ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΙΛΟΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΗΛΕΦΩΝΙΚΟΥ ΚΕΝΤΡΟΥ ΒΑΣΙΣΜΕΝΟΥ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ASTERISK

Πτυχιακή εργασία

Χαρλαύτης Βασίλειος



Αθήνα, 2017



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Καμαλάκης Θωμάς
Επίκουρος Καθηγητής
Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο Αθηνών

Δημητρακόπουλος Γεώργιος
Επίκουρος Καθηγητής
Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο Αθηνών

Δαλάκας Βασίλειος
Εργαστηριακό Διδακτικό Προσωπικό
Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο Αθηνών

Ο Χαρλαύτης Βασίλειος,

δηλώνω υπεύθυνα ότι:

- Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

- Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να ευχαριστήσω την οικογένειά μου που με στήριξε, δίνοντάς μου τα εφόδια για να φτάσω ως εδώ.

Μια ιδιαίτερη ευχαριστία θέλω να δώσω στους:

κ. Καμαλάκη Θωμά (επιβλέποντα καθηγητή μου),

κ. Δημητρακόπουλο Γεώργιο (μέλος της τριμελούς επιτροπής),

κ. Δαλάκα Βασίλειο (μέλος της τριμελούς επιτροπής).

Περιεχόμενα

Περίληψη 7

Abstract 8

Εισαγωγή :

Τι είναι το Asterisk private branch exchange (PBX); 9

Η ιστορία πίσω από το Asterisk 9

Χαρακτηριστικά και δυνατότητες του Asterisk. 11

ΚΕΦΑΛΑΙΟ 1 :

Διαδικασία εγκατάστασης του Asterisk σε Debian περιβάλλον 14

1.1 Θέσπιση πλαισίου εργασίας. 14

1.2 Βήματα εγκατάστασης του Asterisk. 19

1.3 Ταυτόχρονη εκκίνηση του Asterisk και του διακομιστή (server). 32

ΚΕΦΑΛΑΙΟ 2 :

Η επίθεση και η θωράκιση του Asterisk. 38

2.1 Η επίθεση. 38

2.2. Ανάθεση του Asterisk σε χρήστη χαμηλής ιεραρχίας. 39

2.3. Αλλαγή προκαθορισμένων αριθμών θύρας (default port number). 44

2.4 Ενημέρωση του τοίχους ασφαλείας (firewall) για τους νέους κανόνες. 49

2.5 Εγκατάσταση του Fail2Ban για την ενίσχυση του τοίχους ασφαλείας (firewall). 54

ΚΕΦΑΛΑΙΟ 3 :

Παραμετροποίηση του Asterisk. 60

3.1 Περιγραφή της δομής του αρχείου ρυθμίσεων του Asterisk. 38

3.2 Παραμετροποίηση του Asterisk. 64

ΚΕΦΑΛΑΙΟ 4 :

Σύνδεση στην υπηρεσία του Asterisk. 98

4.2 Ιστοσελίδα εγγραφής. 99

4.3 Οδηγίες Σύνδεσης. 122

Περίληψη

Αυτή η πτυχιακή εργασία έχει ως στόχο την εγκατάσταση και την παραμετροποίηση της πλατφόρμας Asterisk σε ένα εικονικό μηχάνημα, έτσι ώστε να είναι δυνατή η παροχή λειτουργιών τηλεφωνικού κέντρου όπως η φωνητική κλήση, η βίντεο κλήση καθώς και η ανταλλαγή άμεσων μηνυμάτων. Κατά την απόπειρα παροχής των παραπάνω προέκυψε η ανάγκη για την υλοποίηση ακόμα δύο υπηρεσιών όπως είναι η θωράκιση του συστήματος από κυβερνοεπιθέσεις και η συμμετοχή στην υπηρεσία μέσω του ενιαίου καταλόγου χρηστών του Χαροκοπείου.

Λέξεις κλειδιά: τηλεφωνικό κέντρο, φωνητική κλήση, βίντεο κλήση, άμεσα μηνύματα, κυβερνοεπίθεση, κατάλογος χρηστών.

Abstract

This diploma thesis aims at installing and customizing the Asterisk platform on a virtual machine so that it is possible to provide call center functions such as voice calling, video calling, and instant messaging. When attempting to provide the above, the need for implementation of two other services such as the shielding of the system from cyber attacks and the log in the service through the single user directory of Harokopio has emerged.

Keywords: call center, voice call, video call, instant messaging, cyber attack, user directory.

Εισαγωγή

Τι είναι το Asterisk private branch exchange (PBX);

Το Asterisk PBX (για συντομία θα αναφέρεται Asterisk) είναι μια εφαρμογή λογισμικού, που υλοποιήθηκε με στόχο την παροχή υπηρεσιών τηλεφωνικού κέντρου με ελάχιστο κόστος. Είναι ανοιχτού κώδικα και γραμμένη στη γλώσσα προγραμματισμού C. Το όνομά του προέρχεται από το σύμβολο του αστερίσκου [*] το όποιο υπάρχει με τη μορφή κουμπιού στο πληκτρολόγιο κάθε τηλεφωνικής συσκευής. Αρχικά, σχεδιάστηκε για να «τρέχει» σε λειτουργικά συστήματα Linux, αλλά με την πάροδο του χρόνου επεκτάθηκε και σε λειτουργικά συστήματα βασισμένα σε Unix, όπως τα: NetBSD, OpenBSD, FreeBSD, macOS και Solaris [1.].

Δίνει την δυνατότητα σε συσκευές, όπως είναι οι υπολογιστές, τα έξυπνα κινητά τηλεφώνα (smartphone), οι προσωπικοί ψηφιακοί βοηθοί (Personal Digital Assistant - PDA) και ορισμένες ψηφιακές σταθερές τηλεφωνικές συσκευές με υποστήριξη του πρωτοκόλλου διαδικτύου (Internet Protocol phone - IP Phone) να πραγματοποιούν κλήσεις σε πρώτη φάση, εντός ιδιωτικών δικτύων ανταλλαγής κλήσεων (Private Branch eXchange - PBX) και στη συνέχεια, σε δημόσια τηλεφωνικά δίκτυα μεταγωγής (Public Switched Telephone Network - PSTN).

Η εταιρία που συντηρεί, υποστηρίζει και προδιαγράφει την εξέλιξη του Asterisk είναι η Digium [2.] της οποίας ιδρυτής είναι ο Mark Spencer, δημιουργός της πρώτης έκδοσης του πυρήνα του Asterisk.

Η ιστορία πίσω από το Asterisk

Το 1999 ο επιχειρηματικός κόσμος άρχισε να αντιλαμβάνεται το μέγεθος των χρηματικών πόρων που θα εξοικονομούσε, αν αφομοίωνε στα συστήματά του το λειτουργικό σύστημα ανοιχτού κώδικα Linux. Το μοναδικό μειονέκτημα που είχε το Linux έναντι των άλλων λειτουργικών συστημάτων της εποχής ήταν ότι δεν παρείχε τεχνική υποστήριξη πελατών.

Έτσι, ο ειδικός από το 1994 στα Linux, Mark Spencer [3.], αποφάσισε να ιδρύσει την LSS (Linux Support Services), μια εταιρία που ως στόχο είχε να καλύψει το κενό που υπήρχε στην τεχνική υποστήριξη του συγκεκριμένου λειτουργικού συστήματος. Ο Spencer μόλις είχε λάβει το πτυχίο του ως μηχανικός υπολογιστών από το πανεπιστήμιο του Όμπερν των ΗΠΑ και, έτσι, είχε πολύ περιορισμένο κεφάλαιο προς επένδυση. Για να στεγάσει κάπου την εταιρία του, αναγκάστηκε να προσφέρει τις υπηρεσίες του σε ένα τοπικό κατάστημα ηλεκτρονικών υπολογιστών στο Χάντσβιλ, με μοναδικό αντάλλαγμα έναν χώρο στο πίσω μέρος του καταστήματος.

Τους επόμενους μήνες η LSS είχε κερδίσει αρκετούς πελάτες ώστε να μπορεί να αντεπεξέλθει στα έξοδα της ενοικίασης ενός μικρού γραφείου και της πρόσληψης ενός αριθμού ειδικών στα Linux. Η συνεχής ανοδική πορεία, όμως, που συνόδευε την LSS σύντομα απαιτούσε ένα δομημένο τηλεφωνικό κέντρο που να μπορεί να μοιράσει σωστά τις κλήσεις στους εργαζόμενους. Παρά όμως την επιτυχία που γνώριζε τότε η LSS, το κόστος αγοράς ενός τέτοιου συστήματος ήταν πολύ μεγαλύτερο από ότι θα μπορούσε να καλύψει η εταιρία.

Η επιχειρηματική λογική του Spencer του έλεγε να πάρει ένα δάνειο. Έχοντας, όμως, μέσα του το μικρόβιο του προγραμματιστή, όλο και φούντωνε η ιδέα να ρισκάρει και να δημιουργήσει το δικό του λογισμικό τηλεφωνικού κέντρου. Το πάθος του προγραμματιστή για δημιουργία υπερίσχυσε και έτσι, μετά από μια χρηματική επένδυση που έλαβε η LSS από την ADTRAN [4.] [*προμηθευτής τηλεπικοινωνιακού εξοπλισμού δικτύωσης και προϊόντων διαδικτύου*] και μερικούς μήνες σκληρής εργασίας, η πρώτη έκδοση του Asterisk κυκλοφόρησε στο διαδίκτυο με άδεια GPL, την άδεια χρήσης που χρησιμοποιούν τα Linux. Το Asterisk αγαπήθηκε και υποστηρίχτηκε αμέσως από την κοινότητα ανοιχτού κώδικα, η οποία το ανέπτυξε υπό την επίβλεψη πάντα του Spencer, σε τέτοιο βαθμό ώστε να εξελιχθεί στο προϊόν που είναι σήμερα.

Το Asterisk θεωρείται σήμερα το πιο αναπτυγμένο ανοιχτό λογισμικό τηλεφωνικού κέντρου, με εκατοντάδες ενεργούς προγραμματιστές ανοιχτού κώδικα που το εξελίσσουν συνεχώς. Δεκάδες άλλα ανοιχτά λογισμικά έχουν αναπτυχθεί, με στόχο ή να προσθέσουν επιπλέον υπηρεσίες στο Asterisk [π.χ. *το FreePBX [5.], που παρέχει στο Asterisk γραφικό περιβάλλον μέσω φυλλομετρητή (browser) για πιο εύκολη διαχείριση*] ή να εκμεταλλευτούν τις

δυνατότητές του γύρω από τα τηλεφωνικά δίκτυα [π.χ. *το LinuxMCE [6.], μια έκδοση Linux , που θέτει σαν σκοπό να μετατρέψει τα κλασικά συμβατικά σπίτια σε τεχνολογικά αναπτυγμένες έξυπνες οικίες, κάνει χρήση του Asterisk για να παρέχει υπηρεσίες ασφάλειας χώρου, τηλεομοιότυπου (Fax) κ.α.*].

Η LSS έχει μετονομαστεί, από το 2001, σε Digium και παράγει προϊόντα που είναι συμβατά με το Asterisk, όπως κάρτες τηλεφωνίας για υπολογιστές, τηλεφωνικές συσκευές, καθώς και υπηρεσίες σχεδιασμού, εγκατάστασης και συντήρησης ιδιωτικών τηλεφωνικών κέντρων σε επιχειρήσεις.

Χαρακτηριστικά και δυνατότητες του Asterisk.

Το Asterisk περιλαμβάνει πολλές δυνατότητες, που εξυπηρετούν τις ανάγκες τις οποίες μπορεί να έχει μια επιχείρηση από ένα ιδιωτικό τηλεφωνικό δίκτυο. Υπηρεσίες, όπως: φωνητικά μηνύματα (voice mail), κλήσεις συνδιάσκεψης (conference calling), κλήσεις βίντεο (video calls), άμεσα μηνύματα (instant messaging), υποστήριξη τηλεομοιοτυπιών (Fax) και αυτόματης κατανομής κλήσεων (automatic call distribution), είναι μόνο ένα πολύ μικρό δείγμα του συνόλου των δυνατοτήτων [7.] του Asterisk, αφού συνεχώς προστίθενται καινούριες υπηρεσίες, ανάλογα με τις ανάγκες και τις καινοτομίες της εποχής.

Οι διαχειριστές του συστήματος μπορούν να δημιουργήσουν νέες λειτουργίες, με τρεις τρόπους [1.] :

1. Γράφοντας δέσμες ενεργειών (script) σε μια γλώσσα προγραμματισμού που λέγεται Dialplan, η οποία έχει δημιουργηθεί αποκλειστικά για τον προγραμματισμό επεκτάσεων στο Asterisk.

2. Τροποποιώντας τον πηγαίο κώδικα του Asterisk, που είναι γραμμένος σε γλώσσα C.

3. Χρησιμοποιώντας μια βιβλιοθήκη Asterisk Gateway Interface (AGI) σε οποιαδήποτε γνωστή γλώσσα προγραμματισμού που είναι ικανή να επικοινωνεί μέσω του τυπικού συστήματος ροών (stdin, stdout) ή μέσω του πρωτοκόλλου ελέγχου μετάδοσης (Transmission Control Protocol – TCP).

Το Asterisk υποστηρίζει τα παρακάτω πρότυπα για την μεταφορά ήχου μέσω του πρωτοκόλλου διαδικτύου (Voice Over Internet Protocol - VOIP), ώστε να είναι δυνατή η πραγματοποίηση κλήσεων μεταξύ τηλεφώνων IP (φυσικών ή εικονικών):

1. SIP (Session Initiation Protocol) [8.]: Πρωτοεμφανίστηκε το 1996, ως πρωτόκολλο τηλεδιασκέψεων. Κατέχει σημαντική θέση στην δικτυακή τηλεφωνία, γιατί είναι το πιο ευρέως διαδεδομένο και υποστηριζόμενο όχι μόνο από φυσικές άλλα και από εικονικές συσκευές (softphone). Λόγω αυτής της δημοτικότητας και της υποστήριξης από ένα μεγάλο μέρος των τηλεφωνικών συσκευών του πανεπιστημίου (Cisco 7942), το SIP είναι το πρωτόκολλο που χρησιμοποιήθηκε σε αυτήν την εργασία.

2. IAX (Inter-Asterisk eXchange) [9.]: Δημιουργήθηκε από τον Mark Spencer το 1999 και μεγάλο μέρος της ανάπτυξής του πραγματοποιήθηκε από την κοινότητα ανοικτού κώδικα του Asterisk. Ο κύριος στόχος του IAX είναι να ελαχιστοποιήσει το εύρος ζώνης (bandwidth), που χρησιμοποιείται στην μετάδοση πολυμέσων κατά την πραγματοποίηση μιας κλήσης.

3. XMPP (eXtensible Messaging and Presence Protocol) [10.]: Αρχικά ονομάστηκε Jabber, και αναπτύχθηκε από την κοινότητα ανοιχτού κώδικα Jabber το 1999, με στόχο την ανταλλαγή άμεσων μηνυμάτων (Instant Message - IM). Σε πρώτο στάδιο, δεν υποστήριζε υπηρεσίες, όπως για παράδειγμα την φωνητική κλήση ή την κλήση βίντεο, άλλα σε μια προσπάθεια να κερδίσει τον ανταγωνισμό πρόσθεσε και αυτές τις παροχές.

4. MGCP (Media Gateway Control Protocol) [11.]: Είναι ένα πρωτόκολλο που βασίζεται σε κείμενο (text-base) για την σηματοδοσία (signaling) και τον έλεγχο των κλήσεων. Είναι διάδοχος του πρωτοκόλλου Simple Gateway Control Protocol (SGCP), το οποίο αναπτύχθηκε από την Bellcore και τη Cisco.

5. SCCP (Skinny Client Control Protocol) [12.]: Η Cisco απέκτησε την τεχνολογία SCCP όταν εξαγόρασε την Selsius Corporation, το 1998. Είναι το πρωτόκολλο που χρησιμοποιείται περισσότερο από τα τηλεφωνικά κέντρα της Cisco και την πιο διαδεδομένη γενιά τηλεφωνικών συσκευών της (Cisco 79xx).

Τέλος, το Asterisk, μέσω ειδικού υλικού υπολογιστή, μπορεί να επικοινωνήσει με δημόσια δίκτυα τηλεφωνίας, όπως τα δημόσια τηλεφωνικά δίκτυα μεταγωγής (Public Switched Telephone Network - PSTN) και με αυτόν τον τρόπο να πραγματοποιήσει κλήσεις πέρα από τα στενά όρια του ιδιωτικού δικτύου που έχει εγκατασταθεί.

Πηγές:

1. https://en.wikipedia.org/wiki/Asterisk_(PBX))
2. https://en.wikipedia.org/wiki/Digium
3. https://wiki.asterisk.org/wiki/display/AST/A+Brief+History+of+the+Asterisk+Project
4. https://en.wikipedia.org/wiki/ADTRAN
5. https://en.wikipedia.org/wiki/FreePBX
6. https://en.wikipedia.org/wiki/LinuxMCE
7. http://www.asterisk.org/get-started/features
8. https://en.wikipedia.org/wiki/Session_Initiation_Protocol
9. https://en.wikipedia.org/wiki/Inter-Asterisk_eXchange
10. https://en.wikipedia.org/wiki/XMPP
11. https://en.wikipedia.org/wiki/Media_Gateway_Control_Protocol
12. https://en.wikipedia.org/wiki/Skinny_Call_Control_Protocol

ΚΕΦΑΛΑΙΟ 1

Διαδικασία εγκατάστασης του Asterisk σε Debian περιβάλλον

Σκοπός αυτού του κεφαλαίου είναι η περιγραφή της διαδικασίας εγκατάστασης του Asterisk σε περιβάλλον Debian, καθώς και ανάλυση κάθε βήματος, έτσι ώστε να είναι κατανοητή η διαδικασία και να μπορεί να αναπαραχθεί από οποιονδήποτε την διαβάσει.

1.1 Θέσπιση πλαισίου εργασίας.

Τεκμηρίωση της επιλογής του Debian ως το λειτουργικό του διακομιστή.

Το Debian είναι μια, ιδιαίτερα γνωστή για την σταθερότητα της, έκδοση Linux. Λόγω της εμπιστοσύνης που η σταθερότητά της εμπνέει, θεωρείται αξιόπιστο λογισμικό και για αυτό χρησιμοποιείται σε μεγάλο βαθμό σε συστήματα διακομιστή (server) [1.]. Αυτή η αξιοπιστία που παρέχει, είναι και ο λόγος που επιλέχθηκε το Debian ως το λειτουργικό σύστημα το οποίο «τρέχει» το Asterisk της πτυχιακής μου.

Στοιχεία διακομιστή.

Ο διακομιστής (server) στον οποίο στήθηκε και αναπτύχθηκε το Asterisk, είναι ένα εικονικό μηχάνημα (Virtual Machine – VM) το οποίο στεγάζεται στο Πανεπιστήμιο. Ακούει στην διεύθυνση πρωτοκόλλου διαδικτύου (IP address) 10.100.51.203, η οποία είναι προσβάσιμη από οποιονδήποτε χώρο ανήκει στο δίκτυο του Χαροκοπείου ή μέσω του πρωτοκόλλου εικονικού ιδιωτικού δικτύου (Virtual Private Network – VPN) [2.], που επιτρέπει την σύνδεση εκτός του Πανεπιστημίου.

Επιλογή υπολογιστή σύνδεσης και παραμετροποίησης του διακομιστή.

Για την σύνδεση στον διακομιστή (server) και για την αναπαραγωγή των βημάτων, καλό θα ήταν η χρήση ενός κελύφους Bash (Bash Unix shell), μέσα από κάποια έκδοση Linux. Το κέλυφος Bash είναι προσβάσιμο και από τα λειτουργικά συστήματα Apple's macOS, Microsoft Windows καθώς και από τα συστήματα που είναι βασισμένα στο Unix [3.]. Κατά την διάρκεια της πτυχιακής μου, όμως, χρησιμοποίησα αποκλειστικά Linux (αρχικά Ubuntu και στη συνέχεια

Mint) και για αυτό δεν καλύπτονται πιθανά προβλήματα ή ασυμβατότητες που μπορούν να εμφανιστούν στα άλλα συστήματα, στον ίδιο βαθμό που καλύπτονται για τα Linux.

Τρόπος σύνδεσης στον Διακομιστή.

Προκειμένου να γίνει το στήσιμο του Asterisk στον διακομιστή (server), χρησιμοποίησα το πρωτόκολλο ασφαλούς κελύφους (Secure Shell – SSH) [4.], το οποίο παρέχει πρόσβαση στο τερματικό (terminal) τύπου Bash του εικονικού μηχανήματος, με μοναδική προϋπόθεση αυτό να ανήκει στο ίδιο δίκτυο ή εικονικό δίκτυο (Virtual Private Network – VPN), δηλαδή να είναι εντός εμβέλειας.

Προκειμένου να πιστοποιηθεί η δυνατότητα ύπαρξης σύνδεσης στο εικονικό μηχάνημα (Virtual Machine – VM), μπορεί να χρησιμοποιηθεί η εντολή ping:

```
ping 10.100.51.203
```

Αν στην οθόνη του τερματικού (terminal) εμφανιστούν τουλάχιστον δύο γραμμές, που υποδεικνύουν αποστολές πακέτων, τότε ο διακομιστής (server) είναι εντός εμβέλειας. Εάν σε αντίθετη περίπτωση δεν εμφανίζεται παραπάνω από μια γραμμή, τότε ο διακομιστής είναι εκτός εμβέλειας (Εικόνα 1.1.1).

```
scripter@scripter-SATELLITE-P50-A-118 ~ $ echo "Εκτός Εμβέλειας"
Εκτός Εμβέλειας
scripter@scripter-SATELLITE-P50-A-118 ~ $ ping 10.100.51.203
PING 10.100.51.203 (10.100.51.203) 56(84) bytes of data.
^C
--- 10.100.51.203 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2015ms

scripter@scripter-SATELLITE-P50-A-118 ~ $ echo "Εντός Εμβέλειας"
Εντός Εμβέλειας
scripter@scripter-SATELLITE-P50-A-118 ~ $ ping 10.100.51.203
PING 10.100.51.203 (10.100.51.203) 56(84) bytes of data.
64 bytes from 10.100.51.203: icmp_seq=1 ttl=62 time=30.8 ms
64 bytes from 10.100.51.203: icmp_seq=2 ttl=62 time=30.8 ms
64 bytes from 10.100.51.203: icmp_seq=3 ttl=62 time=121 ms
64 bytes from 10.100.51.203: icmp_seq=4 ttl=62 time=30.5 ms
^C
--- 10.100.51.203 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 30.547/53.311/121.040/39.104 ms
```

Εικόνα 1.1.1

Πληροφορίες σχετικά με την εντολή ping:

Το βοηθητικό πρόγραμμα Ping [5.] είναι ουσιαστικά ένα εργαλείο διαχειριστή συστήματος που χρησιμοποιείται για την εξακρίβωση αν λειτουργεί ή είναι προσβάσιμος ένας διακομιστής (server), καθώς και για την παροχή μετρήσεων της σύνδεσης. Ένα μικρό πακέτο αποστέλλεται μέσω του δικτύου σε μια συγκεκριμένη διεύθυνση IP. Αυτό το πακέτο αποτελείται από 64 bytes. Ο υπολογιστής που έστειλε το πακέτο, στη συνέχεια, περιμένει ένα πακέτο επιστροφής, που εάν ληφθεί ο διακομιστής (server) είναι προσβάσιμος.

Αφού έχει διασφαλιστεί ότι ο διακομιστής (server) είναι εντός εμβέλειας, μπορεί να πραγματοποιηθεί σύνδεση ασφαλούς κελύφους (Secure Shell – SSH) με την παρακάτω εντολή:

```
ssh root@10.100.51.203 -pxxxx
```

Όπου:

- root είναι ο χρήστης με τον οποίο θέλουμε να πραγματοποιήσουμε σύνδεση. Ο root χρήστης είναι ιεραρχικά ο ανώτερος χρήστης στην ιδεολογία του Linux-Unix και έχει όλα τα δικαιώματα χωρίς περαιτέρω εντολές, όπως π.χ. η εντολή sudo.
- 10.100.51.203 είναι η διεύθυνση πρωτοκόλλου διαδικτύου (IP address) του εικονικού μηχανήματος.
- xxxx είναι η πόρτα στην οποία ακούει το πρωτόκολλο του ασφαλούς κελύφους (Secure Shell – SSH) στον διακομιστή (server). Η προκαθορισμένη πόρτα για το πρωτόκολλο του ssh είναι η 22 [6.], μετά όμως από μια αποτυχημένη επίθεση στο εικονικό μηχάνημα η πόρτα άλλαξε (Σελ. 44).

Πληροφορίες σχετικά με την εντολή ssh:

Πρακτικά, κάθε σύστημα Unix και Linux περιλαμβάνει την εντολή ssh. Αυτή η εντολή χρησιμοποιείται για τη σύνδεση ενός πελάτη (client) με έναν διακομιστή (server) σε κάποιο απομακρυσμένο μηχάνημα, μέσω του πρωτοκόλλου ασφαλούς κελύφους (Secure Shell – SSH).

Η εντολή ssh παρέχει τις δυνατότητες πιστοποίησης εισόδου στο απομακρυσμένο μηχάνημα, τη μεταφορά αρχείων μεταξύ των δύο μηχανών και την εκτέλεση εντολών στο απομακρυσμένο μηχάνημα.

Προσοχή:

Α. Όλες οι συνδέσεις που πραγματοποιήθηκαν κατά την διάρκεια της πτυχιακής μου εργασίας, έγιναν μέσω του χρήστη root ο οποίος, όπως προαναφέρθηκε, “είναι ιεραρχικά ο ανώτερος χρήστης στην ιδεολογία του Linux-Unix και έχει όλα τα δικαιώματα χωρίς περαιτέρω εντολές, όπως π.χ. η εντολή sudo”.
Σε περίπτωση προσπάθειας αναπαραγωγής των βημάτων εγκατάστασης από οποιονδήποτε άλλης ιεραρχίας χρήστη, είναι απαραίτητη η χρήση της εντολής **sudo** για την αποφυγή προβλημάτων τύπου άρνησης πρόσβασης (Permission denied).

Β. Για λόγους ασφαλείας, ο αριθμός θύρας (port number) του πρωτοκόλλου ασφαλούς κελύφους (Secure Shell – SSH) έχει τη μορφή xxxx. Σε περίπτωση που ο αριθμός θύρας δεν έχει αλλαχθεί, τότε μπορεί να χρησιμοποιηθεί η εντολή:

```
ssh root@10.100.51.203
```

Πληροφορίες σχετικά με την εντολή sudo

Είναι ένα πρόγραμμα για υπολογιστές που τρέχουν λειτουργικά συστήματα τύπου Linux–Unix. Επιτρέπει στον εκάστοτε χρήστη να εκτελεί προγράμματα με τα προνόμια ασφαλείας ενός ιεραρχικά ανώτατου χρήστη [7.].

Ο χρήστης, για λόγους ασφάλειάς, πρέπει να χρησιμοποιήσει ένα συνθηματικό (password) πριν απολαύσει αυτά τα δικαιώματα.

Πηγές:

1. https://en.wikipedia.org/wiki/Debian
2. https://en.wikipedia.org/wiki/Virtual_private_network
3. https://en.wikipedia.org/wiki/Bash_(Unix_shell)
4. https://en.wikipedia.org/wiki/Secure_Shell
5. https://directory.fsf.org/wiki/Ping
6. https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
7. https://en.wikipedia.org/wiki/Sudo

1.2 Βήματα εγκατάστασης του Asterisk.

Βήμα 1ο. Λήψη του πακέτου εγκατάστασης του Asterisk.

Όπως γίνεται στις περισσότερες περιπτώσεις που ένα πρόγραμμα δεν είναι εγκατεστημένο στον υπολογιστή μας, έτσι και εδώ το πρώτο βήμα είναι να κατεβάσουμε τα απαραίτητα αρχεία.
Για να αποκτήσουμε, λοιπόν, αυτά τα αρχεία, που ουσιαστικά το μεγαλύτερο πλήθος τους είναι ο πηγαίος κώδικας του Asterisk, θα πρέπει να χρησιμοποιήσουμε μια εντολή που λέγεται wget.

Η σύνταξη της εντολής στην περίπτωση μας είναι αρκετά απλή, καθώς το μόνο που χρειαζόμαστε να ξέρουμε είναι τον υπερσύνδεσμο της έκδοσης του Asterisk που θέλουμε να εγκαταστήσουμε.

