Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών:
Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Bootable Linux Distribution με Φιλικό Γραφικό Περιβάλλον για Δοκιμές Διείσδυσης

Εμμανουήλ Γιατρομανωλάκης

Επιβλέπουσα Καθηγήτρια
Αδαμαντίνη Περατικού

Μάιος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών
Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Bootable Linux Distribution με Φιλικό Γραφικό Περιβάλλον για Δοκιμές Διείσδυσης

Εμμανουήλ Γιατρομανωλάκης

Επιβλέπων Καθηγητής
Αδαμαντίνη Περατικού

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στην Ασφάλεια Υπολογιστών και Δικτύων από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Μάιος 2019

Περίληψη
Την τελευταία δεκαετία παρατηρείται μια τεραστία ανάπτυξη των συστημάτων πληροφορικής και μια ραγδαία εξέλιξη των παρεχόμενων υπηρεσιών που στηρίζονται σε αυτά. Μεγάλες εταιρείες και οργανισμοί στηρίζουν την ύπαρξη τους στο διαδίκτυο και τις υπηρεσίες που παρέχουν σε καθημερινή βάση στους χρήστες. Η παραμικρή διακοπή η απώλεια δεδομένων μπορεί να επιφέρει μεγάλες απώλειες σε ένα οργανισμό. Συνεπώς η ασφάλεια των πληροφοριακών συστημάτων είναι ένας κρίσιμος παράγοντας για την βιωσιμότητα ενός οργανισμού η μιας εταιρείας.

Η παρούσα μεταπτυχιακή διατριβή έχει ως σκοπό την μελέτη και την αξιολόγηση των δοκιμών διείσδυσης ως μέσο ανίχνευσης πιθανών ευπαθειών στα πληροφοριακά συστήματα ενός οργανισμού. Η διατριβή απαρτίζεται από δυο μέρη, το θεωρητικό όπου περιγράφεται η διαδικασία ελέγχου ευπαθειών μέσω εργαλείων σάρωσης και το πρακτικό που επικεντρώνεται στην ανάπτυξη φιλικής εφαρμογής (GUI) με στόχο την εύκολη διενέργεια ελέγχου τρωτότητας σε ένα πληροφοριακό σύστημα.

Η εφαρμογή αναπτύχτηκε με γνώμονα την ευχρηστία και την φορητότητα έτσι ώστε να μπορούμε να εκτελέσουμε άμεσα δοκιμές διείσδυσης. Η υλοποίηση της εφαρμογής ακολούθησε το μοντέλο ανάπτυξης λογισμικού καταρράκτης και δοκιμάστηκε σε πραγματικό περιβάλλον μεγάλου οργανισμού με πολλά συστήματα και δικτυακές υποδομές. Μελετώντας τα αποτελέσματα των δοκίμων σάρωσης μπορούμε να αξιολογήσουμε την ασφάλεια ενός πληροφοριακού συστήματος και να διορθώσουμε τις ευπάθειες του με κύριο στόχο την μείωση των απωλειών σε περίπτωση μη εξουσιοδοτημένης πρόσβασης.

Τέλος οδηγηθήκαμε στο συμπέρασμα ότι η διαδικασία σάρωσης και διείσδυσης για την ανίχνευση των ευπαθειών είναι μια πρόκληση για κάθε οργανισμό και πρέπει να εκτελείται μεθοδικά και επαναλαμβανόμενα, επίσης πρέπει να αναπτύσσεται και να προσαρμόζεται στα νέα λογισμικά, στα νέα πρωτόκολλα και να λαμβάνει υπόψη τις νέες τεχνικές που χρησιμοποιούν οι κακόβουλοι χρήστες ώστε να προσαρμόζεται ανάλογα.

Summary
Over the last decade, there has been a tremendous growth in IT systems and a rapid development of the services that rely on them. Large companies and organizations support their existence on the internet and the services they provide on a daily basis to users. The slightest crash or loss of data can cause great losses to an organism. Therefore, the security of information systems is a critical factor for the viability of an organization or a company.

This master's thesis aims at studying and evaluating penetration probes as a means of detecting potential vulnerabilities in an organization's information systems. The thesis consists of two parts, the theoretical part where the process of testing vulnerabilities through scanning tools and the practice part focused on the development of a friendly application (GUI) aiming at the easy conduct of vulnerability control in an information system.

The application has been developed with usability and portability so that we can immediately perform penetration tests. Implementation of the application followed the waterfall software development model and tested in a real environment of a large organization with many systems and network infrastructures. By studying the results of penetration tests, we can evaluate the security of an information system and fix its vulnerabilities, with the main goal of reducing losses in the unauthorized access event.

Finally, the conclusion of the scanning and penetration process for detecting vulnerabilities is a challenge for each organization and must be performed methodically and repeatedly, also developed and adapted to new software, new protocols, including new techniques that malicious users use to adapt accordingly.

Ευχαριστίες
Θα ήθελα να ευχαριστήσω την καθηγήτρια μου Κ. Περατικού για την ανάθεση της διατριβής και την δυνατότητα που μου δόθηκε να ερευνήσω και να ασχοληθώ με τον σημαντικό τομέα της ασφάλειας δικτύων και συστημάτων.

Να ευχαριστήσω ακόμα το Ίδρυμα Τεχνολογίας και Έρευνας για την παροχή του πειραματικού περιβάλλοντος και την στήριξη που μου παρείχε στην υλοποίηση της.

Τέλος να ευχαριστήσω την οικογένεια μου για την πολύτιμη ηθική υποστήριξη τους καθ' όλη την διάρκεια των μεταπτυχιακών μου σπουδών.

Περιεχόμενα

Πίνακας εικόνων

Κεφάλαιο 1
Εισαγωγή

Η μεταπτυχιακή διατριβή αυτή ασχολείται με τις δοκιμές διείσδυσης και την ανίχνευση των αδυναμιών στα πληροφοριακά συστήματα μέσω γραφικού περιβάλλοντος. Αποτελείται από δυο μέρη το θεωρητικό και το πρακτικό.

Στο θεωρητικό μέρος αναλύονται και παρουσιάζονται ορισμοί που αφορούν την ασφάλεια πληροφοριακών συστημάτων, μέθοδοι σάρωσης μέσω των δοκιμών διείσδυσης και εντοπισμός πιθανών αδυναμιών/απειλών που μπορούν να αποτελέσουν σημείο αδυναμίας για την ασφάλεια των συστημάτων. Στην συνέχεια καταγράφονται τα εργαλεία (penetration tools) που χρησιμοποιούνται, οι αδυναμίες τους και η πιθανή βελτίωση τους.

Στο πρακτικό μέρος υλοποιείται ένα bootable live cdrom με Linux όπου περιέχονται τα εργαλεία που θα χρησιμοποιήσουμε για τις δοκιμές, μια εφαρμογή με πρόσθετες λειτουργιές εκτέλεσης τους παρουσιάζοντας στο χρήστη τα αποτελέσματα με γραφικό περιβάλλον (GUI).

1.1 Ερευνητικά Ερωτήματα

1) Ποια είναι τα χαρακτηριστικά ασφάλειας που πρέπει να πληρούν τα υπολογιστικά συστήματα;
2) Ποιες είναι οι απειλές συστημάτων/δικτύων και πως ανιχνεύονται (penetration tests);
3) Εργαλεία ανίχνευσης/πρόληψης απειλών/vulnerabilities (NMap, dipiscan, winmtr, advanced ip scanner).
4) Ποιες είναι οι ανάγκες για την βέλτιστη και εύκολη λειτουργία των παραπάνω εργαλείων;

1.2 Μεθοδολογία

Αρχικά στο θεωρητικό μέρος θα γίνει μια εκτενή αναφορά στα συστήματα και στις έννοιες ασφάλειας και ποια είναι τα οφέλη και οι μέθοδοι ανάλυσης της επικινδυνότητας των αδυναμιών [1]. Στην συνέχεια θα οριστούν οι απαιτήσεις ασφαλείας που είναι αναγκαίες για την εύρυθμη λειτουργία ενός οργανισμούς/επιχείρησης και τι διαδικασίες επιβάλλονται έτσι ώστε να ελαχιστοποιεί το ρίσκο μιας εισβολής [2].

Θα γίνει αναφορά στις απειλές, στον τρόπο λειτουργίας των εισβολέων , ποιους μεθόδους χρησιμοποιούν , ποια τα είδη κινδύνων και πως μπορούμε να τα διαχειριστούμε. [3]. Ποιο αναλυτικά θα διερευνήσουμε τον τρόπο με τον οποίο οι κακόβουλοι χρήστες χρησιμοποιούν τις δοκιμές σάρωσης για να εισβάλουν στα πληροφοριακά συστήματα ενός οργανισμού. θα καταγράψουμε τα είδη κινδύνων και τις απώλειες που επιφέρουν και τέλος με ποιον τρόπο μπορούμε να ελαχιστοποιήσουμε τους κινδύνους και τις απώλειες που προκαλούν.

Μετά θα αναλυθούν οι δοκιμές διείσδυσης/σάρωσης, ποιοι οι στόχοι τους, με τη κριτήρια αξιολογούνται , πως λειτουργούν σε επίπεδο δικτυού έτσι ώστε να αποκαλύψουν τις ευπάθειες των συστημάτων και τέλος θα αναφερθούν τα ποιο γνωστά εργαλεία διείσδυσης και οι τεχνικές, μέθοδοι σάρωσης [4] [5] [6] [7].

Για την αξιόπιστη και αποτελεσματική λειτουργία των εργαλείων θα πρέπει να αναφερθούν οι αδυναμίες τους, με ποιο τρόπο οι εισβολείς καταφέρνουν να παρακάμψουν την ασφάλεια ενός πληροφορικού συστήματος από αδυναμίες που τα εργαλεία δεν ανακαλύπτουν. Ποια εργαλεία θα επιλέξουμε να χρησιμοποιήσουμε και πως μπορούμε να τα βελτιώσουμε έτσι ώστε να αυξήσουμε την αποτελεσματικότητα τους [8].

Στο πρακτικό μέρος και εφόσον έχουμε επιλέξει τα εργαλεία και τις τεχνικές σάρωσης/διείσδυσης , θα καταγράψουμε τους τρόπους ανάλυσης των αδυναμιών και τους παραμέτρους που θα χρησιμοποιήσουμε για τις δοκιμές μας [9] [10]. Στην συνέχεια θα δημιουργηθεί ένα bootable Linux cdrom και συγγράφοντας shell scripts [11] θα δημιουργήσουμε τα εκτελέσιμα αρχεία για το κάθε είδους σάρωσης/διείσδυσης [12] [3] [4]. Θα εκτελέσουμε αρκετά τεστ σε πραγματικό δίκτυο έτσι ώστε να καταγράφουν τα αποτελέσματα και να αποκαλυφθούν όσο το δυνατό περισσότερες αδυναμίες εκτελώντας διαφορετικούς μεθόδους σάρωσης [6]. Τέλος χρησιμοποιώντας την γλωσσά προγραμματισμού python [13], τις γραφικές βιβλιοθήκες που περιέχει και το μοντέλο ανάπτυξης λογισμικού καταρράκτης ετσι ώστε να δημιουργήσουμε ένα φιλικό γραφικό περιβάλλον χρήσης μέσω του οποίου θα εκτελούνται οι δοκιμές διείσδυσης και θα παρουσιάζονται στο χρήστη σε μορφή αναφοράς.

Στο τέλος της διατριβής θα καταγραφούν τα συμπεράσματα και η αποτελεσματικότητα του cdrom καθώς και η σύγκριση μεταξύ των command line εργαλείων με το GUI που δημιουργήσαμε.

1.3 Βιβλιογραφική Ανασκόπηση

Η ασφάλεια των πληροφοριακών συστημάτων είναι καίριας σημασίας για την ανάπτυξη ενός οργανισμού/επιχείρησης. Ασφάλεια όπως αναφέρεται στο βιβλίο [1] είναι ο κλάδος της πληροφορικής που ασχολείται με την προστασία των δεδομένων και των πληροφοριακών συστημάτων με κύριο στόχο την προστασία τους. Βασική αρχή για την σωστή προστασία είναι η ανάλυση επικινδυνότητας [7] έτσι ώστε να καταγράφουν οι ευπάθειες και οι αδυναμίες που χαρακτηρίζονται κίνδυνοι για τα συστήματα/δεδομένα και στη συνεχεία να παρθούν τα αναγκαία μέτρα για την προστασίας τους.

Για τον εντοπισμό ευπαθειών στα πληροφοριακά συστήματα χρησιμοποιείται μια διαδικασία που αναφέρεται ως port scanning. Όπως περιγράφεται στο άρθρο [7] port scanning είναι ο έλεγχος των ανοιχτών θυρών στα συστήματα έτσι ώστε να αναγνωριστούν οι αδυναμίες τους. Στο άρθρο [8] περιγράφεται η παραπάνω διαδικασία χρησιμοποιώντας το πρόγραμμα snort το όποιο με κατάλληλους παραμέτρους ανιχνεύει όλες τις ανοιχτές θύρες των συστημάτων που μπορούν να αποτελέσουν απειλή για την ασφάλεια τους.

Η επόμενη διαδικασία ανίχνευσης των αδυναμιών είναι οι δοκιμές διείσδυσης . Στο άρθρο [4] ορίζεται η διαδικασία penetration testing ως μια δοκιμή ελέγχου ασφάλειας όπου οι διαχειριστές μιμούνται τις επιθέσεις που θα εκτελούσαν οι κακόβουλοι χρήστες για να αναγνωρίσουν ευπάθειες που θα χρησιμοποιήσουν για να εισβάλουν σε ένα σύστημα. Ο στόχος της παραπάνω διαδικασίας είναι η ανίχνευση των αδυναμιών ασφάλειας πριν τις ανακαλύψουν οι εισβολείς.

Στα sites [9] [10] και στο άρθρο [5] οι συγγράφεις περιγράφουν δέκα εργαλεία ανοιχτού λογισμικού που χρησιμοποιούνται από τους εισβολείς/διαχειριστές για την ανίχνευση των ευπαθειών. Πιο αναλυτικά κατηγοριοποιούν τα εργαλεία σε 3 κατηγορίες: White,Black,Grey box testing's, ανάλογα την πρόσβαση στα συστήματα που θα αναλυθούν. Στο άρθρο [6] ο Hui Liu περιγράφει μια διαδικασία black box testing μέσω της οποίας καταφέρνει να μειώσει τον χρόνο της διαδικασίας και να ανιχνεύσει αδυναμίες χωρίς να γίνει αντιληπτός.

Στο άρθρο [12] ο συγγραφέας χρησιμοποιεί το εργαλείο NMap και πέντε διαφορετικές τεχνικές για να αποκαλύψει τις αδυναμίες ενός συστήματος. Ποιο αναλυτικά περιγράφει τους παραμέτρους του εργαλείου έτσι ώστε να εκτελέσει ένα πλήρη έλεγχο σε όλα τα ενεργά στοιχειά του δικτυού και των πληροφοριακών συστημάτων σχεδιάζοντας τον χάρτη τοπολογίας δικτυού.

Ο Philip Bosco στο άρθρο [3] χρησιμοποίει τεχνικές black box έτσι ώστε να αποκτήσει πρόσβαση στο εσωτερικό δίκτυο και στην συνεχεία εκτελεί white box testing's για την ανιχνεύσει των εσωτερικών αδυναμιών. Η παραπάνω τεχνική χρησιμοποιείται από τους εισβολείς για τον πλήρη έλεγχο των συστημάτων, συνεπώς η συγκεκριμένη τακτική σύμφωνα με το συγγραφέα κρίνεται απαραίτητη.

Στο άρθρο [14] περιγράφονται αναλυτικά 15 εργαλεία δοκιμών διείσδυσης, αναφέροντας άδεια χρήσης, λειτουργικό σύστημα και τι μεθόδους και τεχνικές χρησιμοποίει το καθένα για την βέλτιστη ανάλυση των συστημάτων.

Τέλος στο άρθρο [2] περιγράφονται τυποποιημένες δοκιμές διείσδυσης και ένας καλύτερος τρόπος προσεγγίσεις των δοκιμών για την μεγίστη απόδοση τους και την ανιχνεύσει των ευπαθειών μειώνοντας τον χρόνο και την δυσκολία αναζήτησης των παραμέτρων στα εργαλεία εκτέλεσης τους.

Α. Θεωρητικο Μερος

Κεφάλαιο 2
Ασφάλεια Πληροφοριακών συστημάτων

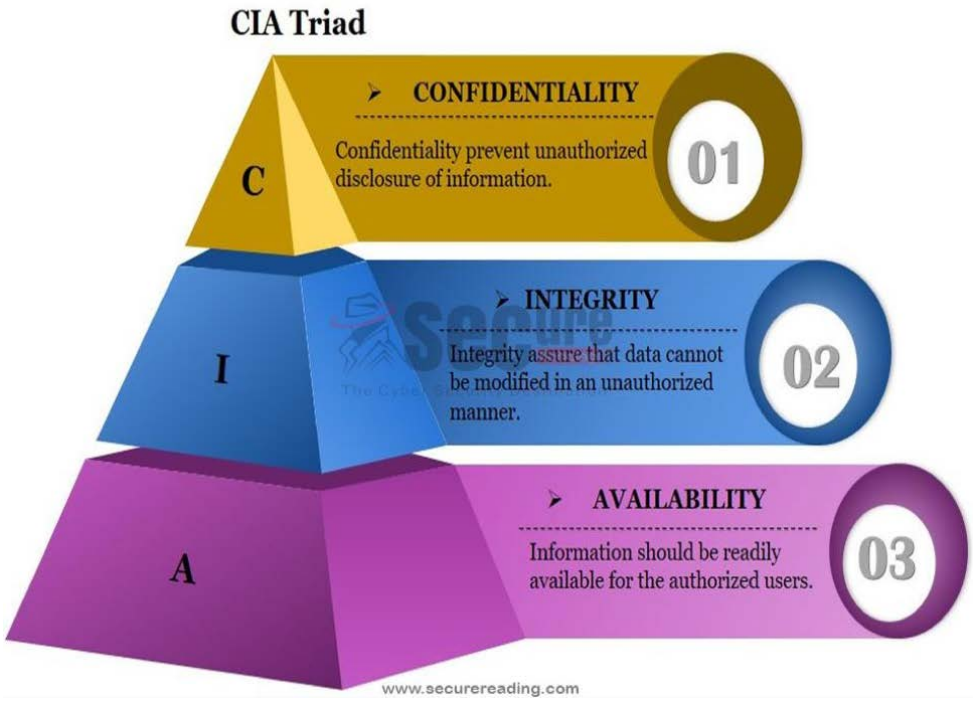
2.1 Βασικά χαρακτηριστικά ασφάλειας

Στο κεφάλαιο αυτό γίνεται μια αναφορά στις βασικές αρχές ασφάλειας και πως αυτές επηρεάζουν ένα πληροφοριακό σύστημα. Ποιο αναλυτικά περιγράφονται έννοιες όπως οι βασικές ιδέες της ασφάλειας, τι εννοούμε με τον ορό επικινδυνότητα, από ποιες απειλές κινδυνεύει ένα σύστημα και πως ταξινομούνται. Τέλος περιγράφεται η ανάλυση επικινδυνότητας, τα αντίμετρα που πρέπει να ληφθούν και τα οφέλη που επιφέρουν σε ένα οργανισμό/επιχείρηση.

2.1.1 Βασικές Αρχές

Με τον όρο ασφάλεια πληροφοριακών συστημάτων [15] αναφερόμαστε στο κλάδο της επιστήμης πληροφορικής που αφορά την προστασία των υπολογιστικών συστημάτων καθώς και των δικτύων που τα συνδέουν. Τα δομικά στοιχεία που συνθέτουν τα πληροφοριακά συστήματα είναι η σύνδεση και τα δεδομένα. Ο βασικός στόχος της ασφάλειας είναι η αποτροπή μη εξουσιοδοτημένης πρόσβασης σε δεδομένα η χρήση αυτών.

Η ασφάλεια των πληροφοριακών συστημάτων στηρίζεται σε τρεις βασικές ιδέες [16] [17].



Εικόνα 2-1 – CIA [16]

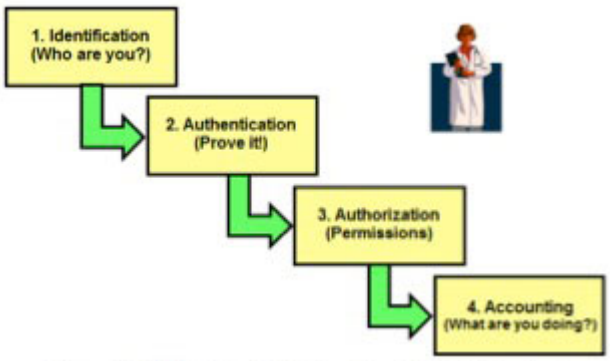
Στην εικόνα 1 αναφέρετε η τριάδα CIA, ποιο αναλυτικά έχουμε:

- **Εμπιστευτικότητα (Confidentiality)** : αφορά την μη εξουσιοδοτημένη πρόσβαση σε δεδομένα χωρίς να τροποποιείται (διαγραφή η αλλοίωση) η κατάσταση τους. Ένα σύστημα που παρέχει εμπιστευτικότητα προφυλάσσει από την αποκάλυψη των πληροφοριών σε μη εξουσιοδοτημένους χρήστες. Αυτό γίνεται με μηχανισμούς προστασίας που

πραγματοποιούν ελέγχους πριν την πρόσβαση στη πληροφορία. Η εμπιστευτικότητα επιτυγχάνεται με τον έλεγχο πρόσβασης καθώς και με την κρυπτογράφηση των δεδομένων.

- **Ακεραιότητα (Integrity)** : αφορά την προστασία των δεδομένων που υπάρχουν η παράγονται από μη εξουσιοδοτημένη τροποποίηση η αλλοίωση τους. Στη βιβλιογραφία ο όρος ακεραιότητα δεδομένων ταυτίζεται με την ιδιότητα της αυθεντικότητας εννοώντας ότι τα δεδομένα παραμένουν στην αρχική τους κατάσταση χωρίς ανεπιθύμητες τροποποιήσεις, προσθήκες η αφαιρέσεις από μη εξουσιοδοτημένους χρήστες. Η ακεραιότητα των δεδομένων επιτυγχάνεται κάνοντας χρήση μηχανισμών αυθεντικοποίησης, ελέγχου πρόσβασης η με ψηφιακές υπογραφές που διασφαλίζουν ότι η πληροφορία δεν θα αλλοιωθεί.
- **Διαθεσιμότητα (Availability)**: αφορά την αδιάλειπτη διαθεσιμότητα των πόρων, των υπηρεσιών, και των δεδομένων ενός πληροφοριακού συστήματος στους εξουσιοδοτημένους χρήστες του. Στόχος της διαθεσιμότητας είναι η έγκαιρη διάθεση των δεδομένων χωρίς προβλήματα καθυστέρησης. Η μη επίτευξη του στόχου μπορεί να οφείλεται σε προβλήματα της υπηρεσίας λόγω ανεπαρκείας των πόρων, πχ υπερβολικά μεγάλη απαίτηση δεδομένων με αποτέλεσμα το σύστημα να μην είναι διαθέσιμο η από μια εχθρική απειλή που στοχεύει στην κατάρρευση των συστημάτων προσωρινά η μόνιμα (DDos attack).

Σήμερα με την εκτεταμένη χρήση της πληροφορικής έχει διαπιστωθεί ότι δεν επαρκούν οι παραπάνω τρεις ιδιότητες έτσι ώστε να προσδιοριστεί με σαφήνεια η έννοια της ασφάλειας των πληροφοριακών συστημάτων.



Εικόνα 2-2 - Access Control Process [18]

Επιπρόσθετες ιδιότητες [19] είναι :

- **Ταυτοποίηση (Identification)**: η διαδικασία αναγνώρισης μιας οντότητας.
- **Αυθεντικοποίηση (Authentication)**: η διαδικασία κατά την οποία επιβεβαιώνεται η ταυτότητα μιας οντότητας από μια άλλη.
- **Εξουσιοδότηση (Authorization)**: μετά την ταυτοποίηση και την αυθεντικοποίηση παρέχεται το δικαίωμα πρόσβασης σε μια υπηρεσία, δεδομένα η αντικείμενα.
- **Απονομή ευθυνών (Accountability)**: η απόδειξη της αναγνώρισης μιας οντότητας και η υπευθυνότητα των πράξεων της.

2.1.2 Επικινδυνότητα

Επικινδυνότητα (Risk) είναι το γινόμενο της επίπτωσης και του απομένοντα κινδύνου. Συνεπώς για την μείωση των επιπτώσεων θα πρέπει να γίνει η ανάλυση επικινδυνότητας έτσι ώστε να

καταγράφουν οι ευπάθειες και οι απειλές των πληροφοριακών συστημάτων και να διαμορφωθεί η πολιτική ασφαλείας.

Έχουμε τους παρακάτω βασικούς ορούς για τον υπολογισμό του ρίσκου στην ανάλυση κινδύνων:

- **Απειλή (Threat):** είναι ένα μη επιθυμητό γεγονός το οποίο μπορεί να είναι τυχαίο η σκόπιμο που μπορεί να προκαλέσει την κατάρρευση του συστήματος με απώτερο σκοπό την μη διάθεση των υπηρεσιών/δεδομένων η την μη εξουσιοδοτημένη αποκάλυψη/αλλοίωση πληροφοριών. Παραδείγματα απειλών είναι : ιοί, εισβολείς, φυσικές καταστροφές, εγκληματικές ενέργειες και τα λάθη του προσωπικού.
- **Ευπάθεια (Vulnerability):** ορίζεται ως το γινόμενο της πιθανότητας να συμβεί μια απειλή με την πιθανότητα να είναι επιτυχής. Ευπάθειες είναι η αδυναμίες , η σχεδιαστικές ατέλειες ενός λογισμικού, η απουσία ενημερώσεων και γενικά οτιδήποτε μπορεί να επιφέρει την παραβίαση της ασφάλειας και την απώλεια της ακεραιότητας του συστήματος.
- **Επιπτώσεις (Effect):** Οι τυχαίες – ανεπιθύμητες καταστάσεις που δημιουργούνται μετά την πραγματοποίηση μιας απειλής η οποία βασίστηκε σε μια αδυναμία του πληροφοριακού συστήματος. Παραδείγματος χάρη διέρρευσε το αρχείο των κωδίκων λόγω προγραμματιστικού λάθους και το σύστημα παραβιάστηκε.
- **Αντίμετρο:** είναι το μέτρο που λαμβάνεται για την αντιμετώπιση των απειλών και την προστασία των συστημάτων. Χρησιμοποιείται σε όλα τα στάδια που αφορούν πρόληψη, ανίχνευση και εντέλει μείωση της απώλειας σε περίπτωση εμφάνισης απειλών.
- **Αγαθά (Asset):** τα περιουσιακά στοιχεία του οργανισμού/επιχείρησης που περιλαμβάνουν δεδομένα, πληροφορίες, hardware, ανθρώπους, πόρους, κλπ.

Εικόνα 2-3 - Risk Management [20]

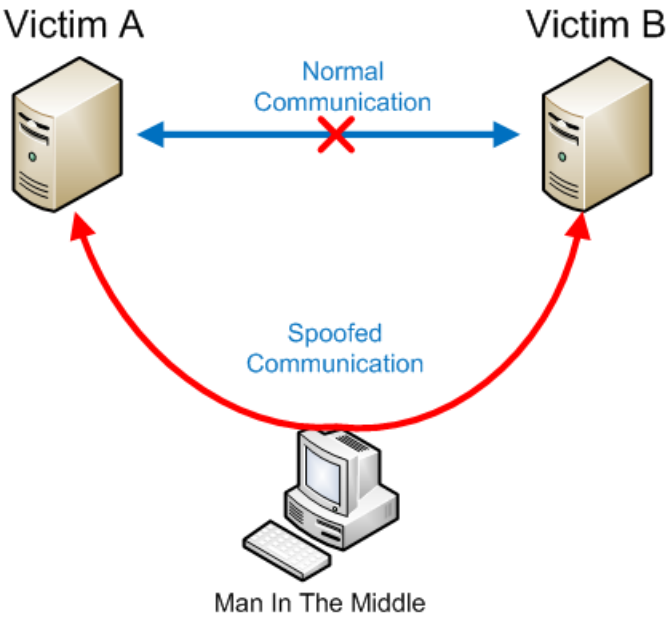
2.1.3 Ταξινόμηση απειλών

Στη βιβλιογραφία συναντάμε δυο ινστιτούτα που ταξινομούν τις απειλές των πληροφοριακών συστημάτων, το ινστιτούτο SANS [21] και το ινστιτούτο NIST [20]. Και τα δυο ινστιτούτα ταξινομούν

τις απειλές σύμφωνα με την τριάδα CIA την οποία περιγράψαμε στο κεφάλαιο 2.1.1 , άρα η ταξινόμηση γίνεται σύμφωνα με την βασική ιδέα που επηρεάζουν.

Συνεπώς έχουμε:

Απειλές κατά του Confidentiality: κύριος στόχος των συγκεκριμένων απειλών είναι η συλλογή πληροφοριών ώστε ο εισβολέας να γίνει γνώστης αυτών χωρίς την αποκάλυψη του και η μελλοντική χρήση τους για την επίθεση στο σύστημα. Η ποιο γνωστή μορφή επίθεσης είναι η επίθεση του ενδιάμεσου (MITM) κατά την οποία ο εισβολέας παρεμβάλλεται σε μια επικοινωνία και συλλέγει δεδομένα χωρίς να γίνεται αντιληπτός από τους χρήστες.



Εικόνα 2-4 – MITM (Source: netcraft)

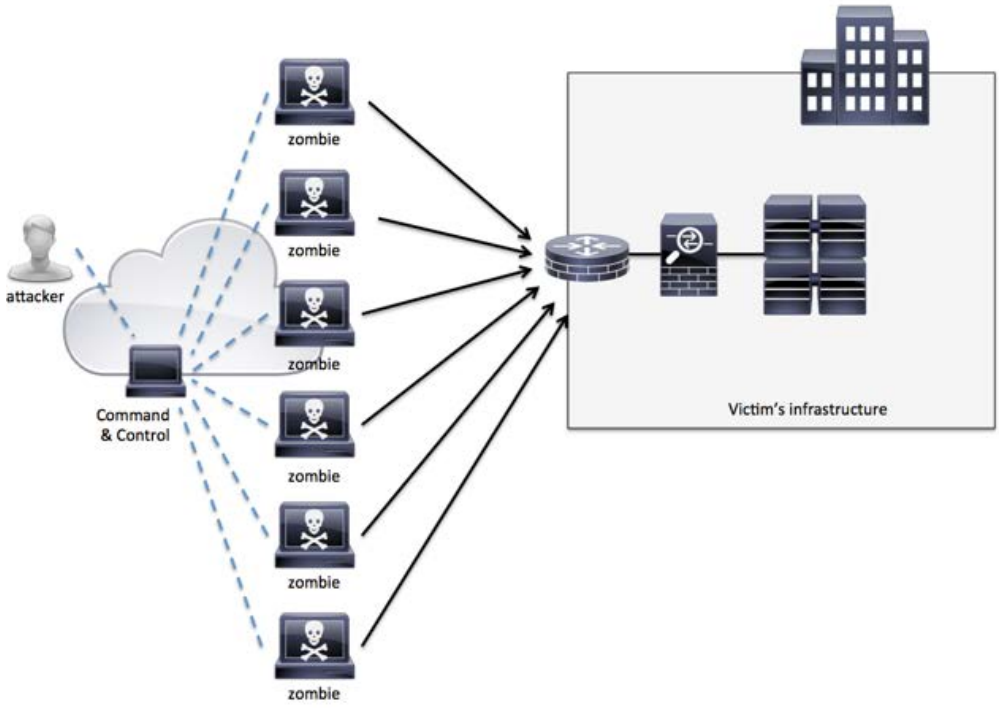
Άλλος τρόπος εισβολής είναι η εισαγωγή κάποιου ιού ο οποίος εκτελείται στο background υποκλέπτοντας πληροφορίες από τον χρήστη, αυτό μπορεί να γίνει είτε σε επίπεδο λογισμικού (Spyware, Malware) είτε σε επίπεδο δικτύου (Sniffers) καταγράφοντας μεγάλο όγκο δεδομένων που περιέχουν ευαίσθητες πληροφορίες (Passwords) και χρησιμοποιώντας σε δεύτερο χρόνο για την παραβίαση του συστήματος.

Άλλος τρόπος επίθεσης κατά της εμπιστευτικότητας είναι η χρήση data mining. Με αυτό τον τρόπο ο εισβολέας συλλέγει πληροφορίες για το σύστημα που θα επιτεθεί μέσω των χρηστών του, έτσι μπορεί να συλλέξει usernames, διευθύνσεις ηλεκτρονικού ταχυδρομείου η ακόμα και πληροφορίες που μπορεί να είναι μέρος των κωδίκων όπως ημερομηνία γέννησης, διεύθυνση, αριθμό αυτοκίνητου , κλπ.

Απειλές κατά του Integrity: κύριος στόχος των απειλών κατά της ακεραιότητας είναι η τροποποίηση και η αλλοίωση των δεδομένων. Ο εισβολέας για να επιτύχει τέτοιου τύπου απειλές πρέπει πρώτα να έχει εισβάλει στο σύστημα και να παρακολουθεί επικοινωνίες η ροές δεδομένων, στη συνέχεια τροποποιεί δεδομένα και τα αποστέλλει στον προορισμό τους χωρίς να γίνει αντιληπτός από τους χρήστες. Απειλές κατά της ακεραιότητας είναι οποιαδήποτε απειλή προκαλεί τροποποίηση των δεδομένων τα οποία αρχικά έχουν προσπελαστεί παράνομα. Παραδείγματα τροποποίησης είναι οι μεταβολές δεδομένων σε βάσεις, η αλλοίωση προγραμμάτων έτσι ώστε να εξυπηρετούν

παράνομους σκοπούς και η τροποποίηση του υλικού έτσι ώστε να υποκλέπτουν η να αλλοιώνουν πληροφορίες στη πηγή τους.

Απειλές κατά του Availability: η απειλή κατά τις διαθεσιμότητας ορίζεται ως η αποτροπή εξουσιοδοτημένης πρόσβασης σε υπηρεσίες, δεδομένα λόγω επίθεσης (DoS) από κακόβουλους χρήστες. Στόχος των επιθέσεων DoS είναι να καταναλώσουν τους πόρους ενός δικτύου η ενός πληροφοριακού συστήματος έτσι ώστε να καταρρεύσει το σύστημα. Αρχικά ο επιτιθέμενος εκμεταλλευόμενος ευπάθειες του συστήματος εγκαθιστά κακόβουλο λογισμικό με αποτέλεσμα να συγκεντρώσει ένα μεγάλο όγκο συστημάτων (Zombies) τα οποία ελέγχει, στο δεύτερο στάδιο ο επιτιθέμενος στέλνει εντολή στα zombies τα οποία ξεκινούν μαζικά να ζητάνε πόρους από το σύστημα – θύμα. Το σύστημα – θύμα προσπαθεί να εξυπηρετήσει όλες τις αιτήσεις πρόσβασης σπαταλώντας όλους τους πόρους του και συνεπώς καταρρέει (Εικόνα 2-5).



Εικόνα 2-5 - DoS Attack (Source: Cisco)

Σύμφωνα με το NIST οι επιθέσεις κατά της διαθεσιμότητας κατηγοριοποιούνται στις εξής κατηγορίες: επιθέσεις κατά των πόρων του δικτύου, επιθέσεις κατά των πόρων (CPU,Memory, Disk) του διακομιστή, επιθέσεις κατά των πόρων ενός λειτουργικού συστήματος και τέλος επιθέσεις που εκμεταλλεύονται σφάλματα του διακομιστή (Bugs, exploits) έτσι ώστε να τον οδηγήσουν στην κατάρρευση η στην παραβίαση.

2.1.4 Ανάλυση Επικινδυνότητας (Risk Analysis)

Σε κάθε πληροφοριακό σύστημα είτε κατά την σχεδίαση του είτε κατά την συντήρηση και ανάπτυξη του θα πρέπει να γίνει μια ανάλυση επικινδυνότητας για να διαπιστωθεί ποιος το απειλεί. Τι είδους συνέπειες έχει μια εισβολή; Τι μηχανισμούς αντιμέτρων μπορούν να μειώσουν η να προλάβουν τους κινδύνους και σε ποια σημεία το σύστημα επιδέχεται αλλαγές έτσι ώστε να γίνει ασφαλέστερο και κατά συνέπεια μια απειλή να έχει τις μικρότερες επιπτώσεις στη λειτουργία του; Αν δοθούν απαντήσεις σε όλες τις παραπάνω ερωτήσεις τότε έχουμε αποκτήσει μια άποψη για την τρέχουσα κατάσταση ασφαλείας του συστήματος και τι επιπτώσεις θα έχουμε σε περίπτωση εισβολής. Επειδή όμως τα συστήματα είναι δυναμικά απαιτείται η συνεχής παρακολούθηση και συντήρηση της

ασφάλειας τους για αυτό το σκοπό υπάρχουν συγκεκριμένες μεθοδολογίες από μεγάλους οργανισμούς πληροφορικής με σκοπό την ανάλυση και την διαχείριση της επικινδυνότητας.

Για την αποτίμηση της ασφάλειας υπάρχουν διάφορες τεχνικές, οι πιο διαδεδομένες είναι η CRAMM [22] και η SBA [23]. Ακολουθώντας μια από τις παραπάνω τεχνικές διαμορφώνουμε την πολιτική και στην συνέχεια το σχέδιο ασφάλειας ανάλογα με τις ανάγκες του οργανισμού για τον οποίο έχει γίνει η μελέτη επικινδυνότητας.

Η ανάλυση κινδύνων βασίζεται στο τύπο ***B > P * L*** όπου B: κόστος, P: πιθανότητα να συμβεί μια απώλεια και L: το κόστος της απώλειας. Αναλυτικότερα η πιθανότητα να συμβεί μια απώλεια είναι συνάρτηση της πιθανότητας μιας απειλής και της ευπάθειας που θα επιτρέψει την πραγματοποίηση της. Αντίστοιχα το κόστος εκτιμάτε με βάση την επίπτωση στα περιουσιακά στοιχεία , αγαθά του οργανισμού. Συνεπώς η επικινδυνότητα είναι συνάρτηση της σοβαρότητας των απειλών, του επιπέδου ευπάθειας και τέλος της αξίας των περιουσιακών στοιχείων του οργανισμού.

Εικόνα 2-6 - Έννοιες Επικινδυνότητας [15]

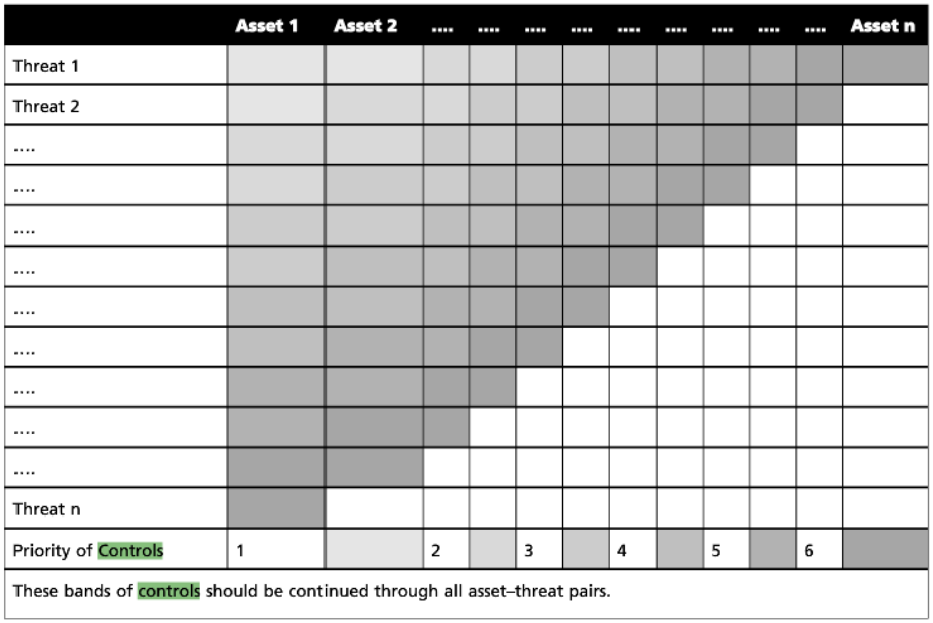
Η τεχνική αυτή αποτιμά την επικινδυνότητα σε κόστος έτσι ώστε να συγκριθεί με το κόστος των αντίμετρων.

Αρχικά γίνεται η καταγραφή των στοιχείων εκείνων που απαιτούν προστασία, για παράδειγμα τα δεδομένα, τα υλικά, τα ενεργά στοιχεία δικτύου, τα λογισμικά, κτλ. Η συλλογή των παραπάνω στοιχείων βασίζεται στα υλικά που απαρτίζουν το σύστημα και στους χρήστες του. Στη συνέχεια γίνεται η αποτίμηση με σκοπό να προσδιοριστεί η σπουδαιότητα των στοιχείων και να ξεχωρίσουμε τις μορφές προστασίας που θα επιβάλουμε ανάλογα της επίπτωση μιας καταστροφής. Πιο συγκεκριμένα εξετάζεται το μέγεθος και το κόστος μια επίπτωσης σε περίπτωση καταστροφής η μη εξουσιοδοτημένης πρόσβασης , η μη διαθεσιμότητας των αγαθών και υπηρεσιών του συστήματος. Μετά το πέρας του πρώτου σταδίου έχουμε μια εκτίμηση της αξίας των αγαθών που απαρτίζουν το πληροφοριακό σύστημα.

Στο επόμενο στάδιο σύμφωνα με τον τύπο υπολογίζεται το επίπεδο των απειλών και το επίπεδο των αδυναμιών έτσι ώστε να υπολογιστεί στην συνέχεια ο βαθμός επικινδυνότητας του συστήματος και να επιλέγουν τα κατάλληλα μέτρα προστασίας των αγαθών. Πιο αναλυτικά προσδιορίζονται οι απειλές για κάθε αγαθό, γίνεται εκτίμηση των απειλών και των αδυναμιών τους, υπολογίζεται ο

βαθμός επικινδυνότητας για κάθε αγαθό-απειλή και τέλος επιβεβαιώνεται ο συνολικός βαθμός επικινδυνότητας του πληροφοριακού συστήματος.

Όπως αναφέραμε στο κεφάλαιο 2.1.3 καταγράφονται οι απειλές (Εικόνα 2-7) ανάλογα με την επίπτωση τους. Έτσι έχουμε απειλές για το λογισμικό το οποίο μπορεί να είναι λειτουργικό σύστημα η λογισμικό εφαρμογών, απειλές του υλικού που μπορεί να είναι διακοπή ρεύματος, απροσεξία ανθρώπων, καταστροφή από φυσικές καταστροφές, κτλ. Τέλος απειλές για τα δεδομένα του οργανισμού που μπορεί να είναι εμπιστευτικά, παραγόμενα από πειράματα και δεδομένα που είναι κρίσιμα για τον οργανισμό όπως είναι μισθοδοσία, κτλ.



Εικόνα 2-7 - Φύλο Αξιολόγησης Απειλών, Ευπαθειών (Source: ΠΕΣ622)

Στη συνέχεια γίνεται μια εκτίμηση της σοβαρότητας των αδυναμιών για κάθε ζεύγος απειλής-αγαθού. Αυτό γίνεται κάνοντας χρήση συγκεκριμένων εργαλείων που παράγουν ερωτηματολόγια για την κάθε αδυναμία και καταλήγουν στη σοβαρότητα και στο μέγεθος της. Τέλος η εκτίμηση αυτή μας παρέχει μια τελική αναφορά ώστε να προχωρήσουμε στην αξιολόγηση της διαδικασίας.

Η μέθοδος στο τελικό της βήμα υπολογίζει τον βαθμό επικινδυνότητας για το σύστημα στο σύνολο του, έχοντας μια αναφορά για κάθε συνδυασμό αγαθού-απειλής και μια εκτίμηση για τις απαιτήσεις ασφαλείας για κάθε αγαθό. Στο επόμενο στάδιο χρησιμοποιώντας τον βαθμό επικινδυνότητας θα γίνει η κατάλληλη επιλογή των αντίμετρων ανάλογα με την σοβαρότητα της κάθε απειλής.

2.1.5 Αντίμετρα Ασφάλειας

Μετά τον υπολογισμό του βαθμού επικινδυνότητας γίνεται η επιλογή των αντιμέτρων σε περίπτωση ενεργοποίησης των απειλών. Τα μέτρα αφορούν τις ενέργειες και τις διαδικασίες που θα εκτελεστούν σε περίπτωση καταστροφής έτσι ώστε να περιορίσουν τις ευπάθειες και τις απειλές του πληροφοριακού συστήματος. Τα αντιμετρά αποτελούνται από 4 κατηγορίες:

- **Πρόληψη** : είναι τα μέτρα που μειώνουν τους κινδύνους από τις απειλές
- **Διασφάλιση**: είναι οι έλεγχοι, οι διαδικασίες και οι τεχνικές που αφορούν την αποτελεσματικότητα των αντιμέτρων.

- **Ανίχνευση** : είναι οι διαδικασίες και οι τεχνικές για την έγκαιρη ανίχνευση των περιστατικών ασφαλείας.
- **Επαναφορά** : είναι οι διαδικασίες που εκτελούνται μετά το πέρας του συμβάντος ασφάλειας έτσι ώστε να επανέλθει το σύστημα σε λειτουργία και να είναι ασφαλές και σταθερό.

Για την επιτυχή εφαρμογή της πολιτικής ασφάλειας καταρτίζεται ειδικό σχέδιο το οποίο περιλαμβάνει όλες τις διαδικασίες που θα πρέπει να εκτελεστούν σε κάθε περίπτωση. Το σχέδιο αναθεωρείται και ενημερώνεται συνεχώς έτσι ώστε να προσαρμόζεται στις αλλαγές και στα πρότυπά που ισχύουν.

Μετά την ολοκλήρωση της πολιτικής ασφάλειας καταρτίζεται το σχέδιο έκτακτης ανάγκης μέσα στο οποίο αναφέρονται όλες οι διαδικασίες ανάκαμψης και αποκατάστασης της λειτουργίας μετά από μια καταστροφή.

Η δημιουργία σχεδίου ασφαλείας είναι μια πολύ δύσκολη και κοστοβόρα διαδικασία με αποτέλεσμα οι διοικήσεις των οργανισμών να μην δίνουν ιδιαίτερη σημασία, όμως εν έτη 2018 είναι ιδιαιτέρα σημαντική η ασφάλεια και επιβάλλεται η δημιουργία σχεδίου για την γρήγορη ανάκαμψη από επιθέσεις κακόβουλων χρηστών και από παραβιάσεις συστημάτων.

