# ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΔΙΚΤΥΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ ΤΗΛΕΜΑΤΙΚΗΣ

Διπλωματική Εργασία

# ΕΚΤΙΜΗΣΗ ΕΥΠΑΘΕΙΩΝ

# ΣΕ ΔΙΑΚΟΜΙΣΤΕΣ ΔΙΑΔΙΚΤΥΟΥ

ΤΕΡΖΟΥΔΗΣ ΔΗΜΗΤΡΗΣ

Αθήνα, 2018

# ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΔΙΚΤΥΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ ΤΗΛΕΜΑΤΙΚΗΣ

## Τριμελής Εξεταστική Επιτροπή

**ΔΑΛΑΚΑΣ ΒΑΣΙΛΕΙΟΣ**[Επιβλέπων]

**Ε.ΔΙ.Π Τμήμα Πληροφορικής & Τηλεματικής, Χαροκόπειο Πανεπιστήμιο**

**ΚΑΜΑΛΑΚΗΣ ΘΩΜΑΣ**

**Αναπληρωτής Καθηγητής,ΤμήμαΠληροφορικής & Τηλεματικής Χαροκόπειο Πανεπιστήμιο**

**ΛΙΜΝΙΩΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ**

**Διδάκτορας,Τμήμα Πληροφορικής & Τηλεματικής, Χαροκόπειο Πανεπιστήμιο**

Ο Τερζούδης Δημήτρης δηλώνω υπεύθυνα ότι:

1 Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2 Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

**Στην οικογένεια μου, που με στήριξε**

**στην σταδιοδρομία μου.**

# Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αγάπη τους, την υποστήριξη τους και την πίστη τους σε εμένα.Κατά κύριο λόγο όμως οφείλω να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Δαλάκα Βασίλειο που με υποστήριξε καθ' όλη τη διάρκεια της διπλωματικής εργασίας. Αισθάνομαι πολύ τυχερός που στο διάστημα αυτό είχα πάντα τη σιγουριά της υλικής αλλά και της ηθικής βοήθειας που χρειαζόμουν για να προχωρήσω.

Αθήνα, Φεβρουάριος 2018

# Περίληψη

Στόχος της παρούσας διατριβής είναι η εξέταση των ευπαθειών όπου δίνεται να εντοπίζονται στα σημερινά πληροφοριακά συστήματα, με χρήση εργαλείων και τεχνικών penetration testing. Τα πληροφοριακά συστήματα παρέχουν μια σύνδεση επικοινωνιακής συνεργασίας ανάμεσα στον άνθρωπο και τον υπολογιστή μέσω δεδομένων και διαφόρων τεχνολογικών διαδικασιών. Απώτερος σκοπός των συστημάτων αυτών είναι η τέλεια υποστήριξη για απλούς χρήστες έως και ανθρώπινο δυναμικό επιχειρήσεων στη διαχείριση λήψης σημαντικών τους αποφάσεων. Στο πλαίσιο της εργασίας, αρχικά γίνεται ο ορισμός της έννοιας ευπάθεια και στην συνέχεια εξετάζονται ποια είναι τα στοιχεία αυτά όπου κάνουν ένα πληροφοριακό σύστημα, να είναι πιθανός ευπαθές ως προς πιθανές επιθέσεις. Στην συνέχεια γίνεται ανάλυση συγκεκριμένων εργαλείων penetration testing καθώς και χρήση αυτών για τον έλεγχο πληροφοριακών συστημάτων. Αναλυτικότερα, αναλύονται τα εργαλεία OpenVas, Nikto και Grabber, η αρχιτεκτονική τους, ο τρόπος λειτουργίας τους καθώς και συνοπτικά η λειτουργία και λειτουργικότητα αυτών. Τα εργαλεία αυτά χρησιμοποιούνται σε συνδυασμό, στο πρακτικό κομμάτι της εργασίας, για την πραγματοποίηση ελέγχου των πληροφοριακών συστημάτων και την ανάλυσης ως προς υπαρκτές ευπάθειες αυτών. Τέλος, έπειτα της λήψης και τις ανάλυσης των αποτελεσμάτων γίνεται η συνολική αξιολόγηση αυτών ως προς την επικινδυνότητα τους αλλά και προτείνονται πιθανοί τρόποι αντιμετώπισης αυτών.

# Abstract

The aim of this dissertation is to examine the vulnerabilities in the present information systems, using penetration testing tools and techniques. Information systems provide a communication link between human and computer via data and various technological processes.The ultimate goal of these systems is the perfect support for simple users and human resources to manage their important decisions. In the context of the work, initially the definition of vulnerability is defined and then it is examined what these elements are in an information system, potentially vulnerable to possible attacks.It then analyzes specific penetration testing tools as well as their use for the control of information systems. In more detail, the OpenVas, Nikto and Grabber tools, their architecture, their operation and their operation and functionality are analyzed. These tools are used in combination, in the practical part of the work, for controlling information systems and analyzing their existing vulnerabilities. Finally, after the results are analyzed, they are evaluated in a comprehensive way as regards their risk, but also possible ways of dealing with them.

# Πίνακας Περιεχομένων

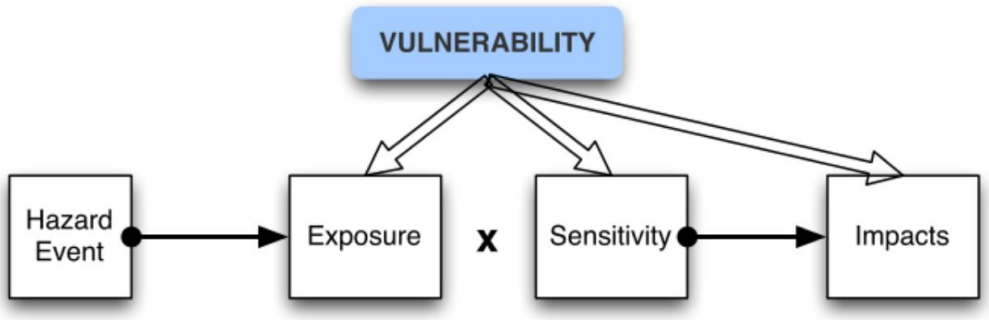
# Πίνακας Εικόνων

# Κεφάλαιο 1

Στο κεφάλαιο αυτό, γίνεται η εισαγωγή στις βασικές έννοιες και όρους της ανάλυσης της διατριβής καθώς και ο ορισμός του στόχου αυτής.

## 1.1 Τι ορίζουμε ευπάθεια;

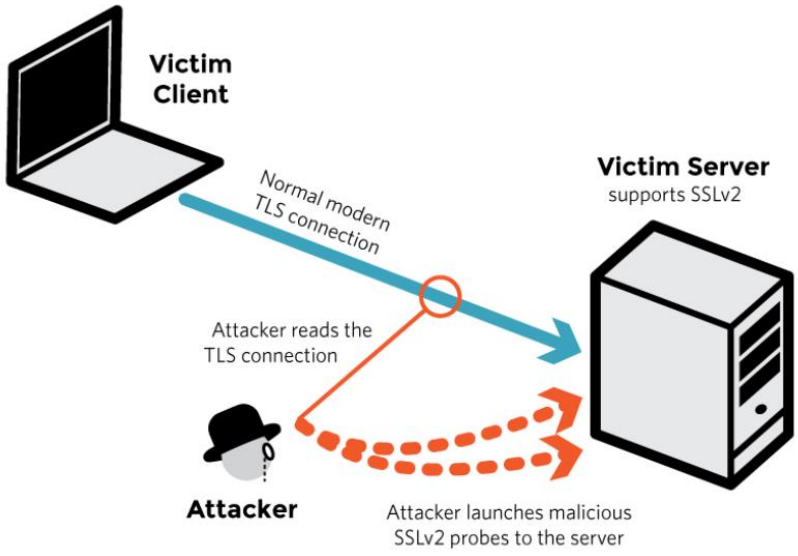
Ως Ευπάθεια *(vulnerability)* ονομάζεται μια "τρύπα" ή μια αδυναμία μιας εφαρμογής, η οποία μπορεί να οφείλεται σε ένα τρωτό σημείο στη σχεδίαση ή ένα σφάλμα υλοποίησης. Αυτή επιτρέπει σε έναν επιτιθέμενο να βλάψει τους ιδιοκτήτες και τους νόμιμους χρήστες της εφαρμογής, ή άλλες οντότητες, που βασίζονται στην εφαρμογή. Ο όρος «ευπάθεια» πολύ συχνά χρησιμοποιείται εσφαλμένα. Πρέπει να διακρίνεται από τους όρους threat(απειλή), attack(επίθεση) και countermeasures(αντίμετρα).




*Εικόνα 1 - Προβολή ευπάθειας και συμβάντων*

### 1.1.1 Απειλή

Απειλή (threat) είναι η ένδειξη του ότι επίκειται κάποιος κίνδυνος ή κάποιο κακό για την εφαρμογή ή το σύστημα γενικότερα. Είναι ο πιθανός κίνδυνος μιας επικείμενης επίθεσης που μπορεί να βλάψει την εφαρμογή. Οποιαδήποτε περίπτωση ή γεγονός με δυνατότητα πρόκλησης ζημιάς σε ένα σύστημα υπό μορφή καταστροφής, κοινοποίησης, τροποποίησης των στοιχείων του, ή/και άρνησης της υπηρεσίας.




*Εικόνα 2 - Σχήμα επίθεσης εντός δικτύου*

### 1.1.2 Επίθεση

Επιθέσεις (*attacks*) ονομάζονται οι τεχνικές, που χρησιμοποιούν οι επίδοξοι εισβολείς *(intruders*) για να εκμεταλλευθούν τις ευπάθειες των διαφόρων εφαρμογών. Οι επιθέσεις αυτές συχνά συγχέονται με τις ευπάθειες των εφαρμογών. Για το λόγο αυτό οφείλουμε να διευκρινίσουμε ότι επίθεση είναι μια πράξη την οποία ο εισβολέας κάνει σε μια εφαρμογή και δεν είναι μια αδυναμία αυτής.

## 1.2 Τι σημαίνει εκτίμηση;

Στο πρώτο στάδιο υπολογίστηκε ο ένας από τους τρεις παράγοντες που συνθέτουν την επικινδυνότητα. Συγκεκριμένα, αποτιμήθηκε η αξία των στοιχείων του πληροφοριακού συστήματος, τα οποία εφόσον έχουν σημαντική αξία ονομάζονται Αγαθά ή Περιουσιακά στοιχεία. Στο δεύτερο στάδιο υπολογίζονται οι άλλοι δύο παράγοντες, που είναι το επίπεδο των απειλών και το επίπεδο των αδυναμιών. Ο συνδυασμός αυτών των τριών παραγόντων θα μας δώσει το βαθμό επικινδυνότητας του συστήματος, έτσι ώστε να επιλεγούν τα κατάλληλα αντίμετρα. Τα βήματα που ακολουθεί το δεύτερο στάδιο είναι:

- Προσδιορισμός των Απειλών που αφορούν το κάθε Αγαθό.
- Εκτίμηση των Απειλών και αδυναμιών.
- Υπολογισμός της Επικινδυνότητας για κάθε συνδυασμό Αγαθού-Απειλής.
- Επιβεβαίωση και επικύρωση του Βαθμού Επικινδυνότητας

Η μέθοδος δεν περιορίζεται στον προσδιορισμό των πιθανών απειλών που καλείται να αντιμετωπίσει ένα πληροφοριακό σύστημα γενικά, αλλά επικεντρώνεται στον προσδιορισμό συγκεκριμένων απειλών για κάθε αγαθό ξεχωριστά. Η διαδικασία της εκτίμησης των κινδύνων δεν είναι και τόσο απλή. Τα άτομα τα οποία θα διεξάγουν αυτήν την εκτίμηση θα πρέπει να κατέχουν την κατάλληλη κατάρτιση, ενημέρωση και γνώση για να μπορέσουν να εκτιμήσουν σωστά τη πιθανότητα κάθε εμφάνισης κινδύνου ή ανεπιθύμητου γεγονότος. Επομένως οι λανθασμένες εκτιμήσεις μπορεί να έχουν και τις ανάλογες επιπτώσεις (π.χ αξιοπιστία όλης της διαδικασίας διαχείρισης κινδύνων).Από την άλλη μεριά υπάρχουν οι αισιόδοξες και απαισιόδοξες εκτιμήσεις.

## 1.3 Τι είναι πληροφοριακό σύστημα

Πληροφοριακό σύστημα ονομάζεται ένα σύνολο διαδικασιών, ανθρώπινου δυναμικού και αυτοματοποιημένων υπολογιστικών συστημάτων, που προορίζονται για τη συλλογή, εγγραφή, ανάκτηση, επεξεργασία, αποθήκευση και ανάλυση πληροφοριών. Τα συστήματα αυτά μπορούν να περιλαμβάνουν λογισμικό, υλικό και τηλεπικοινωνιακό σκέλος. Ένα Πληροφοριακό σύστημα αποτελείται από έξι στοιχεία:

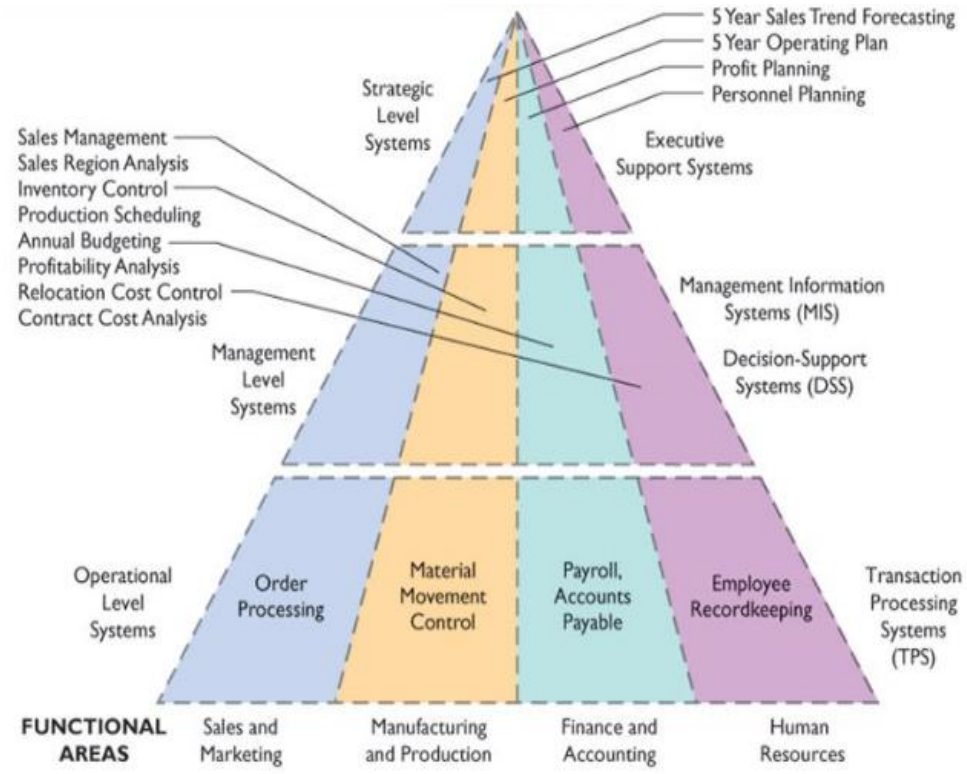
- Άνθρωποι(το σύνολο των ανθρώπων που εργάζονται με το πληροφοριακό σύστημα σε διάφορους ρόλους όπως χρήστες ,διαχειριστές κ.τ.λ.)
- Διαδικασίες(το σύνολο των οδηγιών για τη χρήση και το συνδυασμό όλων των στοιχείων υποδομής ενός ΠΣ)
- Βάσης δεδομένων.
- Λογισμικό.
- Υλικός εξοπλισμός.
- Δίκτυο.

Ένα Πληροφοριακό σύστημα βοηθάει στον έλεγχο, στο συντονισμό, στην ανάλυση προβλημάτων, στη λήψη αποφάσεων και στην ανάπτυξη νέων προϊόντων. Κάθε πληροφοριακό σύστημα πρέπει να προσδιορίζει, αποδοτικά και αποτελεσματικά, τις ανθρώπινες ανάγκες αυτών που χρησιμοποιούν το πληροφοριακό σύστημα καθώς και να επεξεργάζεται όλες τις πληροφορίες με αποτέλεσμα την ικανοποίηση των αναγκών αυτών. Αυτό γίνεται πραγματικότητα με την πιο αποτελεσματική ανάκτηση, αποθήκευση, επεξεργασία, παρουσίαση και διάδοση των πληροφοριών την παροχή των απαραίτητων μέσων και του κατάλληλου περιβάλλοντος μάθησης στους εμπλεκόμενους χρήστες ώστε να βελτιωθεί η αποτελεσματικότητα της διαδικασίας λήψης απόφασης την υποστήριξη των διαδικασιών λειτουργίας, ελέγχου και στρατηγικού σχεδιασμού την επιχείρησης ή του οργανισμού. Ένα πληροφοριακό σύστημα δημιουργείται, αναπτύσσεται, εξελίσσεται και αποσύρεται. Η ύπαρξή του αρχίζει από τη στιγμή που η επιχείρηση ή ο οργανισμός θα αποφασίσει τη δημιουργία του. Μετά έχουμε μια περίοδο στην οποία προσδιορίζονται οι βασικές απαιτήσεις των λειτουργιών του και

σχεδιάζονται οι λειτουργίες που ικανοποιούν τις απαιτήσεις αυτές. Έπειτα αρχίζει μια μεγάλη χρονική περίοδος στην οποία πραγματοποιείται η ανάπτυξή του και η διαρκής εξέλιξή του ώστε να ικανοποιεί τις ανάγκες της επιχείρησης ή του οργανισμού στον οποίο ανήκει. Τέλος όταν η επιχείρηση ή ο οργανισμός αποφασίσει ότι είναι πια αναποτελεσματικό και μη αποδοτικό, το πληροφοριακό σύστημα αποσύρεται. Υπάρχουν κάποιες βασικές αρχές στη χρήση και λειτουργία των πληροφοριακών συστημάτων που οφείλουν να ικανοποιούν κάποιες απαιτήσεις τις οποίες θα αναφέρουμε παρακάτω: Οι μμηχανισμοί ασφαλείας πρέπει να μην επιβαρύνουν τη συνολική αποτελεσματικότητα του συστήματος, όμως εάν αυτό δεν ισχύει τότε πρέπει να υπάρξουν αλλαγές ώστε η απόδοση και η ασφάλεια να βρίσκονται σε ισορροπία.

- Η διακίνηση εμπιστευτικών πληροφοριών προς τρίτους θα γίνεται επιτρεπτή μετά την ολοκλήρωση της εγγραφής άδειας του ενδιαφερόμενου.
- Η διαχείριση πληροφοριών θα πρέπει να πραγματοποιείται από εξουσιοδοτημένο προσωπικό.
- Τα δικαιώματα πρόσβασης πρέπει να προσδιορίζονται με ξεχωριστές και ανεξάρτητες διαδικασίες από αυτές του σταδίου υλοποίησης του συστήματος. Ο καθορισμός των διαδικασιών αυτών πραγματοποιείται σε νομοθετικό, οργανωτικό και δομικό επίπεδο.

Τα παραπάνω σημαίνουν ότι μια αποδοτική λειτουργία σε συνδυασμό με τη σωστή ανάπτυξη των πληροφοριακών συστημάτων είναι μια διαδικασία που εμπεριέχει μια δόμηση πλαισίου ασφαλείας, το οποίο παρέχει εξασφάλιση σε απαιτήσεις όπως διαθεσιμότητα, μυστικότητα και ορθότητα στα περιεχόμενα των πληροφοριών.




*Εικόνα 3 - Σχήμα δομής πληροφοριακού συστήματος*

## 1.4 Μέτρα ασφαλείας πληροφοριακών συστημάτων

Τα μέτρα ασφαλείας – προστασίας αφορούν όλες τις ενέργειες και τις διαδικασίες που περιορίζουν τις απειλές και τους κινδύνους ενός συστήματος, τα οποία διακρίνονται στις τέσσερις εξής παρακάτω κατηγορίες:

- Πρόληψη

  Αναλύεται από τα μέτρα τα οποία καταβάλλουν προσπάθειες για τη μείωση των κινδύνων.

- Διασφάλιση

  Αναλύεται από τα εργαλεία και στρατηγικές που βοηθούν στη διαρκής εξασφάλιση και συνεχής αποτελεσματικότητας των συγκεκριμένων μέτρων.

- Ανίχνευση

  Αναλύεται από τα συγκεκριμένα προγράμματα και τις τεχνικές που βοηθούν στην αντιμετώπιση ανεπιθύμητων περιστατικών.

- Επαναφορά

Αναλύεται από διαδικασίες που έχουν στόχο τη γρήγορη επαναφορά σε ένα ασφαλές περιβάλλον εφόσον υπήρξε παραβίαση ασφάλειας και ταυτόχρονα έρευνα για την αιτία που την προκάλεσε. Για να υπάρχει επιτυχία στην πολιτική της ασφαλείας, στο πλάνο της ασφαλείας οφείλεται να περιλαμβάνεται και η διαδικασία ενημέρωσης (update) ώστε με επισκοπήσεις της εφαρμογής να υπάρχει πάντα σε διαθεσιμότητα το up-to-date αναλόγως με τις τεχνολογικές εξελίξεις και αλλαγές της εταιρίας του αντίστοιχου λογισμικού.