Για να βρούμε αυτόν τον υπερσύνδεσμο, αρχικά, καλό θα ήταν να επισκεφτούμε την σελίδα http://www.asterisk.org/downloads/asterisk/all-asterisk-versions μέσω ενός φυλλομετρητή (browser), από το μηχάνημα το οποίο θέλουμε να πραγματοποιήσουμε τη σύνδεση στον διακομιστή (server).

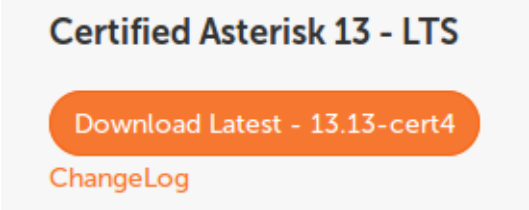
Σε αυτή την ιστοσελίδα, υπάρχει μια λίστα με τις τελευταίες υποστηριζόμενες εκδόσεις. Μια καλή πρόταση θα ήταν να επιλεχθεί η πιο πρόσφατη έκδοση [υποδεικνύεται από τον αριθμό δίπλα στο όνομά της] μακροπρόθεσμης υποστήριξης (Long Term Support – LTS), η οποία να είναι πιστοποιημένη (Certified), δηλαδή οι αναβαθμίσεις να είναι περαιτέρω δοκιμασμένες και ανά αραιά χρονικά διαστήματα (2 - 4 φορές τον χρόνο).

Release Series	Support Type	Release Frequency
Asterisk 14	Standard	6 - 8 Weeks
Certified Asterisk 13	LTS	2 - 4 times per year
Asterisk 13	LTS	6 - 8 Weeks
Certified Asterisk 11	LTS	2 - 4 times per year
Asterisk 11	LTS	6 - 8 Weeks

Εικόνα 1.2.1

Αν ακολουθηθεί η παραπάνω πρόταση, τότε θα αποφευχθεί κατά μεγάλο ποσοστό η πιθανότητα να εντοπιστεί κάποια δυσλειτουργία (bug) κατά την λειτουργία του Asterisk. Με αυτά τα κριτήρια, επιλέχθηκε η έκδοση Certified Asterisk 13.

Στον ίδιο ιστότοπο, κάτω από την λίστα, υπάρχει μια σειρά από πορτοκαλί κουμπιά κατηγοριοποιημένα ανάλογα με την έκδοση την οποία εκπροσωπούν.



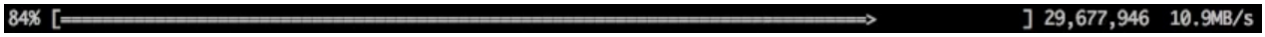
Εικόνα 1.2.2

Κάνοντας δεξί κλικ στο πορτοκαλί κουμπί που αντιστοιχεί στην έκδοση την οποία έχουμε επιλέξει, εμφανίζεται ένα αναδυόμενο μενού (pop-up menu), από το οποίο επιλέγουμε την "Αντιγραφή Τοποθεσίας Συνδέσμου" ή "Copy Link Location", σε περίπτωση που ο φυλλομετρητής (browser) είναι στα αγγλικά.

Στη συνέχεια, έχοντας πραγματοποιήσει σύνδεση με το εικονικό μηχάνημα μέσω ασφαλούς κελύφους (Secure Shell – SSH), πληκτρολογούμε wget, κενό και τον συνδυασμό Ctrl+Shift+v. Το αποτέλεσμα είναι μια γραμμή, η οποία τουλάχιστον στην περίπτωση που έχει επιλεχθεί η έκδοση Certified Asterisk 13 είναι:

```
wget http://downloads.asterisk.org/pub/telephony/certified-asterisk/asterisk-certified-13.13-current.tar.gz
```

Στην πράξη, στον φυλλομετρητή, κάναμε αντιγραφή του υπερσυνδέσμου και τώρα στο τερματικό πατώντας τον συνδυασμό Ctrl+Shift+v κάναμε επικόλληση. Πατώντας το πλήκτρο Enter, βλέπουμε ότι στην τελευταία σειρά που έχει εμφανιστεί υπάρχει μια μπάρα ενημέρωσης ποσοστού ολοκλήρωσης της λήψης του πηγαίου κώδικα.



Εικόνα 1.2.3

Όταν το ποσοστό φτάσει στο 100%, τότε η λήψη έχει ολοκληρωθεί.

Πληροφορίες σχετικά με την εντολή wget:

Το wget είναι ένα εργαλείο με σκοπό την απόκτηση απομακρυσμένων αρχείων [1.], χρησιμοποιώντας τα πιο διαδεδομένα πρωτόκολλα διαδικτύου όπως το πρωτόκολλο μεταφοράς υπερκειμένου (HyperText Transfer Protocol - HTTP), το ασφαλές πρωτόκολλο μεταφοράς υπερκειμένου (*Hypertext Transfer Protocol Secure* – HTTPS) και το πρωτόκολλο μεταφοράς αρχείων (File Transfer Protocol – FTP).

Βήμα 2ο. Προετοιμασία του διακομιστή για την εγκατάσταση.

Στα Linux συστήματα, πολλές φορές πριν την εγκατάσταση ενός προγράμματος είναι απαραίτητο να εγκαταστήσουμε πρώτα ορισμένα πακέτα-βιβλιοθήκες. Στην προκειμένη περίπτωση, χρειάζεται να εγκαταστήσουμε τα παρακάτω:

- build-essential: αυτό το πακέτο περιέχει μια λίστα με πακέτα τα οποία θεωρούνται απαραίτητα για την εγκατάσταση πακέτων Debian.
- libssl-dev: αποτελεί μέρος της υλοποίησης του έργου OpenSSL για τα κρυπτογραφικά πρωτόκολλα ασφαλούς επιπέδου υποδοχών (Secure Sockets Layer - SSL) και ασφάλειας επιπέδου μεταφοράς(Transport Layer Security - TLS) για την ασφαλή επικοινωνία μέσω διαδικτύου.

- libncurses5-dev: είναι μια ανεξαρτήτου τερματικού μέθοδος, που χρησιμεύει στην βέλτιστη ενημέρωση των χαρακτήρων που εμφανίζονται στην οθόνη.
- Libnewt-dev: είναι ένα σύνολο εργαλείων για προγραμματιστές της βιβλιοθήκης newt, που έχει να κάνει με τη δημιουργία παραθύρων .
- Linux-headers-$(uname -r): αυτό το πακέτο περιέχει το σύνολο των αρχείων κεφαλίδας (header files) για μια συγκεκριμένη έκδοση του πυρήνα (Kernel) Linux. Η εντολή uname με όρισμα -r επιστρέφει την έκδοση του πυρήνα (Kernel) Linux του μηχανήματος που καλέστηκε.
- libsqlite3-dev: είναι μια βιβλιοθήκη της C που υλοποιεί μια βάση δεδομένων SQL. Τα προγράμματα που συνδέονται με τη βιβλιοθήκη SQLite μπορούν να έχουν πρόσβαση στη βάση δεδομένων SQL χωρίς να εκτελούν ξεχωριστή διαδικασία.
- Uuid-dev: δημιουργεί και αναλύει καθολικά μοναδικά αναγνωριστικά (Universally Unique IDs - UUIDs) μεγέθους 128-bit.

Η εντολή που θα χρησιμοποιήσουμε για την εγκατάσταση των προαναφερόμενων πακέτων είναι η apt-get.

```
apt-get install build-essential wget libssl-dev libncurses5-dev libnewt-dev linux-headers-$
(uname -r) libsqlite3-dev uuid-dev
```

Κατά την εκτέλεση της εντολής στην οθόνη του τερματικού (terminal), εμφανίζονται τα στάδια λήψης και αποσυμπίεσης των πακέτων (Εικόνα 1.2.4)

```
Need to get 1177 kB of archives.
After this operation, 3583 kB of additional disk space will be used.
Get:1 http://ubuntu.otenet.gr xenial-updates/main i386 libuuid1 i386 2.27.1-6ubuntu3.3 [16,8 kB]
Get:2 http://ubuntu.otenet.gr xenial-updates/main amd64 libuuid1 amd64 2.27.1-6ubuntu3.3 [15,7 kB]
Get:3 http://ubuntu.otenet.gr xenial-updates/main amd64 wget amd64 1.17.1-1ubuntu1.2 [298 kB]
Get:4 http://ubuntu.otenet.gr xenial/main amd64 libtinfo-dev amd64 6.0+20160213-1ubuntu1 [77,4 kB]
Get:5 http://ubuntu.otenet.gr xenial/main amd64 libncurses5-dev amd64 6.0+20160213-1ubuntu1 [175 kB]
Get:6 http://ubuntu.otenet.gr xenial/main amd64 libnewt-dev amd64 0.52.18-1ubuntu2 [60,0 kB]
Get:7 http://ubuntu.otenet.gr xenial/main amd64 libsqlite3-dev amd64 3.11.0-1ubuntu1 [508 kB]
Get:8 http://ubuntu.otenet.gr xenial-updates/main amd64 uuid-dev amd64 2.27.1-6ubuntu3.3 [26,5 kB]
Fetched 1177 kB in 3s (382 kB/s)
(Reading database ... 207029 files and directories currently installed.)
Preparing to unpack .../libuuid1_2.27.1-6ubuntu3.3_amd64.deb ...
De-configuring libuuid1:i386 (2.27.1-6ubuntu3) ...
Unpacking libuuid1:amd64 (2.27.1-6ubuntu3.3) over (2.27.1-6ubuntu3) ...
Preparing to unpack .../libuuid1_2.27.1-6ubuntu3.3_i386.deb ...
Unpacking libuuid1:i386 (2.27.1-6ubuntu3.3) over (2.27.1-6ubuntu3) ...
Preparing to unpack .../wget_1.17.1-1ubuntu1.2_amd64.deb ...
Unpacking wget (1.17.1-1ubuntu1.2) over (1.17.1-1ubuntu1) ...
Selecting previously unselected package libtinfo-dev:amd64.
Preparing to unpack .../libtinfo-dev_6.0+20160213-1ubuntu1_amd64.deb ...
Unpacking libtinfo-dev:amd64 (6.0+20160213-1ubuntu1) ...
Selecting previously unselected package libncurses5-dev:amd64.
Preparing to unpack .../libncurses5-dev_6.0+20160213-1ubuntu1_amd64.deb ...
Unpacking libncurses5-dev:amd64 (6.0+20160213-1ubuntu1) ...
Selecting previously unselected package libnewt-dev:amd64.
Preparing to unpack .../libnewt-dev_0.52.18-1ubuntu2_amd64.deb ...
Unpacking libnewt-dev:amd64 (0.52.18-1ubuntu2) ...
Selecting previously unselected package libsqlite3-dev:amd64.
Preparing to unpack .../libsqlite3-dev_3.11.0-1ubuntu1_amd64.deb ...
Unpacking libsqlite3-dev:amd64 (3.11.0-1ubuntu1) ...
Selecting previously unselected package uuid-dev:amd64.
Preparing to unpack .../uuid-dev_2.27.1-6ubuntu3.3_amd64.deb ...
Unpacking uuid-dev:amd64 (2.27.1-6ubuntu3.3) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
```

Εικόνα 1.2.4

Πληροφορίες σχετικά με την εντολή apt-get:

Είναι ένα ισχυρό μη γραφικό πρόγραμμα διαχείρισης πακέτων [2.], το οποίο χρησιμοποιείται για την εκτέλεση, την εγκατάσταση, την αφαίρεση και την αναβάθμιση των πακέτων του λογισμικού.

Βήμα 3ο. Αποσυμπίεση του πακέτου εγκατάστασης του Asterisk.

Το αποτέλεσμα της wget είναι ένας συμπιεσμένος φάκελος, ο οποίος βρίσκεται στο ευρετήριο (directory) που βρισκόμασταν όταν εκτελέσαμε την wget.

Η εντολή που θα μας βοηθήσει να αποσυμπιέσουμε τον φάκελο είναι η tar, συνοδευόμενη από τα ορίσματα zxvf και το όνομα του συμπιεσμένου αρχείου (στην περίπτωση της έκδοσης Asterisk 13 Certified, είναι asterisk-certified-13.8-current.tar.gz).

```
tar zxvf asterisk-certified-13.8-current.tar.gz.
```

Τα ορίσματα σημαίνουν:

- z --> αποσυμπίεση,
- x --> εξαγωγή αρχείων,
- v --> εμφάνιση των ονομάτων των αρχείων,
- f --> ακολουθεί το όνομα του συμπιεσμένου αρχείου.

Πληροφορίες σχετικά με την εντολή tar:

Είναι ένα βοηθητικό πρόγραμμα λογισμικού για τη συλλογή πολλών αρχείων σε ένα αρχείο συμπίεσης. Το όνομα προέρχεται από **t**ape **ar**chive, δηλαδή αρχείο σε ταινία [3.]. Ο λόγος που έχει αυτήν την ονομασία υποδεικνύει πως είχε αρχικά αναπτυχθεί για να γράφει δεδομένα σε μαγνητικές ταινίες.

Κατά την εκτέλεση της εντολής στην οθόνη του τερματικού (terminal,) εμφανίζονται με την σειρά όλες η τοποθεσίες των αρχείων που αποσυμπιέζονται.

```
ls
```

```
root@asterisk-thesis:~# ls
Desktop    Pictures   asterisk-certified-13.8-cert3
Documents  Public     asterisk-certified-13.8-current.tar.gz.1
Downloads  Templates  downloads
Music      Videos     trunk
```

Εικόνα 1.2.5

Βοήθημα:

Αν δεν θυμάστε ποια έκδοση κατεβάσατε, τότε η εντολή ls θα εμφανίσει όλα τα αρχεία που βρίσκονται κάτω από το συγκεκριμένο ευρετήριο (directory). Αν δεν έχετε μεταβεί σε κάποιο άλλο ευρετήριο (directory), τότε το αρχείο συμπίεσης θα είναι εμφανές (έχει κατάληξη .tar.gz).

Πληροφορίες σχετικά με την εντολή ls:

Είναι μια εντολή Unix – Linux, με σκοπό την εμφάνιση των αρχείων και των καταλόγων που υπάγονται στον συγκεκριμένο κατάλογο (directory) που βρισκόταν ο χρήστης όταν την κάλεσε ή τον κατάλογο (directory) που έθεσε ο χρήστης ως όρισμα δίπλα από την εντολή . Το 'ls' αποτελεί συντομογραφία της αγγλικής λέξης "**list**" που σημαίνει λίστα [4.].

Βήμα 4ο. Εγκατάσταση του πακέτου λογισμικού του Asterisk.

Το επόμενο βήμα είναι, να μετατρέψουμε (compile) τον πηγαίο κώδικα που αποσυμπιέστηκε προηγουμένως, από την γλώσσα προγραμματισμού C σε γλώσσα κατανοητή από τον υπολογιστή. Η διαδικασία αυτή είναι πολύ συνηθισμένη στα Linux, όταν θέλουμε να εγκαταστήσουμε κάποιο ανοικτό λογισμικό από τον πηγαίο κώδικα (source code).

Αρχικά, πρέπει να “μεταφερθούμε” από τον κατάλογο στον οποίο βρισκόμαστε στον κατάλογο που αποσυμπιέσαμε. Η εντολή που μας επιτρέπει αυτήν την “μετακίνηση” είναι η εντολή cd.

```
cd ./asterisk-certified-13.8-cert3
```

Προσοχή:

Α. Το ./ σημαίνει ότι ήδη “βρισκόμαστε” στον πατρικό κατάλογο (directory) του αποσυμπιεσμένου ευρετηρίου (directory). Αν ο αποσυμπιεσμένος κατάλογος βρίσκεται σε άλλο ευρετήριο τότε αντί για ./ πρέπει να γραφτεί ολόκληρη η διαδρομή (path).

π.χ. αν η αποσυμπίεση έγινε στον κατάλογο Downloads, που έχει σαν πατρικό κατάλογο τον root, τότε η εντολή θα είναι :

```
cd root/Downloads/asterisk-certified-13.8-cert3
```

Β. το όνομα του αποσυμπιεσμένου φακέλου είναι το ίδιο με το όνομα του συμπιεσμένου αρχείου που προηγουμένως έγινε tar, χωρίς την κατάληξη .tar.gz . Για πιο εύκολη σύνταξη της εντολής, μπορεί να εκτελεστεί πρώτα η εντολή ls, να μαρκαριστεί με τον κέρσορα το όνομα του αποσυμπιεσμένου αρχείου, να πατηθεί ο συνδυασμός ctrl+shift+c [στο τερματικό (terminal) σημαίνει αντιγραφή], να πληκτρολογηθεί η εντολή cd, μετά κενό και ο συνδυασμός ctrl+shift+v [στο τερματικό (terminal) σημαίνει επικόλληση].

Πληροφορίες σχετικά με την εντολή cd:

Είναι μια εντολή που χρησιμοποιείται για την αλλαγή του τρέχοντος καταλόγου (directory) εργασίας στα λειτουργικά συστήματα Linux-Unix. Το όνομά της προέρχεται από τα αρχικά των λέξεων **c**hange **d**irectory, που στα ελληνικά σημαίνει αλλαγή καταλόγου [5.].

Βοήθημα:

Η εντολή cd, πολλές φορές δεν επιστρέφει στην οθόνη, κάτι το οποίο να πιστοποιεί ότι άλλαξε ο τρέχον κατάλογος (directory). Για να μην δημιουργούνται αμφιβολίες, μπορεί να χρησιμοποιηθεί η εντολή pwd.

```
pwd
```

Η εντολή αυτή, τυπώνει στην οθόνη τον τρέχοντα κατάλογο (directory) (Εικόνα 1.2.6).

```
root@asterisk-thesis:~/asterisk-certified-13.8-cert3# pwd
/root/asterisk-certified-13.8-cert3
```

Εικόνα 1.2.6

Πληροφορίες σχετικά με την εντολή pwd:

Στα λειτουργικά συστήματα Linux-Unix, καθώς και σε κάποια άλλα συστήματα, η εντολή pwd ("**p**rint **w**orking **d**irectory", "τύπωσε τον τρέχοντα κατάλογο") χρησιμοποιείται για να εμφανίζει τη διαδρομή του τρέχοντος καταλόγου εργασίας (directory) [6.].

Προκειμένου να ελεγχθεί αν ο διακομιστής (server) τηρεί όλες τις προϋποθέσεις για την εγκατάσταση του Asterisk, μέσα στον αποσυμπιεσμένο κατάλογο (directory), υπάρχει μια τύπου bash δέσμη εντολών (bash shell script) που «τρέχει» διάφορους ελέγχους. Για να «τρέξουμε» αυτή την δέσμη, απλά χρειάζεται να πληκτρολογήσουμε:

```
./configure
```

Όταν τελειώσει η διαδικασία, αν όλα πήγαν καλά, θα τυπωθεί στην οθόνη το λογότυπο του Asterisk, φτιαγμένο με χαρακτήρες του ASCII (Εικόνα 1.2.7).



Εικόνα 1.2.7

Προσοχή:

Σε περίπτωση που λήφθηκε κάποια άλλη έκδοση του Asterisk, υπάρχει ενδεχόμενο ο έλεγχος να αναδείξει κάποια έλλειψη πακέτου. Σε αυτή την περίπτωση, το πρόβλημα συνήθως επιλύεται εύκολα με την εντολή apt-get install, συνοδευόμενη από το όνομα του πακέτου που λείπει. Αφού εγκατασταθεί, μπορεί να επαναληφθεί η εντολή ./configure.

Πληροφορίες σχετικά με το bash shell script:

Είναι ένα πρόγραμμα υπολογιστή σχεδιασμένο για να «τρέχει» πάνω από το κέλυφος bash των Linux-Unix. Οι τυπικές λειτουργίες που εκτελούνται από τις bash δέσμες εντολών (bash shell script), περιλαμβάνουν τον χειρισμό αρχείων, την εκτέλεση προγραμμάτων και την εκτύπωση κειμένων. Το σύνολο των εντολών του είναι το ίδιο με το σύνολο των εντολών του bash και τα αρχεία του συνήθως -αλλά όχι απαραίτητα- έχουν την κατάληξη .sh [7.].

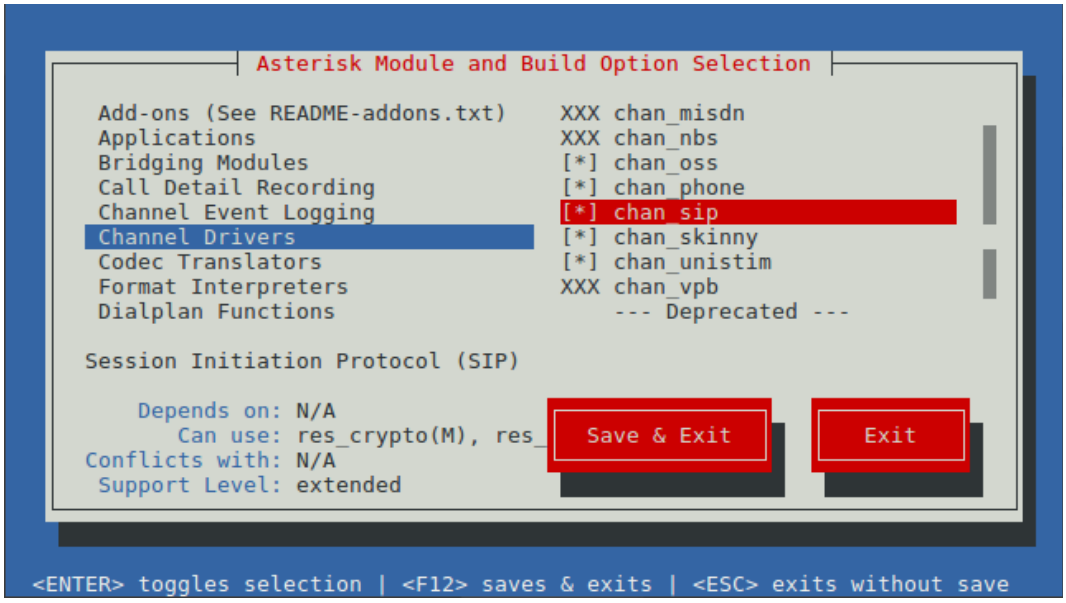
Το επόμενο στάδιο είναι, να μετατραπεί ο πηγαίος κώδικας σε γλώσσα κατανοητή από τον διακομιστή [η διαδικασία αυτή λέγεται compile] και να γίνει η εγκατάσταση του Asterisk. Η εντολή που μας δίνει αυτή την δυνατότητα είναι η make.

Αυτή την εντολή, θα την χρησιμοποιήσουμε σε τέσσερα στάδια:

1. Για να βεβαιωθούμε ότι ο Asterisk θα υποστηρίζει το πρωτόκολλο του SIP (Session Initiation Protocol), θεμιτό θα ήταν να «τρέξουμε» την εντολή:

```
make menuconfig
```

Η οποία θα εμφανίσει στην οθόνη του τερματικού (terminal) ένα μενού. Από αυτό το μενού πρέπει, με το πλήκτρο “κάτω βέλος”, πρέπει να οδηγηθούμε στην καρτέλα (tab) “Channel Drivers” και από εκεί, με το “δεξί βέλος” να μπούμε στις υποεπιλογές. Στη συνέχεια, με την βοήθεια του πλήκτρου “κάτω βέλος” βρίσκουμε την επιλογή “chan sip”, αριστερά της οποίας θέλουμε να υπάρχει -ανάμεσα σε αγκύλες- ένας αστερίσκος [αν οχί, το πλήκτρο enter θα τον εμφανίσει] (Εικόνα 1.2.8). Με ένα ακόμα πάτημα του “δεξιού βέλους” μπορούμε να επιλέξουμε “Save & Exit”, σε περίπτωσή που αλλάξαμε κάποια επιλογή, ή “Exit”, στην αντίθετη



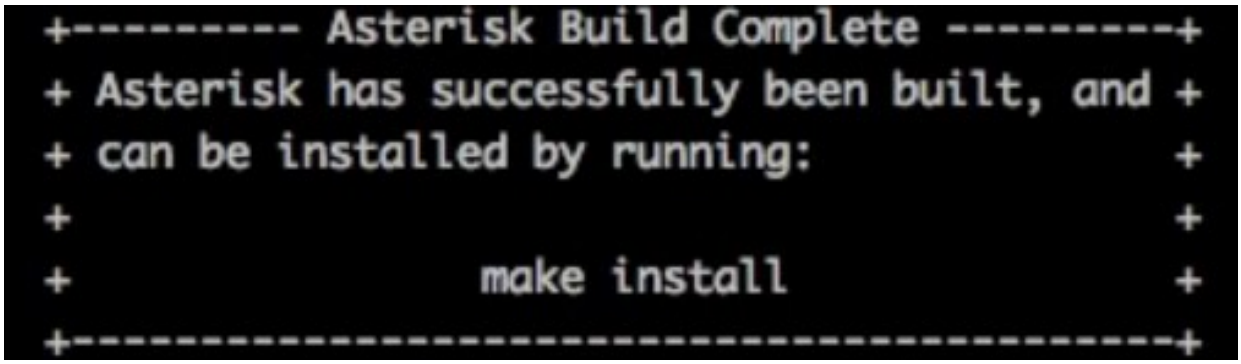
περίπτωση.

Εικόνα 1.2.8

2. Προκειμένου να γίνει η μετατροπή του κώδικα σε γλώσσα υπολογιστή (compile), «τρέχουμε» την εντολή:

```
make
```

Στην οθόνη του τερματικού εμφανίζονται αρκετές γραμμές το δευτερόλεπτο. Η διαδικασία κρατάει αρκετό χρόνο (ο οποίος εξαρτάται από τις δυνατότητες του μηχανήματος). Στο τέλος,



γίνεται ορατή η Εικόνα 1.2.9.

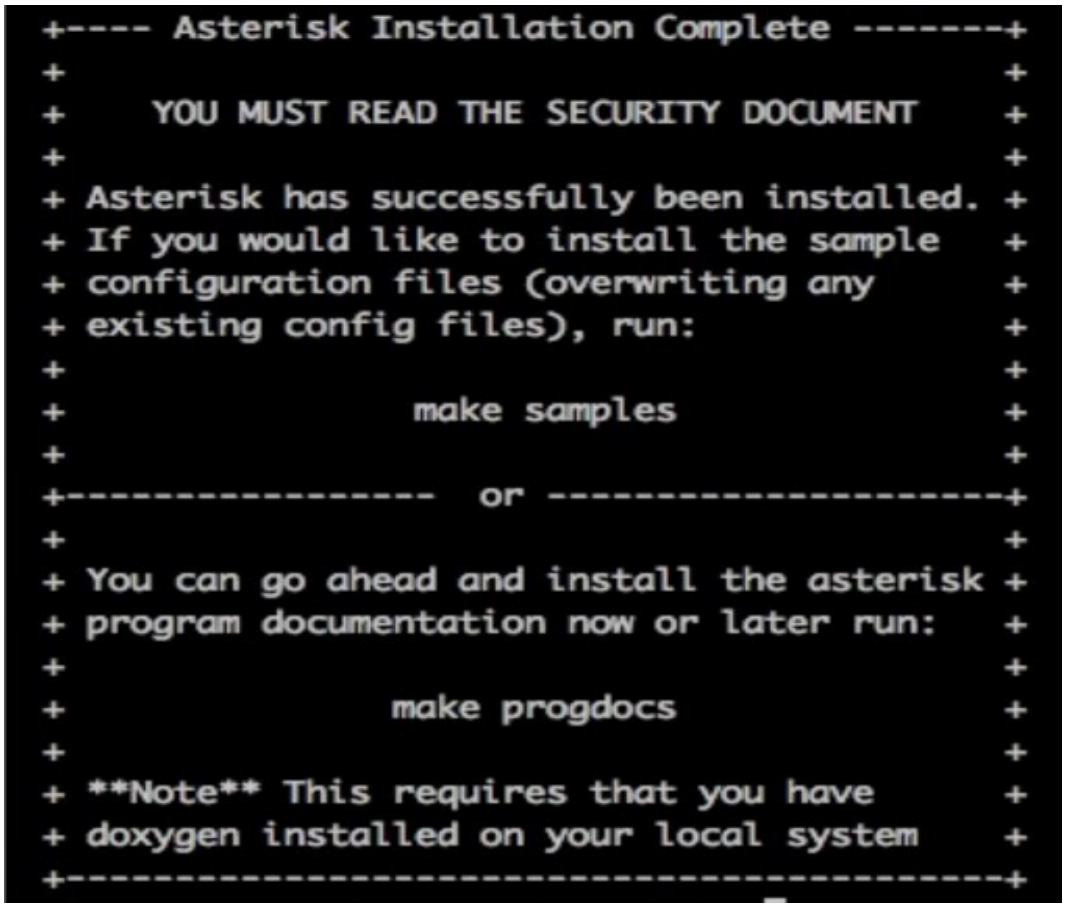
Εικόνα 1.2.9.

