Autenticação baseada em passwords

ISEL – Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 | 1959-007 Lisboa

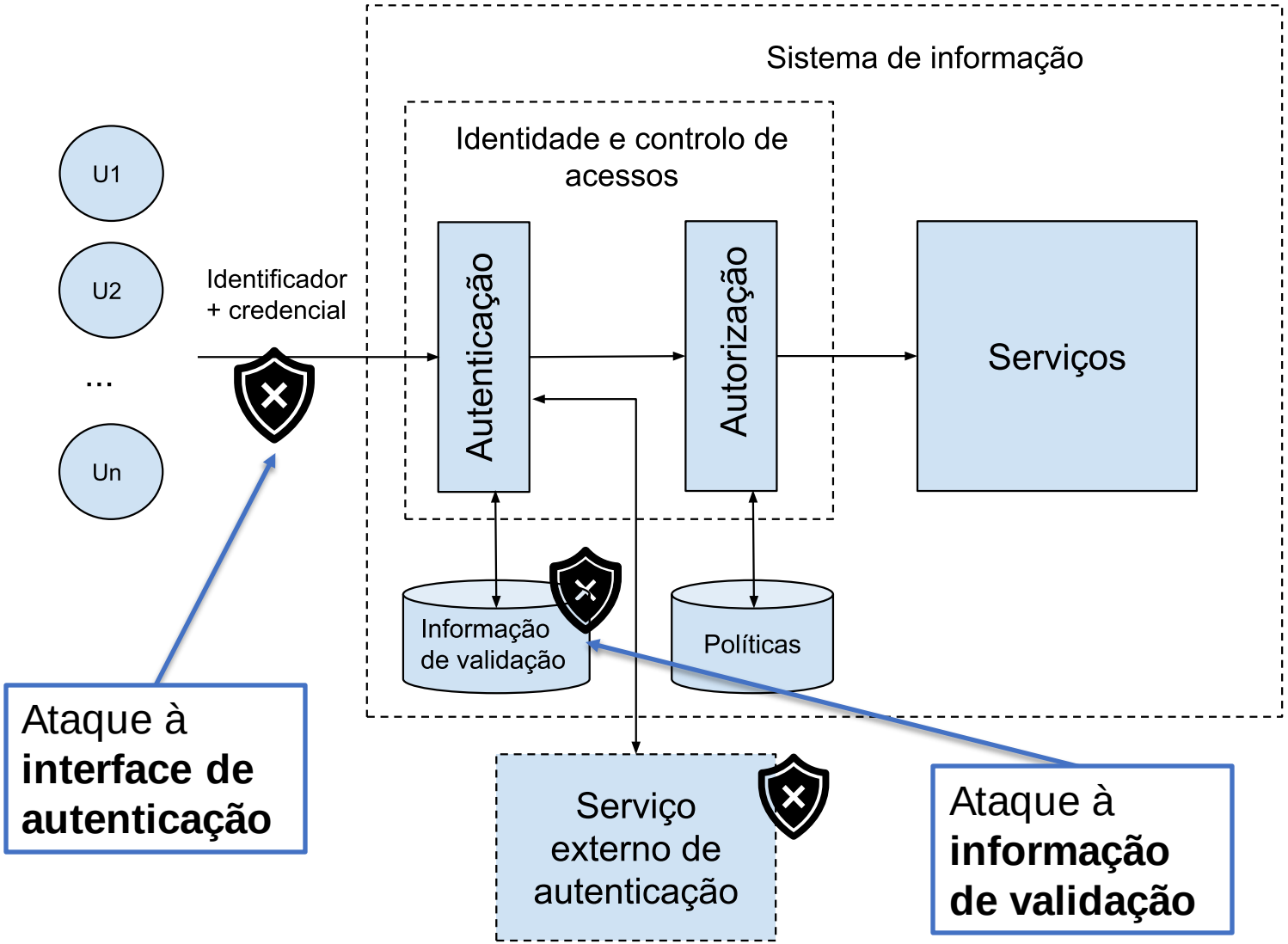
Identificação e Autenticação

- Autenticação é o processo de verificação duma alegada identidade
- Motivação
  - Parâmetro para as decisões de controlo de acessos
  - Parâmetro para as acções de personalização
  - Informação de auditoria
- Exemplo
  - “user” + “password”
    - “user” – identificação
    - “password” – autenticação

Informação de autenticação

- "Algo" que se conhece
  - "Passwords" e "passphrases"
- "Algo" que se possui
  - Ex.: "tokens" criptográficos, RSA SecurID
- "Algo" que se é
  - Ex.: características biométricas
- "Algo" que se faz
  - Ex.: assinatura manual

