ΠΕΡΙΕΧΟΜΕΝΑ

VPN (Virtual Private Network)

ΠΕΡΙΛΗΨΗ

Ένα virtual private network (VPN) είναι ένα δίκτυο υπολογιστών στο οποίο οι συνδέσεις μεταξύ των κόμβων είναι από ανοικτές συνδέσεις ή εικονικά κυκλώματα, σε μεγαλύτερα δίκτυα (όπως το Διαδίκτυο), σε αντιδιαστολή με το ιδιωτικό δίκτυο. Τα πρωτόκολλα Link Layer του ιδεατού (virtual) δικτύου θεωρούνται "ανοιχτά" μέσω του δικτύου μεταφορών. Μια κοινή εφαρμογή είναι να προστατέψει τις επικοινωνίες μέσω του δημόσιου Διαδικτύου. Όμως, ένα VPN δεν χρειάζεται να έχει ιδιότητες ασφαλείας όπως την αυθεντικοποίηση ή κρυπτογράφηση του περιεχομένου. Για παράδειγμα, το VPN μπορεί επίσης να χρησιμοποιηθεί για να διαχωρίσει την κυκλοφορία διαφορετικών χρηστών σε ένα δίκτυο με τα χαρακτηριστικά γνωρίσματα ισχυρής ασφάλειας, ή για να παρέχει την πρόσβαση σε ένα δίκτυο μέσω των προσαρμοσμένων ή ιδιωτικών μηχανισμών δρομολόγησης.

User-visible PPVPN services

Layer 1 services

Virtual private wire and private line services (VPWS and VPLS)

Και στις δύο υπηρεσίες, ο προμηθευτής δεν προσφέρει ένα σύνολο που καθοδηγείται ή που γεφυρώνεται το δίκτυο, αλλά συστατικά από τα οποία ο πελάτης μπορεί να χτίσει τα πελάτης-διαχειριζόμενα δίκτυα. Τα VPWS είναι point-to-point, ενώ τα VPLS μπορεί να είναι point-to-multipoint. Μπορούν να είναι κυκλώματα επιπέδου 1 χωρίς καμία δομή δεδομένων.

Ο πελάτης καθορίζει τη γενική εξυπηρέτηση πελατών VPN, η οποία μπορεί επίσης να περιλάβει τη δρομολόγηση, το γεφύρωμα, ή τα στοιχεία δικτύων των διαφόρων hosts.

Layer 2 services

Virtual LAN

Virtual private LAN service (VPLS)

Αναπτυγμένο από το IEEE, το VLAN επιτρέπει σε πολλαπλά LANs να μοιραστεί την κοινή ζεύξη. Το VLAN περιλαμβάνει συχνά μόνο τις customer-owned εγκαταστάσεις. Το τελευταίο είναι μία τεχνολογία επιπέδου 1 που υποστηρίζει τις τοπολογίες point-to-point αλλά και point-to-multipoint. Η μέθοδος που συζητείται εδώ περιέχει τις επιπέδου 2 τεχνολογίες όπως η ζεύξη του τοπικού LAN 802.1d και 802.1q που τρέχει πέρα από τις μεταφορές.

Όπως χρησιμοποιείται σε αυτό το πλαίσιο, ένα VPLS είναι ένα επιπέδου 2 PPVPN, παρά μια ιδιωτική γραμμή, όπως η πλήρης λειτουργία ενός παραδοσιακού

δικτύου τοπικής περιοχής (τοπικό LAN). Από μια σκοπιά χρηστών, ένα VPLS το καθιστά πιθανό να διασυνδέσει διάφορα τμήματα του τοπικού LAN πέρα από έναν packet-switched, ή οπτικό πυρήνα προμηθευτών, που κάνει τα μακρινά τμήματα του τοπικού LAN να συμπεριφερθούν ως ένα ενιαίο τοπικό LAN.

Pseudo wire (PW)

Το PW είναι παρόμοιο με το VPWS, αλλά μπορεί να παρέχει τα διαφορετικά L2 πρωτόκολλα. Χαρακτηριστικά, η διεπαφή του είναι ένα WAN πρωτόκολλο όπως το Asynchronous Transfer Mode ή το Frame Relay.

IP-only LAN-like service (IPLS)

Ένα υποσύνολο απο VPLS. Οι συσκευές CE πρέπει να έχουν L3 ικανότητες. Το IPLS παρουσιάζει τα πακέτα παρά τα πλαίσια. Μπορεί να υποστηρίξει IPv4 ή IPv6.

L3 PPVPN architectures

Εδώ, εμφανίζονται οι κύριες αρχιτεκτονικές για PPVPN δίκτυα. Αρχικά υπάρχουν οι διπλές διευθύνσεις PE disambiguates σε μια ενιαία περίπτωση δρομολόγησης, και έπειτα ο εικονικός δρομολογητής, στον οποίο το PE περιέχει έναν εικονικό δρομολογητή ανά VPN.

BGP/MPLS PPVPN

Στη μέθοδο που καθορίζεται από το RFC 2547, οι επεκτάσεις BGP "διαφημίζουν" τις διαδρομές στη IPv4 οικογένεια διευθύνσεων VPN, οι οποίες είναι της μορφής 12 σειρών bits, αρχίζοντας με μια διαδρομή 8 bits Distinguisher (RD) και τελειώνοντας με μια διεύθυνση 4 bits IPv4.

Virtual router PPVPN

Η εικονική αρχιτεκτονική δρομολογητών, σε αντιδιαστολή με τις τεχνικές BGP/MPLS, δεν απαιτεί καμία τροποποίηση στα υπάρχοντα πρωτόκολλα δρομολόγησης όπως το BGP. Ο πελάτης που λειτουργεί πάνω σε ένα VPN είναι απολύτως αρμόδιος για το διάστημα διευθύνσεων. Στα διάφορα MPLS tunnels, το διαφορετικό PPVPNs δεν χρειάζεται τα distinguishers δρομολόγησης.

Categorizing VPN security models

Από τη σκοπιά ασφάλειας, το VPN είτε εμπιστεύεται το ίδιο το δίκτυο, είτε πρέπει να επιβάλει την ασφάλεια με τους μηχανισμούς το ίδιο στο VPN. Εκτός αν το εμπιστευόμενο δίκτυο τρέχει μόνο μεταξύ των φυσικά ασφαλών περιοχών, και τα εμπιστευόμενα και ασφαλή πρότυπα χρειάζονται έναν μηχανισμό επικύρωσης για τους χρήστες για να αποκτήσουν πρόσβαση στο VPN.

Authentication before VPN connection

Σε έναν γνωστό εμπιστευμένο χρήστη, μερικές φορές μόνο κατά την χρησιμοποίηση των εμπιστευμένων συσκευών, μπορούν να παρασχεθούν τα

κατάλληλα προνόμια ασφάλειας για να έχει πρόσβαση στους πόρους. Οι κεντρικοί υπολογιστές μπορούν επίσης να πρέπει να επικυρωθούν για να ενώσουν το VPN.

Μια ευρεία ποικιλία των μηχανισμών επικύρωσης υπάρχει. Το VPN μπορεί να εφαρμόσει την επικύρωση στις συσκευές συμπεριλαμβανομένων των αντιπυρικών ζωνών για να έχει πρόσβαση στις πύλες. Μπορούν να χρησιμοποιήσουν τους κωδικούς πρόσβασης ή τις κρυπτογραφικές μεθόδους. Η ισχυρή επικύρωση περιλαμβάνει το συνδυασμό του συστήματος κρυπτογραφίας με έναν άλλο μηχανισμό επικύρωσης. Ο μηχανισμός επικύρωσης μπορεί να απαιτήσει τη ρητή δράση χρηστών, ή μπορεί να ενσωματωθεί στον πελάτη VPN ή τον τερματικό σταθμό.

Trusted delivery networks

Το εμπιστευόμενο VPN δεν χρησιμοποιεί κρυπτογραφικό να ανοίξει, και στηρίζεται αντ' αυτού στην ασφάλεια του δικτύου ενός ενιαίου προμηθευτή για να προστατεύσει την κυκλοφορία. Από μία άποψη, διαμορφώνουν στην συνηθισμένη εργασία δικτύων το σύστημα-διοίκησης.

- Το Multi-Protocol Label Switching (MPLS) χρησιμοποιείται συχνά για να επικαλύψει το VPN με τον έλεγχο υπηρεσιών ενός εμπιστευμένου δικτύου παράδοσης.
- Το Layer 2 Tunneling Protocol (L2TP) που είναι ένας συμβιβασμός που παίρνει τα καλά χαρακτηριστικά γνωρίσματα από κάθε ένα, για δύο ιδιόκτητα πρωτόκολλα VPN: Επίπεδο 2 της Cisco (L2F) και από σημείο σε σημείο ανοίγοντας το πρωτόκολλο της Microsoft (PPTP).

Security mechanisms

Εξασφαλίζουν τα κρυπτογραφικά ανοίγοντας πρωτόκολλα χρήσης VPN για να παρέχουν την προοριζόμενη εμπιστευτικότητα, την επικύρωση αποστολέων, και την ακεραιότητα μηνυμάτων για να επιτύχει τη μυστικότητα. Όταν επιλέγονται κατάλληλα, εφαρμοσμένα, και λειτουργημένα, τέτοιες τεχνικές μπορούν να παρέχουν τις ασφαλείς επικοινωνίες πέρα από τα ακάλυπτα δίκτυα.

Τα πρωτόκολλα ότι VPN περιλαμβάνουν τα εξής:

- IPsec (ασφάλεια IP), όπου χρησιμοποιείται συνήθως στο IPv4, αλλά και στο IPv6.
- SSL/TLS, που χρησιμοποιούνται είτε για να ανοίξει την ολόκληρη κυκλοφορία δικτύων, όπως στο πρόγραμμα OpenVPN, είτε για την ασφάλεια, όπου είναι, ουσιαστικά ένα πληρεξούσιο Ιστού και καλείται SSL VPN.
- OpenVPN, όπου είναι ένα ανοιχτό πρότυπο VPN. Αυτό μπορεί να τρέξει χρησιμοποιώντας το UDP. Οι πελάτες και οι κεντρικοί υπολογιστές είναι διαθέσιμοι για όλα τα σημαντικά λειτουργικά συστήματα.
- DTLS, που χρησιμοποιήθηκε από τη Cisco για ένα προϊόν επόμενης γενεάς VPN καλώντας το Cisco AnyConnect VPN.

- SSTP από τη Microsoft που εισάγεται στον κεντρικό υπολογιστή των Windows 2008 και Vista Service Pack 1. Τα SSTP tunnels PPP ή L2TP βασίζονται στο SSL 3.0 κανάλι.
- L2TPv3 (Layer 2 Tunneling Protocol version 3).
- VPN Quarantine. Η μηχανή πελατών στο τέλος ενός VPN θα μπορούσε να είναι μια απειλή και μια πηγή επίθεσης. Αυτό δεν έχει καμία σύνδεση με το σχέδιο VPN και την περισσότερη άδεια προμηθευτών VPN στη διοίκηση συστημάτων που εξασφαλίζει.
- MPVPN (Multi Path Virtual Private Network).
- Cisco VPN, όπου είναι ένα ιδιόκτητο VPN που χρησιμοποιείται από πολλές συσκευές υλικού της Cisco.

MOBIKE PROTOCOL

ΠΕΡΙΛΗΨΗ

Σ΄ αυτό το κεφάλαιο περιγράφεται το πρωτόκολλο MOBIKE, μια mobility και multihoming επέκταση του Internet Key Exchange (IKEv2). Το MOBIKE επιτρέπει σε IP διευθύνσεις, οι οποίες συνδέονται με IKEv2 και IPSEC Security Associations, να αλλάζουν. Ένας κινητός χρήστης ο οποίος συνδέεται πάνω σε ένα VPN, θα μπορούσε να χρησιμοποιήσει για να κρατήσει τη σύνδεση με την πύλη του VPN που δραστηριοποιείται, ενώ μετακινείται από ένα τόπο σε άλλο. Επίσης, ένας χρήστης θα μπορούσε να χρησιμοποιήσει το MOBIKE για να μετατρέψει την κίνηση (ή το traffic) του δικτύου σε μια διαφορετική διεπαφή (interface), εάν θα ήθελε να σταματήσει την εργασία.

ΕΙΣΑΓΩΓΗ

Το IKEv2 χρησιμοποιείται για την εκτέλεση αμοιβαίας αυθεντικοποίησης, καθώς και τη δημιουργία και τη διατήρηση του IPsec Security Associations (SA). Το IKEv2 πρωτόκολο, το IKE SAs και το tunnel mode IPsec SAs δημιουργήθηκαν ανάμεσα στις IP διευθύνσεις, τα οποία χρησιμοποιούνται όταν εγκαθίσταται το IKE_SA. Αυτές οι IP διευθύνσεις χρησιμοποιούνται σαν εξωτερικές διευθύνσεις για τα πακέτα IPsec. Συνήθως, αυτό δεν είναι δυνατό να αλλάξει αυτές τις διευθύνσεις, μετά την δημιουργία του IKE_SA.

Υπάρχουν περιπτώσεις όπου οι διευθύνσεις IP μπορεί να αλλάξουν. Ένα παράδειγμα είναι η κινητικότητα (mobility), όπου ένας χρήστης μεταβαίνει σε μια σειρά αλλαγών σε ένα σημείο του δικτύου, όπου και λαμβάνει μια νέα διεύθυνση IP. Ένα άλλο παράδειγμα είναι ένας χρήστης multihoming, όπου θα ήθελε να αλλάξει μια διαφορετική διεπαφή (interface), εάν θα ήθελε να σταματήσει την εργασια για κάποιον λόγο.

Αν και το πρόβλημα μπορεί να λυθεί με τη δημιουργία νέων IKE και IPsec SAs, όταν οι διευθύνσεις πρέπει να αλλάξουν, μπορεί να μην είναι βέλτιστο για διάφορους λόγους. Σε ορισμένες περιπτώσεις, η δημιουργία ενός νέου IKE_SA μπορεί να απαιτήσει αλληλεπίδραση του χρήστη για την πιστοποίηση της ταυτότητας, όπως η εισαγωγή ενός κωδικού. Δημιουργόντας ένα νέο SA, συχνά περιλαμβάνονται ακριβής υπολογισμοί και, ενδεχομένως, ένα μεγάλο αριθμό απο round-trips. Για τους λόγους αυτούς, χρειάζεται ένας νέος μηχανισμός για την ενημέρωση των IP διευθύνσεων των IPsec IKE και IPsec SAs.

Το κύριο σενάριο για το MOBIKE είναι που θα επιτρέπει την απομακρυσμένη πρόσβαση των χρηστών στο VPN ενώ μετακινούνται από ένα τόπο σε έναν άλλο, χωρίς την επανεγκατάσταση όλων των ασφαλών ενώσεων με την πύλη VPN. Για παράδειγμα, ένας χρήστης θα μπορούσε να ξεκινήσει τις εργασίες του από το Ethernet στο γραφείο, στη συνέχεια να αποσύνδεε το laptop του, και μετά να μεταβούσε στο γραφείο του μέσω του ασύρματου LAN. Όταν ο χρήστης έφευγε απο το γραφείο, ο φορητός υπολογιστής θα μπορούσε να αρχίσει να χρησιμοποιώντας το General Packet Radio Service (GPRS). Όταν ο χρήστης φτάνει στο σπίτι, το laptop θα

μπορούσε να το αλλάξει προς το ασύρματο LAN του σπιτιου. Το MOBIKE ενημερώνει μόνο τις εξωτερικές διευθύνσεις των IPsec SAS, και τις διευθύνσεις σε άλλα δίκτυα τα οποία μπορούν να μείνουν χωρίς αλλαγές. Έτσι, η κινητικότητα μπορεί να είναι κυρίως για τις αόρατες τις συνδέσεις τους που χρησιμοποιούν VPN.

ΣΚΟΠΟΣ ΚΑΙ ΠΕΡΙΟΡΙΣΜΟΙ (Scope and Limitations)

Η κινητικότητα (mobility) όπου υποστηρίζεται στο MOBIKE , επιτρέπει και στα δύο στοιχεία (Scope και Limitations) να μετακινηθούν, αλλά δεν παρέχει ένα μηχανισμό "συνάντησης", το οποίο θα άφηνε την ταυτόχρονη κίνηση των δύο στοιχείων όταν το IKE_SA εγκαθίσταται για πρώτη φορά. Ως εκ τούτου, το MOBIKE είναι οι πλέον κατάλληλο για καταστάσεις κατά τις οποίες η διεύθυνση σε τελικό στάδιο είναι σχετικά σταθερή και μπορεί να ανακαλυφθεί με χρήση μηχανισμών όπως το DNS.

Το MOBIKE επιτρέπει και στα δύο μέρη να είναι multihomed. Ωστόσο, μόνο ένα ζευγάρι των διευθύνσεων χρησιμοποιείται για την SA σε μια στιγμή.

Το MOBIKE ακολουθεί την πρακτική του IKEv2, όπου το μήνυμα απάντησης αποστέλλεται στην ίδια διεύθυνση και πόρτα από το οποίο το αιτημα έχει ληφθει. Αυτό σημαίνει ότι το MOBIKE δεν λειτουργεί πάνω από τα ζεύγη διευθύνσεων που παρέχουν μονής κατεύθυνσης συνδετικότητας.

Η βασική έκδοση του MOBIKE πρωτοκόλλου δεν καλύπτει όλους τους εν δυνάμει μελλοντικής χρήσης σεναρίων, όπως τον τρόπο μεταφοράς, την εφαρμογή για την εξασφάλιση SCTP.

Protocol Overview

Basic Operation

Το MOBIKE επιτρέπει και στις δυο πλευρές να έχουν διάφορετικές διευθύνσεις, και να υπάρχουν μέχρι N * M ζεύγη διευθύνσεων IP που θα μπορούσαν ενδεχομένως να χρησιμοποιηθούν. Η απόφαση για την οποία αυτά τα ζεύγη χρησιμοποιούνται πρέπει να λαμβάνει υπόψη διάφορους παράγοντες. Πρώτον, οι πλευρές μπορεί να έχουν προτιμήσεις για το ποιά διεπαφή θα πρέπει να χρησιμοποιείται και δεύτερον, η απόφαση περιορίζεται από το γεγονός ότι ορισμένα ζεύγη μπορεί να μην λειτουργούν σε όλες τις IP εκδόσεις και έτσι να υπάρχουν διακοπές στο δίκτυο, προβλήματα σε τοπικό σύνδεσμο σε κάθε άκρο, και ούτω καθεξής.

Το MOBIKE λύνει αυτό το πρόβλημα με τη λήψη μιας απλής προσέγγισης: το κομμάτι που ξεκίνησε το IKE_SA (ο "πελάτης" σε μια απομακρυσμένη πρόσβαση VPN), είναι υπεύθυνο για τη λήψη αποφάσεων στο οποίο το ζεύγος διευθύνσεων χρησιμοποιείται για το Ipsec SAS για τη συλλογή της πληροφορίας που πρέπει να καταβληθεί για την παρούσα απόφαση. Το άλλο μέρος ("πύλη" σε μια απομακρυσμένη πρόσβαση VPN) είναι απλά αυτό που λέει ο διαχειρηστής, αλλά δεν ενημερώνει το IPsec SAS μέχρι να λάβει το μήνυμα από τον ίδιο για να το πράξει. Η προσέγγιση αυτή ισχύει και για τις διευθύνσεις του IPsec SAS. Στην IKE_SA περίπτωση, η ανταλλαγή μπορεί να αποφασίσει ποιές διευθύνσεις χρησιμοποιούνται.

Συνήθως λαμβάνονται κάποιες αποφάσεις σχετικά με την λειτουργία του IKEv2: ο διαχειρίστής αποφασίζει ποιές διευθύνσεις χρησιμοποιεί όταν επικοινωνεί ο responder. Επίσης, έχει νόημα, ιδίως όταν ο initiator είναι ένας κινητός κόμβος: είναι

σε καλύτερη θέση να αποφασίσουν ποια από τις διευθύνσεις του δικτύου θα πρέπει να χρησιμοποιείται και για τις δύο κυκλοφορίες (upstream - downstream).

Οι λεπτομέρειες είναι αυτές που ευθύνονται για το πώς ακριβώς ο initiator λαμβάνει την απόφαση, τί πληροφορία χρησιμοποιείται για την παραγωγή του, τον τρόπο συλλογής των πληροφοριών, πώς οι προτιμήσεις επηρεάζουν μια απόφαση, και, πότε μια απόφαση πρέπει να άλλαξει σε μεγάλο βαθμό εκτός του πεδίου εφαρμογής του MOBIKE. Αυτό δεν σημαίνει ότι αυτές οι πληροφορίες είναι ασήμαντες. Το αντίθετο μάλιστα. Είναι πιθανό να είναι καθοριστικό, σε κάθε πραγματικό σύστημα. Ωστόσο, το MOBIKE ασχολείται με αυτά τα στοιχεία μόνο στο βαθμό που είναι ορατά σε IKEv2/IPsec μηνύματα που ανταλλάσσονται μεταξύ των peers (και ως εκ τούτου πρέπει να τυποποιημένα για τη διασφάλιση της διαλειτουργικότητας).

Πολλά από τα ζητήματα αυτά δεν αφορούν ειδικά το MOBIKE, αλλά είναι κοινά με τη χρήση των υπάρχοντων χρηστών (hosts) σε δυναμικά περιβάλλοντα ή με πρωτόκολλα όπως το Mobile IP [MIP4] [MIP6]. Μια σειρά από μηχανισμούς έχουν δημιουργηθεί ή έχουν ήδη αναπτυχθεί για την αντιμετώπιση αυτών των ζητημάτων. Για παράδειγμα, οι link-layer και IP-layer μηχανισμοί μπορούν να χρησιμοποιηθούν για να παρακολουθούν την θέση της σύνδεσης κατά την τοπική σύνδεση, την κυκλοφορία ανίχνευσης που καθορίζονται για τα δύο IPv4 και IPv6 σε [DNA4], [DNA6], και ούτω καθεξής.

