Passwords, Hashes, and Cracks, Oh My!

How Mac OS X Implements Password Authentication

Dave Dribin

“Why Should I Care?”

- Application Developer
- System Administrator
- End User

Authentication

- Authentication is the process of attempting to verify a user's identity
- Passwords authenticate using a "shared secret"

History

- Mac OS X based on NeXTSTEP
- NeXTSTEP Unix on top of Mach
- Unix developed by AT&T Bell Labs on DEC PDP-11
- Unix based on mainframe time-sharing systems

UNIX Time-Sharing System Version 1

- Released in 1971
- First release of Unix as we know it
- Plaintext passwords

NAME passwd ' -- password file

SYNOPSIS --

DESCRIPTION passwd contains for each user the following information:

name (login name)
password
numerical user ID
default working directory
program to use as Shell

This is an ASCII file. Each field within each user's entry is separated from the next by a colon. Each user is separated from the next by a new-line. If the password field is null, no password is demanded; if the Shell field is null, the Shell itself is used.

This file, naturally, is inaccessible to anyone but the super-user.

This file resides in directory /etc.

FILES --

SEE ALSO /etc/init

DIAGNOSTICS --

BUGS --

OWNER super-user

Plaintext Problems

"Perhaps the most memorable [example] occurred in the early 60's when a system administrator on the CTSS system at MIT was editing the password file and another system administrator was editing the daily message that is printed on everyone's terminal on login. Due to a software design error, the temporary editor files of the two users were interchanged and thus, for a time, the password file was printed on every terminal when it was logged in."

-- Robert Morris and Ken Thompson, April 3, 1978

Unix Versions 3, 4, 5, 6

- Released 1973 through 1975
- Encrypted Password
- Password file is readable by all

NAME passwd -- password file

DESCRIPTION passwd contains for each user the following information:

name (login name, contains no upper case)
encrypted password
numerical user ID
GCOS job number and box number
initial working directory
program to use as Shell

This is an ASCII file. Each field within each user's entry is separated from the next by a colon. The job and box numbers are separated by a comma. Each user is separated from the next by a new-line. If the password field is null, no password is demanded; if the Shell field is null, the Shell itself is used.

This file resides in directory /etc. Because of the encrypted passwords, it can and does have general read permission and can be used, for example, to map numerical user ID's to names.

SEE ALSO login(I), crypt(III), passwd(I)

NAME crypt -- password encoding

SYNOPSIS
```
mov $key,r0
jsr pc,crypt
```

DESCRIPTION On entry, r0 should point to a string of characters terminated by an ASCII NULL. The routine performs an operation on the key which is difficult to invert (i.e. encrypts it) and leaves the resulting eight bytes of ASCII alphanumerics in a global cell called "word".

Login uses this result as a password.

FILES kept in /lib/liba.a

SEE ALSO passwd(I),passwd(V), login(I)

DIAGNOSTICS there are none; garbage is accepted.

BUGS --

Encrypting vs. Hashing

- Encrypting is Two-Way
  - $E_K(M) = C$
  - $D_K(C) = M$
- Hashing is One-Way
  - $H(M) = H$

HAGELIN M-209 CIPHER MACHINE (GVG / PD)

World's First Hash?

"It turned out that the M-209 program was usable, but with a given key, the ciphers produced by this program are trivial to invert. It is a much more difficult matter to find out the key given the cleartext input and the enciphered output of the program. Therefore, the password was used not as the text to be encrypted but as the key, and a constant was encrypted using this key. The encrypted result was entered into the password file."

Methods of Attacking Hashed Passwords

- Exploiting Algorithm Weaknesses
- Brute Force
- Dictionary
- Rainbow Tables

Analysis of 3,289 Passwords

- 15 were a single ASCII character
- 72 were strings of two ASCII characters
- 464 were strings of three ASCII characters
- 477 were string of four alphanumerics
- 706 were five letters, all upper-case or all lower-case
- 605 were six letters, all lower-case

PDP-11/70 Brute Force

n	26 lower-case letters	36 lower-case letters and numbers	62 alphanumeric characters	95 printable characters	all 128 ASCII characters
1	30 msec	40 msec	80 msec	120 msec	160 msec
2	800 msec	2 sec	5 sec	11 sec	20 sec
3	22 sec	58 sec	5 min	17 min	43 min
4	10 min	35 min	5 hrs	28 hrs	93 hrs
5	4 hrs	21 hrs	318 hrs		
6	107 hrs				

