ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

Κατανεμημένο Σύστημα Ασφαλούς Πρόσβασης 3 Σταδίων με Ιδιωτικό Δυναμικό DNS και Secure Shell (SSH), με Εφαρμογές στη Ναυτιλία

Πατάπιος - Γεώργιος Μήτρου - Μπιτζίνης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων: Γεώργιος Σταμούλης, Καθηγητής

Λαμία, Νοέμβριος 2021

UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

3 Tier Distributed Secure Access System Incorporating Private Dynamic DNS and Secure Shell (SSH) for Maritime Applications

Patapios - Georgios Mitrou - Bitzinis

Master Thesis

Supervisor: Georgios Stamoulis, Professor

Lamia, November 2021

Η σελίδα αυτή είναι σκόπιμα κενή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ

«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ, ΔΙΑΧΕΙΡΙΣΗ ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ»

Κατανεμημένο Σύστημα Ασφαλούς Πρόσβασης 3 Σταδίων με Ιδιωτικό Δυναμικό DNS και Secure Shell (SSH), με Εφαρμογές στη Ναυτιλία

Πατάπιος - Γεώργιος Μήτρου - Μπιτζίνης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων: Γεώργιος Σταμούλης, Καθηγητής

Λαμία, Νοέμβριος 2021

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Κατανεμημένο Σύστημα Ασφαλούς Πρόσβασης 3 Σταδίων με Ιδιωτικό Δυναμικό DNS και Secure Shell (SSH), με Εφαρμογές στη Ναυτιλία» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Πατάπιος - Γεώργιος Μήτρου - Μπιτζίνης

Λαμία, 05 Νοεμβρίου 2021

Υπογραφή

Κατανεμημένο Σύστημα Ασφαλούς Πρόσβασης 3 Σταδίων

με Ιδιωτικό Δυναμικό DNS και Secure Shell (SSH),

με Εφαρμογές στη Ναυτιλία

Πατάπιος - Γεώργιος Μήτρου - Μπιτζίνης

Τριμελής Επιτροπή:

Επιβλέπων: Γεώργιος Σταμούλης, Καθηγητής

Αντώνιος Δαδαλιάρης, Επίκουρος Καθηγητής

Γεώργιος Δημητρίου, Επίκουρος Καθηγητής

Αφιερώνεται,

Στην σύζυγο μου, Μαγδαληνή, για την αγάπη και την στήριξη της σε κάθε πτυχή της ζωής μου, αποτελώντας πάντα αστείρευτη πηγή έμπνευσης και δύναμης.

Στον μέντορα και φίλο, Ειδικό Επιστήμονα κ.Γεώργιο Καρούμπαλη, που όλα αυτά τα χρόνια πίστεψε σε εμένα και σε συνδυασμό με τον αυθεντικό του χαρακτήρα, άοκνα με δίδαξε και με καθοδήγησε ώστε να κατακτήσω το πολυπόθητο αυτό σκαλοπάτι της γνώσης.

Πατάπιος - Γεώργιος Μήτρου - Μπιτζίνης

Περίληψη

Παρουσιάζεται και αναλύεται ένα κατανεμημένο σύστημα ασφαλούς πρόσβασης τριών σταδίων, με ιδιωτικό δυναμικό DNS και Secure Shell (SSH). Το σύστημα εξ ολοκλήρου στηρίζεται σε δωρεάν λογισμικό ανοιχτού κώδικα και υλοποιήθηκε σε embedded πλατφόρμα υλικού, για λόγους φορητότητας.

Λέξεις κλειδιά: Dynamic DNS, Secure Shell (SSH), OpenBSD, Απομακρυσμένη Πρόσβαση.

Abstract

A three stage distributed secure remote access system is presented and analyzed, comprising private Dynamic DNS and Secure Shell. The system is entirely based on free, open source software and is implemented on an embedded platform facilitating system portability.

Keywords: Dynamic DNS, Secure Shell (SSH), OpenBSD, Remote Access.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή

1.1 Αντικείμενο - Πεδίο Εφαρμογής

Το σύστημα σχεδιάστηκε και υλοποιήθηκε προκειμένου να λύσει συγκεκριμένο πρόβλημα απομακρυσμένης πρόσβασης χρηστών με φορητά συστήματα σε εφαρμογές υψηλών απαιτήσεων ασφάλειας, όπου ένα απλό ιδιωτικό εικονικό δίκτυο δεν επαρκεί.

1.2 Δομή της Διπλωματικής Εργασίας

Η παρούσα εργασία διαρθρώνεται σε πέντε κεφάλαια. Στα πρώτα δύο γίνεται μια προσπάθεια ώστε να γίνει κατανοητή η ιδέα υλοποίησης του συστήματος και στα υπόλοιπα τρία αναλύονται τα θέματα που αφορούν στην κατασκευή του συστήματος καθώς και στο πρακτικό κομμάτι της εργασίας.

Αναλυτικότερα, στο πρώτο κεφάλαιο γίνεται μια παρουσίαση της αντίληψης γύρω από την οποία στηρίχθηκε η ιδέα υλοποίησης του συστήματος καθώς και τα πιθανά πεδία εφαρμογής αυτού στην πράξη.

Στο δεύτερο κεφάλαιο γίνεται μια εγκυκλοπαιδική ανάλυση των τεχνολογιών που απαιτήθηκαν και εν συνεχεία χρησιμοποιήθηκαν στην κατασκευή του συστήματος, αλλά και τους λόγους τους οποίους επιλέχθηκαν, προσπαθώντας να προετοιμάσει τον αναγνώστη ως προς την κατανόηση της λειτουργίας του.

Στο τρίτο κεφάλαιο παρουσιάζεται ο σκοπός καθώς και η αρχή λειτουργίας του συστήματος. Αναλύεται η δομή του συνολικού συστήματος, περιγράφοντας την αλληλουχία των δύο επιμέρους υποσυστημάτων που το απαρτίζουν και αποτυπώνεται πλήρως η διαμόρφωση των μηχανών του συστήματος, αφορώντας στο πρακτικό και εν συνεχεία στο τεχνικό κομμάτι της εργασίας.

Τέλος, στα τελευταία δύο κεφάλαια, γίνεται αναφορά στα συμπεράσματα που προέκυψαν από τη χρήση του συστήματος καθώς και στη βιβλιογραφία που χρησιμοποιήθηκε και στηρίχθηκε η εργασία.

2. Περιγραφή του Συστήματος

2.1 OpenBSD: Περιγραφή και Ιστορικότητα του Λειτουργικού Συστήματος

Το λειτουργικό σύστημα που χρησιμοποιήθηκε στην ανάπτυξη του project, είναι το OpenBSD. Οι ρίζες του OpenBSD προέρχονται από το λειτουργικό BSD (Berkley Software Distribution) το οποίο είναι ένα σύστημα που διανέμεται από το πανεπιστήμιο Berkley της Καλιφόρνια των ΗΠΑ. Η ανάπτυξη του ξεκίνησε τη δεκαετία του ’70 και σταμάτησε το 1995 στην τελευταία του έκδοση, 4.4BSD.

Το OpenBSD είναι λειτουργικό ανοιχτού κώδικα και ελεύθερα διαθέσιμο προς το κοινό και είναι βασισμένο στο 4.4BSD UNIX. Είναι παγκοσμίως το πιο ασφαλές δικτυακό λειτουργικό σύστημα με ευρεία χρήση του από παρόχους υπηρεσιών διαδικτύου (ISP’s), εταιρίες και κατασκευαστές ενσωματωμένων συστημάτων (Embedded Systems). Επιπλέον είναι διάσημο για τη σταθερότητα που παρουσιάζει σε όλες τις υλοποιήσεις που χρησιμοποιείται, την φορητότητα του μέσα από εικονικά περιβάλλοντα, καθώς και για την αδιαμφισβήτητη ασφάλεια που παρέχει.

2.2 Secure Shell [1](SSH)

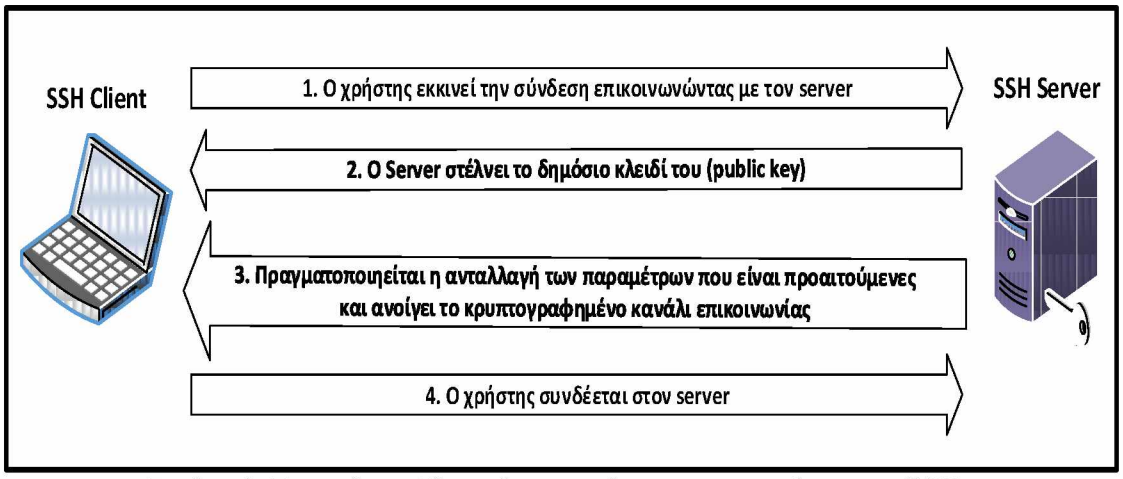
2.2.1 Τι είναι το SSH

Το ασφαλές κέλυφος, Secure Shell ή εν συντομία SSH όπως είναι και ευρέως γνωστό, είναι ένα πολύ ισχυρό και διαδεδομένο πρωτόκολλο, που χρησιμοποιείται για την ασφάλεια κρυπτογραφημένων συνδέσεων δικτύων. Χρησιμοποιεί την αρχιτεκτονική σχεδίασης client/server (πελάτη/εξυπηρετητή) και δημιουργεί ένα κρυπτογραφημένο κανάλι μεταξύ δύο μη έμπιστων εξυπηρετητών (untrusted hosts), με σκοπό να μεταφέρει δεδομένα από έναν σταθμό εργασίας μέσω ενός μη ασφαλούς δικτύου (insecure network) χρησιμοποιώντας μίας αυτόματη διαδικασία κρυπτογράφησης. Όταν τα δεδομένα καταλήξουν στον παραλήπτη στον οποίο προορίζονταν, τότε το SSH αυτόματα πάλι αποκρυπτογραφεί τα δεδομένα. Αυτή η διαδικασία έχει ως αποτέλεσμα, οι χρήστες να μπορούν να εργαστούν και να ανταλλάξουν δεδομένα εφησυχασμένοι ότι οι επικοινωνία τους είναι ασφαλής μέσω του δικτύου που χρησιμοποιούν. Επίσης χρησιμοποιεί όλους τους μοντέρνους ασφαλείς αλγόριθμους κρυπτογράφησης, με αποτέλεσμα να είναι τόσο αποτελεσματική η εφαρμογή του, ακόμη και από μεγάλους οργανισμούς για τις κρίσιμες εφαρμογές που διαθέτουν, είτε δίνοντας τη δυνατότητα μεταφοράς

[1] https://man.openbsd.org/ssh.1

αρχείων ή το κρισιμότερο, να δίνεται η δυνατότητα να εκτελούνται απομακρυσμένα εντολές μεταξύ των hosts.

Ο τρόπος με τον οποίο το πρωτόκολλο του SSH συνδέεται και αποκτά πρόσβαση σε συγκεκριμένο σημείο (destination) στον host, μπορεί να πραγματοποιηθεί με την εντολή `[user@]hostname` ή με τη μορφή URL `ssh://[user@]hostname[:port]`. Εν συνεχεία ο χρήστης πρέπει να πιστοποιήσει την ταυτότητα του στην απομακρυσμένη μηχανή, μέσω της IP διεύθυνσης που διαθέτει. Με αυτό τον τρόπο εξασφαλίζεται η αυθεντικοποίηση του χρήστη.



Εικόνα 1: Τυπική σχεδίαση λειτουργίας του προγράμματος SSH. (Πηγή SSH, The Secure Shell: The Definitive Guide, 2nd Edition)

Στην πραγματικότητα το SSH είναι ένα πρωτόκολλο επικοινωνίας που εξειδικεύεται στο πώς να πραγματοποιούνται ασφαλείς επικοινωνίες μέσω ενός δικτύου που δεν εμπιστευόμαστε (untrusted network).

Το πρωτόκολλο καλύπτει τρείς τομείς που είναι απαραίτητοι για την δημιουργία και πραγματοποίηση μιας ασφαλούς επικοινωνίας:

- **Πιστοποίηση** (Authentication)

Καθορίζει με αξιοπιστία την ταυτότητα οποιουδήποτε προσπαθεί να χρησιμοποιήσει έναν απομακρυσμένο σταθμό εργασίας και καθορίζει μέσω μιας σειράς διαδικασιών εάν όντως έχει την εξουσιοδότηση να συνδεθεί, αλλιώς το SSH απορρίπτει την σύνδεση.

- **Κρυπτογράφηση** (Encryption)

Κρυπτογραφεί τα δεδομένα με τέτοιο τρόπο ώστε να είναι ακατανόητος εκτός από τους προοριζόμενους παραλήπτες, προστατεύοντας τα καθώς ταξιδεύουν μέσω του δικτύου.

- **Ακεραιότητα** (Integrity)

Εγγυάται ότι τα δεδομένα που θα ταξιδέψουν μέσω του δικτύου θα φτάσουν στον παραλήπτη τους αναλλοίωτα. Εάν σε περίπτωση κάποιος τρίτος αποθηκεύσει και μεταβάλει τα δεδομένα κατά τη μεταφορά τους, τότε το SSH θα αναγνωρίσει ότι έχει συμβεί αυτό.

Εν συντομία, το πρωτόκολλο του SSH πραγματοποιεί συνδέσεις δικτύων μεταξύ υπολογιστών, με πολύ ισχυρές εγγυήσεις ότι όσα τερματικά χρησιμοποιούνται και στις δύο άκρες της επικοινωνίας είναι πιστοποιημένα. Επιπλέον εξασφαλίζει ότι το σύνολο των δεδομένων που θα ανταλλάξουν τα τερματικά μεταξύ τους μέσω των συγκεκριμένων συνδέσεων, θα παραληφθεί αναλλοίωτο και δεν θα έχει διαβαστεί από οποιονδήποτε κακόβουλο που προσπάθησε να υποκλέψει την επικοινωνία και τα δεδομένα.

Από τα ανωτέρω προκύπτει ότι, από τα βασικότερα χαρακτηριστικά γνωρίσματα που παρουσιάζει το πρωτόκολλο του SSH, είναι η:

- Πραγματοποίηση ασφαλών απομακρυσμένων συνδέσεων μεταξύ πελάτη/εξυπηρετητή (client/server).

- Ασφαλής μετακίνηση καθώς και η ανταλλαγή δεδομένων ή και ολόκληρων φακέλων με δεδομένα.

- Δυνατότητα που δίνεται σε έναν διαχειριστή συστημάτων να μπορεί να εκτελεί απομακρυσμένα διάφορες εντολές με ασφάλεια.

- Διαδικασία της αυθεντικοποίησης που χρησιμοποιείται από το πρωτόκολλο μέσω του συνδυασμού ιδιωτικού - δημοσίου κλειδιού (private - public key). Μέσω αυτής της διαδικασίας πιστοποιείται ο χρήστης μια φορά και μπορεί να έχει πρόσβαση σε έναν αριθμό από υπολογιστές χωρίς να είναι απαραίτητο να θυμάται κάθε όνομα χρήστη και κωδικό πρόσβασης (username/password) που έχει σε κάθε υπολογιστή ξεχωριστά.

- Δυνατότητα περιορισμού της πρόσβασης σε έναν υπολογιστή. Όταν για παράδειγμα, απαιτείται να δώσουμε πρόσβαση σε ένα άτομο σε έναν υπολογιστή, με περιορισμένα δικαιώματα χρήστη ώστε να μπορεί εκτελέσει μία συγκεκριμένη διεργασία χωρίς να έχει δικαιώματα διαχειριστή ώστε να εκτελέσει οτιδήποτε διαφορετικό.

- Προώθηση θυρών (Port Forwarding ή SSH tunneling) με σκοπό την αύξηση του επιπέδου ασφαλείας διαφόρων εφαρμογών που βασίζονται στο πρωτόκολλο TCP/IP.

Πρακτικά, η προώθηση θυρών επιτρέπει σε πρωτόκολλα επικοινωνίας τα οποία δεν παρέχουν ορισμένο επίπεδο ασφάλειας στα δεδομένα που μεταφέρουν, όπως για παράδειγμα το TCP, UDP, HTTP και SMTP, να

προωθήσουν τη κίνηση των δεδομένων τους μέσω ενός ασφαλούς καναλιού που δημιουργείται μέσω του πρωτοκόλλου SSH.

Με την εφαρμογή κανόνων που υποστηρίζονται από το πρωτόκολλο τα SSH, όπως είναι η κρυπτογράφηση των δεδομένων καθώς και η αυθεντικοποίηση των χρηστών, έχει ως αποτέλεσμα την παροχή αυξημένης ασφάλειας στη διακίνηση των δεδομένων μεταξύ των μηχανών και σε πρωτόκολλα επικοινωνίας που προηγουμένως δεν διέθεταν κανένα χαρακτηριστικό ασφαλείας.

Για τον σκοπό αυτό έχουν αναπτυχθεί διάφορα λογισμικά που ενσωματώνουν την τεχνολογία του πρωτοκόλλου SSH. Η επιλογή όσον αφορά στην υλοποίηση του συστήματος που αναφέρεται στο κεφάλαιο **3. Παρουσίαση του Συστήματος**, πραγματοποιήθηκε με τη χρήση του προγράμματος OpenSSH[2], το οποίο αποτελεί ενσωματωμένο κομμάτι του λειτουργικού συστήματος του OpenBSD.

