

Fundamentos de Segurança Informática (FSI)

2021/2022 - LEIC

Manuel Barbosa
mbb@fc.up.pt

Aula 19

Autenticação 1

Autenticação

- Autenticação de origem de mensagens (aulas anteriores):
 - destinatário tem garantia que mensagem teve origem em emissor específico, e.g. assinaturas, MACs
 - nos canais seguros estende-se a várias mensagens/pacotes e à sequência completa de mensagens enviadas
 - não existe um requisito de tempo: mensagem enviada agora/recentemente
- Autenticação de entidades (esta aula e a próxima):
 - Bob tem a certeza que Alice participou ativamente/agora num passo processual
 - exemplo: autenticação para acesso a um website, para abrir uma porta, etc
 - geralmente seguido de um passo de autorização: Bob decide se Alice deve ter acesso ao recurso

Solução Criptográfica

- Protocolo de Desafio-Resposta (Challenge-Response)
 - Bob cria um desafio fresco, e.g., um valor aleatório, e envia para Alice
 - Alice assina digitalmente ou calcula um MAC sobre o desafio e devolve
 - Bob verifica assinatura/MAC, potencialmente dentro de um tempo limite
- Que propriedades devem ter os componentes?
 - desafio imprevisível: impossível replay
 - assinatura digital, MAC: chaves autênticas, impossível falsificar

Solução Criptográfica

- Protocolo de Desafio-Resposta (Challenge-Response)
 - Bob cria um desafio fresco, e.g., um valor aleatório, e envia para Alice
 - Alice assina digitalmente ou calcula um MAC sobre o desafio e devolve
 - Bob verifica assinatura/MAC, potencialmente dentro de um tempo limite
- Observação:
 - se for um utilizador humano => intervém no processo?
 - como se autenticam humanos?

Autenticação de Utilizadores

- Cenário
 - Utilizador humano (Alice) pretende aceder a recurso on-line
 - Recurso disponível em servidor remoto (Bob)
 - Pretende-se garantir participação ativa/na hora do utilizador

Autenticação de Utilizadores

- Solução:
 - Utilizador humano fornece identidade e pede acesso
 - Servidor remoto solicita prova de identidade
 - Utilizador fornece prova de identidade
- Observação:
 - estrutura parecida com desafio-resposta
 - pode ser implementado com auxílio de protocolo criptográfico anterior

