ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Ψηφιακών Συστημάτων

ΜΠΣ Τεχνοοικονομική Διοίκηση & Ασφάλεια Ψηφιακών Συστημάτων

Μηχανισμός εντοπισμού κακοσχηματισμένων SIP μηνυμάτων

Μεταπτυχιακή Διπλωματική εργασία

Επιβλέπων καθηγητής: Λαμπρινουδάκης Κωνσταντίνος

Συντάκτης: Αυγού Ελένη

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή μελέτη επικεντρώνεται στις αδυναμίες του SIP πρωτοκόλλου που χρησιμοποιείται για την επικοινωνία στο IMS (IP Multimedia Subsystem) και στο VOIP, και τις απειλές που μπορούν να προκύψουν από κακόβουλους χρήστες.

Ειδικότερα μελετάται η απειλή που προκύπτει από κακοσχηματισμένα (malformed) SIP μηνύματα και παρουσιάζεται η υλοποίηση ενός μηχανισμού που είναι σε θέση να διαβάσει αυτά τα μηνύματα, να βρει τις ανωμαλίες τους και να αποφασίσει αν μπορούν να προκαλέσουν ζημιά στον parser που θα τα επεξεργαστεί και θα τα προωθήσει στον κατάλληλο προορισμό. Αξιοποιώντας το μοντέλο αυτό και εντάσσοντάς το σε ένα IMS σύστημα μπορούμε να αποφύγουμε δυσλειτουργίες ή ακόμα και κατάρρευση του συστήματος που δημιουργούνται από κακόσχηματισμένα μηνύματα καθώς θα μπορούσε να εμποδίσει την μεταφορά τους.

Πιο αναλυτικά, στα πρώτα κεφάλαια γίνεται αναφορά στο IP Multimedia Subsystem καθώς και στο Voice Over IP για να γίνει κατανοητή η ευρέως διάδοσή τους και η ανάγκη για ασφάλεια της επικοινωνίας μέσω του SIP πρωτοκόλλου που χρησιμοποιούν. Στη συνέχεια παρουσιάζεται η λειτουργία του SIP πρωτοκόλλου, τα στοιχεία από τα οποία αποτελείται και πως συμβάλλουν στην ανταλλαγή μηνυμάτων μεταξύ των χρηστών. Γίνεται εκτενής ανάλυση των μηνυμάτων που ανταλλάσονται και ποια θα πρέπει να είναι η δομή τους, ανάλογα με την κατηγορία στην οποία ανήκουν, ώστε να αξιοποιηθούν στη συνέχεια στον μηχανισμό που θα παρουσιαστεί. Τέλος, αναφέρονται οι απειλές στις οποίες είναι εκτεθιμένο το SIP πρωτόκολλο λόγω των αδυναμιών που έχει και πώς αυτές έχουν οδηγήσει σε σοβαρές επιθέσεις. Στον μηχανισμό που παρουσιάζεται αξιοποιούνται τα μηνύματα κυρίως του Protos Test Suite για να επαληθευτεί η εγκυρότητά του στον εντοπισμό κακοσχηματισμένων μηνυμάτων. Παρουσιάζονται πιο αναλυτικά 3 test cases μηνυμάτων που θα μπορούσαν να προκαλέσουν πρόβλημα στον parser και πώς εντοπίζονται από τον μηχανισμό αυτόν.

ABSTRACT

This postgraduate study focuses on the weaknesses of the SIP protocol is used for communication in IMS (IP Multimedia Subsystem) and VOIP, and threats can messages arise from malicious users .

Specifically, we study the threat resulting from malformed SIP messages and present an implementation of a mechanism that is able to read these messages , find anomalies , and to decide whether they can cause damage to the parser that will read and forward them to the appropriate destination . Utilizing this model and integrating it into an IMS system can result in avoiding failures or even system crash caused by malformed as it could block the transfer.

More specificaly , in the first chapters we refer to the IP Multimedia Subsystem as well as Voice Over IP to mention the widely spread and the need for security of communication via SIP protocol use . Then, we analyze the SIP protocol, the elements of which it is composed and how it contributes to the exchange of messages between users. Extensive analysis of messages exchanged, the structure that they should follow, depending on the category to which they belong, will be presented. Finally, we present the threats to which the SIP protocol is exposed due to its weaknesses and how these have led to serious attacks. The mechanism which is presented is exploiting the messages of Protos Test Suite to verify its validity in detecting malformed messages. 3 test cases of messages that could cause problem to the parser and how they are detected by this mechanism are shown in the end.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ ΚΑΙ ΔΙΑΓΡΑΜΜΑΤΩΝ

1. ΕΙΣΑΓΩΓΗ

Το Voice Over IP (VOIP) χαρακτηρίζει μια ομάδα πρωτοκόλλων που επιτρέπουν τη φωνητική επικοινωνία μέσω διαδικτύου σε πραγματικό χρόνο. Οι πάροχοι υπηρεσιών VOIP έχουν προτείνει διάφορα πρωτόκολλα επικοινωνίας στο πέρασμα των χρόνων που χρησιμοποιούσαν συνήθως ιδιόκτητες διεπαφές. Τα πιο γνωστά παραδείγματα VOIP επικοινωνίας είναι το SCCP πρωτόκολλο που αναπτύχθηκε από τη CISCO και το Skype.

Όπως το ίδιο το όνομα υπονοεί, το Voice over IP (VoIP) χρησιμοποιεί το Internet Protocol (IP) για να στέλνει/λαμβάνει φωνή σε μορφή πακέτων δεδομένων «πάνω» από (over) ένα δίκτυο IP. Με τη χρησιμοποίηση πρωτοκόλλων VoIP, η μετάδοση φωνής μπορεί να επιτευχθεί σε οποιοδήποτε δίκτυο IP ανεξάρτητα από το αν πρόκειται για το Internet, Intranets ή τοπικά δίκτυα (LANs).

Σε ένα VoIP δίκτυο, το σήμα φωνής αρχικά ψηφιοποιείται, συμπιέζεται και μετατρέπεται σε πακέτα IP και στη συνέχεια στέλνεται μέσω του δικτύου IP. Ένα πρωτόκολλο σηματοδοσίας VoIP (signaling protocol) χρησιμοποιείται για να εγκαθιστά και να τερματίζει τις κλήσεις, να φέρει τις πληροφορίες που απαιτούνται για τον εντοπισμό των χρηστών και να διαπραγματεύεται τους πόρους του δικτύου (όπως, για παράδειγμα, το εύρος ζώνης).

Το 1996, πρωτοεμφανίστηκε το SIP πρωτόκολλο (Session Initiation Protocol) από την IETF MMUSIC Working Group και αρχικά χρησιμοποιήθηκε για τηλεδιασκέψεις. Στη συνέχεια γνώρισε μεγάλη αποδοχή ως ένα ανοικτό πρότυπο που μπορούσε να χρησιμοποιηθεί για την έναρξη, τροποποίηση και τερματισμό μιας αλληλεπιδραστικής συνεδρίας του χρήστη που περιλαμβάνει στοιχεία πολυμέσων όπως βίντεο, φωνή, μηνύματα πραγματικού χρόνου (instant messaging), online παιχνίδια και εικονική πραγματικότητα.

Σημαντική είναι η χρησιμότητά του και στο IP Multimedia Subsystem ή αλλιώς IP Multimedia Core Network Subsystem (IMS). Είναι μια αρχιτεκτονική που σχεδιάστηκε για την διανομή υπηρεσιών πολυμέσων. Καθώς αποτελεί τον πυρήνα του Next Generation Network (NGN) η ασφάλεια της επικοινωνίας και σε αυτά τα συστήματα είναι ιδιάζουσας σημασίας. Το SIP πρωτόκολλο χρησιμοποιείται και εδώ για την επικοινωνία των χρηστών.

Στόχος του SIP πρωτοκόλλου είναι να επιτευχθεί επικοινωνία μέσω συσκευών πολυμέσων και η λειτουργία του οφείλεται σε δύο πρωτόκολλα: το RTP/RTCP και το SDP. Το RTP πρωτόκολλο (Real-time Transport Protocol) χρησιμοποιείται για τη μεταφορά δεδομένων φωνής σε πραγματικό χρόνο (όμοια με το πρωτόκολλο H.323), ενώ το SDP πρωτόκολλο χρησιμοποιείται ως πρωτόκολλο σήμανσης για τη δημιουργία και τον τερματισμό συνεδριών.

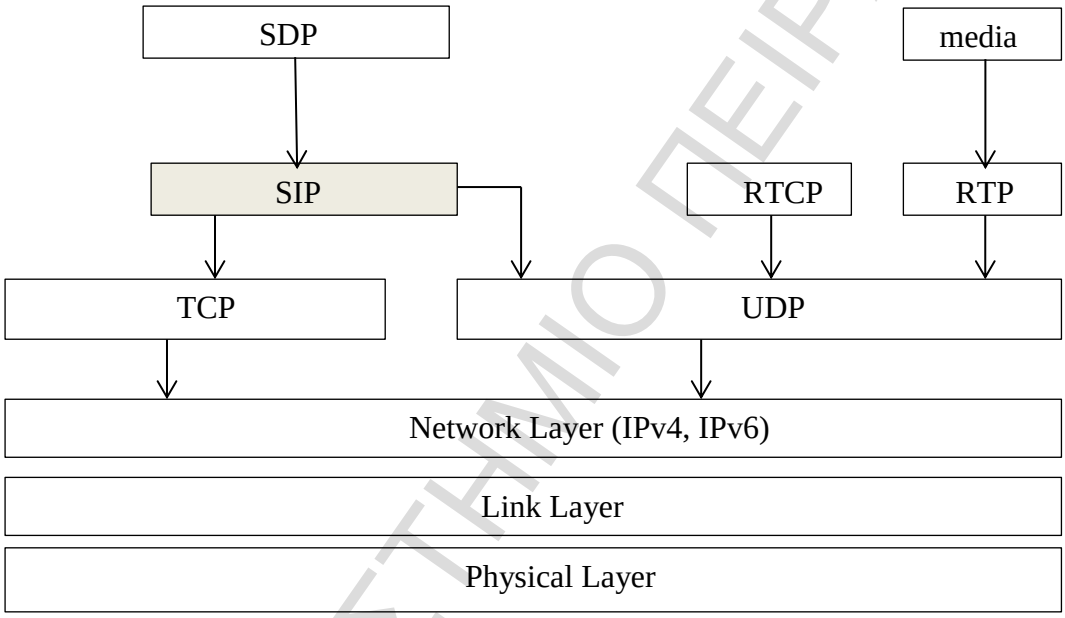
Το VOIP και η IMS τεχνολογία είναι εκτεθιμένα σε πλήθος κακόβουλων επιθέσεων κυρίως λόγω του γεγονότος ότι αξιοποιούνται σε ένα ανοικτό περιβάλλον όπως είναι το Διαδίκτυο. Οι επιθέσεις που έχουν παρατηρηθεί έχουν ως στόχο τόσο τα πρωτόκολλα σηματοδοσίας όπως τα SIP, H.323, MGC όσο και τα πρωτόκολλα μεταφοράς όπως είναι το RTP. Σε αυτές οι τεχνολογίες δεν είχε δοθεί έμφαση στα χαρακτηριστικά ασφαλείας κατά την κατασκευή τους με αποτέλεσμα να αποτελεί μια πρόκληση για κακόβουλους χρήστες να εκμεταλλευτούν τα κενά ασφαλείας:

Για αυτό το λόγο χρειαζόμαστε ένα μοντέλο που θα είναι ικανό να εντοπίζει την κακόβουλη συμπεριφορά και να κάνει έλεγχο για κενά ασφαλείας. Στο μοντέλο αυτό πρέπει να δώσουμε έμφαση κυρίως στον τρόπο με τον οποίο διαβάζονται τα μηνύματα και στη ροή των μηνυμάτων κατά τη επικοινωνία με νόμιμους χρήστες.

2. SIP Πρωτοκολλο

2.1. Το SIP στο Δίκτυο

Το SIP πρωτόκολλο είναι επιπέδου εφαρμογής και σχεδιάστηκε έτσι ώστε να είναι ανεξάρτητο από το επίπεδο μεταφοράς. Σκοπός ήταν να διευκολύνει τις συνεδρίες με χαμηλή καθυστέρηση και χρήση πολυμέσων μεταξύ πολλών χρηστών. Μπορεί να τρέξει σε TCP ,UDP ή SCTP. Η αρχιτεκτονική του SIP πρωτοκόλλου φαίνεται στο παρακάτω διάγραμμα:



Εικόνα 1. Αρχιτεκτονική SIP πρωτοκόλλου

Το SIP συνήθως τρέχει πάνω σε UDP (User Datagram Protocol) ή TCP (Transmission Control Protocol), αλλά μπορεί επίσης να τρέξει πάνω από IPX (Internet Packet Exchange), πλάισια αναμετάδοσης και X.25. Το UDP επιτρέπει στην εφαρμογή να ελέγξει πιο εύκολα τη χρονική στιγμή μετάδοσης των μηνυμάτων και της αναμετάδοσής τους, για να πραγματοποιούνται παράλληλες αναζητήσεις χωρίς να απαιτείται σύνδεση TCP για κάθε ένα request. Το TCP είναι χρήσιμο όταν ο User Agent πρέπει να χειριστεί μεγάλα μηνύματα.

Το SIP είναι ένα text-based πρωτόκολλο που ενσωματώνει πολλά στοιχεία του Hypertext Transfer Protocol (HTTP) και του Simple Mail Transfer Protocol (SMTP) . Είναι υπεύθυνο για τη δημιουργία, την τροποποίηση και τον τερματισμό συνεδριών

όπως είναι οι κλήσεις μέσω Internet. Με το SIP πρωτόκολλο ένας χρήστης μπορεί να προσκαλέσει και τρίτους σε μια ήδη υπάρχουσα συνεδρία (multi-cast conferences). Επίσης μπορούν να προστεθούν καθώς και να αφαιρεθούν πολυμέσα σε μια ήδη υπάρχουσα συνεδρία. Κατά την εγκατάσταση μιας συνεδρίας ο κάθε χρήστης πρέπει να τοποθετηθεί σε μια διεύθυνση ακόμα και αν έχει κάθε φορά δυναμική IP.

2.2. Πρωτόκολλα που περιβάλλουν το SIP

Στις αρχιτεκτονικές πολυμέσων που χρησιμοποιείται το SIP πρωτόκολλο συνήθως ενσωματώνονται και άλλα πρωτόκολλα όπως το Real-Time Transport Protocol (RTP) για μεταφορά δεδομένων σε πραγματικό χρόνο, το Real-Time Control Protocol (RTCP) για να λαμβάνει ενημέρωση για την ποιότητα της υπηρεσίας, το Real-Time Streaming Protocol (RTSP) για να ελέγχει την παραλαβή των δεδομένων ροής και το Session Description Protocol (SDP) για την περιγραφή μιας συνεδρίας πολυμέσων.

Παρόλα αυτά η λειτουργία του SIP πρωτοκόλλου δε βασίζεται σε κανένα από τα πρωτόκολλα που αναφέρθηκαν.Ας δούμε λίγο πιο αναλυτικά τα πρωτόκολλα που περιβάλλουν το SIP και πως συνδέονται με αυτό.

- **Internet Protocol - IP**

Παρότι όπως έχουμε αναφέρει το SIP είναι ανεξάρτητο από το πρωτόκολλο που χρησιμοποιείται στο επίπεδο δικτύου (δηλαδή το IP), στην πράξη βλέπουμε ότι το Internet Protocol (IP) είναι αυτό που αναλαμβάνει τη μεταφορά των μηνυμάτων από την πηγή στον παραλήπτη, όπως κάνει και με οποιαδήποτε άλλη κίνηση στο Διαδίκτυο.

Το IP είναι packed-based πρωτόκολλο και χρησιμοποιείται για την ανταλλαγή δεδομένων σε δίκτυα υπολογιστών. Όλα τα άλλα πρωτόκολλα δημιουργήθηκαν πάνω σε αυτό. Είναι επίσης υπεύθυνο για τη διευθυνσιοδότηση των πακέτων που ανταλλάσονται στο Διαδίκτυο.

Το IP είναι “connectionless” πρωτόκολλο, δηλαδή η επικοινωνία των δύο άκρων μπορεί να γίνει χωρίς να υπάρχει εγκατεστημένη σύνδεση μεταξύ τους. Κάθε πακέτο που μεταφέρεται στο Διαδίκτυο είναι ανεξάρτητο από τα υπόλοιπα και αντιμετωπίζεται ως μια ξεχωριστή οντότητα δεδομένων.

Οι εκδόσεις που χρησιμοποιούνται ευρέως είναι η IPv4 καθώς και η νεότερη IPv6.

