Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Υλοποίηση μιας ασφαλούς client – server εφαρμογής πολλαπλών υπηρεσιών με χρήση δικτύου VPN και κρυπτογραφημένης επικοινωνίας Implementantion of a secure multi-service client-server application using VPN and encrypted communication
Ονοματεπώνυμο Φοιτητή	Παναγιώτης Πετρόπουλος
Πατρώνυμο	Αθανάσιος
Αριθμός Μητρώου	ΜΠΣΠ 13090
Επιβλέπων	Κωνσταντίνος Πατσάκης

Ημερομηνία Παράδοσης Οκτώβριος 2017

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Πατσάκης Κωνσταντίνος Επίκουρος καθηγητής	Τσιχριντζής Γεώργιος Καθηγητής	Αλέπης Ευθύμιος Επίκουρος καθηγητής

Μεταπτυχιακή Διατριβή

Παναγιώτης Πετρόπουλος

ΕΙΣΑΓΩΓΗ

Στην παρούσα μεταπτυχιακή διατριβή επιλέξαμε να αναπτύξουμε μια client-server εφαρμογή κοινότητας και ασφαλούς επικοινωνίας, η οποία θα προσφέρει πολλαπλές δυνατότητες στο χρήστη, οι βασικότερες των οποίων: ασφαλή σύνδεση για κρυπτογραφημένη επικοινωνία, κρυπτογραφημένη περιήγηση στο διαδίκτυο, αυτοματοποίηση της διαδικασίας σύνδεσης χωρίς να απαιτείται ο τελικός χρήστης να έχει καμία τεχνική γνώση επί του θέματος, σύνδεση σε κοινότητα με σκοπό τον διαμοιρασμό πληροφορίας με τα υπόλοιπα μέλη της κοινότητας, χρονομέτρηση και επίβλεψη της σύνδεσης του χρήστη, καθώς και αφαίρεση του δικαιώματος σύνδεσης όταν υπερβεί τον απαιτούμενο χρόνο. **Ο τελικός σκοπός σε αυτή τη μεταπτυχιακή διατριβή ήταν, λαμβάνοντας υπ' όψιν όλα τα παραπάνω, η εφαρμογή μας να έχει χρηστική αξία και τη δυνατότητα να χρησιμοποιηθεί από πολλά είδη χρηστών, και για πολλαπλούς σκοπούς, στον πραγματικό κόσμο.** Ανάμεσα στις υπηρεσίες που προσφέρει η εφαρμογή μας εντός του εσωτερικού VPN δικτύου είναι οι εξής: υπηρεσία προβολής βίντεο, μέσο κοινωνικής δικτύωσης, υπηρεσία ζωντανής ανταλλαγής μηνυμάτων, FTP server, wiki, etherpad.

ABSTRACT

In this thesis, we chose to develop a client-server community and secure communication application that will offer multiple capabilities to the user, the most basic of which: secure connection for encrypted communication, encrypted internet browsing, automation of the connection process without requiring the end-user to have any technical knowledge on the subject, linking to a community to share information with other members, connection timing and controlling of the user, and disconnecting right when a certain amount of time time has exceeded. **The ultimate goal in this thesis was, taking into account all of the above, our application to have a usable value and the ability to be used by many types of users, and for multiple purposes, in the real world.** Among the services offered by our application within the internal VPN network are: video viewing service, social media, live chatting service, FTP server, wiki, etherpad.

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

α) Η ανάγκη για ψηφιακές κοινότητες (virtual communities)

Τα τελευταία χρόνια, η ανάπτυξη του διαδικτύου και του παγκόσμιου ιστού **έχει προκαλέσει την ταχεία εμφάνιση εφαρμογών που έχουν ως σκοπό την αλληλεπίδραση και την ανταλλαγή πληροφορίας απομακρυσμένων χρηστών.** Όσο αναπτύσσεται η τεχνολογία, όλο και περισσότερο αντικαθίσταται η ανταλλαγή πληροφορίας μέσω "κλασικών" τρόπων επικοινωνίας (χαρτί, μέσα αποθήκευσης ψηφιακής πληροφορίας και άλλα) με διαδικτυακές εφαρμογές.

Στο διαδίκτυο συναντάμε συνεχώς τέτοιες εφαρμογές, οι οποίες ποικίλλουν ως προς το είδος της πληροφορίας που διακινείται, από διαμοιρασμό κειμένων, αρχείων, πολυμέσων ή όλων αυτών μαζί, αλλά και ως προς τις ιδιότητες και των αριθμό των χρηστών. Συναντάμε από εφαρμογές που απασχολούν μια μικρή ομάδα χρηστών με κοινές ιδιότητες και ενδιαφέροντα, μέχρι εφαρμογές που απασχολούν εκατομμύρια χρήστες γενικού ενδιαφέροντος. Τις συναντάμε με την ονομασία "ψηφιακές κοινότητες" (virtual communities) και γνωρίζουν πολλές παραλλαγές. Μερικά από τα πιο συνηθισμένα παραδείγματα τέτοιων εφαρμογών είναι οι εφαρμογές κοινωνικής δικτύωσης (social media), τα φόρουμ, οι εφαρμογές διαμοιρασμού πολυμέσων, τα ιστολόγια (blogs), τα wikis, τα web-offices, οι εφαρμογές συζήτησης (chatting applications), και άλλα.

Ταυτόχρονα και με την μεγάλη πρόοδο των εφαρμογών αυτών, αναπτύχθηκαν και αντίστοιχα προβλήματα προς επίλυση, τα οποία έχριζαν μελέτης στην επιστήμη των υπολογιστών. Το πώς θα γίνει η διαχείριση του μεγάλου όγκου της πληροφορίας στη διακίνηση και στην αποθήκευση, το πώς θα γίνει η μετάδοση της πληροφορίας με ασφάλεια, η προστασία των προσωπικών δεδομένων, και άλλα πολλά. **Στην παρούσα μεταπτυχιακή διατριβή θα ασχοληθούμε σε μεγάλο βαθμό με το ζήτημα της ασφαλούς σύνδεσης και επικοινωνίας.**

β) η ανάγκη για ασφαλή επικοινωνία

Από την καθημερινή ζωή μας στο διαδίκτυο, γνωρίζουμε πολλά παραδείγματα ανταλλαγής πληροφορίας μεταξύ δύο οντοτήτων ή χρηστών, στα οποία θέλουμε αποκλειστικά η πληροφορία να είναι προσβάσιμη μόνο από αυτές τις δύο οντότητες. Η "ανοιχτή" και ελεύθερη φύση του διαδικτύου από γεννησιμιού του, καθώς και ότι πολλά από τα πρωτόκολλα τα οποία αναπτύχθηκαν για ανταλλαγή ψηφιακής πληροφορίας δεν κατασκευάστηκαν εξ' αρχής για αυτό το σκοπό, είναι λόγοι οι οποίοι καθιστούν **απαραίτητη την εγκατάσταση μιας επιπλέον υποδομής προκειμένου να προσδίδεται επιπλέον ασφάλεια στην επικοινωνία.** Χωρίς αυτήν, μπορεί εύκολα κάποιος κακόβουλος χρήστης να αποκτήσει πρόσβαση στην πληροφορία που ανταλλάσσεται, ή ακόμα περισσότερο, αυτή η αδυναμία να αποτελέσει την "κερκόπορτα" για πρόσβαση σε άλλα κομμάτια του δικτύου. **Όλες αυτές οι αδυναμίες ενός δικτύου ονομάζονται "ευπάθειες" (vulnerabilities).**

Τις ευπάθειες ενός δικτύου, κάποιος μπορεί να τις εκμεταλλευθεί (exploit) κάνοντας κάποια επίθεση (attack). Υπάρχουν πάμπολλα είδη επιθέσεων ασφαλείας (security attacks) που χρήζουν διάφορους τρόπους επίλυσης και προστασίας. Από ευπάθειες στο ίδιο το δίκτυο ή το λογισμικό που χρησιμοποιούμε (επιλύονται με firewall ή συστήματα ανίχνευσης "ύποπτης" κίνησης στο δίκτυο), μέχρι τεχνικές "πειρατείας" (session hijacking) ή παρακολούθησης πακέτων (packet sniffing) σε μη προστατευμένα δίκτυα. Όλες οι δομές ασφαλείας που είναι γνωστές στην επιστήμη των υπολογιστών και των δικτύων, καλούνται να επιλύσουν όλων αυτών των τύπων τις ευπάθειες, και να ασφαλίσουν τις εφαρμογές ενάντια στις αντίστοιχες επιθέσεις. **Αυτού του τύπου συνεχώς μεταλλάσσονται, αλλάζουν, ανανεώνονται, γι' αυτό και χρειάζεται συνεχής εφάμιλλη ενημέρωση στην ασφάλεια και στη συντήρηση ενός δικτύου.**

Τύποι επίθεσης όπως η υποκλοπή πακέτων κατά την ανταλλαγή τους μεταξύ δύο οντοτήτων σε μια ψηφιακή επικοινωνία, καθώς και πολλοί άλλοι τύποι επίθεσης λιγότερο συνηθισμένοι, μπορούν και ασφαλίζονται σε ικανοποιητικό βαθμό **με χρήση τεχνικών κρυπτογράφησης της επικοινωνίας και πιστοποίησης των χρηστών.** Με αυτές τις τεχνικές καταπιαστήκαμε σε μεγάλο βαθμό κατά την εκπόνηση της παρούσας μεταπτυχιακής διατριβής.

γ) Κρυπτογράφηση και πιστοποίηση

Η πιστοποίηση και η κρυπτογράφηση είναι δύο έννοιες οι οποίες συνδιάζονται συχνά σε τεχνικό επίπεδο, και εφαρμόζονται ευρέως όταν θέλουμε σε ένα δίκτυο να διασφαλίσουμε ότι τα δεδομένα μας παραμένουν ασφαλή. Στις περισσότερες, αν όχι όλες, τεχνικές ασφαλείας δικτύων, η πιστοποίηση δεν υπάρχει χωρίς κρυπτογράφηση, και το ανάποδο.

Η κρυπτογράφηση είναι η διαδικασία κατά την οποία τα μηνύματα που ανταλλάσσονται μετασχηματίζονται σε μία "ακατανόητη" μορφή με τη χρήση κάποιου κρυπτογραφικού αλγορίθμου. Η ανάστροφη διαδικασία χρήσης του αλγορίθμου, για ανάκτηση ξανά του αρχικού μηνύματος, ονομάζεται αποκρυπτογράφηση. Αν ο αλγόριθμος αυτός είναι μοναδικός, και γνωστός μόνο στον αποστολέα και τον παραλήπτη, έτσι ώστε μόνο αυτοί να μπορούν να κρυπτογραφούν και αποκρυπτογραφούν την πληροφορία που ανταλλάσσεται, τότε αυτή η μέθοδος επικοινωνίας μπορεί να θεωρηθεί και επαρκώς ασφαλισμένη. Ακόμα και αν πέσει στα χέρια κάποιου κακόβουλου χρήστη ή λογισμικού κάποιο μήνυμα, αν αυτό είναι κρυπτογραφημένο, του είναι εξαιρετικά χρονοβόρο έως αδύνατον να ανακτήσει το περιεχόμενό του, και άρα του είναι άχρηστο. Τρόποι κρυπτογράφησης μηνυμάτων υπάρχουν πολλοί, και έχουν ως προϋπόθεση ο αλγόριθμος κρυπτογράφησης (cipher) να είναι γνωστός και στους δύο χρήστες. Τα μηνύματα κρυπτογραφούνται με τη χρήση ενός κλειδιού κρυπτογράφησης (key) το οποίο είτε θα είναι γνωστό και στους δύο από πριν (pre-shared key) είτε ο κάθε χρήστης θα διαθέτει δύο κλειδιά, το ιδιωτικό (private key), το οποίο ο χρήστης πρέπει να κρατάει κρυφό, και το δημόσιο (public key), που μπορεί να μοιραστεί με τον άλλον. Τα δύο αυτά κλειδιά έχουν μια μαθηματική σχέση μεταξύ τους, έτσι ώστε το ένα να μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση ενός μηνύματος και το άλλο για την αποκρυπτογράφηση αυτού. Το ότι η γνώση του δημόσιου κλειδιού δεν επιτρέπει με κανέναν τρόπο τον υπολογισμό του ιδιωτικού κλειδιού, είναι που προσδίδει ασφάλεια στην επικοινωνία.

Η πιστοποίηση της προέλευσης και αυθεντικότητας της πληροφορίας χωρίζεται σε δύο σκέλη. Το πρώτο αναφέρεται στις δύο οντότητες που επικοινωνούν, και αποτελεί τη διαδικασία κατά την οποία επιβεβαιώνεται ότι οι δύο χρήστες στα άκρα της επικοινωνίας είναι όντως αυτοί που ισχυρίζονται ότι είναι. Το δεύτερο σκέλος, είναι η διαδικασία κατά την οποία αναγνωρίζεται ότι η προέλευση της πληροφορίας που μεταδίδεται, είναι όντως αυτή που ανήκει στον -ήδη από το προηγούμενο βήμα- πιστοποιημένο χρήστη. Έτσι λοιπόν μέσω της πιστοποίησης επιτυγχάνονται και τα δύο, μπορούμε δηλαδή να διαπιστώσουμε ότι η επικοινωνία γίνεται μεταξύ των "σωστών" οντοτήτων και πως η πληροφορία "ξεκινά και καταλήγει" από αυτούς και σε αυτούς.

Όλες οι εφαρμογές που χρησιμοποιούνται για ασφαλή επικοινωνία σε δίκτυα, χρησιμοποιούν έναν συνδυασμό αυτών των δύο πρακτικών.

δ) Tunneling και VPN

Στον κόσμο των υπολογιστών και των δικτύων συναντάμε συνεχώς διαφόρων μορφών τεχνικές σήραγγας (εφεξής tunneling). Οποτεδήποτε ακούμε για συνδέσεις Εικονικών Ιδιωτικών Δικτύων (Virtual Private Networks, εφεξής VPN) αυτές χρησιμοποιούν tunnels. Με το tunneling μπορούμε να επιτύχουμε ασφαλή επικοινωνία και πλήρη ιδιωτικότητα στις επικοινωνίες δικτύων.

Το tunneling είναι ένας μηχανισμός ο οποίος μπορεί να μεταδώσει ένα "ξένο" πρωτόκολλο επικοινωνίας, εντός ενός καναλιού που από τη φύση του δε μπορεί να το υποστηρίξει. Τα πρωτόκολλα tunneling μας επιτρέπουν να χρησιμοποιήσουμε, για παράδειγμα, το πρωτόκολλο IP (Internet Protocol) για να στείλουν κάποιο άλλο πρωτόκολλο χρησιμοποιώντας το ίδιο κανάλι.

Τα δίκτυα VPN επί της ουσίας είναι εγκατεστημένα ιδιωτικά δίκτυα εντός άλλων κοινόχρηστων δικτύων (όπως πχ. το διαδίκτυο) χρησιμοποιώντας τη μέθοδο tunneling. Από τη στιγμή που ένα πακέτο στο δίκτυο δρομολογηθεί να περάσει μέσα από το tunnel, επιπλέον πληροφορία προστίθεται και "περιβάλλει" το πακέτο (packet encapsulation) έτσι ώστε να "υπακούει" στο νέο πρωτόκολλο, και φυσικά η ωφέλιμη πληροφορία εντός του πακέτου (payload) κρυπτογραφείται. Έτσι οι δύο οντότητες που έχουν συνδεθεί στο δίκτυο VPN, συμπεριφέρονται προσομοιώνοντας απόλυτα την επικοινωνία εντός ενός ιδιωτικού δικτύου (με ξεχωριστές διευθύνσεις IP, κλπ), ενώ στην πραγματικότητα χρησιμοποιούν τους πόρους ενός ευρύτερου ανοιχτού δικτύου.

Φυσικά οι εφαρμογές οι οποίες "μιλούν" μέσω VPN κερδίζουν σε αξιοπιστία, ασφάλεια και διαχείριση δικτύου. Το μοναδικό κόστος που επωμίζονται είναι η μικρότερη "ωφέλιμη" χωρητικότητα πληροφορίας ανά πακέτο, καθώς προσθέτουν επιπλέον πληροφορία στο πακέτο κατά τη διάρκεια του encapsulation.

ε) η δική μας εφαρμογή

Στην παρούσα μεταπτυχιακή διατριβή καταπιαστήκαμε με το θέμα της ανάπτυξης μιας εφαρμογής κοινότητας, η οποία θα έχει ως βασικό χαρακτηριστικό την ασφαλή και κρυπτογραφημένη σύνδεση των χρηστών και επικοινωνία μεταξύ τους μέσω των κατάλληλων υπηρεσιών, ή επικοινωνία με τον "υπόλοιπο κόσμο" μέσω VPN. Θέλουμε να υπάρχει έλεγχος στους χρήστες που εγγράφονται, και στο χρόνο στον οποίο θα έχουν διαθέσιμο για χρήση της εφαρμογής. Η αποθήκευση των ευαίσθητων δεδομένων των χρηστών θα εκτελείται με ασφάλεια, και για λόγους επίσης ασφαλείας οι χρήστες δε θα έχουν δικαίωμα να μένουν συνδεδεμένοι "για πάντα", αλλά αντίθετα, θα αποσυνδέονται όταν λήξει ο συγκεκριμένος χρόνος που έχουν αιτηθεί. Οι χρήστες από τη στιγμή της σύνδεσής τους θα μπορούν είτε να χρησιμοποιήσουν τις διάφορες υπηρεσίες που τους προσφέρει η εφαρμογή μας, είτε να περιηγηθούν στο διαδίκτυο "προστατευόμενοι" από το VPN δίκτυο που είναι συνδεδεμένοι, όντας έτσι και μη ανιχνεύσιμοι.

ε1) Περίπτωση χρήσης (use case) της εφαρμογής

Ο χρήστης ανοίγει την ιστοσελίδα της εφαρμογής και κάνει login χρησιμοποιώντας το όνομα και τον κωδικό πρόσβασης που του έχει δοθεί. Στη συνέχεια ο χρήστης μπορεί να αιτηθεί συνδεσιμότητα στην εφαρμογή, μέσω μιας φόρμας που συμπληρώνει τα λεπτά το χρόνο που επιθυμεί. Για το συγκεκριμένο χρονικό διάστημα, αφού υποβάλλει τη φόρμα αυτή, θα του δίνεται η δυνατότητα να συνδεθεί και να αποσυνδεθεί όσες φορές αυτός το επιθυμεί.

Αφού επιλέξει το λειτουργικό σύστημα στο οποίο θέλει να τρέξει την εφαρμογή (υποστηρίζονται τα λειτουργικά συστήματα Linux και Windows), τότε κάνει λήψη ενός πακέτου αρχείων ικανών (το πακέτο αυτών των αρχείων έχει όνομα linux_app.zip ή windows_app.zip αντίστοιχα) να πραγματοποιήσουν τη σύνδεση στην εφαρμογή απολύτως αυτόματα.

Στο πακέτο των αρχείων αυτών υπάρχει ένα εκτελέσιμο το οποίο αναλαμβάνει αυτή τη δουλειά. Εγκαθιστά όλο το απαραίτητο λογισμικό που απαιτείται, και, αφού ζητήσει ξανά τα στοιχεία (όνομα και κωδικό πρόσβασης) του χρήστη, ξεκινά μια νέα VPN σύνδεση με τον server της εφαρμογής, χρησιμοποιώντας το λογισμικό OpenVPN.

Από τη στιγμή που ο χρήστης είναι συνδεδεμένος στην εφαρμογή μας, μπορεί να βλέπει από την ιστοσελίδα το χρόνο σύνδεσης που του απομένει, καθώς και ένα μήνυμα πως η σύνδεση είναι επιτυχής, και ένα σύνδεσμο (link) για τις επιπλέον λειτουργίες της εφαρμογής μας. Αυτό το link οδηγεί σε ένα "portal" για τις διαθέσιμες εφαρμογές, μεταξύ των οποίων social media, ένα wiki, μια υπηρεσία διαμοιρασμού αρχείων μέσω FTP, μια υπηρεσία προβολής βίντεο, μια υπηρεσίας ζωντανής συζήτησης μεταξύ χρηστών (live chatting) και άλλα. Σε κάθε περίπτωση, από τη στιγμή της σύνδεσής του και μετά, ο χρήστης έχει τη δυνατότητα να περιηγηθεί οπουδήποτε στο διαδίκτυο έχοντας "κρυμμένη" την προσωπική του IP διεύθυνση πίσω από αυτή του VPN server, έχοντας έτσι απόλυτη ασφάλεια στην περιήγησή του, όντας μη ανιχνεύσιμος.

Περισσότερες λεπτομέρειες για την εγκατάσταση, τη λειτουργία και αναλυτικά τις ενέργειες που εκτελεί η εφαρμογή μας, θα περιγραφούν στη συνέχεια της παρούσας διατριβής.

ε2) Η χρηστική αξία της δικής μας εφαρμογής

Πρώτον, ο χρήστης δε χρειάζεται να έχει καμία τεχνική γνώση για να συνδεθεί στον VPN server. Από τις γνωστές εφαρμογές client-server που εγκαθιστούν point-to-point VPN σύνδεση, όπως λόγου χάρη το OpenVPN (που είναι και το λογισμικό που χρησιμοποιούμε εμείς), ο χρήστης για να καταφέρει να συνδεθεί πρέπει οπωσδήποτε να έχει τεχνική γνώση σχετικά με αυτήν, καθώς θα πρέπει να "στήσει" πολλά αρχεία ρυθμίσεων, ενδεχομένως να κάνει μετατροπές στον πίνακα δρομολόγησης (routing table) του συστήματός του, και άλλα πολλά, με αποτέλεσμα να σπαταλάει χρόνο και

διαβάζοντας σωρεία πληροφοριών στο διαδίκτυο ή documentations. Η έλλειψη γνώσης από πλευράς του χρήστη μπορεί επίσης να οδηγήσει σε λανθασμένη διαμόρφωση παραμέτρων και ρύθμιση της εφαρμογής, πράγμα που μπορεί να την κάνει ανασφαλή. Αυτό είναι κάτι που θελήσαμε να αποφύγουμε, και θέσαμε ως βασικό στόχο στην παρούσα μεταπτυχιακή διατριβή.

Ο στόχος που επιτυγχάνει η εφαρμογή μας, είναι να προσφέρει την ασφάλεια και την ιδιωτικότητα ενός VPN δικτύου, χωρίς καν να απαιτείται από το χρήστη να γνωρίζει όχι απλά τεχνικά, αλλά ούτε καν σχετικά με την ύπαρξη αυτού. Μέσα από την ανάπτυξη της εφαρμογής καταφέραμε ο χρήστης να βρίσκεται συνδεδεμένος απολύτως αυτόματα, με το μόνο που χρειάζεται να είναι το "διπλό κλικ" σε ένα εκτελέσιμο αρχείο και εισαγωγή των στοιχείων του, δηλαδή ενός ονόματος χρήστη και ενός κωδικού πρόσβασης.

Δεύτερον, η εφαρμογή μας υποστηρίζεται σε δύο από τα τρία σημαντικότερα λειτουργικά συστήματα, Linux και Windows. Καλύπτει έτσι ένα τεράστιο ποσοστό (90%) ηλεκτρονικών υπολογιστών σε εταιρείες και ιδιώτες χρήστες.

Τρίτον, ο χρόνος παραμονής του χρήστη μέσα στην εφαρμογή είναι απόλυτα ελεγχόμενος από τον διαχειριστή της. Το να μένει ο χρήστης "για πάντα" συνδεδεμένος στον VPN server χωρίς κάποιον έλεγχο, ή χωρίς κάποιον μηχανισμό που να διακόπτει τη σύνδεση μετά από ένα προκαθορισμένο χρονικό διάστημα, μπορεί να δημιουργήσει ποικίλα προβλήματα, στην δέσμευση πόρων και καθυστέρηση στην απόδοση και την ταχύτητα της εφαρμογής, αλλά ακόμη και προβλήματα ασφάλειας ή μη-ιδιωτικότητας πλέον της επικοινωνίας.