3. Σε αυτό το στάδιο, γίνεται η εγκατάσταση του Asterisk, με την χρήση της εντολής:

```
make install
```

Κατά την διαδικασία αυτή, η εντολή make απλά αντιγράφει τα αρχεία που δημιουργήθηκαν στο προηγούμενο στάδιο, στις σωστές τους θέσεις.

Χρονικά, είναι πολύ σύντομη η διάρκειά της, και κατά την λήξη της εμφανίζεται στην οθόνη η Εικόνα 1.2.10.



Εικόνα 1.2.10

4. Το τελευταίο στάδιο παράγει κάποιες προεπιλεγμένες ρυθμίσεις, για την βασική λειτουργία του Asterisk.

```
make samples
```

Οι ρυθμίσεις του Asterisk είναι σε μορφή αρχείων παραμετροποίησης (configuration files), τα οποία πρέπει πάντα να έχουν κατάληξη .conf, και βρίσκονται στον κατάλογο /etc/asterisk. Για τα αρχεία παραμετροποίησης, θα μιλήσουμε εκτενέστερα στην Ενότητα 3.2 (Σελ. 64).

Πληροφορίες σχετικά με την εντολή make:

Είναι ένα εργαλείο, που δημιουργεί αυτόματα εκτελέσιμα προγράμματα και βιβλιοθήκες από πηγαίο κώδικα, διαβάζοντας συγκεκριμένα αρχεία που ονομάζονται Makefiles. Η εντολή make χρησιμοποιείται ευρέως, ειδικά σε λειτουργικά συστήματα Linux-Unix, όχι μόνο για την εγκατάσταση, αλλά και για την αναβάθμιση προγραμμάτων από πηγαίο κώδικα [8.].

Για να βεβαιωθούμε ότι το Asterisk εγκαταστάθηκε με επιτυχία, μπορούμε να «τρέξουμε» την εντολή:

```
asterisk -x "core show version"
```

η οποία θα επιστρέψει την έκδοση του Asterisk, καθώς και από ποιον χρήστη, σε ποιο μηχάνημα και πότε έγινε η εγκατάσταση (Εικόνα 1.2.11).

```
Asterisk certified/13.8-cert3 built by root @ asterisk-thesis on a x86_64 running Linux on 2016-11-26 12:59:55 UTC
```

Εικόνα 1.2.11

Πηγές:

1. https://www.gnu.org/software/wget/
2. https://en.wikipedia.org/wiki/APT_(Debian)
3. https://en.wikipedia.org/wiki/Tar_(computing)
4. https://en.wikipedia.org/wiki/Ls
5. https://en.wikipedia.org/wiki/Cd_(command)
6. https://el.wikipedia.org/wiki/Pwd
7. https://en.wikipedia.org/wiki/Shell_script
8. https://en.wikipedia.org/wiki/Make_(software)
9. https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
10. https://en.wikipedia.org/wiki/HTTPS
11. https://el.wikipedia.org/wiki/File_Transfer_Protocol
12. https://el.wikipedia.org/wiki/SSL
13. https://el.wikipedia.org/wiki/TLS

1.3 Ταυτόχρονη εκκίνηση του Asterisk και του διακομιστή (server).

Τα περισσότερα προγράμματα τα οποία αναπτύχθηκαν με την λογική του εξυπηρέτη (server side software), θέλουμε να είναι ενεργά όσο είναι ενεργός και ο διακομιστής που τα «τρέχει». Για αυτόν τον σκοπό, το Asterisk παρέχει ορισμένα αρχεία δέσμης εντολών τύπου bash (bash shell script), τα οποία φροντίζουν να εκκινούν των Asterisk όσο το δυνατότερο πιο άμεσα μετά την εκκίνηση του διακομιστή (server). Λόγω αυτής της πολύ συγκεκριμένης τους ιδιότητας, αυτά τα αρχεία λέγονται δέσμες εντολών εκκίνησης ή αλλιώς αρχεία εκκίνησης (start script). Τα αρχεία εκκίνησης που παρέχει ο Asterisk βρίσκονται στον υποκατάλογο (subdirectory) .contrib/init.d του αποσυμπιεσμένου καταλόγου (directory), που εύκολα κανείς τον επισκέπτεται με την εντολή :

```
cd ./contrib/init.d
```

Δεδομένου ότι βρισκόμαστε ακόμα στον ίδιο κατάλογο (directory), εκτελέστηκαν οι εντολές make.

```
root@asterisk-thesis:~/asterisk-certified-13.8-cert3# cd ./contrib/init.d
root@asterisk-thesis:~/asterisk-certified-13.8-cert3/contrib/init.d# pwd
/root/asterisk-certified-13.8-cert3/contrib/init.d
```

Εικόνα 1.2.12

Σε αυτό το ευρετήριο (directory), υπάγονται διάφορες κατηγορίες αρχείων δεσμίδων εντολών. Τα αρχεία εκκίνησης είναι αυτά που έχουν τη δομή ονόματος rc.ΟνομάΛογισμικού.asterisk. Για να δούμε μια λίστα μόνο με τα αρχεία εκκίνησης, μπορούμε να πληκτρολογήσουμε:

```
ls -l rc.*.asterisk
```

Όπου :

- το όρισμα l, εμφανίζει την έξοδο (output) της ls σε μορφή λίστας [ένα αρχείο ή ευρετηρίο (directory) ανά γραμμή].
- το rc.*.asterisk, υποδεικνύει στην ls να επιστρέψει μόνο ότι ξεκινάει με “rc.”, ακολουθείται από οτιδήποτε, αρκεί να τελειώνει με “.” και καταλήγει σε “asterisk”.

Στην οθόνη τώρα είναι εμφανής μία λίστα από αρχεία εκκίνησης για διάφορα λειτουργικά συστήματα (Εικόνα 1.2.13).

```
root@asterisk-thesis:~/asterisk-certified-13.8-cert3/contrib/init.d# ls -l rc.*.asterisk
-rwxrwxr-x 1 1023 users 1669 Sep  8  2016 rc.archlinux.asterisk
-rwxrwxr-x 1 1023 users 4363 Sep  8  2016 rc.debian.asterisk
-rwxrwxr-x 1 1023 users 1048 Sep  8  2016 rc.gentoo.asterisk
-rwxrwxr-x 1 1023 users 4860 Sep  8  2016 rc.mandriva.asterisk
-rwxrwxr-x 1 1023 users 4155 Sep  8  2016 rc.redhat.asterisk
-rwxrwxr-x 1 1023 users 1508 Sep  8  2016 rc.slackware.asterisk
-rwxrwxr-x 1 1023 users 4206 Sep  8  2016 rc.suse.asterisk
```

Εικόνα 1.2.13

Η εντολή η οποία μπορεί να μας υπενθυμίσει πια Linux «τρέχουμε», είναι:

```
lsb_release -i
```

```
root@asterisk-thesis:~/asterisk-certified-13.8-cert3/contrib/init.d# lsb_release -i
Distributor ID: Debian
```

Εικόνα 1.2.14

Πληροφορίες σχετικά με την Linux Standard Base (LSB):

Η Linux Standard Base (LSB) είναι ένα κοινό έργο από διάφορες διανομές Linux υπό την οργανωτική δομή του Linux Foundation, για την τυποποίηση της δομής του συστήματος λογισμικού (operation system) [1.].

Το LSB καθορίζει: βιβλιοθήκες, μια σειρά εντολών, την ιεραρχία του συστήματος αρχείων (file system), τα επίπεδα εκτέλεσης, το σύστημα εκτύπωσης κ.α.

‘Ολες οι εκδόσεις του Linux που υποστηρίζουν την LSB έχουν πρόσβαση σε εντολές όπως η lsb_release -i, η οποία εμφανίζει στην οθόνη το όνομα της διανομής.

Με βάση την πληροφορία που λάβαμε από την εντολή lsb_release –i, επιλέγουμε το κατάλληλο αρχείο εκκίνησης και το αντιγράφουμε στην τοποθεσία /etc/init.d/ αλλάζοντας το όνομα σε asterisk.

```
cp rc.debian.asterisk /etc/init.d/asterisk
```

Το αρχείο εκκίνησης έχει μεταφερθεί στον κατάλογο (directory) /etc/init.d/asterisk και χρειάζεται μία μικρή επεξεργασία προκειμένου να είναι λειτουργικό. Προκειμένου να επεξεργαστούμε το αρχείο, θα χρησιμοποιήσουμε έναν μη γραφικό (no GUI) επεξεργαστή κειμένου (text editor) που ονομάζεται pico.

Πληροφορίες σχετικά με την εντολή cp:

Είναι μια εντολή Linux-UNIX για την αντιγραφή αρχείων και καταλόγων (directory). Το πρώτο όρισμα είναι η πηγή [δηλαδή ποιο αρχείο ή κατάλογος (directory) θέλουμε να αντιγράψουμε] και το δεύτερο όρισμα είναι η διαδρομή (path) που θέλουμε να αντιγράφει, με κατάληξη το όνομα που θέλουμε να έχει στην τοποθεσία αυτή [2.].

Πολλές φορές χρησιμοποιείται και για την μετονομασία ενός αρχείου (file) ή ενός καταλόγου (directory), απλά χρησιμοποιώντας ως διαδρομή στο δεύτερο όρισμα την διαδρομή του καταλόγου που βρισκόμαστε και με κατάληξη το νέο όνομα .

Πριν την εκτέλεση του pico χρειαζόμαστε την πληροφορία της τοποθεσίας της εγκατάστασης του Asterisk, που μας την παρέχει η εντολή:

```
type asterisk
```

Πληροφορίες σχετικά με την εντολή type:

Στην φιλοσοφία Linux-UNIX, υπάρχουν εντολές οι οποίες είναι προ-εγκαταστημένες στο κέλυφος (shell) (όπως η cd και η pwd).

Εκτός αυτών, όμως, καθ' όλη την χρήση του συστήματος, προστίθενται και άλλες (π.χ. η εντολή asterisk).

Η εντολή "type" επιστρέφει την διαδρομή (path) στην οποία βρίσκεται το εκτελέσιμο (executable) της εντολής, αν αυτή δεν είναι προ-εγκαταστημένη στο κέλυφος ή το μήνυμα "x is a shell builtin" (η εντολή x είναι προ-εγκαταστημένη στο κέλυφος) αν η εντολή προϋπήρχε στο κέλυφος (shell) [3.].

Το κείμενο το οποίο τυπώνεται στην οθόνη του τερματικού (terminal), καλό θα ήταν να το μαρκάρουμε με τον κέρσορα και πατώντας τον συνδυασμό πλήκτρον ctrl+shift+c να το αντιγράψουμε.

Στη συνέχεια, με την εντολή:

```
pico /etc/init.d/asterisk
```

εμφανίζεται το περιβάλλον επεξεργασίας. Με την βοήθεια των πλήκτρων με τα βέλη κινούμαστε μέσα στο αρχείο και αλλάζουμε τις τιμές :

DAEMON=__ASTERISK_SBIN_DIR__/asterisk
ASTVARRUNDIR=__ASTERISK_VARRUN_DIR__
ASTETCDIR=__ASTERISK_ETC_DIR__

Σε,

DAEMON=/usr/sbin/asterisk
ASTVARRUNDIR=/var/run/asterisk
ASTETCDIR=/etc/asterisk

Για να αποθηκεύσουμε το αρχείο, θα πρέπει να πληκτρολογήσουμε τον συνδυασμό ctrl+o, enter και για να βγούμε από το περιβάλλον επεξεργασίας πατάμε τον συνδυασμό ctrl+x.

Πληροφορίες σχετικά με το pico:

Στα συστήματα Linux-Unix το pico είναι ένας επεξεργαστής κειμένου. Αποτελεί μέρος του προγράμματος διαχείρισης ηλεκτρονικού ταχυδρομείου (e-mail client) pine, το οποίο σχεδιάστηκε από το γραφείο πληροφορικής και επικοινωνιών του Πανεπιστημίου της Ουάσιγκτον (the Office of Computing and Communications at the University of Washington)._

Επειδή το pine ήταν και είναι πολύ εύχρηστο κατά την σύνταξη (compose) μηνυμάτων, το κομμάτι του κώδικα το οποίο έχει να κάνει με αυτή την λειτουργία ανεξαρτητοποιήθηκε και κυκλοφόρησε ως ανεξάρτητο πρόγραμμα σύνταξης αρχείων με το όνομα pico (pine composer).

Ένας κλώνος του pico που ονομάζεται nano, αναπτύχθηκε από την κοινότητα ανοιχτού λογισμικού, επειδή το pico είχε ασαφείς όρους [4.].

Προσοχή:

Το πεδίο DAEMON πρέπει να συμπληρωθεί σβήνοντας το "__ASTERISK_SBIN_DIR__/asterisk" και πατώντας τον συνδυασμό ctrl+shift+v ώστε να πραγματοποιηθεί επικόλληση της τοποθεσίας εγκατάστασης του Asterisk (είχε γίνει αντιγραφή στο βήμα της εντολής type asterisk).

Πληκτρολογώντας:

```
/etc/init.d/asterisk start
```

Εμφανίζεται στην οθόνη το μήνυμα της Εικόνα 1.2.15.

```
root@asterisk-thesis:~# /etc/init.d/asterisk start
[ ok ] Starting asterisk (via systemctl): asterisk.service.
```

Εικόνα 1.2.15

Βοήθημα:

Το αρχείο εκκίνησης παρέχει και την δυνατότητα να διακόψει το Asterisk με την χρήση του ορίσματος “stop” έναντι του “start”.

Τέλος, προκειμένου το αρχείο εκκίνησης (start script) να εκτελείται με την έναρξη του διακομιστή (server), κάνουμε χρήση της εντολής:

```
update-rc.d asterisk defaults
```

Πληροφορίες σχετικά με την εντολή update-rc.d:

Στα συστήματα Linux-Unix, οι διαδικασίες οι οποίες εκκινούνται εν αγνοία του χρήστη όταν κάποιο γεγονός (event) λάβει χώρα λέγονται “Daemons”.

Η εντολή update-rc.d δημιουργεί “Daemons” που “ξυπνάνε” όταν γίνεται η αρχικοποίηση του λογισμικού (init) [5.].

Πηγές:

1. https://en.wikipedia.org/wiki/Linux_Standard_Base
2. https://en.wikipedia.org/wiki/Cp_(Unix)
3. http://linuxcommand.org/lc3_lts0060.php
4. https://en.wikipedia.org/wiki/Pico_(text_editor)
5. https://en.wikipedia.org/wiki/Init
6. https://wiki.debian.org/Daemon

ΚΕΦΑΛΑΙΟ 2

Η επίθεση και η θωράκιση του Asterisk

2.1 Η επίθεση

Κατά την διάρκεια της πτυχιακής μου, παρατήρησα ότι ενώ ο διακομιστής (server) δεν θα έπρεπε να είναι ορατός εκτός του δικτύου της σχολής, δέχτηκε μια διαδικτυακή επίθεση. Οι επιτιθέμενοι έκαναν απόπειρες να διεισδύσουν στο σύστημα, χρησιμοποιώντας σαν όνομα χρήστη (username) τα πιο διαδεδομένα (π.χ. “Admin”, “admin”,”user”κ.α.) και τυχαία συνθηματικά (password). Αυτή η απόπειρα όμως, μετατράπηκε σε επίθεση άρνησης εξυπηρέτησης (Denial of Service – Dos), καθώς τα ονόματα χρήστη που χρησιμοποιούνταν από τους επιτιθέμενους δεν αντιστοιχούσαν σε κανέναν χρήστη (Εικόνα 2.1.1).

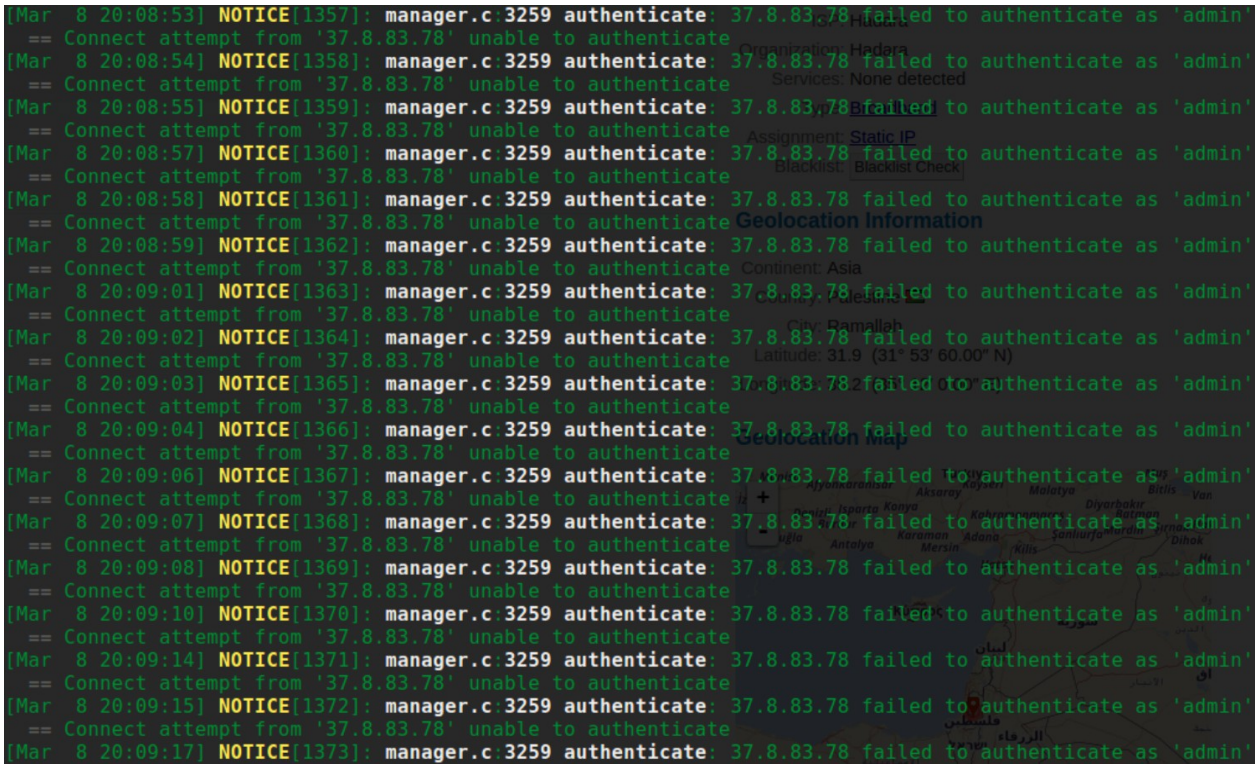
Πληροφορίες σχετικά με την επίθεση άρνησης εξυπηρέτησης (DoS):

Στην επιστήμη της πληροφορικής, μια επίθεση άρνησης εξυπηρέτησης (DoS), όπου ο δράστης βρίσκεται στον κυβερνοχώρο και επιδιώκει να κάνει ένα μηχάνημα ή έναν πόρο δικτύου μη διαθέσιμο στους χρήστες για τους οποίους προορίζεται, διακόπτοντας προσωρινά ή επ’ αόριστον τη λειτουργία τους. Η άρνηση εξυπηρέτησης επιτυγχάνεται, κατά κανόνα, με τον κατακλυσμό του στοχευόμενου μηχανήματος ή πόρου με περιττά αιτήματα, σε μια προσπάθεια να υπερφορτωθεί και έτσι να μην είναι δυνατή η εκπλήρωση ορισμένων ή όλων των νόμιμων αιτημάτων.

Οι επιτιθέμενοι συχνά στοχεύουν σε τοποθεσίες ή υπηρεσίες που φιλοξενούνται σε διακομιστές ιστού υψηλού προφίλ, όπως τράπεζες ή πύλες πληρωμής με πιστωτικές κάρτες. Η εκδίκηση, ο εκβιασμός και ο ακτιβισμός μπορούν να παρακινήσουν αυτές τις επιθέσεις.

Η επίθεση έλαβε τέλος μετά από λίγο καιρό, αφού είχα ενημερώσει τον επιβλέποντα καθηγητή μου κ. Καμαλάκη και τον υπεύθυνο του κέντρου πληροφορικής και δικτύου του τμήματός μας που αποτελεί και μέλος της εξεταστικής επιτροπής της πτυχιακής μου εργασίας κ. Δαλάκα.

Προκειμένου να μεγιστοποιηθεί η θωράκιση του διακομιστή (server) πάνω στον οποίο «τρέχει» το Asterisk, έτσι ώστε να είναι ακόμα πιο ανθεκτικός σε περίπτωση μελλοντικής επίθεσης, ακολουθήθηκαν ορισμένες ενέργειες που αποτελούν ενότητες του συγκεκριμένου κεφαλαίου.



Εικόνα 2.1.1

2.2. Ανάθεση του Asterisk σε χρήστη χαμηλής ιεραρχίας

Επειδή το Asterisk είναι μια εφαρμογή δικτύου η οποία μπορεί να αλληλεπιδράσει με την γραμμή εντολών, σοφό θα ήταν να ανατεθεί η λειτουργία της εφαρμογής σε έναν χρήστη χαμηλότερης ιεραρχίας από τον root.

Για να λειτουργεί το Asterisk σαν ένας νέος χρήστης, πρώτα πρέπει να διακοπεί η λειτουργία του. Η διακοπή της λειτουργίας γίνεται μέσα από το αρχείο εκκίνησης (start script), με το οποίο ασχοληθήκαμε στην Ενότητα 1.3 (Σελ. 32).

```
/etc/init.d/asterisk stop
```

Το μήνυμα “[ok] Stopping asterisk (via systemctl): asterisk.service.”, αποτελεί επιβεβαίωση ότι η υπηρεσία του Asterisk τερματίστηκε (Εικόνα 2.2.1).

```
root@asterisk-thesis:~# /etc/init.d/asterisk stop
[ ok ] Stopping asterisk (via systemctl): asterisk.service.
```

Εικόνα 2.2.1

Στη συνέχεια, δημιουργούμε έναν νέο χρήστη με όνομα asterisk και κατάλογο αφετηρίας (home directory) την διαδρομή (path) /var/lib/asterisk :

```
useradd –d /var/lib/asterisk asterisk
```

Πληροφορίες σχετικά με την εντολή useradd:

Αποτελεί ένα από τα πιο σημαντικά εργαλεία του Linux-Unix. Η εντολή παρέχει την λειτουργία της δημιουργίας νέων χρηστών, χωρίς να αναγκάζει τον χρήστη να επεξεργαστεί όλα τα απαραίτητα αρχεία χειροκίνητα [1.].

Στις τελευταίες εκδόσεις του Linux, έχει προστεθεί η εντολή adduser που σκοπεύει μελλοντικά να αντικαταστήσει την useradd, καθώς παρέχει τις ίδιες λειτουργίες σε ένα πιο φιλικό για τον χρήστη περιβάλλον.

Η useradd δεν επιστρέφει στην οθόνη κανένα μήνυμα. Για να μην υπάρχουν αμφιβολίες σχετικά με το αν προστέθηκε ο καινούριος χρήστης, μέσω της εντολής:

```
cat /etc/passwd | grep "asterisk"
```

εμφανίζεται στο τερματικό η εγγραφή του χρήστη από το αρχείο χρηστών (/etc/passwd) (Εικόνα 2.2.2).

```
root@asterisk-thesis:~# cat /etc/passwd | grep "asterisk"
asterisk:x:1001:1001::/var/lib/asterisk:/bin/sh
```

Εικόνα 2.2.2

Πληροφορίες σχετικά με την διασωλήνωση (pipeline), την εντολή cat και την εντολή grep:

Διασωλήνωση (pipeline): Στα συστήματα Linux-Unix, όταν θέλουμε να ανακατευθύνουμε το αποτέλεσμα (output) μιας εντολής για να το χρησιμοποιήσουμε ως είσοδο (input) σε μια άλλη, τότε χρησιμοποιούμε την έννοια της διασωλήνωσης (pipeline) [2.].

Για να γίνει αυτή η ανακατεύθυνση, γράφουμε αρχικά την εντολή που θέλουμε να επεξεργαστούμε την έξοδο της, πληκτρολογούμε τον χαρακτήρα "|" και τέλος, την εντολή που θέλουμε να επεξεργαστεί την προηγούμενη έξοδο (output).

cat: αποτελεί μια από της πιο παλιές εντολές του Linux-Unix. Προέρχεται από την λέξη con**cat**enate (που σημαίνει συνένωση). Ο ρόλος της εντολής αυτής είναι να διαβάζει τα αρχεία που της δόθηκαν ως ορίσματα (αν είναι παραπάνω από ένα, να τα ενώνει) και να επιστρέφει ως έξοδο αυτό που διάβασε [3.].

grep: συνήθως χρησιμοποιείται στα συστήματα Linux-Unix μέσω της διασωλήνωσης (pipeline). Ψάχνει στην είσοδό της, αν υπάρχει η συμβολοσειρά (string) η οποία της δόθηκε ως όρισμα και τυπώνει στην οθόνη ολόκληρες τις γραμμές όπου εντοπίστηκε η είσοδος [4.].

Ο νέος χρήστης που δημιουργήθηκε πρέπει να έχει πρόσβαση σε κάποιους καταλόγους (directory) και στους υποκαταλόγους (sub-directory) τους. Αυτοί οι κατάλογοι είναι οι /var/spool/asterisk, var/lib/asterisk και var/run/asterisk όπου το Asterisk χειρίζεται αρχεία.

```
chown  -R asterisk  /var/spool/asterisk /var/lib/asterisk /var/lib/asterisk
```

Το όρισμα -R σημαίνει ότι η πρόσβαση επεκτείνεται και στους υποκαταλόγους.

Προκειμένου το Asterisk να αντιληφθεί ότι πρέπει να ενεργεί υπό την ταυτότητα του νέου χρήστη, πρέπει να χρησιμοποιήσουμε ένα αρχείο αρχικοποίησης (configuration file) που μας παρέχεται και αυτό στον πηγαίο κώδικα από όπου έγινε η εγκατάσταση του Asterisk.

Παρόμοια με την Ενότητα 1.3, το πρώτο στάδιο είναι η αντιγραφή του κατάλληλου αρχείου στη σωστή τοποθεσία:

```
cp root/asterisk-certified-13.8-cert/contrib/etc_default_asterisk /etc/default/asterisk
```

Προσοχή:

Η διαδρομή (path) του πρώτου αρχείου, σε περίπτωση που έχει κατέβει διαφορετική έκδοση ή έχει γίνει αποσυμπίεση σε οποιονδήποτε άλλο κατάλογο (directory) εκτός του root, είναι διαφορετική και πρέπει να προσεγγιστεί αντίστοιχα.

Σε δεύτερη φάση, πρέπει να επεξεργαστούμε το αρχείο αρχικοποίησης (configuration file) που τώρα πια ακούει στο όνομα asterisk και βρίσκεται στη διαδρομή (path) /etc/default/.

```
pico /etc/default/asterisk
```

Όταν, λόγω της άνωθεν εντολής, το περιβάλλον επεξεργασίας εμφανιστεί, σβήνουμε το σύμβολο "#" μπροστά από τις γραμμές #AST_USER="asterisk", #AST_GROUP="asterisk" (Εικόνα 2.2.3). Με αυτή μας την πράξη αυτές οι γραμμές παύουν να θεωρούνται σχόλια (comments) και έτσι θα ληφθούν υπόψη κατά την εκκίνηση του Asterisk, με αποτέλεσμα ο διακομιστής (server) να χειρίζεται μέσω της ταυτότητας asterisk όσα αρχεία είναι υπό την ιδιοκτησία του.

```
  GNU nano 2.2.6                                File: /etc/default/asterisk

# Startup configuration for the Asterisk daemon

# Uncomment the following and set them to the user/groups that you
# want to run Asterisk as. NOTE: this requires substantial work to
# be sure that Asterisk's environment has permission to write the
# files required  for  its  operation, including logs, its comm
# socket, the asterisk database, etc.
AST_USER="asterisk"
AST_GROUP="asterisk"
```

Εικόνα 2.2.3

Για να αποθηκεύσουμε το αρχείο, πληκτρολογούμε τον συνδυασμό ctrl+o, enter και για να βγούμε από το περιβάλλον επεξεργασίας πατάμε τον συνδυασμό ctrl+x.