Φυσικά, η ενημέρωση των διευθύνσεων των IPsec SASs πρέπει να λάβει υπόψη διάφορους παράγοντες ασφάλειας. Το MOBIKE περιλαμβάνει δύο χαρακτηριστικά που έχουν σχεδιαστεί για την αντιμετώπιση αυτών των σκέψεων. Πρώτον, ένας "return routability" έλεγχος όπου μπορεί να χρησιμοποιηθεί για την επαλήθευση της διευθύνσης που παρέχονται απο τον χρήστη. Αυτό καθιστά πιο δύσκολη την “πλημμύρα” σε τρίτους όταν υπάρχουν μεγάλα ποσά κυκλοφορίας. Δεύτερον, ένα “NAT prohibition” χαρακτηριστικό εξασφαλίζει ότι οι διευθύνσεις IP δεν έχουν τροποποιηθεί απο NATs, IPv4/IPv6, ή άλλα τέτοια παρόμοια αντικείμενα. Αυτή η δυνατότητα είναι ενεργοποιημένη μόνο απο NAT Traversal, όταν δεν χρησιμοποιείται.

Παράδειγμα πρωτόκολλο ανταλλαγής (Example Protocol Exchanges)

Ένα απλό σενάριο MOBIKE σε ένα κινητό παρουσιάζεται παρακάτω. Ο συμβολισμός στηρίζεται σε [IKEv2]. Επιπλέον, η πηγή / προορισμός (source/destination) διευθύνσεων IP και των ports που εμφανίζονται για κάθε πακέτο είναι: IP_I1, IP_I2, IP_R1, και IP_R2 όπου αντιπροσωπεύουν διευθύνσεις IP που χρησιμοποιούνται από τον initiator και τον responder.

```
   Initiator                  Responder
   -----------                -----------
1) (IP_I1:500 -> IP_R1:500)
   HDR, SAi1, KEi, Ni,
        N(NAT_DETECTION_SOURCE_IP),
        N(NAT_DETECTION_DESTINATION_IP)  -->

                           <--  (IP_R1:500 -> IP_I1:500)
                                HDR, SAr1, KEr, Nr,
                                     N(NAT_DETECTION_SOURCE_IP),
                                     N(NAT_DETECTION_DESTINATION_IP)
```

```
   2) (IP_I1:4500 -> IP_R1:4500)
      HDR, SK { IDi, CERT, AUTH,
                CP(CFG_REQUEST),
                SAi2, TSi, TSr,
                N(MOBIKE_SUPPORTED) }  -->

                              <--  (IP_R1:4500 -> IP_I1:4500)
                                   HDR, SK { IDr, CERT, AUTH,
                                             CP(CFG_REPLY),
                                             SAr2, TSi, TSr,
                                             N(MOBIKE_SUPPORTED) }
```

(Ο initiator παίρνει την πληροφορία από χαμηλότερα layers όπου είναι το συνδεδεμένο σημείο και η διεύθυνση έχει αλλάξει.)

```
3) (IP_I2:4500 -> IP_R1:4500)
      HDR, SK { N(UPDATE_SA_ADDRESSES),
                N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP) }  -->

                              <-- (IP_R1:4500 -> IP_I2:4500)
                                  HDR, SK { N(NAT_DETECTION_SOURCE_IP),
                                            N(NAT_DETECTION_DESTINATION_IP)}
```

(Ο Responder επαληθεύει ότι ο initiator του έδωσε μια σωστή IP διεύθυνση.)

```
  4)                          <-- (IP_R1:4500 -> IP_I2:4500)
                                  HDR, SK { N(COOKIE2) }

      (IP_I2:4500 -> IP_R1:4500)
      HDR, SK { N(COOKIE2) }  -->
```

Το Βήμα 1 είναι η κανονική IKE_INIT ανταλλαγή. Στο βήμα 2, οι χρήστες ενημερώνουν μεταξύ τους ότι υποστηρίζουν το MOBIKE. Στο βήμα 3, ο initiator προειδοποιεί για μια αλλαγή στη δική του διεύθυνση, και ενημερώνει τον responder σχετικά με την αποστολή INFORMATIONAL αίτηματος περιλαμβάνοντας το UPDATE_SA_ADDRESSES. Η αίτηση αποστέλλεται χρησιμοποιώντας τη νέα διεύθυνση IP. Σε αυτό το σημείο, πρέπει επίσης να αρχίσει να χρησιμοποιεί την νέα διεύθυνση ως πηγαία διεύθυνση στη δικιά του εξερχόμενη κυκλοφορία ESP. Όταν λαμβάνεται η UPDATE_SA_ADDRESSES ειδοποίηση, ο responder γράφει τη νέα διεύθυνση και, εάν απαιτείται από την πολιτική, εκτελεί έλεγχο προς αυτήν την διεύθυνση. Όταν αυτός ο έλεγχος ολοκληρωθεί (βήμα 4), ο responder αρχίζει να χρησιμοποιεί τη νέα διεύθυνση σαν προορισμό για την εξερχόμενη κυκλοφορία ESP.

Ένα άλλο πρωτόκολλο που τρέχει σε ένα multihoming σενάριο παρουσιάζεται παρακάτω. Σε αυτό το σενάριο, ο initiator έχει μία διεύθυνση, αλλά ο responder έχει δύο.

```
Initiator                  Responder
     -----------                -----------
   1) (IP_I1:500 -> IP_R1:500)
      HDR, SAi1, KEi, Ni,
           N(NAT_DETECTION_SOURCE_IP),
           N(NAT_DETECTION_DESTINATION_IP)  -->

                              <--  (IP_R1:500 -> IP_I1:500)
                                   HDR, SAr1, KEr, Nr,
```

```
                                        N(NAT_DETECTION_SOURCE_IP),
                                        N(NAT_DETECTION_DESTINATION_IP)

   2) (IP_I1:4500 -> IP_R1:4500)
      HDR, SK { IDi, CERT, AUTH,
                CP(CFG_REQUEST),
                SAi2, TSi, TSr,
                N(MOBIKE_SUPPORTED) }  -->

                              <--  (IP_R1:4500 -> IP_I1:4500)
                                   HDR, SK { IDr, CERT, AUTH,
                                             CP(CFG_REPLY),
                                             SAr2, TSi, TSr,
                                             N(MOBIKE_SUPPORTED),
                                             N(ADDITIONAL_IP4_ADDRESS)}
```

(Ο initiator επιλύει ένα πρόβλημα στο χρησιμοποιούμενο ζεύγος διεύθυνση.)

```
3) (IP_I1:4500 -> IP_R1:4500)
      HDR, SK { N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP) }  -->

      (IP_I1:4500 -> IP_R1:4500)
      HDR, SK { N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP) }  -->

      ...
```

(Τελικά, ο initiator δίνει την τρέχων διεύθυνσή και δοκιμάζει τα άλλα διαθέσιμα ζέυγοι διεύθυνσης).

```
4) (IP_I1:4500 -> IP_R2:4500)
      HDR, SK { N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP) }

                              <--  (IP_R2:4500 -> IP_I1:4500)
                                   HDR, SK {
N(NAT_DETECTION_SOURCE_IP),
                                        N(NAT_DETECTION_DESTINATION_IP)
}
```

(Αυτό λειτουργεί, και ο initiator ζητά από τον χρήστη να στραφεί σε νέες διευθύνσεις.)

```
5) (IP_I1:4500 -> IP_R2:4500)
      HDR, SK { N(UPDATE_SA_ADDRESSES),
                N(NAT_DETECTION_SOURCE_IP),
                N(NAT_DETECTION_DESTINATION_IP),
                N(COOKIE2) }  -->

                              <--  (IP_R2:4500 -> IP_I1:4500)
                                      HDR, SK {
                                      N(NAT_DETECTION_SOURCE_IP),

                                      N(NAT_DETECTION_DESTINATION_IP),
                                      N(COOKIE2) }
```

MOBIKE and Network Address Translation (NAT)

Σε ορισμένα σενάρια MOBIKE, το δίκτυο μπορεί να περιέχει NAT, ή πακέτα φίλτρων. Το NAT Traversal χαρακτηριστικό που καθορίζονται στο [IKEv2] επιτρέπει το IKEv2 να λειτουργεί μέσω NAT, σε πολλές περιπτώσεις, και μπορεί το MOBIKE να προβαίνει σε αυτή τη λειτουργικότητα: όταν οι διευθύνσεις που χρησιμοποιούνται για IPsec SAS έχουν αλλάξει, το MOBIKE μπορεί να ενεργοποιεί ή να απενεργοποιεί το IKEv2 NAT Traversal, όπως αυτό χρειάζεται.

Ωστόσο, υπάρχουν ορισμένοι περιορισμοί στο NAT, επειδή συνήθως εισαγάγει μια ασυμμετρία στο δίκτυο: μόνο τα πακέτα που προέρχονται από την "εντός" κατάσταση μπορούν να δημιουργηθούν. Αυτή η ασυμμετρία που οδηγεί το MOBIKE σε περιορισμούς για το τι μπορούμε να κάνουμε. Σε ένα συγκεκριμένο παράδειγμα, εξετάζουμε μια κατάσταση όπου δύο χρήστες έχουν μόνο μία διεύθυνση, και ο initiator βρίσκεται πίσω από το NAT. Εάν η διεύθυνση του responder αλλάζει, θα πρέπει να στείλει ένα πακέτο στον initiator, χρησιμοποιώντας τη νέα διεύθυνση. Ωστόσο, εάν το NAT είναι, για παράδειγμα, του κοινού " restricted cone" τύπου, αυτό δεν είναι δυνατό να γίνει. Το NAT θα μειώσει τα πακέτα που αποστέλλονται από τη νέα διεύθυνση (στην περίπτωση που δεν έχει την πρωτοβουλία έχει ήδη αποστείλει πακέτα σε αυτή τη διεύθυνση - που δεν μπορεί να το κάνει μέχρι να γνωρίζει την διεύθυνση).

Για απλούστερους λόγους, το MOBIKE δεν επιχειρεί να χειριστεί όλα τα πιθανά NAT σενάρια. Αντ 'αυτού, το MOBIKE υποθέτει ότι, αν υπάρχουν NAT, ο initiator είναι μέρος “πίσω” απο το NAT και η περίπτωση που η διεύθυνση του responder αλλάζει δεν υποστηρίζεται. Οι responders μπορούν επίσης να αγνοήσουν το NAT ή συγκεκριμένα είδη NAT. Ωστόσο, όταν έχει εμφανιστεί μία αλλαγή, όπως το χάσιμο της σύνδεσης, οι responders του MOBIKE θα εξακολουθήσουν να ενημερώνουν τον initiator για την αλλαγή αυτη. Ανάλογα με, το ακριβές είδος του NAT, το παραπάνω μπορεί ή όχι να το επιτύχει.

IPSEC

ΣΧΕΤΙΚΑ

Εξετάζοντας τις περιπτωσιολογικές μελέτες της χρήσης των διαφόρων μηχανισμών ασφάλειας στις εφαρμογές, πακέτων (sockets), και των επιπέδων μεταφοράς (transport layers), η τελική περιπτωσιολογική μελέτη μας θα είναι στο επίπεδο δικτύου (transport layer). Εδώ, θα εξετάσουμε το πρωτόκολλο ασφάλειας IP, συχνότερα γνωστό ως IPsec - μια ακολουθία πρωτοκόλλων που παρέχουν την ασφάλεια στο επίπεδο δικτύου. Για το IPsec υπάρχουν πολλά πράγματα να τυπωθούν, και γι΄αυτό το λόγο διαφορετικά μέρη περιγράφονται σε ένα πάρα πολλά RFCs. Στο συγκεκριμένο σημείο, θα συζητήθεί το IPsec σαν ένα ειδικό πλαίσιο όπου όλοι οι hosts στο Διαδίκτυο υποστηρίζουν IPsec. Αν και αυτό το πλαίσιο είναι πολλά έτη μακριά, απ'την θα απλοποιήσει τη συζήτηση και θα βοηθήσει να καταλάβει κανένας τα κύρια χαρακτηριστικά του IPsec. Δύο βασικά RFCs που υπάρχουν είναι το [RFC 2401], το οποίο περιγράφει τη γενική αρχιτεκτονική ασφάλειας IP και το [RFC 2411], η οποία παρέχει μια επισκόπηση της ακολουθίας πρωτοκόλλου IPsec και των εγγράφων που την περιγράφουν.

Πρίν μπούμε στις λεπτομέρειες του IPsec, ας πάμε πίσω και να εξετάσουμε τι σημαίνει παροχή στην ασφάλεια στο επίπεδο δικτύου. Ας δούμε πρώτα τι σημαίνει παροχή ως προς τη μυστικότητα του επιπέδου δικτύων. Το επίπεδο δικτύων θα παρείχε τη μυστικότητα, εάν όλα τα στοιχεία που πάρθηκαν από όλα τα διαγράμματα δεδομένων IP είχαν κρυπτογραφηθεί. Αυτό σημαίνει ότι όποτε ένας χρήστης host θέλει να στείλει ένα διάγραμμα δεδομένων, κρυπτογραφεί τον τομέα στοιχείων του διαγράμματος δεδομένων προκειμένου να τα "δημοσιεύσει" στο δίκτυο. Σε γενικές γραμμές, η κρυπτογράφηση θα μπορούσε να γίνει με τη συμμετρική βασική κρυπτογράφηση, με τη δημόσια βασική κρυπτογράφηση, ή με κλειδιά που έχουν *negotiated* χρησιμοποιώντας τη δημόσια βασική κρυπτογράφησης. Ο τομέας στοιχείων θα μπορούσε να είναι ένα τμήμα TCP, ένα τμήμα UDP, ένα ICMP μήνυμα, κ.λπ. Εάν μια τέτοια υπηρεσία επιπέδου δικτύου ήταν σε ισχύ, όλα τα στοιχεία που στέλνονται από τους hosts -- συμπεριλαμβανομένου του ηλεκτρονικού ταχυδρομείου, ιστοσελίδας, μηνύματα ελέγχου και διαχείρισης (όπως ICMP και το SNMP) -- θα κρυβόταν από οποιοδήποτε τρίτο που έχει μπει κρυφά στο δίκτυο (Εντούτοις, τα μή κρυπτογραφημένα στοιχεία θα μπορούσαν να κατασκοπευτούν στα σημεία πηγής και προορισμού των hosts). Κατά συνέπεια, μια τέτοια υπηρεσία θα παρείχε μια ορισμένη "γενική κάλυψη" για όλη τη κίνηση του δικτύου, και με αυτόν τον τρόπο θα έδινε σε όλους μας μια ορισμένη αίσθηση ασφαλείας.

Εκτός από τη μυστικότητα, το επίπεδο δικτύου μπορεί να παρέχει επίσης, την αυθεντικότητα της πηγής (source destination). Όταν ένας host προορισμού λαμβάνει ένα διάγραμμα δεδομένων IP με μια συγκεκριμένη διεύθυνση προέλευσης IP, μπορεί να επικυρώσει ή να αυθεντικοποιήσει την πηγή και να σιγουρευτεί ότι το διάγραμμα δεδομένων IP παρήχθει πράγματι από τον host με εκείνη την συγκεκριμένη διεύθυνση

προέλευσης IP. Μια τέτοια υπηρεσία αποτρέπει τους επιτιθεμένους από τις ψεύτικες διευθύνσεις IP.

Στην ακολουθία πρωτοκόλλου IPsec υπάρχουν δύο κύρια πρωτόκολλα: το πρωτόκολλο Authentication Header (AH) και το πρωτόκολλο Encapsulation Security Payload (ESP). Όταν ένας host πηγής στέλνει τα ασφαλή διαγράμματα δεδομένων σε έναν host προορισμού, χρησιμοποιεί είτε το AH είτε με το ESP πρωτόκολλο. Το πρωτόκολλο AH παρέχει την επικύρωση πηγής και την ακεραιότητα στοιχείων αλλά δεν παρέχει τη μυστικότητα. Το ESP πρωτόκολλο παρέχει την ακεραιότητα και τη μυστικότητα στοιχείων. Παρέχοντας περισσότερες υπηρεσίες, το ESP πρωτόκολλο είναι φυσικά πιό περίπλοκο και απαιτεί περισσότερη επεξεργασία από το πρωτόκολλο AH. Θα συζητήσουμε και τα δύο πρωτόκολλα παρακάτω.

Και το AH αλλά και το ESP πρωτόκολλο, πρίν στείλουν τα εξασφαλισμένα διαγράμματα δεδομένων από έναν host πηγής σε έναν host προορισμού, οι hosts πηγής και δικτύων συνδέονται και δημιουργούν μια λογική σύνδεση επιπέδου δικτύου. Αυτό το λογικό κανάλι καλείται security agreement (SA). Κατά συνέπεια, το IPsec μετασχηματίζει την χωρίς σύνδεση επιπέδου δικτύων του Διαδικτύου σε ένα επίπεδο με λογικές συνδέσεις! Η σύνδεση που καθορίζεται από ένα SA είναι μια μονοκατευθυντική σύνδεση, δηλαδή, είναι ομοιοκατευθυνόμενη. Εάν και οι δύο hosts θέλουν να στείλουν τα ασφαλή διαγράμματα δεδομένων ο ένας στον άλλο, κατόπιν δύο SAs (δηλ., λογικές συνδέσεις), είναι ανάγκη να καθιερωθεί μόνο μια σε κάθε κατεύθυνση. Ένα SA προσδιορίζεται μεμονωμένα απο τρία στοιχεία που αποτελείται από:

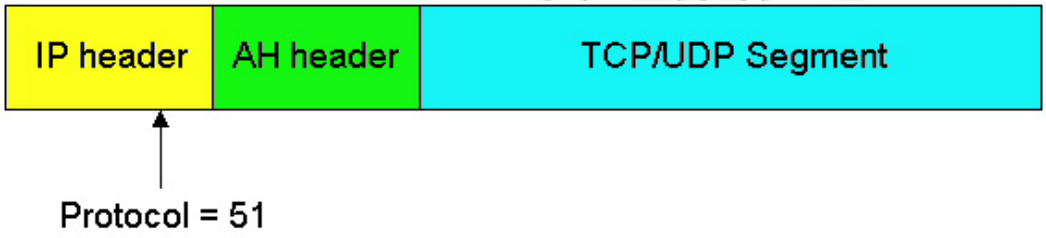
- ένα προσδιοριστικό πρωτόκολλου ασφάλειας (AH ή ESP)
- μία διεύθυνση πηγής IP για τη μονοκατευθυντική σύνδεση
- ένα τριανταδυάμπιτο (32-bit) προσδιοριστικό σύνδεσης αποκαλούμενο ως Security Paramter Index (SPI).

Για ένα δεδομένο SA (δηλαδή μια δεδομένη λογική σύνδεση από τον host της πηγής στον host προορισμού), κάθε διάγραμμα δεδομένων IPsec θα έχει έναν πρόσθετο τομέα για το SPI. Όλα τα διαγράμματα δεδομένων στο SA θα χρησιμοποιήσουν την ίδια τιμή SPI σε αυτόν τον τομέα.

Authentication Header (AH) Protocol

Όπως αναφέρεται παραπάνω, το πρωτόκολλο AH παρέχει τον προσδιορισμό του host της πηγής και την ακεραιότητα στοιχείων, αλλά όχι τη μυστικότητα. Όταν ένας host πηγής θέλει να στείλει ένα ή περισσότερα διαγράμματα δεδομένων σε έναν προορισμό, καθιερώνει αρχικά ένα SA με τον προορισμό. Μετά από μία σύνδεση με το SA, η πηγή μπορεί να στείλει τα εξασφαλισμένα διαγράμματα δεδομένων στον host προορισμού. Τα εξασφαλισμένα διαγράμματα δεδομένων περιλαμβάνουν την επιγραφή AH, που παρεμβάλλεται μεταξύ των αρχικών στοιχείων διαγραμμάτων δεδομένων IP (π.χ., ένα τμήμα TCP ή UDP) και της επιγραφής IP. Κατά συνέπεια η

επιγραφή AH αυξάνει τον αρχικό τομέα στοιχείων. Αυτός ο αυξημένος τομέας στοιχείων είναι τοποθετημένος ως τυποποιημένο διάγραμμα δεδομένων IP. Για τον τομέα πρωτοκόλλου στην επιγραφή IP, η τιμή 51 χρησιμοποιείται για να δείξει ότι το διάγραμμα δεδομένων περιλαμβάνει μια επιγραφή AH. Όταν ο host προορισμού λαμβάνει το διάγραμμα δεδομένων IP, σημειώνει το 51 στον τομέα πρωτοκόλλου, και επεξεργάζεται το διάγραμμα δεδομένων χρησιμοποιώντας το πρωτόκολλο AH. (Ο τομέας πρωτοκόλλου στο διάγραμμα δεδομένων IP χρησιμοποιείται για να διακρίνει μεταξύ UDP, του TCP, ICMP, κ.λπ.). Οι ενδιάμεσοι δρομολογητές επεξεργάζονται τα διαγράμματα δεδομένων ακριβώς όπως έχουν πάντα, εξετάζουν τη διεύθυνση προορισμού IP και καθοδηγούν τα διαγράμματα δεδομένων αναλόγως.



Θέση του AH header στο διάγραμμα δεδομένων IP.

