CSE 484 (Winter 2011)

User Authentication

Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

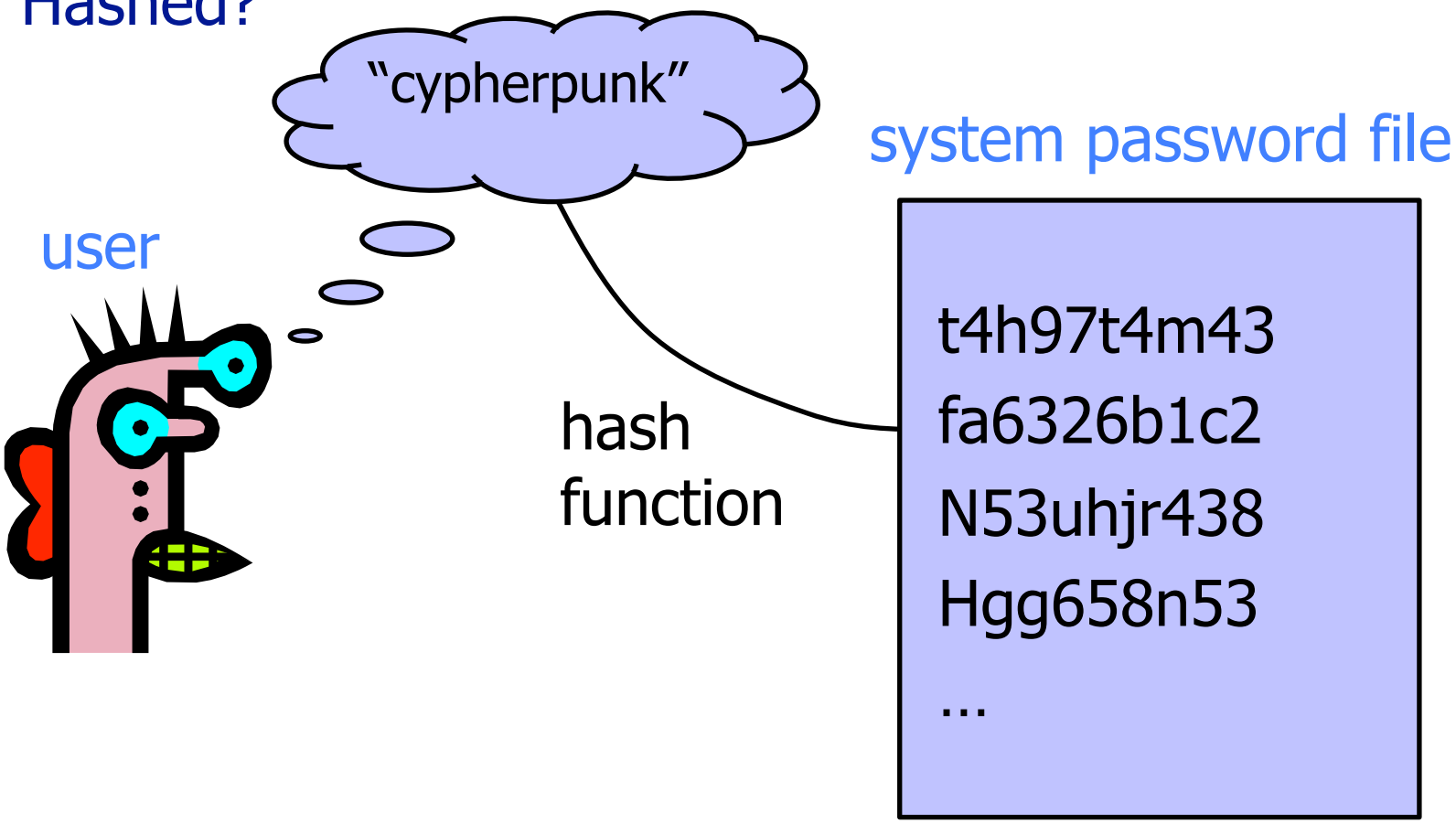
- User authentication
- CELT
- Reminder: HW2, Lab2

Course Evaluation

- Jim Borgford-Parnell
  - Center for Engineering Learning and Teaching (CELT)
- I'd love your feedback on this course
  - Your feedback is valuable! Both positive and negative. (Helpful for the remainder of this course, when possible, and future courses.)
  - Thanks!

UNIX-Style Passwords

- How should we store passwords on a server?
  - In cleartext?
  - Encrypted?
  - Hashed?