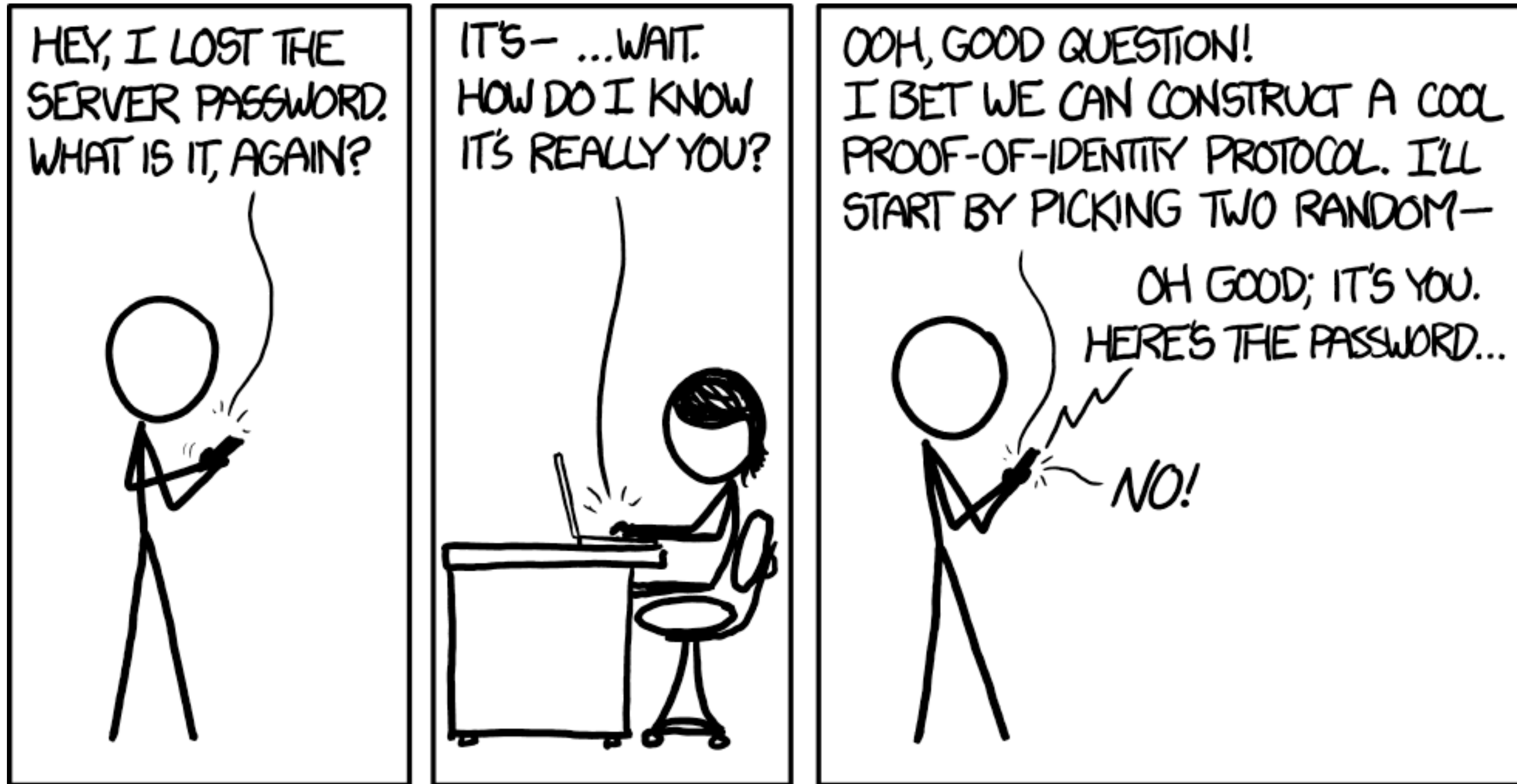
Autenticação de Utilizadores

- Provas de identidade:
 - algo que se sabe/conhece (e.g., password)
 - algo que se possui (e.g., smartcard, telemóvel, etc.)
 - algo que se é intrinsecamente (e.g., biometria)
- Podem ser utilizadas isoladamente
- Podem ser combinadas: autenticação multi-factor

Algo que se sabe

- Um segredo que apenas um humano específico conhece
 - um código secreto gerado para o efeito:
 - e.g., password, Personal Identification Number (PIN)
 - um segredo sobre a pessoa:
 - e.g., nome de animal de estimação, música preferida, etc.
- Na prática prova apenas conhecimento do segredo
 - implica assumir que apenas aquele humano conhece o segredo

