Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής

Πτυχιακή εργασία

Τίτλος: Unity Network ~ Πλατφόρμα κατανεμημένου εικονικού δικτύου με χαρακτηριστικά ασφάλειας και επωνυμίας χρηστών

Καγιαμπάκης Κωνσταντίνος (ΑΜ: 2770)

Επιβλέπων Καθηγητής: Μανιφάβας Χαράλαμπος

Επιτροπή Αξιολόγησης: Γ. Κορνάρος, Ν. Παπαδάκης, Χ. Μανιφάβας

Ημερομηνία Παρουσίασης: Ιούνιος 2014

Ευχαριστίες

Το παρών έργο το αφιερώνω στα άτομα όπου καταλαβαίνουν και αναγνωρίζουν ότι προσπαθώ να αποδείξω και ότι υπερασπίζομαι. Επίσης αφιερώνεται στους μαχητές, στα άτομα εκείνα όπου δεν καταθέτουν τα όπλα και δεν ξεπουλάνε τα πιστεύω τους όταν οι συνθήκες δυσκολεύουν. Τέλος το αφιερώνω στα ελεύθερα και δημιουργικά πνεύματα γιατί ο κόσμος χωρίς αυτά δεν θα είχε πνοή και κίνηση!

Abstract

From a technical aspect this project has the goal to present that a large and distributed virtual network may exist on a network and specifically on the internet that is less related to hardware and more logical in comparison to small rooms of modern VPN implementations. By virtual distributed space we mean that the virtual network is distributed and it runs on several nodes, if a node dies the platform does not die. Moreover it is as less related as it can go from network hardware and Operating Systems and in comparison with other virtual networks thanks to its architect it has a bigger capacity in users. More specifically it has the following attributes:

- Distribution on its nodes so therefore larger scale, resistance to death
- Different OS support
- RSA authentication between nodes
- NAT traversal techniques for the clients behind a NAT/firewall to be able to pass it and connect to the virtual network
- The ability for each user to state many hostnames for his devices and each one to have a unique static IP address

From a philosophical aspect this project shows the many problems internet has nowadays such as NAT, Man In The Middle problems and network trafficking and the lack of encryption and it offers a unique virtual distributed space that can offer three basic principles to its users:

- The ability for a user to have as much hosts as he owns connected to the network and the ability for a user's host machine to be individually identified by an IP address each time this host connects to the network ~ Individuality
- The ability for a user's host to exchange any type of information he wants with anyone ~ Freedom
- The ability for each host to exchange encrypted information with each other ~ Security

Σύνοψη

Στη σημερινή εποχή παρουσιάζεται έντονο το φαινόμενο οι χρήστες που συνδέονται στο διαδίκτυο, να περιορίζονται στο ποιές υπηρεσίες μπορούν να χρησιμοποιούν από ενδιάμεσους κόμβους, ISP ή από διαχειριστές δικτύων. Επίσης είναι γνωστό ότι πλέον στο διαδίκτυο υπάρχουν συνεχώς παραβιάσεις του ιδιωτικού απορρήτου και η συλλογή πληροφοριών από ενδιάμεσους κόμβους είναι αρκετά συχνή.

Η συγκεκριμένη πτυχιακή εργασία έχει ως στόχο αφενός να κατονομάσει τα παραπάνω προβλήματα και αφετέρου να δώσει λύση στα παραπάνω προβλήματα κατασκευάζοντας ένα κατανεμημένο εικονικό δίκτυο με στόχο χρήσης στο διαδίκτυο στο οποίο οι χρήστες όπου θα επιλέξουν να συνδεθούν θα μπορούν να διαμοιράσουν οποιασδήποτε μορφής υπηρεσία επιθυμούν μεταξύ τους και μέσα από μια κρυπτογραφημένη σύνδεση να μην υπάρχει η δυνατότητα υποκλοπής των πληροφοριών όπου ανταλλάσσουν από τρίτους.

Πιο συγκεκριμένα το δίκτυο θα παρουσιάζει τις παρακάτω ιδιότητες:

- Θα παρουσιάζει κατανομή ως προς τους κόμβους του
- Θα προσφέρει κρυπτογραφημένη σύνδεση μεταξύ του υπολογιστή όπου θα επιλέξει να συνδεθεί και του δικτύου
- Θα παρέχει τεχνικές NAT traversal προκειμένου οι υπολογιστές ενός τοπικού δικτύου να μπορούν να διαμοιράσουν οτιδήποτε επιθυμούν στο εικονικό δίκτυο
- Θα παρέχει επωνυμία: ο κάθε host θα έχει στην κατοχή του μοναδική εικονική διεύθυνση IP και ο κάθε χρήστης το δικό του ζευγάρι ιδιωτικού & δημόσιου κλειδιού

Πίνακας περιεχομένων

Πίνακας Εικόνων και σχημάτων

Εισαγωγή

1.1 Τι είναι το Unity Network , ποιοί είναι οι στόχοι του;

Το Unity Network είναι μια πλατφόρμα κατανεμημένου εικονικού δικτύου με χαρακτηριστικά ασφάλειας και επωνυμίας χρηστών όπου έχει στόχο χρήσης το Διαδίκτυο. Στόχος του είναι να μπορέσει να λύσει ή να καλύψει προβλήματα τα οποία αντιμετωπίζει το σύγχρονο Διαδίκτυο, παρέχοντας στους χρήστες του αυξημένες δυνατότητες και υπηρεσίες για ευκολότερη επικοινωνία σε σχέση με αυτό. Οι τρείς βασικές του αρχές είναι η παροχή ελευθερίας επωνυμίας και ασφάλειας στους χρήστες του και τις συσκευές τους.

1.2 Πως μπορεί να περιγραφτεί το Unity αρχιτεκτονικά;

Αρχικά το Unity Network μπορεί να περιγραφεί ως εικονικό δίκτυο καθώς έχει πολλά χαρακτηριστικά από την οικογένεια των Virtual Private Networks. Η πρώτη του βασική διαφορά από τα VPN είναι οτι σε αντίθεση με αυτά χρησιμοποιεί κατανεμημένη λογική για να δρομολογήσει την εικονική του κίνηση καθώς το σύνηθες στα VPN είναι να υπάρχει ένας κεντρικός server όπου διανέμει την κίνηση συγκεντρωτικά. Το δεύτερο και πολύ σημαντικό του χαρακτηριστικό είναι οτι χρησιμοποιεί τεχνικές όπως NAT Traversing προκειμένου να κάνει πελάτες μηχανήματα με σκληρή αστυνόμευση και διαχείριση να είναι σε θέση να συνδεθούν. Επίσης ένα άλλο χαρακτηριστικό του Unity είναι οτι δεν χρησιμοποιεί γνωστά και πρωτυποποιημένα πρωτόκολλα διακίνησης εικονικής κίνησης όπως: L2TP, PPTP καθώς αναγνωρίζονται και αστυνομεύονται αρκετά εύκολα στο Διαδίκτυο. Αντίθετα χρησιμοποιεί UDP datagrams με κρυπτογραφημένο payload. Από άποψη αρχιτεκτονικής της πλατφόρμας αποτελείται από 3 είδη κόμβων:

- Τον RedNode - RN: Το πρόγραμμα πελάτη. Συνδέει το κάθε μηχάνημα όπου επιθυμεί να γίνει host στο VN
- Τον BlueNode - BN: Η εφαρμογή διακίνησης της κίνησης. Υπάρχουν πολλοί BNs όπου συνεργάζονται και ο καθένας τους αναλαμβάνει να εξυπηρετήσει RedNodes
- Τον tracker: Γνωρίζει ποιός RN έχει συνδεθεί σε ποιό BN και έχει τον ρόλο να κάνει λειτουργική την επικοινωνία του δικτύου.

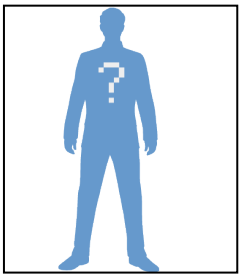
1.3 Τι προβλήματα συναντάμε στο σύγχρονο Διαδίκτυο;

Η ιδέα του Unity δημιουργήθηκε προκειμένου να επιλυθούν συγκεκριμένα προβλήματα από το διαδίκτυο τα οποία ταλαιπωρούν τους χρήστες και περιορίζουν τις πραγματικές του δυνατότητες. Τα σημαντικότερα προβλήματα αναφέρονται παρακάτω:

Έλλειψη Επωνυμίας

IPv4

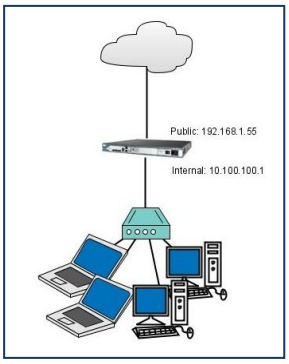
Ένα από τα πιο βασικά προβλήματα του σύγχρονου διαδικτύου έχει να κάνει με την επωνυμία όπου αυτό αποδίδει στους συνδεμένους χρήστες του. Το διαδίκτυο βασίζεται στην τεχνολογία IPv4. Η συγκεκριμένη τεχνολογία, λόγω της ραγδαίας εξάπλωσης του διαδικτύου και της πληθώρας των συσκευών όπου συνδέονται καθημερινά, παρουσιάζει έλλειμμα στις διευθύνσεις όπου μπορεί να αποδώσει σε αυτές. Μία παράγωγη κατάσταση είναι οτι προκειμένου να μπορούν να εξυπηρετηθούν όλοι οι συνδεμένοι υπολογιστές έχει υλοποιηθεί η τεχνολογία NAT.



Εικόνα 1 whoami?

Η τεχνολογία NAT

Επειδή δεν υπάρχουν πολλές διευθύνσεις IPv4 όπως προαναφέρθηκε, πολλοί υπολογιστές σε ένα τοπικό δίκτυο θα πρέπει να επικοινωνούν μοιράζοντας μία μόνο διεύθυνσης στο Διαδίκτυο. Γι' αυτό το λόγο επινοήθηκε το σύστημα μετάφρασης πολλών τοπικών διευθύνσεων σε μία μόνο διεύθυνση διαδικτύου γνωστό και ως Network Address Translation. Το NAT από πολλούς έχει χαρακτηριστεί ως ενδιάμεση λύση, ο λόγος είναι ότι προωθεί δυναμικά την κίνηση από το Διαδίκτυο προς το τοπικό δίκτυο με αποτέλεσμα πολλοί υπολογιστές του τοπικού δικτύου να παρουσιάζονται σαν ένας στο Διαδίκτυο. Τα αποτελέσματα αυτής της βίαιης δρομολόγησης είναι:

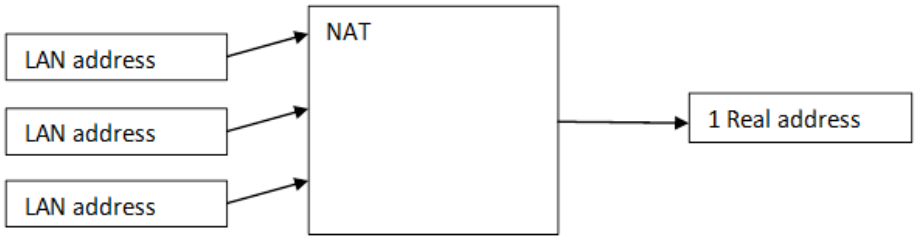


Εικόνα 2 Many hosts behind a NAT

- Όλοι οι υπολογιστές του τοπικού δικτύου θα πρέπει να μοιράζονται τις θύρες μετάδοσης σαν να ήταν ένας υπολογιστής, θύρες που μπορεί να έχει ένας υπολογιστής = αριθμός θυρών/αριθμός υπολογιστών
- Δυο και πάνω υπολογιστές δεν μπορούν να ακούν κίνηση από το διαδίκτυο στην ίδια πόρτα. Αυτό σημαίνει ότι π.χ. δυο host στο τοπικό δίκτυο δεν μπορούν να διανέμουν υπηρεσίες web server στην προεπιλεγμένη θύρα ταυτόχρονα, κάποιος πρέπει να επιλέξει μια άλλη θύρα αυξάνοντας την πολυπλοκότητα χρήσης της υπηρεσίας.
- Περιορισμός των πρωτόκολλων μετάδοσης (transmission protocols) μόνο σε 2 γνωστά, τα TCP και UDP για το λόγο οτι μόνο αυτούς τους header γνωρίζουν τα περισσότερα NAT, επομένως υπάρχει περιορισμός χρήσης των άλλων πρωτοκόλλων μετάδοσης που συνεπάγεται σε θάνατο της έρευνας νέων πρωτοκόλλων μετάδοσης.

Αυτό το πρόβλημα περιγράφεται κυρίως ως το μεγάλο πρόβλημα του NAT και μπορείτε να το δείτε και από εδώ στην πιό επιστημονική του εκδοχή: http://en.wikipedia.org/wiki/Network_address_translation#Drawbacks

επίσης εδώ http://www.useipv6.com/ στην παράγραφο Why not just use Network Address Translation (NAT)?
και εδώ http://ipv6.com/articles/nat/NAT-In-Depth.htm



Εικόνα 3 NAT

Πλήγμα στην επωνυμία των χρηστών

Τα παραπάνω φαινόμενα έχουν αποτέλεσμα πάρα πολλοί χρήστες του διαδικτύου να ομαδοποιούνται πολύ επιπόλαια και οι περισσότεροι να καταλήγουν με μια δυναμική διεύθυνση IP όπου αλλάζει συνεχώς χωρίς να είναι σε θέση ο ένας να εντοπίσει τον άλλο μονοσήμαντα από την διεύθυνση IP όπου του αποδίδεται. Η διάδοχη τεχνολογία η IPv6 έχει εγκλωβιστεί προσωρινά καθώς η αλλαγή της τεχνολογίας στο διαδίκτυο δεν είναι εύκολη υπόθεση.

Τι συνέπειες μπορεί να έχει αυτό το φαινόμενο

Ο κάθε χρήστης αλλάζει διεύθυνση συνεχώς με αποτέλεσμα οι άλλοι χρήστες να μην είναι σε θέση να τον βρίσκουν μονοσήμαντα από τη διεύθυνση του. Πχ. το αντίστοιχο παράδειγμα στο δίκτυο κινητής τηλεφωνίας είναι ο αριθμός κάθε συσκευής να αλλάζει συνεχώς και να μην είναι σε θέση ο ένας χρήστης να εντοπίζει τον άλλο. Στο διαδίκτυο λοιπόν όπου οι χρήστες έχουν συνεχώς εναλλασσόμενες διευθύνσεις, δεν μπορούν να βρεθούν βάση αυτής και καταλήγουν σε επικοινωνία διαμέσω συνδρομητικών υπηρεσιών όπως (facebook, skype, google plus) προκειμένου

να είναι σε θέση να βρεθούν και να μπορέσουν να επικοινωνήσουν. Αυτό το φαινόμενο είναι άμεσο πλήγμα για τους χρήστες του διαδικτύου καθώς προκειμένου να επικοινωνήσουν γίνονται αποδέκτες όρων και συμβάσεων από την κάθε υπηρεσία όπου υπόσχεται επικοινωνία. Αποδέκτης όρων και συμβάσεων σημαίνει αυτόματα οτι ένας χρήστης προκειμένου να μπορεί να επικοινωνήσει γίνεται δέσμιος της εταιρίας και πλέον η ίδια η εταιρία μπορεί να είναι και νομικά δικαιούχος για την παρακολούθηση του! Εμείς εναντιωνόμαστε σε αυτή την τάση και υποστηρίζουμε οτι αφενός ο κάθε χρήστης θα πρέπει να είναι σε θέση να αναγνωρίζεται μονοσήμαντα κάθε φορά που συνδέεται στο διαδίκτυο και αφετέρου είναι αναγκαίο να αρχίσει να δημιουργείται και να εφαρμόζεται ένα ενιαίο νομοθετικό πλαίσιο για το διαδίκτυο σχετικά με την επικοινωνία χρηστών.

Παράγωγα προβλήματα

Αφενός το διαδίκτυο αδυνατεί να καλύψει την ανάγκη απευθείας επικοινωνίας των χρηστών του, αφετέρου οι χρήστες είναι αναγκασμένοι να γίνουν συνδρομητές σε υπηρεσίες προκειμένου να μπορούν να επικοινωνήσουν. Προκειμένου λοιπόν δυο χρήστες να είναι σε θέση να μιλήσουν απευθείας είναι ανάγκη και οι δύο να έχουν γραφτεί σε μία κοινή υπηρεσία επικοινωνίας. Εδώ παρουσιάζεται το φαινόμενο του κατακερματισμού των χρηστών καθώς άλλοι είναι γραμμένοι στην Α υπηρεσία και άλλοι στη Β.

Παρουσιάζονται νέες εξελικτικές τάσεις

Τα προηγούμενα χρόνια ένας χρήστης ή οργανισμός θα αντιστοιχούσε λογικά σε μία μόνο διεύθυνση επειδή ένας χρήστης είχε ένα ή κανένα μηχάνημα όπου συνδεόταν σε δίκτυο. Η σχέση ενός χρήστη μηχανήματος ήταν 1-1. Σήμερα ένας χρήστης κατέχει πολλές συσκευές: κινητό τηλέφωνο, σταθερό υπολογιστή, φορητό υπολογιστή, συσκευές τύπου tablet οι οποίες Θα ήθελε να είναι συνδεμένες ταυτόχρονα πάνω στο διαδίκτυο και μάλιστα να είναι σε θέση να διακριθούν η μια από την άλλη. Πχ. ένας χρήστης μπορεί να έχει την ανάγκη να στείλει ένα αρχείο στο σταθερό υπολογιστή του χρήστη προορισμού ή να τον καλέσει στο κινητό και όχι στον σταθερό του υπολογιστή. Επομένως ένας χρήστης κατέχει πολλές συσκευές 1-Ν και το δίκτυο θα πρέπει να διαμορφωθεί βάση αυτού του μοντέλου. Επίσης μια πρόσφατη εξελικτική τάση είναι συσκευές του ίδιου χρήστη όπου είναι συνδεμένες σε κοινό δίκτυο να έχουν τη δυνατότητα συγχρονισμού και καταμερισμού εργασιών ή ακόμα και διαμοιρασμό κοινού γραφικού περιβάλλοντος, γι' αυτό το λόγο χρειάζονται μόνιμες διευθύνσεις αναγνώρισης.

Περιορισμός ελευθερίας

Ένα άλλο μεγάλο σύνολο προβλημάτων όπου παρουσιάζει το διαδίκτυο έχει να κάνει με την ποιότητα της ελευθερίας όπου απονέμει στους χρήστες του. Με τον όρο ποιότητα ελευθερίας διερευνάται το πόσο εύκολο είναι για ένα χρήστη να διανέμει μια υπηρεσία και αν υπάρχουν ενδιάμεσοι όπου του καταρρίπτουν αυτό το δικαίωμα.

- Π.χ Μπορεί ένας χρήστης να κάνει host ένα web server;
- Έχει ένας υπολογιστής το δικαίωμα να κάνει χρήση μιας συγγεκριμένης θύρας στο διαδίκτυο;

Στο σημερινό διαδίκτυο υπάρχουν διάφορα τεχνικά προβλήματα, «κακοί» ενδιάμεσοι ή και μεγάλα συμφέροντα όπου περιορίζουν ένα χρήστη από το να διανέμει υπηρεσίες. Τέτοια προβλήματα είναι:

Τα προαναφερθέντα προβλήματα του NAT

- Όλοι οι υπολογιστές του τοπικού δικτύου θα πρέπει να μοιράζονται τις θύρες μετάδοσης σαν να ήταν ένας υπολογιστής, θύρες που μπορεί να έχει ένας υπολογιστής = αριθμός θυρών/αριθμός υπολογιστών
- Δυο και πάνω υπολογιστές δεν μπορούν να ακούν κίνηση από το διαδίκτυο στην ίδια πόρτα. Αυτό σημαίνει ότι π.χ. δυο host στο τοπικό δίκτυο δεν μπορούν να διανέμουν υπηρεσίες web server στην προεπιλεγμένη θύρα ταυτόχρονα, κάποιος πρέπει να επιλέξει μια άλλη θύρα αυξάνοντας την πολυπλοκότητα χρήσης της υπηρεσίας.
- Περιορισμός των πρωτόκολλων μετάδοσης (transmission protocols) μόνο σε 2 γνωστά, τα TCP και UDP για το λόγο οτι μόνο αυτούς τους header γνωρίζουν τα περισσότερα NAT, επομένως υπάρχει περιορισμός χρήσης των άλλων πρωτοκόλλων μετάδοσης που συνεπάγεται σε θάνατο της έρευνας νέων πρωτοκόλλων μετάδοσης.

Ο Διαχειριστής τοπικού δικτύου και το Firewall όπου διαχειρίζεται

Εικόνα 4 Firewall

Εδώ διερευνάται αν είναι ηθικό για ένα διαχειριστή τοπικού δικτύου να είναι σε θέση μέσω firewall να επηρεάζει την ελευθερία και την ποικιλία υπηρεσιών των κόμβων όπου επιθυμούν να συνδεθούν στο Διαδίκτυο. Εάν υποθέσουμε οτι έχουμε πολλά τοπικά δίκτυα και στο καθένα εφαρμόζονται διαφορετικές πολιτικές τότε οι κόμβοι όπου θα φτάνουν στο διαδίκτυο δεν θα διέπονται από τα ίδια δικαιώματα. Αυτό σημαίνει οτι υπάρχει τεχνική ανισότητα μεταξύ τους.

1. Ένα απλό παράδειγμα θα ήταν οτι έχουμε 2 χρήστες τον Πάκη και τον Τάκη
2. Οι 2 χρήστες θέλουν να ανταλλάξουν ένα αρχείο με FTP
3. Ο Πάκης βγαίνει στο Διαδίκτυο από ένα δίκτυο όπου του απαγορεύεται να κάνει host ένα ftp service και να συνδεθεί σε ftp
4. Τάκης είναι ελεύθερος να κάνει host αλλά δεν έχει κανένα νόημα!

Επομένως θεωρούμε οτι όλοι οι κόμβοι μέλη του διαδικτύου θα πρέπει να "φτάνουν" στο Διαδίκτυο με τα ίδια δικαιώματα καθώς έτσι μόνο θα μπορούσαν να έχουν πραγματική ελευθερία.

Το πρόβλημα του κακού ενδιάμεσου ((evil-)man-in-the-middle)

Τα πακέτα στο διαδίκτυο δρομολογούνται είτε βάση δρομολογητών είτε βάση των πάροχων internet τους γνωστούς ως ISP. Αυτοί έχουν την υποχρέωση ένα πακέτο να το παραδώσουν στον επόμενο κόμβο δρομολόγησης κοκ. Εφόσον λοιπόν δέχονται κίνηση να περνάει από μέσα τους μπορούν να την φιλτράρουν βάση διευθύνσεων και δεδομένων, να μπλοκάρουν θύρες μετάδοσης με αποτέλεσμα να μην λειτουργούν όλοι οι χρήστες πάνω στο διαδίκτυο με τα ίδια προνόμια. Χάρη σε αυτή την τάση παρακολούθησης οι επιστήμες των Pattern Recognition και Neural Networks έχουν αποκτήσει αρκετά δημοφιλή χαρακτήρα! Το γενικότερο πλαίσιο επιστημών δε όπου ασχολείται με την επιστήμη εξαγωγής πληροφοριών ονομάζεται Data Mining και πλέον θεωρείται επιστημονικός κλάδος της πληροφορικής!

Εφόσον μιλάμε για ανταλλαγή πληροφοριών αυτό είναι ένα φαινόμενο λογοκρισίας και παρακολούθησης - Stalking το οποίο καταργεί το προσωπικό απόρρητο στο διαδίκτυο και αυτή τη στιγμή είναι σύνηθες φαινόμενο.

Έλλειψη Ασφάλειας

Εικόνα 5 data encryption

Ένα μεγάλο πρόβλημα του διαδικτύου είναι η περιορισμένη του ασφάλεια. Καθημερινά στο διαδίκτυο κλέβονται αμέτρητα προσωπικά στοιχεία, κωδικοί, και χρήματα. Ο λόγος της ρίζας αυτού του προβλήματος είναι η φύση του διαδικτύου όπως έχει σήμερα. Οι χρήστες του διαδικτύου χρησιμοποιούν ελάχιστα στοιχεία κρυπτογράφησης καθώς δεν είναι ενημερωμένοι και εξοικειωμένοι για την χρήση τους. Πολλές φορές ένας χρήστης θα πρέπει να ασχοληθεί επιπλέον για την απόκτηση ζευγαριού κλειδιών ή να πληρώσει χρήματα για ένα πιστοποιητικό γνησιότητας. Επίσης η κατοχή ενός ζευγαριού συμμετρικών κλειδιών ανά χρήστη

δεν είναι απαραίτητη αλλά είναι προαιρετική. Διαδικασίες όπως ψηφιακή υπογραφή εγγράφων δεν θεωρούνται κοινωνικά διαδεδομένες. Το αποτέλεσμα των παραπάνω είναι ένα μεγάλο ποσοστό της κίνησης στο διαδίκτυο να μην είναι κρυπτογραφημένο, κάτι που κάνει την δουλειά των MITM αρκετά πολύ πιο εύκολη!

Επίλογος Προβλημάτων

Κλείνοντας διαπιστώνουμε ότι το διαδίκτυο παρόλη την εξέλιξη των τελευταίων χρόνων παρουσιάζει σημαντικά προβλήματα και ελλείψεις τόσο από τον καθημερινό περιορισμό των δικαιωμάτων του χρήστη όσο και σε θέματα προστασίας και ασφάλειας. Τέτοια προβλήματα υποβαθμίζουν την λειτουργία του και καταλήγουν να το καθιστούν ανίκανο να εκτελέσει την βασική του λειτουργία, δηλαδή την επικοινωνία των χρηστών μέσα από ένα ελεύθερο και μη ιδιωτικοποιημένο πλαίσιο . Επιπλέον το διαδίκτυο προσπαθεί να χτίσει υπηρεσίες πάνω στα υπάρχοντα προβλήματα ουσιαστικά επιλέγοντας το λάθος εξελικτικό μονοπάτι.

1.4 Πως το unity αντιμετωπίζει τέτοιου είδους προβλήματα

Όπως προαναφέρθηκε και στην εισαγωγή, το Unity Network έχει στόχο να μπορέσει να αντιμετωπίσει όλα τα παραπάνω προβλήματα παρέχοντας παραπάνω προνόμια στου συνδεμένους χρήστες του και εκπροσωπεί τις παρακάτω αρχές.

Επωνυμία

Το δίκτυο αποδίδει διαφορετικές διευθύνσεις όχι μόνο για κάθε χρήστη αλλά **για κάθε συσκευή** του. Οι διευθύνσεις αυτές δεν αλλάζουν δηλαδή είναι **στατικές**. Ένα μηχάνημα κατά τη σύνδεση θα πάρει τη διεύθυνση όπου έχει κατοχυρωμένη. Επίσης ένας χρήστης θα συνδέεται πιστοποιώντας τον εαυτό του βάση του ζευγαριού ιδιωτικού δημοσίου κλειδιού

Ελευθερία

Το δίκτυο δίνει τη δυνατότητα **σε κάθε συσκευή του χρήστη να μπορεί να διανέμει οποιαδήποτε υπηρεσία** επιθυμεί χωρίς να είναι απαραίτητη η οποιαδήποτε έγκριση 3^{ου} (**be a full host**). Αυτό σημαίνει ότι κάθε συσκευή μπορεί να είναι και client και server σε οποιαδήποτε υπηρεσία. Αυτό το επιτυγχάνει καθώς είναι σε θέση να δέχεται υπολογιστές με μειονεκτική θέση και να τους κάνει host στο Unity, η όλη λογική ορίζεται ως NAT Traversing. Κανένας ενδιάμεσος κόμβος δεν είναι σε θέση να περιορίζει την κίνηση όπως πχ. το διαχειριστή ενός τοπικού δικτύου ή τον διαχειριστή μιας gateway ή ένα ISP. Πρωτόκολλα όπως UPnP IGD όπου **δεν είναι ασφαλή** δεν είναι αναγκαία.

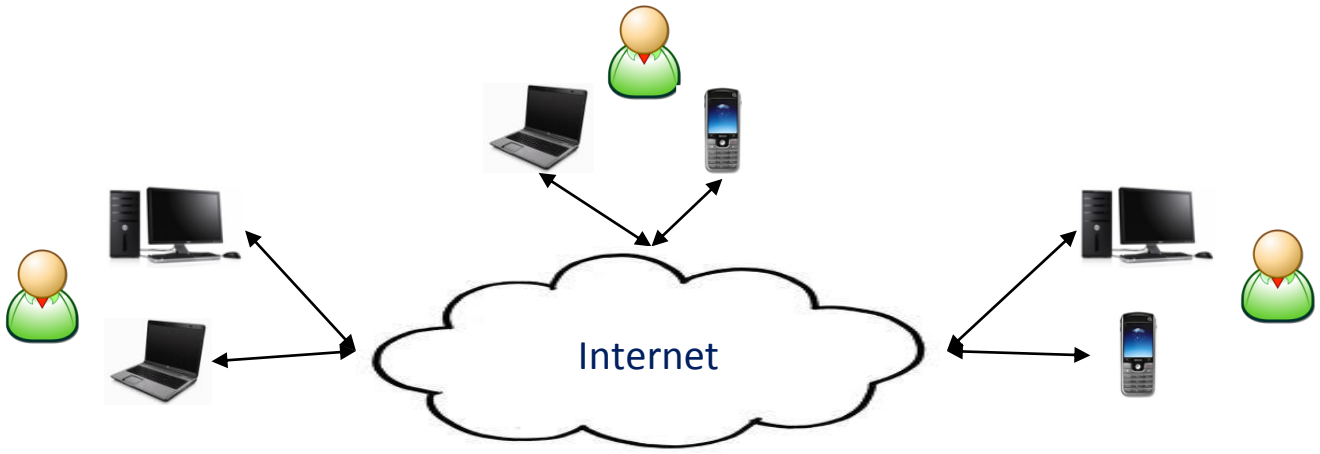
Ασφάλεια

Το δίκτυο παρέχει σε κάθε χρήστη ένα ζευγάρι ασύμμετρων κλειδιών κατά προεπιλογή. Ο χρήστης κατά την είσοδο του στο δίκτυο θα πιστοποιείται βάση του ιδιωτικού του κλειδιού. Επίσης εφόσον κάθε χρήστης στο δίκτυο θα κατέχει ένα ζευγάρι κλειδιών θα ενισχυθεί η κρυπτογραφημένη επικοινωνία. Επιπλέων η ροή μεταφοράς της εικονικής κίνησης θα είναι μη προτυποποιημένη σε απλά UDP encrypted datagramms.

Επομένως ο χρήστες του Unity μπορούν να βιώσουν πραγματική επικοινωνία χωρίς αστυνόμευση και με ασφάλεια!

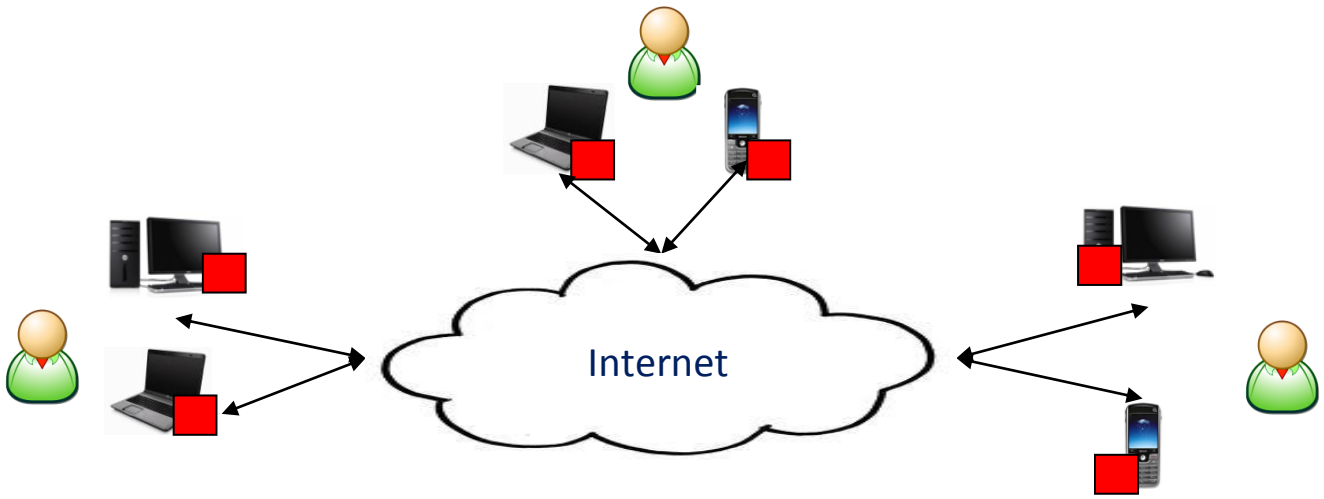
1.5 Η Αίσθηση του δικτύου

Πολλοί χρήστες που έχουν συσκευές δικτύου θέλουν να επικοινωνήσουν μεταξύ τους. Οι συσκευές των χρηστών αρχικά έχουν μια τυπική σύνδεση στο διαδίκτυο με πολλούς περιορισμούς.



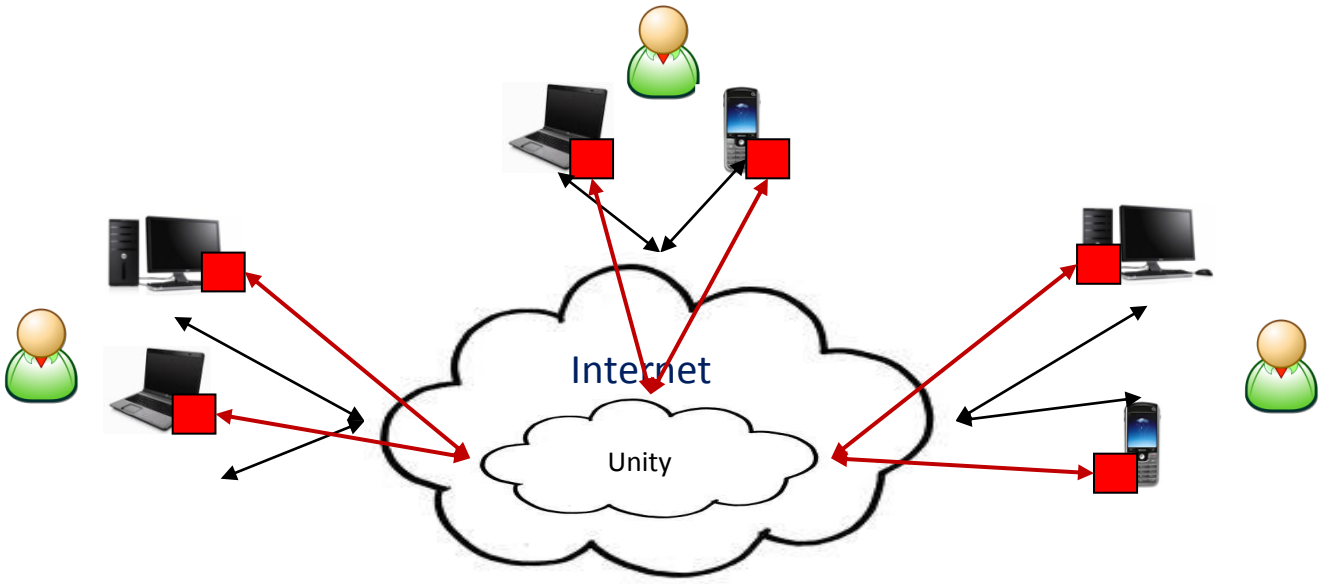
Εικόνα 6 network feeling 1

Ο χρήστες εκτελούν το πρόγραμμα του RedNode σε κάθε συσκευή τους. Αφότου δώσουν το username τους και υποδείξουν το ζευγάρι δημόσιου και ιδιωτικού κλειδιού που κατέχουν συνδέονται στο δίκτυο.

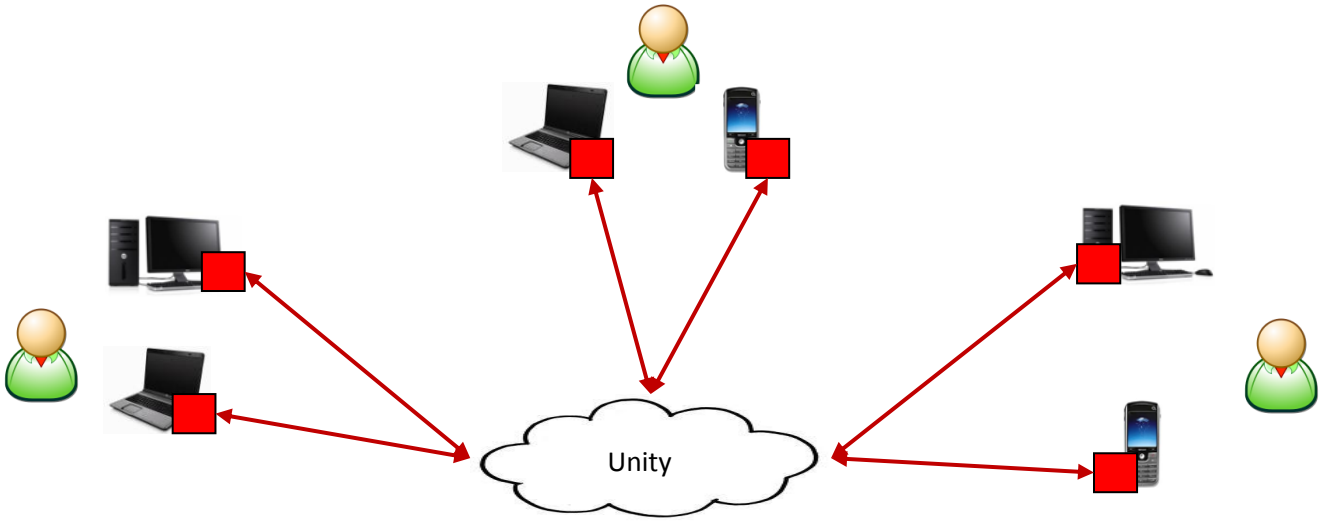


Εικόνα 7 network feeling 2

Μετά τη σύνδεση είναι ελεύθερες να διανέμουν οτιδήποτε επιθυμούν και εντοπίζονται από τις κατοχυρωμένες IP addresses για το κάθε μηχάνημα.



Εικόνα 8 network feeling 3 - Τα μαύρα βελάκια δηλώνουν τις φυσικές συνδέσεις των μηχανημάτων στο διαδίκτυο ενώ τα κόκκινα τις εικονικές



Εικόνα 9 network feeling 4 - Το εικονικό δίκτυο όπου πλέον σχηματίστηκε

1.6 Η αρχιτεκτονική της πλατφόρμας

Γενικά

Στόχος του Unity ως πλατφόρμα είναι να παρέχει ένα ευρείας εμβέλειας **virtual network space** και μεγάλης χωρητικότητας σε συνδεμένους χρήστες. Ουσιαστικά **αποδεικνύει** από αρχιτεκτονικής οπτικής και μέσα από την υλοποίηση του οτι μπορεί να υπάρξει ένα μεγάλο και εκτενές **virtual network space** όπου μπορούν να συνδεθούν πολλοί περισσότεροι χρήστες από ένα απλό VPN room, σε αντίθεση με τις υπόλοιπες VPN πλατφόρμες όπου δημιουργούν μικρής και τοπικής εμβέλειας rooms. Το **virtual network space** όπου δημιουργείται συντηρείται από αρκετούς κόμβους και αυτό σημαίνει οτι είναι ανθεκτικό σε θάνατο καθώς εάν πεθάνει ένας κόμβος (BN) υπάρχουν άλλοι στην πλατφόρμα και το εικονικό δίκτυο δεν χάνεται. Το εικονικό δίκτυο λειτουργεί ως ολοκληρωμένη πλατφόρμα προσφέροντας στους χρήστες όλες τις απαραίτητες διαδικασίες όπως εγγραφή και διαμοιρασμό κλειδιών. Τέλος ένα άλλο πολύ σημαντικό χαρακτηριστικό του είναι οτι τα λειτουργικά του μέρη είναι όσο το δυνατόν πιο ανεξάρτητα και από HW αλλά και από το OS ενός host. Αυτό επειδή δεν εισχωρεί σε βαθιά εγκατάσταση στο OS αλλά και επειδή τρέχει σε πολλά διαφορετικά OS , αυτό το χαρακτηριστικό κάνει την διαχείριση της πλατφόρμας **πιο λογική** και **αντικειμενοστραφή** κάνοντάς τη να ασχολείται μόνο με τα ανώτερα και λογικά επίπεδα.

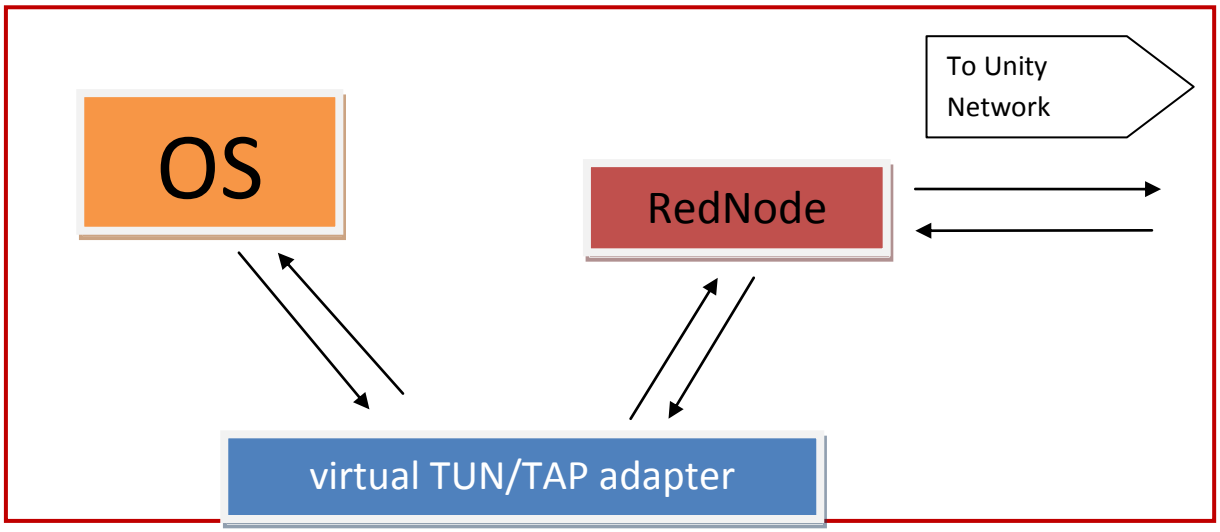
Η πλατφόρμα αποτελείται από 3 είδη κόμβων:

- Τον RedNode όπου είναι το πρόγραμμα host/client πελάτη
- Τον BlueNode όπου αναλαμβάνει τον ρόλο του VPN server και διακινεί κίνηση δικτύου
- Τον Tracker όπου συντονίζει πολλούς BN να δουλεύουν συλλογικά και τους RNs για το πού θα συνδεθούνε

Το Μηχάνημα Πελάτη & Ο RedNode

Το μηχάνημα πελάτης έχει 2 στοιχεία όπου του επιτρέπουν να συνδεθεί στο δίκτυο. Το πρώτο είναι ο virtual TUN/TAP adapter και το πρόγραμμα του RedNode.

Ο virtual TUN/TAP adapter έχει γίνει αρκετά διάσημος καθώς από τη μία αντιμετωπίζεται από το λειτουργικό ως μια πραγματική κάρτα δικτύου και από την άλλη ένα πρόγραμμα μπορεί να αποκτήσει πρόσβαση σε αυτόν κάνοντας του read και write byte arrays δηλαδή πακέτα δικτύου. Στην μία περίπτωση διαβάζει οτι γράφεται από το OS και στην άλλη γράφει πακέτα. Η εφαρμογή όπου διαχειρίζεται τον TUN/TAP είναι ο RedNode όπου κάνει από τη μια read τα αιτήματα του OS και τα στέλνει στο Unity και από την άλλη Write οτι καταφτάνει. Ο RedNode συνδέεται ως client με το δίκτυο και αυτό του επιτρέπει να δέχεται και να στέλνει κίνηση μέσα από πακέτα UDP. Φυσικά την κίνηση από και προς του Unity τη μεταφέρει διαμέσω του πραγματικού adapter υποδύωντας τον client!

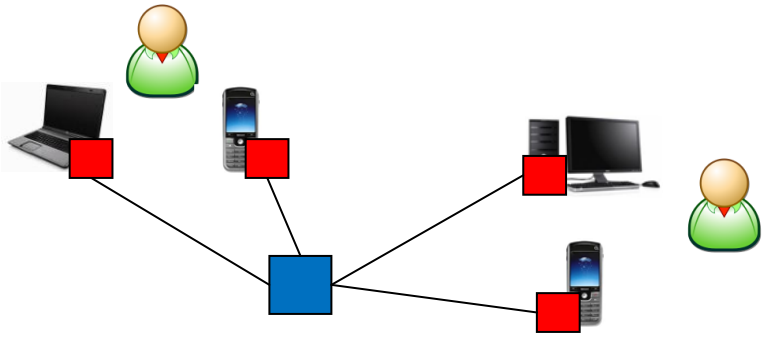


Εικόνα 10 Client machine

Ένα αποτέλεσμα όλης αυτής της μανούβρας είναι ο RedNode εφόσον κάνει read και write από τον TUN/TAP να μπορεί να στείλει και πακέτα δικτύου όπως ARPS και DHCP requests γεγονός που κάνει το δίκτυο να είναι σε θέση να διαχειρίζεται πιο ικανά όλη την κίνηση και τους hosts.

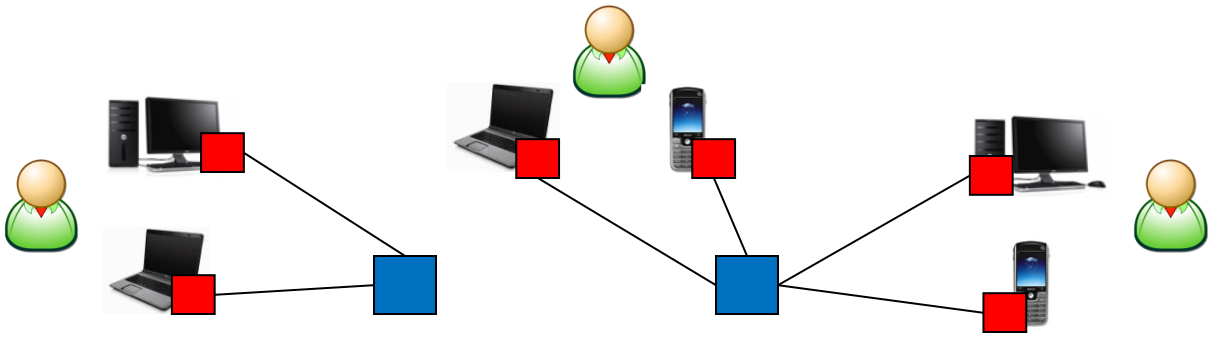
Ο BlueNode

Σε κάθε περιοχή υπάρχει ένας BlueNode όπου κοντινοί RedNodes (με μικρό ping) συνδέονται σε αυτόν και γίνονται μέλος του δικτύου. Ένας BlueNode αρχικά πιστοποιεί τους RedNodes κατά την εισοδό τους. Στη συνέχεια δέχεται UDP datagrams από τους RedNode τα οποία μέσα περιέχουν εικονικά πακέτα δικτύου, τα ανοίγει, βλέπει τον προορισμό τους και τα ανακατευθύνει σε κάποιον άλλο συνδεμένο RedNode. Ο RedNode τα ανοίγει με τη σειρά του και τα φορτώνει στην κάρτα ώστε να γίνουν λειτουργικά πακέτα δικτύου. Μέχρι τώρα έχει περιγραφεί μια απλή υλοποίηση VPN server.

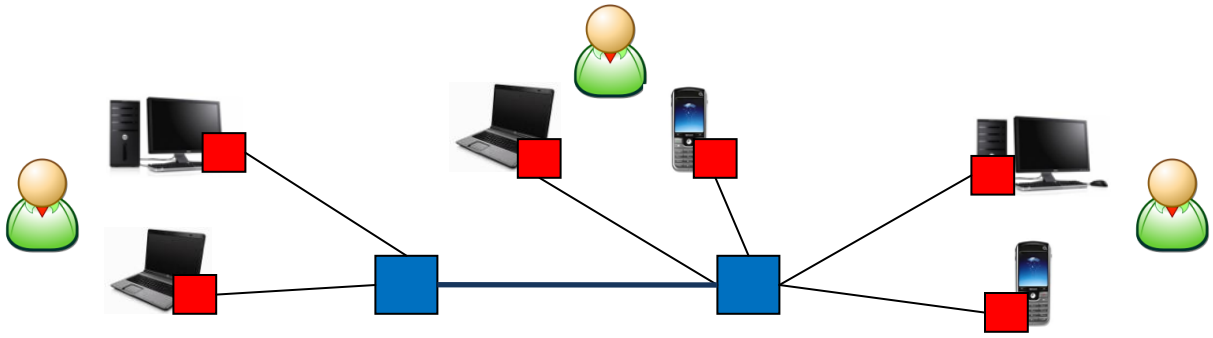


Εικόνα 11 basic unity architect 1

Στη συνέχεια διαφοροποιούμαστε από ένα κλασικό VPN network. Πολλοί BlueNodes γνωρίζονται μεταξύ τους και εάν ένας RN στείλει σε ένα άλλο όχι τοπικό RN στον BN όπου είναι συνδεμένος, τοτε ο ίδιος ο BN θα ψάξει σε ποιον άλλο BN είναι πελάτης ο συγκεκριμένος RN και θα στέλνει εκεί τα πακέτα.