2.1.6 Οφέλη Ανάλυσης Επικινδυνότητας και Έλεγχος ασφάλειας

Τα οφέλη της ανάλυσης επικινδυνότητας και της σχεδίασης της πολιτικής ασφάλειας για ένα οργανισμό είναι :

- Έλεγχος και βελτίωση της ασφάλειας του συστήματος. Καταγράφοντας όλες τις απειλές και τις αδυναμίες ανακαλύπτουμε κινδύνους που υπάρχουν και τους εξαλείφουμε. Συνεπώς η συνολική ασφάλεια του συστήματος βελτιώνεται.
- Κατανοούμε καλύτερα το πληροφοριακό μας σύστημα. Λόγω της διερεύνησης των κινδύνων υπάρχει μια αποτύπωση του συστήματος μας με αποτέλεσμα την καλύτερη διαχείριση του.
- Καθορισμός των στόχων που θέλουμε να επιτύχουμε έτσι ώστε να βελτιώσουμε την ασφάλεια του συστήματος .
- Καταγράφοντας τις απειλές κατανοούμε την αναγκαιότητα ενός σχεδίου ασφάλειας για το σύστημα μας. Συνήθως μετά την καταγραφή των αγαθών ανακαλύπτουμε σε ποιες απειλές είμαστε εκτεθειμένοι και παρατηρούμε ότι είναι αναγκαία η ασφάλεια για το πληροφοριακό μας σύστημα.
- Γνωρίζοντας το κόστος μιας παραβίασης είμαστε πλέον σε θέση να δικαιολογήσουμε τις δαπάνες για την ανάπτυξη του σχεδίου ασφάλειας.

Μετά το πέρας της επιβολής ενός σχεδίου ασφάλειας πρέπει να υπάρχουν ελέγχοι που θα εκτελούνται τακτικά έτσι ώστε να ελέγχετε αν η πολιτική ασφάλειας έχει επιβληθεί καθολικά και δεν υπάρχουν εκτεθειμένα συστήματα σε κινδύνους και απειλές λόγω λαθών του προσωπικού η άλλων παραγόντων. Ένα τέτοιο εργαλείο ελέγχει την αρχιτεκτονική του συστήματος μας καθώς και τις αναβαθμίσεις/προσθήκες σε υλικό για να επιβεβαιώσει ότι είναι συμβατά με την πολιτική ασφάλειας και να ανακαλύψει τυχών κενά στο σχέδιο ασφάλειας που πέρασαν απαρατήρητα.

Πιθανοί έλεγχοι είναι οι παρακάτω:

- Έλεγχος του πληροφοριακού συστήματος για ανοιχτές θύρες η συστήματα εκτεθειμένα έξω από τα όρια του εσωτερικού δικτυού.

- Έλεγχος εκδόσεων λογισμικών, λειτουργικών συστημάτων και εγκατάσταση αναβαθμίσεων.
- Έλεγχος αν το εξωτερικό firewall λειτουργεί και αν υπάρχουν πόρτες που δεν ελέγχονται.
- Έλεγχος των εσωτερικών συστημάτων για malwares, ιούς η προγράμματα που δεν επιτρέπονται.
- Έλεγχος των συσκευών που έχουν δικό τους firmware όπως είναι εκτυπωτές, ενεργά στοιχεία δικτύου, fax, κτλ. αν είναι ενημερωμένα με τις τελευταίες εκδόσεις.

Αν οι διαχειριστές των πληροφοριακών συστημάτων εκτελούν τακτικά τους παραπάνω ελέγχους σε συνδυασμό με την ανάλυση επικινδυνότητας είναι σχεδόν σίγουρο ότι η ασφάλεια του οργανισμού θα βελτιωθεί και θα εξαλείφουν απειλές που θα προκαλούσαν την παραβίαση η ακόμα και την κατάρρευση των πληροφοριακών συστημάτων.

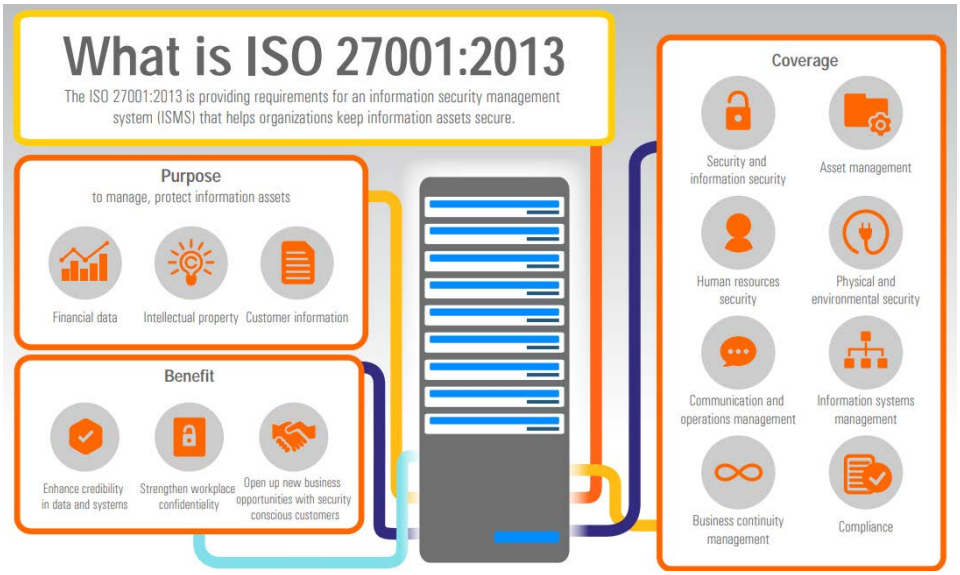
2.2 Πολιτική Ασφάλειας

Όπως έχουμε αναφέρει παραπάνω ένας οργανισμός για να μπορεί να αναπτυχτεί και να πετύχει τους στόχους του επιβάλετε να διασφαλίσει πρώτα την στοιχειώδη ασφάλεια των πληροφοριακών συστημάτων του και σύμφωνα με τον νέο νομό (GDPR [24]) να προστατεύει τα ευαίσθητα δεδομένα που αποθηκεύονται η διαμοιράζονται μέσα σε αυτόν.

Η εφαρμογή ενός σχεδίου ασφάλειας πλέον ακολουθεί διεθνή πρότυπα και τεχνικές μέσω των οποίων πιστοποιείται ένας οργανισμός. Το πρότυπο ISO27001 [25] παρέχει τυποποιημένες διαδικασίες που πρέπει να εκτελέσει μια εταιρία/οργανισμός έτσι ώστε να πιστοποιηθεί κατά ISO27001

2.2.1 Το ISO/IEC 27001

Σε κάθε οργανισμό/εταιρία παράγεται πληροφορία και αποθηκεύεται η επεξεργάζεται ψηφιακά. Σε αυτήν την πληροφορία έχουν πρόσβαση χρήστες από διαφορετικούς υπολογιστές, από διαφορετικές γεωγραφικές περιοχές και με διαφορετικές άδειες προσπέλασης. Σε ένα τέτοιο λειτουργικό περιβάλλον εμφανίζονται κίνδυνοι καταστροφής δεδομένων, μη εξουσιοδοτημένης πρόσβασης, αποκάλυψης σε μη τακτοποιημένα άτομα καθώς και οποιασδήποτε αλλοίωση στα δεδομένα. Για να καταστρωθεί λοιπόν ένα σχέδιο ασφάλειας σύμφωνα με το πρότυπο ISO27001 θα πρέπει να ακολουθηθούν κάποια τυποποιημένα βήματα και διαδικασίες από τον οργανισμό έτσι ώστε να πιστοποιηθεί κατά ISO27001.



Εικόνα 2-8 - ISO27001 ([26])

Έχουμε λοιπόν τις εξής διαδικασίες κατά ISO27001 (Εικόνα 2-8) :

Προσδιορισμός του λειτουργικού περιβάλλοντος:

Καταγραφή της επιχειρηματικής δραστηριότητας του οργανισμού:

Σκοπός της συγκεκριμένης διαδικασίας είναι η καταγραφή του περιβάλλοντος λειτουργίας του οργανισμού. Έτσι καταγράφεται το υπάρχον λειτουργικό περιβάλλον, το αντικείμενο του οργανισμού, οι δραστηριότητες του, η οργανωτική δομή και οι υπηρεσίες που παρέχει. Τέλος καταγράφονται τα μελλοντικά σχέδια του οργανισμού και το επιχειρηματικό πλάνο αν υπάρχει.

Καταγραφή των πληροφοριακών συστημάτων και υποδομών:

Στο συγκεκριμένο στάδιο καταγράφονται τα πληροφοριακά συστήματα τα οποία αποτελούνται από τα κρίσιμα περιουσιακά στοιχεία του οργανισμού διότι σε αυτά γίνεται η συλλογή, αποθήκευση και η επεξεργασία των δεδομένων και αποτελούν σημαντική πληροφορία για την σωστή αξιολόγηση των κινδύνων και γενικά της ασφάλειας του οργανισμού.

Εξέταση των οργανωτικών και τεχνικών διαδικασιών της υπάρχουσας πολιτικής ασφάλειας:

Σε αυτό το στάδιο καταγράφεται η υπάρχουσα πολιτική ασφάλειας και τι δικλείδες ασφάλειας υπάρχουν στην τωρινή κατάσταση του οργανισμού. Αυτό γίνεται για την ευκολότερη καταγραφή των αδυναμιών και πως καλύπτονται από το υπάρχων σύστημα.

Καταγραφή του νομικού πλαισίου που διέπει τον οργανισμό:

Ανάλογα την χώρα που δραστηριοποιείτε ο οργανισμός πρέπει να μελετηθεί η υπάρχουσα νομοθεσία ασφάλειας πληροφοριών που τον διέπει. Η συγκεκριμένη ανάλυση καλύπτει την επιβολή συγκεκριμένων κανόνων από το κράτος οι οποίοι πρέπει να ληφθούν σοβαρά υπόψη κατά την σχεδίαση της πολιτικής ασφαλείας.

Αξιολόγηση, Ανάλυση και Διαχείριση Κινδύνων:

Προσδιορισμός και αξιολόγηση των κινδύνων:

Σε αυτό το στάδιο γίνεται η καταγραφή των κινδύνων ασφάλειας και πως επιδρούν στην καθημερινή λειτουργία του οργανισμού.

Ανάλυση κρισιμότητας των επιπτώσεων στην ασφάλεια πληροφοριών:

Καταγράφονται τα δεδομένα και οι πόροι υπολογιστικής υποδομής του οργανισμού και προσδιορίζεται η κρισιμότητα των πληροφοριών ανάλογα με την περιοχή που ανήκουν. Εκτιμάται η επίδραση της απώλειας της τριάδας CIA και αξιολογείτε η συνολική επίπτωση στον οργανισμό.

Προσδιορισμός Απειλών

Σκοπός της συγκεκριμένης διαδικασίας είναι ο εντοπισμός των κρίσιμων απειλών ασφάλειας του οργανισμού ώστε να γίνουν κατανοητοί οι κίνδυνοι ανά υπηρεσία και δραστηριότητα. Τέλος καταγράφονται όλες οι απειλές και τι κινδύνους επιφέρουν σε περίπτωση ενεργοποίησης τους.

Εντοπισμός αδυναμιών:

Δημιουργείτε λίστα με τις αδυναμίες που θα μπορούσαν να αποτελέσουν αντικείμενο εκμετάλλευσης από τις απειλές. Όπως αναφέραμε στο προηγούμενο κεφάλαιο γίνεται κατηγοριοποίηση των αδυναμιών ανάλογα με το επίπεδο που ανήκουν. ΠΧ ανθρώπινος παράγοντας, προγραμματιστικά λάθη, λάθη στα λειτουργικά συστήματα, κτλ.

Αξιολόγηση πιθανότητας χρήσης των αδυναμιών:

Υπολογισμός της πιθανότητας εκδήλωσης μιας αδυναμίας. Η πιθανότητα χρήσης μιας αδυναμίας μειώνεται με την ύπαρξη δικλείδων ασφαλείας.

Εκτίμηση Κινδύνου:

Για κάθε ζεύγος απειλής-αδυναμίας γίνεται μια εκτίμηση Κινδύνου ανάλογα με την κρισιμότητα των δεδομένων η των πόρων και αποτελεί την απαρχή επιλογής σχεδίου προστασίας έτσι ώστε να επιτευχθεί το επιθυμητό επίπεδο ασφάλειας.

Σχέδιο διαχείρισης Κινδύνου:

Σε αυτό το στάδιο αναπτύσσεται το σχέδιο για την διαχείριση των κινδύνων έτσι ώστε να μειωθούν η να εξαφανιστούν οι κίνδυνοι που έχουν εντοπιστεί. Πιο αναλυτικά προσδιορίζονται όλες οι τεχνικές που σκοπό έχουν την βελτιστοποίηση της ασφάλειας.

Τεκμηρίωση/Παρουσίαση αποτελεσμάτων:

Τέλος τεκμηριώνονται τα αποτελέσματα του ελέγχου και παρουσιάζονται οι προτεινόμενες στρατηγικές του σχεδίου ασφάλειας στην διοίκηση του οργανισμού.

Ανάπτυξη Συστήματος Διαχείρισης Ασφάλειας Πληροφοριών:

Σχεδιασμός συστήματος διαχείρισης:

Το σύστημα διαχείρισης στηρίζεται στους επιχειρηματικούς κινδύνους του οργανισμού και αφορά την υλοποίηση, λειτουργία, συντήρηση και την συνεχή βελτίωση της ασφάλειας του οργανισμού. Αποτελείται από ένα σύνολο οδηγιών, διαδικασιών και απαιτήσεων που πρέπει να τηρούνται έτσι ώστε να κρατηθεί η ασφάλεια στο βέλτιστο επίπεδο. Το κεντρικό μέρος του συστήματος είναι πολιτική ασφάλειας που αποτυπώνει τους μηχανισμούς του οργανισμού έτσι ώστε να μην παραβιάζεται η τριάδα CIA.

Προσδιορισμός εύρους πιστοποίησης:

Καθορισμός του εύρους πιστοποίησης το οποίο προσδιορίζει σε ποια επίπεδα θα εφαρμοστεί η πολιτική ασφάλειας.

Ανάπτυξη σχεδίου διαχείρισης ασφάλειας:

Σε αυτό το στάδιο αναπτύσσεται το σχέδιο ασφάλειας το ποιο περιλαμβάνει, το εύρος πιστοποίησης, το μητρώο αγαθών, την μεθοδολογία αξιολόγησης των κινδύνων, την οργάνωση του σχεδίου και τέλος την τεκμηρίωση σύμφωνα με τα πρότυπα ασφάλειας πληροφοριών.

Προτάσεις υλοποίησης απαιτήσεων σύμφωνα με το ISO27001:

Στο συγκεκριμένο στάδιο γίνονται οι προτάσεις σύμφωνα με τις τεχνικές λεπτομέρειες των δικλίδων ασφαλείας έτσι ώστε η εφαρμογή τους να πληροί τις προϋποθέσεις του ISO27001.

Εκπαίδευση και προετοιμασία της πιστοποίησης:

Σκοπός της συγκεκριμένης φάσης είναι η εκπαίδευση του προσωπικού έτσι ώστε να μπορεί να κατανοήσει και να συμμορφωθεί σύμφωνα με τις ανάγκες ασφάλειας που επιβάλλονται από το πρότυπο.

2.2.2 Απαιτήσεις και κανόνες πολιτικής ασφαλείας

Μετά την εφαρμογή του σχεδίου ασφάλειας καταγράφονται οι απαιτήσεις ασφάλειας έτσι ώστε να αντανακλούν την πολιτική. Συνήθως τα μέτρα ασφάλειας που λαμβάνονται βασίζονται στην ανάλυση επικινδυνότητας και περιλαμβάνουν αναλυτικούς κανόνες για την επίτευξη των στόχων που έχουν τεθεί από το σχέδιο.

Βασικές απαιτήσεις ασφαλείας είναι:

- Καθορισμός των δικαιωμάτων πρόσβασης στα πληροφοριακά συστήματα του οργανισμού.
- Οι μηχανισμοί επιβολής της ασφάλειας δεν πρέπει να μειώνουν την αποτελεσματικότητα και την λειτουργικότητα του συστήματος.
- Η παροχή ευαίσθητων πληροφοριών γίνεται μόνο μετά από απόφαση της διοίκησης.
- Η διαχείριση των δεδομένων και των πληροφοριακών συστημάτων γίνεται μόνο από εξουσιοδοτημένο προσωπικό.
- Σύμφωνα με τον νέο νομό GDPR τα δεδομένα προσωπικού χαρακτήρα θα πρέπει να διαχειρίζονται από το σύστημα με αναλυτική καταγραφή των προσβάσεων από εξουσιοδοτημένο προσωπικό.
- Το σχέδιο ασφάλειας θα πρέπει να εξασφαλίζει την αποδοτική λειτουργία του οργανισμού και συγχρόνως να προάγει την ορθότητα, μυστικότητα, εξουσιοδοτημένη πρόσβαση και διαθεσιμότητα των πληροφοριών.

Η πολιτική ασφάλειας περιλαμβάνει κανόνες οι οποίες καθορίζουν όλες τις αποφάσεις του οργανισμού. Για την υλοποίηση του σχεδίου ασφάλειας απαιτούνται οι παρακάτω κανόνες από το σύστημα και το ανθρώπινο δυναμικό:

- Ταυτοποίηση: κάθε αντικείμενο, χρήστης, πόρος πρέπει να αναγνωρίζεται και να τακτοποιείται.
- Ευθύνη: για κάθε ενέργεια που εκτελείται στο πληροφοριακό σύστημα πρέπει να γίνεται αναγνώριση και καταγραφή του υπεύθυνου.
- Σήμανση βαθμού εμπιστευτικότητας: κάθε αντικείμενο πρέπει να έχει ένα βαθμό εμπιστευτικότητας.
- Ευχρηστία: το σύστημα πρέπει να επιβάλει τους κανόνες ασφάλειας διατηρώντας την ευχρηστία του.
- Αποδοτικότητα: το σύστημα πρέπει να λειτουργεί αποδοτικά, απρόσκοπτα και αξιόπιστα.
- Ασφάλεια: το σύστημα πρέπει να προστατεύει τους πόρους και τα δεδομένα από μη εξουσιοδοτημένη πρόσβαση.
- Ακεραιότητα: τα δεδομένα πρέπει να προστατεύονται από μη εξουσιοδοτημένες μεταβολές.

- Διαθεσιμότητα: το σύστημα πρέπει να εξυπηρετεί απρόσκοπτά τους χρήστες χωρίς καθυστερήσεις στην διαθεσιμότητα των πόρων.
- Πολιτική ασφαλείας: να είναι σαφής οι βασικές αρχές ασφάλειας σε όλο το προσωπικό.
- Επεκτασιμότητα: το σύστημα πρέπει να αναβαθμίζεται εύκολα με άμεση επιβολή των κανόνων ασφάλειας.

2.3 Διαχείριση Κινδύνων-Απειλών

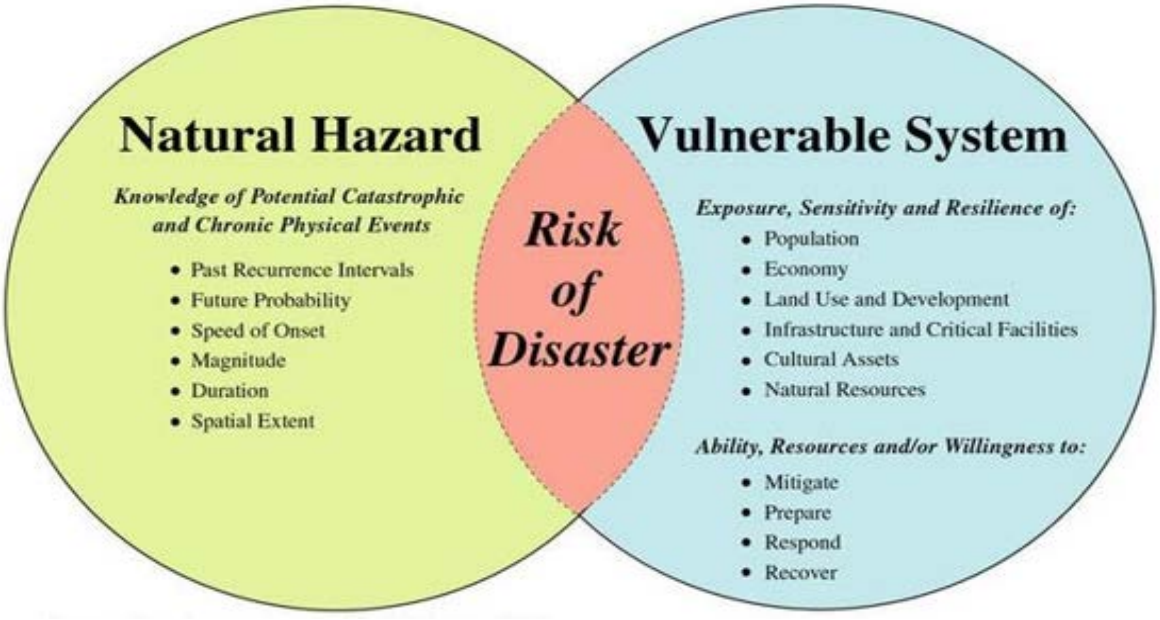
Στο κεφάλαιο αυτό περιγράφονται οι κίνδυνοι που απειλούν ένα πληροφοριακό σύστημα, πως γίνεται ο εντοπισμός τους και πως τους διαχειριζόμαστε. Τέλος περιγράφεται με ποιο τρόπο γίνεται η ανάλυση τους και πως αξιολογούνται έτσι ώστε να γίνει μια εκτίμηση έκθεσης σε αυτούς.

2.3.1 Κατηγορίες κινδύνων

Όπως και στην ζωή μας κίνδυνος είναι ένα απροσδόκητο γεγονός που δεν μπορεί να προβλεφθεί και επιφέρει ζημιές-απώλειες σε εμάς η στο περιβάλλον μας. Ομοίως στα πληροφοριακά συστήματα η έννοια του κινδύνου ορίζεται ως οποιοδήποτε γεγονός μπορεί να προκαλέσει ζημιά η απώλεια στα αγαθά και τους πόρους του οργανισμού. Οι κίνδυνοι των πληροφοριακών συστημάτων δεν μπορούν να προβλεφθούν και μπορεί να προέρχονται από διάφορες πήγες. Το μόνο που μπορεί να καθοριστεί είναι η πιθανότητα εμφάνισης του.

Οι κίνδυνοι διαχωρίζονται σε εσωτερικούς και εξωτερικούς. Εσωτερικοί είναι οι κίνδυνοι που συνδέονται με το εσωτερικό της επιχείρησης (εργαζόμενοι, υλικό, λογισμικό), ενώ εξωτερικοί είναι οτιδήποτε αφορά το εξωτερικό περιβάλλον του οργανισμού (εισβολείς, φυσικές καταστροφές). Κάθε κίνδυνος παράγεται λόγω διαφόρων αιτίων που όταν πυροδοτηθούν προξενούν συνέπειες στη λειτουργικότητα του οργανισμού. Συνεπώς κάθε κίνδυνος αξιολογείται ανάλογα με τις συνέπειες που προκαλεί. Αντίστροφα ένας κίνδυνος μπορεί να προκαλέσει συνέπειες που με την σειρά τους προκαλούν περισσοτέρους κινδύνους.

Για την καλύτερη ανάλυση των κινδύνων επιβάλλεται η κατηγοριοποίηση τους όπως φαίνεται στο παρακάτω σχεδιάγραμμα.

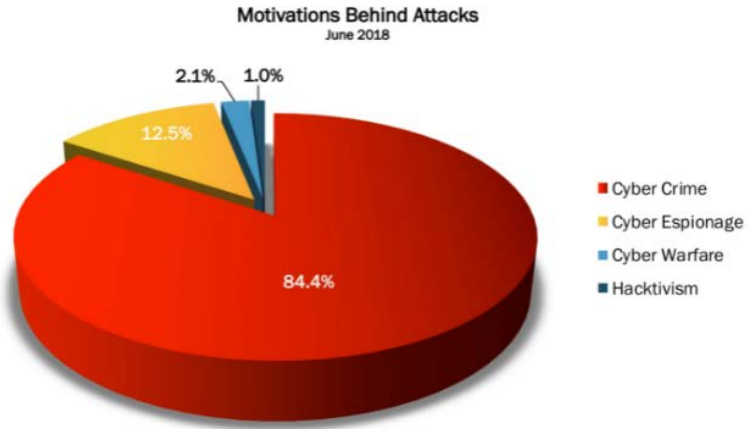


Εικόνα 2-9 – Risk [27]

Παρακάτω περιγράφονται οι ποιο σημαντικές κατηγορίες κινδύνων:

Φυσικές καταστροφές / Ακραία καιρικά φαινόμενα [27] : καταστροφές τέτοιου τύπου μπορεί να είναι οι σεισμοί, οι έντονες βροχοπτώσεις που μπορεί να δημιουργήσουν πλημμύρες, κτλ. οι οποίες μπορούν να πλήξουν όχι μόνο τα πληροφοριακά συστήματα αλλά και τα περιουσιακά στοιχεία του οργανισμού, όπως είναι κτίρια, υπολογιστικά συστήματα, ανθρώπινο δυναμικό, κτλ.

Κυβερνό-επιθέσεις: ένας σοβαρός κίνδυνος που εν έτη 2018 [28] προκαλείται συνεχώς είναι οι επιθέσεις- της τεχνολογίας και μπορούν να προκληθούν είτε από εξωτερικούς παράγοντες ,είτε από εσωτερικούς πχ την εισαγωγή νέου πληροφοριακού συστήματος που έχει ευπάθειες. Αυτού του τύπου κινδύνους καλούμαστε να προβλέψουμε και να μειώσουμε τις απώλειες που επιφέρουν.



Εικόνα 2-10 - Cyber Attacks [28]

Ανθρώπινο δυναμικό: οι κίνδυνοι που αφορούν τους ανθρώπους είναι συνήθως απροσεξίες, πχ διαγραφή δεδομένων από αμέλεια, είτε στοχευμένες επιθέσεις από το ανθρώπινο δυναμικό της εταιρίας πχ υποκλοπή λογισμικού, πατεντών, κτλ.

Κίνδυνοι οργάνωσης η εφαρμογής πολιτικής ασφάλειας: είναι οι κίνδυνοι που αφορούν την οργάνωση της εταιρίας, την ευθύνη για τον σχεδιασμό και την υλοποίηση των έργων καθώς και η λήψη αποφάσεων από την διοίκηση που μπορούν να προκαλέσουν απώλειες στον οργανισμό.

Κίνδυνοι θεσμικού/νομικού πλαισίου: η επιβολή συγκεκριμένων κανόνων έτσι ώστε να εφαρμόζεται καθολικά μια πολιτική ασφάλειας πολλές φορές δεν τηρούνται από το προσωπικό και η διοίκηση δεν μπορεί να πειθαρχήσει σε αυτό, με αποτέλεσμα να δημιουργούνται κίνδυνοι που μπορεί να είναι καταστροφικοί για τον οργανισμό. Πχ εφεδρικά αντίγραφα ασφαλείας, εφεδρική παροχή ηλεκτρισμού, συνεχής έλεγχος για την ενεργοποίηση των αντίμετρων, κτλ.

2.3.2 Διαχείριση/Εντοπισμός των κινδύνων

Στην πράξη η διαχείριση κινδύνων είναι μια επίπονη διαδικασία διότι είναι δύσκολο να μπορέσεις να προβλέψεις τις ευπάθειες και τις απειλές που μπορούν να πυροδοτήσουν ένα κίνδυνο. Συνεπώς όπως αναφέρθηκε στο κεφάλαιο 2.2 ακολουθούνται τυποποιημένα βήματα έτσι ώστε να καταστεί το βέλτιστο δυνατό έργο για την διαδικασία εντοπισμού, ανάλυσης, παρακολούθησης και τέλος αντιμετώπισης των κινδύνων.

Αρχικά αναπτύσσεται ένα σχέδιο διαχείρισης κινδύνων (Εικόνα 2-11), στο οποίο περιγράφονται οι τεχνικές , οι ρολόι και οι αρμοδιότητες της ομάδας διαχείρισης. Στη συνέχεια καταστρώνεται ένας οικονομικός προϋπολογισμός έτσι ώστε να υπάρχει μια εκτίμηση των απωλειών και του κόστους

αντιμετώπισης τους. Τέλος καθορίζονται συχνές συναντήσεις έτσι ώστε να εκπαιδευτεί το προσωπικό και να οριστούν οι τρόποι επικοινωνίας, οι κλίμακες μέτρησης , ο χρόνος που θα διεξάγονται οι έλεγχοι για την εξέλιξη και τον εντοπισμό νέων κινδύνων.

Εικόνα 2-11 - Διαχείριση κινδύνων [20]

Σημαντικό κομμάτι της διαχείρισης είναι η επιλογή του υπεύθυνου διαχειριστή αντιμετώπισης κινδύνων. Το άτομο αυτό αναλαμβάνει την καταγραφή των αρμοδιοτήτων του κάθε εμπλεκόμενου στις δραστηριότητες που αφορούν την αντιμετώπιση των κινδύνων. Αναλαμβάνει να ξεκινήσει διαδικασίες αντιμετώπισης όταν πυροδοτήσει κάποιος κίνδυνος και να καταγράψει τις ζημίες που προκλήθηκαν, στην συνέχεια ενημερώνει την πολιτική ασφάλειας έτσι ώστε να αναθεωρηθεί αν χρειάζεται.

Κατά την διαδικασία του εντοπισμού εκτελούνται οι κατάλληλοι έλεγχοι έτσι ώστε να εντοπιστούν οι κίνδυνοι σε ένα οργανισμό. Αρχικά καταγράφονται πληροφορίες που αφορούν το περιβάλλον και κατηγοριοποιούνται σύμφωνα με το παρακάτω:

- Υπάρχουσες πολιτικές ασφάλειας που εφαρμόζονται ήδη στον οργανισμό.
- Ευαίσθητα προσωπικά δεδομένα και η διαχείριση τους μέσα στον οργανισμό σύμφωνα με τον νέο νόμο προσωπικών δεδομένων GDPR [24].
- Υλικό και λογισμικό που χρησιμοποιούνται από το πληροφοριακό σύστημα.
- Διασυνδέσεις συστημάτων, όπως είναι ροές πληροφοριών, εξωτερικοί συνεργάτες, κτλ.
- Διαχείριση και συντήρηση του συστήματος, στελέχη που απασχολούνται στο IT Department και έχουν καλή γνώση του εξοπλισμού και των δικτύων.
- Και τέλος το επίπεδο προστασίας που πρέπει να επιβληθεί έτσι ώστε να διασφαλιστεί η τριάδα CIA.
- Οι ελέγχοι που εφαρμόζονται σε όλα τα επίπεδα του οργανισμού και είναι οι τεχνικοί, οι λειτουργικοί, οι διοικητικοί και τέλος οι έλεγχοι ασφαλείας.

Οι πληροφορίες συλλέγονται με ειδικά ερωτηματολόγια - συνεντεύξεις στο προσωπικό υποστήριξης και στο προσωπικό της διοίκησης. Στη συνέχεια γίνεται αναθεώρηση των εγγράφων και ανάλυση των μέτρων που έχουν εφαρμοστεί η χρειάζονται αναπροσαρμογή.

2.3.3 Ανάλυση Κινδύνων

Μετά τον εντοπισμό και την καταγραφή των κινδύνων ξεκινά η διαδικασία της ανάλυσης. Τα βήματα επικεντρώνονται στην ποιοτική και στην ποσοτική εκτίμηση των κινδύνων καθώς και των συνεπειών που θα προκαλέσουν σε περίπτωση πυροδότησης.

Η ποιοτική ανάλυση κινδύνων αφορά την πιθανότητα εμφάνισης τους. Αρχικά γίνεται μια πρώτη εκτίμηση και βαθμονομούνται οι κίνδυνοι ανάλογα με τις συνέπειες που θα επιφέρουν, αυτό γίνεται με χρήση βαθμονομημένων κλιμάκων, έτσι χαρακτηρίζονται κίνδυνοι με μεγάλη, μικρή, μεσαία, η ελάχιστη πιθανότητα εμφάνισης (Εικόνα 2-12).

EXAMPLE RISK		Probability				
		Very High	High	Medium	Low	Very Low
Conse-quence	Very High	Very High	Very High	Very High	High	High
	High	Very High	High	High	Medium	Medium
	Medium	High	High	Medium	Medium	Low
	Low	High	Medium	Medium	Low	Very Low
	Very Low	Medium	Low	Low	Very Low	Very Low

Εικόνα 2-12 - Risk analysis table (Source: ICT Institute)

Η διαδικασία εκτίμησης εκτελείται από άτομα που έχουν την κατάλληλη γνώση έτσι ώστε να μπορούν να υπολογίσουν τις πιθανότητες χωρίς περιθώρια λάθους, ώστε να μειωθούν οι επιπτώσεις σε περίπτωση λανθασμένης αξιολόγησης.

Οι κίνδυνοι με υψηλή πιθανότητα εμφάνισης καταγράφονται πρώτοι έτσι ώστε να προχωρήσουν άμεσα στην υλοποίηση αντιμέτρων για την μείωση των συνεπειών. Στη συνέχεια περιγράφονται οι απώλειες σύμφωνα με την τριάδα CIA έτσι ώστε να γίνει αντιστοίχιση μεταξύ κινδύνου και συνέπειας του.

Η ποσοτική αξιολόγηση είναι περισσότερο μαθηματική ανάλυση έτσι ώστε να παρουσιαστεί η πιθανότητα και οι επιπτώσεις με ποιο επιστημονικό τρόπο. Για την διαδικασία υπολογισμού της ποσοτικής εκτίμησης υπάρχουν πολλοί τρόποι που μπορούν να εφαρμοστούν:

Δέντρα γεγονότων: παρουσιάζουν τα αποτελέσματα ενός συμβάντος. Ο στόχος τους είναι να εμφανίσουν την αίτια και τα αποτελέσματα που θα επιφέρει σε περίπτωση πυροδότησης ενός κινδύνου.

Δέντρα σφαλμάτων: είναι μια αναπαράσταση των ανεπιθύμητων γεγονότων – σφαλμάτων που μπορούν να εμφανιστούν σε ένα οργανισμό σε αντιστοίχιση με της αιτίες που θα επιφέρουν.

Προσομοίωση Monte Carlo [29] : είναι μια επαναλαμβανομένη διαδικασία χρησιμοποιώντας στατιστικές μεθόδους και θεωρία τυχαίων αριθμών έτσι ώστε να δημιουργηθεί μια κλίμακα πιθανοτήτων και ο συσχετισμός τους με την λύση των προβλημάτων.

Τεχνική Pert: χρησιμοποιεί θεωρία πιθανοτήτων έτσι ώστε να υπολογίσει την διάρκεια εκτέλεσης ενός έργου ώστε να βρεθεί το κρίσιμο μονοπάτι προς αποφυγή καθυστερήσεων.

Ανάλυση ευαισθησίας είναι μια τεχνική κατά την οποία καθορίζονται ποιοι κίνδυνοι έχουν τις μεγαλύτερες αρνητικές επιπτώσεις στον οργανισμό. Έτσι μπορεί να δοθεί ιδιαίτερη βάση σε κινδύνους που θα προκαλέσουν τις μεγαλύτερες απώλειες.

Τέλος η αναμενομένη τιμή η οποία είναι συνδεδεμένη με την πρόγνωση γεγονότων και πως θα επηρεαστεί ο οργανισμός από αυτά. Υπολογίζει το σύνολο των κινδύνων ανάλογα με την πιθανότητα εμφάνισης τους. Για τον σωστό υπολογισμό θα πρέπει να έχουν καταγραφεί όλοι οι κίνδυνοι του οργανισμού.

Μετά το πέρας της ανάλυσης συντάσσεται η αναφορά αποτελεσμάτων. Η αναφορά περιλαμβάνει όλες τις διαδικασίες που χρησιμοποιήθηκαν για την παραγωγή δεδομένων, τις παρατηρήσεις των ειδικών, τα άτομα που απαρτίζουν το έργο ασφάλειας και την τεκμηρίωση των συμπερασμάτων και των διαδικασιών που χρησιμοποιήθηκαν. Η αναφορά αποτελεί το βασικό σημείο αναφοράς για την όλους τους κινδύνους που απειλούν τον οργανισμό καθώς και τα μέτρα προστασίας που πρέπει να ληφθούν.

2.4 Προστασία και πρόληψη ασφαλείας των πληροφοριακών συστημάτων

Μετά την ανάλυση των κινδύνων, ευπαθειών και γενικά των απειλών ενός πληροφοριακού συστήματος και λαμβάνοντας υπόψη την τελική αναφορά αποτελεσμάτων ξεκινά η διαδικασία δημιουργίας ενός σχεδίου πρόληψης και προστασίας του. Καταρτίζεται ειδικό πλάνο κατά το οποίο αντιμετωπίζονται άμεσα οι κίνδυνοι που επιφέρουν μεγάλες συνέπειες και στην συνέχεια προστατεύονται όλα τα μέρη του συστήματος. Στο κεφάλαιο αυτό περιγράφονται τα κυριότερα αντιμετρά προστασίας και πρόληψης έτσι ώστε να υλοποιηθεί το σχέδιο ασφάλειας.

2.4.1 Μέθοδοι Άμυνας

Για κάθε αδυναμία/ευπάθεια θα πρέπει να υπάρχει ένα αντίμετρο σε περίπτωση πυροδότησης της, όμως όταν ο εισβολέας επιτεθεί είναι μεγάλη η πιθανότητα να υπάρξουν απώλειες, συνεπώς επιβάλλεται να υπάρχει μια διαδικασία άμυνας. Οι βασικές προσεγγίσεις είναι πέντε:

Εμπόδισε την επίθεση: αυτό γίνεται είτε εξαλείφοντας την αδυναμία, είτε προστατεύοντας την. ΠΧ αδυναμία: ανάγκη για ασύρματο δίκτυο, αντίμετρο: εισαγωγή ισχυρού κωδικού.

Δυσκόλεψε την επίθεση: στις οργανωμένες επιθέσεις είναι σχεδόν σίγουρο ότι ο εισβολέας θα προσπαθήσει με διάφορους τρόπους να εισβάλει. ΠΧ brute force attack, δεν πρέπει να επιτρέπετε από του χρήστες να χρησιμοποιούν εύκολα passwords, έτσι ο εισβολέας θα χρειαστεί μέρες αν όχι μήνες για να σπάσει κάποιο συνεπώς θα γίνει αντιληπτός.

Ανακατεύθυνε την επίθεση: οι στόχοι συνήθως είναι συστήματα που είναι δελεαστικά στους εισβολείς, τέτοια συστήματα είναι αυτά που έχουν αδυναμίες εκ φύσεως. Συνεπώς ο διαχειριστής μπορεί να βάλει τέτοια συστήματα-παγίδες (honey spot) και να παρακολουθεί για επιθέσεις. Έτσι ανακαλύπτει τις επιθέσεις πριν προλάβει ο εισβολέας να χτυπήσει κάποιο σημαντικό σύστημα και να υπάρξουν απώλειες.

Ανακάλυψε την επίθεση: πρέπει να υπάρχουν συνεχής αυτοματοποιημένοι έλεγχοι που θα ενημερώνουν τον διαχειριστή για πιθανές επιθέσεις είτε κατά την διάρκεια είτε μετά το πέρας της επίθεσης. Τέτοια συστήματα είναι τα IDS που παρακολουθούν το δίκτυο για συγκεκριμένες ανωμαλίες και τα antivirus που ανακαλύπτουν επιθέσεις από ανθρωπινά λάθη.

Επαναφορά από την επίθεση: όταν ολοκληρωθεί μια επίθεση ο διαχειριστής πρέπει να έχει ένα πλάνο επαναφοράς (recovery plan) κατά το οποίο θα πρέπει να διορθώσει πιθανές απώλειες, να επαναφέρει δεδομένα από τα εφεδρικά αντίγραφα η ακόμα και να αντικαταστεί hardware σε περίπτωση απωλειών από φυσικές καταστροφές.

Σύμφωνα με την Symantec ,τα top 5 μέτρα που πρέπει να ληφθούν έτσι ώστε να αμυνθούμε σε επιθέσεις απεικονίζονται στην (Εικόνα 2-13).

Εικόνα 2-13 - Μέθοδοι άμυνας (Source: Symantec)

2.4.2 Καθορισμός Αντιμέτρων

Μελετώντας την τελική αναφορά ξεκινά η υλοποίηση του σχεδίου ασφάλειας, το οποίο περιλαμβάνει των σχεδιασμό των αντιμέτρων που θα προστεθούν το πληροφοριακό σύστημα από τους κινδύνους που καταγράφηκαν.

Τα αντιμετρά αυτά συνήθως καλύπτουν 5 κατηγορίες:

Οργάνωση και διαχείριση της ασφάλειας του πληροφοριακού συστήματος: το πλάνο της οργάνωσης είναι το σημαντικότερο κομμάτι της ασφάλειας διότι περιλαμβάνει όλες τις διαδικασίες που διέπουν ένα οργανισμό και αφορούν τα πληροφοριακά συστήματα. Αρχικά αποδίδονται ρολόι και αρμοδιότητες στα άτομα που θα συμμετέχουν στην υλοποίηση του πλάνου. Στην συνέχεια συγγράφεται ο κώδικας δεοντολογίας που θα πρέπει να ακολουθεί το προσωπικό, τεκμηριώνονται όλες οι διαδικασίες και οι λειτουργίες του οργανισμού, συγγράφονται εγχειρίδια χρήσης και τέλος ελέγχεται και εποπτεύεται η ασφάλεια του συστήματος. Μετά την τεκμηρίωση εκπαιδεύετε το προσωπικό και ενημερώνετε σχετικά με τους νέους κανόνες ασφάλειας που διέπουν πλέον το σύστημα.

Ασφάλεια συντήρησης και ανάπτυξης του συστήματος: τα μέτρα αυτά αφορούν την ανάπτυξη των εφαρμογών του οργανισμού, την διαχείριση της υποστήριξης του υλικού και τι γίνεται σε περίπτωση ανανέωσης του, την αναβάθμιση του λογισμικού και την συντήρηση του από τους προμηθευτές, κτλ.

Ασφάλεια δεδομένων: τα μέτρα αυτά περιλαμβάνουν μηχανισμούς προστασίας της τριάδας CIA, δηλαδή την εξασφάλιση της ακεραιότητας και της εμπιστευτικότητας των δεδομένων, καθώς και της διαθεσιμότητας τους σε περίπτωση καταστροφής (backup).

Φυσική ασφάλεια: προστασία των κτιριακών υποδομών και του εξοπλισμού πληροφορικής και δικτύων από φυσικές καταστροφές. Τέτοια μέτρα είναι η πυρασφάλεια, η διασπορά του εξοπλισμού σε διαφορετικά κτίρια έτσι ώστε να υπάρχει redundancy, η προστασία τους από πτώση τάσης με χρήση UPS, η προστασία τους από μη εξουσιοδοτημένη φυσική πρόσβαση, κτλ.

Ασφάλεια υπολογιστικής υποδομής: εδώ αναπτύσσονται μηχανισμοί προστασίας της υπολογιστικής υποδομής, όπως διαδικασίες λήψης εφεδρικών αντιγράφων, διαδικασίες ελέγχου για ιούς, malware, bots, rootkits σε διακομιστές αλλά και σε επίπεδο χρηστών, διαδικασίες ελέγχου ισχυρών κωδίκων και διόρθωσης τους, firewalls στα όρια του οργανισμού με τον έξω κόσμο, IDS, διαδικασίες ελέγχου προσπέλασης σε δεδομένα, κρυπτογραφία σε επίπεδο δεδομένων, διαδικασίες καταγραφής συμβάντων η προσπαθειών παραβίασης και τέλος όλα τα μέτρα που αφορούν την ασφάλεια των δικτύων, των διακομιστών, του λογισμικού και των τελικών χρηστών του οργανισμού.

Εικόνα 2-14 - Disaster recovery plan [30]

Τέλος υλοποιείται το σχέδιο έκτακτης ανάγκης (Εικόνα 2-14 - Disaster recovery plan) το οποίο συμπληρώνει το σχέδιο ασφάλειας του οργανισμού. Το σχέδιο έκτακτης ανάγκης περιλαμβάνει την στρατηγική που πρέπει να ακολουθήσει ο οργανισμός σε περίπτωση εισβολής η καταστροφής. Συνήθως μετά από μια καταστροφή τα πληροφοριακά συστήματα μένουν εκτός λειτουργίας για κάποιες ώρες, τότε ενεργοποιείται το σχέδιο έτσι ώστε να επαναφέρει τον οργανισμό σε λειτουργία.

Κεφάλαιο 3
Δοκιμές Διείσδυσης

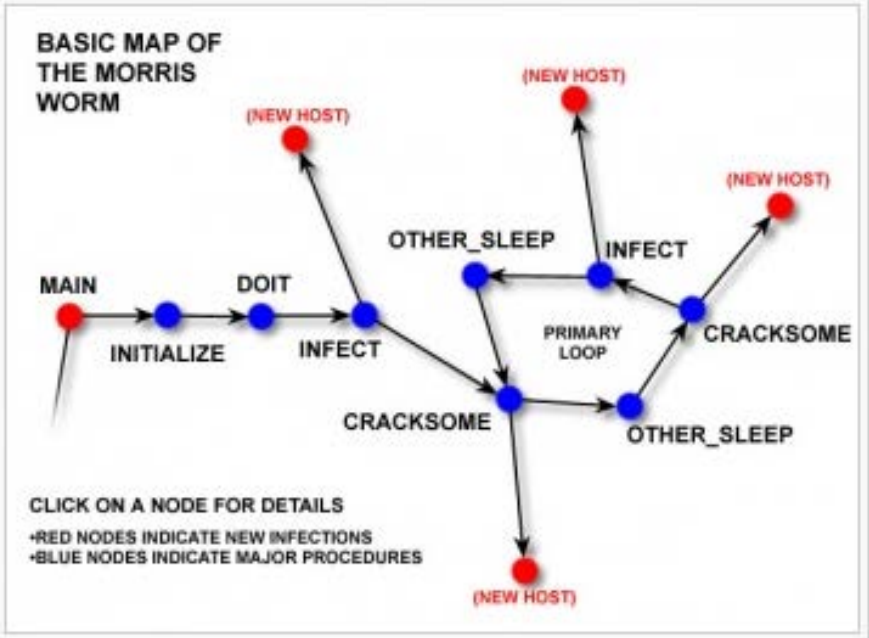
3.1 Εισαγωγή

Στο κεφάλαιο αυτό ορίζεται η έννοια των δοκιμών διείσδυσης, ποιος ο στόχος τους, πως γίνεται η σάρωση για ευπάθειες κάνοντας χρήση των δοκιμών, ποιες μεθοδολογίες χρησιμοποιούν οι εισβολείς για να σαρώσουν τα πληροφοριακά σύστημα και το νομικό πλαίσιο που διέπει τις δοκιμές.

3.1.1 Ορισμός – Ιστορικά στοιχεία

Σύμφωνα με το NIST [20] , δοκιμή διείσδυσης ορίζεται μια δοκιμή ασφάλειας με στόχο να ανιχνευθούν αδυναμίες σε ένα δίκτυο η σε ένα σύνολο πληροφοριακών συστημάτων με κύριο σκοπό την εκμετάλλευση αυτών έτσι ώστε να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε πληροφορίες η σε πληροφοριακά συστήματα.

