«ΠΡΟΓΡΑΜΜΑ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΕΡΕΥΝΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΣΕ ΝΕΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ (ΠΑΒΕΤ-ΝΕ-2004)»

ΕΣΤΙΑ: «Μία ολοκληρωμένη πλατφόρμα ελέγχου ασφάλειας υπολογιστών και παροχής προστασίας στο Διαδίκτυο»

Παραδοτέο Π2.1: «Σχεδιασμός Συστήματος»

Κωδικός Έργου: 04BEN8

Σύντομη περιγραφή: Στο έγγραφο αυτό παρουσιάζουμε το σχεδιασμό του συστήματος ΕΣΤΙΑ, λαμβάνοντας υπόψη τα δεδομένα της φάσης Φ1 όπως αυτά καταγράφονται στο Παραδοτέο Π1.1 (Ανάλυση Απαιτήσεων). Το σύστημα έχει χωριστεί σε υποσυστήματα, καθένα από το οποίο ορίζεται χωριστά. Επίσης ορίζονται οι διεπαφές μεταξύ των υποσυστημάτων. Καθορίζουμε επίσης το εργαλείο διείσδυσης που θα χρησιμοποιεί το σύστημα καθώς και το πως θα ξεπεράσουμε τις ελλείψεις του στα πλαίσια του έργου ΕΣΤΙΑ.

Προβλεπόμενη Ημερομηνία Παράδοσης	**30/09/2005**
Ημερομηνία Παράδοσης	**01/10/2005**
Επίπεδο Ασφάλειας Εγγράφου	**Δημόσιο Έγγραφο**
Συντελεστές	**FORTH, Virtual Trip**

Στο έργο ΕΣΤΙΑ συμμετέχουν οι φορείς:

Virtual Trip	Συντονιστής	Ελλάδα
FORTH	Συνεργάτης	Ελλάδα

Πίνακας Περιεχομένων

Πίνακας Σχημάτων

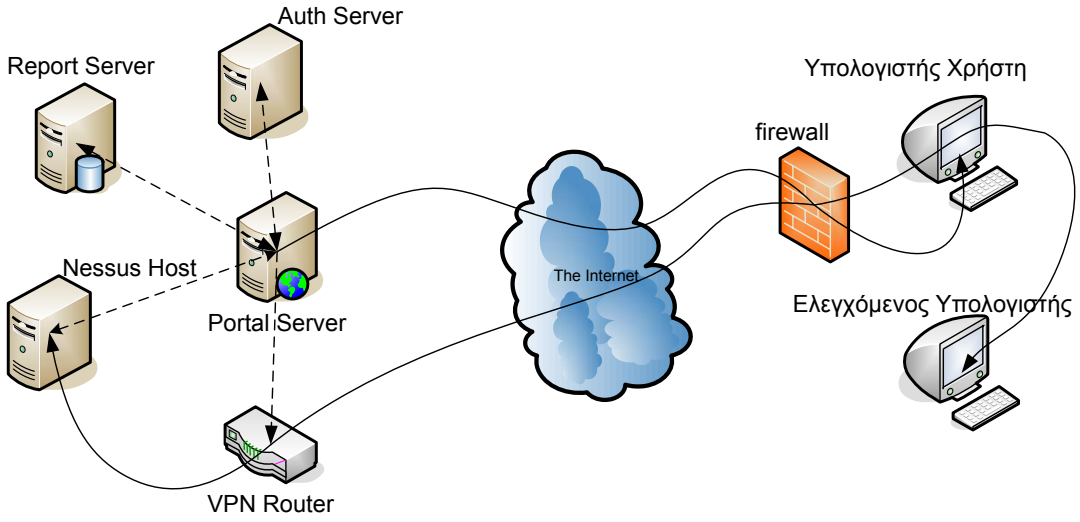
1. Σχεδιασμός συστήματος

Στις παραγράφους που ακολουθούν παρουσιάζεται ο γενικός σχεδιασμός του συστήματος ΕΣΤΙΑ. Ο στόχος του συστήματος ΕΣΤΙΑ είναι η δημιουργία μίας αυτόματης υπηρεσίας ελέγχου και βελτίωσης της ασφάλειας των προσωπικών υπολογιστών. Η υπηρεσία αυτή θα προσφέρεται μέσα από το Διαδίκτυο δίνοντας την δυνατότητα στον απλό χρήστη να ελέγξει και να βελτιώσει την ασφάλεια του υπολογιστή του, αποκτώντας εμπιστοσύνη απέναντι στο μηχάνημά του ειδικότερα και στο Διαδίκτυο γενικότερα.

Πέρα από τους χρήστες που χρησιμοποιούν οικιακούς υπολογιστές, το σύστημα σκοπεύει να παρέχει υπηρεσίες και σε μικρομεσαίες επιχειρήσεις που θέλουν να ελέγξουν και να βελτιώσουν την ασφάλεια των υπολογιστών που χρησιμοποιούν. Οι μικρομεσαίες επιχειρήσεις λόγω του μικρού τους μεγέθους δεν απασχολούν εξειδικευμένο προσωπικό πληροφορικής και επομένως είναι ευάλωτες σε επιθέσεις από το διαδίκτυο. Τα αποτελέσματα μίας τέτοιας επίθεσης μπορεί να είναι καταστροφικά για την επιχείρηση η οποία μπορεί να χάσει όλα τα αρχεία της και επομένως τις δοσοληψίες με τους πελάτες της.

Ο σχεδιασμός γίνεται σε επίπεδο υποσυστημάτων και περιγράφεται η λειτουργικότητα που παρέχει το κάθε υποσύστημα. Λεπτομέρειες για το σχεδιασμό του κάθε υποσυστήματος χωριστά παρουσιάζονται σε επόμενη ενότητα.

Ο σχεδιασμός του συστήματος ΕΣΤΙΑ θα πρέπει να επιτρέπει την κλιμάκωση του συστήματος, ώστε αυτό να μπορεί να καλύψει τις ανάγκες μεγάλου αριθμού χρηστών. Για να το επιτύχουμε αυτό θα πρέπει να αναγνωρίσουμε ποια από τα υποσυστήματα του ενδέχεται να δράσουν ως ανασχετικοί παράγοντες (bottleneck) στη συνολική λειτουργία του συστήματος. Αφού αναγνωρίσουμε τα υποσυστήματα αυτά, θα πρέπει να τα σχεδιάσουμε / επανασχεδιάσουμε, ώστε να λειτουργούν πιο αποτελεσματικά ή/ και να έχουν τη δυνατότητα να κλιμακωθούν ανάλογα με τις ανάγκες όλου του συστήματος.



Σχήμα 1: Σχεδιασμός του συστήματος ΕΣΤΙΑ

Ο σχεδιασμός του συστήματος ΕΣΤΙΑ φαίνεται στο Σχήμα 1. Παρουσιάζουμε τώρα συνοπτικά τα υποσυστήματα που το αποτελούν.

1.1. Ελεγχόμενος Υπολογιστής

Πρόκειται για τον υπολογιστή του οποίου την ασφάλεια επιθυμούμε να ελέγξουμε. Ο υπολογιστής αυτός μπορεί να ταυτίζεται με τον υπολογιστή του χρήστη (βλέπε 1.2 παρακάτω). Επίσης μπορεί να υπάρχουν παραπάνω από ένας ελεγχόμενοι υπολογιστές.

Ιδανικά ο υπολογιστής αυτός θα πρέπει να έχει απευθείας πρόσβαση στο διαδίκτυο. Δηλαδή δεν θα βρίσκεται πίσω από NAT , Firewall ή κάποιο άλλο σύστημα που περιορίζει την πρόσβαση του υπολογιστή στο διαδίκτυο. Σε αυτήν την περίπτωση, δε χρειάζονται επιπλέον ενέργειες από το χρήστη του συστήματος για τον έλεγχο του. Σε αντίθετη περίπτωση όπου ο ελεγχόμενος υπολογιστής (Σχήμα 1 δεξιά κάτω) έχει περιορισμένη πρόσβαση στο παράδειγμα μας (Σχήμα 1) βρίσκεται πίσω από το κόκκινο firewall, απαιτείται η συνεργασία του χρήστη για να ξεπεραστούν οι φραγμοί που υπάρχουν στην επικοινωνία.

1.2. Υπολογιστής Χρήστη

Πρόκειται για τον υπολογιστή στο Σχήμα 1 πάνω δεξιά, τον οποίο χρησιμοποιεί ο χρήστης του συστήματος για να έχει πρόσβαση σε αυτό. Στα περισσότερα σενάρια χρήσης, ταυτίζεται με τον Ελεγχόμενο Υπολογιστή (βλέπε 1.1 παραπάνω). Η πρόσβαση στο σύστημα ΕΣΤΙΑ γίνεται διαμέσου του Παγκόσμιου Ιστού (World Wide Web - WWW) με τη χρήση κάποιου ήδη εγκατεστημένου φυλλομετρητή (web browser).

Στην περίπτωση που η πρόσβαση στο διαδίκτυο του Ελεγχόμενου Υπολογιστή είναι περιορισμένη από κάποιο firewall ή και ανύπαρκτη, ο Υπολογιστής Χρήστη χρησιμοποιείται σαν ενδιάμεσος για να γίνει ο έλεγχος. Σε τέτοιες περιπτώσεις, ο χρήστης θα πρέπει να έχει πρόσβαση στον υπολογιστή του με πλήρη δικαιώματα.

1.3. Portal Server

Ο Portal Server φιλοξενεί την πύλη του συστήματος ΕΣΤΙΑ, την οποία προσπελαύνει ο χρήστης δια μέσου του web browser του. Η πύλη παρέχει στο χρήστη τη δυνατότητα να ξεκινήσει ένα νέο έλεγχο, να παρακολουθήσει την πρόοδο του, καθώς και να ανακτήσεις τα αποτελέσματα παλαιότερων ελέγχων. Όπως φαίνεται και στο Σχήμα 1, ο Portal Server αποτελεί κομβικό σημείο για το σύστημα ΕΣΤΙΑ.