Πληκτρολογώντας:

```
/etc/init.d/asterisk start
```

Το Asterisk επανέρχεται σε λειτουργία και το μόνο που μένει είναι η επιβεβαίωση ότι πράττει ως asterisk.

```
ps -U asterisk -u asterisk u
```

Με την παραπάνω εντολή, στην οθόνη εμφανίζονται όλες οι πληροφορίες των διεργασιών που εκτελούνται αυτή τη στιγμή από το μηχάνημα, άμεσα ή έμμεσα, υπό την ταυτότητα του χρήστη asterisk (Εικόνα 2.2.4).

```
root@asterisk-thesis:~# ps -U asterisk -u asterisk u
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
asterisk 11078  0.0  0.8 1469412 34736 ?       Ssl  13:19   0:16 /usr/sbin/asterisk -U asterisk -G asterisk
```

Εικόνα 2.2.4

Πληροφορίες σχετικά με την εντολή ps:

Η εντολή ps προέρχεται από την φράση “**p**rocess **s**tatus” (δηλαδή κατάσταση διεργασιών). Ο ρόλος της είναι να αποτυπώσει στην οθόνη ένα στιγμιότυπο της κατάστασης των διεργασιών που “τρέχουν” εκείνη την στιγμή στο σύστημα [Σφάλμα: Δεν βρέθηκε η πηγή παραπομπής].

Το αποτέλεσμά της είναι πολύ κοντά σε αυτό της εντολής top, με τη διαφορά ότι η εντολή top δεν προσφέρει ένα στιγμιαίο αποτέλεσμα αλλά επίβλεψη του συστήματος σε πραγματικό χρόνο.

Πηγές:

1. https://pl.wikipedia.org/wiki/Useradd
2. https://en.wikipedia.org/wiki/Pipeline_(Unix)
3. https://en.wikipedia.org/wiki/Cat_(Unix)
4. https://en.wikipedia.org/wiki/Grep
5. https://en.wikipedia.org/wiki/Ps_(Unix)

6. 2.3. Αλλαγή προκαθορισμένων αριθμών θύρας (default port number).

Στον κόσμο των εφαρμογών εξυπηρέτησης (server software), για να είναι δυνατή η επικοινωνία μεταξύ των διαφορετικών συστημάτων έχουν καθοριστεί ορισμένοι αριθμοί θυρών (port number) οι οποίοι είναι συγκεκριμένοι, ανάλογα της υπηρεσίας που προσφέρουν [π.χ. η πόρτα *80* αντιστοιχεί *στο πρωτόκολλο μεταφοράς υπερκειμένου* (Hypertext Transfer Protocol – HTTP) και όλοι όσοι αναζητούν αυτή την υπηρεσία την ανακαλύπτουν σε αυτήν την πόρτα].

Πληροφορίες σχετικά με τους αριθμούς θύρας (port number):

Σε έναν υπολογιστή ανατίθεται μια διεύθυνση διαδικτύου, έτσι ώστε να είναι δυνατή η επικοινωνία του μηχανήματος με οποιοδήποτε άλλο σύστημα στο διαδίκτυο. Σε έναν υπολογιστή, όμως, μπορεί να «τρέχουν» παραπάνω από μία εφαρμογές που απαιτούν επικοινωνία με κάποιο άλλο μηχάνημα. Για να μπορέσει το λειτουργικό σύστημα να αντιληφθεί ποια σύνδεση απευθύνεται σε ποια διεργασία, χρησιμοποιεί τον αριθμό θύρας (port number).

Υπάρχουν χιλιάδες πιθανοί συνδυασμοί. Από αυτούς όμως, 1024 πόρτες έχουν προκαθοριστεί μέσα από μία σύμβαση να είναι ταυτόσημες της υπηρεσίας που παρέχουν [1.].

Το γεγονός ότι οι αριθμοί θύρας (port number) είναι συγκεκριμένοι μπορεί, όμως, να δημιουργήσει προβλήματα ασφαλείας, καθώς ο επιτιθέμενος ξέρει πού να “χτυπήσει”. Ευτυχώς, οι αριθμοί θύρας (port number) δεν είναι δεσμευτικοί και για αυτό μπορούν να αλλάξουν.

Πρωτόκολλο ασφαλούς κελύφους (Secure Shell – SSH):

Η πρώτη αλλαγή είναι του αριθμού της θύρας (port number) του πρωτοκόλλου ασφαλούς κελύφους (Secure Shell – SSH). Αν κάποιος καταφέρει να διεισδύσει μέσα από το ασφαλές κέλυφος (Secure Shell – SSH), τότε, ανάλογα με την ιεραρχία του χρήστη που χρησιμοποίησε μπορεί να προκαλέσει απρόβλεπτα μεγάλη ζημιά.

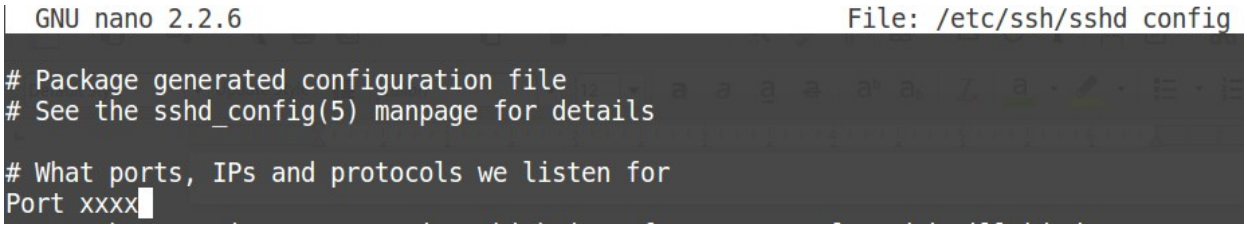
Μια σοφή αρχική κίνηση θα μπορούσε να θεωρηθεί η δημιουργία ενός αντίγραφου ασφαλείας των τωρινών ρυθμίσεων του ασφαλούς κελύφους (Secure Shell – SSH).

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config_backup
```

Συνεχίζοντας, επεξεργαζόμαστε το αρχείο που βρίσκεται στη διαδρομή (path) etc/ssh/sshd_config.

```
pico /etc/ssh/sshd_config
```

Όταν το περιβάλλον επεξεργασίας εκκινηθεί, αφαιρούμε το σύμβολο “#” μπροστά από την λέξη port και γράφουμε τον αριθμό που επιθυμούμε (Εικόνα 2.2.5).



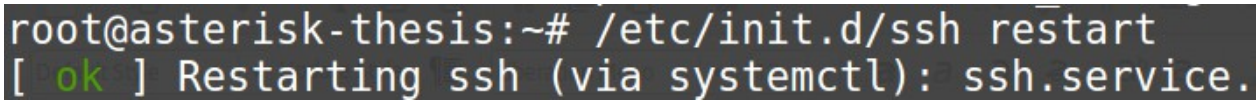
Εικόνα 2.2.5

Πληκτρολογούμε τον συνδυασμό ctrl+o, enter για να αποθηκεύσουμε το αρχείο, και για να βγούμε από το περιβάλλον επεξεργασίας πατάμε τον συνδυασμό ctrl+x.

Μετά σειρά έχει η επανεκκίνηση της διεργασίας (proccess).

```
/etc/init.d/ssh restart
```

Στο τερματικό (terminal) εμφανίζεται το μήνυμα “[ok] Restarting ssh (via systemctl): ssh.service.” που σηματοδοτεί και την επιτυχημένη αλλαγή της πόρτας (Εικόνα 2.2.6).



Εικόνα 2.2.6

Προσοχή:

Α. Για λόγους ασφαλείας, οι καινούριοι αριθμοί θύρας (port number) θα έχουν τη μορφή xxxx.

Β. Σε περίπτωση που χρησιμοποιούσατε μέχρι τώρα για σύνδεση στον διακομιστή (server):

```
ssh root@10.100.51.203
```

τώρα πρέπει να γίνεται χρήση του

```
ssh root@10.100.51.203 -pxxxx
```

όπου xxxx, ο αριθμός που επιλέξαμε στο αρχείο etc/ssh/sshd_config.

Asterisk:

Η δεύτερη εφαρμογή όπου πρέπει να αλλαχθεί ο αριθμός της θύρας (port number), είναι το ίδιο το Asterisk. Λόγο του ότι είναι πολύ δημοφιλές τηλεφωνικό σύστημα, υπάρχει σύμβαση το Asterisk να χρησιμοποιεί την πόρτα (port) 5060. Για να αλλάξουμε τον αριθμό της πόρτας (port number), πρέπει να αλλάξουμε την γραμμή στο αρχείο /etc/asterisk/sip.conf από “bindport = 5060 “ σε “bindport = 5080 “(Εικόνα 2.2.7).

```
pico /etc/asterisk/sip.conf
```

Θα πληκτρολογήσουμε τον συνδυασμό ctrl+o, enter για να αποθηκεύσουμε το αρχείο, και για να βγούμε από το περιβάλλον επεξεργασίας πατάμε τον συνδυασμό ctrl+x.



Εικόνα 2.2.7

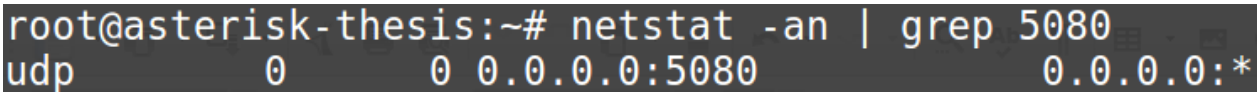
Προκειμένου να τεθεί η αλλαγή σε λειτουργία, πρέπει να εκτελεστεί η παρακάτω εντολή:

```
asterisk -X “sip reload”
```

Για να επιβεβαιώσουμε την αλλαγή, πρέπει να ελέγξουμε αν «τρέχει» στο λειτουργικό κάποια διεργασία (proccess) που να ακούει στην πόρτα (port) 5080.

```
netstat -an | grep 5080
```

Αν όλα πήγαν καλά, τότε πρέπει να εμφανιστεί στην οθόνη του τερματικού (terminal) η Εικόνα 2.2.8



Εικόνα 2.2.8

Πληροφορίες σχετικά με την εντολή netstat:

Η εντολή netstat προέρχεται από την φράση "***net****work* ***stat****istics*" (δηλαδή στατιστικά δικτύου). Ο ρόλος της είναι να αποτυπώσει στην οθόνη πληροφορίες του δικτύου όπως οι συνδέσεις δικτύου για το πρωτόκολλο ελέγχου μετάδοσης (Transmission Control Protocol - TCP), πίνακες δρομολόγησης (routing tables) και στατιστικά στοιχεία πρωτοκόλλου δικτύου (Internet Protocol – IP) [2.].

Διατίθεται σε λειτουργικά συστήματα βασισμένα στο Unix, συμπεριλαμβανομένων macOS, Linux, Solaris και BSD καθώς και σε λειτουργικά συστήματα που βασίζονται στα Windows NT, όπως τα Windows XP, Windows Vista, Windows 7, Windows 8 και Windows 10.

Πηγές:

1. https://en.wikipedia.org/wiki/Port_(computer_networking)
2. https://en.wikipedia.org/wiki/Netstat

2.4 Ενημέρωση του τοίχους ασφαλείας (firewall) για τους νέους κανόνες.

Στον διακομιστή (server), κάθε φορά που αλλάζουμε μια πόρτα (port number), προκειμένου να είναι δυνατή η επικοινωνία της εφαρμογής με το δίκτυο, πρέπει να ενημερωθεί το τείχος ασφαλείας (firewall).

Πληροφορίες σχετικά με το iptables, το τοίχος ασφαλείας του πυρήνα του Linux (firewall):

Τα συστήματα Linux έχουν στον πυρήνα του λογισμικού (Linux Kernel) μια υλοποίηση τοίχου ασφαλείας (firewall), που είναι διαχειρίσιμη μέσω της διεπαφής iptables [1.].

Σε οποιοδήποτε λογισμικό περιβάλλον, το τοίχος ασφαλείας είναι ένα πρόγραμμα που παρακολουθεί και ελέγχει την εισερχόμενη και εξερχόμενη κυκλοφορία του δικτύου, με βάση κάποιους προκαθορισμένους κανόνες.

Το iptables αποτελεί εργαλείο διαχείρισης του τοίχους ασφαλείας (firewall), παράγοντας κανόνες με τους οποίους είναι δυνατό το φιλτράρισμα της κυκλοφορίας του δικτύου και η αντίδραση του τοίχους ασφαλείας (firewall), σε περίπτωση εντοπισμού πακέτου που ταιριάζει στην περιγραφή των κανόνων.

Το iptables παρέχει τρεις βασικές λειτουργίες για την παραγωγή κανόνων:

- INPUT: Χειρίζεται όλα τα πακέτα που προέρχονται από το δίκτυο και προορίζονται για το σύστημα.
- OUTPUT: Ελέγχει την πληροφορία που εξέρχεται από το μηχάνημα.
- FORWARD: Ανακατευθύνει πακέτα που έχουν έρθει από το δίκτυο σε άλλα μηχανήματα που ανήκουν σε αυτό.

Προκειμένου να απλοποιηθεί η διαδικασία και να μην χρειαστεί να επέμβουμε ξανά στο τοίχος ασφαλείας (firewall) αργότερα όταν θα «τρέχουμε» και μια επιπλέον υπηρεσία η οποία θα δημιουργεί χρήστες του Asterisk, θα περάσουμε όλους τους κανόνες από το τελικό στάδιο του συστήματος της πτυχιακής μου στο δικό σας.

Το πρώτο βήμα είναι η δημιουργία αντιγράφου ασφαλείας, προκειμένου να υπάρχει δυνατότητα ανάκτησης της αρχικής κατάστασης, σε περίπτωση λάθους.

```
Iptables-save > ~/iptables.bu
```

Πληροφορίες σχετικά με την ανακατεύθυνση εξόδου :

Στο Linux-Unix πολλές φορές χρειάζεται να αποθηκεύσουμε την έξοδο μιας διεργασίας σε κάποιο αρχείο.

Η ανακατεύθυνση εξόδου γίνεται χρησιμοποιώντας τον ειδικό χαρακτήρα “>”, ανάμεσα στην εντολή και στο αρχείο που θέλουμε να αποθηκεύσουμε το αποτέλεσμα της διεργασίας (proceess).

Η διαφορά με την διασωλήνωση (piping), είναι ότι στην προκειμένη περίπτωση δεν ανακατευθύνουμε την έξοδο μιας διεργασίας (proccess) με σκοπό την χρήση της από κάποια άλλη, αλλά την αποθηκεύουμε σε κάποιο αρχείο με σκοπό την μελλοντική της χρήση.

Στη διαδρομή (path) ~/iptables.bu, έχει δημιουργηθεί ένα αρχείο που περιέχει όλους τους κανόνες του τοίχους ασφαλείας (firewall).

Στη συνέχεια, θεμιτό θα ήταν να αντιγραφεί το παρακάτω κείμενο (οι κανόνες που ισχύουν στο σύστημα της πτυχιακής μου) :

```
*mangle
:PREROUTING ACCEPT [83145:120824770]
:INPUT ACCEPT [83145:120824770]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [46823:2584014]
:POSTROUTING ACCEPT [46823:2584014]
COMMIT
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1:60]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP
-A INPUT -f -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG FIN,SYN,RST,PSH,ACK,URG -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -i eth0 -p tcp -m tcp --dport XXXX -m state --state NEW -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 443 -m state --state NEW -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 80 -m state --state NEW -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 5080 -j ACCEPT
-A INPUT -i eth0 -p udp -m udp --dport 5080 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-port-unreachable
COMMIT
*nat
:PREROUTING ACCEPT [164:6544]
:POSTROUTING ACCEPT [148:8939]
:OUTPUT ACCEPT [148:8939]
COMMIT
```

Να δημιουργηθεί ένα αρχείο :

```
touch ~/iptables.new
```

Πληροφορίες σχετικά με την εντολή touch:

Αποτελεί την εντολή που χρησιμοποιείται από τα συστήματα Linux-Unix για να ενημερώνεται η ημερομηνία τελευταίας επεξεργασίας ενός αρχείου ή ενός καταλόγου (directory) [2.].

Συνήθως, χρησιμοποιείται για τη δημιουργία νέων αρχείων. Η βασική της όμως ιδιότητα είναι ότι ενεργεί αποφεύγοντας το άνοιγμα, το κλείσιμο ή την αποθήκευση του αρχείου, έτσι ώστε να μην επηρεαστούν τα δεδομένα που περιέχει.

Η σωστή ημερομηνία τελευταίας επεξεργασίας ενός αρχείου είναι ζωτικής σημασίας για τα συστήματα Linux-Unix καθώς, με βάση αυτήν, προγραμματίζουν διαδικασίες (proccess) όπως η δημιουργία αρχείων ασφαλείας ή η αναβάθμιση ορισμένων εφαρμογών.

Και αφού ανοιχθεί σε περιβάλλον επεξεργασίας, να γίνει επικόλληση (ctrl+shif+v):

```
pico ~/iptables.new
```

Αποθηκεύστε το αρχείο (ctrl+shift+o, enter) και βγείτε από το περιβάλλον επεξεργασίας (ctrl+shift+x).

Το αρχείο που δημιουργήσαμε περιέχει κανόνες για τα παρακάτω:

- Το πρωτόκολλο ασφαλούς κελύφους (SSH) ακούει στη θύρα xxxx.
- Το ασφαλές πρωτόκολλο μεταφοράς υπερκειμένου (*Hypertext Transfer Protocol Secure* – HTTPS) ακούει στη θύρα 443.
- Το πρωτόκολλο μεταφοράς υπερκειμένου (*Hypertext Transfer Protocol* – HTTP) ακούει στη θύρα 80.
- Το Asterisk ακούει στη θύρα 5080.

Προσοχή:

Το xxxx πρέπει να αντικατασταθεί με τον αριθμό θύρας, που αντικαταστάθηκε στο πρωτόκολλο ασφαλούς κελύφους (SSH) στην προηγούμενη Ενότητα.

Προκειμένου να φορτωθούν οι νέοι κανόνες στο τοίχος ασφαλείας, εκτελούμε την εντολή :

```
iptables-restore ~/iptables.new
```

Οι καινούριοι κανόνες θα είναι σε ισχύ με την επανεκκίνηση της υπηρεσίας:

```
iptables-restore ~/iptables.new
```

Πηγές:

1. https://en.wikipedia.org/wiki/Iptables
2. https://en.wikipedia.org/wiki/Touch_(Unix)

2.5 Εγκατάσταση του Fail2Ban για την ενίσχυση του τοίχους ασφαλείας (firewall).

Το μεγαλύτερο μέρος των εφαρμογών που παρέχουν υπηρεσίες εξυπηρέτησης μέσω ενός διακομιστή (server side software), κρατάνε αρχεία καταγραφής (log file) της κατάστασής τους ανά τακτά χρονικά διαστήματα. Ένα από τα πιο σημαντικά, είναι τα αρχεία που περιέχουν πληροφορίες σχετικά με το ποιος χρησιμοποίησε την υπηρεσία, καθώς και μετά από πόσες προσπάθειες κατάφερε να αποκτήσει πρόσβαση σε αυτήν.

Σε περίπτωση εισβολής, αν κάποιος άνθρωπος διαβάσει έγκαιρα (κατά τη διάρκεια της επίθεσης) αυτά τα αρχεία, θα μπορέσει εύκολα να αντιληφθεί ποιος είναι ο επιτιθέμενος και πολύ πιθανόν να του παρεμποδίσει την είσοδο στην υπηρεσία.

Δυστυχώς, όμως, είναι αδύνατον ένας άνθρωπος να επιβλέπει συνεχώς όλα τα αρχεία καταγραφής (log file). Για αυτό, υπάρχει η ανάγκη να επιβλέπονται από τον ίδιο τον διακομιστή (server). Το πρόγραμμα που κάνει αυτή την δουλειά λέγεται fail2ban.

Πληροφορίες σχετικά με το λογισμικό fail2ban:

Το Fail2Ban λειτουργεί μέσω της παρακολούθησης των αρχείων καταγραφής (log file) των εφαρμογών που θωρακίζει. Επειδή δεν φτιάχνουν με τον ίδιο τρόπο όλες οι εφαρμογές τα αρχεία καταγραφής (log file) τους, δεν υπάρχει μοναδικός τρόπος για την ανάγνωσή τους [1.].

Το fail2ban, για να μπορέσει να κατανοήσει την εκάστοτε εφαρμογή, έχει ορισμένα αρχεία τα οποία περιγράφουν με πολύ συγκεκριμένο τρόπο την δομή των αρχείων καταγραφής (log file).

Κατά την επίβλεψη των αρχείων καταγραφής (log file), το fail2ban προσπαθεί να εντοπίσει και να εμποδίσει οποιεσδήποτε διευθύνσεις διαδικτύου (IP address) να κάνουν πάρα πολλές αποτυχημένες απόπειρες σύνδεσης μέσα σε ένα χρονικό πλαίσιο που ορίζεται από τον διαχειριστή.

Για να εγκατασταθεί το fail2ban, απλά «τρέχουμε» την εντολή:

```
apt-get install fail2ban
```

```
scripter@scripter-SATELLITE-P50-A-118 ~ $ sudo apt-get install fail2ban
[sudo] password for scripter:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  mailx monit
Recommended packages:
  whois python3-pyinotify python3-systemd
The following NEW packages will be installed:
  fail2ban
0 upgraded, 1 newly installed, 0 to remove and 611 not upgraded.
Need to get 227 kB of archives.
After this operation, 1180 kB of additional disk space will be used.
Get:1 http://ubuntu.otenet.gr xenial/universe amd64 fail2ban all 0.9.3-1 [227 kB]
Fetched 227 kB in 1s (139 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 207120 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.9.3-1_all.deb ...
Unpacking fail2ban (0.9.3-1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for ureadahead (0.100.0-19) ...
ureadahead will be reprofiled on next reboot
Processing triggers for systemd (229-4ubuntu4) ...
Setting up fail2ban (0.9.3-1) ...
```

Εικόνα 2.5.1

Όταν η εγκατάσταση ολοκληρωθεί, πρέπει να επεξεργαστούμε το αρχείο logger.conf που ανήκει στο Asterisk (/etc/asterisk/logger.conf).

```
pico /etc/asterisk/logger.conf
```

Η γραμμή που αντιστοιχεί στην μεταβλητή που ξεκινάει με “dateformat=” πρέπει να έχει την μορφή “dateformat=%F %T” και η γραμμή που περιέχει το “security => security” δεν πρέπει να ξεκινάει με τον χαρακτήρα “;”(Εικόνα 2.5.2 και 2.5.3).

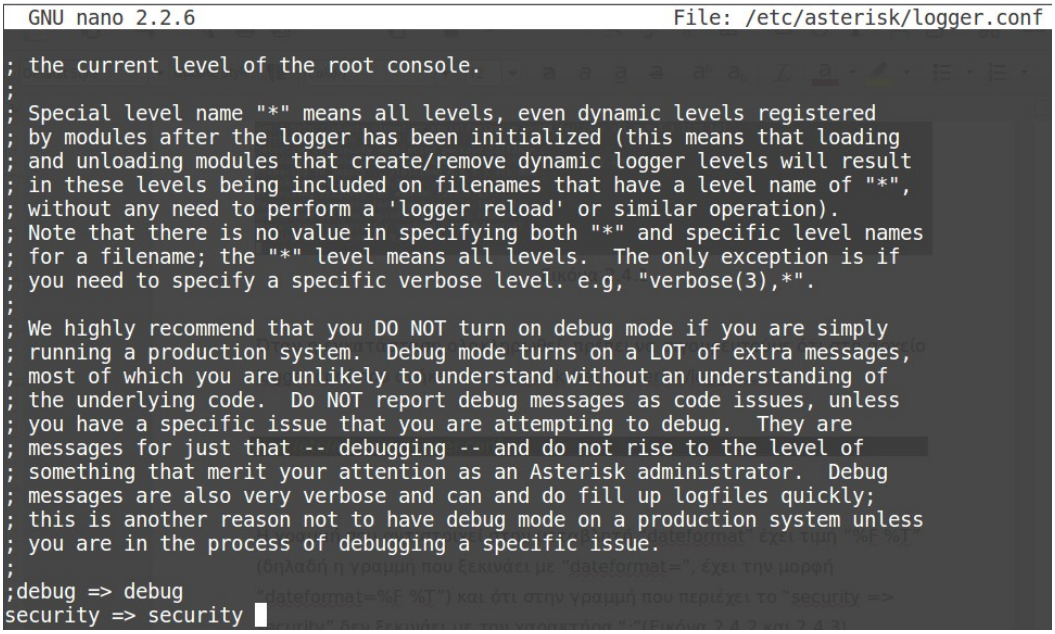
```
  GNU nano 2.2.6                                File: /etc/asterisk/logger.conf

;
; Logging Configuration
;
; In this file, you configure logging to files or to
; the syslog system.
;
; "logger reload" at the CLI will reload configuration
; of the logging system.

[general]
;
; Customize the display of debug message time stamps
; this example is the ISO 8601 date format (yyyy-mm-dd HH:MM:SS)
;
; see strftime(3) Linux manual for format specifiers.  Note that there is also
; a fractional second parameter which may be used in this field.  Use %1q
; for tenths, %2q for hundredths, etc.
;

dateformat=%F %T                ; ISO 8601 date format
```

Εικόνα 2.5.2



Εικόνα 2.5.3

Αυτή μας η κίνηση, επιτρέπει στο Asterisk να κρατάει αρχεία ασφαλείας (security logs), τα οποία να καταγράφουν τα γεγονότα (events) με τη μορφή ημερομηνίας χχχχ-μμ-ηη ωω:λλ:δδ.

Όπου :

- χ : χρόνος
- μ : μήνας
- η : ημέρα
- ω : ώρα
- λ : λεπτό
- δ : δευτερόλεπτο

Το επόμενο βήμα είναι να δημιουργήσουμε το αρχείο περιγραφής της δομής των αρχείων καταγραφής (log file) του Asterisk.

```
touch /etc/fail2ban/filter.d/asterisk.conf
```

Αντιγράφουμε το παρακάτω κείμενο, το οποίο έχει συντάξει η εταιρεία πίσω από το Asterisk Digium:

```
# Fail2Ban configuration file
#
#
# $Revision: 250 $
#

[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
#before = common.conf

[Definition]

#_daemon = asterisk

# Option:  failregex
# Notes.:  regex to match the password failures messages in the logfile. The
#          host must be matched by a group named "host". The tag "<HOST>" can
#          be used for standard IP/hostname matching and is only an alias for
#          (?:::f{4,6}:)?(?P<host>\S+)
# Values:  TEXT
#

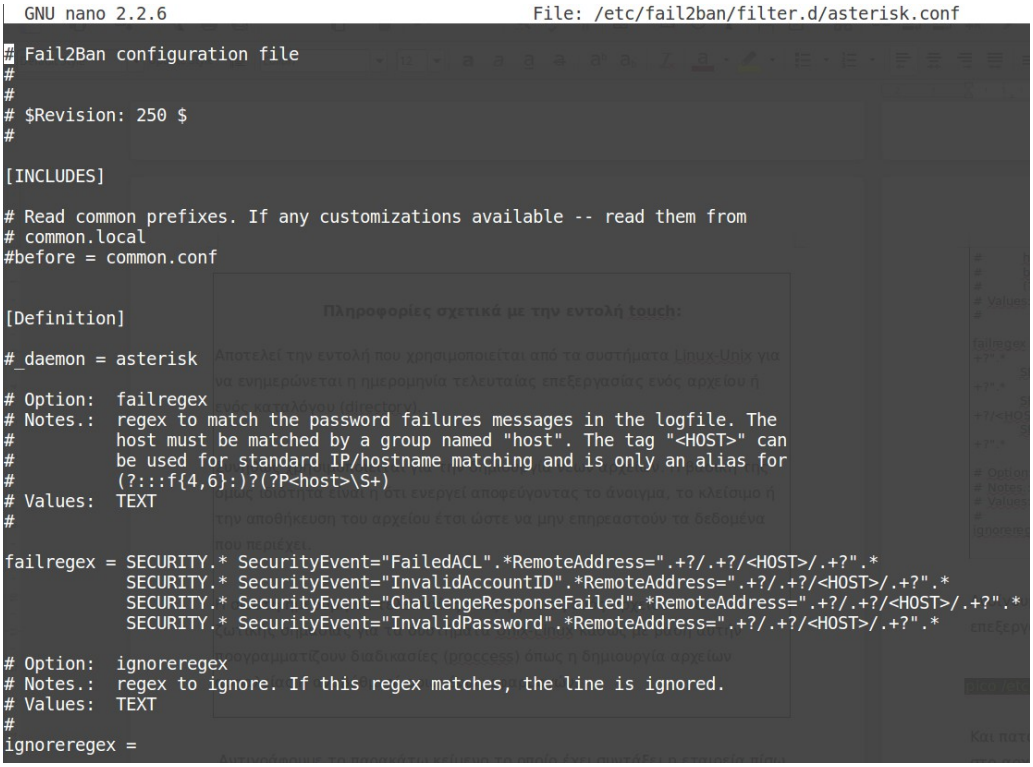
failregex = SECURITY.* SecurityEvent="FailedACL".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
            SECURITY.* SecurityEvent="InvalidAccountID".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
            SECURITY.* SecurityEvent="ChallengeResponseFailed".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
            SECURITY.* SecurityEvent="InvalidPassword".*RemoteAddress=".+?/.+?/<HOST>/.+?".*

# Option:  ignoreregex
# Notes.:  regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Ανοίγουμε το αρχείο που μόλις δημιουργήσαμε, έτσι ώστε να το επεξεργαστούμε:

```
pico /etc/fail2ban/filter.d/asterisk.conf
```

και πατώντας τον συνδυασμό ctrl+shift+v επικολλούμε το άνωθεν κείμενο στο αρχείο (Εικόνα 2.5.4). Αποθηκεύουμε με ctrl+o enter και αποχωρούμε από το περιβάλλον επεξεργασίας με τον συνδυασμό ctrl+x.