Έτσι υλοποιήσαμε έναν μηχανισμό ο οποίος χρονομετρά την περίοδο που έχει τη δυνατότητα να συνδεθεί ο χρήστης. Σε περίπτωση που ο χρόνος τελειώσει όσο είναι ο χρήστης συνδεδεμένος, αυτός εσκεμμένα αποσυνδέεται από το δίκτυο μέχρι να αιτηθεί εκ νέου ανανέωση χρόνου για να μπορέσει να ξανασυνδεθεί και να συνεχίσει. Σε περίπτωση που ο χρόνος τελειώσει χωρίς να είναι συνδεδεμένος ο χρήστης, απλά του απαγορεύεται η εκ νέου σύνδεση. Εάν ο χρήστης αιτηθεί παραπάνω χρόνο τότε η εφαρμογή μας "ξεχνάει" τον υπολειπόμενο χρόνο του χρήστη μέχρι εκείνη τη στιγμή, και ξεκινά να χρονομετρά εκ νέου.

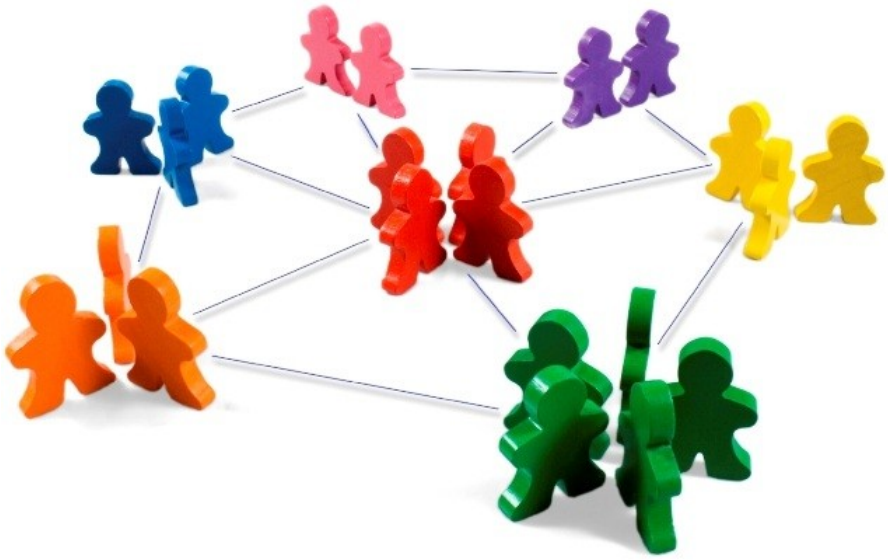
Τέταρτον, η εφαρμογή μας προσφέρει πλήρη ασφάλεια κατά τη σύνδεση του χρήστη και την ανταλλαγή δεδομένων. Η ιστοσελίδα από την οποία μπορεί ο χρήστης να κάνει λήψη όλου του πακέτου αρχείων που χρειάζεται για να συνδεθεί, είναι κάτω από το πρωτόκολλο HTTPS, που είναι επί της ουσίας συνδυασμός του απλού πρωτοκόλλου HTTP με τις δυνατότητες κρυπτογράφησης που μας προσφέρει το πρωτόκολλο Secure Sockets Layer (SSL). Έτσι όλα τα δεδομένα που ανταλλάσσονται σε επίπεδο διαδικτύου παραμένουν προστατευμένα (όπως λόγου χάρη το όνομα και ο κωδικός πρόσβασης που θα δώσει ο χρήστης). Επίσης, όλοι οι κωδικοί πρόσβασης των χρηστών είναι αποθηκευμένοι σε κρυπτογραφημένη μορφή σε μια βάση δεδομένων στον server, σύμφωνα με τη μέθοδο κρυπτογράφησης PBKDF2 (Password-based key Derivation Function 2). Η μέθοδος κρυπτογράφησης PBKDF2 χρησιμοποιείται κατά κόρον σε πολλές εφαρμογές που αποθηκεύουν ευαίσθητα δεδομένα χρηστών, και το πραγματοποιεί με μεγάλη ασφάλεια, χρησιμοποιώντας μεταξύ άλλων και τη μέυοδο salting (αλάτισμα). Η μέθοδος PBKDF2 θα περιγραφεί αναλυτικά στη συνέχεια της παρούσας διατριβής. Φυσικά, όπως προαναφέραμε, η σύνδεση του χρήστη στην εφαρμογή μας γίνεται μέσω Virtual Private Network, και συγκεκριμένα μέσω του πανίσχυρου και ασφαλούς λογισμικού OpenVPN. Ο συνδυασμός αυθεντικοποίησης του χρήστη και κρυπτογράφησης των πακέτων που ανταλλάσσονται δίνει στο κανάλι επικοινωνίας απόλυτη ασφάλεια.

Το εκτελέσιμο αρχείο που θα λάβει ο χρήστης από την ιστοσελίδα της εφαρμογής, είναι αυτό που αναλαμβάνει να παραλάβει τα πιστοποιητικά που χρειάζονται για τη σύνδεσή του στο VPN δίκτυο, τα οποία, μετά τη σύνδεση, σβήνονται τοπικά από το σκληρό του χρήστη, για να μειώσουμε έτσι τις πιθανότητες να μείνουν εκεί για μεγάλο χρονικό διάστημα και άρα να αποκτήσει πρόσβαση σε αυτά κάποιος. Όπως και να' χει, μετά το πέρας του χρόνου που έχει αιτηθεί ο χρήστης, τα πιστοποιητικά αυτά είναι άχρηστα, άρα δεν υπάρχει κίνδυνος αν σε οποιαδήποτε περίπτωση πέσουν στα χέρια κάποιου κακόβουλου ατόμου.

Πέμπτον, είναι προσαρμόσιμη, έχει χρηστικότητα ανάλογα με τις ανάγκες της εκάστοτε κοινότητας. Εξηγήσαμε και παραπάνω τους λόγους για τους οποίους οι εφαρμογές κοινότητας μεγαλώνουν και αναζητούνται συνεχώς όλο και περισσότερο στο ευρύ κοινό. Μαζί με την ασφάλεια και τη χρονομέτρηση των χρηστών, εμείς έχουμε προσδώσει και στην εφαρμογή μας ένα σύνολο από υπηρεσίες και λειτουργίες που μπορούν να χρησιμοποιηθούν ευρέως από διαφορετικής μορφής κοινότητες και να εξυπηρετήσουν διαφορετικές ανάγκες χρηστών. Οι υπηρεσίες που προσφέρουμε είναι: σύστημα διαμοιρασμού αρχείων μέσω FTP, η πρόσβαση σε ένα wiki, η πρόσβαση σε ένα μέσο

κοινωνικής δικτύωσης, η πρόσβαση σε μία βιβλιοθήκη αναπαραγωγής πολυμέσων και βίντεο, υπηρεσία ζωντανής συζήτησης μεταξύ των χρηστών (live chatting), η από κοινού και ταυτόχρονη, συνεργατική επεξεργασία αρχείων κειμένου (etherpad) και άλλες. **Πιστεύουμε πως η κάθε μία ξεχωριστά από τις παραπάνω υπηρεσίες βρίσκει τεράστια χρήση στο ευρύ κοινό όλο και περισσότερο καθημερινά, όμως δεν υπάρχει μέχρι στιγμής στην αγορά κάποια εφαρμογή που να συνδυάζει όλες αυτές μαζί. Αυτό το χαρακτηριστικό θεωρούμε ότι προσδίδει καινοτομία στη δική μας εφαρμογή.**

Έκτον, η εφαρμογή μας έχει μεγάλη επεκτασιμότητα. Όλες οι πρόσθετες υπηρεσίες της εφαρμογής μας, προσφέρονται στον χρήστη ως απλές Web εφαρμογές, απλές ιστοσελίδες προσφερόμενες εντός του ιδιωτικού δικτύου. Αυτό σημαίνει πως, εάν το απαιτούν οι εκάστοτε ανάγκες, με πολύ εύκολο τρόπο μπορούν να προσθαφαιρεθούν επιπλέον Web υπηρεσίες στον server της εφαρμογής.



Εικόνα 1: Η ανάγκη για virtual communities ολοένα και αυξάνεται με την πρόοδο της τεχνολογίας

ε3) παραδείγματα και ιδέες χρήσης της εφαρμογής στον πραγματικό κόσμο

Χρήση σε επίπεδο εταιρείας

Η εφαρμογή μας θα μπορούσε να αναπτύξει όλη της τη χρηστικότητα σε μία εταιρεία. Η εφαρμογή εγκαθίσταται στους servers της εταιρείας, και οι εργαζόμενοι μπορούν είτε μέσα από το δίκτυο της εταιρείας είτε απομακρυσμένα με χρήση του διαδικτύου να συνδεθούν σε αυτή. Ο απόλυτα αυτοματοποιημένος τρόπος της σύνδεσης, δε μας περιορίζει καθόλου ως προς τις τεχνικές γνώσεις του εργαζόμενου σε σχέση με αυτήν. Άρα, συνεπώς, και το αντικείμενο της εταιρείας δεν περιορίζεται μόνο γύρω από αυτό της πληροφορικής και των δικτύων, αλλά μπορεί πραγματικά να βρει εφαρμογή σε οποιαδήποτε φύση εταιρείας.

Η ασφάλεια που προσδίδεται στην επικοινωνία προστατεύει τα ιδιωτικά ή απόρρητα δεδομένα που μπορεί να έχει η εταιρεία εντός της. Οι ποικίλες λειτουργίες της εφαρμογής μας μπορούν να χρησιμοποιηθούν από τους χρήστες από απλό σκοπό κοινωνικοποίησης (όπως το μέσο κοινωνικής δικτύωσης που προσφέρεται) μέχρι και επαγγελματικό σκοπό. Ο διαμοιρασμός αρχείων είναι κάτι που αποτελεί αντικείμενο προς επίλυση διαρκώς, σε όλες τις εταιρείες, πόσο μάλλον όταν αυτά πρέπει να επεξεργάζονται ταυτόχρονα από πολλούς απομακρυσμένους χρήστες. Και σε αυτό το θέμα μπορεί η εταιρεία στην εφαρμογή μας να βρει εύκολη λύση μέσω του wiki, ή της συνεργατικής ταυτόχρονης επεξεργασίας αρχείων μέσω etherpad. Η προβολή και ο διαμοιρασμός βίντεο για εκπαίδευση των νεοπροσληφθέντων εργαζόμενων είναι πρακτική που ακολουθούν πολλές και μεγάλες εταιρείες, αντί των

σεμιναρίων που απαιτούν φυσική παρουσία, κι εδώ θα μπορούσε η εφαρμογή μας να αποδειχθεί πολύ βοηθητική. Σε αυτό το σκοπό μπορεί να αποβεί χρήσιμο και το wiki. Επίσης, τη σήμερον ημέρα δεν υπάρχει εταιρεία που να μη χρησιμοποιεί κάποια εφαρμογή συζήτησης chatting μεταξύ των εργαζόμενων. Κι αυτό είναι κάτι που το προσφέρει η δική μας εφαρμογή.

Φυσικά η κάθε εταιρεία θα μπορούσε να εκμεταλλευτεί τη δυνατότητα επεκτασιμότητας της εφαρμογής, εγκαθιστώντας παραπάνω υπηρεσίες και λειτουργίες στο εσωτερικό του server της εφαρμογής, "πίσω" από το VPN δίκτυο.

Χρήση σε επίπεδο κοινότητας χρηστών με κοινά ενδιαφέροντα

Ο τρόπος με τον οποίο θα επικοινωνούν μικρές ή μεγάλες κοινότητες χρηστών μεταξύ τους, για να μοιράζονται τα ίδια ενδιαφέροντα, είναι από τα βασικά προβλήματα που καλούνται να επιλύσουν πλήθος εφαρμογών, ελεύθερων ή μη, στο διαδίκτυο. Μέσα κοινωνικής δικτύωσης (social media), φόρουμ διαλόγου γενικού περιεχομένου ή με βάση κάποια θεματική, πολλά λογισμικά διαμοιρασμού αρχείων ή γενικώς πληροφορίας, και πολλά άλλα υπάρχουν στην αγορά και χρησιμοποιούνται κατά κόρον. Η δική μας εφαρμογή επιτυγχάνει να μπορεί να συνδυάζει όλα τα παραπάνω, προσθέτοντας και το μεγάλο πλεονέκτημα της ασφάλειας και της ιδιωτικότητας. Έτσι, αν σε μία κοινότητα είναι σκοπός απλά και μόνο η επικοινωνία μεταξύ των χρηστών, τότε αρκεί η σύνδεσή τους και η χρήση του social media και του chat. Αν απαιτούνται παραπάνω πράγματα, τότε ο διαμοιρασμός ή η από κοινού επεξεργασία αρχείων και wiki, η προβολή βίντεο, είναι πράγματα που διατίθενται προς χρήση.

Δεν περιοριζόμαστε και πάλι, ούτε από τις τεχνικές γνώσεις του κάθε χρήστη, αλλά ούτε και από το αντικείμενο που θέλει να ασχοληθεί η κοινότητα. Από μια παρέα απομακρυσμένων φίλων μέχρι και μία αυστηρή θεματική, όλα μπορούν να υποστηριχθούν.

Χρήση για εκπαιδευτική πλατφόρμα

Βλέπουμε συνεχώς στην καθημερινή ζωή, όλο και περισσότεροι άνθρωποι να επιλέγουν να εκπαιδευτούν από ηλεκτρονικά σεμινάρια, σε αντίθεση με σεμινάρια που απαιτούν φυσική παρουσία. Για να καλυφθεί αυτή η ανάγκη, πολλές είναι οι ηλεκτρονικές πλατφόρμες εκπαίδευσης που έχουν αναπτυχθεί. Η λειτουργία προβολής βίντεο που διαθέτει η εφαρμογή μας θα ήταν ιδανική για μια τέτοια δουλειά, σε συνδυασμό με το wiki που θα μπορούσε να παίζει τον ρόλο των σημειώσεων του μαθήματος.

Φυσικά, το μέσο κοινωνικής δικτύωσης που προσφέρουμε, σε συνδυασμό με το διαμοιρασμό αρχείων θα μπορούσε εύκολα να αποτελέσει και τον τρόπο ανταλλαγής σημειώσεων και πληροφοριών για τα σεμινάρια μεταξύ των εκπαιδευόμενων. Επειδή συνήθως αυτά τα ηλεκτρονικά σεμινάρια έχουν χρέωση με την ώρα ή ανά βίντεο, το στοιχείο της εφαρμογής μας, που θα βοηθούσε πάρα πολύ σε αυτή την πλατφόρμα εκπαίδευσης, είναι η χρονομέτρηση του χρήστη όσο αυτός βρίσκεται online. Ο χρήστης δηλαδή να γνωρίζει από πριν ακριβώς το χρόνο που διαρκεί το σεμινάριο που πρόκειται να παρακολουθήσει, οπότε να εγγράφεται για τη συγκεκριμένη ώρα στην εφαρμογή, χωρίς να έχει πρόσβαση αργότερα.

στ) Περιγραφή του σκελετού της παρούσας μεταπτυχιακής διατριβής

Στην παρούσα μεταπτυχιακή διατριβή, προσπαθούμε να αναλύσουμε με λεπτομέρεια όλα τα βήματα που ακολουθήσαμε για την εκπόνηση και τη δημιουργία αυτής της εφαρμογής που περιγράψαμε παραπάνω, στο παρόν κεφάλαιο.

Στο κεφάλαιο 2 της παρούσας διατριβής, καταπιανόμαστε με την αρχιτεκτονική των εικονικών δικτύων VPN, αναλύουμε ποια είναι τα πλεονεκτήματά τους και οι ανάγκη χρήστης τους. Αναλύουμε ποια είναι τα διασημότερα πρωτόκολλα για επικοινωνία μέσω VPN, και περιγράφουμε το λογισμικό OpenVPN. Στη συνέχεια, αναλύουμε όλες τις γνωστές μεθόδους κρυπτογράφησης και εν τέλει, πώς ακριβώς συνδυάσαμε όλα τα παραπάνω για τις ανάγκες της δικιάς μας εφαρμογής, κάνοντας μια αναλυτική περιγραφή της εγκατάστασης και των ρυθμίσεών μας.

Στο κεφάλαιο 3, αναλύουμε εξ' ολοκλήρου όλο το web κομμάτι της εφαρμογής μας. Περιγράφουμε το Flask Web Framework, το πλαίσιο εργασίας που χρησιμοποιήσαμε για την ιστοσελίδα της εφαρμογής, και στη συνέχεια περνάμε στο ειδικό κεφάλαιο που περιγράφει πώς αποθηκεύουμε τα ευαίσθητα

δεδομένα των χρηστών με ασφάλεια. Προχωρώντας παρακάτω, αναλύεται με λεπτομέρεια ο μηχανισμός χρονισμού για τη συνδεσιμότητα του εκάστοτε χρήστη, καθώς και λεπτομέρειες σχετικά με την ανάκληση πιστοποιητικού που πραγματοποιείται στα πλαίσια του OpenVPN, όταν τελειώσει ο χρόνος του χρήστη.

Στο κεφάλαιο 4, περνάμε πλέον στη μεριά του χρήστη, αναλύοντας όλες τις ενέργειες που πραγματοποιούνται από το εκτελέσιμο script της εφαρμογής που θα κάνει λήψη ο χρήστης. Περιγράφεται τι διαφορετικό συμβαίνει σε κάθε λειτουργικό σύστημα ξεχωριστά, πώς πραγματοποιείται η σύνδεση λεπτομερώς, τι συμβαίνει με τον πίνακα δρομολόγησης όταν συνδεόμαστε με VPN.

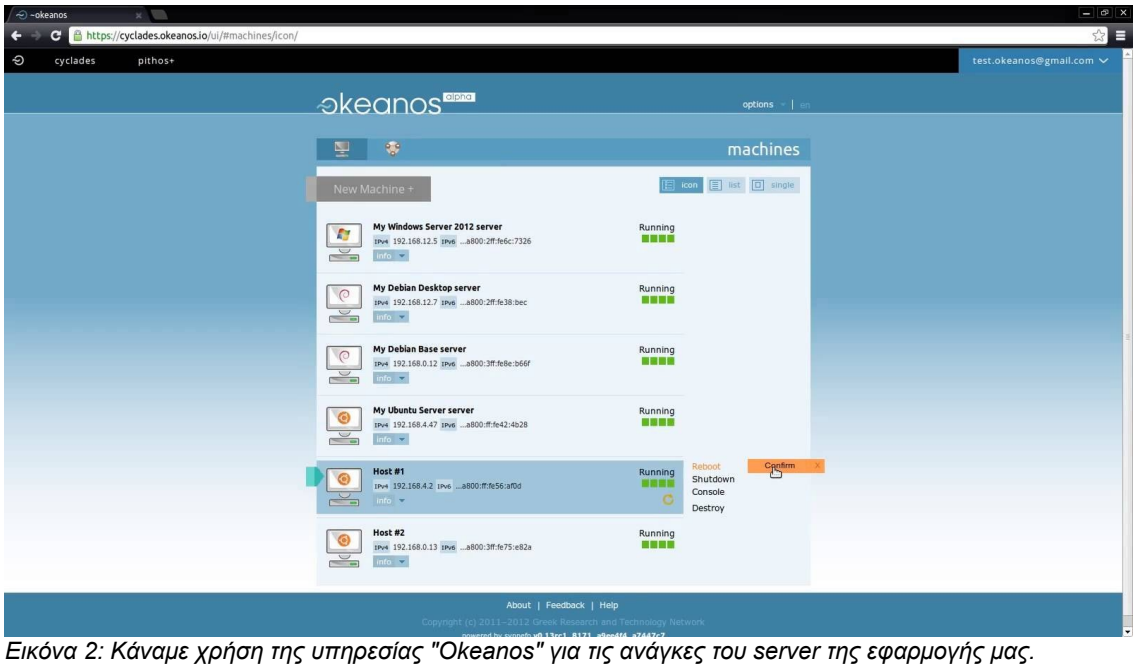
Στο κεφάλαιο 5 περιγράφουμε αναλυτικά τις ενέργειες που πραγματοποιήσαμε στο server της εφαρμογής έτσι ώστε να μπορεί να εξυπηρετήσεις τις διαθέσιμες υπηρεσίες. Στη συνέχεια προβαίνουμε σε μια σύντομη περιγραφή καθεμίας από αυτές.

Στο κεφάλαιο 6 κάνουμε μία ανάλυση για το ποια μπορεί να είναι τα μειονεκτήματα της εφαρμογής, και προτείνουμε λύσεις σε αυτά. Επίσης το κεφάλαιο αυτό περιλαμβάνει ξεχωριστή αναφορά σε όλες τις πιθανές βελτιώσεις που μπορούν να συμπεριληφθούν στην παρούσα εφαρμογή στο μέλλον.

Στο κεφάλαιο 7 παρατίθεται αναλυτικά η βιβλιογραφία που συμβουλευθήκαμε για την εκπόνηση της παρούσας μεταπτυχιακής διατριβής.

ζ) Εγκατάσταση της εφαρμογής

Την εφαρμογή μας την εγκαταστήσαμε σε μία εικονική μηχανή (virtual machine) που μας παρέχει η διαδικτυακή υπηρεσία Okeanos, με εντός του το λειτουργικό Ubuntu Server 14.04. Η αξιοπιστία του Ubuntu Linux ως ένα λειτουργικό σύστημα πολύ σταθερό, σε συνδυασμό με την διαθεσιμότητα σε αυτό όλων των εφαρμογών και προγραμμάτων που θέλουμε να έχουμε στη διάθεσή μας, μας οδήγησε σε αυτή την επιλογή.



Εικόνα 2: Κάναμε χρήση της υπηρεσίας "Okeanos" για τις ανάγκες του server της εφαρμογής μας.

ΚΕΦΑΛΑΙΟ 2

Ασφαλής επικοινωνία και VPN

Καθώς η δημοτικότητα του διαδικτύου μεγάλωνε, όλο και περισσότερο υπήρχε η αναζήτηση για γρήγορη, ασφαλή και έγκυρη επικοινωνία μεταξύ των χρηστών, όσο απομακρυσμένοι κι ήταν. Για να καλυφθεί λοιπόν αυτή η ανάγκη, άρχισαν να δημιουργούνται τα πρώτα εικονικά δίκτυα, τα Virtual Private Networks (VPN). **Ο ορισμός ενός Virtual Private Network είναι ο εξής:**

Ένα Virtual Private Network είναι ένα ιδιωτικό δίκτυο, το οποίο χρησιμοποιεί ένα πιο ευρύ (δημόσιο) δίκτυο, όπως είναι το διαδίκτυο, προκειμένου να επικοινωνεί με αποκλειστικότητα και ιδιωτικότητα. Σε αντίθεση με τη χρήση ενός μοναδικά αφιερωμένου καναλιού για αυτό το σκοπό, όπως είναι η μισθωτή γραμμή (leased line), το VPN χρησιμοποιεί αποκλειστικά εικονικές συνδέσεις δρομολογημένες "εντός" του διαδικτύου, μεταξύ δύο απομακρυσμένων οντοτήτων.

