Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Automating red team attacks in cyber ranges

Χρίστος Δημητρίου

Επιβλέπουσα Καθηγήτρια
Αδαμαντίνη Περατικού

Δεκέμβριος 2021

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Automating red team attacks in cyber ranges

Χρίστος Δημητρίου

Επιβλέπουσα Καθηγήτρια
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..

Δεκέμβριος 2021

Περίληψη

Η ασφάλεια των πληροφοριακών συστημάτων γίνεται όλο και πιο σημαντική χρόνο με τον χρόνο. Η χρήση των υπολογιστικών δικτύων έχει αυξηθεί σε όλους τους τομείς, με την πανδημία του κορονοϊού να έχει επιταχύνει την μετάβαση πολλών δραστηριοτήτων από τον φυσικό στον ψηφιακό κόσμο. Μαζί με την αύξηση της χρήσης των δικτύων παρατηρείται και η αύξηση των κυβερνοεγκλημάτων.

Στόχος της παρούσας διατριβής είναι να μελετηθούν οι τρόποι με τους οποίους μια red team ελέγχει ένα σύστημα για να εντοπίσει ευπάθειες και ευάλωτα σημεία. Με βάση την μελέτη της βιβλιογραφίας φάνηκε ότι τα εργαλεία που υπάρχουν μέχρι τώρα υστερούν σε ευχρηστία, απλότητα, και αυτοματοποίηση.

Γι’ αυτό κρίθηκε αναγκαία η υλοποίηση του αυτοματοποιημένου εργαλείου «ARTAS» ώστε να προστεθεί ακόμα ένα εργαλείο στους επαγγελματίες ασφάλειας πληροφοριακών συστημάτων. Το εργαλείο «ARTAS» δοκιμάστηκε με επιτυχία σε cyber-range περιβάλλον και βελτιώνει σημαντικά την ευχρηστία και την απλότητα των διεργασιών που αυτοματοποιήθηκαν.

Summary

Information systems security is becoming more and more important year by year. The use of computer networks has increased in all sectors and due to the coronavirus pandemic, the transition of many activities from the physical to the digital world has been accelerated. However, along with the increase in the use of networks, there is also an increase in cybercrime.

The purpose of this dissertation is to study the ways in which a red team examines a system in order to identify vulnerable and weak points. The literature review revealed that the existing tools lag behind in usability, simplicity and automation.

For this reason, the implementation of the automated tool "ARTAS" was deemed necessary, in order to provide an additional tool to the information systems security professionals. The "ARTAS" tool has been successfully tested in a cyber-range environment and significantly improves the usability and simplicity of automated procedures.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια μου κα. Αδαμαντίνη Περατικού, για την χρήσιμη καθοδήγηση που μου παρείχε κατά τη διάρκεια εκπόνησης της μεταπτυχιακής μου διατριβής και για την άψογη συνεργασία που είχαμε όλο αυτό το διάστημα. Ευχαριστώ επίσης την οικογένεια και τους φίλους μου για την στήριξη τους.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Στις μέρες μας η ασφάλεια των πληροφοριακών συστημάτων γίνεται όλο και πιο σημαντική καθώς παρατηρείται αύξηση της χρήσης των υπολογιστικών δικτύων για μεταφορά και αποθήκευση προσωπικών δεδομένων και ιατρικών εγγράφων, για εμπορικές συναλλαγές, και άλλα. Με τα νέα δεδομένα που έχει φέρει η πανδημία του κορονοϊού η μετάβαση πολλών δραστηριοτήτων από τον φυσικό στον ψηφιακό κόσμο έχει επιταχυνθεί, με τρανταχτό παράδειγμα την τηλ-εργασία. Μαζί με την αύξηση της χρήσης των δικτύων παρατηρείται και η αύξηση των κυβερνοεγκλημάτων με την ανάγκη βελτίωσης των υφιστάμενων εργαλείων και πρακτικών, που είναι διαθέσιμα για τον έλεγχο και εντοπισμό ευπαθειών ενός συστήματος, να βρίσκεται στις προτεραιότητες. Πολλές εταιρείες προσλαμβάνουν ομάδες ή άτομα ώστε να ελέγξουν τα συστήματα και να παρουσιάσουν τυχόν ευπάθειες που θα εντοπιστούν. Αυτή η διαδικασία κοστίζει αρκετά χρήματα και είναι χρονοβόρα.Στην παρούσα διατριβή έχει μελετηθεί ένας αριθμός από υπάρχοντα attack scripts και ευπαθειών από βάσεις δεδομένων, που μπορούν να χρησιμοποιηθούν, με στόχο την αυτοματοποίηση και την αύξηση της ευχρηστίας και της απλότητας τους. Αυτό έχει ως προσδοκώμενο αποτέλεσμα την γρηγορότερη λήψη αποφάσεων στις εκάστοτε καταστάσεις. Για το σκοπό αυτό έχει δημιουργηθεί ένα εργαλείο και δοκιμάστηκε σε περιβάλλοντα Cyber-Range για να διαπιστωθεί η αποτελεσματικότητα του αλλά και οι αδυναμίες του.

1.1 Ανάγκη για μελέτη

Υπάρχει ένας μεγάλος αριθμός εργαλείων που σκοπός τους είναι ο έλεγχος συστημάτων με στόχο τον εντοπισμό ευπαθειών και ευάλωτων σημείων σε αυτά. Αρκετά από τα εργαλεία υστερούν σε ευχρηστία, απλότητα, και αυτοματοποίηση. Παρατηρείται η ανάγκη μελέτης και υλοποίησης ενός αυτοματοποιημένου εργαλείου που θα καλύψει αυτό το κενό και θα αποφέρει οφέλη όπως την γρηγορότερη και πληρέστερη ανάλυση και εντοπισμό ευπαθειών στο σύστημα.

1.2 Ερευνητικά ερωτήματα

Η διατριβή αυτή θα προσπαθήσει να διερευνήσει τα υφιστάμενα red team attack scripts με σκοπό να χρησιμοποιηθούν και να εκτελούνται αυτόματα. Τα ερευνητικά ερωτήματα που προκύπτουν και προσπαθούν να απαντηθούν είναι εάν υπάρχουν επί του παρόντος αυτόματα attack scripts για περιβάλλοντα Cyber-Range, τι databases μπορούν να χρησιμοποιηθούν για την υλοποίηση τους, καθώς και ποιος τρόπος αυτοματοποίησης είναι ο πιο ευέλικτος για να καλυφθεί όλο το φάσμα αναγκών όπως για παράδειγμα η αυτοματοποίηση από μεριάς χρόνου ή από μεριάς user interface.

1.3 Προσδοκόμενα παραδοτέα

Έχει δημιουργηθεί το εργαλείο «ARTAS – Automatic Red Team Attack Scriptss». Τα απαραίτητα αρχεία βρίσκονται στον πιο κάτω σύνδεσμο: https://gitlab.com/cdimit/artas

1.4 Δομή της Διατριβής

Η παρούσα Διατριβή έχει χωριστεί στα ακόλουθα κεφάλαια τα οποία συμπεριλαμβάνουν:

Κεφάλαιο 2: Βιβλιογραφική ανασκόπηση

Σε αυτό το κεφάλαιο γίνεται μια βιβλιογραφική ανασκόπηση, παρουσιάζονται οι διάφορες ορολογίες που χρησιμοποιήθηκαν και άλλες παρόμοιες έρευνες

Κεφάλαιο 3: Μεθοδολογία

Στο κεφάλαιο “μεθοδολογία” γίνεται μια περιγραφή της υλοποίησης της διατριβής και του αυτοματοποιημένου εργαλείου που δημιουργήθηκε. Αναλύονται διάφορα μοντέλα κύκλου ζωής λογισμικού και ποιο έχει επιλεγεί για την υλοποίηση του εργαλείου. Επίσης γίνεται αναφορά στον τύπο της έρευνας και γιατί προτιμήθηκε η ποσοτική.

Κεφάλαιο 4: Υλοποίηση

Σε αυτό το κεφάλαιο παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν και ο τρόπος υλοποίησης του αυτοματοποιημένου προγράμματος που δημιουργήθη

Κεφάλαιο 5: Επίλογος

Στον επίλογο αναγράφονται τα αποτελέσματα και οι εκτιμήσεις που παρατηρήθηκαν, καθώς και η μελλοντική δουλειά που χρειάζεται να γίνει.

Παράρτημα Α:

Το παράρτημα Α περιλαμβάνει αποσπάσματα κώδικα από το εργαλείο που υλοποιήθηκε.

Κεφάλαιο 2
Βιβλιογραφική Ανασκόπηση

2.1 Background

Το να υποδύεσαι τον ρόλο του επιτιθέμενου μπορεί να αποφέρει βελτιώσεις στην αμυντική κατάρτιση σου και της ομάδας σου. Αυτό τον στόχο έχουν οι ασκήσεις με red team και blue team. Η ιστορία αυτών των όρων προέρχεται από τον πρώτο παγκόσμιο πόλεμο όπου η ιδέα ήταν να σχεδιάζονται επιθετικές τακτικές, ώστε σε περίπτωση που αντιμετώπιζαν επιθέσεις θα ήταν πιο έτοιμοι να τις αντιμετωπίσουν. Το βασικό στοιχείο για την επιτυχια μιας red team είναι να μπορει να μπει στην νοοτροπία ενός επιτιθεμενου. Στον τομέα της κυβερνοασφάλειας, οι εταιρείες σε οποιονδήποτε κλάδο μπορούν να επωφεληθούν από μια άσκηση red team-blue team. Η red team έχει ως στόχο τον εντοπισμο αδυναμων σημείων στο συστημα και η blue team έχει ως στόχο να διασφαλίσει ότι τα περιουσιακά στοιχεία της εταιρίας είναι ασφαλή. Στην περίπτωση που η red team βρει μια ευπάθεια και την εκμεταλλευτεί, η blue team πρεπει να την επανορθώσει άμεσα [1].

2.1.1 Red Team

Στον κυβερνοχωρο η υιοθέτηση της προσέγγισης της Red Team βοήθησε επίσης τους οργανισμούς να διατηρήσουν τα περιουσιακά τους στοιχεία πιο ασφαλή. Τα άτομα που αποτελούν αυτή την ομάδα πρέπει να είναι καλά εκπαιδευμένα, με διάφορες δεξιότητες και να γνωρίζουν τις εκάστοτε απειλές. Η ομάδα red team πρέπει να γνωρίζει τις τάσεις, να κατανοεί πώς γίνονται οι τρέχουσες επιθέσεις και να τις εκτελεί με

στόχο να σπάσει το υπάρχων σύστημα ασφαλείας. Σε ορισμένες περιπτώσεις και ανάλογα με τις απαιτήσεις του οργανισμού, τα μέλη της Red Team πρέπει να έχουν δεξιότητες κωδικοποίησης για να δημιουργήσουν το δικό τους exploit και να το προσαρμόσουν ώστε να εκμεταλλεύονται καλύτερα τις σχετικές ευπάθειες που θα μπορούσαν να επηρεάσουν τον οργανισμό. Με αυτό τον τρόπο εντοπίζονται οι αδυναμίες στην ασφάλεια του συστήματος και είναι ευκολότερο να προστατευθούν τα περιουσιακά στοιχεία του οργανισμού

2.1.2 Βήματα της Επίθεσης

Για την επίτευξη μιας επίθεσης ακολουθείται η πιο κάτω διαδικασία βημάτων.

2.1.2.1 Information Gathering / Αναγνώριση

Το Information Gathering είναι μια διαδικασία συλλογής πληροφοριών του εκάστοτε στόχου. Ο στόχος μπορεί να είναι μια ιστοσελίδα, ένα web application, ένα πρόσωπο και γενικά οτιδήποτε. Συνήθως είναι το πρώτο στάδιο που θα ακολουθήσει η red team. Στόχος είναι ο εντοπισμός και η καταγραφή όσο το δυνατόν περισσότερων πληροφοριών και αυτό μπορεί να γίνει είτε με ενεργό είτε με παθητικό τρόπο. Στον παθητικό τρόπο παίρνεις τις πληροφορίες από το διαδίκτυο, κυρίως με τις μηχανές αναζήτησης και τα social media και εδώ θα αποφασιστεί εάν θα προχωρήσει ή ακυρωθεί μια επίθεση. Στον ενεργό τρόπο αντλείς πληροφορίες απευθείας από τον στόχο αλληλεπιδρώντας μαζί του, κάτι που περιγράφεται στο επόμενο βήμα.

2.1.2.2 Scanning & Enumeration

Σε αυτό το βήμα αρχίζουμε την αλληλεπίδραση με τον στόχο. Με την χρήση εργαλείων όπως το nmap, το nessus ή το nikto προσπαθούμε να εντοπίσουμε πληροφορίες όπως ανοιχτά ports, usernames, services που χρησιμοποιούνται αλλά και γνωστές ευπάθειες στο σύστημα. Τα

στοιχεία που συλλέγονται θα αξιοποιηθούν για να προσφέρουν μη εξουσιοδοτημένα δεδομένα ή πρόσβασ

2.1.2.3 Gaining Access (explotation)

Όταν ολοκληρωθούν τα προηγούμενα βήματα και αφού συλλέχθηκαν όσο το δυνατόν περισσότερες πληροφορίες αφορούν τον στόχο, θα ακολουθήσει η απόπειρα για μη εξουσιοδοτημένη πρόσβαση. Με τον εντοπισμό γνωστών ευπαθειών θα χρησημοποιήσουμε εργαλεία, οπώς το metasploit, που θα εκτελέσουν scripts για να γίνει exploit το σύστημα. Άλλη τεχνική είναι το brute force που μπορούμε να επιχειρήσουμε επίθεση σε κάποιο ανοιχτό open port (π.χ. ssh) ώστε να αποκτήσουμε τους κωδικούς εισόδου. Μία καλά εκπαιδευμένη red team πρέπει να έχει την δυνατότητα να γράφει δικά της scripts και να προσαρμόζει τις επιθέσεις στις δικές της ανάγκες.

2.1.2.4 Maintaining Access

Όταν παραβιαστεί ο στόχος, ο επιτιθέμενος πετυχαίνει μια προσωρινή πρόσβαση στο σύστημα. Αν για κάποιο λόγο διακοπεί η πρόσβαση ή το κενό ασφαλείας διορθωθεί ίσως να μην υπάρχει η δυνατότητα επανασύνδεσης και να πρέπει να αρχίσει η αναζήτηση για ευάλωτο σημείο στο σύστημα από την αρχή. Γι’ αυτό με το που θα πραγματοποιηθεί η πρώτη σύνδεση ο εισβολέας πρέπει να κάνει τις απαραίτητες κινήσεις ώστε η πρόσβαση να διατηρηθεί ή να είναι διαθέσιμη και μελλοντικά. Για να πετύχει μια πιο μόνιμη πρόσβαση απαιτείται η δημιουργία ενός backdoor που θα ξεκινούσε μαζί με το σύστημα. Σε αντίθετη περίπτωση η συνεχής αναζήτηση ευπαθειών στο σύστημα θα μπορούσε να ειδοποιήσει τον admin του συστήματος.

2.1.2.5 Covering Tracks

Η διαδικασία της επίθεσης δεν τελειώνει όταν επιτευχθεί η πρόσβαση. Χρειάζεται να καλυφθούν τα ίχνη που δημιουργήθηκαν ώστε να αποφευχθεί κάποια έρευνα που θα οδηγούσε στον επιτιθέμενο. Κατά την προσπάθεια της επίθεσης έχουν καταγραφεί αρκετά logs από την φάση του σκαναρίσματος αλλά και από τυχόν απόπειρες σπασίματος των κωδικών. Είναι σημαντικό να διαγραφούν ή να τροποποιηθούν όλα τα log files που αφορούν την επίθεση. Επίσης αν υπάρχει ένα κακόβουλο λογισμικό στο σύστημα που χρησιμοποιείται ως backdoor, πρέπει να είναι όσο το δυνατόν πιο διακριτικό στο σύστημα για να μην αφήνει υποψίες. Θα μπορούσαν να χρησιμοποιηθούν τεχνικές στεγανογραφίας ώστε το κακόβουλο λογισμικό να έχει την μορφή ενός άλλου αρχείου (π.χ. εικόνα) όπου θα δυσκόλευε αρκετά τον εντοπισμό του.

2.2 Ορολογίες

2.2.1 Penetration Testing

Το Penetration Testing [2] ορίζεται ως μια νόμιμη και εξουσιοδοτημένη προσπάθεια εντοπισμού και εκμετάλλευσης ενός πληροφοριακού συστήματος με στόχο το συγκεκριμένο σύστημα να γίνει πιο ασφαλές. Η διαδικασία περιλαμβάνει την διερεύνηση ευπαθειών καθώς και επιθέσεις ώστε να αποδεικνύει ότι τα τρωτά σημεία του συστήματος είναι πραγματικά. Στο τέλος γίνονται συγκεκριμένες συστάσεις για την αντιμετώπιση και διόρθωση των προβλημάτων που ανακαλύφθηκαν κατα τη διάρκεια της δοκιμής. Συνολικά, αυτή η διαδικασία χρησιμοποιείται για να βοηθήσει στην ασφάλεια των υπολογιστών και των δικτύων από μελλοντικές επιθέσεις. Η γενική ιδέα είναι να βρεθούν ζητήματα ασφάλειας χρησιμοποιώντας τα ίδια εργαλεία και τεχνικές με έναν επιτιθέμενο και να αντιμετωπιστούν πριν τα εκμεταλλευτεί ένας πραγματικός χάκερ.

