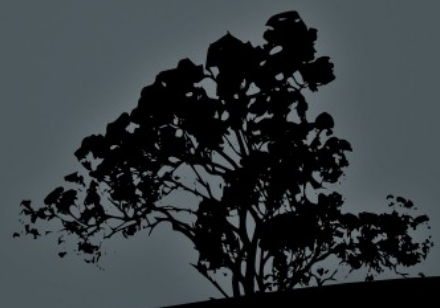
Εξέταση 102 – Μάθημα 18

110.1 Εκτέλεση εργασιών διαχείρισης ασφάλειας



Τα πεδία SUID/SGID

- Τα πεδία **SUID** και **SGID** χρησιμοποιούνται για να δίνεται η δυνατότητα σε κοινούς χρήστες να εκτελούν μια εντολή με τα δικαιώματα του κύριου χρήστη ή ομάδας.
- **-rwsr-xr-x 1 root root 90640 2011-08-09 19:16 /bin/mount** # Το εκτελέσιμο αρχείο **/bin/mount** έχει δικαιώματα εκτέλεσης ως χρήστης **root** από όλους τους χρήστες του συστήματος
- **-rwxr-sr-x 1 root shadow 50760 2011-06-24 12:28 /usr/bin/chage** # Το εκτελέσιμο αρχείο **/usr/bin/chage** έχει δικαιώματα εκτέλεσης ως ομάδα **shadow** από όλους τους χρήστες
- **-rwsr-sr-x 1 daemon daemon 47848 2011-05-16 13:32 /usr/bin/at** # Το εκτελέσιμο αρχείο **/usr/bin/at** έχει δικαιώματα εκτέλεσης ως χρήστης και ομάδα daemon από όλους τους χρήστες

Προβλήματα ασφαλείας με SUID/SGID

- Η παρουσία **SUID/SGID** σε εκτελέσιμα αρχεία μπορεί να είναι βολική αλλά περικλείει σοβαρότατους κινδύνους αν η εντολή έχει κάποιο σχεδιαστικό σφάλμα (πχ buffer overflow).
- Μια εντολή με σχεδιαστικό σφάλμα δίνει την δυνατότητα σε κοινούς χρήστες να την χρησιμοποιήσουν με τρόπο διαφορετικό από αυτό που σχεδιάστηκε. Φανταστείτε πχ η εντολή **mount** να μπορεί να καλέσει το κέλυφος **bash**! Αυτό θα σήμαινε ότι το bash θα έχει τα ίδια δικαιώματα με το SUID χρήστη της mount δηλαδή τον **root**!
- Τα πράγματα είναι ακόμη χειρότερα αν ο κύριος χρήστης ή η ομάδα είναι ο **root**
- Για τους πιο πάνω λόγους θα πρέπει να ελέγχουμε το σύστημα για SUID/SGID και να αποφεύγεται η ανάθεση του σε εντολές που καλούν το κέλυφος πχ vi, emacs κτλ

Εύρεση και αφαίρεση SUID και SGID

- **# find / -perm -4000 -type f** # εύρεση όλων των κανονικών αρχείων με πεδίο **SUID** σε ολόκληρο το σύστημα
- **# find / -perm -2000 -type f** # εύρεση όλων των κανονικών αρχείων με πεδίο **SGID** σε ολόκληρο το σύστημα
- **# find / -perm /6000 -type f -exec ls -l {} \;** # εύρεση όλων των κανονικών αρχείων με πεδίο **SUID** ή **SGID** και εκτέλεση της εντολής **ls -l** σε αυτά
- **# chmod u-s /bin/ping** # αφαίρεση **SUID** από **/bin/ping**
- **# chmod g-s /usr/bin/crontab** # αφαίρεση **SGID** από **/usr/bin/crontab**
- **# chmod -s /usr/bin/at** # αφαίρεση **SUID** και **SGID** από **/usr/bin/at**

Το αρχείο /etc/shadow

- Το αρχείο **/etc/shadow** περιέχει τους κωδικούς των χρηστών αλλά και χρήσιμες πληροφορίες για την παλαιότητα των κωδικών. Τα πεδία στο **/etc/shadow** έχουν τους πιο κάτω ρόλους:
- **user:6UwkipSFw$Jp3JxkKjZJ48zdM:15428:5:20:7:15:15695:**