Τα πλεονεκτήματα των VPN δικτύων:

Χαμηλό κόστος: Οι μισθωμένες γραμμές (leased lines) που αναφέραμε παραπάνω, καθώς και άλλων τύπων δομές αποκλειστικών ιδιωτικών δικτύων, έχουν μεγάλο κόστος στην κατασκευή τους (δημιουργούνται μόνιμα εικονικά κυκλώματα - Permanent Virtual Circuits) και στην πάγια χρέωση. Ειδικά στην περίπτωση που μιλάμε για μεγάλες αποστάσεις, το κόστος αυτών των γραμμών ανεβαίνει εκθετικά. Αντίθετα, για την εγκατάσταση ενός απλού VPN, απαιτείται πολύ λιγότερο έως μηδαμινό κόστος, καθώς κάνει χρήση των ήδη υπάρχοντων πόρων του δικτύου (όπως με το διαδίκτυο).

Προσαρμοστικότητα: Στα παραδοσιακά ιδιωτικά δίκτυα, πρέπει να υπάρχει συμβατή υποδομή και εξοπλισμός που υποστηρίζει όλους τους κόμβους και τις οντότητες επικονωνίας. Στα VPNs ο περιορισμός αυτός μπορεί να αποφευχθεί εύκολα, ειδικά αν για την εγκατάσταση ενός VPN χρησιμοποιείται απλώς λογισμικό. Υπάρχουν πάρα πολλά λογισμικά για εγκατάσταση VPNs, συμβατά με όλα τα λειτουργικά συστήματα και τους τύπους υπολογιστών.

Επεκτασιμότητα: Η χρήστη του διαδικτύου προσφέρει απεριόριστη γεωγραφική επέκταση. Στο ίδιο VPN δίκτυο μπορούν εύκολα να συνδεθούν πάμπολλοι χρήστες. Αυτό είναι που κάνει τις συνδέσεις αυτές εύκολα αναβαθμίσιμες ανάλογα με τις απαιτήσεις.

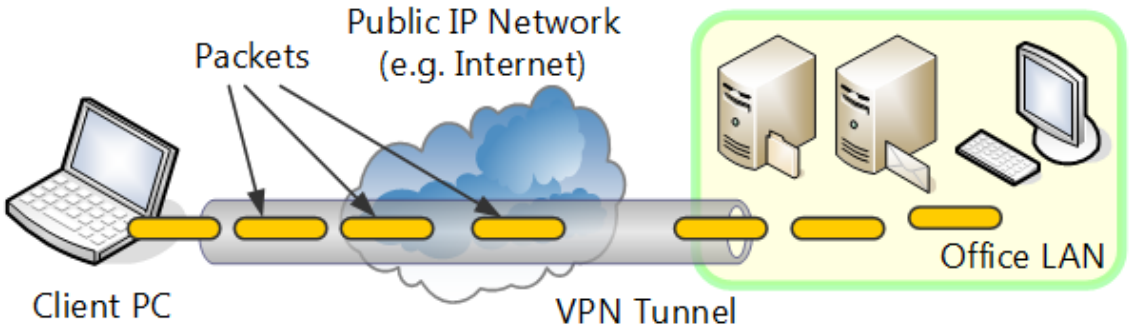
Ασφάλεια: Ίσως το σημαντικότερο στοιχείο για τη λειτουργία ενός VPN. Τα VPN παρέχουν αυξημένη ασφάλεια λόγω των πρωτοκόλλων tunneling και κρυπτογράφησης που χρησιμοποιούνται. Επιπλέον, σε συνδυασμό με άλλα γνωστά μέτρα ασφαλείας που χρησιμοποιούνται στην πλειοψηφία των περιπτώσεων, όπως πχ Firewalls, επιτυγχάνεται ακόμα μεγαλύτερη αποτελεσματικότητα στην ασφάλεια. Το πρωτόκολλο IPSec (Internet Protocol Security) παρέχει βελτιωμένες μεθόδους ασφαλείας, όπως για παράδειγμα καλύτερους αλγορίθμους κρυπτογράφησης και πιο αποτελεσματική πιστοποίηση και αυθεντικοποίηση χρηστών.

Εύκολη και βολική διαχείρηση: Από ένα σημείο, αυτό του διαχειριστή (administrator), μπορεί να γίνει εύκολη και συγκεντρωτική διαχείριση του δικτύου. Από αυτό το σημείο ελέγχονται όλες οι διευθύνσεις IP, πολιτικές πρόσβασης χρηστών, τρόπος ασφάλισης και άλλες συναφείς εργασίες. Η εύκολη προσαρμοστικότητα όλων των παραπάνω μας οδήγησαν και στην αποτελεσματική διαχείριση και της δικής μας εφαρμογής.

Τι είναι το tunneling

Tunneling: Με τον όρο tunneling εννούμε τη μετάδοση δεδομένων μεταξύ δύο όμοιων ή ανόμοιων δικτύων ή μέσω ενός δικτύου "διαμεσολαβητή". Η μέθοδος tunneling περικλείει έναν τύπο πακέτου εντός των πακέτων ενός άλλου, "ξένου" πρωτόκολλου (όπως το IP). Αυτή η διαδικασία λέγεται encapsulation (ενθυλάκωση). Πριν συμβεί το encapsulation, τα πακέτα κρυπτογραφούνται έτσι ώστε τα δεδομένα να είναι αδύνατο να διαβαστούν σε οποιονδήποτε "παρακολουθεί" το δίκτυο και πέσουν στα χέρια του. Αυτά τα encapsulated πακέτα ταξιδεύουν μέσα στο διαδίκτυο, το οποίο διαδίκτυο παίζει το ρόλο του "μεσάζοντα" μέχρι να φτάσουν στον προορισμό τους. Από τη στιγμή που φτάσουν, τα πακέτα αποκρυπτογραφούνται και επιστρέφουν στην αρχική τους μορφή. Το πρωτόκολλο που ενθυλακώνει τα πακέτα είναι γνωστό από το δίκτυο και στον αποστολέα και στον αποδέκτη. Έτσι μπορούν και

στέλνονται τα πακέτα με ασφάλεια εντός του διαδικτύου. Σε όλες τις περιπτώσεις χρησιμοποιούνται ιδιωτικές, μη δρομολογούμενες, IP διευθύνσεις εντός ενός πακέτου που χρησιμοποιεί δημοσίως μια γνωστή διεύθυνση, για να δημιουργήσει το tunnel του ιδιωτικού δικτύου.



Εικόνα 3: Ένα VPN tunnel.

VPN protocols

Φυσικά για την όλη διαδικασία του VPN και του tunneling δεν υπάρχει αποκλειστικά ένα πρωτόκολλο, αλλά ποικίλα που προσφέρουν διάφορες δυνατότητες. Φυσικά το κάθε πρωτόκολλο έχει τα δικά του πλεονεκτήματα και μειονεκτήματα, όπως θα δούμε παρακάτω:

PPTP - Point to Point Tunneling Protocol

Σχεδιάστηκε και αναπτύχθηκε από τη Microsoft τη δεκαετία του 1990. Για εκείνη την εποχή ήταν αρκετά ασφαλές και πολλές επιχειρήσεις το χρησιμοποιούσαν ως πρότυπο. Το PPTP είναι διαθέσιμο σε οποιαδήποτε συσκευή συμβατή με VPN τεχνολογία, και ως εκ τούτου είναι εύκολο να ρυθμιστεί και να εφαρμοστεί γιατί δε χρειάζεται επιπλέον λογισμικό για να τρέξει. Το μεγαλύτερο πλεονέκτημα του PPTP έναντι των άλλων πρωτοκόλλων είναι η μεγάλη του ταχήτυτα. Πλέον όμως, το PPTP δεν αποτελεί και πολύ ασφαλή επιλογή και θεωρείται κάπως ξεπερασμένο για τις σύγχρονες τεχνολογίες δικτύων.

Τα L2TP και L2TP/IPSec - Layer to tunnel protocol

Το Layer to Tunnel Protocol ήταν ένας στενός ανταγωνιστής του PPTP λόγω της εφάμιλλης ταχύτητας και της πολύ καλύτερης ασφάλειας. Το ίδιο δε διέθετε ενσωματωμένο μηχανισμό κρυπτογράφησης πακέτων, αλλά χρησιμοποιούσε το Internet Protocol Security (IPSec) για αυτή τη δουλειά. Επίσης αποτελεί ένα ενσωματωμένο πρωτόκολλο σε όλα τα σύγχρονα λειτουργικά συστήματα, κι έτσι το κάνει εύκολο στη ρύθμιση και τη διαχείριση. Σε αντίθεση με το PPTP, δεν έχει τόσες ευπάθειες, και άρα στο θέμα της ασφάλειας κερδίζει. Το L2TP θεωρείται πολύ ασφαλές και εύκολο στη χρήση. Βασικό του μειονέκτημα είναι πως είναι αρκετά πιο αργό από κάποια άλλα VPN πρωτόκολλα, καθώς επίσης “δυσκολεύεται” να λειτουργήσει πίσω από ισχυρά firewalls.

OpenVPN

Το OpenVPN είναι ένα από τα πιο διάσημα VPN πρότυπα και πρωτόκολλα, και χρησιμοποιείται από σχεδόν όλους τους VPN παρόχους στον κόσμο. Είναι μια πιο σύγχρονη VPN τεχνολογία και χρησιμοποιεί έναν συνδυασμό από τεχνολογίες όπως SSLv3 και OpenSSL για να προσδώσει την καλύτερη δυνατή VPN υπηρεσία. Η OpenSSL βιβλιοθήκη προσφέρει κρυπτογράφηση σύμφωνα με πολλούς διαφορετικούς αλγόριθμους όπως οι AES, Camellia και Blowfish. Τα πλεονεκτήματα του OpenVPN περιλαμβάνουν ευρεία χρηστικότητα και επιλογές στη ρύθμιση, υψηλή ασφάλεια, υποστήριξη

πολλών αλγορίθμων κρυπτογράφησης, πολύ καλή απόδοση και ταχύτητα. Επίσης, είναι ανοιχτού κώδικα. Το βασικό μειονέκτημα του OpenVPN είναι ότι πρώτον, χρειάζεται κάποιο 3rd party λογισμικό για να λειτουργήσει, δεν είναι δηλαδή ενσωματωμένο στα ήδη υπάρχοντα λειτουργικά συστήματα, και δεύτερον, απαιτεί αντίστοιχες τεχνικές γνώσεις για την εγκατάστασή του.

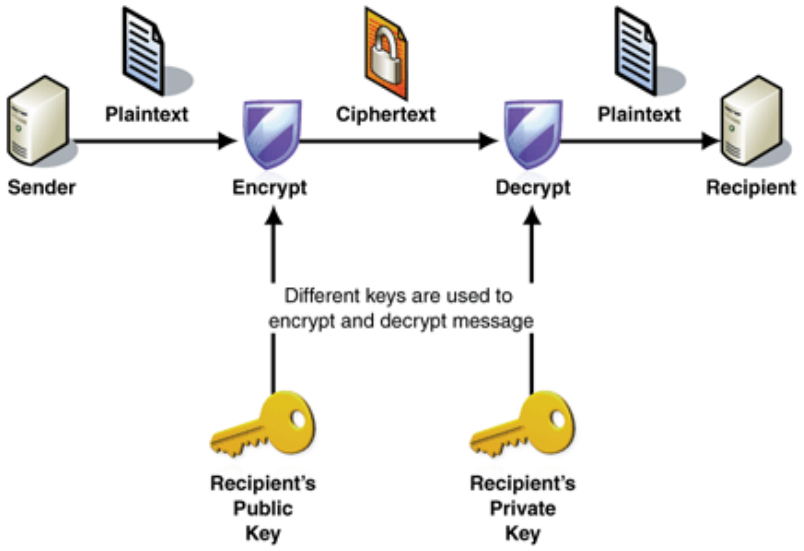
Στην παρούσα μεταπτυχιακή διατριβή επιλέξαμε το πρωτόκολλο OpenVPN για σύνδεση και επικοινωνία με την εφαρμογή μας, έτσι ώστε να μπορέσουμε να εκμεταλλευτούμε όλα του τα πλεονεκτήματα, ενώ ταυτόχρονα ο κύριος σκοπός αυτής της διατριβής είναι να εκμηδενίσουμε τα μειονεκτήματά του, αυτοματοποιώντας απόλυτα την εγκατάσταση και τη χρήση του OpenVPN.

Πολιτικές ασφαλείας και κρυπτογράφησης των VPN

Όταν λέμε πως θέλουμε να ασφαλίσουμε μια VPN σύνδεση, εννοούμε: να προσδώσουμε ιδιωτικότητα σε όλα τα δεδομένα που μεταδίδονται, να διασφαλίσουμε την ακεραιότητα των δεδομένων (να μη φτάσουν αλλοιωμένα στον προορισμό τους), και να βεβαιωθούμε ότι τα δεδομένα θα είναι σίγουρα διαθέσιμα όταν ζητηθούν.

Οι τρόποι κρυπτογράφησης που πληρούν αυτές τις προϋποθέσεις είναι δύο, η συμμετρική και η μη συμμετρική κρυπτογράφηση.

1. **Συμμετρική κρυπτογράφηση.** Με αυτή τη μέθοδο, ο αποστολέας και ο παραλήπτης μοιράζονται το ίδιο κλειδί από κοινού για να κρυπτογραφήσουν και να αποκρυπτογραφήσουν το μήνυμα. Οποιοσδήποτε έχει πρόσβαση στο κλειδί, μπορεί να αποκρυπτογραφήσει τη συνομιλία. Αν το κλειδί "εκτεθεί" δημοσίως, τότε ταυτόχρονα μένει εκτεθειμένο και το VPN δίκτυο. **Αυτό καθιστά τη συμμετρική κρυπτογράφηση πιο ευάλωτη σε μια σειρά από επιθέσεις,** όπως αυτές του τύπου "brute force attack" (επίθεση που επαναλαμβανόμενα δοκιμάζει χαρακτήρες στη θέση του κλειδιού μέχρι να αποκρυπτογραφήσει το μήνυμα), και επιθέσεις τύπου "man in the middle" (ο επιτιθέμενος κακόβουλος χρήστης βρίσκεται "ενδιάμεσα" σε αποστολέα και παραλήπτη, αντιγράφοντας και παραλλάζοντας το μήνυμα χωρίς κανείς από τους δύο να το καταλάβει). Τα συστήματα που χρησιμοποιούν συμμετρική κρυπτογράφηση πρέπει να αλλάζουν το κλειδί συχνά και επίσης να χρησιμοποιούν έναν συνδυασμό τιμής διάρκειας ζωής κλειδιού και μήκους κλειδιού για να "δυσκολέψουν" τον επιτιθέμενο.
2. **Ασύμμετρη κρυπτογράφηση.** Στην ασύμμετρη κρυπτογράφηση, κάθε χρήστης έχει ένα ιδιωτικό και ένα δημόσιο κλειδί (private και public key). Το δημόσιο κλειδί είναι γνωστό σε όλους, και χρησιμοποιείται για να κρυπτογραφηθεί το μήνυμα, και από τη στιγμή που τα δύο αυτά κλειδιά είναι συσχετιζόμενα μέσω ενός μαθηματικού αλγορίθμου, μόνο το ιδιωτικό κλειδί του παραλήπτη μπορεί να αποκρυπτογραφήσει το μήνυμα. Σε αυτό το σύστημα ασύμμετρης κρυπτογράφησης, οι χρήστες πρέπει να είναι βέβαιοι ότι το ιδιωτικό τους κλειδί δεν είναι προσβάσιμο σε κανέναν πλην του εαυτού τους.



Εικόνα 4: Η μέθοδος ασύμμετρης κρυπτογράφησης με χρήση ιδιωτικού και δημόσιου κλειδιού

Άλλες γνωστές μέθοδοι κρυπτογράφησης που χρησιμοποιούνται πρόσθετα:

Συνάρτηση κατακερματισμού (hash function). Η συνάρτηση κατακερματισμού λαμβάνει ως είσοδο ένα σύνολο δεδομένων και επιστρέφει μία συμβολοσειρά σταθερού μεγέθους σε bit, η γνωστή και ως τιμή κατακερματισμού (hash value), έτσι ώστε και η παραμικρή αλλαγή (τυχαία ή εκ προθέσεως) στα δεδομένα θα αλλάξει την τιμή κατακερματισμού. Η ιδανική συνάρτηση κατακερματισμού έχει τέσσερις βασικές παραμέτρους: ο υπολογισμός της τιμής κατακερματισμού είναι εύκολος και έχει σχετικά μικρές απαιτήσεις σε επεξεργαστική ισχύ για οποιονδήποτε τύπο δεδομένων, είναι εξαιρετικά δύσκολο (έως αδύνατο) να πετύχουμε με δοκιμές μια δεδομένη τιμή κατακερματισμού, είναι εξαιρετικά δύσκολο να τροποποιήσουμε ένα σύνολο δεδομένων χωρίς να ακολουθήσουν αλλαγές και στην τιμή κατακερματισμού, και είναι εξαιρετικά απίθανο τα δύο διαφορετικά μηνύματα να έχουν την ίδια τιμή κατακερματισμού. Οι συναρτήσεις κατακερματισμού χρησιμοποιούνται για έλεγχο της ακεραιότητας των δεδομένων, βεβαιώνοντας πως τα δεδομένα μας δεν έχουν απώλειες. Επίσης χρησιμοποιούνται για ψηφιακές υπογραφές, όπως αυτές της ασύμμετρης κρυπτογράφησης μεταξύ των κλειδιών, για να διασφαλιστεί η πιστοποίηση ταυτότητας, και σε άλλες διάφορες εφαρμογές ασφάλειας μετάδοσης πληροφορίας.

Η ανταλλαγή κλειδιών με μέθοδο Diffie-Hellman (Diffie-Hellman key exchange). Το σύστημα κρυπτογράφησης Diffie-Hellman αναπτύχθηκε το 1976 για να λύσει το πρόβλημα της ανταλλαγής των κλειδιών μεταξύ δύο χρηστών. Αυτό το πρωτόκολλο επιτρέπει σε δυο οντότητες, χωρίς προηγούμενη επικοινωνία, να ανταλλάξουν ένα κοινό κλειδί μέσω ενός μη ασφαλούς διαύλου επικοινωνίας.

Η χρήση του OpenVPN στη δική μας εφαρμογή

Το πρώτο βήμα όταν κάποιος θέλει να χτίσει μια OpenVPN εφαρμογή, είναι να καθιερώσει μία Υποδομή Δημοσίου Κλειδιού (PKI - Public Key Infrastructure). Αυτή η υποδομή περιέχει τα εξής:

- Από ένα ξεχωριστό δημόσιο και ιδιωτικό κλειδί για τον server και για κάθε client (εφεξής θα το αποκαλούμε πιστοποιητικό, με αυτή την ορολογία είναι ευρέως γνωστά τα δημόσια κλειδιά των χρηστών)

- Τη δημιουργία μιας αρχής πιστοποίησης (CA - certificate authority) με το πιστοποιητικό της και το ιδιωτικό της κλειδί, που χρησιμοποιείται για να υπογράφει όλα τα πιστοποιητικά των χρηστών.

Το λογισμικό OpenVPN υποστηρίζει αμφίδρομη πιστοποίηση βασισμένη στα πιστοποιητικά, πράγμα που σημαίνει ότι ο client πρέπει να αναγνωρίσει ως αυθεντικό το πιστοποιητικό του server και αντίστροφα, πριν καθιερωθεί η αμοιβαία εμπιστοσύνη μεταξύ των δύο και αρχίσουν να επικοινωνούν.

Ο client και ο server αυθεντικοποιούν ο ένας τον άλλο πρώτον βεβαιώνοντας ότι το πιστοποιητικό που τους παρουσιάζεται έχει την υπογραφή της Αρχής Πιστοποίησης, και ύστερα βλέποντας την πληροφορία που βρίσκεται στην επικεφαλίδα (header) του πιστοποιητικού, όπως το κοινό όνομα του πιστοποιητικού και ο τύπος του.

Πολλά από τα χαρακτηριστικά ασφαλείας που ακολουθεί το λογισμικό OpenVPN έχουν υιοθετηθεί από τις αρχές ασφαλείας των VPN δικτύων. Κάποια από αυτά είναι:

- Ο server χρειάζεται μόνο τα δικά του πιστοποιητικά και κλειδιά. Δε χρειάζεται να γνωρίζει ένα-ένα τα πιστοποιητικά κάθε client ξεχωριστά που πρόκειται να συνδεθεί σε αυτόν.
- Ο server θα αποδέχεται μόνο τα πιστοποιητικά που φέρουν την υπογραφή της Αρχής Πιστοποίησης. Και επειδή ο server μπορεί να πραγματοποιήσει τη βεβαίωση της υπογραφής χωρίς να έχει πρόσβαση στο ιδιωτικό κλειδί της Αρχής Πιστοποίησης (το πιο “ευαίσθητο” κλειδί σε όλη αυτή την επικοινωνία), μπορούμε να κρατήσουμε αυτό το κλειδί κάπου μακριά, ενδεχομένως σε κάποιον υπολογιστή που δεν έχει καν πρόσβαση στο διαδίκτυο, για λόγους ασφαλείας.
- Αν κάποιο ιδιωτικό κλειδί εκτεθεί, τότε μπορεί να “ακυρωθεί” με την πρόσθεσή του στην Λίστα Ανάκλησης Πιστοποιητικών (CRL - Certificate Revocation List). Η CRL λίστα επιτρέπει να απορριφθούν συγκεκριμένα πιστοποιητικά χωρίς να απαιτείται η ανοικοδόμηση εκ νέου ολόκληρου του PKI. Στη δική μας εφαρμογή κάνουμε χρήση της λίστας αυτής για να προσθαφαιρούμε χρήστες ανάλογα με το αν έχουν ξοδέψει τον επιτρεπόμενο χρόνο σύνδεσης ή όχι.
- Ο server μπορεί να εφαρμόσει δικαιώματα πρόσβασης σε κάθε client, βασισμένος σε πεδία ενσωματωμένα εντός του πιστοποιητικού του, όπως πχ. το κοινό όνομα (common name).

Την εγκατάσταση όλης της παραπάνω υποδομής με όλες της τις λειτουργίες, τις πραγματοποιήσαμε χρησιμοποιώντας το εργαλείο easy-rsa, το οποίο “συνοδεύει” όλες τις τελευταίες εκδόσεις του OpenVPN, όντας ενσωματωμένο. Η δημιουργία αρχής πιστοποίησης, η δημιουργία και υπογραφή κλειδιών για τον server και τον client, η ανάκληση πιστοποιητικών όπως θα δούμε παρακάτω, όλα αυτά θα έπρεπε να εκτελεστούν από μια σωρεία μεγάλου μήκους εντολών του OpenSSL και άλλων μηχανισμών πιστοποίησης. Το easy-rsa είναι ένα πολύ εύχρηστο εργαλείο, που μας προσφέρει όλες τις παραπάνω λειτουργίες με πολύ έξυπνο και εύκολο τρόπο, κάνοντας ακριβώς τα ίδια και έχοντας το ίδιο αποτέλεσμα, έχοντας όμως γλυτώσει τις μακρόσυρτες -και επιρρεπείς σε λάθη- εντολές.