## 1.5 Ιστορική αναδρομή ασφάλειας πληροφοριακών συστημάτων.

Η ιστορία της επιστήμης της ασφάλειας πληροφοριακών συστημάτων ξεκινάει λίγο πριν εξαπλωθεί η σύγχρονη επιστήμη των υπολογιστών κάπου στον εικοστό αιώνα. Η πρώτη δημοσίευση είναι γνωστό πως προήλθε από μία ομάδα εργασίας του συμβουλίου αμυντικής επιστήμης του υπουργείου άμυνας των Η.Π.Α. Ήθελαν να εξετάσουν τότε ποια θα είναι τα προβλήματα όταν θα χρειαστεί να συνδεθούν δύο υπολογιστές εξ αποστάσεως με τη χρήση τερματικών. Για να προσεγγίσουν τότε λύσεις προβλημάτων ασφαλείας έπρεπε να απομονώσουν (κλειδώσουν σε ιδιωτικό χώρο) τον κεντρικό υπολογιστή ώστε να ελέγχουν την πρόσβαση σε αυτόν. Αυτό δεν ήταν πρακτικά εφικτό να συνεχιστεί κι έτσι αργότερα αναγκάστηκαν να βρουν νέους τρόπους και μεθόδους για ασφάλεια. Οι ιοί εμφανίστηκαν όταν ξεκίνησε να εξαπλώνει τα πρώτα της βήματα η σύγχρονη επιστήμη των υπολογιστών, γύρω στη δεκαετία του ’70, με τον πρώτο ιό με την ονομασία *Creeper* καθώς και το πρώτο διαδικτυακό σκουλήκι *(worm)* με όνομα *Morris* λίγο πριν το 2000, το οποίο πρόσβαλλε χιλιάδες συστήματα και υπολογιστές. Ο πρώτος ιός που αναφέρεται ως πιο εξαπλωμένος εκτός του συστήματος, υπήρξε ο *Elk Cloner*, ο οποίος δημιουργήθηκε από τον δεκαπέντε ετών *Richard Skrenta*. Ο δεκαπεντάχρονος *Richard* αποθήκευσε τον ιό του σε μία δισκέτα και την έδωσε σε φίλους και γνωστούς του. Πρέπει να σημειωθεί ότι οι υπολογιστές εκείνης της εποχής δε διέθεταν σκληρό δίσκο στο σύστημα τους, έτσι οι ανταλλαγές δισκετών μεταξύ χρηστών ήταν αρκετά συχνό φαινόμενο. Όταν ο υπολογιστής εκκινούσε στο σύστημα του τη μλουσμένη αυτή δισκέτα, ο ιός αντιγραφόταν από μόνος του σε οποιαδήποτε άλλη δισκέτα ήταν συνδεδεμένη στον υπολογιστή. Μετά από κάποιον αριθμό εκκινήσεων από

μολυσμένη δισκέτα, ο υπολογιστής εμφάνιζε ένα μήνυμα υπό μοροφή στίχων. Ο συγκεκριμένος ιός δεν είχε καταστροφικές προθέσεις και ο δημιουργός του τον έφτιαξε ως αστείο. Ο πρώτος ιός που εμφανίστηκε στους προσωπικούς υπολογιστές ήταν ο ιός Brain, ο οποίος δημιουργήθηκε στο Πακιστάν το *1986* από δύο αδερφούς *Basit* και *Amjad Farooq Alvi*. Ο συγκεκριμένος ιός πρόσβαλε τον τομέα εκκίνησης *(boot sector)* του σκληρού δίσκου. Από τότε έως σήμερα έχουν δημιουργηθεί και κυκλοφορήσει χιλιάδες είδη ιών, εκ των οποίων οι περισσότεροι προκαλούν φθορές στο υπολογιστικό σύστημα ή δίκτυο στο οποίο εισχωρούν. Στα πρώτα χρόνια του εικοστού πρώτου αιώνα εκτιμάται ότι με τη ραγδαία αύξηση του διαδικτύου δημιουργήθηκαν και ανακαλύφθηκαν πάνω από 700.000 ιοί. Ωστόσο αυτοί ταξινομούνται σε δύο μεγάλες κατηγορίες, ανάλογα με το σημείο του υλικού/λογισμικού που μολύνουν και ανάλογα με τον τρόπο που πραγματοποιούν τη μόλυνσή τους. Παρακάτω θα αναλύσουμε σχετικά πράγματα για αυτούς, καθώς και τρόπους δράσης, αντιμετώπισης αλλά και πως διαδίδονται.

```
00000000h: FA E9 4A 01 34 12 00 07 09 00 01 00 00 00 00 00 ;   J.4...........
00000010h: 57 65 6C 63 6F 6D 65 20 74 6F 20 74 68 65 20 20 ; Welcome to the
00000020h: 44 75 6E 67 65 6F 6E 20 20 20 20 20 20 20 20 20 ; Dungeon
00000030h: 28 63 29 20 31 39 38 36 20 42 72 61 69 6E 17 26 ; (c) 1986 Brain.&
00000040h: 20 41 6D 6A 61 64 73 20 28 70 76 74 29 20 4C 74 ;  Amjads (pvt) Lt
00000050h: 64 20 20 20 56 49 52 55 53 5F 53 48 4F 45 20 20 ; d   VIRUS_SHOE
00000060h: 52 45 43 4F 52 44 20 20 20 76 39 2E 30 20 20 20 ; RECORD   v9.0
00000070h: 44 65 64 69 63 61 74 65 64 20 74 6F 20 74 68 65 ; Dedicated to the
00000080h: 20 64 79 6E 61 6D 69 63 20 6D 65 6D 6F 72 69 65 ;  dynamic memorie
00000090h: 73 20 6F 66 20 6D 69 6C 6C 69 6F 6E 73 20 6F 66 ; s of millions of
000000a0h: 20 76 69 72 75 73 20 77 68 6F 20 61 72 65 20 6E ;  virus who are n
000000b0h: 6F 20 6C 6F 6E 67 65 72 20 77 69 74 68 20 75 73 ; o longer with us
000000c0h: 20 74 6F 64 61 79 20 2D 20 54 68 61 6E 6B 73 20 ;  today - Thanks
000000d0h: 47 4F 4F 44 4E 45 53 53 21 21 20 20 20 20 20 20 ; GOODNESS!!
000000e0h: 20 42 45 57 41 52 45 20 4F 46 20 54 48 45 20 65 ;  BEWARE OF THE e
000000f0h: 72 2E 2E 56 49 52 55 53 20 20 3A 20 5C 74 68 69 ; r..VIRUS  : \thi
00000100h: 73 20 70 72 6F 67 72 61 6D 20 69 73 20 63 61 74 ; s program is cat
00000110h: 63 68 69 6E 67 20 20 20 20 20 20 70 72 6F 67 72 ; ching      progr
00000120h: 61 6D 20 66 6F 6C 6C 6F 77 73 20 61 66 74 65 72 ; am follows after
00000130h: 20 74 68 65 73 65 20 6D 65 73 73 65 67 65 73 2E ;  these messeges.
00000140h: 2E 2E 2E 2E 20 24 23 40 25 24 40 21 21 20 8C C8 ; .... $#@%$@!!
00000150h: 8E D8 8E D0 BC 00 F0 FB A0 06 7C A2 09 7C 8B 0E ;     ?踳?|?|?
00000160h: 07 7C 89 0E 0A 7C E8 57 00 B9 05 00 BB 00 7E E8 ; .|?.|鄴.?.?~?
00000170h: 2A 00 E8 4B 00 81 C3 00 02 E2 F4 A1 13 04 2D 07 ; *.輾.  ..碠?.-.
00000180h: 00 A3 13 04 B1 06 D3 E0 8E C0 BE 00 7C BF 00 00 ; .?.?釬  ?|?.
00000190h: B9 04 10 FC F3 A4 06 B8 00 02 50 CB 51 53 B9 04 ; ?.  ??.P沃S?
000001a0h: 00 51 8A 36 09 7C B2 00 8B 0E 0A 7C B8 01 02 CD ; .Q?.|??.|?.?
000001b0h: 13 73 09 B4 00 CD 13 59 E2 E7 CD 18 59 5B 59 C3 ; .s.??Y瞽?Y[Y?
000001c0h: A0 0A 7C FE C0 A2 0A 7C 3C 0A 75 1A C6 06 0A 7C ; ?|  ?|<.u.?.|
000001d0h: 01 A0 09 7C FE C0 A2 09 7C 3C 02 75 09 C6 06 09 ; .?|  ?|<.u.?.
000001e0h: 7C 00 FE 06 0B 7C C3 00 00 00 00 32 E3 23 4D 59 ; |.?.|?...2?MY
000001f0h: F4 A1 82 BC C3 12 00 7E 12 CD 21 A2 3C 5F 0C 05 ; 襭 ?.~?.??_..
```

*Εικόνα 4 – Στιγμιότυπο κώδικα ίου Brain*

## 1.6 Λειτουργία 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 διαφορετικά στάδια, 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. Παρακάτω περιγράφονται τα στάδια μιας penetration testing δομής

- Στάδιο αναγνώρισης

  Στο στάδιο αυτό, το στάδιο της προετοιμασίας είναι το πιο σημαντικό στάδιο ενός pentest και επηρεάζει όλα τα επόμενα στάδια. Στην αναγνώριση συλλέγονται όσο το δυνατόν περισσότερες πληροφορίες για ένα σύστημα στόχο με τη βοήθεια μιας απλής Google αναζήτησης ή εργαλείων όπως το whois, nslookup, whois.domaintools.com κ.α.
- Στάδιο ανίχνευσης

  Το στάδιο της ανίχνευσης μπορεί να θεωρηθεί και ως μια επέκταση του σταδίου αναγνώρισης όπου τα στοιχεία που συλλέχτηκαν μπορούν να χρησιμοποιηθούν για την εύρεση περεταίρω πληροφοριών για το στόχο. Ο χρήστης βρίσκει πληροφορίες για τις ανοιχτές πόρτες, την τοπολογία του δικτύου, τον τύπο και το όνομα των υπηρεσιών που εκτελούνται στο στόχο, το λειτουργικό του σύστημα, τις εκδόσεις του λογισμικού που χρησιμοποιούνται και αν αυτές είναι ενημερωμένες. Απώτερος σκοπός είναι να εντοπιστούν πιθανές ευπάθειες στο σύστημαστόχο οι οποίες θα συμβάλλουν τελικά στο να οργανωθεί καλύτερα μια επίθεση.
- Στάδιο απόκτησης πρόσβασης

  Ο χρήστης προσπαθεί να εκμεταλλευτεί τις ευπάθειες που καταγράφθηκαν στο προηγούμενο στάδιο και να αποκτήσει προνόμια διαχειριστή, και κατ’ επέκταση,

πρόσβαση σε υπηρεσίες του συστήματος στόχου ή και σε γνωστοποίηση κωδικών κατηγορίας root για πλήρη έλεγχο του συγκεκριμένου συστήματος.

- Στάδιο διατήρησης πρόσβασης

  Έχοντας αποκτήσει πρόσβαση στο σύστημα-στόχος ο χρήστης προσπαθεί να την διατηρήσει ώστε να μη χάσει τα προνομία και να είναι εφικτός ο έλεγχος και χειρισμός του συστήματος στο μέλλον, ακόμα και χωρίς την εκμετάλλευση κάποιας ευπάθειας. Το στάδιο αυτό συνήθως περιλαμβάνει τη χρήση εργαλείων τα οποία αποκρυπτογραφούν κωδικούς πρόσβασης που έχουν υποκλαπεί κατά την αρχική πρόσβαση.

- Στάδιο αναφοράς

  Το τελευταίο στάδιο, και επίσης πολύ σημαντικό, είναι η σωστή καταγραφή των όσων έχουν παρατηρηθεί κατά τη διάρκεια του pentest. Επισημαίνονται όλα τα κρίσιμα και ευάλωτα σημεία του στόχου προς εκμετάλλευση, οι μέθοδοι που χρησιμοποιήθηκαν, και οι τρόποι με τους οποίους προτείνεται να βελτιωθεί η ασφάλεια του.

## 1.7 Τι θέλουμε να κάνουμε;

Το πεδίο εφαρμογής της παρούσας διατριβής είναι ο έλεγχος ευπαθειών όπου μπορεί να βρίσκονται στους διακομιστές των πληροφοριακών συστημάτων του Χαροκοπίου πανεπιστημίου, με χρήση τεχνικών *(penetration testing).* Καθημερινά μεγάλο πλήθος φοιτητών και χρηστών επισκέπτονται της διαδικτυακές υπηρεσίες όπου παρέχει το πανεπιστήμιο ώστε να παρακολουθήσουν μαθήματα, να εκτελέσουν έρευνα καθώς και να αντλήσουν διάφορες περεταίρω πληροφορίες σχετικά με το πανεπιστήμιο όπως έρευνα, συστήματα σπουδών κ.α. Όπως γίνεται αντιληπτό η χρήση και οι χρήστες ποικίλουν και έτσι οι παροχές αυτών των υπηρεσιών είναι διαφορετικές. Επιπλέον τα δεδομένα όπου έχουν πρόσβαση οι χρήστες είναι περιορισμένα αλλά σε αρκετές περιπτώσεις οι ίδιες οι υπηρεσίες έχουν πρόσβαση σε ευαίσθητα δεδομένα χρηστών της πανεπιστημιακής μονάδας. Σε κάθε περίπτωση, οι χρήστες θα πρέπει να έχουν πρόσβαση, μόνο στα απαραίτητα και διαθέσιμα για αυτούς δεδομένα και ανάλογα με την δικαιοδοσία αυτών. Για παράδειγμα οι επισκέπτες του ιστότοπου έχουν διαφορετικά δεδομένα πρόσβασης σε σχέση με τους φοιτητές της πανεπιστημιακής μονάδας. Επιπλέον πρέπει να υπάρχουν όλοι οι απαραίτητοι μηχανισμοί ώστε καμία κατηγορία χρηστών πλην του/των διαχειριστών να μην μπορούν να αποκτήσουν πρόσβαση σε δεδομένα πλην αυτών της δικαιοδοσίας τους και των δικαιωμάτων τους. Για το στόχο αυτό, γίνεται ανάλυση των υπηρεσιών της πανεπιστημιακής μονάδας, ώστε να εξακριβωθεί ο βαθμός ασφάλειας των υπηρεσιών αυτής καθώς και η ακεραιότητας αυτών. Η ανάλυση των υπηρεσιών ως προς πιθανές υπαρκτές και ρεαλιστικές ευπάθειες, αποτελεί το βασικό αντικείμενο της διατριβής τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο έρευνας.

## 1.8 Κύριες πήγες ευπαθειών

Οι ευπάθειες όπου παρουσιάζουν τα πληροφοριακά συστήματα τις περισσότερες φορές οφείλονται σε προγραμματιστικά λάθη κατά την ανάπτυξή και την δόμηση τους. Τέτοιου είδους ελαττώματα μπορούν να κατηγοριοποιηθούν ακόλουθα:

- 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.
- Ρυθμίσεις**.**

  Οι 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.

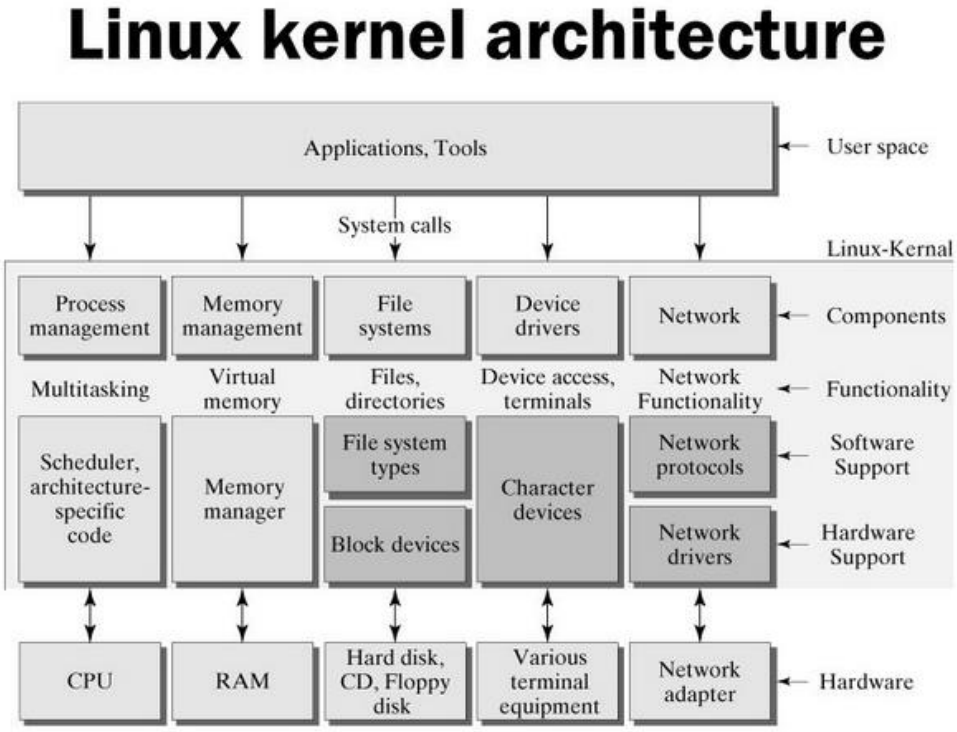
Από τα παραπάνω γίνεται εμφανές ότι οι ευπάθειας μπορούν να προέρχονται από διάφορες πηγές, ενώ καθημερινά εμφανίζονται και νέα, για κάθε ένα από τα οποία υπάρχει και το ανάλογο exploit που μπορεί να οδηγήσει σε μία πετυχημένη επίθεση. Οι εταιρίες ανάπτυξης λογισμικού κάθε τόσο διανέμουν μέσω του διαδικτύου διάφορες διορθώσεις σε ευπάθειες 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.

# Κεφάλαιο 2 – Ανάλυση υπηρεσιών και εφαρμογών

## 2.1 Διαχείριση Linux.

Το Linux είναι μία από τις δημοφιλέστερες εκδόσεις του λειτουργικού συστήματος UNIX. Είναι ανοικτού κώδικα και ο κώδικας του είναι ελεύθερα διαθέσιμος. Το Linux σχεδιάστηκε ώστε να είναι συμβατό με το UNIX. Έτσι η λειτουργικότητα του είναι παρόμοια.

*Συστατικά του λειτουργικού συστήματος Linux*




*Εικόνα 5 - Δομή αρχιτεκτονικής Λ.Σ Linux*

Λειτουργικό σύστημα Linux θα μπορούσαμε να το χωρίσουμε σε τρία κυρίως μέρη:

- **Kernel (Πυρήνας)**

  Ο πυρήνας είναι το βασικό μέρος του Linux. Είναι υπεύθυνος για όλες τις σημαντικές δραστηριότητες του λειτουργικού συστήματος. Αποτελείται από διάφορες ενότητες και αλληλοεπιδρά άμεσα με το υλικό. Ο πυρήνας παρέχει την απαιτούμενη διεπαφή έτσι ώστε να κρύψει λειτουργίες χαμηλού επιπέδου του hardware που αφορούν λειτουργίες του συστήματος ή των εφαρμογών

- **System Library (Βιβλιοθήκες συστήματος)**

  Οι βιβλιοθήκες του συστήματος είναι ειδικές λειτουργίες ή προγράμματα που χρησιμοποιούν άλλα προγράμματα με σκοπό να αποκτήσουν πρόσβαση σε λειτουργίες του πυρήνα. Αυτές οι βιβλιοθήκες υλοποιούν τις περισσότερες από τις λειτουργίες του λειτουργικού συστήματος.

- **System Utility**

  Βοηθητικά προγράμματα του συστήματος που είναι υπεύθυνα για εξειδικευμένες εργασίες σε διαφορετικά επίπεδα του λειτουργικού.

### 2.1.1 Λογισμικά ανάλυσης

Για την ανάλυση του πληροφοριακού συστήματος χρησιμοποιήσαμε αυτοματοποιημένες πλατφόρμες ανάλυσης ευπαθειών *(frameworks)* οι οποίες ειδικεύονται στην ανάλυση ψηφιακών συστημάτων και εμφανίζουν τα αποτελέσματα με συγκεντρωτικό τρόπο ώστε να βοηθήσουν το χρήστη στην ανάλυση-επεξήγηση των αποτελεσμάτων καθώς και την αναζήτηση λεπτομέρειών με εύκολο τρόπο. Παρακάτω θα αναφέρουμε λίγα λόγια για όλα τα εργαλεία που χρησιμοποιήθηκαν καθόλα τη διάρκεια της ανάλυσης του πληροφοριακού συστήματος.

#### *Kali Linux*

Ως λειτουργικό σύστημα *(Operating System)* στην παρούσα διατριβή χρησιμοποιήθηκε το *Kali Linux* για την εκτέλεση διαφόρων ειδών αναλύσεις προς το πληροφοριακό σύστημα. Το *Kali Linux* αποτελεί δημιουργία της ομάδας *Offensive Security* και έχει προ-εγκατεστημένα εξειδικευμένα εργαλεία για την ανάλυση και την εισχώρηση σε πληροφοριακά συστήματα. Επιπλέον αποτελεί ένα ελεύθερο λογισμικό *(Open Source)* και βρίσκεται στην επίσημή σελίδα του *Kali Linux*. Δίνει την δυνατότητα στο χρήστη την επιλογή με τι θέλει να δουλέψει όπως γραφικό περιβάλλον *(gui),* γραμμή εντολών *(terminal)* πάνω στην έκδοση για λογισμικά Unix Gnome 3.

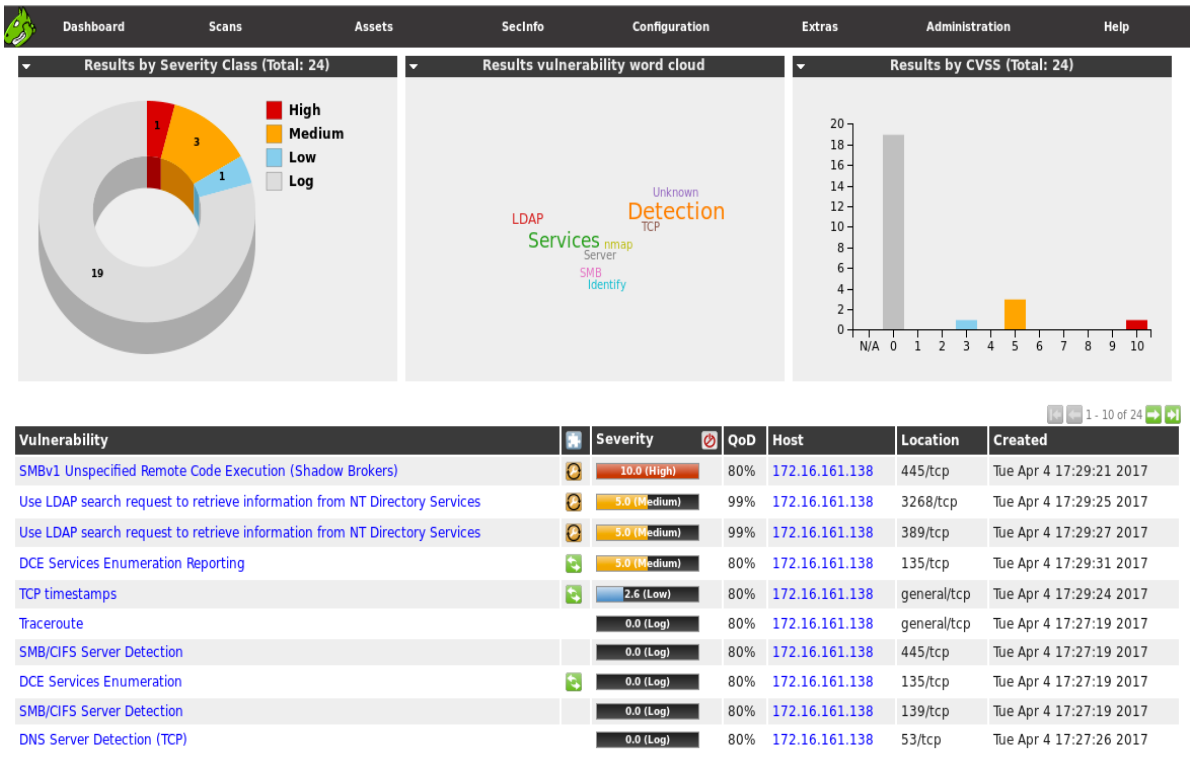

*Εικόνα 6 - Περιβάλον εργασίας Λ.Σ Kali Linux*

## *OpenVas*


*Εικόνα 7 – Λογότυπο Openvas*

Το OpenVas είναι μία αυτοματοποιημένη και ευκολόχρηστη πλατφόρμα για την ανάλυση συγκεκριμένων ηλεκτρονικών διευθύνσεων καθώς και πλήθος ηλεκτρονικών διευθύνσεων ταυτόχρονα και εμφάνιση των αποτελεσμάτων σε ομαδοποιημένη μορφή. Το συγκεκριμένο εργαλείο έχει τη δική του βάση δεδομένων η οποία ενημερώνεται από τους δημιουργούς της για την κρισιμότητα - σοβαρότητα του κάθε αποτελέσματος. Επίσης παρέχει στα αποτελέσματα ,ποσοστά επιτυχίας κάποιας επίθεσης σε τρύπες

ασφαλείας που έχει βρει καθώς και προτεινόμενους τρόπους αντιμετώπισης με παραπομπές σε επίσημές ηλεκτρονικές διευθύνσεις που παρέχουν εξειδικευμένη βοήθεια για κάθε τρύπα των συστημάτων. Είναι προγραμματισμένο αρχικά σε γλώσσα *NASL (Nessus Attack Scripting Language)* και από την έκδοση 2.0 με την *OVAL (Open Vulnerability and Assessment Language)*. Επιπλέον δίνεται η δυνατότητα στο χρήστη να δημιουργήσει δικά του "scripts" και να τα ενσωματώσει στο εργαλείο αυτό με ελάχιστες γνώσεις προγραμματισμού. Ο τρόπος χρησιμοποίησης του εργαλείου αυτού γίνεται μέσω του Internet Browser με τοπική διεύθυνση που θέτει ο χρήστης κατά τη παραμετροποίηση του προγράμματος. Διατίθεται δωρεάν από την επίσημή ηλεκτρονική διεύθυνση καθώς και ο κώδικας του καθώς έχει την ιδιότητα του ανοιχτού λογισμικού *(open source)*.



