Computer Security – Principles and Practice (Pearson, fourth edition)

W. Stallings, L. Brown

* These slides are an adaptation of the original slides of the authors of the book

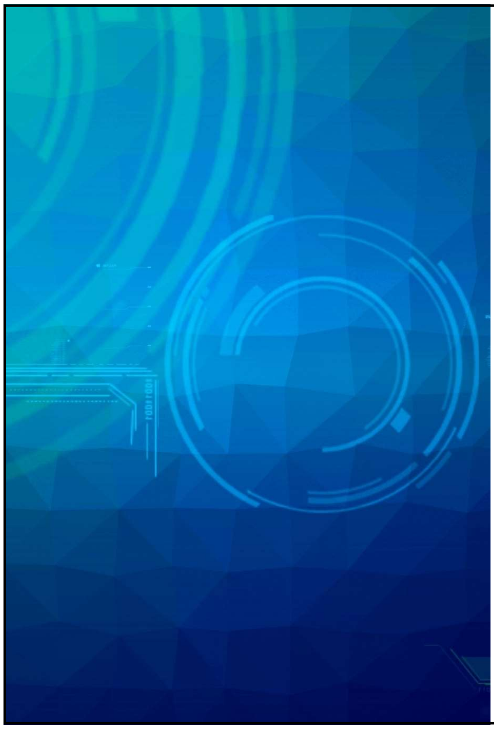
User authentication

1

Learning objectives

- Discuss the four general means of authenticating a user's identity.
- Explain the mechanism by which **hashed passwords** are used for user authentication.
- Present an overview of **token-based user authentication**.
- Introduce the basics of **biometric authentication**.
- Discuss the issues involved and the approaches for **remote user authentication**.
- Summarize some of the key **security issues** for user authentication.

2



Preliminary question

?

WHEN DO YOU THINK YOU HAVE BEEN SUBJECT TO USER AUTHENTICATION IN YOUR EXPERIENCE?

CAN YOU IDENTIFY THE TWO PHASES OF IDENTIFICATION AND VERIFICATION?

3

Two functions for user authentication

1. User identification
   - by means of a credential or an ID provided by the user to the system
2. User verification
   - by the exchange of authentication information
   - establishes the validity of the claim

- Note: user authentication is distinct from message authentication!

4

Digital Authentication Guideline

NIST SP 800-63-3 defines digital user authentication as (October 2016) :

"The process of establishing confidence in user identities that are presented electronically to an information system."

5

Identification and authentication security requirements

Basic Security Requirements (NIST SP 800-171):

1. Identify users, processes acting on behalf of users, or devices.
2. Authenticate (or verify) the identities of those users, processes, or devices
   - prerequisite to allowing access to organizational information systems.

6

Identification and authentication security requirements

Derived Security Requirements:

3. Use multifactor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.

4. Employ replay-resistant authentication mechanisms for network access to all accounts.

5. Prevent reuse of identifiers for a defined period.

6. Disable identifiers after a defined period of inactivity.

7. Enforce a minimum password complexity and change of characters when new passwords are created.

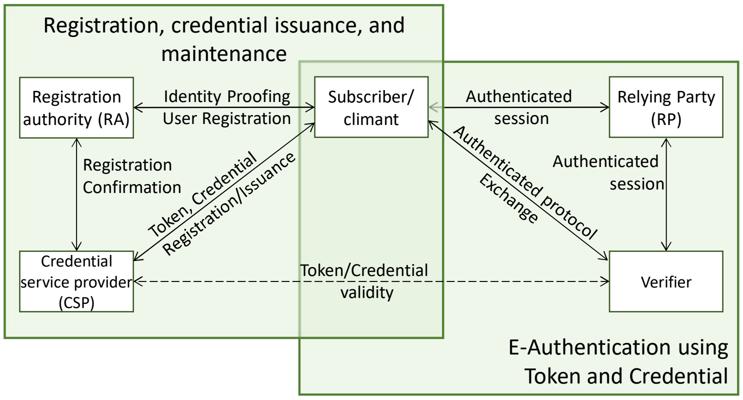
8. Prohibit password reuse for a specified number of generations.

9. Allow temporary password use for system logons with an immediate change to a permanent password.

10. Store and transmit only cryptographically-protected passwords.

11. Obscure feedback of authentication information.

Authentication architectural model (NIST)


Registration, credential issuance, and maintenance
Registration authority (RA)
Identity Proofing
User Registration
Subscriber/ climant
Authenticated session
Relying Party (RP)
Registration Confirmation
Token, Credential Registration/Issuance
Authenticated protocol Exchange
Authenticated session
Credential service provider (CSP)
Token/Credential validity
Verifier
E-Authentication using Token and Credential

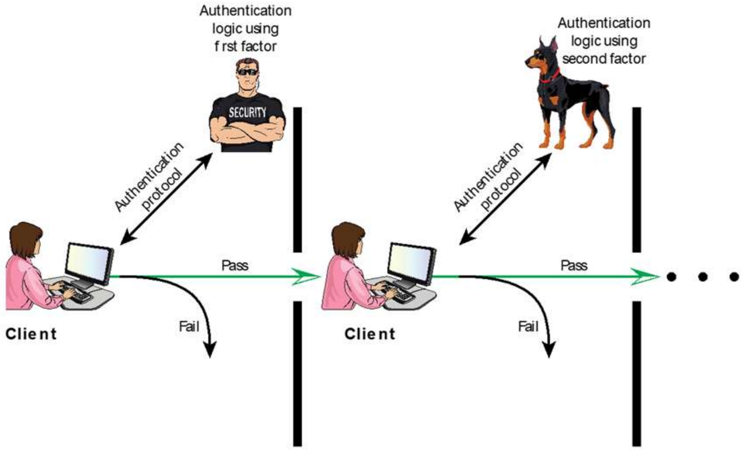
The NIST SP 800-63-3 E-authentication architectural model

four means of authenticating user identity

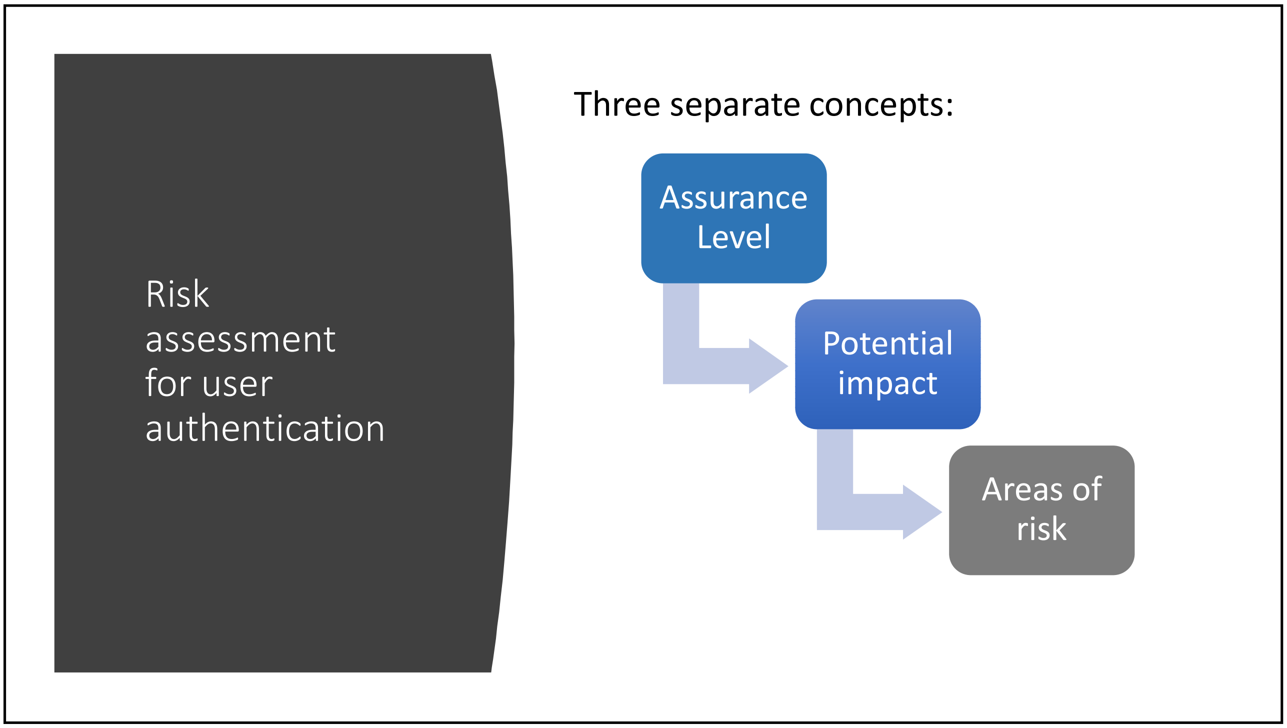
Something the individual knows	Password, PIN, answers to prearranged questions
Something the individual possesses (token)	Smartcard, electronic keycard, physical key
Something the individual is (static biometrics)	Fingerprint, retina, face
Something the individual does (dynamic biometrics)	Voice pattern, handwriting, typing rhythm

9

Multifactor authentication



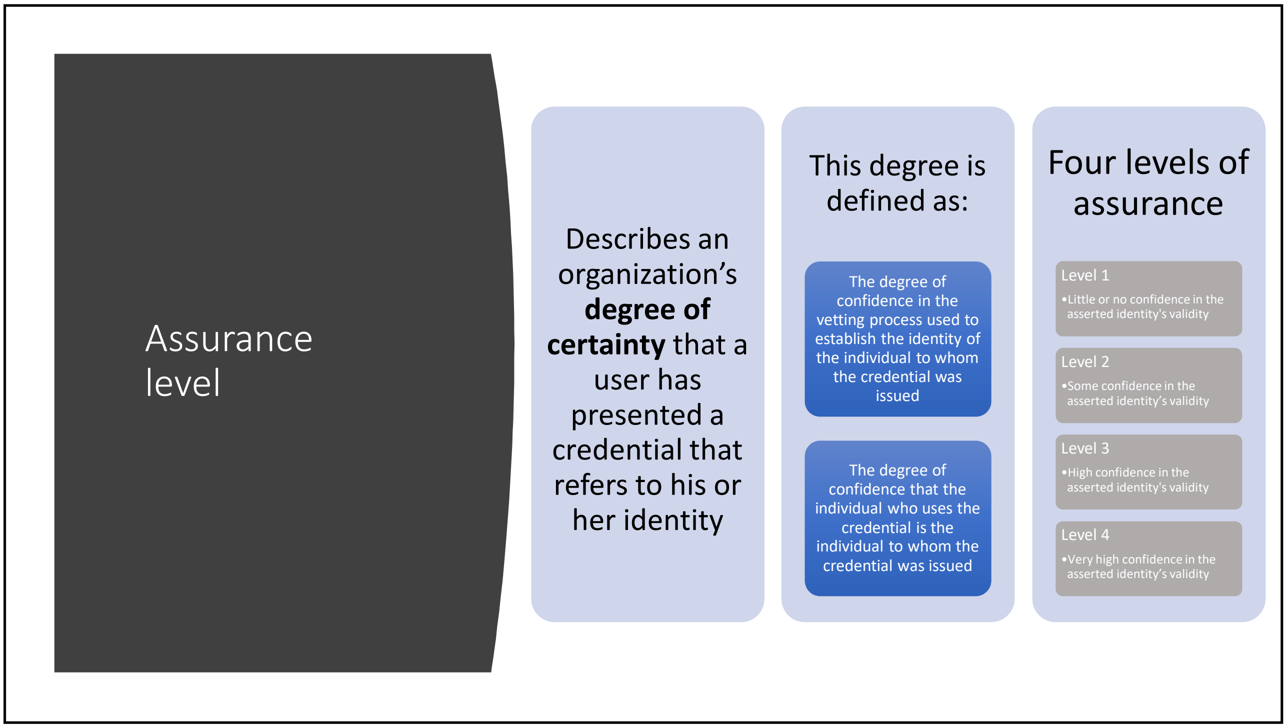
10



Risk assessment for user authentication

Three separate concepts:

11



Assurance level

Describes an organization’s **degree of certainty** that a user has presented a credential that refers to his or her identity

This degree is defined as:

- The degree of confidence in the vetting process used to establish the identity of the individual to whom the credential was issued
- The degree of confidence that the individual who uses the credential is the individual to whom the credential was issued