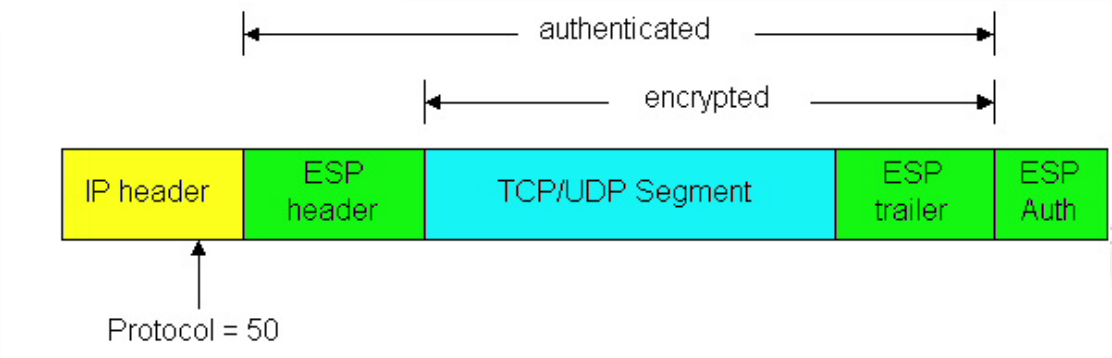
Το AH header περιλαμβάνει διάφορα πεδία, όπως:

- Το πεδίο **Next Header**, το οποίο έχει το ρόλο όπου το πεδίο πρωτοκόλλου έχει ένα συνηθισμένο διάγραμμα δεδομένων. Δείχνει εάν το στοιχείο μετά από το AH header είναι ένα τμήμα TCP, τμήμα UDP, τμήμα ICMP, κ.α. (Ο τομέας πρωτοκόλλου στο διάγραμμα δεδομένων χρησιμοποιείται τώρα για να δείξει το πρωτόκολλο AH, και έτσι αυτό δεν μπορεί πλέον να χρησιμοποιηθεί για να δείξει το πρωτόκολλο transport-layer).
- Το πεδίο **Security Parameter Index (SPI)**, είναι μια αυθαίρετη τριανταδυάμπιτη τιμή, που σε σχέση με την διεύθυνση προορισμού IP και το πρωτόκολλο ασφάλειας, προσδιορίζουν μεμονωμένα το SA για το διάγραμμα δεδομένων.
- Το πεδίο **Sequence Number, είναι** ένας τριανταδυάμπιτος τομέας που περιέχει έναν αριθμό ακολουθίας για κάθε διάγραμμα δεδομένων. Το πρωτόκολλο AH χρησιμοποιεί τους αριθμούς ακολουθίας για να αποτρέψει τις man-in-the-middle (κατ' ατόμου) επιθέσεις.
- Το πεδίο **Authentication Data,** είναι ένας τομέας μεταβλητού μήκους που περιέχει την υπογεγραμμένη σύνοψη μηνυμάτων (δηλαδή μία ψηφιακή υπογραφή) για αυτό το πακέτο. Το ψηφίο (digit) των μηνυμάτων υπολογίζεται πέρα από το αρχικό διάγραμμα δεδομένων IP, παρέχοντας με αυτόν τον τρόπο την επικύρωση των hosts πηγής και την ακεραιότητα διαγραμμάτων δεδομένων IP. Η ψηφιακή υπογραφή υπολογίζεται χρησιμοποιώντας τον αλγόριθμο επικύρωσης που διευκρινίζεται από το SA, όπως DES, MD5 ή SHA.

Όταν ο host προορισμού λαμβάνει ένα διάγραμμα δεδομένων IP με ένα AH header, καθορίζει το SA για το πακέτο, και τότε επικυρώνει την ακεραιότητα του διαγράμματος δεδομένων με την επεξεργασία του τομέα στοιχείων επικύρωσης. Το σχέδιο επικύρωσης IPsec (και για το AH και για το ESP πρωτόκολλο) χρησιμοποιεί ένα σχέδιο αποκαλούμενο ως HMAC, το οποίο είναι μια κρυπτογραφημένη σύνοψη μηνυμάτων. Το HMAC χρησιμοποιεί ένα κοινό μυστικό κλειδί μεταξύ δύο συμβαλλόμενων μερών απο τις δημόσιες βασικές μεθόδους για την επικύρωση μηνυμάτων.

The ESP Protocol

Το ESP πρωτόκολλο παρέχει τη μυστικότητα σε επίπεδο δικτύου καθώς επίσης, και την επικύρωση των hosts πηγής. Για άλλη μια φορά, όλα αρχίζουν με έναν host πηγής που καθιερώνει ένα SA με έναν host προορισμού. Κατόπιν ο host πηγής μπορεί να στείλει τα εξασφαλισμένα διαγράμματα δεδομένων στον host προορισμού. Όπως φαίνεται στο παρακάτω σχήμα, ένα εξασφαλισμένο διάγραμμα δεδομένων δημιουργείται με το να περιβάλει τα αρχικά στοιχεία διαγραμμάτων δεδομένων IP με τους τομείς header και trailer, και έπειτα να εισάγει αυτά τα δεδομένα στον τομέα στοιχείων ενός διαγράμματος δεδομένων IP. Για τον τομέα πρωτοκόλλου στο header του διαγράμματος δεδομένων IP, η τιμή 50 είναι χρησιμοποιημένη για να δείξει ότι το διάγραμμα δεδομένων περιλαμβάνει ένα ESP header και ένα trailer. Όταν ο host προορισμού λαμβάνει το διάγραμμα δεδομένων IP, σημειώνει το 50 στον τομέα πρωτοκόλλου, και επεξεργάζεται το διάγραμμα δεδομένων χρησιμοποιώντας το ESP πρωτόκολλο. Όπως φαίνεται στο παρακάτω σχήμα, τα αρχικά στοιχεία διαγραμμάτων δεδομένων IP μαζί με το ESP τομέα κρυπτογραφούνται. Στη μυστικότητα παρέχεται η κρυπτογράφηση des-CBC. Το ESP header αποτελείται από έναν τριανταδυάμπιτο (32-bit) τομέα για το SPI και απο έναν τριανταδυάμπιτο τομέα για τον αριθμό ακολουθίας, ο οποίος έχει ακριβώς τον ίδιο ρόλο όπως στο πρωτόκολλο AH. Το trailer περιλαμβάνει τον τομέα Next Header, το οποίο έχει επίσης ακριβώς τον ίδιο ρόλο. Σημειώνουμε, ότι επειδή ο τομέας Next Header κρυπτογραφείται μαζί με τα αρχικά στοιχεία, ένας εισβολέας δεν θα είναι σε θέση να καθορίσει το πρωτόκολλο μεταφορών που χρησιμοποιείται. Μετά από το trailer υπάρχει το πεδίο Authentication Data, το οποίο εξυπηρετεί πάλι τον ίδιο ρόλο όπως στο πρωτόκολλο AH.



Οι ESP τομείς στο διάγραμμα δεδομένων IP.

SA and Key Management

Για την επιτυχημένη ανάπτυξη του IPsec, είναι απαραίτητη μία εξελικτική και αυτοματοποιημένη SA διαχείρηση. Διάφορα πρωτόκολλα που έχουν καθοριστεί για αυτούς τους στόχους, περιλαμβάνουν:

- Ο αλγόριθμος Internet Key Exchange (IKE) είναι το βασικό διοικητικό πρωτόκολλο προεπιλογής για το IPsec.
- Το Internet Security Assoication and Key Management Protocol (ISKMP) το οποίο καθορίζει τις διαδικασίες για την καθιέρωση. Η σχέση ασφάλειας ISKMP είναι απολύτως χωριστή από το Internet Key Exchange IKE.

Το IPsec καθορίζει επίσης έναν "transport mode" σε ποιους δρομολογητές εισάγει τη λειτουργία ασφάλειας στους hosts. Τέλος, το IPsec περιγράφει τις διαδικασίες κρυπτογράφησης για IPv6, καθώς επίσης, και για το IPv4.

IPsec Configuration

ΟΝΟΜΑ

ipsec.conf: παραμετροποίηση IPsec και συνδέσεις (IPsec configuration and connections)

ΠΕΡΙΓΡΑΦΗ

Το αρχείο ipsec.conf διευκρινίζει τα περισσότερα configuration και ελέγχους πληροφορίών για το υποσύστημα strongSwan IPsec. Το περιεχόμενό του είναι μη ασφαλές.

Το αρχείο αυτό είναι ένα αρχείο κειμένων, που αποτελείται από ένα ή περισσότερα τμήματα. Το κενό που ακολουθείται από “#” και στη συνέχεια είναι ακολουθούμενο από τίποτα στο τέλος της γραμμής, είναι σχόλια και αγνοείται.

Μια γραμμή η οποία περιλαμβάνει και ένα όνομα αρχείου, χωρισμένη από το κενό, αντικαθίσταται από το περιεχόμενο εκείνου του αρχείου που προηγείται και

ακολουθείται από τις κενές γραμμές. Μόνο ένα ενιαίο όνομα αρχείου μπορεί να παρασχεθεί, και μπορεί να μην περιέχει το κενό, παραδείγματος χάριν:

```
include ipsec.*.conf
```

Ο σκοπός αυτός, περιλαμβάνει τη δυνατότητα να επιτραπεί η κράτηση των πληροφοριών για τις συνδέσεις, ή τα σύνολα συνδέσεων, και να χωρίζεται από κύριο configuration αρχείο. Αυτό επιτρέπει σε τέτοιες περιγραφές σύνδεσης να αλλάζεται, να αντιγράφεται στις άλλες πύλες ασφάλειας σχετικές, κ.λπ., χωρίς να πρέπει συνεχώς να τους εξαγάγει από το configuration αρχείο και κατόπιν να τους εισάγει πίσω σε αυτό.

Ένα τμήμα αρχίζει με μια γραμμή της μορφής:

```
type name
```

όπου ο τύπος προσδιορίζει ποιος τύπος τμήματος ακολουθεί, και το όνομα είναι ένα οποιοδήποτε όνομα που διακρίνει το τμήμα από άλλα του ίδιου τύπου. (Τα ονόματα πρέπει να αρχίσουν με ένα γράμμα και μπορούν να περιέχουν μόνο γράμματα, ψηφία, περιόδους, υπογραμμίσεις, και παύλες.) Όλες οι επόμενες μή κενές γραμμές που αρχίζουν με το κενό, είναι μέρος του τμήματος. Σχόλια μέσα σε ένα τμήμα πρέπει να αρχίζουν επίσης με ένα κενό. Μπορεί να υπάρξει μόνο ένα τμήμα ενός δεδομένου τύπου με ένα δεδομένο όνομα.

Οι γραμμές μέσα στο τμήμα είναι γενικά της μορφής,

```
parameter = value
```

Μπορεί να υπάρξει κενό από κάθε πλευρά =. Τα ονόματα παραμέτρου ακολουθούν την ίδια σύνταξη όπως τα ονόματα τμήματος, και είναι συγκεκριμένα για έναν τύπο τμημάτων. Εκτός αν ειδάλλως είναι ρητά διευκρινισμένο, οτι κανένα όνομα παραμέτρου δεν μπορεί να εμφανιστεί περισσότερο από μία φορά στο τμήμα.

Ένα τμήμα με το όνομα %default διευκρινίζει τις προεπιλογές για τα τμήματα του ίδιου τύπου. Για κάθε παράμετρο σε αυτό, οποιοδήποτε τμήμα εκείνου του τύπου που δεν έχει μια παράμετρο του ίδιου ονόματος παίρνει ένα αντίγραφο αυτό από το τμήμα %default. Μπορούν να υπάρξουν πολλαπλάσια τμήματα %default λαμβάνοντας υπόψη τον τύπο και όλα τα τμήματα %default ενός δεδομένου τύπου πρέπει να προηγηθούν.

Αυτήν την περίοδο υπάρχουν τρεις τύποι τμημάτων: ένα τμήμα config το οποίο διευκρινίζει τις γενικές πληροφορίες διαμόρφωσης για το IPsec, ένα τμήμα conn το διευκρινίζει μια σύνδεση IPsec, ενώ ένα τμήμα ca διευκρινίζει τις πρόσθετες ιδιότητες σε μια αρχή πιστοποίησης.

Ένα τμήμα conn περιέχει μια προδιαγραφή σύνδεσης, καθορίζοντας ένα δίκτυο σύνδεσης που γίνεται χρησιμοποιώντας το IPsec. Το όνομα που δίνεται είναι γενικό, και χρησιμοποιείται για να προσδιορίσει τη σύνδεση. Εδώ είναι ένα απλό παράδειγμα:

```
conn snt
    left=192.168.0.1
    leftsubnet=10.1.0.0/16
    right=192.168.0.2
    rightsubnet=10.1.0.0/16
    keyingtries=%forever
    auto=add
```

Μια σημείωση για την ορολογία: Υπάρχουν δύο ειδών επικοινωνιών: τη μετάδοση των πακέτων χρηστών IP, και τις διαπραγματεύσεις `gateway-to-gateway` για τη διαμόρφωση, τη νέα εισαγωγή, και το γενικό έλεγχο. Το μονοπάτι για να ελέγξει τη σύνδεση καλείται “ISAKMP SA” σε IKEv1 και το μονοπάτι επιπέδου

δεδομένων, είναι αποκαλείται “IPsec SA”. Το strongSwan αυτήν την περίοδο χρησιμοποιεί δύο χωριστές διαμόρφωσείς ***daemons***. Το Pluto το οποίο χειρίζεται όλες τις συνδέσεις IKEv1, και το Charon είναι το νέο daemon το οποίο υποστηρίζει το IKEv2 πρωτόκολλο.

ΠΑΡΑΜΕΤΡΟΙ ΣΥΝΔΕΣΗΣ (CONN PARAMETERS)

ah AH authentication algorithm to be used for the connection, e.g. hmac-md5.

auth whether authentication should be done as part of ESP encryption, or separately using the AH protocol; acceptable alues are esp (the default) and ah. The IKEv2 daemon currently supports only ESP.

authby how the two security gateways should authenticate each other; acceptable values are secret or psk for share secrets, rsasig for RSA digital signatures (the default), secret|rsasig for either, and never if negotiation is never to be attempted or accepted (useful for shunt-only conns). Digital signatures are superior in every way to shared secrets. In IKEv2, the two ends must not agree on this parameter, it is relevant for the outbound authentication method only. IKEv1 additionally supports the values xauthpsk and xauthrsasig that will enable eXtended AUTHentication (XAUTH) in addition to IKEv1 main mode based on shared secrets or digital RSA signatures, respectively. IKEv2 additionally supports the value eap, which indicates an initiator to request EAP authentication. The EAP method to use is selected by the server.

auto what operation, if any, should be done automatically at IPsec startup; currently-accepted values are add , route, start and ignore. Add loads a connection without starting it. Route loads a connection and installs kernel traps. If traffic is detected between leftsubnet and rightsubnet , a connection is established. Start loads a connection and brings it up immediatly. Ignore ignores the connection. This is equal to delete a connection from the config file. Relevant only locally, other end need not agree on it (but in general, for an intended-to-be- permanent connection, both ends should use auto=start to ensure that any reboot causes immediate renegotiation).

compress whether IPComp compression of content is proposed on the connection (link-level compression does not work on encrypted data, so to be effective, compression must be done before encryption); acceptable values are yes and no (the default). A compress whether IPComp compression of content is proposed on the connection (link-level compression does not work on encrypted data, so to be effective, compression must be done before encryption); acceptable values are yes and no (the default). A value of yes causes IPSec to propose both compressed and uncompressed, and prefer compressed. A value of no prevents IPSec from proposing compression; a proposal to compress will still be accepted. IKEv2 does not support IP compression yet. Value of yes causes IPSec to propose both compressed and uncompressed, and prefer compressed. A value of no prevents IPSec from proposing compression; a proposal to compress will still be accepted. IKEv2 does not support IP compression yet.

dpdaction controls the use of the Dead Peer Detection protocol (DPD, RFC 3706) where R_U_THERE notification messages (IKEv1) or empty INFORMATIONAL messages (IKEv2) are periodically sent in order to check the liveliness of the IPsec peer. The values clear, hold, and restart all activate DPD. If no activity is detected, all connections with a dead peer are stopped and unrouted (clear), put in the hold state (hold) or restarted (restart). For KEv1, the default is none which disables the active sending of R_U_THERE notifications. Nevertheless pluto will always send the DPD Vendor ID during connection set up in order to signal the readiness to act passively as a responder if the peer wants to use DPD. For IKEv2, none doesn't make sense, since all messages are used to detect dead peers. If specified, it has the same meaning as the default (clear).

dpddelay defines the period time interval with which R_U_THERE messages/INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received. In IKEv2, a value of 0 sends no additional INFORMATIONAL messages and uses only standard messages (such as those to rekey) to detect dead peers.

dpdtimeout defines the timeout interval, after which all connections to a peer are deleted in case of inactivity. This only applies to IKEv1, in IKEv2 the default retransmission timeout applies, as every exchange is used to detect dead peers.

eap defines the EAP type to propose as server if the client has authby=eap selected. Acceptable values are aka for EAP-AKA, sim for EAP-SIM and md5 for EAP-MD5. Additionally, IANA assigned EAP method numbers are accepted, or a definition in the form eap=type-vendor (e.g. eap=7-12345) can be used to specify vendor specific EAP types.

eap_identity defines the identity the client uses to reply to a EAP Identity request. If defined on the EAP server, the defined identity will be used as peer identity during EAP authentication. The special value %identity uses the EAP Identity method to ask the client for a EAP identity. If not defined, the IKEv2 identity will be used as EAP identity.

esp ESP encryption/authentication algorithm to be used for the connection, e.g. 3des-md5 (encryption-integrity-[dh-group]). If dh-group is specified, CHILD_SA setup and rekeying include a separate diffe hellman exchange (IKEv2 only).

forceencaps Force UDP encapsulation for ESP packets even if no NAT situation is detected. This may help to hurdle restrictive firewalls. To enforce the peer to encapsulate packets, NAT detection payloads are faked (IKEv2 only).

ike IKE/ISAKMP SA encryption/authentication algorithm to be used, e.g. aes128-sha1-modp2048 (encryption-integrity-dhgroup). In IKEv2, multiple algorithms and proposals may be included, such as aes128-aes256-sha1-modp1536-modp2048,3des-sha1-md5-modp1024.

ikelifetime how long the keying channel of a connection ('ISAKMP/IKE SA') should last before being renegotiated.

installpolicy decides whether IPsec policies are installed in the kernel by the IKEv2 charon daemon for a given connection. Allows peaceful co-existence e.g. with the Mobile IPv6 daemon mip6d who wants to control the kernel policies. Acceptable values are yes (the default) and no.

keyexchange method of key exchange; which protocol should be used to initialize the connection. Connections marked with ikev1 are initiated with pluto, those marked with ikev2 with charon. An incoming request from the remote peer is handled by the correct daemon, unaffected from the keyex change setting. The default value ike currently behaves exactly as ikev1.

keyingtries how many attempts (a whole number or %forever) should be made to negotiate a connection, or a replacement for one, before iving up (default %forever). The value %forever means 'never give up'. Relevant only locally, other end need not agree on it.

keylife how long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry; acceptable values are an integer optionally followed by s (a time in seconds) or a decimal number followed by m, h, or d (a time in minutes, hours, or days respectively) (default 1h, maximum 24h). Normally, the connection is renegotiated (via the keying channel) before it expires. The two ends need not exactly agree on keylife, although if they do not, there will be some clutter of superseded connections on the end which thinks the lifetime is longer.

left (required) the IP address of the left participant's public-network interface, in any form accepted by ttoaddr(3) or one of several magic values. If it is %defaultroute, left will be filled in automatically with the local address of the default-route interface (as determined at IPsec startup time). (Either left or right may be %defaultroute, but not both.) The value %any signifies an address to be filled in (by automatic keying) during negotiation. The prefix % in front of a fully-qualified domain name or an IP address will implicitly set leftallowany=yes. If the domain name cannot be resolved into an IP address at IPsec startup or update time then left=%any and leftallowany=no will be assumed.

leftallowany a modifier for left, making it behave as %any although a concrete IP address has been assigned. Recommended or dynamic IP addresses that can be resolved by DynDNS at IPsec startup or update time. Acceptable values are yes and no (the default).

leftca the distinguished name of a certificate authority which is required to lie in the trust path going from the left participant's certificate up to the root certification authority.

leftcert the path to the left participant's X.509 certificate. The file can be coded either in PEM or DER format. OpenPGP certificates are supported as well. Both absolute paths and paths relative to /etc/ipsec.d/certs are accepted. By default leftcert sets leftid to the distinguished name of the certificate's subject and leftca to the distinguished name of the certificate's issuer. The left participant's ID can be

overriden by specifying a leftid value which must be certified by the certificate, though.

leftfirewall whether the left participant is doing forwarding-fire walling (including masquerading) using iptables for traffic from leftsubnet, which should be turned off (for traffic to the other subnet) once the connection is established; acceptable values are yes and no (the default). May not be used in the same connection description with leftupdown. Implemented as a parameter to the default ipsec _updown script. See notes below. Relevant only locally, other end need not agree on it.

leftgroups a comma separated list of group names. If the leftgroups parameter is present then the peer must be a member of at least one of the groups defined by the parameter. Group membership must be certified by a valid attribute certificate stored in /etc/ipsec.d/acerts/ thas has been issued to the peer by a trusted Authorization Authority stored in /etc/ipsec.d/aacerts/. Attribute certificates are not supported in IKEv2 yet.

lefthostaccess inserts a pair of INPUT and OUTPUT iptables rules using the default ipsec _updown script, thus allowing access to host itself in the case where the host's internal interface is part of the negotiated client subnet. Acceptable values are yes and no (the default).

leftid how the left participant should be identified for authentication; defaults to left. Can be an IP address or a fully qualified domain name proceeded by @ (which is used as a literal string and not resolved).