Αναλυτική περιγραφή των ρυθμίσεων στο server.conf αρχείο

Το αρχείο server.conf είναι το αρχείο που χρησιμοποιήσαμε για να χειριστούμε σωστά τις ρυθμίσεις από μεριάς server. Αναλυτικά περιγράφεται παρακάτω:

`dev tun`	Ορίζει το όνομα του interface με το οποίο θα συνδέεται ο server με τους επιμέρους clients. Ανάλογα με το πόσοι είναι συνδεδεμένοι,

	ανοίγουν αντίστοιχα τα interfaces tun0, tun1, κλπ.
`proto udp`	Για τη σύνδεση αυτή προτιμούμε το UDP (Universal Datagram Protocol), ένα από τα πιο δημοφιλή πρωτόκολλα για επικοινωνία στο διαδίκτυο.
`port 1194`	Η θύρα δικτύου που θα δεσμεύει η σύνδεσή μας. Η πιο συνηθισμένη (και συνήθως προεπιλεγμένη) για τις VPN συνδέσεις είναι η 1194.
`server 10.8.0.0 255.255.255.0`	Ορίζει τη διεύθυνση του ιδιωτικού VPN δικτύου που θα δημιουργηθεί. Η μάσκα 255.255.255.0 μας υποδεικνύει ότι θα χρησιμοποιηθούν 254 διευθύνσεις για αυτό το σκοπό, από 10.8.0.1 έως και 10.8.0.254. (θα μπορούσε να συμβολιστεί και ως 10.8.0.0/24)
`ca /usr/share/easy-rsa/keys/ca.crt`	Ορίζει το αρχείο πιστοποιητικού (δημόσιου κλειδιού) της αρχής πιστοποίησης.
`cert /usr/share/easy-rsa/keys/vpnserver.crt`	Ορίζει το αρχείο πιστοποιητικού (δημόσιου κλειδιού) του VPN server.
`key /usr/share/easy-rsa/keys/vpnserver.key`	Ορίζει το ιδιωτικό κλειδί του VPN server.
`dh /usr/share/easy-rsa/keys/dh2048.pem`	Ορίζει τις παραμέτρους με τις οποίες θα γίνει η συναλλαγή Diffie - Hellman που εξηγήσαμε παραπάνω. (δημιουργήθηκε με την εντολή "`openssl dhparam -out dh2048.pem 2048`"
`push "dhcp-option DNS 8.8.8.8"`	"Υποχρεώνει" τον client να χρησιμοποιήσει τη συγκεκριμένη διεύθυνση DNS. Επιλέξαμε να του δώσουμε την πιο κοινή, αυτή της Google.
`push "redirect-gateway def1"`	Δρομολογεί όλη την κίνηση (traffic) του client, μέσω του VPN. Συμπεριλαμβάνει και την κίνηση προς τον "έξω κόσμο", δηλαδή στο διαδίκτυο εκτός του ιδιωτικού VPN δικτύου. Σε περίπτωση που δεν είχαμε αυτή τη ρύθμιση, ο client θα επικοινωνούσε μέσω VPN μόνο εντός του δικτύου, δηλαδή με τις διευθύνσεις 10.8.0.*.
`comp-lzo`	Εφαρμόζει τον αλγόριθμο συμπίεσης LZO στα πακέτα που μεταδίδονται. Δίνει βελτίωση στην ταχύτητα λόγω του μικρότερου μεγέθους πακέτου. Από τη στιγμή που χρησιμοποιηθεί στις ρυθμίσεις το server, είναι απαραίτητο να χρησιμοποιηθεί και σε αυτές του client.

`keepalive 10 60`	Ρύθμιση έτσι ώστε να αδρανοποιείται η σύνδεση μετά από ένα ορισμένο διάστημα “ακινησίας”. Αυτή επανέρχεται μόλις δει ξανά αίτημα για επικοινωνία.
`persist-tun`	Σε περίπτωση διακοπής σύνδεσης λόγω ακινησίας το interface tun παραμένει και δεν καταστρέφεται.
`persist-key`	Σε περίπτωση διακοπής σύνδεσης λόγω ακινησίας δε χρειάζεται να ξαναδιαβαστούν τα κλειδιά από την αρχή.
`user panos`	Το όνομα του Linux χρήστη που τρέχει το OpenVPN.
`group panos`	Το όνομα του Linux γκρουπ που ανήκει ο χρήστης.
`log-append /var/log/openvpn.log`	Το log αρχείο στο οποίο θα καταγράφει όλες τις ενέργειές του το OpenVPN.
`verb 3`	Ορίζει πόσο λεπτομερής θα είναι η καταγραφή των ενεργειών του OpenVPN στο log αρχείο. Μπορεί να λάβει αριθμούς από 1 έως 4, με το 4 να δίνει έμφαση στη μεγαλύτερη δυνατή λεπτομέρεια.
`crl-verify /usr/share/easy-rsa/keys/crl.pem`	Για να επιτρέπει τη σύνδεση ή όχι σε κάποιον client, ο server συμβουλεύεται το pem αρχείο το οποίο μας δείχνει κάθε φορά την κατάσταση της λίστας ανάκλησης (Client Revocation List). Κάθε φορά που ενεργοποιείται ή απενεργοποιείται ένας χρήστης, αυτό το αρχείο πρέπει να ανανεώνεται.
`management localhost 7505`	Με αυτή την επιλογή ενεργοποιείται μία θύρα σύνδεσης telnet, που μας δίνει τη δυνατότητα οποιαδήποτε στιγμή να δώσουμε εντολές για επί τόπου ενέργειες στο OpenVPN. Μας χρησιμεύει στην άμεση αποσύνδεση ενός χρήστη όταν τελειώσει ο χρόνος του.

Αυτή ήταν όλη η εγκατάσταση που χρειάστηκε να κάνουμε για το λογισμικό OpenVPN από τη μεριά του server. Όσο θέλουμε να είναι ενεργή η εφαρμογή μας και να δίνει δυνατότητα στους χρήστες να συνδεθούν, το OpenVPN πρέπει να παραμένει ενεργό, γι’ αυτό και επιλέξαμε να τρέχει ασταμάτητα στο λειτουργικό του server ως daemon, δηλαδή ως ένα πρόγραμμα που τρέχει στο “παρασκήνιο” αδιάκοπα.

Φυσικά για να γίνει και η σύνδεση του κάθε χρήστη στο VPN της εφαρμογής μας, απαιτούνται αντίστοιχες ρυθμίσεις στον OpenVPN client από τη μεριά του χρήστη. Αυτές δημιουργούνται αυτόματα μέσα από το client script που ο χρήστης έχει τη δυνατότητα να κατεβάσει από την εφαρμογή μας.

Στη συνέχεια αυτής της ενότητας θα περιγράψουμε μόνο την τελική μορφή που παίρνει το αρχείο client.conf του χρήστη, αφού σχηματιστεί με την αυτόματη διαδικασία του script. Ο τρόπος με τον οποίο σχηματίζεται τελικά αυτό το αρχείο, σε συνδιασμό με τα βήματα που ακολουθεί ο χρήστης για να φτάσει ως εκεί, περιγράφονται σε επόμενο κεφάλαιο.

Αναλυτική περιγραφή των ρυθμίσεων στο client.conf αρχείο

Με βάση αυτό το αρχείο, ο client που ρυθμίζεται στη μεριά του χρήστη έχει τις εξής παραμέτρους:

`client`	Ορίζει ότι το OpenVPN θα συμπεριφέρεται ως client.
`dev tun`	Ορίζει το όνομα του interface με το οποίο θα συνδέεται ο client στο VPN δίκτυο.
`remote 83.212.x.x`	Ορίζει την IP διεύθυνση του server στον οποίο προκειται να συνδεθεί. Για τις ανάγκες της παρούσας μεταπτυχιακής διατριβής χρησιμοποιήσαμε server από την υπηρεσία Okeanos.
`ca ca.crt`	Ορίζει το αρχείο με το πιστοποιητικό της αρχής πιστοποίησης. Το αρχείο αυτό διαμορφώνεται από το script που θα εκτελέσει ο χρήστης, όπως θα δούμε στη συνέχεια.
`cert client.crt`	Ορίζει το αρχείο με το πιστοποιητικό του client. Το αρχείο αυτό διαμορφώνεται από το script που θα εκτελέσει ο χρήστης, όπως θα δούμε στη συνέχεια.
`key client.key`	Ορίζει το αρχείο με το ιδιωτικό κλειδί του χρήστη. Το αρχείο αυτό διαμορφώνεται από το script που θα εκτελέσει ο χρήστης, όπως θα δούμε στη συνέχεια.
`comp-lzo`	Εφαρμόζει τον αλγόριθμο συμπίεσης LZO στα πακέτα που μεταδίδονται. Δίνει βελτίωση στην ταχύτητα λόγω του μικρότερου μεγέθους πακέτου. Αν δε χρησιμοποιεί την ίδια ρύθμιση και ο server, τότε αυτή η ρύθμιση μπορεί να οδηγήσει σε πρόβλημα σύνδεσης.
`keepalive 10 60`	Ρύθμιση έτσι ώστε να αδρανοποιείται η σύνδεση μετά από ένα ορισμένο διάστημα “ακινησίας”. Αυτή επανέρχεται μόλις δει ξανά αίτημα για επικοινωνία.
`persist-tun`	Σε περίπτωση διακοπής σύνδεσης λόγω ακινησίας το interface tun παραμένει και δεν καταστρέφεται.
`persist-key`	Σε περίπτωση διακοπής σύνδεσης λόγω ακινησίας δε χρειάζεται να ξαναδιαβαστούν τα κλειδιά από την αρχή.
`user testuser`	Ο Linux χρήστης που τρέχει το OpenVPN.
`group testuser`	Το Linux group στο οποίο ανήκει ο χρήστης που

	τρέχει το OpenVPN.

ΚΕΦΑΛΑΙΟ 3

Το Web κομμάτι της εφαρμογής μας

Ένα μεγάλο κομμάτι της ανάπτυξης της εφαρμογής ανήκει στην ανάπτυξη του Web κομματιού της. Επί της ουσίας αποτελεί το πρώτο βήμα του χρήστη έτσι ώστε να μπορέσει να κάνει λήψη του πακέτου με το εκτελέσιμο αρχείο και να αρχίσει να χρησιμοποιεί την εφαρμογή. Επίσης, στη συγκεκριμένη ιστοσελίδα μπορεί να δει ο χρήστης αν είναι επιτυχώς συνδεδεμένος ή όχι, και το πόσα λεπτά διαθεσιμότητας της εφαρμογής του απομένουν.

Στην πρώτη σελίδα, για να αποκτήσει πρόσβαση ο χρήστης στη λήψη του πακέτου του και αργότερα αφού συνδεθεί πρόσβαση στις διάφορες λειτουργίες, του ζητείται να κάνει login με βάση ένα όνομα χρήστη και έναν κωδικό πρόσβασης. Θεωρούμε ότι και οι δύο παράμετροι είναι εκ των προτέρων γνωστές στο χρήστη. Από την πρώτη στιγμή λοιπόν, ο χρήστης καλείται να δώσει ευαίσθητη ιδιωτική πληροφορία στον δικό μας server προκειμένου να συνεχίσει. Αυτή προστατεύεται μέσω του πρωτοκόλλου HTTPS.

Το πρωτόκολλο HTTPS

Γενικότερα, όταν κάποιος χρήστης στείλει αίτημα σε κάποιον απομακρυσμένο εξυπηρετητή στο διαδίκτυο έτσι ώστε να ανταλλάξουν οποιαδήποτε πληροφορία, τα πακέτα του χρήστη περνούν από διάφορα δίκτυα, τα οποία ανήκουν σε διάφορους παρόχους σύνδεσης στο διαδίκτυο. Από αυτήν την τεράστια διαδρομή που ακολουθούν τα δεδομένα μέχρι να φτάσουν στον προορισμό τους, μας είναι αδύνατον να γνωρίζουμε όλους τους παρόχους που μεσολαβούν, πόσο μάλλον τις υποδομές που χρησιμοποιεί ο καθένας έτσι ώστε να ασφαλίσουν τα δίκτυά τους. Αυτό σημαίνει ότι υπάρχει ο κίνδυνος αυτά να ανιχνευθούν και να διαβαστούν από κάποιον σε οποιοδήποτε σημείο στη μέση της διαδρομής.

Το πρωτόκολλο HTTP που χρησιμοποιείται ευρέως για επικοινωνία στο διαδίκτυο δε διαθέτει από μόνο του καμία απολύτως υποδομή για ασφάλεια και απόκρυψη ή κρυπτογράφηση δεδομένων. Πράγμα που σημαίνει, ότι αν αυτά τα δεδομένα περιέχουν κάποια ευαίσθητη ιδιωτική πληροφορία -όπως στην παρούσα περίπτωση το όνομα και ο κωδικός του χρήστη- τότε αυτή μπορεί να γίνει γνωστή σε κάποιο άλλο κακόβουλο άτομο. Αυτός είναι ο λόγος για τον οποίο χτίσαμε όλη τη web εφαρμογή μας κάτω από το πρωτόκολλο HTTPS.

Το πρωτόκολλο HHTPS (Hyper Text Transfer Protocol Secure), αποτελεί την ασφαλή επιλογή να περιηγηθούμε σε μια ιστοσελίδα στο διαδίκτυο με κρυπτογραφημένη επικοινωνία, έτσι ώστε να μπορούμε να στείλουμε ευαίσθητα δεδομένα. Το HTTPS κρυπτογραφεί τα πακέτα HTTP κάνοντας χρήση του SSL (Secure Socket Layer). Η διαδικασία της κρυπτογράφησης - αποκρυπτογράφησης είναι η γνωστή, αυτή του ιδιωτικού - δημόσιου κλειδιού για τις δύο οντότητες επικοινωνίας. Έτσι, όταν ο χρήστης μπαίνει σε μια ιστοσελίδα που χρησιμοποιεί το πρωτόκολλο HTTPS, τότε ο περιηγητής σελίδων του χρήστη, λαμβάνει το πιστοποιητικό του server της ιστοσελίδας, για να μπορεί να αποκρυπτογραφήσει τα δεδομένα που λαμβάνει. Το ίδιο συμβαίνει και προς την αντίθετη κατεύθυνση.

Το HTTPS είναι ένα πρωτόκολλο που χρησιμοποιείται όλο και περισσότερο με την πρόοδο του διαδικτύου, με το 25% των ιστοσελίδων να λειτουργούν αποκλειστικά με αυτό, ποσοστό που όλο και αυξάνεται.

Flask Web Framework

Κατά τη διαδικασία κατασκευής ιστοσελίδων, στις περισσότερες περιπτώσεις οι προγραμματιστές που τα αναπτύσσουν δεν ξεκινούν “from scratch”, δηλαδή από το μηδέν, αλλά χρησιμοποιούν ένα σύνολο έτοιμων βιβλιοθηκών, μεθόδων και λειτουργιών, το οποίο επιταχύνει τη διαδικασία της ανάπτυξης. Όλο αυτό το σύνολο λειτουργιών ονομάζεται πλαίσιο εργασίας (web framework). Στην παρούσα μεταπτυχιακή διατριβή, και για να επωφεληθούμε στο έπακρο τις δυνατότητες του, χρησιμοποιήσαμε το Flask, ένα micro-framework.

Το Flask micro web framework είναι γραμμένο σε Python, στημένο πάνω στις προδιαγραφές του WSGI και χρησιμοποιεί την άδεια λειτουργίας BSD. Η τελευταία σταθερή έκδοση του Flask -και αυτή που χρησιμοποιούμε- είναι v0.12, που υπάρχει από το Δεκέμβριο του 2016. Υπάρχουν πολλές διάσημες εφαρμογές στο διαδίκτυο που κάνουν χρήση του Flask framework και των δυνατοτήτων του, όπως το LinkedIn και το Pinterest.

Το Flask αποκαλείται micro framework γιατί δεν απαιτεί από μόνο του ο χρήστης να έχει προεγκατεστημένα εργαλεία ή άλλες βιβλιοθήκες. Δεν είναι εξαρτώμενο από τίποτα άλλο πλην του Python Standard Library, και μπορεί να λειτουργήσει σαν "μονάδα λογισμικού του ενός αρχείου" (single file module), διαθέτοντας έτσι απλότητα και φοβερή ευκολία στην εγκατάσταση και την εφαρμογή. Με τον όρο "micro" δεν εννοούμε ότι η εφαρμογή μας πρέπει οπωσδήποτε να χωρέσει σε ένα και μοναδικό αρχείο (που όμως κάτι τέτοιο είναι εφικτό), αντίθετα εννοούμε ενώ ότι ο πυρήνας της εφαρμογής παραμένει για πάντα απλός, η δυνατότητα επεκτασιμότητας είναι τεράστια. Το Flask μας αφήνει ελεύθερους να επιλέξουμε εμείς τα πάντα γύρω από την εφαρμογή μας, όπως για παράδειγμα τον τύπο της βάσης δεδομένων.

Αν χρειαστεί στον προγραμματιστή, υποστηρίζει πάμπολλες επεκτάσεις και βιβλιοθήκες οι οποίες είναι τόσο συμβατές και σταθερές, που αν εγκατασταθούν μοιάζουν σαν να έχουν αναπτυχθεί εντός του Flask από την αρχή. Ανάμεσα στις διασημότερες επεκτάσεις περιλαμβάνονται: η αντικειμενοστραφής απεικόνιση (object-relational mapping) που μας επιτρέπει να διαχειριζόμαστε τα αποτελέσματα των ερωτήσεων στη βάση ως αντικείμενα (κυρίως όταν αναπτύσσουμε τη web εφαρμογή μας με αντικειμενοστραφή, object-oriented λογική) πράγμα που οδηγεί την εύκολη μετατροπή του σε πλαίσιο εργασίας μοντέλου-απεικόνισης-ελέγχου (Model-View-Controller, MVC framework), επικυρώσεις έγκυρων δεδομένων σε συμπλήρωση φόρμας από το χρήστη (form validation), διαχείρηση "ανεβάσματος" αρχείων (upload handling), και διάφορα άλλα εργαλεία. Σε πολλά άλλα Frameworks που υπάρχουν για ανάπτυξη ιστοσελίδων, όλα τα παραπάνω υπάρχουν από πριν, σαν 3rd party λογισμικό. Οι λίγες προεγκατεστημένες επεκτάσεις (όπως για παράδειγμα το Jinja2 template engine), μπορούν να αντικατασταθούν με μεγάλη ευκολία.

Εικόνα 5: Για τις ανάγκες του Web Interface της εφαρμογής μας χρησιμοποιήσαμε το Flask Framework.

Πλεονεκτήματα του Flask Framework:

Δρομολόγηση (routing): Κάθε ένα URL της εφαρμογής μας, μπορεί να αντιπροσωπευθεί δυναμικά από μία αντίστοιχη Python συνάρτηση στον κώδικα της εφαρμογής. Ο κώδικας της συνάρτησης υπολογίζει και αποφασίζει για τα δυναμικά, μεταβλητά στοιχεία του template, και ακολούθως φορτώνει (renders) το αντίστοιχο template προς παρουσίαση στο χρήση, απαντώντας στο request.

Templates (σχεδιαγράμματα): Το Flask Framework διαθέτει γρήγορο, εύκολο στη χρήση, και "ταιριαστό" στη λογική της γλώσσας προγραμματισμού Python, ενσωματωμένο μηχανισμό σχεδιαγραμμάτων (built-in template engine) και υποστήριξη των jinja2 templates (είναι αυτά που χρησιμοποιούμε).

Πολλές βοηθητικές ευκολίες: Βολική πρόσβαση στα δεδομένα που προκύπτουν έπειτα από συμπλήρωση φόρμας από το χρήστη, από upload αρχείων, αυτόματη διαχείριση cookies, επικεφαλίδες HTTP και διάφορα άλλα HTTP μετα-δεδομένα που ενδεχομένως να υπάρχουν.

Εξυπηρετητής (server): Προς διάθεση του προγραμματιστή διαθέτει και ενσωματωμένο μικρό server. Μπορεί να λειτουργήσει σαν κανονικός server χωρίς πρόβλημα, όμως, λόγω των περιορισμένων δυνατοτήτων του προτιμήσαμε να τον χρησιμοποιήσουμε μόνο για δοκιμαστικούς σκοπούς κατά τη διάρκεια ανάπτυξης της εφαρμογής. Εκτός αυτού, το Flask framework μπορεί να εξυπηρετηθεί και από οποιοδήποτε λογισμικό το οποίο υποστηρίζει WSGI (όλα τα γνωστά λογισμικά εξυπηρετητή όπως Apache και nginx έχουν αυτή τη δυνατότητα).

Εύκολο σύστημα αυθεντικοποίησης και σύνδεσης - αποσύνδεσης του χρήστη: Για όλα τα URLs δρομολόγησης που έχουμε στο σύστημά μας, είναι εύκολο με τη χρήση annotations, θα διευκρινίσουμε σε ποιους χρήστες είναι προσβάσιμα. Έτσι μας δίνεται η δυνατότητα, όπως θα δούμε παρακάτω, σε κάθε χρήστη να είναι προσβάσιμα μόνο τα URLs που του προσφέρουν τα πιστοποιητικά του, και αποκλειστικά στον διαχειριστή (username admin) της εφαρμογής ορισμένα URLs όπως το /register, για την προσθήκη νέων χρηστών.

Η χρήση του Apache Web Server

Για τις ανάγκες της εφαρμογής μας χρησιμοποιήσαμε τον Apache server, για πολλούς και διάφορους λόγους:

Υποστηρίζει απόλυτα το WSGI, και άρα και το Flask Framework. Το WSGI δεν είναι ούτε server, ούτε κάποια Python οντότητα, κάποιο framework ή κάποιο άλλο παρεμφερές είδος λογισμικού. Είναι μία διεπαφή, ή καλύτερα κάποιες προδιαγραφές διεπαφής (interface specification) με τις οποίες ο server και η εφαρμογή μπορούν να επικοινωνούν Αν κάποια εφαρμογή ή framework έχουν γραφτεί σύμφωνα με τις προδιαγραφές WSGI τότε πρέπει και να "τρέξουν" από έναν server ο οποίος το υποστηρίζει. Όταν λέμε ότι οι εφαρμογές WSGI μπορούν να "στοιβαχτούν" (stacked), σημαίνει πως, όταν ένας χρήστης κάνει κάποιο αίτημα (request) στην εφαρμογή μας, τότε τα δεδομένα επεξεργάζονται με τη σειρά η μία εφαρμογή μετά την άλλη. Οι εφαρμογές που βρίσκονται "στη μέση", μεταξύ server και τελικής εφαρμογής, αποκαλούνται middleware και παίζουν και το ρόλο της εφαρμογής αλλά και το ρόλο του "ενδιάμεσου server", για να περάσουν τα δεδομένα στην επόμενη.

Ένας WSGI server (ή καλύτερα, ένας server που υποστηρίζει WSGI), απλά λαμβάνει το αίτημα από τον client, το περνάει στην εφαρμογή και μετά στέλνει την απάντηση που του έχει δώσει η εφαρμογή πίσω. Δεν κάτι τίποτα παραπάνω, καμιά περαιτέρω επεξεργασία.

Δε χρειάζεται καθόλου να γνωρίζουμε τεχνικές λεπτομέρειες για να χτίσουμε WSGI applications χρησιμοποιώντας τέτοια frameworks. Σε περίπτωση μόνο που χρησιμοποιήσουμε περισσότερες από μία (stacked applications), τότε χρειάζεται μια μικρή γνώση για τη συμπεριφορά τους. Αυτό όμως δε συμβαίνει στη δική μας περίπτωση, όπου αρκούμαστε με το Flask.

Το "εξάρτημα" του Apache Server για υποστήριξη εφαρμογών WSGI είναι το mod_wsgi, το οποίο και ενεργοποιήσαμε για να μπορέσει να εξυπηρετήσει τη Flask εφαρμογή μας επιτυχώς.

Εκτός από την υποστήριξη εφαρμογών WSGI, ο Apache server μας ήταν χρήσιμος και για την εξυπηρέτηση όλων των υπόλοιπων λειτουργιών της εφαρμογής μας. Εκτός λοιπόν από τις ρυθμίσεις καθ' αυτές για την εξυπηρέτηση της σελίδας της εφαρμογής, καλεστήκαμε να "σερβίρουμε" μέσω του Apache server και τις υπόλοιπες σελίδες, όπως το blog, το social media, η σελίδα αναπαραγωγής βίντεο, και άλλες πολλές, οι οποίες όμως έχουν το εξής κοινό χαρακτηριστικό: είναι προσβάσιμες μόνο σε όσους είναι συνδεδεμένοι στο VPN δίκτυο της εφαρμογής μας, και σε κανέναν άλλον.