Η αλληλεπίδραση του Portal Server με τα υπόλοιπα υποσυστήματα γίνεται ως εξής:

- Αφού λάβει μια αίτηση ελέγχου από το χρήστη, ο Portal Server την προωθεί στον Auth Server για να ελέγξει αν ο χρήστης δικαιούται να κάνει τον έλεγχο που ζήτησε. Ο Auth Server περιγράφεται στην παράγραφο 2.4.
- Αν ο Auth Server απαντήσει θετικά και ο Ελεγχόμενος Υπολογιστής έχει περιορισμένη πρόσβαση στο διαδίκτυο, τότε ο Portal Server επικοινωνεί με τον VPN Router. Ο VPN Router θα δημιουργήσει μια δίοδο επικοινωνίας μεταξύ του Ελεγχόμενου Υπολογιστή και του συστήματος ΕΣΤΙΑ ώστε να μπορέσει να γίνει ο έλεγχος. Η δημιουργία της διόδου επικοινωνίας απαιτεί τη συνεργασία του χρήστη. Ο VPN Router περιγράφεται στην παράγραφο 1.7.
- Αφού οι τυχόν περιορισμοί στην επικοινωνία με τον Ελεγχόμενο Υπολογιστή έχουν ξεπεραστεί, ο Portal Server προωθεί την αίτηση ελέγχου στο Nessus

Host. Ο Nessus Host θα πραγματοποιήσει τον έλεγχο ασφαλείας και θα επιστρέψει τα αποτελέσματα στον Portal Server. Ο Nessus Server περιγράφεται στην παράγραφο 1.5.

- Ο Portal Server θα επικοινωνήσει τέλος με τον Report Server για να αποθηκεύσει τα αποτελέσματα του ελέγχου για προβολή αργότερα. Ο Report Server περιγράφεται στην παράγραφο 2.6.

1.4. Auth Server

Το υποσύστημα αυτό, όπως αναφέραμε ήδη, είναι υπεύθυνο για να ελέγξει αν ο χρήστης έχει το δικαίωμα να ζητήσει να γίνει ο έλεγχος ασφάλειας στον Ελεγχόμενο Υπολογιστή. Σε περίπτωση που ο Ελεγχόμενος Υπολογιστής και ο Υπολογιστής Χρήστη ταυτίζονται, ο έλεγχος αυτός είναι ευθύς. Σε αντίθετη περίπτωση, θα πρέπει να επιβεβαιωθεί πως ο χρήστης έχει δικαίωμα να ελέγξει τον υπολογιστή που ζήτησε. Η επιβεβαίωση αυτή μπορεί να γίνει ζητώντας από το χρήστη να απαντήσει σε μια πρόκληση που μόνο ο νόμιμος διαχειριστής του Ελεγχόμενου Υπολογιστή θα μπορούσε να απαντήσει. Στην παράγραφο 2.4.2 παρουσιάζουμε αναλυτικά πως γίνεται αυτό.

1.5. Nessus Host[1]

Στο υποσύστημα Nessus Host τρέχει το λογισμικό Nessus που πραγματοποιεί τον έλεγχο ασφαλείας στον Ελεγχόμενο Υπολογιστή. Το λογισμικό Nessus αποτελείται από δύο τμήματα τα οποία συνεργάζονται για να πραγματοποιηθεί ο έλεγχος: τον Nessus Client και τον Nessus Server. Ο Nessus Server είναι αυτός που πραγματοποιεί τον έλεγχο με βάση τις παραμέτρους που θα λάβει από τον Nessus Client. Πολλοί Nessus Clients μπορούν να συνδεθούν στον ίδιο Nessus Server και ο καθένας απ' αυτούς να ζητήσει τον έλεγχο διαφορετικών μηχανημάτων, με διαφορετικές παραμέτρους. Ο Nessus Server έχει τη δυνατότητα να κάνει παράλληλους ελέγχους ασφάλειας σε πολλά μηχανήματα. Θα εξετάσουμε κατά την αναλυτική περιγραφή του

[1] Ονομάζουμε το υποσύστημα αυτό Nessus Host και όχι Nessus Server για να μην υπάρχει σύγχυση με το λογισμικό Nessus Server.

υποσυστήματος Nessus Host σε ποιο βαθμό θα πρέπει να χρησιμοποιήσουμε τη δυνατότητα αυτή για να ολοκληρώσουμε τους ελέγχους στο μικρότερο δυνατό χρόνο.

1.6. Report Server

Ο Report Server χρησιμοποιείται για την αποθήκευση των αποτελεσμάτων του ελέγχου ασφαλείας ώστε να μπορούν να ανακτηθούν αργότερα. Επίσης θα αποθηκεύει στοιχεία που θα επιτρέπουν στους χρήστες να αποδεικνύουν την ταυτότητα τους και να έχουν πρόσβαση στα αποτελέσματα των ελέγχων που έχουν πραγματοποιήσει από οπουδήποτε.

Η παρουσία του εν λόγω υποσυστήματος, αν και δεν είναι απαραίτητη για το όλο σύστημα, μας δίνει τη δυνατότητα να προσφέρουμε βελτιωμένες υπηρεσίες στους χρήστες του. Επιτρέπει, για παράδειγμα, στο χρήστη να μπορεί να κλείσει τον web browser του κατά τη διάρκεια του ελέγχου, που συνήθως είναι μεγάλη. Επίσης χρησιμοποιώντας τα αποθηκευμένα δεδομένα, ο χρήστης θα μπορεί να ενημερώνεται για νέα προβλήματα ασφαλείας που ανακοινώνονται και αφορούν τον υπολογιστή του. Για παράδειγμα, αν κάποιο CERT ανακοινώσει ένα νέο πρόβλημα για μια συγκεκριμένη έκδοση του Microsoft IIS, το σύστημα θα μπορεί να ενημερώσει αυτόματα όλους τους χρήστες που στον τελευταίο έλεγχο που έκαναν παρουσιάζονται να τρέχουν τη συγκεκριμένη έκδοση. Έτσι οι χρήστες του ΕΣΤΙΑ δε θα χρειάζονται να παρακολουθούν οι ίδιοι τις ανακοινώσεις για προβλήματα ασφαλείας.

1.7. VPN Router

Όπως αναφέρθηκε, ο Ελεγχόμενος Υπολογιστής μπορεί να μην έχει πλήρη πρόσβαση στο διαδίκτυο. Στην περίπτωση αυτή ο VPN Router θα δρομολογήσει τα δεδομένα που στέλνει ο Nessus Host προς τον Ελεγχόμενο Υπολογιστή. Για να γίνει αυτό, πρέπει πρώτα να δημιουργηθεί ένα Virtual Private Network (VPN)[8] μεταξύ του VPN Router και ενός υπολογιστή ο οποίος έχει πρόσβαση στον Ελεγχόμενο Υπολογιστή. Αυτό απαιτεί συνεργασία από τη μεριά του χρήστη. Θα πρέπει δηλαδή ο χρήστης να εγκαταστήσει το λογισμικό που παρέχει τη λειτουργικότητα VPN και να το τρέξει χρησιμοποιώντας τις ρυθμίσεις που θα του παρέχουμε. Αν ο χρήστης θέλει

να ελέγξει περισσότερους από έναν υπολογιστές, τότε θα πρέπει επιπλέον να ρυθμίσει τον Υπολογιστή Χρήστη ώστε να προωθεί στο VPN και τα πακέτα όλων των υπολογιστών που θέλει να ελεγχθούν.

2. Σχεδιασμός υποσυστημάτων

Αφού παρουσιάσαμε στην προηγούμενη ενότητα το σχεδιασμό του συστήματος ΕΣΤΙΑ σε επίπεδο υποσυστημάτων, θα παρουσιάσουμε τώρα τις λεπτομέρειες του σχεδιασμού του κάθε υποσυστήματος χωριστά. Για κάθε υποσύστημα θα παρουσιάσουμε πως αλληλεπιδρά με τα άλλα υποσυστήματα κατά τη λειτουργία του συστήματος ΕΣΤΙΑ. Επίσης παρουσιάζουμε, όπου υπάρχουν, τις διαφορετικές ρυθμίσεις/ τεχνολογίες που μπορούν να χρησιμοποιηθούν κατά την υλοποίηση του συστήματος.

2.1. Ελεγχόμενος Υπολογιστής

Ελεγχόμενος υπολογιστής είναι το μηχάνημα το οποίο ο χρήστης ζητάει να ελεγχθεί για ρήγματα ασφαλείας. Ο υπολογιστής αυτός μπορεί είτε να είναι ο προσωπικός υπολογιστής του χρήστη που ζητάει να πραγματοποιηθεί ο έλεγχος είτε κάποιος άλλος υπολογιστής για τον οποίο έχει εξουσιοδότηση να τον ελέγχει. Επίσης κάποιος χρήστης μπορεί να ζητήσει τον έλεγχο ενός μεγάλου αριθμού υπολογιστών π.χ. όλων των υπολογιστών σε ένα υπό-δίκτυο, εφόσον βέβαια έχει την απαραίτητη εξουσιοδότηση.

Στην ιδανική περίπτωση όλοι οι υπολογιστές που είναι προς εξέταση θα είναι συνδεμένοι απ’ ευθείας στο διαδίκτυο ο έλεγχος μπορεί να γίνει άμεσα. Στην περίπτωση που η πρόσβαση στο διαδίκτυο του Ελεγχόμενου Υπολογιστή ή του υπό-δικτύου που θέλουμε να εξετάσουμε είναι περιορισμένη από κάποιο firewall ή και ανύπαρκτη, ο Υπολογιστής Χρήστη χρησιμοποιείται σαν ενδιάμεσος για να γίνει ο έλεγχος. Σε τέτοιες περιπτώσεις, ο χρήστης θα πρέπει να έχει πρόσβαση με πλήρη δικαιώματα στον υπολογιστή του. Η περίπτωση αυτή φαίνεται στο Σχήμα 1 όπου οι υπολογιστές προστατεύονται από ένα firewall.

2.2. Υπολογιστής Χρήστη

Από τον Υπολογιστή Χρήστη γίνεται η εκκίνηση ενός ελέγχου. Επίσης σε αυτόν παρουσιάζεται και η πρόοδος του ελέγχου αυτού. Και οι δύο αυτές διαδικασίες θα

πρέπει να γίνονται μέσα από τον web browser του χρήστη και να έχουν τις μικρότερες δυνατές απαιτήσεις από πλευράς πρόσθετου λογισμικού.