2.2.2 Blue Team

Η blue team πρέπει να διασφαλίζει ότι τα περιουσιακά στοιχεία της εταιρείας είναι ασφαλή. Στην περίπτωση που η red team βρει μια ευπάθεια και την εκμεταλλευτεί, η blue team πρέπει να την επανορθώσει άμεσα. Τα μέλη της blue team πρέπει να έχουν αρκετές δεξιότητες και έχουν την ευθύνη για την ασφάλεια του συστήματος. Το πόσο ασφαλές είναι ένα σύστημα δεν είναι ποτέ γνωστό καθώς δεν μπορείς να ξέρεις πότε η red team θα το θέσει σε κίνδυνο. Γι’ αυτό πρέπει να ενισχύουν συνεχώς την ασφάλεια των ψηφιακών υποδομών.

2.2.3 Bot

Ένα Internet bot είναι μια εφαρμογή λογισμικού που εκτελεί αυτοματοποιημένα scripts μέσω του Διαδικτύου. Συνήθως, τα ρομπότ εκτελούν εργασίες που είναι απλές και επαναλαμβανόμενες πολύ πιο γρήγορα από ό,τι ένας άνθρωπος. Τα κακόβουλα bots συνήθως είναι ένας συνδυασμός Trojan απομακρυσμένης πρόσβασης και Worm, όπου μπορεί να εξαπλώνεται γρήγορα και παρέχει την δυνατοτητα σε έναν εισβολέα για απομακρυσμένη πρόσβαση [3].

2.2.4 Script-kiddies

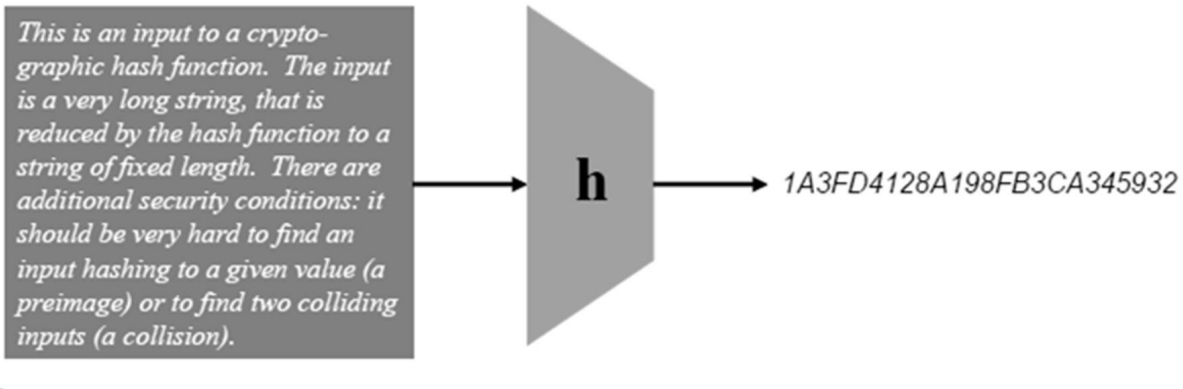
Script kiddie είναι ένα σχετικά ανειδίκευτο άτομο που χρησιμοποιεί προγράμματα και scripts, που έχουν δημιουργηθεί από άλλους, για να επιτεθεί σε συστήματα και δίκτυα υπολογιστών. Συνήθως τα άτομα που θεωρούνται script kiddies δεν έχουν την ικανότητα να γράφουν δικά τους εξελιγμένα προγράμματα ή να μπορούν να εκμεταλευτούν μια ευπάθεια και ο στόχος τους είναι να εντυπωσιάσουν τους φίλους τους. Στον κυβερνοχώρο ο όρος αυτός θεωρείται υποτιμητικός.

2.2.5 Brute Force

Το Brute Force είναι μια επίθεση που χρησιμοποιεί την μέθοδο trial and error για το σπάσιμο κωδικών, συνθηματικών σύνδεσης και κρυπτογραφημένων κλειδιών. Είναι μια πολύ απλή αλλά αξιόπιστη τακτική για την απόκτηση μη εξουσιοδοτημένης πρόσβασης σε λογαριασμού και συστήματα οργανισμών. Στην επίθεση δοκιμάζονται πολλά ονόματα χρηστών και συνθηματικών μέχρι να βρεθεί ο σωστός συνδυασμός που θα περιέχει τις πληροφορίες σύνδεσης.

2.2.6 Hash Functions

Τα Hash Functions (Συναρτήσεις κατατεμαχισμού) έχουν ένα είδος «κρυπτογραφικής» λειτουργίας όπου κάθε είσοδος μετασχηματίζεται σε μία σχετικά μικρότερη έξοδο, μοναδική για κάθε μήνυμα. Μπορούν να δεχτούν ως είσοδο ένα μήνυμα οποιουδήποτε μεγέθους και επιστρέφουν μία έξοδο σταθερού μήκους. Η ανάκτηση του αρχικού μηνύματος από το hash του είναι πρακτικά ανέφικτη [4].



Εικόνα 1. Παράδειγμα μηνύματος εισόδου και εξόδου μετά την συνάρτηση κατατεμαχισμού

2.2 Παρόμοιες Έρευνες

Συχνά τονίζεται η ανάγκη των red teams για την προετοιμασία της ομάδας ασφάλειας ενός συστήματος για τυχόν επιθέσεις. Σε αντίθεση με τους penetration tester που επικεντρώνονται στον εντοπισμό ευπαθειών, οι ομάδες red team αξιολογούν ολόκληρο το δίκτυο εκτελώντας πραγματικές επιθέσεις. Το πρόβλημα της ύπαρξης μιας red team, σε μόνιμη βάση, στις εταιρείες, παρουσιάζεται στο υψηλό κόστος, την επανάληψη της διαδικασίας και την συνεχή εκπαίδευση που απαιτείται στα άτομα που αποτελούν την ομάδα. Συμφονα με τους Applebaum, Andy et al [5] δημιουργήθηκε το framework CALDERA όπου δημιουργεί ένα προσομοιωτή αυτοματοποιημένων επιθέσεων. Τα άτομα που καλούνται να προστατεύσουν ένα σύστημα ή δίκτυο δεν μπορούν να γνωρίζουν ποτέ αν τους έχει διαφύγει κάτι [6]. Με την ύπαρξη μιας red team, που λειτουργεί αυτόματα μέσω του framework, η ομάδα ασφάλειας έχει την δυνατότητα να ερευνήσει αμυντικές τεχνολογίες που μπορούν να χρησιμοποιηθούν μελλοντικά για την καταπολέμηση μιας πραγματικής επίθεσης [7].

Στην έρευνα των Enoch et al [8] παρατηρήθηκε ότι η ποιότητα της ασφάλειας εξαρτάται άμεσα από την ποιότητα των μελών της ομάδας red team. Γι’ αυτό δημιουργήθηκε ένα αυτοματοποιημένο framework που θα εκτελεί κυβερνοεπιθέσεις, χωρίς να επηρεάζει τις ήδη υπάρχουσες διαδικασίες που ακολουθούνται από τις ομάδες red team. Στα πλαίσια της έρευνας έχει προταθεί ένας βελτιωμένος αλγόριθμος εντοπισμού attack path, μαζί με τις απαιτήσεις και τις φάσεις για το πλαίσιο αυτοματισμού, αλλά και ο σχεδιασμός επιθέσεων βάσει μετρήσεων ασφάλειας. Πραγματοποιήθηκαν πειράματα σε πραγματικό δίκτυο και έχει παρατηρηθεί ότι τα αυτόματα tools που εντοπίζουν μεμονωμένες ευπάθειες δυσκολεύονται να εντοπίσουν συνδυασμό ευπαθειών, ενώ αυτά που μπορούν να εντοπίσουν συνδυασμό ευπαθειών είναι δύσκολο να εφαρμοστούν σε μεγάλα δίκτυα.

Οι Holm και Sommestad προσπάθεισαν να διαπιστώσουν αν οι ανησυχίες που προκύπτουν από διάφορους οργανισμούς όσο αφορά την αυτοματοποίηση των επιθετικών εργαλείων, αυξάνουν τον κίνδυνο από scpipt-kiddies [9]. Η μελέτη βασίστηκε στον βαθμό δυσκολίας που χρειάζεται κάποιος για να επιτύχει μια κυβερνοεπίθεση. Τα αποτελέσματα της έρευνας ήταν ενθαρρυντικά καθώς έδειξαν ότι η αυτοματοποίηση red attack εργαλείων δεν αυξάνει τον κίνδυνο ασφάλειας από script-kiddies και ότι είναι σχεδόν απίθανο να επιτύχει μια server-side επίθεση.

Ο κυβερνοχώρος γίνεται όλο και πιο κρίσιμος σε πολλές κοινωνικές, εμπορικές, και στρατιωτικές λειτουργίες καθώς χρόνο με τον χρόνο αναδεικνύεται από μόνος του ως ένας επιχειρησιακός χώρος. Σε αυτό το πλαίσιο, οι υπεύθυνοι λήψης αποφάσεων πρέπει να επιτύχουν την σωστή λειτουργία των δραστηριοτήτων μέσα στον κυβερνοχώρο. Μια μέθοδος που χρησιμοποιείται για τον έλεγχο της ασφάλειας είναι το penetration testing. Οι τεχνικές που χρησιμοποιούνται δεν διαφέρουν από αυτές του επιτιθέμενου και η αποτελεσματική δοκιμή του penetration testing μας προσφέρει μια εικόνα της ασφάλειας του συστήματος. Συχνά όμως οι ευπάθειες αξιολογούνται μεμονωμένα ως προς την προτεραιότητα αποκατάστασης και αρκετές παραμένουν απλά στην αναφορά αποτελεσμάτων. Οι Randwaha, Turnbull, Yuen και Dean παρουσίασαν το Trogdor, ένα αυτοματοποιημένο σύστημα red teaming, που αναλαμβάνει να αναλύσει τους κρίσιμους κόμβους βάσει μοντέλου για να παρουσιάσει οπτικά τον αντίκτυπο των ευάλωτων πόρων [10]. Συγκεκριμένα, αυτή η έρευνα πραγματεύεται τον σκοπό του Trogdor, πως παρέχει κατανόηση των πιθανών επιπτώσεων που προκύπτουν από ευπάθειες στον κυβερνοχώρο και πως επιλέγει πιθανές στρατηγικές για την αποφυγή αυτών των επιπτώσεων.

Στις μέρες μας η ασφάλεια των πληροφοριακών συστημάτων είναι πολύ σημαντική καθώς όλο και περισσότερα δεδομένα αποθηκεύονται σε συστήματα υπολογιστών που είναι συνδεδεμένα στο διαδίκτυο. Αυτό δημιουργεί νέες προκλήσεις και πρέπει να

εξασφαλιστεί ότι τα συστήματα θα είναι όσο το δυνατόν πιο ασφαλή. Ένας τρόπος να ελέγξουμε την ασφάλεια των συστημάτων είναι με την διενέργεια συχνών penetration testing. Οι Holik και Horalek [11] παρουσίασαν τα βασικά στάδια του penetration testing και πως χρησιμοποιείτε για αυτό τον σκοπό το metasploit framework. Καταλήγει εξηγώντας την σημαντικότητα του να ξοδευτεί χρόνος και χρήμα στην εκμάθηση αυτών των τεχνικών καθώς το κόστος της απώλειας ή αλλοίωσης δεδομένων είναι πολύ μεγαλύτερο.

2.2.1 Θετικά / Αρνητικά

Με βάση τις υπάρχουσες έρευνες που μελετήθηκαν στα αρνητικά παρατηρείται ότι το κόστος για την διενέργεια penetration testing και ο εντοπισμός ευπαθειών στο σύστημα είναι πολύ μεγάλο από την στιγμή που αυτή η διαδικασία είναι επαναλαμβανόμενη. Η ποιότητα της ασφάλειας έχει άμεση σχέση με την ποιότητα των μελών της red team και ξοδεύεται χρόνος και χρήμα στην εκμάθηση τους. Επίσης τα αυτόματα tools που εντοπίζουν μεμονωμένες ευπάθειες δυσκολεύονται να εντοπίσουν συνδυασμό ευπαθειών ενώ αυτά που μπορούν να εντοπίσουν συνδυασμό ευπαθειών είναι δύσκολο να εφαρμοστούν σε μεγάλα δίκτυα.Στα θετικά είναι ότι η αυτοματοποίηση επιθέσεων βοηθάει στην ανάπτυξη αμυντικών τεχνικών που μπορούν να χρησιμοποιηθούν μελλοντικά για την καταπολέμηση μιας πραγματικής επίθεσης. Επίσης μπορούν να χρησιμοποιηθούν για να αναλυθούν κρίσιμοι κόμβοι βάσει μοντέλου και να παρουσιαστεί οπτικά ο αντίκτυπος των ευάλωτων πόρων. Το ότι η δημιουργία αυτοματοποιημένων tools δεν αυξάνει τον κίνδυνο ασφάλειας από script-kiddies δεν θέτει αμφιβολίες για την ανάπτυξη τους. Ένα άλλο σημαντικό στοιχείο είναι ότι όσο μεγάλο και να είναι το κόστος για penetration testing και εκπαίδευση της red team, το κόστος της απώλειας ή αλλοίωσης δεδομένων παραμένει πολύ μεγαλύτερο.

2.3 Frameworks

2.3.1 Metasploit

Το metasploit είναι ένα εργαλείο που χρησιμοποιείται για τον εντοπισμό ευπαθειών σε δίκτυα και πληροφοριακά συστήματα και χρησιμοποιείται από κυβερνοεγκληματίες και ηθικούς χάκερς. Είναι ανοιχτού κώδικα, συμβατό με διαφορετικά λειτουργικά συστήματα και εύκολα προσαρμόσιμο στις διαφορετικές απαιτήσεις που προκύπτουν. Η red team που θα κάνει το penetration testing μπορεί εύκολα να χρησιμοποιήσει το metasploit για να εντοπίσει τα αδύναμα σημεία του δικτύου. Αφού εντοπιστούν οι απειλές και τα αδύναμα σημεία, μπορεί κανείς να τα τεκμηριώσει και να αντιμετωπίσει τις αδυναμίες του δικτύου ή του συστήματος και να προχωρήσει στον εντοπισμό μιας λύσης για αυτό [12].

2.3.2 Armitage

Το Armitage είναι ένα γραφικό περιβάλλον εργασίας για το Metasploit. Στόχος του είναι να βοηθήσει τους επαγγελματίες ασφαλείας να κατανοήσουν καλύτερα το hacking και να τους βοηθήσει να συνειδητοποιήσουν τη δύναμη και τις δυνατότητες του Metasploit. Μπορεί το Armitage να φαίνεται σαν ένα όμορφο front-end του Metasploit αλλά στην πραγματικότητα είναι ένα εργαλείο συνεργασίας των ατόμων της red team. Διαθέτει ένα server που επιτρέπει σε μια ομάδα χάκερ να μοιράζονται τις προσβάσεις τους σε παραβιασμένους κεντρικούς υπολογιστές. Υπάρχει επίσης η δυνατότητα να γραφτούν bots σε scripts που συνδέονται στον server της ομάδας και επεκτείνεται στο Armitage [13][14].

2.3.3 Nessus

Το Nessus είναι ένας σαρωτής ευπαθειών δικτύου ανοιχτού κώδικα που χρησιμοποιείται κατά τις αξιολογήσεις ευπαθειών στο penetration

testing. Είναι ένα εργαλείο που ελέγχει τους υπολογιστές για να βρει τρωτά σημεία που μπορούν να εκμεταλλευτούν οι χάκερ. Λειτουργεί δοκιμάζοντας κάθε θύρα σε έναν υπολογιστή, προσδιορίζοντας ποια υπηρεσία εκτελείται και στη συνέχεια δοκιμάζει αυτήν την υπηρεσία για να βεβαιωθεί ότι δεν υπάρχουν ευπάθειες σε αυτήν που θα μπορούσαν να χρησιμοποιηθούν από έναν χάκερ για να πραγματοποιήσει μια κακόβουλη επίθεση [15].

Κεφάλαιο 3
Μεθοδολογία

3.1 Σκοπός

Σε αυτό το κεφάλαιο περιγράφεται η μεθοδολογία που ακολουθήθηκε για την υλοποίηση της διατριβής. Αναλύονται οι τύποι ερευνών και ποιος τύπος είναι κατάλληλος στο υπάρχον ερευνητικό πρόβλημα. Στην συνέχεια μελετήθηκαν διάφορα μοντέλα κύκλου ζωής λογισμικού και εξηγήθηκε πιο είναι κατάλληλο για την ανάπτυξη του εργαλείου που απαιτείτε για τους σκοπούς της συγκεκριμένης μελέτης.

3.2 Τύπος Έρευνας

3.2.1 Ποιοτική έρευνα

Η ποιοτική έρευνα χρησιμοποιείται συνήθως σε περιπτώσεις όπου τα ερευνητικά ερωτήματα είναι γενικά, διευρυμένα και έχουν να κάνουν με σκέψεις και συναισθήματα. Οι απαντήσεις είναι κυρίως σε μορφή κειμένου και η ανάλυση των δεδομένων περιλαμβάνει την ανάλυση κειμένων. Το υλικό που συγκεντρώνεται προέρχεται από συνεντεύξεις, συνομιλίες, φωτογραφίες, μαγνητοφωνήσεις και σημειώσεις σε ημερολόγια. Αυτό σημαίνει ότι η ποιοτική έρευνα μελετά τα πράγματα στο φυσικό τους πλαίσιο, επιχειρώντας να δώσει νόημα ή να ερμηνεύσει τα φαινόμενα [16][17].

3.2.2 Ποσοτική Έρευνα

Η ποσοτική έρευνα χρησιμοποιείται συνήθως σε περιπτώσεις που τα ερευνητικά ερωτήματα είναι συγκεκριμένα και περιορισμένα. Τα στοιχεία που συλλέγονται μπορούν να εκφραστούν ποσοτικά και η ανάλυση των στοιχείων γίνεται αναλύοντας τους αριθμούς με μεθόδους ανάλυσης αριθμητικών δεδομένων. Η συλλογή δεδομένων γίνεται με δομημένα πρωτόκολλα, όπως ερωτηματολόγια, κλίμακες και δοκίμια επιτευγμάτων. Η ερευνητική ομάδα μπορεί να θέσει τις μεταβλητές που επιθυμεί, να συγκεντρώσει ποσοτικά δεδομένα και να καταλήξει στην εισήγηση της για την αξιοποίηση ή μη αξιοποίηση κάποιου χαρακτηριστικού [18].

