ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Σχολη Θετικων Επιστημων

Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών

Μεταπτυχιακή Διπλωματική Εργασία

Πρωτόκολλα Αυθεντικοποίησης (authentications protocols)

Εμμανουήλ – Ζαφείριος Μπόζης,

Α.Μ. 2009018

Επιβλέπων

Νικόλαος Κολοκοτρώνης, *Λέκτορας*

Σεπτέμβριος 2011

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας εργασίας είναι η ανάλυση της ασφάλειας και του τρόπου λειτουργίας ευρέως χρησιμοποιούμενων πρωτοκόλλων αυθεντικοποίησης, όπως είναι το SSL/TLS, το IPsec και το SSH. Μέσα από αυτή την ανάλυση επιχειρείται η ανάδειξη των διαφορών τους όσον αφορά την υιοθέτησή τους, αλλά και θεμάτων ασφαλείας που έχουν πρόσφατα δει το φως και αφορούν στα ίδια τα πρωτόκολλα ή σε υλοποιήσεις τους. Καταβλήθηκε προσπάθεια η εργασία αυτή να είναι όσο το δυνατόν πιο ενημερωμένη με τις τελευταίες εξελίξεις και να περιλαμβάνει τις πλέον πρόσφατες εργασίες που έχουν δημοσιευτεί και σχετίζονται με την ασφάλεια και τη χρήση των πρωτοκόλλων αυθεντικοποίησης. Ωστόσο η ανάλυση πραγματοποιήθηκε στον βαθμό που είναι εφικτή, καθόσον νέες εκδόσεις και θέματα ασφαλείας ανακύπτουν σχεδόν σε καθημερινή βάση, ενώ οι υλοποιήσεις των πρωτοκόλλων είναι σε διαρκή εξέλιξη. Ιδιαίτερο βάρος δόθηκε στην ανάλυση ασφαλείας του IPsec σε λειτουργία μόνο κρυπτογράφησης (encryption only), με αναφορά σε επιθέσεις, τόσο σε συγκεκριμένη υλοποίησή του πρωτοκόλλου στο Linux, όσο και στο ίδιο το πρωτόκολλο, όπως αυτό προσδιορίζεται στα σχετικά κείμενα RFC.

Στην εργασία αυτή, η ανάλυση των μηχανισμών των πρωτοκόλλων γίνεται χωρίς να εξαντλεί κάποιες ιδιαίτερες παραμέτρους και λεπτομέρειες των ίδιων των πρωτοκόλλων, τις οποίες μπορεί κανείς να αναζητήσει στα σχετικά κείμενα RFC. Η παρουσίαση επικεντρώνεται στους μηχανισμούς λειτουργίας των πρωτοκόλλων αυθεντικοποίησης και όχι στα χαρακτηριστικά των κρυπτογραφικών αλγορίθμων που χρησιμοποιούν, τα οποία θεωρούνται ότι είναι στη βάση τους γνωστά στον αναγνώστη.

Τέλος, περιλαμβάνεται ένα παράδειγμα χρήσης του πρωτοκόλλου SSL για τη δημιουργία ενός ασφαλούς καναλιού για τη μετάδοση των δεδομένων μιας εφαρμογής server – client σε περιβάλλον win32. Σε αυτό το παράδειγμα παρουσιάζεται η αναγκαία παραμετροποίηση του προγράμματος stunnel στην μεριά του εξυπηρετητή και του πελάτη, καθώς και τα βήματα που πρέπει να ακολουθήσει κάποιος για να συνδέσει την εφαρμογή με το ασφαλές κανάλι SSL, χωρίς να τροποποιήσει τον κώδικα της εφαρμογής, επωφελούμενος από τα εργαλεία που του παρέχει η υλοποίηση ανοικτού κώδικα OpenSSL.

Πίνακας Περιεχομένων

Κατάλογος Σχημάτων

Κατάλογος Συντομογραφιών

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AH	Authentication Header
BEEP	Blocks Extensible Exchange Protocol
BPP	Binary Packet Protocol
CA	Certificate Authority
CBC	Cipher Block chaining
CTR	Counter Mode
DCCP	Datagram Congestion Control Protocol
DNS	Domain Name System
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
ESP	Encapsulating Security Payload
FISH	Files transferred over Shell protocol
GSSAPI	Generic Security Services Application Program Interface
HMAC	Hash-based Message Authentication Code
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message protocol
IETF	Internet Engineering Task Force
IHL	Internet Header Length
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
IV	Initialization Vector
MAC	Message Authentication Code
NAT	Network Address Translation
NH	Next header
PAM	Pluggable Authentication Modules
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PL	Pad length
PRF	Pseudorandom Function

SA	Security Association
SIGMA	SIGn-and-MAc
SNP	Secure Network Programming
SNP	Secure Network Programming
SPD	Security Policy Database
SSH	Secure Shell
SSL	Secure Sockets Layer
TFC	Traffic Flow confidentiality
TLS	Transport Layer Security
VPN	Virtual Private Networks

1. Εισαγωγή

1.1. Πρωτόκολλα αυθεντικοποίησης και μοντέλα επιθέσεων και ασφάλειας κρυπτογραφικών συστημάτων

Η ταυτοποίηση και αυθεντικοποίηση του χρήστη αποτελούν βασικές απαιτήσεις της ασφάλειας υπολογιστικών συστημάτων. Οι χρήστες αυθεντικοποιούνται όταν αποκτούν πρόσβαση σε υπολογιστικά συστήματα, στη δουλειά ή στο σπίτι κάθε μέρα. Ωστόσο στις περισσότερες περιπτώσεις υπάρχει άγνοια ως προς τον τρόπο και το επίπεδο ασφαλείας της αυθεντικοποίησης, καθώς και των επιπτώσεων της επιλογής ενός τρόπου αυθεντικοποίησης έναντι κάποιου εναλλακτικού. Ένα πρωτόκολλο αυθεντικοποίησης είναι ένας τύπος κρυπτογραφικού πρωτοκόλλου, που έχει ως στόχο την αυθεντικοποίηση των μερών που θέλουν να επικοινωνήσουν.

Η ικανότητα του αντιπάλου να επιτεθεί σε ένα πρωτόκολλο αυθεντικοποίησης κρίνεται από τους υπολογιστικούς πόρους που διαθέτει και την πρόσβαση που έχει στο κρυπτοκείμενο, το απλό κείμενο και στο σύστημα που υλοποιεί το πρωτόκολλο. Οι δυνατότητες επίθεσης ενός αντιπάλου σε ένα σύστημα κρυπτογράφησης, χωρίζονται στις ακόλουθες κατηγορίες:

- Επίθεση στο κρυπτοκείμενο (ciphertext-only). Ο αντίπαλος έχει πρόσβαση μόνο σε ορισμένα κομμάτια του κρυπτοκειμένου και ο αντικειμενικός του σκοπός είναι να αποκρυπτογραφήσει το κρυπτοκείμενο αυτό ή να ανακαλύψει το αντίστοιχο κλειδί. Ένα κρυπτοσύστημα το οποίο είναι ευάλωτο σε τέτοιες επιθέσεις θεωρείται ανασφαλές.
- Επίθεση σε γνωστό απλό κείμενο (known-plaintext). Ο αντίπαλος γνωρίζει αντιστοιχίες κρυπτοκειμένου με απλό κείμενο και ο αντικειμενικός του σκοπός είναι η ανακάλυψη του αντίστοιχου κλειδιού. Πολλές φορές συναντάμε μηνύματα όπου η αρχή και το τέλος τους είναι τυποποιημένα, όπως «αγαπητέ κ.…» και «με εκτίμηση ……». Στον κόσμο των δικτύων των υπολογιστών τα πρωτόκολλα επικοινωνίας εμφανίζουν συστηματικά τυποποιημένα μηνύματα. Ένα κρυπτοσύστημα το οποίο υποπίπτει σε επίθεση γνωστού απλού κειμένου θεωρείται ανασφαλές.

- Επίθεση σε επιλεγμένο απλό κείμενο (chosen-plaintext). Ο αντίπαλος έχει την δυνατότητα πρόσβασης στο κρυπτοσύστημα όπου δεν γνωρίζει το κλειδί και μπορεί να ζητά την κρυπτογράφηση μηνυμάτων. Με αυτόν τον τρόπο μπορεί να ανακαλύψει την αντιστοιχία του απλού κειμένου με το άγνωστο κρυπτοκείμενο.
- Επίθεση προσαρμόσιμου επιλεγμένου απλού κειμένου (adaptive chosen-plaintext). Ο αντίπαλος είναι σε θέση να πραγματοποιήσει επίθεση με επιλεγμένο απλό κείμενο, αλλά επιπλέον μπορεί να εφαρμόσει μεθοδολογία σύμφωνα με την οποία η επόμενη επιλογή του απλού κειμένου εξαρτάται από τις προηγούμενες, προκειμένου να ανακαλύψει το κλειδί ταχύτερα, από την εξαντλητική αναζήτηση (exhaustive search).
- Επίθεση με επιλεγμένο κρυπτοκείμενο (chosen-ciphertext). Υποθέτοντας ότι ο αντίπαλος έχει πρόσβαση στον αλγόριθμο αποκρυπτογράφησης, ο αντικειμενικός σκοπός του είναι να ανακαλύψει το κλειδί αποκρυπτογράφησης προκειμένου να μπορεί να αποκρυπτογραφεί νέα κρυπτοκείμενα, όταν δεν θα έχει πρόσβαση στον αλγόριθμο αποκρυπτογράφησης. Στα περισσότερα συμμετρικά κρυπτοσυστήματα η επίθεση αυτή έχει την ίδια ισχύ με την επίθεση του επιλεγμένου απλού κειμένου. Η επίθεση με επιλεγμένο κρυπτοκείμενο θεωρείται ως η πιο αυστηρή επίθεση.
- Επίθεση προσαρμόσιμου επιλεγμένου κρυπτοκειμένου (adaptive chosen-ciphertext). Η επίθεση αυτή είναι αντίστοιχη του προσαρμόσιμου επιλεγμένου απλού κειμένου, με τη διαφορά ότι ο αντίπαλος έχει πρόσβαση στον αλγόριθμο αποκρυπτογράφησης.

Η δύναμη ενός κρυπτοσυστήματος να αντιστέκεται στις επιθέσεις του αντιπάλου είναι ένα αντικείμενο το οποίο μπορεί να εξεταστεί από πολλές πλευρές. Η ανάγκη καθορισμού αντικειμενικών μέτρων για τη μέτρηση της κρυπτογραφικής δύναμης είχε ως αποτέλεσμα τη δημιουργία διάφορων μαθηματικών μοντέλων. Τα μοντέλα αυτά συνοπτικά είναι τα παρακάτω:
- *Ασφάλεια άνευ όρων (unconditionally secure)*. Ένα κρυπτοσύστημα είναι άνευ όρων ασφαλές όταν το κρυπτοκείμενο δεν δίνει καμιά πληροφορία στον αντίπαλο σχετικά με το απλό κείμενο. Η υπόθεση απαιτεί ότι ο αντίπαλος έχει άπειρη υπολογιστική ισχύ στην διάθεσή του. Το μοντέλο αυτό διατυπώθηκε από τον Shannon το 1949 στην εργασία του με τίτλο “Communication Theory of Secrecy Systems”, όπου η ασφάλεια εξετάζεται κάτω από το πρίσμα της θεωρίας πληροφορίας. Σύμφωνα με τη θεωρία της πληροφορίας, ένα

κρυπτοσύστημα είναι άνευ όρων ασφαλές όταν η πιθανότητα που έχει ο αντίπαλος να σπάσει το κρυπτοκείμενο είναι ίδια με την πιθανότητα που θα έχει εάν του δοθεί λύση για ένα τμήμα του κρυπτοκειμένου.

- *Υπολογιστική ασφάλεια (computationally secure).* Σε αυτό το μοντέλο εισάγεται πλέον η παράμετρος της υπολογιστικής ισχύος του αντιπάλου. Ένα κρυπτοσύστημα είναι υπολογιστικά ασφαλές, όταν προκειμένου να το σπάσει ο αντίπαλος απαιτείται υπολογιστική ισχύς πέραν των δυνατοτήτων του. Ο υπολογισμός γίνεται με βάση τον καλύτερο αλγόριθμο που γνωρίζει ο αντίπαλος προκειμένου να σπάσει το κρυπτοσύστημα. Ο προφανής αλγόριθμος που έχει για να σπάσει ένα κρυπτοσύστημα είναι αυτός της εξαντλητικής αναζήτησης, όπου ο αντίπαλος δοκιμάζει σειριακά τα κλειδιά έως ότου ανακαλύψει το σωστό. Ο αναμενόμενος χρόνος ανακάλυψης του σωστού κλειδιού είναι ανάλογος του μισού του συνολικού αριθμού των κλειδιών. Σε ορισμένα κρυπτοσυστήματα έχουν ανακαλυφθεί και πιο έξυπνοι αλγόριθμοι αναζήτησης κλειδιών, που φθάνουν στο επιθυμητό αποτέλεσμα πιο γρήγορα από την εξαντλητική αναζήτηση. Συνεπώς, η υπολογιστική ασφάλεια δεν εγγυάται την ασφάλεια του κρυπτοσυστήματος, επειδή στο μέλλον μπορεί να ανακαλυφθεί αλγόριθμος κρυπτανάλυσης ο οποίος να μπορεί να εκτελεστεί εντός των υπολογιστικών δυνατοτήτων του αντιπάλου.

- *Ασφάλεια θεωρητικής πολυπλοκότητας (complexity theoretic).* Θεωρείται ότι ο αντίπαλος μπορεί να πραγματοποιήσει επίθεση στο κρυπτοσύστημα η οποία απαιτεί πολυωνυμική υπολογιστική ισχύ. Δηλαδή, οι παράμετροι ασφάλειας του κρυπτοσυστήματος μπορούν να εκφραστούν πολυωνυμικά ως προς τον χώρο και τον χρόνο. Η ανάλυση με βάση το μοντέλο ασφάλειας θεωρητικής πολυπλοκότητας, εξετάζει ασυμπτωτικά την αντοχή του κρυπτοσυστήματος σε κρυπταναλυτικές επιθέσεις και δεν έχει πρακτική αξία. Ωστόσο, μια τέτοια ανάλυση μπορεί να οδηγήσει στη διαπίστωση θεμελιωδών εννοιών και αρχών ασφάλειας των κρυπτοσυστημάτων.

- *Αποδείξιμη ασφάλεια (provable security).* Ένα κρυπτοσύστημα είναι αποδείξιμα ασφαλές όταν μπορούμε να αποδείξουμε ότι η ασφάλειά του είναι ισοδύναμη κάποιου γνωστού και καλά μελετημένου προβλήματος που θεωρείται «δύσκολο». Παραδείγματα τέτοιων προβλημάτων βρίσκουμε στην θεωρία αριθμών, όπως η παραγοντοποίηση ενός μεγάλου σύνθετου αριθμού στους πρώτους παράγοντές του και ο υπολογισμός του διακριτού λογάριθμου ενός αριθμού. Τα κρυπτοσυστήματα που είναι αποδείξιμα ασφαλή ανήκουν σε υποσύνολο των συστημάτων που είναι υπολογιστικά ασφαλή, αλλά ένα

κρυπτοσύστημα αποδείξιμης ασφάλειας έχει πολύ καλές προοπτικές να είναι ασφαλές, αφού το υποκείμενο δύσκολο πρόβλημα έχει υποστεί εκτενείς μελέτες και είναι γενικώς αποδεκτό ως «δύσκολο».

2. Το πρωτόκολλο IPsec

Το IPsec (Internet Protocol Security) είναι ένα σύνολο πρωτοκόλλων που εξασφαλίζει την ασφάλεια της επικοινωνίας, αυθεντικοποιώντας και κρυπτογραφώντας κάθε πακέτο IP. Το IPsec περιλαμβάνει επίσης και πρωτόκολλα για την επίτευξη αμοιβαίας αυθεντικοποίησης μεταξύ των επικοινωνούντων μερών στην αρχή κάθε συνεδρίας (session) και συμφωνίας κρυπτογραφικών κλειδιών που θα χρησιμοποιηθούν στη διάρκεια της.

Το IPsec υλοποιεί ένα σχήμα ασφαλείας μεταξύ δύο σημείων (end-to-end) που μπορεί να χρησιμοποιηθεί για να προστατέψει ροές δεδομένων μεταξύ δύο ξενιστών (host to host), μεταξύ δύο πυλών ασφαλείας (network-to-network) ή μεταξύ μιας πύλης ασφαλείας και ενός ξενιστή (network-to-host). Οι εφαρμογές δεν χρειάζεται να είναι ειδικά σχεδιασμένες ώστε να μπορούν να αξιοποιήσουν το IPsec, καθόσον αυτό λειτουργεί στο Internet Layer του πρωτοκόλλου IP [06], και όχι στα ανώτερα στρώματα (layers) της στοίβας πρωτοκόλλων, όπως στην περίπτωση του SSL-TLS και του SSH (Secure Shell).

Το IPsec είναι ο διάδοχος του Network Layer Security Protocol (NLSP) που έχει προτυποποιηθεί από τον οργανισμό ISO. Το NLSP βασίστηκε στο πρωτόκολλο SP3 που δημοσιεύτηκε από τον οργανισμό NIST, αλλά σχεδιάστηκε στα πλαίσια του έργου Secure Data Network System της NSA (National Security Agent) [18].

Το IPsec έχει προτυποποιηθεί από την IETF (Internet Engineering Task Force) σε μια σειρά από κείμενα RFCs (Request for Comment), επεξηγώντας διάφορα τμήματά του και επεκτάσεις του.

2.1. Αρχιτεκτονική του IPsec και κρυπτογραφικοί αλγόριθμοι που χρησιμοποιεί

Το IPsec χρησιμοποιεί διάφορα πρωτόκολλα που εκτελούν αντίστοιχες λειτουργίες, γι' αυτό και θεωρείται ως ανοικτό πρότυπο. Τα πρωτόκολλα που χρησιμοποιεί είναι το IKE (Internet Key Exchange), το AH (Authentication Header) και το ESP (Encapsulating Security Payload).

Το IKE αναλαμβάνει τη συμφωνία σε ένα σύνολο από κρυπτογραφικούς αλγορίθμους που θα χρησιμοποιηθούν για την κρυπτογράφηση και αυθεντικοποίηση προς μια συγκεκριμένη κατεύθυνση των δεδομένων, καθώς και των αντίστοιχων κλειδιών. Καθένα από τα δύο μέρη της επικοινωνίας μπορεί να εκκινήσει τη διαδικασία εδραίωσης κλειδιού, ορίζοντας έτσι και μια κατεύθυνση επικοινωνίας. Το σύνολο των κλειδιών και μεθόδων κρυπτογράφησης και αυθεντικοποίησης αποτελεί ένα συσχετισμό ασφαλείας (security Association) του IPsec. Το SPI (Security Parameter Index), είναι ένας δείκτης προς τη βάση των συσχετισμών ασφαλείας (security association database ή SADB). Μια διαδικασία εδραίωσης κλειδιού με το IKE, καταλήγει σε δύο συσχετισμούς ασφαλείας, έναν εισερχόμενο και έναν εξερχόμενο. Διαφορετικά SPIs για κάθε συσχετισμό ασφαλείας (το ένα επιλέγεται από αυτόν που εκκινεί τη διαδικασία IKE και το άλλο από το μέρος που απαντάει) εγγυώνται διαφορετικό κλειδί για κάθε κατεύθυνση [07].

Η επιλογή των συγκεκριμένων αλγορίθμων κρυπτογράφησης και αυθεντικοποίησης από μια προκαθορισμένη λίστα διατίθεται στους διαχειριστές του πρωτοκόλλου. Στην περίπτωση πολλαπλών αποδεκτών (multicast), ο συσχετισμός ασφαλείας παρέχεται για όλη την ομάδα και αναπαράγεται για κάθε εξουσιοδοτημένο μέλος της. Μπορεί να υπάρχουν και περισσότεροι από ένας συσχετισμοί ασφαλείας (SA) για μια ομάδα, χρησιμοποιώντας διαφορετικούς δείκτες SPI, επιτρέποντας έτσι διαφορετικά επίπεδα και ρυθμίσεις ασφαλείας μεταξύ των μελών της ομάδας. Κάθε αποστολέας μπορεί να έχει πολλαπλά SA, καθόσον ο δέκτης μπορεί μόνο να ξέρει ότι κάποιος που γνωρίζει τα κλειδιά έχει στείλει μήνυμα αίτησης αυθεντικοποίησης. Το σχετικό πρότυπο μάλιστα, δεν περιγράφει πώς επιλέγεται ο συσχετισμός ασφάλειας SA και μοιράζεται στην ομάδα, καθόσον υποτίθεται ότι το εξουσιοδοτημένο μέλος έχει κάνει αυτό την επιλογή του [18].
Τα πρωτόκολλα ESP και AH αναλαμβάνουν τη μορφοποίηση των πακέτων IPsec, βάση αυτών που έχουν προηγουμένως συμφωνηθεί από τα δύο μέρη της επικοινωνίας με το IKE.

Για να διασφαλιστεί η διαλειτουργικότητα (interoperability) μεταξύ των διαφορετικών υλοποιήσεων, είναι απαραίτητο να προσδιοριστεί ένα σύνολο από υποχρεωτικούς αλγορίθμους που θα πρέπει να υποστηρίζουν όλες οι

υλοποιήσεις, ώστε να υπάρχει πάντα διαθέσιμος ένας κοινός αλγόριθμος. Μια σειρά από κείμενα RFC που δεν επηρεάζουν τον τρόπο εφαρμογής των πρωτοκόλλων ESP και AH, αναφέρονται στους αλγόριθμους που πρέπει υποχρεωτικά να υποστηρίζουν αυτά τα πρωτόκολλα, καθώς και στους αλγορίθμους που προτείνεται να υποστηρίζουν, καθόσον αυτοί στο μέλλον θα γίνουν υποχρεωτικοί με μεγάλη πιθανότητα.

Όπως ήταν αναμενόμενο, νέοι κρυπτογραφικοί αλγόριθμοι προτείνονται συνεχώς και οι υπάρχοντες υπόκεινται σε νέες επιθέσεις. Ένας αλγόριθμος ο οποίος θεωρείται ασφαλής σήμερα, μπορεί να αποδειχθεί ανασφαλής αύριο. Δεδομένου αυτού, η επιλογή για ένα αλγόριθμο ως υποχρεωτικό για την υλοποίηση του πρωτοκόλλου, θα πρέπει να είναι συντηρητική, ώστε να μην υπάρχει πιθανότητα αυτός ο αλγόριθμος να παραβιαστεί σύντομα. Επίσης, θα πρέπει να ληφθεί υπόψη και η απόδοση του αλγορίθμου, αφού αποτελεί σημαντικό παράγοντα στο περιβάλλον χρήσης IPsec. Οι υποχρεωτικοί για υλοποίηση αλγόριθμοι θα πρέπει να αλλάζουν με την πάροδο του χρόνου, ώστε να προσαρμόζονται στις μεταβαλλόμενες συνθήκες. Για τον λόγο αυτό, τα κείμενα RFC που αναφέρονται στη δομή και λειτουργία των ESP και AH, δεν αναφέρονται σε συγκεκριμένους κρυπτογραφικούς αλγορίθμους, ώστε να δώσουν την ευελιξία στο IPsec να εξελίσσεται.

Οι αλγόριθμοι που χρησιμοποιούνται για το ESP, όπως αυτοί αναφέρονται στο σχετικό κείμενο RFC 4835 παρουσιάζονται στο παρακάτω σχήμα 1.

Requirement	Encryption Algorithm (notes)
MUST	NULL [RFC2410] (1)
MUST	AES-CBC with 128-bit keys [RFC3602]
MUST-	TripleDES-CBC [RFC2451]
SHOULD	AES-CTR [RFC3686]
SHOULD NOT	DES-CBC [RFC2405] (2)
MUST	HMAC-SHA1-96 [RFC2404] (3)
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	NULL (1)
MAY	HMAC-MD5-96 [RFC2403] (4)

Σχήμα 1. Κρυπτογραφικοί αλγόριθμοι του ESP

Η κατάταξη των αλγορίθμων σε must, should, should not και may δηλώνει ότι ο αλγόριθμος είναι υποχρεωτικός, προτείνεται, δεν προτείνεται και είναι προαιρετικός αντίστοιχα. Ο NULL δηλώνει τη μη ύπαρξη κρυπτογράφησης. Οι αλγόριθμοι που κατατάσσονται ως “must” ή “should” δεν είναι γνωστό να έχουν παραβιαστεί την τρέχουσα χρονική περίοδο και η έως σήμερα κρυπτογραφική

έρευνα έχει δείξει ότι θα παραμείνουν ασφαλείς στο προσεχές μέλλον. Σε αντίθετη περίπτωση, νέες εκδόσεις κειμένων RFC θα εκδοθούν, στις οποίες θα αντανακλώνται οι καλύτερες σύγχρονες πρακτικές στον χώρο.

Καθόσον η κρυπτογράφηση στο ESP είναι προαιρετική, η υποστήριξη του αλγορίθμου NULL απαιτείται για να υπάρχει συμβατότητα με τον τρόπο που λειτουργούν οι υπηρεσίες του IPsec. Ενώ και η αυθεντικοποίηση και η κρυπτογράφηση μπορούν να είναι NULL, δεν μπορούν να είναι ταυτόχρονα και οι δύο NULL. Ο DES με το μικρό μήκος κλειδιού του και με το δημοσίως παρουσιασμένο ανοιχτής σχεδίασης και ειδικού σκοπού hardware για την παραβίασή του, είναι αμφιβόλου ασφαλείας προς γενική χρήση [10].

Στο SHA-1 υπάρχουν ενδείξεις αδυναμίας, ωστόσο αυτές δεν επηρεάζουν τη χρήση του SHA-1 σε συνδυασμό με το HMAC. Επίσης και στον MD5 υπάρχουν ενδείξεις αδυναμίας, ωστόσο αυτές δεν επηρεάζουν την χρήση του MD5 με HMAC [10].

Στο σχήμα 2, παρουσιάζονται οι απαιτήσεις από το σχετικό κείμενο RFC 4835 για την υλοποίηση του AH, όσον αφορά στους αλγόριθμους ασφαλείας.

Requirement	Algorithm (notes)
MUST	HMAC-SHA1-96 [RFC2404] (1)
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	HMAC-MD5-96 [RFC2403] (2)

Σχήμα 2. Αλγόριθμοι αυθεντικοποίησης (authentication) του ESP

Στο κείμενο RFC4303 περιγράφονται αλγόριθμοι συνδυασμένης λειτουργίας, οι οποίοι παρέχουν υπηρεσίες εμπιστευτικότητας και αυθεντικοποίησης. Το ESP μπορεί να υποστηρίξει τέτοιους αλγορίθμους με κατάλληλη υλοποίηση της δομής του πακέτου ESP. Σε πολλές περιπτώσεις, οι αλγόριθμοι συνδυασμένης λειτουργίας παρέχουν πλεονεκτήματα, μεταξύ των οποίων η αποδοτικότητα και η ταχύτητα. Παρόλο που σήμερα δεν υπάρχουν προτεινόμενοι ή απαιτούμενοι τέτοιοι αλγόριθμοι στο IPsec, οι AES-CCM και AES-GCM είναι υπό εξέταση. Οι CCM (Counter with CBC-MAC) και GCM (Galois Counter Mode) είναι τρόποι κρυπτογράφησης με αυθεντικοποίηση που στη συγκεκριμένη περίπτωση χρησιμοποιούν τον AES, αλλά μπορούν να χρησιμοποιήσουν και οποιονδήποτε κρυπταλγόριθμο τμήματος 128 bit [46].

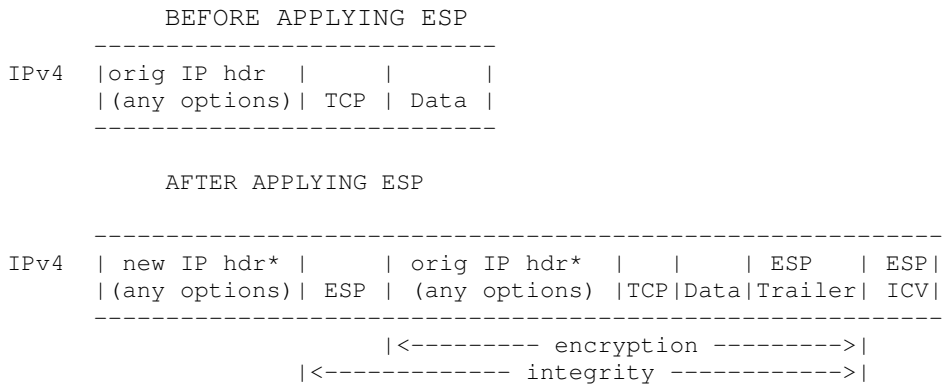
Ο AES-CCM έχει υιοθετηθεί ως προτεινόμενη λειτουργία στο IEEE 802.11i και ο AES-GCM στο IEEE 802.1ae, γνωστό επίσης και ως MACsec [10].

Η ασφάλεια των συστημάτων που βασίζονται στην κρυπτογραφία, εξαρτάται από τη δύναμη των επιλεγόμενων κρυπταλγορίθμων, καθώς και τη δύναμη των κλειδιών που χρησιμοποιούνται από αυτούς. Η ασφάλεια, επίσης, βασίζεται στον σχεδιασμό και παραμετροποίηση του πρωτοκόλλου που χρησιμοποιείται από το σύστημα, ώστε να διασφαλιστεί ότι δεν υπάρχουν τρόποι παράκαμψης των κρυπτογραφικών αλγορίθμων που να ακυρώνουν την ασφάλεια του συνολικού συστήματος.

2.2. Τρόποι λειτουργίας του IPsec

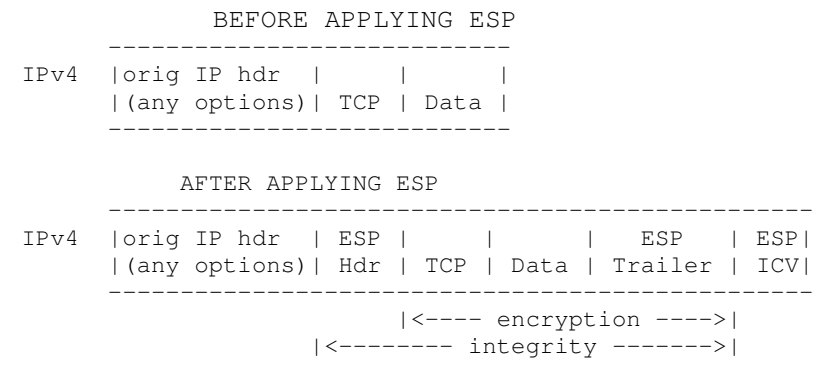
Το IPsec μπορεί να λειτουργήσει με δύο βασικούς τρόπους. Ο ένας είναι της ενθυλάκωσης (tunnel mode) και ο άλλος της μεταφοράς (transport mode). Στη λειτουργία ενθυλάκωσης ολόκληρο το πλαίσιο δεδομένων (IP datagram) του πρωτοκόλλου IP κρυπτογραφείται ή / και αυθεντικοποιείται. Κατά συνέπεια, ένα ολόκληρο πλαίσιο μαζί με τα πεδία ασφαλείας θεωρούνται ως τα δεδομένα προς μεταφορά (payload) ενός εξωτερικού πλαισίου, με τη δικιά του επικεφαλίδα (header), που ονομάζεται εξωτερική επικεφαλίδα. Έτσι, το αρχικό ή εσωτερικό πλαίσιο δεδομένων είναι ενθυλακωμένο στο εξωτερικό πλαίσιο. Όταν εφαρμόζεται η λειτουργία ενθυλάκωσης η επεξεργασία του πρωτοκόλλου IPsec γίνεται σε πύλες ασφαλείας (security gateways) για λογαριασμό των τελικών σημείων επικοινωνίας (endpoint hosts). Οι πύλες ασφαλείας μπορεί να είναι περιμετρικοί δρομολογητές ή firewalls. Η λειτουργία ενθυλάκωσης χρησιμοποιείται για τη δημιουργία εικονικών ιδιωτικών δικτύων (virtual private networks) για διαδικτυακή επικοινωνία (network-to-network), ξενιστή με δίκτυο (host-to-network) και ξενιστή με ξενιστή (host-to-host). Η λειτουργία αυτή υποστηρίζει επικοινωνία με ξενιστή που βρίσκεται πίσω από μια πύλη NAT (Network Address Translation). Στο σχήμα 3, από το κείμενο RFC 4303, φαίνεται η εφαρμογή του τρόπου λειτουργίας ενθυλάκωσης σε ένα πακέτο IP-v4. Το πακέτο αυτό περιέχει μια επικεφαλίδα IP, την επικεφαλίδα του TCP και τα δεδομένα. Από το σχήμα φαίνεται ότι η κρυπτογράφηση εφαρμόζεται σε όλο το πακέτο IP και προστίθενται εξωτερική επικεφαλίδα και ουρά (trailer) για να σχηματιστεί το τελικό πακέτο IPsec που θα αποσταλεί στο δίκτυο.

Περισσότερα για τον τρόπο σχηματισμού του πακέτου αναφέρονται παρακάτω στις σχετικές ενότητες 2.4 και 2.5.



Σχήμα 3. Εφαρμογή λειτουργίας ενθυλάκωσης IPsec σε IP v4.

Στον τρόπο λειτουργίας της μεταφοράς (transport mode) μόνο τα δεδομένα προς μεταφορά (payload) του πακέτου IP κρυπτογραφούνται ή / και αυθεντικοποιούνται (authenticated). Η δρομολόγηση των πακέτων (routing) δεν επηρεάζεται, καθόσον η επικεφαλίδα του πακέτου IP δεν τροποποιείται, ούτε κρυπτογραφείται. Παρόλα αυτά, όταν χρησιμοποιείται το πρωτόκολλο η τιμή της διεύθυνσης IP δεν μπορεί να τροποποιηθεί σε άλλη, καθόσον αυτό θα καθιστούσε την τιμή της συνάρτησης hash μη έγκυρη. Τα στρώματα (layers) της μεταφοράς και της εφαρμογής προστατεύονται πάντα με την τιμή της hash, έτσι ώστε να μην είναι δυνατόν να τροποποιηθούν με κανένα τρόπο (όπως για παράδειγμα εκτρέποντας τα πακέτα σε άλλη θύρα). Στο παρακάτω σχήμα 4, φαίνεται η εφαρμογή της λειτουργίας μεταφοράς σε ένα πακέτο IP-v4. Όπως φαίνεται, η κρυπτογράφηση ξεκινάει αμέσως μετά από την επικεφαλίδα IP και πριν το πρωτόκολλο του ανώτερου επιπέδου που μπορεί να είναι το TCP, UDP, ICMP κλπ.



Σχήμα 4. Εφαρμογή λειτουργίας μεταφοράς IPsec σε IP v4

Η λειτουργία της μεταφοράς χρησιμοποιείται σε επικοινωνία ξενιστή με ξενιστή (host to host), ενώ επίσης υπάρχει τρόπος για τη διάσχιση πύλης NAT με τον μηχανισμό NAT-T. Ο μηχανισμός αυτός προστατεύει το αρχικό IPsec πακέτο ενθυλακώνοντάς το σε ένα άλλο επίπεδο με επικεφαλίδες (headers) UDP και IP. Η επικοινωνία NAT-T κατά τη διάρκεια του πρωτοκόλλου IKE περιγράφεται στο RFC 3947 και η ενθυλάκωση στο πλαίσιο UDP στο RFC 3948.

2.3. Το πρωτόκολλο Internet Key Exchange (IKE)

Κατά τη διάρκεια της δεκαετίας του 1990 διάφορα πρωτόκολλα εγκαθίδρυσης κλειδιού (key establishment protocols) προτάθηκαν για να περιληφθούν στο IPsec. Το IKE αρχικά ορίστηκε τον Νοέμβριο του 1998 από την IETF (Internet Engineering Task Force) σε μια σειρά από δημοσιεύσεις κειμένων RFC, τα RFC 2407, 2408 και 2409. Το IKE αναβαθμίστηκε στην έκδοση 2 τον Δεκέμβριο του 2005, στο σχετικό κείμενο RFC 4306 και αργότερα διευκρινίστηκαν κάποιες λεπτομέρειες του πρωτοκόλλου με τα σχετικά κείμενα αναβαθμίσεων RFC 4718 και RFC 5996. Ο οργανισμός Internet Society (ISOC) κάτω από τον οποίο υπάγεται η IETF, διατηρεί τα δικαιώματα πνευματικής ιδιοκτησίας των προτύπων, ενώ είναι ελεύθερα προσβάσιμα από την κοινότητα του Internet.

Το IKE βασίζεται στα πρωτόκολλα ISAKMP (Internet Security Association and Key Management), Oakley και SKEME. Το ISAKMP είναι ένα απλό πλαίσιο για την ανταλλαγή και διαχείριση κλειδιών, ενώ το IKE είναι μια υλοποίησή του, για χρήση στο IPsec [43]. Η ανταλλαγή κλειδιού βασίζεται στο κρυπτογραφικό πρωτόκολλο ανταλλαγής κλειδιού των Diffie-Hellman [47]. Η ασφάλεια του κρυπτογραφικού αυτού πρωτοκόλλου πηγάζει από τη δυσκολία υπολογισμού διακριτών λογαρίθμων σε ένα πεπερασμένο σώμα.

Οι περισσότερες υλοποιήσεις αποτελούνται από ένα πρόγραμμα daemon που τρέχει στον χώρο του χρήστη και ένα σωρό IPsec στον πυρήνα που επεξεργάζεται τα πακέτα IP. Τα προγράμματα daemon έχουν εύκολη πρόσβαση σε αποθηκευτικό χώρο που περιέχει πληροφορίας ρυθμίσεων, όπως τις διευθύνσεις IP των μερών επικοινωνίας, καθώς και κλειδιά και πιστοποιητικά που απαιτούνται. Οι ρουτίνες του πυρήνα από την άλλη,

μπορούν να επεξεργαστούν τα πακέτα αποτελεσματικά με τον μικρότερο φόρτο, το οποίο είναι σημαντικό για λόγους απόδοσης του συστήματος.

Το πρωτόκολλο IKE χρησιμοποιεί πακέτα UDP, συνήθως στο port 500 και γενικά απαιτεί 4 έως 6 πακέτα με 2 έως 3 χρόνους αποστολής και λήψης για να δημιουργήσει έναν συσχετισμό ασφαλείας SA (security association). Το συμφωνημένο περιεχόμενο κλειδιών δίδεται εν συνεχεία στον σωρό IPsec. Για παράδειγμα, αυτό θα μπορούσε να ήταν ένα κλειδί AES, πληροφορία που προσδιορίζει τα μέρη της επικοινωνίας (endpoints) και τις σχετικές θύρες που πρέπει να προστατευτούν, καθώς και τον τύπο ενθυλάκωσης IPsec που πρέπει να χρησιμοποιηθεί. Το IPsec εν συνεχεία, λαμβάνει τα σχετικά πακέτα και κάνει κρυπτογράφηση / αποκρυπτογράφηση, εάν και όπου απαιτείται. Οι υλοποιήσεις διαφέρουν στο πώς λαμβάνονται τα πακέτα. Για παράδειγμα μερικές χρησιμοποιούν εικονικές συσκευές, άλλες λαμβάνουν ένα κομμάτι από το firewall κλπ.

Το IKE εκτελείται σε δύο φάσεις. Οι φάσεις αυτές είναι ίδιες, όπως και στο ISAKMP. Στην πρώτη φάση εδραιώνεται ένας IKE SA για να προστατέψει τη δεύτερη φάση, ενώ στη δεύτερη φάση εξάγονται τα κλειδιά και συμφωνείται μια κοινή πολιτική για συσχετισμούς ασφαλείας που δεν ανήκουν στο IKE, όπως για παράδειγμα στο TLS.

Υπάρχουν δύο τρόποι για να εδραιωθεί ένας IKE SA στην πρώτη φάση. Ο κύριος τρόπος (main mode) και ο επιθετικός τρόπος (aggressive mode). Ο επιθετικός τρόπος είναι λίγο πιο γρήγορος, αλλά δεν παρέχει προστασία ταυτότητας, ενώ ο κύριος τρόπος βασίζεται στην ανταλλαγή με αυθεντικοποίηση του ISAKMP και παρέχει προστασία ταυτότητας. Στη δεύτερη φάση ο γρήγορος τρόπος χρησιμοποιείται για να εδραιώσει τους συσχετισμούς ασφαλείας των πελατών του IKE.

Υπάρχει, επίσης, ο τρόπος νέας ομάδας (new group mode), ο οποίος δεν είναι ούτε στην πρώτη, ούτε στη δεύτερη φάση, αλλά χρησιμοποιείται μόνο για να εδραιώσει μια καινούργια ομάδα Oakley (Oakley group) για τις ανταλλαγές Diffie-Hellman. Οι ομάδες αυτές είναι πέντε προκαθορισμένες και επειδή προέρχονται από το πρωτόκολλο Oakley, ονομάζονται και ομάδες Oakley. Τρεις από τις ομάδες είναι κλασσικές εκθετικές ομάδες υπολοίπων (modular

exponentiation groups) και δύο είναι ομάδες ελλειπτικών καμπυλών (elliptic curve groups).

Οι παρακάτω συμβολισμοί χρησιμοποιούνται για την επεξήγηση των τρόπων λειτουργίας :

HDR : Το HDR είναι μια επικεφαλίδα ISAKMP, της οποίας ο τύπος ανταλλαγής είναι ο τρόπος λειτουργίας του IKE.