1. Όνομα Χρήστη
2. Κρυπτογραφημένο συνθηματικό (**!** ή **τίποτα**: απουσία συνθηματικού, *: απενεργοποιημένος λογαριασμός, **!<hash>**: κλειδωμένος λογαριασμός. **!!**: δεν ορίστηκε συνθηματικό)
3. Ημερομηνία τελευταίας αλλαγής
4. Ελάχιστος αριθμός ημερών όπου ο χρήστης επιτρέπεται να αλλάξει το συνθηματικό του (0 = μπορεί να τον αλλάξει οποτεδήποτε).
5. Μέγιστος αριθμός ημερών όπου ο χρήστης μπορεί να κρατήσει το ίδιο συνθηματικό (99999 = δεν απαιτείται αλλαγή)
6. Αριθμός ημερών πριν την λήξη όπου θα παίρνει προειδοποιήσεις
7. Αριθμός ημερών μετά από τις οποίες ο λογαριασμός απενεργοποιείται
8. Ημερομηνία λήξης (αριθμός ημερών μετά την 01/01/1970)

Διαχείριση πληροφοριών παλαιότητας με *chage*

- **# chage -l user** # προβολή πληροφοριών παλαιότητας για **user**
- **# chage -E 2012-12-21 user** # καθορισμός ημερομηνίας λήξης
- **# chage -I 15 user** # αριθμός ημερών απενεργοποίησης
- **# chage -m 5 user** # ελάχιστος αριθμός ημερών όπου επιτρέπεται η αλλαγή συνθηματικού
- **# chage -M 20 user** # μέγιστος αριθμός ημερών όπου επιτρέπεται η κατακράτηση του ίδιου συνθηματικού
- **# chage -W 6 user** # αριθμός ημερών για προειδοποιήσεις
- **# chage -d 2012-03-25 user** # καθορισμός ημερομηνίας τελευταίας αλλαγής

Χρήση της date για προβολή ημερομηνιών αλλαγής/λήξης

- **user:6UwkipSFw$Jp3JxkKjZJ48zdM:15428:5:20:7:15:15695:**
- **# date -d "1970/01/01 +15428 days"**
  Thu Mar 29 00:00:00 EEST 2012 # ημερομηνία τελευταίας αλλαγής
- **# date -d "1970/01/01 +15695 days"**
  Fri Dec 21 00:00:00 EET 2012 # ημερομηνία λήξης λογαριασμού

Αυτές οι τιμές μπορούν να χρησιμοποιηθούν στην **chage** για καθορισμών των ημερών μετά την 01/01/1970 (unix epoch)

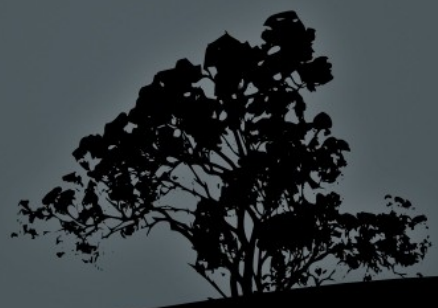
- **# chage -d 15428 user # = chage -E 2012-03-29 user**
- **# chage -E 15695 user # = chage -E 2012-12-21 user**

Χρήση της passwd για διαχείριση πληροφοριών παλαιότητας

- **# passwd -i 15 user** # αριθμός ημερών απενεργοποίησης
- **# passwd -n 5 user** # ελάχιστος αριθμός ημερών όπου επιτρέπεται η αλλαγή συνθηματικού
- **# passwd -x 20 user** # μέγιστος αριθμός ημερών όπου επιτρέπεται η κατακράτηση του ίδιου συνθηματικού
- **# passwd -w 6 user** # αριθμός ημερών για προειδοποιήσεις
- **# passwd -e user** # άμεση λήξη λογαριασμού και υποχρεωτική προτροπή για αλλαγή συνθηματικού
- **# passwd -S user** # προβολή κατάστασης χρήστη user
- **# passwd -Sa** # προβολή κατάστασης όλων των χρηστών

Χρήση της usermod για διαχείριση πληροφοριών παλαιότητας

- **# usermod -e 2012-12-21 user** # καθορισμός ημερομηνίας λήξης
- **# usermod -f 15 user** # αριθμός ημερών απενεργοποίησης
- **# usermod -L user** # κλείδωμα λογαριασμού
- **# usermod -U user** # ξεκλείδωμα λογαριασμού