3.2.3 Περιγραφή της μελέτης

Θα πραγματοποιηθεί Ποσοτική μελέτη για την μεθοδολογία της διατριβής. Επιλέχθηκε αυτός ο τρόπος επειδή έχει αποφασιστεί τι θα μελετηθεί, έχουν τεθεί συγκεκριμένα ερωτήματα μικρού εύρους και τα δεδομένα που θα συγκεντρωθούν μπορούν να εκφραστούν ποσοτικά, αναλύοντας τους αριθμούς με τη χρήση της στατιστικής. Ο σκοπός της παρούσας μελέτης είναι η υλοποίηση ενός αυτοματοποιημένου εργαλείου που θα αποφέρει οφέλη όπως την γρηγορότερη και πληρέστερη ανάλυση και εντοπισμό ευπαθειών στο σύστημα, καλύπτοντας το κενό άλλων εργαλείων που υστερούν σε ευχρηστία, απλότητα, και αυτοματοποίηση. Αφού μελετήθηκε η βιβλιογραφία διαπιστώθηκε ότι το κόστος για την διενέργεια penetration testing και ο εντοπισμός ευπαθειών στο σύστημα είναι πολύ μεγάλο και η ποιότητα της ασφάλειας έχει άμεση σχέση με την ποιότητα των μελών της red team. Η βιβλιογραφία αιτιολογεί την ανάγκη για τη μελέτη γρηγορότερων και λιγότερο δαπανηρών τρόπων για τον εντοπισμό ευπαθειών στα πληροφοριακά συστήματα. Γι’ αυτό το σκοπό δημιουργήθηκε ένα εργαλείο που έχει ως στόχο να βοηθήσει στην γρηγορότερη λήψη αποφάσεων, στον εντοπισμό των ευπαθειών στο σύστημα καθώς και την μείωση του κόστους στην διενέργεια αυτής της διαδικασίας. Το εργαλείο που αναπτύχθηκε είναι ένα bash script που

αυτοματοποιεί διάφορες γνωστές πρακτικές του penetration testing. Η μελέτη απευθύνεται σε εταιρείες, κυβερνήσεις και άτομα που δραστηριοποιούνται στον χώρο της ασφάλειας πληροφοριακών συστημάτων.

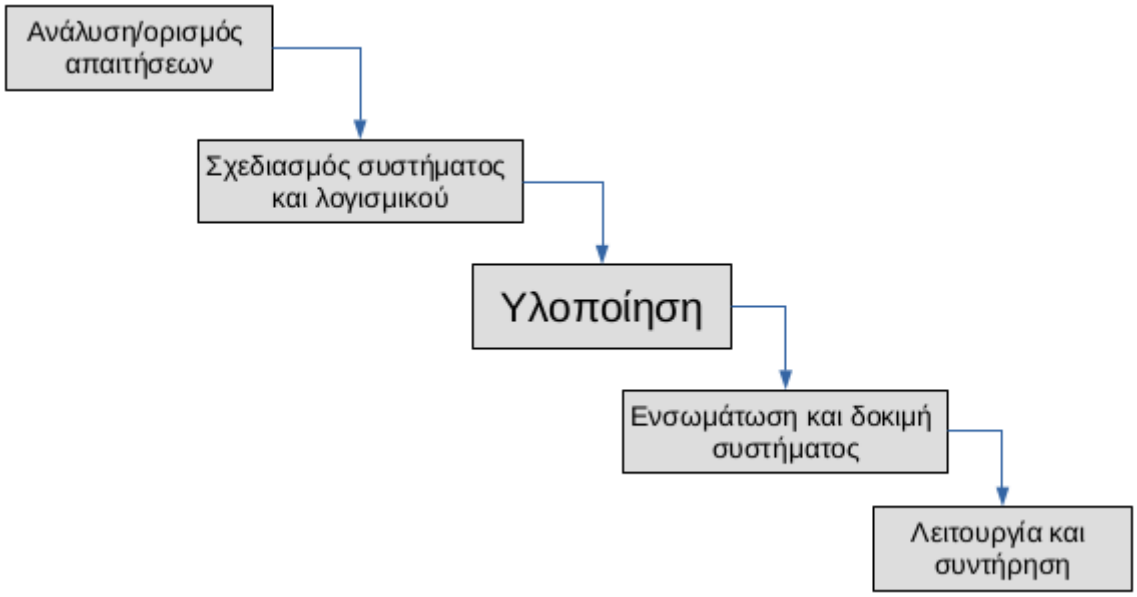
3.3 Μοντέλα Κύκλου Ζωής Λογισμικού

Πολλές φορές η ανάπτυξη ενός λογισμικού είναι πολύπλοκη. Γι’ αυτό μια προσέγγιση με φάσεις ανάπτυξης είναι απαραίτητη για τον έλεγχο των έργων. Δίνουν στους developers ένα χάρτη για το πως να προχωρήσουν με την ανάπτυξη του λογισμικού. Επίσης μπορούν ευκολότερα να προβλέψουν που μπορεί να προκύψουν αδυναμίες στην διαδικασία ανάπτυξης και έτσι να τις αντιμετωπίσουν πριν αυτές συμβούν. Οι προσεγγίσεις διαφέρουν αλλά συνήθως περιλαμβάνουν τις προδιαγραφές, την σχεδίαση, την υλοποίηση, την επικύρωση και την εξέλιξη. Στις προδιαγραφές αναλύεται τι πρέπει να κάνει το σύστημα. Στην σχεδίαση και υλοποίηση καθορίζεται η οργάνωση του συστήματος και πως θα αναπτυχθεί. Στην επικύρωση ελέγχεται αν το σύστημα κάνει αυτά που θέλει ο πελάτης και αν λειτουργεί σωστά. Τέλος στην εξέλιξη γίνονται αλλαγές ανάλογα με τις νέες ανάγκες που μπορεί να προκύψουν από τον πελάτη [19].

3.3.1 Μοντέλο καταρράκτη

Το μοντέλο καταρράκτη έχει σειριακή προσέγγιση. Αυτό σημαίνει ότι για να προχωρήσεις στο επόμενο βήμα πρέπει πρώτα να ολοκληρωθεί το προηγούμενο. Το κάθε βήμα βασίζεται στη δημιουργία προδιαγραφών και η κάθε φάση είναι ξεκάθαρη. Αυτές οι φάσεις είναι η ανάλυση και ορισμός απαιτήσεων, ο σχεδιασμός συστήματος και λογισμικού, η υλοποίηση, η ενσωμάτωση και δοκιμή του συστήματος και η λειτουργία και συντήρηση. Τα πλεονεκτήματα αυτού του μοντέλου είναι ότι οι προδιαγραφές για το σύστημα δημιουργούνται εξ αρχής καθώς και η

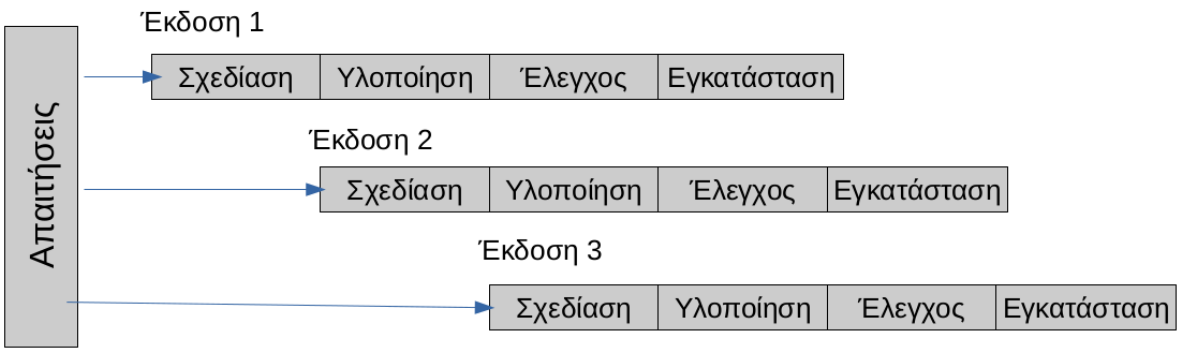
συντήρηση διευκολύνεται. Στα μειονεκτήματα είναι ότι οι προδιαγραφές δεν μπορούν να αλλάξουν στη πορεία και ο πελάτης συμμετέχει μόνο στην αρχή βλέποντας το προϊόν πολύ αργά στη διάρκεια της διαδικασίας. Συνήθως χρησιμοποιείται στην ανάπτυξη μεγάλων έργων λογισμικού, όπου ένα σύστημα αναπτύσσεται σε διάφορες τοποθεσίες και οι απαιτήσεις του συστήματος δεν αλλάζουν [20].



Εικόνα 2. Σχεδιάγραμμα Μοντέλου καταρράκτη

3.3.2 Αυξητικό μοντέλο

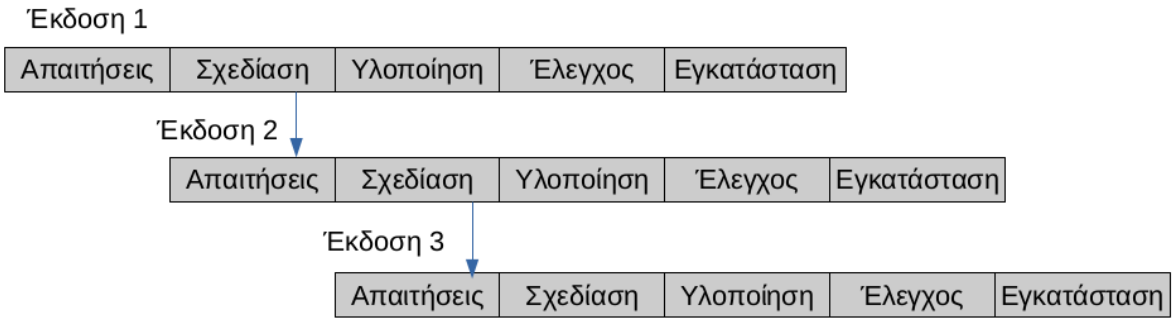
Το αυξητικό μοντέλο σε αντίθεση με το μοντέλο καταρράκτη δεν ακολουθεί μια σειριακή προσέγγιση αλλά μια εξελικτική ανάπτυξη. Δηλαδή το σύστημα λογισμικού παραδίδεται βαθμιαία και ο πελάτης δεν χρειάζεται να περιμένει μέχρι να παραδοθεί όλο το σύστημα για να ξεκινήσει να το χρησιμοποιεί. Αυτό σημαίνει ότι γίνεται επιμερισμός της συνολικής εργασίας σε μικρότερες και κάθε έκδοση επαυξάνει την προηγούμενη με νέες λειτουργίες και χαρακτηριστικά. Η κάθε έκδοση που παραδίδεται είναι λειτουργική αλλά υπάρχει μικρή ευελιξία στις αλλαγές των απαιτήσεων [21].



Εικόνα 3. Σχεδιάγραμμα Αυξητικού μοντέλου

3.3.3 Εξελικτικό/Επαναληπτικό μοντέλο

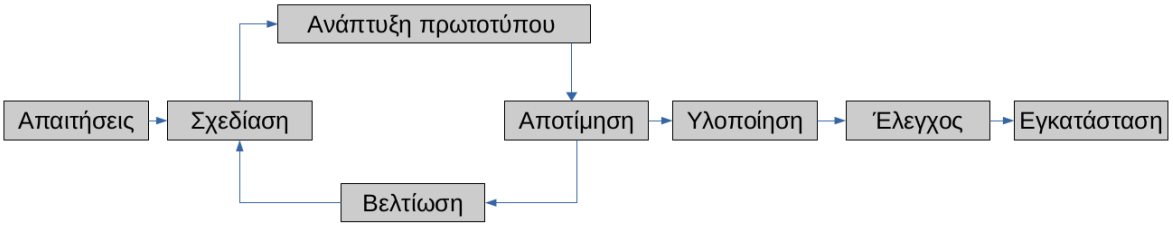
Σε αυτο το μοντέλο γίνονται επαναλαμβανόμενοι κύκλοι σε μικρά χρονικά διαστήματα και νέες εκδόσεις υλοποιούν νέες απαιτήσεις που εξελίσσονται όσο το σύστημα υλοποιείται. Όπως και στο Αυξητικό μοντέλο έτσι και σε αυτό παραδίδονται πολλές εκδόσεις στον πελάτη. Η κάθε έκδοση όμως υλοποιεί νέες απαιτήσεις που προκύπτουν με την συνεχή συμμετοχή του πελάτη. Η συνεχής συμμετοχή του πελάτη στην ανάπτυξη του λογισμικού είναι στα θετικά όμως μπορεί να υπάρξει ο κίνδυνος η ανάπτυξη να ξεκινάει συνεχώς από την αρχή.



Εικόνα 4. Σχεδιάγραμμα Εξελικτικού/Επαναληπτικού μοντέλο

3.3.4 Μοντέλο Πρωτοτυποποίησης

Το μοντέλο πρωτοτυποποίησης παρουσιάζει στον πελάτη ένα πρωτότυπο του συστήματος σε λειτουργία. Αυτό το πρωτότυπο δεν είναι το προϊόν γι’ αυτό πρέπει να αναπτύσσεται γρήγορα και χωρίς μεγάλο κόστος. Δεν υλοποιεί όλες τις λειτουργίες του συστήματος και δεν χρειάζεται να έχει ποιότητα. Αυτό το μοντέλο είναι χρήσιμο στην ανάπτυξη συστημάτων που έχουν χρηστικές διεπαφές και όχι συστήματα που έχουν απαιτητικούς υπολογισμούς. Τα πλεονεκτήματα του μοντέλου πρωτοτυποποίησης είναι ότι το τελικό σύστημα είναι πιο εύκολο στη χρήση, γίνεται καλύτερη ενσωμάτωση των αναγκών των χρηστών και τα προβλήματα εντοπίζονται νωρίτερα. Στα μειονεκτήματα είναι ότι το πρωτότυπο δεν χρησιμοποιείται όπως το τελικό σύστημα και η απόδοση του τελικού συστήματος είναι πολύ χειρότερη. Αυτό μπορεί να προκαλέσει σύγχυση σε χρήστες και προγραμματιστές.



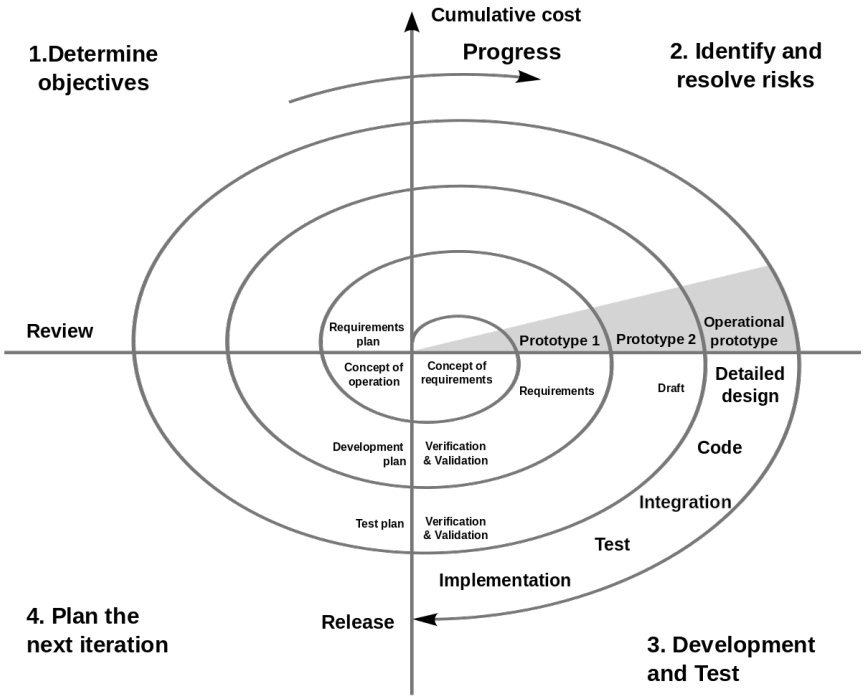
Εικόνα 5. Σχεδιάγραμμα Μοντέλου Πρωτοτυποποίησης

3.3.5 Σπειροειδές μοντέλο

Το σπειροειδές μοντέλο έχει στοιχεία από άλλα μοντέλα ανάπτυξης λογισμικού όπως μοντέλο καταρράκτη και μοντέλο πρωτοτυποποίησης. Αυτό το μοντέλο βασίζεται σε πρότυπα κινδύνου που διασφαλίζει την αποτελεσματική διαδικασία ανάπτυξης. Κάθε φάση του ξεκινά με την ανάλυση ρίσκου και τον σχεδιασμό και τελειώνει με τον πελάτη να δίνει feedback και τον προγραμματισμό της επόμενης φάσης. Γι’ αυτό καθορίζονται στην αρχή ένα μικρό σύνολο των απαιτήσεων και περνάμε

κάθε φάση της ανάπτυξης με αυτές τις απαιτήσεις. Η ομάδα ανάπτυξης του λογισμικού προσθέτει επιπλέον απαιτήσεις σε κάθε νέο κύκλο μέχρι το σύστημα να ολοκληρωθεί. Στα θετικά του σπειροειδούς μοντέλου είναι ότι μπορεί να καθοριστεί τι πρέπει να ελεγχθεί και δεν υπάρχει διαχωρισμός μεταξύ υλοποίησης και συντήρησης. Στα αρνητικά όμως είναι ότι δεν είναι κατάλληλο για μικρά έργα και αν ένας μεγάλος κίνδυνος δεν εντοπιστεί και δεν τον διαχειριστούμε θα προκύψουν προβλήματα. [22]

Img [https://en.wikipedia.org/wiki/Spiral_model]



Εικόνα 6. Σχεδιάγραμμα Σπειροειδές μοντέλου

3.3.6 Μοντέλο Κύκλου Ζωής Της Παρούσας Διατριβής

Για την ανάπτυξη του εργαλείου «ARTAS – Automatic Red Team Attack Scriptss» χρησιμοποιήθηκε το Εξελικτικό μοντέλο καθώς ανά χρονικά διαστήματα νέες εκδόσεις υλοποιούνταν και νέες απαιτήσεις δημιουργούνταν. Λόγω του περιορισμένου χρονικού περιθωρίου που

υπήρχε στα πλαίσια της διατριβής θεωρήθηκε σημαντικό να υπάρχει λειτουργική έκδοση του εργαλείου και στην πορεία να προστίθενται νέες λειτουργίες. Με αυτό τον τρόπο υπάρχει δυνατότητα ώστε η ανάπτυξη του εργαλείου να συνεχιστή εύκολα σε μελλοντικό στάδιο.

