ. 1.8 – Ένα δημόσιο κλειδί 1024 bits

Η κρυπτογράφηση δημοσίου κλειδιού λύνει ένα σημαντικότατο πρόβλημα που υπήρχε στους κρυπτογραφικούς αλγόριθμους συμμετρικού κλειδιού. Συγκεκριμένα, οι κρυπτογραφικοί αλγόριθμοι συμμετρικού κλειδιού χρησιμοποιούν ένα κοινό μυστικό κλειδί, το οποίο το γνωρίζουν τόσο ο αποστολέας του κρυπτογραφημένου μηνύματος όσο και ο παραλήπτης. Αυτό το κοινό μυστικό κλειδί χρησιμοποιείται κατά την διαδικασία κρυπτογράφησης και αποκρυπτογράφησης του μηνύματος.

- Προκύπτει όμως το εξής πρόβλημα. Εάν υποθέσουμε ότι το κανάλι επικοινωνίας δεν είναι ασφαλές, τότε πως γίνεται ο αποστολέας να στείλει το κλειδί κρυπτογράφησης στον παραλήπτη για να μπορέσει αυτός με την σειρά του να αποκρυπτογραφήσει το μήνυμα; Αυτό το πρόβλημα είναι ιδιαίτερα έντονο στις σύγχρονες ψηφιακές επικοινωνίες όπου σε πολλές περιπτώσεις ο αποστολέας δεν γνωρίζει καν τον παραλήπτη και απέχει από αυτόν αρκετές χιλιάδες χιλιόμετρα. Οι κρυπτογραφικοί αλγόριθμοι δημοσίου κλειδιού λύνουν αυτό το πρόβλημα και ανοίγουν νέους δρόμους για εφαρμογές της κρυπτογράφησης (ηλεκτρονικά μηνύματα κοκ).

Η δημουργία του δημόσιου και του ιδιωτικού κλειδιού γίνεται από ειδικές συναρτήσεις οι οποίες δέχονται ως είσοδο έναν μεγάλο τυχαίο αριθμό και στην έξοδο παράγουν το ζεύγος των κλειδιών. Είναι προφανές ότι όσο πιο τυχαίος είναι ο αριθμός που παρέχεται ως είσοδος στην γεννήτρια κλειδιών τόσο πιο ασφαλή είναι τα κλειδιά που παράγονται.



Σχ. 1.9 – Τρόπος λειτουργίας της γεννήτριας κλειδιών

Σε σύγχρονα προγράμματα κρυπτογράφησης ο τυχαίος αριθμός παράγεται ως εξής.

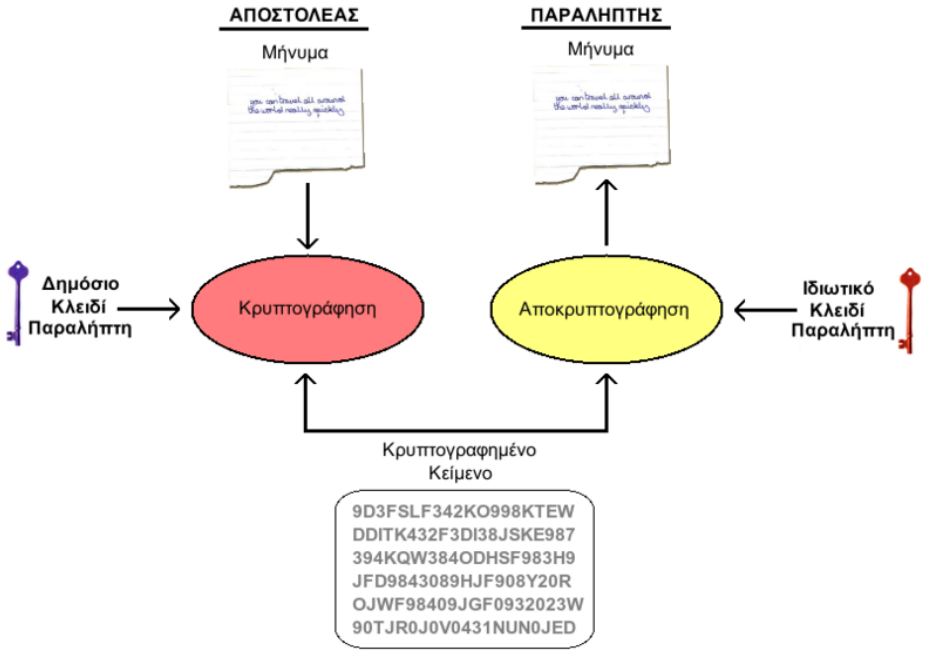
- Κατά την διαδικασία κατασκευής των κλειδιών, το πρόγραμμα σταματάει για 5 λεπτά & καλεί το χρήστη να συνεχίσει να εργάζεται με τον υπολογιστή. Στην συνέχεια για να παράξει τον τυχαίο αριθμό συλλέγει στα 5 αυτά λεπτά τυχαία δεδομένα που εξαρτώνται από την συμπεριφορά του χρήστη (κινήσεις ποντικιού, πλήκτρα του πληκτρολογίου που πατήθηκαν, κύκλοι μηχανής που καταναλώθηκαν κοκ). Με βάση αυτά τα πραγματικά τυχαία δεδομένα υπολογίζεται ο τυχαίος αριθμός και εισάγεται στην γεννήτρια κλειδιών για να κατασκευαστεί το δημόσιο και το ιδιωτικό κλειδί του χρήστη (Σχ 1.9) .

Η **κρυπτογράφηση ασύμμετρου κλειδιού** λοιπόν απαιτεί ένα δημόσιο και ένα προσωπικό κλειδί. Το κλειδί κρυπτογράφησης δημοσιεύεται στο ευρύ κοινό, ενώ αυτό της αποκρυπτογράφησης κρατείται μυστικό (ιδιωτικό κλειδί). Οποιοσδήποτε διαθέτει το δημόσιο κλειδί μπορεί να κρυπτογραφήσει το κείμενο αλλά όχι και να το αποκρυπτογραφήσει.

Το βασικό προτέρημα αυτού του είδους της κρυπτογραφίας (και ο λόγος που αρχικά πολεμήθηκε η διανομή του) είναι ότι επιτρέπει στο ευρύ κοινό να ανταλλάσει μηνύματα με ασφαλή τρόπο. Μέχρι τώρα αυτό ήταν

προνόμιο μόνο των κυβερνήσεων και των μεγάλων οργανισμών που είχαν την οικονομική ευχέρια να αναπτύξουν ασφαλή δίκτυα διανομής των μυστικών κλειδιών τους. Τα δεδομένα κρυπτογραφούνται με το δημόσιο κλειδί του παραλήπτη και αποστέλλονται. Όταν παραληφθούν αποκρυπτογραφούνται με το προσωπικό κλειδί του παραλήπτη. Τα δύο κλειδιά έχουν μαθηματική σχέση μεταξύ τους. Εάν το ένα χρησιμοποιηθεί για την κωδικοποίηση της πληροφορίας, το άλλο θα χρησιμοποιηθεί για την αποκωδικοποίηση τους και αντίστροφα. Το όλο σύστημα βασίζεται στην παραδοχή ότι η γνώση του δημόσιου κλειδιού κρυπτογράφησης δεν επιτρέπει την ανακάλυψη του ιδιωτικού κλειδιού αποκρυπτογράφησης, δηλαδή είναι υπολογιστικά αδύνατο να βρει κανείς το κλειδί της αποκρυπτογράφησης από και τη γνώση και μόνο του κλειδιού κρυπτογράφησης και του αλγόριθμου που χρησιμοποιήθηκε.

Οι κρυπτογραφικοί αλγόριθμοι δημοσίου κλειδιού μπορούν να εγγυηθούν εμπιστευτικότητα (confidentiality), δηλαδή ότι το κρυπτογραφημένο μήνυμα που θα στείλει ο αποστολέας μέσω του διαδικτύου στον παραλήπτη θα είναι αναγνώσιμο από αυτόν και μόνο. Για να επιτευχθεί η εμπιστευτικότητα, ο αποστολέας θα πρέπει να χρησιμοποιήσει το δημόσιο κλειδί του παραλήπτη για να κρυπτογραφήσει το μήνυμα. Στην συνέχεια στέλνει το κρυπτογραφημένο μήνυμα στον παραλήπτη και ο τελευταίος μπορεί να το αποκρυπτογραφήσει με το ιδιωτικό κλειδί του. Δεδομένου ότι το ιδιωτικό κλειδί του παραλήπτη είναι γνωστό μονάχα στον ίδιο και σε κανέναν άλλον, μονάχα ο παραλήπτης μπορεί να αποκρυπτογραφήσει το μήνυμα και να το διαβάσει. Άρα λοιπόν με αυτόν τον τρόπο ο αποστολέας γνωρίζει ότι το κρυπτογραφημένο μήνυμα μπορεί να αποκρυπτογραφηθεί μονάχα από τον παραλήπτη και έτσι διασφαλίζεται η εμπιστευτικότητα του μηνύματος (Σχ. 1.9.1) .

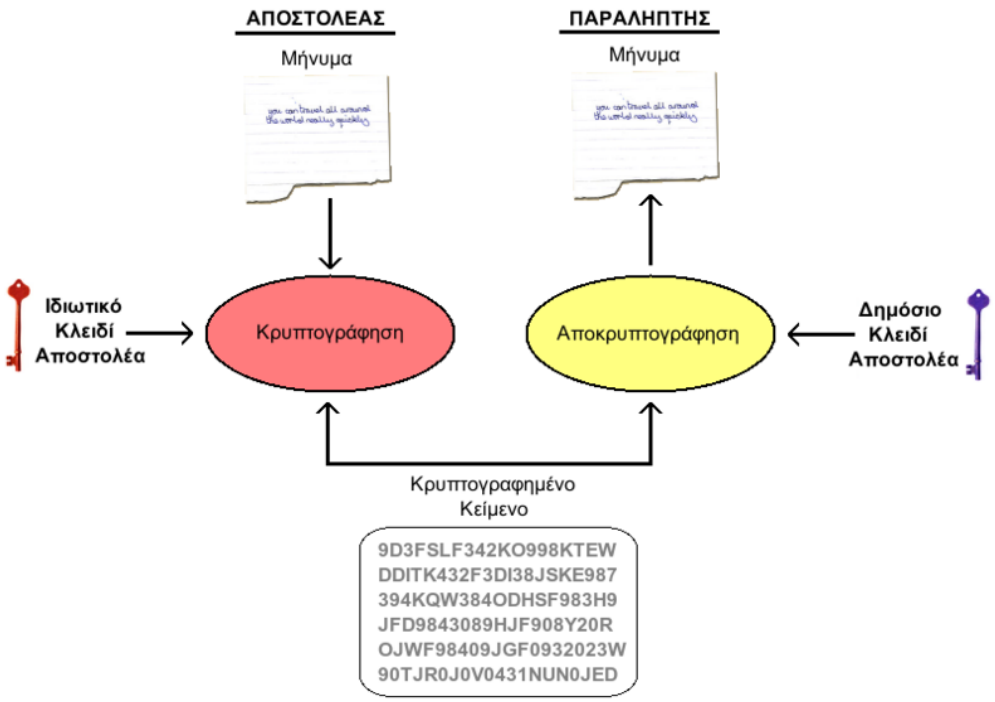


Σχ. 1.9.1 Επίτευξη εμπιστευτικότητας, αλλά όχι πιστοποίησης χρησιμοποιώντας κρυπτογραφικούς αλγόριθμους δημοσίου κλειδιού

Η παραπάνω μέθοδος (Σχ. 1.9.1) μπορεί να εξασφαλίσει την εμπιστευτικότητα, αλλά όχι την πιστοποίηση του αποστολέα. Αυτό με λίγα λόγια σημαίνει πως η παραπάνω μέθοδος δεν μπορεί να εγγυηθεί την ταυτότητα του αποστολέα. Πράγματι, ο αποστολέας μπορεί να δηλώσει ψευδή ταυτότητα και ο παραλήπτης να νομίσει ότι το συγκεκριμένο μήνυμα προήλθε από άλλο πρόσωπο.

Χρησιμοποιώντας κατάλληλα τους κρυπτογραφικούς αλγορίθμους δημοσίου κλειδιού μπορεί να επιτευχθεί πιστοποίηση (**authentication**), δηλαδή ο παραλήπτης να γνωρίζει με ασφάλεια την ταυτότητα του αποστολέα. Για να επιτευχθεί αυτό θα πρέπει ο αποστολέας να χρησιμοποιήσει το ιδιωτικό του κλειδί για την κρυπτογράφηση του μηνύματος. Στην συνέχεια στέλνει το μήνυμα στον παραλήπτη και ο τελευταίος χρησιμοποιεί το δημόσιο κλειδί του αποστολέα για την αποκρυπτογράφησή του. Δεδομένου ότι το ιδιωτικό κλειδί του αποστολέα

είναι γνωστό μονάχα στον ίδιο, ο παραλήπτης μπορεί να είναι σίγουρος για την ταυτότητα του αποστολέα.



Σχ. 1.9.2 Επίτευξη αυθεντικοποίησης, αλλά όχι εμπιστευτικότητας χρησιμοποιώντας κρυπτογραφικούς αλγόριθμους δημοσίου κλειδιού.

Παρόλο που η παραπάνω μέθοδος (Σχ. 1.9.2) εγγυάται την ταυτοποίηση του αποστολέα, δεν δύναται να εγγυηθεί την εμπιστευτικότητα του μηνύματος. Πράγματι, το μήνυμα μπορεί να το αποκρυπτογραφήσει οποιοσδήποτε διαθέτει το δημόσιο κλειδί του αποστολέα. Όπως έχει ήδη ειπωθεί, το δημόσιο κλειδί είναι γνωστό σε όλη την διαδικτυακή κοινότητα, άρα πρακτικά ο οποιοσδήποτε μπορεί να διαβάσει το περιεχόμενο του μηνύματος.

Συνδυάζοντας τις δύο τεχνικές που παρουσιάστηκαν παραπάνω είναι εφικτό να επιτύχουμε εμπιστευτικότητα του μηνύματος και πιστοποίηση του αποστολέα. Δηλαδή αφενός το μήνυμα παραμένει γνωστό μονάχα στον αποστολέα και τον παραλήπτη και αφετέρου ο παραλήπτης γνωρίζει με ασφάλεια ποιος του έστειλε το μήνυμα. Για να επιτευχθεί αυτό ο αποστολέας

μπορεί να κρυπτογραφήσει το μήνυμα πρώτα με το δικό του ιδιωτικό κλειδί και στην συνέχεια με το δημόσιο κλειδί του παραλήπτη. Όταν ο παραλήπτης λάβει το μήνυμα θα πρέπει να χρησιμοποιήσει το ιδιωτικό του κλειδί για να το αποκρυπτογραφήσει (**εμπιστευτικότητα**) και στην συνέχεια να αποκρυπτογραφήσει το αποτέλεσμα χρησιμοποιώντας το δημόσιο κλειδί του αποστολέα (**πιστοποίηση**). Στην συνέχεια αναφέρεται ένα αναλογικό παράδειγμα από την καθημερινή ζωή το οποίο περιγράφει την κρυπτογράφηση συμμετρικού και ασύμμετρου κλειδού.

Έστω λοιπόν ο Α και ο Β, θέλουν να επικοινωνήσουν με ασφάλεια χρησιμοποιώντας το δημόσιο ταχυδρομείο. Ο Α θέλει να στείλει ένα καμουφλαρισμένο-κρυφό μήνυμα στον Β και περιμένει μια καμουφλαρισμένη-κρυφή απάντηση από αυτόν.

- ✓ Σύμφωνα με την ***κρυπτογράφηση συμμετρικού κλειδιού*** ο Α θα βάλει το μήνυμά του μέσα σε ένα κουτί με λουκέτο για το οποίο έχει μόνο αυτός το κλειδί. Στέλνει το κλειδαμπαρωμένο κουτί με το δημόσιο ταχυδρομείο στον Β. Ο Β έχει ένα ίδιο κλειδί (το οποίο έχει πάρει από τον Α στο παρελθόν, σε διαπροσωπική συνάντηση που είχαν) και μόλις λαμβάνει το κουτί, ανοίγει το λουκέτο και διαβάζει το μήνυμα. Ο Β βάζει το μήνυμά του στο κουτί, το κλειδώνει και το στέλνει με δημόσιο ταχυδρομείο στον Α.

  Το πρόβλημα εδώ όπως εύκολα μπορεί καποιος να διαπιστώσει είναι ότι το κλειδί για το λουκέτο είναι κοινό και για τον Α και για τον Β και για να δώσει αντίγραφο του κλειδιού ο ένας με τον άλλον θα πρέπει να συναντηθούν, γιατί δεν είναι ασφαλές να το στείλουν με το δημόσιο ταχυδρομείο (διότι ίσως τότε κάποιος διεφθαρμένος υπάλληλος του ταχυδρομείου, π.χ. ο Γ θα μπορούσε να υποκλέψει το κλειδί και να δημιουργήσει ένα αντίγραφο ώστε στο μέλλον να υποκλέπτει ή να παραποιεί τα μηνύματα που ανταλλάσσονται στο κουτί).

- ✓ Σύμφωνα με την ***κρυπτογράφηση ασύμμετρου κλειδιού (Public Key Cryptography)***, ο Α και ο Β έχουν ξεχωριστές κλειδαριές. Πρώτα ο Α βάζει το μυστικό μήνυμα στο κουτί, το κλειδώνει με το λουκέτο για το οποίο έχει μόνο αυτός το κλειδί και έπειτα στέλνει το κουτί στον Β μέσω απλού δημόσιου ταχυδρομείου. Όταν ο Β λάβει το κουτί, προσθέτει το δικό του λουκέτο στο κουτί και το στέλνει πίσω στον Α. Ο Α λαμβάνει το κουτί με δύο λουκέτα, αφαιρεί το δικό του λουκέτο και το στέλνει πίσω στον Β. Όταν ο Β λάβει το κουτί έχει πάνω μόνο το δικό του λουκέτο, το οποίο μπορεί να ξεκλειδώσει και να δει το μήνυμα του Α. Σε αυτό το παράδειγμα η διαδικασία της αποκρυπτογραφησης είναι ίδια με την διαδικασία της κρυπτογραφησης.

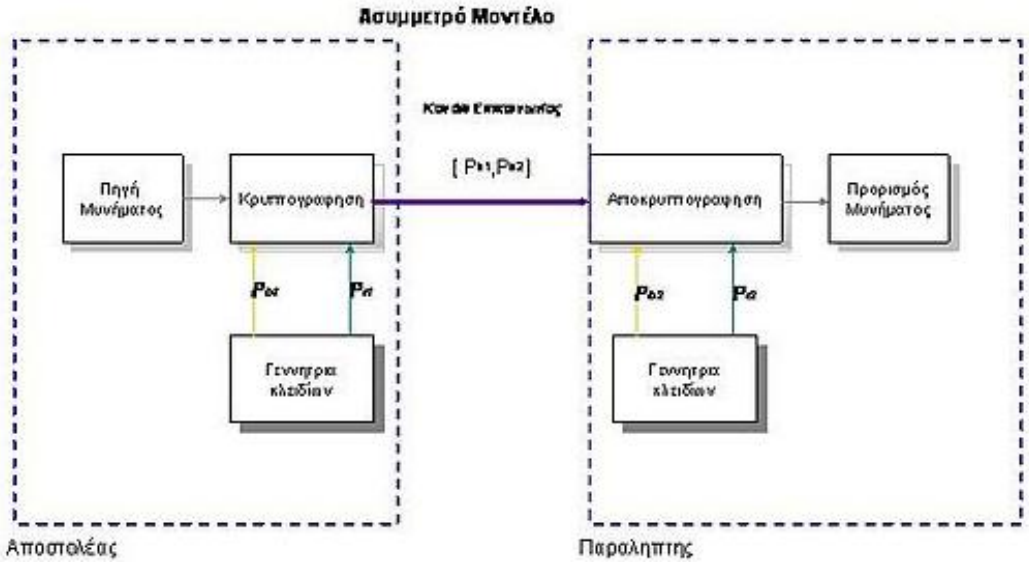
  Η κρίσιμη διαφορά στην κρυπτογράφηση ασύμμετρου κλειδιού είναι ότι η Α και ο Β ποτέ δεν χρειάζεται να στείλουν αντίγραφο του κλειδιού ο ένας στον άλλον. Σε αυτό την περίπτωση αποφεύγουμε την περίπτωση του διεφθαρμένου υπαλληλου στο ταχυδρομείο, του Γ ο οποίος ενδέχεται να υποκλέψει το κλειδί κατά την μεταφορά. Σε αυτή την περίπτωση ο Α και ο Β δεν χρειάζεται να εμπιστευτούν το δημόσιο ταχυδρομείο. Επιπρόσθετα αν για οποιοδήποτε λόγο ο Β επιτρέψει σε κάποιον να αντιγράψει το κλειδί του τότε προφανώς τα μηνύματα μεταξύ των Α και Β θα είναι εκτεθειμένα σε κίνδυνο υποκλοπής. Όμως όλα τα μηνύματα του Α προς άλλους θα παραμένουν μυστικά, αφού οι υπόλοιποι θα παρέχουν διαφορετικά λουκέτα για να κλειδώσει ο Α το μήνυμα στο κουτί πριν το στείλει σε αυτούς.

Πλεονέκτημα της κωδικοποίησης δημόσιου κλειδιού είναι ότι το δημόσιο κλειδί διανέμεται ελεύθερα με αποτέλεσμα την εύκολη σύσταση ασφαλών διαύλων επικοινωνίας μεταξύ δυο απομακρυσμένων χρηστών, χωρίς αυτοί να χρειάζεται να συναντηθούν ή να μεσολαβήσει κάποιο έμπιστο τρίτο μέρος μεταξύ τους.

Συγκρινόμενη με την κωδικοποίηση ιδιωτικού κλειδιού, η κωδικοποίηση δημόσιου κλειδιού απαιτεί περισσότερους υπολογισμούς και για αυτό είναι πιο

αργή. Η ανθεκτικότητα της κωδικοποίησης εξαρτάται περισσότερο από το μέγεθος των κλειδιών που χρησιμοποιούνται παρά από τους αλγόριθμους. Το μέγεθος των κλειδιών μετριέται σε bits. Γενικά κλειδιά μεγάλου μεγέθους παρέχουν ανθεκτικότερη κωδικοποίηση. Για παράδειγμα η κωδικοποίηση 128-bit RC4 είναι 3078 φορές ανθεκτικότερη από την 40-bit RC4. Διαφορετικοί αλγόριθμοι απαιτούν διαφορετικά μήκη κλειδιών για να πετύχουν το ίδιο επίπεδο ανθεκτικότητας κρυπτογράφησης. Για παράδειγμα, ένας αλγόριθμος συμμετρικής κρυπτογράφησης με κλειδί μεγέθους 128 bits παρέχει ανθεκτικότερη κρυπτογράφηση από τον αλγόριθμο κρυπτογράφησης δημόσιου κλειδιού RSA με ίδιο μέγεθος κλειδιού. Για αυτό χρησιμοποιείται κλειδί μεγέθους τουλάχιστον 512 bits προκειμένου η κρυπτογράφηση RSA να θεωρείται ανθεκτική, ενώ οι συμμετρικοί αλγόριθμοι πετυχαίνουν περίπου το ίδιο επίπεδο ανθεκτικότητας με κλειδί μεγέθους 56 bits.

Στα VPNs τα δεδομένα απαιτείται να κωδικοποιούνται σε πραγματικό χρόνο και γι’ αυτό χρησιμοποιούν την κωδικοποίηση ιδιωτικού κλειδιού, το οποίο χρησιμοποιούν μόνο για την συγκεκριμένη συνεδρία. Το μυστικό κλειδί της συνεδρίας κωδικοποιείται χρησιμοποιώντας κωδικοποίηση δημόσιου κλειδιού και στέλνεται μέσω του καναλιού επικοινωνίας. Το ασύμμετρο κρυπτοσύστημα ή κρυπτοσύστημα δημοσίου κλειδιού δημιουργήθηκε για να καλύψει την αδυναμία μεταφοράς κλειδιών που παρουσίαζαν τα συμμετρικά συστήματα.



Σχ. 1.9.3 – Κρυπτοσύστημα Δημοσίου Κλειδιού

Η κρυπτογράφηση δημόσιου κλειδιού μαζί με την συνάρτηση κατατεμαχισμού (hash function) βρίσκει εφαρμογή στις ***ψηφιακές υπογραφές***.