Για την αντιμετώπιση αυτού του προβλήματος ο Apache server προσφέρει κάποιες πολύ χρήσιμες λειτουργίες, όπως ο περιορισμός το να απαντάει σε αιτήματα τα οποία έρχονται μόνο από ένα συγκεκριμένο δίκτυο. Με δεδομένο λοιπόν, ότι ο χρήστης αφού συνδεθεί στο VPN δίκτυο συνεπάγεται και ότι έχει τη δική του, γνωστή IP μέσα στο subnet του VPN, ο Apache server ρυθμίστηκε έτσι ώστε να απαντά μόνο σε αυτές τις IP, και σε κανέναν παραέξω. Έτσι κερδίζουμε όλες οι πρόσθετες λειτουργίες της εφαρμογής μας να είναι προσβάσιμες μόνο από τους συνδεδεμένους χρήστες.

Η βάση δεδομένων που χρησιμοποιήσαμε και ο τρόπος αποθήκευσης των κωδικών πρόσβασης

Το μεγαλύτερο πλεονέκτημα όμως του Flask Framework, και ο βασικότερος λόγος για τον οποίο τον επιλέξαμε είναι ότι το Flask δε διαθέτει από πριν, προεγκατεστημένο "επίπεδο αφαιρετικότητας βάσης δεδομένων" (πιο σωστά Database Abstraction Layer). Αυτό σημαίνει ότι δεν έχει από πριν, προεγκατεστημένο ένα μηχανισμό έτσι ώστε να "μιλάει" με μία βάση δεδομένων και να μας "περιορίζει" στον τρόπο που θα διαχειριστούμε τα δεδομένα μας. Αυτό λοιπόν μας έδωσε τη δυνατότητα να δημιουργήσουμε έναν αποκλειστικό μηχανισμό, για το πώς θα διαχειριστούμε τους χρήστες και την αποθήκευσή τους στη βάση δεδομένων, και ακόμα περισσότερο, να επιλέξουμε τον δικό μας τρόπο κρυπτογράφησης για αποθήκευση των κωδικών πρόσβασης.

Ο τύπος της βάσης δεδομένων που επιλέξαμε είναι ο SQLite. Οι SQLite βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι πολύ ελαφριές, άρα συνάμα κερδίζουν σε ταχύτητα. Η αποθήκευση αλλά και η ανάκτηση δεδομένων είναι εύκολη και αποδοτική. Γνωρίζαμε εξ'αρχής ότι η βάση δεδομένων στην εφαρμογή μας δεν επρόκειτο να μεγαλώσει πολύ σε όγκο. Για την ακρίβεια τη χρειαζόμασταν μόνο για την αποθήκευση των χρηστών και τη διαχείριση "χρονισμού" τους. Έτσι, επιλέξαμε την SQLite ως την ιδανική για αυτή τη δουλειά. Επιπλέον, για το χειρισμό της βάσης σε επίπεδο κώδικα Python, χρησιμοποιήσαμε το SQL Alchemy, μία πολύ ισχυρή βιβλιοθήκη για όλες τις δυνατές λειτουργίες που μπορεί να χρειαστούμε στη βάση δεδομένων μας, και απόλυτα συμβατή ως επέκταση του Flask Framework.

Έπειτα λοιπόν από την εγκατάσταση της βάσης δεδομένων, χρειάστηκε να λύσουμε το πρόβλημα της ασφαλούς αποθήκευσης των κωδικών πρόσβασης. Δεν είναι λίγες οι φορές στο παρελθόν που από τις πιο διάσημες ιστοσελίδες του κόσμου, hackers κατάφεραν να υποκλέψουν κωδικούς πρόσβασης χρηστών. Ως εκ τούτου, η αποθήκευση των κωδικών πρόσβασης όπως ακριβώς είναι, αποτελεί τεράστιο κίνδυνο, καθώς σε περίπτωση υποκλοπής ο δράστης θα έχει πλήρη πρόσβαση στην εφαρμογή και σε όλα τα προσωπικά δεδομένα του χρήστη. Το πρόβλημα λοπόν που καλούμαστε να επιλύσουμε, είναι να βρούμε έναν τρόπο κρυπτογράφησης του κωδικού πρόσβασης, έτσι ώστε να είναι υπέρμετρα δύσκολο έως αδύνατο από το κακόβουλο άτομο να τον ανακαλύψει.

Οι πιο διάσημοι τρόποι κρυπτογράφησης των κωδικών πρόσβασης είναι αυτοί του κατακερματισμού (Hashing) και του "αλατίσματος" (Salting). Οι πιο πολλές μέθοδοι χρησιμοποιούν έναν συνδιασμό των δύο αυτών τρόπων. Εμείς χρησιμοποιήσαμε τη μέθοδο κρυπτογράφησης PBKDF2.

Hashing (κατακερματισμός)

Όταν ένας κωδικός πρόσβασης "κατακερματίζεται", εννοούμε ότι μεταβάλλεται σε μία παραλλαγμένη, κωδικοποιημένη αναπαράσταση του εαυτού του. Η τιμή κατακερματισμού (hash value) του κωδικού πρόσβασης του χρήστη υπολογίζεται αλγοριθμικά, με το συνδυασμό της συμβολοσειράς του και ενός κλειδιού, το οποίο είναι γνωστό στην εφαρμογή. Για να βεβαιωθούμε ότι ο κωδικός πρόσβασης που έχει εισάγει ο χρήστης είναι σωστός, αυτός κατακερματίζεται με βάση τον γνωστό αλγόριθμο και το γνωστό κλειδί, και ύστερα συγκρίνουμε την τιμή που προέκυψε με βάση την τιμή που έχουμε αποθηκεύσει στη βάση δεδομένων.

Η διαδικασία του κατακερματισμού είναι μη-αναστρέψιμη, πράγμα που σημαίνει ότι ο μοναδικός τρόπος για να βρούμε από την τιμή κατακερματισμού τον κωδικό πρόσβασης από τον οποίο δημιουργήθηκε, είναι να δημιουργούμε τιμές κατακερματισμού επαναλαμβανόμενα μέχρι να βρούμε αυτή που ταιριάζει. Αυτή η μέθοδος επίθεσης λέγεται brute-force attack.

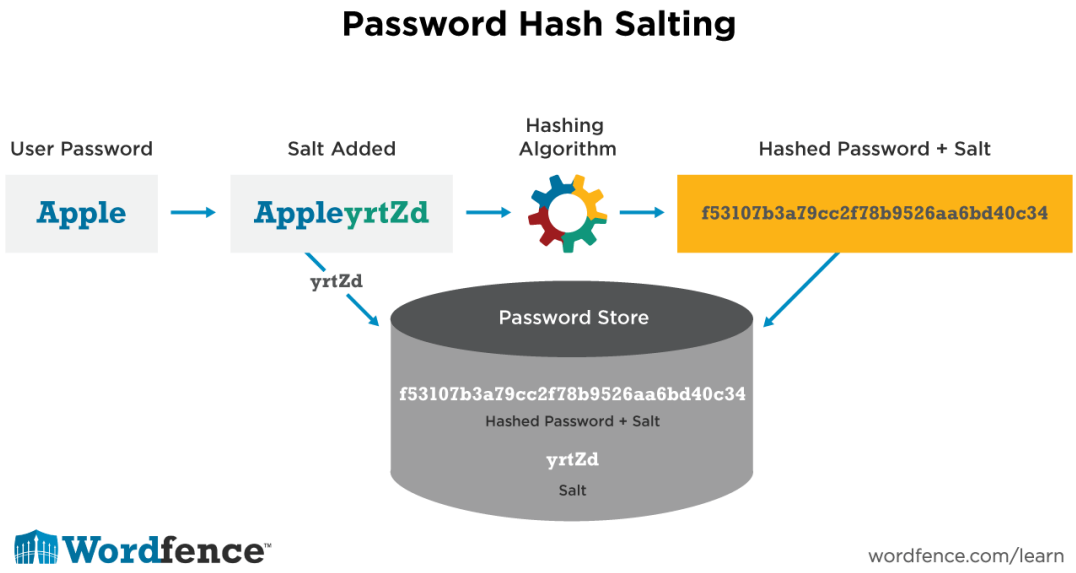
Salting (αλάτισμα)

Πολλές φορές η αποθήκευση των κωδικών πρόσβασης περιγράφεται ως "hashed and salted". Το salting είναι η απλή προσθήκη μιας μοναδικής, τυχαίας συμβολοσειράς στην αρχή ή στο τέλος του κωδικού πρόσβασης αμέσως πριν τον κατατακερματισμό του και την αποθήκευσή του. Συνεπώς, για να εφαρμόσουμε salting σε έναν κωδικό πρόσβασης, είναι απαραίτητο να ακολουθεί η διαδικασία του κατακερματισμού αμέσως μετά.

Η τιμή του salt πρέπει να είναι γνωστή στην εφαρμογή που αποθηκεύει τους κωδικούς πρόσβασης. Έτσι, τις περισσότερες φορές οι εφαρμογές χρησιμοποιούν το ίδιο salt για την αποθήκευση όλων των κωδικών πρόσβασης. Φυσικά αυτό κάνει φυσικά τη μέθοδο λιγότερο αποδοτική.

Η χρήση μοναδικών salts σημαίνει ότι οι συνηθισμένοι κωδικοί πρόσβασης που χρησιμοποιούν πολλοί χρήστες - όπως το "123456" ή ο "password" - δεν αποκαλύπτονται αμέσως όταν εντοπιστεί ένας τέτοιος κωδικός πρόσβασης - γιατί λόγω του salting η τιμή κατακερματισμού της συμβολοσειράς δεν είναι η ίδια. Τα μεγάλα σε μήκος salts προστατεύουν επίσης από συγκεκριμένες μεθόδους επίθεσης σε hashes, συμπεριλαμβανομένων των "rainbow tables" (πίνακες με προϋπολογισμένα hashes όλων των "συνηθισμένων" κωδικών πρόσβασης) ή της καταγραφής κατακερματισμένων κωδικών πρόσβασης που έχουν προηγουμένως σπάσει. Το προτεινόμενο μέγεθος salt είναι τα 64 bits.

Τόσο το hashing όσο και το salting μπορούν να επαναληφθούν περισσότερες από μία φορές για να αυξηθεί η δυσκολία στο άτομο που προσπαθεί να "σπάσει" τους κωδικούς πρόσβασης. Αυτό ακριβώς κάνει και η μέθοδος PBKDF2.



Εικόνα 6: Η διαδικασία "hashing and salting" ενός κωδικού πρόσβασης.

Η μέθοδος κρυπτογράφησης κωδικών πρόσβασης PBKDF2

Το PBKDF2 είναι μια ευρέως χρησιμοποιούμενη μέθοδος για την εξαγωγή μιας συμβολοσειράς δεδομένου μήκους με βάση έναν δεδομένο κωδικό πρόσβασης, ένα salt και έναν αριθμό επαναλήψεων. Σε αυτή την περίπτωση, χρησιμοποιεί συγκεκριμένα τον αλγόριθμο hashing HMAC με τη συνάρτηση κατακερματισμού SHA-1.

Ο HMAC αλγόριθμος έχει μια ενδιαφέρουσα ιδιότητα: εάν η παρεχόμενη συμβολοσειρά που θέλουμε να κατακερματίσουμε είναι μεγαλύτερη από το μέγεθος που υποστηρίζει σαν είσοδο η συνάρτηση κατακερματισμού που χρησιμοποιούμε, τότε χρησιμοποιεί την κατακερματισμένη τιμή της συμβολοσειράς, αντί για την ίδια τη συμβολοσειρά.

Η ιδέα των πολλαπλών επαναλήψεων είναι απλή. Αντί να αποθηκεύσουμε μόνο μια φορά το αποτέλεσμα του SHA1 αλγορίθμου του κωδικού πρόσβασης, το χρησιμοποιούμε σαν είσοδο στον επόμενο SHA1, μετά στον επόμενο, κοκ. Έτσι επαναλαμβανόμενα καλούμε τη συνάρτηση SHA1 στο προηγούμενο αποτέλεσμα. Με αυτόν τον τρόπο, ο επιτιθέμενος χρειάζεται επίσης να καλέσει αντίστοιχες φορές τον αλγόριθμο SHA1, κι έτσι μια brute-force επίθεση θα γίνει πολύ πιο αργή. Αυτόν ακριβώς τον τρόπο

ακολουθεί και η μέθοδος PBKDF2, έχοντας τροποποιήσει ελαφρά την παραπάνω διαδικασία έτσι ώστε να αυξήσει ακόμα περισσότερο την ασφάλεια. Σε περίπτωση που χρησιμοποιήσουμε μία απλή επανάληψη στον SHA1 πριν αποθηκεύσουμε έναν κωδικό πρόσβασης, τότε ο επιτιθέμενος μπορεί να δοκιμάσει 10.000.000 διαφορετικούς κωδικούς πρόσβασης το δευτερόλεπτο. Αν χρησιμοποιήσουμε τον αλγόριθμο PBKDF2, τότε ο επιτιθέμενος μπορεί να δοκιμάσει 10 διαφορετικούς κωδικούς πρόσβασης το δευτερόλεπτο. Αυτή η τεράστια διαφορά επί της ουσίας καθιστά την επίθεση τύπου brute-force άχρηστη.

Η μέθοδος PBKDF2 εφαρμόζει μια "ψευδοτυχαία συνάρτηση" (pseudorandom function), μαζί με τον αλγόριθμο κατακερματισμού HMAC (Hash based message authentication code) στον εισαγόμενο κωδικό πρόσβασης, προσθέτοντας και το salt. Αυτή η διαδικασία επαναλαμβάνεται πολλαπλές φορές για να παράγει το λεγόμενο "παραγόμενο κλειδί" (derived key), το οποίο στη συνέχεια μπορεί να χρησιμοποιηθεί ως κρυπτογραφικό κλειδί σε ακόλουθες επαναλήψεις. Αυτή η μέθοδος είναι γνωστή και ως "έκταση κλειδιού" (key stretching) και καθιστά το "σπάσιμο" του κωδικού πρόσβασης εξαιρετικά δύσκολο έως αδύνατο.

Όταν πρωτοξεκίνησε να εφαρμόζεται η μέθοδος PBKDF2 στις αρχές του 2000, ο προτεινόμενος ελάχιστος αριθμός επαναλήψεων για να θεωρηθεί ασφαλής η αποθήκευση του κωδικού ήταν 1000. Όμως, και όσο ανέβαιναν οι ταχύτητες των επεξεργαστών, αυτός ο αριθμός αυξήθηκε. Αυτή τη στιγμή υπάρχουν και servers οι οποίοι τρέχουν μέχρι και 100.000 επαναλήψεις.

Εγγραφές νέων χρηστών

Αφού εγκαταστήσαμε την SQLite βάση δεδομένων, κατασκευάσαμε και μία μικρή βιβλιοθήκη σε Python (ονόματι auth) η οποία περιέχει την κλάση User που αναφέρεται στους χρήστες που πρόκειται να αποθηκεύσουμε στην εφαρμογή μας. Ο πρώτος χρήστης που εισάγαμε είναι ο χρήστης admin, ο οποίος είναι και ο μόνος υπεύθυνος για τις προσθήκες επιπλέον χρηστών.

Για αυτή τη δουλειά, θέσαμε ένα νέο route στη δρομολόγηση της εφαρμογής μας (/register), στο οποίο μόνο ο χρήστης admin μπορεί να εισέλθει, και με τη συμπλήρωση σε μια φόρμα ενός ονόματος χρήστη, ενός κωδικού πρόσβασης και email μπορεί να προσθέσει νέους χρήστες στην βάση δεδομένων.

Πώς εξυπηρετούνται οι αιτήσεις για τα απαραίτητα πιστοποιητικά από το χρήστη

Όπως θα δούμε στη συνέχεια, το script που θα έχει στα χέρια του ο χρήστης, παίζει το ρόλο του να λαμβάνει όλα τα πιστοποιητικά που χρειάζεται για να συνδεθεί στην εφαρμογή μας μέσω OpenVPN.

Η πολύ εύκολη και χρήσιμη λειτουργία δρομολόγησης μέσω URL (routing) που μας προσφέρει το Flask Framework, μας επέτρεψε να μπορούμε να δώσουμε στον εκάστοτε αυθεντικοποιημένο χρήστη τα πιστοποιητικά που ζητά. Όπως είπαμε και παραπάνω, τα απαραίτητα αρχεία που χρειάζεται κάποιος για να συνδεθεί είναι:

1) Το πιστοποιητικό της Αρχής Πιστοποίησης (certificate authority) που οφείλει να έχει "υπογεγραμμένα" όλα τα πιστοποιητικά των χρηστών για να γίνουν δεκτά. Είναι διαθέσιμο στο URL /cacert.
2) Το κλειδί του χρήστη, έτσι ώστε να μπορεί να αποκρυπτογραφεί όλη την επικοινωνία που γίνεται μέσω VPN. Είναι διαθέσιμο στο URL /clientkey.
3) το πιστοποιητικό του χρήστη, που χρειάζεται για να πιστοποιεί ο χρήστης κάθε φορά ότι είναι όντως το πρόσωπο που υποστηρίζει ότι είναι. Βρίσκεται διαθέσιμο στο URL /clientcert.

Τα τρία παραπάνω αρχεία, είναι δυνατό να γίνουν προσβάσιμα από οποιονδήποτε συνδεδεμένο χρήστη, μπαίνοντας στα αντίστοιχα URLs. Από τη στιγμή που ο χρήστης (ή το script αντίστοιχα) στείλει ένα αίτημα προς κάποιο από αυτά τα URLs, το Flask Framework αναλαμβάνει να στείλει, με τη μορφή αρχείου κειμένου το αντίστοιχο αρχείο που θα υπάρχει μέσα στο φάκελο με τα πιστοποιητικά και τα κλειδιά του OpenVPN, /usr/share/easy-rsa. Τη δυνατότητα αυτή μας τη δίνει η μέθοδος send_file που περιέχεται στις ενσωματωμένες μεθόδους του Flask Framework. Τα πιστοποιητικά αυτά έχουν προφανώς εντός του server μόνο δικαιώματα ανάγνωσης και όχι τροποποίησης. Επίσης είναι προφανές, ότι αν ο χρήστης δεν έχει κάνει πρώτα login στην εφαρμογή, δεν υπάρχει τρόπος να αποκτήσει πρόσβαση σε οποιοδήποτε από αυτά τα αρχεία (προστατεύεται με το annotation @login_required).

Ο μηχανισμός χρονομέτρησης των χρηστών

Όπως είπαμε και παραπάνω, ένα από τα πιο σημαντικά χαρακτηριστικά που δώσαμε στην εφαρμογή μας είναι η καταμέτρηση του χρόνου κατά τον οποίο ο χρήστης έχει το δικαίωμα σύνδεσης, και θέσπιση ενός μηχανισμού για την αφαίρεση αυτού του δικαιώματος και η αποσύνδεση του χρήστη όταν ο χρόνος που έχει αιτηθεί έρχεται εις πέρας.

Αυτή η λειτουργία έρχεται εις πέρας με τη χρήση νημάτων (threads), και συγκεκριμένα της βιβλιοθήκης threading στη γλώσσα προγραμματισμού Python. Η βιβλιοθήκη threading έχει ό,τι χρειάζεται για αποτελεσματική διαχείριση και συγχρονισμό νημάτων. Τα νήματα, και ιδιαίτερα στην γλώσσα Python, χρησιμοποιούνται σε περιπτώσεις που η εκτέλεση μιας εργασίας από τον κώδικα περιλαμβάνει κάποιο χρόνο αναμονής. Ο μηχανισμός threading επιτρέπει στην Python να εκτελέσει κάποιο άλλο κομμάτι κώδικα όσο περιμένει. Αυτό προσομοιώνεται εύκολα με τη μέθοδο sleep.

Στη threading βιβλιοθήκη της Python, η κλάση που αντιπροσωπεύει ένα νήμα λέγεται Thread. Για να ξεκινήσουμε κάθε φορά ένα νήμα, καλούμε τη μέθοδο start από το αντικείμενο Thread που έχουμε δημιουργήσει. Κάθε φορά που ξεκινούμε ένα νήμα, αυτό αποκτά μεταξύ άλλων και ένα όνομα. Μια πολύ χρήσιμη κλάση που επίσης ανήκει στην βιβλιοθήκη threading της Python, είναι η κλάση Timer. Αυτή κληρονομεί όλες τις λειτουργίες της από την μητέρα κλάση Thread, και έχει και μία παραπάνω ιδιότητα: παίρνει σαν όρισμα έναν αριθμό, που είναι ο χρόνος αναμονής του νήματος πριν ξεκινήσει. Αυτή είναι και η ιδιότητα που εκμεταλλευτήκαμε.

Ανάκληση πιστοποιητικού (Certificate Revocation)

Στο λογισμικό OpenVPN, καθώς και σε άλλα λογισμικά που χρησιμοποιούνται για VPN συνδέσεις, υπάρχει η δυνατότητα ανάκλησης του πιστοποιητικού ενός χρήστη. Ανάκληση πιστοποιητικού σημαίνει πως σταματά πια αυτό να είναι έγκυρο, παρ' όλο που έχει προηγουμένως υπογραφεί από την Αρχή Πιστοποίησης (certification authority). Ο χρήστης και κάτοχος αυτού του πιστοποιητικού πλέον δεν πιστοποιείται, κι έτσι δεν του δίνεται το δικαίωμα σύνδεσης και πρόσβασης στο ιδιωτικό VPN δίκτυο.

Για να γνωρίζει το OpenVPN ποια είναι τα ανενεργά πιστοποιητικά, διαβάζει από μία λίστα που λέγεται Λίστα Ανάκλησης Πιστοποιητικών (Certificate Revocation List). Οποιοδήποτε πιστοποιητικό χρήστη υπάρχει εντός αυτής της λίστας δεν αναγνωρίζεται. Για να ελέγχει ο VPN server τη λίστα CRL πριν από τη σύνδεση κάθε χρήστη, χρειάζεται να είναι ρυθμισμένο από πριν στο αρχείο ρύθμισης του OpenVPN, με την επιλογή crl-verify και το μονοπάτι που οδηγεί στο αντίστοιχο pem αρχείο.

```
root@snf-691977:/usr/share/easy-rsa# cat keys/index.txt
V       270831190830Z           01      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=vpnserver/name=EasyRSA/emailAddress=panospet@gmail.com
V       270831191628Z           03      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=admin/name=EasyRSA/emailAddress=panospet@gmail.com
V       270831191750Z           06      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=mastoras/name=EasyRSA/emailAddress=panospet@gmail.com
R       270831203213Z   170905165302Z   07      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=tol/name=EasyRSA/emailAddress=panospet@gmail.com
V       270831191638Z           04      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=panos/name=EasyRSA/emailAddress=panospet@gmail.com
R       270831191032Z   171005220431Z   02      unknown /C=GR/ST=Athens/L=Pireus/O=Unipi/OU=mpsp/CN=test/name=EasyRSA/emailAddress=panospet@gmail.com
root@snf-691977:/usr/share/easy-rsa#
```

Εικόνα 7: Παράδειγμα μιας Certificate Revocation List. Φαίνεται πως οι χρήστες tol και test είναι μη πιστοποιημένοι να χρησιμοποιήσουν την εφαρμογή.

Πώς εφαρμόζουμε την ανάκληση πιστοποιητικού στην εφαρμογή μας

Με το που ένας χρήστης κάνει login στην εφαρμογή μας και αιτηθεί κάποια λεπτά πρόσβασης σε αυτήν, κατευθείαν ενεργοποιείται ένα καινούριο νήμα μέσω της δημιουργίας ενός αντικειμένου Timer στον κώδικά μας. Το όνομα του νέου Timer, μαζί με τη χρονική στιγμή που ξεκίνησε ο Timer και τα λεπτά που θα διαρκέσει, αποθηκεύονται στη βάση δεδομένων της εφαρμογής μας δίπλα στο όνομα του χρήστη που μόλις εκτέλεσε την ενέργεια αυτή.