AT&T Version 7 Unix

- Released in 1979
- Slower Hashing
- Less Predictable Passwords
- Salted Passwords

Data Encryption Standard

- 56-bit key block cipher
- Designed by IBM, published in 1975
- Designed to be slow
- Reviewed (and revised) by NSA
- Government Standard in 1977

DES crypt(3)

- The first 8 bytes of the key are null-padded, and the low-order 7 bits of each character is used to form the 56-bit DES key
- 12 bits of salt, encoded into 2 characters
- Example:
  - `foobar = jFUO4DTdeEmpw`
  - `foobar = cye1Y9RMJIk/2`

Problems with System 7

- Limits length of password to 8 characters
- Small salt
- /etc/passwd still world readable
- DES not exportable outside US

“The” Internet Worm

- Written by Robert T. Morris
- Unleashed in November 1988
- Designed for VAX and Sun-3 running BSD
  - 10% of Internet infected
- Included dictionary attack

Shadow Passwords

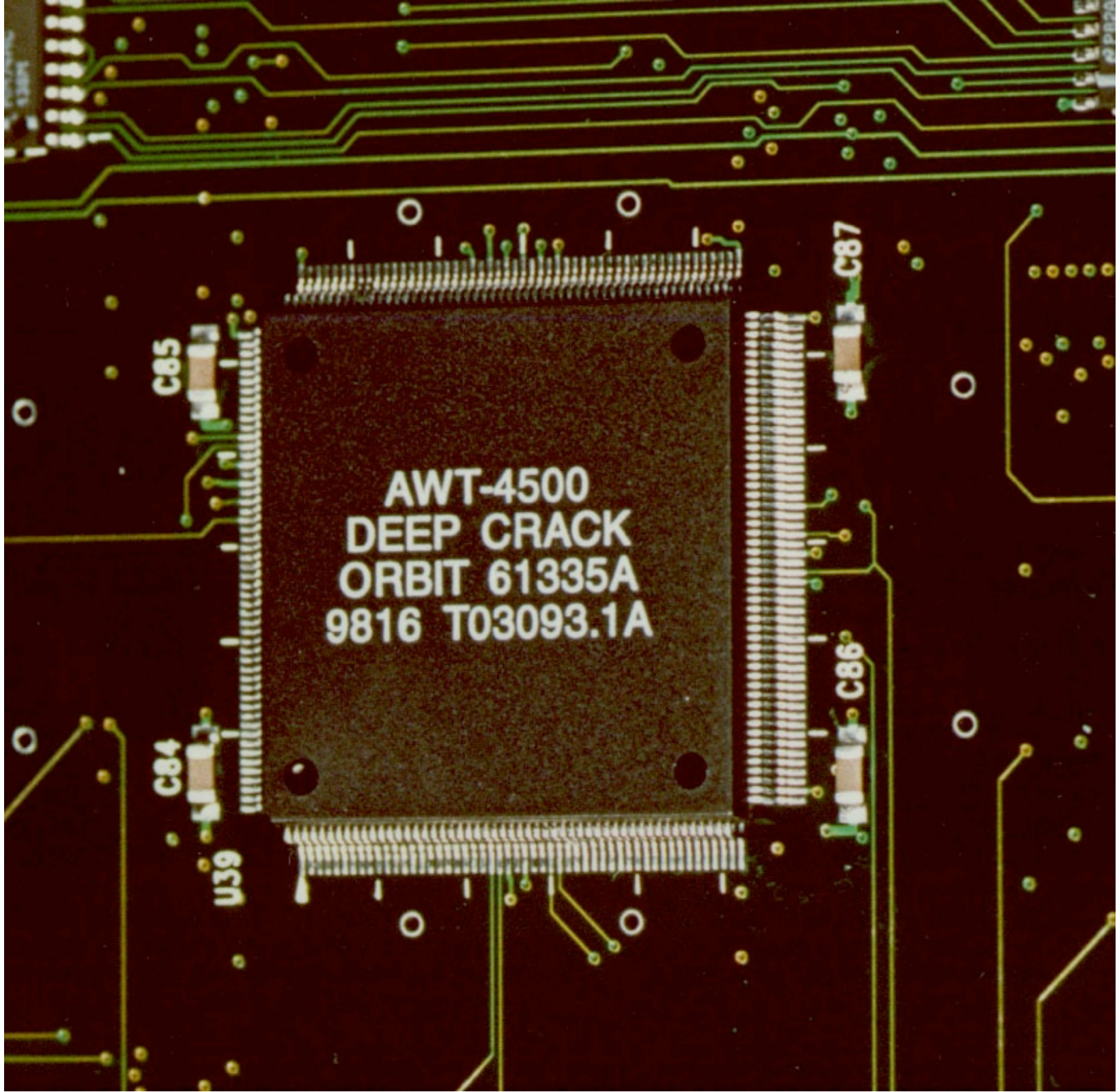
- Hashed password stored in /etc/shadow
- Introduced in System V Release 3.2 (1987)
- BSD4.3 Reno (1990)