Η **Ψηφιακή Υπογραφή** είναι ένα μαθηματικό σύστημα που χρησιμοποιείται για την απόδειξη της γνησιότητας ενός ψηφιακού μηνύματος ή εγγράφου. Μια έγκυρη ψηφιακή υπογραφή δίνει στον παραλήπτη την πιστοποίηση ότι το μήνυμα που δημιουργήθηκε ανήκει στον αποστολέα που το υπέγραψε ψηφιακά και ότι δεν αλλοιώθηκε-παραποιήθηκε κατά την μεταφορά. Οι ψηφιακές υπογραφές χρησιμοποιούν συνδυασμό μιας κρυπτογραφικής συνάρτησης κατατεμαχισμού (hash function) για δημιουργία της σύνοψης (hash) σε συνδυασμό με ασυμμετρική κρυπτογραφία για κρυπτογράφηση/αποκρυπτογράφηση σύνοψης (***ο συνδυασμός σύνοψης και κρυπτογράφησης με ασυμμετρική κρυπτογραφία αποδεικνύει την ακεραιότητας του εγγράφου αλλά και την απόδειξη ταυτότητας του αποστολέα***).

Σε μερικές χώρες όπως τις ΗΠΑ και κάποιες χώρες της Ευρωπαϊκής ένωσης, οι ψηφιακές υπογραφές έχουν και νομική υπόσταση. Οι ψηφιακές υπογραφές σε ψηφιακά έγγραφα είναι παρόμοιες με τις αντίστοιχες

χειρόγραφες υπογραφές σε έντυπα έγγραφα. Όταν οι ψηφιακές υπογραφές υλοποιούνται - εφαρμόζονται σωστά (με χρήση ασφαλών κρυπτογραφικών αλγορίθμων), είναι πολύ δυσκολότερο να πλαστογραφηθούν σε σχέση με τις αντίστοιχες χειρόγραφες. Επίσης το φυσικό πρόσωπο που ψηφιακά υπογράφει το ψηφιακό έγγραφο δεν μπορεί να ισχυριστεί ότι δεν το υπόγραψε (όσο το ιδιωτικό κλειδί που χρησιμοποίησε δεν υποκλάπηκε). Κάποιες υλοποιήσεις των ψηφιακών υπογραφών προσθέτουν και την ημερομηνία υπογραφής του εγγράφου, ώστε και τον ιδιωτικό κλειδί να υποκλαπεί, η ψηφιακή υπογραφή να είναι έγκυρη. Η ψηφιακή υπογραφή μπορεί να προστεθεί σε οποιαδήποτε σειρά από bits (δηλαδή δεδομένα) πχ μηνύματα ηλεκτρονικού ταχυδρομείου, έγγραφα κλπ. Πολλοί οργανισμοί υιοθετούν την χρήση των ψηφιακών υπογραφών ώστε να αποφεύγεται η αποστολή τυπωμένων εγγράφων (επικυρωμένα με χρήση σφραγίδων και υπογραφών)

Η ψηφιακή υπογραφή αποτελείται από τρεις αλγόριθμους:

- Ο αλγόριθμος δημιουργίας δημόσιου και ιδιωτικού κλειδιού: Ο αλγόριθμος αυτός χρησιμοποιεί μια γεννήτρια τυχαίων αριθμών και με βάση αυτόν τον τυχαίο αριθμό δημιουργεί το δημόσιο και ιδιωτικό κλειδί (με το ιδιωτικό κλειδί δημιουργείται η ψηφιακή υπογραφή και με το δημόσιο κλειδί ελέγχεται η ψηφιακή υπογραφή).

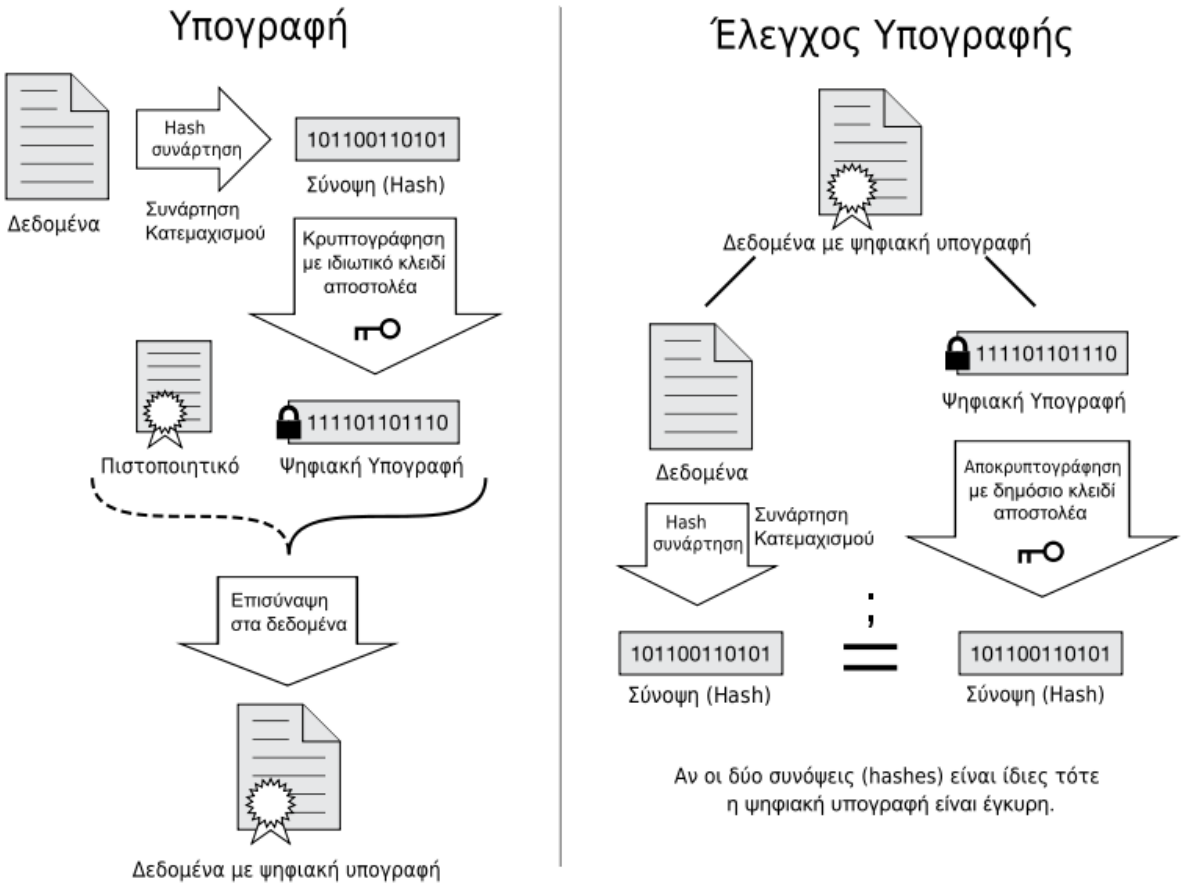
- Ο αλγόριθμος προσθήκης ψηφιακής υπογραφής σε μηνύματα ή έγγραφα: Χρησιμοποιώντας το μήνυμα/έγγραφο και το ιδιωτικό κλειδί (το οποίο ανήκει μόνο σε αυτόν που υπογράφει το έγγραφο), δημιουργεί την ψηφιακή υπογραφή.

- Ο αλγόριθμος έλεγχου ψηφιακής υπογραφής μηνύματος ή εγγράφου: Χρησιμοποιώντας το μήνυμα/έγγραφο και το δημόσιο κλειδί (το δημόσιο κλειδί είναι διαθέσιμο σε όλους, και συσχετίζεται με το

ιδιωτικό κλειδί και ανήκει αυτόν που υπέγραψε ψηφιακά το μήνυμα/έγγραφο), ελέγχει την αυθεντικότητα (ποιος το υπέγραψε) αλλά και ακεραιότητα (ότι το μήνυμα δεν παραποιήθηκε) του μηνύματος/εγγράφου.

Σύμφωνα με την ασυμμετρική κρυπτογράφηση κάποιος που γνωρίζει το δημόσιο κλειδί δεν μπορεί να δημιουργήσει (είναι υπολογιστικά ανέφικτο) το αντίστοιχο ιδιωτικό κλειδί. Επίσης κάποιος ο οποίος έχει το δημόσιο κλειδί μπορεί να ελέγξει την αυθεντικότητα και ακεραιότητα ενός μηνύματος/εγγράφου το οποίο είναι ψηφιακά υπογεγραμμένο.

Ένα πρόβλημα με τις ψηφιακές υπογραφές είναι ότι δεν γνωρίζουμε αν το δημόσιο κλειδί (κατά την διάρκεια ελέγχου της υπογραφής) που έχουμε ανήκει σε αυτόν που ισχυρίζεται ότι είναι. Για αυτό ακριβώς τον λόγο υπάρχει ο ***Πάροχος Υπηρεσιών Πιστοποίησης (CA)*** ο οποίος είναι ένας οργανισμός-οντότητα ο οποίος πιστοποιεί την σχέση ενός ανθρώπου με το δημόσιο κλειδί του. Ο Πάροχος Υπηρεσιών Πιστοποίησης θα πρέπει να εμπνέει εμπιστοσύνη, γιατί είναι η αρχή η οποία εκδίδει ψηφιακά πιστοποιητικά. Τα ψηφιακά πιστοποιητικά ταυτοποιούν ένα δημόσιο κλειδί με τον δικαιούχο του. Πολλές φορές αυτός που υπογράφει ψηφιακά ένα ηλεκτρονικό έγγραφο, ενδέχεται να επισυνάψει στο έγγραφο μαζί με την ψηφιακή υπογραφή και το ψηφιακό πιστοποιητικό του δημόσιου κλειδιού.



Σχ. 1.9.4 Διάγραμμα χρήσης ψηφιακής υπογραφής

Όπως βλέπουμε στο Σχ. 1.9.4 η ψηφιακή υπογραφή είναι η σύνοψη του μηνύματος κωδικοποιημένη με το ιδιωτικό κλειδί του αποστολέα. Μαζί με την ψηφιακή υπογραφή μπορεί να επισυναπτεί και το πιστοποιητικό (από έμπιστη-ο αρχή-οργανισμό) το οποίο πιστοποιεί τον ιδιοκτήτη του δημόσιου κλειδιού (το πιστοποιητικό μπορεί να χρησιμοποιηθεί αργότερα στον έλεγχο της υπογραφής).

Έστω ότι ο Α και ο Β θέλουν να επικοινωνήσουν μεταξύ τους και συγκεκριμένα ο Α θέλει να στείλει στον Β ένα υπογεγραμμένο μήνυμα.

- Αρχικά ο Α και ο Β θα πρέπει να συμφωνήσουν ποιον αλγόριθμο δημόσιου κλειδιού (ασυμμετρικής κρυπτογράφησης: π.χ. PGP, Digital Signature Standard) και ποιον αλγόριθμο κατατεμαχισμού (π.χ. MD5) θα χρησιμοποιήσουν.
- Και ο Α και ο Β έχουν ζευγάρια δημοσίων και ιδιωτικών κλειδιών σύμφωνα με τον αλγόριθμο που επέλεξαν στο προηγούμενο βήμα. Θα πρέπει να ανταλλάξουν μεταξύ τους τα δημόσια κλειδιά τους.
- Ο Α θέλει να στείλει ένα υπογεγραμμένο μήνυμα στον Β. Αρχικά θα περάσει το μήνυμα από τον αλγόριθμο κατατεμαχισμού που επέλεξαν στον πρώτο βήμα και θα παράγει την σύνοψη (digest) του μηνύματος.
- Ο Α θα κρυπτογραφήσει την σύνοψη με το ιδιωτικό κλειδί του και θα προσθέσει την κρυπτογραφημένη εκδοχή του στο τέλος του εγγράφου. Αν θέλει, μπορεί επίσης να προσθέσει και ένα πιστοποιητικό που πιστοποιεί ότι το δημόσιο κλειδί που θα χρησιμοποιηθεί από τον Β αργότερα για την αποκρυπτογράφηση της υπογραφής ανήκει στον Α (το πιστοποιητικό θα πρέπει να έχει εκδοθεί από ένα έμπιστο πάροχο υπηρεσιών πιστοποίησης). Θα αποστείλει στον Β το τελικό έγγραφο (έγγραφο το οποίο έχει ψηφιακά υπογραφεί από τον Α - και ίσως περιέχει και ένα ψηφιακό πιστοποιητικό δημόσιου κλειδιού).
- Ο Β θα ξεχωρίσει την κρυπτογραφημένη σύνοψη από το τέλος του εγγράφου και θα το αποκρυπτογραφήσει χρησιμοποιώντας το δημόσιο κλειδί του Α (το έχει λάβει στον δεύτερο βήμα). Εφόσον η αποκρυπτογράφηση γίνει με επιτυχία γνωρίζει ότι η σύνοψη δεν έχει αλλοιωθεί και ότι ανήκει στον Α. Κατόπιν θα

πάρει το μήνυμα και θα το περάσει από τον αλγόριθμο κατατεμαχισμού που έχει συμφωνήσει στο πρώτο βήμα και θα συγκρίνει την σύνοψη που υπολόγισε ο ίδιος με την σύνοψη που αποκρυπτογράφησε από την ψηφιακή υπογραφή. Αν οι συνόψεις είναι ίδιες, ο Β γνωρίζει ότι το αρχικό μήνυμα δεν έχει αλλοιωθεί. Αν θέλει να βεβαιωθεί ότι το δημόσιο κλειδί που χρησιμοποίησε ανήκει πραγματικά στον Α θα διαβάσει το ψηφιακό πιστοποιητικό του Α.

Η κύρια τεχνική είναι ότι η ψηφιακή υπογραφή είναι η σύνοψη (hash) του μηνύματος κρυπτογραφημένη με το ιδιωτικό κλειδί (χρησιμοποιώντας ασυμμετρική κρυπτογραφία). Υπάρχουν διάφοροι λόγοι που ουσιαστικά εφαρμόζεται η ψηφιακή υπογραφή στην σύνοψη του μηνύματος (hash) και όχι σε ολόκληρο το μήνυμα/έγγραφο:

- Αποτελεσματικότητα (efficiency): Η ψηφιακή υπογραφή είναι πολύ μικρότερη σε μέγεθος και χρειάζεται λιγότερο χρόνος για να εφαρμοστεί η ψηφιακή υπογραφή (σύνοψη/hash) έχει πολύ μικρότερο μέγεθος από ότι ολόκληρο το μήνυμα/έγγραφο).

- Συμβατότητα (compartibility): Τα μηνύματα/έγγραφα είναι ουσιαστικά μεταβλητές δέσμες bits. Ο αλγόριθμος κατατεμαχισμού μπορεί να μετατρέψει μεταβλητού μεγέθους δέσμες bits σε συγκεκριμένο αριθμό bits (σύνοψη - hash).

- Ακεραιότητα (integrity): Αν δεν εφαρμοστεί η συνάρτηση κατατεμαχισμού το αρχικό μήνυμα/έγγραφο θα πρέπει να διαιρεθεί σε μικρότερα μεγέθη bits ώστε ο αλγόριθμος ψηφιακών υπογραφών να εφαρμοστεί σε αυτά. Ο αποδέκτης των πακέτων bits δεν είναι σε θέση να αναγνωρίσει αν όλα τα πακέτα έχουν έρθει και αν βρίσκονται στη σωστή σειρά.

Το επόμενο βήμα στην ασφάλεια των VPNs είναι το secure IP ή όπως συνήθως αναφέρεται IPSec. Το IPSec αποτελείται από μια σειρά προτάσεων που έγιναν από την IETF και περιγράφουν το ασφαλές πρωτόκολλο IP. Αυτές οι προσθήκες προσφέρουν κωδικοποίηση σε επίπεδο IP, σε αντίθεση την κωδικοποίηση σε ανώτερα επίπεδα που το SSL και άλλα πακέτα VPN προσφέρουν. Μια πολύ δημοφιλής εφαρμογή public-key encryption ονομάζεται **Pretty Good Privacy** (PGP), η οποία επιτρέπει να κωδικοποιήσεις σχεδόν τα πάντα.

IPSec – Το Internet Protocol Security Protocol (IPSec) παρέχει εμπλουτισμένα στοιχεία ασφάλειας όπως καλύτερους αλγόριθμους κωδικοποίησης καθώς και πιο εύκολη πιστοποίηση. Το IPSec έχει δύο μεθόδους κωδικοποίησης: **tunnel** και **transport**. Η μέθοδος tunnel κωδικοποιεί την κεφαλή και το ωφέλιμο φορτίο κάθε πακέτου πληροφορίας, ενώ η μέθοδος transport κωδικοποιεί μόνο το πολύτιμο φορτίο. Μόνο τα συστήματα που είναι συμβατά με IPSec μπορούν να εκμεταλλευτούν αυτό το πρωτόκολλο. Ακόμα, όλες οι συσκευές πρέπει να χρησιμοποιούν ένα κοινό κλειδί και τα firewalls του κάθε δικτύου πρέπει να έχει αντίστοιχες τακτικές ασφαλείας.

Το πρωτόκολλο IPSec σχεδιάστηκε να υποστηρίζει δύο λειτουργίες κωδικοποίησης. Η λειτουργία «μεταφοράς» (Transport mode) προστατεύει μόνο το «φορτίο δεδομένων» του κάθε πακέτου, ενώ η λειτουργία «διασωλήνωσης» (tunnel mode) κωδικοποιεί και την κεφαλή, αλλά και το φορτίο δεδομένων του κάθε πακέτου. Όπως λογικά είναι φανερό η λειτουργία «διασωλήνωσης» είναι πιο ασφαλής από την απλούστερη λειτουργία «μεταφοράς», αφού προστατεύει τις ταυτότητες του αποστολέα και του παραλήπτη, καθώς επίσης και άλλα πεδία της διεύθυνσης IP που μπορεί να δώσουν πληροφορίες σε κάποιον «εισβολέα». Απαραίτητη προϋπόθεση για να λειτουργήσει σωστά το πρωτόκολλο IPSec είναι όλες οι επικοινωνούντες συσκευές να μοιράζονται ένα κοινό μυστικό κλειδί. Η διαδικασία ανταλλαγής μυστικών κλειδιών χρησιμοποιεί δημόσια ψηφιακά πιστοποιητικά και βασίζεται στα πρωτόκολλα ISAKMP/Oakley/IKE και στο πρότυπο πιστοποίησης X.509 .

Για να δημιουργηθεί μία διασύνδεση τύπου "tunnel" ανάμεσα σε δύο υπολογιστικά συστήματα χρησιμοποιώντας ένα πρωτόκολλο VPN θα πρέπει και οι δύο εμπλεκόμενες μεριές να έχουν εφαρμόσει παρόμοιες πολιτικές ασφάλειας στο σύστημα. Διαφορετικές αρχιτεκτονικές ασφάλειας θα μπορούσαν να οδηγήσουν σε μία κατάσταση όπου το ένα σύστημα είναι λιγότερο ασφαλές από το άλλο με συνέπεια να είναι και πιο ευάλωτο. Έτσι κάποιος πιθανός «εισβολέας», έχοντας τον έλεγχο του αδύναμου συστήματος θα μπορεί να «επιτεθεί» και στο άλλο σύστημα χρησιμοποιώντας διάφορες τεχνικές τύπου «masquerading».

Tunneling - Η τεχνική του tunneling μας επιτρέπει να τοποθετήσουμε (encapsulate) ένα πακέτο μέσα σε ένα άλλο πακέτο και έτσι να επικοινωνήσουν μεταξύ τους διαφορετικά πρωτόκολλα. Το πακέτο μέσα στο πακέτο μπορεί να είναι του ίδιου πρωτοκόλλου ή και εντελώς διαφορετικών. Έτσι μπορούμε να χρησιμοποιήσουμε αυτήν την τεχνική για να στείλουμε πακέτα τύπου IPX μέσω του διαδικτύου (το οποίο χρησιμοποιεί πρωτόκολλο TCP/IP) έτσι ώστε ο χρήστης να συνδεθεί απομακρυσμένα σε ένα υπολογιστικό σύστημα που χρησιμοποιεί πρωτόκολλο IPX της εταιρίας Novell. Όπως αναφέραμε με την τεχνική αυτή μπορεί κανείς να τοποθετήσει IP πακέτο μέσα σε ένα IP πακέτο. Αυτό πρακτικά σημαίνει ότι μπορούμε να στείλουμε μέσω του διαδικτύου πακέτα με ιδιωτικές διευθύνσεις αποστολέα και παραλήπτη, μέσα σε πακέτα που έχουν δημόσιες (διαδικτυακές) διευθύνσεις αποστολέα και παραλήπτη. Έτσι με αυτόν τον τρόπο μπορούμε να κάνουμε χρήση διευθύνσεων που είναι δεσμευμένες για ιδιωτική χρήση (μόνο για LANs) από την IANA (Internet Assigned Numbers Authority) και να αποκτήσουμε πρόσβαση στο διαδίκτυο.

Τα βασικά δομικά στοιχεία ενός Virtual Private Network είναι :

- **Tunneling** καλείται η διαδικασία με την οποία επιτυγχάνεται η δημιουργία «σήραγγας» (tunnels) για την μετάδοση «πακέτων» (packets) των δεδομένων (data) διαμέσω του Internet.
- **Security** είναι η ασφάλεια που απαιτείται για την προστασία κατά τη μεταφορά αυτών των δεδομένων λόγω της ιδιαιτερότητας του περιβάλλοντος αυτού.
- **Πιστοποίηση** ότι τα δεδομένα έρχονται από την πηγή που διατείνονται,
- **Πρόσβαση** μόνο σε εξουσιοδοτημένους χρήστες ,
- **Εμπιστοσύνη** ότι κανείς δε διαβάζει ή αντιγράφει στοιχεία και
- **Ακεραιότητα** και μη αλλοίωση των δεδομένων κατά τη μεταφορά τους.

Υπηρεσίες ασφάλειας προσφέρονται πλέον σε όλα τα επίπεδα του μοντέλου OSI όπως στα ανώτερα application και session καθώς επίσης στα κατώτερα network και data-link.

2. Ο Τρόπος Λειτουργίας των VPN

Τα VPNs λειτουργούν δημιουργώντας ένα εικονικό tunnel μέσω του δημόσιου Διαδικτύου. Προκειμένου να δημιουργηθεί αυτό το tunnel χρησιμοποιείται συμμετρική κρυπτογράφηση. Οι δύο πλευρές του tunnel μοιράζονται κοινά κλειδιά κρυπτογράφησης/αποκρυπτογράφησης και τα χρησιμοποιούν για να κρυπτογραφήσουν όλη την κυκλοφορία και στις δύο κατευθύνσεις. Η συμμετρική κρυπτογράφηση είναι πολύ γρήγορη και υπάρχουν πολλοί αλγόριθμοι διαθέσιμοι για να την εφαρμόσουν (όπως Blowfish, AES, 3DES).

Υπάρχουν δύο προβλήματα όμως με τη συμμετρική κρυπτογράφηση.

- Κατ' αρχάς, πώς φτάνουν αυτά τα κοινά κλειδιά και στις δύο πλευρές του tunnel (**key exchange** ή key agreement) ;
- Δεύτερον, πώς ξέρουμε ότι ανταλλάσσουμε πληροφορίες με τον σωστό χρήστη στην άλλη μερια του tunnel (**authentication)** ;

Υπάρχουν πολλοί τρόποι να ανταλλαχθούν τα κλειδιά, μερικοί κομψοί και μερικοί όχι και τόσο. Ένας τρόπος για να ανταλλάξεις κλειδιά είναι να καλέσεις τον administraton τηλεφωνικά και να του διαβάσεις το κλειδί από εκεί. Ένας άλλος τρόπος είναι να σταλεί το κλειδί στο mail και να χρησιμοποιηθεί Pretty Good Privacy (PGP) για να κρυπτογραφηθεί η ανταλλαγή. Και οι δύο μέθοδοι θα λειτουργήσουν, αλλά δεν είναι πολύ αποτελεσματικές. Αυτή η διαδικασία αναφέρεται ως pre-shared secret και δεν έχει ούτε την επεκτασιμότητα που θα θέλαμε, αλλά ούτε μας παρέχει την τέλεια μυστικότητα, για την οποία θα μιλήσουμε περισσότερο παρακάτω.

Έτσι λοιπόν εάν θέλουμε να αλλάζουμε τα κλειδιά μας μια φορά την ώρα, ή ακόμα και μια φορά την ημέρα, μπορούμε να δούμε πώς είτε μέσω τηλεφώνου ή ακομα και με την μέθοδο PGP αυτό δεν είναι πραγματικά πρακτικό. Ειδικά εάν υπάρχουν 80 χρήστες VPN με τους οποίους πρέπει να ανταλλάξουμε κλειδιά.

Για να υπερνικήσουμε αυτό το αρκετά δυσκίνητο βασικό ζήτημα ανταλλαγής κλειδιών πολύ συχνά χρησιμοποιούμε πιστοποιητικά (**certificates**). Τα πιστοποιητικά χρησιμοποιούν το Public Key Cryptography, που σημαίνει πως ένας host δημιουργεί ένα ζευγάρι δημοσιων/ιδιωτικών κλειδιών τα οποία συσχετίζονται από μαθηματικής άποψης μεταξύ τους. Οποιαδήποτε στοιχεία κρυπτογραφούνται με το δημόσιο κλειδί (public key) μπορούν να αποκρυπτογραφηθούν μόνο με το ιδιωτικό κλειδί (private key), και αντίστροφα. Κάθε σύστημα έχει το δικό του δημόσιο/ιδιωτικό ζευγάρι κλειδιών.