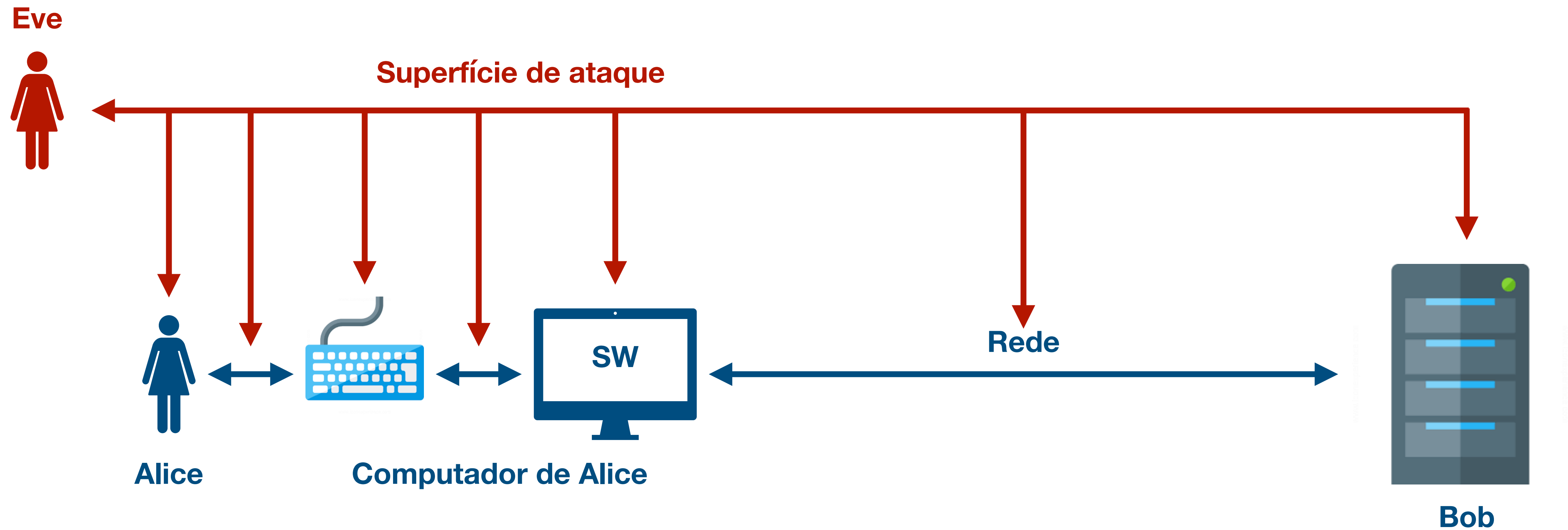
Algo que se sabe



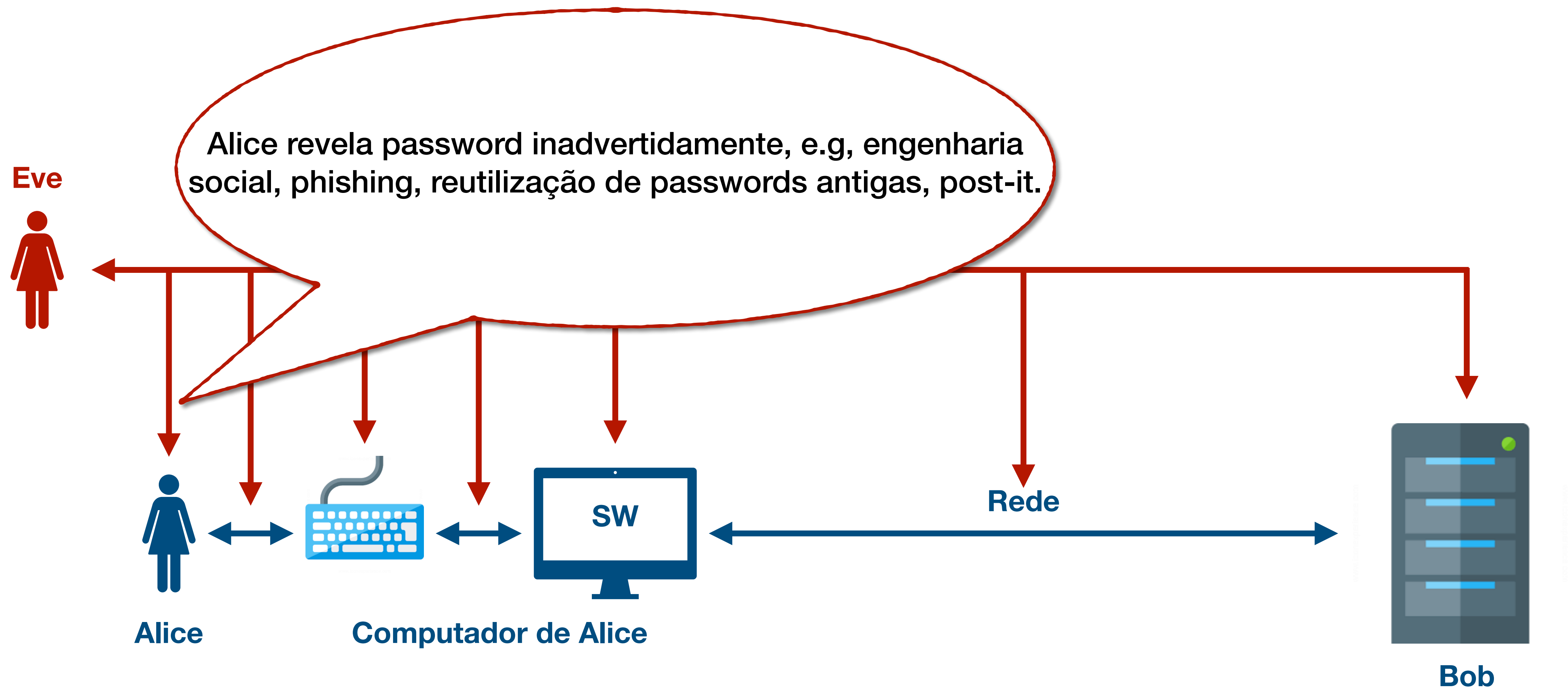
Passwords

- Solução legacy: não conseguimos ainda evoluir para melhor
- Vantagem:
 - simples: Alice prova que conhece password fornecendo-a ao sistema
- Problemas:
 - um atacante passivo pode passar a conhecer a password
 - precisamos de canal seguro para enviar password
 - um atacante ativo pode fazer-se passar pelo servidor
 - Alice precisa de ter a certeza de que está a falar com o servidor