2.2.1. Εκκίνηση Ελέγχου

Η εκκίνηση ελέγχου θα πρέπει να περιλαμβάνει επιλογές με το τι ακριβώς θα ελεγχθεί από το σύστημα ΕΣΤΙΑ. Όπου είναι δυνατό θα πρέπει να συμπεριλαμβάνονται και ενδεικτικοί χρόνοι για κάθε επιλογή. Καθώς οι περισσότεροι χρήστες πιθανό να μη μπορούν να επιλέξουν τον κατάλληλο έλεγχο

2.2.2. Παρουσίαση προόδου

Η παρουσίαση της προόδου ενός ελέγχου θα πρέπει για να είναι πλήρης και κατανοητή να περιέχει τόσο γραφικές ενδείξεις, όσο και επεξηγήσεις με μορφή κειμένου. Η υπάρχουσα τεχνολογία μας δίνει πλήθος επιλογών για να ανακτούμε από τον Portal Server τα στοιχεία που θα παρουσιάζουμε. Συνοπτικά οι επιλογές μας είναι οι ακόλουθες:

- **Μη χρήση δυναμικών στοιχείων:** Μη χρήση δυναμικών στοιχείων σημαίνει πως για να δει ο χρήστης την πρόοδο του ελέγχου που κάνει θα πρέπει να ξαναφορτώνει περιοδικά ολόκληρη τη σελίδα. Η διαδικασία μπορεί να οριστεί να γίνεται περιοδικά με τη χρήση HTML meta-tags. Το πλεονέκτημα αυτής της προσέγγισης είναι πως θα δουλέψει ακόμα και σε web-browsers που δεν υποστηρίζουν τη χρήση γραφικών. Από την άλλη, η συνεχής επαναφόρτωση της σελίδας θεωρείται ενοχλητική για το χρήστη.

- **Δυναμικά στοιχεία Java:** Οι web browsers, με τη χρήση των Java Applets[1] μπορούν να εκτελέσουν προγράμματα Java σε ένα προστατευμένο περιβάλλον. Λόγω της εκτέλεσης σε προστατευμένο περιβάλλον, οι δυνατότητες επικοινωνίας του εκτελούμενου προγράμματος με τον web browser είναι περιορισμένες. Στην περίπτωση μας, το πρόγραμμα Java θα συνδέεται πίσω στον Portal Server και αυτός να το ενημερώνει περιοδικά για την κατάσταση του ελέγχου. Η πληροφορία αυτή θα εμφανίζεται έπειτα στο χρήστη με τη μορφή κειμένου και γραφημάτων.

- **Δυναμικά στοιχεία Flash:** Η τεχνολογία Flash[2] μοιάζει στα χαρακτηριστικά της με τα Java Applets. Επιτρέπει δηλαδή στους web browsers την εκτέλεση προγραμμάτων μέσα σε ένα προστατευμένο περιβάλλον. Όμως, η τεχνολογία Flash επικεντρώνεται περισσότερο στην εκτέλεση πολυμεσικών προγραμμάτων με αποτέλεσμα να δίνει τελικά καλύτερο αισθητικό αποτέλεσμα.

- **Δυναμικά στοιχεία JavaScript:** Τα δυναμικά στοιχεία JavaScript[5], σε αντίθεση με τα δυναμικά στοιχεία Java και Flash, επικοινωνούν απευθείας με τον web browser και έχουν τη δυνατότητα να αλλάξουν δυναμικά τις ιδιότητες και το περιεχόμενο του έγγραφου που προβάλλεται. Στην απλή περίπτωση μπορούν απλά να ζητήσουν να ξαναφορτωθεί ολόκληρη η σελίδα. Επιτρέπουν όμως και να κάνουμε push πληροφορία προς τον web browser, κρατώντας ανοικτή την HTTP σύνδεση και στέλνοντας πάνω από αυτή εντολές JavaScript που αλλάζουν το έγγραφο. Αυτό όμως αντιβαίνει στη λογική του πρωτοκόλλου HTTP πάνω στην οποία έχει βασιστεί η αποδοτική υλοποίηση των web servers. Η χρήση λοιπόν JavaScript για την υλοποίηση push θα πρέπει να αποφευχθεί.

- **Χρήση XML HTTP:** Το XML HTTP[4] είναι μια δυνατότητα των περισσότερων σύγχρονων web browsers που μας επιτρέπει να μεταφέρουμε δεδομένα XML μεταξύ του web browser και κάποιου web server. Η αλληλεπίδραση αυτή είναι ασύγχρονη και γίνεται με πρωτοβουλία του web browser. Επιπλέον με τη χρήση JavaScript, μπορούμε να ενημερώσουμε το έγγραφο που προβάλλει ο browser με την πληροφορία που αποκτήσαμε μέσω του XML HTTP.

Τα πλεονεκτήματα και τα μειονεκτήματα των επιλογών αυτών, παρουσιάζονται συνοπτικά στον επόμενο πίνακα.

Διάσταση Σύγκρισης	**Μέθοδος Παρουσίασης**				
	Χωρίς δυναμικά στοιχεία	Δυναμικά στοιχεία Java	Δυναμικά στοιχεία Flash	Δυναμικά στοιχεία JavaScript	Χρήση XML HTTP
Μεταφερόμενα δεδομένα αρχικά	O(10k)	O(100k)	O(100k)	O(10k)	O(10k)
Μεταφερόμενα δεδομένα ανά ανανέωση	O(10k)	O(100b)	O(100b)	O(10k)	O(100b)
Ανοικτή σύνδεση με τον Portal Server / Δυνατότητα push	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ (μόνο πάνω από HTTP)	ΟΧΙ
Επιπλέον Λογισμικό στον Browser	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ
Συμβατότητα με παλαιότερους web browsers	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ
Αισθητικό αποτέλεσμα	Μέτριο	Μέτριο	Πολύ Καλό	Καλό	Καλό
Δυσκολία υλοποίησης	Μικρή	Μεγάλη	Μεγάλη	Μέτρια	Μέτρια

Πίνακας 1

Το μεγαλύτερο αρχικό κόστος σε μεταφερόμενα δεδομένα έχουν τα δυναμικά στοιχεία Java και Flash, καθώς ουσιαστικά κατεβάζουν από τον Portal Server μια εφαρμογή η οποία τρέχει έπειτα μέσα από το web browser. Όμως έπειτα κάθε ανανέωση είναι φθηνή μια και απαιτείται να μεταφερθούν λίγα δεδομένα και ο αριθμός των ανανεώσεων μπορεί να περιορισθεί στο ελάχιστο εξαιτίας του push μοντέλου. Όμως έχουν μεγάλη δυσκολία υλοποίησης, δεν είναι συμβατές με παλαιότερους web browsers και πιθανώς να χρειαστεί και εγκατάσταση επιπλέον λογισμικού στον web browser.

Το επόμενο μικρότερο κόστος σε μεταφερόμενα δεδομένα το έχει η μέθοδος XML HTTP. Η μέθοδος αυτή είναι σχετικά νέα και υποστηρίζεται από όλους τους τελευταίους web browsers (IE5+, Mozilla 1.0+). Το μικρό της κόστος οφείλεται στο ότι επιτρέπει στον web browser να ανακτά ασύγχρονα δεδομένα από τον web server και να τα προβάλει χωρίς να χρειάζεται η επαναφόρτωση ολόκληρης της σελίδας. Είναι δηλαδή ιδανική για την προβολή σελίδων που αλλάζουν λίγο κάθε φορά.

Τέλος το ίδιο κόστος σε μεταφερόμενα δεδομένα έχουν η μη χρήση δυναμικών στοιχείων και η χρήση δυναμικών στοιχείων JavaScript. Ουσιαστικά και οι δύο

φορτώνουν τη σελίδα προόδου απ' την αρχή. Επίσης και οι δύο μέθοδοι μπορούν να δουλέψουν σε παλαιότερους browsers, αν και η χρήση των δυναμικών στοιχείων JavaScript είναι μάλλον προβληματική εξαιτίας ασύμβατων υλοποιήσεων τους στους παλαιότερους web browsers.

Τα παραπάνω μας οδηγούν στη χρήση της μεθόδου XML HTTP για την παρουσίαση της προόδου ελέγχου, ενώ παράλληλα θα υπάρχει και η δυνατότητα μη-χρήσης δυναμικών στοιχείων σαν ασφαλές fallback.

Όπως φαίνεται στον Πίνακας 1, οι μέθοδοι αυτοί δεν υποστηρίζουν push πληροφορίας από τον εξυπηρετητή πύλης προς τον web browser. Συνεπώς ο web browser θα πρέπει να ζητά περιοδικά από τον εξυπηρετητή πύλης την πληροφορία για την πρόοδο του ελέγχου. Η περιοδική ανάκτηση της πληροφορίας αυτής μπορεί να υλοποιηθεί είτε με JavaScript αν χρησιμοποιούμε δυναμικά στοιχεία, είτε με HTML meta-tags αν δεν χρησιμοποιούμε.

2.3. Portal Server

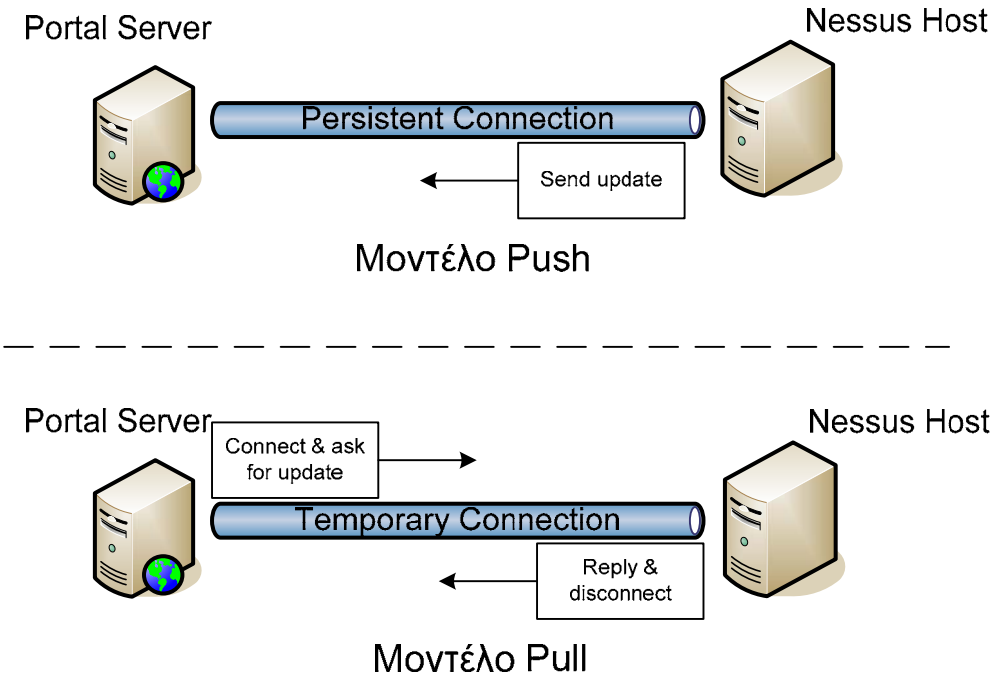
Ο Portal Server παρέχει στο browser που χρησιμοποιεί ο χρήστης μεθόδους για να ξεκινήσει ένα έλεγχο καθώς και πληροφορία για την πρόοδο των ελέγχων που έχουν ξεκινήσει.

