ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ασφαλής διαμοιρασμός πόρων με εξασφάλιση ανωνυμίας πρόσβασης σε περιβάλλοντα προσωπικών δικτύων.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

ΑΝΔΡΕΑ ΣΚΟΥΦΗ
ΣΩΤΗΡΗ ΣΤΑΜΟΚΩΣΤΑ

Επιβλέπων : Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2010

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ασφαλής διαμοιρασμός πόρων με εξασφάλιση ανωνυμίας πρόσβασης σε περιβάλλοντα προσωπικών δικτύων.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

ΑΝΔΡΕΑ ΣΚΟΥΦΗ
ΣΩΤΗΡΗ ΣΤΑΜΟΚΩΣΤΑ

Επιβλέπων : Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Μαρτίου 2010.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Μιχαήλ Θεολόγου	Γεώργιος Στασινόπουλος	Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2010

(Υπογραφή)

.....

ΑΝΔΡΕΑΣ ΣΚΟΥΦΗΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

(Υπογραφή)

.....

ΣΩΤΗΡΙΟΣ ΣΤΑΜΟΚΩΣΤΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα θέλαμε να ευχαριστήσουμε τον καθηγητή κύριο Μιχαήλ Θεολόγου, ο οποίος μας εμπιστεύθηκε την εκπόνηση της συγκεκριμένης εργασίας, μας στήριξε σε όλη την διάρκεια της, προσφέροντας την σημαντική εμπειρία του, πολύτιμες συμβουλές, συνεχή επιστημονική υποστήριξη καθώς και γρήγορες και αποτελεσματικές λύσεις σε προβλήματα που συναντήσαμε στην διαδρομή.

Επιπλέον θέλουμε να ευχαριστήσουμε τον Δρ. Χαράλαμπο Πατρικάκη για την αρχική πρωτότυπη ιδέα πάνω στην οποία βασίστηκε η εργασία αυτή, την επίβλεψή της και την συνεχή υποστήριξή του κατά τη διάρκεια εκπόνησής της. Η πολύτιμη καθοδήγησή του, καθώς και η συνεχής συνεργασία μας οδήγησε στην δημοσίευση τμημάτων της εργασίας σε διεθνές συνέδριο [1].

Η σελίδα αυτή είναι σκόπιμα λευκή.

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη ενός συστήματος για τον ασφαλή διαμοιρασμός πόρων σε περιβάλλοντα προσωπικών δικτύων, με επιπλέον χαρακτηριστικά την εξασφάλιση της ανωνυμίας πρόσβασης, καθώς και τον χρονικό περιορισμό της παραχώρησης πρόσβασης. Για τον σκοπό αυτό αναπτύχθηκε το σύστημα RAST (Resource Access Security Tunnel), ένα πλήρες λογισμικό σύστημα, αποτελούμενο από ειδικά παραμετροποιημένο λειτουργικό σύστημα μαζί με συνοδευτικά προγράμματα για την λειτουργία του.

Συγκεκριμένα, το RAST αναπτύχθηκε πάνω στο Ubuntu Linux με εξειδικευμένες ρυθμίσεις αναγκαίες για την λειτουργία του συστήματος, και η εφαρμογή διαχείρισης αναπτύχθηκε σε περιβάλλον ιστοσελίδων, με χρήση των γλωσσών PHP και Perl. Είναι ένα σύστημα που υλοποιεί την αρχιτεκτονική του ασφαλούς ενδιάμεσου στον διαμοιρασμό πόρων, το οποίο εξασφαλίζει τον διαμοιρασμό πόρων με ανωνυμία και ασφάλεια, και βασίζεται στα πρωτόκολλα SMB και CIFS της Microsoft για να εξασφαλίσει ικανοποιητικό βαθμό συμβατότητας με συσκευές και υπολογιστές.

Η πλατφόρμα που αναπτύχθηκε βασίστηκε σε ανοιχτές αρχιτεκτονικές, οπότε μπορεί να επεκταθεί υποστηρίζοντας περισσότερα πρωτόκολλα και υπηρεσίες, αυξάνοντας έτσι το πεδίο συνεργασίας των εμπλεκομένων χρηστών και συσκευών. Ο σχεδιασμός της πάνω στο λειτουργικό Linux της επιτρέπει να υποστηρίξει την πλειονότητα των νέων τεχνολογιών, μιας και οι περισσότερες από αυτές υποστηρίζονται από το λειτουργικό ή σχετικές βοηθητικές εφαρμογές.

Στην εργασία αυτή, μετά από μια εισαγωγή στις υπάρχουσες αρχιτεκτονικές ασφάλειας και τον προσδιορισμό των κενών που υπάρχουν σε αυτές, παρατίθενται οι απαιτήσεις από ένα σύστημα διαμοιρασμού πόρων σε περιβάλλοντα προσωπικών δικτύων. Στην συνέχεια παρουσιάζεται το σύστημα RAST που αναπτύχθηκε, οι τεχνικές προδιαγραφές του καθώς και παραδείγματα εφαρμογών του συστήματος. Έπειτα γίνεται αξιολόγηση του συστήματος και τέλος παρουσιάζονται προτάσεις για μελλοντική έρευνα.

Λέξεις Κλειδιά: <<Διαμοιρασμός πόρων, ασφάλεια, Linux, Samba, Apache, CUPS, PHP, Perl, προσωπικά δίκτυα, ανωνυμία, πρόσβαση>>

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The purpose of this thesis was to develop a system for secure sharing of resources in personal networks environments, with additional features to ensure the anonymity of access and the time limit the access grant. To this end, the system RAST (Resource Access Security Tunnel) was developed, a complete software system, consisting of specially parameterized operating system together with accompanying programs for its operation.

Specifically, RAST was developed on the Ubuntu Linux platform with a special configuration necessary for the operation of the system, along with an administrative application developed in a web environment, using the languages PHP and Perl. It is a system that implements the architecture of a secure intermediary in resource sharing, which ensures the sharing of resources with anonymity and security, and is based on the SMB and CIFS protocols of Microsoft to ensure a satisfactory level of compatibility with devices and computers.

The platform was developed based on open architectures, so it may be extended by supporting more protocols and services, thereby increasing the cooperation of the participating users and devices. The design on the Linux operating system enables support for the majority of new technologies, since most of them are supported by the operating or relevant applications.

In this thesis, after an introduction to the existing security architectures and the identification of their drawbacks, the requirements of a resource sharing system in personal networks environments are presented. Subsequently, the RAST system is presented, along with its specifications and examples of its applications. After that, the system is assessed, and finally suggestions for future research are presented.

Keywords: <<Resource sharing, security, Linux, Samba, Apache, CUPS, PHP, Perl, personal networks, anonymity, access>>

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας Περιεχομένων

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας Σχημάτων

1

Πρόλογος

Οι πρόσφατες εξελίξεις και σημαντικές καινοτομίες που παρουσιάζονται στον τομέα της δικτύωσης και γενικότερα στον τομέα της μετάδοσης της πληροφορίας αλλάζουν καθημερινά τις δυνατότητες που έχουμε για πρόσβαση στην πληροφορία. Καθημερινά παρουσιάζονται, τόσο στον τομέα της έρευνας, αλλά ακόμα περισσότερο στην αγορά, ασύρματες και κινητές τεχνολογίες, οι οποίες, συνδυασμένες με σύγχρονα εξειδικευμένα προγράμματα και εφαρμογές επιτρέπουν την απόλυτα φορητή πρόσβαση τόσο σε πληροφορίες, όσο και σε υπολογιστικούς πόρους. Ταυτόχρονα, η εξέλιξη των peer-to-peer δικτύων έχει προωθήσει μια νέα γενιά αποκεντρωμένων δικτυακών αρχιτεκτονικών, χωρίς κεντρική διαχείριση.

Έτσι, η έννοια του γραφείου ή του προσωπικού χώρου εργασίας διαρκώς επαναπροσδιορίζεται. Οι γεωγραφικοί περιορισμοί αίρονται, η έννοια του κλασσικού προσωπικού υπολογιστή αλλάζει, και οι διαχωρισμοί μεταξύ υπολογιστών και προσωπικών ηλεκτρονικών συσκευών ολοένα γίνονται και πιο δυσδιάκριτοι. Η πρόσβαση στην πληροφορία και σε άλλους ηλεκτρονικούς πόρους μετατρέπεται σταδιακά σε κυρίαρχο αίτημα, αλλά και διαρκώς βελτιούμενη κατάκτηση από τις νέες τεχνολογίες, είτε πρωτοπαρουσιαζόμενες, είτε ώριμες προσαυξημένες εκδόσεις των παλαιοτέρων. Το παραδοσιακό γραφείο, σε ό,τι αφορά την συγκεκριμένη τοποθεσία και την περιορισμένη πρόσβαση σε πόρους, εγκαταλείπεται, και σταδιακά αντικαθίσταται από ένα εικονικό γραφείο με καθολική και απεριόριστη πρόσβαση στην επιχείρηση, αλλά και στα προσωπικά δεδομένα του χρήστη, απελευθερωμένο από περιορισμούς χώρου και χρόνου.

Αποτέλεσμα όλων αυτών των εξελίξεων αποτελεί η έννοια των προσωπικών δικτύων (Personal Networks, PNs), δηλαδή ένα ομογενοποιημένο σύνολο ετερογενών τεχνολογιών δικτύωσης, ενοποιημένες σε ένα δίκτυο με επίκεντρο τον χρήστη. Μέσω του δικτύου αυτού, καθίσταται απλούστερη και διαφανής η προσφορά νέων υπηρεσιών και εφαρμογών για αυτόν. Παραδείγματα τέτοιου χαρακτήρα ετερογενών δικτύων είναι στη δεδομένη περίπτωση το «έξυπνο» σπίτι και το «έξυπνο αυτοκίνητο» [1].

Εντούτοις, όλες οι εξελίξεις αυτές εισάγουν και αυξημένη επικινδυνότητα. Η δυνατότητα παροχής πόρων πάντα και παντού, εισάγει τον αυξημένο κίνδυνο της ασφάλειας και τις μυστικότητας / ανωνυμίας [2]. Για την αντιμετώπιση των κινδύνων αυτών, εισάγονται περιορισμοί και αναπτύσσονται μηχανισμού προστασία, προκειμένου να προστατευθεί ο κινητός χρήστης από κακόβουλες προσπάθειες να εισβάλλουν στον προσωπικό του χώρο ή από επιθέσεις κατά του δικτύου. Η πλειονότητα των μηχανισμών αυτών όμως είναι συνήθως βασισμένοι στα παραδοσιακά συστήματα ασφάλειας δικτύων, τα οποία όμως έχουν σχεδιαστεί κυρίως για την αποτροπή προσανατολισμένων προς το δίκτυο (network oriented) απειλών, χωρίς ενσωμάτωση των πιο πρόσφατων εξελίξεων στις κινητές τηλεπικοινωνίες.

Οι δυο επικρατέστερες αρχιτεκτονικές ασφάλειας είναι η κεντρική και η αποκεντρωμένη. Στην περίπτωση της κεντρικής ασφάλειας, οι μηχανισμοί επιβολής βασίζονται στην διαχείριση ενός κεντρικού υπολογιστή, στον οποίο όλες οι πληροφορίες για τους χρήστες και τα δικαιώματά τους αποθηκεύονται σε μια κεντρική βάση δεδομένων. Στην περίπτωση πάλι της αποκεντρωμένης ασφάλειας, τα δικαιώματα πρόσβασης των πόρων ανήκουν και ρυθμίζονται από τους ιδιοκτήτες των πόρων.

Σε όλες τις παραπάνω περιπτώσεις, η δυνατότητα υποστήριξης εντός συστήματος πρόσβασης σε πόρους, το οποίο να μπορεί να διαχειριστεί ευέλικτα τα δικαιώματα πρόσβασης σε περιβάλλοντα peer-to-peer, θέτοντας περιορισμούς, τόσο χρονικούς, όσο και τοπικούς, δεν υποστηρίζεται επαρκώς. Επιπλέον, για να πετύχει η πρόσβαση πόρων σε συσκευές, πρέπει να υπάρχει διαθέσιμο ένα κοινό σύνολο προτύπων, προγραμμάτων και οδηγών (framework) για να την υποστηρίξει.

Σε αυτήν την διπλωματική εργασία, παρουσιάζουμε ένα σύστημα διαχείρισης πρόσβασης σε πόρους με ασφαλή και προσωρινό χαρακτήρα κάνοντας χρήση της αρχιτεκτονικής των peer-to-peer δικτύων. Κύρια χαρακτηριστικά του συστήματος

είναι η συμβατότητα με τους προσωπικούς υπολογιστές ανεξαρτήτως, όσο αυτό ήταν δυνατό, από το λειτουργικό σύστημα, καθώς και η δυνατότητα επέκτασης σε φορητές και κινητές συσκευές.

Η φιλοσοφία του συστήματος αυτού βασίστηκε στις εξής δύο απαιτήσεις:

- Η διαχείριση των πόρων και καθώς και της πρόσβασης σε αυτούς εκτελούνται χωριστά από κάθε χρήστη, χωρίς την ανάγκη ύπαρξης εξειδικευμένων και συγκεντρωτικών συστημάτων ή της περίπλοκης ασφάλειας που προστατεύει τα συστήματα αυτά.
- Η πρόσβαση στους πόρους είναι βασισμένη στη δημιουργία προσωρινών λογαριασμών χρηστών που αντιστοιχίζονται στους διαθέσιμους πόρους πάντα με συγκεκριμένους χρονικούς περιορισμούς.

2

Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται συνοπτικά οι τεχνολογικές εξελίξεις, κυρίως σε επίπεδο συσκευών, που οδηγούν στην κατάργηση της έννοιας του παραδοσιακού γραφείου και των κλασσικών δικτυακών αρχιτεκτονικών. Στη συνέχεια παρουσιάζονται οι εξελίξεις δικτύων, τόσο σε ό,τι αφορά τα πρωτόκολλα, όσο και τις δικτυακές αρχιτεκτονικές. Μετά υπογραμμίζονται τα θέματα ασφάλειας και πρόσβασης που προκύπτουν από το τοπίο που προκύπτει, και τέλος παρουσιάζουμε τις ανάγκες που προκύπτουν και αντιμετωπίζεται από την διπλωματική αυτή εργασία.

2.1 Τεχνολογικές Εξελίξεις

2.1.1 Φορητές συσκευές

Την τελευταία εικοσαετία παρατηρείται μία έκρηξη σε ότι αφορά της ψηφιακές φορητές συσκευές. Καθημερινά κυκλοφορούν νέες που παρέχουν καινοτόμες λειτουργίες και δυνατότητες, και οι παλαιότερες εμπλουτίζονται σε λειτουργικότητα. Είναι πλέον σύνηθες κάθε μια από αυτές να περιλαμβάνει, αντί των παλαιών ηλεκτρονικών κυκλωμάτων, έναν μικροεπεξεργαστή, και οι λειτουργίες της συσκευής να υλοποιούνται σε επίπεδο λογισμικού. Έχοντας πλέον ώριμες πλατφόρμες ειδικών λειτουργικών συστημάτων με δυνατότητες συγκρίσιμες αυτών των προσωπικών υπολογιστών, δημιουργούν νέα δεδομένα στον τρόπο εργασίας

αλλά και καθημερινής χρήσης, είτε για χρηστικές εργασίες, είτε για ψυχαγωγία, και επαναπροσδιορίζουν τον τρόπο που αντιμετωπίζουμε την πληροφορική σήμερα.

Ξεκινώντας από τους φορητούς υπολογιστές (laptops), όπου έχουμε έναν πλήρη υπολογιστή έτοιμο να δουλέψει ακόμα και χωρίς παροχή ηλεκτρικού ρεύματος (μέσω της μπαταρίας που διαθέτει), ο περιορισμός του υπολογιστή στο γραφείο ή το σπίτι άρθηκε, και πλέον σήμερα είναι διαθέσιμα στο εμπόριο σε πολύ μικρά μεγέθη, μικρού βάρους, μέσω των οποίων ο χρήστης μπορεί να μεταφέρει το γραφείο του όπου κι αν βρίσκεται.

Παράλληλα, η ανάπτυξη των προσωπικών ψηφιακών βοηθών τσέπης (PDA), συνδυασμένη με την ανάπτυξη των έξυπνων κινητών τηλεφώνων (Smartphone) ενισχύει περεταίρω την φορητή πληροφορική. Ο χρήστης πλέον μπορεί να προμηθευτεί συσκευές τσέπης, βάρους λίγων γραμμαρίων, οι οποίες έχουν λειτουργικότητα συγκρίσιμη με αυτή ενός προσωπικού υπολογιστή. Και με δεδομένη την ευρύτητα της διάδοσης της κινητής τηλεφωνίας, η διάδοση των συσκευών αυτών διευρύνεται σε έντονο βαθμό.

Όμως, η ενσωμάτωση μικροεπεξεργαστών και λογισμικού δεν σταματάει σε αυτές μόνο τις συσκευές. Ολοένα και περισσότερες παραδοσιακές συσκευές αλλάζουν, μετατρέπονται σε ψηφιακές και αποκτούν μικροεπεξεργαστές. Συσκευές όπως η τηλεόραση και η φωτογραφική μηχανή πλέον εμπλουτίζονται με νέες δυνατότητες μέσω λογισμικού, αλλά ακόμα και στο αυτοκίνητο, παρουσιάζονται συσκευές όπως οι προσωπικοί βοηθοί πλοήγησης (PNA / GPS) ή και συσκευές ενσωματωμένες σε αυτό, που με την παρουσία κατάλληλου λογισμικού μετατρέπονται σε συσκευές που φέρουν σε πέρας λειτουργικότητες που παλαιότερα συναντούσαμε μόνο στους υπολογιστές [3].

2.1.2 Τεχνολογίες και αρχιτεκτονικές

Φυσικά, όλες αυτές οι εξελίξεις δημιουργούν και νέες ανάγκες στην επιστήμη της πληροφορικής. Ίσως το σημαντικότερο πρόβλημα που προκύπτει από όλες αυτές τις καινοτομίες είναι το θέμα της μεταξύ τους επικοινωνίας και συνεργασίας. Πράγματι, η λειτουργικότητά τους περιορίζεται αισθητά όταν κάθε συσκευή λειτουργεί ανεξάρτητα από την άλλη, και ο χρήστης πρέπει να καταβάλλει σημαντικές προσπάθειες για να τις αξιοποιήσει όλες. Το πρόβλημα εντείνεται όταν εισάγουμε και το θέμα της συνεργασίας, με τους χρήστες να έχουν ανάγκη για επικοινωνία όχι μόνο των συσκευών τους, αλλά και μεταξύ αυτών και άλλων

χρηστών. Τέλος, με την αισθητή διείσδυση του διαδικτύου (internet) όλες αυτές οι συσκευές μπορούν να επεκτείνουν την λειτουργικότητα και τη χρησιμότητά τους.

Για να καλυφθούν οι ανάγκες αυτές, έχουν αναπτυχθεί και αναπτύσσονται πληθώρα τεχνολογιών, που διευκολύνουν την επικοινωνία και την δικτύωση υπολογιστών και ψηφιακών συσκευών. Οι σημαντικότερες από αυτές θα παρουσιαστούν στο κεφάλαιο αυτό.

2.1.2.1 Ασύρματο δίκτυο Wi-Fi 802.11

Σχήμα 1: Το λογότυπο του Wi-Fi (πηγή: http://wifi.org)

Η τεχνολογία αυτή επιτρέπει τη δικτύωση συσκευών σε μέσες αποστάσεις, τάξης μεγέθους 100 μέτρων, και την επικοινωνία μεταξύ τους, ή ακόμα και με το δίκτυο, αν στο δίκτυο υπάρχει σχετικός δρομολογητής. Σταδιακά εμφανίζεται και η τεχνολογία WiMAX, με το πρωτόκολλο IEEE 802.16, η οποία επιτρέπει σημαντικά αυξημένες αποστάσεις και εύρος ζώνης [4].

2.1.2.2 Bluetooth

Σχήμα 2: Το λογότυπο του Bluetooth (πηγή: http://bluetooth.com)

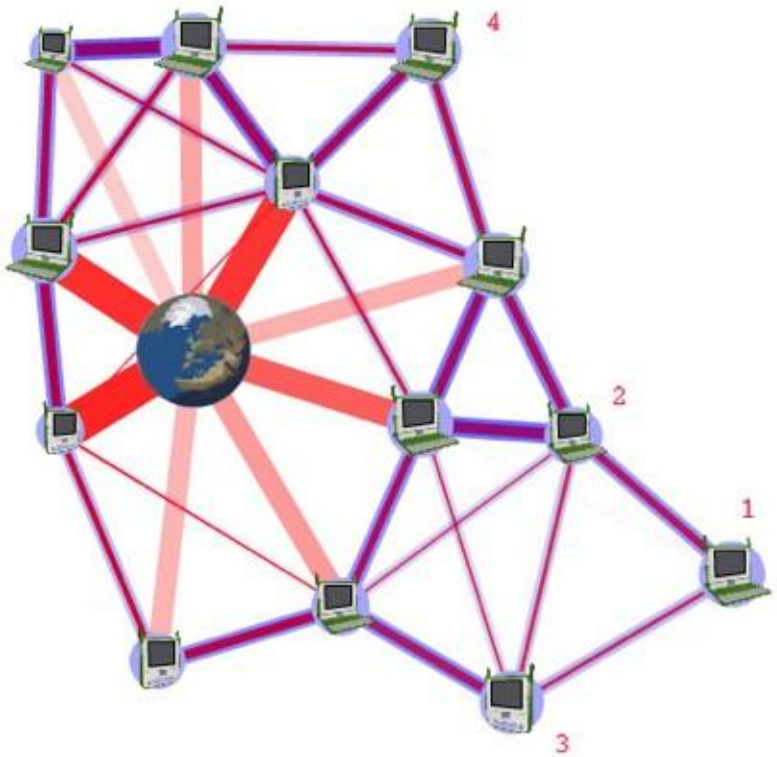
Με το Bluetooth παρέχεται η δυνατότητα επικοινωνίας συσκευών σε μικρές μεταξύ τους αποστάσεις, τάξης μεγέθους των 5 μέτρων. Το Bluetooth επιτρέπει αφενός την δικτύωση, αλλά έχει και κάποιες εξειδικευμένες λειτουργίας συνεργασίας συσκευών, όπως την ασύρματη μετάδοση ήχου, η ασύρματη σύνδεση περιφερειακών σε υπολογιστές και η μετάδοση πληροφοριών από ηλεκτρονικές συσκευές προχωρημένης ιατρικής [5]. Με Bluetooth υλοποιούνται τα συνηθέστερα προσωπικά δίκτυα, όπως θα δούμε παρακάτω.

2.1.2.3 Δίκτυα κινητής τηλεφωνίας τρίτης γενιάς (3G)

Η εξέλιξη του δικτύου κινητής τηλεφωνίας, από απλό δίκτυο μετάδοσης φωνής και δεδομένων σε χαμηλές ταχύτητες, σε σημαντικά υψηλότερες, έχει φέρει τα δίκτυα τρίτης γενιάς στο προσκήνιο. Το δίκτυο αυτό πλέον υποστηρίζει υψηλά εύρη ζώνης (μέχρι 21.6Mbps), και βοηθά τη δικτύωση με μεγάλη γεωγραφική κάλυψη, δεδομένης και της εδραίωσης των εταιριών κινητής τηλεφωνίας, άρα και την ύπαρξη πολλών σταθμών βάσης [6].

2.1.2.4 Τεχνολογίες πλεγματοειδών δικτύων (mesh networks)

Υπάρχουν περιπτώσεις που έχουμε ένα σύνολο υπολογιστών, οι οποίοι δεν έχουν πρόσβαση στον σταθμό βάσης, ώστε πχ. να έχουν σύνδεση στο διαδίκτυο. Παράδειγμα τέτοιας περίπτωσης είναι τα παιδιά ενός σχολείου, που ορισμένα από αυτά δεν βρίσκονται σε σημείο κάλυψης ενός σταθμού βάσης (access point) ασυρμάτου δικτύου. Για να λυθεί αυτό, έχουν υλοποιηθεί πρωτόκολλα πλεγματοειδούς δικτύωσης, τα οποία χαρακτηρίζονται από τις ανεξάρτητες συνδέσεις μεταξύ των συμμετεχόντων κόμβων [7]. Η πιο ευρεία εγκατάσταση τέτοιου τύπου δικτύων είναι οι σχολικοί υπολογιστές του ιδρύματος One Laptop Per Child, OLPC-XO1 [8]. Ακολουθεί ένα σχήμα που επιδεικνύει αναλυτικότερα πως λειτουργεί το δίκτυο αυτό.



Σχήμα 3: Παράδειγμα του πλεγματοειδούς δικτύου από το OLPC (πηγή: Wikipedia)

Όπως φαίνεται, ο κόμβος 4 δεν μπορεί να επικοινωνήσει απευθείας με τον κόμβο 2, οπότε το πρωτόκολλο κάνει αυτόματη δρομολόγηση μέσω του μεταξύ τους κόμβου. Επιπλέον φαίνεται πώς οι κόμβοι 1,2,3,4 δεν έχουν απευθείας πρόσβαση στον σταθμό βάσης που παρέχει πρόσβαση στο διαδίκτυο (φαίνεται στο σχήμα με την εικόνα της γης), οπότε χρησιμοποιούν επικοινωνία μέσω ενδιάμεσων κόμβων.

Μέσα από το σχήμα αυτό φαίνεται η βασική φιλοσοφία των πλεγματοειδών δικτύων, και δεν θα επεκταθεί περεταίρω. Το θέμα αυτό προφανώς δεν μπορεί να καλυφθεί σε αυτή την παράγραφο, καθώς είναι ένας τομέας έρευνας που παρουσιάζει έντονη κινητικότητα, με καθημερινές καινοτομίες.

Εκτός του ιδρύματος One Laptop Per Child, τα mesh δίκτυα έχουν χρησιμοποιηθεί εκτενώς για δίκτυα αισθητήρων, ώστε να περιορίζεται η ανάγκη όλοι οι αισθητήρες να έχουν πρόσβαση στον σταθμό βάση. Παράδειγμα τεχνολογίας πλεγματοειδούς δικτύωσης δικτύων αισθητήρων είναι το δίκτυο ZigBee [9].

2.1.2.5 Τεχνολογίες αυτόματης συνεργασίας συσκευών / έξυπνου γραφείου – σπιτιού

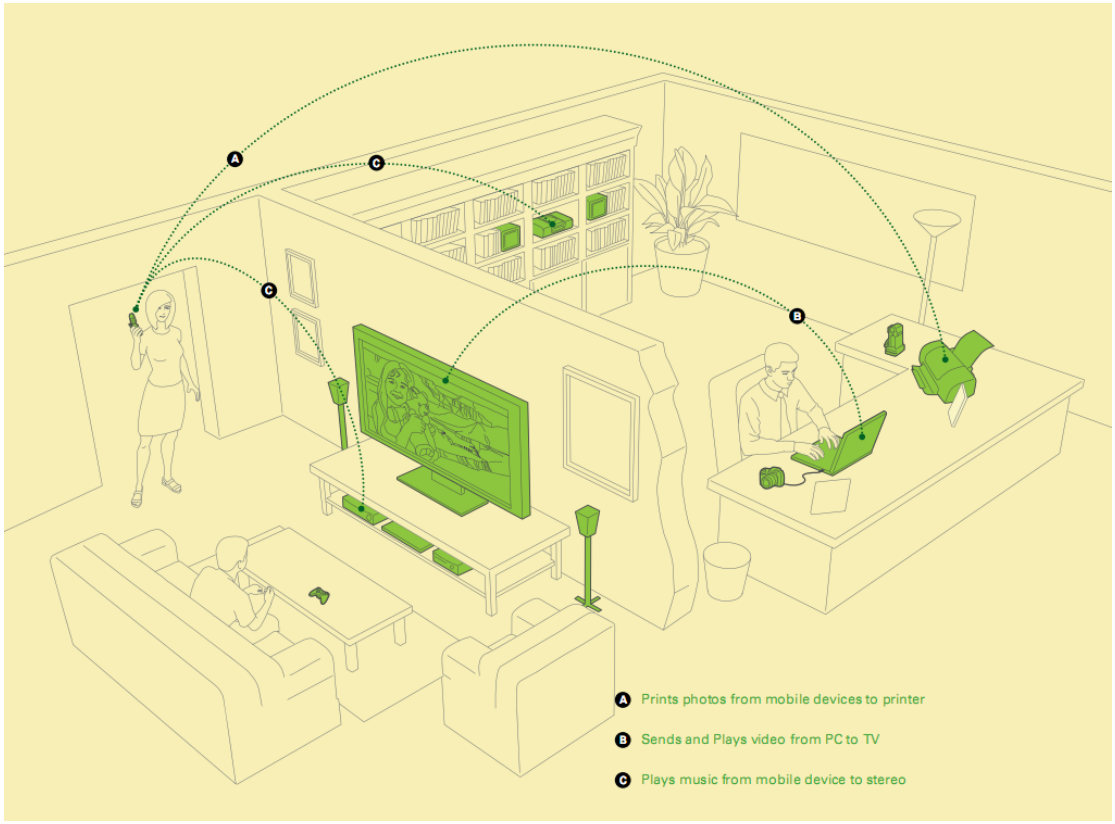
Ο ολοένα αυξανόμενος αριθμός ψηφιακών συσκευών που έχει είτε ένα σπίτι, είτε ένα γραφείο καθιστά δυσκολότερη τη ρύθμιση όλων των συσκευών ώστε να συνεργαστούν μεταξύ τους. Ορισμένες φορές μάλιστα, η συνεργασία είναι αδύνατη λόγω ανυπαρξίας κοινών πρωτοκόλλων επικοινωνίας μεταξύ τους.

Για το λόγο αυτό έχουν αναπτυχθεί και αναπτύσσονται εξειδικευμένα πρωτόκολλα και πρότυπα αυτοματοποίησης της επικοινωνίας και της ρύθμισης των συσκευών. Για την αυτόματη ρύθμιση ώστε να γίνεται η ανακάλυψη συσκευών στο δίκτυο έχει αναπτυχθεί το πρότυπο Universal Plug and Play (UPNP) [10], το οποίο επιτρέπει την δικτύωση χωρίς κάποια ρύθμιση (zero configuration network). Για επικοινωνία και ανταλλαγή πόρων, έχουν αναπτυχθεί πρότυπα, τα περισσότερα με έμφαση στην εικόνα και τον ήχο, όπως το UPNP AV και το Digital Living Network Alliance, DLNA [11].

Σχήμα 4: Το λογότυπο του DLNA (πηγή: http://dlna.org)

Με τις τεχνολογίες αυτές, καθώς και άλλες που αναπτύσσονται καθημερινά ή είναι διαθέσιμες στο εμπόριο, πραγματοποιούνται οι έννοιες του «έξυπνου σπιτιού» και του «έξυπνου γραφείου», χώρους δηλαδή που όλες οι συσκευές επικοινωνούν και συνεργάζονται με αυτόματο τρόπο επεκτείνοντας έτσι τις δυνατότητες που θα είχαν, αν λειτουργούσαν ανεξάρτητα.

Παρακάτω παρατίθεται ένα σχήμα από συσκευές που μετέχουν σε ασύρματο δίκτυο, στο οποίο φαίνονται οι δυνατότητες επικοινωνίας συσκευών μέσω του προτύπου DLNA.



Σχήμα 5: Δυνατότητες επικοινωνίας "έξυπνου σπιτιού" (πηγή: http://dlna.org)

Ωστόσο, στις περιπτώσεις χρήσης αυτές ενέχεται ο κίνδυνος της μη εξουσιοδοτημένης πρόσβασης στους πόρους από τρίτους χρήστες, και η έκθεση σε κακόβουλες ενέργειες στο δίκτυο αυτό. Για παράδειγμα, ένας χρήστης μπορεί να μπει σε κάποιο οικιακό δίκτυο και να υποκλέψει αρχεία ή να προκαλέσει βλάβες λογισμικού στις συσκευές που μετέχουν, εάν το δίκτυο δεν είναι επαρκώς προστατευμένο.

Το πρόβλημα αυτό γίνεται εντονότερο όταν το δίκτυο έχει πιο περίπλοκο χαρακτήρα, όπως στην περίπτωση ενός γραφείου. Συγκεκριμένα, ο αριθμός των χρηστών είναι μεγαλύτερος, οι προσφερόμενες από το δίκτυο υπηρεσίες είναι

περισσότερες και πολυπλοκότερες, και προκύπτουν συχνά εξειδικευμένες ανάγκες, όπως η παροχή πρόσβασης σε κάποιον επισκέπτη ή πελάτη. Έτσι, ένας κακόβουλος χρήστης έχει περισσότερες δυνατότητες απόκτησης πρόσβασης σε ένα τέτοιο δίκτυο, με δυσμενείς για το γραφείο συνέπειες, όπως την υποκλοπή αρχείων ή την πρόκληση προβλημάτων στη λειτουργία του γραφείου.

2.1.3 Ζητήματα πρόσβασης και ασφάλειας

Δεδομένων όλων αυτών των εξελίξεων, τίθενται ορισμένα ζητήματα που αφορούν την πρόσβαση και την ασφάλεια των δικτύων που προκύπτουν, καθώς και των πόρων που διαμοιράζονται σε αυτά. Τα βασικά ζητήματα λοιπόν είναι αυτά της εμπιστοσύνης, της ανωνυμίας, της συμβατότητας και της φιλικότητας προς το χρήστη.

Στη συνέχεια παρουσιάζονται συνοπτικά τα ζητήματα αυτά.

2.1.3.1 Εμπιστοσύνη

Για την ασφαλή μετάδοση πληροφοριών ενδεχομένως προσωπικού χαρακτήρα ή άλλων σημαντικών πληροφοριών, είναι σημαντικό ο χρήστης να μπορεί να επιλέξει ένα δίκτυο εμπιστοσύνης, και εφόσον αυτή η εμπιστοσύνη έχει δηλωθεί κατόπιν να παρέχεται πρόσβαση στους πόρους που θέλει να μοιραστεί [2].

2.1.3.2 Ανωνυμία

Στις περιπτώσεις που έχουμε δημόσιο διαμοιρασμό πόρων σε μια ομάδα χρηστών, για τους οποίους δεν μπορούμε να εξασφαλίσουμε τα θέματα της εμπιστοσύνης, όπως στα peer to peer δίκτυα, κρίνεται απαραίτητη η ύπαρξη μηχανισμού εξασφάλισης της ανωνυμίας. Με αυτό τον τρόπο κάποιος κακόβουλος χρήστης δεν μπορεί να ταυτοποιήσει κάποιον πάροχο του πόρου, ούτε να υποκλέψει άλλα προσωπικά δεδομένα [2].

2.1.3.3 Συμβατότητα

Για να είναι δυνατή η ανταλλαγή πόρων σε κάποιο δίκτυο, είναι απαραίτητος ένας βαθμός συμβατότητας μεταξύ των συσκευών / υπολογιστών που συμμετέχουν στο δίκτυο αυτό. Σε αντίθετη περίπτωση η ανταλλαγή πόρων είναι αδύνατη.

2.1.3.4 Φιλικότητα

Κάθε λύση που προτείνεται για την συνεργασία και δεν παρέχει φιλικότητα προς τον χρήστη είναι πρακτικά ανεφάρμοστη. Για το λόγο αυτό κάθε λύση που

προτείνεται πρέπει να χαρακτηρίζεται από ευκολία στην εκμάθηση για τους χρήστες, με όσο το δυνατόν μικρότερο αριθμό απλών βημάτων ώστε ο χρήστης να μπορεί να συνδεθεί και να ορίσει τις σχέσεις του.

2.2 Ανάγκη για ασφαλή παραχώρηση δικαιωμάτων πρόσβασης σε υπολογιστικούς πόρους

Είναι σύνηθες με την επίσκεψη ενός πελάτη στα γραφεία μιας εταιρίας να προκύπτει η ανάγκη για ανταλλαγή αρχείων και λοιπών πληροφοριών μεταξύ πελάτη και εταιρίας. Για να επιτευχτεί η ανταλλαγή αυτή απαιτείται να δημιουργηθεί ένα δίκτυο μεταξύ πελάτη και εταιρίας, και στην συνηθέστερη περίπτωση απλά παραχωρείται στον πελάτη μια πρόσβαση στο δίκτυο της εταιρίας.

Η ανάγκη αυτή τονίζεται περεταίρω αν λάβουμε υπόψη το σενάριο της επιθυμίας ανταλλαγής πληροφοριών μεταξύ συμμετεχόντων ενός συνεδρίου. Στα πλαίσια του συνεδρίου ένας ομιλητής πιθανώς θέλει να παραχωρήσει στους υπόλοιπους σύνεδρους σημειώσεις μιας ομιλίας του, και να λάβει σχολιασμούς, στοιχεία επικοινωνίας ή να ανταλλάξει λοιπές πληροφορίες.

Από τις περιπτώσεις αυτές που υπογραμμίστηκαν ενδεικτικά παρουσιάζεται η ανάγκη για ένα σύστημα παραχώρησης δικαιωμάτων πρόσβασης προσωρινού χαρακτήρα, το οποίο μάλιστα πρέπει να παρέχει ένα ικανοποιητικό επίπεδο ασφάλειας, ώστε οι προσβάσεις να είναι ασφαλείς και να αποκλείονται περιπτώσεις ανεπιθύμητης υποκλοπής πληροφοριών και λοιπών κακόβουλων ενεργειών.

2.3 Παρούσες αρχιτεκτονικές ασφάλειας

2.3.1 Βασικές αρχιτεκτονικές

Τα υπάρχοντα συστήματα ασφαλείας έχουν διάφορους περιορισμούς, οι περισσότεροι εκ των οποίων προκύπτουν από το γεγονός ότι σχεδιάστηκαν πριν από τις τελευταίες εξελίξεις. Έτσι, αδυνατούν να καλύψουν τις ανάγκες ελέγχου πρόσβασης που έχουν προκύψει.

Οι δυο πιο συνηθισμένες αρχιτεκτονικές ασφάλειας είναι οι εξής:

- **Συγκεντρωτική ασφάλεια** μέσω μιας κεντρικής βάσης χρηστών. Παράδειγμα αυτής της αρχιτεκτονικής είναι η ασφάλεια μέσω Domain Controller σε δίκτυα Windows, αλλά και το πρωτόκολλο Kerberos. Σε αυτήν την

περίπτωση, υπάρχει ένας κεντρικός εξυπηρετητής[1] που διατηρεί όλες τις πληροφορίες για τους χρήστες και τα δικαιώματα που έχουν [12].

- **Αποκεντρωμένη ασφάλεια**. Παράδειγμα αυτής της αρχιτεκτονικής είναι η ασφάλεια που υλοποιούν οι υπολογιστές με Windows τεχνολογίας NT (Windows NT4, Windows 2000, Windows XP, Windows Vista, Windows 7) σε ρύθμιση ομάδας εργασίας (workgroup) [13]. Σε αυτή την περίπτωση ο κάθε πάροχος πόρου διατηρεί πληροφορίες για τους χρήστες και τα δικαιώματά τους.

Οι δυο αυτές αρχιτεκτονικές όμως παρουσιάζουν ορισμένους περιορισμούς για περιπτώσεις προσωρινής δικτύωσης και παροχής πόρων.

Στην περίπτωση της συγκεντρωτικής ασφάλειας ένας από τους βασικούς περιορισμούς είναι ότι η διαχείριση λογαριασμών περιορισμένης διάρκειας δεν είναι εύκολη. Στο παράδειγμα μιας εταιρείας που χρησιμοποιεί κεντρικά έναν domain controller, θα πρέπει ο διαχειριστής να δημιουργήσει τους λογαριασμούς, με τυχαία στοιχεία για να διατηρήσει την ανωνυμία των χρηστών, και να τους εντοπίσει και διαγράψει μόλις τελειώσει ο χρόνος χρήσης. Αυτό, σε περιπτώσεις μεγάλου αριθμού χρηστών γίνεται μια εκτενής, δύσκολη κι επίπονη διαδικασία, με κίνδυνο να ξεχαστεί κάποιος λογαριασμός εντός του συστήματος. Ακόμα εντονότερη γίνεται η δυσκολία αν αναφερόμαστε σε σύνολο χρηστών με διαφορετικούς χρόνους λήξης ο καθένας.

Επιπλέον, στο ίδιο παράδειγμα, αναδεικνύεται ως αδύναμο σημείο του συστήματος ο domain controller, γιατί αν κάποιος κακόβουλος χρήστης εντοπίσει κάποιο κενό ασφαλείας, έχει αυτομάτως συνολική πρόσβαση στο δίκτυο της εταιρίας. Άρα, ένας διαχειριστής τέτοιου συστήματος πρέπει να είναι εξαιρετικά προσεκτικός στην εξασφάλιση της ασφάλειας του domain controller, ώστε να αφαιρεθεί τέτοια δυνατότητα. Βέβαια, δεδομένου του ρυθμού ανακάλυψης νέων «τρυπών» ασφαλείας σε όλα αυτά τα συστήματα, η απόφαση παροχής προσωρινής πρόσβασης σε κάποιον επισκέπτη παραμένει μια απόφαση με ένα σημαντικό ποσοστό ρίσκου.

Τέλος, ένα τέτοιο σύστημα απαιτεί κάποιου είδους συνεργασία μεταξύ ενός παρόχου πόρου και του διαχειριστή, για την σωστή λειτουργία, γεγονός που δυσκολεύει ακόμα περισσότερο την κατάσταση.

Η αποκεντρωμένη ασφάλεια έχει και αυτή σημαντικούς περιορισμούς. Αρχικά κάθε χρήστης πρέπει να εξασφαλίσει την ασφάλεια του υπολογιστή ή συσκευής του

[1] Στις περισσότερες περιπτώσεις μπορούν να εγκατασταθούν και παραπάνω του ενός εξυπηρετητές, αλλά η βάση χρηστών είναι μία, την οποία και συγχρονίζουν μεταξύ τους.

έναντι κάθε κακόβουλης ενέργειας. Κάτι τέτοιο όμως είναι απαγορευτικό για την πλειονότητα των χρηστών οι οποίοι δεν έχουν στη διάθεσή τους τις απαραίτητες τεχνικές γνώσεις για να εξασφαλιστούν.

Επιπλέον, είτε σε ότι αφορά ανεπάρκεια γνώσεων, είτε όμως και σε δυσκολία, απαγορευτική κρίνεται και η διαχείριση μεγάλου αριθμού χρηστών, ιδιαίτερα σε περιπτώσεις ανομοιομορφίας πόρων που πρέπει να μοιραστούν στον καθένα. Έτσι θα γίνεται αρκετά σύνηθες να μη διαγραφτεί κάποιος χρήστης όταν πρέπει, και η πρόσβαση να συνεχίσει να ισχύει.

2.3.2 Νεότερες αρχιτεκτονικές

Μεταξύ των δυο αρχιτεκτονικών που αναφέρθηκαν στο προηγούμενο κεφάλαιο, δημιουργείται μία «γκρίζα» ζώνη, στην οποία κινούνται πολλά υπάρχοντα συστήματα, καθώς και αυτό που αναπτύχθηκε για την διπλωματική αυτή. Ορισμένα παραδείγματα παρουσιάζονται παρακάτω.

2.3.2.1 Δίκτυα ίσος προς ίσο (peer to peer)

Σε αυτά τα δίκτυα κάθε χρήστης είναι ισότιμος με τους υπόλοιπους, και έτσι το σύνολο των χρηστών μπορεί να καταναλώσει όλους τους πόρους που μοιράζονται στο σύστημα, χωρίς κάποιον περιορισμό. Το σημαντικότερο παράδειγμα τέτοιου τύπου δικτύων είναι τα peer to peer προγράμματα διαμοιρασμού αρχείων [14].

2.3.2.2 Κοινωνικά δίκτυα

Σε αυτή την περίπτωση, ο κάθε χρήστης ορίζει σχέσεις εμπιστοσύνης ή φιλίας με καθέναν από τους υπόλοιπους χρήστες του δικτύου, και μέσω αυτών των σχέσεων προκύπτουν τα αντίστοιχα δικαιώματα χρήσης [15].

2.3.2.3 Προσωπικά δίκτυα

Ένα προσωπικό δίκτυο είναι ένα δίκτυο που χρησιμοποιείται για την επικοινωνία ψηφιακών συσκευών κοντά σε έναν χρήστη, ανεξάρτητα από το αν του ανήκουν ή όχι. Όμως, στα προσωπικά δίκτυα δεν αντιμετωπίζεται η συνεργασία των χρηστών μεταξύ τους. Στο παρακάτω κεφάλαιο προσδιορίζουμε τα ζητήματα που τίθενται για να υποστηριχθούν ομοσπονδίες χρηστών [16] από τα προσωπικά δίκτυα.

2.4 Προσδιορισμός των ζητημάτων στις ομοσπονδίες χρηστών μέσω προσωπικών δικτύων

Προκειμένου να υποστηριχθεί η συνεργασία των χρηστών σε προσωπικά δίκτυα, διάφορα ζητήματα πρέπει να αντιμετωπιστούν. Εκτός από τα τεχνολογικά ζητήματα που αφορούν ασυμβατότητες σε πρωτόκολλα, λειτουργικά συστήματα και εφαρμογές, ζητήματα όπως της ασφάλειας, της εμπιστοσύνης και της χρηστικότητας πρέπει να λυθούν αλλιώς δεν μπορεί να υποστηριχθεί κάποιος ισχυρισμός ότι η πλήρης υποστήριξη της συνεργασίας μεταξύ δύο χρηστών είναι εφικτή. Παρακάμπτοντας το ζήτημα των ασυμβατοτήτων που έχει αντιμετωπιστεί καλά μέσω της υιοθέτησης των κοινών πρωτοκόλλων και των τεχνολογιών όπως του Bluetooth, και επίσης μέσω συγκεκριμένων τεχνολογιών λογισμικού που επιτρέπουν σε ετερογενείς συσκευές να επικοινωνήσουν σε ένα κοινό πλαίσιο (framework), όπως η Java, πρέπει να αναπτύξουμε διεξοδικά τα ζητήματα της εμπιστοσύνης και της ασφάλειας, καθώς επίσης και αυτό της χρηστικότητας [1],[16].

2.4.1 Εμπιστοσύνη

Κάθε φορά που διαμορφώνονται ομοσπονδίες μεταξύ των χρηστών, αυτομάτως προκύπτουν θέματα εμπιστοσύνης. Η ανάγκη για μυστικότητα και προστασία των ευαίσθητων και προσωπικών δεδομένων, καθώς επίσης και η ανάγκη για επαλήθευση και εξουσιοδοτημένη πρόσβαση των χρηστών που μπορούν να έχουν πρόσβαση σε προσωπικούς πόρους συσχετίζεται άμεσα με ένα μηχανισμό διαχείρισης εμπιστοσύνης. Είναι επομένως επιτακτική η ανάγκη να διαφοροποιηθούμε από τις δύο προσεγγίσεις στη δημιουργία της εμπιστοσύνης:

- της προσέγγισης της εμπιστοσύνη γύρω από τις συσκευές
- της προσέγγισης της εμπιστοσύνη γύρω από τους χρήστες

Στην πρώτη περίπτωση της εμπιστοσύνης των συσκευών, αν και ολόκληρη η διαδικασία δημιουργίας της εμπιστοσύνης είναι διαφανής στο χρήστη, απαιτώντας την ελάχιστη αλληλεπίδραση για την ανταλλαγή των πληροφοριών έγκρισης, εισάγει το πρόβλημα της μη γνώσης της πραγματικής ταυτότητας του προσώπου στο οποίο τα δικαιώματα εμπιστοσύνης χορηγούνται, δεδομένου ότι ο ιδιοκτήτης της συσκευής μπορεί να αλλάξει, χωρίς αυτό να γίνει αντιληπτό. Στην δεύτερη περίπτωση, ορίζοντας χρήστες ως έμπιστους απαιτείται η συνεχής πιστοποίηση του χρήστη κάθε φορά που ζητείται η πρόσβαση σε πόρους καθώς και η αποθήκευση της πιστοποίησης

σε μία συσκευή. Με αυτόν τον τρόπο, δεν λύνουμε το πρόβλημα αλλά το μετασχηματίζουμε από πιστοποιημένου χρήστη σε έμπιστης συσκευής (αυτής που αποθηκεύονται τα στοιχεία προσδιορισμού του έμπιστου χρήστη), ενώ ταυτόχρονα εισάγεται ο κίνδυνος κλοπής ταυτότητας σε περίπτωση που η συσκευή κλαπεί ή χαθεί.

Ένα άλλο σημαντικό ζήτημα είναι η εμβέλεια και το πεδίο εφαρμογής της παρεχόμενης εμπιστοσύνης. Η εμπιστοσύνη είτε ενός ατόμου είτε μιας συσκευής δεν είναι απλή διαδικασία, δεδομένου ότι τα όρια των επιτρεπόμενων ενεργειών για τα οποία δίνεται η εμπιστοσύνη καθώς επίσης και η χρονική περίοδος στην οποία η εμπιστοσύνη έχει ισχύ είναι πολύ σημαντικά. Παραδείγματος χάριν, το δικαίωμα ενός χρήστη μιας ομοσπονδίας για χρήση ενός εκτυπωτή δεν σημαίνει αυτομάτως ότι και ο ίδιος χρήστης μπορεί να έχει δικαιώματα πρόσβασης στους σκληρούς δίσκους, ενώ ταυτόχρονα η άδεια για εκτύπωση δεν πρέπει να χορηγείται επ' αόριστο. Επιπλέον, ακόμη και εντός της περιόδου εμπιστοσύνης, η συμπεριφορά του χρήστη της ομοσπονδίας μπορεί να οδηγήσει σε πιθανή ανάκληση εμπιστοσύνης [2].

Όλα τα ανωτέρω, υπογραμμίζουν τη σημασία ενός ανεπτυγμένου μηχανισμού ανάθεσης εμπιστοσύνης (και των αντίστοιχων εργαλείων διαχείρισης δικαιωμάτων πρόσβασης) που πρέπει να είναι ευέλικτος και σταθερός, ώστε να ικανοποιηθούν όλες οι ανάγκες της διαχείρισης εμπιστοσύνης σε ομοσπονδίες χρηστών μέσω ειδικών προσωπικών δικτύων που ο καθένας κατέχει.

2.4.2 Διαφάνεια και χρηστικότητα

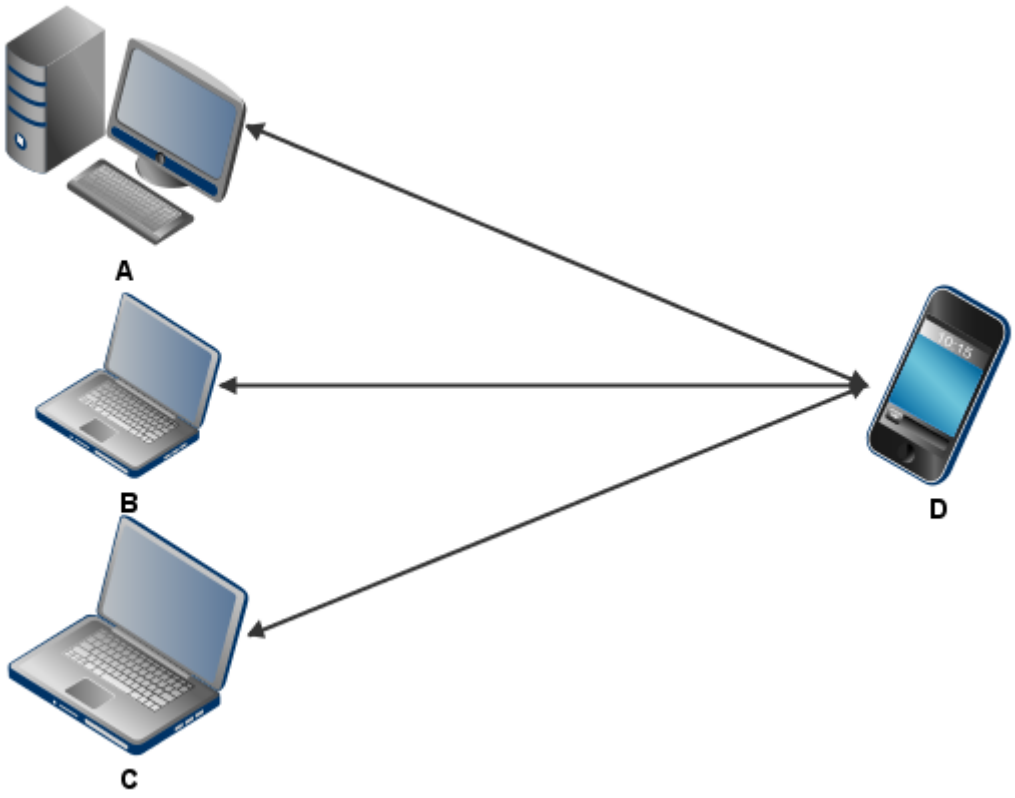
Στην περίπτωση όπου δύο συσκευές πρέπει να συνεργαστούν για την επίτευξη ενός συγκεκριμένου στόχου, είναι απαραίτητη η καθιέρωση μιας κοινής πλατφόρμας στην οποία μπορούν να εκτελεστούν η ανταλλαγή, η αποθήκευση και η διαχείριση των δεδομένων. Επιπλέον, όποτε οι περιφερειακές συσκευές που συμμετέχουν απαιτούν πρόσθετη διαχείριση από το λειτουργικό σύστημα (π.χ. χρήση συγκεκριμένων οδηγών), τότε η πλατφόρμα θα πρέπει να επεκταθεί έτσι ώστε να περιλάβει την απαραίτητη λειτουργικότητα που μπορεί να παρέχει τα απαιτούμενα. Η συνηθισμένη προσέγγιση για να αντιμετωπιστούν τέτοια ζητήματα είναι η χρήση ενός κοινού ή συμβατού λειτουργικού συστήματος με τους αντίστοιχους οδηγούς να ανταλλάσσονται και εγκαθίστανται σε συσκευές που μετέχουν στην ομοσπονδία, ή εναλλακτικά η χρήση των πρόσθετων εφαρμογών που τρέχουν και στις δύο συσκευές (συνήθως ως ένα μοντέλο client-server) παρέχοντας υπηρεσίες τηλεπρόσβασης (πχ

εφαρμογές απομακρυσμένης πρόσβασης σε τοπική επιφάνεια εργασίας), ώστε να παρέχεται απομακρυσμένη πρόσβαση σε τοπικούς πόρους μιας συσκευής.

Ωστόσο, καμία από τις δύο προαναφερθείσες εναλλακτικές λύσεις δεν αντιμετωπίζει άμεσα το ζήτημα της διαφανούς, ανεξάρτητης κοινής εκμετάλλευσης πόρων πλατφορμών μεταξύ των συσκευών κατά τρόπο ξεκάθαρο και διαφανή.