HDR* : Όλα τα δεδομένα μετά την επικεφαλίδα ISAKMP κρυπτογραφημένα

SA : Συσχετισμός ασφαλείας σε μια σειρά δεδομένων εδραίωσης SA με ένα ή περισσότερα πλαίσια δεδομένων πρότασης και μετασχηματισμού (Proposal and Transform payloads)

Nonce : Πλαίσιο δεδομένων που περιέχουν ένα μοναδικό αριθμό

KE : Πλαίσιο δεδομένων ανταλλαγής κλειδιού

IDii : Πλαίσιο δεδομένων ταυτότητας ; Εκκινητής (initiator) φάσης 1

IDci : Πλαίσιο δεδομένων ταυτότητας ; Εκκινητής φάσης 2

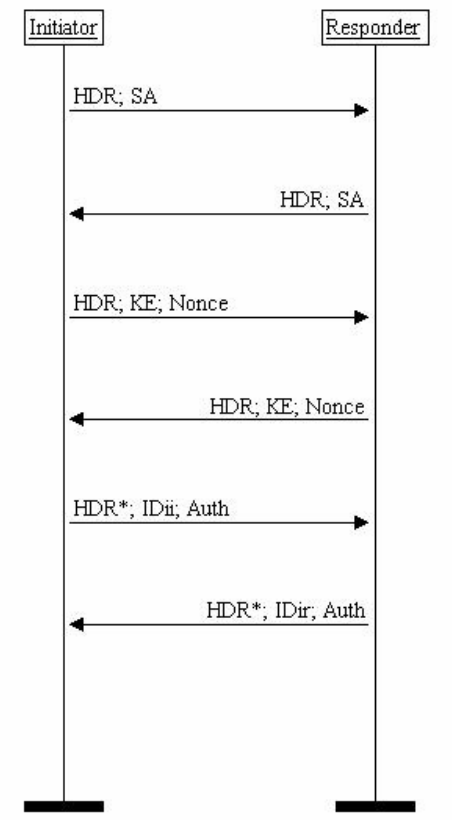
IDir : Πλαίσιο δεδομένων ταυτότητας ; Απαντών (responder) φάσης 1

IDcr : Πλαίσιο δεδομένων ταυτότητας ; Απαντών φάσης 2

Auth : Ένας γενικός μηχανισμός αυθεντικοποίησης, όπως μια συνάρτηση κατακερματισμού ή υπογραφής.

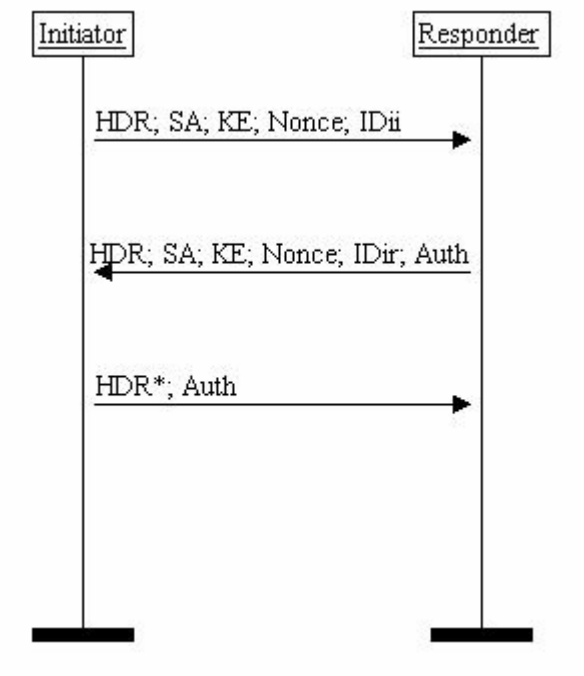
Hash : Πλαίσιο δεδομένων συνάρτησης κατακερματισμού.

Ο κύριος τρόπος είναι μια ανταλλαγή στην πρώτη φάση του IKE/ISAKMP. Τα πρώτα δύο μηνύματα χρησιμοποιούνται για την εδραίωση της πολιτικής ασφαλείας για την ανταλλαγή (σχήμα 5). Τα επόμενα δύο μηνύματα χρησιμοποιούνται για την ανταλλαγή των δεδομένων για τη δημιουργία των κλειδιών Diffie-Hellman. Τα τελευταία δύο μηνύματα είναι για να αυθεντικοποιήσουν τα μέρη με συναρτήσεις κατακερματισμού, υπογραφές ή κατ' επιλογή με πιστοποιητικά. Τα τελευταία δύο μηνύματα είναι κρυπτογραφημένα με το προηγουμένως συμφωνημένο κλειδί και οι ταυτότητες των μερών προστατεύονται από εισβολείς.



Σχήμα 5. Κύριος Τρόπος (main mode)

Ο επιθετικός τρόπος είναι μια ανταλλαγή στην πρώτη φάση του IKE/ISAKMP. Μοιάζει με τον κύριο τρόπο, εκτός από το γεγονός ότι μερικά μηνύματα είναι ενσωματωμένα σε άλλα (σχήμα 6). Το πρώτο μήνυμα προτείνει την πολιτική και περνάει τα δεδομένα για την ανταλλαγή του κλειδιού, έναν μοναδικό αριθμό (nonce) και μερικές πληροφορίες για πιστοποίηση χρήστη. Το δεύτερο μήνυμα είναι μια απάντηση η οποία πιστοποιεί τον απαντών (responder) και περιέχει την πολιτική και την ανταλλαγή κλειδιού. Σε αυτό το σημείο, όλη η πληροφορία για το κλειδί κρυπτογράφησης του ISAKMP SA έχει ανταλλαχθεί. Το μήνυμα μπορούσε να κρυπτογραφηθεί, χωρίς αυτό να είναι υποχρεωτικό. Το τελευταίο μήνυμα χρησιμοποιείται για να αυθεντικοποιήσει τον εκκινητή και παρέχει μια απόδειξη συμμετοχής στην ανταλλαγή. Η ταυτότητα του απαντώντος δεν μπόρεσε να προστατευθεί, αλλά κρυπτογραφώντας το τελευταίο μήνυμα, η ταυτότητα του εκκινητή προστατεύεται.



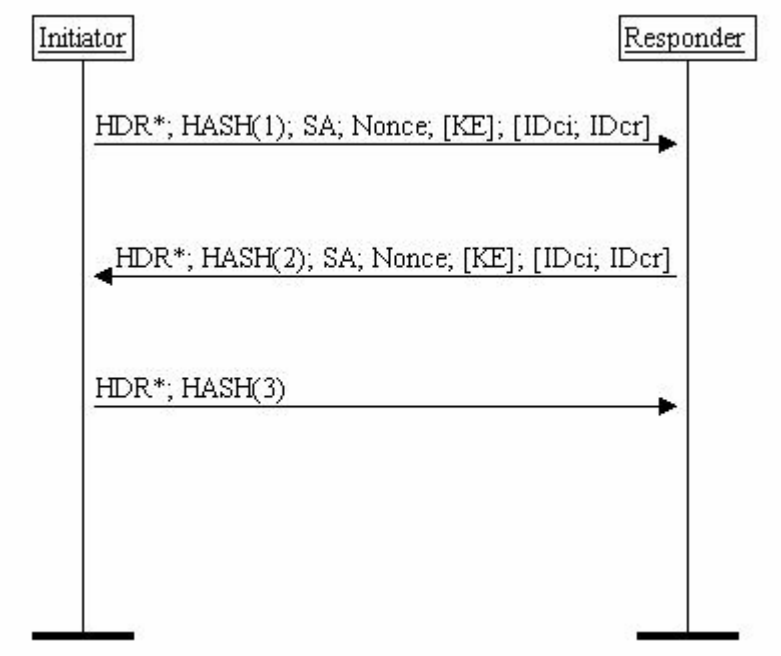
Σχήμα 6. Επιθετικός Τρόπος (Aggressive mode)

Ο γρήγορος τρόπος χρησιμοποιείται για την ανταλλαγή στη δεύτερη φάση του IKE (σχήμα 7). Ένας συσχετισμός ασφαλείας IKE εδραιώνεται στην πρώτη φάση για να προστατέψει την ανταλλαγή της δεύτερης φάσης με τον κύριο ή επιθετικό τρόπο που περιγράφηκε πριν. Ο γρήγορος τρόπος χρησιμοποιείται για να εδραιωθεί ένας συσχετισμός ασφαλείας και να δημιουργηθούν νέα κλειδιά. Όλα τα πλαίσια δεδομένων, εκτός από την επικεφαλίδα ISAKMP, κρυπτογραφούνται. Για να επιτευχθεί τέλεια μυστικότητα προς τα εμπρός (perfect forward secrecy), γίνεται μια ανταλλαγή κλειδιού Diffie-Hellman. Η τέλεια μυστικότητα προς τα εμπρός αναφέρεται στην ιδέα ότι η παραβίαση ενός κλειδιού θα επιτρέψει την πρόσβαση στα δεδομένα που προστατεύονται από ένα μοναδικό κλειδί. Για να υπάρχει τέλεια μυστικότητα προς τα εμπρός, το κλειδί που χρησιμοποιείται για τη μετάδοση δεδομένων δεν πρέπει να χρησιμοποιηθεί για να παραχθούν επιπλέον κλειδιά. Επιπρόσθετα, αν το κλειδί που χρησιμοποιείται για την προστασία μετάδοσης δεδομένων παράχθηκε από κάποια άλλα κλειδιά, τότε αυτά δεν πρέπει να χρησιμοποιηθούν για τη δημιουργία επιπλέον κλειδιών. Για να παρέχουν τέλεια μυστικότητα προς τα εμπρός, και τα δύο μέρη πρέπει:

- να χρησιμοποιούν τον κύριο τρόπο για να προστατεύσουν τις ταυτότητες, όταν εδραιώνουν ένα συσχετισμό ασφαλείας ISAKMP.

- να χρησιμοποιούν τον κύριο τρόπο για να διαπραγματευτούν τον συσχετισμό ασφαλείας.
- να διαγράψουν τους συσχετισμούς ασφαλείας ISAKMP , έπειτα από κάθε ζευγάρι ανταλλαγής γρήγορου τρόπου, ώστε να επιβάλλουν την δημιουργία καινούργιων συσχετισμών ασφαλείας ISAKMP.

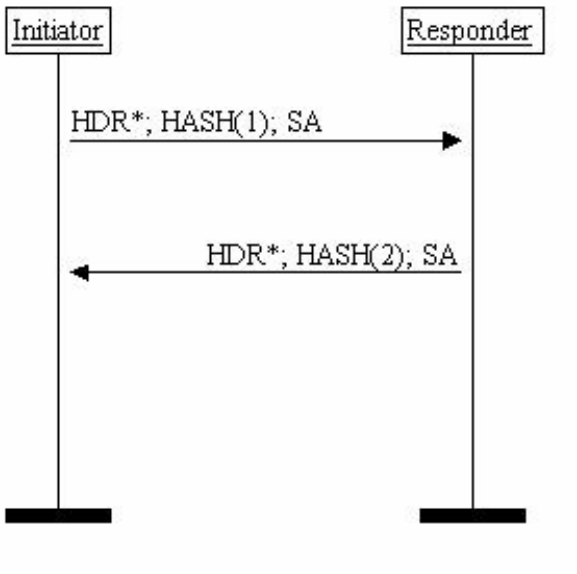
Μόλις διαγραφούν οι συσχετισμοί ασφαλείας, θα δημιουργηθεί ένα νέο κλειδί Diffie-Hellman από νέα κλειδιά και οι συσχετισμοί με τα παλιά κλειδιά χάνονται, διατηρώντας έτσι την τέλεια μυστικότητα προς τα εμπρός.



Σχήμα 7. Γρήγορος Τρόπος (quick mode)

Κατά τη διάρκεια της ανταλλαγής με γρήγορο τρόπο μπορούν να εδραιωθούν πολλά κλειδιά. Καθένα από τα μέρη μπορεί να ξεκινήσει την ανταλλαγή, άσχετα με το ποιο μέρος ξεκίνησε την πρώτη φάση.

Ο τρόπος νέας ομάδας χρησιμοποιείται για να συμφωνηθεί μια νέα ομάδα (εκθετική ομάδα υπολοίπων ή ελλειπτική καμπύλη), πάνω στην οποία βασίζεται η ανταλλαγή Diffie-Hellman (σχήμα 8). Παρόλο που ο τρόπος αυτός δεν είναι ανταλλαγή της δεύτερης φάσης, θα πρέπει πάντα να ακολουθεί την ανταλλαγή της πρώτης φάσης.



Σχήμα 8. Τρόπος νέας ομάδας (new group mode)

Αρχικά, το IKE είχε ένα μεγάλο αριθμό από επιλογές ρύθμισης, αλλά όχι ένα μηχανισμό αυτόματης διαπραγμάτευσης με τις πιο γνωστές και από όλους υποστηριζόμενες ρυθμίσεις. Ως συνέπεια αυτού, και οι δύο πλευρές του IKE πρέπει να συμφωνήσουν στον τύπο του συσχετισμού ασφαλείας που θέλουν να δημιουργήσουν, βήμα προς βήμα, ειδάλλως δεν μπορεί να επιτευχθεί σύνδεση. Επιπλέον, επιπλοκές δημιουργούσε το γεγονός ότι σε πολλές υλοποιήσεις το αποτέλεσμα της ανίχνευσης σφαλμάτων δεν ήταν εύκολο να ερμηνευτεί, ενώ σε άλλες δεν υπήρχε καν ρουτίνα ανίχνευσης σφαλμάτων.

Τα χαρακτηριστικά της πρώτης έκδοσης του IKE μπορούσαν να ερμηνευτούν με διαφορετικό τρόπο, περιοριζόμενα σε σφάλματα σχεδιασμού, με αποτέλεσμα διαφορετικές υλοποιήσεις να μην είναι σε θέση να δημιουργήσουν ένα συσχετισμό ασφαλείας SA σε πολλούς συνδυασμούς ρυθμίσεων, παρόλο που και στα δύο μέρη φαίνονται σωστές οι ρυθμίσεις. Το πρόβλημα αυτό ξεπεράστηκε στη δεύτερη έκδοση του πρωτοκόλλου.

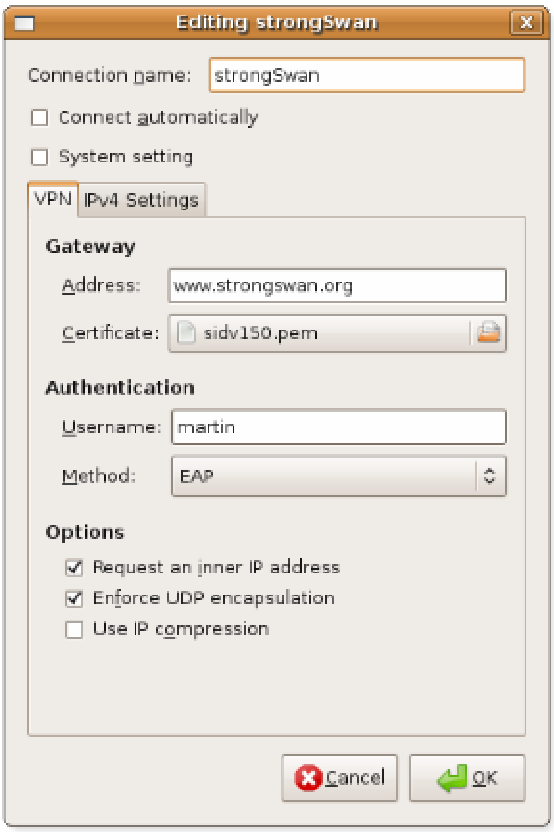
Με τη δεύτερη έκδοση έγιναν βελτιώσεις στο πρότυπο. Μερικές από τις βελτιώσεις είναι οι παρακάτω:

- Το πρότυπο περιγράφηκε σε λιγότερα κείμενα RFC.
- Προστέθηκε υποστήριξη φορητότητας και σε χρήστες με πολλαπλά σημεία πρόσβασης (multihomed users).

- Η ενθυλάκωση του IKE και του ESP στο port 4500 του UDP, επιτρέπει σε αυτά τα πρωτόκολλα να διασχίσουν συσκευές ή firewalls που τρέχουν NAT.
- Υποστηρίχθηκε το πρωτόκολλο SCTP, όπως χρησιμοποιείται στη δικτυακή τηλεφωνία VoIP.
- Παρέχει ένα μηχανισμό τεσσάρων μηνυμάτων αρχικής ανταλλαγής, ενώ η πρώτη έκδοση IKE παρείχε 8 διαφορετικούς διακριτούς μηχανισμούς αρχικής ανταλλαγής, ο καθένας από τους οποίους είχε πλεονεκτήματα και μειονεκτήματα.
- Το IKE v2 χρησιμοποιεί κρυπτογραφικούς μηχανισμούς για να προστατέψει τα πακέτα του, που είναι παρόμοιοι με αυτούς του IPsec ESP (Encapsulating Security Payload). Αυτό οδηγεί σε απλούστερες υλοποιήσεις και πιστοποιήσεις για Common Criteria και FIPS 140-2, τα οποία απαιτούν κάθε κρυπτογραφική υλοποίηση να επαληθεύεται χωριστά.
- Η δεύτερη έκδοση του IKE χρησιμοποιεί διαδοχικούς αριθμούς και αποδείξεις παραλαβής για να παρέχει αξιοπιστία και εκτελεί επεξεργασία σφαλμάτων και διαχείριση κοινών καταστάσεων. Η αρχική έκδοση του IKE μπορεί να καταλήξει σε αδιέξοδη κατάσταση (dead state) λόγω της έλλειψης τέτοιων μέτρων αξιοπιστίας, όπου και τα δύο μέρη περίμεναν το άλλο να ξεκινήσει μια λειτουργία, κάτι που ποτέ δεν συνέβαινε. Εναλλακτικές, όπως η ανίχνευση μη ενεργού μέρους (Dead-Peer-Detection), αναπτύχθηκαν αλλά δεν έχουν προτυποποιηθεί. Αυτό σημαίνει ότι διαφορετικές υλοποιήσεις δεν ήταν πάντα συμβατές.
- Το IKEv2 παρουσιάζει προσαρμοστικότητα στις επιθέσεις Denial of Service (DoS), καθώς δεν πραγματοποιεί μεγάλη επεξεργασία προκειμένου να διαπιστώσει την ύπαρξη ή μη του μέρους που κάνει αίτηση για σύνδεση. Το γεγονός αυτό βελτιώνει μερικά από τα προβλήματα της πρώτης έκδοσης, η οποία εκτελούσε μεγάλη κρυπτογραφική επεξεργασία από τοποθεσίες που δεν ήταν έγκυρες. Ας υποθέσουμε ότι ο ξενιστής HostA έχει δείκτη παραμέτρου ασφαλείας (SPI) A και ο ξενιστής HostB έχει SPI B. Εάν ο ξενιστής HostB αντιμετωπίζει μεγάλο αριθμό από ημιτελείς συνδέσεις IKE init, θα απαντήσει με μια μη κρυπτογραφημένη προειδοποίηση ike_sa_init, σε ένα μήνυμα τύπου cookie. Μετά την αποστολή, ο HostB θα περιμένει να λάβει ένα μήνυμα ike_sa_init με περιεχόμενο την τιμή του cookie. Αυτή η διαδικασία γίνεται για να επιβεβαιωθεί ότι το μέρος που ξεκινάει τη διαδικασία σύνδεσης είναι σε θέση να απαντήσει στα μηνύματα του άλλου μέρους.

Το IKE υποστηρίζεται ως μέρος της υλοποίησης του IPsec στα Windows 2000, Windows XP, Windows Server 2003, Windows Vista και Windows Server 2008. Η υλοποίηση ISAKMP/IKE αναπτύχθηκε σε συνεργασία από την Cisco και τη Microsoft. Τα Microsoft Windows 7 και Windows Server 2008 R2 υποστηρίζουν εξολοκλήρου το IKE-v2 (RFC4306), καθώς και το MOBIKE (RFC 4555) μέσω της εφαρμογής VPN Reconnect, γνωστής και ως Agile VPN.

Επίσης, υπάρχουν αρκετές υλοποιήσεις ανοικτού κώδικα του IPSec με τις σχετικές δυνατότητες του IKE. Στο Linux οι υλοποιήσεις Openswan και strongSwan παρέχουν ένα πρόγραμμα daemon που λέγεται Pluto, το οποίο μπορεί να εδραιώσει συσχετισμό ασφαλείας (Security Associations) με το KLIPS (Kernel Level IP Security) ή NETKEY που βασίζονται στον πυρήνα του Linux. Το NETKEY είναι η υλοποίηση του IPsec της έκδοσης 2.6 του πυρήνα. Στην παρακάτω εικόνα φαίνεται η καρτέλα ρυθμίσεων δικτύου VPN με το λογισμικό Network Manager με StrongSwan NM plug-in. Το plug-in αυτό χρησιμοποιεί πιστοποιητικό για την αυθεντικοποίηση της πύλης ασφαλείας και υποστηρίζει αυθεντικοποίηση με EAP και RSA για τον client [28].



Σχήμα 9. Ρυθμίσεις δικτύου VPN με την υλοποίηση StrongSwan.

Στο σχήμα 10 φαίνεται το αρχείο ρυθμίσεων στην υλοποίηση Openswan του IPsec. Η σύνδεση έχει όνομα tunnelipsec και είναι τύπου ενθυλάκωσης με το πρωτόκολλο ESP. Η γραμμή interfaces λέει στο IPsec να χρησιμοποιήσει την ίδια διεύθυνση IP που είναι προεπιλεγμένη στην κάρτα δικτύου του υπολογιστή. Οι γραμμές left έως rightsubnet ορίζουν την τοπολογία του δικτύου και τις δύο προεπιλεγμένες πύλες. Η γραμμή esp ρυθμίζει τις παραμέτρους του αντίστοιχου πρωτοκόλλου. Οι επιλογές της μεθόδου κρυπτογράφησης και αυθεντικοποίησης θα πρέπει να είναι οι ίδιες και στις δύο πύλες ασφαλείας, αλλιώς δεν θα δουλέψει η σύνδεση. Η γραμμή keyexchange επιλέγει το IKE ως το πρωτόκολλο για την εδραίωση του κλειδιού.

Οι διανομές του λογισμικού Berkeley έχουν, επίσης, υλοποιήσεις IPsec και πρόγραμμα IKE daemon και το πιο σημαντικό είναι ένα κρυπτογραφικό πλαίσιο, το OCF (OpenBSD Cryptographic Framework), το οποίο κάνει εύκολη την υποστήριξη επιταχυντών κρυπτογράφησης. Το OCF έχει, επίσης, μεταφερθεί πρόσφατα και στο Linux.

Αρκετοί κατασκευαστές δικτυακών προϊόντων έχουν αναπτύξει τα δικά τους προγράμματα IKE daemon ή πήραν τα δικαιώματα από άλλον κατασκευαστή. Επίσης, έχουν δημιουργηθεί ομάδες εργασίας (work-shops) σε εργαστήρια, όπως το ICSA LABS, τα οποία εξετάζουν με εξειδικευμένα τεστ τη διαλειτουργικότητα (interoperability) των υλοποιήσεων του IKE v2 από διάφορους κατασκευαστές.

```
# /etc/ipsec.conf - Openswan IPsec configuration file

 config setup
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none

 conn tunnelipsec
        type=           tunnel
        authby=         secret
        left=           202.0.45.170
        leftnexthop=    202.0.45.190
        leftsubnet=     10.69.1.0/24
        right=          203.97.9.162
        rightnexthop=   203.97.9.161
        rightsubnet=    10.7.3.0/24
        esp=            3des-md5-96
        keyexchange=    ike
        pfs=            no
        auto=           start
```

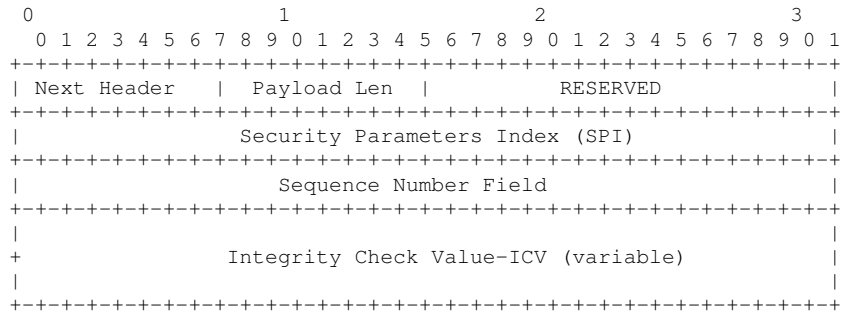
Σχήμα 10. Αρχείο ρυθμίσεων της υλοποίησης OpenSwan του IPsec.

2.4. Το πρωτόκολλο Authentication Header (AH)

Το Authentication Header είναι ένα πρωτόκολλο που χρησιμοποιείται ως τμήμα της ευρύτερου συνόλου πρωτοκόλλων IPsec. Ο στόχος του είναι να διασφαλίσει απρόσκοπτη αυθεντικοποίηση της προέλευσης των πακέτων, καθώς και της μη αλλοίωσης του περιεχομένου τους. Επιπλέον, μπορεί κατά επιλογή του χρήστη, να προστατεύει από επιθέσεις επανάληψης αποστολής πακέτων (replay attacks), χρησιμοποιώντας την τεχνική του ολισθαίνοντος παραθύρου (sliding window), απορρίπτοντας τα παλιά πακέτα που δεν έχουν σταλεί εντός του προκαθορισμένου περιθωρίου.

Στην έκδοση 4 του πρωτοκόλλου IP το AH προστατεύει τα δεδομένα που μεταφέρει το IP συμπεριλαμβανομένων και των επικεφαλίδων (headers), με εξαίρεση τα πεδία εκείνα που μπορεί να τροποποιηθούν κατά τη μεταφορά (για παράδειγμα τα DSCP/TOS, ECN, Flags, Fragment Offset, TTL και Header Checksum). Στην έκδοση 6 του πρωτοκόλλου IP, το AH προστατεύει το ίδιο το πλαίσιο AH, την επικεφαλίδα Destination Options μετά το AH, και το πλαίσιο (payload) IP. Επίσης, προστατεύει την επικεφαλίδα IP-v6 και όλες τις εξωτερικά προσαρτημένες επικεφαλίδες πριν το AH, εκτός αυτών που μεταβάλλονται κατά τη μεταφορά του πακέτου, δηλαδή των DSCP, ECN, Flow Label και Hop Limit. Το AH λειτουργεί ακριβώς στο πάνω επίπεδο του IP και χρησιμοποιεί την τιμή 51 για το πεδίο IP protocol.

Στο σχήμα 11, φαίνεται πως δομείται και ερμηνεύεται κάθε πεδίο ενός πακέτου AH. Το σχήμα αυτό προέρχεται από το επίσημο κείμενο RFC 4302 που επεξηγεί το πρωτόκολλο.



Σχήμα 11. Η διάταξη του πλαισίου Authentication Header

Το πεδίο των 8 bit Next Header που φαίνεται στο σχήμα, δείχνει ποιό ανώτερο πρωτόκολλο προστατεύεται. Αυτό παίρνει τιμές από τη λίστα τιμών του IP protocol. Το Payload Len (8 bits) δηλώνει το μήκος του πακέτου AH σε μονάδες των 4 οκτάδων (4-octet) μείον δύο (για παράδειγμα εάν έχει τιμή μηδέν αυτό σημαίνει (0+2)*4=8 οκτάδες, ενώ η τιμή 1 σημαίνει κατά αντιστοιχία 12 οκτάδες). Παρόλο που το μέγεθος μετράται σε μονάδες των 4 οκτάδων, το μήκος της επικεφαλίδας (AH header) πρέπει να είναι πολλαπλάσιο των 8 οκτάδων εάν μεταφέρεται με ένα πακέτο IPv-6. Αυτός ο περιορισμός δεν υπάρχει για πακέτα IPv-4.

Το πεδίο reserved δηλώνει απλά ότι είναι δεσμευμένο για μελλοντική χρήση και μέχρι τότε περιέχει μόνο μηδενικά bit. Το πεδίο Security Parameters Index περιέχει μια καθορισμένη τιμή, η οποία χρησιμοποιείται μαζί με τη διεύθυνση IP του αποστολέα για να προσδιορίσει το συσχετισμό ασφαλείας του αποστολέα.

Το πεδίο Sequence Number το οποίο αυξάνεται κατά 1 για κάθε πακέτο που αποστέλλεται, χρησιμοποιείται για να αποτρέψει επιθέσεις επανάληψης αποστολής. Όταν είναι ενεργοποιημένη η ανίχνευση επανάληψης, η τιμή του Sequence Number δεν ξαναχρησιμοποιείται διότι ένας καινούργιος συσχετισμός ασφαλείας πρέπει να εδραιωθεί πριν από την προσπάθεια αύξησης του Sequence Number πέρα από τη μέγιστη τιμή.

Υπάρχουν σημαντικές αλλαγές στον τρόπο με τον οποίο αντιμετωπίζονται οι απαριθμητές (sequence numbers) στα κείμενα RFC 2402 και RFC 4302. Οι απαριθμητές που χρησιμοποιούνται στο RFC 2402 έχουν 32 bits μήκος μόνο, έτσι ολόκληρος ο απαριθμητής αποστέλλεται στην επικεφαλίδα AH. Η υπέρβαση του απαριθμητή δεν επιτρέπεται και οι προσπάθειες για αποστολή τέτοιων πακέτων καταγράφονται, εάν έχει επιλεγεί η προστασία από επιθέσεις τύπου επανάληψης αποστολής. Η τυπική ενέργεια του αποστολέα σε περίπτωση αναμενόμενης υπέρβασης του απαριθμητή, είναι να εδραιώνει καινούργιο συσχετισμό ασφαλείας SA, πριν ο απαριθμητής υπερχειλίσει.

Από την άλλη μεριά, το κείμενο RFC 4302 επιτρέπει προαιρετικά τη χρήση εκτεταμένου απαριθμητή ESN (Extended Sequence Number). Αυτό είναι χρήσιμο σε δίκτυα υψηλής ταχύτητας, όπου ο απαριθμητής των 32 bit μπορεί να υπερχειλίσει εύκολα. Τα ESNs έχουν μήκος 64 bit και όλα τα bit χρησιμοποιούνται για τον υπολογισμό του MAC, παρόλο που μόνο τα 32 λιγότερο σημαντικά bits του ESN μεταφέρονται στο πεδίο Sequence Number. Για τον υπολογισμό του MAC τα πιο σημαντικά bits τοποθετούνται μετά το payload. Αυτό σημαίνει ότι το ESN διαχωρίζεται σε δύο μέρη, παρά σε ένα τμήμα 64 συνεχόμενων bits ως είσοδος στο MAC. Αυτό είναι κατά κάποιο τρόπο ασυνήθιστο, αλλά γίνεται για να μην αλλάξει η μορφή του πλαισίου AH, όπως αυτό που ορίζεται στο σχετικό κείμενο RFC 4302, όταν χρησιμοποιούνται απαριθμητές SN των 32 bits. Με τον τρόπο αυτό, επιτυγχάνεται συμβατότητα με το προγενέστερο πρότυπο. Η αποστολή μόνο του μισού πεδίου ESN στα πλαίσια AH οδηγεί στην ανάγκη ενός μηχανισμού συγχρονισμού σε περίπτωση που περισσότερα από 2^{32} συνεχόμενα πακέτα χαθούν. Το κείμενο RFC 4302 προτείνει τη χρήση ESNs αντί για απαριθμητές των 32 bits, ενώ το RFC 4304 εξηγεί πώς το IKE μπορεί να τροποποιηθεί για να υποστηρίζει ESNs.

Τέλος, το πεδίο Integrity Check Value έχει μεταβαλλόμενο μήκος, ακέραιο πολλαπλάσιο των 32 bit (σε IPv-4) ή των 64 bit (σε IPv-6). Μπορεί να περιέχει bit γεμίσματος (padding) για να φτάσει το πλαίσιο σε πολλαπλάσιο των 8 οκτάδων για το IP-v6 ή των 4 οκτάδων για το IP-v4. Ο υπολογισμός της τιμής Integrity Check γίνεται λαμβάνοντας υπόψη:

- Τα πεδία IP ή επέκτασης επικεφαλίδας πριν την επικεφαλίδα AH, που δεν αλλάζουν κατά την μεταφορά του πακέτου ή η τιμή τους είναι προβλέψιμη

για τον συσχετισμό ασφαλείας, κατά την παραλαβή του πακέτου από το άλλο μέρος.

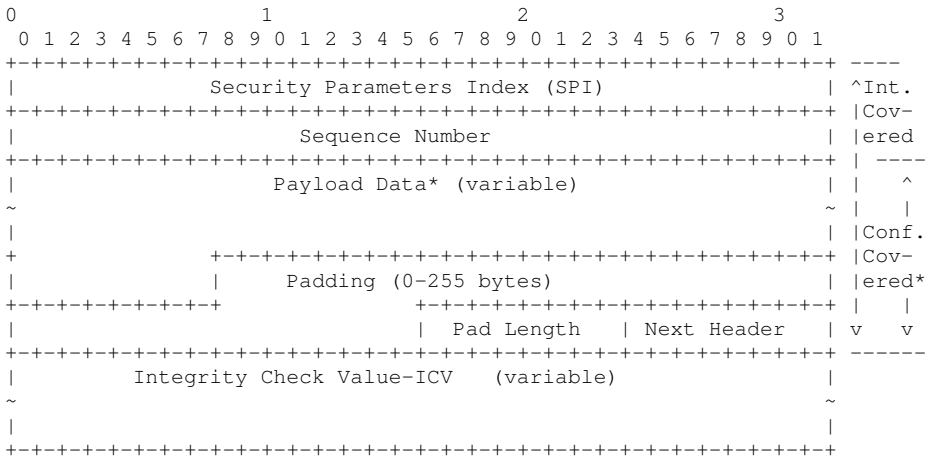
- Την επικεφαλίδα AH (Next Header, Payload Len, Reserved, SPI, Sequence Number (32bits), το ICV (το οποίο τίθεται σε μηδέν για τον υπολογισμό) και τα διακριτά bytes γεμίσματος.
- Όλα όσα στέλνονται μετά το AH και θεωρούνται ότι δεν αλλάζουν κατά τη μεταφορά του πακέτου.
- Τα υψηλής τάξης bits του ESN (εάν χρησιμοποιούνται) και ότι ειδικά bit γεμίσματος απαιτούνται από τον αλγόριθμο ακεραιότητας δεδομένων.

Εάν κάποιο πεδίο αλλάξει κατά τη μεταφορά, η τιμή του τίθεται σε μηδέν για τον υπολογισμό του ICV. Εάν η τιμή αλλάζει, αλλά μπορεί να είναι προβλέψιμη στον δέκτη του πακέτου, τότε αυτή η τιμή λαμβάνεται υπόψη για τον υπολογισμό του ICV. Η προσέγγιση της χρήσης του μηδέν και όχι η παράβλεψη των τιμών που αλλάζουν, επιτρέπει τη διατήρηση ενός σταθερού συνολικού μήκους bits που χρησιμοποιούνται για τον υπολογισμό του ICV.

2.5. Το πρωτόκολλο Encapsulating Security Payload (ESP)

Το ESP ως τμήμα του ευρύτερου πρωτοκόλλου IPsec, παρέχει προστασία αυθεντικοποίησης προέλευσης (origin authenticity), ακεραιότητας (integrity) και εμπιστευτικότητας (confidentiality) για τα πακέτα που διακινούνται στο δίκτυο. Το ESP, επίσης, υποστηρίζει και λειτουργία μόνο-κρυπτογράφησης (encryption-only), καθώς και μόνο-αυθεντικοποίησης (authentication-only), αλλά η χρήση της κρυπτογράφησης χωρίς αυθεντικοποίηση δεν συνίσταται, καθόσον είναι ανασφαλής. Σε αντίθεση με το AH το ESP, δεν προστατεύει την επικεφαλίδα IP του πακέτου. Ωστόσο, σε λειτουργία ενθυλάκωσης, όπου ολόκληρο το πακέτο IP ενσωματώνεται στο πλαίσιο ESP, η προστασία του ESP εφαρμόζεται σε όλο το εσωτερικό πακέτο IP (συμπεριλαμβανομένης της επικεφαλίδας), ενώ η εξωτερική επικεφαλίδα παραμένει χωρίς προστασία. Το ESP λειτουργεί ένα επίπεδο πάνω ακριβώς από το IP, χρησιμοποιώντας την τιμή 50 για το πεδίο IP protocol.

Στο παρακάτω σχήμα 4, φαίνεται πώς δομείται και ερμηνεύεται κάθε πεδίο ενός πακέτου ESP. Το σχήμα αυτό προέρχεται από το επίσημο κείμενο RFC 4303 που επεξηγεί το πρωτόκολλο.



Σχήμα 12. Η μορφή του πακέτου Encapsulating Security Payload

Το 32 bit πεδίο Security Parameters Index περιέχει μια καθορισμένη τιμή, η οποία χρησιμοποιείται μαζί με τη διεύθυνση αποστολέα (source IP address) για να προσδιορίσει τον συσχετισμό ασφαλείας του αποστολέα. Το πεδίο Sequence Number των 32 bit αυξάνει κατά 1 για κάθε απεσταλμένο πακέτο, για την προστασία από επιθέσεις τύπου επανάληψης αποστολής. Για κάθε συσχετισμό ασφαλείας κρατείται και χωριστός απαριθμητής.

Το πεδίο μεταβλητού μήκους Payload data περιέχει τα προστατευμένα δεδομένα του αρχικού πακέτου IP, συμπεριλαμβανομένων των παραμέτρων που χρησιμοποιούνται για την προστασία των δεδομένων (π.χ. το διάνυσμα αρχικοποίησης του κρυπτογραφικού αλγορίθμου). Ο τύπος του περιεχομένου που προστατεύεται δείχνεται από το πεδίο Next Header. Το πεδίο padding με μήκος από 0 έως 255 οκτάδες, περιέχει bit γεμίσματος (padding) για την κρυπτογράφηση, ώστε να επιμηκύνει το payload data σε ένα μέγεθος που να ταιριάζει με το μπλοκ του κρυπταλγορίθμου και να ευθυγραμμίσει το επόμενο πεδίο. Το πεδίο pad length έχει μήκος 8 bit και περιέχει το επιλεγμένο μήκος γεμίσματος του πακέτου σε οκτάδες bit. Το Next Header περιέχει τον τύπο του πρωτοκόλλου του πακέτου που προστατεύεται και παίρνει τιμή από τις προκαθορισμένες τιμές του πρωτοκόλλου IP.

Τέλος, το πεδίο Integrity Check Value έχει μεταβαλλόμενο μήκος και πολλαπλάσιο των 32 bit. Μπορεί να περιέχει bit γεμίσματος (padding) για να καταστήσει το πλαίσιο πολλαπλάσιο του ορίου των 8 οκτάδων για το IP-v6 ή των 4 οκτάδων για το IP-v4.

2.6. Ασφάλεια και γνωστές επιθέσεις σε υλοποιήσεις του IPsec

Οι διάφορες γενιές των κειμένων RFC πάνω στο IPsec δείχνουν μια εξέλιξη του πρωτοκόλλου και όχι εισαγωγή θεμελιωδών αλλαγών. Έχουν προστεθεί νέες δυνατότητες, όπως μεγαλύτερα κρυπτογραφικά κλειδιά και νέοι αλγόριθμοι συνδυασμένης λειτουργίας στο ESP. Η συνολική ευκρίνεια και ποιότητα των προτύπων έχει αναμφισβήτητα βελτιωθεί, πιθανώς ως συνέπεια της συγκέντρωσης ειδικών στον χώρο αυτό, καθώς και της κατανόησης της ανάγκης επίτευξης διαλειτουργικότητας μέσω της ευκρινέστερης προτυποποίησης.

Ωστόσο, η ανάπτυξη των κειμένων RFC περιορίζεται από την ανάγκη διατήρησης συμβατότητας των παλαιότερων συστημάτων. Αυτό είναι φανερό από το γεγονός ότι το ESP επιτρέπει, ακόμη, τη χρήση ρυθμίσεων μόνον κρυπτογράφησης και δεν υποχρεώνει την υποστήριξη των αλγορίθμων συνδυασμένης λειτουργίας (combined mode algorithms) που αναφέρθηκαν στο κεφάλαιο 2.1. Αυτό φαίνεται σαν χαμένη ευκαιρία, καθόσον αυτοί οι αλγόριθμοι είναι διαθέσιμοι και πιο ασφαλείς [03]. Η εξέλιξη του IKE έχει πετύχει περισσότερο. Η δεύτερη έκδοσή του είναι πιο απλή και συμπαγής και η περιγραφή του πιο εξειδικευμένη. Αλλά το γεγονός ότι υποστηρίζει το πρωτόκολλο EAP (Extensible Authentication Protocol), εισάγει πολυπλοκότητα ως αναπόφευκτη συνέπεια της επιθυμίας για ευελιξία [03].

Το IPsec έχει αναλυθεί τα τελευταία χρόνια από την κρυπτογραφική σκοπιά. Η φιλοσοφία της αποδείξιμης ασφάλειας έχει επηρεάσει τα κείμενα RFC του IPsec. Σαν παράδειγμα το HMAC, το οποίο έχει αποδειχθεί ως κρυπτογραφικά ισχυρό [03], χρησιμοποιείται σε μεγάλο βαθμό στο IPsec, ενώ στο κείμενο RFC 4303 παρουσιάζονται οι αποδείξεις των θεμάτων ασφαλείας της εφαρμογής κώδικα αυθεντικοποίησης στα δεδομένα πριν από την κρυπτογράφησή τους. Επιπρόσθετα, το πρωτόκολλο SIGMA (SIGn-and-Mac)

[16] είναι η βάση σχεδιασμού της λειτουργίας αυθεντικοποίησης με υπογραφή της δεύτερης έκδοσης του IKE. Ωστόσο, λίγα από τα συνολικά χαρακτηριστικά σχεδίασης στα κείμενα RFC, φαίνεται να έχουν επιλεχθεί με γνώμονα τη στέρεη θεωρητική βάση τους, όπως έγινε στα παραδείγματα που αναφέρθηκαν πριν. Έτσι, το IPsec προσφέρει πολλές ενδιαφέρουσες προκλήσεις για έρευνα στους θεωρητικούς που αναζητούν κίνητρα από τα προβλήματα ασφαλείας του πρωτοκόλλου σε πραγματικές συνθήκες λειτουργίας.