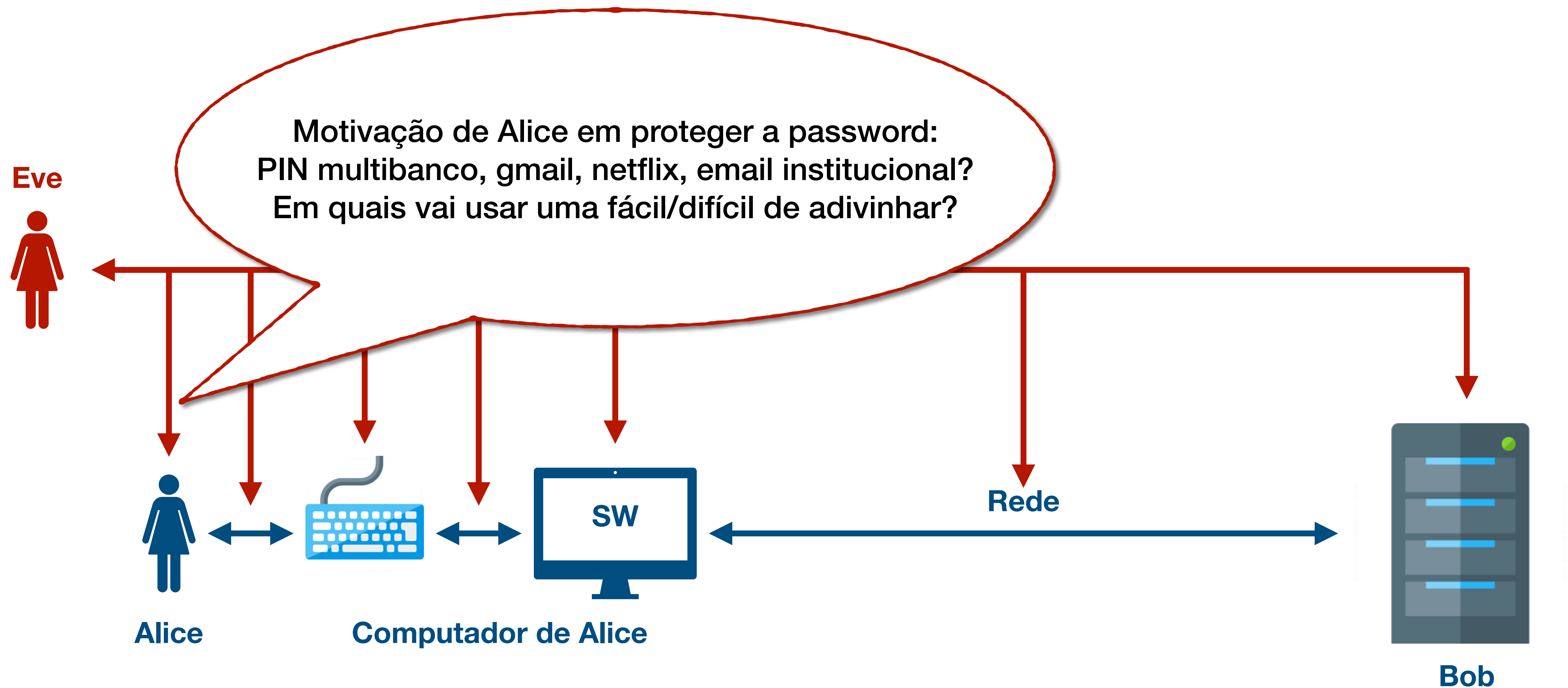
Ataques a Passwords



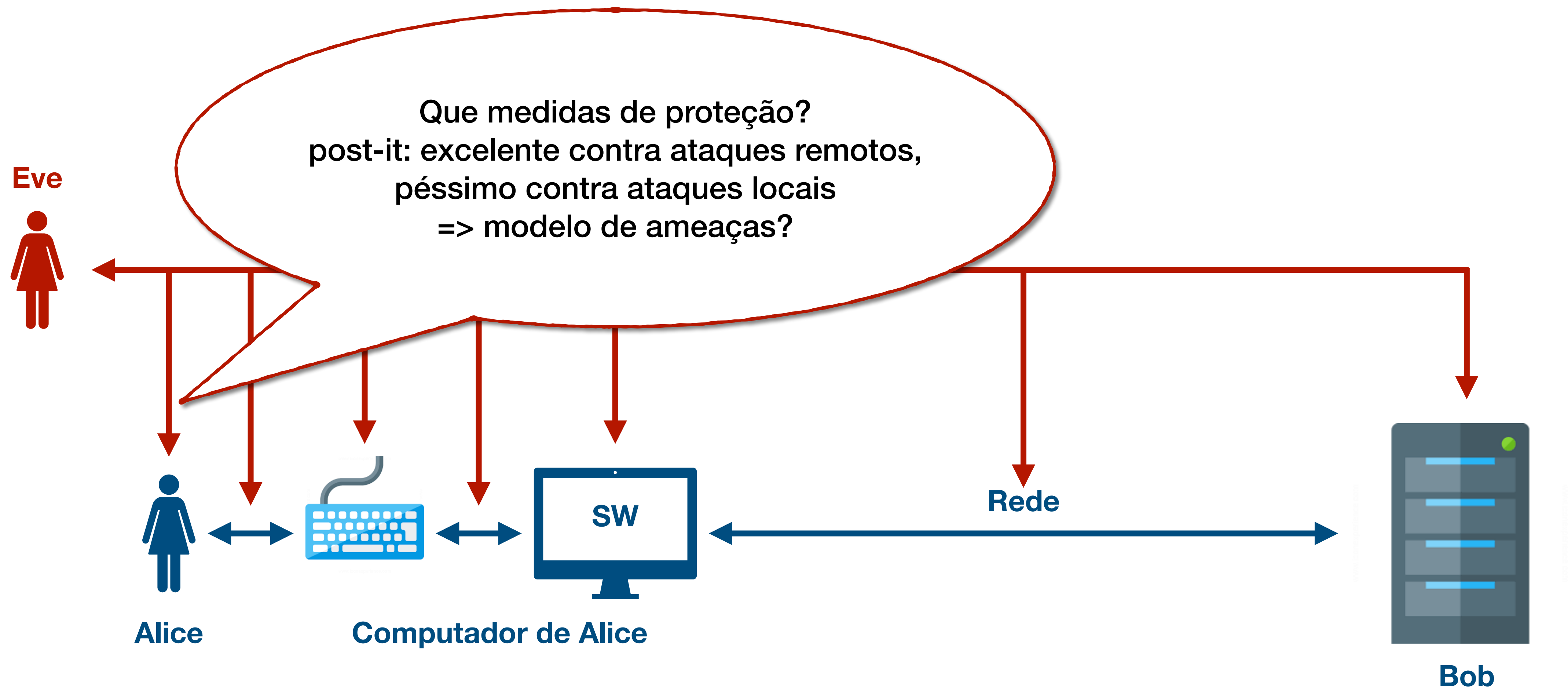
Ataques a Passwords: Alice



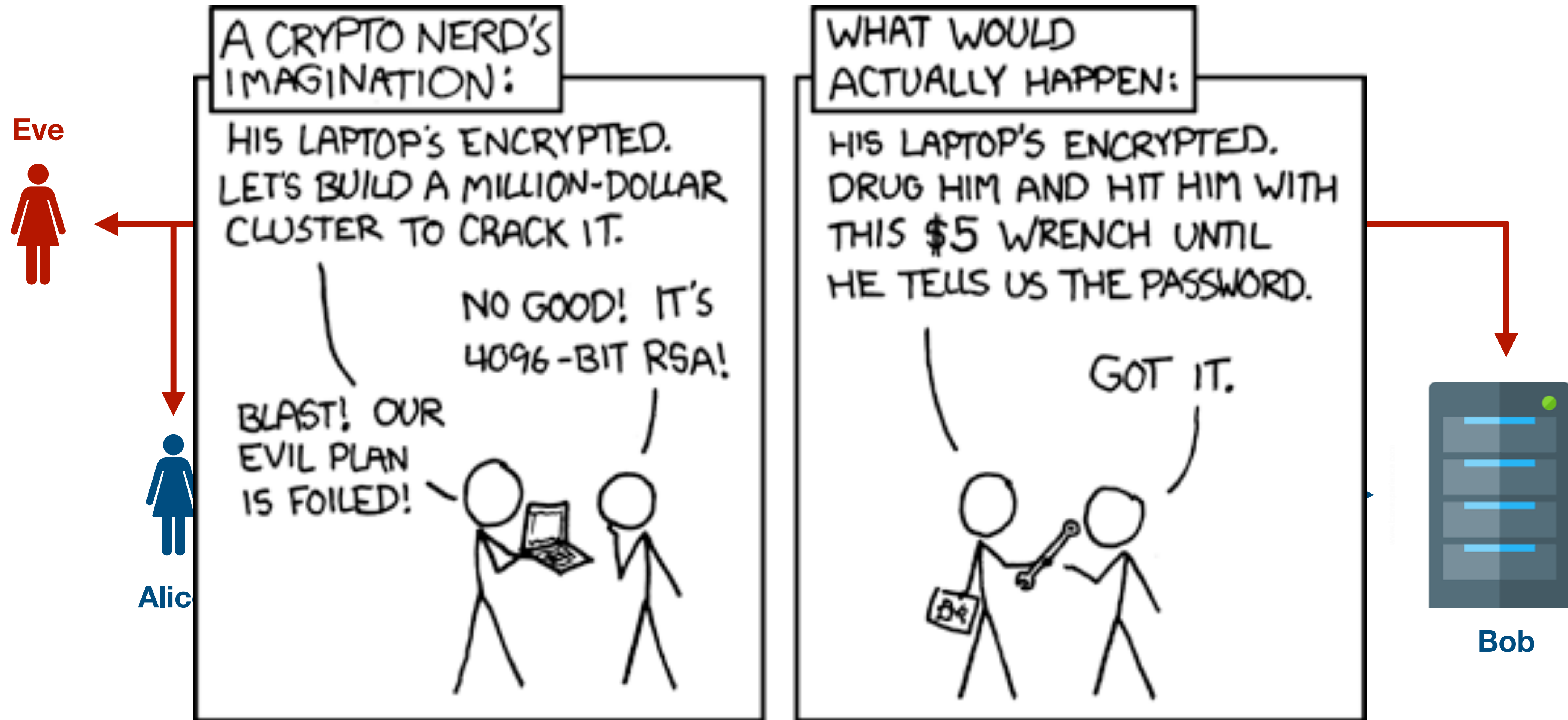
Ataques a Passwords: Alice