Πρώτη φορά που χρησιμοποιήθηκε αυτή η τεχνική ήταν το 1988 από το Morris Worm [31] το οποίο σάρωνε το δίκτυο για υπηρεσίες όπως ήταν το sendmail, rsh, finger και σε περίπτωση που ανακάλυπτε την ύπαρξη τους χρησιμοποιούσε γνωστές ευπάθειες για να αντιγράψει των εαυτό του και να προχωρήσει στο επόμενο σύστημα.



Εικόνα 3-1 - Morris Worm [32]

Το σκουλήκι αρχικά ξεκίνησε σαν πείραμα το οποίο ξέφυγε από τον έλεγχο του δημιουργού του με αποτέλεσμα να εισβάλει σε συστήματα που είχαν πρόσβαση στο internet, προκαλώντας ζημίες εκατοντάδων χιλιάδων δολαρίων. Πολλοί διαχειριστές αναγκάστηκαν να κλείσουν τα συστήματα τους προσπαθώντας να αποφύγουν την προσβολή.

Το 1990 αναπτύχθηκε το πρώτο εργαλείο για την ανεύρεση των ευπαθειών με την ονομασία SATAN [33] το οποίο σάρωνε πληροφοριακά συστήματα για να ανιχνεύσει τα τρωτά τους σημεία. Το εργαλείο εκτελούσε συγκεκριμένες δοκιμές αποκαλύπτοντας πιθανές ευπάθειες και αδυναμίες δίνοντας έτσι την δυνατότητα στους διαχειριστές να διορθώσουν τις αδυναμίες των συστημάτων

τους. Το εργαλείο αργότερα μετονομάστηκε σε SAINT και συνδέθηκε με μια βάση δεδομένων με σκοπό να ανανεώνεται κάθε φορά που ανιχνεύεται μια ευπάθεια.

Σήμερα υπάρχει πληθώρα εργαλείων δοκιμών διείσδυσης , κάποια δωρεάν και κάποια εμπορικά τα οποία ανακτούν δεδομένα από βάσεις ευπαθειών έτσι ώστε να είναι ενημερωμένα με τις τελευταίες ευπάθειες και τα τρωτά σημεία των λογισμικών. Τα εργαλεία αυτά θα αναλύσουμε – δοκιμάσουμε στο πρακτικό μέρος της διατριβής.

3.1.2 Στόχος

Η ιστορία έχει δείξει ότι κάθε πληροφοριακό σύστημα μπορεί να παραβιαστεί άσχετα με τις πολιτικές ασφάλειας που του έχουν επιβληθεί. Στόχος λοιπόν του διαχειριστή είναι να ανακαλύψει πρώτος τις ευπάθειες του συστήματος του πριν ανιχνευτούν από τους εισβολείς. Συνεπώς ο χρόνος είναι ιδιαίτερα σημαντικός για την πρόληψη και την διόρθωση των τρωτών σημείων ενός συστήματος. Για να το επιτύχει αυτό πρέπει να βάλει τον εαυτό του στην θέση του εισβολέα και να λειτουργήσει όπως θα λειτουργούσε αυτός. Ο τελικός λοιπόν στόχος των δοκιμών διείσδυσης είναι να αποκαλύψουν στον διαχειριστή τις ευπάθειες των συστημάτων του πριν τις ανακαλύψει ο εισβολέας.

Συνήθως πριν την εκτέλεση των δοκιμών γίνεται η αξιολόγηση των ευπαθειών (Vulnerability Assessment) έτσι ώστε να εξεταστεί συστηματικά ένα σύστημα και να αναλυθούν τα μέτρα ασφάλειας που το διέπουν. Στην συνέχεια θα εκτελεστούν οι δοκιμές μέσω των οποίων θα επικυρωθούν τα τρωτά σημεία και θα εξακριβωθεί η επάρκεια των πολιτικών ασφάλειας. Τα δυο αυτά στάδια ονομάζονται VAPT (Vulnerability Assessment and Penetration testing) και παρουσιάζονται στη παρακάτω εικόνα.

Εικόνα 3-2 – VAPT [34]

Επιγραμματικά οι στόχοι των δοκιμών διείσδυσης είναι:

- Βελτίωση των πολιτικών ασφαλείας και έλεγχος της επάρκειας τους.
- Εντοπισμός των τρωτών σημείων και των ευπαθειών ενός οργανισμού.
- Έλεγχος και επικύρωση της ασφάλειας για την πιστοποίηση ενός οργανισμού.
- Βελτίωση και αναθεώρηση των πληροφοριακών συστημάτων έτσι ώστε να είναι περισσότερο ασφαλή.

Τέλος το αποτέλεσμα των δοκιμών είναι συνήθως μια αναφορά των τρωτών σημείων του οργανισμού και προτάσεις για την εξάλειψης τους.

3.1.3 Πλεονεκτήματα των δοκιμών διείσδυσης

Σύμφωνα με το περιοδικό Quora [35] τα σημαντικότερα πλεονεκτήματα είναι:

Διαχείριση των ευπαθειών και των τρωτών σημείων: μια σωστή δοκιμή διείσδυσης παρέχει στον διαχειριστή μια λεπτομερή αναφορά για τις ευπάθειες των πληροφοριακών συστημάτων σε συνδυασμό με την κρισιμότητα τους έτσι ώστε η διοίκηση να μπορεί να εφαρμόσει την πολιτική ασφαλείας και να διαχειριστεί τα πληροφοριακά συστήματα ποιο αποδοτικά.

Έλεγχος της απόδοσης σε περιπτώσεις επίθεσης: κατά την διάρκεια μιας δοκιμής με επιθετική προσέγγιση ο διαχειριστής αντιλαμβάνεται την ικανότητα του οργανισμού να ανιχνεύσει άλλα και ανταπεξέλθει σε περιπτώσεις κακόβουλης επίθεσης ελαχιστοποιώντας τις ζημιές και ελέγχοντας την ικανότητα των συστημάτων πρόληψης (IDS, Firewall).

Μικρότερος χρόνος μη-διαθεσιμότητας των πληροφοριακών συστημάτων: μετά το τέλος μια επίθεσης-εισβολής χρειάζεται χρόνος για την επαναφορά του οργανισμού σε λειτουργία. Όσο αυξάνεται ο χρόνος τόσο μειώνεται η αξιοπιστία των υπηρεσιών και η παραγωγικότητα του οργανισμού. Με την διεξαγωγή των δοκιμών διείσδυσης ανιχνεύονται οι ευπάθειες και μειώνεται το ρίσκο και ο χρόνος μη διαθεσιμότητας των πληροφοριακών συστημάτων.

Συμμόρφωση με τα διεθνή πρότυπα: ισχυροποιώντας μια πολιτική ασφάλειας μέσω των δοκιμών διείσδυσης συμμορφώνεται ο οργανισμός με τα διεθνή πρότυπα ασφάλειας που έχουν θύσει τα κράτη και οι αρχές ασφάλειας πληροφορικής.

Φήμη του οργανισμού: ένας οργανισμός ανθεκτικός σε επιθέσεις ασφάλειας θα προσελκύσει περισσότερους πελάτες σε σύγκριση με κάποιον άλλον που είναι ευάλωτος σε επίθεσης. Συνεπώς μια καλή φήμη επιφέρει περισσότερα κέρδη και πελάτες.

3.1.4 Περιορισμοί

Μπορεί οι δοκιμές διείσδυσης να επιφέρουν πολλά οφέλη σε ένα οργανισμό άλλα δεν είναι η πανάκεια για όλα τα προβλήματα ασφάλειας του. Οι κύριοι περιορισμοί [36] των δοκιμών είναι:

Χρόνος: κατά τον σχεδιασμό μιας δοκιμής διείσδυσης δεν λαμβάνεται υπόψη ο χρόνος και η συχνότητα των ελέγχων. Έτσι οι διαχειριστές μπορεί να εκτελούν ανά τακτά χρονικά διαστήματα ελέγχους οι οποίοι μπορεί να διαρκούν αρκετές ώρες μέχρι να ολοκληρωθούν, σε αντίθεση με τους εισβολείς που μπορούν να επιτεθούν οποιαδήποτε στιγμή χωρίς περιορισμούς από ωράρια, διαθεσιμότητες και άδειες πρόσβασης.

Πεδίο εφαρμογής: πολλοί διαχειριστές δεν δοκιμάζουν όλα τα πληροφοριακά συστήματα και τα υποδικία διότι μπορεί να έχουν περιορισμένο πεδίο εφαρμογής των ελέγχων λόγο πόρων, αδειών εκτέλεσης ελέγχων, κανονισμών, χρόνου, κτλ.

Περιορισμοί πρόσβασης: ένα οργανισμός συνήθως έχει πολλά παραρτήματα με διαφορετικούς διαχειριστές, με αποτέλεσμα να μην γίνεται ολοκληρωμένος έλεγχος των συστημάτων διότι δεν έχουν πρόσβαση καθολικά.

Περιορισμοί μεθόδων: ένας διαχειριστής χρησιμοποιεί συγκεκριμένες μεθόδους έτσι ώστε να μην βλάψει την λειτουργία των συστημάτων, σε αντίθεση με τον εισβολέα που αδιαφορεί για αυτό, οπότε χρησιμοποιεί επιθετικούς μεθόδους χωρίς περιορισμούς.

Γνώση γνωστών exploits: οι εισβολείς γνωρίζουν καλά όλες τις ευπάθειες των συστημάτων και συνήθως όταν επιλέγουν ένα στόχο γνωρίζουν ακριβώς τα τρωτά σημεία που θα εκμεταλλευτούν, σε αντίθεση με τους διαχειριστές που ενώ έχουν πρόσβαση σε βάσεις δεδομένων, μαθαίνουν τα exploits όταν είναι πλέον γνωστά και έχουν χρησιμοποιηθεί από τους εισβολείς.

Πειραματικοί περιορισμοί: οι διαχειριστές εκτελούν συνήθως τις ίδιες δοκιμές διείσδυσης χωρίς να δοκιμάζουν κάτι καινούργιο, σε αντίθεση με τους εισβολείς που δοκιμάζουν όλους του πιθανούς τρόπους και πειραματίζονται σε πολλά πληροφοριακά συστήματα.

3.1.5 Νομικό Πλαίσιο

Κάθε ενέργεια δοκιμής διείσδυσης και ερευνάς για ευπάθειες σε ένα οργανισμό θα πρέπει να έχει την σύμφωνη γνώμη της διοίκησης. Αυτή είναι και η διαφορά μεταξύ εισβολέα και διαχειριστή, ο εισβολέας δεν υπόκειται σε κάποια νομοθεσία και εκτελεί την διαδικασία παράνομα χωρίς να έχει τα απαραίτητα δικαιώματα, σε αντίθεση με το διαχειριστή που υπόκειται κάτω από κάποια νομοθεσία και πρέπει να τηρεί συγκεκριμένους κανόνες. Συνεπώς κάθε προσπάθεια εύρεσης και χρήσης ευπαθειών πρέπει να έχει την σύμφωνη γνώμη του οργανισμού και να τηρεί τους κανόνες της ελληνικής νομοθεσίας που αφορούν τις αρχές της ιδιωτικότητας στα πληροφοριακά συστήματα και στις επικοινωνίες. Ποιο αναλυτικά θα πρέπει να τηρεί τους κανόνες της ελληνικής νομοθεσίας, της ευρωπαϊκής νομοθεσίας, του νέου νομού GDPR και της αρχής διασφάλισης των επικοινωνιών διαφορετικά ο χρήστης που εκτελεί τις δοκιμές διείσδυσης χωρίς άδεια μπορεί να διωχθεί ποινικά.

3.2 Κατηγοριοποίηση

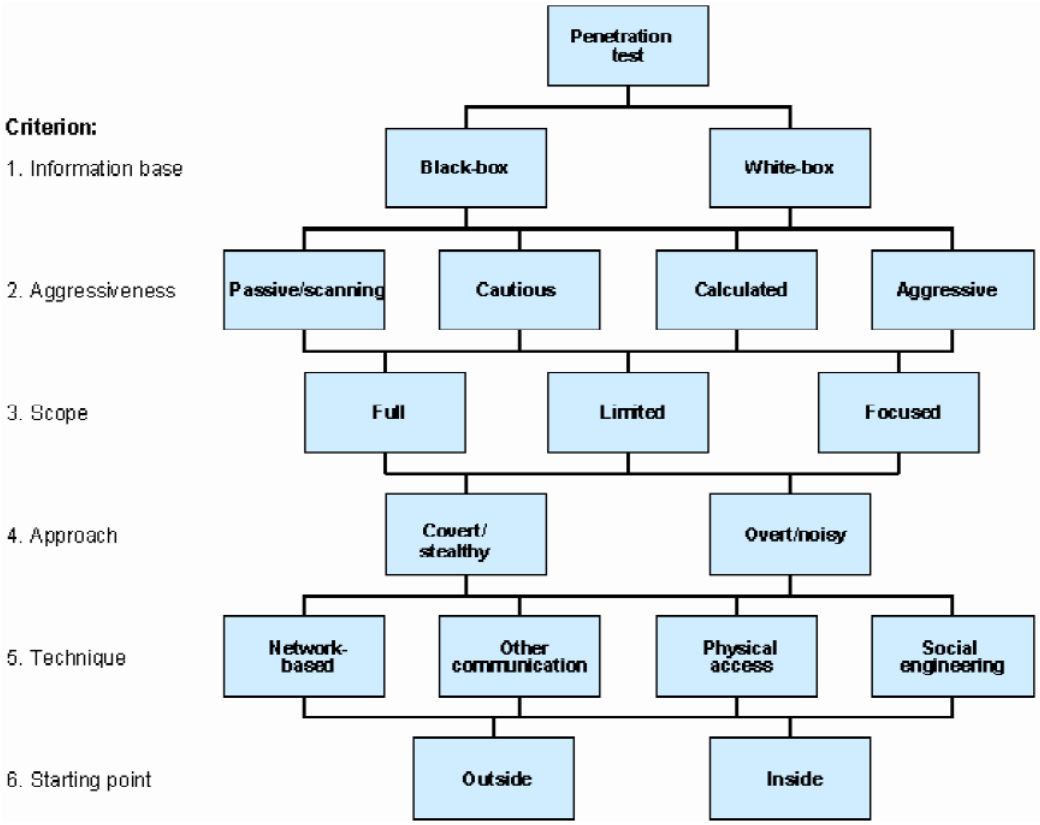
Με την πάροδο των ετών και με την αύξηση των ευπαθειών και των απειλών κατά των πληροφοριακών συστημάτων εμφανίστηκαν αρκετές κατηγορίες δοκιμών διείσδυσης ανάλογα με τον τρόπο, τις πληροφορίες, το πεδίο εφαρμογής , την προσέγγιση και την τεχνική. Στο κεφάλαιο αυτό ταξινομούνται οι δοκιμές διείσδυσης και παρουσιάζονται οι τεχνικές τους.

3.2.1 Ταξινόμηση δοκιμών διείσδυσης

Μια δοκιμή διείσδυσης ανάλογα με τον τρόπο επίθεσης ακολουθεί διαφορετική μεθοδολογία. Έτσι ο εισβολέας προσαρμόζει τους παραμέτρους ώστε να έχει το μέγιστο αποτέλεσμα χωρίς να γίνει εύκολα αντιληπτός. Το γερμανικό γραφείο πληροφοριών κατέγραψε όλα τα κριτήρια και τα χαρακτηριστικά των δοκιμών διείσδυσης σε μια μελέτη [37] προσπαθώντας να της ταξινομήσει και να δώσει στους διαχειριστές συστημάτων μια ολοκληρωμένη περιγραφή όλων τον μεθόδων που ακολουθούν οι εισβολείς.

Η μελέτη ξεχώρισε έξι διαφορετικά κριτήρια, την αφετηρία, την τεχνική, την προσέγγιση, το πεδίο εφαρμογής, την επιθετικότητα και την βάση πληροφοριών. Κάθε κριτήριο έχει διαφορετικά

χαρακτηριστικά και διαφορετικούς παραμέτρους. Ο εισβολέας πριν την εκτέλεση μιας δοκιμής διείσδυσης θα προσπαθήσει να εκμαιεύσει πληροφορίες σχετικά με το πληροφοριακό σύστημα που θα επιτεθεί, στην συνέχεια κάνοντας χρήση αυτών τον δεδομένων θα προχωρήσει στην επιλογή της κατάλληλης δοκιμής έτσι ώστε να αποκτήσει όσο το δυνατόν περισσότερες πληροφορίες.



Εικόνα 3-3 – Κατηγοριοποίηση [37]

Συνεπώς σύμφωνα με το παραπάνω σχήμα έχουμε:

Σημείο εκκίνησης: το σημείο που ο εισβολέας ξεκινά την δοκιμή διείσδυσης. Συνήθως οι επιθέσεις γίνονται από το εξωτερικό δίκτυο η το σημείο σύνδεσης του οργανισμού με το διαδίκτυο, σε αυτή την περίπτωση η επίθεση μπορεί να ανιχνευθεί από το τοίχος προστασίας του οργανισμού, η από το IDS σύστημα που παρακολουθεί την ροή δεδομένων για επιθέσεις. Σε αντίθετη περίπτωση ο εισβολέας έχει ήδη πρόσβαση στο εσωτερικού του οργανισμού και εκτελεί την δοκιμή εκ των έσω, τότε έχει περισσότερες πιθανότητες επιτυχίας να ανιχνεύσει τις ευπάθειες του οργανισμού χωρίς να γίνει αντιληπτός.

Τεχνική: συνήθως μια δοκιμή διείσδυσης εκτελείτε μέσω του διαδικτύου (Network Based). Εκτός από τα κλασικά δίκτυα (Ethernet, tcp/ip) υπάρχουν και επιθέσεις που υλοποιούνται μέσω ασυρμάτων δικτύου, η μέσω γραμμών τηλεφώνου η ακόμα και μέσω Bluetooth στα κινητά τηλεφωνά. Άλλος τρόπος επίθεσης χρησιμοποιεί την κοινωνική μηχανική για να εκμαιεύσει δεδομένα μέσω ανθρωπίνων αδυναμιών. Εκτός από την απομακρυσμένη επίθεση ο εισβολέας μπορεί να εκτελέσει δοκιμές διείσδυσης πηγαίνοντας στο χώρο του οργανισμού (Φυσική πρόσβαση). Τότε η δοκιμή έχει περισσότερες πιθανότητες επιτυχίας διότι εκτελείτε με άμεση πρόσβαση στα πληροφοριακά συστήματα και στα εσωτερικά δίκτυα.

Ορατότητα: κατά την διάρκεια των δοκιμών κύριος σκοπός των εισβολέων είναι να μην γίνουν αντιληπτοί, οπότε πρέπει να επιλέξουν την κατάλληλη προσέγγιση επίθεσης. Σύμφωνα με την αναφορά υπάρχουν δυο προσέγγισης. Η κρυφή κατά την οποία η επίθεση χρησιμοποιεί τεχνικές συγκάλυψης και δεν ανιχνεύεται από τα πληροφοριακά συστήματα και η φανερή κατά την οποία σαρώνονται όλα τα συστήματα και οι υπηρεσίες (port scan) με αποτέλεσμα να γίνει εύκολα αντιληπτή από τους διαχειριστές.

Εφαρμογή: ποια πληροφοριακά συστήματα θα πλήξει η δοκιμή διείσδυσης. Υπάρχουν τρία είδη εφαρμογής, η πλήρης όπου σαρώνονται όλα τα συστήματα και δίκτυα του οργανισμού αποκαλύπτοντας όλες τις ευπάθειες και τα τρωτά σημεία, η περιορισμένη όπου η δοκιμή διείσδυσης σαρώνει ένα δίκτυο του οργανισμού και η επικεντρωμένη κατά την οποία γίνεται σάρωση σε συγκεκριμένα συστήματα η υπό-δίκτυα του οργανισμού. Συνήθως η πρώτη σάρωση πρέπει να έχει πλήρες πεδίο εφαρμογής έτσι ώστε να αποκαλυφθούν όλα τα τρωτά σημεία του οργανισμού.

Επιθετικότητα: το κριτήριο αυτό καθορίζει τον τρόπο που χρησιμοποιεί τα αποτελέσματα της δοκιμής διείσδυσης ο εισβολέας. Έχουμε τέσσερις βαθμούς επιθετικότητας. Η Παθητική, κατά την οποία γίνεται η δοκιμή διείσδυσης, εντοπίζονται οι ευπάθειες αλλά δεν γίνεται καμία προσπάθεια εκμετάλλευσης τους. Προσεκτική, όπου μετά το πέρας της δοκιμής αναλύονται και εκμεταλλεύονται οι ευπάθειες μόνο εάν δεν προκαλούν προβλήματα στο πληροφοριακό σύστημα. Μετρημένη, γίνεται εκμετάλλευση των ευπαθειών που μπορεί να προκαλέσουν κάποια διαταραχή στο σύστημα. Και τέλος η επιθετική όπου ο εισβολέας χρησιμοποιεί τα αποτελέσματα της δοκιμής και θα προκαλέσει προβλήματα στο πληροφοριακό σύστημα (DDos, Memory overflow).

Επίπεδο γνώσης: προσδιορίζει το αρχικό επίπεδο γνώσης που γνωρίζει ο εισβολέας. Σύμφωνα με την μελέτη υπάρχουν δυο τύποι, οι δοκιμές black-box κατά τις οποίες ο εισβολέας δεν έχει καμία εσωτερική πληροφορία για τα συστήματα του οργανισμού, οι δοκιμές white-box όπου ο εισβολέας γνωρίζει εσωτερικές πληροφορίες όπως κωδικούς, συνδεσμολογία δικτύου, κτλ. Μια διαφοροποίηση από την παραπάνω μελέτη αναφέρει και ένα τρίτο επίπεδο γνώσης, το grey-box [38] κατά το οποίο ο επιτιθέμενος έχει μερική γνώση όπως ένα εσωτερικό λογαριασμό με περιορισμένα δικαιώματα η ένα σύστημα που έχει κάποια πρόσβαση εσωτερικά.

Εικόνα 3-4 - Επίπεδο Γνώσης [38]

Προκείμενου να έχουμε το βέλτιστο αποτελέσματα προτείνετε ένας συνδυασμός των παραπάνω μεθόδων, έτσι ώστε να ελεγχτεί η ασφάλεια από διαφορετικές οπτικές γωνίες. Για παράδειγμα αρχικά επιβάλλεται ο πρώτος έλεγχος να γίνεται εξωτερικά από το δίκτυο του οργανισμού και με χαμηλή επιθετικότητα έτσι ώστε να μην υπάρξουν προβλήματα στην λειτουργία του. Στην συνεχεία μπορεί να επιλεγεί μια τεχνική white-box έτσι ώστε να αυξηθεί η αποτελεσματικότητα και να μειωθούν οι ζημιές που μπορεί να προκληθούν από αυτήν.

3.2.2 Κατηγοριοποίηση δοκιμών διείσδυσης ανά περιοχή ελέγχου

Μια δοκιμή διείσδυσης ανιχνεύει τα τρωτά σημεία που έχουν σχέση με την σχεδίαση, τον ανθρώπινο παράγοντα, τα web sites, τα αγαθά και όλες τις υποδομές του οργανισμού. Συνεπώς επιβάλλεται μια ταξινόμηση ανάλογα με την περιοχή ελέγχου. Σύμφωνα με το εγχειρίδιο του NIST [39] οι δοκιμές διείσδυσης κατηγοριοποιούνται ανά περιοχή ελέγχου ως εξής:

Έλεγχος δικτύων: οι δοκιμές ελέγχου δικτύων ανιχνεύουν τα τρωτά σημεία ενός πληροφοριακού συστήματος εξετάζοντας όλα τα μέρη που αποτελείται. Εξετάζουν τα ενεργά μέρη του δικτυού, όπως είναι οι δρομολογητές, τα access points, όλες τις μορφές δικτυού του οργανισμού και τέλος τα πληροφοριακά συστήματα όπως είναι οι διακομιστές, τα workstations, οι συστοιχίες υπολογιστών, κτλ.

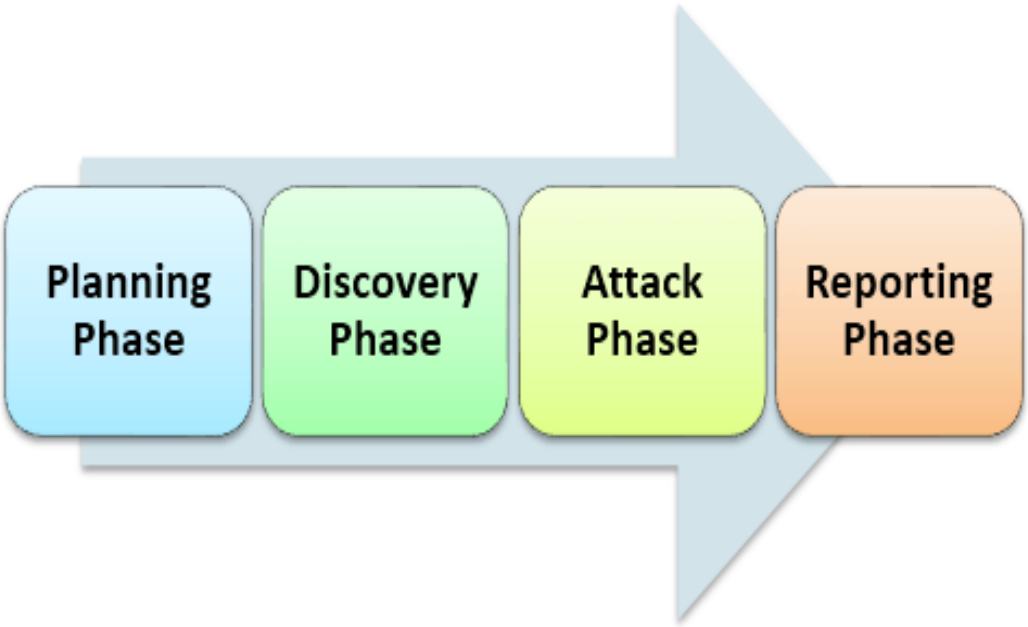
Έλεγχος εφαρμογών: ελέγχονται οι εφαρμογές του οργανισμού που μπορεί να έχουν ευπάθειες οι όποιες θα οδηγήσουν σε απώλεια δεδομένων η σε μη εξουσιοδοτημένη πρόσβαση σε δεδομένα, υπηρεσίες η συστήματα. Ελέγχεται ακόμα ο κώδικας που παράγεται από τους προγραμματιστές για θέματα ασφάλειας που μπορεί να επιφέρουν κατάρρευση της εφαρμογής με απροσδόκητα αποτελέσματα.

Έλεγχος κοινωνικής μηχανικής: γίνεται έλεγχος σε όλες τις διαδικασίες, τεχνικές και εργασίες που εκτελούνται στον οργανισμό με κύριο στόχο τον ανθρώπινο παράγοντα. Συλλέγονται πληροφορίες σχετικά με το ανθρώπινο δυναμικό που εκτελεί καθημερινές διαδικασίες και με ποιο τρόπο μπορούν να αποτελέσουν τρωτό σημείο στην ασφάλεια του οργανισμού.

Έλεγχος εφαρμογών ιστού: κάθε οργανισμός μπορεί να παρέχει υπηρεσίες μέσω εφαρμογών ιστού, αυτού του τύπου οι εφαρμογές πρέπει να προστατευτούν από κακόβουλες επιθέσεις. Ο έλεγχος για τρωτά σημεία γίνεται σε δυο στάδια, στο στάδιο του διακομιστή ο οποίος ελέγχεται σαν πληροφοριακό σύστημα και στο στάδιο της εφαρμογής ιστού όπου πρέπει ελεγχτεί ο κώδικας της εφαρμογής. Συνήθως ο έλεγχος του διακομιστή ακολουθεί την τεχνική του black-box σαρώνοντας από το εξωτερικό δίκτυο του οργανισμού, σε αντίθεση με τον έλεγχο του κώδικα όπου γίνεται εσωτερικά και ακόλουθη προσέγγιση white-box.

3.3 Μεθοδολογία δοκιμών Διείσδυσης

Στο κεφάλαιο αυτό περιγράφετε η μεθοδολογία των δοκιμών, πως σχεδιάζονται, και εκτελούνται έτσι ώστε να είναι πλήρης, επιτυχής και με αξιόπιστα αποτέλεσμα. Για να επιτύχουμε τα παραπάνω πρέπει να ακολουθείτε μια συγκεκριμένη μεθοδολογία, σύμφωνα με άρθρα και οργανισμούς έχουν προταθεί διάφορες προσεγγίσεις ανάλογα με το αντικείμενο και τον σκοπό των δοκιμών διείσδυσης. Αναφορικά, το Institute for security and open Methodologies (ISECOM) προτείνει μια μεθοδολογία ανοιχτού κώδικα OSSTMM όπου διαχωρίζει τους μεθόδους και τα εργαλεία χρησιμοποιώντας λογισμικά ανοιχτού κώδικα. Το NIST αξιολογεί τους μεθόδους και παρέχει στους διαχειριστές μια ολοκληρωμένη λίστα ευπαθειών. Η μεθοδολογία που προτείνει αποτελείται από τον σχεδιασμό, την ανακάλυψη , την επίθεση και τέλος την καταγραφή συμπερασμάτων.



Εικόνα 3-5 - Φάσεις δοκιμών διείσδυσης [40]

Όλες οι μεθοδολογίες που περιγράφονται στο διαδίκτυο περιέχουν τις παραπάνω διαδικασίες. Παρακάτω θα αναλύσουμε τις διαδικασίες ελέγχου σύμφωνα με το άρθρο του PTES Team [40] :

3.3.1 Σχεδιασμός

Η πρώτη φάση για να ξεκινήσει μια δοκιμή διείσδυσης είναι ο σχεδιασμός της στρατηγικής σύμφωνα με τους ελέγχους και τις απαιτήσεις που πρέπει να καλυφτούν. Σε αυτή την φάση λαμβάνονται υπόψη αρχικά όλες οι νομικές διατάξεις μεταξύ ελεγκτή και οργανισμού έτσι ώστε να μην υπάρξουν νομικές συνέπειες. Συνήθως υπογράφεται ένα συμβόλαιο εμπιστευτικότητας έτσι ώστε να προστατευτούν οι πληροφορίες που μπορεί να αποκαλυφθούν κατά την διάρκεια και μετά το τεστ. Ο ελεγκτής είναι υποχρεωμένος να δίνει αναφορά στην διοίκηση του οργανισμού σε όλη την διάρκεια της δοκιμής και να διατηρήσει εμπιστευτικές τις πληροφορίες που θα συλλέξει.

Στην συνεχεία καθορίζονται οι απαιτήσεις του οργανισμού για τα πληροφοριακά συστήματα που πρέπει να ελεγχτούν, αν θα επηρεαστεί η λειτουργία του και ποιες μέθοδοι θα ακολουθηθούν. Ο ελεγκτής ενημερώνεται για τα συστήματα που έχουν μεγίστη προτεραιότητα και αποφασίζει για τους μεθόδους υλοποίησης των δοκιμών. Στην συνεχεία αποφασίζετε ποιες δοκιμές διείσδυσης θα εκτελεστούν και τι πληροφορίες η προσβάσεις θα πρέπει να του δοθούν.

3.3.2 Συλλογή Πληροφοριών

Μετά τον καθορισμό των στόχων ο ελεγκτής θα συλλέξει πληροφορίες για το σύστημα-στόχο. Η συλλογή πληροφοριών (Reconnaissance) είναι ιδιαίτερα σημαντική για την δοκιμή διείσδυσης διότι δίνει την ικανότητα στον επιτιθέμενο να καθορίσει επακριβώς τα τρωτά σημεία του οργανισμού. Σύμφωνα με το άρθρο [40] υπάρχουν δυο ειδών στρατηγικές συλλογής πληροφοριών, η ενεργητική κατά την οποία ο εισβολέας έρχεται σε άμεση επαφή με το σύστημα – στόχο με αποτέλεσμα να γίνεται αντιληπτός και να καταγράφεται στα αρχεία συμβάντων του οργανισμού και την παθητική όπου ο ελεγκτής συλλέγει πληροφορίες από το διαδίκτυο (OSINT) χωρίς να έρχεται σε επαφή με το σύστημα, έτσι δεν γίνεται αντιληπτός από τους διαχειριστές. Αν επιδεχθεί η παθητική προσέγγιση υπάρχουν αρκετά εργαλεία στο διαδίκτυο όπως είναι το google, η βάση δεδομένων whois και προγράμματα όπως είναι το maltego, το dnsmap, κτλ.

Μετά το πέρας της φάσης , ο εισβολέας έχει πλέον πληροφορίες για το δίκτυο , τα συστήματα που το αποτελούν και τις υπηρεσίες που παρέχουν ενώ μπορεί να έχει συλλέξει ευαίσθητες πληροφορίες μέσω κοινωνικής μηχανικής (facebook, phishing) και μπορεί να επιλέξει τις δοκιμές διείσδυσης που θα εκτελέσει.

3.3.3 Σάρωση – Επίθεση

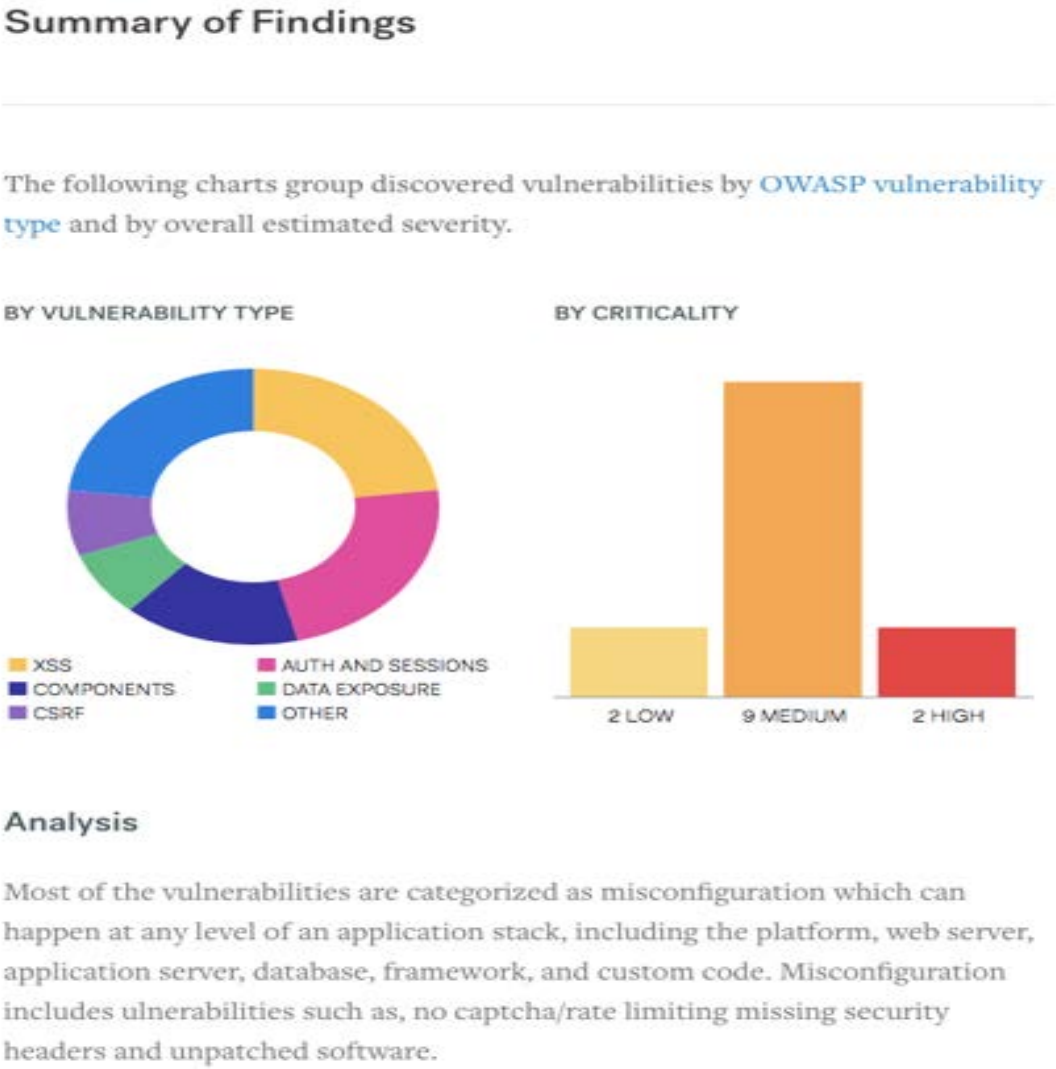
Στη φάση της επίθεσης γίνεται χρήση των πληροφοριών που συλλέχτηκαν στις προηγούμενες φάσεις και χρησιμοποιούνται ως είσοδος στην διαδικασία αυτή. Αρχικά γίνεται μια καταγραφή των συστημάτων που είναι εν ενεργεία κάνοντας χρήση πρωτόκολλων δικτύων όπως το ICMP, στην συνεχεία γίνεται η σάρωση (port scanning) έτσι ώστε να αποκαλυφθούν οι υπηρεσίες που παρέχει ο οργανισμός. Σε αυτή την φάση γίνεται έλεγχος σε όλα τα πρωτόκολλα δικτυού όπως είναι το TCP, το UDP, κτλ. επιλέγοντας ανάλογα με τις απαιτήσεις αν ο έλεγχος θα είναι πλήρης η όχι. Όταν η σάρωση ολοκληρωθεί και ο ελεγκτής έχει μια λίστα με όλες τις πληροφορίες γύρω από την ασφάλεια του οργανισμού προχώρα στην αναζήτηση τρωτών σημείων (enumeration) μέσω των οποίων θα προχωρήσει την επίθεση του. Παραδείγματος χάριν κατά την φάση της σάρωσης ανιχνευτεί ένας ftp server, στην συνεχεία με χρήση κατάλληλων εργαλείων διαπιστώνεται ότι είναι ανοιχτός ο χρήστης anonymous με αποτέλεσμα την παραβίαση του συστήματος με πρόσβαση σε εσωτερικό δίκτυο.

Ο ελεγκτής όταν τελειώσει με την φάση της επίθεσης αναζητά σε βάσεις δεδομένων (securityfocus,NIST) πιθανές αδυναμίες που μπορεί να χρησιμοποιήσει για να παραβιάσει την ασφάλεια του πληροφοριακού συστήματος. Για τον σκοπό αυτό υπάρχουν ειδικά αυτοματοποιημένα εργαλεία όπως είναι το Nessus και το nikto. Στην φάση αυτή γίνεται ουσιαστικά η εκμετάλλευση όλων των πληροφοριών από τις προγενέστερες φάσεις προκείμενου να εκτελέσει επιτυχώς ένα exploit και να παραβιαστεί ένα σύστημα.

3.3.4 Καταγραφή Συμπερασμάτων

Στην τελευταία φάση γίνεται η καταγραφή των συμπερασμάτων από την δοκιμή διείσδυσης. Ο ελεγκτής παρουσιάζει με τεκμηριωμένο τρόπο όλες τις πληροφορίες που συνέλεξε από τις παραπάνω φάσεις, τους τρόπους που παραβίασε τα συστήματα και ποιες ευπάθειες χρησιμοποίησε για να εισχωρήσει.

Η αναφορά είναι πολύ σημαντική καθώς επιτρέπει στον ελεγκτή να παρουσιάσει όλους τους τρόπους εισβολής και να προτείνει αλλαγές και μέτρα ασφαλείας για να επιδιορθώσει τις ευπάθειες που εντοπιστήκαν κατά την φάση της δοκιμής. Αποτελεί ακόμα την εγγύηση για την δουλειά που πραγματοποιήθηκε και χρησιμοποιείται από τον διαχειριστή των συστημάτων ως σημείο αναφοράς για την πολιτική ασφαλείας που παραβιάστηκε έτσι ώστε να πάρει αποφάσεις σχετικά με την αναθεώρησης της. Ένα κομμάτι της αναφοράς παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 3-6 - Αναφορά συμπερασμάτων [41]

3.4 Εργαλεία εκτέλεσης δοκιμών διείσδυσης

Στο κεφάλαιο αυτό παραθέτονται και περιγράφονται εργαλεία δοκιμών διείσδυσης. Ποιο συγκεκριμένα καταγράφονται οι λειτουργιές τους, τα χαρακτηριστικά τους, ο τρόπος που εκτελούν τις δοκιμές και τέλος το περιβάλλον χρήσης που παρέχουν στον τελικό χρήστη.

3.4.1 NMap – ZeNMap [42]

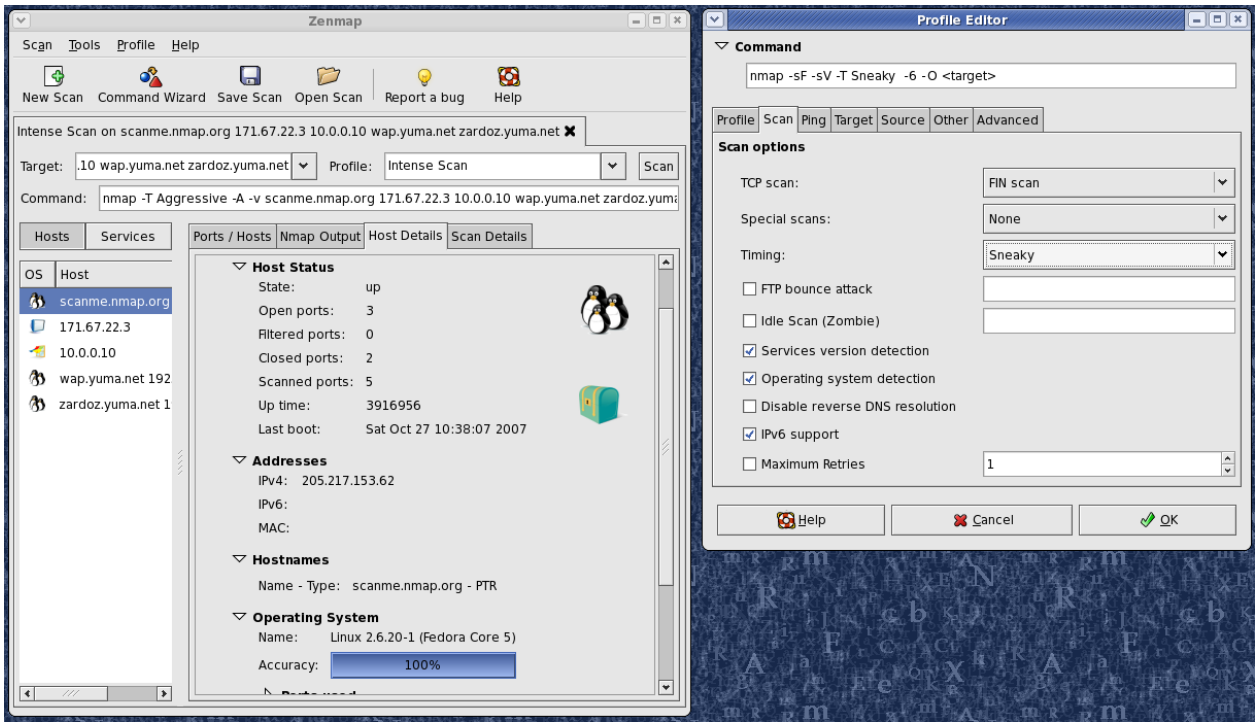
Το εργαλείο δοκιμών διείσδυσης NMap είναι σήμερα το πιο ολοκληρωμένο εργαλείο ανοιχτού κώδικα (GNU) με πληθώρα plugins και αρκετούς παραμέτρους έτσι ώστε να μπορούμε να εκτελέσουμε διαφόρου τύπου δοκιμές. Κυκλοφόρησε το 1997 με κύριο σκοπό την χαρτογράφηση ενός δικτυού και την ανακάλυψη των ενεργών στοιχείων, των υπολογιστών και τις υπηρεσίες που παρέχονται από ένα δίκτυο πληροφοριακών συστημάτων. Σήμερα βρίσκεται στην έκδοση 7.70 και υποστηρίζει τα περισσότερα λειτουργικά συστήματα όπως Linux, Windows, Mac OS, ενώ υπάρχει ο πηγαίος κώδικας του για μεταγλώττιση σε οποιοδήποτε πλατφόρμα απαιτείται.

Με την συνεχή ανάπτυξη του έχει προστεθεί πληθώρα λειτουργιών όπως η καλύτερη ανίχνευση υπηρεσιών, η υποστήριξη όλων των τύπων σάρωσης, η υποστήριξη scripting language και έχτρα εργαλεία όπως nping, ncat, ndiff έτσι ώστε να αναλύει τα αποτελέσματα της δοκιμής, κτλ.

Το NMap από μόνο του δεν ανιχνεύει ευπάθειες σε ένα σύστημα, απλά ιχνηλατεί και σαρώνει συγκεκριμένες πόρτες (port scanning) χρησιμοποιώντας διαφορές τεχνικές όπως FIN, SYN, XMAS, NULL. Η ανίχνευση πραγματοποιείτε στέλνοντας πακέτα σε όλες τις πόρτες ενός πληροφοριακού συστήματος, αν το σύστημα παραλάβει το πακέτο συνεπάγεται ότι παρέχει την υπηρεσία που αντιστοιχεί στην πόρτα αυτή, πχ αν το σύστημα δέχεται συνδέσεις στην πόρτα tcp/23 συνεπάγεται ότι παρέχει την υπηρεσία απομακρυσμένης σύνδεσης telnet.

Με τον παραπάνω τρόπο γίνεται σάρωση σε όλες τις θύρες και σε όλα τα πρωτόκολλα, έτσι ανιχνεύονται όλες οι πιθανές υπηρεσίες που παρέχονται από το σύστημα.