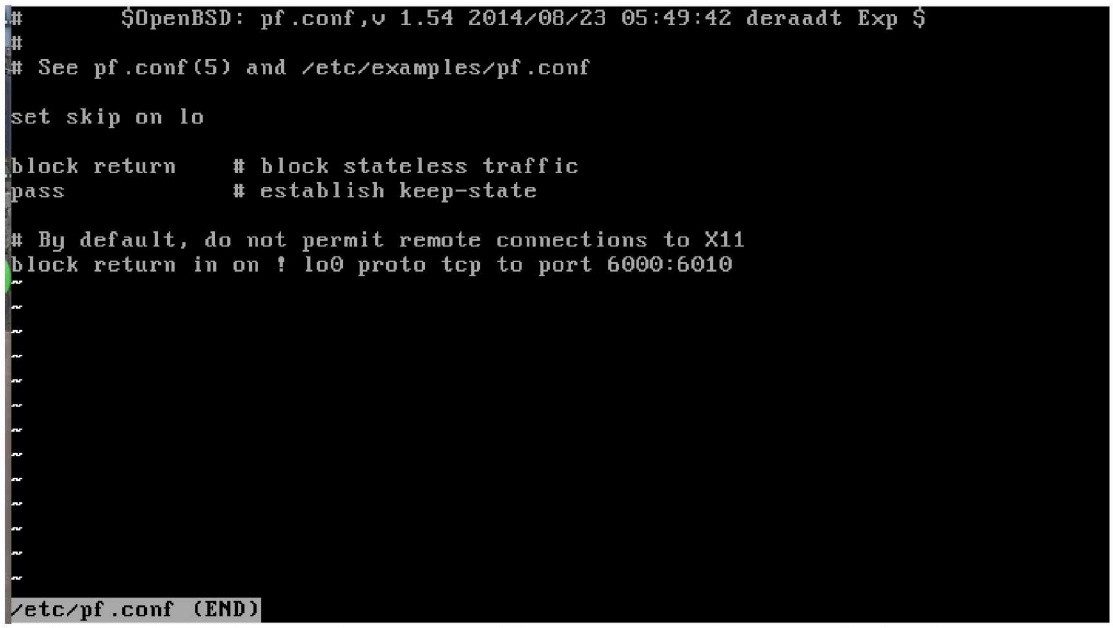
2.3 OpenBSD Packet Filter ή Authenticating Firewall

2.3.1 Packet Filter (PF)

Ξεκινώντας τη διαδικασία κατασκευής του δικτύου που μέσω του οποίου υλοποιήθηκε η διασύνδεση του συστήματος, είναι απαραίτητη η αναφορά στη χρησιμότητα καθώς και στην λειτουργικότητα του firewall που χρησιμοποιήθηκε. Κατά την αρχική εγκατάσταση μίας νέας μηχανής που τρέχει με το λειτουργικό σύστημα του OpenBSD, το πρόγραμμα του PF είναι ενσωματωμένο σε αυτό. Στον φάκελο `/etc/pf.conf` (configuration file) όπου υπάρχουν οι παραμετροποιήσεις του PF, περιέχονται προαποθηκευμένα ορισμένα σχόλια προς βοήθεια για ρυθμίσεις του firewall καθώς και αρχικοί κανόνες filtering.

Ένα πολύ ενδιαφέρον δομικό στοιχείο που χαρακτηρίζει το πρόγραμμα του PF, είναι η μεγάλη ταχύτητα με την οποία μπορεί να διαβάζει της εγγραφές που βρίσκονται καταχωρημένες στους πίνακες του. Αυτό οφείλεται ότι από κατασκευής, κατά την αρχική εκκίνηση του λειτουργικού, οι πίνακες του PF φορτώνονται απευθείας στη μνήμη RAM του υπολογιστή, εκμεταλλευόμενο με αυτό τον τρόπο την μέγιστη ταχύτητα προσπέλασης των δεδομένων.

[2] https://www.openssh.com/openbsd.html



Εικόνα 2: Τυπική απεικόνιση της οθόνης τερματικού στο φάκελο διαμόρφωσης `/etc/pf.conf`. Αναφέρεται η έκδοση του λειτουργικού συστήματος, η ημερομηνία εγκατάστασης, οι αρχικοί κανόνες διακίνησης των πακέτων καθώς και βασικά σχόλια ως προς τις ρυθμίσεις των κανόνων.

Το πρόγραμμα του PF δίνει τη δυνατότητα στον διαχειριστή να καθορίσει ο ίδιος το επίπεδο ασφαλείας κατά την αρχική δημιουργία του firewall, με τους διάφορους κανόνες που θα εφαρμόσει και αφορούν στο φιλτράρισμα της εισερχόμενης ή εξερχόμενης κίνησης, τις συνδέσεις που επιτρέπονται να πραγματοποιούνται καθώς και τα πρωτόκολλα που διακινούνται, ανάλογα με τις απαιτήσεις και τις ανάγκες που έχουν καθοριστεί κατά τη διαδικασία της σχεδίασης του. Οι κανόνες στο φάκελο παραμετροποίησης `/etc/pf.conf`, «φορτώνονται» κατά την αρχική εκκίνηση του λειτουργικού. Η σύνταξη των κανόνων στο PF ακολουθεί ένα συγκεκριμένο τρόπο γραφής. Ωστόσο μια διαδικασία, ένα σύνολο εντολών δηλαδή, μπορεί να περιγραφεί με διαφορετικούς τρόπους γραφής, που όμως να εκτελούν την ίδια διαδικασία. Αυτό είναι ένα πολύ σημαντικό στοιχείο του PF, διότι οι εγγραφές του δύναται να είναι προσαρμοσμένων χαρακτηριστικών (customized) ανάλογα τις απαιτήσεις ή ιδιαιτερότητες ενός δικτύου που έχουν καθοριστεί από τον εκάστοτε διαχειριστή.

2.3.2 authpf

Μια απαραίτητη προϋπόθεση για την σωστή λειτουργία ενός firewall, είναι η ανάγκη αυθεντικοποίησης ενός χρήστη προτού του επιτραπεί η είσοδος σε ένα δίκτυο. Το πρόγραμμα PF εκμεταλλεύεται αυτή την λειτουργικότητα μέσω της εφαρμογής authenticating PF ή κοινώς, **authpf**. Την ύπαρξη αυτής της κρίσιμης λειτουργίας στο πρόγραμμα του PF, την εκμεταλλευτήκαμε κατά τη διάρκεια κατασκευής του συστήματος ώστε σε

συνδυασμό με διάφορα άλλα στοιχεία να διατηρηθεί το επίπεδο ασφαλείας σε υψηλό επίπεδο.

Υλοποιώντας την εφαρμογή του authpf μπορεί να χρησιμοποιηθεί σε διάφορες περιπτώσεις όπου η αυθεντικοποίηση του χρήστη είναι βασικό στοιχείο. Χαρακτηριστικό παράδειγμα χρήσης του authpf είναι κατά την προ-αυθεντικοποίηση χρηστών προκειμένου να τους δοθεί πρόσβαση σε «ευαίσθητα» τμήματα ενός δικτύου, όπως είναι ένα ασύρματο (wifi) τμήμα δικτύου ή όταν πρέπει να δοθεί πρόσβαση σε εργαζόμενους στο δίκτυο της εταιρείας, από το σπίτι τους ή από οπουδήποτε βρίσκονται (αυτοκίνητο, παραλία).

Η εφαρμογή authpf είναι ένα κέλυφος χρήστη που χρησιμοποιείται σε πύλες αυθεντικοποίησης. Μια πύλη αυθεντικοποίησης (authenticating gateway) δεν διαφέρει από μία κανονική πύλη δικτύου, που είναι επίσης γνωστή και ως δρομολογητής (router). Η διαφορά τους είναι ότι για να επιτραπεί η χρήση της ώστε να περάσει η κυκλοφορία του δικτύου, ο χρήστης θα πρέπει να πιστοποιήσει τον εαυτό του πρώτα σε αυτήν. Η πιστοποίηση ενός χρήστη μπορεί να πραγματοποιηθεί είτε μέσω πιστοποιητικών τύπου x.509, είτε σε μορφή αρχείου (μαλακά πιστοποιητικά) είτε μέσω ειδικής συσκευής USB (token), ή πιο απλά με χρήση ονόματος (username) και συνθηματικού (password), ή και σε συνδυασμό των δύο, για τη μεγιστοποίηση της ασφάλειας. Το authpf σαν εργαλείο μπορεί να χρησιμοποιηθεί και από μόνο του ως πύλη, προκειμένου να αυθεντικοποιήσει διάφορους χρήστες ώστε να τους επιτρέψει τη πρόσβαση σε ένα δίκτυο, το οποίο είναι μέρος ενός μεγαλύτερου δικτύου, όπως η πρόσβαση, σε συγκεκριμένες υπηρεσίες μίας εταιρίας ή ενός υπουργείου.

Η σχεδίαση του authpf βασίστηκε σε μία ριζοσπαστική προσέγγιση και ταυτόχρονα σε μία πιο ισχυρή και ευέλικτη διαδικασία που αφορά στην αυθεντικοποίηση των χρηστών ώστε να χρησιμοποιείται μόνο από χρήστες που έχουν τη δυνατότητα να συνδεθούν αποκλειστικά μέσω SSH, σε συνδυασμό με την απαραίτητη ενεργοποίηση των κανόνων στο PF. Η διαδικασία αυτή επιτρέπει την εφαρμογή εξειδικευμένων κανόνων φίλτρου (PF) για κάθε χρήστη χωριστά, οι οποίοι φορτώνονται δυναμικά την ώρα που ο κάθε χρήστης συνδέεται (SSH) επιτυχώς με την εφαρμογή. Οι κανόνες αυτοί παραμένουν σε ισχύ για όσο χρόνο ο χρήστης παραμένει επιτυχώς συνδεδεμένος με την εφαρμογή αυτή.

Μιλώντας σε πιο τεχνικό επίπεδο, για να χρησιμοποιηθεί το πρόγραμμα authpf, αρχικά δημιουργείται ένας χρήστης με τα στοιχεία που θέλουμε σε αυτόν. Για να συνδεθεί ο χρήστης στο δίκτυο, αρχικά συνδέεται μέσω SSH στην πύλη (gateway). Αφού ο χρήστης επιτυχώς ολοκληρώσει την διαδικασία πιστοποίησης του μέσω SSH, το πρόγραμμα authpf φορτώνει και εφαρμόζει δυναμικά τους κανόνες που έχουν προκαθοριστεί (ανά χρήστη),

μέσω του προγράμματος PF. Οι κανόνες αυτοί συνήθως περιέχουν τη διεύθυνση IP προέλευσης του πιστοποιημένου προς σύνδεση χρήστη. Αυτοί οι κανόνες συνήθως έχουν γραφτεί για να ισχύει μόνο η IP διεύθυνση του χρήστη που είναι πιστοποιημένος να συνδεθεί στο δίκτυο. Όταν η συνεδρία μέσω του SSH τερματιστεί, ο χρήστης αποσυνδέεται και οι κανόνες που είχαν φορτωθεί για το συγκεκριμένο προφίλ καταργούνται. Αυτή η διαδικασία μας χρησιμεύει ώστε η κίνηση που δημιουργείται μέσα στο δίκτυο μας, να προκαλείται μόνο από τους πιστοποιημένους μας χρήστες.

Στο σημείο αυτό θέλουμε να προσθέσουμε μια ακόμα διαδικασία πιστοποίησης, η οποία να περιλαμβάνει έναν τρίτο παράγοντα μη απευθείας συσχετιζόμενο με τον client και το server, προκειμένου να ελέγχουμε με πολύ ισχυρότερο τρόπο το ποιός έχει πρόσβαση στην υπηρεσία authpf του τελικού αντικειμενικού σκοπού. Συγκεκριμένα θα χρησιμοποιηθεί μια διαφορετική υπηρεσία, το Σύστημα Ονομάτων Τομέα (DNS - Domain Name System), προκειμένου να προ-αυθεντικοποιήσουμε την IP του χρήστη, προκειμένου αυτός τελικά να αποκτήσει πρόσβαση στο authpf.

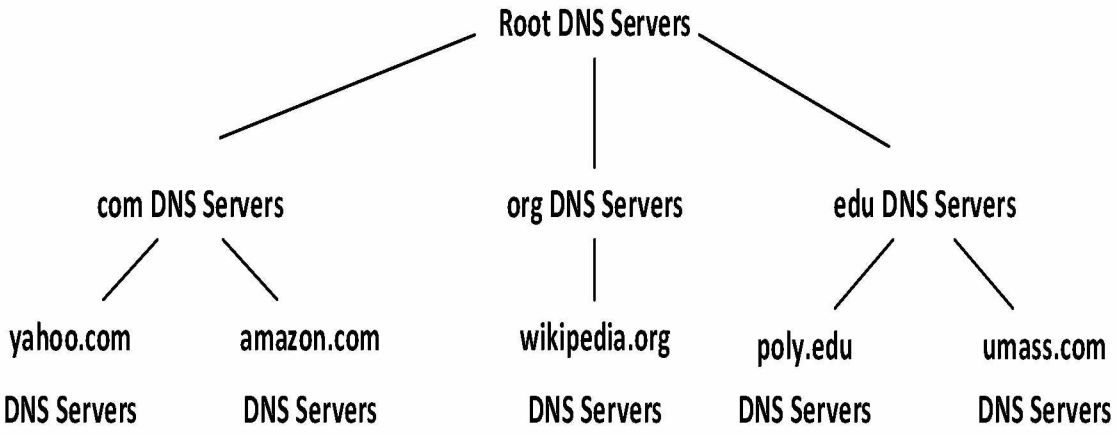
Στο θέμα αυτό θα αναφερθούμε εκτενέστερα και αναλυτικότερα ως προς τη θεωρητική λειτουργία καθώς και στις βασικές έννοιες του DNS στο κεφάλαιο **2.4 Domain Name System (DNS)** και ως προς την πρακτική εφαρμογή και υλοποίηση του, στη παρουσίαση του συνολικού συστήματος στο κεφάλαιο **3. Παρουσίαση του Συστήματος**.

2.4 DNS: Domain Name System

2.4.1 Πώς λειτουργεί το DNS

Το σύστημα ονομάτων τομέα, Domain Name System ή DNS όπως είναι ευρέως γνωστό, είναι μια κατανεμημένη βάση δεδομένων όπου χρησιμοποιείται από τις TCP/IP εφαρμογές (Layer 7 OSI Model) ώστε να χαρτογραφεί τις αντιστοιχίες μεταξύ των ονομάτων του δικτύου (hostnames) με τις IP διευθύνσεις καθώς και να παρέχει στο σύστημα του ηλεκτρονικού ταχυδρομείου (email) τις απαραίτητες πληροφορίες που χρειάζεται ώστε να δρομολογεί τις πληροφορίες τις οποίες διακινεί. Για παράδειγμα όταν ζητηθεί από έναν χρήστη το hostname (ονομασία υπολογιστή) test.somewhere.example, μέσω του προγράμματος-πελάτη DNS, ο χρήστης λαμβάνει την διεύθυνση IP με την οποία είναι καταχωρημένο αυτό το όνομα. Χρησιμοποιείται ο όρος κατανεμημένη διότι σε κανένα κομμάτι της υποδομής του παγκόσμιου ιστού δεν υπάρχει πλήρης η πληροφορία συγκεντρωμένη σε έναν και μόνο διακομιστή. Κάθε ιστότοπος (site) όπως ένα πανεπιστήμιο, μια εταιρεία ή ένα κυβερνητικό τμήμα, συντηρεί τον δικό του διακομιστή DNS (DNS server) όπου και διατηρεί την δική του βάση δεδομένων με τα στοιχεία

που αφορούν το τμήμα του ή την εταιρεία του, οι οποίες όμως είναι διαθέσιμες προς τους διάφορους χρήστες μέσω του διαδικτύου (Internet) που θα θελήσουν πραγματοποιήσουν τα διάφορα ερωτήματα τους σε αυτή. Εν ολίγοις το DNS παρέχει ένα πρωτόκολλο που επιτρέπει στους χρήστες καθώς και στους διακομιστές να επικοινωνεί ο ένας με τον άλλο.



Εικόνα 3: Παράδειγμα της ιεραρχικής δομής της κατανεμημένης βάσης δεδομένων στο Σύστημα του DNS (Πηγή internet)

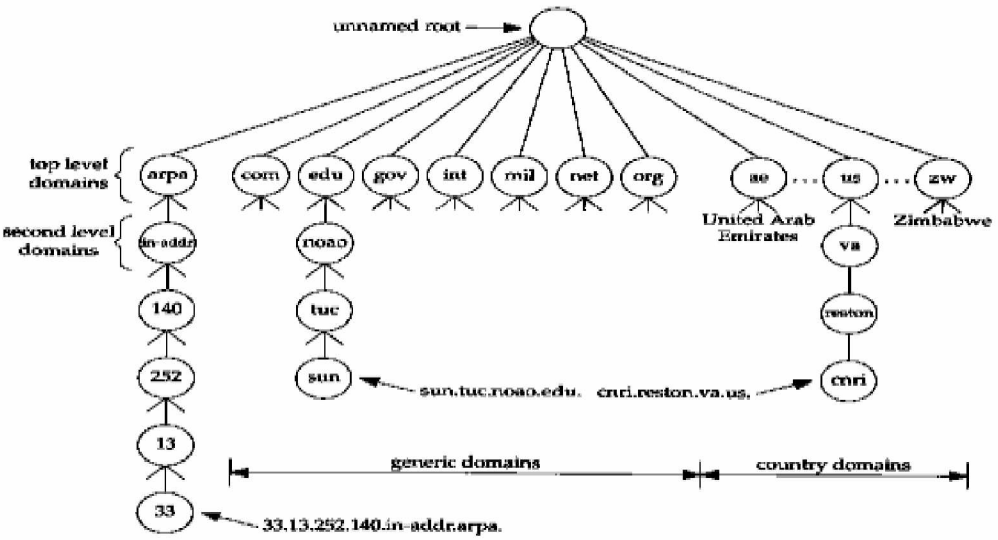
Ο λόγος για τον οποίο δεν υπάρχει μία βάση δεδομένων που είναι συγκεντρωμένες όλες οι καταχωρήσεις των ονομάτων τομέα σε ένα διακομιστή είναι διότι θα υπήρχε τεράστιο πρόβλημα λόγω της κίνησης που θα δημιουργούταν όταν όλοι οι χρήστες παγκοσμίως θα ρωτούσαν έναν συγκεκριμένο διακομιστή και αυτός θα έπρεπε να απαντήσει σε όλα τα ερωτήματα συγχρόνως, η συντήρηση θα ήταν εξαιρετικά δύσκολη λόγω της πολυπλοκότητας και του όγκου της και τέλος θα υπήρχε ένα και μόνο σημείο κατάρρευσης με συνέπεια την ολική κατάρρευση του Internet.

Η πρόσβαση στο DNS (σύστημα των ονομάτων τομέα), πραγματοποιείται μέσω της διαδικασίας επίλυσης (resolver). Πιο συγκεκριμένα, σε έναν εξυπηρετητή (host) και συγκεκριμένα στο λειτουργικό OpenBSD, η διαδικασία επίλυσης αποκτά πρόσβαση πρώτα μέσω δύο λειτουργιών, που αναφέρονται ως `gethostbyname`[3] και `gethostbyaddr`. Η πρώτη λαμβάνει το όνομα του εξυπηρετητή (host name) και σε πιθανό ερώτημα επιστρέφει την IP διεύθυνση που της έχει οριστεί και η δεύτερη λαμβάνει την IP διεύθυνση και ψάχνει πληροφορίες σχετικά με όνομα του εξυπηρετητή (host name) που αντιστοιχεί σε αυτή. Τελικά ο επιλυτής επικοινωνεί με όσους name servers απαιτείται ώστε να χαρτογραφήσει την απάντηση. Η διαδικασία χαρτογράφησης ονομάζεται διαδικασία επίλυσης ονόματος - διεύθυνσης (name-address resolution).