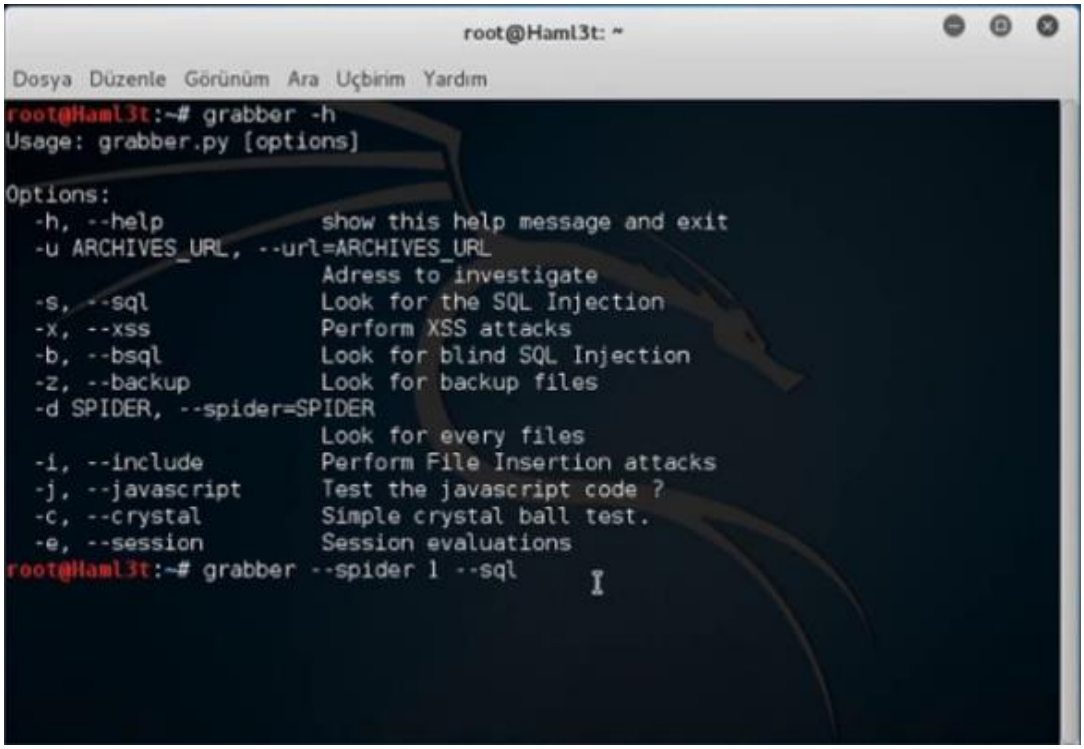

*Εικόνα 8 - Περιβάλον εργασίας OpenVas*

### *Grabber*

Το *Grabber* είναι ένας ανοιχτού κώδικα web server scanner ο όποιος πραγματοποιεί εκτενείς ελέγχους σε web servers με σκοπό τον εντοπισμό πιθανών ευπαθειών. Αποτελεί ένα αρκετά απλό σε δομή και χρήση εργαλείο παρόλο που σε δείκτες ταχύτητας είναι σχετικά αργό σε σχέση με τα εργαλεία όπου αναλύθηκαν προηγουμένως. Το *Grabber* γράφτηκε από τον *Romain Gaucher* και κυκλοφόρησε πρώτη φορά το Δεκέμβριο του 2001 ως η έκδοση 1.00 *Beta* ακολουθούμενο από την νεότερη έκδοση του, την 1.01. Μέσα σε διάστημα 2 ετών το *Nikto* έγινε ένα από τα πιο δημοφιλή διαθέσιμα εργαλεία ανίχνευσης ευπαθειών σε έναν web server. Έπειτα, το Νοέμβριο του 2007 κυκλοφόρησε η έκδοση 2.0, η οποία βελτιώθηκε σταδιακά.

Συνοπτικά, τα βασικότερα χαρακτηριστικά και λειτουργιές του είναι:

- Cross-Site Scripting.
- SQL Injection *(υπάρχει επίσης μια ειδική ενότητα Blind SQL Injection).*
- File Inclusion.
- Έλεγχος αντιγράφων ασφαλείας.
- Απλός έλεγχος χρήσης *AJAX* και πιθανών ευπαθειών της χρήσης αυτού.
- Υβριδική ανάλυση και *Crystal ball* testing για εφαρμογές *PHP* με χρήση *PHP-SAT*
- Αναλυτής πηγαίου κώδικα JavaScript: Αξιολόγηση της ποιότητας / ορθότητας του JavaScript με JavaScript Lint
- Δημιουργία αρχείου [session_id, time (t)] για την επόμενη ανάλυση στατιστικών
- στοιχείων.




Εικόνα 9 - Περιβάλον εργασίας Grabber

### *Nikto*


*Εικόνα 10 - Λογότυπο Nikto*

Το *Nikto* είναι ένας ανοιχτού κώδικα web server scanner ο όποιος πραγματοποιεί εκτενείς ελέγχους σε web servers με σκοπό τον εντοπισμό πιθανών ευπαθειών. Πιο συγκεκριμένα, σε αυτούς τους διεξοδικούς ελέγχους συμπεριλαμβάνονται ο έλεγχος για μη ενημερωμένες εκδόσεις λογισμικού, που μπορούν να αποτελέσουν ένα είδος ευπάθειας, ο έλεγχος πάνω από 6500 επικίνδυνων αρχείων/*CGIs*, ο εντοπισμός του εγκατεστημένου λογισμικού στον *web server* κ.λπ. Το εργαλείο αυτό είναι γραμμένο σε γλώσσα Perl και μπορεί εύκολα να επεκταθεί και να προσαρμοστεί. Με λίγη γνώση

της γλώσσας και έναν text editor, μπορεί κανείς εύκολα να μελετήσει και να τροποποιήσει τον πηγαίο κώδικα ώστε να τον φέρει στα μέτρα του. Σαρώνει τον server σε πολύ μικρό χρονικό διάστημα ελέγχοντας την βάση δεδομένων του για ευπάθειες οι οποίες είναι ενημερωμένες με τις τελευταίες λεπτομέρειες.

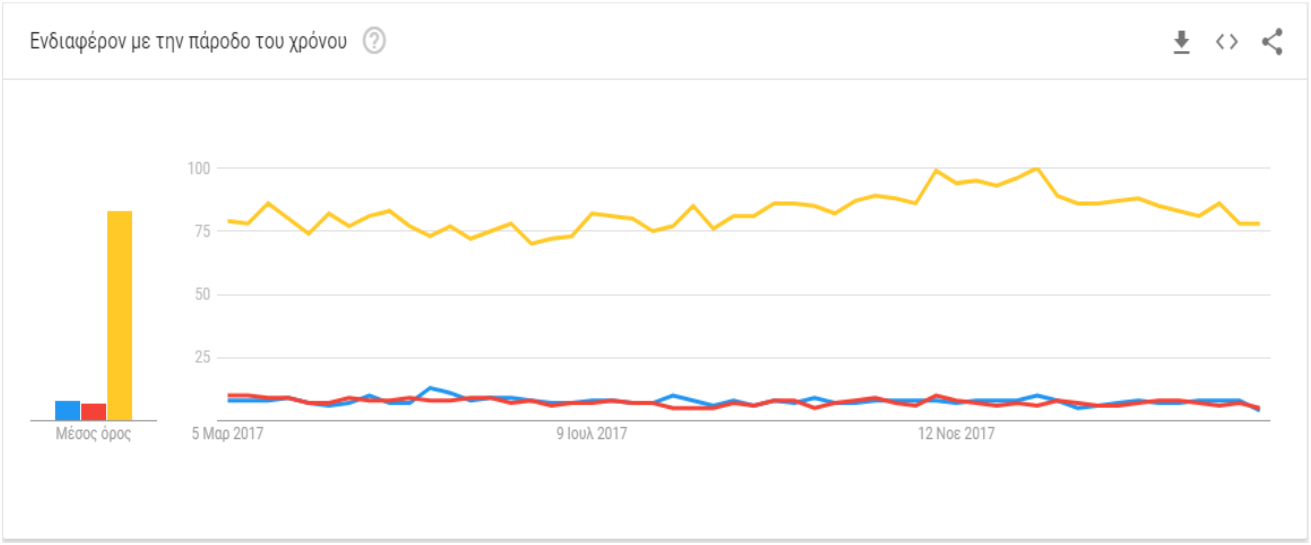
Συνοπτικά, τα βασικότερα χαρακτηριστικά και λειτουργιές του είναι:

- Ελέγχει για μη ενημερωμένες εκδόσεις λογισμικού σε πολλούς server.
- Ανιχνεύει προγράμματα και αρχεία που δεν είναι ασφαλή.
- Πλήρης HTTP Proxy υποστήριξη.
- Απαρίθμηση username Apache.
- Υποστήριξη SSL (Secure Socket Layer).
- Brute forcing.
- Είναι γρήγορο και αποτελεσματικό .

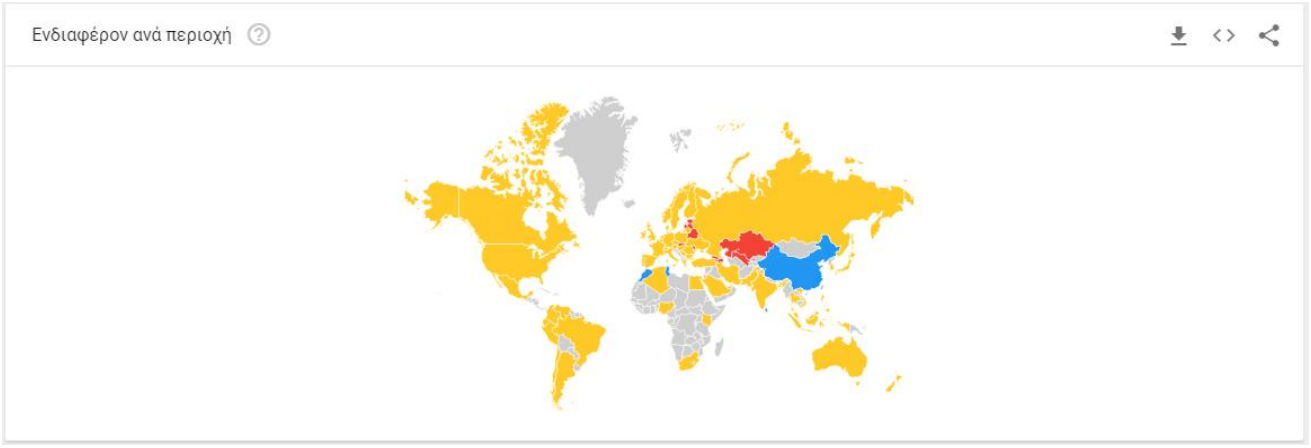
### 2.1.2 Σύγκριση και επιλογή

Μετά από σχετική έρευνα επιλογής των εργαλείων λογισμικού που πρόκειται να χρησιμοποιηθούν για τον έλεγχο ευπαθειών των διακομιστών .Παρακάτω βλέπουμε ένα συγκριτικό ενδιαφέρον με την πάροδο του χρόνου και παράλληλα το ενδιαφέρον που υπάρχει σε κάθε περιοχή ξεχωριστά για το κάθε λογισμικό.

- OpenVas
- Nikto
- Grabber




*Εικόνα 11 –Διάγραμμα Ενδιαφέροντος με την πάροδο του χρόνου*




*Εικόνα 12 – Χάρτης κάλυψης ενδιαφέροντος ανά περιοχή*

### 2.1.3 Γλώσσα Προγραμματισμού NASL

Η NASL αποτελεί μια γλώσσα προγραμματισμού στην όποια έχουν γραφτεί αρκετά vulnerability scanners επιτρέπει σε αναλυτές ασφαλείας να δημιουργήσουν γρήγορα τα δικά τους plugins για ελέγχους ευπαθειών. Το αποτέλεσμα είναι πως οι ομάδες ασφάλειας μπορούν εύκολα να προσθέσουν τις δικές τους γνώσεις στους ελέγχους τους, δημιουργώντας χειροποίητα τεστ ευπαθειών. Επίσης, επιτρέπει σε in-house ομάδες ασφαλείας να κατασκευάσουν ελέγχους ευπαθειών για τα πρωτόκολλα και τις υπηρεσίες που είναι μοναδικά στο κάποιο δίκτυο. Η NASL μοιάζει αρκετά με τη γλώσσα C. Είναι ειδικά σχεδιασμένη για τον τομέα της ασφάλειας καθώς επικοινωνεί μόνο με τον χρήστη που της δόθηκε ως παράμετρος και δεν εκτελεί καμία εντολή τοπικά. Έχοντας αυτό υπόψιν, είναι σχεδόν απίθανο ένα plugin να εκτελέσει ανεπιθύμητες διαδικασίες. Επίσης, η NASL είναι σχεδιασμένη έτσι ώστε να μοιράζει πληροφορίες μεταξύ των ελέγχων ασφαλείας με την χρήση των knowledge bases.

## 2.2 QoD *(Quality of Detection)*

Με τον όρο *QoD* αναφερόμαστε στην ποιότητα αξιολόγησης της ανίχνευσης ενός εργαλείου *offensive security.* Μια αξιολόγηση ευπάθειας, πρέπει να είναι τόσο σε θέση να παρέχει μια αναφορά σχετικά με το είδος της ευπάθειας, όσο και το πόσο επικίνδυνη για την εφαρμογή είναι η συγκεκριμένη ευπάθεια. Οι κατηγορίες ενός *QoD* καθώς και η γενικές λεπτομερείς αυτών, βάση των οποίων λειτουργεί η συγκεκριμένη διατριβή, διακρίνονται στο ακόλουθο πίνακα.

| **Κατηγορία QoD** | **Περιγραφή** |
|---|---|
| 100% exploit | Η ανίχνευση έγινε μέσω αξιοποίησης της αδυναμίας συστήματος και είναι επιβεβαιωμένη η δυνατότητα αξιοποίησης της για την είσοδο στο σύστημα επίθεσης. |

| | |
|---|---|
| 99% remote_vul | Αναφέρεται σε ευπάθειες οι οποίες επιτρέπουν τον απομακρυσμένο ενεργό έλεγχο όπως<br>• Code execution *(εκτέλεση κώδικα).*<br>• Traversal attack *(επίθεση διέλευσης).*<br>• SQL injection *(έγχυση SQL).*<br>καθώς και πληθώρα άλλων επιθέσεων όπου η ανταπόκριση δείχνει καθαρά τη παρουσία της αδυναμίας συστήματος. |
| 98% remote_app | Αναφέρεται σε ευπάθειες οι οποίες επιτρέπουν τον απομακρυσμένο ενεργό έλεγχο όπως<br>• Code execution *(εκτέλεση κώδικα).*<br>• Traversal attack *(επίθεση διέλευσης).*<br>• SQL injection *(έγχυση SQL).*<br>καθώς και πληθώρα άλλων επιθέσεων όπου η ανταπόκριση δείχνει καθαρά τη παρουσία της αδυναμίας συστήματος. |
| 97% package | Αναφέρεται σε ευπάθειες με πιστοποιημένα πακέτα επιβεβαιωμένα για συστήματα Linux. |
| 97% registry | Αναφέρεται σε ευπάθειες με πιστοποιημένα πακέτα επιβεβαιωμένα για συστήματα Windows. |

| | |
|---|---|
| 95% remote_active | Αναφέρεται σε ευπάθειες οι οποίες επιτρέπουν τον απομακρυσμένο ενεργό έλεγχο όπως<br>• Code execution *(εκτέλεση κώδικα).*<br>• Traversal attack *(επίθεση διέλευσης).*<br>• SQL injection *(έγχυση SQL).*<br>καθώς και πληθώρα άλλων επιθέσεων όπου η ανταπόκριση δείχνει καθαρά τη παρουσία της αδυναμίας συστήματος. Κάτω από σπάνιες περιπτώσεις είναι δυνατόν να είναι λάθος εκτίμηση. |
| 80% remote_banner | Αναφέρεται σε ευπάθειες όπου υπάρχουν απομακρυσμένοι έλεγχοι τίτλου-banner εφαρμογών δείχνουν την έκδοση της εφαρμογής. Πολλά προϊόντα το έχουν αυτό. |
| 80% executable_version | Αναφέρεται σε ευπάθειες όπου υπάρχουν πιστοποιημένες εκτελέσιμές εκδόσεις οι οποίες ελέγχουν για υπάρξει συστημάτων Linux ή Windows όπου η εφαρμογή προσφέρει επιπλέον πακέτα στις εκδόσεις. |
| 70% remote_analysis | Αναφέρεται σε ευπάθειες όπου υπάρχουν και προσφέρουν δυνατότητα για απομακρυσμένο έλεγχο που κάνει ανάλυση, αλλά δεν είναι πάντα αξιόπιστος. |

| | |
|---|---|
| 50% remote_probe | Αναφέρεται σε ευπάθειες όπου υπάρχουν και προσφέρουν δυνατότητα για έλεγχο σε ενδιάμεσα συστήματα *(firewalls)* και δίνατε να προσποιηθούν απαντώντας θετικά ώστε να μην είναι ξεκάθαρο ότι απάντησε η εφαρμογή η ίδια. |
| 30% remote_banner_unreliable | Αναφέρεται σε ευπάθειες όπου υπάρχουν απομακρυσμένοι έλεγχοι τίτλου-banner εφαρμογών αλλά δεν δείχνουν την έκδοση της εφαρμογής. Πολλά προϊόντα το έχουν αυτό. |
| 1% general_note | Γενικές πληροφορίες για δυνητικές αδυναμίες χωρίς να υπάρχουν παρούσες εφαρμογές. |

## 2.3 Τρίγωνο CIA

Η ασφάλεια ενός πληροφοριακού συστήματος περιλαμβάνει τις διαδικασίες και τις πολιτικές που ακολουθούνται απο τον διαχειριστή του δικτύου ώστε να αποφευχθεί η μη εξουσιοδοτημένη πρόσβαση, η συνεχής εξουσιοδοτημένη διαθεσιμότητα, και η παραποίηση των δεδομένων ενός δικτύου. Σύμφωνα με τη NIST, η ασφάλεια υπολογιστών περιλαμβάνει 3 βασικές αρχές :

- Ακεραιότητα *(Integrity).*
- Διαθεσιμότητα *(Availability)*.
- Εμπιστευτικότητα *(Confidentiality).*

Αυτά τα 3 βασικά σημεία της ασφάλειας πληροφοριακών συστημάτων συχνά αναφέρονται ως τρίγωνο CIA *(CIA Triad).*

### *Ακεραιότητα*

Η ακεραιότητα μπορεί να οριστεί γενικότερα ως η απαίτηση να είναι τα πράγματα όπως πρέπει να είναι. Στην πληροφορική, ακεραιότητα σημαίνει πρόληψη μη εξουσιοδοτημένης μεταβολής πληροφοριών, δηλαδή, πρόληψη από μη εξουσιοδοτημένη εγγραφή ή διαγραφή, συμπεριλαμβανομένης και της μη εξουσιοδοτημένης δημιουργίας δεδομένων. Επομένως, σημαίνει ότι η μετατροπή, διαγραφή, και δημιουργία των δεδομένων ενός υπολογιστικού συστήματος, γίνεται μόνο από εξουσιοδοτημένα μέρη.

### *Διαθεσιμότητα*

Διαθεσιμότητα ονομάζεται η ιδιότητα του να είναι προσπελάσιμες και χωρίς αδικαιολόγητη καθυστέρηση οι υπηρεσίες ενός πληροφοριακού συστήματος (ΠΣ) όταν τις χρειάζεται μια εξουσιοδοτημένη οντότητα. Αυτό σημαίνει ότι οι εξουσιοδοτημένοι χρήστες των υπολογιστικών συστημάτων και των επικοινωνιακών μέσων δεν αντιμετωπίζουν προβλήματα άρνησης εξυπηρέτησης (denial of service) όταν επιθυμούν να προσπελάσουν τους πόρους του συστήματος. Η διαθεσιμότητα καλύπτει περιοχές πέρα από το φυσικό σκοπό της ασφάλειας. Για παράδειγμα, ένα μεγάλο μέρος της τεχνολογίας που απαιτείται για τη διασφάλιση της διαθεσιμότητας προέρχεται από

άλλες περιοχές, όπως fault – tolerant computing. Για τους σκοπούς της ασφάλειας, μας απασχολεί βασικά η παρεμπόδιση κακόβουλων επιθέσεων που αποσκοπούν στο να παρακωλύσουν την πρόσβαση των νόμιμων χρηστών σε ένα πληροφοριακό σύστημα. Αυτές οι επιθέσεις ονομάζονται επιθέσεις άρνησης παροχής υπηρεσιών (denial of service attacks). Η άρνηση παροχής υπηρεσιών σημαίνει παρεμπόδιση της εξουσιοδοτημένης προσπέλασης πληροφοριών και πόρων ή πρόκληση καθυστέρησης των λειτουργιών που είναι κρίσιμες στο χρόνο (time - critical). Η αντιμετώπισή τους αποσκοπεί στο να υπερνικήσει την σκόπιμη (που προκαλείται από κακόβουλα μέρη) παρά τυχαία απώλεια της διαθεσιμότητας. Ένα παράδειγμα επίθεσης άρνησης παροχής υπηρεσιών είναι οι επιθέσεις «πλημμύρας» στο διαδίκτυο, όπου ο επιτιθέμενος κατακλύζει έναν εξυπηρετητή στέλνοντάς του έναν τεράστιο αριθμό αιτήσεων σύνδεσης. Παρόλο που η διαθεσιμότητα συχνά αναδεικνύεται στο πλέον σημαντικό χαρακτηριστικό της ασφάλειας, εντούτοις λίγοι μηχανισμοί υπάρχουν για να βοηθήσουν στην υποστήριξή της.

### *Εμπιστευτικότητα*

Σε πολλές περιπτώσεις της καθημερινής ζωής οι έννοιες της ασφάλειας και της εμπιστευτικότητας σχεδόν ταυτίζονται, όπως για παράδειγμα στα στρατιωτικά περιβάλλοντα όπου η ασφάλεια έχει τη σημασία του να κρατούνται μυστικές οι πληροφορίες. Η εμπιστευτικότητα σημαίνει πρόληψη μη εξουσιοδοτημένης αποκάλυψης πληροφοριών, δηλαδή πρόληψη από μη εξουσιοδοτημένη ανάγνωση. Επομένως σημαίνει ότι τα δεδομένα ενός υπολογιστικού συστήματος, καθώς και τα διακινούμενα μεταξύ των υπολογιστών δεδομένα, αποκαλύπτονται μόνο σε εξουσιοδοτημένα άτομα. Αυτό αφορά όχι μόνο την προστασία από μη εξουσιοδοτημένη αποκάλυψη των δεδομένων αυτών καθαυτών αλλά ακόμη και από το γεγονός ότι τα δεδομένα απλώς υπάρχουν. Έτσι για παράδειγμα, το γεγονός ότι κανείς έχει φάκελο εγκληματία είναι συχνά το ίδιο σημαντικό όπως και οι λεπτομέρειες για το έγκλημα που διαπράχθηκε.