Εντοπισμός ανοικτών θυρών στο σύστημα

- Οι ανοικτές θύρες (ports) σε ένα σύστημα είναι χρήσιμες για να έχουμε πρόσβαση στις υπηρεσίες που βρίσκονται πίσω από αυτές
- Πολλές φορές όμως διάφορα συστήματα και διανομές έχουν προεγκατεστημένες δικτυακές υπηρεσίες οι οποίες ίσως να μην χρειάζονται
- Είναι καλή τακτική να απενεργοποιούνται οι αχρείαστες υπηρεσίες για εξοικονόμηση πόρων από το σύστημα αλλά κυρίως για να μειωθούν οι πιθανότητες κακόβουλων χρηστών να εκμεταλλευτούν πιθανές τους αδυναμίες για να εισχωρήσουν στο σύστημα
- Για να δούμε ποιες είναι αυτές οι ανοικτές θύρες χρησιμοποιούμε τα εργαλεία **netstat**, **lsof** και **nmap**

Εντοπισμός ανοικτών θυρών με netstat

- Η εντολή **netstat** μεταξύ άλλων προβάλλει τις ανοικτές θύρες στο σύστημα

- **# netstat -lnptu** # προβολή ανοικτών θυρών (tcp/udp) και εφαρμογών σε αριθμητική μορφή