Ataques a Passwords: Alice



Ataques a Passwords: Alice



Passwords Fortes

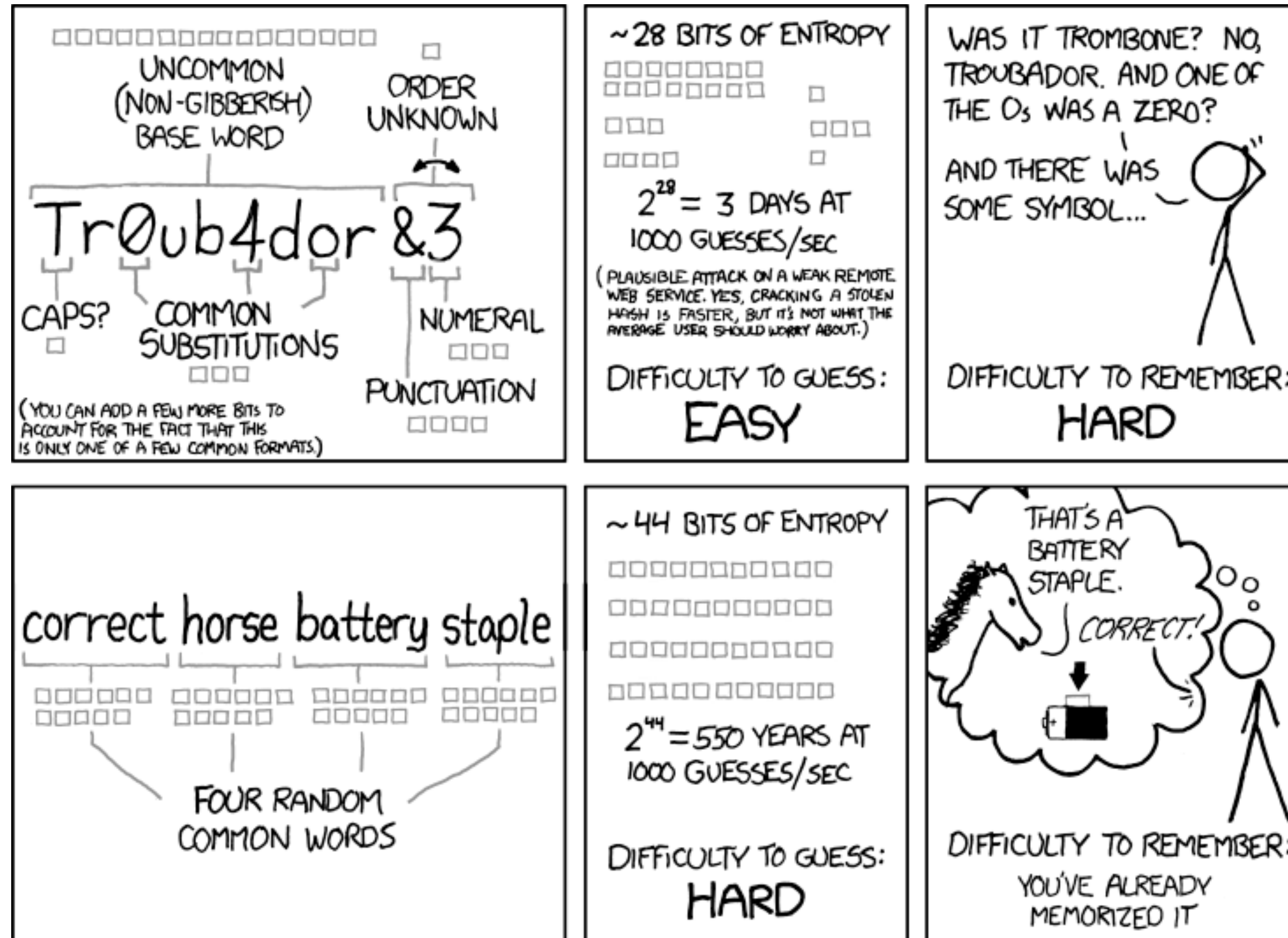
- Ideal: passwords que são difíceis de adivinhar, mas fáceis de memorizar
- Políticas típicas:
 - composição heterogênea: letras maiúsculas e minúsculas, números, símbolos
 - comprimento mínimo, período de duração máximo
 - black-list de passwords banidas
- Por vezes estas regras são contra-producentes (porquê)?