Μια σειρά από επιθέσεις επιλεγμένου κρυπτοκειμένου (chosen ciphertext attacks) σε διάφορες υλοποιήσεις του πρωτοκόλλου, έχουν δημοσιευτεί σε σχετικά άρθρα των Bellovin, Paterson, Yau, Degabriele, και άλλων [04],[05]. Οι επιθέσεις αυτές βασίζονται στα κενά ασφαλείας που παρουσιάζει η κρυπτογράφηση χωρίς τη χρήση αυθεντικοποίησης στο ESP. Το ESP μέχρι και την τρέχουσα έκδοση (v.3) υποστηρίζει τέτοιες ρυθμίσεις ασφαλείας, παρόλο που συνιστά να μη χρησιμοποιούνται από τους εξουσιοδοτημένους χρήστες. Ωστόσο, μπορεί κανείς να βρει υλοποιήσεις που χρησιμοποιούν τέτοιες ρυθμίσεις ως προκαθορισμένες (default), καθώς και οδηγούς ρύθμισης ιδιωτικών Δικτύων (VPNs) που χρησιμοποιούν το IPsec χωρίς αυθεντικοποίηση. Οι επιθέσεις που περιγράφονται στο [04] εφαρμόζονται στην υλοποίηση του IPsec στο Linux, ενώ το σύνολο επιθέσεων [05] εφαρμόζεται στο ίδιο το πρωτόκολλο IPsec, όπως αυτό καθορίζεται στα σχετικά κείμενα RFC.

2.6.1 Επιθέσεις εναντίον της υλοποίησης IPsec του Linux.

Στην εργασία των Kenneth G. Paterson και Arnold K.L. Yau [4], παρουσιάστηκαν επιθέσεις εναντίον της υλοποίησης IPsec του Linux, οι οποίες αναδεικνύουν τα κενά που υπάρχουν ανάμεσα στην κρυπτογραφία, όπως αναλύεται στην θεωρία, όπως ορίζεται στα πρότυπα, όπως υλοποιείται από τους προγραμματιστές και όπως τελικά αξιοποιείται από τους χρήστες. Ένα παράδειγμα, όπως αναφέρθηκε και πριν, είναι οι διαφορές στην οπτική γωνία των θεωρητικών και των χρηστών και πώς αυτό οδήγησε στη χρήση του πρωτοκόλλου ESP με κρυπτογράφηση μόνο (encryption-only) στην πράξη. Ένα άλλο παράδειγμα, είναι πώς οι επιθέσεις αυτές θα έπρεπε κανονικά να είχαν αποφευχθεί από μια συμβατή με τα πρότυπα υλοποίηση του IPsec,

εξαιτίας μιας σειράς φαινομενικά μη κρίσιμων ελέγχων μετά την αποκρυπτογράφηση, οι οποίοι προσδιορίζονται στο πρότυπο του IPsec. Ωστόσο, η ενσωματωμένη υλοποίηση του πρότυπου IPsec στο Linux, δεν πραγματοποιεί αυτούς τους ελέγχους.

Οι επιθέσεις αυτές έχουν αρκετά ελκυστικά χαρακτηριστικά:

- Είναι επιθέσεις τύπου μόνο κρυπτοκειμένου. Έτσι, δεν απαιτούν ιδιαίτερες συνθήκες κάτω από τις οποίες, για παράδειγμα, παράγονται τα κρυπτοκείμενα που ταιριάζουν στα επιλεγμένα απλά κείμενα. Ούτε απαιτούν μεγάλα τμήματα κρυπτοκειμένου για να είναι επιτυχείς. Οι επιθέσεις μπορούν να εφαρμοστούν ακόμα και με ένα απλό πλαίσιο κρυπτογραφημένων δεδομένων.
- Οι επιθέσεις απαιτούν μόνο ο επιτιθέμενος να είναι σε θέση να στείλει μηνύματα στο δίκτυο και να λάβει από αυτό συγκεκριμένες απαντήσεις. Μερικές παραλλαγές μάλιστα επιτρέπουν αυτές τις απαντήσεις να στέλνονται απευθείας στον υπολογιστή του επιτιθέμενου.
- Είναι πολύ αποτελεσματικές. Για παράδειγμα, μια παραλλαγή η οποία έχει υλοποιηθεί, απαιτεί την αποστολή μερικών μόνο πακέτων δεδομένων για να αποκαλύψει όλα τα περιεχόμενα ενός πακέτου κρυπτογραφημένου με χρήση AES.
- Οι επιθέσεις είναι ευέλικτες, έχοντας αρκετές παραλλαγές που εφαρμόζονται σε διαφορετικές περιπτώσεις. Τέλος, υπάρχει υλοποίηση της επίθεσης με κατάλληλο πρόγραμμα client, το οποίο αποδεικνύει ότι οι επιθέσεις δουλεύουν στην πράξη εναντίον της υλοποίησης του IPsec στο Linux. Για παράδειγμα, το πρόγραμμα επίθεσης επιτρέπει αποτελεσματική κρυπτανάλυση σε πραγματικό χρόνο του IPsec σε ρύθμιση μόνο κρυπτογράφησης, με χρήση του αλγορίθμου AES. Για τους παραπάνω λόγους, οι επιθέσεις αυτές εξελίσσουν την πρωτοποριακή δουλειά του Bellovin, η οποία αποκάλυψε πρώτα τα κενά ασφαλείας του IPsec σε λειτουργία μόνο κρυπτογράφησης.

Για να κατανοήσει κάποιος τον τρόπο που εφαρμόζονται οι επιθέσεις, θα πρέπει πρώτα να εστιάσει στον τρόπο με τον οποίο χρησιμοποιείται η μέθοδος CBC (Cipher Block Chaining) στο πρωτόκολλο ESP, σε λειτουργία ενθυλάκωσης. Αρχικά, το εσωτερικό πλαίσιο δεδομένων που πρέπει να προστατευτεί αντιμετωπίζεται σαν μια ακολουθία από bytes. Σε αυτή την

ακολουθία προστίθεται ένας αριθμός από συμπληρωματικά bytes και τέλος ένα byte Next Header, όπως προβλέπεται στο πρωτόκολλο. Είναι αποδεκτό τα bytes γεμίσματος να έχουν μεταβλητό μήκος και να καταλαμβάνουν πολλαπλά blocks (όσο είναι το μήκος του αλγόριθμου κρυπτογράφησης τμήματος). Υποθέτουμε ότι χρησιμοποιείται το ελάχιστο μήκος από συμπληρωματικά bytes, αν και οι επιθέσεις μπορούν να χειριστούν διαφορετικά μήκη με εύκολη τροποποίησή τους. Υποθέτουμε ότι η ακολουθία των bytes, μετά από την προσθήκη των bit γεμίσματος αποτελείται από q blocks, που το καθένα του έχει n bytes. Δηλώνουμε τα blocks αυτά ως $P_1,P_2,P_3,\ldots,P_q$. Το k δηλώνει το κλειδί που χρησιμοποιείται από τον κρυπταλγόριθμο τμήματος και οι συναρτήσεις $e_k(.)$ ($d_k(.)$) δηλώνουν την κρυπτογράφηση (αποκρυπτογράφηση) των blocks με το κλειδί k. Ένα διάνυσμα αρχικοποίησης n bits επιλέγεται τυχαία. Στη συνέχεια, τα κρυπτογραφημένα blocks δημιουργούνται σύμφωνα με την παρακάτω εξίσωση:

$$C_0 = IV, C_i = e_k(C_{i-1} \oplus P_i)\text{, } 1 \le i \le q$$

Το κρυπτογραφημένο τμήμα του εξωτερικού πλαισίου ορίζεται από την ακολουθία των blocks $C_0,C_1,\ldots,C_q$. Κατά τη λήψη των blocks το πλαίσιο δεδομένων του εξωτερικού πακέτου μπορεί να αποκωδικοποιηθεί χρησιμοποιώντας την εξίσωση:

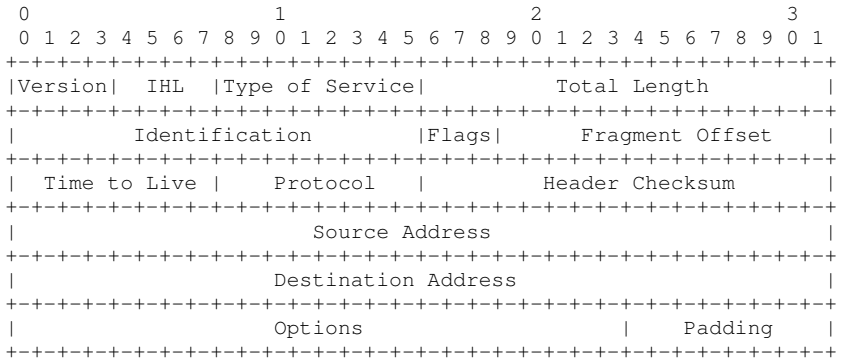
$$P_i = C_{i-1} \oplus d_k(C_i)\text{, } 1 \le i \le q$$

Οποιοδήποτε γέμισμα και το πεδίο Next Header αποκόπτονται από τα δεδομένα. Στο σημείο αυτό το κείμενο RFC που επεξηγεί την αρχιτεκτονική του IPsec, ορίζει ότι η υλοποίηση θα πρέπει να ελέγξει ότι η κρυπτογραφική διαδικασία για την αποκωδικοποίηση του εσωτερικού πακέτου ταιριάζει με αυτή που έχει προσδιοριστεί στις τοπικές ρυθμίσεις του IPsec. Κατά συνέπεια, αν ο έλεγχος αυτός αποτύχει, το πακέτο θα πρέπει να απορριφθεί, χωρίς ωστόσο αυτό να γίνεται σαφές στο σχετικό κείμενο RFC. Στην υλοποίηση του Linux το εσωτερικό πακέτο στέλνεται απευθείας στο λογισμικό του πρωτοκόλλου IP, χωρίς να γίνουν οι προαναφερόμενοι έλεγχοι. Το πρωτόκολλο IP συνήθως δρομολογεί το πακέτο στη διεύθυνση του αποδέκτη του εσωτερικού πακέτου.

Η λειτουργία κρυπτογράφησης CBC έχει ένα κενό ασφαλείας, γνωστό ως αλλοίωση δυαδικών τιμών (bit flipping vulnerability). Υποθέτουμε ότι ο επιτιθέμενος συλλέγει μια σειρά κρυπτογραφημένων δεδομένων $C_0,C_1,\ldots,C_q$. εν συνεχεία αλλάζει το bit j στο block C_{i-1} και στέλνει το τροποποιημένο πακέτο στο δίκτυο. Στην παραλαβή και αποκωδικοποίηση αυτή η αλλαγή του bit j του κρυπτογραφημένου block μετασχηματίζεται σε μια αλλαγή στη θέση j στο block P_i του απλού κειμένου. Αυτό προκύπτει από τη συνάρτηση αποκρυπτογράφησης $P_i = C_{i-1} \oplus d_k(C_i)$. Έτσι, ο επιτιθέμενος μπορεί να εισάγει ελεγχόμενες αλλαγές τιμών στα bit του απλού κειμένου που βλέπει ο αποδέκτης, τροποποιώντας την τιμή συγκεκριμένων bits του κρυπτοκειμένου και εισάγοντας τα πακέτα αυτά στο δίκτυο.

Φυσικά, το πρόβλημα για τον επιτιθέμενο είναι ότι, όποια αλλαγή στο block C_{i-1} οδηγεί σε μια τιμή του P_{i-1} η οποία στην πράξη είναι τυχαία. Αντιθέτως, εάν η αλλαγή γίνει στο διάνυσμα αρχικοποίησης IV, τότε δεν προκύπτει καμία αλλαγή στο blocks του απλού κειμένου.

Η εκτέλεση των επιθέσεων στο ESP σε λειτουργία ενθυλάκωσης, εξαρτάται από τη δομή της επικεφαλίδας του πακέτου IP και τη σειρά με την οποία επεξεργάζονται τα πεδία της επικεφαλίδας. Στην περίπτωση του πρωτοκόλλου IP v4, όπως ορίζεται στο σχετικό πρότυπο RFC 791, η επικεφαλίδα έχει τη μορφή του παρακάτω σχήματος:



Σχήμα 13. Η δομή της επικεφαλίδας IP

Το πεδίο IHL (Internet Header Length) έχει μήκος 4 bit και τιμή μεταξύ 5 και 15. Το πεδίο αυτό δηλώνει το μήκος της επικεφαλίδας σε λέξεις των 32 bit. Τυπική τιμή είναι το 5, ενώ μεγαλύτερη τιμή δηλώνει ότι μετά την κύρια

επικεφαλίδα υπάρχει το πεδίο επιλογών (Options). Το πεδίο αυτό μπορεί να έχει μήκος έως 10 λέξεις των 32 bit (40 bytes). Έχει μια αυστηρή δομή και αν αυτή δεν ακολουθηθεί, τότε δημιουργείται ένα μήνυμα ICMP (Internet Control Message Protocol) που αναφέρει πρόβλημα στις παραμέτρους και δρομολογείται στον υπολογιστή του αποστολέα. Πειράματα επιβεβαιώνουν ότι με την παραλαβή ενός μηνύματος που έχει τυχαία bits στο πεδίο επιλογών, η υλοποίηση του IP του Linux, δημιουργεί ένα μήνυμα ICMP με πιθανότητα 98,5%.

Το πεδίο Protocol έχει 8 bits και δείχνει ποιό πρωτόκολλο ανώτερου επιπέδου μεταφέρεται στο πακέτο IP. Ένα ελάχιστο σύνολο από υποστηριζόμενα πρωτόκολλα περιλαμβάνει το ICMP, TCP και UDP. Όταν το πακέτο IP φθάσει στον προορισμό του (όπως αυτός προσδιορίζεται στο σχετικό πεδίο Destination Address των 32 bits), το πεδίο protocol ανιχνεύεται. Η τιμή του προσδιορίζει σε ποιό πρωτόκολλο ανώτερου επιπέδου θα προωθηθεί το πακέτο. Εάν το σχετικό πεδίο περιέχει ένα πρωτόκολλο που δεν υποστηρίζεται από τον παραλήπτη, τότε η τοπική υλοποίηση του IP θα πρέπει να δημιουργήσει ένα μήνυμα ICMP τύπου protocol unreachable.

Το πεδίο Header Checksum έχει μια τιμή 16 bits (2 bytes), η οποία δημιουργείται από την επεξεργασία της επικεφαλίδας (περιλαμβανομένου και του πεδίου Options, αν υπάρχει) σαν μια ακολουθία λέξεων των 16 bit, προσθέτοντάς τα χρησιμοποιώντας αριθμητική συμπληρωματικής μονάδας και παίρνοντας εν συνεχεία το άθροισμα σαν αποτέλεσμα. Εάν η τιμή που προκύπτει από αυτή τη διαδικασία δεν είναι αναμενόμενη κατά την παραλαβή, τότε το πακέτο απορρίπτεται, χωρίς το γεγονός αυτό να ανακοινώνεται με κάποιο μήνυμα.

Στο Linux, η σειρά των βημάτων επεξεργασίας ενός πακέτου IP είναι η ακόλουθη. Πρώτον, γίνονται βασικοί έλεγχοι στα πεδία Version και IHL. Η επόμενη ενέργεια είναι να ελεγχθεί το πεδίο Header Checksum. Μετά από αυτό, διενεργείται ένας έλεγχος του μήκους του πακέτου χρησιμοποιώντας το πεδίο Total Length. Το πακέτο απορρίπτεται εάν αποτύχει κάποιος από αυτούς τους ελέγχους. Εν συνεχεία, γίνεται η επεξεργασία του πεδίου Options εάν η τιμή του IHL δείχνει ότι υπάρχει το πεδίο αυτό. Υποθέτοντας ότι αυτό έγινε επιτυχώς, λαμβάνεται μια απόφαση δρομολόγησης: είτε το πακέτο

παραλαμβάνεται τοπικά, είτε προωθείται σε άλλον υπολογιστή. Στην πρώτη περίπτωση, το πεδίο Protocol χρησιμοποιείται για να προσδιορίσει το πρωτόκολλο ανώτερου επιπέδου στο οποίο θα παραδοθεί το πακέτο, ενώ στη δεύτερη περίπτωση ελέγχεται η τιμή του πεδίου TTL και απορρίπτεται το πακέτο εάν η τιμή αυτή είναι μηδέν.

Το ICMP είναι ένα σημαντικό μέρος της υλοποίησης του IP, που επιτρέπει να αναφέρονται στους υπολογιστές του δικτύου τα προβλήματα που εμφανίζονται, να ελέγχονται οι διαδρομές και να συλλέγονται διαγνωστικά δεδομένα. Στην περίπτωση παραλαβής ενός προβληματικού πακέτου από έναν υπολογιστή, αυτός δημιουργεί ένα μήνυμα ICMP. Αυτό το μήνυμα περιλαμβάνει ολόκληρη την επικεφαλίδα του πακέτου (συμπεριλαμβανομένου και του πεδίου επιλογών), μαζί με ένα μεταβαλλόμενο αριθμό από bytes του υπόλοιπου πακέτου. Σύμφωνα με το κείμενο RFC 792, τουλάχιστον 8 bytes από το υπόλοιπο πακέτο πρέπει να αποσταλούν, ενώ σύμφωνα με το κείμενο RFC 1812, το μήνυμα ICMP πρέπει να περιλαμβάνει όσο το δυνατόν περισσότερα bytes, χωρίς όμως να ξεπερνάει τα 576. Αυτό έχει ως σκοπό να βοηθήσει τη διάγνωση των σφαλμάτων και με αυτόν τον τελευταίο τρόπο έχει υλοποιηθεί το ICMP στο Linux.

2.6.1.1 Επιθέσεις που βασίζονται στην τροποποίηση του πεδίου Destination Address

Στο σημείο αυτό μπορούμε να δούμε την πρώτη ομάδα επιθέσεων στο πρωτόκολλο ESP με κρυπτογράφηση μόνο σε λειτουργία ενθυλάκωσης. Αρχικά, θεωρούμε το μήκος του block του κρυπταλγόριθμου τμήματος ίσο με 64 bit. Η επίθεση, η οποία περιγράφεται στην ενότητα αυτή και γίνεται σε δύο φάσεις, εισάγει στις πιο μελετημένες επιθέσεις που ακολουθούν στις ενότητες 2.6.1.2 και 2.6.1.3. Η περιγραφή της επίθεσης γίνεται στην περίπτωση ενός ζεύγους διακομιστών ασφαλείας που επικοινωνούν χρησιμοποιώντας το πρωτόκολλο ESP σε λειτουργία ενθυλάκωσης, με μόνο κρυπτογράφηση των διακινούμενων δεδομένων που προστατεύονται. Η επίθεση, επίσης, λειτουργεί και σε διαφορετικές εφαρμογές αυτής της ρύθμισης του ESP.

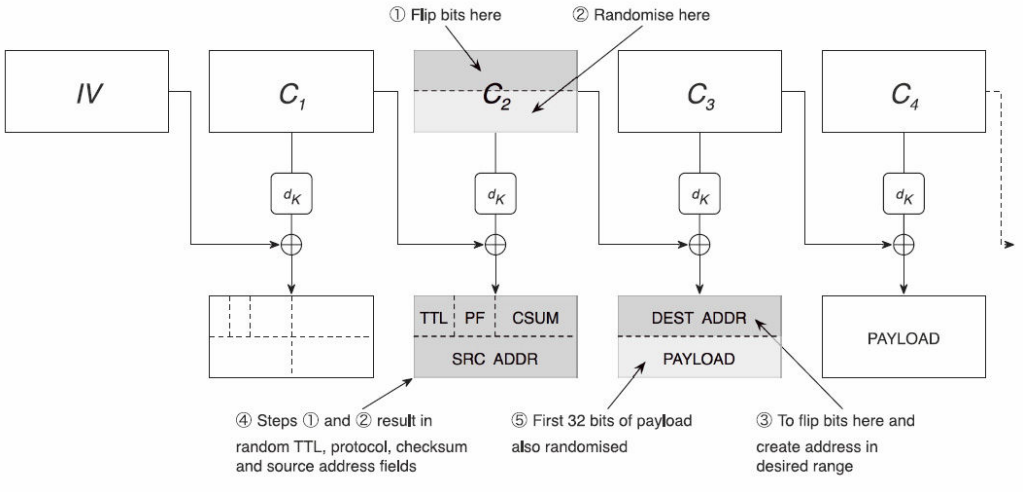
Για να είναι δυνατή η εφαρμογή των επιθέσεων θα πρέπει να γίνει μια κύρια παραδοχή. Ο επιτιθέμενος, ο οποίος ελέγχει τη διεύθυνση IP AttAddr, γνωρίζει

τη διεύθυνση προορισμού DestAddr. Αυτή η παραδοχή στη συνέχεια, δεν θα είναι απαραίτητη για κάποιες παραλλαγές των επιθέσεων.

Το πεδίο της διεύθυνσης προορισμού βρίσκεται στην $5^{η}$ λέξη 32 bit της επικεφαλίδας IP και έτσι σχηματίζει τα πρώτα 32 bits του απλού κειμένου P_3 στην ακολουθία των τμημάτων (blocks) που θα κρυπτογραφηθούν σε λειτουργία CBC από το ESP. Τα υπόλοιπα 32 bits του τμήματος είναι τα πρώτα 32 bits δεδομένων του εσωτερικού πακέτου. Η πρώτη φάση της επίθεσης ακολουθεί στα παρακάτω στάδια, με τον επιτιθέμενο στη διεύθυνση AttAddr να παρακολουθεί τα πακέτα IP κατά τη διάρκεια της επίθεσης:

1. Κατέγραψε από το δίκτυο του πακέτου στόχου που προστατεύεται από το ESP. Έστω ότι η ακολουθία $C_0,C_1,\ldots,C_q$ δηλώνει το κρυπτογραφημένο τμήμα των δεδομένων του πακέτου.
2. Τροποποίησε το τμήμα C_2 στα πρώτα 32 bits, εκτελώντας την πράξη XOR με τη μάσκα μήκους 32 bits M = DestAddr $\oplus$ AttAddr, παίρνοντας ως αποτέλεσμα ένα τροποποιημένο block C_2'.
3. Επανέλαβε την διαδικασία α και β:
   α) Τροποποίηση του τμήματος C_2', τώρα στα τελευταία 32 bits, θέτοντας αυτά τα bits σε μια τυχαία τιμή R. Έστω ότι το C_2'' δηλώνει το τροποποιημένο block.
   β) Ετοιμασία ενός τροποποιημένου πακέτου που είναι ίδιο με αυτό που καταγράφηκε στο βήμα 1, εκτός από την αντικατάσταση του block C_2 με το C_2''. Εισαγωγή του τροποποιημένου πακέτου στο δίκτυο, μέχρι να ληφθεί ένα πακέτο από τον επιτιθέμενο στη διεύθυνση AttAddr.

Στο παρακάτω σχήμα φαίνονται τα τμήματα του κρυπτοκειμένου στα οποία εφαρμόζεται η επίθεση:



Σχήμα 14. Επίθεση τύπου destination rewriting

Για να καταλάβουμε γιατί η επίθεση δουλεύει, πρέπει να προσέξουμε ότι, κάθε τροποποιημένο πακέτο έχει ως διεύθυνση προορισμού τη διεύθυνση του επιτιθέμενου. Έτσι, όταν η πύλη ασφαλείας δέχεται το τροποποιημένο εξωτερικό πακέτο και το αποκρυπτογραφεί, ανακτά το εσωτερικό πακέτο και το δρομολογεί κατευθείαν στον υπολογιστή του επιτιθέμενου (υποθέτουμε σε αυτό το σημείο ότι δεν γίνονται οι έλεγχοι σχετικά με το εάν εφαρμόστηκαν οι σωστές ρυθμίσεις του IPsec, κάτι το οποίο συμβαίνει στην υλοποίηση του IPsec στο Linux, σε αντίθεση με τις απαιτήσεις του σχετικού κειμένου RFC2401).

Το εσωτερικό πακέτο είναι σε μη κρυπτογραφημένη μορφή και τα δεδομένα του θα είναι τα ίδια με αυτά του αρχικού πακέτου, εκτός πιθανώς από τα πρώτα 32 bits που αντιστοιχούν στην τυχαιοποίηση του δεύτερου μισού του C_2. Αυτά τα δεδομένα μπορούν να ανακτηθούν εύκολα, χρησιμοποιώντας τη σχέση $P_3=P_3' (+) (M||R)$, όπου το P_3' είναι το τρίτο block στο λαμβανόμενο πακέτο, M είναι η μάσκα της διεύθυνσης που χρησιμοποιείται στο βήμα 2, και R είναι τα τυχαία bits που λαμβάνονται στο βήμα 3.

Φυσικά, λόγω των αλλαγών που έγιναν στο τμήμα C_2 κατά τη διάρκεια της επίθεσης, το block P_2 του εσωτερικού πακέτου περιέχει τυχαία bits, έτσι η επικεφαλίδα του τροποποιημένου εσωτερικού πακέτου είναι πιθανότατα μη έγκυρη. Το τμήμα P_2 περιέχει τα πεδία του χρόνου ζωής του πακέτου TTL, Protocol, header checksum και διεύθυνσης αποστολέα. Έτσι, η πιθανότητα

επιτυχίας κάθε επαναληπτικού βήματος της επίθεσης εξαρτάται από τη συνδυασμένη πιθανότητα η τιμή του TTL να είναι επαρκώς μεγάλη, ώστε το εσωτερικό πακέτο να φθάσει στον παραλήπτη, ότι το checksum είναι έγκυρο για τη νέα επικεφαλίδα και ότι το νέο πακέτο μπορεί να δρομολογηθεί προς τη διεύθυνση του αποστολέα. Όλα τα άλλα πεδία στην επικεφαλίδα θα είναι σωστά, καθόσον βρίσκονται στο block P_1, το οποίο δεν τροποποιείται από την επίθεση.

Ο επιτιθέμενος που έχει ολοκληρώσει την πρώτη φάση της επίθεσης σε μια σειρά από blocks $C_0,C_1,...,C_q$, δεν χρειάζεται να την επαναλάβει για να αποκτήσει νέα αποκρυπτογραφημένα εσωτερικά πακέτα. Αντί αυτού, μπορεί να ανακτήσει πιο αποτελεσματικά περισσότερα εσωτερικά πακέτα με την παρακάτω διαδικασία.

Ο επιτιθέμενος ξαναχρησιμοποιεί το τμήμα C_0,C_1,C_2'',C_3 του εξωτερικού πακέτου που ήταν επιτυχές στην πρώτη φάση, προσθέτοντας σε αυτά οποιαδήποτε q-6 διαδοχικά κρυπτογραφημένα τμήματα από τα δεδομένα του νέου πακέτου – στόχου και τελειώνοντας το πακέτο με τα τελευταία τρία blocks C_{q-2}, C_{q-1}, C_q του αρχικού στόχου. Μπορούν να χρησιμοποιηθούν και κενά τμήματα, ώστε τελικά να σχηματιστεί ένα πακέτο με συνολικά q τμήματα.

Ο επιτιθέμενος χρησιμοποιεί εν συνεχεία, αυτή την τροποποιημένη ακολουθία από bytes ως τα κρυπτογραφημένα δεδομένα του εξωτερικού πακέτου. Αυτή η κατασκευή διασφαλίζει ότι κατά την αποκρυπτογράφηση από την πύλη ασφαλείας του IPsec, τα δεδομένα έχουν το σωστό μήκος και μεταφράζονται ως ένα εσωτερικό πακέτο με έγκυρη επικεφαλίδα και διεύθυνση προορισμού ίση με AttAddr. Αυτό το πακέτο θα δρομολογηθεί στον υπολογιστή του επιτιθέμενου για τους ίδιους λόγους που έγινε αυτό στην κύρια φάση της επίθεσης. Από αυτό το πακέτο, ένα σύνολο από $64\cdot(q-6)$ bits απλού κειμένου από τα δεδομένα του νέου πακέτου μπορούν να ανακτηθούν. Τα πρώτα 64 bits ανακτούνται χρησιμοποιώντας παρόμοια τεχνική με αυτή που χρησιμοποιήθηκε για την ανάκτηση του P_3 στην πρώτη φάση, ενώ τα υπόλοιπα bits παραμένουν αποκρυπτογραφημένα στα τμήματα από 5 μέχρι q-3 των δεδομένων του πακέτου.

Η κύρια υπόθεση, ότι δηλαδή ο επιτιθέμενος ξέρει τη διεύθυνση αποστολής του εσωτερικού πακέτου, μπορεί να μη ληφθεί υπόψη. Είναι αρκετό να γνωρίζει μόνο ένα τμήμα της διεύθυνσης, σύμφωνα με την παρακάτω ιδέα. Αντί να χρησιμοποιήσει μια μάσκα ίση με DestAddr ⊕ AttAddr στο βήμα 2 της επίθεσης, ο επιτιθέμενος χρησιμοποιεί μια μάσκα η οποία τροποποιεί το τμήμα της διεύθυνσης αποστολής, που είναι γνωστή σ' αυτόν και αντιστοιχεί στο τμήμα της διεύθυνσης του υπολογιστή στόχου. Εν συνεχεία, χρησιμοποιεί έναν απαριθμητή για να τροποποιήσει τα υπόλοιπα bits της διεύθυνσης προορισμού και επαναλαμβάνει την επίθεση για κάθε τιμή του μετρητή. Με μια από τις τιμές του μετρητή θα παραχθεί μια διεύθυνση, η οποία θα ταιριάζει απόλυτα με αυτή του επιτιθέμενου. Για αυτή την τιμή του μετρητή, ο επιτιθέμενος έχει την ίδια πιθανότητα με πριν (περίπου 2^{-17}) να λάβει ένα πακέτο από την πύλη ασφαλείας. Μετά αυτή την τροποποίηση, μπορεί να εφαρμοστεί ξανά η δεύτερη πιο αποδοτική φάση της επίθεσης.

Ως απόδειξη της ορθότητας της θεωρίας, υλοποιήθηκε η έκδοση των 128 bits της πρώτης φάσης της επίθεσης κατά του IPsec και IP, όπως υλοποιείται στο Linux. Βρέθηκε ότι 2^{15} επαναλήψεις ήταν αρκετές για να δημιουργηθεί το επιθυμητό πακέτο απλού κειμένου σε συμφωνία με τη θεωρητική ανάλυση της επίθεσης. Το πείραμα επαλήθευσε το γεγονός ότι η υλοποίηση του Linux δεν πραγματοποιεί τους ελέγχους ρυθμίσεων του IPsec, διότι σε αντίθετη περίπτωση, τα πακέτα θα είχαν απορριφθεί μετά την αποκρυπτογράφησή τους.

2.6.1.2 Επιθέσεις που βασίζονται στον τρόπο επεξεργασίας του πεδίου IP Options

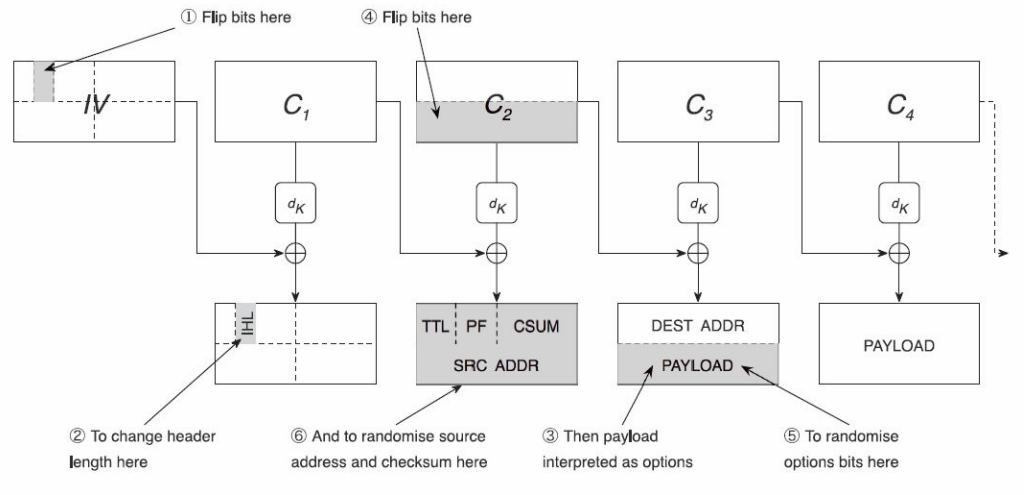
Η επόμενη σειρά επιθέσεων εκμεταλλεύεται τον τρόπο με τον οποίο οι υλοποιήσεις IP δημιουργούν μηνύματα ICMP, όταν λάβουν μηνύματα με πεδία επιλογών μη έγκυρα στις επικεφαλίδες τους. Εστιάζουν στην περίπτωση που το τμήμα του κρυπταλγόριθμου έχει μήκος 64 bit. Περιγράφεται στη συνέχεια, η επίθεση στα δεδομένα που αποστέλλονται μεταξύ ενός ζεύγους πυλών ασφαλείας που επικοινωνούν, χρησιμοποιώντας το ESP με λειτουργία ενθυλάκωσης με μόνο κρυπτογράφηση.

Για να είναι δυνατή η επίθεση, πρέπει να γίνουν κάποιες παραδοχές. Όπως και πριν, υποθέτουμε ότι ο επιτιθέμενος μπορεί να καταγράψει πακέτα προστατευμένα με το ESP και να εισάγει τροποποιημένα πακέτα στο δίκτυο. Επιπλέον, υποθέτουμε ότι ο επιτιθέμενος μπορεί να λάβει από μια πύλη ασφαλείας τα μηνύματα ICMP που δεν αποστέλλονται μέσα από το κανάλι IPsec. Για παράδειγμα, ένας εξωτερικός πάροχος Internet είναι σε θέση να εκτελέσει την επίθεση. Αυτό θα είναι, επίσης, εύκολο να επιτευχθεί, εάν η κίνηση IPsec μεταδίδονταν σε ένα ασύρματο δίκτυο, στο οποίο δεν χρησιμοποιείται προστασία WEP ή αντίστοιχη. Αργότερα, στην ανάλυση θα φανεί ότι, η απαίτηση μπορεί να μην είναι αναγκαία στην περίπτωση των 128 bit, εφόσον ο επιτιθέμενος έχει μερική πληροφόρηση σχετικά με τη διεύθυνση αποστολέα του εσωτερικού πακέτου.

Όπως και στην προηγούμενη επίθεση, ο επιτιθέμενος έχει καταγράψει ένα πακέτο και επιθυμεί να ανακτήσει το απλό κείμενο του κρυπτογραφημένου τμήματος δεδομένων του. Το πεδίο IHL βρίσκεται στο πρώτο byte της επικεφαλίδας IP και έτσι βρίσκεται στο τμήμα απλού κειμένου P_1 στην ακολουθία των τμημάτων που θα κρυπτογραφηθούν με λειτουργία CBC από το ESP. Ο επιτιθέμενος τροποποιεί τα περιεχόμενα του πεδίου IHL του εσωτερικού πακέτου, αλλάζοντας τα αντίστοιχα bits του διανύσματος αρχικοποίησης IV, κάνοντας την τιμή του IHL μεγαλύτερη από 5. Όταν το εσωτερικό πακέτο επεξεργάζεται από το λογισμικό IP στην πύλη ασφαλείας, η πρώτη λέξη (λέξεις) των δεδομένων (που σχηματίζει τα περιεχόμενα του δεύτερου μισού του P_3) θα ερμηνευτεί ως τα bytes από το πεδίο Options. Επιλέγουμε τυχαία την τιμή αυτών των bytes (όπως αυτά φαίνονται στην πύλη ασφαλείας), τοποθετώντας μια τυχαία τιμή στα τελευταία 32 bits του C_2. Τότε, με μεγάλη πιθανότητα, αυτά τα bytes θα έχουν μη έγκυρη μορφή, με αποτέλεσμα τη δημιουργία ενός μηνύματος ICMP «προβλήματος παραμέτρων». Το πλαίσιο δεδομένων αυτού του πακέτου ICMP θα περιέχει την επικεφαλίδα και ένα τμήμα των δεδομένων του εσωτερικού πακέτου. Έτσι, εάν μπορεί να καταγραφεί από τον επιτιθέμενο, μπορεί να ανακτήσει απλό κείμενο από το εσωτερικό πακέτο. Ωστόσο, η τυχαία τιμή των bytes στο C_2 έχει ως αποτέλεσμα τα bytes του P_2 να είναι και αυτά τυχαία μετά την αποκρυπτογράφηση από την πύλη ασφαλείας. Έτσι, το εσωτερικό πακέτο πιθανόν να απορριφθεί από την πύλη ασφαλείας, χωρίς η απόρριψη αυτή να γνωστοποιηθεί στο δίκτυο και πριν γίνει επεξεργασία των ρυθμίσεων IP,

εξαιτίας της μη έγκυρης τιμής checksum. Έτσι, στις περισσότερες περιπτώσεις δεν θα δημιουργηθεί το μήνυμα ICMP. Επιπλέον, εάν δημιουργηθεί το μήνυμα ICMP θα αποσταλεί σε μια τυχαία διεύθυνση, η οποία προσδιορίζεται τώρα στο P_2. Το γεγονός αυτό επιβεβαιώνει ότι το πακέτο ICMP δεν αποστέλλεται από το κανάλι IPsec ανάμεσα στις δύο πύλες ασφαλείας, κάνοντάς το έτσι ορατό στον επιτιθέμενο, αλλά, επίσης, και ότι το πακέτο ίσως δεν είναι δρομολογήσιμο. Αυτά τα προβλήματα μπορούν να ξεπεραστούν, επαναλαμβάνοντας την επίθεση όσο απαιτείται και χρησιμοποιώντας κάθε φορά νέα τυχαία bytes σε κάθε επανάληψη.

Η επίθεση φαίνεται στο παρακάτω σχήμα και τα στάδιά της είναι τα ακόλουθα :



Σχήμα 15. Επίθεση τύπου options processing

1. Κατέγραψε ένα εξωτερικό πακέτο που προστατεύεται με ESP από το δίκτυο. Έστω $C_0,C_1,...,C_q$ δηλώνει το τμήμα κρυπτογραφημένων δεδομένων αυτού του πακέτου.
2. Τροποποίησε το block C_0=IV στο πρώτο byte, κάνοντας την πράξη XOR με μια μάσκα από bits που αυξάνει την τιμή IHL πάνω από 5, λαμβάνοντας ένα block C_0΄.
3. Επανέλαβε:
   α) τροποποίησε το block C_2 στα τελευταία 32 bits, θέτοντας αυτά σε μια τυχαία τιμή R. Έστω C_2΄΄ το τροποποιημένο block.
   β) προετοίμασε ένα τροποποιημένο πακέτο που είναι ίδιο με αυτό που καταγράφηκε στο βήμα 1, εκτός από τα block C_0 και C_2 του

κρυπτογραφημένου τμήματος που έχουν αντικατασταθεί με τα C_0' και C_2''. Εισήγαγε το τροποποιημένο αυτό πακέτο στο δίκτυο,

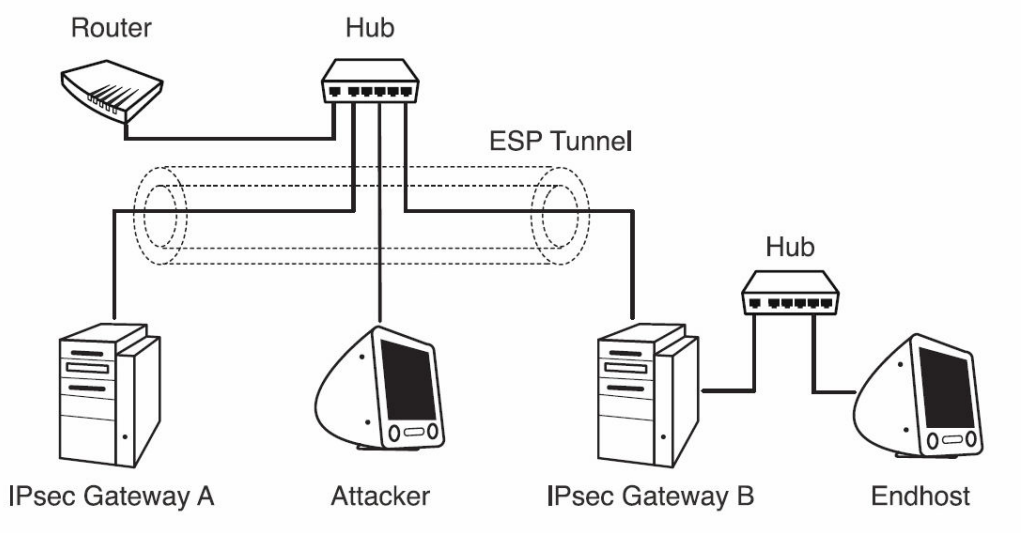
μέχρι να δημιουργηθεί ένα μήνυμα ICMP.

Στη δεύτερη φάση της επίθεσης μπορούν να χρησιμοποιηθούν τεχνικές, ανάλογες με αυτές που αναλύθηκαν στη δεύτερη φάση της προηγούμενης επίθεσης και έχουν στόχο να επιταχύνουν την ανάκτηση των δεδομένων του πακέτου, καθώς και των νέων πακέτων που καταγράφονται από το δίκτυο. Όπως και πριν μια επιτυχημένη επικεφαλίδα μπορεί να ξαναχρησιμοποιηθεί, καθόσον είναι σίγουρο ότι προκαλεί τη δημιουργία ενός μηνύματος ICMP. Η ταχύτητα ανάκτησης του απλού κειμένου στη δεύτερη φάση περιορίζεται μόνον από το χρονικό διάστημα εντός του οποίου επιτρέπεται στην πύλη ασφαλείας να δημιουργεί μηνύματα ICMP και από τον αριθμό των bytes δεδομένων που επιστρέφονται από το πρωτόκολλο ICMP.

Μια παρόμοια επίθεση είναι δυνατή όταν ο κρυπταλγόριθμος τμήματος που χρησιμοποιείται από το ESP έχει μέγεθος τμήματος ίσο με 128 bit. Σε αυτή την περίπτωση ωστόσο, το πεδίο IHL, Header Checksum και Source Address μπορούν να τροποποιηθούν αλλάζοντας τα bits στο C_0=IV. Αυτό επιτρέπει στις πιθανές τιμές του checksum να ελεγχθούν συστηματικά, κάτι που αυξάνει την πιθανότητα επιτυχίας. Τα δεδομένα του πακέτου που θεωρούνται από την πύλη ασφαλείας ως bytes του πεδίου Options, μπορούν να γίνουν τυχαία λαμβάνοντας μια τυχαία τιμή για το C_2. Όπως και πριν, παραπάνω απλό κείμενο μπορεί να ανακτηθεί γρηγορότερα σε μια δεύτερη φάση, η οποία ξαναχρησιμοποιεί την επιτυχή επικεφαλίδα από την πρώτη φάση. Επιπλέον, εάν ο επιτιθέμενος έχει μερική ή πλήρη γνώση της διεύθυνσης προορισμού των εσωτερικών πακέτων, τότε μπορεί να χρησιμοποιήσει παρόμοιες τεχνικές με αυτές που αναλύθηκαν πριν για να κατευθύνει την απάντηση ICMP στον υπολογιστή του, αυτή τη φορά αλλάζοντας τη διεύθυνση προορισμού της εσωτερικής επικεφαλίδας, τροποποιώντας το διάνυσμα αρχικοποίησης IV. Αυτή είναι μια σημαντική παραλλαγή, καθόσον καθιστά μη αναγκαία την αυστηρότερη προϋπόθεση της επίθεσης, ότι δηλαδή ο επιτιθέμενος είναι σε θέση να καταγράψει στην πύλη ασφαλείας μηνύματα ICMP.