Password Hashing

- Instead of user password, store H(password)
- When user enters password, compute its hash and compare with entry in password file
  - System does not store actual passwords!
  - System itself can't easily go from hash to password
    - Which would be possible if the passwords were encrypted
- Hash function H must have some properties
  - One-way: given H(password), hard to find password
    - No known algorithm better than trial and error
    - The attacker doesn't need to find **the** password, just **a** password that hashes to the stored value
  - "Slow" to compute

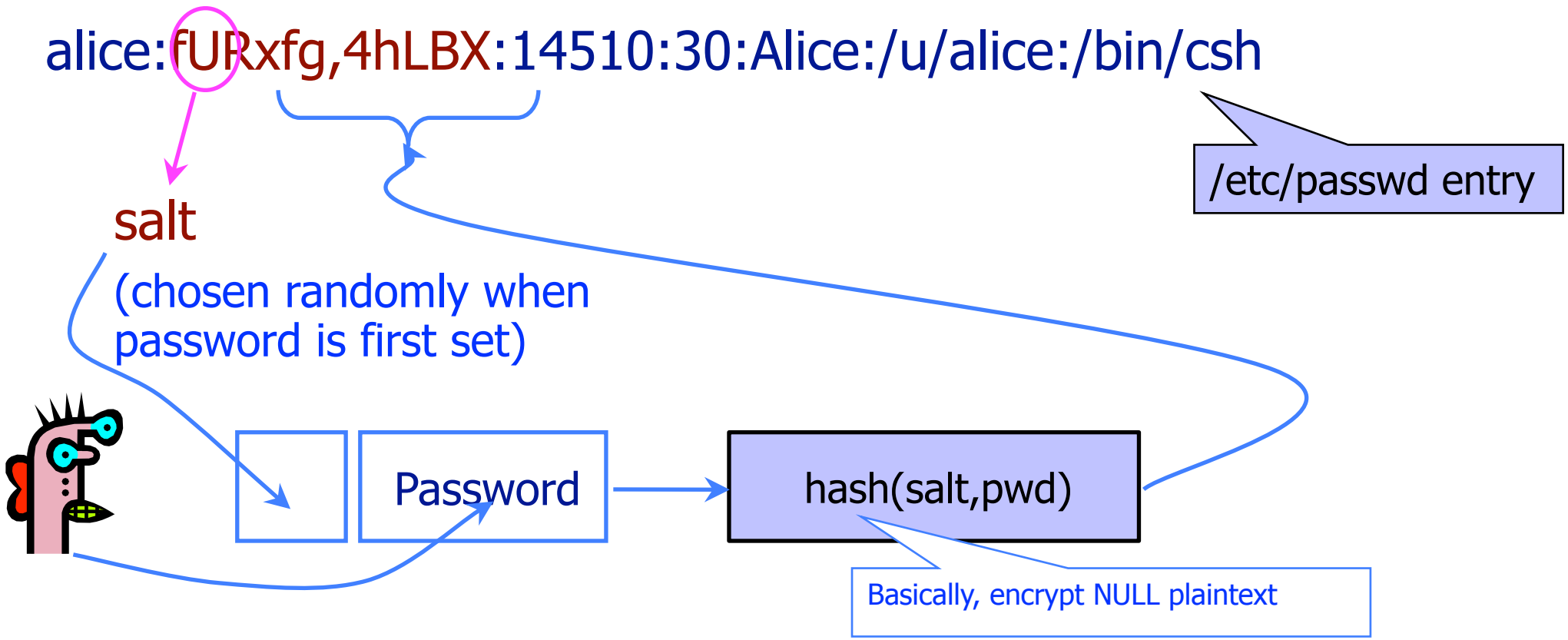
(Early) UNIX Password System

- Uses DES encryption as if it were a hash function
  - Encrypt NULL string using password as the key
    - Truncates passwords to 8 characters!
  - Artificial slowdown: run DES 25 times
    - Why 25 times? Slowdowns like these are important in practice!
  - ("Don't use DES like this at home.")
  - Can instruct modern UNIXes to use cryptographic hash function
- Problem: passwords are not truly random
  - With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are $94^8 \approx$ 6 quadrillion possible 8-character passwords (around 2^{52})
  - Humans like to use dictionary words, human and pet names $\approx$ 1 million common passwords

Dictionary Attack

- Password file /etc/passwd is world-readable
  - Contains user IDs and group IDs which are used by many system programs
- Dictionary attack is possible because many passwords come from a small dictionary
  - Attacker can compute H(word) for every word in the dictionary and see if the result is in the password file
  - With 1,000,000-word dictionary and assuming 10 guesses per second, brute-force online attack takes 50,000 seconds (14 hours) on average
    - This is very conservative. Offline attack is much faster!
  - As described (H(word)), could just create dictionary of "word to H(word)" mapping once -- for all users!!

Salt



- Users with the same password have <u>different</u> entries in the password file
- Online dictionary attack is still possible! (Precomputed dictionaries possible too -- but significantly more expensive.)