[3] https://man.openbsd.org/OpenBSD-5.5/gethostbyname.3

2.4.2 Βασικά Στοιχεία

Ολόκληρη η δομή του συστήματος του DNS είναι κατασκευασμένη όπως ακριβώς η ιεραρχική δομή του συστήματος αρχείων (filesystem) που συναντάμε στα λειτουργικά συστήματα Unix. Στην παρακάτω εικόνα (εικόνα 2.4.2) παρουσιάζεται η βασική ιεράρχηση του συστήματος DNS.



Εικόνα 4: Αποτύπωση της ιεραρχικής δομής του συστήματος του DNS (Πηγή: TCP/IP Illustrated Volume 1, W. Richard Stevens)

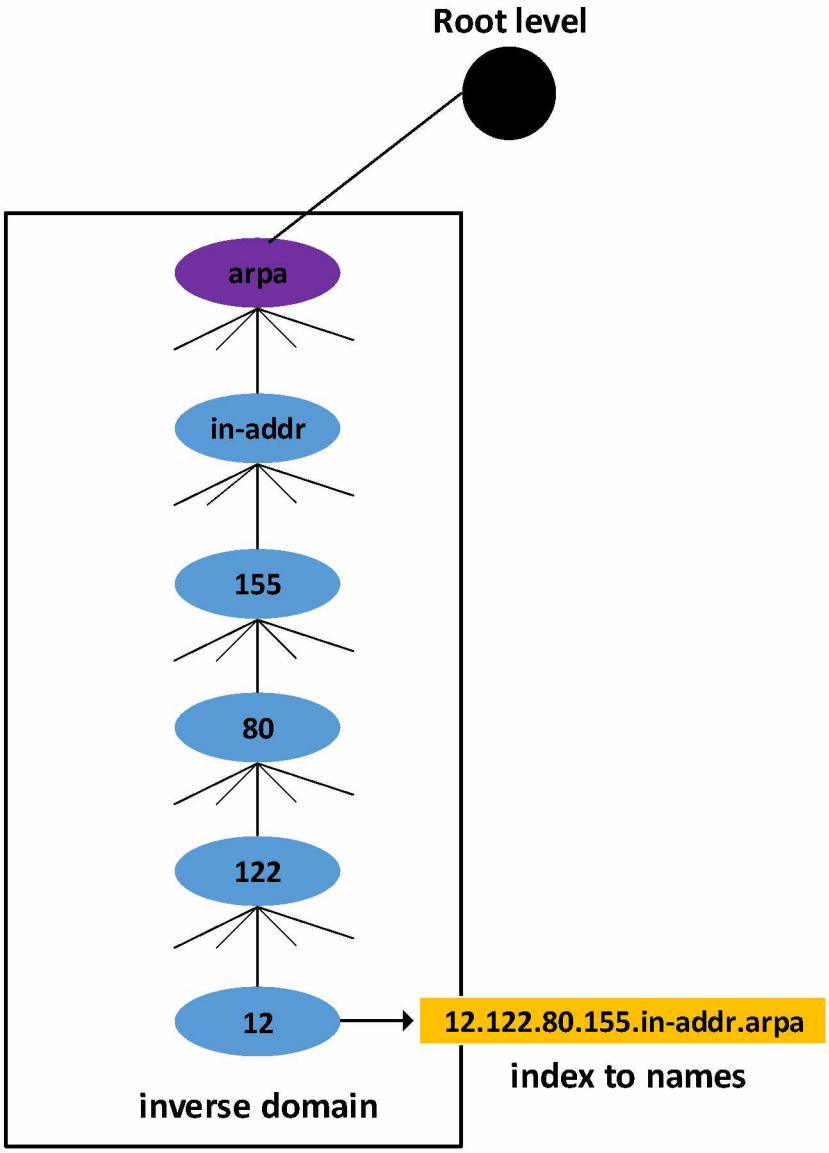
Οι σχεδιασμένοι κύκλοι στην εικόνα 2.4.2, χαρακτηρίζονται ως κόμβοι και ο καθένας από αυτούς μπορεί να λάβει μία ετικέτα (label) που να τον περιγράφει, μέχρι 63 χαρακτήρες. Ο βασικός κόμβος που ονομάζεται root, είναι ένας ειδικός κόμβος ο οποίος δεν έχει ονομασία (null label). Μια κάθετη σειρά από κόμβους, οι οποίοι χωρίζονται με τελείες μεταξύ τους και σχεδιάζουν μια λίστα, ξεκινώντας από τον τελευταίο κόμβο και συνεχίζοντας μέχρι τον αρχικό κόμβο όπου και βρίσκεται ο ριζικός κόμβος (root node), δημιουργούν ένα όνομα τομέα (domain name), όπως για παράδειγμα είναι το test.somewhere.com. Στην δενδρική δομή του συστήματος DNS, σε κάθε κλάδο (branch), ο κάθε κόμβος πρέπει να έχει ένα μοναδικό όνομα τομέα (domain name), όμως η κάθε ετικέτα ξεχωριστά, μπορεί να χρησιμοποιηθεί και σε διαφορετικά σημεία του δέντρου. Ένα όνομα τομέα, που τελειώνει με τελεία στο τέλος του, ονομάζεται πλήρως προσδιορισμένο όνομα τομέα (absolute domain name) ή (fully qualified domain name or FQDN). Το FQDN είναι ένα απόλυτο όνομα που καθορίζει τη θέση του σε σχέση με την απόλυτη ρίζα του συστήματος ονομάτων τομέα. Για παράδειγμα στο όνομα τομέα που δώσαμε πριν, αν προσθέσουμε την τελεία στο τέλος του, αυτόματα δημιουργούμε ένα πλήρως πιστοποιημένο όνομα τομέα (FQDN).

Αναλύοντας την ιεραρχική δομή, μετά τον ειδικό κόμβο που είναι η ρίζα της δομής όπως παραπάνω, συναντάμε τα ανώτατα επίπεδα των τομέων

(top-level domains ή TLD's). Όπως φαίνεται και στο **Σχεδιάγραμμα 2.4.1**, οι ανώτατοι τομείς αναλύονται σε υποτομείς (second level domains ή sub domains). Αυτό σημαίνει ότι κάθε υποτομέας είναι μέρος ενός μεγαλύτερου τομέα, με εξαίρεση την αρχική ρίζα της δομής (root domain). Για παράδειγμα το όνομα εξυπηρετητή (host) test.somewhere.com, είναι υποτομέας του somewhere.com, ο οποίος τομέας είναι υποτομέας του top-level domain .com.

Τα ανώτατα επίπεδα των τομέων χωρίζονται σε τρία πεδία όπως παρακάτω:

- Ο τομέας arpa, είναι ένας ειδικός τομέας που χρησιμοποιείται για την χαρτογράφηση των διευθύνσεων με τα αντίστοιχα ονόματα τομέων και χρησιμοποιείται αποκλειστικά για τεχνικούς σκοπούς που αφορά στην υποδομή (infrastructure) του συστήματος του DNS. Αρχικά το όνομα arpa χρησιμοποιούταν από τον προκάτοχο του Internet, το ARPANET. Με την καθιέρωση όμως του συστήματος του DNS το έτος 1985, ο συγκεκριμένος τομέας χρησιμοποιήθηκε με σκοπό την μετάβαση των ονομάτων και γενικά των εγγραφών που υπήρχαν στο σύστημα του ARPANET, στο σύστημα του DNS. Αφού ο τομέας arpa εκπλήρωσε το σκοπό που του είχε αρχικά ανατεθεί, τελικά η διαγραφή του τομέα arpa αποδείχθηκε μη εφαρμόσιμη διότι ο δεύτερου επιπέδου (second level domain) τομέας in-addr.arpa χρησιμοποιούταν για την πολύ σημαντική διαδικασία της αντίστροφης επίλυσης ονομάτων τομέα (reverse DNS lookup). Η διαδικασία της αντίστροφης επίλυσης ονομάτων τομέα, είναι ερωτήματα που πραγματοποιούνται στο σύστημα του DNS, για να καθοριστεί ένα όνομα τομέα σε ποια IP διεύθυνση αντιστοιχεί. Για κάθε όνομα τομέα που εισάγεται στο σύστημα του DNS, αντίστοιχα, αποκτάται η εξουσιοδότηση για την εγγραφή του στο πεδίο ονομάτων του τομέα in-addr.arpa, σύμφωνα πάντα με την αντιστοιχία της IP διεύθυνσης του που αντιστοιχεί στο συγκεκριμένο όνομα τομέα. Για παράδειγμα, αν για τον εξυπηρετητή (host) somewhere, που αναφέραμε πριν, η διεύθυνση IP που αντιστοιχεί είναι 155.80.122.12, η εγγραφή του θα είναι 12.122.80.155.in-addr.arpa.



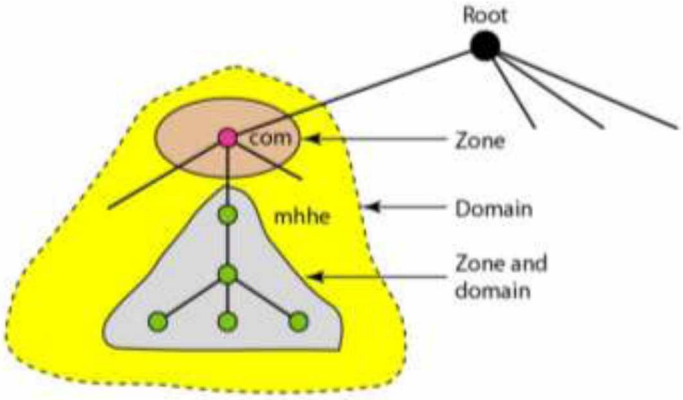
Εικόνα 5: Σχεδιάγραμμα λειτουργίας του τομέα arpa (Πηγή internet)

- Οι τομείς, που αναφέρονται ως γενικοί τομείς (generic domains) ή ως οργανωτικοί τομείς (organizational domains). Εδώ για παράδειγμα, ανήκουν οι τομείς που έχουν κατάληξη .com, .edu, .gov, .int, .mil , net, .org, .info και αναφέρονται σε εμπορικούς οργανισμούς, εκπαιδευτικά ιδρύματα, κυβερνητικές οργανώσεις, στρατό, δίκτυα, διάφορους οργανισμούς καθώς και σε παρόχους υπηρεσιών πληροφόρησης.

- Όλοι οι τομείς που περιγράφουν ονόματα χωρών και αναφέρονται ως τομείς χωρών (country domains) ή ως γεωγραφικοί τομείς (geographical domains), όπως για παράδειγμα οι τομείς .gr που αναφέρεται στην χώρα Ελλάδα και .eu που αφορά στην Ευρωπαϊκή Ένωση.

Μια ακόμη πολύ σημαντική λειτουργία που πραγματοποιείται εντός του συστήματος του DNS, αφορά στην ανάθεση αρμοδιοτήτων (delegation of

responsibility) σε ζώνες (DNS zones). Μια ζώνη DNS είναι ένα κομμάτι της δομής του συστήματος ονοματοθεσίας του DNS, το οποίο διαχειρίζεται και συντηρείται από μια συγκεκριμένη εταιρεία ή έναν οργανισμό, γενικότερα. Μία ζώνη μπορεί να περιέχει πολλαπλούς τομείς, όπως αναλυτικά παρουσιάζεται στο παρακάτω σχεδιάγραμμα επεξήγησης που περιέχει τους τομείς, υποτομείς καθώς και τις ζώνες του συστήματος του DNS.



Εικόνα 6: Σχεδιάγραμμα επεξήγησης μεταξύ τομέων, υποτομέων και ζωνών του συστήματος του DNS (Πηγή Internet)

2.4.3 Δεδομένα που περιέχονται στο σύστημα του DNS

Η επικοινωνία μεταξύ των εξυπηρετητών, που περιέχουν τις πληροφορίες για τις εγγραφές των διαφόρων τομέων που διαχειρίζονται, πραγματοποιείται μέσω μηνυμάτων του συστήματος του DNS (*DNS Message)* που ονομάζονται *DNS Query Message* για την ερώτηση και *DNS Response Message* για την απάντηση. Το πρωτόκολλο του DNS χρησιμοποιεί κοινή μορφή (format) για τα μηνύματα που ανταλλάσσονται μεταξύ των εξυπηρετητών και των χρηστών. Τα μηνύματα αυτά περιέχονται μέσω των πρωτοκόλλων UDP (User Datagram Protocol) ή TCP (Transmission Control Protocol) χρησιμοποιώντας εξ ορισμού την πόρτα 53. Τα μηνύματα που έχουν μέγεθος μικρότερο ή ίσο με 512 bytes χρησιμοποιούν το πρωτόκολλο UDP, ενώ για μεγαλύτερου μεγέθους μηνυμάτων χρησιμοποιείται το πρωτόκολλο TCP. Κάθε ερώτηση περιέχει στοιχεία με διάφορες πληροφορίες που ονομάζονται είδη ερωτημάτων (query types) και οι απαντήσεις τους αντίστοιχα είδη (types), οι οποίες απαντήσεις ονομάζονται επίσης και ως εγγραφές πόρων (Resource Records ή RRs). Αυτή η πληροφορία μπορεί να αποθηκευτεί ως προσωρινή καταχώρηση στη κρυφή μνήμη (cache) των διακομιστών DNS για μεγαλύτερη ταχύτητα μεταξύ των μεταφράσεων. Μερικές από τις τιμές (values) που περιέχονται στα δύο αυτά συγκεκριμένα είδη, αναλύονται ως εξής:

Όνομα	Αριθμητική τιμή	Περιγραφή
A	1	IP address
NS	2	name server record
CNAME	5	canonical name
PTR	12	pointer record
MX	15	mail exchange record

Πίνακας 1: Ανάλυση τιμών που περιέχονται στα ερωτήματα και απαντήσεις του συστήματος του DNS.

Τα πεδία του ανωτέρω πίνακα αντιστοιχούν στα παρακάτω:

type=A, όνομα (NAME) είναι η ονομασία του υπολογιστή (hostname) και τιμή (VALUE) του είναι η διεύθυνση IP στην οποία αντιστοιχεί.

type=NS, όνομα (NAME) είναι το όνομα τομέα και τιμή (VALUE) του είναι η διεύθυνση IP ενός επίσημου εξυπηρετητή όπου ανήκει το όνομα του υπολογιστή (hostname).

type=CNAME, όνομα (NAME) είναι ένα ψευδώνυμο (alias) για τον εξυπηρετητή (host) και τιμή (VALUE) του είναι το κανονικό όνομα (canonical name) του εξυπηρετητή (host).

type=MX, όνομα (NAME) είναι ένα ψευδώνυμο για το ηλεκτρονικό ταχυδρομείο του εξυπηρετητή (host) και τιμή (VALUE) του είναι το κανονικό όνομα (canonical name) του εξυπηρετητή του ηλεκτρονικού ταχυδρομείου (email server).

Επιπλέον, στα εγγραφές πόρων (Resource Record), εμπεριέχεται και η τιμή του **χρόνου ζωής** (time-to-live[4] ή TTL), που αναφέρεται στον χρόνο που μπορεί να παραμείνει ένα Resource Record μέχρι να ενημερωθεί εκ νέου στον εξυπηρετητή προσωρινής αποθήκευσης ερωταπαντήσεων (caching DNS server). Αυτός ο χρόνος καθορίζεται από τις ρυθμίσεις που έχει επιλέξει ο διαχειριστής του συγκεκριμένου διακομιστή και συνήθως ορίζεται στις δύο μέρες.

2.4.4 Κύριες Λειτουργίες των Διακομιστών DNS

Για να αντιληφθούμε την σχεδίαση του συστήματος όσον αφόρα στο κομμάτι που χρησιμοποιεί το σύστημα του DNS, πρέπει να αναφέρουμε τις

[4] https://en.wikipedia.org/wiki/Time_to_live

λειτουργίες που είναι διαθέσιμες μέσω των μοντέρνων διακομιστών DNS. Οι παρακάτω λειτουργίες μπορεί να χρησιμοποιηθούν μεμονωμένα ή και να συνδυαστικά κατά την υλοποίηση ενός διακομιστή DNS.

- **Recursive DNS server** (Διακομιστής διαδοχικής λειτουργίας).
- **Authoritative DNS server** (Υπεύθυνος διακομιστής για συγκεκριμένο τομέα).
- **Caching DNS server** (Διακομιστής αποθήκευσης απαντήσεων σε ερωτήματα DNS).

2.4.5 Unbound (Λογισμικό DNS server): Ένας Επιλυτής Ονομάτων Τομέα

Το λογισμικό του Unbound αναπτύχθηκε στη σημερινή του μορφή το 2006, χρησιμοποιώντας ως βάση τη γλώσσα προγραμματισμού C. Είναι σχεδιασμένο ώστε να επιτύχει τη μέγιστη απόδοση, δημιουργώντας έναν «ελαφρύ», ταχύτατο και ευέλικτο DNS server, ενσωματώνοντας όλα τα μοντέρνα χαρακτηριστικά γνωρίσματα που αφορούν στην ασφάλεια, βασιζόμενο στις προδιαγραφές για πλήρη συμβατότητα όπως καθορίζεται σύμφωνα με τα «ανοιχτά» πρότυπα (Open Standards). Διατίθεται ελεύθερα ως ανοιχτού κώδικα λογισμικό, από την εταιρεία NLnet Labs[5], υπό την άδεια του BSD (BSD license).

Κατά τη φάση της σχεδίασης του Συστήματος Αυθεντικοποίησης επιλέχθηκε ο συγκεκριμένος επιλυτής ονομάτων τομέα, διότι παρουσιάζει στα χαρακτηριστικά του γνωρίσματα, τις ιδιότητες του validating, recursive, and caching DNS resolver και παρέχει την κρίσιμη για το σύστημα δυνατότητα του απομακρυσμένου ελέγχου (remote control).

2.4.6. DDNS: Dynamic Domain Name System

Το δυναμικό σύστημα ονομάτων τομέα, ή πιο συγκεκριμένα Dynamic DNS ή DDNS, είναι μια υπηρεσία στην οποία διατηρούνται και συγχρόνως επιλύονται τα ονόματα τομέα σε δυναμικές διευθύνσεις IP. Σε αντίθεση με το σύστημα ονομάτων τομέα το οποίο λειτουργεί απαραίτητα με στατικές διευθύνσεις IP για τα ονόματα τομέα που διατηρούνται στη βάση δεδομένων από τους διακομιστές, το δυναμικό σύστημα ονομάτων τομέα, λειτουργεί με δυναμικές διευθύνσεις IP δηλαδή με διευθύνσεις που αλλάζουν συχνά. Αυτό υλοποιείται πρακτικά με χρήση πολύ μικρού χρόνου ζωής (TTL) της ζώνης

[5] https://nlnetlabs.nl/projects/unbound/about/

που προσδιορίζει το μέγιστο χρόνο αποθήκευσης της πληροφορίας στην προσωρινή μνήμη (cache) ενός διακομιστή DNS. Η μέθοδος του DDNS πρακτικά είναι μια αυτοματοποιημένη διαδικασία μέσω της οποίας ανανεώνονται οι εγγραφές που διατηρεί ένας διακομιστής ονομάτων τομέα (name server), σχετικά με τα hostnames και τις ip διευθύνσεις που αντιστοιχούν, χωρίς να απαιτείται η παρέμβαση κάποιου διαχειριστή.