*Εικόνα 13 – Σχηματικό Τριγώνου CIA*

### 2.3.1 Antivirus

Τα λογισμικά antivirus αποτελούν τα ευρέος χρησιμοποιούμενα λογισμικά προστασίας υπολογιστών και υπολογιστικών συστημάτων, όπου χρησιμοποιούνται για την πρόληψη, ανίχνευση και αφαίρεση κακόβουλων δεδομένων και λογισμικών, τα οποία έχουν εισχωρήσει *(αλλά δεν εκτελούνται απαραίτητα)* με κάποιον τρόπο στο λειτουργικό σύστημα. Τα λογισμικά antivirus αναπτύχθηκαν με σκοπό να αντιμετωπίζουν και να απομακρύνουν τους ιούς των υπολογιστών, όμως με την πάροδο του χρόνου και τη ραγδαία αύξηση του διαδικτύου, τα κακόβουλα λογισμικά πολλαπλασιάστηκαν και έτσι τα λογισμικά προστασίας άρχισαν να παρέχουν εντατικότερη ασφάλεια για όλο το φάσα των πραγματικών απειλών στα πληροφοριακά και υπολογιστικά συστήματα που συναντάμε στην σήμερον ημέρα. Αναλυτικότερα, ένα λογισμικό προστασίας προστατεύσει ένα υπολογιστικό σύστημα από τις ακόλουθες βασικές κατηγορίες απειλών:

- **Adwares**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο απαρτίζεται από διαφημίσεις που υποστηρίζονται από κάποιο λογισμικό πακέτο που καθιστά

αυτόματα διαφημίσεις προκειμένου να δημιουργηθούν έσοδα για το συγγραφέα του πακέτου.

- **Backdoors**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο είναι σε θέση να παρακάμπτει την πραγματική ταυτότητα και φύση του. Οι κατηγορία λογισμικών *backdoor* χρησιμοποιείται συχνά για την απόκτηση μη εξουσιοδοτημένης απομακρυσμένης πρόσβασης σε έναν υπολογιστή.

- **Browser hijackers**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού που τροποποιεί *(χωρίς την άδεια του χρήστη)* τις ρυθμίσεις του εκάστοτε φυλλομετρητή που χρησιμοποιεί ο χρήστης για την περιήγησή του στο διαδίκτυο. Συνήθως αυτό το λογισμικό προσθέτει την παρουσία από ανεπιθύμητα διαφημιστικά μμηνύματα.

- **Keyloggers**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο επιτρέπει την καταγραφή της πληκτρολόγησης, δηλαδή τη δράση του χρήστη πάνω στο πληκτρολόγιο καθώς εν άγνοιά του οι ενέργειές του βρίσκονται υπό διαρκή παρακολούθηση.

- **Ransomwares**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού που περιορίζει την πρόσβαση στο μολυσμένο υπολογιστή και απαιτεί από τον χρήστη να καταβάλει χρηματικό ποσό στον φορέα του antivirus για να αφαιρεθεί ο περιορισμός αυτός.

- **Rootkits.**

  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο έχει σχεδιαστεί για να επιτρέπει την πρόσβαση σε έναν υπολογιστή εφόσον δεν επιτρέπεται λόγω μη εξουσιοδοτημένου χρήστη αφού ταυτόχρονα καμουφλάρεται η ύπαρξή του από άλλο λογισμικό.

- **Spywares**

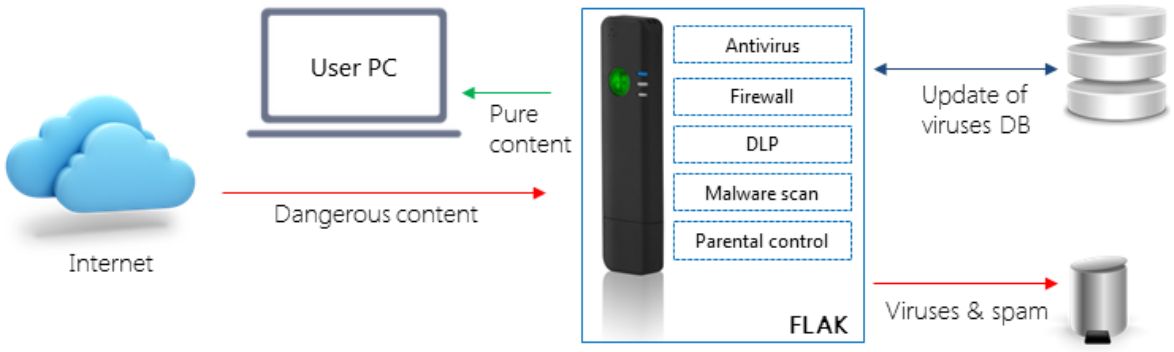
  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο έχει ως στόχο να συγκεντρώσει πληροφορίες για ένα πρόσωπο εν άγνοιά του. Τα συγκεκριμένα

λογισμικά έχουν τη δυνατότητα να στέλνουν τις πληροφορίες αυτές σε άλλο φορέα χωρίς την έγκριση του χρήστη.

- **Trojan horses**

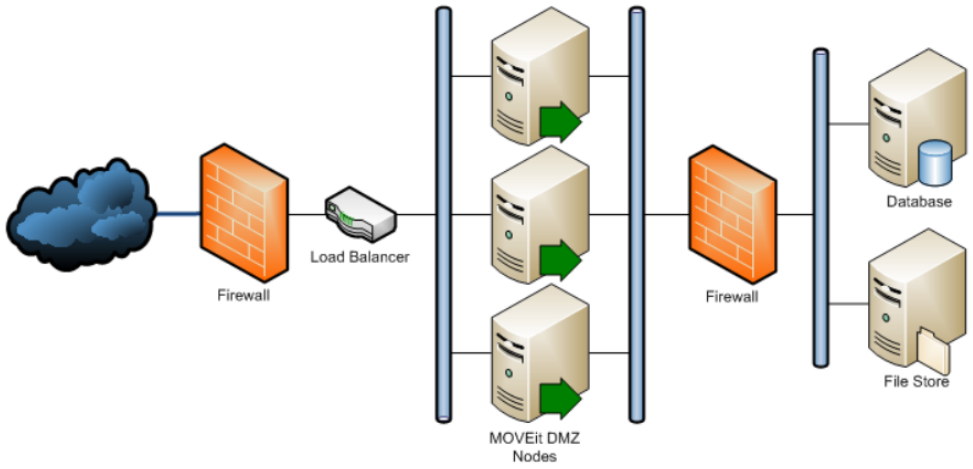
  Αποτελεί μια κατηγορία ανεπιθύμητου λογισμικού το οποίο είναι σχεδιασμένο ώστε για να φανεί χρήσιμο και ενδιαφέρον προς το χρήστη, προκειμένου αυτός να πειστεί και να το εγκαταστήσει.

Ορισμένα antivirus περιλαμβάνουν προστασία και σε άλλες απειλές που ενδέχεται να μμολύνουν ένα σύστημα, μία από αυτές είναι και τα κακόβουλα *URLs* (σύνδεσμοι).




*Εικόνα 14 - Αρχιτεκτονική δομή λογισμικού antivirus*

### 2.3.2 Firewall *(Τοίχος Προστασίας)*




*Εικόνα 15 - Αρχιτεκτονική δομή λογισμικού firewall*

Μια αποτελεσματική και ευρέως χρησιμοποιούμενη λύση για την προστασία στο διαδίκτυο, αποτελεί η εγκατάσταση και εφαρμογή μιας ασφαλούς *gateway (πύλης δικτύου),* η οποία ονομάζεται firewall και συναντάται τοποθετημένη, να λειτουργεί, ανάμεσα στο εσωτερικό δίκτυο και στο διαδίκτυο. Το *firewall* έχει ως κύριο στόχο λειτουργίας το να αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση μεταξύ δικτύων, δηλαδή να παρέχει προστασία στο εσωτερικό δίκτυο μιας τοπολογίας, όπως το δίκτυο υπολογιστών ενός υπολογιστικού κέντρου, από το υπόλοιπο διαδίκτυο. Αν η εγκατάσταση αυτή διαθέτει εγκαταστημένο και λειτουργικό κάποιο μηχανισμό *firewall*, τότε πρέπει να ληφθούν αποφάσεις για το τι πρόκειται να εισαχθεί πέρα από αυτό. Οι αποφάσεις αυτές πρέπει να αναδεικνύονται κατ' ευθείαν από την πολιτική ασφαλείας. Μεγάλη πιθανότητα υπάρχει να περιοριστούν οι διαθέσιμες επιλογές εγκατάστασης των διαδικτυακών υπηρεσιών όπως για παράδειγμα οι γνωστές *UDP, HTTP, FTP, DNS* κ.α. Μια από τις βασικές λειτουργιές του *firewall* που διαρκώς βρίσκεται σε εφαρμογή είναι η εξέταση των IP πακέτων που ταξιδεύουν μμεταξύ του *internet* και του εσωτερικού δικτύου. Το *firewall* υλοποιεί μια πολιτική ασφαλείας με την οποία καθορίζονται οι ισορροπίες ανάμεσα στην διατήρηση της ασφάλειας και στην εύχρηστη διαχείριση. Αν κάποια ενέργεια δεν είναι επιτρεπτή σύμφωνα με την πολιτική του *firewall*, τότε οφείλει να διασφαλίζει την αποτυχία κάθε προσπάθειας εκτέλεσης της ενέργειας αυτής. Για

παράδειγμα, κάθε ενέργεια του ηλεκτρονικού ταχυδρομείου προς συναλλαγή δεδομένων και πληροφοριών μμεταξύ χρηστών, εξετάζεται αν μπορεί να γίνει αποδεκτή σύμφωνα με τις προκαθορισμένες πολιτικές ασφαλείας ώστε να εκτελεστεί η συγκεκριμένη ενέργεια με επιτυχία. Επιπρόσθετα το *firewall* είναι σε θέση να απομονώνει και αναλύει τα δεδομένα αλλά και τα πακέτα πληροφοριών.

### 2.3.3 Επίπεδα προστασίας πληροφοριακού συστήματος

Με βάση όσων έχουμε αναφέρει μέχρι στιγμής, έχει γίνει κατανοητή η σημασία των πληροφοριακών συστημάτων σήμερα καθώς και η αναγκαία προστασία τους από κάθε είδους απειλή. Για να κατηγοριοποιήσουμε την προστασία και την ασφάλεια ώστε να μπορούμε να παρακολουθούμε τις αδυναμίες και να βρίσκουμε λύσεις αποφυγής απωλειών, είναι εφικτό να δημιουργήσουμε ορισμένα επίπεδα. Τα συγκεκριμένα επίπεδα ακούνε στον αριθμό τέσσερα και είναι τα εξής:

- Φυσική ασφάλεια

  Για τη φυσική ασφάλεια καταλαβαίνουμε ότι εννοείται η αντιμετώπιση δυσμενών συνθηκών. Για την ορθή αντιμετώπιση και αποφυγή αυτών, βασιζόμαστε στον κατάλληλο σχεδιασμό του κτηρίου, στη σωστή εκπαίδευση του προσωπικού και την παροχή μηχανισμών ασφαλείας, όπως για παράδειγμα των εξοπλισμό πυρόσβεσης. Αναγκαίο καθήκον είναι και η συστηματική συντήρηση των ηλεκτρικών εγκαταστάσεων. Επίσης, χρήσιμή φαίνεται να είναι και η ύπαρξη γεννήτριας για την παροχή ηλεκτρικής ενέργειας ή το γνωστό σύστημα ασταμάτητης παροχής τάσεως *(UPS),* ώστε να αποφεύγονται απώλειες κατά τη διάρκεια πτώσης της τάσης του ηλεκτρικού ρεύματος. Όμως, τα περισσότερα από τα παραπάνω παραλείπονται για οικονομικούς λόγους.
- Ασφάλεια λειτουργικών συστημάτων. Το σημαντικότερο σημείο ενός πληροφοριακού συστήματος είναι το λειτουργικό σύστημα *(operating system).* Λειτουργικό σύστημα ενός υπολογιστή ορίζεται το προϊόν λογισμικού που ελέγχει τα ήδη εκτελεσμένα προγράμματα και παρέχει διάφορες υπηρεσίες όπως

τη διαχείριση μνήμης, ελέγχου εισόδου-εξόδου, μεταγλώττισης κ.α. Κάποιες από τις ιδιότητες των ΛΣ είναι οι ακόλουθες: αδιαφάνεια, ακεραιότητα, αξιοπιστία, αποδοτικότητα, ασφάλεια, γενικότητα, διαθεσιμότητα, ευελιξία, ευκινησία, επεκτασιμότητα, ευχρηστία, συντηρητικότητα.

- Ασφάλεια δικτύων υπολογιστικών συστημάτων.

  Ο συγκεκριμένος τομέας έχει σχέση με τις ικανότητες που διαθέτει ένας οργανισμός ή μια επιχείρηση ώστε να προστατεύει τα δεδομένα και τις πληροφορίες από πιθανούς κινδύνους. Πέρα από αυτό, γίνεται λόγος και για τη δυνατότητα ενός τέτοιου δικτύου να φέρει αξιόπιστη αντίσταση σε απλές καταστάσεις ή ακόμα και σε επικίνδυνες ενέργειες οι οποίες θέτουν σε κίνδυνο την ακεραιότητα του απορρήτου των δεδομένων. Επίσης, αυτός ο τομέας συνδέεται στενά με τις τρεις βασικές ιδέες που αναφέραμε αρχικώς στις προϋποθέσεις ασφάλειας ενός πληροφοριακού συστήματος.
- Ασφάλεια συστημάτων βάσεων δεδομένων.

Η πληροφορία είναι το σημαντικότερο στοιχείο ενός πληροφοριακού συστήματος, γι' αυτό έχει μεγάλη αξία και γι' αυτό η ασφάλεια βάσεων δεδομένων αποκτά μεγάλη σημασία εφόσον οι οποίες αποθηκεύουν, επεξεργάζονται και μεταδίδουν τις πληροφορίες. Η αξιοπιστία της μετάδοσης ελέγχεται από ειδικά πρωτόκολλα τα οποία εγγυώνται την αποτελεσματική και ορθή ολοκλήρωση μεταφοράς των πληροφοριών και εφαρμόζουν κανόνες ακεραιότητας. Επίσης, η διαθεσιμότητα για τους χρήστες που κατέχουν δικαιώματα εξουσιοδότησης είναι απαραίτητη και αναγκαία. Οι βασικές διαστάσεις παραμέτρων ασφάλειας ΒΔ ακούνε στα ονόματα ακεραιότητα, διαθεσιμότητα, έλεγχος προσπέλασης, εμπιστευτικότητα κ.α., ενώ οι νέες διαστάσεις παραμέτρων ακούνε στα ονόματα διαιρετότητα, άμεση και έμμεση κ.α.

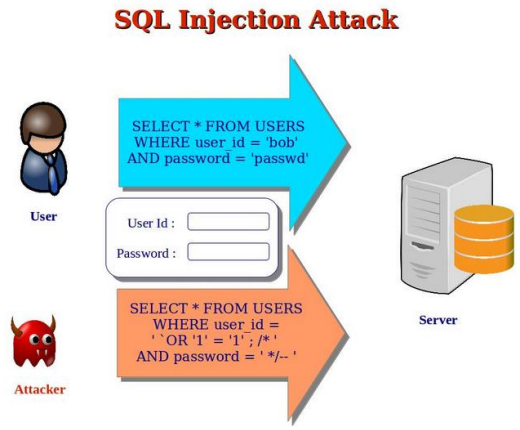
### 2.3.4 Γνωστοί τύποι επιθέσεων-Τεχνικές επίθεσης

Στην συνέχεια γίνεται ανάλυση μερικών εκ των γνωστότερων επιθέσεων σε πληροφοριακά συστήματα οπού συναντάμε.

#### SQL Injection

Οι βάσεις δεδομένων στις μέρες μας έχουνε γίνει ένα από τα σημαντικότερα στοιχεία των λειτουργικών και πληροφοριακών συστημάτων διότι μέσα σε αυτές έχουμε την ικανότητα να αποθηκεύσουμε μεγάλο όγκο από δεδομένα μέρος εκ των οποίων αναλύεται και τις ευαίσθητες προσωπικές μας πληροφορίες. Η έννοια *SQL* προέρχεται από τα αρχικά *Structured Query Language* και αποτελεί ένα μεγάλο μέρος των ευρέων χρησιμοποιούμενων βάσεων δεδομένων. Για να πραγματοποιήσουμε πρόσβαση σε μια βάση δεδομένων ένας εκ των τρόπων είναι της *SQL* γλώσσας, εφόσον αναφερόμαστε σε μια βάση αντίστοιχου τύπου, ώστε να εκτελούμε ερωτήματα, να αναζητούμε δεδομένα αλλά και να εισάγουμε, διαγράφουμε ή και τροποποιήσουμε εγγραφές. Η πρωταρχική μορφή των επιθέσεων *SQL injection* ήταν η απευθείας εισαγωγή κώδικα σε παραμέτρους οι οποίες συνδέονται με *SQL* εντολές. Σε μία λιγότερο άμεση επίθεση εισάγεται κακόβουλος κώδικας μέσω συμβολοσειρών οι οποίες είναι προορισμένες για αποθήκευση σε πίνακα. Όταν οι αποθηκευμένες συμβολοσειρές συνδέονται με δυναμική εντολή SQL ο κακόβουλος κώδικας είχε την δυνατότητα να εκτελεστεί. Επιπλέον σε μια εφαρμογή *Web* η οποία αποτυγχάνει να διασφαλίσει τις παραμέτρους οι οποίες περνάνε στην δημιουργία δυναμικών δηλώσεων SQL είναι δυνατόν για έναν εισβολέα να μεταβάλλει τη δομή αυτών. Όταν ένας εισβολέας του πληροφοριακού συστήματος, είναι ικανός να μεταβάλει μια δήλωση *SQL* η δήλωση θα εκτελεστεί με τα ίδια δικαιώματα που θα εκτελούνταν από τον χρήστη της φόρμας. .Αναλυτικότερα, η μέθοδος *SQL injection* αναφέρεται σε μία τάξη επίθεσης με εισαγωγή κώδικα στην οποία τα δεδομένα που παρέχονται από τον χρήστη εμπεριέχονται σε ένα *SQL query* με τέτοιο τρόπο ώστε μέρος των εισαχθέντων στοιχείων του χρήστη να αντιμετωπίζονται σαν *SQL* κώδικας. Αξιοποιώντας αυτά τα θέματα ευπάθειας ορισμένων εφαρμογών ένας εισβολέας μπορεί να υποβάλει *SQL queries* απευθείας στην βάση δεδομένων. Τα τελευταία χρόνια, η τεχνική της επίθεσης *SQL Injection* χρησιμοποιείται όλο και πιο πολύ

από αρκετούς επιτιθέμενους που τους δίνετε η δυνατότητα να εκτελέσουν εντολές της γλώσσας SQL ενάντια σε κάποιον στόχο *(πολλές φορές σε κάποιον εξυπηρετητή).* Έτσι, κάποιος εισβολέας του πληροφοριακού συστήματος, χρησιμοποιώντας τις ικανότητές του έχει τη δυνατότητα να αποσπάσει σημαντικό πλήθος ευάλωτων πληροφοριών από μια βάση δεδομένων. Ειδικότερα, οι web εφαρμογές οι οποίες είναι ευάλωτες σε *SQL injection* μπορεί να επιτρέψουν στον εισβολέα του συστήματος, να αποκτήσει πλήρη πρόσβαση στις βάσεις δεδομένων τους




*Εικόνα 16 - Σχήμα επίθεσης τύπου SQL Injection*

Αυτού του είδους οι επιθέσεις αποτελούν σοβαρή απειλή για κάθε Web εφαρμογή η οποία δέχεται την εισαγωγή δεδομένων από χρήστες και τα οποία ενσωματώνει σε *SQL queries* σε μία βάση δεδομένων. Οι περισσότερες *Web* εφαρμογές οι οποίες χρησιμοποιούνται στο δίκτυο ή σε εταιρικά συστήματα ακολουθούν αυτή τη λογική λειτουργίας με τον χρήστη και επομένως είναι τρωτές σε *SQL injection*. Η αιτία δημιουργίας τρωτών σημείων ευάλωτων σε *SQL injection* είναι σχετικά απλή και κατανοητή και πρόκειται για την ανεπαρκή επικύρωση των εισαχθέντων δεδομένων από τον χρήστη. Για την αντιμετώπιση αυτού του προβλήματος οι προγραμματιστές έχουν προτείνει μια σειρά κατευθυντήριων γραμμών κωδικοποίησης οι οποίες προωθούν αμυντικές πρακτικές κωδικοποίησης όπως είναι η κωδικοποίηση των εισαχθέντων στοιχείων από το χρήστη και η επικύρωση τους. Μια αυστηρή και συστηματική εφαρμογή αυτών των τεχνικών αποτελεί μια αποτελεσματική αντιμετώπιση της

ευπάθειας της εφαρμογής σε επιθέσεις με *SQL* injection. Ωστόσο στην πράξη η εφαρμογή τέτοιων τεχνικών βασίζεται στην επιλογή του προγραμματιστή και γι' αυτό είναι επιρρεπής σε σφάλματα. Αν και πρόσφατα ξεκίνησε να δίνεται μεγάλη βάση στο πρόβλημα ευπάθειας μιας εφαρμογής με *SQL injection* πολλές από τις προτεινόμενες λύσεις αδυνατούν να αντιμετωπίσουν το πρόβλημα στην πλήρη του έκταση. Υπάρχουν πολλοί τύποι *SQL injection* καθώς και αμέτρητες παραλλαγές των τύπων αυτών. Ερευνητές καθώς και επαγγελματίες συχνά αγνοούν αυτή τη μυριάδα των διαφορετικών τεχνικών που μπορούν να χρησιμοποιηθούν για την εκτέλεση των *SQL injection*. Επομένως οι περισσότερες από τις προτεινόμενες λύσεις αποτρέπουν ή ανιχνεύουν μόνο ένα υποσύνολο των πιθανών *SQL injection*. Για κάθε είδος επίθεσης δίνεται ένας χαρακτηρισμός που απεικονίζει το αποτέλεσμά του. Μία επίθεση με *SQL injection* πραγματοποιείται όταν ο εισβολέας αλλάζει το επιδιωκόμενο αποτέλεσμα ενός SQL query με την εισαγωγή λέξεων-κλειδιά της *SQL*. Αυτός ο άτυπος ορισμός περιλαμβάνει όλες τις παραλλαγές των *SQL injection*. Παρακάτω ορίζουμε δύο σημαντικά χαρακτηριστικά των *SQL injection* που χρησιμοποιούμε για την περιγραφή των επιθέσεων: ο μηχανισμός injection και η πρόθεση επίθεσης *(attack intent).*

### *Cross Site Scripting*

Το *Cross Site Scripting,* αποτελεί είναι ένα από τα πιο διαδεδομένα κενά ασφαλείας *Web* εφαρμογών, το οποίο συμβαίνει όταν μια εφαρμογή αποκτά κάποια αναξιόπιστα δεδομένα και τα προωθεί σε έναν φυλλομετρητή δίχως την κατάλληλη έγκριση. Η παραπάνω επίθεση αφήνει στους κακόβουλους χρήστες, το περιθώριο να εκτελέσουν κώδικα στον φυλλομετρήτη των θυμάτων τους, όταν οι τελευταίοι επισκέπτονται μια ιστοσελίδα με αποτέλεσμα να καταστρέφουν και να οδηγουν ακομα και τον ίδιο τον χρήστη σε κακόβουλες ιστοσελίδες. Επομένως, το XSS βασίζεται στην εκμετάλλευση διάφορων ευπαθειών υπολογιστικών και πληροφοριακών συστημάτων με τη χρήση *HTML* κώδικα ή *JavaScript* μέσα σε κάποιον ιστότοπο. Μέσω των επιθέσεων τύπου XSS ο κακόβουλος χρήστης είναι σε θέση να πραγματοποιήσει ενδεικτικά τις ακόλουθες ενεργείς:

- Κλοπή κωδικών και διάφορων άλλων προσωπικών δεδομένων.
- Αλλαγή ρυθμίσεων της ιστοσελίδας.
- Κλοπή cookies.

