Password encoding in UNIX systems

Νίκος Μαυρογιαννόπουλος

13 Σεπτεμβρίου 2000

1 Password encoding in UNIX systems

1.1 Τι είναι το Password encoding;

Τα UNIX συστήματα μιας και είναι πολυχρηστικά, χρειάζονται κάποιο τρόπο για να ξεχωρίζουν τους χρήστες. Ο κάθε χρήστης διαθέτει ένα μοναδικό αριθμό (user id), το οποίο αντιστοιχεί σε κάποιο όνομα χρήστη (username). Οι παραπάνω αντιστοιχείες καθορίζονται στο αρχείο "/etc/passwd", ονομασμένο έτσι για ιστορικούς λόγους. Μια καταχώριση στο αρχείο αυτό θα μπορούσε να είναι:

nmav:x:500:4:Nikos Mavroyanopoulos:/home/nmav:/bin/bash

Τα πεδία σε αυτό το αρχείο ξεχωρίζουν απο την ανω-κάτω τελεία ':', ενώ τα πεδία που μας ενδιαφέρουν είναι το 1 που περιέχει το όνομα χρήστη **- nmav -** καθώς και το πεδίο 3 όπου αναφέρεται η ταυτότητα του χρήστη **- 500**.

Τα περισσότερα UNIX σύστηματα, για να πιστοποιήσουν την ταυτότητα του χρήστη χρησιμοποιούν συνθηματικές λέξεις, ή απλώς συνθηματικά (passwords). Αυτά τα συνθηματικά αλλάζουν με την εκτέλεση του "/usr/bin/passwd" προγράμματος, το οποίο ανανεώνει τη βάση συνθηματικών του συστήματος. Σε απλά συστήματα αυτή η βάση είναι ένα εκτυπώσιμο αρχείο, και στα περισσότερα παλαιά συστήματα αυτό ήταν το ίδιο το "/etc/passwd". Στα πιο νέα συστήματα είναι ένα αρχείο αναγνώσιμο μόνο απο τον διαχειριστή του συστήματος, και συνήθως είναι το "/etc/shadow".

Αντίθετα με αυτό που θα περίμενε κανείς, η βάση συνθηματικών του συστήματος δεν περιέχει τα ίδια τα συνθηματικά. Τα συνθηματικά κωδικοποιούνται μέσω ενός αλγορίθμου μίας φοράς[1] και αποθηκεύονται. Με αυτόν τον τρόπο δεν μπορεί ούτε ο διαχειριστής του συστήματος να δει τα συνθηματικά των χρηστών (μπορεί όμως να τα αλλάξει). Μιας και τα συνθηματικά δεν υπάρχουν στην βάση ίσως να αναρωτηθήτε πως γίνεται η πιστοποίηση του χρήστη. Αν υποθέσουμε ότι ο αλγόριθμος μίας φοράς είναι $PW(x)$, με x να είναι το συνθηματικό, τότε κατά την

[1] Έστω $y = A(x)$. Σε ένα αλγόριθμο μιας φοράς(A) αν γνωρίζουμε το y, είναι δύσκολο να υπολογίσουμε το x.

εισαγωγή του συνθηματικού(x) το σύστημα αρκεί να υπολογίσει πάλι το $PW(x)$ και να το συγκρίνει με το αποθηκευμένο στην βάση.

1.2 DES Password Encoding

Τα πρώτα UNIX συστήματα κωδικοποιούσαν τα συνθηματικά χρησιμοποιώντας ένα τροποποιημένο DES αλγόριθμο, με τετοιο τρόπο έτσι ώστε να μην μπορεί απο την κωδικοποιημενη μορφή να προκύψει το αρχίκο συνθηματικό. Ο μόνος τρόπος για να προκύψει το αρχικό συνθηματικό είναι η δοκιμή όλων των δυνατών συνδιασμών συνθηματικών (η διαδικασία αυτή συχνά καλείται brute force attack). Αυτό το είδος επίθεσης, ενώ αρχικά ήταν πρακτικά αδύνατο να εφαρμοστεί, τα τελευταία χρόνια, με την πρόοδο των υπολογιστών, μπορεί να εφαρμοστεί χωρίς ιδιαίτερα ακριβό εξοπλισμό.

Ενα τυπικό UNIX έχει καταχώρισεις στο "/etc/shadow" της μορφής:

nmav:ZKZ/wHem5Uv:11000:0:99999:7:::

Το δευτερο πεδίο περιέχει μια εκτυπώσιμη μορφή του DES encoded συνθηματικού. Τα δύο πρώτα του ψηφία είναι η εκτυπώσιμη μορφή ενός τυχαίου salt που αποτελείται απο 12 bits[2]. Χρησιμοποιείται ώστε να διαφοροποιεί την έξοδο του αλγορίθμου, ακόμα και σε ίδια συνθηματικά. Ενα συνθηματικό μπορεί να αποθηκευτεί με 2^{12} τρόπους το πολύ.

