ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΜΠΣ Τμήμα: Ασφάλεια Ψηφιακών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΤΟΜΕΑΣ: ΑΣΦΑΛΕΙΑ ΣΕ ΣΥΣΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΑΚΗΣ ΤΗΛΕΦΩΝΙΑΣ (VoIP)»

«ΘΕΜΑ: ΜΕΛΕΤΗ ΑΣΦΑΛΕΙΑΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΕ IMS/SIP ΣΥΣΤΗΜΑΤΑ»

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ

ΚΑΤΣΑΝΙΚΑΚΗΣ ΧΡΗΣΤΟΣ

ΜΤΕ 0910

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΛΑΜΠΡΙΝΟΥΔΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Ιούνιος 2011

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1. Σκοπός της διπλωματικής εργασίας

Στα πλαίσια της παρούσας διπλωματικής εργασίας γίνεται αρχικά μια σύντομη αναφορά στο ιστορικό δημιουργίας του IMS (IP Multimedia Subsystem), ενώ στην συνέχεια γίνεται μια εκτεταμένη περιγραφή του μοντέλου και του τρόπου λειτουργίας του. Συγκεκριμένα, αναλύονται οι οντότητες και τα σημεία αναφοράς από τα οποία αποτελείται το IMS καθώς και τα πρωτόκολλα που χρησιμοποιεί. Στη συνέχεια γίνεται μία αναφορά στην ασφάλεια του IMS, όπου αναφέρονται πιθανοί κίνδυνοι και απειλές για το IMS καθώς και τρόποι αντιμετώπισής τους. Τέλος ακολουθεί μια σειρά από υλοποιήσεις και εφαρμογές που πραγματοποιήσαμε πάνω στο συγκεκριμένο θέμα και εξάγονται χρήσιμα συμπεράσματα.

1.2. Εισαγωγή στο θέμα

Η IMS (IP Multimedia Subsystem) αρχιτεκτονική είναι μία πραγμάτωση των αρχών του NGN (Next Generation Networks) και βασίζεται στο SIP πρωτόκολλο για τον έλεγχο των συνόδων. Οι IMS προδιαγραφές ορίζουν ολόκληρη την αρχιτεκτονική ελέγχου των πολυμεσικών συνόδων πάνω από τον UMTS τομέα μεταγωγής πακέτου. Με το IMS οι διαχειριστές παρέχουν αξιόπιστο έλεγχο συνόδων και καλύτερες ενοποιημένες υπηρεσίες.

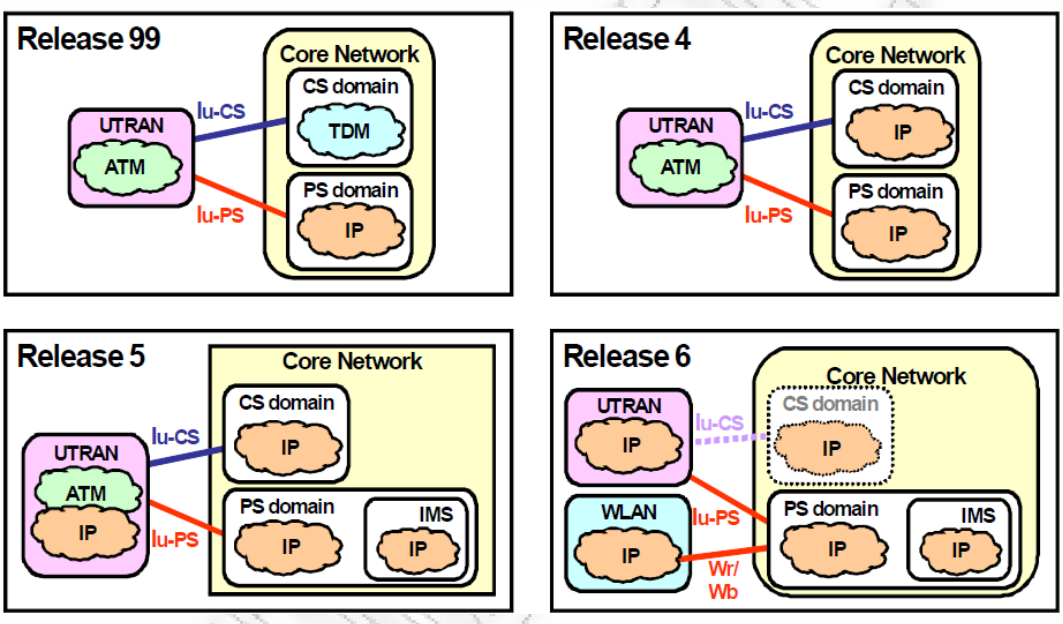
Το IMS υποστηρίζει την εγγραφή του χρήστη και της τερματικής συσκευής σε μία συγκεκριμένη τοποθεσία στο δίκτυο. Ως μέρος της εγγραφής, το IMS υποστηρίζει πιστοποίηση και άλλες ρυθμίσεις ασφαλείας. Το IMS χρησιμοποιεί έλεγχο βασισμένο στο SIP. Οι υπηρεσίες που υποστηρίζονται από το IMS μπορεί να περιέχουν υπηρεσίες πολυμεσικού χαρακτήρα είτε όχι.

1.3. Σύλληψη του IMS

Το IMS (IP Multimedia Subsystem) αφορά ένα καινούργιο πλαίσιο αρχιτεκτονικής που προορίζεται για τα κινητά δίκτυα, παρέχοντας IP τηλεπικοινωνιακές υπηρεσίες. Σχεδιάστηκε αρχικά από ένα βιομηχανικό φόρουμ που ονομαζόταν 3G.IP, το οποίο σχηματίστηκε το 1999. Το 3G.IP ανέπτυξε την αρχική IMS αρχιτεκτονική, την οποία έφερε στο Third Generation Partnership Project (3GPP) ως μέρος της εργασίας προτυποποίησης των 3G συστημάτων κινητής τηλεφωνίας στα UMTS δίκτυα. Η αρχική του μορφή (release 5) , η οποία

παρουσιάστηκε με την εμφάνιση των πολυμέσων βασισμένων στο SIP πρωτόκολλο, αναπαριστούσε μία προσέγγιση παράδοσης « υπηρεσιών Internet » πάνω από το GPRS και το παλιότερο GSM και αποτέλεσε την εξέλιξη των 2G δικτύων σε 3G.

Στη συνέχεια αναπτύχθηκε από τον οργανισμό 3GPP2 μία αρχιτεκτονική IP πολυμέσων αναφερόμενη ως Multimedia Domain (MMD) για τα 3G Code Division Multiple Access 2000 (CDMA2000) δίκτυα, η οποία τελικά εναρμονίστηκε με το IMS. Η υποστήριξη των WLAN δικτύων αλλά και υπηρεσιών πραγματικού χρόνου πραγματοποιήθηκε με την 3GPP release 6, Τέλος, η 3GPP release 7 πρόσθεσε την υποστήριξη για τα σταθερά δίκτυα σε συνεργασία με τη TISPAN release R1.1.



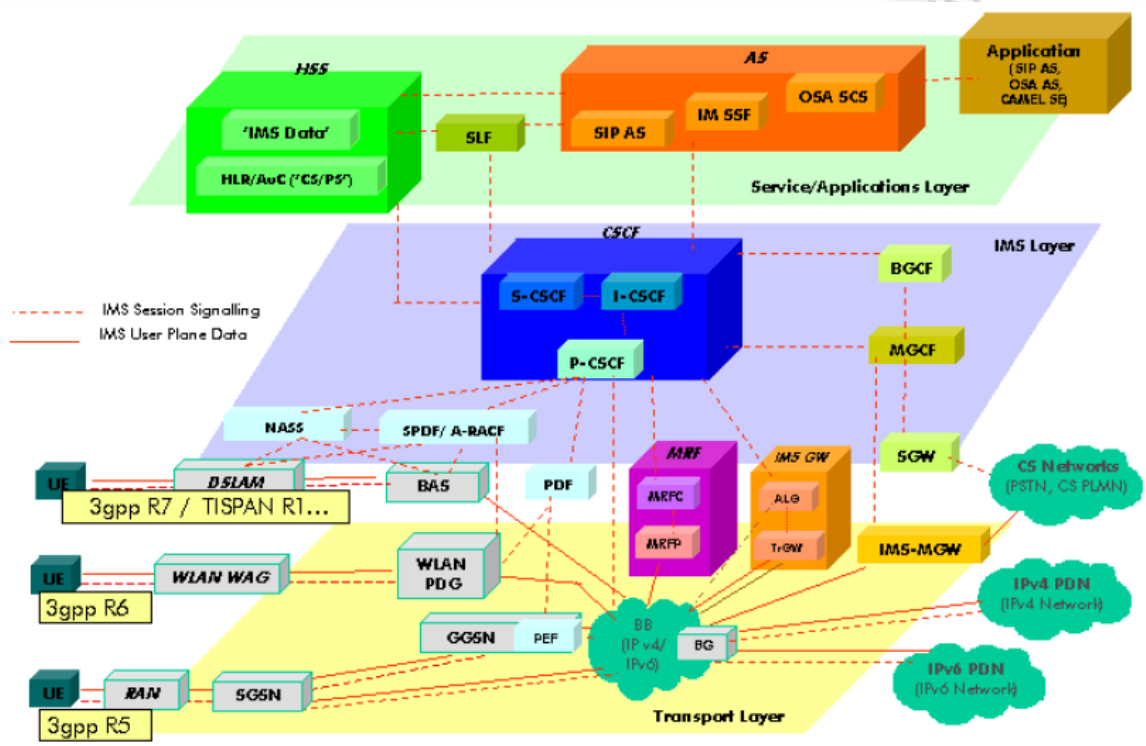
ΕΙΚΟΝΑ 1: ΕΚΔΟΣΕΙΣ ΤΗΣ IMS ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Η σύλληψη του IMS (IP Multimedia Subsystem) έγινε για να ικανοποιήσει τις παρακάτω απαιτήσεις δικτύων και χρηστών:

- Παράδοση person-to-person πολυμεσικών υπηρεσιών πραγματικού χρόνου που είναι βασισμένες στο IP πρωτόκολλο (π.χ. φωνή ή βιντεοτηλεφωνία) καθώς επίσης και person-to-machine επικοινωνιών (π.χ. υπηρεσίες παιχνιδιών).
- Πλήρης ενοποίηση πολυμεσικών υπηρεσιών πραγματικού χρόνου και μη πραγματικού χρόνου (π.χ. ζωντανό streaming και ομιλία).
- Δυνατότητα αλληλεπίδρασης μεταξύ διαφορετικών υπηρεσιών και εφαρμογών (π.χ. συνδυασμένη χρήση παρουσίας και στιγμιαίας μηνυματοδοσίας).
- Εύκολη εγκατάσταση για τον χρήστη πολλαπλών υπηρεσιών σε μία απλή σύνοδο ή πολλαπλών ταυτόχρονων συγχρονισμένων συνόδων.

2. ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ IMS

2.1. Αρχιτεκτονική του IMS



ΕΙΚΟΝΑ 2: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ IMS

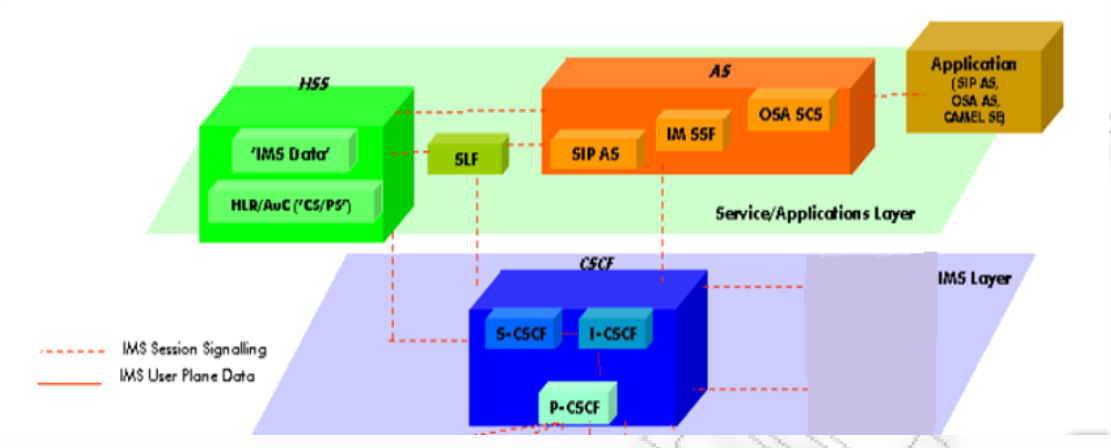
2.1.1. Δίκτυο Πρόσβασης

Ο χρήστης μπορεί να συνδεθεί σε ένα IMS (IP Multimedia Subsystem) δίκτυο με ποικίλους τρόπους, οι οποίοι όλοι χρησιμοποιούν το πρωτόκολλο IP (Internet Protocol). Απευθείας IMS τερματικά (όπως είναι τα κινητά τηλέφωνα, τα PDAs, και οι ηλεκτρονικοί υπολογιστές) μπορούν να καταγραφούν άμεσα σε ένα IMS δίκτυο.

Η μόνη απαίτηση είναι να μπορούν να χρησιμοποιήσουν το IPv6 (επίσης το IPv4 στο αρχικό IMS) και να τρέξουν τους SIP (Session Initiation Protocol) user agents. Υποστηρίζεται η σταθερή πρόσβαση (π.χ. DSL (Digital Subscriber Line), modems, Ethernet), η κινητή πρόσβαση (π.χ. W-CDMA, CDMA2000, GSM, GPRS) και η ασύρματη πρόσβαση (π.χ. WLAN, WiMAX).

Άλλα τηλεφωνικά συστήματα όπως το παλαιό αναλογικό τηλεφωνικό σύστημα (POTS) ,H.323 και μη συμβατά με το IMS VoIP συστήματα υποστηρίζονται μέσω πυλών (gateways).

2.1.2. Δίκτυο Πυρήνα



ΕΙΚΟΝΑ 3: ΑΡΧΙΤΕΚΤΟΝΙΚΗ IMS ΠΥΡΗΝΑ

2.1.2.1. Στοιχεία Ελέγχου

Αυτή η ομάδα στοιχείων αποτελείται από τρία διαφορετικά είδη SIP εξυπηρετητών (proxies), ομαδικά καλούμενοι Call Session Control Function (CSCF), οι οποίοι είναι υπεύθυνοι για την δρομολόγηση των κλήσεων και τον έλεγχο των συνόδων, δηλαδή για την επεξεργασία των SIP πακέτων:

✓ **Proxy-CSCF (P-CSCF)**

Ο Proxy-CSCF (P-CSCF) είναι ένας SIP proxy που αποτελεί το πρώτο σημείο επαφής με το IMS τερματικό. Μπορεί να τοποθετηθεί είτε στο δίκτυο επίσκεψης είτε στο πάτριο δίκτυο (όταν το δίκτυο επίσκεψης δεν είναι συμβατό με το IMS).

Συγκεκριμένα, ο P-CSCF:

- Ανατίθεται σε ένα IMS τερματικό κατά τη διάρκεια της εγγραφής και δεν αλλάζει για όσο διαρκεί η εγγραφή
- Βρίσκεται στο μονοπάτι όλων των μηνυμάτων σηματοδοσίας και μπορεί να επιθεωρεί κάθε μήνυμα
- Πιστοποιεί τον χρήστη και εγκαθιδρύει μία IPsec σχέση ασφάλειας με το IMS τερματικό. Αυτό εμποδίζει τις επιθέσεις παραπλάνησης και προστατεύει την ατομικότητα του χρήστη. Άλλοι κόμβοι εμπιστεύονται το P-CSCF και δεν χρειάζεται να πιστοποιούν την ταυτότητα του χρήστη ξανά

- Εκτελεί συμπίεση και αποσυμπίεση των SIP μηνυμάτων γεγονός που μειώνει τη πλήρη διαδρομή πάνω από αργές ραδιοσυνδέσεις.
- Μπορεί να περιέχει μία Policy Decision Function (PDF), η οποία εξουσιοδοτεί πόρους μέσων (media) όπως ποιότητα εξυπηρέτησης (QoS) πάνω στο επίπεδο μέσων. Χρησιμοποιείται για έλεγχο διαχείρισης, διαχείριση εύρους ζώνης, κ.τ.λ.. Η PDF μπορεί επίσης να αποτελεί μία ξεχωριστή λειτουργία
- Παράγει καταγραφές χρεώσεων

✓ **Serving-CSCF (S-CSCF)**

Ο Serving-CSCF (S-CSCF) είναι ο κεντρικός κόμβος του επιπέδου σηματοδοσίας. Είναι ένας SIP εξυπηρετητής, αλλά διεξάγει επίσης έλεγχο συνόδων. Είναι πάντα τοποθετημένος στο πάτριο δίκτυο. Χρησιμοποιεί τις Diameter Cx και Dx διεπαφές για το HSS για downloading και uploading. Δεν αποθηκεύει τοπικά πληροφορίες χρήστη. Όλες οι απαραίτητες πληροφορίες φορτώνονται από το HSS.

Συγκεκριμένα ο S-CSCF:

- Χειρίζεται τις SIP εγγραφές, γεγονός που του επιτρέπει να δεσμεύει την τοποθεσία του χρήστη (π.χ. την IP διεύθυνση του τερματικού) και την SIP διεύθυνση
- Βρίσκεται στο μονοπάτι όλων των μηνυμάτων σηματοδοσίας και μπορεί να επιθεωρεί κάθε μήνυμα
- Αποφασίζει σε ποιον (ποιους) εξυπηρετητές εφαρμογής (application server) το SIP μήνυμα θα προωθηθεί, έτσι ώστε να παρέχει τις υπηρεσίες
- Επιβάλλει την πολιτική διαχείρισης του δικτύου
- Μπορούν να υπάρξουν πολλαπλά S-CSCF στο δίκτυο για λόγους διαμοίρασης φορτίου και υψηλής διαθεσιμότητας. Το HSS είναι αυτό που αναθέτει το S-CSCF στον χρήστη όταν ζητείται από το I-CSCF.

✓ **Interrogating-CSCF (I-CSCF)**

Ο Interrogating-CSCF (I-CSCF) είναι μία άλλη λειτουργία του SIP που είναι εγκατεστημένη στην άκρη του διοικητικού τομέα. Η IP διεύθυνσή του είναι δημοσιευμένη στον Domain Name System (DNS) του τομέα έτσι ώστε οι απομακρυσμένοι εξυπηρετητές να το βρίσκουν και να το χρησιμοποιούν ως σημείο προώθησης για SIP πακέτα σε αυτό το τομέα. Το I-CSCF επικοινωνεί με το HSS

χρησιμοποιώντας την Diameter Cx διεπαφή για να βρει την τοποθεσία του χρήστη (η Dx διεπαφή χρησιμοποιείται από το I-CSCF στο SLF για να εντοπιστούν τα HSS που χρειάζονται μόνο, και στη συνέχεια δρομολογεί την αίτηση SIP στο ανατιθέμενο SCSCF. Σύμφωνα με το Release 6 μπορεί επίσης να χρησιμοποιηθεί για να κρύψει το πάτριο δίκτυο από τον εξωτερικό κόσμο (κρυπτογραφημένο τμήμα του SIP μηνύματος), όπου σε αυτή τη περίπτωση ονομάζεται Topology Hiding Inter-network Gateway (THIG). Στο Release 7 αυτή η λειτουργία του «σημείου εισόδου» έχει αφαιρεθεί από το I-CSCF και είναι πλέον μέρος της Interconnection Border Control Function (IBCF). Η IBCF χρησιμοποιείται ως πύλη για τα εξωτερικά δίκτυα, και παρέχει NAT και Firewall λειτουργίες.

2.1.2.2. Στοιχεία Βάσης Δεδομένων

✓ **Home subscriber server (HSS)**

Ο Home subscriber server (HSS) είναι η κύρια βάση δεδομένων χρήστη η οποία υποστηρίζει τις οντότητες του IMS δικτύου που ουσιαστικά χειρίζονται τις κλήσεις. Περιέχει πληροφορίες σχετικές με την εγγραφή (user profiles), εκτελεί πιστοποίηση και εξουσιοδότηση του χρήστη, και μπορεί να παρέχει πληροφορίες σχετικά με την φυσική τοποθεσία του χρήστη. Είναι παρόμοιο με το GSM Home Location Register (HLR) και Authentication Center (AUC).