Το δημόσιο κλειδί δίνεται ελεύθερα στον έξω κόσμο για να κρυπτογραφήσουν την κυκλοφορία που δεσμεύεται για το σύστημα, και το ιδιωτικό κλειδί κρατιέται μυστικό για να αποκρυπτογραφήσει αυτήν την κυκλοφορία. Το ιδιωτικό κλειδί μπορεί επίσης να χρησιμοποιηθεί για να αποδείξει ότι κάτι εστάλει πραγματικά από ένα συγκεκριμένο χρήστη, κάτι το οποίο καλείται non-repudiation. Έτσι λοιπόν εάν κρυπτογραφίσω κάτι με το ιδιωτικό κλειδί μου η άλλη μεριά μπορεί να επιβεβαιώσει ότι είμαι πραγματικά εγώ κάνοντας χρήση του δημοσίου κλειδιού μου για την αποκρυπτογράφηση. Το πρόβλημα με αυτό όμως είναι οτι θα χρειαστεί να έχω ένα αντίγραφο του δημόσιου κλειδιού κάθε host με το οποίο θέλω να συνδεθώ. Οπότε εάν έχω VPN με 100 hosts με τις υπάρχουσες συνθήκες από μόνο του αυτό αποτελεί ένα σοβαρό πρόβλημα εξελιξιμότητας (***scalability problem***) του δικτύου μου.

Η λύση έρχεται με την χρησιμοποίηση μιας ***certificate authority*** (CA). Η αρχή πιστοποιητικών (CA) κοιτάζει παραπέρα από τα πιστοποιητικά που έχει κάποιος και πιστοποιεί ότι είναι αυτός που λέει ότι είναι.

Μόλις γίνει η πιστοποίηση του client, η αρχή πιστοποιητικών (CA) θα υπογράψει το δημόσιο κλειδί του client με το ιδιωτικό κλειδί της. Τώρα, προκειμένου να αποδειχθεί ότι ο client είναι πραγματικά ο client που θέλουμε να μιλήσουμε, πρέπει ακόμα να αποδείξει ότι έχει εγκριθεί από την αρχή πιστοποίησης (CA). Ουσιαστικά είναι σαν να λέμε «εμπιστευόμαστε την αρχη πιστοποίησης και άρα όποιον εμπιστεύεται η αρχή αυτή, θα εμπιστευθούμε και εμείς». Για να αποδειχθεί ότι η CA εμπιστεύεται τον client το μονο που χρειάζεται είναι το δημόσιο κλειδί της αρχής πιστοποίησης. Όταν λοιπόν παίρνουμε ένα πιστοποιητικό (certificate) από τον client αυτόν, πρέπει να φέρει την υπογραφή που έχει δημιουργηθεί από το ιδιωτικό κλειδί της αρχής πιστοποίησης (CA).

Χρησιμοποιούμε λοιπόν το δημόσιο κλειδί της αρχής πιστοποίησης, ώστε να γίνει αποκρυπτογράφηση της υπογραφής και να σιγουρευτούμε ότι το πιστοποιητικό βρίσκεται σε ισχύ. Τώρα μπορούμε να έχουμε 100 hosts στο δίκτυο μας που έχουν όλοι τους προεγκριθεί από την CA. Έτσι λοιπόν

μπορούμε να επικυρώσουμε (authenticate) αυτούς τους hosts απλά με τον έλεγχο υπογραφής της CA στα πιστοποιητικά τους με το δημόσιο κλειδί της αρχής πιστοποίησης. Αυτό λύνει το πρόβλημα εξελιξιμότητάς (scalability problem) του VPN μας.

2.1 SSL / TLS VPN

Το νέο παιδί στην πόλη είναι το VPN βασισμένο στο SSL/TLS. Το SSL υπάρχει από την αρχή της δεκαετίας του '90 και αναπτύχθηκε αρχικά από την Netscape και αργότερα ενώθηκε με ένα αντίστοιχο παρακλάδι της Microsoft. Προς το τέλος της δεκαετίας του '90 το IETF δημιούργησε το TLS σε μια προσπάθεια να παγιωθούν οι διαφορετικοί κλάδοι SSL σε έναν κοινό, ανοιχτό πρότυπο. Το TLS είναι ουσιαστικά το SSLv3 με μερικές δευτερεύουσες βελτιώσεις και ενισχύσεις.

Πρόκειται λοιπόν να εστιάσουμε σε ένα ανοιχτού κώδικα SSL VPN, το οποίο ονομάζεται OpenVPN. Υπάρχουν πολλά εμπορικά προϊόντα διαθέσιμα για να δημιουργήσουν SSL VPNs, αλλά τα πιο πολλά αν όχι όλα δεν διαθέτουν την ευχρηστία που απαιτείται για την δημιουργία ενός site-to-site VPN.

Τα SSL VPNs χρησιμοποιούν την ιδιαίτερα ώριμη και διαδεδομένη υποδομή του SSL/TLS για να χειριστεί την δημιουργία του tunnel αλλα και των κρυπτογραφικών στοιχείων που χρειάζονται ώστε να δημιουργηθεί το VPN. Με το SSL/TLS επιτυγχάνεται λοιπόν η δημιουργία της ίδιας site-to-site λειτουργικότητας που συναντάμε στα IPSec VPNs.

Το OpenVPN αναφέρεται ως user-space VPN, επειδή δεν απαιτεί περίπλοκο συνδυασμό με τον πυρήνα του OS για να λειτουργήσει. Λειτουργεί στο Ring3 του OS Ring Architecture το οποίο ειναι ακριβώς εκεί όπου το θέλουμε. Συνήθως, προκειμένου να γίνει η κρυπτογράφηση μιας σύνδεσης, μια εφαρμογή πρέπει να συνδυαστεί με τον πυρήνα για να παρέχει τη χαμηλού επιπέδου πρόσβαση στο interface όπου βρίσκεται η σύνδεση. Τα

VPNs αυτά χρησιμοποιούν ένα «εικονικό interface» το οποίο ελέγχουν και έχουν πρόσβαση χωρίς την εξάρτηση από τον πυρήνα του OS. Αυτό από μόνο του δίνει στο OpenVPN μια ασφαλέστερη αφετηρία από τις τυποποιημένες συσκευές VPN IPSec, καθώς επίσης και παροχή περισσότερης ευελιξίας σε ότι αφορά το θέμα του porting σε άλλα λειτουργικά συστήματα καθώς επίσης και ευκολία στην εγκατάσταση και συντήρησή του. Η ευελιξία αυτής της αρχιτεκτονικής επιτρέπει ακόμη και συνύπαρξη στο ίδιο μηχάνημα που φιλοξενεί IPSec VPNs. Μπορούμε ακόμα να εγκαταστήσουμε το OpenVPN σε Windows μηχανήματα χωρίς κανένα conflict ανάμεσα σε αυτό και σε κάποιο Windows IPSec Client κάτι το οποιο θεωρείται ένα πολύ θετικό χαρακτηριστικό του OpenVPN. Στην πραγματικότητα, μπορούμε να τρέξουμε ένα IPSec VPN από τα Windows, και ακόμα να έχουμε ένα SSL/TLS VPN να τρέχει συγχρόνως !

Το SSL/TLS είναι ένα τυποποιημένο πρωτόκολλο για την κρυπτογράφηση κίνησης του διαδικτύου. Είναι ένα αρκετά ώριμο πρωτόκολλο το οποίο έχει εφαρμοστεί ευρέως και έχει εξεταστεί για αδυναμίες. Εφ' όσον κανείς δεν μπορεί να υπολογίσει γρήγορα μεγάλους ψευδο-πρώτους αριθμούς, το SSL/TLS φαινεται να είναι αρκετό από άποψη ασφαλειας. Ακόμα είναι σημαντικό να σημειωθεί ότι τα VPNs τα οποία είναι βασισμένα στο SSL/TLS είναι σε θέση να κρυπτογραφήσουν κίνηση για site-to-site connectivity όπως ακριβώς τα IPSec VPNs.

> **ΣΗΜΕΙΩΣΗ**: Ένα πιθανό μειονέκτημα του SSL/TLS έχει να κανει με την απόδοση στα dropped packets. Στην περίπτωση του IPSec αυτό θα επιθεωρήσει και θα απορρίψει ένα πακέτο σε χαμηλότερο επίπεδο του protocol stack απότι το SSL/TLS το οποίο θα το πάρει για να το επεξεργαστεί σε υψηλότερο επίπεδο πρίν το απορρίψει. Αυτό θα μπορούσε να είναι ένα ζήτημα με τις επιθέσεις Denial Of Service σε μερικά σενάρια με αποτέλεσμα να χρησιμοποιούνται αρκετοί πόροι του συστήματος. Παρόλαυτα στις περισσότερες των περιπτώσεων αυτό δεν αποτελεί πρόβλημα.

2.2 Η εφαρμογή OpenVPN

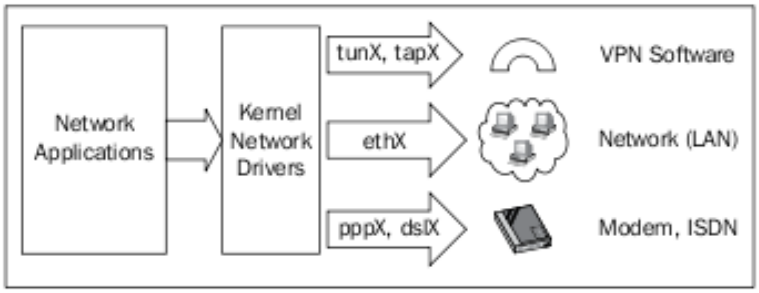
Η εφαρμογή OpenVPN είναι ένα ελεύθερο ανοιχτού κώδικα λογισμικό εικονικών ιδιωτικών δικτύων το οποίο υποστηρίζει τη δημιουργία κρυπτογραφημένων καναλιών (tunnels). Υποστηρίζει συνδέσεις point-to-point ανάμεσα σε 2 ομότιμους εταίρους είτε συνδέσεις πολλαπλών πελατών σε κάποιο εξυπηρετητή.

Το OpenVPN επιτρέπει σε εταίρους να ταυτοποιούνται χρησιμοποιώντας κάποιο κοινό μυστικό κλειδί, ηλεκτρονικά πιστοποιητικά, είτε με το συνδυασμό ονόματος χρήστη και κωδικού. Όταν χρησιμοποιείται σε λειτουργία εξυπηρετητή με πολλαπλούς πελάτες επιτρέπει στον εξυπηρετητή να χρησιμοποιεί πιστοποιητικό για κάθε πελάτη τα οποία παράγονται από βοηθητικά αρχεία του πακέτου με ευκολία.

Τα πλεονεκτήματα της χρήσης ενός κοινού μυστικού κλειδιού είναι η ευκολία στις αρχικές ρυθμίσεις και ότι δε χρειάζεται να διαχειρίζεται και να συντηρεί πιστοποιητικά. Από την άλλη, η λειτουργία αυτή δεν έχει κλιμακωσιμότητα, καθώς υποστηρίζει μονάχα ένα ζεύγος εταίρων, το κάθε κλειδί πρέπει να υπάρχει σε αρχείο τύπου txt σε κάθε μηχάνημα και πρέπει να ανταλλάσσεται με κάποιο τρόπο πριν την έναρξη της λειτουργίας του εικονικού ιδιωτικού δικτύου.

Επίσης, μια σημαντική διαφοροποίηση στη λειτουργία της εφαρμογής είναι η **χρήση δρομολόγησης (routed)** ή **γέφυρας (bridged)**.

Στην πρώτη περίπτωση, χρησιμοποιείται η *συσκευή TAP*, ένα εικονικό interface τύπου Ethernet, ενώ στη δεύτερη περίπτωση χρησιμοποιείται η συσκευή TUN, η οποία δημιουργεί μια ζεύξη σημείου προς σημείου σε επίπεδο IP. Με τη δημιουργία γέφυρας Ethernet, το OpenVPN χρησιμοποιείται για την δημιουργία ενός εικονικού δικτύου Ethernet πάνω από το πρωτόκολλο IP.



Το *TUN device* μπορεί να χρησιμοποιηθεί σαν ένα εικονικό point-to-point interface, όπως ένα modem ή ένα DSL link. Αυτό λοιπόν λέγεται routed mode επειδή προϋποθέτει την ύπαρξη routes ανάμεσα στους χρηστες του VPN. Παρολαυτά το TAP device μπορεί να χρησιμοποιηθεί σαν ένας εικονικός ethernet adapter. Αυτή η δυνατότητα επιτρέπει στον daemon που ακούει στο interface αυτό να κάνει capture ethernet frames, κάτι το οποιο δεν είναι δυνατόν με το TUN interface. Αυτό το mode λέγεται bridging mode, διότι τα δίκτυα συνδεονται σαν να ήταν πάνω σε ένα hardware bridge. Ακόμα οι εφαρμογές έχουν δικαίωμα read/write πάνω σε αυτό το bridge interface.

Το software (δηλαδή ο tunnel driver) θα πάρει τα δεδομένα και θα χρησιμοποιήσει τις βιβλιοθήκες SSL/TLS που είναι υπεύθυνες για την κρυπτογράφηση αυτών και έπειτα τα δεδομένα πακετάρονται και στέλνονται στο άλλο άκρο του tunnel. Το πακετάρισμα θα τελειώσει με τον καθορισμό του είδους των πακέτων (udp/tcp) και του port number. Ο μοναδικός περιορισμός για την επιλογή των παραμέτρων του configuration είναι και τα δυο άκρη του tunnel να έχουν συμφωνήσει σε κοινές παραμέτρους. Το openvpn ακούει τα TUN/TAP devices, παίρνει την κίνηση και την κρυπτογραφεί και έπειτα την στέλνει στο άλλο άκρο του tunnel. Εκεί μια άλλη openvpn process λαμβάνει τα δεδομένα τα αποκρυπτογραφεί και τα παραδίδει στο εικονικό device όπου περιμένει η εφαρμογή.

Στη συνέχεια (κεφάλαιο 3) θα αναλυθεί ο τρόπος λειτουργίας βάσει των ρυθμίσεων που πρέπει να γίνουν στο λογισμικό για υλοποιηση Bridged Interface (TAP driver).

Όταν ένα μηχάνημα/πελάτης συνδέεται με γέφυρα (TAP driver) σε ένα απομακρυσμένο δίκτυο, παίρνει μια διεύθυνση IP από το απομακρυσμένο δίκτυο και συνεπώς είναι σε θέση να επικοινωνεί με τα μηχανήματα αυτά σαν να βρισκόταν στην ίδια τοπική σύνδεση, παρακάμπτοντας firewalls. Στην περίπτωση αυτή, χρειάζεται η χρήση εργαλείου (που είναι διαφορετικό σε κάθε λειτουργικό σύστημα) για τη δημιουργία γέφυρας ανάμεσα στο πραγματικό Ethernet interface του μηχανήματος και το εικονικό Interface TAP.

Σε αντίθετη περίπτωση όταν ένα μηχάνημα συνδέεται στο VPN μέσω δρομολόγησης (TUN driver), χρησιμοποιεί το δικό του ξεχωριστό υποδίκτυο, και κανόνες δρομολόγησης τίθενται και στο μηχάνημα πελάτη και στον εξυπηρετητή προκειμένου να δρομολογούνται τα πακέτα μέσω του VPN.

Τα **πλεονεκτήματα της χρήσης γέφυρας (bridge)** είναι τα εξής:

- Πακέτα τύπου Broadcast διασχίζουν όλο το εικονικό δίκτυο, επιτρέποντας τη χρήση λογισμικού που βασίζεται στην ανταλλαγή μηνυμάτων broadcast στο τοπικό δίκτυο όπως για πχ Lan Games, Windows Netbios για τη κοινή χρήση αρχείων, συσκευών.
- Δε χρειάζεται ρυθμίσεις στη δρομολόγηση.
- Υποστηρίζει οποιοδήποτε πρωτόκολλο λειτουργεί πάνω από το Ethernet και όχι μόνο το IPv4, π.χ. IPv6, Netware IPX και άλλα.

Το κύριο **μειονέκτημα της χρήσης γέφυρας (bridge)** είναι ότι δεν είναι αρκετά κλιμακώσιμη λύση και είναι λιγότερο αποδοτικό από τη χρήση δρομολόγησης. Σε γενικές γραμμές, όταν οι εταίροι που χρειάζεται να συνδεθούν στο εικονικό δίκτυο δεν είναι πολλοί, είναι προτιμότερη η χρήση

γέφυρας, καθώς προσομοιώνει πλήρως ένα τοπικό δίκτυο επιτρέποντας τη χρήση μεγαλύτερου εύρους εφαρμογών.

Ένα πιθανό σενάριο χρήσης του OpenVPN με TAP driver θα μπορούσε να ήταν ανάμεσα σε έναν game server και στους clients του server αυτού. Ας αναλύσουμε την περίπτωση λοιπόν όπου 4 φίλοι θέλουν να παίξουν μεταξύ τους ένα κοινό δικτυακό παιχνίδι στρατηγικής (στο οποιο όμως δεν έχουν τα απαραίτητα κλειδιά για να το κάνουν online). Οποτε αυτός που θα αναλάβει να σηκώσει τον game server θα αναλάβει να σηκώσει και το software VPN σε bridged mode. Όλοι οι φίλοι που θα συνδεθούν στο VPN θα αποκτήσουν lan ips, ώστε να μην χρειάζεται να γίνει verification του cd key στον master game server, αφού όλοι πλέον θα εμφανίζονται σαν να είναι σε lan. Στο συγκεκριμένο παράδειγμα κρίνεται αναγκαία η υλοποίηση με TAP interface, διότι τα lan games βασίζονται στα broadcast πακέτα.

Ένα ακόμα σενάριο θα μπορούσε να ήταν ένας ταξιδιώτης σε ένα τρένο/καράβι που θέλει να σερφάρει με ασφάλεια μέσω της οικιακής του συνδεσης ή ακόμα ένας χρηστης free hot spot που επιθυμεί να συνδεθεί και να σερφάρει ανώνυμα στο διαδίκτυο χωρίς να μπορεί να γίνει sniffing η κίνηση των πακέτων του ή ακόμα και ένας υπάλληλος μιας εταιρίας που επιθυμεί να περάσει κίνηση μέσα από firewalls/proxy στο οποίο έχουν κλήσει τα περισσότερα ports κλπ. Φυσικά όπως πολύ εύκολα μπορεί κάποιος να καταλάβει τα σενάρια χρήσης ενός software VPN είναι αμέτρητα!

3. Εγκατάσταση VPN – ρυθμίσεις

Στη συνέχεια παρουσιάζονται οι ρυθμίσεις και τα πακέτα που απαιτούνται για την εγκατάσταση του OpenVPN σε περιβάλλον Ubuntu Linux. Οι δοκιμές έγιναν στην έκδοση Ubuntu 10.04 LTS.

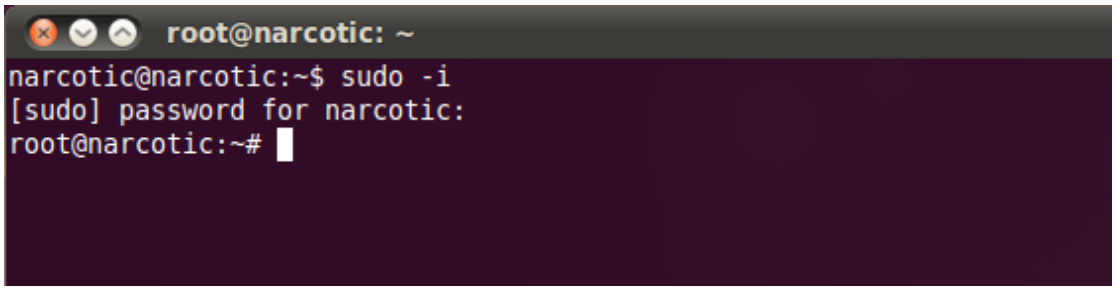
Για τους λόγους που περιγράψαμε στην προηγούμενη ενότητα επιλέχθηκε η χρήση γέφυρας (μέσω του TAP interface) στην παραμετροποίηση και λειτουργία της εφαρμογής. Στη συνέχεια θα παρουσιαστούν οι ρυθμίσεις που χρειάζονται στον εξυπηρετητή (server) και στους πελάτες (clients) του VPN. Στα παρακάτω παραδείγματα, ο server θεωρείται ότι βρίσκεται στο υποδίκτυο 192.168.2.0/24, έχει διαθέσιμη διεύθυνση για τον δαίμονα του OpenVPN την 192.168.2.17 (μέσω της wlan0 ασύρματης κάρτας δικτύου) και το broadcast είναι το 192.168.2.255.

3.1. Εγκατάσταση του Openvpn

Σε κάθε σύστημα ubuntu απαιτείται να τρέξει η παρακάτω εντολή ώστε να αποκτήσουμε πρόσβαση root

sudo –i

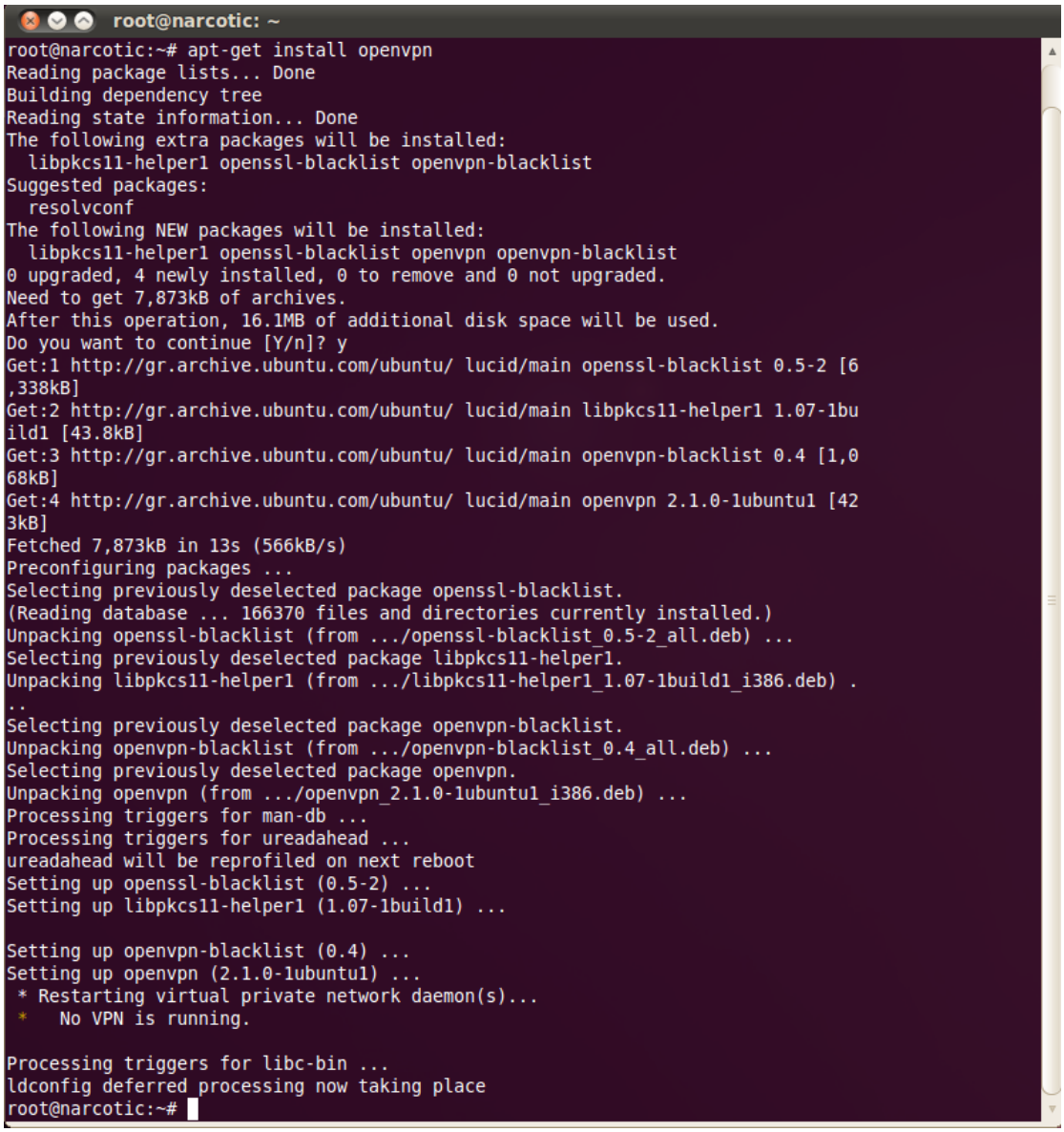
```
narcotic@narcotic:~$ sudo -i
```



Για το OpenVPN χρειαζόμαστε το πακέτο openvpn.

