Λειτουργικά Συστήματα (ΗΥ321)

Διάλεξη 18:
Μηχανισμοί – Πολιτικές Προστασίας

Χρήστος Δ. Αντωνόπουλος
25/5/2017

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Προστασία

- Σκοπός: Προστασία από την καταλάθος ή ηθελημένη κακή χρήση του συστήματος
- Ατυχήματα
  - Ένα πρόγραμμα σβήνει καταλάθος τον κατάλογο /. Το σύστημα είναι πλέον άχρηστο
  - Λύνεται ευκολα: απλά κάνε μικρή την πιθανότητα
- Κακίες
  - Η αρχιχακερού της 3ης γυμνασίου “σπάει” το σύστημα της ΤτΕ και μεταφέρει $3MEuros στο λογαριασμό της για να αγοράσει CD του Sakis.
  - Δύσκολο πρόβλημα, δε μας παίρνει να παίξουμε με πιθανότητες

Πολιτική - Μηχανισμός

- Καλή ιδέα: Να ξεχωρίσουμε την πολιτική (τι) από το μηχανισμό (πως)
- Σύστημα προστασίας: Ο μηχανισμός επιβολής της πολιτικής ασφάλειας.
  - Λίγο πολύ τα ίδια, ανεξαρτήτως πολιτικής
  - Πολιτική: Τι είναι αποδεκτή και τι μη αποδεκτή συμπεριφορά
- Παραδείγματα πολιτικών ασφαλείας:
  - Το πολύ 40MB/χρήστη στο δίσκο
  - Μόνο ο root μπορεί να γράψει το αρχείο με τα passwords
  - Μόνο εγώ μπορώ να διαβάσω το mail μου

Βασικά Συστατικά Μηχανισμού Προστασίας

- Ταυτοποίηση (authentication): Βεβαιώσου ότι αυτός που σου μιλάει είναι αυτός που δηλώνει

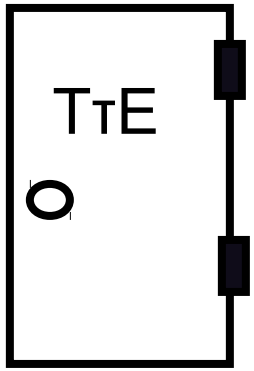
  - Unix: password
  - Πιστωτικές κάρτες: # Ταυτότητας + Διεύθυνση λογαριασμού
  - Οδηγός: Άδεια οδήγησης
- Authorization: Δες αν ο x επιτρέπεται να κάνει το y.
  - Χρειάζεται μια απλή βάση δεδομένων
- Επιβολή: Επέβαλε την απόφαση του authorization
  - Βεβαιώσου ότι δεν υπάρχουν τρύπες
  - Δύσκολο. Ένα μόνο πρόβλημα και το σύστημα καταρρέει

Δεν υπάρχει Τέλειο Σύστημα Προστασίας

- Βασικό σημείο, εύκολο να παραβλεφθεί:
  - Προστασία: μπορεί μόνο να αυξήσει την απαιτούμενη προσπάθεια για να γίνει κάτι κακό
  - Δε μπορεί να το εμποδίσει
- Ακόμα και σε τεχνικά άψογο σύστημα, υπάρχουν τα 4 Bs:
  - Burglary (Διάρρηξη): Αν δε μπορείς να «σπάσεις» το σύστημα, μπορείς να το κλέψεις («φυσική ασφάλεια»)
  - Bribery (Δωροδοκία): Δωροδόκησε κάποιον που έχει νόμιμη πρόσβαση στο σύστημα
  - Blackmail (Εκβιασμός): Εναλλακτικά φωτογράφισέ τον σε μια «δύσκολη» στιγμή.
  - Bludgeoning (Βία): Η απλά μαύρισέ τον στο ξύλο
- Κάθε σύστημα έχει τρύπες (με κάποια μορφή)

Κοινωνική & Τεχνική Επιβολή

- Τεχνικοί μηχανισμοί για την επιβολή της πολιτικής ασφάλειας (π.χ. βάλε πόρτα και κλείδωσέ τη)