Η συγκεκριμένη διαδικασία, διευκολύνει τη σχεδίαση συστήματος με το οποίο μπορεί οποιοσδήποτε να έχει πρόσβαση από οπουδήποτε βρίσκεται ως φυσική παρουσία σε ένα απομακρυσμένο τερματικό, όπως για παράδειγμα, όπως για παράδειγμα απαιτείται κατά τη διάρκεια του ταξιδιού ενός ποντοπόρου πλοίου.

Οι δυνατότητες του δυναμικού συστήματος ονομάτων τομέα, θα χρησιμοποιηθούν αργότερα κατά την ανάπτυξη του συστήματος, ως κρίσιμο μέρος, που βοηθά στην απλούστευση της διαδικασίας σύνδεσης μεταξύ πελάτη και εξυπηρετητή (client – server).

2.6 Τεχνολογία εικονικοποίησης (Virtualization Technology)

Όπως διαπιστώνεται, το συνολικό σύστημα για να μπορέσει να εκμεταλλευτεί τον συνδυασμό των τεχνολογιών που το απαρτίζουν αλλά και για να διέπεται από τα χαρακτηριστικά που του έχουμε προσδώσει κατά τον σχεδιασμό του (αναφέρονται αναλυτικά στο **Κεφάλαιο 3. Παρουσίαση του Συστήματος)**, μονόδρομος είναι η εφαρμογή της τεχνολογίας εικονικοποίησης (Virtualization Technology).

Οι δυνατότητες που προσδίδει η τεχνολογία εικονικοποίησης καθώς και αυτής των εικονικών μηχανών (virtual machines) στον τομέα της πληροφορικής είναι τεράστιες και το αντικείμενο εφαρμογής τους είναι ακριβώς αυτό που αναζητήθηκε προκειμένου να υλοποιηθούν στο έπακρο οι απαιτήσεις που καθορίστηκαν για την κατασκευή του συνολικού συστήματος.

Οι βασικές ιδιότητες των τεχνολογιών εικονικοποίησης και εικονικών μηχανών είναι:

Της τεχνολογίας εικονικοποίησης:

- Μειωμένο κόστος που αφορά στην λειτουργία καθώς και στην υλοποίηση.

- Ελαχιστοποιημένος χρόνος με το σύστημα να βρίσκεται εκτός λειτουργίας (downtime).

- Αυξημένη παραγωγικότητα στο τομέα του IT, που συνδυάζει αποτελεσματικότητα, ευελιξία καθώς και ταχύτητα απόκρισης.

- Ταχύτερα συστήματα παροχής εφαρμογών και πόρων.

- Μέγιστη συνέχεια της επιχείρησης καθώς και δυνατότητα αποκατάστασης σε περίπτωση καταστροφής (disaster recovery).

Των εικονικών μηχανών:

- Η δυνατότητα διχοτόμησης (partitioning), κατά την οποία σε μια φυσική μηχανή μπορεί να λειτουργήσει πλήθος εικονικών μηχανών και να μοιράζονται ταυτοχρόνως τους φυσικούς πόρους της μηχανής, όπως τη μνήμη RAM, τους σκληρούς δίσκους και τις κάρτες δικτύου.

- Η ικανότητα απομόνωσης (isolation) οποιουδήποτε προβλήματος ή θέματος ασφαλείας στο επίπεδο του hardware καθώς και της διατήρησης της απόδοσης μέσω προηγμένων συστημάτων ελέγχου διαχείρισης των πόρων του συστήματος.

- Η ενθυλάκωση (encapsulation), όπου δίνει τη δυνατότητα στον διαχειριστή να αποθηκεύει τη πλήρη κατάσταση μιας εικονικής μηχανής σε φακέλους και κατόπιν να πραγματοποιεί τη μεταφορά και αντιγραφή της εικονικής μηχανής με απόλυτη ευκολία.

- Η απόλυτη ανεξαρτητοποίηση από τις διάφορες απαιτήσεις του hardware και η ικανότητα μετακίνησης μίας εικονικής μηχανής σε οποιονδήποτε φυσική μηχανή χρειάζεται.

3. Παρουσίαση του Συστήματος

Ως αρχική προϋπόθεση που τέθηκε κατά τη φάση της σχεδίασης του συνολικού συστήματος, ήταν η εφαρμογή τεσσάρων βασικών αρχών προκειμένου να είναι απόλυτα εφαρμόσιμο υπό οποιεσδήποτε συνθήκες κληθεί να αντιμετωπίσει, με κύρια έμφαση στην δυναμικότητα, την απόκριση και απόκρυψη λόγω της μεταβολής της θέσης του πελάτη (client) και της μεταβολής των δικτύων, όπως είναι τα δορυφορικά και τα Wi-Fi δίκτυα.

Οι τέσσερις βασικές αρχές είναι:

- **Ασφάλεια Δεδομένων** (Data Integrity). Η διασφάλιση της ανεμπόδιστης διακίνησης των δεδομένων, η προστασία από οποιονδήποτε κακόβουλο καθώς και η αξιοπιστία του συνολικού δικτύου καθώς κανένα φυσικό πρόσωπο δεν μπορεί να παρεμβληθεί εντός. Επιπλέον, ως σημαντικό στοιχείο αξίζει να αναφερθεί ότι, δεν υπάρχει διαχειριστής εμπλεκόμενος στην

διαδικασία, καθιστώντας το σύστημα μη αντιληπτό και αόρατο (stealth) στο σύνολό του.

- **Προσαρμοστικότητα** (Adaptability). Ουδετερότητα από το περιβάλλον το οποίο θα υποστηρίξει.

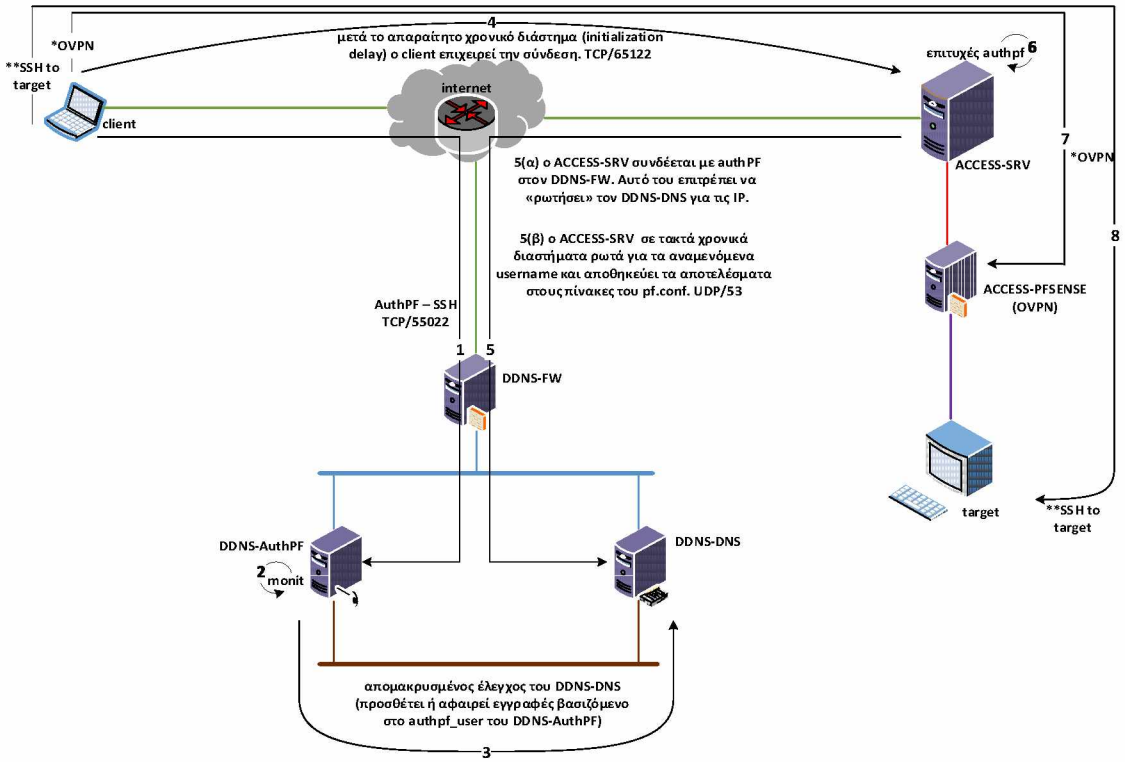
- **Μεταφερσιμότητα** (Portability). Η ικανότητα του συστήματος να μπορεί να μεταφερθεί φυσικά σε οποιοδήποτε χώρο ή τόπο απαιτηθεί.

- **Χαμηλό Κόστος ως προς την Αποτελεσματικότητα** (Cost Effective). Το κόστος κατασκευής του συνολικού συστήματος είναι το μικρότερο δυνατό, παρέχοντας υψηλή αποτελεσματικότητα στον τομέα της ασφάλειας των δεδομένων.

3.1 Σκοπός

Η απομακρυσμένη ασφαλής πρόσβαση ενός client σε έναν server, με τη χρήση ενδιάμεσου συστήματος ιδιωτικού δυναμικού DNS, εξασφαλίζοντας μηδενική αρχική ανταλλαγή δεδομένων ασφάλειας μεταξύ client και server.

3.2 Αρχή Λειτουργίας



Εικόνα 7: Αρχή λειτουργίας του Συνολικού Συστήματος

Η αρχή λειτουργίας του συνολικού συστήματος, όπως παρουσιάζεται στην εικόνα 7, στηρίζεται στην υλοποίηση ενός ενδιάμεσου συστήματος που μέσω συγκεκριμένων διαδικασιών, αυθεντικοποιεί έναν χρήστη για την πρόσβαση σε έναν server, με την απαραίτητη προϋπόθεση ο χρήστης να μπορεί να αποδείξει ότι έχει το δικαίωμα για την υλοποίηση της σύνδεσης του στο server μέσω ορισμένων πολύ συγκεκριμένων βημάτων που καταλήγουν στην αυθεντικοποίηση του.

Ειδικότερα, η ολοκλήρωση του κύκλου των διαδικασιών, στηρίζεται σε μια αυστηρή αλληλουχία οχτώ θεμελιωδών βημάτων. Προϋπόθεση για την επιτυχή έκβαση της αλληλουχίας είναι η «προσκόμιση» των απαραίτητων ψηφιακών πιστοποιητικών.

Παρακάτω, περιγράφονται εν συντομία τα οχτώ βήματα που ακολουθούνται, η ανάλυση των οποίων θα οδηγήσει αργότερα στην λεπτομερέστερη κατανόηση του συστήματος:

Βήμα 1 Ο client συνδέεται μέσω του DDNS-FW στον DDNS-AuthPF στην κατάλληλη πόρτα TCP χρησιμοποιώντας το κατάλληλο ψηφιακό πιστοποιητικό (RSA). Η σύνδεση αυτή θα πρέπει να διατηρηθεί ανοιχτή καθόλη τη διάρκεια της διαδικασίας (SSH session).

Βήμα 2 Εσωτερική διεργασία στον DDNS-AuthPF: Το πρόγραμμα Monit παρακολουθεί τα log file του authpf από όπου εξάγει την πληροφορία όνομα χρήστη/IP διεύθυνση (username/IP) που θα χρησιμοποιηθεί στο βήμα 3.

Βήμα 3 Καταχώρηση στις εγγραφές του DNS η εγγραφή (DNS record) σχετιζόμενη με το συνδυασμό IP και username. Επίσης τρέχουν διαδικασίες εσωτερικού ελέγχου για την ύπαρξη διπλών εγγραφών στις εγγραφές του DNS.

Βήμα 4 Μεταξύ του βήματος 3 και 5, είναι απαραίτητη μια χρονική καθυστέρηση, η οποία εξαρτάται από τη συχνότητα με την οποία ο ACCESS-SERVER ρωτά τον DDNS-DNS. Με τη λήξη της καθυστέρησης ο client επιχειρεί τη σύνδεση με τον server.

Βήμα 5 Ο ACCESS-SERVER σε τακτά χρονικά διαστήματα ρωτά για τα αναμενόμενα usernames και τις IP διευθύνσεις από τις οποίες επιτρέπεται να προέλθουν και αποθηκεύει τα αποτελέσματα στους πίνακες του pf.

Βήμα 6 Αν η IP του client περιέχεται στους πίνακες επιτρέπεται η πρόσβαση στο authpf port. (ο επιτυχής συνδυασμός του 4 και 5 προκαλεί το βήμα 6 Authpf authentication με τα ίδια ή και διαφορετικά πιστοποιητικά με σκοπό να ανοίξει η πόρτα για το OpenVPN (Για λόγους ασφαλείας τα πιστοποιητικά είναι αποθηκευμένα σε usb token για να γίνεται αυθεντικοποίηση και του client και του χρήστη)

Βήμα 7 Με το βήμα αυτό, ο client εκκινεί τη σύνδεση με τον server στην πόρτα και το πρωτόκολλο που προβλέπουν οι ρυθμίσεις του λογισμικού OpenVPN. Με την επιτυχή κατάληξη, ο client έχει όλες τις απαραίτητες πληροφορίες δρομολόγησης ώστε να περάσει την κίνηση που επιθυμεί με το δίκτυο-αντικειμενικό σκοπό. Η κίνηση αυτή είναι κρυπτασφαλισμένη, με αλγόρίθμους που αποφασίζονται κατά τη ρύθμιση του λογισμικού OpenVPN.

Σημειώνεται ότι όλα τα authpf sessions πρέπει να μένουν ανοικτά για είναι δυνατή η επικοινωνία. Επιπλέον οι διαδικασίες της ανανέωσης ή «καθαρισμού» της βάσης δεδομένων του DNS τρέχουν συνεχώς.

Βήμα 8 Στο βήμα αυτό, έχει ολοκληρωθεί επιτυχώς η σύνδεση μεταξύ του client και του δικτύου το οποίο προστατεύει ο ACCESS-SERVER, οπότε ο client προχωρεί στην τελική σύνδεση (συνήθως ssh ή rdp ή vnc κλπ) με κάποια από τις διευθύνσεις IP του προστατευόμενου δικτύου.

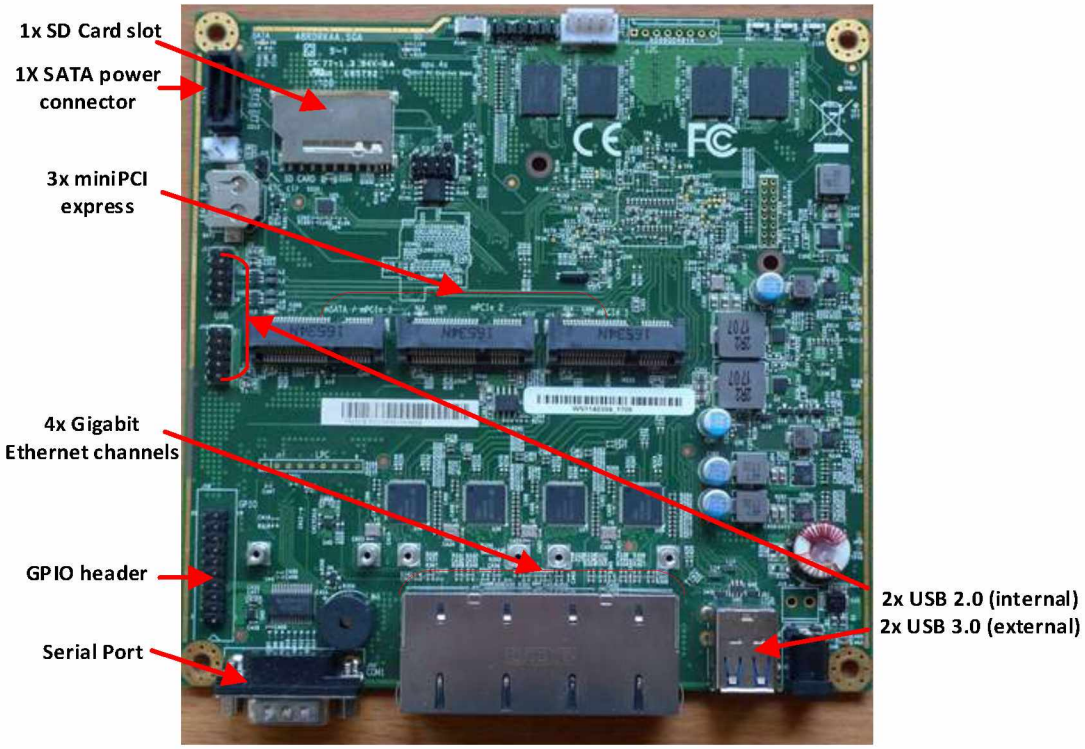
3.3 Δομή του Συστήματος

3.3.1 Περιγραφή

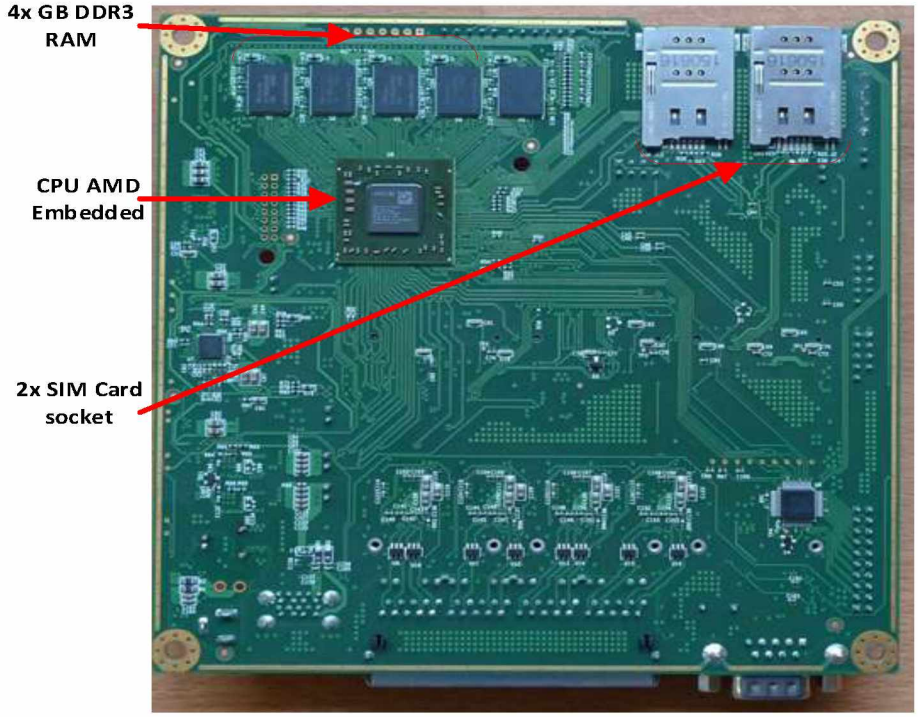
Το συνολικό σύστημα αποτελείται από δύο επί μέρους συστήματα, το **Σύστημα Αυθεντικοποίησης DNS** και το **Σύστημα Πρόσβασης**, αλληλοεπιδρώντας μόνο όταν συγκεκριμένες συνθήκες το επιτρέψουν. Ο συγκεκριμένος διαχωρισμός έγινε εσκεμμένα ώστε να δώσει τη δυνατότητα στο συνολικό σύστημα να υιοθετήσει δύο πολύ βασικά χαρακτηριστικά, που σύμφωνα με αυτά δημιουργήθηκε. Το πρώτο είναι ότι τα δύο επί μέρους συστήματα δεν παρουσιάζουν καμία σχέση μεταξύ τους, φαίνοντας πως ενεργούν το καθένα ξεχωριστά και όχι ως ολοκληρωμένο σύστημα. Αυτό το χαρακτηριστικό οδηγεί στην δεύτερη διαπίστωση ότι το συνολικό σύστημα είναι αόρατο (stealth). Κάποιος κακόβουλος δε μπορεί να εντοπίσει ούτε να παραβιάσει το σύστημα, ούτε να συλλέξει πληροφορία για την απόκριση του εκτελώντας διαδικασία scanning (διεξοδικός έλεγχος διεύθυνσης IP/θύρας ως προς την απόκριση).