apt-get install openvpn

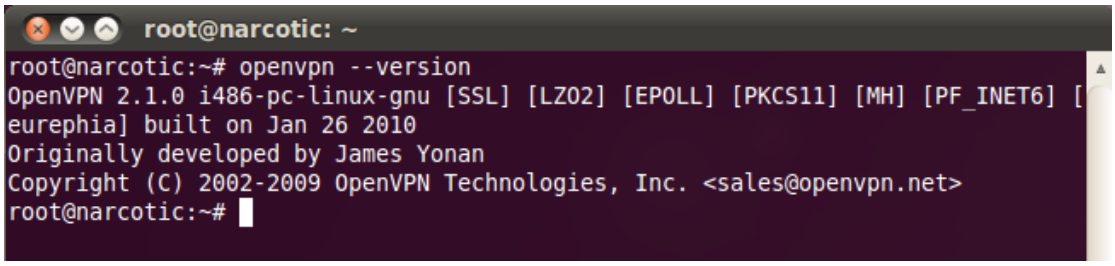
```
root@narcotic:~$ apt-get install openvpn
```



Δίνοντας --version παρατηρούμε τα στοιχεία του πακέτου που μόλις εγκαταστάθηκε.

openvpn –version

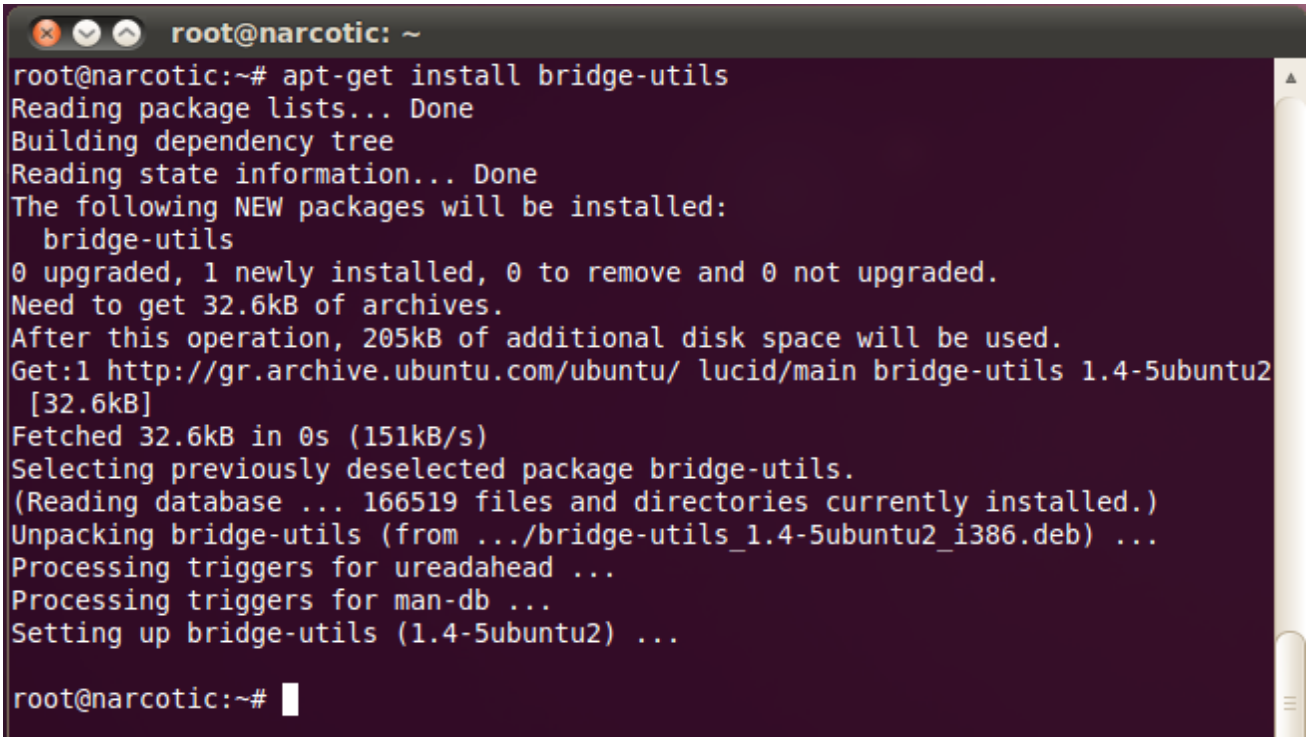
```
root@narcotic:~$ openvpn --version
```



Στην περίπτωση του server θα χρειαστεί και η εγκατάσταση του πακέτου bridge-utils.

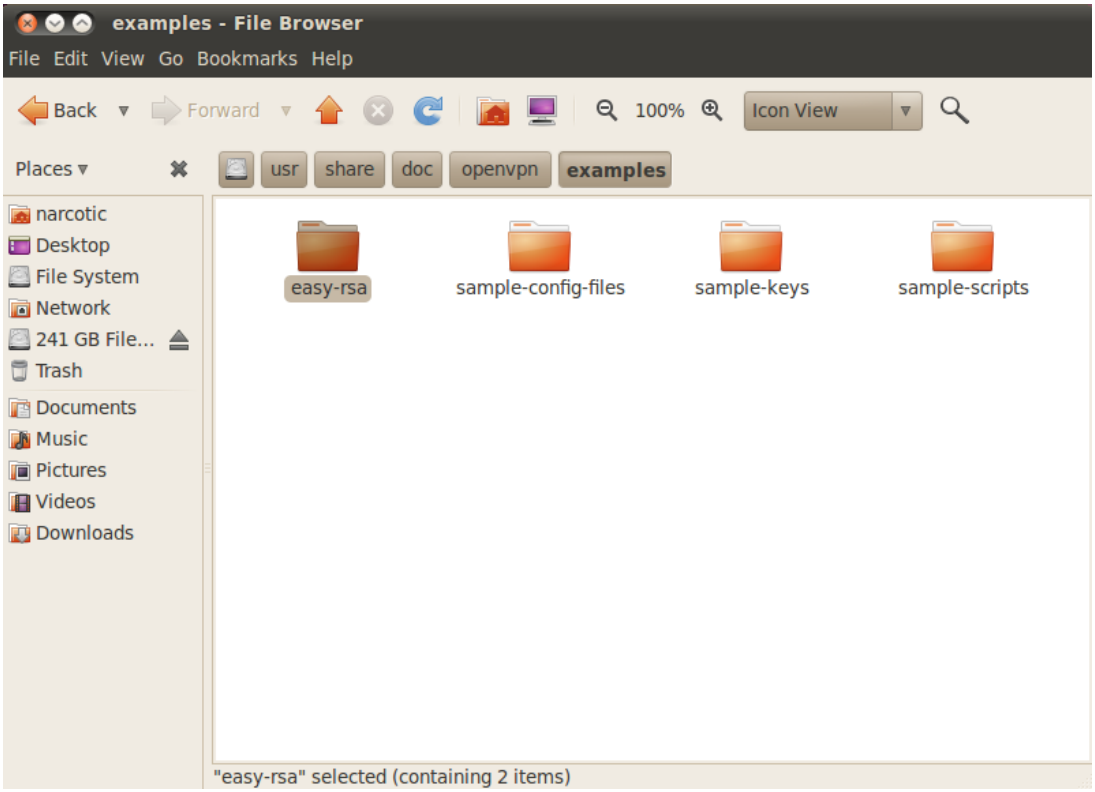
apt-get install bridge-utils

```
root@narcotic:~$ apt-get install bridge-utils
```

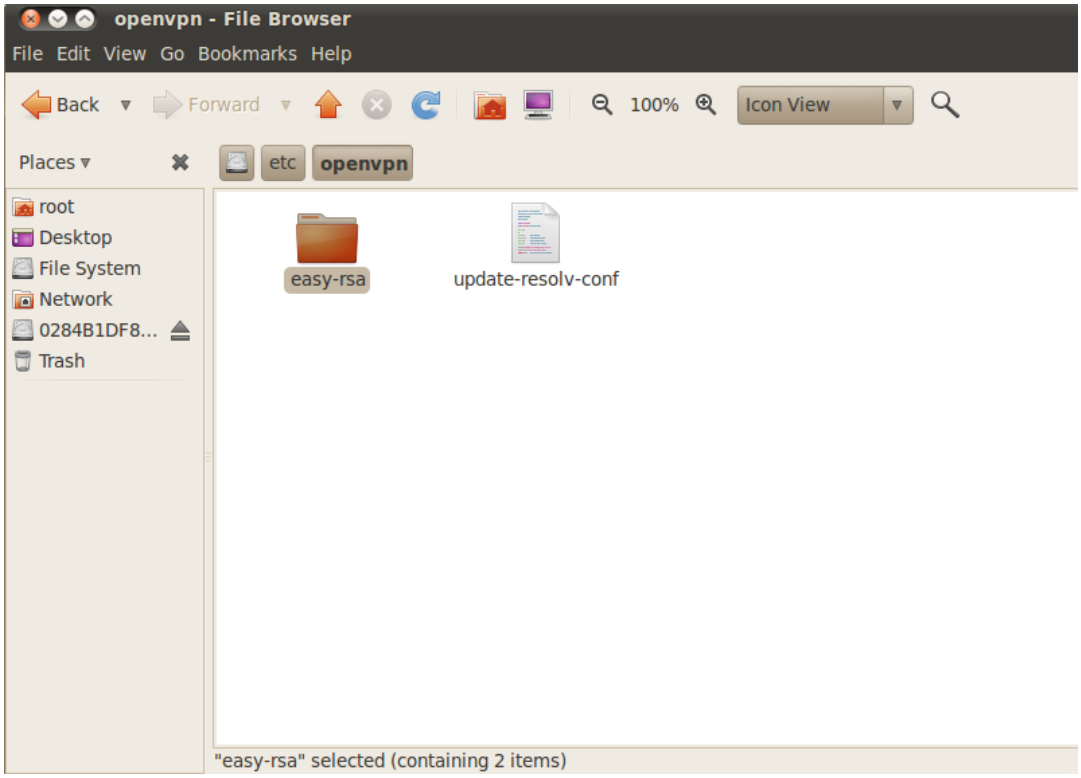


3.2. Δημιουργία πιστοποιητικών

Συνεχίζω την εγκατάσταση αντιγράφοντας τον κατάλογο /easy-rsa απο το **/usr/share/doc/openvpn/examples**/easy-rsa σε ένα νέο φακελο στο **/etc/openvpn**/easy-rsa .



Είναι καλύτερο να γίνει αυτή η αντιγραφή του /easy-rsa από το /usr/.../openvpn/examples/ σε ένα νέο φάκελο πχ όπως έκανα προηγουμένως στο /etc/openvpn πριν γίνει οποιαδήποτε αλλαγή, ώστε μελλοντικά upgrades του πακέτου OpenVPN να μην κάνουν overwrite τις ρυθμίσεις μας. Η αντιγραφή του φακέλου πρέπει να γίνει αφού μπούμε σαν root με sudo nautilus.



Για να μπω σε γραφικό περιβάλλον σαν root πληκτρολογώ στην κονσόλα

sudo nautilus

Έπειτα κάνω παραμετροποίηση σαν root του αρχείου vars στο φάκελο /etc/openvpn/easy-rsa/2.0

```
# In how many days should the root CA key expire?
export CA_EXPIRE=3650
# In how many days should certificates expire?
export KEY_EXPIRE=3650
# These are the default values for fields
# which will be placed in the certificate.
export KEY_COUNTRY="GR"
export KEY_PROVINCE="ATTICA"
export KEY_CITY="ATHENS"
export KEY_ORG="EXAMPLE"
export KEY_EMAIL=me@example.com
```

Έπειτα πατάω αποθήκευση του αρχείου.

3.2.1. Δημιουργία νέας αρχής πιστοποίησης

Μπαίνω από κονσόλα στο φάκελο /etc/openvpn/easy-rsa/2.0 σαν root πάντα (με **sudo -i**) πληκτρολογώντας στην κονσόλα

cd /etc/openvpn/easy-rsa/2.0

```
root@narcotic:~$ cd /etc/openvpn/easy-rsa/2.0
```

Ρυθμιζω που θα ειναι τα scripts

source ./vars

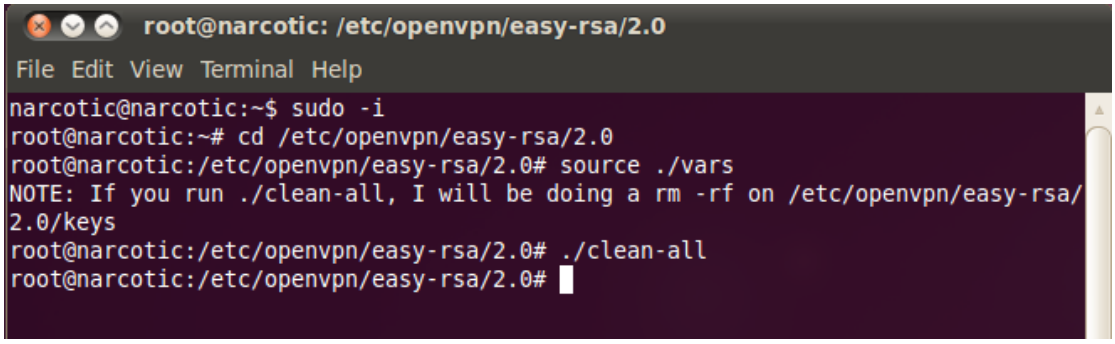
```
root@narcotic:/etc/openvpn/easy-rsa/2.0# source ./vars
```

```
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/etc/openvpn/easy-rsa/2.0/keys
```

Έπειτα σβήνω όλα τα προηγούμενα certificates και keys

./clean-all

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# ./clean-all
```



Για αυτό το βήμα θα χρειαστώ το OpenSSL. Αν δεν υπάρχει ήδη εγκατεστημένο στο server μπορούμε να το εγκαταστήσουμε απλά τρέχοντας

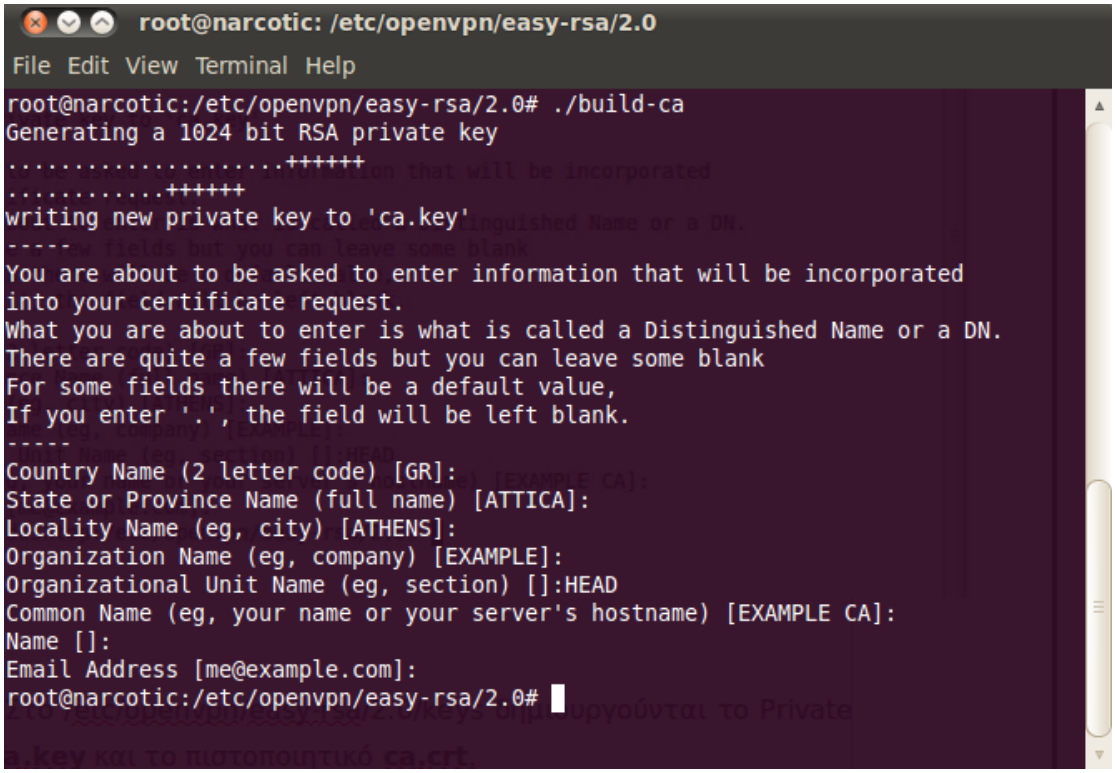
sudo apt-get install openssl

```
root@narcotic:~$ sudo apt-get install openssl
```

Μετά για την δημιουργία της αρχής πιστοποίησης, απλά πληκτρολογώ

./build-ca

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# ./build-ca
```



Στο ***/etc/openvpn/easy-rsa/2.0/keys*** δημιουργούνται το Private key **ca.key** και το πιστοποιητικό **ca.crt**.

Το build-ca.bat script λοιπόν του easy-rsa δημιουργεί ένα certificate file (**ca.crt**) και ενα CA key file (**ca.key**). Το αρχείο ca.crt χρειάζεται από όλα τα

μηχανήματα τα οποια πρόκειται να συνδεθούν στον server, ενώ το αρχείο dh1024.pem πρέπει να βρίσκεται μοναχα στον server.

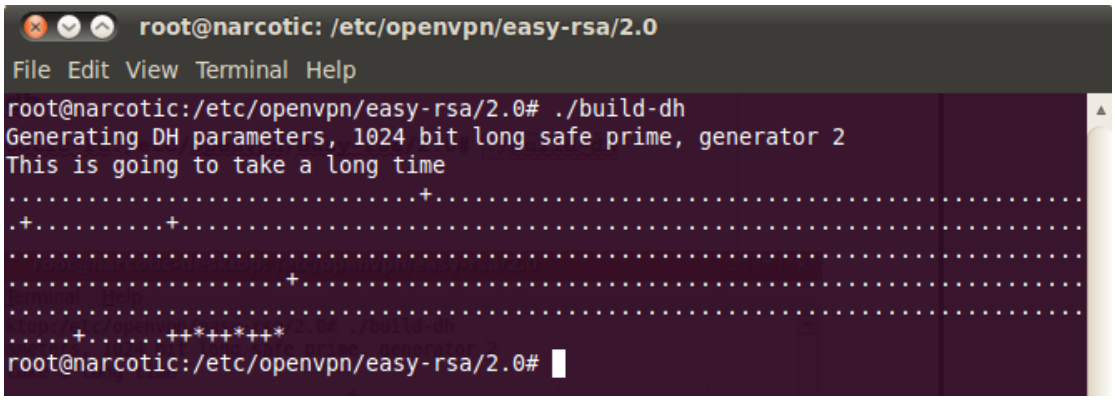
Κάτι ακόμα που πρέπει να δοθεί ιδιαίτερη σημασία είναι ότι οποιος έχει στην διάθεση του το αρχείο ca.key (και ca.crt) είναι σε θέση να κάνει sign requests όπως και η τοπική CA. Για αυτό τα αρχεία αυτά πρέπει να κρατηθούν μυστικά σε ασφαλές μέρος και να μην φύγουν ποτε από τον CA server. Οποτε πρέπει να έχουμε υπόψιν μας ότι το αρχείο αυτό (ca.key) είναι πολύ σημαντικό, διότι είναι το κεντρικό κλειδί του VPN. Αρκετοί ειδικοί συμβουλεύουν για αυτό τον σκοπό να χρησιμοποιηθεί ένα μηχάνημα χωρίς διασύνδεση με το δίκτυο (local login μονο) και αυστηρούς κανόνες για την πρόσβαση σε αυτό.

3.2.2. Δημιουργία παραμέτρων Diffie-Hellman

Η παρακάτω εντολή δημιουργεί το αρχείο dh1024.pem στον server που είναι απαραίτητο για την ασφαλή και επιτυχή λειτουργία του καναλιού SSL.

./build-dh

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# ./build-dh
```



Το **D**iffie-**H**ellman key agreement protocol επιτρέπει σε μια επικοινωνία ανάμεσα σε 2 άκρα να ανταλλάξουν ένα secret key με ασφάλεια, χωρίς να χρειάζεται να προϋπάρχουν ασφαλείς γραμμές. Πρόκειται για να έναν ειδικό μαθηματικό αλγόριθμο ο οποίος εγγυάται ότι μόνο τα 2 άκρα γνωρίζουν το shared key που χρησιμοποιείται.

Το easy-rsa μας παρέχει ένα script (build-dh.bat) το οποίο δημιουργεί το Diffie-Hellman key για μας.

- Το **build-dh.bat** script λοιπόν φτιάχνει ένα dh1024.pem Diffie-Hellman key file, όπου το μέγεθος αυτού του κλειδιού είναι μέρος του ονόματος (αν επομένως χρησιμοποιήσουμε 2048-bit κλειδιά, το αρχείο αυτό θα ονομαστεί dh2048.pem).

Το Diffie-Hellman key agreement protocol (που ονομάζεται ακόμα και *exponential key agreement*) αναπτύχτηκε από τον Diffie και Hellman το 1976 και δημοσιεύτηκε για πρώτη φορά στο paper "*New Directions in Cryptography*". Το protocol αυτό λοιπόν επιτρέπει σε 2 χρήστες να ανταλλάξουν ένα secret key πάνω από ένα μη ασφαλές δίκτυο χωρίς να έχει προηγηθεί μετάδοση κάποιου άλλου μηνύματος.

Η ασφάλεια του πρωτοκόλλου αυτού βασίζεται στο Discrete Logarithm Problem (DLP) και υποθέτει πως είναι υπολογιστικά αδύνατο να υπολογίσεις τα shared secret keys.

Παρολαυτά το Diffie-Hellman key exchange είναι ευάλωτο σε man-in-the-middle επιθέσεις. Σε μια τυχαία επίθεση, ένας πιθανός αντίπαλος ο Γ υποκλέπτει την public value του Α και στέλνει την δικιά του public value στον Β. Όταν μετά ο Β μεταδώσει την δικιά του public value, ο Γ θα την αντικαταστήσει με την δικιά του και θα την αποστείλει στον Α. Έτσι λοιπόν ο Α και ο Γ συμφωνούν σε ένα shared key και ο Γ και ο Β συμφωνούν σε ένα άλλο shared key! Μετά την μετάδοση, ο Γ απλά αποκωδικοποιεί τα μηνύματα που στέλνονται από τον Α ή τον Β και τα διαβάζει ή τα τροποποιεί πριν τα ξανακρυπτογραφήσει με το κατάλληλο key και τα μεταδώσει στο άλλο άκρο.

Αυτή η αδυναμία υπάρχει διότι το Diffie-Hellman key exchange δεν κάνει authenicate σε όσους συμμετέχουν στο VPN. Μια πιθανή λύση θα ήταν η χρήση ψηφιακών υπογραφών και άλλων ρυθμίσεων στο πρωτόκολλο.

Το *authenticated Diffie-Hellman key agreement protocol*, η αλλιώς ***Station-to-Station (STS) protocol***, δημιουργήθηκε από τους Diffie, van Oorschot, και Wiener το 1992 με σκοπό να ξεπεράσει τις επιθέσεις man-in-the-middle στο Diffie-Hellman key agreement protocol. Η ανοσία λοιπόν σε αυτές τις επιθέσεις επιτυγχάνεται επιτρέποντας στα δυο μέλη να κάνουν authenticate μεταξύ τους με την χρήση ψηφιακών signatures και public-key certificates.

Χοντρικά η βασική ιδέα είναι η εξής. Πριν από την εκτέλεση του πρωτοκόλλου τα δυο μέρη ο Α και ο Β αποκτούν ένα ζευγάρι από public/private keys και ένα certificate για το public key τους. Κατά την διαδικασία του πρωτοκόλλου ο Α υπολογίζει το signature σε συγκεκριμένα μηνύματα (την public τιμή) και αντίστοιχα πράττει και ο Β. Με αυτό το τρόπο ενώ ο Γ μπορεί να υποκλέψει μηνύματα ανάμεσα στον Α και τον Β, δεν είναι σε θέση να πλαστογραφήσει τα signatures χωρίς το private key του Α και αντίστοιχα το private key του Β. Με αυτό το τρόπο ξεπερνιέται κάπως το πρόβλημα των επιθέσεων man-in-the-middle.

3.2.3. Δημιουργία πιστοποιητικού για server

Το επόμενο βήμα είναι να προμηθεύσουμε τον VPN Server με ένα certificate και ένα key, το οποιο να έχει γίνει signed από τον CA ή για να είμαστε πιο ακριβείς, θα δημιουργήσουμε ένα certificate request το οποίο θα γίνει signed από τον CA. Ένα unsigned request δεν μπορεί να χρησιμοποιηθεί, όπως ακριβώς ένα διαβατήριο το οποιο είναι μη σφραγισμένο ή ανυπόγραφο από τον τοπικό φορέα δεν μπορεί να χρησιμοποιηθεί. Έτσι λοιπόν δεν έχει χρησιμότητα και ένα unsigned certificate request. Την δουλειά αυτή του signing αναλαμβάνουν τα scripts (batch files) που περιέχει το easy rsa.

Ξεκινώντας το build-key-server script λοιπόν θα δημιουργηθεί ένα 1024-bit private RSA. Υπενθυμίζω ξανά ότι οι τιμές των παραμέτρων στο vars.bat είναι από default ρυθμισμένες και απλά πατώντας enter γίνονται αποδέχτες. Παρολαυτά, στο πεδίο Common Name, πρέπει να είμαστε αρκετά συγκεκριμένοι και να εισάγουμε ένα διακεκριμένο όνομα για τον vpn server. Ακόμα αν χρειαστεί να αυξήσουμε την ασφάλεια του private key μας αρκεί να πάμε στο φάκελο */easy-rsa/vars* και να αλλαξουμε το key size από 1024 σε 2048 bits.