Κεφάλαιο 4
Υλοποίηση

4.1 Σκοπός

Σε αυτό το κεφαλαίο θα περιγραφούν τα εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για τηνυλοποίηση του αυτοματοποιημένου προγράμματος. Στη συνέχεια θα μελετηθουν οι διαφοροι τυποι επιθέσεων που μπορουν να χρησιμοποιηθουν και τέλος θα παρουσιαστούν κάποια από τα σημαντικότερα σενάρια υλοποίησης του συστήματος.

4.2 Εργαλεία Ανάπτυξης

4.2.1 Kali Linux

Το Kali Linux είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο Debian. Απευθύνεται κυρίως σε άτομα που ασχολούνται με penetration testing και θέματα ασφάλειας. Το Kali Linux παρέχει αρκετές εκατοντάδες εργαλεία προεγκατεστημένα στο λογισμικό που μπορούν να χρησιμοποιηθούν σε διάφορες εργασίες ασφάλειας πληροφοριακών συστημάτων. Γι’ αυτό το λόγο επιλέχθηκε η χρήση αυτού του λογισμικού κατά την διάρκεια της υλοποίησης και χρήσης του εργαλείου «ARTAS» καθώς πολλά από τα εργαλεία που έχουν αυτοματοποιηθεί είναι ήδη προεγκατεστημένα ή υπάρχουν στο repository της διανομής [23].

4.2.2 Bash

Αρκετά άτομα που δραστηριοποιούνται στον χώρο της ασφάλειας πληροφοριακών συστημάτων χρησιμοποιούν το Terminal ώστε να δίνουν εντολές στο σύστημα και να βλέπουν τα αποτελέσματα. Το Bash είναι το shell ή η command γλώσσα που χρησιμοποιεί το λειτουργικό σύστημα GNU. Όταν κάποιος ανοίξει ένα παράθυρο terminal συνδέεται αυτόματα σε ένα shell. Οι command γλώσσες είναι αυτές που ελέγχουν τις διάφορες εργασίες σε έναν υπολογιστή και συνήθως έχουν ισχυρότερη επικοινωνία με το λειτουργικό σύστημα. Αυτές οι γλώσσες δεν διαφέρουν ιδιαίτερα στη σύνταξη από άλλες γλώσσες προγραμματισμού. Για την συγγραφή του κώδικα του αυτοματοποιημένου εργαλείου «ARTAS» χρησιμοποιήθηκε το bash ώστε να είναι συμβατό άμεσα σε όλες τις διανομές Unix αλλά και να αυτοματοποιήσει αρκετές διεργασίες που ήδη γίνονταν μέσα από το terminal. Με αυτό τον τρόπο το εργαλείο «ARTAS» δεν χρειάζεται εξειδικευμένη εγκατάσταση στο σύστημα [24].

4.2.3 Scripts

4.2.3.1 Nmap

Το Nmap (Network Mapper) είναι ένα ανοιχτού κώδικα πρόγραμμα για αναζήτηση δικτύου και έλεγχου ασφαλείας. Πολλά συστήματα και διαχειριστές δικτύου το χρησιμοποιούν για εργασίες όπως την παρακολούθηση του χρόνου λειτουργίας ενός host, το uptime μιας υπηρεσίας κλπ. Το Nmap χρησιμοποιεί ακατέργαστα πακέτα IP με διάφορους τρόπους ώστε να προσδιορίσει ποιοι hosts είναι διαθέσιμοι στο δίκτυο, ποιες υπηρεσίες προσφέρουν (όνομα εφαρμογής και έκδοση), ποια λειτουργικά συστήματα και τι έκδοση τρέχουν, ποιο firewall χρησιμοποιείτε και άλλα πολλά Αναγνωρίζεται σήμερα ως ένα από τα καλύτερα εργαλεία στον τομέα αυτό. Έχει σχεδιαστή για γρήγορη σάρωση μεγάλων δικτύων, αλλά λειτουργεί καλά και σε μονούς host. Είναι διαθέσιμο για εγκατάσταση σε Linux, Windows και Mac OS X [25].

4.2.3.2 WPScan

Το WPScan είναι ένα εργαλείο γραμμένο στην γλώσσα προγραμματισμού Ruby. Στόχος του είναι να κάνει ένα έλεγχο ασφαλείας σε ιστοσελίδες που χρησιμοποιούν το WordPress ως σύστημα διαχείρισης περιεχομένου. Με την χρήση του μπορεί να εντοπιστεί η έκδοση του WordPress, των plugins και των themes που είναι εγκατεστημένα και να παρουσιάσει τυχόν γνωστές ευπάθειες. Επίσης μπορεί να εντοπιστεί ο κωδικός από χρήστες με αδύναμο συνθηματικό και άλλα πολλά [26].

4.2.3.3 Nikto

Το Nikto είναι ένα ανοιχτού κώδικα web server scanner που εκτελεί γρήγορους ελέγχους ασφαλείας σε servers. Το Nikto δεν είναι σχεδιασμένο για να κάνει τους ελέγχους αθόρυβα. Κάνει το σκανάρισμα στον γρηγορότερο δυνατό χρόνο και αυτό εντοπίζεται εύκολα στα log files. Ο κάθε έλεγχος που γίνεται δεν έχει απαραίτητα στόχο κάποιο πρόβλημα ασφάλειας, αλλά παρουσιάζονται πληροφορίες που εντοπίζονται στον server που ίσως φανούν χρήσιμες σε ένα Penetration Tester. Ο στόχος του εργαλείου είναι να εντοπίσει πιθανά προβλήματα ασφαλείας και ευπάθειες στον server [27].

4.2.3.4 Gobuster

Το Gobuster είναι ένα εργαλείο που χρησιμοποιείται για brute force σε URLs μιας ιστοσελίδας. Χρησιμεύει στο να εντοπίζονται αρχεία και directories που είναι κρυμμένα και μπορούν να φανούν χρήσιμα για ένα Penetration Tester. Το Gobuster είναι γραμμένο στην γλώσσα Go και τρέχει στην γραμμή εντολών. Αυτό έχει ως αποτέλεσμα να είναι γρηγορότερο από άλλες γλώσσες προγραμματισμού ή scripting γλώσσες όπως είναι η Java ή η Python [28].

4.2.3.5 GoldenEye

Το GoldenEye είναι ένα εργαλείο γραμμένο στην γλώσσα Python και χρησιμοποιείτε για υλοποίηση επιθέσεων άρνησης υπηρεσίας (denial of service attack, DoS). Αυτή η επίθεση γίνεται ενάντια σε ένα υπολογιστή ή μια υπηρεσία με σκοπό να μην μπορούν να δεχτούν άλλες συνδέσεις και έτσι να μην μπορεί να εξυπηρετήσει άλλους πιθανούς πελάτες. Το GoldenEye έχει την δυνατότητα να ξεκινάει παράλληλες συνδέσεις ενάντια σε ένα URL Μπορεί να χρησιμοποιηθεί για να ελιχθεί εάν ένας web server μπορεί να αντεπεξέλθει σε τέτοιου είδους επιθέσεις [29].

4.2.3.5 Hashcat

Το Hashcat είναι ένα εργαλείο που σχεδιάστηκε για να σπάει κωδικούς σε μορφή hash. Για να το πετύχει αυτο, δίνουμε στο εργαλείο ένα λεξικό με διάφορους πιθανούς κωδικούς, τα μετατρέπει σε μορφή hash και στην συνέχεια ελέγχει αν ταιριάζουν τα δύο hashes μαζί [30].

4.2.4 Βοηθητικά εργαλεία ανάπτυξης

4.2.4.1 Sublime

Το Sublime είναι ένας επεξεργαστής κειμένου που υποστηρίζει διάφορα plug-ins και έχει ενσωματωμένο το Git Control. Το Sublime ήταν το εργαλείο συγγραφής κώδικα που χρησιμοποιήθηκε για την υλοποίηση του εργαλείου «ARTAS».

4.2.4.2 GitLab

Το GitLab είναι μια πλατφόρμα που προσφέρει την δυνατότητα χρήσης του Git ως Server as a service. Το Git είναι μια εφαρμογή που καταγράφει όλες τις αλλαγές στα αρχεία με πλήρες ιστορικό και δυνατότητες πλήρους παρακολούθησης της έκδοσης, ανεξάρτητα από την πρόσβαση στο δίκτυο. Συνήθως χρησιμοποιείται όταν πολλοί προγραμματιστές

δουλεύουν στο ίδιο project. Το GitLab ήταν η υπηρεσία που χρησιμοποιήθηκε ως αποθετήριο για τον κώδικα του εργαλείου «ARTAS».

4.2.4.3 Vulnhub

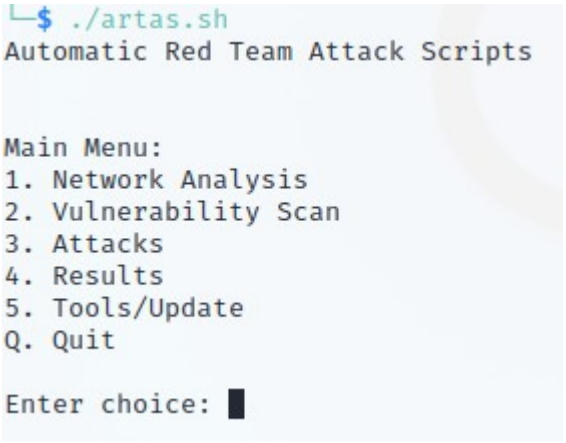
Το Vulnhub.com είναι μια ιστοσελίδα που προσφέρει υλικό σε όποιον επιθυμεί να εξασκηθεί ή να πειραματιστή στον τομέα της ασφάλειας υπολογιστών και δικτύων. Έχει ένα τεράστιο κατάλογο από ευάλωτες μηχανές που προσφέρουν την δυνατότητα να διευρύνεις τις γνώσεις σου. Οι μηχανές που προσφέρονται στο Vulnhub χρησιμοποιήθηκαν για σκοπούς δοκιμής του εργαλείου «ARTAS» σε ένα ασφαλές περιβάλλον.

4.3 Σενάρια υλοποίησης

Σε αυτή την υποενότητα θα παρουσιαστούν οι διαφορές δυνατότητες του εργαλείου «ARTAS». Είναι χωρισμένες σε εφτά κατηγορίες. Η πρώτη κατηγορία έχει να κάνει με το πώς αρχίζει να τρέχει το εργαλείο, οι επόμενες τρεις κατηγορίες περιγράφουν τις διαφορές επιθέσεις που μπορούν να πραγματοποιηθούν και οι τελευταίες κατηγορίες έχουν να κάνουν με δυνατότητες που παρέχονται για την ευκολότερη χρήση του εργαλείου.

4.3.1 Αρχικό Μενού

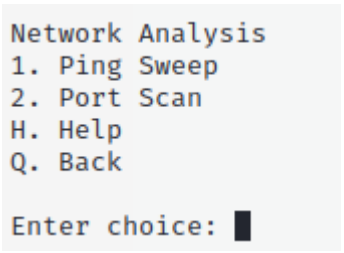
Το εργαλείο «ARTAS» είναι ένα Bash Script και γι’ αυτό τρέχει μέσα από το Terminal. Όταν το ξεκινήσουμε θα εμφανιστή το Αρχικό Μενού όπου υπάρχουν 6 επιλογές. Οι επιλογές είναι: Network Analysis, Vulnerability Scan, Attacks, Results, Tools/Update και Quit. Θα αναλυθεί η κάθε μία ξεχωριστά στην συνέχεια. Με την επιλογή Quit το εργαλείο τερματίζει την λειτουργία του.



Εικόνα 7. Το Αρχικό μενού στο εργαλείο «ARTAS»

4.3.2 Network Analysis

Όταν ο χρήστης επιλέξει την επιλογή Network Analysis θα εμφανιστή ένα νέο μενού με τις επιλογές Ping Sweep και Port Scan. Σε αυτή την κατηγορία βρίσκονται οι επιλογές για σάρωση δικτύου και σάρωση συγκεκριμένου host.



Εικόνα 8. Το Network Analysis μενού στο εργαλείο «ARTAS»

4.3.2.1 Ping Sweep

Το Ping Sweep είναι μια μέθοδος που σαρώνει ένα ολόκληρο δίκτυο για να εντοπίσει hosts που είναι σε σύνδεση Όταν ο χρήστης διαλέξει την επιλογή του Ping Sweep στο εργαλείο καλείτε να δώσει την διεύθυνση που θέλει να σαρώσει και το Subnet Mask. Για το Subnet Mask έχει τρεις επιλογές 8, 16 και 24 με την τελευταία να είναι η προκαθορισμένη. Για την πραγμάτωση αυτής της δυνατότητας στο εργαλείο «ARTAS» αυτοματοποιήθηκε μέρος από τις δυνατότητες του εργαλείου Nmap.

```
Enter choice: 1
Target IP [Default=localhost]:10.10.118.0

Subnet Mask:
1) 8
2) 16
3) 24 [Default]

Enter choice:
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-26 17:38 EST
Nmap scan report for 10.10.118.135
Host is up (0.077s latency).
Nmap scan report for 10.10.118.193
Host is up (0.074s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 64.76 seconds
```

Εικόνα 9. Στην εικόνα βλέπουμε ότι εντοπίστηκαν δυο hosts που βρίσκονται online

4.3.2.2 Port Scan

Το Port Scan χρησιμοποιείτε για να διερευνηθεί ένας server ή host και να εντοπιστούν ανοιχτά ports. Για των εντοπισμό των ανοιχτών ports η διαδικασία που ακολουθείτε είναι να στέλνονται πακέτα στον server/host και ανάλογα με την απάντηση που θα σταλεί πίσω να γνωρίζουμε αν μια υπηρεσία είναι διαθέσιμη ή όχι. Τα είδη των πακέτων που μπορούν να σταλούν αλλά και ο τρόπος αποστολής τους μπορεί να διαφέρει κάθε φορά ανάλογά με τα αποτελέσματα που θέλουμε να δούμε. Στο εργαλείο «ARTAS» αυτοματοποιήθηκε ένα μεγάλο μέρος από τις

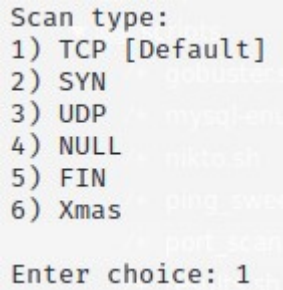
δυνατότητες του εργαλείου Nmap όσο αφορά το Port Scanning. Στις υποενότητες που ακολουθούν θα αναλυθούν όλες οι δυνατές επιλογές που μπορεί να επιλέξει ένας χρήστης.

```
Enter choice: 2
Target URL:10.10.118.135
Scan type:
1) TCP [Default]
2) SYN
3) UDP
4) NULL
5) FIN
6) Xmas

Enter choice: 1

Flags:
0) None [Default]
1) Treat all hosts as online
2) Aggresive Scan
3) Enable OS detection
4) Version detetion
5) Scan ports from 1 through 65535
6) Firewall Evasion: Generate invalid checksum for packets
7) Firewall Evasion: Fragment the packets into smaller pieces

Enter choice: 0

Time (from slower to faster):
1) Paranoid
2) Sneaky
3) Polite
4) Normal [Default]
5) Aggressive
6) Insane

Enter choice: 4
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-26 17:41 EST
Nmap scan report for 10.10.118.135
Host is up (0.094s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 82.38 seconds
```

Εικόνα 10. Στην εικόνα έγινε port scan στην διεύθυνση 10.10.118.135. Η μέθοδος αποστολής των πακέτων που επιλέγηκε ήταν το TCP connect scan, χωρίς επιπλέων flag και με φυσιολογικό ρυθμό αποστολής. Τα αποτελέσματα της σάρωσης έδειξαν ότι στην συγκεκριμένη διεύθυνση υπάρχει ανοιχτό το port 80 με την υπηρεσία HTTP.

4.3.2.2.1 Scan Type

Όταν θέλουμε να κάνουμε port scanning υπάρχουν τρεις βασική τύποι. Αυτοί είναι το TCP, SYN και UDP. Επιπλέων υπάρχουν και τύποι όχι τόσο συνηθισμένοι όπως το NULL, FIN και Xmas. Οι περισσότεροι τύποι χρησιμοποιούνται για πολύ παρόμοιους σκοπούς, όμως ο τρόπος με τον οποίο λειτουργούν διαφέρει στον καθένα. Πιο κάτω θα γίνει μια πιο αναλυτική περιγραφή για των κάθε τύπο ξεχωριστά.



Εικόνα 11. Το μενού με τις επιλογές για Scan Type στο εργαλείο «ARTAS»

4.3.2.2.1.1 TCP Connect Scans

Το TCP Connect Scan λειτουργεί εκτελώντας ένα ολοκληρωμένο three-way handshake για κάθε port στόχο. Στην ουσία προσπαθεί να συνδεθεί σε κάθε port και να καθορίσει αν αυτό είναι ανοιχτό. Στην περίπτωση όπου ένα port είναι κλειστό με το που σταλεί το πρώτο SYN πακέτο επιστρέφεται ένα πακέτο RST. Έτσι αναγνωρίζει ότι το port είναι κλειστό.