Έχουν υλοποιηθεί με επιτυχία οι δύο φάσεις αυτής της επίθεσης στο IPsec και IP όπως υλοποιούνται στο Linux. Το παρακάτω σχήμα δείχνει την πειραματική

διάταξη, με δύο υπολογιστές που τρέχουν Linux και υλοποιούν τις πύλες ασφαλείας για το κανάλι ESP χρησιμοποιώντας τον DES ή τον AES ως κρυπταλγόριθμο τμήματος.



Σχήμα 16. Πειραματική διάταξη επιθέσεων στο IPsec

Ο υπολογιστής Endhost στο παραπάνω σχήμα δεν είναι ενεργός κατά τη διάρκεια της επίθεσης. Αυτοί οι υπολογιστές είναι συνδεδεμένοι σε ένα hub, ώστε να είναι εύκολη η παρακολούθηση και καταγραφή των πακέτων στο δίκτυο. Επίσης, συνδεδεμένος στο hub είναι και ένας router (δρομολογητής), ρυθμισμένος να είναι ο προκαθορισμένος router για τις πύλες ασφαλείας, διασφαλίζοντας έτσι, ότι κάθε μήνυμα ICMP θα διανύσει ένα πρώτο τμήμα του δικτύου μέχρι τον δρομολογητή στην προσπάθειά του να φθάσει στον προορισμό του.

Χρησιμοποιήθηκε η τιμή 6 για το πεδίο IHL, ώστε να μεγιστοποιηθεί ο αριθμός από τα bytes απλού κειμένου που θα επιστρέψουν για κάθε τροποποιημένο πακέτο στη δεύτερη φάση. Παρατηρήθηκε, πειραματικά, ότι η εμφάνιση ενός πακέτου με τυχαία διεύθυνση αποστολέα και τυχαία bytes στο πεδίο Options στην υλοποίηση του Linux οδηγεί με πιθανότητα περίπου 0,85 στη δημιουργία ενός μηνύματος ICMP τύπου προβλήματος παραμέτρων. Επιπλέον, η πιθανότητα μια τυχαία τιμή 16 bit να καθιστά την τιμή checksum του εσωτερικού πακέτου σωστή, είναι περίπου 2^{-16}. Έτσι, η αναμενόμενη πιθανότητα επιτυχίας της πρώτης φάσης της επίθεσης στην περίπτωση των

64 bit, είναι περίπου 0,85 x 2^{-16} ανά επανάληψη [15]. Έτσι, η πιθανότητα επιτυχίας μετά από t επαναλήψεις είναι:

$1-(1-0,85 \times 2^{-16})^t$

Για παράδειγμα, από τον παραπάνω τύπο μπορεί να υπολογιστεί ότι για t = 2^{16} επαναλήψεις, η πιθανότητα επιτυχίας είναι 57%.

Εκτελέστηκε 100 φορές η πρώτη φάση της επίθεσης. Χρειάστηκε ένας μέσος όρος 77.600 επαναλήψεων (που διήρκησε 2,64 λεπτά με το λογισμικό επίθεσης), για να παραχθεί ένα μήνυμα ICMP. Το Linux είναι «γενναιόδωρο» και παρέχει 524 bytes από τα δεδομένα του εσωτερικού πακέτου στα μηνύματα ICMP. Ως συνέπεια, η πρώτη φάση και κάθε τροποποιημένο πακέτο της δεύτερης φάσης αποκαλύπτουν 512 bytes απλού κειμένου (εφόσον τα κρυπτογραφημένα δεδομένα που επιλέγονται στην πρώτη φάση είναι περισσότερα από 568 bytes, περιλαμβανομένου του διανύσματος αρχικοποίησης IV και της κρυπτογραφημένης επικεφαλίδας). Έτσι, η δεύτερη φάση μπορεί να ανακτήσει γρήγορα τα δεδομένα των εσωτερικών πακέτων. Το λογισμικό επίθεσης γραμμένο σε γλώσσα C, συλλέγει πολλαπλά ESP πακέτα, διαλέγει αυτό με το καλύτερο μέγεθος για την πρώτη φάση και, εν συνεχεία, τρέχει τη δεύτερη φάση, που είναι γρηγορότερη στα υπόλοιπα πακέτα. Επίσης, υποστηρίζει και την περίπτωση χρήσης κρυπταλγορίθμων τμήματος μεγέθους 128 bit.

2.6.1.3 Επιθέσεις που βασίζονται στην τροποποίηση του πεδίου Protocol

Μια τρίτη μορφή επιθέσεων, εκμεταλλεύεται τον τρόπο με τον οποίο δημιουργούνται μηνύματα ICMP, όταν τα πακέτα περιέχουν πρωτόκολλα που δεν υποστηρίζονται από τα ανώτερα επίπεδα. Η επίθεση αυτή εφαρμόζεται σε κρυπταλγόριθμους τμήματος με μέγεθος block 128 bit, καθόσον για αυτό το μέγεθος είναι πιο αποδοτική. Οι υποθέσεις της επίθεσης είναι οι ίδιες ακριβώς με αυτές που ελήφθησαν και στην προηγούμενη επίθεση.

Το πεδίο protocol βρίσκεται στο δεύτερο byte της τρίτης λέξης 32 bit της επικεφαλίδας IP και έτσι είναι στο τμήμα P_3 του απλού κειμένου στην ακολουθία των blocks που κρυπτογραφούνται σε λειτουργία CBC από το ESP.

Ο επιτιθέμενος τροποποιεί το περιεχόμενο του πεδίου Protocol του εσωτερικού πακέτου, αλλάζοντας τα κατάλληλα bit στο διάνυσμα αρχικοποίησης IV, κάνοντας τη νέα τιμή του πεδίου να αντιστοιχεί σε ένα πρωτόκολλο που δεν υποστηρίζεται από τον αποδέκτη του πακέτου. Τώρα, όταν φθάσει το πακέτο στον τελικό προορισμό του, θα δημιουργηθεί ένα μήνυμα ICMP τύπου “protocol unreachable”. Τα δεδομένα αυτού του πακέτου ICMP θα περιέχουν την επικεφαλίδα και ένα τμήμα των δεδομένων του εσωτερικού πακέτου. Έτσι, εάν μπορεί να καταγραφεί από τον επιτιθέμενο, τότε αυτός μπορεί να ανακτήσει απλό κείμενο από το εσωτερικό πακέτο. Η διαφορά από την προηγούμενη επίθεση στο πεδίο options, είναι ότι ο υπολογιστής προορισμού (endhost) είναι αυτός που δημιουργεί το μήνυμα ICMP και όχι η πύλη ασφαλείας.

Ο επιτιθέμενος τώρα θα πρέπει να λύσει δύο προβλήματα. Πρώτον, θα πρέπει να αλλάξει τη διεύθυνση αποστολέα του εσωτερικού πακέτου, ώστε το μήνυμα ICMP να μη δρομολογηθεί μέσα από το κανάλι IPsec και να είναι σε θέση να το ανακτήσει. Δεύτερον, θα πρέπει να διορθώσει την τιμή checksum της επικεφαλίδας, ώστε να περιέχει τη σωστή τιμή για την τροποποιημένη επικεφαλίδα. Ευτυχώς, στην περίπτωση των 128 bit και οι δύο αυτοί περιορισμοί μπορούν να ικανοποιηθούν από την επεξεργασία μόνο του IV, με ένα συστηματικό τρόπο που οδηγεί σε αποδοτική επίθεση.

Θεωρούμε έναν επιτιθέμενο που τροποποιεί το πεδίο Protocol αλλάζοντας το bit i ($0 \le i < 8$) του πεδίου και τη διεύθυνση του εσωτερικού πακέτου στο bit j ($0 \le i < 32$). Και οι δύο αυτές αλλαγές μπορούν να γίνουν αλλάζοντας το IV. Για να διορθωθεί η τιμή checksum του εσωτερικού πακέτου ο επιτιθέμενος εφαρμόζει την πράξη XOR με δύο μάσκες διαδοχικά (μία μάσκα για κάθε αλλαγή bits), αλλάζοντας ξανά τα bits στο IV. Μια λεπτομερής ανάλυση του αλγορίθμου checksum δείχνει ότι μία από τις 17 δυνατές μάσκες θα διορθώσει κάθε αλλαγή σε bit. Ο επιτιθέμενος δοκιμάζει αυτά τα ζευγάρια από μάσκες με σειρά μειούμενης πιθανότητας. Θα χρειαστούν το πολύ 17^2=289 επαναλήψεις, με μια αναμενόμενη τιμή μικρότερη, καθόσον οι πιθανότητα επιτυχίας των μασκών είναι κατανεμημένη. Στην πράξη, μια απλή ανάλυση δείχνει ότι όταν $i+8 \neq j \bmod 16$, η αναμενόμενη τιμή επαναλήψεων είναι λίγο μικρότερη από 7 και μικρότερη ακόμη από το 7 όταν $i+8 = j \bmod 16$. Η επίθεση αυτή μπορεί να περιγραφεί όπως ακριβώς και οι προηγούμενες επιθέσεις.

Σε μια σημαντική παραλλαγή αυτής της επίθεσης που τώρα απαιτεί 2^{15} επαναλήψεις, ο επιτιθέμενος μπορεί, επιπροσθέτως, να ανακτήσει πληροφορία για τη διεύθυνση αποστολέα του εσωτερικού πακέτου, ώστε να την ξαναχρησιμοποιήσει, εξασφαλίζοντας έτσι ότι κάθε απάντηση ICMP θα δρομολογηθεί σε έναν υπολογιστή τον οποίο ελέγχει. Αυτό το γεγονός καθιστά μη αναγκαία την παραδοχή ότι ο επιτιθέμενος είναι σε θέση να παρακολουθεί την πύλη ασφαλείας για μηνύματα ICMP.

Όπως και στη δεύτερη επίθεση, μόλις ολοκληρωθεί η πρώτη φάση, μια δεύτερη φάση μπορεί να εφαρμοστεί, η οποία ανακτά τα πλήρη περιεχόμενα του υπολοίπου αρχικού πακέτου, καθώς και επιπλέον πακέτων στόχων.

Μια παρόμοια, αλλά λιγότερη αποδοτική επίθεση, μπορεί να γίνει όταν το μέγεθος του τμήματος είναι 64 bit, μόνο που τώρα το πεδίο protocol αλλάζει θέτοντας τυχαία τιμή στα τελευταία 32 bits του block C_2. Η πιθανότητα επιτυχίας καθορίζεται από την ανάγκη το πεδίο checksum να έχει σωστή τιμή και η τυχαία τιμή στο πεδίο protocol να αναπαριστά ένα πρωτόκολλο που δεν υποστηρίζεται. Στην πράξη αυτή είναι κοντά στην τιμή 2^{-16}, επειδή τυπικά ένας περιορισμένος αριθμός από πρωτόκολλα υποστηρίζεται. Όπως και πριν, περισσότερο απλό κείμενο μπορεί να ανακτηθεί γρηγορότερα σε μια δεύτερη φάση, η οποία ξαναχρησιμοποιεί την επιτυχημένη επικεφαλίδα της πρώτης φάσης.

Έχουν υλοποιηθεί επιτυχώς οι δύο φάσεις της επίθεσης στα 128 bit της υλοποίησης IP και IPsec του Linux. Η πειραματική διάταξη της επίθεσης φαίνεται στο σχήμα 16. Στη συγκεκριμένη επίθεση χρησιμοποιήθηκαν οι τιμές i=0 και j=6, ενώ και άλλα ζεύγη τιμών θα αποδίδανε εξίσου καλά.

Σύμφωνα με την ανάλυση πιθανοτήτων που έγινε στην προηγούμενη επίθεση, ο αναμενόμενος αριθμός επαναλήψεων για την πρώτη φάση, με αυτές τις παραμέτρους, είναι λίγο μικρότερος από το 7. Πραγματοποιήθηκε 1000 φορές η πρώτη φάση της επίθεσης. Μια μέση τιμή 6,53 επαναλήψεων (που διήρκησε 1,34 δευτερόλεπτα) χρειάστηκε για να δημιουργηθεί επιτυχώς ένα μήνυμα ICMP τύπου “protocol unreachable”, που περιείχε πληροφορία απλού κειμένου. Εξαιτίας του τρόπου που υλοποιείται το ICMP στο Linux, η πρώτη

φάση και κάθε τροποποιημένο πακέτο της δεύτερης φάσης αποκαλύπτει 500 bytes απλού κειμένου. Αυτό σημαίνει ότι, το λογισμικό της επίθεσης μπορεί να ανακτήσει μεγάλα τμήματα απλού κειμένου εύκολα στη δεύτερη φάση της επίθεσης. Τέλος, εξαιτίας του μικρού αριθμού των προσπαθειών που χρειάζονται, η επίθεση μπορεί να είναι αποτελεσματική σε πραγματικό χρόνο.

2.6.2 Επιθέσεις εναντίον του πρωτοκόλλου IPsec

Στην εργασία των Jean Paul Degabriele και Kenneth G Paterson [05], παρουσιάστηκαν επιθέσεις εναντίον του ίδιου του πρωτοκόλλου IPsec στην περίπτωση χρήσης του ESP σε λειτουργία ενθυλάκωσης με μόνο κρυπτογράφηση (encryption-only ESP), όπως αυτό περιγράφεται στα σχετικά κείμενα RFC. Οι επιθέσεις αυτές είναι ρεαλιστικές και αποδοτικές. Είναι τύπου μόνο κρυπτοκειμένου (encryption-only) και έχουν ως απαίτηση την παρακολούθηση της κίνησης πακέτων προστατευμένων με ESP και τη δυνατότητα εισαγωγής νέων πακέτων στο δίκτυο.

Όπως φάνηκε στην προηγούμενη ενότητα 2.6.1, οι ρυθμίσεις του IPsec με μόνο κρυπτογράφηση, είναι πλήρως ανασφαλείς. Παρουσιάστηκαν ρεαλιστικές επιθέσεις εναντίον του IPsec σε λειτουργία ενθυλάκωσης και εφαρμόστηκαν στην πράξη εναντίον της υλοποίησης του IPsec στο Linux. Ωστόσο, αυτές οι επιθέσεις δεν μπορούν να εφαρμοστούν σε υλοποιήσεις που είναι συμβατές με τα σχετικά πρότυπα, όπως αυτά περιγράφονται στα κείμενα RFC. Ειδικότερα, οι επιθέσεις αυτές δεν δουλεύουν όταν πραγματοποιούνται συγκεκριμένοι έλεγχοι όπως θα έπρεπε να γίνονται, σύμφωνα με την αρχιτεκτονική του IPsec. Ο λόγος που συμβαίνει αυτό είναι ότι τα τροποποιημένα πακέτα IP που χρησιμοποιούνται στις επιθέσεις δεν θα περνούσαν τους ελέγχους παραμέτρων και θα απορρίπτονταν από το IPsec. Η υλοποίηση του Linux δεν πραγματοποιεί τέτοιους ελέγχους.

Αυτή η θεώρηση αναδεικνύει τον κίνδυνο η εργασία των Paterson και Yau να υποτιμηθεί από τους προγραμματιστές και χρήστες του πρωτοκόλλου, οι οποίοι θα υποθέσουν ότι μια συμβατή με τα πρότυπα υλοποίηση του IPsec είναι ασφαλής σε λειτουργία μόνο κρυπτογράφησης. Τελικά, η πιθανότητα να είναι μια υλοποίηση ανασφαλής είναι πολύ μεγαλύτερη από το να είναι ανασφαλές το ίδιο το πρότυπο. Ωστόσο, από την εργασία των Degabriele και Paterson, φαίνεται ότι η μεγάλη κοινότητα των χρηστών του IPsec δεν πρέπει

να ξεγελαστεί από αυτή τη μη ρεαλιστική αίσθηση ασφαλείας, καθώς παρουσιάζονται επιθέσεις εναντίον του IPsec σε λειτουργία μόνο κρυπτογράφησης, σύμφωνα με τα σχετικά πρότυπα των κειμένων RFC.

Οι νέες επιθέσεις είναι ρεαλιστικές και αποδοτικές και εφαρμόστηκαν σε διάφορες υλοποιήσεις ανοικτού κώδικα του IPsec. Η κύρια επίθεση είναι δε εφαρμόσιμη, μόνο εάν μια υλοποίηση πραγματοποιεί πλήρη έλεγχο της διαδικασίας προσθήκης συμπληρωματικών bytes (full padding length), σε συμφωνία με τις συστάσεις του σχετικού κειμένου RFC. Έτσι, οι επιθέσεις εφαρμόζονται εκεί που οι προηγούμενες του Bellovin αποτρέπονταν και αντιστρόφως. Υπάρχουν, επίσης, και παραλλαγές αυτών, που εφαρμόζονται όταν πραγματοποιούνται όχι τόσο αυστηροί έλεγχοι στη διαδικασία συμπλήρωσης. Συμπερασματικά, οι διάφορες μορφές επιθέσεων καταδεικνύουν ότι η λειτουργία μόνο κρυπτογράφησης στο IPsec δεν είναι ασφαλής, ανεξάρτητα εάν γίνονται ή όχι έλεγχοι στη διαδικασία προσθήκης συμπληρωματικών bytes.

Οι επιθέσεις αυτές εξαρτώνται από τον τρόπο με τον οποίο γίνεται η κρυπτογράφηση σε λειτουργία CBC και η προσθήκη συμπληρωματικών bytes στο πλαίσιο δεδομένων ESP (καθώς και από τις αντίστροφες λειτουργίες στη φάση της αποκρυπτογράφησης). Η παραλλαγή της λειτουργίας CBC που χρησιμοποιεί το ESP σε mode ενθυλάκωσης, αναλύεται παρακάτω. Το αρχικό εσωτερικό πακέτο που πρέπει να προστατευτεί, αντιμετωπίζεται ως μια ακολουθία δεδομένων. Στην ακολουθία αυτή προστίθεται ένα συγκεκριμένο πλαίσιο δεδομένων και εν συνεχεία, το πεδίο Pad Length (PL) και Next Header (NH), όπως φαίνεται στο σχετικό σχήμα 12 του κεφαλαίου 2.5. Κάθε διαφορετικός αλγόριθμος κρυπτογράφησης που χρησιμοποιείται στο ESP περιγράφεται σε διαφορετικό κείμενο RFC και κάθε τέτοιο κείμενο μπορεί να ορίζει τον δικό του τρόπο προσθήκης συμπληρωματικών bytes (padding). Πάντως, κανένα από τα σχετικά κείμενα δεν το κάνουν κι έτσι η προκαθορισμένη μέθοδος που ορίζεται στο RFC 4303 [10] χρησιμοποιείται στην πράξη για όλους τους αλγόριθμους. Η μέθοδος αυτή προσθέτει bytes, έτσι ώστε:

α) Ο συνολικός αριθμός των bytes (περιλαμβανομένων και των bytes των πεδίων PL και NH) να είναι ευθυγραμμισμένος με τα όρια των τμημάτων (blocks) του κρυπταλγόριθμου.

β) Το προστιθέμενο πλαίσιο δεδομένων των bytes είναι είτε ένα κενό αλφαριθμητικό, είτε t bytes της μορφής 1,2,….,t για κάποιο t τέτοιο, ώστε 1 ≤ t ≤ 255.

Είναι επιτρεπτό για το γέμισμα των bytes να είναι μεταβλητού μήκους, παρόλο που αυτή η επιλογή σπάνια συναντάται στην πράξη. Συνήθως, προστίθεται ο ελάχιστος αριθμός από bytes που συμβαδίζει με τους παραπάνω κανόνες. Είναι επιτρεπτό στη λειτουργία ενθυλάκωσης να ακολουθείται η προσθήκη (γέμισμα) των bytes από μια νέα προσθήκη μεταβλητού αριθμού από bytes, που έχουν ως σκοπό την εξασφάλιση εμπιστευτικότητας ως προς τη ροή της κίνησης δεδομένων (Traffic Flow Confidentiality ή TFC). Η τελευταία προσθήκη αποτρέπει την ανάλυση της κίνησης, που έχει ως στόχο την αποκάλυψη του πραγματικού μεγέθους του εσωτερικού πλαισίου δεδομένων. Οι επιθέσεις μπορούν να γίνουν ανεξάρτητα από τον τρόπο γεμίσματος με bytes που χρησιμοποιείται.

Το byte NH υπάρχει, ώστε η οντότητα που αποκρυπτογραφεί το IPsec να μπορεί να γνωρίζει σε πιο πρωτόκολλο θα αποστείλει τα δεδομένα πριν από τα bytes συμπληρώματος. Στη λειτουργία ενθυλάκωσης, η τιμή αυτή πρέπει να είναι 4, δηλώνοντας ενθυλάκωση του IP σε IP, δηλαδή ότι τα bytes δεδομένων είναι ένα πακέτο IP. Το τελευταίο κείμενο RFC του ESP προσδιορίζει ότι μια τιμή ίση με 59 του byte NH δείχνει ότι είναι ένα πακέτο που πρέπει να απορριφθεί (dummy packet). Τέτοια πακέτα αγνοούνται κατά την αποκρυπτογράφηση και χρησιμοποιούνται μαζί με τη συμπλήρωση bytes TFC για να παρέχουν μια υπηρεσία εμπιστευτικότητας ως προς τη ροή δεδομένων.

Μετά τη διαδικασία συμπλήρωσης τα δεδομένα κρυπτογραφούνται χρησιμοποιώντας τη λειτουργία CBC. Ας υποθέσουμε ότι η ακολουθία των bytes μετά το γέμισμα αποτελείται από q blocks, το καθένα αποτελούμενο από n bits (όπου για παράδειγμα το n είναι 64 για triple-DES και 128 για AES). Δηλώνουμε αυτά τα blocks με $P_1,P_2,\ldots,P_q$. Χρησιμοποιούμε το K για να δηλώσουμε το κλειδί του κρυπταλγόριθμου τμήματος και $e_K(.)$ ($d_K(.)$) για την κρυπτογράφηση (αποκρυπτογράφηση) των blocks με το κλειδί αυτό.

Επιλέγεται τυχαία ένα διάνυσμα αρχικοποίησης IV, που έχει μήκος n bits. Εν συνεχεία, τα κρυπτογραφημένα blocks παράγονται σύμφωνα με τις σχέσεις :

$C_0=IV$, $C_i = e_K(C_{i-1} \oplus P_i)$, $(1 \le i \le q)$

Το κρυπτογραφημένο τμήμα του εξωτερικού πακέτου ορίζεται τότε να είναι η ακολουθία των q+1 blocks $C_0,C_1,...,C_q$.

Στην οντότητα που αποκρυπτογραφεί το πακέτο IPsec (και κατέχει το κλειδί K), τα δεδομένα του εξωτερικού πακέτου μπορούν να ανακτηθούν βάσει των παρακάτω εξισώσεων :

$P_i=C_{i-1} (+) d_K(C_i)$, $(1 \le i \le q)$

Τα συμπληρωματικά bytes μαζί με τα PL και NH bytes μπορούν, εν συνεχεία, να αφαιρεθούν, αποκαλύπτοντας το αρχικό εσωτερικό πλαίσιο δεδομένων. Στο σχετικό κείμενο RFC αναφέρεται ότι ο δέκτης θα πρέπει να ανιχνεύσει το μήκος των συμπληρωματικών bytes, διότι κατ' αυτόν τον τρόπο αποτρέπονται συγκεκριμένες επιθέσεις τύπου προσθαφαίρεσης bytes στο πακέτο. Η αναφορά αυτή σχετίζεται με την επίθεση επιλεγμένου απλού κειμένου του Bellovin, η οποία μπορεί να ανακτήσει ένα byte ανά block από κρυπτοκείμενα συγκεκριμένου μεγέθους, ενώ η επιτυχία της εξαρτάται από το λανθασμένο έλεγχο των συμπληρωματικών bytes. Ωστόσο, τα αντίμετρα είναι αποτελεσματικά εάν η διαδικασία συμπλήρωσης με bytes ελέγχεται αυστηρά και το πακέτο απορρίπτεται εάν αυτή δεν εκτελείται όπως αναμένεται. Γι' αυτόν τον λόγο, υποθέτουμε ότι μια συμβατή υλοποίηση πραγματοποιεί αυστηρό έλεγχο, ώστε η διαδικασία προσθήκης bytes να συμφωνεί ακριβώς με την αναμενόμενη τιμή του πεδίου PL και να απορρίπτει το πακέτο εάν αποτύχει ο έλεγχος. Λόγω έλλειψης ακριβούς τεκμηρίωσης, οι υλοποιήσεις χειρίζονται με διαφορετικό τρόπο τη διαδικασία συμπλήρωσης bytes. Η σωστή διαχείριση της διαδικασίας αυτής είναι κρίσιμη όσον αφορά στην ασφάλεια. Στο σχετικό κείμενο RFC του ESP αναφέρεται έμμεσα ότι η αφαίρεση των bytes TFC είναι ευθύνη του ανώτερου επιπέδου και προσδιορίζεται από το byte NH.

Το byte NH πρέπει να εξεταστεί και αν η τιμή είναι 59, τότε το πακέτο απορρίπτεται, χωρίς περαιτέρω επεξεργασία. Στο σημείο αυτό, ο δέκτης

μπορεί να ανακατασκευάσει το εσωτερικό κρυπτογραφημένο πακέτο. Η ακριβής διαδικασία για να γίνει αυτό προσδιορίζεται στα κείμενα RFC 2401 και RFC 4301. Τα κείμενα αυτά, επιπλέον, υποχρεώνουν τις υλοποιήσεις να κάνουν έλεγχο, ώστε η κρυπτογραφική διαδικασία που αποκαλύπτει το εσωτερικό πακέτο να ταιριάζει με αυτή που προσδιορίζεται στις τοπικές ρυθμίσεις του IPsec. Εάν αποτύχει ο έλεγχος, το πακέτο πρέπει να απορριφθεί. Εάν ο έλεγχος είναι επιτυχής, τότε σε λειτουργία ενθυλάκωσης το πακέτο προωθείται στο ανώτερο πρωτόκολλο για περαιτέρω επεξεργασία.

Όταν το ESP εφαρμόζεται χωρίς προστασία ακεραιότητας, ο αριθμός ακολουθίας (sequence number) στην επικεφαλίδα ESP δεν ελέγχεται από τον παραλήπτη.

Οι επιθέσεις βασίζονται στον λεπτομερή τρόπο με τον οποίο δομούνται οι επικεφαλίδες των πακέτων IP, στον τρόπο με τον οποίο τα πεδία των επικεφαλίδων των εσωτερικών πακέτων επεξεργάζονται από την υλοποίηση του IP μετά το τέλος της επεξεργασίας IPsec και στον τρόπο με τον οποίο δημιουργούνται πακέτα ICMP στην περίπτωση σφαλμάτων κατά την επεξεργασία. Στο σχήμα 13 του κεφαλαίου 2.6.1 φαίνεται η δομή της επικεφαλίδας IP.

Ουσιαστικά, η επεξεργασία που πραγματοποιείται σε μια τυπική υλοποίηση του IP είναι η ακόλουθη. Γίνονται βασικοί έλεγχοι στο πεδίο Version και IHL. Το IHL (Internet Header Length) έχει μήκος 4 bits και τιμή ανάμεσα σε 5 και 15. Αυτό το πεδίο δείχνει το μήκος της επικεφαλίδας σε λέξεις των 32 bit. Η τυπική τιμή είναι 5 και υποδηλώνει ότι το μήκος της επικεφαλίδας είναι 20 bytes και ότι δεν υπάρχει πεδίο Options. Εάν είναι πάνω από 5, τότε υποτίθεται ότι επιπλέον bytes υπάρχουν στο πεδίο Options.

Η επόμενη ενέργεια είναι να ελεγχθεί το πεδίο Header Checksum. Αυτό το πεδίο των δύο bytes σχηματίζεται αρχικά, θεωρώντας την επικεφαλίδα (συμπεριλαμβανομένου του πεδίου Options) ως μια ακολουθία λέξεων των 16 bit, τα οποία προσθέτει χρησιμοποιώντας άθροιση συμπληρωματική του 1 και παίρνοντας εν συνεχεία, το συμπλήρωμα του 1 ως αποτέλεσμα για την τιμή του checksum. Κατά τη διάρκεια του υπολογισμού, τα bytes του Header Checksum μηδενίζονται.

Ως συνέπεια, το συμπληρωματικό της μονάδας άθροισμα των 16 bit λέξεων (τώρα περιλαμβανομένου του υπολογισμένου πεδίου Checksum) θα πρέπει να είναι ίσο με μηδέν. Εάν αποτύχει ο έλεγχος, το πακέτο απλά απορρίπτεται.

Μετά από αυτό, γίνεται ο έλεγχος του μήκους. Μια τυπική υλοποίηση όπως του Linux, θα ελέγξει ότι ο αριθμός των bytes στο πλαίσιο δεδομένων είναι τουλάχιστον όσο δείχνει η τιμή του πεδίου Total Length και τουλάχιστον όσο το ελάχιστο μέγεθος της επικεφαλίδας, δηλαδή 20 bytes. Από το πλαίσιο στη συνέχεια, θα αποκοπούν τα bytes που υπερβαίνουν τον αριθμό που ορίζει το πεδίο Total Length. Το πακέτο απορρίπτεται, εάν αποτύχουν αυτοί οι έλεγχοι. Αναφερόμαστε σε αυτό τον τύπο ελέγχου ως χαλαρό. Ο έλεγχος μήκους θα μπορούσε να είναι και αυστηρός στην περίπτωση που η υλοποίηση του IP επιβεβαιώνει ότι ο αριθμός των bytes δεδομένων είναι ακριβώς αυτός που προσδιορίζεται στο πεδίο Total Length, ενώ σε αντίθετη περίπτωση απορρίπτει το πακέτο. Ωστόσο, μια υλοποίηση χαμηλότερου επιπέδου, δεν μπορεί κατ' ανάγκη να γνωρίζει πώς να ρυθμίσει το μήκος των δεδομένων που αποστέλλει στην υλοποίηση IP, ώστε το πακέτο να έχει το σωστό μέγεθος. Επιπλέον, για να είναι δυνατή η λειτουργία του μηχανισμού προσθήκης bytes TFC, θα πρέπει η υλοποίηση IP να εκτελεί 'χαλαρό' έλεγχο και να απορρίπτει τα περιττά bytes που υπερβαίνουν τον αριθμό που ορίζεται στο πεδίο Total Length. Αυτό είναι σε συμφωνία με το πρότυπο του IP, το οποίο δεν προσδιορίζει κάποιο συγκεκριμένο έλεγχο όσον αφορά στο μήκος του πακέτου και αναφέρει ότι η υλοποίηση πρέπει να είναι ελεύθερη στον τρόπο που δέχεται τα πακέτα.

Στη συνέχεια, γίνεται η επεξεργασία στο πεδίο Options, εάν αυτό υπάρχει. Η δομή του πεδίου Options είναι αυστηρή. Εάν η δομή του δεν είναι αυτή που προβλέπεται, τότε απορρίπτεται το πακέτο και δημιουργείται ένα μήνυμα ICMP που αναφέρει πρόβλημα παραμέτρων και αποστέλλεται στον υπολογιστή ο οποίος φαίνεται στο πεδίο Source Address.

Έπειτα, λαμβάνεται μια απόφαση δρομολόγησης. Είτε το πακέτο λαμβάνεται τοπικά, είτε προωθείται σε άλλον υπολογιστή (εάν αυτός έχει ρυθμιστεί για να δρομολογεί πακέτα). Στην πρώτη περίπτωση, το πεδίο Protocol χρησιμοποιείται για να προσδιορίσει το πρωτόκολλο ανώτερου επιπέδου στο

οποίο προωθούνται τα δεδομένα του πακέτου. Εάν το πεδίο περιέχει μια τιμή που δεν αντιστοιχεί σε ένα υποστηριζόμενο πρωτόκολλο, τότε το IP θα πρέπει να απορρίψει το πακέτο και να δημιουργήσει ένα μήνυμα ICMP τύπου 'protocol unreachable', το οποίο στέλνεται πίσω στον αποστολέα του πακέτου. Λίγο παραπάνω από τις μισές από τις 256 δυνατές τιμές του πεδίου Protocol, έχουν αποδοθεί σε συγκεκριμένα πρωτόκολλα ανώτερου επιπέδου (όπως το TCP και το UDP), αλλά ένας τυπικός υπολογιστής μπορεί να εξυπηρετήσει 5 έως 10 διαφορετικά πρωτόκολλα. Στη δεύτερη περίπτωση που το πακέτο δρομολογείται, ελέγχεται το πεδίο TTL (Time to Live). Το πεδίο αυτό χρησιμοποιείται για να καταγράψει την 'ηλικία' του πακέτου. Η τιμή του αρχικοποιείται από τον αποστολέα, με τυπικές τιμές από 64 έως 128 και μειώνεται κατά 1 από κάθε δρομολογητή που μεσολαβεί στη διαδρομή του πακέτου. Εάν η τιμή TTL γίνει μηδέν, το πακέτο απορρίπτεται και ένα μήνυμα ICMP τύπου 'Time Exceeded' αποστέλλεται στον υπολογιστή που έστειλε το πακέτο. Αλλιώς το πακέτο δρομολογείται.

Στις επιθέσεις χρησιμοποιούνται τα πεδία IHL και Protocol, των οποίων κάποιες συγκεκριμένες τιμές προκαλούν τη δημιουργία μηνυμάτων ICMP. Σε μερικές από τις επιθέσεις θα τροποποιήσουμε τα πεδία TTL και Identification. Το πεδίο Identification έχει μια τιμή 16 bit που χρησιμοποιείται για να καταγράψει τα τμήματα (fragments) που αποτελούν το πλαίσιο δεδομένων, στην περίπτωση που το πλαίσιο χωρίζεται σε τμήματα. Συνήθως, η υλοποίηση του IP γεμίζει το πεδίο αυτό με την τιμή ενός απαριθμητή, η οποία αυξάνει για κάθε νέο πλαίσιο δεδομένων.

Έχουμε είδη αναφερθεί σε διάφορες περιπτώσεις, κάτω από τις οποίες δημιουργούνται μηνύματα ICMP. Η συγκεκριμένη δομή και το μήκος των δεδομένων των μηνυμάτων εξαρτώνται από το είδος σφάλματος και την υλοποίηση του ICMP.

Τα κείμενα RFC του IPsec δίνουν σύνθετες οδηγίες σχετικά με το πώς το IPsec πρέπει να χειριστεί τα παραπάνω μηνύματα. Το κείμενο RFC4301, διακρίνει τα μηνύματα σε σφάλματος και μη σφάλματος, με τον δεύτερο τύπο να αντιμετωπίζεται σύμφωνα με τις συγκεκριμένες πολιτικές και συσχετισμούς ασφαλείας (SAs).

Τα μηνύματα σφάλματος ICMP διαχωρίζονται σε δύο τύπους:
- μηνύματα που κατευθύνονται στην υλοποίηση του IPsec
- μεταβατικά μηνύματα (transit).

Όλα τα μηνύματα που θα παραχθούν κατά τη διάρκεια των επιθέσεων θα είναι μεταβατικού τύπου. Στις ρυθμίσεις του IPsec που εφαρμόζονται οι επιθέσεις, τα μηνύματα αυτά δεν θα εμποδιστούν από το IPsec, αλλά θα προωθηθούν σε κρυπτογραφημένη μορφή στο δίκτυο. Θα είναι δε αναγνωρίσιμα από το μήκος τους, το οποίο εξαρτάται από την υλοποίηση.

Στις επιθέσεις τροποποιούνται συγκεκριμένα bits στις επικεφαλίδες των εσωτερικών πακέτων. Κάθε τέτοια τροποποίηση απαιτεί περαιτέρω αλλαγές να γίνουν στην επικεφαλίδα, ώστε η τιμή του Checksum να είναι σωστή, αλλιώς το εσωτερικό πακέτο θα απορριφθεί χωρίς ειδοποίηση.

Η συνθήκη για να είναι σωστή η τιμή του Header Checksum είναι το συμπληρωματικό άθροισμα του 1 των λέξεων 16 bit της επικεφαλίδας να είναι μηδέν. Αυτό, από την πλευρά των αναγκαίων τροποποιήσεων στην εσωτερική επικεφαλίδα ώστε να έχει σωστή τιμή στο παραπάνω άθροισμα, σημαίνει ότι δεν χρειάζεται να τροποποιηθεί το ίδιο το πεδίο Header Checksum, αλλά οποιοδήποτε άλλο τμήμα μήκους 16 bit της επικεφαλίδας. Το πεδίο Identification Field, που βρίσκεται στα bytes 5 και 6 της επικεφαλίδας μπορεί να πάρει οποιαδήποτε τιμή, χωρίς να επηρεάσει την επεξεργασία, όταν τα πακέτα δεν είναι τμηματοποιημένα (fragmented). Εξαιτίας της θέσης του στην επικεφαλίδα, το πεδίο μπορεί να τροποποιηθεί αλλάζοντας τα bytes 5 και 6 του διανύσματος αρχικοποίησης IV, είτε το μέγεθος του block είναι 64 είτε 128. Αυτές οι ιδιότητες κάνουν το Identification Field ιδανική λέξη 16 bit για να τη χρησιμοποιήσει ο επιτιθέμενος, ώστε να διασφαλίσει σωστή τιμή του Checksum. Μπορεί δε να χρησιμοποιηθεί με δύο τρόπους:

α) Εάν έχουν γίνει πολλές αλλαγές στην επικεφαλίδα, τότε μπορούμε να αλλάξουμε συστηματικά την τιμή του πεδίου Identification Header σε όλες τις πιθανές τιμές, χρησιμοποιώντας έναν απαριθμητή 16 bit για την αλλαγή των bytes 5 και 6 του IV. Κατά μέσο όρο θα χρειαστούν 2^{15} και κατά μέγιστο 2^{16} προσπάθειες, για να παραχθεί μια επικεφαλίδα με έγκυρη τιμή Checksum.

β) Εάν έχουν γίνει λίγες αλλαγές στα bit του Header Checksum, τότε μπορούμε να χρησιμοποιήσουμε πίνακες από προκαθορισμένες μάσκες, που αναφέρονται στην εργασία των Paterson και Yau [04], για να μεταβάλουμε την τιμή του Identification Field. Παρόλο που αυτές οι μάσκες σχεδιάστηκαν για να μεταβάλλουν το ίδιο το Header Checksum, είναι κατανοητό, από τα προηγούμενα, ότι είναι κατάλληλες και αν εφαρμοστούν στο Identification Field. Έτσι, για παράδειγμα αν υποθέσουμε ότι μια αλλαγή ενός bit γίνεται στην επικεφαλίδα, εάν χρησιμοποιηθούν οι προαναφερόμενες μάσκες από τον πίνακα T_i [04], τότε θα χρειαστούν το πολύ να δοκιμάσουμε 17 μάσκες και κατά μέσο όρο 2 μάσκες για να παραχθεί μια επικεφαλίδα με έγκυρη τιμή στο πεδίο Checksum.

2.6.2.1 Επιθέσεις ανίχνευσης του αριθμού των συμπληρωματικών bytes της επικεφαλίδας του πακέτου

Ακολουθεί μια σύντομη αναφορά στις επιθέσεις που εισήγαγε πρώτος ο Vaudenay. Στις επιθέσεις αυτές χρησιμοποιείται ένας τρόπος εξαγωγής συμπεράσματος για το εάν η διαδικασία συμπλήρωσης με bytes της επικεφαλίδας έγινε σωστά ή όχι (padding oracle attacks). Στη λήψη ενός πακέτου το τμήμα του λογισμικού επίθεσης που είναι επιφορτισμένο με την αποκάλυψη της σωστής ή όχι δομής του πακέτου ως προς τα συμπληρωματικά bytes, αναφέρει το αποτέλεσμα του ελέγχου αυτού τροποποιώντας ανάλογα την κατάσταση ενός bit.

Έχει αποδειχθεί ότι σε λειτουργία CBC και σε μια σειρά μεθόδων συμπλήρωσης με bytes, η επανειλημμένη χρήση της παραπάνω διαδικασίας μπορεί να χρησιμοποιηθεί για την εξαγωγή της πληροφορίας σχετικά με το εάν έχει γίνει σωστά η αποκρυπτογράφηση ή όχι. Δείχνουμε πως μπορεί να γίνει αυτό για την προκαθορισμένη μέθοδο συμπλήρωσης με bytes του ESP, υποθέτοντας ότι υπάρχει ο μηχανισμός εξαγωγής συμπεράσματος της προηγούμενης παραγράφου. Για απλότητα, υποθέτουμε ότι δεν υπάρχει το byte NH, άρα το PL είναι το δεξιότερο byte του block.