Το NMap είναι ένα εργαλείο γραμμής, δηλαδή εκτελείτε σε terminal εφόσον το λειτουργικό σύστημα είναι Linux η Mac Os η σε cmd αν εκτελείται σε Windows. Δέχεται παραμέτρους μέσω των οποίων καθορίζεται ο τύπος ελέγχου και το δίκτυο στόχο. Μετά το πέρας της εκτέλεσης εμφανίζονται στην οθόνη του χρήστη τα αποτελέσματα. Λόγω της πληθώρας των παραμέτρων δημιουργήθηκε το zeNMap μέσω του οποίου ο χρήστης έχει μια παραθυρική εφαρμογή για την εκτέλεση των δοκιμών επιλέγοντας εύκολα τους παραμέτρους που χρειάζεται. Ένα παράδειγμα εκτέλεσης του zeNMap φαίνεται στην παρακάτω εικόνα:

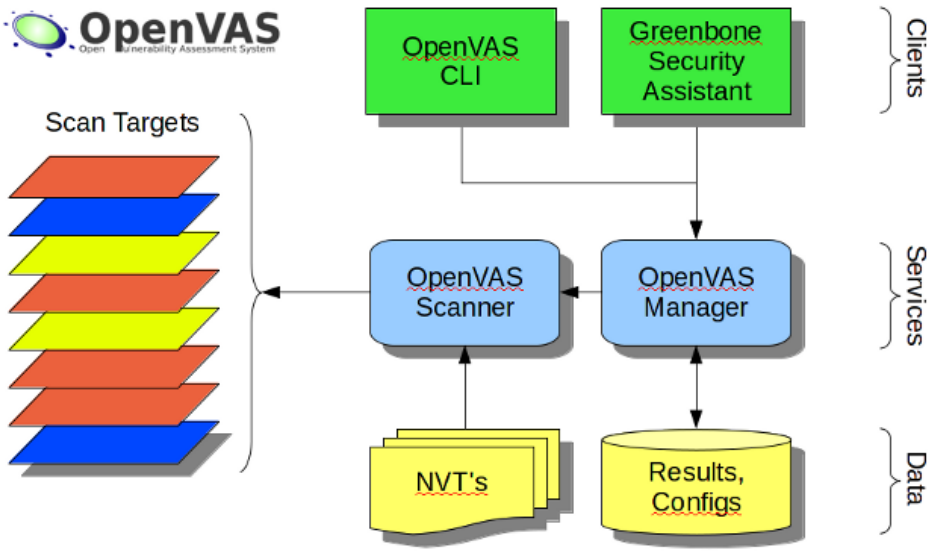


Εικόνα 3-7 – ZenMap [42]

3.4.2 OpenVAS [43]

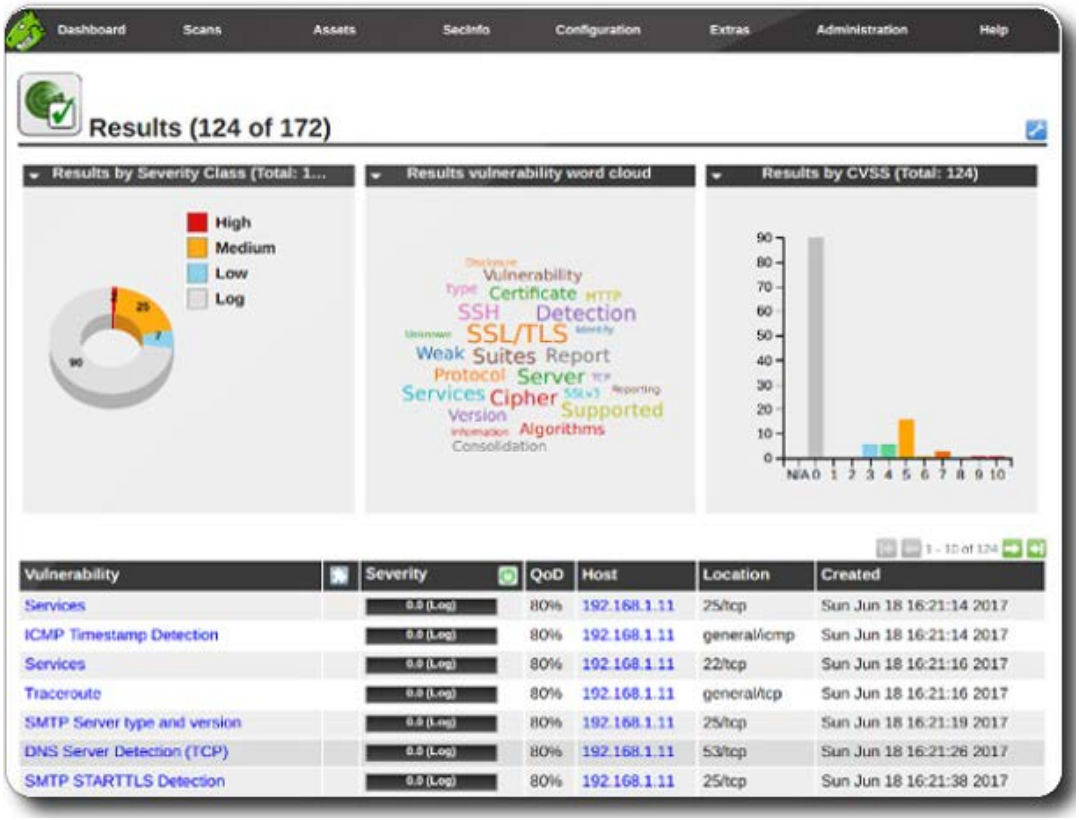
Το OpenVAS είναι ένα εργαλείο ανοιχτού κώδικα που εντοπίζει κενά ασφαλείας σε πληροφοριακά συστήματα. Η διαφοροποίηση του από το NMap είναι ότι μπορεί να ελέγξει για ευπάθειες σε εφαρμογές. Το OpenVAS βρίσκετε στην έκδοση 9 και είναι η συνεχεία του Nessus όταν αυτό έγινε εμπορικό.

Αναπτύσσεται από το 2005 και περιλαμβάνει δυο συστήματα, τον OpenVAS scanner ο οποίος σαρώνει ένα δίκτυο και ανιχνεύει τις ανοιχτές πόρτες και τον OpenVAS manager ο οποίος εκτελεί την αντιστοίχιση σε ευπάθεια χρησιμοποιώντας μεγάλες βάσεις ευπαθειών όπως είναι το CPE και CVE.



Εικόνα 3-8 – OpenVAS [43]

Το OpenVAS έχει αρκετά χαρακτηριστικά με κυριότερο την δυνατότητα αντιστοίχισης σε ευπάθειες. Περιλαμβάνει γραφικό περιβάλλον χρήσης μέσω του οποίου ο χρήστης εκτελεί την σάρωση και αναλύει τα αποτελέσματα. Σήμερα θεωρείται ένα από τα ποιο προηγμένα εργαλεία για δοκιμές διείσδυσης και χρησιμοποιείτε από όλους του IT Managers που θέλουν να ελέγξουν τα δίκτυα τους για ευπάθειες. Ένα στιγμιότυπο οθόνης του OpenVAS φαίνεται στην παρακάτω εικόνα:



Εικόνα 3-9 - OpenVAS environment [43]

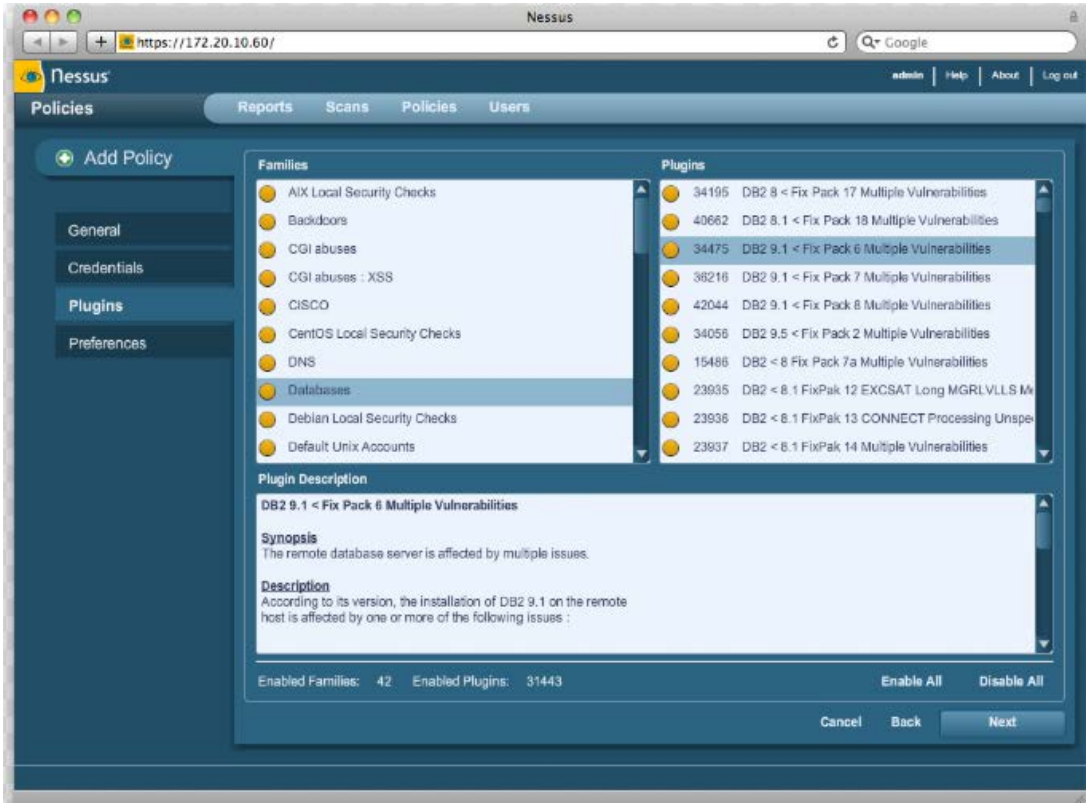
3.4.3 Nessus [44]

Το Nessus είναι ένα εργαλείο με εμπορική άδεια χρήσης το οποίο είναι πολύ δημοφιλές σε μεγάλους οργανισμούς με μεγάλο όγκο πληροφοριακών συστημάτων. Αρχικά δημιουργήθηκε σαν

λογισμικό ανοικτού κώδικα φτάνοντας στην έκδοση 5 το 2008, στην συνεχεία η χρήση του έγινε εμπορική κάτω από την εταιρία tenable με αρχικό κόστος 2500€ ανά έτος. Όπως και τα άλλα εργαλεία δοκιμών διείσδυσης παρέχει πολλές επιλογές ελέγχων ασφάλειας και έχει την ικανότητα ανίχνευσης δικτύου με αντιστοίχιση σε γνώστες ευπάθειες μέσω της βάσης δεδομένων CVEs.

Έχει πολλά plugins μέσω των όποιων μπορεί να οριστεί μια πολιτική ασφαλείας και να ελέγχει για ευπάθειες στους στόχους. Παρέχει γραφικό περιβάλλον καθώς και web interface μέσω του οποίου γίνεται η εκτέλεση των δοκιμών και η εμφάνιση των αναφορών.

Ένα στιγμιότυπο χρήσης παρουσιάζεται στην παρακάτω εικόνα:

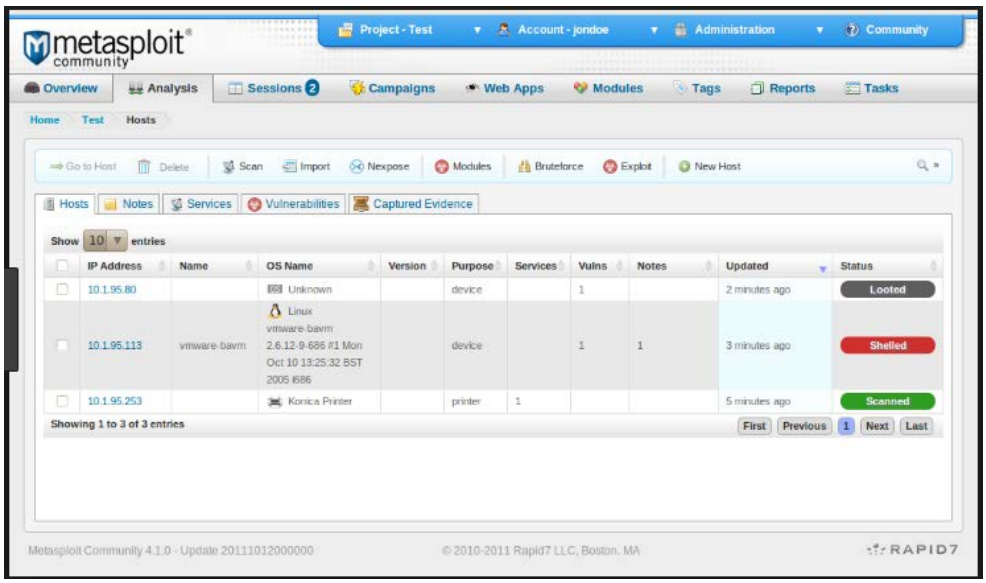


Εικόνα 3-10 – Nessus [44]

3.4.4 MetaSploit [45]

Το metasploit είναι ένα framework πάνω στο οποίο αναπτύσσονται εργαλεία δοκιμών διείσδυσης, χρησιμοποιείτε για τη σκιαγράφηση δικτύων και για ανίχνευση λογισμικών. Είναι ένα εργαλείο ανοικτού λογισμικού μέσω του οποίου γίνεται ανίχνευση ευπαθειών σε ένα πληροφοριακό σύστημα και στην συνεχεία προσπάθεια παραβίασης της. Έχει δυο περιβάλλοντα εργασίας παρέχοντας στους αρχάριους χρήστης ένα web interface ενώ για τους γνώστες παρέχει command line μέσω του οποίου εκτελούνται οι δοκιμές.

Το metasploit παρέχει μια pro έκδοση με ενσωματωμένα modules για την βέλτιστη ανίχνευση των ευπαθειών και community edition για έλεγχο εφαρμογών. Η ροή εργασιών του εργαλείου είναι η ανίχνευση των ευπαθειών, η κατηγοριοποίηση τους και τελικά η εκτέλεση κώδικα για την παραβίαση τους. Η ροή αυτή γίνεται αυτοματοποιημένα και ο τελικός χρήστης εισάγει μόνο δεδομένα στόχους η δεδομένα για μια επιτυχής παραβίαση. Παράδειγμα εκτέλεσης του framework εμφανίζεται παρακάτω:

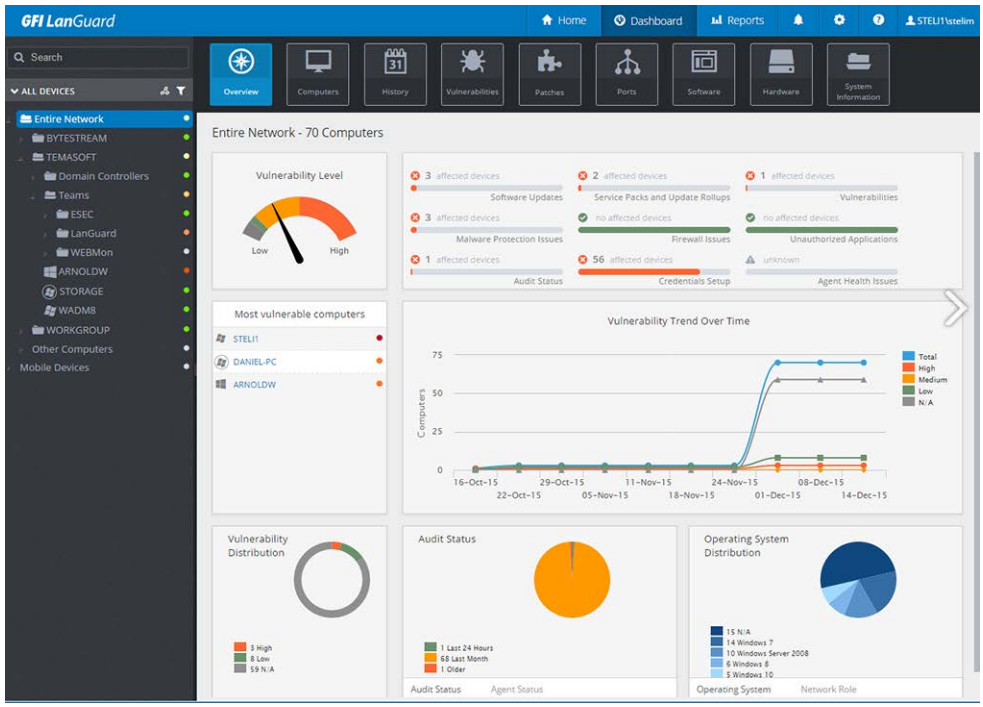


Εικόνα 3-11 – Metasploit [45]

3.4.5 GFI LanGuard [46]

Το GFI LanGuard είναι ένα ολοκληρωμένο εργαλείο που περιλαμβάνει ένα σαρωτή δικτυού, ένα patch management εργαλείο και παρέχει στους διαχειριστές την ικανότητα να ιχνηλατήσουν το δίκτυο τους και να αναλύσουν το ρίσκο ανάλογα με τις ευπάθειες που θα βρεθούν. Το LanGuard παρέχεται μόνο για την πλατφόρμα των Windows και μπορεί να εγκαταστήστε agents σε άλλα λειτουργικά συστήματα για τον καλύτερο έλεγχο.

Το LanGuard βρίσκεται στην έκδοση 12.4 και είναι εμπορικό λογισμικό. Πωλείται με αρχική τιμή 22 € ανά έτος, είναι μια οικονομική λύση για μικρές εταιρίες διότι μέσα σε αυτό το κόστος παρέχεται πρόσβαση σε μια μεγάλη βιβλιοθήκη με ευπάθειες οι όποιες διαχειρίζονται μέσα από ένα πολύ εύκολο γραφικό περιβάλλον. Μετά την εκτέλεση των δοκιμών το πακέτο παράγει εύχρηστες αναφορές και έχει ικανότητα αποθήκευσης και σύγκρισης με μελλοντικές εκτελέσεις.



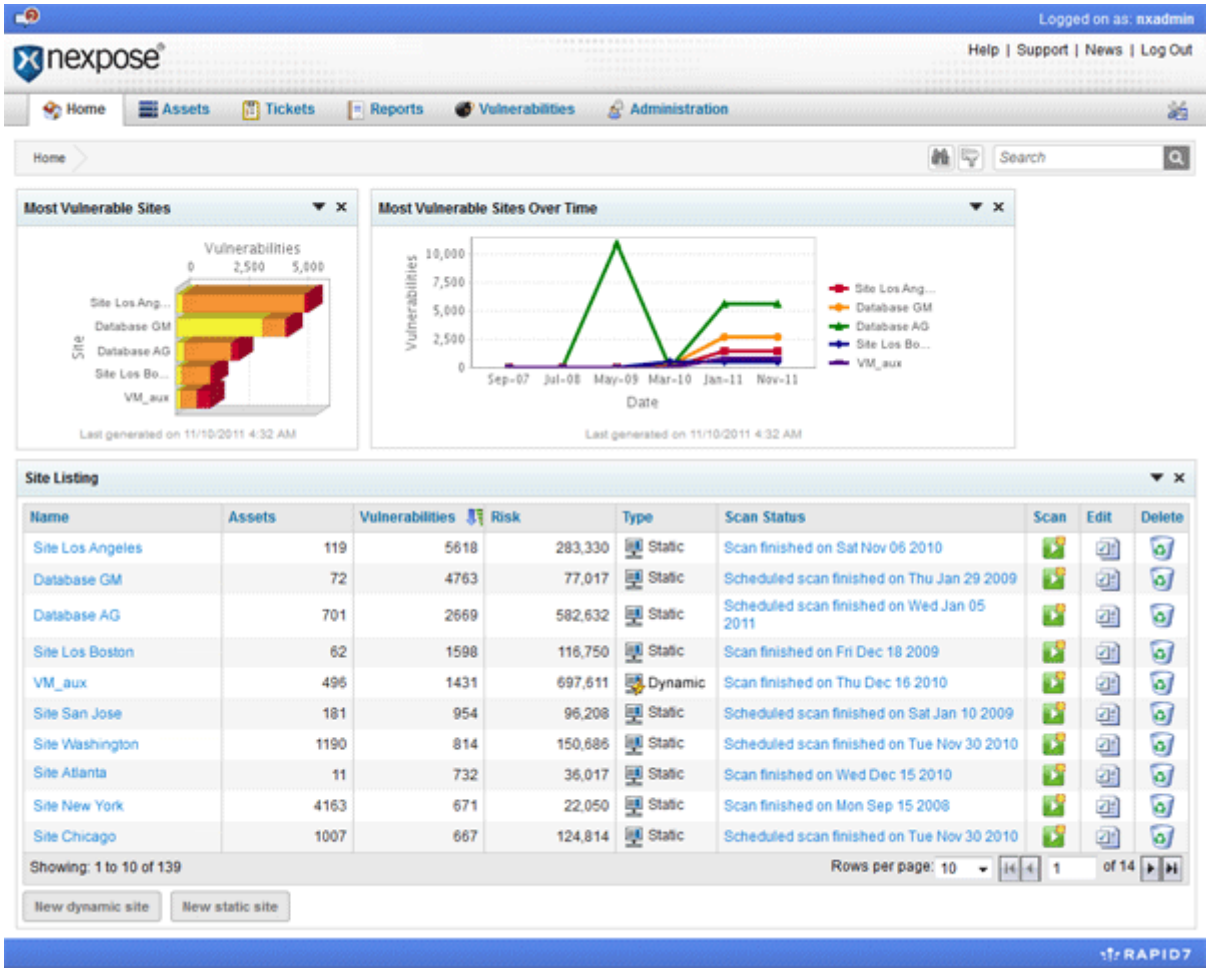
Εικόνα 3-12 - GFI LanGuard [46]

3.4.6 Nexpose [47]

Το Nexpose είναι ένα λογισμικό διαχείρισης ευπαθειών και κινδύνων και παρέχεται δωρεάν από την εταιρία Rapid7. Υποστηρίζει εικονικά περιβάλλοντα και έτσι δεν χρειάζεται εγκατάσταση σε κάποιο λειτουργικό σύστημα.

Το εργαλείο σαρώνει το δίκτυο ανιχνεύοντας ευπάθειες, κακόβουλα λογισμικά όπως ιούς, malwares, κτλ. και παρέχει οδηγίες για την διαχείριση και τον περιορισμό των ευπαθειών. Μετά την σάρωση το πακέτο παρουσιάζει όλα τα πληροφοριακά συστήματα και τις εφαρμογές που παρέχουν ενώ υποστηρίζει σύνδεση με το metasploit framework για την επικύρωση των ευπαθειών.

Σε πολλές αναφορές στο διαδίκτυο το nexpose αναγνώρισε πολύ περισσότερες απειλές και ευπάθειες από άλλα εμπορικά πακέτα και σε συνδυασμό με το metasploit είναι μια πολύ αξιόπιστη λύση για μικρές εταιρίες και οργανισμούς.



Εικόνα 3-13 – Nexpose [47]

3.4.7 Nikto2 [48]

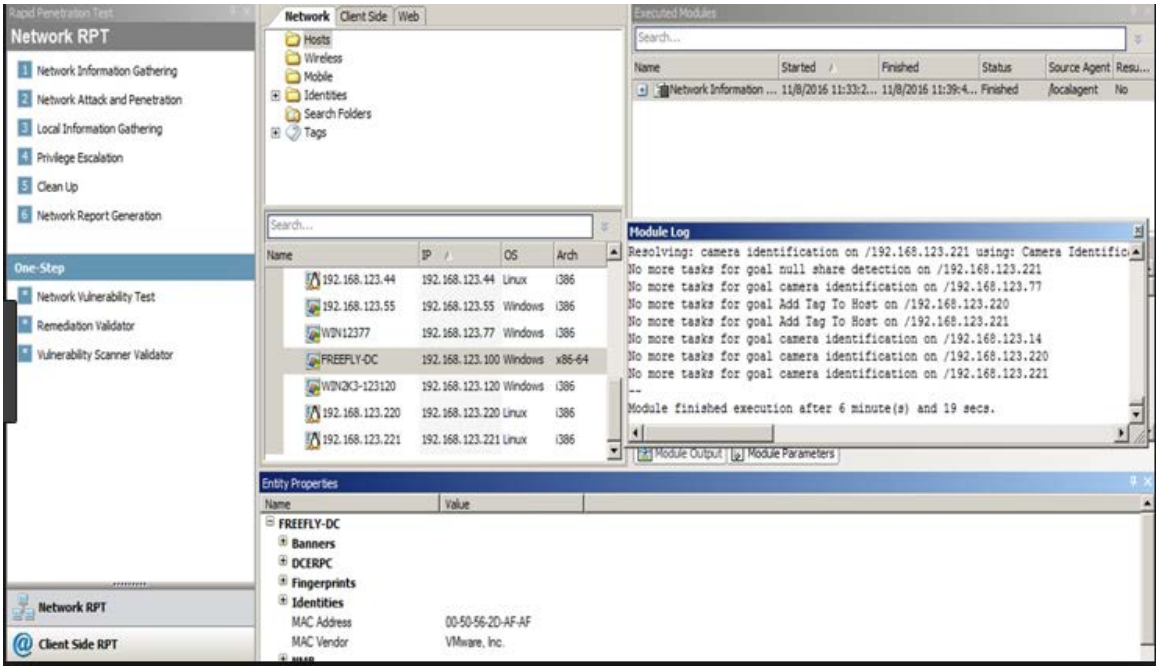
Το Nikto2 είναι ένα εργαλείο δοκιμών διείσδυσης σε web διακομιστές. Ανιχνεύει πάνω από 6000 επικίνδυνα αρχεία η προγράμματα και υποστηρίζει πάνω από 1000 διαφορετικούς web διακομιστές. Το Nikto2 έχει σχεδιαστεί να δουλεύει σαν ένας browser (Stealth mode) έτσι δεν γίνεται αντιληπτό από IDS/IPS η τα firewall του οργανισμού. Υποστηρίζει όλων τον ειδών web pages όπως είναι php,asp,ajax,ssl,html,κτλ και χρησιμοποιείται από τους διαχειριστές για τον έλεγχο των web υπηρεσιών. Παρέχεται μόνο σε πλατφόρμα linux και είναι γραμμένο σε perl, δεν διαθέτει

γραφικό περιβάλλον και εκτελείται από γραμμή εργασιών. Το Nessus το έχει συμπεριλάβει στα εργαλεία που χρησιμοποιεί για τον έλεγχο των web διακομιστών.

3.4.8 Core Impact [49]

Το Core Impact είναι ένα εμπορικό λογισμικό για ελέγχους τρωτότητας και ανίχνευσης ευπαθειών. Έχει πάρα πολλούς αυτοματισμούς και αναλαμβάνει να ανιχνεύσει πιθανά τρωτά σημεία ενός πληροφοριακού συστήματος και στην συνεχεία να εισβάλει σε αυτό εγκαθιστώντας backdoor χωρίς να χρειάζεται καμία επιτήρηση από τους διαχειριστές του.

Το πακέτο κοστίζει 20000€ και είναι το ακριβότερο πακέτο δοκιμών διείσδυσης στο internet. Συγκριτικά με το metasploit το core impact περιέχει 2000 exploits ενώ έχει πρόσβαση στη βάση δεδομένων SCADA που περιλαμβάνει one-day ευπάθειες.



Εικόνα 3-14 - Core Impact

3.5 Συλλογές Εργαλείων (Linux Distributions)

Η δύσκολη εγκατάσταση κάποιων εργαλείων δοκιμών διείσδυσης έφερε την ανάγκη δημιουργίας bootable live CD-ROMs με διάφορες εκδόσεις του Linux το οποίο φορτώνετε σε οποιαδήποτε υπολογιστή και περιέχει όλα τα απαραίτητα εργαλεία για την εκτέλεση των δοκιμών. Στο κεφάλαιο αυτό γίνεται μια αναφορά στα ποιο δημοφιλή distros που χρησιμοποιούνται για δοκιμές διείσδυσης και γενικά για ασφάλεια δικτύων και πληροφοριακών συστημάτων.

Η εκτέλεση των live CD-ROMs έγινε σε περιβάλλον VMware ESXi 6.5 σε εικονική μηχανή με 4GB RAM, 128GB HDD και σύνδεση στο φυσικό δίκτυο του οργανισμού έτσι ώστε να δοκιμαστούν τα εργαλεία δοκιμών διείσδυσης.

3.5.1 Kali Linux ver 2018.4 [50]

Το Kali Linux προήρθε από την διανομή backtrack με κύριο στόχο την εγκατάλειψη του Linux Ubuntu ως βάση και την μεταφορά σε Debian. Υποστηρίζει πάρα πολλές αρχιτεκτονικές όπως είναι x86, x64, arm, Raspberry, κτλ. Η διανομή έχει κύριο σκοπό τις δοκιμές διείσδυσης και περιέχει όλα τα γνωστά εργαλεία που της αφορούν άλλα και προγράμματα εκμετάλλευσης των ευπαθειών στο σύστημα στόχο. Η νέα έκδοση του Kali 2018.4 υποστηρίζει περιβάλλοντα εικονοποίησης με έτοιμα images για

την εγκατάσταση, εκτέλεση σε VMware, hyper-v, kvm. Μια σημαντική αναβάθμιση που περιέχει η διανομή είναι η διορθωμένη έκδοση πυρήνα που αποτρέπει πλέον επιθέσεις cold-boot και AMD secure memory.

Η τελευταία έκδοση περιέχει εργαλεία για hardware, wireless επιθέσεις, όπως το aircrack-ng και το android-sdk για επιθέσεις σε android συστήματα και κινητά. Περιέχει εργαλεία δοκιμών διείσδυσης όπως το NMap, openvas και το Lynis, περιέχει ακόμα εργαλεία για παραβίαση (exploit) των συστημάτων όπως το Armitage, metasploit-framework, maltego, κτλ.

Όπως θα δούμε στην παρακάτω εικόνα η πρώτη οθόνη του Boot cd περιέχει επιλογές για εγκατάσταση της διανομής, η την εκτέλεση της (Live) από το cdrom:

Εικόνα 3-15 -Kali Linux Boot

Μετά από αναμονή μερικών λεπτών το Kali Linux ανάκτησε IP Address από τον DHCP server και εμφάνισε το μενού των εργαλείων χρησιμοποιώντας τον i3 Window manager.

Τα εργαλεία είναι οργανωμένα ανά ομάδα ανάλογα με την δοκιμή που θέλουμε να εκτελέσουμε, όπως φαίνεται στην παρακάτω εικόνα:

Information gathering	Vulnerability Analysis	Web Application Analysis
Database Assessment	Password Attacks	Wireless Attacks
Reserve Engineering	Exploitation Tools	Sniffing, Spoofing
Post Exploitation	Forensics	Social Engineering
System Services	Usual Apps	Activities Overview

Εικόνα 3-16 - Kali Linux Menu

Δεξιά περιέχει τα γνωστότερα εργαλεία όπως το Terminal, Armitage, maltego, κτλ. Συνολικά το Kali Linux είναι μια καλά οργανωμένη διανομή που περιέχει τις τελευταίες εκδόσεις όλων των γνωστών εργαλείων που αφορούν την ασφάλεια, σωστά δομημένα για την εύκολη εύρεση τους από έναν αρχάριο χρήστη.

3.5.2 Knoppix-STD

Η διανομή του Knoppix είναι γνώστη στο κόσμο του Linux σαν η πρώτη live διανομή που επέτρεπε στους χρήστες να δοκιμάσουν το Linux χωρίς να χρειαστεί να διαγράψουν το λειτουργικό τους. Στην συνεχεία προστεθήκαν εργαλεία που αφορούν την ασφάλεια και άλλαξε το όνομα της διανομής σε Knoppix-STD (Security Tools Distribution). Η διανομή περιλαμβάνει εργαλεία με μεγάλη γκάμα χρήσης, όπως είναι εργαλεία για κρυπτογράφηση, για forensics, firewalls, IDS, sniffers και εργαλεία δοκιμών διείσδυσης όπως NMap, hydra, airsnort, κτλ.

Δυστυχώς το knoppix-std παρέχει ένα υποτυπώδη απαρχαιωμένο γραφικό περιβάλλον με τα περισσότερα εργαλεία να εκτελούνταν από γραμμή εντολών. Περιέχει κάποια εργαλεία όπως είναι το Nessus και το metasploit-framework που έχουν παραθυρικό περιβάλλον που όμως είναι παλιές εκδόσεις χωρίς ανανεωμένες βιβλιοθήκες.

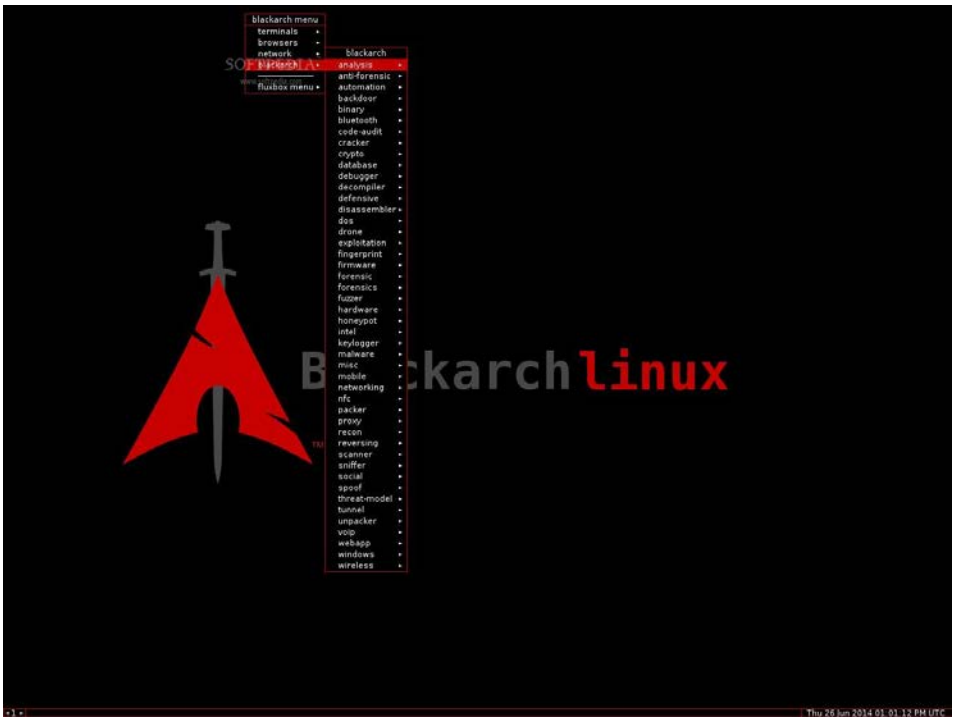
Το 2012 σταμάτησε να αναπτύσσεται το knoppix και μαζί του παρέσυρε και το knoppix-std, έτσι έμεινε στην πρώτη του έκδοση χωρίς να υπάρχουν ανανεώσεις στα εργαλεία που περιέχει.

3.5.3 BlackArch 2018.12.01 [51]

Το Black Arch distro είναι μια διανομή βασισμένη στο Arch Linux. Παρέχει ικανότητα εγκατάστασης άλλα και live λειτουργίας ενώ περιέχει 2100 περίπου εργαλεία για δοκιμές διείσδυσης/exploitation. Θεωρείται μια από της πιο δημοφιλής διανομές για δοκιμές ασφάλειας αν και συγκριτικά με το Kali Linux δεν υποστηρίζει πολλές hardware πλατφόρμες έκτος από τις βασικές x86,x64. Την διανομή μπορούμε να την κατεβάσουμε από την ιστοσελίδα σε ISO μορφή άλλα kai σε OVA image για εγκατάσταση σε περιβάλλοντα εικονοποίησης. Παρέχεται επίσης η ικανότητα να εγκαταστήσουμε τα εργαλεία σε μια ήδη εγκατεστημένη διανομή Arch Linux, κάτι που είναι ιδιαίτερα χρήσιμο αν ο

οργανισμός έχει υπηρεσίες που στηρίζονται στο Arch Linux. Τελευταία έκδοση είναι η 2018.12.01 η οποία περιέχει ανανεωμένες εκδόσεις όλων των εργαλείων.

Στην παρακάτω εικόνα βλέπουμε το κεντρικό μενού του BlackArch Linux:



Εικόνα 3-17 - BlackArch Linux Menu

Παρατηρούμε ότι συγκριτικά με το Kali Linux τα εργαλεία δεν είναι δομημένα σε κατηγορίες άλλα υπάρχουν αλφαβητικά σε ένα μεγάλο μενού με ονομασία BlackArch, κάτι που δεν διευκολύνει το χρήστη στην εκτέλεση συγκεκριμένων δοκιμών διότι πρέπει να ψάχνει συγκεκριμένα ονόματα εργαλείων.

Στην δοκιμή μας το BlackArch εκτελέστηκε σε VMWare περιβάλλον ξεκινώντας κανονικά χωρίς προβλήματα. Ανάκτησε Ip address από τον DHCP server και ένας δειγματοληπτικός έλεγχος σε εργαλεία έδειξε ότι λειτουργούν σωστά.

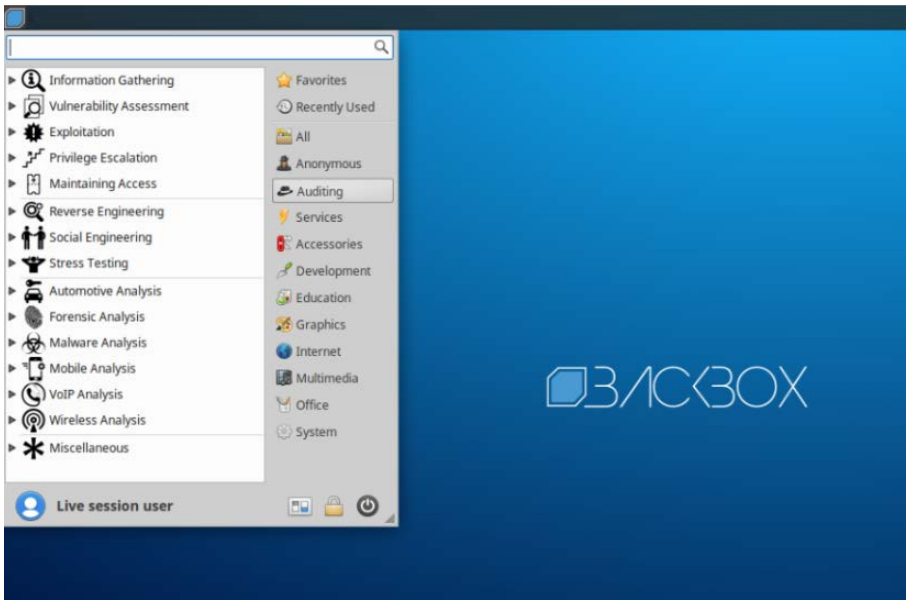
3.5.4 Back Box Linux 5.2 [52]

Η διανομή BackBox είναι βασισμένη στο Ubuntu και παρέχεται για πλατφόρμες x86/x64 και VMware. Περιέχει ένα μεγάλο αριθμό εργαλείων για δοκιμές διείσδυσης και αξιολόγησης ευπαθειών. Το BackBox χρησιμοποιεί τον XFCE window manager που είναι γρήγορος, σταθερός και χαρακτηρίζεται από την ευχρηστία του. Τα μενού του είναι κατηγοριοποιημένα ανάλογα με την χρήση των εργαλείων και περιλαμβάνουν τα παρακάτω:

Information Gathering	Vulnerability Assessment	Exploitation
Privilege Escalation	Maintaining Access	Reverse Engineering
Social Engineering	Stress Testing	Automotive Analysis
Forensics Analysis	Malware Analysis	Mobile Analysis
Wireless Analysis	Miscellaneous	

Έτσι ο χρήστης μπορεί να επιλέξει γρήγορα την δοκιμή που θέλει να εκτελέσει χωρίς να ψάχνει μέσα σε ένα αχανές μενού.

Στη δοκιμή μας το BackBox ήταν αρκετά ποιο γρήγορο από τις άλλες δοκιμές, ειδικά το boot του χρειάστηκε μερικά δευτερόλεπτα, ενώ ο window manager δούλευε απροβλημάτιστα χωρίς καθυστερήσεις. Η πρώτη εικόνα του BackBox Linux 5.2 φαίνεται παρακάτω:



Εικόνα 3-18 - BackBox Linux 5.2

3.5.5 Parrot GNU/Linux 4.4

Το ParrotOS είναι μια ελεύθερη Linux διανομή βασισμένη στο Debian και σχεδιασμένη για διαχειριστές ασφαλείας πληροφοριακών συστημάτων. Περιέχει μερικές εκατοντάδες εργαλεία για δοκιμές διείσδυσης άλλα και εργαλεία ανάπτυξης εφαρμογών για την προστασία της ιδιωτικότητας και την εξάλειψη των αδυναμιών πληροφοριακών συστημάτων. Παρέχεται για πλατφόρμες x86 και x64 και περιλαμβάνει τον MATE desktop manager. Η διαφοροποίηση από τις άλλες διανομές είναι ότι υποστηρίζει το SNAP , ένα universal app store για Linux που παρέχει αρκετά εργαλεία για εγκατάσταση χωρίς να χρειάζεται να γίνουν compile.

Τα μενού του είναι οργανωμένα όπως το Kali Linux παρέχοντας στον χρήστη εύκολη επιλογή των εργαλείων που επιθυμεί να εκτελέσει.

Εικόνα 3-19 - Parrot OS

3.5.6 Bugtraq II BlackWidow

Η διάσημη ομάδα BugTraq η οποία ασχολείται από το 2011 με την ανακάλυψη ευπαθειών και χρήση αυτών για παραβιάσεις γνωστών συστημάτων, δημιούργησε την διανομή BlackWidow. Η διανομή περιέχει εργαλεία για δικανική Η/Υ, δοκιμές διείσδυσης, εργαλεία για network mapping, exploitation tools, ανάλυση ευπαθειών και τέλος προσπαθεί να είναι φιλική και εύχρηστη για τον τελικό χρήστη που θα την χρησιμοποιήσει. Υποστηρίζει όλες τις γνώστες πλατφόρμες υλικού και μπορούμε να κατεβάσουμε έκδοση ανάλογα τον window manager που επιθυμούμε. Επίσης υποστηρίζει όλες τις γνώστες γλώσσες και είναι μεταφρασμένο σε αυτές. Μια διαφοροποίηση από τις άλλες διανομές είναι η υποστήριξη του Android με εργαλεία για τον έλεγχο της ασφάλειας και την ανίχνευση ευπαθειών του.

Υποστηρίζει ακόμα εργαλεία για πλατφόρμες windows, GSM, Bluetooth και RFID. Έκτος από την επιλογή του window manager μπορούμε να επιλέξουμε την διανομή που είναι βασισμένο όπως Ubuntu, Debian η Opensuse.

Το BlackWidow όπως βλέπουμε παρακάτω έχει κατηγοριοποιήσει τα εργαλεία έτσι ώστε να είναι εύκολα στην εύρεση. Παρέχει ακόμα terminal, browser και εργαλεία συστήματος.

Εικόνα 3-20 - BlackWidow Linux 2

3.6 Αξιολόγηση των εργαλείων Διείσδυσης

Στο κεφάλαιο αυτό γίνεται η αξιολόγηση των εργαλείων διείσδυσης. Για την σωστή κατάταξη τους πρέπει να τεθούν αρχικά τα κριτήρια αξιολόγησης τους, στην συνεχεία να δοκιμαστούν τα εργαλεία και να βαθμονομηθούν σύμφωνα με την ταξινόμηση που θα θέσουμε.

Καίριο σημείο για ένα εργαλείο είναι η απόδοση του, κατά πόσο δηλαδή ανιχνεύει όλες τις ευπάθειες στο σύστημα που σαρώνει. Δεύτερο σημαντικό κριτήριο είναι το κόστος του, τρίτο κριτήριο η ευκολία χρήσης του και τέλος η πληρότητα του, αν παρέχει δηλαδή υποστήριξη, αν υπάρχει διαθέσιμη βοήθεια στο internet, αν υποστηρίζει πολλές πλατφόρμες, γλώσσες, κλπ.

Σύμφωνα λοιπόν με τα παραπάνω κριτήρια θα αξιολογηθούν τα εργαλεία που αναφέρθηκαν στο προηγούμενο κεφάλαιο αφού πρώτα εγκατασταθούν και εκτελεστούν σε ένα εικονικό περιβάλλον.

Στην συνεχεία θα βαθμολογηθούν με βαθμούς από το 1-5, όπου 1 ο χαμηλότερος βαθμός και 5 ο υψηλότερος. Τέλος θα υπολογιστεί η συνολική βαθμολογία και θα καταγράφουν οι ελλείψεις του κάθε εργαλείου.

3.6.1 Αξιολόγηση μεμονωμένων εργαλείων

Μετά την δοκιμή των παραπάνω εργαλείων, το NMap-zeNMap με τους καταλλήλους παραμέτρους ανίχνευσε όλα τα hosts και τις υπηρεσίες που παρείχαν ενώ είναι αρκετά δύσκολο στην παραμετροποίηση σε έναν αρχάριο χρήστη. Το κενό στην ευχρηστία έρχεται να καλύψει το zeNMap παρέχοντας ένα γραφικό περιβάλλον (GUI) διευκολύνοντας τον διαχειριστή στην παρουσίαση των αποτελεσμάτων και στην επιλογή των καταλλήλων παραμέτρων. Σε θέματα πληρότητας το NMap έχει μια τεράστια βάση πληροφοριών γύρω από τους τρόπους και την εκτέλεση του, ενώ η κοινότητα δημιουργεί plugins έτσι ώστε να αυξήσουν την ευελιξία του. Παρέχεται ακόμα σε όλες τις γνώστες πλατφόρμες υλικού η σε περίπτωση μη υποστήριξης δίνεται ο πηγαίος κώδικας του για μεταγλώττιση σε οποιαδήποτε υλικό.

Το OpenVas εγκαταστάθηκε πολύ εύκολα κάνοντας deploy το image του στο VMWare server, είχε πολύ φιλικό περιβάλλον σε σύγκριση με το NMap ενώ άργησε πολύ να εκτελέσει τον έλεγχο στο ίδιο δίκτυο που εκτελέστηκε το NMap. Είναι και αυτό ένα δωρεάν εργαλείο όπως το NMap με αρκετή τεκμηρίωση και συνδεσιμότητα με άλλα εργαλεία δοκιμών διείσδυσης.

Το Nessus ξεκινά με αρχικό κόστος τα 2500€ και συγκριτικά με άλλα εμπορικά εργαλεία είναι οικονομικότερο. Για την δοκιμή μας χρησιμοποιήσαμε ένα trial για 30 ήμερες και εκτελέσαμε την σάρωση στον ίδιο χρόνο με το NMap, η ευκολία χρήσης του υπερτερεί σε σύγκριση με άλλα πακέτα διότι παρέχει ένα γραφικό περιβάλλον μέσα από browser, έτσι είναι διαθέσιμο σε όλες τις πλατφόρμες. Ανακάλυψε όλες τις ευπάθειες όπως και το NMap και τις εμφάνισε με καλαίσθητες αναφορές εύκολα αναγνώσιμες από έναν αρχάριο χρήστη.

Το metasploit αναγνωρίζετε ως virus από το antivirus τον windows και αυτό λόγο των exploits που περιέχει, έτσι για την εκτέλεση του έπρεπε να απενεργοποιήσουμε το antivirus. Είναι αρκετά δύσχρηστο παρόλο που παρέχει γραφικό περιβάλλον το όποιο παρουσιάζει τις ανιχνεύσιμες ευπάθειες, στην συνεχεία για να προχωρήσεις την δοκιμή τους πρέπει να εκτελέσεις εντολές στην consola του. Έχει μεγάλη βάση πληροφοριών και ανανεώνεται τακτικά. Συγκριτικά με το NMap μπορούμε να πούμε ότι είναι κοντά σε θέματα απόδοσης.

Το GFI LanGuard κοστίζει 100€ ανά έτος, σε θέματα απόδοσης όμως κόλλησε αρκετές φορές μέχρι να ολοκληρώσει την δοκιμή, είναι ένα μικρό σχετικά πρόγραμμα που η κύρια χρήση του είναι ο έλεγχος των εκδόσεων των λογισμικών και η εγκατάσταση αναβαθμίσεων σε αυτά εγκαθιστώντας έναν agent για τον έλεγχο τους.