Εικόνα 12 δύο BNs όπου ο καθένας εξυπηρετεί διαφορετικούς πελάτες



Εικόνα 13 δύο BNs όπου έχουν συσχετιστεί και όλοι οι RNs μπορούνε και αλληλεπικοινωνούν

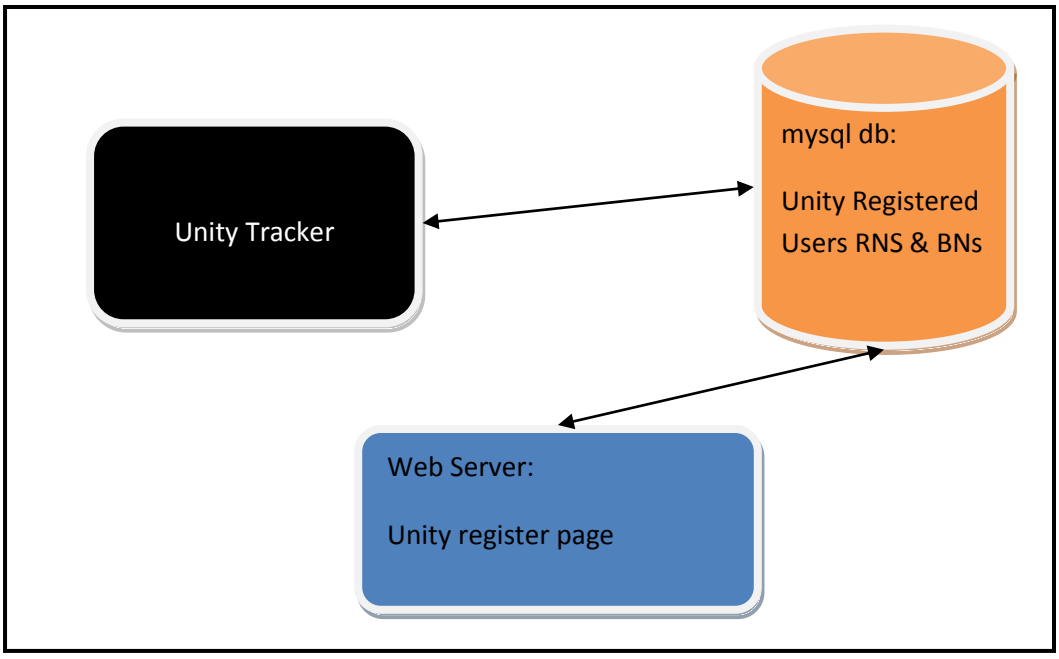
Ο BlueNode αποτελείται από την εφαρμογή του BN όπου είναι γραμμένη σε java και ο υπολογιστής όπου κάνει host πρέπει να έχει τον έλεγχο των θυρών του.

Ο Unity tracker & registry

Οι BNs χρησιμοποιούν ένα κεντρικό Tracker για να εντοπίζονται μεταξύ τους.

- tracker γνωρίζει την IP address του κάθε συνδεμένου στο δίκτυο BN
- Ο tracker θυμάται ποιός RN έχει συνδεθεί σε ποιό BN
- Ο tracker διαχειρίζεται τα δημόσια κλειδιά της πλατφόρμας και θεωρείται έμπιστη οντότητα
- Μέσα από τον Tracker δεν περνάει κίνηση δικτύου και ούτε μπορεί να επιλέξει ποιός θα συνδεθεί και που
- Απαντάει σε συγκεκριμένα ερωτήματα και αυτά ανάλογα με την ιδιότητα αυτού που ρωτάει BN ή RN

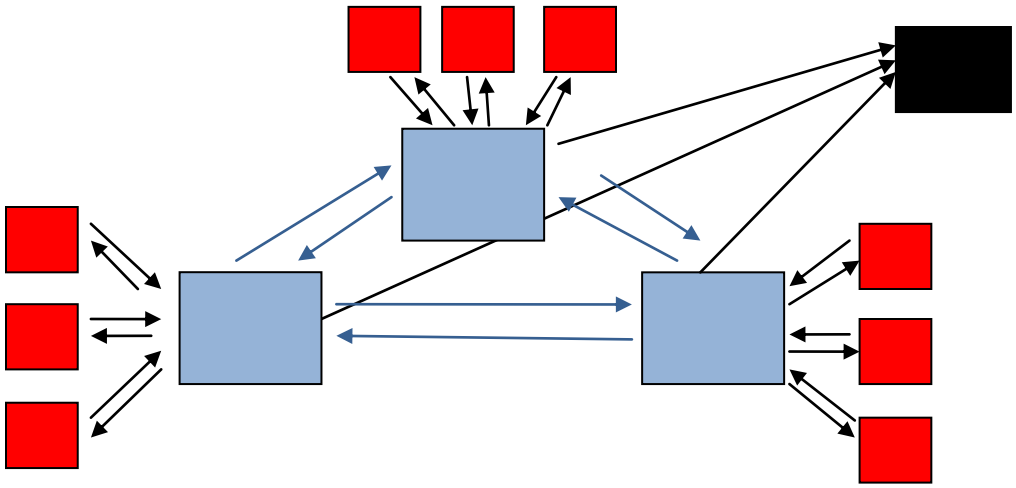
Ο Tracker node αποτελείται από το πρόγραμμα του Tracker όπου είναι γραμμένο se java, μια βάση δεδομένων mysql όπου κρατάει τους εγγεγραμμένους χρήστες και στο ίδιο μηχάνημα θεωρούμε οτι βρίσκεται και ο http server όπου φιλοξενεί τη σελίδα των εγγραφών για το δίκτυο



Εικόνα 14 Tracker Node

1.7 Η Πλήρης εικόνα του δικτύου

Το δίκτυο σε τελική μορφή παρουσιάζει την παρακάτω εικόνα.



Εικόνα 15 Η πλήρης εικόνα του δικτύου!

- Οι RNs συνδέονται στους BNs
- Ο Tracker θυμάται ποιός συνδέθηκε που
- Οι BNs μέσω του Tracker καταφέρνουν και αλληλοεντοπίζονται
- Με αποτέλεσμα όλοι οι RN να ενδοεπικοινωνούν ανεξάρτητα με το που έχουν συνδεθεί

Ένας RedNode κατά τη σύνδεση του στο Unity μπορεί να επιλέξει να ρωτήσει τον Tracker για ένα κοντινό BN ή να συνδεθεί σε κάποιον συγκεκριμένο BN χωρίς να ρωτήσει τον Tracker.

Κατανομή Δικαιωμάτων

Στο Unity καμία οντότητα δεν έχει απόλυτο έλεγχο αλλά αντιθέτως όλες έχουν σχετικό.

- Tracker προσφέρει πληροφορίες συντονισμού της πλατφόρμας αλλά δεν δρομολογεί κίνηση δικτύου
- Οι BlueNodes δρομολογούν κίνηση αλλά δεν έχουν τον έλεγχο της πλατφόρμας
- Οι RedNodes έχουν τον έλεγχο του εαυτού τους δηλαδή μπορούν να συνδεθούν ή να αποσυνδεθούν όποτε θέλουν και σε όποιον BN επιθυμούν

2. Χρήση του Unity Network

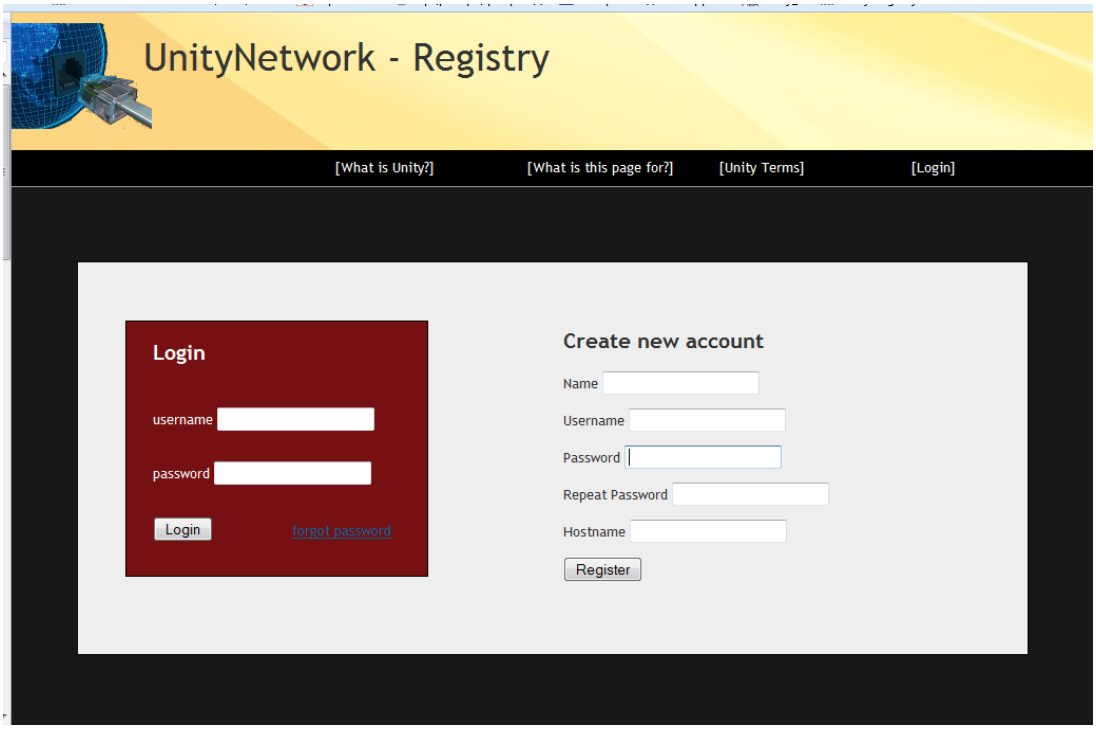
Σε αυτό το κεφάλαιο περιγράφονται όλες οι δυνατές λειτουργίες καθώς και το γραφικό περιβάλλον των προγραμμάτων του μπλε και του κόκκινου κόμβου. Περιγράφονται επίσης όλες οι απαραίτητες ενέργειες που πρέπει να κάνει ένας χρήστης για να συνδεθεί όσο και ένας διαχειριστής για να στήσει την πλατφόρμα ή μέρος της.

2.1 Χρήστης

Το λογισμικό του χρήστη είναι πολύ απλό στη χρήση ώστε ο χρήστης να μην έρχεται αντιμέτωπος με πολύπλοκες διαδικασίες εγκατάστασης και σύνδεσης.

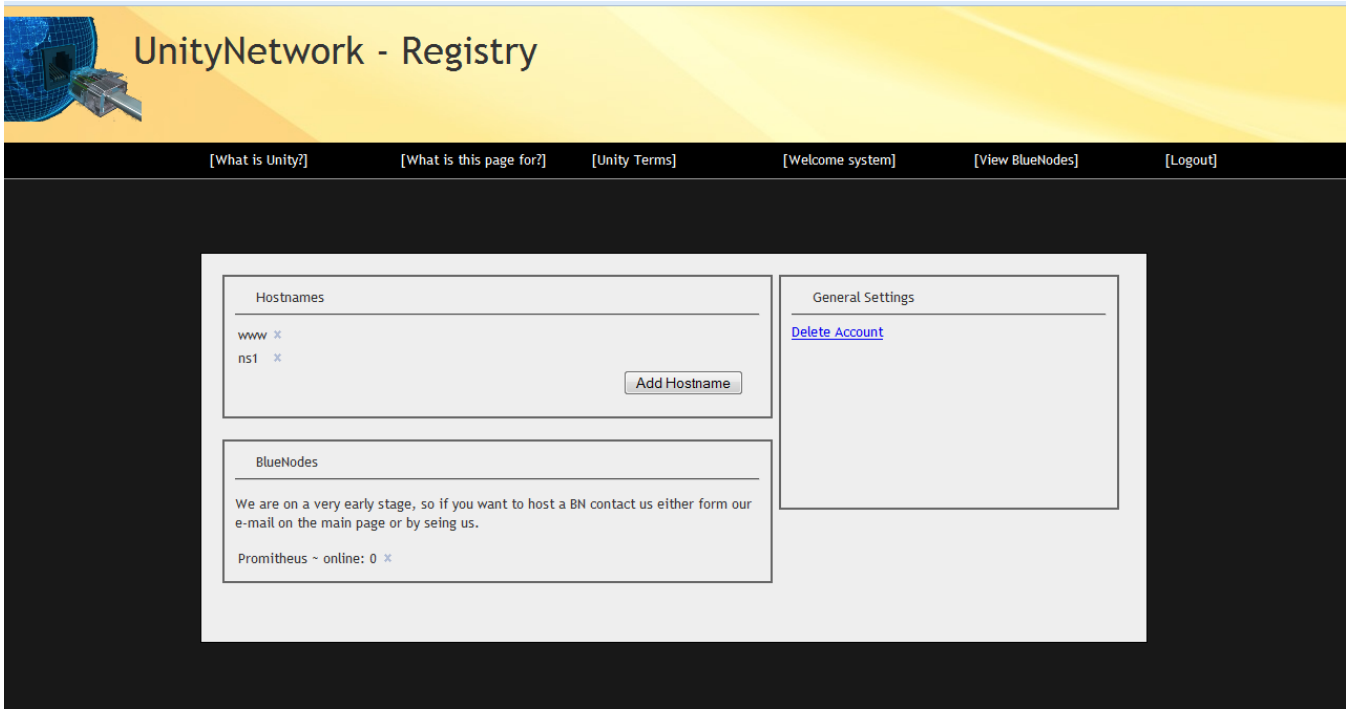
Εγγραφή στο δίκτυο

Η εγγραφή είναι πολύ απλή και έχει βασιστεί σε ήδη υπάρχοντα πρότυπα εγγραφής. Ο χρήστης απλά πηγαίνει στην κεντρική σελίδα εγγραφών και κάνει register.



Εικόνα 16 Registry page

Με το register θα του ζητηθεί το πρώτο hostname που αντιστοιχεί στον υπολογιστή που θέλει να συνδέσει στο δίκτυο. Στη συνέχεια μπορεί να προσθέσει και άλλα hostnames από την σελίδα των προτιμήσεων. Με την ολοκλήρωση αυτών των βημάτων μπορεί πλέον να συνδεθεί στο δίκτυο.



Εικόνα 17 registry page settings

Εγκατάσταση εφαρμογής

Προαπαιτούμενα:

Οποιοδήποτε λειτουργικό σύστημα και αν χρησιμοποιεί ο χρήστης θα πρέπει να έχει εγκατεστημένο το Java Runtime Environment (JRE) από την έκδοση 6 και πάνω.

Αν η εγκατάσταση γίνει σωστά ο χρήστης κάνοντας διπλό κλικ στο εικονίδιο της εφαρμογής θα μπορεί να ανοίξει την εφαρμογή.

Windows:
Στα windows ο χρήστης θα πρέπει επιπλέον να εγκαταστήσει τον driver που έρχεται μαζί με την εφαρμογή ώστε να εγκατασταθεί η εικονική κάρτα δικτύου που χρειάζεται.

Linux:

Στα linux ο χρήστης θα πρέπει να έχει ήδη εγκατεστημένο ένα εικονικό προσαρμογέα, συνήθως είναι εγκατεστημένος αλλά αν δεν είναι μπορεί να εγκατασταθεί με την εντολή

Modprobe tun

Θα πρέπει να έχει το πρόγραμμα για γραμμή εντολών dhclient

Θα πρέπει να έχει δικαιώματα διαχειριστή

Mac:

Τα mac αν και δεν έχουν δοκιμαστεί αν ο χρήστης μπορεί να εγκαταστήσει τον adaptor και το JRE θα μπορεί να έχει πρόσβαση στο δίκτυο.

Εφαρμογή:

Τελικό βήμα είναι ο χρήστης να κατεβάσει την εφαρμογή lvl3RedNode

Εκτέλεση εφαρμογής χρήστη και σύνδεση

Εκτέλεση:

Windows:

Στα windows αρκεί ένα διπλό κλικ στο RedNode και η εφαρμογή ανοίγει

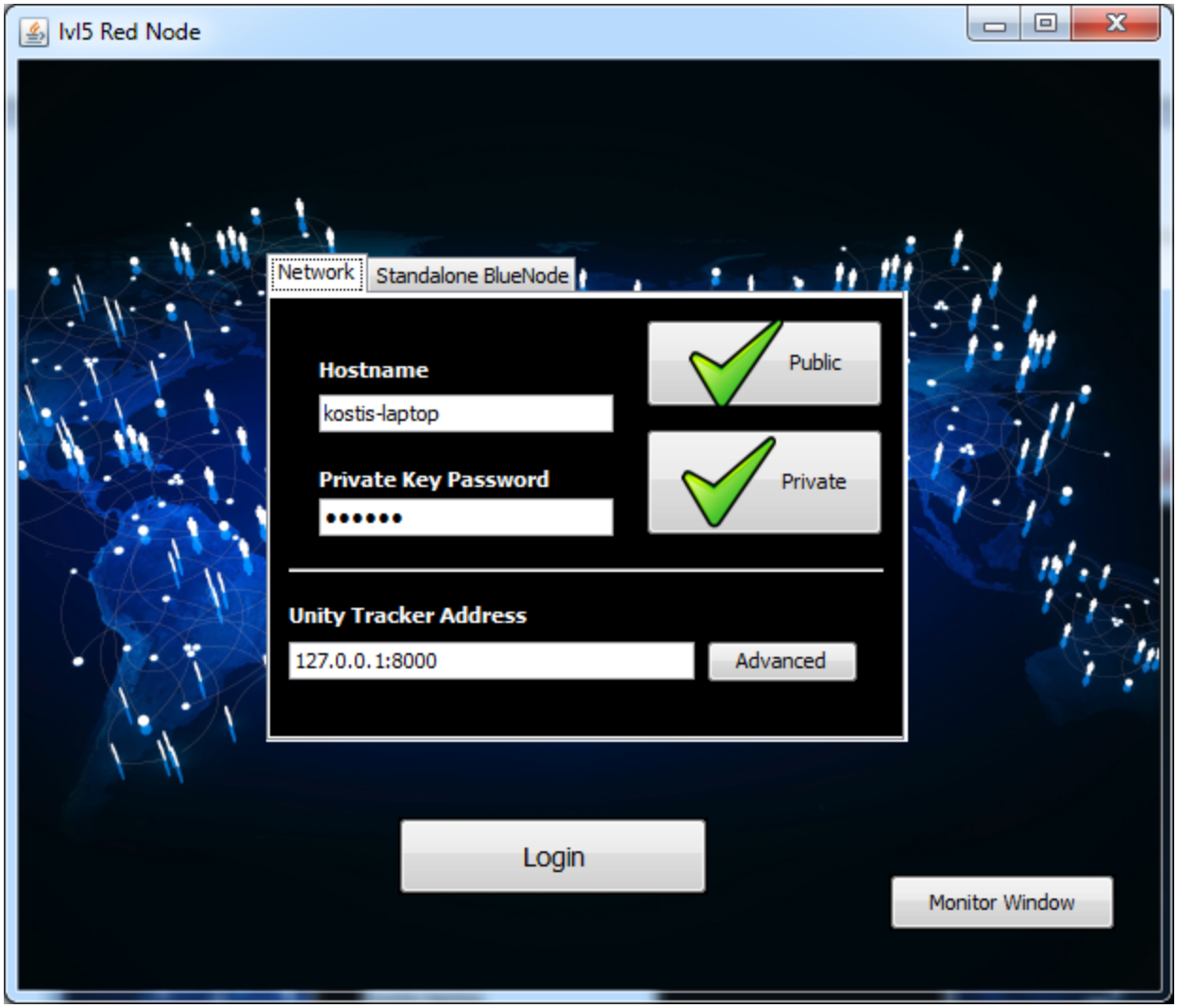
Linux:

Στα linux ο χρήστης θα πρέπει να εκτελέσει το lvl3RedNode με δικαιώματα διαχειριστή ώστε να δημιουργήσει μια κάρτα και να πάρει τον έλεγχο της, επίσης το πρόγραμμα θα πρέπει να έχει δικαιώματα εκτελέσιμου προγράμματος.

Μια εύκολη εκτέλεση του lvl3RedNode μέσω της γραμμής εντολών είναι:

```
sudo java –jar  /path/lvl3RedNode.jar
```

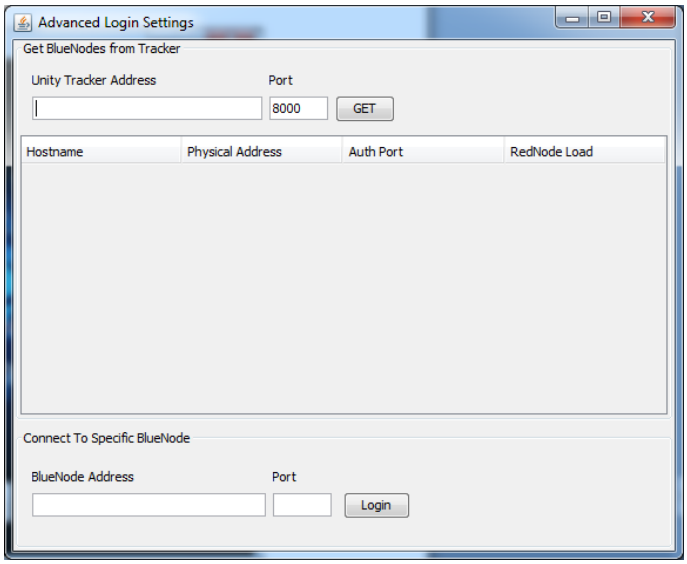
Μόλις εκτελεστεί η εφαρμογή ο χρήστης θα βρεθεί στο αρχικό παράθυρο



Εικόνα 18 Figure 26 lvl3RedNode Αρχικό παράθυρο

Εδώ ένας χρήστης έχει δύο επιλογές:

- Η πρώτη επιλογή είναι να συνδεθεί σε μια πλατφόρμα unity. Για να το κάνει αυτό θα χρειαστεί ένα ζευγάρι ιδιωτικού δημοσίου κλειδιού για το Hostname όπου επιθυμεί να χρησιμοποιήσει και το password για το συγκεκριμένο ιδιωτικό κλειδί!



Εικόνα 19 Advanced

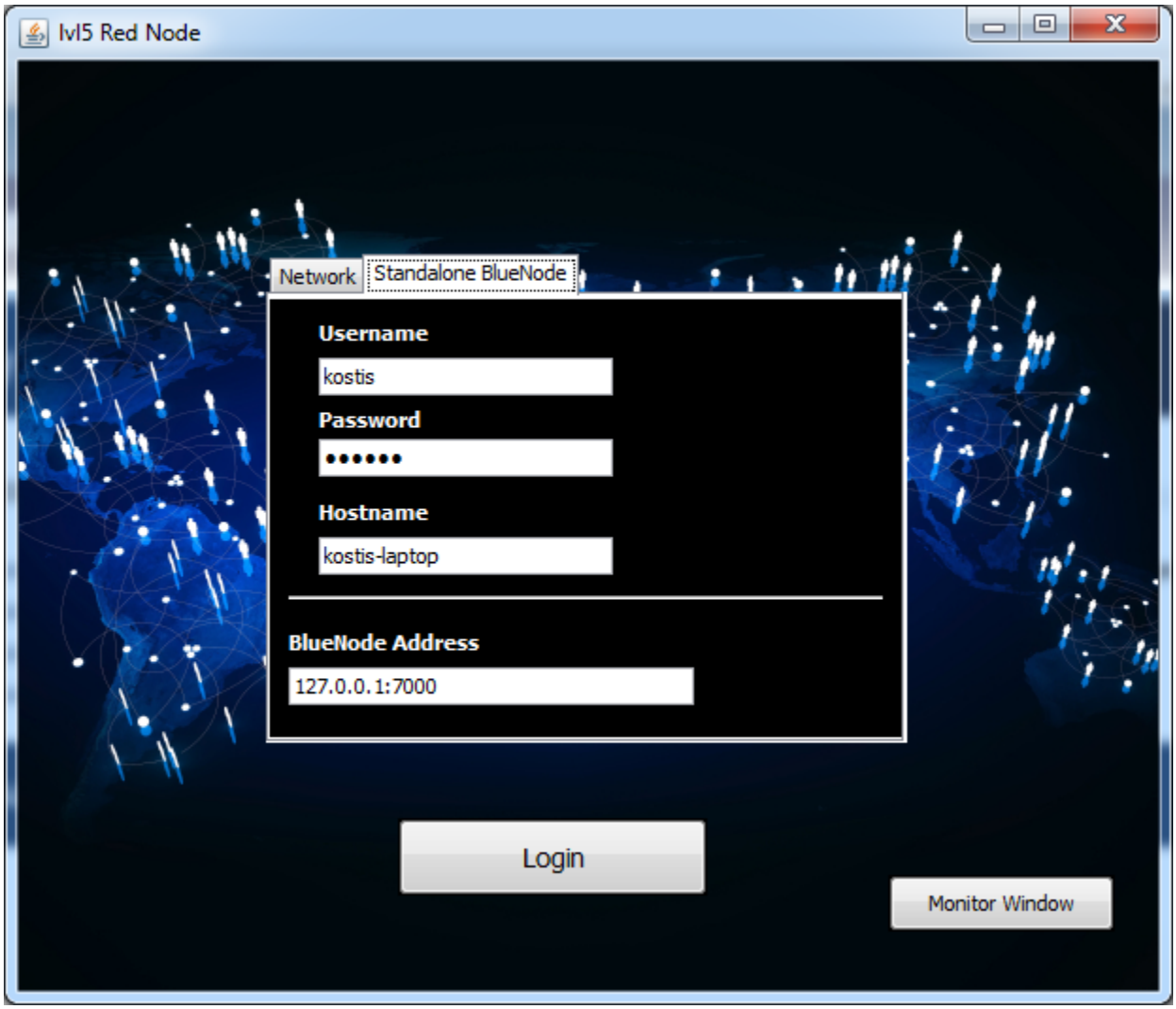
Μετά εισάγει τη διεύθυνση ή το domain του κεντρικού Tracker αν επιθυμεί να του απονέμει αυτόματα ένα BN ο tracker

Είτε πάει στην καρτέλα advanced.

Σε αυτή την καρτέλα είτε μπορεί να ζητήσει από ένα Tracker όλους του γνωστούς του BNs και να συνδεθεί σε ένα συγκεκριμένο

Είτε μπορεί να εισάγει μια συγκεκριμένη διεύθυνση BN για το συγκεκριμένο δίκτυο

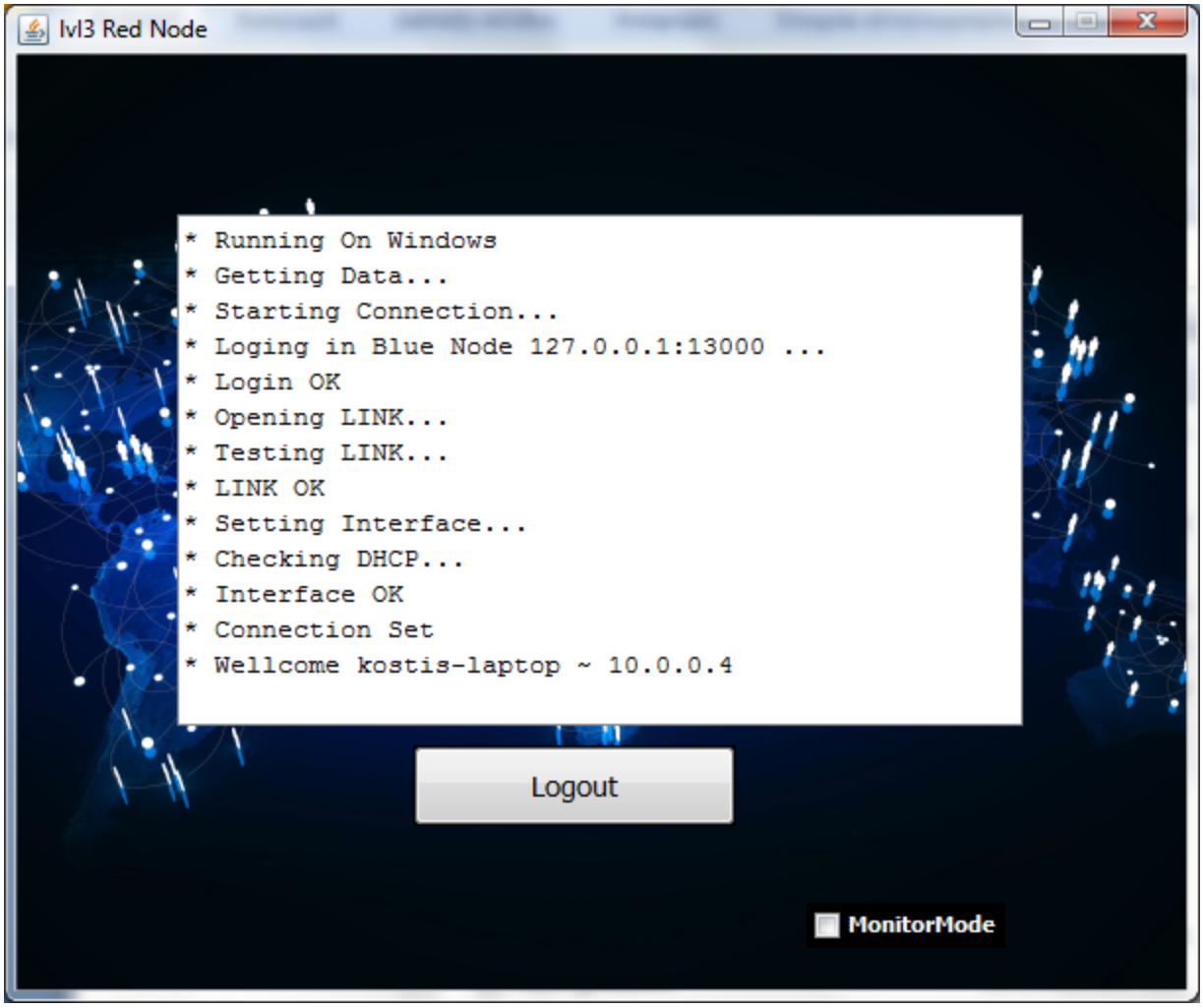
Η δεύτερη επιλογή είναι να συνδεθεί σε ένα τοπικό BN. Σε αυτή την περίπτωση γίνεται πιστοποίηση με password και ο χρήστης δηλώνει username password hostnameκαι τη διεύθυνση του BN.



Εικόνα 20 Figure 26 lvl3RedNode Αρχικό παράθυρο No Network

Σύνδεση:

Μόλις πατήσει login θα αρχίσει η διαδικασία σύνδεσης με τον BN όπου επιλέχθηκε αυτόματα ή επιλέχτηκε χειροκίνητα… Ο χρήστης μπορεί να δει την εξέλιξή της από τα μηνύματα πληροφοριών και δεν ξαναέρχεται σε επαφή με τον Tracker.



Εικόνα 21 lvl3RedNode - logged in

Στη συνέχεια θα μπορεί να ανοίξει οποιαδήποτε εφαρμογή hosting η εφαρμογή client επιθυμεί.

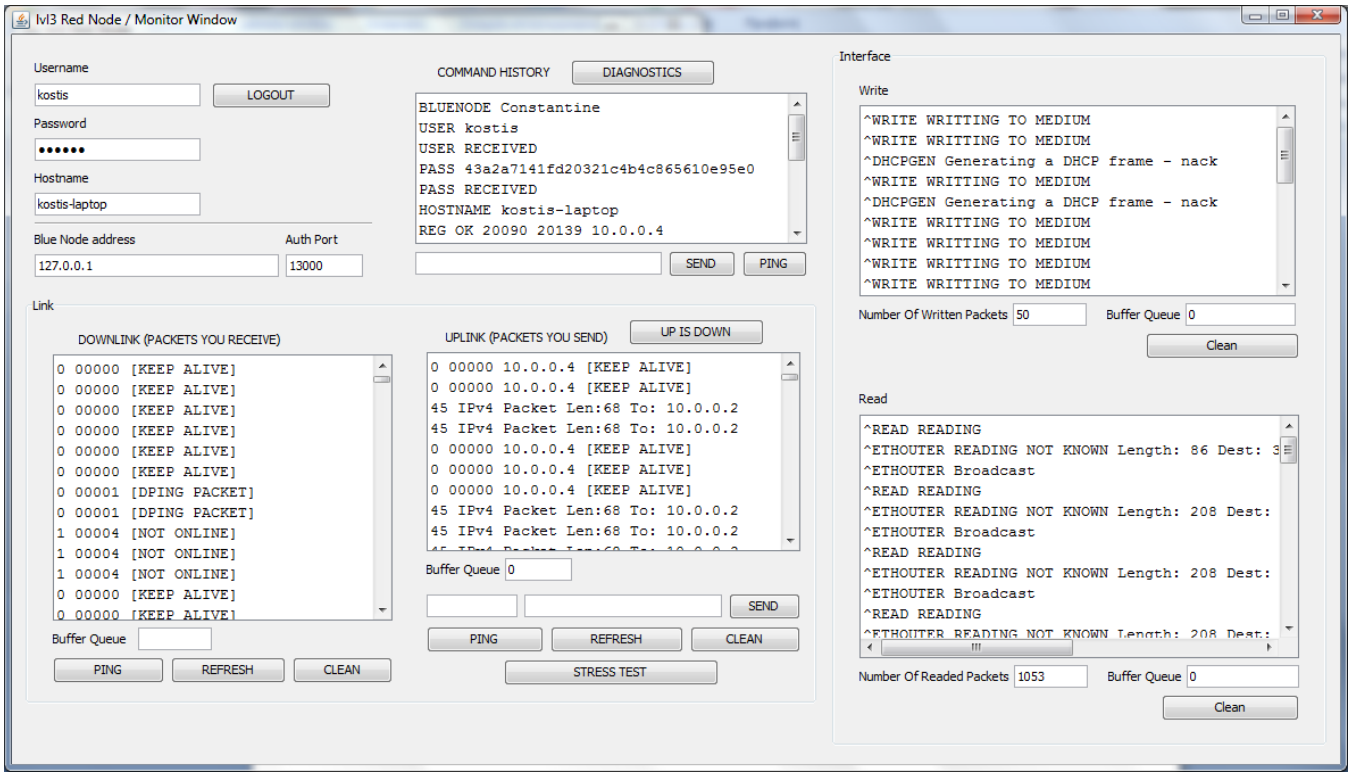
Μερικές είναι:

- ftp server client
- Http server client
- Torrents, p2p networks
- Game servers
- Chat servers όπως το IRC
- Real VoIP
- Οποιασδήποτε άλλης μορφής δικτυακή υπηρεσία

Ο χρήστης δεν είναι υποχρεωμένος ούτε να προωθήσει θύρες στο router του αλλά ούτε και να μοιραστεί τις θύρες με άλλους υπολογιστές στο τοπικό δίκτυο. Επίσης θα πρέπει να ρυθμίσει το firewall του για το τι επιθυμεί να διαμοιράζει στο τοπικό του δίκτυο και τι στο unity.

Connection Debugging

Ο χρήστης αν επιθυμεί μπορεί να ανοίξει το monitor window από κάτω δεξιά και να δει την κίνηση του δικτύου που δέχεται και παίρνει. Από το παράθυρο που θα εμφανιστεί μπορεί να πάρει πληροφορίες και να κάνει tests.



Εικόνα 22 RedNode Monitor View

2.2 BlueNode Host

Όποιος επιθυμεί μπορεί να κάνει host ένα Blue Node.

στην περιοχή του, μεγαλώνοντας το εύρος του δικτύου μιας πλατφόρμας Unity. Με περισσότερους Blue Nodes γίνεται καλύτερη κατανομή του δικτύου και συνεπώς μεγαλύτερες ταχύτητες για τα μέλη του.

Προαπαιτήσεις Blue Node Hosting

Ο διαχειριστής:

- Πρέπει να έχει πλήρη δικαιώματα στο τοπικό του δίκτυο να ανοίγει θύρες
- πρέπει να γνωρίζει να προωθεί θύρες στο router του
- να τρέχει JRE στο μηχάνημά του

Αρχικά βήματα hosting

Επειδή ο BN θεωρείται ένα πρόγραμμα υπηρεσία όλες οι επιλογές και οι ρυθμίσεις του γίνονται πριν εκτελεστεί η εφαρμογή από το config file του. Αυτό συμβαίνει καθώς όποιες υπηρεσίες κατασκευάζουμε καλό θα ήταν να είμαστε σίγουροι από την αρχή για την ορθότητα τους και να υπάρχει ένας μεγάλος έλεγχος κατά την εκκίνηση της υπηρεσίας. Με αυτό τον τρόπο έχουμε πιο σταθερές και πιο ανθεκτικές υπηρεσίες!

Το config file το βρίσκουμε στο ίδιο Dir με τον BN και έχει όνομα **bluenode.conf**. Παρακάτω μπορούμε να το δούμε. Οι γραμμές σχόλια έχουν # και κάθε επιλογή εξηγείται απο πάνω της.

Ένας BN μπορεί να είναι αυτόνομος χωρίς Tracker ή να είναι μέλος της πλατφόρμας.

Εάν είναι μόνος του μπορεί να δηλώσει μια λίστα χρηστών στο αρχείο **users.list** ή να επιτρέπει ελεύθερη σύνδεση σε όποιον χρήστη γνωρίζει την διεύθυνση του.

Έαν θέλουμε ο BN μας να μπορεί να ακούσει στο internet για RNs τότε πρέπει να κάνουμε port forward στο router μας τις θύρες που δηλώσαμε.

Άλλες επιλογές είναι: No Gui (χωρίς γραφικό περιβάλλον για τερματικό)

RedNode Limit (για όριο χρηστών)

κα.

Επίσης ένας BlueNode πρέπει να δηλωθεί στο register page όπου του απονέμεται ένα ζευγάρι κλειδιών RSA-2048 για χρήση με την πλατφόρμα. Η βάση δεδομένων κρατάει το δημόσιο κλειδί και ο χρήστης κατεβάζει το ζευγάρι ως αρχεία. Στη συνέχεια τα τοποθετεί στο dir του BN ώστε να είναι σε θέση να πιστοποιηθεί από το δίκτυο.

Ακόμα μπορούμε να δημιουργήσουμε ένα αρχείο **bluenode.log** όπου θα καταγράφει το ιστορικό χρήσης.

Το αρχείο bluenode.conf

```
################################
#   Blue Node Config File     #
################################

#please do not comment any variable nor remove any. this will result in error
#instead only change the value to an appropriate input as described

#use unity network true ~ false (false means a standalone working BN, true means
#that the BN works on a unity network with a tracker and other BNs)
network = false

#if you use lets determine the central tracker
#with an ip address or with a hostname or domain
#and the central auth port of the tracker 8000 is default

UnityTracker = 192.168.2.11
UnityTrackerAuthPort = 8000

#choose to autologin to the network
#by default is disabled because you can click it from the GUI
AutoLogin = true

#then set the hostname of the BN
#hostname must be registered with central authority if you use one
#and the local auth port 7000 default

Hostname = Pakhs
AuthPort = 7000

#use list true ~ false (false means any client can log in as he states himself ~ true means only a user in users.list can login)
#users.list holds the file
uselist = false

#now give a udprange
#for the RN tunnels where the packets will be forwarded
udpstart = 20000
udpend = 22000

#set the limit of RNs for this BN
RedNodeLimit = 20

#set GUI or command line
#with true or false
UseGUI = true

#choose to verbose traffic in command line
#by default is disabled because you can monitor it
#in GUI but it useful if you are under remote terminal
ConsoleTraffic = false

#logging in bluenode.log
#true ~ false
log = true
```

Το αρχείο users.list

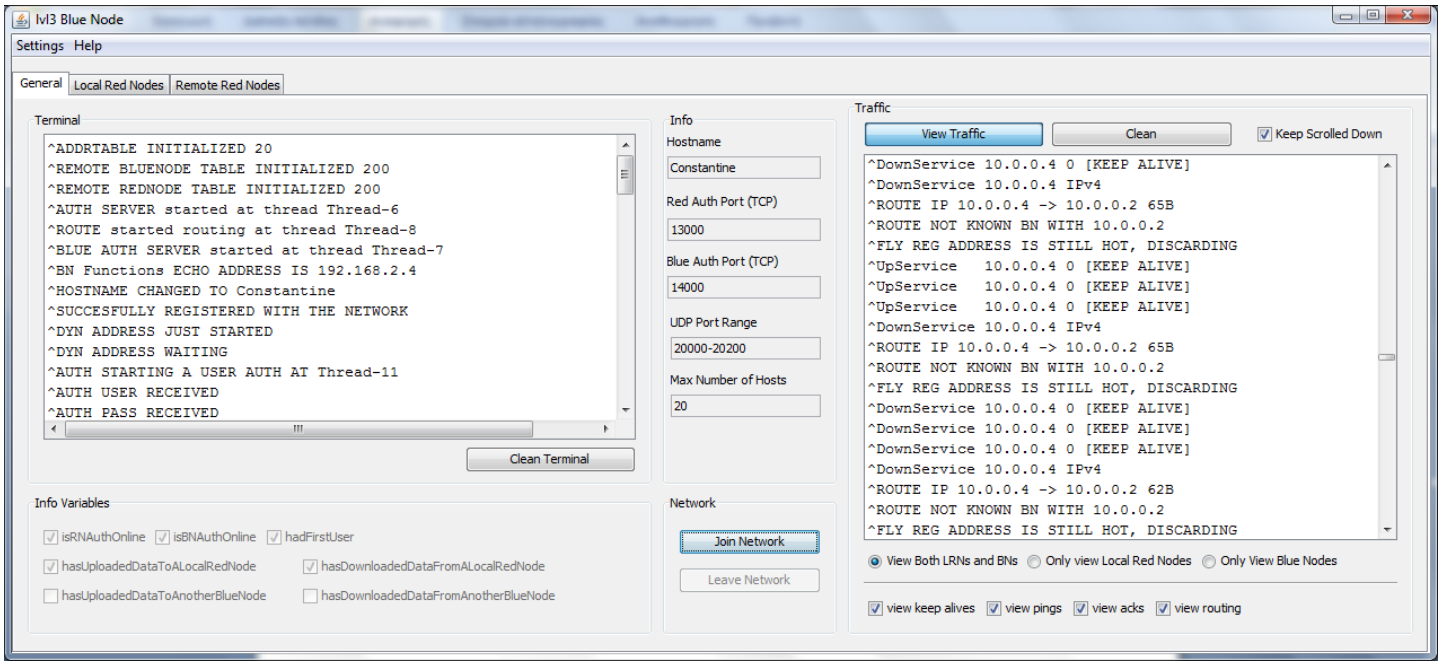
```
#############################################################################
# this is a user list for a BN                                              #
# use like....                                                              #
#                                                                           #
#       username[space]password[space]hostname[space]virtual_ip_to_give      #
# ex:    pakhs   123456  pakhs-pc   10.0.0.1                                #
#                                                                           #
# make sure all the clients share ips from network 10.0.0.0/8               #
# this file must not have empty lines!!!!                                   #
#############################################################################
kostis  12345   kostis-laptop   10.0.0.1
kostis  12345   kostis-pc      10.0.0.2
pakhs   qwerty  pakhs-pc       10.0.0.3
pakhs   qwerty  pakhs-laptop    10.0.0.4
```

Το αρχείο public.key και private.key ενός BN

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1/lOrpfVFXgKTk8QWjYj
VFe0hSRLCvZhM2fULj1sRq0Z89gaahJLO1q/xQIWFuNvDv++9d6apqYhhBJ68WPS
KghXWkc64yy9X62A0Yx9ycNW3k7Vm0GEf9q7lHVYmyrJG1zDPaObbW0IsxXTenNc
dgQP5CCN8hcOMXSNZMR8/0Oce10HX+pjjqItuK5qJ3gLcySrHal5fkY3sAUvukZ/
vGR3FDzpYRfse0RF6OuoWoEL/nL60WNm/ll+j5Mwsw3LWy00rQID31cJ9qaclLgO
sl0+hhnJBikOcjx8bi31jH8biTTt6oOcOZDtBqQ2TlbH6gI4+Hts2IR/0ph7jbzM
LQIDAQAB
-----END PUBLIC KEY-----
```

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA1/lOrpfVFXgKTk
ahJLO1q/xQIWFuNvDv++9d6apqYhh
m0GEf9q7lHVYmyrJG1zDPaObbW0Is
X+pjjqItuK5qJ3gLcySrHal5fkY3sAUvu
0WNm/ll+j5Mwsw3LWy00rQID31cJ9
6oOcOZDtBqQ2TlbH6gI4+Hts2IR/0ph
2ZQDdHEVAGuaT1Spw31+P1lJk41Xv5xbPYZ6zwCQ093tHWJRMDDp+EpWArlBT+ke
wGtzoAedyR42iJUAfEUAYdJH+y1/mA/kxITS1pL5olhLi3BCdMGdVr1//+F/d0GG
Ieq+vMKtba3kVG7b/UYmtR04jqbPbSIn7k9RZtpO/d2vGH41lXg1UyUgtEQgIpOW
WDaG074rL6KuxkM97e5D/e4oPptZBDvi1uQcFT1yIFDkvs6gjJE86DWZH7uIMejf
1pABUsPwzOPeJ5gOaw+qhSBUtPuRzB4VILfX9vvdwNxFSB14RI72DB23kaFyVioK
zTm3iGkCgYEA/Ebj50+zYpTvJWgcjlVMEeu9AOqEHqH7QaM61lyyXlK5zSEfea6L
3vsWisUTWM0oBiUeKJ3n9im3WJ+kgOrOFPrZX503opul4D7xuKVzBnDHbg2Zx+AB
uWyrnUGkW6e4bhKEtdvst8v7PuUWkVjp1xyjvO4Tk0Ua/CqDX2mHA08CgYEA2ylD
V/BcwqGugcthNSGXVQElqtTeaiVvTbby8Zk0jHcrepJG76yQqSW+TMNFtG8Pjj9J
PcvmdbMAdZTqPgiNV2MO88QYHcT4h9Upa/giAsb1BAsnZqlCj6YRHwwcXIWEAuFp
7xsWYeF0jKJB1NfsQD5f1vFyr1kYPPTgZBrgicMCgYEA0IpNVOYGdKSG99YTXPlX
s6y6hWpXIvdlusGTHqZr8BrUaqRJ342RJBdNcBMvRgX5YvMF9i9qE4wyerklBEiV
aLRgQnC1D984hKGjsa5a4mUSBoCJsbcT1dLmHk2n7vg7Ngpq1+ZfzSN6omg/epEU
ZHTRSZlIZ0IF55PBG3shV3MCgYBSwmY30wB0TvHC+bYfhivLYb+DnxbOJoy9YBSl
vyDk2iuFAa/f2d5WwXX3LtYnqLjLEoLp3xGL6KiHvlAmVLxq/3EqBCbHNxZS1N/r
cawGOHNVr5CVZ91GuYNFoiEjnxeWruB99lChba3BXZRWd6MzL1qppEuWg6Jvglkp
9CxOWQKBgQCZ5ZqBmOkLqlL9x9COZ9IL6djMIRmS4HXmCq1tSjzXjOAOg6Y/pw7w
Nz/UWHi/f4RECnVsC2+I5D3875onOAMVjsFwii4oGs/F3F/4EJ58ijs1EDw/ZO3J
wuvKuhNu0kqfc7GT9EgOgQbdK5P/CCYf8W+W+rciz15t/zPJapYorw==
-----END RSA PRIVATE KEY-----
```

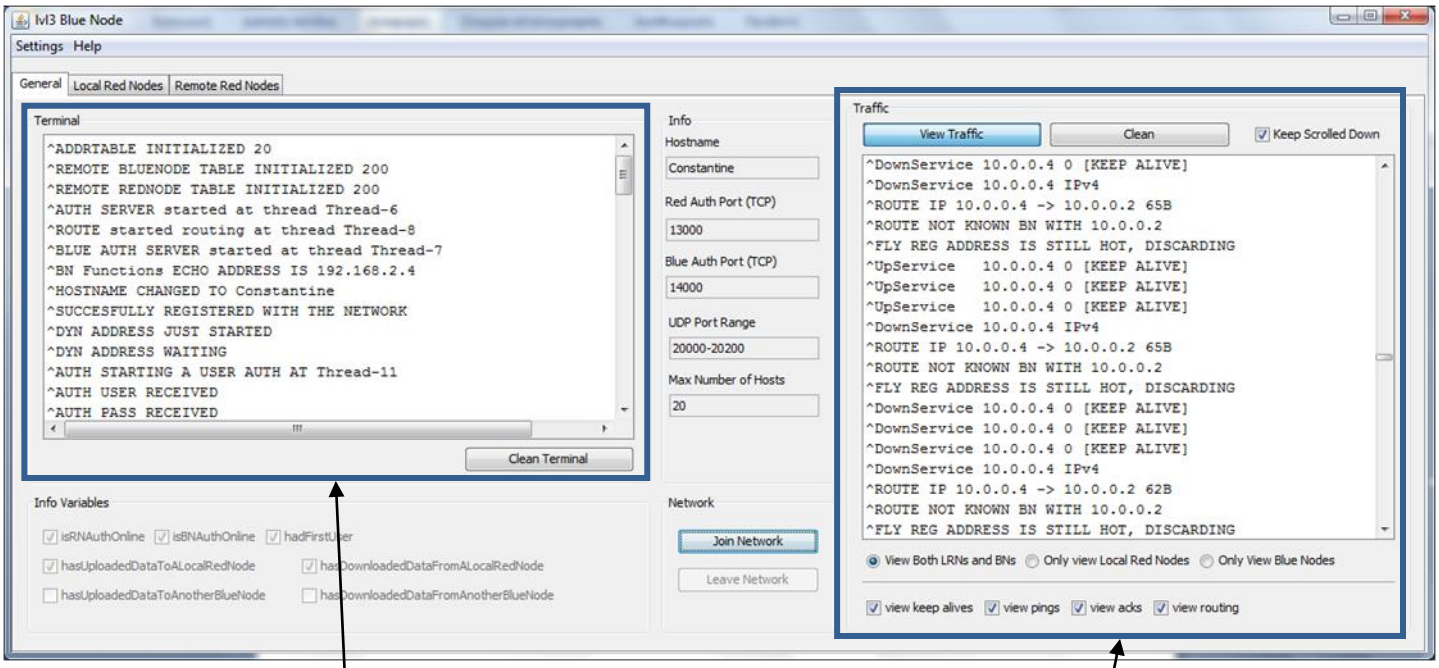
2.2.3 Χρήση και λειτουργίες ενός Blue Node



Εικόνα 23 BlueNode main window

Αυτό είναι το βασικό παράθυρο του blue node. Χωρίζεται σε τρείς καρτέλες οι οποίες έχουν κατηγοριοποιημένα τις πληροφορίες που θα ήθελε να δει ένας διαχειριστής BN. Η πρώτη καρτέλα παραθέτει λειτουργικές πληροφορίες της εφαρμογής, η δεύτερη παραθέτει στοιχεία για τους τοπικά συνδεμένους κόκκινους κόμβους και η $3^{η}$ καρτέλα παραθέτει όλα τα απαραίτητα στοιχεία επικοινωνίας με τους απομακρυσμένους κόκκινους κόμβους.