leftnexthop this parameter is not needed any more because the NETKEY IPsec stack does not require explicit routing entries for the traffic to be tunneled.

leftprotoport restrict the traffic selector to a single protocol and/or port. Examples: leftprotoport=tcp/http or leftproto port=6/80 or leftprotoport=udp

leftrsasigkey the left participant's public key for RSA signature authentication, in RFC 2537 format using encoding. The magic value %none means the same as not specifying a value (useful to override a default). The value %cert (the default) means that the key is extracted from a certificate. The identity used for the left participant must be a specific host, not %any or another magic value. Caution: if two connection descriptions specify different public keys for the same leftid, confusion and madness will ensue.

leftsendcert Accepted values are never or no, always or yes, and ifasked.

leftsourceip The internal source IP to use in a tunnel, also known as virtual IP. If the value is %modeconfig, %modecfg, %con fig, or %cfg, an address is requested from the peer. In IKEv2, a defined address is requested, but the server may change it. If the server does not support it, the address is enforced.

rightsourceip The internal source IP to use in a tunnel for the remote peer. If the value is %config on the responder side, the initiator must propose a address which is then echoed back. The IKEv2 daemon also supports address pools expressed as

network/netmask or the use of an external IP address pool using %poolname, where poolname is the name of the IP address pool used for the lookup.

leftsubnet private subnet behind the left participant, expressed as network/netmask if omitted, essentially assumed to be left/32, signifying that the left end of the connection goes to the left participant only. When using IKEv2, the configured subnet of the peers may differ, the protocol narrows it to the greatest common subnet. Further, IKEv2 supports multiple subnets separated by commas. IKEv1 only interprets the first subnet of such a definition.

leftsubnetwithin the peer can propose any subnet or single IP address that fits within the range defined by leftsubnetwithin. Not relevant for IKEv2, as subnets are narrowed.

leftupdown what ''updown'' script to run to adjust routing and/or firewalling when the status of the connection changes (default ipsec _updown). May include positional parameters separated by white space (although this requires enclosing the whole string in quotes); including shell metacharacters is unwise. Relevant only locally, other end need not agree on it. IKEv2 uses the updown script to insert firewall rules only. Routing is not support and will be implemented directly into Charon.

mobike enables the IKEv2 MOBIKE protocol defined by RFC 4555. Accepted values are yes (the default) and no. If set to no, the IKEv2 charon daemon will not actively propose MOBIKE as initiator and ignore the MOBIKE_SUPPORTED notify as responder.

modeconfig defines which mode is used to assign a virtual IP. Accepted values are push and pull (the default). Currently relevant for IKEv1 only since IKEv2 always uses the configuration payload in pull mode.

pfs whether Perfect Forward Secrecy of keys is desired on the connection's keying channel (with PFS, penetration of the key-exchange protocol does not compromise keys negotiated earlier); acceptable values are yes (the default) and no. IKEv2 always uses PFS for IKE_SA rekeying whereas for CHILD_SA rekeying PFS is enforced by defining a DiffieHellman modp group in the esp parameter.

pfsgroup defines a Diffie-Hellman group for perfect forward secrecy in IKEv1 Quick Mode differing from the DH group used for IKEv1 Main Mode (IKEv1 only).

reauth whether rekeying of an IKE_SA should also reauthenticate the peer. In IKEv1, reauthentication is always done. In IKEv2, a value of no rekeys without uninstalling the IPsec SAs, a value of yes (the default) creates a new IKE_SA from scratch and tries to recreate all IPsec SAs.

rekey whether a connection should be renegotiated when it is about to expire; acceptable values are yes (the default) and no. The two ends need not agree, but while a value of no prevents Pluto/Charon from requesting renegotiation, it does not prevent responding to renegotiation requested from the other end, so no will be largely ineffective unless both ends agree on it.

rekeyfuzz maximum percentage by which rekeymargin should be randomly increased to randomize rekeying intervals (important for hosts with many connections); acceptable values are an integer, which may exceed 100, followed by a '%'. The value of rekeymargin, after this random increase, must not exceed keylife. The value 0% will suppress time randomization. Relevant only locally, other end need not agree on it.

rekeymargin how long before connection expiry or keying-channel expiry should attempts to negotiate a replacement begin; acceptable values as for keylife (default 9m). Relevant only locally, other end need not agree on it.

type the type of the connection; currently the accepted values are tunnel (the default) signifying a host-to-host, host-to-subnet, or subnet-to-subnet tunnel; transport, signifying host-to-host transport mode; transport_proxy, signifying the special Mobile IPv6 transport proxy mode; passthrough, signifying that no IPsec processing should be done at all; drop, signifying that packets should be discarded; and reject, signifying that packets should be discarded and a diagnostic ICMP returned. Charon currently supports tunnel, transport, and tunnel_proxy connection types, only.

xauth specifies the role in the XAUTH protocol if activated by authby=xauthpsk or authby=xauthrsasig. Accepted values are server and client (the default).

CONN PARAMETERS: IKEv2 MEDIATION EXTENSION

The following parameters are relevant to IKEv2 Mediation Extension operation only.

mediation whether this connection is a mediation connection, ie. whether this connection is used to mediate other connections. Mediation connections create no child SA. Acceptable values are no (the default) and yes.

mediated_by the name of the connection to mediate this connection through. If given, the connection will be mediated through the named mediation connection. The mediation connection must set mediation=yes.

me_peerid ID as which the peer is known to the mediation server, ie. which the other end of this connection uses as its leftid on its connection to the mediation server. This is the ID we request the mediation server to mediate us with. If me_peerid is not given, the rightid of this connection will be used as peer ID.

Αυτά είναι προαιρετικά τμήματα που μπορούν να χρησιμοποιηθούν για να ορίσουν τις πρόσθετους παραμέτρους σε μια αρχή πιστοποίησης (Certification Authority (CA)). Αυτές οι παράμετροι δεν είναι υποστηρίζονται στο IKEv2 ακόμα.

auto currently can have either the value ignore or add

cacert defines a path to the CA certificate either relative to /etc/ipsec.d/cacerts or as an absolute path.

crluri defines a CRL distribution point (ldap, http, or file URI)

crluri1 synonym for crluri.

crluri2 defines an alternative CRL distribution point (ldap, http, or file URI)

ldaphost defines an ldap host. Currently used by IKEv1 only.

ocspuri defines an OCSP URI.

ocspuri1 synonym for ocspuri.

ocspuri2 defines an alternative OCSP URI. Currently used by IKEv2 only. Certuribase defines the base URI for the Hash and URL feature supported by IKEv2. Instead of exchanging complete certificates, IKEv2 allows to send an URI that resolves to the DER encoded certificate. The certificate URIs are built by appending the SHA1 hash of the DER encoded certificates to this base URI.

Αυτή τη στιγμή, το μόνο config τμήμα που είναι γνωστό στο λογισμικό IPsec είναι το setup, που περιέχει τις πληροφορίες όταν ξεκινάει το λογισμικό. Εδώ είναι ένα παράδειγμα:

```
config setup
    plutodebug=all
    crlcheckinterval=10m
    strictcrlpolicy=yes
```

cachecrls certificate revocation lists (CRLs) fetched via http or ldap will be cached in /etc/ipsec.d/crls/ under a unique file name derived from the certification authority's public key. Accepted values are yes and no (the default).

charonstart whether to start the IKEv2 Charon daemon or not. Accepted values are yes (the default) or no.

dumpdir in what directory should things started by ipsec starter (notably the Pluto and Charon daemons) be allowed to dumpcore? The empty value (the default) means they are not allowed to. This feature is currently not yet supported by ipsec starter.

plutostart whether to start the IKEv1 Pluto daemon or not. Accepted values are yes (the default) or no.

strictcrlpolicy defines if a fresh CRL must be available in order for the peer authentication based on RSA signatures to succeed. Accepted values are yes and no (the default). IKEv2 additionally recognizes ifuri which reverts to yes if at least one CRL URI is defined and to know if no URI is known.

uniqueids whether a particular participant ID should be kept unique, with any new (automatically keyed) connection using an ID from a different IP address deemed to replace all old ones using that ID; acceptable values are yes (the default) and no. Participant IDs normally are unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one. The IKEv2 daemon also accepts the value replace which is identical to yes and the value keep rejecting new IKE_SA setups and keep the duplicate established earlier.

Οι ακόλουθες παράμετροι τμημάτων config χρησιμοποιούνται από το IKEv1 Pluto daemon μόνο:

crlcheckinterval interval in seconds. CRL fetching is enabled if the value is greater than zero. Asynchronous, periodic checking for fresh CRLs is currently done by the IKEv1 Pluto daemon only.

keep_alive interval in seconds between NAT keep alive packets, the default being 20 seconds.

nat_traversal activates NAT traversal by accepting source ISAKMP ports different from udp/500 and being able of floating to udp/4500 if a NAT situation is detected. Accepted values are yes and no (the default).

nocrsend no certificate request payloads will be sent. Accepted values are yes and no (the default). Used by IKEv1 only, NAT traversal always being active in IKEv2.

pkcs11initargs non-standard argument string for PKCS#11 C_Initialize() function; required by NSS softoken.

pkcs11module defines the path to a dynamically loadable PKCS #11 library.

pkcs11keepstate PKCS #11 login sessions will be kept during the whole lifetime of the keying daemon. Useful with pin-pad smart card readers. Accepted values are yes and no (the default).

pkcs11proxy Pluto will act as a PKCS #11 proxy accessible via the whack interface. Accepted values are yes and no (the default).

Plutodebug how much Pluto debugging output should be logged. An empty value, or the magic value none, means no debugging output (the default). The magic value all means full output. Otherwise only the specified types of output (a quoted list, names without the --debug- prefix, separated by white space) are enabled.

plutostderrlog Pluto will not use syslog, but rather log to stderr, and redirect stderr to the argument file.

postpluto shell command to run after starting Pluto (e.g., to remove a decrypted copy of the ipsec.secrets file). It's run in a very simple way; complexities like I/O redirection are best hidden within a script. Any output is redirected for logging, so running interactive commands is difficult unless they use /dev/tty or equivalent for their interaction. Default is none.

prepluto shell command to run before starting Pluto (e.g., to decrypt an encrypted copy of the ipsec.secrets file). It's run in a very simple way; complexities like I/O redirection are best hidden within a script. Any output is redirected for logging, so running interactive commands is difficult unless they use /dev/tty or equivalent for their interaction. Default is none.

virtual_private defines private networks using a wildcard notation.

Οι ακόλουθες παράμετροι τμημάτων config χρησιμοποιούνται μόνο από το IKEv2 Charon daemo:

Charondebug how much Charon debugging output should be logged. A comma separated list containing type level/pairs may be specified, e.g: dmn 3, ike 1, net -1. Acceptable values for types are dmn, mgr, ike, chd, job, cfg, knl, net, enc, lib and the level is one of -1, 0, 1, 2, 3, 4 (for silent, audit, control controlmore, raw, private).

Οι ακόλουθες παράμετροι τμημάτων config έχουν μόνο νόημα εάν το stack KLIPS IPsec χρησιμοποιείται, αντί του stack προεπιλογής NETKEY του Linux πυρήνα 2.6:

fragicmp whether a tunnel's need to fragment a packet should be reported back with an ICMP message, in an attempt to make the sender lower his PMTU estimate; acceptable values are yes (the default) and no.

hidetos whether a tunnel packet's TOS field should be set to 0 rather than copied from the user packet inside; acceptable values are yes (the default) and no.

interfaces virtual and physical interfaces for IPsec to use: a single virtual=physical pair, a (quoted!) list of pairs separated by white space, or %none. One of the pairs may be written as %default route, which means: find the interface d that the default route points to, and then act as if the value was ''ipsec0=d''.%defaultroute is the default; %none must be used to denote no interfaces.

overridemtu value that the MTU of the ipsecn interface(s) should be set to, overriding IPsec's (large) default.

Ipsec commands

Το IPsec είναι μια εντολή που περιλαμβάνει μια συλλογή των μεμονωμένων υπο- εντολών της μορφής,

ipsec <*command*> **[** <*argument*> **] [** <*options*> **]**

όπου μπορεί να χρησιμοποιηθεί για να ελέγξει και να επιτηρήσει τις συνδέσεις IPsec, καθώς επίσης και το IKE daemons.

Control Commands

ipsec start [<*starter options*> **]**

Καλεί το ipsec starter [<*starter options*>]], το οποίο αναλύει στη συνέχεια το ipsec.conf και αρχίζει το IKEv1 pluto και IKEv2 charon daemons.

ipsec stop

Ολοκληρώνει όλη τη σύνδεση IPsec και σταματά το IKEv1 pluto και IKEv2 charon daemons με την αποστολή ενός σήματος *TERM* στον ipsec starter.

ipsec restart [<*starter options*> **]**

Είναι ισοδύναμος με το ipsec stop που ακολουθείται από το ipsec start[<starter options>] μετά από μια περίοδο 2 δευτερολέπτων.

ipsec update

Στέλνει ένα σήμα HUP στον ipsec starter που καθορίζει στη συνέχεια οποιεσδήποτε αλλαγές στο ipsec.conf και ενημερώνει τη διαμόρφωση "τρέχοντας" το IKEv1 pluto και IKEv2 charon daemons, αντίστοιχα.

ipsec reload

Στέλνει ένα USR1 σήμα στον ipsec starter που ξαναφορτώνει στη συνέχεια ολόκληρη τη διαμόρφωση (configuration) τρέχοντας το IKEv1 pluto και IKEv2 charon daemons το οποίο είναι βασισμένο στο πραγματικό ipsec.conf.

ipsec up <*name*>

Λέει στο αρμόδιο IKE daemon να ξεκινήσει τη σύνδεση με το όνομα <name>. Εκτελείται με την κλήση του ipsec whack -- όνομα <name> --initiate και/ή ipsec stroke up <*name*>.

ipsec down <*name*>

Λέει στο αρμόδιο IKE daemon να ολοκληρώσει τη σύνδεση <name>. Εκτελείται με την κλήση του ipsec whack --όνομα <name> -- terminate και/ή ipsec stroke down <*name*>.

ipsec route <*name*>

Λέει στο αρμόδιο IKE daemon να παρεμβάλει μια πολιτική IPsec στον πυρήνα για τη σύνδεση <name>. Το πρώτο πακέτο ωφέλιμων φορτίων που ταιριάζει με την πολιτική IPsec θα προκαλέσει αυτόματα μια οργάνωση σύνδεσης IKE. Εκτελείται με την κλήση του ipsec whack --όνομα <name> -- route και/ή ipsec stroke route <*name*>.

ipsec unroute <*name*>

Αφαιρεί την πολιτική IPsec στον πυρήνα για τη σύνδεση <name>. Εκτελείται με την κλήση του ipsec whack --όνομα <name> -- unroute ή/και ipsec stroke unroute <name>.

ipsec status [<*name*> **]**

Επιστρέφει τις συνοπτικές πληροφορίες θέσης για τη σύνδεση <name>, σε όλες τις συνδέσεις. Εκτελείται με την κλήση του ipsec whack [--όνομα <name>] --status and/or ipsec stroke status [<*name*>].

ipsec statusall [<*name*> **]**

Επιστρέφει τις λεπτομερείς πληροφορίες θέσης για τη σύνδεση <name>, σε όλες τις συνδέσεις. Εκτελείται με την κλήση του ipsec whack [-- όνομα <name>] statusall ή/και ipsec stroke statusall [<*name*>].

Info Commands

ipsec version

Επιστρέφει την έκδοση ipsec υπό μορφή **Linux strongSwan U<strongSwan userland version>/K<Linux kernel version>,** εάν το strongSwan χρησιμοποιεί τον NETKEY IPsec stack του πυρήνα Linux που τρέχει.

ipsec copyright

Επιστρέφει τις πληροφορίες δικαωμάτων.

ipsec –confdir

Επιστρέφει τον κατάλογο SYSCONFDIR σαν ορισμένο (default) απο επιλογές του configuration.

ipsec --directory

Επιστρέφει τον κατάλογο LIBEXECDIR σαν ορισμένο απο επιλογές του configuration.

ipsec –help

Επιστρέφει τις πληροφορίες χρήσης για την εντολή ipsec.

ipsec –versioncode

Επιστρέφει τον αριθμό έκδοσης ipsec υπό μορφή **U<strongSwan userland version>/K<Linux kernel version>,** εάν το strongSwan χρησιμοποιεί τον IPsec του πυρήνα Linux που τρέχει.

List Commands

ipsec listaacerts [--utc]

Επιστρέφει μία λίστα πιστοποιητικών απο X.509 Authorization Authority (AA) που φορτώθηκαν τοπικά από το IKE daemon από τον κατάλογο του /etc/ipsec.d/aacerts. Εκτελείται με την κλήση του ipsec whack -- listaacerts και/ή ipsec stroke listaacerts.

ipsec listacerts [--utc]

Επιστρέφει μία λίστα απο X.509 Attribute certificates που φορτώθηκαν τοπικά από το IKE daemon από τον κατάλογο του /etc/ipsec.d/acerts. Εκτελείται με την κλήση [[IpsecWhacklipsec whack] -- listacerts και/ή [wikiIpsecStroke ipsec stroke]] listacerts.

ipsec listalgs

Επιστρέφει μία λίστα όλων των υποστηριζόμενων αλγορίθμων κρυπτογράφησης IKE και hash, τις διαθέσιμες ομάδες diffie-Hellman, καθώς επίσης, και όλους τους ESP αλγορίθμους κρυπτογράφησης και επικύρωσης που εγγράφονται μέσω Crypto API του πυρήνα Linux. Υποστηρίζεται από το IKEv1 pluto daemon μόνο. Εκτελείται με την κλήση του ipsec whack --listalgs.

ipsec listcacerts [--utc]

Επιστρέφει μία λίστα απο πιστοποιητικά X.509 Certification Authority (CA) που φορτώθηκαν τοπικά από το IKE daemon από τον κατάλογο του /etc/ipsec.d/cacerts ή παραλήφθηκαν στα PKCS#7 πιστοποιητικά μέσω του πρωτοκόλλου IKE. Εκτελείται με την κλήση του ipsec whack -- listcacerts και/ή ipsec stroke listcacerts.

ipsec listcainfos [--utc]

Επιστρέφει τις πληροφορίες αρχής πιστοποίησης (Certification Authority) (σημεία διανομής CRL, κεντρικοί υπολογιστές OCSP URIs, LDAP), που καθορίστηκαν από τα τμήματα CA στο ipsec.conf. Εκτελείται με την κλήση του ipsec whack -- listcainfos και/ή ipsec stroke listcainfos.

ipsec listcards [--utc]

Απαριθμεί όλα τα πιστοποιητικά που βρίσκονται στις έξυπνες κάρτες. Υποστηρίζεται από το IKEv1 pluto daemon μόνο. Εκτελείται με την κλήση του ipsec whack -- listcards.

ipsec listcrls [--utc]

Επιστρέφει μία λίστα απο Certificate Revocation Lists (CRLs) που φορτώθηκαν από το IKE daemon από τον κατάλογο etcipsecdcrls. Εκτελείται με την κλήση του ipsec whack -- listcrls ή/και ipsec stroke listcrls.

ipsec listcerts [--utc]

Επιστρέφει μία λίστα των X.509 ή/και πιστοποιητικών OpenPGP που είτε φορτώθηκαν τοπικά από το IKE daemon είτε παραλήφθηκαν μέσω του πρωτοκόλλου IKEv2. Εκτελείται με την κλήση του ipsec whack --listcerts ή/και ipsec stroke listcerts.

ipsec listgroups [--utc]

Επιστρέφει μία λίστα όλων των ομάδων που χρησιμοποιούνται για να καθοριστούν τα σχεδιαγράμματα έγκρισης χρηστών. Υποστηρίζεται από το IKEv1 pluto daemon μόνο. Εκτελείται με την κλήση του ipsec whack --listgroups.

ipsec listocsp [--utc]

Επιστρέφει πληροφορίες ανάκλησης που προσκομίζονται από τους κεντρικούς υπολογιστές OCSP. Εκτελείται με την κλήση [[IpsecWhacklipsec whack] --listocps ή/και ipsec stroke listocsp.

ipsec listocspcerts [--utc]

Επιστρέφει μία λίστα πιστοποιητικών απο X.509 OCSP Signer που είτε φορτώθηκαν τοπικά από το IKE daemon από τον κατάλογο του /etc/ipsec.d/ocspcerts είτε στάλθηκαν από έναν κεντρικό υπολογιστή OCSP. Εκτελείται με την κλήση του ipsec whack --listocspcerts ή/και ipsec stroke listocspcerts.

ipsec listpubkeys [--utc]

Επιστρέφει μία λίστα δημόσιων κλειδιών RSA (RSA Public Keys) που είτε φορτώθηκαν με το βασικό σχήμα είτε δημιουργήθηκαν από τα πιστοποιητικά X.509 και OpenPGP. Υποστηρίζεται από το IKEv1 pluto daemon μόνο. Εκτελείται με την κλήση του ipsec whack -- listpubkeys.

ipsec listall [--utc]

Επιστρέφει όλες τις πληροφορίες που παράγονται από τις εντολές που υπόθηκαν παραπάνω. Κάθε εντολή μπορεί να κληθεί με την επιλογή _--utc_ που επιδεικνύει όλες τις ημερομηνίες σε UTC αντί της τοπικής ώρας. Εκτελείται με την κλήση του ipsec whack -- listall ή/και ipsec stroke listall.