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1154/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1333/cupsd
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      1645/postgres
tcp        0      0 0.0.0.0:21405           0.0.0.0:*               LISTEN      3184/skype
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      1624/mysqld
...
udp        0      0 127.0.0.1:46773         0.0.0.0:*                           3184/skype
udp        0      0 0.0.0.0:38894           0.0.0.0:*                           1177/avahi-daemon:
udp        0      0 0.0.0.0:68              0.0.0.0:*                           5209/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*                           5186/dhclient
udp        0      0 10.100.1.66:123         0.0.0.0:*                           5339/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           5339/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           5339/ntpd
```

Εντοπισμός ανοικτών θυρών με *lsof*

- Η εντολή **lsof** χρησιμοποιείται για την προβολή των ανοικτών αρχείων στο σύστημα. Αρχεία θεωρούνται και οι υποδοχείς (sockets) και άρα και οι θύρες
- **# lsof** # προβολή όλων των ανοικτών αρχείων στο σύστημα
- **# lsof -i** # προβολή όλων των συνδέσεων και θυρών TCP/IP
- **# lsof -iTCP -s:LISTEN -P** # προβολή ανοικτών θυρών TCP σε αριθμητική προβολή (-P)
- **# lsof -iUDP | grep -v "\->"** # προβολή ανοικτών θυρών UDP
- **# lsof -p 6543** # προβολή των ανοικτών αρχείων της διεργασίας 6543
- **# lsof -c apache2** # προβολή των ανοικτών αρχείων της εντολής apache2
- **# lsof -u user** # προβολή των ανοικτών αρχείων του χρήστη user
- **# lsof /mnt** # προβολή των διεργασιών που χρησιμοποιούν το /mnt

Εντοπισμός ανοικτών θυρών με *nmap*

- Αντίθετα από τις εντολές **netstat** και **lsof** η **nmap** μας ενημερώνει για ανοικτές θύρες σε άλλους υπολογιστές. Σε κάποιες χώρες είναι παράνομο να χρησιμοποιείται σε συστήματα που δεν σας ανήκουν.
- **$ nmap -sT www.network.dom** # (TCP Connect scan) προκαθορισμένος τύπος scan για μη προνομιούχους χρήστες
- **# nmap -sS www.network.dom** # (TCP Syn scan) προκαθορισμένος τύπος scan για προνομιούχους χρήστες (γρηγορότερο)
- **# nmap -p 65-87,100 www.network.dom** # έλεγχος θυρών 65 μέχρι 87 και 100 (TCP)
- **# nmap -p 1-65535 -O www.network.dom** # έλεγχος όλων των θυρών και ανίχνευση λειτουργικού συστήματος
- **# nmap -sU -n www.network.dom** # UDP scan και αριθμητική προβολή
- **# nmap -sP 10.0.0.0/24** # ping sweep για ανίχνευση ανοικτών κόμβων
- **# nmap -sV 10.0.0.3** # ανίχνευση υπηρεσιών και εκδόσεων πίσω από θύρες

Αλλαγή χρήστη με την εντολή *su*

- Η εντολή **su** χρησιμοποιείται για είσοδο στο σύστημα σαν άλλος χρήστης **χρησιμοποιώντας το συνθηματικό του νέου χρήστη**. Αν δεν ορίσουμε όνομα χρήστη εννοείται ο **root**.
- **$ su** # είσοδος στο σύστημα σαν χρήστης **root** κληρονομώντας στοιχεία από το περιβάλλον του προηγούμενου χρήστη
- **$ su -** # είσοδος στο σύστημα σαν χρήστης **root.** Το περιβάλλον να είναι το ίδιο όπως όταν κάνουμε login (μετάβαση στον προσωπικό κατάλογο, εκτέλεση **.bash_profile** ή **.profile** κτλ)
- **$ su user** # είσοδος στο σύστημα σαν χρήστης **user**
- **$ su - user** # είσοδος στο σύστημα σαν χρήστης **user** με περιβάλλον όπως στο **login**
- **# su - user** # ο χρήστης **root** μπορεί να αναλάβει τον ρόλο οποιουδήποτε χρήστη του συστήματος χωρίς εισαγωγή συνθηματικού!
- **$ su -c "find /etc"** # εκτέλεση εντολής **find** με προνόμια **root**

Εκτέλεση εντολών σαν άλλος χρήστης με *sudo*

- Η εντολή **sudo** χρησιμοποιείται για την εκτέλεση εντολών σαν άλλος χρήστης **χρησιμοποιώντας το συνθηματικό του αρχικού χρήστη**. Για να γίνει αυτό θα πρέπει ο χρήστη που έχει δικαίωμα εκτέλεσης εντολών sudo να έχει δηλωθεί στο αρχείο **/etc/sudoers** ή να ανήκει σε μια ομάδα που βρίσκεται σε αυτό το αρχείο
- **$ sudo /etc/init.d/ssh restart** # εκτέλεση εντολής σαν **root**
- **$ sudo -uotheruser mail** # εκτέλεση εντολής σαν **otheruser**
- **$ sudo -i** # εισαγωγή σε κέλυφος bash με προνόμια χρήστη **root**
- **$ sudo -b updatedb** # εκτέλεση εντολής στο παρασκήνιο σαν **root**

http://xkcd.com/149/

Ρύθμιση sudo με */etc/sudoers*

- Στο αρχείο **/etc/sudoers** δηλώνονται όσοι χρήστες ή ομάδες έχουν δικαιώματα χρήσης της **sudo**. Έχει άδειες χρήσης για ανάγνωση μόνο και δεν συστήνεται να το ανοίγουμε με οποιονδήποτε άλλο κειμενογράφο παρά μόνο με τον δικό του, το **visudo**
- **# visudo** # άνοιγμα **/etc/sudoers** για επεξεργασία
- **user ALL=(ALL) ALL** # δικαίωμα στο χρήστη **user** να εκτελεί σε όλα τα συστήματα, σαν οτιδήποτε χρήστης, οποιανδήποτε εντολή
- **user hostname = (operator) /bin/ls, /bin/kill, /usr/bin/lprm** # δικαίωμα στο χρήστη **user** να εκτελεί στο σύστημα **hostname**, σαν χρήστης **operator**, τις εντολές **ls**, **kill** και **lprm**

Ρύθμιση sudo με */etc/sudoers*

- **user hostname = (operator : operator) /bin/ls, /bin/kill**

  # δικαίωμα στο χρήστη **user** να εκτελεί στο σύστημα **hostname**, σαν χρήστης και ομάδα **operator**, τις εντολές **ls** και **kill**

- **user hostname = (operator) /bin/ls, (root) /bin/kill**

  # δικαίωμα στο χρήστη **user** να εκτελεί σαν χρήστης **operator** την εντολή **ls** και σαν **root** την εντολή **kill**

- **user ALL = NOPASSWD: /bin/kill, PASSWD: /bin/ls**

  # δικαίωμα στο χρήστη **user** να εκτελεί σαν χρήστης **root** την εντολή **kill** χωρίς συνθηματικό και την εντολή **ls** με συνθηματικό

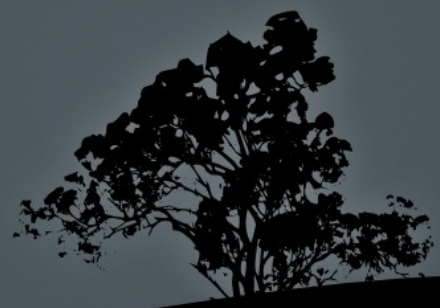
- **%admin ALL=(ALL) ALL** # παραχώρηση όλων των δικαιωμάτων στους χρήστες της ομάδας **admin**

Καθορισμός περιορισμών σε */etc/security/limits.conf*

- Το αρχείο **/etc/security/limits.conf** δηλώνονται τα όρια για διάφορους πόρους του συστήματος
- Η δομή του είναι:
  <domain> <type> <item> <value>
- **domain**: όνομα χρήστη (**user**), ομάδας (**@group**) ή προκαθορισμένη ρύθμιση (*)
- **type**: **soft** (ελαστικό όριο), **hard** (αυστηρό όριο), **–** (και τα δύο). Το ελαστικό όριο μπορεί να ξεπεραστεί από τους χρήστες με την εντολή **ulimit** ενώ το αυστηρό όχι
- **item**: καθορισμός πόρου για περιορισμό πχ, **maxlogins**, **nproc**, **cpu**, **rss**, κτλ
- **value**: η τιμή του ορίου. Μπορεί να είναι σε kB για πόρους δεδομένων ή χρόνος για πόρους χρόνων ή ακόμη και απλά αριθμός αρχείων, διεργασιών κτλ

Καθορισμός περιορισμών σε */etc/security/limits.conf*

- Παραδείγματα items:
  - **maxlogins**: μέγιστος αριθμός συνεδριών
  - **nproc**: αριθμός διεργασιών
  - **rss**: (resisent set size) μέγεθος παραμένουσας μνήμης
  - **stack**: μέγεθος μνήμης σωρού (stack)
  - **memlock**: μέγεθος κλειδωμένης μνήμης
  - **as**: μέγεθος χώρου μνήμης
  - **cpu**: χρόνος χρήσης επεξεργαστή
  - **fsize**: μέγεθος αρχείων
  - **nofiles**: αριθμός αχρείων



Καθορισμός περιορισμών σε */etc/security/limits.conf*

- Παράδειγμα ρυθμίσεων **/etc/security/limit.conf**