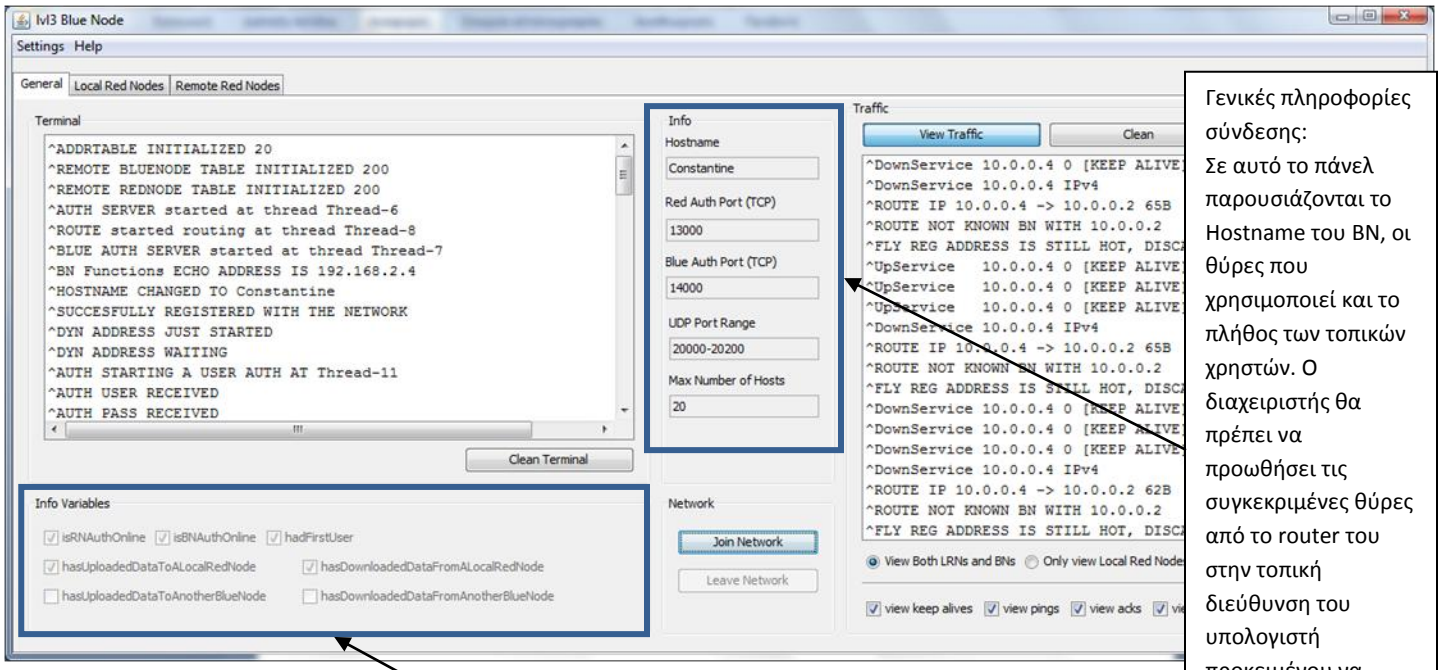
1η Καρτέλα Γενικές Πληροφορίες



Παράθυρο Ροής Πληροφοριών:
Σε αυτό το παράθυρο παραθέτονται πληροφορίες που παράγονται κατά την εκτέλεση του προγράμματος. Αν υπάρξει κάποιο πρόβλημα ο διαχειριστής μπορεί να αντιγράψει όλες τις πληροφορίες από το σημείο που θέλει σε ένα αρχείο κειμένου για περεταίρω ανάλυση. Με το κουμπί Clean Terminal καθαρίζεται το παράθυρο.

Παράθυρο Ροής Πακέτων:
Σε αυτό το παράθυρο παραθέτονται πληροφορίες για όλα τα πακέτα που δέχεται, δρομολογεί και στέλνει ο μπλε κόμβος. Ο διαχειριστής μπορεί να τα βλέπει ή όχι πατώντας το κουμπί view traffic ενώ με το κουμπί clean καθαρίζεται το παράθυρο.

Εικόνα 24 BlueNode console & traffic information



Γενικές πληροφορίες σύνδεσης:
Σε αυτό το πάνελ παρουσιάζονται το Hostname του BN, οι θύρες που χρησιμοποιεί και το πλήθος των τοπικών χρηστών. Ο διαχειριστής θα πρέπει να προωθήσει τις συγκεκριμένες θύρες από το router του στην τοπική διεύθυνση του υπολογιστή προκειμένου να λειτουργήσει η εφαρμογή και στο διαδίκτυο.

Μεταβλητές Πληροφοριών:
Οι συγκεκριμένες μεταβλητές μας ενημερώνουν για γεγονότα όπου έχουν συμβεί κατά τη διάρκεια εκτέλεσης της εφαρμογής. Όταν κάτι είναι επιλεγμένο σημαίνει ότι έχει συμβεί. Οι μεταβλητές είναι:

- isRNAuthOnline ~ ο μπλε κόμβος έχει ανοίξει τις απαραίτητες θύρες που χρειάζεται από το ΛΣ και περιμένει αιτήσεις από RNs
- isBNAuthOnline ~ ο μπλε κόμβος έχει ανοίξει τις απαραίτητες θύρες που χρειάζεται από το ΛΣ και περιμένει αιτήσεις από BNs
- hadFirstUser ~ ό πρώτος χρήστης συνδέθηκε στο σύστημα
- hasUploadedDataToALocalRedNode ~ έχει μεταφορτώσει δεδομένα σε ένα τοπικό κόκκινο κόμβο.
- hasDownloadedDataFromALocalRedNode ~ έχει δεχτεί δεδομένα από ένα τοπικό κόκκινο κόμβο.
- hasUploadedDataToAnotherBlueNode ~ έχει μεταφορτώσει δεδομένα σε ένα μπλε κόμβο.
- hasDownloadedDataFromAnotherBlueNode ~ έχει δεχτεί δεδομένα από ένα μπλε κόμβο.

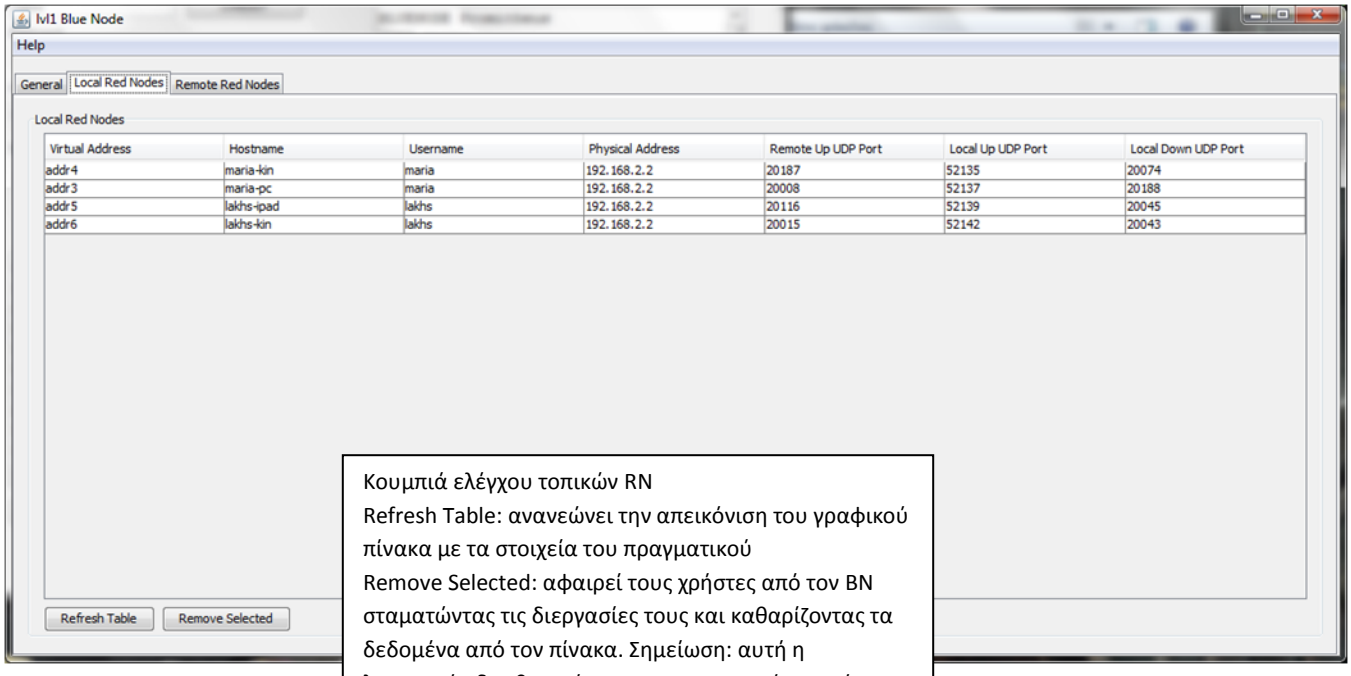
Εικόνα 25 BlueNode info variables ports and hostname

2η Καρτέλα Τοπικοί Κόκκινοι Κόμβοι

Σε αυτήν την καρτέλα αναγράφονται βασικές πληροφορίες για τους συνδεμένους κόκκινους κόμβους. Όταν κάποιος κόμβος εισέρχεται ή αποχωρεί ο πίνακας ανανεώνεται αυτόματα ενώ υπάρχουν και κουμπιά όπου διευκολύνουν τον χειρισμό. Ο διαχειριστής μπορεί να επιλέξει πολλά κελιά κρατώντας πατημένο το Ctrl και κάνοντας κλικ στις εγγραφές που τον ενδιαφέρουν για κάποια ενέργεια από τα κουμπιά. Επίσης μπορεί να κάνει διπλό κλικ σε ένα κελί εγγραφής για να πάρει τα στοιχεία του.

Ο κάθε κόκκινος κόμβος έχει τα εξής στοιχεία:

- εικονική διεύθυνση (virtual address) ~ το κύριο γνώρισμα ενός κόκκινου κόμβου η εικονική διεύθυνση είναι μοναδική για κάθε μηχάνημα.
- Hostname ~ κάθε hostname είναι μοναδικό και αντιστοιχίζεται σε μία μόνο εικονική διεύθυνση
- Username ~ σε ποιόν χρήστη ανήκει η συγκεκριμένη εικονική διεύθυνση
- Physical Address ~ η διεύθυνση από την οποία είναι συνδεμένος
- Uplink port ~ η θύρα που χρησιμοποιεί ο BN για να στέλνει κίνηση
- Downlink port ~ η θύρα που χρησιμοποιεί ο BN για να δέχεται κίνηση

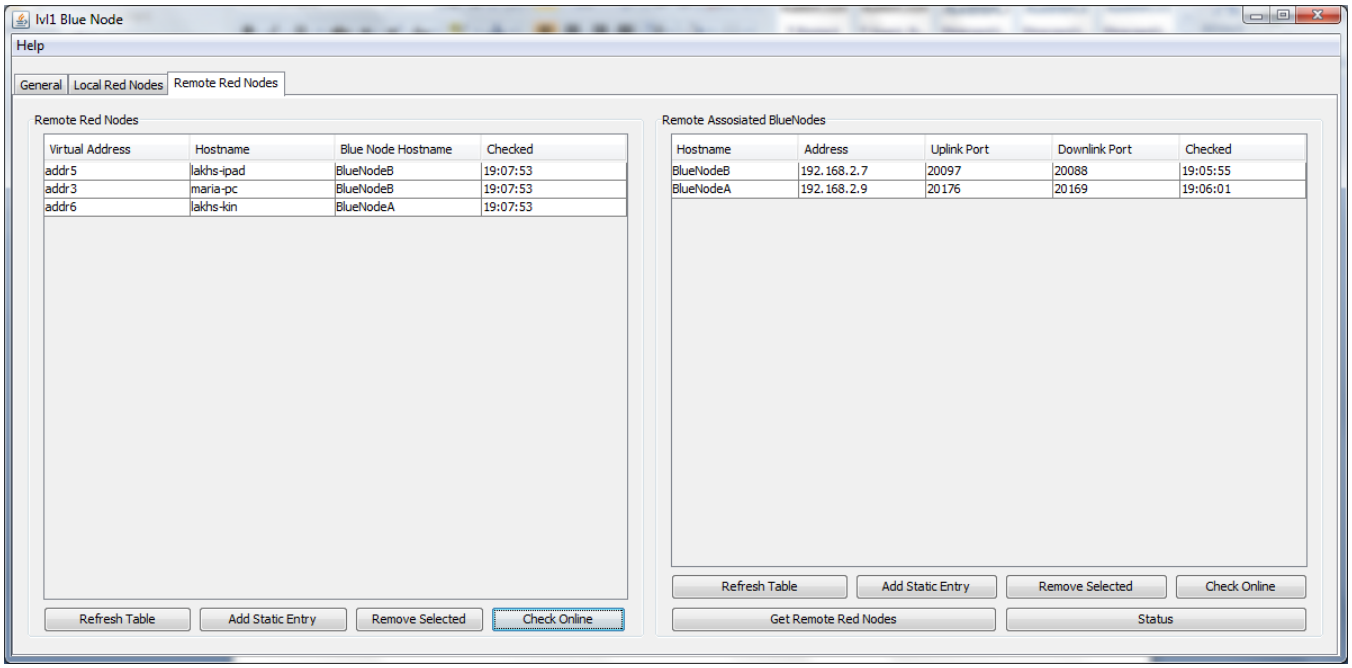


Κουμπιά ελέγχου τοπικών RN
Refresh Table: ανανεώνει την απεικόνιση του γραφικού πίνακα με τα στοιχεία του πραγματικού
Remove Selected: αφαιρεί τους χρήστες από τον BN σταματώντας τις διεργασίες τους και καθαρίζοντας τα δεδομένα από τον πίνακα. Σημείωση: αυτή η λειτουργία δεν θα πρέπει να χρησιμοποιείται από τον διαχειριστή για να πετάει κόμβους αλλά για να μπορεί να διαγράψει νεκρά δεδομένα που έχουν κολλήσει από το σύστημα.

Εικόνα 26 BlueNode Local Red Nodes

3η Καρτέλα Απομακρυσμένοι Κόκκινοι Κόμβοι

Σε αυτή την καρτέλα παραθέτονται πληροφορίες για τους απομακρυσμένους κόκκινους κόμβους. Να θυμηθούμε ότι οι απομακρυσμένοι RNs φιλοξενούνται από άλλους BNs και ότι ένας BN μπορεί να έχει πολλούς RN. Επομένως θα πρέπει να υπάρχουν 2 διαφορετικοί πίνακες ένας για τους Remote Red Nodes - RRNs και ένας για τους BNs όπου οι RNs φιλοξενούνται. Ο διαχειριστής μπορεί να κάνει και εδώ διπλό κλικ σε ένα κελί για τα στοιχεία του και Μπορεί να επιλέγει πολλές εγγραφές με το Ctrl και μια ενέργεια από τα κουμπιά για όλα τα επιλεγμένα στοιχεία.



Εικόνα 27 BlueNode Remote Red Nodes (RRDs)

Σε αυτή την καρτέλα βλέπουμε ότι ο BN μας γνωρίζει 4 RRNs εκ των οποίων οι 2 ανήκουν στον BlueNodeB και ο ένας στον BlueNodeA. Στον δεξιό πίνακα βλέπουμε τους γνωστούς BNs δλδ τον BlueNodeA και τον BlueNodeB .

BN Table

Πεδία:

- Hostname ~ Το όνομα του
- Address ~ Η φυσική διεύθυνση του BN (τα BNs δεν χρειάζονται virtual address)
- Uplink port ~ Η θύρα από την οποία ο BN πηγής στέλνει στο BN προορισμού
- Downlink port ~ Η θύρα από την οποία ο BN πηγής δέχεται από το BN προορισμού

Κουμπιά:

- Refresh table ~ ανανέωση πίνακα
- Add Static Entry ~ προθήκη στατικής εγγραφής BN (ανοίγει νέο μενού)
- Remove Selected ~ αφαιρεί τους επιλεγμένους BNs
- Check Online ~ ελέγχει τους επιλεγμένους BNs
- Get Remote Red Nodes ~ ζητάει τους RRNs που είναι καταχωρημένοι από τους επιλεγμένους BNs
- Exchange Red Nodes ~ ανταλλαγή RRNs μεταξύ των 2 κόμβων
- Status ~ ανοίγει ένα παράθυρο διαχείρισης του συγκεκριμένου BN

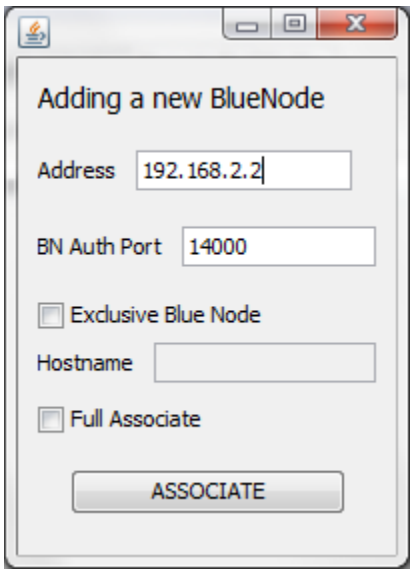
AddStaticEntry:

Πατώντας ο διαχειριστής το κουμπί ανοίγει ένα νέο παράθυρο. Σε αυτό το παράθυρο εισάγει τη διεύθυνση και τη θύρα του BN που θέλει να προσθέσει.

Υπάρχουν άλλες 2 επιπλέον επιλογές:

Μπορεί να επιλέξει exclusive BN και να συνδεθεί με ένα BN με συγκεκριμένο όνομα

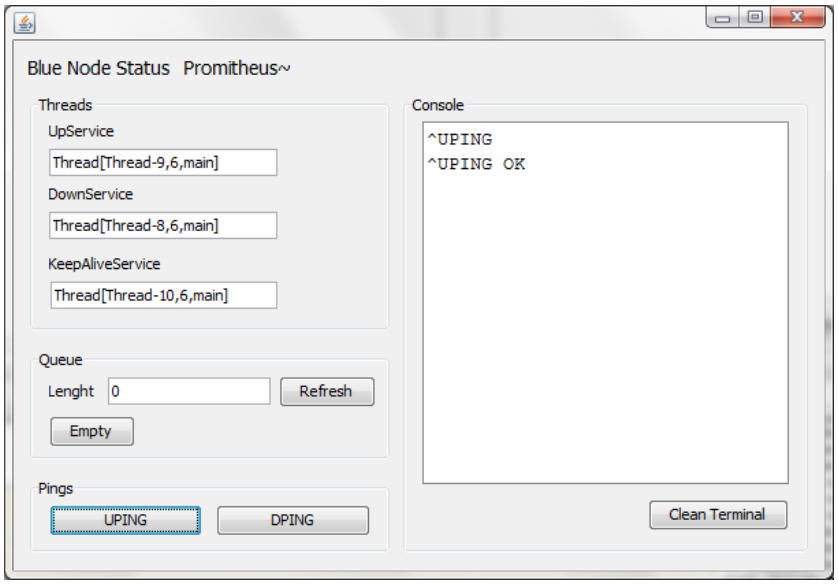
Full Associate: άμα επιθυμεί οι BN να ανταλλάξουν και RNs



Εικόνα 28 adding a BlueNode association manually

Status:

Στο status ανοίγει ένα μενού για τον επιλεγμένο BN όπου αναγράφει ποια threads έχει δεσμεύσει ο BN επίσης έχει επιλογή καθαρισμού δεδομένων προς αποστολή και 2 ping, ένα uping και ένα dping.



Εικόνα 29 A BlueNode's personal test window

RRN Table

Πεδία:

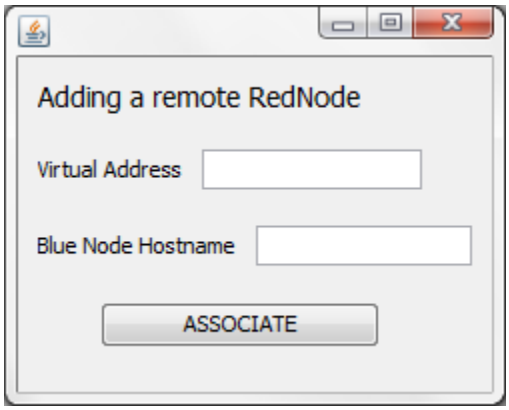
- εικονική διεύθυνση (virtual address) ~ το κύριο γνώρισμα ενός κόκκινου κόμβου η εικονική διεύθυνση είναι μοναδική για κάθε μηχάνημα.
- Hostname ~ κάθε hostname είναι μοναδικό και αντιστοιχίζεται σε μία μόνο εικονική διεύθυνση
- Blue Node Hostname ~ Ο BN ο οποίος φιλοξενεί τον συγκεκριμένο RN
- Checked ~ πότε ελέγχθηκε για τελευταία φορά η εγγραφή (θυμηθείτε στα local red nodes δεν είχαμε αυτό το πεδίο γιατί οι συνδέσεις εκεί ήταν συνδεσμοστρεφείς, εδώ έχει γίνει ανταλλαγή δεδομένων)

Κουμπιά:

- Refresh Table ~ ανανέωση της απεικόνισης του γραφικού table
- Add Static Entry ~ προσθήκη νέας στατικής εγγραφής. Ανοίγει ένα νέο παράθυρο που το βλέπουμε παρακάτω
- Remove Selected ~ αφαίρεση επιλεγμένων RRNs
- Check Online ~ έλεγχος ύπαρξης του RRN

Add Static Entry

Ο διαχειριστής μπορεί να τοποθετήσει τη virtual address του RRN και το όνομα του BN. Για να δουλέψει η διαδικασία θα πρέπει πρώτα να είναι καταχωρημένος ο BN που δηλώνουμε. Εάν είναι τότε η διαδικασία θα ψάξει τον συγκεκριμένο BN να δει άμα ο RRN είναι ενεργοποιημένος. Αν είναι τότε θα προστεθεί στον πίνακα.



Εικόνα 30 Add an RRD

2.3 Tracker και στήσιμο όλης της πλατφόρμας

Unity Registry

Προκειμένου να στηθεί όλη η πλατφόρμα για πλήρη χρήση θα πρέπει να υπάρχει ένας Tracker & το registry. Ο tracker & το registry είναι ένα μηχάνημα ο οποίος πέρα των άλλων θα πρέπει να διαθέτει:

ένα mysqld

ένα httpd με υποστήριξη php και mysql

Στα Linux είναι αρκετά εύκολο για ένα εξοικειωμένο χρήστη να έχουμε τα παραπάνω και σε όλες τις debian εφαρμογές εκτελούμε:

```
su

apt-get install mysql-server mysql-client

apt-get install apache2

apt-get install php5 libapache2-mod-php5

/etc/init.d/apache2 restart
```

και κάνουμε και install το PhpMyAdmin, θυμόμαστε το password της mysql

Στα windows αρκεί να βάλουμε XAMPP

Στη συνέχεια πάμε είτε από το PhpMyAdmin είτε από το mysql client της γραμμής εντολών δημιουργούμε την "unity_db"

και βάζουμε τα στοιχεία του unity_db.sql με import

Βάζουμε στα hdocs ή στο /var/www/ το direcory του register page (το unityreg)

Ανοίγουμε το αρχείο με το database.php κάτω από το dir και δηλώνουμε τα στοιχεία της βάσης δεδομένων.

```
<?php
$db_host="localhost";
$db_user="root";
$db_password="";
$db_name="unity_db";
?>
```

Ανοίγουμε τη θύρα 80 από firewalls και router.

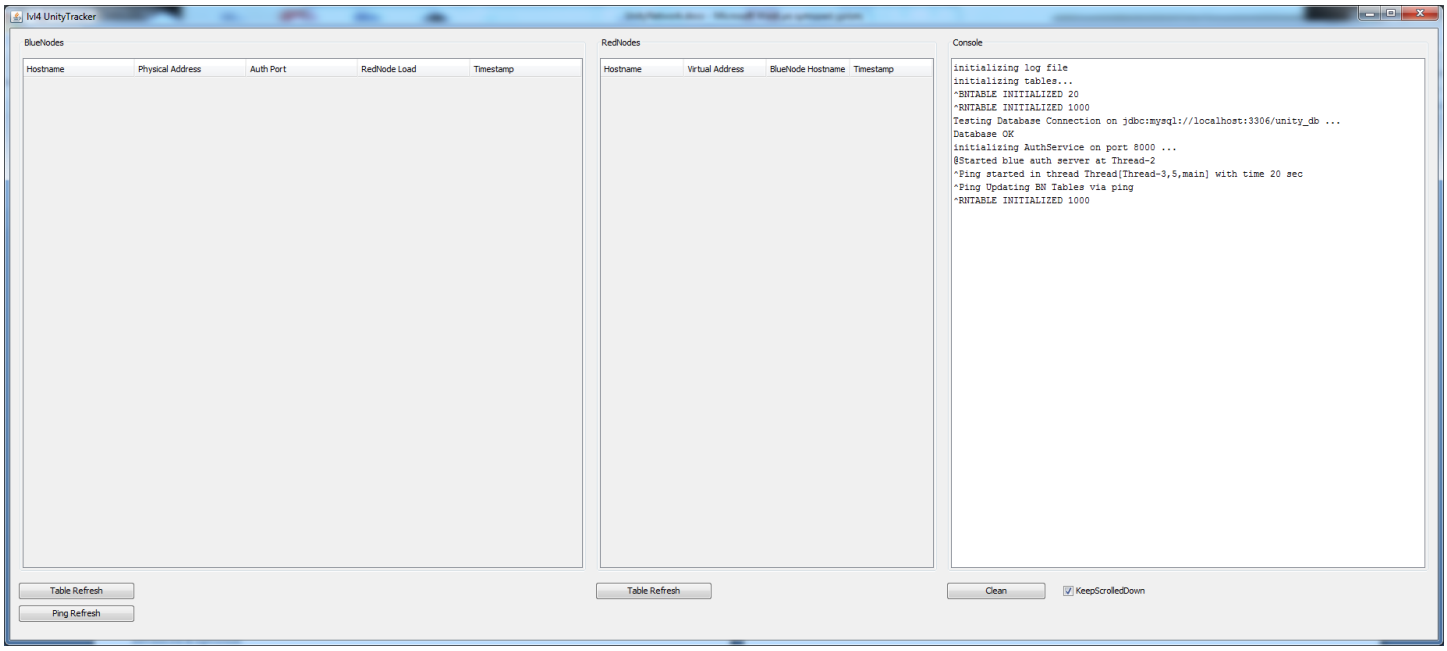
Unity Tracker & Registry

Βάζουμε τον Tracker σε ένα Dir και ανοίγουμε το tracker.conf Εδώ πέρα σημαντικότερο είναι να δηλώσουμε την βάση δεδομένων

```
##############################################
#        Unity Tracker Config File                              #
##############################################

#please do not comment any variable nor remove any. this will result in error
#instead only change the value to an appropriate input as described
#WARNING DO NOT MAKE SPACES AFTER A NUMBER

#first of all what shall be the name of the netwrok?
#and the authport! default 8000

NetworkName = UnityNetwork
AuthPort = 8000

#database settings
#the url should be in this type of form
# jdbc:mysql://IPaddress:port/database
# jdbc:mysql://192.168.3.6:3306/unity_db

DatabaseUrl = jdbc:mysql://192.168.3.13:3306/unity_db
DatabaseUser = unitytracker
DatabsePassword = 12345

#enable GUI
enableGUI = true

#network capacity
RedNodeCapacity = 1000
BlueNodeCapacity = 20

#logging... in tracker.log
#true ~ false
log = true

#ping time in sec
#ping is the function where the tracker searches for all active BNs ion order to find
#if someone does not respond
#because a BN is ought to be connectionless with tracker

ping = 20
```

Στη συνέχεια πρέπει να προωθήσουμε την θύρα (AuthPort) από firewall και router. Ο tracker είναι έτοιμος να λειτουργήσει και τον εκτελούμε

Παρακάτω βλέπουμε το παράθυρο του tracer αν εκτελεστεί με GUI



Εικόνα 31 Tracker main window

Αν όλα είναι σωστά σχετικά με τη βάση δεδομένων θα πρέπει να δούμε Database OK!

Και πλέον μπορούμε να εγγράψουμε BNs στην βάση δεδομένων ως γνωστούς. Οι BNs με τη σειρά τους θα πρέπει να γνωρίζουν τη διεύθυνση του tracker και να είναι πιστοποιημένοι με το δίκτυο.

Όλοι οι χρήστες συνδέονται στην σελίδα του unityreg η οποία πρέπει να δουλεύει για να γράφονται χρήστες.

advanced & optional

Μπορούμε να έχουμε τον Tracker και το Registry σε διαφορετικά μηχανήματα αν το επιθυμούμε.

Επίσης θα μπορούσε να υπάρχει και ένας εσωτερικός DNS στο δίκτυο όπως ο bind9

Σε αυτή την περίπτωση έχουμε ένα DNS όπου συνδέεται με RedNode μέσα στο εικονικό δίκτυο και διαχειρίζεται τα hostnames, η 10.0.0.2 ip addr έχει κρατηθει για υποστήριξη DNS.

Αυτά είναι τα τρία είδη κόμβων του Unity, στο κεφάλαιο 5 παρουσιάζονται 2 σενάρια πραγματικής χρήσης του δικτύου το ένα σε LAN και το άλλο στο internet ώστε να παρουσιαστεί καλύτερα στην πράξη. Επίσης αν κάποιος επιθυμεί να ελέγξει το δίκτυο ένας BN αρκεί με μία λίστα χρηστών.

Επίσης ο Tracker για λόγους διευκόλυνσης έχει υλοποιηθεί και σε .vdi virtual disk image μέσα από τα LUBUNTU!

3. Αρχικές έννοιες VPN & Ανάλυση γνωστών υλοποιήσεων

3.1 Αρχικές έννοιες

Η σκοπιά ενός δικτύου, μίας διεύθυνσης και γενικότερα ενός VPN

"The first point indicates that an IP address only has meaning within the VPN in which it exists. For this reason, it is necessary to identify the VPN in which a particular IP address has meaning, the "scope" of the IP address." ~ RFC 2685

Πιο συγκεκριμένα η "σκοπία" ενός VPN ή VN γενικότερα σημαίνει από ποιά οπτική γωνία το μελετάμε. Υπάρχουν δύο σκοπίες η εξωτερική και η εσωτερική. Η εξωτερική σκοπιά σημαίνει οτι μελετάμε το δίκτυο απ' έξω δηλαδή από την φυσική του πλευρά (δηλαδή πως έχει στηθεί) ενώ εσωτερική σκοπιά σημαίνει οτι το μελετάμε από μέσα σαν να ήμασταν μέλη του δικτύου. Το ίδιο συμβαίνει και με τις IP διευθύνσεις, κάθε μηχάνημα όπου έχει συνδεθεί έχει και από μια εικονική διεύθυνση όπου έχει απήχηση μόνο στο εικονικό δίκτυο (εσωτερική διεύθυνση) και έχει και την διεύθυνση όπου του έχει αποδοθεί από το LAN (εξωτερική).

Πότε ξεκίνησαν τα VPN και ποιοί ήταν οι αρχικοί τους στόχοι

Τα εικονικά δίκτυα ξεκίνησαν τη δεκαετία του 2000 με την εμφάνιση του πρωτοκόλλου PPTP. Στόχος της εποχής τότε ήταν κυρίως η διευκόλυνση από τις εταιρείες και τις δημόσιες υπηρεσίες στα άτομα τα οποία ήταν φυσικά μακριά και ήθελαν πρόσβαση σε ένα πόρο της επιχείρησης απομακρυσμένα όπως για παράδειγμα: ένα δικτυακό εκτυπωτή, ένα file server, ή μια εσωτερική ιστοσελίδα. Σε αυτή την περίπτωση μπορούσαν να συνδεθούν με ένα πρωτόκολλο VPN και να παρουσιαστούν στο LAN ως τοπικοί hosts. Στη συνέχεια έκαναν access στον πόρο όπου επιθυμούσαν (σαν να είχαν φυσική παρουσία στο LAN) και στο τέλος αποσυνδεόταν.

Τι ρόλο έχουν τα VPN σήμερα

Έως και σήμερα τα VPN χρησιμοποιούνται με την παραπάνω λογική αλλά έχουν εξελιχθεί και σε διαφορετικούς τομείς. Στη δεκαετία όπου διανύουμε οι έννοιες όπως virtualization, clouding, αντικειμενοστρέφεια και modularity είναι πολύ δημοφιλείς και αυτό γιατί υπάρχει μια μεγάλη ανάγκη να μπορούμε να εξομοιώσουμε φυσικά αντικείμενα στο μέχρι τώρα χώρο της πληροφορικής (δηλαδή αντικείμενα όπου καταλαμβάνουν φυσικό χώρο όπως μηχανήματα server) σε λογικά αντικείμενα όπου "ζούν" δλδ έχουν state (satefull) μέσα σε άλλους υπολογιστές. Αυτό αφενός επειδή το hardware κοστίζει και αφετέρου για καλύτερη κατανομή πόρων με περισσότερο λογικό έλεγχο παρά φυσικό. Επίσης μας απασχολούν τα δικαιώματα όπου έχει το εξωτερικό περιβάλλον σε σχέση με το αντικείμενο όπου είναι στην κατοχή του (δηλαδή εάν είναι σε θέση να του αλλάξει την κατάσταση του ή όχι ή να έχει πρόσβαση στα περιεχόμενα του) όπου εκφράζονται μέσα από την αντικειμενοστρέφεια ως έννοια.

Τα VPN ουσιαστικά περνάνε αυτή τη φάση και έρχονται στο μεταίχμιο της αλλαγής τους ώστε να εξαρτώνται όλο και λιγότερο από το HW, να γίνουν πιο λογικά και πιο κατανεμημένα και αυτός είναι και ένας από τους στόχους της συγκεκριμένης πτυχιακής, πώς ένα VPN θα μπορεί να ανεξαρτητοποιηθεί από HW και θα γίνει πιο μεγάλο και κατανεμημένο. Δηλαδή πως ένα VPN μπορεί να γίνει VN ευρείας κλίμακας!

VPNs Vs Clouding

Όπως μπορούμε να διαπιστώσουμε τα VPN και τα Clouds ανήκουν στην ίδια εξελικτική ομάδα (το virtualization). Σε σχέση με τα VPN τα cloud είναι πολύ πιο δημοφιλή σε σημείο trend και παρουσιάζονται ως το "μέλλον" σε αντίθεση με τα VPN όπου πλέον δεν χρηματοδοτούνται. Εγώ αρχικά θα παραθέσω τις ομοιότητες και μετά πρέπει να αναρωτηθούμε αν είναι όντως έτσι τα πράγματα.

Οι ομοιότητα VPN και Cloud είναι η κοινή ανάγκη παροχής υπηρεσιών

- Ένα Cloud συνήθως παρέχει ένα πρόγραμμα πελάτη όπου συνδέει ένα χρήστη και στη συνέχεια του παρέχει υπηρεσίες. Το πρόγραμμα είναι αρκετά πιο εύκολο εφόσον ή "διαφάνεια" όπου βιώνει ένας χρήστης (δηλαδή το πόσο εύκολο και γρήγορο είναι να συνδεθεί) είναι πολύ αυξημένη.
- Ένα VPN κάνει το ίδιο πράγμα μόνο που στα VPN ο χρήστης πρέπει πρώτα να συνδεθεί στο δίκτυο και μετά για να έχει πρόσβαση σε μια υπηρεσία πρέπει να ανοίξει την εφαρμογή της. Η υπηρεσία δε είναι IP.

Ουσιαστικά εάν δεν το έχουμε καταλάβει ακόμα τα Clouds έχουν κατασκευαστεί με ιδιαίτερη έμφαση στην κλειστή αρχιτεκτονική. Ο χρήστης έρχεται σε επαφή **μόνο με το τελικό μηχάνημα** παροχής υπηρεσίας και συνήθως δεν μπορεί να δεί από πίσω από το τελικό μηχάνημα τι υπάρχει. Άρα διακρίνουμε μια **κλειστή αρχιτεκτονική** όπου ευνοεί αρκετά την τάση για απόκρυψη μηχανισμών και πληροφοριών όπου συντελεί στην δημιουργία ανθρώπων δέσμιων στην τεχνολογία (δηλαδή όπου εκτελούν αυτοματοποιημένες ρουτίνες χωρίς να έχουν τον έλεγχο του τι υπάρχει από πίσω). Σε αντίθεση τα VPN επιτρέπουν σε ένα χρήστη να γίνει **μέλος** του δικτύου και να δει τα εσωτερικά μηχανήματα μιας πλατφόρμας. Στη συνέχεια μπορεί να συνδεθεί με ένα πρόγραμμα συγκεκριμένης υπηρεσίας σε κάποιο κόμβο. Επομένως διαπιστώνουμε οτι τα VPN είναι πολύ πιο ανοικτά και πολύ πιο ελεύθερα από τα clouds γι' αυτό έχει αρκετή σημασία να τα εξελίξουμε σε σημείο όπου να είναι πιο σύγχρονα και πιο κατανεμημένα σε σχέση με τώρα.

3.2 Μελέτη ήδη υπαρχόντων πρωτοκόλλων κίνησης VPN

Σε αυτό το κεφάλαιο θα μελετήσουμε συνοπτικά διάσημα πρωτόκολλα VPN τα οποία έχουν προτυποποιηθεί από διεθνείς οργανισμούς και εταιρείες για το πως λειτουργούνε και στη συνέχεια θα δούμε το OpenVPN όπου βασίζεται σε ανοιχτό κώδικα C και είναι μη προτυποποιημένο. Στη συνέχεια θα δούμε τις διαφορές μεταξύ τους και θα οδηγηθούμε σε συμπεράσματα.

PPTP

Το πρωτόκολλο PPTP υλοποιήθηκε από μία ένωση εταιρειών στην οποία συμμετείχε η Microsoft, η Ascend Communications, η 3Com και άλλες. Η περιγραφή του δημοσιεύτηκε στα αρχεία RFC τον Ιούνιο του 1999 ως RFC 2637. Το πρωτόκολλο περιγράφει μια μέθοδο προκειμένου να δημιουργηθεί ένα τούνελ από όπου θα περάσει κίνηση δικτύου (PPP) πάνω από δίκτυα IP δημιουργώντας ένα δίκτυο VPN μεταξύ των δύο άκρων. Το πρωτόκολλο είναι συνδεσμοστραφές μεταξύ των δύο άκρων και περιγράφει τις απαραίτητες διαδικασίες για την δημιουργία ενός τούνελ όπως πιστοποίηση πελάτη, την ανταλλαγή κίνησης και την αποσύνδεση του.

Σύνδεση

Πιο συγκεκριμένα χρησιμοποιεί ένα κανάλι ελέγχου πάνω από TCP όπου μέσω αυτού ελέγχεται η εγκατάσταση του τούνελ, την αποσύνδεση και αποδέσμευσή του και την παρακολούθηση και διατήρηση της συνεδρίας.

Για την μεταφορά κίνησης χρησιμοποιεί μια τροποποιημένη έκδοση του πρωτόκολλου GRE και μέσα στο κανάλι του μεταφέρονται PPP δεδομενογράμματα. Το πρωτόκολλο GRE περιέχει εκτός των άλλων χαρακτηριστικών του έλεγχο ροής δεδομένων και αλγόριθμους για αντιμετώπιση συμφόρησης.

Προκειμένου να περαστεί κίνηση GRE οι δρομολογητές και τα NAT στα άκρα θα πρέπει να επιλέγουν να μην εμποδίζουν την κίνηση του πρωτοκόλλου GRE το οποίο είναι transmission protocol.

Ασφάλεια και κρυπτογράφηση

Για την αρχική πιστοποίηση χρησιμοποιεί τα υποπρωτόκολλα:

PAP, CHAP, MS-CHAPv2

Ένα άλλο του χαρακτηριστικό είναι οτι τα GRE πακέτα είναι μη κρυπτογραφημένα και οποιοσδήποτε man in the middle μπορεί να συλλέξει κίνηση ή να τροποποιήσει πακέτα και να τα ξαναστείλει.

L2TP/IPsec

Το συγκεκριμένο πρωτόκολλο είναι συνδυασμός των υποπρωτοκόλλων L2TP και IPsec τα οποία δουλεύουν αυτόνομα και συνεργάζονται μαζί. Γι' αυτό θα πρέπει να τα μελετήσουμε και ξεχωριστά. Αρχικά θα δούμε το L2TP το οποίο προϋπήρχε του IPsec στη συνέχεια το πρωτόκολλο IPsec μόνο του και τέλος τον συνδυασμό τους μαζί σε ένα κοινό τρόπο συνεργασίας και δημιουργίας τούνελ.

L2TP

Το πρωτόκολλο L2TP δημιουργήθηκε ως εξελιγμένη έκδοση του PPTP και του L2F (του layer 2 Forwrding Protocol της Cisco). Η λειτουργία του έχει δημοσιευτεί ως RFC 2661. Πιο συγγεκριμένα η λειτουργία του είναι κοινή με τους προκάτοχους του και αυτή είναι να προωθεί πακέτα PPP πάνω από ένα δίκτυο IP.

Σύνδεση

Το πρωτόκολλο έχει ανάγκη προκειμένου να μεταφέρει τα πακέτα του να τα ενθυλακώσει σε πακέτα μεταφοράς τύου UDP ή σε άλλα πρωτόκολλα όπως ATM.

Μεταφέρει δύο ειδών μηνύματα: μηνύματα ελέγχου του καναλιού ή ενθυλακωμένα δεδομενογράμματα PPP.

Ασφάλεια και κρυπτογράφηση

Σε αυτή την κατηγορία από μόνο του παρουσιάζει πολλά κοινά χαρακτηριστικά με το PPTP τα υποπρωτόκολλα πιστοποίησης είναι κοινά (CHAP, CHAPv2)

Το L2TP από μόνο του δεν κρυπτογραφεί την κίνηση και ισχύουν οι ίδιοι κίνδυνοι που προαναφέραμε.

IPsec

Το IPsec είναι ένα πρωτόκολλο το οποίο παρουσιάζει μορφές κρυπτογράφησης και πιστοποίησης δεδομενογραμμάτων IP (εμπιστευτικότητα, επωνυμία) με αποτέλεσμα τη γενικότερη ασφάλεια της κίνησης μεταξύ δύο κόμβων σε δίκτυο IP, χωρίς την ανάγκη χρήσης αξιόπιστης υπηρεσίας στο transmission layer (π.χ. όπως SSL, HTTPS, SSH).

Τα άκρα της σύνδεσης μπορούνε να έχουν συμφωνήσει σε ένα κοινό κωδικό - συμμετρική κρυπτογράφηση ή να έχει το κάθε άκρο ζευγάρι δημοσίου ιδιωτικού κλειδιού και να επιτευχθεί ασύμμετρη ή υβριδική κρυπτογράφηση / αποκρυπτογράφηση των πακέτων.

Από μόνο του δεν παρουσιάζει χαρακτηριστικά VPN παρά μόνο χαρακτηριστικά κρυπτογράφησης.

L2TP/IPsec

Ουσιαστικά τα δύο πρωτόκολλα αναλαμβάνουν να συνεργαστούν μαζί συνδιάζοντας τα δυνατά τους στοιχεία. Το L2TP αναλαμβάνει την εκγαθύδρηση του τούνελ, την αποσύνδεση, την πιστοποίηση και τον έλεγχο μετάδοσης ενώ το IPsec την μεταφορά κρυπτογραφημένης κίνησης.

Η αρχική κίνηση πιστοποίησης όπως και η τελική κίνηση αποσύνδεσης, βάση ανάλυσης δικτύου, βασίζεται στο πρωτόκολλο επιπέδου εφαρμογής ISAKMP το οποίο μεταφέρεται μέσω του πρωτόκολλου μεταφοράς UDP.

Η κίνηση εικονικού δικτύου όπου δημιουργείται είναι κρυπτογραφημένη πακέτα ESP (transmission layer) τα οποία είναι κρυπτογραφημένα βάση του IPsec.

OpenVPN

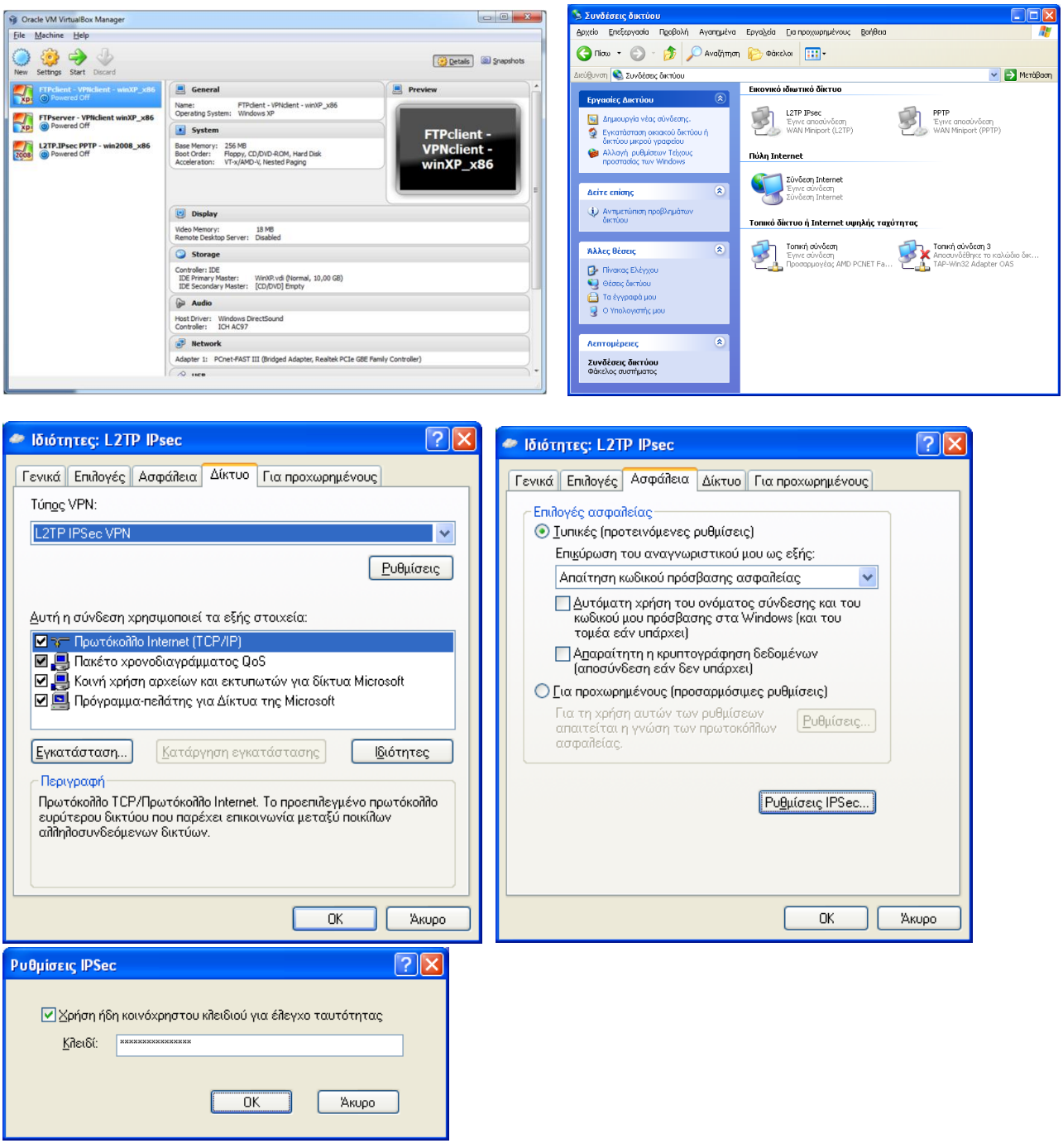
Το OpenVPN είναι ένα λογισμικό ανοιχτού κώδικα το οποίο υλοποιεί τεχνικές VPN προκειμένου να επιτευχθούνε ασφαλή τούνελ δικτύου. Το έχει γράψει ο James Yonan σε γλώσσα προγραμματισμού C και είναι κάτω από άδεια GPL. Μερικές από τις ιδιότητες του είναι: η χρήση SSL/TLS, έχει ιδιότητες NAT-traversing, παρουσιάζει χαρακτηριστικά αυτονομίας ως προς την εφαρμογή σε σχέση με το λειτουργικό και θεωρείται ο ελβετικός σουγιάς των vpn καθώς μπορεί να διεκπεραιώσει πάρα πολλές αρχιτεκτονικές και υλοποιήσεις VPN. Μερικές είναι:

Εικόνα 32 OpenVPN

- Σύνδεση πολλών LAN σε ένα κοινό
- Σύνδεση πολλών χρηστών σε ένα κοινό LAN με πόρους
- Δημιουργία κρυπτογραφημένων τούνελ για μετάδοση κίνησης δικτύου σε συνδυασμό με αποφυγή της αστυνόμευσης.
- Πολλαπλά VPN σε σειρά όπου μπορούν να προστατεύσουν την ταυτότητα ενός χρήστη διακινώντας τα πακέτα διάμεσο τους.
- Η δυνατότητα κάποιος χρήστης να παίρνει πραγματική διεύθυνση η οποία μπορεί να έχει διαφορετικά γεωγραφικά χαρακτηριστικά με αποτέλεσμα ξεκλείδωμα υπηρεσιών
- Διαίρεση επεξεργαστικού φόρου με πολλούς VPN server στο ίδιο LAN

3.3 VPN Protocol Reverse Engineering

Προκειμένου να παρατηρήσω την λειτουργία τους χρησιμοποίησα virtualBox VMs και εξομοίωσα τρία λειτουργικά συστήματα. Οι τρείς αυτοί κόμβοι είναι γεφυρωμένοι στην κάρτα δικτύου του υπολογιστή με αποτέλεσμα να μπορούν να επικοινωνούν μέσω τοπικού δικτύου. Πιο συγκεκριμένα τα δύο nodes με τη βοήθεια του 3ου συνδέονται σε ένα δικό τους προσωπικό εικονικό δίκτυο και ανταλλάσουν ένα αρχείο μέσω ftp. Τα υπόλοιπα μέλη του τοπικού δικτύου δεν έχουν γνώση για το τι ανταλλάσουν αυτοί οι δύο κόμβοι και επιπλέον οι δύο κόμβοι έχουν την δυνατότητα να διαθέσουν τις υπηρεσίες μόνο μέσα από το vpn χωρίς οι κόμβοι του LAN να έχουν γνώση οτι υπάρχει καν αυτή η υπηρεσία. Στη συνέχεια καταγράφηκε η κίνηση δικτύου μεσω wireshark και ελέγχοντας την με την χρήση του έχουμε τα παρακάτω συμπεράσματα.