2.3.1. Επικοινωνία με τον Nessus Host

Καθώς επιθυμούμε ο χρήστης να μπορεί να παρακολουθεί «ζωντανά» την εξέλιξη του ελέγχου, η πιο συχνή δραστηριότητα του εξυπηρετητή πύλης θα είναι να επικοινωνεί με τον Nessus Host όποτε χρειάζεται να δώσει στον web browser του χρήστη νέα πληροφορία για αυτήν. Όπως αναφέρθηκε στο 2.2.2 παραπάνω, η ανάκτηση της πληροφορίας αυτής γίνεται με πρωτοβουλία του web-browser. Θα πρέπει τώρα να καθορίσουμε με ποιο τρόπο την αποκτά ο Portal Server για να προωθήσει την πληροφορία αυτή από το Nessus Host. Εφόσον στη πλευρά του εξυπηρετητή πύλης έχουμε τη πληροφορία για τους ελέγχους που είναι ενεργοί κάθε στιγμή αυτό που θα γίνεται είναι να ανανεώνεται αυτόματα η κατάσταση του κάθε ελέγχου στον Portal Server και η επικοινωνία με τον χρήστη δεν χρειάζεται να αλλάξει. Οπότε ο χρήστης

απλώς θα παρακολουθεί τη κατάσταση του ελέγχου από το site που θα ανανεώνεται αυτόματα.

Αρχικά θα πρέπει να αποφασίσουμε αν θα χρησιμοποιήσουμε pull ή push μοντέλο. Αν δηλαδή ο Portal Server θα ζητάει την πληροφορία όποτε τη χρειάζεται ή αν ο Nessus Host θα προωθεί τη νέα πληροφορία όποτε αυτή υπάρχει. Τα δύο μοντέλα αυτά για την ανάκτηση της πληροφορίας από τον Portal Server, εμφανίζονται στο Σχήμα 2.



Σχήμα 2: Push και Pull στην επικοινωνία Portal Server και Nessus Host

Στο μοντέλο push, καθώς μπορεί να υπάρχει μεγάλος αριθμός από ελέγχους που δεν παρακολουθούνται «ζωντανά», θα πρέπει ο Portal Server να παρακολουθεί ποιοι είναι οι έλεγχοι αυτοί και να ενημερώνει το Nessus Host. Αλλιώτικα θα έχουμε μεγάλο αριθμό από περιττές επικοινωνίες. Η καταγραφή των ελέγχων που παρακολουθούνται «ζωντανά» προσθέτει επιπλέον πολυπλοκότητα στην υλοποίηση του Portal Server.

Στο μοντέλο pull από την άλλη, ο Portal Server θα ζητάει ο ίδιος τα δεδομένα που του χρειάζονται. Τα δεδομένα αυτά του χρειάζονται μόνο όποτε του ζητούνται από τον

web browser του χρήστη, συνεπώς δεν χρειάζεται ο Portal Server να καταγράφει ποιοι είναι οι έλεγχοι που παρακολουθούνται «ζωντανά». Η υλοποίηση της καταγραφής στο μοντέλο pull μπορεί να προστεθεί αργότερα σαν βελτιστοποίηση. Η προσθήκη της θα έχει ως αποτέλεσμα να μειωθεί ο αριθμός των επικοινωνιών στο push μοντέλο, καθώς ο Nessus Host θα μπορεί να στέλνει παραπάνω δεδομένα από όσα του ζητήθηκαν (piggybacking).

Το pull μοντέλο λοιπόν συνδυάζει μικρότερο αρχικό κόστος υλοποίησης σε σχέση με το push μοντέλο, με τη δυνατότητα να βελτιστοποιηθεί η λειτουργία του αν παραστεί η ανάγκη. Αυτό μας οδηγεί στο να το προτιμήσουμε για την επικοινωνία Porta Server με Nessus Host.

2.4. Auth Server

Ο Auth Server παίρνει την απόφαση για το αν έχει ή όχι ο χρήστης δικαίωμα να κάνει τον έλεγχο που ζήτησε. Οι διαστάσεις που καθορίζουν τον τρόπο που θα ληφθεί αυτή η απόφαση παρουσιάζονται στον επόμενο πίνακα.

	Πλήθος Ελεγχόμενων Υπολογιστών	
Πρόσβαση στο Διαδίκτυο	Πλήρης / Ένας	Πλήρης / Πολλοί
	Περιορισμένη / Ένας	Περιορισμένη / Πολλοί

Πίνακας 2

2.4.1. Πλήρης Πρόσβαση / Ένας Ελεγχόμενος Υπολογιστής

Στην περίπτωση αυτή η απόφαση του Auth Server μπορεί να ληφθεί εξετάζοντας τη διεύθυνση IP από την οποία ο Portal Server παρέλαβε την αίτηση. Αν η διεύθυνση είναι η ίδια για την οποία ζητείται και έλεγχος, τότε η αίτηση εγκρίνεται. Ειδάλλως απορρίπτεται. Αν ο Ελεγχόμενος Υπολογιστής είναι διαφορετικός από τον Υπολογιστή του Χρήστη, τότε πρέπει να ακολουθηθεί η διαδικασία που περιγράφεται στο 2.4.2 παρακάτω.

Προσοχή χρειάζεται η περίπτωση που υπάρχει μεν πλήρης πρόσβαση στο διαδίκτυο, αλλά μεταξύ του χρήστη και του Portal Server παρεμβάλλεται κάποιος HTTP Proxy Server. Ο HTTP Proxy μπορεί να είναι ορατός ή αόρατος (transparent). Η πρώτη

περίπτωση μπορεί να γίνει αντιληπτή από τους HTTP headers που προσθέτει ο Proxy και να ζητηθεί από το χρήστη να τον απενεργοποιήσει προσωρινά. Για να παρακάμψουμε τυχόν αόρατο Proxy, αρκεί κατά την τελική υποβολή της αίτησης ελέγχου από το χρήστη με τη χρήση του web browser του, να χρησιμοποιήσουμε το πρωτόκολλο HTTPS αντί του απλού HTTP.

2.4.2. Πλήρης Πρόσβαση / Πολλοί Ελεγχόμενοι Υπολογιστές

Σε αυτήν την περίπτωση, θα πρέπει να ο Auth Server να επιβεβαιώσει πως ο χρήστης που ζήτησε τον έλεγχο για τους συγκεκριμένους υπολογιστές, είναι πράγματι υπεύθυνος για τη συντήρηση και την ασφάλεια τους. Μια και δεν είναι δυνατό να πάρει ο Auth Server αυτή την πληροφορία μόνο από τα περιεχόμενα της αίτησης ελέγχου, θα πρέπει να ακολουθηθεί άλλη διαδικασία.

Κατά την υποβολή της αίτησης, ο χρήστης θα επιλέγει ένα από τα ονόματα χρηστών (usernames) που αντιστοιχούν στους διαχειριστές του δικτύου ενός οργανισμού. Τέτοια για παράδειγμα είναι τα root, webmaster, postmaster κλπ. Ο Auth Server θα προσδιορίζει έπειτα σε ποιο οργανισμό ανήκουν οι διευθύνσεις που ζητήθηκε να ελεγχθούν και θα στέλνει ένα email στον αντίστοιχο χρήστη του οργανισμού αυτού. Το email θα περιέχει ένα μυστικό κωδικό επιβεβαίωσης που θα είναι απαραίτητος για να προχωρήσει ο έλεγχος. Μέχρι ο χρήστης να εισάγει το μυστικό κωδικό (χρησιμοποιώντας τον web browser του) ο έλεγχος δε θα ξεκινήσει. Αν ο χρήστης καθυστερήσει υπερβολικά (π.χ. 2 ώρες), να εισάγει το μυστικό κωδικό, ο έλεγχος ακυρώνεται εντελώς.

2.4.3. Περιορισμένη Πρόσβαση/ Ένας Ελεγχόμενος Υπολογιστής

Στην περίπτωση αυτή, θα πρέπει να δημιουργηθεί ένα κανάλι επικοινωνίας μεταξύ του ελεγχόμενου υπολογιστή και του Nessus Host διαμέσου του VPN Router. Δηλαδή θα δημιουργείται ένα VPN μεταξύ του ελεγχόμενου υπολογιστή και του VPN Router, και ο VPN Router θα δρομολογεί την κίνηση από και προς τον ελεγχόμενο υπολογιστή μέσα από το VPN αυτό. Για τη διαδικασία δημιουργίας του VPN, μπορούμε να διαχωρίσουμε δύο περιπτώσεις, ανάλογα με τον τύπο των περιορισμών πρόσβασης στο διαδίκτυο.

(a) Περιορισμένη πρόσβαση στο διαδίκτυο λόγω ύπαρξης NAT

Για τη δημιουργία του VPN αυτού χρειάζονται επιπλέον ενέργειες από την πλευρά του χρήστη. Αν ο χρήστης βρίσκεται πίσω από κάποιο NAT Router τότε θα λαμβάνει οδηγίες για να εγκαταστήσει ένα πρόγραμμα που θα δημιουργεί το VPN. Για λόγους αξιοπιστίας, το πρόγραμμα αυτό θα πρέπει κατά προτίμηση να προέρχεται από πηγή που δε σχετίζεται με το σύστημα ΕΣΤΙΑ. Επίσης ο χρήστης θα λαμβάνει και ένα αρχείο με τις ρυθμίσεις για το VPN που θα δημιουργηθεί. Το αρχείο θα πρέπει να είναι σε μορφή κατανοήσιμη απευθείας από το πρόγραμμα που δημιουργεί το VPN. Έτσι ο χρήστης δε θα χρειαστεί να γνωρίζει τεχνικές λεπτομέρειες σχετικά με τα VPN και τη χρήση τους για να το χρησιμοποιήσει.