Υποθέτουμε ότι ο επιτιθέμενος θέλει να αποκρυπτογραφήσει ένα block κρυπτοκειμένου C_i, $i \geq 1$, από ένα κρυπτοκείμενο παραγόμενο με CBC και αποτελούμενο από τα blocks $C_0,C_1,\ldots,C_q$. Υποθέτουμε ότι κάθε block C_j έχει t

bytes, που θα αναφέρουμε $C_{j,0}$, $C_{j,1}$, …, $C_{j,t-1}$ από αριστερά προς τα δεξιά. Ονομάζουμε τα άγνωστα bytes του $d_K(C_i)$ ως $(d_K(C_i))_0$, $(d_K(C_i))_1$,…,$(d_K(C_i))_{t-1}$ και τα άγνωστα bytes του P_i ως $P_{i,0},P_{i,1},\ldots,P_{i,t-1}$. Ο επιτιθέμενος δημιουργεί ένα κρυπτοκείμενο με δύο block της μορφής R,C_i και το στέλνει για έλεγχο συμπληρωματικών bytes (padding). Το R είναι ένα τυχαίο block, που έχει bytes τα $R_0,R_1,\ldots,R_{t-1}$. Εάν η συμπλήρωση bytes είναι σωστή, τότε το δεξιότερο byte του $R \oplus d_K(C_i)$ είναι πιθανότατα μηδέν, μια τιμή του PL που δηλώνει ότι δεν υπάρχουν συμπληρωματικά bytes. Ο επόμενος πιο πιθανός συνδυασμός συμπληρωματικού byte και τιμής PL είναι 1,1. Αυτός ο συνδυασμός είναι 256 φορές λιγότερο πιθανόν να συμβεί και μπορεί να προσδιοριστεί με μια επιπλέον σειρά από προσπάθειες εύρεσης του μήκους συμπληρωματικών bytes. Στη συνέχεια, ο επιτιθέμενος μπορεί εύκολα να υπολογίσει το $(d_K(C_i))_{t-1}$ και έτσι το $P_{i,t-1}$. Εάν ανιχνευτεί μη σωστός αριθμός από byte συμπλήρωσης, ο επιτιθέμενος μπορεί να προσπαθήσει ξανά με μια διαφορετική τιμή για το R_{t-1}. Είναι εύκολο τώρα να αντιληφθεί κανείς, ότι ο επιτιθέμενος που αλλάζει τιμή στο R_{t-1} και ανιχνεύει αν η συμπλήρωση των bytes είναι σωστή, μπορεί να εξάγει το δεξιότερο byte απλού κειμένου του P_i χρησιμοποιώντας 128 προσπάθειες κατά μέσο όρο, με μέγιστο τις 256 προσπάθειες.

Η ιδέα αυτή μπορεί να επεκταθεί, ώστε να εξαχθεί κάθε byte του P_i. Υποθέτουμε ότι το δεξιότερο byte έχει εξαχθεί, έτσι ο επιτιθέμενος γνωρίζει το $(d_K(C_i))_{t-1}$. Για να πάρει το byte $P_{i,t-2}$ μπορεί να διορθώσει το R_{t-1} ώστε $R_{t1} \oplus (d_K(C_i))_{t-1} = 1$ και να μεταβάλλει το R_{t-2} μέχρι να ανιχνευθεί ένα πακέτο με σωστό αριθμό από συμπληρωματικά byte. Όταν συμβεί αυτό, γνωρίζουμε ότι το byte συμπλήρωσης και το byte PL είναι της μορφής 1,1. Από αυτό, το byte $P_{i,t-2}$ μπορεί να εξαχθεί με παρόμοιο τρόπο. Η διαδικασία αποκρυπτογράφησης συνεχίζεται με αυτό τον τρόπο, με το σχήμα του συμπληρωματικού byte να αυξάνει σε μήκος κατά 1 σε κάθε βήμα. Η ανάκτηση κάθε byte απλού κειμένου απαιτεί κατά μέσο όρο 128 προσπάθειες ανίχνευσης της ορθότητας του αριθμού των συμπληρωματικών bytes.

Δυστυχώς, η επίθεση που περιγράφηκε παραπάνω, δεν θα δουλέψει σε μια υλοποίηση του IPsec. Πρώτον, δεν έχουμε διασφαλίσει ότι πραγματικά μπορεί να υπάρξει ένας μηχανισμός ανίχνευσης της ορθότητας του αριθμού των συμπληρωματικών bytes, καθώς και της γνωστοποίησης του αποτελέσματος

αυτού στον επιτιθέμενο. Η ανάλυση των σχετικών κειμένων RFC προτείνει τη διενέργεια αυστηρών ελέγχων, με την αποτυχία να καταγράφεται και το πακέτο με μη έγκυρο αριθμό συμπληρωματικών bytes να απορρίπτεται. Ωστόσο για να αξιοποιήσει κανείς την πληροφορία της καταγραφής, θα πρέπει να έχει πρόσβαση στο σχετικό αρχείο καταγραφής (log file) ή να μπορεί να μετρήσει το μέγεθος αυτού του αρχείου. Δεν είναι άμεσα αντιληπτό πώς μπορεί να διαχωριστεί η έλλειψη δραστηριότητας του δέκτη λόγω αποτυχίας λήψης του σωστού αριθμού των συμπληρωματικών bytes ή έλλειψη δραστηριότητας του δέκτη λόγω διάφορων άλλων λόγων. Εν αντιθέσει, στην περίπτωση του SSL/TLS η αποτυχία στη διαδικασία των συμπληρωματικών bytes, οδηγεί στη δημιουργία ενός μηνύματος, το οποίο μπορεί να ανιχνευτεί. Ωστόσο, ταυτόχρονα προκαλείται και η διακοπή της σύνδεσης SSL.

Επιπλέον, η επίθεση όπως περιγράφηκε, δεν λαμβάνει υπόψη της την παρουσία του byte NH και την επίδρασή του στην επεξεργασία μετά την αποκρυπτογράφηση και αφαίρεση των συμπληρωματικών bytes. Η επίθεση δουλεύει τοποθετώντας το block κρυπτοκειμένου που είναι στόχος της, ως το τελευταίο block σε ένα κρυπτοκείμενο μετά το τυχαίο block. Στην επίθεση όπως αυτή περιγράφεται, υπονοείται ότι το byte NH στην πραγματικότητα είναι τυχαίο. Αλλάζοντας το R, το byte NH μπορεί να τροποποιηθεί, αλλά δεν είναι ξεκάθαρο εξ' αρχής τί αποτέλεσμα θα έχει αυτή η τροποποίηση. Στην πραγματικότητα, εάν γίνονται έλεγχοι στις ρυθμίσεις και χρησιμοποιείται λειτουργία ενθυλάκωσης, τότε τα αποκρυπτογραφημένα δεδομένα από τα οποία έχουν αφαιρεθεί τα συμπληρωματικά bytes, θα απορρίπτονται πάντα, εκτός εάν το byte NH είναι ίσο με 4. Οποιαδήποτε άλλη τιμή θα έδειχνε ένα πρωτόκολλο ανώτερου επιπέδου διαφορετικό από το IP, υποδηλώνοντας έτσι ότι το ESP χρησιμοποιεί λειτουργία μεταφοράς και οδηγώντας σε παραβίαση των αναμενόμενων αποτελεσμάτων των ελέγχων. Σε αυτή την περίπτωση, η διόρθωση των συμπληρωματικών bytes δεν θα μπορούσε από μόνη της να διασφαλίσει ότι δεν θα απορριφθούν τα δεδομένα. Θα χρειάζεται, επιπλέον, η τιμή NH να είναι συγκεκριμένη. Στην ανάλυση που ακολουθεί, θεωρούμε ότι τα δεδομένα απορρίπτονται πάντα εκτός και αν η τιμή του NH είναι ίση με 4.

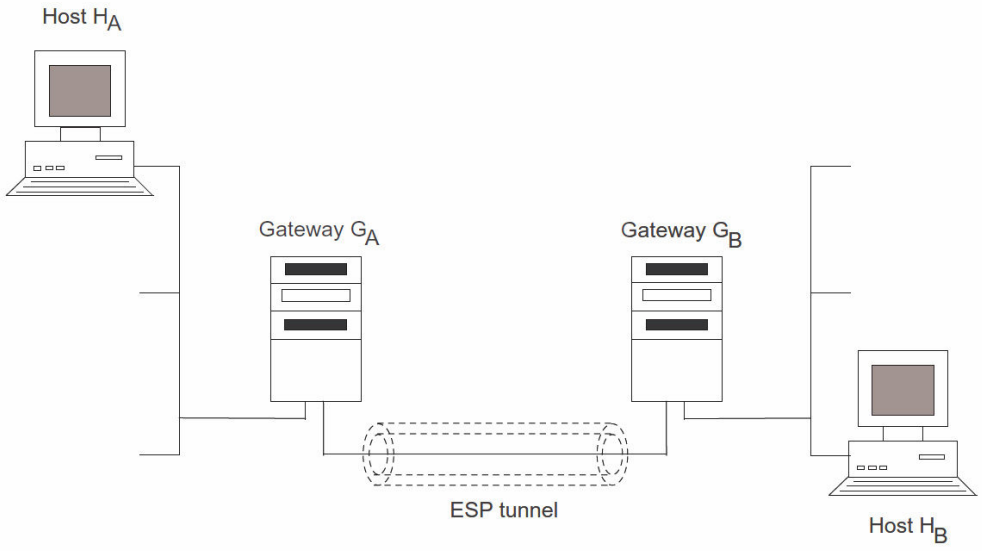
Ακόμα και αν υποθέσουμε ότι μπορεί το byte NH να ρυθμιστεί στην τιμή 4 ώστε να υποδηλώνει λειτουργία ενθυλάκωσης, υπάρχει το πρόβλημα του πώς τα αποκρυπτογραφημένα δεδομένα χωρίς τα συμπληρωματικά bytes, τα οποία

αποτελούν ένα πακέτο IP, επεξεργάζονται στη συνέχεια. Η επίθεση που περιγράφεται παρακάτω χρησιμοποιεί κρυπτοκείμενα της μορφής R,C_i, όπου το C_i είναι το block που ο επιτιθέμενος θέλει να αποκρυπτογραφήσει. Τα δεδομένα που παράγονται μετά την αποκρυπτογράφηση ανταποκρίνονται με πολύ μικρή πιθανότητα σε ένα έγκυρο πλαίσιο δεδομένων IP. Στην περίπτωση αυτή, η περαιτέρω επεξεργασία από το IP είναι σχεδόν βέβαιο ότι θα οδηγήσει σε απόρριψη του πακέτου και αδράνεια του δέκτη, η οποία φαίνεται αδύνατο να διακριθεί από την αδράνεια, λόγω εσφαλμένου αριθμού συμπληρωματικών bytes. Έτσι, πρέπει να είμαστε πιο προσεκτικοί στον τρόπο που δημιουργούμε τα κρυπτοκείμενα, ώστε οι επιθέσεις βασιζόμενες στην ανίχνευση του αριθμού των συμπληρωματικών bytes να είναι επιτυχείς.

2.6.2.2 Επιθέσεις επιλεγμένου απλού κειμένου

Ακολουθεί η περιγραφή μιας επίθεσης επιλεγμένου απλού κειμένου που δείχνει τις βασικές αρχές για τις επιθέσεις επιλεγμένου κρυπτοκειμένου, οι οποίες παρατίθενται κατωτέρω. Για πληρότητα, υποθέτουμε ότι το μέγεθος του block του κρυπταλγόριθμου τμήματος που χρησιμοποιείται στο ESP είναι 64 bits. Είναι εύκολο να προσαρμόσουμε την επίθεση στην περίπτωση που έχουμε 128 bit. Όπως και με τις υπόλοιπες επιθέσεις κάνουμε τις παρακάτω παραδοχές:

α) Χρησιμοποιείται ESP μόνο κρυπτογράφησης σε λειτουργία ενθυλάκωσης ανάμεσα σε ένα ζευγάρι πυλών ασφαλείας G_A και G_B. Αυτές οι πύλες παρέχουν ασφάλεια για τα επικοινωνούντα μέρη H_A και H_B που βρίσκονται πίσω από τις πύλες, όπως φαίνεται στο παρακάτω σχήμα. Αυτή είναι μια τυπική ρύθμιση για εικονικά δίκτυα VPN σε IPsec.
β) Το κλειδί που χρησιμοποιείται για την κρυπτογράφηση ESP της κίνησης πακέτων IP από το G_A στο G_B είναι αμετάβλητο κατά τη διάρκεια της επίθεσης.
γ) Ο επιτιθέμενος μπορεί να παρακολουθήσει και να καταγράψει τα πακέτα ESP που διακινούνται μεταξύ των δύο πυλών ασφαλείας και
δ) Ο επιτιθέμενος μπορεί να εισάγει τροποποιημένα πακέτα στο δίκτυο ανάμεσα στο G_A και το G_B.



Σχήμα 17. Διάταξη δικτύου στις επιθέσεις στο ESP σε λειτουργία ενθυλάκωσης

Εδώ η απαίτηση για τη δυνατότητα παρακολούθησης της κίνησης των πακέτων είναι πιο ρεαλιστική από ότι στις επιθέσεις που περιγράφηκαν στο προηγούμενο τμήμα 2.6.1 στην υλοποίηση του IPsec στο Linux [04], όπου απαιτούνταν η ικανότητα παρακολούθησης όλης της κίνησης από την πύλη ασφαλείας.

Υποθέτουμε ότι κατάλληλοι συσχετισμοί ασφαλείας έχουν εδραιωθεί και προστατεύουν την κίνηση από το G_A στο G_B και αντίστροφα. Όπως αναφέρθηκε και πριν, ειδικοί συσχετισμοί ασφαλείας μπορεί να χρησιμοποιούνται ώστε να προστατεύουν, σε αυτή την περίπτωση, τα μηνύματα ICMP από το G_A στο G_B και αντίστροφα. Η επίθεση μπορεί να εφαρμοστεί, ανεξάρτητα αν χρησιμοποιούνται οι ίδιοι συσχετισμοί ασφαλείας για να προστατέψουν την κίνηση ICMP και μη ICMP πακέτων.

Υποθέτουμε, επίσης, ότι ο επιτιθέμενος έχει συλλέξει ένα κρυπτοκείμενο $C=C_0,C_1,..,C_q$ από τα δεδομένα ενός εξωτερικού πακέτου IP που δρομολογήθηκε στο G_B. Αυτό για παράδειγμα, θα μπορούσε να αντιπροσωπεύει κίνηση από το H_A στο H_B.

Ακόμη, υποθέτουμε ότι ο επιτιθέμενος έχει στην κατοχή του τα κρυπτοκείμενα C^j , που αντιστοιχούν σε ένα σύνολο 7 επιλεγμένων απλών κειμένων P^j ($0 \leq j \leq$

6). Εδώ το P^j έχει επιλεγεί ώστε να περιέχει ένα εσωτερικό πακέτο IP με διεύθυνση αποστολέα H_A, διεύθυνση αποστολής H_B και τιμή TTL ίση με 1. Επιπλέον, υποθέτουμε ότι το P^j περιέχει ακριβώς j+12 bytes δεδομένων ή ένα σύνολο j+32 bytes. Το απλό κείμενο P^j περιέχει, επίσης, τα συμπληρωματικά (padding) bytes και τα πεδία PL και NH. Το P^j έχει j bytes δεδομένων στο τελευταίο block, έτσι, δεδομένου του κανόνα των συμπλήρωσης με bytes, αυτά θα ακολουθούνται από S-j-2 συμπληρωματικά bytes, το PL και στο τέλος το NH.

Για παράδειγμα, το P^6 τελειώνει με την ακολουθία από bytes 0,4, ενώ το P^0 με την ακολουθία 1,2,3,4,5,6,4, η οποία γεμίζει ολόκληρο το τελευταίο block. Έτσι, το P^j έχει 5 blocks και το αντίστοιχο κρυπτοκείμενο C^j έχει 6 blocks (συμπεριλαμβανομένου και του διανύσματος IV).

Εάν το C^j εισαχθεί στο δίκτυο ως δεδομένα ενός εξωτερικού πακέτου με διεύθυνση αποστολής G_B, τότε το αντίστοιχο εσωτερικό πλαίσιο δεδομένων θα ανακτηθεί στο G_B και θα προωθηθεί στην υλοποίηση του IP στο G_B. Αυτό θα μειώσει την τιμή του TTL του εσωτερικού πακέτου, κάνοντάς την ίση με μηδέν. Η υλοποίηση του IP στο G_B, θα δημιουργήσει τότε, ένα μήνυμα σφάλματος ICMP (τύπου 11 και κώδικα 0), δηλώνοντας το γεγονός ότι έχει ξεπεραστεί ο χρόνος ζωής του πακέτου (time to live exceeded). Υποθέτοντας ότι η υλοποίηση IP ακολουθεί το πρότυπο και περιλαμβάνει την επικεφαλίδα και τα πρώτα 64 bits των δεδομένων του αρχικού πακέτου, το συνολικό μήκος του μηνύματος ICMP (συμπεριλαμβανομένης της επικεφαλίδας IP με την οποία μεταφέρεται) θα είναι 56 bytes. Το μήνυμα σφάλματος θα έχει ως διεύθυνση αποστολής το H_A, και έτσι θα γίνει ένα κρυπτογραφημένο με ESP πακέτο στο G_B. Χρησιμοποιώντας τον κανόνα συμπλήρωσης με bytes, θα προστεθεί ένα επιπλέον block με συμπληρωματικά bytes και το NH byte και το τελικό κρυπτοκείμενο θα αποτελείται από 9 blocks (συμπεριλαμβανομένου του IV). Έτσι, ένα εξωτερικό πακέτο που μεταφέρει το κρυπτοκείμενο των 9 blocks, θα παρατηρηθεί να κινείται στην αντίθετη κατεύθυνση του καναλιού, λίγο μετά την εισαγωγή του C^j. Στην πράξη αυτό το μήνυμα ICMP μπορεί να ανιχνευτεί από τον επιτιθέμενο με βάση το μήκος του, παρόλο που μεταφέρεται μέσα σε ένα πλαίσιο δεδομένων που έχει κρυπτογραφηθεί από το IPsec.

Εάν χρησιμοποιούνται στο κανάλι συμπληρωματικά bytes TFC ή μεταβλητού μήκους, τότε το σχετικό μήνυμα ICMP μπορεί να είναι πιο δύσκολο να διακριθεί από την υπόλοιπη κρυπτογραφημένη κίνηση δεδομένων. Στην περίπτωση αυτή, μπορούμε να ανιχνεύσουμε συσχετίσεις ανάμεσα στον χρόνο εισαγωγής των δοκιμαστικών δεδομένων και την εμφάνιση κρυπτογραφημένων μηνυμάτων στην αντίθετη κατεύθυνση του καναλιού. Αυτό θα είναι πιο φανερό, εάν η κίνηση στην αντίθετη κατεύθυνση είναι σχετικά μικρή. Εάν όχι, τότε υπάρχουν πιθανότητες σφάλματος και για να αυξηθεί η ακρίβεια, θα πρέπει να γίνουν πολλαπλές προσπάθειες για κάθε δοκιμαστικό πακέτο που εισάγεται στο δίκτυο. Υποθέτουμε στην από εδώ και πέρα ανάλυση, ότι αυτή η τεχνική εφαρμόζεται στην περίπτωση ανίχνευσης με βάση το μήκος του πακέτου, όταν χρησιμοποιούνται μέτρα TFC.

Ο επιτιθέμενος θα χρησιμοποιήσει τροποποιήσεις των κρυπτοκειμένων C^j στην επίθεσή του. Στη συνέχεια, η ακολουθία C^j_0, C^j_1,…,C^j_5 θα δηλώνει τα blocks του C^j (το C^j_0 είναι το IV).

Ο επιτιθέμενος επιλέγει ένα block C_i, $i \geq 1$, από την ακολουθία κρυπτοκειμένου. Θεωρούμε το αποτέλεσμα της εισαγωγής ενός εξωτερικού πακέτου που περιέχει ως δεδομένα το παρακάτω κρυπτοκείμενο:
$C' = C^6_0, C^6_1, C^6_2, C^6_3, R^6, C_i$, όπου το R^6 είναι ένα τυχαίο block.

Η εισαγωγή του R^6 δεν επηρεάζει την εσωτερική επικεφαλίδα στο αντίστοιχο απλό κείμενο P^j. Έτσι, το πρώτο τμήμα του P^j θα αντιστοιχεί ακόμα σε ένα πλαίσιο δεδομένων, που μεταφέρει δεδομένα ενός πρωτοκόλλου ανώτερου επιπέδου που έχουν προορισμό το H_B. Ωστόσο, τα δεδομένα, τα συμπληρωματικά bytes, το PL και το NH θα επηρεαστούν. Σύμφωνα με τη διαδικασία αποκρυπτογράφησης, αφαίρεσης των συμπληρωματικών bytes του ESP και επεξεργασίας του πεδίου NH, ξέρουμε ότι η πύλη G_B θα απορρίψει το εσωτερικό πακέτο, εκτός και αν τα συμπληρωματικά bytes είναι έγκυρα και το byte NH είναι ίσο με 4. Με το R^6 να είναι τυχαίο, το πιο πιθανό σχήμα συμπληρωματικών bytes είναι αυτό με μήκος μηδέν, με το πεδίο PL να περιέχει την τιμή 0. Όταν αυτό ισχύει και το byte NH είναι 4, η αποκρυπτογράφηση και η αφαίρεση των συμπληρωματικών bytes του C′ παράγει ένα έγκυρο πακέτο IP, παρόλο που δεδομένα του έχουν αλλάξει από το C^6. Τα πεδία length και checksum στην επικεφαλίδα IP θα είναι ακόμα

σωστά και η τιμή του TTL θα είναι 1. Έτσι, σε αυτή την περίπτωση, το εσωτερικό πακέτο θα προωθηθεί στην υλοποίηση του IP στην G_B και θα λάβουμε ένα μήνυμα σφάλματος ICMP προς την αντίθετη κατεύθυνση του καναλιού. Αυτό το μήνυμα θα είναι κρυπτογραφημένο σε ένα κρυπτοκείμενο 9 block και ο επιτιθέμενος μπορεί να το ανιχνεύσει από το μήκος.

Είναι τώρα προφανές ότι, ο επιτιθέμενος πρέπει να μεταβάλει συστηματικά τα bytes 6 και 7 του R^6, να εισάγει τα τροποποιημένα κρυπτοκείμενα και να ανιχνεύει την εμφάνιση του χαρακτηριστικού πακέτου 9 block προς την αντίθετη κατεύθυνση του καναλιού. Όταν αυτό ανιχνευτεί, ο επιτιθέμενος ξέρει ότι τα τελευταία 2 bytes του $R^6(+)d_K(C_i)$ είναι 0 και 4 με μεγάλη πιθανότητα. Ένα επιπλέον τεστ, τροποποιώντας το byte R^6_5, μπορεί να ελαχιστοποιήσει τη μικρή πιθανότητα όπου τα συμπληρωματικά bytes είναι έγκυρα, αλλά το byte PL δεν είναι 0. Από αυτά τα δύο bytes είναι εύκολο να εξαχθούν τα bytes $P_{i,6}$, $P_{i,7}$. Ο μέγιστος αριθμός προσπαθειών εισαγωγής πακέτων στο δίκτυο είναι 2^{16} και κατά μέσο όρο 2^{15}.

Βλέπουμε ότι, χρησιμοποιώντας επιλεγμένα απλά κείμενα, έχουμε εδραιώσει έναν μηχανισμό ανίχνευσης της ουράς του πλαισίου δεδομένων ESP, που μπορεί να χρησιμοποιηθεί για να εξαχθούν τα bytes 6 και 7 του P_i ταυτόχρονα. Τα υπόλοιπα bytes του P_i μπορούν να εξαχθούν χρησιμοποιώντας τον μηχανισμό που περιγράφεται στη συνέχεια.

Ο επιτιθέμενος μπορεί να συνεχίσει να εξάγει τα υπόλοιπα bytes πιο αποτελεσματικά, δουλεύοντας σε σειρά από τα δεξιά ($P_{i,5}$) προς τα αριστερά ($P_{i,0}$). Αυτή τη φορά χρησιμοποιεί ένα κρυπτοκείμενο της μορφής :

$C' = C^5_0, C^5_1, C^5_2, C^5_3, R^5, C_i$

όπου το R^5 αρχικά ρυθμίζεται στην τιμή του R^6, που παρήγαγε ένα μήνυμα ICMP όταν εξαγάγαμε τα δύο δεξιότερα bytes, εκτός από ότι θέτουμε το $R^5_6 = R^6_6 \oplus 1$. Αυτό εξασφαλίζει ότι το P′, που είναι το αποκρυπτογραφημένο κείμενο του C′ χωρίς τα συμπληρωματικά bytes, τελειώνει με τα bytes 1,4. Τώρα, ο επιτιθέμενος συστηματικά μεταβάλει το 5^o byte του R^5 και εισάγει στο δίκτυο το τροποποιημένο κρυπτοκείμενο. Μόνο όταν το P′ τελειώνει σε 1,1,4 θα παρατηρήσουμε ένα κρυπτοκείμενο 9 block προς την αντίθετη κατεύθυνση.

Όταν αυτό ανιχνευτεί, ο επιτιθέμενος ξέρει ότι το 5^{ο} byte του $R^6 \oplus d_K(C_i)$ είναι ίσο με 1. Από αυτό, μπορεί να εξάγει την τιμή του $P_{i,5}$.

Συνεχίζοντας με αυτό τον τρόπο, είναι τώρα εύκολο να διαπιστώσει κανείς πώς μπορούν να εξαχθούν όλα τα bytes του P_i, με μέσο όρο 128 προσπαθειών και κατά μέγιστο 256 προσπαθειών ανά byte, χρησιμοποιώντας τροποποιημένες εκδόσεις του C^j και επεκτείνοντας τον αριθμό των συμπληρωματικών bytes κατά ένα την φορά.

Η πολυπλοκότητα της συνολικής επίθεσης είναι το πολύ $2^{16} + 6 \bullet 2^8$ ανά κρυπτοκείμενο και κατά μέσο όρο το μισό αυτού. Η κύρια απαίτηση εδώ προκύπτει από την ανάγκη να εξαχθούν τα δύο δεξιότερα bytes ταυτόχρονα. Μια τέτοια απαίτηση υπάρχει σε όλες τις επιθέσεις αυτού του τύπου. Δεν έχει βρεθεί ένας τρόπος για να μειωθεί η απαίτηση για δύο bytes, εκτός από επιθέσεις σε συγκεκριμένες υλοποιήσεις του IPsec, όπου γίνονται ασθενέστεροι έλεγχοι όσον αφορά στα συμπληρωματικά bytes.

Η επίθεση δουλεύει ακόμα και αν η υλοποίηση IP του G_B διενεργεί αυστηρό έλεγχο μήκους. Σε κάθε στάδιο το εσωτερικό πακέτο που παράγεται μετά από επιτυχή αποκρυπτογράφηση και αφαίρεση των συμπληρωματικών bytes, έχει ακριβώς τον σωστό αριθμό από bytes, που δείχνει το πεδίο Total Length. Εάν εφαρμόζεται χαλαρός έλεγχος, τότε μπορούμε να μειώσουμε τον αριθμό από τα επιλεγμένα απλά κείμενα που χρειαζόμαστε από 7 σε 1, δουλεύοντας μόνο με το απλό κείμενο P^o. Αυτό το πλαίσιο έχει τιμή στο πεδίο Total Length, η οποία δείχνει 12 bytes δεδομένων μετά την επικεφαλίδα IP και τελειώνει με 7 συμπληρωματικά bytes και το byte NH. Τώρα οι τροποποιήσεις στα τελευταία δύο block του C^0 που γίνονται στην επίθεση ανίχνευσης του αριθμού των συμπληρωματικών bytes, παράγουν είτε απλά κείμενα με λάθος αριθμό συμπληρωματικών bytes, είτε με σωστό αριθμό, αλλά στα οποία υπάρχουν παραπάνω bytes στο εσωτερικό πλαίσιο απ' ότι δείχνει το πεδίο Total Length. Αυτά τα απλά κείμενα περνάνε τον χαλαρό έλεγχο μήκους.

Μπορεί ο αναγνώστης να αναρωτηθεί γιατί δεν χρησιμοποιήθηκαν απλά τα μηνύματα τύπου echo request ως επιλεγμένα απλά κείμενα, με το σκεπτικό ότι οι σωστές τιμές του NH και των συμπληρωματικών bytes θα οδηγούσαν στη δημιουργία ενός μηνύματος απάντησης ICMP στην αντίστροφη κατεύθυνση

του καναλιού. Αυτό θα ήταν μια καλή ιδέα, όμως δεν δουλεύει για τον λόγο ότι τα μηνύματα ICMP echo request περιλαμβάνουν μια τιμή checksum η οποία καλύπτει ολόκληρο το τμήμα του μηνύματος και θα αλλοιώνονταν με την εισαγωγή τυχαίων bytes κατά τη διάρκεια της επίθεσης. Έτσι, η τιμή checksum θα ήταν σχεδόν πάντα μη έγκυρη και τα μηνύματα ICMP θα απορρίπτονταν πριν δημιουργηθεί το μήνυμα απάντησης.

2.6.2.3 Επιθέσεις που βασίζονται στον τρόπο επεξεργασίας του πεδίου Options από το IP

Η επίθεση που περιγράφηκε πριν, εισάγει την κύρια ιδέα που χρησιμοποιείται στη συνέχεια, ώστε από επίθεση επιλεγμένου απλού κειμένου να μετασχηματιστεί σε επίθεση μόνο κρυπτοκειμένου. Στη συνέχεια, παρουσιάζεται μια επίθεση μόνο κρυπτοκειμένου, εναντίον του ESP σε λειτουργία μόνο κρυπτογράφησης, η οποία βασίζεται στην επεξεργασία του πεδίου Options. Εστιάζεται στην περίπτωση των 64 bit, ενώ στο τέλος περιγράφεται συνοπτικά και η περίπτωση των 128 bit. Γίνεται η υπόθεση ότι πραγματοποιείται χαλαρός έλεγχος μήκους από το IP, ενώ περιγράφεται αργότερα η περίπτωση του αυστηρού ελέγχου. Κατά τα λοιπά, ισχύουν οι ίδιες υποθέσεις λειτουργίας, όπως και στην προηγούμενη επίθεση.

Η βασική ιδέα είναι να μετασχηματιστεί ένα υπάρχον κρυπτοκείμενο σε ένα κρυπτοκείμενο το οποίο φέρει ένα εσωτερικό πακέτο, το οποίο παράγει ένα μήνυμα ICMP εξαιτίας σφάλματος στην επεξεργασία των επιλογών (options). Το μετασχηματισμένο κρυπτοκείμενο θα χρησιμοποιηθεί ακολούθως σε μια επίθεση ανίχνευσης αριθμού συμπληρωματικών bytes, όπως και η προηγούμενη. Υποθέτουμε ότι ο επιτιθέμενος έχει συλλέξει έναν αριθμό από κρυπτοκείμενα της μορφής $C = C_o, C_1, C_2,\ldots, C_q$ από το πλαίσιο δεδομένων ενός εξωτερικού πακέτου IP, που κατευθύνεται στην πύλη G_B. Αυτό για παράδειγμα μπορεί να αντιπροσωπεύει κίνηση από το H_A στο H_B.

Έστω ότι C′ δηλώνει τον στόχο για αποκρυπτογράφηση ενός από τα κρυπτοκείμενα. Η φάση προετοιμασίας θα γίνει μια φορά στο C′, μετά την οποία κάθε block εντός του διανύσματος αρχικοποίησης IV μπορεί να αποκρυπτογραφηθεί αποτελεσματικά. Είναι βολικό στην πράξη να επιλεχθεί όσο το δυνατόν μικρότερο C′. Έστω ότι τα block του C′ είναι $C'_0, C'_1,...,C'_r,$

όπου γνωρίζουμε ότι το r ≥ 3 (επειδή το C′ πρέπει να περιλαμβάνει την επικεφαλίδα του εσωτερικού πακέτου, που καταλαμβάνει τουλάχιστον 3 block). Η φάση προετοιμασίας μπορεί να κωδικοποιηθεί ως ακολούθως:

1. Τροποποίησε το block C'_0, αλλάζοντας τα bit 6 και 7, δημιουργώντας ένα νέο block C''_0.
2. Θέσε την τιμή του μετρητή i στο 0.
3. Επανέλαβε :
   α) Τροποποίησε το block C''_0 , έτσι ώστε τα bytes 4 και 5 να περιέχουν την δυαδική αναπαράσταση του i. Έστω C^{+}_0 δηλώνει το τροποποιημένο block.
   β) Προετοίμασε ένα εξωτερικό πακέτο προς το G_B με πλαίσιο δεδομένων ίδιο με το C′, εκτός από ότι το C'_0 έχει αντικατασταθεί με το C^{+}_0. Εισήγαγε αυτό το τροποποιημένο πακέτο στο δίκτυο.
   γ) Αύξησε το i.

μέχρι ένα κρυπτοκείμενο που παραπέμπει σε μήνυμα ICMP να εμφανιστεί στο κανάλι από το G_B στο G_A.

Ξεκινάμε αλλάζοντας τα bit 6 και 7 του C'_0, δηλαδή το διάνυσμα αρχικοποίησης IV. Αυτό έχει ως αποτέλεσμα την αλλαγή της τιμής του πεδίου IHL της εσωτερικής επικεφαλίδας από τα bit 5 έως 6, δείχνοντας ότι υπάρχει μια λέξη 32 bit στο πεδίο Options. Ο υπολογισμός της τιμής checksum από το IP λαμβάνει υπόψη του όλα τα bytes της επικεφαλίδας, περιλαμβανομένων και αυτών του πεδίου Options, έτσι κατά πάσα πιθανότητα θα έχει λάθος τιμή. Μεταβάλλοντας συστηματικά τα bytes 4 και 5 του C'_0 μέσα στο βρόγχο επανάληψης, τα θέτουμε σε όλες τις πιθανές τιμές που είναι 2^{16}. Αυτό έχει ως αποτέλεσμα να αλλάξει το πεδίο Identification στο εσωτερικό διάγραμμα, όπως είχε αναφερθεί και πριν. Εισάγουμε τα τροποποιημένα αυτά πακέτα στο δίκτυο, ως πλαίσια δεδομένων εξωτερικών πακέτων που κατευθύνονται στο G_B. Μετά την αποκρυπτογράφηση και την αφαίρεση των συμπληρωματικών bytes, τα διαμορφούμενα εσωτερικά πακέτα θα προωθηθούν στην υλοποίηση IP του G_B για περαιτέρω επεξεργασία. Εξαιτίας της τροποποίησης του πεδίου Identification, μόνο ένα από αυτά θα έχει σωστή τιμή checksum. Επίσης, θα περάσει όλους τους σχετικούς ελέγχους μήκους πακέτου, είτε αυτοί είναι αυστηροί, είτε είναι χαλαροί. Εν συνεχεία, θα γίνει επεξεργασία του πεδίου Options, με τα πρώτα 4 bytes που ακολουθούν την αρχική επικεφαλίδα να

λαμβάνονται τώρα ως bytes του πεδίου Options. Αυτά τα bytes θα έχουν σίγουρα λάθος δομή και θα οδηγήσουν στη δημιουργία ενός μηνύματος ICMP τύπου 12 (προβλήματος παραμέτρων). Αυτό το μήνυμα θα σταλεί στην αντίθετη κατεύθυνση του καναλιού στο H_A και θα έχει χαρακτηριστικό μήκος 9 block (υποθέτοντας ότι έχει τηρηθεί το σχετικό κείμενο RFC του ICMP και το εσωτερικό πακέτο έχει τουλάχιστον 64 bit δεδομένων).

Η πιθανότητα δημιουργίας ενός μηνύματος ICMP τύπου 12, κατά τη λήψη πακέτου με λάθος στο πεδίο Options είναι για την υλοποίηση του Linux 0,985. Έτσι, μπορεί να είμαστε απόλυτα σίγουροι ότι η επίθεση θα είναι επιτυχής μέσα σε 2^{16} και κατά μέσο όρο σε 2^{15} προσπάθειες. Στην με πολύ μικρή πιθανότητα περίπτωση που το πεδίο Options έχει σωστό περιεχόμενο, μπορεί να δοκιμαστεί ένα νέο κρυπτοκείμενο με την επανεκκίνηση του σταδίου αυτού.

Στο τέλος του σταδίου προετοιμασίας ο επιτιθέμενος έχει στην κατοχή του ένα κρυπτοκείμενο C^* με r+1 blocks που γνωρίζει ότι θα οδηγούν πάντα στη δημιουργία ενός μηνύματος ICMP, όταν αυτό αποκρυπτογραφηθεί, αφαιρεθούν τα συμπληρωματικά bytes και προωθηθεί στο πρωτόκολλο IP.

Ο επιτιθέμενος είναι τώρα έτοιμος να χρησιμοποιήσει τα r+1 block κρυπτοκειμένου C^* που δηλώνονται ως C^*_0, C^*_1, …..,C^*_r για να επιτεθεί σε έναν επιλεγμένο στόχο κρυπτοκειμένου το C_i. Ακολουθεί παρόμοιο τρόπο με αυτόν της επίθεσης επιλεγμένου απλού κειμένου. Εισάγει στο δίκτυο ένα εξωτερικό πακέτο που περιέχει ως δεδομένα τα παρακάτω r+3 block κρυπτοκειμένου:

$C^{\#} = C^*_0, C^*_1, C^*_2,\ldots,C^*_r, R^6, C_i$

όπου το R^6 είναι ένα τυχαίο block και το C_i ο στόχος της επίθεσης. Το απλό κείμενο που αντιστοιχεί σε αυτά τα block, ερμηνεύεται ως τα τελευταία bytes του πλαισίου δεδομένων, τα συμπληρωματικά bytes και τα PL και NH. Γνωρίζουμε ότι η πύλη ασφαλείας G_B θα απορρίψει το απλό κείμενο, εκτός και αν ο αριθμός των συμπληρωματικών bytes είναι έγκυρος και το byte NH έχει τιμή 4. Το πιο πιθανό έγκυρο σχήμα συμπληρωματικών bytes είναι αυτό με μήκος 0 και τιμή PL ίση με 0. Όταν το PL είναι 0 και το NH ίσο με 4, η αποκρυπτογράφηση και η αφαίρεση των συμπληρωματικών bytes του $C^{\#}$

παράγει ένα εσωτερικό πακέτο που προωθείται στην υλοποίηση IP του G_B. Η εισαγωγή του R^6 και C^i δεν επηρεάζει το εσωτερικό πακέτο. Έτσι, θα έχουμε ένα εσωτερικό πακέτο που θα έχει μια σωστή τιμή checksum της επικεφαλίδας και αρκετό αριθμό από bytes, ώστε να περάσει το χαλαρό έλεγχο μήκους που διενεργούνται από το IP. Βλέπουμε ότι τα παλιά συμπληρωματικά bytes, καθώς και τα PL και NH που κρυπτογραφούνται στα blocks C^*_{r-1}, C^*_r, θεωρούνται τώρα ως περιττά από την επεξεργασία IP, καθόσον είναι πέρα από το τελευταίο byte που υποδεικνύει το πεδίο Total Length της εσωτερικής επικεφαλίδας. Έτσι, αποκόπτονται από το IP. Εξαιτίας της τιμής του πεδίου IHL, το εσωτερικό πακέτο προκαλεί την επεξεργασία του πεδίου Options στο G_B. Και εξαιτίας του περιεχομένου του πεδίου Options, δημιουργείται ένα μήνυμα ICMP στην αντίθετη κατεύθυνση του καναλιού. Ο επιτιθέμενος μπορεί να το ανιχνεύσει, βάσει του μήκους του.

Ως συνήθως, ο επιτιθέμενος συστηματικά αλλάζει τα bytes 6 και 7 του R^6, εισάγει τα τροποποιημένα κρυπτοκείμενα και περιμένει για το κρυπτοκείμενο που μεταφέρει το μήνυμα ICMP στην αντίθετη κατεύθυνση. Όταν αυτό ανιχνευτεί, ο επιτιθέμενος γνωρίζει ότι τα τελευταία 2 bytes του $R^6 \oplus d_K(C_i)$ είναι ίσα με 0,4 με μεγάλη πιθανότητα. Ένα επιπρόσθετο απλό τεστ που τροποποιεί το $R^5{}_6$, μπορεί να χρησιμοποιηθεί στην με μικρή πιθανότητα περίπτωση όπου τα συμπληρωματικά bytes είναι έγκυρα, αλλά το byte PL δεν έχει τιμή 0. Από αυτό είναι εύκολο να εξάγει κανείς τα bytes 6 και 7 του απλού κειμένου P_i. Ο μέγιστος αριθμός προσπαθειών είναι 2^{16}, με μέσο όρο 2^{15}.

Ο επιτιθέμενος μπορεί να συνεχίσει για να εξάγει τα υπόλοιπα bytes, περίπου όπως και στην προηγούμενη επίθεση τύπου επιλεγμένου γνωστού κειμένου. Ο επιτιθέμενος τώρα χρησιμοποιεί ένα κρυπτοκείμενο της μορφής:

$C^{\#} = C^*_0, C^*_1, C^*_2, \ldots..,C^*_r, R^5, C_i$

όπου το R^5 είναι ίσο με την τιμή του R^6, που παρήγαγε πριν ένα μήνυμα ICMP, εκτός από ότι θέτουμε $R^5{}_6 = R^6{}_6 + 1$. Αυτό διασφαλίζει ότι το $P^{\#}$, το αποκρυπτογραφημένο απλό κείμενο μετά την αφαίρεση των συμπληρωματικών bytes, τελειώνει με τα bytes 1,4. Τώρα ο επιτιθέμενος αλλάζει συστηματικά το byte 5 του R^5 και εισάγει τα τροποποιημένα κρυπτοκείμενα. Μόνο όταν το P^i τελειώνει σε 1,1,4, θα παρατηρήσουμε ένα

κρυπτοκείμενο που παραπέμπει σε μήνυμα ICMP στην αντίθετη διεύθυνση του καναλιού. Όταν αυτό ανιχνευτεί, ο επιτιθέμενος γνωρίζει ότι το byte 5 του $R^5 \oplus d_K(C_i)$ είναι ίσο με 1. Από αυτό μπορεί να εξάγει το byte 5 του P_i. Συνεχίζοντας με αυτόν τον τρόπο, είναι εύκολο για κάποιον να διαπιστώσει τον τρόπο με τον οποίο ο επιτιθέμενος μπορεί να εξάγει όλα τα byte του P_i, δαπανώντας κατά μέσο όρο 128 και μέγιστο 256 προσπάθειες ανά byte.

Η πολυπλοκότητα της φάσης προετοιμασίας είναι κατά μέσο όρο λίγο παραπάνω από 2^{15} προσπάθειες. Το αντίστοιχο κόστος ανά block είναι περίπου της ίδιας κατά μέσο όρο πολυπλοκότητας. Μια παραλλαγή αυτής της επίθεσης μπορεί, επίσης, να γίνει στην περίπτωση που το μέγεθος του block είναι 128 bit. Και εδώ η μέση πολυπλοκότητα είναι ίδια με αυτή των 64 bit.

