CS 465

Passwords

Slides by Kent Seamons and Tim van der Horst
Last Updated: Nov 30, 2011

Goals

- Understand UNIX pw system
  - How it works
  - How to attack
- Understand Lamport’s hash and its vulnerabilities

History of UNIX passwords

- Originally the actual passwords were stored in a plaintext file
  - “Excessively vulnerable to lapses in security”
- Improved approach used encryption to protect passwords
  - Led to brute force/dictionary attacks

Pass Phrases

- Passwords is a misnomer
  - Do not use single words or variants
  - Supposedly, a large number of passwords in Dallas is some variant of the word cowboys
    - Any cougar passwords out there!
- Use a pass-phrase
  - Memorable and harder to guess
  - First letter of a long phrase
    - Rastcao - Rise and shout the cougars are out

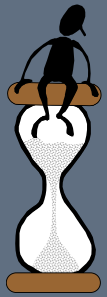
How to Attack Password Systems

- Guess the user's password
  - Online attack
    - Attempt to login as the user would
  - Offline attack
    - Repeated guessing involving an encrypted form of the user's password
- Shoulder surfing
- Users write down their passwords
- Users give away their passwords
  - Phishing, social engineering

Problems with Passwords

- Users have too many passwords
  - Encourages password reuse
  - Leads to forgotten passwords
  - Burdens users and administrators
- Attempts to increase password strength inconvenience users
- Random passwords
  - Only as random as the initialization of the salt value

Time estimates



- What is the maximum number of attempts to guess a password?
  - Password length = 8 characters
  - Assume password is alphanumeric (26+26+10)
  - $(26+26+10)^8 = 62^8$
- How many attempts on average? Divide maximum number by 2 (this assumes brute force attack and passwords chosen randomly)

Unix Passwords

Unix Password File

- Original password file /etc/passwd was world readable
  - Anyone could copy the file offline and perform a dictionary attack
  - You could find sample files on Google courtesy of naïve system admins!

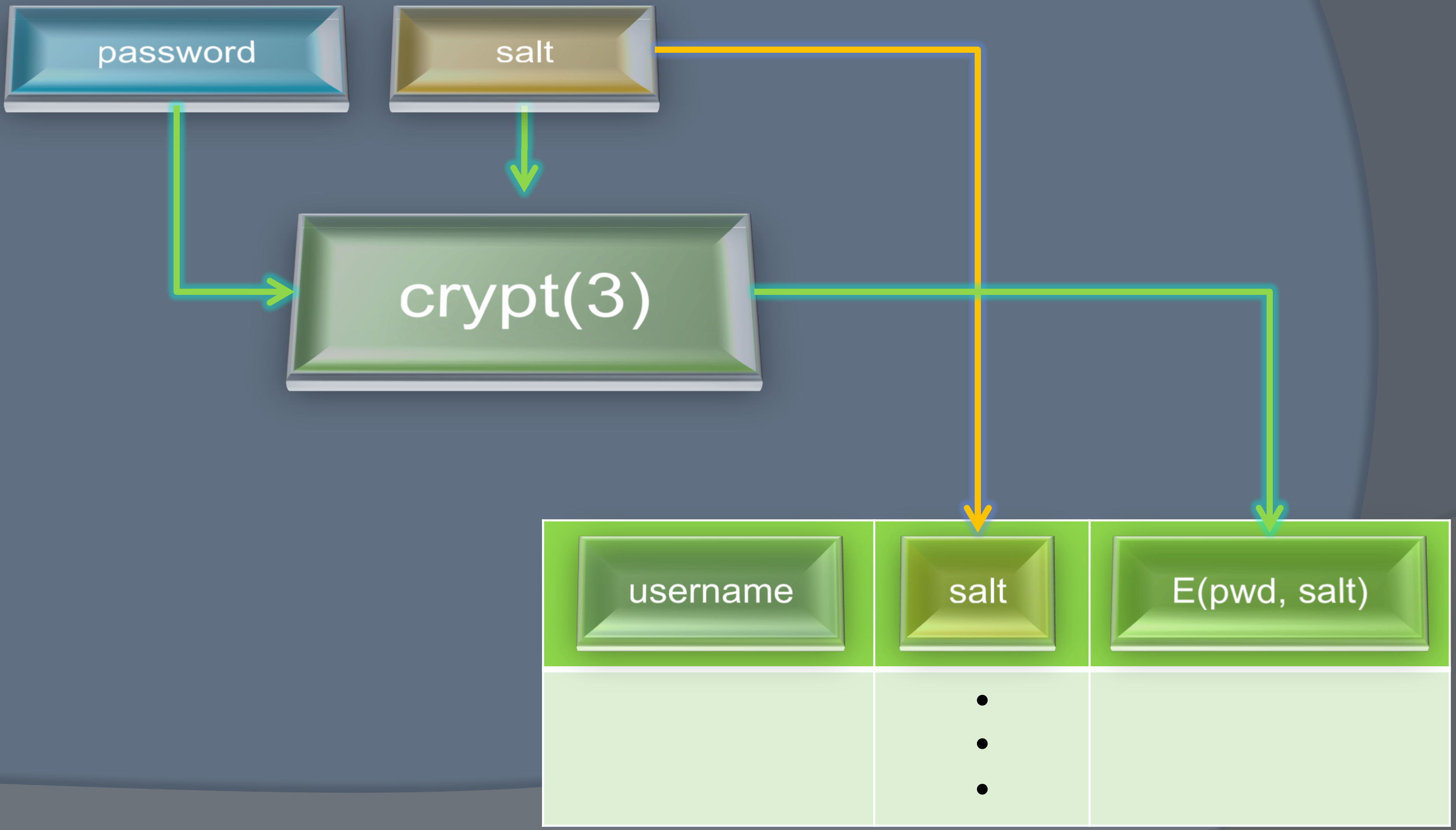
- Later, the encrypted password was moved to a shadow file /etc/shadow that required root privileges to access

THE UNIX CRYPT FUNCTION

crypt(3)

Slower is better

Unix Password File Creation



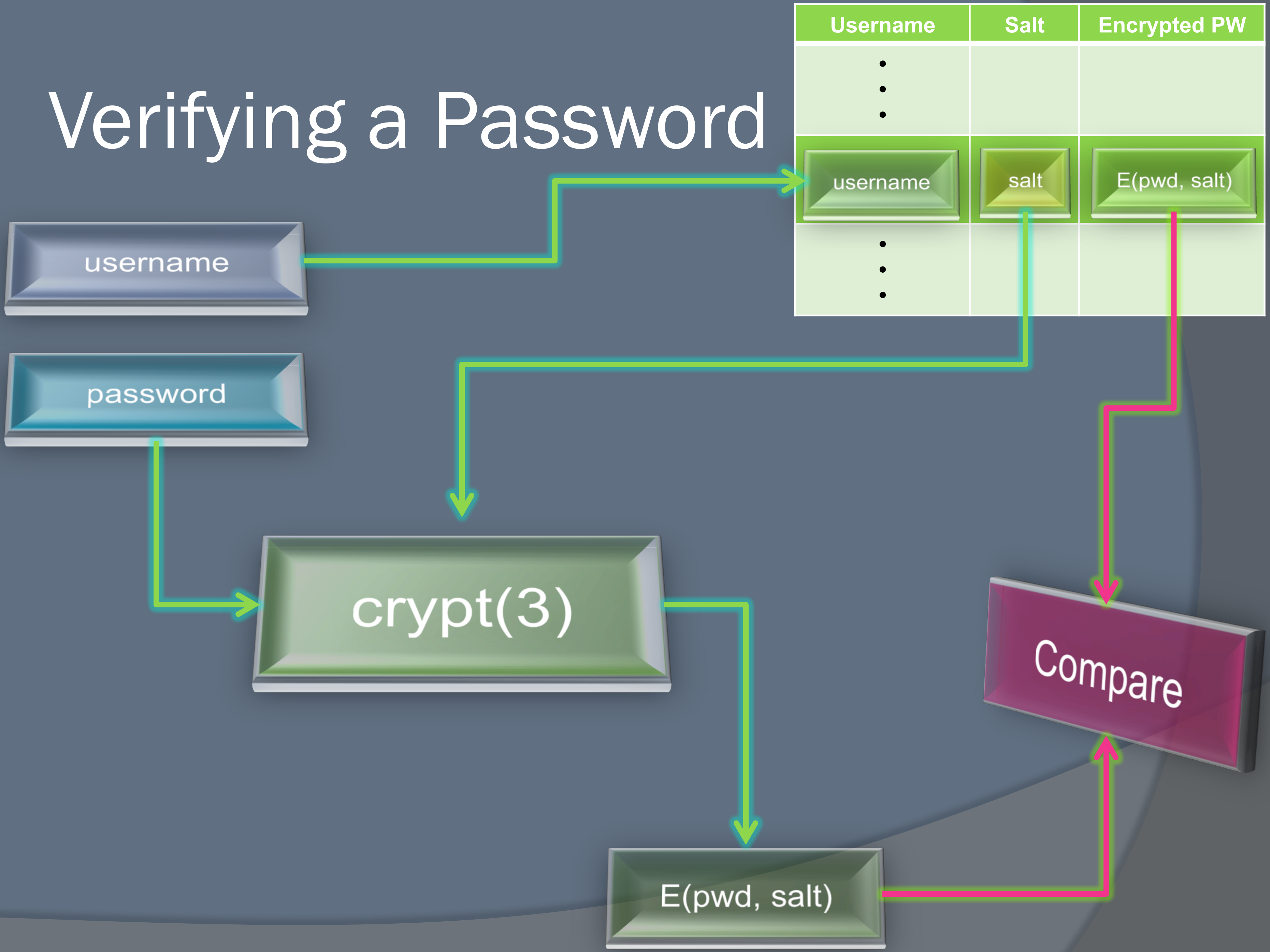
password

salt

crypt(3)

username	salt	E(pwd, salt)
	⋮	

Verifying a Password

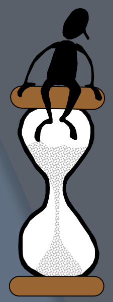


Username	Salt	Encrypted PW
⋮		
username	salt	E(pwd, salt)
⋮		

Password Salts

- Why do Unix password files use a salt?
  - Prevents the identification of identical passwords
    - Provided each user has a different salt
  - All password guesses are salt-specific
    - Guess made with one salt aren’t helpful for another
    - Increases the cost of offline attack to crack any password in the file
    - Increases the size requirement for a pre-computed database of hashed passwords

Password Attacks with Salt



- How many guesses do password attacks need when a salt is used?
  - Off-line attack – one attempt for each unique salt in the file
- How does the salt impact on-line attacks? It doesn’t
- How does the salt impact an attempt to crack a specific user’s password in the file? It doesn’t change the number of attempts, but it does prevent a pre-computed database of passwords

crypt(3)

- 2 approaches
  - A modified DES implementation (uses a salt)
    - Can’t use off the shelf DES hardware
    - Effectively limits the size of all passwords to eight characters
  - MD5 hash
    - Any size password
    - Hash function is invoked 1000 times
      - An attack that would have taken 1 day now takes 3 years
      - Minimal impact to user and the system

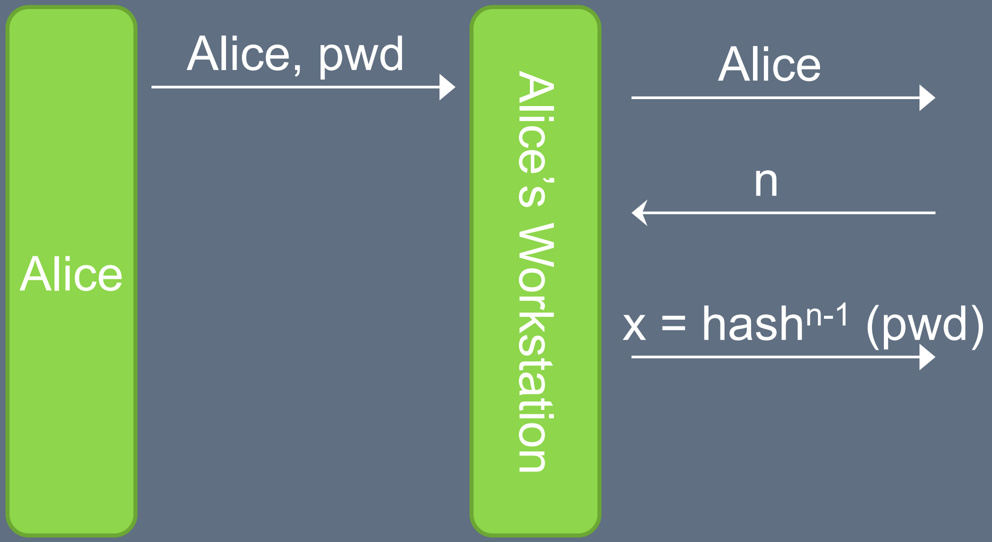
Password Guessing Attacks

- Brute-force
- Dictionary
- Substitution
  - password, passw0rd

Lamport’s Hash

Lamport’s Hash

- One time password scheme
  see http://lodestone.org/people/hoss/ops/node5.html



Alice

Alice, pwd

Alice’s Workstation

Alice

n

x = hash^{n-1} (pwd)

Bob

BOB

knows ⟨ n, hash^{n}(password) ⟩

Compares hash(x) to
hash^{n} (password);

If equal, replaces
⟨ hash^{n} (password) ⟩
With
⟨ n-1, x ⟩

Attack on Lamport’s Hash

- Small n attack
  - Active attacker intercepts servers reply message with n and changes it to a smaller value
  - Attacker can easily manipulate the response (repeatedly) to impersonate Alice
- Eavesdropper captures Alice’s hashed reply and conducts off-line attack
- Replay Alice’s response to other servers where Alice may use the same password
  - Thwart using salt at the server – server hashes pw || salt and sends n and the salt to Alice during login
  - Salt also permits automatic password refresh when n reaches 1

Related articles (optional)

- The Curse of the Secret Question http://www.schneier.com/essay-081.html
- Sarah Palin Yahoo! account hacked http://www.informationweek.com/news/security/cybercrime/showArticle.jhtml?articleID=210602271 http://en.wikipedia.org/wiki/Sarah_Palin_email_hack
- Secret Questions Too Easily Answered http://www.technologyreview.com/web/22662/
- Scientists claim GPUs make passwords worthless http://www.pcpro.co.uk/news/security/360313/scientists-claim-gpus-make-passwords-worthless