✓ **Subscriber Location Function (SLF)**

Ο Subscriber Location Function (SLF) είναι αναγκαίος για να προσδιορίζει τις διευθύνσεις των χρηστών όταν χρησιμοποιούνται πολλαπλοί HSSs. Το HSS και το SLF επικοινωνούν διαμέσου του πρωτοκόλλου Diameter. Αυτό το πρωτόκολλο ονομάζεται επίσης και AAA πρωτόκολλο (Authentication, Accounting, Authorization).

2.1.2.3. Στοιχεία υπηρεσιών

✓ **Application Servers (SIP AS) ,**

τα οποία φιλοξενούν και εκτελούν υπηρεσίες και αλληλεπιδρούν με το SCSCF με τη χρήση του πρωτοκόλλου SIP. Ανάλογα με την υπηρεσία, ο AS μπορεί να λειτουργήσει ως SIP proxy, SIP UA (user agent) ή SIP B2BUA (back to back user agent). Το S-CSCF, προωθεί το SIP αίτημα στο κατάλληλο SIP AS, το οποίο με τη σειρά του εκτελεί τη σχετική λογική της υπηρεσίας.

2.2. Πρωτόκολλα που χρησιμοποιεί το IMS (IP Multimedia Subsystem)

- ✓ **SIP Protocol**
- ✓ **Diameter Protocol**
- ✓ **Session Description Protocol (SDP)**
- ✓ **Common Open Policy Service protocol (COPS)**

2.2.1. SIP Protocol

Είναι ένα πρωτόκολλο ελέγχου στο επίπεδο εφαρμογής. Το SIP είναι πρωτόκολλο σηματοδοσίας για τη δημιουργία, μεταβολή και τερματισμό συνόδων με έναν ή περισσότερους συμμετέχοντες. Οι βασικές λειτουργίες του SIP είναι ο εντοπισμός ενός τερματικού σημείου, η ένδειξη επιθυμίας για επικοινωνία, η διαπραγμάτευση των παραμέτρων συνόδου για την αποκατάσταση της συνόδου και η διάλυση της συνόδου μετά την αποκατάστασή της.

2.2.2. Diameter Protocol

Η IMS αρχιτεκτονική έχει σημαντική αξία για τους παροχείς υπηρεσιών, αφού τους επιτρέπει να παρέχουν καινούργιες πολυμεσικές υπηρεσιές πραγματικού χρόνου και παρέχει στους τερματικούς χρήστες μία χωρίς ραφή εμπειρία σε πολλαπλά δίκτυα πρόσβασης. Το Base Diameter Protocol είναι ένα πρωτόκολλο ορισμένο από την IETF που παρέχει στις εφαρμογές ένα πλαίσιο εργασίας για λειτουργίες Πιστοποίησης (Authentication), Εξουσιοδότησης(Authorization), και Διαχείρισης Λογαριασμών (Accounting) που είναι γνωστές ως AAA πρωτόκολλο. Το Diameter λειτουργεί πάνω από τα αξιόπιστα πρωτόκολλα μεταφοράς όπως το TCP και το SCTP. Το Diameter πρωτόκολλο παρέχει τις παρακάτω λειτουργίες:

- Διαχείριση συνδέσεων και συνόδων
- Πιστοποίηση του χρήστη και διαπραγμάτευση δυνατοτήτων
- Αξιόπιστη διανομή των Attribute Value Pairs (AVPs)
- Υποστήριξη πρακτόρων (agents) για proxy, redirect και relay εξυπηρετητές
- Βασικές υπηρεσίες χρέωσης και διαχείρισης λογαριασμών

2.2.3. Session Description Protocol (SDP)

Παρέχει μία αναπαράσταση πληροφοριών, π.χ. για τα media, για τις transport διευθύνσεις και άλλων session description metadata, ανεξάρτητα από το πώς μεταδίδονται οι πληροφορίες αυτές. Το SDP είναι απλά ένα μορφότυπο (format) για

τη περιγραφή συνόδου (session description) και ενσωματώνει κάποιο πρωτόκολλο μεταφοράς (transport protocol) αλλά επιδιώκει να χρησιμοποιεί διάφορα πρωτόκολλα.

2.2.4. Common Open Policy Service protocol (COPS)

Το IMS χρησιμοποιεί το COPS για να εξασφαλίσει ποιότητα υπηρεσιών (quality of service, QoS), η οποία είναι σημαντική για την τηλεφωνία καθώς και για άλλη κίνηση που είναι ευαίσθητη στην καθυστέρηση. Το COPS επιτρέπει την επικοινωνία του QoS και άλλων πληροφοριών διαχείρισης κίνησης μεταξύ ενός policy decision point (PDP) σε ένα δίκτυο και των policy enforcement points (PEPS). Η διανομή της κίνησης πραγματοποιείται σύμφωνα με τις επιθυμητές προτεραιότητες της υπηρεσίας.

3. SIP ΠΡΩΤΟΚΟΛΛΟ

3.1. Γενικά

Το SIP, συντομογραφία του Session Initiation Protocol (Πρωτόκολλο εκκίνησης συνόδου), είναι ένα πρωτόκολλο σηματοδοσίας τηλεφωνίας IP που χρησιμοποιείται για την πραγματοποίηση, την τροποποίηση και τον τερματισμό τηλεφωνικών κλήσεων VOIP. Το SIP δημιουργήθηκε από τον οργανισμό IETF και δημοσιεύτηκε ως RFC 3261. Το SIP περιγράφει την επικοινωνία που χρειάζεται για την πραγματοποίηση μιας τηλεφωνικής κλήσης. Το SIP πρωτόκολλο μοιάζει με το HTTP, βασίζεται σε κείμενο και είναι πολύ ανοιχτό και ευέλικτο.

3.2. Sip μηνύματα

Υπάρχουν δύο διαφορετικά είδη sip μηνυμάτων: "Requests and Responses", αιτήματα και απαντήσεις αντίστοιχα. Στην πρώτη γραμμή κάθε sip μηνύματος καθορίζεται η φύση του αιτήματος, μαζί με τον προορισμό που θα ακολουθήσει το αίτημα. Το RFC 3261 καθορίζει τις ακόλουθες μεθόδους που αποτελούν τη λίστα των μεθόδων για τα Sip αίτήματα(requests):

- Register: Χρησιμοποιείται για να μπορεί ο χρήστης να εγγραφεί στην υπηρεσία που επιθυμεί.
- Invite: Χρησιμοποιείται για να μπορούν οι χρήστες μιας υπηρεσίας να επικοινωνήσουν
- Ack: Επιβεβαιώνει αξιόπιστες ανταλλαγές μηνυμάτων.
- Cancel: Διακόπτει ένα αίτημα που εκκρεμεί.
- Bye: Διακόπτει μια σύνοδος μεταξύ δύο χρηστών σε μια διάσκεψη.
- Options: Αιτήσεις πληροφοριών σχετικά με τις δυνατότητες ενός καλούντος, χωρίς τη δημιουργία μιας κλήσης.

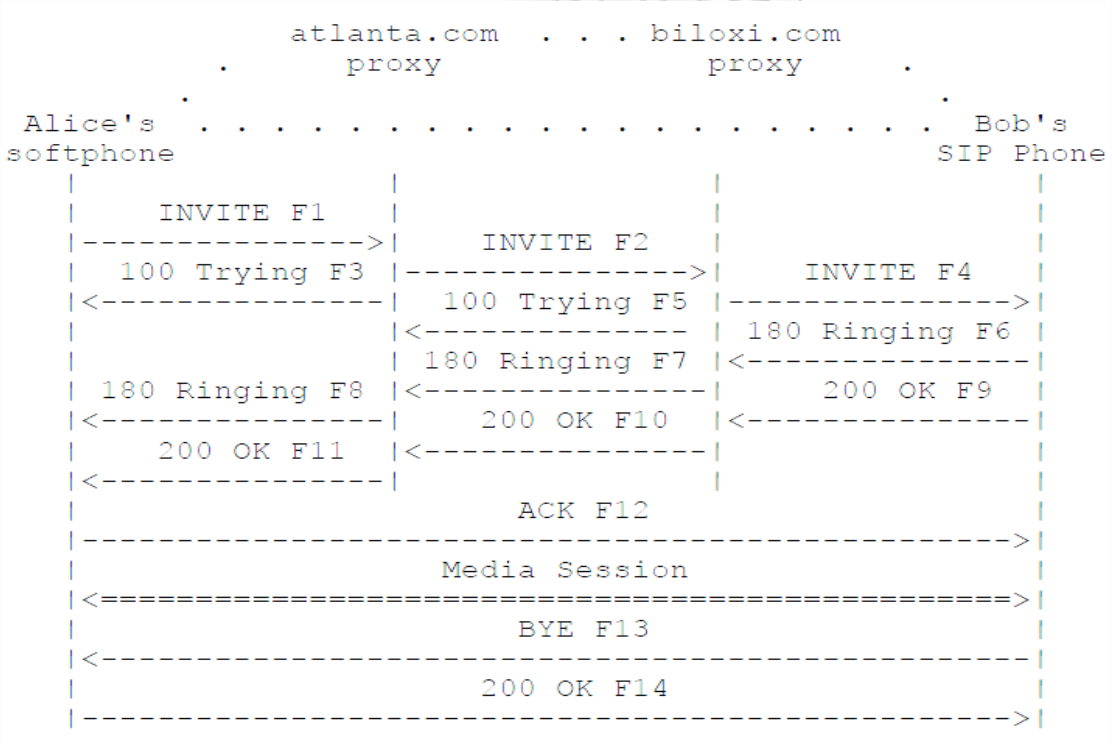
Από την άλλη, το RFC 3261 καθόρίζει επίσης τα είδη των Sip απαντήσεων (responses):

- Provisional (1xx): Ελήφθει το αίτημα και βρίσκεται υπό επεξεργασία.
- Success (2xx): Η ενέργεια ελήφθει με επιτυχία, έγινε κατανοητή και αποδεκτή.
- Redirection (3xx): Πρέπει να ληφθούν περαιτέρω μέτρα για να συμπληρωθεί το αίτημα.

- Client Error (4xx): Το αίτημα περιλαμβάνει κακή σύνταξη ή δεν μπορεί να εγκριθεί από το server.
- Server Error (5xx): Ο server παρέλειψε να εκπληρώσει ένα φαινομενικά έγκυρο αίτημα.
- Global Failure (6xx): Το αίτημα δεν μπορεί να εκπληρωθεί από τον οποιοδήποτε server.

3.2.1. Διαδικασία ανταλλαγής μηνυμάτων

Ένα τυπικό παράδειγμα ανταλλαγής μηνυμάτων SIP είναι μεταξύ δύο χρηστών της Alice και του Bob. Στο παράδειγμα, η Alice χρησιμοποιεί το διαδίκτυο και μία εφαρμογή SIP(που αναφέρεται ως “softphone” για να καλέσει στο τηλέφωνο SIP του Bob. Επίσης παρουσιάζονται δύο πληρεξούσιοι εξυπηρετητές SIP, οι οποίοι ενεργούν για λογαριασμό των Alice και Bob, ώστε να διευκολύνουν την αποκατάσταση της συνόδου.



ΕΙΚΟΝΑ 4: ΑΠΛΟΥΣΤΕΥΜΕΝΟ ΠΑΡΑΔΕΙΓΜΑ ΠΡΑΓΜΑΤΟΠΟΙΗΣΗΣ ΣΥΝΟΔΟΥ

Η Alice «καλεί» τον Bob χρησιμοποιώντας την ταυτότητα SIP του Bob, το οποίο αναφέρεται ως Uniform Resource Identifier - URI, που καλείται SIP-URI. Ο SIP-URI έχει αντίστοιχη μορφή με μία διεύθυνση ηλεκτρονικού ταχυδρομείου, περιέχοντας συνήθως ένα όνομα χρήστη και ένα όνομα εξυπηρετητή. Στο παράδειγμα, το SIP-URI

του Bob είναι “sip:bob@biloxi.com”, όπου “biloxi.com” είναι ο εξυπηρετητής υπηρεσιών SIP του Bob. Επειδή το τηλέφωνο λογισμικού δεν γνωρίζει τη φυσική θέση του Bob ή τον εξυπηρετητή SIP, στέλνει το αίτημα “INVITE” στον πληρεξούσιο εξυπηρετητή SIP που εξυπηρετεί τον τομέα της Alice, δηλαδή στον “atlanta.com”.

Στο παράδειγμα αυτό, ο πληρεξούσιος εξυπηρετητής λαμβάνει το αίτημα “INVITE” και επιστρέφει μία απάντηση “Trying” στο τηλέφωνο λογισμικού της Alice. Η απάντηση “Trying” δηλώνει ότι το αίτημα “INVITE” έχει ληφθεί και ο πληρεξούσιος εξυπηρετητής προσπαθεί να το δρομολογήσει στον προορισμό του για λογαριασμό της Alice. Ο πληρεξούσιος εξυπηρετητής “atlanta.com” χρησιμοποιώντας ένα συγκεκριμένο τύπο αναζήτησης, DNS (Domain Name Service), εντοπίζει τον πληρεξούσιο εξυπηρετητή SIP που υποστηρίζει τον τομέα “biloxi.com”. Στη συνέχεια προωθεί το αίτημα “INVITE” στον πληρεξούσιο εξυπηρετητή του “biloxi.com”.

Ο πληρεξούσιος εξυπηρετητής του “biloxi.com” λαμβάνει το αίτημα “INVITE” και ανταποκρίνεται με μία απάντηση με κωδικό 100 (Trying) στον πληρεξούσιο εξυπηρετητή του “atlanta.com” για να του δείξει ότι έχει λάβει το αίτημα “INVITE” και το επεξεργάζεται. Ο εξυπηρετητής αυτός συμβουλεύεται μία βάση δεδομένων, η οποία περιλαμβάνει την τρέχουσα διεύθυνση IP του Bob. Το τηλέφωνο SIP του Bob λαμβάνει το αίτημα “INVITE” και ενημερώνει το Bob για την εισερχόμενη κλήση από την Alice, ώστε να αποφασίσει εάν θα απαντήσει ή όχι την κλήση, δηλαδή το τηλέφωνο του Bob χτυπά.

Το τηλέφωνο SIP δείχνει ότι αυτή είναι μία απόκριση “Ringing”, η οποία δρομολογείται μέσω των δύο πληρεξουσίων εξυπηρετητών στην αντίστροφη κατεύθυνση. Κάθε πληρεξούσιος εξυπηρετητής χρησιμοποιεί το πεδίο κεφαλίδας “Via” για να καθορίσει το πού θα στείλει την απόκριση και αφαιρεί τη δική του διεύθυνση από την κορυφή. Ως αποτέλεσμα, ενώ για τη δρομολόγηση του αρχικού αιτήματος “INVITE” απαιτήθηκε η χρήση του DNS και της υπηρεσίας εντοπισμού, η απόκριση (Ringing) μπορεί να επιστραφεί στον καλούντα απευθείας. Λαμβάνοντας την απόκριση (Ringing), το τηλέφωνο λογισμικού της Alice ενημερώνει την Alice με την παρουσίαση σχετικού μηνύματος στην οθόνη της.

Όταν ο Bob αποφασίσει να απαντήσει στην κλήση, σηκώνει το ακουστικό και το SIP τηλέφωνό του στέλνει μία απόκριση “OK” (με κωδικό 200) για να δείξει ότι η κλήση απαντήθηκε.

3.3. Δομή SIP Μηνύματος

Στο παράδειγμα αυτό, η συναλλαγή ξεκινά με το τηλέφωνο λογισμικού της Alice να στέλνει ένα αίτημα πρόσκλησης "INVITE" προς τον SIP-URI του Bob. Το αίτημα "INVITE" είναι ένα SIP μήνυμα το οποίο έχει μια συγκεκριμένη δομή όπως το παρακάτω παράδειγμα.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;
        branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;
        tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

ΕΙΚΟΝΑ 5: SIP ΜΗΝΥΜΑ – ΑΙΤΗΜΑ INVITE

Η πρώτη γραμμή του μηνύματος περιλαμβάνει το όνομα της μεθόδου INVITE.

Το *πεδίο "Via"* περιλαμβάνει τη διεύθυνση (pc33.atlanta.com) στην οποία η Alice περιμένει τις αποκρίσεις στο αίτημα της ("INVITE"). Επιπλέον περιλαμβάνει μία παράμετρο κλάδου (branch parameter), η οποία ταυτοποιεί αυτή τη συναλλαγή.

Το *πεδίο "Max-Forwards"* χρησιμοποιείται για να περιορίσει τον αριθμό των αλμάτων (hops) που επιτρέπεται να κάνει ένα αίτημα μέχρις ότου φτάσει στον προορισμό του. Αποτελείται από έναν ακέραιο αριθμό που αυξάνει κατά ένα σε κάθε άλμα.

Το *πεδίο "To"* περιέχει το όνομα παρουσίασης (display name) του παραλήπτη (Bob) και έναν SIP-URI ή SIPS-URI (sip:bob@biloxi.com), στον οποίο απευθύνεται το αίτημα.

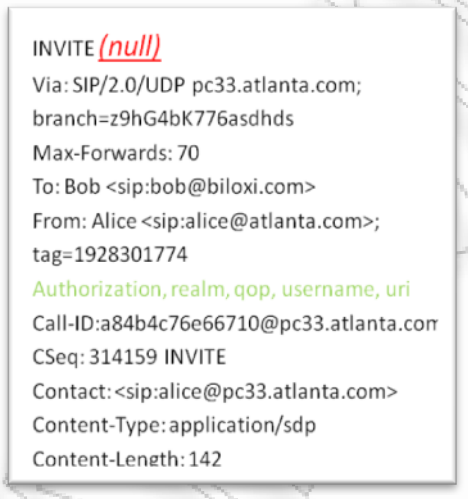
Το *πεδίο "From"* περιέχει το όνομα παρουσίασης του αποστολέα (Alice) και έναν SIP-URI ή SIPS-URI (sip:alice@atlanta.com), που παρουσιάζουν την αφετηρία του αιτήματος. Διαθέτει επίσης μία μικρή ετικέτα (tag parameter) που εξυπηρετεί σκοπούς ταυτοποίησης. Η ετικέτα περιλαμβάνει μια τυχαία στοιχειοσειρά (string) (1928301774) η οποία προστέθηκε στον SIP-URI από το τηλέφωνο λογισμικού.

Το *πεδίο "Call-ID"* περιέχει ένα μοναδικό αναγνωριστή (globally unique identifier) για την κλήση αυτή, που παράγεται από το συνδυασμό μιας τυχαίας στοιχειοσειράς και του ονόματος του εξυπηρετητή ή της διεύθυνσης IP του τηλεφώνου λογισμικού.
Το *πεδίο "Command Sequence - CSeq"* περιλαμβάνει έναν ακέραιο αριθμό, ο οποίος δείχνει τη σειρά της τρέχουσας εντολής, καθώς επίσης και το όνομα της μεθόδου ("INVITE" στο παράδειγμα) που περιγράφει το αίτημα.
Το *πεδίο "Contact"* περιέχει έναν SIP-SIP ή SIPS-URI, ο οποίος αναπαριστά την άμεση διαδρομή επικοινωνίας με τον αποστολέα του αιτήματος (Alice).
Το πεδίο *"Content-Type"* περιλαμβάνει την περιγραφή του σώματος του μηνύματος (δεν παρουσιάζεται στο παράδειγμα).
Το *πεδίο "Content*-Length" περιλαμβάνει τον αριθμό των bytes που περιέχονται στο σώμα του μηνύματος.

3.4. Malformed Μηνύματα

Ένα μήνυμα SIP μπορεί να είναι είτε μια αίτηση είτε επιβεβαίωση σε αντίστοιχο αίτημα, αποτελούμενο από το πεδίο της επικεφαλίδας και από το κύριο σώμα του μηνύματος. Τα SIP μηνύματα είναι βασισμένα σε κείμενο και μοιάζουν με την οργάνωση του HTTP, αλλά με μεγαλύτερο βαθμό ελευθερίας, επομένως ένας αποδοτικός συντακτικός αναλυτής είναι αναγκαίος ώστε να αναλύει την σημαντική πληροφορία των μηνυμάτων. Παρόλα αυτά, ακόμα και ένα τέλειο και έγκυρο μήνυμα SIP μπορεί να δομηθεί με τέτοιο τρόπο ώστε να παρακωλύει την ορθή συντακτική του ανάλυση. Ένα τέτοιο σενάριο αφορά την περίπτωση στην οποία ο επιτιθέμενος δημιουργεί μεγάλου μήκους μηνύματα με πολλαπλά πεδία επικεφαλίδας και αυξημένου μήκους καθώς και μεγάλου μεγέθους κύριο σώμα. Όλα τα μηνύματα SIP μπορούν να εμπεριέχουν σώμα ακόμα και όταν δεν χρειάζονται σε κάθε μήνυμα. Αυτό έχει ως αποτέλεσμα όχι μόνο τη μείωση της επεξεργαστική ισχύς αλλά και την αύξηση της εκμετάλλευσης του δικτύου.
Τα υποσυστήματα του IMS έχουν σχεδιαστεί και αναπτυχθεί για να επεξεργάζονται μηνύματα τα οποία είναι έγκυρα και συμβατά με τη σύνταξη του πρωτοκόλλου SIP. Παρόλα αυτά, εξαιτίας ορισμένων λαθών υλοποίησης και λαμβάνοντας υπόψη την πολυπλοκότητα του SIP συντακτικού αναλυτή, σε ορισμένες περιπτώσεις ακόμα και ορθώς σχηματισμένα μηνύματα μπορούν να οδηγήσουν τον SIP εξυπηρετητή σε κατάρρευση.