2.5 Αρχιτεκτονική έμπιστου ενδιάμεσου στην επικοινωνία

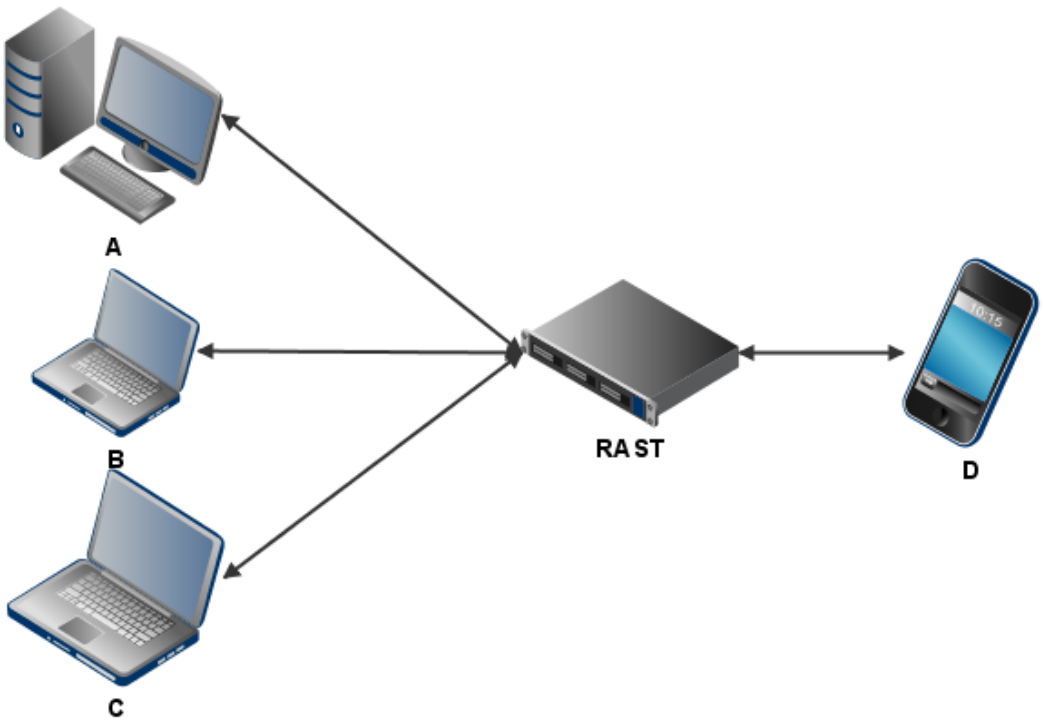
Η επικοινωνία σε peer to peer περιβάλλοντα έχει ένα βασικό μειονέκτημα: αν η συσκευή D χρειάζεται πρόσβαση σε πόρους των υπολογιστών A, B και C (βλέπε σχήμα), πρέπει να γνωρίζει ορισμένα στοιχεία για καθέναν από αυτούς, όπως είναι η διεύθυνση δικτύου. Οπότε, εφόσον ο ιδιοκτήτης της συσκευής D αποφασίσει να δράσει κακόβουλα, γνωρίζει τις δικτυακές διευθύνσεις οπότε θα μπορούσε στο σενάριό μας να ανιχνέυσει τους A, B, C για πιθανές τρύπες ασφαλείας, και να αποκτήσει παραπάνω προσβάσεις από αυτές που έπρεπε να έχει, ή να προκαλέσει προβλήματα στην λειτουργία τους.



Σχήμα 6: Απευθείας επικοινωνία

Έτσι, προκύπτουν αυτόματα τα προβλήματα της αποκεντρωμένης ασφάλειας που έχουν αναφερθεί παραπάνω. Μία λύση στο πρόβλημα αυτό είναι να δημιουργηθεί ένας ενδιάμεσος κόμβος, που σημειώνεται με RAST στο επόμενο σχήμα. Κατόπιν αυτής της αλλαγής, ένα παράδειγμα λειτουργίας θα είναι το εξής:

- Οι A, B και C παρέχουν πρόσβαση στους πόρους που θέλουν να διαθέσουν στο RAST.
- Ο D αποκτά πρόσβαση στο RAST.
- Τώρα ο D βλέπει όλους τους πόρους που δίνουν οι A, B, C μόλις συνδεθεί στο RAST.
- Έστω τώρα ότι ο D θέλει να χρησιμοποιήσει κάποιον πόρο του A.
- Ο D δεν γνωρίζει ότι το RAST έχει πρόσβαση στους A, B, C, ότι είναι 3, ή οποιαδήποτε άλλη πληροφορία. Σε ό,τι τον αφορά, οι πόροι φαίνονται να ανήκουν στο RAST.
- Κάνει μια αίτηση για πόρο του A στο RAST.
- Το RAST προωθεί την αίτηση στον A για λογαριασμό του.
- Ο A δίνει την απάντηση.
- Το RAST προωθεί την απάντηση στον D που την αιτήθηκε.



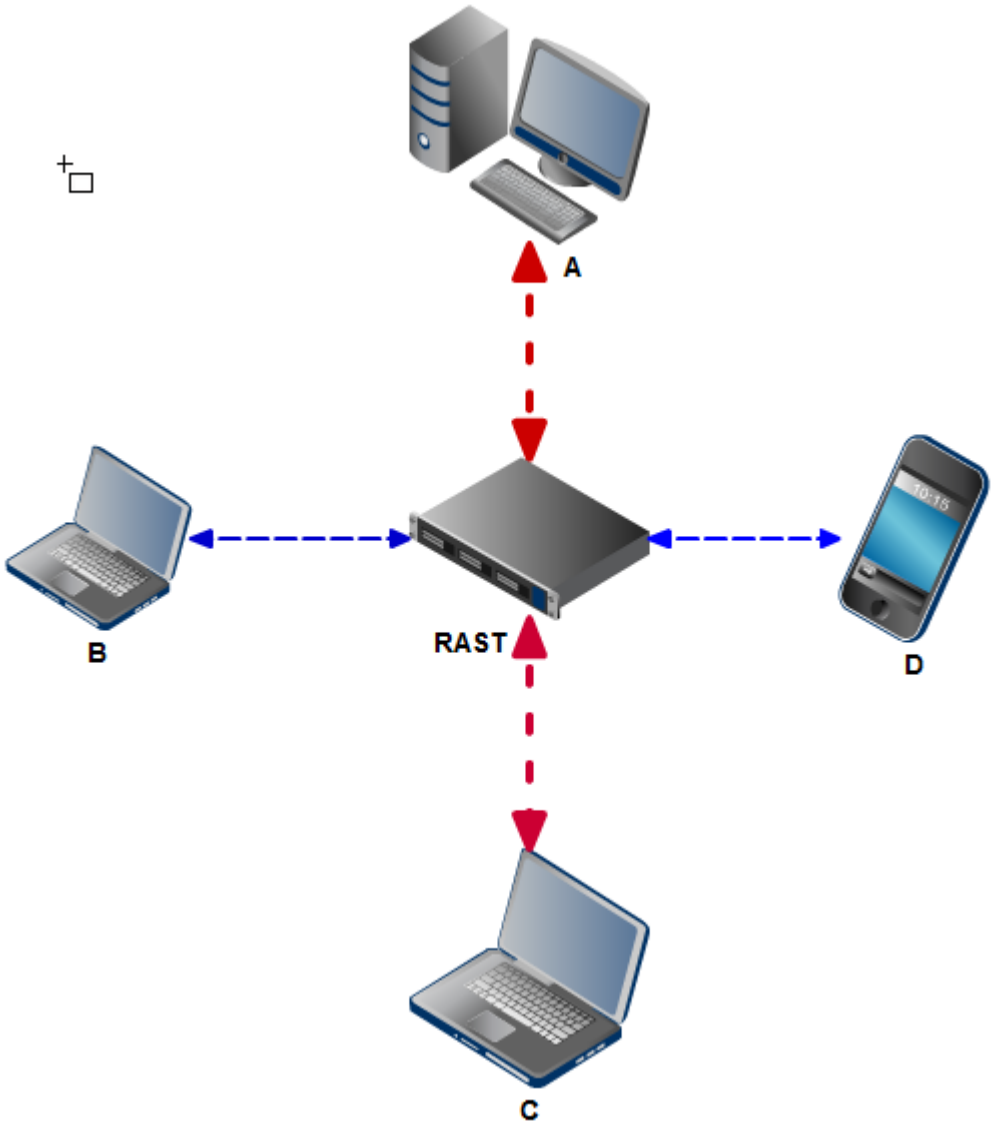
Σχήμα 7: Με ενδιάμεσο

Με αυτόν τον τρόπο, εφόσον εξασφαλιστεί η ασφάλεια του RAST, εξασφαλίζεται αυτόματα η ασφάλεια των A, B και C. Επιπλέον, εφόσον ο D δεν μπορεί να αναγνωρίσει πόσοι και ποιοι είναι οι πάροχοι των πόρων, εξασφαλίζεται και η ανωνυμία του συστήματος. Άρα, εφόσον το RAST είναι έμπιστο, η ασφάλεια και η ανωνυμία του συστήματος εξασφαλίζονται.

Η δυνατότητα αυτή της αυτόματης προώθησης αιτήσεων και απαντήσεων ονομάζεται στη διεθνή βιβλιογραφία tunneling, και το RAST ονομάζεται tunnel, όρους που θα χρησιμοποιήσουμε από αυτό το σημείο και στη συνέχεια της εργασίας αυτής, εφόσον δεν υπάρχει ο αντίστοιχος ελληνικός.

Μια ενδιαφέρουσα δυνατότητα που προσφέρεται πλέον, όπως φαίνεται και στο επόμενο σχήμα, είναι η εξής:

- Στην περίπτωση αυτή, όλοι οι εμπλεκόμενοι, δηλαδή ο A, B, C, D ταυτόχρονα παρέχουν αλλά και καταναλώνουν πόρους.
- Τώρα οι A, B, C, D παρέχουν στο RAST πρόσβαση στους πόρους που διαθέτουν.
- Όπως φαίνεται λοιπόν στο σχήμα, ο D χρησιμοποιεί κάποιον πόρο του B (φαίνεται στο σχήμα με την αραιή διακεκομμένη διαγράμμιση), ενώ ταυτόχρονα ο C χρησιμοποιεί κάποιον πόρο του A (φαίνεται στο σχήμα με την πυκνή διακεκομμένη διαγράμμιση).
- Η μόνη γνώση που έχουν όλοι οι συμμετέχοντες στο σενάριο αυτό είναι οι πόροι που ο καθένας έχει παραχωρήσει στο σύστημα.
- Έτσι, ο D και ο C δεν μπορούν να μάθουν την πηγή του πόρου που αξιοποίησαν. Σε ό,τι τους αφορά θα μπορούσαν να έχουν κάνει «ανταλλαγή» πόρων, χωρίς να έχουν γνώση αυτού του γεγονότος.



Σχήμα 8: Παράδειγμα χρήσης

3

Απαιτήσεις συστήματος / Ανάγκες

Στην εισαγωγή συζητήθηκαν εκτενώς οι εξελίξεις και τα ζητήματα που προκύπτουν και διαμορφώνουν ανάγκες για συστήματα όμοια με αυτό που αναπτύχθηκε στα πλαίσια της εργασίας αυτής. Στο παρόν κεφάλαιο θα αναφερθούν οι συγκεκριμένες απαιτήσεις που είχαμε από το σύστημα.

3.1 Δικτυακή πρόσβαση σε πόρους

Το βασικό αντικείμενο του συστήματος είναι η διαμοίραση πόρων. Οπότε το σύστημα θα πρέπει να μπορεί να παραχωρήσει πρόσβαση σε πόρο που παρέχει στο σύστημα ένας χρήστης Α, τον οποίο πόρο θα πρέπει να μπορεί να καταναλώσει ένας χρήστης Β. Επιπλέον, το σύστημα θα πρέπει να είναι σε θέση να εξυπηρετήσει έναν σημαντικό αριθμό χρηστών, οι οποίοι ταυτόχρονα θα αξιοποιούν τις δυνατότητές του.

Σαν πόροι που μπορούν να διαμοιραστούν από το σύστημα επιλέχθηκαν η διαμοίραση φακέλων και αρχείων, καθώς και η διαμοίραση εκτυπωτών. Κι αυτό διότι αυτοί οι δυο τύποι πόρων είναι από τους συνηθέστερους που προκύπτουν σαν ανάγκη διαμοιρασμού. Θα μπορούσε ενδεχομένως να προστεθεί ο διαμοιρασμός δυνατότητας πρόσβασης στο διαδίκτυο, αλλά αυτό είναι ένα πεδίο που έχει καλυφθεί επαρκώς με λύσεις διαθέσιμες στην αγορά.

3.2 Ασφάλεια

Για να μπορέσει να λειτουργήσει ένα τέτοιο σύστημα σε κάποια εκτεταμένη εγκατάσταση, είναι ανάγκη να εξασφαλίζονται τόσο οι πάροχοι πόρων στο σύστημα, όσο και οι καταναλωτές, από πιθανές κακόβουλες ενέργειες άλλων χρηστών του συστήματος. Για το λόγο αυτό το σύστημα θα πρέπει να μπορεί να χειρίζεται τα δεδομένα του συστήματος καθώς και τους πόρους που διαμοιράζονται με έναν ικανοποιητικό βαθμό ασφαλείας, χωρίς να αναμένεται ότι ο κάθε χρήστης που μετέχει του συστήματος έχει εξασφαλίσει επαρκώς το δικό του σύστημα.

Αν για παράδειγμα ένας χρήστης Α θέλει να προσφέρει πρόσβαση σε έναν χρήστη Β σε αρχείο που περιλαμβάνει πνευματική ιδιοκτησία του Α, τότε είναι προφανές πως είναι απαγορευτικό για το σύστημα ένας χρήστης Γ να καταφέρει να αποκτήσει πρόσβαση στο αρχείο του Α.

3.3 Ανωνυμία

Είναι σημαντικό για ένα σύστημα που φιλοδοξεί να αξιοποιηθεί από πολλούς χρήστες, και σε μεγάλες εγκαταστάσεις, είναι σημαντικό να εξασφαλίζει την ανωνυμία των συμμετεχόντων. Κι αυτό γιατί η εμπιστοσύνη που προσφέρεται από κάποιον πάροχο πόρου για έναν συγκεκριμένο πόρο δεν σημαίνει ότι ο πάροχος αυτός θέλει να αναγνωριστεί και εντοπιστεί. Άρα, ένα σύστημα το οποίο στοχεύει στην μεγιστοποίηση των πόρων που μοιράζονται μέσα από αυτό, πρέπει να εξασφαλίζει την ανωνυμία σε όλους τους πάροχους, αλλά και καταναλωτές των πόρων που τα διαμοιράζονται.

Βέβαια, το σύστημα πρέπει ταυτόχρονα να είναι σε θέση να εντοπίζει πιθανές κακόβουλες ενέργειες, και για αυτό το λόγο εσωτερικά χρειάζεται κάποιον μηχανισμό αποτύπωσης του ιστορικού όλων των λειτουργιών που έχουν εκτελεστεί. Για το λόγο αυτό θα θεωρηθεί ότι όλοι οι συμμετέχοντες εμπιστεύονται το σύστημα και τον διαχειριστή του. Θα μπορούσε να μην προχωράει το ίδιο το σύστημα σε καμία καταγραφή, αλλά κάτι τέτοιο θα είχε αρνητικές επιπτώσεις στην ασφάλεια του συστήματος, άρα και όλων των συμμετεχόντων, και συνεπώς ακόμα και στα προσωπικά τους δεδομένα.

3.4 Συμβατότητα με τους τελικούς χρήστες

Σύμφωνα και με τις εξελίξεις που παρουσιάστηκαν στην εισαγωγή αυτής της εργασίας, οι χρήστες πλέον διαθέτουν πληθώρα ψηφιακών συσκευών που είναι σε θέση να καταναλώσουν πόρους. Για το λόγο αυτό, ένα σύστημα που διαχειρίζεται τον διαμοιρασμό των πόρων πρέπει ιδανικά να είναι συμβατό με όλες τις πιθανές συσκευές που μπορεί να έχει ένας χρήστης. Σε κάθε άλλη περίπτωση θα μένουν χρήστες εκτός του συστήματος, μειώνοντας έτσι την ζήτηση σε πόρους, άρα μειώνοντας το ενδιαφέρον για τους πάροχους να διαμοιράζουν, και ταυτόχρονα μειώνοντας και τις δυνατότητες παροχής του μέγιστου αριθμού πόρων.

Για την επίτευξη του σκοπού αυτού είναι θεμιτό να γίνει χρήση ανοιχτών και ευρέως διαδεδομένων πλατφόρμων επικοινωνίας και διασύνδεσης, με το μέγιστο δυνατό βαθμό συμβατότητας με τις ψηφιακές συσκευές της αγοράς.

3.5 Φιλικότητα

Για την μεγιστοποίηση της χρησιμότητας του συστήματος και του κάθε συστήματος που κυκλοφορεί στην αγορά, διαδραματίζει κεντρικό ρόλο η φιλικότητα προς τους χρήστες που θα το χρησιμοποιήσουν. Αν για οποιαδήποτε διαδικασία ένας χρήστης πρέπει να προβεί με μεγάλο αριθμό βημάτων, ή να πρέπει να ασχοληθεί διαβάζοντας μακροσκελείς τεκμηριώσεις της κάθε ρύθμισης, τότε μειώνεται σε μεγάλο βαθμό η πιθανότητα αξιοποίησης, και σε ορισμένους χρήστες που έχουν περιορισμένες γνώσεις σε θέματα επικοινωνιών και πληροφορικής, στερείται ακόμα και η δυνατότητα.

Για το λόγο αυτό ένα σύστημα πρέπει να απαιτεί τις ελάχιστες δυνατές ρυθμίσεις, και στα σημεία που μπορούν να εξειδικευτούν, να έχει προεπιλεγμένες τις πιο συνηθισμένες ρυθμίσεις για τυπική χρήση. Με αυτόν τον τρόπο ελαχιστοποιούνται οι νέες γνώσεις που απαιτούνται από τον χρήστη, και αυξάνεται η δυνατότητα αξιοποίησης.

3.6 Ευκολία στην εγκατάσταση και διαχείριση

Πέραν από την ευκολία για τους τελικούς χρήστες, το σύστημα πρέπει να είναι φιλικό και όσο το δυνατόν απλούστερο για τον διαχειριστή του. Ο διαχειριστής

θα είναι αυτός που θα κληθεί να το ρυθμίσει έτσι ώστε να είναι ασφαλέστερο και πρακτικότερο για τους τελικούς χρήστες.

Για το λόγο αυτό, η πρώτη μέριμνα είναι να είναι ένα σύστημα που να έχει τον ελάχιστο χρόνο και τα ελάχιστα βήματα πρώτης εγκατάστασης σε κάποιο χώρο ή δίκτυο. Η εγκατάσταση πρέπει να είναι σύντομη και σαφής.

Ταυτόχρονα, πρέπει να είναι εύκολο στις αλλαγές ρυθμίσεων, και να παρουσιάζει όλα τα απαραίτητα για τον διαχειριστή εργαλεία με επαρκή και φιλική διεπαφή (interface).

Ακόμα, πρέπει να είναι εγκατεστημένο σε ανοικτές πλατφόρμες, με εκτενή τεκμηρίωση, γνωστές και ευρέως χρησιμοποιούμενες στην αγορά ώστε ο διαχειριστής να έχει στη διάθεσή του μεγάλο όγκο βιβλιογραφίας ώστε να λύσει τυχόν προβλήματα που προκύπτουν.

Τέλος, πρέπει να έχει αναλυτική καταγραφή όλων των συμβάντων του συστήματος, ώστε ο διαχειριστής να μπορεί αφενός να βγάλει συμπεράσματα για την καθημερινή χρήση του συστήματος, και αφετέρου να μπορεί να ανιχνεύσει πιθανές ανεπιθύμητες ενέργειες από κάποιον χρήστη, ώστε να έχει τη δυνατότητα να τον ταυτοποιήσει και να προβεί στις απαραίτητες ενέργειες προάσπισης της ακεραιότητας του συστήματος και της λειτουργίας του.

3.7 Χρονικά περιορισμένες προσβάσεις

Μια σημαντική καινοτομία που έρχεται να εισάγει το σύστημα, είναι αυτή των χρονικά περιορισμένων προσβάσεων. Δηλαδή, κάθε λογαριασμός που δημιουργείται στο σύστημα έχει συγκεκριμένο χρόνο ζωής εντός του, μετά από τον οποίο διαγράφεται από το σύστημα.

Έτσι, το σύστημα πρέπει να προσφέρει έναν μηχανισμό, στον οποίο ταυτόχρονα με τη δημιουργία ενός χρήστη, να ορίζεται και αυτομάτως ο χρόνος στον οποίο ο λογαριασμός αυτός θα πάψει να ισχύει. Ταυτόχρονα, με το πέρας αυτού του χρόνου, το σύστημα πρέπει να διαγράφει τον λογαριασμό αυτόν, και να αφαιρεί κάθε δικαίωμα πρόσβασης από τον χρήστη.

3.8 Δυνατότητα περιορισμών ανάλογα με τις περιστάσεις

Το σύστημα θα πρέπει να είναι σε θέση να υποστηρίξει περιορισμούς βάση των ισχύων περιστάσεων (context aware), όπως περιορισμούς ανάλογα με την τοποθεσία του χρήστη, την ώρα της ημέρας και άλλες περιστάσεις που πιθανώς να συντρέχουν, επιτρέποντας ή αναιρώντας τις προσβάσεις αντίστοιχα των αναγκών που προκύπτουν.

Επειδή όμως η φύση του συστήματος είναι ανώνυμη, ορισμένες σχετικές πληροφορίες δεν του είναι διαθέσιμες. Για το λόγο αυτό πρέπει να είναι συμβατό με άλλα συστήματα σχετικών περιορισμών.

3.9 Επεκτασιμότητα

Μιας και το πεδίο εφαρμογής του τομέα της συνεργασίας πάνω από δίκτυα επικοινωνιών είναι τόσο ευρύ, γίνεται προφανές ότι πολλές εγκαταστάσεις θα συναντήσουν εξειδικευμένες ανάγκες και απαιτήσεις, που θα πρέπει να καλυφθούν, ώστε το σύστημα να εξυπηρετήσει επιτυχώς την περίσταση.

Για το λόγο αυτό, το σύστημα πρέπει να είναι φτιαγμένο κατά τέτοιο τρόπο, ώστε να μπορεί να επεκταθεί για να καλύψει λειτουργικότητες που πιθανώς να είναι αναγκαίες για κάποιες περιστάσεις. Έτσι, το σύστημα πρέπει να είναι προγραμματισμένο πάνω σε πλατφόρμες που υποστηρίζουν όλες τις εξελίξεις στους τομείς της πληροφορικής και των επικοινωνιών, εύκολες στον προγραμματισμό και ευρέως διαδεδομένες. Ταυτόχρονα θα είναι θετικό να είναι βασισμένο σε ανοιχτές και ελεύθερες πλατφόρμες, ώστε πέραν του κόστους ανάπτυξης κάποιας επέκτασης, να μην υπάρχουν περεταίρω κόστη για αγορές αδειών.

4

Γενική Περιγραφή του συστήματος Resource Access Security Tunnel (RAST)

Στο κεφάλαιο αυτό θα παρουσιαστεί το σύστημα RAST που υλοποιήθηκε για τις ανάγκες της παρούσας εργασίας. Πρώτα παρατίθενται διάφοροι ορισμοί για λεξιλόγιο που θα χρησιμοποιηθεί στη συνέχεια, έπειτα παρουσιάζονται οι διάφοροι ρόλοι των χρηστών εντός του συστήματος. Τέλος παρουσιάζονται οι πόροι που μπορούν να μοιραστούν στο σύστημα. Μετά από αυτές τις εισαγωγικές ενότητες παρουσιάζεται αναλυτικότερα το σύστημα.

4.1 Ορισμοί

Στην ενότητα αυτή παρουσιάζονται συνοπτικά ορισμοί για έννοιες που θα χρησιμοποιηθούν εκτενώς στην περιγραφή του συστήματος RAST.

- RAST: Το σύστημα που υλοποιήθηκε, αλλά σε ορισμένες περιπτώσεις και το σύνολο του συστήματος και του υπολογιστή που το εκτελεί, για συντομία.
- Χρήστες: Τα άτομα που έχουν πρόσβαση στο RAST, και μπορούν είτε να διαθέσουν πόρους στο σύστημα, είτε να χρησιμοποιήσουν πόρους.
- Διαχειριστής: Ο διαχειριστής του RAST
- Δικαίωμα χρήσης: Το δικαίωμα που έχει κάθε χρήστης να χρησιμοποιήσει έναν πόρο.

- Client: ο υπολογιστής ή ψηφιακή συσκευή που έχει πρόσβαση στο σύστημα, είτε για να μοιράσει πόρους, είτε για να καταναλώσει.
- Driver: ειδικό πρόγραμμα που υποστηρίζει την δυνατότητα της εκτύπωσης, μοναδικό για κάθε εκτυπωτή.
- Πάροχος: Αυτός που διαθέτει προς χρήση κάποιον πόρο στο σύστημα (provider).
- Καταναλωτής: Αυτός που απολαμβάνει πρόσβαση σε έναν πόρο.

4.2 Ρόλοι των χρηστών στο σύστημα

Στην ενότητα αυτή παρουσιάζονται οι ρόλοι των διαφόρων χρηστών στο σύστημα. Οι χρήστες του συστήματος μπορούν να αναλάβουν δύο ρόλους: είτε να παρέχουν πόρους στο σύστημα, είτε να χρησιμοποιούν πόρους από το σύστημα. Το RAST δεν αποκλείει έναν χρήστη από την ταυτόχρονη ανάληψη και των δυο ρόλων. Απαιτείται πάντα από τον κάθε χρήστη για να μπορέσει να συνδεθεί στο σύστημα, να δώσει όνομα χρήστη και κωδικό πρόσβασης, το οποίο αποκτά μόνο από τον διαχειριστή του συστήματος. Παρακάτω παρουσιάζονται αναλυτικά οι ρόλοι καθώς και η λειτουργικότητα που απολαμβάνει ο καθένας.

4.2.1 Ρόλος Παρόχου

Ο χρήστης που επιλέγει αυτόν τον ρόλο έχει την δυνατότητα μέσω μιας διαδικτυακής εφαρμογής με διπροσωπία ιστοσελίδων (web interface) που βρίσκεται σε προκαθορισμένη από τον διαχειριστή διεύθυνση να συνδεθεί, αφού δώσει το όνομα χρήστη και κωδικό πρόσβασης[2], και να διαθέσει πόρους στο Security Tunnel που θα μπορούσαν να χρησιμοποιηθούν από χρήστες που έχουν το ρόλο καταναλωτή πόρων. Ο χρήστης σε αυτήν την περίπτωση:

- Επιλέγει τους πόρους που επιθυμεί να είναι διαθέσιμοι στους χρήστες.
- Επιλέγει σε ποιους χρήστες θα δώσει πρόσβαση στους αντίστοιχους πόρους.

[2] Στην παρούσα υλοποίηση η ταυτοποίηση του χρήστη δεν έχει υλοποιηθεί, γιατί δόθηκε μεγαλύτερο βάρος στην καινοτόμα λειτουργικότητα του συστήματος, και όχι σε ένα σύστημα ταυτοποίησης, όμοια του οποίου συναντώνται ευρέως στην βιβλιογραφία και το διαδίκτυο.

4.2.2 Ρόλος Καταναλωτή

Ο χρήστης που αναλαμβάνει αυτόν τον ρόλο έχει την δυνατότητα να συνδεθεί με το RAST είτε πληκτρολογώντας κατάλληλα την διεύθυνση του μηχανήματος που εκτελεί το RAST[3], είτε μεταφορτώνοντας ειδικό αρχείο σεναρίου (script) αν βρίσκεται σε συμβατό περιβάλλον. Θα του ζητηθεί να δώσει το όνομα χρήστη και κωδικό πρόσβασης και ανάλογα με τα δικαιώματα πρόσβασης που έχουν καθοριστεί ειδικά για αυτόν να καταναλώσει πόρους. Η πρόσβασή του αυτή είναι προσωρινού χαρακτήρα και μετά το τέλος του διαθέσιμου χρόνου δεν θα έχει κανένα δικαίωμα χρήσης των πόρων. Έτσι, ο χρήστης σε αυτή την περίπτωση:

- Συνδέεται στο σύστημα από το οποίο ενημερώνεται για τους πόρους στους οποίους έχει πρόσβαση.
- Μπορεί να αξιοποιήσει οποιονδήποτε πόρο στον οποίο του έχει παραχωρηθεί σχετικό δικαίωμα.

4.2.3 Ο διαχειριστής του συστήματος

Ο διαχειριστής του συστήματος είναι ένας ειδικός λογαριασμός, ο οποίος διαχειρίζεται συνολικά το σύστημα. Η βασική του ιδιότητα είναι να διαχειρίζεται το σύνολο των χρηστών, προσθέτοντας, αφαιρώντας και μεταβάλλοντας τα στοιχεία των χρηστών. Επιπλέον είναι ο χρήστης που του έχει ανατεθεί η ευθύνη να εγκαθιστά, συντηρεί και εξασφαλίζει την απρόσκοπτη λειτουργία του συστήματος. Έτσι, ο διαχειριστής:

- Επιλέγει σε ποιους χρήστες θα επιτρέψει την πρόσβαση στο Security Tunnel.
- Επιλέγει σε ποιους χρήστες θα δώσει πρόσβαση στους κοινούς από το σύστημα πόρους μόνο όταν απαιτείται από τις περιστάσεις.
- Παρέχει τα ονόματα χρήστη και τους κωδικούς πρόσβασης για όλους τους χρήστες που θα συμμετέχουν στο σύστημα.
- Καθορίζει το χρονικό διάστημα για το οποίο έχουν ισχύ τα στοιχεία πρόσβασης του κάθε χρήστη ή ομάδας χρηστών.
- Διαχειρίζεται και συντηρεί το RAST, εξασφαλίζοντας την απρόσκοπτη και απροβλημάτιστη λειτουργία του.

[3] Κατάλληλα διαμορφωμένη σύμφωνα με το URI του πρωτοκόλλου SMB/CIFS που χρησιμοποιήθηκε για την εφαρμογή. Πληροφορίες για αυτήν την διαμόρφωση παρουσιάζονται στο εγχειρίδιο χρήσης του συστήματος.

- Διαχειρίζεται και συντηρεί τόσο το υλικό του υπολογιστή που εκτελεί το RAST, όσο και το λειτουργικό σύστημα στο οποίο βασίζεται το RAST για τυχόν ενημερώσεις ασφαλείας ή βελτιώσεις επιδόσεων.

4.3 Διαμοιραζόμενοι πόροι

Στο σύστημα Security Tunnel διαμοιράζονται οι εξής πόροι:

- Αρχεία και φάκελοι
- Εκτυπωτές

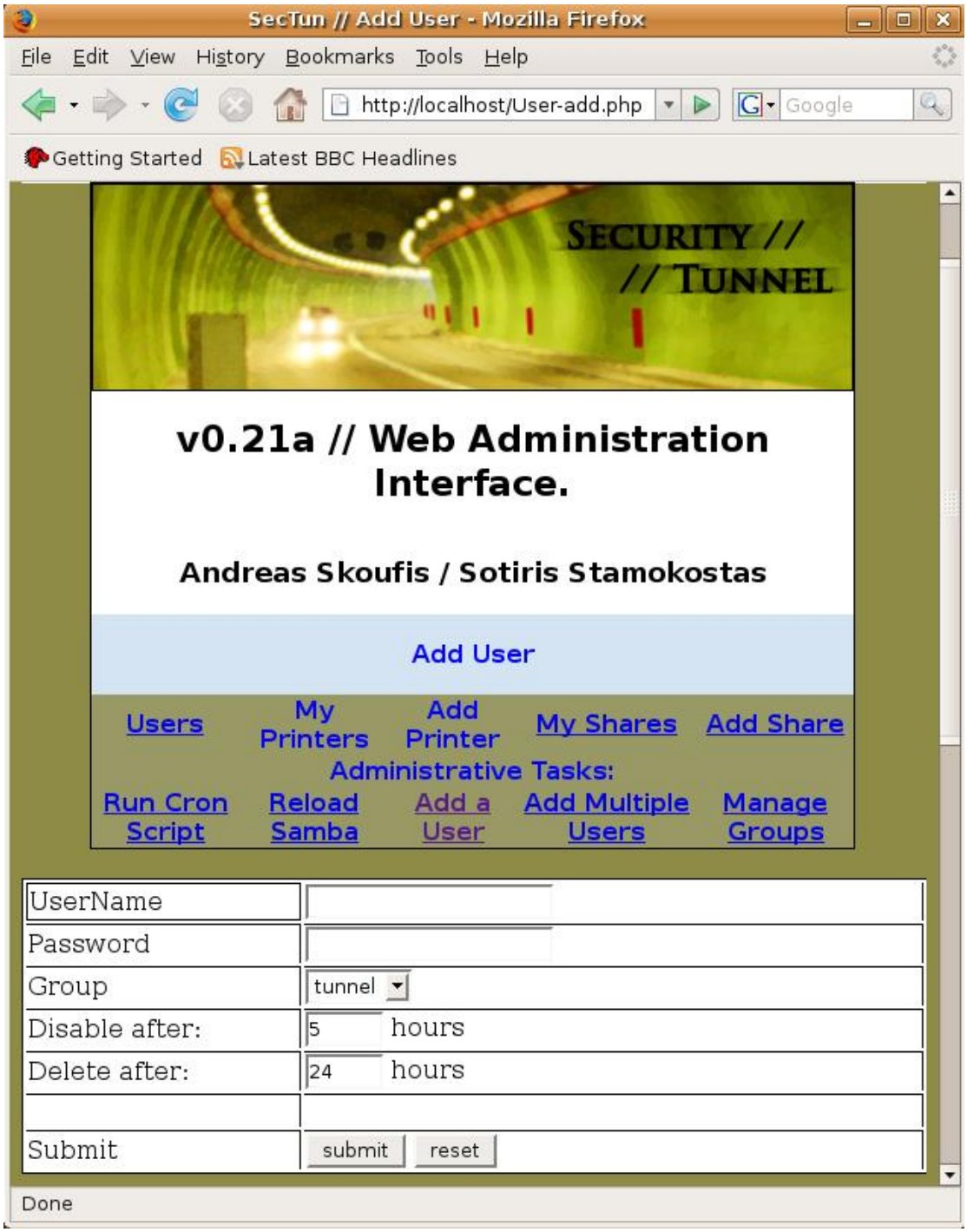
Και οι δυο ανωτέρω πόροι διαμοιράζονται μέσω του πρωτοκόλλου SMB/CIFS [17], και τα δικαιώματα καθώς και οι περιορισμοί στην πρόσβαση υποστηρίζονται από το ίδιο το πρωτόκολλο.

Πιο συγκεκριμένα για τα αρχεία, ο πάροχος επιλέγει κάποιον φάκελο τον οποίο διαμοιράζει στο σύστημα, ο οποίος φάκελος μπορεί να περιέχει αρχεία, ή και υποφακέλους. Σε αυτόν τον φάκελο, εφόσον έχει το αντίστοιχο δικαίωμα, μπορεί να συνδεθεί ο καταναλωτής και να χρησιμοποιήσει τα αρχεία που προσφέρονται. Ο πάροχος επιπλέον μπορεί να θέσει τον φάκελο ως μόνο για ανάγνωση (read only), πράγμα που σημαίνει ότι ο καταναλωτής μπορεί μόνο να αναγνώσει τα αρχεία, αλλά όχι να τα διαγράψει, να προσθέσει κάποιο αρχείο ή να μεταβάλλει κάποιο υφιστάμενο. Στην αντίθετη περίπτωση, μπορεί να δώσει τον πλήρη έλεγχο του φακέλου στον καταναλωτή, οπότε αυτός να μπορεί να προχωρήσει σε πράξεις όπως η διαγραφή κάποιου αρχείου.

Σε ό,τι αφορά τους εκτυπωτές, οι επιλογές είναι περιορισμένες, καθώς ο πάροχος επιλέγει μοναχά τους χρήστες με δικαίωμα εκτύπωσης. Κάθε δικαίωμα που δίνεται από τον πάροχο μπορεί να αρθεί οποιαδήποτε στιγμή, είτε με σχετική εντολή του πάροχου, είτε κατόπιν σχετικής ρύθμισης από τον διαχειριστή.

4.4 Περιγραφή του συστήματος

Το RAST είναι ένα σύστημα ασφάλειας το οποίο διευκολύνει στον διαμοιρασμό πόρων στο περιβάλλον ενός δικτύου. Το σύστημα εγκαθίσταται σε έναν υπολογιστή (από εδώ και πέρα θα τον αναφέρουμε και αυτόν ως RAST), ο οποίος μόλις συνδεθεί στο δίκτυο ξεκινά και παρέχει τις υπηρεσίες που έχουν υλοποιηθεί.



Σχήμα 9: Οθόνη του συστήματος RAST

Συγκεκριμένα, ο διαχειριστής του συστήματος αρχικά ορίζει όλους τους χρήστες που επιθυμεί να έχουν πρόσβαση στο σύστημα, παρέχοντας στο σύστημα τα ονόματα χρήστη, τους αντίστοιχους κωδικούς πρόσβασης, και έπειτα παραδίδει αυτά τα στοιχεία ταυτοποίησης στους ενδιαφερόμενους χρήστες. Επιπλέον ο διαχειριστής μπορεί να ορίσει πόρους που διαθέτει το μηχάνημα που τρέχει το RAST, ή πόρους που βρίσκονται στο δίκτυο εκτός των προσωπικών των χρηστών του συστήματος.

Έπειτα, με τα στοιχεία που τους έχει δώσει ο διαχειριστής, μπορούν πλέον να συνδεθούν στο διαδικτυακό περιβάλλον οι χρήστες του συστήματος, ώστε να εισάγουν πόρους προς διάθεση στο σύστημα. Εκεί, εφόσον έχουν επιτρέψει το διαμοιρασμό των πόρων πρώτα από την συσκευή ή τον υπολογιστή τους, εισάγουν τα απαραίτητα στοιχεία σε σχετική φόρμα, επιλέγοντας μια ονομασία για τον πόρο που επιθυμούν να διαθέσουν στο σύστημα, τους χρήστες στους οποίους επιτρέπουν την πρόσβαση (μέσω κατάλληλης επιλογής μπορούν να επιλέξουν να επιτρέψουν την πρόσβαση σε όλους τους χρήστες της εφαρμογής). Προσθέτουν όποιο σχόλιο έχουν για τον διαμοιραζόμενο πόρο, επιλέγοντας ταυτόχρονα, στην περίπτωση φακέλων, αν ο χρήστης μπορεί να μεταβάλλει τα περιεχόμενα του φακέλου, ή μπορεί μόνο να τα αναγνώσει (read only access). Τέλος, συμπληρώνουν τα απαραίτητα για την πρόσβαση από το RAST στοιχεία του πόρου που θέλουν να μοιράσουν, έτσι ώστε το RAST να αποκτήσει πρόσβαση στον συγκεκριμένο πόρο, για να είναι σε θέση έπειτα να τον μοιράσει στους υπόλοιπους χρήστες του συστήματος.

Εφόσον λοιπόν ένας χρήστης έχει λογαριασμό στο RAST, μπορεί πλέον να συνδεθεί στο σύστημα μέσω του πρωτοκόλλου SMB, χρησιμοποιώντας τα στοιχεία ταυτοποίησης (όνομα χρήστη, κωδικός πρόσβασης) που του έχουν δοθεί από τον διαχειριστή του συστήματος. Εκεί, θα του παρουσιαστούν όλοι οι φάκελοι και οι εκτυπωτές στους οποίους έχει πρόσβαση. Για να συνδεθεί με οποιονδήποτε πόρο αρκεί να έχει το αντίστοιχο δικαίωμα από τον πάροχο του πόρου. Οι προσβάσεις έχουν χαρακτήρα προσωρινό, οπότε με το πέρας της ισχύς του λογαριασμού αναιρούνται αυτόματα. Επιπλέον, το σύστημα εξασφαλίζει την ανωνυμία του πάροχου, εφόσον όλοι οι πόροι φαίνονται στον καταναλωτή σαν τοπικοί του RAST πόροι, χωρίς κάποια πληροφορία για τον πάροχο, την ταυτότητά του, την τοποθεσία του ή το μηχάνημα που παρέχει τους πόρους.

Με αυτές λοιπόν τις βασικές αρχές το σύστημα RAST υλοποιεί τους περιορισμούς πρόσβασης και ταυτόχρονα δίνει τις απαραίτητες δυνατότητες για παροχή πόρων στο σύστημα και ταυτόχρονη παροχή δυνατότητας κατανάλωσης των πόρων αυτών από τους χρήστες του.

4.5 Χαρακτηριστικά του συστήματος RAST

Στην ενότητα αυτή παρουσιάζονται αναλυτικότερα ορισμένα σημαντικά χαρακτηριστικά του συστήματος RAST τα οποία χρήζουν εκτενέστερης ανάλυσης.

Παρουσιάζεται εκτενέστερα η δυνατότητα του tunneling, περιγράφονται αναλυτικά οι δυνατότητες για τον διαμοιρασμό εκτυπωτών και αρχείων, παρουσιάζονται οι δυνατότητες καταγραφής και αυτόματης συντήρησης του συστήματος και τέλος αναφέρεται αναλυτικότερα ο ρόλος του διαχειριστή στην λειτουργία του συστήματος.

4.5.1 Tunneling πόρων

Το σύστημα RAST δρα ως ένας έμπιστος ενδιάμεσος κόμβος μεταξύ του παρόχου ενός πόρου, και του καταναλωτή του, όπως παρουσιάστηκε στο σχετικό κεφάλαιο της εισαγωγής. Έτσι, αφενός ο πάροχος δεν γνωρίζει τη θέση ή λοιπά χαρακτηριστικά του καταναλωτή, και αντίστοιχα αποκρύπτεται η ταυτότητα και τα λοιπά στοιχεία του παρόχου από τον καταναλωτή.

Αντί της ανταλλαγής μεταξύ αυτών στοιχείων πρόσβασης, η διαδικασία που προκύπτει γίνεται ως εξής:

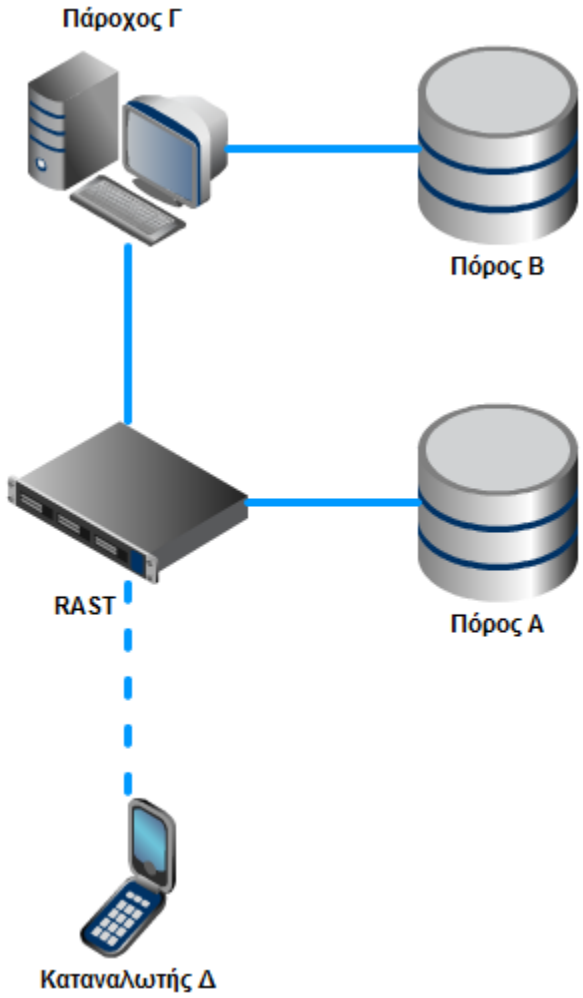
1. Ο πάροχος μπαίνει στο RAST και δίνει πρόσβαση στο σύστημα σε κάποιον πόρο που αυτός διαθέτει.
2. Ο καταναλωτής συνδέεται στον πόρο αυτό μέσω του RAST, εφόσον του έχει δοθεί τέτοιο δικαίωμα από τον πάροχο.

Εδώ αξίζει να σημειωθεί ότι ο κάθε χρήστης δεν γνωρίζει την πραγματική ταυτότητα ενός άλλου με τον οποίο συνδέεται, όπως και άλλα χαρακτηριστικά πιθανώς οδηγούσαν σε ανεπιθύμητη αποκάλυψη πληροφοριών, για παράδειγμα την διεύθυνση IP. Το μόνο που χρειάζεται ένας πάροχος να γνωρίζει είναι τα δικά του χαρακτηριστικά ταυτοποίησης στο RAST (όνομα χρήστη, κωδικός χρήστη) καθώς και τα χαρακτηριστικά (ονόματα χρήστη ή ονόματα ομάδων χρηστών) των χρηστών-καταναλωτών των πόρων που θέλει να παραχωρήσει για πρόσβαση. Αυτά τα χαρακτηριστικά δεν είναι δυνατόν να ταυτοποιήσουν τον κάθε χρήστη, διατηρώντας έτσι την ανωνυμία του συστήματος. Έτσι, όλη η επικοινωνία μεταξύ παρόχου και καταναλωτή συμβαίνει μέσω του συστήματος RAST χωρίς απευθείας επικοινωνία παρόχου-καταναλωτή, που θα μπορούσε να αποκαλύψει την ταυτότητα του ενός στον άλλο.

Η επικοινωνία γίνεται με τρόπο διαφανή στο σύστημα, και όχι με προσωρινή αποθήκευση του πόρου στο RAST. Δηλαδή κάθε αίτηση του καταναλωτή προωθείται αυτομάτως στον πάροχο, και αντίστροφα. Έτσι επιτυγχάνεται το tunneling.

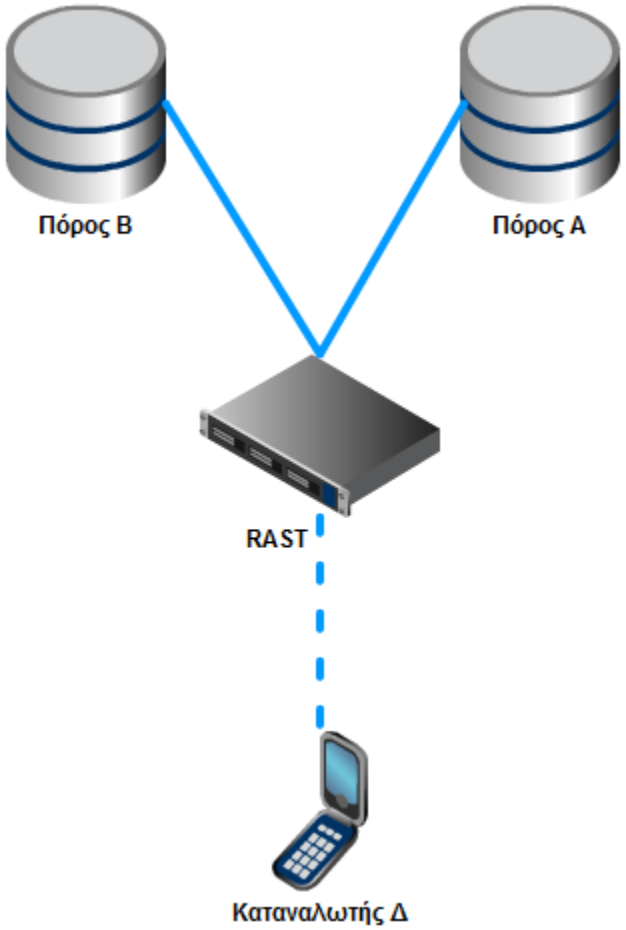
Για καλύτερη κατανόηση του tunneling, και των επιπτώσεων που έχει στην ανωνυμία του συστήματος, παρατίθεται το παρακάτω παράδειγμα:

Στην παρούσα υλοποίηση, το RAST διαμοιράζεται έναν δικό του πόρο Α, και ένας χρήστης, ο Πάροχος Γ, διαμοιράζει μέσω του συστήματος τον πόρο Β. Τέλος, για το παράδειγμά μας, ένας ακόμα χρήστης, ο καταναλωτής Δ, συνδέεται και καταναλώνει τους πόρους Α και Β. Η αρχιτεκτονική αυτή φαίνεται στο επόμενο σχήμα:



Σχήμα 10: Η πραγματική εικόνα της διάθεσης των πόρων

Όμως, ο καταναλωτής Δ δεν μπορεί να αντιληφθεί την προέλευση των πόρων. Έτσι, σε αυτόν φαίνεται σαν οι πόροι να ανήκαν όλοι στο RAST, όπως φαίνεται στο παρακάτω σχήμα:



Σχήμα 11: Ο τρόπος με τον οποίο αντιλαμβάνεται τον διαμοιρασμό ο καταναλωτής

Με αυτό τον τρόπο αποκρύπτεται η ταυτότητα και οποιοδήποτε άλλο αναγνωριστικό στοιχείο του πάροχου Γ από τον καταναλωτή Δ. Φυσικά, δεν μπορεί να γίνει έλεγχος από το σύστημα αν ο πάροχος Γ διαμοιράζει αρχεία που περιλαμβάνουν προσωπικά του δεδομένα, οπότε για το περιεχόμενο των πόρων που μοιράζει είναι αποκλειστικά υπεύθυνος για την τήρηση της ανωνυμίας του.

4.5.2 Διαμοιρασμός φακέλων και αρχείων

Για την δυνατότητα διαμοιρασμού φακέλων και αρχείων, το RAST χρησιμοποιεί το πρωτόκολλο Server Message Block ή SMB [17], που είναι και η παρεχόμενη τεχνολογία διαμοιρασμού από το λειτουργικό σύστημα Microsoft Windows, η οποία όμως είναι διαθέσιμη και στην πλειοψηφία των λειτουργικών συστημάτων που βρίσκονται αυτή τη στιγμή στην αγορά, είτε μέσω απευθείας υποστήριξης του λειτουργικού συστήματος, είτε μέσω πρόσθετης εφαρμογής που διατίθεται για το λειτουργικό αυτό.

Ο πάροχος σε πρώτη φάση κάνει διαθέσιμο κάποιον φάκελο μοιράζοντας τον στο δίκτυο με τις κατάλληλες ρυθμίσεις ασφάλειας. Έπειτα εγγράφει τον συγκεκριμένο φάκελο στο σύστημα RAST, με όλα τα απαραίτητα στοιχεία που χρειάζεται το σύστημα για να αποκτήσει πρόσβαση σε αυτόν, και ελέγχοντας την πρόσβαση που θέλει να προσφέρει στους χρήστες του RAST. Ο καταναλωτής συνδέεται στο RAST και έτσι αποκτά πρόσβαση στους παρεχόμενους σε αυτόν φακέλους, μέσω του ίδιου πρωτοκόλλου.

Καθένας από αυτούς δεν αντιλαμβάνεται την σύνδεση που έχει απευθείας με τον άλλο. Ο καταναλωτής συνδέεται στο RAST και από κει αποκτά πρόσβαση στους φακέλους, και ο πάροχος βλέπει εισερχόμενες συνδέσεις στον φάκελο που έχει μοιράσει από το RAST. Έτσι, καθίσταται αδύνατο οποιοσδήποτε από τους δυο να ταυτοποιήσει τον άλλο, εξασφαλίζοντας την ανωνυμία του συστήματος.

Μία ενδιαφέρουσα «παρενέργεια» του συγκεκριμένου πρωτοκόλλου στην προτεινόμενη υλοποίηση είναι τα δυο στρώματα ασφαλείας, ένα που παρέχεται από το σύστημα RAST και ένα δεύτερο από τον ίδιο τον πάροχο. Έτσι, ας θεωρήσουμε ότι ένας πάροχος θέλει να διαμοιράσει έναν φάκελο, με δικαιώματα μόνο για ανάγνωση. Μετατρέπει τον φάκελο αυτό στον υπολογιστή του σαν διαμοιραζόμενο, θέτοντας τον ως φάκελο μόνο ανάγνωσης (read only). Έπειτα, συνδέει τον φάκελο στο RAST, αλλά κατά λάθος δεν τον θέτει ως μόνο ανάγνωσης στην φόρμα εισαγωγής του RAST. Όμως, αν δοκιμάσει κάποιος καταναλωτής να διαγράψει κάποιο αρχείο, δεν θα του επιτραπεί, περιορισμός ο οποίος δεν τέθηκε από το RAST, αλλά από τις τοπικές ρυθμίσεις ασφαλείας του χρήστη – πάροχου. Επιπλέον, ο πάροχος μπορεί να αποφασίσει να σταματήσει τον διαμοιρασμό του συγκεκριμένου πόρου στο σύστημα. Εκτός της δυνατότητας να διαγράψει τον πόρο από το RAST, μπορεί απλά να θέσει τον φάκελό του σαν μη διαμοιραζόμενο από το τοπικό του λειτουργικό. Έτσι, παρότι ο πόρος δεν έχει διαγραφεί από το RAST, κανένας καταναλωτής πλέον δεν μπορεί να έχει πρόσβαση. Με τα παραδείγματα αυτά γίνεται σαφές το διπλό στρώμα ασφαλείας, και οι σχετικές δυνατότητες που δίνονται στους πάροχους για περεταίρω εξασφάλιση των πόρων τους.

4.5.3 Διαμοιρασμός εκτυπωτών

Και εδώ, το πρωτόκολλο που χρησιμοποιήθηκε είναι το SMB. Ο πάροχος σε πρώτη φάση μοιράζει στο δίκτυο τον εκτυπωτή του με τα στοιχεία ασφάλειας που επιθυμεί. Έπειτα, εγγράφει τον εκτυπωτή στο RAST μαζί με όλα τα απαραίτητα

στοιχεία που χρειάζεται, καθώς και τον τύπο του εκτυπωτή, ώστε το RAST να επιλέξει τον κατάλληλο οδηγό (driver). Έτσι, ο καταναλωτής μπορεί να συνδεθεί στο RAST και μέσω αυτού να χρησιμοποιήσει τον εκτυπωτή.

Και σε αυτήν την περίπτωση, λόγω της διαμεσολάβησης του RAST, δεν διαρρέουν στοιχεία ταυτότητας είτε για τον πάροχο και τον υπολογιστή / εκτυπωτή του, ούτε για τον καταναλωτή και τον υπολογιστή ή την συσκευή του.

4.5.4 Διαχειριστής του συστήματος

Ο διαχειριστής είναι ένας ειδικός λογαριασμός στο σύστημα RAST. Ο βασικός του ρόλος είναι να δημιουργεί και να καταργεί τους λογαριασμούς των υπόλοιπων χρηστών στο σύστημα. Έτσι, συνδέεται μέσω μιας διεπαφής ιστοσελίδων (web interface) όπου του παρέχονται οι δυνατότητες προσθήκης, επεξεργασίας και διαγραφής των χρηστών.

Για την δημιουργία χρηστών, παρέχονται δυο δυνατότητες. Η πρώτη αφορά την δημιουργία ενός μεμονωμένου χρήστη, και η δεύτερη αφορά την μαζική δημιουργία χρηστών. Η τελευταία είναι και η πιο σημαντική, γιατί είναι απαίτηση στο σύστημα να μπορεί να διαχειριστεί με ευκολία μεγάλο αριθμό χρηστών. Στην δημιουργία των χρηστών ορίζεται και ο χρόνος λήξης του λογαριασμού τους (κοινός για όλους στην περίπτωση της μαζικής δημιουργίας) ώστε να ικανοποιηθεί το αίτημα του εφήμερου χαρακτήρα των προσβάσεων που το σύστημα παρέχει.