Το Nexpose είναι ένα δωρεάν εργαλείο και συγκριτικά με τα εμπορικά προϊόντα αναγνώρισε περισσότερες απειλές και ευπάθειες όπως εύκολους κωδικούς, εκτιθέμενες υπηρεσίες, κτλ παρέχοντας στο διαχειριστή μια ολοκληρωμένη εικόνα για το δίκτυο και τα συστήματα του. Έχει κάποια δυσκολία στην ρύθμιση σύνδεσης με το metasploit όπου ο χρήστης πρέπει να έχει αρκετές γνώσεις που αφορούν τα δίκτυα και καλή γνώση της metasploit consoles.

Το CoreImpact είναι το ακριβότερο λογισμικό για ελέγχους ευπαθειών. Το μεγάλο πλεονέκτημα του είναι η αυτοματοποίηση, δυστυχώς δεν μπορέσαμε να το δοκιμάσουμε διότι η εταιρία δεν μας έδινε demo version, έτσι η αξιολόγηση του έγινε βάση πληροφοριών που ανακτηθήκαν από το internet. Συγκριτικά με τα άλλα εργαλεία είναι το ποιο ολοκληρωμένο προσπαθώντας να

αντικαταστήσει τον ανθρώπινο παράγοντα και να ανακαλύψει όλες τι ευπάθειες σε όλα τα επίπεδα του πληροφοριακού συστήματος. Δυστυχώς είναι απροσπέλαστο σε μικρές εταιρίες μιας και το κόστος του υπερβαίνει τις 30000€.

Συγκεντρωτικά τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες:

Όνομα εργαλείου	Απόδοση	Κόστος	Ευκολία Χρήσης	Πληρότητα
NMap – ZeNMap	5	5	3	4
OpenVas	4	5	4	3
Nessus	5	3	5	3
Metasploit	5	5	3	4
GFI LanGuard	3	3	4	2
Nexpose	4	5	3	4
Core Impact	5	1	5	4

Όνομα εργαλείου	Αποτέλεσμα
NMap – ZeNMap	17
OpenVas	16
Nessus	16
Metasploit	17
GFI LanGuard	12
Nexpose	17
Core Impact	15

Παρατηρούμε ότι τα εργαλεία ανοιχτού κώδικα εκτελούν εξίσου αποδοτικά τις δοκιμές διείσδυσης με τα εμπορικά προϊόντα που παρέχουν ένα περισσότερο φιλικό περιβάλλον στον χρήστη. Όλα τα εμπορικά προϊόντα παίρνουν χαμηλότερη βαθμολογία λόγο του κόστους άλλα και την μη υποστήριξη εγκατάστασης σε άλλες πλατφόρμες.

Τέλος πρέπει να αναφέρουμε την απόδοση του NMap στην ανίχνευση των ευπαθειών άλλα και την ταυτότητα του σε σύγκριση με τα άλλα εργαλεία. Σε stealth mode ανακάλυψε τις ίδιες ευπάθειες με τα άλλα πακέτα χωρίς να γίνει αντιληπτό από το IDS. Αυτό που λείπει από το nap είναι ένα περιβάλλον χρήσης που να παρέχει την ικανότητα επιλογής των παραμέτρων και την παρουσίαση των αποτελεσμάτων σε μια μορφή αναφοράς.

3.6.2 Αξιολόγηση συλλογών εργαλείων

Ομοίως με τα εργαλεία δοκιμάσαμε τις συλλογές εργαλείων εκτελώντας τις σε εικονικό περιβάλλον. Θα αξιολογήσουμε τις συλλογές με τα ιδία κριτήρια που χρησιμοποιήσαμε για τα εργαλεία με την μονή διαφορά ότι θα αφαιρέσουμε το κόστος διότι όλα τα live cdrom’s είναι ανοιχτού λογισμικού και συνεπώς δωρεάν.

Το Kali Linux έχει τα περισσότερα εργαλεία ελέγχου τρωτότητας και το γρηγορότερο περιβάλλον χρήσης, εύκολα δομημένο έτσι ώστε ο χρήστης να βρίσκει γρήγορα αυτό που χρειάζεται. Βασισμένο στο debian stable παρέχει σταθερά εργαλεία δοκιμών διείσδυσης και υποστηρίζει ασύρματα δίκτυα, Bluetooth σάρωση και εργαλεία για έλεγχο αρχείων, δεδομένων και πακέτων. Το περιβάλλον χρήσης του είναι παραμετροποιήσιμο και είναι μεταγλωττισμένο σε πολλές γλώσσες.

Το Knoppix-STD παρόλο που ξεκίνησε σαν μια πολλά υποσχόμενη διανομή έμεινε πίσω με ένα απαρχαιωμένο περιβάλλον χρήσης με τα βασικά εργαλεία ελέγχων, ενώ ήταν η ποιο αργή διανομή.

Το περιβάλλον χρήσης της είναι παλιό και τα περισσότερα εργαλεία εκτελούνται σε τερματικό εισάγοντας χειροκίνητα τους παραμέτρους. Συγκριτικά με τις άλλες διανομές έρχεται τελευταία στην κατάταξη μιας και έχει σταματήσει η ανάπτυξη της.

Το BlackArch παρέχει λιγότερα εργαλεία από το Kali Linux ενώ δεν υποστηρίζει αρκετές πλατφόρμες υλικού σε σύγκριση με άλλες διανομές. Παρέχει όμως εύκολη εγκατάσταση σε περιβάλλοντα εικονοποίησης όπως είναι το VMWare και το KVM. Ο Window manager του είναι λιγότερο εύχρηστος από άλλες διανομές, ενώ δεν παρέχει ικανότητα παραμετροποίησης.

Το BackBox Linux είναι μια έκδοση Ubuntu με εργαλεία δοκιμών διείσδυσης και διαφορά πακέτα ελέγχου ασφάλειας. Χρησιμοποιεί τον XFce window manager που είναι αρκετά λιτός και γρήγορος. Έχει ενημερωμένα εργαλεία και νέους οδηγούς για τον βέλτιστη διαχείριση του υλικού. Το μενού ελέγχου του είναι δομημένο όπως το Kali Linux και παρέχει γρήγορη πρόσβαση στο τελικό χρήστη.

Το ParrotOS δεν είναι τόσο δημοφιλές όπως οι άλλες διανομές σε θέματα ασφάλειας, άλλα παρέχει ένα όμορφο και εύχρηστο περιβάλλον για την εκτέλεση των δοκιμών. Τα μειονεκτήματα της διανομής είναι ο λίγο Άργος στην απόκριση window manager και η μη υποστήριξη άλλης πλατφόρμας έκτος από τις x86,x64. Το μεγάλο πλεονέκτημα του είναι ο package manager SNAP μέσω του οποίου μπορούμε να εγκαταστήσουμε οποιοδήποτε πακέτο.

Το Bugtraq 2 – black widow είναι μια διανομή προσανατολισμένη στο hacking παρέχοντας όλα τα εργαλεία για την ανίχνευση ευπαθειών άλλα και πακέτα για παραβίαση (Exploitation) ενός συστήματος. Παρέχετε για όλες τις πλατφόρμες ενώ υποστηρίζει και εργαλεία εγκληματολογικής-δικανικής ανάλυσης. Υποστηρίζει ακόμα εργαλεία ελέγχου GSM, WIFI, RFID, Bluetooth και συγκριτικά με τις άλλες διανομές ήταν η ποιο ολοκληρωμένη. Το μεγάλο πλεονέκτημα της είναι η ύπαρξη σεναρίων και η εκτέλεση τους μέσω του μενού, έτσι δεν χρειάζεται να μελετήσουμε για τους τρόπους και την εκτέλεση των δοκιμών, απλά επιλεγούμε το σενάριο και η διανομή εκτελεί όλα τα απαραίτητα εργαλεία.

Συγκεντρωτικά τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες:

Όνομα εργαλείου	Απόδοση	Ευκολία Χρήσης	Πληρότητα
Kali Linux	5	4	5
Knoppix-STD	3	1	1
BlackArch	4	4	3
BackBox	4	4	4
ParrotOS	4	5	4
Bugtraq 2	5	5	5

Όνομα εργαλείου	Αποτέλεσμα
Kali Linux	14
Knoppix-STD	5
BlackArch	11
BackBox	12
ParrotOS	13
Bugtraq 2	15

Όλες οι διανομές ήταν εξίσου αποδοτικές έκτος από το Knoppix-STD. Στην ευκολία χρήσης το bugtraq 2 μαζί με το ParrotOS ήταν το ποιο γρήγορο και εύχρηστο. Σε θέματα

απόδοσης/πληρότητας το bugtraq 2 υπερτερεί από όλες τις διανομές παρέχοντας μια τεράστια γκάμα εργαλείων για όλες τις δοκιμές και ελέγχους. Όπως βλέπουμε και στην τελική κατάταξη το Bugtraq είναι στην κορυφή συγκριτικά με τις άλλες διανομές, ενώ ακολουθούν το Kali Linux και το ParrotOS.

Β. Πρακτικο Μερος

Κεφάλαιο 4
Δοκιμές Σάρωσης στη Πράξη

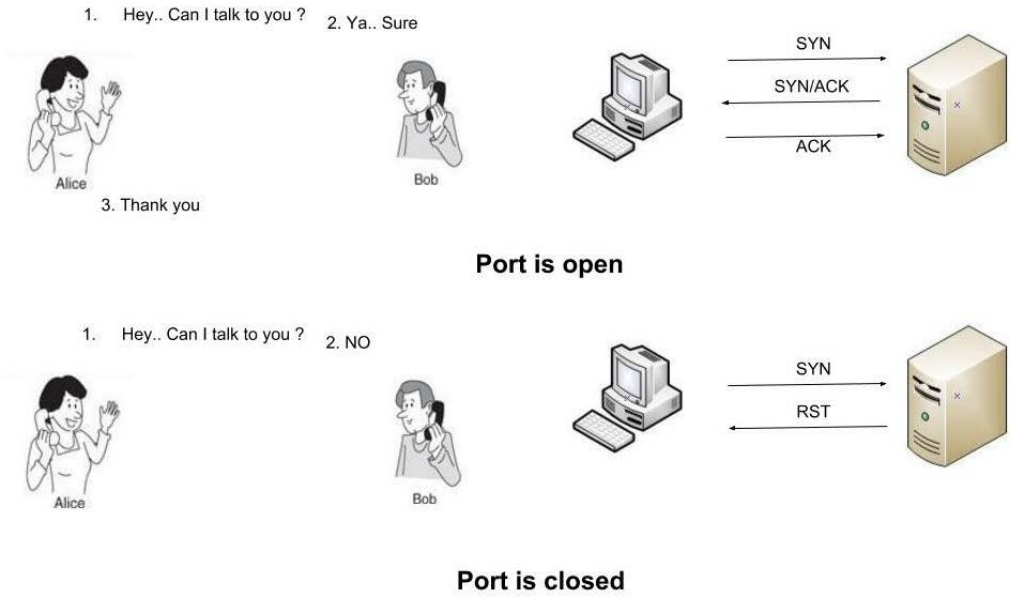
Στο πρώτο μέρος της διατριβής προσεγγίσαμε θεωρητικά το θέμα των δοκιμών διείσδυσης, αναλύσαμε του κινδύνους και τις απειλές σε ένα πληροφοριακό σύστημα και με ποιο τρόπο οι δοκιμές μπορούν να προστατεύσουν τον οργανισμό από ζημίες προερχόμενες από ευπάθειες των λογισμικών. Στο κεφάλαιο αυτό θα περιγράψουμε τις λειτουργίες του εργαλείου δοκιμών διείσδυσης NMap το οποίο θα χρησιμοποιήσουν ως backend για την δημιουργία του φιλικού περιβάλλοντος στο πρακτικό μέρος. Ποιο αναλυτικά θα περιγράφουν οι τεχνικές που χρησιμοποιεί το εργαλείο, οι παράμετροι για την βέλτιστη εκτέλεση του και τέλος το πειραματικό περιβάλλον πάνω στο οποίο θα εκτελεστούν οι δοκιμές διείσδυσης.

4.1 Σάρωση Δικτύων και Πληροφοριακών Συστημάτων

Όπως αναφέραμε και στο θεωρητικό μέρος ο σκοπός των δοκιμών διείσδυσης είναι να χαρτογραφήσουν το δίκτυο ενός οργανισμού ανακαλύπτοντας όλα τα πληροφοριακά συστήματα που το αποτελούν. Στην συνέχεια σαρώνουν τα συστήματα ανιχνεύοντας τις υπηρεσίες που παρέχουν και αναγνωρίζοντας πιθανές ευπάθειες. Στο κεφάλαιο αυτό θα περιγράψουμε τις τεχνικές που χρησιμοποιούνται έτσι ώστε να είναι επιτυχής μια δοκιμή διείσδυσης αλλά και να μη ανιχνευτεί από τους μηχανισμούς άμυνας του πληροφοριακού συστήματος.

4.1.1 Τεχνικές Σάρωσης

Πριν την εκτέλεση της σάρωσης γίνεται μια ανάλυση στο δίκτυο στόχο έτσι ώστε να επιλέγουν οι κατάλληλες τεχνικές. Μετά την ανάλυση αποφασίζουμε σε ποιο μέρος του δικτύου θα εστιάσουμε, ποιες υπηρεσίες θα σαρώσουμε, τι τύπου υπηρεσίες θα ανιχνεύσουμε και τέλος το εύρος της σάρωσης. Η ανίχνευση των υπηρεσιών γίνεται με προσπάθεια σύνδεσης σε ανοιχτές θύρες προσποιούμενη την διαδικασία του client. Στην παρακάτω εικόνα φαίνεται ο τρόπος ανίχνευσης των υπηρεσιών και η ροή των δεδομένων από και προς το σύστημα-στόχο :



Εικόνα 4-1 - TCP connection - Full Scan [3]

Έτσι προκύπτουν οι παρακάτω τεχνικές :

Πλήρη σάρωση: το εργαλείο διείσδυσης σαρώνει όλα τα μέρη του πληροφοριακού συστήματος με πλήρη ανίχνευση θυρών 1-64324 σε όλα τα πρωτόκολλα δικτύου (tcp,udp). Η τεχνική αυτή είναι η πλέον διαδεδομένη διότι ανιχνεύει οτιδήποτε είναι ενεργό και παρέχει υπηρεσία και συνεπώς μπορεί να αποτελεί ευπάθεια. Τα μειονεκτήματα της τεχνικής αυτής είναι ο χρόνος εκτέλεσης της, ο οποίος ανάλογα το μέγεθος του δικτύου μπορεί να εκτελείτε για μέρες η εβδομάδες και η μεγάλη πιθανότητα ανίχνευσης της και η παρεμπόδιση της από συστήματα ανίχνευσης επιθέσεων η από τοίχους προστασίας.

Στοχευμένη σάρωση: με την συγκεκριμένη τεχνική σαρώνουμε συγκεκριμένα hosts για συγκεκριμένα πρωτόκολλα δικτύων. Με τον τρόπο αυτό ανιχνεύουμε τις ευπάθειες ενός διακομιστή χωρίς να απαιτείτε χρόνος με σκοπό να μη γίνει αντιληπτή η διαδικασία σάρωσης.

Σάρωση πρωτοκόλλου TCP/UDP με συγκεκριμένες παραμέτρους: η τεχνική αυτή χρησιμοποιεί τα RFC flags του πρωτοκόλλου TCP για την ανίχνευση των ανοιχτών θυρών με κύριο στόχο να μην γίνει αντιληπτή από το σύστημα-στόχος. Το πλεονέκτημα αυτών των σαρώσεων είναι η ταχύτητα και η μη ανίχνευση τους από φίλτρα πακέτων η τοίχων προστασίας.

Σάρωση ενεργών συστημάτων (PING): με την σάρωση αυτή ανιχνεύουμε τα προσβάσιμα συστήματα στο δίκτυο-στόχος. Η τεχνική αυτή χρησιμοποιείτε για την χαρτογράφηση του δικτύου ιχνηλατώντας τα συστήματα που είναι ενεργά (up) μια συγκεκριμένη στιγμή. Είναι μια γρήγορη τεχνική που εμφανίζει άμεσα αποτελέσματα. Το μειονέκτημα της είναι ότι χρησιμοποιεί το πρωτόκολλο δικτύου ICMP που συνήθως είναι απενεργοποιημένο από τα τοίχοι προστασίας.

Γρήγορη σάρωση: με την τεχνική αυτή γίνεται έλεγχος μόνο σε θύρες γνωστών υπηρεσιών, όπως είναι το web, mail, ssh, κτλ. Με αυτό τον τρόπο μειώνεται ο χρόνος ελέγχου αλλά δεν ανιχνεύονται όλες οι θύρες που πιθανόν να είναι ανοιχτές.

Σάρωση με απόκρυψη: με αυτήν την τεχνική η σάρωση κρύβει (spoofing) το διακομιστή που πυροδοτεί την σάρωση, συνεπώς το σύστημα-στόχος δεν μπορεί να αναγνωρίσει τον πραγματικό σαρωτή και αρά δεν μπορεί να τον αποκλείσει με κάποιο rule στο τοίχος προστασίας.

Σάρωση με αναλυτικές πληροφορίες (Verbose): με αυτή την τεχνική εμφανίζονται αναλυτικά οι πληροφορίες για τις υπηρεσίες που ανιχνεύονται καθώς και το λειτουργικό που εκτελείται στο σύστημα-στόχος.

Ο συνδυασμός των παραπάνω τεχνικών επιφέρει το βέλτιστο αποτέλεσμα ως προς την ανίχνευση περισσότερων ευπαθειών και συνεπώς καλύτερη εικόνα για τους κινδύνους που απειλούν τα πληροφοριακά μας συστήματα.

4.1.2 Αρχεία καταγραφών

Για την καλύτερη ανάλυση των αποτελεσμάτων είναι απαραίτητη μια καλή αναφορά που να εστιάζει στα κυριότερα σημεία της σάρωσης. Στα προγράμματα με γραφικό περιβάλλον αυτό γίνεται χρησιμοποιώντας ειδικά charts στα οποία παρουσιάζονται οι ευπάθειες και οι πληροφορίες για τα συστήματα στόχους. Η πληροφορία αυτή μπορεί να είναι μεγάλου μεγέθους ειδικά όταν πρόκειται για σάρωση ολοκλήρων δικτύων class b,c συνεπώς επιβάλετε η αποθήκευση των αποτελεσμάτων για μετέπειτα ανάλυση.

Στα συστήματα Linux με εργαλεία γραμμής εργασιών αυτό γίνεται χρησιμοποιώντας αρχεία καταγράφων (Log Files). Κατά την εκτέλεση μιας δοκιμής σάρωσης ανακατευθύνουμε το αποτέλεσμα σε ένα αρχείο στο οποίο καταγράφεται το αποτέλεσμα της σάρωσης, στην συνέχεια αναλύουμε το αρχείο διαβάζοντας το με κάποιο text editor.

Στο προηγούμενο κεφάλαιο αναφερθήκαμε σε προγράμματα οπού παρουσιάζουν τις αναφορές σάρωσης σε pdf, σε html, η σε κάποιο δικό τους format. Είναι σημαντικό λοιπόν μετά το πέρας μιας σάρωσης να μπορεί ο χρήστης να αναλύσει τα δεδομένα και να τα παρουσιάσει με λυτό και κατανοητό τρόπο εστιάζοντας στα κρίσιμα σημεία των ευπαθειών.

4.1.3 Παράκαμψη μηχανισμών άμυνας

Για να ολοκληρωθεί μια δοκιμή σάρωσης πρέπει να μη γίνει αντιληπτή από το σύστημα-στόχος.

Τα συστήματα ανίχνευσης σάρωσης ονομάζονται IDS (Intrusion Detection System) και εκτελούνται πάνω στα ενεργά συστήματα δικτύου «ακούγοντας» τις γραμμές δεδομένων και ανιχνεύοντας κακόβουλες προσπάθειες η επιθέσεις, στην συνέχεια ενημερώνουν τον διαχειριστή και αποκλείουν το κακόβουλο σύστημα με ειδικές εγγραφές στο τοίχος προστασίας (firewall rules).

Στην περίπτωση που μια δοκιμή σάρωσης γίνει αντιληπτή, το εργαλείο που την εκτελεί δεν επιστρέφει αποτελέσματα και εμφανίζει όλα τα συστήματα offline. Ο μονός τρόπος να συνεχιστεί η δοκιμή είναι να αλλάξει ip address το σύστημα-πηγή που εκτελεί την δοκιμή διείσδυσης. Αυτή η διαδικασία ονομάζετε ip spoofing κατά την οποία ο κακόβουλος χρήστης δημιουργεί πακέτα με ψευδή διεύθυνση αποστολέα, έτσι εκτελεί την σάρωση και μόλις πάρει απάντηση από το σύστημα στόχο αποστέλλει νέο πακέτο με διαφορετικό αποστολέα μπερδεύοντας το σύστημα ανίχνευσης.

Το εργαλείο δοκιμών διείσδυσης NMap έχει ειδικούς παραμέτρους που ενεργοποιούν την απόκρυψη της πραγματικής διεύθυνσης.

Άλλοι τρόποι παράκαμψης των μηχανισμών άμυνας είναι η χρήση στενευμένων τεχνικών σάρωσης. Με αυτό τον τρόπο δεν εκτελούμε μια πλήρη σάρωση η οποία θα γίνει αντιληπτή από τα συστήματα άμυνας λόγω του όγκου δεδομένων που διακινούνται και του χρόνου ολοκλήρωσης αλλά μια στενευμένη σάρωση σε ένα σύστημα την φορά και με διαφορετικούς παραμέτρους. Έτσι τα συστήματα ανίχνευσης θεωρούν ότι είναι προσπάθειες επικοινωνίας από εξουσιοδοτημένους χρήστες και δεν αποκλείουν την σάρωση.

Προφανώς τα συστήματα ανίχνευσης γίνονται όλο και εξυπνότερα χρησιμοποιώντας δεδομένα από άλλες επιθέσεις, όμως πάντα υπάρχει κάποιος τρόπος που δεν έχει προβλεφθεί αφήνοντας εκτεθειμένες ευπάθειες που στην συνέχεια γίνονται απειλές για τα πληροφοριακά συστήματα του οργανισμού.

4.2 Το εργαλείο NMAP [10]

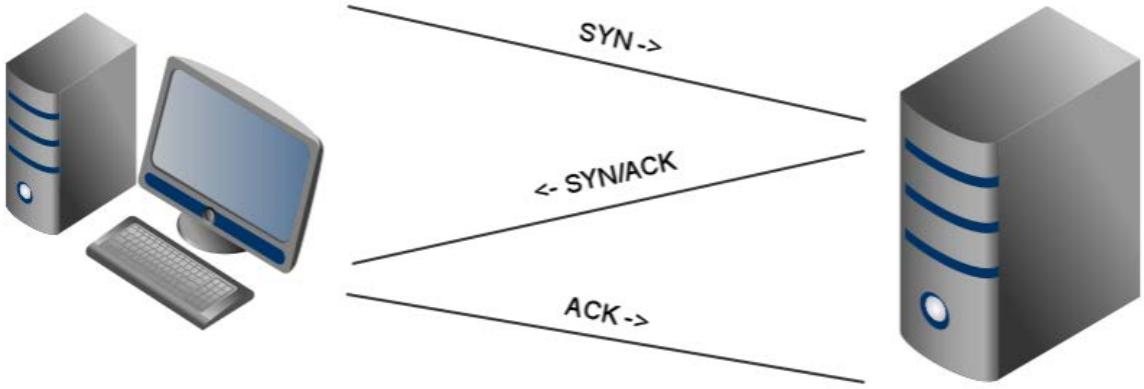
Στο κεφάλαιο αυτό θα περιγράφει το εργαλείο NMap . Θα αναλυθούν οι τεχνικές και οι παράμετροι που χρησιμοποιεί έτσι ώστε να σαρώσει επιτυχώς δίκτυα και πληροφοριακά συστήματα. Το NMap θα χρησιμοποιεί ως backend για την διενέργεια των δοκιμών διείσδυσης στο πρακτικό μέρος της διατριβής.

4.2.1 Γενικά

Το εργαλείο NMap είναι ένα ελεύθερο λογισμικό που σχεδιάστηκε για την χαρτογράφηση δικτύων. Τα αρχικά του N, Map αναλύονται σε Network Mapping. Το NMap σαρώνει και ανιχνεύει όλα τα πληροφοριακά συστήματα που αποτελούν ένα δίκτυο και ανιχνεύει τα λειτουργικά συστήματα που

εκτελούνται, τις εφαρμογές που παρέχουν υπηρεσίες, τα ενεργά συστήματα δικτύου καθώς και όλες τις ανοιχτές θύρες μέσω των οποίων συνδέονται οι clients και πιθανόν να είναι ευπάθειες η απειλές για τον οργανισμό.

Το NMap χρησιμοποιεί IP packages για την ανίχνευση των ανοιχτών θυρών, παίρνει το ρόλο του client και συνδέετε σε όλες τις θύρες στα συστήματα-στόχος. Ο τρόπος σύνδεσης και τα IP πακέτα για το πρωτόκολλο tcp φαίνεται στην παρακάτω εικόνα:



Εικόνα 4-2 - TCP/IP - Three-way handshake

Ο τρόπος λειτουργίας είναι η αποστολή raw ip πακέτων σε όλες τις θύρες με το ip flag = SYN, αν το σύστημα παρέχει αυτή την υπηρεσία, «ακούει» δηλαδή σε αυτή την θύρα απαντά με ενα SYN/ACK οπότε το NMap ανιχνεύει ότι η συγκεκριμένη θύρα είναι ανοιχτή. Αν δεν επιστραφεί απάντηση σημαίνει ότι το σύστημα δεν ακούει σε αυτή την θύρα και χαρακτηρίζεται ως κλειστή.

Εκτός όμως από τις open/closed θύρες το εργαλείο έχει την ικανότητα να ανιχνεύσει αν η θύρα βρίσκεται πίσω από τοίχος προστασίας η αν είναι native πάνω σε κάποιο σύστημα. Οπότε ανάλογα με τα ACK πακέτα που λαμβάνει χαρακτηρίζει τις θύρες στις παρακάτω κατηγορίες:

Open: ανοικτή θύρα που δέχεται συνδέσεις και αρά το σύστημα παρέχει αυτή την υπηρεσία.

Closed: κλειστή θύρα που δεν δέχεται συνδέσεις και αρά δεν παρέχεται η υπηρεσία.

Filtered: αν το NMap λάβει πακέτο ACK αλλά δεν μπορεί να αποφασίσει αν είναι ανοικτή η κλειστή διότι δεν ολοκληρώνετε το tcp three-way handshake τότε χαρακτηρίζει την θύρα ως φιλτραρισμένη.

Non-filtered: η περίπτωση οπού η θύρα απαντά στο πακέτο SYN αλλά το NMap δεν μπορεί να αποφασίσει αν είναι ανοικτή η όχι.

Open/Closed filtered: όταν το NMap δεν μπορεί να χαρακτηρίσει μια θύρα ανοιχτή, κλειστή η φιλτραρισμένη.

Το NMap εκτελείται από command line και απαιτεί root privilege για να είναι ενεργές όλες οι λειτουργίες του. Δέχεται ορίσματα τις μορφής :

```
NMap [είδος σάρωσης] [παράμετροι] [στόχος]
```

και δηλώνουν το είδος σάρωσης, την απόκρυψη πληροφοριών προς το σύστημα-στόχο, την ενεργοποίηση plugins και τέλος το σύστημα η το δίκτυο στόχο. Ο στόχος δίδεται στο NMap με

μορφή hostname, ip address η network ip, ετσι η εκτέλεση του εργαλείου με παράμετρο 192.168.1.0/24 θα ξεκινήσει την σάρωση στα 256 hosts του δικτύου 192.168.1.0.

Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα σάρωσης με μοναδική παράμετρο το σύστημα-στόχο scanme.nmap.org :

```
[root@xl ~]# nmap scanme.nmap.org

Starting Nmap 5.51 ( http://nmap.org ) at 2019-02-23 20:16 EET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
445/tcp   filtered microsoft-ds
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 14.64 seconds
[root@xl ~]#
```

Εικόνα 4-3 - NMap scan example

4.2.2 Τεχνικές σάρωσης στη πράξη [53]

Το NMap μας δίνει την ικανότητα να ορίσουμε τεχνικές σάρωσης των δοκιμών διείσδυσης, παρακάτω αναφέρονται οι κυριότερες τεχνικές που θα χρησιμοποιήσουμε στο boot cd και με ποιους παραμέτρους καθορίζονται.

TCP, UDP Connect (-sT, -sU): ο default τρόπος εκτέλεσης του NMap χρησιμοποιεί rfc calls για να συνδεθεί στις θύρες του συστήματος-στόχου ανοίγοντας connection για κάθε υπηρεσία. Το NMap στέλνει TCP connections σε κάθε θύρα του στόχου περιμένοντας απάντηση με ACK πακέτο. Αντίστοιχα σε UDP πρωτόκολλο ελέγχοντας για υπηρεσίες όπως είναι radius, dhcp, κτλ. Η τεχνική αυτή είναι εύκολα ανιχνεύσιμη από το σύστημα-στόχος αναγνωρίζοντας όλες τις συνδέσεις και καταγράφοντας τις σε αρχεία καταγραφής. Πολλά συστήματα σταματούν αυτές τις σαρώσεις αμέσως μόλις ξεκινήσουν ανιχνεύοντας τις συνεχείς προσπάθειες σύνδεσης.

TCP SYN (-sS): επιλέγοντας αυτή την τεχνική το NMap δεν αποστέλλει το πακέτο RST που ισοδυναμεί με το κλείσιμο της θύρας. Έτσι ουσιαστικά όλες οι θύρες που ανιχνεύονται παραμένουν ανοιχτές εφόσον δεν αποστέλλεται πακέτο RST . Η τεχνική αυτή είναι δύσκολα ανιχνεύσιμη εφόσον οι θύρες δεν κλείνονται από το NMap θα πρέπει να τις κλείσει το σύστημα-στόχος μετά το πέρας του time out, ετσι δεν καταγράφονται στα αρχεία καταγραφής και είναι δύσκολο να εντοπιστούν.

TCP FIN (-sF), TCP NULL (-sN), TCP Xmas (-sX): όπως και στις παραπάνω τεχνικές οι τρεις αυτές σαρώσεις χρησιμοποιούν τα RFC flags του TCP πρωτοκόλλου για να ανιχνεύσουν τις ανοιχτές θύρες. Ποιο αναλυτικά η τεχνική FIN ενεργοποιεί το flag FIN με αποτέλεσμα να προσπερνά τα firewalls τα οποία θεωρούν ότι η σύνδεση κλίνει και έτσι το επιτρέπουν. Η τεχνική NULL δεν ενεργοποιεί κανένα TCP flag αλλά αποστέλλει συγκεκριμένα πακέτα με σκοπό την ανιχνεύσει των filtered ports. Τέλος η τεχνική Xmas ενεργοποιεί τα flags URG, FIN, PSH ξεγελώντας το σύστημα-στόχο επιβάλλοντας του

να απαντήσει με RST πακέτο για τις ανοιχτές θύρες. Τα πλεονεκτήματα των παραπάνω τεχνικών είναι η αύξηση της ταχύτητας της σάρωσης και συνεπώς η δυσκολία εντοπισμού της. Επίσης μπορούν να ξεγελούν τα τοίχοι προστασίας και τα συστήματα IDS με τελικό αποτέλεσμα την ανιχνεύσει όλων των υπηρεσιών/θυρών.

TCP ACK (-sA): με την τεχνική αυτή το NMap ανιχνεύει τους κανόνες (rules) του τοίχου προστασίας. Είναι σημαντικό να γνωρίζουμε τις θύρες που προστατεύονται από firewall διότι σε περίπτωση αποτυχίας της προστασίας είμαστε εκτεθειμένοι στο internet. Το NMap στέλνει ένα πακέτο ACK σε όλες τις θύρες, αν το σύστημα-στόχος απαντήσει με πακέτο RST χαρακτηρίζει την θύρα μη φιλτραρισμένη αλλιώς η θύρα φιλτράρετε από τοίχος προστασίας.

TCP Window (-sW): είναι ακριβώς ίδια τεχνική με την TCP ACK με την διαφορά τον έλεγχο του πεδίου window στο RST πακέτο που λαμβάνει το NMap. Αν το πακέτο RST δεν είναι μηδέν η θύρα χαρακτηρίζεται ανοικτή, αν είναι μηδέν τότε η θύρα είναι κλειστή και αν δεν επιστραφεί απάντηση τότε η θύρα είναι φιλτραρισμένη.

PING Scan (-sn): εμφάνιση όλων των ενεργών συστημάτων σε ένα δίκτυο. Πολλές φορές πριν ξεκινήσει μια δοκιμή διείσδυσης είναι καλό να γνωρίζουμε ποια συστήματα είναι ενεργά έτσι ώστε να περιορίσουμε το εύρος της δοκιμής με σκοπό να μην γίνουμε αντιληπτοί. Με την παράμετρο αυτή το NMap αποστέλλει ένα πακέτο ICMP, σε περίπτωση που ένα σύστημα είναι ενεργό απαντά με ένα ICMP echo request οπότε χαρακτηρίζετε ενεργό.

Fast scan (-F): σκοπός της τεχνικής αυτής είναι η αύξηση της ταχύτητας σάρωσης. Το NMap σαρώνει μόνο τις γνωστές θύρες υπηρεσιών όπως ssh(22), mail(25), www(80), κτλ.

IPv6 Scan (-6): σαρώνει τα συστήματα-στόχους χρησιμοποιώντας ipv6 πρωτόκολλο. Πολλές υπηρεσίες υποστηρίζουν την έκδοση 6 του tcp οι οποίες δεν ανιχνεύονται όταν χρησιμοποιήσουμε ipv4. Με αυτήν την παράμετρο ενεργοποιούμε την υποστήριξη ipv6.

Idle decoys scan (-sI, -D): με αυτή την τεχνική το NMap χρησιμοποιεί ένα άλλο σύστημα σαν zombie έτσι ώστε να αποκρύψει την πραγματική ταυτότητα του σαρωτή (spoofing). Αυτό γίνεται αλλάζοντας τα ip headers των πακέτων έτσι ώστε να περιλαμβάνετε η ip address του zombie computer μη επιτρέποντας σε IDS και στο σύστημα-στόχο να ανιχνεύσουν την πραγματική διεύθυνση που προέρχεται η δοκιμή διείσδυσης. Το NMap υποστηρίζει την εισαγωγή πολλών zombies hosts και τον διαμοιρασμό των σαρώσεων σε αυτά, έτσι το σύστημα-στόχος καταγραφεί συνδέσεις από πολλά συστήματα χωρίς να είναι ικανό να διακρίνει τον πραγματικό σαρωτή.

Verbose scan (-sV, -O): με την συγκεκριμένη επιλογή το NMap εμφανίζει πληροφορίες σχετικά με το λειτουργικό σύστημα του στόχου, τις εκδόσεις των υπηρεσιών στις ανοιχτές θύρες και περισσότερες πληροφορίες για το δίκτυο (mac address, system vendor)

Aggressive scan (-A): η συγκεκριμένη τεχνική απαιτεί root privileges στο σύστημα-σαρωτή, διότι δίνει την ικανότητα στο NMap να εκτελέσει εξωτερικά scripts, plugins με τα οποία αυξάνουν τα αποτελέσματα της σάρωσης. Πχ σε περίπτωση που το σύστημα-στόχος παρέχει την υπηρεσία απομακρυσμένης σύνδεσης (ssh) μπορεί να εκτελέσει προσπάθειες εισόδου χρησιμοποιώντας κωδικούς που βρίσκονται σε κάποιο λεξικό (brute force attack). Η άλλο παράδειγμα είναι να εκτελέσει την εντολής traceroute για να ανιχνεύσει το μονοπάτι δικτύου μέσω του οποίου γίνεται η προσπέλαση στο σύστημα.

Συνδυάζοντας τις παραπάνω τεχνικές και τις παραμέτρους μπορούμε να ανιχνεύσουμε, χαρτογραφήσουμε πλήρως ένα δίκτυο πληροφοριακών συστημάτων. Το NMap παρέχει πληθώρα παραμέτρων και επιλογών με κύριο στόχο την βέλτιστη σάρωση χωρίς να γίνουμε αντιληπτοί από το σύστημα-στόχο.

4.3 Πειραματικό περιβάλλον

Πριν την δημιουργία του GUI επιβάλετε η επιλογή ενός πειραματικού περιβάλλοντος έτσι ώστε να εκτελούνται οι δοκιμές. Για τις ανάγκες τις διατριβής επιλέχτηκε ένα πραγματικό δίκτυο που περιλαμβάνει ενεργά στοιχεία, διακομιστές, μονάδες αποθήκευσης, θέσεις εργασίας καθώς και τοίχοι προστασίας/IDS. Στο κεφάλαιο αυτό παρουσιάζεται το πληροφοριακό σύστημα στο οποίο θα γίνουν οι δοκιμές διείσδυσης.

4.3.1 Το πληροφοριακό σύστημα

Το πληροφοριακό σύστημα περιλαμβάνει συνολικά 8 δίκτυα κλάσης C τα οποία βρίσκονται πίσω από ένα τοίχος προστασίας/IDS.

Ποιο αναλυτικά έχουμε:

Εξωτερικός δρομολογητής: συνδέει τα πληροφοριακά συστήματα με το internet. Στο δρομολογητή υπάρχει το τοίχος προστασίας όπου με κανόνες επιλέγονται ποιες υπηρεσίες θα παρέχονται στο εξωτερικό δίκτυο. Επίσης εκτελείται ένα λογισμικό IDS το οποίο ανιχνεύει πιθανές δοκιμές διείσδυσης και τις μπλοκάρει με rules στο τοίχος προστασίας.

Εσωτερικοί δρομολογητές: κάθε εσωτερικό δίκτυο έχει ένα δικό του δρομολογητή ο οποίος εκτελεί χρέη δρομολόγησης χωρίς τοίχος προστασίας.

Non-routed δίκτυα: όντος του οργανισμού υπάρχουν δίκτυα που δεν δρομολογούνται για τις ανάγκες του εσωτερικού ελέγχου (monitoring) άλλα και της απομακρυσμένης πρόσβασης σε διακομιστές και ενεργά στοιχεία. Το δίκτυο αυτό είναι πολύ σημαντικό διότι παρέχει πρόσβαση τερματικά διαχείρισης των διακομιστών συνεπώς προστατεύετε από εξειδικευμένο τοίχος προστασίας παρέχοντας πρόσβαση σε συγκεκριμένες ip addresses.

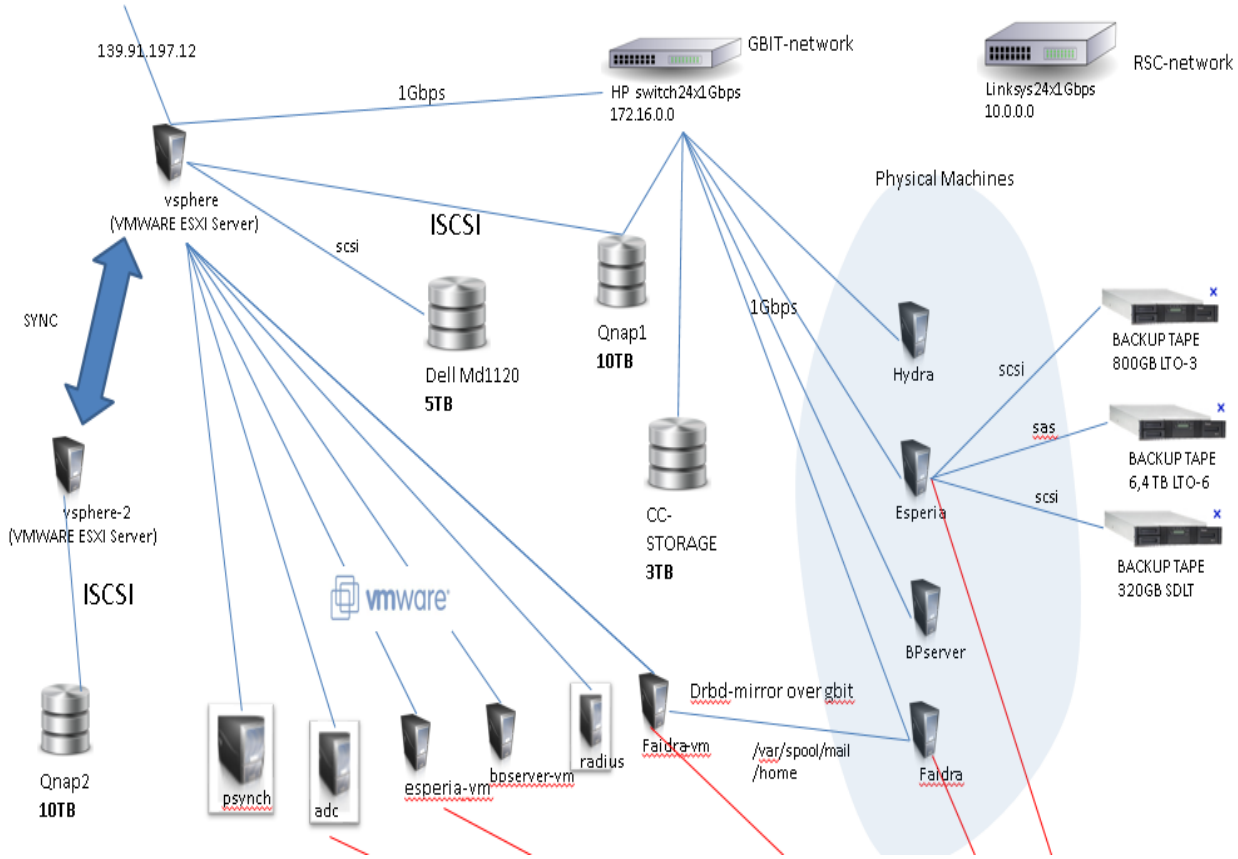
Διακομιστές: στο πληροφοριακό σύστημα υπάρχουν διακομιστές που παρέχουν υπηρεσίες ονοματοθεσίας, ηλεκτρονικού ταχυδρομείου, απομακρυσμένης πρόσβασης (ssh), διαχείρισης αρχείων μέσω cloud, υπηρεσίες καταλόγου (active directory) καθώς και διακομιστές που παρέχουν συγκεκριμένα επιστημονικά λογισμικά όπως HPC, προγράμματα ανάλυσης δεδομένων και προγράμματα ελέγχου επιστημονικών οργάνων. Τα συστήματα αυτά προστατεύονται από δικά τους τοίχοι προστασίας όπως iptables η windows firewall μέσω των οποίων καθορίζονται τα συστήματα ελέγχου και η απομακρυσμένη πρόσβαση σε αυτά.

Συστήματα αποθήκευσης – εφεδρικά αντίγραφα δεδομένων: όλα τα δεδομένα του οργανισμού βρίσκονται σε συστήματα αποθήκευσης διαφόρων κατασκευαστών και είναι πίσω από non-routed δίκτυα, έτσι ώστε να μην υπάρχει καμία σύνδεση με το εξωτερικό δίκτυο πάρα μόνο συνδέσεις από συγκεκριμένους διακομιστές.

Υποδομή εικονοποίησης (Virtualization): για τις ανάγκες του οργανισμού υπάρχει περιβάλλον εικονοποίησης μέσω του οποίου παρέχονται εικονικές μηχανές και υπηρεσίες. Το περιβάλλον αυτό βρίσκεται πίσω από non-routed δίκτυα και προστατεύεται από τοίχος προστασίας που επιτρέπει μόνο συνδέσεις διαχείρισης.

4.3.2 Διασύνδεση πληροφοριακών συστημάτων, τρόποι εκτέλεσης δοκιμών διείσδυσης

Στην παρακάτω εικόνα παρουσιάζονται όλα τα στοιχεία του πειραματικού περιβάλλοντος και με ποιον τρόπο είναι συνδεδεμένα μεταξύ τους.



Εικόνα 4-4 - Πειραματικό περιβάλλον

Όλοι οι διακομιστές συνδέονται στο εξωτερικό δίκτυο μέσω εσωτερικού δρομολογητή, στην εικόνα φαίνεται με κόκκινες γραμμές. Οι μπλε γραμμές δικτυού είναι το εσωτερικό / Non-routed δίκτυο και άρα μη προσπελάσιμο από το εξωτερικό και το εσωτερικό δίκτυο των θέσεων εργασίας.

Αρχικά θα εκτελεστούν οι δοκιμές διείσδυσης από σύστημα έκτος εσωτερικού δικτυού με κύριο στόχο την ανίχνευση των συστημάτων που παρέχουν υπηρεσίες στο εξωτερικό δίκτυο. Αυτό θα γίνει χρησιμοποιώντας τις τεχνικές γρήγορης σάρωσης, decoy scans σε συνδυασμό με τεχνικές FIN, NUL και XMAS έτσι ώστε να μην γίνουμε αντιληπτοί από το IDS του εξωτερικού δρομολογητή.

Στην συνεχεία θα εκτελεστούν τεχνικές aggressive σάρωσης σε μεμονωμένα συστήματα με προσπάθεια ανίχνευσης των υπηρεσιών και brute force attack σε όσες το επιτρέπουν με κύριο στόχο την πρόσβαση στο εσωτερικό.

Τέλος θα εκτελεστούν οι δοκιμές διείσδυσης από το εσωτερικό δίκτυο με στόχο την ανίχνευση των εσωτερικών υπηρεσιών και λογισμικών και απώτερο σκοπό την πρόσβαση στο εσωτερικό δίκτυο των διακομιστών. Οι δοκιμές αυτές έχουν στόχο να καταρρίψουν τα εσωτερικά τοίχοι προστασίας και να ανιχνεύσουν πιθανές ευπάθειες μέσω των ανοικτών θυρών.

Θα γίνει χρήση των τεχνικών verbose scanning σε όλα τα πρωτόκολλα δικτύων (TCP,UDP) καθώς και έλεγχος ipv6.