- **Session Description Protocol - SDP**

Το πρωτόκολλο SDP (Session Description Protocol)περιγράφεται λεπτομερώς στο RFC 2327. Είναι ένα πρωτόκολλο περιγραφής σύνταξης και χρησιμοποιείται για να περιγράψει multicast συνόδους. Αρχικά χρησιμοποιήθηκε στο Session Announcement Protocol (SAP) για να δημοσιεύσει συνόδους που χρησιμοποιούν το Internet Multicast Backbone (MBONE). Τα μηνύματα SAP περιέχουν ένα μήνυμα SDP και αποτέλεσαν το πρότυπο για την χρήση του SDP στο SIP. Αν και σχεδιάστηκε για multicast, το SDP εφαρμόστηκε για τη λύση του γενικότερου προβλήματος της περιγραφής των πολυμεσικών συνόδων που εγκαθίστανται χρησιμοποιώντας το SIP.

- **UDP/TCP**

Τόσο το Transmission Control Protocol (TCP) οσο και το User Datagram Protocol (UDP) ανήκουν στη οικογένεια IP πρωτοκόλλου που αναφέρθηκε παραπάνω. Καθώς το SIP δεν απαιτεί κάποιο αξιόπιστο πρωτόκολλο μεταφοράς, ο User Agent μπορεί να χρησιμοποιήσει το UDP αντί για το TCP. Ωστόσο τα δύο άκρα πρέπει να υποστηρίζουν και τα δύο αυτά πρωτόκολλα.

Το TCP είναι ένα πρωτόκολλο που βασίζεται στη συνδεσιμότητα και είναι αξιόπιστο. Η σύνδεση αυτή επιτυγχάνεται με τη λεγόμενη 3-μερή χειραψία (3-way handshake). Το TCP είναι ένα αξιόπιστο πρωτόκολλο καθώς εγγυάται την παραλαβή των πακέτων που αποστέλλονται. Χρησιμοποιεί μια τεχνική που ονομάζεται Positive Acknowledgment with Retransmission (PAR) και ουστιαστικά ενημερώνει τον αποστολέα αν τα πακέτα παραλήφθησαν. Στην περίπτωση που δεν λάβει τέτοια ενημέρωση τα πακέτα ξαναστέλνονται. Επίσης, είναι υπεύθυνο για τον τεμαχισμό των μηνυμάτων σε πακέτα ώστε να μπορεί να τα διαχειριστεί το IP και στην επανασυγκρότησή τους στο αρχικό μήνυμα μόλις φτάσουν στο άλλο άκρο.

Το UDP είναι ένα πρωτόκολλο που δεν εγγυάται αξιόπιστη επικοινωνία. Τα πακέτα UDP που αποστέλλονται μπορεί να φτάσουν στον παραλήπτη με λάθος σειρά, διπλά ή να μην φτάσουν καθόλου εάν το δίκτυο έχει μεγάλο φόρτο, καθώς δε διαθέτει κανένα μηχανισμό ελέγχου ή και τεμαχισμού των μηνυμάτων σε πακέτα και

επανασυγκρότησής τους. Η έλλειψη αυτών των μηχανισμών αυτών το καθιστούν γρήγορο και αποτελεσματικό για τις εφαρμογές που δεν απαιτούν αξιόπιστη επικοινωνία.

- **RTP/RTCP**

Το Real-Time Transport Protocol (RTP) παρέχει επικοινωνία από άκρο σε άκρο και είναι κατάλληλο για δεδομένα πραγματικού χρόνου όπως είναι το βίντεο και η φωνή. Οι υπηρεσίες που προσφέρει περιλαμβάνουν αναγνώριση του τύπου του ωφέλιμου φορτίου, ακολουθιακή αρίθμηση, χρονικό έλεγχο καθώς και έλεγχο παραλαβής των δεδομένων. Το RTP συνήθως χρησιμοποιείται πάνω από το UDP και αντισταθμίζει τα μειονεκτήματα που αναφέρθηκαν παραπάνω.

Το Real-Time Control Protocol (RTCP), που είναι αντίστοιχο του RTP, βασίζεται στην περιοδική μεταφορά των πακέτων ελέγχου σε όλους τους συμμετέχοντες μιας συνεδρίας. Αυτά τα πακέτα ελέγχου χρησιμοποιούν τους ίδιους μηχανισμούς ελέγχου παραλαβής με τα πακέτα δεδομένων. Το κύριο χαρακτηριστικό του RTCP είναι ότι παρέχει πληροφορίες για την ποιότητα της κατανομής των δεδομένων, δηλαδή με το RTCP όλοι οι συμμετέχοντες μπορούν να γνωρίσουν πόσο καλά μπορούν να λαμβάνουν δεδομένα πραγματικού χρόνου.

SIP MESSAGE	SIP MANAGED DATA
ETHERNET	ETHERNET
IPV4	IPV4
TCP/UDP	UDP
SIP	RTP
SDP	DATA

Εικόνα 2. Δομή ενός SIP Πακέτου

2.3. Στοιχεία του SIP Δικτύου

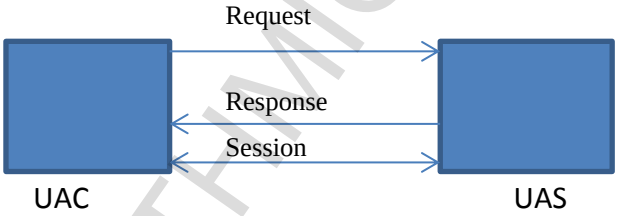
Για να καταλάβουμε καλύτερα το SIP πρωτόκολλο ας εξετάσουμε τα στοιχεία από τα οποία αποτελείται. Το πρωτόκολλο SIP χρησιμοποιεί ένα μοντέλο πελάτη-εξυπηρετητή, όπου ο πελάτης (client) αρχικοποιεί μια συνεδρία στέλνοντας ένα

αίτημα (request) και ο εξυπηρετητής (server) απαντά στο συγκεκριμένο αίτημα (response). Τα στοιχεία που χρησιμοποιεί ένα τέτοιο σύστημα είναι τα παρακάτω:

2.3.1. User Agent (UA)

Είναι μια εφαρμογή που αποτελείται τόσο από τον User Agent Client (UAC) όσο και από τον User Agent Server (UAS). Ο UAC είναι υπεύθυνος για τις εξερχόμενες κλήσεις ενώ ο UAS χειρίζεται τις εισερχόμενες. Η διάρκεια ζωής τους είναι για όσο διαρκεί η συνεδρία.

Στο SIP πρωτόκολλο, όπως και στο HTTP, ο User Agent μπορεί να ταυτοποιήσει τον εαυτό του χρησιμοποιώντας στα μηνύματα την κεφαλίδα 'User Agent', όπου μπορεί να περιέχει μια περιγραφή για το software, hardware ή ακόμα και για το προϊόν το οποίο χρησιμοποιείται στη συγκεκριμένη συνεδρία. Το πεδίο 'User Agent' στέλνεται στα request μηνύματα, κάτι που σημαίνει ότι ο SIP εξυπηρετητής που τα λαμβάνει είναι σε θέση να δει αυτήν την πληροφορία. Αυτή η πληροφορία συχνά αποθηκεύεται από τα SIP στοιχεία δικτύου και μπορεί να φανούν χρήσιμα στον εντοπισμό προβλημάτων ασυμβατότητας.



2.3.2. Proxy Server (PS)

Είναι ένα πρόγραμμα που προωθεί τις κλήσεις από τους User Agents σε άλλες τοποθεσίες. Ο Proxy Server είναι επίσης υπεύθυνος για τη δρομολόγηση των μηνυμάτων, την αυθεντικοποίηση, τις συναρτήσεις κοστολόγησης κ.λ.π.
Υπάρχουν δύο τύποι SIP proxy server, οι stateless και stateful.

- ✓ Stateful proxies:

Οι stateful όπως προδίδει και το όνομά τους, κρατάνε στοιχεία για κάθε SIP συνεδρία καθώς θυμούνται την κατάσταση του κάθε client που συμμετέχει.

Κύριος σκοπός των stateful proxies είναι να μπορεί να κάνει διακλάδωση (fork) των εισερχομένων αιτήσεων. Αυτό σημαίνει ότι κατά την λήψη ενός μηνύματος μπορεί να σταλούν δύο ή περισσότερα μηνύματα.

Το SIP επιτρέπει σε έναν χρήστη να έχει μια διεύθυνση για τη δουλειά του, μια για το σπίτι του και μια στο κινητό του. Ένας stateful proxy μπορεί να δρομολογήσει μια εισερχόμενη αίτηση και στις τρεις αυτές διευθύνσεις και να αποφασίσει ποια απάντηση είναι πιο ικανοποιητική για να προωθήσει την αίτηση σε αυτήν.

Με τους stateful proxies μπορούμε να αποφύγουμε την αναμετάδοση μηνυμάτων καθώς μπορούν να γνωρίζουν μέσω της κατάστασης που έχει αποθηκευτεί ότι το συγκεκριμένο μήνυμα έχει ήδη παραληφθεί.

Επίσης οι stateful proxies μπορεί να αλλάξουν σε stateless αν ισχυριστούν ότι έχει ολοκληρωθεί κάθε διαδικασία που θα απαιτούσε αποθήκευση κατάστασης.

- ✓ Stateless proxies:

Οι stateless proxies είναι πολύ πιο απλοί από τους stateful και παρέχουν ικανότητες δρομολόγησης και προώθησης για ταυτοποιημένους τελικούς χρήστες. Είναι πιο γρήγοροι από τους stateful proxy servers αλλά υστερούν στην αποφυγή αναμετάδοσης των μηνυμάτων καθώς και στην πραγματοποίηση κάποιας πιο περίπλοκης δρομολόγησης, όπως είναι η διακλάδωση (forking), που αναφέρθηκε παραπάνω, ή ακόμα και στην αναμετάδοση μηνυμάτων προς τα πίσω (recursive traversal).

2.3.3. Registrar Server

Είναι ο server που δέχεται τις αιτήσεις εγγραφής από τους user agents. Είναι επίσης υπεύθυνος για την αυθεντικοποίηση των αιτήσεων αυτών.

2.3.4. Redirection Server

Δέχεται τις αιτήσεις του SIP πρωτοκόλλου και δίνει σε αυτές είτε τη διεύθυνση μηδέν είτε άλλες νέες διευθύνσεις και τις στέλνει στον client.

3. SIP TRANSACTIONS

3.1. INVITE Transactions

Σχεδόν όλες οι επικοινωνίες μέσω SIP γίνονται χρησιμοποιώντας INVITE transactions που ξεκινούν όταν ένα από τα μέλη που λαμβάνουν μέρος στην επικοινωνία στέλνει ένα INVITE μήνυμα. INVITE transactions ονομάζεται ο μηχανισμός με τον οποίο τα τελικά μέρη μιας SIP επικοινωνίας ξεκινούν ή τροποποιούν συνεδρίες. Μια ολοκληρωμένη SIP συναλλαγή περιλαμβάνει έναν χρήστη που στέλνει ένα INVITE μήνυμα σε έναν ή περισσότερους άλλους χρήστες, αυτοί απαντούν με ένα μήνυμα OK, και τέλος ο αρχικός χρήστης απαντά με ένα ACK μήνυμα.

Ένα αίτημα INVITE περιέχει πεδία που ταυτοποιούν τον αποστολέα, τον παραλήπτη, τον μεσάζοντα και τους nonces. Όταν ο χρήστης στείλει ένα INVITE μήνυμα, αν ο δεύτερος χρήστης απαντήσει στην κλήση θα σταλεί ένα 200 OK μήνυμα πίσω στον αρχικό αποστολέα. Αυτό το δεύτερο μήνυμα περιέχει σχεδόν τα ίδια στοιχεία με το αρχικό INVITE μήνυμα.

Στη συνέχεια ο αρχικός αποστολέας θα απαντήσει με ένα ACK μήνυμα και θα εγκατασταθεί μια SIP συνεδρία πάνω από τις θύρες που έχουν ήδη καθοριστεί από τα προηγούμενα SDP μηνύματα. Το SDP (Session Description Protocol) περιγράφεται αναλυτικά RFC 2327 και είναι ένα format που περιγράφει τα generic objects. Στο SIP τα SDP formats χρησιμοποιούνται για να περιγράψουν στοιχεία που αφορούν τη συνεδρία, όπως ποιες θύρες, ποια πρωτόκολλα φωνής ή βίντεο κλπ. να χρησιμοποιηθούν στη συγκεκριμένη συνομιλία. Το INVITE και το ACK μήνυμα που περιγράψαμε προηγουμένως μπορεί να περιέχει SDP μήνυμα. Το μήνυμα 200 OK που θα έρθει ως απάντηση στο INVITE θα περιέχει επίσης SDP μήνυμα.

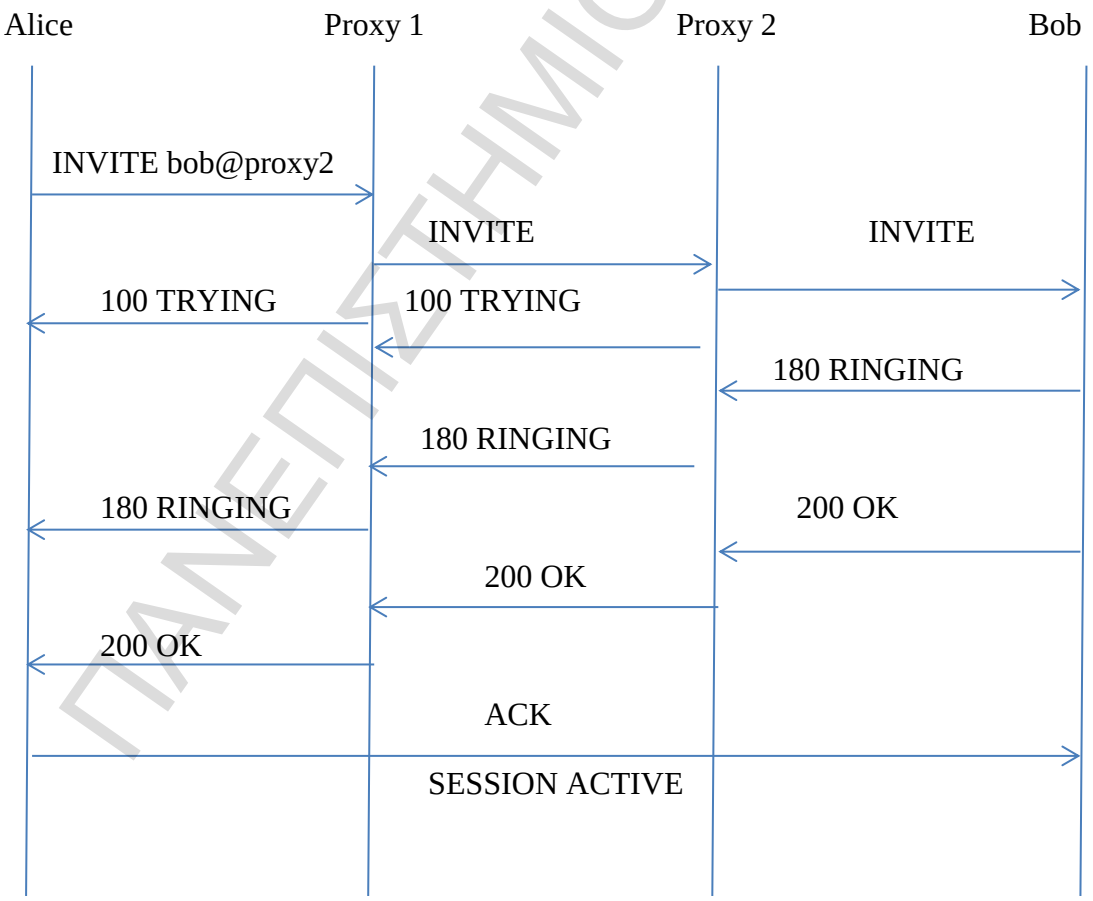
```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP
pc.atlanta.com:5060;branch=z9hG4bKnashd8;received=66.87.48.68
From: Alice <sip:alice@atlanta.com>;tag=192342987
To: Bob <sip:bob@biloxi.com>
Max-Forwards: 70
Call-ID: A84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc.atlanta.com>
Content-Type: application/sdp
```

```
Content-Length: 142

SDP Message <SDP message Contents>
```

Εικόνα 3. Sip Invite Request

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.30:5060;received=66.87.48.68
From: Bob <sip:bob@biloxi.com>
To: Alice <sip:alice@atlanta.com>;tag=192342987
Call-ID: A84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