Η διαγραφή μπορεί να είναι μεμονωμένη ή μαζική (διαγραφή πολλών χρηστών), και η επεξεργασία μπορεί να αφορά είτε στοιχεία του χρήστη, είτε μεταβολή του χρόνου λήξης του λογαριασμού του.

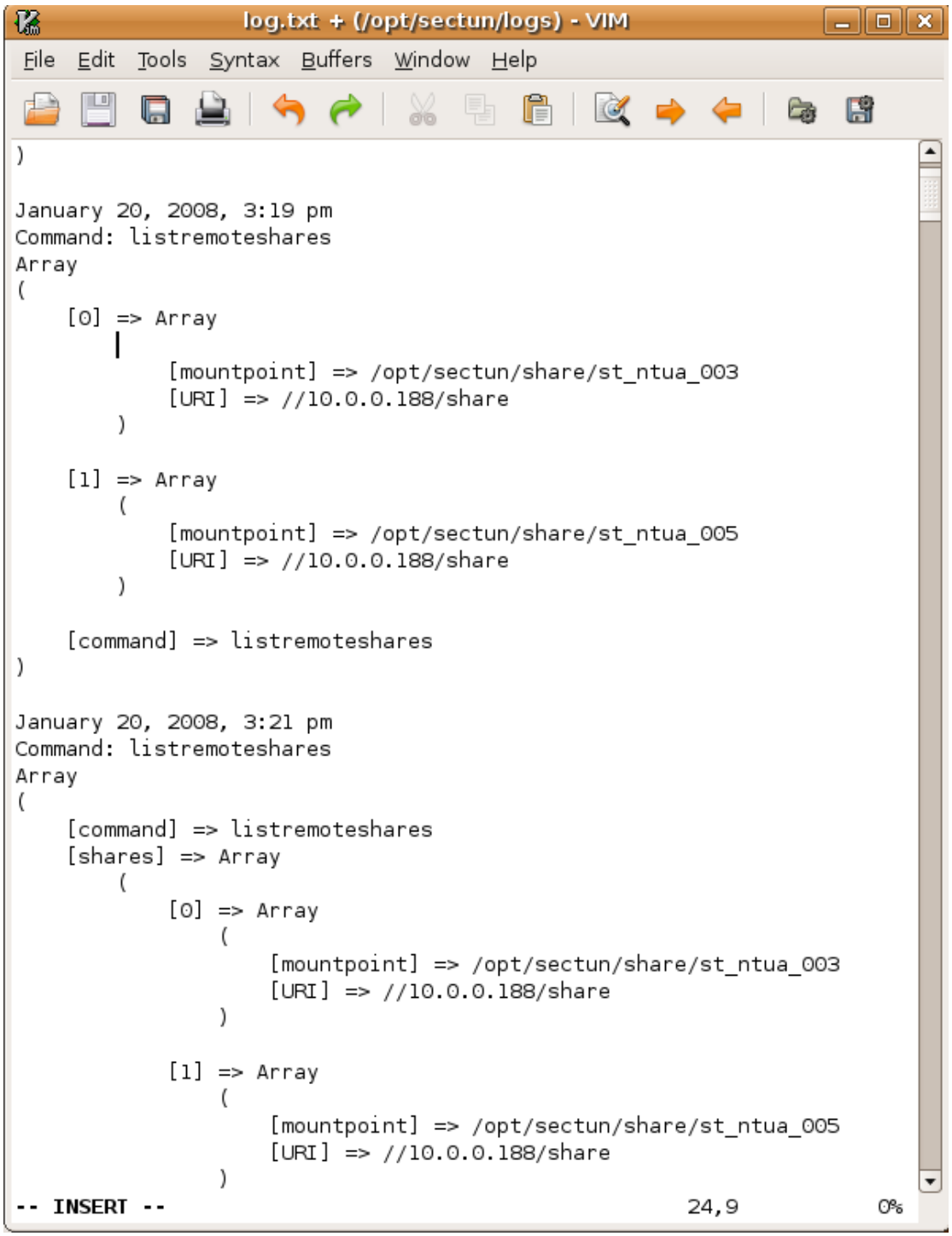
Επιπλέον, ο διαχειριστής έχει τη δυνατότητα ενεργοποίησης, απενεργοποίησης και παραμετροποίησης του συστήματος σύμφωνα με τις ανάγκες της εφαρμογής.

4.5.5 Δυνατότητα καταγραφής συμβάντων (logging)

Καθώς το σύστημα RAST είναι ένα σύστημα ασφαλείας, έχει δοθεί έμφαση στην εκτεταμένη καταγραφή κάθε κίνησης που γίνεται στο σύστημα. Έτσι, αν προκύψει οποιοδήποτε σφάλμα ή καμφθεί η ασφάλειά του, ο διαχειριστής μπορεί να ερευνήσει το περιστατικό με τη βοήθεια των καταγεγραμμένων συμβάντων, ώστε να εντοπίσει και να διορθώσει τυχόν πρόβλημα του συστήματος.

Επιπλέον, η καταγραφή μπορεί να βοηθήσει σε συνεργασία και με άλλα εργαλεία στην διαμόρφωση στατιστικών χρήσης του συστήματος αλλά και για ανάλυση της απόδοσής του (benchmarking).

Ακολουθεί ένα στιγμιότυπο του αρχείου καταγραφής συμβάντων.



Σχήμα 12: Στιγμιότυπο του αρχείου καταγραφής συμβάντων

4.5.6 Πρόγραμμα περιοδικής συντήρησης του συστήματος (cron script)

Ένα σημαντικό δομικό στοιχείο του συστήματος είναι το πρόγραμμα περιοδικής συντήρησης. Το cron script [18] ελέγχει τις βάσεις του συστήματος και προβαίνει στις αντίστοιχες ενέργειες, διατηρώντας το σύστημα στο ακέραιο.

Έτσι, αν λήξει κάποιος λογαριασμός, το cron script του αφαιρεί όλα τα δικαιώματα, διαγράφει όλους τους πόρους που είναι μοιρασμένοι μέσω αυτού του λογαριασμού στο σύστημα και τέλος διαγράφει τον ίδιο τον λογαριασμό. Ακόμα, αν κάποιος πόρος πάψει να είναι διαθέσιμος στο σύστημα (πχ. ένας πάροχος κλείσει τον υπολογιστή του), το πρόγραμμα αυτό αναλαμβάνει να τον αφαιρέσει από το σύστημα. Με αυτόν τον τρόπο εξασφαλίζεται η ακεραιότητα του συστήματος.

Το cron script τρέχει και περιοδικά (κάθε 2 λεπτά), αλλά και σε ορισμένες ειδικές περιπτώσεις. Οι περιπτώσεις αυτές είναι αφενός να το καλέσει άμεσα ο διαχειριστής του συστήματος, και επιπλέον εκτελείται αμέσως μετά από οποιαδήποτε χειροκίνητη διαγραφή χρηστών, ώστε να εξασφαλιστεί με αμεσότητα η διακοπή πρόσβασης αυτών στο σύστημα.

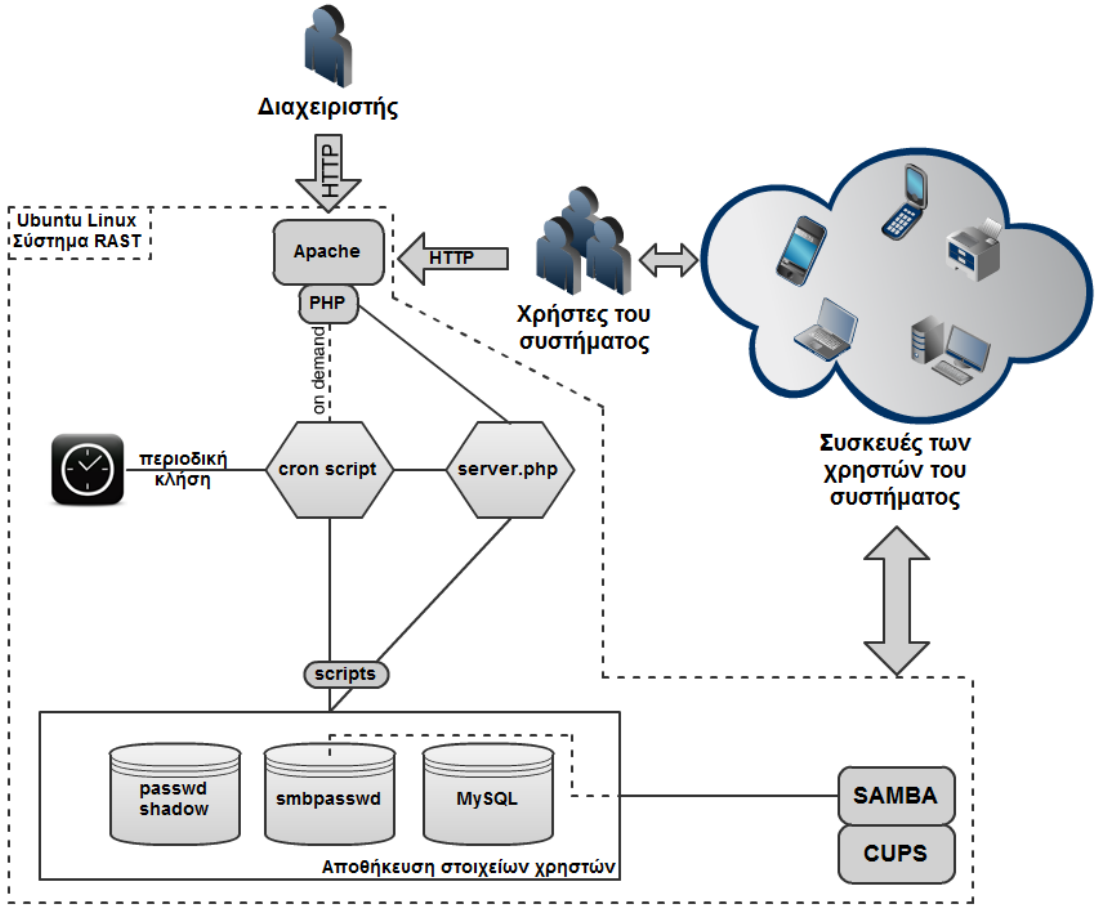
5

Τεχνικές προδιαγραφές του συστήματος RAST

Στο κεφάλαιο αυτό θα παρουσιαστούν τα διάφορα τεχνικά χαρακτηριστικά του συστήματος RAST. Αρχικά αναλύεται η βασική αρχιτεκτονική, και όλα τα συστατικά στοιχεία που παρέχουν την λειτουργικότητά του. Στη συνέχεια παρουσιάζονται αναλυτικότερα όλες οι τεχνολογίες και πλατφόρμες που χρησιμοποιήθηκαν, οι λόγοι για τους οποίους επιλέχθηκαν, και μια συνοπτική παρουσίαση της κάθε μιας.

5.1 Η αρχιτεκτονική του συστήματος RAST

Στην ενότητα αυτή θα παρουσιαστεί η γενική αρχιτεκτονική του RAST, καθώς και οι μηχανισμοί που του επιτρέπουν να παρέχει τις δυνατότητές του. Παρακάτω ακολουθεί ένα συνοπτικό διάγραμμα με τα διάφορα συστατικά στοιχεία του συστήματος:



Σχήμα 13: Γενική αρχιτεκτονική του συστήματος

Σαν λειτουργικό σύστημα για την εφαρμογή επιλέχθηκε το GNU/Linux και συγκεκριμένα η διανομή Ubuntu Linux, μια διανομή που έχει δώσει μεγάλο βάρος στην ευκολία και την αμεσότητα της ενημέρωσης των στοιχείων της. Για την υποστήριξη της διεπαφής ιστοσελίδων (web interface) χρησιμοποιήθηκε ο εξυπηρετητής ιστοσελίδων Apache Web Server, υποστηριζόμενος από την γλώσσα PHP για τα δυναμικά κομμάτια της εφαρμογής. Για την διαμοίραση φακέλων και εκτυπωτών χρησιμοποιήθηκε το πρόγραμμα Samba, ενώ για τις συγκεκριμένες λειτουργίες εκτύπωσης, καθώς και την υποστήριξη μεγάλου αριθμού εκτυπωτών χρησιμοποιήθηκε το σύστημα CUPS. Για την αποθήκευση των διαφόρων στοιχείων εκτέλεσης της εφαρμογής χρησιμοποιήθηκαν οι βάσεις χρηστών passwd/shadow του Linux, η βάση smbpasswd του samba, η σχεσιακή βάση δεδομένων MySQL καθώς και ειδικά αρχεία και φάκελοι στο σύστημα διαχείρισης αρχείων (filesystem) του λειτουργικού. Όλες οι λειτουργίες διαχείρισης χρηστών και πόρων έχουν υλοποιηθεί μέσω εντολών του Linux, όπως το mount, ενώ για τα κομμάτια που δεν υπήρχε αντίστοιχη εντολή, χρησιμοποιήθηκαν εξειδικευμένα αρχεία σεναρίου (scripts), η

πλειονότητα των οποίων είναι υλοποιημένα στην γλώσσα PERL, τα οποία βρέθηκαν από το πρόγραμμα FAUS και διορθώθηκαν ώστε να είναι συμβατά με την εφαρμογή. Τέλος, για τον προγραμματισμό της περιοδικής συντήρησης χρησιμοποιήθηκε το πρόγραμμα cron του Unix.

Ο πυρήνας της εφαρμογής, το κομμάτι αυτό δηλαδή που παραμετροποιεί όλα τα υπόλοιπα υποσυστήματα ώστε να παρέχεται η απαιτούμενη λειτουργικότητα, είναι το server.php, ένα ειδικό αρχείο σεναρίου (script) που λαμβάνει εντολές από την διεπαφή ιστοσελίδων (web interface) και αναλαμβάνει να συντονίσει όλα τα υπόλοιπα υποσυστήματα. Ειδικό ρόλο έχει και το cron script, το αρχείο σεναρίου (script) δηλαδή που εκτελείται περιοδικά. Το συγκεκριμένο πρόγραμμα έχει την ευθύνη της διατήρησης της ακεραιότητας όλων των βάσεων του συστήματος, και ταυτόχρονα της διακοπής προσβάσεων που έχουν λήξει ή κατά οποιοδήποτε άλλο τρόπο αρθεί.

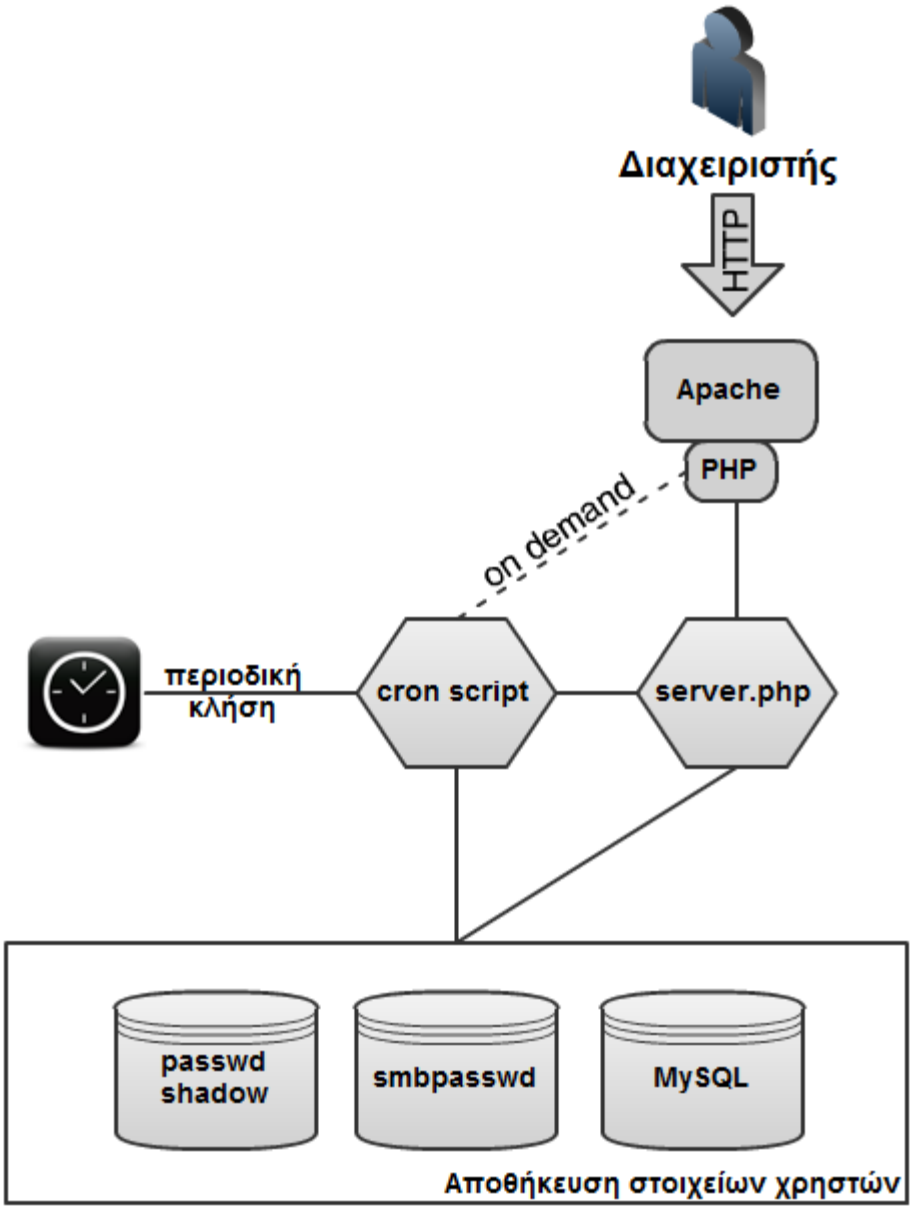
Έχοντας λοιπόν περιγράψει όλα τα συστατικά του συστήματος, η λειτουργικότητα έχει ως εξής:

Η ρύθμιση των παραμέτρων όλων των υπηρεσιών που χρησιμοποιούνται γίνεται μέσω ειδικών ιστοσελίδων. Εκεί, ο διαχειριστής εισάγει τους χρήστες, και οι χρήστες εισάγουν τους πόρους τους. Κάθε διαδικασία τέτοιου είδους, δίνει εντολή στο server.php να ενημερώσει τις αντίστοιχες βάσεις, όπως θα φανεί στις αναλυτικότερες περιπτώσεις που παρουσιάζονται παρακάτω. Το server.php αναλαμβάνει να ελέγξει την ορθότητα των εισαγόμενων στοιχείων, έπειτα εκτελεί εντολές ενημέρωσης των βάσεων, ορισμένες εκ των οποίων είναι αρχεία σεναρίου (scripts), και τέλος επιστρέφει πίσω στις δυναμικές σελίδες πληροφορίες σχετικά με την επιτυχία ή την αποτυχία της εντολής.

Ταυτόχρονα, το cron script εκτελείται περιοδικά, και αναγνωρίζει εργασία που πρέπει να γίνουν για να συντηρηθεί το σύστημα. Παράδειγμα λειτουργικότητάς του είναι να ελέγχει την λήξη ενός λογαριασμού, και εφόσον ο λογαριασμός έχει όντως λήξει, να τον διαγράφει μαζί με τους πόρους του. Επιπλέον, αν ένας λογαριασμός έχει διαγραφεί από τη μια βάση δεδομένων, το cron script θα προχωρήσει στην διαγραφή του συγκεκριμένου λογαριασμού από όλες τις βάσεις.

Παρακάτω, και σε χωριστές ενότητες, θα αναλυθεί η λειτουργικότητα όλων αυτών των δομικών του συστήματος στοιχείων, ώστε να γίνει απολύτως κατανοητή η λειτουργία του.

5.1.1 Διαχείριση χρηστών από την διεπαφή ιστοσελίδων



Σχήμα 14: Αρχιτεκτονική συστήματος χρηστών

Το παραπάνω σχήμα απεικονίζει την διαδικασία διαχείρισης χρηστών. Για όλη την διαδικασία υπεύθυνος είναι ο διαχειριστής του συστήματος, που ορίζει τους χρήστες. Όταν λοιπόν ζητήσει από το σύστημα την προσθήκη ενός χρήστη, καλείται το server.php με τις κατάλληλες παραμέτρους (όνομα χρήστη, κωδικός πρόσβασης, χρόνος λήξης κλπ), και αυτό με τη σειρά του δημιουργεί τον χρήστη στις 3 βάσεις του συστήματος. Αν η κλήση αυτή είναι επιτυχής, επιστρέφεται μήνυμα επιτυχίας, αλλιώς διαγράφει τον χρήστη από τυχόν βάσεις που κατάφερε να τον εγγράψει και επιστρέφει μήνυμα αποτυχίας (πχ. αδυναμία σύνδεσης με τη βάση MySQL).

Αντίστοιχη είναι και η ενημέρωση των στοιχείων του χρήστη (συνήθως μια ενημέρωση αφορά επιμήκυνση του χρόνου παραμονής του χρήστη στο σύστημα). Και πάλι το server.php ενημερώνει τις σελίδες php για την επιτυχία ή την αποτυχία του εγχειρήματος.

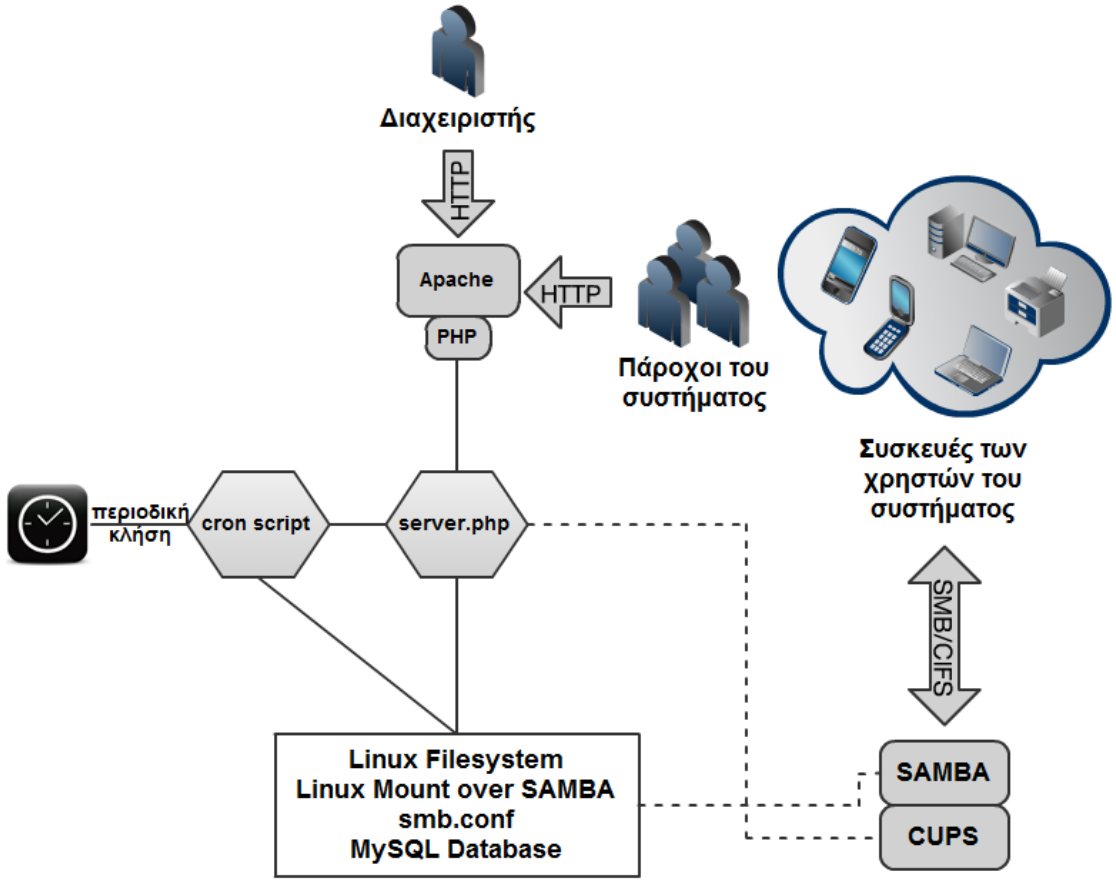
Ενδιαφέρον παρουσιάζει η περίπτωση της διαγραφής. Στην διαγραφή το server.php απλά διαγράφει τον χρήστη από τη βάση MySQL, και καλεί το cron script για να διαγράψει τελείως τον χρήστη και τους πόρους που αυτός έχει μοιράσει στο σύστημα. Αυτή η κλήση φαίνεται στο σχήμα με την διακεκομμένη γραμμή που σημειώνεται «on demand».

Στις περιπτώσεις προσθήκης και διαγραφής, υπάρχουν εργαλεία που υποστηρίζουν την μαζική προσθήκη ή διαγραφή λογαριασμών. Στην μαζική προσθήκη δίνονται στο σύστημα σαν παράμετροι το πρόθεμα των χρηστών, και δημιουργείται ένας αριθμός λογαριασμών με το πρόθεμα καθώς και κάποιος αναγνωριστικός αριθμός (πχ. με πρόθεμα lab δημιουργούνται λογαριασμοί lab001, lab002, lab003 κ.ο.κ.) και επιστρέφονται οι λογαριασμοί με τους αντίστοιχους κωδικούς πρόσβασης, που το σύστημα δημιουργεί τυχαία. Το server.php ουσιαστικά αφού επιλέξει κωδικό χρήστη και φτιάξει το αντίστοιχο όνομα χρήστη, στέλνει την αίτηση εγγραφής νέου χρήστη μια φορά για κάθε απαιτούμενο λογαριασμό. Η μαζική διαγραφή μοιράζεται αυτό το τελευταίο χαρακτηριστικό, καθώς το σύστημα την χειρίζεται σαν μια ακολουθία από μεμονωμένες διαγραφές.

Τέλος, το cron script τρέχει και περιοδικά, για να διαγράψει τυχόν ληγμένους λογαριασμούς, πόρους που δεν ισχύουν και να διορθώσει πιθανές ανωμαλίες στις βάσεις δεδομένων.

Όλες φυσικά οι παραπάνω διαδικασίες καταγράφονται αναλυτικά στα αρχεία καταγραφής του συστήματος, μαζί με όλες τις παραμέτρους και όλες τις επιστροφές, ώστε ο διαχειριστής να είναι σε θέση να εντοπίσει την πηγή πιθανής αστοχίας ή σφάλματος.

5.1.2 Προσθήκη πόρου στο σύστημα



Σχήμα 15: Γενική εικόνα προσθήκης πόρων στο σύστημα

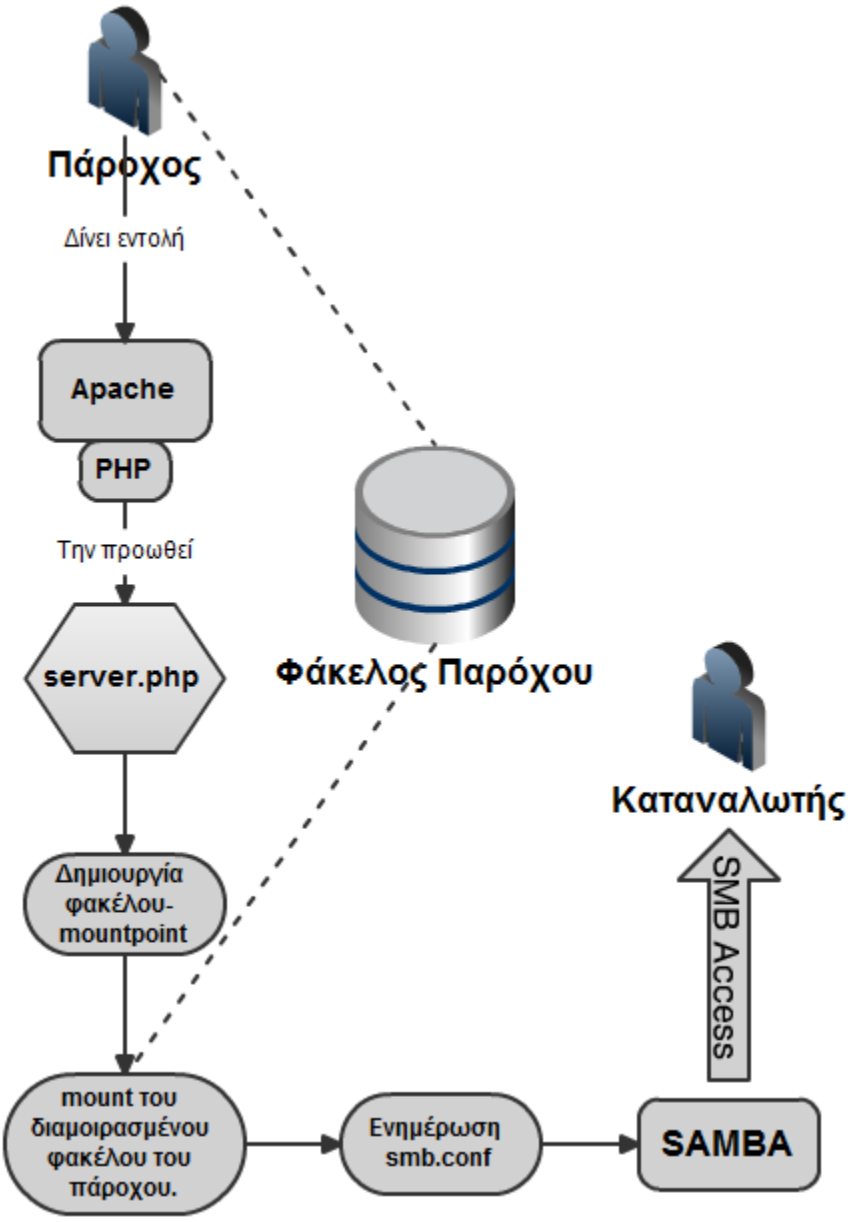
Για να προστεθεί ένας πόρος στο σύστημα, ακολουθείται μια αντίστοιχη με τα παραπάνω διαδικασία.

Έστω λοιπόν ότι ένας πάροχος θέλει να διαμοιράσει κάποιον φάκελο. Εφόσον έχει μοιράσει τον φάκελο αυτόν στο δίκτυο, μπαίνει στην φόρμα εισαγωγής φακέλου στο σύστημα. Εφόσον το σύστημα πάρει έγκυρα στοιχεία, δημιουργεί έναν κενό φάκελο, που θα χρησιμοποιηθεί σαν mountpoint[4]. Έπειτα, κάνει mount[5] τον διαμοιραζόμενο φάκελο στο mountpoint. Τέλος, μοιράζει τον φάκελο ενημερώνοντας το αρχείο ρυθμίσεων του Samba smb.conf[6]. Έτσι, ο φάκελος αυτός γίνεται διαθέσιμος στο σύστημα. Πιο αναλυτικά η διαδικασία φαίνεται στο παρακάτω σχήμα:

[4] Mountpoint: εικονικός φάκελος που χρησιμεύει για να συνδεθεί κάποιο εξωτερικό σύστημα αρχείων. Πχ., κάτω από τον φάκελο /mnt/cdrom συνδέονται τα περιεχόμενα ενός CD.

[5] Mount: εντολή του unix που συνδέει εξωτερικό σύστημα αρχείων (πχ. cd, δεύτερος σκληρός δίσκος, φάκελος μοιραζόμενος στο δίκτυο) σε κάποιον εικονικό φάκελο – mountpoint.

[6] Περισσότερες πληροφορίες για το smb.conf παρέχονται στο κεφάλαιο αναλυτικότερης επεξήγησης του SAMBA.



Σχήμα 16: Διαδικασία εισαγωγής φακέλου στο σύστημα.

Στην περίπτωση του εκτυπωτή, το σύστημα πέραν της θέσης του εκτυπωτή στο δίκτυο, χρειάζεται και την πληροφορία για το μοντέλο του εκτυπωτή, για να εγκαταστήσει κατάλληλο οδηγό (driver). Έπειτα, στέλνει την εντολή αντίστοιχης ρύθμισης στο CUPS ώστε αυτό να τον αναγνωρίσει. Τέλος, στέλνει την εντολή στο CUPS για διαμοιρασμό του συγκεκριμένου εκτυπωτή. Έτσι, επιτυγχάνεται η υποστήριξη της εκτύπωσης από το σύστημα.

Όλες φυσικά οι παραπάνω διαδικασίες καταγράφονται αναλυτικά στο αρχείο καταγραφής, και όλοι οι πόροι διαγράφονται από το σύστημα μέσω του cron script, όταν ο αντίστοιχος πάροχος σταματήσει να έχει πρόσβαση στο σύστημα.

5.2 Τεχνικές προδιαγραφές

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιήσαμε, καθώς και μια αιτιολόγηση των επιλογών που έγιναν.

5.3 Open source τεχνολογίες / γενικά

Σύμφωνα με τον οργανισμό OSI (Open Source Initiative) [19], ανοικτό λογισμικό (open source) δεν σημαίνει μόνο ελεύθερη πρόσβαση στον κώδικα. Οι όροι της πρόσβασης σε αυτό, σύμφωνα με τον ορισμό του ανοικτού λογισμικού που δημοσιεύει ο συγκεκριμένος οργανισμός (OSD – Open Source Definition) [20], πρέπει να συμβαδίζουν με συγκεκριμένα κριτήρια:

1. **Ελεύθερη αναδιανομή:** Η άδεια χρήσης δεν πρέπει να περιορίζει κανένα συμβαλλόμενο μέρος από την πώληση ή τη δωρεά του λογισμικού ως στοιχείου μιας συνολικής διανομής λογισμικού που περιέχει προγράμματα από αρκετές διαφορετικές πηγές. Η άδεια χρήσης δεν πρέπει να απαιτεί δικαιώματα εκμετάλλευσης ή άλλη αμοιβή για τέτοιου είδους πώληση.
2. **Πηγαίος κώδικας:** Το πρόγραμμα πρέπει να περιλαμβάνει τον πηγαίο κώδικα, ενώ συγχρόνως πρέπει να επιτρέπεται η διάθεσή του είτε ως πηγαίος κώδικας είτε σε μεταγλωττισμένη μορφή. Όταν κάποιο πρόγραμμα δε διανέμεται μαζί με τον πηγαίο του κώδικα, πρέπει να υπάρχει ένας ευρέως γνωστός τρόπος μέσω του οποίου κανείς μπορεί με ελάχιστο κόστος να τον αποκτήσει (προτιμότερος τρόπος είναι η διάθεσή του μέσω του Διαδικτύου χωρίς χρέωση). Ο πηγαίος κώδικας πρέπει να είναι η μορφή του προγράμματος την οποία οι προγραμματιστές θα προτιμούν προκειμένου να προβούν σε τροποποιήσεις του. Πηγαίος κώδικας που προκαλεί εσκεμμένα σύγχυση δεν είναι επιτρεπτός. Ενδιάμεσες μορφές, όπως το αποτέλεσμα ενός προεπεξεργαστή (preprocessor) ή ενός μεταφραστή δεν είναι επιτρεπτές.
3. **Παραγόμενα έργα:** Η άδεια χρήσης πρέπει να επιτρέπει τροποποιήσεις του προγράμματος, καθώς και πιθανά παραγόμενα

έργα, τα οποία πρέπει να διανέμονται με τους ίδιους όρους που διέπουν το αρχικό λογισμικό.

4. **Ακεραιότητα του πηγαίου κώδικα του συγγραφέα:** Η άδεια χρήσης έχει τη δυνατότητα να απαγορεύει τη διανομή του κώδικα όταν αυτός έχει τροποποιηθεί, μόνο αν επιτρέπεται η διανομή "βελτιωτικών αρχείων" μαζί με τον πηγαίο κώδικα, ούτως ώστε να πραγματοποιείται η τροποποίηση του προγράμματος κατά το χρόνο δημιουργίας. Η άδεια χρήσης πρέπει να επιτρέπει ρητά τη διανομή λογισμικού που παράγεται από τροποποιημένο πηγαίο κώδικα. Επίσης, η άδεια ενδέχεται να απαιτεί από τα παραγόμενα έργα να διαθέτουν διαφορετικό όνομα ή διαφορετική έκδοση από το αρχικό λογισμικό.
5. **Καμία διάκριση εναντίον ατόμων ή ομάδων ατόμων:** Η άδεια χρήσης δεν πρέπει περιλαμβάνει διακρίσεις εναντίον ατόμων ή ομάδων ατόμων.
6. **Καμία διάκριση εναντίον κάποιων τομέων δραστηριοποίησης:** Η άδεια χρήσης δεν πρέπει να περιορίζει κανέναν από τη χρησιμοποίηση του προγράμματος σε κάποιο συγκεκριμένο πεδίο δραστηριοποίησης. Για παράδειγμα, δεν μπορεί να περιορίζει τη χρήση του προγράμματος για την εξυπηρέτηση των αναγκών μιας επιχείρησης ή μιας ερευνητικής ομάδας που εξετάζει ζητήματα γενετικής.
7. **Διανομή της άδειας χρήσης:** Τα δικαιώματα του προγράμματος πρέπει να ισχύουν για όλα τα άτομα στα οποία αυτό αναδιανέμεται, χωρίς να απαιτείται από αυτά να κάνουν χρήση κάποιας επιπρόσθετης άδειας χρήσης.
8. **Η άδεια χρήσης δεν πρέπει να αφορά μόνο ένα συγκεκριμένο προϊόν:** Τα δικαιώματα του προγράμματος δεν πρέπει να εξαρτώνται από το αν αυτό είναι τμήμα μιας συγκεκριμένης διανομής λογισμικού. Αν το πρόγραμμα αυτό εξαχθεί από τη διανομή αυτή και χρησιμοποιηθεί ή διανεμηθεί με τους όρους της άδειας χρήσης του προγράμματος, τότε όλα τα άτομα στα οποία αυτό θα αναδιανεμηθεί πρέπει να διαθέτουν τα ίδια δικαιώματα με αυτά που παραχωρούνται στην αρχική διανομή του λογισμικού.
9. **Η άδεια χρήσης δεν πρέπει να περιορίζει άλλα λογισμικά:** Η άδεια χρήσης ενός λογισμικού δεν πρέπει να θέτει περιορισμούς σε άλλα

λογισμικά τα οποία διανέμονται μαζί με αυτό. Για παράδειγμα, η άδεια χρήσης δεν πρέπει να απαιτεί όλα τα υπόλοιπα προγράμματα που υπάρχουν στην ίδια διανομή να είναι ανοιχτού κώδικα.

10. **Η άδεια χρήσης πρέπει να είναι τεχνολογικά ουδέτερη:** Κανένας όρος της άδειας χρήσης δεν πρέπει να επιβάλλει τη χρήση συγκεκριμένων τεχνολογιών ή διεπαφών.

Υπάρχουν περισσότερες από 50 άδειες χρήσης "ανοιχτού λογισμικού" ή "ελεύθερου λογισμικού", αλλά ευτυχώς μπορούν να κατηγοριοποιηθούν με πολύ εύκολο τρόπο σε αυτές που [21]:

- **"παρέχουν αναγνώριση":** επιτρέπεται η χρήση, η τροποποίηση και η αναδιανομή, όμως στην περίπτωση που το πρόγραμμα αναδιανέμεται πρέπει να αναγνωριστεί η προσφορά του αρχικού συγγραφέα. Σχετικά παραδείγματα είναι: η BSD license και η Apache license v2.
- **"παρέχουν διορθώσεις":** επιτρέπεται η χρήση, η τροποποίηση και η αναδιανομή, όμως στην περίπτωση που το πρόγραμμα αναδιανέμεται όλες οι αλλαγές του πηγαίου κώδικα πρέπει να αποσταλούν στον αρχικό συγγραφέα. Σχετικά παραδείγματα είναι: οι άδειες χρήσης τύπου Mozilla (Mozilla Public License)
- **"παρέχουν τα πάντα":** επιτρέπεται η χρήση, η τροποποίηση και η αναδιανομή, όμως στην περίπτωση της αναδιανομής πρέπει ο πηγαίος κώδικας οποιουδήποτε παραγόμενου προϊόντος να είναι διαθέσιμος. Σχετικό παράδειγμα είναι η GPL.

Στην παρούσα διπλωματική χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες :

1. **Linux / Ubuntu Linux** που έχει γραφτεί και διανέμεται υπό την 3η έκδοση της GNU General Public License [22], η οποία επιτρέπει την ελεύθερη διανομή του κώδικα και την διαθεσιμότητά του στο κοινό, όπως διατυπώθηκε από το Ίδρυμα Ελεύθερου Λογισμικού (Free Software Foundation).
2. **Samba** που διανέμεται κάτω από την GNU General Public License [23].
3. **CUPS** που παρέχεται κάτω από την GNU General Public License (GPL) και GNU Library General Public License (LFPL), 2η έκδοση [24].

4. **Apache Server** που παρέχεται με μια ειδικά διαμορφωμένη άδεια ανοιχτού κώδικα. Αυτή είναι σύμφωνη με τις βασικές αρχές των αδειών ανοιχτού/ελεύθερου κώδικα, αφού αποτελεί βασικά μια παραλλαγή της GNU/GPL. Έχει την ονομασία Apache License 2 [25].
5. **PHP** που παρέχεται κάτω από την άδεια χρήσης PHP License [26]. Η τελευταία είναι μια άδεια ανοιχτού κώδικα σχεδιασμένη να ενθαρρύνει την ευρεία υιοθέτηση του πηγαίου κώδικά της. Η αναδιανομή της επιτρέπεται είτε σε μορφή κώδικα είτε σε μορφή εκτελέσιμου αρχείου με ή χωρίς τροποποιήσεις υπό την προϋπόθεση ότι:
    a. Θα περιέχεται η PHP License
    b. Η λέξη "PHP" δεν περιέχεται στον τίτλο καμίας παράγωγης εργασίας
    c. Η ακόλουθη αναγνώριση θα περιέχεται σε οποιαδήποτε μορφή αναδιανέμεται ο κώδικας: "This product includes PHP software, freely available from <http://www.php.net/software/>"
6. **PERL** που μπορεί να χρησιμοποιηθεί ελεύθερα κάτω από τους όρους της GNU General Public License [27].
7. **MySQL** που μπορεί να χρησιμοποιηθεί ελεύθερα κάτω από τους όρους της GPL License, υπό την προϋπόθεση ότι αναπτύσσεις και διανέμεις λογισμικό ανοικτού κώδικα κάτω από τους όρους της GPL License [28].
8. **FAUS** που διανέμεται κάτω από την GNU General Public License (GPL) όπως φαίνεται και στη ηλεκτρονική διεύθυνση: http://sourceforge.net/projects/faus/develop.

5.4 Linux & Ubuntu Linux

Σχήμα 17: Το λογότυπο του Ubuntu Linux (πηγή: http://ubuntu.org)

5.4.1 Γενικά

Το Linux [29],[30] είναι ένα ανοικτού κώδικα, ελεύθερο και δωρεάν λειτουργικό σύστημα βασισμένο στον πυρήνα Unix. Ξεκίνησε να αναπτύσσεται το 1991 και πλέον είναι το πιο διαδεδομένο από τα ελεύθερα λειτουργικά συστήματα, ενώ πλέον ανταγωνίζεται και τα μη-ελεύθερα λειτουργικά συστήματα. Στην καθιέρωσή του βοήθησε η εκτεταμένη χρήση του σε servers, καθώς και η πρόσθετη ασφάλεια και σταθερότητα που προσφέρουν τα τύπου Unix συστήματα. Στην ανάπτυξή του συμμετέχουν χιλιάδες εθελοντές αλλά και εταιρείες. Κυκλοφορεί σε πολλές διαφορετικές διανομές, δηλαδή συλλογές προγραμμάτων που αποτελούνται από τον πυρήνα και διαφορετικά συνοδευτικά προγράμματα, αναλόγως την ειδική χρήση και το επίπεδο χρηστών στο οποίο απευθύνεται η καθεμιά: άλλες διανομές έχουν σαν στόχο τη φιλικότητα στο χρήστη, άλλες τις εφαρμογές πολυμέσων, την ευκολία παραμετροποίησης κ.α. Παρ' όλο που το όνομα Linux αφορά τον πυρήνα του λειτουργικού συστήματος, δηλαδή το κομμάτι που επικοινωνεί με τις συσκευές, τον επεξεργαστή, παρέχει το σύστημα αρχείων και τη δικτύωση, πλέον συνηθίζεται να αναφερόμαστε σε αυτό εννοώντας όλο το λειτουργικό σύστημα, που περιλαμβάνει και το περιβάλλον εργασίας, και το συνοδευτικό λογισμικό.

Ο πυρήνας Linux είναι μία πρωτότυπη υλοποίηση πυρήνα λειτουργικού συστήματος. Αν και δεν χρησιμοποιεί κώδικα του UNIX, μπορεί να θεωρηθεί παρεμφερές σύστημα (Unix-like) ή ελεύθερη υλοποίησή του, αφού διαθέτει τις περισσότερες εντολές του και την ίδια σχεδόν δομή αρχείων, ενώ η φιλοσοφία της σχεδίασής του πλησιάζει περισσότερο το UNIX από οποιοδήποτε άλλο λειτουργικό σύστημα. Σήμερα το Linux παρέχει όλα όσα θεωρούνται αναγκαία για ένα σύγχρονο πυρήνα λειτουργικού, όπως:

- υποστήριξη πολυεπεξεργαστικών συστημάτων (SMP)
- πραγματική πολυδιεργασία
- εικονική μνήμη
- διαμοιραζόμενες βιβλιοθήκες
- σωστή διαχείριση μνήμης
- δικτύωση μέσω TCP/IP κ.α.

Ο πυρήνας Linux αρχικά σχεδιάστηκε για επεξεργαστές της οικογένειας x86 (386/486/Pentium), αλλά σήμερα είναι συμβατός με πολύ μεγάλη ποικιλία επεξεργαστών, όπως οι Alpha (64 bit), οι Motorola 68000, PowerPC, MIPS κ.α.

Η ανάπτυξη του πυρήνα Linux ξεκίνησε περίπου το 1991 [31] από τον Φινλανδό Linus Torvalds (τότε φοιτητή ακόμη), ο οποίος με βοήθεια πολλών εθελοντών προγραμματιστών (είτε επαγγελματιών είτε ασχολούμενων από χόμπι) μέσω του Internet, κατάφερε να δημιουργήσει έναν πυρήνα που ανταγωνίζεται αντίστοιχους μεγάλων εταιριών. Αρχικά είχε σαν πρότυπο το Minix, ένα άλλο λειτουργικό τύπου Unix, το οποίο είχε αναπτύξει ο Andrew Tanenbaum για εκπαιδευτικούς σκοπούς. Ο τελευταίος εκείνη την εποχή δεν επέτρεπε την τροποποίηση και επέκταση του Minix και για το λόγο αυτό, ο Torvalds [32] δημιούργησε εξαρχής έναν πυρήνα για το αντικαταστήσει. Αρχικά ο πυρήνας αυτός ονομάστηκε FreaX (από τους όρους free και freak, με την κατάληξη X να υποδηλώνει ένα σύστημα τύπου Unix) αλλά αργότερα έλαβε την ονομασία Linux, ονομασία που επινόησε ο Ari Lemmke.

Η προσπάθεια για τη δημιουργία ενός ελεύθερου λειτουργικού συστήματος είχε ξεκινήσει παλαιότερα, το 1985, από τον Richard Stallman, ιδρυτή του Free Software Foundation και του εγχειρήματος GNU. Έως ότου ξεκινήσει η ανάπτυξη του Linux, το εγχείρημα GNU είχε ήδη δημιουργήσει ένα C μεταγλωττιστή (τον gcc) και μια πλειάδα υψηλής ποιότητας προγραμματιστικών εργαλείων, ενώ είχε έτοιμα προγράμματα που αντικαθιστούσαν όλα τα βασικά προγράμματα σε ένα UNIX σύστημα. Το μόνο που έλειπε ήταν ένας σταθερός πυρήνας. Έτσι το GNU βρήκε έναν πυρήνα για να λειτουργήσει, και το Linux βρήκε έτοιμη μια μεγάλη ποικιλία προγραμμάτων (το εγχείρημα GNU συνεχίζει και σήμερα την ανάπτυξη του λειτουργικού του συστήματος, του GNU Hurd, το οποίο βασίζεται στον μικροπυρήνα Mach). Λόγω της σύζευξης αυτής μεταξύ GNU και Linux, υποστηρίζεται από το FSF και αρκετούς χρήστες ότι η σωστή ονομασία είναι GNU/Linux.

Σήμερα το Linux και τα εργαλεία και εφαρμογές που προορίζονται για αυτό αναπτύσσονται από χιλιάδες προγραμματιστές σε όλο τον κόσμο. Κάθε διανομή υποστηρίζεται από μια οργανωμένη κοινότητα χρηστών και προγραμματιστών, ενώ ορισμένες από τις διανομές υποστηρίζονται και από εταιρίες λογισμικού (π.χ. η Suse από τη Novell, η Fedora από τη Redhat, η Ubuntu από την Canonical) που πωλούν είτε εμπορικές εκδόσεις είτε τεχνική υποστήριξη για το λειτουργικό. Επίσης, περίπου 200 εταιρίες έχουν συνεισφέρει τα τελευταία χρόνια στην ανάπτυξη του πυρήνα Linux -ανάμεσα στις οποίες πολύ γνωστές όπως η IBM, η Intel, η Google, η Hewlett Packard- κυρίως για λόγους καλύτερων πωλήσεων του harware τους -με δεδομένη τη διάδοση του Linux στην αγορά των server, των κινητών τηλεφώνων και των

netbooks. Το Linux αναπτύσσεται με βάση το POSIX πρότυπο, το οποίο είναι μία προσπάθεια τυποποίησης όλων των συστημάτων που βασίζονται ή προσομοιάζουν στο UNIX.

Σήμερα υπάρχουν πολλές διαφορετικές διανομές που καλύπτουν διαφορετικές ανάγκες. Μερικές χαρακτηριστικές είναι:

- **Debian GNU/Linux:** Οργανωμένο από μια ομάδα εθελοντών, και είναι η διανομή με τα περισσότερα πακέτα σήμερα. Είναι η σημαντικότερη διανομή που αποτελείται μόνο από ελεύθερα πακέτα.
- **Ubuntu Linux:** Ίσως η πιο δημοφιλής διανομή αυτή τη στιγμή. Βασίζεται στο Debian και ένα από τα βασικά στοιχεία της φιλοσοφίας της είναι η φιλικότητα προς το χρήστη.
- **Knoppix:** Live διανομή, που δεν χρειάζεται εγκατάσταση αλλά λειτουργεί απ'ευθείας από το CD, που βασίζεται στο Debian. Πολύ χρήσιμη διανομή σε περιπτώσεις ανάκτησης δεδομένων όταν το κυρίως λειτουργικό σύστημα του υπολογιστή δεν μπορεί να ξεκινήσει.
- **Damn Small Linux:** Ακόμα μια διανομή βασισμένη στο Knoppix Linux που καταλαμβάνει μόνο 50MB χώρου και περιλαμβάνει πλήρες σετ εφαρμογών. Λόγω της ταχύτητας της μπορεί να χρησιμοποιηθεί άνετα σε παλιούς υπολογιστές.
- **Slackware Linux:** Η «αγαπημένη» διανομή αυτών που ξεκίνησαν με το Linux στις αρχές της δεκαετίας του '90.
- **Redhat Linux:** Η Redhat είναι μία από τις πρώτες εταιρείες που αντιμετώπισαν σοβαρά το Linux. Σήμερα κατέχει ένα μεγάλο ποσοστό της αγοράς. Διατίθεται μόνο σε εμπορική έκδοση.
- **Fedora Core:** Διανομή που προήλθε από το Redhat Linux και υποστηρίζεται από τη Redhat. Λειτουργεί ως δοκιμαστικό πεδίο για τις σταθερές εκδόσεις του Redhat Linux αλλά αποτελεί και η ίδια μια πολύ σταθερή διανομή. Σε αντίθεση με το Redhat Linux διατίθεται ελεύθερα προς χρήση.
- **SuSe Linux:** Έγινε ιδιαίτερα δημοφιλής λόγω της φιλικότητάς της προς τον χρήστη και των πολλών πακέτων που διαθέτει.

- **Mandriva Linux:** Βασισμένη στο Redhat, αλλά με ιδιαίτερα προσεγμένο γραφικό περιβάλλον. Μέχρι πρότινος ήταν γνωστή ως Mandrake.
- **Gentoo Linux:** Διανομή που μπορεί να παραμετροποιηθεί στο έπακρο αφού όλα τα προγράμματα, αλλά και το ίδιο το λειτουργικό, μπορούν να μεταφράζονται (compile) κατά την εγκατάσταση τους. Γι' αυτό το λόγο αποτελεί μια από τις ταχύτερες διανομές.

Το Linux όπως προαναφέραμε είναι ο πυρήνας, και πάνω σε αυτόν μπορεί να εκτελεστεί οποιοδήποτε περιβάλλον εργασίας. Το πιο διαδεδομένο παραθυρικό σύστημα όμως είναι το X Window System και πιο συγκεκριμένα η υλοποίηση από την ομάδα Xfree86. Το X Window System (ή πιο απλά τα X), είναι ένα γραφικό σύστημα που συντηρείται και αναπτύσσεται σήμερα από το OpenGroup και πέρα από της συνήθεις λειτουργίες ενός παραθυρικού συστήματος, είναι κατασκευασμένο για δικτυακή λειτουργία. Δηλαδή μπορεί πολύ απλά μια παραθυρική εφαρμογή να εκτελείται στον Α υπολογιστή, και η έξοδος (τα παράθυρα) να εμφανίζονται στον δικό μας υπολογιστή. Πέρα όμως από αυτές τις χαμηλού επιπέδου λειτουργίες του διακομιστή X, δεν διαθέτει τίποτα παραπάνω. Αυτό το κενό καλύπτουν τα λεγόμενα περιβάλλοντα εργασίας (Desktop Environments), τα οποία μπορεί να περιέχουν γραμμές εργασιών (Taskbars), εικονίδια στην επιφάνεια εργασίας, εικόνες φόντου (backgrounds), προφύλαξη οθόνης (screensaver), καθώς και ένα αριθμό προγραμμάτων που διευκολύνουν την διαχείριση της επιφάνειας εργασίας ή και του συστήματος. Τα πιο υψηλού επιπέδου περιβάλλοντα εργασίας για Linux είναι τα KDE και GNOME, τα οποία έχουν ήδη φτάσει (αν όχι ξεπεράσει) τα αντίστοιχα περιβάλλοντα εργασίας σε άλλα UNIX workstations.

Είναι διαδεδομένη η άποψη ότι δεν ενδείκνυται η χρήση του από νέους χρήστες των υπολογιστών, ή από χρήστες χωρίς ιδιαίτερες γνώσεις στους υπολογιστές. Ωστόσο, σήμερα, υπάρχει μια εξαιρετικά μεγάλη βάση υψηλού επιπέδου προγραμμάτων, που επιτρέπουν την διαχείριση του συστήματος χωρίς την γνώση των βασικών εντολών του UNIX.

Αυτό που το κάνει να διαφέρει από τα υπόλοιπα λειτουργικά συστήματα, είναι η ευκολία με την οποία μπορεί να επεκταθεί για να καλύψει και τις πιο απαιτητικές ανάγκες. Ακόμα και αν δεν έχει κάποιος γνώσεις προγραμματισμού, μπορεί να προτείνει βελτιώσεις στους αρχικούς προγραμματιστές ή ακόμα να χρηματοδοτήσει

κάποιον για να υλοποιήσει αυτές τις βελτιώσεις (πολλά ελεύθερα προγράμματα χρηματοδοτούνται και αναπτύσσονται με αυτόν τον τρόπο).