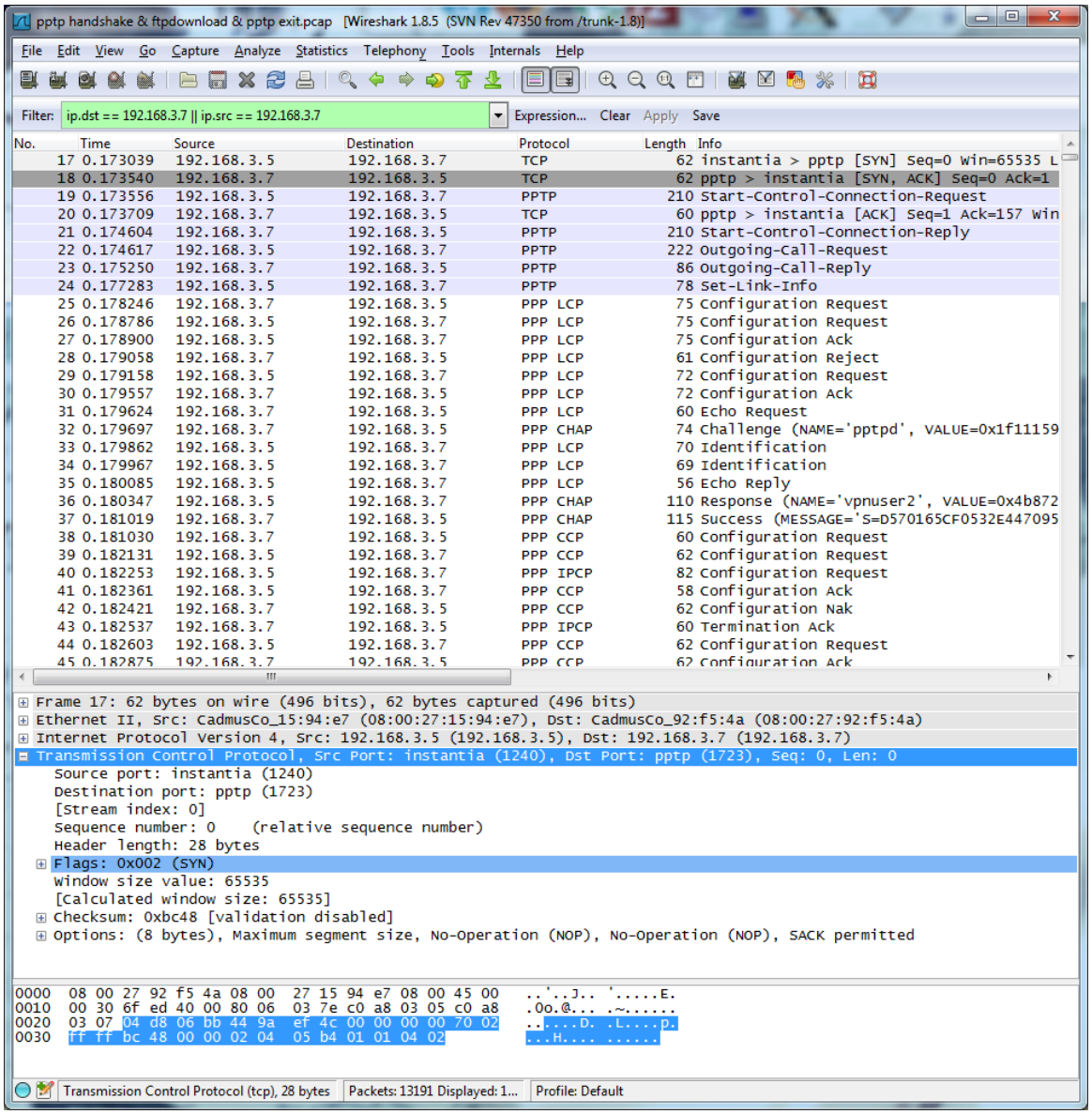
Εικόνα 33 Διάφορες ρυθμίσεις από τη δοκιμή πρωτοκόλλων VPN

PPTP

αρχικά στο αποθηκευμένο αρχείο καταγραφής δίνουμε το παρακάτω φίλτρο

ip.dst == 192.168.3.7 || ip.src == 192.168.3.7

και αυτό για να μας εμφανίσει κίνηση οποιοδήποτε πρωτοκόλλου σχετίζεται με τον VPN server του οποίου η διεύθυνση είναι η 192.168.3.7



Εικόνα 34 Wireshark PPTP

Όπως γνωρίζαμε ήδη θα δούμε πρώτα τη θύρα ελέγχου ή οποία λειτουργεί πάνω σε TCP εκτελεί την αρχικοποίηση της σύνδεσης και την πιστοποίηση του χρήστη.

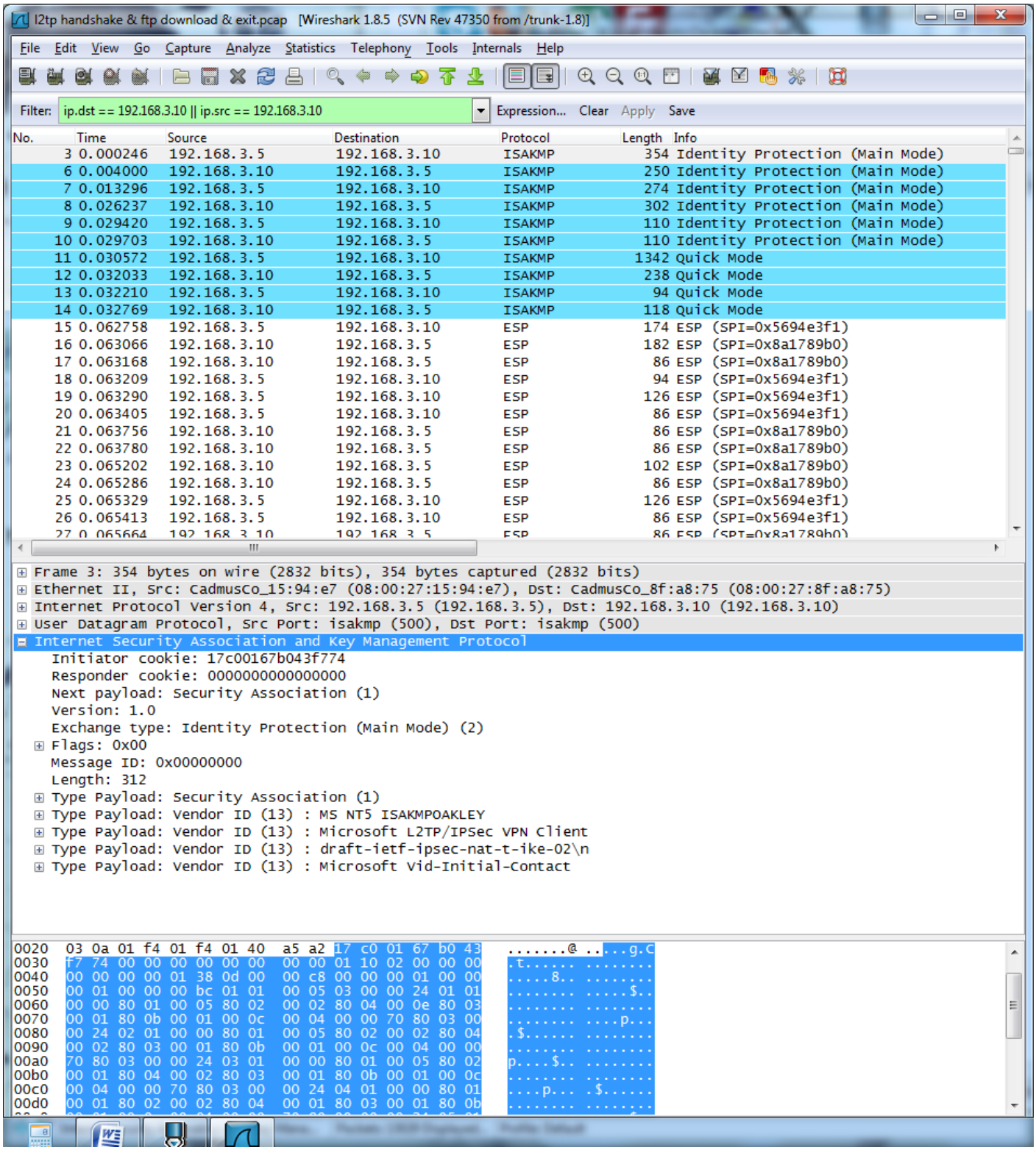
Στη συνέχεια το πρωτόκολλο PPP αναλαμβάνει να στείλει συμπιεσμένα datagramms από και πρός τον vpn server και τον client.

Η κίνηση είναι του αρχείου που κατεβάζαμε μέσω του FTP και η γενικότερη κίνηση μεταξύ των 2 host

Στο τέλος αναλαμβάνει πάλι το socket ελέγχου να κλείσει την συνεδρία καθώς αποσυνδέεται ο client.

L2TP/IPsec

Όμοια χρησιμοποιούμε φίλτρο για τον VPN server, αυτή τη φορά βρίσκεται στην διεύθυνση 192.168.3.10



Εικόνα 35 Wireshark L2TP

Αρχικά το πρωτόκολλο εφαρμογής ISAKMP πάνω από UDP αναλαμβάνει την εγκαθίδρυση και το στήσιμο του τούνελ καθώς και την πιστοποίηση του χρήστη.

Στη συνέχεια κρυπτογραφημένα (αυτή τη φορά και οχι συμπιεσμένα) πακέτα τύπου ESP (transmission layer) μεταφέρουν κρυπτογραφημένη κίνηση καθώς και μηνύματα ελέγχου ροής δεδομένων.

Τέλος μόλις μεταφέρουμε το συνηθισμένο αρχείο μέσω FTP αποσυνδέεται ο χρήστης και το πρωτόκολλο ISAKMP αναλαμβάνει τον τερματισμό του τούνελ.

Συμπεράσματα

Το PPTP και το L2TP/IPsec είναι και τα δύο πρωτόκολλα τούνελ! Αυτό σημαίνει οτι τα ίδια τα πρωτόκολλα δεν ενδιαφέρονται για κάτι παραπάνω από το τούνελ μετάδοσης και την διαδικασία δημιουργίας καταστροφής και διατήρησης του με έλεγχο ροής. Π.χ δεν ενδιαφέρονται για την διαδικασία εγγραφής του χρήστη στην υπηρεσία ή το πόση ώρα ένας χρήστης χρησιμοποιεί την υπηρεσία ή το άμα και πως θα πρέπει να συνεννοηθούν οι hosts μεταξύ τους.

Πιστοποίηση

Στην πιστοποίηση και τα δύο πρωτόκολλα χρησιμοποιούσαν όμοιες τεχνικές

Κρυπτογράφηση

Το PPTP δεν παρουσιάζει κάποιας μορφής κρυπτογράφησης παρά μόνο συμπίεση σε αντίθεση με το L2TP/IPsec όπου μεταφέρει κρυπτογραφημένη κίνηση.

Δρομολόγηση και ενθυλάκωση

Στο transmission το PPTP χρησιμοποιεί TCP για έλεγχο γραμμής ενώ GRE για μεταφορά της κίνησης. Το L2TP/IPsec χρησιμοποιεί UDP και ESP (IPsec over UDP). Αυτό μας προτρέχει να σκεφτούμε οτι οι ενδιάμεσοι δρομολογητές και τα NAT των άκρω θα πρέπει να επιτρέπουν την μετάδοση των τεσσάρων παραπάνω πρωτοκόλλων.

Το PPTP χρησιμοποιεί δύο γραμμές μια για έλεγχο μεταφοράς και άλλη μια για μετάδοση πακέτων σε αντίθεση με το L2TP/IPsec όπου χρησιμοποιεί μια γραμμή.

Απόδοση διευθύνσεων

Ένα αρκετά καλό χαρακτηριστικό των δύο πρωτοκόλλων είναι οτι δεν χρειάζονται απαραίτητα τη χρήση υπηρεσίας DHCP για απόδοση διευθύνσεων. Τα ίδια τα πρωτόκολλα θα φροντίσουν για την απονομή διευθύνσεων. Παρόλα αυτά εάν δεν χρησιμοποιηθεί DHCP οι διευθύνσεις μοιράζονται με τη σειρά εισόδου των χρηστών και δεν υπάρχει αντιστοίχηση μοναδικής διεύθυνσης - μηχανήματος χρήστη. Εαν συνδεθεί ο ίδιος χρήστης στην υπηρεσία ξανά πιθανόν να πάρει διαφορετική διεύθυνση δικτύου.

Αρχιτεκτονική

Και τα δύο είναι πολύ στενά συνδεδεμένα με το λειτουργικό σύστημα και η εγκατάσταση τους βασίζεται στην σωστή αρχικοποίηση του λειτουργικού συστήματος. Είναι μια διαδικασία που μπορεί να είναι αρκετά πολύπλοκη και μπορεί να εκτελεστεί μόνο από διαχειριστές. Η διαδικασία περιλαμβάνει εγκατάσταση, ρύθμιση των υπηρεσιών και του ίδιου του λειτουργικού για προώθηση κίνησης καθώς και ρύθμιση του firewall.

Επίσης η δρομολόγηση των πακέτων από τον ένα χρήστη σε ένα άλλο χρήστη της υπηρεσίας γίνεται εσωτερικά του υπολογιστή από το ένα άκρο μέχρι το άλλο άκρο του τούνελ. Αυτό σημαίνει ότι το ίδιο το OS θα πρέπει να επιλέξει να επιτρέπει την εσωτερική δρομολόγηση πακέτων. Ένα καλό παράδειγμα είναι οτι στον server του παραδείγματος ο ένας client μπορεί να έχει τούνελ με το ένα πρωτόκολλο και ο άλλος με το δεύτερο και να επικοινωνούν κανονικά.

Συγκεκριμένα για το L2TP είναι ανεξάρτητο πρωτόκολλο από το ipsec και σηκώνονται απο διαφορετικό service το καθένα. Η επικοινωνία τους γίνεται μέσω του OS όπου φέρει το ρόλο του μεσάζοντα.

Προτυποποίηση

Και τα δύο πρωτόκολλα είναι προσωποποιημένα με τέτοιο τρόπο ώστε να γνωστοποιούν στο LAN οτι πρόκειται να μεταφέρουν κίνηση vpn επομένως ο διαχειριστής του LAN έχει τον πρώτο και τελευταίο λόγο για το άμα θα τα επιτρέψει ή οχι. **Δηλαδή αστυνομεύονται πολύ εύκολα!** Χρειαζόμαστε ένα router το οποίο θα επιτρέπει διέλευση ESP GRE packets τα οποία δεν είναι TCP ή UDP όπου είναι πιο αναγνωρίσιμα και αποδεκτά.

Σε αντίθεση όπως θα δούμε παρακάτω στο Unity και στο OpenVPN **η κίνηση είναι μη προτυποποιημένη**! θα μπορούσε απλά ο χρήστης να ανοίγει μια TCP/TLS σύνδεση χωρίς να είναι υποχρεωμένος να γνωστοποιήσει τι θα μεταφέρει στο διαχειριστή του LAN και στο υπόλοιπο internet μετατρέποντας την κίνηση του ως προσωπικό δεδομένο και χωρίς την χρήση μη διαδεδομένων πρωτοκόλλων στο transmission.

OpenVPN

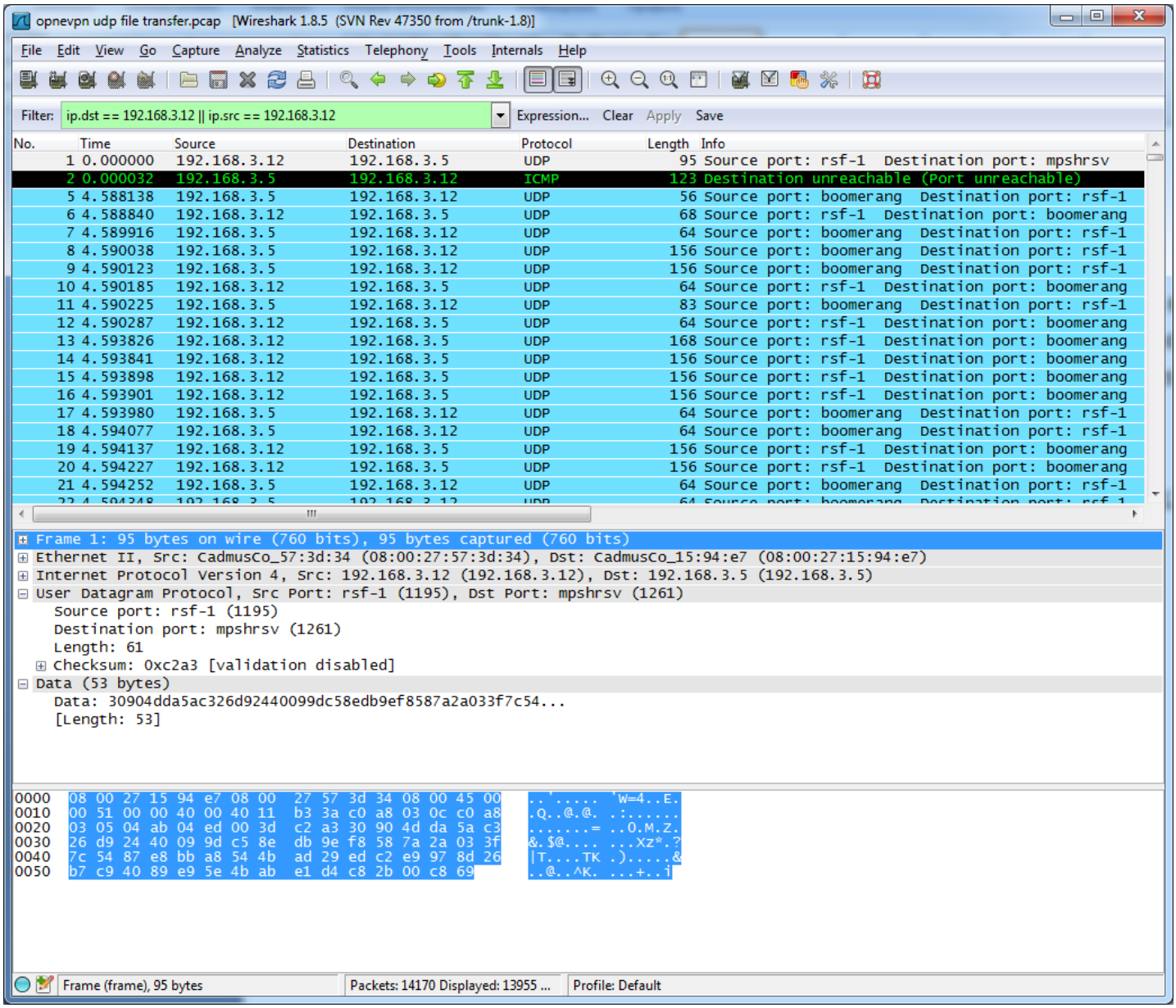
Η δημιουργία του openVPN έγκειται σε συγκεκριμένους παράγοντες όπου αφορούν τα κλασικά πρωτόκολλα VPN και στην αδυναμία τους να καλύψουν συγκεκριμένες ανάγκες. Όπως είδαμε προηγουμένως τα κλασικά πρωτόκολλα παρουσίαζαν υπερβολικά προτυποποιημένα χαρακτηριστικά σύνδεσης και ειδικούς transmission headers με αποτέλεσμα η κίνηση τους να αστυνομεύεται αρκετά εύκολα από αναλυτές κίνησης δικτύου ή να μπλοκάρεται. Σε αντίθεση το openVPN χρησιμοποιεί SSL/TLS επικοινωνίες με επιλογή για UDP ή TCP, με εσωτερική πιστοποίηση και χωρίς μεγάλη γνωστοποίηση του περιεχομένου όπου μεταδίδεται με αποτέλεσμα την καλύτερη αποφυγή αστυνόμευσης και φραγής. Το openVPN χρησιμοποιεί μόνο ένα socket για όλη την ανταλλαγή δεδομένων. Ένα άλλο του προσόν είναι οτι η εφαρμογή server παρουσιάζει χαρακτηριστικά αυτονομίας σε σχέση με το λειτουργικό σύστημα. Αυτό έρχεται σε αντίθεση με τα πρωτόκολλα τα οποία χρησιμοποιούν μεγάλη αλληλεπίδραση της εφαρμογής με το λειτουργικό. Αυτό σημαίνει οτι αφενός το openVPN καταλήγει να είναι πιο εύκολο στην εγκατάσταση και αφετέρου περιορίζει την διαρροή πληροφοριών στο λειτουργικό.

αρχικά θα πρέπει να σημειωθεί οτι το openVPN είναι σε θέση να ενθυλακώσει τα πακέτα εικονικής κίνησης τόσο σε UDP οσο και σε TCP. Για κάθε περίπτωση εμείς έχουμε αρχικοποιήσει το server κάθε φορά ώστε να υποστηρίζει τα πρωτόκολλα αντίστοιχα.

UDP κίνηση

χρησιμοποιώντας το γνωστό φίλτρο αυτή τη φορά με 192.168.3.12

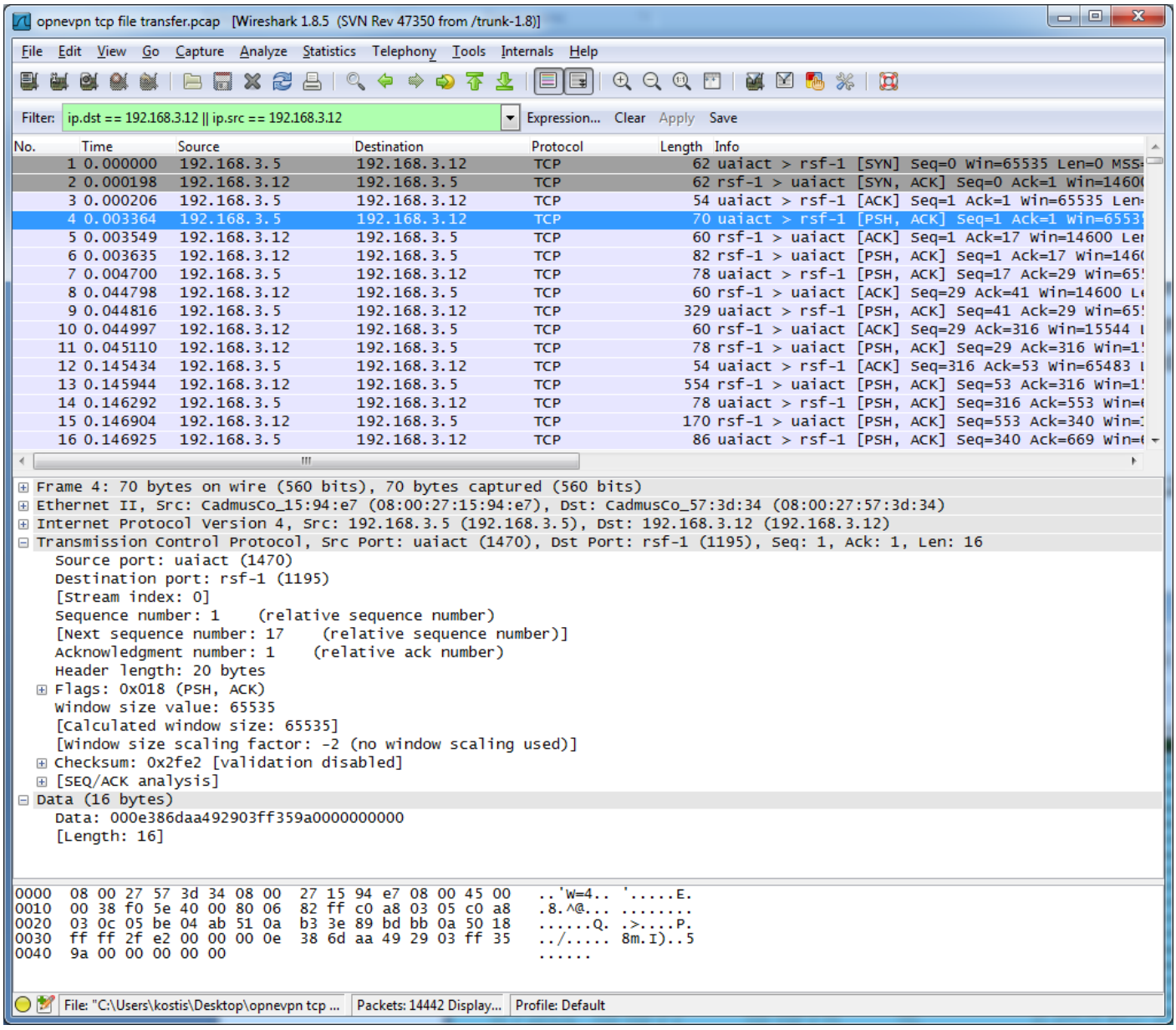
ip.dst == 192.168.3.12 || ip.src == 192.168.3.12



Εικόνα 36 Wireshark OpenVPN UDP

Βλέπουμε την πλήρη κίνηση από και προς τον server. Αυτό που παρατηρούμε άμεσα με μια ματιά είναι οτι **ολόκληρη η υπηρεσία είναι πλήρως κρυπτογραφημένη και ενθυλακωμένη σε UDP** και **δεν υπάρχουν ίχνη ούτε πιστοποίησης αλλά ούτε μεταφοράς πακέτων**. Ένας αναλυτής δικτύου ο οποίος έπαιρνε την κίνηση στα χέρια του μη γνωρίζοντας οτι είναι vpn δεν θα μπορούσε να το διακρίνει. Αυτό το χαρακτηριστικό είναι ακριβώς οτι επιθυμούμε να επιτύχουμε και στο unity!

TCP κίνηση



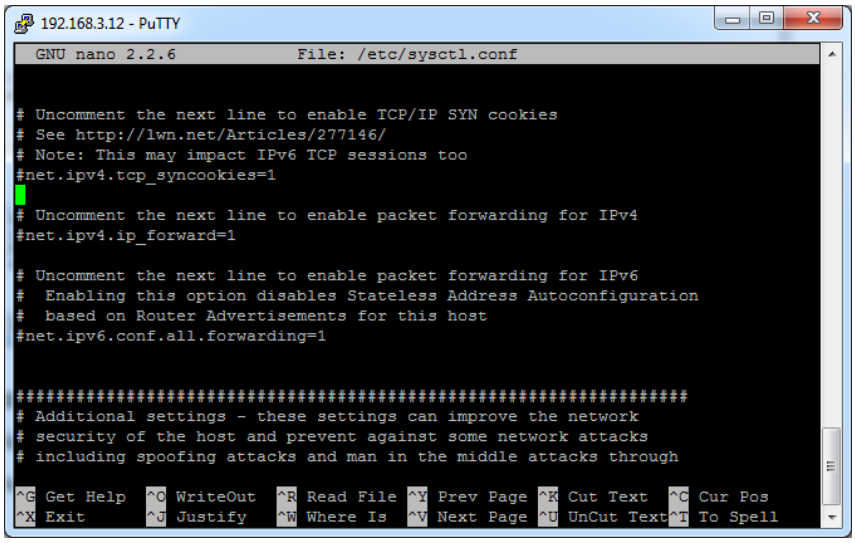
Εικόνα 37 Wireshark OpenVPN TCP

Αντίστοιχα και για TCP παρατηρούμε οτι δεν είναι εύκολο να διακρίνουμε οτι η κίνηση που μεταφέρεται είναι κίνηση δικτύου καθώς οι πληροφορίες της σύνδεσης προστατεύονται από το τούνελ.

Αυτονομία εφαρμογής και λειτουργικό σύστημα

Ένα άλλο πολύ σημαντικό χαρακτηριστικό είναι οτι η εσωτερική δρομολόγηση πακέτων στο λειτουργικό σύστημα είναι απενεργοποιημένη παρόλα αυτά ο openVPN λειτουργεί κανονικά! Αυτό μπορούμε να το διαπιστώσουμε κάνοντας read το αρχείο

/etc/sysctl.conf



Εικόνα 38 /etc/sysctl.conf

Θα διαπιστώσουμε οτι η γραμμή

net.ipv4.ip_forward=1

είναι σε σχόλια που είναι εντολή στο λειτουργικό να απαγορεύει την εσωτερική δρομολόγηση πακέτων

Παρόλα αυτά σε άλλες υλοποιήσεις του openVPN ίσως χρειαστεί να την ενεργοποιήσουμε γιατί ο στόχος μας μπορεί να είναι να ενώσουμε LAN

3.4 Σύγκριση με το unity

Στοιχεία ενδιαφέροντος

- Είδαμε οτι στις περιπτώσεις όπου υπάρχει μεγάλη προτυποποίηση μειώνεται δραματικά η ασφάλεια
- Επίσης όταν υπάρχει μεγάλη προτυποποίηση η κίνηση αναγνωρίζεται πιο εύκολα
- Το χαρακτηριστικό του openVPN οτι κρατάει όλες τις πληροφορίες πιστοποίησης ελέγχου ροής καθώς και τα πακέτα κάτω από μια θύρα είναι πολύ σημαντικό.
- Μας αρέσουν τα χαρακτηριστικά nat-traversing όπου προσφέρει το openVPN
- Μας αρέσει η ικανότητα αυτονομίας και αντικειμενοστρέφειας του του σε σχέση με το λειτουργικό σύστημα

Σημείωση: Να υπενθυμίσουμε οτι η κίνηση όπου είδαμε είναι εξωτερική αυτό σημαίνει οτι την έχουμε καταγράψει σαν να μην ήμασταν μέλη του δικτύου (από εξωτερική σκοπιά) ένα μέλος του VPN θα έβλεπε την εσωτερική κίνηση δηλαδή τα πακέτα του FTP κτλ.

Επίσης στο παράρτημα μπορούν να βρεθούν τα αρχεία κίνησης για επιπλέων μελέτη των πρωτοκόλλων!

Οι διαφορές των πρωτοκόλλων και τον openVPN με το Unity

Η πρώτη και βασικότερη διαφορά του unity με τα πρωτόκολλα VPN είναι οτι το Unity δεν είναι πρωτόκολλο, αντιθέτως είναι πλατφόρμα. Αυτό σημαίνει οτι το PPTP ή το L2TP δεν τα νοιάζει τίποτα άλλο παρά μόνο το VPN τούνελ και η αρχική του πιστοποίηση. Δεν ενδιαφέρονται ούτε για την εγγραφή του χρήστη σε μια υπηρεσία όπου θα τα χρησιμοποιούσε και ούτε παρουσιάζουν από μόνα τους κάποια κατανομή δικτύου. Ένας διαχειριστής θα τα εγκαταστήσει σε μια αρχιτεκτονική όπου θα αποφασίσει χειροκίνητα. Σε αντίθεση, το Unity είναι πλατφόρμα, αυτό σημαίνει οτι ενδιαφέρεται και για την αρχική εγγραφή του χρήστη και για την πιστοποίηση και επίσης πέρα από το τούνελ διαχειρίζεται και τους κόμβους όπου είναι υπεύθυνοι για την δρομολόγηση ως σύνολο! Ένας κόμβος μπορεί να προστεθεί και να αφαιρεθεί από το Unity δυναμικά.

Σε ποιά σημεία το Unity υπερέχει έναντι του OpenVPN

Το OpenVPN μοιράζεται αρκετά κοινά στοιχεία με το unity όπως οτι και αυτό παρουσιάζει περισσότερο χαρακτήρα πλατφόρμας παρά πρωτοκόλλου. Η διαφορά με το Unity έγκειται στην κατανομή! Πιο συγκεκριμένα το openVPN χρησιμοποιεί ένα κεντρικό server για εξυπηρέτηση πελατών (συγκεντρωτική λογική). Αυτό αρχικά σημαίνει:

- μεγάλη εξάρτηση από HW και θάνατος της πλατφόρμας στο θάνατο του κεντρικού μηχανήματος
- απουσία κατανομής, ο φόρτος της συνολικής πλατφόρμας είναι και ο φόρτος όπου αντέχει το κεντρικό μηχάνημα
- υπάρχει δυνατότητα επέκτασης αλλά μόνο με τη χρήση επιπλέον HW με 2ο VPN server όπου πρέπει να είναι στο ίδιο LAN και ο κάθε VPN server πιστοποιεί ανεξάρτητα τους χρήστες.

Σε αντίθεση το Unity εισάγει κατανεμημένη λογική:

- Ένας χρήστης όταν συνδέεται σε ένα BlueNode (vpn server) γνωστοποιείται σε όλη την πλατφόρμα
- BlueNodes (vpn servers) μπορούν να προστίθενται και να αφαιρούνται δυναμικά από τη πλατφόρμα χωρίς απαραίτητα να είναι στο ίδιο LAN
- Δημιουργία ενός μεγάλου εικονικού δωματίου χάρη στην κατανομή
- Ανεξαρτητοποίηση από το HW (άμα πέσει ένας BN δεν πέφτει όλη η πλατφόρμα και οι χρήστες μπορούν να συνδεθούν από αλλού)
- Μεγαλύτερη αντικειμενοστρέφεια: τα προγράμματα της πλατφόρμας δεν έχουν ανάγκη από ρύθμιση δικτύου είναι πλήρως αντικειμενοστραφείς και αυτό σημαίνει ότι απλά ανοίγουν και τρέχουν χωρίς να πειράζουν χαρακτηριστικά του λειτουργικού όπως packet forwarding

4. Λογική & ανάλυση της πλατφόρμας από δικτυακής όψης

4.1 Virtual Networking με TUN/TAP & Java

Εφόσον καταλάβαμε τις βασικές έννοιες για τα VPN αρχικά πρέπει να δούμε το πιο βασικό. Δηλαδή πως μπορεί το Unity με το RedNod και γενικότερα ένα πρόγραμμα να επικοινωνήσει με ένα Virtual TUN/TAP adapter.

Το πρώτο πράγμα όπου πρέπει να κάνει ένα μηχάνημα όπου επιθυμεί να χειρίζεται TUN/TAP είναι να τον εγκαταστήσει! Στις σύγχρονες διανομές Ubuntu υπάρχει ήδη εγκατεστημένος και για να ελέγξουμε οτι υπάρχει θα πρέπει να τρέξουμε

modprobe tuntap

αν δεν μας πετάξει κάποιο λάθος είναι εγγατεστημένος. Στα windows μπορούμε να χρησιμοποιήσουμε τον standalone installer του OpenVPN μόνο για τον TUN/TAP adapter του. (όπου περιέχεται και στο CD).

Μόλις βάλουμε τον adapter και συγκεκριμένα για **java** θα χρησιμοποιήσουμε την **clib** μια σειρά από precompiled libraries για όλα τα OSs και ένα πακέτο για εσωτερική χρήση στο πρόγραμμα το οποίο ανήκει στο **p2pvpn project**. Δεν θα γίνω πιο αναλυτικός στο πως βρίσκουμε τα αρχεία καθώς περιέχονται στο CD! Μέσα στο project ανοίγοντας την κλάση TunTap.java θα παρατηρήσουμε τις παρακάτω συναρτήσεις όπου είναι ήδη compiled στα libs.

```
/**
    * @return the name of the virtuel network device
    */
   abstract public String getDev();

   /**
    * Close the device.
    */
   abstract public void close();

   /**
    * Send a packet to the virtual network adapter.
    *
    * @param b the packet
    * @param len the length of the packet
    */
   abstract public void write(byte[] b, int len);

   /**
    * Read a packet from the virtual network adapter.
    *
    * @param b the packet
    * @return length if the packet
    */
   abstract public int read(byte[] b);
```

```
/**
 * Set the IP address of the virtual network adapter.
 *
 * @param ip the IP
 * @param subnetmask the subnet mask
 */
public void setIP(String ip, String subnetmask) {
   try {
      this.ip = InetAddress.getByName(ip).getAddress();
   } catch (UnknownHostException ex) {
   }
}

/**
 * Return the last set IP address.
 *
 * @return the ip address
 */
public byte[] getIPBytes() {
   return ip;
}
```

Όπως μπορούμε να δούμε οι δύο πιό σημαντικές συναρτήσεις σε ένα TUN/TAP είναι η write και η read

write(byte[] b, int len);

int read(byte[] b);

Ουσιαστικά αν ανοίξουμε την κάρτα δικτύου στο πρόγραμμα, με τη μία συνάρτηση διαβάζουμε πακέτα σε bytes και με την άλλη γράφουμε πακέτα σε Bytes! Φυσικά εφόσον μιλάμε για κάρτα δικτύου δεν θα πρέπει η κάρτα να κάνει read και μετά να περιμένει να κάνει write, οι 2 μέθοδοι θα πρέπει να λειτουργούν **ασύγχρονα!** Επίσης το πρόγραμμα και το OS έχουν access στον πόρο διά κοινού αυτό σημαίνει οτι αν για παράδειγμα ο χρήστης πάει να ανοίξει μια ιστοσελίδα από το browser του το OS θα κάνει use την κάρτα γράφοντας και διαβάζοντας πακέτα!

Πως ο RedNode χρησιμοποιεί τον adapter

Παρακάτω φαίνεται το κομμάτι κώδικα που κάνει access τον TUN/TAP όπου βρίσκεται μέσα στο **ConnectionManager.java**

```
public boolean startInterface() {

    lvl5RedNode.login.writeInfo("Setting Interface...");
    //starting the real interface
    try {
        tuntap = TunTap.createTunTap();
    } catch (Exception ex) {
        return false;
    }

    //stupid interface
    if (tuntap.getDev() == null) {
        if (libError == false) {
            for (int i = 0; i < 3; i++) {
                try {
                    tuntap = TunTap.createTunTap();
                } catch (Exception ex) {
                    return false;
                }
                if (tuntap.getDev() != null) {
                    break;
                }
                try {
                    sleep(2000);
                } catch (InterruptedException ex) {
                    Logger.getLogger(ConnectionManager.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
            lvl5RedNode.login.writeInfo("COULD NOT OPEN INTERFACE, check if your interface is activated and if its not being used");
        }
        tuntap = null;
        return false;
    }


     if (tuntap != null) {
         readMan = new QueueManager(20);
         writeMan = new QueueManager(1000);

         read = new InterfaceRead(lvl5RedNode.login.connection.tuntap);
         read.start();

         write = new InterfaceWrite(lvl5RedNode.login.connection.tuntap);
         write.start();

         router = new EthernetRouter();
         router.start();

         vrouter = new VirtualRouter();
         vrouter.start();
         return true;
     } else {
         return false;
     }
 }
```

Αρχικά γίνονται 3 προσπάθειες να ανοίξει το TUN/TAP. Εάν αυτό είναι εφικτό τότε θα ανοίξει τα δυο προαναφερθέντα threads (το ένα θα κάνει read και το άλλο write από αυτόν). Τα 2 αυτά thread με τη σειρά τους, το ένα θα προωθεί τα πακέτα τoy στο EthernetRouter, το InterfaceRead δηλαδή και το άλλο, το write θα γράφει οτι πει το ethernetRouter σε αυτό. Το ethernetRouter με τη σειρά του αν αποφασίσει οτι ένα πακέτο έχει νόημα δρομολόγησης (δηλαδή δεν είναι broadcast ή LAN spam γενικότερα) θα το στείλει στο virtual router και το virualRouter αν έχει κάτι όπου έφτασε από το network θα το στείλει στο EthernetRouter για να πακεταριστεί σε Frame!

Σύγχρονη Λίστα Παραγωγού Καταναλωτή

Εδώ πέρα γεννιέται ένα μεγάλο ερώτημα… Πως θα μεταφέρονται ουρές δεδομένων (δηλαδή πακέτων) μεταξύ ασύγχρονων νημάτων???

Την απάντηση σε αυτή την ερώτηση έρχεται να δώσει το παρακάτω κομμάτι κώδικα ο οποίος **είναι ο ελβετικός σουγιάς για τον προγραμματισμό δικτύων**

```
5   package Routing;
6
7   import java.util.LinkedList;
8   import java.util.Queue;
9   import java.util.logging.Level;
10  import java.util.logging.Logger;
11
12  /**
13   *
14   * @author kostis lvl3RedNode.ReceivedPacketQueue
15   */
16
17  public class QueueManager extends Thread {
18      private final int capacity;
19      private Queue<byte[]> queue;
20
21      public QueueManager(int capacity) {
22          this.capacity = capacity;
23          queue = new LinkedList();
24      }
25
26      public synchronized void offer(byte[] data) {
27
28          while(queue.size() == capacity) {
29              try {
30                  wait();
31              } catch (InterruptedException ex) {
32                  Logger.getLogger(QueueManager.class.getName()).log(Level.SEVERE, null, ex);
33              }
34          }
35
36          queue.add(data);
37          notify();
38      }
39
40      public synchronized byte[] poll() {
41          while(queue.isEmpty()) {
42              try {
43                  wait();
44              } catch (InterruptedException ex) {
45                  Logger.getLogger(QueueManager.class.getName()).log(Level.SEVERE, null, ex);
46              }
47          }
48
49          byte[] data  = queue.poll();
50          notify();
51          return data;
52      }
53
54      public synchronized void clear(){
55          queue.clear();
56          notify();
57      }
58
59      public int getlen() {
60          return queue.size();
61      }
62  }
63
```

Εικόνα 39 byte array list of synchronized producer and consumer

Στο παραπάνω κομμάτι κώδικα περιγράφεται μια ουρά δεδομένων για Byte arrays (δλδ στην προκειμένη πακέτα) με ενσωματωμένο **παραγωγό** και **καταναλωτή**, στόχος της να δουλέψει ανάμεσα σε 2 ασύγχρονες διεργασίες όπου η μία κάνει **push** και η άλλη **poll**. Όταν η ουρά είναι άδεια το νήμα που ζητάει πακέτα κοιμάται και όταν είναι γεμάτη το νήμα όπου παράγει κοιμάται. Με αυτό τον τρόπο καταφέρνουμε να περνάνε τα πακέτα από το ένα νήμα στο άλλο χωρίς να υπάρχουν **deadlocks!**

Η παραπάνω λίστα είναι κλάση και αντικείμενα της έχουν αρχικοποιηθεί σε όλη την έκταση του προγράμματος από νήματα που στέλνουν σε νήματα πακέτα.

για παράγειγμα...

InterfaceRead >queue> EthernetRouter >queue> VirtualRouter >queue> UDP socket write

Το InterfaceRead και τον InterfaceWrite φαίνονται παρακάτω, όσο για το Ethernet και Virtual router θα τα δούμε πιο μετά

```java
import RedNode.lvl5RedNode;
import org.p2pvpn.tuntap.TunTap;

/**
 *
 * @author kostis
 */
public class InterfaceWrite extends Thread {

    private String pre = "^WRITE ";
    boolean kill = false;
    byte[] data;
    private TunTap adapter;

    public InterfaceWrite(TunTap adapter) {
        this.adapter = adapter;
    }

    @Override
    public void run() {
        System.out.println("@Interface write started at " + Thread.currentThread().getName());

        int i = 0;
        while (!kill) {
            //tha pairnei paketa apo thn oura kai tha ta grafei sto meso, ama einai adeia tha koimatai gia ligo
            try {
                data = lvl5RedNode.login.connection.writeMan.poll();
            } catch (java.lang.NullPointerException ex1) {
                continue;
            } catch (java.util.NoSuchElementException ex) {
                continue;
            }

            lvl5RedNode.login.monitor.writeToIntWrite(pre
            lvl5RedNode.login.connection.tuntap.write(data
            lvl5RedNode.login.monitor.jTextField14.setText
            lvl5RedNode.login.monitor.jTextField12.setText
            i++;
        }
    }

    public void kill() {
        kill = true;
    }
}
```

```java
 */
public class InterfaceRead extends Thread{

    private String pre = "^READ ";
    TunTap adapter = null;
    boolean kill = false;

    public InterfaceRead(TunTap adapter) {
       this.adapter = adapter;
    }

    @Override
    public void run() {
        System.out.println("@Interface read started at "+Thread.currentThread().getName());

        byte[] buffer = new byte[2048];
        int i=0;
        while(!kill){
            lvl5RedNode.login.monitor.writeToIntRead(pre+"READING");
            int len = lvl5RedNode.login.connection.tuntap.read(buffer);
            lvl5RedNode.login.monitor.jTextField13.setText(""+lvl5RedNode.login.connection.readMan.getlen());
            if (len > 14) {
                byte[] frame = new byte[len];
                System.arraycopy(buffer, 0, frame, 0, len);
                lvl5RedNode.login.connection.readMan.offer(frame);
                lvl5RedNode.login.monitor.jTextField11.setText(""+i);
                i++;
            }
        }
    }

    public void kill(){
        kill=true;
    }
}
```

o writeMan και o readMan είναι τα queues όπου εχουν αρχικοποιηθεί στο connection και τα νήματα τα βρίσκουν από κει

Εικόνα 40 writeMan, readMan Threads

4.2 Ethernet Routing & ARP/DHCP Packet forging & IP checksum generate

Ωραία πλέον έχουμε μια βασική εικόνα για το πώς ο RedNode διαβάζει και γράφει πακέτα από την κάρτα και για το πως μεταφέρονται τα πακέτα μέσω ουρών από το ένα νήμα στο άλλο. Τώρα μένει να δούμε όσο αναφορά το low networking level πώς τα frames απο/ενθυλακόνωνται σε IP packets και πώς μπορούμε έχοντας τον έλεγχο της κάρτας, αρχικά να κάνουμε forge DHCP packets ώστε το host του RN να πάρει την Virtual IP όπου του απονέμει το δίκτυο, και πώς να παράγουμε ARP packets ώστε κάθε ψευδοτοπικός host να έχει και από μια ψευδο-mac.

Low Read

Αρχικά θα δούμε πρώτα το FrameRead που είναι πιο εύκολο και στη συνέχεια το FrameWrite. Όπως θυμόμαστε από πριν, όταν ένα πακέτο διαβαστεί από το InterfaceRead στελνεται στον readMan (την ουρά μας). Ο readMan διαβάζεται από τον ethernetRouter (όπου κάνει poll). Παρακάτω μπορούμε να δούμε το source του

```
22  */
23  public class EthernetRouter extends Thread {
24
25      private String pre = "^ETHOUTER ";
26      private boolean kill = false;
27      private String pver;
28      private int len;
29
30      @Override
31      public void run() {
32          byte[] frame;
33          MacAddress sourcemac;
34          MacAddress destmac;
35          InetAddress source;
36          InetAddress dest;
37          FrameType type;
38          byte[] ippacket;
39          EthernetConnection connection = new EthernetConnection();
40
41          while (!kill) {
42              //tha pairnei paketa apo thn oura kai tha ta grafei sto meso, ama einai adeia tha koimatai gia ligo
43              try {
44                  frame = lvl5RedNode.login.connection.readMan.poll();
45              } catch (java.lang.NullPointerException ex1) {
46                  continue;
47              } catch (java.util.NoSuchElementException ex) {
48                  continue;
49              }
50
51              sourcemac = Frame.GetSourceMacAddress(frame);
52              destmac = Frame.GetDestMacAddress(frame);
53              type = Frame.getFrameType(frame);
54
55              if (sourcemac == null || destmac == null || type == null) {
56                  continue;
57              }
58
59              String info;
60              info = pre + "READING ";
61              info = info + type.toString() + " ";
62              info = info + "Length: " + frame.length + " ";
63              info = info + "Dest: " + destmac.toString() + " ";
64              info = info + "Source: " + sourcemac.toString();
65
66              lvl5RedNode.login.monitor.writeToIntRead(info);
```

```
            //unicast packets
            if (!destmac.isBroadcast()) {
                lvl5RedNode.login.monitor.writeToIntRead(pre + "Unicast");
                if (type.toString().equals("IP")) {
                    ippacket = Packet.GetPacket(frame);

                    source = Packet.getSourceAddress(ippacket);
                    dest = Packet.getDestAddress(ippacket);
                    pver = Packet.getVersion(ippacket);
                    len = ippacket.length;

                    if (source == null || dest == null || !pver.equals("45")) {
                        continue;
                    }
                    if (len <= 0 || len > 1500) {
                        System.out.println("Discarded, wrong size");
                        continue;
                    }

                    String info2 = pre + "IP Frame Version: " + pver + " ";
                    info2 = info2 + "Source: " + source.getHostAddress() + " ";
                    info2 = info2 + "Dest: " + dest.getHostAddress();
                    lvl5RedNode.login.monitor.writeToIntRead(info2);

                    if (connection.clearToSendIP(ippacket)) {
                        lvl5RedNode.login.connection.arpTable.getByIP(dest).getTrafficMan().clearToSend();
                        lvl5RedNode.login.connection.upMan.offer(ippacket);
                    }
                } else {
                    lvl5RedNode.login.monitor.writeToIntRead(pre + "NOT INTRESTING FRAME");
                }
            } else {
                //broadcast packets
                lvl5RedNode.login.monitor.writeToIntRead(pre + "Broadcast");
                if (type.toString().equals("ARP")) {
                    lvl5RedNode.login.monitor.writeToIntRead(pre + "ARP");
                    connection.giveARP(frame);
                } else if (type.toString().equals("IP")) {

                    //make sure its bootstrap
                    if (DHCPrequest.isBootstrap(frame)) {
                        lvl5RedNode.login.monitor.writeToIntRead(pre + "BOOTSTRAP");
                        connection.giveBootstrap(frame);
                    } else {
                        lvl5RedNode.login.monitor.writeToIntRead(pre + "NOT RELEVANT");
                    }
                }
            }
        }
        lvl5RedNode.login.connection.readMan.clear();
        lvl5RedNode.login.monitor.jTextField11.setText("");
    }

    public void kill() {
        kill = true;
    }
}
```

Εικόνα 41 Frame routing algorithm

Σε όλη την διάρκεια υπάρχουν στατικές συναρτήσεις όπου διαβάζουν χαρακτηριστικά του frame προτού αποφασιστεί τίποτα. Αρχικά πραγματοποιείται ένας έλεγχος για null πακέτο. Εάν δεν ειναι null τοτε μέσω των συναρτήσεων διαβάζεται το source και dest mac και ο τύπος του payload του frame!