Εικόνα 2.5.4

Το αρχείο που δημιουργήσαμε περιέχει επεξηγήσεις για το fail2ban στα γεγονότα:

- Απόπειρας εισόδου με μη υπαρκτό χρήστη.
- Απόπειρας εισόδου με υπαρκτό χρήστη, αλλά λανθασμένο συνθηματικό.
- Μη ορθής επιβεβαίωσης στοιχείων μετά από κάποιο χρονικό διάστημα σύνδεσης.
- Κακής διαγωγής.

Προκειμένου να ορίσουμε στο fail2ban πώς να συμπεριφερθεί όταν εντοπίσει μια πιθανή απειλή, αντιγράφουμε το παρακάτω μοτίβο (συντάκτης Digium):

```
[asterisk-iptables]
enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, protocol=all]
logpath  = /var/log/asterisk/security
maxretry = 3
bantime = 86400
```

Ανοίγουμε το αρχείο με διαδρομή (path) /etc/fail2ban/jail.conf σε περιβάλλον επεξεργασίας:

```
pico /etc/fail2ban/jail.conf
```

και στο τέλος του αρχείου, με τον συνδυασμό των πλήκτρον ctrl+shift+v, επικολλούμε το κείμενο.

Σε αυτό το σημείο, ουσιαστικά, έχουμε προγραμματίσει το fail2ban όταν εντοπίσει τρεις φορές ύποπτη κίνηση, να απαγορεύσει την πρόσβαση στην ύποπτη διεύθυνση διαδικτύου για μία ημέρα.

Πηγές:

1. https://en.wikipedia.org/wiki/Fail2ban

ΚΕΦΑΛΑΙΟ 3

Παραμετροποίηση του Asterisk.

3.1 Περιγραφή της δομής του αρχείου ρυθμίσεων του Asterisk.

Σε αυτό το κεφάλαιο θα κάνουμε τις βασικές ρυθμίσεις ώστε το Asterisk να μπορεί να χρησιμοποιηθεί ως τηλεφωνικό κέντρο. Για να πετύχουμε την ορθή λειτουργία του, αρχικά καλό θα ήταν να κατανοήσουμε πλήρως τη δομή του αρχείου ρυθμίσεων του Asterisk (Asterisk configuration file).

Πληροφορίες σχετικά με τα αρχεία ρυθμίσεων (configuration file):

Στην επιστήμη της πληροφορικής, τα αρχεία ρυθμίσεων (configuration file) έχουν ως σκοπό την παραμετροποίηση και την αρχικοποίηση των μεταβλητών ορισμένων προγραμμάτων [1.].

Στα λειτουργικά συστήματα Linux-Unix, ειδικά στις εκδόσεις που δεν έχουν γραφικό περιβάλλον (server edition), οι περισσότερες εφαρμογές απαιτούν από τον διαχειριστή να δημιουργήσει ή να τροποποιήσει τα αρχεία ρυθμίσεων (configuration file)και για αυτόν τον λόγο, είναι απαραίτητο να είναι γραμμένα σε απλό κείμενο κατανοητό από τον άνθρωπο (human-editable plain text.).

Μερικά προγράμματα ή λειτουργικά, φορτώνουν τα αρχεία ρυθμίσεων (configuration file) κατά την εκκίνησή τους, ενώ άλλα ανά τακτά χρονικά διαστήματα. Στην πρώτη περίπτωση, αν παραμετροποιηθούν τα αρχεία ρυθμίσεων (configuration file), τότε στο πρόγραμμα ή στο λειτουργικό πρέπει να δοθεί η εντολή να τα επαναφορτώσει.

Πέρα από το ποια κατάληξη θα έχουν (στα Linux-Unix `.cnf`, `.conf`, `.cfg`, `.cf` και `.ini`), δεν υπάρχει καμία άλλη σύμβαση για το πώς θα πρέπει να είναι η δομή τους.

Τα αρχεία ρυθμίσεων του Asterisk (Asterisk configuration file), έχουν μια πολύ συγκεκριμένη δομή. Αποτελούνται από context (πλαίσια), variable (ετικέτες), value (τιμές) και σχόλια (Παράδειγμα 3.1.1).

Παράδειγμα 3.1.1:
[context] variable = value ; σχόλιο

Βοήθημα:

Τα σχόλια αποτελούν επιπλέον πληροφορία για τον αναγνώστη του αρχείου ρυθμίσεων (configuration file) αλλά όχι για το ίδιο το Asterisk.

Ένα σχόλιο ξεκινάει με τον χαρακτήρα “;” και τελειώνει στο τέλος της γραμμής.

Τα context (πλαίσια), διακρίνουν ένα σύνολο από ζευγάρια variable (μεταβλητή) και value (τιμής) από ένα άλλο αντίστοιχο σύνολο. Με λίγα λόγια, κατακερματίζουν το εκάστοτε αρχείο ρυθμίσεων (configuration file) σε διακριτά μέρη, με σκοπό την σωστότερη και ευκολότερη σύνταξή του. Αποτέλεσμα αυτού είναι η πιο ολοκληρωμένη λειτουργία του Asterisk, δίνοντάς του την δυνατότητα παροχής υπηρεσιών ανάλογα με την περίσταση (π.χ. αν ένας χρήστης είναι φοιτητής ή καθηγητής).

Ένα context (πλαίσιο), είναι η περιοχή που διακρίνεται από το μοτίβο “[Όνομα context]” μέχρι την επανάληψη του μοτίβου ή το τέλος του αρχείου (Σχήμα 3.1.1).

```
[Μήλο]
χρώμα = κόκκινο
[Αχλάδι]
χρώμα = πράσινο
[Μπανάνα]
χρώμα = κίτρινο
```

Σχήμα 3.1.1

Το context (πλαίσιο) σε πολλά σημεία θυμίζει την έννοια της κλάσης στον αντικειμενοστραφή προγραμματισμό. Ο κύριος ρόλος του είναι η δημιουργία δομών (struct), δηλαδή τμημάτων που αποτελούν διακριτές οντότητες και μπορούν να κληρονομηθούν. Δηλαδή τα ζευγάρια variable (μεταβλητή) και value (τιμή) που έχουν συμπεριληφθεί σε αυτό το context (πλαίσιο)

να μπορούν να χρησιμοποιηθούν από κάποιο άλλο αυτούσια, χωρίς να ξαναγραφτούν.

Κατά την διάρκεια της ανάλυσης του αρχείου ρυθμίσεων (configuration file parse), το Asterisk κατανοεί την κληροδότηση από το μοτίβο “[Όνομα context] (Όνομα κάποιου άλλου context που έχει οριστεί πιο πάνω)”.

Υπάρχουν δύο είδη context (πλαισίων) που μπορούν να κληρονομηθούν:

Α. Τα context (πλαίσια) τα οποία αποτελούν μόνο πρότυπο.

Αυτό το είδος context (πλαισίου) αγνοείται από το Asterisk σαν οντότητα, κατά την ανάλυση του αρχείου ρυθμίσεων (configuration file parse) και χρησιμοποιείται μόνο σαν μία μονάδα που μπορεί να κληρονομηθεί. Αυτή η κατηγορία context (πλαισίου) γίνεται κατανοητή κατά την διάρκεια της ανάλυσης του αρχείου ρυθμίσεων (configuration file parse) από το Asterisk από το “(!)” δίπλα από το μοτίβο “[Όνομα context]” (παράδειγμα 3.1.2).

Παράδειγμα 3.1.2:	
Τρόπος γραφής:	Τρόπος ανάγνωσης από το Asterisk:
[Τρίκλινο](!) κατηγορία = δωμάτιο κρεβάτια = τρία [Δωμάτιο 312](Τρίκλινο) τηλέφωνο = 312	[Δωμάτιο 312] κατηγορία = δωμάτιο κρεβάτια = τρία τηλέφωνο = 312

Β. Τα context (πλαίσια) τα οποία αποτελούν πρότυπο και προσφέρουν και λειτουργικότητα.

Σε αντίθεση με την προηγούμενη κατηγορία, εδώ το context (πλαίσιο) αποτελεί ένα αναγκαίο τμήμα του αρχείου ρυθμίσεων (configuration file) και πρέπει να συμπεριληφθεί κατά την ανάλυσή του (Παράδειγμα 3.1.3).

Παράδειγμα 3.1.3:	
Τρόπος γραφής:	Τρόπος ανάγνωσης από το Asterisk:
[Χρήστης 1] Οδός = Ομήρου [Χρήστης 2] Οδός = Γράμμου [Λίστα Οδών](Χρήστης 1, Χρήστης 2)	[Χρήστης 1] Οδός = Ομήρου [Χρήστης 2] Οδός = Γράμμου [Λίστα Οδών] Οδός = Ομήρου Οδός = Γράμμου

Πηγές:

1. https://www.voip-info.org/wiki/view/Asterisk+config+files

3.2 Παραμετροποίηση του Asterisk.

Για να μπορέσει το Asterisk να παρέχει υπηρεσίες τηλεφωνικού κέντρου, κατά ελάχιστο, θα πρέπει να τροποποιηθούν δυο αρχεία ρυθμίσεων (configuration file):

- sip.conf (/etc/asterisk/sip.conf): που αποτελεί το αρχείο περιγραφής του καναλιού (channel) επικοινωνίας. Περιέχει πληροφορίες που έχουν να κάνουν με την ρύθμιση των τεχνικών παραμέτρων των διαύλων επικοινωνίας (π.χ. τον τρόπο που θα κωδικοποιείται η φωνή), καθώς και τους χρήστες που μπορούν να το χρησιμοποιούν.
- extensions.conf (/etc/asterisk/extensions.conf): Σε αυτό το αρχείο προγραμματίζουμε την συμπεριφορά του τηλεφωνικού κέντρου, με βάση τον κωδικό που κάλεσε ο τελικός χρήστης (π.χ. την παράδοση μηνυμάτων όταν ένας χρήστης είναι εκτός σύνδεσης). Η γλώσσα που χρησιμοποιείται λέγεται Dialplan.

Α. Επεξεργασία του sip.conf.

Πληκτρολογούμε την εντολή:

```
pico /etc/asterisk/sip.conf
```

Όταν το περιβάλλον επεξεργασίας εμφανιστεί, παρατηρούμε ότι υπάρχει ήδη μια στοιχειώδης δομή. Αυτή η δομή παράχθηκε αυτοματοποιημένα στην Ενότητα 1.2 (Σελ. 30) και αποτελεί μια πολύ ωραία βάση που παρέχει τις βασικές ρυθμίσεις. Για να φτιάξουμε τις δικές μας, πιο εξιδανικευμένες, ρυθμίσεις καλό θα ήταν να κατανοήσουμε τα υποχρεωτικά context (πλαίσια):

- general : περιέχει όλες τις γενικές ρυθμίσεις που ισχύουν για όλους τους χρήστες και αφορούν στο δίκτυο, την ασφάλεια και την παροχή υπηρεσιών.
- authentication: περιέχει τις ρυθμίσεις ταυτοποίησης και των δικαιωμάτων των χρηστών.

Context [general]

Το context [general], για την ευκολότερη κατανόησή του, μπορεί να αναλυθεί σε τρία μέρη :

1. Το κομμάτι που παραμετροποιεί το δίκτυο.

2. Το κομμάτι που προσφέρει επιπλέον ασφάλεια.

3. Το κομμάτι που περιγράφει τις παραχωρούμενες υπηρεσίες.

Δίκτυο:

Το παρακάτω κείμενο παρουσιάζει τις ρυθμίσεις παραμετροποίησης του δικτύου.

```
[general]


;
;Network


transport = udp  ;(1)
udpbindaddr = 0.0.0.0 ;  (2)
bindport = 5080 ;(3)
tcpenable = no  ; (4)
```

Επεξηγήσεις:

(1): Επιλέγουμε το πρωτόκολλο δεδομενογραμμάτων χρήστη (UDP), γιατί θέλουμε επικοινωνία και δεν μας απασχολεί να χαθεί κάποιο μικρό μέρος της.

(2): Με την ρύθμιση 0.0.0.0, το Asterisk ακούει όλες τις διευθύνεις του πρωτοκόλλου διαδικτύου (IP address), του πρωτοκόλλου δεδομενογραμμάτων χρήστη (UDP) που έχουν ανατεθεί στο μηχάνημα (στην περίπτωση αυτή μόνο την 10.100.51.203).
(3): Όπως είδαμε και στην Ενότητα 2.3 (Σελ. 46), το Asterisk ακούει στην πόρτα δικτύου (port number) 5080.

(4): Απενεργοποιούμε την λειτουργία του πρωτοκόλλου ελέγχου μεταφοράς (TCP), για να «τρέχει» μόνο το πρωτόκολλο δεδομενογραμμάτων χρήστη (UDP).



Εικόνα 3.1.1

Πληροφορίες σχετικά με το πρωτόκολλο δεδομενογραμμάτων χρήστη (UDP).

Το πρωτόκολλο δεδομενογραμμάτων χρήστη (User Datagram Protocol - UDP) είναι ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται στο διαδίκτυο.

Ένα από τα κύρια χαρακτηριστικά του UDP, είναι ότι δεν εγγυάται αξιόπιστη επικοινωνία. Χρησιμοποιείται όταν η "γρήγορη" παράδοση των πακέτων είναι πιο σημαντική από την "ακριβή" παράδοση, π.χ. στη μετάδοση ομιλίας και βίντεο. Αντιθέτως, το πρωτόκολλο TCP διαθέτει όλους τους απαραίτητους μηχανισμούς ελέγχου και επιβολής της αξιοπιστίας και συνεπώς μπορεί να εγγυηθεί την αξιόπιστη επικοινωνία μεταξύ των υπολογιστών. Η έλλειψη των μηχανισμών αυτών από το πρωτόκολλο UDP το καθιστά αρκετά πιο γρήγορο και αποτελεσματικό, τουλάχιστον για τις εφαρμογές εκείνες που δεν απαιτούν αξιόπιστη επικοινωνία.

Οι εφαρμογές audio και video streaming χρησιμοποιούν κατά κόρον πακέτα UDP. Για τις εφαρμογές αυτές είναι πολύ σημαντικό τα πακέτα να παραδοθούν στον παραλήπτη σε σύντομο χρονικό διάστημα, ούτως ώστε να μην υπάρχει διακοπή στη ροή του ήχου ή της εικόνας. Κατά συνέπεια, προτιμάται το πρωτόκολλο UDP γιατί είναι αρκετά γρήγορο, παρόλο που υπάρχει η πιθανότητα μερικά πακέτα UDP να χαθούν. Στην περίπτωση που χαθεί κάποιο πακέτο, οι εφαρμογές αυτές διαθέτουν ειδικούς μηχανισμούς διόρθωσης και παρεμβολής, ούτως ώστε ο τελικός χρήστης να μην παρατηρεί καμία αλλοίωση ή διακοπή στη ροή του ήχου και της εικόνας λόγω του χαμένου πακέτου. Σε αντίθεση με το πρωτόκολλο TCP, το UDP

υποστηρίζει broadcasting, δηλαδή την αποστολή ενός πακέτου σε όλους τους υπολογιστές ενός δικτύου, και multicasting, δηλαδή την αποστολή ενός πακέτου σε κάποιους συγκεκριμένους υπολογιστές ενός δικτύου. Η τελευταία δυνατότητα, χρησιμοποιείται πολύ συχνά στις εφαρμογές audio και video streaming (όπως στην περίπτωση του Asterisk), ούτως ώστε μία ροή ήχου ή εικόνας να μεταδίδεται ταυτόχρονα σε πολλούς συνδρομητές [1.].

Ασφάλεια:

Πέρα από τις ενέργειες που κάναμε στο κεφάλαιο 2 (Σελ. 38), το Asterisk μπορεί να δημιουργήσει ένα είδος αυτοπροστασίας με τις ρυθμίσεις που παρουσιάζονται στο παρακάτω κείμενο (Εικόνα 3.1.2).

```
;
;Security


context = default  ; (1)
subscribecontext = default ;(2)
allowoverlap = no  ; (3)
alwaysauthreject = yes ; (4)
allowguest = no ; (5)
directmedia = no ; (6)
```

Επεξηγήσεις:

(1): Με αυτή την ρύθμιση, ορίζουμε σε πιο context (πλαίσιο) θα υπάγεται κάποιος χρήστης ο οποίος δεν έχει ορίσει κάποιο άλλο. Συνήθως, αυτού του είδους οι χρήστες είναι εισβολείς και για αυτό και το context (πλαίσιο) default στο αρχείο extensions.conf (που θα αναλύσουμε παρακάτω) είναι κενό, μην επιτρέποντας καμία ενέργεια.

(2): Το subscribecontext καθορίζει αν κάποιος χρήστης δεν περιλαμβάνεται στο αρχείο sip.conf, πού να ψάξει να το βρει στο αρχείο extensions.conf. Συνήθως, και σε αυτή την περίπτωση υπάρχει εισβολή και για αυτό το context (πλαίσιο) default στο αρχείο extensions.conf είναι κενό, όπως αναφέρθηκε και πάνω.

(3): Η λειτουργία overlap (επικάλυψης), μοιάζει με το autocomplete στα κινητά ή τους

υπολογιστές. Είναι απενεργοποιημένη, γιατί αν ένας εισβολέας προσπαθήσει να μαντέψει κάποιους τηλεφωνικούς αριθμούς, η δουλειά του θα γίνει πολύ πιο εύκολη διότι με την εισαγωγή μερικών ψηφίων αυτόματα θα του προτείνονται και τα υπόλοιπα.

(4): Η ρύθμιση αυτή επιστρέφει την ίδια απάντηση, όταν επιχειρεί ένας επιτιθέμενος να μαντέψει τους κωδικούς πρόσβασης, είτε έκανε λάθος το συνθηματικό (password), είτε το όνομα χρήστη, δυσχεραίνοντας το έργο του.

(5): Δεν πρέπει να επιτρέπεται η σύνδεση σε άτομα που δεν θεωρούμε χρήστες.

(6): Παρόλο που το να μην υποστηρίζονται απευθείας συνδέσεις μεταξύ των χρηστών κοστίζει σε φόρτο του διακομιστή (server), είναι σοφό να περνάνε τα δεδομένα πρώτα από το Asterisk, έτσι ώστε να μπορεί να διαχειριστεί την κλήση ανάλογα (έχουν σημειωθεί επιθέσεις που μέσω του sip, ο επιτιθέμενος έχει πάρει τον έλεγχο της άλλης συσκευής και η κλήση δεν μπορεί να τερματιστεί από το θύμα).

```
;
;Security

context = default  ; Default context for incoming calls. Defaults to 'default',before default was public
subscribecontext = default
allowoverlap = no  ; Disable overlap dialing support. (Default is yes)
alwaysauthreject=yes;  sipvicious protection
allowguest = no ; security support
directmedia = no ; do not try to reinvite peers
```

Εικόνα 3.1.2

Υπηρεσίες:

Η παροχές που θέλουμε να προσφέρουμε, πέραν της φωνητικής κλήσης, είναι η κλήση βίντεο και η αποστολή μηνυμάτων .

```
;
; Fetures options


qualify = yes ; (1)
videosupport=yes ; (2)
accept_outofcall_message = yes ; (3)
outofcall_message_context = message ; (4)
auth_message_requests=yes ; (5)
dtmfmode = info ; (6)
```

Επεξηγήσεις:

(1): Προκειμένου να είναι δυνατή η αποστολή μηνυμάτων όταν ο χρήστης είναι εκτός σύνδεσης, αυτή η ρύθμιση πρέπει να είναι ενεργή. Αυτό που κάνει σαν λειτουργία, είναι κάθε ένα λεπτό να ελέγχει αν ο χρήστης είναι εντός σύνδεσης.

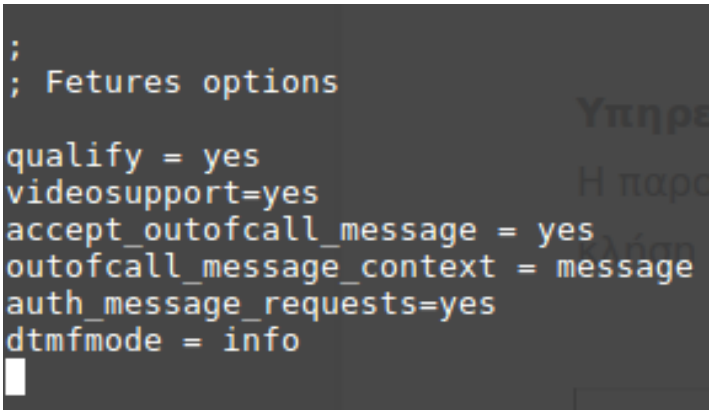
(2): Η ρύθμιση αυτή παρέχει την δυνατότητα μεταφοράς εικόνας μέσω του καναλιού επικοινωνίας.

(3): Το Asterisk, σε παλαιότερες εκδόσεις, δεν υποστήριζε μηνύματα εκτός κλήσης και για αυτό, αν θέλουμε αυτήν την παροχή θα πρέπει, στις σύγχρονες εκδόσεις, να την ενεργοποιήσουμε.

(4): Καθορίζει σε ποιο context (πλαίσιο) να απευθυνθούν τα μηνύματα τα οποία στάλθηκαν εκτός κλήσης.

(5): Με αυτή την ρύθμιση ενεργοποιημένη, το Asterisk έχει πρόσβαση στο περιεχόμενο των μηνυμάτων που ανταλλάσσονται μεταξύ των χρηστών. Όσο αθέμιτο και αν ακούγεται κάποιος να έχει πρόσβαση στο περιεχόμενο των μηνυμάτων, είναι αναγκαίο κακό, προκειμένου να είναι δυνατή η αποστολή σε έναν χρήστη που είναι εκτός σύνδεσης.

(6): Αυτή η ρύθμιση καθορίζει με ποιο πρότυπο σηματοδότησης πολλαπλών συχνοτήτων διπλής συχνότητας (Dual-tone multi-frequency signaling - DTMF) θα γίνεται η ανταλλαγή σημάτων κατά την κλήση.



Εικόνα 3.1.3

Πληροφορίες σχετικά με τη σηματοδότηση πολλαπλών συχνοτήτων διπλής συχνότητας (DTMF).

Η σηματοδότηση πολλαπλών συχνοτήτων διπλής συχνότητας (DTMF) είναι ένα τηλεπικοινωνιακό σύστημα σηματοδότησης που χρησιμοποιεί τη ζώνη συχνοτήτων της φωνής για την κωδικοποίηση μηνυμάτων και την διακίνησή τους μέσω τηλεφωνικών γραμμών [2.].

Χρησιμοποιεί ένα σύνολο οκτώ συχνοτήτων ήχου, που εκπέμπονται σε ζεύγη, για να αντιπροσωπεύουν 16 σήματα. Κάθε συχνότητα παράγεται από το πάτημα ενός κουμπιού στο πληκτρολόγιο της τηλεφωνικής συσκευής.

Context [authentication]

Το context [authentication], για την ευκολότερη κατανόησή του, μπορεί να αναλυθεί σε δύο μέρη :

1. Το κομμάτι που περιέχει τα δομικά πρότυπα.
2. Το κομμάτι που περιέχει τους χρήστες και τα χαρακτηριστικά τους.

Δομικά πρότυπα:

Όπως αναφέραμε και στην Ενότητα 3.1 (Σελ. 60), υπάρχει ένα είδος context (πλαισίου), το οποίο αποτελεί μόνο δομικό στοιχείο, καθώς δεν αποτελεί βασική πληροφορία αλλά ένα πρότυπο δομής που κληροδοτείται.

Στην περίπτωση μας, μπορούν να φτιαχτούν πέντε πρότυπα. Τα δύο πρώτα θα προσφέρουν στους απογόνους τους βασικά δικαιώματα, όπως το δικαίωμα κλήσης βίντεο (Video-Audio) ή μόνο φωνητικής κλήσης (Audio), ενώ τα τρία επόμενα θα περιγράφουν την ιεραρχία χρηστών (Καθηγητές>Προσωπικό>Φοιτητές).

Δομές Δικαιωμάτων.

Σε αυτές τις δύο δομές, πρέπει να ορίσουμε με μεγάλη ακρίβεια ποιες κωδικοποιήσεις θα είναι επιτρεπτές στην περίπτωση της κλήσης βίντεο [Video-Audio] και ποιες στην περίπτωση της κλήσης φωνής [Audio], ώστε να μπορεί να καταλάβει το ένα άκρο το άλλο.

Context [Audio]

Επειδή όλες οι ιεραρχίες χρηστών θα έχουν δυνατότητα φωνητικής κλήσης, αυτή αποτελεί τον μεγαλύτερο πρόγονο, αφού θα την κληρονομούν όλα τα είδη των χρηστών.

```
[authentication]

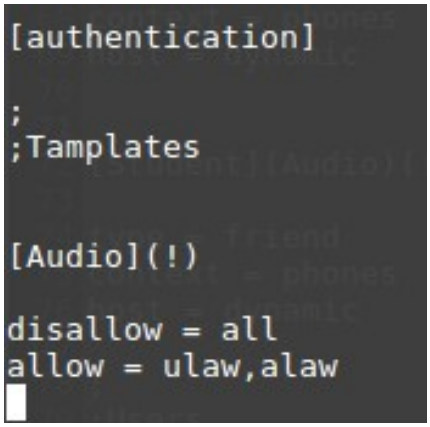
;
;Tamplates

[Audio](!)

disallow = all ; (1)
allow = ulaw,alaw ; (2)
```

Επεξηγήσεις:

(1): Απενεργοποιούμε όλες τις κωδικοποιήσεις.
(2): Για να ενεργοποιήσουμε τις δύο πιο διαδεδομένες κωδικοποιήσεις φωνής.



Εικόνα 3.1.4

Context [Video-Audio]

Ένας χρήστης με δυνατότητα κλήσης βίντεο, πρέπει να έχει και δυνατότητα παροχής φωνητικής κλήσης. Για αυτόν τον λόγο, το context (πλαίσιο) αυτό πρέπει να κληρονομεί το αντίστοιχο της φωνητικής κλήσης.

```
[Video-Audio](Audio)(!)

;Video
allow = h264 ;(1)
allow = h263 ;(1)
allow = vp8 ;(1)
```

Επεξηγήσεις:

(1): Ενεργοποίηση των κωδικοποιήσεων που είναι πιο διαδεδομένες στην περίπτωση του βίντεο.

Δομές Ιεραρχίας.

Στο σύστημά μας υπάρχουν τρία επίπεδα ιεραρχίας :

1. Το ανώτερο επίπεδο, που είναι οι καθηγητές - [Professor].
2. Το δεύτερο επίπεδο, που είναι το προσωπικό του Πανεπιστημίου – [Staff].
3. Το επίπεδο των φοιτητών, που είναι και το πιο ανίσχυρο - [Student].

Οι καθηγητές και το προσωπικό έχουν δικαίωμα φωνητικής και βίντεο κλήσης, ενώ οι φοιτητές μόνο φωνητικής. Έτσι, οι καθηγητές και το προσωπικό κληρονομούν το context (πλαίσιο) της κλήσης βίντεο και οι φοιτητές μόνο της φωνητικής.

```
[Professor](Video-Audio)(!)

type = friend ; (1)
context = phones ; (2)
host = dynamic ; (3)

[Staff](Video-Audio)(!)

type = friend ; (1)
context = phones ; (2)
host = dynamic; (3)

[Student](Audio)(!)

type = friend ; (1)
context = phones ; (2)
host = dynamic; (3)
```

Επεξηγήσεις:

(1): Όταν ένας χρήστης δηλωθεί ως “friend”, τότε μπορεί να έχει πρόσβαση σε όλες τις κλήσεις, χωρίς καμία εξαίρεση.