SDP Message <SDP message Contents>
```

Εικόνα 4. Sip Response



Εικόνα 5. Διάγραμμα SIP Συνεδρίας

Εκτός από την εγκατάσταση μιας επικοινωνίας, τα INVITE μηνύματα χρησιμοποιούνται και σε ήδη υπάρχουσες συναλλαγές για την τροποποίηση αυτών. Αυτό μπορεί να σημαίνει, προσθήκη νέων συμμετεχόντων σε μια επικοινωνία, προσθήκη νέου καναλιού επικοινωνίας όπως βίντεο πάνω από την φωνή, ή ακόμα και τροποποίηση του πρωτοκόλλου σύμφωνα με το οποίο μεταφέρονται τα δεδομένα, πχ αύξηση του bandwidth της υπάρχουσας επικοινωνίας.

3.2. Registration Transactions

Για να υπάρξει μια SIP επικοινωνία πρέπει ο κάθε συμμετεχοντας να έχει ένα δημόσιο Uniform Resource Indicator (URI) , μια δημόσια διεύθυνση δηλαδή, όπως είναι το alice@atlanta.com ή το alice@192.168.234.1. Τα URIs περιγράφουν την τοποθεσία ενός proxy server. Στο παράδειγμα μας η Alice πρέπει να συνδεθεί με τη συγκεκριμένη IP διεύθυνση. Αυτό θα γίνει με την αποστολή ενός REGISTER αιτήματος στον proxy server που είναι υπεύθυνος για την Atlanta.com.

Σε μια τυπική επικοινωνία όταν ο client σηκώνει το τηλέφωνο στέλνει ένα REGISTER αίτημα στον registrar server χωρίς όμως κάποια στοιχεία αυθεντικοποίησης. Σε αυτήν την αίτηση ο server απαντάει αρνητικά και στέλνει ένα nonce ζητώντας στοιχεία αυθεντικοποίησης. Το σύστημα αυθεντικοποίησης που χρησιμοποιείται στις SIP συναλλαγές βασίζεται στο HTTP Digest Authentication και αξιοποιείται ως εξής:

Ο client στέλνει πίσω το nonce με το username του και ένα μίγμα από hashes που περιέχει το nonce, το username και ένα μυστικό κοινό password. Η σωστή αποκωδικοποίηση του hash ταυτοποιεί τον χρήστη στο server. Από τη στιγμή που ο χρήστης αυθεντικοποιηθεί στον server μπορεί να στέλνει INVITE αιτήματα μέσω του server αλλά και να παραλαμβάνει στον ταυτοποιημένο υπολογιστή του εισερχόμενα μηνύματα που θα του προωθούνται από τον server.

Εκτός από το HTTP Digest Authentication το SIP επιτρέπει στα τηλέφωνα να ταυτοποιούν τον server και μέσω Transport Layer Security (TLS) πριν αποσταλούν τα στοιχεία αυθεντικοποίησης, κάτι που προστατεύει από το eavesdropping (και ακολούθως από το brute force) του password με την κρυπτογράφησή του, καθώς επίσης αυθεντικοποιεί τον server στον SIP client μέσω των πληροφοριών που περιέχονται στο αυθεντικοποιημένο certificate.

Παρακάτω βλέπουμε ένα SIP Register μήνυμα:

```
REGISTER sip:unipi.gr SIP/2.0
From: sip:home@container.com
To: sip:office@unipi.gr
Via: SIP/2.0/UDP 135.180.130.133
Call-ID:0077@10.0.0.1
CSeq: 1 REGISTER
Contact: client@unipi.gr
Content-Length: 0
```

4. SIP ΜΗΝΥΜΑΤΑ

Τα SIP μηνύματα χωρίζονται σε μηνύματα αιτήσεων (Request) και απόκρισης (Response). Request είναι τα μηνύματα που όπως αναφέρθηκε και παραπάνω αποστέλλονται από τον client στον server ενώ response τα μηνύματα που αποστέλλονται από τον server στον client ως απάντηση στο αρχικό request μήνυμα. Και οι δύο αυτοί τύποι μηνυμάτων αποτελούνται από τη "start line" , την αρχική δηλαδή γραμμή του μηνύματος που καθορίζει και τον τύπο του, μια ή περισσότερες κεφαλίδες ("headers"), μια κενή γραμμή και προαιρετικά το σώμα ("body") του μηνύματος.

4.1. SIP Request

Το SIP Request μήνυμα ξεκινά με το όνομα της μεθόδου που είναι case sensitive. Παρακάτω φαίνονται οι διαθέσιμες μέθοδοι ενός SIP μηνύματος. Οι μέθοδοι που δεν υποστηρίζονται από έναν proxy και έναν redirect server κατατάσσονται στην κατηγορία OPTIONS και προωθούνται ανάλογα.

INVITE : Η αίτηση INVITE καλεί έναν χρήστη ή μια υπηρεσία να ξεκινήσει μια επικοινωνία.

ACK : Η μέθοδος ACK αποτελεί μια επιβεβαίωση ότι ο πελάτης έχει λάβει μια τελική απάντηση σε μια INVITE αίτηση. Στην πράξη η ACK μέθοδος χρησιμοποιείται μόνο σε συνδυασμό με την INVITE.

OPTIONS : Ένας User Agent στέλνει μια OPTIONS αίτηση σε κάποιον άλλον UA ή proxy server όταν θέλει να λάβει πληροφορίες για τις δυνατότητες και τα χαρακτηριστικά του. Η απάντηση που λαμβάνει αποτελείται από μια λίστα με κεφαλίδες όπως "Allow" και "Supported".

BYE : Η BYE αίτηση χρησιμοποιείται για τον τερματισμό μιας συνεδρίας και της επικοινωνίας που σχετίζεται με αυτήν. Η αίτηση αυτή μπορεί να σταλεί είτε από τον καλούμενο είτε από τον καλούντα.

CANCEL : Η CANCEL αίτηση ακυρώνει μια εκκρεμή INVITE αίτηση. Ωστόσο δεν επηρεάζει μια ολοκληρωμένη αίτηση ή ήδη υπάρχουσες συνεδρίες. Μια αίτηση θεωρείται ολοκληρωμένη όταν ο server επιστρέψει μια τελική απάντηση.

REGISTER : Ένα άκρο στέλνει μια REGISTER αίτηση για να καταχωρήσει τη θέση του σε έναν registar server.

4.2. SIP Response

Το response μήνυμα περιέχει έναν τριψήφιο κωδικό (“status code”) που περιγράφει το αποτέλεσμα της προσπάθειας του server να χειριστεί το request. Πολλοί από τους κωδικούς απάντησης στο SIP είναι ίδιοι με αυτούς του HTTP/1.1 πρωτοκόλλου. Κάθε SIP application που λαμβάνει το response μήνυμα δεν είναι αναγκαίο να αναγνωρίσει ολόκληρο το status code αρκεί να διακρίνει την κατηγορία στην οποία ανήκει, κάτι που φαίνεται από τον πρώτο αριθμό του τριψήφιου κωδικού. Παρακάτω φαίνονται οι κατηγορίες των status codes των sip response μηνυμάτων. Υπάρχουν δύο κατηγορίες status codes, οι final και οι non-final.

Οι non-final κωδικοί (1xx) χρησιμοποιούνται για πληροφοριακούς λόγους και δηλώνουν ότι ένα request μπορεί να πάρει χρόνο αλλά βρίσκεται στη διαδικασία της επεξεργασίας.

Όλοι οι υπόλοιποι κωδικοί είναι final και δηλώνουν το τέλος της συναλλαγής. Κάθε non-final κωδικός ακολουθείται από έναν final.

- **1xx:** Πληροφοριακές αποκρίσεις. Μια τέτοιου είδους απόκριση στέλνεται για να δείξει ότι το αίτημα έχει παραλειφθεί και υποβάλλεται σε επεξεργασία αλλά το αποτέλεσμα της επεξεργασίας αυτής δεν έχει γίνει ακόμα γνωστό. Αυτά τα μηνύματα στέλνονται μόνο όταν δεν τελειώσει απευθείας η επεξεργασία του αιτήματος. Ο αποστολέας πρέπει να μην ξαναστείλει το αίτημα από τη στιγμή που έχει λάβει ένα τέτοιο μήνυμα. Κατά βάση, ένας proxy στέλνει ένα 100 Trying μήνυμα απόκρισης με το που λάβει το αίτημα και αρχίσει να το επεξεργάζεται και ένας UA στέλνει ένα μήνυμα 180 Ringing όταν ο καλούμενος έχει λάβει το μήνυμα.
- **2xx**: Αποτελούν θετικές τελικές (final) αποκρίσεις προς τον αποστολέα του αιτήματος. Περιέχουν το αποτέλεσμα της επεξεργασίας του συγκεκριμένου αιτήματος και σηματοδοτούν επίσης το τέλος μιας συνεδρίας. Οι απαντήσεις αυτές παίρνουν τιμές από 200 έως 299 και σημαίνουν ότι το αίτημα επεξεργάστηκε και έγινε δεκτό με επιτυχία. Για παράδειγμα η απάντηση 200 OK σημαίνει ότι ένα INVITE αίτημα που στάλθηκε νωρίτερα έγινε δεκτό.

Αν ένας proxy κάνει διακλάδωση (fork) των εισερχομένων αιτήσεων, όπως περιγράφηκε παραπάνω, τότε στέλνει INVITE αιτήματα σε διάφορους UAS οπότε ο UAC στη συνέχεια θα λάβει πολλές 200 OK αποκρίσεις. Σε αυτήν την περίπτωση ξεχωρίζουμε την κάθε απόκριση από την παράμετρο tag στην κεφαλίδα To.

- **3xx:** Οι αποκρίσεις αυτές χρησιμοποιούνται για αναδρομολόγηση του server και είναι επίσης τελικές (final). Δίνουν πληροφορίες για τη νέα τοποθεσία του χρήστη ή για κάποια εναλλακτική υπηρεσία που μπορεί να χρησιμοποιήσει ο καλών για να ικανοποιήσει την κλήση. Αυτές οι αποκρίσεις συνήθως στέλνονται από proxy servers. Σε περιπτώσεις όπου ο proxy λάβει μια αίτησης που για οποιοδήποτε λόγο δεν μπορεί να την επεξεργαστεί, στέλνει στον καλούντα μια τέτοια απάντηση προτείνοντάς του μέσα σε αυτήν μια άλλη τοποθεσία που θα μπορούσε να χρησιμοποιήσει. Η τοποθεσία αυτή μπορεί να είναι κάποιος άλλος proxy ή η τοποθεσία του καλούμενου.
- **4xx**: Είναι αρνητικές τελικές αποκρίσεις. Μια τέτοια απόκριση σημαίνει ότι το πρόβλημα βρίσκεται στη μεριά του αποστολέα. Το αίτημα δεν μπόρεσε να επεξεργαστεί επιτυχώς λόγω κάποιου συντακτικού λάθους ή κάποιων κοινών συνθηκών που δεν μπόρεσαν να ικανοποιηθούν.
- **5xx**: Αυτές οι απαντήσεις σημαίνουν ότι υπάρχει κάποιο πρόβλημα στον server. Το αίτημα φαίνεται να είναι έγκυρο αλλά ο server για κάποιο λόγο δεν μπόρεσε να το κάνει δεκτό. Σε αυτές τις περιπτώσεις ο client συνήθως ξαναπροσπαθεί μετά από κάποιο χρονικό διάστημα.
- **6xx**: Αυτός ο κωδικός σημαίνει ότι το αίτημα δεν μπορεί να γίνει αποδεκτό σε κανέναν server. Αυτές οι αποκρίσεις στέλνονται από έναν server όταν έχει τελικές πληροφορίες για τον συγκεκριμένο χρήστη. Για παράδειγμα, οι UAs στέλνουν συνήθως μια 603 Decline απόκριση όταν ο χρήστης δε θέλει να συμμετάσχει σε μια συνεδρία.

5. ΔΟΜΗ ΕΝΟΣ SIP ΜΗΝΥΜΑΤΟΣ

Τα SIP μηνύματα έχουν συγκεκριμένη δομή. Η δομή του πρωτοκόλλου ακολουθεί το μοντέλο του HTTP πρωτοκόλλου. Κάθε συναλλαγή αποτελείται από ένα request που στέλνεται από τον client και περιλαμβάνει μια συγκεκριμένη μέθοδο ή μια συνάρτηση προς τον server και τουλάχιστον ένα response. Το SIP επαναχρησιμοποιεί τα περισσότερα πεδία κεφαλίδων, κανόνες κρυπτογράφησης και κωδικούς κατάστασης του HTTP, παράγοντας ένα αναγνώσιμο format.

Κάθε πόρος ενός SIP δικτύου, όπως για παράδειγμα ένας user agent ή ένα voice mailbox, ταυτοποιείται από ένα Uniform resource identifier (URI), που βασίζεται σε ένα γενικό συντακτικό το οποίο χρησιμοποιείται ευρέως και σε web services και σε email. Ένα SIP URI έχει την εξής μορφή sip:username:password@host:port

Αν απαιτείται ασφαλής συναλλαγή χρησιμοποιείται το sips: και δίνει εντολή ότι κάθε βήμα στο οποίο το request προωθείται πρέπει να γίνεται μέσω Transport Layer Security (TLS). Το τελευταίο βήμα, από τον proxy στον τελικό χρήστη πρέπει να ακολουθεί τις πολιτικές ασφαλούς μεταφοράς. Το TLS μπορεί να προστατέψει ενάντια σε αυτούς που προσπαθούν να ακούσουν από το link σηματοδοσίας. Δεν αποτελεί πραγματική από άκρη σε άκρη ασφάλεια καθώς η κωδικοποίηση γίνεται μόνο κατά τη μεταφορά οπότε κάθε ένας proxy πρέπει να είναι έμπιστος.

Αν συμβουλευτούμε το RFC 3261 θα δούμε ότι το SIP μήνυμα μπορεί να είναι είτε ένα Request (SIP method) είτε ένα Response (SIP state code) μήνυμα. Τα SIP μηνύματα κωδικοποιούνται σε απλό κείμενο, χρησιμοποιώντας UTF-8 κωδικοποίηση. Ένα SIP μήνυμα αποτελείται από 3 μέρη:

- **Γραμμή εκκίνησης (Start line):** Αυτή είναι η πρώτη γραμμή του μηνύματος και περιλαμβάνει τον τύπο του μηνύματος (μέθοδος στα request μηνύματα και κωδικός απάντησης-status code στα response μηνύματα. Για τα requests αυτή η γραμμή ονομάζεται Request-Line ενώ για τα responses Status-Line.

  Η Request-Line περιλαμβάνει ένα Request URI που ταυτοποιεί το χρήστη ή την υπηρεσία στην οποία απευθύνεται το μήνυμα.

Στην Status-Line περιλαμβάνεται ένας αριθμός κατάστασης (status code) που αντιστοιχεί σε μια απάντηση όπως περιγράφηκε παραπάνω.

- **Κεφαλίδες (Headers):** Οι κεφαλίδες ενός SIP μηνύματος χρησιμοποιούνται για να μεταφέρουν τα χαρακτηριστικά του μηνύματος και να τροποποιήσουν την έννοιά του. Είναι παρόμοια, συντακτικά και σημασιολογικά με τα πεδία κεφαλίδων του HTTP και του SMTP και ακολουθούν τη μορφή όνομα : τιμή. Παρακάτω φαίνονται οι βασικότερες κεφαλίδες που χρησιμοποιούνται στα SIP Request και Response μηνύματα.

  - *To* : Η διεύθυνση του παραλήπτη του αιτήματος
  - *From* : Η global διεύθυνση του αποστολέα
  - *Contact* : Το τρέχων URI όπου βρίσκεται ο αποστολέας
  - *CSeq* : Αποτελείται από έναν ακέραιο και το όνομα μιας μεθόδου, όπου ο ακέραιος χρησιμοποιείται για τον εντοπισμό της μη παραλαβής ενός μηνύματος ή την παραλαβή μηνυμάτων εκτός της ακολουθιακής σειράς. Κατά το ξεκίνημα της συνεδρίας ο αριθμός αυτός είναι τυχαίος και κατά την άφιξη ενός νέου μηνύματος αυξάνεται κατά ένα.
  - *Call-ID* : Ένας μοναδικός αριθμός ξεχωριστός για κάθε κλήση και περιέχει και την host διεύθυνση. Αυτή η κεφαλίδα χαρακτηρίζει μια συγκεκριμένη κλήση και είναι ίδια για κάθε μήνυμα της συγκεκριμένης συνεδρίας.
  - *Via* : Χρησιμοποιείται για να αποθηκεύσει όλες τις οντότητες από τις οποίες έχει περάσει ως τώρα το συγκεκριμένο αίτημα. Περιέχει το πρωτόκολλο μεταφοράς, όπως είναι το UDP ή το TCP, όπως επίσης και τη δρομολόγηση του αιτήματος ώστε να μπορεί να εντοπίσει τους βρόχους δρομολόγησης και να μπορεί να δρομολογήσει σωστά τα μηνύματα απάντησης απέναντι στον αποστολέα του αιτήματος.
  - *Route* : Περιέχει ένα σύντολο από hosts από τους οποίους πρέπει να περάσει ένα μήνυμα πριν φτάσει στον τελικό προορισμό του.

- *Record-Route* : Ένα σύνολο από ενδιάμεσες οντότητες από τις οποίες θα περάσει το μήνυμα στα επόμενα στάδια.

Στο κεφάλαιο 9 περιγράφουμε και τη μορφή που θα πρέπει να έχουν οι κεφαλίδες αυτές λίγο πιο αναλυτικά, ώστε τα στοιχεία αυτά να αξιοποιηθούν στο μοντέλο που θα παρουσιαστεί.

- **Σώμα (body/content)** : Το μέρος αυτό είναι προαιρετικό και χρησιμοποιείται για να περιγράψει την συνεδρία που θα ξεκινήσει. Για παράδειγμα, σε μια συνεδρία πολυμέσων μπορεί να περιέχει διάφορα είδη κωδικοποιητών βίντεο και ήχου, ποσοστά δειγματοληψίας κλπ. Εναλλακτικά μπορεί να περιέχει δεδομένα κειμένου ή δυαδικά που αφορούν κατά έναν τρόπο τη συνεδρία. Τα message bodies μπορούν να εμφανιστούν είτε σε request είτε σε response μήνυμα. Το SIP πρωτόκολλο κάνει σαφή διάκριση μεταξύ των πληροφοριών σηματοδοσίας που βρίσκονται στην γραμμή εκκίνησης και στις κεφαλίδες του SIP μηνύματος, και στις πληροφορίες περιγραφής της συνεδρίας, που είναι εκτός του πεδίου εφαρμογής του SIP.

5.1. SIPS

Το RFC 3261 προσφέρει και ένα επίπεδο ασφάλειας για τις SIP επικοινωνίες. Οι clients που θέλουν να υπάρξει κρυπτογράφηση των δεδομένων στην επικοινωνία τους θα πρέπει να βάλουν μπροστά από τη διεύθυνσή τους το αρχικό “sips:” αντί για το “sip:”. Το SIPS είναι μια άλλη μορφή URI που έχει σχεδιαστεί για να εξασφαλίσει το Transport Layer Security μεταξύ όλων των βημάτων της SIP επικοινωνίας. Ωστόσο, επειδή οι συμμετέχοντες στην επικοινωνία και οι servers ίσως δεν έχουν TLS capabilities δεν είναι εγγυημένη η TLS επικοινωνία από άκρο σε άκρο.

6. SESSION DESCRIPTION PROTOCOL - SDP

Το Session Description Protocol που περγράφεται λεπτομερώς στο RFC 4566 επιτρέπει στους συμμετέχοντες μιας συνεδρίας να ανταλλάξουν λεπτομέρειες για τη συγκεκριμένη συνεδρία όπως για το πρωτόκολλα τα οποίο χρησιμοποιούνται, τις διευθύνσεις δικτύου, τις θύρες κ.ά. Με αυτόν τον τρόπο και το SIP πρωτόκολλο χρησιμοποιεί το SDP για να περιγράψει τις δυνατότητες και τα μέσα που μπορούν να χρησιμοποιηθούν από τα δύο άκρα. Αυτή η πληροφορία παρέχεται στο σώμα του SIP μηνύματος που χαρακτηρίζεται ως μια οντότητα δεδομένων MIME (Multipurpose Internet Mail Extension) και περιγράφεται στο RFC 2045.

Το SDP είναι και αυτό ένα text-based πρωτόκολλο, όπως και το SIP, που χρησιμοποιεί κωδικοποίηση UTF-8. Περιέχει πληροφορίες για:

- Διεύθυνση IP (διεύθυνση IPv4 ή όνομα συστήματος)
- Αριθμός πύλης (που χρησιμοποιείται από το UDP ή το TCP για τη μεταφορά)
- Το χρόνο για τον οποίο η συνεδρία είναι ανοικτή (έναρξη και λήξη)
- Τα μέσα που χρησιμοποιούνται (ήχος, βίντεο κλπ)
- Σχήμα κωδικοποίησης πολυμέσων (PCM A-Law, MPEG II video, κτλ)
- Το σκοπό (θέμα) της συνεδρίας
- Πληροφορίες που χρειάζονται για να παραλάβει ο χρήστης αυτά τα μέσα, όπως θύρες και διευθύνσεις.

Το SDP χρησιμοποιεί συντακτικό της μορφής <type> = <value> όπου το <type> αναφέρεται σε μια συγκεκριμένη τιμή όπως φαίνεται και στον παρακάτω πίνακα. Η μορφή του <value> είναι ανάλογη με το <type> στο οποίο αναφέρεται.

Οι βασικοί τύποι του SDP:

Τύπος	Περιγραφή
v	Έκδοση του πρωτοκόλλου (Protocol version)
o	Ιδιοκτήτης της συνεδρίας (owner)
s	Ονομα της συνεδρίας (Session name)
i	Πληροφορίε συνόδου
u	URI. Ενιαίο αναγνωριστικό συνόδου
c	Πληροφορίες για τη σύνδεση (connection information).
t	Χρόνος για τον οποίο η συνεδρία είναι ανοικτή (time session active)
m	Όνομα του μέσου και διεύθυνση μεταφοράς (media name)
a	Περισσότερες πληροφορίες για τα χαρακτηριστικά των μέσων που χρησιμοποιούνται

*Οι τύποι c και a είναι προαιρετικοί

Εικόνα 6. Στοιχεία SDP πρωτοκόλλου

Παρακάτω φαίνεται ένα παράδειγμα SDP σε ένα πραγματικό SIP μήνυμα