Η ενέργεια που έχουμε θέσει στον Timer να εκτελέσει αφού τελειώσει η αντίστροφη μέτρηση, είναι αυτή της ανάκλησης πιστοποιητικού. Για λόγους ευκολίας, επιλέξαμε αυτές οι ενέργειες (ανάκληση και άρση ανάκλησης) να γραφτούν σε bash scripts τα οποία θα καλούνται από την Python εντός του κώδικα στο Flask Framework.

Για την διαδικασία της ανάκλησης, το εργαλείο easy-rsa που υπάρχει εγκατεστημένο μαζί με το OpenVPN, μας παρέχει αυτή τη δυνατότητα μέσω μιας εντολής, που προσθέτει το πιστοποιητικό στη

λίστα ανάκλησης και ανανεώνει, το pem αρχείο από το οποίο διαβάζει το OpenVPN. Εκτός αυτού, το συγκεκριμένο script κοιτάζει να αποσυνδέσει ακαριαία τον χρήστη, σε περίπτωση που αυτός είναι συνδεδεμένος ακριβώς τη στιγμή που αυτό εκτελείται. Σε περίπτωση που το script αυτό δεν πραγματοποιούσε αυτή την ενέργεια, τότε θα περιμέναμε μέχρι να συμβεί ο περιοδικός έλεγχος των certificates που γίνεται ανά τακτά χρονικά διαστήματα κατά τη διάρκεια μιας OpenVPN σύνδεσης, πράγμα που θα οδηγούσε τον -μη επιθυμητό πλέον- χρήστη να παραμείνει για παραπάνω χρόνο σε σύνδεση από όσο του αναλογεί.

Η παραπάνω αποσύνδεση του χρήστη πραγματοποιείται με το άνοιγμα μιας telnet σύνδεσης στο OpenVPN, και τη χρήση της εντολής kill. Γενικότερα το OpenVPN μας δίνει τη δυνατότητα διαχείρισης των λειτουργιών και της συμπεριφοράς μιας VPN σύνδεσης σε πραγματικό χρόνο, μέσω της σύνδεσης telnet.

Πώς ενεργοποιούμε ξανά ένα πιστοποιητικό χρήστη

Τα πιστοποιητικά των χρηστών ενεργοποιούνται ξανά όταν αυτοί αιτηθούν εκ νέου χρόνο σύνδεσης στην εφαρμογή μας, από τη στιγμή δηλαδή, που κάνουν λήψη εκ νέου του πακέτου με τα εκτελέσιμα αρχεία.

Λόγω του ότι η διαδικασία επανενεργοποίησης δεν προσφέρεται έτοιμη από το εργαλείο easy-rsa, πρέπει εμείς χειροκίνητα να την πραγματοποιήσουμε. Το script unrevoke.sh, αφού ελέγξει ότι το συντακτικό και το περιεχόμενο της λίστας ανάκλησης είναι όπως πρέπει, παραλλάσσει ξανά αυτή τη λίστα, φέρνοντας τη γραμμή η οποία αναφέρεται στο ανακληθέν πιστοποιητικό στην αρχική της μορφή, αυτή που το παριστάνει ως ενεργό. Στη συνέχεια, με βάση τη νέα διαμορφωμένη λίστα ανάκλησης ανανεώνει το pem αρχείο από το οποίο ενημερώνεται το OpenVPN για την κατάσταση των ενεργών πιστοποιητικών. Από δω και στο εξής, και μέχρι να τελειώσει ο χρόνος του νέου Timer που δημιουργήθηκε, ο χρήστης θα μπορεί να συνδεθεί χωρίς πρόβλημα.

Αρχείο καταγραφής (Log file)

Για λόγους παρακολούθησης της "ζωής" της εφαρμογής, καθώς και για να κρατάμε ιστορικό με το ποιοι χρήστες εγγράφονται, ποιοι και πόσο χρόνο αιτούνται, και ποιες είναι οι χρονικές στιγμές που γίνεται ανάκληση των πιστοποιητικών τους, επιλέξαμε να καταγράφουμε όλες αυτές τις κινήσεις σε ένα αρχείο. Αυτό έγινε με εύκολο τρόπο, με χρήση της γλώσσας Python εντός της Flask εφαρμογής.

ΚΕΦΑΛΑΙΟ 4

Από τη μεριά του χρήστη (client side)

Είπαμε και σε προηγούμενο κεφάλαιο, ότι ένας από τους βασικούς στόχους της παρούσας μεταπτυχιακής διατριβής ήταν, από τη μεριά του χρήστη, να μη χρειάζονται καθόλου τεχνικές γνώσεις περί δικτύων ή και περί υπολογιστών γενικότερα για να μπορέσει να γίνει χρήση της εφαρμογής επιτυχώς.

Για να μπορέσει ο χρήστης να εισέλθει στην εφαρμογή, χρειάζεται να γνωρίζει εκ των προτέρων μόνο το όνομα χρήστη (username) και τον κωδικό πρόσβασης, στοιχεία τα οποία θεωρούμε ότι του είναι ήδη γνωστά. Από τη στιγμή που γνωρίζει αυτά τα δύο, μπορεί να κάνει χρήση της εφαρμογής στο σύνολό της.

Στην αρχική σελίδα της εφαρμογής, ζητείται το username και το password από το χρήστη. Μόλις ο χρήστης τα συμπληρώσει επιτυχώς, οδηγείται στην πρώτη σελίδα της εφαρμογής. Εκεί βλέπει αν είναι ήδη συνδεδεμένος ή όχι, και πόσος χρόνος του απομένει. Προφανώς, αν είναι η πρώτη φορά που εισέρχεται, δε θα λάβει ειδοποίηση ότι είναι συνδεδεμένος και ότι δεν έχει χρόνο χρήσης της εφαρμογής στη διάθεσή του.

Ακριβώς από κάτω, ο χρήστης της εφαρμογής βλέπει δύο κενά πεδία, για να συμπληρώσει τον αριθμό των λεπτών που επιθυμεί να μείνει σε δυνατότητα σύνδεσης. Το πρώτο πεδίο, αναφέρεται σε χρήστες Linux, το δεύτερο πεδίο, αναφέρεται σε χρήστες Windows. Άρα, ο χρήστης γνωρίζοντας το λειτουργικό του σύστημα, αναλόγως επιλέγει το ένα από τα δύο. Προφανώς υπάρχουν διαφορές μεταξύ των εκτελέσιμων των δύο λειτουργικών, τις οποίες περιορίσαμε όσο το δυνατόν περισσότερο χρησιμοποιώντας τις βιβλιοθήκες και τις δυνατότητες της γλώσσας Python. Μέχρι όμως να εγκατασταθεί η Python στο λειτουργικό μαζί με όλες τις απαραίτητες βιβλιοθήκες, το κάθε εκτελέσιμο διαφέρει, κυρίως ως προς τον τρόπο με τον οποίο αποκτούνται τα δικαιώματα διαχείρισης. Τα δύο scripts διαφέρουν, δηλαδή, στον τρόπο με τον οποίο γίνεται η εγκατάσταση της Python μαζί με όλες τις απαραίτητες βιβλιοθήκες. Από εκεί και πέρα, το αρχείο client_script.py είναι κοινό και εκτελείται όμοιο και για τα δύο.

Αναλυτικότερα, το script για Windows, ξεκινάει κάνοντας κάποιες κινήσεις έτσι ώστε να αποκτήσει administrator δικαιώματα. Αυτές οι κινήσεις προσομοιώνουν την κίνηση “δεξί κλικ - Run as administrator” που μπορεί να γίνει από το λειτουργικό κανονικά. Επειδή όμως θέλουμε την πλήρη αυτοματοποίηση της διαδικασίας, δε θέλουμε να δώσουμε την οδηγία στο χρήστη να το κάνει αυτό χειροκίνητα. Αναλαμβάνει το script να το κάνει αυτό. Στη συνέχεια, και αφού αποκτήσει πλήρη δικαιώματα, το script εγκαθιστά την Python 2.7 στο λειτουργικό, αφού πρώτα ελέγξει ότι δεν είναι εγκατεστημένη ήδη.

Στο Linux script η απόκτηση sudo δικαιωμάτων είναι αρκετά πιο εύκολη, με χρήση της εντολής sudo μπροστά από τις εντολές. Η αντίστοιχη διαδικασία ακολουθείται κι εκεί, με το script να ελέγχει αν η Python είναι εγκατεστημένη (σχεδόν σε όλες τις σύγχρονες Linux διανομές είναι από πριν) και αν όχι, να την εγκαθιστά.

Η απόκτηση administrator permissions από το εκάστοτε script, πέραν του ότι είναι απαραίτητη για τις εγκαταστάσεις του λογισμικού και των βιβλιοθηκών, είναι επίσης απαραίτητη και για την επαναδιαμόρφωση του πίνακα δρομολόγησης (Routing Table) του λειτουργικού, όταν αυτό θα χρειαστεί στη συνέχεια από το OpenVPN. Το πώς τελικά διαμορφώνεται η δρομολόγηση αφού εκτελεστεί το OpenVPN θα το δούμε στη συνέχεια.

Διαχειριστής πακέτων Python - PIP package manager

Θέλοντας τα δύο scripts να έχουν όσο το δυνατόν περισσότερα κοινά στοιχεία, επιλέξαμε να εγκαταστήσουμε όλες τις βιβλιοθήκες που χρειαζόμαστε κάνοντας χρήση του pip, ενός λογισμικού διαχείρισης πακέτων, που χρησιμοποιείται ευρέως για να εγκαθιστά και να διαχειρίζεται πακέτα

λογισμικού γραμμένα σε Python. Το pip εγκαθίσταται και το ίδιο από ένα εκτελέσιμο σε Python, που βρίσκουμε εύκολα από τα επίσημα repositories, το get-pip.py. Το script μας λοιπόν ξεκινά με την εκτέλεση αυτού του αρχείου και την εγκατάσταση του pip.

Οι Python βιβλιοθήκες που εγκαθιστούμε μέσω pip και απαιτούνται για την εκτέλεση του βασικού script είναι οι εξής:

PycURL: Είναι μία διεπαφή (interface) γραμμένη σε Python που χρησιμοποιεί τη γνωστή ενσωματωμένη βιβλιοθήκη libcurl. Χρησιμοποιείται για να στέλνει αιτήματα και να λαμβάνει αντικείμενα από ένα URL. Είναι παρόμοια σε πολλά σημεία με το urllib της Python. Το PycURL είναι αρκετά αποδοτικό σε ταχύτητα, και επίσης υποστηρίζει αιτήματα διαφόρων ειδών πρωτοκόλλων, όπως FTP, HTTP, HTTPS, SMTP, και άλλα πολλά, καθώς επίσης και πιστοποιητικά SSL. Τα παραπάνω την έχρισαν ιδανική για να στέλνει HTTPS αιτήματα στον δικό μας server, καθώς και να τα αυθεντικοποιεί ανταλλάσσοντας SSL πιστοποιητικά

Certifi: Η Certifi είναι μια προσεκτικά επιμελημένη συλλογή πιστοποιητικών ρίζας (Root Certificates) για την επικύρωση της αξιοπιστίας των πιστοποιητικών SSL, και επαληθεύει την ταυτότητα των servers που πρόκειται να συνδεθούμε. Το χρησιμοποιούμε για να γίνεται η αυθεντικοποίηση με τον server, έτσι ώστε να είναι απόλυτα ασφαλή τα αιτήματα που στέλνουμε.

Το Python script του χρήστη

Ο βασικός ρόλος του Python script, είναι να αιτηθεί και να κατεβάσει επιτυχώς από τον server της εφαρμογής όλα τα απαραίτητα αρχεία και στη συνέχεια να διαμορφώσει το αρχείο ρύθμισης (configuration file) για να μπορέσει ο χρήστης στη συνέχεια να συνδεθεί επιτυχώς στο OpenVPN.

Για να πιστοποιηθεί το script, χρειάζεται ξανά να γίνει login στη σελίδα, αυτή τη φορά με τη χρήση της PycURL. Ο server, κατά τη διάρκεια του αιτήματος για λήψη του πακέτου αρχείων, έχει ήδη επεξεργαστεί το script, πριν το στείλει στο χρήστη, δίνοντας στη μεταβλητή username το όνομα χρήστη. Για λόγους όμως ασφαλείας, ο κωδικός πρόσβασης πρέπει να δοθεί ξανά από τον χρήστη. Έτσι λοιπόν, η νέα συνεδρία (session) με τον server που δημιουργεί η PycURL, είναι πλέον πιστοποιημένη, και έτοιμη να αιτηθεί από ένα ένα τα URLs τη λήψη των αρχείων ca.crt (πιστοποιητικό αρχής πιστοποίησης), client.crt (πιστοποιητικό του χρήστη) και client.key (κλειδί του χρήστη). Με χρήση της Python αυτά εγγράφονται στα αντίστοιχα αρχεία, και υπάρχουν πλέον τοπικά, στη διάθεση του χρήστη.

Στη συνέχεια, το Python script πρέπει να φέρει εις πέρας τη δημιουργία του αρχείου ρύθμισης (configuration file) για να μπορέσει ο χρήστης να συνδεθεί με OpenVPN. Έτσι, με χρήση της Python δημιουργεί το αρχείο client.conf (client.ovpn για το Windows script) του οποίου τα περιεχόμενα έχουν αναλυθεί σε προηγούμενο κεφάλαιο. Να αναφέρουμε ότι για την συμπλήρωση των user και group που χρειάζονται αρχείο ρύθμισης, κάναμε χρήση της ενσωματωμένης βιβλιοθήκης Python pwd.

Η πραγματοποίηση της σύνδεσης

Από τη στιγμή που έχουμε στη διάθεσή μας όλα τα απαραίτητα αρχεία (πιστοποιητικά, κλειδιά και διαμορφωμένα σωστά τα αρχεία ρύθμισης) μένει μόνο να τρέξουμε το λογισμικό OpenVPN για σύνδεση στον server της εφαρμογής. Η διαδικασία αυτή, διαφέρει σημαντικά μεταξύ των δύο εκτελέσιμων, αυτού για Linux και αυτού για Windows.

Σύνδεση σε λειτουργικό σύστημα Linux

Τα πράγματα για αυτό το λειτουργικό σύστημα είναι πάντα απλούστερα, αρκεί μία εγκατάσταση του πακέτου OpenVPN που βρίσκεται μέσα στα επίσημα repositories όλων των διανομών linux, και έπειτα η εκτέλεσή του, χρησιμοποιώντας μαζί με την παράμετρο --config της εντολής, το όνομα του αρχείου ρύθμισης, client.conf. Απαραίτητο στοιχείο για επιτυχή σύνδεση, είναι τα αρχεία που έχει δημιουργήσει το εκτελέσιμο script να εξακολουθούν να βρίσκονται στο ίδιο directory.

Σύνδεση σε λειτουργικό σύστημα Windows

Εδώ κληθήκαμε να επιλέξουμε τον τρόπο με τον οποίο θα εγκαθίσταται το λογισμικό OpenVPN στο σύστημά μας. Επιλέξαμε το OpenVPN portable, μία φορητή έκδοση του OpenVPN, η οποία έχει το

βασικό πλεονέκτημα να μην εγκαθιστά κανένα αρχείο στο σύστημα, παρά μόνο κάποια βασικά αρχεία τοπικά, στο φάκελο στον οποίο βρίσκεται (γίνεται με χρήση της εντολής --config-dir).

Και στα δύο λειτουργικά συστήματα, για εκτέλεση του αρχείου και σύνδεση με το OpenVPN, απαιτούνται δικαιώματα διαχειριστή συστήματος για τον άνθρωπο που τα εκτελεί. Στο λειτουργικό σύστημα Linux αυτό λύνεται με ένα απλό sudo, στο λειτουργικό σύστημα Windows αυτά αποκτούνται με κάποια βήματα στην αρχή του script (προσομοιώνουν την κίνηση "δεξί κλικ - Run as Administrator). Ο βασικότερος λόγος που έχουμε ανάγκη αυτά τα δικαιώματα είναι η τροποποίηση του πίνακα δρομολόγησης (routing table) στις ρυθμίσεις δικτύου του συστήματος.

Πώς διαμορφώνεται ο πίνακας δρομολόγησης (routing table) σε επίπεδο λειτουργικού συστήματος

Σε αυτό το σημείο της παρούσας μεταπτυχιακής διατριβής προτιμήσαμε να δώσουμε ένα παράδειγμα με βάση το routing table όπως διαμορφώνεται στο λειτουργικό σύστημα Linux. Οι αλλαγές στο λειτουργικό σύστημα Windows είναι ακριβώς οι ίδιες και δεν παρουσιάζουν διαφορές.

Ο πίνακας δρομολόγησης διαμορφώνεται από όλες τις πιθανές διαδρομές που ακολουθούν τα πακέτα μας στο σύστημά μας. Όλες τις πιθανές δρομολογήσεις πακέτων μας τις δείχνει η εντολή ip route list. Πριν τη σύνδεσή μας με το OpenVPN, όλες οι πιθανές δρομολογήσεις πακέτων μέσα στο δίκτυο είναι οι παρακάτω:

```
panos@pc:~$ ip route list
default via 192.168.1.1 dev eth0  proto static  metric 600
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.4  metric 600
```

και ο πίνακας δρομολόγησης (εμφανίζεται με την εντολή route -n):

```
panos@pc:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eth0
```

Από αυτό διαπιστώνουμε το εξής απλό: Η IP διεύθυνση 192.168.1.1 είναι αυτή της συσκευής δρομολόγησης (router) που μας έχει δώσει ο πάροχος. Ο όρος default χρησιμοποιείται για να περιλαμβάνει όλα τα πακέτα που έχουν ως προορισμό κάποια διεύθυνση για την οποία δεν έχουμε κάποιο κανόνα δρομολόγησης. Συνεπώς, από τον παραπάνω πίνακα δρομολόγησης καταλαβαίνουμε ότι για όλο τον "έξω κόσμο", τα πακέτα θα χρησιμοποιήσουν ως προεπιλεγμένη πύλη (Default Gateway) τη συσκευή router μας, ενώ για τα πακέτα που πάνε προς τις IP διευθύνσεις 192.168.1.*, περνούν απλώς μέσα από το interface eth0 (πράγμα λογικό, καθώς σημαίνει ότι βρίσκονται στο τοπικό δίκτυο, και συνδέονται στο ίδιο router, άρα η κίνηση δε χρειάζεται να βγει προς τα έξω).

Από τη στιγμή που συνδεθεί επιτυχώς ο client με τον OpenVPN server, τότε έχουμε σαφείς αλλαγές στις διαδρομές και το routing table του συστήματος:

```
panos@pc:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.8.0.9        128.0.0.0       UG    0      0        0 tun0
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
10.8.0.1        10.8.0.9        255.255.255.255 UGH   0      0        0 tun0
10.8.0.9        0.0.0.0         255.255.255.255 UH    0      0        0 tun0
83.212.116.170  192.168.1.1     255.255.255.255 UGH   0      0        0 eth0
128.0.0.0       10.8.0.9        128.0.0.0       UG    0      0        0 tun0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eth0

panos@a9pp:~/Downloads/linux_app$ ip route list
0.0.0.0/1 via 10.8.0.9 dev tun0
default via 192.168.1.1 dev eth0  proto static  metric 600
10.8.0.1 via 10.8.0.9 dev tun0
10.8.0.9 dev tun0  proto kernel  scope link  src 10.8.0.10
83.212.116.170 via 192.168.1.1 dev eth0
128.0.0.0/1 via 10.8.0.9 dev tun0
192.168.1.0/24 dev eth0 proto kernel  scope link  src 192.168.1.4  metric 600
```

Από τα routes που χρησιμοποιούν σαν Default Gateway την IP 10.8.0.9, καταλαβαίνουμε ότι πλέον κίνηση στο σύστημά μας γίνεται μέσω του interface tun0 και άρα του OpenVPN. Όσο κι αν φαίνεται περίεργο στην αρχή, αυτοί οι κανόνες δρομολόγησης περιλαμβάνουν όλες τις δυνατές IP τιμές που μπορεί να έχει σαν προορισμό το πακέτο μας. Οι απλοί κανόνες subnetting μας οδηγούν εύκολα σε αυτό το συμπέρασμα. Η IP με προορισμό 128.0.0.0 και Netmask πάλι 128.0.0.0, ισοδυναμεί με όλες τις πιθανές IP διευθύνσεις με εύρος από 0.0.0.0 μέχρι 127.255.255.255. Αντίστοιχα, η IP διεύθυνση με προορισμό 0.0.0.0 και Netmask 128.0.0.0, καλύπτει όλες τις πιθανές διευθύνσεις IP με εύρος από 128.0.0.0 μέχρι 255.255.255.255. Όλες οι πιθανές IPv4 διευθύνσεις δηλαδή, καλύφθηκαν από αυτούς τους δύο κανόνες.

Το λογισμικό OpenVPN προτιμά να θέσει δύο νέους κανόνες για δρομολόγηση default πακέτων από το να τροποποιήσει ή να σβήσει τον ήδη υπάρχοντα, γιατί θέλει να έχει όσο το δυνατόν μικρότερο "παρεμβατισμό" στις ήδη υπάρχουσες ρυθμίσεις του συστήματος. Πράγμα λογικό, καθώς όταν η σύνδεση σταματήσει, αυτοί οι δύο κανόνες θα αφαιρεθούν και το σύστημα θα επανέλθει ακριβώς στην αρχική του κατάσταση. Οι κανόνες αυτοί "σπάνε" στα δύο, γιατί όσον αφορά τον πίνακα δρομολόγησης, το σύστημα προτιμά πάντοτε έναν την πιο "λεπτομερή" δρομολόγηση σε σχέση με μια απλή. Έτσι οι κανόνες αυτοί θεωρούνται πιο "συγκεκριμένοι" από τον default κανόνα, και τους δίνεται πάντοτε προτεραιότητα.

Αντίστοιχα, παρατηρούμε και τον κανόνα ο οποίος μας λέει ότι για όλα τα πακέτα με προορισμό 10.8.0.*, επίλεξε να πας μέσω του interface tun0, δηλαδή μέσω του VPN. Πάνω στο συγκεκριμένο subnet είναι που γίνεται και η εξυπηρέτηση των επιπλέον λειτουργιών της εφαρμογής μας. Άρα, γνωρίζουμε από πριν που θα καταλήξουν τα αιτήματα που προορίζονται για πρόσβαση σε αυτές τις εφαρμογές.

Είδαμε ότι η όλη η κίνηση γίνεται πλέον μέσω του δικτύου VPN. Αυτό σημαίνει, ότι οποιοδήποτε αίτημα από μεριάς client επαναπροωθείται από τον VPN server. Άρα, αυτός που φαίνεται να εκτελεί τα αιτήματα είναι ο server με τη δική του διεύθυνση, χωρίς κανείς να μπορεί να γνωρίζει πως ο client παραμένει "κρυμμένος πίσω του". Πράγμα λογικό, αφού η επικοινωνία client-server είναι πλέον ιδιωτική και κρυπτογραφημένη. Ένας εύκολος τρόπος να το διαπιστώσουμε αυτό, είναι να δούμε την IP διεύθυνση

του client μέσω κάποιον κοινών εργαλείων που υπάρχουν (όπως το whatsmyip). Για αυτή τη δουλειά, επιλέξαμε το ipinfo.io, γιατί μπορεί να μας δώσει πολύ εύκολα, μετά από χρήση της εντολής cURL σε αυτή τη διεύθυνση, σε json μορφή όλα τα απαραίτητα στοιχεία της διεύθυνσής μας που μπόρεσε να ανιχνεύσει (IP διεύθυνση, πάροχο, κλπ).

```
panos@pc:~$ curl ipinfo.io
{
  "ip": "83.212.116.170",
  "hostname": "snf-691977.vm.okeanos.grnet.gr",
  "city": "Athens",
  "region": "Attica",
  "country": "GR",
  "loc": "37.9833,23.7333",
  "org": "AS5408 Greek Research and Technology Network S.A"
}
```

Βλέπουμε λοιπόν, ότι ενώ είμαστε στον υπολογιστή του client, οποιαδήποτε κίνησή μας στον έξω κόσμο φαίνεται πλέον σαν να την πραγματοποιούμε εκ μέρους του server. Όλες μας οι κινήσεις γίνονται πλέον υπό πλήρη ιδιωτικότητα.