(2): Το context καθορίζει ποιες υπηρεσίες θα παρέχει το Asterisk στον εκάστοτε χρήστη. Οι υπηρεσίες περιγράφονται με την μορφή κώδικα στο αρχείο extensions.conf, το οποίο θα αναλύσουμε αργότερα.

(3): Αυτή η ρύθμιση λέει στο Asterisk ότι ο χρήστης δεν έχει σταθερή διεύθυνση διαδικτύου (IP address) και πρέπει να την δηλώνει κατά την είσοδό του.

Χρήστες.

```
;
;Users

[ProfessorLevel](Professor)
callerid = Professor Test Account ; (1)
md5secret =14a01a3a233e958415e7ee5f3bb0a0c9 ; (2)

[StaffLevel](Staff)
callerid = Staff Test Account ; (1)
md5secret =1dbebaf2219d7e37c602865f4e469bbd ; (2)

[StudentLevel](Student)
callerid = Student Test Account ; (1)
md5secret =184f6ea8dcbb74ab2174e305cfd2f226 ; (2)
```

Επεξηγήσεις:

(1): Το callerid δηλώνει τι να εμφανίζεται στην οθόνη του καλούντος.

(2): Το md5secret περιέχει το κρυπτογράφημα του συνθηματικού (password).

Βοήθημα:

Για να παράγουμε συνθηματικά με τον αλγόριθμο md5sum, χρησιμοποιούμε την εντολή:

```
echo -n "<Username>:<asterisk>:<Password>" | md5sum
```

όπου το Username αποτελεί το όνομα χρήστη και το Password το συνθηματικό.

Τέλος, αποθηκεύουμε το αρχείο (ctrl+o, enter) και αποχωρούμε από το περιβάλλον επεξεργασίας (ctrl+x).

Η εντολή:

```
asterisk -x “sip reload”
```

Ενημερώνει το Asterisk ότι έχουν γίνει αλλαγές στο sip.conf και αυτό το ελέγχει για τυχόν λάθη.

Το αρχείο sip.conf πρέπει να έχει την παρακάτω μορφή:

```
;!
;! Filename: sip.conf (/etc/asterisk/sip.conf)
;! Creator: Bill Charlaftis
;!

[general]

;
;Network

transport = udp  ; Set the default transports.  The order determines the primary default transport.
udpbindaddr = 0.0.0.0 ; IP address to bind UDP listen socket to (0.0.0.0 binds to all)
bindport = 5080
tcpenable = no  ; Enable server for incoming TCP connections (default is no)

;
;Security

context = default  ; Default context for incoming calls. Defaults to 'default',before default was public
subscribecontext = default
allowoverlap = no  ; Disable overlap dialing support. (Default is yes)
alwaysauthreject=yes;  sipvicious protection
allowguest = no ; security support
directmedia = no ; do not try to reinvite peers


;
; Fetures options

qualify = yes
videosupport=yes
accept_outofcall_message = yes
outofcall_message_context = message
auth_message_requests=yes
dtmfmode = info
```

```
[authentication]


;
;Tamplates

[Audio](!)

disallow = all
allow = ulaw,alaw

[Video-Audio](Audio)(!)

;Video
allow = h264
allow = h263
allow = vp8

[Professor](Video-Audio)(!)

type = friend
context = phones
host = dynamic

[Staff](Video-Audio)(!)

type = friend
context = phones
host = dynamic

[Student](Audio)(!)

type = friend
context = phones
host = dynamic

;
```

```
;Users

[ProfessorLevel](Professor)

callerid = Professor Test Account
md5secret =14a01a3a233e958415e7ee5f3bb0a0c9

[StaffLevel](Staff)

callerid = Staff Test Account
md5secret =1dbebaf2219d7e37c602865f4e469bbd

[StudentLevel](Student)

callerid = Student Test Account
md5secret =184f6ea8dcbb74ab2174e305cfd2f226
```

Πηγές:

1. https://el.wikipedia.org/wiki/UDP
2. https://www.voip-info.org/wiki/view/Asterisk+sip+dtmfmode

Εισαγωγή στις δομές του extensions.conf.

Στο αρχείο sip.conf αναφερθήκαμε στα context (πλαίσια) [phones] και [message] του αρχείου extensions.conf. Αυτά τα context (πλαίσια) περιέχουν δέσμες εντολών, που περιγράφουν στο Asterisk πώς να συμπεριφερθεί όταν ο χρήστης καλέσει έναν αριθμό ή μια σειρά χαρακτήρων (π.χ. καλεί τον "Γιώργο"). Ο τρόπος αντίδρασης κατά την κλήση, περιγράφεται με μια γλώσσα προγραμματισμού που λέγεται Dialplan.

Πληροφορίες σχετικά με τη γλώσσα προγραμματισμού Dialplan.

Η Dialplan είναι μια γλώσσα δέσμης ενεργειών (script programming language), ειδικά σχεδιασμένη για το Asterisk. Επιτρέπει την σύνδεση όλων των δυνατοτήτων του, παρέχοντας έτσι την δυνατότητα δρομολόγησης και χειρισμού των κλήσεων με προγραμματιστικό τρόπο.

Έχει πολλές ομοιότητες με την γλώσσα προγραμματισμού BASIC καθώς και με την ASSEMBLY.

Η Dialplan είναι πραγματικά η καρδιά κάθε συστήματος Asterisk. Ορίζει τον τρόπο με τον οποίο το Asterisk χειρίζεται εισερχόμενες και εξερχόμενες κλήσεις. Τα πάντα, από το φωνητικό ταχυδρομείο, τη διάσκεψη, έως και τα φωνητικά μενού, γίνονται μέσω της Dialplan.

Η Dialplan είναι μια γλώσσα η οποία έχει ρουτίνες, οι οποίες ενεργοποιούνται από γεγονότα τα οποία προκαλούνται από τον χρήστη. Τέτοια γεγονότα, συνήθως, είναι ένας συγκεκριμένος αριθμός ή μία σειρά χαρακτήρων που πληκτρολογήθηκαν από τον χρήστη πριν την πραγματοποίηση μιας κλήσης. Όταν αντιληφθεί ένα τέτοιο γεγονός, το Asterisk ανατρέχει στο κατάλληλο context και ψάχνει να βρει την μέγιστη δυνατή ομοιότητα του γεγονότος με τη συνθήκη που έχει περιγραφεί με τη Dialplan.

Σύνταξη της Dialplan.

Θα ξεκινήσουμε με ένα πολύ απλό παράδειγμα. Θα δώσουμε εντολή στο Asterisk να απαντήσει σε μια κλήση, να παίξει ένα αρχείο ήχου και στη συνέχεια να κλείσει την γραμμή (Παράδειγμα 1.3.1).

Παράδειγμα 1.3.1:
exten => 123,1, Answer() exten => 123,2, playback(hello-world) exten => 123,3,hangup()

Η ρουτίνα του παραπάνω παραδείγματος ενεργοποιείται από τον χρήστη, όταν αυτός καλέσει το 123.

Κάθε γραμμή της Dailplan ξεκινάει με το μοτίβο “exten =>”, που ουσιαστικά περιγράφει ότι ακολουθεί μια τριπλέτα που αποτελείται από μια σειρά χαρακτήρων ή αριθμών που κάλεσε ο χρήστης (extension), την προτεραιότητα εκτέλεσης της γραμμής (priority) και τέλος, την εντολή (application).

exten=> extension, priority, application

Κάθε ρουτίνα πρέπει να ξεκινάει με το μοτίβο της γραμμής “exten =>”, αλλά επειδή είναι κουραστικό και άσκοπο κάθε φορά να γράφουμε το ίδιο exten, extension και σε κάθε γραμμή το ίδιο priority αυξημένο κατά μια μονάδα, υπάρχει και το μοτίβο :

same=> n, application

το οποίο αυξάνει αυτόματα το priority διατηρώντας ανέπαφα τα exten, extension με αυτά της πρώτης γραμμής της ρουτίνας.

Παράδειγμα 1.3.2:
exten => 123,1, Answer() same => n, playback(hello-world) same => n,hangup()

Το παράδειγμα 1.3.2 αποτελεί την εξέλιξη του παραδείγματος 1.3.1 με την προσθήκη του μοτίβου “same => n,”.

Το σύνολο των application είναι προσβάσιμο με την εντολή:

```
asterisk -x “core show applications”
```

Πληροφορίες σχετικά με κάποιο συγκεκριμένο application μπορούν να αναζητηθούν με:

```
asterisk -x “core show application applicationName”
```

όπου applicationName το όνομα του application για το οποίο αναζητούμε πληροφορίες.

Μεταβλητές

Οι μεταβλητές μπορούν να χρησιμοποιηθούν για να προστεθεί σαφήνεια στον κώδικα.

Υπάρχουν δύο είδη μεταβλητών στην αρχιτεκτονική της Dialplan:

Οι global variable (καθολικές μεταβλητές), που είναι προσβάσιμες από όλο το αρχείο ανεξάρτητα από ποια ρουτίνα γίνει η κλήση.

Οι Global variable πρέπει να ορίζονται στο context [globals], στην αρχή του αρχείου extensions.conf. Μπορούν βέβαια να οριστούν και προγραμματιστικά με την εντολή SetGlobalVar αλλά θεωρείται κακή πρακτική (Παράδειγμα 1.3.3).

SetGlobalVar(VarName = VarPrice)

όπου το VarName είναι το όνομα της μεταβλητής και το VarPrice η τιμή που θέλουμε να της αναθέσουμε.

Παράδειγμα 1.3.3:
[globals] JOHN=345 [phones] exten => ${JOHN}, Answer() same => n, playback(hello-world) same => n,hangup()
[phones] exten => 123, 1, SetGlobalVar(JOHN=345) exten => ${JOHN}, Answer() same => n, playback(hello-world) same => n,hangup()

Προσοχή:

Στη Dialplan, η ανάθεση τιμής σε μία μεταβλητή γίνεται απλά με το μοτίβο VarName = VarPrice. Για να ανακτήσουμε, όμως, την τιμή της μεταβλητής, πρέπει να χρησιμοποιήσουμε το μοτίβο ${VarName}.

Οι channel variable (μεταβλητές καναλιού), είναι οι μεταβλητές που υφίστανται μόνο στην κλήση που “ξύπνησε” την ρουτίνα. Σε αντίθεση με τις global, οι channel variable έχουν διάρκεια ζωής όση και η διάρκεια κλήσης.

Οι channel variable μπορούν να οριστούν με την εντολή Set (Παράδειγμα 1.3.4).

Set(VarName = VarPrice)

Παράδειγμα 1.3.4:

```
[phones]

exten => 123, Answer()
same => n, Set(Hello = Hello World)
same => n, NoOp(${Hello}) ; τυπώνει Hello World
same => n,hangup()
```

Προσοχή:

Υπάρχουν πολλές προκαθορισμένες μεταβλητές καναλιού, που είναι διαθέσιμες για χρήση στην Dialplan. Τις τιμές αυτών τις καθορίζει το Asterisk και έχουμε δικαίωμα μόνο ανάγνωσης. Τέτοιου είδους μεταβλητή μπορεί να είναι η THISDEVSTATE που μας δίνει πρόσβαση στην κατάσταση ενός χρήστη. Το σύνολο των μεταβλητών που είναι προκαθορισμένες μπορεί να βρεθεί στο αρχείο README.variables, στον κατάλογο (directory) doc του φακέλου που βρίσκεται ο πηγαίος κώδικας .

Ροή του κώδικα στην Dialplan.

Η ροή του κώδικα ξεκινάει από την αρχή της ρουτίνας που αφυπνίστηκε από τον χρήστη και εκτελεί συριακά όλες τις γραμμές, μέχρι να τελειώσει η ρουτίνα (αλλάξει δηλαδή το extension) ή συναντηθεί η εντολή hangup(), η οποία τερματίζει την κλήση. Είναι δυνατόν βέβαια να αλλάξουμε αυτή την συριακή εκτέλεση με την χρήση των εντολών **Goto**, **GotoIf**.

Η **Goto** είναι εμπνευσμένη από την γλώσσα προγραμματισμού Assembly. Επιτρέπει την πραγματοποίηση αλμάτων από ένα σημείο του κώδικα σε κάποιο άλλο. Το πού θα γίνει το άλμα, μπορεί να καθοριστεί με την τριπλέτα context, priority, application με τα context, priority να είναι προαιρετικά.

Goto([context,] [extension,] priority)

Οπότε, οι δυνατές μορφές που μπορούμε να συναντήσουμε την Goto είναι:

Goto(context,extension,priority)

Goto(extension,priority)

Goto(priority)

Παράδειγμα 1.3.5:
[exp1] exten => 123, 1, Answer() same => n, Goto(exp2, 456, 1) [exp2] exten => 456, 1, playback(hello-world) same => n,Goto(goodBye,1) exten => goodBye, 1, playback(goodbye) exten => goodBye, 2, Goto(4) exten => goodBye, 3, NoOp(This line never printed) exten => goodBye, 4, hangup()

Το Παράδειγμα 1.3.5 παράγει το ίδιο αποτέλεσμα με το Παράδειγμα 1.3.1, άλλα δεν εκτελείται συριακά. Το Asterisk απαντά στην κλήση, κατόπιν “μετακινείται” στο context --> exp2, extension --> 456, priority -->1, παίζει ένα αρχείο ήχου, “μετακινείται” στο extension --> goodBye, priority -->1 του ίδιου context και τέλος κάνει το τελευταίο “άλμα“ στο priority -->4 του ίδιου context και extension.

Η **GotoIf** ουσιαστικά εκτελεί την ίδια λειτουργία με την Goto, με μόνη διαφορά ότι το αν θα γίνει το άλμα εξαρτάται από μία συνθήκη. Η σύνταξη της εντολής είναι η εξής:

GotoIf($[condition]?[context,] [extension,] priority)

Σε αντιστοιχία με την Goto, οι δυνατές μορφές που μπορούμε να συναντήσουμε την GotoIf είναι:

Goto($[condition]?context,extension,priority)

Goto($[condition]?extension,priority)

Goto($[condition]?priority)

Παράδειγμα 1.3.6:
[exp1] exten => 123, 1, Answer() same => n, Set(x = 4) same => n, GotoIf($[${x}= ${2+2}]?exp2, 456, 1) [exp2] exten => 456, 1, playback(hello-world) same => n,GotoIf($[${x}= ${2*2}]?goodBye,1) exten => goodBye, 1, playback(goodbye) exten => goodBye, 2, GotoIf($[${x}= ${1*2}]?4) exten => goodBye, 3, NoOp(This line printed because 1*2 is not 4) exten => goodBye, 4, hangup()

Το Παράδειγμα 1.3.6 είναι πανομοιότυπο με το Παράδειγμα 1.3.5. Η μόνη διαφορά είναι ότι, πριν εκτελεστεί το “άλμα”, πρώτα ελέγχεται η ορθότητα της συνθήκης. Αν η συνθήκη είναι αληθής, τότε πραγματοποιείται το άλμα. Σε αντίθετη περίπτωση, συνεχίζει η εκτέλεση του κώδικα συριακά.

Ετικέτες – label

Πολλές φορές, επειδή το priority, μέχρι να φτάσουμε στην τελική μορφή του κώδικα, μπορεί να αυξομειώνεται χωρίς να γίνεται αντιληπτό (ειδικά αν χρησιμοποιούμε το μοτίβο “same => n,”), η χρήση των Goto, GotoIf γίνεται δύσκολη.
Προκειμένου να ξεπεραστεί αυτό το πρόβλημα, δημιουργήθηκε η έννοια της ετικέτας (label). Η **label** αποτελεί μια ονομασία που δίνουμε σε μια γραμμή της Dialplan. Για να δώσουμε μια ονομασία σε μια γραμμή, αρκεί να γράψουμε μετά το priority και πριν το κόμμα το όνομα που επιλέξαμε μέσα σε παρενθέσεις.

exten=> extension, priority (label), application ή same=> n (label), application

Το όνομα του label μπορεί να χρησιμοποιηθεί αντί για το priority στις εντολές Goto, GotoIf.

Β. Επεξεργασία του extensions.conf.

Πληκτρολογούμε την εντολή:

```
pico /etc/asterisk/extensions.conf
```

για να επεξεργαστούμε το αρχείο.

Όπως και στην περίπτωση του sip.conf, υπάρχει ήδη μια βασική δομή η οποία όμως αυτή την φορά μας είναι αδιάφορη, καθώς έχει πολύ μικρή ομοιότητα με την τελική δομή που θέλουμε να έχει το αρχείο μας.

Βοήθημα:
Για να σβήσουμε όλες τις γραμμές από το αρχείο, πατάμε ctrl+shift+6 (κάτω από τα F), κατεβαίνουμε με την ροδέλα μέχρι το τέλος του αρχείου και πατάμε τον συνδυασμό ctrl+k.

Το extensions.conf πρέπει να έχει τέσσερα context :

1. Το phones, το οποίο περιγράφει τις διαδικασίες της φωνητικής και της βίντεο κλήσης.
2. Το message, το οποίο φροντίζει να παραδοθούν τα μηνύματα ακόμα και αν ο παραλήπτης είναι εκτός σύνδεσης.
3. Το app-fakeanswer, το οποίο είναι επικουρικό context του message.
4. Το default, το οποίο είναι κενό και αποτελεί δικλίδα ασφαλείας.

Context [phones]

Το phones context περιέχει πέντε extensions και, άρα, πέντε είδη ρουτίνας:

1. Extension _X.

Η ρουτίνα του extension _X. μπορεί να αφυπνιστεί από οποιονδήποτε αριθμό καλέσει ο χρήστης.

Επειδή, για λόγους ασφαλείας, οι χρήστες δεν έχουν user-names τα οποία να αποτελούνται αποκλειστικά από χαρακτήρες, όπως θα δούμε και στο τελευταίο κεφάλαιο (Σελ. 98), έχει χρησιμοποιηθεί η βάση δεδομένων του Asterisk για την αντιστοίχηση των user-names σε μοναδικούς αριθμούς κλήσης. Ο λόγος που έχει γίνει αυτό, είναι ότι οι φυσικές τηλεφωνικές γραμμές σπάνια έχουν την δυνατότητα εισαγωγής χαρακτήρων.

Πληροφορίες σχετικά με το pattern matching .

Επειδή θα ήταν ιδιαίτερα δύσκολο να προσθέσουμε κάθε πιθανό extension στη Dialplan, το Asterisk προσφέρει την δυνατότητα αντιστοίχισης μοτίβου (pattern matching). Το pattern matching μας επιτρέπει την αντιστοίχηση πολλών extensions σε μία ρουτίνα.

Όταν χρησιμοποιούμε pattern matching, αρχίζουμε πάντα με μια κάτω παύλα [_]. Αυτό λέει στο Asterisk ότι ακολουθεί ένα μοτίβο και όχι ένα όνομα επέκτασης. Μετά την κάτω

παύλα, χρησιμοποιούμε έναν ή περισσότερους από τους παρακάτω χαρακτήρες:

- Το X αναπαριστά οποιοδήποτε ψηφίο από 0 έως 9.
- Το Z αναπαριστά οποιοδήποτε ψηφίο από το 1 έως το 9.
- N Αναπαριστά οποιοδήποτε ψηφίο από 2 έως 9.
- [Ascii_Entry0 - Ascii_EntryN] αναπαριστά οποιοδήποτε εύρος χαρακτήρων του Ascii. Π.χ. το [A-z] παριστάνει όλους τους πεζούς και κεφαλαίους χαρακτήρες της αγγλικής γλώσσας.
- Το . επισημαίνει ότι το προηγούμενο patern matching μπορεί να επαναληφθεί πολλές φορές. Π.χ. το X. παριστάνει όλους τους ακέραιους.

Έτσι, αυτή η ρουτίνα αναλαμβάνει να αντιστοιχήσει τους αριθμούς σε χρήστες, ώστε να είναι δυνατή η δυνατότητα κλήσης – βίντεο κλήσης από το επόμενο extension.

```
;
; Convert Numbers to users
;
; Students: dhome _1X., dgeo _2X., ddiet _3X., dit _4X.
;
exten => _X.,1,NoOp(converting user ${EXTEN} to valid exten )
same => n,GoTo(phones,${DB(exten/${EXTEN})},1)
same => n,hangup()
```

2. Extension _[A-z].

Η ρουτίνα του extension _[A-z]. μπορεί να αφυπνιστεί από οποιονδήποτε αλφαριθμητικό συνδυασμό καλέσει ο χρήστης (ή από το extension _X.) και είναι υπεύθυνο για πραγματοποίηση της κλήσης - βίντεο κλήσης.

```
;
; Universal call
;
exten => _[A-z].,1,NoOp(call to ${EXTEN} from ${CALLERIDNAME})
same => n,Dial(SIP/${EXTEN})
osame => n,Hangup()
```

3. Extension 002

Η ρουτίνα αυτή ενεργοποιείται με την κλήση του αριθμού 002. Η λειτουργία που προσφέρει αυτό το extension, αποτελεί επίδειξη της δυνατότητας που έχει το Asterisk να διαβάζει αριθμούς με γυναικεία, αντρική ή ουδέτερη μηχανική φωνή.

```
;
; Exten reading with three voices
;
exten => 002,1,Answer()
same => n,Playback(hello)
same => n,SayNumber(${EXTEN},f)
same => n,SayNumber(${EXTEN},m)
same => n,SayNumber(${EXTEN},c)
same => n,Playback(vm-goodbye)
same => n,Hangup()
```

4. Extension 003.

Όταν κληθεί ο αριθμός αυτός, τότε το Asterisk, με γυναικεία φωνή, καλωσορίζει και ενημερώνει για την ώρα τον χρήστη στα αγγλικά.

```
;
; Time Informations
;
exten => 003,1,Answer()
same => n,Playback(hello)
same => n,Playback(time)
same => n,Playback(is)
same => n,SayUnixTime(,localtime,R )
same => n,Hangup()
```

5. Extension 004.

Το extension αυτό είναι ανενεργό, μέχρι την κλήση του αριθμού 004. ‘Οταν ενεργοποιείται, συνθέτει το “Imperial Order” από τις ταινίες “Star Wars” και το μεταδίδει στην κλήση.

```
;
; Imperial Order
;
exten => 004, 1, Answer()
same => n,Wait(1)
same => n,Playtones(!440/500,!440/500,!440/500,!349/350,!523/150,!440/500,!349/350,!523/150,!440/650)
same => n,Wait(4.5)
same => n,Playtones(!659/500,!659/500,!659/500,!698/350,!523/150,!415/500,!349/350,!523/150,!440/650)
same => n,Wait(4.5)
same => n,Playtones(!880/500,!440/300,!440/150,!880/500,!830/325,!784/175,!740/125,!698/125,!740/250)
same => n,Wait(2.75)
same => n,Playtones(!455/250,!622/500,!587/325,!554/175,!523/125,!466/125,!523/250)
same => n,Wait(3.0)
same => n,Playtones(!349/250,!415/500,!349/350,!440/125,!523/500,!440/375,!523/125,!659/650)
same => n,Wait(4.5)
same => n,Playtones(!880/500,!440/300,!440/150,!880/500,!830/325,!784/175,!740/125,!698/125,!740/250)
```

```
same => n,Wait(2.75)
same => n,Playtones(!455/250,!622/500,!587/325,!554/175,!523/125,!466/125,!523/250)
same => n,Wait(3.0)
same => n,Playtones(!349/250,!415/500,!349/375,!523/125,!440/500,!349/375,!523/125,!587,325,!554,175,!
523,125,!466,125,!523,250)
same => n,Wait(3.0)
same => n,StopPlaytones
same => n,Hangup
```

Προσοχή:

Το Asterisk πάντα προσπαθεί να κάνει την ακριβέστερη αντιστοίχηση των extension. Έτσι, όταν κληθεί το 002, 003 ή το 004, θα τα αντιστοιχίσει στις ομώνυμες ρουτίνες και όχι στην _X.

Context [message]

Το context message είναι ουσιαστικά ο διαχειριστής της λειτουργίας ανταλλαγής μηνυμάτων. Έχει μόνο δύο extensions:

1. Extension _[A-z].

Το extension _[A-z] φροντίζει την παράδοση του μηνύματος, ακόμα και αν χρησιμοποιηθεί ο αριθμός αντί για το user-name. Σε περίπτωση που δεν μπορεί να γίνει αντιστοίχηση, στέλνει μήνυμα λάθους στον αποστολέα.

```
;
; Alpharithmetic to numeric convertor
;
exten => _[A-z].,1,NoOp(converting user ${MESSAGE(to)} to valid exten )
same => n,GotoIf($[${REGEX("[@]" ${MESSAGE(to)})} == 0]?ItWasClear)
same => n,Set(to=${CUT(MESSAGE(to),@,1)})
same => n,Set(EXTENTO=${to:4})
same => n,Set(reiciver=${DB(exten/${EXTENTO})})
same => n,GotoIf($["${reiciver}" == ""]?invalidUser)
```

```
same => n,Goto(message,${DB(exten/${EXTENTO})},1)
same => n,Hangup()
same => n(ItWasClear),NoOp(reiciver=DB(exten/${MESSAGE(to)}))
same => n,Set(reiciver=${DB(exten/${MESSAGE(to)})})
same => n,GotoIf($["${reiciver}" == ""]?invalidUser)
same => n,Goto(message,${DB(exten/${MESSAGE(to)})},1)
same => n,Hangup()

;
; Invalid User
same => n(invalidUser),Set(MESSAGE(body)=Ο χρήστης που επιθημήτε να στείλετε μήνημα είναι μη υπαρκτός.)
same => n,Set(from=${CUT(MESSAGE(from),@,1)})
same => n,Set(SRC=${from:5})
same => n,MessageSend(sip:${SRC},Support Center)
same => n,Hangup()
```

2. Extension _X.

Το extension αυτό φροντίζει την παραλαβή του μηνύματός ακόμα και στην περίπτωση που ο δέκτης είναι εκτός σύνδεσης.

Για την παραλαβή των μηνυμάτων που εστάλησαν όταν ο χρήστης ήταν εκτός σύνδεσης, δημιουργείται ένα αρχείο .call που θα «τρέχει» ανά τακτά χρονικά διαστήματα, έως ότου ο χρήστης επανασυνδεθεί.

Πληροφορίες σχετικά με τα αρχεία .call .