Το Σύστημα Αυθεντικοποίησης DNS, στηρίζεται σε μια συστοιχία τριών μηχανών καθώς το Σύστημα Πρόσβασης αποτελείται από δύο μηχανές, το σύνολο των οποίων βασίζονται στο ανοιχτού κώδικα λειτουργικό σύστημα, OPENBSD στην έκδοση 5.9, με μοναδική εξαίρεση η μηχανή ACCESS-PFSENSE του Συστήματος Πρόσβασης, που «τρέχει» επίσης στο ανοιχτού κώδικα λειτουργικό FREEBSD, μέσω της οποίας πραγματοποιείται η δημιουργία της VPN σύνδεσης μεταξύ του ACCESS-SRV με τον target. Επιλέχθηκε η χρήση των εν λόγω λειτουργικών, χάρη στα χαρακτηριστικά που παρουσιάζουν ως προς την ασφάλεια, τη δομή και την σταθερότητα τους. Η υλοποίηση του συνολικού συστήματος αναπτύχτηκε πάνω στη πλατφόρμα (system board) της PC engines, APU-4C4/GTX-412TC 4GB[6] (εικόνες 8 και 9)

[6] https://www.pcengines.ch/apu4c4.htm



Εικόνα 8: System Board APU-4C4/GTX412TC 4GB (κάτοψη)



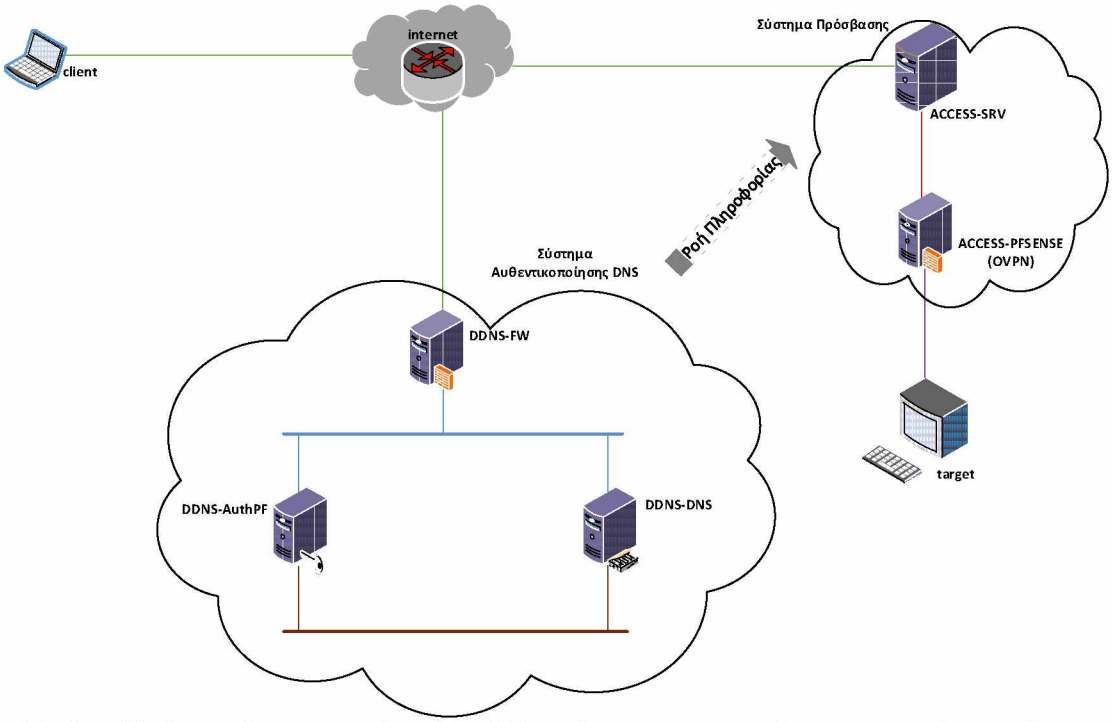
Εικόνα 9: System Board APU-4C4/GTX412TC 4GB (πίσω όψη)

Με τη βοήθεια της τεχνολογίας εικονικοποίησης (Virtualization Technology), εγκαταστάθηκε λογισμικό που υποστηρίζει την υλοποίηση εικονικών μηχανών (bare-metal hypervisor) της VMware και συγκεκριμένα ο ESXi 6.0 στην δωρεάν έκδοση του. Σκοπός είναι η συγκέντρωση του συνόλου

του λογισμικού σε έναν server, ώστε να επιτευχθεί όσο το δυνατόν μικρότερο κόστος ανάπτυξης και συντήρησης του όλου συστήματος σε συνδυασμό με τα πλεονεκτήματα της εύκολης και άμεσης φυσικής μεταφοράς του, οπουδήποτε.

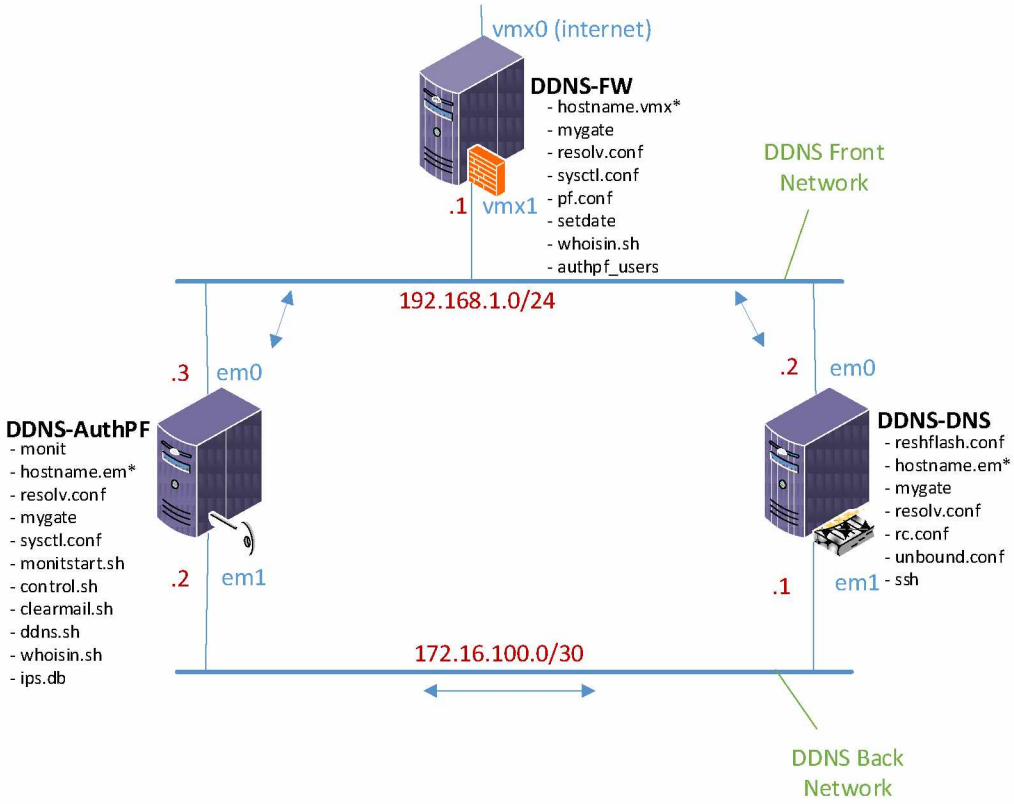
Για τις ανάγκες της παρουσίασης του συστήματος καθώς και για πρακτικούς λόγους, οι εξωτερικές στατικές IP (routable static IP's) που απαιτείται να διαθέτουν οι μηχανές DDNS-FW και ACCESS-SRV στις κάρτες δικτύου που είναι συνδεδεμένες με το internet, αντικαταστάθηκαν από ιδιωτικές στατικές IP (non routable static IP's – RFC 1918) καθώς το internet προσομοιάστηκε από έναν router που τοποθετήθηκε ως εικονική μηχανή στο εικονικό σύστημα.

Η διαμόρφωση της κάθε μηχανής του συστήματος ξεχωριστά, αποτέλεσε μια ιδιαίτερη πρόκληση για την λειτουργία και κυρίως για τη σταθερότητα του συνολικού συστήματος. Το βασικό ζητούμενο είναι η απρόσκοπτη λειτουργία του συστήματος, που έχει ως αποτέλεσμα την συνεχή και αδιάλειπτη σύνδεση του χρήστη με τη μηχανή-στόχο (target) σε οποιαδήποτε χρονική στιγμή και για όσο χρονικό διάστημα απαιτηθεί για να ολοκληρωθεί η κάθε εργασία.



Εικόνα 10: Απεικόνιση του Συνολικού Συστήματος με αποτύπωση των Συστημάτων Αυθεντικοποίησης και Πρόσβασης

3.3.2 Ανάλυση του Συστήματος Αυθεντικοποίησης



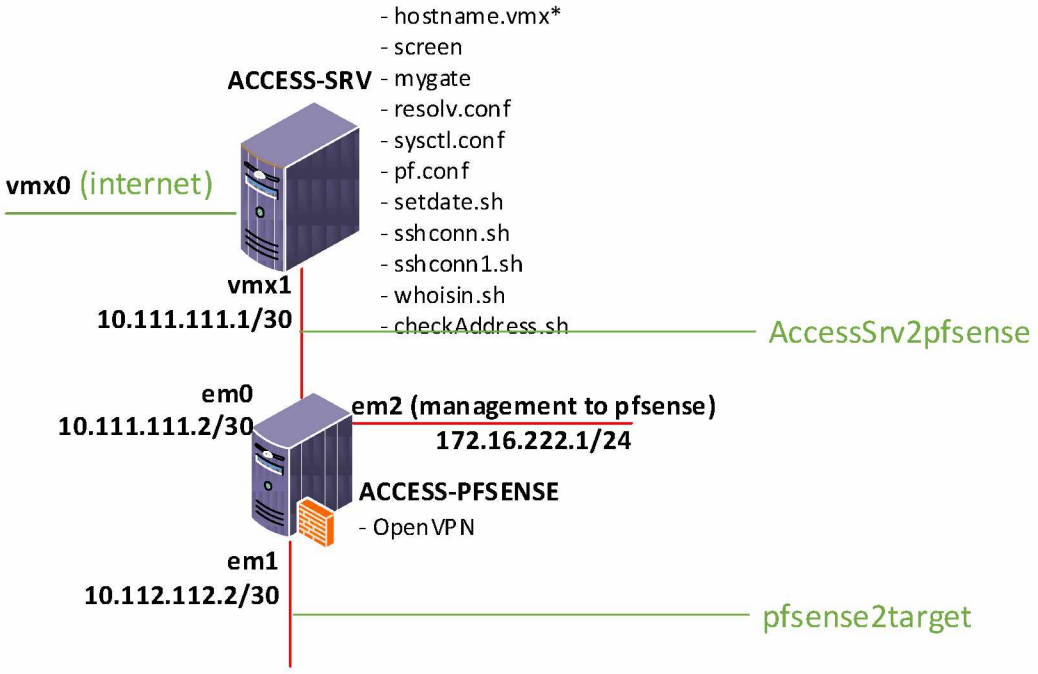
Εικόνα 11: Απεικόνιση του Συστήματος Αυθεντικοποίησης DNS

Μελετώντας την απεικόνιση του συστήματος αυθεντικοποίησης DNS, φαίνεται η αλληλουχία καθώς και ο συσχετισμός μεταξύ των μηχανών που το υλοποιούν. Σε αυτό το σημείο θα ήταν χρήσιμο να αναφερθεί ότι οι ονομασίες των μηχανών καθώς και των εσωτερικών δικτύων που δημιουργήθηκαν, έχουν δοθεί με τέτοιο τρόπο ώστε να περιγράφουν την λειτουργία τους ξεχωριστά, προκειμένου να γίνονται εύκολα κατανοητές και αναγνωρίσιμες στον διαχειριστή. Συγκεκριμένα, η ονομασία των μηχανών προήλθε από το αρχικό συνθετικό DDNS, το οποίο καθορίζει ότι το σύστημα μας στηρίζεται στην λειτουργία του δυναμικού DNS (DDNS) και το δεύτερο συνθετικό περιγράφει την λειτουργία της κάθε μηχανής ξεχωριστά. Επιπλέον η ονομασία των καρτών δικτύου, δίνεται αυτόματα από τον hypervisor (ESXi), αναγνωρίζοντας το μέσο εάν είναι αρχικά φυσικό ή εικονικό, (physical or virtual interface) σε συνδυασμό και με τον κατασκευαστή του μέσου, κατά την διάρκεια της αρχικής εγκατάστασης. Στην περίπτωση μας η ονομασία των interfaces θα είναι **vmx** και **em**. Ο αριθμός στη συνέχεια καθορίζει το πλήθος των interfaces που υπάρχουν σε κάθε μηχανή, ξεκινώντας με τον αριθμό 0.

Η πρώτη μηχανή, η οποία από εδώ και στη εξής θα αποκαλείται **DDNS-FW**, έχει το ρόλο της λειτουργίας ως Firewall για την προστασία του συστήματος από μη εξουσιοδοτημένους χρήστες. Η συγκεκριμένη μηχανή αποτελείται από δύο κάρτες δικτύου, την **vmx0** (εξωτερική κάρτα δικτύου) η

οποία συνδέεται στο internet και την **vmx1** (εσωτερική κάρτα δικτύου), η οποία συνδέεται με ένα εσωτερικό δίκτυο, που ονομάζεται **DDNS-Front Network** (192.168.1.0/24), μέσω του οποίου πραγματοποιείται η επικοινωνία με τις άλλες δύο μηχανές του συστήματος, την **DDNS-AuthPF** και την **DDNS-DNS**. Η δεύτερη μηχανή η **DDNS-AuthPF**, μέσω μιας σειράς διαδικασιών, ελέγχει εάν ο χρήστης που προσπαθεί να μπει στον target έχει την εξουσιοδότηση να το πράξει και ή του επιτρέπει την πρόσβαση, ή τον απορρίπτει. Αποτελείται από δύο κάρτες δικτύου, χρησιμοποιώντας την κάρτα δικτύου em0 ώστε να συνδέονται με το δίκτυο DDNS-Front Network και την κάρτα δικτύου em1 ώστε να συνδεθεί στο δεύτερο εσωτερικό ανεξάρτητο δίκτυο, που ονομάζεται **DDNS-Back Network** (172.16.100.0/30) διαμέσου του οποίου επικοινωνεί με την τρίτη μηχανή, την **DDNS-DNS** που είναι υπεύθυνη να καταχωρεί καθώς και να διαγράφει τις εγγραφές στο DNS, τις οποίες εγγραφές τις αποστέλλει η μηχανή DDNS-AuthPF. Η διασύνδεση της δικτυακά ακολουθεί την ίδια φιλοσοφία με την δεύτερη μηχανή, έχοντας δύο κάρτες δικτύου, χρησιμοποιώντας την κάρτα δικτύου em0 ώστε να συνδέεται με το δίκτυο DDNS-Front Network και την κάρτα δικτύου em1 ώστε να συνδεθεί στο δεύτερο εσωτερικό δίκτυο, το DDNS-Back Network, το οποίο δίκτυο χρησιμεύει για την απομακρυσμένη πρόσβαση της DDNS-AuthPF στις εγγραφές του DNS της DDNS-DNS.

3.3.3 Ανάλυση του Συστήματος Πρόσβασης



Εικόνα 12: Απεικόνιση του Συστήματος Πρόσβασης

Το σύστημα αποτελείται από δύο μηχανές, την ACCESS-SRV και τη ACCESS-PFSENSE. Η ACCESS-SRV διαθέτει δύο κάρτες δικτύου, με την vmx0 να χρησιμοποιείται ώστε να συνδέεται με το internet και την vmx1, μέσω ενός εσωτερικού δικτύου, συνδέεται στην επόμενη μηχανή την ACCESS-PFSENSE. Η μηχανή ACCESS-PFSENSE, είναι ένα επιπλέον firewall (pfsense Open Source Security[7]), υλοποιώντας μια OpenVPN σύνδεση με τον client και διαθέτει 3 κάρτες δικτύου. Μέσω της em0 συνδέεται σε εσωτερικό δίκτυο με την ACCESS-SRV, με την em2 να χρησιμοποιείται ώστε να πραγματοποιείται η τελική σύνδεση με τον target, ενώ η κάρτα δικτύου em1 χρησιμοποιείται για την διαχείριση της μηχανής.

Η βασική λειτουργία του συστήματος στηρίζεται στον ACCESS-SRV και χρησιμεύει ώστε να ταυτοποιήσει το χρήστη και την IP από την οποία προέρχεται.

Εφόσον είναι έγκυρος, επιτρέπει την πρόσβαση σε πακέτα UDP στη πόρτα 65130 προερχόμενα από την **έγκυρη** διεύθυνση IP και τα ανακατευθύνει στον επόμενο server, τον ACCESS-PFSENSE, που θα υλοποιήσει την OPENVPN σύνδεση με τον target.

Η εγκυρότητα ελέγχεται κάνοντας ερωτήσεις τύπου A record (host) στον δηλωμένο DNS server (DDNS-DNS), που είναι για όλους τους δηλωμένους χρήστες, κάτι που ανανεώνεται ανά ένα λεπτό. Την ευθύνη της ανανέωσης του πίνακα `<ddns>` την έχει το script `/root/checkAddress.sh`, το οποίο εκτελείται μέσω της διαδικασίας `cron` κάθε ένα λεπτό.

Κατά την εκκίνηση εκτελείται σύνδεση SSH (τύπου AuthPF) από το ACCESS-SRV στον firewall (FW) του συστήματος DDNS (DDNS-FW). Η σύνδεση κρατείται ενεργή συνεχώς μέσω του script `/root/sshconn.sh`, η οποία μεταφέρεται άλλο terminal (tty) μέσω του προγράμματος screen.