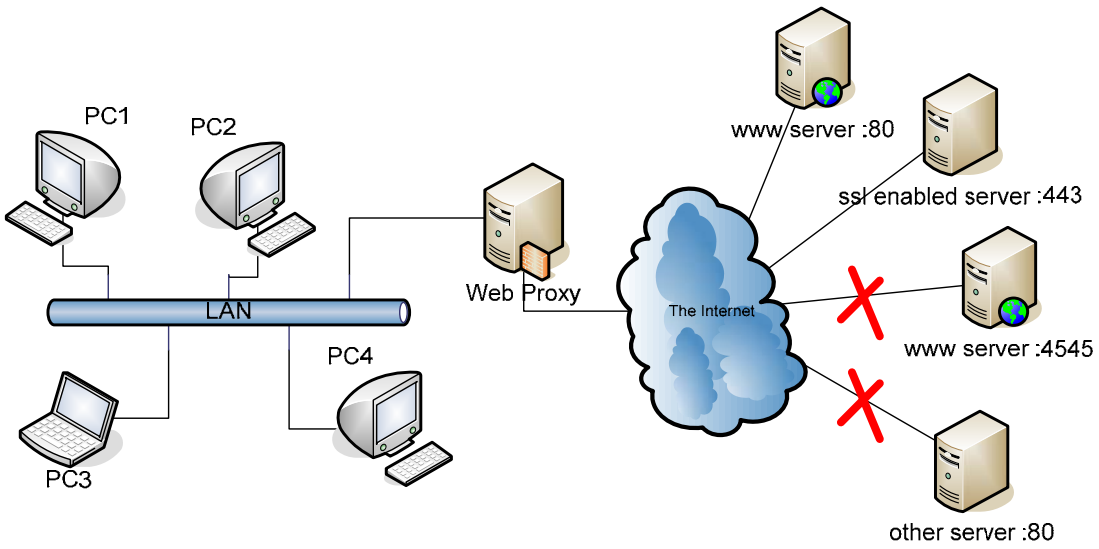
(b) Πρόσβαση μόνο στην υπηρεσία Παγκόσμιου Ιστού (WWW) του διαδικτύου

Αν ο χρήστης έχει πρόσβαση μόνο στον Παγκόσμιο Ιστό, τότε πριν δημιουργήσει το VPN θα λαμβάνει επιπλέον οδηγίες για να δημιουργήσει μια σύνδεση SSH[6] μέχρι τον VPN Router. Μέσα από τη σύνδεση αυτή θα περνάει η κίνηση προς τον VPN Server που τρέχει στον VPN Router. Και τελικά μέσα από το VPN που θα δημιουργηθεί θα περνάει η κίνηση από τον Nessus Host προς τον ελεγχόμενο υπολογιστή. Αυτή η στρωμάτωση των πρωτοκόλλων επικοινωνίας παρουσιάζεται στον επόμενο πίνακα.

Τύπος Κίνησης	Άκρο 1	Άκρο 2	Περιγραφή
TCP/IP traffic	Ελεγχόμενος Υπολογιστής	Nessus Host	Οποιουδήποτε είδους TCP/IP κίνηση. Η ύπαρξη του VPN δίνει στης εφαρμογές την ψευδαίσθηση πως ο Ελεγχόμενος Υπολογιστής είναι πλήρως προσβάσιμος από τον Nessus Host.
VPN traffic	Ελεγχόμενος Υπολογιστής	VPN Router	Κίνηση του VPN. Περιέχει encapsulated IP πακέτα που περνάνε πάνω από το ιδεατό δίκτυο.
SSH traffic	Ελεγχόμενος Υπολογιστής	VPN Router	Κίνηση μεταξύ του SSHd που τρέχει στον VPN server και ενός SSH client στον

Τύπος Κίνησης	Άκρο 1	Άκρο 2	Περιγραφή
			ελεγχόμενο υπολογιστή.
HTTPS traffic	Ελεγχόμενος Υπολογιστής	Web Proxy	Κίνηση που ο Web Proxy αντιλαμβάνεται ως HTTPS, περιέχει όμως τελικά την κίνηση SSH.
TCP/IP traffic	Ελεγχόμενος Υπολογιστής	Web Proxy	Η κίνηση μεταξύ του ελεγχόμενου υπολογιστή και του Web Proxy.

Πίνακας 3



Σχήμα 3: Υπολογιστές με πρόσβαση μόνο στον Παγκόσμιο Ιστό

Για να γίνει πιο κατανοητός ο τρόπος δημιουργίας του καναλιού επικοινωνίας μεταξύ Ελεγχόμενου Υπολογιστή και Nessus Host, στο Σχήμα 3 παρουσιάζεται μια τοπολογία με υπολογιστές που έχουν πρόσβαση μόνο στον Παγκόσμιο Ιστό. Οι υπολογιστές PC1, PC2, PC3, PC4, συνδέονται μεταξύ τους δια μέσου ενός τοπικού δικτύου. Η μόνη πρόσβαση που έχουν στο Internet, είναι στον Παγκόσμιο Ιστό χρησιμοποιώντας τον Web Proxy που είναι συνδεμένος στο ίδιο τοπικό δίκτυο.

Ο Web Proxy θα μπορούσε να χρησιμοποιηθεί σαν γενικός proxy χρησιμοποιώντας την εντολή CONNECT του πρωτοκόλλου HTTP. Όμως ένας σωστά στημένος Web Proxy δεν επιτρέπει την χρήση της CONNECT παρά μόνο για πρόσβαση σε web servers που περιμένουν για αιτήσεις στην πόρτα 80 καθώς και secure web servers που

περιμένουν για αιτήσεις στην πόρτα 443. Οπότε δεν θα επιτρέψουν σύνδεση ούτε με μια υπηρεσία που τρέχει στην πόρτα 80 και δεν είναι web server, αλλά ούτε και με έναν web server που δεν τρέχει στην πόρτα 80. Ένας τέτοιος σωστά ρυθμισμένος Web Proxy απεικονίζεται και στο Σχήμα 3.

Για να προσπεράσουμε τους περιορισμούς που θέτει ο Web Proxy, θα εκμεταλλευτούμε την αδυναμία του να «καταλάβει» τον τύπο της κίνησης μεταξύ ενός υπολογιστή στο LAN και ενός Secure Web Sever. Η αδυναμία αυτή οφείλεται στο πως η κίνηση αυτή είναι κρυπτογραφημένη. Η κρυπτογράφηση γίνεται με τη χρήση του πρωτοκόλλου SSL[7]. Ο Web Proxy μπορεί να καταλάβει μόνο αν έγινε η αρχική χειραψία του SSL προτού αρχίσει η ροή κρυπτογραφημένων δεδομένων.

Εκτός από τους Secure Web Servers, και οι SSH Servers πραγματοποιούν τη χειραψία SSL αρχικά προτού ξεκινήσει η ροή κρυπτογραφημένων δεδομένων. Οπότε ο Web Proxy δε μπορεί να διαχωρίσει έναν SSH Server που τρέχει στην πόρτα 443 από έναν Secure Web Server. Επιπλέον το πρωτόκολλο SSH προβλέπει τη δημιουργία SSH Tunnels που επιτρέπουν να χρησιμοποιήσουμε μια σύνδεση SSH για να μεταφέρουμε οποιουδήποτε είδους κίνησης μεταξύ των δύο άκρων της. Με τη χρήση λοιπόν μιας σύνδεσης SSH και ενός SSH Tunnel μπορούμε να μεταφέρουμε την κίνηση του VPN προς τον VPN Router ξεπερνώντας τους περιορισμούς του υπολογιστή να έχει πρόσβαση στο διαδίκτυο.

2.4.4. Περιορισμένη Πρόσβαση / Πολλοί Ελεγχόμενοι Υπολογιστές

Στην περίπτωση αυτή, θα πρέπει να δημιουργηθεί ένα κανάλι επικοινωνίας μεταξύ του Υπολογιστή Χρήστη και του Nessus Host διαμέσου του VPN Router. Θα πρέπει όμως επιπλέον ο Υπολογιστής Χρήστη να ρυθμιστεί ώστε να δρομολογεί τα δεδομένα που ανταλλάσσουν Nessus Host και Ελεγχόμενοι Υπολογιστές προς τη σωστή κατεύθυνση.

2.5.Nessus Host

Ο Nessus Host οπώς αναφέρθηκε και προηγουμένως είναι το υποσύστημα στο οποίο τρέχει το λογισμικό Nessus που ελέγχει τα μηχανήματα για τα οποία θέλουμε να διαπιστώσουμε αν υπάρχουν ρήγματα ασφαλείας. Το λογισμικό Nessus είναι

σχεδιασμένο με την αρχιτεκτονική client-server. Ο Nessus Server εκτελεί τον έλεγχο ασφαλείας με βάση κάποιες παραμέτρους που δέχεται από το Nessus Client. Οι παράμετροι αυτοί είναι για παράδειγμα τα μηχανήματα που θα πρέπει να εξεταστούν, ποίες πόρτες και με ποίον τρόπο θα πρέπει να εξεταστούν καθώς και για τι είδους ρήγματα ασφαλείας θα πρέπει να ελέγξει ο Nessus Server. Όλες αυτές οι παράμετροι καθορίζονται σε κάποιο προφίλ ελέγχου το οποίο δίνεται από το Nessus Client στο Nessus Server.