Τα .call αρχεία τα οποία, όταν μετακινούνται στον κατάλογο (directory) /var/spool/asterisk/outgoing/, είναι σε θέση να πραγματοποιούν αυτόματα κλήσεις. Τα αρχεία κλήσεων είναι ένας πολύ καλός τρόπος να πραγματοποιούν κλήσεις αυτόματα χωρίς να χρησιμοποιούν πιο σύνθετες λειτουργίες απαιτώντας ταυτόχρονα πολύ λίγες τεχνικές γνώσεις.

```
;
; Message handler
exten => _X.,1,Set(DST=${DB(exten/${EXTEN})})
same => n,Set(from=${CUT(MESSAGE(from),@,1)})
same => n,Set(SRC=${from:5})
same => n,Set(MSG=${MESSAGE(body)})
same => n,MessageSend(sip:${DST},${SRC})
same => n,GotoIf($["${MESSAGE_SEND_STATUS}" != "SUCCESS"]?sendfailedmsg)
same => n,Hangup()
;
; Handle failed messaging
same => n(sendfailedmsg),GotoIf($["${DB_EXISTS(message_${SRC}_to_${DST}/Messages)}" != "1"]?
informOffline);try only once
same => n,Set(DB(message_${SRC}_to_${DST}/Messages)=${DB(message_${SRC}_to_${DST}/Messages)}$
{STRFTIME(${EPOCH},,Ημέρα: %d-%m-%Y ώρα: %H:%M:%S)}:${MSG}<br>)
same => n,Hangup()
;
; Inform offline
same => n(informOffline),NoOp(Sending error to user)
same => n,Set(MESSAGE(body)="Κέντρο Υποστήριξης: Ο χρήστης ${DST} (${EXTEN}) είναι εκτός σύνδεσης θα
ενημερωθεί για το μήνυμα σας όταν θα είναι διαθέσιμος.")
same => n,MessageSend(sip:${SRC},${DST});αναποδα για απαντηση
same => n,GoTo(message,${EXTEN},offlineMessage)
same => n,Hangup()
;
; Queueing messaging for offline
same => n(offlineMessage),Set(DB(message_${SRC}_to_${DST}/Messages)=${STRFTIME(${EPOCH},,Ημέρα: %d-
%m-%Y ώρα: %H:%M:%S)}:${MSG})
same => n,Set(Path2Created=/var/spool/asterisk/temp/${STRFTIME(${EPOCH},,%Y-%m-%d-%H:%M:%S)}.call)
same => n,SYSTEM(echo "Channel:  Local/${DST}@app-fakeanswer\nMaxRetries: 1440\nRetryTime: 15\nContext:
message\nExtension:${DST}\nPriority: 1\nSet: MessageBody= message_${SRC}_to_${DST}/Messages\nSet:
MESSAGE(body)= \nSet: MESSAGE(to)=${DST}\nSet: MESSAGE(from)=${from}\nSet: dataBaseEntry= message_$
{SRC}_to_${DST} Messages" >> ${Path2Created}) ;
same => n,SYSTEM(chown asterisk:asterisk ${Path2Created})
same => n,SYSTEM(chmod +777 ${Path2Created})
same => n,SYSTEM(sleep 3)
same => n,SYSTEM(cp ${Path2Created} ${Path2Created}.bu )
same => n,SYSTEM(mv ${Path2Created} /var/spool/asterisk/outgoing/);
same => n,Hangup()
```

Context [app-fakeanswer]

Τέλος, το extension app-fakeanswer δημιουργεί τα πλαίσια μιας εικονικής κλίσης έτσι ώστε να παραδοθεί το μήνυμα στα πλαίσια αυτού του καναλιού.

```
[app-fakeanswer]

exten => _[A-z].,1,NoCDR
same => n,Set(DESTDEV=${EXTEN})
same => n,Set(THISDEVSTATE=${DEVICE_STATE(SIP/${DESTDEV})})
same => n,GotoIf($["${THISDEVSTATE}" = "UNAVAILABLE"]?hang)
same => n,GotoIf($["${THISDEVSTATE}" = "UNKNOWN"]?hang)
same => n,GotoIf($["${THISDEVSTATE}" = "INVALID"]?hang)
same => n,NoOp(db ${MessageBody} --> ${DB(message_Giannis_to_Tilemachos/Messages)})
same => n,Set(MESSAGE(body)=${DB(message_Giannis_to_Tilemachos/Messages)})
same => n,NoOp(Attempt to delete db entry ${dataBaseEntry})
same => n,Set(result=${SHELL(asterisk -x'database del ${dataBaseEntry}')});
same => n,Answer
same => n,Hangup()
same => n(hang),Hangup()
```

Η ολοκληρωμένη μορφή του αρχείου extensions.conf είναι :

```
[incoming]
exten => _2109549X.,1,NoOp(${EXTEN:-3})
same => n,Goto(phones,${EXTEN:-3},1)

[phones]
;
; Convert Numbers to users
; Students: dhome _1X., dgeo _2X., ddiet _3X., dit _4X.
;
exten => _X.,1,NoOp(converting user ${EXTEN} to valid exten )
same => n,GoTo(phones,${DB(exten/${EXTEN})},1)
same => n,Hangup
```

```
;
; Universal call
;
exten => _[A-z].,1,NoOp(call to ${EXTEN} from ${CALLERIDNAME})
same => n,Dial(SIP/${EXTEN})
same => n,Hangup()
;
; Exten reading with three voices
;
exten => 002,1,Answer()
same => n,Playback(hello)
same => n,SayNumber(${EXTEN},f)
same => n,SayNumber(${EXTEN},m)
same => n,SayNumber(${EXTEN},c)
same => n,Hangup()

;
; Time Informations
;
exten => 003,1,Answer()
same => n,Playback(hello)
same => n,Playback(time)
same => n,Playback(is)
same => n,SayUnixTime(,localtime,R )
same => n,Hangup()

;
; Imperial Order
;
exten => 004, 1, Answer()
same => n,Wait(1)
same => n,Playtones(!440/500,!440/500,!440/500,!349/350,!523/150,!440/500,!349/350,!523/150,!440/650)
same => n,Wait(4.5)
same => n,Playtones(!659/500,!659/500,!659/500,!698/350,!523/150,!415/500,!349/350,!523/150,!440/650)
same => n,Wait(4.5)
same => n,Playtones(!880/500,!440/300,!440/150,!880/500,!830/325,!784/175,!740/125,!698/125,!740/250)
same => n,Wait(2.75)
same => n,Playtones(!455/250,!622/500,!587/325,!554/175,!523/125,!466/125,!523/250)
```

```
same => n,Wait(3.0)
same => n,Playtones(!349/250,!415/500,!349/350,!440/125,!523/500,!440/375,!523/125,!659/650)
same => n,Wait(4.5)
same => n,Playtones(!880/500,!440/300,!440/150,!880/500,!830/325,!784/175,!740/125,!698/125,!740/250)
same => n,Wait(2.75)
same => n,Playtones(!455/250,!622/500,!587/325,!554/175,!523/125,!466/125,!523/250)
same => n,Wait(3.0)
same => n,Playtones(!349/250,!415/500,!349/375,!523/125,!440/500,!349/375,!523/125,!587,325,!554,175,!
523,125,!466,125,!523,250)
same => n,Wait(3.0)
same => n,StopPlaytones
same => n,Hangup

exten => _9X.,1,NoOp(${EXTEN:1})
same => n,Goto(outgoing,${EXTEN:1},1)

[outgoing]

exten => _X.,1,Dial(SIP/outside)

[message]

;
; Alpharithmetic to numeric convertor
;
exten => _[A-z].,1,NoOp(converting user ${MESSAGE(to)} to valid exten )
same => n,GotoIf($[${REGEX("[@]" ${MESSAGE(to)})} == 0]?ItWasClear)
same => n,Set(to=${CUT(MESSAGE(to),@,1)})
same => n,Set(EXTENTO=${to:4})
same => n,Set(reiciver=${DB(exten/${EXTENTO})})
same => n,GotoIf($["${reiciver}" == ""]?invalidUser)
same => n,Goto(message,${DB(exten/${EXTENTO})},1)
same => n,Hangup()
same => n(ItWasClear),NoOp(reiciver=DB(exten/${MESSAGE(to)}))
same => n,Set(reiciver=${DB(exten/${MESSAGE(to)})})
same => n,GotoIf($["${reiciver}" == ""]?invalidUser)
same => n,Goto(message,${DB(exten/${MESSAGE(to)})},1)
same => n,Hangup()
```

```
;
; Invalid User
same => n(invalidUser),Set(MESSAGE(body)=Ο χρήστης που επιθημήτε να στείλετε μήνημα είναι μη υπαρκτός.)
same => n,Set(from=${CUT(MESSAGE(from),@,1)})
same => n,Set(SRC=${from:5})
same => n,MessageSend(sip:${SRC},Support Center)
same => n,Hangup()

;
; Message handler
exten => _X.,1,Set(DST=${DB(exten/${EXTEN})})
same => n,Set(from=${CUT(MESSAGE(from),@,1)})
same => n,Set(SRC=${from:5})
same => n,Set(MSG=${MESSAGE(body)})
same => n,MessageSend(sip:${DST},${SRC})
same => n,GotoIf($["${MESSAGE_SEND_STATUS}" != "SUCCESS"]?sendfailedmsg)
same => n,Hangup()

;
; Handle failed messaging
same => n(sendfailedmsg),GotoIf($["${DB_EXISTS(message_${SRC}_to_${DST}/Messages)}" != "1"]?
informOffline);try only once
same => n,Set(DB(message_${SRC}_to_${DST}/Messages)=${DB(message_${SRC}_to_${DST}/Messages)}$
{STRFTIME(${EPOCH},,Ημέρα: %d-%m-%Y ώρα: %H:%M:%S)}:${MSG}<br>)
same => n,Hangup()

;
; Inform offline
same => n(informOffline),NoOp(Sending error to user)
same => n,Set(MESSAGE(body)="Κέντρο Υποστήριξης: Ο χρήστης ${DST} (${EXTEN}) είναι εκτός σύνδεσης θα
ενημερωθεί για το μήνυμα σας όταν θα είναι διαθέσιμος.")
same => n,MessageSend(sip:${SRC},${DST});αναποδα για απαντηση
same => n,GoTo(message,${EXTEN},offlineMessage)
same => n,Hangup()

;
; Queueing messaging for offline
same => n(offlineMessage),Set(DB(message_${SRC}_to_${DST}/Messages)=${STRFTIME(${EPOCH},,Ημέρα: %d-
```

```
%m-%Y ώρα: %H:%M:%S)}:${MSG})
same => n,Set(Path2Created=/var/spool/asterisk/temp/${STRFTIME(${EPOCH},,%Y-%m-%d-%H:%M:%S)}.call)
same => n,SYSTEM(echo "Channel:  Local/${DST}@app-fakeanswer\nMaxRetries: 1440\nRetryTime: 15\nContext:
message\nExtension:${DST}\nPriority: 1\nSet: MessageBody= message_${SRC}_to_${DST}/Messages\nSet:
MESSAGE(body)= \nSet: MESSAGE(to)=${DST}\nSet: MESSAGE(from)=${from}\nSet: dataBaseEntry= message_$
{SRC}_to_${DST} Messages" >> ${Path2Created}) ;
same => n,SYSTEM(chown asterisk:asterisk ${Path2Created})
same => n,SYSTEM(chmod +777 ${Path2Created})
same => n,SYSTEM(sleep 3)
same => n,SYSTEM(cp ${Path2Created} ${Path2Created}.bu )
same => n,SYSTEM(mv ${Path2Created} /var/spool/asterisk/outgoing/);
same => n,Hangup()

[app-fakeanswer]

exten => _[A-z].,1,NoCDR
same => n,Set(DESTDEV=${EXTEN})
same => n,Set(THISDEVSTATE=${DEVICE_STATE(SIP/${DESTDEV})})
same => n,GotoIf($["${THISDEVSTATE}" = "UNAVAILABLE"]?hang)
same => n,GotoIf($["${THISDEVSTATE}" = "UNKNOWN"]?hang)
same => n,GotoIf($["${THISDEVSTATE}" = "INVALID"]?hang)
same => n,NoOp(db ${MessageBody} --> ${DB(message_Giannis_to_Tilemachos/Messages)})
same => n,Set(MESSAGE(body)=${DB(message_Giannis_to_Tilemachos/Messages)})
same => n,NoOp(Attempt to delete db entry ${dataBaseEntry})
same => n,Set(result=${SHELL(asterisk -x'database del ${dataBaseEntry}')});
same => n,Answer
same => n,Hangup()
same => n(hang),Hangup()

[default]
```

ΚΕΦΑΛΑΙΟ 4

Σύνδεση στην υπηρεσία του Asterisk.

Το sip μπορεί να αποτελέσει δίοδο επικοινωνίας σε συσκευές όπως κινητά τηλέφωνα, ηλεκτρονικοί υπολογιστές και ορισμένες τηλεφωνικές συσκευές.

Στις ενότητες που ακολουθούν θα αναλύσουμε την διαδικασία δημιουργίας λογαριασμού για την υπηρεσία του Asterisk καθώς και την διαδικασία που πρέπει να ακολουθηθεί προκειμένου να συνδεθούν στον διακομιστή (server) τρία είδη συσκευών [έναν υπολογιστής και ένα Android κινητό].

4.1 Δημιουργία λογαριασμού χρήστη μέσω του προτύπου LDAP.

Ένας από τους βασικούς στόχους καθ' όλη την διάρκεια της υλοποίηση της πτυχιακής μου εργασίας ήταν το τηλεφωνικό κέντρο να είναι προσβάσιμο από οποιονδήποτε συμπεριλαμβάνεται στην υπηρεσία καταλόγου χρηστών LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol) του Πανεπιστημίου μας.

Πληροφορίες σχετικά με τον LDAP:

Το LDAP (Lightweight Directory Access Protocol) είναι ένα ανοιχτό πρωτόκολλο το οποίο επιτρέπει την πρόσβαση και συντήρηση υπηρεσιών κατανεμημένων καταλόγων μέσω δικτύου. Οι υπηρεσίες καταλόγου παίζουν σημαντικό ρόλο στην ανάπτυξη εφαρμογών εσωτερικού δικτύου (intranet) και διαδικτύου (internet) επιτρέποντας την ανταλλαγή πληροφοριών σχετικά με χρήστες, συστήματα, δίκτυα, υπηρεσίες και εφαρμογές [1.].Μια κοινή χρήση του LDAP είναι η αποθήκευση ονομάτων χρηστών, κωδικών πρόσβασης και άλλων στοιχείων που μπορούν να χρησιμοποιηθούν για την ταυτοποίηση και την είσοδο ενός χρήστη σε μια υπηρεσία.

Για λόγους ασφαλείας απορρήτου δεν ήταν δυνατή η απευθείας σύνδεση του Asterisk με τον LDAP. Η μοναδική λύση προκειμένου να συνδέεται κανείς με το τηλεφωνικό κέντρο με τα ίδια στοιχεία που συνδέεται με οποιαδήποτε άλλη υπηρεσία παρέχει το Πανεπιστήμιο, ήταν η δημιουργία λογαριασμού μετά από την συναίνεση του χρήστη.

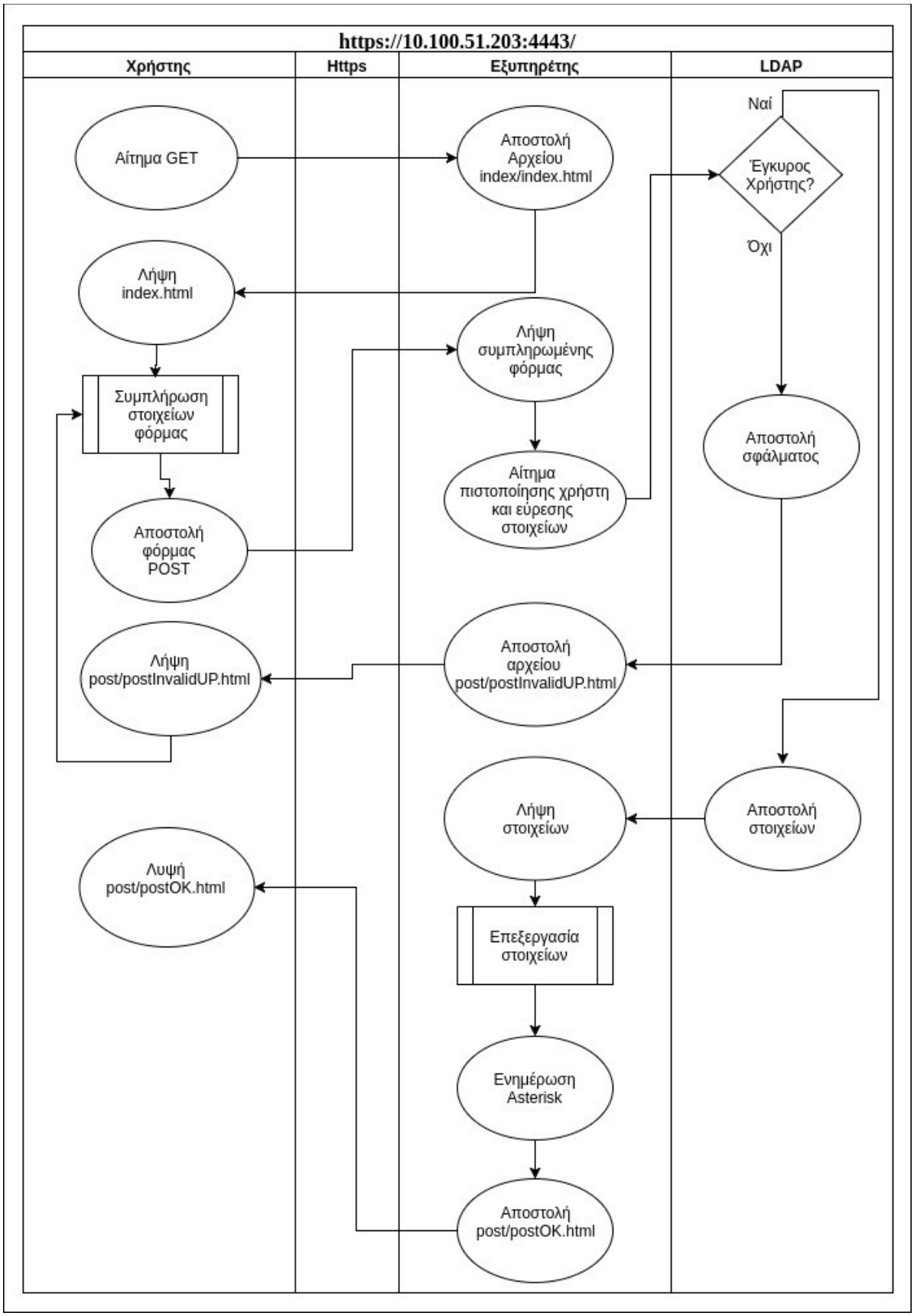
Προκειμένου να δημιουργηθεί μια δίοδος επικοινωνίας μεταξύ του κέντρου και του τελικού χρήστη, μέσα από την οποία ο τελευταίος θα μπορούσε να συναινέσει έτσι ώστε το Asterisk να αποκτήσει πρόσβαση στην υπηρεσία του LDAP, κατασκευάστηκε μία ιστοσελίδα, που λειτουργεί σαν μεσολαβητής των δύο υπηρεσιών.

4.2 Ιστοσελίδα εγραφής

Ο ιστότοπος αποτελείται από:

Α. Το αρχείο https.py που περιέχει τον κώδικα υλοποίησης του εξυπηρέτη.
Β. Τους φακέλους index και post, οι οποίοι περιέχουν τις στατικές σελίδες που αποστέλλονται στους χρήστες.
Γ. Το αρχείο server.pem που περιλαμβάνει το ιδιωτικό κλειδί κρυπτογράφησης, που μας εξασφαλίζει την ύπαρξη του ασφαλούς πρωτοκόλλου μεταφοράς υπερκειμένου (**H**yper**T**ext **T**ransfer **P**rotocol **S**ecure - HTTPS).

Το παρακάτω σχεδιάγραμμα δείχνει περιληπτικά πώς δουλεύει η σελίδα:



Σχεδιάγραμμα 4.1.1

Δημιουργία περιβάλλοντος ιστοσελίδας

Πληκτρολογούμε την εντολή:

```
mkdir ~/www
```

Το αποτέλεσμα είναι η δημιουργία του κατάλογου κάτω από τον οποίο θα ενεργεί ο ιστότοπός μας.

Για να δημιουργήσουμε το αρχείο python [κατάληξη .py] πληκτρολογούμε την εντολή:

```
touch ~/www/https.py
```

ανοίγουμε το περιβάλλον επεξεργασίας:

```
pico ~/www/https.py
```

και επικολλούμε (ctrl+shift+v) τον κώδικα python3 ο οποίος διαχειρίζεται τα αιτήματα:

```
#!/usr/bin/env python3
"""
Very simple HTTP server in python.
Usage::
    ./dummy-web-server.py [<port>]
Send a GET request::
    curl http://localhost
Send a HEAD request::
    curl -I http://localhost
Send a POST request::
    curl -d "foo=bar&bin=baz" http://localhost

generate server.pem with the following command:
#    openssl req -new -x509 -keyout server.pem -out server.pem -days 365 -nodes
"""
from http.server import BaseHTTPRequestHandler, HTTPServer
from collections import Counter
```

```
import socketserver
import shutil
import socket
import ssl
import hashlib
import cgi
import os
import sys
import urllib

form = cgi.FieldStorage() # μεταβλιτή φορμάς για post

"""

"""
class S(BaseHTTPRequestHandler):

    usr = ""
    pasw = ""
    cat = ""
    ldap=-1
    stdDepNum=""

    def _set_headers(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()

    def do_GET(self):
        self._set_headers()

        args = {}
        idx = self.path.find('?')
```

```
    if idx >= 0:
        rpath = self.path[:idx]
        args = cgi.parse_qs(self.path[idx+1:])
    else:
        rpath = self.path

    # Print out logging information about the path and args.
    if 'content-type' in self.headers:
        ctype, _ = cgi.parse_header(self.headers['content-type'])
        print('(GET) TYPE %s' % (ctype))

    print('(GET) PATH %s' % (rpath))
    print('(GET) ARGS Number %d' % (len(args)))

    file = open("index/index.html","rb+")
    print('Sending index/index.html')
    self.wfile.write(file.read())

def do_HEAD(self):
    self._set_headers()

def do_POST(self):
    # http://joelinoff.com/blog/?p=1658
    self._set_headers()

    ctype, pdict = cgi.parse_header(self.headers['content-type'])

    if ctype == 'multipart/form-data':
        postvars = cgi.parse_multipart(self.rfile, pdict)
    elif ctype == 'application/x-www-form-urlencoded':
        length = int(self.headers['content-length'])
        postvars = cgi.parse_qs(self.rfile.read(length), keep_blank_values=1)
```

```
else:
        postvars = {}

# Print out logging information about the path and args.
print('(POST) TYPE %s' % (ctype))
print('(POST) PATH %s' % (self.path))

print('(POST) %d ARGS recieved ' % (len(postvars)))

if len(postvars):

        self.usr = str(*postvars[b'usr'])[2:-1]
        self.pasw = str(*postvars[b'psw'])[2:-1]
        self.cat = str(*postvars[b'cat'])[2:-1]

#print("Userame: "+usr+" Password: "+pasw+" Category: "+cat)

if self.usr == '' :

      self.ldap = -1
else:

      self.usr = self.usr.replace('@hua.gr','')
      self.contactLDAP()  #LDAP connection and querry


print("\n   RESULT : "+str(self.ldap)+'  '+str(self.ldap == 0.0))

if self.ldap == 0.0 :

        self.ValidPUResponse()
        self.configurator() #Turn querry result to sip conf file
```

```
        else:

            self.InvalidPUResponse()


    def contactLDAP(self):

        print("Category : "+self.cat)

        if Counter(self.cat) == Counter("Dr"):

            #Kathigites
            self.ldap=os.system('ldapsearch -x -b "OU=Users,OU=DEP,DC=hua,DC=gr" -s sub
"(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w '+self.pasw+'
-D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)

        elif Counter(self.cat) == Counter("Stf"):

            #Staff
            self.ldap=os.system('ldapsearch -x -b "OU=STAFF,DC=hua,DC=gr" -s sub
"(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w '+self.pasw+'
-D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)

        elif Counter(self.cat) == Counter("Std"):

            if self.usr.startswith("it"):

                #IT students
                self.ldap=os.system('ldapsearch -x -b "OU=Users,OU=DIT,DC=hua,DC=gr" -s sub
"(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w '+self.pasw+'
-D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)
                self.stdDepNum="4"+self.usr[2:]
                print(self.stdDepNum)
```

```
        if self.usr.startswith("gs"):

            #GEO students
            self.ldap=os.system('ldapsearch -x -b "OU=Users,OU=DDIET,DC=hua,DC=gr" -s
sub "(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w
'+self.pasw+' -D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)
            self.stdDepNum="2"+self.usr[2:]
            print(self.stdDepNum)

        if self.usr.startswith("dp"):

            #DIET students
            self.ldap=os.system('ldapsearch -x -b "OU=Users,OU=DGEO,DC=hua,DC=gr" -s sub
"(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w '+self.pasw+'
-D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)
            self.stdDepNum="3"+self.usr[2:]
            print(self.stdDepNum)

        if self.usr.startswith("hp"):

            #HOME students
            self.ldap=os.system('ldapsearch -x -b "OU=Users,OU=DHOME,DC=hua,DC=gr" -s
sub "(mailNickname='+self.usr+')" mailNickname telephoneNumber displayName -w
'+self.pasw+' -D "'+self.usr+'@hua.gr" -h hercules.hua.gr >./'+self.usr)
            self.stdDepNum="1"+self.usr[2:]
            print(self.stdDepNum)

      else :
            #Invalid Category
            print("Not valid Category")
            self.InvalidPUResponse()
```

```
def InvalidPUResponse(self):

    file = open("post/postInvalidUP.html","rb+")
    self.wfile.write(file.read())
    file.close()


def ValidPUResponse(self):

    file = open("post/postOK.html","rb+")
    self.wfile.write(file.read())
    file.close()


def configurator(self):

    f = open(self.usr)
    lines = f.readlines()
    telephoneNmbr=0
    name=""
    mail=""
    video="yes"

    for i in range(0,len(lines)):
        if lines[i].find("telephoneNumber") == 0:
            telephoneNmbr=lines[i].replace("telephoneNumber: ","").replace("\n","")
        if  lines[i].find("displayName") == 0:
            name=lines[i].replace("displayName: ","").replace("\n","")
        if  lines[i].find("mailNickname") == 0:
            mail=lines[i].replace("mailNickname: ","").replace("\n","")+"@hua.gr"

    os.remove(self.usr)
```

```
        if(Counter(self.cat) == Counter("Std")):
            video="no"

        password = hashlib.md5(self.pasw.encode('utf-8')).hexdigest()
        try:



            file = open(self.usr+".conf",'w+')   # Trying to create a new file or open one

            file.write('['+self.usr+']\ntype = friend\ncontext = phones\noutofcall_message_context =
message\nallow = ulaw,alaw\md5secret ='+password+'\nhost =
dynamic\nallow=h263\nallow=h264\nallow=vp8\nvideosupport='+video+")
            file.close()

            os.system('chown asterisk:asterisk ./'+self.usr+".conf")
            shutil.move(str('./'+self.usr+'.conf'),str('../extra_extensions/'+self.usr+'.conf'))

        except:

            print('ERROR When trying to create new file invalid user will be returned to user...')
            self.InvalidPUResponse()
            self.ldap= -1

        os.system("asterisk -x 'database put exten "+self.stdDepNum+" "+self.usr+"'")
        os.system("asterisk -x 'database put exten "+self.usr+" "+self.stdDepNum+"'")

def run(server_class=HTTPServer, handler_class=S, port=4443):
    #server_address = (str(socket.gethostbyname(socket.gethostname())), port)
    server_address = ("", port)
    httpd = server_class(server_address, handler_class)
    httpd.socket = ssl.wrap_socket (httpd.socket, certfile='./server.pem', server_side=True)
    print('\n-----> Starting httpd...<-----\n\n\t\tServer running:
```

```
'+str(socket.gethostbyname(socket.gethostname()))+":"+str(port))
    httpd.serve_forever()

if __name__ == "__main__":
    from sys import argv

    if len(argv) == 2:
        run(port=int(argv[1]))
    else:
        run()
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλων επεξεργασίας (ctrl+x).

Στην συνέχεια προκειμένου να μπορέσει η σελίδα να επικοινωνήσει με το σύστημα του LDAP πρέπει να εγκατασταθεί το πακέτο ldap-util.