Κεφάλαιο 5
Υλοποίηση της εφαρμογής

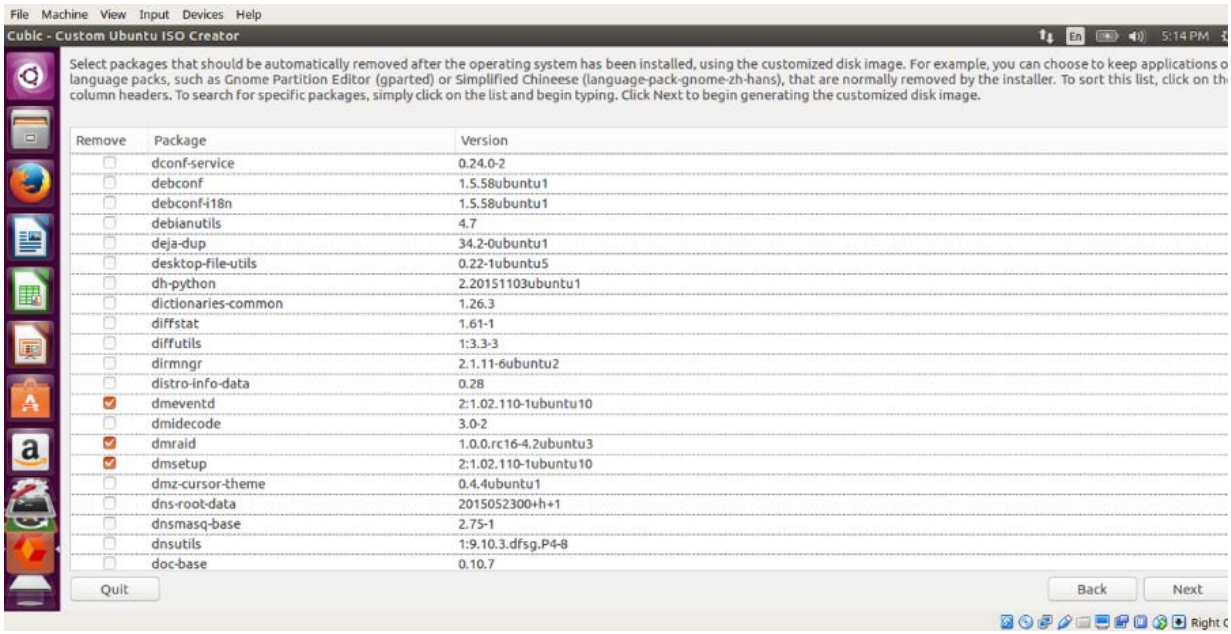
5.1 Bootable Linux Distributions

Ο στόχος της διατριβής είναι η δημιουργία ενός bootable live Linux cdrom συμβατό με υπολογιστές γραφείου, φορητούς και διακομιστές μέσω του οποίου θα εκτελείται το frontend gui του nmap. Το cdrom θα ξεκινά χωρίς να χρειάζεται εγκατάσταση (live), θα αναγνωρίζει το δίκτυο που βρίσκεται ο υπολογιστής και στην συνεχεία θα λαμβάνει ip address εφόσον υπάρχει διακομιστής δυναμικών διευθύνσεων η θα επιτρέπει την εισαγωγή διεύθυνσης μέσω γραφικού περιβάλλοντος. Το περιβάλλον εργασίας πρέπει να είναι φιλικό στο χρήστη εισάγοντας το σύστημα η το δίκτυο στόχος και επιλέγοντας την τεχνική που θα χρησιμοποιηθεί για την δοκιμή διείσδυσης.

Συνεπώς η επιλογή του κατάλληλου Live Linux είναι σημαντική, διότι θα πρέπει να είναι γρήγορο και συμβατό με τους περισσότερους υπολογιστές. Δοκιμάσαμε συνολικά 5 διαφορετικές διανομές που παρέχουν custom δημιουργία live cdrom. Η πιο γρήγορη από όλες είναι το Ubuntu Desktop 18.10 που έτρεξε άψογα σε εικονικό περιβάλλον αναγνωρίζοντας το υλικό και ρύθμισε το δίκτυο χωρίς να παρέμβει ο χρήστης.

Ακόμα εμπεριέχει μια μεγάλη συλλογή λογισμικών, παρέχοντας στους χρήστες τις τελευταίες εκδόσεις σε όλα τα προγράμματα που μας είναι απαραίτητα, όπως είναι το nmap, python modules και libraries, υποστήριξη ipv6 άλλα και εγκατάσταση οδηγών υλικού τρίτων κατασκευαστών έτσι ώστε να αναγνωρίζει υλικό σε φορητούς υπολογιστές και διακομιστές.

Για την δημιουργία [54] του custom live cdrom θα χρησιμοποιήσουμε την εφαρμογή Cubic η οποία εγκαθιστάτε σε μια διανομή Ubuntu δίνοντας μας την ικανότητα να εγκαταστήσουμε εφαρμογές (εικόνα 5-1) στο Linux, να τροποποιήσουμε το configuration του και να εγκαταστήσουμε την δίκια μας εφαρμογή. Στο τέλος μας επιτρέπει να το αποθηκεύσουμε σε ISO format έτσι ώστε να το γράψουμε σε cdrom η σε USB stick.



Εικόνα 5-1 - Cubic

5.2 Bash Scripting

Για να δώσουμε λειτουργικότητα στο GUI θα πρέπει να δημιουργήσουμε μικρά scripts τα οποία θα εκτελούν το nmap με παραμέτρους ανάλογα την τεχνική που θα επιλέξει ο χρήστης, στην συνεχεία θα αποθηκεύει τα αποτελέσματα της σάρωσης σε ένα αρχείο από όπου το GUI θα τα αντλεί και παρουσιάζοντας τα στην οθόνη. Τα αποτελέσματα της κάθε σάρωσης θα παραμένουν στο αρχείο για μελλοντική χρήση η για αποστολή με email στο χρήστη.

Στο κεφάλαιο αυτό θα αναλύσουμε την δομή των scripts και με ποιο τρόπο θα εκτελούν τις λειτουργίες του nmap ανεξάρτητα από το φιλικό περιβάλλον, έτσι ώστε να δοθεί η δυνατότητα στο χρήστη να εκτελέσει τις σαρώσεις μέσα από τερματικό σε περίπτωση που δεν λειτουργεί το γραφικό περιβάλλον.

5.2.1 Linux Bash

Για την δημιουργία των scripts θα χρησιμοποιηθεί το bash [55] που υπάρχει εγκατεστημένο στο Ubuntu. Το bash είναι ένας interpreter μέσω του οποίου μπορούμε να εκτελέσουμε τις εσωτερικές και εξωτερικές εντολές του Linux. Για τις ανάγκες της διατριβής θα χρησιμοποιήσουμε κάποιες δομές ελέγχου έτσι ώστε να βεβαιωθούμε ότι θα εκτελεστεί το Nmap με τις σωστές παραμέτρους.

Η δομή του script σε ψευδοκώδικα είναι η παρακάτω:

```
1 #interpreter
2 if (target host not exists) then print error message and exit
3 print process id
4 ARGUMENTS="…"
5 run nmap with arguments and target host
6 save results to text file
```

1. η επιλογή του interpreter, στην περίπτωση μας /bin/bash.

2. έλεγχος αν η παράμετρος target host έχει αποδοθεί από το GUI, αν όχι το script τερματίζει ενημερώνοντας τον χρήστη ότι δεν έχει ορίσει target host.

3. εκτύπωση του process id της σάρωσης έτσι ώστε να αναγνωρίζεται από το GUI σε περιπτώσεις που θέλουμε να ακυρώσουμε την συγκεκριμένη σάρωση.

4. δημιουργία μεταβλητής όπου αποθηκεύονται οι παράμετροι του nmap ανάλογα με την τεχνική που θα επιλέξει ο χρήστης.

5. εκτέλεση του nmap με παραμέτρους τα δεδομένα της μεταβλητής και το target host που αποδίδεται από το GUI.

6. αποθήκευση των αποτελεσμάτων σε αρχείο για την παρουσίαση τους από το GUI και την μελλοντική διαχείριση τους.

5.2.3 Scripts

Για κάθε τεχνική σάρωσης θα χρησιμοποιηθεί ένα script στο οποίο θα αλλάζουν τα arguments. Όποτε όλα τα scripts θα έχουν την δομή που παρουσιάστηκε στο παραπάνω κεφάλαιο.

Μετατρέποντας των ψευδοκώδικα σε bash commands έχουμε παρακάτω το script που θα χρησιμοποιηθεί για τις τεχνικές σάρωσης :

```
#!/bin/sh
if [ ! "$*" ]; then
      echo "Insert target host."
      exit 127
fi
echo scan id: $$
ARGUMENTS="…… "
nmap $ARGUMENTS $1 > /tmp/nmapgui.$$
```

Με μοναδική διαφοροποίηση τους παραμέτρους (ARGUMENTS) τα οποία καθορίζουν το profile της σάρωσης.

Συνολικά το GUI θα δίνει την ικανότητα στο χρήστη να εκτελέσει τις παρακάτω τεχνικές σάρωσης (profiles):

Fast scan: ARGUMENTS="-F -sV"

Aggressive scan: ARGUMENTS="-sV -sS -A -T4"

Full scan with ACK technique: ARGUMENTS="--stats-every 60s -p 1-65535 -sV -sA -T4"

Full scan with FIN technique: ARGUMENTS="--stats-every 60s -p 1-65535 -sV -sF -T4"

Full scan with NULL technique: ARGUMENTS="-p 1-65535 -sV -sN -T4"

Full scan with TCPSYN technique: ARGUMENTS="-p 1-65535 -sV -sS -T4"

Full scan with WINDOW technique: ARGUMENTS="--stats-every 60s -p 1-65535 -sV -sW -T4"

Full scan with XMAS technique: ARGUMENTS="--stats-every 60s -p 1-65535 -sV -sX -T4"

TCP, UDP Full Scan: ARGUMENTS="--stats-every 60s -sV -sT -sU -T4"

5.3 Python QT5 Framework [56]

Για την δημιουργία του γραφικού περιβάλλοντος χρησιμοποιήθηκε το Python QT5 framework που είναι μια βιβλιοθήκη γραμμένη σε C++ μέσω τις οποίας μπορούμε να δημιουργήσουμε εφαρμογές με γραφικά σε python. Το PyQT έχει άδεια χρήσης LGPL συνεπώς μπορεί να χρησιμοποιηθεί ελευθέρα σε οποιοδήποτε project.

Η βιβλιοθήκη παρέχει ρουτίνες για όλες τις ανάγκες ενός προγραμματιστή, κάνοντας χρήση τους μπορούμε να δημιουργήσουμε μενού και να εισάγουμε λειτουργικότητα που αφορά το δίκτυο, τα αρχεία και την αλληλεπίδραση του χρήστη. Ακόμα παρέχει το QT Designer μέσω του οποίου μπορούμε να δημιουργήσουμε κώδικα σχεδιάζοντας τα παράθυρα και τα μενού, στην συνέχεια χρησιμοποιώντας το PyCharm IDE να εισάγουμε λειτουργικότητα στο κώδικα.

5.3.1 Εγκατάσταση PyQT5

Στη διανομή της επιλογής μας υπάρχει η 4η έκδοση της PyQT, συνεπώς θα πρέπει να εγκαταστήσουμε την έκδοση 5. Η εγκατάσταση γίνετε εύκολα χρησιμοποιώντας την εντολή pip (python package manager) την οποία πρέπει να εγκαταστήσουμε με τον package manager του Ubuntu: ***apt-get install python3-pip***

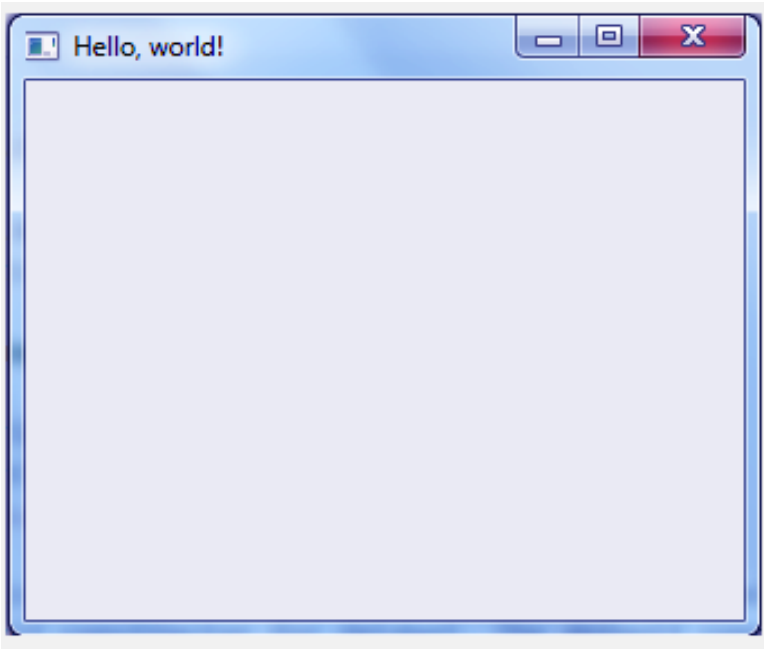
Στην συνέχεια με την εντολή : ***pip3 install PyQT5*** ο manager κατεβάζει το πακέτο PyQT5 και το εγκαθιστά στο κατάλογο με τα modules της python.

5.3.2 Δημιουργία widgets με το PyQT5

Για να δοκιμάσουμε ότι το module έχει εγκατασταθεί σωστά θα δημιουργήσουμε ένα παράθυρο χρησιμοποιώντας την βιβλιοθήκη qtwidgets [57]:

```
1. >>> import sys
2. >>> from PyQt5.QtWidgets import QApplication, QWidget
3. >>> app=QApplication(sys.argv)
4. >>> root=QWidget()
5. >>> root.resize(320,240)
6. >>> root.setWindowTitle('Hello, world!')
7. >>> root.show()
```

Εκτελώντας τον κώδικα εμφανίζεται το παρακάτω window:

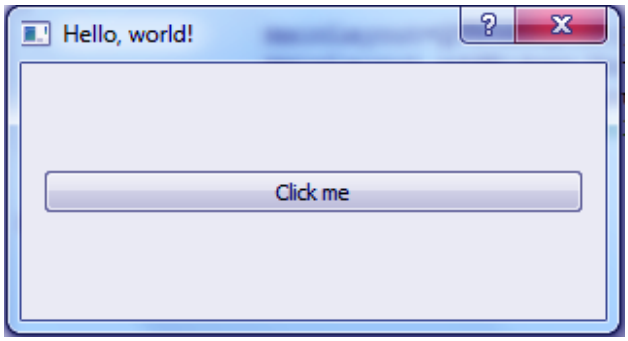


Εικόνα 5-2 - Python Window

Στη συνέχεια θα εισάγουμε λειτουργικότητα δημιουργώντας ένα button το όποιο θα επιλεγούμε και το παράθυρο θα κλείνει:

```
1. >>> from PyQt5.QtWidgets import *
2. >>> import sys
3. >>> class Dialog(QDialog):
4. def slot_method(self):
5. print("Calling the slot")
6. def __init__(self):
7. super(Dialog,self).__init__()
```

```
8.  button=QPushButton("Click me")
9.  button.clicked.connect(self.slot_method)
10. mainLayout=QVBoxLayout()
11. mainLayout.addWidget(button)
12. self.setLayout(mainLayout)
13. self.setWindowTitle("Hello, world!")
14. >>> if __name__=='__main__':
15. app=QApplication(sys.argv)
16. dialog=Dialog()
17. >>> dialog.exec_()
```

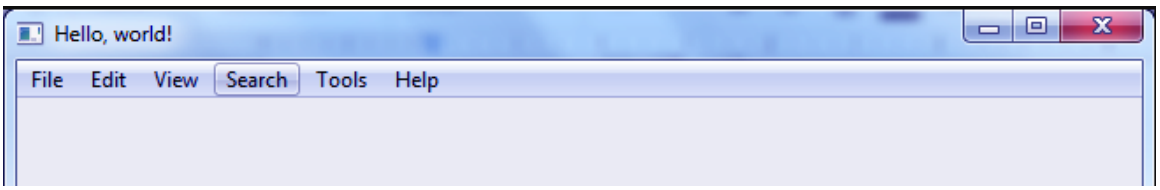


Εικόνα 5-3 - Python button

Επόμενο βήμα είναι η δημιουργία μενού επίλογων :

```
1.  >>> import sys
2.  >>> from PyQt5.QtWidgets import QMainWindow, QApplication, QWidget, QPushButton, QAction
3.  >>> from PyQt5.QtGui import QIcon
4.  >>> from PyQt5.QtCore import pyqtSlot
5.  >>> class App(QMainWindow):
6.  def __init__(self):
7.  super().__init__()
8.  self.title = 'Hello, world!''
9.  self.left = 10
10. self.top = 10
11. self.width = 640
12. self.height = 400
13. self.initUI()
14. def initUI(self):
15. self.setWindowTitle(self.title)
16. self.setGeometry(self.left,self.top,self.width,self.height)
17. mainMenu=self.menuBar()
18. fileMenu=mainMenu.addMenu('File')
19. editMenu=mainMenu.addMenu('Edit')
20. viewMenu=mainMenu.addMenu('View')
21. searchMenu=mainMenu.addMenu('Search')
22. toolsMenu=mainMenu.addMenu('Tools')
23. helpMenu=mainMenu.addMenu('Help')
24. exitButton=QAction(QIcon('exit24.png'), 'Exit', self)
25. exitButton.setShortcut('Ctrl+Q')
26. exitButton.setStatusTip('Exit application')
27. exitButton.triggered.connect(self.close)
28. fileMenu.addAction(exitButton)
29. self.show()
30. >>> if __name__=='__main__':
31. app=QApplication(sys.argv)
```

```
32. ex=App()
```



Εικόνα 5-4 - Python Menus

Συνεπώς κάνοντας χρήση των παραπάνω ρουτινών μπορούμε να δημιουργήσουμε γραφικά και στην συνέχεια με την Python να δώσουμε λειτουργικότητα σε αυτά.

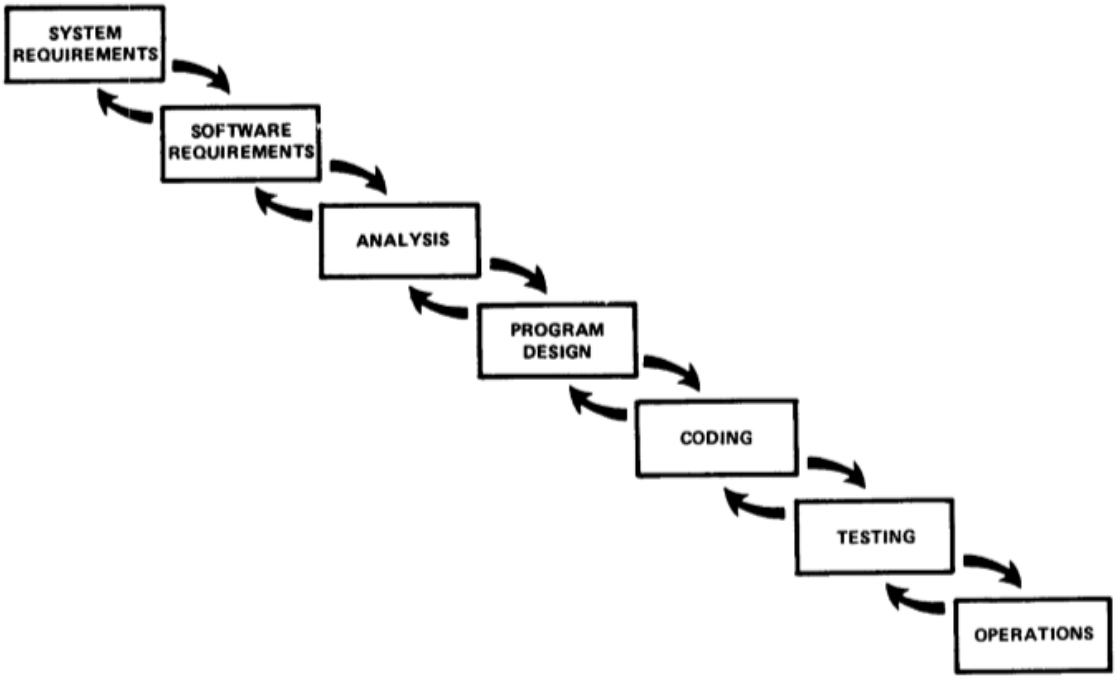
5.4 Σχεδίαση/δημιουργία του φιλικού περιβάλλοντος

Στο κεφάλαιο αυτό ακολουθώντας την διαδικασία που περιγράφετε στο κύκλο ανάπτυξης λογισμικού (systems development cycle) [58] θα καταγράψουμε τις ανάγκες δημιουργίας του περιβάλλοντος χρήσης, το πλάνο ανάπτυξης του, θα αναλύσουμε και θα σχεδιάσουμε την δομή του και τέλος θα το υλοποιήσουμε χρησιμοποιώντας τα εργαλεία που αναφέρθηκαν παραπάνω. Η διαδικασία αυτή ονομάζεται κύκλος ανάπτυξης λογισμικού και περιλαμβάνει αρκετά βήματα μέχρι την παράδοση του συστήματος. Στην παρακάτω εικόνα βλέπουμε ένα ολοκληρωμένο κύκλο ανάπτυξης λογισμικού.

Εικόνα 5-5 – SDLC [59]

5.4.1 Το μοντέλο του καταρράκτη [60]

Για την ανάπτυξη του λογισμικού θα χρησιμοποιήσουμε το μοντέλο του καταρράκτη το όποιο είναι ένα γραμμικό μοντέλο κύκλου ζωής που αποτελείται από συγκεκριμένες διαδικασίες. Μετά το τέλος της κάθε διαδικασίας δημιουργείται ένα προϊόν-κομμάτι του έργου, έτσι διαχωρίζεται σε φάσεις και είναι ευκολότερη η διαχείριση του. Το λογισμικό ολοκληρώνεται όταν περατωθούν όλες οι διαδικασίες με κατάληξη την φάση τις επικύρωσης (testing) μετά την οποία παραδίδεται στην παράγωγη. Οι διαδικασίες που αποτελούν το μοντέλο παρουσιάζονται και αναλύονται παρακάτω:



Εικόνα 5-6 - waterfall model [60]

Ανάλυση απαιτήσεων: κατά την διαδικασία αυτή καταγράφονται οι απαιτήσεις του έργου. Μετά το πέρας της φάσης καταλήγουμε στις προδιαγραφές/απαιτήσεις του λογισμικού και την μοντελοποίηση του έργου.

Σχεδιασμός: κατά την φάση αυτή γίνεται η σχεδίαση του λογισμικού, καταγράφονται τα υλικά, τα δεδομένα, τα επιμέρους προγράμματα, η αρχική εικόνα του περιβάλλοντος χρήσης, η διεπαφή με το χρήστη και τέλος το σχέδιο υλοποίησης του έργου.

Υλοποίηση - ανάπτυξη: μετά το τέλος της ανάλυσης και του σχεδιασμού, τα παραδοτέα που παράχθηκαν δίνονται στους προγραμματιστές για την υλοποίηση του λογισμικού. Η φάση αυτή συνήθως είναι η ποιο χρονοβόρα διότι αποτελείται από επιμέρους φάσεις υλοποίησης, εγκατάστασης, ελέγχου και πλάνο αξιολόγησης.

Έλεγχος - επικύρωση: μετά την φάση της υλοποίησης το λογισμικό παραδίδεται στους χρήστες για έλεγχο και επιβεβαίωση κάλυψης προδιαγραφών. Στην διαδικασία αυτή δοκιμάζονται (testing framework) όλες οι λειτουργίες και η διεπαφή με το χρήστη. Στο τέλος καταγράφονται τα προβλήματα και οι αστοχίες και παραδίδονται ξανά στους προγραμματιστές για την διόρθωση τους.

Λειτουργία – συντήρηση: μετά τον επιτυχή έλεγχο το λογισμικό δίδεται σε παραγωγική λειτουργία και εγκαθίσταται στα συστήματα του οργανισμού. Τέλος καταστρώνεται ένα πλάνο συντήρησης του

λογισμικού έτσι ώστε να υπάρχει επικοινωνία με την ομάδα ελέγχου-ανάπτυξης για μελλοντικές αναβαθμίσεις και διορθώσεις όποτε απαιτηθούν.

Στα επόμενα κεφάλαια θα ακολουθήσουμε τα σταδία ανάπτυξης λογισμικού σύμφωνα με το μοντέλο καταρράκτη για την υλοποίηση του φιλικού περιβάλλοντος διείσδυσης.

5.4.2 Ανάλυση – Πλάνο δημιουργίας λογισμικού

Στα παραπάνω κεφάλαια αναπτύξαμε τις ανάγκες δημιουργίας μιας εφαρμογής μέσω της οποίας ο χρήστης θα εκτελεί δοκιμές διείσδυσης ξεκινώντας τον Η/Υ του με ένα live Linux cdrom. Για την κατασκευή του έργου πρέπει να αναλυθούν οι πτυχές που θα καλυφτούν με το λογισμικό. Στα θεωρητικό μέρος της διατριβής αναφέρθηκαν αρκετά προγράμματα που εκτελούν δοκιμές διείσδυσης και παρέχουν φιλικό περιβάλλον. Τα προγράμματα αυτά είτε έκαναν πολλά διαφορετικά πράγματα μπερδεύοντας τον χρήστη, είτε ήταν δύσχρηστα, είτε ήταν επί πληρωμής με αποτέλεσμα να μην είναι διαθέσιμα για όλους. Αναλύοντας τα παραπάνω μπορούμε να καταγράψουμε τις προδιαγραφές του προγράμματος:

Απλό και φιλικό στο χρήστη: θα πρέπει να εμφανίζει με απλό τρόπο δοκιμές σάρωσης και διείσδυσης.

Λίγες και στοχευόμενες επιλογές: θα πρέπει να έχει κατανοητές επιλογές με κύριο σκοπό την σάρωση ενός στόχου και στην συνέχεια την προσπάθεια διείσδυσης.

Εύκολη εγκατάσταση και εκτέλεση: το πρόγραμμα πρέπει να εκτελείτε σε οποιοδήποτε υπολογιστή χωρίς να χρειάζεται επικοινωνία με τον χρήστη.

Όσο το δυνατόν λιγότερη εισαγωγή πληροφορίας: ο χρήστης θα εισάγει μόνο το σύστημα-στόχο. Όλες οι άλλες επιλογές θα είναι με drop-down menu.

Ικανότητα εκτέλεσης χωρίς την χρήση γραφικών: ο ποιο έμπειρος χρήστης πρέπει να μπορεί να εκτελέσει τις δοκιμές από την γραμμή εντολών.

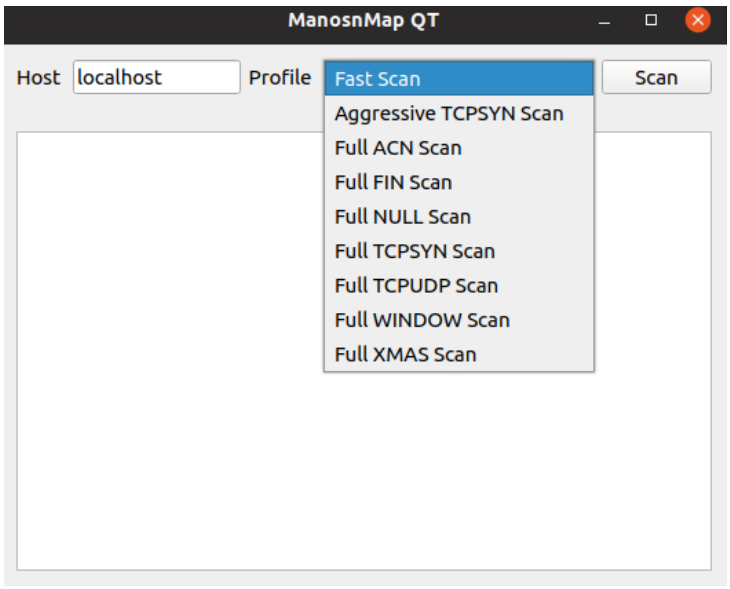
Ικανότητα αποστολής αποτελεσμάτων: εφόσον πρόκειται για live cdrom δεν υπάρχει χώρος αποθήκευσης και μετά την επανεκκίνηση τα δεδομένα χάνονται, άρα πρέπει να παρέχεται η επιλογή αποστολής των αποτελεσμάτων με email.

Μετά την καταγραφή των αναγκών του λογισμικού οργανώνουμε στο πλάνο δημιουργίας του. Η ροη κατασκευής θα περιέχει δυο διαφορετικά σκέλη, την σχεδίαση του γραφικού περιβάλλοντος και την εισαγωγή λειτουργικότητας σε αυτό. Στην συνέχεια θα εκτελέσουμε δοκιμές του λογισμικού και θα διορθώσουμε τα πιθανά προβλήματα που θα εμφανιστούν. Η τελευταία έκδοση του προγράμματος θα πρέπει να καλύπτει τους στόχους της ανάλυσης.

5.4.3 Σχεδίαση

Για την σχεδίαση του γραφικού περιβάλλοντος (GUI) θα λάβουμε υπόψη τις προδιαγραφές που τέθηκαν στο παραπάνω κεφάλαιο. Το GUI θα πρέπει να έχει όσο το δυνατόν λιγότερες επιλογές έτσι ώστε ο χρήστης καταβάλλοντας την λιγότερη προσπάθεια να εκτελέσει μια δοκιμή διείσδυσης. Για την σχεδίαση του γραφικού περιβάλλοντος χρησιμοποιήσαμε το qt designer μέσω του οποίου σχεδιάσαμε το main window της εφαρμογής, τα buttons και τα dropdown menu. Μετά την σχεδίαση αποθηκεύσαμε τα γραφικά μας σε αρχείο ui και μέσω του εργαλείου pyuic5 τα μετατρέψαμε σε python language (pyuic5 xyz.ui -o xyz.py).

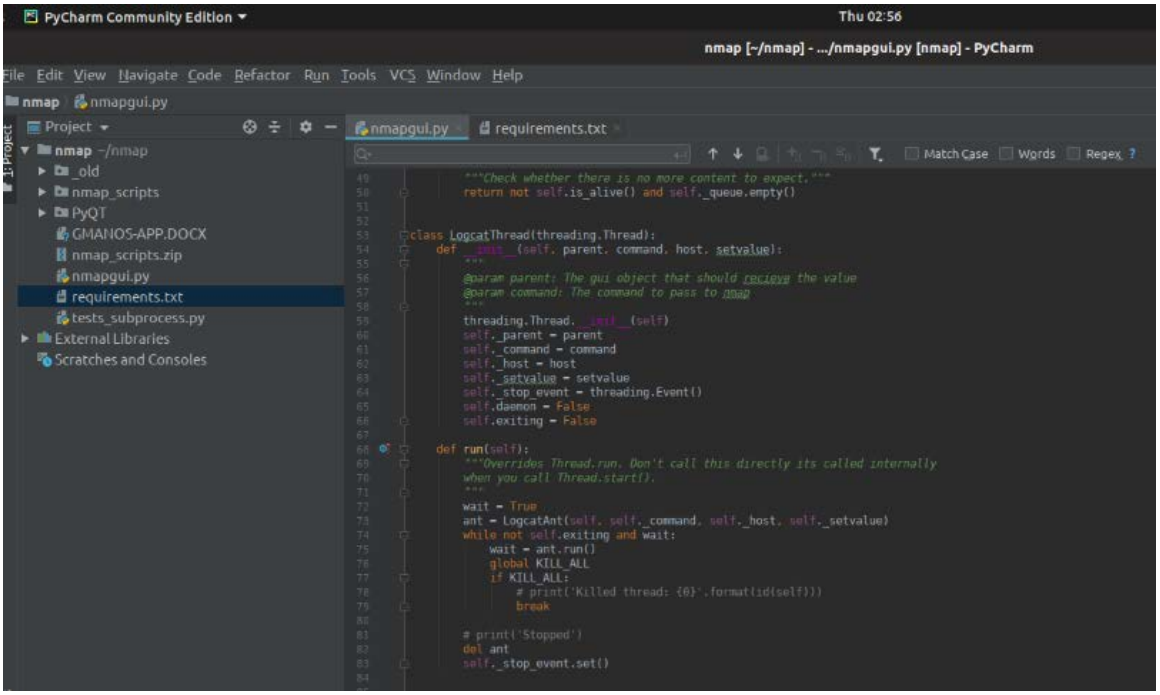
Το τελικό αποτέλεσμα της σχεδίασης φαίνεται στην παρακάτω εικόνα με τον χρήστη να εισάγει το σύστημα-στόχος και να επιλεγεί την τεχνική σάρωσης (profile) και τέλος την έναρξη της δοκιμής. Συνεπώς καλύπτουμε τις δυο πρώτες προδιαγραφές και την λιγότερη εισαγωγή πληροφορίας από το χρήστη.



Εικόνα 5-7 - Σχεδίαση GUI

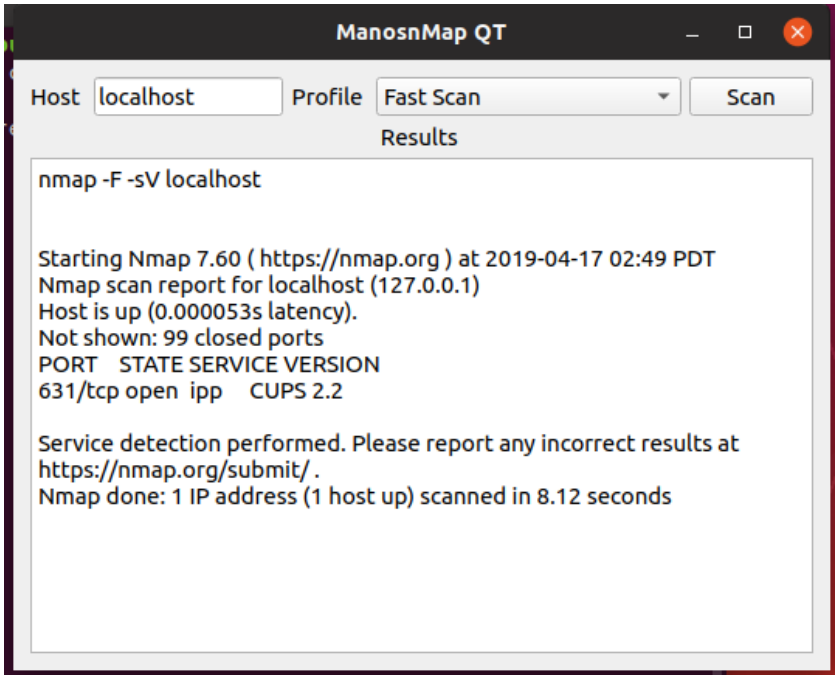
5.4.4 Ανάπτυξη εφαρμογής

Σύμφωνα με το μοντέλο καταρράκτη μετά την σχεδίαση της εφαρμογής προχώρημα στη συγγραφή κώδικα (Παράρτημα 1) έτσι ώστε να εισάγουμε λειτουργικότητα στο γραφικό περιβάλλον. Για την συγγραφή του κώδικα σε pyqt5 χρησιμοποιήθηκε το IDE PyCharm community edition το οποίο είναι ένα περιβάλλον ανάπτυξης κώδικα με ικανότητες όπως preview, αυτόματη διόρθωση, εκτέλεση βήμα - βήμα έτσι ώστε να μπορεί να γίνει αποσφαλμάτωση, κτλ. Στη παρακάτω εικόνα βλέπουμε το IDE στην πράξη.



Εικόνα 5-8 – PyCharm

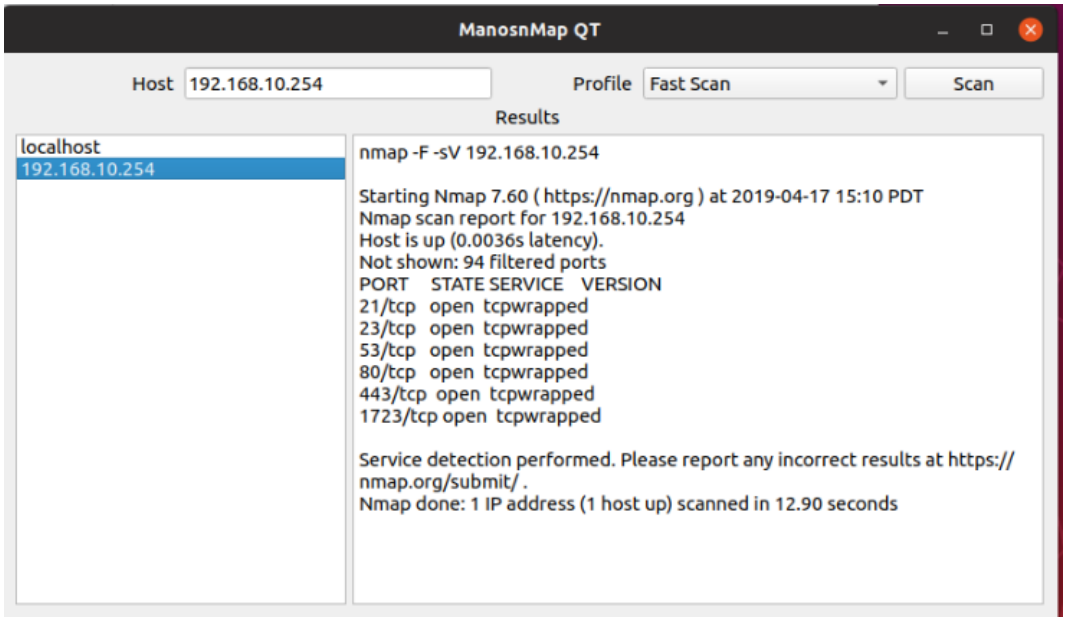
Στο κεφάλαιο 5.2 αναπτύξαμε τα scripts που θα χρησιμοποιήσουμε μέσα στο GUI για την εκτέλεση των δοκιμών διείσδυσης. Συνεπώς δημιουργήθηκαν τα scan profiles και αντιστοιχήθηκαν σε εντολές nmap, στην συνέχεια δημιουργήσαμε python threads έτσι ώστε να μπορούμε να ελέγχουμε τις σαρώσεις και τα αποτελέσματα τους μέσα από το GUI. Η εκτέλεση της 1ης έκδοσης φαίνεται παρακάτω:



Εικόνα 5-9 – nmapgui ver 1

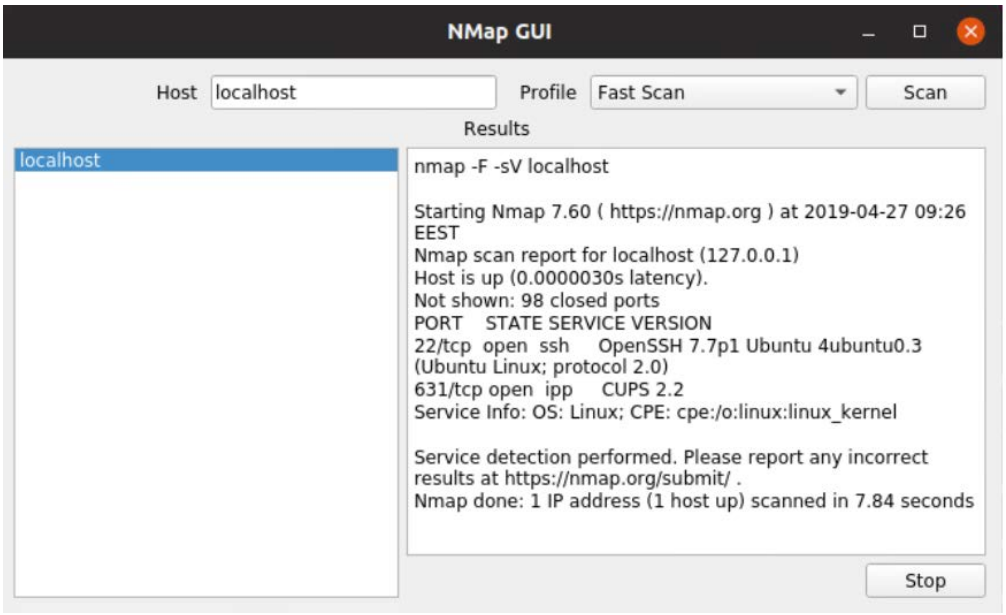
Ακολουθώντας το μοντέλο ανάπτυξης λογισμικού εκτελέσαμε αρκετές φορές το λογισμικό μας έτσι ώστε να διορθώσουμε πιθανά σφάλματα στο κώδικα και στην ροή του προγράμματος. Στο δεύτερο επίπεδο ανάπτυξης , στα πλαίσια των προδιαγραφών θα πρέπει να προστεθεί ιστορικό εκτελέσεων έτσι ώστε να μπορούν να αποσταλούν στο χρήστη.

Στην παρακάτω εικόνα βλέπουμε την έκδοση 2 του GUI με την εισαγωγή ιστορικού:



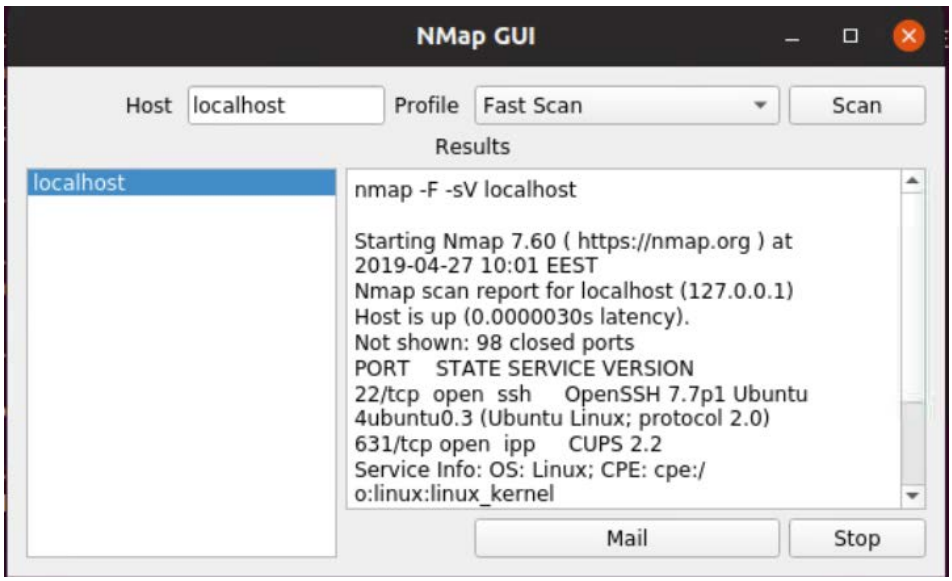
Εικόνα 5-10 - nmapgui ver 2

Μετά από αρκετές εκτελέσεις διαπιστώσαμε ότι κάποιες δοκιμές διείσδυσης δεν παράγουν αποτελέσματα και δεν τελειώνουν πότε, είτε λόγο διακοπής από το τοίχος προστασίας είτε ανίχνευσης από το IDS. Συνεπώς πρέπει να εισάγουμε μια διεργασία για την ακύρωση μιας δοκιμής. Σύμφωνα με τον κύκλο ανάπτυξης ανανεώνουμε τις προδιαγραφές μας προσθέτοντας την εισαγωγή διεπαφής (button) για την ακύρωση μιας δοκιμής που εκτελείται. Στον 3ο κύκλο ανάπτυξης εισήγαμε το button ακύρωσης όπως φαίνεται στην παρακάτω εικόνα:



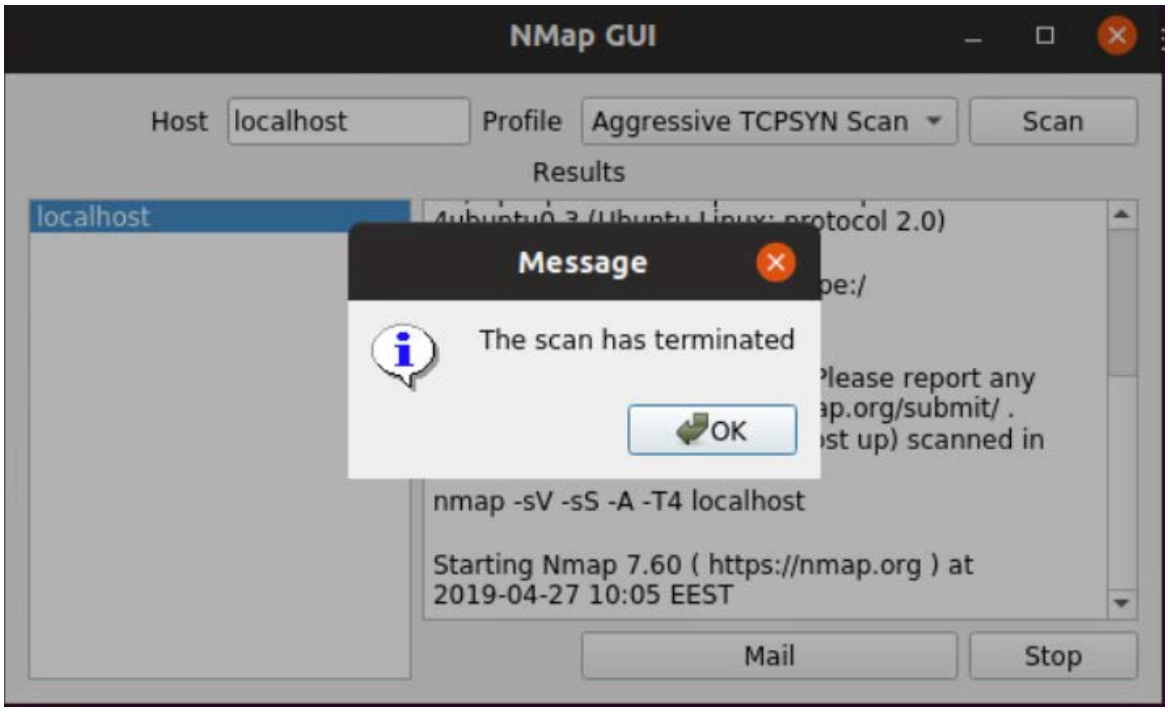
Εικόνα 5-11 - nmapgui ver 3

Στον 4ο βρόγχο θα προστεθεί η διεπαφή αποστολής αποτελεσμάτων στο χρήστη όπως ορίζεται στις προδιαγραφές. Για την υλοποίηση της συγκεκριμένης ανάγκης θα πρέπει να χρησιμοποιηθεί ένας free mail διακομιστής έτσι ώστε η εφαρμογή μας να συνδέεται με κάποια credentials και να αποστέλλει στο χρήστη τα αποτελέσματα. Για την ανάπτυξη της παραπάνω απαίτησης χρησιμοποιήσαμε το Python module smtplib μέσω του οποίου ορίστηκε ο smtp διακομιστής, η θύρα σύνδεσης και τα στοιχεία του χρήστη. Στη παρακάτω εικόνα βλέπουμε την υλοποίηση της τελευταίας προδιαγραφής:

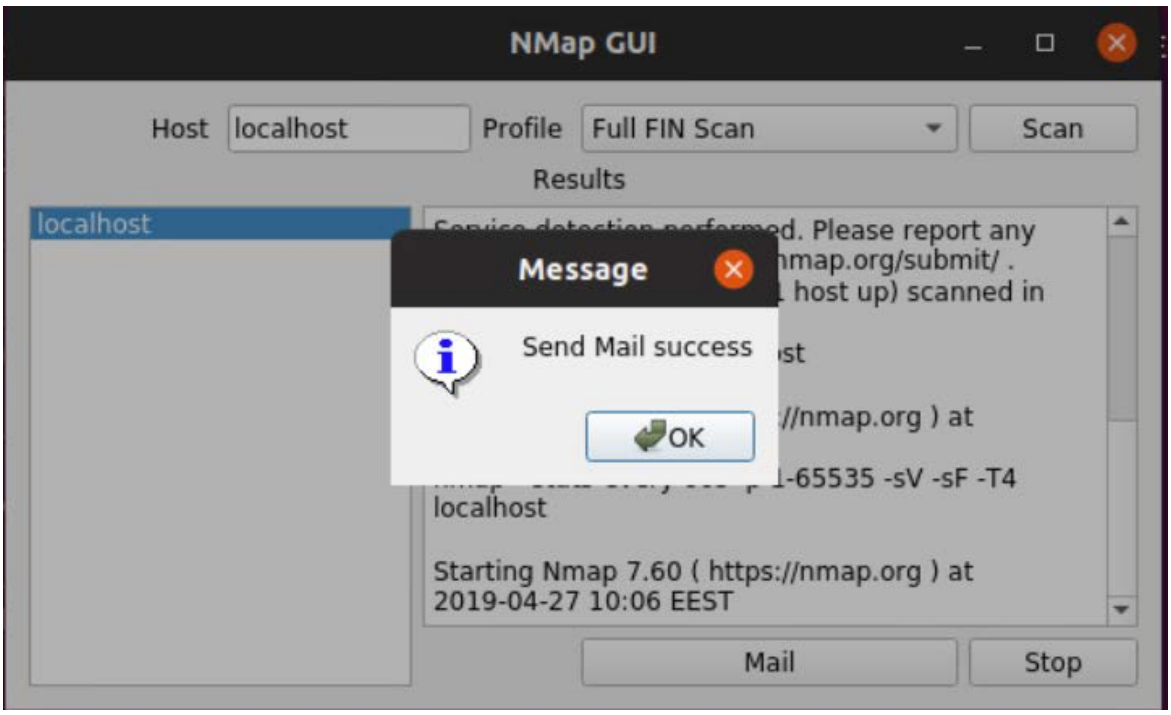


Εικόνα 5-12 - nmapgui ver 4

Στον επόμενο κύκλο ανάπτυξης εισήγαμε μηνύματα προς το χρήστη έτσι ώστε να ενημερώνετε για το πέρας των επίλογων. Παρακάτω βλέπουμε δυο εικόνες με τα μηνύματα διακοπής μιας σάρωσης και αποστολής αποτελεσμάτων στο χρήστη:



Εικόνα 5-13 - Stop message



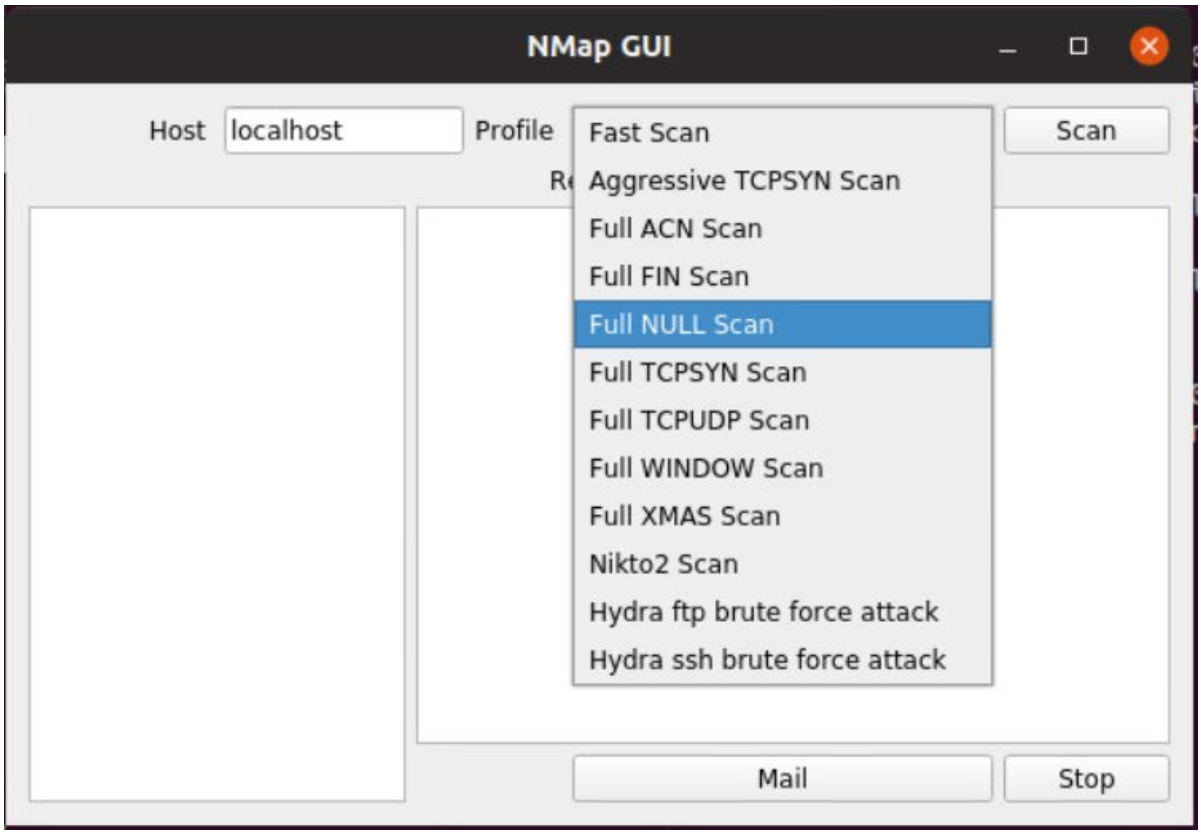
Εικόνα 5-14 - Send mail message

Στο 5ο κύκλο ανάπτυξης προστεθήκαν 3 τεχνικές διείσδυσης που θα μπορούν να εκτελεστούν μετά την σάρωση και την ανίχνευση ανοικτών θυρών από το nmap. Η επιλογή των προγραμμάτων διείσδυσης έγινε με κριτήριο δημοτικότητας και ελέγχου των πιο κοινών υπηρεσιών που παρέχονται σήμερα από διακομιστές. Το 1ο πρόγραμμα είναι το nikto [48] το οποίο ελέγχει διακομιστές web για

ευπάθειες και λάθη εγκατάστασης. Έτσι ο χρήστης θα εκτελεί πρώτα την σάρωση μέσω του nmap και στην περίπτωση που το σύστημα στόχος παρέχει υπηρεσίες web θα μπορεί να προχωρήσει στον έλεγχο του μέσω του nikto.