Four levels of assurance

Level 1

- Little or no confidence in the asserted identity’s validity

Level 2

- Some confidence in the asserted identity’s validity

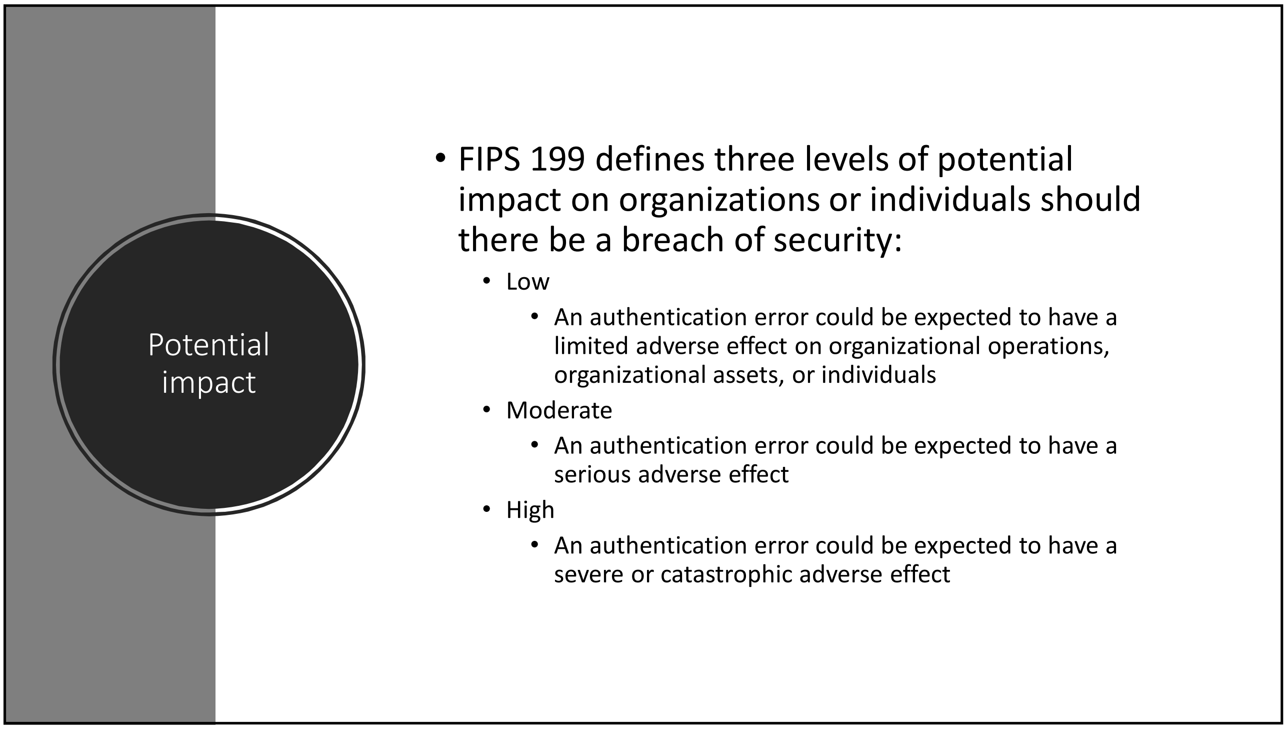
Level 3

- High confidence in the asserted identity’s validity

Level 4

- Very high confidence in the asserted identity’s validity

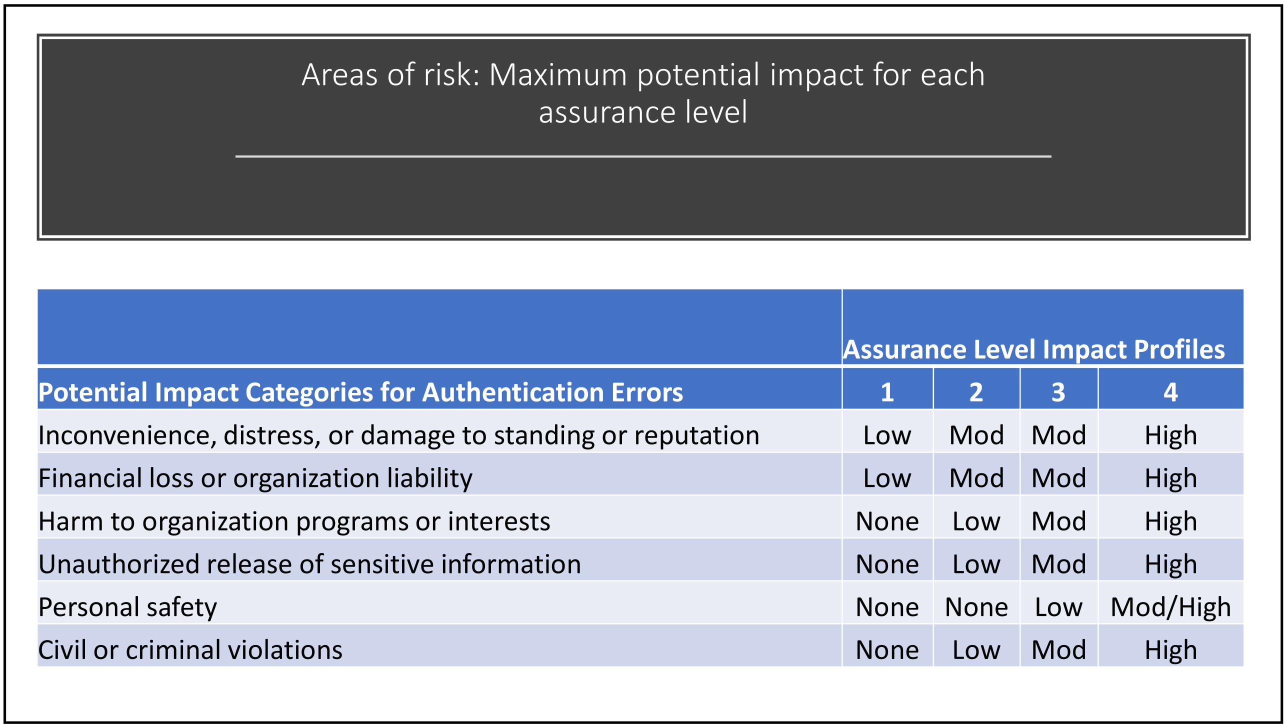
12



Potential impact

- FIPS 199 defines three levels of potential impact on organizations or individuals should there be a breach of security:
  - Low
    - An authentication error could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals
  - Moderate
    - An authentication error could be expected to have a serious adverse effect
  - High
    - An authentication error could be expected to have a severe or catastrophic adverse effect

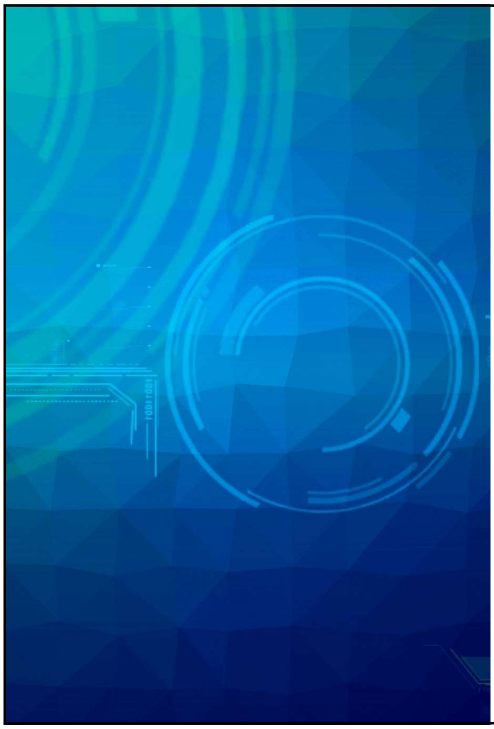
13



Areas of risk: Maximum potential impact for each assurance level

	Assurance Level Impact Profiles			
Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
Inconvenience, distress, or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or organization liability	Low	Mod	Mod	High
Harm to organization programs or interests	None	Low	Mod	High
Unauthorized release of sensitive information	None	Low	Mod	High
Personal safety	None	None	Low	Mod/High
Civil or criminal violations	None	Low	Mod	High

14



Review question

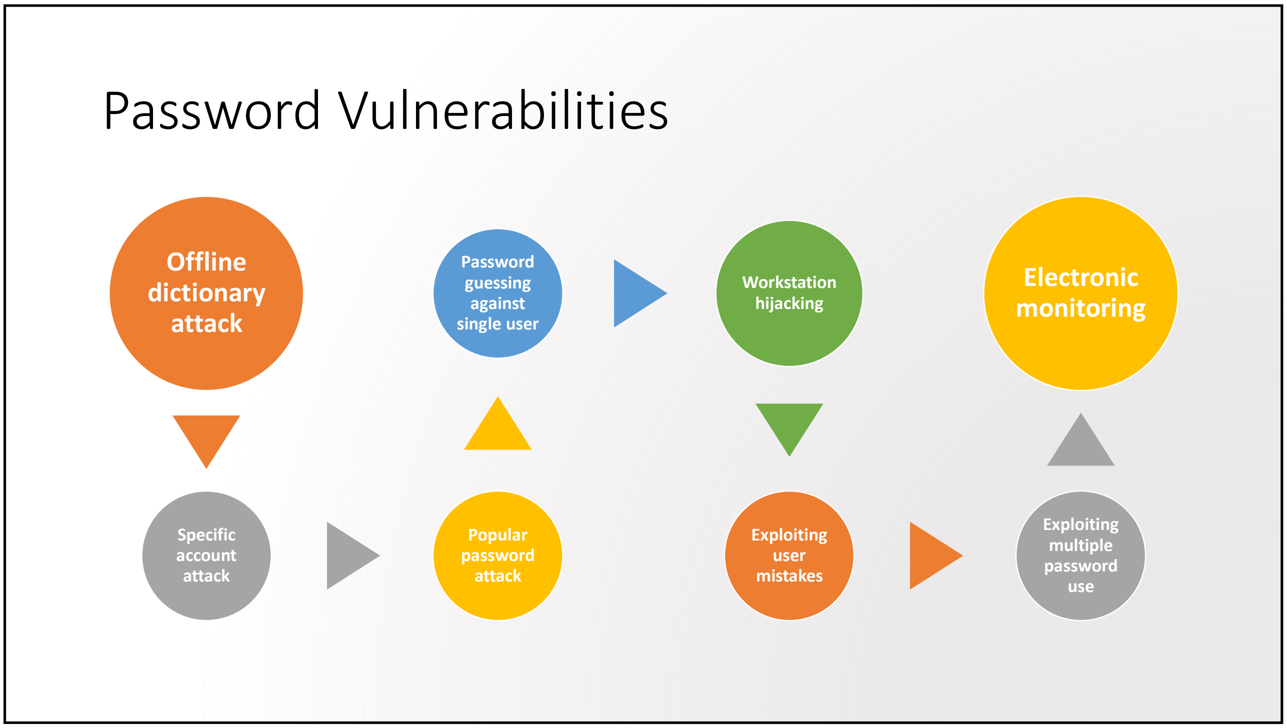
REFERRING TO THE NIST SP 800-63-3 MODEL, DID YOU EVER EXPERIENCE A CASE OF AUTHENTICATION THAT FOLLOWS THAT MODEL?
DESCRIBE THAT AUTHENTICATION SYSTEM FROM THE USER PERSPECTIVE.

15

Password-based authentication

- Widely used line of defense against intruders
  - User provides name/login and password
  - System compares password with the one stored for that specified login
- The user ID:
  - Determines that the user is authorized to access the system
  - Determines the user’s privileges
  - Is used in discretionary access control

16



Password Vulnerabilities

Offline dictionary attack

Specific account attack

Popular password attack

Password guessing against single user

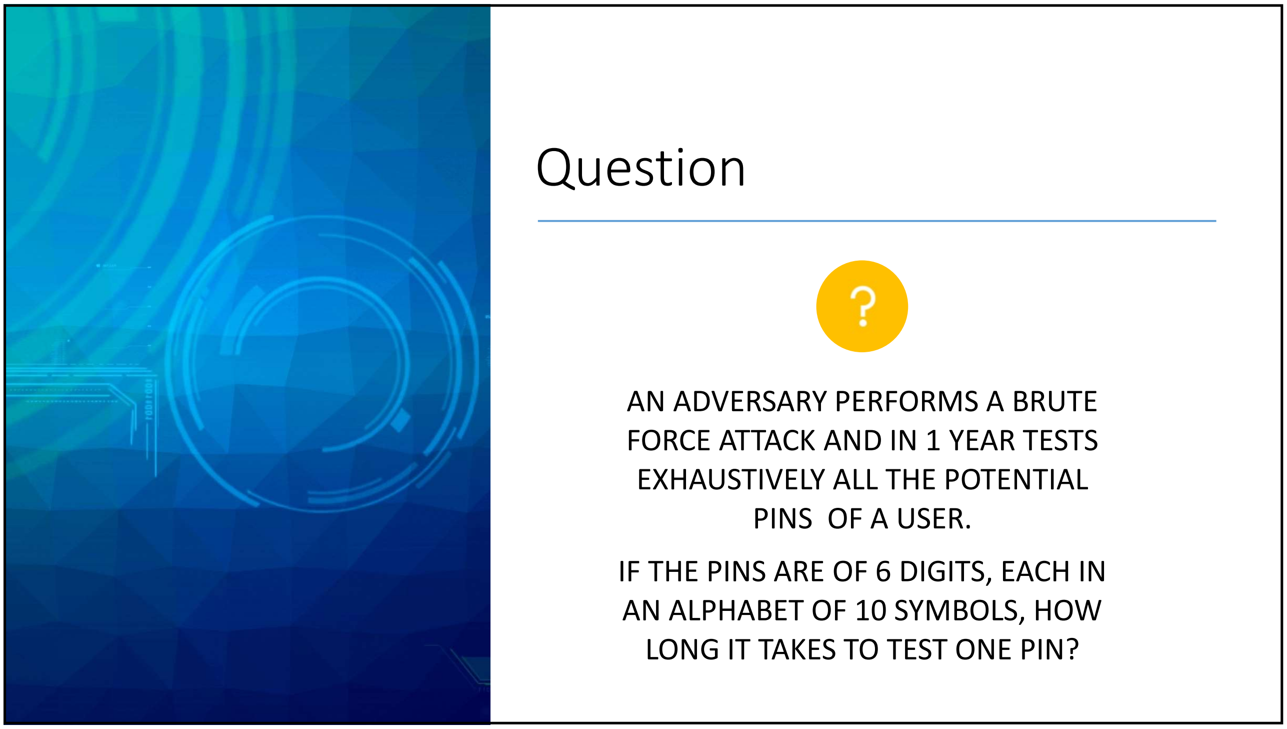
Workstation hijacking

Exploiting user mistakes

Exploiting multiple password use

Electronic monitoring

17



Question

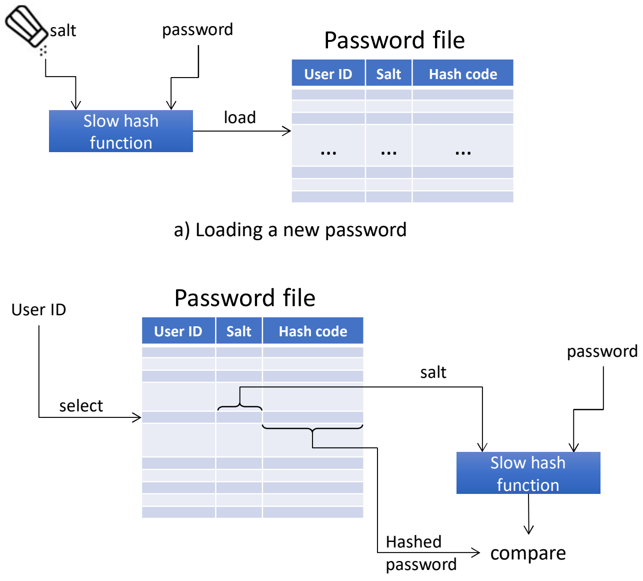
?

AN ADVERSARY PERFORMS A BRUTE FORCE ATTACK AND IN 1 YEAR TESTS EXHAUSTIVELY ALL THE POTENTIAL PINS OF A USER.

IF THE PINS ARE OF 6 DIGITS, EACH IN AN ALPHABET OF 10 SYMBOLS, HOW LONG IT TAKES TO TEST ONE PIN?

18

Unix password scheme



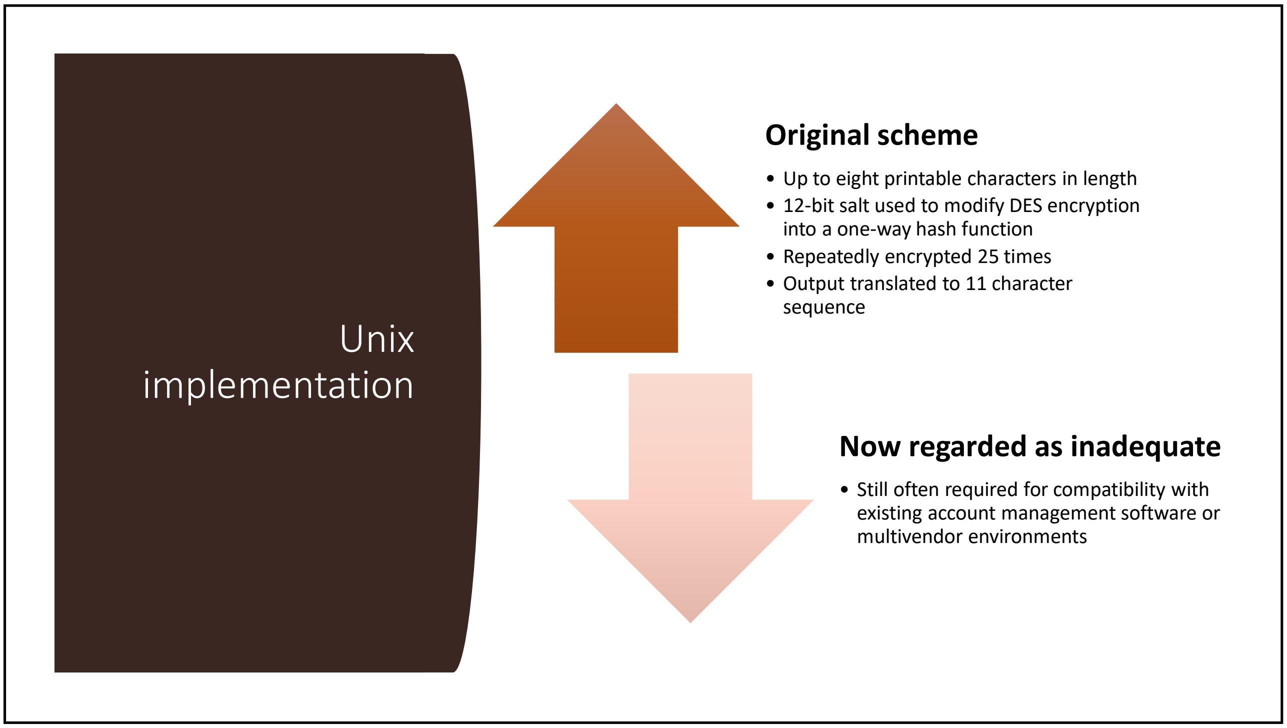
a) Loading a new password

b) Verifying a password

Unix password scheme

The salt serves three purposes:

- It prevents duplicate passwords from being visible in the password file.
- It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.
- It greatly increases the difficulty of offline dictionary attacks.
  - Say the attacker obtains a copy of the password file and salt is not used
    - … and its goal is to guess a single password of any user.
  - The attacker may submit a large number of likely passwords to the hashing function:
    - If any of the guesses matches one of the hashes in the file, then the attacker has found a password that is in the file.
  - Using salt instead, the attacker must take each guess and submit it to the hash function once **for each** salt value in the dictionary file, multiplying the number of guesses that must be checked.



Unix implementation

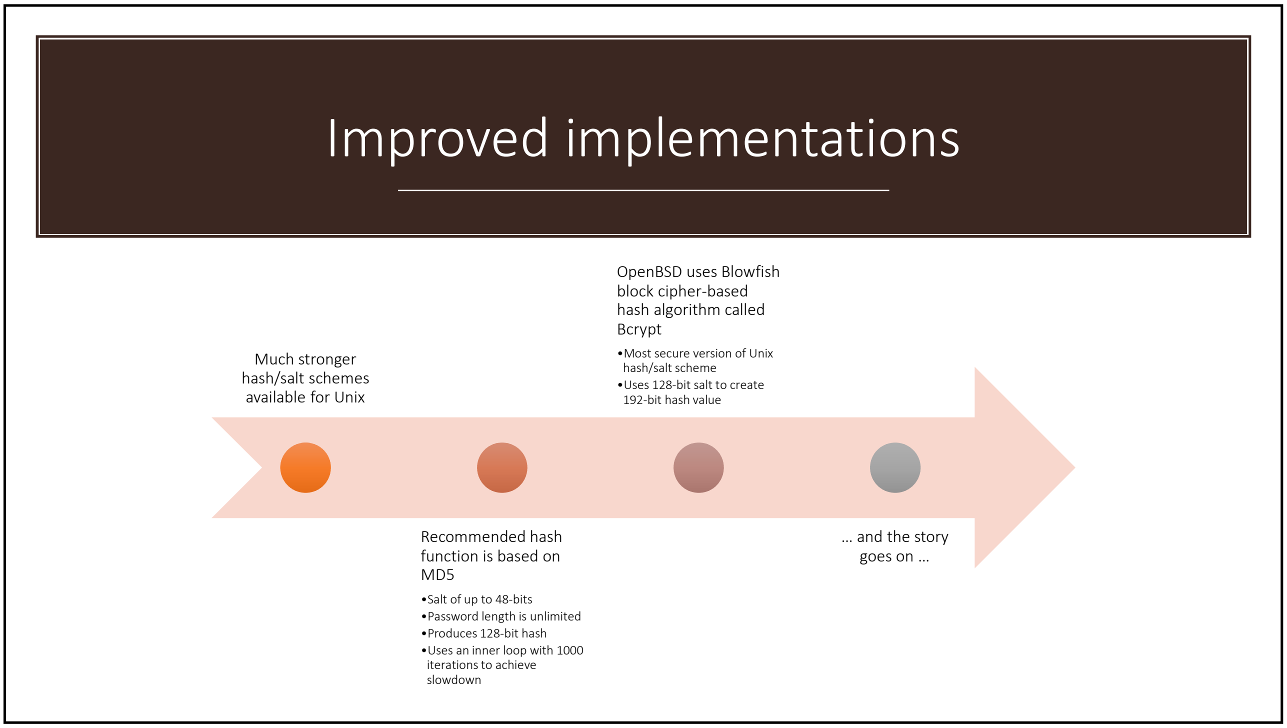
Original scheme

- Up to eight printable characters in length
- 12-bit salt used to modify DES encryption into a one-way hash function
- Repeatedly encrypted 25 times
- Output translated to 11 character sequence

Now regarded as inadequate

- Still often required for compatibility with existing account management software or multivendor environments

21



Improved implementations

Much stronger hash/salt schemes available for Unix

Recommended hash function is based on MD5

- Salt of up to 48-bits
- Password length is unlimited
- Produces 128-bit hash
- Uses an inner loop with 1000 iterations to achieve slowdown

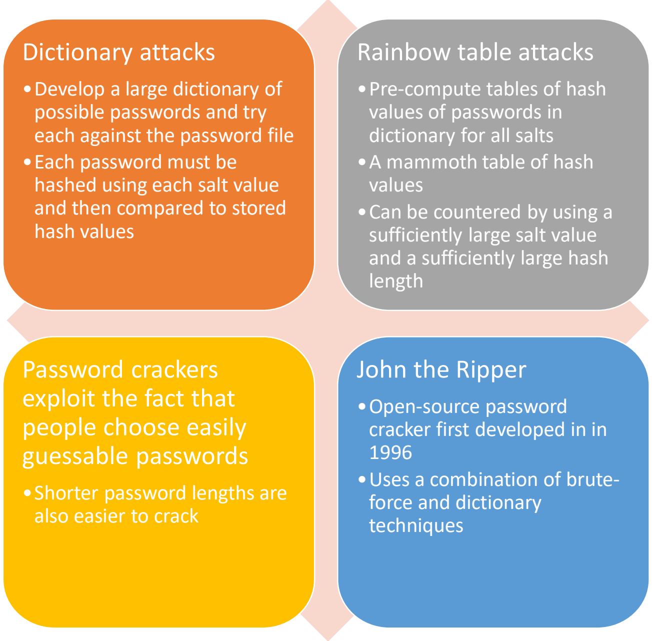
OpenBSD uses Blowfish block cipher-based hash algorithm called Bcrypt

- Most secure version of Unix hash/salt scheme
- Uses 128-bit salt to create 192-bit hash value

… and the story goes on …

22

Password cracking



Dictionary attacks

- Develop a large dictionary of possible passwords and try each against the password file
- Each password must be hashed using each salt value and then compared to stored hash values

Rainbow table attacks

- Pre-compute tables of hash values of passwords in dictionary for all salts
- A mammoth table of hash values
- Can be countered by using a sufficiently large salt value and a sufficiently large hash length

Password crackers exploit the fact that people choose easily guessable passwords

- Shorter password lengths are also easier to crack

John the Ripper

- Open-source password cracker first developed in in 1996
- Uses a combination of brute-force and dictionary techniques

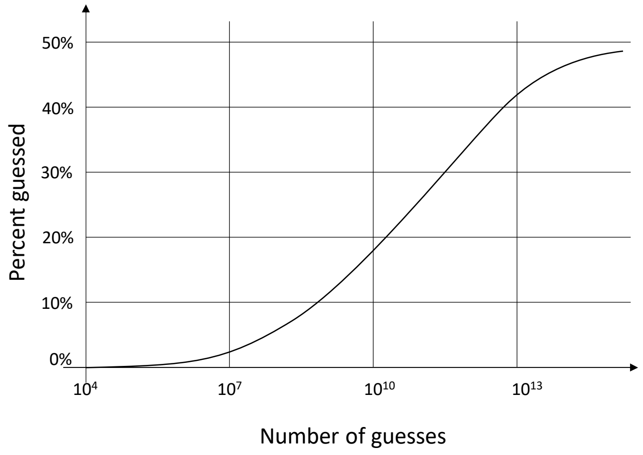
23

Modern approaches

- Complex password policy
  - Forcing users to pick stronger passwords
- However password-cracking techniques have also improved
  - The processing capacity available for password cracking has increased dramatically
  - The use of sophisticated algorithms to generate potential passwords
  - Studying examples and structures of actual passwords in use

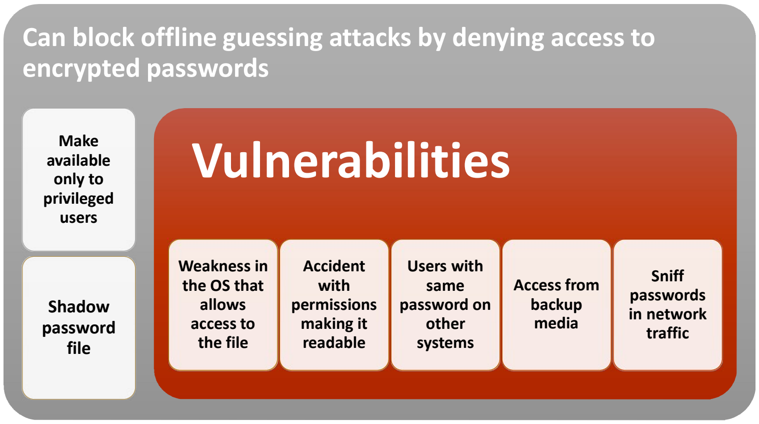
24

Percentage of passwords guess after a given number of guesses



25

Password file access control

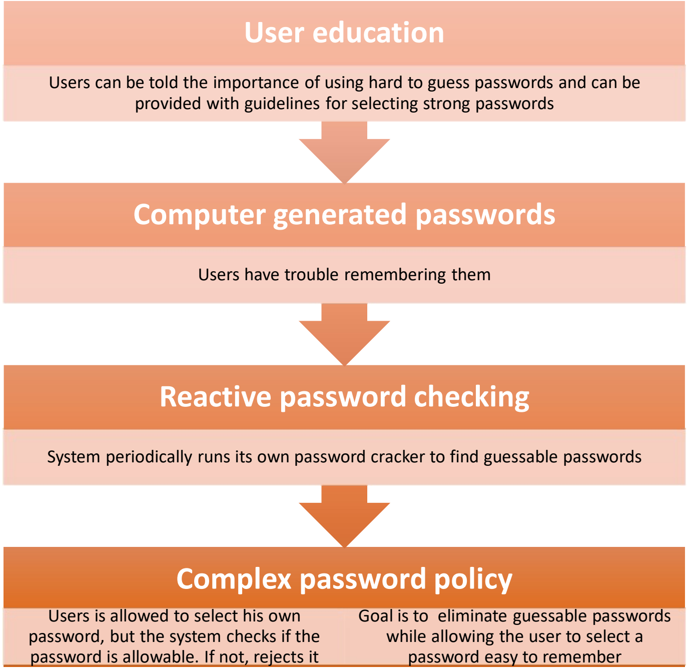


Can block offline guessing attacks by denying access to encrypted passwords

Make available only to privileged users
Shadow password file
Vulnerabilities
Weakness in the OS that allows access to the file
Accident with permissions making it readable
Users with same password on other systems
Access from backup media
Sniff passwords in network traffic

26

Password selection strategies



User education

Users can be told the importance of using hard to guess passwords and can be provided with guidelines for selecting strong passwords

Computer generated passwords

Users have trouble remembering them

Reactive password checking

System periodically runs its own password cracker to find guessable passwords

Complex password policy

Users is allowed to select his own password, but the system checks if the password is allowable. If not, rejects it

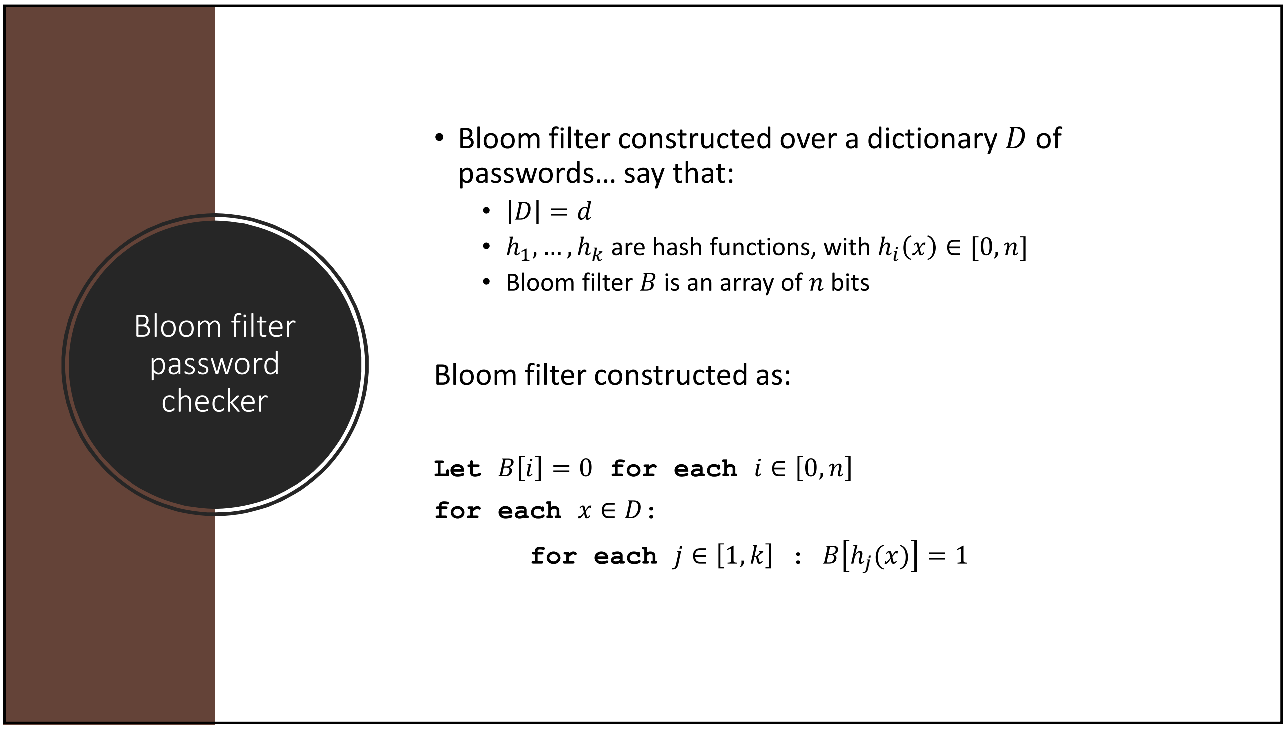
Goal is to eliminate guessable passwords while allowing the user to select a password easy to remember

27

Proactive password checking

- Rule enforcement
  - Specific rules that passwords must adhere to
- Password checker
  - Compile a large dictionary of “bad” passwords not to use
- But how to make a fast pswd check?
  - Bloom filter, used to implements password checkers
    - Exploits *k* hash functions to build tables of hash values
    - Check desired password against this table
    - It’s a probabilistic, efficient way to check if a password is in the dictionary of forbidden passwords

28



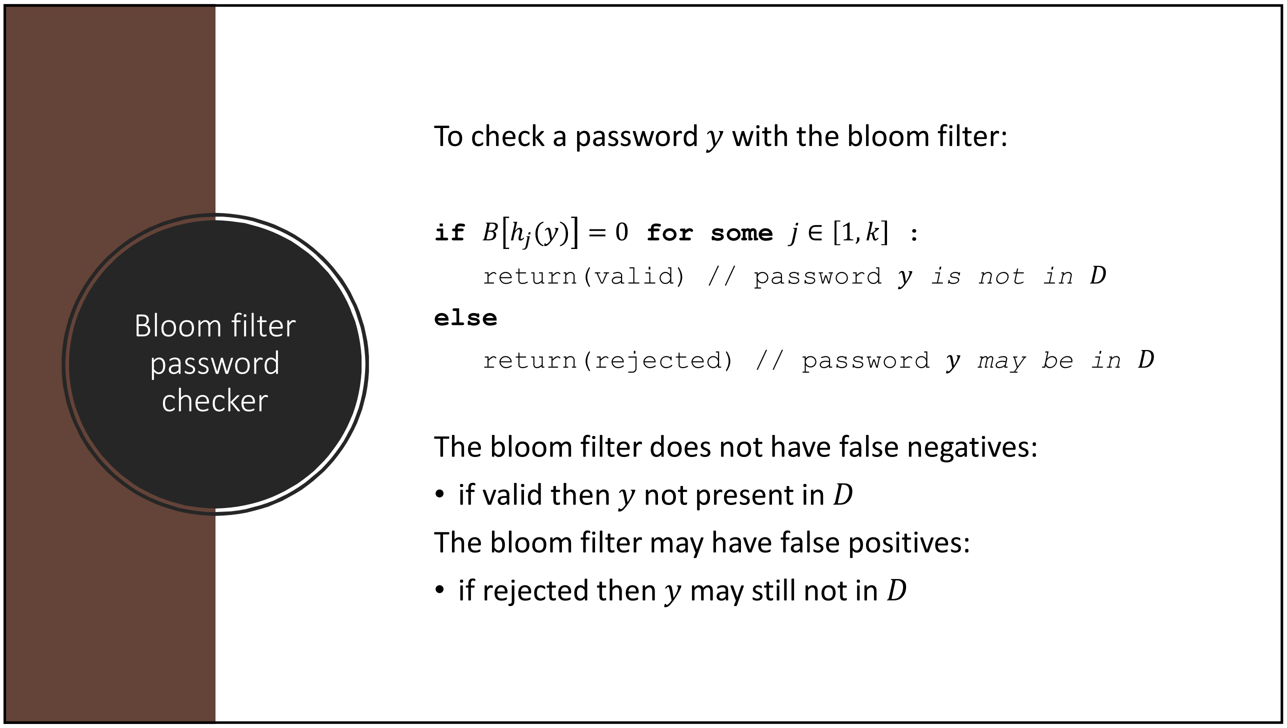
Bloom filter password checker

- Bloom filter constructed over a dictionary D of passwords… say that:
  - $|D| = d$
  - $h_1, \dots, h_k$ are hash functions, with $h_i(x) \in [0, n]$
  - Bloom filter B is an array of n bits

Bloom filter constructed as:

```
Let $B[i] = 0$ for each $i \in [0, n]$
for each $x \in D$:
    for each $j \in [1, k]$ : $B[h_j(x)] = 1$
```

29



Bloom filter password checker

To check a password y with the bloom filter:

```
if $B[h_j(y)] = 0$ for some $j \in [1, k]$ :
   return(valid) // password $y$ is not in $D$
else
   return(rejected) // password $y$ may be in $D$
```

The bloom filter does not have false negatives:

- if valid then y not present in D

The bloom filter may have false positives:

- if rejected then y may still not in D

30

Bloom filter password checker

Example: $D = \{Stefano, Paolo\}$

Say that:

$$h_1(Stefano) = 10;\ h_2(Stefano) = 7; h_3(Stefano) = 3$$
$$h_1(Paolo) = 7;\ h_2(Paolo) = 12; h_3(Paolo) = 1$$

Hence $B =$

0	1	2	3	4	5	6	7	8	9	10	11	12
0	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**

Now, consider passwords $Rita$ and $Sofia$, and that:

$$h_1(Rita) = 3;\ h_2(Rita) = 12; h_3(Rita) = 1$$
$$h_1(Sofia) = 2;\ h_2(Sofia) = 9; h_3(Sofia) = 7$$

Then:

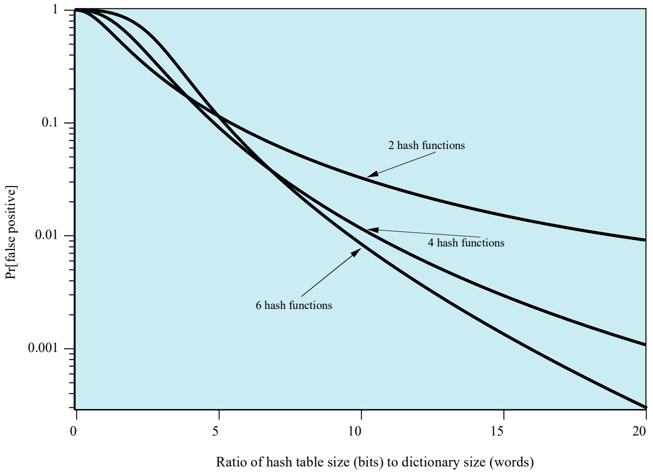
- $Rita$ is rejected
- $Sofia$ is valid

31

Bloom filter password checker

Need to find a proper configuration of the parameters (values of k, m, n), else the mechanism may be unusable

- it may give too many false positives



Example:
$d = 10^6$
$k = 6$
$n = 10 \cdot 10^6$
Filter takes about 1.2MBytes

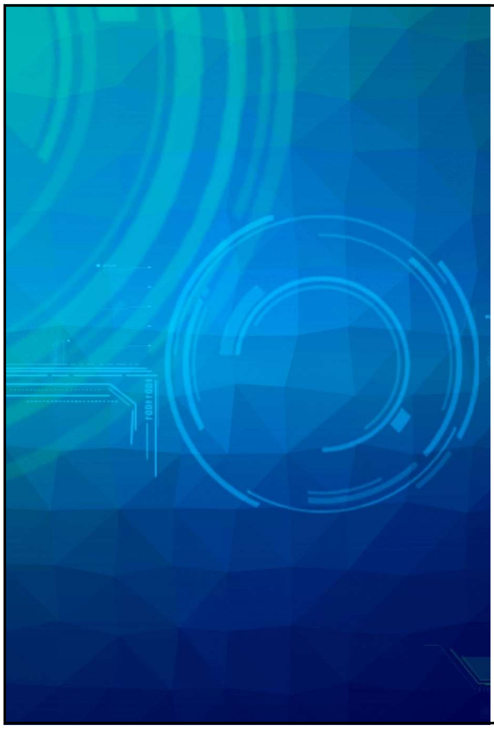
32

curiosity

- Several account (and passwords) breaches, even recently
- Want to know whether your account had been broken?

  https://haveibeenpwned.com/
- contains a DB of many breaches

34

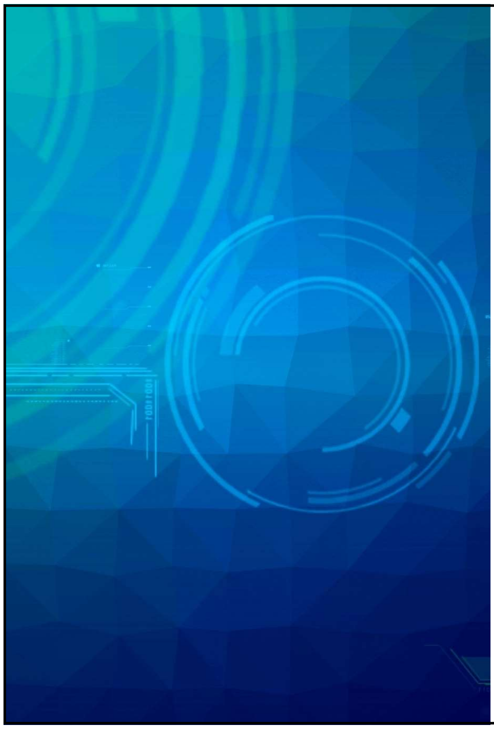


Question

COMMENT ABOUT THE SUITABILITY OF THE PASSWORDS:

a. Back#To#Black#
b. 09876
c. viaFermi3
d. ketchup
e. onafetS
f. S0LI3van7e

35



Question

?

ASSUME A SYSTEM THAT USE RANDOMLY-GENERATED PASSWORDS. PASSWORDS ARE 8 CHARACTERS LONG IN THE ALPHABET OF THE CAPITAL LETTERS.

WHAT SHOULD BE THE APPROPRIATE RANGE FOR THE PSEUDO-RANDOM NUMBER GENERATOR?

36

Token-based authentication

- Objects that a user possesses for the purpose of user authentication are called tokens:
  - Memory cards
  - Smart tokens/cards
- Applications / Features:
  - Electronic identity cards
  - Eid functions
  - Passwords authenticated connection establishment (PACE)

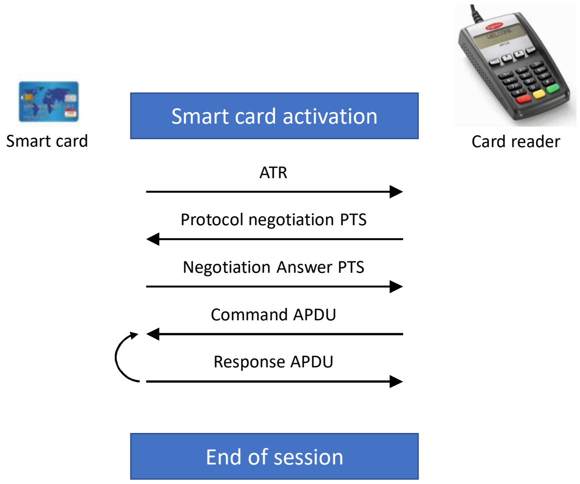
37

Types of cards used as tokens

Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Old bank/telephone card
Memory	Electronic memory inside	Prepaid phone card
Smart tokens Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

38

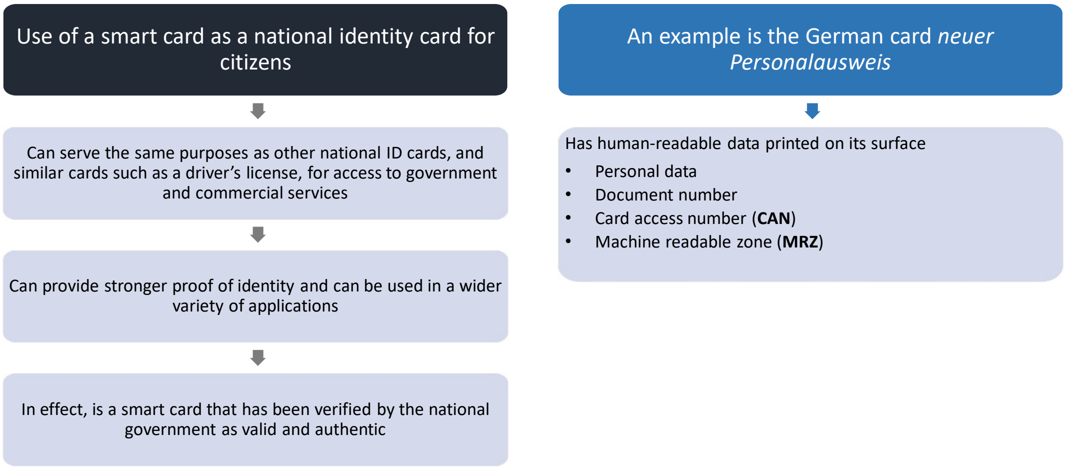
Smart cart / reader exchange



ATR: Anwser to reset
PTS: protocol type selection
APDU: Application protocol data unit

42

Electronic identity cards (eID)



Use of a smart card as a national identity card for citizens

Can serve the same purposes as other national ID cards, and similar cards such as a driver's license, for access to government and commercial services

Can provide stronger proof of identity and can be used in a wider variety of applications

In effect, is a smart card that has been verified by the national government as valid and authentic

An example is the German card *neuer Personalausweis*

Has human-readable data printed on its surface

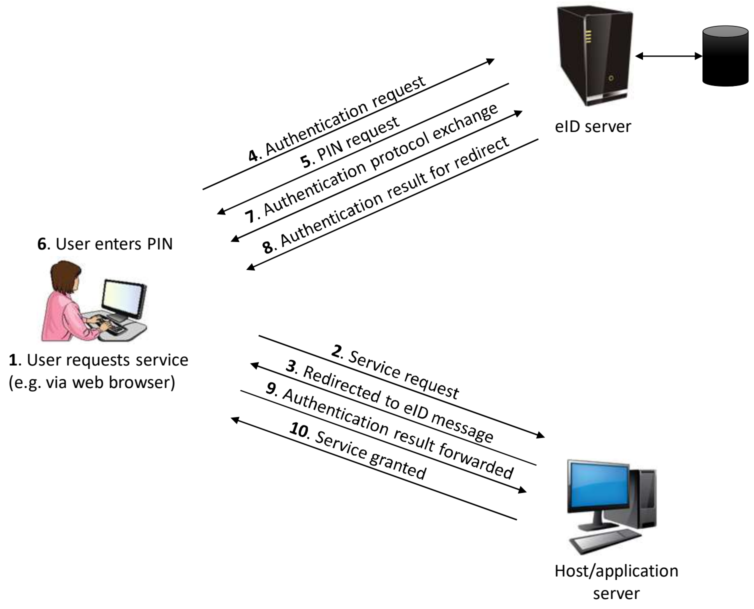
- Personal data
- Document number
- Card access number (**CAN**)
- Machine readable zone (**MRZ**)

Electronic functions for eID cards

Function	Purpose	PACE Password	Data	Users
ePass (mandatory)	Authorized offline inspection systems read the data.	CAN or MRZ	Face image; two fingerprint images (optional); MRZ data	Offline biometric identity verification reserved for government access
eID (activation optional)	Online applications read the data or access functions as authorized.	eID PIN	Family and given names; artistic name and doctoral degree: date and place of birth; address and community ID; expiration date	Identification; age verification; community ID verification; restricted identification (pseudonym); revocation query Offline inspection systems read the
	Offline inspection systems read the data and update the address and community ID.	CAN or MRZ		
eSign (certificate optional)	A certification authority installs the signature certificate online.	eID PIN	Signature key; X.509 certificate	Electronic signature creation
	Citizens make signature creation electronic signature with eSign PIN.	CAN		

CAN = card access number
MRZ = machine readable zone
PACE = password authenticated connection establishment
PIN = personal identification number

User authentication with eID


eID server
4. Authentication request
5. PIN request
7. Authentication protocol exchange
8. Authentication result for redirect
6. User enters PIN
1. User requests service (e.g. via web browser)
2. Service request
3. Redirected to eID message
9. Authentication result forwarded
10. Service granted
Host/application server

45

Password Authenticated Connection Establishment (PACE)

Ensures that the contactless RF chip in the eID card cannot be read without explicit access control

Online applications: the user enters the 6-digit PIN (which should only be known to the holder of the card)

For offline applications, either the MRZ printed on the back of the card or the CAN printed on the front is used

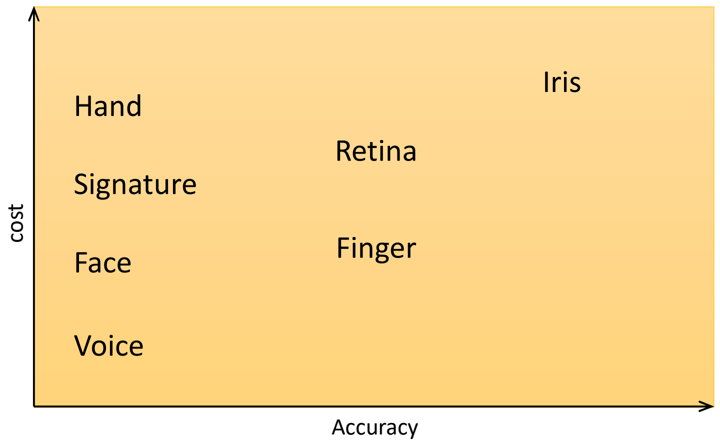
46

Biometric authentication

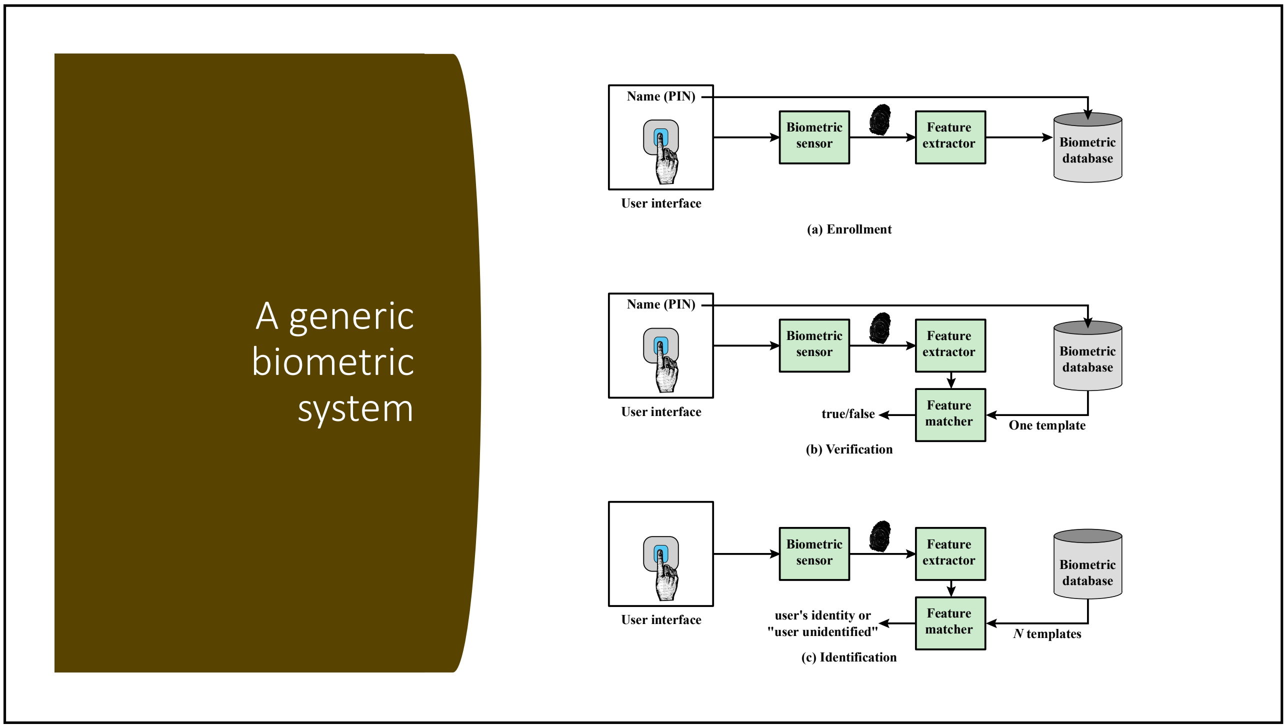
- Attempts to authenticate an individual based on unique physical characteristics
- Based on pattern recognition
- Is technically complex and expensive when compared to passwords and tokens
- Physical characteristics used include:
  - Facial characteristics
  - Fingerprints
  - Hand geometry
  - Retinal pattern
  - Iris
  - Signature
  - Voice

47

Cost vs accuracy of biometric characteristics in user authentication schemes



48



A generic biometric system

Name (PIN)

User interface

Biometric sensor

Feature extractor

Biometric database

(a) Enrollment

Name (PIN)

User interface

Biometric sensor

Feature extractor

Biometric database

Feature matcher

true/false

One template

(b) Verification

User interface

Biometric sensor

Feature extractor

Biometric database

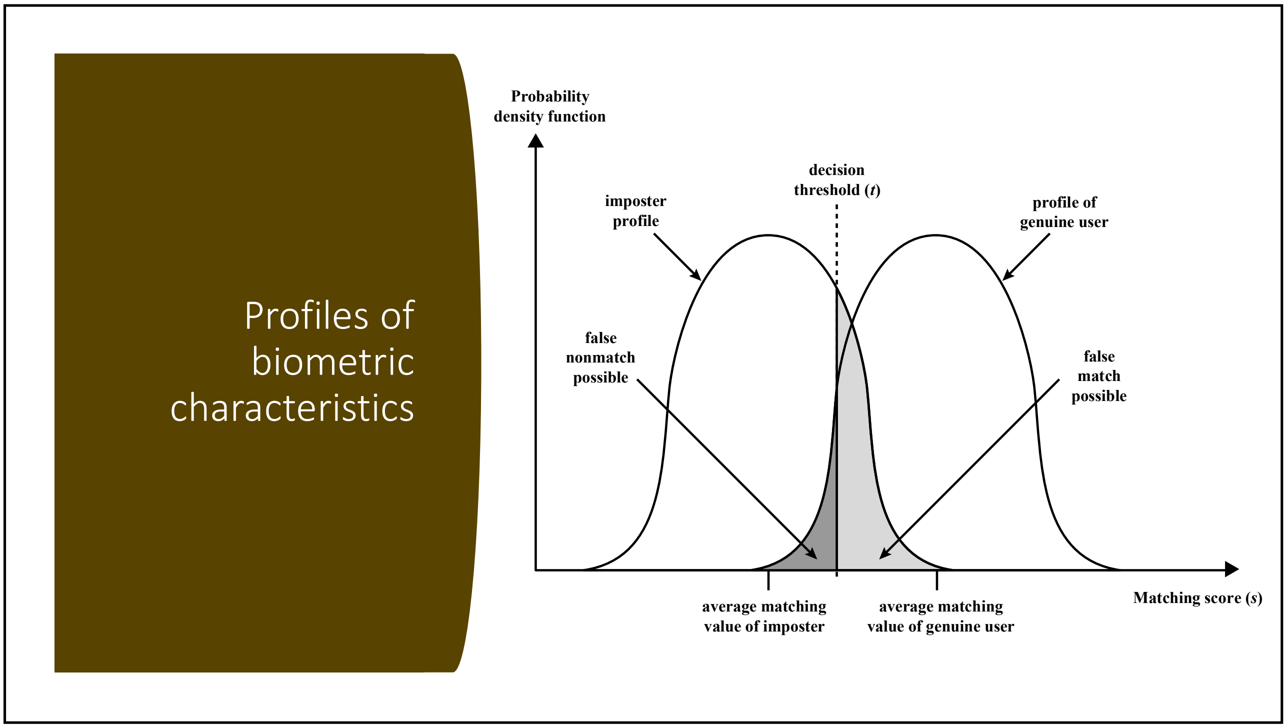
Feature matcher

user's identity or "user unidentified"

N templates

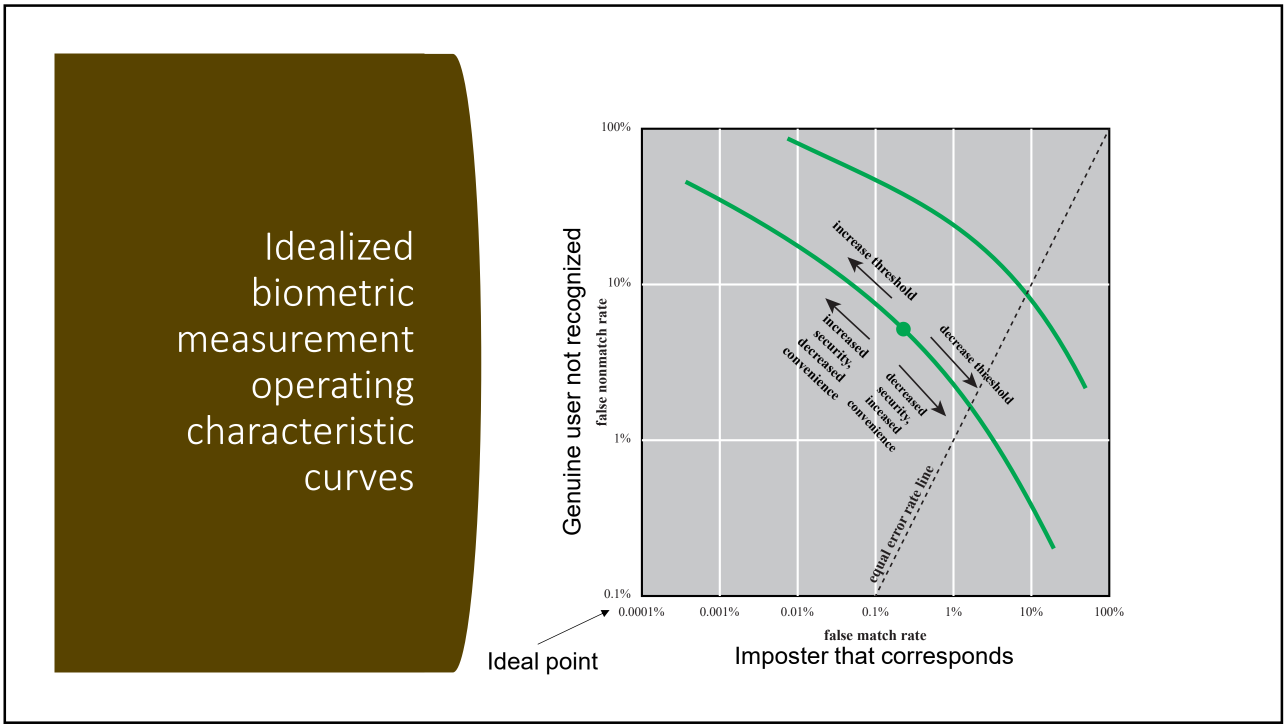
(c) Identification

49



Profiles of biometric characteristics

50

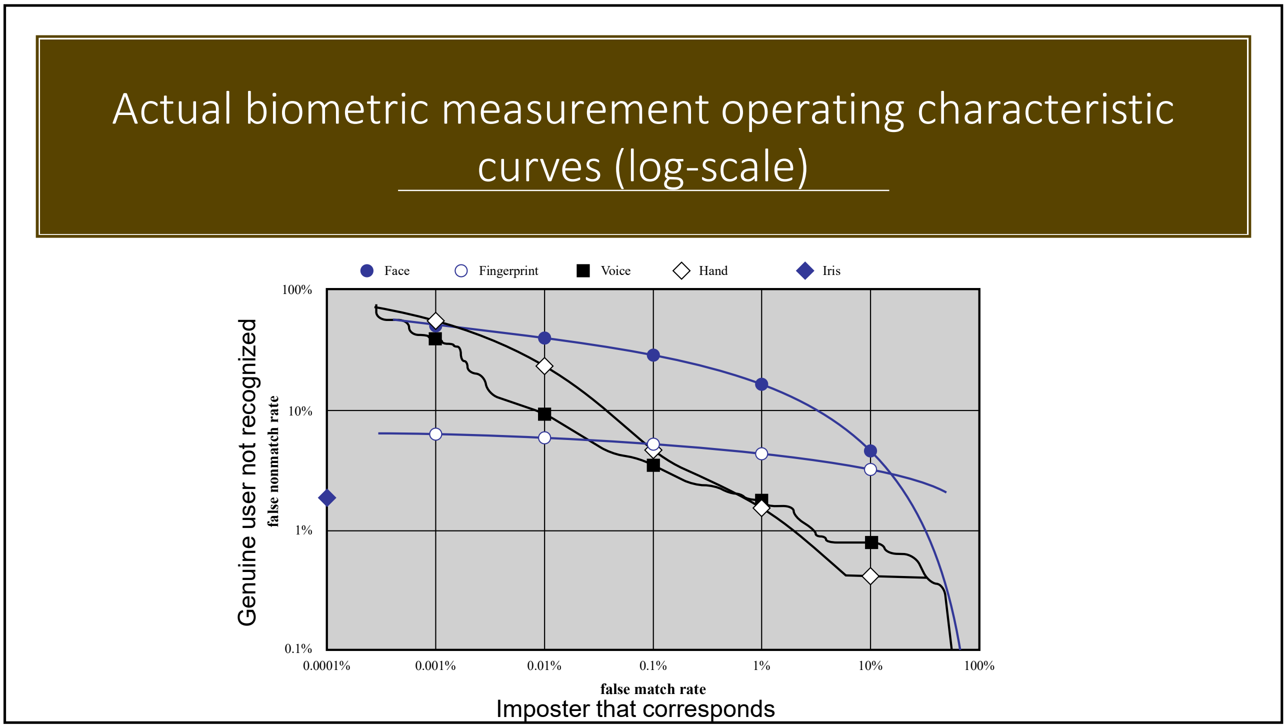


Idealized biometric measurement operating characteristic curves

Genuine user not recognized

Ideal point

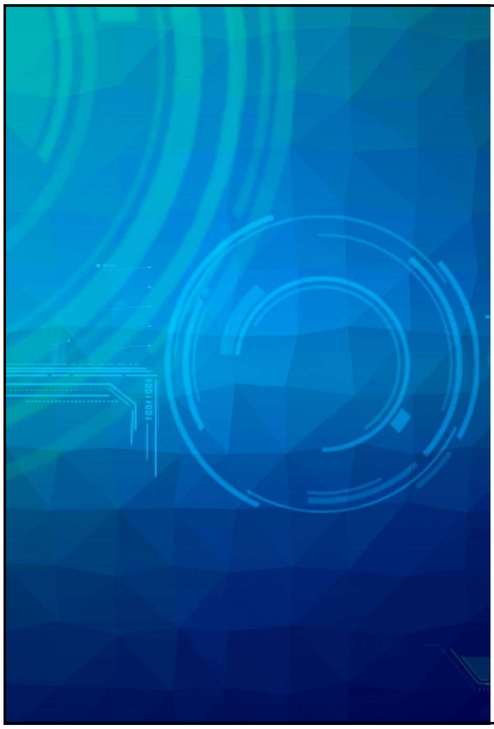
Imposter that corresponds



Actual biometric measurement operating characteristic curves (log-scale)

Genuine user not recognized

Imposter that corresponds



Review question

?

"JANE SPLIT A STONE INTO TWO PIECES, KEPT ONE FOR HERSELF, AND GAVE THE OTHER TO JASON. SO SHE SAID - WHEN YOU WILL SEND YOUR EMISSARY GIVE HIM THIS HALF OF THE STONE AND I WILL RECOGNIZE HIM."

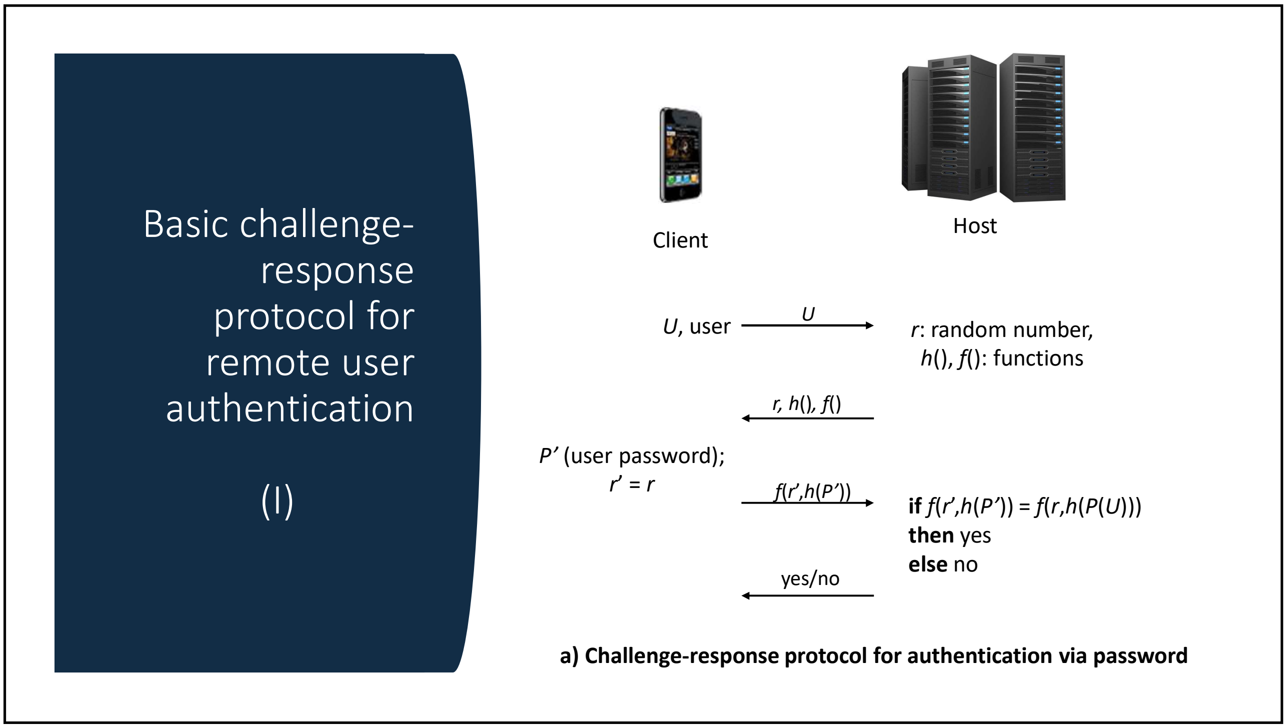
WHAT KIND OF AUTHENTICATION IS THIS?

53

Remote user authentication

- Authentication over a network, the Internet, or a communications link is more complex
- Additional security threats such as:
  - Eavesdropping, capturing a password, replaying an authentication sequence that has been observed
- Generally rely on some form of a challenge-response protocol to counter threats

54



Basic challenge-response protocol for remote user authentication

(I)

Client

Host

U, user $\xrightarrow{U}$

r: random number,
$h()$, $f()$: functions

$\xleftarrow{r,\ h(),\ f()}$

P' (user password);
$r' = r$

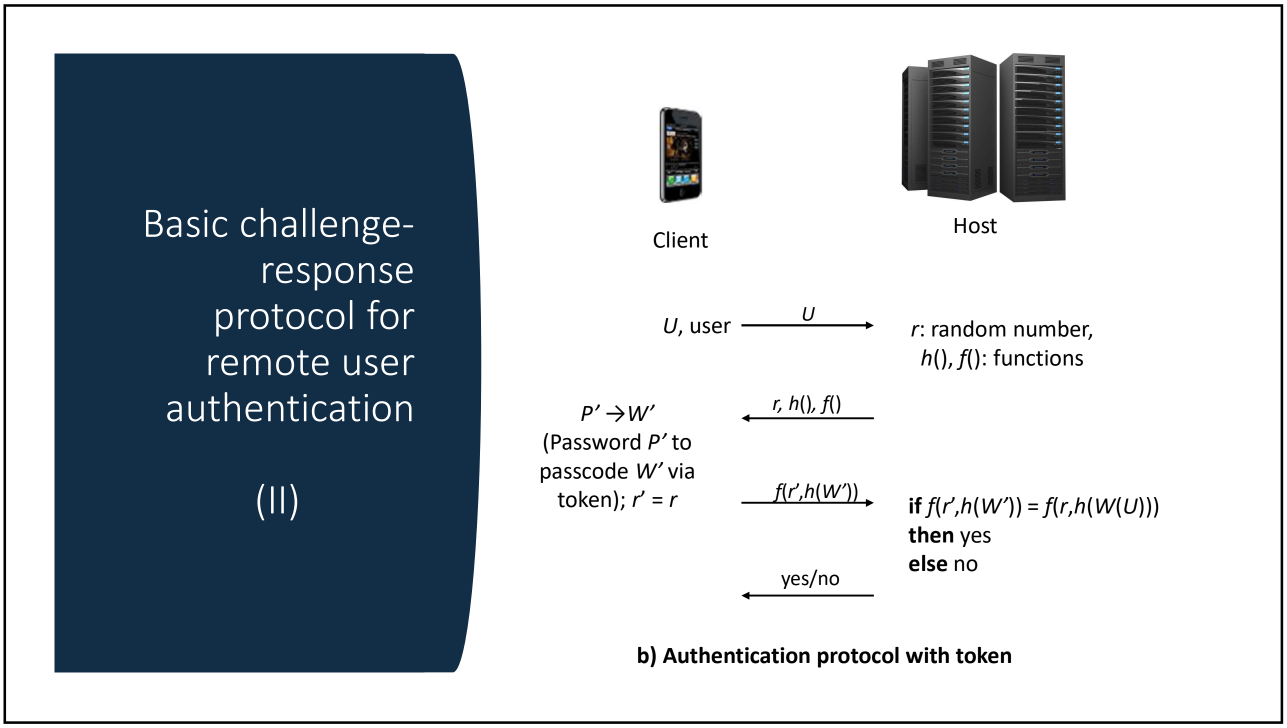
$\xrightarrow{f(r',h(P'))}$

if $f(r',h(P')) = f(r,h(P(U)))$
then yes
else no

$\xleftarrow{\text{yes/no}}$

a) Challenge-response protocol for authentication via password

55



Basic challenge-response protocol for remote user authentication

(II)

Client

Host

U, user $\xrightarrow{U}$

r: random number,
$h()$, $f()$: functions

$\xleftarrow{r,\ h(),\ f()}$

$P' \rightarrow W'$
(Password P' to passcode W' via token); $r' = r$

$\xrightarrow{f(r',h(W'))}$

if $f(r',h(W')) = f(r,h(W(U)))$
then yes
else no

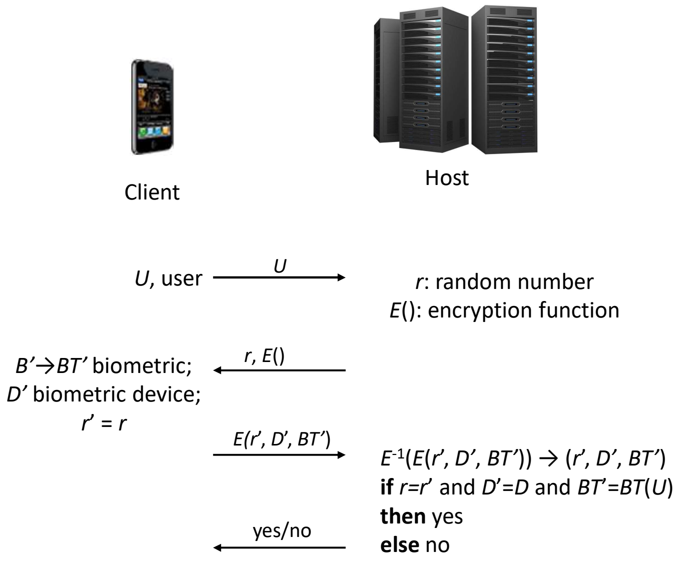
$\xleftarrow{\text{yes/no}}$

b) Authentication protocol with token

56

Basic challenge-response protocol for remote user authentication

(III)



Client

Host

U, user → *U* →

r: random number
E(): encryption function

← *r*, *E*() ←

B′→*BT′* biometric;
D′ biometric device;
r′ = *r*

→ *E*(*r′*, *D′*, *BT′*) →

$E^{-1}(E(r', D', BT')) \rightarrow (r', D', BT')$
if *r*=*r′* and *D′*=*D* and *BT′*=*BT*(*U*)
then yes
else no

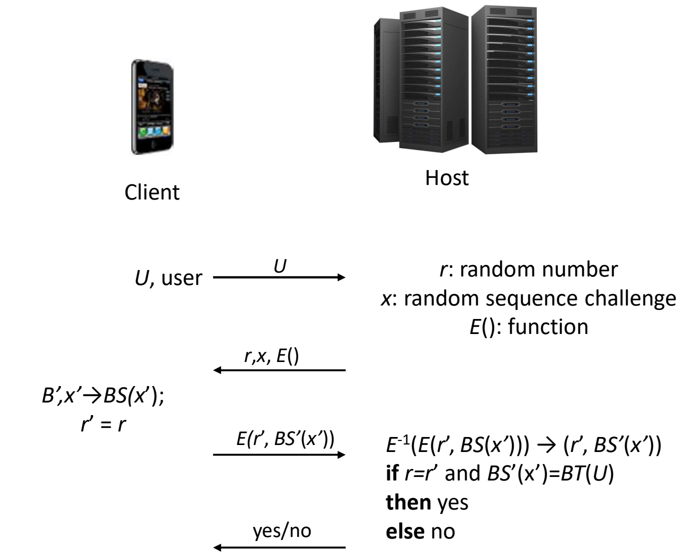
← yes/no ←

c) Authentication protocol with static biometric

57

Basic challenge-response protocol for remote user authentication

(IV)



Client

Host

U, user → *U* →

r: random number
x: random sequence challenge
E(): function

← *r*,*x*, *E*() ←

B′,*x′*→*BS*(*x′*);
r′ = *r*

→ *E*(*r′*, *BS′*(*x′*)) →

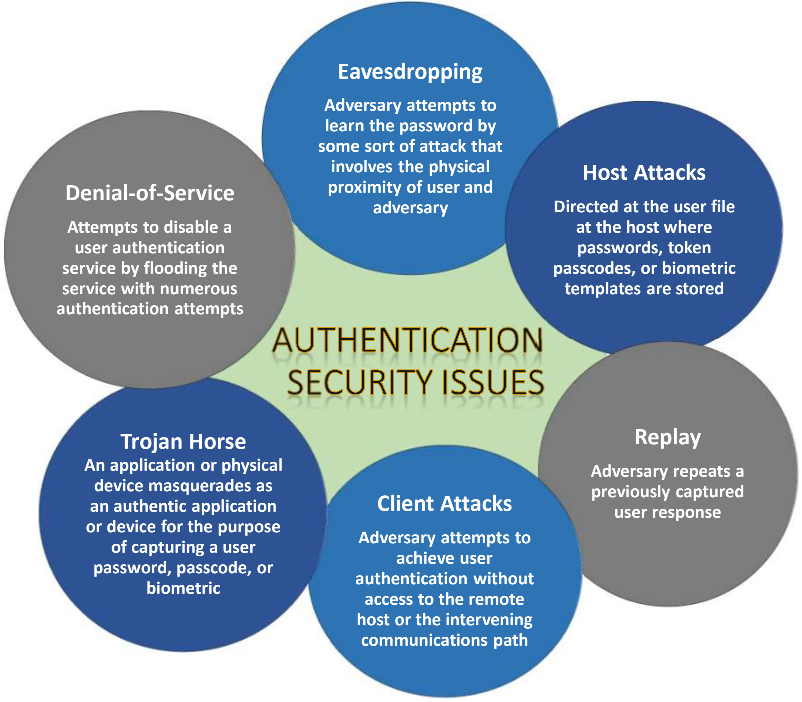
$E^{-1}(E(r', BS(x'))) \rightarrow (r', BS'(x'))$
if *r*=*r′* and *BS′*(x′)=*BT*(*U*)
then yes
else no

← yes/no ←

d) Authentication protocol with dynamic biometric

58

Authentication security issues


AUTHENTICATION SECURITY ISSUES
Eavesdropping
Adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary
Host Attacks
Directed at the user file at the host where passwords, token passcodes, or biometric templates are stored
Replay
Adversary repeats a previously captured user response
Client Attacks
Adversary attempts to achieve user authentication without access to the remote host or the intervening communications path
Trojan Horse
An application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric
Denial-of-Service
Attempts to disable a user authentication service by flooding the service with numerous authentication attempts

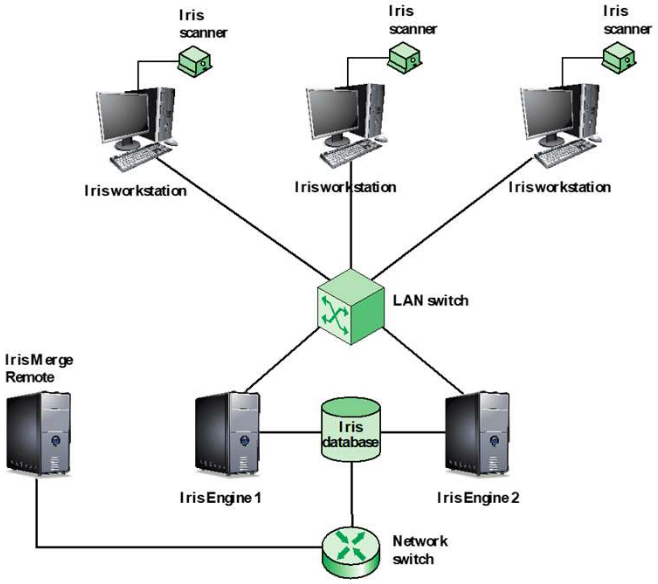
59

Some potential attacks, susceptible authenticators and typical defenses

Attacks	Authenticators	Examples	Typical Defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
"	Token	Exhaustive search	Large entropy; limited attempts; theft of object requires presence
"	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
"	Token	Passcode theft	Same as password; 1-time passcode
"	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
"	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
"	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
"	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
"	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token

60

General iris scan site architecture for UAE system

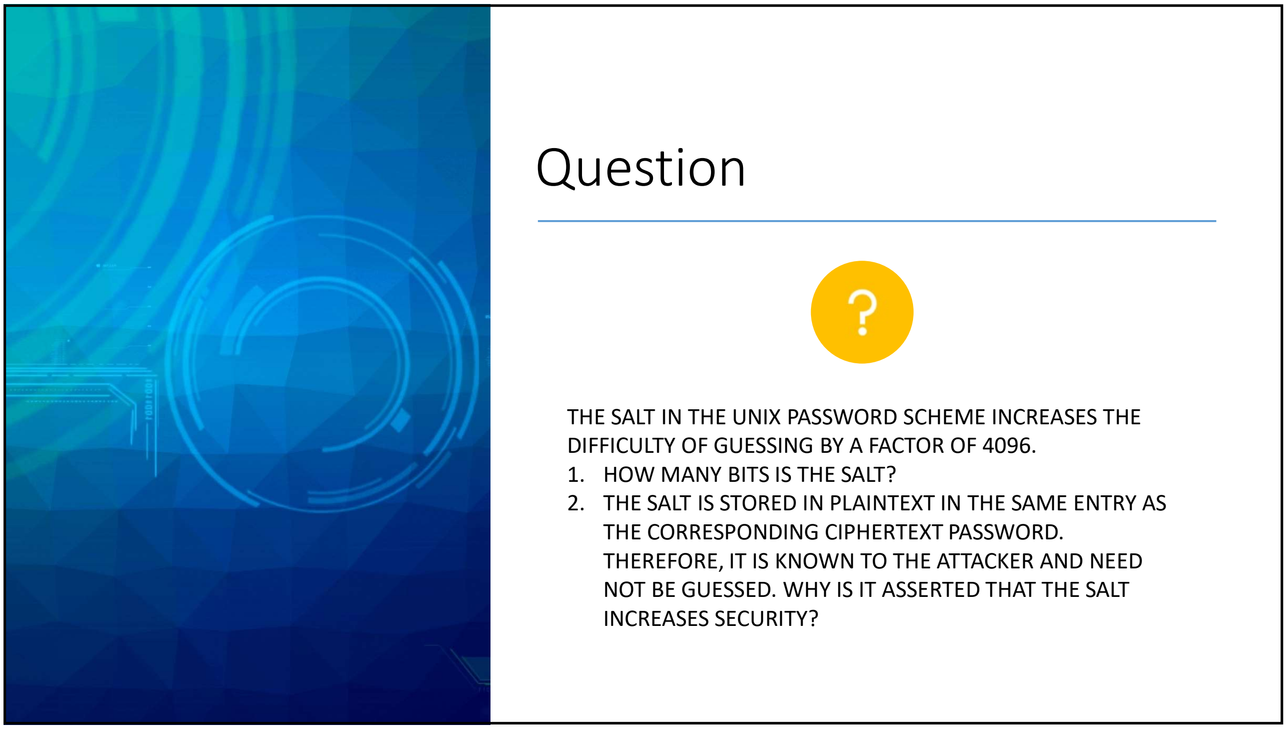


61

Summary

- Digital user authentication principles
  - A model for digital user authentication
  - Means of authentication
  - Risk assessment for user authentication
- Password-based authentication
  - The vulnerability of passwords
  - The use of hashed passwords
  - Password cracking of user-chosen passwords
  - Password file access control
  - Password selection strategies
- Token-based authentication
  - Memory cards
  - Smart cards
  - Electronic identity cards
- Biometric authentication
  - Physical characteristics used in biometric applications
  - Operation of a biometric authentication system
  - Biometric accuracy
- Remote user authentication
  - Password protocol
  - Token protocol
  - Static biometric protocol
  - Dynamic biometric protocol
- Security issues for user authentication

63



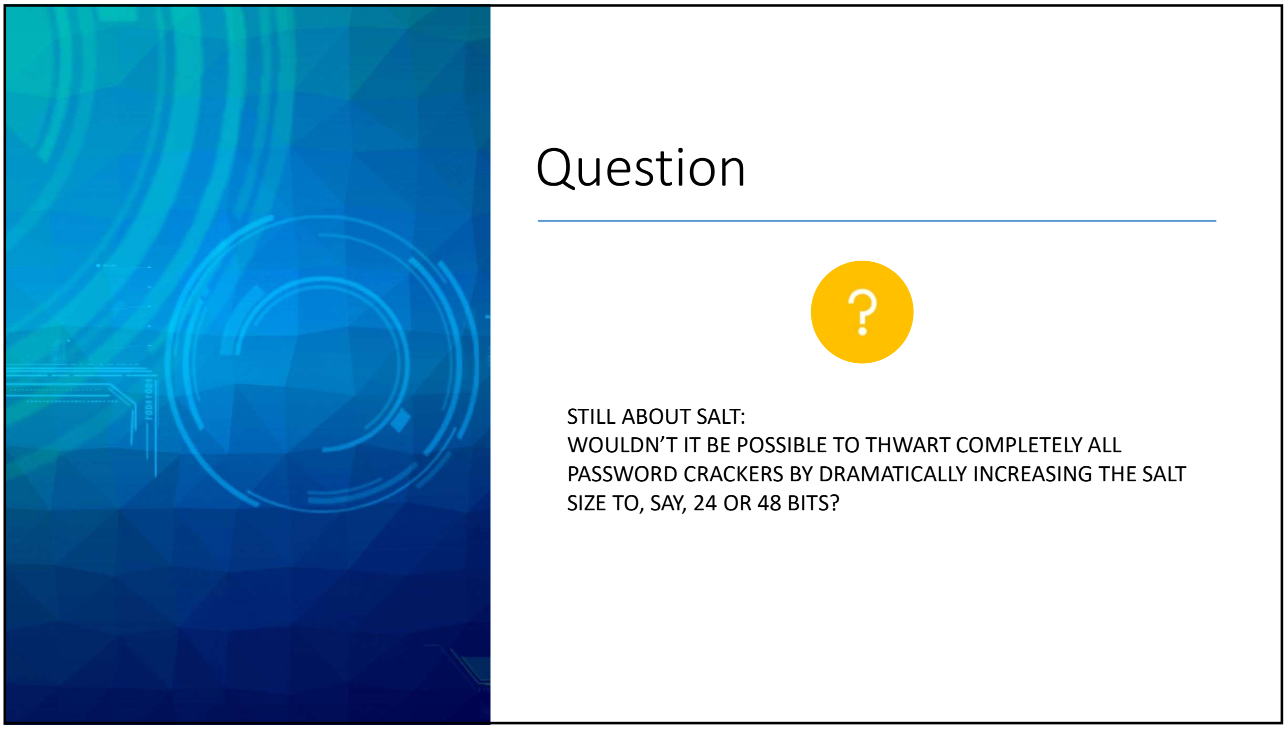
Question

?

THE SALT IN THE UNIX PASSWORD SCHEME INCREASES THE DIFFICULTY OF GUESSING BY A FACTOR OF 4096.

1. HOW MANY BITS IS THE SALT?
2. THE SALT IS STORED IN PLAINTEXT IN THE SAME ENTRY AS THE CORRESPONDING CIPHERTEXT PASSWORD. THEREFORE, IT IS KNOWN TO THE ATTACKER AND NEED NOT BE GUESSED. WHY IS IT ASSERTED THAT THE SALT INCREASES SECURITY?

64



Question

?

STILL ABOUT SALT:
WOULDN’T IT BE POSSIBLE TO THWART COMPLETELY ALL PASSWORD CRACKERS BY DRAMATICALLY INCREASING THE SALT SIZE TO, SAY, 24 OR 48 BITS?

65

Exercise 1

A system requests the users to choose passwords at least 8 and at most 10 characters long and that are chosen within an alphabet of 40 symbols. The system combines the passwords with a 10 bits salt to produce a hash code for each password that is stores, along with the salt, in the password file. The hash is 128 bits long.

Assume also that testing a password takes 0.1 milliseconds.

1. An adversary that got access to the password file performs a brute force attack to crack the password of a specific user. How long it will take in the worst case and on average?
2. Assume the system has 10,000 users, and the adversary is interested in breaking the password of one arbitrary user (any user would be OK to get in). How long it will take on average?
   How long it would take if no salt was used?

66

Solution 1.1

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

1) Brute force attack to crack the password of a specific user.

How long it will take in the worst case and on average?

The number of different passwords are: ___________

Each password is combined with a salt randomly chosen in __________ combinations

Thus the number of combinations to be generated is: ________________________

In the worst case it will take: _______________________

On average it will take: ________________

67

Solution 1.1

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

68

Solution 1.2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

2) breaking the password of an arbitrary user

How long it will take in the worst case and on average?

How long it would take if no salt was used?

The number of different passwords are : ________________________

Each password is combined with a salt randomly chosen in _________ combinations.

On average ________________________________

Thus it will take on average: ________________________

If no salt was used it will take on average: ________________________

69

Solution 1.2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 128 bits.
- Testing a password takes 0,1 milliseconds.
- 10,000 users in the system

70

Exercise 2

A system requests the users to choose passwords at least 8 and at most 10 characters long and that are chosen within an alphabet of 40 symbols. The system combines the passwords with a 10 bits salt to produce a hash code for each password that is stores, along with the salt, in the password file. The hash is **32 bits long**.
Assume also that testing a password takes 0.1 milliseconds.

An adversary that got access to the password file performs a brute force attack to crack the password of a specific user. How long it will take in the worst case and on average?

71

Solution 2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 32 bits.
- Testing a password takes 0,1 milliseconds.

Brute force attack to crack the password of a specific user.

How long it will take on average?

The number of different passwords are: _________________________

Each password is combined with a salt randomly chosen in ________ combinations

The number of different hashes in which the password is encoded is: ________

On average, the number of combinations to be generated is: _______________________

On average it will take: ________________________

72

Solution 2

- Passwords from 8 to 10 chars
- Alphabet of 40 symbols.
- Salt of 10 bits and hash of 32 bits.
- Testing a password takes 0,1 milliseconds.

73