```
v=0
o=user1 53655765 2353687637 IN IP4 128.3.4.5
s=Mbone Audio
t=3149328700 0
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Κάθε ένας από τους τύπους του SDP μηνύματος μπορεί να πάρει συγκεκριμένες τιμές και ακολουθούν μια συγκεκριμένη δομή. Παρακάτω περιγράφονται αναλυτικότερα τα στοιχεία αυτά ώστε να μπορέσουν να αξιοποιηθούν στο μοντέλο που θα υλοποιηθεί καθώς οι κανόνες του parsing στο πρωτόκολλο αυτό είναι πολύ αυστηροί.

Πεδία του SDP:

- Έκδοση πρωτοκόλλου

Το πεδίο αυτό συμβολίζεται με το γράμμα (v) και παίρνει ως τιμή την έκδοση του SDP πρωτοκόλλου. Προς το παρόν δεν υπάρχει κάποια άλλη έκδοση εκτός από τη 0 οπότε το πεδίο v=0 είναι η μόνη έγκυρη τιμή για ένα SDP μήνυμα.

- Το πεδίο Owner (προέλευση)

Συμβολίζεται με το γράμμα (o) και περιλαμβάνει πληροφορίες για τον ιδιοκτήτη της συνεδρίας και τα χαρακτηριστικά της. Η μορφή αυτού του πεδίου είναι:

o=username session-id version network-type address-type address
όπου,
username : το όνομα/αναγνωριστικό χρήστη ή του σταθμού εργασίας. Σε περίπτωση που καμία από αυτές τις πληροφορίες δεν είναι διαθέσιμη αντικαθίσταται με «-»,
session-id : αποτελεί το αναγνωριστικκό της συνόδου και είναι μια χρονοσφραγίδα σύμφωνη με το Network Time Protocol (NTP) ή ένας τυχαίος αριθμός που εξασφαλίζει τη μοναδικότητά της.
version : ένας αριθμός που αναφέρεται στην έκδοση της συνόδου καθώς αυξάνεται με κάθε αλλαγή που γίνεται σε αυτήν. Προτιμάται και εδώ να είναι μια χρονοσφραγίδα NTP.
network-type : συμβολίζει τον τύπο του δικτύου και παίρνει πάντα την τιμή IN για το διαδίκτυο
address-type : συμβολίζει τον τύπο της διεύθυνσης και παίρνει τιμές IP4 ή IP6 για διευθύνσεις τύπου IPv4 ή IPv6 αντίστοιχα.
Address : περιέχει μια IP διεύθυνση

- Το πεδίο όνομα συνόδου

Το πεδίο αυτό συμβολίζεται με το γράμμα (s) και περιέχει ένα όνομα για τη σύνοδο.

- Το πεδίο URI

Το πεδό αυτό είναι προαιρετικό και συμβολίζεται με το γράμμα (u). Περιέχει ένα URI με πληροφορίες για τη σύνοδο

- Το πεδίο Δεδομένων της Σύνδεσης

Συμβολίζεται με το γράμμα (c) και περιέχει πληροφορίες για τη σύνδεση. Έχει τη μορφή:

c=network-type address-type connection-address

Τα network-type και address-type ακολουθούν τις προδιαγραφές που περιγράψαμε παραπάνω για το πεδίο owner.
Η παράμετρος connection-address είναι η διεύθυνση IP, η οποία θα αποστέλλει τα πακέτα πολυμέσων και η οποία μπορεί να είναι είτε ulticast είτε unicast. Αν είναι multicast, το πεδίο connection-address έχει τη μορφή:
connection-address=base-multicast-address/ttl/number-of-addresses
όπου
ttl είναι η διάρκεια και number-of-addresses ο αριθμός των multicast διευθύνσεων που περιλαμβάνονται ξεκινώντας από την base-multicast-address.

7. ΕΠΙΘΕΣΕΙΣ ΣΤΟ SIP PROTOCOL

Από έρευνες που έχουν γίνει έως τώρα μπορούμε να κατηγοριοποιήσουμε τις επιθέσεις VOIP σε έξι κατηγορίες ανάλογα με το πρωτόκολλο και τη συμπεριφορά τους. Οι τρεις συνδέονται με το SIP πρωτόκολλο ενώ οι υπόλοιπες με το RTP.
Επειδή το SIP έχει να κάνει με την αρχικοποίηση, τη σύνδεση και τον τερματισμό μιας συνεδρίας, είναι πολύ σημαντικό να εξεταστούν με προσοχή. Οι επιθέσεις που γίνονται στο SIP πρωτόκολλο δεν μπορούμε να πούμε ότι είναι όμοιες με αυτές που γίνονται στο IP όπως είναι για παράδειγμα η επίθεση spoofing. Η προσοχή μας επικεντρώνεται στις επιθέσεις που έχουν να κάνουν με τα χαρακτηριστικά του SIP πρωτοκόλλου όπως είναι επίθεση κακοσχηματισμένων μηνυμάτων (malformed message attack) και η επίθεση πλημμύρας (flooding attack) .

- ✓ Επιθέσεις κακοσχηματισμένων μηνυμάτων

Αυτή η επίθεση είναι η πιο αντιπροσωπευτική καθώς είναι μια επιθέση που εκμεταλλεύεται τις αδυναμίες του text-based SIP πρωτοκόλλου. Οι επιτιθέμενοι καταφέρνουν να δημιουργήσουν δυσλειτουργίες στον proxy server καθώς και στον User-Agent «πειράζοντας» τις κεφαλίδες του SIP μηνύματος. Για παράδειγμα, τα πολλά κενά σε ένα μήνυμα , η έλλειψη τιμών στις κεφαλίδες του μηνύματος, η έλλειψη συγκεκριμένων υποχρεωτικών κεφαλίδων ή ακόμα και η χρήση χαρακτήρων που δεν έχουν ASCII και UTF-8 κωδικοποίηση, είναι μερικοί από τους τρόπους που μπορεί να δημιουργηθεί ένα κακοσχηματισμένο μήνυμα που θα προκαλέσει προβλήματα κατά το parsing.

- ✓ Επιθέσεις πλημμύρας

Στη συγκεκριμένη επίθεση τα IP τηλέφωνα δημιουργούν requests και responses για να αποσταλούν σε έναν συγκεκριμένο User-Agent . Στόχος είναι ο UA να λάβει τόσα πολλά SIP μηνύματα σε ένα περιορισμένο χρονικό διάστημα ώστε να μην είναι ικανός να τα επεξεργαστεί και να λειτουργήσει κανονικά. Η πιο συνηθισμένη περίπτωση μηνυμάτων που χρησιμοποιούνται σε τετοιες επιθέσεις είναι τα INVITE μηνύματα. Αυτή η επίθεση παρότι τη συναντάμε συχνά και στο IP στρώμα στην περίπτωση του VOIP η επίθεση πλημμύρας με INVITE μηνύματα μπορεί να γίνει

και πολύ ενοχλητική καθώς υπάρχει ένα συνεχές κουδούνισμα από το σύνολο των αιτήσεων που δέχεται ο χρήστης.

- ✓ Επιθέσεις σηματοδότησης (spoofing)

Υπάρχουν δύο είδη spoofing επίθεσης, αυτή που συνδέεται με το IP και αυτή που συνδέεται με το URI. Η IP spoofing επίθεση προσπαθεί να αλλάξει τη διεύθυνση IP ώστε να παρουσιαστεί ο επιτιθέμενος ως ένας νόμιμος χρήστης. Ωστόσο η IP spoofing επίθεση έχει να κάνει με το TCP/ IP πρωτόκολλο και όχι με το VOIP. Στην περίπτωση του VOIP η URI spoofing επίθεση είναι αυτή που μας ενδιαφέρει να εξετάσουμε περισσότερο καθώς ουσιαστικά αφορά μια συγκεκριμένου τύπου malformed επίθεση. Ο επιτιθέμενος προσπαθεί να αλλάξει το πεδίο URI ώστε να κρύψει τα ίχνη του. Αν για παράδειγμα γίνει spoofing σε ένα BYE request (BYE DOS επίθεση) η κλήση θα τερματιστεί από τον επιτιθέμενο.

Πέρα από τις SIP επιθέσεις υπάρχουν, όπως αναφέρθηκε και προηγουμένως, και οι RTP επιθέσεις που μπορούν να χωριστούν και αυτές σε τρεις κατηγορίες: στις RTP επιθέσεις πλημμύρας (flooding), στις επιθέσεις ενδιαμέσου (man-in-the-middle - MITM) και στις επιθέσεις media spamming. Οι RTP επιθέσει πλημμύρας είναι παρόμοιες με τις SIP επιθέσεις πλημμύρας αλλά χρησιμοποιούν RTP πακέτα. Οι media spamming επιθέσεις, γνωστές και ως SPIT (Spam over Internet Telephony) , έχουν να κάνουν με τις κλήσεις που λαμβάνει ο χρήστης για διαφημιστικούς λόγους χωρίς τη συγκατάθεσή του. Τέλος οι επιθέσεις ενδιαμέσου είναι παρόμοιες με την επίθεση eavesdropping και είναι από τις πιο κρίσιμες RTP επιθέσεις.

8. ΚΑΚΟΣΧΗΜΑΤΙΣΜΕΝΑ SIP ΜΗΝΥΜΑΤΑ

Η δομή των SIP μηνυμάτων όπως αναφέρθηκε παραπάνω έχει πολλά κοινά στοιχεία με την HTTP δομή. Μαζί με τη δομή κληρονομεί και πολλές αδυναμίες ασφάλειας του HTTP πρωτοκόλλου. Αυτό κάνει τους επίδοξους hackers να το προτιμούν από άλλα πρωτόκολλα όπως H.323, MGCP, SKINNY που έχουν πιο πολύπλοκη δομή. Γι' αυτό ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί τα τρωτά σημεία του SIP πρωτοκόλλου και να προκαλέσει ζημιά είτε σε έναν τελικό χρήστη είτε σε κάποιον SIP proxy server.

Όπως είναι λογικό χρειάζονται CPU πόροι για να διαβαστεί ένα εισερχόμενο SIP μήνυμα. Επίσης ο SIP proxy πρέπει να είναι σε θέση να αναλύσει τουλάχιστον μέρος του μηνύματος και να κάνει έναν έλεγχο συνοχής σε αυτό. Καθώς το SIP πρωτόκολλο βασίζεται στη δομή απλού κειμένου δίνει μεγάλη ευελιξία στα μηνύματά του και μπορεί να επιτρέψει τη δημιουργία ενός μηνύματος το οποίο παρά το ότι συντακτικά μπορεί να είναι σύμφωνο με τις προδιαγραφές του, να έχει σκοπό να παρεμποδίσει τη σωστή λειτουργία του server. Ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί τυχόν ελλείψεις στην υλοποίηση μιας SIP οντότητας και να κατασκευάσει ένα πακέτο που να δημιουργεί υπερβολική κατανάλωση πόρων με αποτέλεσμα την κατάρρευση του συστήματος ή την επανεκκίνησή του. Κατά βάση οι SIP parsers περιμένουν να λάβουν ένα «καλοφτιαγμένο» μήνυμα με αποτέλεσμα οποιαδήποτε παρεμβολή στη σωστή δομή του μηνύματος να προκαλεί προβλήματα. Μερικές τέτοιες επιθέσεις είναι οι παρακάτω επιθέσεις Parser:

1. Πολλαπλές τιμές στις κεφαλίδες
   1.1. Πολλαπλές τιμές σε μία κεφαλίδα ή
   1.2. Μια κεφαλίδα να εμφανίζεται πολλαπλές φορές.