Reread Commands

ipsec rereadaacerts

Διαβάζει όλα τα αρχεία πιστοποιητικών που περιλαμβάνονται στον κατάλογο του /etc/ipsec.d/aacerts και τα προσθέτει στη λίστα πιστοποιητικών Authorization Authority (AA). Εκτελείται με την κλήση του ipsec whack --readaacerts ή/και ipsec stroke rereadaacerts.

ipsec rereadacerts

Διαβάζει όλα τα αρχεία πιστοποιητικών που περιλαμβάνονται στον κατάλογο του /etc/ipsec.d/acerts και τα προσθέτει στη λίστα των πιστοποιητικών ιδιοτήτων. Εκτελείται με την κλήση του ipsec whack --rereadacerts ή/και ipsec stroke rereadacerts.

ipsec rereadcacerts

Διαβάζει όλα τα αρχεία πιστοποιητικών που περιλαμβάνονται στον κατάλογο του /etc/ipsec.d/cacerts και τα προσθέτει στη λίστα Certification Authority (CA). Εκτελείται με την κλήση του ipsec whack --rereadcacerts ή/και ipsec stroke rereadcacerts.

ipsec rereadcrls

Διαβάζει όλους τους καταλόγους ανάκλησης πιστοποιητικών (Certificate Revocation Lists "CRL") που περιλαμβάνονται στον κατάλογο του /etc/ipsec.d/crls και τους προσθέτει στη λίστα των CRLs. Το παλαιότερο CRLs αντικαθίσταται από νεώτερο. Εκτελείται με την κλήση του ipsec whack --rereadcrls ή/και ipsec stroke rereadcrls.

ipsec rereadocspcerts

Διαβάζει όλα τα αρχεία πιστοποιητικών που περιλαμβάνονται στον κατάλογο του /etc/ipsec.d/ocspcerts και τα προσθέτει στη λίστα πιστοποιητικών OCSP signer. Εκτελείται με την κλήση του ipsec whack -- rereadocspcerts ή/και ipsec stroke rereadocspcerts.

ipsec rereadsecrets

Ξαναδιαβάζει όλα τα μυστικά που καθορίζονται στο ipsec.secrets. Εκτελείται με την κλήση του ipsec whack -- rereadsecrets ή/και ipsec stroke rereadsecrets.

ipsec secrets

Είναι ισοδύναμο με το ipsec rereadsecrets.

ipsec rereadall

Εκτελεί όλες τις ξαναδιαβασμένες εντολές που φαίνονται παραπάνω. Εκτελείται με την κλήση του ipsec whack --rereadall ή/και ipsec stroke rereadal.

PKCS11 Proxy Commands

ipsec scencrypt <*value*> **[--inbase** <*base*> **] [--outbase** <*base*> **] [--keyid** <*id*> **]**

Υποστηρίζεται μόνο από το IKEv1 pluto daemon. Εκτελείται με την κλήση του ipsec whack – scencrypt.

ipsec scdecrypt <*value*> **[--inbase <base>] [--outbase** <*base*> **] [--keyid** <*id*> **]**

Υποστηρίζεται μόνο από το IKEv1 pluto daemon. Εκτελείται με την κλήση του ipsec whack --scdecrypt εντολή.

ΤΟ STRONGSWAN

ΣΧΕΤΙΚΑ

Το strongSwan είναι μια εφαρμογή OpenSource IPsec για το λειτουργικό σύστημα Linux. Είναι βασισμένο στο πρόγραμμα FreeS/WAN και το patch X.509 που αναπτύχθηκε κατά τη διάρκεια των τελευταίων τριών ετών. Προκειμένου να υπάρξει μια σταθερή πλατφόρμα IPsec για να βασίσουμε τις μελλοντικές επεκτάσεις μας πάνω στο X.509, αποφασίστηκε η εκτέλεση του strongSwan προγράμματος.

Το strongSwan εστιάστηκε στα εξής:

- Στην απλότητα της διαμόρφωσης (configuration).
- Σε ισχυρές μεθόδους κρυπτογράφησης και επικύρωσης.
- Σε ισχυρές πολιτικές IPsec που υποστηρίζουν τα μεγάλα και σύνθετα δίκτυα VPN (Virtual Private Network).

Ο αρχιπρογραμματιστής του strongSwan είναι ο Andreas Steffen, ο οποίος είναι καθηγητής για την ασφάλεια στις επικοινωνίες, προϊστάμενος του ιδρύματος για τις τεχνολογίες Διαδικτύου, τις εφαρμογές στο πανεπιστήμιο των εφαρμοσμένων επιστημών Rapperswil στην Ελβετία, καθώς και Πρόεδρος της συμβουλευτικής εταιρείας strongSec GmbH.

ΕΙΣΑΓΩΓΗ

Το strongSwan είναι μια OpenSource IPsec λύση για το λειτουργικό σύστημα Linux. Υποστηρίζει τις ακόλουθες σημαντικές λειτουργίες:

- Τρέχει και στον Linux 2.4 (KLIPS) αλλά και στον Linux 2.6 (native IPsec) πυρήνα,
- Ισχυρή 3DES, AES, Serpent, Twofish, ή Blowfish κωδικοποίηση.
- Αυθεντικοποιήση βασισμένη στην X.509 πιστοποίηση (certificate).
- Ισχυρές πολιτικές IPsec βασισμένες σε wildcards ή σε CAs.
- Ανάκτηση απο Certificate Revocation Lists (CRLs) μέσω HTTP ή LDAP.
- Πλήρη υποστήριξη απο το πρωτόκολλο Online Certificate Status Protocol (OCSP, RFC 2560).
- Υποστήριξη απο το πρωτόκολλο Dead Peer Detection Protocol (DPD, RFC 3706).
- Διαχείρηση λειτουργικότητας CA περικλύοντας OCSP, CRL URIs και LDAP server.
- Αποθήκευση ιδιωτικών κλειδιών (private keys) σε έξυπνες κάρτες (smartcards) ή USB crypto tokens μέσω PKCS #11 API.
- NAT-Traversal.
- Υποστήριξη απο Virtual IPs μέσω στατικής παραμετροποίησης (configuration).
- Υποστήριξη απο διαγραμμένο SA και ενημερωτικά μηνύματα ανακοίνωσης.
- Dead Peer Detection (DPD, RFC 3706).

Quickstart

Στα ακόλουθα παραδείγματα υποθέτουμε για λόγους σαφήνειας οτι υποδεικνύεται στα αριστερά (left) ο τοπικός χρήστης (local host) και στα δεξιά είναι ο μακρινός χρήστης (remote host). Τα πιστοποιητικά για τους χρήστες, ή hosts και τις πύλες εκδίδονται από ένα strongSwan CA (Certificate Authority). Πώς να παράγει τα ιδιωτικά κλειδιά και τα πιστοποιητικά που χρησιμοποιώντας το OpenSSL θα αναφερθεί παρακάτω. Το πιστοποιητικό CA, strongswanCert.pem, πρέπει να είναι παρόν σε όλα τα σημεία του VPN προκειμένου να είναι σε θέση να επικυρώσει τα ζεύγοι χρηστών μεταξύ τους.

Στη συνέχεια θα αναφερθούν κάποια παραδείγματα προκειμένου να γίνει κατανοητή (όσο γίνεται) και η παραμετροποίηση για το StrongSwan. Σχετικά με το τί κάνει μια εντολή αναφέρθηκε παραπάνω στο κεφάλαιο IPSec.

Site-to-site case

Σε αυτό το σενάριο δύο ασφαλές πύλες (gateways) ο moon και ο sun θα συνδέσουν δύο μάσκες υποδικτύων moon-net και sun-net το ένα με το άλλο, μέσω ενός VPN tunnel μεταξύ των δύο πυλών:

10.1.0.0/16 -- | 192.168.0.1 | === | 192.168.0.2 | -- 10.2.0.0/16
moon-net moon sun sun-net

Configuration on gateway *moon*:

/*Εδώ, τοποθετούνται τα πιστοποιητικά αλλά και το κλειδί

/etc/ipsec.d/cacerts/strongswanCert.pem //CA Certificates

/etc/ipsec.d/certs/moonCert.pem //Certificates

/etc/ipsec.secrets: //Αρχείο καταγραφής ιδιωτικών RSA κλειδιών

: RSA moonKey.pem //Ιδιωτικό Κλειδί

/etc/ipsec.conf: //Αρχείο Παραμετροποίησης

```
conn net-net      //Σύνδεση conn
    left=%defaultroute
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    right=192.168.0.2
    rightsubnet=10.2.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=sun.strongswan.org"
    auto=start
```

Configuration on gateway *sun*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/sunCert.pem

/etc/ipsec.secrets:

: RSA sunKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn net-net
    left=%defaultroute
    leftsubnet=10.2.0.0/16
    leftcert=sunCert.pem
    right=192.168.0.1
```

```
    rightsubnet=10.1.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Host-to-host case

Αυτό είναι ένα set up μεταξύ δύο ενιαίων hosts που δεν έχουν ένα υποδίκτυο απο πίσω από τους. Αν και ο τρόπος μεταφορών IPsec θα ήταν ικανοποιητικός για τις συνδέσεις host-host θα χρησιμοποιήσουμε τον προεπιλεγμένο τρόπο **IPsec tunnel**.

| 192.168.0.1 | === | 192.168.0.2 |
moon sun

Configuration on host *moon*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/moonCert.pem

/etc/ipsec.secrets:

```
: RSA moonKey.pem "<optional passphrase>"
```

/etc/ipsec.conf:

```
conn host-host
    left=%defaultroute
    leftcert=moonCert.pem
    right=192.168.0.2
    rightid="C=CH, O=Linux strongSwan, CN=sun.strongswan.org"
    auto=start
```

Configuration on host *sun*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/sunCert.pem

/etc/ipsec.secrets:

```
: RSA sunKey.pem "<optional passphrase>"
```

/etc/ipsec.conf:

```
conn host-host
    left=%defaultroute
    leftcert=sunCert.pem
    right=192.168.0.1
```

```
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Four tunnel case

Σε ένα set up site – to – site ένας διαχειριστής συστημάτων που συνδέεται με την τοπική πύλη συχνά θα επιθυμούσε να έχει πρόσβαση στην ίδια αυτή πύλη ή έναν κεντρικό υπολογιστή στο υποδίκτυο πίσω από την πύλη αυτή μέσω ενός ασφαλούς **IPsec tunnel**. Δεδομένου ότι τα πακέτα IP που αφήνουν μια πύλη μέσω μιας εξωτερικής δικτυακής διεπαφής φέρνουν τη διεύθυνση IP, και τα τέσσερα **IPsec Security Associations (SAs)** πρέπει να σεταριστούν για να επιτύχουν την πλήρη συνδεσιμότητα. Στο παράδειγμα που ακολουθει φαίνετα πώς αυτό μπορεί να γίνει χωρίς πολλή πρόσθετη εργασία δακτυλογράφησης, χρησιμοποιώντας τη μακροεντολή που περιλαμβάνει τους ορισμούς σύνδεσης που καθορίζονται στο αρχείο ipsec.conf.

10.1.0.0/16 -- | 192.168.0.1 | === | 192.168.0.2 | -- 10.2.0.0/16
moon-net moon

sun sun-net

Configuration on gateway *moon*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/moonCert.pem

/etc/ipsec.secrets:

: RSA moonKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn net-net
    leftsubnet=10.1.0.0/16
    rightsubnet=10.2.0.0/16
    also host-host

conn net-host
    leftsubnet=10.1.0.0/16
    also host-host

conn host-net
    rightsubnet=10.2.0.0/16
    also host-host

conn host-host
    left=%defaultroute
```

```
    leftcert=moonCert.pem
    right=192.168.0.2
    rightid="C=CH, O=Linux strongSwan, CN=sun.strongswan.org"
    auto=start
```

Configuration on gateway *sun*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/sunCert.pem

/etc/ipsec.secrets:

: RSA sunKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn net-net
    leftsubnet=10.2.0.0/16
    rightsubnet=10.1.0.0/16
    also=host-host

conn net-host
    leftsubnet=10.2.0.0/16
    also=host-host

conn host-net
    rightsubnet=10.1.0.0/16
    also=host-host

conn host-host
    left=%defaultroute
    leftcert=sunCert.pem
    right=192.168.0.1
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Four Tunnel case the elegant way with source routing

Η πλήρης περίπτωση τεσσάρων tunnels που περιγράφεται στο προηγούμενο τμήμα γίνεται αρκετά σύνθετη. Εάν θα μπορούσε κάποιος να αναγκάσει τη διεύθυνση προέλευσης των πακέτων IP αφήνοντας την πύλη μέσω της εξωτερικής διεπαφής για να πάρουμε τη διεύθυνση IP της εσωτερικής διεπαφής, τότε θα μπορούσε να χρησιμοποιήσει το απλό subnet-to-subnet tunnel. Ένα τέτοιο setup είναι δυνατό εάν χρησιμοποιούμε την πηγαία δρομολόγηση των διαδρομών IP που χρησιμοποιείται ήδη από το strongSwan.

10.1.0.0/16 -- | 192.168.0.1 | === | 192.168.0.2 | -- 10.2.0.0/16
moon-net moon

sun sun-net

Εάν υποθέτουμε ότι η εσωτερική διεύθυνση IP του moon getaway είναι το 10.1.0.1 και η εσωτερική διεύθυνση IP του sun getaway είναι το 10.2.0.1, τότε η παράμετρος θα είναι:

leftsourceip=10.1.0.1

στον καθορισμό σύνδεσης του moon και

leftsourceip=10.2.0.1

στον sun, αντίστοιχα, θα εγκαταστήσει τη δρομολόγηση πηγής και στις δύο πύλες. Κατά συνέπεια η εντολή:

ping 10.2.0.1

εκτελείται στον moon και θα αφήσει την πύλη με μια διεύθυνση προέλευσης του 10.1.0.1 και επομένως θα πάρει το net-net IPsec tunnel.

Configuration on gateway *moon*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/moonCert.pem

/etc/ipsec.secrets:

: RSA moonKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn net-net
    left=%defaultroute
    leftsourceip=10.1.0.1
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    right=192.168.0.2
    rightsubnet=10.2.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=sun.strongswan.org"
    auto=start
```

Configuration on gateway *sun*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/sunCert.pem

/etc/ipsec.secrets:

: RSA sunKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn net-net
    left=%defaultroute
    leftsourceip=10.2.0.1
    leftsubnet=10.2.0.0/16
    leftcert=sunCert.pem
    right=192.168.0.1
    rightsubnet=10.1.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Roadwarrior case

Αυτή είναι μια πολύ κοινή περίπτωση όπου μια strongSwan πύλη εξυπηρετεί έναν αυθαίρετο αριθμό μακρινών πελατών (clients) VPN που συνήθως έχουν τις δυναμικές διευθύνσεις IP.

10.1.0.0/16 -- | 192.168.0.1 | === | x.x.x.x |
moon-net moon carol

Configuration on gateway *moon*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/moonCert.pem

/etc/ipsec.secrets:

: RSA moonKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn rw
    left=%defaultroute
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    right=%any
    auto=add
```

Configuration on roadwarrior *carol*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/carolCert.pem

/etc/ipsec.secrets:

: RSA carolKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn home
    left=%defaultroute
    leftcert=carolCert.pem
    right=192.168.0.1
    rightsubnet=10.1.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Roadwarrior case with virtual IP

Το Roadwarrior συνήθως ορίζει τις δυναμικές διευθύνσεις IP από ISP που είναι αυτήν την περίοδο συνημμένες. Προκειμένου να απλοποιηθεί η δρομολόγηση από το moon-net πίσω στην εξ' αποστάσεως πρόσβαση του πελάτη θα ήταν επιθυμητό, εάν το roadwarrior είχε μια διεύθυνση IP που επιλέγεται εσωτερική.

10.1.0.0/16 -- | 192.168.0.1 | === | x.x.x.x | -- 10.3.0.1
moon-net moon carol

virtual IP

Αυτή η εικονική διεύθυνση IP μπορεί να οριστεί σε ένα strongSwan roadwarrior με την προσθήκη της παραμέτρου:

leftsourceip=10.3.0.1

στο ipsec.conf του roadwarrior. Φυσικά η εικονική IP κάθε roadwarrior πρέπει να είναι ευδιάκριτη. Στο παράδειγμά μας επιλέγεται από τη διεύθυνση

rightsubnetwithin=10.3.0.0/16

η οποία μπορεί να προστεθεί στο ipsec.conf της πύλης έτσι ώστε ένας ενιαίος καθορισμός σύνδεσης να μπορεί να χειριστεί τα πολλαπλάσια roadwarriors.

Configuration on gateway *moon*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/moonCert.pem

/etc/ipsec.secrets:

: RSA moonKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn rw
    left=%defaultroute
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    right=%any
    rightsubnetwithin=10.3.0.0/16
    auto=add
```

Configuration on roadwarrior *carol*:

/etc/ipsec.d/cacerts/strongswanCert.pem

/etc/ipsec.d/certs/carolCert.pem

/etc/ipsec.secrets:

: RSA carolKey.pem "<optional passphrase>"

/etc/ipsec.conf:

```
conn home
    left=%defaultroute
    leftsourceip=10.3.0.1
    leftcert=carolCert.pem
    right=192.168.0.1
    rightsubnet=10.1.0.0/16
    rightid="C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
    auto=start
```

Generating certificates and CRLs with OpenSSL

Αυτή η παράγραφος δεν είναι ένα πραγματικό σεμινάριο στο πώς χρησιμοποιείται το OpenSSL. Απαριθμεί ακριβώς μερικά σημεία που είναι σχετικά με την παραγωγή πιστοποιητικών (Certificates, Keys) και το CRLs για τη χρήση με strongSwan.

Generating a CA certificate

Η δήλωση του OpenSSL,

openssl req -x509 -days 1460 -newkey rsa:2048 -keyout strongswanKey.pem -out strongswanCert.pem

δημιουργεί ένα ιδιωτικό RSA κλειδί των 2048 bits το strongswanKey.pem και μια υπογεγραμμένη CA (Certificate Authority) πιστοποίηση την strongswanCert.pem διάρκειας τεσσάρων χρόνων.

Η εντολή *openssl x509 -in cert.pem -noout -text*

Παρουσιάζει μία λίστα απο τις ιδιότητες του X.509 cert.pem πιστοποιητικού. Επιτρέπει να ελέγξει εάν οι προεπιλογές του configuration στο openssl.cnf έχουν εισαχθεί σωστά.

Εάν προτιμά κάποιος τα πιστοποιητικά CA για να χρησιμοποιεί το δυαδικό σχήμα DER έπειτα η ακόλουθη εντολή επιτυγχάνεται με αυτόν τον μετασχηματισμό:

openssl x509 -in strongswanCert.pem -outform DER -out strongswanCert.der

Ο κατάλογος */etc/ipsec.d/cacerts/* περιλαμβάνει όλα τα απαραίτητα CA πιστοποιητικά είτε σε δυαδική DER μορφή, είτε σε PEM. Ανεξάρτητα από την κατάληξη των αρχείων, ο Pluto "αυτόματα" καθορίζει το σωστό σχήμα.

Generating a host or user certificate

Η δήλωση του OpenSSL,

openssl req -newkey rsa:1024 -keyout hostKey.pem -out hostReq.pem

δημιουργεί ένα ιδιωτικό RSA κλειδί απο 1024 bits, το hostKey.pem, και ένα αίτημα πιστοποίησης, το hostReq.pem, τα οποία έχουν υπογραφτεί απο το CA.

Εάν θέλει κάποιος να προσθέσει ένα πεδίο *subjectAltName στον host,* τότε πρέπει να διαμορφώσει το OpenSSL configuration openssl.cnf και να προσθέσει την παρακάτω γραμμή στο κομμάτι *[usr_cert]:*

subjectAltName=DNS:moon.strongswan.org

εάν θέλει κάποιος να προσδιορίσει τον host απο το δικό του Fully Qualified Domain Name (*FQDN*), ή

subjectAltName=IP:160.85.22.3

εάν καποιος θέλει το ID να είναι τύπου IPV4_ADDR. Σίγουρα μπορεί να περικλύει και τα δύο παραπάνω, όπως,

subjectAltName=DNS:moon.strongswan.org,IP:160.85.22.3

αλλά η χρήση μιας διεύθυνσης IP για τον προσδιορισμό ενός host πρέπει να αποθαρρυνθεί οπωσδήποτε.

Για τα πιστοποιητικά χρηστών ο κατάλληλος τύπος ταυτότητας είναι ο USER_FQDN που μπορεί να διευκρινιστεί όπως,

subjectAltName=email:carol@strongswan.org

ή εάν η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη είναι μέρος του διακεκριμένου ονόματος,

subjectAltName=email:copy

Εάν έχει κάποιος μόνο ένα ενιαίο σημείο HTTP τότε θα υπάρχει η εντολή,

crlDistributionPoints="URI:http://crl.strongswan.org/strongswan.crl"

όπου επίσης λειτουργεί.

Τώρα το αίτημα πιστοποιητικών μπορεί να υπογραφεί από το CA με την εντολή:

openssl ca -in hostReq.pem -days 730 -out hostCert.pem –notext

Εάν παραλείψει κάποιος την επιλογή -days τότε η τιμή default_days (365 ημέρες) θα είναι προεπιλεγμένη στο openssl.cnf. Η επιλογή - notext δηλώνει ότι μια κατανοήσιμη από τον άνθρωπο λίστα του πιστοποιητικού είναι στο κωδικοποιημένο base64 των πιστοποιητικών.

Συνήθως ένας βασισμένος σε WINDOWS VPN πελάτης χρειάζεται το ιδιωτικό κλειδί του, το πιστοποιητικό του host, και το πιστοποιητικό CA. Ο καταλληλότερος τρόπος να φορτωθούν αυτές οι πληροφορίες είναι να τεθούν όλα σε ένα αρχείο PKCS#12:

openssl pkcs12 -export -inkey carolKey.pem -in carolCert.pem -name "carol" -certfile strongswanCert.pem -caname "strongSwan Root CA" -out carolCert.p12.

Generating a CRL

Ένα κενό CRL που υπογράφεται από το CA μπορεί να παραχθεί με την εντολή:

openssl ca -gencrl -crldays 15 -out crl.pem

Εάν παραλείψει κάποιος την επιλογή – crldays, τότε το default_crl_days θα είναι 30 μέρες το οποίο αναφέρεται στο αρχείο openssl.cnf.

Εάν προτιμηθεί το CRL να είναι σε δυαδικό σχήμα DER, τότε αυτή η μετατροπή μπορεί να επιτευχθεί με την εξής εντολή:

openssl crl -in crl.pem -outform DER -out cert.crl

Ο κατάλογος /etc/ipsec.d/crls/ περιέχει όλο το CRLs είτε σε δυαδικό DER είτε στο σχήμα base64 PEM. Ανεξάρτητα από το με την κατάληξη των αρχείων (der ή pem), ο pluto "αυτόματα" καθορίζει το σωστό σχήμα.

Revoking a certificate

Ένα συγκεκριμένο πιστοποιητικό των hosts που αποθηκεύεται στο αρχείο host.pem ανακαλείται με την εντολή:

openssl ca -revoke host.pem

Μετά το CRL μπορεί να ενημερωθεί με την εντολή:

openssl ca -gencrl -crldays 15 -out crl.pem

Το περιεχόμενο του αρχείου CRL μπορεί να φανεί με την εντολή:

openssl crl -in crl.pem -noout –text.

Αυτό γίνεται στην περίπτωση που είναι base64 CRL. Διαφορετικά σε DER διαμόρφωση θα είναι:

openssl crl -inform DER -in cert.crl -noout –text.

Configuring the connections - ipsec.conf

Configuring my side

Συνήθως η τοπική πλευρά (local side) είναι η ίδια για όλες τις συνδέσεις. Επομένως έχει νόημα να φτιαχτεί το configuration χαρακτηρίζοντας τη strongSwan πύλη ασφάλειας στο τμήμα conn %default του αρχείου /etc/ipsec.conf (configuration file). Εάν υποθέτουμε σε όλο το παρόν έγγραφο ότι η strongSwan πύλη ασφάλειας είναι το *left* και το peer είναι το *right*, τότε μπορούμε να γράψουμε:

```
conn %default
        # my side is left - the strongSwan SG
        left=160.85.22.2
        leftcert=moonCert.pem
       # load connection definitions automatically
        auto=add
```

Το πιστοποιητικό X.509 από το οποίο η strongSwan πύλη ασφάλειας θα επικυρώσει στέλνοντας μία δυαδική μορφή σ' εκείνον τον χρήστη (peer) ως τμήμα της βασικής ανταλλαγής Διαδικτύου (IKE) φαίνεται από την παρακάτω γραμμή:

leftcert=moonCert.pem

Το πιστοποιητικό μπορεί είτε να αποθηκευτεί σε base64 PEM διαμόρφωση, είτε στο δυαδικό DER. Ανεξάρτητα από την κατάληξη αρχείων, ο Pluto "αυτόματα" μπορεί να το αναγνωρίσει. Επομένως,

leftcert=moonCert.der

ή

leftcert=moonCert.cer

Κατά τη χρησιμοποίηση των σχετικών ονομάτων διαδρομής (pathnames), όπως στα παραπάνω παραδείγματα, τα αρχεία πιστοποιητικών πρέπει να αποθηκευτούν μέσα στον κατάλογο /etc/ipsec.d/certs. Προκειμένου να διακριθούν τα strongSwan πιστοποιητικά από τα τοπικά αποθηκευμένα εμπιστευμένα όμοια πιστοποιητικά, θα μπορούσαν επίσης να αποθηκευτούν σε ένα subdirectory μέσα στον κατάλογο /etc/ipsec.d/certs, όπως π.χ. μέσα στο

leftcert=mycerts/moonCert.pem

Φυσικά ονόματα διαδρομών είναι δυνατό να εκφραστούν,

leftcert=/usr/ssl/certs/moonCert.pem

Σαν ταυτότητα για την πύλη VPN προτείνεται η χρήση ενός Fully Qualified Domain Name (FQDN) της μορφής,

```
conn rw
        right=%any
        leftid=@moon.strongswan.org
```

Σημαντικό: Όταν ένα προσδιοριστικό FQDN χρησιμοποιείται, πρέπει να περιληφθεί ρητά το αποκαλούμενο subjectAltName του τύπου dnsName (DNS:) στο πιστοποιητικό που υποδεικνύεται από το leftcert.

Εάν δεν θέλει κάποιος να χρησιμοποιήσει τα subjectAltNames, μπορεί να χρησιμοποιήσει το διακεκριμένο όνομα του πιστοποιητικού (DN) αντ' αυτού, το οποίο είναι ένα προσδιοριστικό του τύπου DER_ASN1_DN και που μπορεί να γραφτεί π.χ. με το σχήμα LDAP ως εξής:

```
conn rw
        right=%any
        leftid="C=CH, O=strongSec GmbH, CN=moon.strongswan.org"
```

Δεδομένου ότι το DN είναι μέρος του πιστοποιητικού, το leftid δεν ειναι απαραίτητο να δηλωθεί ρητά. Κατά συνέπεια η είσοδος

```
conn rw
        right=%any
```

αυτόματα υποθέτει ότι το DN του leftcert είναι η ταυτότητα hosts.

Multiple certificates

Το strongSwan υποστηρίζει πολλαπλά πιστοποιητικά διάφορων τοπικών hosts καθώς επίσης και αντίστοιχα ιδιωτικά κλειδιά RSA:

```
conn rw1
        right=%any
        rightid=@peer1.domain1
        leftcert=myCert1.pem
         # leftid is DN of myCert1

conn rw2
        right=%any
        rightid=@peer2.domain2
        leftcert=myCert2.pem
        # leftid is DN of myCert2
```

Όταν ο peer1 αρχίζει μια σύνδεση, τότε το strongSwan θα στείλει ένα myCert1 και θα το υπογράψει με το myKey1 που καθορίζεται στο /etc/ipsec.secrets, ενώ το myCert2 και το myKey2 θα χρησιμοποιηθούν σε μια σύνδεση που αρχίζει από τον peer2.

Configuring the peer side using CA certificates

Τώρα μπορούμε να προχωρήσουμε να καθορίσουμε τις συνδέσεις μας. Η ακόλουθη απλούστερη δήλωση:

```
conn rw
        right=%any
```

καθορίζει τη γενική περίπτωση roadwarrior. Η γραμμή right=%any κυριολεκτικά σημαίνει ότι οποιοσδήποτε IPSec peer γίνεται αποδεκτός, ανεξάρτητα από την τρέχουσα διεύθυνση προέλευσης IP, καθώς και την ταυτότητά της, εφ' όσον παρουσιάσει ο peer ένα έγκυρο πιστοποιητικό X.509 που υπογράφεται από το CA και η strongSwan πύλη ασφάλειας την εμπιστεφτεί. Επιπλέον η υπογραφή κατά τη διάρκεια του κύριου τρόπου IKE δίνει την απόδειξη ότι ο peer κατέχει το ιδιωτικό κλειδί RSA που ταιριάζει με το δημόσιο κλειδί που περιλαμβάνεται στο διαβιβασθέν πιστοποιητικό.

Η ταυτότητα από την οποία ένας peer προσδιορίζεται κατά τη διάρκεια του κύριου τρόπου IKE, μπορεί να βρίσκεται σε οποιουσδήποτε από τους τύπους ταυτότητας ID όπως IPV4_ADDR, FQDN, USER_FQDN ή DER_ASN1_DN. Εάν ένας από τους πρώτους τρεις τύπους ταυτότητας χρησιμοποιείται, κατόπιν το πιστοποιητικό X.509 του peer πρέπει να περιέχει ένα subjectAltName τομέα του τύπου ipAddress (IP:), dnsName (DNS:) ή rfc822Name (ηλεκτρονικό ταχυδρομείο:), αντίστοιχα. Με τον τέταρτο τύπο DER_ASN1_DN, το προσδιοριστικό πρέπει εντελώς να ταιριάξει με το θεματικό χώρο του πιστοποιητικού του peer. Μια από τις δύο πιθανές αντιπροσωπεύσεις ενός διακεκριμένου ονόματος (DN) είναι το LDAP και φαίνεται απο την εντολή,

```
rightid="C=CH,O=Linux strongSwan, CN=sun.strongswan.org"
```

Το πρόσθετο κενό (whitespace) μπορεί να προστεθεί παντού δεδομένου ότι θα παραβλεθεί αυτόματα από το X.509. Μια εξαίρεση είναι το ενιαίο whitespace μεταξύ

των μεμονωμένων λέξεων, όπως π.χ. σε Linux strongSwan, το οποίο συντηρείται από το X.509.

Τα σχετικά διακεκριμένα ονόματα (Relative Distinguished Names ή RDNs) μπορούν εναλλακτικά να χωριστούν από μια κάθετο (“/”) αντί ενός κόμματος (“,”), όπως φαινεται παρακάτω.

rightid="/C=CH/O=Linux strongSwan/CN=sun.strongswan.org"

Η αντιπροσώπευση που εξάγεται από το πιστοποιητικό από την επιλογή γραμμών εντολής OpenSSL είναι,

openssl x509 -in sunCert.pem -noout –subject

Τα παρακάτω X.509 RDNs υποστηρίζοται απο το strongSwan,

DC	Domain Component
C	Country
ST	State or province
L	Locality or town
O	Organisation
OU	Organisational Unit
CN	Common Name
ND	Name Distinguisher, used with *CN*
N	Name
G	Given name
S	Surname
I	Initials
T	Personal title
E	E-mail
Email	E-mail
emailAddress	E-mail
SN	Serial number
serialNumber	Serialnumber
D	Description
UID	User ID
ID	X.500 Unique Identifier
TCGID	[Siemens] Trust Center Global ID
unstructuredName	Unstructured Name
UN	Unstructured Name
employeeNumber	Employee Number
EN	Employee Number

Με τον καθορισμό σύνδεσης roadwarrior που φαίνεται παραπάνω, μπορεί να εγκατασταθεί ένα IPsec SA για τη strongSwan πύλη ασφαλείας moon.strongswan.org. Εάν να είναι σε θέση οποιοδήποτε roadwarrior να φθάσει π.χ. στα δύο υποδίκτυα 10.0.1.0 /24 και το 10.0.3.0 /24 πίσω από την πύλη ασφάλειας, τότε οι ακόλουθοι ορισμοί σύνδεσης θα το καταστήσουν αυτό πιθανό

```
conn rw1
        right=%any
        leftsubnet=10.0.1.0/24

conn rw3
        right=%any
                leftsubnet=10.0.3.0/24
```

Σε κάποιους peers στην κατοχή ενός πιστοποιητικού X.509 που υπογράφεται από μια συγκεκριμένη αρχή πιστοποιητικών δεν θα δοθεί η πρόσβαση στην πύλη ασφάλειας Linux, και έπειτα ένα υποσύνολο τους μπορεί να φραχτεί με την απαρίθμηση των σειριακών αριθμών των πιστοποιητικών τους σε έναν κατάλογο ανάκλησης πιστοποιητικών (CRL). Διαφορετικά, ως εναλλακτική λύση, η πρόσβαση μπορεί να ελεγχθεί με ρητά να βάλει μια είσοδο roadwarrior για κάθε επιλέξιμο peer στο ipsec.conf:

```
conn sun
        right=%any
        rightid=@sun.strongswan.org

 conn carol
        right=%any
        rightid=carol@strongswan.org

 conn dave
        right=%any
        rightid="C=CH, O=Linux strongSwan, CN=dave@strongswan.org"
```

Όταν η διεύθυνση IP ενός peer είναι γνωστή, μπορεί να διευκρινιστεί επίσης. Αυτή η είσοδος είναι υποχρεωτική όταν θέλει ο strongSwan host να ενεργήσει ως αρχικός σε μια σύνδεση IPSec.

```
conn sun
        right=192.168.0.2
        rightid=@sun.strongswan.org

 conn carol
        right=192.168.0.100
        rightid=carol@strongswan.org

conn dave
        right=192.168.0.200
        rightid="C=CH, O=Linux strongSwan, CN=dave@strongswan.org"

conn venus
        right=192.168.0.50
```

Στο τελευταίο παράδειγμα οι τύποι ταυτότητας FQDN, USER_FQDN, DER_ASN1_DN και IPV4_ADDR, χρησιμοποιήθηκαν αντίστοιχα. Φυσικά όλοι οι ορισμοί σύνδεσης που παρουσιάζονται μέχρι τώρα έχουν περιλάβει τις γραμμές στο τμήμα conn %defaults, περιλαμβάνοντας μεταξύ των άλλων ένα *left* και *leftcert* μιας είσοδου.

Handling Virtual IPs and wildcard subnets

Συχνά τα roadwarriors είναι πίσω από τα NAT-boxes με το IPsec, το οποίο αναγκάζει την εσωτερική διεύθυνση προέλευσης IP ενός IPsec tunnel, να είναι διαφορετική από την εξωτερική διεύθυνση προέλευσης IP που ορίζεται συνήθως δυναμικά από ISP. Εκτιμώντας ότι η εξωτερική διεύθυνση IP μπορεί να αντιμετωπιστεί από την εντολή right=%any, η εσωτερική διεύθυνση ή το υποδίκτυο IP πρέπει πάντα να δηλωθεί σε έναν καθορισμό σύνδεσης. Επομένως για τα τρία roadwarriors rw1 μέχρι rw3 στη σύνδεση με μια strongSwan πύλη ασφάλειας, απαιτούνται οι ακόλουθες καταχωρήσεις στο /etc/ipsec.conf:

```
conn rw1
        right=%any
        righsubnet=10.0.1.5/32

 conn rw2
        right=%any
        rightsubnet=10.0.1.5.47/32

 conn rw3
        right=%any
        rightsubnet=10.0.1.128/28
```

Με τη νέα παράμετρο rightsubnetwithin αυτές οι τρεις καταχωρήσεις μπορούν να μειωθούν σε έναν απλό ορισμό σύνδεσης,

```
conn rw
        right=%any
        rightsubnetwithin=10.0.1.0/24
```

Οποιοσδήποτε host θα γίνει αποδεκτός (φυσικά μετά από την επιτυχή επικύρωση βασισμένη στο πιστοποιητικό X.509 του peer), εάν δηλώσει ένα υποδίκτυο πελατών (clients) που καθορίζεται απο του εξ' ορισμού υποδικτύου (στο παράδειγμά μας 10.0.1.0 /24). Για κάθε roadwarrior θα δημιουργηθεί μια περίπτωση σύνδεσης που προσαρμόζεται στο υποδίκτυο του πελάτη, βασισμένο στο γενικό πρότυπο rightsubnetwithin.

Αυτό το χαρακτηριστικό γνώρισμα στο strongSwan μπορεί επίσης να είναι χρήσιμο με τους πελάτες VPN που παίρνουν μια δυναμικά ορισμένη εσωτερική IP από έναν κεντρικό υπολογιστή DHCP που βρίσκεται στο δρομολογητή NAT.

Protocol and port selectors

Το strongSwan έχει την δυνατότητα να περιορίζει το πρωτόκολλο και προαιρετικά τις πόρτες (ports) σε ένα IPsec SA που χρησιμοποιούν τις παραμέτρους rightprotoport και leftprotoport.

Μερικά παραδείγματα:

```
conn icmp
        right=%any
```

```
        rightprotoport=icmp
        left=%defaultroute
        leftid=@moon.strongswan.org
        leftprotoport=icmp

conn http
        right=%any
        rightprotoport=6
        left=%defaultroute
        leftid=@moon.strongswan.org
        leftprotoport=6/80

conn lt2p      # with port wildcard for Mac OS X Panther interoperability
        right=%any
         rightprotoport=17/%any
        left=%defaultroute
        leftid=@moon.strongswan.org
        leftprotoport=17/1701

conn dhcp
        right=%any
        rightprotoport=udp/bootpc
        left=%defaultroute
        leftid=@moon.strongswan.org
        leftsubnet=0.0.0.0/0 #allows DHCP discovery broadcast
        leftprotoport=udp/bootps
        rekey=no
        keylife=20s
        rekeymargin=10s
         auto=add
```

Τα πρωτόκολλα και οι πόρτες μπορούν να εμφανίζονται από τις αριθμητικές τιμές τους που καθορίζονται στο /etc/services φαίνονται απο το,

ipsec status

και απαριθμεί τους ακόλουθους ορισμούς σύνδεσης:

```
"icmp": 192.168.0.1[@moon.strongswan.org]:1/0...%any:1/0
"http": 192.168.0.1[@moon.strongswan.org]:6/80...%any:6/0
"l2tp": 192.168.0.1[@moon.strongswan.org]:17/1701...%any:6/%any
"dhcp": 0.0.0.0/0===192.168.0.1[@moon.strongswan.org]:17/67...%any:17/68
```

Με βάση τις επιλογές πρωτοκόλλου και πορτών τα κατάλληλα eroutes θα σεταριστούν, έτσι ώστε μόνο οι διευκρινισμένοι τύποι ωφέλιμων φορτίων θα περάσουν μέσω του Ipsec tunnel.

IPsec policies based on wildcards

Στα μεγάλα VPN δίκτυα στην απομακρισμένη πρόσβαση υπάρχει συχνά μια απαίτηση ότι η πρόσβαση στα διάφορα μέρη ενός εσωτερικού δικτύου πρέπει να χορηγηθεί επιλεκτικά, π.χ. ανάλογα με την ιδιότητα μέλους ομάδας της απομακρισμένης σύνδεσης του χρήστη. Το strongSwan καθιστά αυτό πιθανό με την

εφαρμογή του φιλτραρίσματος στο διακεκριμένο όνομα του χρήστη VPN (ID_DER_ASN1_DN).

Αυτό γίνεται κατανοητό με το παρακάτω παράδειγμα:

Μια οργάνωση έχει ένα τμήμα πωλήσεων (OU=Sales) και μια ερευνητική ομάδα (OU=Research). Στο ενδοδίκτυο (intranet) της επιχείρησης υπάρχουν χωριστά υποδίκτυα για τις πωλήσεις (10.0.0.0 /24) και την έρευνα (10.0.1.0 /24), όπου και οι δύο ομάδες μοιράζονται έναν κοινό κεντρικό υπολογιστή δικτύου (web server) (10.0.2.100). Οι πελάτες VPN χρησιμοποιούν τις εικονικές διευθύνσεις IP που είτε ορίζονται σαν στατικές (statically), είτε μέσω DHCP-over-IPsec. Οι πωλήσεις και τα ερευνητικά τμήματα χρησιμοποιούν διευθύνσεις IP από ξεχωριστες διευθύνσεις DHCP (10.1.0.0 /24) και (10.1.1.0 /24), αντίστοιχα. Ένα πιστοποιητικό X.509 διανέμεται σε κάθε πελάτη, περιέχοντας στο υπαγόμενο διακεκριμένο όνομά του τη χώρα (C=CH), την επιχείρηση (O=ACME), την ιδιότητα μέλους ομάδας (OU=Sales ή OU=Research) και το κοινό όνομα (π.χ. CN=Bart Simpson).

Η πολιτική IPsec που καθορίζεται παραπάνω μπορεί τώρα να γίνει κατανοητή με τις ακόλουθες τρεις συνδέσεις ασφάλειας IPsec:

```
conn sales
        right=%any
        rightid="C=CH, O=ACME, OU=Sales, CN=*"
        rightsubnetwithin=10.1.0.0/24  # Sales DHCP range
        leftsubnet=10.0.0.0/24         # Sales subnet

conn research
        right=%any
        rightid="C=CH, O=ACME, OU=Research, CN=*"
        rightsubnetwithin=10.1.1.0/24   # Research DHCP range
        leftsubnet=10.0.1.0/24          # Research subnet

  conn web
        right=%any
        rightid="C=CH, O=ACME, OU=*, CN=*"
        rightsubnetwithin=10.1.0.0/23   # Remote access DHCP range
        leftsubnet=10.0.2.100/32        # Web server
        rightprotoport=tcp              # TCP protocol only
        leftprotoport=tcp/http          # TCP port 80 only
```

Ο «*» χαρακτήρας χρησιμοποιείται ως μπαλαντέρ (wildcard) στα σχετικά διακεκριμένα ονόματα (RDNs). Προκειμένου να ταιριάξει με ένα πρότυπο μπαλαντέρ, το ID_DER_ASN1_DN ενός peer πρέπει να περιέχει τον ίδιο αριθμό RDNs που εμφανίζεται στην ακριβή διαταγή που καθορίζεται από το πρότυπο:

"C=CH, O=ACME, OU=Research, OU=Special Effects, CN=Bart Simpson"

Και ταιριάζει με τα πρότυπα,

"C=CH, O=ACME, OU=Research, OU=*, CN=*"
"C=CH, O=ACME, OU=*, OU=Special Effects, CN=*"
"C=CH, O=ACME, OU=*, OU=*, CN=*"

Και όχι με τα προτυπα,

"C=CH, O=ACME, OU=*, CN=*"

τα οποία δεν έχουν τον ίδιο RDN αριθμό.

IPsec policies based on CA certificates

Σαν εναλλακτική λύση των βασισμένων πολιτικών IPsec που περιγράφονται στο παραπάνω κεφάλαιο, η πρόσβαση στο συγκεκριμένο πελάτη και στα υποδίκτυα μπορούν επίσης να ελεγχθούν βάσει του CA (Certificate Authority) που εξέδωσε το πιστοποιητικό ενός peer.