Ο πρώτος έλεγχος είναι αν το πακέτο είναι broadcast ανάλογα με τη διεύθυνση προορισμού. Αν δεν είναι bc υπάρχει μεγάλη πιθανότητα να απευθύνεται σε ένα άλλο host στο unity ενώ αν αν είναι bc υπάρχει πιθανότητα για ARP ή DHCP ή LAN spam. Στην πρώτη περίπτωση, το IP αποθυλακώνεται και ελέγχεται για σωστό destination (oxi bc, default ή μηδέν). Αν το dst φαίνεται να είναι μέλος του δικτύου θα μπεί στο read του VirtualRouter όπου διαχειρίζεταί μονο valid IP packets και από εκεί και πέρα είναι το λογικό μονοπάτι του δικτύου. Αν έχουμε κάτι σε ARP broadcast πάει να πει οτι μάλον ο host μας ψάχνει να μάθει σε ποια MAC αντιστοιχεί μια virtual IP του δικτύου. Στην συγκεκριμένη

περίπτωση θα ειδοποιήσει το connection να τον ταΐσει με ένα ARP response. Επίσης ελέγχεται και για DHCP και αν εντοπιστεί request ειδοποιεί πάλι το connection για την εμφάνιση του DHCP.

Αυτά όσο αναφορά το ethernetRead.

Low Write

Όπως μπορούμε να υποπτευθούμε στο write γράφονται IP πακέτα όπου έχουν φτάσει σε ένα RN διαμέσω του unity αφού πρώτα ενθυλακωθούν σε frame με το αντίστοιχο mac της ip. Επίσης υποπτευόμαστε οτι θα πρέπει να γίνονται generate τα ARP και DHCP replies και για να γίνει αυτό, στα DHCP θα πρέπει υπογράφουμε και το πακέτα με IP checksums.

DHCP Generate

Το DHCP είναι μια πολύπλοκη διαδικασία όπου έχει φτιαχτεί σε low level reverse engineering με τη χρήση του wireshark. Περιλαμβάνει DHCP request , ack και nack τα οποία χρησιμεύουν στην αρχική εγκαθίδρυση της σύνδεσης. Αν θέλει να το μελετήσει κάποιος καλύτερα να δεί τον κώδικα.

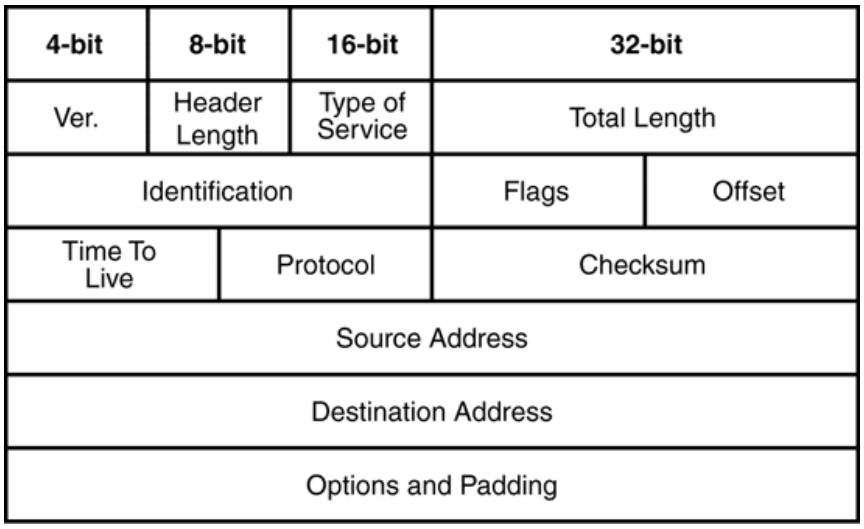
Για όλα αυτά τα πακέτα δημιουργούνται με

Sytem.arraycopy σε byte arrays στο παρακάτω στύλ

```
48          //type
49          opt[0] = new byte[]{0x35, 0x01, 0x02};
50
51          //54 dhcp server ip
52          byte[] didtype = new byte[]{0x36};
53          byte[] didlen = new byte[]{0x04};
54          dhcpIp = null;
55          try {
56              dhcpIp = InetAddress.getByName("10.0.0.1");
57          } catch (UnknownHostException ex) {
58              Logger.getLogger(DHCPGenerate.class.getName()).log(Level.SEVERE, null, ex);
59          }
60          opt[1] = new byte[didtype.length + didlen.length + dhcpIp.getAddress().length];
61          System.arraycopy(didtype, 0, opt[1], 0, didtype.length);
62          System.arraycopy(didlen, 0, opt[1], 1, didlen.length);
63          System.arraycopy(dhcpIp.getAddress(), 0, opt[1], 2, dhcpIp.getAddress().length);
64
65          //51 ip addr lease time
66          byte[] lstype = new byte[]{0x33};
67          byte[] lslen = new byte[]{0x04};
68          byte[] lstime = new byte[]{(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff};
69          opt[2] = new byte[lstype.length + lslen.length + lstime.length];
70          System.arraycopy(lstype, 0, opt[2], 0, lstype.length);
71          System.arraycopy(lslen, 0, opt[2], 1, lslen.length);
72          System.arraycopy(lstime, 0, opt[2], 2, lstime.length);
73
74          //12 Host name
75          byte[] hostype = new byte[]{0x0c};
76          byte[] hostlen = new byte[]{(byte) lvl5RedNode.login.connection.Hostname.getBytes().length};
77          byte[] hostname = lvl5RedNode.login.connection.Hostname.getBytes();
78          opt[3] = new byte[hostype.length + hostlen.length + hostname.length];
79          System.arraycopy(hostype, 0, opt[3], 0, hostype.length);
80          System.arraycopy(hostlen, 0, opt[3], 1, hostlen.length);
81          System.arraycopy(hostname, 0, opt[3], 2, hostname.length);
82
83          //1 subnet mask
84          byte[] stype = new byte[]{0x01};
85          byte[] slen = new byte[]{0x04};
86          byte[] smask = new byte[4];
87          try {
88              smask = InetAddress.getByName("255.0.0.0").getAddress();
89          } catch (UnknownHostException ex) {
90              Logger.getLogger(DHCPGenerate.class.getName()).log(Level.SEVERE, null, ex);
```

Εικόνα 42 A packet is a byte[] sandwich

Ουσιαστικά ένα πακέτο δεν είναι τίποτα άλλο από ένα byte[] array σάντουιτς που προσπαθεί να αποδώσει το IP μοντέλο.



Εικόνα 43 IP Header

μόλις τελειώσουμε το generate ενός IP πακέτου το υπογράφουμε με την παρακάτω συνάρτηση IP checksum , στη συνέχεια το ενθυλακώνουμε σε frame και το κάνουμε offer στο writeMan με την παρακάτω δήλωση

lvl5RedNode.login.connection.writeMan.offer(genframe);

```
    * @param buf The message
    * @return The checksum
    */
   public static long calculateChecksum(byte[] buf) {
     int length = buf.length;
     int i = 0;

     long sum = 0;
     long data;

     // Handle all pairs
     while (length > 1) {
       // Corrected to include @Andy's edits and various comments on Stack Overflow
       data = (((buf[i] << 8) & 0xFF00) | ((buf[i + 1]) & 0xFF));
       sum += data;
       // 1's complement carry bit correction in 16-bits (detecting sign extension)
       if ((sum & 0xFFFF0000) > 0) {
         sum = sum & 0xFFFF;
         sum += 1;
       }

       i += 2;
       length -= 2;
     }

     // Handle remaining byte in odd length buffers
     if (length > 0) {
       // Corrected to include @Andy's edits and various comments on Stack Overflow
       sum += (buf[i] << 8 & 0xFF00);
       // 1's complement carry bit correction in 16-bits (detecting sign extension)
       if ((sum & 0xFFFF0000) > 0) {
         sum = sum & 0xFFFF;
         sum += 1;
       }
     }

     // Final 1's complement value correction to 16-bits
     sum = ~sum;
     sum = sum & 0xFFFF;
     return sum;
   }
 }
```

Εικόνα 44 IP checksum generator

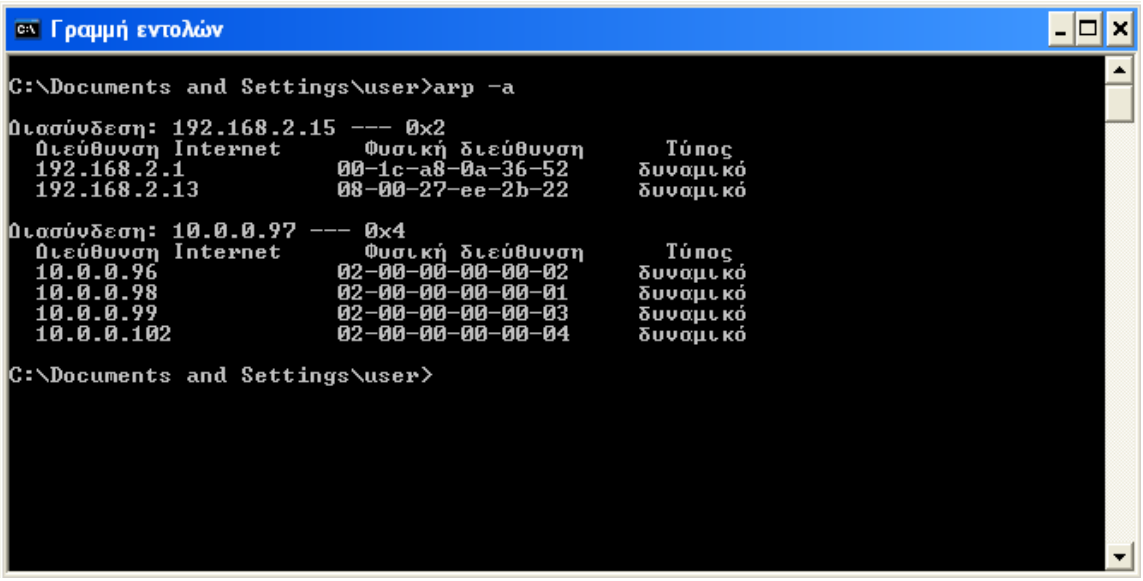
Reverse ARP table

Στο ARP ακολουθεί μια παρόμοια διαδικασία. Είναι πιο απλό πρωτόκολλο και έχει μόνο request και reply. Στόχος του είναι η κάθε virtual IP να έχει και μια mac. Προκειμένου να γίνει κάτι τέτοιο, το πρόγραμμα χρειάζεται ένα εσωτερικό ARP table όπου θα κρατάει τις αντιστοιχίες και θα βάζει τη σωστή mac στο frame ενός IP. Μια εγγραφή στο εσωτερικό ARP table μπορεί να δημιουργηθεί είτε όταν ψάξει ο host ένα node ή όταν φτάσει κάτι από ένα node. Έκτοτε μια IP θα έχει την ίδια mac και αν το OS ξεχάσει και ξανακάνει request θα πάρει την ίδια mac!

Επομένως ο RedNode δημιουργεί πλασματικές αντιστοιχίες. Μπορούμε αν θέλουμε όσο τρέχει να δώσουμε την εντολή

arp -a

και να δούμε όλες τις ψευδο-mac όπου έχουν δημιουργηθεί στο host μας.



4.3 Virtual Routing

Ωραία έχουμε πλέον ξεφύγει από το low networking level και ξέρουμε ότι μέσω του παραπάνω μηχανισμού ένα έγκυρο IP πακέτο θα πρέπει με κάποιο τρόπο να φύγει από τον RN και να σταλεί στον BN και επίσης με κάποιο τρόπο πρέπει ο RN να δέχεται και πακέτα από το BN.

Η απάντηση σε αυτό το ερώτημα είναι οτι ο RN από εξωτερικής σκοπιάς του δικτύου συνδέεται ως UDP client στον BN με 2 νήματα, το ένα εφόσον φακελώσει ένα IP πακέτο σε envelope το στέλνει μέσω UDP στον BN και το άλλο οτι δεχτεί το αποφακελώνει και το στέλνει για frame στην κάρτα. Το envelope έχει μερικές ακόμα πληροφορίες όπως αριθμό πακέτου και τύπο για καλύτερο manage.

Ο BlueNode δεν χρειάζεται καν εικονική κάρτα δικτύου, βλέπει τα πακέτα που φτάνουν ελέγχει την IP και τα προωθεί αντίστοιχα στο σωστό UDP τούνελ ενός συνδεμένου RN.

Επίσης Οι BNs και αυτοί συνδέονται μεταξύ τους με UDP και όπως έχουμε εξηγήσει εάν ένα πακέτο απευθύνεται σε ένα όχι τοπικό RN τότε ο BN ψάχνει για τον απομακρυσμένο υπεύθυνο BN και στέλνει σε εκείνον το πακέτο.

Αντίστοιχα εάν δεν γνωρίζει ποιός BN είναι υπεύθυνος για τον RN προορισμό ρωτάει τον tracker

Η λογική ενός BN στη δρομολόγηση ενός πακέτου φαίνεται παρακάτω.

```
25      @Override
27      public void run() {
28          lvl5BlueNode.ConsolePrint(pre + "started routing at thread " + Thread.currentThread().getName());
29
30          while (true) {
31              /*
32               * ok you got something... now lets check if destination is
33               * registered check vaddress table and sent to specific udp vaddr -
34               * addr:udp then when you get the stuff send it
35               */
36
37              try {
38                  data = lvl5BlueNode.manager.poll();
39              } catch (java.lang.NullPointerException ex1) {
40                  continue;
41              } catch (java.util.NoSuchElementException ex) {
42                  continue;
43              }
44
45              version = IpPacket.getVersion(data);
46
47              if (version.equals("45") || version.equals("1") || version.equals("2")) {
48                  if (version.equals("45")) {
49                      this.destvaddress = IpPacket.getDestAddress(data).getHostAddress();
50                      this.sourcevaddress = IpPacket.getSourceAddress(data).getHostAddress();
51                      lvl5BlueNode.TrafficPrint(pre + "IP " + sourcevaddress + " -> " + destvaddress + " " + data.length + "B", 3, 0);
52                  } else {
53                      this.destvaddress = IpPacket.getUDestAddress(data).getHostAddress();
54                      this.sourcevaddress = IpPacket.getUSourceAddress(data).getHostAddress();
55                      lvl5BlueNode.TrafficPrint(pre + version + " " + sourcevaddress + " -> " + destvaddress + " " + data.length + "B", 3, 0);
56                  }
57
58                  if (BlueNode.lvl5BlueNode.localRedNodesTable.checkOnlineByVAddr(destvaddress)) {
59                      //load the packet data to target users lifo
60                      BlueNode.lvl5BlueNode.localRedNodesTable.getRedNodeInstanceByAddr(destvaddress).getQueueMan().offer(data);
61                      lvl5BlueNode.TrafficPrint(pre + "LOCAL DESTINATION", 3, 0);
62                  } else if (lvl5BlueNode.joined) {
63                      if (BlueNode.lvl5BlueNode.remoteRedNodesTable.checkAssociatedByVAddr(destvaddress)) {
64                          String hostname = BlueNode.lvl5BlueNode.remoteRedNodesTable.getRedRemoteAddressByVAddr(destvaddress).getBlueNodeHostname();
65                          System.out.println("BN is "+hostname);
66                          BlueNode.lvl5BlueNode.BlueNodesTable.getBlueNodeInstanceByHn(hostname).getQueueMan().offer(data);
67                          lvl5BlueNode.TrafficPrint(pre + "REMOTE DESTINATION -> " + hostname, 3, 1);
                        } else {
```

```
68                          lvl5BlueNode.TrafficPrint(pre + "NOT KNOWN BN WITH " + destvaddress, 3, 1);
69                          lvl5BlueNode.flyreg.seekDest(sourcevaddress, destvaddress);
70                      }
71                  } else {
72                      lvl5BlueNode.TrafficPrint(pre + "NOT IN THIS BN " + destvaddress, 3, 1);
73                  }
74              } else {
75                  System.err.println("wrong header packet detected in router " + Thread.currentThread());
76              }
77          }
78      }
79  }
```

Εικόνα 45 Virual Routing

BlueNode Register on the Fly

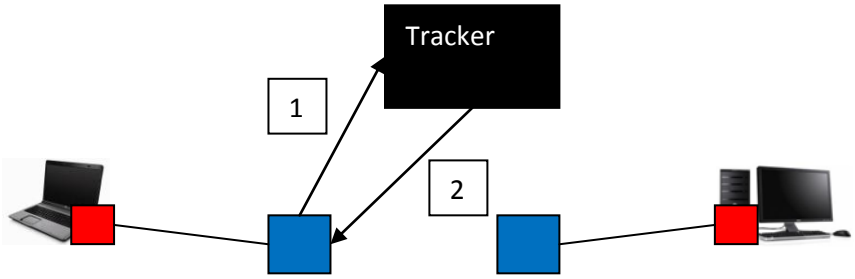
Εδώ πέρα είναι και όλα τα λεφτά του δικτύου! Έστω λοιπόν οτι έχουμε μια κατάσταση όπου ένας RN θέλει να στείλει σε ένα οχι τοπικό RN. Επίσης οι BNs τους δεν έχουν αναγνωριστεί. Τι γίνεται σε αυτή την περίπτωση;

Εικόνα 46 Register On The Fly A

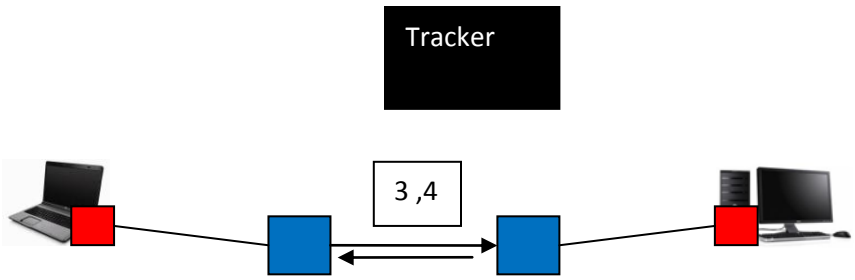
Για αυτή την περίπτωση όπως είδαμε στον κώδικα υπάρχει η δήλωση

lvl5BlueNode.flyreg.seekDest(sourcevaddress, destvaddress);

Ο flyreg (από το Register on the Fly) είναι ένα νήμα όπου δέχεται αγνώστους προορισμούς από πακέτα, ρωτάει τον tracker για την θέση του προορισμού, κάνει εγκαθίδρυση γραμμής με τον BN προορισμού και από εκεί και πέρα τα πακέτα στέλνονται κανονικά στον προορισμό τους. Αυτή η διαδικασία θα κρατήσει λίγα δευτερόλεπτα και το μόνο αντίκτυπο θα είναι ο RN να χάσει τα πρώτα πακέτα που έστειλε στον Remote RN. Με αυτό τον τρόπο το δίκτυο γίνεται απόλυτα δυναμικό καθώς βρίσκει μόνο του ποιός ανήκει που και τους ταιριάζει!



Εικόνα 47 Register On The Fly B



Εικόνα 48 Register On The Fly C

5. Λογική & ανάλυση της πλατφόρμας από όψη κατανομής

Σε αυτό το κεφάλαιο θα αναλύσουμε την πλατφόρμα από την εξωτερική της σκοπιά ως κατανεμημένο σύστημα για να δούμε το πως μπορεί να λειτουργεί συλλογικά, τι μορφή έχουν τα μηνύματα όπου ανταλλάσσονται καθώς και πως ο ένας κόμβος πιστοποιεί τον άλλο ως γνωστό και έμπιστο. Τέλος θα αναλυθεί η λειτουργία της κεντρική ιστοσελίδας και θα παρουσιαστούν θέματα ασφάλειας όπου την αφορούν.

5.1 Ανταλλαγή μηνυμάτων

Όλα τα είδη κόμβων στο unity προκειμένου να συνεννοηθούν ανταλλάσουν σύντομα μηνύματα ερωτήσεων και απαντήσεων μέσω **TCP socket**. Οι συνδέσεις είναι μη-συνδεσμοστραφείς δηλαδή εάν ένας κόμβος ρωτήσει κάποιον και στη συνέχεια θέλει να ρωτήσει και κάτι άλλο θα πρέπει να ξανά-πιστοποιηθεί. Αυτή η μέθοδος επιλέχτηκε καθώς όλοι οι κόμβοι ανταλλάσουν μικρά και πολύ σύντομα μηνύματα. Επίσης ο κάθε κόμβος μιλάει με πολλούς άλλους και το να κρατάει συνδεσμοστρεφείς συνδέσεις εκεί όπου δεν είναι απαραίτητο θα δημιουργούσε περιττό φόρτο και πολυπλοκότητα.

Επίσης δεν έχει επιλεχθεί κάποιο πρωτόκολλο όπως ws4d ή SOAP ή γενικότερα web services όπου είναι στη μόδα καθώς εισάγουν περιττή πολυπλοκότητα στο συνολικό project. Το unity λόγω της εκτενής του διείσδυσης σε αρκετούς τομείς θα πρέπει να παρουσιάζει όσο διακριτή και αυτόνομη λογική γίνεται. Δεν έχει ανάγκη από κάποιο νέο και γυαλιστερό πρωτόκολλο, έχει ανάγκη από αυτονομία και σταθερότητα καθώς έχει αρκετά πολύπλοκη αρχιτεκτονική. Εάν ακολουθούσαμε αυτό το μοντέλο θα έπρεπε να διαχειριζόμαστε και να αναβαθμίζουμε μεγάλες και εκτενείς βιβλιοθήκες για πολλά πράγματα όπου κάνουμε και μικρή χρήση εν τέλει. Αυτό θα σήμαινε συχνή αλλαγή του κώδικα βάση των άλλων βιβλιοθηκών και κάτι τέτοιο δεν μας αρέσει.

Tracker service

Παρακάτω θα δούμε τι είδους ερωτήσεις μπορεί να κάνει κάποιος στον κεντρικό Tracker και τι απαντήσεις μπορεί να δεχτεί πίσω.

Tracker σε BlueNodes:

Αρχικά μόλις δηλωθεί οτι είναι BlueNode απαιτείται πιστοποίηση μέσω RSA , θα εξηγηθεί αργότερα πώς γίνεται. Στη συνέχεια ένας BN μπορεί να κάνει εγγραφή στο δίκτυο με LEASE. Αφού κάνει LEASE μπορεί να ξεκλειδώσει και άλλες ερωτήσεις.

1. [Authenticate]

- LEASE : εισέρχεται λειτουργικά στο δίκτυο

1. [Authenticate]
2. [if LEASED]

- LEASE_RN : εισάγει ένα RN στο δίκτυο
- RELEASE : αποσυνδέεται από την πλατφόρμα
- RELEASE_RN : αποσυνδέει ένα RN από την πλατφόρμα
- UPDATE : ανανεώνει την IP του (ακριβώς σαν dyndns)
- GETPH : ζητάει την IP ενός άλλου BN
- CHECKRN : ζητάει να μάθει αν ένας RedNode βρίσκεται στο δίκτυο βάση του hostname του
- CHECKRNA : ζητάει να μάθει αν ένας RN βρίσκεται στο δίκτυο βάση της IP του
- GETBNPUBKEY : ζητάει να κατεβάσει το δημόσιο κλειδί ενός άλλου BN
- GETRNPUBKEY : ζητάει να κατεβάσει το δημόσιο κλειδί ενός RN

Tracker σε RedNodes:

Για τους RNs δεν απαιτεί πιστοποίηση καθώς δίνει μόνο πληροφορίες για την είσοδό τους στο δίκτυο όπως που μπορούν να βρουν BNs και ποιός είναι ο προτεινόμενος για σύνδεση. Οι εντολές πιο αναλυτικά είναι:

- GETBNS : επιστρέφει μια λίστα με τους διαθέσιμους ενεργούς BN στην πλατφόρμα
- GETRBN : επιστρέφει τον προτεινόμενο BN για σύνδεση με το χαμηλότερο φόρτο
- GETBNPUBKEY : επιστρέφει το δημόσιο κλειδί ενός BN βάση του hostname

Όπως είπαμε η σύνδεση είναι RAW οπότε ένας RN μπορεί να συνδεθεί και με κάποιον RAW TCP client όπως ncat ή telnet και να πάρει απάντηση εάν επιθυμεί

Βέβαια εάν επιχειρήσει την ίδια λογική για BN θα φάει πόρτα καθώς απαιτείται πιστοποίηση!

BlueNodes service

Σε αυτή την κατηγορία οι BNs είναι οι servers. Πρόκειται για τον πιο πολυάσχολο server σε όλη την πλατφόρμα καθώς ο Tracker συνεχώς ρωτάει για το αν η διεύθυνση ανανεώθηκε, οι RNs ζητάνε να πιστοποιηθούν, Οι BNs μεταξύ τους πρέπει να κάνουν associate και υπάρχει πολύ συχνή επικοινωνία και ανταλλαγή πληροφοριών.

BlueNode σε RedNode

[Authenticate]

- LEASE : σύνδεση στο δίκτυο

BlueNode σε BlueNode

[Authenticate]

- ASSOCIATE : εγκαθίδρυση σύνδεσης με ένα άλλο BN
- FULL_ASSOCIATE : εγκαθίδρυση σύνδεσης με ένα άλλο BN και ανταλλαγή RNs
- GET_RED_HOSTNAME : ένας BN δίνει μια IP ως όρισμα και ζητάει να μάθε σε ποιόν RN αντιστοιχεί
- GET_RED_VADDRESS : ένας BN δίνει ένα RN hostname ως όρισμα και ζητάει να μάθε σε ποιά vaddr αντιστοιχεί
- CHECK_H : ελέγχει άμα ένας RN είναι online στο συγκεκριμένο BN βάση hostname
- CHECK_V : ελέγχει άμα ένας RN είναι online στο συγκεκριμένο BN βάση virtual address
- RELEASE : αποδεσμεύει τον εαυτό του από τον BN
- UPING : έλεγχος λειτουργίας εξερχόμενης σύνδεσης
- DPING : έλεγχος λειτουργίας εισερχόμενης σύνδεσης
- GET_RED_NODES : ζητάει να μάθει όλους τους RN όπου εξυπηρετεί ο συγκεκριμένος BN
- EXCHANGE_RED_NODES : ανταλλαγή RN μεταξύ των 2 BN
- FEED_RETURN_ROUTE : ταΐζει τον BN με μία διεύθυνση επιστροφής (ώστε να γλυτώσει να ρωτήσει τον tracker)

BlueNode σε Tracker

- GETREDNODES : Ο BN στέλνει στον Tracker όλους τους RN όπου είναι συνδεμένοι πάνω του

5.2 Πιστοποίηση κόμβων μέσω RSA

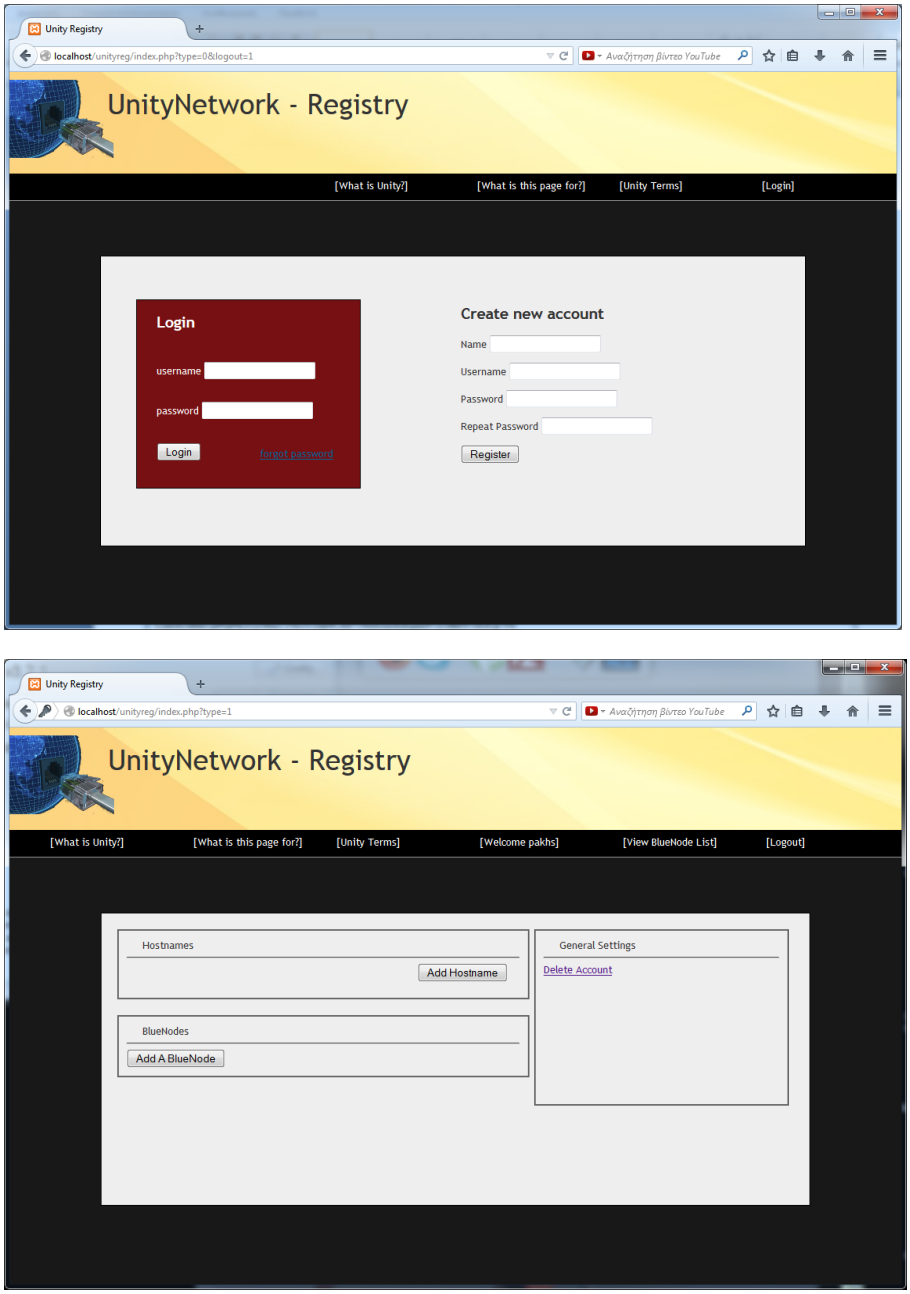
Μέχρι τώρα έχουμε καταλάβει πως η πλατφόρμα λειτουργεί ως σύνολο και παρέχει υπηρεσίες. Αυτό όπου δεν έχουμε δει ακόμα είναι πώς καταφέρνει να πιστοποιεί τους κόμβους της ως έμπιστους και να μην επιτρέπει σε υποδυόμενους κόμβους να συνδέονται. Η πλατφόρμα χρησιμοποιεί πιστοποίηση μέσω RSA.

Ας πάρουμε τα πράγματα από την αρχή:

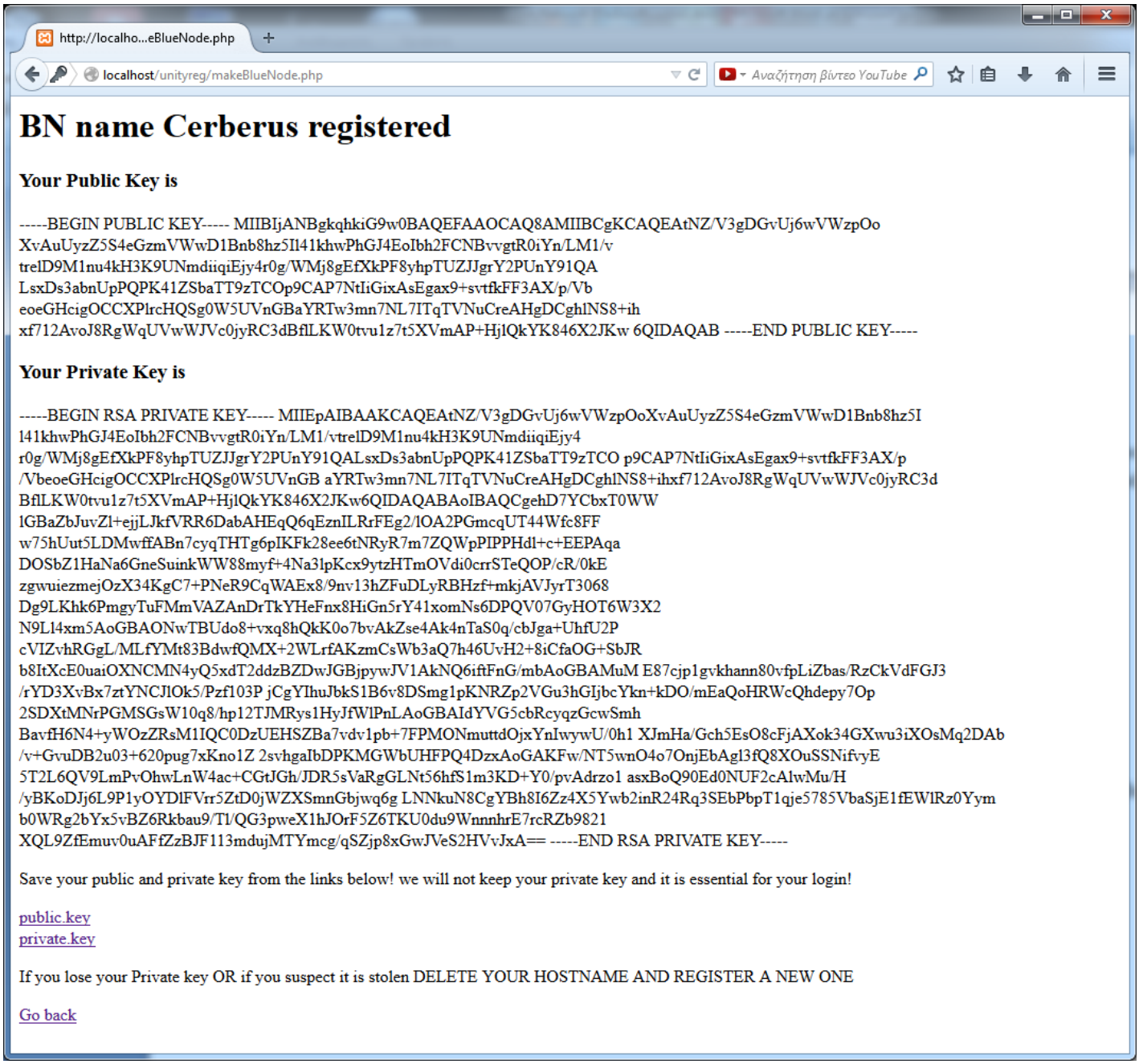
Registry page

Αρχικά ένας χρήστης θα πρέπει να έχει ένα λογαριασμό στην σελίδα του registry

αν θέλει να κατοχυρώσει ένα RN ή ένα BN εισέρχεται στην κεντρική σελίδα του registry με το username και το password του και δηλώνει το hostname. Η register page με τη σειρά της θα παράγει ένα ζευγάρι κλειδιών!



Εικόνα 49 Registry page



Εικόνα 50 Generate Public Private key for a BlueNode

Στη συνέχεια ο χρήστης σώζει το δημόσιο και το ιδιωτικό του κλειδί κάτω από το dir του BN ή του RN

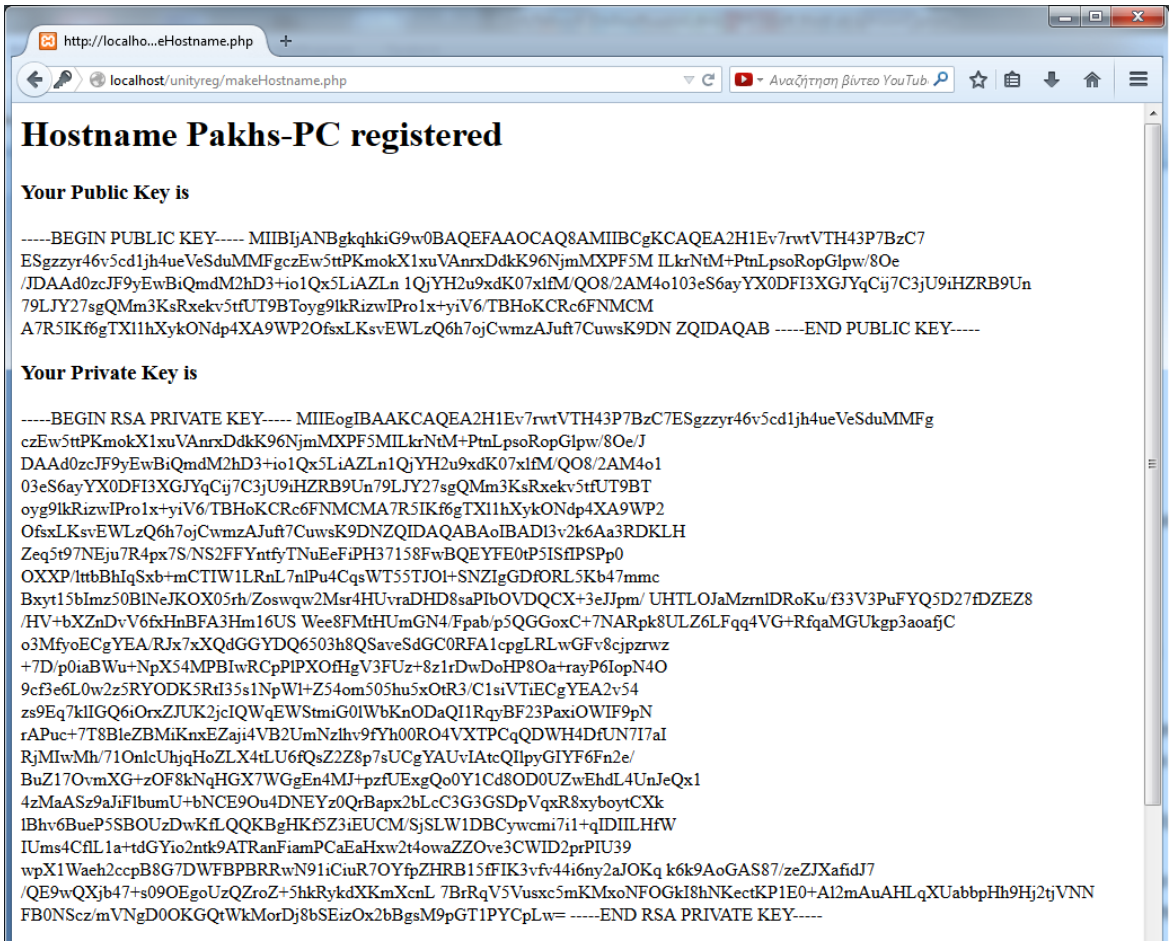
Η Βάση δεδομένων αναλαμβάνει να κρατήσει **MONO** το δημόσιο κλειδί και πλέον εάν ο συγκεκριμένος node δηλώσει το συγκεκριμένο name δεν μπορεί να μπει στο δίκτυο **παρά μόνο** εάν κατέχει το ιδιωτικό κλειδί!

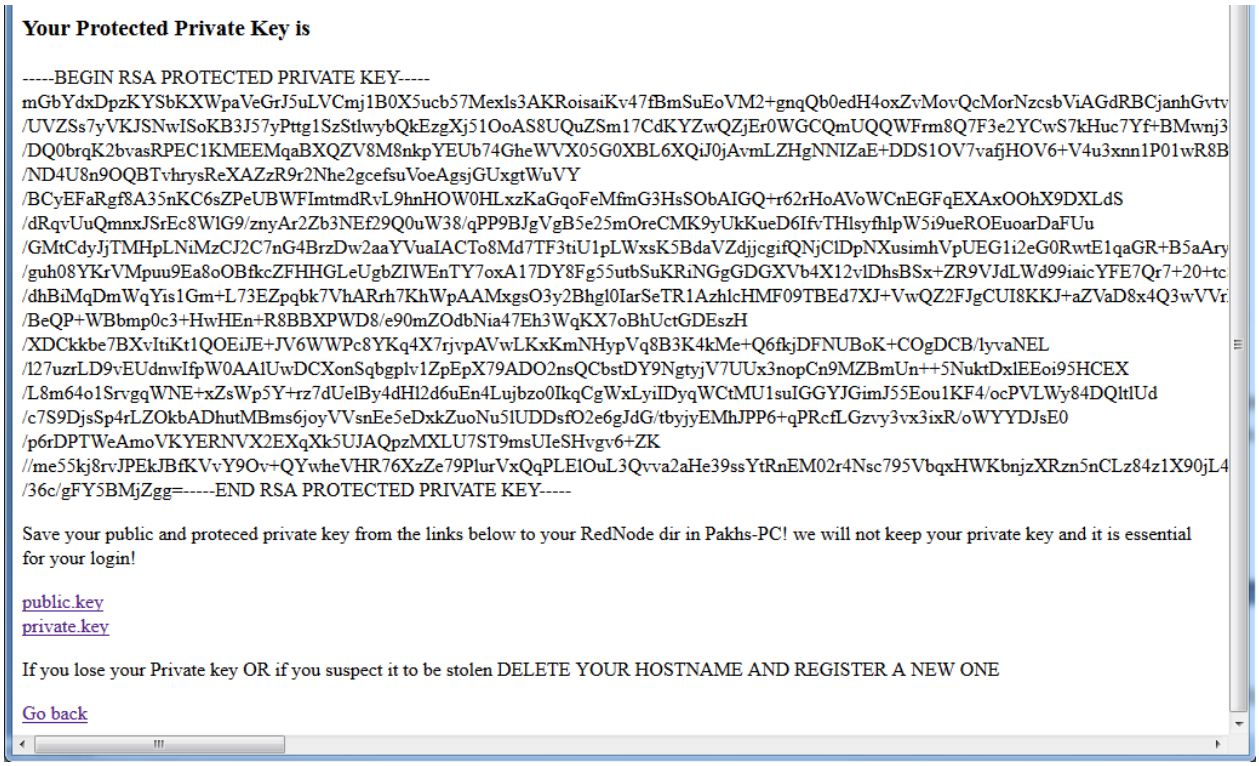
+ Επιλογές

	id	name	publicKey	userid
Επεξεργασία Αντιγραφή Διαγραφή	31	Lakhs	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	25
Επεξεργασία Αντιγραφή Διαγραφή	32	Pakhs	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	25
Επεξεργασία Αντιγραφή Διαγραφή	33	Takhs	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	25
Επεξεργασία Αντιγραφή Διαγραφή	34	Makhs	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	25
Επεξεργασία Αντιγραφή Διαγραφή	35	NIKOS_BN1	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	26
Επεξεργασία Αντιγραφή Διαγραφή	36	NIKOS_BN3	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	26
Επεξεργασία Αντιγραφή Διαγραφή	37	NIKOS_BN4	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	26
Επεξεργασία Αντιγραφή Διαγραφή	38	NIKOS_BN5	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	26
Επεξεργασία Αντιγραφή Διαγραφή	39	Cerberus	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQ...	25

Εικόνα 51 BlueNodes table on database

Για το RN υπάρχει ένα επιπλέον μέτρο ασφάλειας: Αρχικά δημιουργείται ένα ζευγάρι κλειδιών και στη συνέχεια **Κρυπτογραφείται το ιδιωτικό κλειδί με password**.





Εικόνα 52 RedNode Gernerate Public & Protecred Key

Αυτό το μέτρο ασφάλειας γίνεται για να έχει ο user **διαφορετικό password για κάθε hostname** και αυτό είναι αναγκαίο για 3 λόγους:

1. Για να μην κάνει χρήση του master password του λογαριασμού του όταν συνδέεται αλλά να κάνει χρήση του password του εκάστοτε hostname. Το master password χρησιμοποιείται μόνο στη σελίδα και έτσι είναι πιο ασφαλές και δύσκολο να κλαπεί
2. Εάν κλαπεί το προστατευμένο ιδιωτικό κλειδί του χρήστη να είναι άχρηστο καθώς ο επιτιθέμενος δεν έχει το password για να το ανοίξει
3. Ακόμα και το ιδιωτικό κλειδί να κλαπεί αλλά και το password να βρεθεί από τον επιτιθέμενο με κάποιο τρόπο δεν παραβιάζεται η ασφάλεια και των υπολοίπων Hostname αλλά ούτε και του λογαριασμού, ο χρήστης μπορεί να διαγράψει το hostname και να το ξαναδημιουργήσει με **νέο ζευγάρι!**

Σημείωση:

Σε αυτό το σημείο θα πρέπει να δηλώσουμε οτι αυτή η μέθοδος δημιουργίας ζευγαριού κλειδιών **δεν είναι απόλυτα ασφαλής**. Εμείς για λόγου απλότητας της πλατφόρμας κάναμε την κεντρική σελίδα να παράγει το ζευγάρι κλειδιών των hostname. Αυτό είναι λάθος (παρόλο που δεν σώζουμε το ιδιωτικό κλειδί στην βάση). Ένας χρήστης **πάντα** θα πρέπει να δημιουργεί ένα ζευγάρι κλειδιών **τοπικά** στον υπολογιστή του και στη συνέχεια να κάνει **upload** το δημόσιο. Με αυτό τον τρόπο είναι 100% σίγουρος οτι το ιδιωτικό του κλειδί δεν έχει **κατακρατηθεί!** Εμείς τα δημιουργούμε στη σελίδα για χάρη απλότητας και ευκολίας στο χρήστη και επειδή η πλατφόρμα είναι κατασκευασμένη ερευνητική χρήση!

Για τους αλγόριθμους κρυπτογράφησης της σελίδας χρησιμοποιήθηκε η phpseclib. Μια βιβλιοθήκη για php όπου παρέχει αλγόριθμους κρυπτογράφησης. Μπορούμε να την βρούμε στο http://phpseclib.sourceforge.net/ .

Συνολική εικόνα

Επομένως μέσω του μηχανισμού που περιγράφεται παραπάνω έχουμε τα παρακάτω χαρακτηριστικά:

- Tracker θυμάται τα δημόσια κλειδιά τόσο των BN όσο και των RN αλλά οχι τα ιδιωτικά
- Ο κάθε BN έχει το ζευγάρι των κλειδιών όπου αντιστοιχεί στο όνομα του
- Ο κάθε RN έχει το ζευγάρι των κλειδιών όπου αντιστοιχεί στο όνομα του και το ιδιωτικό του είναι προστατευμένο με password

Η πλατφόρμα τα έχει όλα μοιρασμένα προκειμένου να επικοινωνεί εμπιστευτικά

Μέσα στο πρόγραμμα

Μέσα στα προγράμματα του Tracker και του BN υπάρχουν αλγόριθμοι πιστοποίησης όπου ακολουθούν το παρακάτω μοντέλο προκειμένου να υπάρξει πιστοποίηση.

1. Συνδέεται ο client (BN ή RN)
2. Ο server (BN ή Tracker) βρίσκει το δημόσιο κλειδί του client βάση του ονόματος που έδωσε ο client
3. Ο server δημιουργεί ένα τυχαίο κλειδί **challenge** μεγάλου μεγέθους
4. **Το κρυπτογραφεί με το δημόσιο του client**
5. Το στέλνει στον client
6. **Ο client το αποκρυπτογραφεί με το ιδιωτικό του** και στέλνει πίσω το αποκρυπτογραφημένο μήνυμα
7. Ο server τα **συγκρίνει** και αν είναι ίδιο το challenge που φτιάχτηκε με αυτό που ήρθε, ο client πιστοποιείται επιτυχώς!

Γενικά το πρόγραμμα είναι γραμμένο σε java και η βιβλιοθήκη όπου χρησιμοποιείται είναι η **bouncy castle** https://www.bouncycastle.org/ και για αλγόριθμους κρυπτογράφησης αλλά και για βοηθητικές μαθηματικές συναρτήσεις.

Session key generate

```
41      public static String generateQuestion(int len) {
42          SecureRandom random = new SecureRandom();
43          return new BigInteger(len, random).toString(32);
44      }
```

Εικόνα 53 session key generate function

Μια πολύ απλή αλλά πολύ σημαντική συνάρτηση, μπορούμε να δώσουμε το συνολικό μέγεθος του κλειδιού και να μας παράγει το κλειδί συνεδρίας στο μέγεθος όπου επιθυμούμε.