```
apt-get install ldap-util
```

Επίσης για να είναι δυνατόν ο εξυπηρέτης που περιγράφεται στο https.py να είναι λειτουργήσιμος απαραίτητο είναι να δημιουργηθούν τα αρχεία ~/www/post/postInvalidUP.html, ~/www/post/postOK.html και ~/www/post/index/index.html, τα οποία θα περιέχουν κώδικα γραμμένο στην γλώσσα σήμανσης υπερκειμένου (HTML - **H**yper**T**ext **M**arkup **L**anguage):

A) ~/www/post/postInvalidUP.html :

```
touch ~/www/post/postInvalidUP.html
```

Ανοίγουμε το περιβάλλον επεξεργασίας:

```
pico ~/www/post/postInvalidUP.html
```

και επικολλούμε το κείμενο που ακολουθεί (ctrl+shift+v).

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Asterisk invalid User or Pass</title>
<style>
      body {

  font-size:25;
  background:
url("http://www.dbnet.ece.ntua.gr/~stefanak/web_map_book/programs/HTML/photos/central_b
uilding.jpg") no-repeat center;
  background-size: cover;
}
.wrapper {
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
  width: 100%;
  height: 100%;
  margin: 0 auto;
  overflow: auto;
  text-shadow: #222 0 0 10px;
  color:white;
  font-weight:bold;
  font-family: 'Gloria Hallelujah', cursive;
  background-color: rgba(0, 0, 0, 0.3)
}
</style>
</head>
<body>
<div class="wrapper">
<p class="title" style="text-align: center;" >Δημιουργία λογαριασμού στην υπηρεσία
```

```
τηλεφωνικού κέντρου</p>
<form method="post">
<input type="text" placeholder="Username(@hua.gr)" name="usr" style="width: 70%;margin: 3 auto; margin-left: 15% ;text-align: center"><br/>
<input type="password" placeholder="Password (ίδιο με maill)" name="psw" style="width: 70%;margin: 3 auto; margin-left: 15%;text-align: center"><br/>
<select name="cat" style="width: 70%;margin: 3 auto; margin-left: 15% ;text-align: center"><br/>
<option value="Dr">Καθηγητής</option>
<option value="Stf">Προσωπικό</option>
<option value="Std">Φοιτητής</option>
</select>
<br/>
<input class="chk" type="checkbox" name="permition" value="permition" onclick="document.getElementById('submit').disabled = ! document.getElementById('submit').disabled;" style=";margin: 6 auto;margin-left: 15%"/>
<div class="permition" style="font-size: 12; margin-left: 25%; margin-top: -20px" >Συμφωνώ με τους
<a href="#" onClick="alert('Αν πατήσετε «Συμφωνώ με τους όρους», συναινείτε στην αναζήτηση και αποθήκευσή του αριθμού τηλεφώνου που χρησιμοποιείται εντός του πανεπιστημίου, μέσω της υπηρεσία καταλόγου (LDAP) πιστοποιώντας την πρόσβασή σ' αυτήν με τα στοιχεία Username, Password που εσείς παρέχεται. Τα προαναφερόμενα στοιχεία, θα χρησιμοποιούνται για την σύνδεσή σας στην υπηρεσία τηλεφωνικού κέντρου «Asterisk» , ενώ για την δική σας ασφάλεια το Password θα κρυπτογραφηθεί.')" style="color: #ffffff ">όρους</a>.</p><br/>
<input type="submit"  id="submit" disabled="disabled" class="state" value="Δημιουργία Λογαριασμού" style="width: 92%;margin: 3 auto;margin-left: -12%">
<p class="title" style="text-align: center;margin-left: -25%;font-size: 14">Λάθος Username ή Password.</p>
</form>
</p>
</div>
</html>
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλον επεξεργασίας (ctrl+x).

B) ~/www/post/postOK.html :

```
touch ~/www/post/postOK.html
```

Ανοίγουμε το περιβάλλον επεξεργασίας:

```
pico ~/www/post/postOK.html
```

και επικολλούμε το κείμενο που ακολουθεί (ctrl+shift+v).

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Asterisk account created</title>
<style>
     body {
  font-size:25;
  background:
url("http://www.dbnet.ece.ntua.gr/~stefanak/web_map_book/programs/HTML/photos/central_b
uilding.jpg") no-repeat center;
  background-size: cover;
}
.wrapper {
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
  width: 100%;
  height: 100%;
  margin: 0 auto;
  overflow: auto;
  text-shadow: #222 0 0 10px;
  color:white;
  font-weight:bold;
```

```
  font-family: 'Gloria Hallelujah', cursive;
  background-color: rgba(0, 0, 0, 0.2)
}
</style>
</head>
<body>
<div class="wrapper">
<p class="title" style="text-align: center;" >Επιτυχής εγγραφή στην πλατφόρμα Asterisk του Χαροκοπείου Πανεπιστημιού Αθηνών</p>
</div>
</html>
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλων επεξεργασίας (ctrl+x).

Γ) ~/www/post//index/index.html :

```
touch ~/www/post//index/index.html
```

Ανοίγουμε το περιβάλλον επεξεργασίας:

```
pico ~/www/post//index/index.html
```

και επικολλούμε το κείμενο που ακολουθεί (ctrl+shift+v).

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Asterisk account creator</title>
<style>
     body {

  font-size:25;
  background:
```

```
url("http://www.dbnet.ece.ntua.gr/~stefanak/web_map_book/programs/HTML/photos/central_building.jpg") no-repeat center;
  background-size: cover;
}

.wrapper {
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
  width: 100%;
  height: 100%;
  margin: 0 auto;
  overflow: auto;
  text-shadow: #222 0 0 10px;
  color:white;
  font-weight:bold;
  font-family: 'Gloria Hallelujah', cursive;
  background-color: rgba(0, 0, 0, 0.3)
}
</style>
</head>
<body>
<div class="wrapper">
<p class="title" style="text-align: center;" >Δημιουργία λογαριασμού στην υπηρεσία τηλεφωνικού κέντρου</p>
<form method="post">
<input type="text" placeholder="Username(@hua.gr)" name="usr" style="width: 70%;margin: 3 auto; margin-left: 15% ;text-align: center"><br/>
<input type="password" placeholder="Password (ίδιο με maill)" name="psw" style="width: 70%;margin: 3 auto; margin-left: 15%;text-align: center"><br/>
<select name="cat" style="width: 70%;margin: 3 auto; margin-left: 15% ;text-align: center"><br/>
```

```
<option value="Dr">Καθηγητής</option>
<option value="Stf">Προσωπικό</option>
<option value="Std">Φοιτητής</option>
</select>
<br/>
<input class="chk" type="checkbox" name="permition" value="permition"
onclick="document.getElementById('submit').disabled = !
document.getElementById('submit').disabled;" style=";margin: 6 auto;margin-left: 15%"/>
<div class="permition" style="font-size: 12; margin-left: 25%; margin-top: -20px" >Συμφωνώ
με τους
<a href="#" onClick="alert('Αν πατήσετε «Συμφωνώ με τους όρους», συναινείτε στην
αναζήτηση και αποθήκευσή του αριθμού τηλεφώνου που χρησιμοποιείται εντός του
πανεπιστημίου, μέσω της υπηρεσία καταλόγου (LDAP) πιστοποιώντας την πρόσβασή σ' αυτήν
με τα στοιχεία Username, Password που εσείς παρέχεται. Τα προαναφερόμενα στοιχεία, θα
χρησιμοποιούνται για την σύνδεσή σας στην υπηρεσία τηλεφωνικού κέντρου «Asterisk» , ενώ
για την δική σας ασφάλεια το Password θα κρυπτογραφηθεί.')" style="color: #ffffff
">όρους</a>.</p><br/>
<input type="submit"  id="submit" disabled="disabled" class="state" value="Δημιουργία
Λογαριασμού" style="width: 92%;margin: 3 auto;margin-left: -12%">
</form>
</p>
</div>
</html>
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλον επεξεργασίας (ctrl+x).

Βοήθημα:

Το κείμενο που μόλις επικολλήσαμε περιέχει τον κώδικα. Είναι γραμμένο στην γλώσσα προγραμματισμού python3, οι οποία αποτελεί μια από τις πιο γνωστές γλώσσες προγραμματισμού (scripted programing language). Η σύνταξη της python3 είναι εκτός των πλαισίων αυτής της εργασίας, αλλά στην αρχή της ενότητας υπάρχει το Σχεδιάγραμμα 4.1.1 το οποίο περιγράφει όχι μόνο την ροή λειτουργίας του https.py αλλά και όλου του συστήματος της σελίδας.

Πληροφορίες σχετικά με την HTML και την Python:

Α. Η γλώσσα **HTML** είναι εργαλείο περιγραφής δομής ιστοσελίδων και εφαρμογών ιστού. Οι φιλομετριτές (browser) λαμβάνουν έγγραφα HTML από έναν διακομιστή και αποδίδουν το περιεχόμενο σε μορφή ιστοσελίδας.

Β. Η **Python** είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού η οποία δημιουργήθηκε με κύριο στόχο να έχει υψηλή αναγνωσιμότητα, ευκολία χρήσης και εύκολο συντακτικό. Είναι ιδιαίτερα δημοφιλής γιατί επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ' ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java.

Για την δημιουργία ασφαλούς διόδου επικοινωνίας όπως επιβάλει το πρότυπο ασφαλούς μεταφοράς υπερκειμένου (HTTPS), πρέπει να παραχθεί το αρχείο server.pem το οποίο θα περιέχει το ιδιωτικό μας κλειδί.

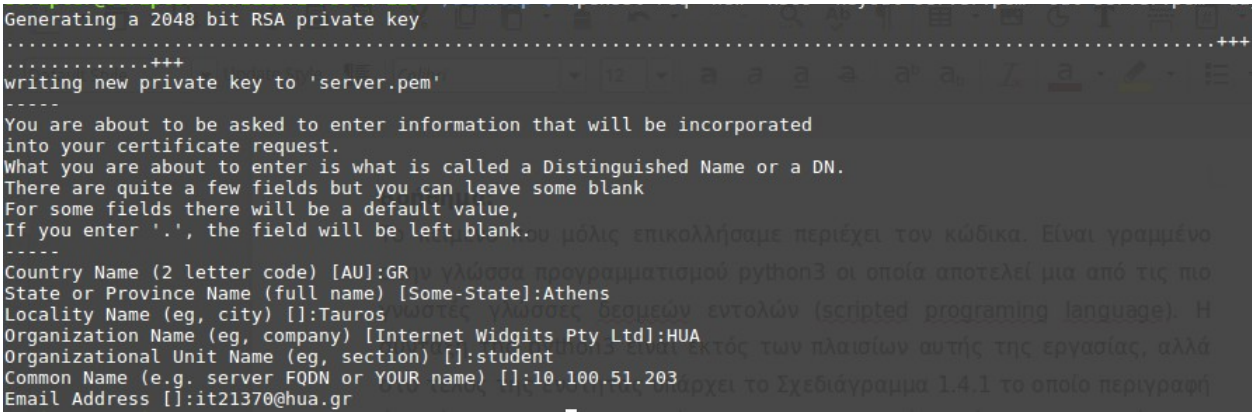
Για την παραγωγή του κλειδιού απαραίτητο είναι η εγκατάστασή του openssl:

```
apt-get install openssl
```

Όταν η εγκατάσταση ολοκληρωθεί το επόμενο βήμα είναι η παραγωγή του κλειδιού:

```
openssl req -new -x509 -keyout server.pem -out server.pem -days 365 -nodes
```

Αυτή η εντολή έχει σαν αποτέλεσμα την εμφάνιση στο τερματικό μιας σειράς από πεδία τα οποία πρέπει να απαντηθούν όσο το δυνατόν πιο ολοκληρωμένα προκειμένου να υπάρξει το HTTPS (Εικόνα 4.1.1).



Εικόνα 4.1.1

Μεταφέρουμε το κλειδί στον κατάλογο ~/www/

```
mv ./server.pem ~/www
```

Όπως και στην περίπτωσή του Asterisk ο εξυπηρέτης πρέπει να ενεργοποιείται μαζί με το μηχάνημα. Προκειμένου να επιτευχθεί η παραπάνω παρατήρηση πρέπει να δημιουργηθεί μια δέσμη εντολών τύπου Bash η οποία όταν εκτελεστεί να ενεργοποιεί το https.py.

```
touch ~/www/post/httpsLaunch.sh
```

Ανοίγουμε το περιβάλλον επεξεργασίας:

```
pico ~/www/post/httpsLaunch.sh
```

και επικολλούμε το κείμενο που ακολουθεί (ctrl+shift+v).

```
#!/bin/sh
#httpsLancer.sh

cd /
cd ~/www
python3 ./https.py > logs/$(date)
cd /
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλον επεξεργασίας (ctrl+x).

Συνεχίζοντας πρέπει να δώσουμε αδεία εκτέλεσης στην δέσμη εντολών που μόλις επικολλήσαμε.

```
chmod 755 ~/www/post/httpsLaunch.sh
```

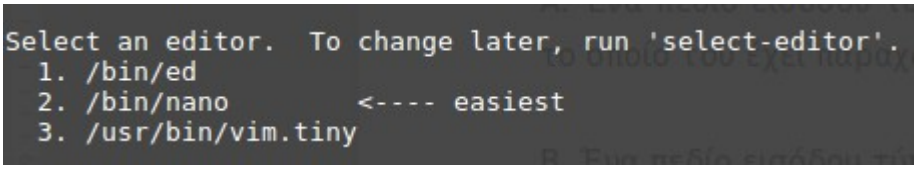
Δημιουργούμε τον κατάλογο που θα αποθηκεύονται τα αρχεία καταγραφής (logs):

```
touch ~/www/logs
```

Τέλος χρησιμοποιούμε την εντολή crontab:

```
crontab -e
```

η οποία θα μας ρωτήσει ποιον επεξεργαστή κειμένου προτιμάμε [επιλέγουμε nano] και κατόπιν θα τον εκκινήσει (Εικόνα 4.1.2).



Εικόνα 4.1.2

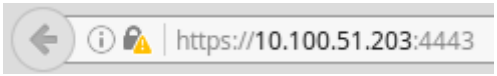
Στο τέλος του αρχείου πρέπει να επικολλήσουμε την γραμμή:

```
@reboot sh /etc/asterisk/https/httpsLauncher.sh >/etc/asterisk/https/logs 2>&1
```

Αποθηκεύουμε (ctrl+o) και αποχωρούμε από το περιβάλλον επεξεργασίας (ctrl+x).

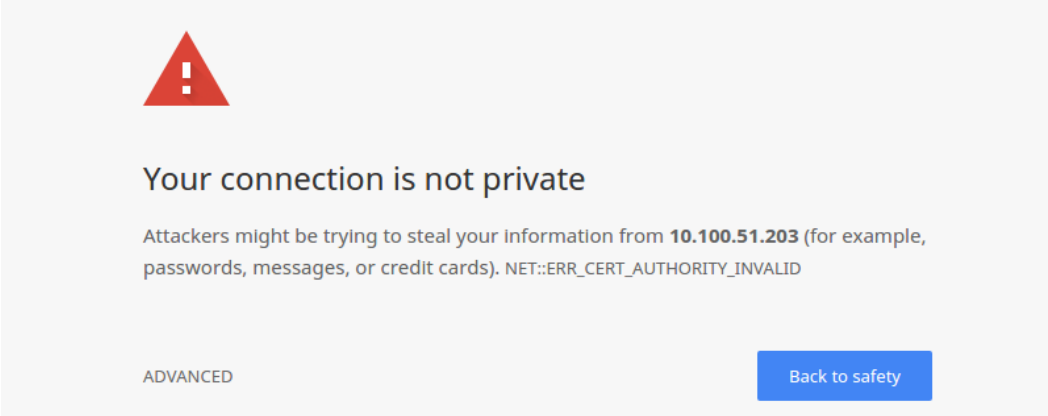
Οδηγίες Χρήσης της ιστοσελίδας

Το πρώτο βήμα είναι να επισκεφτούμε την σελίδα πληκτρολογώντας https://10.100.51.203:4443/ στον φιλομετριτή της αρεσκείας μας (Εικόνα 4.12) .



Εικόνα 4.1.2

Κατά την επίσκεψη στην ιστοσελίδα είναι πολύ πιθανόν να εμφανιστεί ένα μήνυμα στο φιλομετριτή (browser) το οποίο θα μας ενημερώνει ότι η σελίδα είναι μη ασφαλής (Εικόνα 4.1.3). Αυτό συμβαίνει διότι παρόλο που το κλειδί μας είναι έγκυρο δεν έχει πιστοποιηθεί από κάποιον τρίτο. Υπάρχουν πολλές δικτυακές εφαρμογές που παρέχουν τέτοιες υπηρεσίες αλλά καμία δεν υποστηρίζει σελίδες εσωτερικού δικτύου.



Εικόνα 4.1.3

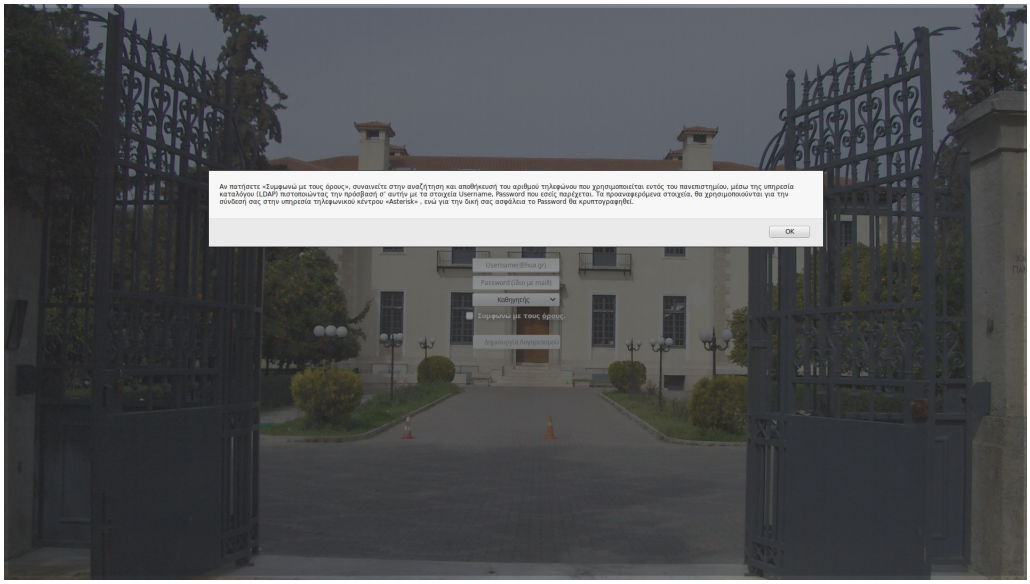
Παρακάμψετε την προειδοποίηση [οι τρόποι ποικίλουν ανάλογα τον φιλομετριτή (browser)].



Εικόνα 4.1.4

Σε αυτή την σελίδα συμπληρώνουμε τα στοιχεία μας στα πεδία Username [το maill το οποίο έχει παραχωρήσει το Πανεπιστήμιο (με ή χωρίς @hua.gr)] και Password [ο κωδικός που χρησιμοποιείται σε όλες τις υπηρεσίες του Χαροκοπείου].

Για να ενεργοποιηθεί το κουμπί “Δημιουργία λογαριασμού” πρέπει να πατηθεί το κουμπί επιλογής το οποίο βρίσκεται δίπλα από το “Συμφωνώ με τους όρους”.
Οι όροι είναι προσβάσιμοι με το πάτημα της λέξης “όροι“ (Εικόνα 4.1.5).



Εικόνα 4.1.5

Σε περίπτωση που έχουμε κάνει κάποιο λάθος στα στοιχεία μας, τότε εμφανίζεται η Εικόνα 4.1.6.



Εικόνα 4.1.6

Στην περίπτωση που τα στοιχεία είναι σωστά εμφανίζεται η Εικόνα 4.1.7.



Εικόνα 4.1.7

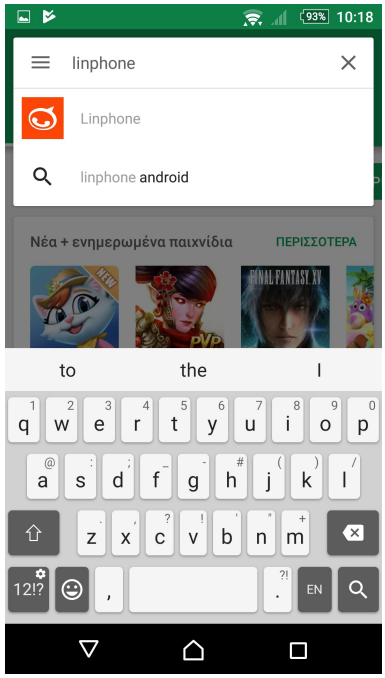
Πηγές:

1. https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

4.3 Οδηγίες Σύνδεσης

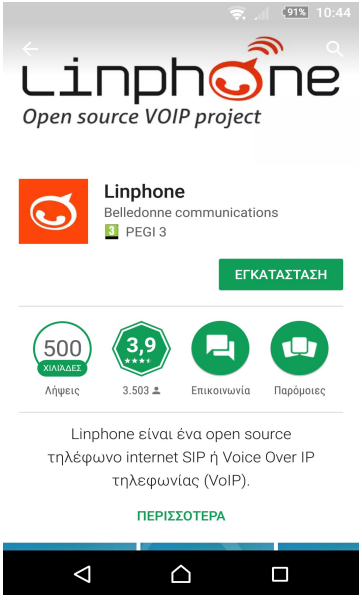
Android

Το πρώτο βήμα είναι η είσοδός μας στο Play Store και η αναζήτηση της εφαρμογής Linphone (Εικόνα 4.1.8).



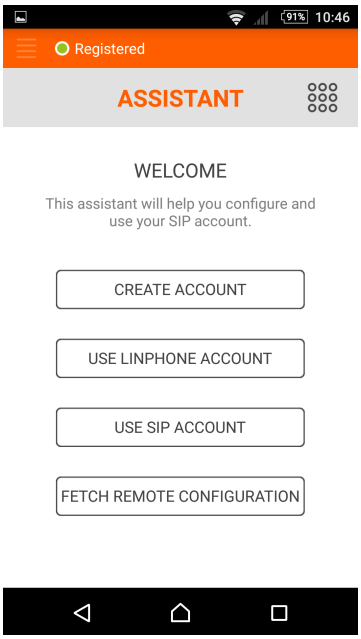
Εικόνα 4.1.8

Στην συνέχεια πατάμε το πλήκτρο “ΕΓΚΑΤΑΣΤΑΣΗ “(Εικόνα 4.1.9).



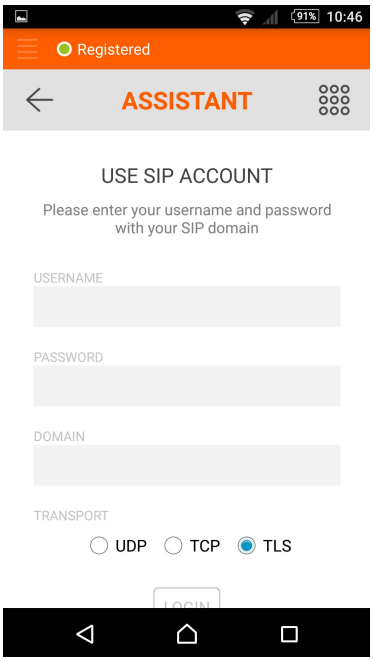
Εικόνα 4.1.9

Όταν ολοκληρωθεί η εγκατάσταση ανοίγουμε την εφαρμογή και στην οθόνη εμφανίζεται η Εικόνα 4.1.10.

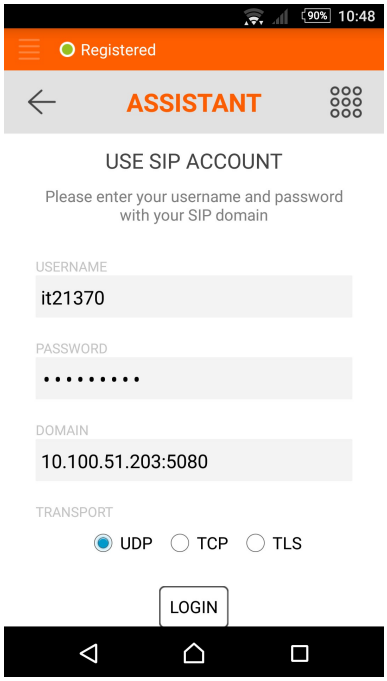


Εικόνα 4.1.10

Επιλέγοντας την επιλογή “USE SIP ACCOUNT” εμφανίζεται στην οθόνη η Εικόνα 4.1.11.



Εικόνα 4.1.11



Εικόνα 4.1.12

Στο πεδίο “USERNAME” βάζουμε το mail μας χωρίς την κατάληξη @hua.gr. Αντίστοιχά στο πεδίο “PASSWORD” βάζουμε το συνθηματικό του mail μας. Στο “DOMAIN” πληκτρολογούμε 10.100.51.203:5080 που είναι η διεύθυνση δικτύου της υπηρεσίας Asterisk. Φροντίζουμε το “TRANSPORT” να είναι UDP και πατάμε το “LOGIN” Εικόνα 4.1.12

Οι ρυθμίσεις ολοκληρώθηκαν. Μπορείτε να καλέσετε τον αριθμό οποιουδήποτε χρήστη συμμετέχει στο κέντρο ή τους αριθμούς επίδειξης που αναφέρονται στο Κεφάλαιο 3 (Σελ. 60) για δοκιμή. Μια άλλη επιλογή θα ήταν να στείλετε μήνυμα.

Linux

Αρχικά για να εγκαταστήσουμε το linphone για linux καλό θα ήταν να ενσωματωθούν οι απαραίτητες βιβλιοθήκες.

```
sudo apt-get install git build-essential automake autoconf libtool   \
intltool libgtk2.0-dev libspeexdsp-dev libavcodec-dev libswscale-dev \
libx11-dev libxv-dev libgl1-mesa-dev libreadline-dev libgsm1-dev     \
libtheora-dev libsqlite3-dev libupnp-dev libsrtp-dev open-vm-tools   \
open-vm-tools-dev cmak* libmbedtls-dev libmbedtls-doc libmbedtls10   \
libcunit1 libcunit1-dev libantlr3c-dev libantlr3-runtime-java antlr3 \
libortp-dev libortp9 libmediastreamer-base3 libmediastreamer-dev     \
extra-cmake-modules yasm doxygen graphviz \
```

Ακολουθεί το κατέβασμα του πηγαίου κώδικα (source code):

```
git clone https://github.com/BelledonneCommunications/linphone-desktop.git
```

Η ενημέρωσή του:

```
cd linphone-desktop
```

```
git submodule sync && git submodule update --init --recursive
```

Η προετοιμασία του συστήματός:

```
./prepare.py
```

και η εγκατάστασή:

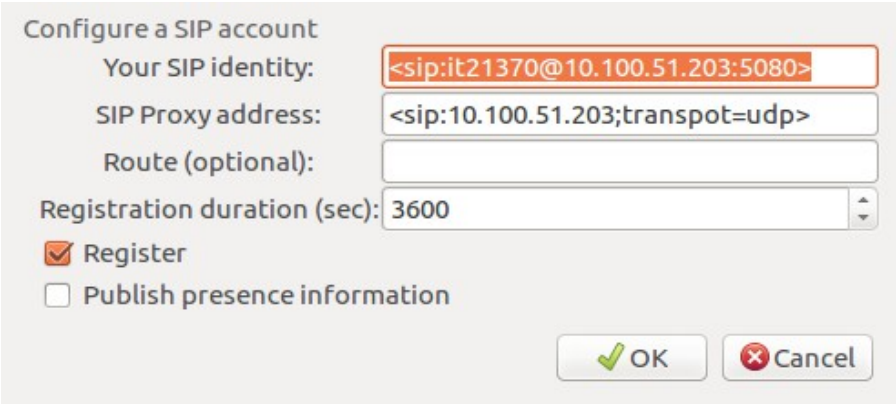
```
make
```

```
sudo make install
```

Προσοχή:

Η εγκατάστασή του Linphone γίνεται σε κάποιο μηχάνημα εκτός του διακομιστή.

Όταν η εγκατάσταση τελειώσει ανοίγουμε το linphone. Κλείνουμε το παράθυρό που μας εμφανίζεται μόνο κατά την πρώτη εκκίνηση και από το options --> Preferences --> Manage SIP Accounts --> Add συμπληρώνουμε τα στοιχεία μας κατά αντιστοιχία με αυτά της Εικόνας 4.1.13.



Εικόνα 4.1.13

Βοήθημα:

Αντικαταστήστε το it21370 με το Username σας (χωρίς @hua.gr) και το Password θα σας ζητηθεί αργότερα.

Επίλογος

Το VoIP είναι το μέλλον της τηλεφωνίας.

Οι μεγαλύτερες εταιρίες τηλεπικοινωνιών έχουν αρχίσει είδη να εγκαταλείπουν το παραδοσιακό PSTN και να κατευθύνονται στο VoIP. Τα πλεονεκτήματα είναι πάρα πολλά καθώς εκμεταλλεύονται το γεγονός ότι το σήμα κινείται πλέον πάνω από τα πρωτόκολλα διαδικτύου με αποτέλεσμα να είναι πιο εύκολο και οικονομικό να δημιουργηθούν πακέτα τα οποία να παρέχουν περισσότερες δυνατότητες ανάλογα με τις ανάγκες του εκάστοτε καταναλωτή.

Εκτός από τις εταιρίες τηλεπικοινωνιών ενδιαφέρων στο VoIP δείχνουν και οι κυρίαρχοι του χώρου της τεχνολογίας όπως είναι η Google, το Facebook, η Microsoft, η Electronic Games κ.α. οι οποίοι προσδοκούν στην προσφορά ευκολότερης και φθηνότερης επικοινωνίας στους χρηστές τους.

Το 2011 κυκλοφόρησε το WebRTC (Web **R**eal **T**ime **C**omunication) που αποτελεί το πρότυπο δημιουργίας καναλιού επικοινωνίας μεταξύ δύο χρηστών ενός τηλεφωνικού κέντρου μέσα από φιλομετριτή (Browser). Το WebRTC αποτελεί υπο-πρότυπο του SIP άρα ανήκει στο σώμα της VoIP τεχνολογίας.