Από τη στιγμή που ο χρήστης συνδεθεί επιτυχώς στην εφαρμογή, ο server το γνωρίζει και τον ενημερώνει στην πρώτη σελίδα ότι είναι επιτυχώς συνδεδεμένος. Μπορεί λοιπόν πλέον να περιηγηθεί στο διαδίκτυο με ασφάλεια και ιδιωτικότητα, ή να περιηγηθεί στις πάμπολλες λειτουργίες που του προσφέρει η εφαρμογή μας. Ο σύνδεσμος για αυτές τις λειτουργίες παρέχεται από τον server αμέσως αφού εντοπίσει τη σύνδεση του χρήστη.

ΚΕΦΑΛΑΙΟ 5

Οι πρόσθετες υπηρεσίες της εφαρμογής μας

Όπως είπαμε και παραπάνω, εκτός από την προστασία της επικοινωνίας του με τον έξω κόσμο μέσω του δικτύου VPN, οι χρήστες της εφαρμογής μας έχουν τη δυνατότητα να χρησιμοποιήσουν και μια σειρά από υπηρεσίες, που είναι εγκατεστημένες στον server της εφαρμογής.

Ρύθμιση του Apache server για εξυπηρέτηση των σελίδων μόνο στους χρήστες της εφαρμογής.

Κατά την εκπόνηση της παρούσας μεταπτυχιακής διατριβής, πήραμε την απόφαση ο χρήστης να μην εγκαταστήσει το παραμικρό στο σύστημά του. Ως εκ τούτου, όλες οι υπηρεσίες που επιλέξαμε να του προσφέρουμε είναι Web services, στις οποίες μπορεί να έχει πρόσβαση μόνο από τον περιηγητή ιστοσελίδων (web browser) του.

Για κατάλληλη εξυπηρέτηση των υπηρεσιών μόνο σε χρήστες οι οποίοι είναι συνδεδεμένοι στο VPN δίκτυο, χρειάστηκαν και οι κατάλληλες ρυθμίσεις στον Apache server. Αναλύσαμε σε προηγούμενο κεφάλαιο, ότι η IP διεύθυνση του ιδιωτικού δικτύου που διαμορφώνει στο σύστημά μας το λογισμικό OpenVPN, είναι η 10.8.0.0, και η διεύθυνση του interface tun0 που εγκαταστάθηκε στον server και επικοινωνεί με το OpenVPN δίκτυο είναι η 10.8.0.1.

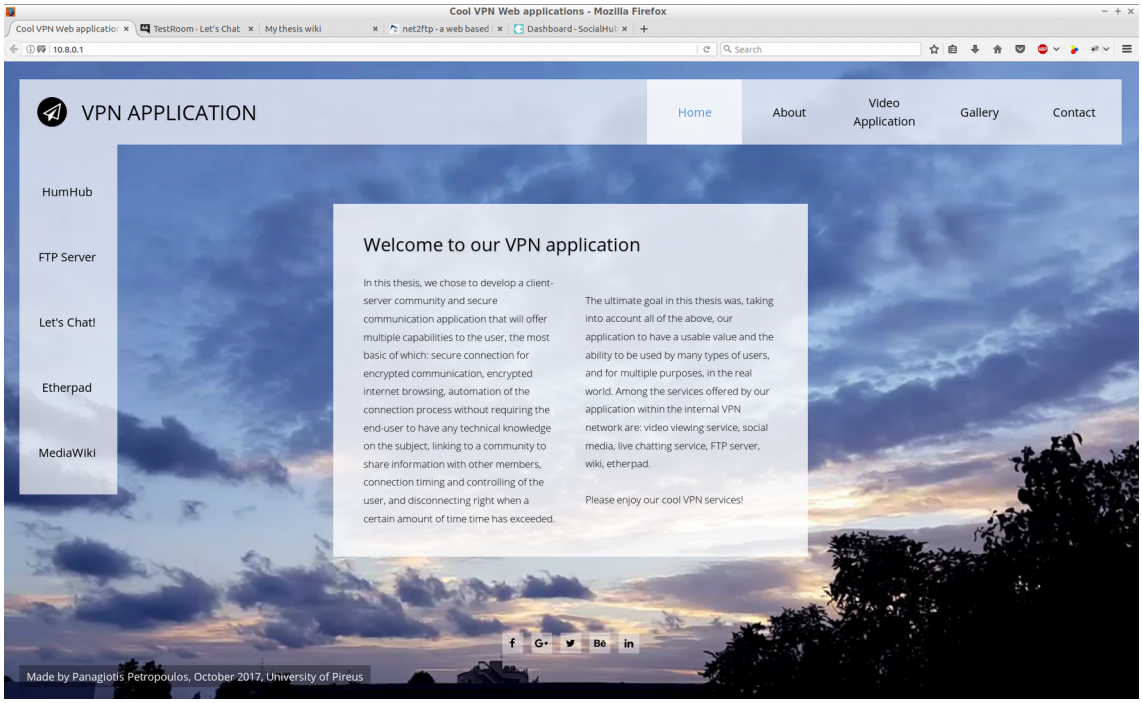
Στο φάκελο με τις ρυθμίσεις του Apache που βρίσκεται στο σύστημα του server, και συγκεκριμένα στο φάκελο /etc/apache2/sites-available, βρίσκονται τα αρχεία ρυθμίσεων με τα οποία γίνεται η εξυπηρέτηση των sites που θέλουμε.

Όταν ξεκινά το service του Apache, δεσμεύεται σε κάποια θύρα και διεύθυνση στο τοπικό μηχάνημα και περιμένει εισερχόμενη κίνηση. Από προεπιλογή, ο Apache "ακούει" στην κίνηση από οποιαδήποτε διεύθυνση και αν προέρχεται.

Η οδηγία "Listen" λέει στον Apache να δέχεται εισερχόμενα αιτήματα μόνο από συγκεκριμένες θύρες ή συνδυασμούς διευθύνσεων και θυρών. Έτσι λοιπόν, προσθέτοντας την οδηγία "Listen 10.8.0.1" περιορίζουμε τον Apache server να δέχεται και να απαντάει σε κίνηση που καταφθάνει μόνο από το συγκεκριμένο interface tun0, και να απορρίπτει οποιαδήποτε άλλη κίνηση έχει αποσταλεί από κάποια άλλη διεύθυνση.

Οι υπηρεσίες που απολαμβάνει ο χρήστης με τη σύνδεσή του

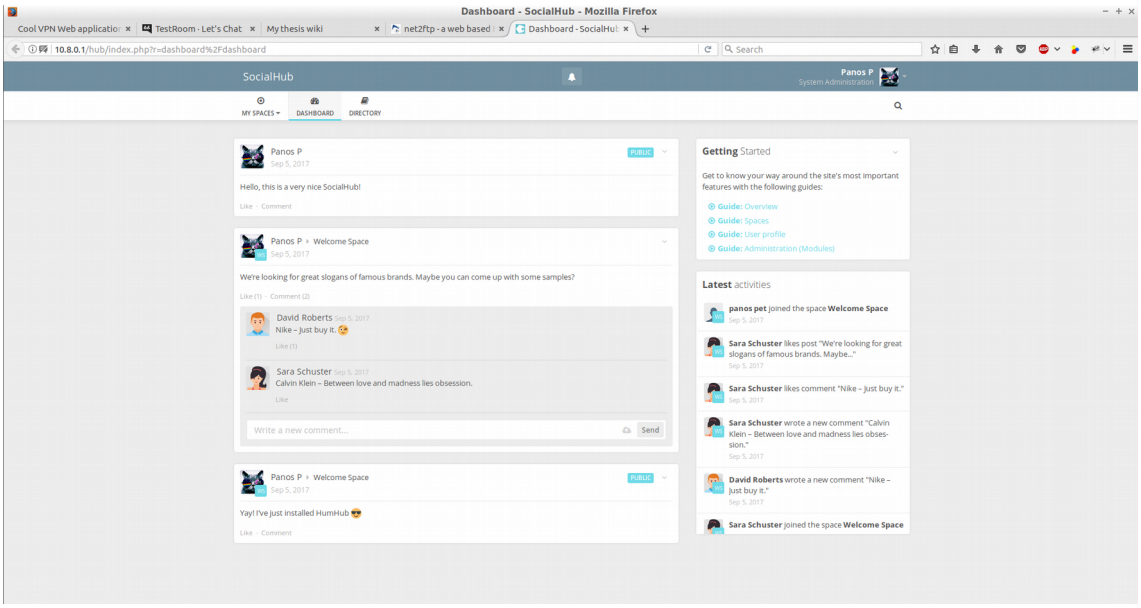
Εισαγωγική σελίδα, portal προς τις άλλες υπηρεσίες της εφαρμογής: Από τη στιγμή της σύνδεσής του και έπειτα, ο χρήστης είναι σε θέση να δει στην αρχική σελίδα της εφαρμογής το σύνδεσμο που τον οδηγεί στο portal των υπηρεσιών που το προσφέρουμε, μια απλή σελίδα χτισμένη σε HTML5, που διαθέτει συνδέσμους προς όλες τις υπηρεσίες.



Εικόνα 8: Η πρώτη σελίδα της εφαρμογής μας.

Μέσο κοινωνικής δικτύωσης: Για την ελεύθερη επικοινωνία μεταξύ των χρηστών, επιλέξαμε να εγκαταστήσουμε στον server της εφαρμογής την υπηρεσία HumHub, η οποία λειτουργεί ακριβώς όπως ένα μέσο κοινωνικής δικτύωσης. Είναι γραμμένο σε PHP και έχει ως βάση το Yii Web Framework.

Στα πλαίσια του HumHub, οι χρήστες γίνονται φίλοι μεταξύ τους, και μπορούν να μοιραστούν με τους υπόλοιπους κείμενο, φωτογραφίες, βίντεο, συνδέσμους προς άλλες σελίδες, και άλλα. Το HumHub αποτελεί ένα ελεύθερο λογισμικό κοινωνικής δικτύωσης και ένα πλαίσιο στο οποίο μπορούμε να έχουμε όλες αυτές τις λειτουργίες επικοινωνίας εύκολα εγκατεστημένες. Είναι ελαφρύ, πολύ φιλικό προς το χρήστη, και απόλυτα παραμετροποιήσιμο, και συνολικά σαν υπηρεσία και ειδικά από κάθε χρήστη. Ο χρήστης μπορεί να δημιουργήσει λογαριασμό ελεύθερα, κάτι που δεν αποτελεί πρόβλημα καθώς είναι απόλυτα ασφαλισμένη η επικοινωνία χρήστη - εφαρμογής μέχρι να φτάσει σε αυτό το σημείο. Η δυνατότητα δικτύωσης και επικοινωνίας των χρηστών, είτε σε πλαίσιο ομάδας κοινών ενδιαφερόντων (όπως πχ. σε ένα φόρουμ), είτε σε πλαίσιο εργαζόμενων εταιρείας, είτε απλών φίλων, λύνεται απόλυτα με τη χρήση του HumHub.



Εικόνα 9: Το social media που διαθέτει η εφαρμογή μας, ονόματι HumHub.

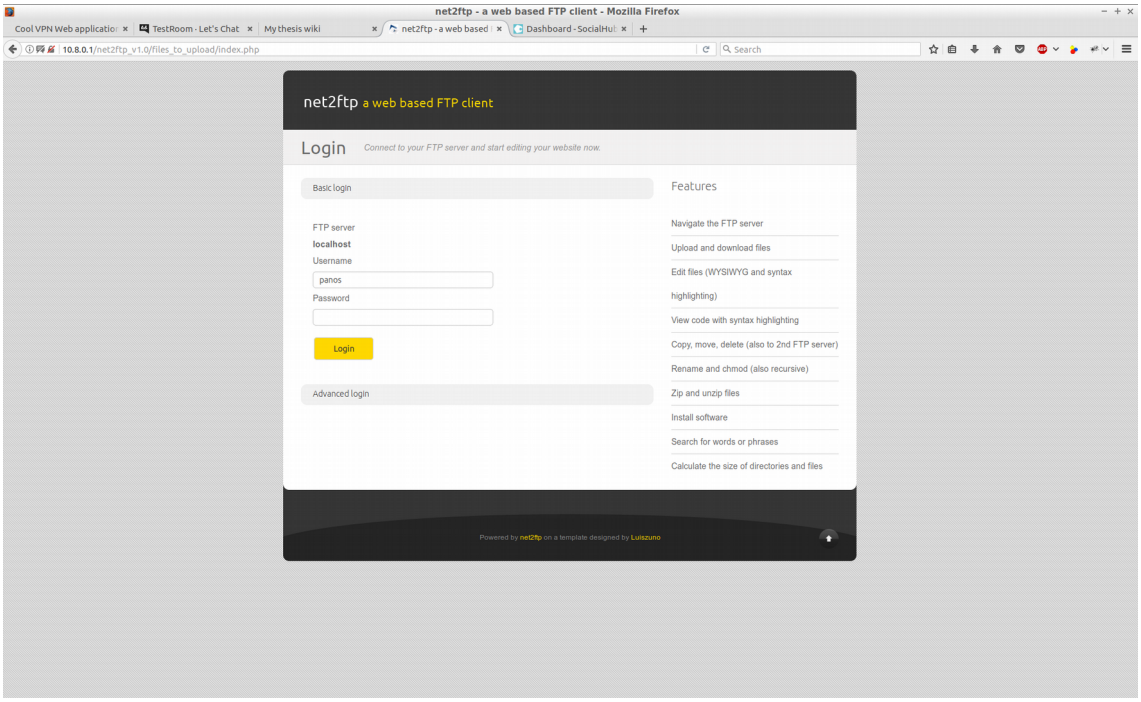
Πρόσβαση στον FTP server της εφαρμογής: Για την εγκατάσταση του FTP (File Transfer Protocol) server της εφαρμογής μας, επιλέξαμε το λογισμικό vsftp, ένα από τα πιο γνωστά λογισμικά για αυτόν το σκοπό. Ο vsftp server λειτουργεί σε επίπεδο Linux, άρα χρειάζεται και επιπλέον δημιουργία χρηστών στο λειτουργικό του server. Το μοντέλο ασφαλείας του vsftp, έχει κάποια ενδιαφέροντα στοιχεία, που φάνηκαν εξαιρετικά χρήσιμα για την εφαρμογή μας. Αυτά είναι:

Διαχωρισμός της κάθε διεργασίας, έτσι ώστε οι διαφορετικές διεργασίες να έχουν τα ελάχιστα δυνατά δικαιώματα για να εκτελεστούν.

Η κάθε διεργασία εκτελείται ξεχωριστά πίσω από ένα "chroot jail", πράγμα που σημαίνει, ότι όπου αυτό είναι δυνατό, οι διεργασίες έχουν δικαιώματα επεξεργασίας μόνο στο φάκελο για τον οποίο ξεκίνησαν να εκτελούνται. Για παράδειγμα, αν ο διαμοιρασμένος φάκελος είναι ο /home/user/ftp, και είναι ο πρωτεύον διαμοιρασμένος φάκελος, τότε ο vsftpd θεωρεί αυτόν ως τον καινούριο root directory, γνωστό ως /. Έτσι το όλο σύστημα με όλα τα υπόλοιπα directories παραμένει προστατευμένο από πιθανές βλαβερές ενέργειες κάποιου κακόβουλου ατόμου.

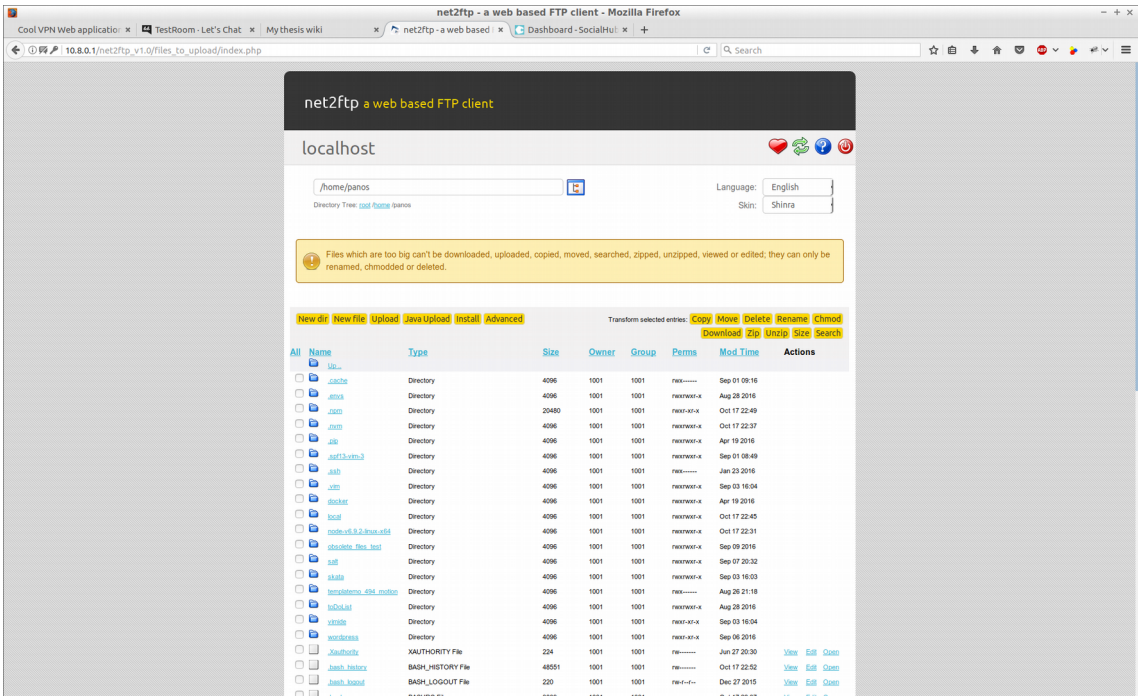
Φυσικά, όπως και όλες οι υπόλοιπες υπηρεσίες της εφαρμογής μας, ο vsftp είναι κι αυτός ρυθμισμένος να δέχεται αιτήματα μόνο από διευθύνσεις IP οι οποίες ανήκουν στο VPN δίκτυο της εφαρμογής. Αυτό διευθετήθηκε από το αρχείο ρύθμισης του vsftp, vsftpd.conf.

Αμέσως μετά την εγκατάσταση του FTP server, έπρεπε να εγκαταστήσουμε και το κατάλληλο web interface για τη χρήση του. Για αυτό το σκοπό χρησιμοποιήσαμε το net2ftp, ένα εύχρηστο και φιλικό προς το χρήστη περιβάλλον. Από τις ρυθμίσεις του net2ftp περιορίσαμε τη σύνδεση να είναι δυνατή μόνο για τον τοπικό ftp server (localhost).



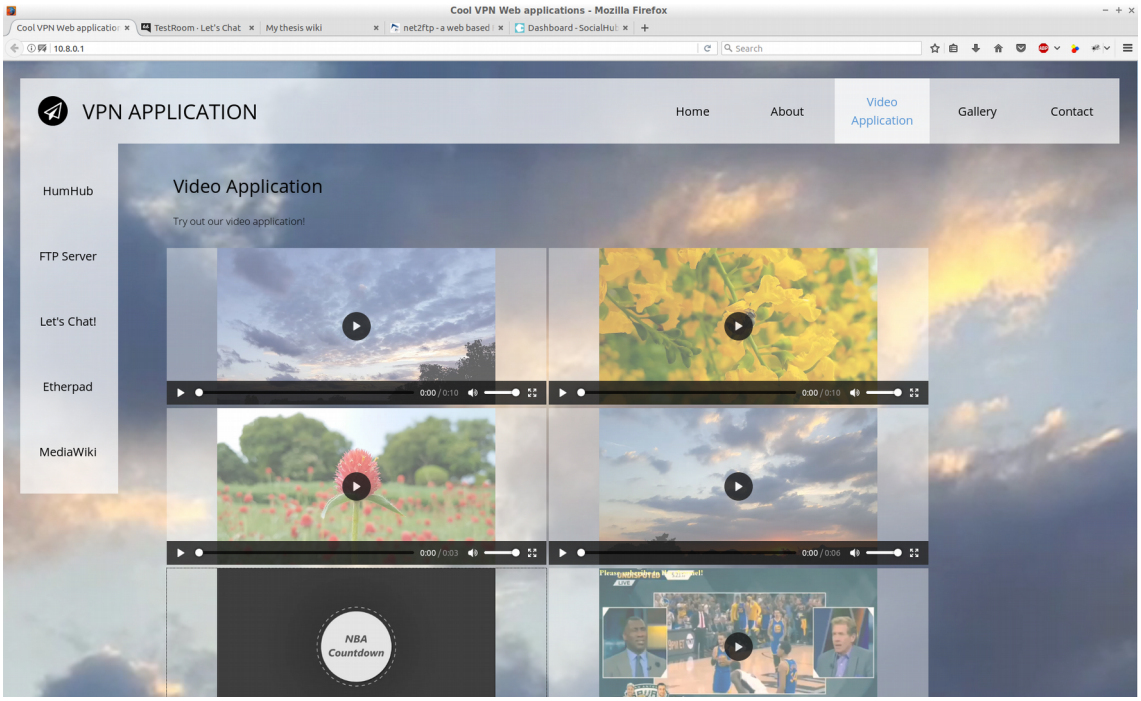
Εικόνα 10: Η σελίδα σύνδεσης με username/password στο web interface του FTP server, ονόματι net2ftp.

Από τη στιγμή που ο χρήστης συμπληρώσει το όνομα χρήστη του και τον κωδικό πρόσβασης, εισέρχεται στο περιβάλλον του net2ftp που μπορεί να δει τους φακέλους και τα αρχεία που έχει ήδη στο home folder του, και μονάχα αυτά. Εκεί, βρίσκεται ένας φάκελος ονόματι “Videos”, στον οποίο, ό,τι αρχείο βίντεο (επέκτασης mp4, mpeg ή avi) ανεβάσει ο χρήστης, εμφανίζεται αυτομάτως και στην επόμενη υπηρεσία που προσφέρει η εφαρμογή μας, αυτή της προβολής και του διαμοιρασμού βίντεο.



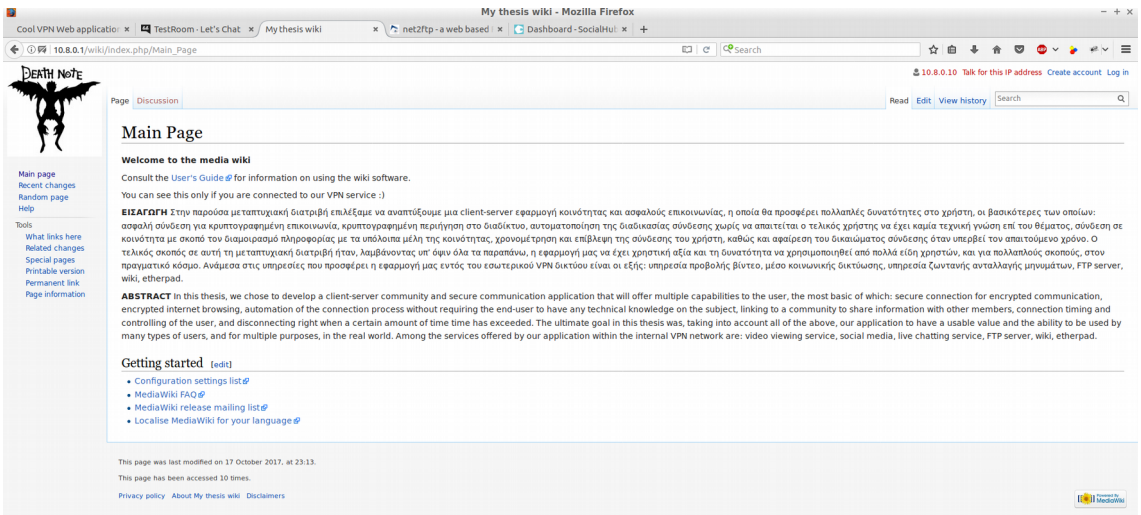
Εικόνα 11: Αφού ο χρήστης συνδεθεί στον FTP server, βλέπει όλα τα αρχεία και τους φακέλους που υπάρχουν στο home folder του.