Αν το τρέξουμε για διάφορα μήκη θα πάρουμε κάτι παρόμοιο όπως τα παρακάτω

512:

1bsh8nv8gk9nrqjdt8p2ehfpq80vjg7o3dsfhm3ta6uotr1mv0p8lmlnuu8s89n80d37kbnr4u0ve9ffga1lte54ej8m73tc0p622sr

1024:

f9gbooaft3r4aq2e6ertcm31ba4m001grek8tccsiothggbo1vh0tndcjmdldghc1anulktglcmj06ncv7tbt51mteloe5m8fa96idku58n9ca9sqaflhupnidgo71u0tn55126i1qn76uv1tho6kt3ufak0bh2deqqolia28hp3n6u07v2eb7odk08u8qqfh7gr0h38q4cvp

2048:

41ap1fphnb63nbpu8nnsh94bvnap3nsen81sss39ag9p70ek2gdqf9e2eq0f5mmrtegnmp805f66ij6u5m1fk8pogar95hlok3avcc17jrl39fa1pfnq65k2fufjvt3tgq4o6orpitajiag53uslpi5v4fmdavb8mhhfe8v7v2apkhf8qvpnvtkvbn8sa6pk73i3j3nfhfefqpn28k2e5b5peqbgauuv3ed0vi73k30ankenms020a46s37f809hemu0k7tghev2u1j8vilcld3spkcotjefarr78ho6o7m0tfe7iii52hgsvupfb7t5jq7225ol7nn85hu7tr16u7a98j75rf6p9l45ppmq38t6fq4te2m1s5jilulsdtrblafflvm9lfc24dka996fb353pv

Την ταυτοποίηση προτού την εγκαταστήσω στην πλατφόρμα την έφτιαξα αρχικά να γίνεται ολόκληρη στο ίδιο πρόγραμμα καθώς ένας προγραμματιστής, εάν διαχειρίζεται ένα μεγάλο project θα πρέπει να είναι σίγουρος για τις αλλαγές όπου κάνει! Το παρακάτω πρόγραμμα εξηγεί την παραπάνω λογική της ταυτοποίησης όπου περιγράψαμε και μάλιστα χρησιμοποιείται στην αρχή του προγράμματος για key validation. Ουσιαστικά ελέγχει αν ταιριάζει το δημόσιο με το ιδιωτικό κλειδί

```
    public static void main(String[] args) throws InvalidKeySpecException, IllegalBlockSizeException, NoSuchProviderException {

        String question = generateQuestion(1024);
        System.out.println(question);

        //get your public key
        String publicKey = "-----BEGIN PUBLIC KEY-----\n"
                + "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtk1cGKhieoVArJR6iY94\n"
                + "iABzPpFdVC3zy3mhHyJmR9m8qSnnrp+ip7n9Zy+ai5YIdd9agj+67hRjTeTGC/Ec\n"
                + "+x7zDDjnBBrlKYsyvdGfuKlBIrXgdl9uh4lnVkKHfECo9fnje6dVmZCK84qhUANB\n"
                + "AtRvkZFTByMQNxvBPOm/G8p6L6pOLBiiuhNuzHrSYG/AfGDTyIg4M7GAkU+uUYBv\n"
                + "e9TDY328vkzBASCdNOXC8vEi6vEIPk/yI2Hnlk07gofhx05cryhO6PbftPykdle+\n"
                + "DDTOxHvElW/MkQVX9rc0D0yJ9D2rj1Klm+lODQ8HaqX8fW5JSI6AMsWN2WFipmcJ\n"
                + "6QIDAQAB\n"
                + "-----END PUBLIC KEY-----";

        System.out.println(publicKey);
        PublicKey pubkey = getPublicKeyFromString(publicKey);

        byte[] chiperedQuestion = RSAAuthenticateChallenge(question, pubkey);

        //now we reached the other side!!!
        String privateKey = "-----BEGIN RSA PRIVATE KEY-----\n"
                + "MIIEowIBAAKCAQEAtk1cGKhieoVArJR6iY94iABzPpFdVC3zy3mhHyJmR9m8qSnn\n"
                + "rp+ip7n9Zy+ai5YIdd9agj+67hRjTeTGC/Ec+x7zDDjnBBrlKYsyvdGfuKlBIrXg\n"
                + "dl9uh4lnVkKHfECo9fnje6dVmZCK84qhUANBAtRvkZFTByMQNxvBPOm/G8p6L6pO\n"
                + "LBiiuhNuzHrSYG/AfGDTyIg4M7GAkU+uUYBve9TDY328vkzBASCdNOXC8vEi6vEI\n"
                + "Pk/yI2Hnlk07gofhx05cryhO6PbftPykdle+DDTOxHvElW/MkQVX9rc0D0yJ9D2r\n"
                + "j1K1m+lODQ8HaqX8fW5JSI6AMsWN2WFipmcJ6QIDAQABAoIBAQCogrulo0hcXn4Y\n"
                + "yKq4KyFL/baJWE8/t7ZKGGTh5adLtS3Z5H1fAfqVNavRzMP7UTUC1/HOweAloDzm\n"
                + "zJhwg3C5g7NAUfzg44d+rke6BGGyjOlDj4Erii0eJdmad6bLKO3FaTZon5XVfDGk\n"
                + "yzkvP8LBPeLfWMi5qSSc/A/UIXDg2uiYadFPswd+SdmaJgH59IEd2lfP+bHmSGBS\n"
                + "dD/YB9En9PTb4OSEIVfLL03ABwsxeV7IHv3hbcqZ31fhfhsdMhCkTjG7Tox1p0fD\n"
                + "UxnpfjLj5lsbYrJOIV8wUVoQQwkUzp/snMq0NFVoxZ2D5fkQ8KAS8PwNj4GXfcEl\n"
                + "M8U+cnP9AoGBAO0XkhpFT+THLFAMdoOFWW2/5pVHN3KuvmlZML3tueNqIVlKYNfc\n"
                + "EisKqLGTHtq6Sz8GTCQOXGsdMs+3U7HIdKe6snfEyrYO2JM/dIF/hITPUWSdFO56\n"
                + "cJ5ccdEDvFBhVu61bosoM/TONoDmfl3AI/xxwpANwExWkn9WdI1Z379/AoGBAMTX\n"
                + "NWI3DJijG7DL9fkp1rtTT2Y5EDnUGWmngR4nqNe0fFoXVKHBYbJasYHCZ+qsoRr8\n"
                + "+Tb2UwHKtHAbs2gmohZ4Tm78W7acIC1r8Ik+uU8PVkiBTdZZ7hkdg2mHt9k0R8v9\n"
                + "bWfkZGSmLpw1XttKfzlFKNXOnKlv4DWvgjMTHeqXAoGAAqMle+dTeS8B/i31T4c3\n"
                + "NHJTBUwSgNMSySc11JcFX1M55b1fEGehSBtJPxhs2nACEERoqmoCeyqK+yaF5s9d\n"
                + "BNSd0Zk9zAKkRBcLm7koZzXLKPxaVEDGaeyLU5DgEmDSz7ry7NdYpJt6nbpyo2ZU\n"
                + "wCUfzexpPDAmVwZGK6BZTc8CgYACLjgjLGTxU+08miXRass8LAIXKc6qNVVKvFZL\n"
                + "1TijmxY9kUCYwiGo7iRFQbgQ+3SVbfP8zeHBhVNWYpgsMTFeelq0FAuYDEa2+hki\n"
                + "DBXVcGAOUZBhLYHbuV35T02UFGYvNlF98yPBka22gUjZuQuLwN5g7/cAUYL0VUtl\n"

                + "8XJFZQKBgBIeSj9okLRLQmGaWlgEJRhlFeM1oI5OKfhG01mX/9q1Nk8r/42mXHhL\n"
                + "QYC+GgMG4+lhzsP82WeSbv08aYgA5lx01JnJrA9aDx/OZGqMvLIyUmL8gbyJO7m2\n"
                + "zVEej4v/Jf/pKn1DiigFQphTuq8q2CsJtzxWD9FPVNQy+zrIkR2K\n"
                + "-----END RSA PRIVATE KEY-----";

        //make your private key usable
        PrivateKey privkey = getPrivateKeyFromString(privateKey);

        //decrypt the question
        String answer = new String(RSAAuthenticateResponce(chiperedQuestion, privkey));

        //THE MOMENT WE ALL WAITED FOR
        System.out.println(answer);
        System.out.println(question);
        if (answer.equals(question)) {
            System.out.println("Match!!!!!!!!!");
        }
    }
}
```

Εικόνα 54 Public & Private key Validate algorithm

Φυσικά μέσα στο πρόγραμμα αυτό το κομμάτι εκτελείται και στις 2 πλευρές client και server και επικοινωνεί με την βάση δεδομένων για να κατεβάσει το δημόσιο κλειδί.

Σημείωση:

Οποιοσδήποτε προγραμματιστής επιθυμεί να ασχοληθεί με κρυπτογράφηση πάνω από TCP συνδέσεις θα πρέπει να γνωρίζει οτι οι αλγόριθμοι κρυπτογράφησης προκειμένου να εκτελεστούν μπορούν να πάρουν ωφέλιμο χρόνο μέχρι και 4ή 5 δευτερόλεπτα! Αυτό το χρονικό διάστημα για ένα κανάλι TCP θεωρείται αρκετά μεγάλο και το κανάλι στην άλλη πλευρά θα κλείσει θεωρώντας οτι έγινε κάποιο λάθος. Επομένως θα πρέπει να ορίσουμε ένα socket timeout που πιστεύουμε οτι είναι αρκετό για την διεκπεραίωση των αλγορίθμων στο κανάλι.

socket.setSoTimeout(8000);

5.3 θέματα ασφάλειας της ιστοσελίδας της registry και της βάσης δεδομένων

Σε αυτή την ενότητα θα εξηγήσουμε μερικά θέματα ασφάλειας της κεντρικής ιστοσελίδας του registry και θα πάρουμε μια εικόνα της βάσης δεδομένων.

Σε τι μορφή μια ασφαλής σελίδα θα πρέπει να αποθηκεύει τα password στη βάση δεδομένων

- Μία ιστοσελίδα δεν πρέπει ποτέ να σώζει στη βάση δεδομένων ένα password στην plain του μορφή
- Θα πρέπει να σχηματίζει την ψηφιακή του σύνοψη έτσι ώστε και αν κλαπούν οι εγγραφές της βάσης δεδομένων ο επιτιθέμενος να έχει μόνο την ψηφιακή σύνοψη στα χέρια του και όχι το password!
- Ο αλγόριθμος της ψηφιακής σύνοψης δεν θα πρέπει να θεωρείται legacy θα πρέπει να είναι όσο πιο σύγχρονος γίνεται. Εμείς στη σελίδα χρησιμοποιούμε SHA-256.
- Επίσης η μετατροπή password σε hash θα πρέπει να έχει και ένα επιπλέον χαρακτηριστικό ασφάλειας, το salt
- Το salt προστίθεται στο τέλος του password έτσι ώστε κάποιος επιτιθέμενος μη μπορώντας να βρεί πιο είναι το salt να αδυνατεί να υπολογίσει τις ψηφιακές συνόψεις ή τουλάχιστον να μην τις βρει από βάση δεδομένων εύκολα

Όταν ένας χρήστης συνδέεται στη σελίδα και δίνει το password του, η λογική της σελίδας το ενώνει με το salt, παράγει την σύνοψη, και στη συνέχεια **συγκρίνει** τη σύνοψη της βάσης δεδομένων με την παράγωγη σύνοψη. Αν είναι ίδιες δίνει στο χρήστη ένα κουλουράκι και τον αφήνει να περάσει!

Στην περίπτωση όπου κάποιος επιτιθέμενος καταφέρει να έχει την db στα χέρια του θα παρατηρήσει κάτι σαν τις παρακάτω εγγραφές όπου του είναι άχρηστες για να εισχωρήσει στο σύστημα!

id	username	password	name
27	pakhs	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e...	pakhs
26	nikos	26bea56ec31a482b5bd1cc115db707efbc3606ec4c87a98612...	nikos
25	kostis	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e...	kostis

Εικόνα 55 user entries on database

Το Κουλουράκι

Εικόνα 56 A cookie!

Ξεκινώντας αυτό το κομμάτι θα πρέπει να πούμε οτι υπάρχουν πάρα πολλά attacks με cookies και σε δημόσια wi-fi είναι πολύ εύκολα. Ο κυριότερος λόγος είναι οτι πολλοί υπολογιστές βγαίνουν από το NAT με κοινή δημόσια IP όπως εξηγήθηκε και παραπάνω. Αυτό σημαίνει οτι είναι αρκετά εύκολο ο ένας host να υποδυθεί τον άλλο βάση της IP. Επίσης εάν τα κουλουράκια δεν έχουν **ημερομηνία λήξης** ένας επιτιθέμενος μπορεί να τα συλλέξει και να τα χρησιμοποιεί όποτε επιθυμεί (replay attacks). Επίσης ένα cookie θα πρέπει να έχε πέρα από την ημερομηνία λήξης και πληροφορίες από το user-agent του browser του χρήστη. Αυτό προσθέτει επιπλέον ασφάλεια καθώς ο επιτιθέμενος θα πρέπει να γνωρίζει επακριβώς το user-agent του χρήστη για μια επιτυχημένη επίθεση και να τον εξομοιώνει σε δικό του browser (session hijack). Τέλος ποτέ δεν θα πρέπει το cookie να περιέχει το password του account μέσα του καθώς άμα βρεθεί η γίνει decrypt να μην ανακτηθεί. Θα πρέπει να είναι ένα κλειδί συνεδρίας όπου να μην έχει αξία πληροφορίας και να λήγει και σε συγκεκριμένο χρόνο !

Cookie factory

Παρόμοια λογική με τα password. Σε αυτή την περίπτωση πρέπει να συλλέξουμε πληροφορίες όπως η ώρα, ο UA, ο χρήστης, να παράγουμε την ψηφιακή σύνοψη τους όπου θα είναι το body του cookie.

Όταν ο χρήστης χρησιμοποιεί το cookie θα πρέπει να υπολογίζονται οι παραπάνω παράγοντες και να συγκρίνονται με το cookie αν είναι κοινές οι συνόψεις ο χρήστης έχει δικαιώματα αλλιώς το cookie μηδενίζεται

Ημερομηνία λήξης

Το Unity δεν είναι μια πλατφόρμα ακόμα απόλυτα ασφαλής καθώς κάτι τέτοιο θα απαιτούσε πολλά άτομα να δουλεύουν ασταμάτητα για κενά ασφάλειας. Ο ρόλος του registry είναι να δείξει τη λειτουργία του κεντρικού μητρώου και γι' αυτό το λόγο το μόνο μέτρο ασφάλειας όπου λάβαμε από τα παραπάνω στο cookie είναι η ημερομηνία λήξης! Εάν κάποιος επιθυμούσε να το ανεβάσει στο internet θα πρέπει να χρησιμοποιήσει τη σελίδα ως **παράδειγμα.**

Το να κάνεις ημερομηνία σε cookie στην php είναι απλό

```
$today = getdate();
$timestamp = $today['hours']."_".$today['mday']."_".$today['mon']."_".$today['year'];
```

Το $timestamp μπορούμε να δούμε οτι θα αλλάξει τιμή στην επόμενη ώρα καθώς τίποτα άλλο δεν αλλάζει πέρα του $today['hours']. Στη συνέχεια έπειτα από μια πιστοποίηση θα δημιουργηθεί το cookie.

```
if (hash('sha256',$password) == $retpassword) {
    $isadmin = 1;
    $username = $_REQUEST['rusername'];
    $password = $_REQUEST['rpassword'];
    $cookievalue = hash('sha256', $username.$salt.hash('sha256',$password).$timestamp);
    setcookie('user', $cookievalue, time() + 3600);
  } else {
    $info = "Wrong username or password";
  }
```

Από τον παραπάνω κώδικα μπορούμε να δούμε οτι το password φυλάσσεται μέσα στο hash του cookie το οποίο δεν είναι ασφαλές!

Σε ένα άλλο κομμάτι του κώδικα, στην αποσύνδεση, πρέπει να σβήσουμε το cookie

```
} else if (isset($_REQUEST['logout'])) {
        setcookie('user', 'null', time() - 3600);
        $isadmin = 0;
 }
```

Man against the machines

Τέλος ένα άλλο θέμα είναι οτι προκειμένου να έχουμε μια ασφαλή σελίδα εγγραφών πρέπει να έχουμε προστασία εναντίων bot. Τα bot είναι προγράμματα τα οποία μπορούν να συνδεθούν σε μια σελίδα και να δημιουργήσουν π.χ 1000 account με πλασματικά στοιχεία. Για να μη συμβεί κάτι τέτοιο στη σελίδα όπου γίνεται register το account θα πρέπει να υπάρχει και μια ερώτηση όπου μόνο ένας άνθρωπος θα μπορούσε να απαντήσει. Είτε μπορεί να είναι ένας αλγόριθμος captcha είτε μια ερώτηση όπως:

Εικόνα 57 A bot!

Bob had three apples and then he got another one more! How many apples Bob had?

Ένας άνθρωπος μπορεί πολύ απλά να γράψει 4 και να πιστοποιηθεί ως άνθρωπος, ένα bot δεν μπορεί!

5.4 Η αρχιτεκτονική της βάσης δεδομένων

Τέλος από άποψη κατανομής καλό θα ήταν να ρίξουμε μια ματιά στη βάση δεδομένων για να δούμε την αρχιτεκτονική της

users:

Id	Username	Password	Name

hostnames:

id	hostname	Public Key	userid

bluenodes:

id	name	Public Key	userid

Η βάση δεδομένων είναι απλή και λειτουργική! Φυλάει τα δημόσια κλειδιά τόσο των BN όσο και των RN. Τα hostnames και τα bluenodes έχουν το userid το οποίο είναι reference στο id του user και αυτό γιατί μπορεί να έχει όσα hostname ή BN θέλει.

Όλα τα πεδία id καθώς και τα πεδία userneme, hostname, name(bluenodes) είναι unique για να μην υπάρξει ποτέ διπλός user ή διπλό hostname.

Οι BN θεωρούνται του συστήματος αλλά προκειμένου η βάση να μην κρατάει 2 είδη χρηστών προτιμήθηκε να μπορεί να τους δημιουργεί ένας χρήστης.

Τέλος κάποιος πονηρεμένος θα αναρωτηθεί **Που φυλάσσονται οι εικονικές IP διευθύνσεις δεν τις βλέπω πουθενά;**

Η απάντηση σε αυτή την ερώτηση είναι οτι το να φυλάμε τις virtual IP είναι πλεονασμός!

Αυτό συμβαίνει καθώς κάθε hostname έχει ένα id το οποίο τυχαίνει να είναι αύξοντας αριθμός. Έτσι αν είχαμε ένα δίκτυο τύπου 10.0.0.0/8 ο 1ος θα έπαιρνε την 10.0.0.1 , ο 2ος την 10.0.0.2, ο 3ος την 10.0.0.3 κοκ μέχρι να τελειώσουν οι διευθύνσεις!

Την παραπάνω τεχνική την καταφέρνουμε με 2 συναρτήσεις όπου έγραψα όπου μετατρέπουν ένα ακέραιο σε μία εικονική διεύθυνση IP. Τις συναρτήσεις την βλέπουμε παρακάτω.

```
17
18      public static String numberTo10ipAddr(String vaddress) {
19          byte[] networkpart = new byte[]{0x0a};
20          int hostnum = Integer.parseInt(vaddress) + 1;
21
22          byte[] hostpart = new byte[]{
23              (byte) ((hostnum) >>> 16),
24              (byte) ((hostnum) >>> 8),
25              (byte) (hostnum)};
26
27          byte[] address = new byte[4];
28          System.arraycopy(networkpart, 0, address, 0, networkpart.length);
29          System.arraycopy(hostpart, 0, address, 1, hostpart.length);
30          try {
31              return InetAddress.getByAddress(address).getHostAddress();
32          } catch (UnknownHostException ex) {
33              Logger.getLogger(VAddressFunctions.class.getName()).log(Level.SEVERE, null, ex);
34              return null;
35          }
36      }
37
38      public static String _10ipAddrToNumber(String vaddress) {
39          InetAddress addr = null;
40          try {
41              addr = InetAddress.getByName(vaddress);
42          } catch (UnknownHostException ex) {
43              Logger.getLogger(VAddressFunctions.class.getName()).log(Level.SEVERE, null, ex);
44          }
            byte[] address = addr.getAddress();
46          byte[] hostpart = new byte[3];
47          System.arraycopy(address, 1, hostpart, 0, 3);
48          int hostnum = 0;
49          for (int i = 0; i < hostpart.length; i++) {
50              hostnum = (hostnum << 8) + (hostpart[i] & 0xff);
51          }
52          hostnum = hostnum - 1;
53          return "" + hostnum;
54      }
55  }
56
```

Για να τις δούμε στην πράξη:

```
55
56      public static void main(String[] args) {
57          System.out.println(numberTo10ipAddr("1"));
58          System.out.println(numberTo10ipAddr("2"));
59          System.out.println(numberTo10ipAddr("300"));
60          System.out.println(numberTo10ipAddr("501"));
61      }
62  }
63
```

VAddressFunctions > _10ipAddrToNumber >

Output - lvl5UnityTracker (run)

```
run:
10.0.0.2
10.0.0.3
10.0.1.45
10.0.1.246
BUILD SUCCESSFUL (total time: 0 seconds)
```

θα παρατηρήσουμε οτι η αρίθμηση ξεκινάει από το 2 γιατι η 10.0.0.1 έγινε η GW για το δίκτυο.

Πάμε να δούμε και την αντιστροφή!

```
55
56      public static void main(String[] args) {
57          System.out.println(numberTo10ipAddr("1"));
58          System.out.println(numberTo10ipAddr("2"));
59          System.out.println(numberTo10ipAddr("300"));
60          System.out.println(numberTo10ipAddr("501"));
61
62          System.out.println(_10ipAddrToNumber("10.0.0.2"));
63          System.out.println(_10ipAddrToNumber("10.0.0.3"));
64          System.out.println(_10ipAddrToNumber("10.0.1.45"));
65          System.out.println(_10ipAddrToNumber("10.0.1.246"));
66      }
67  }
68
```

```
Output - lvl5UnityTracker (run)
run:
10.0.0.2
10.0.0.3
10.0.1.45
10.0.1.246
1
2
300
501
BUILD SUCCESSFUL (total time: 0 seconds)
```

Εικόνα 58 IP to Int and reverse functions & test

6. Επίλογος για τους συμφοιτητές/πληροφορικούς/προγρμματιστές

Για το έργο

Το έργο, δηλαδή η πλατφόρμα, έχει ανέβει στο Sourceforge κάτω από την Apache v2.0 License στο παρακάτω URL

https://sourceforge.net/projects/unitynetwork/

Συνοπτικά ηapace v2.0 άδεια επιτρέπει την αντιγραφή ή τροποποίηση μέρους ή όλου του κώδικα του project και σε άλλες υλοποιήσεις εμπορικές ή μή (GNU ή Academic ή Closed source). Απαραίτητη υποχρέωση εκείνου όπου ασκεί το δικαίωμα της άδειας είναι να μην αφαιρέσει τις κεφαλίδες από τις κλάσεις όπως και να συμπεριλάβει το NOTICE.txt στο παράγωγο project (Ουσιαστικά κάνοντας αναφορά στο αρχικό project).

Για τους συμφοιτητές/πληροφορικούς/προγρμματιστές

Σε αυτό το κομμάτι θέλω λίγο να αναλύσω τον τρόπο σκέψης και δράσης ενός σύγχρονου πληροφορικού πιστεύοντας οτι η **μεθοδολογία σκέψης** και δράσης μπορεί να βοηθήσει αρκετά τους συμφοιτητές μου και μελλοντικούς πληροφορικούς.

Αρχικά πρέπει να παρατηρήσω οτι έχει εξαπλωθεί ένας φόβος σχετικά με τη "δυσκολία" του προγραμματισμού! Οι σύγχρονοι πληροφορικοί θα πρέπει να σταματήσουν να φοβούνται να κατασκευάζουν έργα πληροφορικής και να καταλάβουν οτι η πληροφορική δεν είναι copy paste και οτι ο σωστός προγραμματιστής δεν ράβει κομμάτια κώδικα. Αντιθέτως ένας καλός προγραμματιστής ξεκινάει από την αρχή (από το μηδέν), χτίζει με οργάνωση αυτό όπου θέλει να καταφέρει και ζητάει βοήθεια μόνο σε πολύ μικρά και εξειδικευμένα κομμάτια κώδικα τα οποία δεν γνωρίζει. Ο ίδιος κανόνας ισχύει και στο διάβασμα. Ένας σωστός developer θα πρέπει να έχει διαβάσει και 2,3 βιβλία ή άρθρα και να μην googlarei την τελευταία στιγμή για να βρει μια πληροφορία και να κάνει κάτι στα γρήγορα. Εάν κάποιος χτίσει γερές βάσεις στη συνέχεια γίνεται όλο και πιο εύκολη η κατασκευή έργων και αισθάνεται και ο ίδιος αυτοπεποίθηση για αυτό που κάνει και αυτό που είναι!

Επίσης ένας κομπιουτεράς δεν πρέπει να φοβάται να εισχωρεί σε low level όταν οι συνθήκες το απαιτούν. Το low level, αν και πολλοί το παίρνουν από φόβο είναι "λογικό"! Αυτό σημαίνει οτι ορίζεται από αυστηρούς κανόνες όπου μένουν σταθεροί και θα τα βρει πολύ εύκολα μαζί του εάν ξεθαρρέψει και ασχοληθεί να μάθει τους κανόνες όπου το διέπουν. Μόλις τους μάθει θα δει οτι δεν ήταν ουσιαστικά τίποτα για φόβο και οτι μπορεί να κινηθεί πολύ εύκολα και μάλιστα έχει ικανότητες ως άτομο όπου δεν είχε προηγουμένως!

Ακόμα η πλήρη εξειδίκευση δεν είναι πάντα δημιουργική. Για παράδειγμα στο συγκεκριμένο project έπρεπε να γίνει χρήση και php, mysql, καθώς και dhcp, arp και άριστες γνώσεις δικτύου όπως και java, threads, data structs και RSA, AES. Πολλές φορές αν και έχουμε μια εξειδίκευση θα πρέπει να έχουμε και την γενική εικόνα σε ένα project ειδικά άμα εργαζόμαστε με άλλους και είμαστε σε μία θέση όπως project leader/manager!

Τέλος θέλω να υποστηρίξω οτι προκειμένου ένας πληροφορικός να είναι επιτυχημένος θα πρέπει να είναι σε θέση να διαβάζει άψογα αγγλικά και μάλιστα να αγοράζει βιβλία πληροφορικής κυρίως στα αγγλικά και να τους δίνει χρόνο και αφοσίωση στη μελέτη. Εκ πείρας πρέπει να δηλώσω οτι τα καλύτερα βιβλία είναι εκείνα που προσφέρουν περιβάλλον δοκιμών και σταδιακής εκμάθησης και θα πρέπει να προτιμούνται από "βιβλία εγκυκλοπαίδειες" όπου απλά απαριθμούν χαρακτηριστικά χωρίς στόχο εκμάθησης.

Βιβλιογραφία

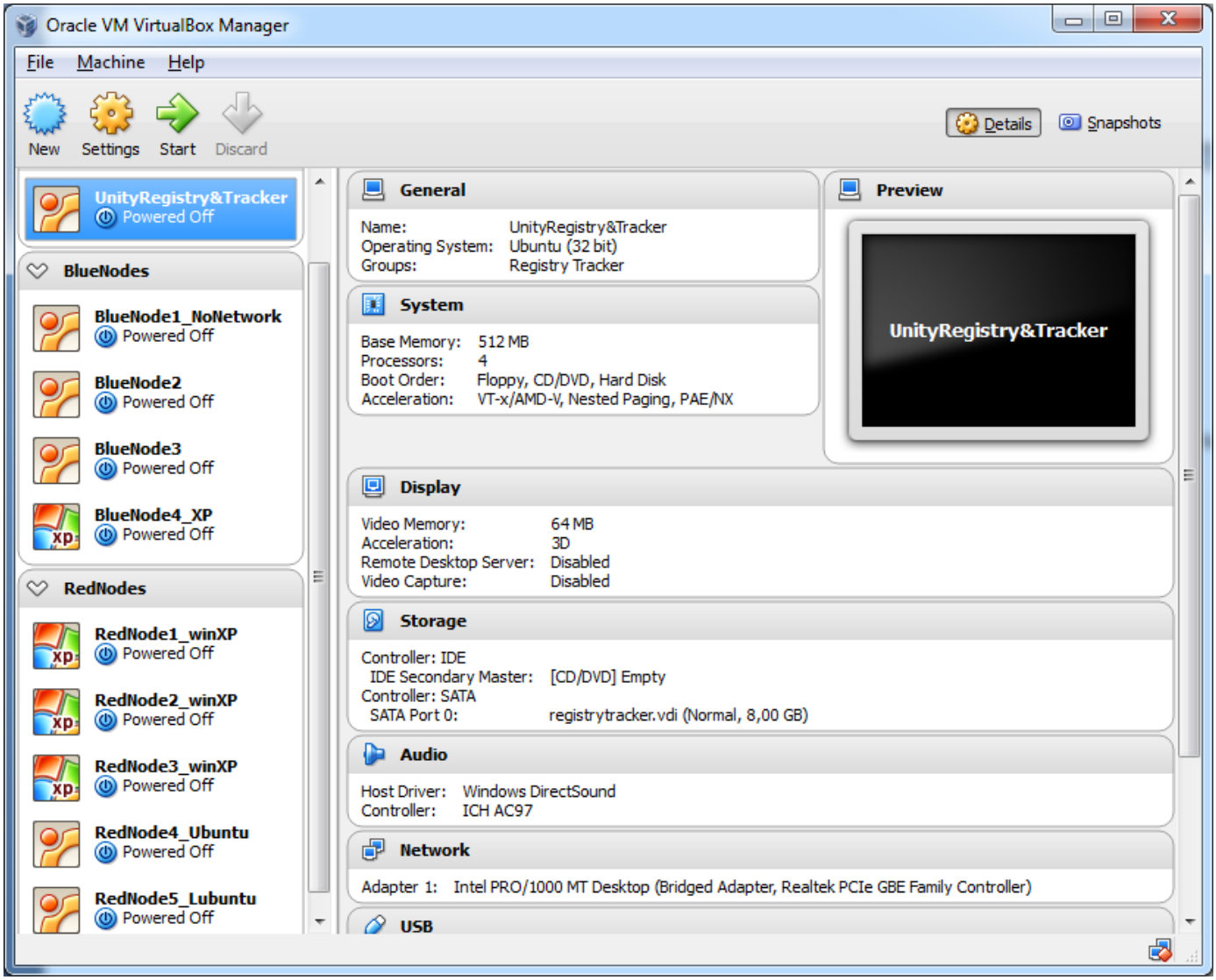
[1] IPv6.com NAT-IN-DEPTH Available: http://ipv6.com/articles/nat/NAT-In-Depth.htm
[2] UseIPv6.com Why not just use Network Address Translation (NAT)? Available: http://www.useipv6.com/
[3] OpenVPN (TUN/TAP adaptor is part of OpenVPN)Available: https://openvpn.net/
[4] P2PVPN project Available: http://www.p2pvpn.org/
[5] Phpseclib Available: http://phpseclib.sourceforge.net/
[6] Bouncy castle Available: https://www.bouncycastle.org/
[7] PhpMyAdmin Available: http://www.phpmyadmin.net/home_page/index.php
[8] RFC 2685 Available: http://www.rfc-editor.org/info/rfc2685
[9] Wireshark Available: http://www.wireshark.org/
[10]VirualBox Available: https://www.virtualbox.org/
[11] Nmap utilities Available: http://nmap.org/

ΠΑΡΑΡΤΗΜΑ

Παράδειγμα χρήσης της πλατφόρμας

Σε αυτό το τμήμα θα παρουσιαστεί με εικόνες η πλατφόρμα σε λειτουργία.

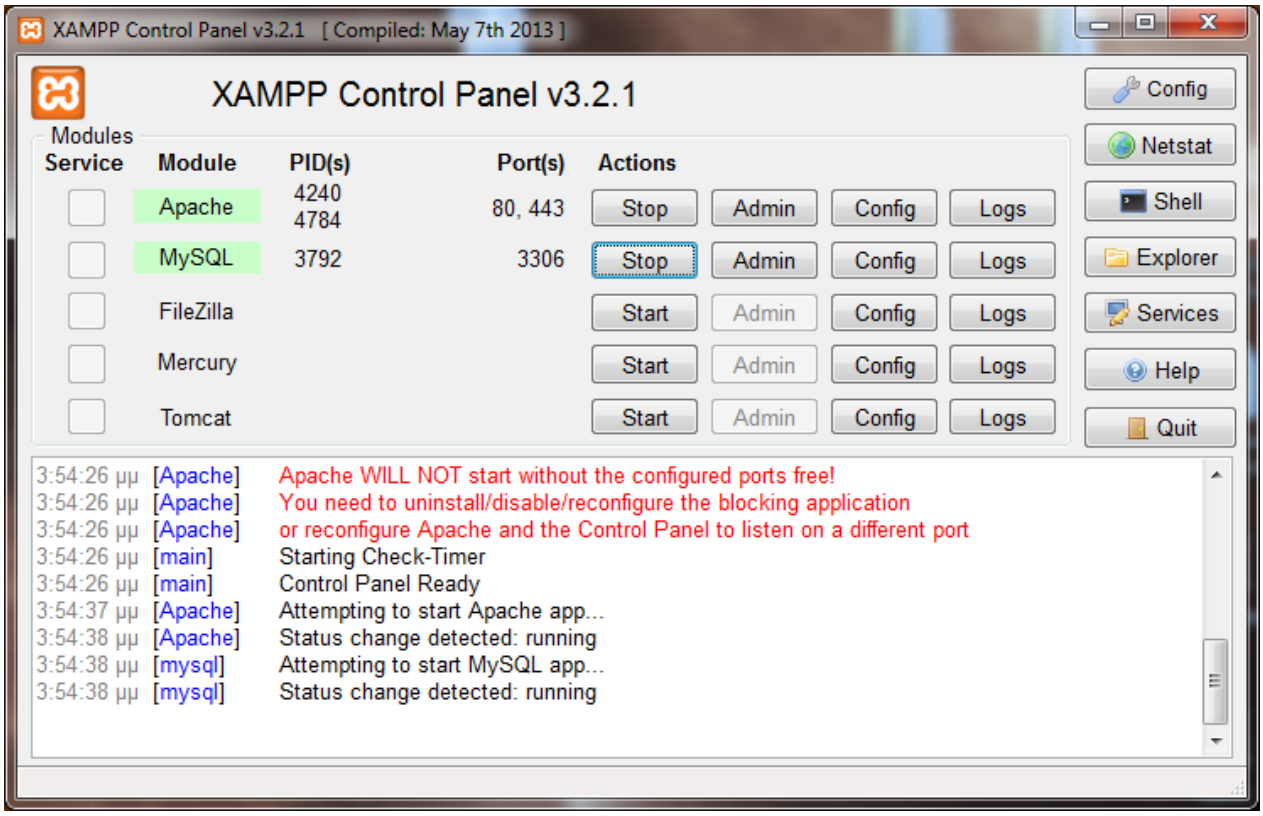
- Έχουμε στη βάση δεδομένων δηλωμένους μερικούς χρήστες όπου ο καθένας έχει hostnames και BN names.
- Έχουμε κατεβάσει τα κλειδιά στους στα αντίστοιχα μηχανήματα στα dirs των RN και BN.
- Την πλατφόρμα θα την στήσουμε με VMs τα οποία είναι bridged και το καθένα έχει και άλλη IP



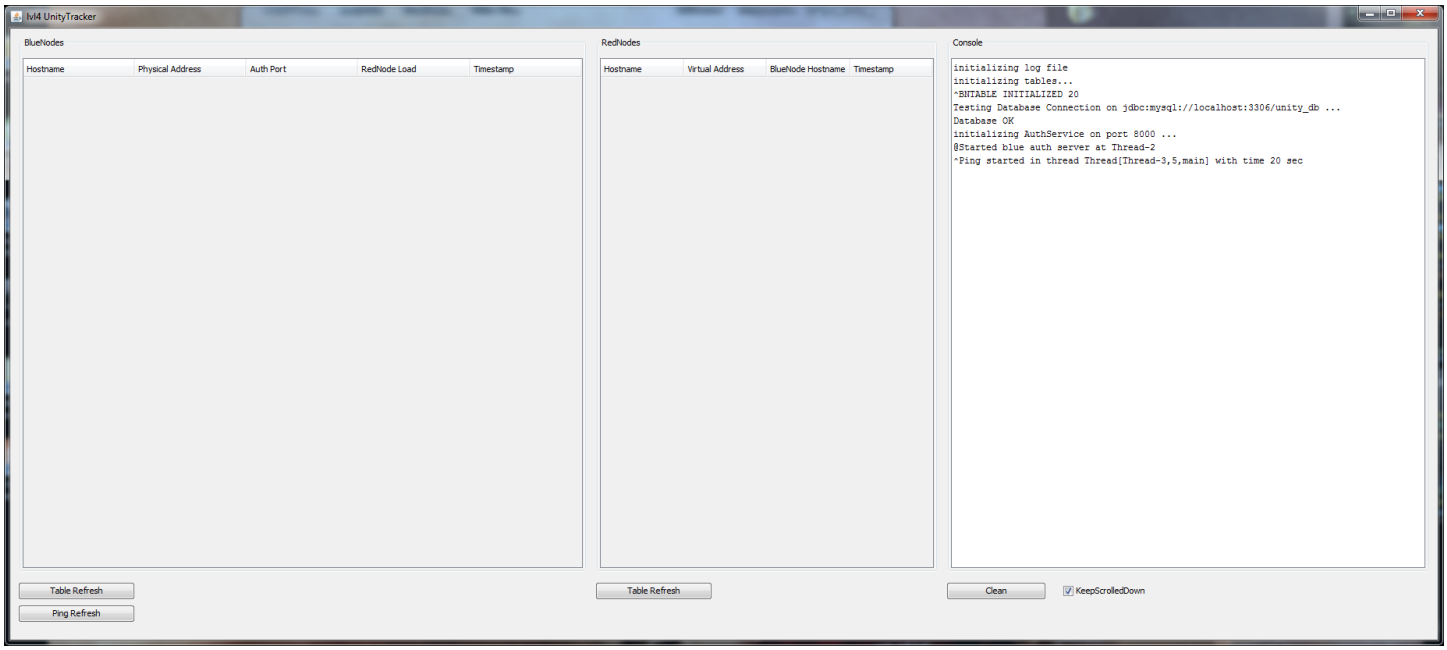
Ο tracker θα υπάρχει στο μηχάνημα όπου κάνει host τα VMs καθώς και αυτό έχει IP address στο LAN και μπορούμε να το δούμε καλύτερα.

Εκκίνηση του tracker

Αρχικά εκκινούμε από το XAMPP apache & mysql



πάμε στο dir του tracker και ελέγχουμε το tracker.conf αν είναι OK και ανοίγουμε τον Tracker. Στη συνέχεια περιμένουμε το OK από την DB.



θα παρατηρήσουμε οτι ο Tracker είναι κενός. Ο Tracker έχει εκκινήθει και περιμένει για τουλάχιστον ένα BN προκειμένου να δουλέψει η πλατφόρμα.

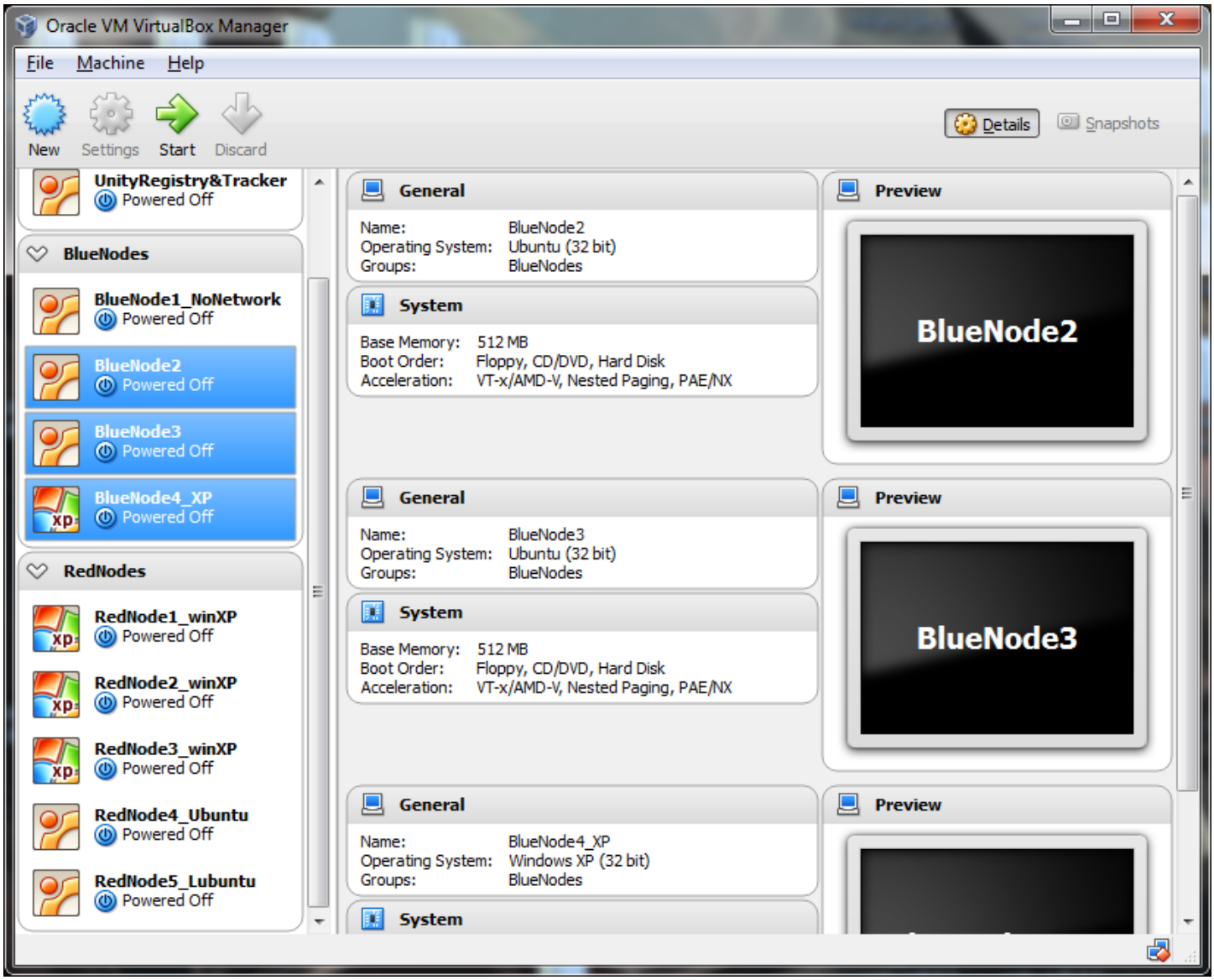
Η IP του Tracker είναι η:

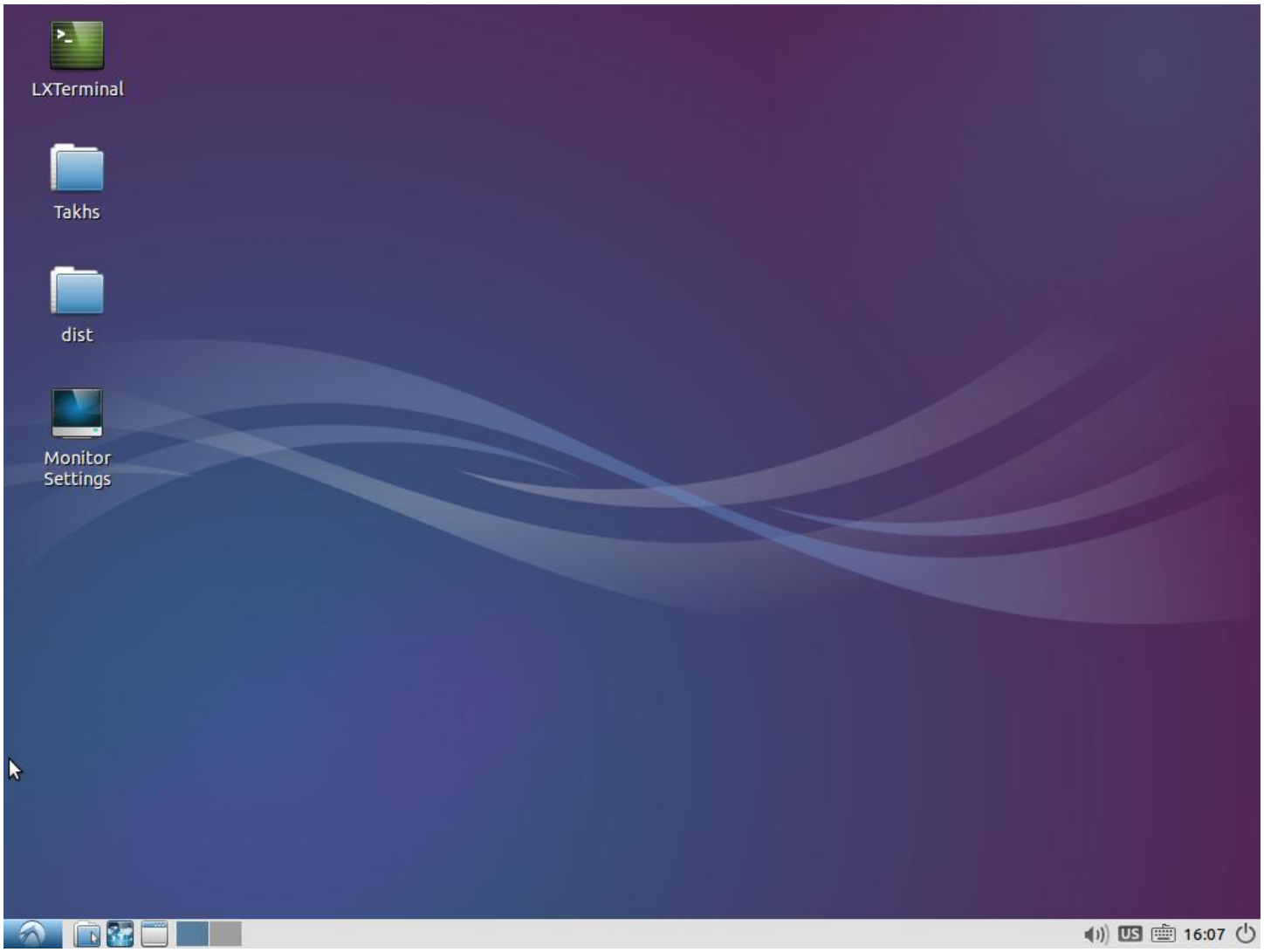
```
Επίθημα DNS συγκεκριμένης σύνδεσης:
Διεύθυνση IPv6 τοπικής σύνδεσης . : fe80::7070:5080:455:d959%13
Διεύθυνση IPv4. . . . . . . . . . : 192.168.2.8
Μάσκα υποδικτύου. . . . . . . . . : 255.255.255.0
Προεπιλεγμένη πύλη . . . . . . . . : 192.168.2.1
```

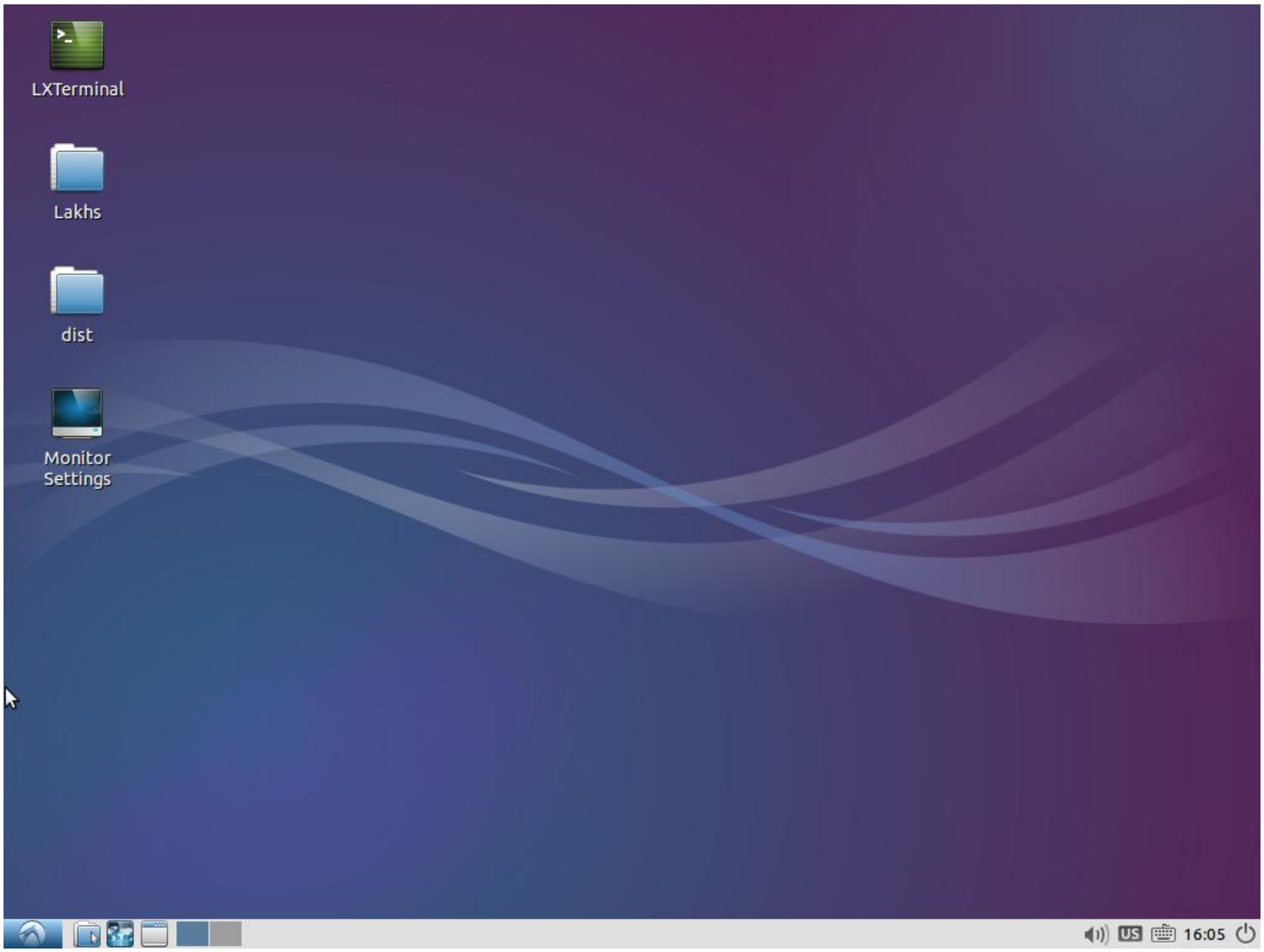
Εκκίνηση των BlueNode

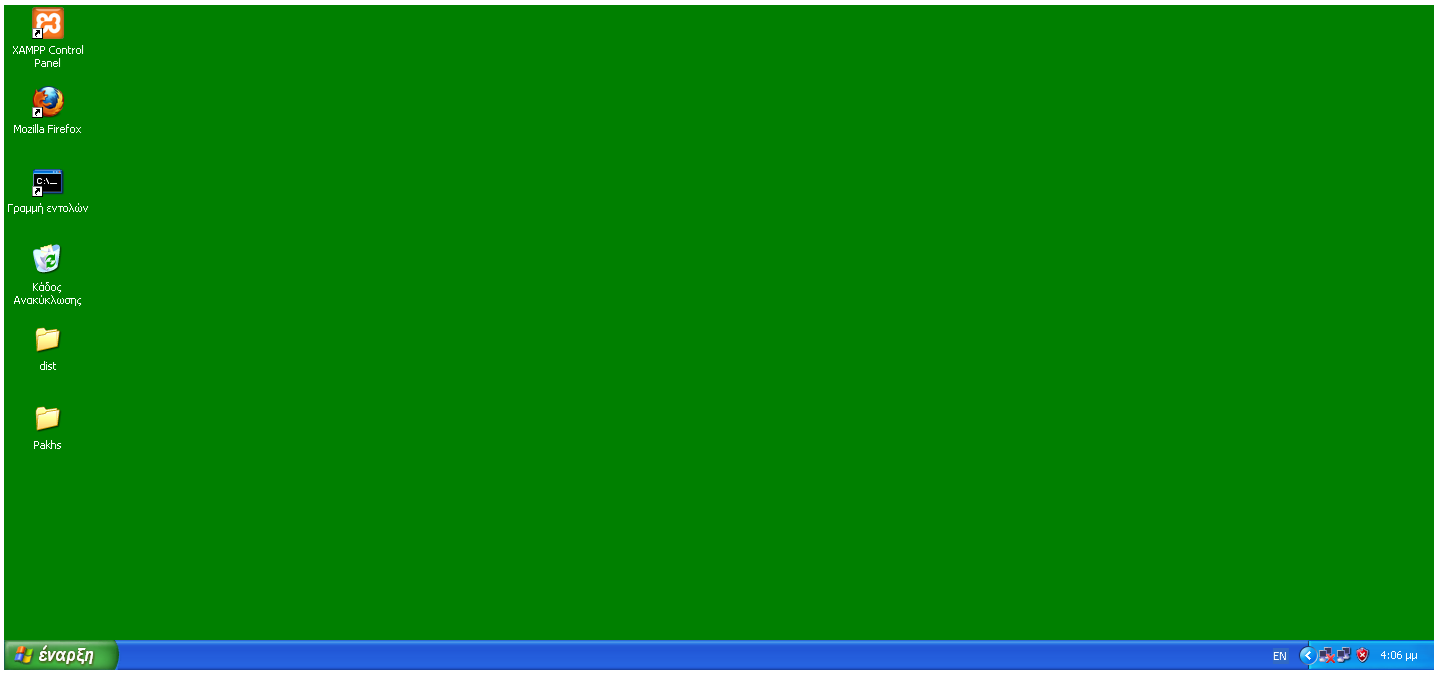
Στην παρούσα φάση θα χρησιμοποιήσουμε τρείς BN για να έχει ο καθένας και από άλλους RN και να δείξουμε την κατανομή καλύτερα.