   Αν για παράδειγμα, ένα μήνυμα διαθέτει πολλές φορές την κεφαλίδα From ή η κεφαλίδα From διαθέτει πολλαπλές τιμές, αυτό θα δυσκολέψει τον SIP parser να επεξεργαστεί το μήνυμα και μπορεί να προκαλέσει κάποια άγνωστη αποτυχία
2. Υπερβολικά μεγάλο μήνυμα
   2.1. SIP μηνύματα με πολύ μεγάλο μήκος.

2.2. Συνεχόμενη εμφάνιση συγκεκριμένων χαρακτήρων, υπερβολικά μεγάλων φιγούρων κλπ

3. Όταν δεν έχει UTF-8 κωδικοποίηση
   3.1. Ο SIP parser μπορεί να μην μπορεί να αναγνωρίσει αυτούς τους χαρακτήρες

Για να τα δούμε λίγο πιο αναλυτικά, ένας κακόβουλος χρήστης μπορεί να αυξήσει το μέγεθος ενός μηνύματος χρησιμοποιώντας κεφαλίδες που έχουν κυρίως πληροφορικό χαρακτήρα όπως είναι οι “Supported” και “Allow”. Αυτές οι κεφαλίδες μπορούν να χρησιμοποιηθούν σε συνδυασμό με ένα μεγάλου μεγέθους σώμα του μηνύματος μολονότι το σώμα δεν είναι υποχρεωτικό σε κάθε μήνυμα. Ένας καλοφτιαγμένος parser βέβαια μπορεί να αγνοήσει τις άγνωστες κεφαλίδες, οπότε μια καλή υλοποίηση είναι να χρησιμοποιούνται μονο οι κεφαλίδες που περιγράφονται στο RFC 3261. Ένα πολύ μεγάλο μήνυμα απαιτεί περισσότερη μνήμη, επεξεργαστική ισχύ και εύρος ζώνης δικτύου. Επίσης στα κακόβουλα μηνύματα μπορούμε να συμπεριλάβουμε και αυτό που περιλαμβάνει σώμα με μεγαλύτερο μέγεθος από αυτό που αναφέρεται στην κεφαλίδα Content-Length του μηνύματος. Κάτι τέτοιο μπορεί επίσης να προκαλέσει κατάρρευση του συστήματος.

Μια άλλη μέθοδος που μπορεί να παρεμποδίσει την διαδικασία ανάλυσης του μηνύματος είναι να γεμίσει το μήνυμα με την επανάληψη κεφαλίδων του ίδιου πεδίου σε πολλά σημεία του. Οι SIP προδιαγραφές επιτρέπουν τη διανομή συγκεκριμένων κεφαλίδων που έχουν πολλαπλές τιμές σε μοναδικά πεδία που περιέχουν μόνο μια τιμή. Τέτοιες κεφαλίδες είναι οι: Accept-Language, Allow, Contact, Route, Supported και Via.

Ένα παράδειγμα τέτοιας διανομής φαίνεται παρακάτω:

From:…		From:		Via:
To:….		To:		From:
Via:	ή	Via:	ή	Via:
Via:				To:
Via:				Via:

8.1. SQL injection

Εκτός από τις επιθέσεις που στοχεύουν απευθείας στον parser υπάρχουν και αυτές που στοχεύουν στη βάση δεδομένων του proxy. Παρακάτω μπορούμε να δούμε ένα παράδειγμα όπου ένας επιτιθέμενος μπορεί να στείλει κακόβουλο SQL κώδικα μέσα από μια κεφαλίδα Authorization. Με αυτόν τον τρόπο κάθε φορά που ένας proxy ζητά ταυτοποίηση και αναζητά σε βάση δεσομένων τα στοιχεία της ταυτοποίησης μπορεί να ξεγελαστεί από μια τέτοιου είδους επίθεση.

WWW-Authenticate: Digest realm="telestar.com";
Update subscriber set first_name=Bob
where username='Alice',
nonce="qD2yEcfg3RRO9BP2zKHbxayQemj7DwAAlfGkN3vjDRw=", algorithm=AKAv1-MD5

Είκονα 7. SQL injection

Στο παραπάνω παράδειγμα με τον κακόβουλο SQL κώδικα που στέλνεται μέσω της κεφαλίδας “WWW-Authenticate” ο επιτιθέμενος καταφέρνει να αλλάξει τον χρήστη με username Alice σε Bob.

8.2. SIP Parsers

Οι SIP parsers, όπως αναφέρθηκε, είναι σχεδιασμένοι για να δέχονται και να επεξεργάζονται «καλοφτιαγμένα» μηνύματα, δηλαδή μηνύματα που ακολουθούν τις προδιαγραφές που περιγράψαμε παραπάνω και είναι συμβατές με τη γραμματική που αναφέρεται στο RFC 3261. Ωστόσο ένας επιτιθέμενος μπορεί να στείλει πολλούς τύπους μη καλοφτιαγμένων μηνυμάτων με σκοπό να προκαλέσει μη επιθυμητές καταστάσεις όπως Denial of Service (DoS), ασταθή κατάσταση του server ή μη εξουσιοδοτημένη πρόσβαση. Αυτό μπορεί να συμβεί καθώς ο parser στον SIP proxy server δε θα είναι ικανός να χειριστεί επιτυχώς τα μηνύματα που του στέλνονται με αποτέλεσμα να τα απορρίπτει (drop). Για παράδειγμα κατά το ξεκίνημα μιας

συνεδρίας ένας επιτιθέμενος μπορεί να στείλει πολλαπλά malformed μηνύματα αντί να στείλει ένα μοναδικό INVITE μήνυμα.

Το παρακάτω μήνυμα είναι κακοσχηματισμένο και δεν μπορεί να δημιουργηθεί από τους κανόνες του SIP πρωτοκόλλου. Ένα μήνυμα που δε συμβαδίζει με τις προδιαγραφές του SIP πρωτοκόλλου μπορεί να δημιουργήσει προβλήματα σε κάθε SIP σύστημα ωστόσο είναι δύσκολο να ξεχωρίσουμε αν ένα μήνυμα είναι νόμιμο ή όχι.

Κακοσχηματισμένο request μήνυμα:

```
aaaaaaaaa sip:1001@192.168.10.10 SIP/2.0
Via: SIP/2.0/UDP ccntd1.example.com:5060;branch=z9hG4bK00002000005
From::::::::: 2 <sip:user@cc-ntd1.example.com>;tag=2
To: Receiver <sip:1001@192.168.10.10>
Call-Id: 555555555555555555555-555555555555555555555555555-
55555555555555555-
5555555-555555555555555555555-5555555555555-55555555555555555-
555555555-
555555555@cc-ntd1.example.com
CSeq: 1 INVITE
Contact: 2 <sip:user@cc-ntd1.example.com>
Expires: 1200
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 143

v: = = = = = =0
o: 2 2 2 IN IP4 CAL-D600-5814.cc-ntd1.example.com
s: Session SDP
c: IN IP4 192.168.10.10
t: 0 0
m : audio
```

Στο παραπάνω μήνυμα βλέπουμε πολλά σημεία που θα μπορούσαν να προκαλέσουν προβλήματα στον parser. Άρχικα δεν υπάρχει σωστή μορφή του Request-URI καθώς δεν περιέχει καμία από τις request μεθόδους ενός SIP μηνύματος (INVITE, BYE, REGISTER, CANCEL, ACK, OPTIONS, SUBSCRIBE). Αναλύοντας και τις κεφαλίδες του μηνύματος βλέπουμε ότι και η From κεφαλίδα καθώς και η Call-Id έχουν λάθος μορφή. Στην κεφαλίδα From έχουμε υπερχείλιση του χαρακτήρα «:» ενώ η τιμή της κεφαλίδας Call-Id είναι πολύ μεγαλύτερη από την τιμή που θα έπρεπε να περιέχει. Στο συγκεκριμένο μήνυμα υπάρχει λάθος μορφή και στο SDP πρωτόκολλο που ακολουθεί στο σώμα του μηνύματος καθώς η τιμή της έκδοσης (version – v) δεν είναι σωστή.

Ένας server που θα παραλάμβανε αυτό το μη αναμενόμενο μήνυμα θα μπορούσε να «μπερδευτεί» (fuzzed) και να αντιδράσει με διάφορους τρόπους που έχουν να κάνουν και με την υλοποίηση του κάθε server.

Τυπικά, τα προβλήματα που θα μπορούσαν να δημιουργθούν είναι:

- Κατά τη διάρκεια του parsing να μπει σε άπειρο επαναλαμβανόμενο βρόχο (loop)
- Υπερχείλιση, που μπορεί να επιτρέψει την εκτέλεση αυθαίρετου κώδικα
- Αδυναμία να επεξεργαστεί άλλα νόμιμα μηνύματα
- Συντριβή του συστήματος

Όλα τα παραπάνω μπορούν να προκύψουν λόγω:

1. Των αδυναμιών στις προδιαγραφές του πρωτοκόλλου

   Τα περισσότερα VOIP πρωτόκολλα, όπως και το SIP που εξετάζουμε, είναι ανοικτά στο κοινό και δεν προδιαγράφουν αυστηρά κάθε γραμμή του μηνύματος. Με αυτόν τον τρόπο οι επιτιθέμενοι μπορούν να αναλύσουν το πρωτόκολλο και να βρουν τις συντακτικές του αδυναμίες. Επίσης υπάρχουν πολλά πεδία και ετικέτες που μπορούν να πάρουν προσαρμόσιμες τιμές.
2. Ευκολίας στη δημιουργία κακοσχηματισμένων μηνυμάτων

   Μηνύματα όπως το παραπάνω είναι πολύ εύκολο να δημιουργηθούν, όχι μόνο από συνηθισμένους προγραμματιστές αλλά και από λιγότερο σχετικούς χρήστες καθώς υπάρχουν πολλά εργαλεία που θα μπορούσαν να χρησιμοποιήσουν για τα δημιουργήσουν.
3. Έλλειψη χειρισμού εξαιρέσεων στην υλοποίηση

   Στις περισσότερες υλοποιήσεις των servers δεν δίνεται μεγάλη έμφαση σε χειρισμό μη αναμενόμενων περιπτώσεων.
4. Δυσκολία ελέγχου όλων των κακόβουλων περιπτώσεων

   Είναι πολύ δύσκολο να ελεγχθούν όλες οι περιπτώσεις κακοσχηματισμένων μηνυμάτων, ακόμα και αν πολλά εργαλεία πλεον έχουν βοηθήσει προς αυτήν την κατεύθυνση. Η απειλή αυτή των κακοσχηματισμένων μηνυμάτων μπορεί να αποτραπεί όταν ο parsing αλγόριθμος μπορεί να τα χειριστεί και να τα απορρίψει.

Όπως σε κάθε επίθεση δεν υπάρχει μια τυπική διαδικασία την οποία ακολουθεί ο επιτιθέμενος για να πετύχει το στόχο του. Έτσι και στην περίπτωση της επίθεσης στο SIP πρωτόκολλο η συμπεριφορά της μπορεί να είναι απρόβλεπτη. Για παράδειγμα ο επιτιθέμενος μπορεί να δοκιμάσει όλους τους δυνατούς συνδυασμούς για να δημιουργήσει ένα κακοσχηματισμένο μήνυμα. Εναλλακτικά, ο επιτιθέμενος μπορεί να ακολουθήσει μια πιο γενική διαδικασία για να επιτεθεί σε ένα SIP σύστημα. Τα παρακάτω βήματα μπορεί να αποτελέσουν τον αλγόριθμο της διαδικασίας αυτής:

1. Αναγνώριση των μειονεκτημάτων του SIP στόχου
2. Δημιουργία κακοσχηματισμένου μηνύματος
3. Αποστολή των μηνυμάτων αυτών στο SIP σύστημα

Το κύριο πλεονέκτημα αυτού του είδους επίθεσης είναι ότι δεν μπορεί να γίνει εύκολα αντιλληπτή στα πρώτα του βήματα καθώς σε αυτό το στάδιο οι μηχανισμοί άμυνας δεν είναι συνήθως ικανοί να τις αναγνωρίσουν.

9. INTRUSION DETECTION

9.1. Γενικοί κανόνες

Για να κατασκευάσουμε έναν μηχανισμό αντιμετώπισης των κακοσχηματισμένων χρειάζεται να μελετήσουμε τα χαρακτηριστικά των SIP μηνυμάτων. Όπως φάνηκε και στην παραπάνω ανάλυση μπορούμε να κατατάξουμε τις κεφαλίδες των SIP μηνυμάτων σε υποχρεωτικές, προαιρετικές και μη επιτρεπτές ανάλογα με τον τύπο του μηνύματος. Για κάθε τύπο μηνύματος (INVITE, ACK, BYE κλπ) η κατηγοριοποίηση αυτή είναι διαφορετική.

Ακολουθώντας όλα όσα αναφέρθηκαν παραπάνω καταλήγουμε σε κάποιους γενικούς κανόνες που πρέπει να ακολουθεί ένα SIP μήνυμα

- Πρέπει να έχει στην πρώτη γραμμή ένα SIP method (πχ, INVITE, BYE, REQUEST κλπ) μαζί με ένα ή περισσότερα SIP URI/s ακολουθούμενο από τις απαραίτητες κεφαλίδες.
- Προαιρετικά θα περιέχει ένα «σώμα» για το μήνυμα. Η παρουσία του ή όχι εξαρτάται από το SIP method.
- Κάθε SIP μήνυμα δεν μπορεί να έχει κενό SIP method ή κεφαλίδες.
- Το μήκος του SIP method και του body του μηνύματος δεν μπορεί να είναι μεγαλύτερο από ένα συγκεκριμένο όριο.

Κεφαλίδες για SIP INVITE requests:

Υποχρεωτικά	**Προαιρετικά**
From	Via
To	Max-Forwards
Call-Id	User-agent
CSeq	Authorization
Contact	Event
Content-Length	Record-Route
Content-type	

Κεφαλίδες για SIP REGISTER requests:

Υποχρεωτικά	**Προαιρετικά**
From	Via
To	Max-Forwards
Call-Id	User-agent
CSeq	Authorization
Contact	Event
Content-Length	

Κεφαλίδες για SIP BYE/CANCEL/ACK requests:

Υποχρεωτικά	**Προαιρετικά**
Max-Forwards	Via
From	User-agent
To	Authorization
Call-Id	
CSeq	
Content-Length	

Κεφαλίδες για SIP SUBSCRIBE requests:

Υποχρεωτικά	**Προαιρετικά**
Max-Forwards	Via
From	Contact
To	Record-Route
Call-Id	Expires
CSeq	User-agent
Content-Length	Authorization
	Content-type

Το μοντέλο που πρέπει να δημιουργήσουμε για τον εντοπισμό των κακοσχηματισμένων μηνυμάτων πρέπει να ακολουθεί τους κανόνες που προδιαγράφονται στο RFC3261. Αν το μήνυμα δεν είναι σύμφωνο με αυτές τις προδιαγραφές πρέπει να χαρακτηριστεί ως κακόβουλο και να απορριφτεί.

Υπάρχουν πάνω από 280 κανόνες στο RFC3261 που μπορούν να προσδιορίσουν ποια είναι η σωστή μορφή που πρέπει να έχει κάθε κεφαλίδα.

Ωστόσο στην πράξη διαπιστώνουμε ότι μόνο η συμμόρφωση με τις προδιαγραφες του RFC3261 δεν αρκεί για να εντοπιστούν όλα τα κακόβουλα μηνύματα. Για παράδειγμα σύμφωνα με τους κανόνες του RFC το userinfo πρέπει να έχει τη μορφή

userinfo: ((#user#)(: #password#)?)

port=1 *DIGIT

Πρακτικά όμως διαπιστώνουμε ότι αν ένας επιτιθέμενος δώσει στο userinfo μια πολύ μεγάλη τιμή σε μήκος θα προκαλέσει υπερχείλιση μνήμης. Το ίδιο ισχύει και με το port όπου πρέπει να είναι ένας μόνο ακέραιος, ωστόσο οι προδιαγραφές του RFC δεν το προβλέπουν αυτό. Για να αποφύγουμε τέτοιου είδους επιθέσεις πλημμύρας, πρέπει να επεκτείνουμε το μηχανισμό του parser ώστε να προλαμβάνει και επιθέσεις που έχουν να κάνουν με το μήκος των δεδομένων πέρα από την καθαρά μορφοποίησή τους.

Μια άλλη παράμετρος που πρέπει να λάβουμε υπόψην είναι τα δεδομένα τα οποία περιέχονται στο μήνυμα. Ένας επιτιθέμενος όπως θα δούμε και στο παρακάτω μήνυμα μπορεί να προκαλέσει SQL Injection αν δεν ελέγξουμε τα δεδομένα τα οποία περνώνται στις κεφαλίδες του μηνύματος. Ο μεγαλύτερος κίνδυνος βρίσκεται στην κεφαλίδα “Authentication” όπου ο επιτιθέμενος μπορεί να περάσει κατάλληλες παραμέτρους ώστε να παρακάμψει το βήμα της αυθεντικοποίησής του και να αποκτήσει πρόσβαση στο SIP σύστημα.