```
<domain>        <type>  <item>          <value>

*               hard    rss             10000
@student        hard    nproc           20
@faculty        soft    nproc           20
@faculty        hard    nproc           50
ftp             hard    nproc           0
@student        -       maxlogins       4
```

- Το προκαθορισμένο αυστηρό όριο για **rss** είναι **10000 kB**
- Το αυστηρό όριο **nproc** για ομάδα **student** είναι 20 διεργασίες
- Το ελαστικό και αυστηρό όριο **nproc** για ομάδα **faculty** είναι 20 και 50 διεργασίες αντίστοιχα
- Ο χρήστης **ftp** δεν δικαιούται να εκτελεί διεργασίες
- Η ομάδα **student** δικαιούται μόνο 4 ταυτόχρονες συνεδρίες

Καθορισμός περιορισμών χρηστών με *ulimit*

- Η εντολή **ulimit** χρησιμοποιείται για τον προσωρινό καθορισμό των ορίων του κελύφους όπου δουλεύουμε, και των θυγατρικών διεργασιών, αλλά και για την αλλαγή των ορίων
- Μόνο ο χρήστης **root** μπορεί να καθορίζει τα όρια και μόνο αυτός μπορεί να αλλάξει το αυστηρό (hard) όριο του
- Οι κοινοί χρήστες μπορούν μόνο να επανακαθορίσουν το δικό τους ελαστικό (soft) όριο

$ uname -a # = **uname -Sa**. Προβολή μαλακών ορίων
max locked memory (kbytes, -l) 64 # κλειδωμένη μνήμη
max memory size (kbytes, -m) unlimited # παραμένουσα μνήμη
stack size (kbytes, -s) 8192 # μνήμη σωρού (stack)
cpu time (seconds, -t) unlimited # χρόνος επεξεργαστή
max user processes (-u) 30966 # αριθμός διεργασιών
virtual memory (kbytes, -v) unlimited # εικονική μνήμη

Καθορισμός περιορισμών χρηστών με *ulimit*

- **$ ulimit -Ha** # προβολή αυστηρών ορίων χρήστη
- **$ ulimit -u 45000** # αύξηση ορίου διεργασιών (soft) σε 45000
- **$ ulimit -m 1024000000** # αύξηση παραμένουσας μνήμης (rss) (soft) σε 1GB
- **# ulimit -Hs 16384** # καθορισμός σκληρού ορίου σωρού (stack) σε 16MB
- **$ ulimit -St 2** # αύξηση ελαστικού ορίου χρόνο επεξεργαστή σε 2 λεπτά
- **# ulimit -v 2048000000** # αύξηση εικονικής μνήμης (soft και hard) σε 2GB
- **$ ulimit -l 128** # αύξηση ελαστικού ορίου κλειδωμένης μνήμης σε 128kB