OS Security

Jonathan Windle

University of East Anglia

J.Windle@uea.ac.uk

June 10, 2017

Overview I

Important Concepts

- Confidentiality:
  - Data confidentiality: Private information is discoldes to only those with appropriate authorisation
  - Privacy: Users control the information about them that is collected and stored and how it may be disclosed
- Integrity:
  - Data is changed only in authorised ways
- Availability:
  - Access to a system should not be denied to those that are authorised to use it.

Basiscs of Cryptography

- Purpose is to take a message of file called plaintext and encrypt it into ciphertext in such a way that only authorised people know how to convert it back
- The encryption and decryption algorithms (functions) should always be public.
- Security depends on parameters to the algorithms called keys
- $C = E(P, K_E)$ where C is the ciphertext file, E is the encryption function, P is the plaintext file, K_E is the encryption key.

Secret-Key Cryptography (SKC)

- The encryption key is secret
- The system appears safe because although the crypanalyst knows the general system he does not know which of a huge number of possible keys is in use.
- The basic strategy for attack takes advantage of statistical properties of natural languages
- Breaking a cipher using a computer to try different guesses is actually straightforward.
- Both sender and receiver need to possess shared secret key

Public-Key Cryptography (PKC)

- Distinct keys used for encryption and decryption
- Given a well chosen encryption key it is virtually impossible to determine the decryption key
- For example RSA exploits fact that multiplying really big numbers is easy for a computer but factoring them is hard.
- In PKC, everyone picks a public/private key pair and publishes the public key. The public key is the encryption key and the private key is the decryption key.
- To send the message, the correspondant encrypts it with the receivers public key. The receiver then decrypts with the private key.

Mechanisms for Security

- Authentication
- Authorisation
- Enforcement

Authentication

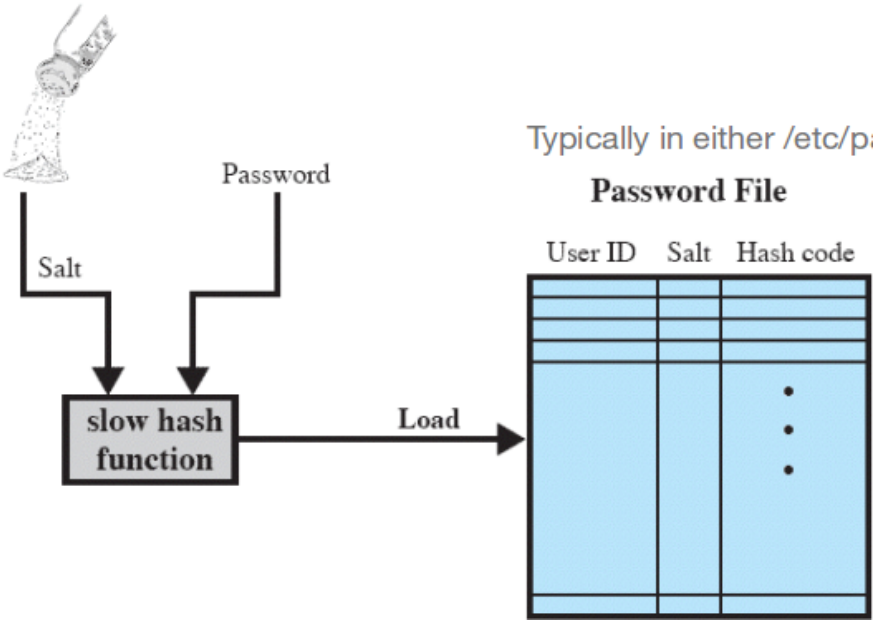
- Three broad mechanisms:
  1. Use something the user knows about (e.g. password etc.)
  2. Use something the user possesses (key card etc.)
  3. Use something intrinsic to the user (biometric) e.g. face scan, iris scan, thumb scan

Password Based Authentication

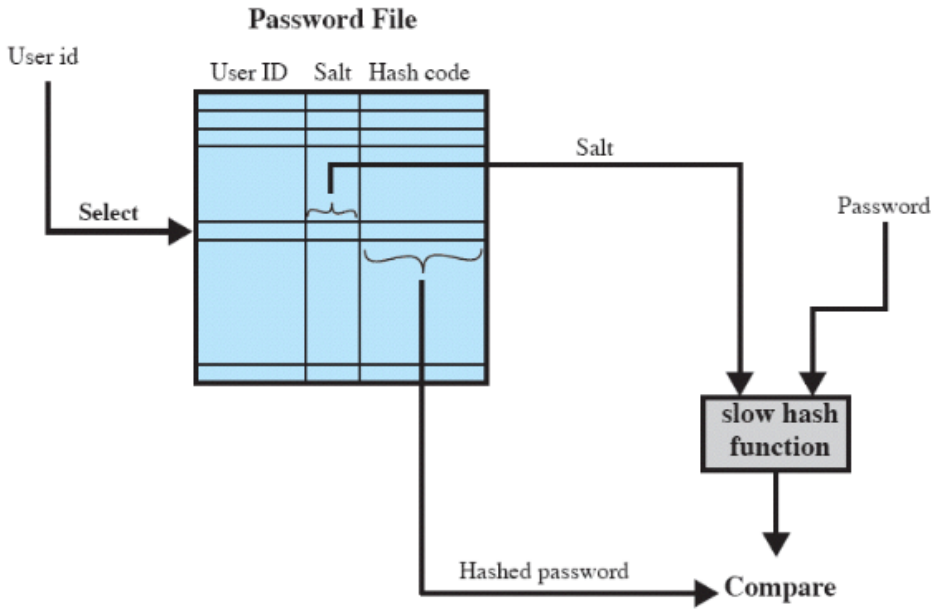
- The system checks an entered password against a stored password for the user ID-straightforward?
- Problems with password-based authentication
  - Too short, easy to remember and guess
  - Too long, hard to remember and guesss
  - People make unwise choices for their password
  - Passwords using real words are subject to "dictionary" attacks.
  - Users write passwords down
- If pasword is entered correctly the assumption is the user is who they claim to be
- System must store the passwords
  - Stored in encrypted hashed form
  - System compares encrypted versions
- Robustness improved by adding "salt" value
  - 12-bit random number added to password
  - Makes passwords 4096 times more difficult to guess

Salted Passwords

Generating

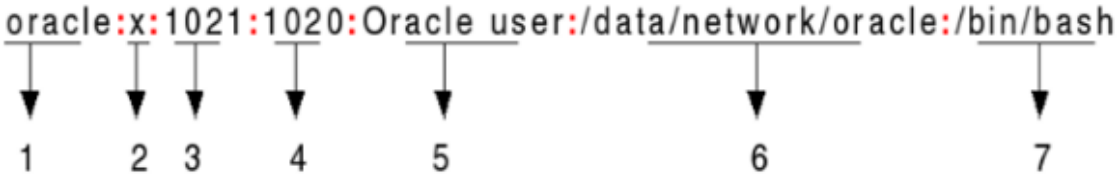


Authenticating



Linux Authentication

/etc/passwd



1. Username - should be between 1-32 characters in length
2. Password - An x character the password is stored in file `/etc/shadow`
3. User ID (UID) - Each user must have have an assigned UID (0-999 are reserved).
4. Group ID (GID) - Users primary GID (stored in file `/etc/group`
5. User ID info - Comment field
6. Home Directory - absolute path to the directory the user will be in when they log in
7. Command/Shell - absolute path to a command or shell (e.g. `/bin/bash`

Encryption

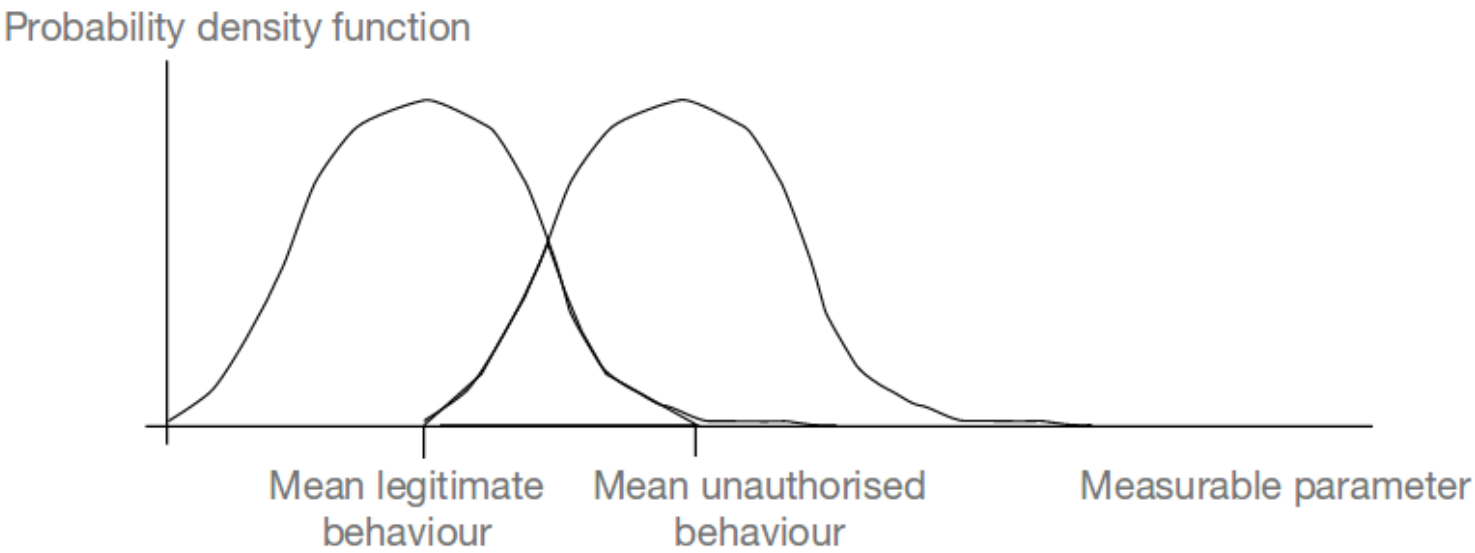
- Hashed form generated by a "one-way" function
  - A one-way function that is easy to compute on every input, but hard to invert given the image of a random input
  - Here "easy" and "hard" are to be understood in the sense of computational complexity.
  - There are several candidates for "one-way" functions. A common approach is based on multiplication and factorization
  - To factor 232-digit number (RSA-76A) took two years (100s of machines) in 2009.
- Making the cryptographic hash function "slow" to execute improves robustness against attack
- As does enforcing a delay before passwords can be re-entered

Zero-Knowledge Autgentication

- Solves problems inherent in password authentication
- The user knows some "secret" but must prove they know the secret without ever revealing it
- The system verifies the user knows the secret without having to learn the secret
- Since the secret is never revealed this method is zero-knowledge verification/authentication.

Intrusion Detection

- A specialist software layer designed to detect abnormal patterns of behaviour.
- Sensors collect data describing behaviour patterns
- Statistical analysis is used to determine likelihood of attempted unauthorised access
- All multi-user systems maintain audit records activity logs; who performed what action with a particular object
- Typically an audit record might contain subject, action, object, exception thrown, resource usage, time-stamps



Access Control Matrices

- File access controlled by 2D array:
  - Rows list users
  - Cols list files
- Set row/col cell to 1 if user is allowed access
- Allows access to be set on individual files
- Matrix is very large for large file systems
- Set permissions on a wider scale using user classes typically three groups:
  - User/Owner, groups, everyone else (world)

Summary

- Security is vital at all levels of a system if any comoponent is compromised the the whole system could be compromised
- Password authentication can combat eavesdropping
- Intrusion detection looks for unusual patterns of behaviour threshold setting can be a challenge
- Authorisation is enforced by access control mechanisms

The End