9.2. Ανάλυση των κεφαλίδων και της μορφής τους

9.2.1. Κεφαλίδα From

Η κεφαλίδα From είναι υποχρεωτική σε όλα τα μηνύματα SIP και χρησιμοποιείται για να προσδιορίσει τον αποστολέα του αιτήματος. Στις περιπτώσεις απόκρισης το μήνυμα από τον καλούμενο στον καλούντα περιέχει τη διεύθυνση του καλούμενου στην κεφαλίδα From. Περιέχει ένα προβαλλόμενο όνομα (display name), ένα URI, καθώς και κάποιες παραμέτρους (tag). Οι παράμετροι και το προβαλλόμενο όνομα είναι προαιρετικά. Όταν κάποιο από αυτά εμφανίζεται το SIP URI πρέπει υποχρεωτικά να εσωκλείεται σε < >. Επίσης αν ένα σύστημα θέλει να παραμείνει κρυφή η ταυτότητα του πελάτη πρέπει να χρησιμοποιήσει το προβαλλόμενο όνομα “Anonymous”.

Δυο πεδία κεφαλίδας From μπορούν να θεωρηθούν ισοδύναμα αν τα URIs και οι παράμετροι κεφαλίδας ταιριάζουν. Οι επιπρόσθετες παράμετροι που δεν λαμβάνονται υπόψην στη σύγκριση οπότε και η παρουσία ή απουσία των < > δεν παίζει ρόλο σε αυτήν.

Η εναλλακτική μορφή της κεφαλίδας είναι f. Παρακάτω φαίνονται μερικά παραδείγματα:

From: "A. G. Bell" <sip:agb@bell-telephone.com> ;tag=a48s

From: sip:+12125551212@server.phone2net.com;tag=887s

f: Anonymous <sip:c8oqz84zk7z@privacy.org>;tag=hyh8

9.2.2. Κεφαλίδα To

Η κεφαλίδα To είναι υποχρεωτική κεφαλίδα για κάθε μήνυμα SIP και χρησιμοποιείται για να προσδιορίσει τη διεύθυνση του παραλήπτη του αιτήματος. Οι αποκρίσεις που στέλνονται από τον UA περιέχουν την κεφαλίδα αυτή μαζί με ένα tag. Ο κάθε proxy πρέπει να προσθέτει ένα tag στην κεφαλίδα To σε κάθε απόκριση που παράγει. Η μορφή της κεφαλίδας To είναι:

To: user <sip:user@unipi.gr>;tag=1234

Επιτρέπει όπως φαίνεται την εμφάνιση ενός προβαλλόμενου ονόματος (display name) πριν από το URI το οποίο όμως είναι προαιρετικό. Το SIP URI εσωκλείεται σε υποχρεωτικά σε < > όταν υπάρχει προβαλλόμενο όνομα ή όταν περιέχει κάποιες

παραμέτρους ή περιέχονται παράμετροι που αφορούν αναγνωριστικά.Η εναλλακτική μορφή της κεφαλίδας είναι t.

9.2.3. Κεφαλίδα Call-ID

Η κεφαλίδα Call-ID είναι υποχρεωτική σε όλα τα αιτήματα SIP και όλες τις αποκρίσεις. Είναι μέρος του διαλόγου και χρησιμοποιείται για να προσδιορίσει μοναδικά μια κλήση μεταξύ δύο UAs. Το Call-ID πρέπει να είναι μοναδικό για όλες τις κλήσεις, εκτός από την περίπτωση του Call-ID που υπάρχει στα αιτήματα εγγραφής. Όλα τα αιτήματα REGISTER που προέρχονται από έναν UA πρέπει να χρησιμοποιούν το ίδιο Call-ID. Το Call-ID δημιουργείται πάντοτε από τον UA και δεν τροποποιείται ποτέ από τον εξυπηρετητή.

Το Call-ID αποτελείται συνήθως από ένα τοπικό ID, θα πρέπει να είναι ένα κρυπτογραφημένο αναγνωριστικό, ή ένα τοπικό ID, το σύμβολο @ και ένα host name ή IP διεύθυνση. Η εναλλακτική μορφή της κεφαλίδας Call-ID είναι i. Παρακάτω παρουσιάζονται παραδείγματα κεφαλίδων Call-ID:

Call-ID: 34a5d553192cc35@15.34.3.1

Call-ID: 12-45-A5-F5@atlanta.com

9.2.4. Κεφαλίδα CSeq

Η κεφαλίδα CSeq (command sequence) απαιτείται σε κάθε αίτημα. Η κεφαλίδα CSeq περιέχει έναν ακέραιο αριθμό που αυξάνεται για κάθε αίτημα. Συνήθως αυξάνεται κατά μία μονάδα για κάθε νέο αίτημα, με εξαίρεση τα αιτήματα CANCEL και ACK, τα οποία χρησιμοποιούν τον ίδιο αριθμό CSeq με το αίτημα INVITE στο οποίο αναφέρεται.

Ο μετρητής CSeq χρησιμοποιείται από τους UAS για να καθορίσουν αιτήματα που έρχονται εκτός σειράς ή για να διακρίνουν ένα νέο αίτημα (διαφορετικό CSeq) από μια επανεκπομπή (ίδιο CSeq). Η κεφαλίδα CSeq χρησιμοποιείται από τους UACs για να αντιστοιχίσουν την απόκριση στο αίτημα στο οποίο αναφέρεται. Για παράδειγμα, ένας UAC που στέλνει ένα αίτημα INVITE και ακολούθως ένα αίτημα

CANCEL μπορεί να ξεχωρίσει σε ποιο αίτημα αναφέρεται μια απόκριση 200 OK που λαμβάνει, διαβάζοντας το περιεχόμενο της κεφαλίδας CSeq.

9.2.5. Κεφαλίδα Via

Η κεφαλίδα Via είναι και αυτή υποχρεωτική και χρησιμοποιείται για να καταγράψει τη διαδρομή του αιτήματος και για να δρομολογήσει την απάντηση πίσω στον δημιουργό του αιτήματος. Κάθε UA που παράγει ένα αίτημα καταγράφει τη δική του διεύθυνση στην κεφαλίδα Via του αιτήματος. Ενώ η σειρά των περισσότερων κεφαλίδων SIP δεν είναι σημαντική, η σειρά των κεφαλίδων Via είναι σημαντική γιατί χρησιμοποιείται για τη δρομολόγηση των αποκρίσεων.

Κάθε proxy που προωθεί ένα αίτημα προσθέτει μια κεφαλίδα Via που περιέχει τη δική του διεύθυνση (προαιρετικά μαζί με το port) στην κορυφή της λίστας με τις κεφαλίδες Via. Επίσης συμπεριλαμβάνει μια ετικέτα branch, η οποία περιέχει ένα κρυπτογραφημένο hash των κεφαλίδων To, From, Call-ID και του URI που υπάρχει στη γραμμή αιτήματος. Όταν ένας proxy ή ένας UA παράγει μια απόκριση, αντιγράφει όλες τις κεφαλίδες από το αίτημα στην απόκριση με την ίδια σειρά και μετά στέλνει αυτή την απόκριση στη διεύθυνση που ορίζεται στην πρώτη κεφαλίδα Via. Όταν ένας proxy λαμβάνει μια απόκριση ελέγχει την πρώτη κεφαλίδα Via για να διασφαλίσει ότι ταιριάζει με τη δική του διεύθυνση. Αν δεν συμβαίνει αυτό η απόκριση έχει δρομολογηθεί λανθασμένα και πρέπει να απορριφθεί. Στην αντίθετη περίπτωση η πρώτη κεφαλίδα Via αφαιρείται και η απόκριση προωθείται στη διεύθυνση που προσδιορίζεται από την επόμενη κεφαλίδα Via.

Οι κεφαλίδες Via πρέπει να περιέχουν το όνομα και την έκδοση του πρωτοκόλλου, το πρωτόκολλο μεταφοράς (SIP/2.0/UDP, SIP/2.0/TCP, κτλ) και μπορεί να περιέχουν τον αριθμό της πόρτας και παραμέτρους όπως οι: received, branch, maddr, ttl. Η branch παράμετρος είναι υποχρεωτική και χρησιμοποιείται τόσο από τον client όσο και από τον server ως αναγνωριστικό της συναλλαγής γι'αυτό και πρέπει να είναι μοναδική για όλα τα αιτήματα που στέλνονται από τον UA. Εξαίρεση αποτελούν οι αιτήσεις CANCEL καθώς και οι ACK για αποκρίσεις εκτός των 2xx, που περιέχουν την ίδια τιμή branch με την αίτηση στην οποία αναφέρονται. Κάθε branch παράμετρος, σύμφωνα με το RFC3261, πρέπει να ξεκινάει με την τιμή "z9hG4bK" για να εξασφαλίσει ότι τηρεί αυτές τις νεώτερες προδιαγραφές. Μια

παράμετρος received προστίθεται σε μια κεφαλίδα Via αν ένας UA ή proxy λάβει το αίτημα από μια διεύθυνση που είναι διαφορετική από αυτή που υπάρχει στην πρώτη κεφαλίδα Via. Αυτό δείχνει ότι υπάρχει ένα NAT ή ένα ανάχωμα ασφαλείας σε κάποιον proxy που βρίσκεται στη διαδρομή του μηνύματος. Αν υπάρχει μια ετικέτα received, τότε αυτή χρησιμοποιείται στη δρομολόγηση της απόκρισης. Η εναλλακτική μορφή της κεφαλίδας είναι v.

9.2.6. Κεφαλίδα Contact

Η κεφαλίδα Contact χρησιμοποιείται για να μεταφέρει ένα URI που προσδιορίζει τον παραλήπτη ή τον δημιουργό του αιτήματος, ανάλογα με το αν είναι παρούσα σε ένα αίτημα ή σε μια απόκριση. Μόλις ληφθεί μια κεφαλίδα Contact, το URI που περιέχει μπορεί να αποθηκευτεί και να χρησιμοποιηθεί για τη δρομολόγηση των μελλοντικών αιτημάτων του διαλόγου. Για παράδειγμα, μια κεφαλίδα που υπάρχει σε μια απόκριση 200 OK για ένα αίτημα INVITE μπορεί να επιτρέψει στο μήνυμα επιβεβαίωσης ACK και σε όλα τα μελλοντικά αιτήματα κατά τη διάρκεια αυτής της κλήσης να παρακάμψουν τους proxy και να αποσταλούν απευθείας στο καλούμενο μέλος. Παρόλα αυτά η παρουσία μιας κεφαλίδας Record-Route σε προηγούμενα αιτήματα ή μια προεπιλεγμένη δρομολόγηση μέσω proxy κάποιου UA μπορεί να αγνοήσει αυτή τη συμπεριφορά. Όταν χρησιμοποιείται ένα Contact URI στη γραμμή αιτήματος, όλες οι παράμετροι URI επιτρέπονται με εξαίρεση την παράμετρο method, η οποία αγνοείται. Η κεφαλίδα Contact είναι υποχρεωτικό να είναι παρούσα στα αιτήματα INVITE και στις αποκρίσεις 200 OK σε προσκλήσεις. Το URI που περιέχει θα εσωκλείεται σε < > αν υπάρχει κάποιο προβαλλόμενο όνομα, το οποίο μπορεί να βρίσκεται μέσα σε εισαγωγικά, ή αν υπάρχουν παράμετροι.

Σύμφωνα με το RFC3261 χρησιμοποιούνται δυο επιπρόσθετες παράμετροι που έχουν οριστεί για χρήση στην κεφαλίδα Contact: q και expires. Αυτές οι παράμετροι εμφανίζονται σε Register αιτήματα ή αποκρίσεις και σε 3xx αποκρίσεις. Τοποθετούνται στο τέλος του URI και διαχωρίζονται με ελληνικά ερωτηματικά. Η τιμή της παραμέτρου q χρησιμοποιείται για να δείξει τη σχετική προτίμηση, η οποία παρουσιάζεται με ένα δεκαδικό αριθμό που κυμαίνεται από το 0 μέχρι το 1. Η παράμετρος expires προσδιορίζει για πόσο χρονικό διάστημα είναι έγκυρο το URI. Παρακάτω φαίνεται η μορφή αυτής της κεφαλίδας όταν περιέχει δύο SIP URIs:

Contact: "Mr. Watson" <sip:watson@worcester.bell-telephone.com>
;q=0.7; expires=3600,
"Mr. Watson" <mailto:watson@bell-telephone.com> ;q=0.1

9.2.7. Κεφαλίδα Max-Forwards

Η κεφαλίδα Max-Forwards χρησιμοποιείται για να προσδιορίσει τον μέγιστο αριθμό κόμβων από τους οποίους μπορεί να περάσει ένα αίτημα SIP. Η τιμή της κεφαλίδας μειώνεται κατά μία μονάδα για κάθε proxy που προωθεί το αίτημα. Όταν ο proxy λάβει ένα μήνυμα που η τιμή του είναι μηδέν, το απορρίπτει και αποστέλλει μια απόκριση 483 Too Many Hops πίσω στο δημιουργό του αιτήματος.

Η κεφαλίδα Max-Forwards είναι υποχρεωτική σύμφωνα με το RFC 3261. Η προτεινόμενη αρχική τιμή της είναι 70 κόμβοι. Ένα απλό παράδειγμα είναι:

Max-Forwards: 10

10. PROTOS SUITE

Το PROTOS είναι ένα εργαλείο fuzzer, κατασκευασμένο για SIP που έχει ως στόχο να εντοπίσει τις αδυναμίες του SIP Parser. Fuzzers ονομάζουμε τα εργαλεία που προσπαθούν να εκμεταλλευτούν τις αδυναμίες του συστήματος και να προκαλέσουν ζημιά με τη δημιουργία ασυνήθιστης εισόδου όπως μπορεί να θεωρηθεί η είσοδος μηνυμάτων με πολύ μεγάλο μήκος ή με ασυνήθιστο συνδυασμό χαρακτήρων, ώστε στη συνέχεια να εξεταστεί η συμπεριφορά του συστήματος στόχου όταν έρχεται αντιμέτωπο με αυτές τις περιπτώσεις.

Στη συγκεκριμένη μελέτη χρησιμοποιείται αυτό το εργαλείο ώστε να εξετάσουμε τη συμπεριφορά της υλοποίησής μας σε πολλές περιπτώσεις κακοσχηματισμένων και ασυνήθιστων μηνυμάτων.

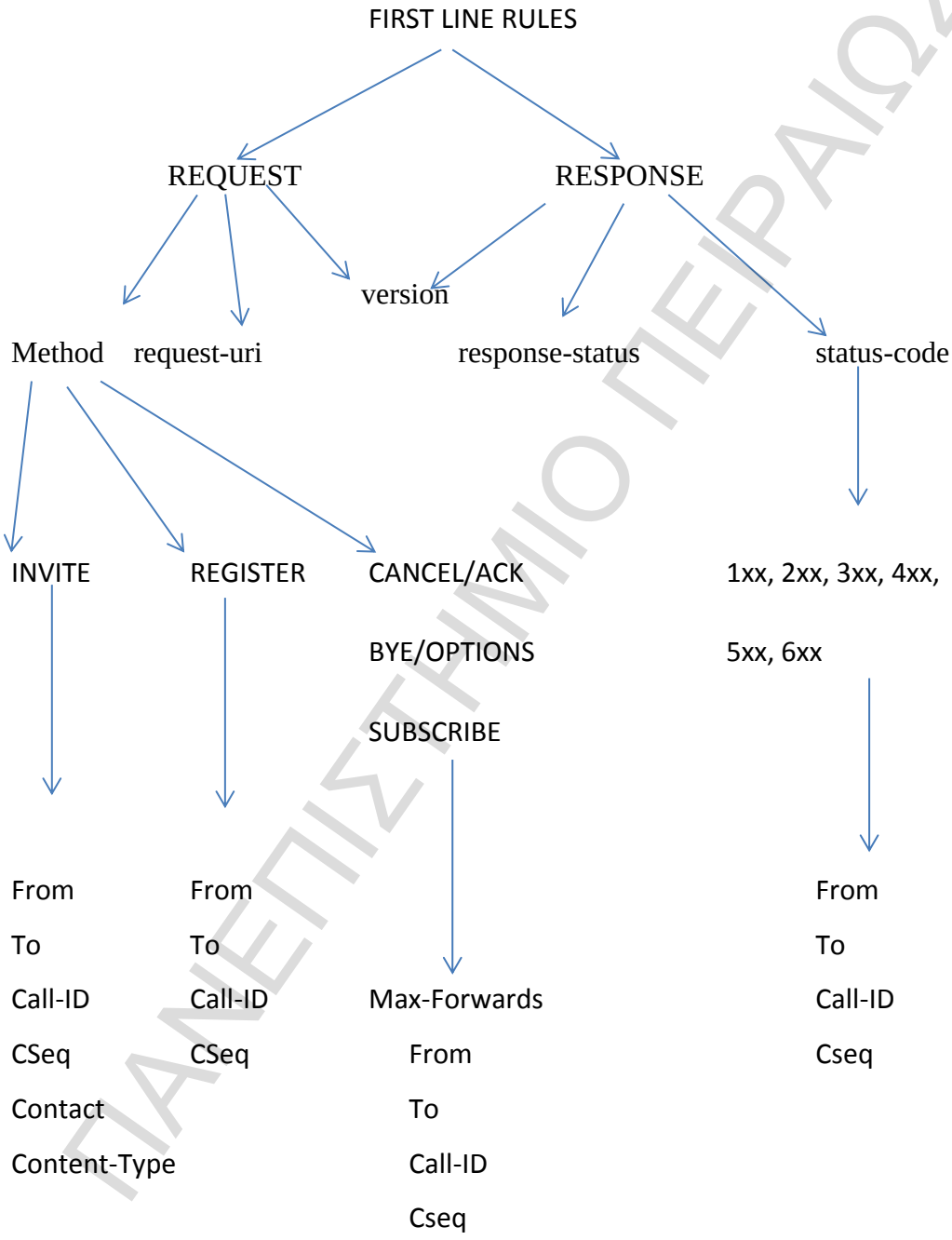
Το PROTOS διαθέτει 4527 test cases που έχουν σχεδιαστεί για να εντοπιστούν λάθη στο SIP λογισμικό όταν οι παράμετροι δε συμβαδίζουν με το SIP πρωτόκολλο.