Υπηρεσία προβολής βίντεο: Από την πρώτη σελίδα της εφαρμογής, ο χρήστης μπορεί να κάνει κλικ στον σύνδεσμο “Video gallery”, και από εκεί να οδηγηθεί στη σελίδα, όπου σε μορφή συλλογής μπορεί να προβάλλει ένα-ένα τα βίντεο που έχουν ανεβάσει οι χρήστες. Ο video player που χρησιμοποιήσαμε για αυτό το σκοπό λέγεται VideoJS, και είναι γραμμένος σε γλώσσα HTML5 και Javascript.



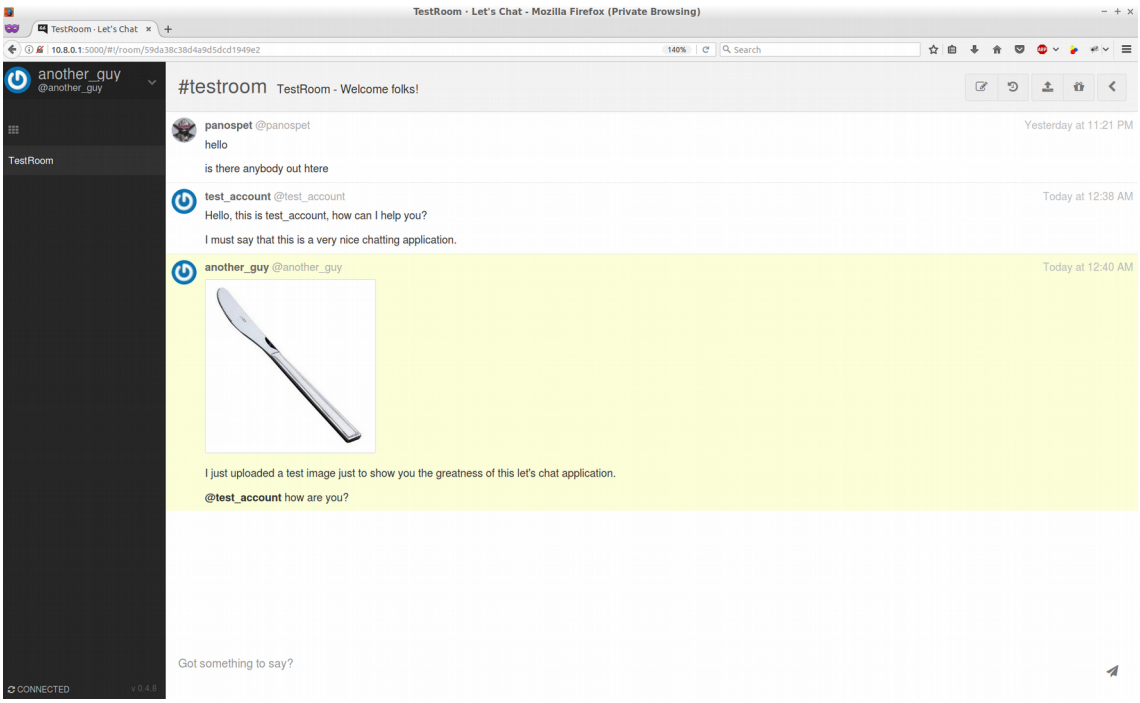
Εικόνα 12: Η υπηρεσία προβολής βίντεο της εφαρμογής μας. Διακρίνονται σε λίστα όλα τα διαθέσιμα βίντεο.

Wiki: Η διαδικασία της από κοινού επεξεργασίας κειμένων, άρθρων, και άλλων δομών είναι κάτι εξαιρετικά χρήσιμο για οποιαδήποτε εφαρμογή κοινότητας. Ένα wiki είναι μια διαδικτυακή εφαρμογή, η οποία επιτρέπει στους χρήστες της να προσθέτουν, να τροποποιούν ή να διαγράφουν το περιεχόμενο μιας ιστοσελίδας σε συνεργασίας με άλλους χρήστες. Σε ένα τυπικό wiki, το κείμενο είναι γραμμένο με μια απλουστευμένη γλώσσα σήμανσης (γνωστή ως "wiki markup"). Το μεγαλύτερο και πιο πετυχημένο παράδειγμα μιας τεράστιας κοινότητας wiki είναι η διαδικτυακή εγκυκλοπαίδεια Wikipedia. Το συγκεκριμένο λογισμικό wiki που χρησιμοποιήσαμε είναι ένα από τα πιο διάσημα και ονομάζεται mediaWiki και είναι γραμμένο σε PHP.



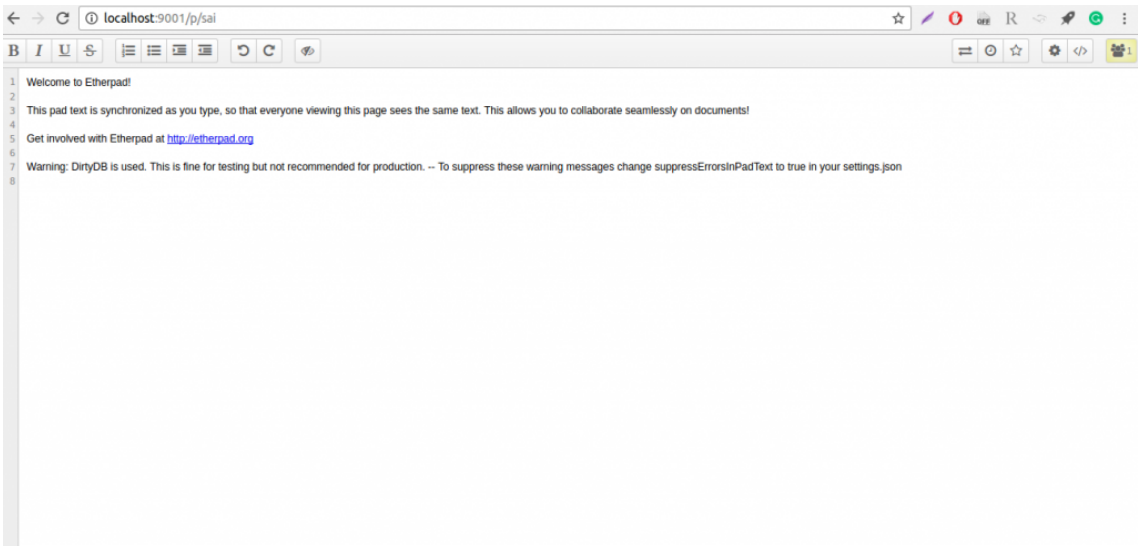
Εικόνα 13: Η σελίδα με το διαθέσιμο mediaWiki της εφαρμογής μας.

Υπηρεσία ζωντανής συζήτησης μηνυμάτων (chatting): Για αυτή την υπηρεσία χρησιμοποιήσαμε το "Let's Chat", μια πανίσχυρη, φιλική προς το χρήστη και πολύ εύκολη στην εγκατάσταση εφαρμογή web chatting. Είναι γραμμένη σε Javascript, και τρέχει μέσω NodeJS, το οποίο με μία μικρή προσαρμογή στον κώδικα, ρυθμίσαμε να εξυπηρετεί μόνο αιτήματα που έρχονται από την εσωτερική διεύθυνση του VPN. Το Let's Chat είναι μια υπηρεσία ανταλλαγής μηνυμάτων σχεδιασμένη έτσι ώστε να εγκαθίσταται με ευκολία, και ταιριάζει πολύ στη λογική της δικής μας εφαρμογής.



Εικόνα 14: Η υπηρεσία ζωντανής γραπτής συνομιλίας chatting, ονόματι Let's chat.

Etherpad: Το Etherpad είναι ένας επεξεργαστής κειμένου, ο οποίος δίνει τη δυνατότητα σε πολλούς χρήστες, ταυτόχρονα και σε πραγματικό χρόνο, να επεξεργάζονται συνεργατικά το ίδιο αρχείο κειμένου. Είναι ανοιχτού κώδικα και είναι εξαιρετικά προσαρμόσιμος με βάση τις εκάστοτε ανάγκες. Εκτελείται αποκλειστικά στο πρόγραμμα περιήγησης ιστοσελίδων (browser) του χρήστη, είναι εξαιρετικά ελαφρύ και ανοικτού κώδικα.



ΚΕΦΑΛΑΙΟ 6

Μειονεκτήματα και προτάσεις για βελτίωση της εφαρμογής - συμπεράσματα

Διαχείριση του server της εφαρμογής

Εξηγήσαμε σε προηγούμενα κεφάλαια, πως, από τη μεριά του χρήστη, δεν απαιτείται καμία απολύτως τεχνική γνώση έτσι ώστε να μπορέσει να κάνει χρήση της εφαρμογής. Αντιθέτως, φροντίσαμε ώστε το εκτελέσιμο αρχείο στο οποίο έχει πρόσβαση ο χρήστης να εκτελεί τη διαδικασία σύνδεσης απολύτως αυτόματα. Δε μπορούμε όμως να ισχυριστούμε το ίδιο από την πλευρά του server. Για το στήσιμο και τη ρύθμιση του VPN server, καθώς και όλου του web interface το οποίο είναι γραμμένο σε Flask Framework, χρειάζεται εξειδικευμένη γνώση. Αυτό λοιπόν σημαίνει ότι η εφαρμογή χρειάζεται διαχειριστή (administrator) ο οποίος θα έχει την κατάλληλη γνώση να την εγκαταστήσει σε κάποιον Linux server (έχοντας φυσικά πρόσβαση στον πηγαίο κώδικά της) αλλά και που θα μπορεί να την κρατά ενήμερη σε ενδεχόμενες ανανεώσεις όλων των λογισμικών που περιλαμβάνει. Για παράδειγμα, οι νέες εκδόσεις του Flask Framework ή του OpenVPN έχουν ως επί το πλείστον ανανεώσεις σχετικά με την ασφάλεια των λογισμικών αυτών, προσαρμογή και οχύρωση ενάντια σε νέους τύπους επιθέσεων που θα έχουν εμφανιστεί από την προηγούμενη έκδοση, και άλλα. Λαμβάνοντας λοιπόν υπ' όψιν ότι ένα από τα μεγαλύτερα πλεονεκτήματα της παρούσας εφαρμογής, είναι η απόλυτη ασφάλεια και κρυπτογραφημένη επικοινωνία, είναι αναγκαία η παρουσία ενός διαχειριστή από τη μεριά του server. Σε αυτό, έρχεται να προστεθεί ότι ένας διαχειριστής είναι αναγκαίος και για οποιαδήποτε τεχνικά προβλήματα ενδέχεται να παρουσιαστούν.

Το Docker ως πιθανή λύση εύκολης διαχείρισης και εγκατάστασης

Το Docker είναι μία τεχνολογία που, στηριζόμενη πάνω στα Linux Containers, μας δίνει τη δυνατότητα της εικονικής λειτουργίας (virtualization) ενός πλήρους λειτουργικού, το οποίο όμως είναι απομονωμένο (isolation, το βασικό χαρακτηριστικό ενός linux container) μοιραζόμενο έτσι με το μηχάνημα που το φιλοξενεί μόνο τον πυρήνα (kernel) του λειτουργικού. Έχει πολύ μεγαλύτερη ευχρηστία και είναι πιο ελαφρύ από ένα απλό virtual machine, λόγω του ότι δεν "δεσμεύει" μνήμη και επεξεργαστική ισχύ από το λειτουργικό που τον φιλοξενεί, αντιθέτως, όταν δεν εκτελείται καμία διεργασία εντός του, κάνει μηδενική σπατάλη των πόρων του συστήματος.

Η τεχνολογία Docker έφερε την επανάσταση στον τρόπο με τον οποίο εγκαθίστανται οι εφαρμογές έως τώρα. Όσο αναπτύσσεται η τεχνολογία Docker όλο και περισσότερες διάσημες εφαρμογές την υιοθετούν, κάνοντας έτσι την εγκατάσταση και ρύθμιση (με μία λέξη deployment) των εφαρμογών καθώς και την ανανέωσή τους ευκολότερη. Η πλήρης απομόνωση που απολαμβάνει ο Docker container σε σχέση με το υπόλοιπο λειτουργικό σύστημα, θα προσέδιδε πολύ μεγαλύτερη ασφάλεια στην εφαρμογή μας.

Έτσι, ερχόμαστε στο συμπέρασμα πως, το "dockerization" της εφαρμογής μας, η δημιουργία δηλαδή ενός Docker container όπου μέσα του θα περιείχε το λειτουργικό Ubuntu, με όλα όσα χρειαζόμαστε για να εκτελεστεί η εφαρμογή μας προεγκατεστημένα (Python 2.7, PHP5.5, Flask Framework, Apache2 web server, πηγαίος κώδικας της εφαρμογής και άλλα) θα μπορούσε να προσδώσει φορητότητα και μεγαλύτερη ευκολία στην εγκατάστασή της σε οποιοδήποτε μηχάνημα, ανεξαρτήτως λειτουργικού συστήματος. Όλη η εγκατάσταση των απαιτούμενων προγραμμάτων για την εκτέλεση της εφαρμογής μας, αρκεί να συμπεριληφθεί στο λεγόμενο Dockerfile, το script δηλαδή που εκτελείται κατά το deployment και διαμορφώνει τον Docker container με βάση τις οδηγίες που περιγράφονται σε αυτό το αρχείο.

Χωρίς να μπορούμε να πούμε με ευκολία ότι η δημιουργία ενός Docker container της εφαρμογής λύνει το θέμα το administrator της εφαρμογής από την πλευρά του server, σίγουρα είναι κάτι που περιορίζει στο ελάχιστο τον χρόνο συντήρησης και ανανέωσης της εφαρμογής.

Μηχανισμός χρέωσης του χρήστη επιπρόσθετα στη χρονομέτρηση σύνδεσης

Είπαμε και σε προηγούμενο κεφάλαιο πως τα σεμινάρια που απαιτούν φυσική παρουσία όλο και περισσότερο αντικαθίσταται με τα web seminars, τα οποία είναι πλατφόρμες εκπαίδευσης μέσω διαδικτύου, που στις περισσότερες των περιπτώσεων κοστολογούν και το χρήστη κάποια χρήματα αναλόγως την ποσότητα ή τις ώρες των μαθημάτων που έχουν παρακολουθήσει.

Η δική μας εφαρμογή, λόγω της υπηρεσίας προβολής βίντεο, μπορεί εύκολα να παίξει αυτό το ρόλο, δηλαδή μιας πλατφόρμας εκπαιδευτικού υλικού με τη μορφή βίντεο, κειμένου, και άλλων τρόπων. Οπότε, μια πιθανή βελτίωση της εφαρμογής, θα μπορούσε να είναι η χρονομέτρηση του χρήστη, όχι πια με βάση τα λεπτά συνδεσιμότητας που έχει αιτηθεί, αλλά με βάση τα χρήματα τα οποία έχει σπαταλήσει έτσι ώστε να αποκτήσει πρόσβαση. Με μια απλή αναπροσαρμογή της τιμής αυτής, υπολογίζεται σε πόσα λεπτά συνδεσιμότητας αντιστοιχούν τα χρήματα που έχει σπαταλήσει ο χρήστης. Ο υπόλοιπος μηχανισμός, αυτός δηλαδή της αποσύνδεσης του χρήστη όταν τα διαθέσιμα λεπτά τελειώσουν (άρα και τα διαθέσιμα χρήματα), και η ανάκληση του πιστοποιητικού του χρήστη. μπορούν να συνεχίσουν να λειτουργούν ως έχουν.

Η εφαρμογή να μπορεί να είναι διαθέσιμη και για Mac OS

Στην παρούσα μεταπτυχιακή διατριβή, επιλέξαμε να καλύψουμε το 90% των ιδιωτών χρηστών και των εταιρειών, υλοποιώντας την εφαρμογή αυτή για τα λειτουργικά συστήματα Windows και Linux. Λόγω έλλειψης συστήματος Mac κατά την εκπόνηση της διατριβής αυτής, δεν κατέστη δυνατό να προσαρμόσουμε το εκτελέσιμο script του χρήστη και για το λειτουργικό Mac OS, φτάνοντας έτσι την κάλυψη στο 99,9% των διαθέσιμων χρηστών.

Αυτό θα μπορούσε να αποτελέσει ακόμα μία βελτίωση στην παρούσα εφαρμογή. Μπορούμε να πούμε ότι, λόγω των πολλών κοινών που παρουσιάζει το λειτουργικό Mac OS με το Unix, ειδικά σε επίπεδο διαχείρισης, μια τέτοια αλλαγή δε θα ήταν καθόλου δύσκολη ή χρονοβόρα.

Η εφαρμογή να μπορεί να είναι διαθέσιμη και για κινητά τηλέφωνα - smartphones

Όσο εξελίσσεται η τεχνολογία των κινητών τηλεφώνων, όλο και μεγαλύτερο μεγαλύτερο μέρος της διαδικτυακής επικοινωνίας γίνεται μέσω αυτών. Αυτή τη στιγμή ένα τεράστιο κομμάτι του πληθυσμού χρησιμοποιεί αποκλειστικά “έξυπνες συσκευές” (smartphones ή tablets) στην καθημερινότητά του, και αυτές οι συσκευές όλο και περισσότερο αντικαθιστούν τους “κλασικούς” ηλεκτρονικούς υπολογιστές (desktops ή laptops) στην αγορά.

Λαμβάνοντας υπ’ όψιν τα παραπάνω, και με δεδομένο πως ένας από τους σκοπούς της παρούσας εφαρμογής είναι αυτή να είναι όσο το δυνατόν πιο φιλική προς το χρήστη, θα αποτελούσε μία σαφέστατη βελτίωση της εφαρμογής η δημιουργία ενός Application για έξυπνα κινητά τηλέφωνα, της οποίας οι ενέργειες που θα εκτελούνταν να ήταν οι ίδιες με αυτές του εκτελέσιμου script του χρήστη. Αντίστοιχα, εφαρμογές για smartphone οι οποίες εκτελούν σύνδεση με OpenVPN υπάρχουν πάμπολλες, οπότε και η δική μας θα μπορούσε να κάνει χρήση μιας εξ’ αυτών. Μια τέτοια επέκταση της εφαρμογής θα μπορούσε να καλύψει ακόμα περισσότερες ανάγκες χρηστών, είτε πρόκειται για χρήση της εφαρμογής σε επίπεδο εταιρείας είτε σε επίπεδο κοινότητας χρηστών που μοιράζονται τα ίδια ενδιαφέροντα. Αντίστοιχα, πολλές από τις web υπηρεσίες οι οποίες προσφέρονται εντός της εφαρμογής μας (Let’s chat, etherpad, net2ftp), έχουν ήδη Responsive Design το οποίο καλύπτει και κινητά τηλέφωνα. Έτσι δε θα χρειαζόταν παραπάνω κόπος για την αναπροσαρμογή αυτών.

Συμπεράσματα - επίλογος

Στην παρούσα μεταπτυχιακή διατριβή φέραμε εις πέρας μια client - server διαδικτυακή εφαρμογή κοινότητας στην οποία μπορεί να συνδεθεί ο χρήστης με απόλυτη ασφάλεια σε ένα δίκτυο VPN, και από εκεί, είτε να περιηγηθεί στο διαδίκτυο κάτω από πλήρη κρυπτογραφημένη επικοινωνία, είτε να χρησιμοποιήσει τις διαθέσιμες υπηρεσίες της εφαρμογής μας.

Θεωρούμε πως η εφαρμογή αυτή που εκπονήσαμε μπορεί να σταθεί αυτοτελώς, πλήρης, και να εκπληρώσει στο έπακρο τις ανάγκες μιας εταιρείας ή μιας ομάδας χρηστών κοινών ενδιαφερόντων.

Πιστεύουμε στην υψηλή χρησιμότητά της, γιατί συνδυάζει πολλά χαρακτηριστικά διαφόρων διαδικτυακών εφαρμογών σε μία: η εφαρμογή μας θα μπορούσε συνδυαστικά να παίξει το ρόλο μιας πλατφόρμας ηλεκτρονικών σεμιναρίων και βίντεο, ενός φόρουμ συζήτησης και ανταλλαγής απόψεων, μιας εταιρικής πλατφόρμας εκπαίδευσης και σύνδεσης εργαζόμενων σε επίπεδο εταιρείας, και άλλα πολλά.

ΚΕΦΑΛΑΙΟ 7

Βιβλιογραφία

Ολόκληρος ο πηγαίος κώδικας της εφαρμογής μας στην ιστοσελίδα GitHub:
https://github.com/panospet/auto_VPN

Φυσικά, www.wikipedia.org

Ασφάλεια πληροφοριακών συστημάτων, tee.gr
http://portal.tee.gr/portal/page/portal/teetkm/DRASTHRIOTHTES/SEMINARIA/PALAIOTERA_SEMINARIA/SHMEIWSEIS_ASFALEIA_PLHROFORIAKWN_SYSTHMATWN/Asfaleia_diadiktyakwn_efarmogwn.pdf

Networking 101: Understanding Tunneling, Enterprise Working Planet
http://www.enterprisenetworkingplanet.com/netsp/article.php/3624566/Networking-101-Understanding-Tunneling.htm

VPN networks, academia.edu ebook
http://www.academia.edu/1343134/Virtual_Private_Network_VPN_

Microsoft technet, how VPN works
https://technet.microsoft.com/en-us/library/cc779919(v=ws.10).aspx

CactusVPN – introduction to VPN
https://www.cactusvpn.com/beginners-guide-to-vpn/introduction-to-vpn/

HTTPS certificates, by Hartley Brody
https://blog.hartleybrody.com/https-certificates/

Techieinspire: HTTPS and SSL
http://www.techieinspire.com/https-and-ssl-and-how-it-works/

WSGI tutorial by codepoint
http://wsgi.tutorial.codepoint.net/

The guardian: Passwords hashing and salting
https://www.theguardian.com/technology/2016/dec/15/passwords-hacking-hashing-salting-sha-2

Iterative Password Hashing: Sjoerd Langkemper
https://www.sjoerdlangkemper.nl/2016/05/25/iterative-password-hashing/

RFC2898: Password-Based Cryptography Specification
https://tools.ietf.org/html/rfc2898

get pip script
https://bootstrap.pypa.io/get-pip.py

PycURL library website
http://pycurl.io/

OpenVPN documentation
https://openvpn.net/index.php/open-source/documentation/howto.html

Linux Networking “bible”, by Debian
http://www.aboutdebian.com/network.htm

net2ftp documentation
http://www.net2ftp.com/homepage/help-administrator.html

DigitalOcean: how to install media wiki on ubuntu server
https://www.digitalocean.com/community/tutorials/how-to-install-mediawiki-on-ubuntu-14-04

HumHub installation
http://docs.humhub.org/admin-installation.html

Let’s chat installation
https://www.tecmint.com/install-lets-chat-on-centos-ubuntu-debian/
https://github.com/sdelements/lets-chat

Apache and nodejs on the same server
https://gist.github.com/stagas/754303

Ubuntu etherpad installation
https://help.ubuntu.com/community/Etherpad-liteInstallation