Autenticação e Controlos de acessos

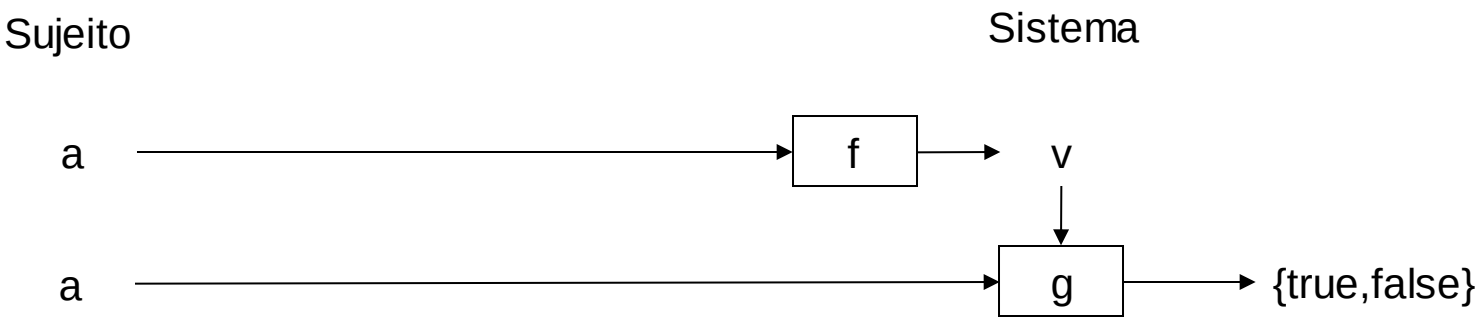


Vulnerabilidades de passwords textuais

- Ataques de dicionário à palavra-passe
  - Atacante usa uma lista de palavras-passe conhecidas, ou prováveis, e tenta exaustivamente as entradas da lista, em 1 ou mais utilizadores
  - Estes ataques têm como alvo a interface de autenticação dos sistemas ou o local onde está guardada a informação de validação
  - https://haveibeenpwned.com/Passwords
  - https://nordpass.com/most-common-passwords-list/

Sistema de autenticação

- Formalização
  - Conjunto **A** de informação de autenticação
  - Conjunto **V** de informação de validação
  - Função **f: A → V**
  - Função **g: V → A → {true, false}**
- Exemplo
  - **f(a) = H(a)**
  - **g(v)(a) = (v = H(a))**



Ataques de dicionário

- **Ataques do tipo 1**
  - Entrada: informação de validação - **v**
  - Saída: informação de autenticação
  - Para cada **a'** pertencente a **Dicionário**
    - Se **f(a') = v** retornar **a'**
  - Retornar "falha"

- **Ataques do tipo 2**
  - Entrada: função de autenticação – **g(v)**
  - Saída: informação de autenticação
  - Para cada **a'** pertencente a **Dicionário**
    - Se **g(v)(a') = true** retornar **a'**
  - Retornar "falha"

Protecção contra ataques de dicionário

- Aumentar a incerteza da "password"
  - Passwords aleatórias
  - Selecção proactiva
  - Verificação *offline*
- Controlar o acesso à informação de verificação
- Aumentar o tempo de processamento da função **f**
  - **f = H**, onde **H** é uma função de *hash*
  - Solução: $\mathbf{f = H^R}$
- Aumentar o tempo de processamento ou limitar o acesso à função **g(v)**

Ataques com pré-computação

- Baseai-se no facto da função **f** ser igual para todos os utilizadores
- Seja **D** um dicionário de palavras prováveis e **M** um *array* associativo
- Pré-computação
  - Para todos $\mathbf{a'_i}$ em **D**, calcular e armazenar o par $\mathbf{(f(a'_i), a'_i)}$ em **M** (tal que $\mathbf{M[f(a'_i)] = a'_i}$)
- Ataque
  - Dado **v**, retornar **M[v]**
- A pré-computação é usada para obter a "password" de qualquer utilizador

Protecção: “salt”

- Protecção contra os ataques de dicionário descritos anteriormente
- Solução: tornar a função **f** diferente para cada utilizador
- Exemplo: $\mathbf{f_U(a) = H(salt_U \mid a)}$, onde
  - $\mathbf{f_U}$ é a função associada ao utilizador **U**
  - $\mathbf{salt_U}$ é uma sequência de “bytes” gerada aleatoriamente para cada utilizador
- Neste cenário, a pré-computação depende de **salt**
  - é específica de cada utilizador do sistema
  - não pode ser utilizada para atacar todos os utilizadores do sistema

Protecção contra ataques tipo 2

- Limitar o acesso à função de autenticação **g(v)** após a detecção de tentativas de autenticação erradas
- Técnicas
  - *Backoff*
    - O tempo de execução de **g(v)** depende do número anterior de tentativas erradas
  - Terminação da ligação
    - Terminação da ligação em caso de erro
  - Bloqueamento
    - Bloqueamento da função **g(v)** após um número de tentativas erradas
  - *Jailing*
    - Acesso ao serviço com funcionalidade limitada
- Problema: garantir a disponibilidade do serviço

Aumentar o custo dos pedidos

- Diminuir o número de pedidos realizados através do aumento do seu custo para o cliente

- Desafio computacional que tem de ser calculado pelo "user-agent" cliente antes da realização do pedido
  - Necessita de computação do lado do cliente (ex. Javascript)

- CAPTCHA - "Completely Automated Public Turing Test to Tell Computers and Humans Apart" - http://www.captcha.net/
  - Fácil para humanos
  - Difícil para computadores