```
#Increase this to 2048 if you are paranoid, this will slow down TLS
#negotiation performance as well as the one-time DH parms
#generation process.
export KEY_SIZE=1024
```

./build-key-server server

```
root@narcotic:/etc/openvpn/easy-rsa/2.0#./build-key-server
server
```

Αφού κάνω τις απαραίτητες ρυθμίσεις στο certificate επιλέγω να γίνει sign και έπειτα commit πληκτρολογώντας ‘y’ όπου ερωτηθώ, ώστε να προστεθούν οι αλλαγές στην βάση μας.

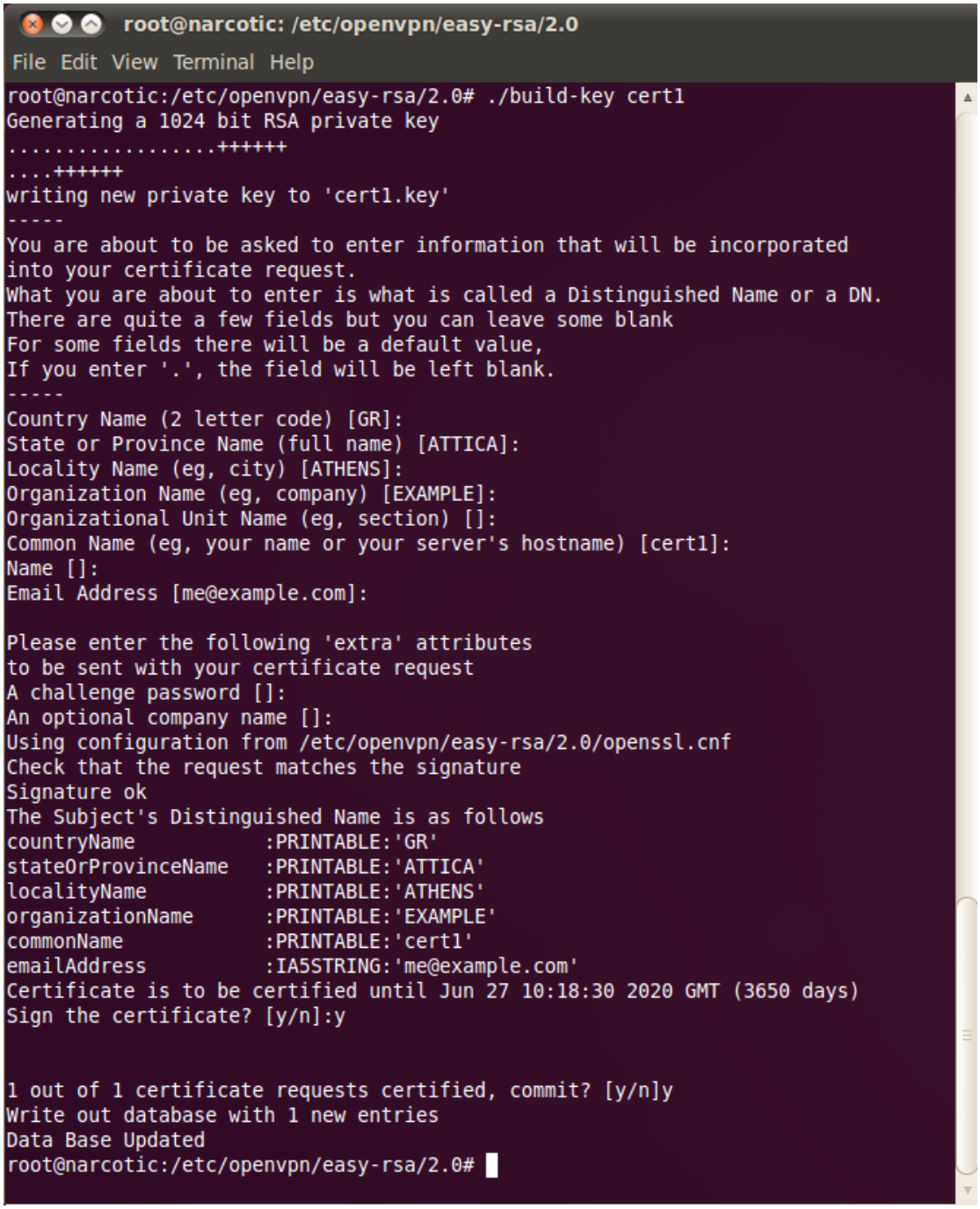
```
      Certificate is to be certified until Sep  9 18:54:58 2019 GMT
(3650 days)
      Sign the certificate? [y/n]:y

      1 out of 1 certificate requests certified, commit? [y/n]:y
      Write out database with 1 new entries
      Data Base Updated
```

3.2.4. Δημιουργία πιστοποιητικού για client

Για την δημιουργία του πιστοποιητικού για τον πρώτο client πληκτρολογώ

./build-key cert1

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# ./build-key cert1
```



Όπως και προηγουμένως αφού κάνω τις απαραίτητες ρυθμίσεις στο certificate επιλέγω να γίνει sign και έπειτα commit πληκτρολογώντας ‘y’ όπου ερωτηθώ.

```
Certificate is to be certified until Sep  9 19:02:55 2019 GMT
(3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
```

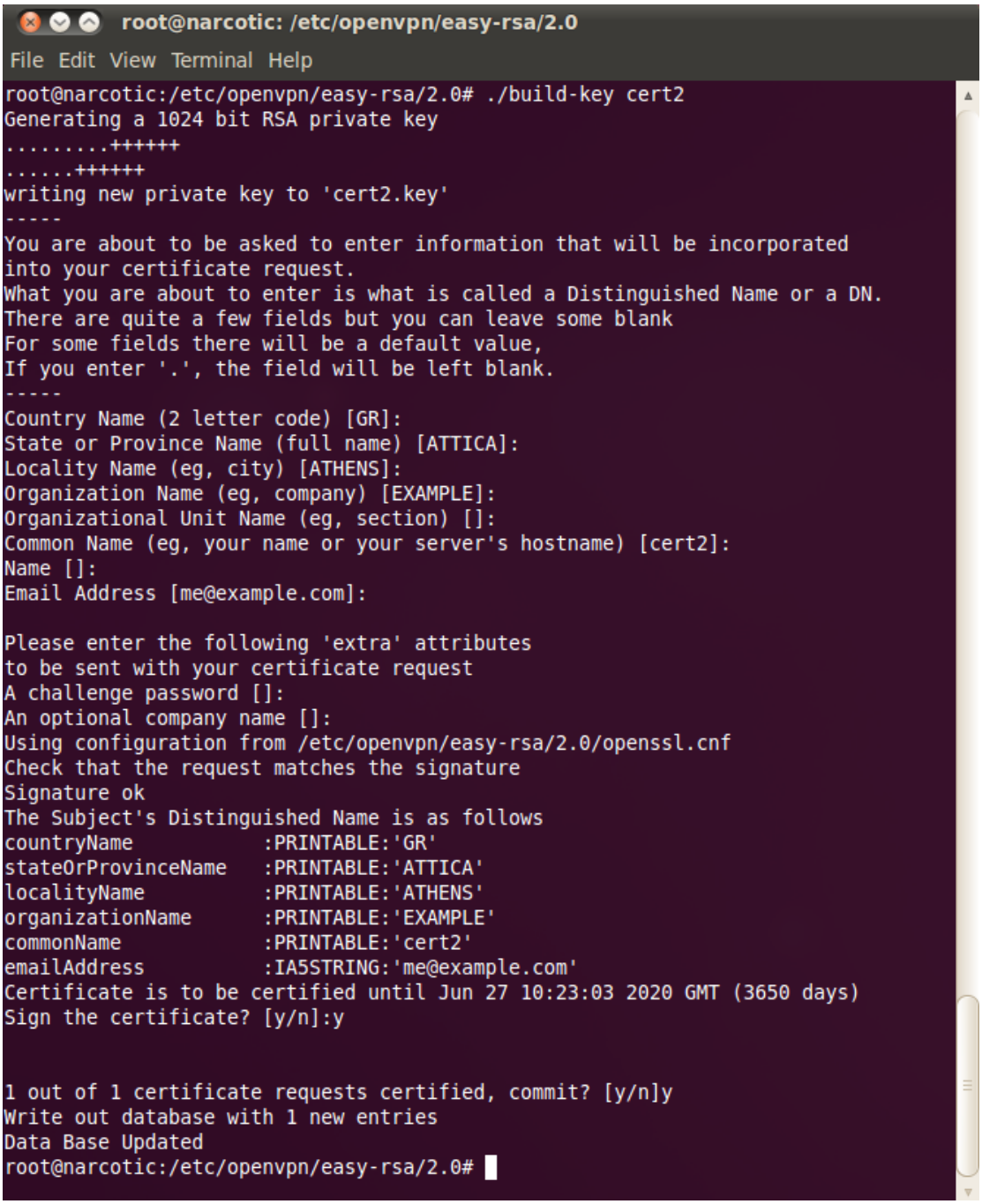
Το ζεύγος cert1.crt και cert1.key πρέπει να σταλούν στον client μέσω ασφαλούς καναλιού. Επειδή η μεταφορά αυτή θα γίνει μια φορά και μέχρι να λήξει ή να καταργηθεί το πιστοποιητικό, μπορούν να μεταφερθούν offline τα πιστοποιητικά στους clients, πχ σε ένα CD το οποίο μετά θα καταστραφεί.

Επαναλαμβάνω την ίδια διαδικασία για δημιουργία περισσοτέρων πιστοποιητικών για τους υπόλοιπους clients (αν υπάρχουν). ΠΧ για το δεύτερο client δημιουργώ το cert2.crt και cert2.key και για το τρίτο client το cert3.crt και cert3.key με τον ίδιο ακριβώς τρόπο όπως και για το πρώτο πιστοποιητικό.

Οπότε για το δεύτερο client έχω

./build-key cert2

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# ./build-key cert2
```



Και για το τρίτο client έχω

./build-key cert3

```
root@narcotic:/etc/openvpn/easy-rsa/2.0#./build-key cert3
```

Τώρα μέσα στο φάκελο /etc/openvpn/easy-rsa/2.0 θα πρέπει να υπάρχουν όλα τα certificates μαζί με τα απαραίτητα κλειδιά. Από την στιγμή που χρησιμοποιούμε ένα Linux μηχάνημα σαν server, είναι καλο να προσαρμόσουμε και τα file permissions πληκτρολογώντας σαν root την εντολή:

chmod go-x /etc/openvpn/keys/*.*

Η εντολή αυτή κάνει τα κλειδιά και τα certificates only readable σαν root.

3.2.5. Ανάκληση πιστοποιητικού για client

Η ανάκληση ενός πιστοποιητικού σημαίνει ακύρωση ενός υπογεγραμμένου πιστοποιητικού, έτσι ώστε να μην μπορεί πλέον να χρησιμοποιηθεί για λόγους επικύρωσης.

Οι χαρακτηριστικοί λόγοι για να ανακαλέσουμε ένα πιστοποιητικό περιλαμβάνονται στα εξής:

- Το ιδιωτικό κλειδί που συνδέεται με το πιστοποιητικό έχει ανακληθεί διότι το κλειδί που συνδέεται με αυτό έχει παραβιαστεί ή κλαπεί με αποτέλεσμα να μην είναι πλέον αξιόπιστο.
- Ο χρήστης ενός κρυπτογραφημένου ιδιωτικού κλειδιού ξεχνά τον κωδικό πρόσβασης στο κλειδί.
- Θέλουμε απλά να τερματίσουμε την πρόσβαση του χρήστη στο VPN .

Ως παράδειγμα, θα ανακαλέσουμε το πιστοποιητικό του client3, το οποίο δημιουργήσαμε προηγουμένως με την ονομασία cert3. Πρώτα λοιπόν ανοίγουμε ένα shell και μπαίνουμε στο φάκελο όπου βρίσκεται ο κατάλογος του /easy-rsa και πληκτρολογούμε

. ./vars
./revoke-full cert3

Έπειτα θα πρέπει να δούμε μια έξοδο που να μοιάζει με το εξής.

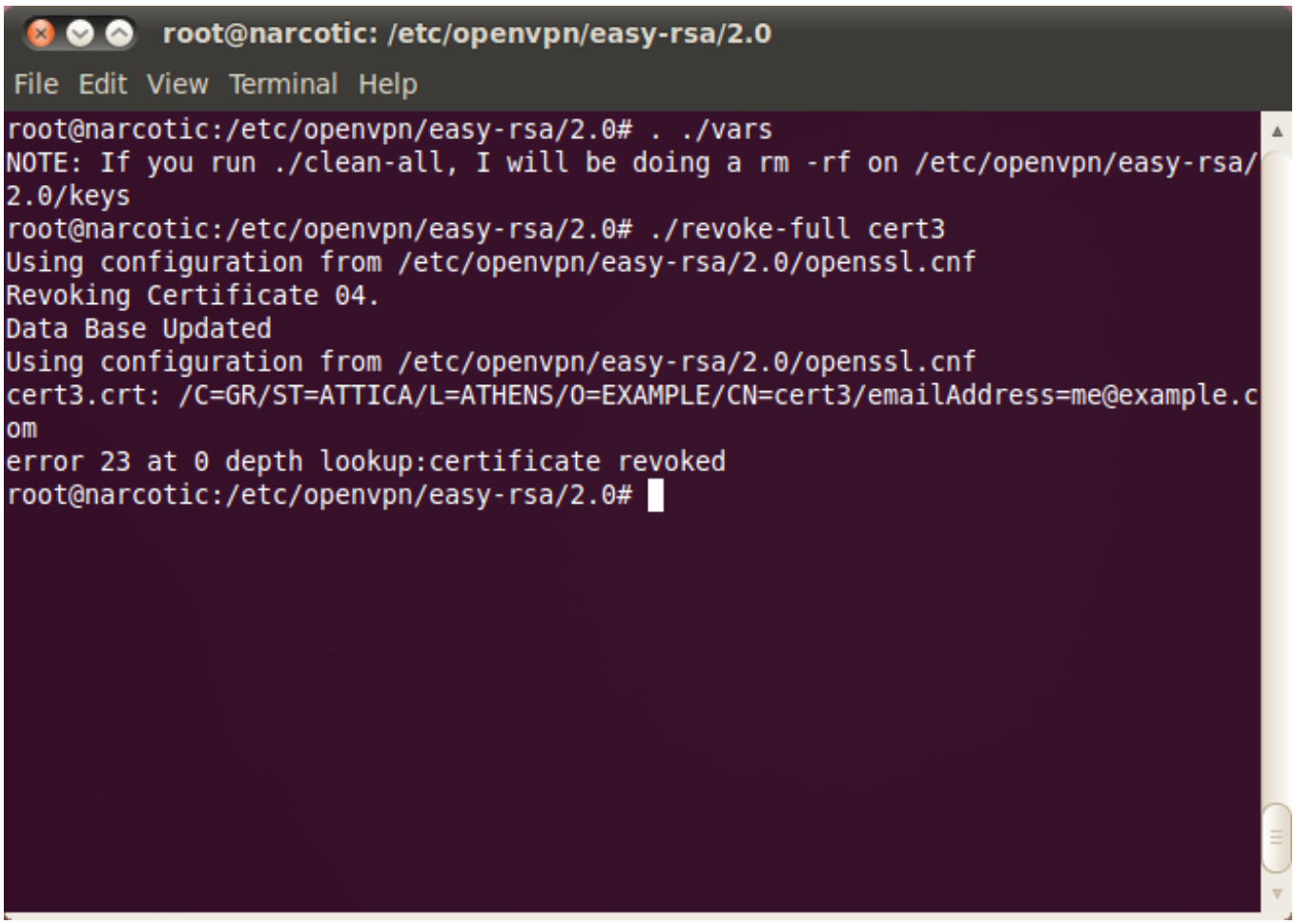
```
Using configuration from
/root/openvpn/2.0/openvpn/tmp/easy-
Rsa/openssl.cnf
DEBUG[load_index]: unique_subject = "yes"
Revoking Certificate 04.
Data Base Updated
Using configuration from
/root/openvpn/20/openvpn/tmp/easy-
rsa/openssl.cnf
DEBUG[load_index]: unique_subject = "yes"
client2.crt: /C=KG/ST=NA/O=OpenVPN-TEST/CN=cert3/
emailAddress=me@myhost.mydomain
error 23 at 0 depth lookup:certificate revoked
```

Παρατηρούμε το "error 23" στην τελευταία γραμμή. Αυτό είναι που θέλουμε να δούμε, διότι δείχνει ότι η επαλήθευση του πιστοποιητικού που ανακαλέσαμε έχει αποτύχει.

Το revoke-full script θα δημιουργήσει στον φάκελο keys ένα αρχείο CRL (certificate revocation list) που ονομάζεται crl.pem. Το αρχείο αυτό πρέπει να αντιγραφεί σε ένα φάκελο όπου ο OpenVPN Server έχει πρόσβαση πχ **`/etc/openvpn/easy-rsa/2.0/keys/`** και έπειτα να προστεθεί και στο configuration του server (**`server.conf`**) ως εξής:

```
crl-verify /etc/openvpn/easy-rsa/2.0/keys/crl.pem
```

Οποτε πλέον όλοι οι clients που συνδέονται στο VPN θα έχουν τα πιστοποιητικά τους από το CRL, με αποτέλεσμα οποιοδήποτε ταύτιση γίνεται να πετάει τον πελάτη αυτόν από το VPN.



Πριν συνεχίσουμε λοιπόν στα επόμενα βήματα για την δημιουργία της γέφυρας και την παραμετροποίηση του server και client θα μείνω λίγο εδώ για να ξεκαθαρίσω ποιος είναι ο σκοπός των αρχείων αυτών που δημιουργήθηκαν καθώς και ποια από αυτά τα αρχεία πρέπει να παραμείνουν όπως και δήποτε κρυφά.

Για να ρυθμίσω λοιπόν το πρώτο client αρκεί να κάνω copy από το μηχάνημα που δημιουργήσαμε τα certificates στο pc του client 1 τα εξής αρχεία.

ca.crt
client1.crt
client1.key
client1.conf

Όπως ξεκάθαρα αναφέρεται και στο πίνακα που ακολουθεί το ca.crt είναι το root certificate το οποίο δεν είναι μυστικό και μοιράζεται στο server και σε όλους τους clients του VPN δικτύου. Το client1.crt είναι το certificate του πρώτου client και το client1.key είναι το αντίστοιχο κλειδί του πρώτου client το οποίο και θα πρέπει να παραμείνει μυστικό. Αντίστοιχα ο client2 θα χρειαστεί το ca.crt, το client2.key και το client2.crt κλπ.

Το πιο σημαντικό αρχείο από όλα είναι αυτό που ονομάζεται ca.key. Αυτό είναι η certificate authority όπως χαρακτηρίζεται, διότι χρησιμοποιείται για να δημιουργήσει νέα κλειδιά για clients που θα συνδεθούν πάνω στο server. Αν για οποιοδήποτε λόγο χαθεί θα είναι αδύνατον νέοι clients να συνδεθούν στο VPN server. Σε ακόμα χειρότερη περίπτωση, αν κλαπεί, θα ήταν δυνατόν ο κλέφτης αυτός να συνδεθεί πάνω στο server. Για αυτό το λόγο θα πρέπει το ca.key να φυλάγεται σε ασφαλή τοποθεσία και αν είναι δυνατόν κάπου που δεν υπάρχει πρόσβαση στο internet.

Τώρα ιδανικά για την μεταφορά των αρχείων από το μηχάνημα που δημιουργήθηκαν στους client θα πρέπει να χρησιμοποιηθεί ένα secure channel, για παράδειγμα ένα scp με RSA authentication.

Σίγουρα είναι λογικό να αναρωτηθεί κάποιος για το αν θα μπορούσε να γίνει η εγκατάσταση του PKI χωρίς να προϋπάρχει κάποιο secure channel. Η απάντηση είναι πως και φυσικά θα μπορούσε. Στο παράδειγμα που ανέφερα προηγουμένως για χάριν συντομίας, δημιούργησα όλα τα private keys στο ίδιο μέρος. Με λίγο προσπάθεια παραπάνω θα μπορούσα να το είχα κάνει διαφορετικά.

Για παράδειγμα, αντί να δημιουργήσω το client certificate και keys στο server, θα μπορούσα να έχω το client να δημιουργήσει τα δικά του κλειδιά τοπικά, και μετά να κάνω ένα **Certificate Signing Request (CSR)** στο key-signing machine. Με την σειρά του το key-signing machine θα μπορούσε να επεξεργαστεί το **CSR** και να μου επιστρέψει ένα signed certificate για τον client. Αυτό θα μπορούσε λοιπόν να γίνει χωρίς να χρειαστεί ποτέ ένα secret.key να φύγει από το σκληρό δίσκο του μηχανήματος που δημιουργήθηκε.

Στο παρακάτω πίνακα αναφέρονται συγκεντρωμένα τα αρχεία του VPN μαζί με το ποιος είναι ο σκοπός του καθενός, ποιος τα χρειάζεται και αν πρέπει να κρατηθούν μυστικά ή όχι.

Filename	Needed By	Purpose	Secret
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman parameters	NO
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES
client2.crt	client2 only	Client2 Certificate	NO
client2.key	client2 only	Client2 Key	YES
client3.crt	client3 only	Client3 Certificate	NO
client3.key	client3 only	Client3 Key	YES

Ανακεφαλαιώνοντας λοιπόν στον server πρέπει να βρίσκονται το server.crt το οποιο είναι το signed certificate του vpn server, το server.key το οποίο είναι το private RSA key του vpn server και το server.csr το οποιο είναι το certificate signing request του vpn server (όπου μπορεί και να διαγραφεί).

Στο πρώτο client χρειάζεται να υπάρχει το client1.crt το οποίο είναι το signed certificate του πρώτου vpn client, το client1.key το οποίο είναι το private RSA key του πρώτου vpn client και το client1.csr το οποίο είναι και το

signing request certificate του πρώτου client (το οποίο μπορεί και να διαγραφεί). Τα αντίστοιχα αρχεία πρέπει να το τοποθετηθούν στους υπολοίπους clients.

Το ca.crt είναι το CA certificate και πρέπει να υπάρχει σε όλα τα μηχανήματα (server / clients).

Το ca.key είναι το κλειδί για τον CA και πρέπει να βρίσκεται μόνο στον CA και να κρατηθεί private, διότι μπορεί να χρησιμοποιηθεί για να κάνει sign valid certificates όπου επιτρέπουν πρόσβαση στις υπηρεσίες και στο δίκτυο μας.

3.3. Δημιουργία γέφυρας

Στα Linux, Windows XP, και Windows 2003, μπορούμε να χρησιμοποιήσουμε VPN tunnels σαν ένα μεγάλο λογικό ethernet δίκτυο. Έτσι λοιπόν συνδέοντας (bridging) ένα εικονικό OpenVPN interface με ένα πραγματικό ethernet interface ουσιαστικά συνδέουμε (bridge) τα δίκτυα πίσω από αυτά τα interface με αποτέλεσμα να φτιάχνουμε ένα εικονικό ethernet δίκτυο ανάμεσα στους hosts του δικτύου όπου μπορεί να γίνεται κανονική ανταλλαγή ethernet frames. Αυτή η δυνατότητα μπορεί να φανεί πάρα πολύ χρήσιμη για χρήστες Windows όπου χρειάζεται να ανταλλάσσουν broadcast πακέτα μέσα από το vpn tunnel πχ για network browsing, lan parties κλπ.

Όταν το σύστημα εκκινήσει και πριν ενεργοποιηθεί ο OpenVPN Server πρέπει να δημιουργηθεί η γέφυρα (bridge).

Για να φτιάξω λοιπόν την γέφυρα θα χρειαστώ δυο script τα οποία και δημιουργώ στην τοποθεσία /etc/openvpn/. Το πρώτο είναι το bridge-start.sh το οποίο θα κάνει το bridge ανάμεσα στο πραγματικό (wlan - 192.168.2.17) του OpenVPN Server και στο εικονικό interface (TAP) το οποιο δημιουργείται από το OpenVPN. Αντίστοιχα το δεύτερο script είναι το bridge-stop.sh το οποίο θα το απενεργοποιεί.

Το περιεχόμενο των scripts ακολουθεί παρακάτω.

Για το **bridge-start.sh** script έχω:

```
#!/bin/bash
################################
# Set up Ethernet bridge on Linux
# Requires bridge-utils
################################
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="wlan0"
eth_ip="192.168.2.17"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.2.255"
for t in $tap; do
   openvpn --mktun --dev $t
done

brctl addbr $br
brctl addif $br $eth
for t in $tap; do
   brctl addif $br $t
done
for t in $tap; do
   ifconfig $t 0.0.0.0 promisc up
done
ifconfig $eth 0.0.0.0 promisc up
ifconfig $br $eth_ip netmask $eth_netmask broadcast $eth_broadcast
```

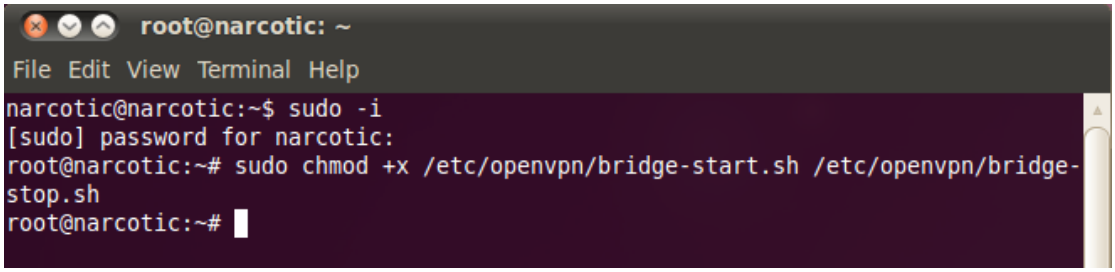
Για το **bridge-stop.sh** script έχω:

```
#!/bin/bash

#####################################
# Tear Down Ethernet bridge on Linux
#####################################

# Define Bridge Interface
br="br0"

# Define list of TAP interfaces to be bridged together
tap="tap0"

ifconfig $br down
brctl delbr $br

for t in $tap; do
    openvpn --rmtun --dev $t
done
```

Έπειτα δίνω και στα δυο script read/write δικαιώματα πληκτρολογώντας στην κονσόλα **sudo -i** για να μπούμε σαν root και μετά

sudo chmod +x /etc/openvpn/bridge-start.sh /etc/openvpn/bridge-stop.sh

```
root@narcotic:/etc/openvpn#    sudo chmod +x
/etc/openvpn/bridge-start.sh /etc/openvpn/bridge-stop.sh
```

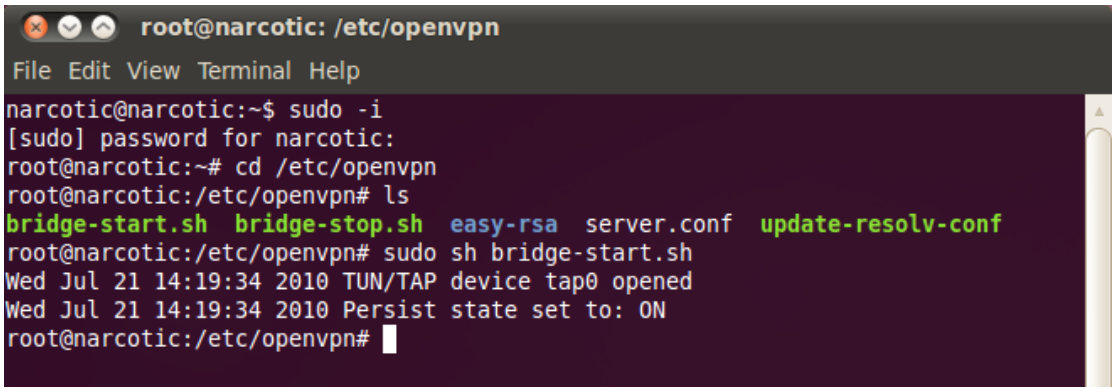


Ουσιαστικά λοιπόν απλά ρυθμίζω τις παραμέτρους br, tap, eth, eth_ip, eth_netmask και eth_broadcast σύμφωνα με το φυσικό interface που θέλω να κάνω bridge. Στα προηγούμενα παραδείγματα του bridge-start και stop script έχω σημειώσει με κόκκινο τα σημεία που χρειάζεται να επέμβουμε και να τροποποιήσουμε κάτι. Για παράδειγμα αν θέλαμε να κάναμε bridge το wired interface (ενσύρματη κάρτα δικτύου – eth0), απλά θα τροποποιούσαμε το eth="eth0" και eth_ip/netmask/broadcast με τα στοιχεία της wired κάρτας. Ακόμα μπορούμε να χρησιμοποιήσουμε την εντολή ifconfig από κονσόλα για να πάρουμε τις απαραίτητες πληροφορίες (για τα network interfaces) όπου χρειάζεται να συμπληρώσουμε στο script "`/etc/openvpn/bridge-start`".

Για να ξεκινήσω λοιπόν το bridge των interface του OpenVPN Server αρκεί να τρέξω το bridge-start script για το οποίο πληκτρολογώ

sudo sh bridge-start.sh

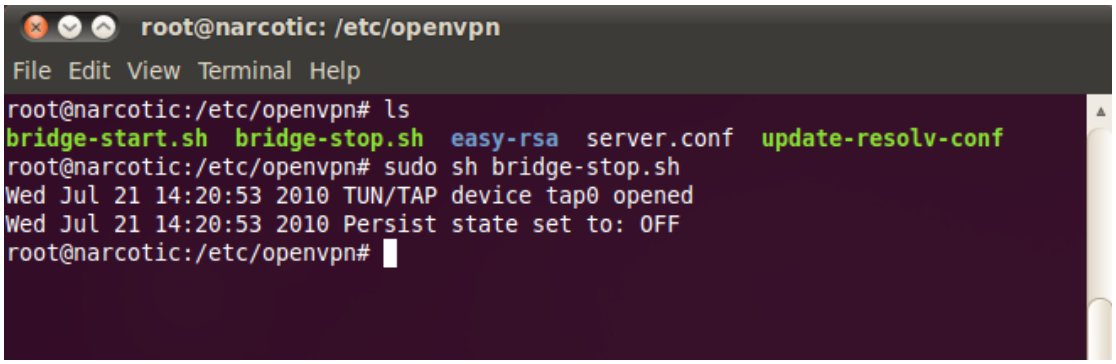
```
root@narcotic:/etc/openvpn# sudo sh bridge-start.sh
```



Για να τερματίσω το bridge των interface αρκεί να τρέξω το bridge-stop script για το οποίο πληκτρολογώ

sudo sh bridge-stop.sh

```
root@narcotic:/etc/openvpn# sudo sh bridge-stop.sh
```



Γενικότερα αρκεί να θυμάμαι ότι ο OpenVPN Server (με TAP interface) πρέπει να ξεκινήσει & να σταματήσει χρησιμοποιώντας την εξής σειρά

- Τρέχω το script **bridge-start**
- Ξεκινάω το openvpn
- Σταματάω το openvpn
- Τρέχω το script **bridge-stop**

Εναλλακτικά για την αυτόματη δημιουργία της γέφυρας κάθε φορά που ξεκινάει ο υπολογιστής αρκεί να προσθέσω στο script όπου τρέχει κατά την εκκίνηση του συστήματος (και βρίσκεται στην τοποθεσία /etc/rc.local) το δικό μας script για το bridge. Οπότε αρκεί να συμπληρώσω στο /rc.local το script bridge-start.sh το οποίο και δίνεται παραπάνω.

Για να προσθέσω λοιπόν boot scripts στην εκκίνηση του συστήματος μου αρκεί να κάνω την εξής διαδικασία. Ενώ είμαι με πρόσβαση root κάνω save το script μου bridge-start.sh στο **/etc/openvpn/easy-rsa/2.0/**bridge-start.sh . Η πρώτη γραμμή του κώδικα του script πρέπει να είναι #!/bin/sh ,

ενώ έπειτα με πρόσβαση πάλι σαν root δίνω στο script execute permission πληκτρολογώντας στο terminal

chmod 0755 /etc/openvpn/easy-rsa/2.0/bridge-start

Οπότε αρκεί να προσθέσω στο /etc/rc.local το scipt μου για το bridge. Έπειτα σαν root επεξεργάζομαι το αρχείο rc.local στο /etc/rc.local και προσθέτω την τοποθεσία του script μαζί με ένα "exit 0" στο τέλος όπως περιγράφω από κάτω.

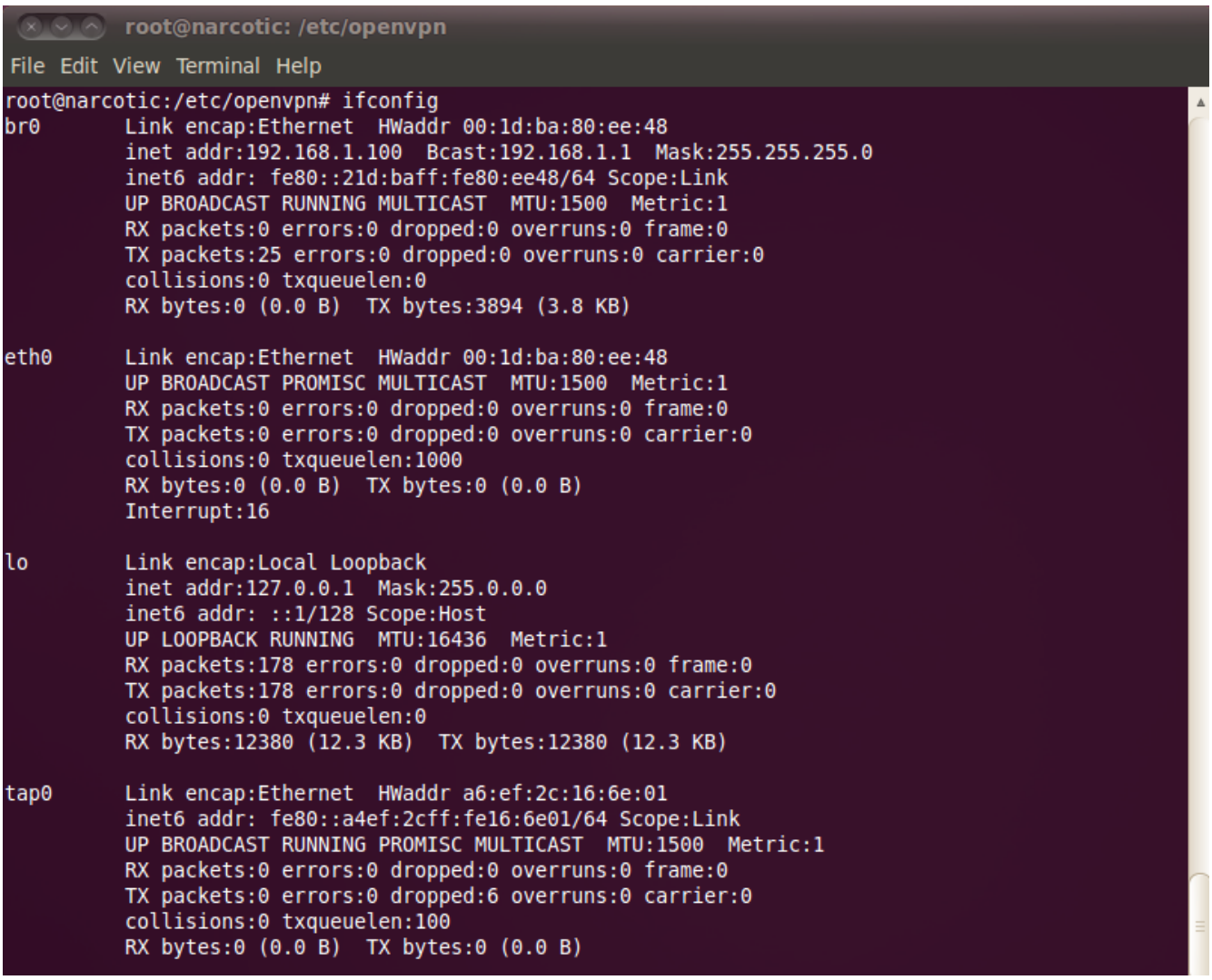
#Περιεχόμενα /etc/rc.local
Το script αυτό ξεκινάει το bridge
/etc/openvpn/easy-rsa/2.0/bridge-start.sh
exit 0

Τέλος απλά κάνω ένα reboot για να πάρει τις αλλαγές.

3.4. Ρυθμίσεις στο Linux Firewall

Εξαρχής παρατηρώ τα δυο νέα interface **tap0** και **br0** που έχουν δημιουργηθεί.

root@narcotic:/etc/openvpn# **ifconfig**

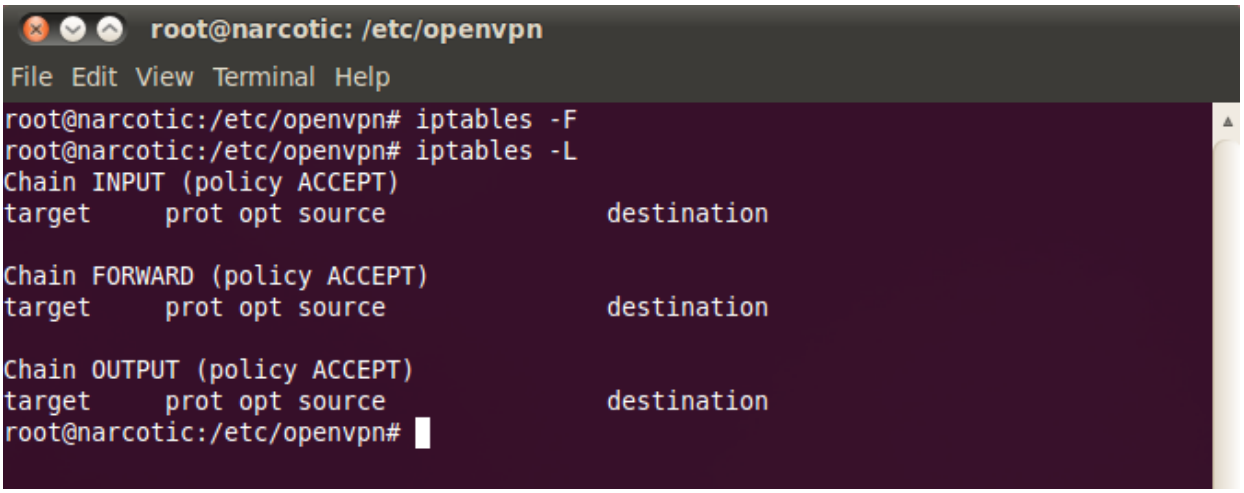


Έπειτα θα πρέπει να ρυθμίσω το Linux firewall (iptables), να επιτρέπει στα πακέτα να μετακινούνται ελεύθερα πάνω στα δυο αυτά νέα interfaces που δημιουργήσαμε **tap0** kai **br0**.

Για να γίνει αυτό αφού αποκτήσω από κονσόλα (sudo -i) πρόσβαση root κάνω flush ότι ρυθμίσεις (rules) προϋπάρχουν στα iptables με την εντολή iptables -F και έπειτα χρησιμοποιώ την εντολή iptables -L για να επιβεβαιώσω ότι όλα τα policy των iptables έχουν σβηστεί.

root@narcotic:/etc/openvpn# **iptables -F**

root@narcotic:/etc/openvpn# **iptables -L**



Έπειτα για να ρυθμίσω τα policy των iptables σωστά, ώστε το linux firewall να επιτρέπει στα πακέτα να μετακινούνται ελεύθερα πάνω στα tap0 και br0 τις εξής εντολές

iptables -A INPUT -i tap0 -j ACCEPT

root@narcotic:/etc/openvpn# iptables -A INPUT -i tap0 -j ACCEPT

iptables -A INPUT -i br0 -j ACCEPT

root@narcotic:/etc/openvpn# iptables -A INPUT -i br0 -j ACCEPT

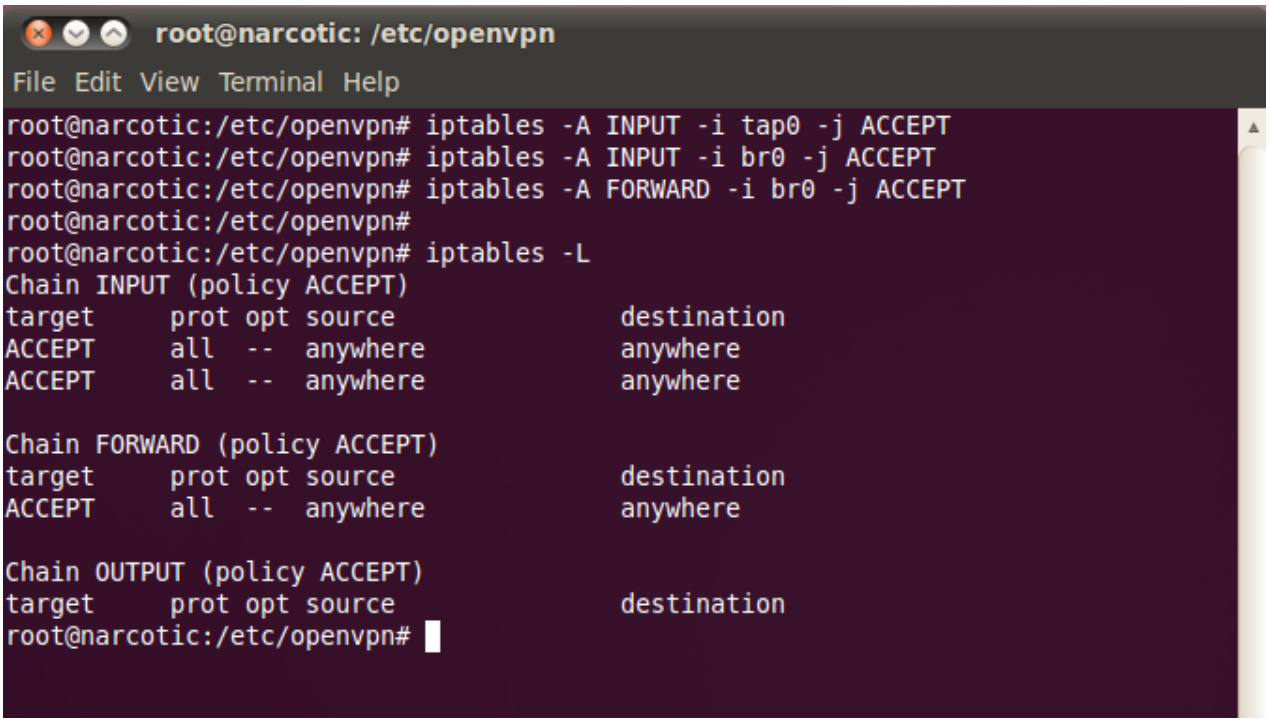
iptables -A FORWARD -i br0 -j ACCEPT

root@narcotic:/etc/openvpn# iptables -A FORWARD -i br0 -j ACCEPT

και για να δω πλέον τις νέες αλλαγές αρκεί να πληκτρολογίσω

iptables -L

root@narcotic:/etc/openvpn# **iptables -L**



3.5. Παραμετροποίηση Server

Στη συνέχεια παρατίθενται μέρη από τη παραμετροποίηση του server και του client. Το πλήρες αρχείο ρυθμίσεων του server server.conf παρατίθεται στο παράρτημα.

Οι παρακάτω εντολές δηλώνουν που ακούει ο δαίμονας και το είδος της σύνδεσης (TAP)

```
local 192.168.2.17
port 1194
proto udp
dev tap0
```

Έπειτα παρατίθενται τα αρχεία στα οποία βρίσκεται το πιστοποιητικό της αρχής πιστοποίησης, το πιστοποιητικό και το ιδιωτικό κλειδί του server:

```
ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt
cert /etc/openvpn/easy-rsa/2.0/keys/server.crt
key /etc/openvpn/easy-rsa/2.0/keys/server.key
dh /etc/openvpn/easy-rsa/2.0/keys/dh1024.pem
```

Οι παρακάτω εντολές δίνουν το Pool των διευθύνσεων που είναι διαθέσιμες στους πελάτες, και περνάνε σε κάθε πελάτη την εντολή δρομολόγησης για την πρόσβαση στο τοπικό υποδίκτυο του server.

```
server-bridge 192.168.2.1 255.255.255.0 192.168.2.200
192.168.2.202
push "route 192.168.2.0 255.255.255.0"
```

Για περισσότερη ασφάλεια από αυτή που προσφέρει το SSL/TLS, δημιουργούμε ένα "HMAC firewall" το οποίο θα μας βοηθήσει να εμποδίσουμε DoS attacks και UDP port flooding. Για να δημιουργήσω το ta.key αρκεί να πληκτρολογήσω από κονσόλα

openvpn –genkey –secret ta.key

```
root@narcotic:/etc/openvpn/easy-rsa/2.0# openvpn --genkey --secret ta.key
root@narcotic:/etc/openvpn/easy-rsa/2.0#
```

Ο server και οι clients πρέπει να έχουν ένα αντίγραφο από αυτό το κλειδί (ta.key) και αντίστοιχη εγγραφη στο configuration τους. Η δεύτερη παράμετρος που θα πρέπει να ρυθμιστεί είναι πως τελειώνει η εντολή αυτή "0" για το server και "1" για τους clients. Έτσι λοιπόν για το configuration του server αρκεί να προσθέσω

```
tls-auth /etc/openvpn/easy-rsa/2.0/keys/ta.key 0
# This file is secret
```

Η παρακάτω εντολή επιτρέπει την επικοινωνία ανάμεσα σε clients που δεν επιτρέπεται εξ ορισμού.

```
client-to-client
```

Μέγιστος αριθμός πελατών που επιτρέπεται να συνδεθούν στον server

```
max-clients 3
```

Ενεργοποίηση συμπίεσης με τη χρήση της βιβλιοθήκης lzo.

```
comp-lzo
```

3.6. Παραμετροποίηση Client

Ακολουθεί το αρχείο ρυθμίσεων του client. Δηλώνονται το είδος της σύνδεσης, udp με tap interface, η διεύθυνση και η πόρτα όπου ακούει ο OpenVPN server, το πιστοποιητικό, το ιδιωτικό κλειδί και η αρχή πιστοποίησης για την δημιουργία του καναλιού SSL, και η χρήση συμπίεσης με τη βιβλιοθήκη lzo. Στο remote βάζουμε την εξωτερική ip του VPN Server (αυτή δηλαδή που βγαίνει στο net) και διπλα το port. Προσέξουμε πολύ ώστε ότι ρυθμίσεις έχουμε κάνει στο server configuration να υπάρχουν οι αντίστοιχες και στο client.

Τέλος, προαιρετικά, μπορεί να δρομολογείται ολόκληρη η κίνηση και όχι μόνο προς το υποδίκτυο 192.168.2.0/24 αν βγάλει το σχόλιο «;» από την τελευταία γραμμή redirect-gateway του αρχείου ρυθμίσεων.