3.3.4 Ανάλυση της Διαδικασίας Σύνδεσης (client -> target)

Η αναλυτική διαδικασία που ακολουθείται, όταν ένας χρήστης (client) θέλει να συνδεθεί με ασφάλεια στον ACCESS-SERVER και από εκεί εν συνεχεία στο μηχάνημα στόχο (target) μέσα από το σύστημα ασφαλείας, εφαρμόζει τα βήματα όπως περιγράφονται στην αρχή λειτουργίας.

Αρχικά, ο client προκειμένου να έχει επιτυχή σύνδεση με τον target θα πρέπει να συντρέχουν οι δύο παρακάτω προϋποθέσεις:

- Να έχει καταχωρημένη δήλωση χρήστη (user) και public certificate στον DDNS-AuthPF και ACCESS-SERVER.

[7] https://www.pfsense.org/

- Να υπάρχει συμβατή με το username δήλωση σε πίνακα του ACCESS-SERVER, του DNS του (πχ user01.xyz).

Από τη στιγμή που οι παραπάνω δύο προϋποθέσεις ισχύουν, η διαδικασία σύνδεσης είναι η εξής: Ο client εκκινεί σύνδεση TCP στη πόρτα 55022 (TCP/55022) μέσω του DDNS-FW στον DDNS-AuthPF. Με την επιτυχή κατάληξη ενεργοποιείται η διαδικασία εξαγωγής της IP και του username από τα log files του DDNS-AuthPF. Με βάση την πληροφορία αυτή το ειδικό κείμενο (script) `control.sh` εκτελεί απομακρυσμένο έλεγχο του DDNS-DNS, μέσω του εσωτερικού ανεξάρτητου δικτύου (DDNS-Back Network) που συνδέονται μεταξύ τους, προσθέτοντας την εγγραφή DNS που περιμένει να βρει ο ACCESS-SERVER στον επόμενο κύκλο ερώτησης του. Παράλληλα εκτελείται από τον DDNS-AuthPF διαδικασία ελέγχου πολλαπλών εγγραφών.

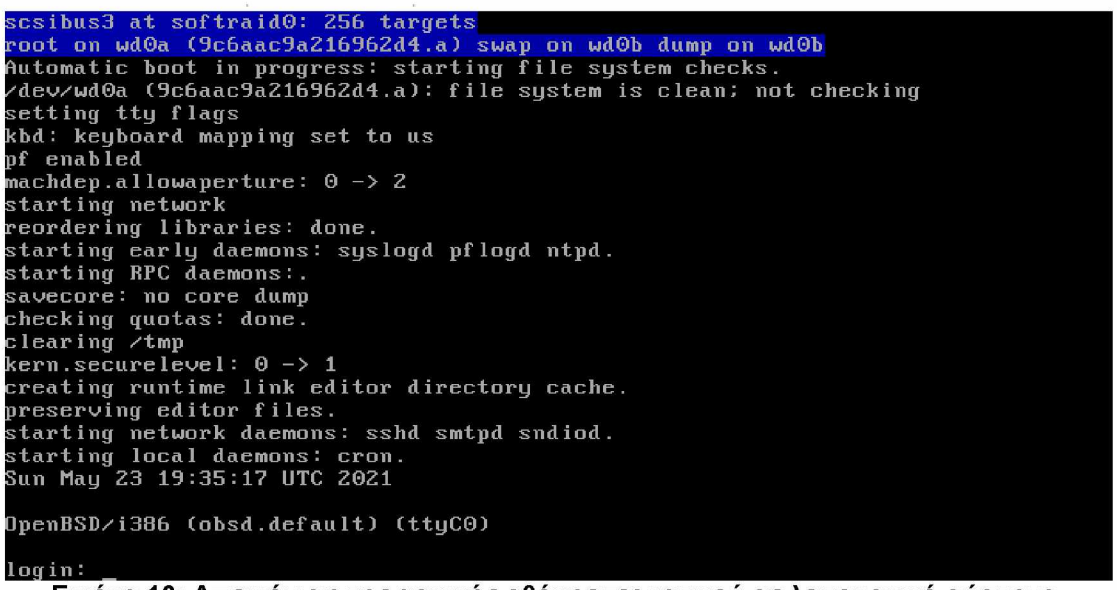
Μετά το απαραίτητο χρονικό διάστημα, που είναι λίγο μεγαλύτερο από τον κύκλο επανάληψης ερωτήσεων DNS του ACCESS-SERVER, ο client επιχειρεί σύνδεση TCP στη πόρτα 65122 στον ACCESS-SERVER. Εφόσον όλα τα παραπάνω συντρέχουν ευνοϊκά η σύνδεση αυτή αποκαθίσταται και θα παραμείνει ενεργή, όσο χρόνο ο client διατηρεί ενεργή και τη σύνδεση στον DDNS-AuthPF.

Έτσι τελικώς επιτρέπεται στον client να εκκινήσει τη σύνδεση OpenVPN (UDP/ πόρτα 55162), η οποία θα επιτρέψει τελικά την ολοκλήρωση της τελικής επιθυμητής σύνδεσης SSH με τον target.

Το σύνολο των ειδικών script καθώς και η διαμόρφωση των μηχανών του Συνολικού Συστήματος παρουσιάζονται αναλυτικά στην επόμενη παράγραφο.

3.3.5 Διαμόρφωση των μηχανών του Συνολικού Συστήματος (Configuration)

Η διαμόρφωση των μηχανών του συνολικού συστήματος, κρίνεται απαραίτητο να αποτυπωθεί πλήρως ώστε να μπορέσει να γίνει κατανοητή η λειτουργία του συστήματος σε τεχνικό επίπεδο.



Εικόνα 13: Απεικόνιση της αρχικής οθόνης τερματικού σε λειτουργικό σύστημα OpenBSD

3.3.5.1 Σύστημα Αυθεντικοποίησης DNS

3.3.5.1.1 DDNS-FW

Η συγκεκριμένη μηχανή έχει δημιουργηθεί από ένα ειδικά προσαρμοσμένο στιγμιότυπο (image) του λειτουργικού OpenBSD που ονομάζεται flashrd[8]. Μέσω της διαδικασίας του flashrd όλα τα partitions του file system είναι προκαθορισμένα να λειτουργούν ως read-only (χρήσιμο σε embedded συστήματα).

/etc/hostname.vmx0 [9]

```
inet 192.168.69.2 255.255.255.252 NONE
```

/etc/hostname.vmx1

```
inet 192.168.1.1 255.255.255.0 NONE
```

/etc/resolv.conf [10]

```
lookup file bind
nameserver 8.8.8.8
nameserver 8.8.4.4
```

/etc/mygate[11]

```
192.168.69.1
```

[8] https://www.nmedia.net/flashrd
[9] https://man.openbsd.org/hostname.if.5
[10] https://man.openbsd.org/resolv.conf.5
[11] https://man.openbsd.org/mygate.5

/etc/sysctl.conf[12]

Λόγω της ανάγκης που προκύπτει για την δρομολόγηση των πακέτων (IP forwarding ή κοινώς routing), ενεργοποιείται μόνο η πρώτη παράμετρος στο αρχείο `/etc/sysctl.conf`, ενώ όλες οι υπόλοιπες παραμένουν στην προκαθορισμένη ρύθμιση τους.

Το `sysctl.conf` είναι ένα αρχείο όπου περιέχει διάφορες μεταβλητές που αναφέρονται και ως System Controls και επηρεάζουν τον πυρήνα του λειτουργικού κατά την εκκίνηση του. Αυτό σημαίνει ότι ένας διαχειριστής μπορεί να επέμβει στο συγκεκριμένο αρχείο και να επιλύσει διάφορα προβλήματα, να αλλάξει την συμπεριφορά ή λειτουργία της μηχανής ακόμη και να πραγματοποιήσει την αναμεταγλώττιση (recompiling) του πυρήνα.

Ρυθμίζοντας τις μεταβλητές κατάλληλα, με μία απλή αφαίρεση του συμβόλου hash «#» καθώς επίσης αλλάζοντας την τιμή τους σε `0` ή `1`, ενεργοποιούνται ή απενεργοποιούνται. Στη συγκεκριμένη περίπτωση, έχει ενεργοποιηθεί η ανακατεύθυνση (routing) των πακέτων, με την μεταβλητή `net.inet.ip.forwarding=1`, να έχει αφαιρεθεί αρχικά το σύμβολο «#» ώστε να ενεργοποιηθεί κατά την εκκίνηση του λειτουργικού και στη συνέχεια έχει μεταβληθεί η τιμή της από `0` σε `1`, ώστε να λειτουργήσει η ανακατεύθυνση των πακέτων.

```
#	$OpenBSD: sysctl.conf,v 1.4 2015/04/03 15:50:28 millert
Exp $
#
# This file contains a list of sysctl options the user wants
set at
# boot time.  See sysctl(3) and sysctl(8) for more information
on
# the many available variables.
#
net.inet.ip.forwarding=1	# 1=Permit forwarding (routing) of
IPv4 packets
#net.inet.ip.mforwarding=1	# 1=Permit forwarding (routing) of
IPv4 multicast packets
#net.inet.ip.multipath=1	# 1=Enable IP multipath routing
#net.inet.icmp.rediraccept=1	# 1=Accept ICMP redirects
#net.inet6.ip6.forwarding=1	# 1=Permit forwarding (routing) of
IPv6 packets
#net.inet6.ip6.mforwarding=1	# 1=Permit forwarding
(routing) of IPv6 multicast packets
#net.inet6.ip6.multipath=1	# 1=Enable IPv6 multipath routing
#net.inet.tcp.always_keepalive=1 # 1=Keepalives for all
connections (e.g. hotel/airport NAT)
#net.inet.tcp.keepidle=100	# 100=send TCP keepalives every 50
seconds
#net.inet.esp.enable=0		# 0=Disable the ESP IPsec
protocol
```

[12] https://man.openbsd.org/sysctl.conf.5

```
#net.inet.ah.enable=0       # 0=Disable the AH IPsec protocol
#net.inet.esp.udpencap=0    # 0=Disable ESP-in-UDP
encapsulation
#net.inet.ipcomp.enable=1   # 1=Enable the IPCOMP protocol
#net.inet.etherip.allow=1   # 1=Enable the Ethernet-over-IP
protocol
#net.inet.tcp.ecn=1         # 1=Enable the TCP ECN extension
#net.inet.carp.preempt=1    # 1=Enable carp(4) preemption
#net.inet.carp.log=3        # log level of carp(4) info,
default 2
#net.pipex.enable=1         # 1=Enable pipex(4) for npppd(8)
#ddb.panic=0                # 0=Do not drop into ddb on a
kernel panic
#ddb.console=1              # 1=Permit entry of ddb from the
console
#ddb.log=1            # 1=Log ddb output in kernel message
buffer
#fs.posix.setuid=0          # 0=Traditional BSD chown()
semantics
#vm.swapencrypt.enable=0    # 0=Do not encrypt pages that go to
swap
#vfs.nfs.iothreads=4        # Number of nfsio kernel threads
#net.inet.ip.mtudisc=0           # 0=Disable tcp mtu discovery
#kern.splassert=2           # 2=Enable with verbose error
messages
#kern.nosuidcoredump=3           # 3=Put suid coredumps in
/var/crash/progname
#kern.watchdog.period=32    # >0=Enable hardware watchdog(4)
timer if available
#kern.watchdog.auto=0       # 0=Disable automatic watchdog(4)
retriggering
#hw.allowpowerdown=0        # 0=Disable power button shutdown
#machdep.allowaperture=2    # See xf86(4)
#machdep.apmhalt=1          # 1=powerdown hack, try if halt -p
doesn't work
#machdep.kbdreset=1         # permit console CTRL-ALT-DEL to do
a nice halt
#machdep.lidsuspend=0       # do not suspend laptop upon lid
closing
#machdep.userldt=1          # allow userland programs to play
with ldt,
                    # required by some ports
#kern.emul.linux=1          # enable running Linux binaries
```

/var/cron/tabs/root[13]

Το αρχείο `crontab` περιέχει οδηγίες προς το πρόγραμμα `cron` που εκτελείται στο υπόβαθρο (αυτόνομη διεργασία - daemon) να εκτελεί διάφορες διαδικασίες του συστήματος ή scripts του διαχειριστή σε συγκεκριμένα χρονικά διαστήματα που δηλώνονται σε αυτό. Μέσω του `crontab` δίνεται η δυνατότητα για την αυτοματοποίηση ορισμένων διαδικασιών.

[13] https://man.openbsd.org/crontab.5

Στην συγκεκριμένη περίπτωση έχει δηλωθεί το script `/root/setdate` να εκτελείται κάθε 30 λεπτά.

```
#(Cron version V5.0 -- $OpenBSD: crontab.c,v 1.57 2009/01/29
22:50:16 sobrado Exp $)
#
SHELL=/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log
#
#minute    hour  mday  month wday  command
#
#sendmail clientmqueue runner
*/30  *     *     *     *     /usr/sbin/sendmail -L sm-msp-queue
-Ac -q
#
#rotate log files every hour, if necessary
0     *     *     *     *     /usr/bin/newsyslog
#send log file notifications, if necessary
#1-59 *     *     *     *     /usr/bin/newsyslog -m
#
#do daily/weekly/monthly maintenance
30    1     *     *     *     /bin/sh /etc/daily
30    3     *     *     6     /bin/sh /etc/weekly
30    5     1     *     *     /bin/sh /etc/monthly
#0    *     *     *     *     /usr/libexec/spamd-setup
*/30  *     *     *     *     /root/setdate
```

/etc/pf.conf[14]

Στο αρχείο διαμόρφωσης (configuration file) του pf.conf περιγράφονται και εφαρμόζονται οι κανόνες του firewall επιτρέποντας σε συγκεκριμένα πρωτόκολλα και συγκεκριμένα στα TCP, UDP και ICMP, την είσοδο και έξοδο τους, συνδυάζοντας παράλληλα NAT[15][16] (Network Address Translation) και Traffic Redirection[17] (Port Forwarding) για την κίνηση των πακέτων.

Επιπλέον καλείται ο πίνακας <bruteforce> που σκοπό έχει να αποθηκεύει τις ip διευθύνσεις από τις οποίες η μηχανή μπορεί να δεχτεί επιθέσεις τύπου brute-force. Αυτή η διαδικασία χρησιμεύει ώστε ο διαχειριστής να διαθέτει άμεσα αυτή την πληροφορία με σκοπό τη

[14] https://man.openbsd.org/pf.conf.5
[15] https://www.openbsd.org/faq/pf/nat.html
[16] Το σύστημα NAT λειτουργεί σε κάποιον δρομολογητή, ο οποίος συνδέει συνήθως δύο δίκτυα και μεταφράζει τις ιδιωτικές διευθύνσεις του εσωτερικού δικτύου σε νόμιμες διευθύνσεις προτού τα πακέτα προωθηθούν σε άλλο δίκτυο. Σαν μέρος αυτής της λειτουργίας το NAT μπορεί να ρυθμιστεί να κάνει γνωστή μόνο μία διεύθυνση στον έξω κόσμο για ολόκληρο το δίκτυο που συνδέει με αυτόν. Αυτό το χαρακτηριστικό παρέχει επιπλέον ασφάλεια αφού κρύβει ολόκληρο το εσωτερικό δίκτυο από το κόσμο πίσω από μία διεύθυνση.
[17] https://www.openbsd.org/faq/pf/rdr.html

δρομολόγηση επιπλέον ενεργειών που μπορεί να απαιτηθούν ώστε να ενισχυθεί η ασφάλεια του συστήματος.

Οι μακροεντολές που αναφέρονται αρχικά και αντιστοιχούν στα interfaces της μηχανής, έχουν χρησιμοποιηθεί για την διευκόλυνση του διαχειριστή.

```
#MACROS
wlan="vmx0"
ilan="vmx1"

table <bruteforce> persist
set skip on lo

#NAT
pass out on $wlan from any to any nat-to ($wlan)

#Redirections for all users (special rdrs through authpf)
pass in quick on $wlan proto tcp from any to ($wlan) port
55022 \
     rdr-to 192.168.1.3 port 22

#Redirection Rules for PFAuthenticated users (remote access)
anchor "authpf/*"

#Rules
pass quick proto icmp

pass in quick proto udp from 195.46.12.60 to port 53 \
     rdr-to 192.168.1.3 port 53
block quick from <bruteforce>
block in on $wlan
pass in quick on $wlan proto tcp from any to port 60022 flags
S/SA keep state (max-src-conn-rate 3/30, overload <bruteforce>
flush global)

pass in on $ilan
pass out on $ilan
```

/root/setdate.sh

Στο script `setdate.sh` καθορίζονται οι πληροφορίες που αφορούν στην ημερομηνία καθώς και την ώρα που λαμβάνει η μηχανή από συγκεκριμένο Network Time Protocol (NTP) server[18]. Αυτό χρησιμεύει στη μηχανή να γνωρίζει τον πραγματικό χρόνο ώστε εν συνεχεία να μπορούν να υλοποιηθούν οι εισερχόμενες συνδέσεις authpf.

```
#!/bin/sh
/usr/sbin/rdate -4 -n ntp.grnet.gr
```

[18] https://grnet.gr/services/internet-services/ntp/

/root/whoisin.sh

Το script `/root/whoisin.sh` εμφανίζει τις ενεργές συνδέσεις `authpf`, ώστε το σύστημα να γνωρίζει ανά πάσα στιγμή ποιοι χρήστες είναι συνδεδεμένοι και για πόσο χρονικό διάστημα διήρκησε η κάθε σύνδεση ξεχωριστά.

```
#!/bin/sh
/bin/ps -ax | /usr/bin/grep authpf
```

3.3.5.1.2 DDNS-AuthPF

/etc/hostname.em0

```
inet 192.168.1.3 255.255.255.0
```

/etc/hostname.em1

```
inet 172.16.100.2 255.255.255.252
```

/etc/resolv.conf

```
lookup file bind
nameserver 192.168.1.2
```

/etc/mygate

```
192.168.1.1
```

/etc/pf.conf

Στο αρχείο διαμόρφωσης (configuration file) του `pf.conf` αρχικά καλείται ο πίνακας `<authpf_users>` όπου περιέχει καταχωρημένες τις IP διευθύνσεις των χρηστών που έχουν το δικαίωμα να εκτελέσουν σύνδεση `authpf` στην μηχανή. Η μεταβλητή `persist` υποδηλώνει στον πυρήνα (kernel) του λειτουργικού να κρατά τον συγκεκριμένο πίνακα στη μνήμη flash ακόμη και στη περίπτωση που κανένα πακέτο δεν αναφέρεται σε αυτόν. Λόγω του συγκεκριμένου τρόπου αποθήκευσης, οι ιδιότητες που παρουσιάζει η χρήση των πινάκων που δημιουργούνται εντός του αρχείου κανόνων του `pf.conf`, είναι η εξαιρετικά γρήγορη ταχύτητα αναζήτησης των εγγραφών που περιέχουν.