Message Digest 5

- 128-bit hash
- Designed by Ron Rivest (of RSA fame)
- Released in 1991

MD5 Passwords

- FreeBSD introduced MD5 based passwords in version 2.0 (1994)
- No limit on password length
- Larger salt
- Example:
  - foobar = 1exIjwX64$cqLmh9lATzNuXZU388mzq0


AWT-4500
DEEP CRACK
ORBIT 61335A
9816 T03093.1A
C85
C87
C84
C86
U39

Other Algorithms

- Blowfish block cipher, 1993
- SHA-1 hash, released in 1995
- SHA-2 hash, released in 2001
- AES block cipher, released 2001
  - Replaces DES as Gov't standard

Mac OS X 10.2, Jaguar

- Release in 2002
- Unix Version 7 DES crypt(3)
- World-readable
  - `nidump passwd .`

Mac OS X 10.3, Panther

- Released in 2003
- Shadow Passwords
  - /var/db/shadow/hash
- Uses SHA-1 (no salt)
- Windows LanManager (LANMAN)

LANMAN Weaknesses

- Case-insensitive
- Effectively limited to 7 characters
  - Truncated at 14
- Computationally cheap (10x DES)
- No salt

Mac OS X 10.4, Tiger

- Released in 2005
- Still shadowed in /var/db/shadow/hash
- SHA-1 with 64-bit salt
- LANMAN only if Windows sharing enabled

Password Attacking Tools

- John the Ripper
- RainbowCrack

Rainbow Tables

- Complete, 7 character LANMAN
  - 229.96 Gb -- 1 year, 7 months
- Complete, 7 character unsalted MD5
  - 1.4 TB -- 10 years, 3 months
- Impractical on salted hashes

Protection

- A good algorithm
- Good passwords

“Good” Passwords

- Random is best!
  - Don’t use words (or variations)
  - Don’t use personal information
- Don’t use password hints
- The longer the better
  - Use passphrase

Password Entropy

- Number of bits required to represent set of all possible passwords
- Example: 11 alphanumeric characters
  - Characters = 26+26+10 = 62
  - Combinations = $62^{11} = 5.2 \times 10^{19}$
  - Entropy, $E = \log_2(62^{11}) = 65.5$ bits

Diceware

- Generates easy to remember random passphrases
- Each word created by rolling 5 dice
  - 14324 = blaze
- Each word has an entropy of 12.9 bits
  - $\log_2(6^5)$ = 12.9 bits
  - 5 words = 64.5 bits

Diceware vs. Random

- 11 character alphanumeric (65.5 bits)
  - 3AsIPRQY6pD
- 5 word Diceware (64.5 bits)
  - 52163-42366-41666-11125-41366
  - roar mirage mccoy aback macon

Four Words

- Four words (51.7 bits) are breakable with a hundred or so PCs
- "You would be content to keep paper copies of the encrypted documents in an ordinary desk or filing cabinet in an unsecured office"

Five Words

- Five words (64.5 bits) are only breakable by an organization with a large budget
- "You need or want strong security, but take no special precautions to protect your computer from unauthorized physical access, beyond locking the front door of your house or office"

More Words

- Six words (77.4 bits) appear unbreakable for the near future, but may be within the range of large organizations by around 2014
- Seven words (90.3 bits) are unbreakable with any known technology, but may be within the range of large organizations by around 2030
- Eight words (103.2 bits) should be completely secure through 2050

Alphanumeric Brute Force

	7	9	11
Entropy	41.7 bits	53.6 bits	65.5 bits
LANMAN	8 days	8 days	8 days
SSHA1	71 days	751 years	2 million years
SMD5	28 years	110,342 years	424 million years

Demo Time