Εκτός από τα λάθη υλοποίησης, είναι πιθανό ο επιτιθέμενος να δοκιμάσει διάφορους συνδυασμούς κακοσχηματισμένων μηνυμάτων για να εντοπίσει προβλήματα ή λάθη ασφαλείας στο υποσύστημα. Για παράδειγμα το INVITE μήνυμα είναι λανθασμένο και δεν μπορεί να παραχθεί με βάση την καθορισμένη σύνταξη του SIP πρωτοκόλλου, εξαιτίας της έλλειψης του πεδίου REQUEST-URI, το οποίο θα πρέπει να είναι ορισμένο. Κάθε μήνυμα που δεν συμμορφώνεται ή παραβιάζει τις προδιαγραφές του πρωτοκόλλου SIP μπορεί να προκαλέσει αδυναμίες στην ασφάλεια του IMS υποσυστήματος, ενώ συνήθως είναι δύσκολο να διαχωριστούν τα νόμιμα από τα παράνομα μηνύματα. Πιθανόν να υπάρχουν σημεία τα οποία δεν έχουν ληφθεί υπόψη κατά την υλοποίηση της στοίβας SIP σε κάθε SIP προϊόν.



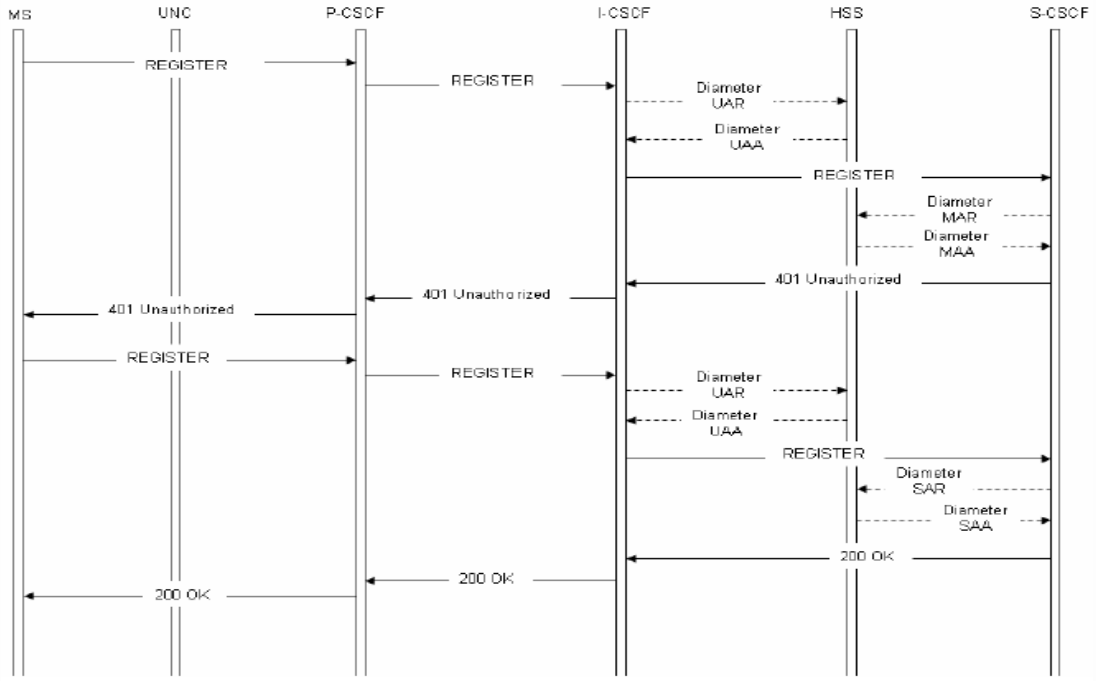
ΕΙΚΟΝΑ 6: MALFORMED SIP MESSAGE

3.5. Εγγραφή χρήστη

Το αποτέλεσμα της εξέλιξης του SIP για εφαρμογή του σε ασύρματα δίκτυα οδήγησε στον ορισμό ενός συνόλου από προϋποθέσεις που προσαρμόζουν το SIP στα 3GPP δίκτυα. Η εφαρμογή λύσεων για τις προϋποθέσεις αυτές σε ότι αφορά τα ασύρματα δίκτυα οδήγησε το 3GPP να καθορίσει επιλογές και επεκτάσεις του SIP και σε άλλα πρωτόκολλα.

Από τη στιγμή που το IMS τερματικό έχει ολοκληρώσει τις διαδικασίες πρόσβασης σε ένα IP δίκτυο παροχής πρόσβασης, έχει αποκτήσει διεύθυνση IPv6 και έχει ανακαλύψει την IPv6 διεύθυνση του P-CSCF (Proxy-CSCF) του, τότε μπορεί το τερματικό να εγγραφεί στο επίπεδο του IMS. Στα πλαίσια της διαδικασίας εγγραφής στο IMS ο χρήστης ζητά εξουσιοδότηση για την χρήση των υπηρεσιών του IMS. Το δίκτυο πιστοποιεί και εξουσιοδοτεί τον χρήστη ώστε να έχει πρόσβαση στο IMS. Ο

χρήστης συνδέει την δημόσια URI (Uniform Resource Identifier) διεύθυνσή του με την URI διεύθυνση που περιέχει το όνομα του κόμβου ή την IP διεύθυνση του τερματικού, όπου ο χρήστης είναι καταχωρημένος. Σε αντίθεση με την συνήθη SIP διαδικασία η Εγγραφή με το IMS υποχρεωτικά πρέπει να ολοκληρωθεί προτού το τερματικό μπορέσει να εγκαταστήσει μια σύνοδο. Η διαδικασία χρησιμοποιεί την αίτηση SIP REGISTER, αλλά για να ικανοποιούνται τα κριτήρια του 3GPP για ελάχιστο αριθμό κύκλων, η διαδικασία είναι υπερφορτωμένη. Ο στόχος είναι να μπορεί να ολοκληρωθεί σε δύο κύκλους όπως φαίνεται παρακάτω.



ΕΙΚΟΝΑ 7: ΕΓΓΡΑΦΗ ΣΤΟ IMS

Έτσι αφού ο καλών εγγραφεί στο IMS μέσω του P-CSCF εξουσιοδοτείται και πιστοποιείται. Αυτό επιτυγχάνεται με την ανταλλαγή και σύγκρισή μυστικών κλειδιών μεταξύ χρήστη και του HSS στο δίκτυο-έδρα του χρήστη. Όμως ο HSS δεν είναι σε θέση να καταλάβει μηνύματα SIP και για αυτό, το ρόλο του παρόχου πιστοποίησης αναλαμβάνει ο S-CSCF (Serving).

Ο κατάλληλος S-CSCF υποδεικνύεται από τον I-CSCF (Interrogating) βάση των στοιχείων που περιέχονται στο HSS για τον χρήστη. Επίσης ο I-CSCF παρέχει τις κατευθύνσεις που θα χρησιμοποιηθούν για την πιστοποίηση. Η νέα σύνοδος εγκαθίσταται με την επιβεβαίωση των στοιχείων πιστοποίησης μεταξύ χρήστη και S-CSCF.

3.6. Μηχανισμοί αυθεντικοποίησης

Οι προδιαγραφές του πρωτοκόλλου SIP δεν περιλαμβάνουν κάποιο συγκεκριμένο μηχανισμό ασφαλείας. Εξαιτίας αυτού έχει προταθεί η χρήση άλλων γνωστών μηχανισμών ασφαλείας του Internet. Η ασφάλεια στο SIP μπορεί να επιτευχθεί είτε από βήμα σε βήμα είτε από άκρη σε άκρη κατά τη διάρκεια μιας επικοινωνίας. Συγκεκριμένα, οι μηχανισμοί αυθεντικοποίησης που υπάρχουν στον OpenIMS είναι:

- HTTP Digest (ETSI)
  - Αμοιβαία αυθεντικοποίηση
  - Αυθεντικοποίηση με την χρήση username ενός password
  - Αντιστοιχεί την IP διεύθυνση ενός χρήστη στο αυθεντικοποιημένο URI και στην ανάλογη εγγραφή UPI
- Digest – AKA (3GPP)
  - Αμοιβαία αυθεντικοποίηση
  - Χρήση IPsec tunnel για αμοιβαία αυθεντικοποίηση
- Digest – MD5 (FOKUS)
- SIP Digest (3GPP)
- Early – IMS (3GPP)
- NASS Bundled (ETSI)

4. ΑΔΥΝΑΜΙΕΣ ΣΤΟ SIP-IMS

4.1. Γενικά

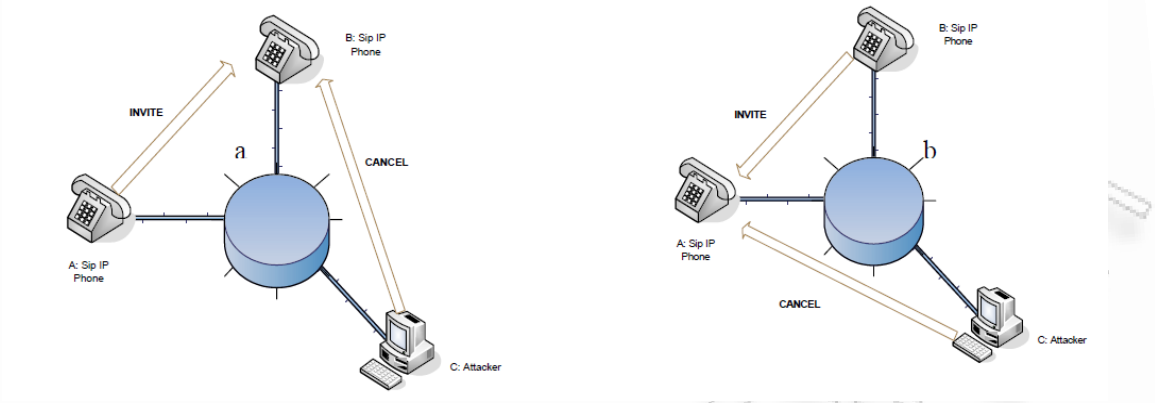
Το SIP είναι ένα βασικό πρωτόκολο για δίκτυα real time επικοινωνίας, συμπεριλαμβανομένων VoIP, IMS και IPTV δικτύων. Είναι βασισμένο σε IP οπότε είναι εξίσου ευπαθή σε επιθέσεις Denial-of-Service που δέχονται οι SIP servers.

4.2. SIP Bombing

Η επίθεση αυτή περιλαμβάνει τη διαβίβαση μιας μεγάλης ποσότητας από πλαστά SIP μηνυμάτα σε ένα στοχοθετημένο σύστημα. Η τηλεφωνία μέσω IP είναι πολύ ευαίσθητη σε θέματα διαθεσιμότητας. Μια μεγάλη ποσότητα από ψεύτικα SIP μηνύματα απαιτεί την κατανομή των υπολογιστικών πόρων για την αποκωδικοποίηση και την ερμηνεία τους. Καθώς το σύστημα είναι απασχολημένο με την επεξεργασία και την αντιμετώπιση των ψεύτικων μηνυμάτων, τα έγκυρα μηνύματα θα επεξεργάζονται με χαμηλότερο ρυθμό από τον οποίο θα έπρεπε να επεξεργάζονται από το σύστημα. Αυτός ο τύπος επίθεσης έχει ήδη δοκιμαστεί με επιτυχία σε VoIP συσκευές. Hardware IP τηλέφωνα έχουν συντριβεί στην έντονη κυκλοφορία SIP μηνυμάτων στο δίκτυο.

4.3. SIP-Cancel/Bye/Re-Invite/Update DoS

Εδώ έχουμε να αντιμετωπίσουμε ξεχωριστές κακόβουλες ενέργειες. Βρίσκονται στην ίδια κατηγορία διότι το προσχέδιο για την επίθεση είναι παρόμοιο για όλες τις περιπτώσεις. Ένας επιτιθέμενος στέλνει ένα SIP μήνυμα στο θύμα με το οποίο φαίνεται ότι το θύμα πραγματοποίησε μια επικοινωνία με έναν άλλο χρήστη. Από τη στιγμή λοιπόν που ο επιτιθέμενος έχει στην διάθεση του χρήσιμες πληροφορίες για την επικοινωνία των δύο χρηστών μπορεί να στείλει ότι αίτημα θέλει και να προκαλέσει τα αντίστοιχα για την κάθε μία αίτηση αποτελέσματα.



ΕΙΚΟΝΑ 8: SIP – CANCEL/BYE ΕΠΙΘΕΣΗ

4.4. SIP based Man in the Middle/Call Hijacking

Δεδομένου ότι οι δύο αυτές επιθέσεις έχουν κοινές συνέπειες και τα προσχέδια τους είναι κοινά, θα τα αντιμετωπίσουμε ως μια οντότητα. Μόλις μια κλήση κλαπεί είναι εύκολο να οδηγηθούμε στον κανονικό καλούντα, με αποτέλεσμα μιας Man in the middle επίθεσης. Δύο βασικές κατηγορίες man in the middle βασισμενες στο SIP μπορούν να διακριθούν. **Η χειραγώγηση των πληροφοριών για τις εγγραφές** (Manipulation of Registration Records) και οι επιθέσεις **βασισμένες στον 3xx κώδικα απάντησης** που υποδηλώνει στον server να προβεί σε διαδικασίες ανακατεύθυνσης.

4.4.1. Η χειραγώγηση των πληροφοριών για τις εγγραφές

Η επίθεση αυτή δίνει τη δυνατότητα στον επιτιθέμενο,συνήθως με βάση το ίδιο δίκτυο με αυτό του θύματος, να λαμβάνει όλες τις κλήσεις του θύματος με την χειραγώγηση της ταυτότητας του θύματος όπως αυτή έχει ανακοινωθεί στο δίκτυο (SIP URI). Ο καταχωρητής αξιολογεί την ταυτότητα του χρήστη από την κεφαλίδα “from” η οποία μπορεί εύκολα να τροποποιηθεί με συνέπεια να υπάρχει κακόβουλη καταχώρηση. Όταν χρησιμοποιείται απλώς το UDP πρωτόκολο δεν υάρχει κάποια δυσκολία για να επιτευχθεί μια τέτοιου είδους επίθεση. Όταν χρηστιμοποιείται μηχανισμός αυθεντικοποίησης (MD5 digest), και πάλι δεν είναι δυνατό να αποτραπεί μια τέτοια επίθεση μόνο από ένα username-password, αφού μπορεί να υάρχουν και εξωτερικοί επιτιθέμενοι που είναι σε θέση να εγγραφούν με επιτυχία στο σύστημα, Παρόλα αυτά η τελευταία περίπτωση είναι αξαιρετικά απίθανη.

4.4.2. Επίθεση βασισμένη σε 3xx κώδικα απάντησης

Ο κώδικας 3xx στο SIP ενημερώνει τον ενδιαφερόμενο χρήστη ότι χρειάζεται να γίνουν περαιτέρω ενέργειες για να ολοκληρωθεί επιτυχώς το αίτημα. Οι επίθέσεις τέτοιου είδους επικαλούνται πλαστές απαντήσεις. Το περίγραμμα της επίθεσης είναι το εξής: το θύμα εκδίδει ένα αίτημα SIP (invite αίτημα). Ο επιτιθέμενος στέλνει ένα 3xx μηνυμα στον χρήστη με σκοπό να ανακατευθύνει το αίτημα του χρήστη έτσι ώστε να περνάει η επικοινωνία μέσω του κακόβουλου.

4.5. SIP Μηνύματα – Επιθέσεις Πλημμύρας

Το SIP πρωτόκολο είναι ένα βασισμένο σε κείμενο, οπότε υπάρχουν αρκετά είδη επίθεσης μηνύματων πλημμύρας. Οι πιο σημαντικές από αυτές είναι: invite flooding, register flooding, register response flooding. Σύμφωνα με την πρώτη ο επιτιθέμενος στέλνει ένα μεγάλο αριθμό SIP μηνυμάτων με μια πλαστή IP διεύθυνση προς το θύμα αναγκάζοντας τον να καταναλώσει πόρων για την επεξεργασία των εισερχόμενων μηνυμάτων. Η επίθεση πλημμύρας register λειτουργεί παρόμοια με την προηγούμενη επίθεση, αλλά χρησιμοποιεί αντί για invite μηνύματα, register μηνύματα. Σύμφωνα με την τελευταία επίθεση ο επιτιθέμενος στέλνει ένα μεγάλο ποσό register μηνύματα με λάθος τα διαπιστευτήριά του προς τον πληρεξούσιο SIP server για να το συντρίψει. Υπάρχουν τρείς κύριες πηγές που μπορούν να στοχευθούν από μια SIP επίθεση πλημμύρας: το bandwidth, η CPU και η μνήμη.

4.5.1. Bandwidth

Ο στόχος είναι πλημμυρισμένος με περισσότερα μηνύματα από ότι το δίκτυο μπορεί να χειριστεί, π.χ. ο εισβολέας καταφέρνει να δημιουργήσει μια επίθεση ποσοστό 10 GB / s, ενώ ο στόχος είναι συνδεδεμένος στο Διαδίκτυο μέσω γραμμής 1 GB / s. Αυτό βέβαια είναι ένα γενικό πρόβλημα πλημμυρών DoS και δεν είναι ειδικά για SIP δίκτυα.

4.5.2. CPU

Ο στόχος είναι πλημμυρισμένος με περισσότερα μηνύματα από ό, τι μπορεί να επεξεργαστεί σε μια δεδομένη χρονική στιγμή, καθότι το SIP είναι πρωτόκολλο βασισμένο σε κείμενο, πρέπει να αναλύσει κάθε εισερχόμενο μήνυμα.

4.5.3. Μνήμη

Πολλές SIP αιτήσεις δημιουργούν καταστάσεις λειτουργίας στον στόχο. Ένα INVITE μήνυμα στέλνεται στον proxy και ο proxy περιμένει μερικά δευτερόλεπτα για μια απάντηση. Κατά τη διάρκεια αυτής της κατάστασης καταναλώνεται μνήμη στον proxy. Εάν λοιπόν ο proxy αντιμετωπίζει μια πληθώρα τέτοιων μηνυμάτων κάποια στιγμή θα εξαντληθεί η μνήμη που έχει στη διάθεση του με αποτέλεσμα να υπολειτουργήσει αν όχι να καταρρεύσει. Μια πλημμύρα μπορεί να επιτευχθεί με διαφοτερικά SIP μηνύματα (INVITE,REGISTER…).

4.6. TCP SYN και TCP/ACKs Πλημμύρας

Η γνωστή κατηγορία των επιθέσεων επιτυχνάνει τους στόχους της, με το να δημιουργεί μισάνοιχτες συνδέσεις στο θύμα. Μια τέτοια κατάσταση εμφανίζεται όταν ο server στέλνει ένα μήνυμα SYN-ACK, αλλά ποτέ δεν λαμβάνει ένα μήνυμα ACK από τον χρήστη. Συγκεκριμένα, ο εισβολέας στέλνει ένα πλαστό SYN πακέτο με μία απρόσιτη IP διεύθυνση προέλευσης. Κατά την υποδοχή, το θύμα θα απαντήσει με ένα μήνυμα SYN-ACK, αλλά το δίκτυο δεν είναι σε θέση να το δρομολογήσει. Ως εκ τούτου, το θύμα δεν λαμβάνει ποτέ ACK μήνυμα που να ανταποκρίνεται στο δικό του SYN-ACK. Ακόμα χειρότερα, η μνήμη που διατίθενται για αναμονή της σύνδεσης μπορεί να αποδεσμευθεί μόνο όταν η TCP σύνδεση γίνει timeout. Μια παρόμοια επίθεση με TCP SYN είναι το TCP /ACKs πλημμύρα. Για την μεγιστοποίηση των συνεπειών μπορεί ο κακόβουλος να χρησιμοποιήσει και τις δύο επιθέσεις παράλληλα.