Συνεπώς, η ευπάθεια γίνεται αντιληπτή όταν το σύστημα αρχίζει να εμφανίζει σημάδια αδυναμίας και δεν είναι σε θέση να υποστηρίξει τις υποχρεώσεις του, καθώς και να φιλτράρει όλες τις εισόδους έτσι ούτως ώστε να διαχωρίσει τις επιβλαβείς και να τι απορρίψει. Περισσότεροι από τους ειδικούς διαχωρίζουν τις ευπάθειες από τις XSS επιθέσεις σε δύο βασικές κατηγορίες, τις μόνιμες και τις μη μόνιμες. Επίσης, δύο άλλες κατηγορίες που μπορούν να χωριστούν είναι οι παραδοσιακές επιθέσεις (οι οποίες προκαλούνται από την πλευρά του εξυπηρετητή) και οι επιθέσεις βασισμένες σε DOM (οι οποίες προκαλούνται από την πλευρά του πελάτη).

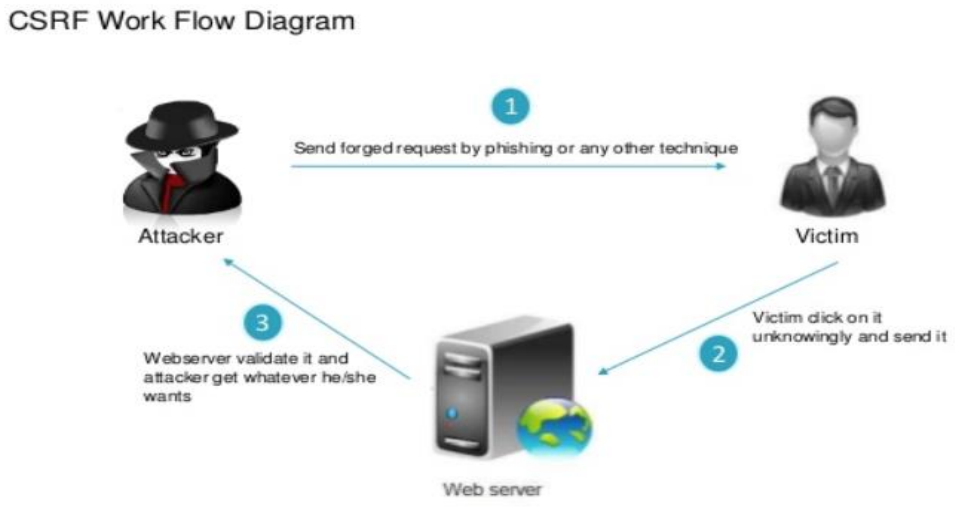

*Εικόνα 17 - Διάγραμμα επίθεσης XSS*

### *Content Spoofing(phising)*

Η τεχνική επίθεσης *Content spoofing* γνωστή και ως *text based injection*, αποτελεί ένα είδος exploit που χρησιμοποιείται από κακόβουλους χρήστες για την παρουσίαση μιας πλαστής ή τροποποιημένης ιστοσελίδας σε ένα ανυποψίαστο χρήστη ως αυθεντική και νόμιμη. Στόχος και λογική της τεχνικής αυτής, είναι η εξαπάτηση των χρηστών και η παραποίηση της πραγματικής πηγής προέλευσης ή ενός ατόμου. Σύμφωνα με τον οργανισμό *OWASP* αυτή η τεχνική ονομάζεται επίσης ως *Content Injection* ή *Virtual*

*Defacement*. Η συγκεκριμένη επίθεση χρησιμοποιείται συνήθως με επιθέσεις τύπου *Social Engineering* και εκμεταλλεύεται τον κώδικα και την εμπιστοσύνη του θύματος ως προς την ιστοσελίδα. Ο επιτιθέμενος μερικές φορές χρησιμοποιεί απλό κείμενο ενώ σε άλλες περιπτώσεις μπορεί να τροποποιήσει τις πληροφορίες και τους συνδέσμους, καθώς και άλλες *html* ετικέτες, μέσα από τον εξυπηρετητή της εφαρμογής. Με αποτέλεσμα σε μια φαινομενικά αξιόπιστη ιστοσελίδα, το θύμα δεν αντιλαμβάνεται την απάτη αφού οπτικά δεν υπάρχει καμία διαφορά, και το *URL* που εμφανίζεται στον περιηγητή φαίνεται να είναι νόμιμο. Ουσιαστικά, η επίθεση αυτή βασίζεται στην ευπάθεια του ακατάλληλου χειρισμού εισόδου και εξόδου των δεδομένων που εισάγει ο χρήστης μέσα σε φόρμες συμπλήρωσης στοιχείων ή φόρμες σύνδεσης. Τις περισσότερες φορές, ο στόχος της επίθεσης είναι η υποκλοπή των προσωπικών δεδομένων του θύματος.

### *Cross Site Request Forgery*



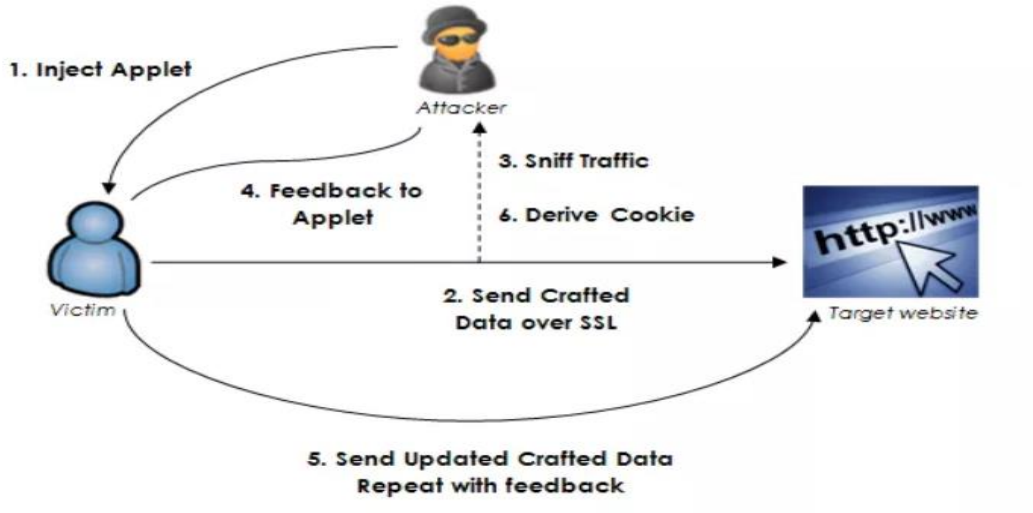

*Εικόνα 18-Cross site Request Forgery*

Το CSRF είναι μια επίθεση που ξεγελάει το θύμα να υποβάλει ένα κακόβουλο αίτημα. Κληρονομεί την ταυτότητα και τα προνόμια του θύματος και εκτελεί ανεπιθύμητες λειτουργίες για λογαριασμό του επιτιθέμενου. Για τους περισσότερους ιστότοπους, τα αιτήματα του προγράμματος περιήγησης περιλαμβάνουν αυτόματα οποιαδήποτε

διαπιστευτήρια που σχετίζονται με τον ιστότοπο , όπως το cookie id , τη διεύθυνση IP, τα διαπιστευτήρια τομέα των Windows. Επομένως, αν ο χρήστης είναι προς το παρόν επικυρωμένος στον ιστότοπο, ο ιστότοπος δεν θα έχει τρόπο να κάνει διάκριση μεταξύ του πλαστογραφημένου αιτήματος που έχει αποσταλεί από τον επιτιθέμενο και ενός νόμιμου αιτήματος που έχει σταλεί από τον απλο χρήστη.

Το CSRF επιτίθεται σε λειτουργίες του στόχου που προκαλούν αλλαγές στη βάση δεδομένων του διακομιστή , όπως η αλλαγή της διεύθυνσης ηλεκτρονικού ταχυδρομείου ή του κωδικού πρόσβασης του θύματος. Αναγκάζοντας το θύμα να ανακτήσει τα δεδομένα ούτως ώστε ο επιτιθέμενος με την σειρά του να είναι σε θέση να τα αποκρυπτογραφήσει .Μερικές φορές είναι δυνατό να εκτελεστεί η επίθεση CSRF σε κάποιον ευάλωτο ιστότοπο. Τέτοιες ευπάθειες ονομάζονται "αποθηκεμένες ευπάθειες CSRF". Αυτό μπορεί να επιτευχθεί με την απλή αποθήκευση μιας ετικέτας IMG ή IFRAME σε ένα πεδίο που αποδέχεται HTML, . Εάν ο επιτιθέμενος μπορεί να αποθηκεύει CSRF κακόβουλο κώδικα στον ιστότοπο, ενισχύεται η σοβαρότητα της επίθεσης. Ειδικότερα, η πιθανότητα εκμετάλλευσης αυξάνεται εάν το θύμα επιλέξει να κλικάρει πάνω στο Link οπού θα τρέξει ο κακόβουλος κώδικας από κάποια τυχαία σελίδα στο Διαδίκτυο. Η πιθανότητα αυξάνεται επίσης επειδή ο επιτιθέμενος είναι σχεδόν βέβαιο ότι το 'θύμα' θα έχει ήδη πιστοποιηθεί στον ιστότοπο.

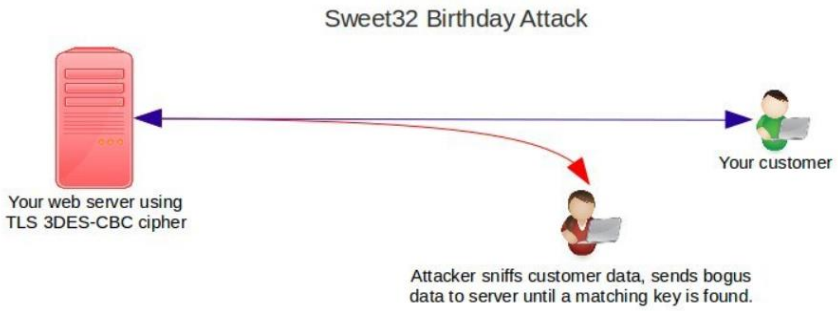
*Breach-Attack*




*Εικονα 19 -Breach Attack*

Η επίθεση BREACH είναι γνωστοποιημένη με την έκδοση του πρωτοκόλλου SSL / TLS που χρησιμοποιείται και είναι αποτελεσματική σε κάθε τύπο κρυπτογράφησης εφόσον πληρούνται οι ακόλουθες συνθήκες: Η εφαρμογή web να υλοποιείται με τη χρήση συμπίεσης σε επίπεδο HTTP και να αντανακλά τα δεδομένα που παρέχονται από το χρήστη και ένα στατικό μυστικό κλειδί ανταπόκρισης HTTP(CSRF Token). Ο εισβολέας ξέρει τι πρέπει να ψάξει και είναι σε θέση να παρακολουθεί την κυκλοφορία μεταξύ του χρήστη και της εφαρμογής ιστού (man-in-the middle), προκειμένου να ανακτήσει το μήκος των απαντήσεων HTTP. Ο επιτιθέμενος είναι σε θέση να πείσει τον χρήστη να επισκεφτεί έναν ιστότοπο που περιέχει κακόβουλο κώδικα με αποτέλεσμα το πρόγραμμα περιήγησης του θύματος πλέον να είναι ικανό να αποστείλει κακόβουλα αιτήματα στον ιστότοπο προορισμού.

*Sweet32-Attack*




*Εικονα 20- Sweet 32 Attack*

Τα κρυπτογραφικά πρωτόκολλα, όπως το TLS, το SSH, το IPsec και το OpenVPN, χρησιμοποιούν συνήθως αλγόριθμους κρυπτογράφησης, όπως το AES, το Triple-DES και το Blowfish, για την κρυπτογράφηση δεδομένων μεταξύ υπολογιστών-πελατών και διακομιστών. Για να χρησιμοποιηθούν τέτοιοι αλγόριθμοι, τα δεδομένα διασπώνται σε κομμάτια σταθερού μήκους, που ονομάζονται μπλοκ, και κάθε τεμάχιο κρυπτογραφείται ξεχωριστά σύμφωνα με έναν τρόπο λειτουργίας. Τα παλαιότερα κρυπτογραφημένα μπλοκ, όπως το Triple-DES και το Blowfish, χρησιμοποιούν ένα μέγεθος μπλοκ 64 bit, ενώ το AES χρησιμοποιεί ένα μέγεθος μπλοκ 128 bit.Είναι γνωστό στην κρυπτογραφική κοινότητα ότι ένα μικρο μέγεθος μπλοκ κάνει έναν κωδικό μπλοκ ευάλωτο, ακόμα και αν δεν υπάρχουν κρυπτογραφικές επιθέσεις κατά του ίδιου του κρυπτογραφικού τεμαχίου. Παρατηρούμε ότι τέτοιες επιθέσεις έχουν πλέον καταστεί πρακτικές για την κοινή χρήση κρυπτογραφημενων μπλοκ 64 bit σε δημοφιλή πρωτόκολλα όπως το TLS και το OpenVPN. Παρόλα αυτά, η συγκεκριμένη κρυπτογράφηση είναι ευρέως εφαρμοσμενη στο Διαδίκτυο. Το Blowfish είναι σήμερα ο προεπιλεγμένος κρυφός κώδικας στο OpenVPN και το Triple-DES υποστηρίζεται από σχεδόν όλους τους διακομιστές ιστού HTTPS και χρησιμοποιείται σήμερα για περίπου το 1-2% των συνδέσεων HTTPS μεταξύ mainstream browsers και web servers.Εχει επαληθευθει ότι ένας εισβολέας δικτύου εαν παρακολουθησει μια μακρόχρονη σύνδεση Triple-DES HTTPS μεταξύ ενός προγράμματος περιήγησης ιστού και ενός ιστότοπου μπορεί να ανακτήσει ασφαλή cookie HTTP, λαμβάνοντας περίπου 785 GB δεομενων, χρησιμοποιώντας κακόβουλο κωδικα Javascript για τη δημιουργία επισκεψιμότητας . Η διατήρηση μιας ζωντανής σύνδεσης στο διαδίκτυο για δύο ημέρες μπορεί να μην φαίνεται πολύ πρακτική, αλλα λειτούργει εύκολα σε εργαστηριακες συνθήκες. Όσον αφορά την υπολογιστική πολυπλοκότητα,

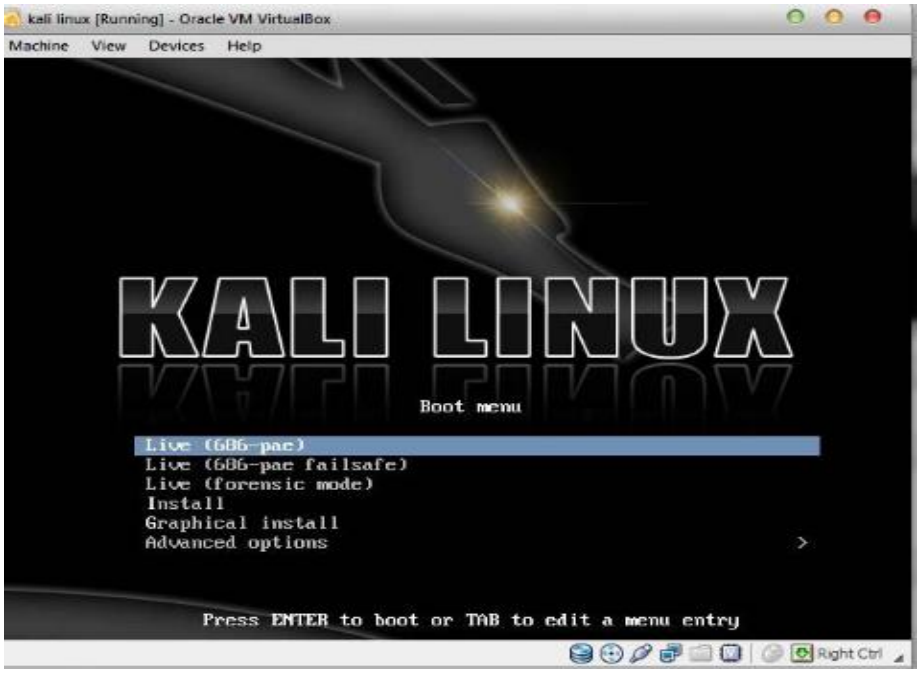
αυτή η επίθεση είναι συγκρίσιμη με τις πρόσφατες επιθέσεις κατά του RC4.Τα αντίμετρα εφαρμόζονται επί του παρόντος από τους προγραμματιστές του προγράμματος περιήγησης, και παράλληλα πληροφορώντας τους τελικούς χρήστες να έχουν εγκατεστημένες τις τελευταίες και πιο ενημερωμένες εκδόσεις περιηγητών.

# Κεφάλαιο 3 – Πρακτική εφαρμογή

Στο παρών κεφάλαιο της διατριβής αναλύεται η πρακτική η οποία και ακολουθήθηκε για την λήψη πειραματικών και ρεαλιστικών αποτελεσμάτων σχετικά με την ασφάλεια πληροφοριακών συστημάτων. Στην συνέχεια του κεφαλαίου γίνεται η ανάλυση του περιβάλλοντος χρήσης ανάλυσης, των εργαλείων που χρησιμοποιούνται και τέλος τα αποτελέσματα τα όποια εξάγονται από την παραπάνω διαδικασία.

## 3.1 Προετοιμασία υπηρεσιών

Για τις ανάγκες της παρούσας διατριβής χρησιμοποιήθηκε το πρόγραμμα *Oracle VM Virtual Box (Win7, 64-bit)* και δημιουργήθηκε ένα περιβάλλον εργασίας φιλικό ως προς τον pentester λειτουργικό περιβάλλον για της ανάγκες της εργασίας. Συγκεκριμένα, εγκαταστάθηκε το virtual image του  Kali Linux, που έπαιξαν το ρολό του λειτουργικού συστήματος του επιτιθέμενου και του συστήματος-στόχου, δηλαδή των διακομιστων www.hua.gr, application.hua.gr, webms.hua.gr.

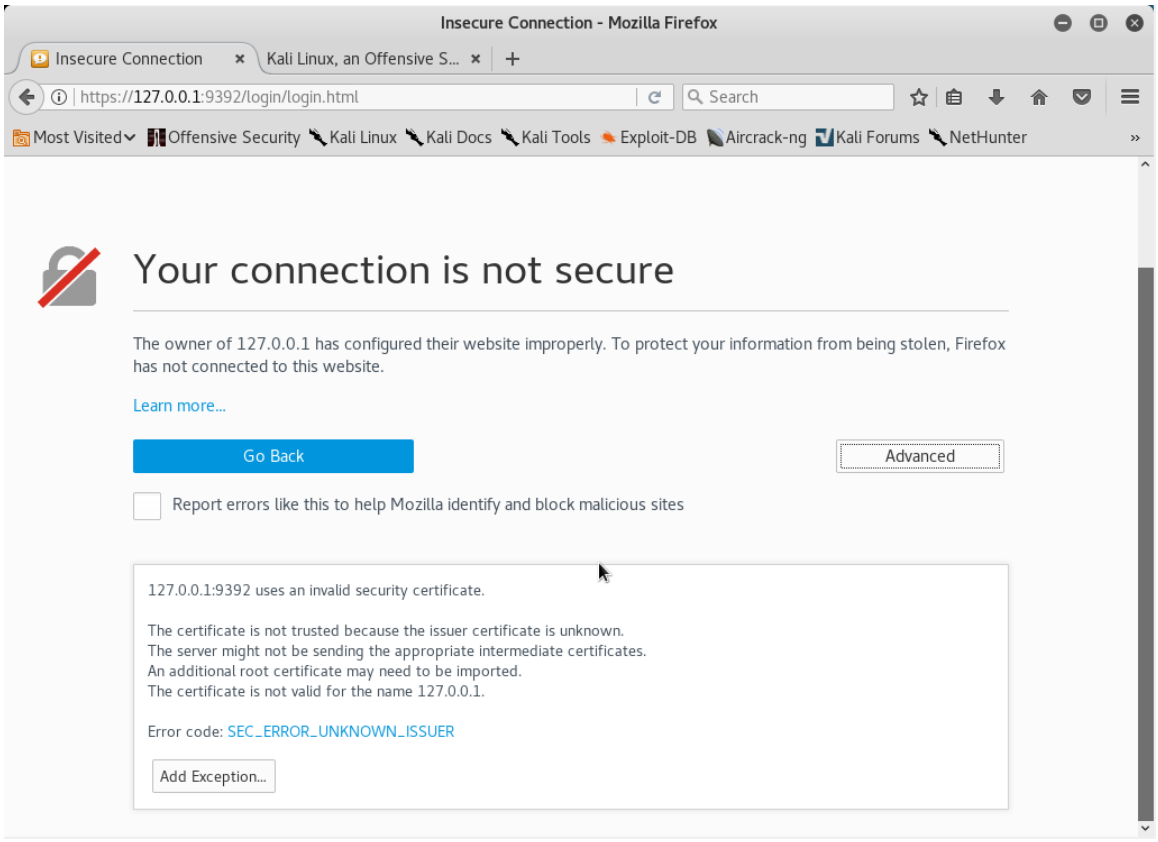



*Εικονα 21-kali linux installation screen*

Για την ανάλυση του πληροφοριακού συστήματος χρησιμοποιήσαμε αυτοματοποιημένες πλατφόρμες ανάλυσης (frameworks) οι οποίες ειδικεύονται στην ανάλυση ψηφιακών συστημάτων και εμφανίζουν τα αποτελέσματα με συγκεντρωτικό τρόπο ώστε να βοηθήσουν το χρήστη στην ανάλυση-επεξήγηση των αποτελεσμάτων καθώς και την αναζήτηση λεπτομέρειών με εύκολο τρόπο.