Τα είδη των λαθών που το PROTOS είναι ικανό να εντοπίσει έχουν να κάνουν αυστηρά με το μηχανισμό parsing που εντοπίζει malformed input. Το PROTOS διαθέτει μηνύματα με τις ακόλουθες περιπτώσεις εξαιρέσεων:

- Κενά στις τιμές των στοιχείων
- Εξαιρέσεις σε IPv4 διευθύνσεις
- Υπερχείλιση από χαρακτήρες 'a' (0x61) έως 128k
- Υπερχείλιση από χαρακτήρες '/' έως 128k
- Υπερχείλιση από χαρακτήρες ':' έως 128k
- Υπερχείλιση από χαρακτήρες ' ' έως 128k
- Υπερχείλιση από χαρακτήρες 'a' (0x61) και NULLs ενδιάμεσα
- Υπερχείλιση από χαρακτήρες '<' και '>' έως 128k
- Υπερχείλιση από χαρακτήρες '@' έως 128k
- Υπερχείλιση από χαρακτήρες '=' έως 128k
- Επαναλαμβανόμενες ακολουθίες (πχ. %s%s%s)
- Μη έγκυρη κωδικοποίηση UTF-8
- Αρνητικές τιμές ακεραίων
- Escape χαρακτήρες
- Λάθος μορφή στο Uri
- Λάθος τιμή σε version και content-type
- Λάθος τιμή στα tags
- Ανακολουθία CRLF

11. ΒΗΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ ΑΛΓΟΡΙΘΜΟΥ

Η υλοποίηση του μοντέλου που παρουσιάζεται είναι σε Java. Το γενικό διάγραμμα το οποίο ακολουθήθηκε είναι το παρακάτω:



Αρχικά το μήνυμα που διαβάζεται κατηγοριοποιείται σε μήνυμα Request ή Response και στη συνέχεια ελέγχεται η γραμμή εκκίνησης. Ανάλογα με τη μέθοδο, στην

περίπτωση του Request ελέγχονται αν υπάρχουν οι απαραίτητες κεφαλίδες και αν έχουμε επανάληψη κάποιων που δε θα έπρεπε. Τέλος, ελέγχονται οι τιμές των πεδίων.

Στην υλοποίηση που ακολουθήσαμε προσπαθήσαμε να προβλέψουμε όλες τις δυνατές περιπτώσεις κακοσχηματισμένων μηνυμάτων που θα μπορούσαν να προκαλέσουν πρόβλημα στον parser. Πιο αναλυτικά οι περιπτώσεις τις οποίες εξετάζουμε είναι :

- Μέγεθος μηνύματος: τα πολύ μεγάλα μηνύματα απορρίπτονται
- Κωδικοποίηση: Αν οι τιμές των headers δεν περιέχουν UTF-8 κωδικοποίηση μπορεί να προκαλέσουν προβλήματα στην κατανόησή τους από τον parser
- Έλεγχο για επαναλαμβανόμενες αλληλουχίες
- Overflow από ίδιους χαρακτήρες ή σύμβολα
- Έλεγχο συγκεκριμένων τιμών, για παράδειγμα η τιμή της κεφαλίδας Content-Length πρέπει να είναι ίση με το μέγεθος του «body» του μηνύματος
- Έλεγχο για επανάληψη των υποχρεωτικών κεφαλίδων όπως είναι οι From, To, Max-Forwards, Call-ID ή CSeq.
- Έλεγχο για ύπαρξη χαρακτήρων που θα μπορούσαν να δημιουργήσουν πρόβλημα κατά τις προσπέλαση του μηνύματος όπως είναι οι «%», «.», «/» , «\», «”»
- Έλεγχο για ύπαρξη SQL statement στην κεφαλίδα Authenticate που σκοπό έχει να προκαλέσει SQL Injection
- Ύπαρξη των απαραίτητων κεφαλίδων ανάλογα με τη request method
- Έλεγχο για δομή και την τιμή της κεφαλίδας αν είναι σύμφωνη με τις προδιαγραφές του RFC3261, με τη χρήση regular expressions
- Έλεγχο του «body» του μηνύματος για την σωστή τιμή των fields που περιέχει ανάλογα με τον αν αναφέρεται σε version (v), media (m), owner (o), session (s), timer (t), c ή a

Αξιοποιήθηκαν κυρίως τα μηνύματα του PROTOS test suite για να αξιολογηθεί η εγκυρότητα των αποτελεσμάτων της υλοποίησης μας, ωστόσο επεκτείναμε το μοντέλο ώστε να γίνεται έλεγχος και για SQL Injection, για μηνύματα απόκρισης (response) και για μηνύματα αιτήσεων (request), πέρα από τα INVITE requests που περιλαμβάνει το PROTOS.

Παρακάτω φαίνονται τρια test-cases που θα μπορούσαν να προκαλέσουν προβλήματα στον parser.

12. TEST CASES

1. **Κακοσχηματισμένη Γραμμή εκκίνησης:**

Ας υποθέσουμε ότι ο parser λαμβάνει το ακόλουθο Request μήνυμα:

```
aaaaaaaaa sip:<To> SIP/2.0
Via: SIP/2.0/UDP <From-Address>:<Local-
Port>;branch=z9hG4bK00002<Branch-ID>
From: 2 <sip:<From>>;tag=2
To: Receiver <sip:<To>>
Call-ID: <Call-ID>@<From-Address>
CSeq: <CSeq> INVITE
Contact: 2 <sip:<From>>
Expires: 1200
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: <Content-Length>

v=0
o=2 2 2 IN IP4 <From-Address>
s=Session SDP
c=IN IP4 <From-IP>
t=0 0
m=audio 9876 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Το μήνυμα αυτό δε διαθέτει σωστή request μέθοδο. Το μοντέλο που υλοποιείται στη συγκεκριμένη εργασία διαβάζει το μήνυμα και από την γραμμή εκκίνησης του μηνύματος το κατατάσει σε request ή response ανάλογα με τη μορφή της.

Στη συνέχεια εξετάζει καθένα από τα στοιχεία που περιέχει η γραμμή εκκίνησης αν είναι σύμφωνα με τις προδιαγραφές.

Στη συγκεκριμένη περίπτωση το σφάλμα εντοπίζεται στη μέθοδο **processFirstLine()** . Πιο συγκεκριμένα στη μέθοδο αυτή για την περίπτωση request μηνύματος ελέγχεται αν τηρείται η μορφή:

Method_sip:<To>_version

όπου

- method είναι μια από τις γνωστές request μεθόδους που αναλύσαμε παραπάνω,

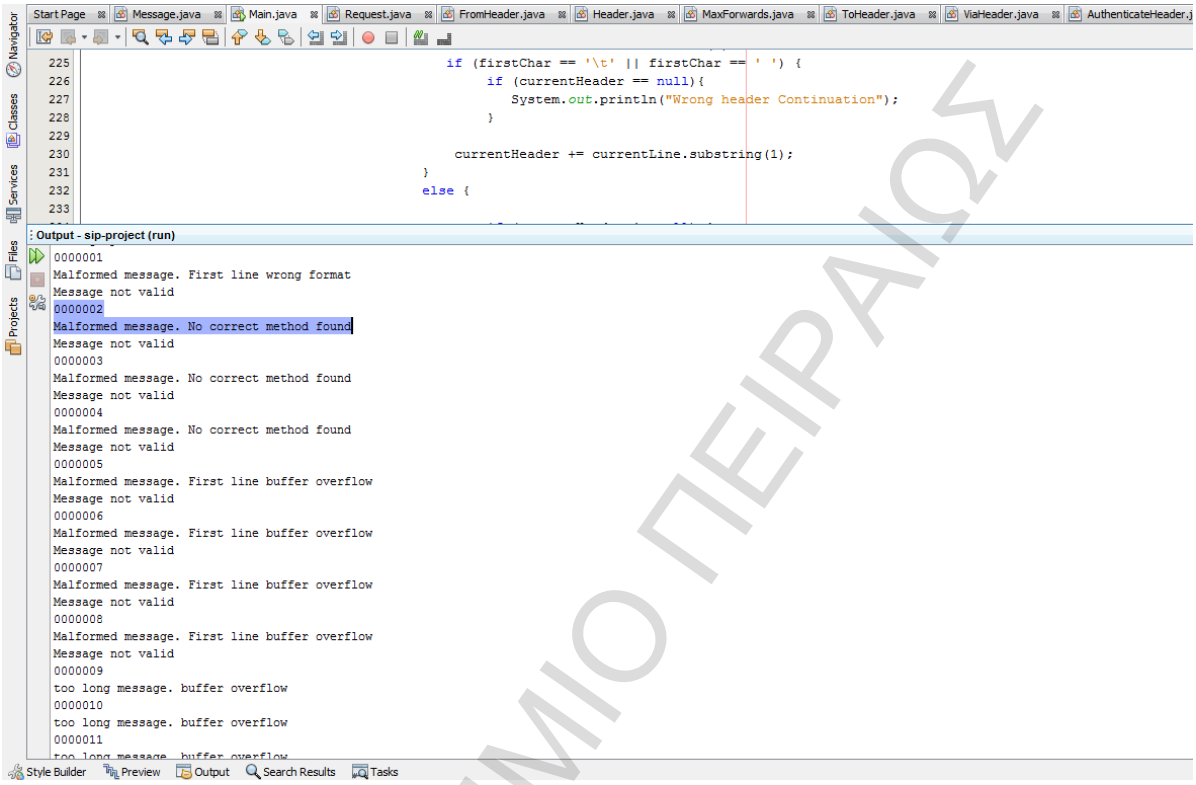
- <To> αντιπροσωπεύει το request-uri και πρέπει να είναι της μορφής username:password@host:port (χωρίς το host και το port να είναι υποχρεωτικά) και
- Version είναι η έκδοση sip που χρησιμοποιείται (SIP/2.0)

Παράλληλα γίνεται έλεγχος για το encoding των τιμών, αν είναι κενές, αν περιέχουν invalid characters και αν ανταποκρίνονται στην τιμή ή τη μορφή που θα έπρεπε να έχουν.

Παρακάτω φαίνεται το αντίστοιχο κομμάτι κώδικα:

```
private static Message processFirstLine(String firstLine)  {

        if (!firstLine.startsWith("SIP/2.0")) { //Case Request
                Request  message = new Request();
                try {
                    String[] requestLine = firstLine.split("\\s+"); //split to get method, request-uri, version

                         if(requestLine.length == 4){
                        if((requestLine[1]).length()<35){
                             byte[] myBytes = null;
                         myBytes = requestLine[1].getBytes();

                         if(!isValidEncoding(requestLine[1])){
                             System.out.println("Not valid encoding");
                             return null;
                         }
                    if(!(requestLine[1]).equals("")){
                        message.setMethod(requestLine[1]);


                          if (message.getMethod().equals(Request.INVITE)|| message.getMethod().equals(Request.SUBSCRIBE)
                          || message.getMethod().equals(Request.REGISTER)|| message.getMethod().equals(Request.ACK)
                          || message.getMethod().equals(Request.BYE) || message.getMethod().equals(Request.REFER)
                          || message.getMethod().equals(Request.OPTIONS)|| message.getMethod().equals(Request.CANCEL) ) {
                               if(requestLine[2].length() >40){
                                   System.out.println("Sip Uri Overflow");
                                  return null;
                               }
                              if(requestLine[2].equals("")){
                                  System.out.println("Empty Sip Uri");
                                  return null;
                              }
                              String re1="(sip)|(sips)";      // Variable Name
                              String re2="(:)";       // Any Single Character
                              String re3="(\\s+)*";   // White Space
                              String re4="((?:[a-z][a-z]*[0-9]+[a-z0-9]*))";  // Alphanum
                              String re5="(@)";       // Any Single Character
                String re6="((?:[a-z][a-z\\.\\d\\-]+)\\.(?:[a-z][a-z\\-]+))(?![\\w\\.])";// Fully Qualified Domain Name 1
            String re7="|sip:<To>";
                Pattern p2 = Pattern.compile(re1+re2+re3+re4+re5+re6+re7,Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
                Matcher m3 = p2.matcher(requestLine[2]);
                if (m3.find())
                {
                   message.setRequestLine(requestLine[2]);
                }else{
                    System.out.println("Wrong form Sip Uri");
                    return null;
                }
                //Check version
                if (requestLine[3].equals("SIP/2.0"))
                {

                }else{
                    System.out.println("Wrong form Sip version");
                    return null;
                }
        }else{
             System.out.println("Malformed message. No correct method found");
            return null;
        }
        }else{
            System.out.println("Malformed message. No method found");
            return null;
        }
        return message;
    }else{
        System.out.println("Malformed message. First line buffer overflow");
        return null;
    }
   }else{
        System.out.println("Malformed message. First line wrong format");
        return null;
    }
```

Το μήνυμα που φαίνεται παραπάνω εντοπίζεται από τον μηχανισμό και χαρακτηρίζεται ως «`Malformed message. No correct method found`»



2. Κακοσχηματισμένη Via κεφαλίδα

Στην περίπτωση αυτή εξετάζουμε το παρακάτω μήνυμα:

```
INVITE sip:<To> SIP/2.0
Via: SIP/2.0/UDP
%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%
s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s
%s%s:<Local-Port>;branch=z9hG4bK0000357<Branch-ID>
From: 357 <sip:<From>>;tag=357
To: Receiver <sip:<To>>
Call-ID: <Call-ID>@<From-Address>
CSeq: <CSeq> INVITE
Contact: 357 <sip:<From>>
Expires: 1200
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: <Content-Length>

v=0
o=357 357 357 IN IP4 <From-Address>
s=Session SDP
c=IN IP4 <From-IP>
t=0 0
m=audio 9876 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Το μήνυμα αυτό περιέχει μη επιτρεπτή τιμή στην κεφαλίδα Via. Υπάρχει επανάληψη μη επιτρεπτής ακολουθίας που μπορεί να οδηγήσει το σύστημα σε buffer overflow.