4.7. Επιθέσεις προς Αναλυτές Μηνυμάτων

Καθώς το SIP πρωτόκολλο είναι βασισμένο σε απλό κείμενο με ένα υψηλό βαθμό ελευθερίας, ένας αποτελεσματικός αναλυτής (parser), αναλύει τα μηνύματα που λαμβάνει μέχρι το σημείο των πληροφοριών που απαιτούνται. Ωστόσο, ακόμη και ένα πλήρες SIP μήνυμα μπορεί να κατασκευαστεί με τρόπο που να παρεμποδίζει την ορθή ανάλυση. Για παράδειγμα, ένας εισβολέας μπορεί να δημιουργήσει αδικαιολόγητα μεγάλα μηνύματα με απλό τρόπο, με την προσθήκη επιπλέον κεφαλίδων, σε συνδυασμό με ένα μεγάλο μήνυμα-σώματος. Έτσι αντί να καταστρέφουν μόνο την ενέργεια του επεξεργαστή, τα μεγάλα μηνύματα αυξάνουν σε χρόνο την χρήση του δικτύου και του καταναλώνουν περισσότερη μνήμη.

4.7.1. Επιθέσεις που Αξιοποιούν Μη Συμβατά Μηνύματα

Είναι γνωστό ότι οι υλοποιήσεις πρωτοκόλλων και δικτυακών εφαρμογών σε πολλές περιπτώσεις δεν συμμορφώνονται πλήρως με τις προδιαγραφές που έχουν τεθεί ή εμπεριέχουν λάθη, τα οποία μπορεί να προκαλέσουν την αποστολή μη ορθών μηνυμάτων. Η επεξεργασία και ανάλυση των SIP μηνυμάτων είναι υψηλής σημασίας καθώς αποτελεί αναπόσπαστό τμήμα όλων των δικτυακών οντοτήτων του SIP. Οι περισσότεροι αναλυτές μηνυμάτων στο SIP έχουν σχεδιαστεί για την
επεξεργασία μηνυμάτων που είναι απόλυτα συμβατά με τις προδιαγραφές του. Σε ελάχιστες περιπτώσεις είναι δυνατόν να διαχειρισθούν κακόβουλα μηνύματα που δε συμμορφώνονται με τη γραμματική του SIP, απορρίπτοντας τα σε αρχικό στάδιο της επεξεργασίας. Η περαιτέρω επεξεργασία τέτοιων μηνυμάτων οδηγεί την αντίστοιχη SIP δικτυακή οντότητα σε μια από τις ακόλουθες μη επιθυμητές καταστάσεις :

1. Άρνηση παροχής υπηρεσίας (Denial of Service)-(DoS)
2. Μη σταθερή λειτουργία (Unstable operation)
3. Μη εξουσιοδοτημένη πρόσβαση (unauthorized access)

Η μορφή κειμένου που αξιοποιείται από τα SIP μηνύματα προσελκύει περισσότερους επιτιθέμενους που αναζητούν διαφορετικούς τρόπους για να προκαλέσουν μια εκ των προαναφερόμενων καταστάσεων. Για παράδειγμα, ένας επιτιθέμενος αντί να στείλει ένα SIP REGISTER μήνυμα βασισμένο στις προδιαγραφές του SIP μπορεί να αποστείλει ένα μη έγκυρο SIP REGISTER μήνυμα με λάθος κεφαλίδες ή με ελλιπή στοιχεία στις κεφαλίδες ακόμα και παραπλανητικές πληροφορίες στις κεφαλίδες με αποτέλεσμα να επιβαρύνει τους αναλυτές SIP μηνυμάτων. Τα μηνύματα αυτού του είδους που δύναται ένας επιτιθέμενος να δημιουργήσει είναι αναρίθμητα.

Για παράδειγμα, ο επιτιθέμενος μπορεί να ακολουθήσει τη μέθοδο εξαντλητικής αναζήτησης δημιουργώντας διαφορετικούς συνδυασμούς κακόβουλων μηνυμάτων για να αποκτήσει πρόσβαση στη διαδικτυακή οντότητα στόχο. Εναλλακτικά, ο επιτιθέμενος μπορεί να ελέγξει τη ρωμαλεότητα ενός SIP αναλυτή εντοπίζοντας τα μηνύματα τα οποία δεν υποστηρίζει, αποστέλλοντας του μη υποστηριζόμενα, μη συμβατά μηνύματα για τον έλεγχο της απόκρισης του σε αυτές τις περιπτώσεις.

Τα μηνύματα που υποστηρίζονται από μια δικτυακή SIP οντότητα εμπεριέχονται τόσο στα SIP REGISTER όσο και στα SIP OPTIONS μηνύματα. Συνεπώς, κάποιος επιτιθέμενος είτε θα υποκλέψει το SIP REGISTER κατά την αρχική διαδικασία εγγραφής του χρήστη στόχου, είτε θα αποστείλει στη συσκευή στόχο ένα μήνυμα SIP

OPTIONS, όπου στην απάντηση που θα παραλάβει θα υπάρχουν, μεταξύ των άλλων, και όλα τα υποστηριζόμενα μηνύματα.

4.7.2. Επιθέσεις που Αξιοποιούν Συμβατά Μηνύματα

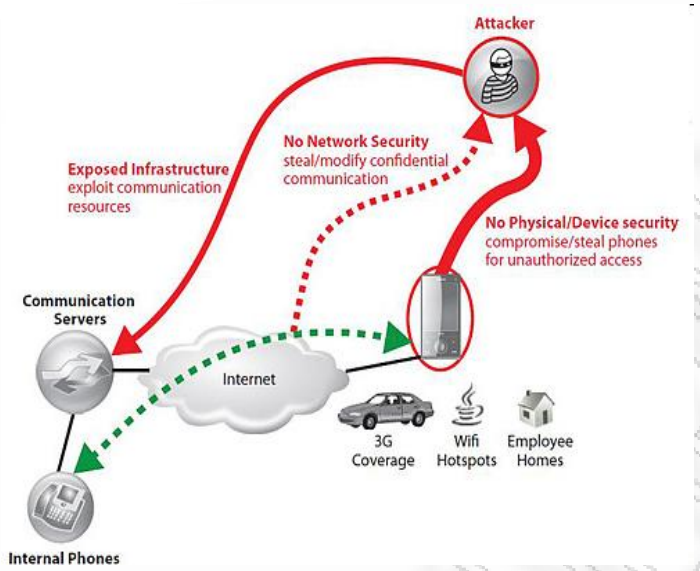
Ακόμα και ένα απολύτως συμβατό με τις προδιαγραφές του SIP μήνυμα μπορεί να δημιουργήσει προβλήματα αστάθειας στις δικτυακές οντότητες που το επεξεργάζονται. Πιο συγκεκριμένα, ο επιτιθέμενος μπορεί να δημιουργήσει μηνύματα μεγάλου μήκους προσθέτοντας κεφαλίδες οι οποίες δεν είναι αναγκαίες για την επεξεργασία του συγκεκριμένου αιτήματος ή/και συμπεριλαμβάνοντας πολλά μη αναγκαία δεδομένα στο κύριο σώμα του μηνύματος. Εναλλακτικά, ο επιτιθέμενος δύναται να δημιουργήσει μηνύματα τα οποία έχουν πολλαπλές τιμές οι οποίες διαχωρίζονται αντιστοίχως σε πολλαπλές κεφαλίδες, κάθε μία από τις οποίες περιέχει μια μοναδική τιμή. Τέτοια μηνύματα είναι απολύτως συμβατά με τις προδιαγραφές του SIP. Στην περίπτωση που οι κεφαλίδες δρομολόγησης (route, via …) τοποθετηθούν στο τέλος του μηνύματος είναι ξεκάθαρο ότι προκαλούν μεγαλύτερη επεξεργαστική πολυπλοκότητα, αφού οι απαραίτητες πληροφορίες για την ορθή δρομολόγηση του μηνύματος αναζητούνται στις αρχικές κεφαλίδες.

4.8. SQL injection

Αντιθέτα από όλες τις προαναφερθείσες επιθέσεις αυτή η επίθεση είναι ανεξάρτητη του χρόνου. Επιθέσεις με αλλοιωμένα μηνύματα σε εφαρμογές SIP είναι αρκετά εύκολες, λόγω της φύσης των SIP μηνυμάτων. Η SQL injection έχει ήδη αποδείξει την αποτελεσματικότητά της στο Διαδίκτυο, αλλά κατά κανόνα μπορεί να χρησιμοποιηθεί εναντίον κάθε εφαρμογής που κατασκευάζει και εκτελεί SQL δηλώσεις. Ο πρωταρχικός στόχος αυτής της επίθεσης είναι κακόβουλη τροποποίηση των δεδομένων, αλλά μπορεί επίσης να χρησιμοποιηθεί για να καταστρέψει τις υπηρεσίες δεδομένων που παρέχουν οι SIP proxies.

4.9. Υποκλοπές Κλήσεων στη Διαδικτυακή Τηλεφωνία

Μια από τις πιο γνωστές επιθέσεις που εκδηλώνεται σχεδόν σε όλα τα τηλεπικοινωνιακά συστήματα που υποστηρίζουν υπηρεσίες τηλεφωνίας είναι η υποκλοπή συνδιαλέξεων- κλήσεων (eavesdropping).



ΕΙΚΟΝΑ 9: ΥΠΟΚΛΟΠΗ ΣΥΝΔΙΑΛΕΞΕΩΝ- ΚΛΗΣΕΩΝ

Όλα τα μηνύματα σηματοδοσίας εμπεριέχουν πληροφορίες σχετικά με τα αναγνωριστικά των χρηστών, τις διευθύνσεις επαφής, κλειδιά ασφαλείας, καθώς και όλες τις βασικές παραμέτρους που απαιτούνται για την αποκατάσταση συνδέσεων μεταξύ δύο ή περισσότερων χρηστών. Όλες οι προαναφερόμενες πληροφορίες θα πρέπει να τηρούνται εμπιστευτικές. Παρ' όλα αυτά, η εύκολη πρόσβαση στο μέσο μετάδοσης, η ύπαρξη διαφόρων ειδών εργαλείων υποκλοπής δεδομένων στο διαδίκτυο, σε συνδυασμό με τα μηνύματα κειμένου που χρησιμοποιούνται στο SIP, καθιστά την διαδικασία υποκλοπής κλήσεων σχετικά απλή.

5. ΥΛΟΠΟΙΗΣΗ

5.1. Σκοπός της διπλωματικής εργασίας

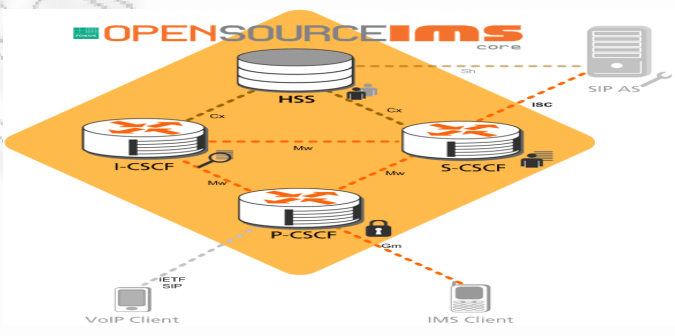
Για την παρούσα διπλωματική εργασία έγιναν μια σειρά από υλοποιήσεις με κύριο σκοπό την ασφάλεια σε υπηρεσίες διαδικτυακής τηλεφωνίας. Μελετήσαμε προσεκτικά την τεχνολογία ενός IMS δικτύου και τον τρόπο που λειτουργεί καθώς και ποίοι σκοποί επιτυγχάνονται με τη χρήση τέτοιων τεχνολογιών, θέματα που αναλύθηκαν στην δεύτερη ενότητα αυτής της διπλωματική εργασίας. Προβήκαμε λοιπόν σε μία σειρά από υλοποιήσεις στο εργαστηριακό περιβάλλον του Πανεπιστημίου Πειραιώς, που θα αναφέρουμε παρακάτω.

5.2. Υλοποίηση του OpenIMS CORE

Υλοποιήσαμε μια πλατφόρμα που ονομάζεται OpenIMS. Το Open IMS CORE είναι μια ανοιχτού κώδικα υλοποίηση για IMS συνόδους. Το OpenIMS CORE αποτελείται από τρις ξεχωριστούς servers (p-cscf,s-cscf,i-cscf) καθώς και έναν server βάσης δεδομένων(hss- home subscriber server). Όλα μαζί τα παραπάνω στοιχεία αποτελούν τα βασικά στοιχεία για όλες τις IMS/NGN αρχιτεκτονικές που προσδιορίζονται έως και σήμερα μέσα από 3GPP, 3GPP2, ETSI, TISPAN και PacketCable initiative. Σκοπός του OpenIMS παγκοσμίως είναι να δώσει την δυνατότητα στον κόσμο να μελετήσει την τεχνολογία του IMS, που όλο και περισσότερο χρησιμοποιείται για τις υπηρεσίες που παρέχει, και μέσα από τη έρευνα να καλυφθούν διάφορα κενά ασφάλειας που παρατηρούνται στον IMS.

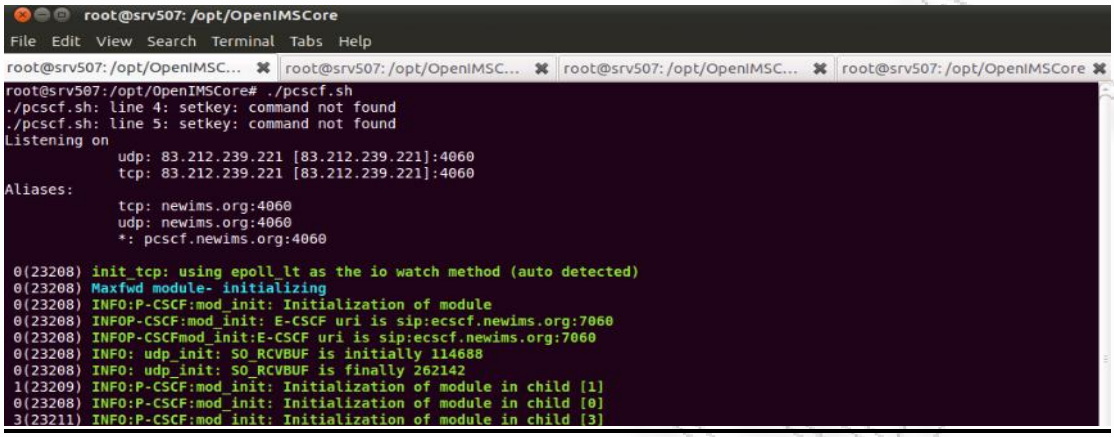
Πιο συγκεκριμένα για την υλοποίηση που πραγματοποιήσαμε. Στήσαμε στον εργαστηριακό χώρο του Πανεπιστημίου Πειραιώς τον OpenIMS πυρήνα που αποτελείται, από του συνολικά τέσσερις servers που αναφέραμε παραπάνω (p-cscf,s-cscf,i-cscf, fhoss).

Για την κατανόηση της υλοποίησης παραθέτουμε την παρακάτω εικόνα.



ΕΙΚΟΝΑ 10: SERVERS ΠΟΥ ΕΓΚΑΤΑΣΤΗΣΑΜΕ ΓΙΑ ΤΟ OPENIMS

- **P-CSCF**

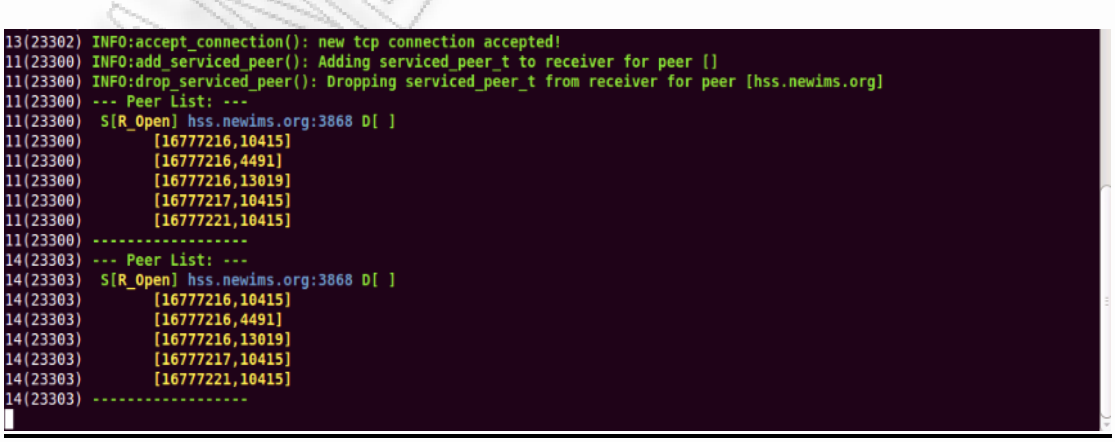


ΕΙΚΟΝΑ 11: ΣΕ ΛΕΙΤΟΥΡΓΙΑ Ο P-CSCF

- **S-CSCF**

ΕΙΚΟΝΑ 12: ΣΕ ΛΕΙΟΥΡΓΙΑ Ο S-CSCF

- **I-CSCF**

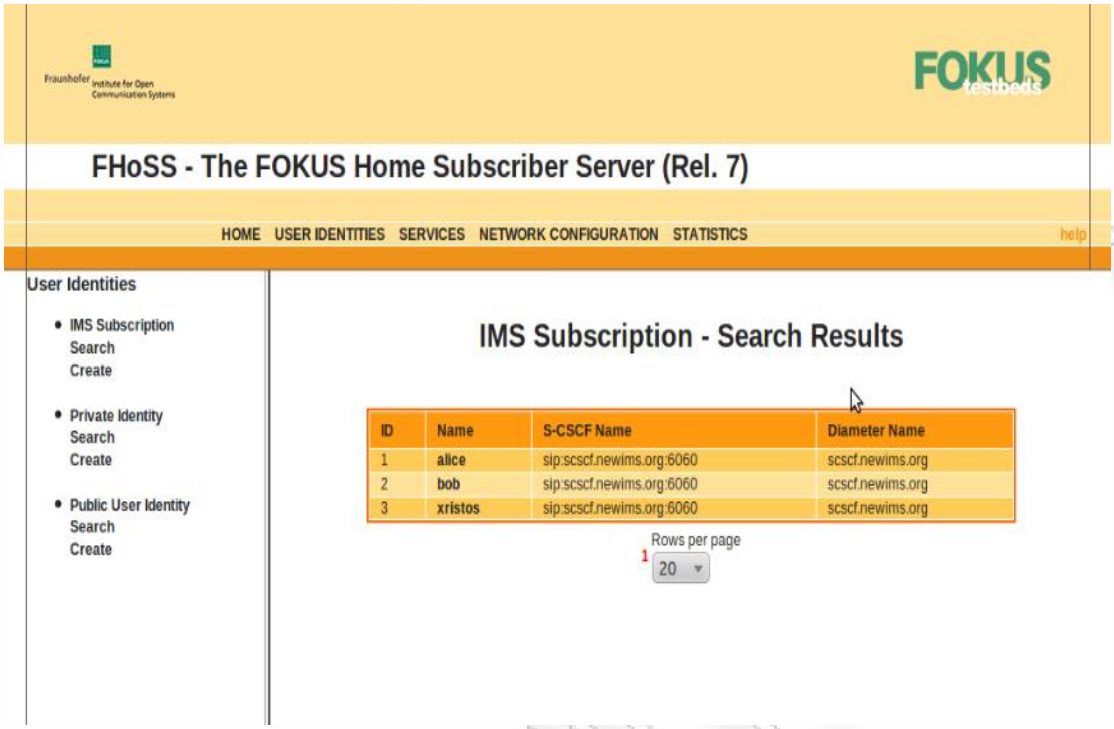


ΕΙΚΟΝΑ 13: ΣΕ ΛΕΙΟΥΡΓΙΑ Ο I-CSCF

- **fHoSS**

ΕΙΚΟΝΑ 14: ΣΕ ΛΕΙΟΥΡΓΙΑ Ο fHoSS

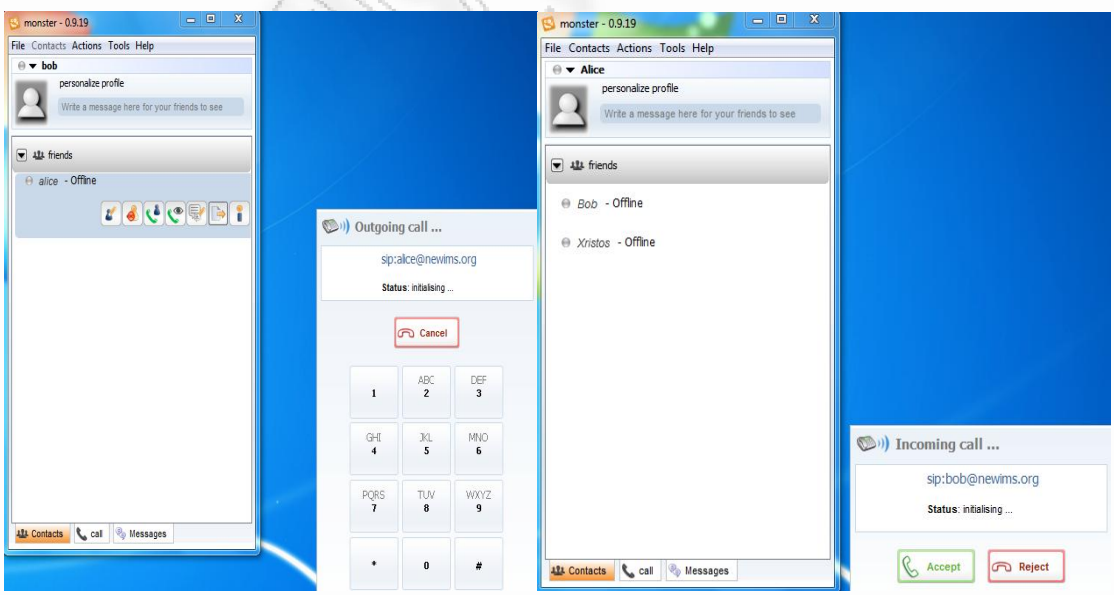
Έπειτα από τη στιγμή που στήσαμε του servers επεξεργαστήκαμε τα στοιχεία του OpenIMScore με σκοπό να δημιουργήσουμε τον δικό μας provider που θα παρέχει τις υπηρεσίες που μας ενδιαφέρουν. Έτσι δημιουργήσαμε την υπηρεσία “newims.org”. Κάναμε τα απαραίτητα configuration στους εγκατεστημένους servers με σκοπό να μην τρέχουν την default υπηρεσία “openims.test” αλλά στην θέση της να υπάρχει η υπηρεσία “newims.org”. Εγκαταστήσαμε και τον κατάλληλο DNS server ώστε να αναγνωρίζει IPs και ονομασίες και να κάνει τις αντιστοιχίες που είναι αναγκαίες. Στην συνέχεια μπήκαμε στο interface του fHoSS και ορίσαμε και εκεί την υπηρεσία “newims.org”. Ακολούθως κάναμε εγγραφές στην υπηρεσία έναν χρήστη με όνομα xristos όπου το SIP –URI είναι “sip:xristos@newims.org”. Επίσης κρατήσαμε και τους default χρήστες Alice και Bob με ταυτότητες “sip:alice@newims.org” και “sip:bob@newims.org” αντίστοιχα. Στην συνέχεια ορίσαμε με ποιους τρόπους θα αυθεντικοποιούνται οι χρήστες στην υπηρεσία καθώς και ποιες υπηρεσίες έχουν δικαίωμα να χρησιμοποιήσουν.