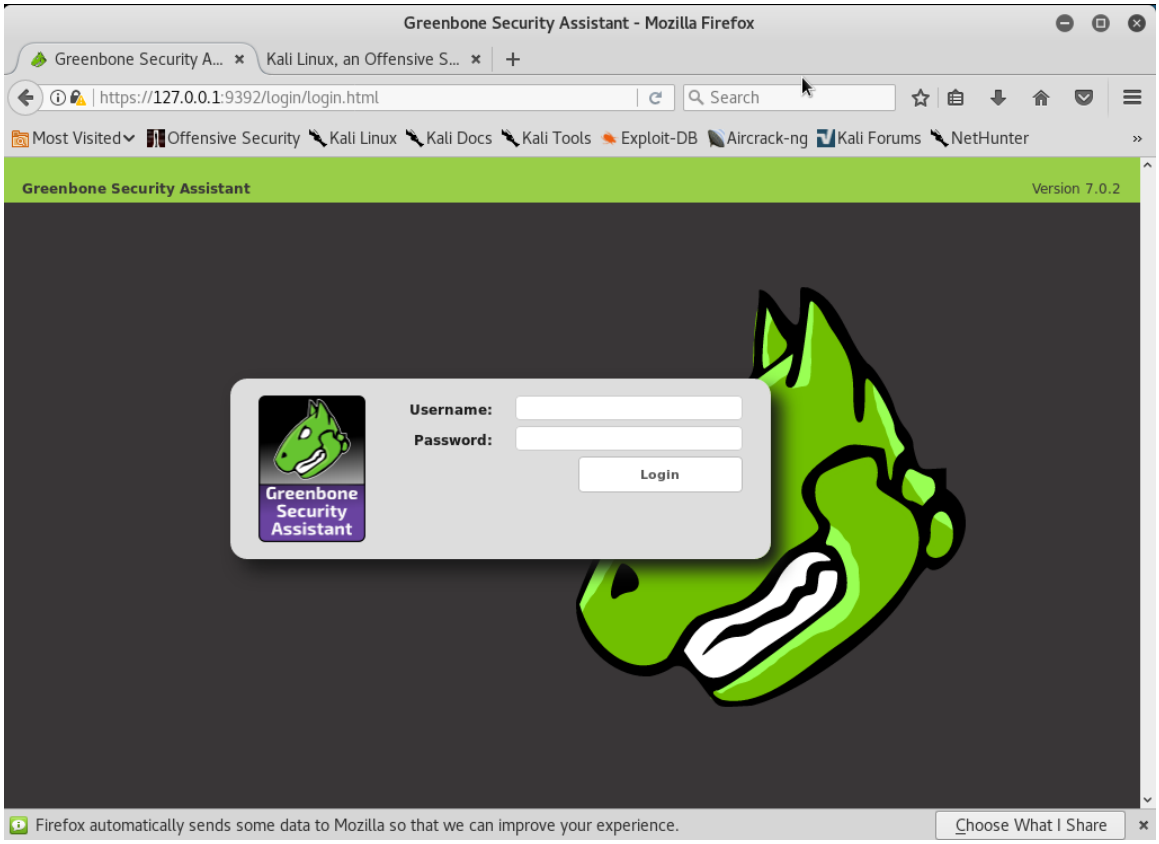
### 3.1.2 Εγκατάσταση και Εκτέλεση OpenVas

Για την διαδικασία εκτέλεσης του *OpenVas* σε μία διεύθυνση ενασχόλησης ακολουθούμε τα παρακάτω βήματα. Αρχικά μέσω του *browser* του συστήματος κάνουμε μετάβαση στην τοπική διεύθυνση του συστήματος *(localhost)* και δεχόμαστε το υπογεγραμμένο πρωτόκολλο *ssl.* Η παραπάνω ενέργεια είναι υποχρεωτική.




*Εικόνα 22 - Προσθήκη πρωτοκόλλου SSL*

Στη συνέχεια ο χρήστης είναι σε θέση να συνδεθεί στη πλατφόρμα, και στο γραφικό περιβάλλον αυτής, μέσω του οποίου και γίνονται όλες οι ενέργειες ανάλυσης και λήψης αποτελεσμάτων του εργαλείου. Η την σύνδεση του χρήστη στην πλατφόρμα γίνεται μέσω των προσωπικών του στοιχείων τα οποία παράγονται κατά την εγκατάσταση του πακέτου.



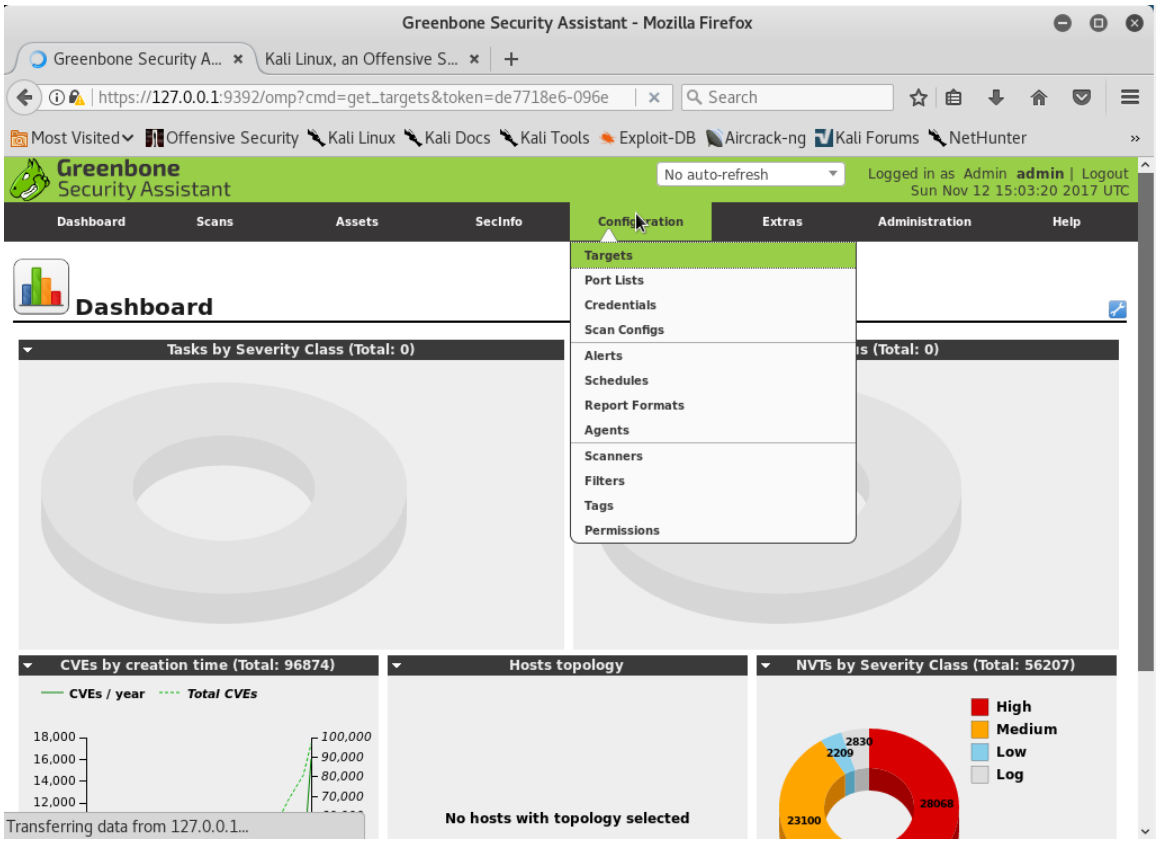

*Εικόνα 23 - Σελίδα σύνδεσης χρήστη OpenVas*

Έπειτα της σύνδεση του χρήστης, ο τελευταίος είναι σε θέση να δημιουργεί νέους στόχους ανίχνευσης και αξιολόγησης. Η παραπάνω λειτουργία γίνεται από το κεντρικό *menu* επιλογών το οποίο του παρέχει το σύστημα στο γραφικό περιβάλλον της σελίδας. *(Σημείωση: στην συνέχεια του παραδείγματος γίνεται χρήση του στόχου hua.gr)*

Find IP Results: 12 Nov 2017 05:06:20 PM

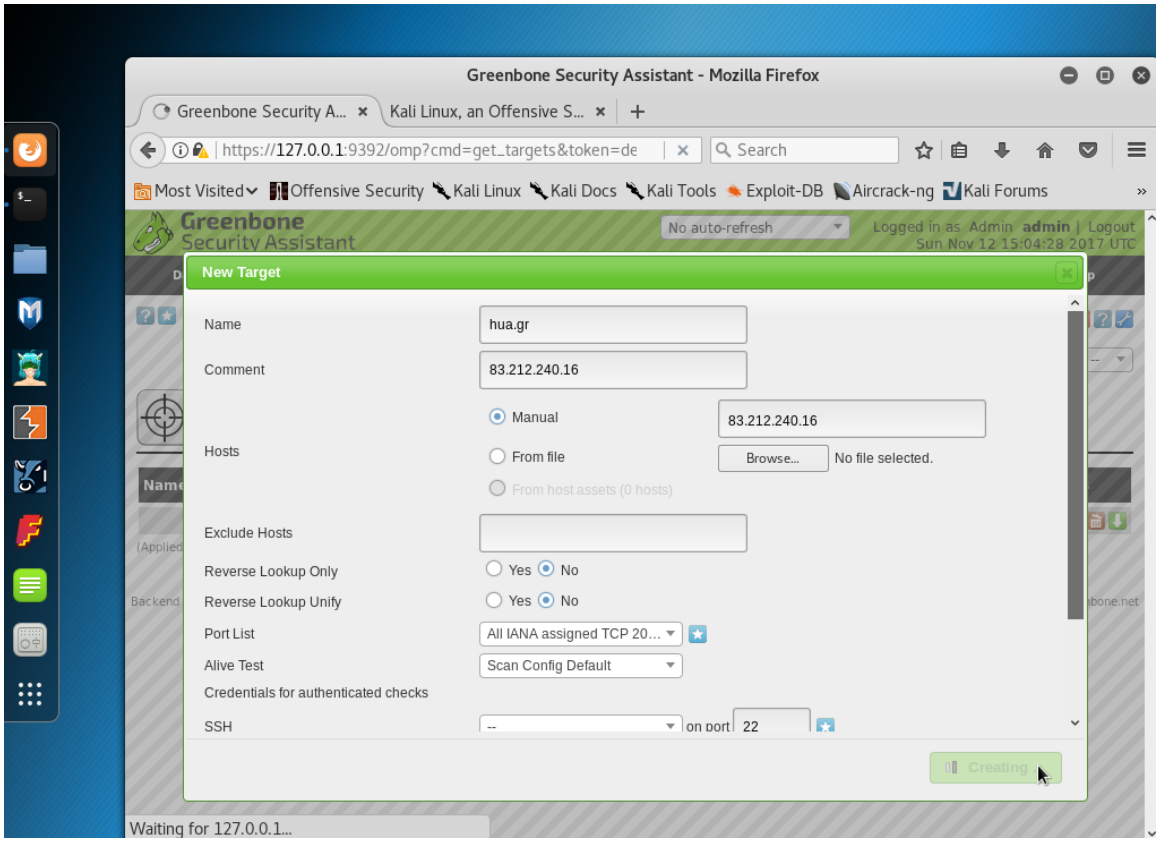
| S. No. | Domain Name | IP Address |
|---|---|---|
| 1 | www.hua.gr | ww.hua.gr./83.212.240.16 |

*Εικόνα 24 - Ανάλυση διεύθυνσης IP στόχου*




*Εικόνα 25 - Δημιουργία νέου στόχου ανάλυσης OpenVas*

Για την δημιουργία του στόχου, πρέπει μέσω της επιλογής *target (στόχος)* ο χρήστης να επιλέξει την δημιουργία ενός «νέου στόχου», όπου στην συνέχεια θα εισάγει τα απαραίτητα πεδία, στην περίπτωση μας την διεύθυνση *IP* του διακομιστή της ιστοσελίδας ανάλυσης. Ο χρήστης είναι σε θέση σε αυτό το σημείο να παραμετροποιήσει παράταιρο τον τρόπο της ανίχνευσης όπως της θύρες όπου θα ανιχνευθούν πιθανές ευπάθειες αλλά και πρωτόκολλα όπως *proxy* ή *ssh* ώστε να επεκτείνει την πολυπλοκότητα της έρευνας του. Για τις ανάγκες της παρούσας διατριβής γίνεται χρήση των αρχικών τιμών όπου καθορίζει το εργαλείο. Στην περίπτωση της συγκεκριμένης έρευνας δεν γίνεται κάποιος πολύπλοκος έλεγχος αποτελεσμάτων καθώς θέλουμε να αναγνωρίσουμε συγκεκριμένα απλά αποτελέσματα αναγνωρισμένων ευπαθειών. Έπειτα ο χρήστης μπορεί να προχωρήσει στην έναρξη της διερεύνησης των αποτελεσμάτων και την παραγωγή των όλων πιθανών ελέγχων.




*Εικόνα 26 - Επίλυση στόχου OpenVas*

## 3.2 Διαδικτυακοί Στόχοι

Για την παρούσα διατριβή, οι στόχοι οι οποίοι γίνεται ανάλυση ευπαθειών διακρίνονται στον ακόλουθο πίνακα. Για κάθε διακομιστή διακρίνεται το *Domain* καθώς και η διεύθυνση *IP*.

| **Διακομιστής** | **IP** |
|---|---|
| application.hua.gr | 83.212.240.17 |
| webms.hua.gr | 83.212.240.12 |
| www.hua.gr | 83.212.240.16 |

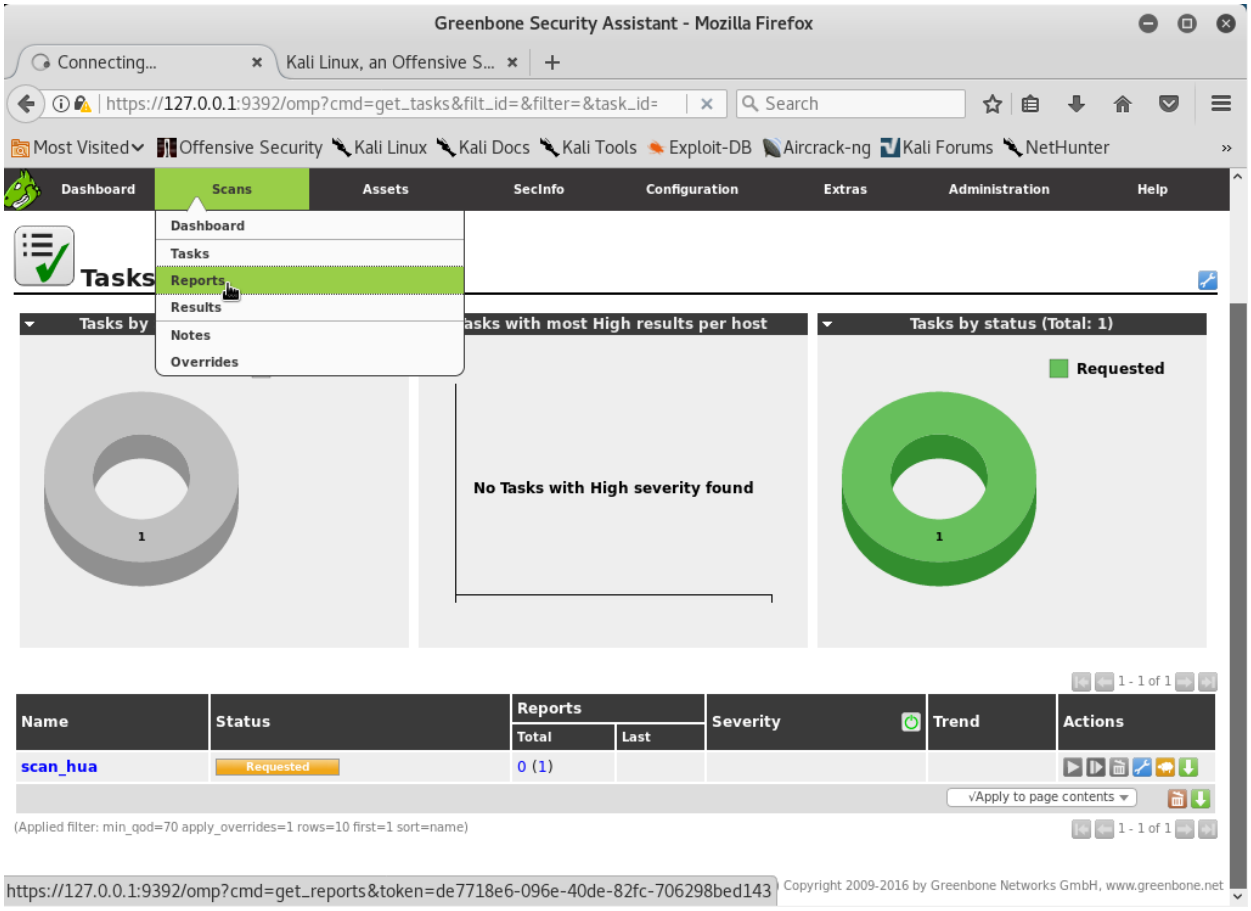
Ως *Domain* ορίζεται ένα όνομα χώρου, δηλαδή μια μοναδική ταυτότητα μιας εταιρίας, μίας οργάνωσης ή ενός ατόμου, η οποία μπορεί να χρησιμοποιηθεί σα βάση για πολλές

συναλλαγές στο Διαδίκτυο. Το domain name είναι μία λέξη που επιλέγουμε προκειμένου να μπορούμε με εύκολο τρόπο να συνδεθούμε με έναν υπολογιστή στο Internet. Η λέξη αυτή πάντα προσδιορίζεται περαιτέρω από μία κατάληξη που χαρακτηρίζει κατά κάποιο τρόπο την "περιοχή" του δικτύου στην οποία ανήκει. Έτσι, για το χώρο ονομάτων με κατάληξη [.gr], ένα domain name θα έχει την μορφή domain.gr . Για να επισκεφθούμε τις σελίδες που του αντιστοιχούν, θα πρέπει να πληκτρολογήσουμε σε κάποιο πρόγραμμα πλοήγησης, μια διεύθυνση της μορφής http://www.domain.gr.

### 3.2.1 Αποτελέσματα εκτίμησης

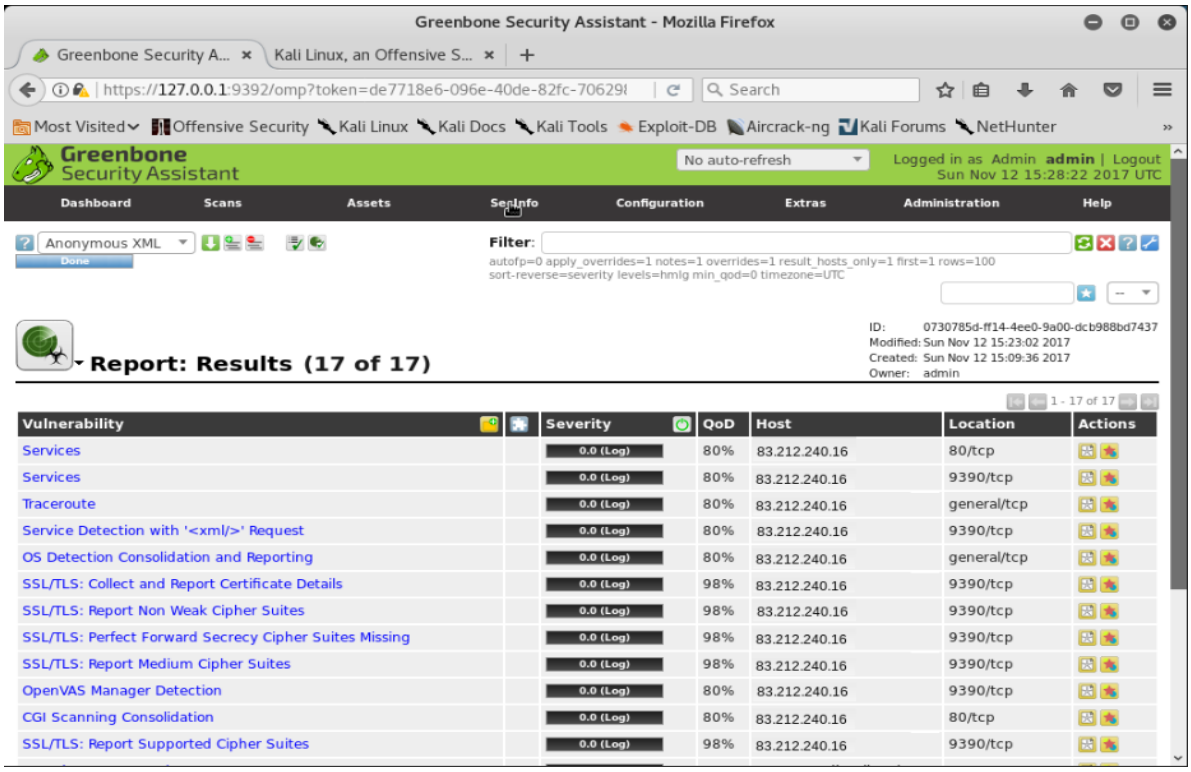
- Αποτελέσματα OpenVas

Για την δημιουργία της παραπάνω κατηγοριοποιημένης αναφοράς ο χρήστης δημιουργεί μια νέα task *(ενέργεια)* την οποία στη συνέχεια εκτελεί.




*Εικόνα 27-Open vas Report*

Εφόσον η κατάσταση της παραπάνω ενέργειας είναι ολοκληρωμένη τότε ο χρήστης είναι σε θέση να διαβάσει τα αποτελέσματα αυτής.




*Εικόνα 28-Open vas Report severity*

Στην περίπτωση μας το αναλυτικό *log* των αποτελεσμάτων διακρίνεται στη συνέχεια. Ο πίνακας της ανάγνωσης στου στόχους είναι ο ακόλουθος καθώς και το port όπου γίνεται η κάθε πιθανή ανίχνευση αποτελεσμάτων.

| Service (Port) | Threat Level |
|---|---|
| 443/tcp | High |
| 443/tcp | Medium |

| Service (Port) | Threat Level |
|---|---|
| 111/tcp | Log |
| 443/tcp | Log |
| 80/tcp | Log |
| 22/tcp | Log |
| 5353/tcp | Log |
| 111/udp | Log |

*Για παραδειγμα παρακάτω βλέπουμε την εξαγωγή μιας ευπάθειας σε μορφή pdf απο το ίδιο το εργαλείο αυτοματοποιημένα.*

| High (CVSS: 7.5)<br>NVT: Joomla! Core Two-factor Authentication Bypass Vulnerability Nov17 |
|---|
| **Αποτέλεσμα ανίχνευσης ευπάθειας**<br><br>cpe: / a: joomla: joomla: 3.4.3<br>Εντοπίστηκε από την ανίχνευση έκδοσης του Joomla<br>(OID: 1.3.6.1.4.1.25623.1.0.100330) |
| |
| **Σύνοψη** |
| Αυτός ο εξυπηρετητής εκτελεί το Joomla και είναι επιρρεπής<br>σε μια ευπάθεια παράκαμψης ταυτότητας. |
| **Αποτέλεσμα ανίχνευσης ευπάθειας**<br><br>Εγκατεστημένη έκδοση: 3.4.3<br>Σταθερή έκδοση: 3.8.2 |
| **Επίπτωση**<br><br>Η επιτυχής εκμετάλλευση αυτής της ευπάθειας θα επιτρέψει σε απομακρυσμένους εισβολείς<br>να παρακάμψουν ορισμένους περιορισμούς ασφαλείας και να εκτελέσουν μη εξουσιοδοτημένες<br>ενέργειες, κάτι που μπορεί να βοηθήσει στην εκτόξευση περαιτέρω επιθέσεων.<br><br>Επίπεδο Επιπτώσεων: Εφαρμογή |
| **Επίλυση**<br><br>Τρόπος επίλυσης: VendorFix<br>Αναβάθμιση σε έκδοση Joomla 3.8.2 ή νεότερη έκδοση.<br>Για ενημερώσεις, ανατρέξτε στη διεύθυνση https://www.joomla.org |
| **Λογισμικό / λειτουργικό σύστημα που επηρεάζεται**<br><br>Βασική έκδοση Joomla 3.2.0 έως 3.8.1 |
| **Εντοπισμός ευπάθειας**<br><br>Το ελάττωμα υπάρχει εξαιτίας σφάλματος που σχετίζεται<br>με τη μέθοδο επαλήθευσης 2-παραγόντων. |

Στους ακόλουθους πίνακες διακρίνουμε το σύνολο των αποτελεσμάτων εύρεσης ευπαθειών από το εργαλείο *OpenVas*, όπου έγιναν στους ακόλουθους τρεις διακομιστές εσωτερικά του δικτύου μέσω vpn(Inside VPN) καθώς και εξωτερικά(Outside VPN).