4.3.2.2.1.2 SYN "Half-open" Scans

Όπως και στο TCP scan έτσι και στο SYN Scan ελέγχεται το κάθε port ξεχωριστά. Ωστόσο η διαφορά είναι ότι ένω το TCP Scan κάνει ένα ολόκληρο three-way handshake στον στόχο, το SYN scan στέλνει ένα RST πακέτο όταν πάρει SYN/ACK πακέτο ως απάντηση από τον Server. Γι’ αυτό και το SYN Scan ονομάζεται και Half-open επειδή δεν ολοκληρώνει το three-way handshake αλλά μόλις αντιληφθεί ότι είναι ανοιχτό το port το σταματά. Στην περίπτωση όπου ένα port είναι κλειστό όπως και προηγούμενος ο server επιστρέφει ένα πακέτο RST.

4.3.2.2.1.3 UDP Scans

Σε αντίθεση με το TCP, το UDP στέλνει πακέτα στο port στόχο ελπίζοντας ότι θα φτάσει. Όταν ένα πακέτο UDP φτάσει σε ένα port κανονικά δεν θα πρέπει να υπάρχει ανταπόκρηση. Όταν συμβαίνει αυτό το συγκεκριμένο port καταγράφεται ως ανοιχτό ή φιλτραρισμένο. Με λίγα λόγια το port μάλλον είναι ανοιχτό εκτός αν υπάρχει κάποιο firewall που απορρίπτει τα εισερχόμενα πακέτα. Όταν ένα port είναι κλειστό, ο στόχος απαντάει με ένα ICMP πακέτο με μήνυμα ότι το port είναι απρόσιτο. Με αυτό τον τρόπο γνωρίζουμε ότι το port είναι κλειστό.

4.3.2.2.1.4 TCP Null Scans

Το NULL scan είναι όταν ένα TCP πακέτο στέλνεται χωρείς καθόλου flags. Με αυτό τον τρόπο ο στόχος θα απαντήσει με ένα RST πακέτο και έτσι θα γνωρίζουμε ότι το port είναι κλειστό.

4.3.2.2.1.5 TCP FIN Scans

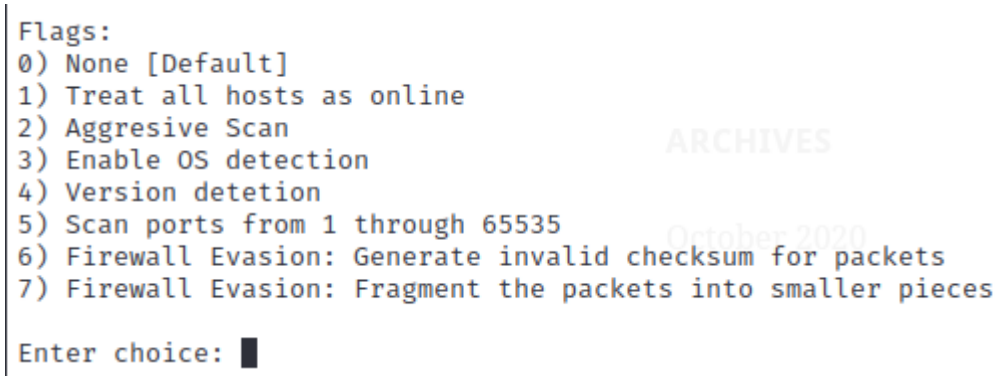
Το FIN scan λειτουργεί σχεδόν με τον ίδιο τρόπο όπως και το NULL Scan. Η διαφορά είναι ότι αντί να στέλνεται ένα άδειο πακέτο, στέλνεται ένα πακέτο FIN. Πάλι όπως και πριν αν ο στόχος απαντήσει με ένα RST πακέτο τότε το port είναι κλειστό.

4.3.2.2.1.6 TCP Xmas Scans

Όπως και με τα προηγούμενα δυο scans, το Xmas scan στέλνει ένα πακέτο στο port στόχο, με τα flags PSH, URG και FIN, και αν πάρει απάντηση ένα RST πακέτο τότε το port είναι κλειστό. Η χρησιμότητα των NULL, FIN και Xmas scan βρίσκεται στο ότι πολλά firewalls απορρίπτουν τα TCP πακέτα που περιέχουν το flag SYN και με αυτόν τον τόπο κάποιες φορές μπορούμε να αποφύγουμε το μπλοκάρισμα από τον server.

4.3.2.2.2 Flags

Τα Flags έχουν σαν στόχο να προσθέσουν δυνατότητες και να κάνουν το port scanning πιο ευέλικτο. Στην συνέχεια περιγράφονται αναλυτικά οι επιλογές που προσφέρονται στο εργαλείο «ARTAS».



Εικόνα 12. Το μενού με τις επιλογές για Flags στο εργαλείο «ARTAS»

4.3.2.2.2.1 None

Η πρώτη επιλογή είναι να μην υπάρξει κανένα flag. Αυτή είναι και η προκαθορισμένη επιλογή καθώς είναι συνηθισμένο όταν γίνεται ένα port scan για πρώτη φορά να είναι προτιμότερο να γίνει στα γρήγορα χωρίς επιπλέον παραμέτρους.

4.3.2.2.2.2 Treat all hosts as online

Συνήθως οι host απαντάνε όταν τους σταλεί ένα πακέτο για να ενημερώσουν ότι είναι online. Το Nmap στέλνει αυτό το πακέτο και περιμένει απάντηση. Εάν δεν την πάρει θεωρεί τον host ότι δεν είναι σε σύνδεση και δεν προχωράει στο Port Scanning. Επειδή πολλοί hosts δεν απαντάνε σε τέτοια πακέτα, προτείνεται αυτή η επιλογή ώστε να μην δοθεί σημασία εάν ο host είναι σε σύνδεση και να γίνει απευθείας το port scanning.

4.3.2.2.2.3 Aggresive Scan

Αυτή η επιλογή προσφέρει πολύ περισσότερες πληροφορίες από το συνηθισμένο scan. Ενεργοποιεί την δυνατότητα για OS detection, Version detection, Script scanning και tracerout. Αυτές οι δυνατότητες μπορούν να επιλεγούν και ανεξάρτητα.

4.3.2.2.2.4 OS detection

Το OS detection προσπαθεί να αναγνώριση το λειτουργικό σύστημα που τρέχει στον server στόχο. Αυτό επιτυγχάνεται με το να στέλνονται διάφορα πακέτα και να συλλέγονται οι απαντήσεις. Από τις απαντήσεις συλλέγονται κάποια fingerprints και ελέγχονται σε μια βάση δεδομένων. Εάν ταιριάζουν τότε μπορούμε να αναγνωρίσουμε το λειτουργικό σύστημα.

4.3.2.2.2.5 Version detetion

Το Version detection διερευνά τα ανοιχτά ports που έχουν βρεθεί και καθορίζει τι υπηρεσία και πια έκδοση τρέχει.

4.3.2.2.2.6 Scan ports from 1 through 65535

Το απλό scanning γίνεται μεταξύ 1 και 1023 port. Εάν θέλουμε να ψάξουμε σε όλα τα πιθανά ports που μπορεί μια υπηρεσία να χρησιμοποιήσει διαλέγουμε αυτή την επιλογή.

4.3.2.2.2.7 Firewall Evasion: Generate invalid checksum for packets

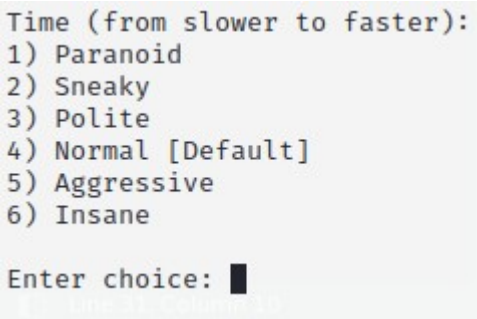
Αυτή η επιλογή δημιουργεί ένα μη έγκυρο ckecksum στο πακέτο που αποστέλλεται. Οποιαδήποτε πραγματική TCP/IP στοίβα έπαιρνε αυτό το πακέτο θα το απέρριπτε αμέσως. Στην περίπτωση όμως που υπάρχεί firewall είναι πολύ πιθανών να πάρουμε κάποια αυτοματοποιημένη απάντηση καθώς δεν θα ελεγχθεί καν η εγκυρότητα του checksum στο πακέτο. Με αυτό τον τρόπο μπορούμε να διαπιστώσουμε ότι στον στόχο υπάρχει παρουσία κάποιου firewall.

4.3.2.2.2.8 Firewall Evasion: Fragment the packets into smaller piece

Με αυτή την επιλογή το κάθε πακέτο που αποστέλλεται κατακερματίζεται σε μικρότερα κομμάτια. Στόχος αυτής της ενέργειας είναι να μειωθεί η πιθανότητα ένας firewall να αναγνωρίσει τα πακέτα ως κακόβουλα.

4.3.2.2.3 Time

Σε αυτή την επιλογή διαλέγουμε τον ρυθμό αποστολής των πακέτων στο στόχο. Υπάρχουν έξι δυνατότητες όπου ο χρόνος είναι πάρα πολύ αργός μέχρι πάρα πολύ γρήγορος. Ο έλεγχος του ρυθμού αποστολείς των πακέτων έχει σαν στόχο να βελτιώσει την ποιότητα του scanning και να αυξήσει την πιθανότητα να έχουμε τα επιθυμητά αποτελέσματα.



Εικόνα 13. Το μενού με τις επιλογές για Time στο εργαλείο «ARTAS»

4.3.2.2.3.1 Insane

Αυτή η επιλογή στέλνει ένα πακέτο κάθε 5 χιλιοστά του δευτερολέπτου. Η αποστολή γίνεται υπερβολικά γρήγορα και περιμένει μόνο 0.3 δευτερόλεπτα για κάθε απάντηση. Με αυτό το τρόπο το σκανάρισμα γίνεται ταχύτατα αλλά θυσιάζεται η ακρίβεια. Εάν το σκανάρισμα δεν ολοκληρωθεί μέσα σε 15 λεπτά διακόπτετε αυτόματα. Αυτή η επιλογή πρέπει να χρησιμοποιείτε μόνο σε πολύ γρήγορα δίκτυα διαφορετικά μπορεί να επηρεάσει την λειτουργιά του δικτύου.

4.3.2.2.3.2 Aggressive

Αυτή η επιλογή στέλνει ένα πακέτο κάθε 10 χιλιοστά του δευτερολέπτου. Η αποστολή γίνεται αρκετά γρήγορα και περιμένει μόνο 1.25 δευτερόλεπτα για κάθε απάντηση. Συστήνεται για χρήση σε αξιόπιστα δίκτυα.

4.3.2.2.3.3 Normal

Αυτή είναι η προεπιλογή στην ταχύτητα όπου ο ακριβής χρόνος δεν είναι προσδιορισμένος.

4.3.2.2.3.4 Polite

Αυτό χρησιμοποιείτε για να στέλνει πακέτα γρηγορότερα από τις δυο επόμενες επιλογές αλλά παραμένει ακόμα πιο αργό από το Normal. Στέλνει ένα πακέτο κάθε 0.4 δευτερόλεπτα.

4.3.2.2.3.5 Sneaky

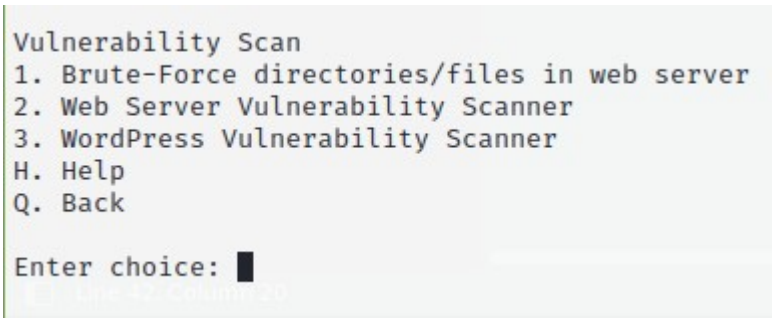
Αυτή η επιλογή στέλνει τα πακέτα αρκετά αργά. Η διαφορά στο χρόνο αποστολής ανάμεσα σε κάθε πακέτο είναι 15 δευτερόλεπτα.

4.3.2.2.3.6 Paranoid

Αυτή η επιλογή χρησιμοποιείτε για να στέλνονται τα πακέτα υπερβολικά αργά και μόνο ένα port σαρώνετε κάθε φόρα. Η διαφορά στο χρόνο αποστολής ανάμεσα σε κάθε πακέτο είναι 5 λεπτά. Ο λόγος να χρησιμοποιήσει κάποιος μια τόσο αργή σάρωση είναι για να αποφύγει τα πιθανά firewalls που έχει ο στόχος.

4.3.3 Vulnerability Scan

Το Vulnerability Scan είναι μια αυτόματη διαδικασία που εντοπίζει στα συστήματα ευπάθειες ασφαλείας. Στο εργαλείο «ARTAS» υπάρχουν 3 επιλογές. Η μια είναι για Brute Force σε καταλόγους και αρχεία ενός server, και οι άλλες δυο για εύρεση ευπαθειών σε ιστοσελίδες.



Εικόνα 14. Οι πιθανές επιλογές του Vulnerability Scan στο εργαλείο «ARTAS»

4.3.3.1 Brute-Force directories/files in web server

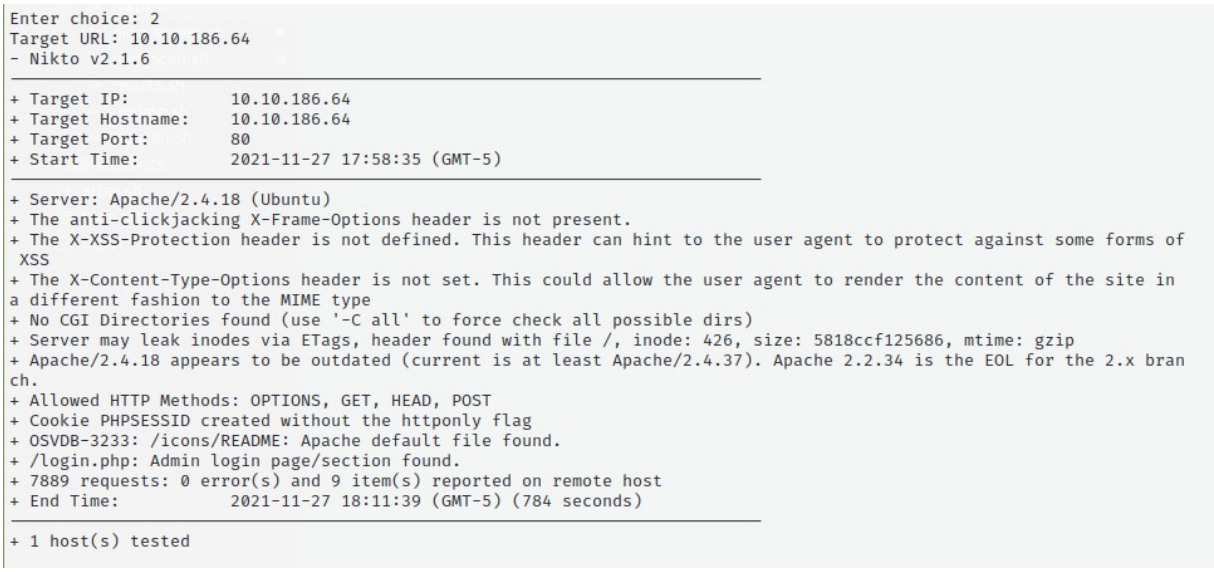
Η επιλογή αυτή κάνει brute force σε URLs μιας ιστοσελίδας. Χρησιμεύει στο να εντοπίζονται αρχεία και directories που είναι κριμένα. Για την εκπόνηση αυτής της λειτουργίας χρησιμοποιήθηκε το εργαλείο Gobuster. Αρχικά το εργαλείο «ARTAS» ζητάει να δοθεί η διεύθυνση του στόχου. Στην συνέχεια υπάρχουν τρεις επιλογές για λεξικό (wordlist) που είναι: η κοινή (common) με τα πιο συνηθισμένα, η μικρή (short) και η μεγάλη (big). Μετά από αυτό υπάρχει η επιλογή αν θέλουμε να γίνει αναζήτηση και για αρχεία εκτός από διευθύνσεις. Σε αυτή την επιλογή γίνεται αναζήτηση για αρχείο που έχουν κατάληξη σε html, txt και php.

```
Enter choice: 1
Target URI: http://10.10.186.64
Wordlist: [C]ommon (default), [S]mall, [B]ig:c
File extensions: [Y]es, [N]o (default)y
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.186.64
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                ./files/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              html,txt,php
[+] Timeout:                 10s
===============================================================
2021/11/27 17:25:43 Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 291]
/.hta.txt            (Status: 403) [Size: 295]
/.hta.php            (Status: 403) [Size: 295]
/.hta.html           (Status: 403) [Size: 296]
/.htaccess.txt       (Status: 403) [Size: 300]
/.htpasswd           (Status: 403) [Size: 296]
/.htaccess.php       (Status: 403) [Size: 300]
/.htpasswd.txt       (Status: 403) [Size: 300]
/.htaccess.html      (Status: 403) [Size: 301]
/.htpasswd.php       (Status: 403) [Size: 300]
/.htaccess           (Status: 403) [Size: 296]
/.htpasswd.html      (Status: 403) [Size: 301]
/assets              (Status: 301) [Size: 313] [--> http://10.10.186.64/assets/]
/denied.php          (Status: 302) [Size: 0] [--> /login.php]
/index.html          (Status: 200) [Size: 1062]
/index.html          (Status: 200) [Size: 1062]
/login.php           (Status: 200) [Size: 882]
/portal.php          (Status: 302) [Size: 0] [--> /login.php]
/robots.txt          (Status: 200) [Size: 17]
/robots.txt          (Status: 200) [Size: 17]
/server-status       (Status: 403) [Size: 300]

===============================================================
2021/11/27 17:28:27 Finished
===============================================================
```

Εικόνα 15. Στην εικόνα βλέπουμε ότι η διεύθυνση στόχος είναι το 10.10.186.64. Έχει επιλεγεί το λεξικό common και να γίνει η αναζητήσει μαζί με τα αρχεία. Στο τέλος φαίνονται τα αποτελέσματα της επίθεσης.