ΕΙΚΟΝΑ 15: INTERFACE ΤΟΥ FHOSS

5.3. Εγκατάσταση των ims-clients

Το επόμενο βήμα της υλοποίησης λοιπόν ήταν να εγκαταστήσουμε ims-clients προγράμματα (Monster client software) με τα οποία θα πραγματοποιήσουμε κλήσεις μεταξύ των επαφών μας.

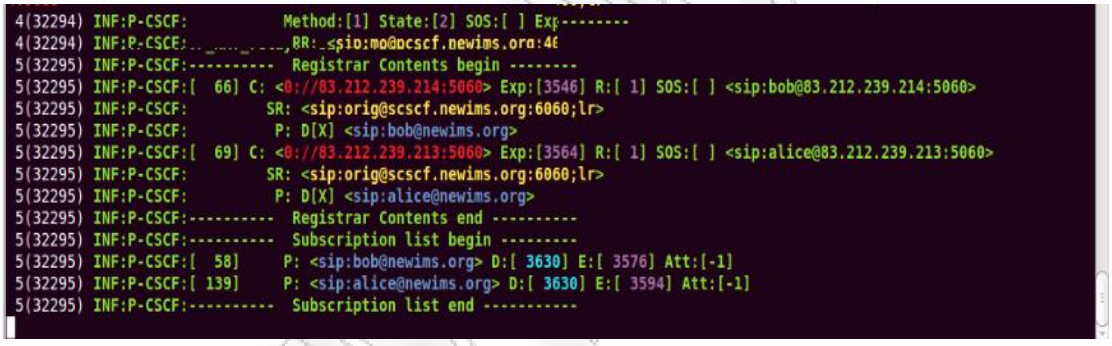


ΕΙΚΟΝΑ 16: IMS-CLIENT OUTGOING CALL/INCOMING CALL

Χρησιμοποιήσαμε εκτός από τον υπολογιστή που εγκαταστήσαμε τον OpenIMS ακόμα τρεις υπολογιστές όπου τους εγκαταστήσαμε τα ims-client λογισμικά (Monster client software) προκειμένου να επικοινωνίσουν μεταξύ τους. Έτσι λοιπόν ορίσαμε τους χρήστες: xristos, alice και bob στους τρεις υπολογιστές και κάθε επαφή είχε σαν επιλογές για επικοινωνία τους άλλους δύο. Για παράδειγμα στον υπολογιστή που υπήρχε η επαφή xristos στον ims-client, προσθέσαμε σαν φίλους του xristos και έτοιμους για επικοινωνία τις επαφές alice και bob. Έπειτα πραγματοποιήσαμε κλήσεις μεταξύ των επαφών για να βεβαιωθούμε ότι όλα λειτουργούν σωστά και επιτυχνάνεται η επικοινωνία μεταξύ των επαφών μας.

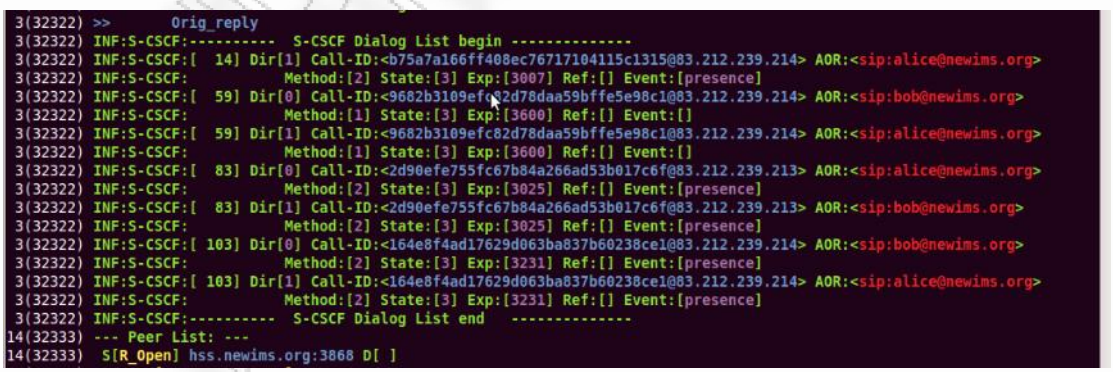
Οι παρακάτω εικόνες, είναι οι καταγραφές των sip-servers την ώρα που πραγματοποιήθηκε μια κλήση.

- P-CSCF



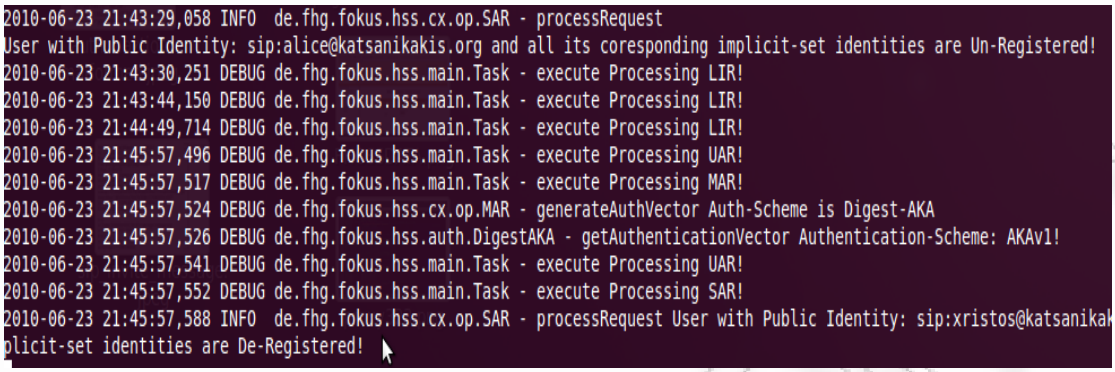
ΕΙΚΟΝΑ 17: ΚΑΤΑΓΡΑΦΕΣ ΣΤΟΝ P-CSCF

- S-CSCF



ΕΙΚΟΝΑ 18: ΚΑΤΑΓΡΑΦΕΣ ΣΤΟΝ S-CSCF

- FHoSS



ΕΙΚΟΝΑ 19: ΚΑΤΑΓΡΑΦΕΣ ΣΤΟΝ fHoSS

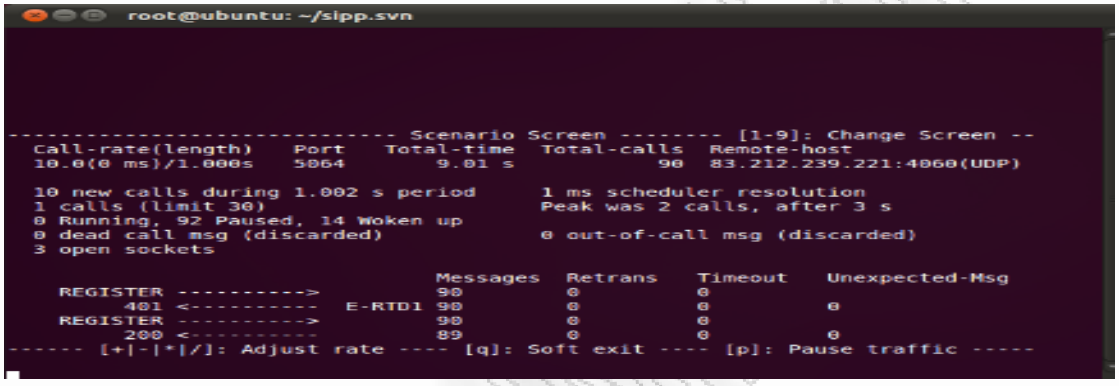
5.4. Δημιουργία SIP μηνυμάτων

Στη συνέχεια ασχοληθήκαμε με τα sip μηνύματα που ανταλλάσσονται μεταξύ των επαφών και των επαφών με τον server του OpenIMS με στόχο να μελετήσουμε τον ακριβή τρόπο επικοινωνίας ενός τέτοιου δικτύου. Έχουμε αναφέρει σε προηγούμενη ενότητα τη μορφή που έχουν τα sip μηνύματα κάπως και την σειρά των μηνυμάτων που ανταλλάσσονται μεταξύ επαφής – server – επαφής για να πραγματοποιηθεί για παράδειγμα μία κλήση μεταξύ των επαφών. Αφού λοιπόν κατανοήσαμε τη σειρά των sip μηνυμάτων που ανταλλάσσονται μέσα στο δίκτυο, δημιουργήσαμε τα δικά μας sip σενάρια με τις επαφές που είχαμε δημιουργήσαμε. Με αυτή μας την κίνηση θέλαμε να ξεφύγουμε από το γραφικό περιβάλλον ενός ims-client λογισμικού και να εισχωρήσουμε ένα βήμα πιο μέσα στο δίκτυο. Έτσι αφού φτιάξαμε τα sip σενάρια μας, τα στέλναμε στο δίκτυο και το δίκτυο ανταποκρινόταν αναλόγως με τα requests-responses που δεχότανε με ένα ειδικό πρόγραμμα όπως ονομάζεται “sipp”.

5.5. Δημιουργία επικοινωνίας μέσω SIP μηνυμάτων-σεναρίων

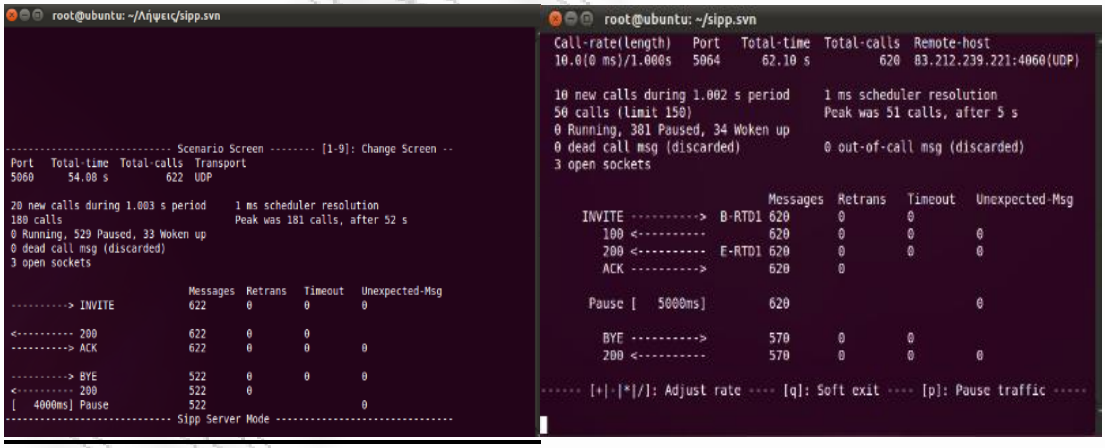
Το “sipp” είναι ένα πρόγραμμα το οποίο διαβάζει sip μηνύματα και μας δίνεται η ευκαιρία μέσα από τα sip σενάρια μας να δημιουργήσουμε μία επικοινωνία μεταξύ των επαφών μας. Για την καλύτερη κατανόηση παραθέτονται οι παρακάτω εικόνες. Αρχικά για να είναι σε θέσει κάποιος χρήστης να χρησιμοποιήσει το δίκτυο και να πραγματοποιήσει κάποια κλήση προς άλλον χρήστη, πρέπει πρώτα να περάσει από την κατάσταση της αυθεντικοποίησης του στο σύστημα με κάποιο συγκεκριμένο username-password, για να εγγραφεί στην υπηρεσία. Δημιουργήσαμε λοιπόν ένα SIP σενάριο όπου περιλαμβάνει SIP μηνύματα με τα οποία ο χρήστης εγγράφεται και

αυθεντικοποιείται στην υπηρεσία. Αυτό το βήμα είναι απαραίτητο για κάθε χρήστη που επιθυμεί να χρησιμοποιήσει την υπηρεσία μας “newims.org”. Αυτό το SIP σενάριο περιλαμβάνει αρχικά ένα REGISTER μήνυμα με το οποίο κάνει γνωστό στο δίκτυο ότι θέλει να εγγραφεί στην υπηρεσία. Έπειτα περιμένει να λάβει ένα 401-unauthorized μήνυμα από τον server και στη συνέχεια στέλνει πάλι ένα REGISTER μήνυμα με τα διαπιστευτήρια που χρειάζονται για να εγγραφεί. Τέλος περιμένει την έγκριση από τον server με ένα 200OK και πλέον είναι αυθεντικοποιημένος και μπορεί να χρησιμοποιήσει την υπηρεσία (παράρτημα 1) .



ΕΙΚΟΝΑ 20: ΕΓΓΡΑΦΗ ΜΕ SIP ΣΕΝΑΡΙΟ

Στην παρακάτω εικόνα έχουμε πραγματοποιήσει μια επικοπινωνία μεταξύ του bob και της alice.

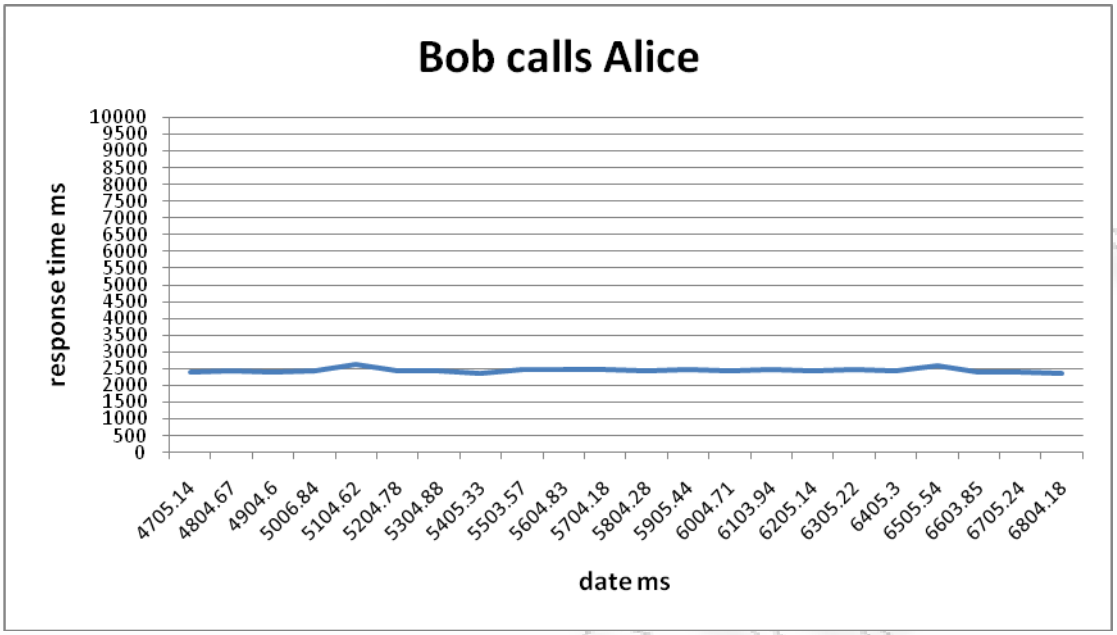


ΕΙΚΟΝΑ 21: ΕΠΙΚΟΙΝΩΝΙΑ BOB-ALICE ΜΕ SIP ΣΕΝΑΡΙΑ

Στον bob έχουμε φτιάξει sip μήνυμα το οποίο έπειτα από αυθεντικοποίηση του στο σύστημα στέλνει αίτημα επικοινωνίας με την alice οπότε και στέλνει μήνυμα invite προς της alice. Συνολικά το sip σενάριο του bob κάνει αίτημα για invite, έπειτα περιμένει το 200OK μήνυμα από την alice,όταν λάβει το 200OK στην συνέχεια στέλνει ένα ACK και υπάρχει παύση η οποία ισοδυναμεί με την επικοινωνία που έχουν οι επαφές μεταξύ τους. Έπειτα ο bob στέλνει BYE μήνυμα για να τερματίσει την κλήση, και περιμένει το 200OK από την alice για να το δεχτεί και να τερματιστεί η κλήση (παράρτημα 2).

Από την μεριά τώρα της alice. Το sip σενάριο που δημιουργήσαμε για την alice είναι να περιμένει κάποιο invite από το δίκτυο, όταν έρθει κάποιο invite να το δεκτή και να στείλει για απάντηση 200OK, στη συνέχεια να περιμένει ένα ACK από την επαφή για να υπάρξει η επικοινωνία. Στη συνέχεια να περιμένει ένα BYE μήνυμα για να της τερματίζουν την επικοινωνία και να το δέχεται απαντώντας με ένα 200OK και να τερματίζεται η επικοινωνία. Απαραίτητη προϋπόθεση βέβαια είναι να έχει αυθεντικοποιηθεί και η alice από το δίκτυο αλλιώς η επικοινωνία δεν θα πραγματοποιηθεί (παράρτημα 3).

Σημαντικό σε αυτό το σημείο είναι να παρατηρήσουμε ότι μέσω του “sipp” πραγματοποιούσαμε δέκα κλήσεις ανά δευτερόλεπτο (10calls/s) και μετέπειτα είκοσι κλήσεις ανά δευτερόλεπτο (20calls/s). Μέσω του “sipp” κάναμε μετρήσεις για να βγάλουμε χρήσιμα συμπεράσματα όσο αναφορά το χρόνο που ανταποκρίνεται το δίκτυο στις δέκα έως είκοσι κλήσεις ανά δευτερόλεπτο. Σύμφωνα με τα στατιστικά στοιχεία που παραθέτουμε παρακάτω παρατηρούμε μια ήπια ανταπόκριση του δικτύου της τάξης των 2500ms χρόνος ανταπόκρισης της κάθε κλίσης από το δίκτυο (παράρτημα 4).