```
conn sales
right=%any
rightca="C=CH, O=ACME, OU=Sales, CN=Sales CA"
rightsubnetwithin=10.1.0.0/24 # Sales DHCP range
leftsubnet=10.0.0.0/24        # Sales subnet

conn research
right=%any
rightca="C=CH, O=ACME, OU=Research, CN=Research CA"
rightsubnetwithin=10.1.1.0/24 # Research DHCP range
leftsubnet=10.0.1.0/24        # Research subnet

conn web
right=%any
rightca="C=CH, O=ACME, CN=ACME Root CA"
rightsubnetwithin=10.1.0.0/23 # Remote access DHCP range
leftsubnet=10.0.2.100/32      # Web server
rightprotoport=tcp            # TCP protocol only
leftprotoport=tcp/http        # TCP port 80 only
```

Στο παραπάνω παράδειγμα, η σύνδεση «sales» μπορεί να χρησιμοποιηθεί από τους peers παραουσιάζοντας τα πιστοποιητικά που εκδίδονται μόνο από το Sales CA. Με τον ίδιο τρόπο, η χρήση της σύνδεσης «research» είναι περιορισμένη στους ιδιοκτήτες των πιστοποιητικών που εκδίδονται από την το Research CA. Η σύνδεση "web"είναι ανοικτή στους peers "Sales" αλλά και "Research", επειδή το απαραίτητο *ACME Root CA* είναι ο παράγοντας των Sales και Research. Εάν καμία παράμετρος rightca δεν είναι παρούσα, τότε απο οποιοδήποτε έγκυρο πιστοποιητικό που εκδίδεται από ένα εμπιστευμένο CA στο /etc/ipsec.d/cacerts μπορεί να χρησιμοποιηθεί από τον peer.

Η παράμετρος leftca συνήθως δεν ειναι απαραίτητο να τεθεί ρητά, επειδή εξ ορισμού τίθεται ο τομέας παραγόντων του πιστοποιητικού που φορτώνεται μέσω του leftcert. Η δήλωση,

```
rightca=%same
```

θέτει το CA που ζητείται από τον peer στο CA που χρησιμοποιείται από η ίδια τη αριστερή πλευρά όπως π.χ.

```
conn sales
        right=%any
        rightca=%same
        leftcert=mySalesCert.pem
```

Sending certificate requests

Η παρουσία της παραμέτρου rightca αναγκάζει επίσης το CA να σταλεί ως τμήμα του μηνύματος του αιτήματος πιστοποιητικών όταν το strongSwan είναι ο διαχειριστής. Μια πρόσθετη περίπτωση που εμφανίζεται, είναι όταν το strongSwan αποκρίνεται σε ένα roadwarrior. Εάν διάφορες συνδέσεις roadwarrior είναι βασισμένες σε διαφορετικό CA, καθορίζονται τότε όλα τα επιλέξιμα CA θα απαριθμηθούν στο μήνυμα αιτήματος πιστοποιητικών του Pluto.

IPsec policies based on group attributes

Τα πιστοποιητικά X.509, είναι ο ισχυρότερος μηχανισμός για τις πολιτικές ασφαλείας IPsec. Η παράμετρος rightgroups σε έναν καθορισμό σύνδεσης περιορίζει την πρόσβαση στα μέλη των απαριθμημένων ομάδων. Ένας peer IPsec πρέπει να εκδώσει ένα έγκυρο πιστοποιητικό ιδιοτήτων από το Authorization Authority και μια λίστα μια έγκρισης των απαραίτητων ιδιοτήτων ομάδας προκειμένου να γίνει αναγνωρίσιμος.

```
conn sales
        right=%any
        rightgroups="Sales"
        rightsubnetwithin=10.1.0.0/24 # Sales DHCP range
        leftsubnet=10.0.0.0/24        # Sales subnet

conn research
        right=%any
        rightgroups="Research"
        rightsubnetwithin=10.1.1.0/24 # Research DHCP range
        leftsubnet=10.0.1.0/24        # Research subnet

conn web
        right=%any
        rightgroups="Sales, Research"
        rightsubnetwithin=10.1.0.0/23 # Remote access DHCP range
        leftsubnet=10.0.2.100/32      # Web server
        rightprotoport=tcp            # TCP protocol only
        leftprotoport=tcp/http        # TCP port 80 only
```

Στα παραπάνω παραδείγματα απαιτούνται τα μέλη της ομάδας Sales για τις συνδέσεις Sales, και τα μέλη της ομάδας Research για την σύνδεση Research, ενώ η σύνδεση είναι προσβάσιμη και για τις δύο ομάδες. Αυτήν την περίοδο οι ιδιότητες των πιστοποιητικών των peers πρέπει να φορτωθούν στατικά μέσω του καταλόγου του /etc/ipsec.d/acerts/. Στο μέλλον οι ιδιότητες του strongSwan θα είναι δυνατές να προσκομίζονται από έναν κεντρικό υπολογιστή LDAP.

Configuring certificates and CRLs

Installing the CA certificates

Τα X.509 πιστοποιητικά που παραλαμβάνονται από το strongSwan κατά τη διάρκεια του πρωτοκόλλου IKE, επικυρώνονται αυτόματα και μπαίνουν σε λίστα εμπιστοσύνης, όπου και επιτυγχάνεται να υπογραφτεί ένα πιστοποιητικό, το root CA. Συνήθως τα πιστοποιητικά των hosts υπογράφονται άμεσα από ένα root CA, αλλά το strongSwan, επίσης, υποστηρίζει multi-level ιεραρχίες τις οποίες μπορεί να τις επιτύχει με το CA. Όλα τα CA πιστοποιητικά που ανήκουν σε μια αλυσίδα εμπιστοσύνης πρέπει να αντιγραφούν είτε με το δυαδικό σχήμα DER είτε με το base64 PEM στον κατάλογο,

/etc/ipsec.d/cacerts/

Εδώ υποστηρίζεται το CA, αλλά προς το παρόν δημιουργείται μια μεγάλη ομάδα των έγκυρων πιστοποιητικών χρηστών ή hosts και δεν μπορούν να διοριστούν στους συγκεκριμένους ορισμούς σύνδεσης στο /etc/ipsec.conf.

Installing optional certificate revocation lists (CRLs)

Με την αντιγραφή ενός πιστοποιητικού CA στο /etc/ipsec.d/cacerts/, αυτόματα όλα τα πιστοποιητικά χρηστών ή hosts που εκδίδονται από αυτό το CA κηρύσσονται έγκυρα. Δυστυχώς, τα ιδιωτικά κλειδιά πρέπει να πάρουν συμβιβασμένα ακούσια ή σκόπιμα, τα προσωπικά πιστοποιητικά των. Σε αυτήν την ανάκληση τα certificate revocation lists (CRLs) έχουν δημιουργηθεί. Το CRLs περιέχει τους σειριακούς αριθμούς όλων των πιστοποιητικών χρηστών ή hosts που έχουν ανακληθεί λόγω των διάφορων λόγων.

Μετά από την επιτυχή επαλήθευση της αλυσίδας εμπιστοσύνης X.509, ο Pluto ψάχνει τον κατάλογο του CRLs, το οποίο είτε λαμβάνεται με τη φόρτωση από τον κατάλογο του /etc/ipsec.d/crls/, είτε προσκομίζεται δυναμικά από ένα HTTP, είτε εκδίδεται το ldap του κεντρικού υπολογιστή για την παρουσία ενός CRL από το CA, όπου έχει υπογραφτεί το πιστοποιητικό.

Online Certificate Status Protocol (OCSP)

Το Online Certificate Status Protocol καθορίζεται από το RFC 2560. Μπορεί να χρησιμοποιηθεί για να ρωτήσει έναν κεντρικό υπολογιστή OCSP για την παρούσα κατάσταση ενός πιστοποιητικού X.509. Χρησιμοποιείται συχνά ως η δυναμικότερη εναλλακτική λύση ενός στατικού Certificate Revocation List (CRL). Και τα δύο αιτήματα OCSP που στέλνονται από τον πελάτη αλλά και η απάντηση OCSP που

επιστρέφεται από τον κεντρικό υπολογιστή μεταφέρονται μέσω μιας τυποποιημένης σύνδεσης TCP/HTTP. Επομένως η υποστήριξη cURL πρέπει να είναι ενεργοποιημένη στο pluto/Makefile:

```
# Uncomment this line to enable OCSP fetching using HTTP
LIBCURL=1
```

Στην απλούστερη οργάνωση OCSP, υπάρχει μια προεπιλογή URI κάτω από την οποία ο κεντρικός υπολογιστής OCSP όπου μπορεί να προσπελαστεί ένα δεδομένο CA και καθορίζεται στο ipsec.conf:

```
config setup
        crlcheckinterval=600

ca strongsec
        cacert=strongswanCert.pem
        ocspuri=http://ocsp.strongswan.org:8880
        auto=add
```

Η HTTP πόρτα μπορεί να επιλεχτεί ελεύθερα. Στο παράδειγμά μας έχουμε υποθέσει την πόρτα TCP 8880. Στο crlcheckinterval πρέπει να τεθεί μια τιμή διαφορετική από μηδέν. Διαφορετικά προσκομίζοντάς το, το OCSP δεν θα αρχίσει.

Το γνωστό openssl-0.9.7 από το http://www.openssl.org περιέχει έναν κεντρικό υπολογιστή OCSP που μπορεί να χρησιμοποιηθεί από κοινού με μια openSSL δημόσια βασική υποδομή. Ο πελάτης OCSP που ενσωματώνεται στον Pluto δεν περιέχει οποιοδήποτε κώδικα OpenSSL, αλλά είναι βασισμένος στην υπάρχουσα λειτουργία ASN.1 του patch X.509.

Το openSSL βασισμένο στον κεντρικό υπολογιστή OCSP αρχίζει με την ακόλουθη εντολή:

```
openssl ocsp -index index.txt -CA strongswanCert.pem -port 8880  -rkey
ocspKey.pem -rsigner ocspCert.pem  -resp_no_certs -nmin 60 –text
```

Η εντολή αποτελείται από τις παραμέτρους,

- **index**: το index.txt είναι ένα αντίγραφο του αρχείου δεικτών OpenSSL που περιέχει τον κατάλογο όλων των πιστοποιητικών. Η κατάσταση πιστοποιητικών στο indext.txt υποδεικνύεται είτε από το V για έγκυρη κατάσταση, είτε το R για ανακαλέσιμη κατάσταση. Εάν ένα νέο πιστοποιητικό προστίθεται ή εάν ένα πιστοποιητικό ανακαλείται χρησιμοποιώντας την εντολή ca του openssl, ο κεντρικός υπολογιστής OCSP πρέπει να ξαναξεκινήσει για τις αλλαγές στο index.txt που εφαρμόζονται.
- **CA** *: η πιστοποίηση CA*
- **port:** η http πόρτα στην οποία υπάρχει ο OCSP server.
- **rkey:** το κλειδί που χρησιμοποιείται σαν ιδιωτικό, για να υπογράψει την απάντηση OCSP. Η χρήση του ιδιωτικού κλειδιού ca δεν συστήνεται, δεδομένου ότι αυτό μπορεί να διακινδυνεύσει την ασφάλεια του PKI, εάν ο κεντρικός υπολογιστής OCSP δέχεται

επίθεση. Είναι πολύ καλύτερο να παραχθεί ένα ιδιωτικό κλειδί RSA μόνο για OCSP υπογράφοντας τη χρήση αντ' αυτού.

- **rsigner:** το πιστοποιητικό του κεντρικού υπολογιστή OCSP που περιέχει ένα δημόσιο κλειδί που ταιριάζει με το ιδιωτικό κλειδί και καθορίζεται απο το rkey και που μπορεί να χρησιμοποιηθεί από τον πελάτη για να ελέγξει την εμπιστοσύνη της υπογεγραμμένης απάντησης OCSP.
- **resp_no_certs:** με αυτήν την επιλογή η υπογράφουσα OCSP πιστοποίηση που καθορίζεται απο το rsigner, δεν συμπεριλαμβάνεται στην απάντηση OCSP.
- **nmin:** το διάστημα ισχύος μιας απάντησης OCSP και δίνεται σε λεπτά. 2*crlcheckinterval πριν από τη λήξη των απαντήσεων OCSP μια νέα ερώτηση αρχίζει από το Pluto προσκομίζοντας ένα νήμα (thread). Εάν ένα nmin λείπει ή να τίθεται η τιμή μηδέν, τότε το διάστημα ισχύος προεπιλογής που συντάσσεται απο το Pluto θα είναι 2 λεπτά, οδηγώντας σε μια σχεδόν one-time χρήση της απάντησης θέσης OCSP που δεν θα αναζωογονηθεί περιοδικά από το προσκομίζων νήμα.
- **text:** αυτή η επιλογή ενεργοποιεί μια παραγωγή αναγραφών, που παρουσιάζει το περιεχόμενο και του λαμβανόμενου αιτήματος OCSP και της σταλμένης απάντησης OCSP.

Πώς Pluto παίρνει τη λαβή του πιστοποιητικού υπογραφόντων OCSP; Υπάρχουν δύο δυνατότητες: Να τοποθετηθεί το πιστοποιητικό OCSP στον κατάλογο προεπιλογής,

/etc/ipsec.d/ocspcerts/

ή εναλλακτικά ο Pluto να μπορεί να τον λάβει ως τμήμα της απάντησης OCSP από το μακρινό κεντρικό υπολογιστή OCSP.

Configuring the peer side using locally stored certificates

Εάν δεν θέλει κάποιος να χρησιμοποιήσει τις αλυσίδες εμπιστοσύνης βασισμένες στα παραπάνω πιστοποιητικά ca, μπορεί εναλλακτικά να εισάγει τα εμπιστευόμενα όμοια πιστοποιητικά άμεσα στον Pluto. Κατά συνέπεια δεν είναι απαραίτητο να στηριχθεί στο πιστοποιητικό που διαβιβάζεται από τον peer ως τμήμα του πρωτοκόλλου IKE.

Εάν τα πιστοποιητικά ενός peer φορτώνονται τοπικά, τότε δεν υπάρχει καμία αίσθηση στην αποστολή οποιωνδήποτε πιστοποιητικών στο τέλος μέσω του κύριου πρωτοκόλλου IKE. Ειδικά εάν χρησιμοποιούνται υπογεγραμμένα πιστοποιητικά, δεν θα γίνονταν αποδεκτά απο οποιοδήποτε τρόπο στην άλλη πλευρά. Σε αυτές τις περιπτώσεις συστήνεται να προστεθεί,

leftsendcert=never

στον καθορισμό σύνδεσης [s] προκειμένου να αποφευχθεί η αποστολή του πιστοποιητικού του host. Η προκαθορισμένη τιμή είναι:

leftsendcert=always.

Εάν ένα πιστοποιητικό ενός peer περιέχει μια επέκταση subjectAltName, κατόπιν αυτό μπορεί να χρησιμοποιηθεί για να διευκρινίσει τον τύπο ενός εναλλακτικού rightid ή leftid, αντίστοιχα όπως παρουσιάζεται το παράδειγμα "conn sun". Εάν καμία είσοδος rightid ή leftid δεν είναι παρόν, τότε το υπαγόμενο διακεκριμένο όνομα που περιλαμβάνεται στο πιστοποιητικό λαμβάνεται ως ταυτότητα.

Χρησιμοποιώντας τους ίδιους κανόνες σχετικά με τα ονόματα διαδρομής κυκλώματος που ισχύουν για τα strongSwan πιστοποιητικά, οι ακόλουθοι δύο ορισμοί ισχύουν επίσης για τα εμπιστευμένα πιστοποιητικά των peers:

rightcert=peercerts/carolCert.der

ή

rightcert=/usr/ssl/certs/carolCert.der.

Installing the private key - ipsec.secrets

Loading private key files in PKCS#1 format

Εκτός από το τυποποιημένο FreeS/WAN RSA ιδιωτικό βασικό format, ο Pluto έχει διαμορφωθεί έτσι ώστε να φορτώνει τα ιδιωτικά κλειδιά RSA στη μορφή αρχείου PKCS#1. Τα βασικά αρχεία μπορούν να εξασφαλιστούν προαιρετικά με ένα passphrase.

Τα ιδιωτικά βασικά αρχεία RSA δηλώνονται στο /etc/ipsec.secrets χρησιμοποιώντας τη σύνταξη,

: RSA <my keyfile> "<optional passphrase>"

Το βασικό αρχείο μπορεί να είναι είτε σε σχηματισμό PEM base64 είτε σε δυαδικό DER. Η πραγματική κωδικοποίηση ανιχνεύεται "αυτόματα" από το Pluto. Το παράδειγμα,

: RSA moonKey.pem

χρησιμοποιεί ένα σχετικό όνομα διαδρομής. Σε εκείνη την περίπτωση ο Pluto θα ψάξει το ιδιωτικό βασικό αρχείο στον κατάλογο,

/etc/ipsec.d/private/

Σαν εναλλακτική λύση ένα απόλυτο όνομα διαδρομής μπορεί να δοθεί σαν,

: RSA /usr/ssl/private/moonKey.pem

Και στις δύο περιπτώσεις πρέπει τα βασικά αρχεία είναι αναγνώσιμα.

Συχνά ένα ιδιωτικό κλειδί πρέπει να μεταφερθεί από την αρχή πιστοποίησης όπου παρήχθη, στην πύλη ασφάλειας του προορισμού, όπου πρόκειται να χρησιμοποιηθεί. Προκειμένου να προστατευθεί το κλειδί μπορεί να κρυπτογραφηθεί με 3DES χρησιμοποιώντας ένα συμμετρικό κλειδί μεταφορών που προέρχεται από ένα κρυπτογραφικά ισχυρό passphrase.

openssl genrsa -des3 -out moonKey.pem 1024

Λόγω της αδύνατης ασφάλειας, τα βασικά αρχεία που προστατεύονται από το ενιαίο DES δεν θα γίνουν αποδεκτά από τον Pluto!!!

Μιά φορά στην πύλη ασφάλειας το ιδιωτικό κλειδί μπορεί καθένα να "ξεκλειδωθεί" μόνιμα έτσι ώστε να μπορεί να χρησιμοποιηθεί από τον Pluto χωρίς να πρέπει να είναι γνωστό ένα passphrase

openssl rsa -in moonKey.pem -out moonKey.pem

ή ως επιλογή το βασικό αρχείο να μπορεί να παραμείνει εξασφαλισμένο. Σε αυτήν την περίπτωση το passphrase που ξεκλειδώνει το ιδιωτικό κλειδί πρέπει να προστεθεί μετά από το όνομα διαδρομής κυκλώματος στο /etc/ipsec.secrets

: RSA moonKey.pem "This is my passphrase"

Μερικά CA διανέμουν τα ιδιωτικά κλειδιά που ενσωματώνονται σε ένα αρχείο PKCS#12. Δεδομένου ότι ο Pluto δεν είναι ικανός ακόμα να διαβάσει αυτό το σχηματισμό άμεσα, το ιδιωτικό μέρος κλειδί πρέπει πρώτα να εξαχθεί χρησιμοποιώντας την εντολή,

openssl pkcs12 -nocerts -in moonCert.p12 -out moonKey.pem

εάν το βασικό αρχείο moonKey.pem πρόκειται να εξασφαλιστεί πάλι από ένα passphrase, ή

openssl pkcs12 -nocerts -nodes -in moonCert.p12 -out moonKey.pem

εάν το ιδιωτικό κλειδί πρόκειται να αποθηκευτεί ξεκλειδωμένο.

Entering passphrases interactively

Σε μια πύλη VPN θα μπορούσε κάποιος να βάλει το passphrase προστατεύοντας το ιδιωτικό κλειδί αρχείων στο /etc/ipsec.secrets όπως περιγράφεται στην προηγούμενη παράγραφο, έτσι ώστε η πύλη να μπορεί να ξεκινήσει με ασφαλές τρόπο. Ο κίνδυνος κρατώντας τα μυστικά σε έναν κεντρικό υπολογιστή μπορεί να ελαχιστοποιηθεί σαν την "τοποθέτηση του κουτιού σε ένα κλειδωμένο δωμάτιο". Εφ' όσον δεν μπορεί να πάρει κανένας την πρόσβαση root στη μηχανή, τα ιδιωτικά κλειδιά είναι ασφαλή.

Σε έναν κινητό φορητό προσωπικό υπολογιστή η κατάσταση είναι αρκετά διαφορετική. Ο υπολογιστής μπορεί να κλαπεί ή ο χρήστης τον αφήνει αφύλακτο έτσι ώστε τα αναρμόδια πρόσωπα μπορούν να πάρουν την πρόσβαση σε αυτόν. Σε αυτές

τις περιπτώσεις θα ήταν προτιμότερο να μην κρατηθούν οποιαδήποτε passphrases ανοιχτά στο /etc/ipsec.secrets. Αυτό γίνεται εύκολα με τον καθορισμό,

: RSA moonKey.pem %prompt

Δεδομένου ότι το strongSwan αρχίζει συνήθως κατά τη διάρκεια της εκκίνησης διεργασίας, καμία κονσόλα των Windows δεν είναι διαθέσιμη που μπορεί να χρησιμοποιηθεί από τον Pluto για να προτρέψει το passphrase. Αυτό πρέπει να αρχίσει από το χρήστη με τη δακτυλογράφηση,

ipsec secrets

το όποιο είναι πραγματικά ένα ψευδώνυμο (alias) για την υπάρχουσα εντολή,

ipsec rereadsecrets

το οποίο εμφανίζεται με την εντολή,

need passphrase for '/etc/ipsec.d/private/moonKey.pem'
Enter:

Εάν το passphrase ήταν σωστό και το ιδιωτικό βασικό αρχείο θα μπορούσε να αποκρυπτογραφηθεί επιτυχώς,τότε θα ίσχυε,

valid passphrase

αποτελέσματα. Διαφορετικά η εντολή,

invalid passphrase, please try again
Enter:

θα δώσει μια άλλη δοκιμή.

Multiple private keys

Το strongSwan υποστηρίζει και πολλαπλάσια ιδιωτικά κλειδιά. Επειδή οι συνδέσεις που καθορίζονται στο ipsec.conf μπορούν να βρούν το σωστό ιδιωτικό κλειδί βάσει του δημόσιου κλειδιού που περιλαμβάνεται στο πιστοποιητικό που ορίστηκε μέσω της παραμέτρου leftcert, οι ιδιωτικοί βασικοί ορισμοί χωρίς συγκεκριμένο ID που μπορουν να μπορούν να χρησιμοποιηθούν θα είναι,

: RSA myKey1.pem "<optional passphrase1>"

: RSA myKey2.pem "<optional passphrase2>"

Configuring CA properties - ipsec.conf

Εκτός από τον καθορισμό των συνδέσεων IPsec το αρχείο ipsec.conf μπορεί επίσης, να χρησιμοποιηθεί για να διαμορφώσει μερικές ιδιότητες των αρχών πιστοποίησης που χρειάζεται για να εγκαταστήσει το X.509. Το ακόλουθο παράδειγμα παρουσιάζει τις παραμέτρους που είναι διαθέσιμες σήμερα:

```
ca strongswan
        cacert=strongswanCert.pem
        ocspuri=http://ocsp.strongswan.org:8880
        crluri=http://crl.strongswan.org/strongswan.crl'
        crluri2="ldap:///O=Linux strongSwan, C=CH?certificateRevocationList"
         ldaphost=ldap.strongswan.org
         auto=add
```

Με παρόμοιο τρόπο όπως τα conn τμήματα τα οποία χρησιμοποιούνται για τους ορισμούς σύνδεσης, ένας αυθαίρετος αριθμός προαιρετικών τμημάτων ca καθορίζει τις βασικές ιδιότητες του CA.

Κάθε τμήμα ca ονομάζεται με μια μοναδική ετικέτα,

```
ca strongswan
```

Η μόνη υποχρεωτική παράμετρος είναι,

```
cacert=strongswanCert.pem
```

σύμφωνα με την οποία σημείο στο πιστοποιητικό CA βρίσκεται συνήθως στον κατάλογο /etc/ipsec.d/cacerts/. Θα μπορούσε, επίσης, να ανακτηθεί μέσω ενός απόλυτου ονόματος μονοπατιού. Εάν το πιστοποιητικό CA αποθηκεύεται σε μια έξυπνη κάρτα, τότε θα ισχύει,

```
cacert=%smartcard#<>>
```

ή διαφορετικά

```
cacert=%smartcard<optional slot nr>:<key id>
```

και μπορεί να χρησιμοποιηθεί. Από το πιστοποιητικό εξάγονται το διακεκριμένο όνομα του CA και ο σειριακός αριθμός. Εάν ένας προαιρετικός subjectKeyAuthentifier είναι παρών, τότε μπορεί να χρησιμοποιηθεί για να προσδιορίσει μεμονωμένα τις διαδοχικές γενεές των πιστοποιητικών CA που φέρνουν το ίδιο διακεκριμένο όνομα.

Το OCSP URI

```
ocspuri=http://ocsp.strongswan.org:8880
```

επιτρέπει να καθορίσει έναν μεμονωμένο κεντρικό υπολογιστή OCSP ανά CA. Επίσης, μέχρι δύο πρόσθετα σημεία διανομής CRL (CDPs) που μπορούν να καθοριστούν ειναι,

```
crluri=http://crl.strongswan.org/strongswan.crl'
crluri2="ldap:///O=Linux strongSwan, C=CH?certificateRevocationList"
```

τα οποία προστίθονται σε οποιοδήποτε CDP και είναι "παρών" στα λαμβανόμενα πιστοποιητικά. Η τελευταία παράμετρος,

```
ldaphost=ldap.strongswan.org
```

μπορεί να χρησιμοποιηθεί για να συμπληρώσει το πραγματικό όνομα κεντρικών υπολογιστών σε LDAP CDP, όπου ο host λείπει όπως π.χ. στο παραπάνω crluri2. Στο μέλλον χρησιμοποιείται η παράμετρος ldaphost για να ανακτήσει τα πιστοποιητικά χρηστών ή hosts και ιδιοτήτων.

Με τη δήλωση auto=add καθορίζεται το ca και φορτώνεται αυτόματα ο Pluto κατά τη διάρκεια του ξεκινήματος συστημάτων. Η ρύθμιση auto=ignore θα αγνοήσει το τμήμα ca. Οι πρόσθετοι ορισμοί ca μπορούν να φορτωθούν από το ipsec.conf κατά τη διάρκεια του χρόνου εκτέλεσης με την εντολή,

```
ipsec auto --type ca --add strongswan-sales
```

και το

```
ipsec auto --type ca --delete strongswan-sales
```

διαγράφει το επονομαζόμενο ca. Και τελικά η εντολή,

```
ipsec auto --type ca --replace strongswan
```

αρχικά διαγράφει τον παλαιό καθορισμό Pluto στη μνήμη και φορτώνει έπειτα, την ενημερωμένη έκδοση από ipsec.conf. Οποιεσδήποτε παράμετροι που εμφανίζονται σε διάφορους ορισμούς ca μπορούν να τεθούν σε ένα κοινό τμήμα ca %default

```
ca %default
        ldaphost=ldap.strongswan.org
```

Smartcard Support

Configuring a smartcard-based connection

Ο καθορισμός μιας smartcard βασισμένης σύνδεσης σε ipsec.conf είναι:

```
conn sun
    right=192.168.0.2
    rightid=@sun.strongswan.org
    left=%defaultroute
    leftcert=%smartcard
    auto=add
```

Στις περισσότερες περιπτώσεις υπάρχει ένας ενιαίος αναγνώστης έξυπνων καρτών ή και μόνο ένα ιδιωτικό κλειδί RSA που αποθηκεύεται ακίνδυνα στη crypto συσκευή. Κατά συνέπεια, συνήθως η είσοδος,

leftcert=%smartcard

η οποία αντιπροσωπεύει την πλήρη σημείωση,

leftcert=%smartcard#1

είναι ικανοποιητική όπου χρησιμοποιείται απο το πρώτο πιστοποιητικό/ ιδιωτικό κλειδί που απαριθμείται από το PKCS#11. Εάν διάφορα πιστοποιητικά/ιδιωτικά κλειδιά είναι παρόντα, τότε το ένατο αντικείμενο μπορεί να επιλεγει χρησιμοποιώντας την παράμετρο,

leftcert=%smartcard#<n>

Η εντολή,

ipsec listcards

δίνει μια επισκόπηση σε όλα τα αντικείμενα πιστοποίησης που παρέχονται από την ενότητα PKCS#11. Τα πιστοποιητικά ca είναι αυτόματα διαθέσιμα ως εμπιστεύσιμα χωρίς την ανάγκη να αντιγραφούν πρώτα στον κατάλογο του /etc/ipsec.d/cacerts/.

Σαν εναλλακτική λύση το ID πιστοποιητικών ή/και ο αριθμός των slots που καθορίζονται από τα πρότυπα PKCS#11 μπορούν να διευκρινιστούν χρησιμοποιώντας τη σημείωση,

leftcert=%smartcard<slot nr>:<key id in hex format>

Κατά συνέπεια η εντολή,

leftcert=%smartcard:50

θα κοιτάξει σε όλες τα διαθέσιμα slots για την ταυτότητα 0x50 που αρχίζει με την πρώτο slot, ενώ η εντολή,

leftcert=%smartcard4:50

θα ελέγξει άμεσα το slot 4 για ένα κλειδί με την ταυτότητα 0x50.

Configuring the clients

StrongSwan

Μία strongSwan σε strongSwan σύνδεση είναι συμμετρική. Οποιοιδήποτε από τους τέσσερις καθορισμένους τύπους ID μπορούν να χρησιμοποιηθούν, ακόμη και σε διαφορετικούς τύπους στο τέλος της κάθεμιας σύνδεσης, αν και αυτό δεν θα είχε πολύ νόημα.

Connection Definition	ID type	subjectAltName
rightid (strongSwan)	DER_ASN1_DN	-
	FQDN	DNS:
	USER_FQDN	Email:
	IPV4_ADDR	IP:
leftid (strongSwan)	DER_ASN1_DN	-
	FQDN	DNS:
	USER_FQDN	Email:
	IPV4_ADDR	IP:

Windows 2000/XP

Τα Windows 2000 και τα Windows XP στέλνουν πάντα τον τύπο ID DER_ASN1_DN. Επομένως το rightid στον καθορισμό σύνδεσης της strongSwan ασφάλειας της πύλης πρέπει να είναι ένα διακεκριμένο ASN.1 όνομα.

Στα λαμβάνοντα Windows κατεύθυνσης 2000 ή τα Windows XP δέχονται και τους τέσσερις τύπους ID από το strongSwan.

Connection Definition	ID type	subjectAltName
rightid (Windows 2000/XP)	DER_ASN1_DN	-
leftid (strongSwan)	DER_ASN1_DN	-
	FQDN	DNS:
	USER_FQDN	Email:
	IPV4_ADDR	IP:

Authentication with raw RSA public keys

Το FreeS/WAN δεδομένου ότι είναι διαθέσιμο από το www.freeswan.org καλεί το δημόσιο κλειδί βάση αυθεντικοποίησης απο τα δημόσια κλειδιά RSA που καθορίζονται άμεσα στο /etc/ipsec.conf

rightrsasigkey=0sAq4c....

Όταν η έκδοση 1.x FreeS/WAN λαμβάνει ένα αίτημα πιστοποιητικών "certificate request" (CR), διαπραγματεύεται αμέσως, επειδή δεν ξέρει πώς να απαντήσει στο αίτημα. Είναι γεγονός ότι ένας workaround strongSwan δεν στέλνει ένα CR, εάν το κλειδί RSA έχει φορτωθεί στατικά χρησιμοποιώντας το rightrsasigkey. Ένα πρόβλημα που παραμένει είναι, πότε τα roadwarriors αρχίζουν μια σύνδεση. Όταν το strongSwan δεν ξέρει την ταυτότητα του peer εκ των προτέρων, θα στείλει πάντα ένα CR, προκαλώντας τη "ρήξη" της διαπραγμάτευσης IKE, εάν ο peer είναι τυποποιημένος host FreeS/WAN. Για να παρακάμψει αυτό το

πρόβλημα η παράμετρος διαμόρφωσης nocrsend μπορεί να τεθεί ως στόχος στο τμήμα "σεταρίσματος" config του /etc/ipsec.conf:

```
config setup:
        nocrsend=yes
```

Με αυτήν την παράμετρο κανένα αίτημα πιστοποιητικών δεν στέλνεται σε οποιαδήποτε σύνδεση. Η ρύθμιση προεπιλογής είναι nocrsend=no.

Additional features

Authentication and encryption algorithms

Το strongSwan υποστηρίζει την ακόλουθη ακολουθία των αλγορίθμων κρυπτογράφησης και επικύρωσης και για IKE και για ESP τα ωφέλιμα φορτία.

IKE algorithms (negotiated in Phase 1 Main Mode)	
Encryption algorithms	3des, aes, serpent, twofish, blowfish
Hash algorithms	md5, sha, sha2
DH groups	1024, 1536, 2048, 3072, 4096, 6144, 8192

ΣΗΜΕΙΩΣΗ: Για IKE oSHA-1 αλγόριθμος δείχνεται από το sha.

Οι κρυπτογραφικοί αλγόριθμοι IKE που φαίνονται παραπάνω, είναι ένα σταθερό μέρος της strongSwan διανομής. Οι ξεχωριστοί αλγόριθμοι μπορούν να προστεθούν ή να αφαιρεθούν απο τον κατάλογο προγράμματα/pluto/alg.

ESP algorithms (negotiated in Phase 2 Quick Mode)	
Encryption algorithms	3des, aes, serpent, twofish, blowfish
Hash algorithms	md5, sha1, sha2
PFS groups	1024, 1536, 2048, 3072, 4096, 6144, 8192

Οι κρυπτογραφικοί ESP αλγόριθμοι που φαίνονται παραπάνω είναι ένα σταθερό μέρος της strongSwan διανομής. Εάν ο πυρήνας 2.4 ή 2.6 Linux περιλαμβάνει το CryptoAPI, τότε οι αλγότιθμοι του πρόσθετου ESP μπορούν να προστεθούν ή να διαγραφούν ως kernel modules.

Οι κρυπτογραφικοί αλγόριθμοι IKE και ESP που προτείνονται στον peer ως αρχικοί, και μπορεί να διευκρινιστεί σε κάθε σύνδεση στη μορφή,

```
conn normal
        ...
        ike=aes128-sha-modp1536,3des-sha-modp1536
         esp=aes128-sha1,3des-sha1
        ...
```

ή (με ρίσκο)

```
conn paranoid
        ...
         ike=aes256-sha2_512-modp8192
        esp=aes256-sha2_256
        ...
```

Εάν οι οι παράμετροι διαμόρφωσης ike και ESP λείπουν στο ipsec.conf, τότε οι προεπιλεγμένες ρυθμίσεις θα είναι,

```
ike=3des-md5-modp1536,3des-sha-modp1536, 3des-md5-modp1024,3des-sha-modp1024
esp=3des-md5,3des-sha1
```

Ο 3DES αλγόριθμος κρυπτογράφησης και οι MD5 και SHA-1 hash αλγόριθμοι είναι στο strongSwan και δεν μπορούν να αφαιρεθούν.

Εάν προτιμηθεί η Perfect Forward Secrecy (PFS), κατόπιν μια τέτοια ομάδα μπορεί να συνταχθεί ως εξής:

```
conn make_sure
        ...
        pfs=yes
        pfsgroup=modp2048,modp1536
        ...
```

Εάν η παράμετρος pfs λείπει, τότε εξ' ορισμού θα ισχύει pfs=yes. Αυτό σημαίνει ότι προκειμένου να τεθεί εκτός λειτουργίας το PFS, πρέπει να ρυθμιστεί το pfs=no.

Εάν η παράμετρος pfsgroup λείπει, τότε η προεπιλεγμένη τιμή θα είναι,

pfsgroup=<*Phase1 DH group*>

Οι παράμετροι ike και ESP χρησιμοποιούνται για να διατυπώσουν μια ή περισσότερες προτάσεις μετατροπής που στέλνονται στον peer εάν strongSwan είναι ο διαχειριστής.

Προσοχή! Σαν παραλήπτη η πρώτη πρόταση που παραλαμβάνεται από τον peer γίνεται αποδεκτή, και αυτό υποστηρίζεται απο έναν από τους καταχωρημένους αλγορίθμους που απαριθμούνται από την εντολή,

```
ipsec listalgs
```

Εάν το strongSwan ως παραλήπτη θέλει να περιορίσει τις επιτρεπώμενες cipher ακολουθίες, τότε μπορεί να χρησιμοποιηθεί η ακριβής σημαία "!" (σημάδι θαυμαστικών). Το configuration,

```
conn normal_but_strict
         ...
        ike=aes128-sha-modp1536,3des-sha-modp1536!
```

```
esp=aes128-sha1,3des-sha1!
...
```

θα επιτρέψει μόνο τους απαριθμημένους αλγορίθμους που καθορίζονται παραπάνω. Όλες οι άλλες μέθοδοι απορρίπτονται ακόμα κι αν το strongSwan θα ήταν σε θέση να τους υποστηρίξει.

Dead peer detection

Το strongSwan εφαρμόζει τη διαμόρφωση του RFC 3706, Dead Peer Detection (DPD) keep-alive. Εάν ένα καθιερωμένο IPsec SA είναι μη απασχολούμενο (δηλ. χωρίς οποιαδήποτε κυκλοφορία) για τα δευτερόλεπτα N (dpddelay=N), τότε το strongSwan στέλνει ένα μήνυμα "hello" (R_U_THERE), και εάν ο peer υποστηρίζει DPD, τότε απαντά με αναγνωρίστικό μήνυμα (R_U_THERE_ACK). Εάν δεν λαμβάνεται καμία απάντηση, κατόπιν τα μηνύματα R_U_THERE επαναλαμβάνονται έως ότου έχει παρέλθει ένα διάλειμμα DPD των δευτερολέπτων M (dpdtimeout=M). Εάν, ακόμα, καμία κυκλοφορία ή κανένα πακέτο R_U_THERE_ACK δεν έχούν παραληφθεί, τότε ο peer δηλώνεται για να είναι εκτός και όλο το SA όπου ανήκει σε μια κοινή φάση ενός SA, διαγράφεται.

Η υποστήριξη DPD είναι tuneable σε κάθε σύνδεση με τη χρησιμοποίηση του dpdaction, dpddelay και dpdtimeout των οδηγιών:

```
conn roadwarrior
        right=%any
        left=%defaultroute
       leftsubnet=10.1.0.0/16
       dpdaction=clear

conn net-to-net
       right=192.168.0.1
       rightsubnet=10.2.0.0/16
       left=%defaultroute
       leftsubnet=10.1.0.0/16
       dpdaction=hold
       dpddelay=60
       dpdtimeout=500
```

Στο πρώτο παράδειγμα το dpdaction=clear ενεργοποιεί το μηχανισμό DPD με την προϋπόθεση ότι ο peer υποστηρίζει το RFC 3706. Οι τιμές dpddelay=30s και dpdtimeout=120s υποτίθενται εξ ορισμού, έτσι ώστε κατά τη διάρκεια της μη απασχόλησης περιόδων σε ένα πακέτο R_U_THERE να στέλνονται κάθε 30 δευτερόλεπτα. Εάν καμία κυκλοφορία ή ένα πακέτο R_U_THERE_ACK δεν παραλαμβάνονται από τον peer μέσα σε μια χρονική περίοδο διάρκειας δευτερολέπτων 120, ο peer θα κηρυχτεί εκτος σύνδεσης και όλο το SA και οι συνδεμένοι eroutes θα είναι ανενεργοί.

Στο δεύτερο παράδειγμα R_U_THERE, τα πακέτα στέλνονται κάθε 60 δευτερόλεπτα και η παράμετρος που τίθεται dpdaction=hold θα βάλει το eroute της κομμένης σύνδεσης σε μία κατάσταση %trap, έτσι ώστε, όταν θα εμφανιστεί η νέα

εξερχόμενη κυκλοφορία, correspondig θα επαναδιαπραγματευθεί η σύνδεση αυτόματα μόλις είναι πάλι επάνω ο peer.

Συστήνεται να χρησιμοποιείται dpdaction=hold για τις statically καθορισμένες συνδέσεις και dpdaction=clear για τις δυναμικές συνδέσεις roadwarrior. Η προκαθορισμένη τιμή είναι dpdaction=none, το οποίο θέτει εκτός λειτουργίας το DPD.

IKE Mode Config Pull Mode

Το πρωτόκολλο IKE Mode Config <draft-ietf-ipsec-isakmp-mode-cfg-04.txt>, επιτρέπει τη δυναμική ανάθεση πληροφοριών των εικονικών διευθύνσεων IP και ενός προαιρετικού DNS αλλά και WINS server στους πελάτες IPsec. Σαν προεπιλογή ο «Mode Config Pull Mode» χρησιμοποιείται, όταν ο πελάτης στέλνει ενεργά ένα αίτημα Mode Config στον κεντρικό υπολογιστή (server), προκειμένου να ληφθεί μια εικονική IP. Ο κεντρικός υπολογιστής απαντά με ένα μήνυμα απάντησης Mode Config που περιέχει τις ζητούμενες πληροφορίες.

Client μεριά Configuration (carol)

```
conn home
        right=192.168.0.1
        rightsubnet=10.1.0.0/16
        rightid=@moon.strongswan.org
        left=%defaultroute
        leftsourceip=%modeconfig
        leftcert=carolCert.pem
        leftid=carol@strongswan.org
        auto=start
```

Server μεριά configuration (moon)

```
conn roadwarrior
        right=%any
        rightid=carol@strongswan.org
        rightsourceip=10.3.0.1
        left=%defaultroute
        leftsubnet=10.1.0.0/16
        leftcert=moonCert.pem
        leftid=@moon.strongswan.org
        auto=add
```

Ο wildcard %modeconfig που χρησιμοποιείται στην παράμετρο leftsourceip του πελάτη θα προκαλέσει ένα αίτημα Mode Config. Αυτήν την στιγμή ο κεντρικός υπολογιστής θα επιστρέψει την εικονική διεύθυνση IP, που καθορίζεται από την παράμετρο rightsourceip. Στο μέλλον θα χρησιμοποιηθεί ένας LDAP βασισμένος μηχανισμός συμβούλευσης.

IKE Mode Config Push Mode

Ο εξοπλισμός της Cisco VPN χρησιμοποιεί τον εναλλακτικό «Mode Config Push Mode», όπου ο αρχικός πελάτης περιμένει τον κεντρικό υπολογιστή να ξεκινήσει κάτω μια εικονική διεύθυνση μέσω ενός καθορισμένου μηνύματος Mode Config. Η παραλαβή αναγνωρίζεται από τον πελάτη με ένα μήνυμα Mode Config ack. Ο Mode Config Push Mode ενεργοποιείται από την παράμετρο,

modeconfig=push

ως τμήμα του καθορισμού σύνδεσης στο ipsec.conf. Η προκαθορισμένη τιμή για τον Mode Config είναι modeconfig=pull.

Βιβλιογραφία

Internet

http://www.strongswan.org/

http://en.wikipedia.org/wiki/Virtual_private_network

http://tools.ietf.org/html/rfc4555

http://manpages.ubuntu.com/manpages/intrepid/man5/ipsec.conf.5.html

Βιβλία

Computer Networking A Top-Down Approach Featuring the Internet James F. Kurose and Keith W. Ross (Παράγραφος 7.8).

E-Books

LinuxTag2008-strongSwan.pdf