4.3.3.2 Web Server Vulnerability Scanner

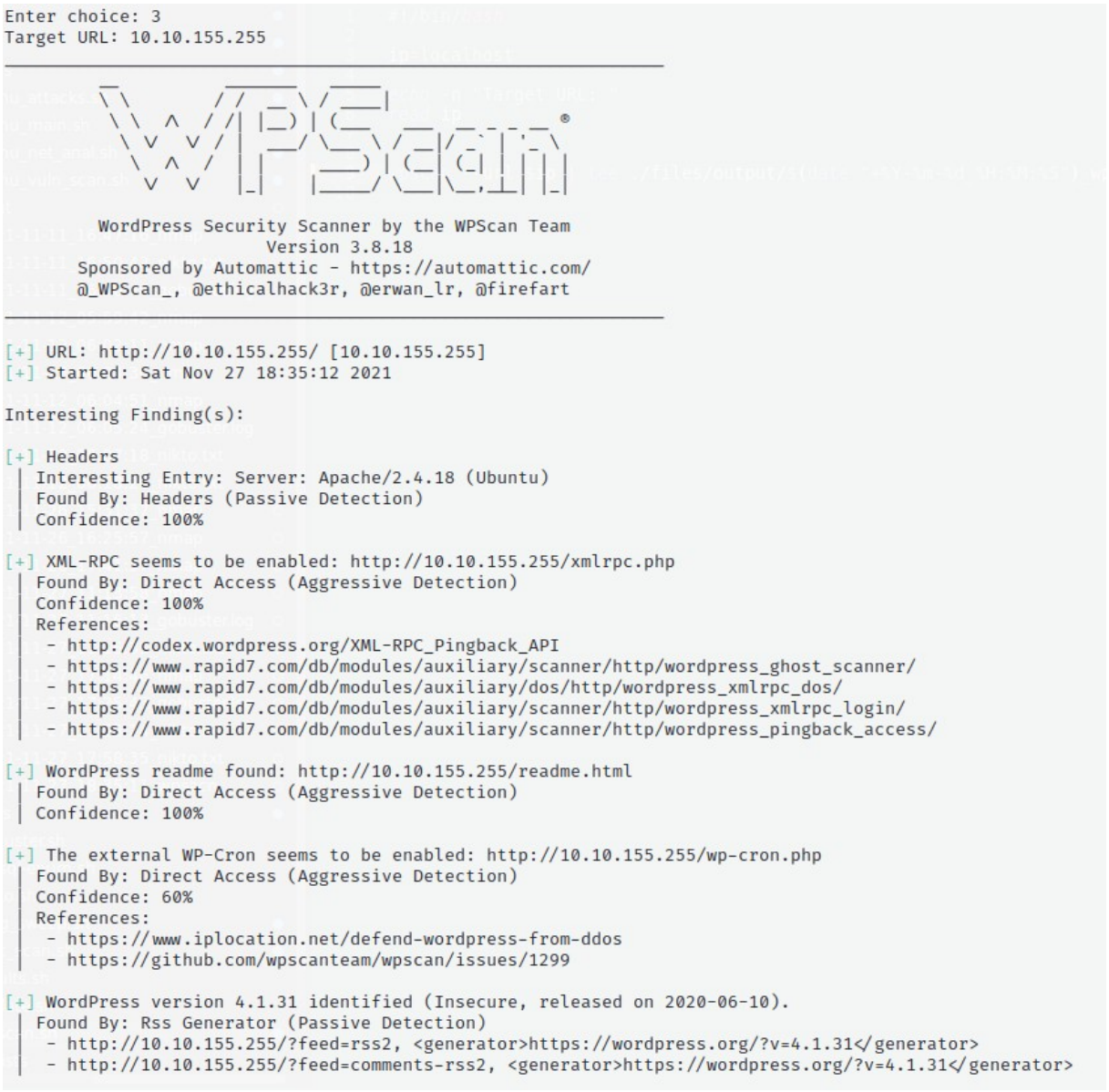
Σε αυτή την επιλογή χρησιμοποιείτε το εργαλείο Nikto. Στην αρχή ζητείτε η διεύθυνση του στόχου και στην συνέχεια γίνεται ένας γρήγορος έλεγχος στον server. Τα επιθυμητά αποτελέσματα είναι να εντοπιστούν πιθανές ευπάθειες ή πληροφορίες του συστήματος.



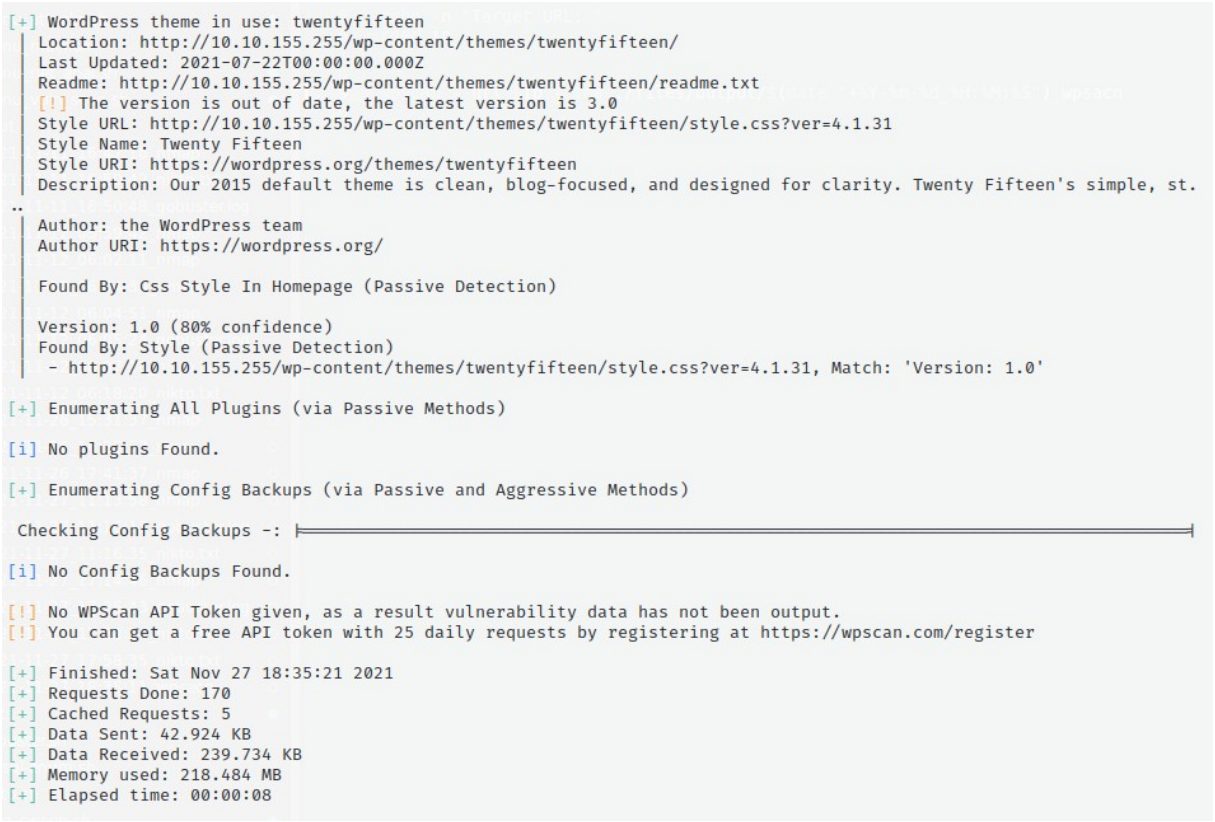
Εικόνα 16. Στην εικόνα βλέπουμε πως λειτουργεί η επιλογή Web Server Vulnerability Scanner. Η διεύθυνση στόχος είναι η 10.10.186.64 και φαίνονται τα αποτελέσματα της επίθεσης.

4.3.3.3 WordPress Vulnerability Scanner

Σε αυτή την επιλογή χρησιμοποιείτε το εργαλείο WPScan. Το εργαλείο «ARTAS» ζητάει την διεύθυνση του στόχου και στην συνέχεια κάνει ένα έλεγχο ασφαλείας σε ιστοσελίδες που χρησιμοποιούν το WordPress ως σύστημα διαχείρισης περιεχομένου. Με την χρήση του μπορεί να εντοπιστεί η έκδοση του WordPress, των plugins και των themes που είναι εγκατεστημένα και να παρουσιάσει τυχόν γνωστές ευπάθειες. Επίσης μπορεί να εντοπιστεί ο κωδικός από χρήστες με αδύναμο συνθηματικό και άλλα πολλά


Enter choice: 3
Target URL: 10.10.155.255
WordPress Security Scanner by the WPScan Team
Version 3.8.18
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
[+] URL: http://10.10.155.255/ [10.10.155.255]
[+] Started: Sat Nov 27 18:35:12 2021
Interesting Finding(s):
[+] Headers
| Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
| Found By: Headers (Passive Detection)
| Confidence: 100%
[+] XML-RPC seems to be enabled: http://10.10.155.255/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
[+] WordPress readme found: http://10.10.155.255/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] The external WP-Cron seems to be enabled: http://10.10.155.255/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
[+] WordPress version 4.1.31 identified (Insecure, released on 2020-06-10).
| Found By: Rss Generator (Passive Detection)
| - http://10.10.155.255/?feed=rss2, <generator>https://wordpress.org/?v=4.1.31</generator>
| - http://10.10.155.255/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.1.31</generator>

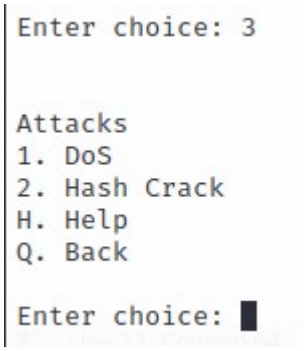
Εικόνα 17



Εικόνα 18. Στην εικόνα 17 και 18 βλέπουμε πως λειτουργεί η επιλογή WordPress Vulnerability Scanner. Η διεύθυνση στόχος είναι η 10.10.155.255 και φαίνονται τα αποτελέσματα της επίθεσης.

4.3.4 Attacks

Στο Attacks μενού υπάρχουν τα εργαλεία που κάνουν κάποιο είδος επίθεσης. Αυτή την στιγμή το εργαλείο «ARTAS» έχει δυο επιλογές. Η πρώτη είναι το Denial of service (DoS) που στοχεύει web servers και η δεύτερη είναι το Hash Crack που προσπαθεί να σπάσει το hash που θα του δώσει ο χρήστης.



Εικόνα 19. Το μενού Attacks στο εργαλείο «ARTAS»

4.3.4.1 DoS

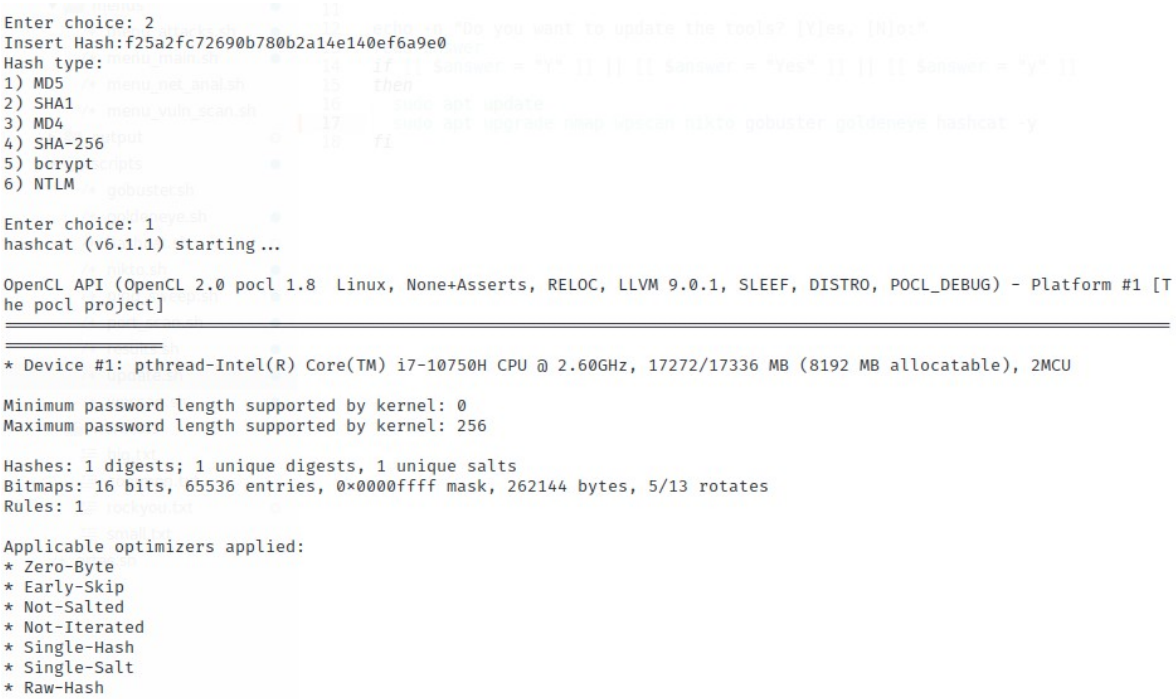
Σε αυτή την επιλογή χρησιμοποιείτε το εργαλείο GoldenEye. Το εργαλείο «ARTAS» ζητάει την διεύθυνση του στόχου και στην συνέχεια ξεκινάει μια επίθεση άρνησης υπηρεσίας.

```
Enter choice: 1
Target URL: http://10.10.10.10

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 10 workers running 500 connections each. Hit CTRL+C to cancel.
^CCTRL+C received. Killing all workers
0 GoldenEye strikes hit. (457 Failed)
0 GoldenEye strikes hit. (457 Failed)
0 GoldenEye strikes hit. (457 Failed)
0 GoldenEye strikes hit. (457 Failed)
0 GoldenEye strikes hit. (457 Failed)
Shutting down GoldenEye
```

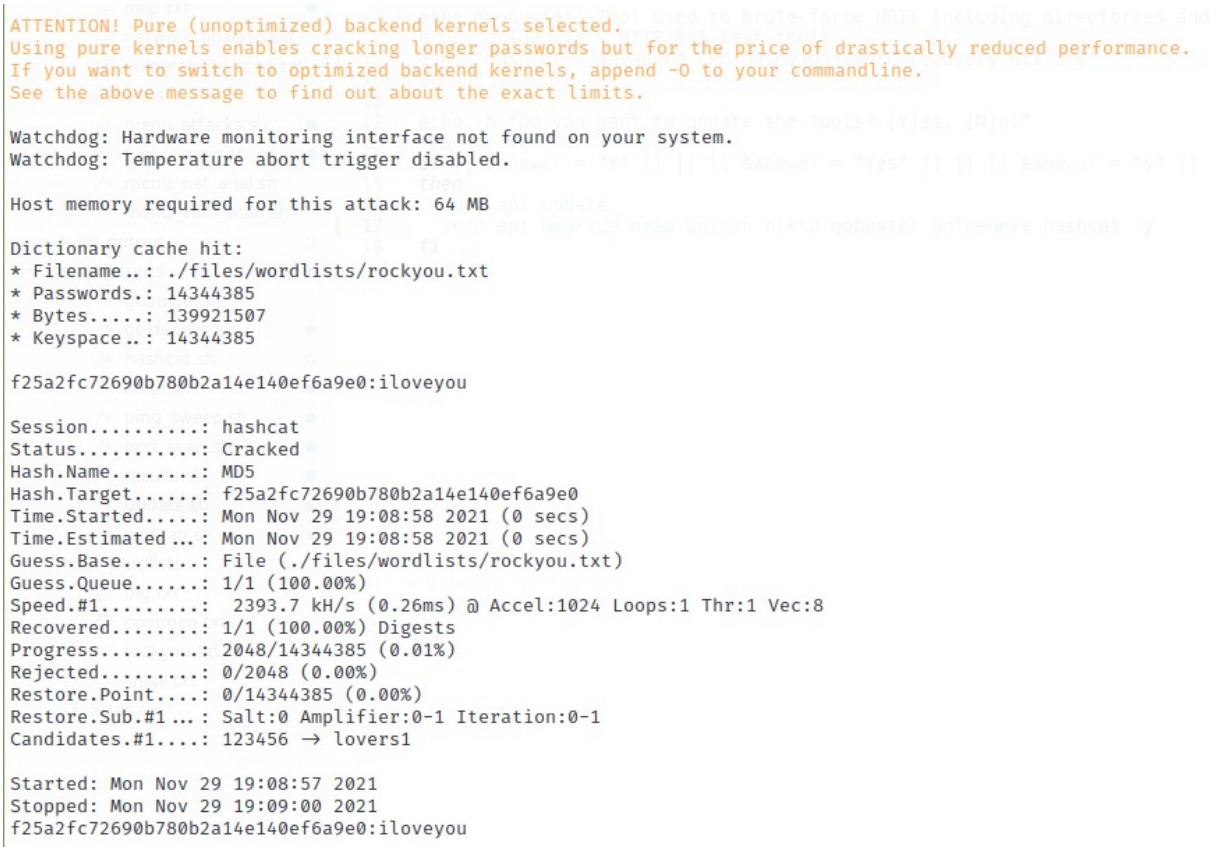
Εικόνα 20. Στην εικόνα βλέπουμε πως λειτουργεί η επιλογή DoS. Η διεύθυνση στόχος είναι η 10.10.10.10 και η επίθεση δεν θα σταματήσει μέχρι να πατήσουμε CTRL+C.