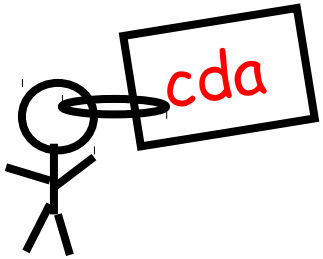
- Ή κοινωνικοί μηχανισμοί: Κάνε την κλοπή παράνομη
  - +: «Εγκαθίσταται» άμεσα παντού, 0 επιβάρυνση, πίσω συμβατότητα, κλπ.
  - -: Απαιτεί αποδοτικό μηχανισμό αποκάλυψης και επιβολής.
    - Οι νόμοι δύσκολο να επιβληθούν
  - Δουλεύει καλά όταν η απόσταση είναι μικρή. Δύσκολο να επιβληθεί στην Ελλάδα νόμος από την Αγγλία

Ταυτοποίηση

- Συνήθως με passwords.
  - «Κακός» τρόπος. Πρέπει να το θυμόμαστε
  - Συνήθως βασίζεται στο όνομα του έτερου ήμισυ, παιδιού κλπ.
- Τα passwords δεν πρέπει να αποθηκεύονται σε αναγνώσιμη μορφή
  - Χρησιμοποίησε μη αντιστρέψιμο μετασχηματισμό και αποθήκευσε το αποτέλεσμα
  - Δείτε το /etc/shadow: Κάτι «κινέζικα» δίπλα σε κάθε όνομα. Είναι το κρυπτογραφημένο password
- Πρόβλημα: Για να εμποδίσουμε τις «μαντεψιές» («επιθέσεις λεξικού») τα passwords θα πρέπει να είναι μεγάλα και περίεργα
  - Ξεχνιούνται εύκολα και συνήθως γράφονται κάπου

Εναλλακτικές

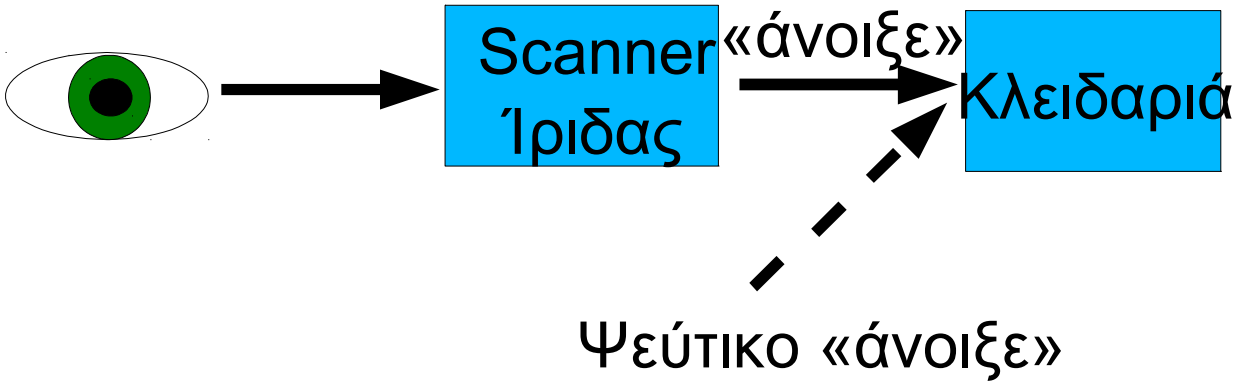
- Κλειδί / κάρτα



- Δεν είναι ανάγκη να μείνει μυστικό
  - Ενίοτε έχει φωτογραφία και φοριέται στο πέτο (π.χ. σε στρατιωτικές βάσεις)
- Δεν πρέπει να μπορεί να αντιγραφεί ή να πλαστογραφηθεί
- Μπορεί να κλαπεί, αλλά ο ιδιοκτήτης θα το καταλάβει
  - Τι κάνουμε τότε; Αν εκδοθεί άλλο, πώς θα ακυρωθεί το παλιό;

Βιομετρικές Μέθοδοι

- Βιομετρικές μέθοδοι
  - Αναγνώριση με χρήση χαρακτηριστικών φυσιολογίας ή συμπεριφοράς
- Παραδείγματα:
  - Πρόσωπο, αποτυπώματα, γεωμετρία παλάμης, γραφικός χαρακτήρας, ίρις, κόρη, φλέβες, φωνή
- Καλή μέθοδος, αλλά πάλι χρειάζεται πλήρες «ασφαλές μονοπάτι».