Εάν εφαρμόζεται αυστηρός έλεγχος μήκους από την υλοποίηση IP, τότε χρειάζεται ένα πιο πολύπλοκο στάδιο προετοιμασίας. Στο στάδιο αυτό έχουμε ως στόχο να παράγουμε από το C^* ένα σετ κρυπτοκειμένων που δημιουργούν ICMP μηνύματα (C^{*0}, C^{*1}, …. , C^{*6}) στα οποία το C^{*j} ενθυλακώνει ένα εσωτερικό πακέτο του οποίου το μήκος είναι ίσο με j modulo 8 και το οποίο περιέχει ακριβώς τη σωστή δομή των συμπληρωματικών bytes (χωρίς περιττά bytes, μετά την επεξεργασία των δεδομένων από το IP). Αυτό το σετ κρυπτοκειμένων μπορεί να δημιουργηθεί σε δύο στάδια. Πρώτα βρίσκουμε την έκταση των συμπληρωματικών bytes στο C^* τροποποιώντας τα bytes από αριστερά προς τα δεξιά, αρχίζοντας από το προτελευταίο block του C^*. Εάν η δομή των συμπληρωματικών bytes είναι μη έγκυρη, τότε το πακέτο θα απορριφθεί από το IPsec. Από την άλλη μεριά, εάν η τροποποίηση γίνει στα bytes δεδομένων που είναι αριστερά των συμπληρωματικών, τότε η δομή των συμπληρωματικών θα είναι έγκυρη και θα δημιουργηθεί το μήνυμα ICMP. Τότε ταυτόχρονα ρυθμίζουμε το πεδίο Total Length, Header Checksum (και τα δύο αυτά αλλάζοντας το IV του C^*) και τα συμπληρωματικά bytes (τροποποιώντας το προτελευταίο block στο C^*). Ο πιο αποτελεσματικός τρόπος να προχωρήσει κάποιος, είναι με μια επαναληπτική διαδικασία, στην οποία το πεδίο Total Length μειώνεται κατά ένα και το σχήμα συμπλήρωσης εκτείνεται κατά ένα byte σε κάθε επανάληψη. Στο σημείο αυτό παραλείπονται οι λεπτομέρειες και η ανάλυση πολυπλοκότητας. Εάν εφαρμόζονται αυστηροί έλεγχοι μήκους, πρέπει να είμαστε λίγο πιο προσεκτικοί στην εξαγωγή του απλού κειμένου από το κρυπτοκείμενο. Δουλεύουμε αρχικά με παραλλαγές του C^{*6}, όταν εξάγουμε

τα bytes 6 και 7, και στη συνέχεια με τα κρυπτοκείμενα C^{*5} έως C^{*0} όταν επιτιθέμεθα τα άλλα bytes. Αυτά τα κρυπτοκείμενα περιέχουν εσωτερικά πακέτα των οποίων τα πεδία Total Length έχουν τις κατάλληλες τιμές για να εξάγουν τα bytes απλού κειμένου.

2.6.2.4 Επιθέσεις που βασίζονται στο πεδίο Protocol της επικεφαλίδας IP

Παρόμοιες ιδέες που αναπτύχθηκαν στο τμήμα 2.6.2.3, μπορούν να εφαρμοστούν για να αναπτυχθεί μια επίθεση η οποία δουλεύει στην περίπτωση των 128 bit, αλλάζοντας το πεδίο Protocol του εσωτερικού πακέτου [05]. Η κύρια ιδέα είναι ότι όταν ο H_B λάβει ένα πακέτο το οποίο έχει τιμή στο πεδίο Protocol που δεν υποστηρίζεται στον H_B, θα παραχθεί ένα μήνυμα ICMP. Καθόσον οι γνωστοί αριθμοί του πρωτοκόλλου που έχουν εκχωρηθεί είναι από 138 έως 252, η αλλαγή των δύο πιο σημαντικών bit του πεδίου Protocol θα οδηγεί πάντα στη δημιουργία ενός μηνύματος ICMP.

Συμπερασματικά, η πολυπλοκότητα του IPsec δεν βοηθάει τις ομάδες υλοποίησής του, καθόσον σημαντικοί έλεγχοι που αφορούν την ασφάλεια του ESP μπορεί να περιέχονται σε κάποιο άλλο κείμενο RFC μαζί με άλλες σχετικές πληροφορίες. Αυτό έχει ως συνέπεια, τη δυσκολία υιοθέτησης όλων των απαιτήσεων ασφαλείας από την ομάδα υλοποίησης του πρωτοκόλλου, η οποία θα πρέπει να εξετάσει έναν μεγάλο αριθμό από κείμενα RFC και να συνδυάσει τις πληροφορίες που περιέχονται σε αυτά.

Είναι ανησυχητικό, επίσης, ότι η ασφάλεια της λειτουργίας μόνον κρυπτογράφησης συναρτάται αποκλειστικά στη διενέργεια των ελέγχων που αφορούν στις επιλογές ρυθμίσεων της συσχέτισης ασφαλείας (SA), που περιγράφηκαν στο τμήμα 2.6.1. Η ασφάλεια έτσι κρέμεται από μία κλωστή, όπως αυτό γίνεται φανερό, από επιθέσεις σε συγκεκριμένη υλοποίηση του πρωτοκόλλου στο Linux [04]. Στο σημείο αυτό θα βοηθούσε, εάν ο λόγος διενέργειας των ελέγχων εξηγούνταν απολύτως στο πρότυπο. Αυτό θα έδινε σε αυτόν που υλοποιεί το πρότυπο ένα ισχυρότερο κίνητρο να κάνει την υλοποίηση σωστά.

Δυστυχώς, η IETF δεν άδραξε την ευκαιρία να αποκλείσει τη χρήση λειτουργίας μόνον κρυπτογράφησης από την επόμενη γενιά εκδόσεων του

πρωτοκόλλου. Αντιθέτως, η ασφάλεια του πρωτοκόλλου επαφίεται στη γνώση αυτών που υλοποιούν το πρότυπο, σχετικά με το πώς θα διαχειριστούν τις προειδοποιήσεις των κειμένων RFC, καθώς και των τελικών χρηστών οι οποίοι θα επιλέξουν τις σωστές ρυθμίσεις κατά τη λειτουργία του. Είναι φανερό όμως ότι ο κίνδυνος να επιλέξουν μη ασφαλείς ρυθμίσεις είναι υπαρκτός. Μια πιο συντηρητική προσέγγιση, η οποία θα ήταν κατάλληλη σε ένα πρότυπο ασφαλείας, θα μπορούσε να είχε οδηγήσει σε μεγάλες αλλαγές μετά τη δημοσιοποίηση των επιθέσεων του Bollovin. Όμως η ανάγκη συμβατότητας με παλαιότερες εκδόσεις υπερίσχυσε των ανησυχιών για την ασφάλεια. Είναι μάλιστα αξιοσημείωτο το γεγονός ότι, η λύση της διενέργειας ελέγχων του γεμίσματος με συμπληρωματικά bytes (padding) που πρότεινε η IETF, δεν διασφαλίζει τελικά τη συνολική ασφάλεια, όπως προκύπτει από την εργασία των Degabriele και Paterson [05].

Με την εμφάνιση αυτών των νέων μορφών επιθέσεων [05], μερικές από τις συμβουλές προς τις ομάδες υλοποίησης του πρωτοκόλλου, σχετικά με την χρησιμότητα του ESP σε ρυθμίσεις μόνον κρυπτογράφησης, όπως αποδείχτηκε τώρα είναι αβάσιμες. Για παράδειγμα, η ιδέα ότι μπορεί να εφαρμοστεί ανεξάρτητα και σε υψηλότερα επίπεδα η αυθεντικοποίηση είναι εσφαλμένη, καθόσον οι γνωστές επιθέσεις δουλεύουν ακόμα και όταν εφαρμόζεται προστασία σε ανώτερα επίπεδα ή παρέχεται από το AH προστασία ακεραιότητας δεδομένων (integrity).

2.7. Εξέλιξη του IPsec

Η ομάδα εργασίας του IPsec είναι σε ετοιμότητα μετά την ολοκλήρωση των προσπαθειών για την ανάπτυξη της νέας γενιάς των προτύπων του πρωτοκόλλου [03]. Η δραστηριότητα της IETF είναι πιθανόν να συνεχιστεί για κάποιο καιρό σε τρεις σχετικούς τομείς που αφορούν στην αξιοποίηση και ευελιξία του πρωτοκόλλου. Η ομάδα εργασίας PKI4IPsec εξελίσσει την υποδομή δημοσίου κλειδιού (Public Key Infrastructure) και τα πρότυπα των πιστοποιητικών για το IPsec, με στόχο τελικά να αυξήσει τη χρήση μεθόδων αυθεντικοποίησης δημοσίου κλειδιού. Η ομάδα εργασίας IKE Mobility επεξεργάζεται επεκτάσεις του πρωτοκόλλου, ώστε αυτό να επιτρέψει τη χρήση του σε εφαρμογές με εξυπηρετητές πολλαπλών διευθύνσεων (multihoming), καθώς και κινητών και υπηρεσιών περιαγωγής (roaming). Η ομάδα εργασίας

Better-Than-Nothing-Security (btns) έχει επιφορτιστεί με την εύρεση επεκτάσεων στην αρχιτεκτονική του IPsec, που θα το επιτρέψουν να χρησιμοποιεί μη αυθεντικοποιημένους συσχετισμούς ασφαλείας (SA), με στόχο την αξιοποίησή του σε εφαρμογές όπου επιτρέπεται αυτό, με απλούστερο και γρηγορότερο τρόπο.

Η πολυπλοκότητα του IKE v2, έχει οδηγήσει μέχρι σήμερα σε ανάλυση ασφαλείας μόνο απλοποιημένων εκδόσεών του. Έτσι, ολόκληρο το πρωτόκολλο θα πρέπει να εκτεθεί σε μια συστηματική ανάλυση. Τα θεωρητικά πρωτόκολλα τείνουν να περιγράφονται σε 6 γραμμές ή και λιγότερο, ενώ για το IKE-v2 απαιτήθηκαν σχεδόν 100 σελίδες στο RFC4306. Είναι επιθυμητή η ανάπτυξη αλγορίθμων AEAD (Authenticated Encryption with Associated Data) κατάλληλων για το ESP, κατά προτίμηση όχι προστατευμένων από πατέντες. Οι αλγόριθμοι AEAD είναι κρυπταλγόριθμοι τμήματος, οι οποίοι κρυπτογραφούν ένα τμήμα του μηνύματος και ταυτόχρονα αυθεντικοποιούν το συνολικό μήνυμα, συμπεριλαμβανομένου και των εξαρτώμενων από αυτό δεδομένων, όπως είναι η επικεφαλίδα του [42]. Επίσης, η μελέτη της επίδρασης της πρόσφατης κρυπτανάλυσης των συναρτήσεων hash στην ασφάλεια του HMAC είναι ένας ενδιαφέρον τομέας. Η μελέτη δηλαδή του κατά πόσο αποτελεσματικές μπορεί να είναι οι επιθέσεις στις συναρτήσεις hash του HMAC, σε ένα λογικό επίπεδο ασφαλείας του [03].

Από την ενασχόληση με τα πρότυπα ασφαλείας γίνεται φανερό ότι η πολυπλοκότητα του ορισμού τους και της σωστής εφαρμογής τους σε διαφορετικά συστήματα και περιβάλλοντα είναι ενδογενής. Το γεγονός αυτό αφήνει ανοιχτά παράθυρα για διαφορετικές ερμηνείες ως προς την υλοποίησή τους και ανοίγει τον δρόμο για τον σχεδιασμό επιθέσεων, όπως στην περίπτωση του ελέγχου της διαδικασίας γεμίσματος (padding) που εφαρμόζεται από το IPsec.

Η κρυπτογραφία είναι ένας μόνο παράγοντας του συνολικού επιπέδου ασφαλείας του πρωτοκόλλου. Ωστόσο, το να γίνεται η κρυπτογραφία σωστά είναι ένα κρίσιμο σημείο που βεβαιώνει ότι το πρωτόκολλο ανταποκρίνεται στο εκτιμώμενο επίπεδο ασφαλείας.

3. Το πρωτόκολλο SSL / TLS

Το TLS (Transport Layer Security) και ο προκάτοχός του το SSL (Secure Socket), είναι κρυπτογραφικά πρωτόκολλα που παρέχουν ασφάλεια στις επικοινωνίες στο διαδίκτυο. Το TLS και το SSL κρυπτογραφούν τα πλαίσια των συνδέσεων δικτύου πάνω από το επίπεδο μεταφοράς (transport layer), χρησιμοποιώντας συμμετρική κρυπτογραφία για την εμπιστευτικότητα και κώδικα αυθεντικοποίησης μηνύματος (MAC) με κλειδί για την αξιοπιστία του μηνύματος.

Διάφορες εκδόσεις των πρωτοκόλλων χρησιμοποιούνται ευρέως σε εφαρμογές, όπως σε φυλλομετρητές, ηλεκτρονικά μηνύματα, fax μέσω διαδικτύου, ανταλλαγή μηνυμάτων και φωνής μέσω IP. Το TLS είναι ένα πρότυπο της IETF, με τελευταία ενημέρωσή του το κείμενο RFC 5246 και βασίζεται στο προϋπάρχον πρωτόκολλο SSL που αναπτύχθηκε από την εταιρεία Netscape.

Το TLS επιτρέπει στις εφαρμογές πελάτη και εξυπηρετητή να επικοινωνούν σε ένα δίκτυο με τρόπο που αποτρέπει την αλλοίωση των δεδομένων ή την υποκλοπή τους. Στη συνήθη χρήση τελικού χρήστη, η αυθεντικοποίηση TLS είναι μονόπλευρη. Μόνον ο εξυπηρετητής αυθεντικοποιείται (ο πελάτης ξέρει την ταυτότητα του εξυπηρετητή), αλλά όχι αντίστροφα (ο πελάτης δεν αυθεντικοποιείται και παραμένει ανώνυμος). Το TLS υποστηρίζει επίσης, και την πιο ασφαλή λειτουργία αμφίδρομης σύνδεσης που συνήθως χρησιμοποιείται σε επιχειρησιακές εφαρμογές, στις οποίες και τα δύο μέρη της επικοινωνίας μπορούν να είναι σίγουρα με ποιόν επικοινωνούν (με την προϋπόθεση ότι επιμελώς εξετάζουν την πληροφορία της ταυτότητας του πιστοποιητικού του άλλου μέρους), γνωστή ως αμοιβαία αυθεντικοποίηση. Η αμοιβαία αυθεντικοποίηση απαιτεί από την πλευρά του πελάτη TLS να κατέχει ένα πιστοποιητικό (το οποίο δεν συμβαίνει στην περίπτωση του τελικού χρήστη φυλλομετρητή).

Συνήθως η πληροφορία του κλειδιού και τα απαραίτητα πιστοποιητικά για το TLS διατηρούνται στη μορφή πιστοποιητικών X.509, τα οποία ορίζουν τα απαραίτητα πεδία και τη μορφή των δεδομένων. Το SSL λειτουργεί με

αρθρωτό τρόπο. Από τον σχεδιασμό του είναι επεκτάσιμο, με υποστήριξη συμβατότητας με προγενέστερα και μεταγενέστερα συστήματα και συμφωνίας μεταξύ των μερών της επικοινωνίας.

Προγενέστερες ερευνητικές προσπάθειες για την ανάπτυξη του TLS, περιλαμβάνανε το SNP (Secure Network Programming) API (Application programming interface), το οποίο το 1993 ακολούθησε την προσέγγιση της υιοθέτησης ενός ασφαλούς επιπέδου μεταφοράς σε επίπεδο εφαρμογής, που έμοιαζε στο Berkeley sockets και διευκόλυνε την προσαρμογή των προηγούμενων δικτυακών εφαρμογών στα μέτρα ασφαλείας.

Το αρχικό πρωτόκολλο SSL αναπτύχθηκε από την εταιρεία Netscape. Η έκδοση 1 δεν δημοσιεύτηκε ποτέ. Η δεύτερη έκδοσή του δημοσιεύτηκε το 1995, αλλά περιείχε ένα αριθμό από κενά ασφαλείας, γεγονός που οδήγησε τελικά στην ανάπτυξη της τρίτης έκδοσης που δημοσιεύτηκε το 1996.

Η πρώτη έκδοση του TLS (v1.0) περιγράφηκε αρχικά στο κείμενο RFC 2246 τον Ιανουάριο του 1999 ως αναβάθμιση του SSL version 3.0. Όπως αναφέρεται στο κείμενο αυτό, οι διαφορές ανάμεσα στο TLS και το SSL 3.0 δεν είναι δραματικές, αλλά είναι σημαντικές ώστε να μην επιτρέπουν στα δύο πρωτόκολλα να είναι συμβατά. Επίσης, το TLS 1.0 δεν παρέχει δυνατότητα σε μια υλοποίησή του να υποβαθμίσει τη σύνδεση με χρήση του SSL 3.0. Το TLS 1.1 περιγράφηκε στο κείμενο RFC 4346 τον Απρίλιο του 2006. Ήταν μια αναβάθμιση από την προηγούμενη έκδοσή του την 1.0. Σημαντικές προσθήκες σε αυτή την έκδοση περιλαμβάνουν τα παρακάτω:

α) την προσθήκη προστασίας κατά επιθέσεων λειτουργίας μεταφοράς CBC (Cipher Block Chaining)
β) το προκαθορισμένο διάνυσμα αρχικοποίησης IV (Initialization Vector) αντικαταστάθηκε με ένα οριζόμενο κάθε φορά διάνυσμα αρχικοποίησης.
γ) αλλαγή στον τρόπο χειρισμού των σφαλμάτων από τη διαδικασία γεμίσματος (padding)
δ) υποστήριξη δήλωσης παραμέτρων στον φορέα IANA (Internet Assigned Numbers Authority)

Η έκδοση 1.2 του TLS περιγράφηκε στο κείμενο RFC 5246 τον Αύγουστο του 2008. Βασίζεται στην προηγούμενη έκδοση του πρωτοκόλλου. Οι μεγαλύτερες διαφορές περιλαμβάνουν:

α) Ο συνδυασμός MD5-SHA-1 στην ψευδοτυχαία συνάρτηση αντικαταστάθηκε με τον SHA-256, με δυνατότητα επιλογής ψευδοτυχαίων συναρτήσεων από μια σουίτα κρυπταλγορίθμων.

β) Ο συνδυασμός MD5-SHA-1 στο ολοκληρωμένο μήνυμα αντικαταστάθηκε με τον SHA-256, με δυνατότητα επιλογής συναρτήσεων κατακερματισμού από μια σουίτα κρυπταλγορίθμων.

γ) Ο συνδυασμός MD5-SHA-1 στα ψηφιακά υπογεγραμμένα στοιχεία αντικαταστάθηκε με μία συνάρτηση κατακερματισμού που συμφωνείται κατά τη διαδικασία χειραψίας, με προεπιλεγμένη την SHA-1.

δ) Βελτίωση της ικανότητας του εξυπηρετητή και του πελάτη να προσδιορίσουν ποιούς αλγορίθμους κατακερματισμού και υπογραφής θα κάνουν αποδεκτούς.

ε) Επέκταση της υποστήριξης κρυπταλγορίθμων αυθεντικοποίησης, που χρησιμοποιούνται κυρίως για λειτουργία GCM (Galois/Counter Mode) και CCM της κρυπτογράφησης AES (Advanced Encryption Standard).

στ) Προστέθηκαν επεκτάσεις του TLS και σουίτες κρυπταλγορίθμων του AES.

Στον σχεδιασμό εφαρμογών το TLS συνήθως υλοποιείται στην κορυφή όσων πρωτοκόλλων μεταφοράς υπάρχουν, ενθυλακώνοντας πρωτόκολλα των εφαρμογών όπως το HTTP, FTP, SMTP, NNTP και XMPP. Ιστορικά χρησιμοποιήθηκε κυρίως με αξιόπιστα πρωτόκολλα, όπως το TCP. Ωστόσο, έχει επίσης υλοποιηθεί με πρωτόκολλα μεταφοράς πλαισίων δεδομένων, όπως το UDP (User Datagram Protocol) και το DCCP (Datagram Congestion Control Protocol), η χρήση του οποίου έχει προτυποποιηθεί ανεξάρτητα με τον όρο DTLS (Datagram Transport Layer Security).

Μια κύρια χρήση του TLS ήταν για να ασφαλίσει την κυκλοφορία δεδομένων στον παγκόσμιο ιστό που μεταφέρονταν με το HTTP, στη μορφή του HTTPS. Αξιοσημείωτες εφαρμογές της παραπάνω χρήσης του πρωτοκόλλου περιλαμβάνουν το ηλεκτρονικό εμπόριο και τη διαχείριση κεφαλαίων. Επίσης, το SMTP (Simple Mail Transfer Protocol) προστατεύεται με το πρωτόκολλο TLS, όπως περιγράφεται στο σχετικό κείμενο RFC 3207. Αυτές οι εφαρμογές

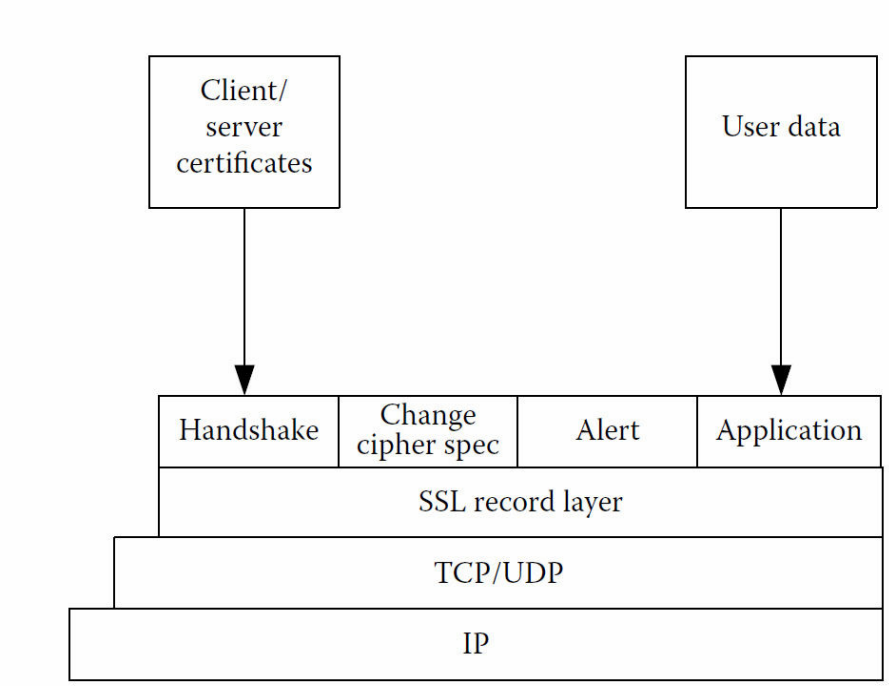
χρησιμοποιούν πιστοποιητικά δημοσίου κλειδιού για να επιβεβαιώσουν την ταυτότητα των τελικών χρηστών.

Το TLS μπορεί επίσης, να χρησιμοποιηθεί για να δρομολογήσει μέσα από ασφαλές κανάλι την κίνηση ενός δικτύου προκειμένου να γίνει ιδιωτικό δίκτυο (Virtual Private Network), όπως στην περίπτωση του OpenVPN. Πολλές εταιρείες σε εφαρμογές τους συνδυάζουν τις ικανότητες κρυπτογράφησης και αυθεντικοποίησης του TLS, με την εξουσιοδότηση (authorization). Υπάρχει επίσης, ουσιαστική ανάπτυξη από το τέλος του 1990, της δημιουργίας εφαρμογών πελάτη ανεξάρτητων του φυλλομετρητή, που να υποστηρίζουν το μοντέλο πελάτη / εξυπηρετητή. Συγκρινόμενο με την τεχνολογία εικονικών ιδιωτικών δικτύων (VPN) με IPsec, το TLS έχει κάποια ενδογενή πλεονεκτήματα όσον αφορά στη διαπέραση προγραμμάτων προστασίας (firewalls) και δρομολογητών NAT, που το καθιστούν πιο εύκολο στη διαχείριση πολλών και απομακρυσμένων κόμβων. Το TLS είναι επίσης, μια συνήθης μέθοδος προστασίας των σημάτων ελέγχου του πρωτοκόλλου SIP (Session Initiation Protocol). Το TLS μπορεί να χρησιμοποιηθεί για να παρέχει αυθεντικοποίηση και κρυπτογραφία στο SIP που σχετίζεται με εφαρμογές VoIP (Voice over IP) και άλλων εφαρμογών που βασίζονται σε αυτό.

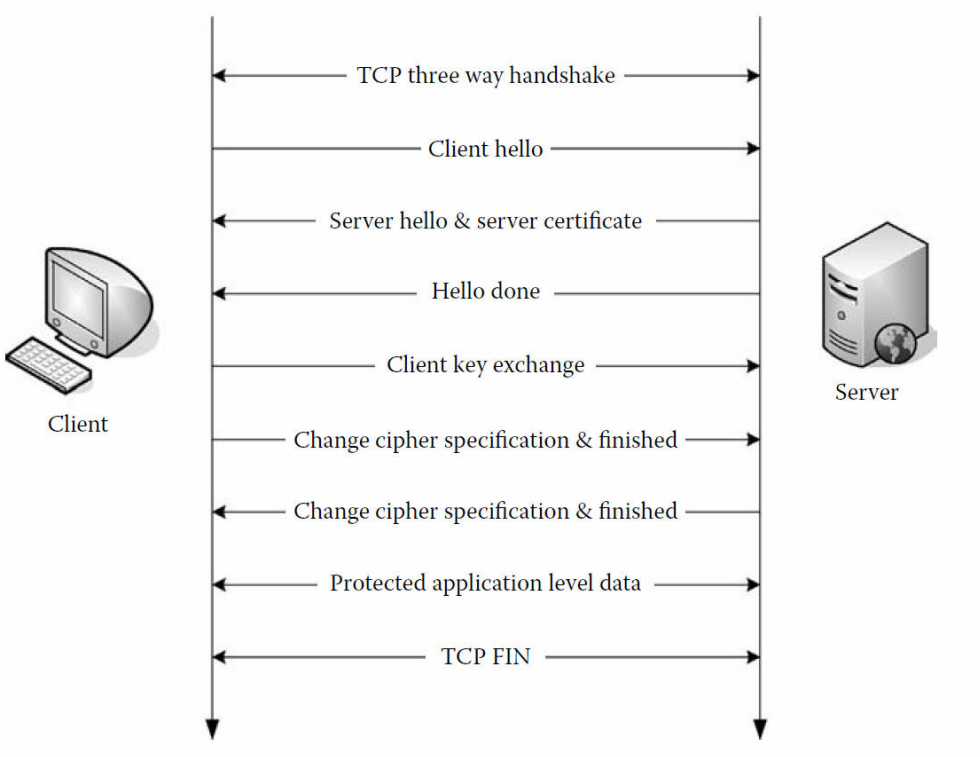
Το TLS αποτελείται από δύο επίπεδα: το επίπεδο εγγραφής (record layer) και το επίπεδο χειραψίας (handshake layer). Στο σχήμα 18 φαίνεται η θέση των δύο επιπέδων του πρωτοκόλλου στη συνολική στοίβα των πρωτοκόλλων που εμπλέκονται στην επικοινωνία δύο μερών. Το επίπεδο χειραψίας χρησιμοποιείται για να συλλέξει πληροφορίες για τον server και τον client και εν συνεχεία, να δημιουργήσει κλειδιά συνόδου για την προστασία των δεδομένων. Η πληροφορία αυτή μπορεί να χρησιμοποιηθεί αργότερα για την αναγνώριση και αυθεντικοποίηση του server και του χρήστη. Στην πραγματικότητα το επίπεδο χειραψίας αποτελείται από τρία υποεπίπεδα (sublayers): (1) το πραγματικό πρωτόκολλο χειραψίας (handshake protocol), (2) το πρωτόκολλο ειδοποίησης (alert protocol), το οποίο μεταφέρει πληροφορίες κατάστασης από τον server στον client και (3) το πρωτόκολλο Change Cipher Spec που χρησιμοποιείται για να επιβεβαιώσει τις παραμέτρους προστασίας. Το επίπεδο εγγραφής είναι υπεύθυνο για την προστασία του μηνύματος χρησιμοποιώντας τις παραμέτρους που

συμφωνήθηκαν στο επίπεδο χειραψίας και χρησιμοποιείται από τα ανώτερα επίπεδα για να στείλουν και να λάβουν πληροφορία.

Σύμφωνα με το κείμενο RFC 2246, ο server και ο client μπορούν να εδραιώσουν την ασφάλεια μέσα από τέσσερις φάσεις χειραψίας, οι οποίες φαίνονται στο σχήμα 19. Οι φάσεις αυτές είναι η φάση χαιρετισμού (Hello phase), η φάση αυθεντικοποίησης του εξυπηρετητή (server authentication phase), η φάση αυθεντικοποίησης του πελάτη (client authentication phase) και η φάση συμφωνίας έναρξης προστασίας (Negotiate Start of Protection phase). Οι γενικές αυτές φάσεις χειραψίας εξειδικεύονται και επεξηγούνται στις ενότητες 3.1.1 έως 3.1.3.



Σχήμα 18. Τα επίπεδα του πρωτοκόλλου SSL



Σχήμα 19. Βήματα εδραίωσης ασφαλούς σύνδεσης με SSL

3.1. Πρωτόκολλο χειραψίας (handshaking protocol) σε TLS

Ο πελάτης και εξυπηρετητής του TLS μπαίνουν σε κατάσταση σύνδεσης χρησιμοποιώντας μια διαδικασία χειραψίας (handshaking). Κατά τη διάρκεια αυτής της διαδικασίας, ο πελάτης και ο server συμφωνούν σε διάφορες παραμέτρους που αφορούν στην ασφάλεια της σύνδεσης. Η χειραψία ξεκινάει όταν ένας πελάτης συνδέεται σε έναν server TLS και αιτείται μιας ασφαλούς σύνδεσης παρουσιάζοντας μια λίστα υποστηριζόμενων κρυπταλγορίθμων και συναρτήσεων κατακερματισμού (hash functions). Από τη λίστα αυτή, ο server επιλέγει τις ισχυρότερες διαθέσιμες συναρτήσεις hash και κρυπταλγόριθμους, που υποστηρίζει και αυτός, ενημερώνοντας το πρόγραμμα πελάτη για την επιλογή του. Ο server στέλνει πίσω την ταυτότητά του σε μορφή ψηφιακού πιστοποιητικού. Το πιστοποιητικό συνήθως περιέχει το όνομα του εξυπηρετητή, την αρχή έκδοσης έμπιστων πιστοποιητικών (CA) και το δημόσιο κλειδί κρυπτογράφησής του. Το πρόγραμμα του πελάτη μπορεί να επικοινωνήσει με την αρχή έκδοσης του πιστοποιητικού για να διασφαλίσει τη γνησιότητά του πριν προχωρήσει στα επόμενα στάδια. Προκειμένου να δημιουργήσει τα κλειδιά συνόδου (session keys) που θα χρησιμοποιηθούν για

την ασφαλή σύνδεση, ο πελάτης κρυπτογραφεί ένα τυχαίο αριθμό RN με το δημόσιο κλειδί του server (PbK) και στέλνει το αποτέλεσμα στον server. Μόνο ο server θα πρέπει να είναι σε θέση να το αποκρυπτογραφήσει με το ιδιωτικό του κλειδί PvK. Αυτός είναι ο ένας παράγοντας που καθιστά τα κλειδιά κρυφά από τρίτα μέρη, καθόσον μόνο ο server και ο πελάτης έχουν πρόσβαση σε αυτά τα δεδομένα. Ο πελάτης ξέρει το PbK και το RN, ενώ ο server γνωρίζει το PvK και μετά την αποκρυπτογράφηση του μηνύματος του πελάτη, το RN. Ένα τρίτο μέρος μπορεί να μάθει το RN μόνο αν έχει υποκλέψει ή ανακτήσει το PvK. Από τον τυχαίο αριθμό και τα δύο μέρη δημιουργούν τα κλειδιά για την κρυπτογράφηση και αποκρυπτογράφηση.

Στο σημείο αυτό τελειώνει η διαδικασία της χειραψίας και ξεκινάει η ασφαλής σύνδεση, η οποία κρυπτογραφείται με τα κλειδιά μέχρι τη λήξη της σύνδεσης.

Το TLS ανταλλάσσει πεδία, τα οποία ενθυλακώνουν τα δεδομένα που πρέπει να μεταφερθούν. Κάθε πεδίο μπορεί να συμπιεστεί, να προστεθούν σε αυτό bit γεμίσματος, καθώς και μήνυμα κώδικα αυθεντικοποίησης (MAC) ή να κρυπτογραφηθεί, αναλόγως της κατάστασης της σύνδεσης. Κάθε πεδίο έχει ένα τμήμα τύπου περιεχομένου (content type) που προσδιορίζει το πεδίο, ένα τμήμα μήκους πεδίου και ένα τμήμα που ορίζει την έκδοση του TLS. Όταν ξεκινάει η σύνδεση, το πεδίο ενθυλακώνει ένα άλλο πρωτόκολλο μηνυμάτων χειραψίας (handshaking messaging protocol) το οποίο έχει τιμή 22 στο τμήμα τύπου περιεχομένου.

3.1.1 Διαδικασία χειραψίας TLS με αυθεντικοποίηση μόνο του server

Ακολουθεί ένα παράδειγμα απλής χειραψίας TLS, στην οποία ο server μόνο και όχι ο πελάτης, αυθεντικοποιείται με το πιστοποιητικό του.

α) Φάση διαπραγμάτευσης (negotiation phase)

- Ο πελάτης στέλνει ένα μήνυμα ClientHello προσδιορίζοντας τη μεγαλύτερη έκδοση του πρωτοκόλλου TLS που υποστηρίζει, έναν τυχαίο αριθμό, μια λίστα διαθέσιμων κρυπταλγορίθμων και προτεινόμενων μεθόδων συμπίεσης. Εάν ο πελάτης προσπαθεί να επανεκκινήσει μια

προηγούμενη μη ολοκληρωμένη διαδικασία χειραψίας, μπορεί να στείλει και το αναγνωριστικό της συνεδρίας (session ID).

- Ο εξυπηρετητής απαντάει με ένα μήνυμα ServerHello, που περιέχει την επιλεγμένη έκδοση του πρωτοκόλλου, έναν τυχαίο αριθμό, τη μέθοδο κρυπτογράφησης και συμπίεσης, από τις διαθέσιμες επιλογές που παρείχε ο πελάτης. Για να επιβεβαιώσει ή να επιτρέψει μια διαδικασία χειραψίας που είχε διακοπεί μπορεί να στείλει το αναγνωριστικό της συνεδρίας. Η έκδοση του πρωτοκόλλου που θα επιλεγεί θα πρέπει να είναι η μέγιστη δυνατή από αυτές που υποστηρίζουν ο πελάτης και ο server. Για παράδειγμα, εάν το πρόγραμμα του πελάτη υποστηρίζει το TLS1.1 και ο server το TLS1.2, θα πρέπει να επιλεγεί το TLS1.1 και όχι το SSL-v3 που είναι παλαιότερη έκδοση.
- Ο εξυπηρετητής στέλνει μήνυμα με το πιστοποιητικό του. Ανάλογα με την επιλεγμένη μέθοδο κρυπτογράφησης, αυτό το βήμα μπορεί να παραληφθεί.
- Ο εξυπηρετητής στέλνει ένα μήνυμα ServerHelloDone, που δείχνει ότι έχει τελειώσει με τη διαδικασία χειραψίας.
- Ο πελάτης απαντάει με ένα μήνυμα ClientKeyExchange, το οποίο μπορεί να περιέχει μια τιμή PreMasterSecret, ένα δημόσιο κλειδί ή τίποτα (εξαρτάται από την επιλεγμένη μέθοδο κρυπτογράφησης).
- Ο πελάτης και ο server χρησιμοποιούν εν συνεχεία τους τυχαίους αριθμούς και την τιμή PreMasterSecret για να υπολογίσουν ένα κοινό μυστικό που λέγεται master secret. Όλα τα άλλα δεδομένα για το κλειδί της σύνδεσης εξάγεται από το master secret και τις τυχαίες τιμές που παράγουν ο server και ο client, οι οποίες παίρνουν τιμή από προσεκτικά σχεδιασμένες ψευδοτυχαίες συναρτήσεις.

β) Το πρόγραμμα του πελάτη στέλνει τώρα ένα πεδίο ChangeCipherSpec, που ενημερώνει τον server ότι από εδώ και πέρα ό,τι του ‘λέει’ θα είναι αυθεντικοποιημένο και κρυπτογραφημένο, εάν υπήρχαν παράμετροι κρυπτογράφησης στο πιστοποιητικό του server. Το ChangeCipherSpec είναι από μόνο του ένα πρωτόκολλο επιπέδου πεδίου (record level protocol) με τιμή του τύπου περιεχομένου ίση με 20.

- Τέλος, ο πελάτης στέλνει ένα αυθεντικοποιημένο και κρυπτογραφημένο μήνυμα Finished, το οποίο περιέχει την τιμή μια συνάρτησης

κατακερματισμού (hash) και ένα κώδικα αυθεντικοποίησης μηνύματος (MAC) από τα προηγούμενα μηνύματα χειραψίας.

- Ο server θα προσπαθήσει να αποκρυπτογραφήσει το μήνυμα Finished του πελάτη και να επαληθεύσει τις τιμές hash και MAC. Εάν η αποκρυπτογράφηση ή η επαλήθευση αποτύχει, η χειραψία θεωρείται ότι έχει αποτύχει και η σύνδεση θα πρέπει να διακοπεί.

γ) Τέλος ο εξυπηρετητής στέλνει ένα μήνυμα ChangeCipherSpec, λέγοντας στο πρόγραμμα του πελάτη ότι από εδώ και πέρα ό,τι του στέλνει θα είναι αυθεντικοποιημένο (και κρυπτογραφημένο με το ιδιωτικό κλειδί του server που σχετίζεται με το δημόσιο κλειδί στο πιστοποιητικό του server, εάν είχε συμφωνηθεί κρυπτογραφία)

- Ο server αποστέλλει το αυθεντικοποιημένο και κρυπτογραφημένο μήνυμα Finished.
- Ο πελάτης εκτελεί την ίδια αποκρυπτογράφηση και επαλήθευση.

δ) Διαδικασία εφαρμογής: σε αυτό το σημείο η χειραψία ολοκληρώνεται και το πρωτόκολλο της εφαρμογής ενεργοποιείται, με τιμή τύπου περιεχομένου ίση με 23. Τα μηνύματα της εφαρμογής που ανταλλάσσονται μεταξύ πελάτη και εξυπηρετητή θα αυθεντικοποιηθούν προαιρετικά, ακριβώς όπως στα μηνύματα Finished. Αλλιώς ο τύπος περιεχομένου θα αλλάξει σε 25 και ο πελάτης δεν θα αυθεντικοποιηθεί.

3.1.2 Διαδικασία χειραψίας TLS με αυθεντικοποίηση client και server

Το παρακάτω παράδειγμα δείχνει την αυθεντικοποίηση ενός πελάτη (επιπροσθέτως του server) με χρήση TLS και ανταλλαγή πιστοποιητικών μεταξύ και των δύο μερών. Τα στάδια αυτής της διαδικασίας είναι:

α) Φάση διαπραγμάτευσης (negotiation phase)

- Ο πελάτης στέλνει ένα μήνυμα ClientHello, προσδιορίζοντας την πιο πρόσφατη έκδοση του πρωτοκόλλου TLS που υποστηρίζει, έναν τυχαίο αριθμό και μια λίστα από προτεινόμενους κρυπταλγόριθμους και τρόπους συμπίεσης.

- Ο εξυπηρετητής απαντάει με μήνυμα ServerHello, που περιέχει την επιλεγμένη έκδοση του πρωτοκόλλου, έναν τυχαίο αριθμό, το σύνολο κρυπταλγορίθμων και τη μέθοδο συμπίεσης από αυτές που πρότεινε ο πελάτης. Ο server μπορεί επίσης, να στείλει και ένα αναγνωριστικό της συνεδρίας (session ID) ως τμήμα του μηνύματος για να συνεχίσει μια διαδικασία χειραψίας που είχε διακοπεί.
- • Ο server στέλνει το μήνυμα Certificate. Ανάλογα με το επιλεγμένο σύνολο κρυπταλγορίθμων το στάδιο αυτό μπορεί να παραληφθεί.
- Ο server ζητάει την αποστολή πιστοποιητικού από τον πελάτη, έτσι ώστε η σύνδεση να μπορεί να αυθεντικοποιηθεί αμοιβαία, με ένα μήνυμα CertificateRequest.
- Ο server στέλνει ένα μήνυμα ServerHelloDone, που δείχνει ότι τελείωσε με τη διαδικασία χειραψίας.
- Ο πελάτης απαντάει με ένα μήνυμα Certificate, που περιέχει το πιστοποιητικό του.
- Ο πελάτης στέλνει το μήνυμα ClientKeyExchange, το οποίο μπορεί να περιέχει ένα δημόσιο κλειδί PreMasterSecret ή τίποτα, ανάλογα με το επιλεγμένο σετ κρυπταλγορίθμων. Το PreMasterSecret κρυπτογραφείται με το δημόσιο κλειδί του πιστοποιητικού του server.
- Ο πελάτης στέλνει το μήνυμα CertificateVerify, το οποίο είναι μια ψηφιακή υπογραφή των προηγούμενων μηνυμάτων χειραψίας, χρησιμοποιώντας το ιδιωτικό κλειδί του πιστοποιητικού του. Αυτή η υπογραφή μπορεί να επιβεβαιωθεί χρησιμοποιώντας το δημόσιο κλειδί του πιστοποιητικού του πελάτη. Αυτό επιτρέπει στον server να γνωρίζει ότι ο πελάτης έχει πρόσβαση στο ιδιωτικό κλειδί του πιστοποιητικού, άρα το πιστοποιητικό του ανήκει.
- Ο πελάτης και ο εξυπηρετητής χρησιμοποιούν τους τυχαίους αριθμούς και το PreMasterKey για να υπολογίσουν ένα κοινό μυστικό, που λέγεται master secret. Όλα τα άλλα κλειδιά για αυτή τη σύνδεση εξάγονται από αυτό το master secret και τις τυχαίες τιμές που δημιούργησαν ο πελάτης και ο εξυπηρετητής, μέσα από μια προσεκτικά σχεδιασμένη ψευδοτυχαία συνάρτηση.