Επίσης για να δείξουμε και τη μεταφερσιμότητα των προγραμμάτων από το ένα OS στο άλλο και την ανομοιογένεια την οποία μπορεί να υποστηρίξει το project. Ο ένας είναι windows XP και οι άλλοι 2 LUBUNTU Linux. Τους ξεκινάμε αρχικά από το περιβάλλον του virualbox.



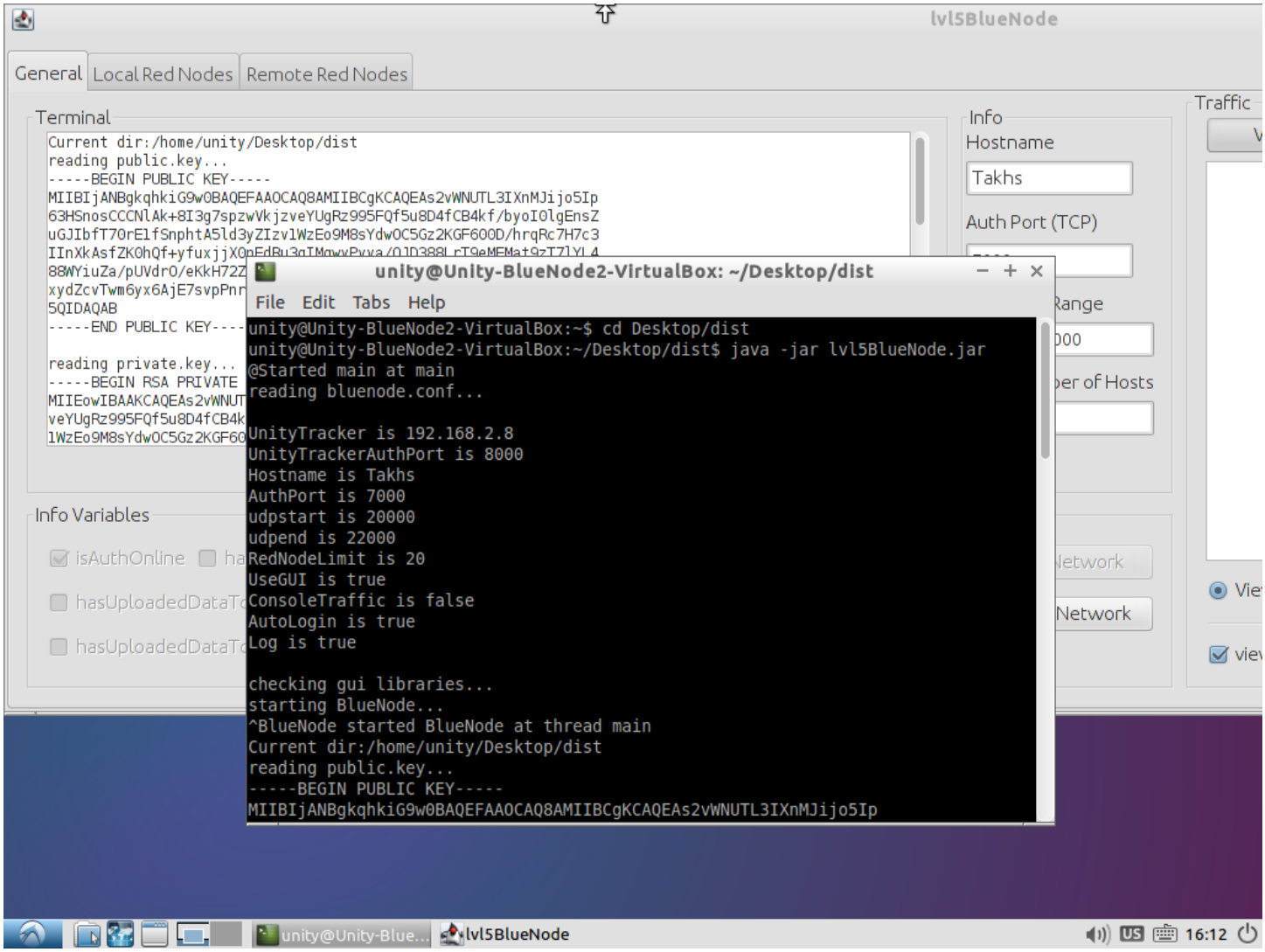

LXTerminal
Takhs
dist
Monitor Settings
US
16:07

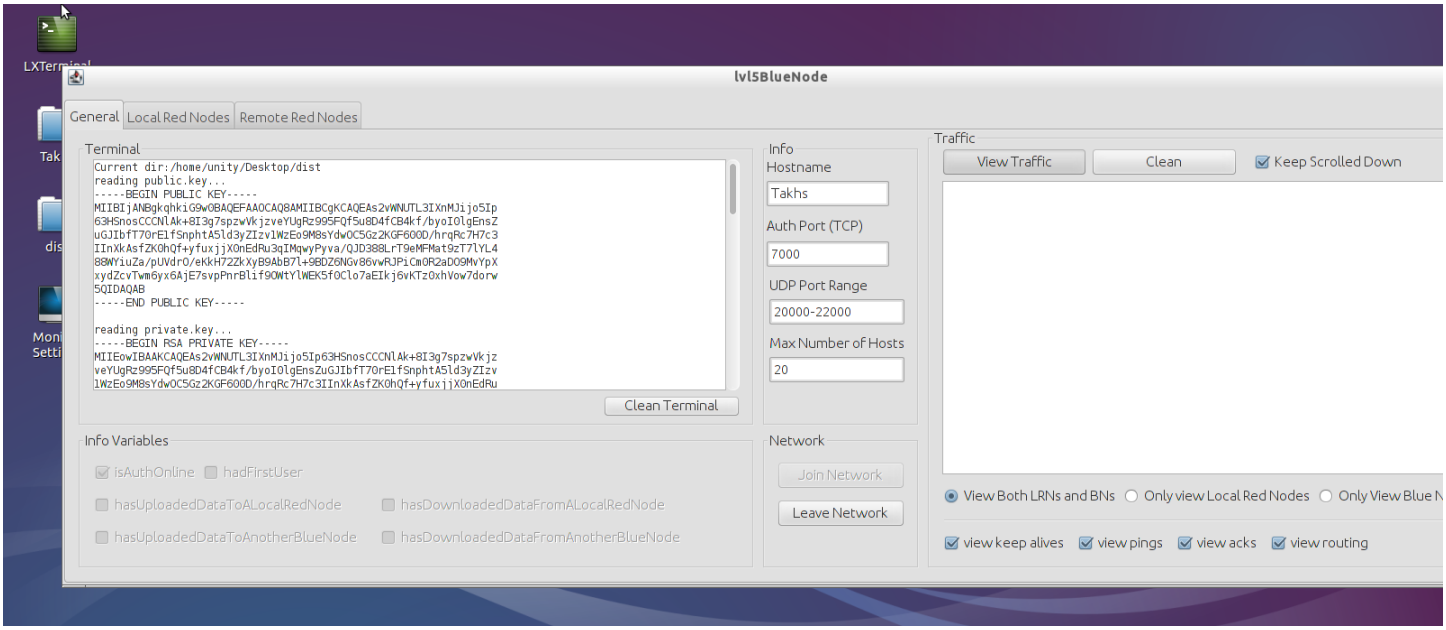

LXTerminal
Lakhs
dist
Monitor Settings
US
16:05


XAMPP Control Panel
Mozilla Firefox
Γραμμή εντολών
Κάδος Ανακύκλωσης
dist
Pakhs
έναρξη
EN
4:06 μμ

Οι 3 αυτοί hosts έχουν πιστοποιηθεί στο δίκτυο με ένα όνομα ο καθένας, έχουν ένα ζευγάρι κλειδιών ο καθένας το οποίο αντιστοιχεί σε συγκεκριμένο name και έχουν τροποποιήσει το bluenode.conf ώστε να δηλώνεται το όνομα τους και η διεύθυνση του tracker!

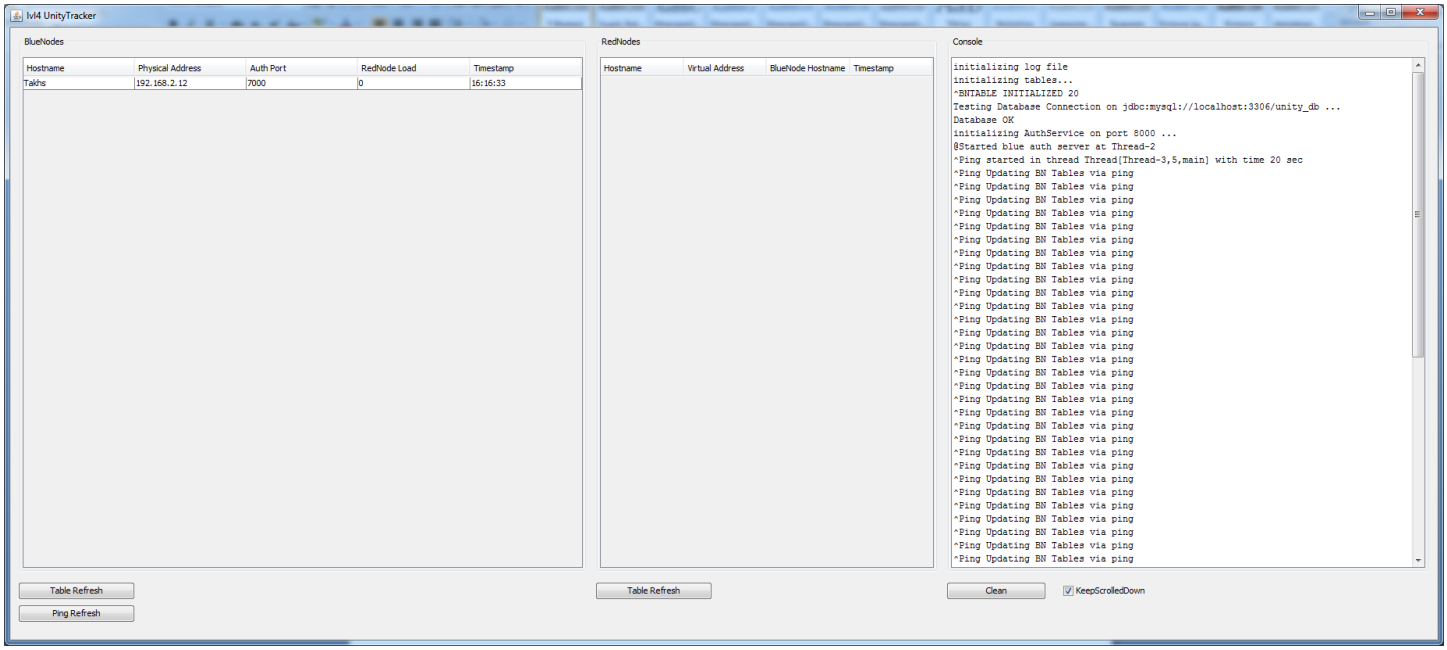
Μόλις εκκινήσουμε τους BN θα κάνουν αυτόματη πιστοποίηση στο δίκτυο και θα ενημερωθεί το GUI του tracker!





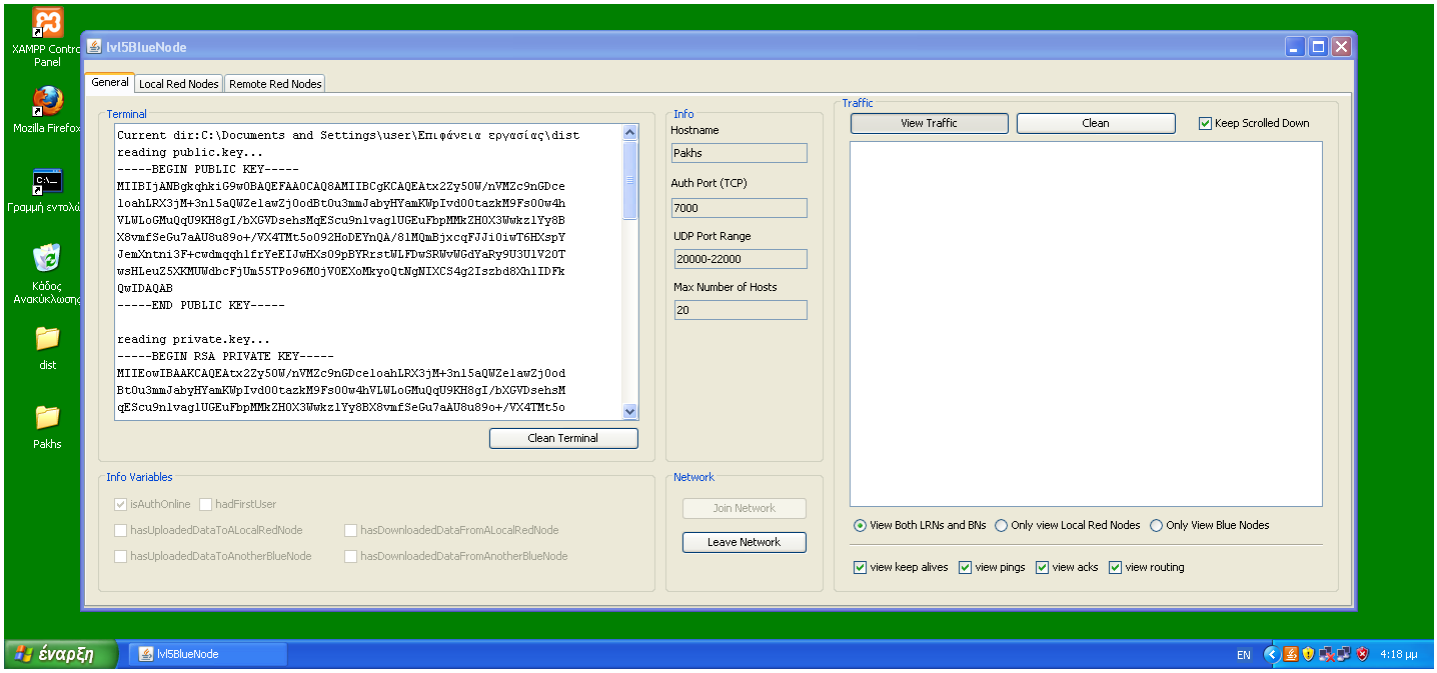
Ο BN Takhs έχει IP 192.168.2.12

Μόλις συνδεθεί ο tracker έχει ανανεωθεί και πλέον έχει ένα BN!

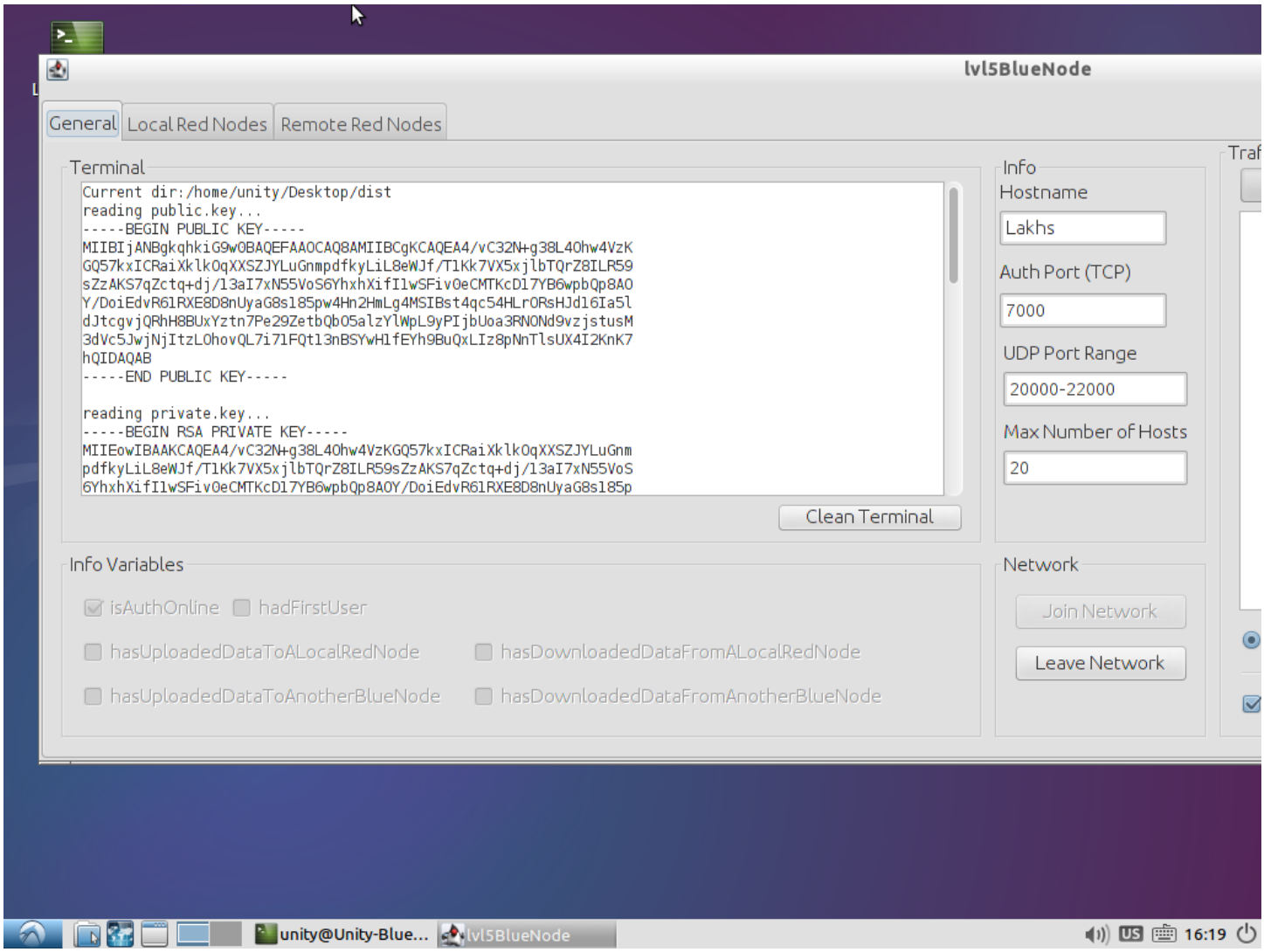


Βλέπουμε ακόμα οτι ο BN μας έχει μηδενικό load ακόμα καθώς δεν υπάρχουν RNs συνδεμένοι.

Κάνουμε το ίδιο και για τους άλλους 2 BNs

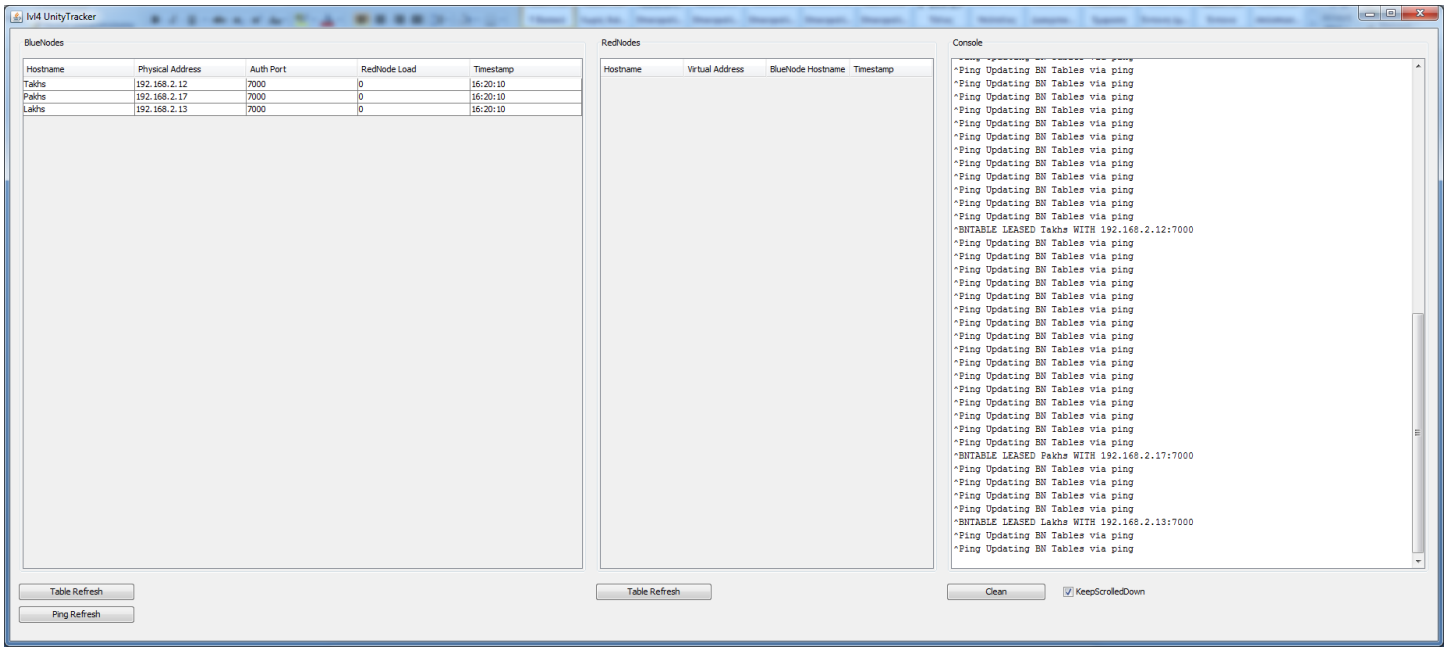


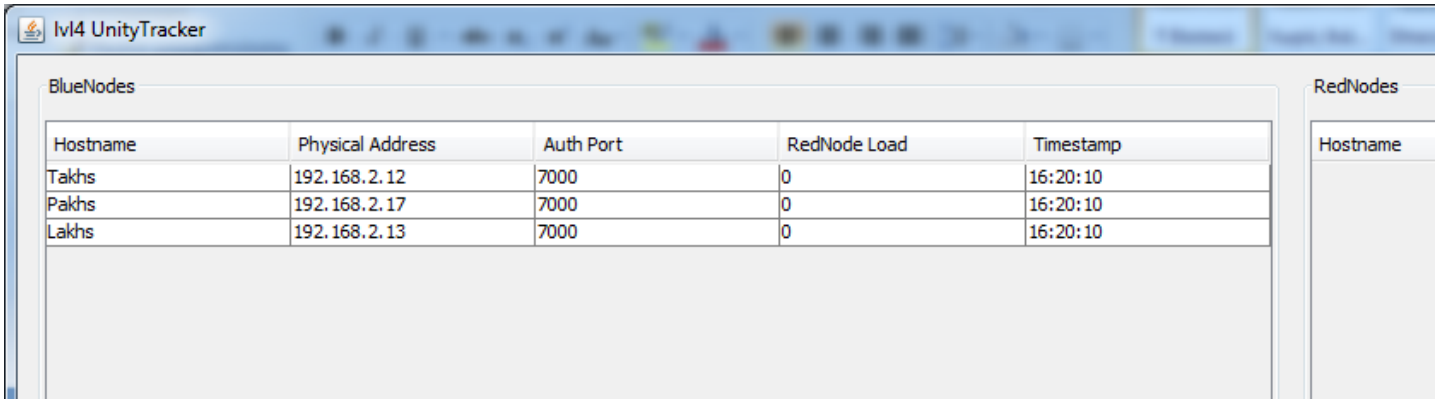
BN Pakhs, ip: 192.168.2.17



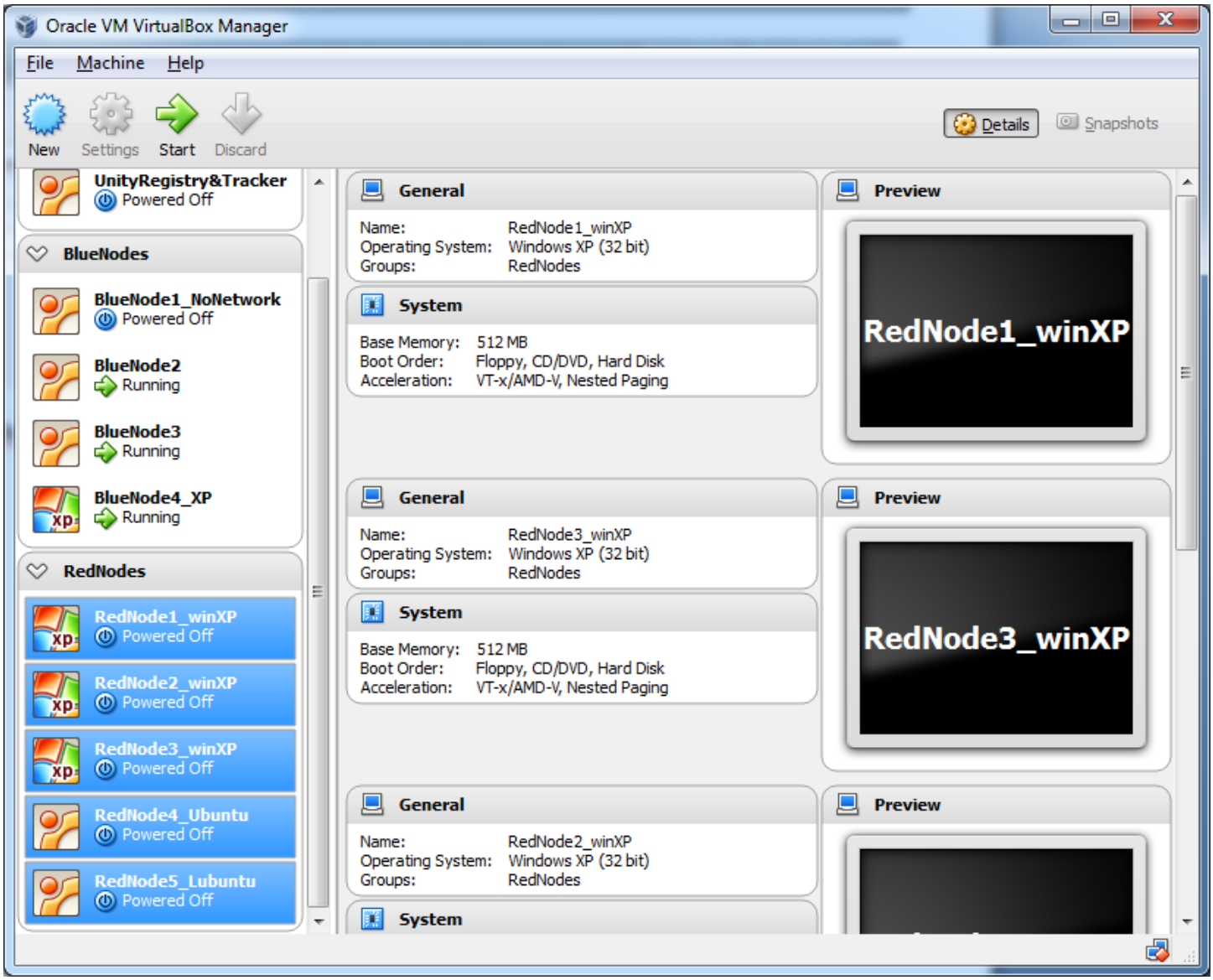
BN Lakhs ip: 192.168.2.13

Ο tracker έχει ενημερωθεί για τους BN



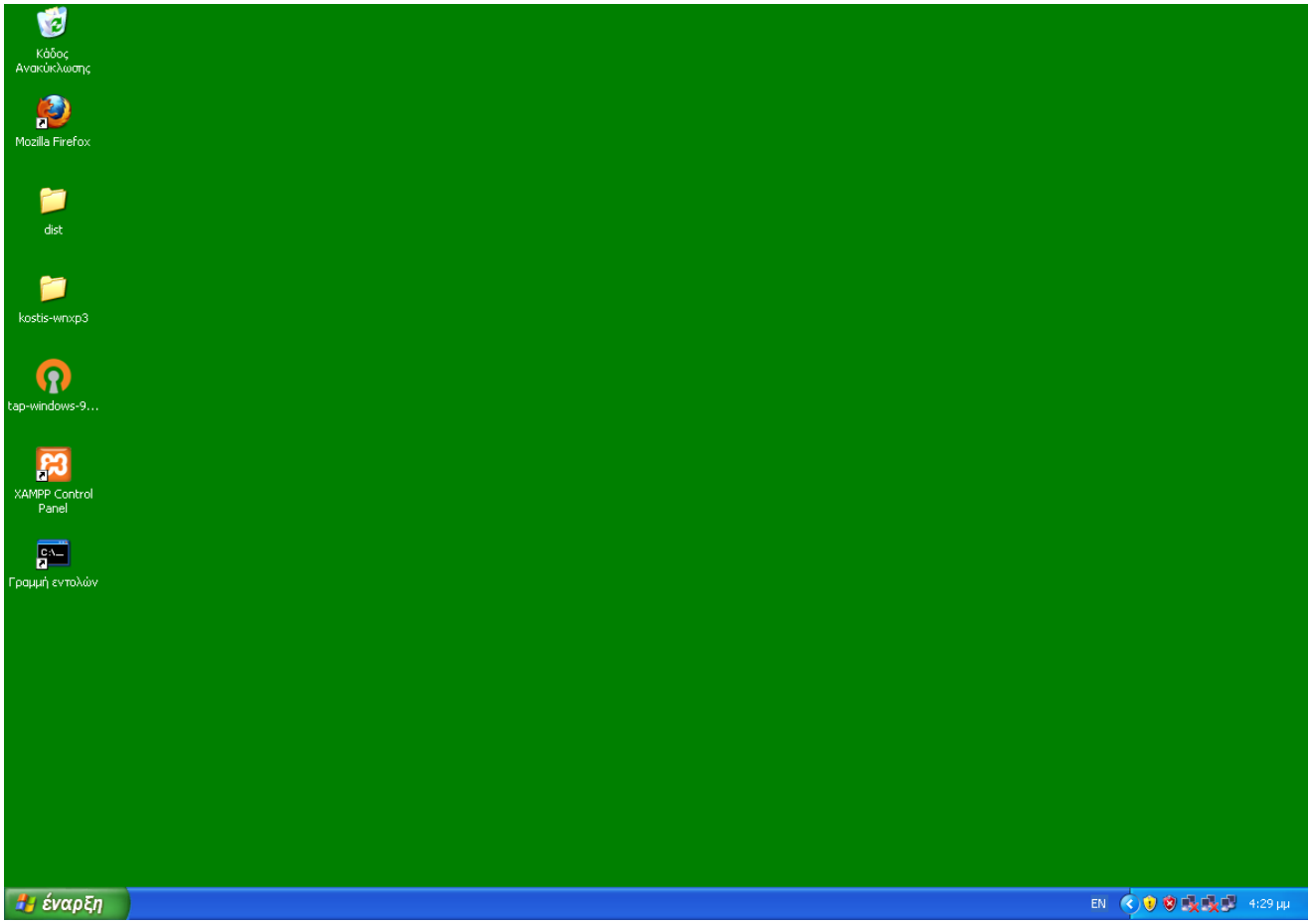


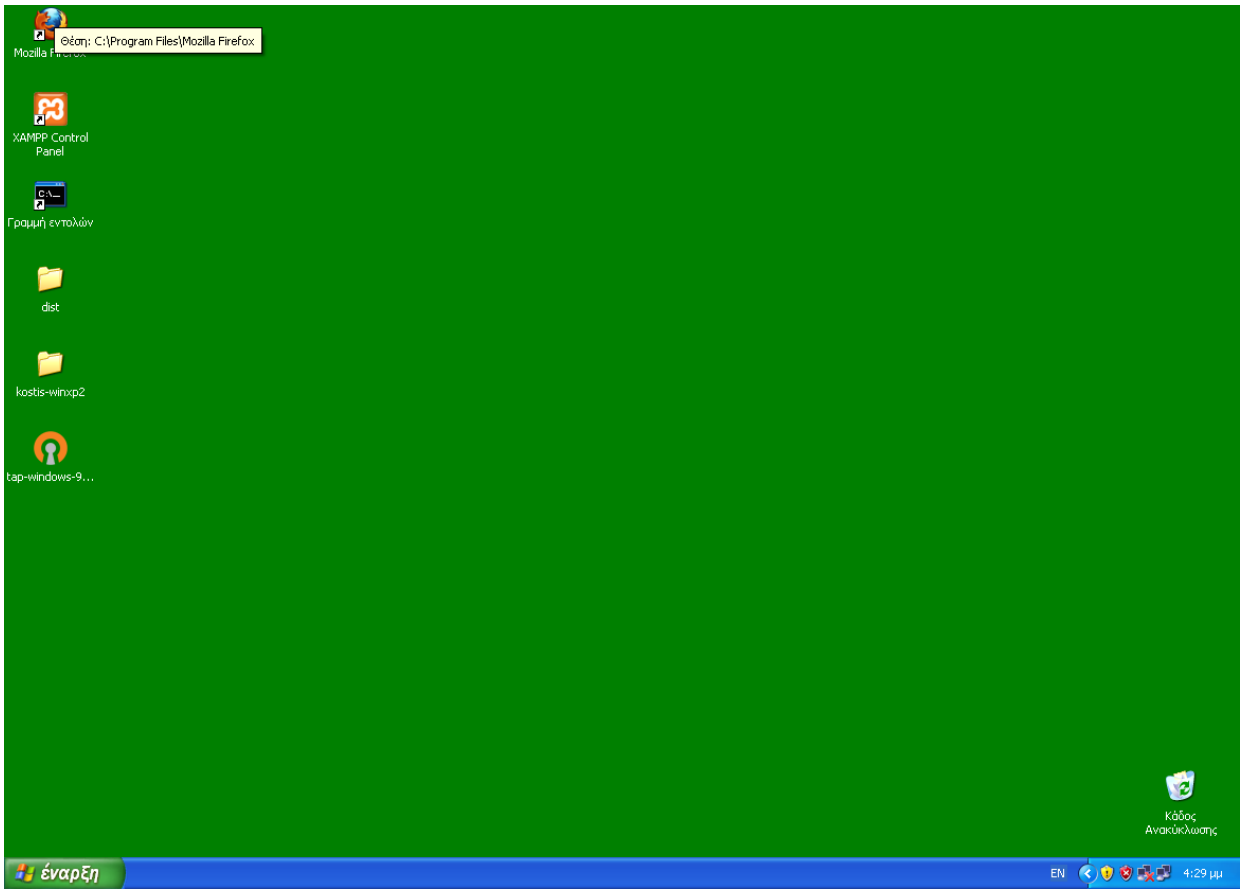
Πλέον μένει να συνδεθούν οι RedNode για να επικοινωνήσουν και να ανταλάξουν δεδομένα μέσω του εικονικού δικτύου.

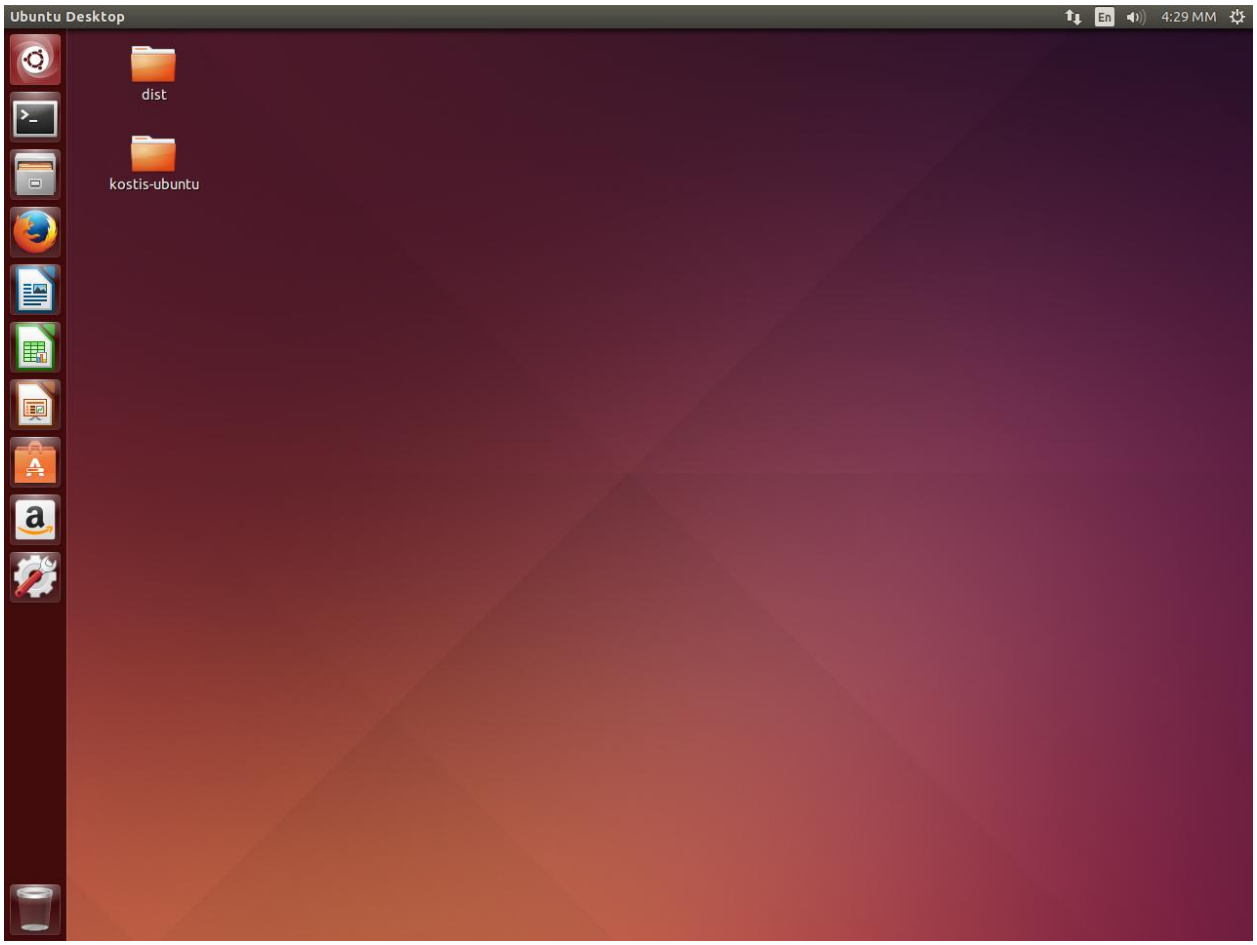


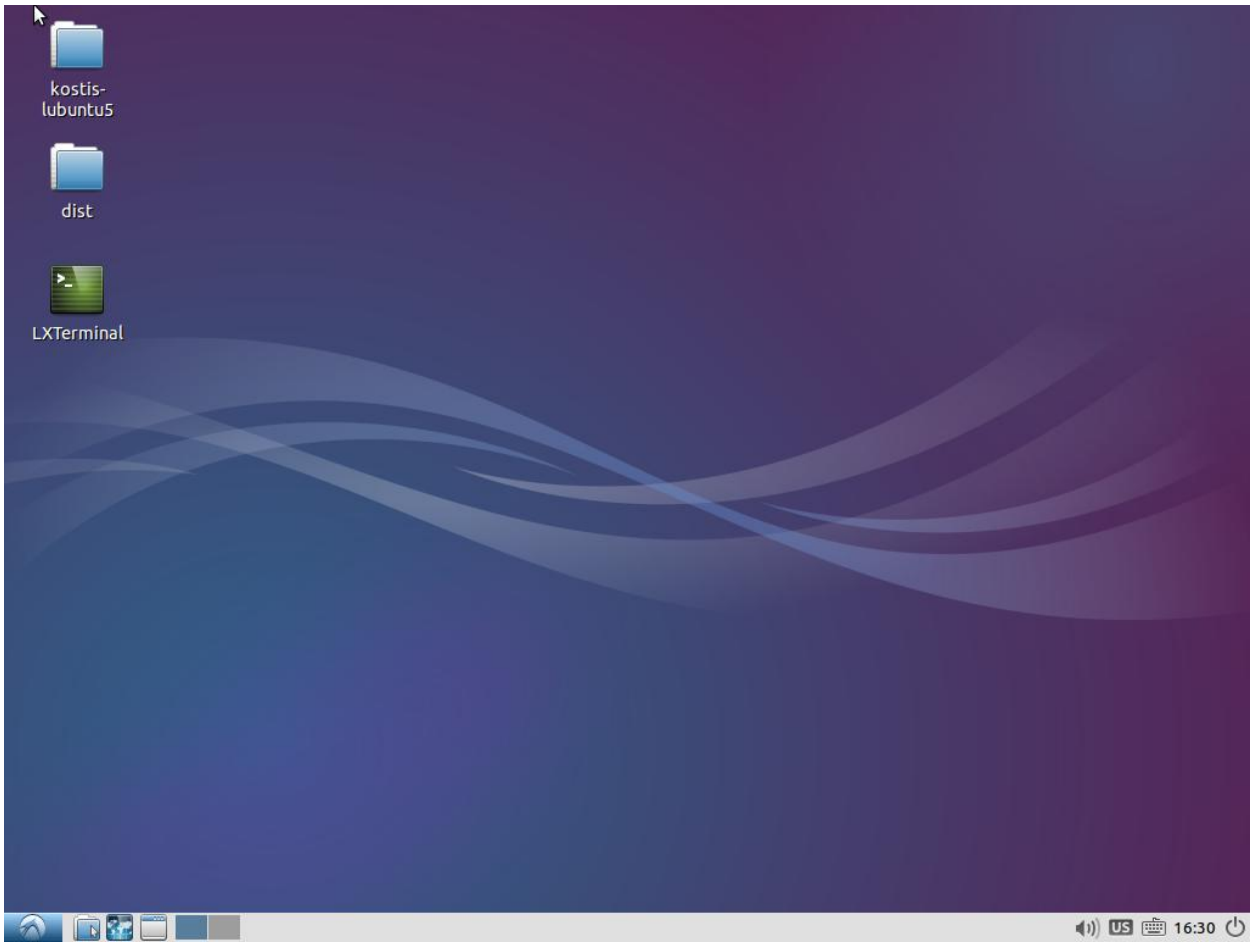
Αντίστοιχα όπως και πριν ένας RN μπορεί να υπάρξει σε πολλά είδη OS. Εδώ υπάρχουν και πελάτες με windows xp, lubuntu αλλά και για hi-end λειτουργικά όπως Ubuntu.