4.3.4.1 Hash Crack

Στην επιλογή Hash Crack χρησιμοποιείται το εργαλείο Hashcat. Το εργαλείο «ARTAS» αρχικά ζητάει το hash που επιθυμούμε να σπάσουμε και στην συνέχεια επιλέγουμε με πιο αλγόριθμο έχει κατατεμαχιστεί ο κωδικός. Το λεξικό που χρησιμοποιείται για την επίθεση είναι το rockyou. Αυτό το λεξικό είναι δημοφιλές καθώς έχει τους πιο συνηθισμένους κωδικούς που χρησιμοποιούνται.



Εικόνα 21



Εικόνα 22: Στην εικόνα 21 και 22 βλέπουμε πως λειτουργεί η επιλογή Hash Crack. Το hash που θέλουμε να σπάσουμε είναι το f25a2fc72690b780b2a14e140ef6a9e0 και επιλέξαμε τον MD5 ως αλγόριθμό hashing. Το αποτέλεσμα του hash είναι ο κωδικός iloveyou.

4.3.5 Results

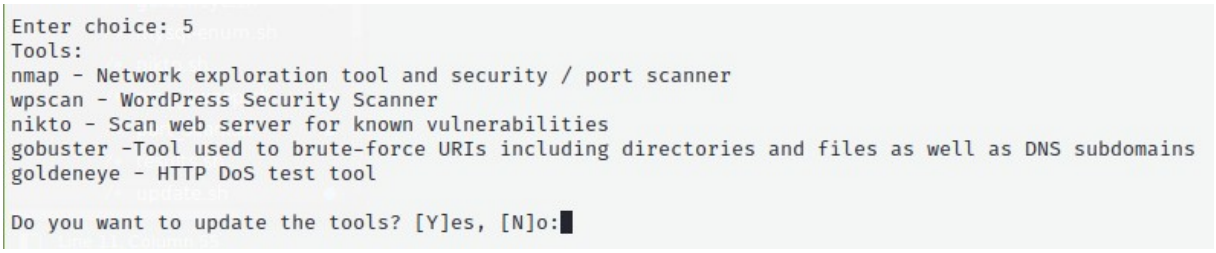
Το εργαλείο «ARTAS» προσφέρει την δυνατότητα τα αποτελέσματα όλων των επιθέσεων που έγιναν να αποθηκεύονται. Με αυτό τον τρόπο στην επιλογή Results μπορούμε να δούμε όλες τις επιθέσεις που έγιναν διαχωρισμένες με την ημερομηνία και ώρα που εκτελέστηκαν. Όταν επιλέξουμε μια από αυτές βλέπουμε τα αποτελέσματα.

```
Enter choice: 4
1) 2021-11-11_16:47:16_nmap
2) 2021-11-11_16:50:42_nikto.txt
3) 2021-11-11_16:50:48_gobuster.log
4) 2021-11-12_05:59:42_nmap
5) 2021-11-12_06:02:11_nmap
6) 2021-11-12_06:04:36_nmap
7) 2021-11-12_06:04:51_nmap
8) 2021-11-12_06:05:24_gobuster.log
9) 2021-11-12_06:07:18_nikto.txt
10) 2021-11-12_06:18:20_nikto.txt
11) 2021-11-26_15:31:37_nmap
12) 2021-11-26_16:25:57_nmap
13) 2021-11-26_17:41:37_nmap
14) 2021-11-27_11:13:58_nmap
15) 2021-11-27_11:16:13_gobuster.log
16) 2021-11-27_11:16:35_nikto.txt
17) 2021-11-27_17:14:46_nmap
18) 2021-11-27_17:25:43_gobuster.log
19) 2021-11-27_17:58:24_nikto.txt
20) 2021-11-27_17:58:35_nikto.txt
21) 2021-11-27_18:35:11_wpsacn
22) 2021-11-27_19:29:27_wpsacn
Choose number to show the result:
18
/.hta                 (Status: 403) [Size: 291]
/.hta.txt             (Status: 403) [Size: 295]
/.hta.php             (Status: 403) [Size: 295]
/.hta.html            (Status: 403) [Size: 296]
/.htaccess.txt        (Status: 403) [Size: 300]
/.htpasswd            (Status: 403) [Size: 296]
/.htaccess.php        (Status: 403) [Size: 300]
/.htpasswd.txt        (Status: 403) [Size: 300]
/.htaccess.html       (Status: 403) [Size: 301]
/.htpasswd.php        (Status: 403) [Size: 300]
/.htaccess            (Status: 403) [Size: 296]
/.htpasswd.html       (Status: 403) [Size: 301]
/assets               (Status: 301) [Size: 313] [--> http://10.10.186.64/assets/]
/denied.php           (Status: 302) [Size: 0] [--> /login.php]
/index.html           (Status: 200) [Size: 1062]
/index.html           (Status: 200) [Size: 1062]
/login.php            (Status: 200) [Size: 882]
/portal.php           (Status: 302) [Size: 0] [--> /login.php]
/robots.txt           (Status: 200) [Size: 17]
/robots.txt           (Status: 200) [Size: 17]
/server-status        (Status: 403) [Size: 300]
```

Εικόνα 23. Οι επιθέσεις που έχουν πραγματοποιηθεί και τα αποτελέσματα χωρισμένα με την ημερομηνία που εκτελέστηκαν. Στην συνέχεια επιλέγει μια επίθεση και τα αποτελέσματα εμφανίζονται στην οθόνη.

4.3.6 Tools/Update

Εδώ εμφανίζονται ενημερωτικά όλα τα Scripts που αυτοματοποιούνται στο εργαλείο «ARTAS». Γίνεται μια πάρα πολύ σύντομη περιγραφή του τι είναι το κάθε ένα και στην συνέχεια υπάρχει η επιλογή να τα αναβαθμίσουμε στην τελευταία τους έκδοση. Εάν ένα εργαλείο δεν είναι εγκατεστημένο στο σύστημα, το εργαλείο «ARTAS» θα το κατεβάσει αυτόματα μαζί με τις υπόλοιπες αναβαθμίσεις.



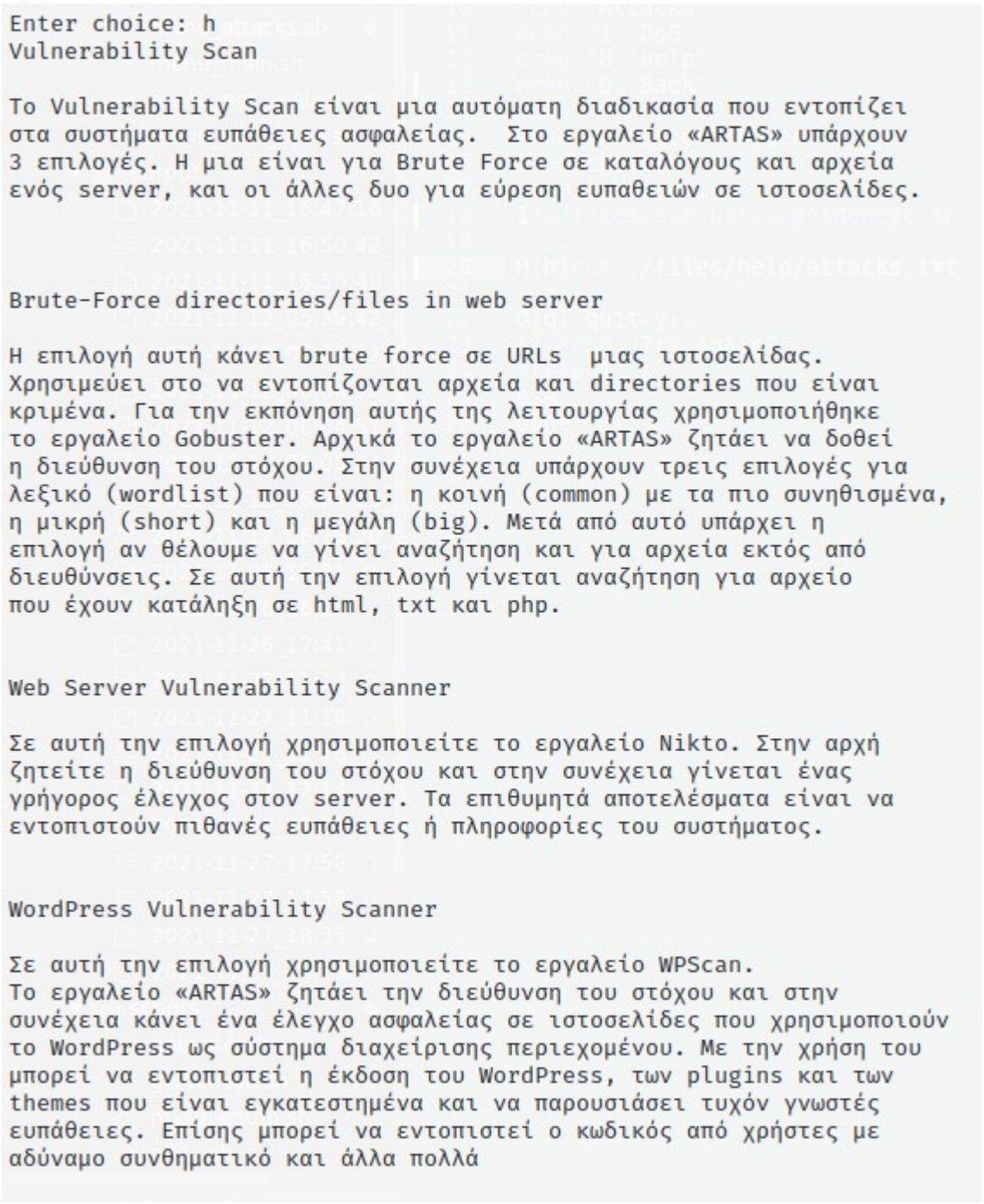
Εικόνα 24. Η επιλογή Tools/Update στο εργαλείο «ARTAS».

4.3.7 Help

Στα μενού Network Analysis, Vulnerability Scan και Attacks υπάρχει η επιλογή Help. Σε αυτή την επιλογή δίνονται μερικές πληροφορίες ως προς το τι κάνει η κάθε επίθεση. Επίσης πρίν την εκτέλεση του εργαλείου αν δωθεί το flag “-h” πέρνουμε κάποιες πληροφορίες ως προς τες δυνατότητες του εργαλείου.

```
Enter choice: h
Network Analysis

Όταν ο χρήστης επιλέξει την επιλογή Network Analysis θα εμφανιστή
ένα νέο μενού με τις επιλογές Ping Sweep και Port Scan. Σε αυτή
την κατηγορία βρίσκονται οι επιλογές για σάρωση δικτύου και σάρωση
συγκεκριμένου host.

##########################

Ping Sweep

Το Ping Sweep είναι μια μέθοδος που σαρώνει ένα ολόκληρο δίκτυο για
να εντοπίσει hosts που είναι σε σύνδεση Όταν ο χρήστης διαλέξει την
επιλογή του Ping Sweep στο εργαλείο καλείτε να δώσει την διεύθυνση
που θέλει να σαρώσει και το Subnet Mask. Για το Subnet Mask έχει
τρεις επιλογές 8, 16 και 24 με την τελευταία να είναι η προκαθορισμένη.
Για την πραγμάτωση αυτής της δυνατότητας στο εργαλείο «ARTAS»
αυτοματοποιήθηκε μέρος από τις δυνατότητες του εργαλείου Nmap.

_____


Port Scan

Το Port Scan χρησιμοποιείτε για να διερευνηθεί ένας server ή host και
να εντοπιστούν ανοιχτά ports. Για των εντοπισμό των ανοιχτών ports η
διαδικασία που ακολουθείτε είναι να στέλνονται πακέτα στον server/host
και ανάλογα με την απάντηση που θα σταλεί πίσω να γνωρίζουμε αν μια
υπηρεσία είναι διαθέσιμη ή όχι. Τα είδη των πακέτων που μπορούν να
σταλούν αλλά και ο τρόπος αποστολής τους μπορεί να διαφέρει κάθε φορά
ανάλογά με τα αποτελέσματα που θέλουμε να δούμε. Στο εργαλείο «ARTAS»
αυτοματοποιήθηκε ένα μεγάλο μέρος από τις δυνατότητες του εργαλείου
Nmap όσο αφορά το Port Scanning. Στις υποενότητες που ακολουθούν θα
αναλυθούν όλες οι δυνατές επιλογές που μπορεί να επιλέξει ένας χρήστης.



TCP Connect Scans

Το TCP Connect Scan λειτουργεί εκτελώντας ένα ολοκληρωμένο three-way
handshake για κάθε port στόχο. Στην ουσία προσπαθεί να συνδεθεί σε κάθε
port και να καθορίσει αν αυτό είναι ανοιχτό. Στην περίπτωση όπου ένα port
είναι κλειστό με το που σταλεί το πρώτο SYN πακέτο επιστρέφεται ένα πακέτο
RST. Έτσι αναγνωρίζει ότι το port είναι κλειστό.


SYN "Half-open" Scans

Όπως και στο TCP scan έτσι και στο SYN Scan ελέγχεται το κάθε port ξεχωριστά.
Ωστόσο η διαφορά είναι ότι ένω το TCP Scan κάνει ένα ολόκληρο three-way
handshake στον στόχο, το SYN scan στέλνει ένα RST πακέτο όταν πάρει SYN/ACK
```

Εικόνα 25. Οι βοηθητικές πληροφορίες στο Network Analysis μενού



Εικόνα 26. Οι βοηθητικές πληροφορίες στο Vulnerability Scan μενού

```
Attacks
1. DoS
2. Hash Crack
H. Help
Q. Back

Enter choice: h
Attacks

Στο Attacks μενού υπάρχουν τα εργαλείου που κάνουν κάποιο είδος
επίθεσης. Αυτή την στιγμή το εργαλείο  «ARTAS» έχει δυο επιλογές.
Η πρώτη είναι το Denial of service (DoS) που στοχεύει web servers
και η δεύτερη είναι το Hash Crack που προσπαθεί να σπάσει το hash
που θα του δώσει ο χρήστης.


DoS

Σε αυτή την επιλογή χρησιμοποιείτε το εργαλείο GoldenEye.
Το εργαλείο «ARTAS» ζητάει την διεύθυνση του στόχου και στην
συνέχεια ξεκινάει μια επίθεση άρνησης υπηρεσίας.


Hash Crack

Στην επιλογή Hash Crack χρησιμοποιείτε το εργαλείο Hashcat.
Το εργαλείο «ARTAS» αρχικά ζητάει το hash που επιθυμούμε να σπάσουμε
και στην συνέχεια επιλέγουμε με πιο αλγόριθμο έχει κατατεμαχιστεί ο
κωδικός. Το λεξικό που χρησιμοποιείτε για την επίθεση είναι το rockyou.
Αυτό το λεξικό είναι δημοφιλή καθώς έχει τους πιο συνηθισμένους κωδικούς
που χρησιμοποιούνται.
```

Εικόνα 27. Οι βοηθητικές πληροφορίες στο Attacks μενού

```
└─$ ./artas.sh -h
Automatic Red Team Attack Scripts

1. Network Analysis
1.1. Ping Sweep
1.2. Port Scan

2. Vulnerability Scan
2.1. Brute-Force directories/files in web server
2.2. Web Server Vulnerability Scanner
2.3. WordPress Vulnerability Scanner

3. Attacks
3.1. DoS
3.2. Hash Crack
```

Εικόνα 28. Οι βοηθητικές πληροφορίες πριν την εκκίνηση του εργαλείου «ARTAS»

Κεφάλαιο 5
Επίλογος

5.1 Συμπέρασμα

Η ασφάλεια των πληροφοριακών συστημάτων γίνεται όλο και πιο σημαντική χρόνο με τον χρόνο. Στόχος της παρούσας διατριβής ήταν να μελετηθούν οι τρόποι με τους οποίους μια red team ελέγχει ένα σύστημα για να εντοπίσει ευπάθειες και ευάλωτα σημεία. Με βάση την μελέτη της βιβλιογραφίας φάνηκε ότι τα εργαλεία που υπάρχουν μέχρι τώρα υστερούν σε ευχρηστία, απλότητα, και αυτοματοποίηση. Γι’ αυτό κρίθηκε αναγκαία η υλοποίηση του αυτοματοποιημένου εργαλείου «ARTAS» ώστε να προστεθεί ακόμα ένα εργαλείο στους επαγγελματίες ασφάλειας πληροφοριακών συστημάτων. Ως γλώσσα υλοποίησης έχει επιλεγεί η bash script ώστε το «ARTAS» να είναι συμβατό άμεσα σε όλες τις διανομές Unix αλλά και να αυτοματοποιήσει αρκετές διεργασίες που ήδη γίνονταν μέσα από το terminal. Το εργαλείο «ARTAS» δοκιμάστηκε με επιτυχία σε cyber-range περιβάλλον και βελτιώνει σημαντικά την ευχρηστία και την απλότητα των διεργασιών που αυτοματοποιήθηκαν που γίνονταν ήδη μέσα από το terminal.

5.2 Μελλοντική εργασία

Ως μελλοντική εργασία βρίσκεται η περαιτέρω ανάπτυξη του εργαλείου «ARTAS». Μέχρι στιγμής το εργαλείο έχει αυτοματοποιήσει ένα πολύ μικρό αριθμό επιθέσεων σε σύγκριση με τον αριθμό των scripts που χρησιμοποιεί ένας επαγγελματίας ασφάλειας πληροφοριακών συστημάτων. Η προσθήκη περισσότερων red team scripts έχει ως αποτέλεσμα την αύξηση της αυτοματοποίησης και απλότητας στον τομέα

αυτό. Επίσης προτείνεται η ανάπτυξη του εργαλείου που υλοποιήθηκε και σε γραφικό περιβάλλον, πέραν της γραμμής εντολών. Αυτό θα βοηθούσε περισσότερο τους νέους στον χώρο της ασφάλειας πληροφοριακών συστημάτων ή τα άτομα που δεν είναι εξοικειωμένα με την χρήση του terminal.