Παρακάτω καταγράφεται αριθμητικά σε μορφη πίνακα το σύνολο των ευπαθειών στις αναλογες ανοιχτές πόρτες αλλά και η κρισιμότητα αυτών.

## Inside VPN

| DNS | THREAT LEVEL | | | | | | |
|---|---|---|---|---|---|---|---|
| **Hostnames** | NVT(Network vulnerability tests) | High importance | Medium importance | Low importance | Log | Service ports | |
| | | | | | | TCP | UDP |
| www.hua.gr [83.212.240.16] | 10 | 2 | 8 | 0 | 0 | 443,80,22, 111 | 5353, 111 |
| **webms.hua.gr** [83.212.240.12] | 25 | 0 | 0 | 0 | 25 | 443,80,22, 111,3306 | |
| **application.hua.gr** [83.212.240.17] | 11 | 0 | 11 | 0 | 0 | 443,80 8080,8181 4848,3700 9999,8082 22,111, 9996 | |

## Outside VPN

| DNS | THREAT LEVEL | | | | | | |
|---|---|---|---|---|---|---|---|
| **Hostnames** | NVT(Network vulnerability tests) | High importance | Medium importance | Low importance | Log | Service ports | |
| | | | | | | TCP | UDP |
| **www.hua.gr** [83.212.240.16] | 10 | 2 | 8 | 0 | 0 | 443,80,111 | |
| **webms.hua.gr** [83.212.240.12] | 5 | 0 | 0 | 0 | 5 | 111 | |
| **application.hua.gr** [83.212.240.17] | 11 | 0 | 11 | 0 | 0 | 443,80 8080,8181 4848,3700 9999,8082 111,9996 | |

**NVT(**Network vulnerability tests)

Στον server https://www.hua.gr εκτελεστηκαν οι παρακάτω ρουτίνες δικτυακών ευπαθειών αυτοματοποιημένα απο την βαση δεδομένων του εργαλείου όπου και βρέθηκαν αριθμητικά οι παρακάτω ευπάθειες,**ομοίως εσωτερίκα και εξωτερικά** του δικτύου στον διακομιστή με IP 83.214.240.16 ως πρός τίς αντίστοιχες πόρτες:

1. Joomla!Core Two-factor Authentication Bypass Vulnerability [443tcp]
2. Joomla Alternative PHP File Extension File upload and information disclosure [443tcp]
3. Missing secure cookie attribute [443tcp]
4. Joomla! 3.8.0 LDAP information Disclosure vulnerability [443tcp]
5. Joomla! Core LDAP information Disclosure vulnerability [443tcp]

6.Joomla! CVE-2017-7988 Security Bypass [443tcp]
7. Joomla! core Privillege Escalation vulnerability [Port 443]
8.Joomla! Information disclosure and Cross-Site scripting vulnerability [443tcp]
9.Multiple full path information disclosure vulnerabilities [443tcp]
10.'swf' file upload and multiple cross-site scripting vulnerabilities [443tcp]

Αντίστοιχα στον server https://webms.hua.gr με IP 83.212.240.12 βρέθηκαν οι παρακατω 'τρύπες' ασφαλείας σε μή βαθμονομημένη σήμανση σοβαρότητας απο το ιδιο το εργαλείο, ουσιαστικά η κρισιμοτητα της ευπαθειας εξαρτάται απο τις ικανότητες του επιτιθέμενου να δημιουργησει συνθήκες εκμετάλευσης. Αυτο σημαίνει οτι παρέχονται πολλες πληροφορίες σε επίπεδο log ως προς τις αντίστοιχες πόρτες:

1. OS Detection consolidation and Reporting [General/TCP]
2. Traceroute [General/TCP]
3. Services [22tcp]
4. Check open ports [22tcp]
5. RPC portmapper [111tcp]
6. Obtain list of all port mapper registered programs via RPC [111tcp]
7. CPE inventory [General CPE-T]
8. Services [3306tcp]
9. MySQL/MariaDB Detection [3306tcp]
10. Database Open Access Vulnerability[3306tcp]
11. Services [80tcp]
12. Apache web server version detection[80tcp]
13. CGI scanning Consolidation[80tcp]
14. HTTP security Headers Detection[80tcp]
15. Nikto (NASL wrapper) [80tcp]
16. DIRB (NASL wrapper) [80tcp]
17. Check open ports [80tcp]
18. Services TLS [443tcp]
19. Services SSL [443tcp]
20. Apache web server detection [443tcp]
21.CGI scanning Consolidation [443tcp]
22. HTTP Security Headers Detection [443tcp]
23.Nikto (NASL Wrapper) [443tcp]
24. DIRB (NASL wrapper) 443[tcp]
25. Check open ports [443tcp]

Στον server https://application.hua.gr με IP 83.212.240.17 εκτελεστηκαν οι παρακάτω ρουτίνες και βρεθηκαν οι ιδιες ευπαθειες **ομοίως εσωτερίκα και εξωτερικά** του δικτύου με επίπτωση μετρίας επικινδυνότητας:

1. Untrusted SSL/TLS Certicate Authorities [8181tcp]
2. SSL/TLS Report weak cipher suites [9996tcp]
3. SSL/TLS report vulnerable cipher suites for HTTPS [443tcp]
4. SSL/TLS report vulnerable cipher suites for HTTPS [8181tcp]
5. SSL/TLS Report weak cipher suites [8181tcp]
6. SSL/TLS Diffie-Hellman Key exchange Insufficient DH Group Strength [8181tcp]
7. Untrusted SSL/TLS Certicate Authorities [4848tcp]
8. SSL/TLS report vulnerable cipher suites for HTTPS [4848tcp]
9. Oracle Glass Fish Server Directory Traversal [4848tcp]
10. SSL/TLS Report weak cipher suites [4848tcp]
11. SSL/TLS Diffie-Hellman Key exchange Insufficient DH Group Strength [4848tcp]

Στην συνέχεια ακολουθεί μια σύνοψη των κρισιμότερων ευπαθειών, όπου αυτές βρέθηκαν, από τις αναφορές του εργαλείου *OpenVas* για κάθε ένα διακομητή. Σε κάθε περίπτωση γίνεται μια ανάλυση της/των κρισιμότερων ευπαθειών, την φύση αυτών καθώς και την επικινδυνότητα τους, ενώ τέλος δίνεται και μια προτεινόμενη διαδικασία επίλυσης της εκάστοτε περίπτωσης.

- Για τον διακομιστή *www.hua.gr* παρατηρούμε εύρεση δύο ευπαθειών με σήμανση κρισιμότητας υψηλού κινδύνου όπου αναλύονται στην συνέχεια.
    1. Alternative PHP File Extensions File Upload and Information Disclosure Vulner- abilities.

       Η συγκεκριμένη ευπάθεια καταρχάς γίνεται αντιληπτή από το εργαλείο λόγο της δυνατότητας εύρεσης της έκδοσης του CSM Joomla από αυτό. Στην τωρινή έκδοση της πλατφόρμας υπάρχει η γνωστή ευπάθεια αυτή. Η επικινδυνότητα της ευπάθειας είναι αρκετά υψηλή, καθώς εάν ο επιτιθέμενος στο σύστημα την αξιοποιήσει, μπορεί να επιτύχει το ανέβασμα και ως εκ του του την

εκτέλεση δικού του php κώδικα *(μέσω επεκτάσεων *.php6, *.php7, *.phtml και *.phpt)*. Αποτελεί δηλαδή μια τρύπα στον διακομιστή όπου ο χρήστης μπορεί να προσθέσει και να εκτελέσει δίκες του λειτουργίες πάνω στον ίδιο τον διακομιστή. Το συγκεκριμένο πρόβλημα υπάρχει εξαιτίας σφάλματος στον μηχανισμό της πλατφόρμας όπου ελέγχει ορθός το περιεχόμενο των αρχείων όπου ανεβαίνουν στον διακομιστή. Η απλούστερη και ποιο ορθή διόρθωση της ευπάθειας είναι η αναβάθμιση της εγκατεστημένης έκδοσης της πλατφόρμας *Joomla* στην επομένη ή ακόμα καλύτερα τελευταίας της έκδοσης. Η παραπάνω ενέργεια θα διορθώσει το πρόβλημα αυτό αλλά πιθανός και άλλες γνωστές ευπάθειας προηγούμενων εκδόσεων.

2. Core two-factor Authentication Bypass Vulnerability.

   Η συγκεκριμένη ευπάθεια καταρχάς γίνεται και πάλι αντιληπτή από το εργαλείο λόγο της δυνατότητας εύρεσης της έκδοσης του CSM Joomla από αυτό. Η επικινδυνότητα της ευπάθειας είναι αρκετά υψηλή, καθώς εάν ο επιτιθέμενος στο σύστημα την αξιοποιήσει, μπορεί να επιτύχει την προσπέλαση των μηχανισμός ασφαλείας της πλατφόρμας σχετικά με την προστασία ενεργειών ενός χρήστη, όπου μπορεί να οδηγήσει τον επιτιθέμενο σε περεταίρω δικαιώματα εκτέλεσης μη εξουσιοδοτημένων ενεργειών. Το συγκεκριμένο πρόβλημα υπάρχει εξαιτίας σφάλματος στον μηχανισμό της πλατφόρμας όπου σχετίζεται με την λειτουργία των δύο επιπέδων ταυτοποίησης ενός χρήστη. Η απλούστερη και ποιο ορθή διόρθωση της ευπάθειας είναι η αναβάθμιση της εγκατεστημένης έκδοσης της πλατφόρμας *Joomla* στην επομένη ή ακόμα καλύτερα τελευταίας της έκδοσης. Η παραπάνω ενέργεια θα διορθώσει το πρόβλημα αυτό αλλά πιθανός και άλλες γνωστές ευπάθειας προηγούμενων εκδόσεων.

- Για τον διακομιστή *application.hua.gr* παρατηρούμε εύρεση αρκετών ευπαθειών με σήμανση κρισιμότητας μεσαίου κινδύνου. Στην συνέχεια αναλύεται μια εξ αυτών ως παράδειγμα αναφοράς των αποτελεσμάτων.

    1. Untrusted SSL/TLS Certicate Authorities.

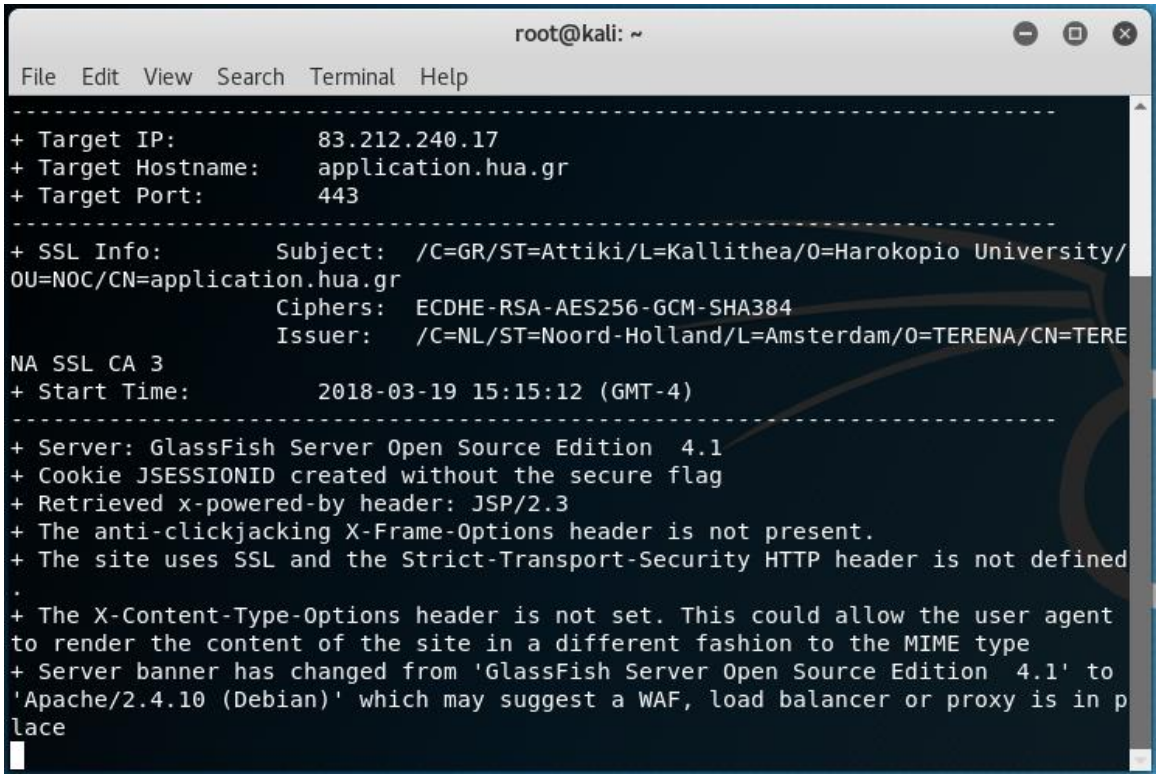
       Η συγκεκριμένη ευπάθεια καταρχάς γίνεται αντιληπτή από το εργαλείο λόγο της εύρεσης υπογραφής στο SLL/TLS πρωτόκολλο από μη έμπιστή υπηρεσία. Η επικινδυνότητα της ευπάθειας αν και είναι μεσαίου κίνδυνου είναι σημαντική, καθώς εάν ο επιτιθέμενος στο σύστημα την αξιοποιήσει, μπορεί να επιτύχει με τεχνικές επίθεσης man in the middle, πρόσβαση σε κρίσιμες πληροφορίες και ως αποτέλεσμα εκτέλεσης περαιτέρω επιθέσεων, όπου μπορεί να οδηγήσει τον επιτιθέμενο σε περεταίρω δικαιώματα εκτέλεσης μη εξουσιοδοτημένων ενεργειών. Η απλούστερη και ποιο ορθή διόρθωση της ευπάθειας είναι η διώρθωση της παραπάνω υπογραφής στο SLL/TLS πρωτόκολλο από έμπιστή υπηρεσία.

- Τέλος για τον διακομιτή *webms.hua.gr* παρατηρούμε εύρεση αρκετών ευπαθειών με σήμανση κρισιμότητας μη υπαρκτού κινδύνου *(επίπεδο αναφοράς log)*.

### ➢ Αποτελέσματα Nikto

Για την λήψη αποτελέσματων του *Nikto,* ο χρήστης πρέπει να κάνει χρήση απλών εντολών μέσω ενός παραθύρου *terminal.* Ο χρήστης εισάγη μια εντολή της παραπάνω μορφής ώστε να λάβει τα αποτελέσματα

$$nikto - h < domain|ip >$$




*Εικόνα 29- Αποτελεσματα Nikto*

Στον ακόλουθο πίνακα διακρίνουμε μέρους του σύνολου των αποτελεσμάτων εύρεσης ευπαθειών από το εργαλείο *Nikto*, όπου έγιναν στους ακόλουθους τρεις διακομιστές καθώς και συνοπτικά το σύνολο των ευπαθειών όπου βρέθηκαν, αλλά και την κρισιμότητα αυτών.

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          83.212.240.16
+ Target Hostname:    www.hua.gr
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:                    Subject:  /C=GR/ST=Attiki/L=Kallithea/O=Harokopio
University/OU=noc/CN=*.hua.gr
                    Ciphers:  ECDHE-RSA-AES256-GCM-SHA384
                    Issuer:   /C=NL/ST=Noord-Holland/L=Amsterdam/O=TERENA/CN=TERENA SSL CA 3
+ Start Time:         2018-03-04 15:52:20 (GMT2)
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect
against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ OSVDB-3092: /manual/: Web server manual found
+ Cookie 80e21241be726e559f37c7bf8f7df3e2 created without the secure flag
  + Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final
release) and 2.2.29 are also current.
+ /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed or renamed.
+ End Time:           2018-03-04 16:18:28 (GMT2) (1568 seconds)
+ 1 host(s) tested
```

Παρακάτω θα αναφέρθουν οι κρισιμότερες ευπάθειες που βρέθηκαν για τους 3 διακομιστές ξεχωριστά με την χρήση του εργαλειου Nikto:

*Για τον SERVER* ***www.hua.gr*** ***εσωτερικα μεσω vpn:***

1 The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS.

Η κεφαλίδα απόκρισης HTTP X-XSS-Protection είναι μια λειτουργία του Internet Explorer, του Chrome και του Safari που σταματά τις σελίδες από τη φόρτωση όταν εντοπίζουν επιθέσεις αντανακλάσεων μεταξύ ιστότοπων (XSS). Παρόλο που αυτές οι προστασίες είναι σε μεγάλο βαθμό περιττές στα σύγχρονα προγράμματα περιήγησης όταν οι ιστότοποι εφαρμόζουν μια ισχυρή Πολιτική Ασφάλειας Περιεχόμενο που απενεργοποιεί τη χρήση του ενσωματωμένου JavaScript ("unsafe-inline"), μπορούν ακόμα να παρέχουν προστασία για χρήστες παλαιότερων περιηγητών ιστού που δεν έχουν ακόμα υποστήριξη CSP

2.The anti-clickjacking X-Frame-Options header is not present.

Η κεφαλίδα απόκρισης HTTP X-Frame-Options μπορεί να χρησιμοποιηθεί για να υποδείξει εάν ένα πρόγραμμα περιήγησης θα πρέπει να επιτρέπεται να εκτυπώνει μια σελίδα σε πλαίσιο <frame>, <iframe> ή <object>. Οι ιστότοποι μπορούν να το χρησιμοποιήσουν για να αποφύγουν τις επιθέσεις clickjacking, διασφαλίζοντας ότι το περιεχόμενό τους δεν είναι ενσωματωμένο σε άλλους ιστότοπους. Η προστιθέμενη ασφάλεια παρέχεται μόνο εάν ο χρήστης που έχει πρόσβαση στο έγγραφο χρησιμοποιεί πρόγραμμα περιήγησης το οποίο υποστηρίζει επιλογές X-Frame.

3. Cookie 80e21241be726e559f37c7bf8f7df3e2 created without the secure flag

Τα ασφαλή cookies είναι ένας τύπος cookie που μεταδίδεται μέσω κρυπτογραφημένης σύνδεσης HTTP. Κατά τη ρύθμιση του cookie, το χαρακτηριστικό ασφαλείας δίνει εντολή στο πρόγραμμα περιήγησης ότι το cookie πρέπει να επιστρέφεται μόνο στην εφαρμογή μέσω κρυπτογραφημένων συνδέσεων. Το ασφαλές χαρακτηριστικό δεν προστατεύει το coookie κατά τη μεταφορά από την εφαρμογή στο πρόγραμμα περιήγησης, τόσο το Firefox όσο και ο Internet Explorer επιτρέπουν τη ρύθμιση των cookies με το χαρακτηριστικό Secure μέσω HTTP.

Για την πλήρη προστασία ενός cookie, το χαρακτηριστικό HttpOnly και SameSite πρέπει επίσης να εφαρμοστεί στο cookie. Το HttpOnly προστατεύει το cookie από την πρόσβαση, για παράδειγμα, από τη JavaScript, ενώ το χαρακτηριστικό SameSite επιτρέπει μόνο το cookie να αποστέλλεται στην εφαρμογή εάν η αίτηση προέρχεται από τον ίδιο τομέα.

4. The site uses SSL and the Strict-Transport-Security HTTP header is not defined

ο HTTP Strict Secure Transport (HSTS) είναι ένα βελτιωτικό ασφάλειας που επιτρέπει την ενεργοποίηση μιας ιστοσελίδας μέσω της χρήσης ειδικής κεφαλίδας απόκρισης. Μόλις ένα υποστηριζόμενο πρόγραμμα περιήγησης λάβει αυτή την κεφαλίδα, το πρόγραμμα περιήγησης θα αποτρέψει την αποστολή οποιωνδήποτε επικοινωνιών μέσω HTTP στον καθορισμένο τομέα και θα στείλει όλες τις επικοινωνίες μέσω HTTPS. Επίσης, εμποδίζει το HTTPS να κάνει αναδρομολόγηση μέσα από τις προτροπές στα προγράμματα περιήγησης.

5. Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.

Το αποτέλεσμα αυτής της ευπάθειας έχει αναφερθεί και αναλυθεί από τα αποτελέσματα με την χρήση του εργαλείου openvas.

6. OSVDB-3092: /manual/: Web server manual found.

Αυτή η αναφορά παραπέμπει στο γεγονός οτι πληκτρολογώντας το URL https://hua.gr/manual

ο επιτιθέμενος μπορεί να ανακτήσει το αναλυτικό manual του Apache server.

7. /htaccess.txt: Default Joomla! htaccess.txt file found. This should be removed or renamed

Για τον SERVER www.hua.gr εξωτερικά βρεθηκε επιπλεον η παρακάτω ευπάθεια:

8. OSVDB-3092: /web.config: ASP config file is accessible.

Η ευπάθεια αυτή είναι πολύ κρίσιμη λόγο του ότι το URL:

https://www.hua.gr/web.config

Μας δίνει τις παρακάτω πληροφορίες,οπού κατά πάση πιθανότητα αν παραμετροποιηθούν μπορεί να οδηγήσουν τον server σε λάθος αποκρίσεις:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
   <location path=".">
   <system.webServer>
       <directoryBrowse enabled="false" />
       <rewrite>
           <rules>
               <rule name="Joomla! Rule 1" stopProcessing="true">
                   <match url="^(.*)$" ignoreCase="false" />
                   <conditions logicalGrouping="MatchAny">
                       <add                          input="{QUERY_STRING}"
pattern="base64_encode[^(]*\([^)]*\)" ignoreCase="false" />
                       <add                          input="{QUERY_STRING}"
pattern="(&gt;|%3C)([^s]*s)+cript.*(&lt;|%3E)" />
                       <add                          input="{QUERY_STRING}"
pattern="GLOBALS(=|\[|\%[0-9A-Z]{0,2})" ignoreCase="false" />
                       <add                          input="{QUERY_STRING}"
pattern="_REQUEST(=|\[|\%[0-9A-Z]{0,2})" ignoreCase="false" />
                   </conditions>
                   <action     type="CustomResponse"     url="index.php"
statusCode="403"  statusReason="Forbidden"  statusDescription="Forbidden"
/>
               </rule>
               <rule name="Joomla! Rule 2">
                   <match url="(.*)" ignoreCase="false" />
                   <conditions logicalGrouping="MatchAll">
                     <add      input="{URL}"      pattern="^/index.php"
ignoreCase="true" negate="true" />
                     <add  input="{REQUEST_FILENAME}"  matchType="IsFile"
ignoreCase="false" negate="true" />
                     <add                      input="{REQUEST_FILENAME}"
matchType="IsDirectory" ignoreCase="false" negate="true" />
                   </conditions>
                   <action type="Rewrite" url="index.php" />
               </rule>
           </rules>
       </rewrite>
   </system.webServer>
   </location>
</configuration>
```

Τα σημαντικότερα αποτελέσματα για τον server webms.hua.gr εξωτερικά και εσωτερικά μέσω vpn είναι ομοίως χωρίς διαφοροποίηση τα παρακάτω:

1 The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS

Η ευπάθεια εχει αναφερθεί ομοίως παραπάνω.