Mozilla Firefox
XAMPP Control Panel
Γραμμή εντολών
tap-windows-9...
dist
kostis-winxp1
Κάδος Ανακύκλωσης
έναρξη
EN
4:29 μμ


Κάδος Ανακύκλωσης
Mozilla Firefox
dist
kostis-winxp3
tap-windows-9...
XAMPP Control Panel
Γραμμή εντολών
έναρξη
EN
4:29 μμ


Θέση: C:\Program Files\Mozilla Firefox
XAMPP Control Panel
Γραμμή εντολών
dist
kostis-winxp2
tap-windows-9...
Κάδος Ανακύκλωσης
έναρξη
EN
4:29 μμ


Ubuntu Desktop
En
4:29 MM
dist
kostis-ubuntu

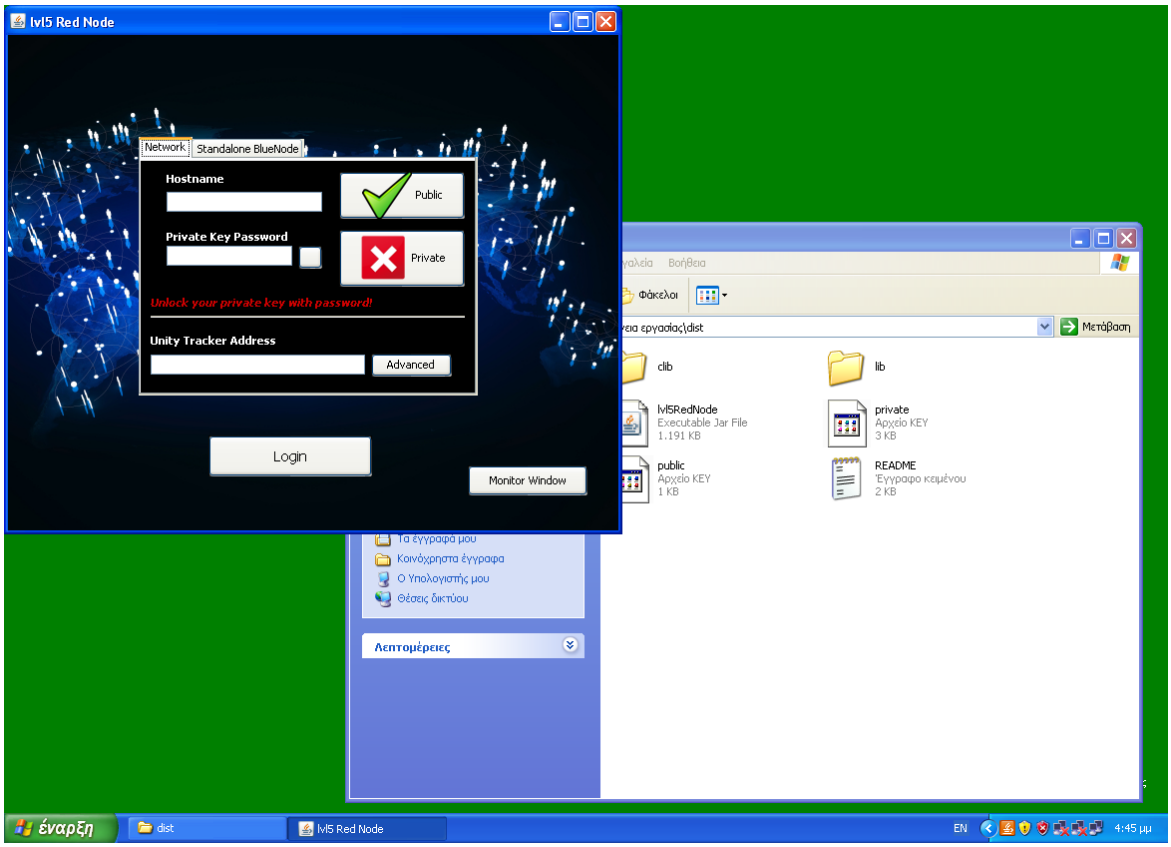


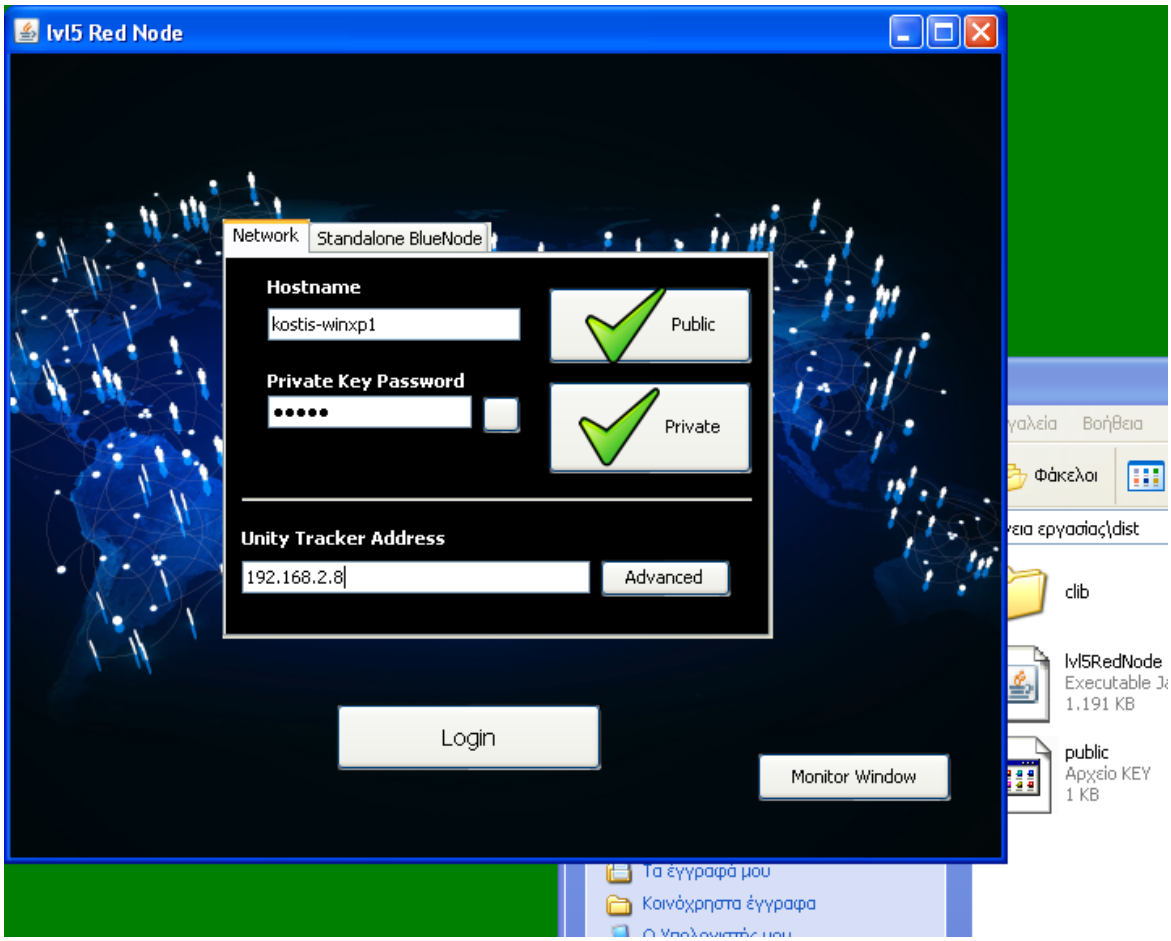
Τα στοιχεία των RN φαίνονται παρακάτω

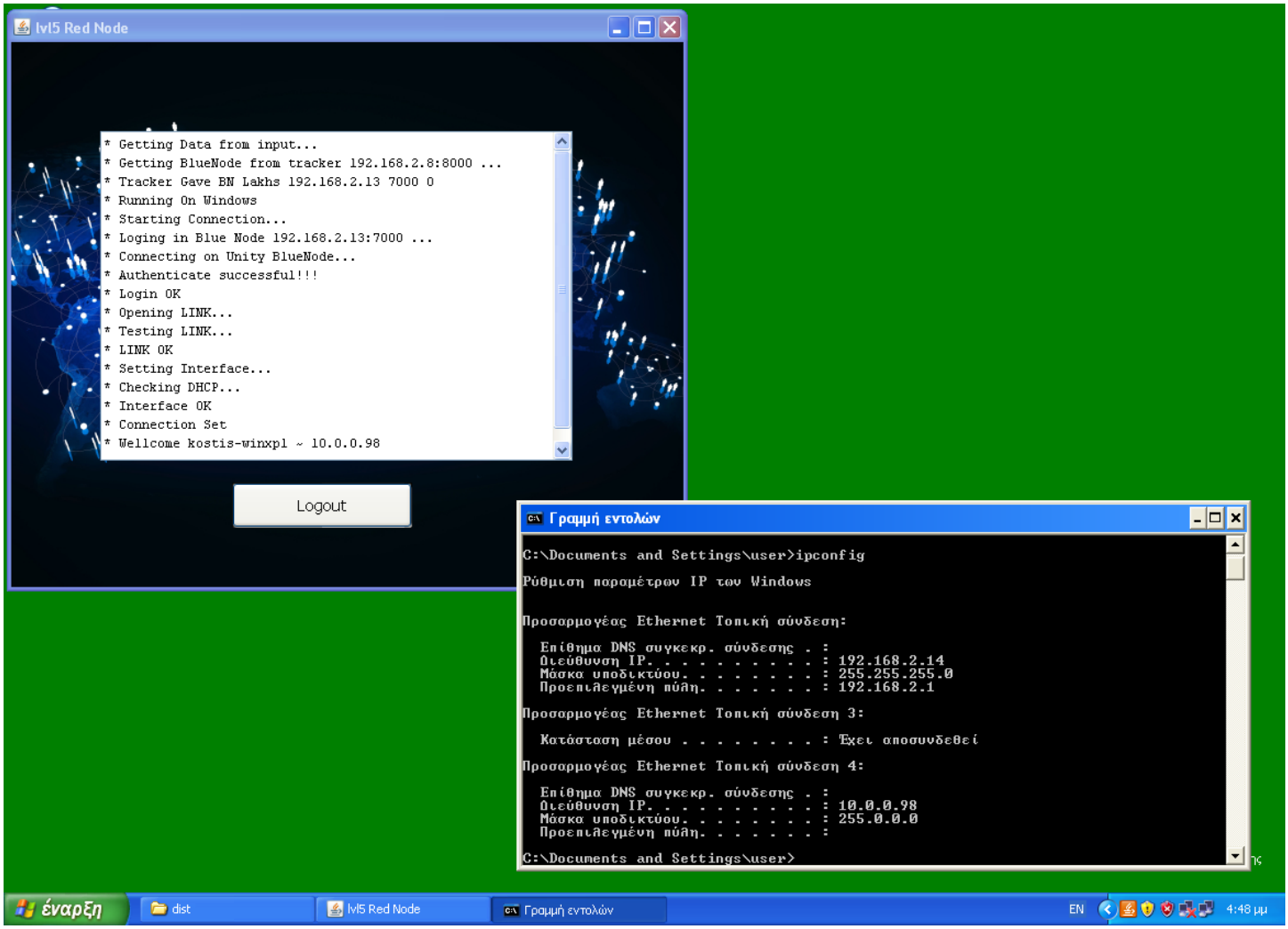
hostname	IP	OS
kostis-winxp1	192.168.2.14	WinXP_32
kostis-winxp2	192.168.2.18	WinXP_32
kostis-winxp3	192.168.2.15	WinXP_32
kostis-ubuntu	192.168.2.19	Ubuntu_32
kostis-lubuntu5	192.168.2.16	LUbuntu_32

Οι RNs έχουν τα κλειδιά στο dir τους και ανοίγουν το πρόγραμμα του RN για να συνδεθούν. Θα δούμε ενδεικτικά ένα winxp και το ubuntu να συνδέεται. Όταν συνδέονται οι κόμβοι ο tracker θα τους μοιράσει στους BN ισάξια ώστε να έχουν ίδιο φόρτο.

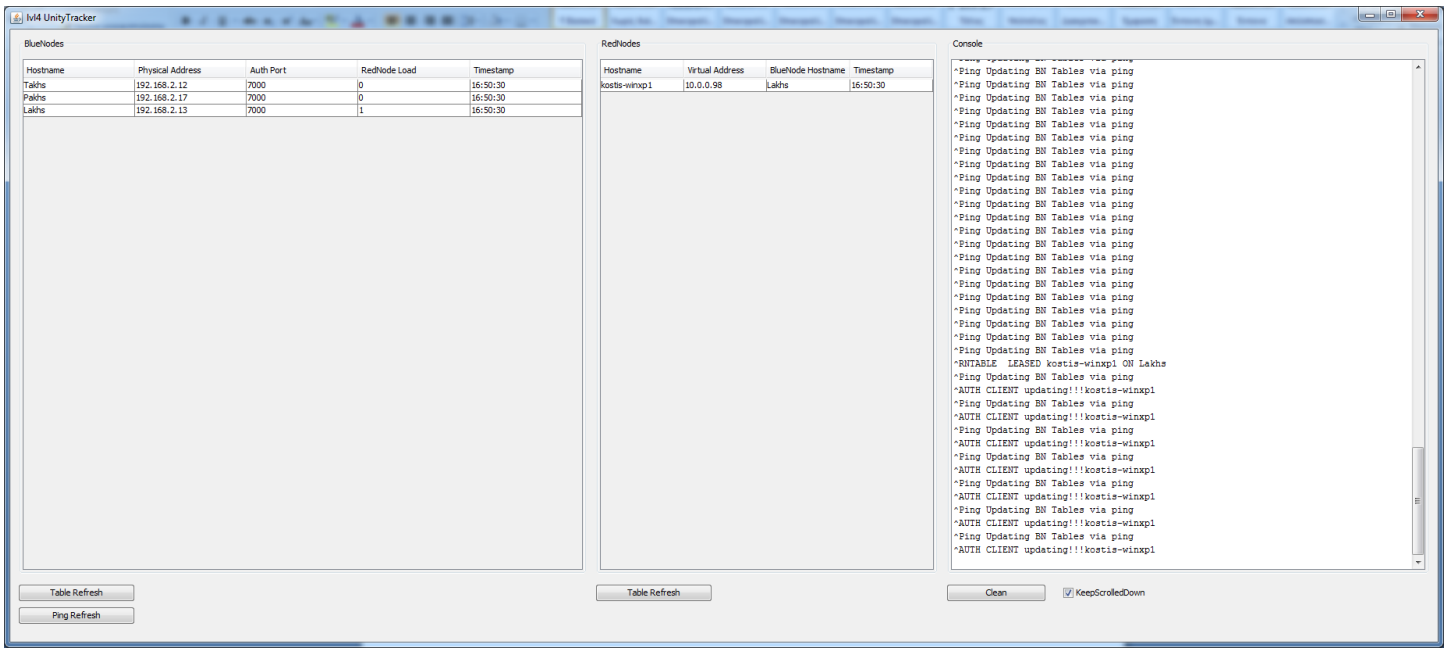
Στα winxp στο kostis-winxp1:

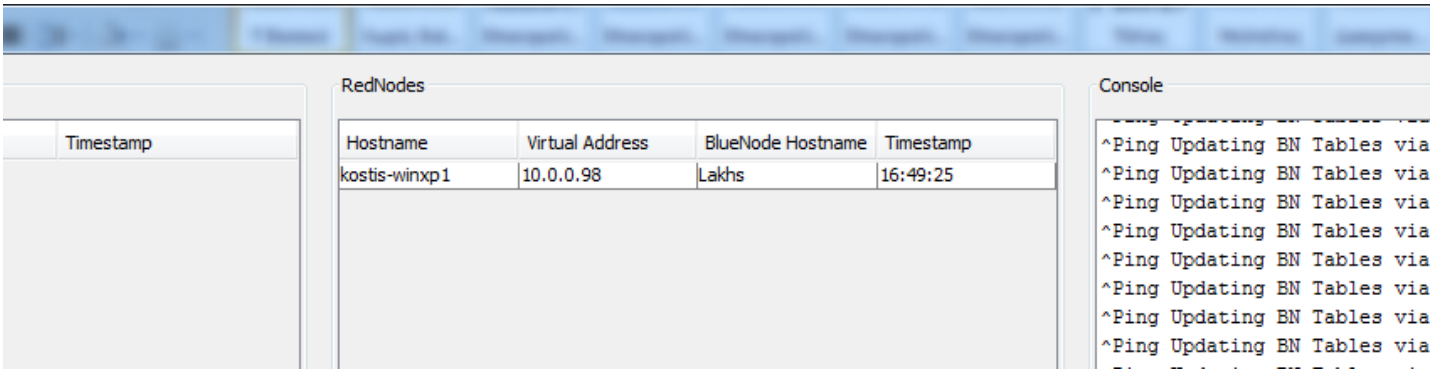


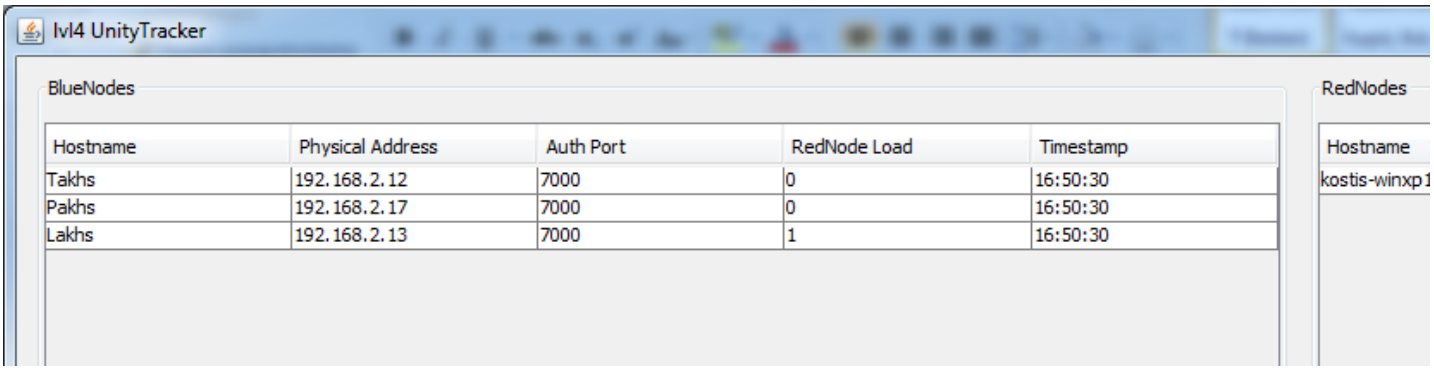




Αντίστοιχα στον tracker φαίνεται ο συνδεμένος RN





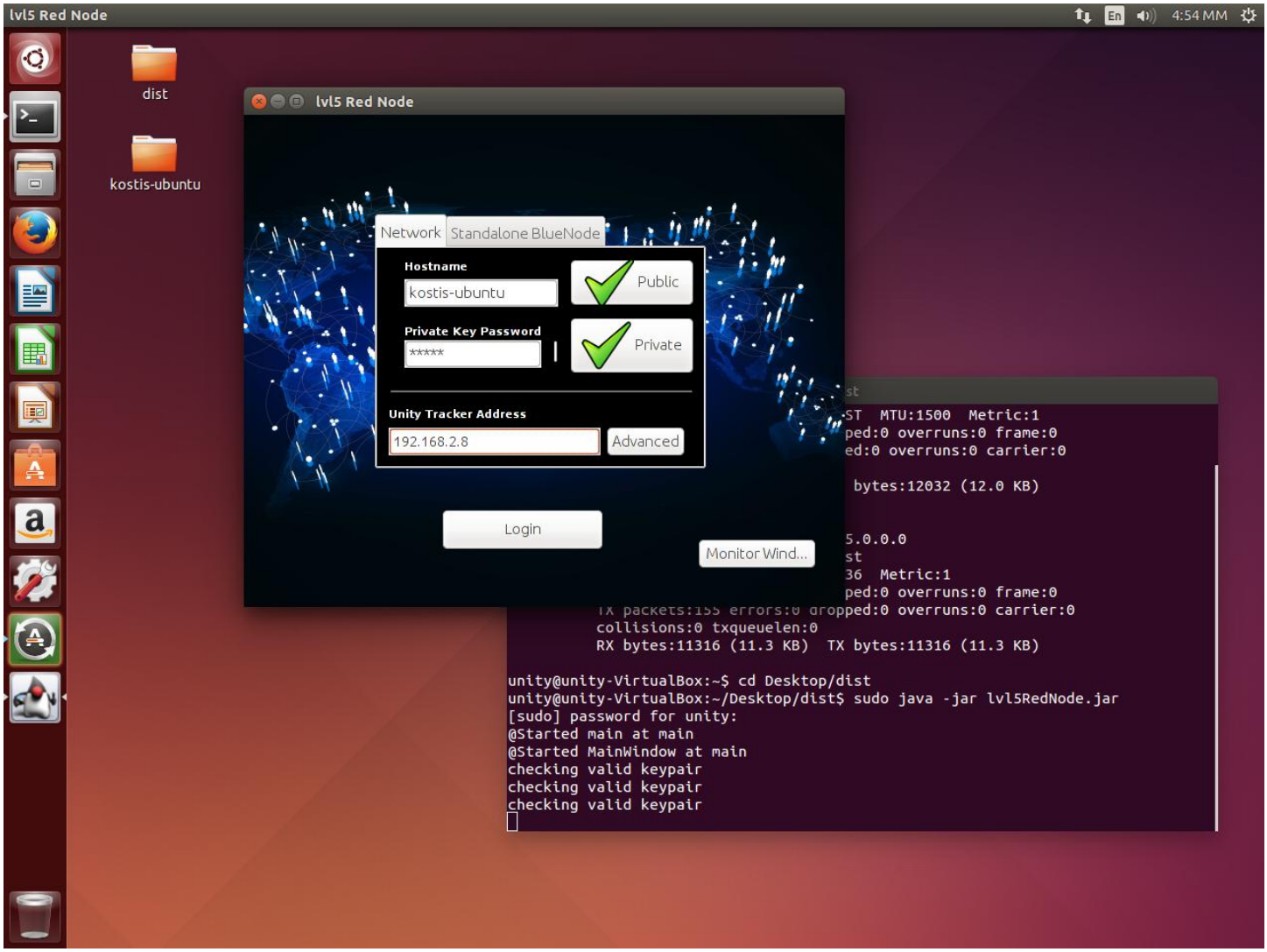


Τον RN τον έχει αναλάβει ο BN Lakhs και έχει αυξηθεί ο φόρτος του κατά 1!

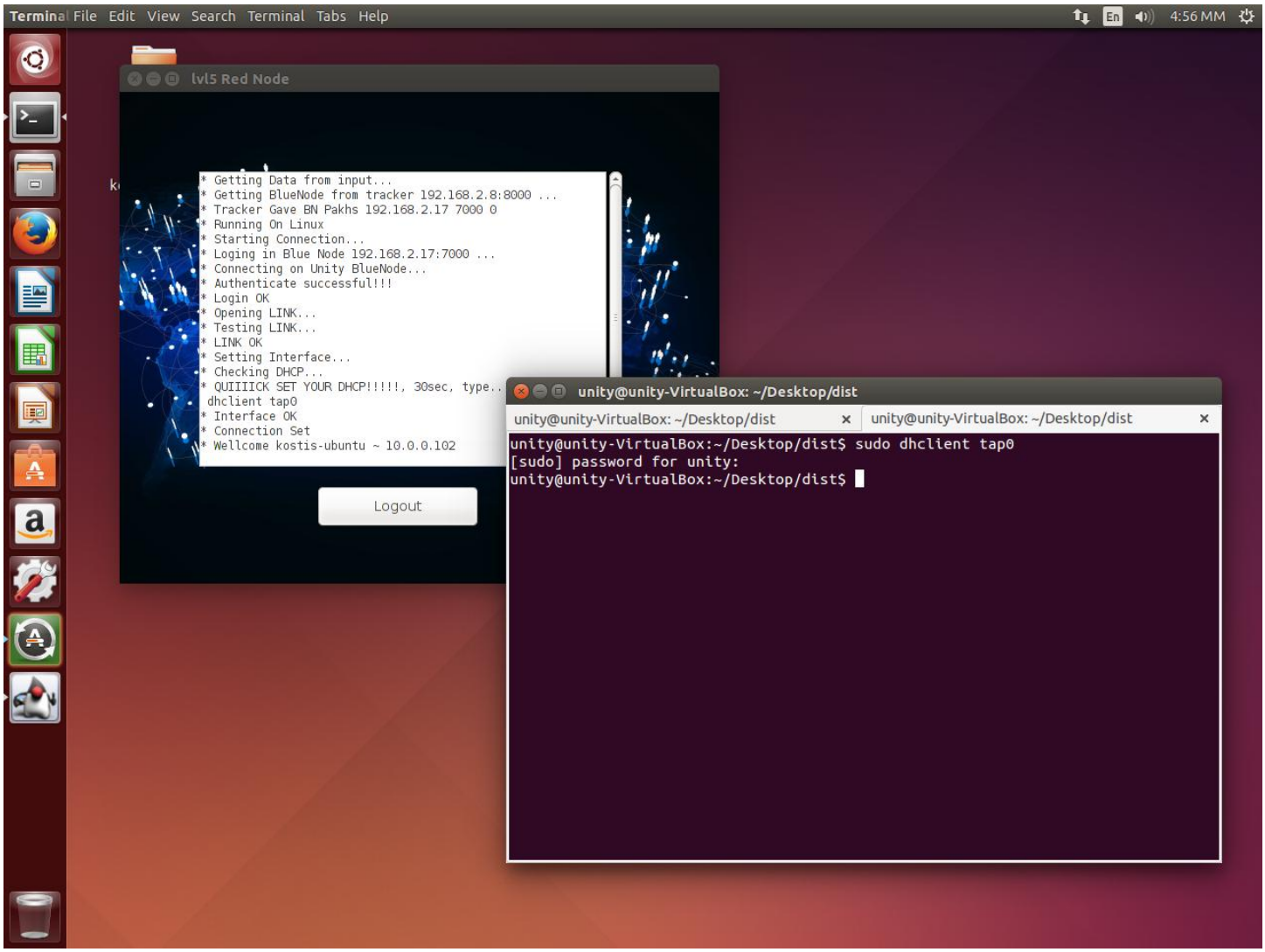
kostis-ubuntu

Τώρα πάμε να συνδέσουμε τα Ubuntu.

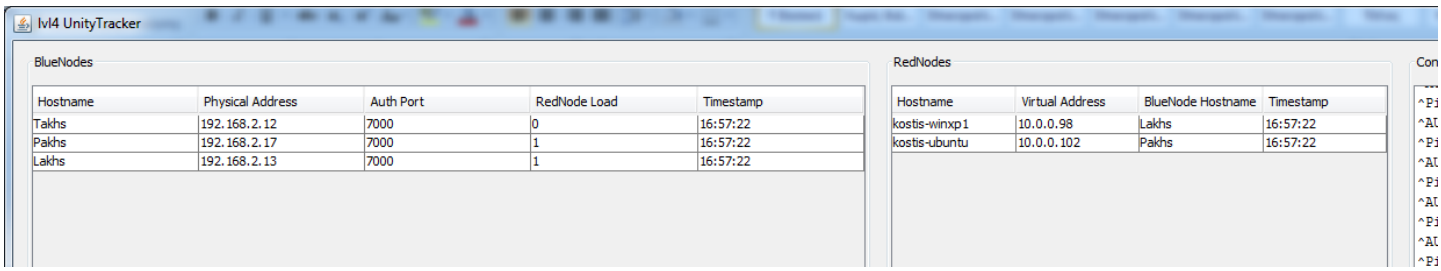
Στα ubuntu ο RN επειδή ανοίγει tuntap θέλει δικαιώματα διαχειριστή! (άρα τρέχει με sudo)



Επίσης κατά τη σύνδεση θα ζητηθεί να ενεργοποιηθεί ο dhclient για να κάνει ερώτημα dhcp και να πάρει απάντηση.

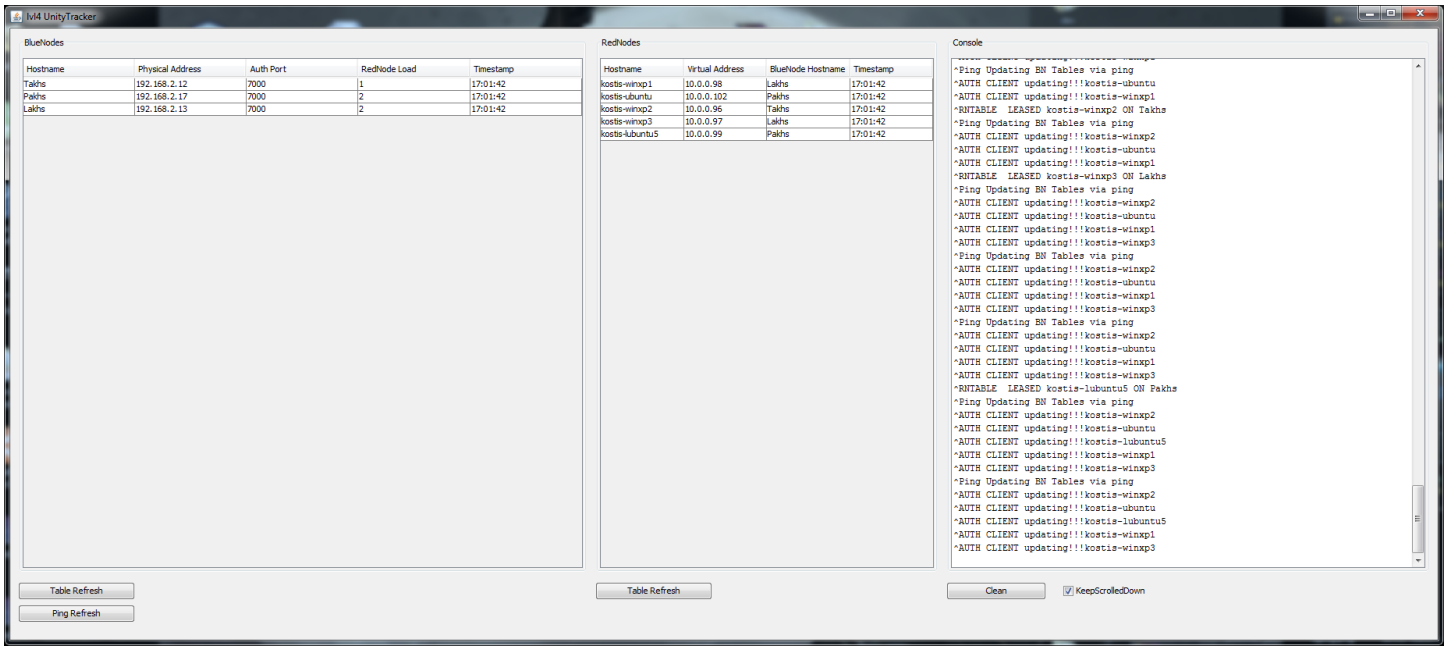


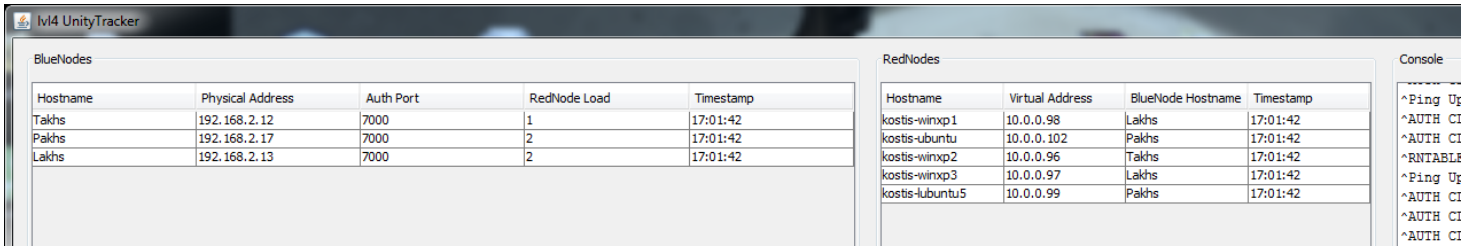
Στον tracker έχουμε 2 RNs σε 2 BNs



Όμοια ανοίγουμε όλους τους RN και έχουμε την παρακάτω τελική εικόνα.

Τελική εικόνα του Tracker για το test!





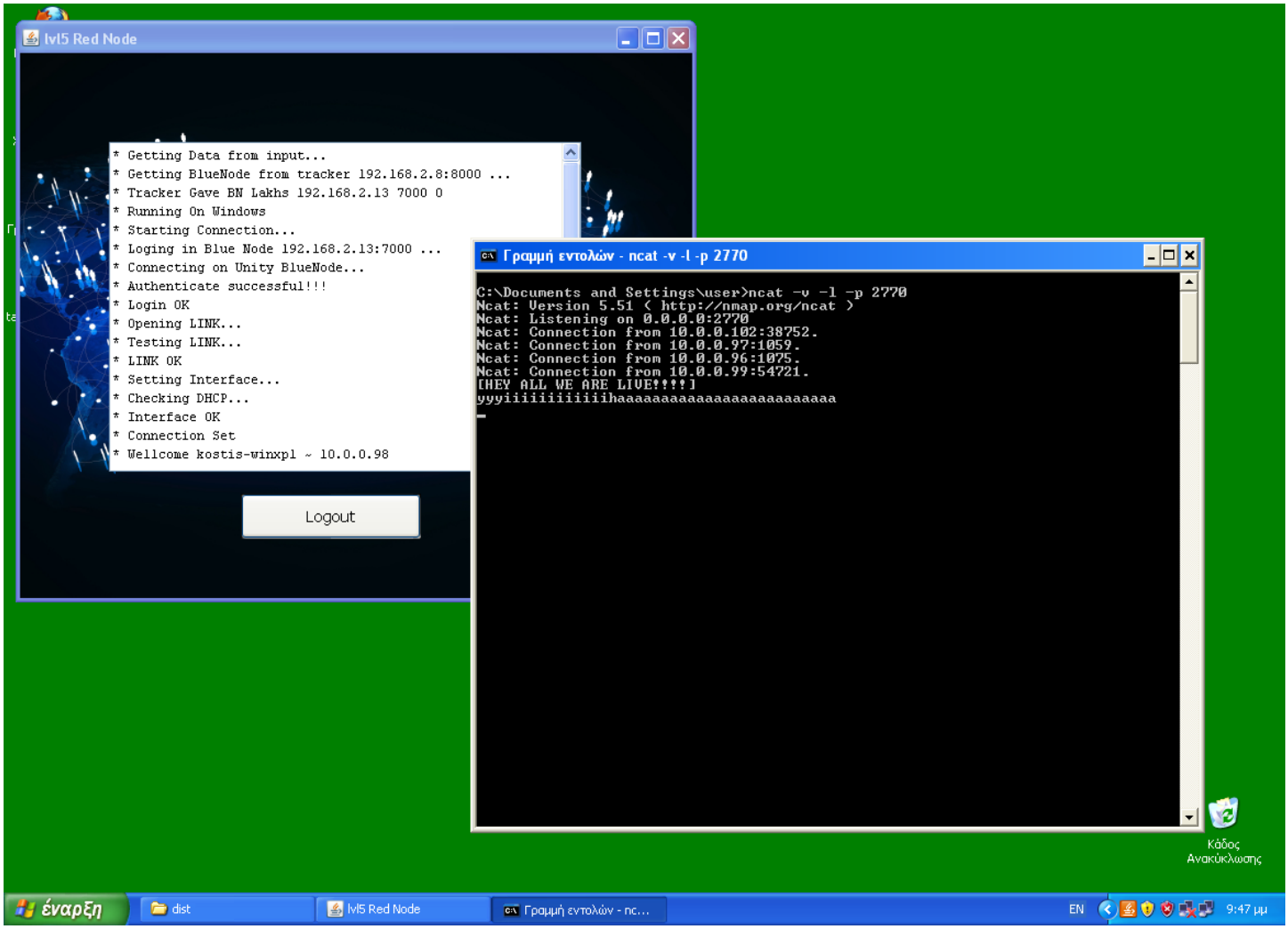
Πλέον έχουν συνδεθεί και οι 5 RedNode.

Μέχρι τώρα τα στοιχεία των RN είναι:

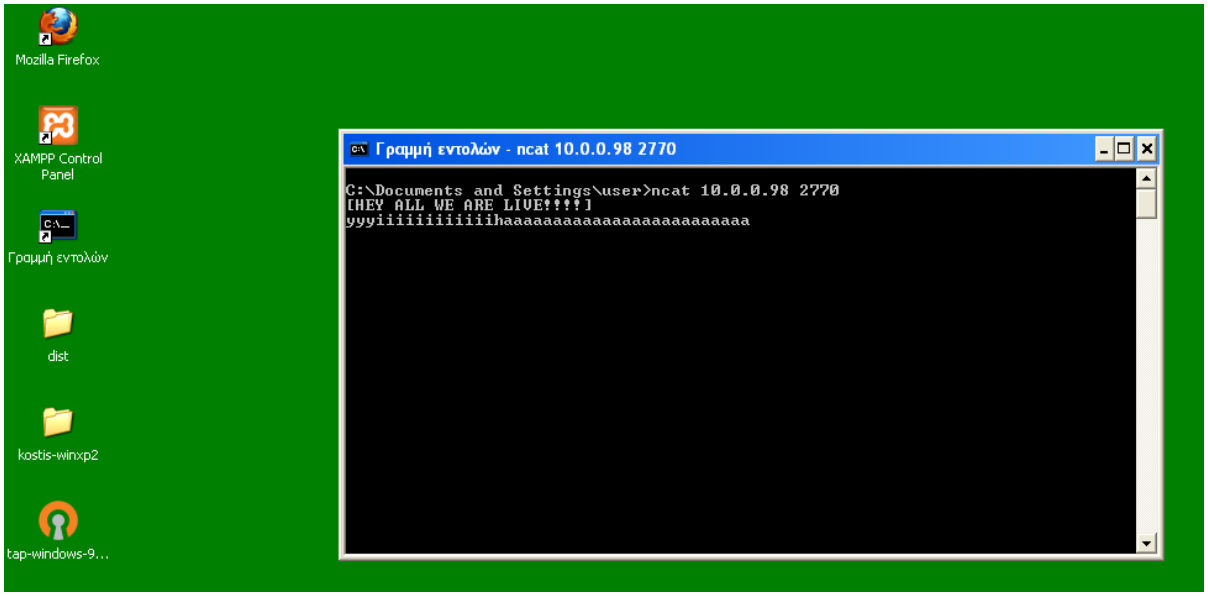
hostname	LAN IP (external)	Virtual IP (Internal)	OS
kostis-winxp1	192.168.2.14	10.0.0.98	WinXP_32
kostis-winxp2	192.168.2.18	10.0.0.96	WinXP_32
kostis-winxp3	192.168.2.15	10.0.0.97	WinXP_32
kostis-ubuntu	192.168.2.19	10.0.0.102	Ubuntu_32
kostis-lubuntu5	192.168.2.16	10.0.0.99	LUbuntu_32

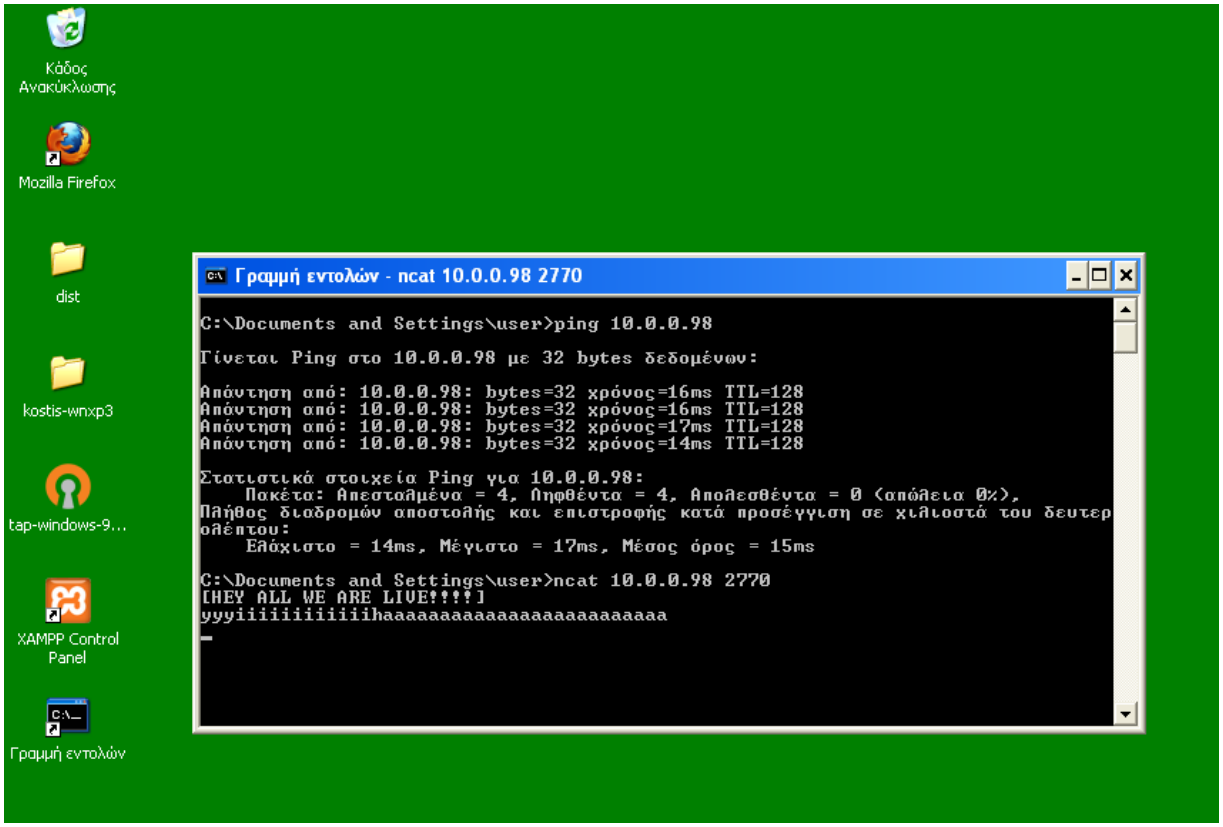
Τώρα σειρά έχει να δοκιμαστεί το δίκτυο!

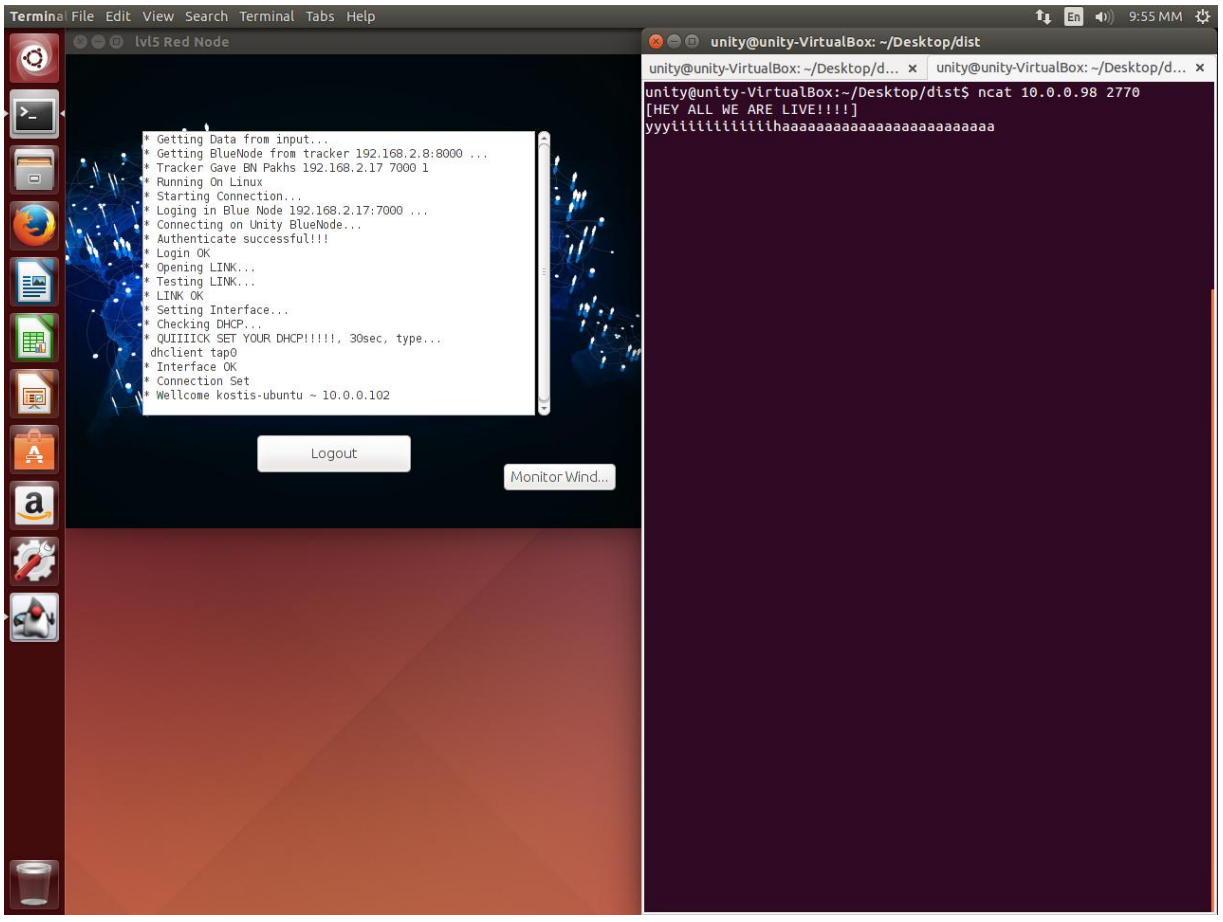
Για να γίνει αυτό θα πρέπει να γίνει χρήση μιας υπηρεσίας IP. Στην προκειμένη περίπτωση ένα TCP chat είναι οτι πρέπει. Ο kostis-winxp1 θα αναλάβει να κάνει ένα δωμάτιο και οι υπόλοιποι 4 θα συνδεθούνε εκεί, Στη συνέχεια θα στείλει ένα μύνημα όπου θα το λάβουν όλοι αλλά μέσω του virtual network!

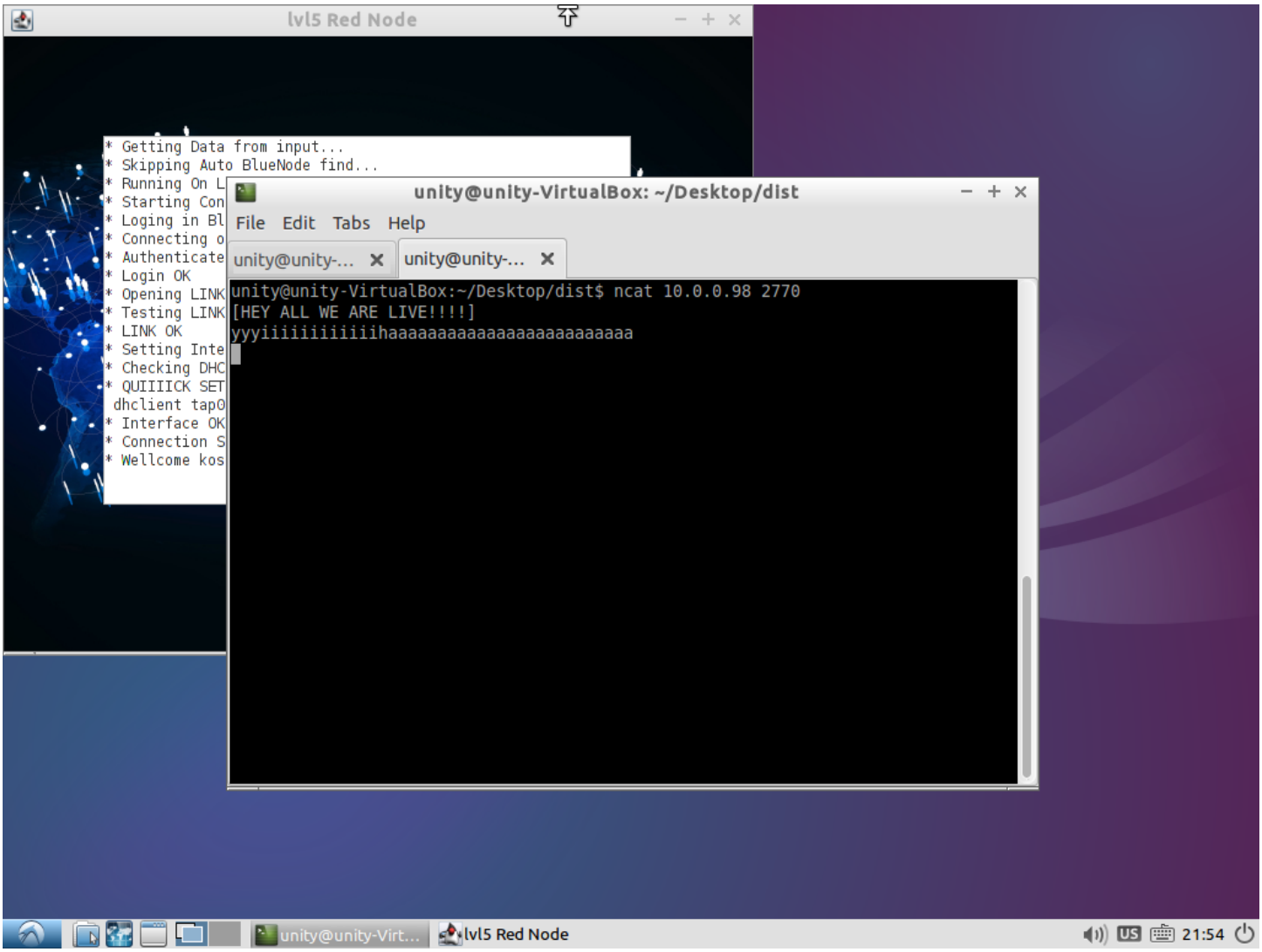


Υπενθυμίζουμε οτι ο winxp1 και οι υπόλοιποι που πρόκειται να μιλήσουν **δεν** βρίσκονται όλοι στον ίδιο BN. Αυτό σημαίνει οτι οι BN Θα πρέπει να βρεθούν μέσω του tracker και αυτό θα χάσει μερικά πακέτα στους υπόλοιπους. Οπότε κάνουν ένα μικρό ping για αρχή και όταν πλέον οι BN έχουν ταιριάξει συνδέεται στον TCP server.




Κάδος Ανακύκλωσης
Mozilla Firefox
dist
kostis-wnxp3
tap-windows-9...
XAMPP Control Panel
Γραμμή εντολών
Γραμμή εντολών - ncat 10.0.0.98 2770
C:\Documents and Settings\user>ping 10.0.0.98
Γίνεται Ping στο 10.0.0.98 με 32 bytes δεδομένων:
Απάντηση από: 10.0.0.98: bytes=32 χρόνος=16ms TTL=128
Απάντηση από: 10.0.0.98: bytes=32 χρόνος=16ms TTL=128
Απάντηση από: 10.0.0.98: bytes=32 χρόνος=17ms TTL=128
Απάντηση από: 10.0.0.98: bytes=32 χρόνος=14ms TTL=128
Στατιστικά στοιχεία Ping για 10.0.0.98:
Πακέτα: Απεστολμένα = 4, Ληφθέντα = 4, Απολεσθέντα = 0 (απώλεια 0%),
Πλήθος διαδρομών αποστολής και επιστροφής κατά προσέγγιση σε χιλιοστά του δευτερολέπτου:
Ελάχιστο = 14ms, Μέγιστο = 17ms, Μέσος όρος = 15ms
C:\Documents and Settings\user>ncat 10.0.0.98 2770
[HEY ALL WE ARE LIVE!!!!]
yyyiiiiiiiiiiihaaaaaaaaaaaaaaaaaaaaaaaa


Terminal File Edit View Search Terminal Tabs Help
9:55 MM
lvl5 Red Node
* Getting Data from input...
* Getting BlueNode from tracker 192.168.2.8:8000 ...
* Tracker Gave BN Pakhs 192.168.2.17 7000 1
* Running On Linux
* Starting Connection...
* Loging in Blue Node 192.168.2.17:7000 ...
* Connecting on Unity BlueNode...
* Authenticate successful!!!
* Login OK
* Opening LINK...
* Testing LINK...
* LINK OK
* Setting Interface...
* Checking DHCP...
* QUIIIICK SET YOUR DHCP!!!!!, 30sec, type...
dhclient tap0
* Interface OK
* Connection Set
* Wellcome kostis-ubuntu ~ 10.0.0.102
Logout
Monitor Wind...
unity@unity-VirtualBox: ~/Desktop/dist
unity@unity-VirtualBox: ~/Desktop/d...
unity@unity-VirtualBox: ~/Desktop/d...
unity@unity-VirtualBox:~/Desktop/dist$ ncat 10.0.0.98 2770
[HEY ALL WE ARE LIVE!!!!]
yyyiiiiiiiiiiihaaaaaaaaaaaaaaaaaaaaaaaa



Οι RNs μπορούν να χρησιμοποιήσουν οποιαδήποτε υπηρεσία IP επιθυμούν και εκτός από chat όπως για παράδειγμα file transfer, video conference, voice, games κ.α.