Authorization

- Αφού βεβαιωθούμε για την ταυτότητα, τι επιτρέπεται να κάνει;
- Αναπαράσταση ως «πίνακας πρόσβασης»
  - 1 γραμμή ανά οντότητα, 1 στήλη ανά πόρο

	File A	Printer 1	TTY 3	…
Usr1	R	W	RW	
Usr2	RW	W	---	
Usr3	---	W	---	

Στην πράξη…

- Ο πλήρης πίνακας λίγο… μεγάλος. 2 «συμπιεσμένες» εναλλακτικές
  - Ικανότητες (Capabilities): Γραμμές. Για κάθε οντότητα, τι επιτρέπεται να κάνει με τον πόρο;
  - Λίστες ελέγχου πρόσβασης (Access control lists – ACLs): Στήλες. Για κάθε πόρο, ποιος επιτρέπεται να τον χρησιμοποιήσει;

	File A	Printer 1	TTY 3	…
Usr1	R	W	RW	
Usr2	RW	W	---	
Usr3	---	W	---	

Access Control Lists (ACLs)

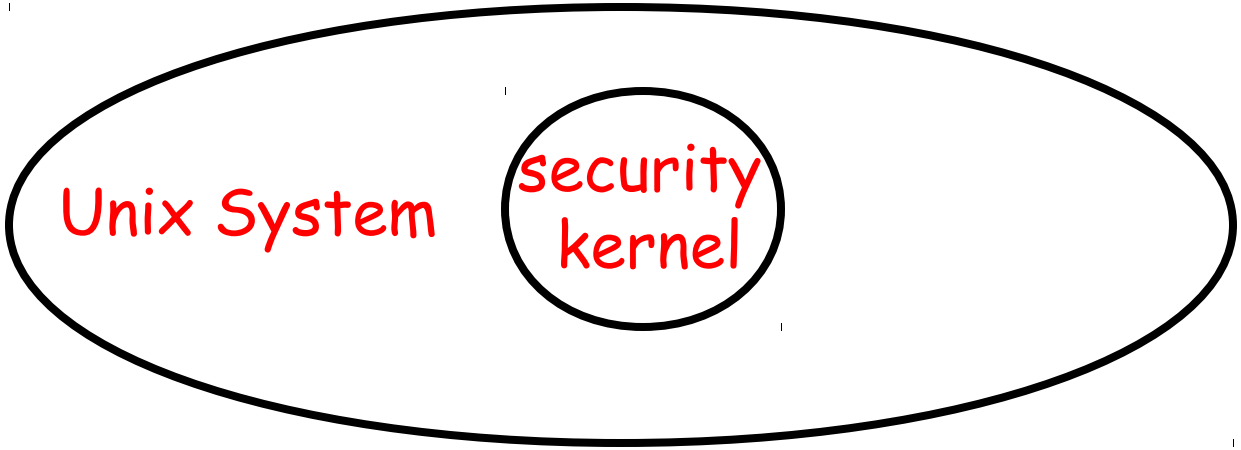
- Για κάθε αντικείμενο, καθόρισε ποιοι χρήστες μπορούν να κάνουν τι
  - Γενικά, κάθε αντικείμενο έχει λίστα από ζεύγη <user,privileges>.
  - Π.χ.: NTFS ACLs
- Πρόβλημα: Μέγεθος των ACLs
  - Επέτρεψε ομάδες: π.χ. Unix 9bits self, group, other.
- ACLs απλά και μπορούν να χρησιμοποιηθούν σε όλα τα συστήματα αρχείων

Ικανότητες (Capabilities)

- Για κάθε χρήστη καθόρισε ποιους πόρους μπορεί να προσπελάσει και με ποιους τρόπους
  - Λίστες ζευγών <object, privilege> για κάθε χρήστη.
    - *Capability List*
- Συχνά χρησιμοποιούνται και για ονοματοδοσία και για προστασία. Μπορείς να δεις ένα αντικείμενο μόνο αν έχεις “capability” για αυτό.
  - Default: Κανένα δικαίωμα.
- Χρησιμοποιούνται σε συστήματα που πρέπει να είναι πολύ ασφαλή
- Π.χ.:
  - Πίνακες σελίδων
  - Τηλ. αριθμοί εκτός καταλόγου
  - Σελίδες WWW που δεν τις «δείχνει» κανείς

Επιβολή

- Κάποιος πρέπει να έχει την ευθύνη της:
  - Επιβολής πολιτικής πρόσβασης.
  - Προστασίας της πληροφορίας ταυτοποίησης.
- Αυτό το τμήμα είναι ο «Θεός» του συστήματος. Κάνει ότι θέλει
  - Αν έχει bug … καταστροφή για το σύστημα
  - Το θέλουμε μικρό (πολύ μικρό) και απλό (πολύ απλό)



  - Τα περισσότερα ΛΣ (Win, Unix) θεωρούν όλο το λειτουργικό άξιο εμπιστοσύνης

Ο… «Αδύναμος Κρίκος»

- Ασφάλεια : Τόσο ισχυρή όσο ο πιο αδύναμος κρίκος του μηχανισμού προστασίας
  - Υλικό;
  - Λογισμικό;
  - Χρήστες!!!