Στην τρίτη σειρά πραγματοποιείται η συσχέτιση των κανόνων του `authpf` με τους κύριους κανόνες που ακολουθούνται μέσω του αρχείου `pf.conf`. Σε οποιοδήποτε σημείο τοποθετείται η χρήση του κανόνα `anchor` εντός της σειράς των κανόνων του `pf.conf`, είναι το σημείο στο οποίο το

πρόγραμμα του PF παρεκκλίνει από την ακολουθία τις σειράς των κύριων κανόνων που περιέχει ώστε να εξετάσει και να αξιολογήσει τους κανόνες του `authpf`.

Τέλος περιγράφονται και εφαρμόζονται οι κανόνες του firewall επιτρέποντας στα συγκεκριμένα πρωτόκολλα TCP, UDP και ICMP, την είσοδο και έξοδο τους, συνδυάζοντας παράλληλα NAT για την κίνηση των πακέτων.

```
table <authpf_users> persist

set skip on lo

anchor "authpf/*"

pass out quick on em0 proto {tcp,udp,icmp} from any to any /
nat-to (em0)
pass in on em1

pass on em1
```

/etc/sysctl.conf

Λόγω της ανάγκης που προκύπτει για την δρομολόγηση των πακέτων, ενεργοποιείται μόνο η πρώτη παράμετρος (`net.inet.ip.forwarding=1`) στο αρχείο `/etc/sysctl.conf`, ενώ όλες οι υπόλοιπες παραμένουν στην προκαθορισμένη ρύθμιση τους.

```
#    $OpenBSD: sysctl.conf,v 1.4 2015/04/03 15:50:28 millert
Exp $
#
# This file contains a list of sysctl options the user wants
set at
# boot time.  See sysctl(3) and sysctl(8) for more information
on
# the many available variables.
#
net.inet.ip.forwarding=1    # 1=Permit forwarding (routing) of
IPv4 packets
#net.inet.ip.mforwarding=1  # 1=Permit forwarding (routing) of
IPv4 multicast packets
#net.inet.ip.multipath=1    # 1=Enable IP multipath routing
#net.inet.icmp.rediraccept=1     # 1=Accept ICMP redirects
#net.inet6.ip6.forwarding=1# 1=Permit forwarding (routing) of
IPv6 packets
#net.inet6.ip6.mforwarding=1     # 1=Permit forwarding
(routing) of IPv6 multicast packets
#net.inet6.ip6.multipath=1  # 1=Enable IPv6 multipath routing
#net.inet.tcp.always_keepalive=1 # 1=Keepalives for all
connections (e.g. hotel/airport NAT)
#net.inet.tcp.keepidle=100  # 100=send TCP keepalives every 50
seconds
```

```
#net.inet.esp.enable=0           # 0=Disable the ESP IPsec
protocol
#net.inet.ah.enable=0       # 0=Disable the AH IPsec protocol
#net.inet.esp.udpencap=0    # 0=Disable ESP-in-UDP
encapsulation
#net.inet.ipcomp.enable=1   # 1=Enable the IPCOMP protocol
#net.inet.etherip.allow=1   # 1=Enable the Ethernet-over-IP
protocol
#net.inet.tcp.ecn=1         # 1=Enable the TCP ECN extension
#net.inet.carp.preempt=1    # 1=Enable carp(4) preemption
#net.inet.carp.log=3        # log level of carp(4) info,
default 2
#net.pipex.enable=1         # 1=Enable pipex(4) for npppd(8)
#ddb.panic=0                # 0=Do not drop into ddb on a
kernel panic
#ddb.console=1              # 1=Permit entry of ddb from the
console
#ddb.log=1            # 1=Log ddb output in kernel message
buffer
#fs.posix.setuid=0          # 0=Traditional BSD chown()
semantics
#vm.swapencrypt.enable=0    # 0=Do not encrypt pages that go to
swap
#vfs.nfs.iothreads=4        # Number of nfsio kernel threads
#net.inet.ip.mtudisc=0           # 0=Disable tcp mtu discovery
#kern.splassert=2           # 2=Enable with verbose error
messages
#kern.nosuidcoredump=3           # 3=Put suid coredumps in
/var/crash/progname
#kern.watchdog.period=32    # >0=Enable hardware watchdog(4)
timer if available
#kern.watchdog.auto=0       # 0=Disable automatic watchdog(4)
retriggering
#hw.allowpowerdown=0        # 0=Disable power button shutdown
#machdep.allowaperture=2    # See xf86(4)
#machdep.apmhalt=1          # 1=powerdown hack, try if halt -p
doesn't work
#machdep.kbdreset=1         # permit console CTRL-ALT-DEL to do
a nice halt
#machdep.lidsuspend=0       # do not suspend laptop upon lid
closing
```

Monit

Η χρηστικότητα του προγράμματος `Monit` έγκειται ώστε να λειτουργεί ως διαχειριστής καθώς και παρατηρητής διαφόρων διαδικασιών, προγραμμάτων, φακέλων και γενικότερα του συστήματος αρχείων φακέλων στα λειτουργικά συστήματα Unix. Μπορεί να εκκινήσει μία διαδικασία ή ακόμη και να την επανεκκινήσει.

Αυτή την πολύ χρήσιμη ιδιότητα εκμεταλλεύεται η μηχανή DDNS-Authpf ώστε να τρέχει διαρκώς τη διαδικασία ελέγχου του `authpf`.

/root/monitstart.sh

Το script που εκκινεί το πρόγραμμα Monit.

```
#!/bin/sh
/usr/local/bin/monit -c /etc/monitrc
exit 0
```

/root/whoisin.sh

Το script `/root/whoisin.sh` εμφανίζει τις ενεργές συνδέσεις `authpf`, όπως ακριβώς περιγράφεται στη προηγούμενη παράγραφο για την μηχανή DDNS-FW.

```
#!/bin/sh
/bin/ps -ax | /usr/bin/grep authpf
```

/sql/clearmail.sh

Το script `/sql/clearmail.sh` φροντίζει για το σταδιακό καθαρισμό του αρχείου mail του διαχειριστή από τις διάφορες εγγραφές που προκαλούνται από τις διεργασίες που θέλουν να τον ενημερώσουν για οτιδήποτε.

Αυτή η διαδικασία εκτελείται προκειμένου μακροπρόθεσμα να μην γεμίσει το αποθηκευτικό σύστημα της μηχανής.

```
#!/bin/sh
rm /var/mail/root
```

/sql/control.sh

Το script `/sql/control.sh` ελέγχει και διορθώνει τις πιθανές ανωμαλίες στη βάση δεδομένων.

```
#!/bin/sh


#*********************************************
# for every user with UID > 1000
#*********************************************

for D in `getent passwd | awk -F: '$3 > 1000 {print $1}'`

#*********************************************
# Loop
#*********************************************

do
```

```
#*********************************************
# check if there are more than one records in ddns
# for each user (stored in variable result):
# If >1 then remove all ddns records
# for this user,
# find and kill all authpf processes for this user
# and finally Bann the user (create file in /etc/authpf/banned
#*********************************************
# If records =1 then check if there is an
# authpf session for this particular user.
# --if YES all is normal
# --if NO it is not normal. Log and delete ddns record
#*********************************************
# If =0 then check if there is an
# authpf session for this particular user.
# --if NO this is normal
# --if YES this is not normal. Log and kill authpf session
#*********************************************
# $D is the user name
# $result is the number of records in DDNS for this user
# $authres is the authpf concurrency for this user
# $pid is the authpf process for that user
#*********************************************

user="$D.x.y"
result=$(nslookup $user | grep $user | grep -v 'find' | wc -l)
authres=$(ps -a | grep "authpf: $D" | grep -v 'grep' | wc -l)
pid=$(ps -a | grep "authpf: $D" | grep -v 'grep' | awk '{print
$1}')

echo $D $result $authres $pid

# check number of records and act if >1
if [ $result -gt 1 ]; then
echo "alert";
/usr/sbin/unbound-control -s 172.16.100.1 local_data_remove
"$user.";
for X in `ps -a | grep "authpf: $D" | grep -v 'grep' | awk
'{print $1}'`
do
     kill $X
done
echo "you have been banned." > /etc/authpf/banned/$D
else
 :
fi

# number of records is 1 and check authpf sessions
if [ $result -eq 1 ] && [ $authres -eq 0 ]; then
# delete dns record
/usr/sbin/unbound-control -s 172.16.100.1 local_data_remove
"$user.";
fi

# number of records is 0 and authpf session exist
if [ $result -eq 0 ] && [ $authres -eq 1 ]; then
```

```
# kill authpf session
for X in `ps -a | grep "authpf: $D" | grep -v 'grep' | awk
'{print $1}'`
do
      kill $X
done
fi

done
```

/sql/ddns.sh

Το script `/sql/ddns.sh` παράγει εγγραφές στο DNS (unbound) μέσω απομακρυσμένου ελέγχου για κάθε χρήστη ο οποίος έχει επιτυχώς αυθεντικοποιεί στο `authpf`.

Παράλληλα καταργεί τις εγγραφές των χρηστών που έχουν αποσυνδεθεί από το `authpf`.

```
#!/bin/sh

#*****************************************************
#           Check for Previous executions that       *
#          that are not finished. Wait for      *
#          them to finish.                      *
#*****************************************************

while [ -f /sql/.test.test ]; do
null
done
touch /sql/.test.test

#*****************************************************
#           Insertion Routine                        *
#*****************************************************
# clear the temporary table from any contents
sqlite3 /sql/ips.db "delete from currentIPS;"
# main routine
# for each line returning from process searching
# with authpf keyword do the following:
# 1st extract the IP
# 2nd extract the user
# 3rd add to the DNS server the corresponding entry for the
user
# 4th insert into permanent table the above action only if
#     the insertion is already done from previous execution
#     and the user is still active
# 5th insert into temporary (only for this session) table
#     the previous entry anyway regardless its existence in
the
#     permanent table, so it will not be deleted since it is
#     still active
#******************* 1st
```

```
ps -a | grep authpf | grep -v "grep" | while read line
      do
ip=$(echo $line | grep -o '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-
9]\{1,3\}\.[0-9]\{1,3\}')
#******************* 2nd
w1="-authpf:"
w2="@"
user=$(echo $line | awk -v w1="$w1" -v \
        w2="$w2" 'match($0, w1 ".*" w2){print
substr($0,RSTART+length(w1),RLENGTH-length(w1 w2))}')
user=${user#\ }
#******************* 3rd
/usr/sbin/unbound-control -s 172.16.100.1 local_data
"$user.x.y. IN A $ip" > /dev/null
#******************* 4th
sqlite3 /sql/ips.db "insert into ips (type, IP, user,
insertTime, removeTime) \
      select 'I', '$ip', '$user', current_timestamp, null \
      where not exists(select 1 from ips where user='$user' and \
      removeTime is null);"
#******************* 5th
sqlite3 /sql/ips.db "insert into currentIPS (IP,user) values \
      ('$ip','$user');"
done

#*****************************************************
#           Removal   Routine                        *
#*****************************************************
#
# Return one at a line every user that is not removed
#     from the permanent table and does not belong
#     to the temporary table (meaning he is not active)
# For each line do the following
# 1st find the id of the permanent table line
# 2nd find the user name
# 3rd remove from the DDNS server the corresponding entry
# 4th update the permanent table with the removal action
# 5th remove the test file to clear for the next script
execution
#******************* 1st
sqlite3 /sql/ips.db "select id, user \
      from ips where removeTime is null \
      and user not in (select user from currentIPS);" | while
read line
      do
id=$(echo $line | awk -F "\|" '{print $1}')
#******************* 2nd
user=$(echo $line | awk -F "\|" '{print $2}')
#******************* 3rd
/usr/sbin/unbound-control -s 172.16.100.1 local_data_remove
"$user.x.y."
#******************* 4th
sqlite3 /sql/ips.db \
      "update ips set \
      type='R', \
      removeTime=current_timestamp where id=$id;"
```

```
done
#****************** 5th
rm -f /sql/.test.test
exit 0
```

/sql/ips.db

Είναι η κύρια Βάση Δεδομένων και έχει ονομαστεί `ips.db`. Είναι κατασκευασμένη με την ανοιχτού λογισμικού SQLite στην έκδοση 3.9.2, η οποία και είναι προεγκατεστημένη στο λειτουργικό σύστημα OpenBSD. Εδώ αποθηκεύονται όλα τα στοιχεία που απαιτούνται κατά τη διάρκεια εκτέλεσης των διαδικασιών που περιγράφονται στο script `/sql/ddns.sh`. Το σχήμα της Βάσης Δεδομένων αποτελείται από δύο πινάκες. Τον πίνακα `ips` όπου περιέχει έξι πεδία, με το πεδίο `id` να είναι και το κύριο κλειδί και τον πίνακα `currentIPS` όπου περιέχει δύο πεδία.

Παρακάτω παρουσιάζεται ως παράδειγμα το σχήμα της Βάσης Δεδομένων καθώς και ένα μέρος των δεδομένων που περιέχει. Διακρίνονται οι πίνακες, τα πεδία που περιέχονται σε αυτούς, οι τιμές που λαμβάνουν καθώς και δεδομένα που είναι αποθηκευμένα.

```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE ips (
id integer primary key, type text,
IP text, user text, insertTime DATETIME, removeTime DATETIME);
INSERT INTO "ips" VALUES(1,'R','95.95.95.95','client01','2021-
01-20 08:58:11','2021-01-20 08:59:01');
INSERT INTO "ips" VALUES(2,'R','35.35.35.35','client02','2021-
01-20 09:01:02','2021-01-20 09:01:22');
CREATE TABLE currentIPS (IP text, user text);
COMMIT;
```

3.3.5.1.3 DDNS-DNS

Η συγκεκριμένη μηχανή έχει υλοποιηθεί με τη χρήση ενός ακόμη εργαλείου του λογισμικού OpenBSD που ονομάζεται reshflash, δημιουργώντας με αυτό τον τρόπο στιγμιότυπα (images) του λειτουργικού, που είναι ανθεκτικά σε πιθανή αλλοίωση του δίσκου του συστήματος ή σε περίπτωση απώλειας του ρεύματος διότι το σύστημα αρχείων του «τρέχει» στη μνήμη RAM απευθείας και λειτουργεί ως read-only.

/cfg/var/reshflash.conf

```
# Configure automatic file backups on shutdown
# Copyright Brian Conway <bconway@rcesoftware.com>, see
LICENSE for details
```

```
# Save targets can be files, directories, or '.' (entire /etc
or /var).
# Save targets MUST use paths relative to /etc or /var.
# Wildcards ARE supported, but '.' is preferable to '*' for
entire /etc or /var.
#
# NOTE: This file is automatically saved when save_etc or
save_var are enabled.
# If both are later disabled, it should be removed manually
from /cfg/etc.

#save_etc='resolv.conf'
save_var='db/host.random db/pkg'
# Configure automatic file backups on shutdown
```

/cfg/etc/hostname.em0

```
inet 192.168.1.2 255.255.255.0 NONE
```

/cfg/etc/hostname.em1

```
inet 172.16.100.1 255.255.255.252 NONE
```

/cfg/etc/resolv.conf

```
lookup file bind
nameserver 8.8.8.8
```

/cfg/etc/mygate

```
192.168.168.1
```

/cfg/etc/rc.conf

Το αρχείο `rc.conf` καθορίζει μέσω του συγκεκριμένου αρχείου τη παραμετροποίηση των αυτόνομων διεργασιών του συστήματος που εκτελούνται στο παρασκήνιο (daemon). Το αρχείο περιέχει διάφορες μεταβλητές που αφορούν στις συγκεκριμένες διεργασίες, οι οποίες λαμβάνουν δύο τιμές, με την `NO` να είναι ήδη προκαθορισμένη ως τιμή σχεδόν στο σύνολο αυτών και με βάση την οποία είναι απενεργοποιημένες και σε περίπτωση που επιλέξει ο διαχειριστής την ενεργοποίηση μιας λειτουργίας μεταβάλει την τιμή της σε `YES`. Όλες οι λειτουργίες είναι προκαθορισμένες και εγκατεστημένες ήδη από την αρχική εγκατάσταση του λειτουργικού και ενεργοποιούνται από τον διαχειριστή ανάλογα την λειτουργία που έχει επιλέξει να εκτελεί η μηχανή που έχει δημιουργήσει. Με το σύμβολο hash (#) εμφανίζονται τα διάφορα σχόλια που υπάρχουν μέσα στο αρχείο και έχουν επεξηγηματικό χαρακτήρα για τις διάφορες διεργασίες ως επιπλέον βοήθεια προς τον διαχειριστή.

Στην περίπτωση της μηχανής DDNS-DNS, έχει ενεργοποιηθεί η παράμετρος που αφορά στην λειτουργία του προγράμματος `unbound`

(unbound_flags='') όπου και είναι ο επιλυτής ονομάτων τομέα, ώστε η συγκεκριμένη μηχανή να εκτελεί την εργασία ενός DNS server που απαιτείται για το Σύστημα Αυθεντικοποίησης.

```
#	$OpenBSD: rc.conf,v 1.211 2015/12/06 13:51:41 rpe Exp $

# DO NOT EDIT THIS FILE!!
#
# This file defines the default service selection as shipped
in a
# release.  Upgrades of your system will modify this file.
#
# To select the service options you desire, please override
these
# options in the file /etc/rc.conf.local
#
# DO NOT EDIT THIS FILE!!

# Set these variables to "NO" to turn the respective service
off.
# Set them to "" to run them with the default flags.
# Otherwise, these variables override the default flags.
apmd_flags=NO
bgpd_flags=NO
bootparamd_flags=NO
cron_flags=
dhcpd_flags=NO
dhcrelay_flags=NO	# for normal use: "-i interface [server]"
dvmrpd_flags=NO
eigrpd_flags=NO
ftpd_flags=NO		# set to NO if ftpd is running out of
inetd
ftpproxy_flags=NO
ftpproxy6_flags=NO
hostapd_flags=NO
hotplugd_flags=NO
httpd_flags=NO
identd_flags=NO
ifstated_flags=NO
iked_flags=NO
inetd_flags=NO
isakmpd_flags=NO
iscsid_flags=NO
ldapd_flags=NO
ldattach_flags=NO	# for normal use: "[options] linedisc
cua-device"
ldomd_flags=NO
ldpd_flags=NO
lpd_flags=NO		# for normal use: "" (or "-l" for
debugging)
mopd_flags=NO
mrouted_flags=NO # be sure to enable multicast below
npppd_flags=NO
nsd_flags=NO
```

```
ntpd_flags=
ospfd_flags=NO
ospf6d_flags=NO
pflogd_flags=          # add more flags, e.g. "-s 256"
radiusd_flags=NO
rarpd_flags=NO
rbootd_flags=NO
relayd_flags=NO
rebound_flags=NO
ripd_flags=NO
route6d_flags=NO # be sure to set net.inet6.ip6.forwarding=1
rtadvd_flags=NO        # for normal use: list of interfaces
                 # be sure to set net.inet6.ip6.forwarding=1
sasyncd_flags=NO
sensorsd_flags=NO
slowcgi_flags=NO
smtpd_flags=
sndiod_flags=
snmpd_flags=NO
spamd_flags=NO         # also see spamd_black below
spamlogd_flags=        # use eg. "-i interface" and see
spamlogd(8)
sshd_flags=
syslogd_flags=         # add more flags, e.g. "-u -a
/chroot/dev/log"
tftpd_flags=NO
tftpproxy_flags=NO
unbound_flags=''
vmd_flags=NO
watchdogd_flags=NO
wsmoused_flags=NO      # for enabling console mouse support
(i386 alpha amd64)
                 # for ps/2 or usb mice: "", serial: "-p
/dev/cua00"
xdm_flags=NO           # on some architectures, you must also
                 # disable console getty in /etc/ttys

# services related to RPC, NFS, and YP
amd_flags=NO           # also see amd_master below
lockd_flags=NO
mountd_flags=NO
nfsd_flags=NO
portmap_flags=NO # note: inetd(8) rpc services need portmap too
statd_flags=NO
ypbind_flags=NO
ypldap_flags=NO
ypserv_flags=NO

# set the following to "YES" to turn them on
pf=YES                 # Packet filter / NAT
ipsec=NO         # IPsec
check_quotas=YES # NO may be desirable in some YP environments
accounting=NO          # process accounting (using
/var/account/acct)

# Multicast routing configuration
```

```
# Please look at netstart(8) for a detailed description if you
change these
multicast=NO          # Reject IPv4 multicast packets by
default

# miscellaneous other flags
amd_master=/etc/amd/master  # AMD 'master' map
savecore_flags=             # "-z" to compress
spamd_black=NO              # set to YES to run spamd without
greylisting
shlib_dirs=                 # extra directories for ldconfig,
separated
                      # by space

# rc.d(8) packages scripts
# started in the specified order and stopped in reverse order
pkg_scripts=
```

/cfg/var/unbound/etc/unbound.conf

Στον συγκεκριμένο φάκελο βρίσκεται το αρχείο διαμόρφωσης του επιλυτή ονομάτων τομέα που χρησιμοποιεί το Σύστημα Αυθεντικοποίησης και ονομάζεται `unbound`. Παρατηρώντας την διαμόρφωση του αρχείου, φαίνονται όλα τα απαραίτητα στοιχεία που διέπουν έναν επιλυτή ονομάτων τομέα και είναι απαραίτητα για την ορθή λειτουργία του (Κεφάλαιο 2.4 Domain Name System).