Passwords Fortes

Top 25 most common passwords by year according to SplashData

Rank	2011 ^[6]	2012 ^[7]	2013 ^[8]	2014 ^[9]	2015 ^[10]	2016 ^[5]	2017 ^[11]	2018 ^[12]	2019 ^[13]
1	password	password	123456	123456	123456	123456	123456	123456	123456
2	123456	123456	password	password	password	password	password	password	123456789
3	12345678	12345678	12345678	12345	12345678	12345	12345678	123456789	qwerty
4	qwerty	abc123	qwerty	12345678	qwerty	12345678	qwerty	12345678	password
5	abc123	qwerty	abc123	qwerty	12345	football	12345	12345	1234567
6	monkey	monkey	123456789	123456789	123456789	qwerty	123456789	111111	12345678
7	1234567	letmein	111111	1234	football	1234567890	letmein	1234567	12345
8	letmein	dragon	1234567	baseball	1234	1234567	1234567	sunshine	iloveyou
9	trustno1	111111	iloveyou	dragon	1234567	princess	football	qwerty	111111
10	dragon	baseball	adobe123 ^[a]	football	baseball	1234	iloveyou	iloveyou	123123
11	baseball	iloveyou	123123	1234567	welcome	login	admin	princess	abc123
12	111111	trustno1	admin	monkey	1234567890	welcome	welcome	admin	qwerty123
13	iloveyou	1234567	1234567890	letmein	abc123	solo	monkey	welcome	1q2w3e4r
14	master	sunshine	letmein	abc123	111111	abc123	login	666666	admin
15	sunshine	master	photoshop ^[a]	111111	1qaz2wsx	admin	abc123	abc123	qwertyuiop
16	ashley	123123	1234	mustang	dragon	121212	starwars	football	654321
17	bailey	welcome	monkey	access	master	flower	123123	123123	555555
18	passw0rd	shadow	shadow	shadow	monkey	passw0rd	dragon	monkey	lovely
19	shadow	ashley	sunshine	master	letmein	dragon	passw0rd	654321	7777777
20	123123	football	12345	michael	login	sunshine	master	!@#\$%^&*	welcome
21	654321	jesus	password1	superman	princess	master	hello	charlie	888888
22	superman	michael	princess	696969	qwertyuiop	hottie	freedom	aa123456	princess
23	qazwsx	ninja	azerty	123123	solo	loveme	whatever	donald	dragon
24	michael	mustang	trustno1	batman	passw0rd	zaq1zaq1	qazwsx	password1	password1
25	Football	password1	000000	trustno1	starwars	password1	trustno1	qwerty123	123qwe

Passwords Fortes



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

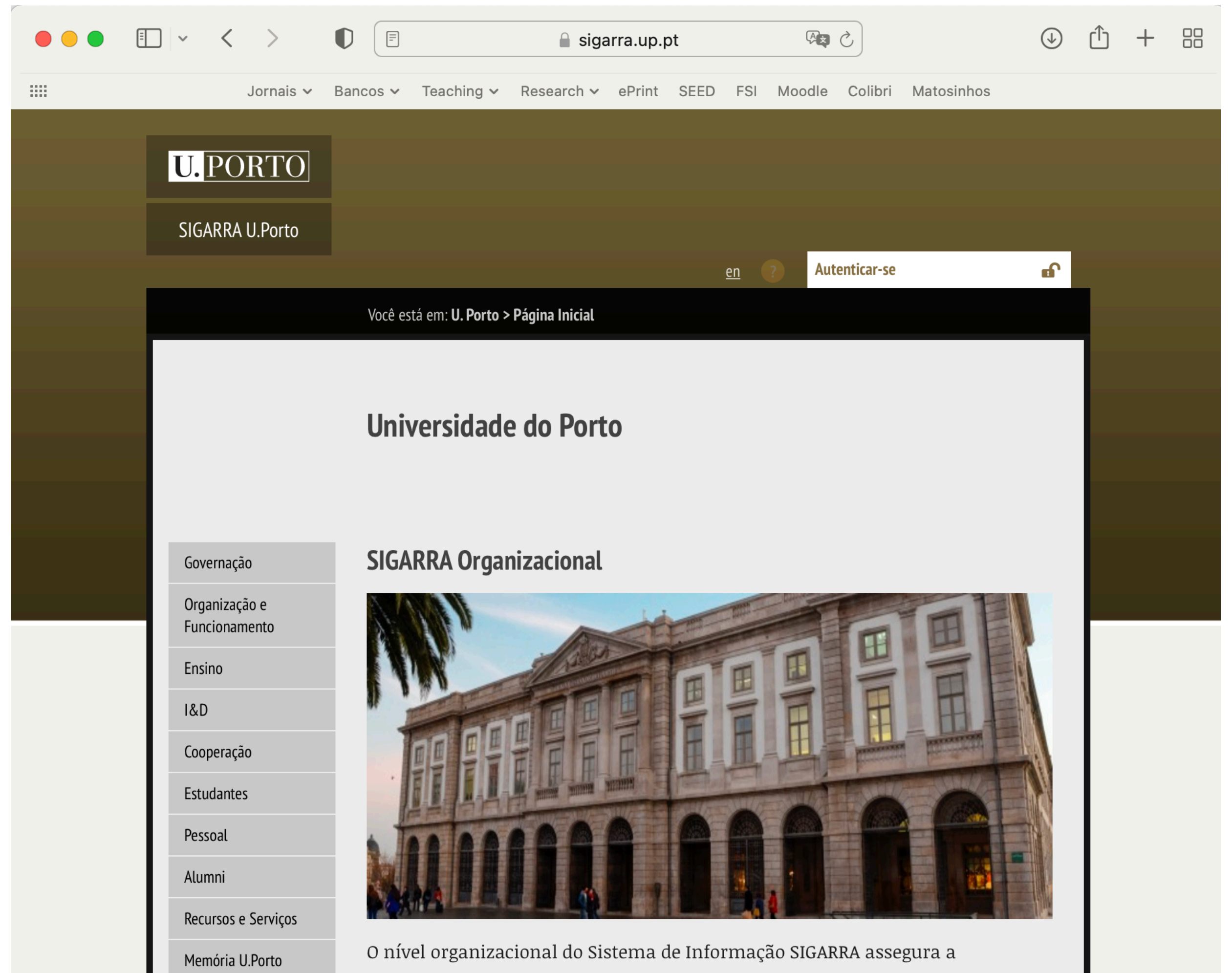
Phishing

- Eve leva Alice a simplesmente dar-lhe a password
- Vector de ataque:
 - a Alice pode não perceber que está a fazer login no servidor errado



Phishing

- O que é que o nome de domínio no browser diz?
- HTTPS => certificado válido com nome de domínio
- Ligados a servidor UP?
 - Não necessariamente
- Ligados a servidor que
 - possui chave privada
 - associada a certificado
 - com DN = sigarra.up.pt



Phishing

- O que é que o nome de domínio no browser diz?