β) Ο πελάτης τώρα στέλνει το πεδίο ChangeCipherSpec, ενημερώνοντας τον server ότι από εδώ και πέρα ό,τι στέλνει θα είναι αυθεντικοποιημένο και κρυπτογραφημένο, εάν αυτό είχε συμφωνηθεί. Το ChangeCipherSpec είναι

από μόνο του ένα πρωτόκολλο επιπέδου πεδίου με τιμή τύπου περιεχομένου 20 και όχι 22.

- Ο πελάτης στέλνει ένα κρυπτογραφημένο μήνυμα Finished, που περιέχει μια τιμή hash και MAC από τα προηγούμενα μηνύματα χειραψίας.
- Ο server θα προσπαθήσει να αποκρυπτογραφήσει το μήνυμα Finished του πελάτη και να επαληθεύσει τις τιμές hash και MAC. Εάν η αποκρυπτογράφηση και η επαλήθευση αποτύχει, η χειραψία θεωρείται ότι έχει αποτύχει και θα πρέπει να διακοπεί η σύνδεση.

γ) Ο server στέλνει το μήνυμα ChangeCipherSpec, λέγοντας στον πελάτη ότι από εδώ και πέρα ό,τι στέλνει θα είναι αυθεντικοποιημένο και κρυπτογραφημένο, αν αυτό είχε συμφωνηθεί.

- Ο server στέλνει το δικό του κρυπτογραφημένο μήνυμα Finished.
- Ο πελάτης κάνει αποκρυπτογράφηση και επαλήθευση.

δ) Φάση εφαρμογής: στο σημείο αυτό η χειραψία έχει ολοκληρωθεί και ενεργοποιείται το πρωτόκολλο εφαρμογής με τιμή τύπου περιεχομένου ίση με 23. Τα μηνύματα της εφαρμογής που ανταλλάσσονται μεταξύ client και server θα είναι επίσης κρυπτογραφημένα, όπως και τα μηνύματα Finished. Η εφαρμογή δεν θα επιστρέψει ποτέ ξανά πληροφορία κρυπτογράφησης TLS, χωρίς τον τύπο περιεχομένου 32 της απολογίας.

Οι λειτουργίες δημοσίου κλειδιού, όπως ο RSA, είναι σχετικά δαπανηρές όσον αφορά στην υπολογιστική δύναμη που απαιτούν. Το TLS παρέχει τη δυνατότητα ασφαλούς παράκαμψης στον μηχανισμό χειραψίας για τις λειτουργίες αυτές. Στη συνηθισμένη πλήρη διαδικασία χειραψίας, ο server στέλνει ένα αναγνωριστικό συνεδρίας (session id) ως τμήμα του μηνύματος ServerHello. Ο πελάτης συσχετίζει το session id με τη διεύθυνση IP και τη θύρα TCP του server, έτσι ώστε όταν ο πελάτης συνδεθεί ξανά στον server, μπορεί να χρησιμοποιήσει το session id για να παρακάμψει τη χειραψία. Στον εξυπηρετητή το session id αντιστοιχεί στις κρυπτογραφικές παραμέτρους που είχαν προηγουμένως συμφωνηθεί, ιδιαίτερα στο master secret. Και οι δύο πλευρές θα πρέπει να έχουν το ίδιο master secret, ειδάλλως η ανακτημένη χειραψία θα αποτύχει, αποτρέποντας έτσι τον κακόβουλο εισβολέα να χρησιμοποιήσει το session id για να εισβάλει στη σύνδεση. Τα τυχαία δεδομένα στα μηνύματα ClientHello και ServerHello εγγυώνται ότι τα

δημιουργούμενα κλειδιά της σύνδεσης θα είναι διαφορετικά από την προηγούμενη σύνδεση.

3.1.3 Διαδικασία συντομευμένης χειραψίας TLS (abbreviated handshake)

Στα κείμενα RFC και τη βιβλιογραφία, αυτός ο τύπος χειραψίας λέγεται συντομευμένη χειραψία (abbreviated handshake) ή επανεκκίνησης (reset).

Τα στάδια της συντομευμένης χειραψίας είναι:

α) Φάση διαπραγμάτευσης:
- Ο πελάτης στέλνει το μήνυμα ClientHello, προσδιορίζοντας τη νεότερη έκδοση του πρωτοκόλλου TLS που υποστηρίζει, έναν τυχαίο αριθμό, ένα σετ από προτεινόμενους κρυπταλγόριθμους και μεθόδους συμπίεσης. Σε αυτό το μήνυμα περιλαμβάνεται το session id από την προηγούμενη σύνδεση TLS.
- Ο server απαντάει με ένα μήνυμα ServerHello, που περιέχει την επιλεγμένη έκδοση του πρωτοκόλλου, έναν τυχαίο αριθμό, το σετ κρυπταλγορίθμων και τη μέθοδο συμπίεσης από αυτές που του προτάθηκαν από τον πελάτη. Εάν ο server αναγνωρίσει το session id που του στάλθηκε, απαντάει με το ίδιο session id. Ο πελάτης το χρησιμοποιεί αυτό για να αναγνωρίσει ότι πραγματοποιείται μια ανακτημένη χειραψία (resumed handshake). Εάν ο server δεν αναγνωρίσει το session id που του στάλθηκε από τον πελάτη, στέλνει διαφορετική τιμή του session id. Αυτό δείχνει στον πελάτη ότι δεν θα γίνει ανακτημένη χειραψία. Στο σημείο αυτό, και τα δύο μέρη έχουν το master key και τα τυχαία δεδομένα για να εξάγουν τα κλειδιά για αυτή τη σύνδεση.

β) Ο πελάτης στέλνει τώρα το πεδίο ChangeCipherSpec, δηλώνοντας στον server ότι από εδώ και πέρα ό,τι στέλνει θα είναι κρυπτογραφημένο. Το ChangeCipherSpec είναι από μόνο του ένα πρωτόκολλο επιπέδου πεδίου, με τύπο περιεχομένου 20 και όχι 22.
- Ο πελάτης στέλνει ένα κρυπτογραφημένο μήνυμα Finished, που περιέχει τιμές hash και MAC από τα προηγούμενα μηνύματα χειραψίας.

- Ο server θα προσπαθήσει να αποκρυπτογραφήσει το μήνυμα Finished του πελάτη και να επαληθεύσει τις τιμές hash και MAC. Εάν η αποκρυπτογράφηση ή η επαλήθευση αποτύχει, η χειραψία θεωρείται ότι απέτυχε και η σύνδεση θα πρέπει να διακοπεί.

γ) Ο server στέλνει το πεδίο ChangeCipherSpec, δηλώνοντας στον πελάτη ότι από εδώ και πέρα ό,τι στέλνει θα είναι κρυπτογραφημένο.
   - Ο server στέλνει το δικό του κρυπτογραφημένο μήνυμα Finished
   - Ο πελάτης πραγματοποιεί την ίδια αποκρυπτογράφηση και επαλήθευση.

δ) Φάση εφαρμογής: στο σημείο αυτό η χειραψία ολοκληρώθηκε και ενεργοποιείται το πρωτόκολλο εφαρμογής με τύπο περιεχομένου 23. Τα μηνύματα της εφαρμογής θα κρυπτογραφηθούν, όπως ακριβώς και στα μηνύματα Finished.

Εκτός από το πλεονέκτημα της απόδοσης, οι ανακτημένες συνεδρίες (resumed sessions) μπορούν επίσης, να χρησιμοποιηθούν για μεμονωμένες αιτήσεις, εφόσον είναι βέβαιο ότι η αρχική συνεδρία, όπως και κάθε άλλη ανακτημένη συνεδρία προήλθαν από τον ίδιο πελάτη. Το τελευταίο έχει ιδιαίτερη σημασία στο πρωτόκολλο FTP πάνω σε σύνδεση TLS/SSL, το οποίο σε άλλη περίπτωση θα ήταν ευάλωτο σε επιθέσεις τύπου άνθρωπος στη μέση (man in the middle), στις οποίες ο επιτιθέμενος θα μπορούσε να παρεμβληθεί στο περιεχόμενο δευτερευουσών συνδέσεων δεδομένων.

3.2. Το πρωτόκολλο εγγραφής (TLS record protocol)

Το πρωτόκολλο TLS record είναι ένα πρωτόκολλο με επίπεδα. Σε κάθε επίπεδο τα μηνύματα μπορεί να περιέχουν πεδία για το μήκος, την περιγραφή και το περιεχόμενο. Το πρωτόκολλο παίρνει τα προς μετάδοση μηνύματα, τοποθετεί τα δεδομένα σε διαχειρίσιμα τμήματα (blocks), συμπιέζει προαιρετικά τα δεδομένα, εφαρμόζει έναν κώδικα αυθεντικοποίησης μηνύματος (MAC), κρυπτογραφεί και μεταδίδει το τελικό αποτέλεσμα αυτής της διαδικασίας. Κατά ανάλογο τρόπο, τα δεδομένα που φθάνουν αποκρυπτογραφούνται, επαληθεύονται, αποσυμπιέζονται, αναδομούνται και παραδίδονται σε πελάτες (clients) υψηλότερου επιπέδου.

Υπάρχουν τέσσερα πρωτόκολλα που χρησιμοποιούν το TLS record: το πρωτόκολλο χειραψίας (handshake protocol), προειδοποίησης (alert protocol), αλλαγής ρυθμίσεων κρυπταλγορίθμου (change cipher spec protocol) και εφαρμογής δεδομένων (application data protocol). Προκειμένου να είναι δυνατή η επέκταση του πρωτοκόλλου, επιπρόσθετοι τύποι περιεχομένου πεδίου μπορεί να υποστηρίζονται από το πρωτόκολλο TLS record. Οι τύποι αυτοί καταχωρούνται από την IANA στο Μητρώο Τύπου Περιεχομένου του TLS. Οι υλοποιήσεις δεν θα πρέπει να στέλνουν τύπους πεδίων που δεν υπάρχουν στο σχετικό κείμενο RFC, εκτός και αν αυτό έχει συμφωνηθεί σε κάποια επέκταση του πρωτοκόλλου. Εάν μια υλοποίηση TLS λάβει έναν άγνωστο τύπο πεδίου, θα πρέπει να στείλει ένα μήνυμα unexpected_message.

Η κατάσταση σύνδεσης (TLS connection state) είναι το περιβάλλον λειτουργίας του πρωτοκόλλου TLS. Προσδιορίζει τον αλγόριθμο συμπίεσης, κρυπτογράφησης και έναν αλγόριθμο αυθεντικοποίησης μηνύματος (MAC). Επιπροσθέτως, οι παράμετροι αυτών των αλγορίθμων είναι γνωστοί: το κλειδί του κώδικα αυθεντικοποίησης μηνύματος MAC και τα κλειδιά για την κρυπτογράφηση της σύνδεσης και προς τις δύο κατευθύνσεις, ανάγνωσης και εγγραφής. Λογικά μπορεί να υπάρξουν τέσσερις καταστάσεις στο προσκήνιο: οι τρέχουσες καταστάσεις ανάγνωσης και εγγραφής και οι εκκρεμείς καταστάσεις ανάγνωσης και εγγραφής. Όλα τα πεδία επεξεργάζονται με βάση την τρέχουσα κατάσταση ανάγνωσης και εγγραφής.

Οι παράμετροι ασφαλείας για τις εκκρεμείς καταστάσεις μπορούν να οριστούν από το πρωτόκολλο χειραψίας (TLS Handshake protocol) και το ChangeCipherSpec μπορεί να κάνει τρέχουσες οποιεσδήποτε από τις εκκρεμείς καταστάσεις. Στην περίπτωση αυτή, η ανάλογη τρέχουσα κατάσταση εγκαταλείπεται και αντικαθίσταται από την εκκρεμή κατάσταση, ενώ η εκκρεμής κατάσταση αρχικοποιείται σε κενή κατάσταση. Δεν είναι επιτρεπτό να γίνει μια κατάσταση τρέχουσα, αν δεν έχει αρχικοποιηθεί με παραμέτρους ασφαλείας. Η αρχική κατάσταση πάντα προσδιορίζει ότι δεν θα χρησιμοποιηθεί κρυπτογράφηση, συμπίεση και αυθεντικοποίηση μηνύματος.

Οι παράμετροι ασφαλείας για την κατάσταση ανάγνωσης και εγγραφής της σύνδεσης TLS, ορίζονται παρέχοντας συγκεκριμένες τιμές, οι οποίες

αναφέρονται αναλυτικά στο σχετικό κείμενο RFC 5246. Οι τιμές αυτές αναφέρονται στο αν η οντότητα είναι server ή client, τον αλγόριθμο δημιουργίας κλειδιών από το master secret, στα μήκη κλειδιών που θα χρησιμοποιηθούν, στους τύπους λειτουργίας κρυπταλγορίθμων (π.χ. τμήματος, ροής), σε τυχαίες τιμές που απαιτούνται από τους αλγορίθμους (client random, server random), καθώς και σε άλλες παραμέτρους.

Το επίπεδο εγγραφής θα χρησιμοποιήσει αυτές τις παραμέτρους ασφαλείας για να υπολογίσει με έναν αλγόριθμο που περιγράφεται στο πρότυπο, έξι οντότητες, μερικές από τις οποίες δεν απαιτούνται από όλους τους αλγόριθμους κρυπτογράφησης και μένουν κενές. Οι οντότητες αυτές είναι client write MAC key, server write MAC key, client write encryption key, server write encryption key, client write IV και server write IV. Οι παράμετροι εγγραφής του πελάτη (client write) χρησιμοποιούνται από τον server όταν δέχεται και επεξεργάζεται εγγραφές και το αντίστροφο, για τις εγγραφές του server. Μόλις οριστούν οι παράμετροι ασφαλείας και δημιουργηθούν τα κλειδιά, οι καταστάσεις σύνδεσης μπορούν να αρχικοποιηθούν γινόμενες τρέχουσες. Αυτές οι τρέχουσες καταστάσεις πρέπει να ενημερώνονται για κάθε εγγραφή που επεξεργάζεται.

3.3. Υλοποίηση ασφαλούς καναλιού SSL για την επικοινωνία και τη μεταφορά δεδομένων από server σε client με OpenSSL

Όλα τα δεδομένα που διακινούνται μεταξύ συστημάτων server – client μπορούν εύκολα να γίνουν προσβάσιμα σε τρίτους μέσω του κατάλληλου λογισμικού που επεμβαίνει στα σημεία δρομολόγησης της κίνησης και σύνδεσης των χρηστών των δικτύων, ακόμα και σε LAN. Με τη σύνδεση σε ένα server telnet, FTP, POP3 ή IMAP, αποστέλλεται ο κωδικός του χρήστη στο δίκτυο και μπορεί να τον δει ο επιτιθέμενος που έχει συνδεθεί και παρακολουθεί την κίνηση στο δίκτυο (capturing). Το SSL μπορεί σε αυτή την περίπτωση, να προστατέψει τα διακινούμενα δεδομένα, αλλά θα πρέπει να το υποστηρίζουν και ο server και ο client.

Σε μια έτοιμη εφαρμογή η οποία δεν χρησιμοποιεί ασφαλή σύνδεση, θα πρέπει να ενσωματωθεί μια υλοποίηση του SSL/TLS και στα δύο προγράμματα (server και client). Μια εναλλακτική λύση η οποία δεν απαιτεί την υποστήριξη

του SSL/TLS από την εφαρμογή, είναι η χρήση του προγράμματος Stunnel, το οποίο βασίζεται στην υλοποίηση του OpenSSL. Το Stunnel είναι ένα απλό πρόγραμμα που μετατρέπει μια μη κρυπτογραφημένη σύνδεση σε κρυπτογραφημένη με SSL. Συνήθως ρυθμίζεται για να λειτουργεί σε ένα super server όπως ο inetd ή ο xinetd, και εν συνεχεία αξιοποιείται από ένα άλλο πρόγραμμα, όπως ο POP3 server που δεν υποστηρίζει SSL. Αυτός ο τρόπος λειτουργίας επιτρέπει την προστασία του server που συνήθως τρέχει από τον inetd, όπως είναι το telnet, NNTP και το IMAP. Ωστόσο, δεν μπορούν όλοι οι servers να προστατευθούν με κρυπτογράφηση, επειδή δεν υπάρχουν αντίστοιχα προγράμματα πελάτη (client) που να υποστηρίζουν SSL. Για παράδειγμα δεν υπάρχει πρόγραμμα πελάτη FTP που να χρησιμοποιεί SSL, επειδή υπάρχει εναλλακτικά το πακέτο SSH που επιτρέπει κρυπτογραφημένες συνεδρίες αλλά και τη μεταφορά αρχείων.

Σε περιβάλλον UNIX – Linux υπάρχουν ειδικά εργαλεία που επιτρέπουν την παραμετροποίηση του Stunnel, όπως το webmin module. Σε περιβάλλον win32, η ρύθμιση του Stunnel θα πρέπει να γίνει χειροκίνητα μέσα από το αρχείο ρυθμίσεων Stunnel.conf και για τον server και για τον client.

Στο παρακάτω σχήμα φαίνονται τα αρχεία ρυθμίσεων stunnel.conf του server και του client ενός καναλιού SSL. Μέσα από αυτό το κανάλι διακινούνται τα δεδομένα της εφαρμογής door_control σε περιβάλλον windows για τη μετάδοση σε απομακρυσμένα σημεία της κατάστασης (ανοικτή – κλειστή) θυρών, καθώς και για τον χειρισμό τους (άνοιγμα – κλείσιμο) μέσω δικτύου (LAN ή Internet). Στην εφαρμογή αυτή η χρήση κρυπτογραφημένου καναλιού είναι απαραίτητη, καθόσον τα δεδομένα της κατάστασης των θυρών, όπως και των εντολών ανοίγματος – κλεισίματος, διακινούνται μη κρυπτογραφημένα και ο επιτιθέμενος που έχει πρόσβαση στο δίκτυο θα μπορούσε να δώσει εντολές στην εφαρμογή ή να καταγράψει την τρέχουσα κατάσταση.

Ρυθμίσεις στον server (περιεχόμενο αρχείου stunnel.conf)

```
; SERVER side
; Sample stunnel configuration file by Manolis Bozis
; Some options used here may not be adequate for your particular configuration
; Certificate/key is needed in server mode and optional in client mode
; This certificate is provided only for testing and should not
; be used in a production environment
cert = mbozis.pem
key = privkey.pem
; Some performance tunings
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
;debug = 7
output = stunnel.log
; Service-level configuration
[door_control]
accept  = 1001
connect = 125
```

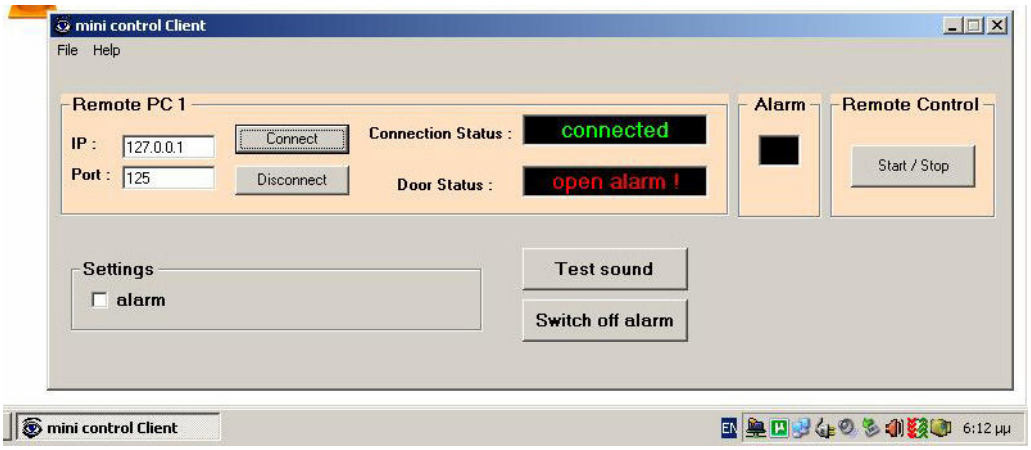
Ρυθμίσεις στον client (περιεχόμενο αρχείου stunnel.conf)

```
; CLIENT
client = yes
; Service-level configuration
output = stunnel.log
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1
 [door_control]
accept  = 125
connect = 192.168.0.3:1001
```

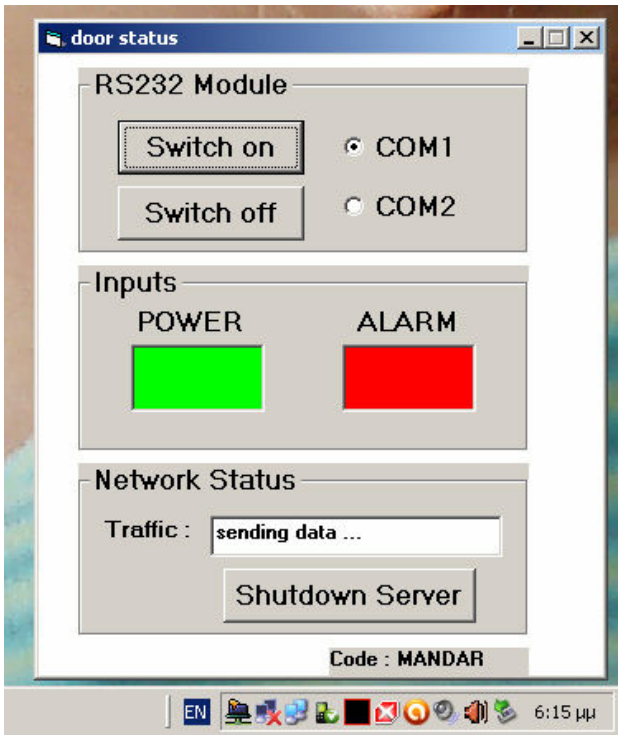
Σχήμα 20. Ρυθμίσεις του stunnel για την δημιουργία κρυπτογραφημένου καναλιού SSL

Στην περίπτωση αυτή ο server τρέχει στο pc διεύθυνση ip 192.168.0.3 και η εφαρμογή του πελάτη σε κάποια άλλη διεύθυνση ip εσωτερική του δικτύου. Η εφαρμογή χρησιμοποιεί το port 125 για τη διακίνηση των πακέτων, ενώ το port 1001 χρησιμοποιείται από το stunnel για τη δημιουργία ασφαλούς καναλιού μεταξύ των δύο μερών.

Στα παρακάτω σχήματα φαίνονται screenshots από την εφαρμογή του πελάτη (σχήμα 21) και του εξυπηρετητή (σχήμα 22).



Σχήμα 21. Screenshot της εφαρμογής door_control client και stunnel



Σχήμα 22. Screenshot της εφαρμογής door_control server και stunnel

Η διαδικασία ξεκινάει με την έναρξη της εφαρμογής stunnel (το σχετικό εικονίδιο  του προγράμματος φαίνεται στη γραμμή εργασιών των Windows στους υπολογιστές του εξυπηρετητή και του πελάτη). Το stunnel στον υπολογιστή του πελάτη δέχεται αιτήσεις για σύνδεση στο port 125 και συνδέεται με τον υπολογιστή του server με ασφαλή σύνδεση SSL στο port 1001. Ο server με τη σειρά του, παραλαμβάνει τα δεδομένα και, αφού τα αποκωδικοποιήσει, τα προωθεί στο port 125 όπου τα δέχεται η εφαρμογή

door_control server. Το port 1001 δεν έχει αποδοθεί σε κάποια γνωστή υπηρεσία σύμφωνα με το σχετικό κείμενο της IANA, άρα μπορεί να χρησιμοποιηθεί από τον χρήστη. Επίσης, ο χρήστης θα πρέπει να βεβαιωθεί ότι δεν τρέχει στον υπολογιστή του εφαρμογή που χρησιμοποιεί το port 125, το οποίο έχει αποδοθεί στην εφαρμογή Locus PC-Interface Net Map Ser σύμφωνα με την IANA. Εάν χρησιμοποιεί την παραπάνω εφαρμογή, θα πρέπει να αλλάξει την τιμή του port σε μια διαθέσιμη για το σύστημά του. Όπως φαίνεται στο σχετικό σχήμα 21, η εφαρμογή του πελάτη αντί να συνδεθεί απευθείας στη διεύθυνση του server (192.168.0.3), συνδέεται στη θύρα 125 του τοπικού υπολογιστή (127.0.0.1) όπου λαμβάνει τα δεδομένα το stunnel και τα προωθεί στον server μέσα από το ασφαλές κανάλι SSL.

Στη σύνδεση αυτή SSL αυθεντικοποιείται μόνο ο server με χρήση πιστοποιητικού. Για τη δημιουργία του πιστοποιητικού θα χρησιμοποιήσουμε την υλοποίηση OpenSSL. Αρχικά, δημιουργούμε το δημόσιο και το ιδιωτικό κλειδί που είναι απαραίτητα για το πιστοποιητικό. Αυτά συνήθως αποθηκεύονται σε ένα αρχείο ως ζευγάρι, με το ένα μισό να είναι το δημόσιο και το άλλο μισό το ιδιωτικό κλειδί. Στην OpenSSL το ιδιωτικό κλειδί περιέχει επίσης, και την πληροφορία του δημόσιου, έτσι δεν χρειάζεται να δημιουργηθεί χωριστά. Τα δημόσια κλειδιά υπάρχουν σε διάφορες παραλλαγές με χρήση διαφορετικών κρυπτογραφικών αλγορίθμων. Οι πιο δημοφιλείς που σχετίζονται με πιστοποιητικά είναι ο RSA και ο DSA. Στην εφαρμογή αυτή δημιουργούμε ένα κλειδί RSA, το οποίο μπορεί να χρησιμοποιηθεί και για κρυπτογράφηση και για υπογραφή. Η δημιουργία του στην OpenSSL, γίνεται με την εντολή:

openssl genrsa –out privkey.pem 2048

Ο αριθμός 2048 είναι το μέγεθος του κλειδιού σε bits. Σήμερα, 2048 ή και περισσότερο, συνιστάται για κλειδιά RSA, καθόσον μικρότερη τιμή θεωρείται ή σε μικρό χρονικό διάστημα θα θεωρηθεί ανασφαλής. Υπάρχει επίσης, η δυνατότητα στην παραπάνω εντολή να χρησιμοποιηθεί και ο διακόπτης *–des3*. Στην περίπτωση αυτή θα ζητηθεί από τον χρήστη να εισάγει έναν κωδικό προστασίας του κλειδιού. Όμως, αυτό καλό είναι να αποφεύγεται, εάν το κλειδί πρόκειται να χρησιμοποιηθεί για πιστοποιητικό server, όπως στην περίπτωσή

μας, καθόσον θα έπρεπε κάποιος να πληκτρολογεί τον κωδικό κάθε φορά που ο server αποκτούσε πρόσβαση στο κλειδί.

Για τη δημιουργία του πιστοποιητικού, θα πρέπει να κάνουμε αίτηση υπογραφής του σε μια πιστοποιημένη αρχή έκδοσης (Certified Authority). Η αρχή αυτή το υπογράφει και επιστρέφει το αποτέλεσμα πίσω στον χρήστη. Η δημιουργία της αίτησης στην OpenSSL μπορεί να γίνει με την εντολή:

openssl req –new –key privkey.pem –out cert.csr

Το αρχείο cert.scr που δημιουργείται μπορεί να σταλεί στην πιστοποιημένη αρχή, εάν αυτή μπορεί να επεξεργαστεί αρχεία της μορφής pem. Εάν όχι, υπάρχουν δυνατότητες μέσα από την OpenSSL να μετατραπεί το αρχείο σε άλλη μορφή. Όταν η αρχή ολοκληρώσει τους ελέγχους που πρέπει να κάνει και λάβει τη σχετική αμοιβή της, αποστέλλει υπογεγραμμένο το πιστοποιητικό. Βέβαια αυτό τις περισσότερες φορές δεν θα είναι σε μορφή pem, αλλά σε κάποια άλλη κωδικοποιημένη, όπως για παράδειγμα PKCS7 ή PKCS12. Ανάλογα με τις εφαρμογές, αυτές μπορεί να υποστηρίζονται ή όχι. Σε κάθε περίπτωση το OpenSSL περιέχει εργαλεία για τη μετατροπή σε κάποιο άλλο υποστηριζόμενο format.

Υπάρχει επίσης, η δυνατότητα δημιουργία πιστοποιητικού για ίδια χρήση. Αυτό είναι παρόμοιο με την προηγούμενη διαδικασία μόνο που, αντί για αίτηση χορήγησης πιστοποιητικού, παράγεται το πιστοποιητικό υπογεγραμμένο από τον ίδιο τον χρήστη. Αυτός βέβαια δεν είναι ο ενδεδειγμένος τρόπος, αλλά είναι επαρκής τρόπος για την ανάγκη της συγκεκριμένης εφαρμογής. Στην OpenSSL γίνεται, δίνοντας την εντολή:

openssl req –new –x509 –key privkey.pem –out mbozis.pem –days 1095

Η εκτέλεση της παραπάνω εντολής δημιουργεί το αρχείο πιστοποιητικού mbozis.pem, το οποίο μαζί με το αρχείο που περιέχει το ιδιωτικό κλειδί privkey.pem, θα πρέπει να υπάρχουν στη διαδρομή που βρίσκεται το αρχείο stunnel.exe. Στο αρχείο ρυθμίσεων του stunnel στη μεριά του server, βλέπουμε ότι γίνεται αναφορά στα αρχεία του πιστοποιητικού και του ιδιωτικού κλειδιού.

Κατά τη δημιουργία του κλειδιού ζητούνται κάποια προσωπικά στοιχεία του χρήστη, όπως είναι η διεύθυνσή του, χώρα, email, όνομα, οργανισμός κλπ, τα οποία εμφανίζονται κατά την προβολή του πιστοποιητικού. Τα στοιχεία αυτά δεν είναι άμεσα αναγνώσιμα από το σχετικό αρχείο με κατάληξη pem που τα περιέχει, καθόσον αυτό είναι σε κρυπτογραφημένη μορφή. Για να μπορέσει κάποιος να δει τα περιεχόμενα το πιστοποιητικού, θα πρέπει να δώσει την εντολή:

openssl x509 –in mbozis.pem –text –out mbozis.txt

Μετά την εκτέλεση της παραπάνω εντολής τα περιεχόμενο του πιστοποιητικού βρίσκεται σε μορφή απλού κειμένου στο σχετικό αρχείο mbozis.txt.

Τα περιεχόμενα αυτού του αρχείου που δημιουργήθηκε για την εφαρμογή, φαίνονται στο παρακάτω σχήμα :

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            e6:8e:a1:77:37:d0:29:00
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=GR, ST=Greece, L=Tripolis, O=UOP, OU=Computer
Faculty, CN=Manolis Bozis/emailAddress=pcst0918@uop.gr
        Validity
            Not Before: Feb 12 18:11:51 2011 GMT
            Not After : Feb 11 18:11:51 2014 GMT


        Subject: C=GR, ST=Greece, L=Tripolis, O=UOP, OU=Computer
Faculty, CN=Manolis Bozis/emailAddress=pcst0918@uop.gr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
```

Σχήμα 23. Περιεχόμενο του πιστοποιητικού του server

```
00:9b:d5:c4:88:58:a1:cb:80:aa:87:e4:86:f7:5f:

0a:68:8d:85:10:b8:eb:c5:f0:56:e3:c6:66:b4:ba:

15:52:bb:5e:2e:fb:32:85:3f:b6:65:b2:7c:8d:eb:

93:88:86:e5:f5:53:90:76:96:78:2e:da:43:c2:5d:

73:fc:58:b9:1f:0b:3a:43:4f:ac:63:19:dd:2c:bd:

4b:5f:c1:bf:79:cc:91:8d:d0:eb:df:7c:7f:41:4b:

4b:4a:3b:2d:fc:12:5c:97:ad:b1:8f:09:91:34:86:

19:b1:15:b2:69:1d:53:15:51:89:6a:c8:22:2f:21:

b6:7e:84:71:1b:59:91:e7:39:9e:6d:bc:ef:39:31:

53:ce:1f:5c:0b:c0:bc:6f:2f:06:45:2f:27:85:d2:

6d:32:00:d6:7a:e6:90:5e:ab:8d:49:59:5b:07:49:

a7:a7:78:80:77:b7:b8:64:90:ff:18:fd:ab:10:84:

9c:3c:b5:ea:2e:80:5a:41:7f:51:da:18:5d:f7:6d:

dd:78:51:ce:7f:6f:26:1e:1b:c2:47:c3:d3:ec:e4:

22:5d:1e:92:84:9f:4a:f4:30:02:de:0a:22:dc:df:

3b:ec:ef:19:e8:eb:95:68:50:7a:95:13:14:74:71:

35:89:6f:4a:cb:61:42:5d:b6:12:ef:64:a0:b0:2a:
                    95:fd
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:

CE:07:5B:01:AD:1E:AB:C2:E7:6F:E4:C5:67:21:B0:6F:B6:39:03:8C
            X509v3 Authority Key Identifier:

keyid:CE:07:5B:01:AD:1E:AB:C2:E7:6F:E4:C5:67:21:B0:6F:B6:39:03:8
C

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
```

Σχήμα 23. Περιεχόμενο του πιστοποιητικού του server (συνέχεια)

```
        88:91:31:b3:0b:6a:9e:1b:de:d2:9e:b6:0f:ae:70:8c:a2:73:
        25:93:ff:56:25:0f:1b:51:70:ff:2d:78:ee:88:d3:0d:ab:98:
        0e:1b:fc:0b:7c:47:cd:0f:87:ea:e0:88:63:84:fe:40:f7:f4:
        c7:fe:fc:17:58:74:c1:6d:c4:83:3a:f9:c2:18:e4:f4:cc:84:
        0a:33:b8:46:84:ac:5f:68:4d:16:77:61:04:fa:78:d9:1c:9f:
        e5:12:83:9a:91:37:08:17:ef:ad:d7:7e:21:f9:52:f2:f5:84:
        3a:d3:a8:4b:92:b2:a0:7a:50:e5:be:6e:77:dd:78:61:d6:e5:
        b4:33:b9:1b:b8:fd:59:df:60:18:ae:d9:3b:45:63:fb:ad:af:
        2a:25:d3:a6:77:c6:eb:df:9d:ff:f3:4a:e3:c8:20:37:c3:36:
        9f:82:c2:3e:6a:5a:88:c8:e0:62:d0:de:bc:b8:f1:68:c5:e6:
        31:0b:c8:02:67:91:fa:88:2e:e8:26:a7:f6:ea:89:6a:d2:f6:
        25:bd:37:77:ed:52:d4:7f:04:1b:d3:ac:fa:fc:67:1c:cc:6f:
        4f:7e:c2:aa:e9:13:a8:26:76:15:b6:4e:79:11:e0:61:13:c6:
        9f:d1:33:d6:fc:75:a4:b6:8b:73:55:57:44:c8:e8:2d:f0:9f:
        39:44:7b:20
-----BEGIN CERTIFICATE-----
MIID+TCCAuGgAwIBAgIJAOaOoXc30CkAMA0GCSqGSIb3DQEBBQUAMIGSMQswCQYD
VQQGEwJHUjEPMA0GA1UECAwGR3JlZWNlMREwDwYDVQQHDAhUcmlwb2xpczEMMAoG
A1UECgwDVU9QMRkwFwYDVQQLDBBDb21wdXRlciBGYWN1bHR5MRYwFAYDVQQDDA1N
YW5vbGlzIEJvemlzMR4wHAYJKoZIhvcNAQkBFg9wY3N0MDkxOEB1b3AuZ3IwHhcN
MTEwMjEyMTgxMTUxWhcNMTQwMjExMTgxMTUxWjCBkjELMAkGA1UEBhMCR1IxDzAN
BgNVBAgMBkdyZWVjZTERMA8GA1UEBwwIVHJpcG9saXMxDDAKBgNVBAoMA1VPUDEZ
MBcGA1UECwwQQ29tcHV0ZXIgRmFjdWx0eTEWMBQGA1UEAwwNTWFub2xpcyBCb3pp
czEeMBwGCSqGSIb3DQEJARYPcGNzdDA5MThAdW9wLmdyMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAm9XEiFihy4Cqh+SG918KaI2FELjrxfBW48ZmtLoV
UrteLvsyhT+2ZbJ8jeuTiIbl9VOQdpZ4LtpDwl1z/Fi5Hws6Q0+sYxndLL1LX8G/
ecyRjdDr33x/QUtLSjst/BJcl62xjwmRNIYZsRWyaR1TFVGJasgiLyG2foRxG1mR
5zmebbzvOTFTzh9cC8C8by8GRS8nhdJtMgDWeuaQXquNSVlbB0mnp3iAd7e4ZJD/
GP2rEIScPLXqLoBaQX9R2hhd923deFHOf28mHhvCR8PT7OQiXR6ShJ9K9DAC3goi
3N877O8Z6OuVaFB6lRMUdHE1iW9Ky2FCXbYS72SgsCqV/QIDAQABo1AwTjAdBgNV
HQ4EFgQUzgdbAa0eq8Lnb+TFZyGwb7Y5A4wwHwYDVR0jBBgwFoAUzgdbAa0eq8Ln
b+TFZyGwb7Y5A4wwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAiJEx
swtqnhve0p62D65wjKJzJZP/ViUPG1Fw/y147ojTDauYDhv8C3xHzQ+H6uCIY4T+
QPf0x/78F1h0wW3Egzr5whjk9MyECjO4RoSsX2hNFndhBPp42Ryf5RKDmpE3CBfv
rdd+IflS8vWEOtOoS5KyoHpQ5b5ud914YdbltDO5G7j9Wd9gGK7ZO0Vj+62vKiXT
pnfG69+d//NK48ggN8M2n4LCPmpaiMjgYtDevLjxaMXmMQvIAmeR+ogu6Can9uqJ
atL2Jb03d+1S1H8EG9Os+vxnHMxvT37CqukTqCZ2FbZOeRHgYRPGn9Ez1vx1pLaL
c1VXRMjoLfCfOUR7IA==
-----END CERTIFICATE-----
```

Σχήμα 23. Περιεχόμενο του πιστοποιητικού του server (συνέχεια)

Από το παραπάνω σχήμα εξάγουμε την πληροφορία ότι η χώρα του κατόχου του πιστοποιητικού είναι η Ελλάδα (Greece) με διακριτικό Gr, η πόλη του είναι η Tripolis, το email του είναι pcst0918@uop.gr, ο οργανισμός είναι ο UOP, κλπ. Επίσης, φαίνονται τα 2048 bit του δημόσιου κλειδιού σε δεκαεξαδική μορφή.

Στο παρακάτω σχήμα φαίνεται το αρχείο log του προγράμματος, όπως καταγράφεται στη διάρκεια μιας συνεδρίας (session). Στο αρχείο αυτό καταγράφεται το ιστορικό των συνδέσεων και τυχόν προβλήματα που μπορεί να παρουσιαστούν κατά τη διάρκεια της επικοινωνίας του εξυπηρετητή με τον πελάτη SSL.