Το 2ο πρόγραμμα είναι το THC Hydra [10] το οποίο εκτελεί brute force attack σε διακομιστές που παρέχουν υπηρεσίες απομακρυσμένης σύνδεσης όπως είναι το ftp και το ssh. Η Hydra χρησιμοποίει ένα λεξικό με κοινούς κωδικούς σύνδεσης εκτελώντας προσπάθειες διείσδυσης με διαφορετικό όνομα χρήστη και κωδικό.

Εισάγοντας τις δυο τεχνικές διείσδυσης ολοκληρώσαμε τον κύκλο ανάπτυξης και το GUI έχει την παρακάτω τελική μορφή:



Εικόνα 5-15 - nmapgui Final Release

Μετά το πέρας της ανάπτυξης προχωρήσαμε στο στάδιο του deployment/installation όπου δημιουργήσαμε το περιβάλλον εκτέλεσης της εφαρμογής εγκαθιστώντας τα προγράμματα που χρειαζόμαστε, τις βιβλιοθήκες της python για την σωστή εκτέλεση του GUI και τέλος την εγκατάσταση του cubic για την κατασκευή του disk image όπως περιγράφεται στο επόμενο κεφάλαιο.

5.4.5 Δημιουργία bootable Linux cdrom με το πρόγραμμα cubic

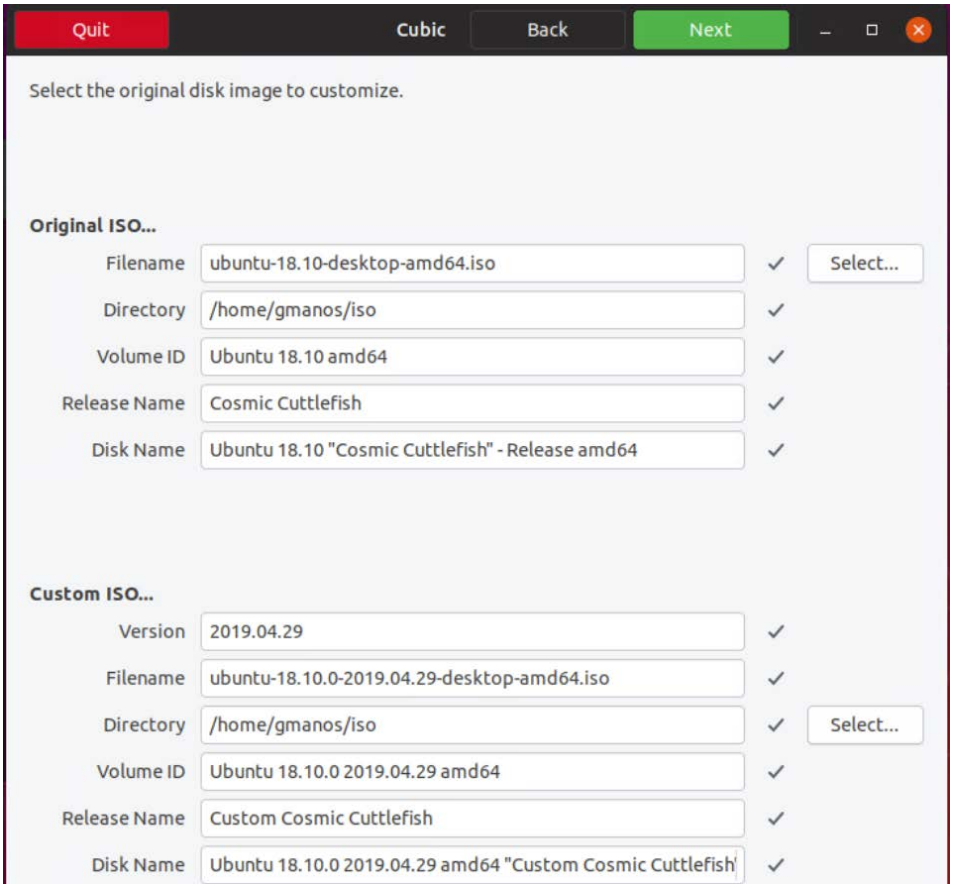
Αρχικά εγκαταστήσαμε το πρόγραμμα cubic ακολουθώντας τις οδηγίες [54], στην συνέχεια το εκτελέσαμε με μορφή wizard. Όπως αναφέραμε σε προηγούμενο κεφάλαιο θα χρησιμοποιήσουμε το Ubuntu 18.10 σαν βάση του cdrom μας. Πάνω σε αυτό θα εγκαταστήσουμε το nmap, το nikto, το LHC hydra, τα python modules και την τελευταία έκδοση της python έτσι ώστε να λειτουργεί σωστά το NMap GUI.

Στο κεφάλαιο αυτό παρουσιάζεται η διαδικασία δημιουργίας του τελικού disk image.

Αρχικά όπως βλέπουμε στη παρακάτω εικόνα το πρόγραμμα ζητά ένα κατάλογο μέσα στον οποίο θα αποθηκεύσει το αρχείο.

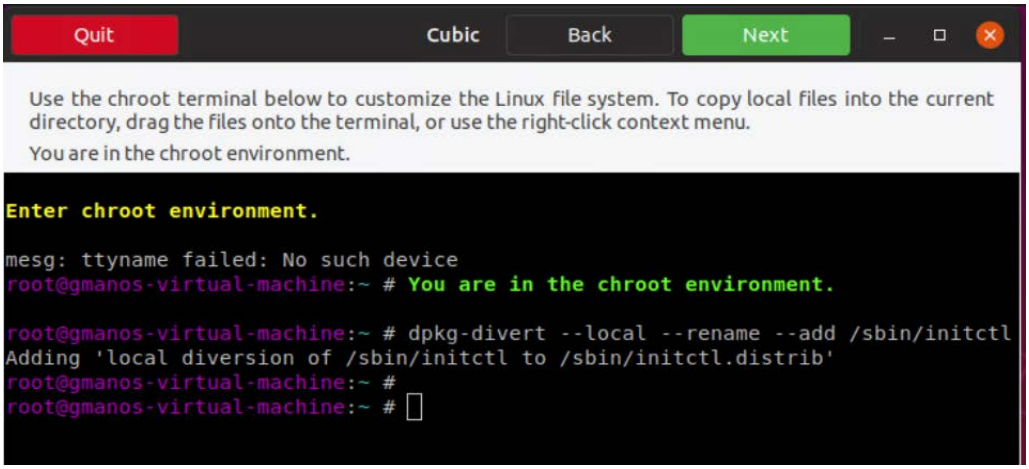
Εικόνα 5-16 - cubic iso creator

Στο επόμενο βήμα εισάγουμε το original iso και τις περιγραφές του custom iso:

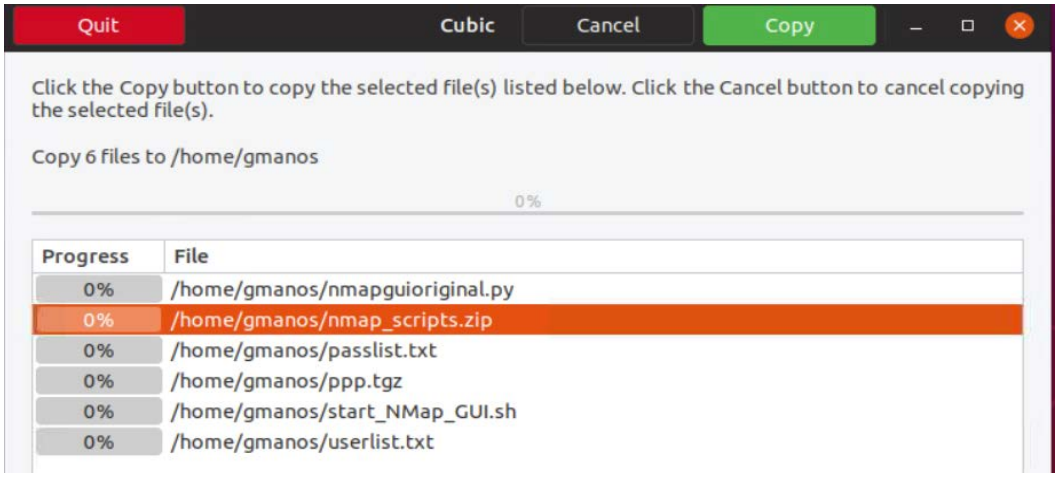


Εικόνα 5-17 - cubic os selection

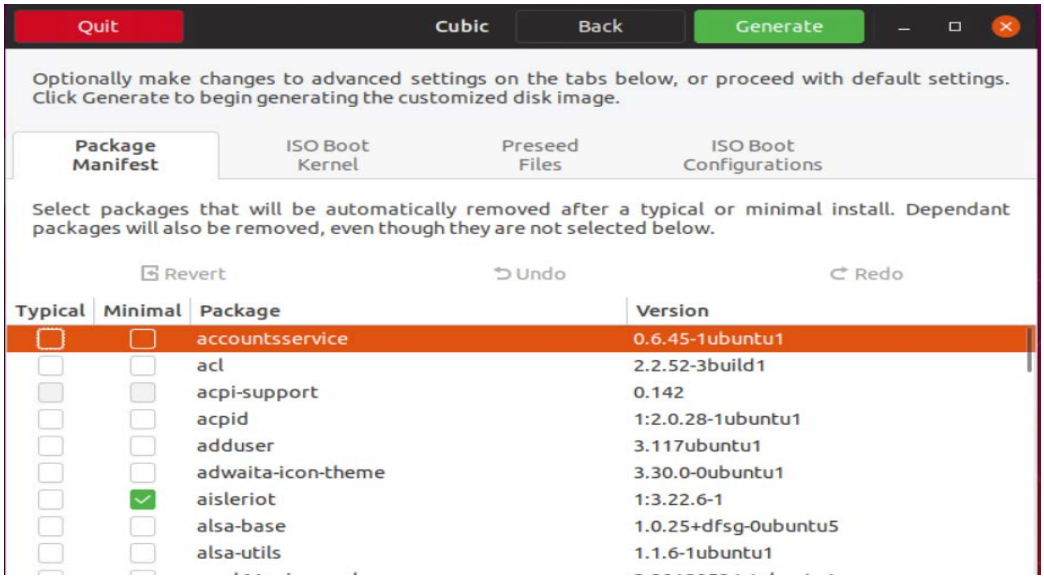
Μετά την αποσυμπίεση του iso αρχείου, το cubic μας εισάγει σε ένα chrooted περιβάλλον έτσι ώστε να παραμετροποιήσουμε το Ubuntu και να προσθέσουμε τα δικά μας αρχεία (εικόνες 5-17, 5-18).



Εικόνα 5-18- cubic chroot



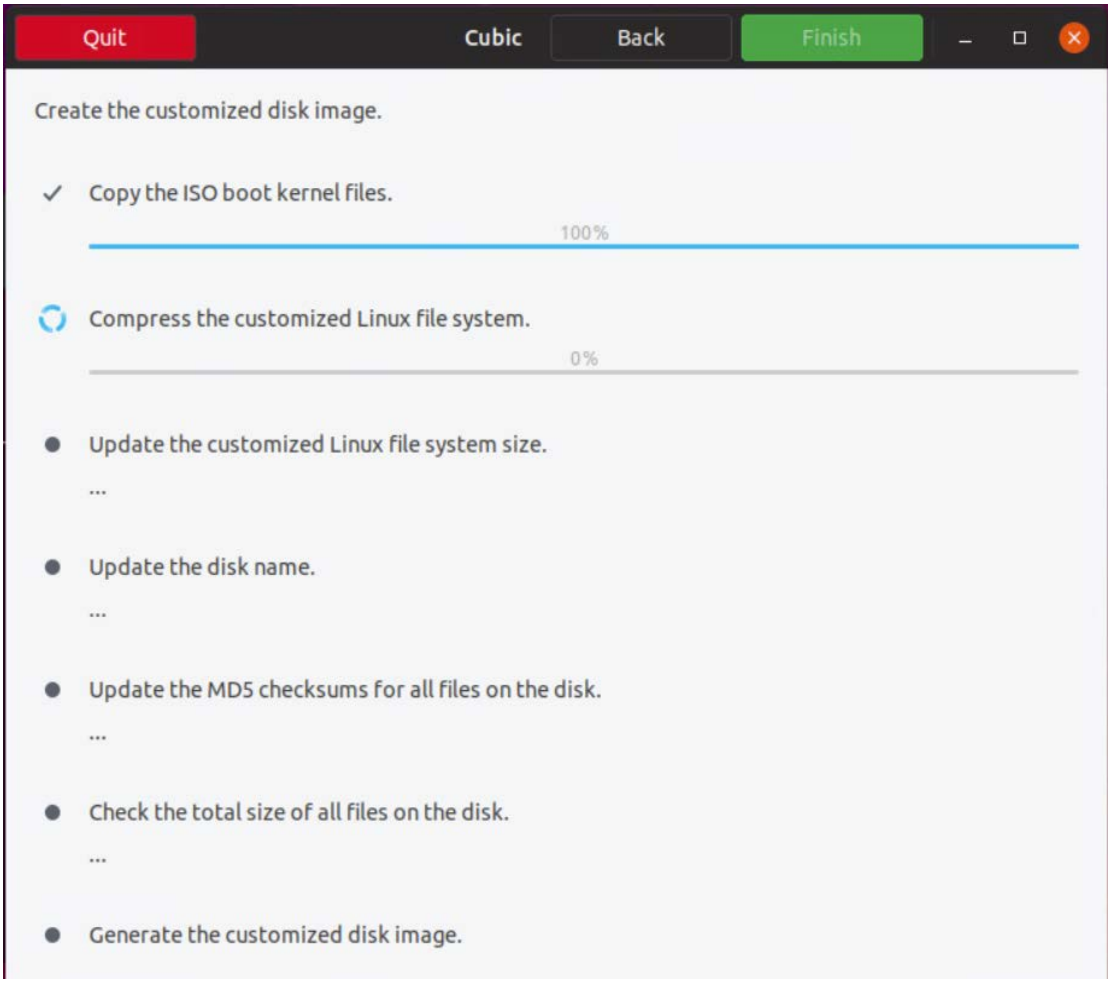
Εικόνα 5-19 - cubic transfer file



Εικόνα 5-20 - cubic software installation

Τέλος επιλεγούμε τα πακέτα που θα εγκατασταθούν σε περίπτωση επιλογής εγκατάστασης.

Μετά το πέρας της παραμετροποίησης το cubic δημιουργεί το iso disk image.



Εικόνα 5-21 - generate disk image

5.5 Παρουσίαση και Εκτέλεση του GUI

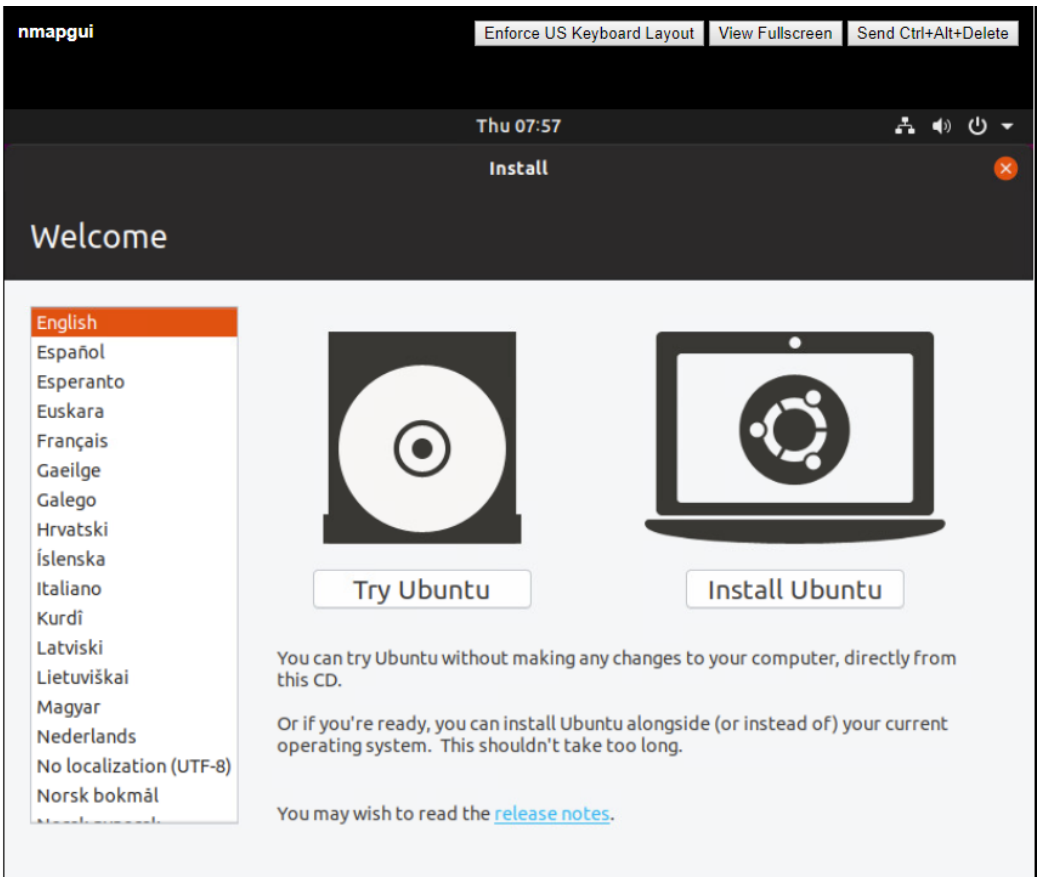
Στο κεφάλαιο αυτό γίνεται η παρουσίαση της τελικής έκδοσης του bootable Linux cdrom, ξεκινώντας από την εκκίνηση του σε μια εικονική μηχανή, την σύνδεση του στο δίκτυο του οργανισμού, την εκτέλεση του GUI και τέλος την προσπάθεια σάρωσης και διείσδυσης στα πληροφοριακά συστήματα του οργανισμού.

Η εκτέλεση των δοκιμών διείσδυσης θα γίνει από δυο διαφορετικά δίκτυα οργανισμού έτσι ώστε να ελέγξουμε την ανίχνευση των δοκιμών από το τοίχος προστασίας και το IDS που εκτελείται στον εξωτερικό δρομολογητή του δικτύου.

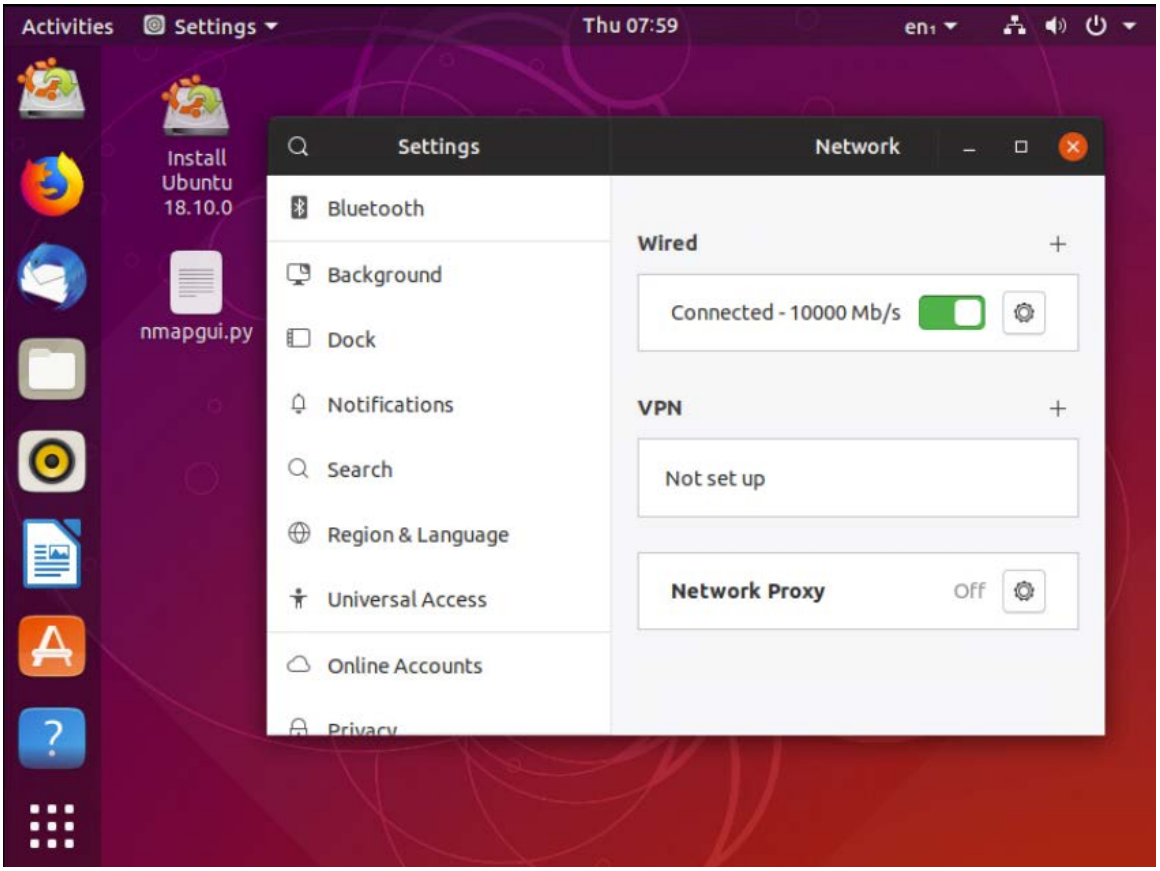
5.5.1 Εκκίνηση Linux cdrom

Για την δοκιμή του Linux cdrom δημιουργήθηκε μια εικονική μηχανή σε περιβάλλον εικονοποίησης VMWare με 2GB RAM και 16GB HDD. Η μηχανή είναι συνδεδεμένη σε ένα από τα εσωτερικά δίκτυα του οργανισμού στο οποίο υπάρχει διακομιστής δυναμικών διευθύνσεων.

Συνεπώς προσαρτήσαμε το ISO στην εικονική μηχανή και την ξεκινήσαμε μέσα από το GUI του VMWare. Στις παρακάτω εικόνες διακρίνονται οι επιλογές που καλείτε να εισάγει ο χρήστης μέχρι την έναρξη του γραφικού περιβάλλοντος χρήσης (window manager) και την εκτέλεση του NMap GUI με κύριο σκοπό την εκκίνηση των δοκιμών διείσδυσης.



Εικόνα 5-22 - Επιλογή δοκιμής Ubuntu

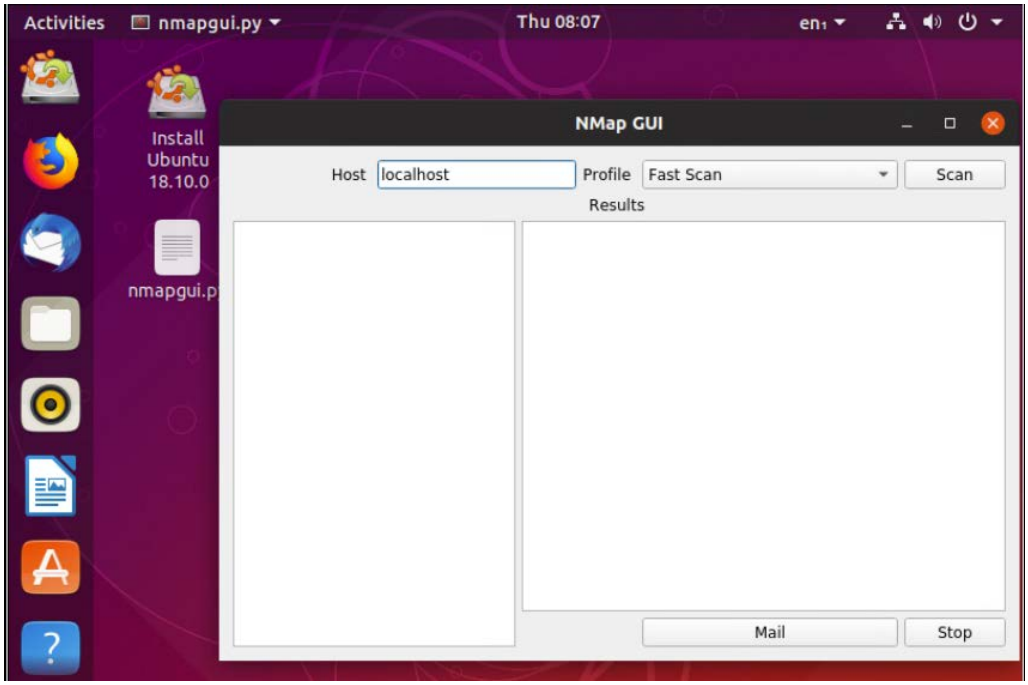


Εικόνα 5-23 - Network setup

Το Linux ξεκίνησε σωστά, αναγνώρισε το δίκτυο και συνδέθηκε σε αυτό, ενώ εμφανίζεται η εφαρμογή μας στο desktop του χρήστη. Επόμενο βήμα η εκτέλεση του nmap gui.

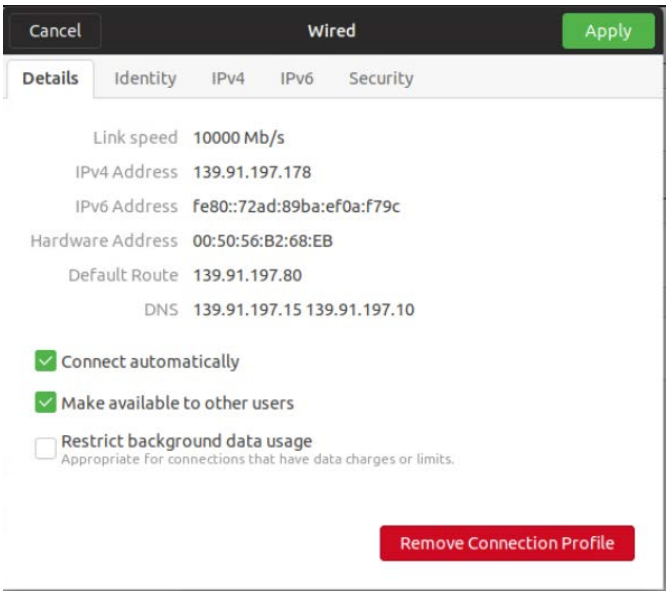
5.5.2 NMap GUI

Για την εκτέλεση του nmap gui έχει επιλεγεί μέσω της εντολής sudo να εκτελείται σαν administrator, έτσι ώστε να έχει πρόσβαση στις τεχνικές διείσδυσης που απαιτούν full privileges.



Εικόνα 5-24 - NMap GUI

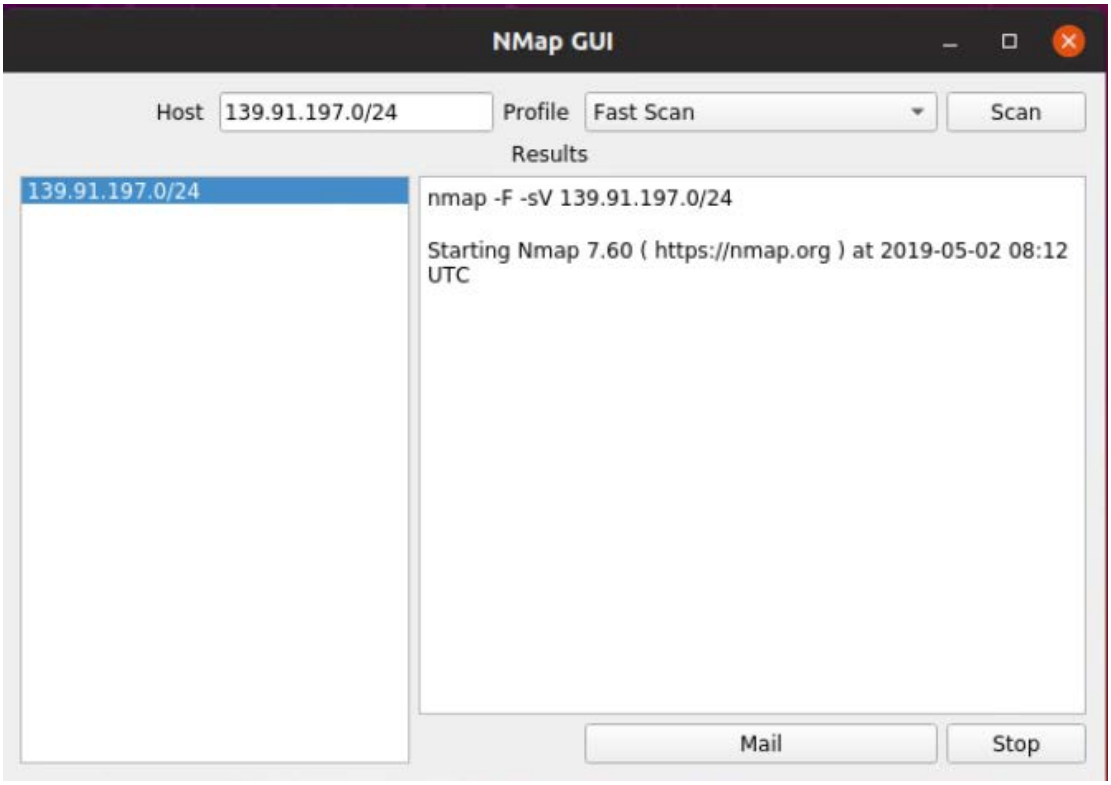
Για την επιλογή του δικτύου σάρωσης θα πρέπει να ανακαλύψουμε το δίκτυο που είμαστε συνδεδεμένοι. Επιλέγοντας network→wired connection→advanced settings, εμφανίζεται η ip address του μηχανήματος μας:



Εικόνα 5-25 - NMap GUI IP Address

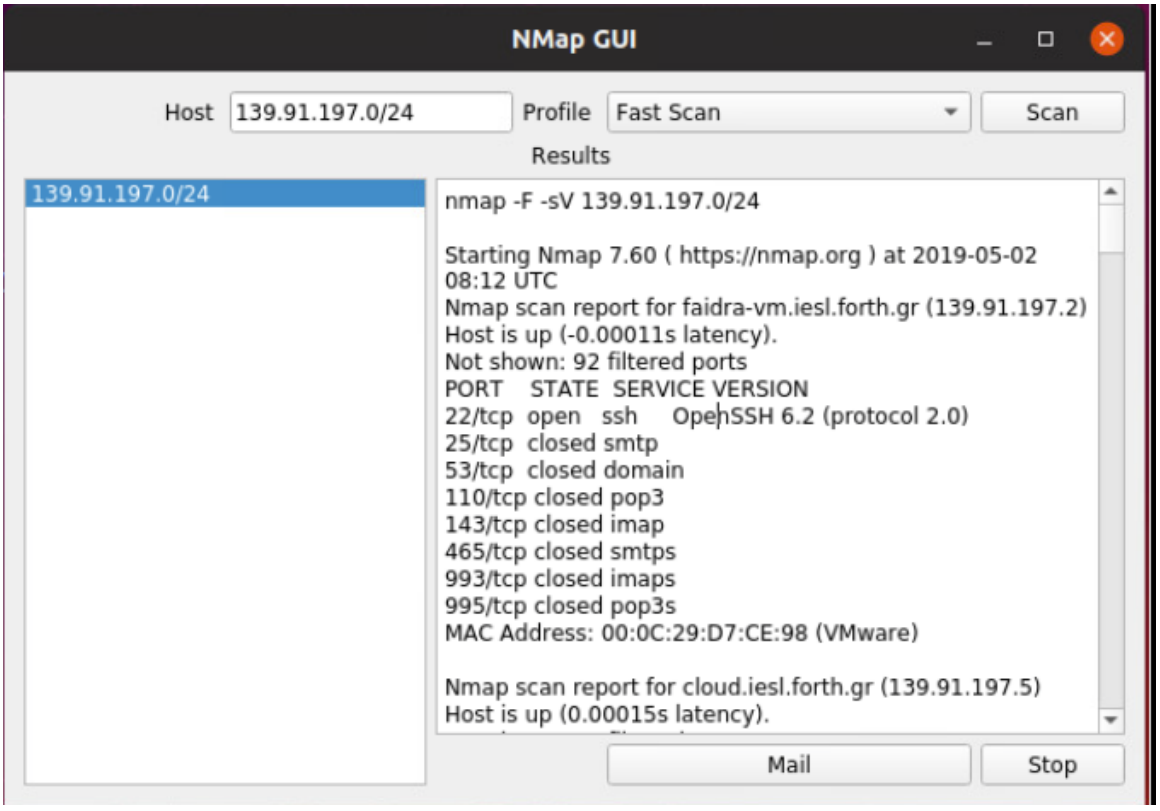
Συνεπώς θα επιλέξουμε το δίκτυο 139.91.197.0 για την πρώτη δοκιμή διείσδυσης, έτσι ώστε να ανιχνεύσουμε τις υπηρεσίες και τα πληροφοριακά συστήματα που ανήκουν στο δίκτυο αυτό.

Εισάγουμε την διεύθυνση του δικτύου και επιλεγούν την 1η τεχνική σάρωσης FAST Scan:



Εικόνα 5-26 - 1η σάρωση – Fast Scan

Μετά από αρκετά λεπτά εκτέλεσης της σάρωσης, το nmap gui εμφάνισε τα αποτελέσματα της δοκιμής:



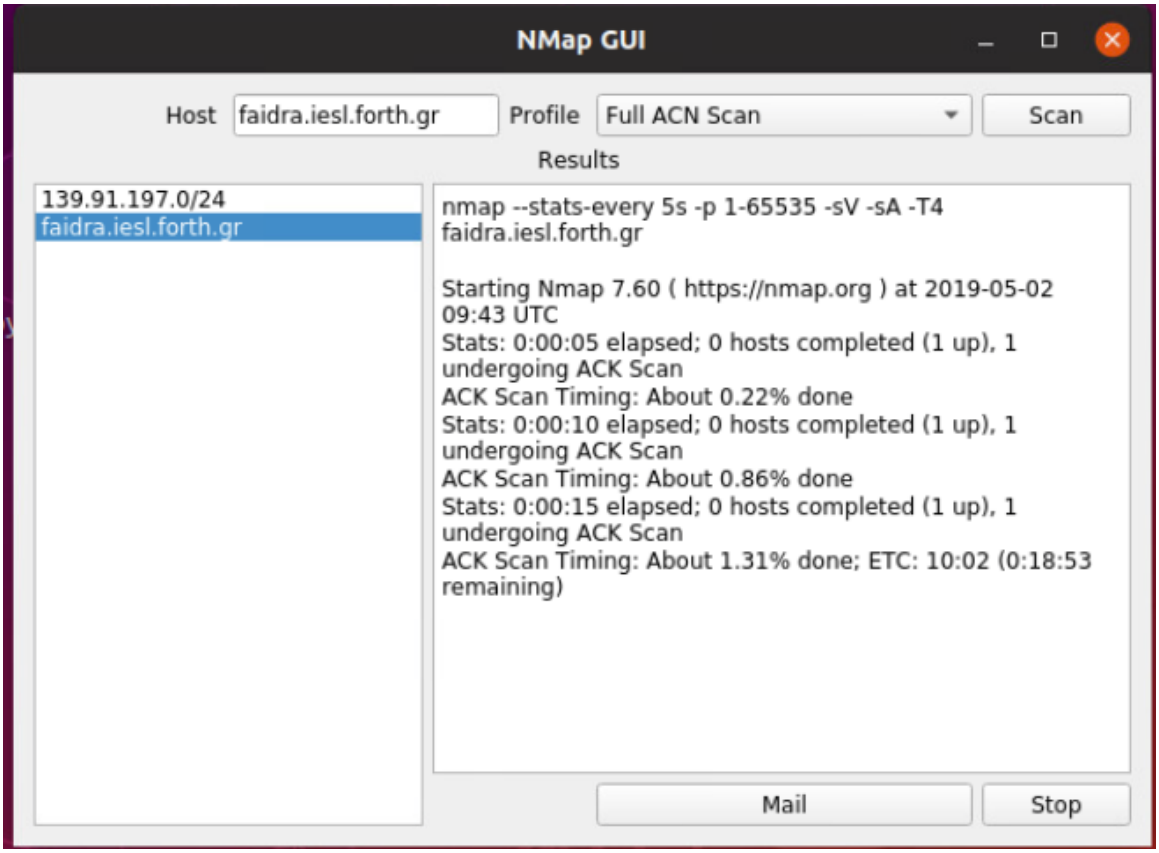
Εικόνα 5-27 - nmap gui run

Παρατηρούμε ότι το NMap GUI σάρωσε όλα τα πληροφοριακά συστήματα του δικτύου και μας εμφάνισε τις θύρες που δέχονται συνδέσεις η είναι closed η filtered. Με την ανάλυση των αποτελεσμάτων μπορούμε να διακρίνουμε τον διακομιστή ηλεκτρονικού ταχυδρομείου, τον διακομιστή ονοματοδοσίας (DNS) καθώς και τον web διακομιστή του δικτύου. Παρακάτω παρουσιάζονται τα τρία αυτά συστήματα στα οποία θα προχωρήσουμε σε περεταίρω δοκιμές σάρωσης:

host	**results**
faidra.iesl.forth.gr	PORT STATE SERVICE VERSION **25/tcp open smtp** Sendmail (Not accepting mail) 53/tcp open domain ISC BIND 9.9.6-P1 80/tcp open http Apache httpd 2.2.12 110/tcp open pop3 Dovecot pop3d 143/tcp open imap Dovecot imapd 443/tcp open ssl/http Apache httpd 2.2.12 ((Linux/SUSE)) 465/tcp open ssl/smtp Sendmail 8.14.3 587/tcp open smtp Sendmail (Not accepting mail) 993/tcp open ssl/imap Dovecot imapd 995/tcp open ssl/pop3 Dovecot pop3d MAC Address: 90:B1:1C:18:C8:09 (Dell) Service Info: Hosts: mail.iesl.forth.gr, iesl.forth.gr; OS: Unix
esperia.iesl.forth.gr	PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 5.3 (protocol 2.0) **53/tcp open domain** ISC BIND 9.8.2rc1 80/tcp open http Apache httpd 2.2.15 ((CentOS)) 443/tcp open ssl/http Apache httpd 2.2.15 ((CentOS)) 3306/tcp open mysql MySQL (unauthorized) MAC Address: 00:0C:29:FB:6C:B5 (VMware) Service Info: OS: Red Hat Enterprise Linux 6; CPE: cpe:/o:redhat:enterprise_linux:6
cc-webserver.iesl.forth.gr	PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 5.3 (protocol 2.0) **80/tcp open http** Apache httpd 2.2.15 443/tcp open ssl/http Apache httpd 2.2.15 ((CentOS)) MAC Address: 00:0C:29:96:E8:AF (VMware) Service Info: Host: www.iesl.forth.gr

Παρατηρούμε επίσης την ανίχνευση των εκδόσεων των προγραμμάτων που εκτελούνται καθώς και το είδος του υλικού βρισκόμενο από το mac address των συστημάτων.

Θα επιλέξουμε τον διακομιστή faidra για την εκτέλεση των επόμενων δοκιμών διείσδυσης με κύριο στόχο τον έλεγχο χωρίς την ανακάλυψη από το IDS του διακομιστή.



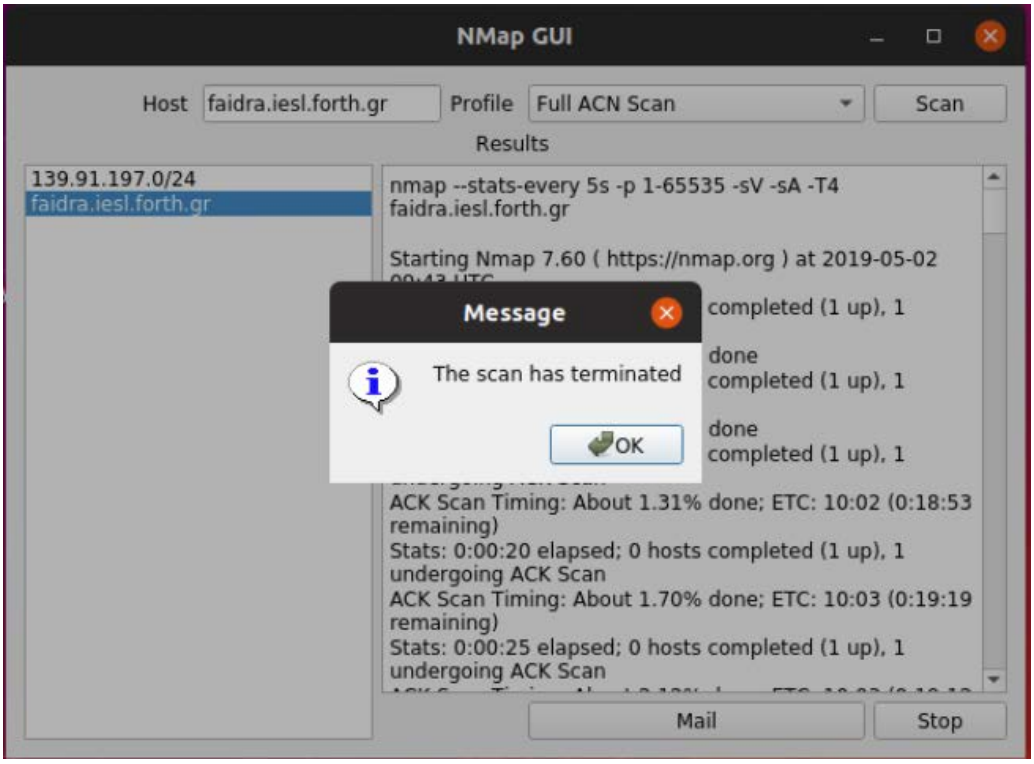
Εικόνα 5-28 - Full ACN Scan

Με την επιλογή τεχνικών full σάρωσης εμφανίζονται progress messages έτσι ώστε να γνωρίζουμε ποσοστιαία το υπόλοιπο του χρόνου μέχρι να ολοκληρωθεί η δοκιμή.