Advantages of Salting

- Without salt, attacker can pre-compute hashes of all dictionary words once for all password entries
  - Same hash function on all UNIX machines
  - Identical passwords hash to identical values; one table of hash values can be used for all password files
- With salt, attacker must compute hashes of all dictionary words once for each password entry
  - With 12-bit random salt, same password can hash to 2^{12} different hash values
  - Attacker must try all dictionary words for each salt value in the password file
- Pepper: Secret salt (not stored in password file)

Today on Slashdot

Are You Sure SHA-1+Salt Is Enough For Passwords?

Posted by **CmdrTaco** on Wednesday February 09, @08:47AM
from the good-enough-for-govt-work dept.

Melchett writes

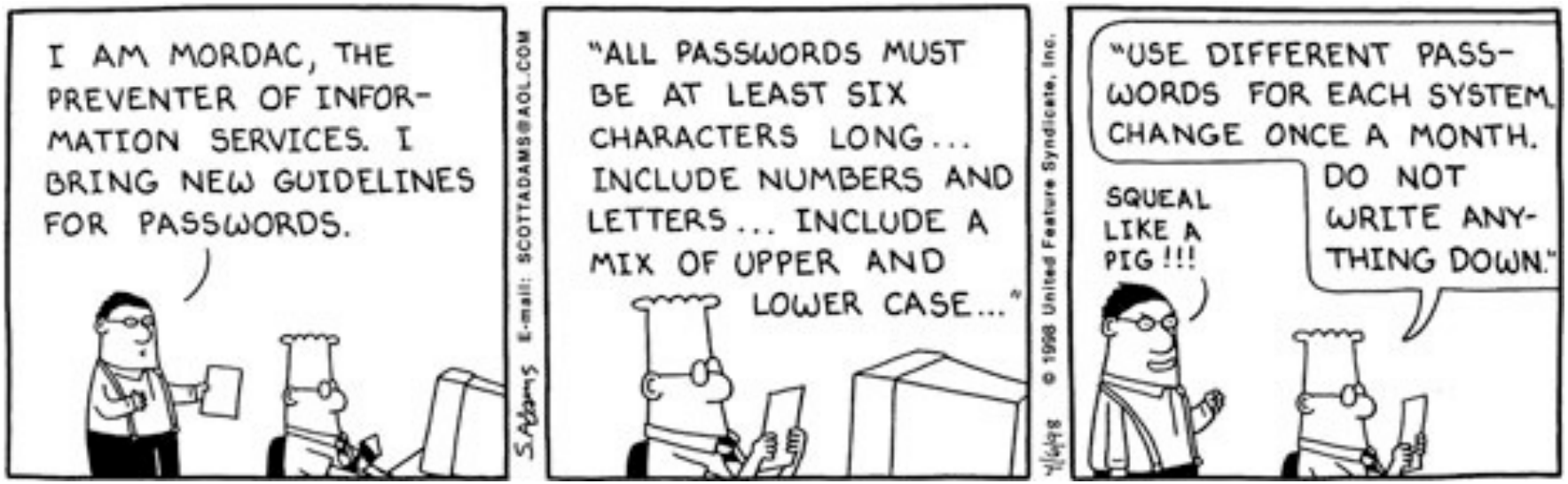
> "It's all too common that Web (and other) applications use MD5, SHA1, or SHA-256 to hash user passwords, and more enlightened developers even salt the password. And over the years I've seen heated discussions on just how salt values should be generated and on how long they should be. Unfortunately in most cases people overlook the fact that MD and SHA hash families are designed for computational speed, and the quality of your salt values doesn't really matter when an attacker has gained full control, as happened with rootkit.com. When an attacker has root access, they will get your passwords, salt, and the code that you use to verify the passwords."

Read the **265** comments

slashdot it security passwords

Other Password Issues

- Keystroke loggers
  - Hardware
  - Software / Spyware
- Shoulder surfing
  - It's happened to me!
- Online vs offline attacks
  - Online: slower, easier to respond
- Multi-site authentication
  - Share passwords?


I AM MORDAC, THE PREVENTER OF INFORMATION SERVICES. I BRING NEW GUIDELINES FOR PASSWORDS.
S. Adams E-mail: SCOTTADAMS@AOL.COM
"ALL PASSWORDS MUST BE AT LEAST SIX CHARACTERS LONG... INCLUDE NUMBERS AND LETTERS... INCLUDE A MIX OF UPPER AND LOWER CASE..."
© 1998 United Feature Syndicate, Inc.
"USE DIFFERENT PASSWORDS FOR EACH SYSTEM. CHANGE ONCE A MONTH. DO NOT WRITE ANYTHING DOWN."
SQUEAL LIKE A PIG!!!

"Improving" Passwords

- Add biometrics
  - For example, keystroke dynamics or voiceprint
  - Revocation is often a problem with biometrics
- Graphical passwords
  - Goal: increase the size of memorable password space
- Password managers
- Two-factor authentication
  - Leverages user's phone (or other device) for authentication