Το Linux, καθώς και τα περισσότερα συνοδευτικά προγράμματα, διανέμεται υπό τους όρους του GNU General Public License. Η άδεια αυτή δημιουργήθηκε για να παράγει προγράμματα που θα διανεμηθούν ελεύθερα, αλλά και για να διατηρήσει αυτή την ελευθερία των προγραμμάτων. Έτσι κάποιο πρόγραμμα κάτω από την GNU GPL πρέπει υποχρεωτικά να συνοδεύεται από τον πηγαίο κώδικα του, ενώ στην περίπτωση που κάποιος τροποποιήσει ένα τέτοιο πρόγραμμα και θέλει να το διανέμει είναι υποχρεωμένος να διανέμει τον αρχικό κώδικα καθώς και τις δικές του τις αλλαγές στον κώδικα. Με την έννοια του ελεύθερου προγράμματος δεν υπονοείται ότι είναι δωρεάν, αλλά ότι διανέμεται ελεύθερα, με τον πηγαίο κώδικά του διαθέσιμο στον καθένα, και αυτό δίνει την δυνατότητα όχι μόνο της χρησιμοποίησης του λειτουργικού, αλλά και της αναδιανομής του, της πώλησης του, τροποποίησής του, της επέκτασής του, πρόσβαση στον πηγαίο κώδικα και συνήθως σε εκτενή τεκμηρίωση.

Το λογισμικό του Linux που υπόκειται στην άδεια GNU GPL μπορεί να αντιγραφεί, να παραχωρηθεί ή ακόμη και να πωληθεί ελεύθερα. Το αν αυτό ισχύει για ολόκληρες διανομές του Linux, εξαρτάται πρωτίστως από την συγκεκριμένη διανομή. Αν η διανομή αυτή είναι η Debian GNU/Linux ή το Slackware, τότε όλα τα παραπάνω επιτρέπονται (μιας και δεν συμπεριλαμβάνει, στη βασική διανομή, μη ελεύθερα προγράμματα). Οι υπόλοιπες διανομές μπορεί να περιέχουν και μη ελεύθερο λογισμικό. Σε αυτές τις διανομές επιτρέπονται τα παραπάνω μόνο στα ελεύθερα προγράμματα τα οποία πρέπει να ξεχωρίσει ο ενδιαφερόμενος (συνήθως η κάθε διανομή δίνει μια ελεύθερη έκδοσή της, ή έχει σε ξεχωριστά CD τα μη ελεύθερα προγράμματα).

Οι διανομές συνήθως διαθέτουν πολύ μεγάλη ποικιλία προγραμμάτων. Υπάρχουν τα ελεύθερα προγράμματα τα οποία έρχονται με άδεια παρόμοια με του Linux (ή χαλαρότερη), τα οποία δεν υστερούν (κάποιες φορές εκτιμάται ότι ξεπερνούν) σε ποιότητα τα αντίστοιχα κλειστά προγράμματα (proprietary). Ίσως παλαιότερα να ήταν εύκολο να αριθμήσει κάποιος τις κατηγορίες προγραμμάτων για τις οποίες υπάρχει ελεύθερο λογισμικό. Σήμερα συντηρούνται μεγάλες βάσεις δεδομένων ώστε να ταξινομηθούν αυτά τα προγράμματα. Περισσότερες πληροφορίες για υπάρχοντα προγράμματα βρίσκονται σε ιστοσελίδες όπως:

- http://www.freshmeat.net

- http://www.sourceforge.net

Κλειστά προγράμματα (proprietary) υπάρχουν και στο Linux, και καλύπτουν αρκετούς τομείς εξειδικευμένου λογισμικού (παιχνίδια, βάσεις δεδομένων, εφαρμογές γραφείου, οδηγοί συσκευών κ.α.).

Πέρα από την μεγάλη ποικιλία εφαρμογών που έχουν δημιουργηθεί για το Linux, υπάρχει επίσης η δυνατότητα (όχι πάντα) να χρησιμοποιηθούν σε περιβάλλον Linux και προγράμματα που έχουν κατασκευαστεί για MS Windows. Αυτό γίνεται χρησιμοποιώντας κάποια "ενδιάμεση" εφαρμογή όπως είναι π.χ. το Wine, το οποίο είναι μία ελεύθερη υλοποίηση του API των Windows, και η οποία αναλαμβάνει να γεφυρώσει το χάσμα.

Σε ιδιαίτερη κατηγορία ανήκουν οι οδηγοί συσκευών (drivers). Λόγω της ιδιαίτερης φύσης τους, μπορεί να απαιτούνται για τη συγγραφή τους συγκεκριμένες πληροφορίες για τις προδιαγραφές και το σχεδιασμό της ελεγχόμενης συσκευής. Οι πληροφορίες αυτές δεν είναι πάντοτε διαθέσιμες, καθώς οι κατασκευαστές πολλές φορές διστάζουν να τις κοινοποιήσουν, επικαλούμενοι τα ιδιοκτησιακά τους δικαιώματα. Μερικές φορές είναι δυνατόν να δημιουργηθεί ένας "ελεύθερος" οδηγός με τη χρήση reverse engineering, και πράγματι για πολλά περιφερειακά υπάρχουν σήμερα τέτοιοι οδηγοί που λειτουργούν ικανοποιητικά. Σε άλλες περιπτώσεις, οι κατασκευαστές παρέχουν τις απαραίτητες πληροφορίες, αποβλέποντας στη διάδοση του προϊόντος τους στην επεκτεινόμενη κοινότητα των χρηστών του Linux. Με μερικές συσκευές οι χρήστες του Linux είναι υποχρεωμένοι να χρησιμοποιήσουν κλειστού κώδικα οδηγούς. Αυτό δημιουργεί μεγάλα προβλήματα στη διάδοση του ελεύθερου λογισμικού, καθώς οι οδηγοί αυτοί δεν μπορούν να διανεμηθούν ελεύθερα, ούτε και να τροποποιηθούν κατάλληλα, ακολουθώντας την εξέλιξη του Linux, και οι χρήστες τους εξαρτώνται ουσιαστικά από την καλή θέληση των κατασκευαστών. Το πρόβλημα αυτό είναι ιδιαίτερα έντονο σε σχέση με τα μόντεμ, διότι πολλά μόντεμ (software modems) είναι σχεδιασμένα να λειτουργούν με οδηγούς που είναι διαθέσιμοι μόνο για MS Windows. Τα μόντεμ αυτά καλούνται συνήθως "winmodems", ενώ για όσα από αυτά καθίσταται δυνατό να λειτουργήσουν με ελεύθερους οδηγούς έχει επικρατήσει ο όρος "linmodems". Επίσης αντίστοιχο πρόβλημα υπάρχει και με τις σύγχρονες κάρτες γραφικών, όπου ο χρήστης για να μπορέσει να εκμεταλλευτεί πλήρως τις 3D ικανότητες της κάρτας του, είναι αναγκασμένος να κατεβάσει τον κατάλληλο οδηγό του κατασκευαστή. Οι ανάλογοι

οδηγοί ελεύθερου λογισμικού περιορίζονται μόνο στην υποστήριξη των 2D ικανοτήτων της κάρτας ή υποστηρίζουν ένα μικρό σύνολο των 3D δυνατοτήτων τους.

Οι περισσότεροι διανομείς του Linux καταρτίζουν σε τακτά διαστήματα έναν κατάλογο συσκευών που είναι "συμβατές" με τη διανομή τους του Linux. Αυτές οι λίστες ονομάζονται "λίστες συμβατότητας υλικού" (Hardware Compatibility Lists) ή HCL για συντομία.

5.4.2 Τα πλεονεκτήματα του LINUX [33]

Μηδενικό κόστος: Το Ubuntu είναι Ελεύθερο Λογισμικό, και είναι δωρεάν. Είναι επίσης Ελεύθερο με την έννοια ότι δίνει όλα τα δικαιώματα που πηγάζουν από την Ελευθερία του Λογισμικού!

Καμία πειρατεία: Το μεγαλύτερο ποσοστό των προγραμμάτων για Linux είναι δωρεάν. Βέβαια, υπάρχουν και εμπορικά προγράμματα, όπως το Nero για Linux, αλλά υπάρχουν αντίστοιχα προγράμματα, τα οποία ίσως να είναι και καλύτερα από τα εμπορικά.

Χαμηλές προδιαγραφές συστήματος: Τα Windows σε κάθε καινούρια έκδοση χρειάζονται καλύτερο υλικό, περισσότερη μνήμη RAM, καλύτερο επεξεργαστή και κάρτα γραφικών, με τις απαιτήσεις να φτάνουν πραγματικά πολύ ψηλά με την κυκλοφορία των Vista. Οι απαιτήσεις του Linux είναι σαφώς χαμηλότερες, κάτι που το κάνει ικανό να τρέξει χωρίς προβλήματα σε υπολογιστές εφτά ή και οχτώ χρόνων.

Πληθώρα προγραμμάτων: Ελάχιστα προγράμματα υπάρχουν προεγκατεστημένα σε έναν υπολογιστή με Windows. Στο Linux όμως, όλες οι διανομές έχουν προεγκατεστημένα πληθώρα προγραμμάτων όπως επεξεργαστή κειμένου, υπολογιστικά φύλλα, πρόγραμμα αλληλογραφίας, φυλλομετρητή, λογισμικό επεξεργασίας εικόνας, προβολή PDF αρχείων και εκπαιδευτικό λογισμικό.

Εύκολη αναζήτηση νέων προγραμμάτων: Δεν υπάρχει δυσκολία αναζήτησης συγκεκριμένων πρόγραμμων, η πλειονότητα των οποίων είναι συγκεντρωμένη στα repositories, και το μόνο που χρειάζεται είναι μια αναζήτηση και η διαδικασία εγκατάστασης είναι εξαιρετικά απλή στις πιο διαδεδομένες διανομές.

Workspaces: Τα Workspaces είναι εικονικές επιφάνειες εργασίας στα οποία μοιράζονται τα ανοικτά παράθυρα, και ο χρήστης επιλέγει το Workspace όπου υπάρχει το πρόγραμμα στο οποίο θέλει να δουλέψει τη συγκεκριμένη στιγμή.

Εύκολη αναβάθμιση: Όλα τα εγκατεστημένα προγράμματα, αλλά και το ίδιο το λειτουργικό αναβαθμίζονται με μεγάλη ευκολία. Το Ubuntu Linux ενημερώνει αυτόματα και ο χρήστης απλά επιλέγει το αν και πότε θα κάνει την αναβάθμιση.

Συνεχής υποστήριξη: Εκτός από την επίσημη υποστήριξη που παρέχουν τόσο η Microsoft για τα Windows, όσο και οι διάφορες εταιρίες που βρίσκονται πίσω από τις μεγάλες διανομές του Linux, το Linux έχει το προτέρημα μιας πολύ δραστήριας κοινότητας ατόμων πρόθυμων να βοηθήσουν κάθε χρήστη που αντιμετωπίζει κάποιο πρόβλημα. Η κοινότητα του Ubuntu είναι ίσως η μεγαλύτερη. Παράλληλα, παρέχεται, προαιρετικά, εταιρική υποστήριξη από την εταιρία Canonical.

Επανεκκίνηση: Στο Linux, οι φορές που ζητείται επανεκκίνηση είναι ελάχιστες και πάντα μετά από κάποια σοβαρή αναβάθμιση, ενώ σπάνια χρειάζεται μετά την εγκατάσταση κάποιας μικρής σημασίας ενημέρωσης.

Ιοί: Ιοί, Trojans, Adware, Spyware δεν υπάρχουν στο Linux. Άρα, αφενός δεν υπάρχει και ανάγκη για Antivirus που καταναλώνει πόρους από το σύστημα, κι αφετέρου το σύστημα είναι πιο ασφαλές λόγω του χαρακτηριστικού αυτού.

5.4.3 Ubuntu Linux

Το Ubuntu [34],[35],[36] είναι ένα ανοικτού κώδικα, ελεύθερο και δωρεάν λειτουργικό σύστημα βασισμένο στον πυρήνα Linux [37]. Το όνομά του προέρχεται από την έννοια ubuntu των Ζουλού και Κόσα (Xhosa), που σημαίνει "Είμαι ότι είμαι λόγω όσων όλοι είμαστε" (humanity towards others). Το Ubuntu ξεκίνησε το 2004, βασισμένο στη διανομή Debian. Ο στόχος του Ubuntu είναι η παροχή ενός διαρκώς ενημερωμένου, σταθερού λειτουργικού συστήματος για τον μέσο χρήστη, με ενισχυμένη έμφαση στην ευκολία χρήσης και εγκατάστασης. Το Ubuntu έχει χαρακτηριστεί ως η πιο δημοφιλής διανομή Linux για επιτραπέζιους υπολογιστές, διεκδικώντας περίπου το 30% επί του συνόλου των Linux συστημάτων σύμφωνα με έρευνα του 2007.

Το Ubuntu είναι ελεύθερο και ανοικτού κώδικα λειτουργικό, που σημαίνει ότι διανέμεται χωρίς χρέωση αλλά και ότι μπορεί να βελτιωθεί από κάθε προγραμματιστή που θέλει να συμμετάσχει στην ομάδα ανάπτυξης. Το Ubuntu χρηματοδοτείται από την Canonical Ltd., μία ιδιωτική επιχείρηση που ιδρύθηκε από τον Νοτιοαφρικανό επιχειρηματία Mark Shuttleworth. Αντί να πωλεί το Ubuntu καθεαυτό, η Canonical καταγράφει έσοδα από την επί πληρωμή τεχνική υποστήριξη

που παρέχει για το προϊόν της. Διατηρώντας το Ubuntu ελεύθερο και ανοικτό η Canonical δέχεται και την βοήθεια τρίτων προγραμματιστών για την ανάπτυξή του. Χρησιμοποιεί επίσης εφαρμογές και κώδικα της διανομής Debian, από την οποία και προέκυψε αρχικά το 2004.

Το σύνθημα του Ubuntu είναι "Linux για ανθρώπους" (Linux for human beings), που περιγράφει τον πρωταρχικό σκοπό – τη δημιουργία μίας Linux διανομής περισσότερο εύκολης στη χρήση από τις υπόλοιπες. Η ευκολία της χρήσης του Ubuntu έχει οδηγήσει, με ορισμένες τροποποιήσεις, στην υιοθέτησή του από τις κυβερνήσεις της Γαλλίας και της πρώην Γιουγκοσλαβικής Δημοκρατίας της Μακεδονίας για χρήση από το κοινό, τους μαθητές και τις υπηρεσίες τους.

5.5 Samba

Σχήμα 18: Το λογότυπο του SAMBA (πηγή: http://samba.org)

Το όλο θέμα της δικτύωσης είναι να επιτρέπει σε υπολογιστές να ανταλλάσουν μεταξύ τους πληροφορίες εύκολα. Αυτή η ανταλλαγή πληροφοριών μεταξύ μηχανημάτων Linux είναι εύκολη χρησιμοποιώντας τα πρωτόκολλα FTP (File Transfer Protocol) και NFS (Network File System), αφού διατίθενται πολλά εύκολα εργαλεία για τα πρωτόκολλα αυτά. Όμως, οι περισσότεροι υπολογιστές στον κόσμο τρέχουν το λειτουργικό σύστημα Windows. Όταν χρειάζεται να συνδεθεί το Linux με τα Windows, αυτό μπορεί να γίνει με Samba [38].

Το Samba [39] είναι ένα σύνολο προγραμμάτων που δίνει σε ένα Linux μηχάνημα την δυνατότητα να επικοινωνεί μέσω του SMB (Server Message Block) με υπολογιστές που τρέχουν Windows. Το SMB είναι το πρωτόκολλο που χρησιμοποιείται για να είναι δυνατή η ανταλλαγή φακέλων και εκτυπωτών μεταξύ υπολογιστών που τρέχουν OS/2 και Windows. Το πρωτόκολλο αυτό είναι ανάλογο με ένα συνδυασμό του NFS (Network File System), του lpd (ο τυπικός τρόπος εκτύπωσης από το UNIX) και ενός συστήματος ταυτοποίησης, όπως είναι τα NIS και Kerberos.

Καθώς τρέχει το Samba στο Linux μηχάνημα τότε αυτό εμφανίζεται στο "Network Neighborhood" -δηλαδή στους γείτονές του- σαν να ήταν άλλο ένα Windows μηχάνημα. Οι χρήστες των Windows μηχανημάτων μπορούν να συνδεθούν με μηχανήματα που τρέχουν Linux, και ανάλογα με τα δικαιώματα πρόσβασης που έχουν, να ανταλλάξουν αρχεία και να δώσουν εντολές εκτύπωσης. Η ταυτοποίηση των χρηστών δεν αφήνεται μόνο στο λειτουργικό σύστημα Linux, αλλά παρέχεται απευθείας από το Samba.

Το Samba είναι διαθέσιμο δωρεάν σύμφωνα με την GNU κοινόχρηστη άδεια χρήσης όπως είναι και το Linux. Ο συνδυασμός Linux και Samba είναι μια πολύ καλή και φθηνή λύση εναλλακτική αυτής που προτείνει η Microsoft (στηριζόμενη στην τεχνολογία NT) στο διαμοιρασμό πόρων και στην δικτυακή ταυτοποίηση χρηστών.

Για να καταλάβουμε ευκολότερα πως δουλεύει το Samba πρέπει να ξέρουμε λίγα πράγματα για το πώς δουλεύει η δικτύωση των Windows. Οι πελάτες Windows για να έχουν πρόσβαση σε αρχεία και εκτυπωτές που βρίσκονται σε έναν Server μεταδίδουν "Server Message Block" χρησιμοποιώντας μια συνεδρία (session) NetBIOS πάνω από το πρωτόκολλο TCP/IP, γεγονός που έχει πολλαπλά πλεονεκτήματα.

Το TCP/IP υπάρχει πλέον σε κάθε λειτουργικό σύστημα. Συνεπώς είναι πολύ εύκολο να συνδέσεις μέσω του Samba έναν υπολογιστή που τρέχει Linux με υπολογιστές που τρέχουν Windows ή άλλα συστήματα που υποστηρίζουν το TCP/IP. Επίσης η χρησιμοποίηση του TCP/IP καθιστά το Samba εύκολο να συνδεθεί με μεγάλα δίκτυα TCP/IP, όπως είναι το διαδίκτυο. Αναγνωρίζοντας αυτά τα πλεονεκτήματα, η Microsoft μετονόμασε τον συνδυασμό του SMB και NetBIOS πάνω από TCP/IP ως Common Internet Filesystem (CIFS) και επιδιώκει να γίνει αποδεκτό το CIFS ως το πλέον κοινό διαδικτυακό πρωτόκολλο για μεταφορά αρχείων.

Ο Samba Server αποτελείται από δύο επιμέρους servers: smbd και nmbd. Ο smbd server είναι ο πυρήνας του Samba. Πραγματοποιεί τις συνδέσεις, επαληθεύει τους πελάτες και παρέχει πρόσβαση στο σύστημα αρχείων και στους εκτυπωτές. Από την άλλη, ο nmbd server είναι υπεύθυνος για το "network browser". Ο ρόλος του είναι να διαφημίζει τις υπηρεσίες που καλείται ο Samba Server να παρέχει. Στον nmbd server οφείλεται η εμφάνιση του υπολογιστή στο "Network Neighborhood"

των Windows NT και αυτός επιτρέπει στους χρήστες να έχουν μια λίστα με τους διαθέσιμους πόρους.

Το Samba περιλαμβάνει και το εργαλείο smbclient, που είναι ένας πελάτης του πρωτοκόλλου SMB που τρέχει από την κονσόλα. Είναι παρόμοιο με το FTP και επιτρέπει την ανταλλαγή αρχείων με SMB servers και την πρόσβαση σε πόρους SMB εκτυπωτών.

Η παραμετροποίηση του Samba στις εκάστοτε ανάγκες μας στηρίζεται στο αρχείο smb.conf (samba's configuration file). Όπως σε όλες τις εφαρμογές του Linux είναι πολύ εύκολο να κάνεις μια διαφορετική διαμόρφωση του Samba τροποποιώντας το αρχείο smb.conf. Ανάλογα με τις ρυθμίσεις για το δίκτυο μας μπορούμε να μεταβάλλουμε πάνω από 170 διαφορετικές παραμέτρους τους smb.conf. Επίσης, είναι εύκολο να ανιχνευτούν λάθη στο αρχείο smb.conf εκτελώντας την εντολή testparm. Τέλος, με την εντολή smbpasswd δίνουμε στον samba server ένα έγκυρο ζεύγος username / password, προκειμένου να προσθέσει χρήστες, ή υπολογιστές χρήστες (στην περίπτωση που θέλουμε να περιορίσουμε τους υπολογιστές που θα μπορούν να ταυτοποιούνται). Οι χρήστες αποθηκεύονται σε ένα ξεχωριστό αρχείο με όνομα smbpasswd σε κωδικοποιημένη μορφή, ενώ πρέπει να είναι ταυτόχρονα και χρήστες του λειτουργικού Linux.

Το αρχείο smb.conf περιλαμβάνει τομείς όπως τους [global], [homes] or [printers]. Κάθε παράμετρος του smb.conf είναι είτε global parameter, που σημαίνει ότι επηρεάζει τον διακομιστή συνολικά, είτε service parameter, που σημαίνει ότι επηρεάζει την αντίστοιχη υπηρεσία. Το [homes] είναι ένας ειδικός τομέας για παροχή φακέλων που αντιστοιχίζονται δυναμικά στον προσωπικό χώρο του κάθε χρήστη (user's home directory). Το [printers] είναι ένας τομέας που παρέχει ένα εύκολο τρόπο για να διαμοιραστεί κάθε εκτυπωτής που περιγράφεται στο αρχείο εκτυπωτών του συστήματος (system's printcap file). Τόσο [homes] το όσο και το [printers] έχουν τις δικές τους παραμέτρους για τον πλήρη έλεγχό τους και την δική τους ασφάλεια.

Από όλα τα παραπάνω γίνεται σαφές ότι το Samba είναι ένα εργαλείο που γεφυρώνει το κενό μεταξύ των λειτουργικών Linux και Windows. Επιπλέον, το Samba είναι κλασσικό παράδειγμα δωρεάν λογισμικού με γνωρίσματα που συγκρίνονται με αντίστοιχες εμπορικές εφαρμογές. Το Samba είναι ένα σύστημα πολύ καλά υποστηριζόμενο και συνεχώς εξελισσόμενο.

5.6 CUPS

Σχήμα 19: Το λογότυπο του CUPS (πηγή: http://cups.org)

Τα συστήματα εκτύπωσης στο UNIX ανέκαθεν χαρακτηριζόταν από κάποια σύγχυση. Η ρύθμιση για κάθε εκτυπωτή ήταν διαφορετική και συχνά απαιτούσε αρκετές ώρες επεξεργασίας δυσνόητων αρχείων ρυθμίσεων. Συν τοις άλλοις, τις περισσότερες φορές δεν υπήρχε υποστήριξη για τους νεώτερους και φτηνότερους εκτυπωτές inkjet, μόνο για ακριβούς εκτυπωτές Postscript. Ακόμη και όταν υπήρχε όμως, η ποιότητα ήταν σαφώς κατώτερη από την αντίστοιχη των οδηγών για άλλα λειτουργικά συστήματα (π.χ. Windows).

Για το σκοπό αυτό, αναπτύχθηκε το CUPS (Common Unix Printing System) [40], το οποίο και είναι πλέον το προκαθορισμένο σύστημα εκτύπωσης σε όλα τα σύγρονα UNIX συστήματα, ακόμη και στο MacOS X. Αναπτύχθηκε από την Easy Software Products [41] για να προωθήσει μια στάνταρ λύση εκτυπώσεων για όλους τους χρήστες και κατασκευαστές UNIX υποστηρίζοντας τους περισσότερους PostScript και raster εκτυπωτές. Το CUPS, παρέχει μια πιο σύγχρονη αντιμετώπιση της διαχείρισης ενός εκτυπωτής και υποστηρίζει όλα τα σύγχρονα και μή συστήματα εκτύπωσης, όπως τοπικούς και απομακρυσμένους δικτυακούς εκτυπωτές, LPD (Line Printer Daemon) εκτυπωτές, IPP (Internet Printing Protocol ή Πρωτόκολλο Εκτύπωσης Διαδικτύου), IPP/HTTP, SMB (Server Message Block) και AppSocket (γνωστό ως JetDirect), καθώς και τοπικούς εκτυπωτές συνδεδεμένους παράλληλα, σειριακά ή στη θύρα USB, κλπ. Χρησιμοποιεί πρότυπα αρχεία PPD (Postscript Printer Definition) που στην ουσία ρυθμίζουν το σύστημα στις προδιαγραφές του κάθε εκτυπωτή, και αυτή τη στιγμή δύσκολα θα βρείτε εκτυπωτή που να μην υποστηρίζεται, καθώς ο αριθμός των υποστηριζόμενων εκτυπωτών ξεπερνάει τις

3000. Ακόμη υποστηρίζει απευθείας εκτύπωση μιας σελίδας σε αρχείο Postscript ή PDF για περαιτέρω επεξεργασία.

Η ρύθμιση και διαχείριση του συστήματος και των εκτυπωτών μπορεί να γίνει μέσω ενός browser, ενώ παρέχει και εύκολους τρόπους εγκατάστασης νέου εκτυπωτή με βήματα. Προσφέρει, συμβατότητα με τα υπάρχοντα συστήματα εκτύπωσης, όπως LPD, LPRng, μέσω ειδικών εντολών wrappers και πλήρη συνεργασία με γραφικά περιβάλλοντα όπως το KDE και GNOME. Γενικά, είναι ένα πολύ δυνατό σύστημα και πολύ πιο εύκολο στη χρήση του από τα παλαιότερα συστήματα.

Τέλος, υπάρχει δυνατότητα χρήσης ενός εκτυπωτή που διαμοιράζεται από ή προς ένα δίκτυο Windows. Χρησιμοποιώντας το σύστημα Samba και την εντολή smbprint (που περιέχεται στο πακέτο smbclient), δημιουργείται μια καταχώρηση στο αρχείο /etc/printcap για τον κάθε εκτυπωτή και στην συνέχεια με την εντολή lpr μπορούμε να εκτυπώσουμε. Ο υπολογιστής που τρέχει το CUPS λειτουργεί σαν ένας Printer Server που δέχεται εντολές εκτύπωσης από πελάτες, τις δρομολογεί και τις στέλνει στον κατάλληλο εκτυπωτή.

5.7 Apache

Σχήμα 20: Το λογότυπο του Apache (πηγή: http://apache.org)

Ο Apache HTTP Server είναι ένας εξυπηρετητής ιστού για συστήματα τύπου Unix, Microsoft windows, Novell και άλλα. Από την πρώτη εμφανισή του, στις αρχές του 1995, αποτέλεσε τον μόνο αξιόπιστο εξυπηρετητή ιστού ανοιχτού κώδικα, μαζί με τον εξυπηρετητή Sun Java System Web server της Sun. Η χρήση του αυξανόταν διαρκώς τόσο από απλούς χρήστες όσο και από επιχειρήσεις και επιστημονικούς οργανισμούς, με αποτέλεσμα σήμερα να θεωρείται στάνταρ σύγκρισης για όλους τους εξυπηρετητές ιστού. Σύμφωνα με στοιχεία για τον Φεβρουάριο του 2007, ο Apache είναι ο δημοφιλέστερος εξυπηρετητής, με ποσοστό 58,70%. Υπεύθυνος οργανισμός για την ανάπτυξη του είναι ο Apache Software Foundation και άδεια

χρήσης του είναι μια ειδικά διαμορφωμένη άδεια ανοιχτού κώδικα με την ονομασία Apache Licence 2.

Ως προς τα τεχνολογικά χαρακτηριστικά του, ο Apache εξυπηρετεί τόσο στατικές όσο και δυναμικές ιστοσελίδες. Η λειτουργικότητα που προσφέρει αποτελεί de facto βάση για την εγκατάσταση και λειτουργία πολλών δικτυακών συστημάτων κι εφαρμογών. Συχνά δε, χρησιμοποιείται και για την διανομή περιεχομένου μέσω ιστού, με τρόπο ασφαλή και αξιόπιστο, ή για την αποσφαλμάτωση (debugging) δικτυακών εφαρμογών στον τοπικό υπολογιστή του προγραμματιστή. Ο ίδιος ο Apache περιλαμβάνει ένα πλήθος μεταγλωττισμένων τμημάτων κώδικα (modules), τα οποία επεκτείνουν τα λειτουργικά χαρακτηριστικά του πυρήνα. Παράδειγμα τέτοιων είναι αυτά που προσθέτουν υποστήριξη γλωσσών προγραμματισμού όπως η Perl, η Python και η PHP. Δίνεται επίσης η δυνατότητα της εικονικής φιλοξενίας (virtual hosting), κατά την οποία ένα μηχάνημα με εγκατεστημένο τον Apache HTTP Server μπορεί να εξυπηρετεί ταυτοχρόνως πολλές διαφορετικές ιστοσελίδες, με διαφορετικά ονόματα τομέα (domain names). Ένα άλλο εξελιγμένο τεχνολογικό χαρακτηριστικό είναι η διαπραγμάτευση περιεχομένου (content negotiation), η οποία επιτρέπει σε διαφορετικές εκδόσεις ενός δικτυακού εγγράφου, σε μια συγκεκριμένη ιστοθέση, να προβάλλονται, ανάλογα με τις δυνατότητες και τις ρυθμίσεις του υπολογιστή του τελικού χρήστη.

Πέρα των τεχνολογικών χαρακτηριστικών του και της διάδοσής του, οι λόγοι, οι οποίοι οδήγησαν στην επιλογή του Apache για τους σκοπούς της υλοποίησης έναντι κάποιου άλλου HTTP εξυπηρετητή είναι [42]:

- Η **άδεια χρήσης** του: Η άδεια Apache Licence είναι σύμφωνη με της βασικές αρχές των αδειών ανοιχτού/ελεύθερου κώδικα, αφού αποτελεί βασικά μια παραλλαγή της GNU/GPL. Έχει ήδη αναφερθεί ότι στην παρούσα υλοποίηση θα προτιμώνται εφαρμογές ανοιχτού κώδικα κατά το δυνατόν.
- Η **άψογη συνεργασία** του με δικτυακές εφαρμογές σε PHP και MySQL βάσεις: Δεδομένου ότι το Joomla είναι εξ ολοκλήρου γραμμένο σε PHP και απαιτεί τη δημιουργία μιας βάσης MySQL για τα δεδομένα του, ο Apache HTTP Server αποτελεί τη δημοφιλέστερη λύση στην κοινότητα των χρηστών του Joomla.

- Μικρές απαιτήσεις σε **μνήμη**: Ο Apache είναι αποδοτικότερος και οικονομικότερος σε απαιτήσεις μνήμης, για τις μικρές υπολογιστικές απαιτήσεις μιας πύλης σε PHP, σε σχέση με άλλες λύσεις, όπως ο Tomcat ή ο Java System Web Server, που προάγουν τη χρήση Java Servlets.
- Παρέχει δυνατότητες **ελέγχου** του μέσω της **γραμμής εντολών** έτσι καθίσταται πολύ εύκολη η παραμετροποίηση και ο δυναμικός έλεγχος του όλου συστήματος.
- Οι γλώσσες σεναρίου (**scripting languages**) που είναι διαθέσιμες για τον έλεγχο του Apache παρέχουν πάρα πολλές δυνατότητες σε αντίθεση με αυτές σε άλλους HTTP εξυπηρετητές.
- Επιτρέπει την χρήση της Perl και εντολών του φλοιού με πολύ απλό τρόπο για την κάλυψη των κάθε είδους δικτυακών απαιτήσεων.
- Τα **αρχεία καταγραφής** της πρόσβασης (access log files) είναι πλήρη χωρίς απώλειες καταγραφής ενεργειών. Έτσι, η παρακολούθηση του συστήματος πραγματοποιείται με αξιοπιστία και ασφάλεια.

5.8 PHP

Σχήμα 21: Το λογότυπο της PHP (πηγή: http://php.net)

Η PHP είναι γενικού σκοπού γλώσσα συμβάντων, σχεδιασμένη ειδικά για το Web [43],[44]. Η κύρια χρήση της συνίσταται στη δημιουργία scripts (συμβάντων) για δυναμικές ιστοσελίδες κι όχι γλώσσα προγραμματισμού. Αυτό σημαίνει ότι είναι σχεδιασμένη ώστε να εκτελεί μια ενέργεια μετά από κάποιο συμβάν, όπως για παράδειγμα αν ο χρήστης πατήσει κάποιο link στην ιστοσελίδα. Επιπλέον η Php λειτουργεί στην πλευρά του Server, δηλαδή εγκαθίσταται στον Server και τα script που είναι γραμμένα σε αυτή χρησιμοποιούν πόρους απ τον υπολογιστή-Server για την εκτέλεσή τους και τα αποτελέσματα της εκτέλεσης στέλνονται στον client σε μορφή

html. Επιπλέον ο κώδικας της Php παρεμβάλλεται σε κώδικα HTML (με κατάλληλη σήμανση στην αρχή και το τέλος του κώδικα Php). Ο κώδικας της Php δεν εκτελείται αυτόνομα αλλά ταυτόχρονα (γραμμή προς γραμμή) με τον κώδικα της html. Μπορεί να χρησιμοποιηθεί και για command line scripting με τη βοήθεια του κατάλληλου μεταγλωττιστή όπως και για εγγραφή client-side GUI εφαρμογών. Αυτή η δυνατότητά της χρησιμοποιήθηκε εκτενώς για το server.php καθώς και το cron script του συστήματος RAST.

Τα πλεονεκτήματα της PHP που οδήγησαν στη επιλογή της περιγράφονται παρακάτω [45], [46]:

- **Κόστος**: Η PHP είναι γλώσσα ανοικτού κώδικα, που σημαίνει ότι ο πηγαίος κώδικας είναι διαθέσιμος σε όλους για χρήση, για τροποποίηση και αναδιανομή χωρίς κάποιο κόστος.
- **Υποστήριξη**: Η Zend, η εταιρία που υποστηρίζει την PHP, χρηματοδοτεί την ανάπτυξη της, προσφέροντας υποστήριξη και σχετικό λογισμικό σε εμπορική βάση.
- **Συμβατότητα**: Η PHP είναι διαθέσιμη σε πολλά λειτουργικά συστήματα και συνήθως ο κώδικας δουλεύει χωρίς αλλαγές σε διαφορετικά λειτουργικά συστήματα που τρέχουν την PHP. Παραδείγματα λειτουργικών συστημάτων στα οποία λειτουργεί η Php είναι τα Windows, το Linux, FreeBSD, Solaris, IRIX.
- **Απόδοση**: Η PHP είναι πολύ αποδοτική. Με ένα φθηνό διακοσμητή μπορούμε να εξυπηρετήσουμε εκατομμύρια επισκέψεων σε ημερήσια βάση.
- **Υποστήριξη Βάσεων Δεδομένων**: Η PHP υποστηρίζει συνδέσεις με πολλά γνωστά συστήματα βάσεων δεδομένων όπως: MySQL, PostgreSQL, mSQL, Oracle, dbm, filepro, Hyperwave, Informix, InterBase, Sybase και αλλες. Έχει επίσης ενσωματωμένη SQL διασύνδεση στο επίπεδο αρχείο SQLite. Με τη χρήση του standard ODBC μπορεί να συνδεθεί σε οποιαδήποτε βάση έχει πρόγραμμα οδήγησης ODBC (όπως π.χ. Τα προϊόντα της Microsoft).
- **Ενσωματωμένες Βιβλιοθήκες**: Η PHP έχει πολλές ενσωματωμένες βιβλιοθήκες που εκτελούν πολλές χρήσιμες λειτουργίες. Δυναμική

δημιουργία εικόνων GIF, σύνδεση με άλλες υπηρεσίες δικτύων, ανάλυση XML, αποστολή e-mail, δημιουργία εγγράφων PDF.

- **Αντικειμενοστραφής υποστήριξη**: Στην PHP υπάρχουν οι γνωστές από τη C++ και Java αντικειμενοστραφείς λειτουργίες όπως η κληρονομικότητα, οι ιδιωτικές και προστατευμένες ιδιότητες και μέθοδοι, οι αφηρημένες κλάσεις και μέθοδοι, οι διασυνδέσεις, οι συναρτήσεις δημιουργίας, αποδιάρθρωση. Επίσης υπάρχουν και άλλες λιγότερο δημοφιλείς λειτουργίες.
- **Ευκολία Εκμάθησης**: Η σύνταξη της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού, κυρίως στην C και στην Perl. Η γνώση μιας γλώσσας προγραμματισμού της οικογένειας της C επιτρέπουν σε κάποιον να ξεκινήσει αμέσως τον προγραμματισμό στην PHP. Έχει εύκολη σύνταξη και είναι πολύ ευέλικτη.

5.9 PERL

Σχήμα 22: Το λογότυπο της PERL (πηγή: http://perl.org)

Η Perl είναι γενικού σκοπού γλώσσα [47]. Η κύρια χρήση της συνίσταται στο command line scripting δηλαδή στη δημιουργία scripts με σκοπό την καλύτερη παραμετροποίηση του Linux. Τα πλεονεκτήματα της Perl που οδήγησαν στη επιλογή της περιγράφονται παρακάτω [48]:

1. Η Perl υποστηρίζει υποτυπώδες CGI και είναι η γλώσσα προγραμματισμού που έκανε το CGI δημοφιλές στο διαδίκτυο.
2. Η mod_perl είναι μια εξελιγμένη εφαρμογή της Perl που τρέχει στον Apache web server. Παρέχει εξαιρετικά γρήγορη απόδοση και πλήρη πρόσβαση στις εσωτερικές ρυθμίσεις του Apache.
3. Αλληλεπίδραση με βάση δεδομένων: Η Perl παρέχει μια εξαιρετική διεπαφή με όλες σχεδόν τις βάσεις δεδομένων, καθώς και ένα αφαιρετικό στρώμα που επιτρέπει να αλλάξεις βάση δεδομένων χωρίς σημαντικές αλλαγές στον κώδικα.
4. Αρχιτεκτονική επαναχρησιμοποίησης κώδικα: Η αρχιτεκτονική της Perl επιτρέπει και ενθαρρύνει την επαναχρησιμοποίηση του κώδικα.

5. CPAN (Comprehensive Perl Archive Network), είναι μία από τις μεγαλύτερες βιβλιοθήκες ελεύθερου κώδικα στον κόσμο. Έτσι, παρέχονται δυνατότητες και λύσεις για πλήθος λειτουργιών μέσω έτοιμων λύσεων, που παρέχονται δωρεάν.

6. Πολυχρηστικότητα : Η Perl μπορεί να χρησιμοποιηθεί στην ανάπτυξη λογισμικού δικτυακές εφαρμογές, υπολογισμούς, ανάλυση δεδομένων , χειρισμό κειμένου, εργαλεία και εφαρμογές κονσόλας, και τέλος γραφικές εφαρμογές.

7. Μπορεί να συνεργαστεί με C, C++, Java κτλ.

8. Είναι συμβατή με διάφορα περιβάλλοντα όπως Linux, MS Windows και άλλα.

5.10 MySQL

Σχήμα 23: Το λογότυπο της MySQL (πηγή: http://mysql.com)

Η MySQL [49] είναι ένα, ανοικτού κώδικα, σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Το σύστημα διαχείρισης MySQL λοιπόν δίνει τη δυνατότητα της αποθήκευσης, αναζήτησης, ταξινόμησης, ομαδοποίησης, ανάκλησης δεδομένων με βάση τη γλώσσα ερωτημάτων SQL. Το γεγονός ότι η MySQL είναι σχεσιακή συνεπάγεται ότι η οργάνωση των δεδομένων γίνεται σε διαφορετικούς πίνακες οι οποίοι σχετίζονται μεταξύ τους με κάποιο σαφώς ορισμένο τρόπο. Η MySQL επιπλέον δύναται να ελέγχει την πρόσβαση στα δεδομένα, εξασφαλίζοντας έτσι τη δυνατότητα η πρόσβαση να γίνεται από διαφορετικούς χρήστες. Κάθε χρήστης έχει συγκεκριμένα δικαιώματα πάνω στις βάσεις δεδομένων που του τα δίνει η MySQL.

Τα πλεονεκτήματα της MySQL που οδήγησαν στη επιλογή της περιγράφονται παρακάτω [50]:

- **Απόδοση**. Η MySQL είναι αρκετά γρήγορη και αξιόπιστη. Πολλές δοκιμές που έχουν γίνει δείχνουν ότι υπερέχει σε ταχύτητα έναντι των ανταγωνιστών της.
- **Κόστος**. Η MySQL είναι προϊόν ανοικτού κώδικα και διατίθεται δωρεάν για προσωπική χρήση. Η εμπορική άδεια της διατίθεται σε χαμηλό κόστος.
- **Ευκολία Χρήσης**. Η MySQL είναι αρκετά εύκολη στην εκμάθηση της, ακόμα και για κάποιον που δεν έχει ξαναχρησιμοποιήσει παρόμοια προϊόντα κατασκευής βάσεων δεδομένων.
- **Συμβατότητα**. Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά σύγχρονα λειτουργικά συστήματα. είναι συμβατή με πολλές εκδόσεις των Microsoft Win-dows και με λειτουργικά Unix, όπως οι διάφορες εκδόσεις του δημοφιλούς λειτουργικού ανοικτού κώδικα Linux.
- **Υποστήριξη**. Στην σελίδα www.mysql.com υπάρχει μια τεράστια υποστήριξη πάνω στη MySQL με manual, tutorial, βοήθεια σε πιθανά προβλήματα. Υπεύθυνη για την ανάπτυξή της είναι η εταιρία MySQL AB, η οποία αποτελεί μια από τις μεγαλύτερες εταιρίες ανάπτυξης ανοιχτού λογισμικού.
- Υποστηρίζει τη λειτουργία πολλών **νημάτων** (multithread) και πολλών χρηστών (multiuser). Αυτό σημαίνει ότι μπορεί να χρησιμοποιεί πολλαπλούς επεξεργαστές εφόσον είναι διαθέσιμοι.
- Υποστηρίζει πολλά και διαφορετικά **API** : C, C++, Eiffel, Java, Perl, Python, Tcl.
- Υποστηρίζει πολλούς και διαφορετικούς **τύπους δεδομένων**: FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM.
- Υποστηρίζει με **απλό** και **λειτουργικό τρόπο** την χρήση των βασικών εντολών SELECT και WHERE.
- Πραγματοποιεί πολύ γρήγορες **συνδέσεις** (joins) χρησιμοποιώντας μια βελτιστοποιημένη σύνδεση.
- Μπορεί ο χρήστης να καλέσει πίνακες από διαφορετικές βάσεις δεδομένων στην ίδια δήλωση.
- Υποστηρίζει πλήρως τις προτάσεις **GROUP BY** και **ORDER BY**.
- Είναι γραμμένη σε C και C++ και μεταφρασμένη με πολλούς και διαφορετικούς μεταγλωττιστές.

- Διαθέτει πίνακες στην προσωρινή μνήμη, που χρησιμοποιούνται σαν **προσωρινοί πίνακες**.
- Μπορεί να διαχειριστεί **μεγάλες βάσεις δεδομένων**. Χρησιμοποιούνται mySQL βάσεις με 50.000.000 φακέλους και 60.000 πινάκες.
- Όλες οι στήλες διαθέτουν **προκαθορισμένες τιμές**.
- Δεν έχει **διακοπές μνήμης**.
- Παρέχει τη δυνατότητα στον χρήστη να **συνδεθεί** σε έναν εξυπηρετητή με mySQL χρησιμοποιώντας TCP/IP Sockets, Unix Sockets,Named Pipes.

5.11 FAUS

Σχήμα 24: Το λογότυπο του FAUS (πηγή: http://faus.sourceforge.net)

Το FAUS είναι ένα Perl CGI που επιτρέπει την διαχείριση χρηστών μέσω μιας δικτυακής διεπαφής [51].

Κύρια χαρακτηριστικά του FAUS [52]:

- Μπορεί να διαχειριστεί **χρήστες** τόσο για το **UNIX** όσο και για το **SAMBA** με μία μόνο εντολή.
- Δεν χρησιμοποιεί τον χρήστη root ή το suid πρόγραμμα για να εκτελέσει λειτουργίες στα αρχεία /etc/passwd και smbpasswd (δηλαδή τα αρχεία καταγραφή των χρηστών). Αντίθετα, χρησιμοποιεί το "**Sudo**" που παρέχει τα απαραίτητα δικαιώματα στον Apache για να εκτελέσει συγκεκριμένα scripts ως root. Τα scripts αυτά έχουν περιορισμένη εμβέλεια και δεν θα επιτρέψουν ενέργειες που μπορεί να οδηγήσουν στην κατάρρευση του συστήματος.
- Είναι εφικτό να γίνει χρήση **διαφορετικών φορμών ταυτοποίησης** χρησιμοποιώντας τον Apache, παρέχοντας και την δυνατότητα σύνδεσης SSL.

- **Πολυγλωσσική υποστήριξη**: όλα τα μηνύματα και οι log πληροφορίες μπορούν να υπάρξουν σε μορφή απλού text κειμένου και ταυτόχρονα και HTML κώδικα.
- Υποστήριξη **Log**: όλες οι λειτουργίες καταγράφονται λεπτομερώς στο αρχείο /var/log/httpd/errors.log με τον ίδιο τρόπο που καταγράφονται και τα μηνύματα του Apache.

Μεγάλο κομμάτι της λειτουργικότητας του FAUS χρησιμοποιείται στο σύστημα RAST. Για το λόγο αυτό χρησιμοποιήθηκαν κομμάτια κώδικα από το FAUS, τα οποία με τις κατάλληλες τροποποιήσεις εντάχθηκαν στο RAST.

6

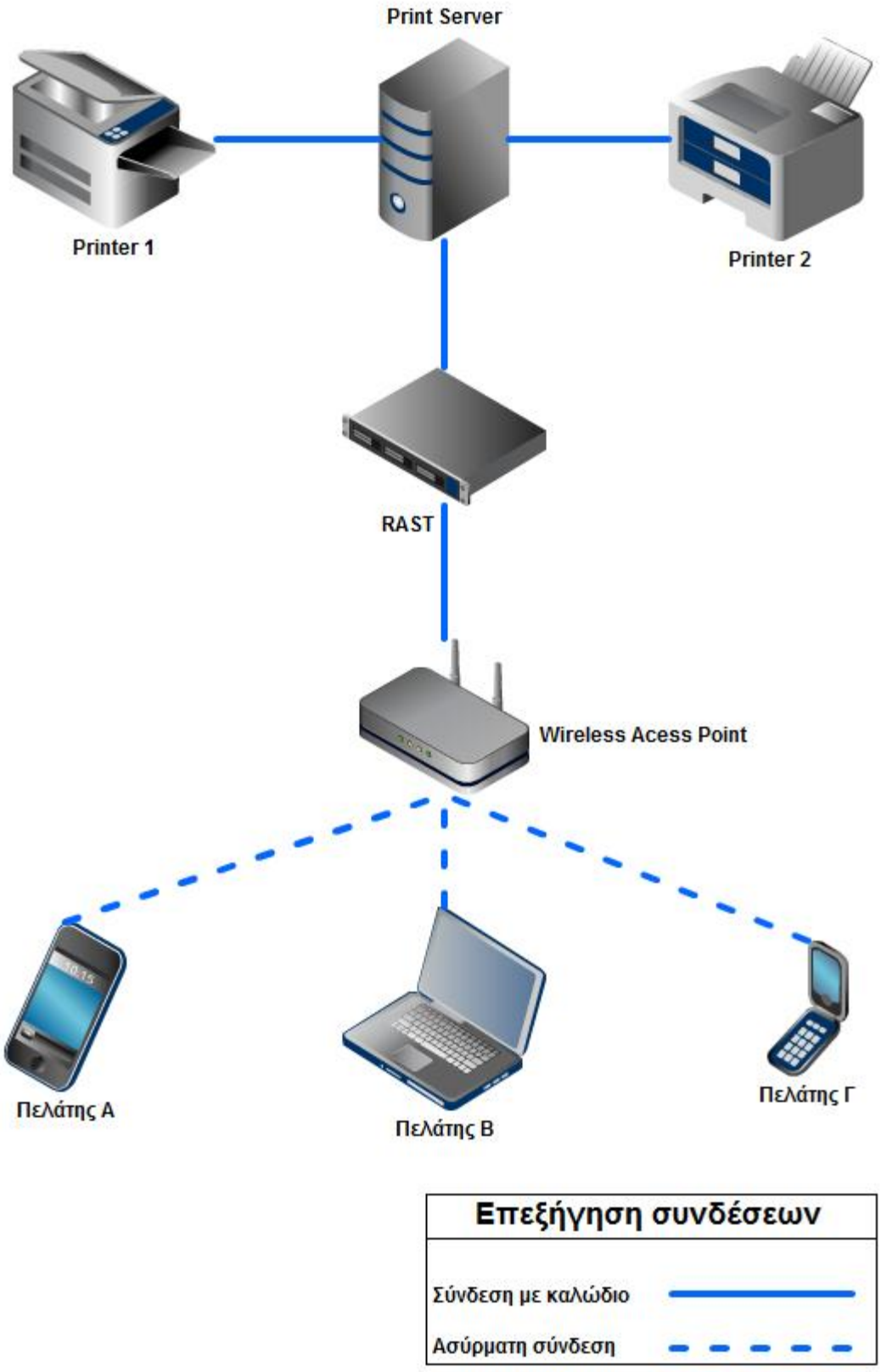
Παραδείγματα εφαρμογών του συστήματος

Στο παρόν κεφάλαιο θα καλυφθούν ορισμένα παραδείγματα – σενάρια χρήσης του συστήματος RAST, ώστε να αναδειχθούν τα πεδία εφαρμογής που θα μπορούσε πιθανώς να έχει.

6.1 Εταιρία υπηρεσιών εκτύπωσης (Print Shop)

Στην ενότητα αυτή θα εξετάσουμε το σενάριο της (υποθετικής) εταιρίας XYZ Printing. Η εταιρία έχει εντός του χώρου της δυο εκτυπωτές, τους οποίους και θέλει να δώσει προς χρήση σε πελάτες που βρίσκονται στο κατάστημα. Για να γίνει αυτό, θα θεωρήσουμε ότι η εταιρία διαθέτει και διακομιστή εκτύπωσης για τους δυο εκτυπωτές.

Η εταιρία λοιπόν προμηθεύεται ένα σύστημα RAST, και αφού το έχει εγκαταστήσει για λειτουργία στο δικό της δίκτυο, εγκαθιστά κι έναν ασύρματο σταθμό βάσης Wi-Fi (wireless access point). Έτσι, ο διαχειριστής του RAST συνδέει με το σύστημα τους δυο αυτούς εκτυπωτές. Το σχήμα λοιπόν του σεναρίου αυτού φαίνεται παρακάτω:



Σχήμα 25: Σχήμα σεναρίου καταστήματος εκτύπωσης

Θα θεωρήσουμε περεταίρω ότι η χρήση των εκτυπωτών γίνεται με την ώρα, και οι χρήστες χρεώνονται επιπλέον με τον όγκο του χαρτιού που εκτύπωσαν.

Έτσι, αν έρθουν τρεις πελάτες, όπως φαίνεται και στο σχήμα, πληρώνουν τη διάρκεια χρήσης του εκτυπωτή, και ο διαχειριστής τους εκδίδει λογαριασμούς πρόσβασης με χρονική διάρκεια ίση με αυτή που πλήρωσαν οι πελάτες. Με το πέρας του χρόνου αυτού κάθε πελάτης μπορεί να ανανεώσει το λογαριασμό του, ή σε αντίθετη περίπτωση διαγράφεται ο λογαριασμός και η πρόσβαση διακόπτεται.

Με τον τρόπο αυτό, και με την βοήθεια του συστήματος RAST αναδεικνύεται η δυνατότητά του για χρονικά περιορισμένες προσβάσεις σε πόρους.

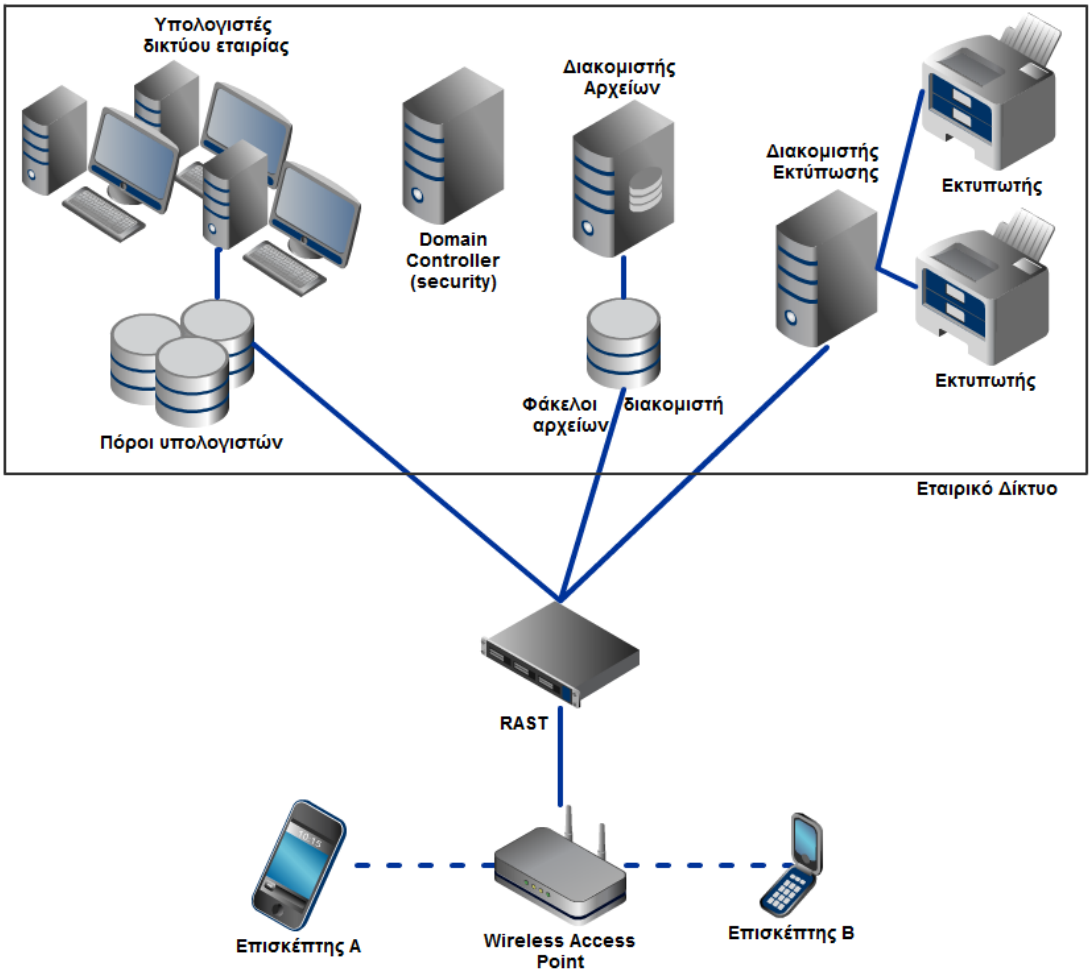
6.2 Πρόσβαση επισκεπτών σε πόρους εταιρικού δικτύου

Μέσω κατάλληλης εγκατάστασης και χρήσης, το RAST μπορεί να εξυπηρετήσει έναν οργανισμό ή μια μεγάλη εταιρία να ανταλλάσσει πληροφορίες με επισκέπτες ή/και ακόμα να τους δίνει πρόσβαση σε πόρους και υπηρεσίες της εταιρίας. Το συγκριτικό πλεονέκτημα του RAST είναι ότι το δίκτυο του οργανισμού δεν χρειάζεται να αλλοιωθεί ή να επηρεαστεί στο ελάχιστο για να παρέχει αυτήν την υπηρεσία. Αρκεί να συνδεθεί στο ίδιο δίκτυο με τους διακομιστές του οργανισμού που κατέχουν τους προς διαμοιρασμό πόρους.

