BÀI 6.
XÁC THỰC DANH TÍNH

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

Nội dung

- Khái niệm chung
- Xác thực dựa trên mật khẩu
- Các giao thức xác thực dựa trên mật khẩu
- Giao thức zero-knowledge
- Giới thiệu một số phương pháp xác thực khác

Xác thực danh tính là gì?

- Danh tính (Identifier) ~ Định danh
- Xác thực danh tính: Ai đang truy cập/trao đổi dữ liệu?

→ Các bên truy cập/trao đổi dữ liệu cần chứng minh được liên kết về chủ thể ở thế giới thực với danh tính

- Các phương pháp xác thực chính:
  - Cái chủ thể biết (What the entity knows)
  - Cái chủ thể có (What the entity has)
  - Chủ thể là gì (What the entity is)
  - Vị trí của chủ thể (Where the entity is)
- Xác thực đa yếu tố: sử dụng >1 yếu tố xác thực

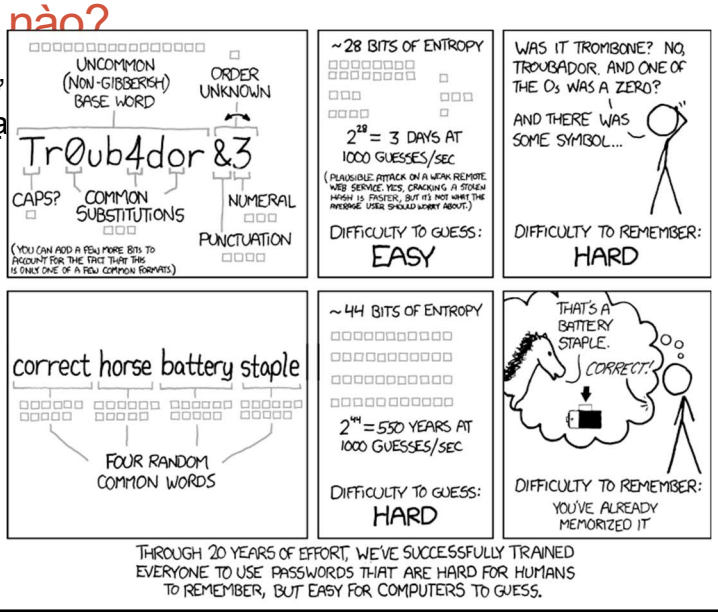
1. XÁC THỰC BẰNG MẬT KHẨU

Xác thực bằng mật khẩu

- Mật khẩu: một chuỗi ký tự hoặc một nhóm từ được sử dụng để xác thực danh tính của thực thể nào đó
  - Thực thể(Entity) cần xác thực (**người dùng**, thiết bị, ứng dụng...)
  - Người thẩm tra(Verifier): kiểm tra tính hợp lệ của mật khẩu
- Các lỗ hổng làm giảm sự tin cậy:
  - Lưu trữ mật khẩu trong CSDL không an toàn
  - Truyền mật khẩu trên kênh không an toàn
  - Người dùng không cẩn trọng

Người dùng có thể bị đánh cắp tài khoản như thế nào?

- Con người cho là "mạ



Người dùng có thể bị đánh cắp tài khoản như thế nào?

- Vì vậy, người dùng thường dùng lại mật khẩu cho các tài khoản khác nhau
  - Với hy vọng sẽ không xảy ra điều gì tồi tệ
- Khi mật khẩu của 1 tài khoản nào đó bị lộ?
  - Kẻ tấn công có mật khẩu của người dùng
  - Và đăng nhập vào các tài khoản khác
- Thực tế: Hacker đã thử tấn công đánh cắp các tài khoản vận hành mạng lưới điện của Mỹ theo hướng tiếp cận này
  - Gửi email giả mạo chia sẻ tài liệu Dropbox
  - Tấn công vào website có yêu cầu xác thực người dùng

Không đổ lỗi cho người dùng

- Thông thường, chúng ta thường đổ lỗi cho người dùng khi họ sơ ý bị kẻ tấn công khai thác
- Chúng ta cần xây dựng hệ thống có khả năng hỗ trợ người dùng không hành động sai
- Ví dụ: thư giả mạo (phising email)

Giải pháp cho người dùng

- Phần mềm quản lý mật khẩu
  - Ví dụ: StickyPassword Free, RoboForm,…
  - Có thể tạo ra các mật khẩu "mạnh" và quản lý tài khoản sử dụng mật khẩu này
  - Người dùng chỉ cần nhớ 1 mật khẩu (Master Password) để mở "kho mật khẩu"
- Thẻ xác thực 2 yếu tố U2F Security Keys
  - Người dùng cần kết nối thẻ này với máy tính khi đăng nhập
  - Có khả năng giảm thiểu nguy cơ bị tấn công phishing

- Kích hoạt tùy chọn xác thực hai yếu tố (2FA) trên các hệ thống dịch vụ

Tấn công vào hệ xác thực bằng mật khẩu

- Tấn công thụ động: nghe lén, quan sát quá trình nhập mật khẩu
  - Nhìn trộm
  - Sử dụng chương trình key logging
  - Tấn công kênh bên
  - Chặn bắt gói tin
- Tấn công chủ động:
  - Đoán và thử
  - Giải mạo chương trình cung cấp dịch vụ (server)
  - Giả mạo chương trình khách (client)
  - Tấn công man-in-the-middle
  - Tấn công vào máy chủ vật lý cung cấp dịch vụ

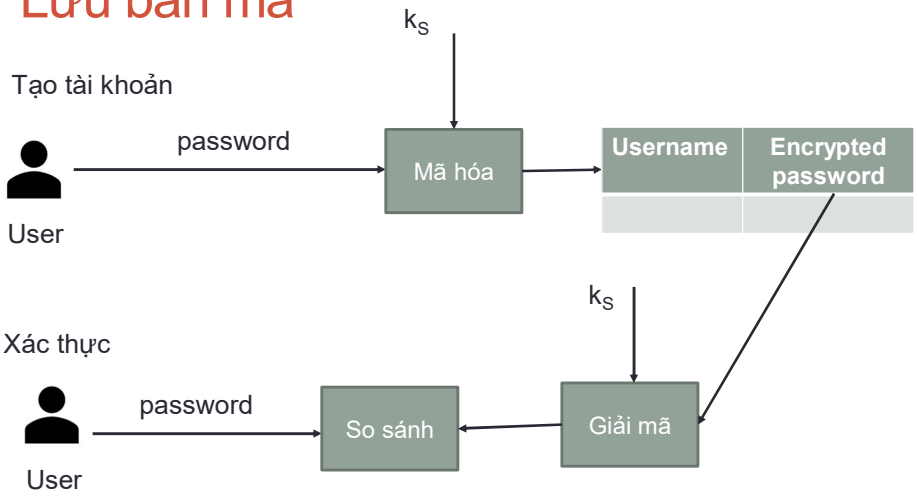
Tấn công đoán và thử

- Cách thức: dò thử lần lượt các mật khẩu, quan sát kết quả xác thực hệ thống trả lại
- Đặc điểm:
  - Tương tác trực tiếp với hệ xác thực
  - Có thể thử trên 1 hoặc đồng thời nhiều tài khoản
- Xác suất tấn công thành công: $P \geq (T \times G)/N$
  - G: Tốc độ kẻ tấn công dò thử
  - T: Thời gian kẻ tấn công dò thử
  - N: Số mật khẩu hệ thống có thể tạo ra
- Giảm thiểu:
  - Tăng độ dài của mật khẩu
  - Quy định số lần thử xác thực tối đa trong một khoảng thời gian

Tấn công lợi dụng lỗ hổng lưu trữ

- Lưu mật khẩu dưới dạng rõ:
  - Nguy cơ mất an toàn cao nhất
- Lưu mật khẩu dưới dạng bản mã:
  - An toàn khi sử dụng hệ mật mã tốt, bảo vệ khóa giải mã an toàn
  - Hạn chế: cần thao tác giải mã bất cứ khi nào cần xác thực
- Lưu mật khẩu dưới dạng mã băm:
  - Chi phí thấp hơn
  - Hạn chế: nguy cơ bị tấn công dò đoán dựa trên từ điển. Có thể hạn chế bằng cách đưa thêm “salt” vào mật khẩu trước khi băm

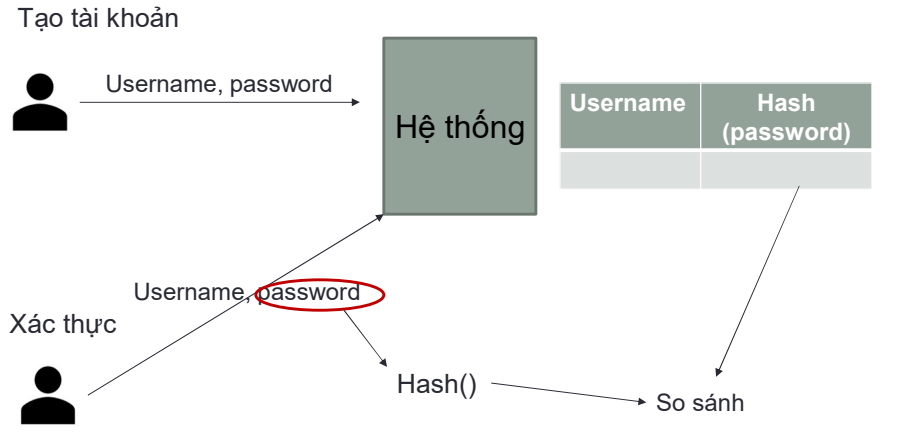
Lưu bản mã



13

13

Lưu mã băm mật khẩu



14

14

Tấn công từ điển/Tấn công cầu vồng

- Dictionary attack/Rainbow attack

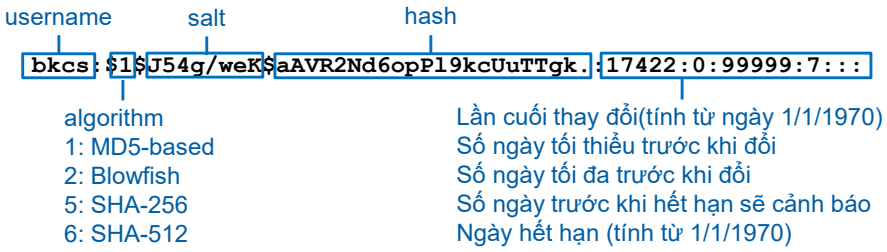
password	Hash(password)
Abcdef	H1
123456	H2
Password	H3
Guesme	..
Iloveyou	..
Qwerty	..

- Giảm thiểu nguy cơ: Hash(Password, Salt)
  - Yêu cầu sử dụng salt là gì?

15

Băm mật khẩu với "salt"

- Lưu trữ [salt,Hash(password || salt)
- Ví dụ: Hệ điều hành Linux

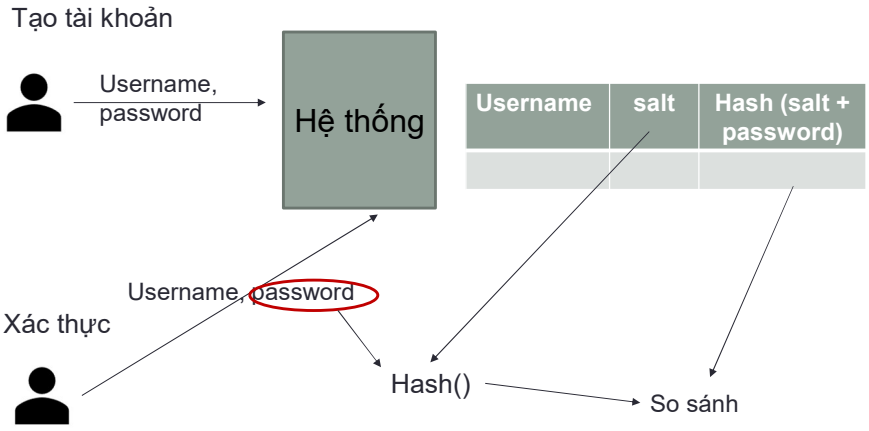


- Lưu trữ trong CSDL

username	salt	hash
levn	iU9KjTeD	5myyo4W7zppTOEdVUeP8/E6Km…
tungbt	r.PhJ0HG	Y.xOpTBqJbWpc3f0uri.g8ErCu4wliUGq

16

Băm cùng salt



17

Tấn công từ điển

Không dùng salt

password	Hash(password)
Abcdef	H1
123456	H2
Password	H3
Guesme	..
Iloveyou	..
Qwerty	..

Sử dụng bảng băm có sẵn

Có dùng salt

password	salt	Hash(salt + password)
Abcdef	X1	
123456	X2	
Password	X3	
Guesme	X4	
Iloveyou	X5	
Qwerty	X6	

- Phải băm lại
- Mỗi lần băm chỉ xác định được mật khẩu của tối đa 1 tài khoản

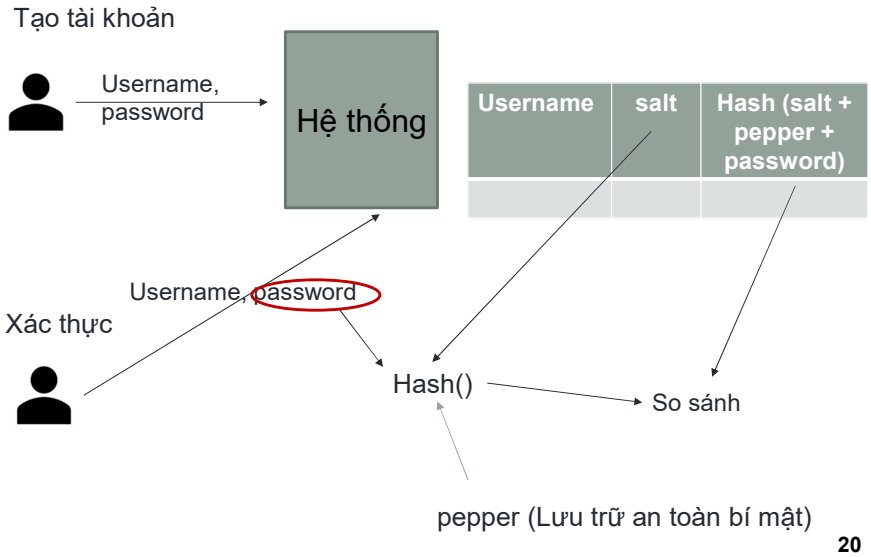
18

Băm mật khẩu với "salt" – Nâng cao an toàn

- Kẻ tấn công có thể tạo ra từ điển mới với các giá trị "salt"
- Băm nhiều lần: hash(hash(…..hash(password || salt)))))
  - Mục đích: làm chậm thời gian tính toán giá trị xác thực → làm chậm thời gian tấn công dò tìm…
  - …nhưng kẻ tấn công có thể kiên nhẫn hơn nữa tạo ra từ điển mới
- Băm mật khẩu với một giá trị "pepper" bí mật
  - Mục đích: ngăn chặn kẻ tấn công tạo ra từ điển mới
- Sử dụng một trong thuật toán bcrypt, scrypt, PBKDF2 thay cho các hàm băm thông thường

19

Băm cùng salt + pepper



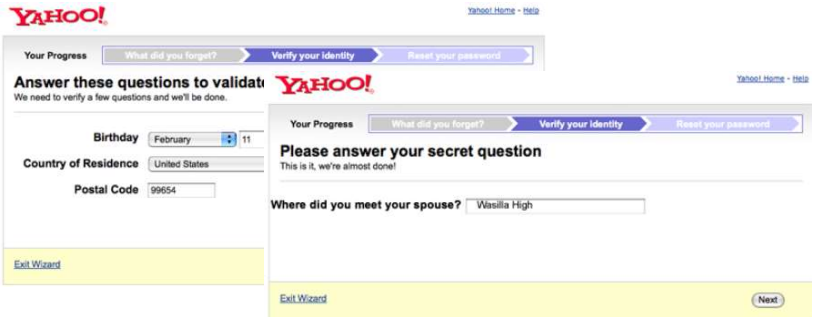
20

Khôi phục mật khẩu

- Làm thế nào để người dùng có thể khôi phục mật khẩu khi họ quên?
  - Gửi trực tiếp qua email
  - Reset qua email
  - Câu hỏi bí mật
  - Sử dụng tin nhắn SMS
  - ...
- Cơ chế khôi phục không được dễ dàng hơn cơ chế xác thực để kẻ tấn công vượt qua

Sử dụng câu hỏi bí mật còn an toàn?

- Năm 2008, ứng viên Phó Tổng thống Hoa Kỳ Sarah Palin bị đánh cắp tài khoản Yahoo Mail



- Năm 2012, ứng viên Tổng thống Mitt Romney bị đánh cắp tài khoản Hotmail

Một số chính sách sử dụng mật khẩu

- Mục đích: tăng cường an toàn cho hệ xác thực dựa trên mật khẩu
- Quy định độ dài tối thiểu
- Quy định các ký tự bắt buộc phải sử dụng
- Thay đổi mật khẩu định kỳ
- Hạn chế sử dụng lại mật khẩu cũ trong một khoảng thời gian nhất định
- Hạn chế số lần thử xác thực
- Tăng thời gian chờ thử xác thực lại
- Yêu cầu đổi mật khẩu sau lần đăng nhập đầu tiên
- Tuy nhiên, luôn phải cân nhắc sự trả giá cho tính tiện lợi

23

2. MỘT SỐ GIAO THỨC XÁC THỰC

24

Giao thức PAP

- Password Authentication Protcol
- Được sử dụng trong giao thức mạng PPP trước đây
- Nội dung:

(1) U → S: ID || Password

(2) Server kiểm tra trong CSDL

S → U: ACK/NAK

- Không an toàn

25

Xác thực 1 chiều dựa trên hệ mật mã KĐX

- Giả sử 2 bên đã trao đổi một giá trị khóa bí mật K_S(Có thể là mật khẩu)

(1) U → S: Request

(2) S → U: Challenge

(3) U → S: f(Pass, Challenge)

Hàm f: có thể là các hàm mã hóa KĐX, hàm băm

Pass : mật khẩu

- **Bài tập:** Phân tích các điểm yếu của sơ đồ này

26

Xác thực 1 chiều dựa trên hệ mật mã KCK

ISO/IEC 9798-3 / FIPS-196

(1) A → B: Request
(2) B → A: TokenID || N_B
(3) A → B: TokenID || $Cert_A$ || TokenAB

TokenID: chứa thông tin của phiên
TokenAB = N_A || N_B || $E(K_{RA}, N_A || N_B)$

Chữ ký số

27

Giao thức CHAP

- Challenge Handshake Authentication Protocol

(1) U → S: Request
(2) S → U: Challenge
(3) U → S: ID || Hash(ID || Hash(Password) || Challenge)
(4) Server kiểm tra
S → U: ACK / NAK

- Challenge: chuỗi ký tự ngẫu nhiên
- Hash: MD5

28

Giao thức EAP

- Extensible Authentication Protocol
- Có khoảng 40 biến thể kết hợp thêm nhiều cơ chế khác nhau:
  - EAP-MD5: tương tự CHAP
  - EAP-TLS, EAP-TTLS, PEAP: kết hợp TLS
  - EAP-POTP: kết hợp One-Time-Password
  - EAP-PSK: kết hợp pre-shared key
  - ...

29

Xác thực 2 chiều sử dụng hệ mật mã KĐX

- Giả sử A và B đã chia sẻ khóa K_S

(1) A → B: ID_A

(2) B → A: N_B

(3) A → B: $f(K_S, N_B) \,||\, N_A$

(4) B → A: $f(K_S, N_A)$

Hàm f: có thể là các hàm mã hóa KĐX, hàm băm

K_S : khóa hoặc mật khẩu

30

Bài tập

- Xem xét tính an toàn của giao thức xác thực sau:

(1) A → B: ID_A || N_A

(2) B → A: $f(K_S, N_A)$ || N_B

(3) A → B: $f(K_S, N_B)$

- Nhận xét: người bắt đầu giao dịch phải là người chứng minh trước

31

Xác thực 2 chiều sử dụng hệ mật mã KCK

ISO/IEC 9798-3 / FIPS-196

(1) A → B: Request

(2) B → A: TokenID || N_B

(3) A → B: TokenID || $Cert_A$ || TokenAB

(4) B → A: TokenID || $Cert_B$ || TokenBA

TokenAB = N_A || N_B || $E(K_{RA}, N_A || N_B)$

TokenBA = N_A || N_B || $E(K_{RB}, N_A || N_B)$

32

Giao thức dạng zero-knowledge (ZKP)

- Peggy có bí quyết để phân biệt được các loại rượi vang khác nhau
- Victor muốn bỏ tiền để mua lại bí quyết
- Làm thế nào để Peggy chứng minh với Victor mà không làm lộ bí quyết?

33

Giao thức ZKP (Đọc thêm)

- Là các giao thức cho phép một bên chứng minh được thông tin của mình mà không làm lộ nội dung thông tin đó cho các bên còn lại (bên thứ 2 hoặc kẻ tấn công)
- Các bên tham gia giao thức:
  - Peggy-Người chứng minh: Peggy nắm được một số thông tin nào đó và muốn chứng minh cho Victor nhưng không muốn để lộ thông tin này
  - Victor-Người thẩm tra: Được quyền hỏi một số câu hỏi đến khi chắc chắn Peggy nắm thông tin. Victor không thể đoán thông tin từ câu trả lời của Peggy, hoặc do cố tình lừa Peggy tiết lộ thông tin
  - Eve-Kẻ nghe lén: Giao thức cần chống lại việc Eve nghe lén thông tin
  - Mallory: có nhiều quyền hơn Eve, có thể nghe lén, sửa đổi bản tin hoặc phát lại bản tin

34

Một ví dụ - Giao thức Feige–Fiat–Shamir

- Khởi tạo: Peggy chọn p, q là 2 số nguyên tố:
  - Tính $n = p \times q$
  - Chọn s sao cho $UCLN(s, n) = 1$, v sao cho $v = s^2 \ mod \ n$
  - Công bố (n,v). Peggy cần chứng minh cho Victor biết mình nắm giữ giá trị s
- Giao thức:

(1) P → V: $x = r^2 \ mod \ n$ r: số ngẫu nhiên

(2) V chọn ngẫu nhiên $b \in \{0, 1\}$

V → P: b

(3) P → V: $y = r \times s^b \ mod \ n$

(4) V kiểm tra phương trình đồng dư $y^2 \equiv x \times v^b \ (mod \ n)$

Hoặc viết dưới dạng khác $y^2 \ mod \ n = x \ \times \ v^b \ mod \ n$

35

Giả mạo

- Mallory có thể giả mạo bằng 2 cách:

(1) Bắt các cặp giá trị *(x, y)* và phát lại

(2) Phán đoán giá trị của bit *b* mà Victor thử thách:
  - Đoán b = 0, Mallory gửi $x = r^2 \ mod \ n$ và $y = r \ mod \ n$
  - Đoán b = 1, Mallory chọn *y* trước và tính *x* sao cho

$$y^2 \equiv x \times v \ (mod \ n)$$

- Xác suất thành công của Mallory là bao nhiêu?
- Làm thế nào để giảm xác suất thành công của Mallory trong 1 vòng kiểm tra?

36

Nhận xét

- Vì Peggy nắm được giá trị của s nên có thể qua được vô số vòng kiểm tra (Tính đầy đủ - Completeness)
- Nếu Mallory không biết s, thì xác suất giả mạo thành công lớn nhất là 2^{-n} với n là số vòng kiểm tra (Tính vững chãi-Soundness)
- Mallory không thể sử dụng lại bộ số *(x,y)* để lừa Victor
- Victor không biết gì về s vì bài toán tính căn bậc 2 rời rạc là khó
- Tương tự, Eve nghe trộm được mọi bộ số *(x,y,b)* cũng không thể đoán được s

37

Các nguy cơ

- Peggy không thay đổi r sau mỗi vòng kiểm tra
- Chess Grandmaster Problem
- Mafia Problem
- Terrorist Problem

38

Giao thức ZKP dựa trên hệ mật mã RSA (Một ví dụ khác)

- Peggy có khóa công khai K_U = (e,n) cần chứng minh anh ta có bí mật m
- Khởi tạo: Peggy tính *c = m^e mod n*
- Giao thức:

(1) P → V: $x = r^e \ mod \ n$ r: số ngẫu nhiên

(2) V chọn ngẫu nhiên $b \in \{0, 1\}$

V → P: b

(3) P → V: $y = r \times m^b \ mod \ n$

(4) V kiểm tra phương trình đồng dư $y^e \equiv x \times c^b \ (mod \ n)$

Tự kiểm tra tính đầy đủ và bền vững của giao thức.

Hãy đọc thêm lý thuyết tổng quan về ZKP trong tài liệu.

39

3. ONE TIME PASSWORD (OTP)

40

Xác thực đa yếu tố

- Phương pháp xác thực sử dụng mật khẩu không đủ an toàn
- Sử dụng mật khẩu một cách an toàn:
  - Đủ dài và khó đoán
  - Không dùng chung cho nhiều tài khoản
  - Thay đổi thường xuyên
  - … → hầu hết người dùng không thực hiện được
- Kẻ tấn công có khả năng đánh cắp CSDL và thực hiện tấn công từ điển
- Xác thực đa yếu tố (MFA – Multi Factor Authentication)
  - Cái người dùng biết: mật khẩu
  - Yếu tố còn lại: tài sản, mẫu sinh trắc

  → kẻ tấn công chỉ biết mật khẩu là không đủ để vượt qua xác thực

41

One Time Password

- Mật khẩu chỉ dùng để xác thực cho 1 phiên hoặc 1 giao dịch
- Phân loại:
  - S/Key OTP
  - Hash-based OTP (HOTP)

  } Event-based OTP
  - Time-based OTP (TOTP)
- Cách thức phân phối:
  - SMS
  - Ứng dụng
  - Email
  - Token

42

S/Key OTP(RFC 1760)

- Sử dụng trong một số hệ điều hành Unix
- Pha sinh mật khẩu:

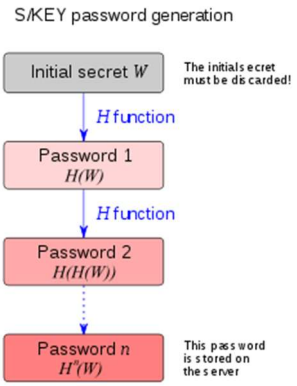
(1) Server chọn một giá trị bí mật W

(2) Áp dụng hàm băm (hoặc HMAC) n lần lên W

(3) Lưu H_n trong CSDL

(4) Cung cấp cho client H_n, H_{n-1},…, H_1

(5) Client hủy giá trị H_n



S/Key OTP(tiếp)

- Xác thực lần đầu

(1) Client gửi H_{n-1}

(2) Server so sánh Hash(H_{n-1}) với H_n trong CSDL

(3) Nếu bước 3 xác thực đúng, thay H_n bằng H_{n-1}. Gửi thông báo xác thực thành công

(4) Client xóa H_{n-1} nếu đăng nhập thành công

- Xác thực các phiên kế tiếp: tương tự

HOTP (RFC 4226)

- Bộ đếm: C (8 byte)
- Giá trị bí mật: K đã chia sẻ trước với client
- Hàm *HOTP(K, C)*

(1)Tính *HS = HMAC-SHA-1(K,C)*

(2)Trích xuất 4 bytes từ HS bằng hàm Dynamic Truncation

$$Sbits = DT(HS)$$

(3) Chuyển *Sbits* sang dạng thập phân. Lấy giá trị HOTP với số chữ số *k* tùy ý.

$$Snum = StToNum(Sbits)$$
$$D = Snum \bmod 10^k$$

45

Hàm DT

- Đầu vào: Chuỗi 20 byte S
- Xử lý:
  - Lấy OffsetBits = 4 bit thấp của S[19]
  - Biến đổi sang dạng thập phân *Offset = StToNum(OffsetBits)*
  - Trích xuất 4 byte trong chuỗi S bắt đầu từ vị trí *Offset* được chuỗi P
- Đầu ra: Xóa bit đầu tiên của P

46

Sử dụng HOTP trong giao thức xác thực

- Yêu cầu: Chia sẻ khóa K và C một cách an toàn
- Server: C ← C + 1. Tính HOTP(K, C) và lưu trong CSDL
- Client: C ← C + 1. Tính HOTP(K, C) và người dùng gửi cho server
- Server:
  - Nếu OTP nhận được là hợp lệ tạo OTP mới thay cho giá trị cũ trong CSDL
  - Nếu OTP nhận được không hợp lệ, thực hiện đồng bộ lại với tham số đồng bộ s. Yêu cầu xác thực lại.
  - Sau T lần xác thực lại không hợp lệ, khóa tài khoản

47

Đồng bộ trong HOTP

- Khi sử dụng HOTP trên thiết bị OTP Hardware Token, mã OTP được sinh ra theo yêu cầu người dùng
- Tính trạng mất đồng bộ: người dùng yêu cầu mã OTP nhưng không xác thực → giá trị bộ đếm của Token và Server khác nhau
- Đồng bộ hóa:
  - Server tính toán HOTP cho s lần kế tiếp
  - Yêu cầu người dùng gửi một chuỗi (2-3, hoặc hơn) các giá trị HOTP sinh được từ Token
  - So sánh chuỗi HOTP của người dùng với chuỗi HOTP đã sinh và thực hiện đồng bộ

48

TOTP(RFC 6238)

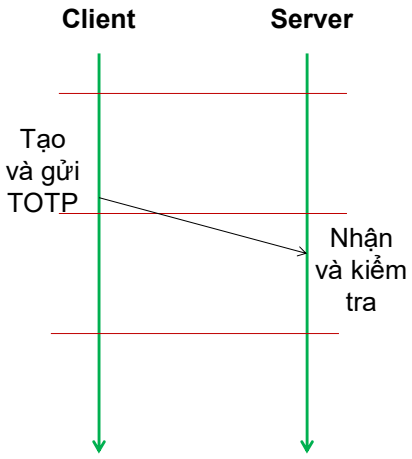
- Thực hiện tương tự HOTP
- Thay thế bộ đếm C bằng giá trị thời gian:

C = (Current UnixTime – T_0)/X

T_0: Mốc thời gian

X: Bước thời gian (time step)

- Vấn đề trễ xử lý
- Client có thể gửi cùng 1 TOTP trong 1 bước thời gian, nhưng server chỉ chấp nhận cho 1 lần xác thực



49

Mất đồng bộ trong TOTP

- Thời điểm tạo OTP ở client và kiểm tra ở server thuộc 2 chu kỳ khác nhau, hoặc
- Mất đồng bộ đồng hồ
- Phía kiểm tra cho phép chấp nhận một giá trị OTP nằm trong khoảng sai số cho phép
- Miền chấp nhận [TOTP(T_p) , TOTP(T_f)]

T_p = (Current UnixTime – 2X + 1 – T_0)/X

T_f = (Current UnixTime + X – 1 – T_0)/X

50

SMS OTP

- Giá trị OTP được sinh ở server và gửi cho người dùng qua tin nhắn SMS
- Không đảm bảo an toàn:
  - Điện thoại người dùng bị nghe lén
  - Giả mạo trạm BTS
  - Tấn công lợi dụng lỗ hổng của giao thức SS7

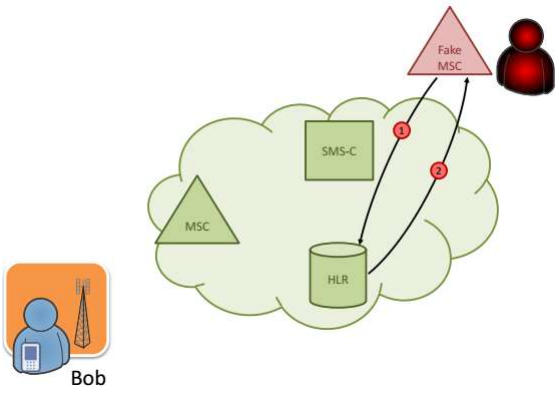
51

Tấn công lỗ hổng của SS7 (Đọc thêm)

- SS7(Signaling System 7): bộ giao thức điều khiển truyền dữ liệu giữa các cell trong mạng đi động
- Không có cơ chế xác thực
- IMSI: Định danh của thẻ SIM
- IMEI: Định danh của thiết bị
- MSISDN: Số thuê bao
- HLR(Home Location Register): CSDL thuê bao
- MSC(Mobile Switching Center): Bộ chuyển mạch
- MAP(Mobile Application Part): giao thức điều phối truyền dữ liệu giữa các thành phần trong phiên dịch vụ

52

Tấn công SS7 – Bước 1



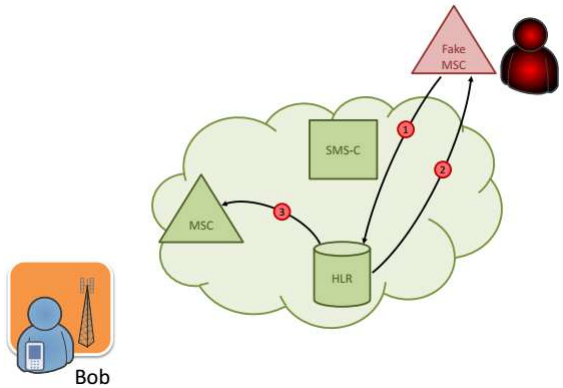
(1) Kẻ tấn công gửi thông điệp SendRoutingInfoForSM chứa MSISDN tới HLR

(2) HLR gửi thông điệp trả lời chứa:

- Số thuê bao
- Địa chỉ của MSC đang xử lý kết nối của nạn nhân(Bob)
- IMSI của nạn nhân

53

Tấn công SS7 – Bước 2



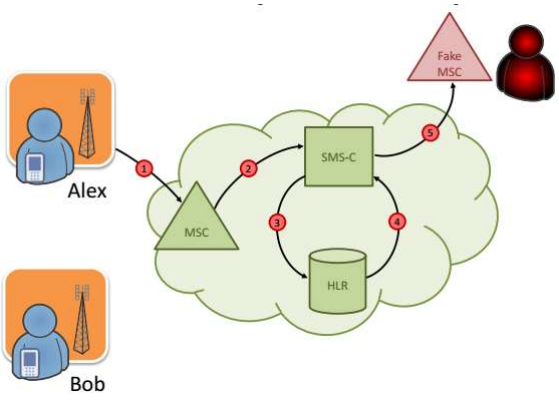
(1) Kẻ tấn công đăng ký thông tin của Bob trên MSC giả mạo (Fake MSC)

(2) HLR cập nhật vị trí mới của Bob

(3) HLR yêu cầu MSC cũ giải phóng thông tin

54

Tấn công SS7 – Bước 3



(1) Alex gửi tin nhắn SMS cho Bob
(2) MSC chuyển tiếp tin nhắn tới SMS-C
(3)SMS-C gửi thông điệp tới HLR yêu cầu vị trí của Bob
(4) HLR trả lại địa chỉ của Fake MSC
(5) SMS-C chuyển tiếp tin nhắn tới Fake MSC

55

Một vụ việc tấn công xác thực người dùng

56

Một vụ việc tấn công xác thực người dùng

Kịch bản sử dụng dịch vụ:

- B1: Khách hàng đăng nhập vào hệ thống eBanking
- B2: Khách hàng nhập lệnh chuyển tiền
- B3: Hệ thống eBanking gửi mã OTP qua tin nhắn SMS tới số điện thoại mà khách hàng đã đăng ký
- B4: Khách hàng nhập mã OTP nhận được vào hệ thống để xác nhận chuyển tiền

→Xác thực đa yếu tố:

(1) Mật khẩu truyền thống

(2) SMS OTP

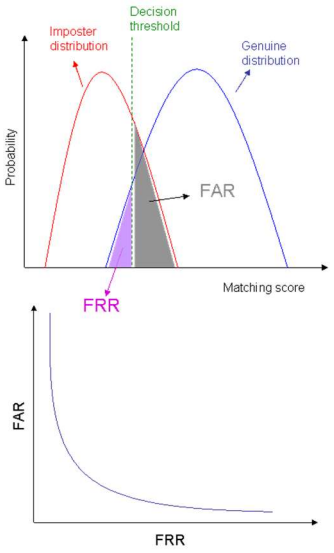
57

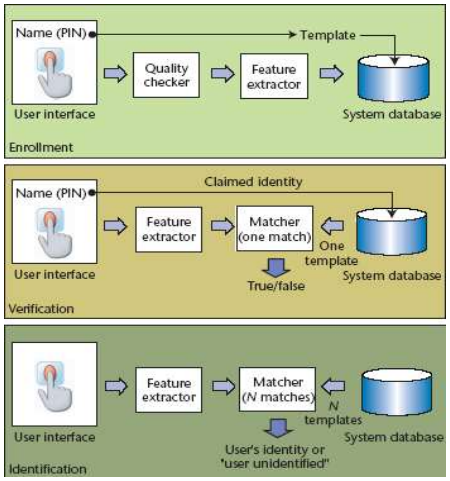
Một vụ việc tấn công xác thực người dùng

- Vietcombank cung cấp ứng dụng di động Vietcombank Smart OTP cung cấp mã xác thực OTP
- B1: Mở ứng dụng và điền số ĐT đăng ký SMS Banking
- B2: Hệ thống gửi mã xác thực OTP tới số điện thoại
- B3: Người dùng nhập mã xác thực vào ứng dụng
- B4: Nếu mã OTP đúng, ứng dụng được kích hoạt

58

Xác thực bằng sinh trắc (biometric)





59

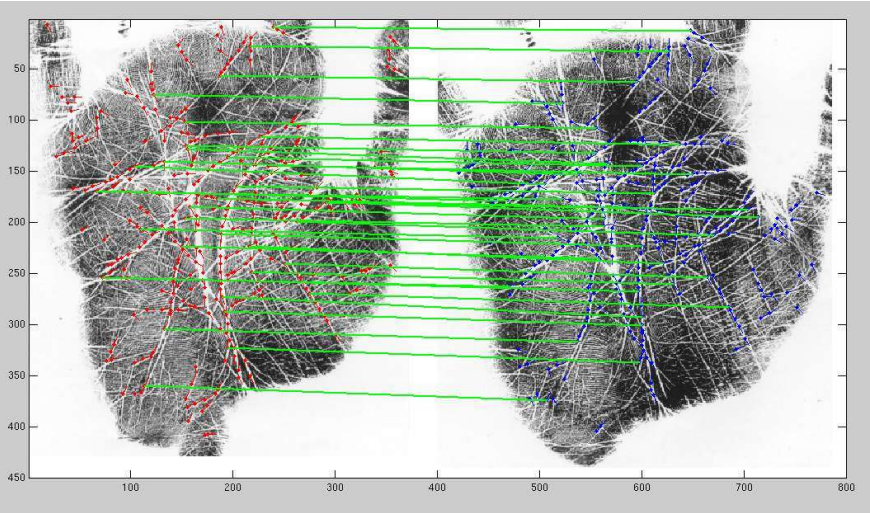
59

Dấu vân tay

60

60

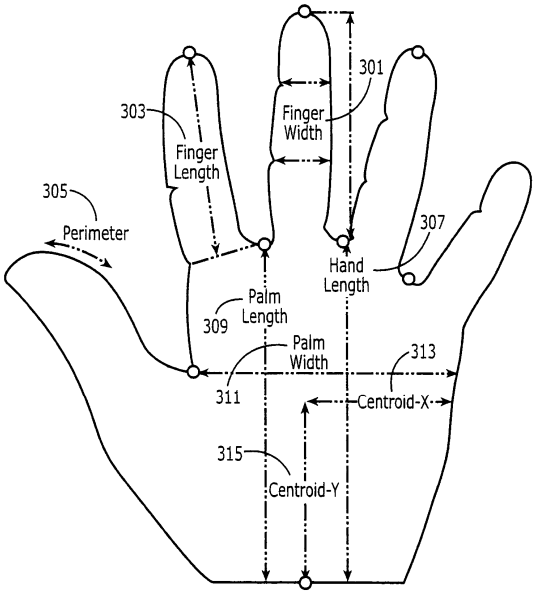
Vân lòng bàn tay



61

61

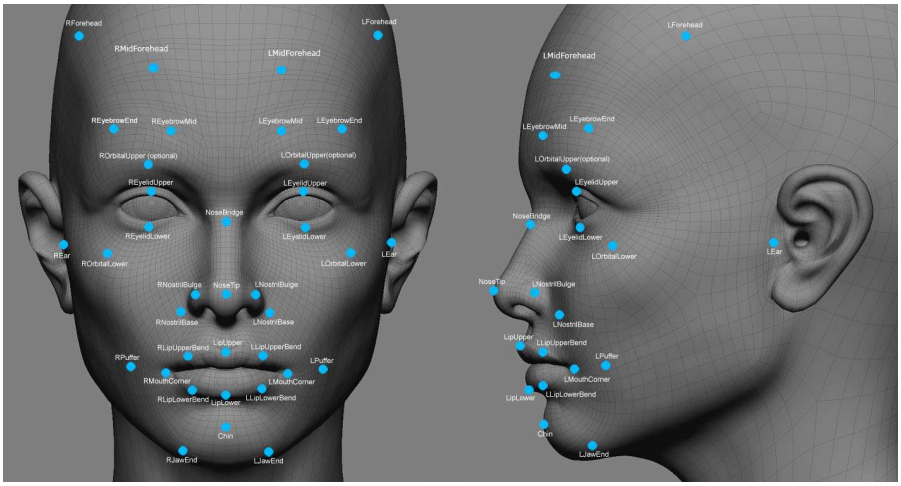
Cấu trúc bàn tay



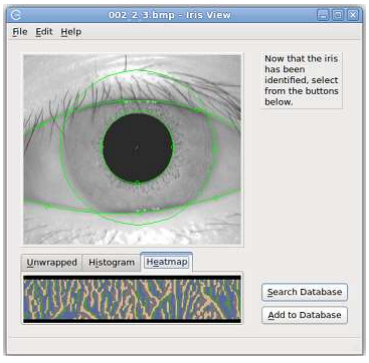
62

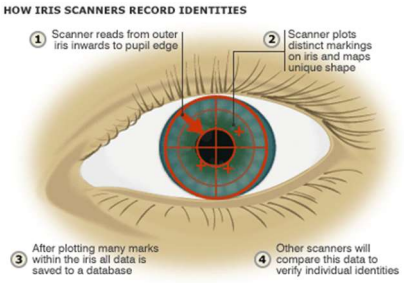
62

Khuôn mặt

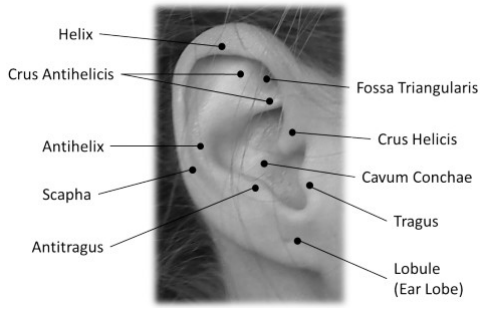


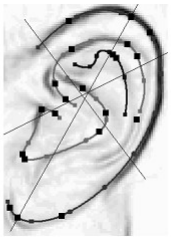
Mống mắt

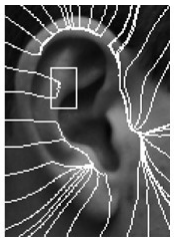




Vành tai



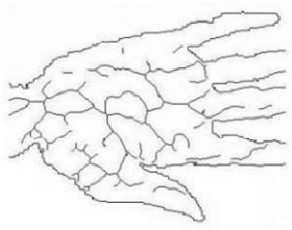




65

65

Mạch máu



66

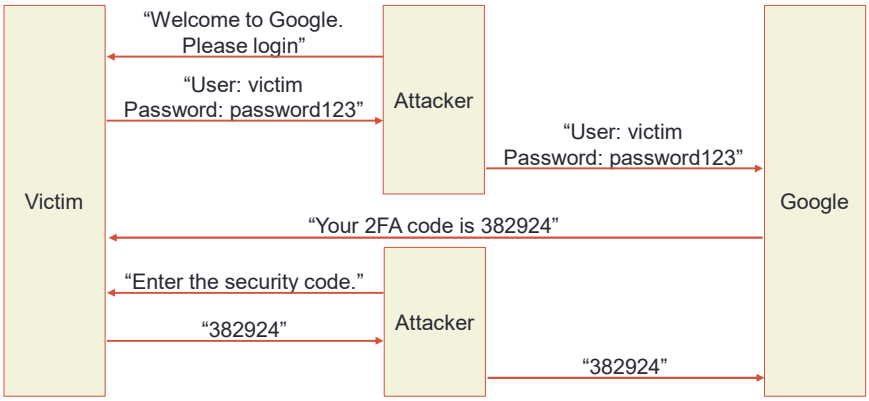
66

Vượt qua xác thực đa yếu tố

- Tấn công chuyển tiếp(relay attack): sử dụng kỹ thuật giả mạo để đánh cắp các yếu tố
- Ví dụ:
  - Sơ đồ xác thực 2 yếu tố:
    - ✓Yếu tố thứ 1: mật khẩu
    - ✓Yếu tố thứ 2: mã OTP gửi tới điện thoại
  - Tấn công:
    - ✓Website giả mạo yêu cầu người dùng nhập mật khẩu
    - ✓Kẻ tấn công đăng nhập ngay vào website thật với mật khẩu đã có
    - ✓Website thật gửi mã OTP tới điện thoại người dùng
    - ✓Website giả mạo yêu cầu người dùng nhập mã OTP
    - ✓Kẻ tấn công sử dụng mã OTP để hoàn thành xác thực bước 2

67

Tấn công chuyển tiếp – Ví dụ

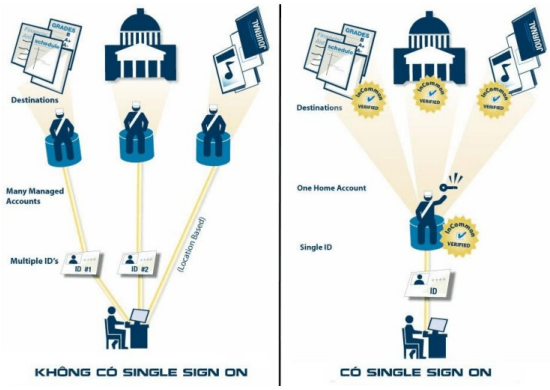


68

4. SINGLE SIGN ON(SSO)

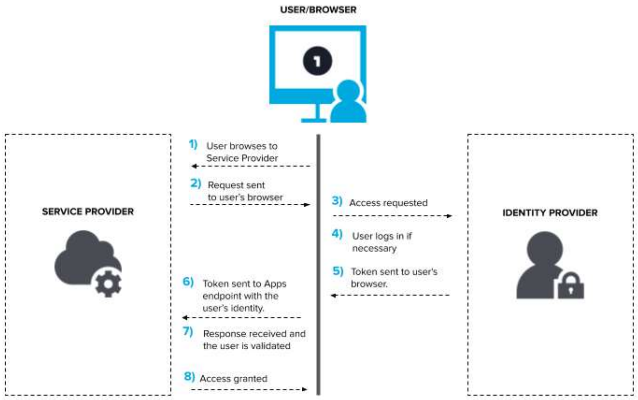
Khái niệm

- SSO là một cơ chế xác thực yêu cầu người dùng đăng nhập vào chỉ một lần với một tài khoản và mật khẩu để truy cập vào nhiều ứng dụng trong 1 phiên làm việc (session).



Single Sign On

Service Provider Initiated Workflow



Các giải pháp SSO khác

- OpenID Connect
- CAS
- Open SAML
- CA Single Sign On
- Java Open Single Sign On
- Google Sign-In
- Facebook Login