```
client
dev tap0

proto udp
remote 79.166.232.155 1194

resolv-retry 0
nobind
persist-key
persist-tun
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/cert1.crt
key /etc/openvpn/keys/cert1.key
tls-auth /etc/openvpn/keys/ta.key 1
ns-cert-type server
comp-lzo
verb 3
;redirect-gateway
```

Για να ρυθμίσω το openvpn ώστε όταν ξεκινάει κάθε φορά να τρέχει το config της επιλογής μου αρκεί να προσθέσω στο /etc/default/openvpn (βάζοντας σχόλιο «;» σε όλες τις άλλες γραμμές) το εξής

AUTOSTART="client1"

Αυτή η γραμμή λέει στο OpenVPN πιο configuration file (στην συγκεκριμένη περίπτωση το client1.conf) πρέπει να είναι το default που θα τρέξει όταν ξεκινάει το πρόγραμμα. Τα configuration files είναι μέσα στο /etc/openvpn . Έτσι εκτος από το παραδοσιακό τρόπο για να τρέξω manual ένα configuration

sudo openvpn --config /etc/openvpn/client1.conf

μπορώ να χρησιμοποιήσω εναλλακτικά και αυτόν τον τρόπο μέσω του autostart τρέχοντας την εντολή

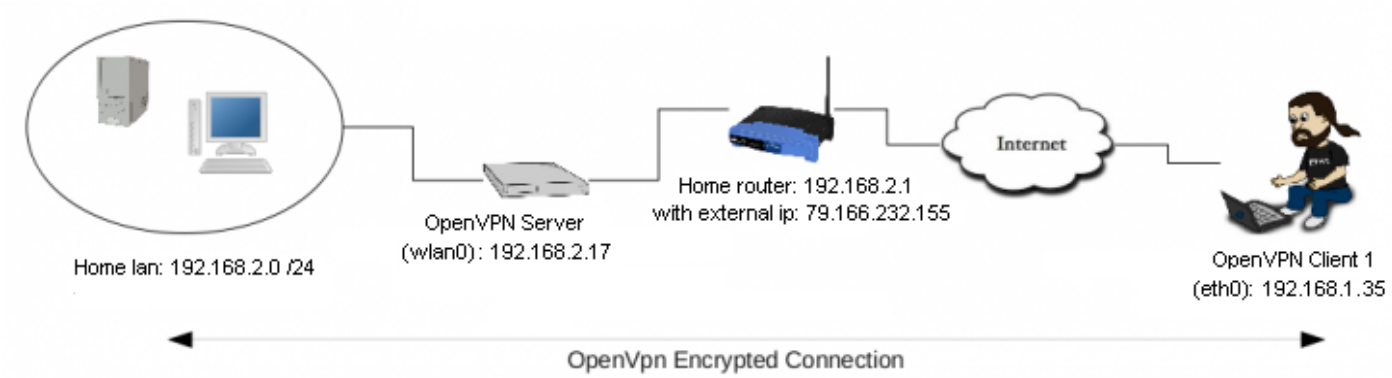
/etc/init.d/openvpn start

Αντίστοιχα για να σταματήσω με τον τρόπο αυτό το VPN αρκεί να πληκτρολογήσω

/etc/init.d/openvpn stop

4. Υλοποίηση του VPN

Όσον αφορά το πρακτικό κομμάτι λοιπόν της πτυχιακής θα προσπαθήσω να περιγράψω όσο καλύτερα μπορώ το δίκτυο πάνω στο οποιο δοκιμάστηκε η υλοποίηση αυτή του software VPN. Για αυτό το σκοπό έφτιαξα μια γραφική αναπαράσταση του πως έχει δομηθεί το δίκτυο της δοκιμης και πιο κάτω έχω συνεχίσει εξηγώντας επιμέρους πράγματα, ώστε να γίνομαι αρκετά κατανοητός σε όσους δεν διαθέτουν την απαιτούμενη εξοικείωση με το αντικείμενο.



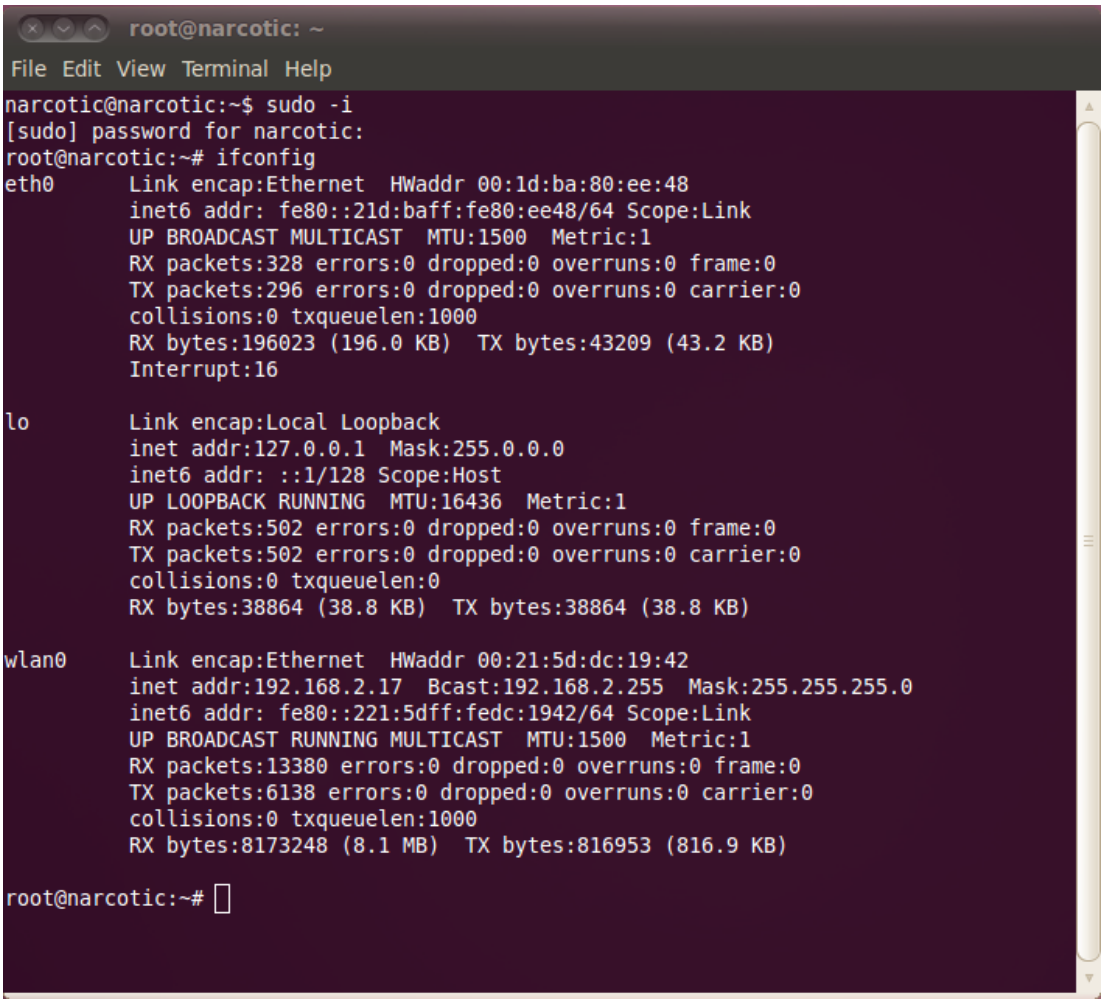
Το ένα δίκτυο λοιπόν στο οποιο θα ανήκει ο OpenVPN Server αποτελείται από ένα cisco modem/router με wireless interface και gateway ip την 192.168.2.1 . Ο OpenVPN Server εγκαταστήθηκε σε ένα laptop με static ip 192.168.2.17 το οποιο συνδέθηκε ασύρματα μέσω του wlan0 interface πάνω στο wireless router κάνοντας χρήση WPA2 Sec. Η εξωτερική ip του δικτύου που βρίσκεται ο OpenVPN Server είναι η 79.166.232.155 . Επειδή το δίκτυο μας δεν διαθέτει static external ip από κάποιον provider επιλέχτηκε για την υλοποίηση η χρησιμοποίηση μιας dynamic dns υπηρεσίας όπως το dyndns.com . Με τις υπηρεσίες DDNS μπορείς πολύ εύκολα κάνοντας register (free) ένα δικό σου hostname του στιλ home.dyndns.org να κατεβάσεις μια εφαρμογή όπου θα τρέχει σαν service στο background και κάθε πχ 10 λεπτά θα στέλνει την δυναμική εξωτερική ip στον server του dyndns.com. Με αυτό το τρόπο καταφέρνεις να έχεις ένα στατικό dns πάνω από μια δυναμική ip. Στην δικια μου περίπτωση ούτε αυτό χρειάστηκε να γίνει, διότι το modem

διαθέτει την επιλογή να κάνω login με τα στοιχεία μου στο dyndns.com απευθείας οπότε και θα τερματίζει εκεί σε επίπεδο modem και όχι pc. Με την χρήση της υπηρεσίας αυτής λοιπόν πετυχαίνω όπου και αν βρίσκομαι σαν client να γνωρίζω ποια είναι η ip του server μου (μέσω του dns που έχω κάνει register) και να μην χρειάζεται να μπαίνω σε περιττές διαδικασίες/κόστη. Οποτε αντίστοιχα στο configuration των client αντί για την δυναμική εξωτερική ip του server 79.166.232.155 (όπου θα χρειαζόταν να την αλλάζω κάθε φορα που γίνεται release), θα βάζω το στατικό dns που έχω δημιουργήσει στο server, δηλαδή το home.dyndns.org . Ακόμα από την μεριά του server θα χρειαστεί να κάνω login στο modem για να ρυθμίσω το NAT, ώστε να επιτρέψει να περάσει η udp κίνηση του port 1194 (πάνω στο οποιο θα δουλεύει το VPN μας) στην εσωτερική ip του openvpn server 192.168.2.17 . Εκτός από το portforward ίσως χρειαστεί ανάλογα με το modem να ρυθμίσω και κάποια rules στο firewall του modem, ώστε να επιτρέπει την κίνηση αντίστοιχα στο port αυτό.

Από την μεριά του OpenVPN Client έχω ένα USRobotics modem/router με wireless interface και gateway ip την 192.168.1.1 . Ο OpenVPN Client εγκαταστάθηκε σε ένα desktop με ip 192.168.1.35 μέσω του ενσύρματου interface eth0 . Αντίστοιχα με τον server έχουν γίνει τα port forward και στο εσωτερικό δίκτυο του client, ώστε η κίνηση από το udp port 1194 του NAT να προωθείται στην εσωτερική ip 192.168.1.35 που είναι συνδεδεμένος ο OpenVPN Client. Ακόμα το hardware firewall έχει ρυθμιστεί ώστε να μην απορρίπτει την αντίστοιχη κίνηση.

Αφού λοιπόν έχω ετοιμάσει το hardware κομμάτι του δικτύου μου για την υποδοχή του VPN συνεχίζω με όσα περιέγραψα στα προηγούμενα κεφαλαια για την δημιουργία των κλειδιών και certificates που απαιτούνται. Όταν τελειώσω με την ασφαλή μεταφορά των αρχείων στους clients ξεκινάω την διαδικασία δημιουργίας του VPN Tunnel το οποιο και θα περιγράψω αναλυτικότερα παρακάτω όσον αφορά τις ενέργειες όπου θα χρειαστεί να γίνουν τόσο στο κομμάτι του OpenVPN Server όσο και σε αυτό του OpenVPN Client.

- Αφού βρίσκομαι λοιπόν στο pc του ***OpenVPN Server*** κάνω login σαν root και πληκτρολογώ ifconfig , ώστε να δω τα φυσικά interface που υπάρχουν.



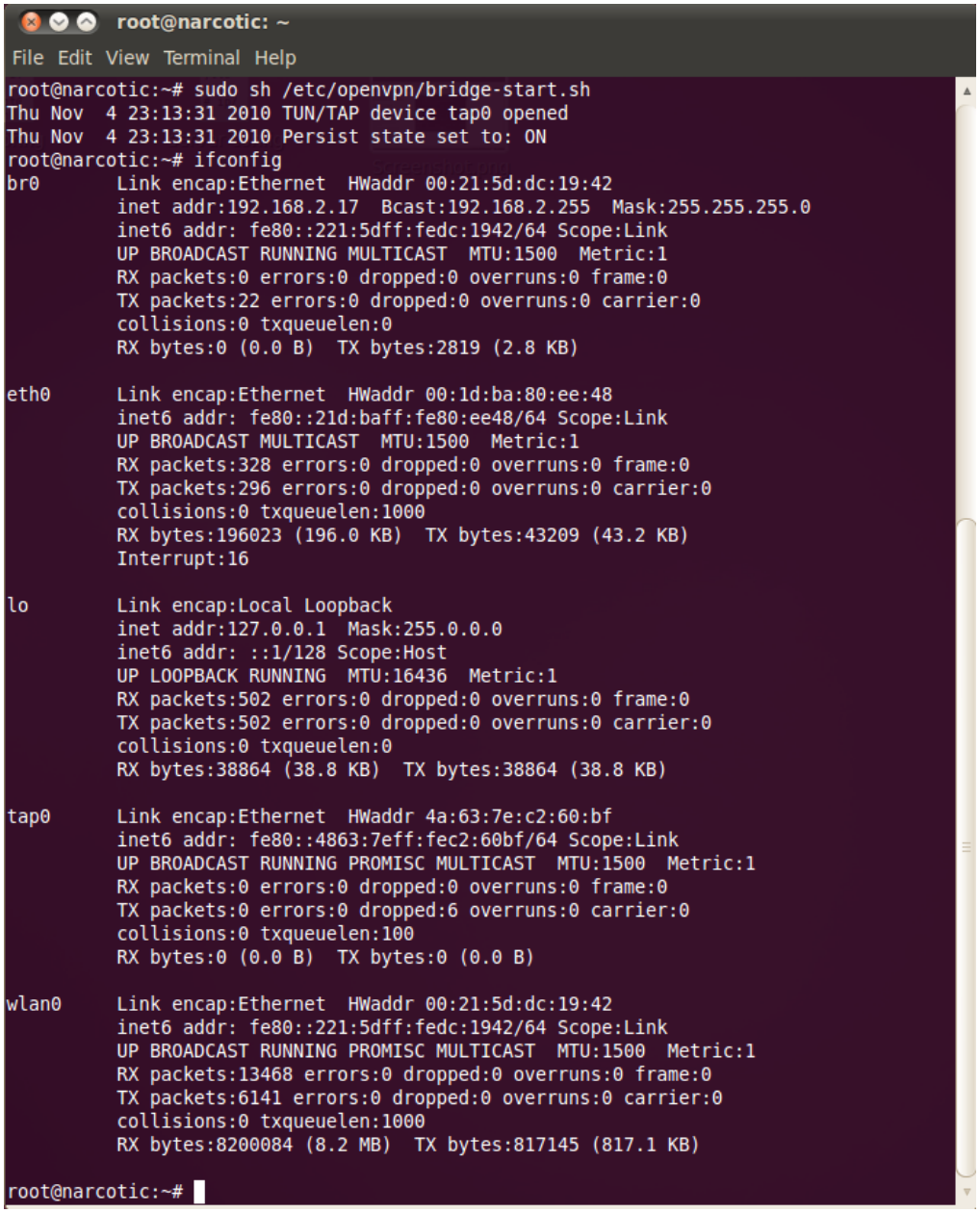
Παρατηρώ λοιπόν ότι το σύστημα που βρίσκεται εγκατεστημένος ο OpenVPN Server διαθέτει ένα ενσύρματο eth0 και ένα ασύρματο wlan0 interface. Όπως φαίνεται και στο προηγούμενο σχεδιάγραμμα υλοποίησης, ο OpenVPN Server βγαίνει προς τα έξω μέσω του wireless interface (wlan0) έχοντας μια internal static ip 192.168.2.17 η οποία συνδέεται σε ένα Cisco wireless router 192.168.2.1 με external dynamic ip 79.166.232.155 .

Στην συνεχεια ενεργοποιώ το bridge-start.sh script το οποίο περιέγραψα σε προηγούμενο κεφάλαιο, ώστε να δημιουργήσει το bridge ανάμεσα στο φυσικό interface wlan0 και στο tap0 του TAP driver του Openvpn. Το αποτέλεσμα αυτού του bridge είναι να δημιουργηθεί ένα άλλο εικονικό interface το br0 το

οποιο όπως θα παρατηρήσω παρακάτω θα πάρει πλέον την ip του lan μου. Αρκεί λοιπόν να πληκτρολογήσω

sudo sh /etc/openvpn/bridge-start.sh

και **ifconfig** για να δω τις αλλαγές που έχει δημιουργήσει στο δίκτυο μου το bridge.



Έπειτα για να συνεχίσω να έχω πρόσβαση προς τα έξω (internet) θα χρειαστεί να κάνω ένα route add ώστε να ρυθμίσω τον gateway του συστήματος μας, απλά πληκτρολογώντας

route add default gw 192.168.2.1