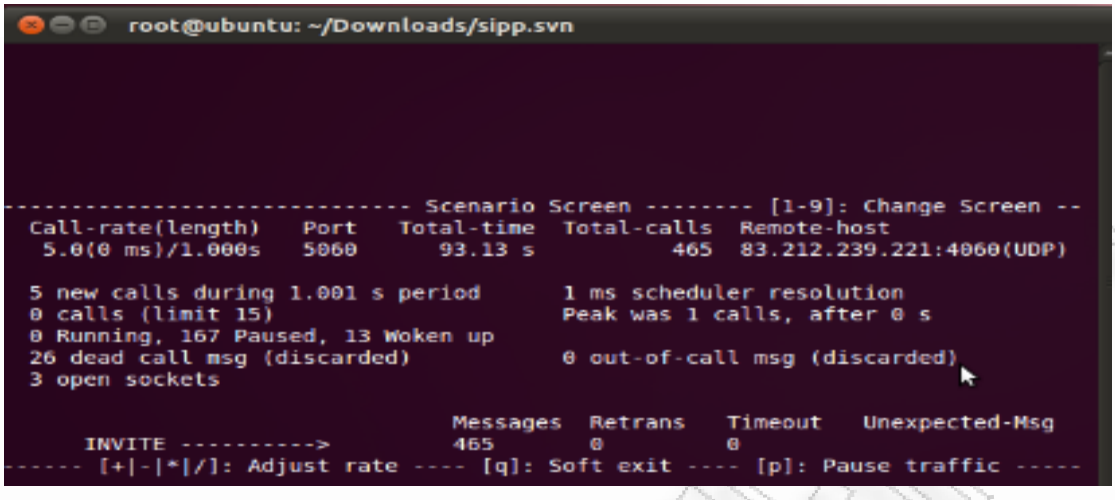


ΕΙΚΟΝΑ 22: ΧΡΟΝΟΣ ΑΝΤΑΠΟΚΡΗΣΗΣ ΤΟΥ ΔΙΚΤΥΟΥ

5.6. Δημιουργία malformed SIP μηνύματος

Όπως είδαμε με τα SIP σενάρια που πραγματοποιήσαμε η επικοινωνία μεταξύ bob και alice εξελισσόταν κανονικά με το δίκτυο να έχει ορθή ανταπόκριση στις κλήσεις που γινότανε ανά δευτερόλεπτο. Μετά απο την απλή επικοινωνία των δύο χρηστών, δημιουργήσαμε πάλι σε ένα SIP σενάριο ένα malformed μήνυμα. Το σκεπτικό μας είναι να δημιουργήσουμε ένα malformed SIP μήνυμα έτσι ώστε να κάνουμε επίθεση στο δίκτυο και να δούμε τι συμπεράσματα μπορούμε να βγάλουμε. Έτσι λοιπόν δημιουργήσαμε ένα SIP σενάριο που περιέχει μέσα ένα INVITE μήνυμα στο οποίο βρίσκουμε να επαναλαμβάνεται πολλές φορές η κεφαλίδα “To”, περίπου τριάντα φορές η κεφαλίδα “ To” (παράρτημα 5) . Σύμφωνα με τέτοιου είδους επιθέσεις, τα SIP μηνύματα τέτοιου είδους περνάμε στο δίκτυο αλλά το μπερδεύουν και προκαλούν καθυστερήσεις και σε ακραίο βαθμό κατάρρευση του δικτύου.

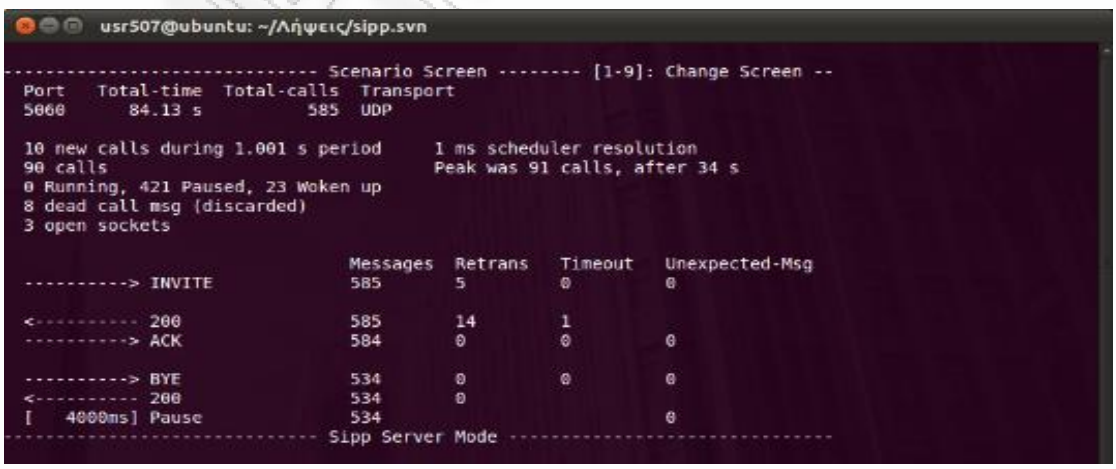
Επομένως το σενάριο της επίθεσης έχει ως εξής. Δημιουργούμε πάλι απλή επικοινωνία μεταξύ Bob και Alice όπως και πριν, και ενώ εξελίσσεται η μεταξύ τους επικοινωνία, ξεκινάμε από τον χρήστη “xristos” να στέλνουμε αυτό το malformed μήνυμα προς την Alice που την προκειμένη στιγμή εξυπηρετεί τον Bob. Ξεκινάμε με πέντε κλήσεις το δευτερόλεπτο και μετέπειτα αυξάνουμε το ρύθμο μέχρι και σαράντα κλήσεις το δευτερόλεπτο.



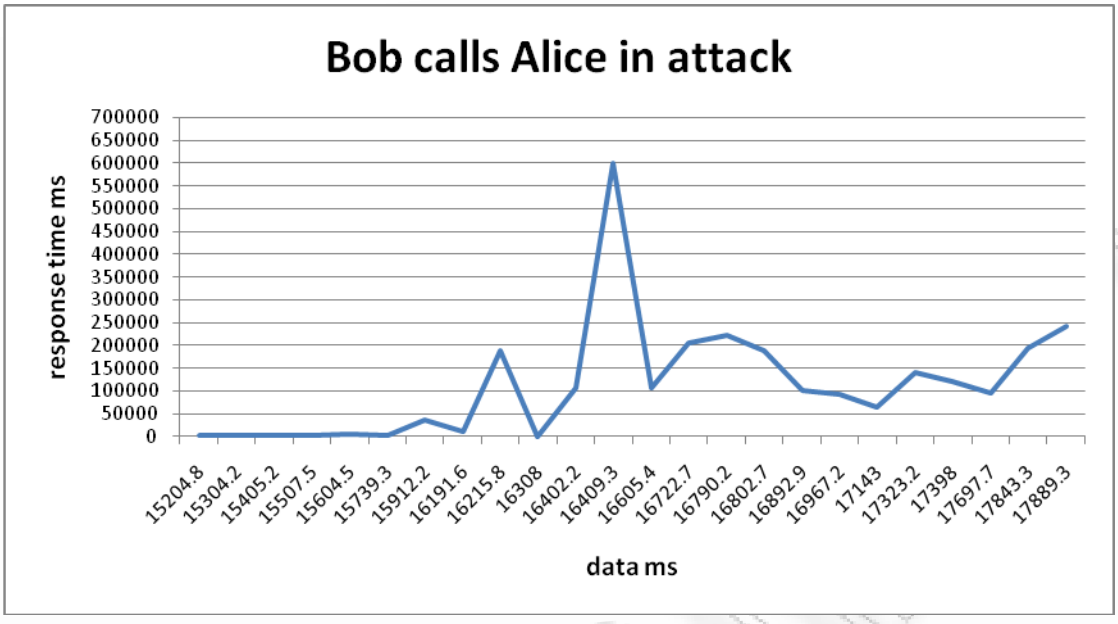
ΕΙΚΟΝΑ 23: ΕΠΙΘΕΣΗ ΜΕ MALFORMED SIP ΜΗΝΥΜΑ

5.7. Υλοποίηση επίθεσης με malformed μήνυμα

Στην αρχή της επίθεσης ακριβώς επειδή ο ρυθμός των κλήσεων είναι χαμηλός δεν παρατηρούμε τίποτα στις μετρήσεις μας. Μετέπειτα όμως παρατηρούμε στην Alice να αυξάνονται τα RETRANSMITIONS, λογικό αφού το malformed μήνυμα μας έχει μπερδέψει το δίκτυο και συν ότι περιμένει κάποιο ACK το οποίο ο επιτιθέμενος δεν θα το στείλει ποτέ. Επίσης πολλά συμπεράσματα βγαίνουν από την μεριά του Bob καθώς στην αρχή της επίθεσης δεν βλέπουμε κάτι περίεργο,έπειτα όταν αυξάνεται ο ρυθμός της επίθεσης παρατηρούμε στον Bob ότι αυξάνεται κατακόρυφα ο χρόνος που ανταποκρίνεται η Alice στις κλήσεις του (20calls/s) καθώς και ότι πολλές από αυτές αποτυγχάνουν να φτάσουν στην Alice αφού οι πόροι της Alice αρχίζουν να εξαντλούνται από την επίθεση (παράρτημα 6).



ΕΙΚΟΝΑ 24:ΕΠΙΚΟΙΝΩΝΙΑ ΕΠΑΦΩΝ ΜΕΤΑ ΤΗΝ ΕΠΙΘΕΣΗ

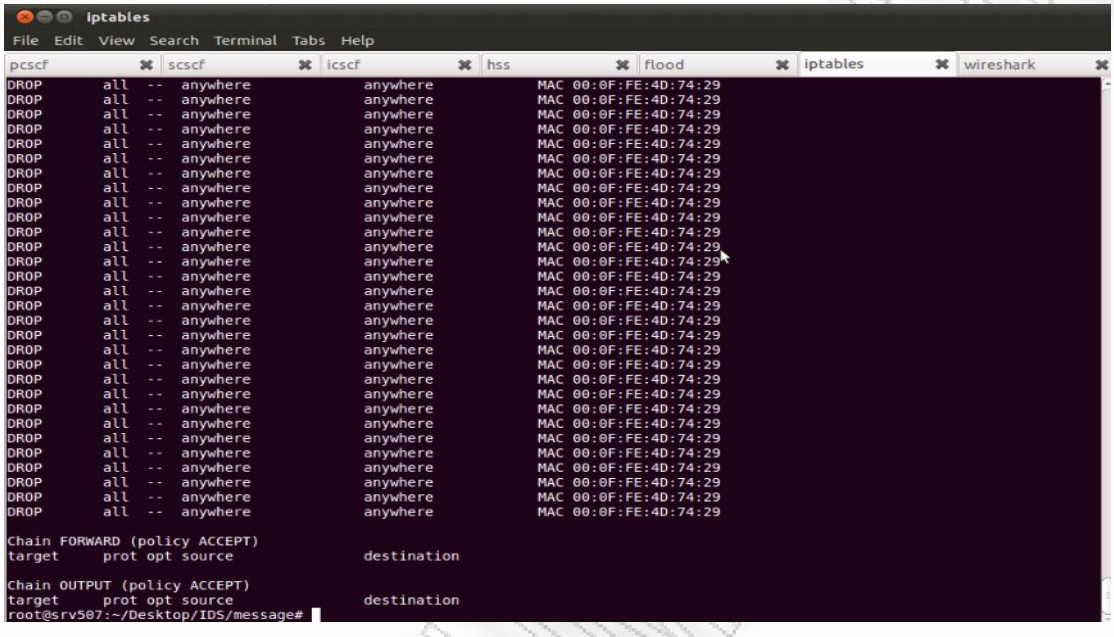


ΕΙΚΟΝΑ 25: ΧΡΟΝΟΣ ΑΝΤΑΠΟΚΡΗΣΗΣ ΤΟΥ ΔΙΚΤΥΟΥ ΚΑΤΑ ΤΗΝ ΔΙΑΡΚΕΙΑ ΤΗΝ ΕΠΙΘΕΣΗΣ

5.8. Μελέτη και υλοποίηση ενός IDS

Βλέποντας και παρατηρώντας τα αποτελέσματα και τις φθορές που προκαλέσαμε στο δίκτυο με το malformed μήνυμα που δημιουργήσαμε, μπήκαμε στη διαδικασία να φτιάξουμε έναν ανιχνευτή τέτοιων μηνυμάτων με αποτέλεσμα να ανταποκριθούμε στην αδυναμία του δικτύου σε τέτοιου είδους επιθέσεις. Το συγκεκριμένο malformed μήνυμα το δημιουργήσαμε αυξάνοντας τον αριθμό των “To” κεφαλίδων. Επομένως, δημιουργήσαμε ένα IDS/IPS (intrusion detection system/ intrusion prevention system) σύστημα, δηλαδή ένα σύστημα που εντοπίζει τέτοιου είδους επιθέσεις στο δίκτυο και προλαμβάνει το δίκτυο από τέτοιου είδους μελλοντικές επιθέσεις. Πιο συγκεκριμένα το σύστημα μας αναγνωρίζει ότι υπάρχει SIP μήνυμα με πολλαπλά headers από “To” και απορρίπτει αυτό το μήνυμα πριν το δρομολογήσει στο δίκτυο και δημιουργήσει πρόβλημα. Αφού αναγνωρίσει την επίθεση, εντοπίζει τον αποστολέα από την MAC address του και περνάει αυτόματα κανόνα στο firewall του δικτύου (iptables), κανόνα που λέει να μην δέχεται το δίκτυο πλέον SIP μηνύματα από τον συγκεκριμένο κακόβουλο αποστολέα. Το συγκεκριμένο IDS/IPS έχει γραφεί σε γλώσσα C++ και το έχουμε εγκαταστήσει στο δίκτυο μας “newims.org”. Ακολούθως βάλαμε το IDS/IPS σε λειτουργία, δημιουργήσαμε κανονικά μια επικοινωνία μεταξύ bob και alice και προσπαθήσαμε να κάνουμε με τον ίδιο τρόπο όπως και πριν την επίθεση μας. Το σύστημα μας ανίχνευσε την επίθεση λαμβάνοντας το πρώτο SIP malformed μήνυμα που δέχτηκε, ενημέρωσε με ένα FOUND! Ότι ανιχνεύτηκε επίθεση τέτοιας μορφής

με πολλαπλά headers και πέρασε κανόνα στα iptables να μπλοκάρει τον επίδοξο επιτιθέμενο, πράγμα που έγινε επιτυχώς απώτερο αποτέλεσμα η κανονική επικοινωνία μεταξύ bob και alice να συνεχιστεί κανονικά (παράρτημα 7), (παραρτημα 8).



ΕΙΚΟΝΑ 26: ΜΗΝΥΜΑ IDS/IPS ΠΟΥ ΜΠΛΟΚΑΡΕΙ ΤΗΝ ΕΠΙΘΕΣΗ

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ασφάλεια των υπηρεσιών διαδικτυακής τηλεφωνίας αποτελεί ακρογωνιαίο λίθο για την αποδοχή τους από το ευρύ κοινό, καθώς επηρεάζει σε πολύ μεγάλο βαθμό την αξιοπιστία και τη διαθεσιμότητα τους. Η αξιοποίηση δικτύων ανοιχτής αρχιτεκτονικής δημιουργεί νέες ευκαιρίες επιθέσεων και παραβίασης της ασφάλειας των παρεχόμενων υπηρεσιών με αποτέλεσμα, πολλές φορές, οι χρήστες να είναι διστακτικοί στη χρήση τους. Είναι λοιπόν απαραίτητο, να υλοποιηθούν μέτρα ασφαλείας που στοχεύουν στη βελτίωση της αξιοπιστίας και διαθεσιμότητας των υπηρεσιών διαδικτυακής τηλεφωνίας, προσελκύοντας ταυτόχρονα την εμπιστοσύνη των χρηστών προς τις υπηρεσίες αυτές.

Στην παρούσα διπλωματική εργασία στήσαμε ένα IMS δίκτυο με τους απαιτούμενους servers που χρειαζόταν. Στην συνέχεια εκτελέσαμε κανονικές επικοινωνίες μεταξύ των χρηστών του δικτύου με συγκεκριμένα μέτρα ασφάλειας όπως τις μεθόδους αυθεντικοποίησης που παρέχει και υποστηρίζει το ίδιο το δίκτυο. Έπειτα κάναμε επίθεση στο δίκτυο με malformed sip μήνυμα και διαπιστώσαμε ότι προκαλούσαμε delay στο δίκτυο όσο αναφορά τον χρόνο που ανταποκρινόταν στις υποχρεώσεις που είχε να κάνει (για παράδειγμα να συνεχίσει να παρέχει τις υπηρεσίες του στην επικοινωνίες που υπήρχαν την ώρα την επίθεσης). Τέλος, με βάση την πετυχημένη επίθεση που εκτελέσαμε, δημιουργήσαμε ένα IDS/IPS σύστημα με το οποίο ανιχνεύει τέτοιου είδους επιθέσεις με πολλαπλά headers μέσα σε ένα sip μήνυμα που σκοπό έχουν να μπερδέψουν το δίκτυο και να του καταναλώσουν πόρους.

Το IMS, αφενός προσφέρει πολλές δυνατότητες και πολλές διαφορετικές υπηρεσίες στους χρήστες, αφετέρου, προκαλεί προβλήματα ως προς την παρεχόμενη ασφάλεια και παρουσιάζει αδυναμίες στην υποδομή του. Αυτές τις αδυναμίες θα μπορούσαν να γίνουν αντικείμενο μελέτης συστημάτων που θα παρέχουν την απαιτούμενη ασφάλεια για τις απαιτήσεις συστημάτων όπως αυτό του IMS. Τέτοια συστήματα ασφάλειας θα μπορούσαν να είναι συστήματα ανίχνευσης παρείσφρησης (IDS) και σύστημα πρόληψης παρείσφρησης (IPS).

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]:Sven Ehlert, Dimitris Geneiatakis, Thomas Magedanz, "Survey of network security systems to counter SIP-based denial-of-service attacks", ScienceDirect, ELSERVIER, September 2009.

[2]:Σιδηροπούλου Χριστίνα «Δομή υπηρεσιών στα δίκτυα επομένης γενιάς», Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών, 2009.

[3]:Andreea Ancuta Onofrei, Yacine Rebahi, Thomas Magedanz, "Preventing Distributed Denial-of-Service Attacks on the IMS Emergency Services Support through Adaptive Firewall Pinholing", The International Journal of Next Generation Network, Vol.2, No.1, March 2010.

[4]:Γενειατάκης Δημήτρη, «Πλαίσιο Ανίχνευσης και Αντιμετώπισης Περιστατικών Ασφαλείας σε Συστήματα Διαδικτυακής Τηλεφωνίας», Πανεπιστήμιο Αιγαίου, 2008.

[5]:Dimitris Geneiatakis, Geogrios Kambourakis, Costas Lambrinoudakis, Tasos Dagiuklas, Stefanos Gritzalis, "A framework for protecting a SIP-based infrastructure against malformed message attacks", ScienceDirect, ELSERVIER, November 2006.

[6]:Dorgham Sisalem, John Floroiu, Jiri Kuthan, Ulrich Abend, Henning Schulzrinne "SIP SECURITY", John Wiley & Sons Ltd., 2009.