Επιπλέον αποτυπώνεται η διαδικασία που εκτελείται ώστε να πραγματοποιείται ο απομακρυσμένος έλεγχος του επιλυτή ονομάτων τομέα από την μηχανή DDNS-AuthPF.

Σε αυτό το σημείο να αναφερθεί ότι η συγκεκριμένη μηχανή, μέσω της διαμόρφωσης της καθώς και της συνολικής δικτυακής διαμόρφωσης του Συστήματος Αυθεντικοποίησης, θεωρεί ότι βρίσκεται εντός εσωτερικού δικτύου και κατά συνέπεια δεν έχει επικοινωνία με τον έξω κόσμο, δηλαδή με το internet, το οποίο είναι ουσιαστικά μια παραπλάνηση για τον DNS server, διότι έχει άμεση επαφή με τον ACCESS-SRV του Συστήματος Πρόσβασης.

```
# $OpenBSD: unbound.conf,v 1.7 2016/03/30 01:41:25 sthen Exp $


server:
      root-hints: "/var/unbound/etc/named.cache"
      interface: 192.168.1.2
      #interface: 127.0.0.1@5353 # listen on alternative port
      #interface: ::1
      #do-ip6: no

      # override the default "any" address to send queries; if
multiple
```

```
      # addresses are available, they are used randomly to
counter spoofing
      outgoing-interface: 192.168.1.2
      #outgoing-interface: 2001:db8::53

      access-control: 0.0.0.0/0 allow
      access-control: ::0/0 refuse

      hide-identity: yes
      hide-version: yes

      # Uncomment to enable qname minimisation.
      # https://tools.ietf.org/html/draft-ietf-dnsop-qname-
minimisation-08
      #
      qname-minimisation: yes

      # Uncomment to enable DNSSEC validation.
      #
      #auto-trust-anchor-file: "/var/unbound/db/root.key"
      # Serve zones authoritatively from Unbound to resolver
clients.
      # Not for external service.
      #
      local-zone: "myzone." static
      local-data: "mycomputer.myzone. IN A 192.168.1.2"
      local-data: "pc01.myzone. IN A 192.168.1.100"

      #local-zone: "2.0.192.in-addr.arpa." static
      #local-data-ptr: "192.0.2.51 mycomputer.local"

      # UDP EDNS reassembly buffer advertised to peers. Default
4096.
      # May need lowering on broken networks with
fragmentation/MTU issues,
      # particularly if validating DNSSEC.
      #
      #edns-buffer-size: 1480

      # Use TCP for "forward-zone" requests. Useful if you are
making
      # DNS requests over an SSH port forwarding.
      #
      #tcp-upstream: yes

      # DNS64 options, synthesizes AAAA records for hosts that
don't have
      # them. For use with NAT64 (PF "af-to").
      #
      #module-config: "dns64 validator iterator"
      #dns64-prefix: 64:ff9b::/96# well-known prefix (default)
      #dns64-synthall: no

remote-control:
      control-enable: yes
      control-use-cert: yes
```

```
        control-interface: 172.16.100.1
        server-key-file: /var/unbound/etc/unbound_server.key
        server-cert-file: /var/unbound/etc/unbound_server.pem
        control-key-file: /var/unbound/etc/unbound_control.key
        control-cert-file: /var/unbound/etc/unbound_control.pem

# Use an upstream forwarder (recursive resolver) for specific
zones.
# Example addresses given below are public resolvers valid as
of 2014/03.
#
#forward-zone:
#     name: "."                   # use for ALL queries
#     forward-addr: 74.82.42.42        # he.net
#     forward-addr: 2001:470:20::2          # he.net v6
#     forward-addr: 8.8.8.8            # google.com
#     forward-addr: 2001:4860:4860::8888     # google.com v6
#     forward-addr: 208.67.222.222          # opendns.com
#     forward-first: yes               # try direct if
forwarder fails
```

3.3.5.2 Σύστημα Πρόσβασης

3.3.5.2.1 ACCESS-SERVER

/etc/hostname.vmx0

```
inet 192.168.168.2 255.255.255.252 NONE
```

/etc/hostname.vmx1

```
inet 10.111.111.1 255.255.255.252 NONE
```

/etc/resolv.conf

```
nameserver 192.69.69.2
```

/etc/mygate

```
192.168.168.1
```

./misc/screen[19] [multi-*screen* window manager]

Το πρόγραμμα `screen` πολυπλέκει ένα τερματικό (terminal window) σε πολλές διεργασίες. Έτσι ο διαχειριστής μπορεί να εναλλάσσεται μεταξύ των διαφορετικών προγραμμάτων που εκτελούνται χρησιμοποιώντας ένα μόνο ένα τερματικό παράθυρο.

[19] https://linux.die.net/man/1/screen

/etc/sysctl.conf

Ισχύουν ακριβώς οι ίδιες παράμετροι που αναφέρθηκαν και στο αρχείο `/etc/sysctl.conf` της μηχανής DDNS-FW.

```
#$OpenBSD: sysctl.conf,v 1.4 2015/04/03 15:50:28 millert Exp $

# This file contains a list of sysctl options the user wants
set at
# boot time. See sysctl(3) and sysctl(8) for more information
on the many available variables.
#
net.inet.ip.forwarding=1    # 1=Permit forwarding (routing) of
IPv4 packets
#net.inet.ip.mforwarding=1  # 1=Permit forwarding (routing) of
IPv4 multicast packets
#net.inet.ip.multipath=1    # 1=Enable IP multipath routing
#net.inet.icmp.rediraccept=1     # 1=Accept ICMP redirects
#net.inet6.ip6.forwarding=1# 1=Permit forwarding (routing) of
IPv6 packets
#net.inet6.ip6.mforwarding=1     # 1=Permit forwarding
(routing) of IPv6 multicast packets
#net.inet6.ip6.multipath=1  # 1=Enable IPv6 multipath routing
#net.inet.tcp.always_keepalive=1 # 1=Keepalives for all
connections (e.g. hotel/airport NAT)
#net.inet.tcp.keepidle=100  # 100=send TCP keepalives every 50
seconds
#net.inet.esp.enable=0           # 0=Disable the ESP IPsec
protocol
#net.inet.ah.enable=0       # 0=Disable the AH IPsec protocol
#net.inet.esp.udpencap=0    # 0=Disable ESP-in-UDP
encapsulation
#net.inet.ipcomp.enable=1   # 1=Enable the IPCOMP protocol
#net.inet.etherip.allow=1   # 1=Enable the Ethernet-over-IP
protocol
#net.inet.tcp.ecn=1         # 1=Enable the TCP ECN extension
#net.inet.carp.preempt=1    # 1=Enable carp(4) preemption
#net.inet.carp.log=3        # log level of carp(4) info,
default 2
#net.pipex.enable=1         # 1=Enable pipex(4) for npppd(8)
#ddb.panic=0                # 0=Do not drop into ddb on a
kernel panic
#ddb.console=1              # 1=Permit entry of ddb from the
console
#ddb.log=1            # 1=Log ddb output in kernel message
buffer
#fs.posix.setuid=0          # 0=Traditional BSD chown()
semantics
#vm.swapencrypt.enable=0    # 0=Do not encrypt pages that go to
swap
#vfs.nfs.iothreads=4        # Number of nfsio kernel threads
#net.inet.ip.mtudisc=0           # 0=Disable tcp mtu discovery
#kern.splassert=2           # 2=Enable with verbose error
messages
```

```
#kern.nosuidcoredump=3           # 3=Put suid coredumps in
/var/crash/progname
#kern.watchdog.period=32    # >0=Enable hardware watchdog(4)
timer if available
#kern.watchdog.auto=0       # 0=Disable automatic watchdog(4)
retriggering
#hw.allowpowerdown=0        # 0=Disable power button shutdown
#machdep.allowaperture=2    # See xf86(4)
#machdep.apmhalt=1          # 1=powerdown hack, try if halt -p
doesn't work
#machdep.kbdreset=1         # permit console CTRL-ALT-DEL to do
a nice halt
#machdep.lidsuspend=0       # do not suspend laptop upon lid
closing
#machdep.userldt=1          # allow userland programs to play
with ldt,
                            # required by some ports
#kern.emul.linux=1          # enable running Linux binaries
```

/etc/setdate.sh

Ισχύουν ακριβώς οι ίδιες παράμετροι που αναφέρθηκαν και στο script `setdate.sh` της μηχανής DDNS-FW.

```
#!/bin/sh
/usr/sbin/rdate -4 -n ntp.grnet.gr
```

/etc/pf.conf

Στο configuration του `/etc/pf.conf` καλούνται δύο πίνακες. Ο πρώτος είναι ο `<bruteforce>` του οποίου η λειτουργία και η χρησιμότητα του είναι ήδη γνωστές.

Ο δεύτερος πίνακας είναι ο `<ddns>` στον οποίο περιέχονται οι έγκυρες εγγραφές DNS για όλους τους δηλωμένους χρήστες ώστε να εκτελείται επιτυχώς η διαδικασία της αυθεντικοποίησης μέσω του `authpf`.

Επιπλέον εφαρμόζονται κανόνες firewall επιτρέποντας σε συγκεκριμένα πρωτόκολλα την είσοδο και έξοδο τους, εκτελώντας παράλληλα NAT.

```
table <bruteforce> persist
table <ddns> persist

set skip on lo

block quick from <bruteforce>

anchor "authpf/*"
```

```
pass in quick on vmx0 proto tcp from <ddns> to any port 65122 \
     flags S/SA keep state (max-src-conn-rate 3/30, overload \
     <bruteforce> flush global)

pass out quick on vmx0 proto {tcp, udp} from any to any nat-to
(vmx0)

block in on vmx0
pass on vmx1
```

/root/sshconn.sh

Το script `/root/sshconn.sh` καλεί το παρακάτω script `/root/sshconn.sh1` εντός του προγράμματος `screen`.

```
#!/bin/sh
/usr/local/bin/screen -dmS test /root/sshconn1.sh
```

/root/sshconn1.sh

```
#!/bin/sh
while [ true ]; do
/usr/bin/ssh ddns
sleep 5
done
```

/root/whoisin.sh

Το script `/root/whoisin.sh` εμφανίζει τις ενεργές συνδέσεις `authpf`.

```
#!/bin/sh
/bin/ps -ax | /usr/bin/grep authpf
```

/root/checkAddress.sh

Στο script `root/checkAddress.sh` εκτελείται η διαδικασία ελέγχου της σύνδεσης.

Οι εγγραφές DNS πραγματοποιούνται στο βασικό αρχείο που ονομάζεται `/root/ddns.ips`. Ουσιαστικά πρόκειται για έναν κατάλογο των εγγραφών DNS που είναι ενεργές και έχουν εκτελέσει επιτυχώς της διαδικασία του authpf σε συνεργασία πάντα με το Σύστημα Αυθεντικοποίησης.

```
#!/bin/sh
# 1. Step would be to check icmp connectivity. In this case
this is
#not possible due to firewall rules
# 2. check if nslookup is running. If yes exit
```

```
lockdir="/tmp/test.test"
if mkdir "$lockdir"
then
      trap 'rm -rf "$lockdir"' 0
else
      exit 112
fi

# the following is our base file
input="/root/ddns.ips"

# delete temporary file
/bin/rm -f /tmp/recs.data

# read the base file line by line
while IFS= read -r var
do
records=$(nslookup $var - 192.168.69.2 | /usr/bin/grep -A1
'Name:' | /usr/bin/grep Address | /usr/bin/awk -F': ' '{print
$2}')

if [ -z $records ]
then
/bin/echo "$var" > /dev/null
else
/bin/echo $records >> /tmp/recs.data
fi

done < "$input"

touch /tmp/recs.data

if [ -z /tmp/recs.data ]
then
/bin/echo "test" > /dev/null
else
/sbin/pfctl -t ddns -T flush
/sbin/pfctl -t ddns -T add -f /tmp/recs.data
fi
```

3.3.5.2.2 ACCESS-PFSENSE

Η συγκεκριμένη μηχανή έχει εγκατεστημένο το πρόγραμμα `pfsense` μέσα από το οποίο έχει ρυθμιστεί κατάλληλα το λογισμικό του `OpenVPN`[20], ώστε να εκτελεστεί επιτυχώς το βήμα 7 που περιγράφεται στην αρχή λειτουργίας του συνολικού συστήματος.

Διαθέτει τρείς κάρτες δικτύου, από τις οποίες η πρώτη είναι υπεύθυνη για τη διασύνδεση με τον ACCESS-SRV, δηλαδή με την εισερχόμενη κίνηση (`lan` - /etc/hostname.em0), η δεύτερη με την εξερχόμενη κίνηση (`wan` –

[20] https://docs.netgate.com/pfsense/en/latest/recipes/openvpn-ra.html#configuring-openvpn-server-settings

/etc/hostname.em1) προς τον target και η τρίτη κάρτα (mngt – /etc/hostname.em2) χρησιμοποιείται για την διαχείριση της μηχανής από τον διαχειριστή.

/etc/hostname.em0

```
inet 10.111.111.2 255.255.255.252 NONE
```

/etc/hostname.em1

```
inet 10.112.112.2 255.255.255.252 NONE
```

/etc/hostname.em2

```
inet 172.16.222.1 255.255.255.0 NONE
```

4. Επίλογος

Στη παρούσα διπλωματική εργασία εξετάστηκε η διαδικασία απομακρυσμένης πρόσβασης μεταξύ client και server και χρησιμοποιήθηκε στο σύνολο της λογισμικό ανοιχτού κώδικα. Ο συνδυασμός της χρήσης του λογισμικού, με τη μεθοδολογία που ακολουθήθηκε και περιγράφτηκε βήμα προς βήμα, βοήθησε ώστε να υλοποιηθεί ένα πρωτότυπο σύστημα απομακρυσμένης πρόσβασης, το οποίο συμβάλλει στην άμβλυνση του τεράστιου ζητήματος ασφαλείας που υφίσταται στη σύγχρονη εποχή.

Το σύστημα αποτελεί προϊόν πνευματικής ιδιοκτησίας, το οποίο και προσφέρεται για τη χρήση του ελεύθερα στο κοινό.

5. Βιβλιογραφία - Αναφορές

[1] The OpenBSD Project

https://www.openbsd.org/

[2] OpenBSD manual page server

https://man.openbsd.org/

ssh(1) - pf.conf(5) - sysctl.conf(5) - crontab(5) - authpf(8) - hostname.if(5) - resolv.conf(5) - mygate(5) - rc.conf(8) - unbound.conf(5) - sqlite3(1) - gethostbyname(3) - gethostbyaddr(3) -

[3] Daniel J. Barrett, Robert G. Byrnes, Richard, E. Silverman (May 2005), **SSH, the Secure Shell, 2nd Edition,** O'Reilly Media
(ISBN: 0-596-00895-3)

[4] Michael W. Lucas (2013), **Absolute OpenBSD, 2nd Edition**
(ISBN-10: 1-59327-476-9, ISBN-13: 978-1-59327-476-4)

[5] Peter N.M. Hansteen (2015), **The Book of PF, 3rd Edition**
(ISBN-10: 1-59327-589-7, ISBN-13: 978-1-59327-589-1)

[6] Peter N.M. Hansteen (2005 - 2017), **Firewalling with OpenBSD's PF packet filter**
https://home.nuug.no/~peter/pf/en/long-firewall.html

[7] Allan Liska, Geoffrey Stowe, Timothy Gallo, Technical Editor (2016), **DNS Security, Defending the Domain Name System**
(ISBN: 978-0-12-803306-7)

[8] W. Richard Stevens, **TCP/IP Illustrated: The Protocols**
(ISBN: 0-201-63346-9)

[9] ssh.com – OpenSSH

https://www.ssh.com/academy/ssh/openssh

[10] OpenSSH for OpenBSD

https://www.openssh.com/openbsd.html

[11] Monit - utility for monitoring services on a Unix system

https://mmonit.com/monit/documentation/monit.html

[12] Screen [multi-*screen* window manager]

https://linux.die.net/man/1/screen

[13] pfsense, Open Source Security, Copyright 2021 Electric Sheep Fencing, LL, All Rights Reserved

https://www.pfsense.org/

[14] Unbound, a validating, recursive, caching DNS resolver, Copyright 2021 Stichting NLnet Labs, Amsterdam, The Netherlands

https://nlnetlabs.nl/projects/unbound/about/

[15] Unbound DNS server

https://en.wikipedia.org/wiki/Unbound_(DNS_server)

[16] Best books online library Flylib.com

Authenticating PF - https://flylib.com/books/en/2.717.1.258/1

[17] OpenBSD - flashdr

https://www.nmedia.net/flashrd/

[18] Resilient OpenBSD images for flash memory - reshflash

https://stable.rcesoftware.com/resflash/

[19] VMware Documentation

https://www.vmware.com/support/pubs/

[20] Vmware ESXi Installation and Setup

https://docs.vmware.com/en/VMware-vSphere/7.0/vsphere-esxi-70-installation-setup-guide.pdf

[21] PC Engines, System Board APU4c4

https://www.pcengines.ch/apu4c4.htm