Για το συγκεκριμένο παράδειγμα θα θεωρήσουμε ότι ο οργανισμός έχει εγκατεστημένο windows domain, αλλά δεν επιθυμεί να δημιουργήσει λογαριασμούς για τους επισκέπτες (οι λόγοι που κάτι τέτοιο είναι επιθυμητό θα αναλυθούν παρακάτω). Επιπλέον όμως, επιθυμεί να δώσει πρόσβαση στους επισκέπτες σε γενικό πληροφοριακό υλικό. Σε ορισμένους μάλιστα, που για παράδειγμα μετέχουν κάποιας συνάντησης, θέλουν κατ' εξαίρεση να τους παραχωρήσουν πρόσβαση σε εξειδικευμένα για την παρουσίαση αρχεία.

Για την εφαρμογή αυτή, θα μπορούσε να γίνει μια εγκατάσταση του συστήματος RAST. Έτσι, όλοι οι απαραίτητοι για τους επισκέπτες πόροι αρκεί να μοιραστούν εντός του δικτύου, και να ενταχθούν εντός του RAST. Έτσι, ο διαχειριστής μπορεί να παρέχει σε κάθε επισκέπτη στοιχεία πρόσβασης, με τα οποία ο επισκέπτης μπορεί να δει τα διαμοιραζόμενα αρχεία της εταιρίας, χωρίς να δημιουργηθεί για αυτόν ειδικός λογαριασμός στο domain του οργανισμού. Επιπλέον, σε επιλεγμένους επισκέπτες, όπως για παράδειγμα τους μετέχοντες μιας συνάντησης, μπορεί να δοθεί ειδική πρόσβαση σε αρχεία που αφορούν τη συγκεκριμένη συνάντηση.

Ακολουθεί ένα διάγραμμα για το συγκεκριμένο σενάριο:

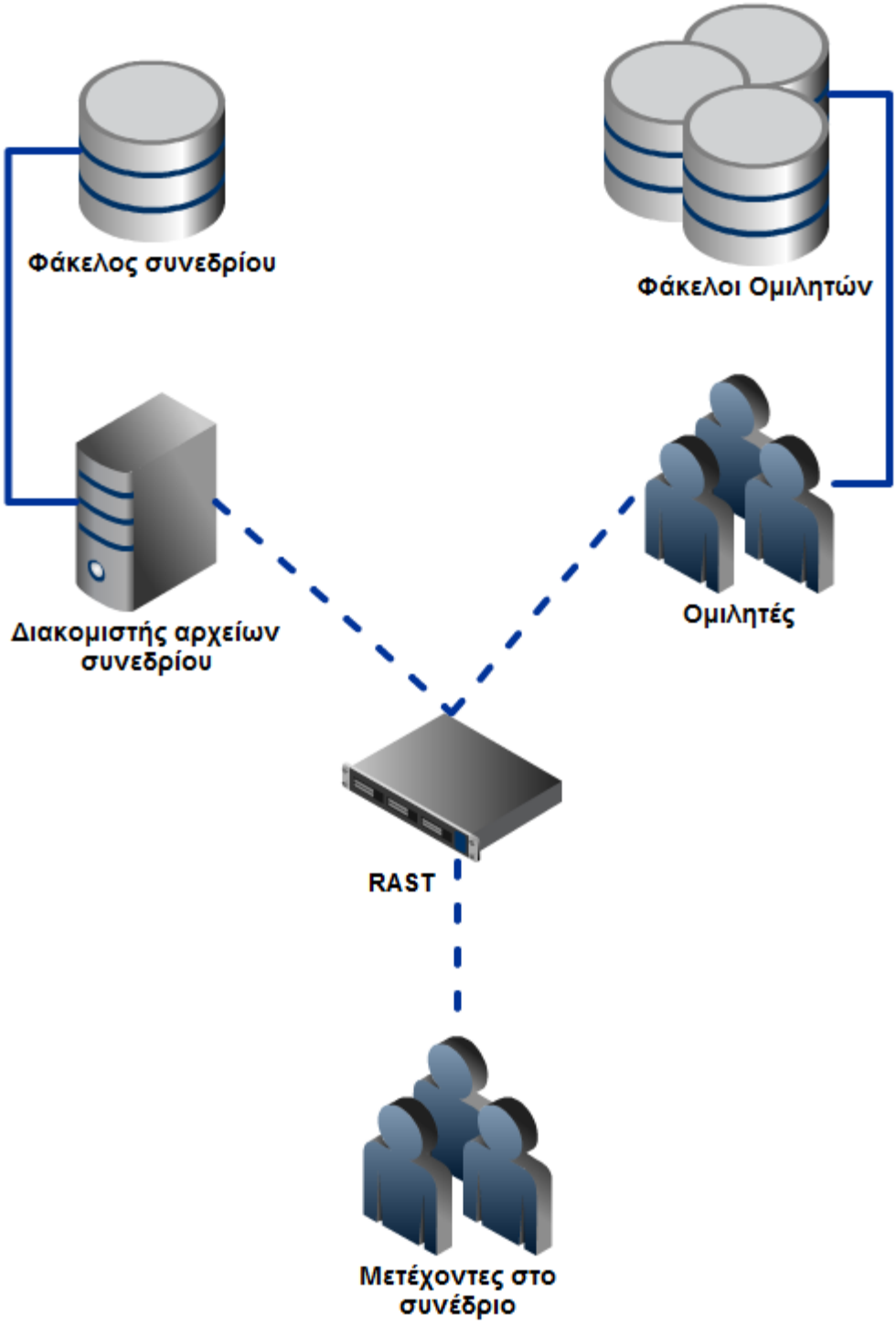


Σχήμα 26: Διάγραμμα σεναρίου εταιρικού δικτύου

Εδώ αξίζει να σημειωθεί ότι γενικά αποφεύγεται να δίνεται πρόσβαση σε επισκέπτες ενός οργανισμού στο domain του οργανισμού, ακόμα και με λογαριασμούς περιορισμένων δικαιωμάτων επισκέπτη (guest accounts). Και τούτο διότι πολλές φορές αποκαλύπτονται τρύπες στην ασφάλεια του κεντρικού διακομιστή, που επιτρέπουν σε κάποιον κακόβουλο χρήστη να αποκτήσει υψηλότερων δικαιωμάτων πρόσβαση εντός του domain και είτε να υποκλέψει πληροφορίες, είτε να δημιουργήσει προβλήματα στην ομαλή λειτουργία του οργανισμού. Για αυτά τα προβλήματα βέβαια, όταν ανακαλύπτονται, ο κατασκευαστής του λογισμικού διαθέτει σχετικές ενημερώσεις που επιλύουν τα προβλήματα ασφαλείας αυτά, αλλά πολλοί διαχειριστές επιλέγουν να μην δίνουν τέτοιες προσβάσεις για λόγους προληπτικούς νέων τέτοιου τύπου αποκαλύψεων.

6.3 Πρόσβαση και ανταλλαγή πόρων σε συνέδριο

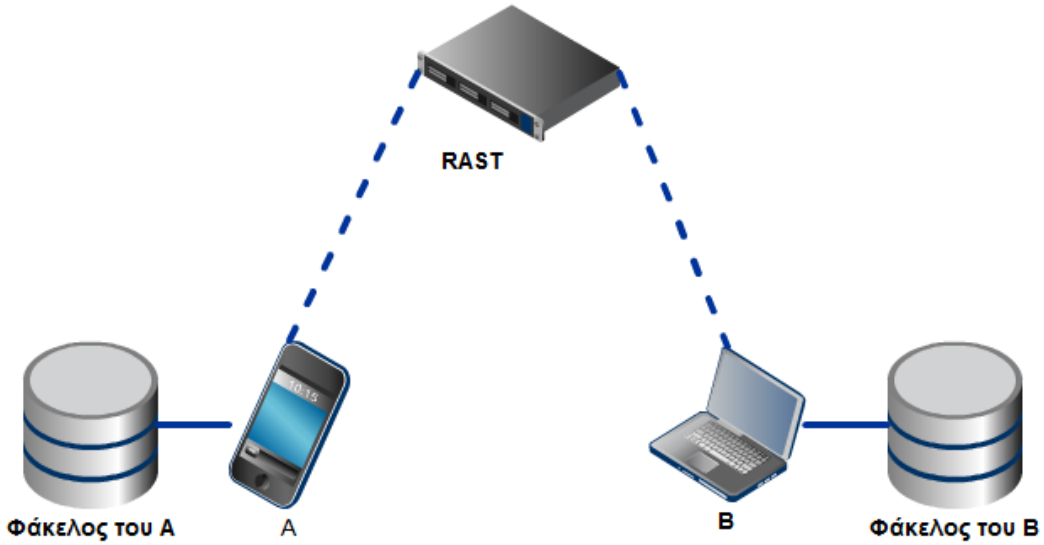
Ένα συνέδριο ή κάποιου είδους συνεστίαση ανθρώπων για συνεργασία παρουσία ενός διοργανωτή, αποτελεί και το παράδειγμα πάνω στο οποίο δομήθηκε και δημιουργήθηκε η εργασία αυτή. Ένα τέτοιο περιβάλλον είναι και το ιδανικό για την τεκμηρίωση ομοσπονδιών χρηστών, με σημαντικό πεδίο έρευνας στις συνεργατικές τεχνολογίες. Το γεγονός ότι υπάρχει διοργανώτρια αρχή δίνει και τη λύση στο ερώτημα για το ποιος θα κάνει την εγκατάσταση του συστήματος RAST και θα παραχωρήσει τα αντίστοιχα δικαιώματα χρήσης στους ενδιαφερόμενους.



Σχήμα 27: Εγκατάσταση σε συνέδριο

Στο παράδειγμα αυτό λοιπόν, η διοργάνωση κάνει μια εγκατάσταση του συστήματος RAST, και δίνει σε όλους τους συμμετέχοντες στοιχεία λογαριασμών. Επιπλέον παρέχει έναν φάκελο με υλικό που αφορά το συγκεκριμένο συνέδριο. Κάθε ομιλητής έχει τη δυνατότητα επιπλέον να δώσει πρόσβαση σε αρχεία που αφορούν την δικιά του παρουσίαση. Τέλος, ας θεωρήσουμε το σενάριο δυο χρήστες, ο Α και ο

Β, γνωρίζονται προσωπικά στα πλαίσια του συνεδρίου, και επιθυμούν να ανταλλάξουν βιογραφικά. Έτσι, ο Α δημιουργεί ειδικό μοιραζόμενο φάκελο με το βιογραφικό του, και το διαμοιράζει στο RAST επιτρέποντας την ανάγνωση μόνο από τον Β. Αντίστοιχα και ο Β δημιουργεί σχετικό κεφάλαιο επιτρέποντας πρόσβαση μόνο στον Α. Έτσι, χωρίς στοιχεία ο ένας για την συσκευή του άλλου ή άλλες πληροφορίες, καταφέρνουν να ανταλλάξουν τα επιθυμητά στοιχεία. Όλες οι συναλλαγές είναι ασφαλείς και κανένας εκτός των επιθυμητών δεν καταφέρνει να έχει πρόσβαση στα αρχεία που διαμοιράστηκαν εντός του συστήματος. Ακολουθεί σχετικό διάγραμμα:



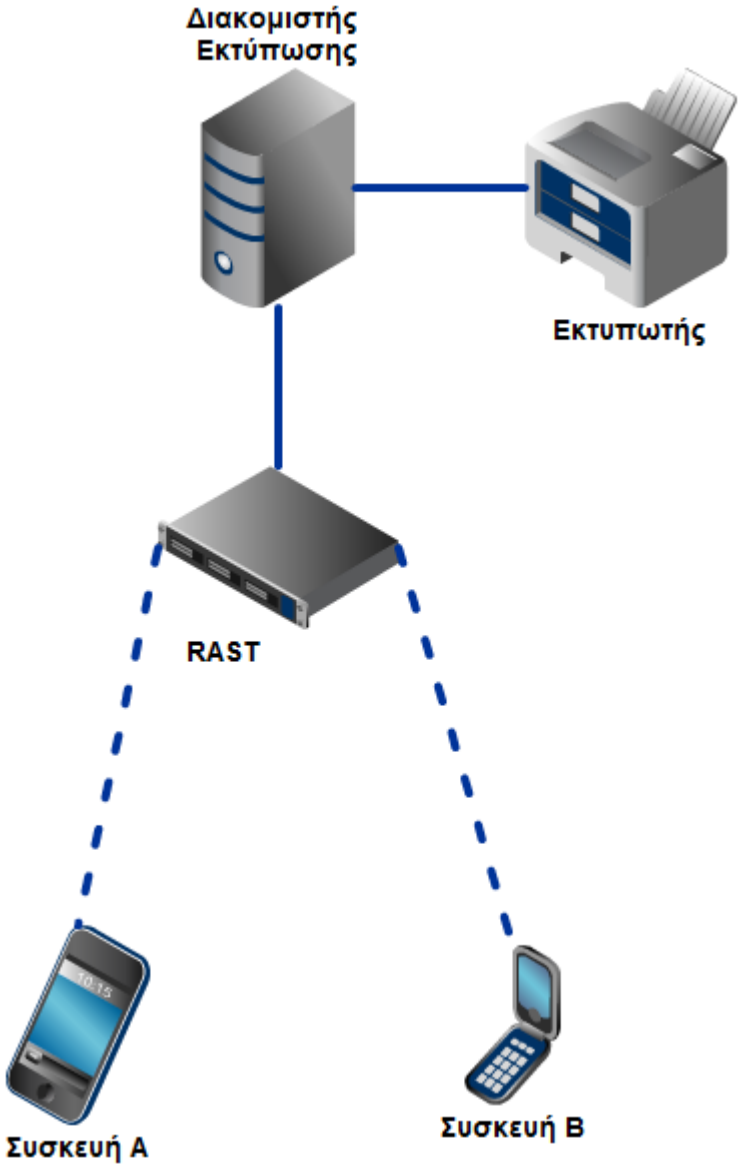
Σχήμα 28: Συνεργασία χρηστών εντός του συνεδρίου

Είναι προφανές πως το παράδειγμα αυτό μπορεί να επεκταθεί σε οποιουδήποτε τύπου συνεύρεση ανθρώπων, δεδομένης της παρουσίας ενός τρίτου διοργανωτή ο οποίος θα αναλάβει να εγκαταστήσει το σύστημα, και τον οποίο εμπιστεύονται όλοι οι συμμετέχοντες. Δεδομένης και της υποστήριξης του συστήματος για φορητές συσκευές, δημιουργείται ένα φυσικό περιβάλλον για συνεργατικότητα με χρήση ομοσπονδιών προσωπικών δικτύων, το οποίο διευκολύνεται από την παρουσία του συστήματος RAST. Ο προσωρινός χαρακτήρας των δικαιωμάτων που παραχωρούνται από το σύστημα ενισχύει περεταίρω την εφαρμογή αυτή, μιας και όλες αυτού του τύπου οι συνευρέσεις έχουν δεδομένη χρονική διάρκεια, μετά το πέρας της οποίας το σύστημα διαγράφει όλα τα στοιχεία που τους αφορούν.

6.4 Παροχή δυνατότητας εκτύπωσης σε χρήστες φορητών συσκευών

Το συγκεκριμένο παράδειγμα δίνεται για να τονιστεί η δυνατότητα του συστήματος να υποστηρίξει φορητές συσκευές. Δεδομένης της πολύ γρήγορης ανάπτυξης των έξυπνων κινητών τηλεφώνων (smartphones) και της ευρείας διάδοσής τους στην αγορά, γίνεται προφανές ότι ολοένα και θα αυξάνεται η ανάγκη για συστήματα που μπορούν να παρέχουν υπηρεσίες σε τέτοιες συσκευές.

Έτσι, μπορεί να γίνει μια εγκατάσταση του συστήματος RAST, συνδεδεμένη με κάποιον εκτυπωτή, ώστε να δίνει τη δυνατότητα σε χρήστες φορητών συσκευών να κάνουν κάποια εκτύπωση. Ο διαχειριστής δημιουργεί τους λογαριασμούς για χρόνο που επιθυμεί, και παρέχει την πρόσβαση στους ενδιαφερόμενους. Σχηματικά το βλέπουμε παρακάτω:



Σχήμα 29: Παροχή δυνατότητας εκτύπωσης σε φορητές συσκευές

Για λόγους απλότητας, το σενάριο αυτό είναι περιορισμένο, αλλά θα μπορούσαν να ενταχθούν και σε αυτό κι άλλοι πόροι, όπως φάκελοι ή πρόσθετοι εκτυπωτές. Το σενάριο αυτό δόθηκε για να τονιστεί η δυνατότητα συνεργασίας του συστήματος με φορητές συσκευές.

7

Μειονεκτήματα / περιορισμοί υλοποίησης

Μιας και ο σκοπός της εκπόνησης αυτής της διπλωματικής εργασίας ήταν η παρουσίαση μιας λύσης για τα θέματα ασφάλειας και διαμοιρασμού πόρων, υπάρχουν ορισμένα μειονεκτήματα που προκύπτουν από την υλοποίηση αυτή, τα οποία αφορούν κυρίως βελτιώσεις που δεν ήταν απαραίτητες για την επίδειξη της προτεινόμενης λύσης. Επιπλέον, υπάρχουν και ορισμένοι σχεδιαστικοί περιορισμοί που προκύπτουν από τις ίδιες τις απαιτήσεις του συστήματος.

Στην παράγραφο αυτή αναφέρονται τα σημαντικότερα από αυτά με προτάσεις για πιθανή επίλυση / βελτίωση αυτών.

7.1 Πολλαπλές βάσεις χρηστών

Ένα από τα κύρια μειονεκτήματα της υλοποίησης είναι η διατήρηση της βάσης των χρηστών του συστήματος σε τρία μέρη. Συγκεκριμένα, τα στοιχεία των χρηστών αποθηκεύονται:

1. στην βάση passwd / shadow του linux. Από αυτήν δημιουργούνται τα απαραίτητα δικαιώματα για την πρόσβαση στο σύστημα, και οι πάροχοι μπορούν να εντάξουν πόρους τους στο σύστημα.
2. Στην βάση smbpasswd του samba. Μέσω αυτής της βάσης οι καταναλωτές αποκτούν πρόσβαση στους διαμοιραζόμενους από το σύστημα πόρους.
3. Στην βάση MySQL.

Η ύπαρξη στοιχείων για τους χρήστες σε αυτές τις τρεις βάσεις αποτελεί πρόβλημα ασφαλείας, καθώς κάθε κακόβουλος χρήστης που επιθυμεί να ανακτήσει τον έλεγχο του συστήματος, αρκεί να καταφέρει να αποκτήσει πρόσβαση σε μία από τις τρεις αυτές βάσεις. Έτσι, ο διαχειριστής του συστήματος οφείλει να παρακολουθεί την λειτουργία και των τριών αυτών βάσεων συστηματικά, γεγονός που αφενός προσθέτει στην πολυπλοκότητα του συστήματος, και αφετέρου τριπλασιάζει τις πιθανότητες κάποια από αυτές τις βάσεις να παρουσιάσει σχεδιαστική τρύπα ασφαλείας (security hole).

Προτείνεται η ενοποίηση των τριών διακριτών βάσεων δεδομένων χρηστών σε μία. Με αυτό τον τρόπο, ο διαχειριστής του συστήματος χρειάζεται να εξασφαλίσει την ασφάλεια μόνο σε αυτήν τη βάση, εργασία σαφώς ευκολότερη και με μικρότερα περιθώρια σφάλματος. Σύμφωνα με την βιβλιογραφία, θα μπορούσε αφενός να χρησιμοποιηθεί κάποιος LDAP server, όπως ο OpenLDAP [53], υλοποίηση που είναι και συνηθέστερη, μιας και το πρωτόκολλο LDAP έχει σχεδιαστεί και χρησιμοποιείται ευρέως για την ταυτοποίηση χρηστών [54]. Ενδεχομένως πιο απλή στον σχεδιασμό και την υλοποίηση είναι η χρήση κάποιας σχεσιακής βάσης δεδομένων όπως η MySQL [55]. Επειδή ήδη στο RAST χρησιμοποιείται για αποθήκευση στοιχείων αυτή η βάση, είναι ενδεχομένως μικρότερη εργασία η μεταφορά της ταυτοποίησης των χρηστών σε αυτή τη βάση, με μικρές σχετικά παραμετροποιήσεις στον κώδικα της εφαρμογής.

Συνοπτικά, μπορεί να ειπωθεί πως με την ενοποίηση των βάσεων σε μία, το σύστημα θα ελαφρυνθεί από την ύπαρξη των τριών βάσεων, τις εργασίες συγχρονισμού των δεδομένων και στις τρεις αυτές βάσεις, οπότε θα γίνει πιο απλή η συντήρηση και η μελλοντική επέκταση του συστήματος.

7.2 Εκτέλεση προγράμματος με δικαιώματα υπερχρήστη από την εφαρμογή

Για την εκτέλεση των διαφόρων λειτουργιών της εφαρμογής, απαιτούνται δικαιώματα υπερχρήστη (superuser) στο λειτουργικό σύστημα Unix. Και επειδή η εφαρμογή παραμετροποιείται μέσα από διεπαφή ιστοσελίδων (web interface), καλούνται τμήματα της εφαρμογής με δικαιώματα υπερχρήστη από τον διακομιστή ιστοσελίδων (web server). Αυτή η πρακτική είναι αρκετά ασυνήθιστη, καθώς συχνά

στη βιβλιογραφία αναφέρεται πως αυτή η προσέγγιση προκαλεί σημαντικά προβλήματα ασφάλειας.

Οι πιο διαδεδομένοι διακομιστές ιστοσελίδων (IIS, Apache) δημιουργούν στο λειτουργικό σύστημα έναν λογαριασμό χρήστη με εξαιρετικά περιορισμένα δικαιώματα, ακριβώς για να ελαχιστοποιήσουν τις κακόβουλες παραμετροποιήσεις που μπορούν να γίνουν μέσω του διαδικτύου στην ασφάλεια, σταθερότητα και ακεραιότητα του συστήματος. Έτσι τονίζεται περεταίρω η πρακτική αποφυγής της πρακτικής μια διαδικτυακή εφαρμογή να εκτελείται μέσω λογαριασμών συστήματος αυξημένων δικαιωμάτων.

Από την άλλη όμως, οι λειτουργίες που καλείται να εκτελέσει το σύστημα RAST, όπως η διαχείριση των χρηστών (προσθήκη, διαγραφή κλπ), και η διαχείριση των παραμέτρων της εφαρμογής Samba απαιτούν αυξημένα δικαιώματα στο σύστημα. Έτσι, χωρίς μεγάλες δομικές αλλαγές στο σύστημα, δεν είναι δυνατή η υλοποίηση ενός συστήματος σαν το RAST χωρίς την εκτέλεση με αυξημένα δικαιώματα εφαρμογών από τον διακομιστή ιστοσελίδων.

Για να περιοριστεί όσο γίνεται η πρόσβαση, τα αυξημένα δικαιώματα δεν ισχύουν για όλη την εφαρμογή. Έγινε χρήση της εντολής sudo και του αρχείου sudoers, ώστε ο διακομιστής να μπορεί να εκτελέσει με αυξημένα δικαιώματα μόνο ένα συγκεκριμένο κομμάτι της εφαρμογής, το server.php. Αυτό το κομμάτι της εφαρμογής έχει δεδομένες δυνατότητες, και έτσι για αύξηση της ασφάλειας του διαβεβαιώθηκε ότι δεν του έχει δοθεί καμία δυνατότητα αλλαγής των βασικών ρυθμίσεων του συστήματος. Αντίστοιχες ρυθμίσεις έχουν γίνει σε σχετικές εφαρμογές διαχείρισης του συστήματος Unix μέσω ιστοσελίδων, όπως για παράδειγμα οι εφαρμογές FAUS και WEBMIN [56].

Μια ενδεχόμενη εναλλακτική προσέγγιση θα ήταν όλες οι εντολές να γράφονται σε κάποια βάση δεδομένων από λογαριασμό με μειωμένα δικαιώματα, και κάποια εφαρμογή που τρέχει με αυξημένα δικαιώματα περιοδικά να ελέγχει αυτή τη βάση και να εκτελεί τις εντολές που δίνονται από εκεί. Κάτι τέτοιο όμως αυξάνει σημαντικά την πολυπλοκότητα του συστήματος, μειώνει την αμεσότητα των αποτελεσμάτων των εντολών που δίνονται στο σύστημα, και δεν εξαφανίζει τον κίνδυνο, μιας και όταν κάποιος κακόβουλος χρήστης καταφέρει να ελέγξει αυτή τη βάση, παραμένει η δυνατότητά του για εκτέλεση ανεπιθύμητων ενεργειών στο σύστημα.

7.3 Αδυναμία ελέγχου για ιούς και λοιπά κακόβουλα αρχεία

Το σύστημα RAST έχει υλοποιηθεί με την λογική του tunneling, δηλαδή με απευθείας πρόσβαση του καταναλωτή στον πάροχο. Έτσι, το σύστημα RAST δεν κρατάει κάποιο προσωρινό αντίγραφο των αρχείων που διακινούνται εντός του συστήματος, ώστε να μπορεί να ελέγξει για πιθανό κακόβουλο περιεχόμενο σε κάποιο από τα αρχεία. Έτσι, για προστασία από ιούς και λοιπά κακόβουλα προγράμματα θα πρέπει ο κάθε χρήστης που συμμετέχει να έχει φροντίσει ο ίδιος.

Για να αντιμετωπιστεί αυτό το πρόβλημα, θα έπρεπε να αλλάξει η προσέγγιση του συστήματος, από tunneling σε αποθήκευσης και μετά αποστολής (store-and-forward). Έτσι, στην προσωρινή αποθήκευση μπορούν να γίνουν οι απαραίτητοι έλεγχοι ώστε να ελαχιστοποιηθεί τέτοιος κίνδυνος.

Μια τέτοια αντιμετώπιση όμως έχει ορισμένα σοβαρά μειονεκτήματα με τη σειρά της. Αφενός μειώνεται σημαντικά η απόδοση του RAST, μιας και οι διαδικασίες αποθήκευσης και εκ νέου προώθησης απαιτούν και χώρο, και επεξεργασία, και εισάγουν σημαντικές καθυστερήσεις στο σύστημα. Επιπλέον, μια τέτοια προσέγγιση δεν μπορεί να γίνει μέσω του πρωτοκόλλου SMB, μιας και κάτι τέτοιο δεν υποστηρίζεται στο πρωτόκολλο. Άρα, θα χρειαζόταν να δημιουργηθεί νέο πρωτόκολλο, βλάπτοντας την ευκολία και συμβατότητα του συστήματος. Τέλος, ακριβώς λόγω του ενδιάμεσου βήματος, οι εργασίες που γίνονται από πάροχο και καταναλωτή δεν πραγματοποιούνται σε πραγματικό χρόνο, οπότε προκύπτουν θέματα συγχρονισμού των αλλαγών, που είναι εξαιρετικά πολύπλοκα στην επίλυσή τους και η ολοκληρωτική αντιμετώπισή τους δεν είναι δυνατή.

8

Μελλοντική έρευνα

Η βασική ιδέα για την ανάπτυξη του RAST δόθηκε από το μεγάλο κενό που υπάρχει μεταξύ των υπαρχόντων τεχνολογιών και αρχιτεκτονικών ασφάλειας και των καταιγιστικών εξελίξεων στους τομείς της φορητότητας, του ubiquitous computing, του context-aware computing και των προσωπικών δικτύων. Είναι πλήρως κατανοητό πως το RAST δεν μπορεί να απαντήσει σε όλες τις απαιτήσεις που προκύπτουν σε αυτό το περιβάλλον εξελίξεων, οπότε έμφαση δόθηκε στο να προταθεί κάποια λύση για ένα υποσύνολο προβλημάτων.

Όμως η αρχιτεκτονική που προτείνεται μπορεί να επεκταθεί και να απαντήσει σε μεγαλύτερο αριθμό προβλημάτων. Το λειτουργικό σύστημα Linux που χρησιμοποιήθηκε μπορεί να υποστηρίξει πολλές περισσότερες λειτουργίες και τεχνολογίες, οι οποίες μπορούν να ολοκληρωθούν πάνω σε μελλοντικές εκδόσεις του συστήματος.

Επιπλέον, λόγω των στενών περιθωρίων, αναπτύχθηκε περισσότερο σε μια λογική proof-of-concept παρά σαν μία αυτόνομη και αξιόπιστη λύση έτοιμη για αξιόπιστες επαγγελματικές και βιομηχανικές εγκαταστάσεις. Σαν απλή διπλωματική εργασία δεν είχε πρόσβαση σε ειδικούς επαγγελματίες των χώρων της ασφάλειας, των λειτουργικών συστημάτων και του προγραμματισμού, ώστε να αποτελεί μια ώριμη, γρήγορη και αξιόπιστη πλατφόρμα για ευρύτερη αξιοποίηση από την αγορά.

Για αυτούς τους λόγους, θεωρήθηκε σκόπιμο να συμπεριληφθούν ορισμένες κατευθυντήριες γραμμές για όποιον ασχοληθεί μελλοντικά να βελτιώσει το σύστημα

και ενδεχομένως να το αναπτύξει για την αγορά. Σε καμία περίπτωση δεν πρέπει να θεωρηθεί ότι η λίστα αυτή είναι πλήρης.

Έτσι λοιπόν, το κεφάλαιο αυτό θα εστιαστεί αφενός σε προτεινόμενες βελτιώσεις της υπάρχουσας υλοποίησης για μεγαλύτερη αξιοπιστία, και αφετέρου σε ένα υποσύνολο των πιθανών επεκτάσεων που μπορούν να γίνουν, διευρύνοντας το αντικείμενο του συστήματος και των λύσεων που μπορεί να προσφέρει.

8.1 Προτάσεις για βελτίωση της παρούσας υλοποίησης

8.1.1 Μέτρηση επιδόσεων

Το σύστημα RAST δημιουργήθηκε για να μπορεί να εξυπηρετήσει μεσαίο μέχρι μεγάλο αριθμό χρηστών ταυτόχρονα. Για αυτό το λόγο, έχει μεγάλη σημασία να έχει τέτοια απόδοση ώστε να μπορεί να εξυπηρετήσει αυτό τον όγκο. Η εμπειρική μελέτη που έγινε σε αυτή τη διπλωματική έδειξε ότι το σύστημα χειρίζεται με ευκολία δέκα ταυτόχρονους χρήστες, αλλά δεν έγινε περεταίρω έρευνα. Για οποιαδήποτε αντικειμενική όμως μέτρηση, χρειάζεται να αναπτυχθούν εξειδικευμένα εργαλεία μέτρησης επιδόσεων, τα οποία να λειτουργήσουν παράλληλα με σχετικά εργαλεία που βρίσκονται στο εμπόριο, ώστε να αποτιμηθούν οι απόλυτες δυνατότητες του συστήματος.

Η μέτρηση των επιδόσεων, εκτός του να αποτιμήσει τις υπάρχουσες δυνατότητες, μπορεί να δείξει και τα σημεία εκείνα της εφαρμογής που έχουν τα μεγαλύτερα προβλήματα, και καθυστερούν περισσότερο την εφαρμογή. Έτσι, αν για παράδειγμα μετρηθεί ότι η εισαγωγή των χρηστών έχει υψηλή καθυστέρηση, είναι κατανοητό ότι βελτίωση αυτού του υποπρογράμματος θα βελτιώσει σημαντικά την ταχύτητα όλης της εφαρμογής.

Για αυτούς τους λόγους, προτείνεται να αναπτυχθούν ή να αναζητηθούν στην αγορά εργαλεία μέτρησης των επιδόσεων και εξομοίωσης χρήσης από πολλούς ταυτόχρονους χρήστες, και στις συνολικές λειτουργίες του συστήματος, όπως προσθαφαίρεση χρηστών, φακέλων και εκτυπωτών, αλλά και στις επιμέρους, όπως εισαγωγή στη MySQL βάση, ή καταγραφή συμβάντων. Επιπλέον μπορεί να μελετηθεί η συμπεριφορά του συστήματος σε συνάρτηση με το υλικό (hardware) που είναι εγκατεστημένο. Έτσι θα είναι εφικτή η απόδοση του RAST πχ. σε συνάρτηση με την ταχύτητα της μνήμης ή του σκληρού. Σίγουρα όμως η αύξηση του εύρους

ζώνης στη σύνδεση με τους χρήστες προσφέρει σημαντικά στο σύστημα, μιας και ο διαμοιρασμός αρχείων απαιτεί σημαντικές ταχύτητες, οπότε γίνεται σαφές πως με πολλούς ταυτόχρονους χρήστες, σε συνδέσεις με μικρό εύρος, οι επιδόσεις θα πλήττονται σημαντικά.

Συνολικά, για κάθε τέτοιου είδους σύστημα είναι ιδιαίτερα σημαντικό να γνωρίζουμε τις επιδόσεις του, όπως και τις αντοχές στις ανάγκες των εγκαταστάσεων. Έτσι, θα είναι και πιο εύκολο για τις μελλοντικές επεκτάσεις να εντοπίσουν πιθανά σημεία που μπορούν να βελτιωθούν, βελτιώνοντας έτσι τις συνολικές επιδόσεις του RAST.

8.1.2 Ανάλυση και βελτίωση επιδόσεων

Με την πολυπλοκότητα του συστήματος, καθώς και την πλειάδα των προγραμμάτων που παίζουν ρόλο στο σύστημα, εισάγονται πλήθος καθυστερήσεων στο σύστημα. Στο διαδίκτυο και σε άλλες πηγές, υπάρχει εκτενής έρευνα και προτάσεις για την βελτιστοποίηση των επιμέρους συστατικών του συστήματος, ώστε να μειωθούν όλες αυτές οι καθυστερήσεις όσο είναι δυνατόν. Επιπλέον, μπορεί και να μειωθεί και η πολυπλοκότητα του συστήματος, με τεχνικές όπως την ενοποίηση των βάσεων χρηστών στην ταχύτερη δυνατή λύση, όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο.

Μια συνοπτική λίστα των συστατικών του συστήματος και των πιθανών βελτιώσεων ακολουθεί:

1. Βελτιστοποίηση του λειτουργικού συστήματος Linux:
    - Με δημιουργία εξειδικευμένου πυρήνα (kernel) απαλλαγμένου από οδηγούς (drivers) και λοιπές λειτουργίες άσχετες προς την εφαρμογή, όπως οδηγούς γραφικών και ήχου.
    - Με αφαίρεση όλων των άσχετων προς την υλοποίηση εγκατεστημένων προγραμμάτων και υπηρεσιών, όπως αυτές που αφορούν τον ήχο και τα γραφικά. Επιπλέον μπορούν να αφαιρεθούν και όσα αρχεία δεν είναι απαραίτητα για την λειτουργία του συστήματος, όπως τα διάφορα αρχεία τεκμηρίωσης.
    - Με βελτιστοποίηση και αντίστοιχη παραμετροποίηση των διαφόρων σχετικών οδηγών όπως οδηγοί για τον επεξεργαστή, την πρόσβαση στον σκληρό δίσκο και το υποσύστημα δικτύου.

2. Βελτιστοποίηση του Samba
    - Αφαίρεση όλων των μη απαραίτητων υποσυστημάτων.
    - Ενδεχόμενη επαναμεταγλώττιση (recompile) του Samba για ακόμα μεγαλύτερη απόδοση.
3. Βελτιστοποίηση MySQL
    - Αφαίρεση όλων των δυνατοτήτων άσχετων προς το σύστημα, πχ οδηγούς.
    - Δημιουργία όλων των απαραίτητων indexes για ταχύτητα στη βάση.
4. Βελτιστοποίηση κώδικα
    - Βελτιστοποίηση όλων των σελίδων και υποσυστημάτων σε PHP και Perl.
    - Μεταφορά ορισμένων υποσυστημάτων σε πιο αποδοτικές γλώσσες προγραμματισμού, όπως η C.

Φαίνεται λοιπόν ότι το σύστημα RAST επιδέχεται πλήθος βελτιώσεων, που μπορούν να επιταχύνουν σημαντικά το σύστημα, αυξάνοντας τους χρήστες που μπορεί να εξυπηρετήσει ταυτόχρονα. Η εκτενής μελέτη των επιδόσεων, όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, μπορεί να υποδείξει τα σημεία εκείνα της εφαρμογής που την καθυστερούν περισσότερο, οπότε η σωστή μεθοδολογία μετρήσεων μπορεί να αποδείξει κάθε πιθανή βελτίωση και να την μετρήσει ποσοτικά.

8.1.3 Καλύτερη υποστήριξη εκτύπωσης

Το σύστημα CUPS που χρησιμοποιείται από την εφαρμογή για την υποστήριξη της εκτύπωσης παρέχει ορισμένα σημαντικά στοιχεία που θα μπορούσαν να χρησιμοποιηθούν από το RAST, επεκτείνοντας την λειτουργικότητά του.

Μια δυνατότητα λοιπόν που μπορεί να προστεθεί είναι η ενημέρωση του καταναλωτή για την κατάσταση του εκτυπωτή στον οποίο προσπαθεί να εκτυπώσει, όπως για παράδειγμα αν τυπώνει ήδη κάτι, ή αν ο εκτυπωτής είναι εκτός λειτουργίας.

Ακόμα, το σύστημα εκτυπώνει σύμφωνα με τις προκαθορισμένες ρυθμίσεις του οδηγού του κάθε εκτυπωτή. Έτσι, ο πάροχος θα μπορούσε να παραμετροποιήσει αυτές τις ρυθμίσεις ώστε για παράδειγμα να μην είναι δυνατή η έγχρωμη εκτύπωση, ή η εκτύπωση σε υψηλή ποιότητα.

Επιπλέον, μπορεί να μπει περιορισμός στον αριθμό σελίδων που μπορεί να εκτυπώσει κάποιος, προσθήκη η οποία καθιστά το σύστημα εξαιρετικά ελκυστικό για εταιρίες που θέλουν να προσφέρουν υπηρεσίες εκτύπωσης, όπως αναφέρεται και στο σχετικό σενάριο.

Γενικότερα το θέμα της εκτύπωσης έχει πολύ μεγάλο αριθμό μεταβλητών που μπορούν να παραμετροποιηθούν, και έτσι να παρέχεται συστηματικότερη υποστήριξη. Στην παρούσα εργασία η έμφαση δόθηκε στο να παρουσιαστεί η δυνατότητα εκτύπωσης, και όχι στην εξάντληση όλων των δυνατοτήτων.

8.1.4 Εκτενής έρευνα για πιθανά κενά ασφαλείας

Το σύστημα χρησιμοποιεί πολλές τεχνολογίες για να φέρει σε πέρας τις λειτουργίες του. Συνοπτικά:

- Πυρήνας Linux
- Υποσύστημα δικτύου
- Εντολές του λειτουργικού όπως η mount
- Perl Scripts
- PHP σελίδες και εφαρμογές
- Samba Server
- MySQL Server
- CUPS
- Apache Server

Κάθε μια από αυτές τις τεχνολογίες ενδεχομένως να παρουσιάσει κάποιο κενό ασφαλείας, ή να χρειάζεται συγκεκριμένες παραμετροποιήσεις ώστε να καταστεί ασφαλής για την χρήση στις εφαρμογές για τις οποίες προτείνεται το RAST. Γιατί ενδεχόμενο κενό ασφαλείας σε ένα από όλα τα συστατικά του συστήματος να θέσει σε κίνδυνο όλο το σύστημα και τις πληροφορίες / πόρους που διακινούνται μέσα σε αυτό.

Όμως, τέτοιες ενδελεχείς έρευνες γίνονται στην αγορά από ειδικούς επιστήμονες στον τομέα της ασφάλειας, συχνά εξειδικευμένους σε περιορισμένο αριθμό τεχνολογιών. Για αυτό, εφόσον προχωρήσει κάποιος σε εμπορική διάθεση ενός τέτοιου συστήματος, υπάρχει η ανάγκη συμβολής τέτοιου είδους επιστημόνων. Κάτι τέτοιο φυσικά εκπίπτει από το ενδιαφέρον ενός μηχανικού, αλλά προτείνεται η μελλοντική εκτενής έρευνα στο σύστημα, ώστε να γίνει γνωστή η αξιοπιστία του σε

επίπεδο ασφάλειας. Αντίστοιχα εκτενής πρέπει να είναι και η συνεχής ενημέρωση του συστήματος με όλες τις ενημερώσεις ασφαλείας που βγαίνουν για τα συστατικά του στοιχεία, μιας και στον τομέα της ασφάλειας, τρύπες ανακαλύπτονται και διορθώνονται σχεδόν με καθημερινούς ρυθμούς. Οπότε σε πιθανή εγκατάσταση, πρέπει κάποιος εξειδικευμένος διαχειριστής του συστήματος (administrator) να παρακολουθεί στενά τις εξελίξεις, ώστε να ενημερώνει το αμεσότερο δυνατό το σύστημα.

8.2 Προτάσεις για περεταίρω εξέλιξη

8.2.1 Βελτίωση των υπηρεσιών

Αυτή τη στιγμή το σύστημα είναι σε θέση να μοιράσει πόρους τύπου φακέλων (αρχείων) και εκτυπωτών. Και όλα αυτά αξιοποιώντας αποκλειστικά για τη σύνδεση των χρηστών με τους πόρους (είτε αφορά την κατανάλωση, είτε την παροχή) το πρωτόκολλο SMB. Οι περιορισμοί πρόσβασης που τίθενται αφορούν προσωπικό επίπεδο (λογαριασμός χρήστη), χρονικό περιορισμό, καθώς και χωρικό, μέσω σχετικού δικτυακού εξοπλισμού πάνω στον οποίο μπορεί να εγκατασταθεί το RAST.

Όμως, οι υπηρεσίες αυτές μπορούν να βελτιωθούν ποιοτικά. Αφενός με ένταξη στο σύστημα επιπλέον πρωτοκόλλων διαμοιρασμού και πρόσβασης σε πόρους, επεκτείνοντας την υποστήριξή του για άλλα πρωτόκολλα διαμοιρασμού αρχείων ή για άλλα πρωτόκολλα υποστήριξης εκτυπωτών. Αφετέρου μπορούν να δοθούν επιπλέον δυνατότητες παραμετροποίησης για τους πάροχους πόρων, ώστε να έχουν μεγαλύτερο έλεγχο στο τι και πώς θέλουν να παρέχουν.

Σε ότι αφορά τα αρχεία, προτείνονται οι εξής επεκτάσεις:

- Δυνατότητα διαμοιρασμού τοπικών φακέλων του συστήματος RAST, αν κάποια εφαρμογή απαιτεί κάτι τέτοιο.
- Πρόσβαση σε αρχεία μέσω του πρωτοκόλλου NFS, καθώς και παροχή πρόσβασης μοιραζόμενων πόρων μέσω NFS [57].
- Πρόσβαση σε αρχεία μέσω του πρωτοκόλλου SFTP (ή του απλού FTP, αν και κάτι τέτοιο δεν συνίσταται για λόγους ασφάλειας), καθώς και παροχή πρόσβασης μοιραζόμενων πόρων μέσω SFTP [58].
- Πρόσβαση σε αρχεία μέσω του πρωτοκόλλου WEBDAV, καθώς και παροχή πρόσβασης μοιραζόμενων πόρων μέσω WEBDAV [59].

- Παροχή πρόσβασης σε αρχεία μέσω ιστοσελίδων, από ειδικά διαμορφωμένη δικτυακή εφαρμογή, ή εναλλακτικά με ενσωμάτωση στο σύστημα κάποιας υπάρχουσας στην αγορά σχετικής λύσης, όπως το πρόγραμμα BytesFall Explorer [60]

Στο θέμα της βελτίωσης του υποσυστήματος εκτύπωσης, προτείνονται οι εξής επεκτάσεις:

- Δυνατότητα εκτύπωσης σε τοπικούς εκτυπωτές, δηλαδή σε εκτυπωτές συνδεδεμένους με το μηχάνημα που τρέχει το RAST.
- Δυνατότητα εκτύπωσης μέσω του πρωτοκόλλου Internet Printing Protocol (IPP) [61], καθώς και δυνατότητα πρόσβασης σε εκτυπωτές που είναι συνδεδεμένοι στο δίκτυο και υποστηρίζουν το συγκεκριμένο πρωτόκολλο. Αν και το CUPS σύμφωνα με τις δοκιμές μας υποστηρίζει να μοιράσει μέσω του πρωτοκόλλου SMB έναν εκτυπωτή που χρησιμοποιεί αυτό το πρωτόκολλο, δεν είναι εξίσου εύκολο ένας SMB εκτυπωτής να διαμοιραστεί με το IPP.
- Δυνατότητα εκτύπωσης σε εκτυπωτές με χρήση των τεχνολογιών ασύρματης δικτύωσης Wi-Fi ή με χρήση Bluetooth.
- Δυνατότητα περιορισμού στην ποιότητα εκτύπωσης (πχ αναγκαστική εκτύπωση σε πρόχειρη ποιότητα, αναγκαστική ασπρόμαυρη εκτύπωση).
- Δυνατότητα περιορισμού σελίδων εκτύπωσης ανά λογαριασμό. Με τους τελευταίους δυο τρόπους μπορεί να υλοποιηθεί καλύτερα το σενάριο ενός print shop.
- Υπηρεσία εκτύπωσης μέσω τοποθέτησης συμβατών με το Ubuntu Linux αρχείων σε ειδικό φάκελο. Ο φάκελος θα ελέγχεται περιοδικά για νέα αρχεία, και για καθένα από αυτά θα εκτελείται κάποιο πρόγραμμα εκτύπωσης (πχ ghostscript [62] για αρχεία ps και pdf, openoffice για αρχεία τύπου doc και xls, και αντίστοιχα προγράμματα για αρχεία εικόνας και απλού κειμένου και άλλα). Μετά την εκτύπωση τα αρχεία θα διαγράφονται.

Και στους δυο τύπους πόρων όμως θα ήταν επιθυμητό να επεκταθεί όσο είναι δυνατόν ο αριθμός των υποστηριζόμενων πλατφόρμων, κυρίως με τις σύγχρονες

φορητές πλατφόρμες (mobile platforms) για υποστήριξη, τουλάχιστον σε κατανάλωση πόρων, από τις νέες φορητές συσκευές νέας γενιάς.

Φαίνεται λοιπόν πως η υποστήριξη πρωτοκόλλων και η περεταίρω παραμετροποιήσεις που προσφέρονται από το Ubuntu Linux είναι αρκετές περισσότερες από όσες υλοποιήθηκαν για τον σκοπό της διπλωματικής, και είναι στη διάθεση του κάθε ενδιαφερόμενου να τις ενσωματώσει σε επόμενες εκδόσεις του RAST, επεκτείνοντας έτσι τη λειτουργικότητα και τη συμβατότητά του.

8.2.2 Επέκταση των πόρων που μπορεί να διαμοιράσει το σύστημα

Οι δυο τύποι πόρων που αντιμετώπισε το RAST (εκτύπωση, αρχεία) δεν είναι οι μόνοι τύποι πόρων που μπορούν να διαμοιραστούν από ένα τέτοιου τύπου σύστημα. Αποτελούν όμως τις πιο συνηθισμένες ανάγκες που συναντάμε σε δικτυακά περιβάλλοντα, με εξαιρετικά δημοφιλή και ευρεία χρήση.

Παρακάτω θα παρουσιαστούν συνοπτικά ορισμένοι άλλοι τύποι πόρων που μπορούν να προσφερθούν τέτοιου είδους συστήματα, και μπορούν να ενσωματωθούν στην ήδη υπάρχουσα λειτουργικότητα του RAST.

8.2.2.1 Πρόσβαση σε υπολογιστικούς πόρους / υπερυπολογιστές

Υπάρχουν ορισμένα υπολογιστικά προβλήματα που απαιτούν ιδιαίτερα μεγάλο αριθμό υπολογισμών για να επιλυθούν, αριθμό τέτοιο, που οι πόροι ενός προσωπικού υπολογιστικού συστήματος δεν επαρκούν ώστε να επιλυθούν σε εύλογο χρονικό διάστημα. Παραδείγματα τέτοιων προβλημάτων είναι οι διάφορες εξομοιώσεις που κάνουν οργανισμοί και επιστήμονες, οι οποίες έχουν μεγάλο αριθμό παραμέτρων και βημάτων για την επίλυσή τους, καθώς και πολύπλοκα αριθμητικά προβλήματα του τομέα της αριθμητικής ανάλυσης και των πεπερασμένων στοιχείων.

Για τέτοιου είδους προβλήματα υπάρχουν ορισμένοι υπολογιστές με σημαντικά μεγαλύτερη ταχύτητα / μνήμη και λοιπούς πόρους, που τους ονομάζουμε υπερυπολογιστές (supercomputers) [63]. Επιπλέον υπάρχουν διαθέσιμες αρχιτεκτονικές ώστε ένα σύνολο πολλών υπολογιστών να ασχολείται ταυτόχρονα και παράλληλα με την επίλυση ενός προβλήματος, πολλαπλασιάζοντας έτσι τους πόρους από την διαίρεση του προβλήματος και την επίλυση τμήματος από καθέναν από τους επί μέρους υπολογιστές. Παραδείγματα τέτοιων αρχιτεκτονικών είναι τα Cluster [64], καθώς και οι διατάξεις GRID [65], που χρησιμοποιούνται ευρέως στην αγορά στη σύγχρονη εποχή.

Τέλος, λειτουργικά συστήματα σαν το Unix προσφέρουν δυνατότητες μέσω των κελυφών τους (shell) που δεν είναι διαθέσιμες σε άλλα λειτουργικά, και μπορούν να δουλέψουν με απομακρυσμένη ασφαλή πρόσβαση (πχ Secure Shell / SSH) [66].

Σε όλες αυτές τις περιπτώσεις μπορεί ένα σύστημα σαν το RAST να παρέχει λογαριασμούς πρόσβασης περιορισμένης χρονικής διάρκειας σε οργανισμούς που έχουν την ανάγκη να διαμοιράσουν τέτοιου είδους πόρους σε περισσότερους του ενός χρήστη, ακόμα και σε περιπτώσεις επισκεπτών. Κι επειδή τα συστήματα ταυτοποίησης τέτοιου είδους συστημάτων είναι συμβατά με αυτά που παρέχονται από το Ubuntu Linux, μπορεί να ενσωματωθεί η δυνατότητα δημιουργίας τέτοιου τύπου λογαριασμών στο RAST.

8.2.2.2 Πρόσβαση σε ασύρματα δίκτυα

Το RAST μπορεί να επεκταθεί ώστε να δημιουργεί λογαριασμού για χρήση σε πρόσβαση ασύρματου δικτύου, ώστε ένας χρήστης να αποκτά πρόσβαση στο internet μέσω της τεχνολογίας Wi-Fi για περιορισμένα χρονικά διαστήματα. Οι λογαριασμοί που θα παράγονται από το RAST σε μια τέτοια περίπτωση θα εντάσσονται σε συστήματα ταυτοποίησης σαν αυτό της υπηρεσίας ασυρμάτου δικτύου (wireless) της τηλεματικής υπηρεσίας του Εθνικού Μετσόβιου Πολυτεχνείου [67].

8.2.2.3 Πρόσβαση σε proxy server, Εικονικά Ιδιωτικά Δίκτυα (VPN), Secure Shell (SSH)

Με την κατάλληλη επέκταση, οι δημιουργούμενοι από το RAST λογαριασμοί χρηστών μπορούν να ισχύσουν για μια σειρά από τεχνολογίες ασφαλούς πρόσβασης σε δικτυακές εφαρμογές. Συγκεκριμένα, αυτές οι τεχνολογίες είναι οι εξής:

- Secure Shell / SSH
    - Μέσω αυτού δίνεται η πρόσβαση σε συστήματα τύπου Unix, με ταυτόχρονη δυνατότητα πρόσβασης και ανταλλαγής αρχείων με τα πρωτόκολλα SFTP και SCP, δυνατότητα tunneling, εκτέλεσης εφαρμογών X11 [68], όπως και δυνατότητα προώθησης TCP Ports (TCP Port forwarding) [69] για μεταφορά εξειδικευμένων δικτυακών εφαρμογών που χρησιμοποιούν συνδέσεις TCP.
- Proxy server με ταυτοποίηση (authenticating proxy server)
    - Μέσω ενός proxy server [70] μπορεί να παραχωρηθεί δικαίωμα πρόσβασης σε web εφαρμογές με προσωρινό χαρακτήρα, όπως

και σε άλλες εφαρμογές που υποστηρίζονται από έναν proxy, όπως πολυμέσα κατά παραγγελία (multimedia on demand), μεταφορά πολυμέσων σε πραγματικό χρόνο (real time streaming).

- Εικονικό Ιδιωτικό Δίκτυο (Virtual Private Network / VPN) [71]
    - Η υπηρεσία αυτή επιτρέπει σε απομακρυσμένους χρήστες να αποκτούν πρόσβαση σε ένα τοπικό δίκτυο, αποκτώντας τοπική IP διεύθυνση και προσβάσεις όμοιες με αυτές που θα είχαν αν είχαν συνδεθεί τοπικά στο ιδεατό δίκτυο. Έτσι, με ενσωμάτωση αυτής της τεχνολογίας στο RAST μπορούν να προσφερθούν τέτοιες συνδέσεις με προσωρινό χαρακτήρα, όπως για παράδειγμα σύνδεση ενός εξωτερικού συνεργάτη σε δίκτυο της εταιρείας για τη διάρκεια μιας συνεργασίας.

8.2.3 Σύνδεση του συστήματος με συστήματα κεντρικής ταυτοποίησης

Το τελευταίο χρονικό διάστημα, μεγάλος αριθμός ιστοσελίδων απαιτεί ή εξυπηρετείται καλύτερα μέσω ταυτοποίησης του χρήστη. Έτσι, δημιουργείται το πρόβλημα της συντήρησης από ένα χρήστη μεγάλου αριθμού στοιχείων ταυτοποίησης σε πολλές δικτυακές σελίδες. Για την αντιμετώπιση αυτού του φαινομένου, έχουν δημιουργηθεί κεντρικές υπηρεσίες ταυτοποίησης, ώστε με στοιχεία ταυτοποίησης μόνο στις κεντρικές υπηρεσίες αυτές, ο χρήστης να αποκτά τη δικιά του ψηφιακή ταυτότητα, παρέχοντάς του τη δυνατότητα αυτομάτως να ταυτοποιηθεί σε μεγάλο αριθμό άλλων σελίδων.