Ο αλγόριθμος DES (Data Encryption Standard) είναι κρυπτογραφικός αλγόριθμος που λειτουργεί με 64 bit μπλοκ. Ήταν πρότυπο τις δεκαετίες του '70 και '80, αλλά είναι ξεπερασμένος σήμερα. Ο τρόπος με το οποίο χρησιμοποιείται είναι:

- εκκίνηση του DES
- Μετατροπή του Expansion Permutation συναρτήσει του salt
- Κρυπτογράφηση για 25 φορές 8 null bytes χρησιμοποιώντας σαν κλειδί το συνθηματικό του χρήστη
- Το τελικό αποτέλεσμα είναι η εκτυπώσιμη μορφή των κρυπτογραφημένων 64 bit μαζί με τα 12 bit του salt.

Ο παραπάνω αλγόριθμος είναι ουσιαστικά αλγόριθμος μιας φοράς, αφού δεν υπάρχει τρόπος αντιστροφής της διαδικασίας και εξαγωγής του συνθηματικού απο την κωδικοποιημένη μορφή, χωρίς να δοκιμάσουμε όλους τους δυνατούς συνδυασμούς. Έχει το λιγότερο, την ασφάλεια του DES, αλγόριθμου δοκιμασμένου για αρκετά χρόνια. Ένας περιορισμός που έχει είναι ότι τα συνθηματικά περιορίζονται στους 8 χαρακτήρες από τους οποίους χρησιμοποιούνται τα 56 bits (λόγω του μικρού ευρους κλειδιών του DES).

[2] 8 bit == 1 byte

Μιας και τα E-boxes του DES είναι σταθερά ανά salt, είναι δυνατό με κάποιο δυνατό υπολογιστή να δοκιμάσουμε όλα τα πιθανά συνθηματικά σε λογικό χρόνο. Ο DES όπως είδαμε χρησιμοποιεί μόνο 56 bits για κλειδί οπότε όλοι οι πιθανοί συνδιασμοί που πρέπει να δοκιμάσει κανείς είναι 2^{56}, αριθμός οχι υπερβολικά μεγαλός.

Έτσι ενώ αρχικά τα κωδικοποιημένα συνθηματικά αποθηκεύονταν στο "/etc/passwd", αρχείο αναγνώσιμο απο όλους τους χρήστες του συστήματος, τα τελευταία χρόνια μεταφέρθηκαν σε ειδικό αρχείο αναγνώσιμο απο τον διαχειριστή μόνο του συστήματος. Αυτό η κίνηση φανέρωσε την αδυναμία του αλγορίθμου κωδικοποίησης να προστατέψει αποτελεσματικά τα συνθηματικά των χρηστών.

1.3 MD5 crypt

Λύση στο παραπάνω πρόβλημα και στο ότι ο DES είναι κρυπτογραφικός αλγόριθμος και υπήρχε παλαιότερα δυσκολία εξαγωγής του από τις Η.Π.Α, κυρίως για τα μη εμπορικά UNIX λειτουργικά, έδωσε ο Paul-Henning Kamp για το FreeBSD λειτουργικό. Σχεδιάσε ένα αλγόριθμο κωδικοποίησης συνθηματικών βασισμένο στον MD5 [3] hash αλγόριθμο. Ο MD5 είναι απο τον σχεδιασμό του μίας φοράς (one-way), οπότε οι μετατροπές σε αυτόν ήταν μικρές, ενώ προστέθηκε salt απο 12 μέχρι 48 bits.

Ο αλγορίθμος αυτός σε γενικές γραμμές κωδικοποιεί το salt και το συνθηματικό, με τον MD5, με αρκετούς διαφορετικούς τρόπους, έτσι ώστε να καθυστερήσει την διαδικασία. Ο MD5 αλγόριθμος παράγει ένα 128 bit hash και αφού μετατραπεί σε εκτυπώσιμη μορφή μαζί με το salt αποθηκεύεται στο αρχείο συνθηματικών. Ενα τυπικό σύστημα με MD5 κωδικοποιημένα συνθηματικά θα έχει καταχωρίσεις στο /etc/shadow της μορφής:

nmav:$1$4Wcrq7pj$l8uWovJXI1QBP6MXRrWdt0:11000:0:99999:7:::

Οι χαρακτήρες '1' καθορίζουν τον συγκεκριμένο αλγόριθμο που χρησιμοποιείται (versioning) και μιας και η DES κωδικοποίηση δεν χρησιμοποιεί τον χαρακτήρα '$', δεν είναι πρόβλημα για ένα πρόγραμμα να τα διακρίνει. Το salt ξεχωρίζει απο την έξοδο του MD5 με τον χαρακτήρα '$'.