```
root@narcotic:~# route add default gw 192.168.2.1
root@narcotic:~#
```

Αφού έχω λοιπόν προηγουμένως δημιουργήσει το bridge, για να ξεκινήσω τον OpenVPN Server αρκεί να πληκτρολογήσω

sudo openvpn --config /etc/openvpn/server.conf

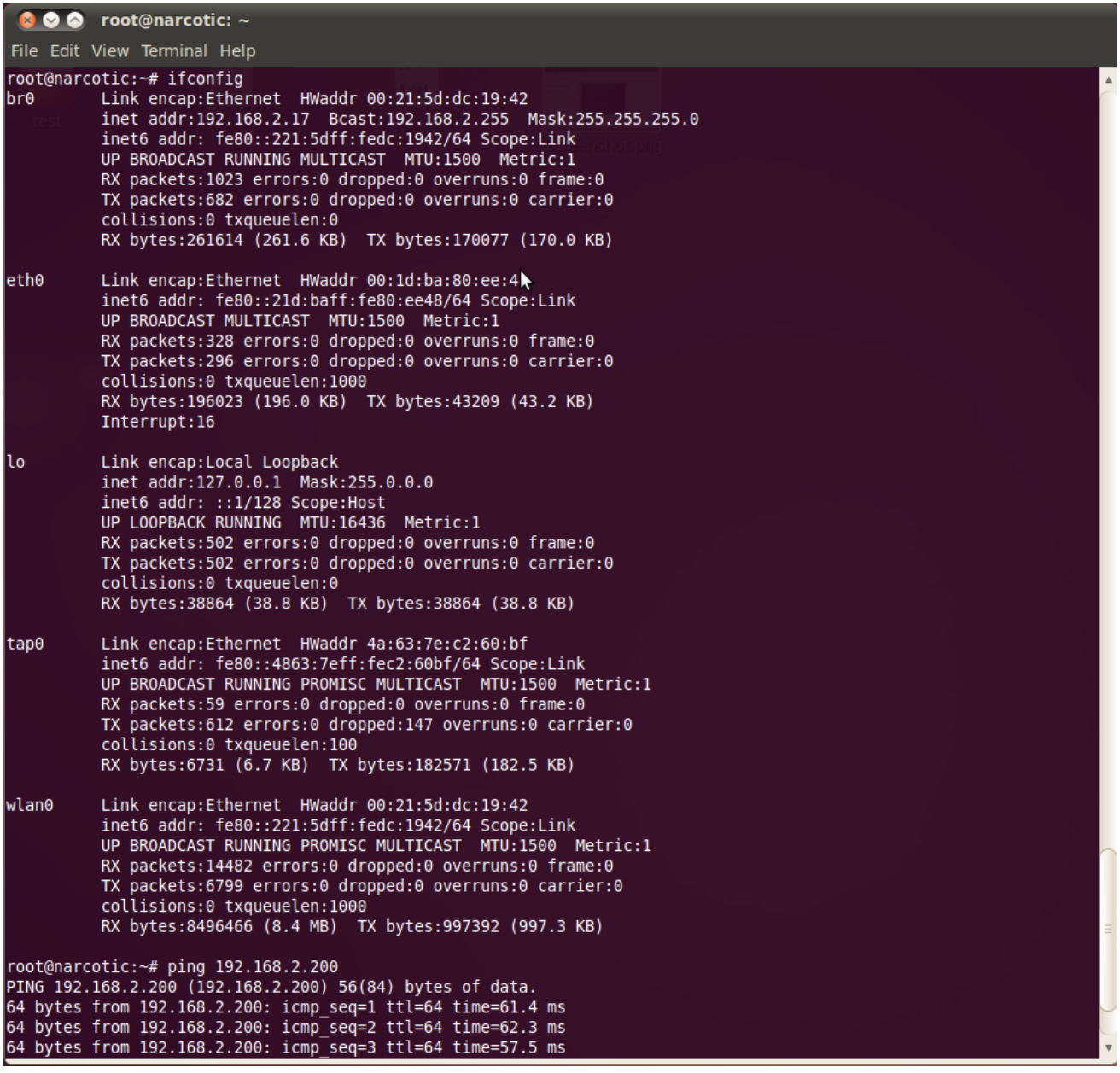
```
root@narcotic: ~
File  Edit  View  Terminal  Help
narcotic@narcotic:~$ sudo -i
[sudo] password for narcotic:
root@narcotic:~# sudo sh /etc/openvpn/bridge-start.sh
Sat Nov  6 19:47:55 2010 TUN/TAP device tap0 opened
Sat Nov  6 19:47:55 2010 Persist state set to: ON
root@narcotic:~# route add default gw 192.168.2.1
root@narcotic:~# sudo openvpn --config /etc/openvpn/server.conf
```

Συνεχίζω κάνοντας ένα **netstat –rn** για να δω τις αλλαγές στο ip routing table του συστήματος μου.

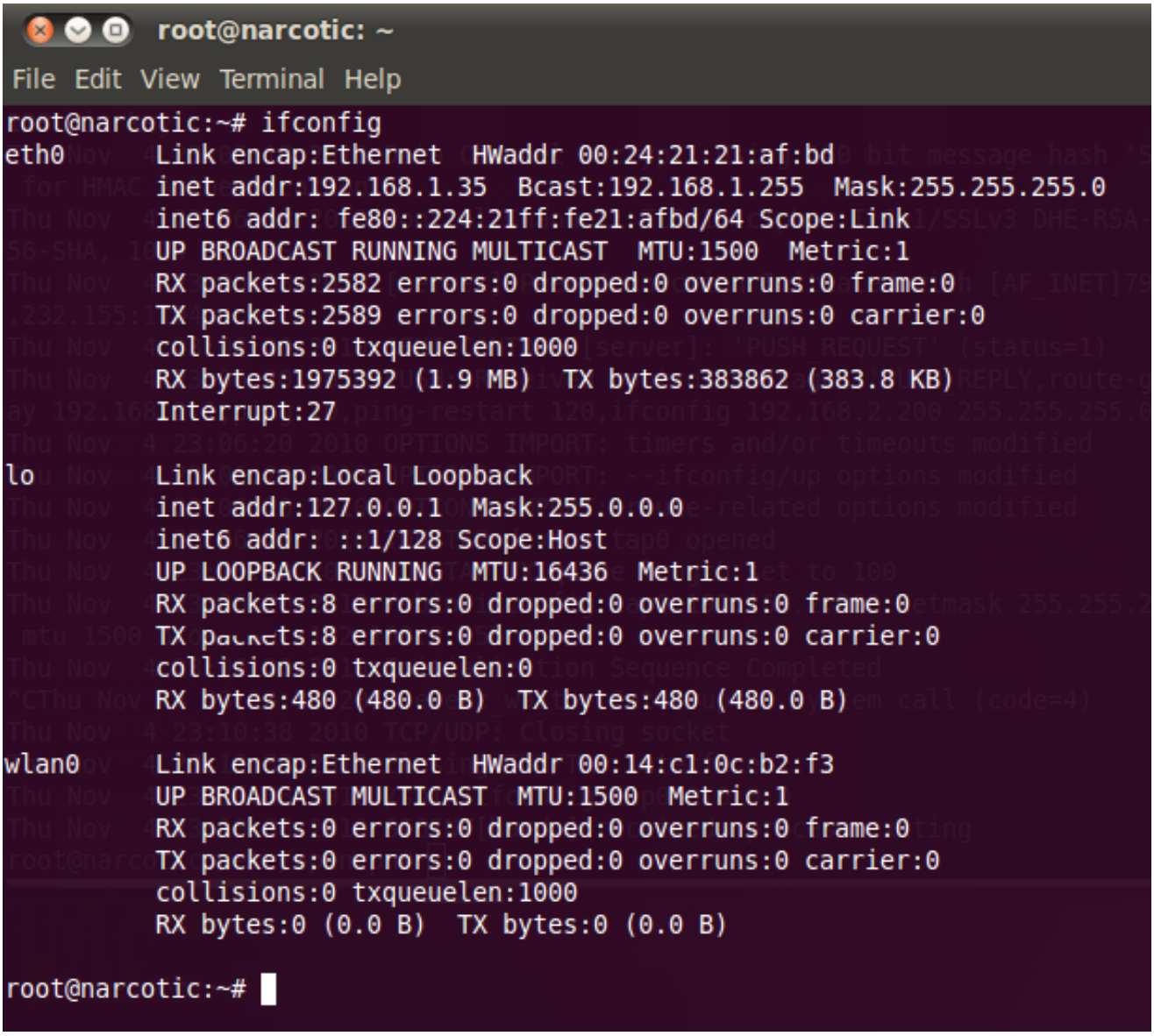
```
root@narcotic: ~
File  Edit  View  Terminal  Help
root@narcotic:~# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.2.0     0.0.0.0         255.255.255.0   U         0 0          0 br0
0.0.0.0         192.168.2.1     0.0.0.0         UG        0 0          0 br0
root@narcotic:~#
```

Βλεπω λοιπόν ότι με το **route add default gw 192.168.2.1** δημιουργήθηκε μια εγγραφη στο ip routing table του συστήματος μου, όπου λέει ότι οποιαδήποτε δρομολόγηση αφορά το bridge interface (br0) του δικτύου μου θα γίνεται πλέον μέσω του gateway 192.168.2.1 .

Αφού δημιουργήθηκε το VPN tunnel και συνδέθηκε επιτυχώς ο client πάνω κάνω ένα **ifconfig** πάλι στο server για να δω τις ρυθμίσεις των interface καθώς και ένα **ping** στην ip 192.168.2.200 του client1 που πήρε από το vpn client pool και έχω ρυθμίσει στο server.conf .



- Αντίστοιχα λοιπόν στο ***OpenVPN Client 1*** του VPN κάνω ένα ifconfig για να δω τα interface που διαθέτει το σύστημα μας. Παρατηρώ ότι το wired interface (eth0) έχει ήδη πάρει μια static ip 192.168.1.35 από το USRobotics router/gateway και μέσω αυτού του interface βγαίνει προς τα έξω.



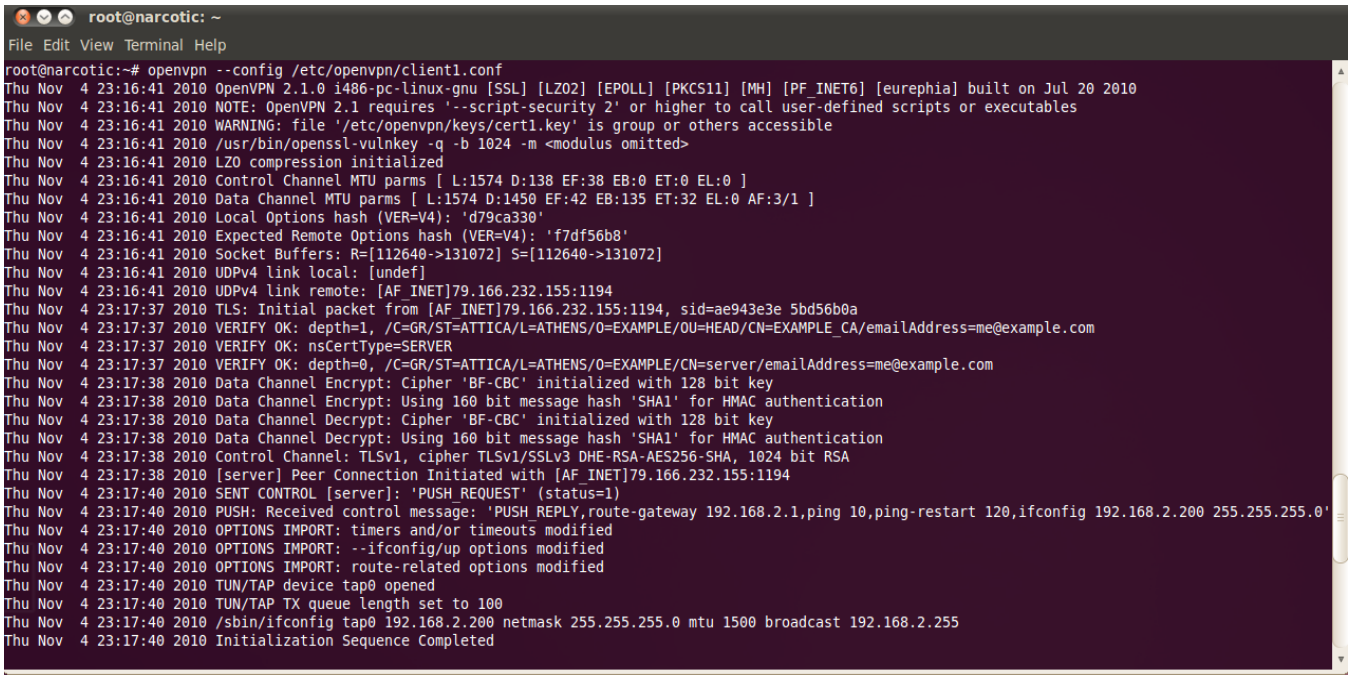
Αφού λοιπόν έχω εγκαταστήσει το openvpn στον client και έχω μεταφέρει με ασφάλεια όλα τα απαιτούμενα αρχεία που χρειάζομαι ξεκινάω την διαδικασία για να συνδεθώ στον OpenVPN Server. Πριν το κάνω αυτό απλά υπενθυμίζω ότι τα αρχεία που χρειάζεται να υπάρχουν στον client1 είναι ca.crt , cert1.crt , cert1.key , ta.key , client1.conf.

Έτσι λοιπόν απλά πληκτρολογώντας από κονσόλα την παρακάτω εντολή σαν root ουσιαστικά τρέχω το client1.conf το οποιο δοκιμάζει να συνδεθεί στον OpenVPN Server. Αν η διαδικασία είναι επιτυχής στο τέλος θα μας εμφανίσει το μήνυμα **Initialization Sequence Completed**.

Για να ξεκινήσουμε τον client 1 αρκεί να πληκτρολογήσουμε από κονσόλα σαν root το εξής

openvpn –config /etc/openvpn/client1.conf

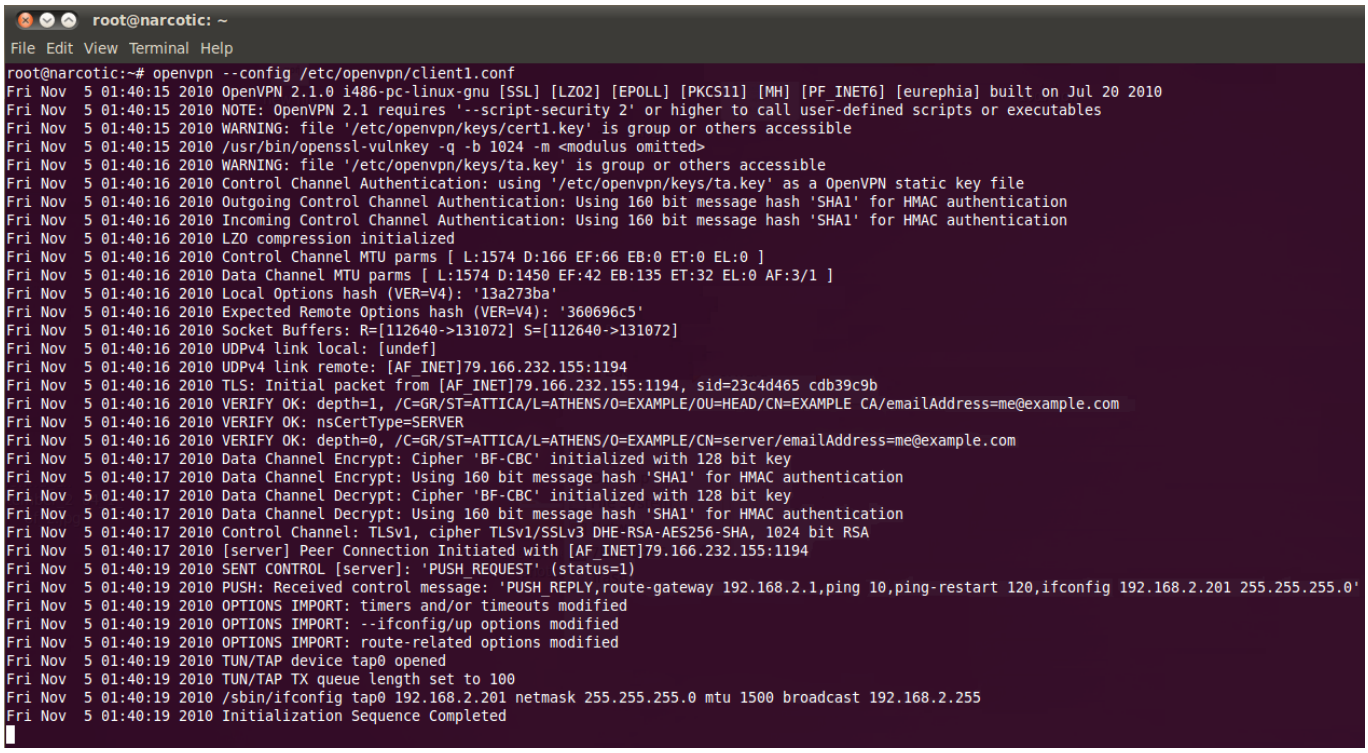
Από κάτω βλέπω τι εμφανίζεται στον client όταν δοκιμάζει να συνδεθεί στον server ***χωρίς να έχω ενεργοποιήσει το extra ta.key option του TLS***.



Παρατηρώ ότι δεν λαμβάνει χωρα κανένα ingoing/outgoing control channel authentication σε αντίθεση με την περίπτωση που έχω ενεργοποιημένη την παράμετρο αυτή.

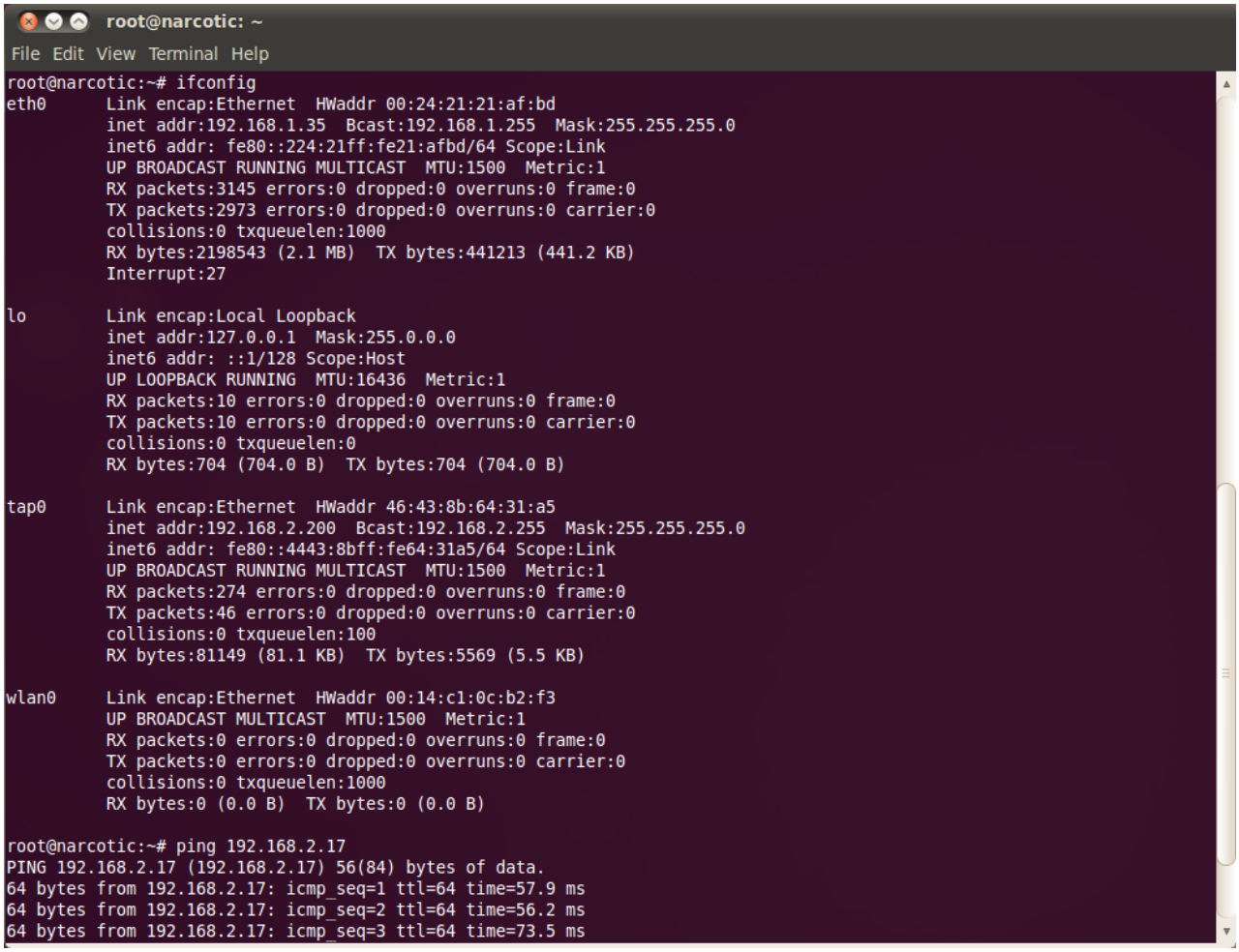
Αντίστοιχα λοιπόν ***με το ta.key option ενεργοποιημένο*** στον client και server εμφανίζεται το παρακάτω. Υπενθυμίζω ότι το ta.key δεν είναι

αναγκαίο για να παίξει το VPN, αλλα το προσθέτουμε σαν μια επιπλέον δικλείδα ασφαλειας για προστασία από UDP Flooding και DOS Attacks. Όταν λοιπόν θελήσουμε να το ενεργοποιήσουμε θα πρέπει να προστεθεί τόσο από την μεριά του client configuration όσο και αυτή του server αντίστοιχα με το τρόπο που περιέγραψα στο αντίστοιχο κεφάλαιο της παραμετροποίησης του server kai client configuration.

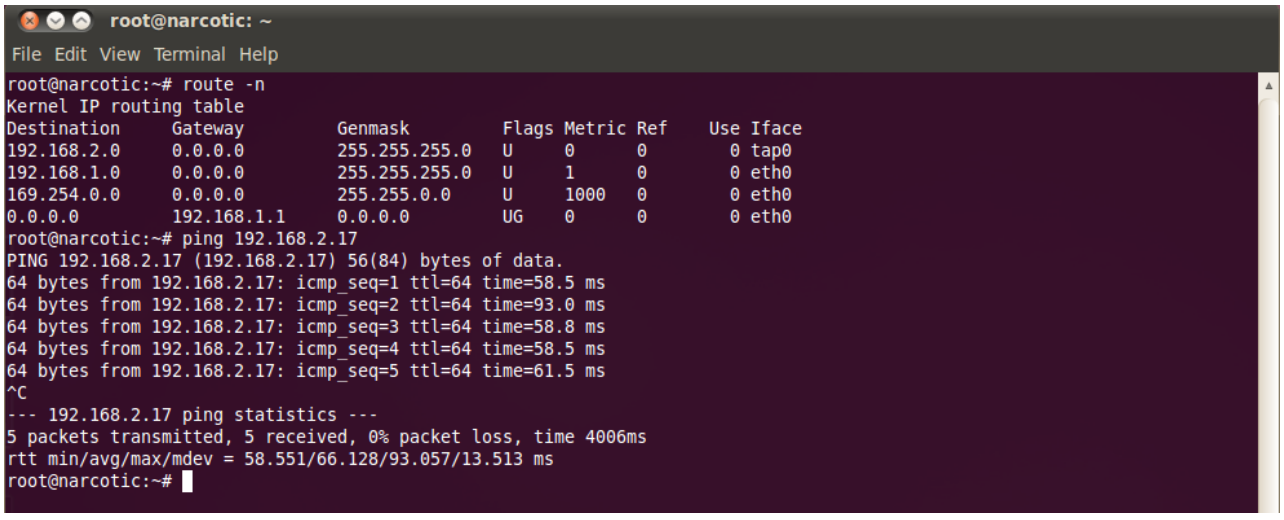


Αφού έχω συνδεθεί κάνω ένα **ifconfig** για να δω τα interface και παρατηρώ ότι έχει δημιουργηθεί και ένα ακόμα εικονικό interface το tap0 το οποιο έχει πάρει την εικονική vpn ip 192.168.2.200 που είχαμε ρυθμίσει στο server.conf.

Έπειτα συνεχίζοντας κάνω ένα **ping** στην ip 192.168.2.17 του OpenVPN Server για να δω ότι υπάρχει connectivity.



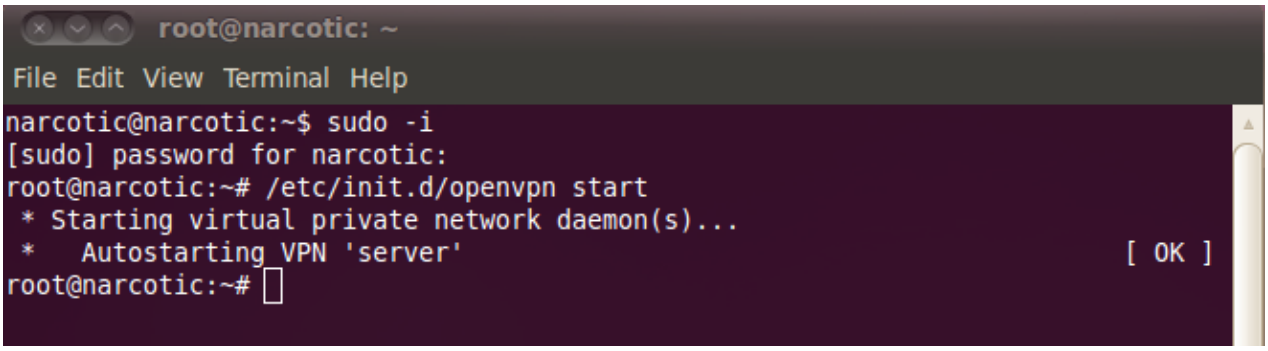
Κάνοντας και ένα **route -n** βλέπω το ip routing table του client 1.



Οι εντολές που χρειάζεται να γνωρίζω για το χειρισμό του OpenVPN περιγράφονται παρακάτω.

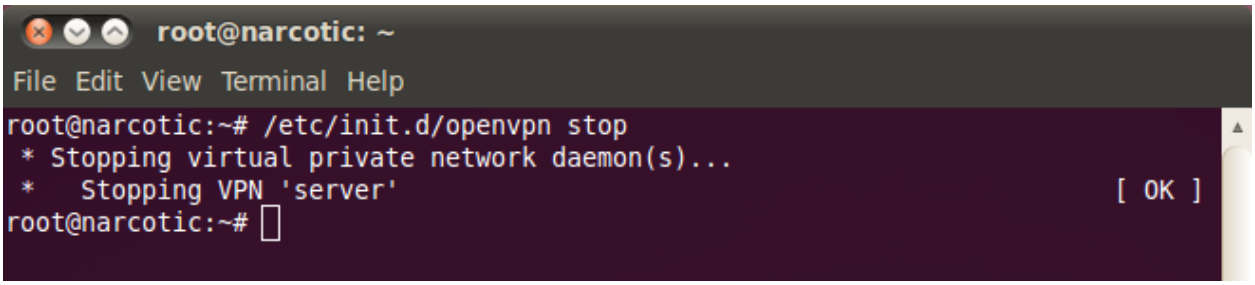
Για να ξεκινήσω λοιπόν το OpenVPN μπορώ να το κάνω χρησιμοποιώντας

- είτε την ***sudo openvpn --config /etc/openvpn/server.conf***
- ή με ***/etc/init.d/openvpn start***



Για να σταματήσω το OpenVPN μπορώ να το κάνω είτε

- με CTRL+C αν έχω χρησιμοποιήσει τον πρώτο τρόπο
- ή με ***/etc/init.d/openvpn stop*** αν έχω χρησιμοποιήσει το ***/etc/init.d/openvpn start*** για να ξεκινήσει.



Ακόμα να σημειωθεί ότι κάθε φορά που αλλάζω κάποια ρύθμιση στο server.conf (configuration του server) πρέπει να ξαναξεκινάω το OpenVPN.

Από κάτω ακολουθεί η σωστή σειρά εκτέλεσης των εντολών για την λειτουργία του VPN.

- Οποτε από την ***μεριά του OpenVPN Server*** χρειάζεται να τηρηθεί η εξής σειρά εκτέλεσης των εντολών.

  - Πρώτα ξεκινάω το bridge των interface

    sudo sh /etc/openvpn/bridge-start.sh

    root@narcotic:# **sudo sh /etc/openvpn/bridge-start.sh**

  - Μετά ξεκινάω τον OpenVPN Server με

    openvpn –config /etc/openvpn/server.conf

    root@narcotic:# **openvpn –config /etc/openvpn/server.conf**

  - Προσθέτω από CLI ένα ***route add gw*** για να έχω connectivity αφού πλέον θα βγαίνω μέσω του bridge (br0 interface)

    route add default gw 192.168.2.1

  - Έπειτα όταν θελήσω να σταματήσω τον OpenVPN Server το κάνω απλά με ένα **CTRL + C** .

  - Τέλος σταματάω το bridge των interface με

    sudo sh /etc/openvpn/bridge-stop.sh

    root@narcotic:# **sudo sh /etc/openvpn/bridge-stop.sh**

❖ Ακόμα από την μεριά ***του OpenVPN Client*** αρκεί

- Να ξεκινήσω τον OpenVPN Client τρέχοντας

  openvpn –config /etc/openvpn/client1.conf

  root@narcotic:# **openvpn –config /etc/openvpn/client1.conf**

- Έπειτα όταν θελήσω να σταματήσω τον OpenVPN Client το κάνω απλά με ένα **CTRL + C** .

Ακόμα πολλές φορες μπορεί να κολλήσουν τα physical interface και να χρειαστεί να γίνει από command line η επαναφορά. Έτσι λοιπόν ας υποθέσουμε ότι έχει χαθεί το connectivity του OpenVPN Server με το wireless router. Εμείς για να επαναφέρουμε την σύνδεση θα πρέπει από cli να πληκτρολογήσουμε τα εξής.

Για να ρίξουμε το wlan0 interface

sudo /sbin/ifconfig wlan0 down

Για να δούμε το status των interface

ifconfig

Για να επαναφέρουμε το wlan0 interface

sudo /sbin/ifconfig wlan0 up

Αντίστοιχα μπορεί να χρησιμοποιηθούν οι ίδιες εντολές και για τα υπόλοιπα physical interfaces (eth0 κλπ).

Ακόμα μερικές εντολές που ίσως φανούν αρκετά χρήσιμες είναι και οι παρακάτω.

Με αυτήν κανεις ότι ακριβώς αναφέρει, δηλαδή flush το ip routing table στην περίπτωση που κάποια routes που έχεις κάνει δεν είναι χρήσιμα πλέον.

ip route flush table main

Αντίστοιχα πρέπει να δημιουργήσουμε και αυτό το φάκελο ***/var/log/openvpn/*** με τα αντιστοιχα logs και να δοθούν user/group δικαιώματα. Με αυτό το τρόπο το OpenVPN θα μπορεί να κάνει log όλα τα στοιχεια σχετικά με τα VPN μας πράγμα πολύ χρήσιμο ειδικά αν έχουμε πολλούς συνδεδεμένους clients.

/var/log/openvpn/openvpn-status.log
/var/log/openvpn/openvpn.log

Επιπλέον αν θελήσουμε ποτε να βάλουμε static τα στοιχεια από cli σε μια κάρτα δικτύου eth0 αρκεί έχοντας κάνει login σαν root να πληκτρολογήσουμε

ifconfig eth0 192.168.2.17 broadcast 192.168.2.255 netmask 255.255.255.0

Βιβλιογραφία

- **Tanenbaum, Andrew S.**, «Δίκτυα Υπολογιστών», Κλειδάριθμος, 2003
- **J.F. Kurose, K.W. Ross**, «Δικτύωση Υπολογιστών», Α. Γκιούρδα, 2008
- **www.ietf.org -** The Internet Engineering Task Force-Internet Drafts και RCF pages
- **www.openvpn.net -** Εταιρία OpenVPN technologies που συντηρεί και το λογισμικό ανοιχτού κώδικα OpenVPN
- **www.microsoft.com** - Πληροφορίες για το PPTP.
- **www.google.com** - Η πιο γνωστή μηχανή αναζήτησης
- **wikipedia - T**he free encyclopedia
- **www.cisco.com -** Εταιρεία Δικτυακών λύσεων με προϊόντα , υπηρεσίες, ολοκληρωμένες λύσεις κ.λ.π
- **www.wiley.com -** Εκδοτικός Οργανισμός-Βιβλία για VPN και links σε προϊόντα ,υπηρεσίες, VPN tests, Internet Drafts και RCF pages κ.λ.π.
- **www.strongvpn.com -** Εταιρεία VPN λύσεων με προιόντα, ολοκληρωμένες λύσεις κ.λ.π
- **www.nortelnetworks.com -** Εταιρεία Δικτυακών λύσεων με προϊόντα, υπηρεσίες ,ολοκληρωμένες λύσεις κ.λ.π
- **www.3com.com -** Εταιρεία Δικτυακών λύσεων με προϊόντα, υπηρεσίες, ολοκληρωμένες λύσεις κ.λ.π
- **www.shiva.com -** Εταιρεία Δικτυακών λύσεων με προϊόντα, υπηρεσίες, ολοκληρωμένες λύσεις κ.λ.π
- **www.signal9.com -** Εταιρεία Δικτυακών λύσεων με προϊόντα (firewalls, VPNs κ.λ.π)

Παράρτημα

Παραμετροποίηση server

```
# Which local IP address should OpenVPN
# listen on? (optional)
local 192.168.2.17

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
proto udp
;proto tcp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap" if you are ethernet bridging.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tap0
;dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
```

```
# (see "pkcs12" directive in man page).
ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt

cert /etc/openvpn/easy-rsa/2.0/keys/server.crt

key /etc/openvpn/easy-rsa/2.0/keys/server.key
# This file should be kept secret


# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh /etc/openvpn/easy-rsa/2.0/keys/dh1024.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
; server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
# server-bridge SERVER_GATEWAY SUBNET CLIENT_VPN_POOL_START END
server-bridge     192.168.2.1     255.255.255.0     192.168.2.200
192.168.2.202

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.2.1 255.255.255.255 net_gateway 1"
;push "route 192.168.2.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
```

```
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"
```

```
# Comment this directive to disallow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth /etc/openvpn/easy-rsa/2.0/keys/ta.key 0
# This file is secret
;crl-verify /etc/openvpn/easy-rsa/2.0/keys/crl.pem

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
max-clients 3
```

```
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
user nobody
group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
;status openvpn-status.log
status /var/log/openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         openvpn.log
log-append  /var/log/openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```