Παραδείγματα τέτοιων συστημάτων είναι το Windows Live ID της Microsoft [72], το Facebook Connect [73] από την εταιρία Facebook, με πιο διαδεδομένο να είναι το OpenID [74] του IETF. Αυτά τα συστήματα παρέχουν προγραμματιστικές πλατφόρμες ώστε να είναι δυνατή η εύκολη σύνδεσή τους με άλλα υπάρχοντα συστήματα ταυτοποίησης.

Έτσι, το σύστημα RAST θα μπορούσε να επεκταθεί κατά τέτοιο τρόπο, ώστε να μπορεί να συνδέσει τους λογαριασμούς που δημιουργεί με υπάρχουσες ψηφιακές ταυτότητες, ώστε οι χρήστες του συστήματος να ταυτοποιούνται μέσω της υπάρχουσας ψηφιακής τους ταυτότητας, χωρίς την ανάγκη απομνημόνευσης ενός ακόμα κωδικού. Θα μπορούσε επιπλέον να μπει στην σελίδα του διαχειριστή δυνατότητα να προσθέτει χρήστες με βάση την ψηφιακή τους ταυτότητα, ή

εναλλακτικά δυνατότητα στον χρήστη, με την πρώτη εισαγωγή του στο σύστημα, να συνδέσει τον λογαριασμό του με την ταυτότητα.

8.2.4 Ενσωματωμένη πλατφόρμα (embedded platform)

Επειδή το σύστημα εκτελεί έναν αυτόνομο ρόλο, και με δυνατότητες απομακρυσμένης πρόσβασης, μπορούν να αναπτυχθεί με τέτοιο τρόπο ώστε να μην χρειάζεται σε κανένα σημείο φυσική πρόσβαση. Έτσι, από έναν εξειδικευμένο server, μπορεί να μεταφερθεί σε κάποια ενσωματωμένη πλατφόρμα [75], ώστε να αποκτήσει την μορφή μιας δικτυακής συσκευής, στη λογική «μαύρου κουτιού» (black box). Έτσι μπορεί να επιτευχθεί η απόκρυψη της μεγάλης πολυπλοκότητας του από τους τελικούς χρήστες, καθώς και η πιο εύκολη και άμεση εγκατάστασή του.

Ενσωματωμένες πλατφόρμες κυκλοφορούν στην αγορά με ποικίλους επεξεργαστές, και άλλα χαρακτηριστικά, όπως και σε διάφορα κόστη. Υπάρχουν πλατφόρμες βασισμένες στην αρχιτεκτονική x86, αυτή των προσωπικών υπολογιστών, που χρησιμοποιήθηκε στην ανάπτυξη του RAST, οπότε σε αυτές είναι εύκολη η μεταφορά του συστήματος. Υπάρχουν και άλλες πλατφόρμες, για τις οποίες προσφέρεται υποστήριξη από το Linux, αλλά θα πρέπει να γίνει έρευνα για την διαθεσιμότητα και τη συμβατότητα των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη του RAST.

Με αυτό τον τρόπο, το RAST μπορεί να μετατραπεί σε συσκευή δικτυακών υπηρεσιών, όμοια με συσκευές όπως οι συσκευές δικτυακής αποθήκευσης Network Attached Storage (NAS).

8.2.5 Σύνδεση του συστήματος με Cloud Computing εφαρμογές

Μια πρόσφατη εξέλιξη στον χώρο της πληροφορικής είναι οι νέες υπηρεσίες που προσφέρονται ηλεκτρονικά από σχετικές εταιρίες, ενώ το υλικό που τις εκτελεί βρίσκεται εγκατεστημένο σε δικούς τους χώρους και με δικιά τους ευθύνη συντήρησης. Έτσι, επιτυγχάνεται υψηλό επίπεδο υπηρεσιών, στις οποίες ο πελάτης έχει πρόσβαση μέσω του διαδικτύου, χωρίς την ανάγκη εγκατάστασης άλλου εξοπλισμού. Οι υπηρεσίες αυτές είναι γνωστές ως Cloud Computing Services ή Cloud Services [76].

Μια κατηγορία τέτοιων εφαρμογών είναι η προσφορά αποθηκευτικού χώρου, γνωστή και ως Cloud Storage. Παραδείγματα τέτοιων εφαρμογών είναι το Amazon S3 [77], το Rackspace Cloud Files [78] και το Pithos [79] του ΕΔΕΤ. Σε τέτοιες

λοιπόν υπηρεσίες μπορεί να συνδεθεί το RAST παρέχοντας πρόσβαση στους καταναλωτές που έχουν ενεργό λογαριασμό.

8.2.6 Εργαλείο ανάλυσης της καταγραφής του RAST και εξαγωγής στατιστικών

Το RAST έχει ένα εκτεταμένο σύστημα καταγραφής των συμβάντων, με μεγάλες λεπτομέρειες για κάθε διαδικασία που συμβαίνει σε αυτό. Έτσι, σε κάποια μεγάλη εγκατάσταση, ενώ είναι χρήσιμο εργαλείο για εξαγωγή συμπεράσματος σε κάποιο συγκεκριμένο συμβάν, ο μεγάλος όγκος της πληροφορίας που θα παρέχει θα δυσκολέψει κάποιον στην παρακολούθηση του συστήματος.

Προτείνεται λοιπόν να αναπτυχθεί εργαλείο ανάλυσης της καταγραφής αυτής, ομαδοποίησης των όμοιων συμβάντων, και με φιλική διεπαφή προς τον χρήστη. Ένα τέτοιο εργαλείο θα επέτρεπε στον διαχειριστή του συστήματος να το παρακολουθεί με μεγαλύτερη ευκολία, και να βγάζει πιο ακριβή συμπεράσματα για την καθημερινή λειτουργία του.

Προτείνεται προσθέτως να δημιουργηθεί εργαλείο εξαγωγής αναλυτικών στατιστικών χρήσης του RAST, με ταυτόχρονη εξαγωγή στατιστικών για το λειτουργικό σε όποιον σχετικό με την λειτουργία του RAST υπάρχουν στοιχεία (πχ χρήση μνήμης, χρήση επεξεργαστή, χρήση εύρους ζώνης του δικτύου). Με την βοήθεια αυτού του εργαλείου ο διαχειριστής θα είναι σε θέση να γνωρίζει την συνολική κίνηση, το ποσοστό αξιοποίησής του. Πρόσθετα, μπορεί να αναγνωρίσει την ανάγκη αναβάθμισης του συστήματος σε όποιον τομέα ενδεχομένως να υστερεί, βελτιώνοντας αισθητά την ποιότητα της υπηρεσίας που παρέχεται στους τελικούς χρήστες.

Με την ανάπτυξη αυτών των εργαλείων εκτιμάται ότι ο διαχειριστής θα είναι σε θέση να εξάγει αξιόπιστα συμπεράσματα που αφορούν την λειτουργία του συστήματος, αλλά και τα επίπεδα αξιοποίησης από τους τελικούς χρήστες. Συνδυαζόμενο με ενδεχόμενη σχετική έρευνα – ερωτηματολόγιο στους τελικούς χρήστες, θα μπορεί να παρέχει σημαντικά συμπεράσματα για την συνεισφορά του συστήματος σε όλες τις διαδικασίες που θα κληθεί να εξυπηρετήσει.

8.2.7 Συμβατότητα με συστήματα έξυπνου σπιτιού / DLNA

Το πρότυπο έξυπνου σπιτιού DLNA ορίζει τεσσάρων κατηγοριών συσκευές:

- Εξυπηρετητές Ψηφιακών Μέσων (Digital media servers – DMS)

- Συσκευές Αναπαραγωγής Ψηφιακών Μέσων (Digital media players – DMP)
- Ελεγκτές Ψηφιακών Μέσων (Digital media controllers – DMC)
- Μετατροπείς Ψηφιακών Μέσων (Digital media renderers – DMR)

Χωρίς να εισέλθουμε σε λεπτομέρειες για τις κατηγορίες συσκευών, οι οποίες τεκμηριώνονται επαρκώς στη βιβλιογραφία, το σύστημα RAST θα μπορούσε να εμπλουτιστεί ώστε να παρέχει την δυνατότητα να μετατρέπεται σε εξυπηρετητής ψηφιακών μέσων (DMS). Έτσι, αρχεία πολυμέσων αποθηκευμένα εντός του συστήματος θα μπορούν να διαμοιραστούν σε συμβατές συσκευές για αναπαραγωγή στο χρονικό περιθώριο που έχει δοθεί. Για να γίνει αυτό πρέπει στο σύστημα να προστεθεί η δυνατότητα συνεργασίας των συστημάτων ασφαλείας του RAST, και να προστεθεί το πρωτόκολλο επικοινωνίας του DLNA. Ένας τρόπος για να γίνει αυτό θα ήταν η υποστήριξη και συνεργασία του συστήματος με το λογισμικό TVMOBiLi [80], το οποίο υποστηρίζει την αντίστοιχη λειτουργία.

8.2.8 Σύνδεση του συστήματος με εξυπηρετητές FAX

Πολλοί εξυπηρετητές FAX του εμπορίου εγκαθίστανται ως οδηγοί εκτύπωσης. Επιπλέον, αντίστοιχοι οδηγοί υπάρχουν για τους πελάτες ενός τέτοιου εξυπηρετητή που επιτρέπουν στο χρήστη να εισάγει τον τηλεφωνικό αριθμό του προορισμού. Έτσι, λόγω της υποστήριξης εκτύπωσης, το RAST θα μπορούσε να επεκταθεί με μικρή προσπάθεια ώστε να υποστηρίξει σχετικούς οδηγούς, και εντός των πόρων που διαμοιράζονται να προστεθεί η δυνατότητα προσωρινής παροχής δυνατότητας για αποστολή FAX.

9

Επίλογος

Σε αυτή την εργασία παρουσιάστηκε ένα σύστημα που επιτρέπει την πρόσβαση σε πόρους και δεδομένα με αρχιτεκτονική, η οποία αν και στηρίζεται σε ένα κεντρικό σύστημα μέσω του οποίου διευκολύνεται η επικοινωνία, έχει χαρακτηριστικά δικτύων ισότιμων (peer to peer) χρησιμοποιώντας προσωρινά δικαιώματα πρόσβασης. Παρουσιάστηκαν επιπροσθέτως πραγματικές εφαρμογές και περιπτώσεις χρήσης με πιθανές επεκτάσεις τους για πραγματικά σενάρια. Το σύστημα είναι εύκολο να αξιοποιηθεί σε εγκαταστάσεις πολλών τύπων συσκευών, ακόμα και σε περιβάλλοντα ετερογενών τεχνολογιών.

Επιπρόσθετα στην κεντρική λειτουργικότητα του συστήματος, το σύστημα μπορεί να επεκταθεί για να παρέχει και γενικές λειτουργίες στους χρήστες. Στις ομοσπονδίες μεταξύ χρηστών και συσκευών, που είναι ουσιαστικά συλλογή τοπικών ή απομακρυσμένων προσωπικών κόμβων και συσκευών, είναι κρίσιμο να καθιερωθεί ένας μηχανισμός εμπιστοσύνης που θα διαφυλάττει την ασφάλεια των προσωπικών πόρων και δεδομένων, εξασφαλίζοντας ότι καμιά ανεπιθύμητη ομοσπονδία δεν θα πραγματοποιείται. Για να επιτευχθεί αυτό, προστατευόμενοι από σφάλματα κακόβουλης χρήσης και λάθη συστημάτων, απαιτούνται λειτουργίες διαχείρισης των χρηστών, των συσκευών και της μυστικότητας ώστε να γίνει ολόκληρη η αρχιτεκτονική περισσότερο αυτοματοποιημένη, απαιτώντας την ελάχιστη ανάμειξη του χρήστη, με εξαίρεση την πρώτη εγκατάσταση του συστήματος. Συνεπώς, οι χρήστες θα πρέπει να απαλλαχθούν από εργασίες με υψηλή επαναληπτικότητα και σχεδόν όλες οι διαχειριστικές δραστηριότητες πρέπει να ανέλθουν στο επίπεδο τους

συστήματος. Τέλος, η διαδικασία της δημιουργίας ομοσπονδιών πρέπει να επεκταθεί και να γίνει γρήγορη, εύκολη και φιλική προς το χρήστη καθώς στο μέλλον, οι ομοσπονδίες αναμένεται να είναι δραστηριότητα υψηλής συχνότητας [1].

Ένα πρώτο βήμα προς την υποστήριξη της δυνατότητας αυτής είναι η ολοκλήρωση ενός σχήματος διαχείρισης προφίλ, το οποίο μπορεί να είναι βασισμένο σε ιστοσελίδες (web based) και να υποστηρίζεται από την αντίστοιχη υπηρεσία, ώστε οι χρήστες να έχουν πρόσβαση σε όλες τις πληροφορίες που τους αφορούν (πχ τις ρυθμίσεις τους) και στα προφίλ των συσκευών τους. Ένα ακόμα βήμα είναι η δημιουργία ενός σχήματος διασφάλισης της μυστικότητας, όπου οι χρήστες μπορούν να επιλέξουν το προσωπικά δεδομένα που επιθυμούν να αποκαλύψουν στην αλληλεπίδραση με άλλους χρήστες, ώστε να προστατευθεί η μυστικότητα. Οι υπάρχοντες μηχανισμοί όπως το Cardspace της Microsoft [81] μπορούν να χρησιμοποιηθούν για τον σκοπό αυτό.

Ο αναμενόμενος αντίκτυπος από την επέκταση μιας τέτοιας λύσης θα είναι ορατός μέσω της ελαχιστοποίησης της αλληλεπίδρασης που απαιτείται για τον διαμοιρασμό πόρων μεταξύ των όμοιων συσκευών χρηστών, που παρακάμπτουν τις υπάρχουσες χρονοβόρες διαδικασίες, όπως η ανταλλαγή συγκεκριμένων πληροφοριών για την εγκατάσταση των αντίστοιχων οδηγών, καθώς και για την προσωρινή δημιουργία δικαιωμάτων και την αντίστοιχη ανάκληση των δικαιωμάτων μόλις πάψει να υπάρχει η ανάγκη για τη διαμοιρασμό των πόρων. Επιπλέον, ο αντίκτυπος στην ασφάλεια, δεδομένου ότι ο ιδιοκτήτης των πόρων δεν δίνει την πρόσβαση στους πόρους, αλλά σε εικονικούς πόρους σε επίπεδο συνδέσμων που δείχνουν τους πραγματικούς πόρους. Αυτές οι συνδέσεις μπορούν να δημιουργηθούν εύκολα, όπως και να αφαιρεθούν, καθώς και να σταματήσουν την πρόσβαση σε μια συσκευή.

Σε μια εποχή όπου η φορητότητα είναι καθεστώς, και πού οι κινητές και φορητές συσκευές έχουν καθιερώσει ήδη τη θέση τους στα χέρια μας, το σύστημα RAST, όπως προτάθηκε μέσα από την εργασία αυτή, μπορεί να βοηθήσει ουσιαστικά στην επιτάχυνση των χρόνων συνεργασίας σε επαγγελματικά περιβάλλοντα.

10

Βιβλιογραφία

[1] Ch. Z. Patrikakis, I. G. Nikolakopoulos, A. Skoufis, S. Stamokostas, "Safe access to computing resources in personal networking environments", accepted for poster presentation to the ICT-MobileSummit, Stockholm, Sweden, June 2008.

[2] C. Patrikakis, D. Kyriazanos, and N. Prasad, “Establishing Trust Through Anonymous and Private Information Exchange Over Personal Networks,” *Wireless Personal Communications*, vol. 51, no. 1, pp. 121-135, 2009.

[3] "Smart device - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Smart_device.

[4] "Wi-Fi - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Wifi.

[5] "Bluetooth - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Bluetooth.

[6] "3G - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/3g.

[7] "Mesh networking - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Mesh_network.

[8] "One Laptop per Child (OLPC), a low-cost, connected laptop for the world's children's education," http://laptop.org/en/.

[9] "ZigBee - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Zigbee.

[10] "Universal Plug and Play - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Upnp.

[11] "Home - DLNA," http://www.dlna.org/home.

[12] "Active Directory - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Active_Directory.

[13] M. Bradley. "Workgroup - Workgroup in Computer Networking," http://compnetworking.about.com/cs/design/g/bldef_workgroup.htm.

[14] "Peer-to-peer - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Peer-to-peer.

[15] "Social network service - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Social_network_service.

[16] C. Z. Patrikakis, D. M. Kyriazanos, A. S. Voulodimos *et al.*, "Privacy and resource protection in personal network federations." pp. 1-5.
[17] "Server Message Block - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Cifs.
[18] "cron - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Cron.
[19] "About the Open Source Initiative | Open Source Initiative," http://www.opensource.org/about.
[20] "The Open Source Definition | Open Source Initiative," http://www.opensource.org/docs/osd.
[21] "Ελεύθερο Λογισμικό / Λογισμικό ανοιχτού κώδικα," http://www.ellak.gr/index.php?option=com_openwiki&Itemid=103&id=ellak:τι_είναι_το_ελλακ.
[22] "Linux Online - GNU General Public License," http://www.linux.org/info/gnu.html.
[23] "Samba - opening windows to a wider world," http://samba.anu.edu.au/samba/.
[24] "Software License Agreement - Documentation - CUPS," http://www.cups.org/documentation.php/doc-1.4/license.html.
[25] "Licenses - The Apache Software Foundation," http://www.apache.org/licenses/.
[26] "PHP License - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/PHP_License.
[27] "Perl Licensing - dev.perl.org," http://dev.perl.org/licenses/.
[28] "MySQL :: MySQL Licensing Policy," http://www.mysql.com/about/legal/licensing/index.html.
[29] "Linux Online - About the Linux Operating System," http://www.linux.org/info/.
[30] "Linux - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Linux.
[31] "Linux History," http://www.tamos.net/ieee/history.html.
[32] "History of Linux - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/History_of_Linux.
[33] "Why Linux is better," http://www.whylinuxisbetter.net/index_el.php?lang=el.
[34] "Ubuntu-gr | Η ελληνική κοινότητα του Ubuntu," http://ubuntu-gr.org/.
[35] "Ubuntu Home Page | Ubuntu," http://www.ubuntu.com/.
[36] "ubuntistas," http://ubuntistas.ubuntu-gr.org/.
[37] "Ubuntu - Βικιπαίδεια," http://el.wikipedia.org/wiki/Ubuntu.
[38] "Introducing Samba," http://www.linuxjournal.com/article/2716.
[39] "Samba (software) - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Samba_(software).
[40] "CUPS - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/CUPS.
[41] "Εναλλακτικές Λύσεις για τον Στάνταρ Spooler," http://www.freebsd.org/doc/el/books/handbook/printing-lpd-alternatives.html.
[42] "Why Apache Server is Better?," http://www.articlesbase.com/internet-articles/why-apache-server-is-better-648685.html.
[43] "PHP: Hypertext Preprocessor," http://www.php.net/.
[44] "phpworld," http://www.phpworld.com/.
[45] A. Trachtenberg. "Why PHP 5 Rocks! - O'Reilly Media," http://onlamp.com/pub/a/php/2004/07/15/UpgradePHP5.html.

[46] "Lessons Learned: Why PHP won," http://www.startuplessonslearned.com/2009/01/why-php-won.html.
[47] "The Perl Programming Language - www.perl.org," http://www.perl.org/.
[48] "Why Perl is a Valid Choice - www.perl.org," http://www.perl.org/advocacy/whyperl.html.
[49] "MySQL :: The world's most popular open source database," http://www.mysql.com/.
[50] "MySQL :: Why MySQL?," http://www.mysql.com/why-mysql/.
[51] "FAUS | freshmeat.net," http://freshmeat.net/projects/faus.
[52] "Download FAUS 1.4.5 for Linux - FAUS is a Perl CGI to permit user administration through a Web interface. - Softpedia," http://linux.softpedia.com/get/System/System-Administration/FAUS-8721.shtml.
[53] "OpenLDAP, Main Page," http://www.openldap.org/.
[54] V. Koutsonikola, and A. Vakali, "LDAP: Framework, practices, and trends," *IEEE INTERNET COMPUTING*, vol. 8, no. 5, pp. 66-72, 2004.
[55] S. E. Randall. "Samba authentication through PAM with MySQL," http://www.freebsddiary.org/samba-pam.php.
[56] "Webmin," http://www.webmin.com/.
[57] "Network File System (protocol) - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Network_File_System_(protocol).
[58] "File Transfer Protocol - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Ftp.
[59] "WebDAV - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Webdav.
[60] "BytesFall Explorer," http://bfexplorer.sourceforge.net/.
[61] "Internet Printing Protocol - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Internet_Printing_Protocol.
[62] "Ghostscript - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Ghostscript.
[63] "Supercomputer - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Supercomputer.
[64] "Cluster (computing) - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Cluster_(computing).
[65] "Grid computing - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Grid_computing.
[66] "Secure Shell - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Secure_Shell.
[67] "ΚΕΝΤΡΟ ΔΙΚΤΥΩΝ Ε.Μ.Π. :: Χτίζοντας το Ακαδημαϊκό Δίκτυο," http://www.noc.ntua.gr/index.php?module=ContentExpress&func=display&btitle=CE&mid=&ceid=106.
[68] "X Window System - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/X11.
[69] "Port forwarding - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Port_forwarding.
[70] "Proxy server - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Proxy_server.
[71] "Virtual private network - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Vpn.

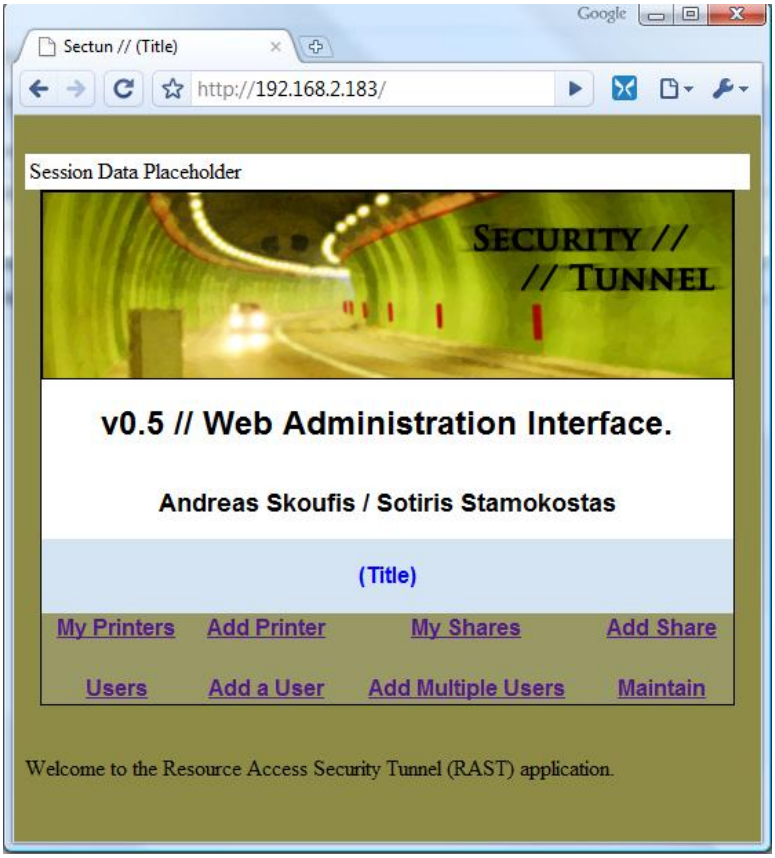
[72] "Windows Live ID - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Windows_Live_ID.
[73] "Facebook Developers | Facebook Connect," http://developers.facebook.com/connect.php.
[74] "OpenID - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Openid.
[75] "Embedded system - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Embedded_system.
[76] "Cloud computing - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Cloud_services#Services.
[77] "Amazon Simple Storage Service (Amazon S3)," http://aws.amazon.com/s3/.
[78] "Cloud Files - Unlimited Online Storage & CDN File Delivery," http://www.rackspacecloud.com/cloud_hosting_products/files.
[79] "Pithos," http://pithos.grnet.gr/.
[80] "TVMOBiLi, a free UPnP and DLNA Media Server For Mac, Windows, and Linux," http://www.tvmobili.com/.
[81] "Windows CardSpace - Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/Cardspace.

11

Παραρτήματα

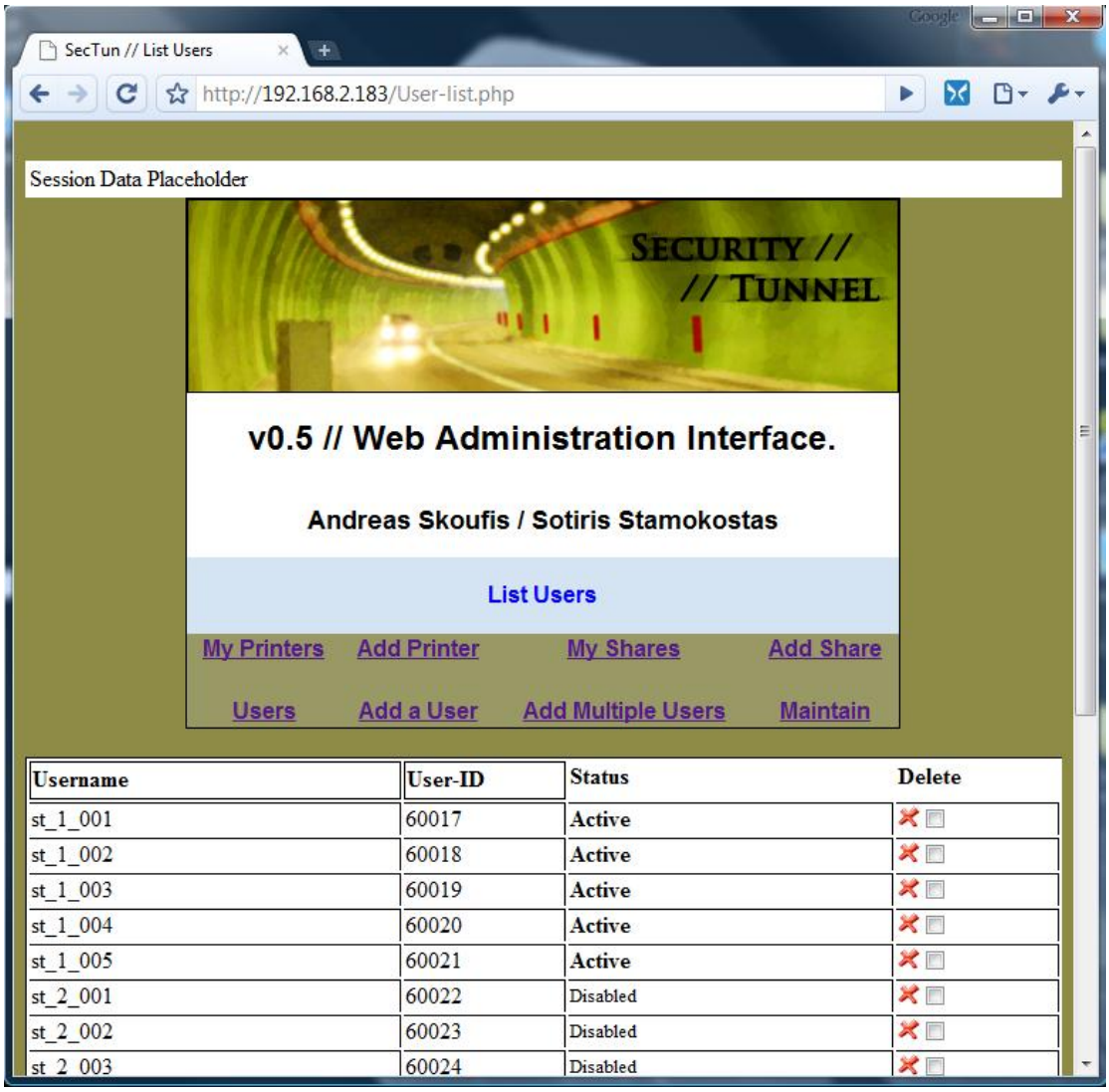
11.1 Εγχειρίδιο χρήσης

Ακολουθεί σύντομη επίδειξη και εγχειρίδιο της εφαρμογής.



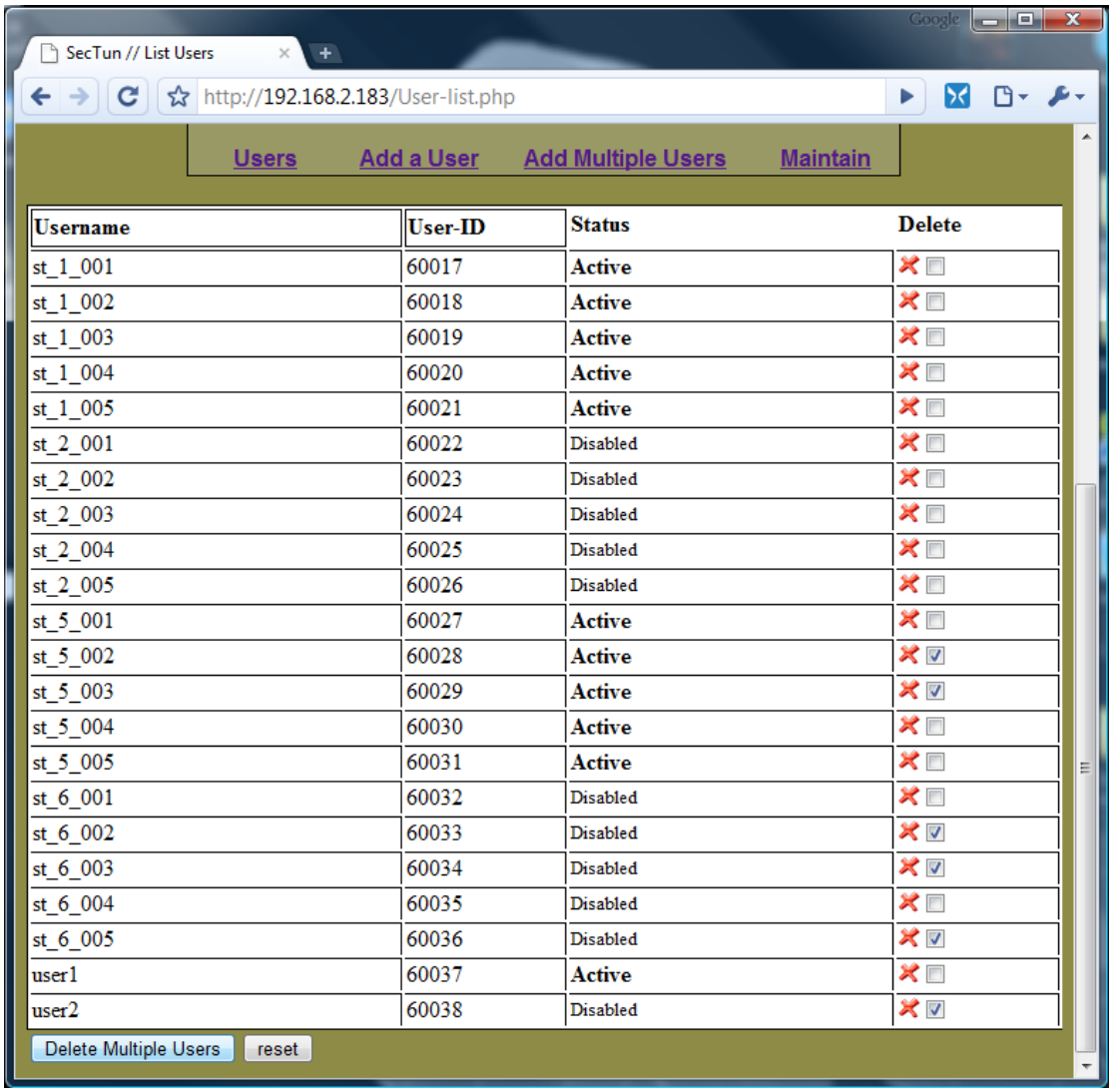
Σχήμα 30: Εισαγωγική οθόνη του RAST

Με την σύνδεσή του ο χρήστης του RAST βλέπει την παραπάνω σελίδα, όπου του παρουσιάζονται οι διαθέσιμες σελίδες για την διαμόρφωση του συστήματος.



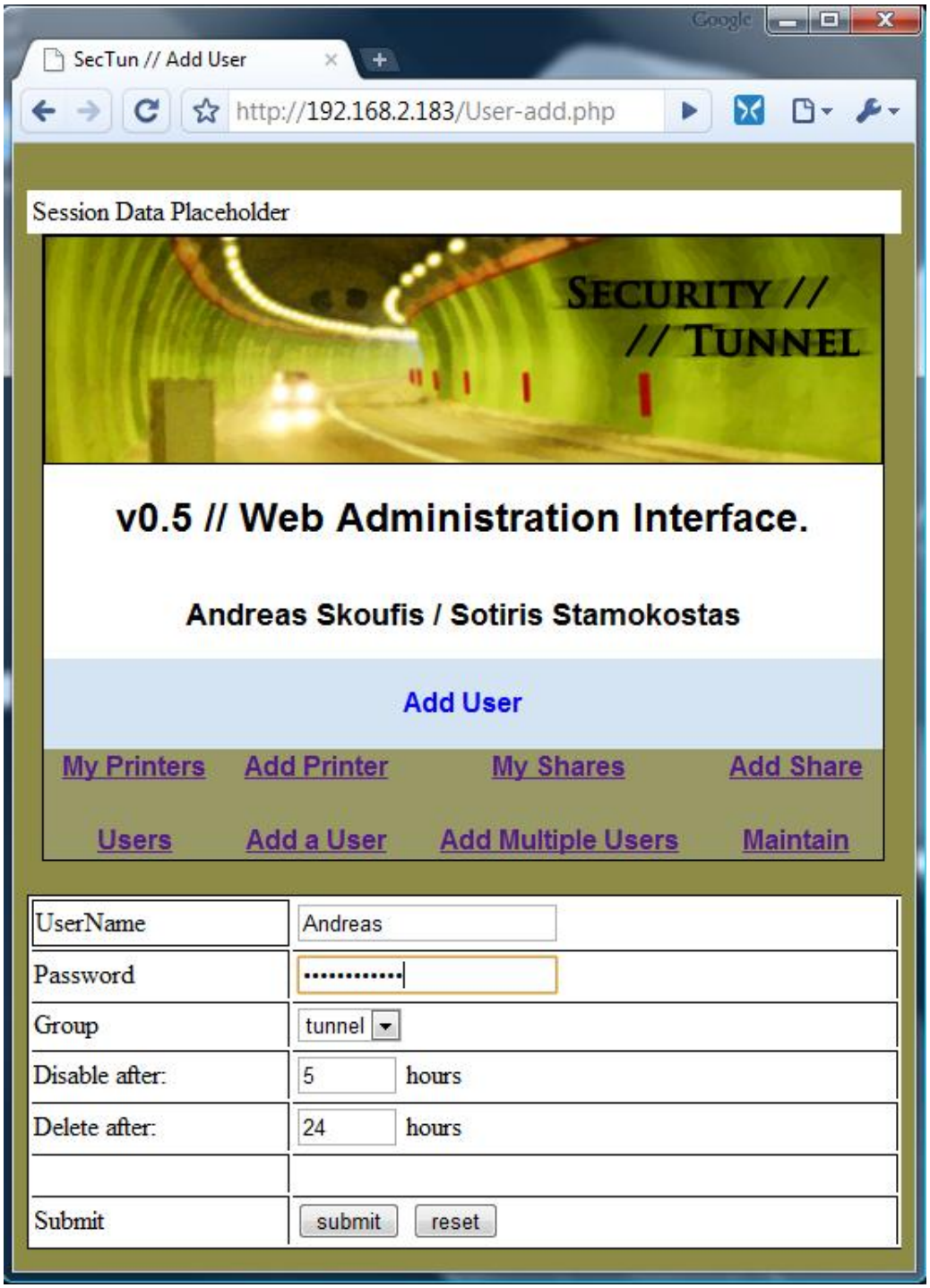
Σχήμα 31: Λίστα χρηστών

Όταν ο χρήστης επιλέξει την σελίδα Users, του παρουσιάζεται λίστα με τους χρήστες του συστήματος, είτε ενεργούς (Active), είτε ανενεργούς (Disabled). Επιλέγοντας το σήμα X δίπλα σε κάθε χρήστη δίνεται η δυνατότητα διαγραφής μεμονωμένα κάθε χρήστη.



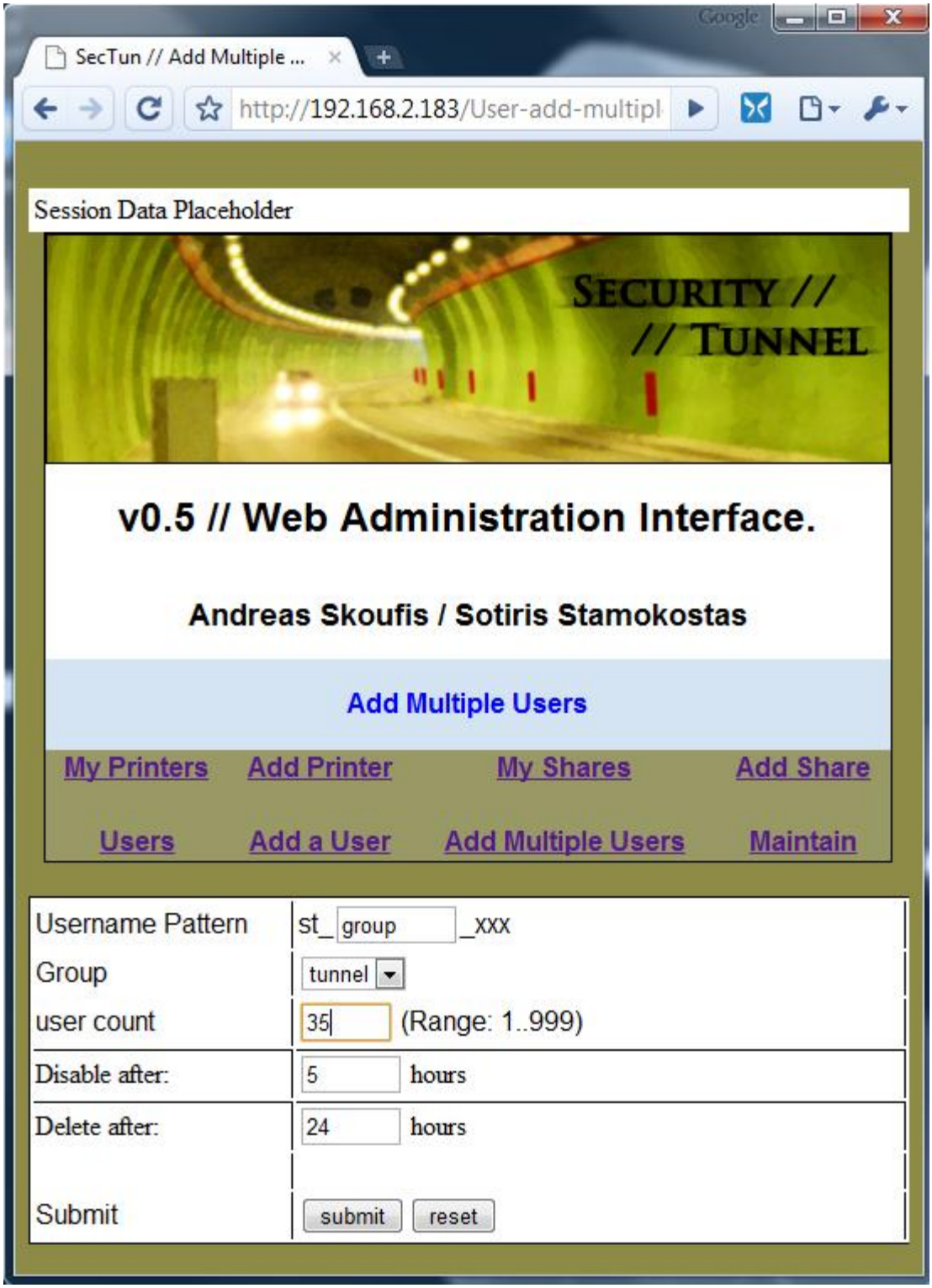
Σχήμα 32: Μαζική διαγραφή χρηστών

Επιλέγοντας τα μικρά κουτάκια όπως φαίνεται στην εικόνα και πατώντας το πλήκτρο Delete Multiple Users επιτυγχάνεται η μαζική διαγραφή χρηστών. Ακολουθεί σελίδα επιβεβαίωσης που παρουσιάζονται όλοι οι προς διαγραφή χρήστες.



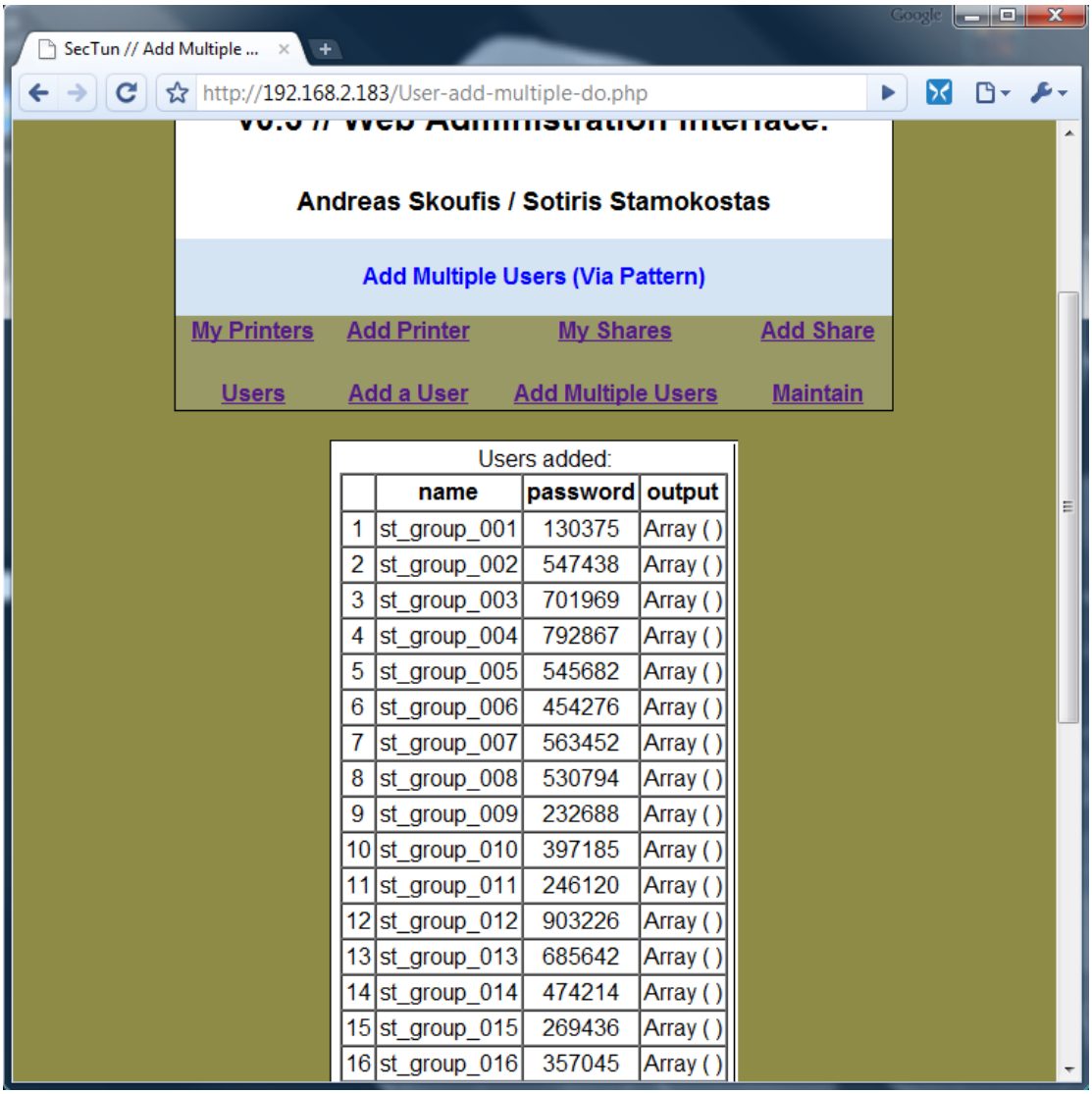
Σχήμα 33: Προσθήκη μεμονωμένου χρήστη

Στην σελίδα Add a User μπορεί να προστεθεί μεμονωμένα ένας χρήστης, με συγκεκριμένο χρόνο απενεργοποίησης (Disable after, στο παράδειγμά μας είναι 5 ώρες μετά την δημιουργία του) και διαγραφής (Delete After, 24 ώρες εδώ). Ο Διαχειριστής πρέπει να εισάγει όνομα χρήστη και κωδικό πρόσβασης (username & password).



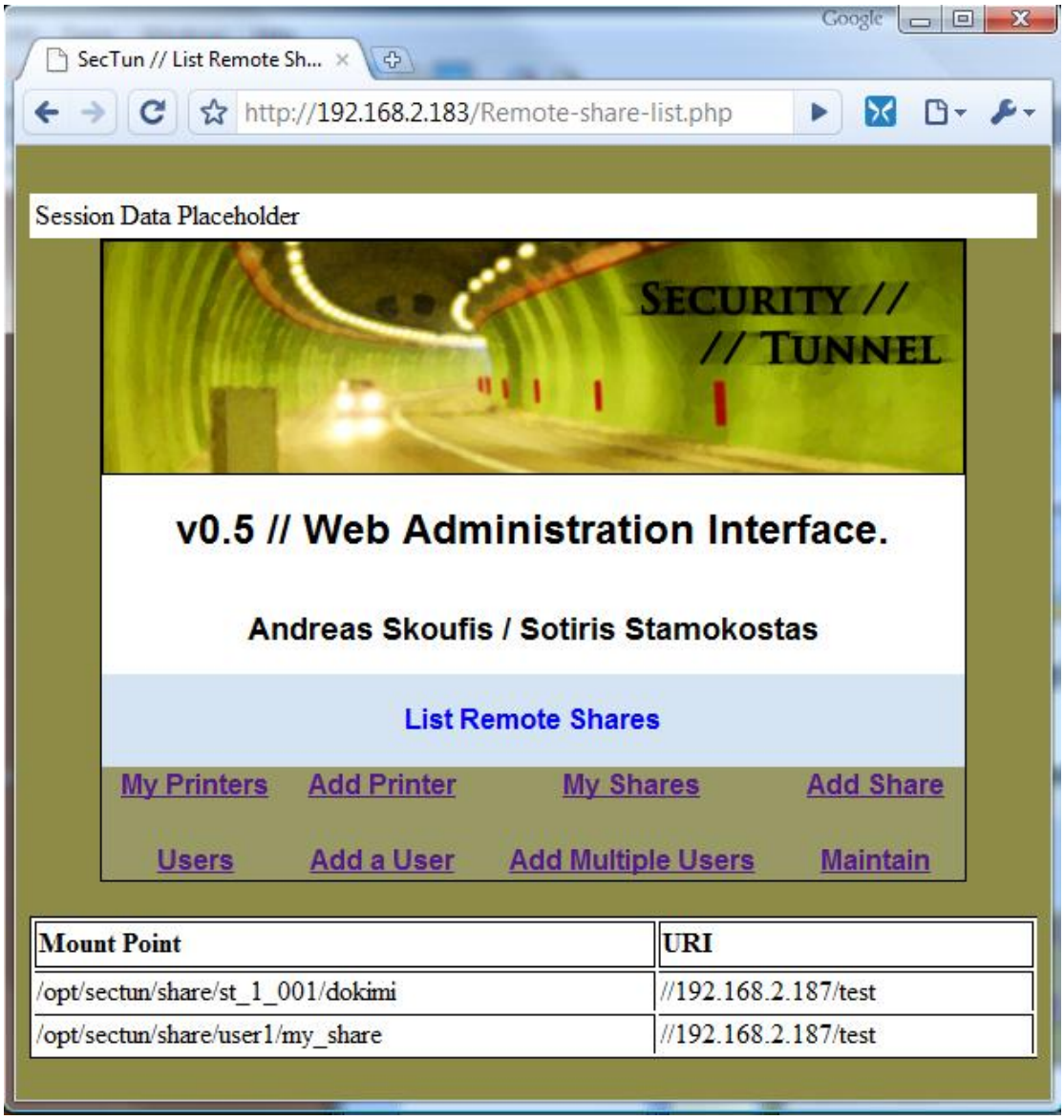
Σχήμα 34: Μαζική προσθήκη χρηστών

Από την σελίδα Add Multiple Users, ο διαχειριστής μπορεί να προσθέσει group χρηστών, από 1 μέχρι 999. Ονομάζει το group και προκύπτουν οι χρήστες όπως φαίνεται στο παρακάτω σχήμα.



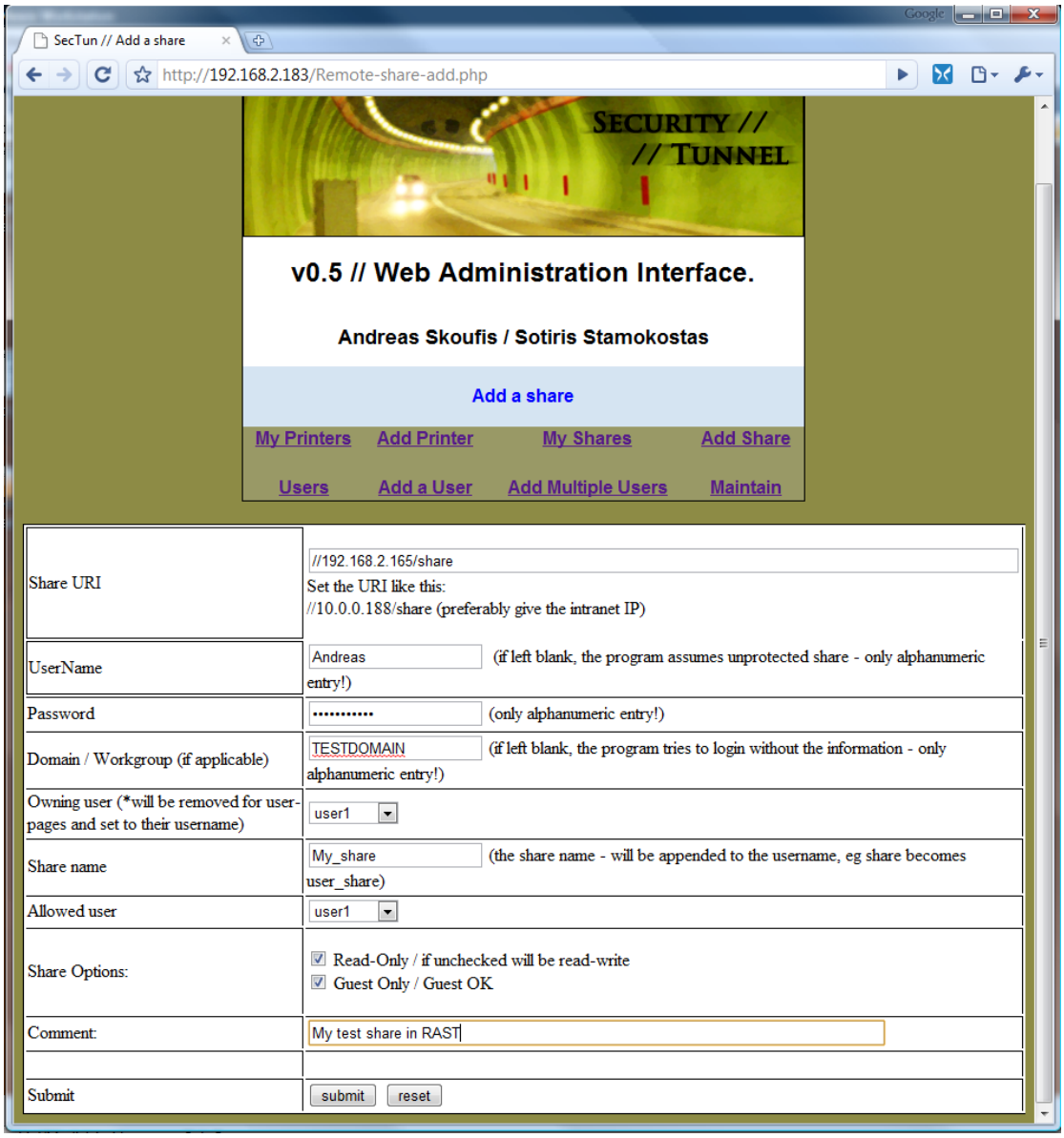
Σχήμα 35: Αποτέλεσμα μαζικής προσθήκης χρηστών

Στο παραπάνω σχήμα φαίνεται το αποτέλεσμα της μαζικής προσθήκης χρηστών. Το σύστημα δημιουργεί αυτόματα τους χρήστες, καθώς και τυχαίους εξαψήφιους κωδικούς αποτελούμενους από νούμερα (η αλλαγή σε δημιουργία πιο απλών κωδικών γίνεται με απλή αλλαγή του κώδικα της εφαρμογής).



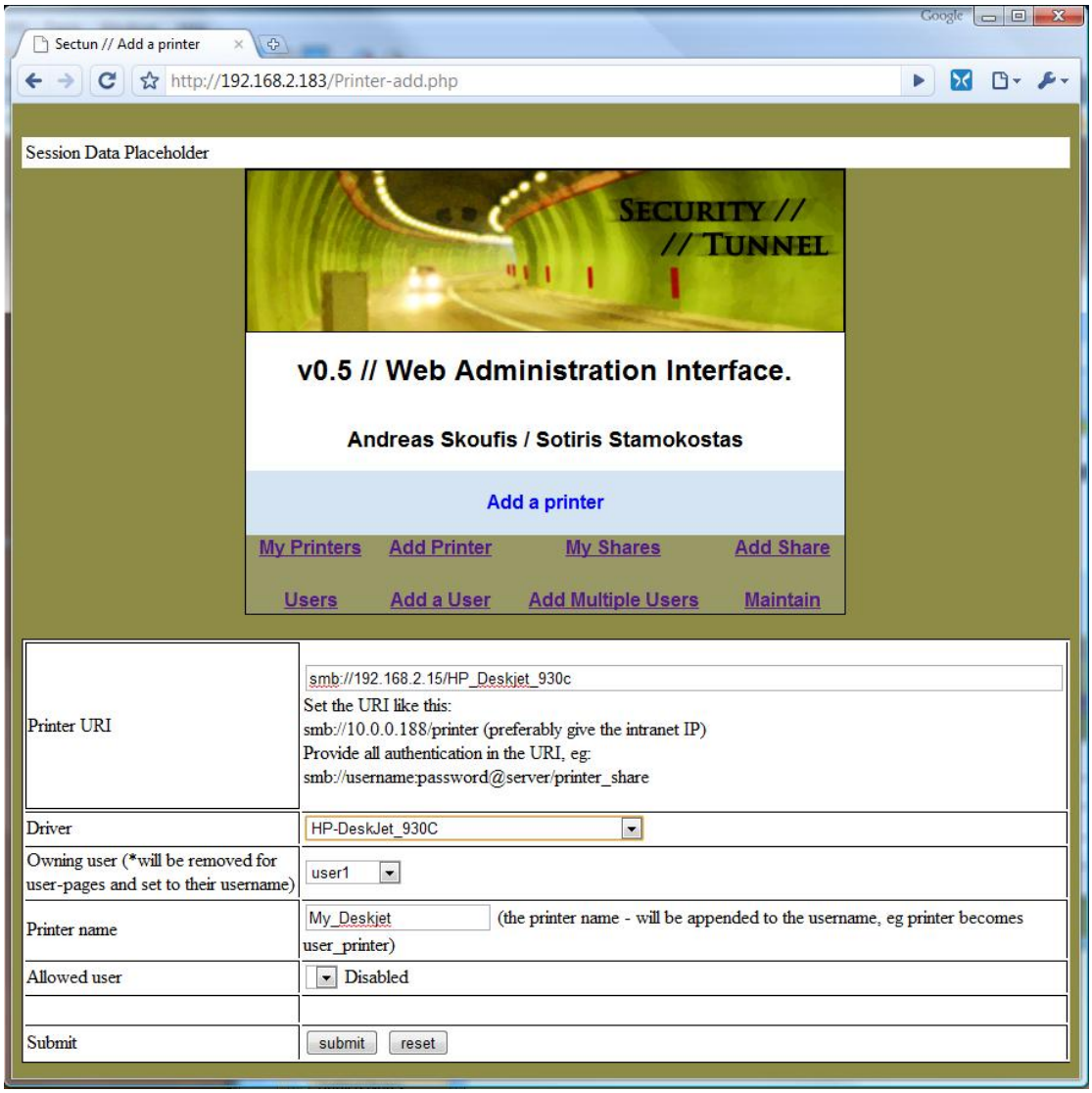
Σχήμα 36: Διαθέσιμοι φάκελοι του συστήματος

Στην σελίδα My Shares φαίνονται οι μοιραζόμενοι στο σύστημα φάκελοι. Στο πεδίο Mount Point φαίνεται ο εικονικός φάκελος που έχει δημιουργηθεί για την σύνδεση, ενώ στο πεδίο URI φαίνεται η διεύθυνση του μοιραζόμενου φακέλου στον υπολογιστή / συσκευή του παρόχου.