Ελέγχοντας τα αρχεία καταγραφής του διακομιστή παρατηρούμε ότι το τοίχος προστασίας ανίχνευσε την δοκιμή διείσδυση με αποτέλεσμα να μπλοκάρει την σάρωση.

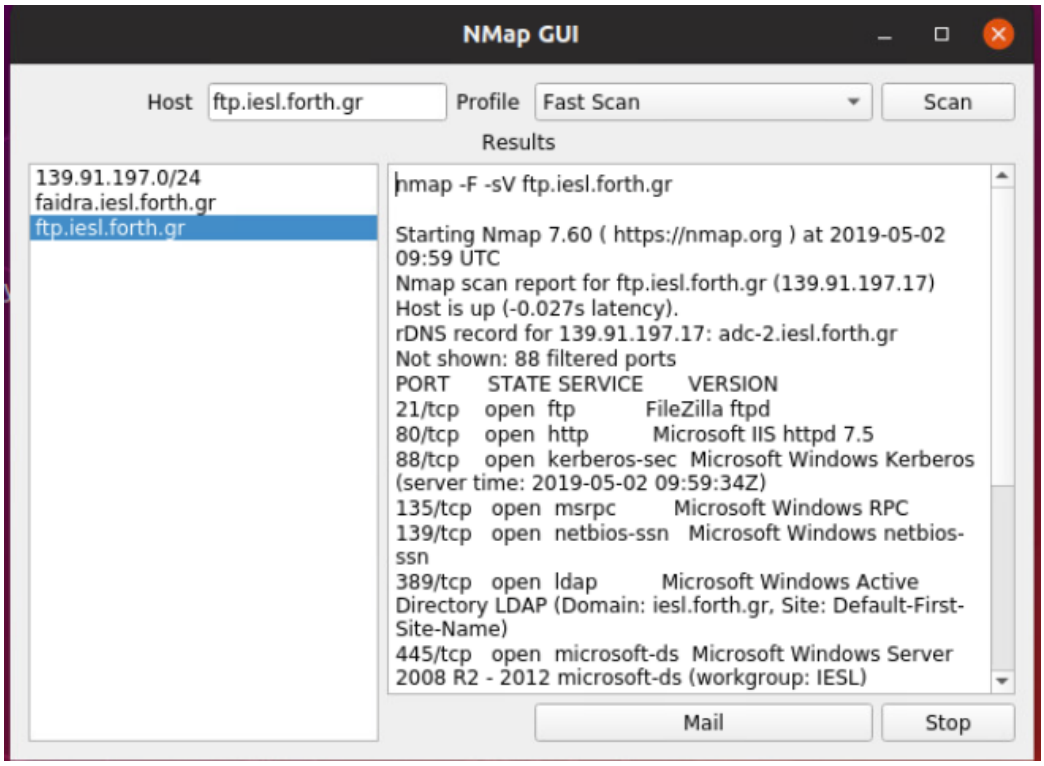
*[44417694.933810] SFW2-INext-**DROP**-DEFLT IN=em1 OUT= MAC=90:b1:1c:18:c8:09:00:50:56:b2:68:eb:08:00 SRC=139.91.197.178 DST=139.91.197.6 LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=11975 PROTO=TCP SPT=41441 DPT=6000 WINDOW=1024 RES=0x00 SYN URGP=0 OPT (020405B4)*
*[44417734.907770] SFW2-INext-**DROP**-DEFLT IN=em1 OUT= MAC=90:b1:1c:18:c8:09:00:50:56:b2:68:eb:08:00 SRC=139.91.197.178 DST=139.91.197.10 LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=62487 PROTO=TCP SPT=41441 DPT=144 WINDOW=1024 RES=0x00 SYN URGP=0 OPT (020405B4)*
*[44417734.907770] SFW2-INext-**DROP**-DEFLT IN=em1 OUT= MAC=90:b1:1c:18:c8:09:00:50:56:b2:68:eb:08:00 SRC=139.91.197.178 DST=139.91.197.10 LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=62487 PROTO=TCP SPT=41441 DPT=144 WINDOW=1024 RES=0x00 SYN URGP=0 OPT (020405B4)*

Συνεπώς θα σταματήσουμε την δοκιμή εφόσον δεν θα εξάγει αποτελέσματα χρησιμοποιώντας το stop button.



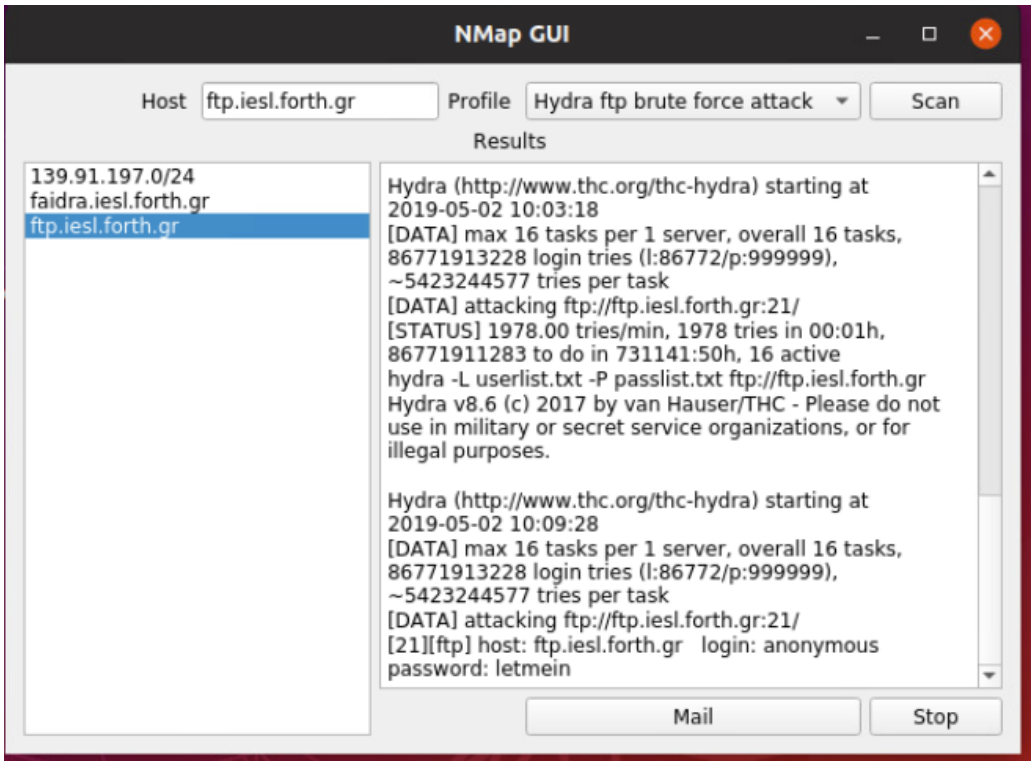
Εικόνα 5-29 - nmapgui stop button

Στην επόμενη δοκιμή θα επιλέξουμε ένα διακομιστή παροχής απομακρυσμένης σύνδεσης, έτσι ώστε να ελέγξουμε την ικανότητα διείσδυσης μέσω brute force attack:



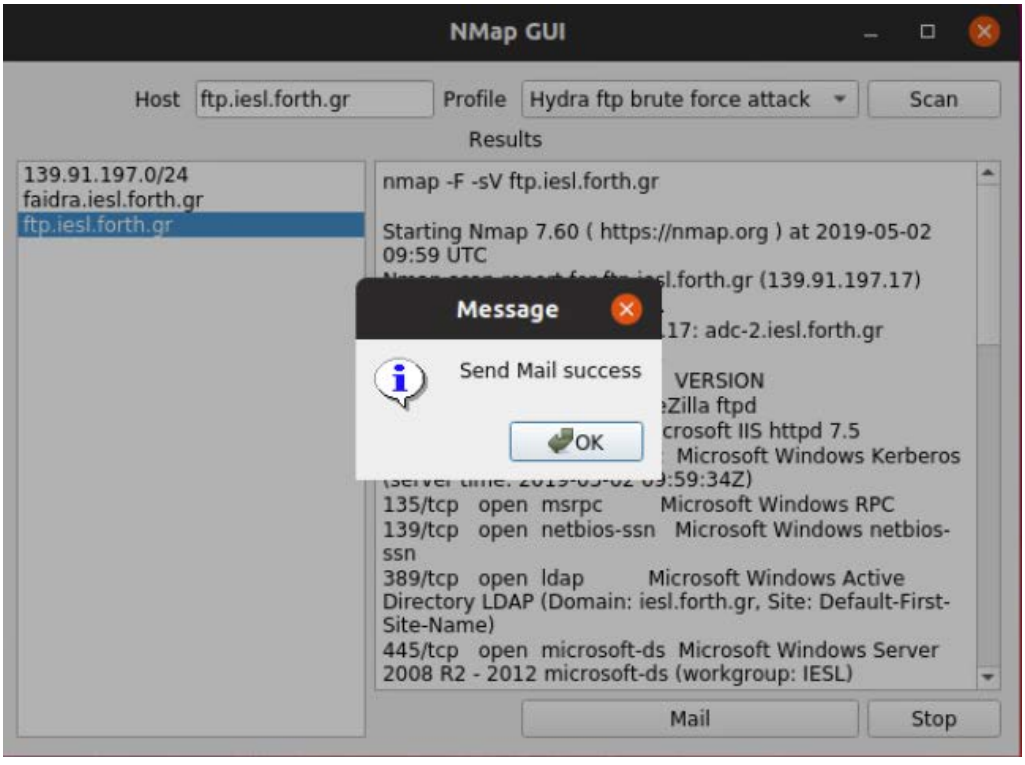
Εικόνα 5-30 - ftp server scan

Μετά από 4 λεπτά το brute force attack με το πρόγραμμα hydra ανίχνευσε ένα anonymous ftp διακομιστή με password οποιαδήποτε λέξη. Στο παρακάτω παράθυρο φαίνεται το αποτέλεσμα της δοκιμής:



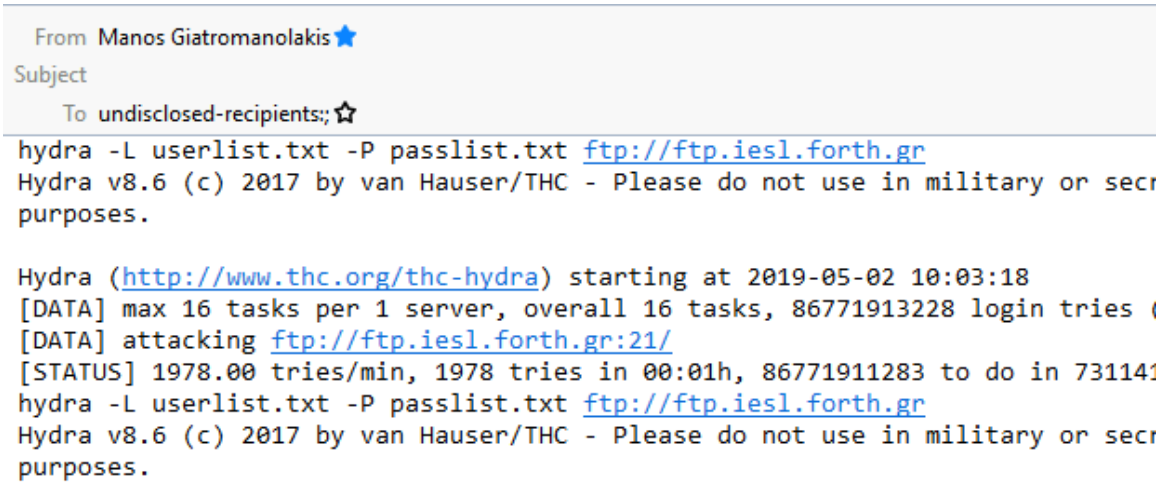
Εικόνα 5-31 - hydra ftp attack

Εφόσον έχουμε αποτελέσματα μπορούμε να δοκιμάσουμε την λειτουργία αποστολής αποτελεσμάτων με email:



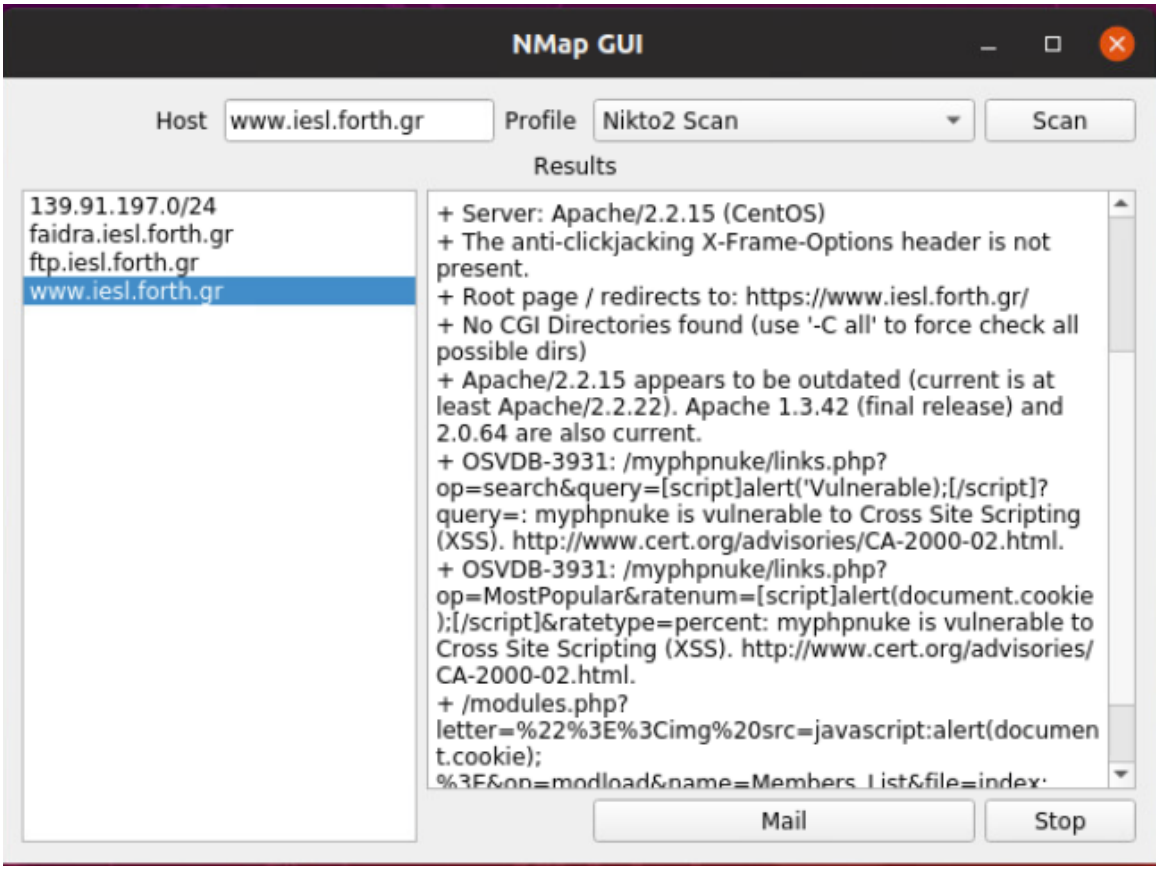
Εικόνα 5-32 - send email feature

Με μικρή αναμονή παραλάβαμε email με τα αποτελέσματα της σάρωσης και της δοκιμής brute force:



Εικόνα 5-33 - email results

Επόμενη δοκιμή διείσδυσης είναι ο έλεγχος του web διακομιστή με το πρόγραμμα nikto:



Εικόνα 5-34 - nikto2 scan

Το nikto2 ανίχνευσε την έκδοση του λογισμικού www καθώς και τα modules που είναι ενεργά, πρότεινε επίσης την αντικατάσταση του search module το οποίο χαρακτήρισε ως ευπάθεια.

Μετά το πέρας των παραπάνω δοκιμών διαπιστώνετε η καλή λειτουργία του bootable Linux cdrom και η σωστή υλοποίηση όλων των προδιαγραφών που τέθηκαν στην φάση της ανάλυσης.

5.6 Αξιολόγηση λογισμικού

Μετά την υλοποίηση του λογισμικού θα πρέπει να αξιολογηθεί ως προς την κάλυψη των απαιτήσεων και των προδιαγραφών που τέθηκαν κατά την ανάλυση και τον σχεδιασμό του. Η διαδικασία της αξιολόγησης γίνεται μετά την εγκατάσταση του στα πλαίσια υποστήριξης έτσι ώστε να εντοπιστούν πιθανές αλλαγές και βελτιώσεις που πρέπει να προωθηθούν στο τμήμα ανάπτυξης.

Η διαδικασία ξεκινά δημιουργώντας ένα checklist το όποιο συμπληρώνετε από τους χρήστες του λογισμικού μετά από κάποιο διάστημα χρήσης. Για την δημιουργία του εντύπου-checklist λαμβάνονται υπόψη οι αρχικές προδιαγραφές που τέθηκαν κατά την φάση του σχεδιασμού. Μετά την συμπλήρωση του εντύπου, τα αποτελέσματα καταγράφονται και αναλύονται έτσι ώστε να βεβαιωθούμε ότι το παραδοτέο πληροί τις αρχικές απαιτήσεις. Τέλος στην περίπτωση που κάποια σημεία επιδέχονται βελτιώσεις παραδίδονται στους προγραμματιστές για την ανάπτυξη της επόμενης έκδοσης του λογισμικού.

5.6.1 Έντυπο αξιολόγησης λογισμικού

Σύμφωνα με τις προδιαγραφές που καταγράψαμε στο κεφάλαιο 5.4.2 δημιουργήσαμε το παρακάτω έντυπο αξιολόγησης με αρχικό template το software testing checklist [61] το οποίο συμπληρώθηκε από εμένα με την ιδιότητα του IT Manager και μετα από αρκετές δοκιμές του NMap GUI.

Application Testing Checklist			
Tested By	Tester	Date	11/5/2019
Application Name	NMAP GUI		

Procedure	Expected Result	Pass/Fail (P/F)	Actual Results/Comments
Bootable Linux Functionality			
Εύκολη εγκατάσταση	ΝΑΙ	P	Εγγραφή του disk image σε cdrom η usb stick
Γρήγορη εκκίνηση	ΝΑΙ	P	Το Ubuntu Linux ξεκινά σε 17 δευτερόλεπτα χωρίς να απαιτείται διεπαφή με το χρήστη.
Εύκολο περιβάλλον παραμετροποίησης συστήματος	ΝΑΙ	P	Το Ubuntu Linux παρέχει στο χρήστη πινάκα ελέγχου για την διαχείριση του δικτύου και των εφαρμογών
Αναγνώριση υλικού	ΝΑΙ	P	Το Ubuntu Linux εμπεριέχει αρκετά αρθρώματα πυρήνα (modules) για την αναγνώριση των περισσοτέρων υπολογιστών και υλικού
Application Functionality			
Εύκολη πρόσβαση στην εφαρμογή	ΝΑΙ	P	Υπάρχει εικονίδιο στην επιφάνεια εργασίας του χρήστη

Υποστήριξη εκτέλεσης δοκιμών διείσδυσης χωρίς την χρήση γραφικού περιβάλλοντος	ΝΑΙ	P	Για κάθε δοκιμή διείσδυσης παρέχετε bash script για την εκτέλεση της από περιβάλλον τερματικού
Εύκολη επιλογή τεχνικής σάρωσης από το χρήστη	ΝΑΙ	P	Εμφάνιση των τεχνικών σάρωσης με μορφή pull down menu
Εύκολη εισαγωγή νέων δοκιμών διείσδυσης	ΟΧΙ	F	Ο χρήστης πρέπει να εισάγει γραμμές κώδικα στην εφαρμογή
Η διεπαφή με το χρήστη να είναι όσο το δυνατόν ελαχίστη	ΝΑΙ	P	Ο χρήστης εισάγει μόνο το σύστημα-στόχος
Ενημέρωση για την κατάσταση σάρωσης	ΝΑΙ	P	Το nmap ενημερώνει κάθε 10 δευτερόλεπτα για το status της δοκιμής
Ικανότητα εκτέλεσης brute-force attack	ΝΑΙ	P	Το πρόγραμμα παρέχει δυο επιλογές brute force μέσω των προγραμμάτων Nikto2 και Hydra
Ικανότητα ακύρωσης μιας δοκιμής	ΝΑΙ	P	Υπάρχει button «Stop» που ακυρώνει μια σάρωση
Ικανότητα αποστολής αποτελεσμάτων με email	ΝΑΙ	P	Υπάρχει button «Mail» που αποστέλλει τα αποτελέσματα των δοκιμών με email στο χρήστη
Εισαγωγή διαφορετικού email	ΟΧΙ	F	Ο χρήστης θα πρέπει να εισάγει το email του μέσα στον κώδικα πριν την εκτέλεση της εφαρμογής

5.6.2 Ανάλυση αποτελεσμάτων αξιολόγησης

Σύμφωνα με το checklist το λογισμικό μας καλύπτει τις προδιαγραφές που τέθηκαν στο στάδιο της ανάλυσης. Κατά την δοκιμή του λογισμικού σε πραγματικό περιβάλλον χρήσης διαπιστώθηκαν ελλείψεις που αφορούν την περεταίρω ανάπτυξη του λογισμικού. Αυτό καταχωρήθηκε και στο έντυπο αξιολόγησης έτσι ώστε να δοθεί μελλοντικά στους προγραμματιστές για την υλοποίηση τους.

Ποιο αναλυτικά σύμφωνα με το checklist υπάρχει ανάγκη δημιουργίας διεπαφής με το χρήστη για την εισαγωγή νέων δοκιμών σάρωσης και τεχνικών διείσδυσης καθώς και την διεπαφή για την εισαγωγή email του χρήστη. Οι συγκεκριμένες απαιτήσεις δεν καταχωρήθηκαν στην φάση της ανάλυσης με αποτέλεσμα δικαίως να είναι ελλιπές στην υλοποίηση τους. Σύμφωνα με το μοντέλο καταρράκτης εφόσον το έργο έφτασε στο τελικό στάδιο και δόθηκε σε παραγωγικό επίπεδο, τα αποτελέσματα της αξιολόγησης προωθούνται στην ομάδα ανάπτυξης του έργου για να συμπεριληφθούν στην επόμενη έκδοση του λογισμικού.

Κεφάλαιο 6

Συμπεράσματα – Επίλογος

Η προστασία των πληροφοριακών συστημάτων είναι μια πρόκληση για κάθε διαχειριστή συστημάτων που δεν σταματά πότε. Καθημερινά νέες ατέλειες σε λογισμικά άλλα και σε πρωτόκολλα δικτύων και λειτουργικών συστημάτων δημιουργούν ευπάθειες που μπορούν να προκαλέσουν τεράστιες ζημιές σε οργανισμούς και υπηρεσίες. Ο εντοπισμός αυτών των ευπαθειών πρέπει να είναι διαρκής έτσι ώστε να ανιχνεύονται και να διαρθρώνονται πριν γίνουν απειλές. Τα αποτελέσματα των δοκιμών διείσδυσης επιτρέπουν στους διαχειριστές να διαρθρώσουν τις ευπάθειες και να αναπτύξουν ένα σχέδιο ασφάλειας που θα προστατεύει τον οργανισμό από κακόβουλες επιθέσεις. Επιπλέον προλαμβάνουν τις απώλειες από τυχόν επιτυχείς επιθέσεις και αυξάνουν την αξιοπιστία και το κύρος του οργανισμού προς τους πελάτες του.

Η παρούσα μεταπτυχιακή διατριβή ασχολήθηκε με τον έλεγχο τρωτότητας των πληροφοριακών συστημάτων, αναλύοντας εργαλεία δοκιμών διείσδυσης και σάρωσης τα οποία αξιολόγησε. Στην συνέχεια έλεγξε και κατέγραψε την ευχρηστία τους, το κόστος τους και την αποτελεσματικότητας τους εκτελώντας τα σε διαφορετικά δίκτυα και με διαφορετικούς παραμέτρους. Τα αποτελέσματα των δοκιμών έκριναν την απόδοση των εργαλείων και εμφάνισαν τις αδυναμίες τους.

Με γνώμονα τις αδυναμίες των προγραμμάτων, δημιουργήσαμε μια εφαρμογή με βασικό σημείο την ευχρηστία και την φορητότητα έτσι ώστε να μπορούμε να εκτελέσουμε άμεσα δοκιμές διείσδυσης από οποιοδήποτε δίκτυο και υπολογιστή. Η υλοποίηση της εφαρμογής ακολούθησε το μοντέλο ανάπτυξης λογισμικού καταρράκτης και δοκιμάστηκε σε πραγματικό περιβάλλον μεγάλου οργανισμού με πολλά συστήματα και υποδομές. Η εφαρμογή στηρίχτηκε στο nmap που με κατάλληλους παραμέτρους μπορεί αν εκτελέσει δοκιμές σάρωσης χωρίς να γίνει αντιληπτό από τα συστήματα παρακολούθησης δικτυού, έτσι μπορούμε να αξιολογήσουμε την ασφάλεια ενός πληροφοριακού συστήματος και να ανιχνεύσουμε τις ευπάθειες του.

Συμπερασματικά η διαδικασία σάρωσης και διείσδυσης για την ανίχνευση των ευπαθειών είναι μια συνεχής διεργασία που πρέπει να εκτελείται μεθοδικά και επαναλαμβανόμενα, επίσης πρέπει να αναπτύσσεται και να προσαρμόζεται στα νέα λογισμικά, στα νέα πρωτόκολλα και να λαμβάνει υπόψη τις νέες τεχνικές που χρησιμοποιούν οι κακόβουλοι χρήστες έτσι ώστε να προσαρμόζεται ανάλογα. Τέλος τα αποτελέσματα των δοκιμών θα πρέπει να εξετάζονται και να αναλύονται εξονυχιστικά ώστε να βοηθούν στην διατήρηση και στην ανάπτυξη της πολιτικής ασφάλειας του οργανισμού.

Μελλοντική ανάπτυξη

Η εφαρμογή που υλοποιήσαμε στην παρούσα διατριβή παρέχει τις βασικές τεχνικές σάρωσης/διείσδυσης για την ανίχνευση των ευπαθειών με κύριο στόχο την μη εξουσιοδοτημένη πρόσβαση στα πληροφοριακά συστήματα που ελέγχονται. Ο συνδυασμός αυτών των τεχνικών μας δίνει μια ολοκληρωμένη εικόνα για την ασφάλεια του οργανισμού με κύριο σκοπό την αναπροσαρμογή της πολιτικής ασφαλείας πριν οι ευπάθειες χρησιμοποιηθούν από κακόβουλους χρήστες και προκαλέσουν απώλειες.

Μελλοντικά μπορούν να εισαχθούν νέες τεχνικές σάρωσης χρησιμοποιώντας νέα εργαλεία και μεθόδους έτσι ώστε η εφαρμογή να είναι περισσότερο αποτελεσματική. Μια άλλη σημαντική προσθήκη είναι η ικανότητα ελέγχου των αποτελεσμάτων από την ιδία την εφαρμογή έτσι ώστε να προτείνει μεθόδους επιδιόρθωσης των ευπαθειών. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας τις βάσεις ευπαθειών που υπάρχουν και να προτείνει λύσεις και επιδιορθώσεις προγραμμάτων για την

εξάλειψη των ευπαθειών. Σημαντική προσθήκη στην εφαρμογή η οποία αποκαλύφθηκε κατά την αξιολόγηση της είναι η υλοποίηση ενός πίνακα ελέγχου, έτσι ώστε ο χρήστης να μπορεί να εισάγει το email του, τον smtp server καθώς και νέες μεθόδους η τεχνικές σάρωσης/διείσδυσης.

Τέλος λόγω της ραγδαίας ανάπτυξης του Internet πάντα θα υπάρχουν ευπάθειες και πάντα οι κακόβουλοι χρήστες με συνεχείς προσπάθειες θα τις εκμεταλλευτούν έτσι ώστε να αποκτήσουν πρόσβαση στο σημαντικότερο αγαθό σήμερα, την πληροφορία. Συνεπώς μια ολοκληρωμένη στρατηγική ελέγχου ασφάλειας μόνο θετικά αποτελέσματα μπορεί να επιφέρει στην βιωσιμότητα ενός οργανισμού.

Βιβλιογραφία

[1] G. Weidman, Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press.

[2] W. Knowles, A. Baron and T. McGarr, "The simulated security assessment ecosystem: Does penetration testing need standardisation?," *computers & s e c u r i t y,* vol. 62, 2016.

[3] P. Bosco, "Intrusion Detection and Prevention Systems Cheat," 2016.

[4] T. Andrew, "A guide to penetration," *Network Security,* vol. 9, Aug 2014.

[5] Y. John, "Using penetration testing to enhance your company's security," *Computer Fraud & Security,* vol. 17, Apr 2013.

[6] H. Liu and Z. Li, "Methodology of Network Intrusion Detection System Penetration Testing," in *The Ninth International Conference on Web-Age Information Management*, China, 2013.

[7] E. Bou-Harb, M. Debbabi and C. Assi, "Cyber Scanning: A Comprehensive Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 3, 2014.

[8] K. Satyendra and S. Abhilash, "Rule-Based Network Intrusion Detection System for Port Scanning with Efficient Port Scan Detection Rules Using Snort," *International Journal of Future Generation Communication and Networking,* vol. 9, no. 6, 2016.

[9] G. Tarun , 2018. [Online]. Available: https://www.computerweekly.com/tip/5-penetration-test-tools-to-secure-your-network.

[10] Apr 2018. [Online]. Available: https://resources.infosecinstitute.com/the-top-5-pentesting-tools-you-will-ever-need/.

[11] S. Tushar and S. Lakshman, Linux Shell Scripting, 2 ed., BIRMINGHAM: Packt Publishing, 2013.

[12] K. Urupoj, S. Surasak and J. Wipa, "A Rule-based Approach for Port Scanning Detection," 2013.

[13] M. Ohlson, Python 2.6 Graphics Cookbook, BIRMINGHAM: Packt Publishing Ltd., 2010.

[14] J. Narayan Goel and M. bm, "Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology," *Procedia Computer Science,* vol. 710, 2015.

[15] wikipedia, "https://el.wikipedia.org," wikipedia, 2018. [Online]. Available: https://el.wikipedia.org/wiki/Ασφάλεια_πληροφοριακών_συστημάτων.

[16] SecureReading, "INFOBASICS-Basic Concept of Information Security," *INFOBASICS,* October 2016.

[17] TechTarget, "Confidentiality, integrity and availability, also known as the CIA triad," 2014. [Online]. Available: https://whatis.techtarget.com/definition/Confidentiality-integrity-and-

availability-CIA.

[18] wbisct, "https://wbisct.net/," WBISCT PTY LTD, 2018. [Online]. Available: https://wbisct.net/2018/04/27/10-security-domains-as-per-isc2-org/.

[19] isc2, "https://www.isc2.org," 2018, [Online]. Available: https://en.wikibooks.org/wiki/Fundamentals_of_Information_Systems_Security.

[20] NIST, "I N F O R M A T I O N S E C U R I T Y," Information Technology Laboratory, Gaithersburg, 2012.

[21] J. Bayne, "An Overview of Threat and Risk Assessment," *Auditing & Assessment,* 2002.

[22] WikiPedia, "https://en.wikipedia.org," WikiPedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/CRAMM.

[23] G. Leynolds and M. Stanclift, "https://www.partneresi.com," Parthner Engineering and Science Inc, [Online]. Available: https://www.partneresi.com/services/environmental-consulting/records-search-and-risk-assessment.

[24] EU, "https://eugdpr.org/," 2018. [Online]. Available: https://eugdpr.org/.

[25] WikiPedia, "https://en.wikipedia.org," WikiPedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/ISO/IEC_27001.

[26] SGS, "https://www.sgsgroup.com.hk/," 2017. [Online]. Available: https://www.sgsgroup.com.hk/.

[27] N. Wood, "Understanding Risk and Resilience to Natural Hazards," *USGS,* p. 1, 2011.

[28] C. Biancotti, "https://voxeu.org," VOX, Jun 2017. [Online]. Available: https://voxeu.org/article/cyber-attacks-economic-policy-challenge.

[29] Palisade, "http://www.palisade.com," Palisade, [Online]. Available: http://www.palisade.com/risk/monte_carlo_simulation.asp.

[30] WordPress, "https://www.webnots.com," 5 November 2017. [Online]. Available: https://www.webnots.com/why-disaster-recovery-plan-for-wordpress-site-is-important/.

[31] WikiPedia, "WikiPedia," [Online]. Available: https://en.wikipedia.org/wiki/Morris_worm.

[32] UMSI, "UMSI," [Online]. Available: http://si410wiki.sites.uofmhosting.net/index.php/Morris_Worm.

[33] J. Anderson, "The History of Penetration Testing," [Online]. Available: https://resources.infosecinstitute.com/category/certifications-training/pentesting-certifications/pentesting-history/#gref.

[34] A. Indicium, "indiciumassessment," November 2015. [Online]. Available: indiciumassessment.blogspot.com/2015/09/what-is-vapt-testing-and-vapt-types.html.

[35] Akhil, "What are the advantages of penetration testing services?," Quora, January 2017. [Online]. Available: https://www.quora.com/What-are-the-advantages-of-penetration-testing-services.

[36] TutorialsPoint, "https://www.tutorialspoint.com," SimplyEasyLearning, 2016. [Online]. Available: https://www.tutorialspoint.com/penetration_testing/penetration_testing_limitations.htm.

[37] F. O. f. I. S. (BSI), "A penetration testing model," BSI, Bonn, 2015.

[38] I. Institute, "https://resources.infosecinstitute.com," InfoSec Institute, 30 July 2018. [Online]. Available: https://resources.infosecinstitute.com/what-are-black-box-grey-box-and-white-box-penetration-testing/.

[39] K. Scarfone and M. Souppaya , "Technical Guide to Information Security Testing and Assessment".*Special Publication 800-115 .*

[40] T. P. Team, "The Penetration Testing Execution Standard Documentation," *PTES,* vol. 1, no. 1, 2017.

[41] K. Bourne, "Introducing In-App Pen Test Reports," blog.cobalt.io, 29 March 2017. [Online]. Available: https://blog.cobalt.io/introducing-in-app-pen-test-reports-8c7b551e8a2b.

[42] G. Lyon, "https://nmap.org/," NMap, 2018. [Online]. Available: https://nmap.org/.

[43] g. networks, "OpenVas," 2005, [Online]. Available: http://www.openvas.org/.

[44] Tenable, "Nessus," Tenable, 2018. [Online]. Available: https://www.tenable.com.

[45] R. LLC, "MetaSploit," MetaSploit, 2018. [Online]. Available: https://www.metasploit.com.

[46] M. Sarrel, "GFI LanGuard," PCMag, 2015. [Online]. Available: https://www.pcmag.com/article2/0,2817,2488088,00.asp.

[47] Rapid, "Nexpose," Rapid7, 2018. [Online]. Available: https://www.rapid7.com/products/nexpose/.

[48] C. Sullo, "Nikto2," Cirt, 2018. [Online]. Available: https://www.cirt.net/Nikto2.

[49] J. Irvin, "Core Impact," SecureAuth, 2018. [Online]. Available: https://www.secureauth.com/products/penetration-testing/core-impact.

[50] Kali, "Kali Linux," Kali, 2018. [Online]. Available: https://www.kali.org/.

[51] BlackArch, "BlackArch," BlackArch, 01 12 2012. [Online]. Available: https://blackarch.org/.

[52] BackBox, "Backbox," Backbox, 2018. [Online]. Available: https://backbox.org.

[53] S. Chawla, "NMAP CHEAT-SHEET," Medium, 20 5 2018. [Online]. Available: https://medium.com/@infosecsanyam/nmap-cheat-sheet-nmap-scanning-types-scanning-commands-nse-scripts-868a7bd7f692.

[54] SK, "How To Create Custom Ubuntu Live CD Image," ostechnix, 23 October 2018. [Online]. Available: https://www.ostechnix.com/create-custom-ubuntu-live-cd-image/.

[55] WikiBooks, "wikibooks," wikibooks.org, 2019. [Online]. Available: https://en.wikibooks.org/wiki/Bash_Shell_Scripting.

[56] riverbankcomputing, "www.riverbankcomputing.com," 1 3 2019. [Online]. Available: https://www.riverbankcomputing.com/software/pyqt/intro.

[57] T. Dataflair, "data-flair.training," data-flair.training, 1 3 2019. [Online]. Available: https://data-flair.training/blogs/python-pyqt5-tutorial/.

[58] WikiPedia, "Systems development life cycle," WikiPedia, 1 Jan 2019. [Online]. Available: https://en.wikipedia.org/wiki/Systems_development_life_cycle.

[59] SmartSheet, "https://www.smartsheet.com," 10 Mar 2018. [Online]. Available: https://www.smartsheet.com/system-development-life-cycle-guide.

[60] W. Royce, "USC Student Computing Facility," 2015. [Online]. Available: http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf.

[61] Q. Strong, "strongqa.com," 5 2017. [Online]. Available: https://strongqa.com/qa-portal/testing-docs-templates/checklist.

ΠΑΡΑΡΤΗΜΑ 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import os
import signal
import sys
import subprocess
import threading
import time
import queue
from PyQt5.QtWidgets import (QWidget, QLabel, QLineEdit, QComboBox,
QTextEdit, QListWidget, QPushButton, QGridLayout, QApplication)
from PyQt5.QtWidgets import QMessageBox, QDesktopWidget
from PyQt5.QtCore import Qt, pyqtSignal, QObject

import smtplib

INTERVAL = float(1)/float(16)
KILL_ALL = False

SERVER = 'mail.iesl.forth.gr'
PORT = 25
FROM = 'nmapgui@iesl.forth.gr'
PSWD = ''
TO = ['gmanos@iesl.forth.gr']
SUBJECT = u'New nMap report'

profiles = {
    'Fast Scan': 'nmap -F -sV {0}',
    'Aggressive TCPSYN Scan': 'nmap -sV -sS -A -T4 {0}',
    'Full ACN Scan': 'nmap --stats-every 5s -p 1-65535 -sV -sA -T4 {0}',
    'Full FIN Scan': 'nmap --stats-every 60s -p 1-65535 -sV -sF -T4 {0}',
    'Full NULL Scan': 'nmap -p 1-65535 -sV -sN -T4 {0}',
    'Full TCPSYN Scan': 'nmap -p 1-65535 -sV -sS -T4 {0}',
    'Full TCPUDP Scan': 'nmap --stats-every 60s -sV -sT -sU -T4 {0}',
    'Full WINDOW Scan': 'nmap --stats-every 60s -p 1-65535 -sV -sW -T4
{0}',
    'Full XMAS Scan': 'nmap --stats-every 60s -p 1-65535 -sV -sX -T4 {0}',
    'Nikto2 Scan': 'nikto -h {0}',
    'Hydra ftp brute force attack': 'hydra -L userlist.txt -P passlist.txt
ftp://{0}',
    'Hydra ssh brute force attack': 'hydra -L userlist.txt -P passlist.txt
ssh://{0}',
}
process_threads = dict()

class AsyncFileReader(threading.Thread):
    """
    Helper class to implement asynchronous reading of a file
    in a separate thread. Pushes read lines on a queue to
    be consumed in another thread.
    """
    def __init__(self, fd, mqueue):
        assert isinstance(mqueue, queue.Queue)
        assert callable(fd.readline)
        threading.Thread.__init__(self)
        self._fd = fd
        self._queue = mqueue

    def run(self):
```

```
        """The body of the tread: read lines and put them on the queue."""
        try:
            for line in iter(self._fd.readline, ''):
                self._queue.put(line)
        except ValueError:
            # When ant process is terminated an exception is raised and
this ensures thread exit
            pass

    def eof(self):
        """Check whether there is no more content to expect."""
        return not self.is_alive() and self._queue.empty()


class LogcatThread(threading.Thread):
    def __init__(self, parent, command, host, setvalue):
        """
        @param parent: The gui object that should recieve the value
        @param command: The command to pass to nmap
        """
        threading.Thread.__init__(self)
        self._parent = parent
        self._command = command
        self._host = host
        self._setvalue = setvalue
        self._stop_event = threading.Event()
        self._ant = None
        self.daemon = False
        self.exiting = False

    def run(self):
        """Overrides Thread.run. Don't call this directly its called
internally
        when you call Thread.start().
        """
        wait = True
        self._ant = LogcatAnt(self, self._command, self._host,
self._setvalue)
        while not self.exiting and wait:
            wait = self._ant.run()
            global KILL_ALL
            if KILL_ALL:
                # print('Killed thread: {0}'.format(id(self)))
                break

        # print('Stopped')
        self._ant = None  # del self._ant
        self._stop_event.set()

    def kill(self):
        try:
            self._ant.kill()
            self._stop_event.set()
        except AttributeError:
            pass


class LogcatAnt:
    def __init__(self, parent, command, host, setvalue):
        self.parent = parent
        self.command = command
        self.host = host
        self.setvalue = setvalue
```

```
        self.message_queue = queue.Queue()
        self.process = None
        self._done = False

    def run(self):
        self.process = subprocess.Popen(self.command, shell=True,
stdout=subprocess.PIPE, preexec_fn=os.setsid)

        # Launch the asynchronous readers of the process' stdout.
        stdout_reader = AsyncFileReader(self.process.stdout,
self.message_queue)
        stdout_reader.start()

        # Check the queues if we received some output (until there is
nothing more to get).
        while not stdout_reader.eof():
            time.sleep(INTERVAL)
            while not self.message_queue.empty():
                line = self.message_queue.get()
                self.setvalue(self.host, line.decode('utf-8').strip())
                self.message_queue.task_done()
                if line.decode('utf-8').startswith('Nmap done'):
                    self._done = True
                    break
            if self._done:
                self.process.terminate()
                self.process.communicate()
                return False

        return False

    def kill(self):
        try:
            self._done = True
            self.message_queue = queue.Queue()
            # self.process.kill()
            # self.process.terminate()
            # self.process.communicate()
            os.killpg(os.getpgid(self.process.pid), signal.SIGTERM)
        except AttributeError:
            pass


class Signal(QObject):
    new_line = pyqtSignal()


class ManosnMapFrame(QWidget):
    def __init__(self):
        super().__init__()

        self.profile_names = []
        self.profile_values = []
        self.history = dict()
        self._signal = Signal()
        self._signal.new_line.connect(self.on_history_changed)

        for key, value in profiles.items():
            self.profile_names.append(key)
            self.profile_values.append(value)

        label_host = QLabel('Host ')
        label_host.setAlignment(Qt.AlignRight | Qt.AlignVCenter)
```

```
        self.host = QLineEdit()
        label_profile = QLabel('Profile ')
        label_profile.setAlignment(Qt.AlignRight | Qt.AlignVCenter)
        self.profile = QComboBox()
        self.host.setText('localhost')
        self.profile.addItems(self.profile_names)
        self.profile.setCurrentIndex(0)

        self.scan = QPushButton('Scan')
        self.scan.clicked.connect(self.on_scan)

        label_results = QLabel('Results ')
        label_results.setAlignment(Qt.AlignCenter)
        self.history_view = QListWidget()

self.history_view.itemSelectionChanged.connect(self.on_history_changed)
        self.results = QTextEdit()

        self.mail = QPushButton('Mail')
        self.mail.clicked.connect(self.on_mail)
        self.stop = QPushButton('Stop')
        self.stop.clicked.connect(self.on_stop)

        grid = QGridLayout()
        grid.setSpacing(5)

        grid.addWidget(label_host, 1, 0)
        grid.addWidget(self.host, 1, 1, 1, 2)
        grid.addWidget(label_profile, 1, 3)
        grid.addWidget(self.profile, 1, 4)
        grid.addWidget(self.scan, 1, 5)

        grid.addWidget(label_results, 2, 0, 1, 6)
        grid.addWidget(self.history_view, 3, 0, 5, 2)
        grid.addWidget(self.results, 3, 2, 4, 4)
        grid.addWidget(self.mail, 7, 4)
        grid.addWidget(self.stop, 7, 5)

        grid.setRowStretch(3, 1)
        # grid.setColumnStretch(1, 1)
        # grid.setColumnStretch(3, 1)
        self.setLayout(grid)

        self.setGeometry(690, 600, 350, 300)
        self.setWindowTitle('NMap GUI')
        self.show()
        self.center()

    def on_scan(self):
        host = self.host.text().strip()
        if host != u'':
            profile = self.profile_values[self.profile.currentIndex()]
            command = profile.format(host)

            if host not in self.history.keys():
                self.history[host] = command
                self.history_view.addItem(host)

self.history_view.setCurrentItem(self.history_view.item(self.history_view.c
ount()-1))
                self.results.setText('')
            else:
                self.history[host] += '\n'+command
```

```
            global process_threads
            try:
                host_thread = process_threads.pop(host)
                host_thread.kill()
            except (KeyError, AttributeError):
                pass

            process_threads[host] = LogcatThread(self, command, host,
self.on_append)
            process_threads[host].start()

    def on_mail(self):
        host = self.host.text().strip()
        if host != u'':
            profile = self.profile_values[self.profile.currentIndex()]

            body = self.results.toPlainText()

            message = u"""\
            From: %s
            To: %s
            Subject: %s

            %s
            """ % (FROM, ", ".join(TO), SUBJECT, body)

            try:
                server = smtplib.SMTP(SERVER)
                server.set_debuglevel(False)
                server.ehlo_or_helo_if_needed()
                #server.starttls()
                #server.ehlo_or_helo_if_needed()
                #server.login(FROM, PSWD)
                server.sendmail(FROM, TO, message)
                server.quit()
                QMessageBox.information(self, 'Message', 'Send Mail
success', QMessageBox.Ok, QMessageBox.Ok)
            except BaseException as e:
                QMessageBox.information(self, 'Message',
                    'Send Mail failed with the following
error:\n{0}'.format(e.__str__()),
                    QMessageBox.Ok, QMessageBox.Ok)

    def on_stop(self):
        host = self.host.text().strip()
        if host != u'':
            global process_threads
            process_threads[host].kill()
            QMessageBox.information(self, 'Message', 'The scan has
terminated', QMessageBox.Ok, QMessageBox.Ok)

    def on_append(self, host, line):
        self.history[host] += '\n'+line
        # current = self.history_view.currentItem()
        self._signal.new_line.emit()

    def on_history_changed(self):
        current = self.history_view.currentItem()
        if current is not None:
            self.results.setText(self.history[current.text()])
        else:
            self.results.setText('')
```

```
    def center(self):
        qr = self.frameGeometry()
        cp = QDesktopWidget().availableGeometry().center()
        qr.moveCenter(cp)
        self.move(qr.topLeft())

    def closeEvent(self, event):
        reply = QMessageBox.question(self, 'Message', 'Are you sure to
quit?', QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
        if reply == QMessageBox.Yes:
            global KILL_ALL
            KILL_ALL = True
            event.accept()
        else:
            event.ignore()


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = ManosnMapFrame()
    sys.exit(app.exec_())
```