Στο μοντέλο που υλοποιούμε όταν εξετάζεται την κεφαλίδα Via ελέγχεται να τηρείται η μορφή

Via: SIP/2.0/UDP <From-Address>:<Local-Port>;branch=z9hG4bK{id}<Branch-ID>

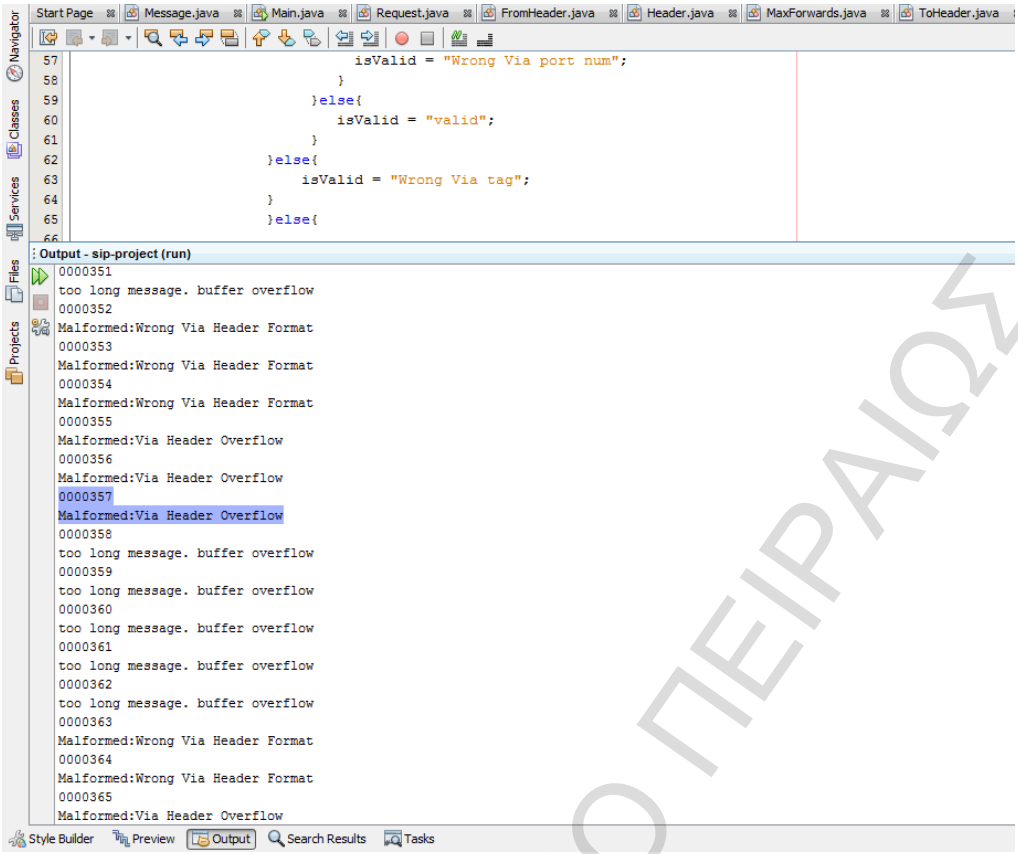
Με τις αντίστοιχες επιτρεπτές τιμές που μπορούν να πάρουν οι <From-Address>, <Local-Port>, {id} και <Branch-ID>

Στη θέση του branch θα μπορούσαμε επίσης να έχουμε ttl ή maddr ή received.

Παρακάτω φαίνεται το κομμάτι του κώδικα που ελέγχει την κεφαλίδα Via. Ο έλεγχος εδώ σταματάει στο σημείο οπου η κεφαλίδα είναι μεγαλύτερη από κάποιο επιτρεπτή σταθερή τιμή (constant) αλλά ακόμα και αν περνούσε τον έλεγχο αυτόν θα εντοπιζόταν παρακάτω ως μη επιτρεπτή μορφή της Via κεφαλίδας:

```
public String checkViaHeader(String header){

     String[] headerLine = header.split("\\s+");
     String isValid= "";

     if(headerLine[1].equals("SIP/2.0/UDP")){
         if( headerLine[2].indexOf( ';' ) >= 0 ) {


         String[] ipv4_host = headerLine[2].split(";");
          if(ipv4_host[0].length()< 40){

              String[] port  = ipv4_host[0].split(":");
              Pattern p2 = Pattern.compile(
                     "^([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
                       "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
                       "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +
                       "([01]?\\d\\d?|2[0-4]\\d|25[0-5])$(:[\\d]{1,5})"+
                    "|((?:[a-z][a-z\\.\\d\\-]+)\\.(?:[a-z][a-z\\-]+))(?![\\w\\.])(:[\\d]{1,5})" +
                    "|<From-Address>(:<Local-Port>)|" +
                    "<From-Address>(:[\\d]{1,5})",
                    Pattern.CASE_INSENSITIVE | Pattern.DOTALL);

               Matcher m3 = p2.matcher(ipv4_host[0]);

               if (m3.find()){


                 Pattern via_tag = Pattern.compile(
                                    "(branch=z9hG4bK.*?)|maddr=.*?|ttl=(\\d+)|received.*?" +
                                    "|branch=z9hG4bK.*?<Branch-ID>" ,
                                    Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
                 Matcher tag = via_tag.matcher(ipv4_host[1]);
                 if (tag.find()){
                     if(!port[1].equals("<Local-Port>")  ){
                         if(port[1].length() >5){
                            isValid = "Long Via port num";
                         }else if(Integer.parseInt(port[1])<1024 || Integer.parseInt(port[1]) > 49151){
                          isValid = "Wrong Via port num";
                        }
                     }else{
                        isValid = "valid";
                     }
                 }else{
                     isValid = "Wrong Via tag";
                 }
                 }else{

                    isValid = "Wrong Via Header Format";

                 }

          }else
          {
               isValid = "Via Header Overflow";
          }
        }
     }else{
        isValid = "Wrong Via Sip Version";
     }
    return isValid;
}
```

Ο έλεγχος εδώ μας εμφανίζει το ακόλουθο μήνυμα: «`Malformed:Via Header Overflow`»



3. Λάθος μορφή του Request URI στην κεφαλίδα From

Στην περίπτωση αυτή εξετάζουμε το παρακάτω μήνυμα:

```
INVITE sip:<To> SIP/2.0
Via: SIP/2.0/UDP <From-Address>:<Local-
Port>;branch=z9hG4bK0000897<Branch-ID>
From: 897 <aaaaa:noone@sip.no.invalid>;tag=897
To: Receiver <sip:<To>>
Call-ID: <Call-ID>@<From-Address>
CSeq: <CSeq> INVITE
Contact: 897 <sip:<From>>
Expires: 1200
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: <Content-Length>

v=0
o=897 897 897 IN IP4 <From-Address>
s=Session SDP
```

```
c=IN IP4 <From-IP>
t=0 0
m=audio 9876 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

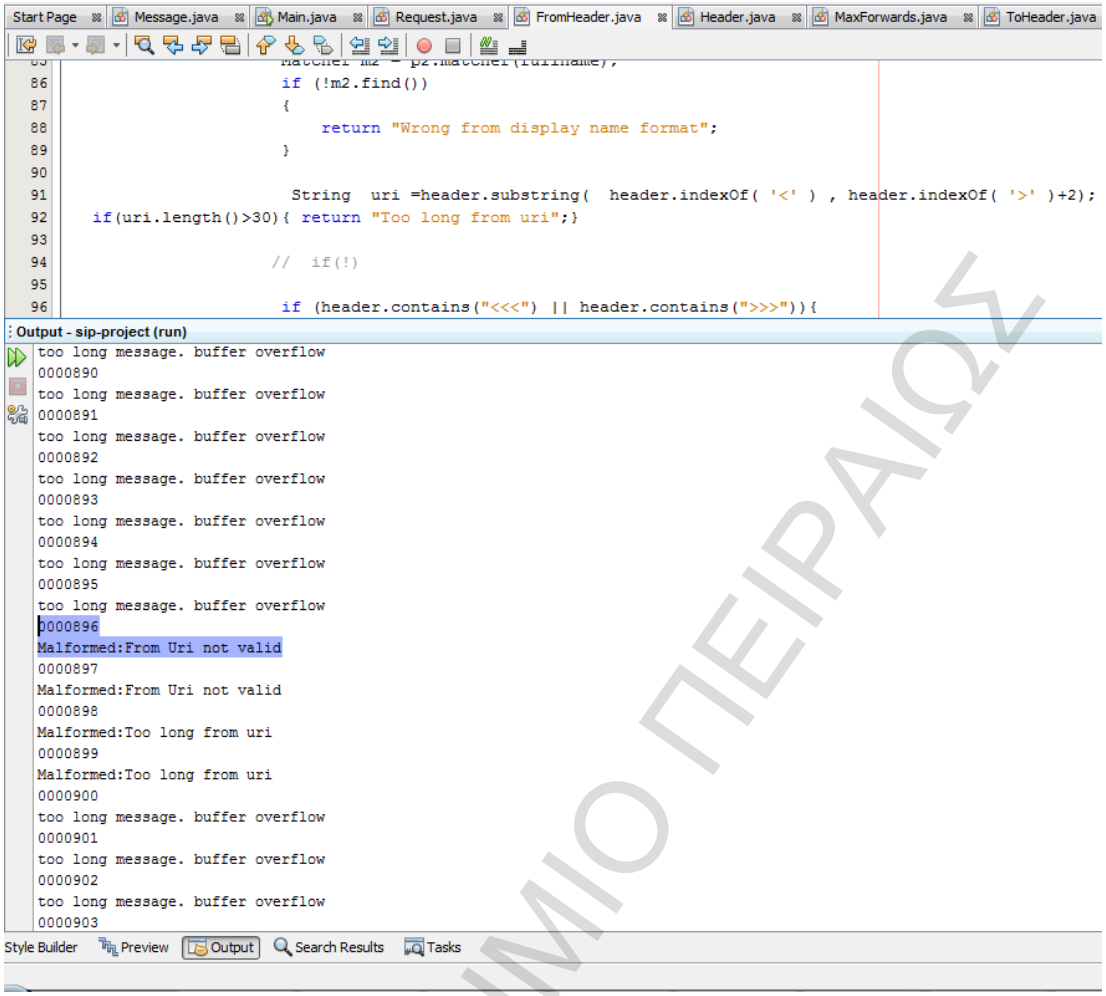
Το μήνυμα αυτό διαθέτει λάθος τιμή στην From κεφαλίδα καθώς η διεύθυνση sip που θα έπρεπε να περιέχει δεν έχει σωστή μορφή. Το λάθος εντοπίζεται πρώτα στο ότι θα έπρεπε να ξεκινάει με «sip:» και όχι με «aaaaa:». Η διεύθυνση που ακολουθεί επίσης δεν είναι έγκυρη αλλά το μοντέλο που ακολουθούμε έχει ήδη εντοπίσει κακή μορφή. Παρακάτω παρατίθεται ο κώδικας που εξετάζει το URI:

```
protected boolean parseUri(String parseinput)
{
    String inputline;
    String userinfo;
    String hostport;
    String parameters;
    boolean isvalid;
    isvalid = false;

    inputline = parseinput.trim();

    if( inputline.indexOf( '<' ) >= 0 ) {

        if( inputline.indexOf( '>' ) == -1 ) {
            return false;
        }


        String beforeangle = inputline.substring( 0, inputline.indexOf( '>' ) );
        parseUri( beforeangle.substring( beforeangle.indexOf( '<' ) + 1 ));

        return false;
    }

    if( inputline.toLowerCase().startsWith( "sip:" ) || inputline.toLowerCase().startsWith( "sips:" )
        ) {
            int colonPos = inputline.indexOf( ':' );

            inputline = inputline.substring( colonPos + 1 );
    }


    if( inputline.indexOf( '@' ) >= 0 ) {
        userinfo = inputline.substring( 0, inputline.indexOf( '@' ) );

        if(userinfo.length() > 35){ isvalid=false; return isvalid;}

            if(userinfo == null ||userinfo.equals(" ")||userinfo.equals("")){ return false;}
```

```
            if(userinfo.contains("%") || userinfo.contains(".") || userinfo.contains("/") || userinfo.contains("\\")|| userinf
                return false;
            }

            if(!isValidUTF8(userinfo.getBytes())){
                return isvalid;
            }

for(int i=0; i<userinfo.length(); i++ ){

                if( userinfo.charAt(i) == (char)0x00 || userinfo.charAt(i) == (char)0x61 || userinfo.charAt(i) == (char)0x08
                    isvalid = false;

                }
            }
            inputline = inputline.substring(inputline.indexOf( '@' ) + 1 );
        }

        // If we have any parameters
        if( inputline.indexOf( ';' ) >= 0 ) {
            hostport = inputline.substring( 0, inputline.indexOf( ';' ) ).trim();
            inputline = inputline.substring( inputline.indexOf( ';' ) );
        } else if( inputline.indexOf( '?' ) >= 0 ) {
            hostport = inputline.substring( 0, inputline.indexOf( '?' ) ).trim();
            inputline = inputline.substring( inputline.indexOf( '?' ) + 1);

        }

        isvalid = true;

        return isvalid;
    }
```

Η μέθοδος αυτή καλείται κατά τον έλεγχο της κεφαλίδας From και επιστρέφει false όταν βρεθεί κάποιο λάθος στο URI.

Η απάντηση του μοντέλου μας στον παραπάνω έλεγχο είναι «`Malformed: From Uri not valid`»



13. ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως φάνηκε από την παραπάνω ανάλυση οι επιθέσεις κακοσχηματισμένων μηνυμάτων είναι πολύ εύκολο να δημιουργηθούν από κάποιον κακόβουλο χρήστη και εξαιρετικά δύσκολο να εντοπιστούν με ακρίβεια από τον parser του μηνύματος. Οι υλοποιήσεις που έχουμε δει ως τώρα δεν δίνουν ιδιαίτερη έμφαση στον εντοπισμό κακοσχηματισμένων μηνυμάτων. Καθώς όμως η διάδοση της χρήσης του SIP πρωτοκόλλου αυξάνεται και γίνεται όλο και πιο συχνά στόχος επιθέσεων μια τέτοια υλοποίηση αρχίζει να γίνεται πολύ σημαντική.

Ο parser δεν είναι σε θέση να εντοπίσει όλα τα κακόβουλα μηνύματα τα οποία μπορεί να δεχτεί και συχνά προκαλούνται σοβαρά προβλήματα. Η υλοποίηση του παραπάνω μοντέλου και η αξιοποίηση του πριν το μήνυμα φτάσει στον parser θα μειώσει ακόμα περισσότερο τις επιθέσεις κακοσχηματισμένων μηνυμάτων.

14. ΠΑΡΑΡΤΗΜΑ ΑΚΡΩΝΥΜΙΩΝ

DoS – Denial of Service: Άρνηση της Υπηρεσίας.

HTTP – Hypertext Transfer Protocol: Πρωτόκολλο Μεταφοράς Υπερκειμένου.

IP – Internet Protocol: Πρωτόκολλο Διαδικτύου.

RTP – Real-time Transport Protocol: Πρωτόκολλο Μεταφοράς σε Πραγματικό Χρόνο.

URI – Uniform Resource Identifier: Ενιαίων Αναγνωριστικών Πόρων.

URL – Uniform Resource Locator: Ενιαίων Εντοπιστών Πόρων

TCP – Transmission Control Protocol: Πρωτόκολλο Ελέγχου Μετάδοσης.

TLS – Transport Layer Security: Ασφάλεια Επιπέδου Μεταφοράς.

SDP – Session Description Protocol: Πρωτόκολλο Περιγραφής Συνόδου.

SIP – Session Initiation Protocol: Πρωτόκολλο Έναρξης Περιόδου Εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, T. Dagiuklas, and S. Gritzalis. A framework for protecting a SIP-based infrastructure against malformed message attacks.

[2] Ove Liljeqvist, Service Availability for Session Initiation Protocol in a hostile Environment

[3] PROTOS test-suite: c07-sip. http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/

[4] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, T. Dagiuklas, and S. Gritzalis.Novel Protecting Mechanism for SIP-Based Infrastructure against Malformed Message Attacks: Performance Evaluation Study

[4] K. Rieck, S. Wahl, P.Laskov, P.Domschitz, Klaus-Robert M¨uller, A Self-Learning System for Detection of Anomalous SIP Messages

[5] http://en.wikipedia.org/

[6] http://tools.ietf.org/html/rfc3261

[7] http://tools.ietf.org/html/rfc4566

[8] C. Kruegel, T. Toth, and E. Kirda. Service specific anomaly detection for network intrusion detection.

[9] Karthik Budigere Ramakrishna, Helsinki University of Technology,Defending against common attacks in SIP

[10] Dongwon Seo, Heejo Lee, and Ejovi Nuwere, Detecting More SIP Attacks on VoIP Services by Combining Rule Matching and State Transition Models

[11] C. Wieser, M. Laakso, H. Schulzrinne, Security testing of SIP implementations

[12] Sven Ehlert, Ge Zhang, Dimitris Geneiatakis, Georgios Kambourakis, Tasos Dagiuklas, Ji˘r´ı Markl and Dorgham Sisalem, Two Layer Denial of Service Prevention on SIP VoIP Infrastructures

[13] D. Geneiatakis, C. Lambrinoudakis, An Ontology Description for SIP Security Flaws

[14] Jiangbo Yin, MSc THESIS , Session Initiation Protocol Benchmark Suite

[15] Lucas E. Dobson, Master thesis, SECURITY ANALYSIS OF SESSION INITIATION PROTOCOL

[16] Hongbin Li, Hu Lin, Xuehua Yang, Feng Liu, A RULES-BASED INTRUSION DETECTION AND PREVENTION FRAMEWORK AGAINST SIP MALFORMED MESSAGES ATTACKS

[17] S. Niccolini, R. G. Garroppo, S. Giordano B, G. Risi, S. Ventura , SIP Intrusion Detection and Prevention: Recommendations and Prototype Implementation