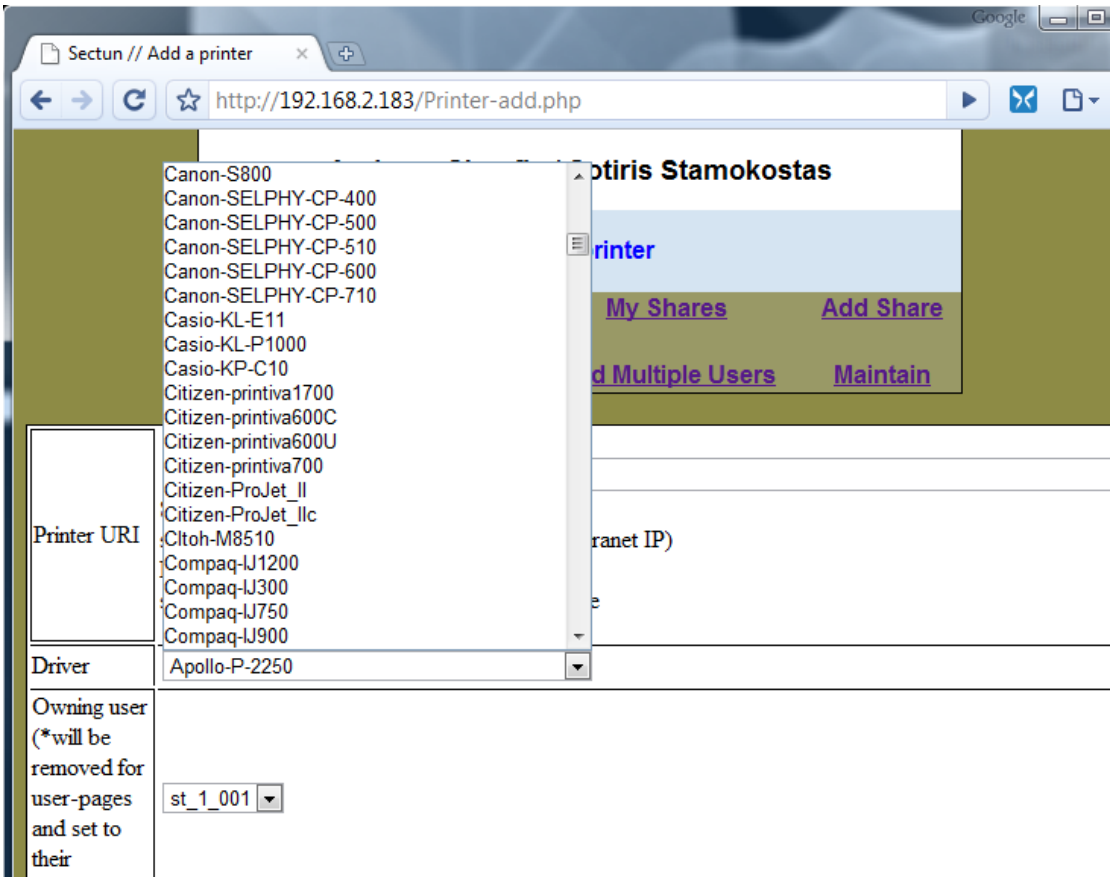
Σχήμα 37: Προσθήκη μοιραζόμενου φακέλου στο σύστημα

Στη σελίδα Add Share, ο πάροχος μπορεί να προσθέσει έναν φάκελο στο σύστημα. Συμπληρώνει την διαδρομή για τον φάκελο (//computer_ip/shared_folder, δηλαδή σαν τις διαδρομές στο λειτουργικό Windows με ανεστραμμένες τις καθέτους), το όνομα χρήστη, τον κωδικό και το Domain ή Workgroup στο οποίο ανήκει (οι τελευταίες τρεις πληροφορίες ενδεχομένως δεν είναι απαραίτητες αν ο διαμοιρασμός είναι ρυθμισμένος ώστε να δέχεται επισκέπτες), το όνομα χρήστη στο σύστημα RAST, το όνομα με το οποίο θέλει να μοιραστεί ο φάκελος, τον χρήστη που θα επιτρέψει την πρόσβαση, αν ο φάκελος θα είναι μόνο για ανάγνωση, αν θα επιτρέπεται η πρόσβαση σε επισκέπτες καθώς και ένα σχόλιο. Πατώντας το πλήκτρο submit, και εφόσον τα στοιχεία είναι σωστά, ο φάκελος συνδέεται στο RAST.



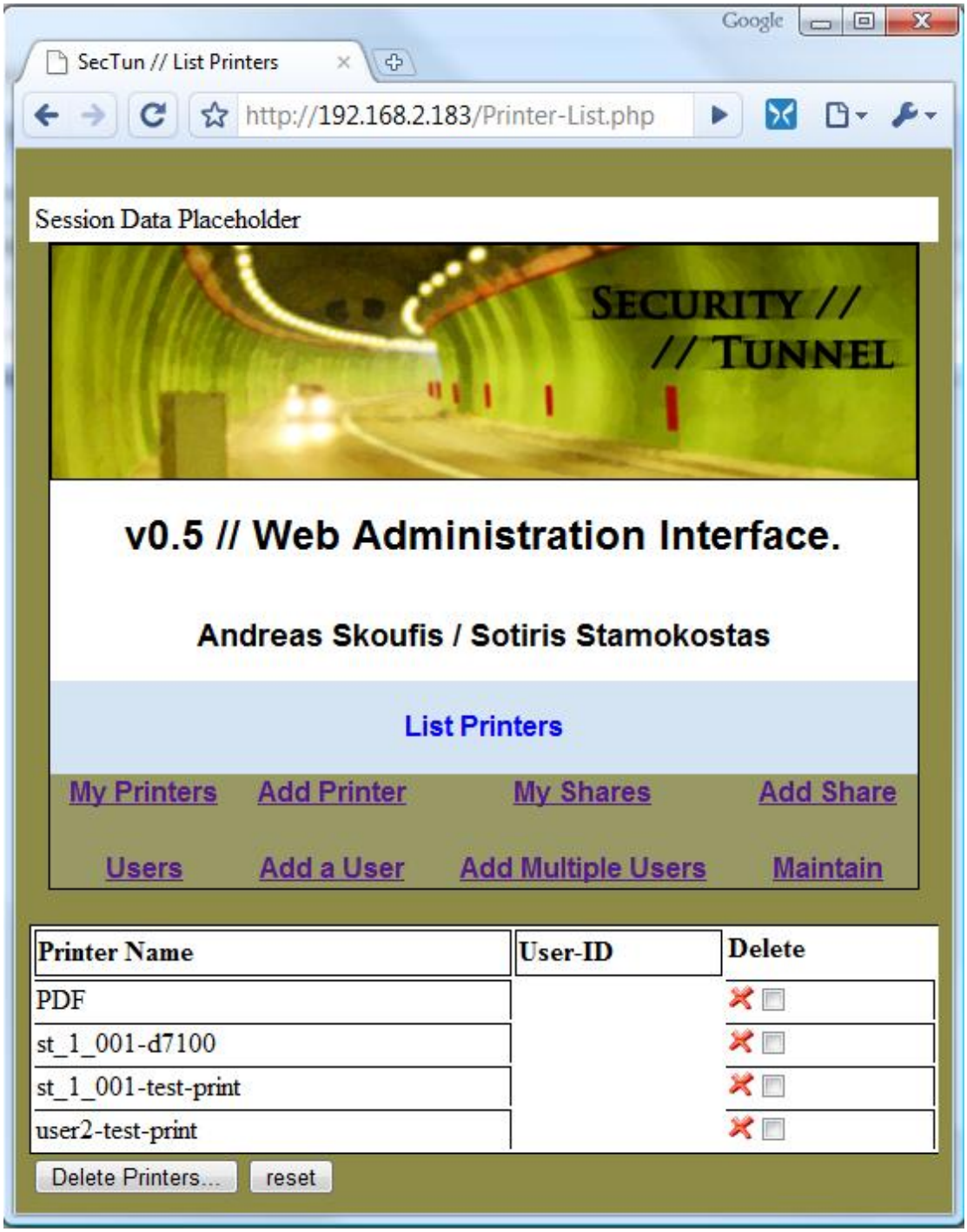
Σχήμα 38: Σελίδα προσθήκης εκτυπωτή

Στη σελίδα Add Printer, ο πάροχος μπορεί να προσθέσει έναν εκτυπωτή στο σύστημα. Συμπληρώνοντας το πλήρες URI του εκτυπωτή, τον οδηγό, τον όνομα χρήστη του παρόχου, καθώς και το όνομα με το οποίο θα μοιραστεί στο σύστημα ο εκτυπωτής, προστίθεται ο εκτυπωτής στο σύστημα. Στη συγκεκριμένη σελίδα, αντίθετα από την σελίδα προσθήκης φακέλου, χρειάζεται να μπει το πλήρες URI, όπως φαίνεται στο παράδειγμα. Κι αυτό για να μπορούν να υποστηριχθούν κι άλλου τύπου μοιραζόμενοι εκτυπωτές, όπως πχ. IP Printers, μέσω του πρωτοκόλλου IPP.



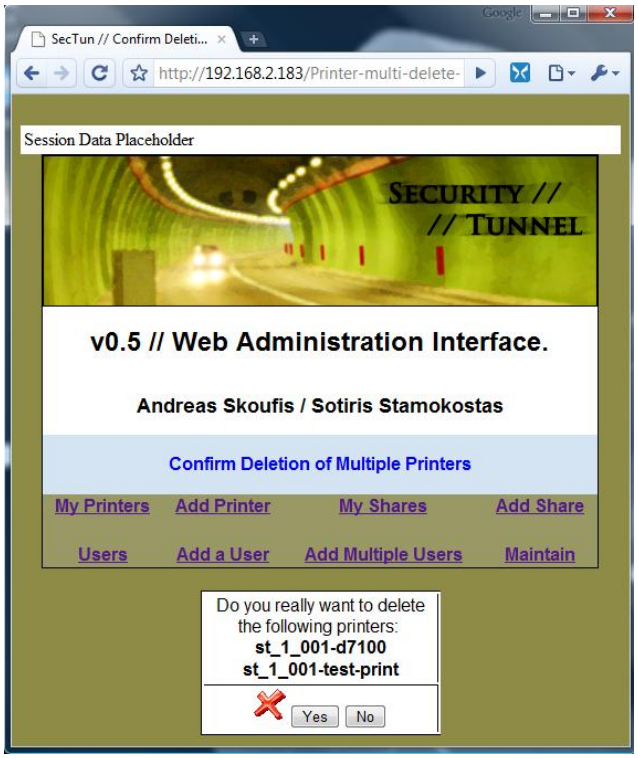
Σχήμα 39: Επιλογή οδηγού για εκτυπωτή

Όπως φαίνεται και στο σχήμα, το σύστημα υποστηρίζει εκατοντάδες διαφορετικούς εκτυπωτές μέσω των εγκατεστημένων οδηγών που έχει. Η υποστήριξη περεταίρω εκτυπωτών γίνεται με ενημέρωση των συστημάτων CUPS και foomatic.

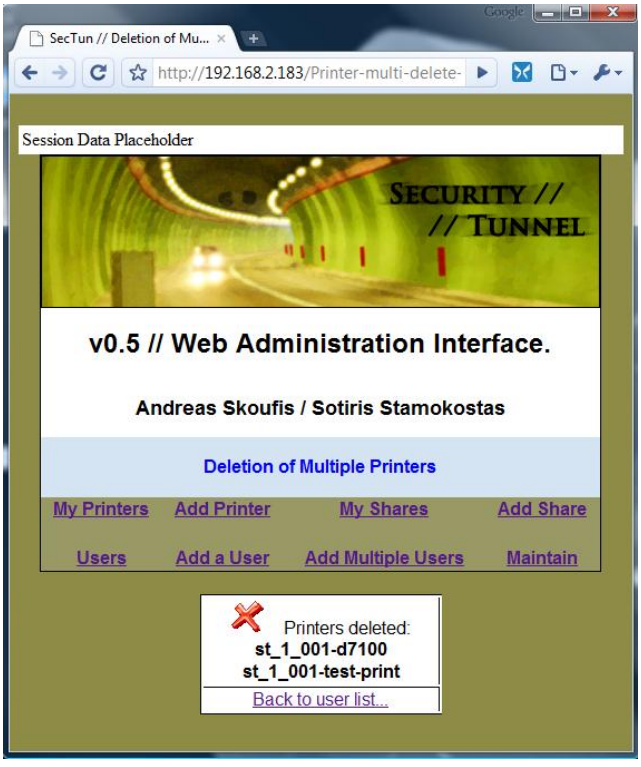


Σχήμα 40: Λίστα εκτυπωτών στο σύστημα

Στην σελίδα My Printers φαίνονται οι συνδεδεμένοι στο σύστημα εκτυπωτές. Επιλέγοντας στα κουτάκια στην στήλη Delete και πατώντας το πλήκτρο Delete Printers ... επιτυγχάνεται η διαγραφή των εκτυπωτών, με βήματα επιβεβαίωσης που φαίνονται παρακάτω.

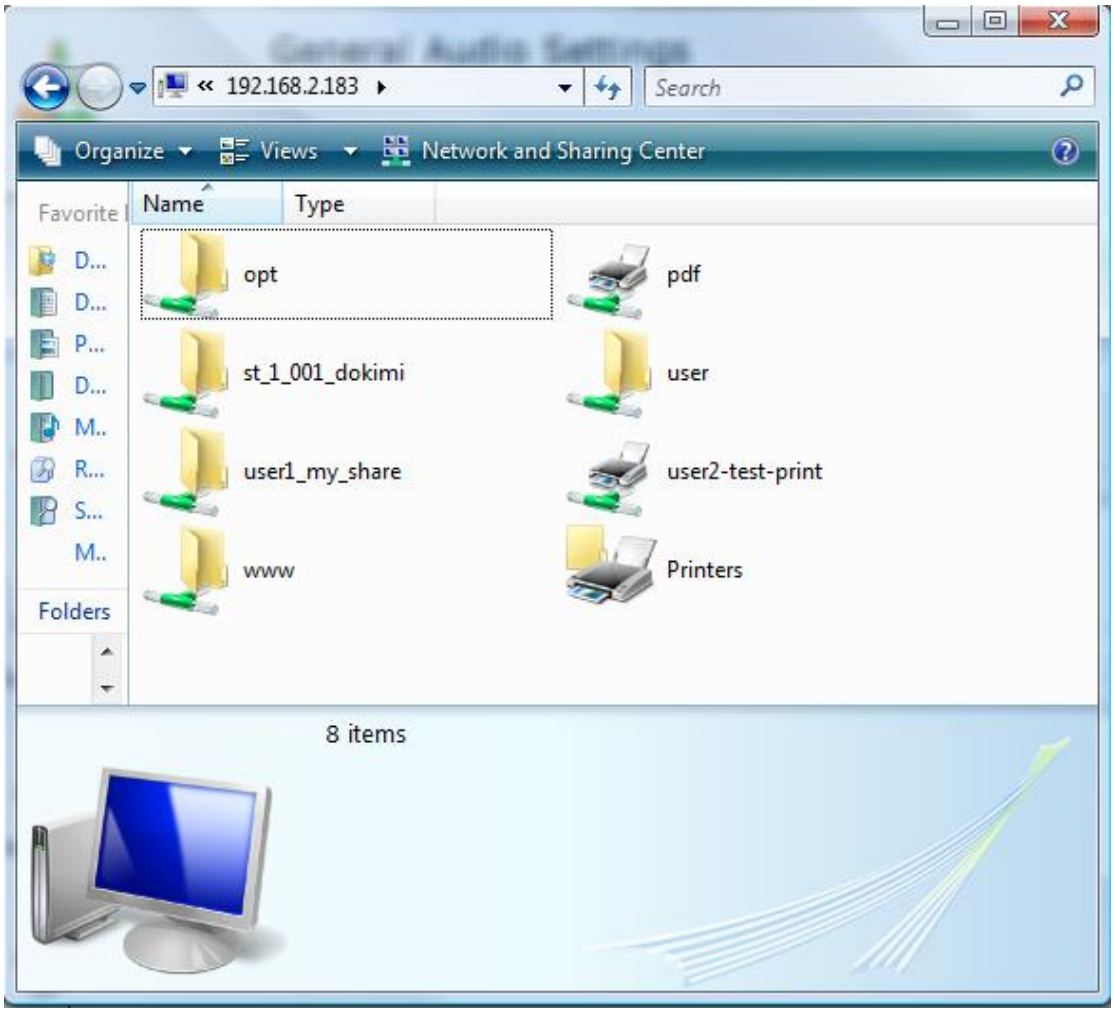


Σχήμα 41: Λίστα εκτυπωτών προς διαγραφή



Σχήμα 42: Επιβεβαίωση διαγραφής εκτυπωτών

Στα παραπάνω δυο σχήματα φαίνεται η διαδικασία επιβεβαίωσης της διαγραφής, με λίστα των εκτυπωτών προς διαγραφή. Αντίστοιχη είναι και η διαδικασία επιβεβαίωσης διαγραφής πολλαπλών χρηστών.



Σχήμα 43: Σύνδεση καταναλωτή στο RAST

Για να συνδεθεί ένας πελάτης με λειτουργικό Windows στο RAST και να δει τους πόρους στους οποίους έχει πρόσβαση, πρέπει να πάει στη διεύθυνση \\RASTSERVER\, όπου RASTSERVER είναι η διεύθυνση του υπολογιστή που τρέχει το RAST. Φαίνεται ότι ο χρήστης δεν έχει γνώση των διευθύνσεων των λοιπών υπολογιστών που παρέχουν πόρους στο σύστημα. Για να εγκαταστήσει έναν εκτυπωτή αρκεί ένα διπλό κλικ στον εκτυπωτή, και το RAST θα του στείλει τον απαραίτητο postscript οδηγό, και σε κάθε εκτύπωση θα αναλάβει να μετατρέψει την postscript στην γλώσσα του αντίστοιχου εκτυπωτή, αξιοποιώντας τους οδηγούς που περιλαμβάνονται σε αυτό. Από τον χρήστη δεν απαιτείται κανένα βήμα για την εγκατάσταση του οδηγού.

11.2 Κώδικας εφαρμογής

Index.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>
<?php require_once("/var/www/options/options.php"); ?>
<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Sectun // (Title)</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>
<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>(Title)</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
```

```
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<p>Welcome to the Resource Access Security Tunnel (RAST)
application.</p>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>
return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Debug-Cronscript.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>
<?php require_once("/var/www/options/options.php"); ?>
<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Sectun // test</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>
```

```
<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>test</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<p><pre>
<?
//    print_r (unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php remoteshare-list')));
//    print_r (unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php printer-add')));
      print_r (unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/cronscript.php execute')));
      print_r (unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/cronscript.php execute')));
      $return= unserialize(exec("sudo /usr/bin/php-cgi -q
/opt/sectun/server.php debug-reload-samba"));
?>
</pre></p>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
```

```
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>
return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Printer-add.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Sectun // Add a printer</title>
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var
i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2];
val=document.getElementById(args[i]);
      if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
          if (p<1 || p==(val.length-1)) errors+='- '+nm+' must
contain an e-mail address.\n';
        } else if (test!='R') { num = parseFloat(val);
          if (isNaN(val)) errors+='- '+nm+' must contain a
number.\n';
          if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
            if (num<min || max<num) errors+='- '+nm+' must contain a
number between '+min+' and '+max+'.\n';
      } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is
required.\n'; }
```

```
    } if (errors) alert('The following error(s) occurred:\n'+errors);
    document.MM_returnValue = (errors == '');
} }
//-->
</script>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>Add a printer</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
```

```
<form action="Printer-add-do.php" method="get" name="Prn-add-form"
id="Prn-add-form">
<table style="width: 100%" class="ms-grid1-main">
  <!-- MSTableType="nolayout" -->
  <!-- fpstyle: 16,011111100 -->
  <tr>
    <td style="width: 30%" class="ms-grid1-tl">Printer URI</td>
    <td valign="top" class="ms-grid1-top"><p>
      <input name="PrinterURI" type="text" id="PrinterURI" size="100"
maxlength="100" />
      <br />
      Set the URI like this: <br />
      smb://10.0.0.188/printer (preferably give the intranet IP)<br
/>
      Provide all
    authentication in the URI, eg:<br />
    smb://username:password@server/printer_share
    </p></td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left">Driver</td>
    <td class="ms-grid1-even"><select name="driver" id="driver">
      <?

      $file = "/opt/sectun/printdrivers.txt";
      $f = fopen($file, "r");
      while ( $line = fgets($f, 1000) ) {
            print "<option>".$line."</option>\n";
      }

?>
    </select>
       </td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left">Owning user (*will
be removed for user-pages and set to their username)</td>
    <td class="ms-grid1-even"><select name="owner" id="owner">
      <?
      $return = unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-list'));
      foreach ($return[users] as $value) {
            print "<option>".$value[username]."</option>\n";
      }

?>
    </select>
       </td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left">Printer name</td>
    <td class="ms-grid1-even"><input type="text" name="printername"
id="printername" />
      (the printer name - will be appended to the username, eg
printer becomes user_printer)</td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left">Allowed user</td>
    <td class="ms-grid1-even"><select name="allowed" id="allowed">
      <? /*
      foreach ($return[users] as $value) {
```

```
            print "<option>".$value[username]."</option>\n";
      } */

?>
      </select> Disabled</td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left"> </td>
    <td class="ms-grid1-even"> </td>
  </tr>
  <tr>
    <td style="width: 30%" class="ms-grid1-left">Submit</td>
    <td class="ms-grid1-even"><input name="Submit1" type="submit"
value="submit" />
      <input name="Reset1" type="reset" value="reset" /></td>
  </tr>
</table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Printer-add-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
```

```
<title>SecTun // Add a printer</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Add a printer</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
    <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
```

```
                  <td style="width: 24%; height: 21px;"
class="style2">
                     
<?
//receive the form variables:
$share[uri]             = $_GET['PrinterURI'];
$share[driver]          = $_GET['driver'];
$share[owner]           = $_GET['owner'];
$share[name]            = $_GET['printername'];

//Validate
$validation = 0; //let's consider everything okay
$share[uri]=sectuntrim($share[uri]);
      $temp = explode("/",$share[uri]);
      if ((count($temp)!=4)||$temp[3]=="") {
            $validation = 1; //ton poulo
            $validation_error[]="URI Not Valid";
      }
      if ($share[name]=="") {
            $validation = 1; //ton poulo
            $validation_error[]="Share Name Not Valid";
      }

//execute
if ($validation == 0) {
      exec('sudo /usr/bin/php-cgi -q /opt/sectun/server.php printer-
add \''.serialize($share).'\'');

//    if ($return[status]==0) print "Success"; else print "Failure";
      // TA SAMBA ERRORS MAS ERXONTAI APO TO OPTIONS.PHP
$mount_error_codes[error]
} else {
      print "<br/><strong>Validation errors:</strong><br/>";
      foreach ($validation_error as $val) print "$val<br/>";
}
      //          print "$share[name]_$share[uri]";
?>


                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Go to the user list...</a>
                  </td>
            </tr>
      </tbody>
      </table>
    <p> </p>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
```

```
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Printer-List.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // List Printers</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
```

```
            <p>List Printers</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="get" action="Printer-multi-delete-confirm.php">
<table style="width: 100%" class="ms-grid1-main">
      <!-- MSTableType="nolayout" -->
      <!-- fpstyle: 16,011111100 -->
      <tr>
            <td style="height: 23px;" class="ms-grid1-
tl"><strong>Printer Name</strong></td>
            <td style="width: 10%; height: 23px;" class="ms-grid1-
tl"><strong>User-ID</strong></td>
            <td valign="top" class="ms-grid1-t1" style="height: 23px;
width: 10%">
                  <strong>Delete</strong></td>
      </tr>

<?
      $return = unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php printer-list'));

      $counter=0;

      if ($return[count]>0) { //an exoume apotelesmata.

            foreach ($return[printers] as $value) {
                  $counter++;
                  print "<tr>";
                  print "<td style='width: 23%' class='ms-grid1-
left'>$value[printername]</td><td> </td>";
                  print "<td class='ms-grid1-even'><img alt=''
src='images-yliko/icon_delete.png' width='16' height='16' /><input
name='cb[]' type='checkbox' value='$value[printername]'/></td>";
                  print "</tr>";
            }
      }
?>
```

```
      </table>
      <input name="Submit1" type="submit" value="Delete Printers..."
/><input name="Reset1" type="reset" value="reset" />
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Printer-multi-delete-confirm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Confirm Deletion of Multiple Printers</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
```

```
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Confirm Deletion of Multiple Printers</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="post" action="Printer-multi-delete-do.php">
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 26px;"
class="style2">
                  Do you really want to delete the following
printers:
                  <strong>
                  <?
                  $cb=$_GET['cb'];
                  foreach ($cb as $printer) {
                        print "<br/>$printer<input name='cb[]'
type='hidden' value='$printer'/>";
                        }
                  ?>
                  </strong>
```

```
                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" /><input name="Submit1" type="submit"
value="Yes" checked="checked" /><input name="Button1" type="button"
value="No" onclick="history.go(-1);" />
            </tr>
      </tbody>
      </table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Printer-multi-delete-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Deletion of Multiple Printers</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
```

```
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Deletion of Multiple Printers</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" />   
<?
```

```
                  $cb=$_POST['cb'];
                  foreach ($cb as $printer) {
                        $delete = $delete."$printer ";
                  }
                  $exec = 'sudo /usr/bin/php-cgi -q
/opt/sectun/server.php printer-deletemultiple '.$delete;
//                print "<strong><pre>\n".$exec."\n</pre></strong>";

      $return=exec($exec);

?>
                  Printers deleted:<br/>
<?
                  foreach ($cb as $printer) {
                        print "<b>$printer</b><br/>\n";
                  }
?>

                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Back to user
list...</a></td>
            </tr>
      </tbody>
      </table>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Remote-share-add.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add a share</title>
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var
i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2];
val=document.getElementById(args[i]);
      if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
          if (p<1 || p==(val.length-1)) errors+='- '+nm+' must
contain an e-mail address.\n';
        } else if (test!='R') { num = parseFloat(val);
          if (isNaN(val)) errors+='- '+nm+' must contain a
number.\n';
          if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
            if (num<min || max<num) errors+='- '+nm+' must contain a
number between '+min+' and '+max+'.\n';
      } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is
required.\n'; }
    } if (errors) alert('The following error(s) occurred:\n'+errors);
    document.MM_returnValue = (errors == '');
} }
function FDK_StripChars(theFilter,theString)
{
      var strOut,i,curChar

      strOut = ""
      for (i=0;i < theString.length; i++)
      {
            curChar = theString.charAt(i)
            if (theFilter.indexOf(curChar) < 0) // if it's not in the
filter, send it thru
                  strOut += curChar
      }
      return strOut
}

function
FDK_AddToValidateArray(FormName,FormElement,Validation,SetFocus)
{
    var TheRoot=eval("document."+FormName);

    if (!TheRoot.ValidateForm)
    {
        TheRoot.ValidateForm = true;
        eval(FormName+"NameArray = new Array()")
        eval(FormName+"ValidationArray = new Array()")
        eval(FormName+"FocusArray = new Array()")
```

```
    }
    var ArrayIndex = eval(FormName+"NameArray.length");
    eval(FormName+"NameArray[ArrayIndex] = FormElement");
    eval(FormName+"ValidationArray[ArrayIndex] = Validation");
    eval(FormName+"FocusArray[ArrayIndex] = SetFocus");

}

function FDK_ValidateAlphaNum(FormElement,Required,ErrorMsg)
{
      var msg = "";
      var i, m, s, firstNonWhite
      var theString = FormElement.value;
      var msgInvalid = ErrorMsg;

      if (FDK_StripChars(" ",theString).length == 0)       {
            if (!Required)       {
          return "";
        }
            else       {
          return msgInvalid;
        }
    }
      //Strip spaces off of the sides of the string
      theString = FDK_Trim(theString);

    for (var n=0; n<theString.length; n++)     {
      theChar = theString.substring(n,n+1);
      if (!FDK_AllInRange("0","9",theChar) &&
!FDK_AllInRange("A","Z",theChar.toUpperCase()) && !(theChar == " "))
{
        return msgInvalid;
      }
    }

    return "";
}

function FDK_Trim(theString)
{
 var i,firstNonWhite

 if (FDK_StripChars(" \n\r\t",theString).length == 0 ) return ""

      i = -1
      while (1)
      {
            i++
            if (theString.charAt(i) != " ")
                  break
      }
      firstNonWhite = i
      //Count the spaces at the end
      i = theString.length
      while (1)
      {
            i--
            if (theString.charAt(i) != " ")
                  break
      }
```

```
      return theString.substring(firstNonWhite,i + 1)

}

function FDK_AllInRange(x,y,theString)
{
      var i, curChar

      for (i=0; i < theString.length; i++)
      {
            curChar = theString.charAt(i)
            if (curChar < x || curChar > y) //the char is not in
range
                  return false
      }
      return true
}

function
FDK_AddAlphaNumericValidation(FormName,FormElementName,Required,SetFo
cus,ErrorMsg)  {
  var ValString =
"FDK_ValidateAlphaNum("+FormElementName+","+Required+","+ErrorMsg+")"

FDK_AddToValidateArray(FormName,eval(FormElementName),ValString,SetFo
cus)
}
//-->
</script>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>Add a share</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
```

```
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form action="Remote-share-add-do.php" method="get" name="Shr-add-
form" id="Shr-add-form">
  <table style="width: 100%" class="ms-grid1-main">
    <!-- MSTableType="nolayout" -->
    <!-- fpstyle: 16,011111100 -->
    <tr>
      <td style="width: 30%" class="ms-grid1-tl">Share URI</td>
      <td valign="top" class="ms-grid1-top"><p>
        <input name="ShareURI" type="text" id="ShareURI" size="100"
maxlength="100" />
        <br />
        Set the URI like this:
        <br />
      //10.0.0.188/share (preferably give the intranet IP)</p>
</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-tl">UserName</td>
      <td valign="top" class="ms-grid1-top"><input name="username"
type="text" id="username" />
         (if left blank, the program assumes unprotected share -
only alphanumeric entry!)</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Password</td>
      <td class="ms-grid1-even"><input name="pass" type="password"
id="pass" />
        (only alphanumeric entry!)</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Domain / Workgroup
(if applicable)</td>
      <td class="ms-grid1-even"><input type="text" name="domain"
id="domain" />
        (if left blank, the program tries to login without the
information - only alphanumeric entry!)</td>
    </tr>
    <tr>
```

```
      <td style="width: 30%" class="ms-grid1-left">Owning user (*will
be removed for user-pages and set to their username)</td>
      <td class="ms-grid1-even"><select name="owner" id="owner">
<?
      $return = unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-list'));
      foreach ($return[users] as $value) {
            print "<option>".$value[username]."</option>\n";
      }

?>
        </select>
         </td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Share name</td>
      <td class="ms-grid1-even"><input type="text" name="sharename"
id="sharename" />
        (the share name - will be appended to the username, eg share
becomes user_share)</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Allowed user</td>
      <td class="ms-grid1-even"><select name="allowed" id="allowed">
        <?
      foreach ($return[users] as $value) {
            print "<option>".$value[username]."</option>\n";
      }

?>
      </select></td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Share
Options:</td>
      <td class="ms-grid1-even"><p>
        <input type="checkbox" name="readonly" id="readonly" />
          Read-Only / if unchecked will be read-write<br />
          <input type="checkbox" name="guest" id="guest" />
          Guest Only / Guest OK
          <br />
      </p>      </td>
    </tr>

    <tr>
      <td style="width: 30%" class="ms-grid1-left">Comment:</td>
      <td class="ms-grid1-even"><input name="comment" type="text"
id="comment" size="80" /></td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left"> </td>
      <td class="ms-grid1-even"> </td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Submit</td>
      <td class="ms-grid1-even"><input name="Submit1" type="submit"
onclick="FDK_AddAlphaNumericValidation('Shr-add-form','document.Shr-
add-form.username',false,true,'\'Please enter letters and numbers
only. Special Characters are not
allowed.\'');FDK_AddAlphaNumericValidation('Shr-add-
form','document.Shr-add-form.pass',false,true,'\'Please enter letters
```

```
and numbers only. Special Characters are not
allowed.\'');FDK_AddAlphaNumericValidation('Shr-add-
form','document.Shr-add-form.domain',false,true,'\'Please enter
letters and numbers only. Special Characters are not
allowed.\'');FDK_AddAlphaNumericValidation('Shr-add-
form','document.Shr-add-form.sharename',true,true,'\'Please enter
letters and numbers only. Special Characters are not
allowed.\'');MM_validateForm('ShareURI','','R','sharename','','R');re
turn document.MM_returnValue" value="submit" />
        <input name="Reset1" type="reset" value="reset" /></td>
    </tr>
  </table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Remote-share-add-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add a Remote Share</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
```

```
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Add a Remote Share</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
                     
<?
//receive the form variables:
```

```
$share[URI]             = $_GET['ShareURI'];
$share[user]            = $_GET['username'];
$share[pass]            = $_GET['pass'];
$share[domain]          = $_GET['domain'];
$share[owner]           = $_GET['owner'];
$share[expiry]          = $_GET['expiry'];
$share[sharename] = $_GET['sharename'];
$share[comment]         = $_GET['comment'];
$share[allowed]         = $_GET['allowed'];
$share[isreadonly]      = $_GET['readonly'];
$share[guestok]         = $_GET['guest'];

//Validate
$validation = 0; //let's consider everything okay
$share[URI]=sectuntrim($share[URI]);
      $temp = explode("/",$share[URI]);
      if ((count($temp)!=4)||$temp[3]=="") {
            $validation = 1; //ton poulo
            $validation_error[]="URI Not Valid";
      }
$share[user]=sectuntrim2($share[user]);
$share[pass]=sectuntrim2($share[pass]);
$share[domain]=sectuntrim2($share[domain]);
$share[owner]=sectuntrim2($share[owner]); //adiaforo gia twra
$share[comment]=sectuntrim3($share[comment]);
$share[sharename]=sectuntrim2($share[sharename]);
      if ($share[sharename]=="") {
            $validation = 1; //ton poulo
            $validation_error[]="Share Name Not Valid";
      }

//execute
if ($validation == 0) {
      $return=unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php remoteshare-add \''.serialize($share).'\''));
      $return[share] = $share;

      if ($return[status]==0) print "Success"; else print "Failure";
      // TA SAMBA ERRORS MAS ERXONTAI APO TO OPTIONS.PHP
$mount_error_codes[error]
} else {
      print "<br/><strong>Validation errors:</strong><br/>";
      foreach ($validation_error as $val) print "$val<br/>";
}
?>



                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Go to the user list...</a>
                  </td>
            </tr>
      </tbody>
      </table>
    <p>page notes:    </p>
    <p>* add validation
      <br />
      * do NEVER allow the ' character! </p>
```

```
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Remote-share-list.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // List Remote Shares</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
```

```
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>List Remote Shares</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<table style="width: 100%" class="ms-grid1-main">
      <!-- MSTableType="nolayout" -->
      <!-- fpstyle: 16,011111100 -->
      <tr>
            <td style="height: 23px;" class="ms-grid1-
tl"><strong>Mount Point</strong></td>
<? //       <td class="ms-grid1-tl" style="height:
23px;"><strong>Share Name</strong></td>
?>
            <td style="height: 23px;" class="ms-grid1-
tl"><strong>URI</strong></td>
      </tr>
<?
      $return=unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php remoteshare-list'));
      $counter=0;
      unset ($return[shares][count]);
      foreach ($return[shares] as $value) {
            $counter++;
            print "<tr>";
            print "<td class='ms-grid1-
left'>$value[mountpoint]</td>";
            print "<td class='ms-grid1-left'>$value[URI]</td>";
```

```
//          print "<td class='ms-grid1-even'>$counter</td>";
            print "</tr>";
      }

?>
      </table>

<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-add.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add User</title>
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var
i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2];
val=document.getElementById(args[i]);
      if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
```

```
          if (p<1 || p==(val.length-1)) errors+='- '+nm+' must
contain an e-mail address.\n';
        } else if (test!='R') { num = parseFloat(val);
          if (isNaN(val)) errors+='- '+nm+' must contain a
number.\n';
          if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
            if (num<min || max<num) errors+='- '+nm+' must contain a
number between '+min+' and '+max+'.\n';
      } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is
required.\n'; }
    } if (errors) alert('The following error(s) occurred:\n'+errors);
    document.MM_returnValue = (errors == '');
} }
//-->
</script>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>Add User</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
```

```
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form action="User-add-do.php" method="get" name="User-add-form"
id="User-add-form">
  <table style="width: 100%" class="ms-grid1-main">
    <!-- MSTableType="nolayout" -->
    <!-- fpstyle: 16,011111100 -->
    <tr>
      <td style="width: 30%" class="ms-grid1-tl">UserName</td>
      <td valign="top" class="ms-grid1-top"><input name="name"
type="text" id="name" />
         </td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Password</td>
      <td class="ms-grid1-even"><input name="pass" type="password"
id="pass" /></td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Group</td>
      <td class="ms-grid1-even"><select name="group">
          <?
      foreach ($valid_groups as $value) {
            $temp=posix_getgrgid($value);
            print "<option>".$temp[name]."</option>\n";
      }

?>
        </select>
         </td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Disable
after:</td>
      <td class="ms-grid1-even"><input name="disable" type="text"
id="disable" value="5" size="4" maxlength="3" />
      hours</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Delete after:</td>
      <td class="ms-grid1-even"><input name="delete" type="text"
id="delete" value="24" size="4" maxlength="3" />
      hours</td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left"> </td>
      <td class="ms-grid1-even"> </td>
    </tr>
    <tr>
      <td style="width: 30%" class="ms-grid1-left">Submit</td>
      <td class="ms-grid1-even"><input name="Submit1" type="submit"
onclick="MM_validateForm('name','','R','pass','','R');return
document.MM_returnValue" value="submit" />
```

```
        <input name="Reset1" type="reset" value="reset" /></td>
    </tr>
  </table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-add-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add a User</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
```

```
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Add a User</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
                     
<?
      $username = $_GET['name'];
      $password = $_GET['pass'];
      $group    = $_GET['group'];
      $delete   = $_GET['delete']; // int containing the hours after
the current time to disable the account
      $disable  = $_GET['disable']; // int containing the hours after
the current time to delete the account
      //$comment  = $_GET['comment'];
      $comment = "testing phase";
```

```
      $return=unserialize(exec("sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-add $username $password $group $disable
$delete '$comment'"));
      if (!$return[status]) {
?>
                  User <strong>&#39;<?=$return[username]
?>&#39;</strong> added
<?
      } else {
?>
                  Could not add user <strong>&#39;<?=$_GET['name']
?>&#39;</strong>
<?
      }
?>



                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Go to the user
list...</a><?= $group ?>//<?= $password ?>//<?= $username ?>
                  <pre><? print_r ($_GET); ?></pre></td>
            </tr>
      </tbody>
      </table>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-add-multiple.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add Multiple Users</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Add Multiple Users</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
```

```
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="post" action="User-add-multiple-do.php">
<table style="width: 100%" class="ms-grid1-main">
      <!-- MSTableType="nolayout" -->
      <!-- fpstyle: 16,011111100 -->
      <tr>
            <td style="width: 30%" class="style4">Username
Pattern</td>
            <td valign="top" class="style4">
                  st_<input name="pattern" type="text" size="6"
maxlength="6" />_xxx</td>
      </tr>
      <tr>
            <td style="width: 30%" class="style4">Group</td>
            <td class="style4"><select name="group">
<?
      foreach ($valid_groups as $value) {
            $temp=posix_getgrgid($value);
            print "<option>".$temp[name]."</option>\n";
      }

?>
            </select> </td>
      </tr>

      <tr>
            <td style="width: 30%" class="style4">user count</td>
            <td class="style4">
            <input name="count" type="text" size="3" maxlength="3" />
(Range:
            1..999)</td>
      </tr>
      <tr>
      <td style="width: 30%" class="ms-grid1-left">Disable
after:</td>
        <td class="ms-grid1-even"><input name="disable" type="text"
id="disable" value="5" size="4" maxlength="3" />
          hours</td>
    </tr>
      <tr>
      <td style="width: 30%" class="ms-grid1-left">Delete after:</td>
        <td class="ms-grid1-even"><input name="delete" type="text"
id="delete" value="24" size="4" maxlength="3" />
          hours</td>
    </tr>

      <tr>
            <td style="width: 30%" class="style4"> </td>
            <td class="style4"> </td>
      </tr>
      <tr>
            <td style="width: 30%" class="style4">Submit</td>
            <td class="style4">
            <input name="Submit1" type="submit" value="submit"
/><input name="Reset1" type="reset" value="reset" /></td>
      </tr>
```

```
</table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-add-multiple-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Add Multiple Users</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
```

```
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Add Multiple Users (Via Pattern)</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
                     
<?
      $pattern = 'st_'.$_POST['pattern'];
      $group   = $_POST['group'];
      $count   = $_POST['count'];
      $disable = $_POST['disable'];
      $delete  = $_POST['delete'];

      $return=unserialize(exec("sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-addmultiple $pattern $group $count
$disable $delete"));
?>                Users added:
<table cellpadding="2" cellspacing="0" border="1" align="center">
```

```
<tr><th> </th><th>name</th><th>password</th><th>output</th></tr>
<?
$counter = 1;
foreach ($return[users] as $one) {
      print
"<tr><td>$counter</td><td>$one[username]</td><td>$one[password]</td><
td>";
      print_r ($one[output]);
      print "</td></tr>\n";
      $counter++;
}
?>
</table>
                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Go to the user
list...</a></td>
            </tr>
      </tbody>
      </table>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-delete-confirm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>
```

```
<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Confirm Deletion of a User</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
      <p>Confirm Deletion of a User</p>
      <!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="get" action="User-delete-do.php">
```

```
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 26px;"
class="style2">
                  Do you really want to delete user <strong><? print
$_GET['name']; ?></strong>?</td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" /><input name="Submit1" type="submit"
value="Yes" checked="checked" /><input name="Button1" type="button"
value="No" onclick="history.go(-1);" />
                  <input name="name" type="hidden"
value="<?=$_GET['name'] ?>" /></td>
            </tr>
      </tbody>
      </table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-delete-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>
```

```
<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Deletion of a User</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Deletion of a User</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
```

```
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" />   
<?
      $return=unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-delete '.$_GET['name']));
      if (!$return[status]) {
?>
                  User <strong>&#39;<?=$return[username]
?>&#39;</strong> deleted
<?
      } else {
?>
                  Could not delete user <strong>&#39;<?=$_GET['name']
?>&#39;</strong>
<?
      }
?>
                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Back to user
list...</a></td>
            </tr>
      </tbody>
      </table>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-list.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // List Users</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>List Users</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
```

```
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="get" action="User-multi-delete-confirm.php">
<table style="width: 100%" class="ms-grid1-main">
      <!-- MSTableType="nolayout" -->
      <!-- fpstyle: 16,011111100 -->
      <tr>
            <td style="height: 23px;" class="ms-grid1-
tl"><strong>Username</strong></td>
<? /*
            <td style="width: 12%; height: 23px;" class="ms-grid1-
tl"><strong>Group-ID</strong></td>
            <td style="width: 36%; height: 23px;" class="ms-grid1-
tl"><strong>Group</strong></td> */
?>
            <td style="width: 10%; height: 23px;" class="ms-grid1-
tl"><strong>User-ID</strong></td>
            <td valign="top" class="ms-grid1-t1" style="height: 23px;
width: 20%">
                  <strong>Status</strong></td>
            <td valign="top" class="ms-grid1-t1" style="height: 23px;
width: 10%">
                  <strong>Delete</strong></td>
      </tr>

<?
      $return = unserialize(exec('sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-list'));

      $counter=0;

      if ($return[usercount]>0) { //an exoume apotelesmata.

            foreach ($return[users] as $value) {
                  $counter++;
                  print "<tr>";
                  print "<td style='width: 23%' class='ms-grid1-
left'>$value[username]</td>";
//                print "<td style='width: 12%' class='ms-grid1-
left'>$value[groupid]</td>";
//                print "<td style='width: 36%' class='ms-grid1-
left'>$value[group]</td>";
                  print "<td style='width: 10%' class='ms-grid1-
left'>$value[userid]</td>";
                  $tempval = str_split($value[flags]);
                  if ($tempval[1]== 1) $userstatus = "<font
size=2>Disabled</font>";
                  else $userstatus = "<strong>Active</strong>";
                  print "<td class='ms-grid1-even' style='width:
401px'>$userstatus</td>";
                  print "<td class='ms-grid1-even'><a href='User-
delete-confirm.php?name=$value[username]'><img alt='' src='images-
yliko/icon_delete.png' width='16' height='16' /></a><input
name='cb[]' type='checkbox' value='$value[username]'/></td>";
```

```
                print "</tr>";
            }
        }
?>
        </table>
        <input name="Submit1" type="submit" value="Delete Multiple
Users" /><input name="Reset1" type="reset" value="reset" />
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-multi-delete-confirm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Confirm Deletion of Multiple Users</title>
<!-- InstanceEndEditable -->
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
```

```
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Confirm Deletion of Multiple Users</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
<form method="post" action="User-multi-delete-do.php">
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 26px;"
class="style2">
                  Do you really want to delete the following users:
                  <strong>
                  <?
                  $cb=$_GET['cb'];
                  foreach ($cb as $user) {
                        print "<br/>$user<input name='cb[]'
type='hidden' value='$user'/>";
```

```
                        }
                  ?>
                  </strong>

                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" /><input name="Submit1" type="submit"
value="Yes" checked="checked" /><input name="Button1" type="button"
value="No" onclick="history.go(-1);" />
            </tr>
      </tbody>
      </table>
</form>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

User-multi-delete-do.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/template.dwt.php"
codeOutsideHTMLIsLocked="false" -->
<head>

<?php require_once("/var/www/options/options.php"); ?>

<meta http-equiv="Cache-Control" content="No-Cache">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>SecTun // Deletion of Multiple Users</title>
<!-- InstanceEndEditable -->
```

```
<link href="sectun.css" rel="stylesheet" type="text/css" />
</head>
<body style="background-color: #8D8B45; border-top-width: 0px">
<table style="width: 100%; background-color: white">
<tr>
<td> Session Data Placeholder
</td>
</tr>
</table>

<table style="width: 500px; height: 79px" align="center"
class="style1">
  <tr style="background-color:black">
    <td style="background-color:black" colspan="4"><img alt="Banner"
src="banner.png" width="500" height="135" /></td>
  </tr>
  <tr>
    <td colspan="4" ><h1>v0.5 // Web Administration
Interface.</h1></td>
  </tr>
  <tr>
    <td colspan="4"><h2>Andreas Skoufis / Sotiris
Stamokostas</h2></td>
  </tr>
  <tr>
    <td class="title" colspan="4"><!-- InstanceBeginEditable
name="Title" -->
            <p>Deletion of Multiple Users</p>
<!-- InstanceEndEditable --> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="Printer-List.php">My
Printers</a></td>
    <td class="menuitems"><a href="Printer-add.php">Add
Printer</a></td>
    <td class="menuitems"><a href="Remote-share-list.php">My
Shares</a></td>
    <td class="menuitems"><a href="Remote-share-add.php">Add
Share</a></td>
  </tr>
  <tr class="menuitems">
    <td colspan="4" class="menuitems"> </td>
  </tr>
  <tr class="menuitems">
    <td class="menuitems"><a href="User-list.php">Users</a></td>
    <td class="menuitems"><a href="User-add.php">Add a User</a></td>
    <td class="menuitems"><a href="User-add-multiple.php">Add
Multiple Users</a></td>
    <td class="menuitems"><a href="Debug-
Cronscript.php">Maintain</a></td>
  </tr>
</table>
<br/>
<!-- InstanceBeginEditable name="PageData" -->
      <table style="width: 40%" class="ms-grid1-main" align="center">
            <!-- MSTableType="nolayout" -->
            <!-- fpstyle: 16,011111100 -->
            <tbody>
            <tr>
                  <td style="width: 24%; height: 21px;"
class="style2">
```

```
                  <img alt="" src="images-yliko/icon_delete.png"
width="32" height="32" />   
<?
                  $cb=$_POST['cb'];
                  foreach ($cb as $user) {
                        $delete = $delete."$user ";
                  }
                  $exec = 'sudo /usr/bin/php-cgi -q
/opt/sectun/server.php user-deletemultiple '.$delete;
//                print "<strong><pre>\n".$exec."\n</pre></strong>";

      $return=exec($exec);

?>
                  Users deleted:<br/>
<?
                  foreach ($cb as $user) {
                        print "<b>$user</b><br/>\n";
                  }
?>

                  </td>
            </tr>
            <tr>
                  <td style="width: 24%" class="style3">
                  <a href="User-list.php">Back to user
list...</a></td>
            </tr>
      </tbody>
      </table>
<!-- InstanceEndEditable -->
<? if ($debug) { ?>
<p>Version history:</p>
<p>* 0.5   ==&gt; Final Cleanups, adding printer.<br />
  * 0.21a ==&gt; completing cron script functions...<br />
  * 0.20b ==&gt; User management in the db. // add user enters info
in the db<br />
  * 0.19b ==&gt; Started connecting the database to the application.
// initial db support in server.php<br />
  * 0.18a ==&gt; Remote Share Add now kind of fully works!
whoohoooo<br />
  * 0.17c ==&gt; Started working on share.php cleanup (c: renamed
every command for server.php)<br />
  * 0.16b ==&gt; Working on the &quot;add remote share&quot; function
// List Active Remote Shares Partially working<br />
  * 0.15a ==&gt; Moved to Dreamweaver (better PHP+Database
handling)<br />
  * 0.11 ==&gt; User management works! (error messages suck
though)<br />
* 0.10 ==&gt; Initial. Working on Microsoft Expression Web</p>

return dump:
<pre>
<? print_r ($return) ?>
</pre>
return: <? print $return ?>
<? } ?>
</body>
<!-- InstanceEnd --></html>
```

Options.php

```
<?
      function mysqltime ($timestamp) { //input: php timestamp.
output: mysql timestamp
            return date('YmdHis', $timestamp);
      };
      function phptime ($mysqltimestamp) { //input: php timestamp.
output: mysql timestamp
      }; //pipes akoma

      function sectuntrim($string) { //sbinei tous axristous
xaraktires
            return str_replace(array("
","\\","\n","\r","'","\t","\0","\x0B",'"'),"",$string);
      };
      function sectuntrim2($string) { //sbinei tous axristous
xaraktires kai to /
            return str_replace(array("
","\\","\n","\r","'","\t","\0","\x0B","/",'"'),"",$string);
      };
      function sectuntrim3($string) { //sbinei tous axristous
xaraktires alla afinei ta kena
            return
str_replace(array("\\","\n","\r","'","\t","\0","\x0B",'"'),"",$string
);
      };

      $mount_error_codes[0]   = "success";
      $mount_error_codes[1]   = "incorrect invocation or
permissions";
      $mount_error_codes[2]   = "system error (out of memory, cannot
fork, no more loop devices)";
      $mount_error_codes[4]   = "internal mount bug";
      $mount_error_codes[8]   = "user interrupt";
      $mount_error_codes[16]  = "problems writing or locking
/etc/mtab";
      $mount_error_codes[32]  = "mount failure";
      $mount_error_codes[64]  = "some mount succeeded";


      $valid_groups = array (60000);  // valid groups
      $debug = 0; // 1 = ON, 0 = OFF

?>
```

Sectun_DB.php

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_Sectun_DB = "10.0.0.50";
$database_Sectun_DB = "sectun";
$username_Sectun_DB = "user";
$password_Sectun_DB = "user";
$Sectun_DB = mysql_pconnect($hostname_Sectun_DB, $username_Sectun_DB,
$password_Sectun_DB) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

root_includes.php

```
<?
require_once ("/var/www/options/options.php");   //just in case
require_once ("/opt/sectun/root_options.php");
```

```
function dbinsert($query) {
	global $prefs;
	try {
		$dbh = new PDO($prefs[db][dsn], $prefs[db][username],
$prefs[db][password]);
		$dbh->query($query);
		$return[dbErrorInfo] = $dbh->errorInfo();
		if (count($return[dbErrorInfo])==1) $return[status]=0;
		else {
			$return[status]=4;
			$return[errormessage]="Query Problem:
".$return[dbErrorInfo][2];
		}
		$dbh = null;
	} catch (PDOException $e) {
		$return[status]=2;
		$return[errormessage] = "DB Problem: " . $e-
>getMessage();
	}
	return $return;
}

function dbselect($query) {
	global $prefs;
	try {
		$dbh = new PDO($prefs[db][dsn], $prefs[db][username],
$prefs[db][password]);
		$a = $dbh->query($query);
		$return[dbErrorInfo] = $dbh->errorInfo();
		if (count($return[dbErrorInfo])==1) {
			$return[status]=0;
			$return[results] = $a->fetchAll();
		}
		else {
			$return[status]=2; //2: query problem
			$return[errormessage]="Query Problem:
".$return[dbErrorInfo][2];
		}
		$dbh = null;
	} catch (PDOException $e) {
		$return[status]=1; //1: db problem
		$return[errormessage] = "DB Problem: " . $e-
>getMessage();
	}
	return $return;
}

// i idia me tin dbinsert
function dbdelete($query) {
	return dbinsert($query);
}
function dbupdate($query) {
	return dbinsert($query);
}


	//tou kwlou!
function writelog ($data) {
	global $prefs;
	$fh = fopen($prefs[log][file], 'a') or die("can't open file");
```

```
	$log = print_r($data,TRUE);
	fwrite($fh, date("F j, Y, g:i a")."\n");
	fwrite($fh, "Command: ".$data[command]."\n");
	fwrite($fh, $log."\n");
	fclose($fh);
}	//tou kwlou!
function writecronlog ($data) {
	global $prefs;
	$fh = fopen($prefs[log][cronlogfile], 'a') or die("can't open
file");
	$log = print_r($data,TRUE);
	fwrite($fh, date("F j, Y, g:i a")."\n");
	fwrite($fh, $log."\n");
	fclose($fh);
}

	//etoimi?
function getuserlist() {
		$getpassfile = exec ('perl -T
/opt/sectun/getpassfile.pl');
		//pairnoume tous xristes 1-1:
		$temp_users = explode('||',$getpassfile);
		array_pop($temp_users); //afairoume ton teleftaio keno

		foreach ($temp_users as $key => $value) {
			$temp_users2[$key]=explode('//',$value);
			/* return type:
			Username = $users2[][1]
			Group-ID = $users2[][2]
			User-ID  = $users2[][3]
			Flags    = $users2[][0]
			*/
		}

		foreach ($temp_users2 as $key => $value) {
	if ($value[2]>=60000) {
	  //metatropi se pio anagnwsimi morfi
	  $return[users][$key][username] = $value[1];
	  $return[users][$key][groupid] = $value[2];
	  $return[users][$key][userid] = $value[3];
	  $return[users][$key][flags] = $value[0];
	  $temp = posix_getgrgid($value[2]);
	  $return[users][$key][group] = $temp[name]; //lambanoume to
onoma tou group
	}
		}
		$return[usercount] = count ($return[users]); //to plithos
twn xristwn.
		return $return;
		/*
		 * morfi epistrofis:
		 * $return[users][0][username]
		 * $return[users][0][groupid]
		 * $return[users][0][userid]
		 * $return[users][0][flags]
		 * $return[users][0][group]
		 * kok gia xristi 1,2,3....
		 */
}

	//doulevei?
```

```
function userexists($username) {
      // returns 1 if user exists, 0 otherwise
      $userlist = getuserlist();
      foreach ($userlist[users] as $test) {
            if ($username==$test[username]) return 1;
      }
      return;
}

      //doulevei --> kalytero $return.
function getremotesharelist($username = "",$URI = "") {

      if ($username=="") $userparm = "";
      else $userparm = " | grep /$username/";

      if ($URI!="") $userparm.=" |grep $URI";

      exec("mount |grep //$userparm", $output, $exitcode);
      if ($exitcode==0) {
            foreach ($output as $key => $value) {
                  $temp = explode (" ", $value);
                  $shares[$key][mountpoint] = $temp[2];
                  $shares[$key][URI] = $temp[0];
            }
            $shares[count]=count($shares);
      }
      else $shares[count]=0;
      return $shares;
      /*
       * return example:
    [0] => Array
            [mountpoint] => /opt/sectun/share/st_ntua_003
            [URI] => //10.0.0.188/share
    [1] => Array
            [mountpoint] => /opt/sectun/share/st_ntua_004
            [URI] => //10.0.0.188/share
      */
}

function getprinterlist() {

      exec("lpstat -p", $output, $exitcode);
      if ($exitcode==0) {
            foreach ($output as $key => $value) {
                  $temp = explode (" ", $value);
                  $printers[printers][$key][printername] = $temp[1];
            }
            $printers[count]=count($printers[printers]);
      }
      else $printers[count]=0;
      return $printers;
      /*
       * return example:
    [0] => Array
            [printername] => hp_deskjet_690c
    [1] => Array
            [printername] => canon_pixma_ip20
      */
}
```

```
      //gia ton poutso, tha doulepsei me mysql.
function remoteshareexists($user,$URI) {

      // returns 1 if remote exists, 0 otherwise
      $sharelist = getremotesharelist($user,$URI);
      if ($sharelist[count]==0) return 0;
      return 1;
}

function getshares($share = "") {
      $command = "/opt/sectun/list_shares.pl |grep _";
      if ($share!="") $command .= " | grep $share";
      exec($command, $output,$rval);
      foreach ($output as $key => $val) {
            $output2[$key] = exec ("/opt/sectun/get_share.pl $val");
            $output3[$key] = explode ('||',$output2[$key]);
            array_pop ($output3[$key]);
            foreach ($output3[$key] as $key2 => $val2) {
                  $output3[$key][$key2] = explode("\\\\", $val2);
            }
      }
      foreach ($output3 as $key => $val) {
            foreach ($val as $key2 => $val2) {
                  if ($key2!=0) $return[$val[0][1]][$val2[0]] =
$val2[1];
            }
      }
      return $return;
}

function getmounts() {
      exec ("mount | grep /opt/sectun/share", $output, $rval);
      if ($rval==0) {
            foreach ($output as $key => $val) {
                  $temp = explode (" ",$val);
                  $temp2 = explode ("/",$temp[2]);
                  $user = $temp2[4];
                  $share = $temp2[5];
                  $sharename = $user."_".$share;
                  $return[$sharename][user]=$user;
                  $return[$sharename][sharename]=$share;
                  $return[$sharename][URI]=$temp[0];
                  $return[$sharename][fullsharename]=$sharename;
                  $return[$sharename][mountpoint] = $temp[2];
            }
      }
      return $return;
}
?>
```

root_options.php

```
<?