2.5.1. Λειτουργία του λογισμικού Nessus

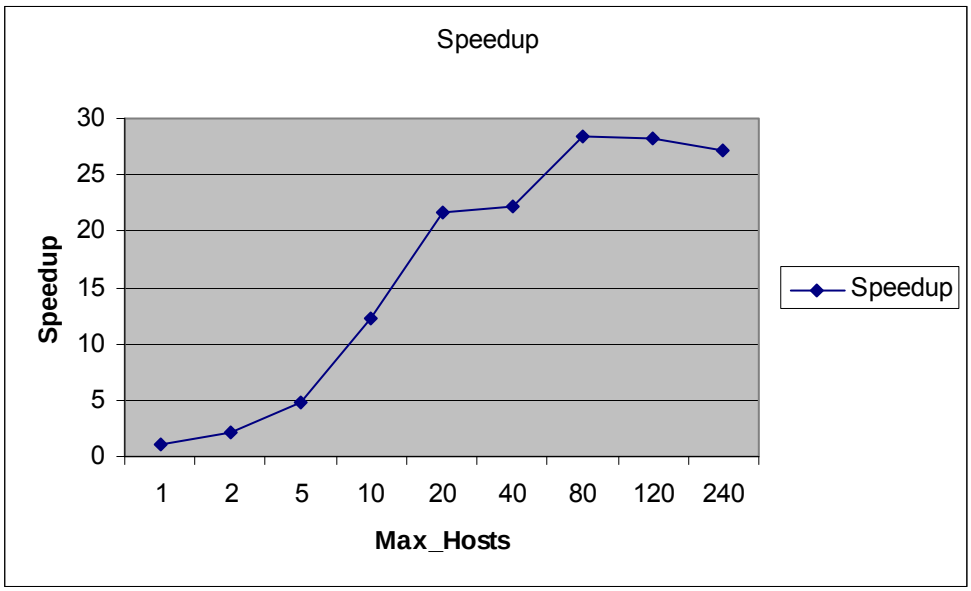
Ο Nessus Server μπορεί να δουλέψει σε δύο διαφορετικά modes: batch mode και standard mode. Σε batch mode ο Nessus Client δίνει όλα τα ορίσματα που χρειάζεται ο Nessus Server από τη γραμμή εντολών χρησιμοποιώντας ως προφίλ ελέγχου κάποιο προκατασκευασμένο αρχείο καθώς επίσης και το αρχείο που περιέχει τις IP των μηχανημάτων τα οποία θέλουμε να εξετάσουμε. Στο τέλος του ελέγχου ο Nessus Client αποθηκεύει τα αποτελέσματα σε κάποιο αρχείο που έχουμε ορίσει και στη μορφή που έχουμε ζητήσει π.χ. HTML, XML. Όταν ο Nessus Client τρέχει σε standard mode θα πρέπει να καθορίσει ο χρήστης όλες τις παραμέτρους του ελέγχου δια μέσου μιας διεπαφής χρήστη. Αυτές οι παράμετροι που θα καθορίσει ο χρήστης μπορούν να αποθηκευτούν σε κάποιο αρχείο και να αποτελέσουν μελλοντικό προφίλ ελέγχου. Τα αποτελέσματα παρουσιάζονται με γραφικό τρόπο στο τέλος του έλεγχου και ο χρήστης μπορεί να τα αποθηκεύσει σε κάποιο αρχείο στη μορφή που επιθυμεί.

Στο ΕΣΤΙΑ θα χρησιμοποιήσουμε το Nessus Client σε batch mode για να μπορούμε να χρησιμοποιούμε προκαθορισμένα προφίλ ελέγχου και να παίρνουμε τα αποτελέσματα αυτόματα σε κάποιο αρχείο στη μορφή που θέλουμε. Έτσι θα μπορούμε να επεξεργαστούμε τα αποτελέσματα, να τα αποθηκεύσουμε στον Report Server και να τα παρουσιάσουμε στο χρήστη.

2.5.2. Επιδόσεις του λογισμικού Nessus

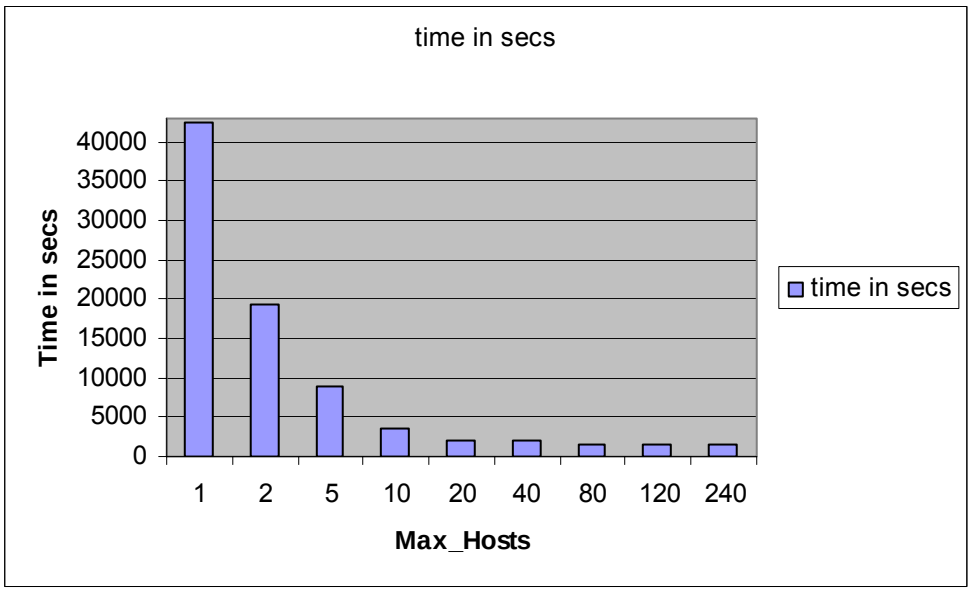
Ο Nessus Server έχει τη δυνατότητα να πραγματοποιεί ταυτόχρονα πολλούς ελέγχους σε διαφορετικά μηχανήματα. Για να εξετάσουμε την επίδραση της χρήση αυτής της δυνατότητας στην ταχύτητα με την οποία γίνονται οι έλεγχοι, αλλά και για να βρούμε

ποια είναι η κατάλληλη τιμή για το βαθμό της παραλληλίας που θέλουμε να έχουμε στο σύστημα μας, κάναμε το ακόλουθο πείραμα.



Σχήμα 4: Speedup Ανάλογα με το βαθμό παραλληλίας

Ελέγξαμε ένα αριθμό (210) μηχανημάτων από δύο διαφορετικά υποδίκτυα δοκιμάζοντας διαφορετικές τιμές για το μέγεθος της παραλληλίας που θέλουμε να μας προσφέρει ο Nessus Server και μετρήσαμε το χρόνο που χρειάστηκε για να ολοκληρωθεί ο έλεγχος όλων των μηχανημάτων. Τα αποτελέσματα φαίνονται στο Σχήμα 4 και στο Σχήμα 5.



Σχήμα 5: Χρόνος ολοκλήρωσης ανάλογα με το βαθμό παραλληλίας

Στο Σχήμα 5 βλέπουμε τον χρόνο που χρειάστηκε για τον έλεγχο όλων (210) των μηχανημάτων για διαφορετικές τιμές παραλληλίας. Βλέπουμε ότι το καλύτερο χρόνο τον επιτυγχάνουμε όταν το «Max_Hosts=80» το οποίο σημαίνει ότι ο Nessus Server ελέγχει 80 μηχανήματα ταυτόχρονα. Μάλιστα γι' αυτή τη τιμή ο Nessus Server ολοκληρώνει τον έλεγχο 28.4 φορές γρηγορότερα απ' ότι με τον σειριακό τρόπο. Η βελτίωση σε σχέση με το σειριακό τρόπο φαίνεται καθαρά στο Σχήμα 3 όπου υπολογίζουμε το "Speedup" δηλαδή τη βελτίωση που επιτυγχάνουμε σε σχέση το σειριακό. Όπως φαίνεται στο σχήμα αυξάνοντας το βαθμό της παραλληλίας μειώνεται ο χρόνος που χρειάζεται για την ολοκλήρωση του ελέγχου σχεδόν αντίστροφος ανάλογα με την αύξηση της παραλληλίας. Μετά το «Max_Hosts=80» βλέπουμε ότι δεν μειώνεται ο χρόνος αλλά αντιθέτως αυξάνεται αυτό οφείλεται στο ότι για να επιτύχει ο Nessus Server τη ζητούμενη παραλληλία δημιουργεί πολλές διαφορετικές διεργασίες που αναλαμβάνουν να κάνουν ένα υπό-τμήμα του ελέγχου. Όταν όμως ο αριθμός των διεργασιών ξεπεράσει ένα όριο τότε έχουμε πολλές εναλλαγές μεταξύ τους και έτσι αργεί η κάθε διεργασία να τρέξει και να διαβάσει τα δεδομένα τα οποία χρειάζεται καθυστερώντας το όλο σύστημα.

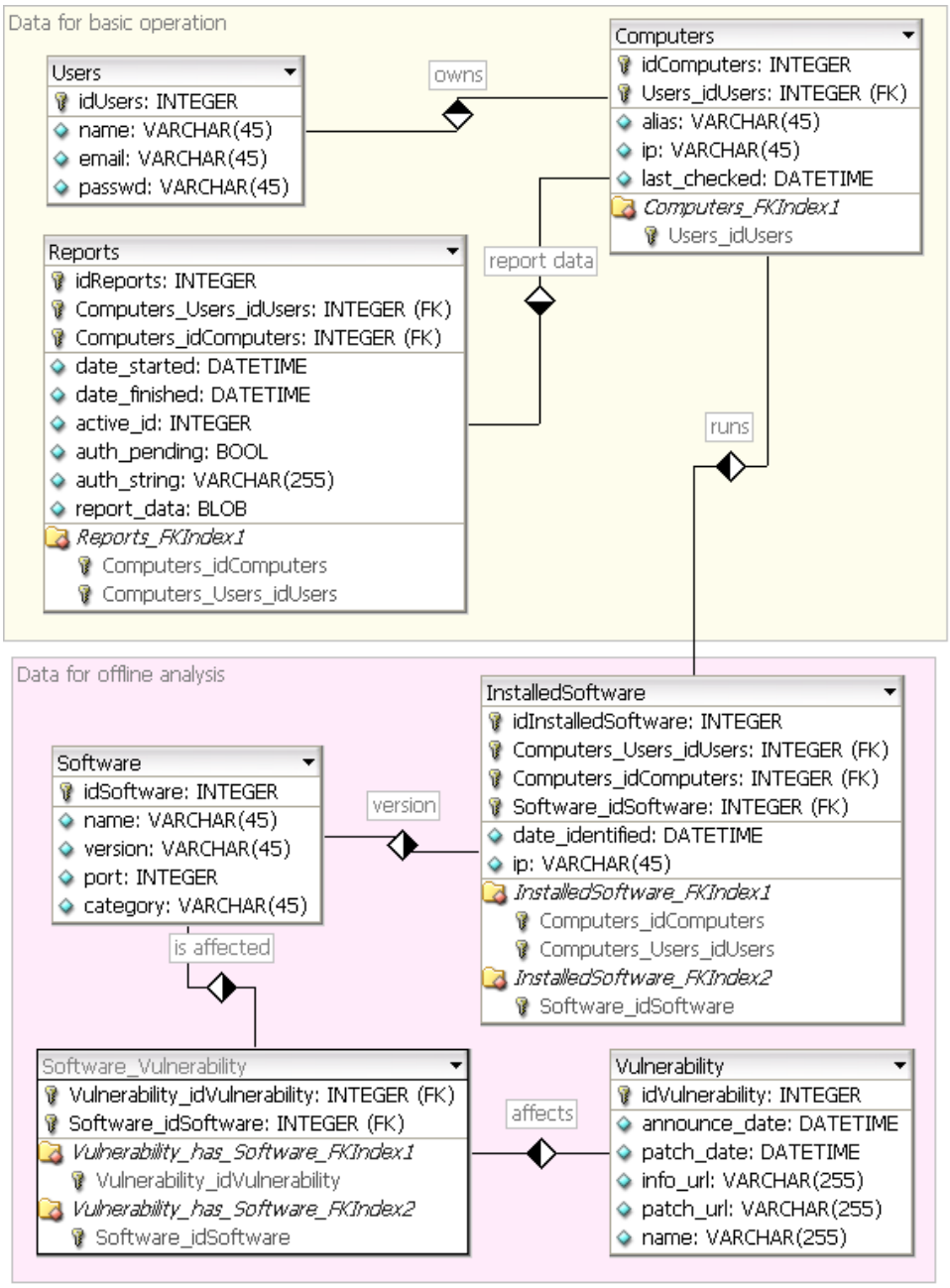
Για να επιτρέψουμε στο σύστημα να ελέγχει μεγάλο αριθμό από υπολογιστές ταυτόχρονα, μπορούμε να το επεκτείνουμε βάζοντας πολλαπλούς Nessus Hosts οι οποίο θα έχουν τη δυνατότητα να ελέγχουν πολλαπλά μηχανήματα ταυτόχρονα ο καθένας. Επιπλέον θα πρέπει να υπάρχει και κάποιος κατανεμητής φόρτου (load balancer) που θα επιλέγει ποίο Nessus Host θα είναι υπεύθυνο να τον επόμενο έλεγχο ασφάλειας που ζητάει κάποιος χρήστης. Την απόφαση θα τη παίρνει βασιζόμενος στο φόρτο των Nessus Hosts και στα διάφορα χαρακτηρίστηκα τους, π.χ. την γεωγραφική τους τοποθεσία. Ο κατανεμητής φόρτου θα μπορούσε εύκολα να ενσωματωθεί στον Portal Server. Προς το παρών ένα Nessus Host μπορεί να ελέγξει ένα αρκετά μεγάλο αριθμό μηχανημάτων σε λιγότερο από 30 λεπτά.