[7]:Christian Wieser, Marko Laakso, "SIP Robustness Testing for Large-Scale Use", Department of Electrical and Information Engineering University of Oulu, 2009.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ 1: SIP ΣΕΝΑΡΙΟΓΙΑ ΕΓΓΡΑΦΗ ΤΟΥ ΧΡΗΣΤΗ ΣΤΟ ΔΙΚΤΥΟ

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<scenario name="registration">

<send retrans="500">
<![CDATA[
REGISTER sip:newims.org SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
Max-Forwards: 20
From: "alice" <sip:alice@newims.org>;tag=[call_number]
To: "alice" <sip:alice@newims.org>
Call-ID: reg///[call_id]
CSeq: 1 REGISTER
Contact: <sip:alice@[local_ip]:[local_port]>
Expires: 3600
Content-Length: 0
User-Agent: Sipp v1.1-TLS, version 20061124
Authorization: Digest username="alice@newims.org",
realm="newims.org"
Supported: path
]]>
</send>

<recv response="401" auth="true" rtd="true">
<action> <ereg regexp=".*" search_in="hdr" header="Service-Route" assign_to="1" /> </action>
</recv>

<send retrans="500">
<![CDATA[
REGISTER sip:newims.org SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
Route: [$1]
Max-Forwards: 20
From: "alice" <sip:alice@newims.org>;tag=[call_number]
To: "alice" <sip:alice@newims.org>
Call-ID: reg///[call_id]
CSeq: 2 REGISTER
Contact: <sip:alice@[local_ip]:[local_port]>
Expires: 3600
Content-Length: 0
User-Agent: Sipp v1.1-TLS, version 20061124
[authentication username=alice@newims.org password=alice]
Supported: path
]]>
</send>

<recv response="200">
</recv>

<ResponseTimeRepartition value="10, 20"/>
<CallLengthRepartition value="10"/>
</scenario>
```

ΠΑΡΑΡΤΗΜΑ 2: SIP ΣΕΝΑΡΙΟΓΙΑ ΕΠΙΚΟΙΝΩΝΙΑ ΧΡΗΣΤΗ ΣΤΟ ΔΙΚΤΥΟ

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<scenario name="bob_calls_alice">

<send retrans="500" start_rtd="1" >
<![CDATA[
INVITE sip:alice@newims.org SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
Max-Forwards: 20
Route: <sip:orig@scscf.newims.org:6060;lr>
P-Preferred-Identity: <sip:bob@newims.org>
Privacy: none
P-Access-Network-Info: 3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
From: <sip:bob@newims.org>;tag=[call_number]
To: <alice@newims.org>
Call-ID: [call_id]
CSeq: 10 INVITE
Supported: 100rel
Contact: <sip:bob@[local_ip]:[local_port]>
User-Agent: Sipp v1.1-TLS, version 20061124
Allow: ACK, BYE, CANCEL, INVITE, REFER, OPTIONS, INFO, REGISTER, NOTIFY,
UPDATE, SUBSCRIBE, PRACK
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP4 [local_ip]
s=-
c=IN IP4 [local_ip]
t=0 0
m=audio 30000 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=sendrecv
]]>
</send>

<recv response="200" rrs="true" rtd="1">
</recv>

<send crlf="true">
<![CDATA[
ACK [next_url] SIP/2.0
[routes]
[last_Via:]
Max-Forwards: 20
From: <sip:bob@newims.org>;tag=[call_number]
[last_To:]
Call-ID: [call_id]
CSeq: 10 ACK
Content-Length: 0
]]>
</send>

<pause milliseconds="5000" crlf="true" />

<send retrans="500">
<![CDATA[
BYE sip:[next_url] SIP/2.0
```

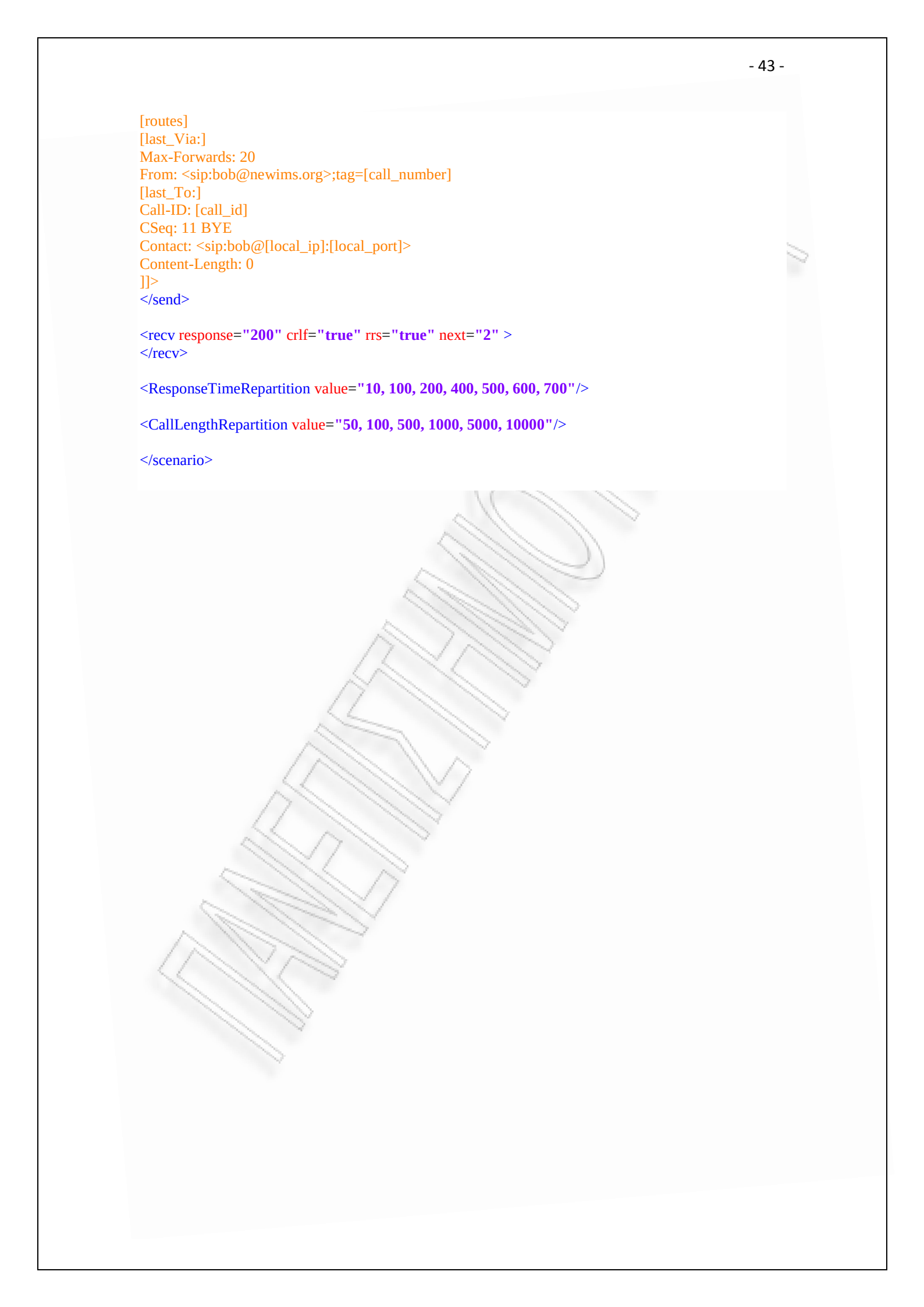
```
[routes]
[last_Via:]
Max-Forwards: 20
From: <sip:bob@newims.org>;tag=[call_number]
[last_To:]
Call-ID: [call_id]
CSeq: 11 BYE
Contact: <sip:bob@[local_ip]:[local_port]>
Content-Length: 0
]]>
</send>

<recv response="200" crlf="true" rrs="true" next="2" >
</recv>

<ResponseTimeRepartition value="10, 100, 200, 400, 500, 600, 700"/>

<CallLengthRepartition value="50, 100, 500, 1000, 5000, 10000"/>

</scenario>
```



ΠΑΡΑΡΤΗΜΑ 3: SIP ΣΕΝΑΡΙΟΓΙΑ ΑΠΟΔΟΧΗ ΕΠΙΚΟΙΝΩΝΙΑΣ ΧΡΗΣΤΗ ΣΤΟ ΔΙΚΤΥΟ

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<scenario name="alice_waiting_calls">

 <recv request="INVITE" crlf="true">
 </recv>

 <send retrans="20">
<![CDATA[
SIP/2.0 200 OK
[last_Record-Route:]
[last_Via:]
[last_From:]
[last_To:];tag=[call_number]
[last_Call-ID:]
[last_CSeq:]
Contact: sip:alice@[local_ip]:[local_port];transport=[transport]
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[media_ip_type] [media_ip]
t=0
m=audio [media_port] RTP/AVP 0
a=rtpmap:0 PCMU/8000

]]>
 </send>

 <recv request="ACK"  crlf="true">
 </recv>

 <recv request="BYE">
 </recv>

<send>
<![CDATA[
SIP/2.0 200 OK
[last_Via:]
[last_From:]
[last_To:]
[last_Call-ID:]
[last_CSeq:]
Contact: <sip:alice@[local_ip]:[local_port];transport=[transport]>
Content-Length: 0
]]>
</send>

<pause milliseconds="4000"/>

<ResponseTimeRepartition value="10, 100, 200, 400, 500, 600, 700"/>

<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>
```

ΠΑΡΑΡΤΗΜΑ 4: ΜΕΤΡΗΣΕΙΣ ΑΠΟ ΤΙΣ ΑΝΤΑΠΟΚΡΙΣΕΙΣ ΤΟΥ ΔΙΚΤΥΟΥ ΣΤΗΝ ΚΑΝΟΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑ

Date_ms	response_time_ms	Date_ms	response_time_ms	Date_ms	response_time_ms
1047.21	3118	5204.78	2423	10304.60	2392
2042.58	2408	5304.88	2434	10403.60	2398
3038.07	2477	5405.33	2376	10504.80	2506
4055.69	2594	5503.57	2458	10604.90	2444
5040.60	2483	5604.83	2488	10704.10	2481
6034.94	2445	5704.18	2464	10804.10	2442
7050.71	2354	5804.28	2441	10904.40	2420
8045.20	2353	5905.44	2468	11004.50	2429
9040.69	2481	6004.71	2422	11104.10	2665
1004.84	2575	6103.94	2466	11204.80	2447
1104.95	2435	6205.14	2449	11303.80	2459
1204.98	2422	6305.22	2467	11404.90	2395
1304.76	2401	6405.30	2432	11505.30	2437
1404.17	2345	6505.54	2577	11605.40	2423
1504.83	2522	6603.85	2403	11704.10	2658
1604.33	2467	6705.24	2407	11805.30	2445
1704.26	2549	6804.18	2356	11905.40	2433
1803.77	2441	6904.26	2440	12005.40	2418
1903.44	2361	7005.40	2451	12105.40	2536
2003.96	2373	7104.16	2436	12204.60	2532
2103.99	2528	7204.48	2431	12303.70	2495
2203.59	2494	7304.59	2504	12403.80	2443
2304.19	2488	7404.57	2356	12503.90	2496
2405.12	2405	7504.69	2523	12605.10	2454
2504.61	2440	7604.85	2512	12704.00	2425
2605.20	2488	7704.14	2435	12805.10	2418
2704.67	2422	7804.71	2451	12905.30	2510
2805.20	2451	7904.55	2368	13003.50	2480
2904.62	2400	8004.66	2451	13104.80	2516
3004.14	2417	8104.42	2681	13203.90	2459
3104.70	2437	8204.70	2512	13305.40	2415
3205.20	2370	8304.76	2509	13405.40	2416
3303.60	2436	8404.74	2402	13504.90	2572
3405.30	2553	8505.40	2551	13605.00	2420
3504.85	2502	8604.32	2383	13705.20	2468
3604.76	2876	8705.46	2476	13804.20	2425
3704.54	2451	8804.51	2476	13905.20	2500
3803.99	2383	8903.62	2539	14004.50	2409
3903.49	2417	9004.79	2446	14104.60	2575
4004.07	2473	9104.99	2449	14203.50	2448
4103.66	2459	9205.00	2416	14304.70	2433
4205.47	2325	9304.05	2466	14403.90	2515
4305.08	2511	9404.10	2419	14505.10	2438
4403.44	2358	9505.35	2508	14605.10	2406
4504.05	2479	9604.29	2411	14704.00	2799
4603.57	2496	9704.15	2582	14805.20	2536
4705.14	2413	9804.25	2404	14905.00	2908
4804.67	2453	9905.48	2525	15004.50	2630
4904.60	2387	10004.5	2424	15104.90	2858
5006.84	2421	10103.9	2445	15204.80	2599
5104.62	2628	10204.1	2535	15304.20	2569

ΠΑΡΑΡΤΗΜΑ 5: MALFORMED SIP ΣΕΝΑΡΙΟ

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<scenario name="xristos_calls_alice_malformed">

<send retrans="500" >
<![CDATA[
INVITE sip:alice@newims.org SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
Max-Forwards: 20
Route: <sip:orig@scscf.newims.org:6060;lr>
P-Preferred-Identity: <sip:bob@newims.org>
Privacy: none
P-Access-Network-Info: 3GPP-UTRAN-TDD;utran-cell-id-3gpp=C359A3913B20E
From: <sip:bob@newims.org>;tag=[call_number]
To: <alice@newims.org>
To: <alice@newims.org>
To: <alice@newims.org>
To: <alice@newims.org>
To: <alice@newims.org>
              :
              :
To: <alice@newims.org>


Call-ID: [call_id]
CSeq: 10 INVITE
Supported: 100rel
Contact: <sip:bob@[local_ip]:[local_port]>
User-Agent: Sipp v1.1-TLS, version 20061124
Allow: ACK, BYE, CANCEL, INVITE, REFER, OPTIONS, INFO, REGISTER, NOTIFY,
UPDATE, SUBSCRIBE, PRACK
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP4 [local_ip]
s=-
c=IN IP4 [local_ip]
t=0 0
m=audio 30000 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=sendrecv
]]>
</send>

</scenario>
```

150 φορές το header “To:<alice@newims.org”

ΠΑΡΑΡΤΗΜΑ 6: ΜΕΤΡΗΣΕΙΣ ΑΠΟ ΤΙΣ ΑΝΤΑΠΟΚΡΙΣΕΙΣ ΤΟΥ ΔΙΚΤΥΟΥ ΣΤΗΝ ΚΑΝΟΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑ ΕΝΩ ΓΙΝΕΤΑΙ Η ΕΠΙΘΕΣΗ

Date_ms	response_time_ms	Date_ms	response_time_ms	Date_ms	response_time_ms
15405.2	3284	37704.9	302465	64610.1	307695
15507.5	5151	37835.1	333361	64789.7	287977
15604.5	2429	37888.8	28700	64795.5	393298
15739.3	37266	38008.2	305742	64795.7	693172
15912.2	11026	38096.2	294398	64844.0	241748
16191.6	189699	38352.2	450393	64933.4	231885
16215.8	113960	38399.7	296732	65092.1	289996
16308.0	105682	38504.9	303413	65201.8	299619
16402.2	599582	38636.8	335614	65351.6	349309
16409.3	106834	38703.6	300626	65510.9	308905
16605.4	204336	38801.1	299439	65643.2	342012
16722.7	221515	38911.7	309075	65694.8	1793150
16790.2	187353	38988.7	287589	65709.7	307332
16802.7	101205	39099.3	298129	65811.8	309226
16892.9	91559	39193.1	291399	66020.4	317930
16967.2	65427	39335.2	333182	66109.4	307901
17143.0	140482	39410.2	308576	66203.2	301728
17323.2	121264	39503.0	301234	66405.3	303951
17398.0	95487	39603.2	300326	66467.7	465684
17697.7	195616	39757.0	1755770	66498.3	295451
17843.3	241852	39809.0	306375	66565.7	263083
17889.3	187084	39923.7	321911	66972.5	370767
17892.1	90980	40060.3	359167	67048.4	346108
17905.1	503025	40126.2	323785	67164.1	361738
17910.5	8348	40166.8	764813	67230.4	327541
18053.4	51113	40210.3	307722	67317.9	316447
18211.9	1109530	40308.9	306950	67432.0	329353
18234.2	32484	40514.2	311360	67522.7	320025
18376.0	73420	40605.2	303793	67609.9	308616
18494.0	92318	40711.0	308648	67625.0	1572208
18530.7	29092	40840.7	337679	67701.4	300328
18673.5	570979	40931.3	329892	67784.8	282899
18673.6	72427	41013.4	310838	68228.0	326035
18824.0	121374	41106.6	303910	68340.2	337092
18828.9	27460	41187.3	285322	68404.1	301035
18962.2	60593	41319.7	317950	68532.6	331055
19117.4	114591	41408.6	306283	68608.9	306098
19118.2	15860	41504.2	301670	68700.7	299074
19255.4	52390	41598.2	29680	68804.5	303246
19365.9	63443	41673.0	271336	68898.9	296063
19404.3	2865	41843.1	341091	68960.9	259081
19504.4	3349	41917.1	1814360	69336.2	334384
19604.5	2585	42019.4	317831	69401.8	299828
19705.9	4225	42145.8	343067	69479.9	277933
19804.1	2754	42304.2	302421	69947.9	346679
19918.2	16325	42417.7	316704	70002.6	299729
20010.9	8302	42529.4	326553	70097.6	296161
20106.4	4972	42780.0	377779	70193.8	292047
20204.2	2569	42804.4	303211	70271.5	269026
20364.9	62241	42933.7	332467	70358.4	256356
20405.6	3306	43019.6	318542	70796.8	294076
20504.8	3263	43106.2	303192	70898.3	296085
20610.0	8411	43211.4	309066	70908.6	706753
20706.1	3234	43276.9	274853	71058.2	257149
20812.5	10649	43408.8	307351	71205.6	303934
20903.9	2815	43509.3	307872	71338.0	336637
21005.6	2636	43705.2	304070	71522.5	819843
21108.2	5619	43816.5	314315	71526.1	324305
21205.0	2851	43895.5	293047	71796.4	393515
21305.9	2986	43966.9	264240	71913.9	412827
21405.6	3110	44065.8	1762780	72159.3	456698
21505.8	3398	44206.3	304274	72247.3	444427
21607.2	4335	44302.1	300273	72338.4	436219
21705.5	2491	44440.2	337064	72444.8	442565

21803.9	2776	44608.3	306158	72561.6	458592
21927.6	25399	44702.4	300321	72667.3	464348
22005.1	2594	44875.9	273751	72731.9	430854
22110.4	9157	44999.7	796951	72832.6	330826
22205.7	4295	45000.6	298853	72887.2	1783004
22325.0	23912	45270.6	267830	72889.0	486521
22406.1	3032	45310.1	807782	72918.6	316329
22733.8	31415	45312.1	510856	73526.6	325224
22849.2	47034	48217.2	31500	73621.1	319426
22986.8	85025	48418.9	317816	73780.4	378652
23005.2	2674	48490.4	288918	73825.8	324037
23105.9	2961	48594.3	292401	73969.4	367188
23206.0	3004	48683.7	280809	74128.8	325899
23305.6	3118	48802.5	300316	74499.8	297601
23406.2	3340	48916.4	314481	74723.3	320525
23503.5	2450	48998.2	296217	75122.3	319902
23603.7	2593	49112.2	310446	75255.2	352709
23704.9	3503	49206.2	304156	75323.4	321193
23805.7	3014	49309.4	307882	75424.9	323202
23906.3	3549	49398.9	294105	75507.2	305136
24038.8	37186	49593.6	290873	75600.4	298368
24112.6	10208	49687.6	285324	75656.0	254501
24204.5	2873	49787.2	286155	75992.7	290872
24305.6	3494	50000.6	798883	76117.8	315126
24404.8	2459	50006.3	304559	76323.3	320921
24516.7	14985	50101.9	299576	76441.8	338777
24655.0	53949	50226.8	324037	76650.2	347964
24759.0	56711	50356.5	254425	76712.6	309712
24880.8	78967	50437.7	236538	77134.7	333635
24904.6	2846	50580.7	278033	77213.3	311056
25005.9	3527	50609.1	207981	77415.7	314561
25119.6	18193	50813.1	210489	77515.9	312841
25231.4	29689	50895.6	193375	77610.6	309569
25348.4	46060	51134.7	232605	77809.5	308425
25439.6	3860	51231.1	229883	77837.7	835536
25552.6	50544	51350.6	247995	77904.0	301106
25625.0	23174	51431.5	229804	77995.6	292738
25709.7	7798	51478.1	3675340	78002.4	1801060
25836.1	34822	51524.8	221958	78096.2	294150
25953.8	51032	51637.4	234647	78220.1	318457
26041.4	38611	51691.9	189173	78327.2	326114
26118.0	15453	51790.9	188190	78530.8	427355
26219.1	16040	51997.1	194413	78693.7	392537
26342.2	39718	52182.0	1679380	78725.9	324355
26460.8	58031	52191.3	188708	78852.5	651457
26584.0	81511	52221.0	118707	78870.8	369481
26724.3	121490	52354.8	152558	78927.0	318756
26763.9	61270	52380.2	678575	79114.2	311854
26895.6	93179	52544.4	242697	79258.4	355579
26904.1	2881	52589.9	188296	79323.3	321786
27012.0	10817	52718.8	215781	79423.7	321469
27135.4	33623	52778.2	176697	79549.8	346985
27266.2	63220	52836.3	135241	79618.9	317597
27447.1	145407	53043.1	240181	79727.0	325428
27469.4	66927	53098.7	196424	79834.4	331858
27577.7	75134	53192.5	189785	79929.9	327262
27605.0	3375	53283.0	179611	80002.4	299660
27705.9	3541	53335.3	133038	80105.1	302933
27901.0	98041	53564.4	262235	80230.4	329190
27909.7	8582	53710.5	308722	80329.1	326608
28103.6	101018	53805.7	304107	80517.8	1816690
28110.1	8657	53901.6	298839	80624.3	322714
28238.3	35293	53923.0	220263	80650.8	449173
28359.9	57793	54021.2	219461	80725.4	323473
28492.9	91173	54049.7	148555	80826.7	324002
28504.8	2888	54144.6	143317	80896.5	794049
28616.5	13550	54373.1	270287	80947.2	344992
28733.7	31623	54515.4	31250	81024.8	323739
28850.0	47937	54588.7	287102	81130.4	327953