2. The anti-clickjacking X-Frame-Options header is not present.

Η ευπάθεια έχει αναφερθεί ομοίως παραπάνω.

3. Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current

Η ευπάθεια έχει αναφερθεί ομοίως παραπάνω.

4. Allowed HTTP Methods: GET, HEAD, POST, OPTIONS

**5.** OSVDB-637: Enumeration of users is possible by requesting ~username (responds with 'Forbidden' for users, 'not found' for non-existent users)

Τα σημαντικότερα αποτελέσματα για τον server application.hua.gr εσωτερικά του δικτύου παρουσιάζονται παρακάτω:

1.The anti-clickjacking X-Frame-Options header is not present.

Η ευπάθεια έχει αναφερθεί παραπάνω.

**2.** The site uses SSL and the Strict-Transport-Security HTTP header is not defined.

Η ευπάθεια έχει αναφερθεί παραπάνω

3. Cookie JSESSIONID created without the secure flag

Η παραπάνω ευπάθεια υπόκειται στο γεγονός ότι η java πλατφόρμα όπως π.χ tomcat, χρησιμοποιεί εξ αρχής cookies.Όταν κάποιος χρήστης για πρώτη φορά αποκτά πρόσβαση στην web εφαρμογή ,ένα session δημιουργείτε το οποίο βασίζεται στην επικοινωνία μέσω HTML,JSP(java server platform) η με ένα servlet. Εάν τώρα ένας χρήστης ζητήσει κάτι από την βάση με την σειρά της η πλατφόρμα απαντάει στο αίτημα

με ένα request.getSession (true) το οποίο ουσιαστικά είναι το session που δημιουργείτε εφόσον δεν υπάρχει. Στην συνέχεια αποστέλλεται στον client το cookie id το οποίο αποθηκεύεται στον browser του χρήστη. Κάθε φορά που ο χρήστης κάνει log off από την συνεδρία το παλιό cookie id καταστρέφεται αυτόματα.

Το secure flag για τον server έχει σημασία λόγο του ότι θέτοντας ενεργή την επιλογή 'secure flag' οι περιηγητές οι οποίοι υποστηρίζουν αυτή την επιλογή θα αποστείλουν αιτήματα μόνο μέσα από https socket της ιστοσελίδας , αποφεύγοντας έτσι να αποστέλλονται πληροφορίες μέσα από μη κρυπτογραφημένα sessions http. Θέτοντας ενεργό το secure flag αποφεύγουμε την πιθανότητα επικοινωνίας μέσω μη κρυπτογραφημένων καναλιών

4. Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH

5. OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server

6.HTTP method: 'PATCH' may allow client to issue patch commands to server

7. The Content-Encoding header is set to "deflate" this may mean that the server is vulnerable to the BREACH attack

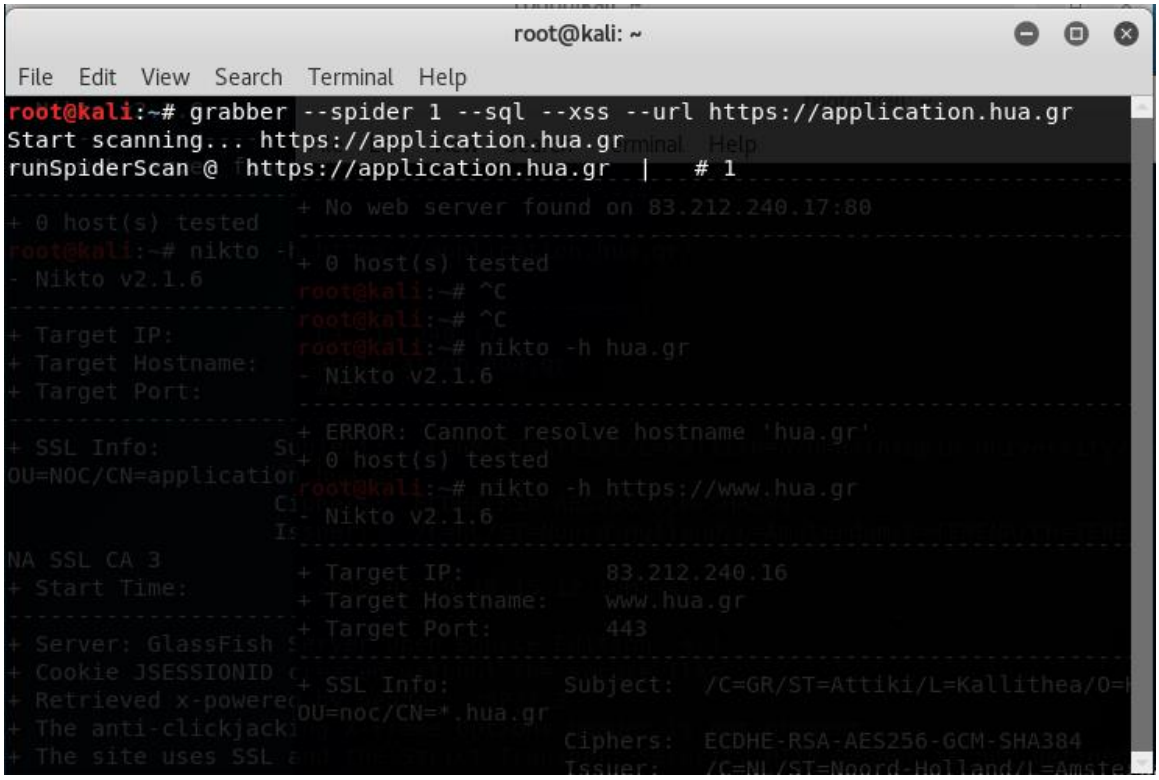
8. /nsn/..%5Cutil/dsbrowse.bas: Netbase util access is possible which means that several utility scripts might be run (**including directory listings**, NDS tree enumeration and running .bas files on server

**9.** /..\..\..\..\..\..\temp\temp.class: Cisco ACS 2.6.x and 3.0.1 (build 40) allows authenticated remote users to retrieve any file from the system. Upgrade to the latest version.

### ➢ Αποτελέσματα Grabber

Για την λήψη αποτελεσμάτων του εργαλείου *Grabber,* ο χρήστης πρέπει να κάνει χρήση απλών εντολών μέσω ενός παραθύρου *terminal.* Ο χρήστης εισάγει μια εντολή της παραπάνω μορφής ώστε να λάβει τα αποτελέσματα

$$grabber < option > --url < domain|ip >$$




*Εικόνα 30 –Αποτελεσματα Grabber*

Στον ακόλουθο πίνακα διακρίνουμε μέρους του σύνολου των αποτελεσμάτων εύρεσης ευπαθειών από το εργαλείο *Grabber*, όπου έγιναν στους ακόλουθους τρεις διακομιστές καθώς και συνοπτικά το σύνολο των ευπαθειών όπου βρέθηκαν, αλλά και την κρισιμότητα αυτών.

```
Start scanning... https://application.hua.gr
runSpiderScan @ https://application.hua.gr | # 1
runSpiderScan @ https://application.hua.gr | # 0
runSpiderScan @ https://application.hua.gr/user/login | # 0
runSpiderScan @ https://application.hua.gr/?lang=el_GR | # 0
runSpiderScan @ https://application.hua.gr/?lang=en_US | # 0
runSpiderScan @ https://application.hua.gr/).length,t.html | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/).length,t.html
runSpiderScan @ https://application.hua.gr/).length,t.htm | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/).length,t.htm
runSpiderScan @ https://application.hua.gr/(function(e,t){var n,r,i=typeof t,o=e.location,a=e.do | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/(function(e,t){var n,r,i=typeof t,o=e.location,a=e.do
runSpiderScan @ https://application.hua.gr/)[0],r.style.css | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/)[0],r.style.css
runSpiderScan @ https://application.hua.gr/],_default:x.support.html | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/],_default:x.support.html
runSpiderScan @ https://application.hua.gr/],_default:x.support.htm | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/],_default:x.support.htm
runSpiderScan @ https://application.hua.gr/),filter:function(e,n){var r,i,o,s=n.button,l=n.fromElement;return
null==e.pageX&&null!=n.clientX&&(i=e.target.ownerDocument||a,o=i.do | # 0
```

```
Grabber cannot retrieve the given url:
https://application.hua.gr/),filter:function(e,n){var
r,i,o,s=n.button,l=n.fromElement;return
null==e.pageX&&null!=n.clientX&&(i=e.target.ownerDocument||a,o=i.do
runSpiderScan @
https://application.hua.gr/],function(e,t){x.propHooks[t]={get:function(e){return
e.getAttribute(t,4)}}}),x.support.style||(x.attrHooks.style={get:function(e){return
e.style.css | # 0
…
…
Grabber cannot retrieve the given url:
https://application.hua.gr/],function(e,t){x.propHooks[t]={get:function(e){return
e.getAttribute(t,4)}}}),x.support.style||(x.attrHooks.style={get:function(e){return
e.style.css
runSpiderScan @
https://application.hua.gr/,data:n}).done(function(e){o=arguments,s.html | # 0
Grabber cannot retrieve the given url:
https://application.hua.gr/,data:n}).done(function(e){o=arguments,s.html
runSpiderScan @
https://application.hua.gr/,data:n}).done(function(e){o=arguments,s.htm | # 0
Grabber cannot retrieve the given url:
https://application.hua.gr/,data:n}).done(function(e){o=arguments,s.htm
runSpiderScan @
https://application.hua.gr/!==l||x.contains(e.ownerDocument,e)||(l=x.style(e,n)),Yt.t
est(l)&&Ut.test(n)&&(i=u.width,o=u.minWidth,a=u.maxWidth,u.minWidth=u.maxWid
th=u.width=l,l=s.width,u.width=i,u.minWidth=o,u.maxWidth=a)),l}):a.do | # 0
Grabber cannot retrieve the given url:
https://application.hua.gr/!==l||x.contains(e.ownerDocument,e)||(l=x.style(e,n)),Yt.t
```

```
est(l)&&Ut.test(n)&&(i=u.width,o=u.minWidth,a=u.maxWidth,u.minWidth=u.maxWidth=u.width=l,l=s.width,u.width=i,u.minWidth=o,u.maxWidth=a)),l}):a.do
runSpiderScan @ https://application.hua.gr/,function(n,r,i){var o,a,s,l=n.js | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/,function(n,r,i){var o,a,s,l=n.js
runSpiderScan @ https://application.hua.gr/===x.css | # 0
Grabber cannot retrieve the given url: https://application.hua.gr/===x.css
runSpiderScan @ https://application.hua.gr/* Includes: jquery.ui.core.js | # 0
…
…
Method = GET https://application.hua.gr
[Cookie] 0 : <Cookie JSESSIONID=fc3f5d3debe9b3f46cfb9b574db6 for application.hua.gr/>
Method = GET https://application.hua.gr
[Cookie] 0 : <Cookie JSESSIONID=fc3f5d3debe9b3f46cfb9b574db6 for application.hua.gr/>
[Cookie] 1 : <Cookie locale="redirect_302_/a.jpg http://victimsite.com/admin.asp&deleteuser" for application.hua.gr/>
[Cookie] 0 : <Cookie JSESSIONID=fc3f5d3debe9b3f46cfb9b574db6 for application.hua.gr/>
[Cookie] 1 : <Cookie locale="redirect_302_/a.jpg http://victimsite.com/admin.asp&deleteuser" for application.hua.gr/>
```

### 3.2.4 Επιπτώσεις πληροφοριακού συστήματος

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, αλλά και επιβεβαιώθηκε από την διαδικασία η οποία προηγήθηκε στην αρχή του κεφαλαίου, η μη εξουσιοδοτημένη αποκάλυψη πληροφοριών και δεδομένων κρούει τον κώδωνα του κινδύνου για το πληροφοριακό Σύστημα διότι εκτίθεται η ασφάλεια. Όταν το νομικό αυτό δικαίωμα ριψοκινδυνεύει και γίνεται ευάλωτο τότε υπάρχει ευπαθή άμυνα που σημαίνει πως κάποιος τρίτος έχει την ικανότητα να εκμεταλλευτεί αυτό το μειονέκτημά και να προκαλέσει σχετικές φθορές και απώλειες στο σύστημα. Θεωρείται πως όταν ένα σύστημα παραβιασθεί από κάποιο μη εξουσιοδοτημένο πρόσωπο εκμεταλλεύονται την ευπάθεια, διαπράττει επίθεση. Αυτό πρέπει να εξαλειφθεί στο έπακρο με διαρκή και συνεχή συστηματικό έλεγχο. Αυτός ο έλεγχος είναι ένα από τα πιο προτεινόμενα προστατευτικά μέτρα που μπορεί να υλοποιηθεί μέσω ενεργειών, είτε συσκευών, είτε τεχνικών διαδικασιών. Τα συστήματα περιέχουν τρία αντικείμενα και τέσσερα είδη απειλών τους. Τα αντικείμενα ακούν στα ονόματα υλικό, λογισμικό και δεδομένα ενώ οι απειλές ακούν στα ονόματα διακοπή, παρεμπόδιση, αλατοποίηση και τροποποίηση, στα οποία θα εξετάσουμε τις έννοιες τους παρακάτω.

- Διακοπή

  Με τη διακοπή σημαίνει πως τα αντικείμενα που αναφέραμε παραπάνω δεν είναι στη διάθεση χρήσης και αυτό γιατί μάλλον χάνονται. Αυτό ενδέχεται να συμβεί όταν κάποιος προσπαθήσει να καταστρέψει τη συσκευή αυτή ή να διαγράψει πρόγραμμα με αρχεία και δεδομένα ώστε να αποσκοπήσει στη διακοπή λειτουργία του συστήματος.

- Παρεμπόδιση

  Με την παρεμπόδιση σημαίνει ότι κάποια πρόσωπα που δεν έχουν δικαίωμα εξουσιοδοτημένης πρόσβασης έχουν αποκτήσει αυτό το προνόμιο για αυτό το σύστημα. Δεν είναι αναγκαίο αυτά τα πρόσωπα να είναι απαραίτητα άτοκά, ενδέχεται να είναι είτε προγράμματα τους, είτε κάποιο άλλο πληροφοριακό σύστημα. Αυτή η αποτυχία μπορεί να προέλθει από την παράνομή αντιγραφή προγραμμάτων ή δεδομένων, όμως μπορεί να προέλθει και από υποκλοπές

συνομιλιών ώστε να αποκτηθούν διάφοροι κωδικοί που είναι απαραίτητοι για την πρόσβαση στο εκάστοτε δίκτυο.

- Πλαστογράφηση.

  Με την πλαστογράφηση σημαίνει πως κάποια ομάδα μη εξουσιοδοτημένων χρηστών έχει την ικανότητα να πλαστογραφήσει αρχεία δεδομένων σε ένα σύστημα. Οι εισβολείς μπορούν να κάνουν αλλαγές σε εγγραφές στις βάσεις δεδομένων όμως γρήγορα σε μικρό χρονικό διάστημα μπορεί να διαπιστωθεί πρακτικά από τους εξουσιοδοτημένους πως όντως αυτές οι εγγραφές είναι πλαστές, εκτός και αν οι παρανοείς είναι τόσο περίτεχνοι που τα χαρακτηριστικά δε διαφέρουν καθόλου από τα γνήσια

# Κεφάλαιο 4 – Συνολικά

Στο τελευταίο κεφάλαιο της διατριβής, γίνεται η σύνοψη και η αξιολόγηση των πληροφοριών όπου αντλήθηκαν σχετικά με την ασφάλεια και τις ευπάθειες των πληροφορικών συστημάτων κατά τα προηγούμενα. Στην συνέχεια του κεφαλαίου και αφού έχει προηγηθεί μια ανάλυση των εύρεσης και πιθανής αντιμετώπισης ευπαθειών, γίνεται μια συμπερασματική αναφορά στον τρόπο όπου πρέπει στις μέρες μας, τόσο οι διαχειριστές των πληροφοριακών συστημάτων όσο και οι χρήστες αυτών να λειτουργούν πάνω σε αυτά, με στόχο της διασφάλισης της μειώσεις των ευπαθειών αυτών ή ακόμα και της θωράκισης αυτών.

## 4.1 Συμπεράσματα

Μετά το πέρας της ολοκλήρωσης της αναζήτησης ευπαθειών στο πληροφοριακό σύστημα του Ιδρύματος και τη καταγραφή τους, καταλήγουμε στις εξής παρατηρήσεις:

- Πρέπει να γίνεται συνεχόμενος έλεγχος ασφαλείας στα συστήματα.
- Να πραγματοποιείται τακτική αναβάθμιση των λειτουργικών συστημάτων και εφαρμογών όπου χρησιμοποιούνται και πλαισιώνουν την/της εφαρμογές του συστήματος. Όπως και αναλύθηκε εκτενώς στο κεφάλαιο *3.2.3* και τις υπό-ενότητες αυτού, οι περισσότερες σημαντικές ευπάθειες όπου εντοπίστηκαν σχετίζονται με την μη τακτική αναβάθμιση των υπηρεσιών και τον στοιχείων αυτών.
- Την ενημέρωση των διαφόρων χρηστών για την ορθή χρήση των συστημάτων. Σημαντικός παράγοντας είναι και η τακτική ενημέρωση των χρηστών ενός συστήματος για την ορθή χρήση μιας υπηρεσίας ή εφαρμογής. Πολλές φορές ένα σύστημα μπορεί να έχει όλα τα αντίμετρα όπου χρειάζονται για την αντιμετώπιση πιθανών ευπαθειών, αλλά πολλές φορές οι ίδιοι οι χρήστες θέτουν ένα σύστημα σε κίνδυνο μέσω της λάθος χρήσης αυτού.
- Συνεχή παρακολούθηση των τεχνικών διείσδυσης σε πληροφοριακά συστήματα και εφαρμογές για την έγκαιρη ενημέρωση και εξάλειψη των ευπαθειών που εμφανίζονται. Όπως αναλύεται και στο κεφάλαιο *3* τα εργαλεία *pentetration*

*testing* προσπαθούν συνεχώς να αναβαθμίζονται σχετικά με τις νέες ευπάθειες και τεχνικές διείσδυσης σε πληροφοριακά συστήματα. Βεβαίως οι pentesters πρέπει να είναι σε θέση πέραν της χρήσης έτοιμων εργαλείων να αναγνωρίζουν και να προλαμβάνουν αντίστοιχες ευπάθειες συστημάτων όπου τα εργαλεία είτε δεν είναι σε θέση να αντιμετωπίζουν είτε να αναγνωρίζουν.

## 4.2 Επίλογος

Είναι γεγονός ότι, παρά την προφανή της χρησιμότητα, η λήψη των απαραίτητων μέτρων ασφάλειας δημιουργεί πολλές φορές κάποια πρόσθετη επιβάρυνση στην απόδοση και το κόστος λειτουργίας του πληροφοριακού συστήματος του οργανισμού. Θα πρέπει ακόμη να αποδεχτούμε το κόστος της ασφάλειας και ως κόστος χρόνου και ως κόστος χρήματος. Συνεπώς, μπορεί να θεωρηθεί ότι η ασφάλεια βρίσκεται σε σχέση αντιστρόφως ανάλογη με την αποδοτικότητα του πληροφοριακού συστήματος του οργανισμού. Αυτό όμως δεν είναι σωστό γιατί η ασφάλεια είναι κόστος αναγκαίο για την ομαλή και εύρυθμη λειτουργία του. Το συγκεκριμένο κόστος για την ασφάλεια των πληροφοριακών συστημάτων ενός οργανισμού εξαρτάται και προκύπτει από την εκάστοτε ακολουθούμενη πολιτική ασφάλειας. Απαιτείται συνεπώς μια πολιτική ασφάλειας η οποία θα πρέπει να εξισορροπεί το κόστος εισαγωγής ασφάλειας από την μία πλευρά και το κόστος ζημιών από πιθανολογούμενο κίνδυνο από την άλλη. Επίσης, θα πρέπει να δημιουργούνται τέτοιες συνθήκες ασφάλειας ώστε να μη παρεμποδίζεται η ευελιξία και η ανάπτυξη του οργανισμού. Η αναγκαία πολιτική ασφάλειας καθορίζεται από μία δυναμική εκτίμηση του κόστους των μέτρων ασφάλειας σε σχέση με τις συνέπειες που θα έχει για τον οργανισμό οποιαδήποτε πρόκληση δυσλειτουργίας. Ο βασικός αυτός κανόνας ισχύει για όλους τους τομείς και όλα τα επίπεδα ασφάλειας. Εάν η τιμή των δύο αυτών παραμέτρων είναι υψηλή, τότε πρέπει απαραίτητα να ληφθούν

μέτρα, ανεξάρτητα από το κόστος πρόληψης. Τέλος, πρέπει να σημειωθεί ότι η ασφάλεια χαρακτηρίζεται από την φύση της ως δυναμική παράμετρος και όχι στατική, καθώς η τεχνολογία, ο ανταγωνισμός, η πολυπλοκότητα των πληροφοριακών συστημάτων και η ολοένα βελτιούμενη επιτηδειότητα των 'επιτιθέμενων', απαιτούν τη λήψη νέων και συνεχώς αυστηρότερων μέτρων ασφάλειας. Συνεπώς, η ακολουθούμενη πολιτική ασφαλείας θα πρέπει να επανεξετάζεται τακτικά και να διορθώνεται όπου αυτό κρίνεται απαραίτητο. Κλείνοντας, όντας σε καιρούς όπου η ασφάλεια των υπολογιστικών συστημάτων αποκτά ολοένα και μεγαλύτερη σημασία και η ανάπτυξη εργαλείων ελέγχου διείσδυσης παρουσιάζει κάθε χρόνο ραγδαίες εξελίξεις με την εμφάνιση νέων ευπαθειών, καθίσταται η ανάγκη για την εφαρμογή κριτηρίων που θα ταξινομήσουν τα εργαλεία αυτά και θα τα χαρακτηρίσουν ως προς την αποτελεσματικότητα τους ανάμεσα σε άλλα, βάσει του σκοπού της χρήσης τους και του υπόβαθρου των ανθρώπων ασφάλειας που θα τα χρησιμοποιήσουν.

# Βιβλιογραφία

[1] Σωκράτης Κάτσικας & Δημήτρης Γκρίτζαλης & Στέφανος Γκρίτζαλης, Ασφάλεια πληροφοριακών συστημάτων, 2004

[2] Κάτσικας Σ., Γκρίτζαλης Δ., "Ασφάλεια Πληροφοριακών Συστημάτων Υγείας", Ασφάλεια Πληροφοριών, Τεχνικά, Νομικά και Κοινωνικά θέματα, Εκδόσεις ΕΠΥ, Αθήνα, 1995

[3] Joel Scambray Stuart McClure Ebook Hacking Exposed Windows third Edition Widows Security Secrets & Solutions , 2007

[4] Καλαμπούνια Χ., Έλεγχος Τρωτότητας Δικτυοκεντρικών Πληροφοριακών Συστημάτων, Διπλωματική εργασία, Σχόλη Θετικών Επιστήμων και Τεχνολογίας Ελληνικό Ανοιχτό Πανεπιστήμιο, Πάτρα, 2010

[5] David Kennedy , Jim O'Gorman , Devon Kearns , Mati Aharoni - Metasploit The Penetration Tester's Guide, 2012

[6] Rash, Michael et al, Intrusion Prevention and Active Response: Deployment Network and Host IPS, Syngress, 2005.

[7] Google Trends, "Compare," [Online]

[8] http://www.openvas.org/documentation.html

[9] https://www.kali.org

[10] https://en.wikipedia.org/wiki/Nikto_Web_Scanner