2.6. Report Server

Όπως αναφέρθηκε και στην παράγραφο 1.6, ο Report Server χρησιμοποιείται για την αποθήκευση των αποτελεσμάτων του ελέγχου ασφαλείας ώστε να μπορούν να ανακτηθούν αργότερα καθώς και στοιχείων και κωδικών των χρηστών του συστήματος. Τα στοιχεία αυτά θα αποθηκεύονται σε μια σχεσιακή βάση δεδομένων. Αν και στο Σχήμα 1 ο Report Server παρουσιάζεται ξεχωριστά από τον Portal Server, τελικά οι δύο Servers μπορεί να τρέχουν στο ίδιο μηχάνημα. Αυτό γιατί αν τρέχουν σε ξεχωριστά μηχανήματα, η επικοινωνία τους θα πρέπει να γίνεται πάνω από το δίκτυο. Η επικοινωνία πάνω από δίκτυο θα είναι πολύ πιο αργή από την επικοινωνία με τη χρήση μηχανισμών IPC που θα συνέβαινε αν και οι δύο servers έτρεχαν στο ίδιο μηχάνημα.

Για την επικοινωνία του Report Server με τον Portal Server δεν χρειάζεται περαιτέρω μέριμνα. Κάθε λύση για Portal Server υποστηρίζει από ένα αριθμό σχεσιακών βάσεων δεδομένων. Οπότε αρκεί κατά την υλοποίηση του συστήματος να φροντίσουμε να υλοποιήσουμε τον Report Server πάνω σε μια σχεσιακή βάση δεδομένων που υποστηρίζεται από την τεχνολογία Portal Server που επιλέξαμε.

Παρουσιάζουμε στο Σχήμα 6 παρακάτω ένα ενδεικτικό σχήμα για τη βάση δεδομένων του Report Server. Για τον κάθε χρήστη αποθηκεύονται τα ελάχιστα δυνατά στοιχεία στον πίνακα Users. Ο χρήστης μπορεί να δημιουργήσει ομάδες υπολογιστών που αποθηκεύονται στον πίνακα Computers. Για κάθε ομάδα αποθηκεύονται οι διευθύνσεις που περιέχονται σε αυτήν καθώς και πότε ελέγχθηκε αυτή για τελευταία φορά. Για κάθε έλεγχο ασφαλείας σε μια ομάδα, δημιουργείται μια εγγραφή στον πίνακα Reports. Ο πίνακας αυτός περιέχει την ώρα που ο έλεγχος ξεκίνησε και τελείωσε. Αν ο έλεγχος είναι ακόμα ενεργός, το πεδίο active_id χρησιμεύει για να ληφθεί πληροφορία για την κατάσταση του απευθείας από τον Nessus Host. Επίσης αν ο έλεγχος αφορά περισσότερες από μια IP διευθύνσεις, το πεδίο auth_string περιέχει τη συμβολοσειρά που θα πρέπει να εισάγει ο χρήστης για να επιβεβαιώσει πως δικαιούται να κάνει τον έλεγχο. Φυσικά όταν τελειώσει ο έλεγχος, τα αποτελέσματα αποθηκεύονται στο πεδίο report_data μαζί με συμβουλές για το πως θα επιλύσει πιθανά προβλήματα.



Σχήμα 6: Ενδεικτικό σχήμα βάσης για χρήση στο σύστημα ΕΣΤΙΑ

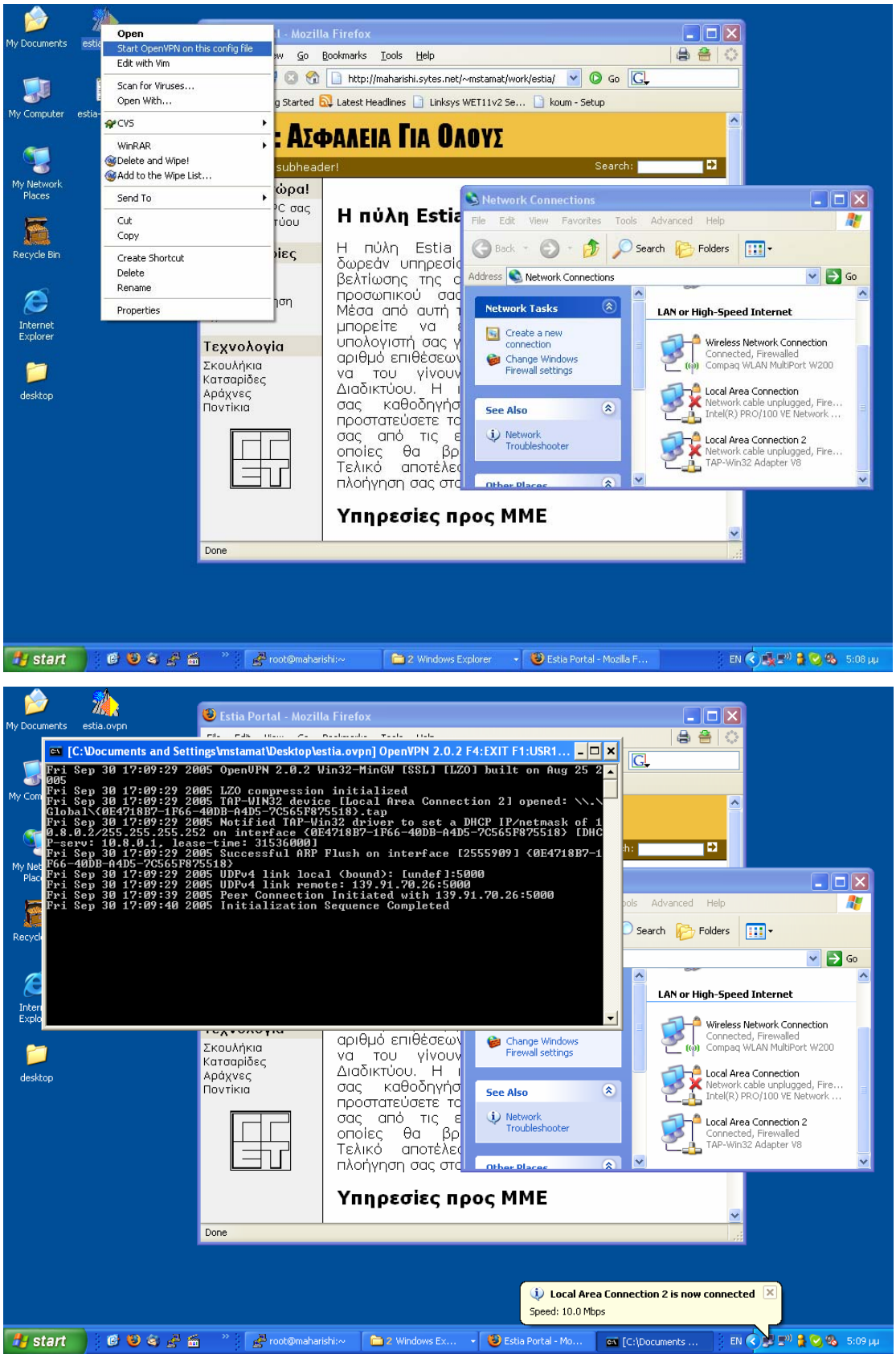
Πέρα από αυτά τα δεδομένα που επιτρέπουν τη βασική λειτουργία του συστήματος, η βάση μπορεί να αποθηκεύει και επιπλέον δεδομένα που θα επιτρέψουν στο χρήστη να ενημερωθεί έγκαιρα για νέα προβλήματα ασφαλείας που ανακαλύπτονται και αφορούν τον υπολογιστή του. Για κάθε ομάδα υπολογιστών καταγράφονται σε κάθε έλεγχο μια σειρά από εγγραφές για το λογισμικό που ανιχνεύθηκε, πότε ανιχνεύθηκε και ποιος ήταν ο υπολογιστής ο οποίος το έτρεχε. Έτσι από τη μια ο χρήστης μπορεί

να πληροφορηθεί για νέα προβλήματα ασφαλείας, όμως έχει επιπλέον και τη δυνατότητα να δει ένα ιστορικό ασφαλείας του υπολογιστή.

2.7. VPN Router

Ο VPN Router όπως αναφέραμε και σε προηγούμενη παράγραφο, αναλαμβάνει τη δρομολόγηση των δεδομένων μεταξύ του Nessus Host και του Ελεγχόμενου Υπολογιστή αν ο δεύτερος βρίσκεται σε περιορισμένο περιβάλλον. Για να μπορέσει να λειτουργήσει ο VPN Router αρκεί να επιτρέπεται η στον Ελεγχόμενο Υπολογιστή η επικοινωνία με το Internet πάνω από μια και μόνο πόρτα, όπως περιγράφηκε στην παράγραφο 2.4.3 παραπάνω.

Ο VPN Router ενημερώνεται από τον Portal Server για τα στοιχεία του Ελεγχόμενου Υπολογιστή, μόλις ο Portal Server βεβαιωθεί πως ο χρήστης έχει το δικαίωμα να κάνει τον έλεγχο που ζήτησε. Ο VPN Router τότε ενημερώνει τις ρυθμίσεις του για να μπορέσει να συνδεθεί ο Ελεγχόμενος Υπολογιστής σε αυτόν. Επίσης παράγει ένα αρχείο ρυθμίσεων το οποίο κατεβάζει ο χρήστης διαμέσου του Portal Server. Με τη χρήση του αρχείου αυτού και δεδομένου πως έχει ήδη εγκαταστήσει το λογισμικό για τη δημιουργία του VPN, ο χρήστης συνδέεται στον VPN Router και το VPN δημιουργείται. Στο Σχήμα 7 φαίνεται η διαδικασία δημιουργίας ενός VPN. Ο χρήστης αφού κατεβάσει το αρχείο ρυθμίσεων που θα δημιουργηθεί από τον VPN Router, αρκεί να κάνει δεξί κλικ σε αυτό και να επιλέξει να ξεκινήσει η εφαρμογή που υλοποιεί το VPN (πρώτη εικόνα). Αμέσως μετά, η εφαρμογή ξεκινά και ο χρήστης ενημερώνεται πως έχει συνδεθεί με το σύστημα (δεύτερη εικόνα). Με το τέλος του ελέγχου ο Portal Server, αφού αποθηκεύσει τα αποτελέσματα του ελέγχου, θα ενημερώσει τον VPN Router ώστε να κλείσει το VPN.



Σχήμα 7: Δημιουργία σύνδεσης VPN μεταξύ Υπολογιστή Χρήστη και VPN Router

3. Ορισμός διεπαφών υποσυστημάτων

Στην αρχιτεκτονική που έχουμε περιγράψει παραπάνω υπάρχουν τέσσερα διαφορετικά υποσυστήματα τα οποία επικοινωνούν μεταξύ τους. Ο τρόπος που επικοινωνούν μεταξύ τους θα περιγραφεί σε αυτή την ενότητα.

3.1.1. Διεπαφή Portal Server – Nessus Host

Ο Portal Server επικοινωνεί με το Nessus Host για να του δώσει τα απαραίτητα δεδομένα για να ξεκινήσει τον έλεγχο. Καθώς δεν θα επιθυμούσαμε την τροποποίηση του λογισμικού Nessus για να γίνεται αυτή η επικοινωνία, την επικοινωνία θα την αναλαμβάνει ένας proxy. Αυτό που κάνει ο proxy είναι να δέχεται HTTP αιτήσεις από τον Portal Server και ανάλογα είτε να αρχίζει ένα νέο έλεγχο είτε να απαντάει με την προόδου ενός ήδη υπάρχοντος ελέγχου. Η προσέγγιση αυτή έχει το πλεονέκτημα πως από τη μια η αποσφαλμάτωση του συστήματος θα διευκολυνθεί, ενώ ταυτόχρονα οι νέες εκδόσεις του λογισμικού Nessus θα μπορούν να ενσωματωθούν στο σύστημα με ελάχιστη προσπάθεια.

Για να ξεκινήσει ένα νέο έλεγχο ο Portal Server στέλνει μια αίτηση HTTP της μορφής «GET /start?target=<IP-range>&profile=<profile-description>». Ο proxy ξεκινάει τότε έναν Nessus Client με τα σωστά ορίσματα για να πραγματοποιήσει τον έλεγχο. Σαν αποτέλεσμα της αίτησης, ο proxy επιστρέφει στον Portal Server ένα αναγνωριστικό (ID) για να μπορέσει αργότερα ο Portal Server να ζητήσει την κατάσταση ενός ελέγχου καθώς και τα τελικά αποτελέσματα του.

Η ανάκτηση της κατάστασης ενός ελέγχου καθώς και των αποτελεσμάτων του, γίνονται στέλνοντας αιτήσεις HTTP παρόμοιες με αυτή που άρχισε τον έλεγχο. Για παράδειγμα για να ζητήσει ο Portal Server την κατάσταση ενός ελέγχου στέλνει μια αίτηση της μορφής ένα «GET /status?id=<scan-id>». Ο proxy θα αναλάβει τότε να επικοινωνήσει με το Nessus Client και να επιστρέψει την κατάσταση του ελέγχου στον Portal Server σαν απάντηση της HTTP αίτησης. Με παρόμοιο τρόπο επιστρέφονται και τα τελικά αποτελέσματα ενός ελέγχου.

3.1.2. Διεπαφή Portal Server – Auth Server

Η λειτουργικότητα του Auth Server μπορεί σχετικά εύκολα να υλοποιηθεί μέσα στον Portal Server. Στην περίπτωση αυτή οι αιτήσεις προς τον Auth Server θα γίνονται με τη χρήση κάποιου εσωτερικού API που εξαρτάται από την τεχνολογία υλοποίησης του Portal Server, π.χ. με μορφή συναρτήσεων PHP. Αν αποφασιστεί τα δύο υποσυστήματα να δουλεύουν σε χωριστά μηχανήματα, τότε θα χρησιμοποιηθεί πάλι το πρωτόκολλο HTTP για την επικοινωνία τους, με τρόπο ανάλογο που θα χρησιμοποιηθεί και για την επικοινωνία του Portal Server με το Nessus Host.

3.1.3. Διεπαφή Portal Server – Report Server

Όπως έχουμε ήδη αναφέρει, οι μηχανισμοί επικοινωνίας Portal Server – Report Server έρχονται έτοιμοι μαζί με κάθε τεχνολογία υλοποίησης για Portals. Οι μηχανισμοί αυτοί επιτρέπουν την επικοινωνία των δύο υποσυστημάτων με τη χρήση επερωτήσεων SQL. Αν τα δύο υποσυστήματα βρίσκονται σε διαφορετικά μηχανήματα, οι επερωτήσεις και οι απαντήσεις σε αυτές μεταφέρονται αυτόματα πάνω από συνδέσεις TCP.

3.1.4. Διεπαφή Portal Server – VPN Router

Και η επικοινωνία Portal Server – VPN Router είναι πολύ βολικό να γίνει με τη χρήση του πρωτοκόλλου HTTP. Για παράδειγμα, για να επιτραπεί η δημιουργία ενός νέου VPN ο Portal Server θα κάνει μια HTTP αίτηση της μορφής: «GET /newvpn?endpoint=<IP>&scanned=<IP range>». Η χρήση του HTTP πρωτοκόλλου σε όσες επικοινωνίες είναι δυνατό θα διευκολύνει την αποσφαλμάτωση του συστήματος. Επίσης θα επιταχύνει και την υλοποίηση, καθώς υπάρχουν πολλές έτοιμες λύσεις για το χειρισμό του HTTP όποια τεχνολογία και αν επιλεγεί για κάθε υποσύστημα.

4. Σύνοψη σχεδιασμού του συστήματος ΕΣΤΙΑ

Στο έγγραφο αυτό περιγράψαμε το σχεδιασμό του συστήματος ΕΣΤΙΑ. Το σύστημα ΕΣΤΙΑ θα είναι μια αυτόματη υπηρεσία ελέγχου των προσωπικών υπολογιστών. Η υπηρεσία αυτή θα προσφέρεται μέσα από το Διαδίκτυο δίνοντας την δυνατότητα στον απλό χρήστη να ελέγξει και να βελτιώσει την ασφάλεια του υπολογιστή του.

Το σύστημα ΕΣΤΙΑ έχει σχεδιαστεί ως ένα σύνολο από υποσυστήματα τα οποία συνεργάζονται μεταξύ τους και μας δίνουν τελικά την επιθυμητή λειτουργικότητα. Περιγράψαμε το κάθε υποσύστημα ξεχωριστά καθώς και τις αλληλεπιδράσεις μεταξύ τους. Κατά την περιγραφή του κάθε υποσυστήματος παρουσιάσαμε τα κρίσιμα θέματα που το αφορούν και τα οποία θα πρέπει να επιλυθούν. Τα σημαντικότερα από αυτά είναι η τεχνολογία παρουσίασης της προόδου ενός ελέγχου στο χρήστη, ο βαθμός παραλληλισμού που θα πρέπει να χρησιμοποιήσουμε στο Nessus Host, ο τρόπος που θα ξεπεράσουμε τους τυχόν περιορισμούς στην επικοινωνία του Ελεγχόμενου Υπολογιστή. Για κάθε ένα από τα θέματα αυτά σκιαγραφήσαμε τις επιλογές που έχουμε, δείχνοντας το δρόμο στον οποίο θα κινηθούμε κατά την υλοποίηση του συστήματος.

Τέλος περιγράψαμε τις διεπαφές των υποσυστημάτων. Επιλέξαμε οι διεπαφές να χρησιμοποιούν το πρωτόκολλο HTTP όπου είναι δυνατό έτσι ώστε να διευκολυνθεί η υλοποίηση αλλά και η αποσφαλμάτωση του συστήματος.

Έχουμε την πίστη πως ο σχεδιασμός που παρουσιάστηκε στο έγγραφο αυτό θα αποτελέσει μια στέρεα βάση για την υλοποίηση και έπειτα λειτουργία του συστήματος ΕΣΤΙΑ.

Παραπομπές

[1] Τεχνολογία Java Applets: http://java.sun.com/applets

[2] Τεχνολογία Flash: http://www.macromedia.com/software/flash/flashpro/

[3] Λογισμικό Nessus: http://www.nessus.org

[4] XML HTTP: http://en.wikipedia.org/wiki/XMLHttpRequest

[5] JavaScript: http://en.wikipedia.org/wiki/JavaScript

[6] Το πρωτόκολλο SSH: http://www.ietf.org/html.charters/secsh-charter.html

[7] Το πρωτόκολλο SSL: http://wp.netscape.com/eng/ssl3/

[8] Virtual Private Networks: http://en.wikipedia.org/wiki/Vpn