```
2011.02.12 21:13:40 LOG5[3028:2812]: Reading configuration from
file stunnel.conf

2011.02.12 21:13:40 LOG5[3028:2812]: Configuration successful

2011.02.12 21:13:40 LOG5[3028:2812]: No limit detected for the
number of clients

2011.02.12 21:13:40 LOG5[3028:2812]: stunnel 4.34 on x86-pc-
mingw32-gnu with OpenSSL 1.0.0a 1 Jun 2010

2011.02.12 21:13:40 LOG5[3028:2812]: Threading:WIN32 SSL:ENGINE
Sockets:SELECT,IPv6

2011.02.12 21:13:45 LOG5[3028:2884]: Service door_control
accepted connection from 127.0.0.1:4160

2011.02.12 21:13:45 LOG5[3028:2884]: connect_blocking: connected
192.168.0.3:1001

2011.02.12 21:13:45 LOG5[3028:2884]: Service door_control
connected remote server from 192.168.0.4:4161

2011.02.12 21:15:42 LOG5[3028:2884]: Connection closed: 1165
bytes sent to SSL, 233 bytes sent to socket

2011.02.12 21:16:02 LOG5[3028:3256]: Service door_control
accepted connection from 127.0.0.1:4202

2011.02.12 21:16:02 LOG5[3028:3256]: connect_blocking: connected
192.168.0.3:1001

2011.02.12 21:16:02 LOG5[3028:3256]: Service door_control
connected remote server from 192.168.0.4:4203

2011.02.12 21:16:17 LOG5[3028:3256]: Connection closed: 155
bytes sent to SSL, 31 bytes sent to socket
```

Σχήμα 24. Περιεχόμενο του αρχείου log του stunnel μετά από μια επιτυχημένη συνεδρία SSL

3.4. Ασφάλεια – γνωστές επιθέσεις

Κάθε πρωτόκολλο που σχεδιάζεται για χρήση με το TLS πρέπει να είναι προσεκτικά σχεδιασμένο έτσι, ώστε να μπορεί να αντισταθεί σε κάθε δυνατή επίθεση εναντίον του. Πρακτικά, αυτό σημαίνει ότι ο σχεδιαστής του πρωτοκόλλου πρέπει να γνωρίζει ποια χαρακτηριστικά ασφαλείας παρέχει το TLS, ώστε να μην βασίζεται σε χαρακτηριστικά ασφαλείας που δεν παρέχονται.

Για παράδειγμα, ο τύπος και το μήκος του πεδίου δεν είναι κρυπτογραφημένα. Εάν αυτή η πληροφορία είναι από την φύση της ευαίσθητη, οι σχεδιαστές της εφαρμογής ίσως θα πρέπει να κάνουν επιπλέον βήματα, π.χ. γέμισμα (padding) ή κίνηση κενών πακέτων για κάλυψη της πληροφορίας (cover traffic) για να ελαχιστοποιήσουν τη διαρροή πληροφορίας.

Τα SSL και το TLS λαμβάνουν μια σειρά από μέτρα ασφαλείας όπως:

- Προστασία κατά της χρήσης παλαιότερης και λιγότερο ασφαλούς έκδοσης του πρωτοκόλλου ή ασθενέστερων αλγορίθμων κρυπτογράφησης
- Αρίθμηση των πεδίων της εφαρμογής κατά σειρά και χρήση αυτού του αριθμού στους κώδικες αυθεντικοποίησης μηνύματος (MAC)
- Χρήση αλγόριθμου message digest με κλειδί (ώστε μόνον ο κάτοχος του κλειδιού να μπορεί να ελέγξει την τιμή MAC). Η κατασκευή του HMAC που υποστηρίζεται από τα σετ κρυπτογράφησης του TLS, παρουσιάζεται στο σχετικό κείμενο RFC 2104.
- Το μήνυμα που τελειώνει τη διαδικασία χειραψίας (Finished) στέλνει μια τιμή hash από όλα τα μηνύματα χειραψίας που ανταλλάχθηκαν, την οποία βλέπουν και τα δύο μέρη.
- Στο TLS μόνον, η ψευδοτυχαία συνάρτηση διαχωρίζει τα εισερχόμενα δεδομένα στα δύο και επεξεργάζεται το καθένα τμήμα με διαφορετικό αλγόριθμο κατακερματισμού (MD5 και SHA-1), έπειτα για να δημιουργήσει το MAC προσθέτει τα δύο με πράξη XOR. Αυτή η διαδικασία παρέχει προστασία ακόμα και εάν ένας από αυτούς τους αλγόριθμους βρεθεί να είναι ευάλωτος σε επιθέσεις.
- Η έκδοση 3 του SSL βελτίωσε την έκδοση 2, προσθέτοντας αλγόριθμους που βασίζονται στο SSH-1 και υποστήριξη για αυθεντικοποίηση με πιστοποιητικά.

Τον Αύγουστο του 2009, ανακαλύφθηκε ένα κενό ασφαλείας στη διαδικασία επίτευξης συμφωνίας εκ νέου μεταξύ των μερών (renegotiation), η οποία μπορούσε να οδηγήσει σε επιθέσεις έγχυσης απλού κειμένου στην έκδοση 3 του SSL και σε όλες τις τρέχουσες εκδόσεις του TLS. Για παράδειγμα, επιτρέπει στον επιτιθέμενο να παρεμβαίνει στην αρχή της συνομιλίας του πελάτη με τον εξυπηρετητή διαδικτύου (web server) και να ταιριάζει τις αιτήσεις τους σε μια σύνδεση https. Ο επιτιθέμενος δεν μπορεί να αποκρυπτογραφήσει τη συνομιλία πελάτη με εξυπηρετητή, έτσι διαφέρει η επίθεση αυτή από τις

επιθέσεις τύπου man-in-the-middle. Μια βραχυπρόθεσμη λύση για τους web servers είναι να σταματήσουν να υποστηρίζουν τη διαδικασία renegotiation, η οποία συνήθως δεν απαιτεί άλλες αλλαγές, εκτός και αν χρησιμοποιείται αυθεντικοποίηση με πιστοποιητικό πελάτη. Για την εξάλειψη αυτού του κενού ασφαλείας προτάθηκε μια επέκταση του TLS για ένδειξη της διαδικασίας renegotiation και απαιτεί ο πελάτης και ο server να περιλαμβάνουν και να επαληθεύουν πληροφορία σχετικά με τις προηγούμενες χειραψίες, σε κάθε χειραψία τύπου renegotiation. Όταν ο χρήστης δεν προσέχει την ένδειξη του φυλλομετρητή ότι η συνεδρία είναι ασφαλής (συνήθως ένα εικονίδιο κλειδαριάς), το κενό ασφαλείας μπορεί να μετατραπεί σε επίθεση τύπου man-in-the-middle. Η επέκταση αυτή έγινε προτεινόμενο πρότυπο στο σχετικό κείμενο RFC 5746.

Υπάρχουν επίσης επιθέσεις όχι κατά του ίδιου του πρωτοκόλλου, αλλά κατά υλοποιήσεων του. Μερικές από αυτές είναι:

- Οι περισσότεροι οργανισμοί έκδοσης πιστοποιητικών (CAs) δεν καθορίζουν με ακρίβεια την τιμή basicConstrains CA = FALSE για τους κόμβους φύλλων και πολλοί φυλλομετρητές και άλλες υλοποιήσεις του SSL (περιλαμβανομένου του Internet Explorer, Konqueror, OpenSSL, κ.α.) δεν ελέγχουν αυτό το πεδίο. Αυτό μπορεί να αξιοποιηθεί από τις επιθέσεις man-in-the-middle σε όλες τις πιθανές συνδέσεις SSL.
- Μερικές υλοποιήσεις (περιλαμβανομένου προγενέστερων εκδόσεων του Microsoft Cryptographic API, Network Security Services και GnuTLS) σταματάνε να διαβάζουν τους χαρακτήρες που ακολουθούν τον κενό χαρακτήρα στο όνομα πεδίου του πιστοποιητικού. Το γεγονός αυτό μπορεί να αξιοποιηθεί για να ξεγελάσει τον πελάτη και να διαβάσει το πιστοποιητικό σαν να ήταν ένα που προέρχεται από το αυθεντικό site, για παράδειγμα το paypal.com\0.attacker.com θα λαμβάνονταν λανθασμένα ως paypal.com και όχι attacker.com.

Η ασφάλεια της έκδοσης 2 του SSL, μπορεί να παραβιαστεί με διάφορους τρόπους:

- Ίδια κρυπτογραφικά κλειδιά χρησιμοποιούνται για την αυθεντικοποίηση και την κρυπτογράφηση.
- Η έκδοση 2 του SSL έχει αδύναμη κατασκευή του MAC που χρησιμοποιεί την συνάρτηση κατακερματισμού MD5 με ένα μυστικό πρόθεμα, γεγονός

που την καθιστά ευάλωτη σε επιθέσεις επέκτασης μήκους (length extension attacks).

- Το SSL v2 δεν έχει προστασία για τη χειραψία, που σημαίνει ότι μια επίθεση τύπου man-in-the-middle downgrade μπορεί να γίνει χωρίς να ανιχνευτεί.
- Το SSL v2 χρησιμοποιεί το TCP connection close για να δείξει το τέλος των δεδομένων. Αυτό σημαίνει ότι είναι δυνατές επιθέσεις συναλλαγής (transaction attacks). Ο επιτιθέμενος απλά πλαστογραφεί ένα TCP FIN, αφήνοντας τον αποδέκτη απληροφόρητο για το παράτυπο τέλος του μηνύματος δεδομένων (η έκδοση 3 επιλύει το πρόβλημα με ένα ιδιαίτερο μήνυμα κλεισίματος).
- Το SSL v2 υποθέτει μια μόνο υπηρεσία και ένα πιστοποιητικού σταθερού domain, το οποίο συγκρούεται με το συνηθισμένο χαρακτηριστικό της εικονικής φιλοξενίας (virtual hosting) στους web servers. Αυτό σημαίνει ότι τα περισσότερα websites είναι πρακτικά ανίκανα να χρησιμοποιήσουν SSL. Το TLS/SNI το επιδιορθώνει αυτό, αλλά δεν έχει αναπτυχθεί ακόμη σε web servers.

Η έκδοση 2 του SSL είναι απενεργοποιημένη στις αρχικές ρυθμίσεις των Internet Explorer, Mozilla Firefox 2 και 3 και Safari. Αφού στείλει ένα μήνυμα TLS ClientHello, εάν ο Mozilla Firefox βρει ότι ο server δεν μπορεί να ολοκληρώσει την χειραψία, θα προσπαθήσει να υιοθετήσει το SSL 3.0 ClientHello στο format του SSL v2 για να μεγιστοποιήσει την πιθανότητα επιτυχημένης χειραψίας με παλαιότερους servers. Η υποστήριξη για το SSL v2 και τα αδύναμα κλειδιά των 40 και 56 bit, έχουν εγκαταλειφθεί εντελώς από την έκδοση 9.5 του Opera.

4. Το πρωτόκολλο SSH (secure shell)

Το Secure Shell (SSH) είναι ένα δικτυακό πρωτόκολλο που επιτρέπει τη μεταφορά δεδομένων, χρησιμοποιώντας ένα ασφαλές κανάλι μεταξύ δύο δικτυακών συσκευών. Οι δύο κύριες εκδόσεις του αναφέρονται ως SSH-1 και SSH-2. Χρησιμοποιήθηκε αρχικά σε συστήματα Linux και Unix ως εργαλείο πρόσβασης σε λογαριασμούς του συστήματος, αντικαθιστώντας το Telnet και άλλα ανασφαλή εργαλεία πρόσβασης στη γραμμή εντολών του λειτουργικού, τα οποία στέλνουν ευαίσθητες πληροφορίες μη κρυπτογραφημένες, καθιστώντας τα πρωτόκολλα αυτά ευάλωτα στην ανάλυση σε επίπεδο πακέτου. Η κρυπτογράφηση που εφαρμόζεται από το SSH παρέχει εμπιστευτικότητα και ακεραιότητα των δεδομένων που διακινούνται σε ένα ανασφαλές δίκτυο όπως το Internet.

Το SSH χρησιμοποιεί κρυπτογραφία δημοσίου κλειδιού για να αυθεντικοποιήσει τον απομακρυσμένο υπολογιστή και να επιτρέψει σε αυτόν να αυθεντικοποιήσει τον χρήστη, αν αυτό είναι απαραίτητο. Το SSH χρησιμοποιείται συνήθως για να συνδεθεί κάποιος σε απομακρυσμένο υπολογιστή και να εκτελέσει εντολές, αλλά επίσης υποστηρίζονται λειτουργία ενθυλάκωσης (tunneling), προώθηση θυρών TCP και συνδέσεων X11. Μπορεί επίσης, να μεταφέρει αρχεία χρησιμοποιώντας τα πρωτόκολλα SFTP και SCP. Το SSH χρησιμοποιεί το μοντέλο εξυπηρετητή – πελάτη. Η θύρα 22 του TCP έχει αντιστοιχηθεί στους εξυπηρετητές SSH.

Ένα πρόγραμμα πελάτη χρησιμοποιείται συνήθως για να εδραιωθούν συνδέσεις στο πρόγραμμα του εξυπηρετητή που δέχεται αιτήματα συνδέσεων. Και τα δύο προγράμματα αυτά υπάρχουν στα περισσότερα λειτουργικά συστήματα, όπως είναι το Mac OS X, Linux, FreeBSD, Solaris και OPENVMS. Υπάρχουν εκδόσεις προγραμμάτων εμπορικές, ελεύθερου λογισμικού και ανοικτού κώδικα, διαφορετικής πολυπλοκότητας και με διαφορετικό βαθμό ολοκλήρωσης του προτύπου.

Η πρώτη έκδοση του πρωτοκόλλου (SSH-1) σχεδιάστηκε το 1995 από τον Tatu Ylönen, έναν ερευνητή του Πανεπιστημίου Τεχνολογίας του Ελσίνκι στη Φινλανδία, παροτρυνόμενος από μια επίθεση στους κωδικούς πρόσβασης

(password sniffing attacks) στο πανεπιστήμιό του. Ο στόχος του SSH ήταν να αντικαταστήσει τα πρωτόκολλα rlogin, TELNET και rsh, τα οποία δεν παρείχαν ισχυρή αυθεντικοποίηση και εγγύηση της εμπιστευτικότητας. Ο Ylönen έδωσε την υλοποίησή του ως ελεύθερο λογισμικό τον Ιούλιο του 1995 και το εργαλείο αυτό γρήγορα έγινε δημοφιλές. Προς το τέλος του 1995, η βάση χρηστών αριθμούσε περί τους 20.000 χρήστες σε 50 χώρες. Στα τέλη του 1995, ο Ylönen ίδρυσε την εταιρεία SSH Communication Security για να προωθήσει και να αναπτύξει το SSH. Η αρχική έκδοση χρησιμοποίησε διάφορα κομμάτια ελεύθερου λογισμικού, αλλά οι μετέπειτα εκδόσεις εστιάστηκαν περισσότερο στο εμπορικό λογισμικό. Εκτιμάται ότι το 2000 το πρωτόκολλο είχε δύο εκατομμύρια χρήστες.

Το 1999, οι προγραμματιστές που θέλανε μια έκδοση ελεύθερου λογισμικού, μεταβήκανε στην παλιά έκδοση 1.2.12 του αρχικού προγράμματος SSH, η οποία ήταν η τελευταία έκδοση ανοικτού κώδικα. Από αυτόν τον κώδικα εξελίχθηκε το OSSH από τον Björn Grönvall. Λίγο αργότερα, οι προγραμματιστές του OpenBSD πήραν τον κώδικα του Grönvall και έκαναν εκτεταμένη δουλειά σε αυτόν, δημιουργώντας το OpenSSH, το οποίο ενσωματώθηκε στην έκδοση 2.6 του OpenBSD. Από αυτή την έκδοση δημιουργήθηκε ένας κλάδος μεταφοράς του OpenSSH σε άλλα λειτουργικά συστήματα. Ως το 2005, το OpenSSH ήταν η πιο δημοφιλής υλοποίηση του πρωτοκόλλου, που ήταν ενσωματωμένη σε πολλά λειτουργικά συστήματα. Τώρα συνεχίζει να αναπτύσσεται και υποστηρίζει τις εκδόσεις 1.x και 2.x. Το επίσημο όνομα της ομάδας εργασίας της IETF πάνω στη δεύτερη έκδοση του προτύπου, είναι Secsh. Το 2006 μια αναθεωρημένη έκδοση του SSH-2, έγινε αποδεκτή ως πρότυπο. Η έκδοση αυτή δεν ήταν συμβατή με την SSH-1, παρείχε όμως βελτιώσεις στα χαρακτηριστικά λειτουργίας και ασφάλειας.

Το πρότυπο του SSH-2 για χρήση στο Internet περιγράφεται σε μια σειρά από κείμενα RFC που εκδόθηκαν από την IETF. Το πρωτόκολλο μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές σε διάφορες πλατφόρμες, όπως τα Microsoft Windows, Apple Mac OS και Linux. Μερικές από τις εφαρμογές μπορεί να απαιτούν λειτουργίες που είναι διαθέσιμες σε συγκεκριμένες μόνο υλοποιήσεις εξυπηρετητών ή πελατών SSH. Από τις υλοποιήσεις του SSH, μόνο η OpenSSH μέχρι σήμερα, υποστηρίζει την υλοποίηση ενός Εικονικού Ιδιωτικού Δικτύου (VPN).

Συνοπτικά το SSH μπορεί να χρησιμοποιηθεί για:

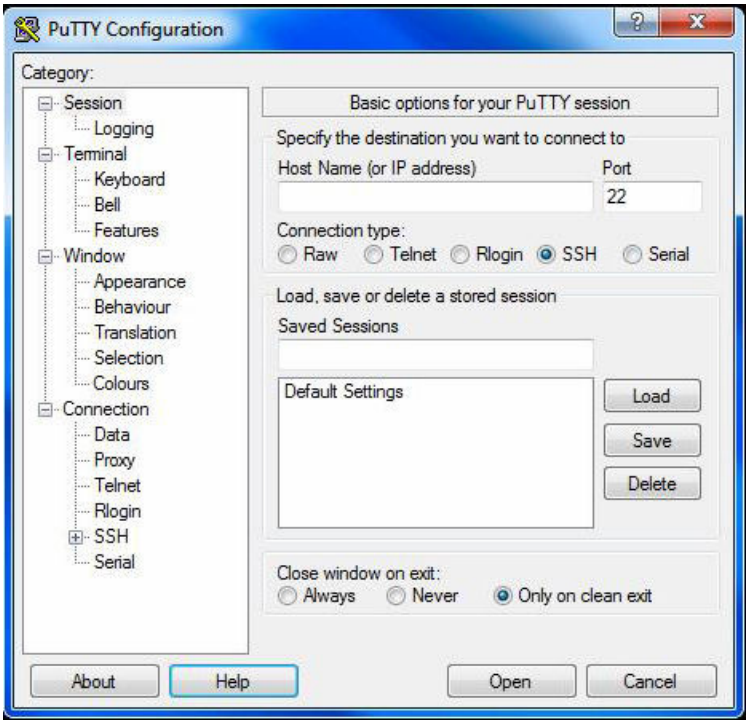
- σύνδεση σε λογαριασμό απομακρυσμένου υπολογιστή, αντικαθιστώντας το Telnet και το rlogin
- εκτέλεση μιας εντολής σε απομακρυσμένο ξενιστή, αντικαθιστώντας το rsh
- ασφαλή μεταφορά αρχείων
- λήψη αντιγράφων ασφαλείας, αντιγραφή και αντικατοπτρισμό αρχείων (mirroring) αποτελεσματικά και με ασφάλεια, σε συνδυασμό με το rsync
- προώθηση της κίνησης σε μια θύρα ή τη λειτουργία ενθυλάκωσης (tunneling)
- δημιουργία Εικονικών Ιδιωτικών Δικτύων (VPN)
- προώθηση του περιβάλλοντος X από έναν απομακρυσμένο υπολογιστή
- φυλλομετρητές (web browsers) διαμέσου μιας κρυπτογραφημένης σύνδεσης proxy που υποστηρίζει το πρωτόκολλο SOCKS
- να προσαρτηθεί με ασφάλεια ένας φάκελος σε έναν απομακρυσμένο εξυπηρετητή σαν ένα σύστημα αρχείων σε τοπικό υπολογιστή χρησιμοποιώντας το SSHFS
- αυτόματη απομακρυσμένη επίβλεψη και διαχείριση (monitoring and management) σε server με μία από τις παραπάνω τεχνικές και μηχανισμούς.

Πέρα των ανωτέρω χρήσεων, υπάρχουν διάφοροι μηχανισμοί μεταφοράς αρχείων με χρήση των πρωτοκόλλων Secure Shell. Το SSH File Transfer Protocol (SFTP) είναι μια ασφαλέστερη εναλλακτική στο FTP. Το SCP έχει εξελιχθεί από το RCP σε συνδυασμό με το SSH. Τέλος, το πρωτόκολλο FISH (Files transferred over Shell protocol) εκδόθηκε το 1998 και ήταν εξέλιξη της χρήσης εντολών του φλοιού του UNIX πάνω σε ένα ασφαλές κανάλι SSH.

Για τα παραπάνω πρωτόκολλα δεν έχουν αναπτυχθεί πρότυπα της IETF. Μια σειρά από προτάσεις της IETF έχουν γίνει για το SFTP, αλλά σταμάτησαν το 2006, εξαιτίας του προβλήματος ότι το SFTP είναι κατ' ουσία ένα σύστημα αρχείων.

Ένα από τα πιο γνωστά προγράμματα πελάτη του ssh είναι το PuTTY [44]. Είναι ανοικτού κώδικα και μπορεί να αντικαταστήσει το telnet, εκεί όπου απαιτείται ασφαλής σύνδεση για την απομακρυσμένη εκτέλεση εντολών. Μια

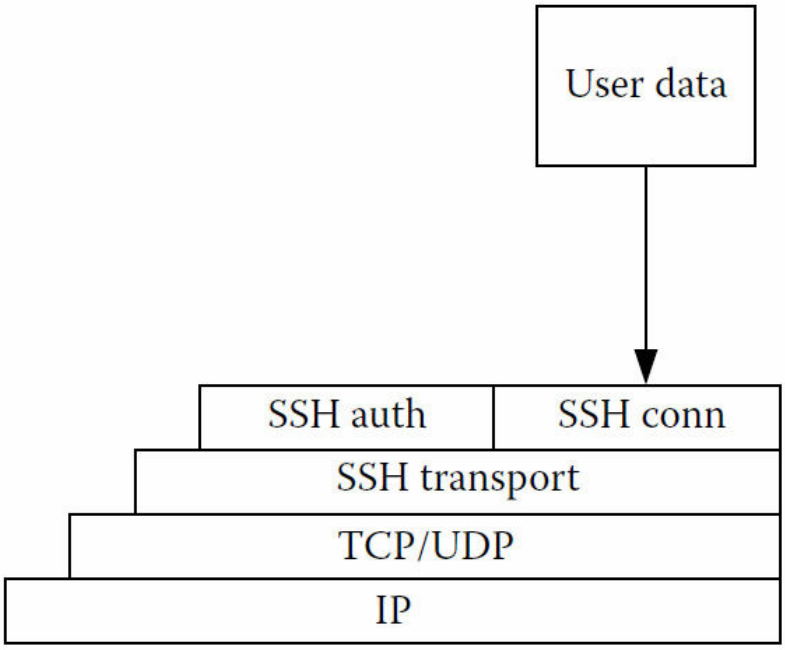
εικόνα του κύριου παραθύρου της εφαρμογής φαίνεται στο παρακάτω σχήμα. Οι κύριες δυνατότητές του, μεταξύ άλλων, είναι ο έλεγχος των κλειδιών κρυπτογράφησης του SSH, η υποστήριξη του SFTP και SCP μέσω γραμμής εντολών, καθώς και η δυνατότητα προώθησης επικοινωνίας μέσω SSH (port forwarding).



Σχήμα 25. Εικόνα από το πρόγραμμα πελάτη SSH PuTTY

4.1. Αρχιτεκτονική του πρωτοκόλλου

Η έκδοση 2 του SSH έχει μια εσωτερική αρχιτεκτονική που ορίζεται στο κείμενο RFC 4251, με σαφώς διαχωρισμένα επίπεδα (layers), τα οποία φαίνονται στο παρακάτω σχήμα και είναι:



Σχήμα 26. Αρχιτεκτονική του πρωτοκόλλου SSH

α) Το επίπεδο μεταφοράς (transport layer) που περιγράφεται στο RFC 4253. Αυτό το επίπεδο αναλαμβάνει την ανταλλαγή του αρχικού κλειδιού και αυθεντικοποίηση του server και ορίζει την επαλήθευση κρυπτογράφησης, συμπίεσης και ακεραιότητας. Παρέχει στο ανώτερο επίπεδο μία διεπαφή (interface) λήψης και αποστολής πακέτων απλών κειμένων (plaintexts) μέχρι και 32768 bytes το καθένα (παραπάνω μπορεί να το επιτρέπει η συγκεκριμένη υλοποίηση). Το επίπεδο μεταφοράς φροντίζει επίσης, για επανανταλλαγή των κλειδιών συνήθως μετά από μεταφορά 1 GB δεδομένων ή μετά την πάροδο μίας ώρας (όποιο από τα δύο συμβεί νωρίτερα).

β) Το επίπεδο αυθεντικοποίησης χρήστη (user authentication layer). Το επίπεδο αυτό αναλαμβάνει την αυθεντικοποίηση του πελάτη (client authentication) και παρέχει έναν αριθμό από μεθόδους αυθεντικοποίησης. Η διαδικασία αυθεντικοποίησης εκκινείται από τον πελάτη, όταν καλείται ο χρήστης να εισάγει τον κωδικό πρόσβασης. Συνήθως αυτό γίνεται από το πρόγραμμα SSH του πελάτη και όχι από τον server. Ο server σπάνια απαντά σε αιτήσεις αυθεντικοποίησης του πελάτη. Οι συνήθεις μέθοδοι αυθεντικοποίησης περιλαμβάνουν τα παρακάτω:

- Κωδικός πρόσβασης (password): Μέθοδος απευθείας αυθεντικοποίησης με κωδικό, που παρέχει δυνατότητα αλλαγής του password. Αυτή η μέθοδος δεν υλοποιείται από όλα τα προγράμματα.
- Δημόσιο κλειδί (public key): Μέθοδος αυθεντικοποίησης δημοσίου κλειδιού, που συνήθως υποστηρίζει ζευγάρια κλειδιών DSA και RSA, ενώ κάποιες υλοποιήσεις και πιστοποιητικά X.509.
- Διαδραστικά (keyboard interactive): Ευπροσάρμοστη μέθοδος στην οποία ο server στέλνει μία ή περισσότερες προσκλήσεις για παροχή πληροφοριών και ο πελάτης τις δείχνει στον χρήστη και στέλνει πίσω τις πληροφορίες που πληκτρολογεί αυτός. Χρησιμοποιείται για να παρέχει αυθεντικοποίηση με κωδικό μίας χρήσης, όπως το S/key και SecurID. Μερικές φορές οδηγεί σε αδυναμία σύνδεσης με τον πελάτη που υποστηρίζει μόνο την απλή μέθοδο αυθεντικοποίησης κωδικού πρόσβασης (password authentication), σε ρυθμίσεις του OpenSSH όπου ο μηχανισμός αυθεντικοποίησης του ξενιστή είναι ο PAM (Pluggable Authentication Modules).
- GSSAPI: Παρέχουν ένα εκτεταμένο σχήμα που παρέχει αυθεντικοποίηση SSH, χρησιμοποιώντας εξωτερικούς μηχανισμούς όπως το Kerberos 5 ή το NTLM, δίδοντας δυνατότητα υπογραφής στις συνεδρίες του SSH. Αυτές οι μέθοδοι συνήθως υλοποιούνται από εμπορικό λογισμικό SSH για χρήση σε οργανισμούς, παρόλο που και το OpenSSH έχει μια υλοποίηση που υποστηρίζει το GSSAPI.

γ) Το επίπεδο σύνδεσης (connection layer). Το επίπεδο αυτό ορίζει την έννοια των καναλιών, αιτημάτων για κανάλι (channel request) και γενικών αιτημάτων (global request), χρησιμοποιώντας τις υπηρεσίες του SSH που παρέχονται. Μια σύνδεση SSH μπορεί να εξυπηρετεί πολλαπλά κανάλια ταυτόχρονα, καθένα από τα οποία μεταφέρει δεδομένα προς δύο κατευθύνσεις. Τα αιτήματα για κανάλια χρησιμοποιούνται για την αναμετάδοση εκτός πλαισίου δεδομένων που αφορούν στο κανάλι, όπως είναι η αλλαγή του μεγέθους του παραθύρου ενός τερματικού ή ο κώδικας εξόδου μιας διαδικασίας από την πλευρά του εξυπηρετητή (server). Το πρόγραμμα του πελάτη του SSH ζητάει την προώθηση της θύρας στην πλευρά του server χρησιμοποιώντας ένα γενικό αίτημα. Οι συνήθεις τύποι καναλιών περιλαμβάνουν:

- Φλοιός Shell για τερματικά, αιτήματα SFTP και exec (περιλαμβανομένου μεταφορές SCP)
- Απευθείας TCP-IP για προωθημένες συνδέσεις πελάτη σε εξυπηρετητή.
- Προωθημένες TCP-IP για προωθημένες συνδέσεις εξυπηρετητή σε πελάτη.
- Το νέο πεδίο που ονομάζεται SSHFP στο DNS παρέχει ένα αποτύπωμα του δημοσίου κλειδιού του ξενιστή (host) με σκοπό την επαλήθευση της ταυτότητας του ξενιστή.

Η ανοικτή αρχιτεκτονική παρέχει σημαντική ευελιξία, επιτρέποντας το SSH να χρησιμοποιηθεί σε μια πλειάδα εφαρμογών πέρα από τον ασφαλή φλοιό (secure shell). Η λειτουργικότητα του επιπέδου μεταφοράς είναι συγκρίσιμη με αυτή του TLS (Transport Layer Security). Το επίπεδο αυθεντικοποίησης του χρήστη είναι αρκετά επεκτάσιμο με διάφορες μεθόδους αυθεντικοποίησης. Το επίπεδο σύνδεσης παρέχει τη δυνατότητα πολυπλεξίας πολλών δευτερευουσών συνεδριών (sessions) σε μία μοναδική σύνδεση SSH, ένα χαρακτηριστικό αντίστοιχο του πρωτοκόλλου BEEP (Blocks Extensible Exchange Protocol) που δεν διατίθεται στο TLS. Το BEEP [45] είναι ένα πλαίσιο για τη δημιουργία δικτυακών πρωτοκόλλων εφαρμογών. Περιέχει ένα πυρήνα για την επίτευξη ασύγχρονων συνδέσεων και μπορεί να χρησιμοποιηθεί τόσο για μηνύματα κειμένου, όσο και για δυαδικά (binary), τα οποία ανταλλάσσουν οντότητες εφαρμογών ενός χρήστη.

4.2. Ασφάλεια – γνωστές επιθέσεις

Καθόσον το SSH-1 έχει ενδογενή προβλήματα στον σχεδιασμό του που το καθιστούν ευάλωτο σε επιθέσεις τύπου άνθρωπος στη μέση (man-in-the-middle attacks), είναι πλέον γενικώς αποδεκτό ότι θα πρέπει να αποφεύγεται η χρήση του και να μην επιτρέπεται η μετάπτωση στο SSH-1 για χάρη συμβατότητας παλαιότερων συστημάτων. Παρόλο που οι περισσότεροι σύγχρονοι servers υποστηρίζουν SSH-2, μερικοί οργανισμοί χρησιμοποιούν ακόμη λογισμικό που δεν το υποστηρίζει και έτσι η χρήση του SSH-1 δεν μπορεί να αποφευχθεί. Σε όλες τις εκδόσεις του SSH είναι σημαντικό να επαληθευτούν τα δημόσια κλειδιά των ξενιστών (hosts) πριν αυτά γίνουν αποδεκτά από τον χρήστη, διότι σε αντίθετη περίπτωση ο επιτιθέμενος μπορεί

να καταφέρει να υποκλέψει το μεταδιδόμενο κωδικό πρόσβασης επιτρέποντας επιθέσεις τύπου man-in-the-middle [30].

Μια σειρά από επιθέσεις αποκάλυψης απλού κειμένου στην υλοποίηση OpenSSH δημοσιεύτηκαν από τους Albrecht, Paterson και Watson το 2009 [13]. Στις επιθέσεις αυτές μπορεί να αποκαλυφθούν και να επαληθευτούν 14 bits του απλού κειμένου από ένα επιλεγμένο block του κρυπτοκειμένου με πιθανότητα 2^{-14} και 32 bits του απλού κειμένου με πιθανότητα 2^{-18}. Οι επιθέσεις αυτές γίνονται στην προεπιλεγμένη ρύθμιση των 128 bit κρυπταλγορίθμου τμήματος σε λειτουργία CBC (cipher block chaining). Η δυνατότητα αυτών των επιθέσεων προέρχεται από ένα κενό ασφαλείας στον σχεδιασμό του πρωτοκόλλου στο κείμενο RFC που περιγράφει το SSH BPP (Binary Packet Protocol). Το BPP είναι το τμήμα του πρωτοκόλλου SSH που είναι υπεύθυνο για να παρέχει υπηρεσίες εμπιστευτικότητας και ακεραιότητας σε όλα τα μηνύματα που ανταλλάσσονται σε μια σύνδεση SSH. Καταρχήν, το SSH έχει ένα κρυπτογραφημένο πεδίο μήκους στο πρώτο τμήμα του κρυπτοκειμένου, το οποίο χρησιμοποιείται για να προσδιορίσει πόσα δεδομένα αναμένονται για ένα συγκεκριμένο πακέτο και πρέπει να ληφθούν υπόψη στον υπολογισμό της τιμής MAC, πριν αυτή είναι δυνατό να επαληθευτεί. Εν συνεχεία, η εξάρτηση από τον τρόπο λειτουργίας του CBC (ακόμα και με αλυσίδα αρχικών διανυσμάτων IV), επιτρέπει στον επιτιθέμενο να εισάγει ένα block επιλεγμένου κρυπτοκειμένου σε ένα νέο πακέτο BPP ως το πρώτο block του πακέτου και η αποκρυπτογράφηση αυτού του επιλεγμένου τμήματος στην αρχική του θέση να σχετίζεται με ένα γνωστό τρόπο με την αποκρυπτογράφηση του νέου πακέτου. Αυτοί οι δύο παράγοντες, σε συνδυασμό με την ικανότητα του επιτιθέμενου να εισάγει δεδομένα ανά block σε μια σύνδεση SSH και να ανιχνεύσει πότε συμβαίνει ένα λάθος στην επαλήθευση της τιμής MAC, επιτρέπουν στον επιτιθέμενο να αποκαλύψει κάποια bits του απλού κειμένου που σχετίζονται με ένα επιλεγμένο κρυπτοκείμενο, παρατηρώντας πόσα blocks χρειάζονται για να προκαλέσουν ένα σφάλμα στην τιμή του MAC. Οι έλεγχοι μήκους του OpenSSH σπάνια είναι ένας παράγοντας που μειώνει τις πιθανότητες επιτυχίας αυτών των επιθέσεων.

Οι επιθέσεις αυτές έγιναν σε μια υλοποίηση του SSH BPP, η οποία είχε αποδειχθεί ασφαλής, σε προγενέστερη εργασία των M. Bellare,T. Kohno και C. Namprempre (2004). Ωστόσο, στις παραδοχές του μοντέλου το οποίο

χρησιμοποιήθηκε για την απόδειξη της ασφάλειας, δεν ελήφθησαν ορισμένοι παράγοντες υπόψη. Για παράδειγμα, ενώ αναγνωρίζεται ότι η λειτουργία αποκρυπτογράφησης στο SSH δεν είναι μονοδιάστατη και μπορεί να αποτύχει με διάφορους τρόπους, το μοντέλο ασφαλείας δεν κάνει διάκριση μεταξύ των διάφορων τύπων σφάλματος, όταν αυτά αναφέρονται στο άλλο μέρος της σύνδεσης. Επιπλέον, το μοντέλο δεν λαμβάνει σαφώς υπόψη του το γεγονός ότι το ποσό δεδομένων που χρειάζεται για να συμπληρωθεί η λειτουργία αποκρυπτογράφησης προσδιορίζεται από τα δεδομένα που πρέπει να αποκρυπτογραφηθούν (το μήκος πεδίου). Δυστυχώς, φαίνεται ότι οι κρυπτογραφικές υλοποιήσεις πραγματικών συστημάτων είναι πιο πολύπλοκες από τα σημερινά μοντέλα ασφαλείας που περιγράφουν το SSH.

Το 2010 παρουσιάστηκε από τους Keneth G. Paterson και Gaven J. Watson [14], ένα εκτεταμένο μοντέλο ανάλυσης που αποδεικνύει την ασφάλεια του SSH σε λειτουργία CTR (counter mode). Το μοντέλο αυτό περιγράφει το SSH-CTR σχετιζόμενο στενά με τις προδιαγραφές των κειμένων RFC και την υλοποίηση OpenSSH. Η προσέγγιση του μοντέλου λαμβάνει υπόψη της και τις επιθέσεις που αναφέρθηκαν στην προηγούμενη παράγραφο και το γεγονός αυτό βοηθά στην εξάλειψη του κενού ανάμεσα στην τυπική ανάλυση ασφάλειας του SSH και στον τρόπο που θα έπρεπε να λειτουργεί το SSH και υλοποιείται στην πράξη.

Επίσης, η προσέγγιση αυτή που παρουσιάστηκε στην προαναφερόμενη δημοσίευση [14], είναι μια προσπάθεια να μεγενθύνει το εύρος της αποδείξιμης ασφάλειας, ώστε να ενοποιήσει τις λεπτομέρειες των κρυπτογραφικών υλοποιήσεων. Θεωρεί δε ένα μεγαλύτερο και πιο ρεαλιστικό σύνολο από τρόπους αλληλεπίδρασης του επιτιθέμενου με το πρωτόκολλο, από ότι η προηγούμενη ανάλυση. Η προσέγγιση αυτή καταγράφει περισσότερα από τα κρυπτογραφικά χαρακτηριστικά του SSH BBP, όπως αυτά που σχετίζονται με την εξάρτηση από το απλό κείμενο, τη σειρά αποκρυπτογράφησης των bytes και τον τρόπο μοντελοποίησης των σφαλμάτων που μπορεί να προκύψουν κατά την επεξεργασία αποκρυπτογράφησης.

5. Επίλογος - Συμπεράσματα

Η εργασία αυτή επικεντρώθηκε στην παρουσίαση των πιο ευρέως χρησιμοποιούμενων πρωτοκόλλων αυθεντικοποίησης, των δυνατοτήτων και βασικών αρχών της σχεδίασής τους, καθώς και σε παραδείγματα χρήσης τους και θέματα ασφαλείας των ίδιων ή των υλοποιήσεών τους.

Στο πλαίσιο αυτό παρουσιάστηκε η παραμετροποίηση του προγράμματος stunnel για τη δημιουργία ενός ασφαλούς καναλιού SSL για τη διακίνηση των δεδομένων μιας εφαρμογής server-client παρακολούθησης και διαχείρισης απομακρυσμένων σημείων εισόδου (θυρών). Στην εφαρμογή αυτή η χρήση κρυπτογραφημένου καναλιού είναι απαραίτητη, καθόσον τα δεδομένα της κατάστασης των θυρών, όπως και των εντολών ανοίγματος – κλεισίματος είναι μη κρυπτογραφημένα και οποιοσδήποτε έχει πρόσβαση στο δίκτυο θα μπορούσε να δώσει εντολές στην εφαρμογή ή να καταγράψει την τρέχουσα κατάσταση. Η παραμετροποίηση αυτή μπορεί να αποτελέσει οδηγό και για άλλες εφαρμογές server –client σε περιβάλλον win32.

Επίσης, στην εργασία αυτή δεν καλύφθηκαν όλες οι εξελίξεις, αλλά και τα ειδικότερα θέματα των πρωτοκόλλων αυθεντικοποίησης, κατά τη χρήση τους σε συγκεκριμένες εφαρμογές.

Επιπρόσθετα, καταβλήθηκε προσπάθεια να είναι όσο το δυνατόν πιο ενημερωμένη ως προς τις τελευταίες εξελίξεις, όσον αφορά στην ασφάλεια των υλοποιήσεων και στην κυκλοφορία νέων διαφοροποιημένων εκδόσεων των πρωτοκόλλων.

Τέλος, από την ενασχόληση με τα θέματα ασφαλείας των πρωτοκόλλων αυθεντικοποίησης γίνεται φανερό ότι είναι δύσκολο να υιοθετηθεί και να μελετηθεί ένα πλήρες μοντέλο ασφαλείας, το οποίο να αποδεικνύει ένα πρωτόκολλο ως απόλυτα ασφαλές. Αυτό συμβαίνει όχι μόνο διότι ισχυροί κρυπταλγόριθμοι μπορεί στο μέλλον να καταστούν επισφαλείς με την εξέλιξη της τεχνολογίας, αλλά και γιατί οι μηχανισμοί αυθεντικοποίησης που εφαρμόζονται στα σύγχρονα συστήματα, βασίζονται σε μια πλειάδα δικτυακών

πρωτοκόλλων και υλοποιήσεών τους σε διαφορετικά λειτουργικά συστήματα και υλικό (hardware).

6. Βιβλιογραφία

[01] Protocols for authentication and key Establishment, Colin Boyd & Anish Mathuria, 2003 Springer-Verlag.

[02] Mechanics of user Identification and Authentication, Dobromir Todorov, 2007 Auerbach Publications

[03] Paterson, Kenneth G, “A cryptographic tour of the IPsec standards”, Information Security Technical Report, Volume 11, Issue 2, 2006, Pages 72-81

[04] Paterson, Kenneth G & Yau, Arnold K.L, "Cryptography in theory and practice: The case of encryption in IPsec" , Eurocrypt 2006, Lecture Notes in Computer Science Vol. 4004. Berlin, Pages 12–29.

[05] Jean Paul Degabriele & Kenneth G. Paterson, ”Attacking the IPsec standards in Encryption-only Configurations”, 2007 IEEE Symposium on Security and Privacy (SP’07)

[06] S. Kent και R. Atkinson, “Security Architecture for the Internet Protocol”, RFC 2401, Νοέμβριος 1998

[07] D Harkins, D Carrel, “The Internet Key Exchange (IKE)”, RFC 2409, Νοέμβριος 1998

[08] S. Kent, ”IP Authentication Header.” RFC 4302, Δεκέμβριος 2005

[09] S. Kent, “IP Encapsulating Security Payload (ESP).” RFC 4303, Δεκέμβριος 2005

[10] V. Manral, “Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)” RFC4835, Απρίλιος 2007

[11] J. Schlyter, W. Griffin, “Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints”, RFC 4255, Ιανουάριος 2006

[12] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, RFC 5246, Αύγουστος 2008.

[13] Martin R. Albrecht, Kenneth G. Paterson and Gaven J. Watson, “Plaintext Recovery Attacks Against SSH”, 2009 30th IEEE Symposium on Security and Privacy.

[14] Kenneth G. Paterson and Gaven J. Watson, “Plaintext-Dependent Decryption: A Formal Security Treatment of SSH-CTR”, Eurocrypt 2010.

[15] Arlond K. L. Yau, Side Channel Analyses of CBC Mode Encryption, Information Security Group, Department of Mathematics, Royal Holloway, University of London 2009

[16] H. Krawczyk, ”SIGMA: The ‘SIGn-and-Mac’ Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols.” In D. Boneh (ed.),Advances in Cryptology –CRYPTO 2003,LNCS Vol.2729, Springer-Verlag,2001,pp.310-331

[17] Β.Α. Κάτος – Γ.Χ. Στεφανίδης, Τεχνικές Κρυπτογραφίας και Κρυπτανάλυσης, 2003, Εκδόσεις Ζυγός

7. Αναφορές στο διαδίκτυο

[18] http://en.wikipedia.org/wiki/IPsec
[19] http://en.wikipedia.org/wiki/Security_association
[20] http://en.wikipedia.org/wiki/Internet_Key_Exchange
[21] http://en.wikipedia.org/wiki/Secure_Sockets_Layer
[22] http://en.wikipedia.org/wiki/X.509
[23] http://en.wikipedia.org/wiki/ISAKMP
[24] http://en.wikipedia.org/wiki/Network_address_translation
[25] http://en.wikipedia.org/wiki/NAT_traversal
[26] http://en.wikipedia.org/wiki/NAT-T
[27] http://datatracker.ietf.org/wg/mobike/charter/
[28] http://wiki.strongswan.org/projects/strongswan/wiki/NetworkManager
[29] http://wiki.openswan.org/index.php/Openswan/CiscoPIX
[30] http://en.wikipedia.org/wiki/Secure_Shell
[31] http://en.wikipedia.org/wiki/Files_transferred_over_shell_protocol
[32] http://en.wikipedia.org/wiki/S/Key
[33] http://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority
[34] http://doxfer.webmin.com/Webmin/SSLTunnels
[35] http://stunnel.org/static/stunnel.html
[36] http://www.iana.org/assignments/port-numbers
[37] http://www.openssl.org/docs/HOWTO/certificates.txt
[38] http://www.openssl.org/docs/HOWTO/keys.txt
[39] http://en.wikipedia.org/wiki/X.509
[40] http://en.wikipedia.org/wiki/FIPS_140-2
[41] http://en.wikipedia.org/wiki/Common_Criteria
[42] http://www.cs.ucdavis.edu/~rogaway/papers/ad.pdf
[43] http://www.niksula.cs.hut.fi/~sjsavola/SoN/essay.html
[44] http://en.wikipedia.org/wiki/PuTTY
[45] http://en.wikipedia.org/wiki/BEEP
[46] http://www.rfc-editor.org/rfc/rfc5084.txt
[47] http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange