

Authentication Mechanisms and Protocols

Authentication (Authn)

Proof that an entity has an attribute it claims to have

- 1:
- Hi, I'm Joe
 - Prove it!
 - Here is my **proof**,
calculated with **Joe's credentials** that I've agreed with you
 - Proof accepted/not accepted ← Verifica a prova!,,

- 2:
- Hi, I'm over 18
 - Prove it!
 - Here is a **claim** issued by a competent authority,
which I can also **prove** that I'm the owner
 - Proof and claim accepted/not accepted

Provar que uma entidade é alguma "coisa"

Com o certificado de estudante
garante que seu estudante
da UA

Autenticação

Authn: Proof Types

Something we know

- A secret memorized (or written down...) by Joe

Something we have

- An object/token solely held by Joe

s.w.: Cartão de Crédito

→ O segredo está dentro do equipamento

Biometria

Something we are

- Joe's Biometry

Multi-factor authentication

- Simultaneous use of different proof types
- 2FA = Two Factor Authentication
 - Algo que sabemos
 - Algo que temos

Risk-based MFA

- Variable MFA
- Higher attack risk, more factors or less risky factors
- Lower attack risk, less or easier factors



Próxima aula:
① Autenticação (validar quem é)
② Autorização (verificar se tem acesso ao recurso!)

Login no Google:
Para um segundo login
• O que temos? (Cookie no browser, SMS para o telemóvel)
• O que sabemos? username + password

Authn : Goals

Authenticate interactors

- People, services, servers, hosts, networks, etc.

Enable the enforcement of authorization policies and mechanisms

- Authorization ≠ authentication
- Authorization ⇒ authentication

FIRST



Facilitate the exploitation of other security-related protocols

- e.g. key distribution for secure communication

Authn : Requirements

Trustworthiness

Confiança!

- Face ID ≠ different levels of trustworthiness
- Touch ID

- How good is it in proving the identity of an entity?
- How difficult is it to be deceived?
- Level of Assurance (LoA)

Secrecy

- No disclosure of secret credentials used by legit entities

Quando desenharmos
o serviço definiremos
o LoA

NIST 800-63				
LoA <i>Level of Assurance</i>	DESCRIPTION	TECHNICAL REQUIREMENTS		
		IDENTITY PROOFING REQUIREMENTS	TOKEN (SECRET) REQUIREMENTS	AUTHENTICATION PROTECTION MECHANISMS REQUIREMENTS
1	Little or no confidence exists in the asserted identity; usually self-asserted; essentially a persistent identifier	Requires no identity proofing	Allows any type of token including a simple PIN	Little effort to protect session from off-line attacks or eavesdropper is required.
2	Confidence exists that the asserted identity is accurate; used frequently for self service applications	Requires some identity proofing	Allows single-factor authentication. Passwords are the norm at this level.	On-line guessing, replay and eavesdropping attacks are prevented using FIPS 140-2 approved cryptographic techniques.
3	High confidence in the asserted identity's accuracy; used to access restricted data	Requires stringent identity proofing	Multi-factor authentication , typically a password or biometric factor used in combination with a 1) software token, 2) hardware token, or 3) one-time password device token	On-line guessing, replay, eavesdropper, impersonation and man-in-the-middle attack are prevented. Cryptography must be validated at FIPS 140-2 Level 1 overall with Level 2 validation for physical security.
4	Very high confidence in the asserted identity's accuracy; used to access highly restricted data.	Requires in-person registration	Multi-factor authentication with a hardware crypto token.	On-line guessing, replay, eavesdropper, impersonation, man-in-the-middle, and session hijacking attacks are prevented. Cryptography in the hardware token must be validated at FIPS 140-2 level 2 overall, with level 3 validation for physical security.

Authn : Requirements

Robustness

- Prevent attacks to the protocol data exchanges
- Prevent on-line DoS attack scenarios *← se o processo que está exposto ao mundo*
- Prevent off-line dictionary attacks

Simplicity

- It should be as simple as possible to prevent entities from choosing dangerous shortcuts *pior de 4 dígitos*

Deal with vulnerabilities introduced by people

- They have a natural tendency to facilitate or to take shortcuts
- Deal with phishing! *↳ "alguém passe o cartão de entrada por mim"*

Authn: Entities and deployment model

Entities

Deployment model

People

Along the time

"normal" → • Only when interaction starts

◦ Continuously along the interaction

Authn interactions: Basic approaches

Direct approach

① Começando pelo cliente



② Começando pelo servidor
(o cliente pede o desafio...)

④ Seguro

Forma a prova.

1. Provide credentials
2. Wait for verdict

- Advantage: no computations by the presenter
- Disadvantage: credentials can be exposed to malicious validators

Challenge-response approach

1. Get challenge
2. Provide a response computed from the challenge and the credentials
3. Wait for verdict

Responde o desafio

- Advantage: credentials are not exposed to malicious validators
- Disadvantage: requires computations by the presenter

Authn of subjects: Direct approach w/ known password

A password is checked against a value previously stored

- For a claimed identity (username)

PBKDF2 ou diretamente

Personal stored value:

- Transformed by a unidirectional function
- Windows: digest function
- UNIX: DES hash + salt
- Linux: MD5 + salt
 - hash is configurable

Password
Key-Based
Derivation
Function 2

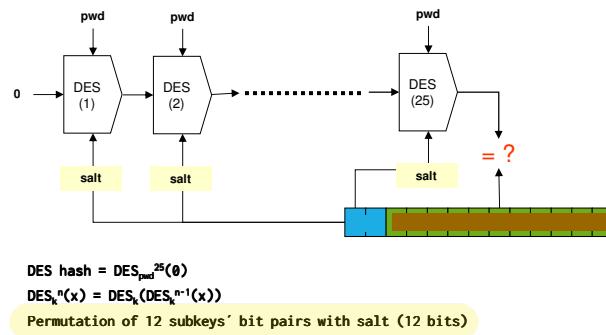
Login
password

H(c) →
check if
equal to the
one stored in
database

PKBPF2

Optimal: PBKDF2, Script with high complexity

Authentication of subjects: Direct approach w/ known password



Authn of subjects: Direct approach w/ known password

Advantage

- Simplicity!

A senha para o utilizador
 é sempre igual !! ~~TOP~~

O seu segredo é PROBLEMATICO

Problems

- Usage of weak passwords
- Enable dictionary attacks
juntar palavras comuns utilizadas para filtrar
- Transmission of passwords along insecure communication channels
 - Eavesdroppers can easily learn the password
 - e.g. Unix remote services, PAP

Password
Authentication
Protocol

Top Ten 2017
from Splashdata

1. 123456
2. Password
3. 12345678
4. qwerty
5. 12345
6. 123456789
7. letmein
8. 1234567
9. football
10. iloveyou

Authn of people: Direct approach with biometrics

People get authenticated using body measures

- Biometric samples
- Fingerprint, iris, face geometry, voice timber, manual writing, vein matching, etc.

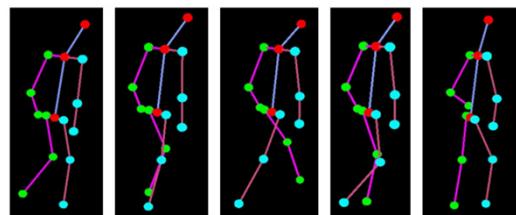
Measures are compared with personal records

- Biometric references (or template)
- Registered in the system with a previous enrolment procedure

Identification vs authentication

- Identification: 1-to-many check for a match
- Authentication: 1-to-1 check for a match

PROBLEMA!,,



Authn of people: Direct approach with biometrics

Advantages

- People do not need to use memory, or carry something
 - Just be their self
- People cannot choose weak passwords
 - In fact, they don't choose anything
- Authentication credentials cannot be transferred to others
 - One cannot delegate its own authentication

Authentication of people: Direct approach with biometrics

Problems

- Biometric methods are still incipient
 - In many cases it can be fooled with ease (Face Recognition, Fingerprint)
- People cannot change credentials
 - If the credentials or templates are stolen
- Credentials cannot be transferred between individuals
 - If it is required in extraordinary scenarios
- Can pose risks to individuals
 - Physical integrity can be compromised by an attacker in order to acquire biometric data
- It is not easy to be implemented in remote systems
 - It is mandatory to have secure and trusted biometric acquisition devices
- Biometrics can reveal other personal secrets
 - Diseases

Authn of subjects:
Direct approach with one-time passwords

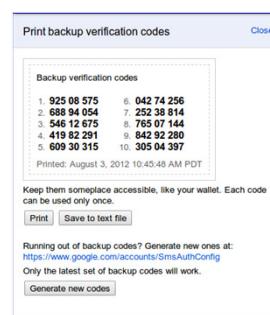
Senha discartável →

One-Time Passwords = Secrets that can be used only once

- Pre-distributed directly, or the result of a generator function

Example: Bank codes, Google Backup Codes

- ① Autenticação (login)
....Fazer intransigência ...
- ② Nova autenticação (matriz de códigos)



© André Zúquete, João Paulo Barraca

Information and Organizational Security

17

Códigos backup
para recuperar
a conta !!

Authn of subjects:
Direct approach with one-time passwords

Advantages

- Can be eavesdropped, allowing its use in channels without encryption
- Can be chosen by the authenticator, which may enforce a given complexity
- Can depend on a shared password *Não só pessoas a conhecer!*

SK $\xrightarrow{\text{}} \text{OTP}_1 \xrightarrow{\text{}} \text{OTP}_2 \xrightarrow{\text{}} \text{OTP}_3$

Problems

- Interacting entities need to know which password to use on each occasion
 - Implies some form of synchronization (e.g., index, coordinates)
- Individuals may require additional resources to store/generate the passwords
 - Sheet of paper, application, additional device, etc. *Pode ser roubado...*

*Alguma "coisa" tem de guardar
a OTP - one time password !,,*

© André Zúquete, João Paulo Barraca

Information and Organizational Security

18

Yubico company
↓

Yubikey

For OTP - one time password ...

Personal Authentication Device

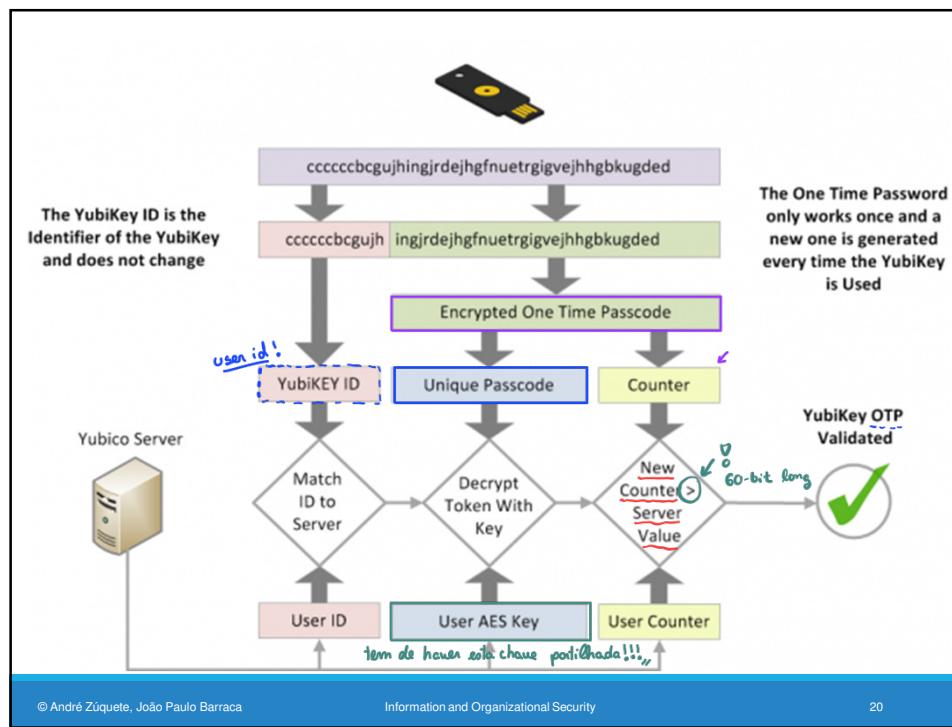
- USB, Bluetooth and/or NFC



"API" que emula um teclado
e que fornece uma senha
grande e está sempre a
mudar...

Activation generates a 44 characters key

- Emulates a USB keyboard (besides an own API)
- Supports HOTP (events) or TOPT (Temporal)
- If a challenge is provided, user must touch the button to obtain a result
- Several algorithms, including AES 256



ATE AGORA: direct approach, onde nós é que dávamos a password e o servidor valida com algo pre-stored...

Challenge-response Approach

MUITO bom...

The authenticator provides a challenge

- A nonce (value not once used)
- Usually random
- Can be a counter

The authenticated entity transforms the challenge

- The transformation method is shared with the authenticator

The result is sent to the authenticator

The authenticator verifies the result

- Calculates a result using the same method and challenge
- Or produces a value from the result and evaluates if it is equal to the challenge, or to some related value

© André Zúquete, João Paulo Barraca

Information and Organizational Security

21

Challenge-Response Approach

Advantages

- Authentication credentials are not exposed
- An eavesdropper will see the challenge and the result
but has no knowledge about the transformation *Não sabe o "shift"*

Problems

- Authenticated entities must have the capability of calculating results to challenges
 - Hardware token ou software application
- The authenticator may need to keep shared secrets (in clear text)
 - Secrets can be stolen
 - Individuals may reuse secrets in other systems, enabling lateral attacks
- May be possible to calculate all results to a single (or all) challenge(s)
 - Can reveal the secret used *lo atacante testa todos*
- May be vulnerable to dictionary attacks *Atacante pode reenviar a resposta antiga...*
- Authenticator should NEVER issue the same challenge to the same user

Robusto contra MITM
Credenciais não são expostas
Compatível com outras aproximações
Autenticador escolhe transformação e complexidade do desafio

→ Dentro do servidor e dentro do hardware!

© André Zúquete, João Paulo Barraca

Information and Organizational Security

22

Authn of Subjects: Challenge-Response with Smartcards

Authentication Credentials

- Having the smartcard *Algo que eu tenho!* ①
- e.g., the Citizen Card
- The private key stored inside the smartcard *Não sabemos a chave... Nem conseguimos replicar a resposta...*
- The PIN code to access the key *Algo que eu sei!* ②

The authenticator knows

- The user public key
public keys via CAs certificates...

Robust against:

- Dictionary attacks
- Offline attacks to the database
- Insecure channels



© André Zúquete, João Paulo Barraca

Information and Organizational Security

23

↑ Problem... they can track you

Authn of Subjects: Challenge-Response with Smartcards

Challenge-Response Protocol

- The authenticator generates a challenge *Fazendo mais segura!*
- Smartcard owner ciphers the challenge with their private key
 - Stored in the smartcard, protected by the PIN code
 - In alternative, can sign the challenge
- The authenticator deciphers the result with the public key
 - If the decrypted result matches the challenge, the authentication is successful
 - In alternative, it can verify the signature (which is the same process)



© André Zúquete, João Paulo Barraca

Information and Organizational Security

24

For today's world,
↓

Authn of Subjects: Challenge-Response with other tokens



FIDO2 tokens (FIDO Alliance)

- For both mobile and desktop environments
- Web Authentication (WebAuthn) specification
- Client-to-Authenticator Protocol (CTAP)
- Security
 - Credentials never leave the user's device and are never stored on a server
 - No risks of phishing, no password theft (still, tokens can be stolen)
 - No replay attacks
 - Token certification levels
- Privacy
 - Credentials are unique per website
 - Tracking is not possible (different web sites, different public keys for the same token)
 - Biometric data, when used, never leaves the user's device



<https://www.inoxev.de/de/blog/fido2-webauthn-in-practice/>

FIDO2 certification

FIDO Authenticator Certification Examples

L3+		USB U2F Token built on a CC-certified Secure Element	<u>Certification: L3+</u>		
L3		USB U2F Token built on a basic simple CPU, OS, is certified. Good physical anti-tampering enclosure	UAF implemented as a TA running on a certified TEE with POP memory		
L2		UAF implemented as a TA in an uncertified TEE			
L1+		UAF in downloadable app using white box crypto and other techniques	<u>Certification: L1+</u>		
L1		Downloaded app making use of Touch ID on iOS	FIDO2 making use of the Android keystore. Keystore is not certified	FIDO2 built into a downloadable web browser app	<u>Certification: L1</u>

Authn of Subjects: Challenge-Response with Shared Secret

Quando não existem senhas!!

Authentication Credentials

- Password selected by the individual

The authenticator knows:

- Bad approach: the shared password
- Better approach: A transformation of the shared password
 - The transformation should be unidirectional

PROBLEMA



Nunca ter a shared
Key diretamente, pois
ela é possibilidade
de ataques por
dicionário...

Authentication of Subjects: Challenge-Response with Shared Secret

Basic Challenge-Response Protocol

- The authenticator generates a challenge
- The individual calculates a transformation of the challenge and the password
 - result = hash(challenge || password)
 - or... result = encrypt(challenge, password)
- The authenticator reverts the process and checks if the values match
 - result == hash(challenge || password)
 - or ... challenge == decrypt(result, password)
- Examples with shared passwords: CHAP, MS-CHAP v1/v2, S/Key
- Examples with shared keys: SIM & USIM (cellular communications)

we use it in wifi...
PEAP + MS-CHAP v2,

{ Usar desafios
sem senhas,

one-time-password
used for authentication in Unix

PAP and CHAP (RFC 1334, 1992, RFC 1994, 1996)

shared secret

Protocols user for PPP (Point-to-Point Protocol)

- Unidirectional authentication
 - The authenticator authenticates users, but users do not authenticate the authenticator

PAP (PPP Authentication Protocol)

Sem desvio!

- Simple presentation of a UID/password pair
- Insecure transmission (in clear text)

O que mais interessa!

CHAP (Challenge-response Authentication Protocol)

Aut → U : authID, challenge

U → Aut: authID, MD5(authID, secret, challenge), identity

Aut → U : authID, OK/not OK

Vulnerável a
ataque por dicionário
se for uma senha fixa

Nota: usa transformação
da shared-key

- The authenticator can request further authentication at any time

Authentication of subjects: Challenge-Response with Shared Key

Uses a cryptographic key instead of a password

- Robust against dictionary attacks
- Requires a device to store the shared key

- No SIM temos isso!

↓

Tem vários algoritmos

GSM Subscriber authentication

Não do servido! Autentica o cliente dentro de...

Uses a secret shared between the **HLR** and the **subscriber phone**

- Uses 128-bit shared key (not an asymmetric key pair)
- Key is stored in the SIM card
- SIM card is unlocked by a user PIN
- SIM card answers challenges using the shared key
depende da operadora!

the PIN is not shared!

↑
subscribes a network,

Uses (initially unknown algorithms):

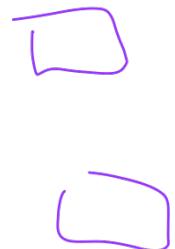
- A3 for authentication
- A8 to generate the session key
- A5 is a stream cipher for communication

Carões do telemóvel! //

A3 and A8 executed by the **SIM, A5 executed by the baseband**

- A3 and A8 can be chosen by the operator

*Autenticação: A3
Geração de chaves: A8
Comunicação: A5*



↓

GSM Subscriber authentication

MSC requests triples from HLR/AUC

- RAND, SRES, Kc
- It can ask one or several

desafio →

HLR generates RAND and the triples using the **subscriber Ki**

- RAND** random value (128 bits)
- SRES = A3 (Ki, RAND)** (32 bits)
- Kc = A8 (Ki, RAND)** (64 bits)

*Não universal
Ki é unico*

*Ki = shared key entre
o SIM e HLR*

Challenge-response Approach

Frequently uses COMP128 for the A3/A8 algorithms

- Recommended by the **GSM consortium**
- [SRES, Kc] = **COMP128 (Ki, RAND)**

Authentication of Systems

By name (DNS) or MAC/IP address

- Extremely weak, without cryptographic proof
 - Still... it is used by some services
 - e.g., NFS, TCP wrappers

in google we only know the name { we use CA to authenticate the public Keys...

With cryptographic keys

- Secret keys, shared between entities that communicate frequently
- Asymmetric key pairs, one per host
 - Public keys pre-shared with entities that communicate frequently
 - Public keys certified by a third party (a CA)
x509

Authentication of Services

Authentication of the host

- All services co-located in the same host are automatically and indirectly authenticated

Credentials exclusive to each service

Authentication:

- Secret keys shared with clients
 - When they require authentication of the clients
- Asymmetric key pairs by host/service
 - Certified by others or not

Confidencialidade
 +
 Integridade
 +
 Autenticidade

TLS (Transport Layer Security, RFC 2246)

~~• TLS 1.3 / TLS 1.0 / 1.1 are not secure!~~

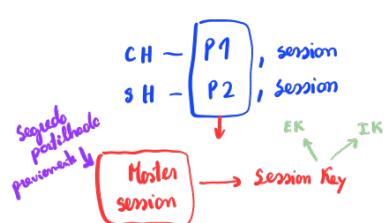
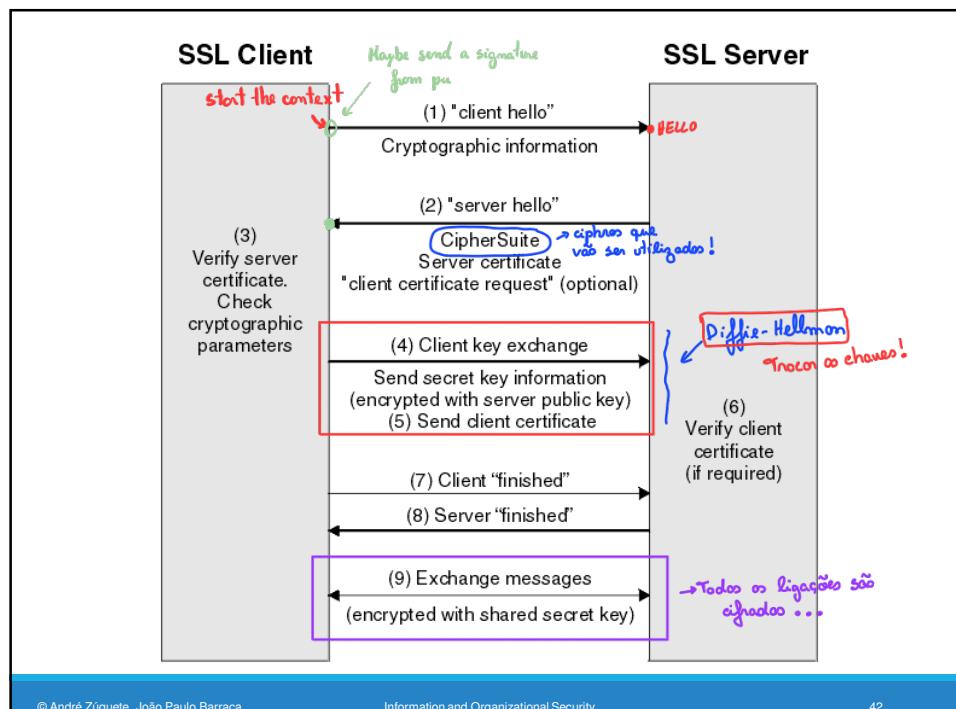
Secure Communication Protocol over TCP/IP

- Evolved from the SSL V3 (Secure Sockets Layer) standard
- Manages secure sessions over TCP/IP, individual to each application
 - Initially designed for HTTP traffic
 - Currently used for many other types of traffic



Security mechanisms

- Confidentiality and integrity of the communication between entities
 - Key distribution, negotiation of ciphers, digests and other mechanisms
- Authentication of the intervention entities
 - Servers, services, etc...
 - Clients (not so common)
 - Both executed with asymmetric keys and X.509 certificates



TLS Ciphersuites

If a server supports a single algorithm, it cannot expected for all clients to also support it

- More powerful/limited, older/newer

Linhos de texto que listam os algoritmos negociados!

The Ciphersuite concept allows the negotiation of mechanisms between client and server

- Both send their supported ciphersuites, and select one they both share
- The server chooses



Exemplo: ECDHE-RSA-AES128-GCM-SHA256

Format:

- Key negotiation algorithm: ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)
- Authentication algorithm: RSA a.g.: Elliptic Curve são apagados a cada sessão...
- Cipher algorithm and cipher mode: AES-128 GCM integridade + confidencialidade
- Integrity control algorithm: SHA256

SSH (Secure SHell)

Gerir sessões interativas
(acessos remotos, ...)

Manages secure console sessions over TCP/IP

- Initially designed to replace the Telnet application/protocol
- Currently used in many other applications
 - Execution of remote commands in a secure manner (rsh/rexec)
 - Secure copy of contents from/to remote hosts (rcp)
 - Secure FTP (sftp)
 - Secure (Generic) communication tunnels (carry standard IP packets)

Security Mechanisms

- Confidentiality and integrity of the communications
- Key distribution
- Authentication of the intervention entities
 - Server / Hosts
 - Client users
- Both achieved through several, and differentiated mechanisms

→ Na primeira ligação o utilizador confirma (ou não) a chave pública do servidor !!

Vulnerável a man-in-the-middle

SSH: Authentication Mechanisms

Server: a pair of asymmetric keys

- Keys are distributed during the interaction
 - Not certified!
- Clients store the public keys from previous interactions
 - Key should be stored in some trusted environment
 - If the key changes the client is warned
 - e.g., server is reinstalled, key is regenerated, an attacker is hijacking the connection
- Client can refuse to continue with the authentication process

Nem sempre...

Clients: authentication is configurable

- Default: username and password
- Other: username + private key
 - e.g. instantâneo!
 - The public key MUST be pre-installed in the server
- Other: integration with PAM for alternative authentication mechanisms

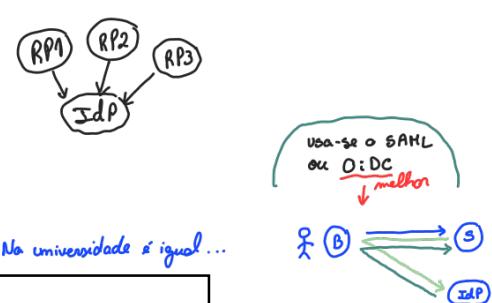
Centralized network authentication

Used for restricting network access to known clients

- In cabled networks
- In wireless networks
- In VPNs (Virtual Private Networks)

Usually implemented by a central service

- AAA server
 - Authentication, Authorization and Accounting
 - e.g. RADIUS and DIAMETER
- This server defines which network services the user can make use of



Authentication by an IdP

Unique, centralized authentication for a set of federated services

- The identity of a client, upon authentication, is given to all federated services
- The identity attributes given to each service may vary
- The authenticator is called **Identity Provider (IdP)**
- The federated service is called a **Relying Party (RP)** *Ponto confiável da autenticação!!!*
- In some cases, the provided identity attributes are shown to the client

Examples

- Authentication at UA
 - Performed by a central, institutional IdP (idp.ua.pt) *um único serviço*
 - The identity attributes are securely conveyed to the service accessed by the user
- Autenticação.gov (www.autenticacao.gov.pt)
 - Performed by a central, national IdP
 - The identity attributes are shown to the user
- Other:
 - Services used worldwide: Google, Facebook, etc.

Serviço federado

SA

Continente Wells Worlen

Na UA não temos o IdP

Serviços federados

com base em serviços
têm acesso a diferentes
atributos

Centralized authentication

Advantages:

- Can reuse same credentials over multiple systems/services
- Single secure repository for credentials
 - More difficult to steal credentials when used in many services
- Can implement restrictions to services/systems

Disadvantages:

- Requires additional servers
- Single point of failure: without authentication systems, no one will be authenticated
 - Important to also deploy local credentials for admins
- Introduces delays in the authentication process

Single Sign-On

→ Ao nível do IdP ele tem um tempo de expiração do cookie!!!

usa o cookie do serviço e se estiver válido ele aceita sem sign-in

A facility usually associated with IdP

- Both not mandatory nor always appropriate

SSO exists for simplifying users' life

- They login just one for accessing several federated services during a given time period

Podemos ter:

IdP som single sign-on

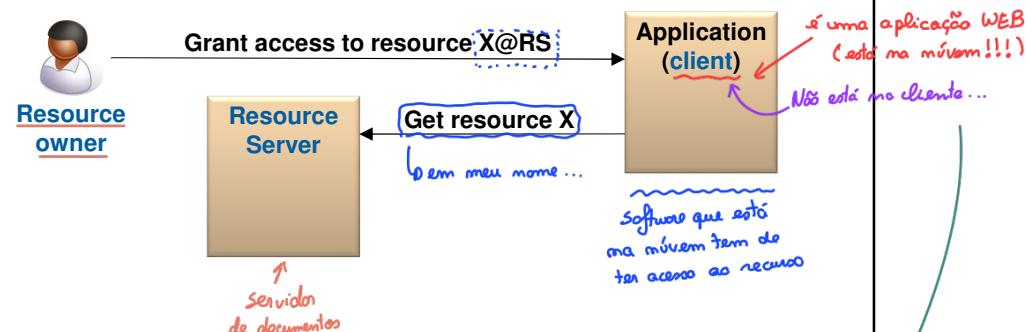
Não podemos ter:

Single sign-on sem IdP
(de forma federada)

Funcionamento minimal de um Website?

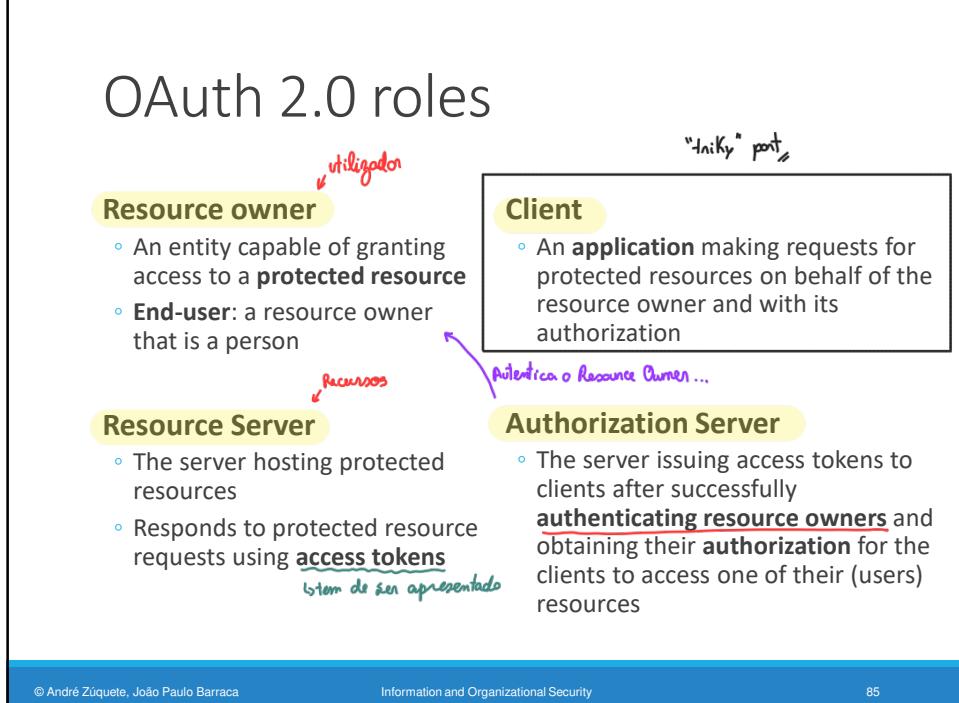
OAuth 2.0: delegation (RFC 6749)

A framework to allow users to delegate access to their resources on their behalf



Nota: Não esquecer que o Servidor Web não está do lado do cliente... E precisa de aceder a API.

OAuth 2.0 roles

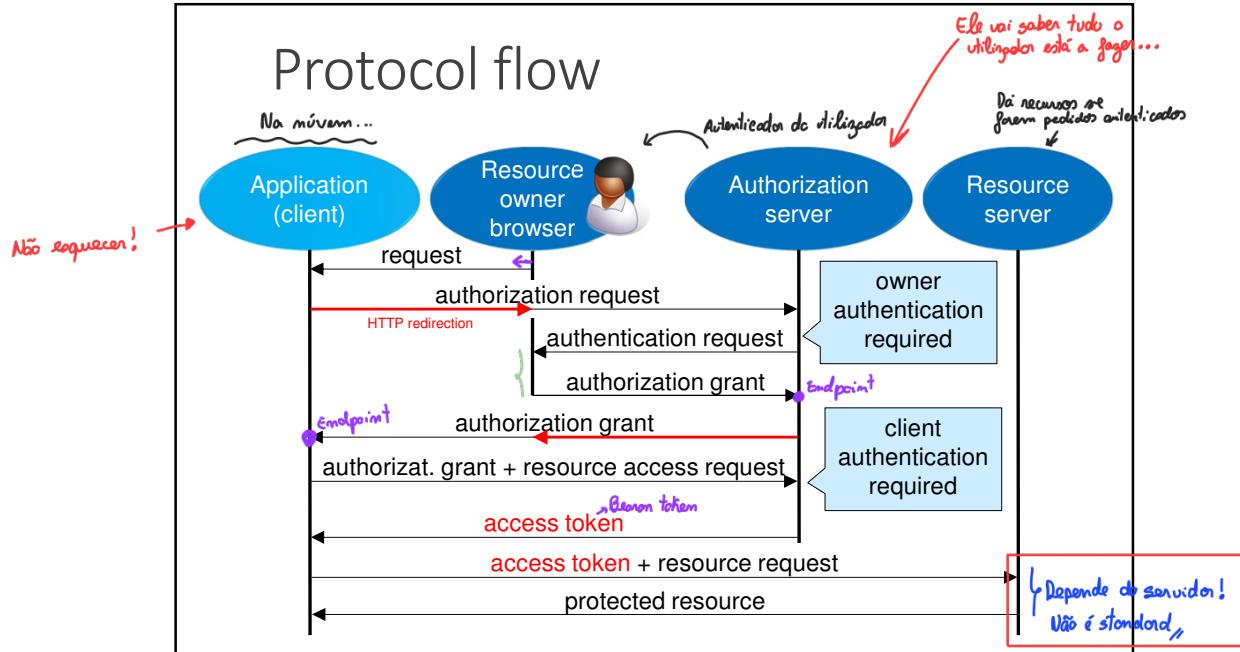


© André Zúquete, João Paulo Barraca

Information and Organizational Security

85

Protocol flow



© André Zúquete, João Paulo Barraca

Information and Organizational Security

86

Definir os
atributos de
identidade que
se pretende usar...

OpenID Connect (OIDC)

Open ID Connect

used in IdP

An identification layer on top of OAuth 2.0

- OAuth 2.0 provides the fundamental centralized authentication
- The protected resources are identity attributes
 - Packed in scopes
 - The attributes are called (identity) claims

o serviço X precisa de A,B,C,...