Μερικές εύκολες επιθέσεις

- Κατάχρηση νόμιμου δικαιώματος:
  - Στο Unix ο root κάνει τα πάντα. Μπορεί να διαβάσει το mail σας, να στείλει mail με το όνομά σας, κλπ.
  - Σβήνετε τον κώδικα του 4ου μέρους του project 4. Οι συνεργάτες σας τα παίρνουν….
- Denial of service (DoS)
  - Χρησιμοποίησε όλους τους πόρους και «γονάτισε» το σύστημα.
    - Script: “while(1) { mkdir foo; cd foo; }”
    - Πρόγραμμα C: “while(1) { fork(); malloc(1000)[40] = 1; }”
- Ωτακουστής:
  - Κρυφάκουσε την κίνηση του δικτύου
  - Θα δεις passwords αν γραφτούν μη κρυπτογραφημένα.
  - Η ψάξε για μετακινήσεις αρχείων: Θα μεταφερθούν ως κείμενο.

Παραδείγματα Εισβολής

- Tenex page-fault attack:
  - Σπάσιμο passwords μετρώντας το χρόνο για τον έλεγχο ορθότητας!

- Sendmail/finger worm: Άνοιξη1988. 2 επιθέσεις:
  - Επίθεση Sendmail: Εκμετάλλευση εντολής “debug” που είχε μείνει ενεργή και επέτρεπε εκτέλεση κώδικα ως root. Εγκατάσταση δούρειων ίππων στο σύστημα.
  - Επίθεση Fingerd: Δώσε μεγάλο όνομα στον fingerd, που υπερχειλίζει έναν buffer, αλλάζει τη stack με έξυπνο τρόπο και προκαλεί την εκτέλεση συνάρτησης με δικαιώματα root. Συνηθισμένη επίθεση…

Το Unix hack του Thomson

- Μη εντοπίσιμος δούρειος ίππος
- Κάνε το πρόγραμμα login να αναγνωρίζει ένα «ειδικό» όνομα.
- Πολύ προφανές, άλλαξε τον compiler ώστε ο σχετικός κώδικας να προστίθεται στη μεταγλώττιση.
- Ορατό στον κώδικα του compiler, βάλε κώδικα στον compiler του compiler (bootstrap compiler) ο οποίος θα βάζει κώδικα στον compiler για να βάζει κώδικα στο πρόγραμμα login!
  - Η «παλιοδουλειά» γίνεται μόνο στον bootstrap compiler!

- Όταν γίνει εισβολή το σύστημα μπορεί να μη μπορεί να εξασφαλιστεί ποτέ ξανά
  - Ο εισβολέας μπορεί να έχει αφήσει τρύπες για να ξαναπάρει τον έλεγχο

Παρατηρήσεις

- Δεν είναι εύκολο να βρεις την εισβολή. Ο κακός μπορεί να έχει καθαρίσει τα ίχνη του.
  - Π.χ.: Εταιρεία αρνιόταν την εισβολή στα συστήματά της, τη στιγμή που ο εισβολέας είχε ομολογήσει!

- Αν δεν είμαστε σίγουροι ότι το σύστημα δεν έχει bugs, δεν είμαστε σίγουροι ότι είναι ασφαλές
  - Ενδεχομένως bug = τρύπα ασφαλείας

Λύσεις: Τίποτα δε δουλεύει τέλεια

- Καταγραφή (logging): Κατέγραψε όλες τις σημαντικές ενέργειες και χρήσεις δικαιωμάτων σε αρχείο που δε σβήνεται
  - Μπορεί να χρησιμοποιηθεί για την αναγνώριση αποπειρών εισβολής
  - Κάποιος πρέπει να το κοιτάζει…
  - Πρέπει το αρχείο να προστατεύεται!
- Αρχή ελάχιστων δικαιωμάτων: κάθε τμήμα του συστήματος προσπελαύνει την ελάχιστη δυνατή πληροφορία
  - π.χ. το σύστημα αρχείων δε μπορεί να αγγίξει τους πίνακες σελίδων κλπ.
  - Δύσκολο
- Αποδείξεις ορθότητας
  - Πολύ δύσκολες.
  - Αποδεικνύουν συμμόρφωση με προδιαγραφές, όχι ορθότητα προδιαγραφών