Ο MD5 δεν έθετε πρακτικούς περιορισμούς στο μέγεθος του συνθηματικού του χρήστη. Είχε όμως το πρόβλημα του DES encoding... έχει συγκεκριμένη πολυπλοκότητα και κάποια στιγμή στο μέλλον θα ήταν εφικτή η brute force attack και σε αυτόν, άρα θα έπρεπε να αντικατασταθεί και πάλι.

[3] Ο MD5 (ή Message Digest 5) είναι Αλγόριθμος που παράγει ένα 128 bits χαρακτηριστικό αριθμό για κάθε μηνύμα. Σχεδιαστηκε απο τον Ron Rivest και περιγράφεται στο RFC1321

1.4 Blowfish crypt (bcrypt)

Την λύση σε αυτή την αδυναμία δίνει ένας αλγόριθμος σχεδιασμένος απο τον Niels Provos και τον David Mazieres για το OpenBSD σύστημα. Ο αλγόριθμος αυτός είναι βασισμένος στον blowfish[4] αλγόριθμο, με εκθετικά μεταβαλλόμενη πολυπλοκότητα $O(2^n)$, και το n καθορίζεται απο τον διαχειριστή του συστήματος. Ο τροποποιημένος αλγόριθμος καλείται eksblowfish, ή πιο αναλυτικά Expensive Key Schedule Blowfish.

Η επιλογή του blowfish έγινε λόγω του ότι διαθέτει s-boxes εξαρτώμενα από το κλειδί, και συνεπώς έχει πολύ μεγαλύτερη πολυπλοκότητα, κατά την εκκίνηση του, από τον DES αλγόριθμο. Επιπρόσθετα έχει πολύ μεγαλύτερο εύρος κλειδιών (448 bits), κάτι που δεν θέτει πρακτικούς περιορισμούς στο μέγεθος των συνθηματικών (55 bytes).

Ο eksblowfish χρησιμοποιεί μια τροποποιημένη μορφή της δημιουργίας των s-boxes του αλγορίθμου, και ένα 128 bits salt. Σε γενικές γραμμές η λειτουργία του αλγορίθμου συνοψίζεται στο:

- Εκκίνηση του Blowfish (δημιουργία των s-boxes)
- Από 1 μέχρι 2^n τροποποίηση των s-boxes και των sub-keys συναρτήσει του salt και συναρτήσει του κλειδιού
- Κρυπτογράφιση 64 φορές, μιας αλυσίδας χαρακτήρων (192 bits) με το κλειδί

Το κωδικοποιημένο string είναι η εκτυπώσιμη μορφή του salt και της εξόδου του αλγορίθμου. Ενα σύστημα με Bcrypt κωδικοποιημένα συνθηματικά θα έχει καταχωρίσεις στο /etc/shadow της μορφής:

nmav:$2a$05$abcdefghijklmnopqrstuu5s2v8.iXieOjg/.AySBTTZIIVFJeBui:11000:0:99999:7:::

Οπως παρατηρούμε και αυτός ο αλγόριθμος χρησιμοποιεί τους πρώτους χαρακτήρες ('$2a$'), του κωδικοποιημένου συνθηματικού για να δείξει την έκδοση του αλγορίθμου, ενώ πάλι ο χαρακτήρας '$' χρησιμοποιείται σαν σημείο διαχωρισμού πεδίων. Το πρώτο πεδίο όπως είδαμε περιέχει την έκδοση του αλγορίθμου, το δεύτερο το μέγεθος του κόστους(n), ενώ το τρίτο περιέχει το 128 bits salt και το κωδικοποιημένο συνθηματικό.

Μια πρόχειρη υλοποίηση του bcrypt, καθώς και ένα patch για το Linux pam_unix module, βρίσκεται στο 'http://members.hellug.gr/nmav/bcrypt'. Στο http://www.openbsd.org, μπορούν να βρεθούν περισσότερες πληροφορορίες για τον αλγόριθμο, καθώς και η αρχική υλοποίηση του.