      // Database options
      $prefs[db][dsn]         = "mysql:host=localhost;dbname=sectun";
      $prefs[db][username]    = "user";
      $prefs[db][password]    = "user";

      // Log options
      $prefs[log][file]             = "/opt/sectun/logs/log.txt";
      $prefs[log][cronlogfile]= "/opt/sectun/logs/cronlog.txt";
```

```
        // Base group. Everyone is a member of this group
        $prefs[basegroupid]           = 60000;
        $prefs[basegroup]       = "tunnel";

?>
```

getdriverlist.sh

```
#!/bin/bash
rm ./printdrivers2.txt
rm ./printdrivers.txt
foomatic-ppdfile -A | grep -o -E "\=.+'[ ]Dr" | grep -o -E \'.+\' |
awk '{print substr($0,2,length($0)-2)}' >./printdrivers2.txt
sort ./printdrivers2.txt -o ./printdrivers.txt
rm ./printdrivers2.txt
```

addsmb.pl

```
#!/usr/bin/perl

# Adds user to /etc/passwd and smbpasswd

use strict;
use warnings;
use File::Copy;
use Crypt::SmbHash '0.12';
use File::Temp;
use Fcntl qw(:DEFAULT :flock);
use Passwd::Linux qw( setpwinfo );
use FAUS::Helper qw( :DEFAULT shadow_enc maxuid );
use Getopt::Std;
use sigtrap qw( handler abort normal-signals );
use constant PASSWD_FILE => '/etc/passwd';
use constant VERSION => 1.3;
use constant FAKE_SHELL => '/bin/false';

clean_env();

my $temp_file = tmpnam();
# smbpasswd file handler reference
my $fh_smbfile;

# putting everything in a array to pass as a reference to functions
# sequence is: file, u, password, group, username
my %opts;
getopts( 'u:p:g:n:', \%opts );

validate();
add_passwd();
add_smb();

################################
# functions
################################

#checking data for values and bad characters and untaint them
sub validate {

    my $key;
    my @obligatory_keys = qw ( u p g );

    help_message() unless( %opts );
```

```
    foreach $key( @obligatory_keys ) {

      if ( exists( $opts{$key} ) ) {

          help_message() unless ( exists( $opts{$key} ) );

      } else {

          help_message();

      }

    }

    error( 'Invalid characters in Samba password file pathname' )
unless ( SMBPASSWD =~ /^[\w\/]+smbpasswd$/ );

    $opts{u} =~ /^(\w+)$/ ? ( $opts{u} = $1 ) : error( 'Invalid data
as parameters in -u argument' );

    $opts{g}  =~ /^(\w+)$/ ? ( $opts{g} = $1 ) : error( 'Invalid data
as parameters in -g argument' );

    $opts{p} =~ /^([\w\!\?\#\@\&\%\*]+)$/ ? ( $opts{p} = $1 ) :
error( 'Invalid characters in -p argument' );

# getting GID
    $opts{g} = getgrnam( $opts{g} );

    if ( exists( $opts{n} ) and ( $opts{n} ne '' ) ) {

      chomp( $opts{n} );
      $opts{n} =~/^([\w\s]+)$/ ? ( $opts{n} = $1 ) : error( 'Invalid
data as parameters in -n argument' );

    } else {

      $opts{n} = 'Samba user';

    }

    if ( $opts{u} eq 'root') {

      error( 'I will not deal with system users!' );

    }
# end of validate function
}


# Adding to /etc/passwd
sub add_passwd {

    my $new_uid = maxuid() + 1;
    my $home = '/home/' . $opts{u};
    $new_uid = MINIMUM_UID + 1 unless ( $new_uid > MINIMUM_UID );
    my $return;

# if there is an username
    if ( defined( $opts{n} ) ) {
```

```
      $return = setpwinfo( $opts{u},
                           shadow_enc( $opts{p} ),
                           $new_uid,
                           $opts{g},
                           $opts{n},
                           $home,
                           FAKE_SHELL );

    } else {

      $return = setpwinfo( $opts{u},
                           shadow_enc( $opts{p} ),
                           $new_uid,
                           $opts{g},
                           $opts{u},
                           $home,
                           FAKE_SHELL );

    }

    if ( $return < 0 ) {

      error( "Cannot add $opts{u} in " . PASSWD_FILE . ": $!" );

    }

    if ( $return == 1 ) {

      error( 'Operation attemped on uid 0' );

    }

    # creates user home directory
    set_home( $opts{u}, $home, $opts{g} );

}

# Adding to smbpasswd

sub add_smb {

    if ( -e SMBPASSWD ) {

      copy( SMBPASSWD, $temp_file ) or error( "Backup copy failed:
$!" );

    }

    sysopen( FILE, SMBPASSWD, O_RDWR | O_CREAT ) or error( "Cannot
access smbpasswd: $!" );

    $fh_smbfile = \*FILE;

    flock( FILE, LOCK_EX ) or error( "Cannot lock file smbpasswd:
$!", $fh_smbfile );

    my ( $login, undef, $uid ) = getpwnam( $opts{u} );
```

```
    error( 'Impossible to retrieve information from '. PASSWD_FILE .
" of $opts{u}", $fh_smbfile, $opts{file}, $temp_file ) unless ( (
defined( $login ) ) and ( defined( $uid ) ) );

# check if the user already exists in the file
    my $user_exists = 0;
    my $search = quotemeta( $opts{u} );

    while (<FILE>) {

      next unless /\w+/;

      if ( /^$search/o ) {

          $user_exists = 1;
          last;

      }

    }

    unless ( $user_exists ) {

      my ( $lm, $nt );
      ntlmgen( $opts{p}, $lm, $nt );
      printf FILE "%s:%d:%s:%s:[%-11s]:LCT-%08X:\n", $login, $uid,
$lm, $nt, 'UX', time;
      close( FILE );
      unlink( $temp_file ) or error ( "Failed to remove backup file
$temp_file: $!" );

    } else {

      error( "The user $opts{u} already exists in the smbpasswd file"
);

    }
}


# dies, but before tries to close the reference file
sub error {

    my $message = shift;

    close( $fh_smbfile ) if ( defined( $fh_smbfile ) );

    if ( -e $temp_file ) {

      copy( $temp_file, SMBPASSWD) or warn 'Failed to restore '.
SMBPASSWD . "backup file: $!\n";

      unlink( $temp_file ) or warn "Failed to remove backup file
$temp_file: $!\n";
    } else {

      warn "Backup file does not exist\n";

    }

    die get_label() . $message . "\n";
```

```
}

sub abort {

    my $signame = shift;
    error (" Aborted due an received SIG$signame signal." );

}

sub set_home {

    my $user = shift;
    my $home_dir = shift;
    my $gid = shift;

    my $uid = getpwnam( $user );

    error( "Cannot create $home_dir home directory: $!" ) unless(
mkdir( $home_dir ) );

    error( "Error when changing permissions of $home_dir" ) unless(
chmod DIR_MASK, $home_dir );

    error( "Error when changing default owner for $home_dir" )
unless( chown $uid, $gid, $home_dir );

}

sub help_message {

    my $version = \VERSION;
    my $regards_to = \REGARDS_TO;
    my $message = <<BLOCK;
addsmb.pl - version $$version
addsmb.pl $$regards_to
Usage: addsmb.pl -u<userid> -p<password> -g<group> [-n<username>]
       -n use is optional

BLOCK

    die $message;

}
```

create_share.pl

```
#!/usr/bin/perl

# example
# sudo perl create_share.pl share_name sotiris home path /mnt/samba
available yes "valid users" "sotiris mitros"

%share=@ARGV;
&create_share($share{'share_name'});

# create_share(name)
# Add an entry to the config file
sub create_share
{
&open_tempfile(CONF, ">> /etc/samba/smb.conf");
&print_tempfile(CONF, "\n");
```

```
&print_tempfile(CONF, "[$_[0]]\n");
foreach $k (grep {!/share_name/} (keys %share)) {
      &print_tempfile(CONF, "\t$k = $share{$k}\n");
      }
&close_tempfile(CONF);
}

# open_tempfile([handle], file, [no-error], [no-tempfile], [safe?])
# Returns a temporary file for writing to some actual file
sub open_tempfile
{

      # Actually opening
      local ($fh, $file, $noerror, $notemp, $safe) = @_;
      local %gaccess = &get_module_acl(undef, "");
      if ($file =~ /\r|\n|\0/) {
            if ($noerror) { return 0; }
            else { &error("Filename contains invalid characters"); }
            }
      if (&is_readonly_mode() && $file =~ />/ && !$safe) {
            print "Read-only mode .. veto all writes";
            # Read-only mode .. veto all writes
            print STDERR "vetoing write to $file\n";
            return open($fh, ">$null_file");
            }
      elsif ($file =~ /^(>|>>)\/dev\// || lc($file) eq "nul") {
            print "Writes to /dev/null or TTYs don't need to be
handled";
            # Writes to /dev/null or TTYs don't need to be handled
            return open($fh, $file);
            }
      elsif ($file =~ /^>\s*(([a-zA-Z]:)?\/.*)$/ && !$notemp) {
            print "Over-writing a file, via a temp file";
            # Over-writing a file, via a temp file
            $file = $1;
            $file = &translate_filename($file);
            while(-l $file) {
                  # Open the link target instead
                  $file = &resolve_links($file);
                  }
            if (-d $file) {
                  # Cannot open a directory!
                  if ($noerror) { return 0; }
                  else { &error("Cannot write to directory"); }
                  }
            local $tmp = &open_tempfile($file);
            local $ex = open($fh, ">$tmp");
            if (!$ex && $! =~ /permission/i) {
                  # Could not open temp file .. try opening actual
file
                  # instead directly
                  $ex = open($fh, ">$file");
                  delete($main::open_tempfiles{$file});
                  }
            else {
                  $main::open_temphandles{$fh} = $file;
                  }
            binmode($fh);
            if (!$ex && !$noerror) {
                  &error(&text("efileopen", $file, $!));
                  }
```

```
            return $ex;
            }
      elsif ($file =~ /^>\s*(([a-zA-Z]:)?\/.*)$/ && $notemp) {
            # Just writing direct to a file
            print "Just writing direct to a file";
            $file = $1;
            $file = &translate_filename($file);
            local $ex = open($fh, ">$file");
            $main::open_temphandles{$fh} = $file;
            if (!$ex && !$noerror) {
                  &error(&text("efileopen", $file, $!));
                  }
            binmode($fh);
            return $ex;
            }
      elsif ($file =~ /^>>\s*(([a-zA-Z]:)?\/.*)$/) {
            # Appending to a file .. nothing special to do
            $file = $1;
            $file = &translate_filename($file);
            local $ex = open($fh, ">>$file");
            $main::open_temphandles{$fh} = $file;
            if (!$ex && !$noerror) {
                  &error(&text("efileopen", $file, $!));
                  }
            binmode($fh);
            return $ex;
            }
      elsif ($file =~ /^([a-zA-Z]:)?\//) {
            print $file;
            print "\n";
            print "Read mode .. nothing to do here";
            # Read mode .. nothing to do here
            $file = &translate_filename($file);
            print $file;
            return open($fh, $file);
            }
      elsif ($file eq ">" || $file eq ">>") {
            local ($package, $filename, $line) = caller;
            if ($noerror) { return 0; }
            else { &error("Missing file to open at
${package}::${filename} line $line"); }
            }
      else {
            print "XXX";
            # XXX append / update support?
            local ($package, $filename, $line) = caller;
            &error("Unsupported file or mode $file at
${package}::${filename} line $line");
            }
}

# close_tempfile(file|handle)
# Copies a temp file to the actual file, assuming that all writes
were
# successful.
sub close_tempfile
{
local $file;
if (defined($file = $main::open_temphandles{$_[0]})) {
      # Closing a handle
      close($_[0]) || &error(&text("efileclose", $file, $!));
```

```
	delete($main::open_temphandles{$_[0]});
	return &close_tempfile($file);
	}
elsif (defined($main::open_tempfiles{$_[0]})) {
	# Closing a file
	local @st = stat($_[0]);
	if ($gconfig{'os_type'} =~ /-linux$/ && &has_command("chcon"))
{
		# Set original security context
		system("chcon --reference=".quotemeta($_[0]).
		       " ".quotemeta($main::open_tempfiles{$_[0]}).
		       " >/dev/null 2>&1");
		}
	rename($main::open_tempfiles{$_[0]}, $_[0]) || &error("Failed
to replace $_[0] with $main::open_tempfiles{$_[0]} : $!");
	if (@st) {
		# Set original permissions and ownership
		chmod($st[2], $_[0]);
		chown($st[4], $st[5], $_[0]);
		}
	delete($main::open_tempfiles{$_[0]});
	@main::temporary_files = grep { $_ ne
$main::open_tempfiles{$_[0]} } @main::temporary_files;
	if ($main::open_templocks{$_[0]}) {
		&unlock_file($_[0]);
		delete($main::open_templocks{$_[0]});
		}
	return 1;
	}
else {
	# Must be closing a handle not associated with a file
	close($_[0]);
	return 1;
	}
}

# print_tempfile(handle, text, ...)
# Like the normal print function, but calls &error on failure
sub print_tempfile
{
local ($fh, @args) = @_;
#(print $fh @args) || &error(&text("efilewrite",
#                    $main::open_temphandles{$fh} || $fh, $!));

print $fh @args;
}


# get_module_acl([user], [module], [no-rbac], [no-default])
# Returns a hash  containing access control options for the given
user
sub get_module_acl
{
local %rv;
local $u = defined($_[0]) ? $_[0] : $base_remote_user;
local $m = defined($_[1]) ? $_[1] : $module_name;
local $mdir = &module_root_directory($m);
if (!$_[3]) {
	&read_file_cached("$mdir/defaultacl", \%rv);
	}
local %usersacl;
```

```
if (!$_[2] && &supports_rbac($m) && &use_rbac_module_acl($u, $m)) {
	# RBAC overrides exist for this user in this module
	local $rbac = &get_rbac_module_acl(
			defined($_[0]) ? $_[0] : $remote_user, $m);
	local $r;
	foreach $r (keys %$rbac) {
		$rv{$r} = $rbac->{$r};
		}
	}
elsif ($gconfig{"risk_$u"} && $m) {
	# ACL is defined by user's risk level
	local $rf = $gconfig{"risk_$u"}.'.risk';
	&read_file_cached("$mdir/$rf", \%rv);

	local $sf = $gconfig{"skill_$u"}.'.skill';
	&read_file_cached("$mdir/$sf", \%rv);
	}
else {
	# Use normal Webmin ACL
	&read_file_cached("$config_directory/$m/$u.acl", \%rv);
	if ($remote_user ne $base_remote_user && !defined($_[0])) {

	&read_file_cached("$config_directory/$m/$remote_user.acl",\%rv)
;
		}
	}
if ($tconfig{'preload_functions'}) {
	&load_theme_library();
	}
if (defined(&theme_get_module_acl)) {
	%rv = &theme_get_module_acl($u, $m, \%rv);
	}
return %rv;
}


# module_root_directory(module)
# Given a module name, returns its root directory
sub module_root_directory
{
local $d = ref($_[0]) ? $_[0]->{'dir'} : $_[0];
if (@root_directories > 1) {
	local $r;
	foreach $r (@root_directories) {
		if (-d "$r/$d") {
			return "$r/$d";
			}
		}
	}
return "$root_directories[0]/$d";
}


# read_file_cached(file, &assoc)
# Like read_file, but reads from a cache if the file has already been
read
sub read_file_cached
{
local $realfile = &translate_filename($_[0]);
if (defined($main::read_file_cache{$realfile})) {
	%{$_[1]} = ( %{$_[1]}, %{$main::read_file_cache{$realfile}} );
```

```
	}
else {
	local %d;
	&read_file($_[0], \%d, $_[2], $_[3], $_[4]);
	%{$main::read_file_cache{$realfile}} = %d;
	%{$_[1]} = ( %{$_[1]}, %d );
	}
}


# translate_filename(filename)
# Applies all relevant registered translation functions to a filename
sub translate_filename
{
local $realfile = $_[0];
local @funcs = grep { $_->[0] eq $module_name ||
		      !defined($_->[0]) } @main::filename_callbacks;
local $f;
foreach $f (@funcs) {
	local $func = $f->[1];
	$realfile = &$func($realfile, @{$f->[2]});
	}
return $realfile;
}


# read_file(file, &assoc, [&order], [lowercase], [split-char])
# Fill an associative array with name=value pairs from a file
sub read_file
{
local $_;
local $split = defined($_[4]) ? $_[4] : "=";
local $realfile = &translate_filename($_[0]);
&open_readfile(ARFILE, $_[0]) || return 0;
while(<ARFILE>) {
	chomp;
	local $hash = index($_, "#");
	local $eq = index($_, $split);
	if ($hash != 0 && $eq >= 0) {
		local $n = substr($_, 0, $eq);
		local $v = substr($_, $eq+1);
		chomp($v);
		$_[1]->{$_[3] ? lc($n) : $n} = $v;
		push(@{$_[2]}, $n) if ($_[2]);
		}
	  }
close(ARFILE);
if (defined($main::read_file_cache{$realfile})) {
	%{$main::read_file_cache{$realfile}} = %{$_[1]};
	}
return 1;
}


# open_readfile(handle, file)
# Opens some file for reading. Returns 1 on success, 0 on failure
sub open_readfile
{
local ($fh, $file) = @_;
local $realfile = &translate_filename($file);
return open($fh, "<".$realfile);
```

```
}


# supports_rbac([module])
# Returns 1 if RBAC client support is available
sub supports_rbac
{
return 0 if ($gconfig{'os_type'} ne 'solaris');
eval "use Authen::SolarisRBAC";
return 0 if ($@);
if ($_[0]) {
      #return 0 if (!-r &module_root_directory($_[0])."/rbac-
mapping");
      }
return 1;
}


# is_readonly_mode()
# Returns 1 if the current user is in read-only mode, and thus all
writes
# to files and command execution should fail.
sub is_readonly_mode
{
if (!defined($main::readonly_mode_cache)) {
      local %gaccess = &get_module_acl(undef, "");
      $main::readonly_mode_cache = $gaccess{'readonly'} ? 1 : 0;
      }
return $main::readonly_mode_cache;
}


# error([message]+)
# Display an error message and exit. The variable $whatfailed must be
set
# to the name of the operation that failed.
sub error
{
if (!$main::error_must_die) {
      print STDERR "Error: ",@_,"\n";
      }
&load_theme_library();
if ($main::error_must_die) {
      die @_;
      }
elsif (!$ENV{'REQUEST_METHOD'}) {
      # Show text-only error
      print STDERR "$text{'error'}\n";
      print STDERR "-----\n";
      print STDERR ($main::whatfailed ? "$main::whatfailed : " :
""),@_,"\n";
      print STDERR "-----\n";
      if ($gconfig{'error_stack'}) {
            # Show call stack
            print STDERR $text{'error_stack'},"\n";
            for($i=0; my @stack = caller($i); $i++) {
                  print STDERR &text('error_stackline',
                        $stack[1], $stack[2], $stack[3]),"\n";
                  }
            }
```

```
	}
elsif (defined(&theme_error)) {
	&theme_error(@_);
	}
else {
	&header($text{'error'}, "");
	print "<hr>\n";
	print "<h3>",($main::whatfailed ? "$main::whatfailed : " :
"",@_,"</h3>\n";
	if ($gconfig{'error_stack'}) {
		# Show call stack
		print "<h3>$text{'error_stack'}</h3>\n";
		print "<table>\n";
		print "<tr> <td><b>$text{'error_file'}</b></td> ",
			"<td><b>$text{'error_line'}</b></td> ",
			"<td><b>$text{'error_sub'}</b></td> </tr>\n";
		for($i=0; my @stack = caller($i); $i++) {
			print "<tr>\n";
			print "<td>$stack[1]</td>\n";
			print "<td>$stack[2]</td>\n";
			print "<td>$stack[3]</td>\n";
			print "</tr>\n";
			}
		print "</table>\n";
		}
	print "<hr>\n";
	if ($ENV{'HTTP_REFERER'} && $main::completed_referers_check) {
		&footer($ENV{'HTTP_REFERER'}, $text{'error_previous'});
		}
	else {
		&footer();
		}
	}
&unlock_all_files();
&cleanup_tempnames();
exit;
}


# load_theme_library()
# For internal use only
sub load_theme_library
{
return if (!$current_theme || !$tconfig{'functions'} ||
	   $loaded_theme_library++);
do "$theme_root_directory/$tconfig{'functions'}";
}


# unlock_all_files()
# Unlocks all files locked by this program
sub unlock_all_files
{
foreach $f (keys %main::locked_file_list) {
	&unlock_file($f);
	}
}


# cleanup_tempnames()
# Remove all temporary files
```

```
sub cleanup_tempnames
{
local $t;
foreach $t (@main::temporary_files) {
      &unlink_file($t);
      }
@main::temporary_files = ( );
}


# text(message, [substitute]+)
sub text
{
local $rv = $text{$_[0]};
local $i;
for($i=1; $i<@_; $i++) {
      $rv =~ s/\$$i/$_[$i]/g;
      }
return $rv;
}
```

delete_share.pl

```
#!/usr/bin/perl
$share_name=@ARGV;
&delete_share(@ARGV);

# delete_share(share)
# Delete some share from the config file
sub delete_share
{
local($_, @conf, $deleting);
&open_readfile(CONF, "/etc/samba/smb.conf");
@conf = <CONF>;
close(CONF);
&open_tempfile(CONF, "> /etc/samba/smb.conf");
for($i=0; $i<@conf; $i++) {
      chop($_ = $conf[$i]); s/;.*$//g;
      if (/^\s*\[([^\]]+)\]/) {
            if ($deleting) { $deleting = 0; }
            elsif ($1 eq $_[0]) {
                  &print_tempfile(CONF, "\n");
                  $deleting = 1;
                  }
            }
      if (!$deleting) {
            &print_tempfile(CONF, $conf[$i]);
            }
      }
&close_tempfile(CONF);
}


# open_readfile(handle, file)
# Opens some file for reading. Returns 1 on success, 0 on failure
sub open_readfile
{
local ($fh, $file) = @_;
local $realfile = &translate_filename($file);
return open($fh, "<".$realfile);
}
```

```
# translate_filename(filename)
# Applies all relevant registered translation functions to a filename
sub translate_filename
{
local $realfile = $_[0];
local @funcs = grep { $_->[0] eq $module_name ||
                      !defined($_->[0]) } @main::filename_callbacks;
local $f;
foreach $f (@funcs) {
      local $func = $f->[1];
      $realfile = &$func($realfile, @{$f->[2]});
      }
return $realfile;
}


# open_tempfile([handle], file, [no-error], [no-tempfile], [safe?])
# Returns a temporary file for writing to some actual file
sub open_tempfile
{
if (@_ == 1) {
      # Just getting a temp file
      if (!defined($main::open_tempfiles{$_[0]})) {
            $_[0] =~ /^(.*)\/(.*)$/ || return $_[0];
            local $dir = $1 || "/";
            local $tmp = "$dir/$2.webmintmp.$$";
            $main::open_tempfiles{$_[0]} = $tmp;
            push(@main::temporary_files, $tmp);
            }
      return $main::open_tempfiles{$_[0]};
      }
else {
      # Actually opening
      local ($fh, $file, $noerror, $notemp, $safe) = @_;
      local %gaccess = &get_module_acl(undef, "");
      if ($file =~ /\r|\n|\0/) {
            if ($noerror) { return 0; }
            else { &error("Filename contains invalid characters"); }
            }
      if (&is_readonly_mode() && $file =~ />/ && !$safe) {
            # Read-only mode .. veto all writes
            print STDERR "vetoing write to $file\n";
            return open($fh, ">$null_file");
            }
      elsif ($file =~ /^(>|>>)\/dev\// || lc($file) eq "nul") {
            # Writes to /dev/null or TTYs don't need to be handled
            return open($fh, $file);
            }
      elsif ($file =~ /^>\s*(([a-zA-Z]:)?\/.*)$/ && !$notemp) {
            # Over-writing a file, via a temp file
            $file = $1;
            $file = &translate_filename($file);
            while(-l $file) {
                  # Open the link target instead
                  $file = &resolve_links($file);
                  }
            if (-d $file) {
                  # Cannot open a directory!
                  if ($noerror) { return 0; }
                  else { &error("Cannot write to directory"); }
```

```
                }
            local $tmp = &open_tempfile($file);
            local $ex = open($fh, ">$tmp");
            if (!$ex && $! =~ /permission/i) {
                # Could not open temp file .. try opening actual
file
                # instead directly
                $ex = open($fh, ">$file");
                delete($main::open_tempfiles{$file});
                }
            else {
                $main::open_temphandles{$fh} = $file;
                }
            binmode($fh);
            if (!$ex && !$noerror) {
                &error(&text("efileopen", $file, $!));
                }
            return $ex;
            }
      elsif ($file =~ /^>\s*(([a-zA-Z]:)?\/.*)$/ && $notemp) {
            # Just writing direct to a file
            $file = $1;
            $file = &translate_filename($file);
            local $ex = open($fh, ">$file");
            $main::open_temphandles{$fh} = $file;
            if (!$ex && !$noerror) {
                &error(&text("efileopen", $file, $!));
                }
            binmode($fh);
            return $ex;
            }
      elsif ($file =~ /^>>\s*(([a-zA-Z]:)?\/.*)$/) {
            # Appending to a file .. nothing special to do
            $file = $1;
            $file = &translate_filename($file);
            local $ex = open($fh, ">>$file");
            $main::open_temphandles{$fh} = $file;
            if (!$ex && !$noerror) {
                &error(&text("efileopen", $file, $!));
                }
            binmode($fh);
            return $ex;
            }
      elsif ($file =~ /^([a-zA-Z]:)?\//) {
            # Read mode .. nothing to do here
            $file = &translate_filename($file);
            return open($fh, $file);
            }
      elsif ($file eq ">" || $file eq ">>") {
            local ($package, $filename, $line) = caller;
            if ($noerror) { return 0; }
            else { &error("Missing file to open at
${package}::${filename} line $line"); }
            }
      else {
            # XXX append / update support?
            local ($package, $filename, $line) = caller;
            &error("Unsupported file or mode $file at
${package}::${filename} line $line");
            }
      }
```

```
}

# close_tempfile(file|handle)
# Copies a temp file to the actual file, assuming that all writes
were
# successful.
sub close_tempfile
{
local $file;
if (defined($file = $main::open_temphandles{$_[0]})) {
      # Closing a handle
      close($_[0]) || &error(&text("efileclose", $file, $!));
      delete($main::open_temphandles{$_[0]});
      return &close_tempfile($file);
      }
elsif (defined($main::open_tempfiles{$_[0]})) {
      # Closing a file
      local @st = stat($_[0]);
      if ($gconfig{'os_type'} =~ /-linux$/ && &has_command("chcon"))
{
            # Set original security context
            system("chcon --reference=".quotemeta($_[0]).
                   " ".quotemeta($main::open_tempfiles{$_[0]}).
                   " >/dev/null 2>&1");
            }
      rename($main::open_tempfiles{$_[0]}, $_[0]) || &error("Failed
to replace $_[0] with $main::open_tempfiles{$_[0]} : $!");
      if (@st) {
            # Set original permissions and ownership
            chmod($st[2], $_[0]);
            chown($st[4], $st[5], $_[0]);
            }
      delete($main::open_tempfiles{$_[0]});
      @main::temporary_files = grep { $_ ne
$main::open_tempfiles{$_[0]} } @main::temporary_files;
      if ($main::open_templocks{$_[0]}) {
            &unlock_file($_[0]);
            delete($main::open_templocks{$_[0]});
            }
      return 1;
      }
else {
      # Must be closing a handle not associated with a file
      close($_[0]);
      return 1;
      }
}

# print_tempfile(handle, text, ...)
# Like the normal print function, but calls &error on failure
sub print_tempfile
{
local ($fh, @args) = @_;
#(print $fh @args) || &error(&text("efilewrite",
#                     $main::open_temphandles{$fh} || $fh, $!));

print $fh @args;
}


# get_module_acl([user], [module], [no-rbac], [no-default])
```

```
# Returns a hash  containing access control options for the given
user
sub get_module_acl
{
local %rv;
local $u = defined($_[0]) ? $_[0] : $base_remote_user;
local $m = defined($_[1]) ? $_[1] : $module_name;
local $mdir = &module_root_directory($m);
if (!$_[3]) {
      &read_file_cached("$mdir/defaultacl", \%rv);
      }
local %usersacl;
if (!$_[2] && &supports_rbac($m) && &use_rbac_module_acl($u, $m)) {
      # RBAC overrides exist for this user in this module
      local $rbac = &get_rbac_module_acl(
                  defined($_[0]) ? $_[0] : $remote_user, $m);
      local $r;
      foreach $r (keys %$rbac) {
            $rv{$r} = $rbac->{$r};
            }
      }
elsif ($gconfig{"risk_$u"} && $m) {
      # ACL is defined by user's risk level
      local $rf = $gconfig{"risk_$u"}.'.risk';
      &read_file_cached("$mdir/$rf", \%rv);

      local $sf = $gconfig{"skill_$u"}.'.skill';
      &read_file_cached("$mdir/$sf", \%rv);
      }
else {
      # Use normal Webmin ACL
      &read_file_cached("$config_directory/$m/$u.acl", \%rv);
      if ($remote_user ne $base_remote_user && !defined($_[0])) {

      &read_file_cached("$config_directory/$m/$remote_user.acl",\%rv)
;
            }
      }
if ($tconfig{'preload_functions'}) {
      &load_theme_library();
      }
if (defined(&theme_get_module_acl)) {
      %rv = &theme_get_module_acl($u, $m, \%rv);
      }
return %rv;
}


# module_root_directory(module)
# Given a module name, returns its root directory
sub module_root_directory
{
local $d = ref($_[0]) ? $_[0]->{'dir'} : $_[0];
if (@root_directories > 1) {
      local $r;
      foreach $r (@root_directories) {
            if (-d "$r/$d") {
                  return "$r/$d";
                  }
            }
      }
```

```
return "$root_directories[0]/$d";
}


# read_file_cached(file, &assoc)
# Like read_file, but reads from a cache if the file has already been
read
sub read_file_cached
{
local $realfile = &translate_filename($_[0]);
if (defined($main::read_file_cache{$realfile})) {
      %{$_[1]} = ( %{$_[1]}, %{$main::read_file_cache{$realfile}} );
      }
else {
      local %d;
      &read_file($_[0], \%d, $_[2], $_[3], $_[4]);
      %{$main::read_file_cache{$realfile}} = %d;
      %{$_[1]} = ( %{$_[1]}, %d );
      }
}


# read_file(file, &assoc, [&order], [lowercase], [split-char])
# Fill an associative array with name=value pairs from a file
sub read_file
{
local $_;
local $split = defined($_[4]) ? $_[4] : "=";
local $realfile = &translate_filename($_[0]);
&open_readfile(ARFILE, $_[0]) || return 0;
while(<ARFILE>) {
      chomp;
      local $hash = index($_, "#");
      local $eq = index($_, $split);
      if ($hash != 0 && $eq >= 0) {
            local $n = substr($_, 0, $eq);
            local $v = substr($_, $eq+1);
            chomp($v);
            $_[1]->{$_[3] ? lc($n) : $n} = $v;
            push(@{$_[2]}, $n) if ($_[2]);
            }
        }
close(ARFILE);
if (defined($main::read_file_cache{$realfile})) {
      %{$main::read_file_cache{$realfile}} = %{$_[1]};
      }
return 1;
}


# supports_rbac([module])
# Returns 1 if RBAC client support is available
sub supports_rbac
{
return 0 if ($gconfig{'os_type'} ne 'solaris');
eval "use Authen::SolarisRBAC";
return 0 if ($@);
if ($_[0]) {
      #return 0 if (!-r &module_root_directory($_[0])."/rbac-
mapping");
      }
```

```
return 1;
}


# is_readonly_mode()
# Returns 1 if the current user is in read-only mode, and thus all
writes
# to files and command execution should fail.
sub is_readonly_mode
{
if (!defined($main::readonly_mode_cache)) {
      local %gaccess = &get_module_acl(undef, "");
      $main::readonly_mode_cache = $gaccess{'readonly'} ? 1 : 0;
      }
return $main::readonly_mode_cache;
}

# error([message]+)
# Display an error message and exit. The variable $whatfailed must be
set
# to the name of the operation that failed.
sub error
{
if (!$main::error_must_die) {
      print STDERR "Error: ",@_,"\n";
      }
&load_theme_library();
if ($main::error_must_die) {
      die @_;
      }
elsif (!$ENV{'REQUEST_METHOD'}) {
      # Show text-only error
      print STDERR "$text{'error'}\n";
      print STDERR "-----\n";
      print STDERR ($main::whatfailed ? "$main::whatfailed : " :
""),@_,"\n";
      print STDERR "-----\n";
      if ($gconfig{'error_stack'}) {
            # Show call stack
            print STDERR $text{'error_stack'},"\n";
            for($i=0; my @stack = caller($i); $i++) {
                  print STDERR &text('error_stackline',
                        $stack[1], $stack[2], $stack[3]),"\n";
                  }
            }

      }
elsif (defined(&theme_error)) {
      &theme_error(@_);
      }
else {
      &header($text{'error'}, "");
      print "<hr>\n";
      print "<h3>",($main::whatfailed ? "$main::whatfailed : " :
""),@_,"</h3>\n";
      if ($gconfig{'error_stack'}) {
            # Show call stack
            print "<h3>$text{'error_stack'}</h3>\n";
            print "<table>\n";
            print "<tr> <td><b>$text{'error_file'}</b></td> ",
                  "<td><b>$text{'error_line'}</b></td> ",
```

```
                "<td><b>$text{'error_sub'}</b></td> </tr>\n";
            for($i=0; my @stack = caller($i); $i++) {
                print "<tr>\n";
                print "<td>$stack[1]</td>\n";
                print "<td>$stack[2]</td>\n";
                print "<td>$stack[3]</td>\n";
                print "</tr>\n";
                }
            print "</table>\n";
            }
        print "<hr>\n";
        if ($ENV{'HTTP_REFERER'} && $main::completed_referers_check) {
            &footer($ENV{'HTTP_REFERER'}, $text{'error_previous'});
            }
        else {
            &footer();
            }
        }
&unlock_all_files();
&cleanup_tempnames();
exit;
}


# error([message]+)
# Display an error message and exit. The variable $whatfailed must be
set
# to the name of the operation that failed.
sub error
{
if (!$main::error_must_die) {
        print STDERR "Error: ",@_,"\n";
        }
&load_theme_library();
if ($main::error_must_die) {
        die @_;
        }
elsif (!$ENV{'REQUEST_METHOD'}) {
        # Show text-only error
        print STDERR "$text{'error'}\n";
        print STDERR "-----\n";
        print STDERR ($main::whatfailed ? "$main::whatfailed : " :
"",@_,"\n";
        print STDERR "-----\n";
        if ($gconfig{'error_stack'}) {
            # Show call stack
            print STDERR $text{'error_stack'},"\n";
            for($i=0; my @stack = caller($i); $i++) {
                print STDERR &text('error_stackline',
                    $stack[1], $stack[2], $stack[3]),"\n";
                }
            }

        }
elsif (defined(&theme_error)) {
        &theme_error(@_);
        }
else {
        &header($text{'error'}, "");
        print "<hr>\n";
```

```
	print "<h3>",($main::whatfailed ? "$main::whatfailed : " :
"",@_,"</h3>\n";
	if ($gconfig{'error_stack'}) {
		# Show call stack
		print "<h3>$text{'error_stack'}</h3>\n";
		print "<table>\n";
		print "<tr> <td><b>$text{'error_file'}</b></td> ",
			"<td><b>$text{'error_line'}</b></td> ",
			"<td><b>$text{'error_sub'}</b></td> </tr>\n";
		for($i=0; my @stack = caller($i); $i++) {
			print "<tr>\n";
			print "<td>$stack[1]</td>\n";
			print "<td>$stack[2]</td>\n";
			print "<td>$stack[3]</td>\n";
			print "</tr>\n";
			}
		print "</table>\n";
		}
	print "<hr>\n";
	if ($ENV{'HTTP_REFERER'} && $main::completed_referers_check) {
		&footer($ENV{'HTTP_REFERER'}, $text{'error_previous'});
		}
	else {
		&footer();
		}
	}
&unlock_all_files();
&cleanup_tempnames();
exit;
}


# load_theme_library()
# For internal use only
sub load_theme_library
{
return if (!$current_theme || !$tconfig{'functions'} ||
	   $loaded_theme_library++);
do "$theme_root_directory/$tconfig{'functions'}";
}


# unlock_all_files()
# Unlocks all files locked by this program
sub unlock_all_files
{
foreach $f (keys %main::locked_file_list) {
	&unlock_file($f);
	}
}


# cleanup_tempnames()
# Remove all temporary files
sub cleanup_tempnames
{
local $t;
foreach $t (@main::temporary_files) {
	&unlink_file($t);
	}
@main::temporary_files = ( );
```

```
}


# text(message, [substitute]+)
sub text
{
local $rv = $text{$_[0]};
local $i;
for($i=1; $i<@_; $i++) {
      $rv =~ s/\$$i/$_[$i]/g;
      }
return $rv;
}


# load_theme_library()
# For internal use only
sub load_theme_library
{
return if (!$current_theme || !$tconfig{'functions'} ||
         $loaded_theme_library++);
do "$theme_root_directory/$tconfig{'functions'}";
}


# unlock_all_files()
# Unlocks all files locked by this program
sub unlock_all_files
{
foreach $f (keys %main::locked_file_list) {
      &unlock_file($f);
      }
}


# cleanup_tempnames()
# Remove all temporary files
sub cleanup_tempnames
{
local $t;
foreach $t (@main::temporary_files) {
      &unlink_file($t);
      }
@main::temporary_files = ( );
}


# text(message, [substitute]+)
sub text
{
local $rv = $text{$_[0]};
local $i;
for($i=1; $i<@_; $i++) {
      $rv =~ s/\$$i/$_[$i]/g;
      }
return $rv;
}
```

delsmb.pl

```
#!/usr/bin/perl -T
```

```
# delsmb.pl version
# removes both Linux and Samba user
#
# Copyright (C) 2002 Alceu Rodrigues de Freitas Jr.
(glasswalk3r@yahoo.com.br)
#    This program is free software; you can redistribute it and/or
modify
#    it under the terms of the GNU General Public License as
published by
#    the Free Software Foundation; either version 2 of the License,
or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public
License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-
1307  USA
########################################################################
###
use strict;
use warnings;
use Fcntl qw (:DEFAULT :flock);
use Crypt::SmbHash '0.12';
use File::Temp;
use Tie::File;
use File::Copy;
use File::Path;
use Passwd::Linux qw( rmpwnam );
use FAUS::Helper qw( :DEFAULT PASSWD );
use sigtrap qw(handler abort normal-signals);
use constant VERSION => 1.8;
use Getopt::Std;

clean_env();
my %opts;
getopts( 'u:r', \%opts );

my $temp_file = tmpnam();
my $file_obj;
my @smb_content;

validate();
smb_remove();
unix_remove();

################
# functions
################

sub validate {

    help_message() unless( %opts );
    help_message() unless( exists( $opts{u} ) );
    help_message() unless( defined( $opts{u} ) );
```

```
    error( 'Invalid characters in the Samba password file pathname' )
unless( SMBPASSWD =~ /^[\w\/]+smbpasswd$/ );

    my $user_id = getpwnam( $opts{u} );

    error( 'You must give a username as parameter' ) unless (
defined( $opts{u} ) );
    error( "The user $opts{u} does not exist" ) unless ( defined(
$user_id ) );
    error( 'I will not deal with system users!' ) unless ( is_uid_ok(
$user_id ) );
    $opts{u} =~/^([\w\$]+)$/ ? ( $opts{u} = $1 ) : error( 'Invalid
characters in user parameter' );

}

# Erasing first from smbpasswd file
sub smb_remove {

    my $user_exists = 0;
    my $counter = 0;
    my $search = quotemeta( $opts{u} );

    if ( -e SMBPASSWD ) {

      copy( SMBPASSWD, $temp_file ) or error( "Backup copy failed:
$!" );

    } else {

      error( "The file SMBPASSWD does not exist" );

    }

    $file_obj = tie( @smb_content, 'Tie::File', SMBPASSWD, mode =>
O_RDWR, memory => 0 ) or error( "Cannot read SMBPASSWD: $!" );
    $file_obj->flock(LOCK_EX);

# check if the user already exists in the file

    foreach ( @smb_content ) {

      if ( /^$search/o ) {

         $user_exists = 1;
         last;

      } else {

         $counter++;

      }

    }


    if ( $user_exists ) {

# removes now the user entry
      splice( @smb_content, $counter, 1 );
      undef $file_obj;
```

```
      untie @smb_content;
      unlink( $temp_file ) or error( "Failed to remove backup file
$temp_file: $!" );

    } else {

      error("The user $opts{u} does not exists in SMBPASSWD file" );

    }
}


# removes from /etc/passwd and /etc/shadow
sub unix_remove {

# the user demands to erase everything
    if ( $opts{r} ) {

# can't delete home directory of machine accounts

      if ( $opts{u} =~ /\$$/) {

        remove( $opts{u} );

      } else {

#checks to see if the user HOME directory does exists
          if ( -e ( getpwnam( $opts{u} ) )[7] ) {

            removeall( $opts{u} );

          } else {

            remove( $opts{u} );

          }

      }

# only the entries in smbpasswd and /etc/passwd; maintain home
directory
    } else {

      remove( $opts{u} );

    }
}

sub remove {

    my $user = shift;
    my $return = rmpwnam( $user );

    error( "Cannot remove user $user from " . PASSWD . ": $!" ) if (
$return < 0 );
    error( 'Operation attemped on uid 0' ) if ( $return == 1 );
    error( 'User $user does not exists' ) if ( $return == 2 );

}

sub removeall {
```

```
    my $user = shift;
    my $user_home = ( getpwnam( $user ) )[7];

    error( "Cannot remove $user home diretory $user_home" ) unless(
rmtree( $user_home, 0, 1 ) );
    remove( $user );

}

# dies, but before tries to close the reference file
sub error {

    my $message = shift;
    my $date = localtime( time );

    untie @smb_content if ( @smb_content );
    undef $file_obj if ( defined( $file_obj ) );

#restores backup
    if ( -e $temp_file ) {

      copy( $temp_file, SMBPASSWD ) or warn "Failed to restore
SMBPASSWD backup file: $!\n";
      unlink( $temp_file ) or warn "Failed to remove backup file:
$!\n";

    } else {

      warn "Backup file does not exists\n";

    }

    die get_label() . $message . "\n";

}


sub abort {

    my $signame = shift;
    error( "Aborted due an received SIG$signame signal." );

}

sub help_message {

    my $regards_to = \REGARDS_TO;
    my $version = \VERSION;
    my $message =<<BLOCK;
delsmb - version $$version
delsmb $$regards_to
Usage: delsmb -u<user> [-r]
       -r removes the user /home directory, if it exists

BLOCK

    die $message;

}
```

getpassfile.pl

```
#!/usr/bin/perl -T
# getpassfile.pl
# Reads smbpasswd file and formats it's output to send to samba.cgi
# Copyright (C) 2002 Alceu Rodrigues de Freitas Jr.
(glasswalk3r@yahoo.com.br)
#    This program is free software; you can redistribute it and/or
modify
#    it under the terms of the GNU General Public License as
published by
#    the Free Software Foundation; either version 2 of the License,
or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public
License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-
1307  USA
#########################################################################
###
use FAUS::Helper;
use constant VERSION => 1.7;

# very crude help message
if ( @ARGV ) {

    my $version = \VERSION;
    my $regards_to = \REGARDS_TO;

    my $message = <<BLOCK;
getpassfile version $$version
getpassfile $$regards_to
Usage: getpassfile

BLOCK

    die $message;

}

clean_env();

my @fields;
my @passwd_info;
my $entry;
my @flags;
my @new_flags;

# untainting value
die get_label() . "Invalid characters in pathname parameter.\n"
unless( SMBPASSWD =~ /^[\w\/]+smbpasswd$/ );

open( FILE, '<' . SMBPASSWD ) or die get_label() . "Cannot read the
smbpasswd file: $!\n";

while (<FILE>) {
```

```
      chomp;
      @fields = split( /:/, $_);
      @passwd_info = getpwnam( $fields[0] );

# identifying flags
      $fields[4] =~ tr/[]//d;
      my @flags = unpack( 'AAA', $fields[4] );

# positions and means
# 0 - type of user (U = regular user; W = machine account)
# 1 - is user blocked? (true or false)
# 2 - password expires? (true or false)
      my @new_flags = qw( U 0 1 );
      my $flag;

      foreach $flag( @flags ) {
          CASE: {
                  last CASE unless ( defined( $flag ) );
                  if ( $flag eq 'W' ) {
                        $new_flags[0] = 'W';
                        last CASE;
                  }
                  if ( $flag eq 'U' ) {
                        # this is already the default
                        last CASE;
                  }
                  if ( $flag eq 'D' ) {
                        $new_flags[1] = 1;
                        last CASE;
                  }
                  if ( $flag eq 'X' ) {
                        $new_flags[2] = 0;
                        last CASE;
                  }
          }
      }

#     $entry = pack( 'A8AAA2A8', $fields[0], @new_flags,
$passwd_info[3] );
      print @new_flags, "//",
$fields[0],"//",$passwd_info[3],"//",$passwd_info[2],"||";
#     print @passwd_info;
#     print $entry, "\\\\";
      @fields = ();
      @passwd_info = ();
#     $entry = '';
      @new_flags = ();
      @flags = ();
}
close(FILE);
```

list_shares.pl

```
#!/usr/bin/perl
@shares=&list_shares();

# print the shares
foreach (@shares)
{
      if($_=~ /global|print/)
      {
```

```
	}
	else
	{
		print($_);
		print("\n");
	}
}

# list_shares()
# List all the shares from the samba config file
sub list_shares
{
local(@rv, $_);
&open_readfile(SAMBA, "/etc/samba/smb.conf");
while(<SAMBA>) {
	chop; s/;.*$//g; s/^\s*#.*$//g;
	if (/^\s*\[([^\]]+)\]/) {
		push(@rv, $1);
	#	print ($1);
	#	print ("\n");
		}
	}
close(SAMBA);

# Check for an include directive in the [global] share
local %global;
&get_share("global", \%global);
local $inc = &getval("include", \%global);
if ($inc && $inc !~ /\%/) {
	# XXX
	}

return @rv;
}


# open_readfile(handle, file)
# Opens some file for reading. Returns 1 on success, 0 on failure
sub open_readfile
{
local ($fh, $file) = @_;
local $realfile = &translate_filename($file);
return open($fh, "<".$realfile);
}


# translate_filename(filename)
# Applies all relevant registered translation functions to a filename
sub translate_filename
{
local $realfile = $_[0];
local @funcs = grep { $_->[0] eq $module_name ||
		      !defined($_->[0]) } @main::filename_callbacks;
local $f;
foreach $f (@funcs) {
	local $func = $f->[1];
	$realfile = &$func($realfile, @{$f->[2]});
	}
return $realfile;
}
```

```
# get_share(share, [array])
# Fills the associative array %share with the parameters from the
given share
sub get_share
{
local($found, $_, $first, $arr);
$arr = (@_==2 ? $_[1] : "share");
undef(%$arr);
&open_readfile(SAMBA, "/etc/samba/smb.conf");
while(<SAMBA>) {
      chop; s/^\s*;.*$//g; s/^\s*#.*$//g;
      if (/^\s*\[([^\]]+)\]/) {
            # Start of share section
            $first = 1;
            if ($found) { last; }
            elsif ($1 eq $_[0]) { $found = 1; $$arr{share_name} = $1;
}
            }
      elsif ($found && /^\s*([^=]*\S)\s*=\s*(.*)$/) {
            # Directives inside a section
            if (lc($1) eq "read only") {
                  # bastard special case.. change to writable
                  $$arr{'writable'} = $2 =~ /yes|true|1/i ? "no" :
"yes";
                  }
            else { $$arr{lc($1)} = $2; }
            }
      elsif (!$first && /^\s*([^=]*\S)\s*=\s*(.*)$/ && $_[0] eq
"global") {
            # Directives outside a section! Assume to be part of
[global]
            $$arr{share_name} = "global";
            $$arr{lc($1)} = $2;
            $found = 1;
            }
      }
close(SAMBA);
return $found;
}


# getval(name, [&hash])
# Given the name of a key in %share, return the value. Also looks for
synonyms.
# If the value is not found, a default is looked for.. this can come
from
# a copied section, the [global] configuration section, or from the
SAMBA
# defaults. This means that getval() always returns something..
sub getval
{
local $hash = $_[1] || \%share;
local($_, $copy);
if ($synon{$_[0]}) {
      foreach (split(/,/, $synon{$_[0]})) {
            if (defined($hash->{$_})) { return $hash->{$_}; }
            }
      }
if (defined($hash->{$_[0]})) {
      return $hash->{$_[0]};
```

```
	}
elsif ($_[0] ne "copy" && ($copy = $hash->{"copy"})) {
	# this share is a copy.. get the value from the source
	local(%share);
	&get_share($copy);
	return &getval($_[0]);
	}
else {
	# return the default value...
	return &default_value($_[0]);
	}
return undef;
}


# default_value(name)
# Returns the default value for a parameter
sub default_value
{
local($_, %global);

# First look in the [global] section.. (unless this _is_ the global
section)
if ($share{share_name} ne "global") {
	&get_share("global", "global");
	if ($synon{$_[0]}) {
		foreach (split(/,/, $synon{$_[0]})) {
			if (defined($global{$_})) { return $global{$_}; }
			}
		}
	if (defined($global{$_[0]})) { return $global{$_[0]}; }
	}

# Else look in the samba defaults
if ($synon{$_[0]}) {
	foreach (split(/,/, $synon{$_[0]})) {
		if (exists($default_values{$_})) {
			return $default_values{$_};
			}
		}
	}
return $default_values{$_[0]};
}
```