28948.5	47319	54666.6	263617	81212.2	309180
29069.6	67451	54756.9	255463	81360.7	357912
29168.0	66331	54896.0	293636	81438.3	335871
29207.0	5729	55005.6	302860	81652.6	351042
29305.3	3243	55103.4	300417	81740.0	338701
29539.6	137436	55189.7	288234	81837.1	335062
29725.2	224051	55213.3	211525	81921.4	320334
29891.4	290399	55344.8	241969	82110.0	308537
29984.2	281640	55457.6	256188	82213.0	310939
30068.2	266172	55599.8	297931	82312.2	310777
30155.6	253421	55688.7	286736	82432.0	330039
30289.0	286497	55729.4	226870	82511.0	309212
30378.6	275766	55808.7	206662	82601.9	299189
30503.1	301645	55893.0	190319	82731.6	329159
30598.2	297111	56019.1	217327	82909.2	307120
30682.1	279039	56134.8	233024	83031.5	329027
30761.3	258611	56165.0	163844	83156.2	353798
30874.9	273315	56317.7	215196	83260.7	357096
30992.9	290378	56537.9	336614	83312.8	311006
31091.3	289107	56615.0	313615	83545.7	343241
31214.2	311556	56783.5	381472	83622.1	320660
31292.3	289962	56905.3	403095	83725.6	323852
31377.4	276274	56923.8	321752	83818.3	317232
31510.1	307582	56991.9	290746	83891.7	789741
31797.6	295284	57074.3	271356	83935.1	332103
31922.2	319809	57135.7	233361	84012.7	311437
32000.4	298711	57308.5	306743	84244.5	442581
32080.5	278427	57339.9	237725	84275.1	273499
32169.7	268477	57405.3	202484	84395.1	293113
32258.2	256079	57482.2	179467	84559.6	357286
32365.4	263586	57768.4	266210	84677.0	774816
32488.6	285568	57837.2	234189	84712.3	309331
32561.6	258731	57914.9	212413	84920.6	317897
32683.5	281818	58104.7	302355	85000.9	3798250
32762.0	259370	58212.3	310560	85003.9	301450
32890.1	287558	58328.2	326361	85085.4	283603
32978.2	276493	58418.1	315486	85166.9	264421
33072.3	270071	58533.2	331581	85292.7	290604
33199.9	296738	58569.2	7767480	85388.0	285004
33298.0	295738	58611.2	309239	85466.6	264519
33415.8	313304	58724.1	322710	85566.9	265418
33502.4	300158	58808.3	306578	85771.2	268609
33601.1	299772	59075.4	273080	85856.4	255082
33691.6	290116	59167.0	176507	85958.1	256043
33786.5	284016	59273.5	271829	86014.7	212573
33892.8	291549	59653.2	351243	86119.8	716739
33984.4	281450	59732.0	330271	86192.3	1691220
34083.8	281092	59815.7	312876	86236.8	235096
34208.0	305508	59894.1	292167	86304.6	3802570
34298.3	297260	59967.9	266362	86308.1	406923
34380.9	278272	60032.2	130793	86351.8	250221
34479.6	278004	60351.6	250581	86434.1	232344
34613.1	312066	60510.7	307765	86555.2	254125
34695.1	292977	60608.8	307131	86616.0	213501
34781.6	280528	60727.0	325161	86750.6	247979
34857.1	255646	60786.1	283631	86879.0	277824
34975.6	272906	60820.8	218979	86913.1	210023
35079.8	277023	60983.6	281023	86984.1	181390
35169.7	268013	61057.9	256175	87084.0	181388
35260.4	258291	61279.3	376936	87241.6	238903
35355.7	254182	61300.3	298126	87394.9	192044
35450.3	248883	61332.4	231022	87544.6	242292
35572.1	269989	61677.5	275029	87633.2	230733
35674.8	271754	61768.2	265853	87688.1	185393
35797.6	295594	62019.2	317016	87733.0	131713
35887.8	286140	62139.8	336815	87751.9	649694
36020.4	319164	62221.8	319324	87836.6	135555
36110.0	307975	62314.6	312758	87952.4	150045
36208.8	305752	62482.6	180571	88134.5	231794

36291.8	289772	62572.6	171524	88252.8	251348
36407.3	304318	62724.8	222774	88308.7	207014
36497.7	295591	63008.7	307558	88351.6	148568
36593.9	292520	63123.4	321663	88527.1	224635
36692.1	289767	63243.2	341828	88625.2	222473
36780.9	278839	63320.8	317759	88790.0	287830
36927.5	326095	63434.9	333304	88988.6	287034
37009.6	308554	63514.8	312056	89141.3	239615
37125.1	323754	63688.6	287265	89146.6	343698
37211.3	309294	63706.9	204283	89194.4	192977
37312.7	310713	64152.2	851204	89308.1	206081
37435.0	332918	64193.8	392447	89324.0	722733
37512.5	310308	64275.5	272427	89393.3	191018
37607.5	305206	64505.5	30402		

ΠΑΡΑΡΤΗΜΑ 7: ΚΩΔΙΚΑΣ ΤΟΥ IDS/IPS ΣΥΣΤΗΜΑΤΟΣ

```
/* sniffex.c
 * Sniffer example of TCP/IP packet capture using libpcap.
*/
#define APP_NAME        "sniffex"
#define APP_DESC        "Sniffer example using libpcap"
#define APP_COPYRIGHT   "Copyright (c) 2005 The Tcpdump Group"
#define APP_DISCLAIMER  "THERE IS ABSOLUTELY NO WARRANTY FOR THIS
PROGRAM."

#include <pcap.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#include <netdb.h>

#include <netinet/if_ether.h>
//#include <net/ethernet.h>
#include <netinet/ether.h>

int coot = 0; //ATJ: metavliti counter gia ton pinaka me ta stoixeia pou theloume
int eidos; //ATJ: metavliti gia na ksexorizoume ta INVITE apo ta 200 OK paketa
int table_var = 0; //ATJ: metavliti gia tin euresh idou iptables rule

/* default snap length (maximum bytes per packet to capture) */
#define SNAP_LEN 1518

/* ethernet headers are always exactly 14 bytes [1] */
#define SIZE_ETHERNET 14

/* Ethernet addresses are 6 bytes */
//#define ETHER_ADDR_LEN    6

/* Ethernet header */
struct sniff_ethernet {
        u_char  ether_dhost[ETHER_ADDR_LEN];    /* destination host address */
        u_char  ether_shost[ETHER_ADDR_LEN];    /* source host address */
        u_short ether_type;                     /* IP? ARP? RARP? etc */
};

/* IP header */
struct sniff_ip {
        u_char  ip_vhl;                 /* version << 4 | header length >> 2 */
        u_char  ip_tos;                 /* type of service */
        u_short ip_len;                 /* total length */
        u_short ip_id;                  /* identification */
        u_short ip_off;                 /* fragment offset field */
        #define IP_RF 0x8000            /* reserved fragment flag */
        #define IP_DF 0x4000            /* dont fragment flag */
        #define IP_MF 0x2000            /* more fragments flag */
        #define IP_OFFMASK 0x1fff       /* mask for fragmenting bits */
        u_char  ip_ttl;                 /* time to live */
        u_char  ip_p;                   /* protocol */
```

```
        u_short ip_sum;                 /* checksum */
        struct  in_addr ip_src,ip_dst;  /* source and dest address */
};
#define IP_HL(ip)               (((ip)->ip_vhl) & 0x0f)
#define IP_V(ip)                (((ip)->ip_vhl) >> 4)

/* UDP header */
struct sniff_udp {
        u_short uh_sport;               /* source port */
        u_short uh_dport;               /* destination port */
        u_short uh_ulen;                /* udp length */
        u_short uh_sum;                 /* udp checksum */

};

#define SIZE_UDP        8               /* length of UDP header */


void
got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char *packet);

void
print_payload(const u_char *payload, int len);

void
print_hex_ascii_line(const u_char *payload, int len, int offset);

void
print_app_banner(void);

void
print_app_usage(void);

/*
 * app name/banner
 */
void
print_app_banner(void)
{

    printf("%s - %s\n", APP_NAME, APP_DESC);
    printf("%s\n", APP_COPYRIGHT);
    printf("%s\n", APP_DISCLAIMER);
    printf("\n");

return;
}

/*
 * print help text
 */
void
print_app_usage(void)
{

    printf("Usage: %s [interface]\n", APP_NAME);
    printf("\n");
    printf("Options:\n");
    printf("    interface    Listen on <interface> for packets.\n");
    printf("\n");
```

```
return;
}

/*
 * print data in rows of 16 bytes: offset   hex   ascii
 *
 * 00000   47 45 54 20 2f 20 48 54  54 50 2f 31 2e 31 0d 0a   GET / HTTP/1.1..
 */
void
print_hex_ascii_line(const u_char *payload, int len, int offset)
{

    int i;
    int gap;
    const u_char *ch;

    /* offset */
    printf("%05d   ", offset);

    /* hex */
    ch = payload;
    for(i = 0; i < len; i++) {
        printf("%02x ", *ch);
        ch++;
        /* print extra space after 8th byte for visual aid */
        if (i == 7)
            printf(" ");
    }
    /* print space to handle line less than 8 bytes */
    if (len < 8)
        printf(" ");

    /* fill hex gap with spaces if not full line */
    if (len < 16) {
        gap = 16 - len;
        for (i = 0; i < gap; i++) {
            printf("   ");
        }
    }
    printf("   ");

    /* ascii (if printable) */
    ch = payload;
    for(i = 0; i < len; i++) {
        if (isprint(*ch))
            printf("%c", *ch);
        else
            printf(".");
        ch++;
    }

    printf("\n");

return;
}

/*
 * print packet payload data (avoid printing binary data)
 */
```

```
void
print_payload(const u_char *payload, int len)
{

    int len_rem = len;
    int line_width = 16;           /* number of bytes per line */
    int line_len;
    int offset = 0;                /* zero-based offset counter */
    const u_char *ch = payload;

    if (len <= 0)
        return;

    /* data fits on one line */
    if (len <= line_width) {
        print_hex_ascii_line(ch, len, offset);
        return;
    }

    /* data spans multiple lines */
    for ( ;; ) {
        /* compute current line length */
        line_len = line_width % len_rem;
        /* print line */
        print_hex_ascii_line(ch, line_len, offset);
        /* compute total remaining */
        len_rem = len_rem - line_len;
        /* shift pointer to remaining bytes to print */
        ch = ch + line_len;
        /* add offset */
        offset = offset + line_width;
        /* check if we have line width chars or less */
        if (len_rem <= line_width) {
            /* print last line and get out */
            print_hex_ascii_line(ch, len_rem, offset);
            break;
        }
    }

return;
}

/*
 * dissect/print packet
 */
void
got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char *packet)
{

    static int count = 1;                   /* packet counter */

    /* declare pointers to packet headers */
    const struct sniff_ethernet *ethernet;  /* The ethernet header [1] */
    const struct sniff_ip *ip;              /* The IP header */
    const struct sniff_udp *udp;            /* The TCP header */
    const char *payload;                    /* Packet payload */

    int size_ip;
    int size_udp;
    int size_payload;
```

```
    printf("\n######################\n");
    printf("\nPacket number %d:\n", count);
    count++;

    /* define ethernet header */
    ethernet = (struct sniff_ethernet*)(packet);

    /* define/compute ip header offset */
    ip = (struct sniff_ip*)(packet + SIZE_ETHERNET);
    size_ip = IP_HL(ip)*4;
    if (size_ip < 20) {
       printf("   * Invalid IP header length: %u bytes\n", size_ip);
       return;
    }

    /* print source and destination IP addresses */
    printf("       From: %s\n", inet_ntoa(ip->ip_src));
    printf("         To: %s\n", inet_ntoa(ip->ip_dst));

    const char *dassip = inet_ntoa(ip->ip_dst); //ATJ: metavliti gia tin FROM IP
    const char *ipfrom = inet_ntoa(ip->ip_src);



    struct ether_header *eptr;  // ATJ: Domh gia tin mac address


   eptr = (struct ether_header *) packet;

   fprintf(stdout,"    MAC From: %s\n",
               ether_ntoa((struct ether_addr*)eptr->ether_shost));
   fprintf(stdout,"    MAC   To: %s\n",
               ether_ntoa((struct ether_addr*)eptr->ether_dhost));

    char *s = ether_ntoa((struct ether_addr*)eptr->ether_dhost);
       char dassmac[30];
     int a, b, c, d, e, f;
       sscanf(s, "%X:%X:%X:%X:%X:%X", &a, &b, &c, &d, &e, &f);
       sprintf(dassmac, "%02X:%02X:%02X:%02X:%02X:%02X", a, b, c, d, e, f);
       //printf(o);

    //const char *dassmac = ether_ntoa((struct ether_addr*)eptr->ether_dhost); //ATJ: metavliti gia tin
FROM MAC



    /* determine protocol */
    switch(ip->ip_p) {
       case IPPROTO_TCP:
          printf("   Protocol: TCP\n");
          return; //ATJ: return if TCP
       case IPPROTO_UDP:
          printf("   Protocol: UDP\n");
          break; //ATJ: and break if UDP
       case IPPROTO_ICMP:
          printf("   Protocol: ICMP\n");
          return;
       case IPPROTO_IP:
```

```
            printf("   Protocol: IP\n");
            return;
        default:
            printf("   Protocol: unknown\n");
            return;
    }

    /*
     *  ATJ: OK, this packet is UDP now...
     */

    /* define/compute tcp header offset */

    udp = (struct sniff_udp*)(packet + SIZE_ETHERNET + SIZE_UDP);

    //ATJ: exoume udp twra, no need for the below
    /*size_tcp = TH_OFF(tcp)*4;
    if (size_tcp < 20) {
        printf("   * Invalid TCP header length: %u bytes\n", size_tcp);
        return;
    } */

    printf("   Src port: %d\n", ntohs(udp->uh_sport));
    printf("   Dst port: %d\n", ntohs(udp->uh_dport));

    /* define/compute tcp payload (segment) offset */
    payload = (u_char *)(packet + SIZE_ETHERNET + size_ip + SIZE_UDP);

    char search_for[]="INVITE"; //ATJ: TERM TO SEARCH IN PAYLOAD
    char search_ok[]="SIP/2.0 200 OK"; //ATJ: string gia na ksexorizoume ta 200 OK paketa



    if ((strstr(payload, search_for)) ) { //ATJ: if search_for is found in payload, then should print it ;)

    /* compute udp payload (segment) size */
    size_payload = ntohs(ip->ip_len) - (size_ip + SIZE_UDP);
        if (size_payload > ntohs(udp->uh_ulen))
            size_payload = ntohs(udp->uh_ulen);


    /*
     * Print payload data; it might be binary, so don't just
     * treat it as a string.
     */

    eidos = 1; //ATJ: arxikopoihsh tis metavlitis - otan eidos = 1 shmainei oti to paketo einai INVITE
aplo
    if (strstr(payload, search_ok)) {

        eidos = 2; //ATJ: an to payload periexei to string "search_ok" tote einai 200 OK minima
    }


    if (size_payload > 0) {
        //printf("   Payload (%d bytes):\n", size_payload);
        //print_payload(payload, size_payload); ATJ : comment sto "poluploko print"
        printf("   Packet Data :\n%s \n", payload); //ATJ: aplo print, mono ta ascii
```

```
        // char str[] ="Mozilla/5.0 Gecko/20101206 Linux Error/10 (Julia) Firefox/3.6.13"; //ATJ: test
string
          if ((!strcmp(ipfrom, "83.212.239.221")!=0) && (strcmp(dassmac, "00:0F:FE:4D:45:A5")!=0))
{
          char * pinakas[3][50]; //ATJ: pinakas pou tha krataei ta data poy theloume
          char *tch;
               char *saved;
               char *array[50];
               int count = 0, i;

               tch = strtok (payload, "<:;>\n");
               while (tch != NULL)
               {
                  int savenext = 0;
                  if (!strcmp (tch, "sip"))
               {
                  savenext = 1;
                  }
                 // printf ("%s\n", tch);
                 tch = strtok (NULL, "<:;>\n");
                 if (savenext == 1)
              {
               saved = tch;
              }

                 if (count == 0)
                 {
               array[count] = saved;
               count++;
                 }
                 else if ((count > 0) && (savenext == 1))
                 {
                  int i = 0;
                  while (i < count)
                  {
                 if ((strcmp (array[i], saved) == 0) && ( i > 10 ))
                 {

                  printf ("FOUND!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
                 char iptables[200] ;
                 sprintf(iptables, "iptables -A INPUT -m mac --mac-source %s -j DROP", dassmac);
                printf("iiiiiiiiiiiii %s iiiiiiiiiiiiii\n",iptables);
                               system(iptables);
                  break;
                 }
                 i++;
                  }
                  if (i == count)
                  {
                 array[count] = saved;
                 count++;
                  }

                 }
                  }

                for (i = 0; i < count; i++)
                    printf ("\n@@@@@@@ %s @@@@@@@@", array[i]);
             for (i = 0; i < count; i++)
                 array[i]= "";
```

```
            }
                    }

          }
            if ( coot == 50 ) { coot = 0; } //ATJ: otan o pinakas ftasei sto 50, mhdenizoume kai pame apo
tin arxh



return;
}
int main(int argc, char **argv)
{

   char *dev = NULL;          /* capture device name */
   char errbuf[PCAP_ERRBUF_SIZE];      /* error buffer */
   pcap_t *handle;            /* packet capture handle */

   char filter_exp[] = "port 4060";    /* filter expression [3] */
   struct bpf_program fp;         /* compiled filter program (expression) */
   bpf_u_int32 mask;           /* subnet mask */
   bpf_u_int32 net;           /* ip */
   int num_packets = -1;          /* number of packets to capture */
   //set to -1 for infite loops of pcap_loop()

   print_app_banner();

   /* check for capture device name on command-line */
   if (argc == 2) {
      dev = argv[1];
   }
   else if (argc > 2) {
      fprintf(stderr, "error: unrecognized command-line options\n\n");
      print_app_usage();
      exit(EXIT_FAILURE);
   }
   else {
      /* find a capture device if not specified on command-line */
      dev = pcap_lookupdev(errbuf);
      if (dev == NULL) {
         fprintf(stderr, "Couldn't find default device: %s\n",
            errbuf);
         exit(EXIT_FAILURE);
      }
   }

   /* get network number and mask associated with capture device */
   if (pcap_lookupnet(dev, &net, &mask, errbuf) == -1) {
      fprintf(stderr, "Couldn't get netmask for device %s: %s\n",
         dev, errbuf);
      net = 0;
      mask = 0;
   }

   /* print capture info */
   printf("Device: %s\n", dev);
   printf("Number of packets: %d\n", num_packets);
   printf("Filter expression: %s\n", filter_exp);

   /* open capture device */
```

```
    handle = pcap_open_live(dev, SNAP_LEN, 1, 1000, errbuf);
    if (handle == NULL) {
        fprintf(stderr, "Couldn't open device %s: %s\n", dev, errbuf);
        exit(EXIT_FAILURE);
    }

    /* make sure we're capturing on an Ethernet device [2] */
    if (pcap_datalink(handle) != DLT_EN10MB) {
        fprintf(stderr, "%s is not an Ethernet\n", dev);
        exit(EXIT_FAILURE);
    }

    /* compile the filter expression */
    if (pcap_compile(handle, &fp, filter_exp, 0, net) == -1) {
        fprintf(stderr, "Couldn't parse filter %s: %s\n",
            filter_exp, pcap_geterr(handle));
        exit(EXIT_FAILURE);
    }

    /* apply the compiled filter */
    if (pcap_setfilter(handle, &fp) == -1) {
        fprintf(stderr, "Couldn't install filter %s: %s\n",
            filter_exp, pcap_geterr(handle));
        exit(EXIT_FAILURE);
    }

    /* now we can set our callback function */
    pcap_loop(handle, num_packets, got_packet, NULL);

    /* cleanup */
    pcap_freecode(&fp);
    pcap_close(handle);

    printf("\nCapture complete.\n");

return 0;
}
```

ΠΑΡΑΡΤΗΜΑ 8: ΚΑΝΟΝΑΣ ΣΤΟ IPTABLES

```
#!/bin/sh
sudo iptables -F
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A INPUT -p udp --dport 4060 -m string --string "REGISTER" -j ACCEPT --algo bm
sudo iptables -A INPUT -p udp --dport 4060 -j DROP
```