Βιβλιογραφία

[1] Diogenes, Y. and Ozkaya, E., 2019. Cybersecurity - Attack and Defense Strategies - Second Edition. 2nd ed. pp.39-48.

[2] Engebretson, P., 2013. The basics of hacking and penetration testing. Amsterdam: Syngress, an imprint of Elsevier, p.xi.

[3] Dunham, K. and Melnick, J., 2009. Malicious Bots: An Inside Look into the Cyber-Criminal Underground of the Internet. pp.1-6.

[4] Συναρτήσεις κατακερματισμού, Σημειώσεις μαθήματος «ΑΥΔ621 - Κρυπτογραφία» του τμήματος Ασφάλεια Υπολογιστών και Δικτύων, Ανοικτό Πανεπιστήμιο Κύπρου, διάλεξη 8,
σελίδα 32, Κ. Λιμνιώτης, 2016.

[5] Applebaum, Andy et al. “Intelligent, automated red team emulation.” Proceedings of the 32nd Annual Conference on Computer Security Applications, 2016.

[6] Miller, Doug et al. “Automated Adversary Emulation : A Case for Planning and Acting with Unknowns.”, 2018.

[7] Nutile, T., 2021. Testing Your Network Defenses by Imitating Malicious Adversaries. [online] The MITRE Corporation. Available at: <https://www.mitre.org/publications/project-stories/testing-your-network-defenses-by-imitating-malicious-adversaries> [Accessed 30 November 2021].

[8] Enoch, Simon Yusuf et al. “HARMer: Cyber-Attacks Automation and Evaluation.” IEEE Access 8, 2020.

[9] Holm, Hannes and Teodor Sommestad. “So long, and thanks for only using readily available scripts.” Inf. Comput. Secur. 25, 2017.

[10] Randhawa, Suneel et al. “Mission-Centric Automated Cyber Red Teaming.” Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018.

[11] Holík, Filip et al. “Effective penetration testing with Metasploit framework and methodologies.” 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), 2014.

[12] Framework, M., 2021. Metasploit Framework | Complete Guide to Metasploit Framework. [online] EDUCBA. Available at: <https://www.educba.com/metasploit-framework/> [Accessed 30 November 2021].

[13] Cobalt Strike Research and Development. 2021. Getting Started with Armitage and the Metasploit Framework | Cobalt Strike. [online] Available at: <https://blog.cobaltstrike.com/2013/02/06/getting-started-with-armitage-and-the-metasploit-framework-2013/> [Accessed 30 November 2021].

[14] Offensive-security.com. 2021. Armitage | Offensive Security. [online] Available at: <https://www.offensive-security.com/metasploit-unleashed/armitage/> [Accessed 30 November 2021].

[15] ITperfection - Network Security. 2021. What is NESSUS and How Does it Work? - ITperfection - Network Security. [online] Available at: <https://www.itperfection.com/network-security/network-monitoring/what-is-nessus-and-how-does-it-work-network-munitoring-vulnerabilit-scaning-security-data-windows-unix-linux/> [Accessed 30 November 2021].

[16] ΣΑΕ521 Ποιοτική έρευνα, Σημειώσεις μαθήματος «ΣΑΕ521 - Ερευνητικές Μέθοδοι» Σχολή Θετικών και Εφαρμοσμένων Επιστημών, Ανοικτό Πανεπιστήμιο Κύπρου, διάλεξη 5, σελίδες 28-33, Α. Γερατζιώτης, J. Otterbacher, Χ. Κατσάνος, Α. Περατικού, 2018.

[17] Ίσαρη, Φ. and Πουρκός, Μ., 2015. Ποιοτική Μεθοδολογία Έρευνας. pp.11-14.

[18] Ucy.ac.cy. 2021. Είδη Ερευνών. [online] Available at: <https://www.ucy.ac.cy/pakepe/el/research-services/research-kind> [Accessed 30 November 2021].

[19] Μοντέλα Κύκλου Ζωής Λογισμικού, Σημειώσεις μαθήματος «ΕΠΛ 361 - Τεχνολογία Λογισμικού» του τμήματος Πληροφορικής, Πανεπιστημίου Κύπρου, διάλεξη 3-4, Γ. Καπιτσακη, 2015

[20] Unito. 2021. SDLC: Software Development Life Cycle Phases and Methodologies - Unito. [online] Available at: <https://unito.io/blog/sdlc-methodology-guide/> [Accessed 30 November 2021].

[21] Μοντέλα και Μεθοδολογίες Ανάπτυξης Λογισμικού, Σημειώσεις μαθήματος Τεχνολογία Λογισμικού της Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο, διάλεξη 2, Κ. Σαΐδης, 2017

[22] Guru99. 2021. Spiral Model: When to Use? Advantages & Disadvantages. [online] Available at: <https://www.guru99.com/what-is-spiral-model-when-to-use-advantages-disadvantages.html> [Accessed 30 November 2021].

[23] 2021. What is Kali Linux. [online] Available at: <https://www.kali.org/docs/introduction/what-is-kali-linux/> [Accessed 30 November 2021].

[24] Gnu.org. 2020. Bash Reference Manual. [online] Available at: <https://www.gnu.org/software/bash/manual/bash.pdf> [Accessed 30 November 2021].

[25] Calderon, Paulino. “Nmap : network exploration and security auditing cookbook : a complete guide to mastering Nmap and its scripting engine, covering practical tasks for penetration testers and system administrators.” 2017.

[26] GitHub. 2021. WPScan User Documentation · wpscanteam/wpscan Wiki. [online] Available at: <https://github.com/wpscanteam/wpscan/wiki/WPScan-User-Documentation> [Accessed 30 November 2021].

[27] GitHub. 2021. Overview & Description · sullo/nikto Wiki. [online] Available at: <https://github.com/sullo/nikto/wiki/Overview-&-Description> [Accessed 30 November 2021].

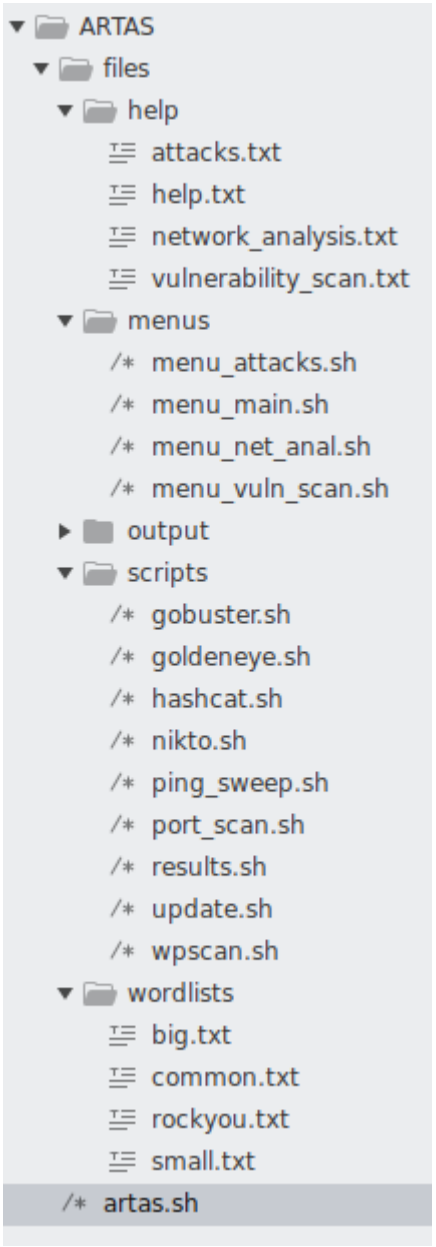
[28] GitHub. 2021. GitHub - OJ/gobuster: Directory/File, DNS and VHost busting tool written in Go. [online] Available at: <https://github.com/OJ/gobuster> [Accessed 30 November 2021].

[29] GitHub. 2021. GitHub - jseidl/GoldenEye: GoldenEye Layer 7 (KeepAlive+NoCache) DoS Test Tool. [online] Available at: <https://github.com/jseidl/GoldenEye> [Accessed 30 November 2021].

[30] Hashcat.net. 2021. hashcat - advanced password recovery. [online] Available at: <https://hashcat.net/hashcat/> [Accessed 30 November 2021].

Παράρτημα Α
Κώδικας Bash

Α.1 Τα αρχεία του εργαλείου «ARTAS»



A.2 Main Menu

```
#!/bin/bash


quit=n
while [  "$quit"   =   "n"  ]
do
#clear
echo
echo
echo "Main Menu:"
echo "1. Network Analysis"
echo "2. Vulnerability Scan"
echo "3. Attacks"
echo "4. Results"
echo "5. Tools/Update"
echo "Q. Quit"
echo
echo -n "Enter choice: "
read choice
case $choice in
1)./files/menus/./menu_net_anal.sh
  ;;
2)./files/menus/./menu_vuln_scan.sh
  ;;
3)./files/menus/./menu_attacks.sh
  ;;
4)out=`ls ./files/output/`
  if [ -z "$out" ]
    then
        echo "No Results"
    else
        ./files/scripts/results.sh
    fi
  ;;
5)./files/scripts/update.sh
  ;;
Q|q) quit=y;;
*) echo "Try Again"
sleep 1
esac
done
echo "Thank you, Come again"
```

A.3 Network Analysis Menu

```
#!/bin/bash


quit=n
while [  "$quit"   =   "n"  ]
do
#clear
echo
echo
echo "Network Analysis"
echo "1. Ping Sweep"
echo "2. Port Scan"
echo "H. Help"
echo "Q. Back"
echo
echo -n "Enter choice: "
read choice
case $choice in
1)./files/scripts/./ping_sweep.sh
  ;;
2)./files/scripts/./port_scan.sh
  ;;
H|h)cat ./files/help/network_analysis.txt
  ;;
Q|q) quit=y;;
*) echo "Try Again"
sleep 1
esac
done
```

A.4 Vulnerability Scan Menu

```
#!/bin/bash


quit=n
while [  "$quit"   =   "n"  ]
do
#clear
echo
echo
echo "Vulnerability Scan"
echo "1. Brute-Force directories/files in web server"
echo "2. Web Server Vulnerability Scanner"
echo "3. WordPress Vulnerability Scanner"
echo "H. Help"
echo "Q. Back"
echo
echo -n "Enter choice: "
read choice
case $choice in
1)./files/scripts/./gobuster.sh
  ;;
2)./files/scripts/./nikto.sh
  ;;
3)./files/scripts/./wpscan.sh
  ;;
H|h)cat ./files/help/vulnerability_scan.txt
  ;;
Q|q) quit=y;;
*) echo "Try Again"
sleep 1
esac
done
```

A.5 Attacks Menu

```
#!/bin/bash


quit=n
while [  "$quit"   =   "n"  ]
do
#clear
echo
echo
echo "Attacks"
echo "1. DoS"
echo "2. Hash Crack"
echo "H. Help"
echo "Q. Back"
echo
echo -n "Enter choice: "
read choice
case $choice in
1)./files/scripts/./goldeneye.sh
  ;;
2)./files/scripts/./hashcat.sh
  ;;
H|h)cat ./files/help/attacks.txt
  ;;
Q|q) quit=y;;
*) echo "Try Again"
sleep 1
esac
done
```

A.6 Ping Sweep

```
1   #!/bin/bash
2
3   ip=localhost
4   mask=16
5
6   echo -n "Target IP [Default=localhost]:"
7   read ip
8   if [[  -z "$ip" ]]
9   then
10      ip="localhost"
11  fi
12
13  echo ""
14  echo "Subnet Mask:"
15  echo "1) 8"
16  echo "2) 16"
17  echo "3) 24 [Default]"
18  echo
19  echo -n "Enter choice: "
20  read mask
21  case $mask in
22      1) mask="8";;
23      2) mask="16" ;;
24      *) mask="24" ;;
25  esac
26
27  nmap -sn $ip/$mask  #Ping Sweep
```

A.7 Port Scan

```
#!/bin/bash

ip=localhost

echo -n "Target URL:"
read ip
echo "Scan type:"
echo "1) TCP [Default]"
echo "2) SYN"
echo "3) UDP"
echo "4) NULL"
echo "5) FIN"
echo "6) Xmas"
echo
echo -n "Enter choice: "
read scan_type
case $scan_type in
    2) scan_type="-sS" ;;
    3) scan_type="-sU" ;;
    4) scan_type="-sN" ;;
    5) scan_type="-sF" ;;
    6) scan_type="-sX" ;;
    *) scan_type="-sT" ;;
esac

echo ""
echo "Flags:"
echo "0) None [Default]"
echo "1) Treat all hosts as online"
echo "2) Aggresive Scan"
echo "3) OS detection"
echo "4) Version detetion"
echo "5) Scan ports from 1 through 65535"
echo "6) Firewall Evasion: Generate invalid checksum for packets"
echo "7) Firewall Evasion: Fragment the packets into smaller pieces"
echo
echo -n "Enter choice: "
read flag
case $flag in
    0) flag="" ;;
    1) flag="-Pn" ;;
    2) flag="-A" ;;
    3) flag="-O" ;;
    4) flag="-sV" ;;
    5) flag="-p-" ;;
    6) flag="--badsum" ;;
    7) flag="-f" ;;
esac
```

```
49
50  echo ""
51  echo "Time (from slower to faster):"
52  echo "1) Paranoid"
53  echo "2) Sneaky"
54  echo "3) Polite"
55  echo "4) Normal [Default]"
56  echo "5) Aggressive"
57  echo "6) Insane"
58  echo
59  echo -n "Enter choice: "
60  read time
61  case $time in
62      1) time="-T0" ;;
63      2) time="-T1" ;;
64      3) time="-T2" ;;
65      5) time="-T4" ;;
66      6) time="-T5" ;;
67      *) time="-T3" ;;
68  esac
69
70  nmap $scan_type $flag $time $ip -oN ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_nmap
```

A.8 Brute-Force directories/files in web server

```
1   #!/bin/bash
2
3   ip=localhost
4
5   echo -n "Target URI: http://"
6   read ip
7   echo -n "Wordlist: [C]ommon (default), [S]mall, [B]ig:"
8   read wordlist
9   case $wordlist in
10      s|S) wordlist=./files/wordlists/small.txt;;
11      b|B) wordlist=./files/wordlists/big.txt;;
12      *) wordlist=./files/wordlists/common.txt;;
13  esac
14
15  echo -n "File extensions: [Y]es, [N]o (default)"
16  read ext
17  case $ext in
18      y|Y) gobuster dir -x txt,php,html -u $ip -w $wordlist -o ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_gobuster.log;;
19      *) gobuster dir -u $ip -w $wordlist -o ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_gobuster.log;;
20  esac
```

A.9 Web Server Vulnerability Scanner

```
1   #!/bin/bash
2
3   ip=localhost
4
5   echo -n "Target URL: "
6   read ip
7
8
9   nikto -h $ip -output ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_nikto.txt
```

A.10 WordPress Vulnerability Scanner

```
1  #!/bin/bash
2
3  ip=localhost
4
5  echo -n "Target URL: "
6  read ip
7
8
9  wpscan --url $ip | tee ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_wpsacn
10
```

A.11 DoS

```
1  #!/bin/bash
2
3
4  echo -n "Target URL: http://"
5  read ip
6  ip="http://${ip}"
7
8
9  goldeneye $ip
```

A.12 Hash Crack

```
#!/bin/bash


echo -n "Insert Hash:"
read hash
echo $hash > /tmp/artas.hash
echo "Hash type:"
echo "1) MD5"
echo "2) SHA1"
echo "3) MD4"
echo "4) SHA-256"
echo "5) bcrypt"
echo "6) NTLM"
echo
echo -n "Enter choice: "
read type
case $type in
    1) type="0" ;;
    2) type="100" ;;
    3) type="900" ;;
    4) type="1400" ;;
    5) type="3200" ;;
    *) type="1000" ;;
esac

hashcat -m $type /tmp/artas.hash ./files/wordlists/rockyou.txt

hashcat -m $type /tmp/artas.hash --show > /tmp/artas.hashed

if [ -s /tmp/artas.hashed ]; then
        # The file is not-empty.
        cat /tmp/artas.hashed > ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_hashcat.txt
        cat ./files/output/$(date "+%Y-%m-%d_%H:%M:%S")_hashcat.txt
else
        echo "No Results"
fi
```

A.13 Update

```
#!/bin/bash

echo "Tools:"
echo "nmap - Network exploration tool and security / port scanner"
echo "wpscan - WordPress Security Scanner"
echo "nikto - Scan web server for known vulnerabilities"
echo "gobuster -Tool used to brute-force URIs including directories and files as well as DNS subdomains"
echo "goldeneye - HTTP DoS test tool"
echo "hashcat - Advanced CPU-based password recovery utility"
echo ""

echo -n "Do you want to update the tools? [Y]es, [N]o:"
read answer
if [[ $answer = "Y" ]] || [[ $answer = "Yes" ]] || [[ $answer = "y" ]]
then
  sudo apt update
  sudo apt upgrade nmap wpscan nikto gobuster goldeneye hashcat -y
fi
```

A.14 Results

```
#!/bin/bash

out=`ls ./files/output/ | tr -s ' ' '\n'`
n=1
while read -r line;
do
echo "$n) $line"
n=$((n+1))
done < <(printf '%s\n' "$out")

echo "Choose number to show the result:"
read res
n=1
while read -r line;
do
if [[ "$res" == "$n" ]]
then
    cat ./files/output/$line
fi
n=$((n+1))
done < <(printf '%s\n' "$out")
```