[4] κρυπτογραφικός συμμετρικός αλγόριθμος σχεδιασμένος από τον Bruce Schneier

1.5 Cracking Passwords

Η αδυναμία του DES encoding, να προσαρμοστεί στον χρόνο και στην συνεχώς αυξανόμενη υπολογιστική δύναμη, προκάλεσε την δημιουργία προγραμμάτων που προσπαθούσαν απο την κωδικοποιημένη μορφη να παράξουν το συνθηματικό σε σχετικά μικρό χρόνο. Τα προγράμματα αυτά ονομάστηκαν password crackers, και βασίζοντουσαν στην χαλαρή επιλογή συνθηματικών απο τους χρήστες, αλλά καθώς το υλικό βελτιωνώταν, το «σπάσιμο» ενός όχι πολυ σύνθετου συνθηματικού έγινε εφικτό. Αρχικά βασίζονταν κυρίως σε κάποιο λεξικό που κρατάει συνθηματικά που επιλέγαν πιο συχνά οι χρήστες (dictionary attack). Πιο τελευταία προγράμματα χρησιμοποιούν αρκετά πιο εξελιγμένες τεχνικές.

Μερικά προγράμματα είναι:

- John the Ripper του "Solar Designer", http://www.openwall.com/john
- QCrack του "Crypt Keeper", ftp://chaos.infospace.com/pub/qcrack/
- Crack του Alec Muffett, http://www.users.dircon.co.uk/~ crypto/

Μια ενδιαφέρουσα ερευνα για τις cracking εφαρμογές και την αποδοσή τους έγινε από τον Kurt Hockenbury και μπορεί να βρεθεί στο 'http://attila.stevens-tech.edu/~ khockenb/crypt3.html'. Στο κείμενο αυτό παρουσιάζεται η αναποτελεσματικότητα του DES σε συνθηματικά με λιγότερους απο 6 χαρακτήρες και σε ορισμένες μορφές συνθηματικών με 8 χαρακτήρες, ενώ το hardware που χρησιμοποιήθηκε είναι εύκολα προσβάσιμο απο οποιονδήποτε φοιτητή στο πανεπιστήμιο του.

1.6 Συμπεράσματα και γεγονότα

Γενικά οι περισσότεροι διανομείς και εταιρίες παραγωγής UNIX χρησιμοποιούν ακόμα την DES κωδικοποίηση, αν και ξεπερασμένη, για λόγους συμβατότητας. Αυτοί οι λόγοι συνήθως προκύπτουν όταν διαφορετικά συστήματα διαμοιράζονται αρχεία συνθηματικών πχ. με NIS ή NIS+. Ακόμα χειρότερα, μερικές εταιρίες ακόμη χρησιμοποιούν την παλιά μορφή του "/etc/passwd", που περιέχει το κωδικοποιημένο συνθηματικό.

Η 'bcrypt' κωδικοποίηση δεν χρησιμοποιείται, μέχρι σήμερα, σε άλλα λειτουργικά πέρα απο το OpenBSD, κυρίως επειδή χρησιμοποιεί κρυπτογραφικό αλγόριθμο με μεγάλο εύρος κλειδιών, οπότε και η εξαγωγή απο τις Η.Π.Α (στις οποίες έχουν την έδρα τους οι περισσότερες επιχειρήσεις) δεν ήταν μέχρι πρόσφατα, νόμιμη.

Ορισμένα συστήματα, όπως το QNX[5], επέλεξαν διαφορετικούς τρόπους αποθήκευσης των συνθηματικών, αλλά απέτυχαν στον σχεδιασμό των αλγορίθμων τους. Περισσότερες πληροφορίες για την συγκεκριμένη αδυναμία του αλγορίθμου θα βρείτε στο 'http://www.i-opener-linux.net/decrypt/'.

[5] Real-time λειτουργικό σύστημα

Βέβαια δεν αρκεί ένας καλός αλγόριθμος κωδικοποίησης για να είναι ασφαλής ένας λογαριασμός. Ο χρήστης έχει σημαντικό ρόλο στην ασφάλεια του λογαριασμού του και αυτό γιατί η επιλογή του συνθηματικού ανήκει σε αυτόν. Οσο μεγάλο και να είναι το salt, όσο καλός και να είναι ο αλγόριθμος, αν το συνθηματικό είναι ΄12345΄ είναι ζητημα δευτερολέπτων να βρεθεί.

Για την αποφυγή ευκολων συνθηματικών από τους χρήστες η shadow μορφή των συνθηματικών, καθώς και η πιστοποίηση με PAM (Plugable Authentication Modules), επιτρέπουν στον διαχειριστή του συστήματος να θέσει μέγιστο χρόνο ζωής ενός συνθηματικού, επιτρεπόμενο χρόνο αλλαγής ενός συνθηματικού, λήξη λογαριασμών κλπ, κάτι που επιβάλεται σε ένα σύστημα που προορίζεται για ασφαλές. Επίσης υπάρχει η CrackLib του Alec Muffett, που μπορεί να χρησιμοποιηθεί απο το πρόγραμμα αλλαγής συνθηματικών, για να απορρίπτει τα πολύ εύκολα από αυτά.