- HTTPS => certificado
nome de domínio

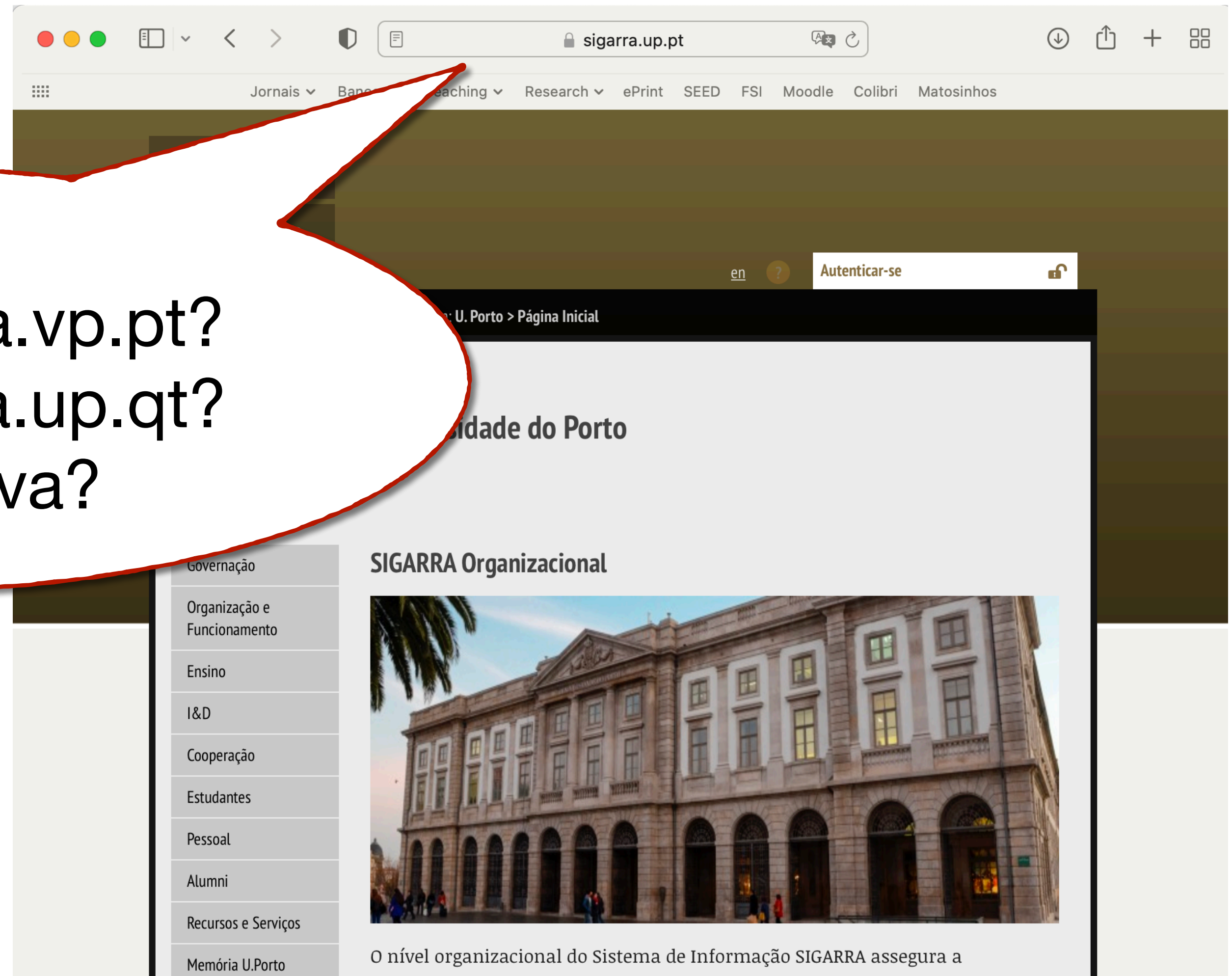
- Ligados a

E se fosse sigarra.vp.pt?
E se fosse sigarra.up.qt?
Quem reparava?

- Não necessariamente

- Ligados a servidor que

- possui chave privada
- associada a certificado
- com DN = sigarra.up.pt



Phishing

- O que é que o nome de domínio no browser diz?

- HTTPS => certificado
nome de domínio

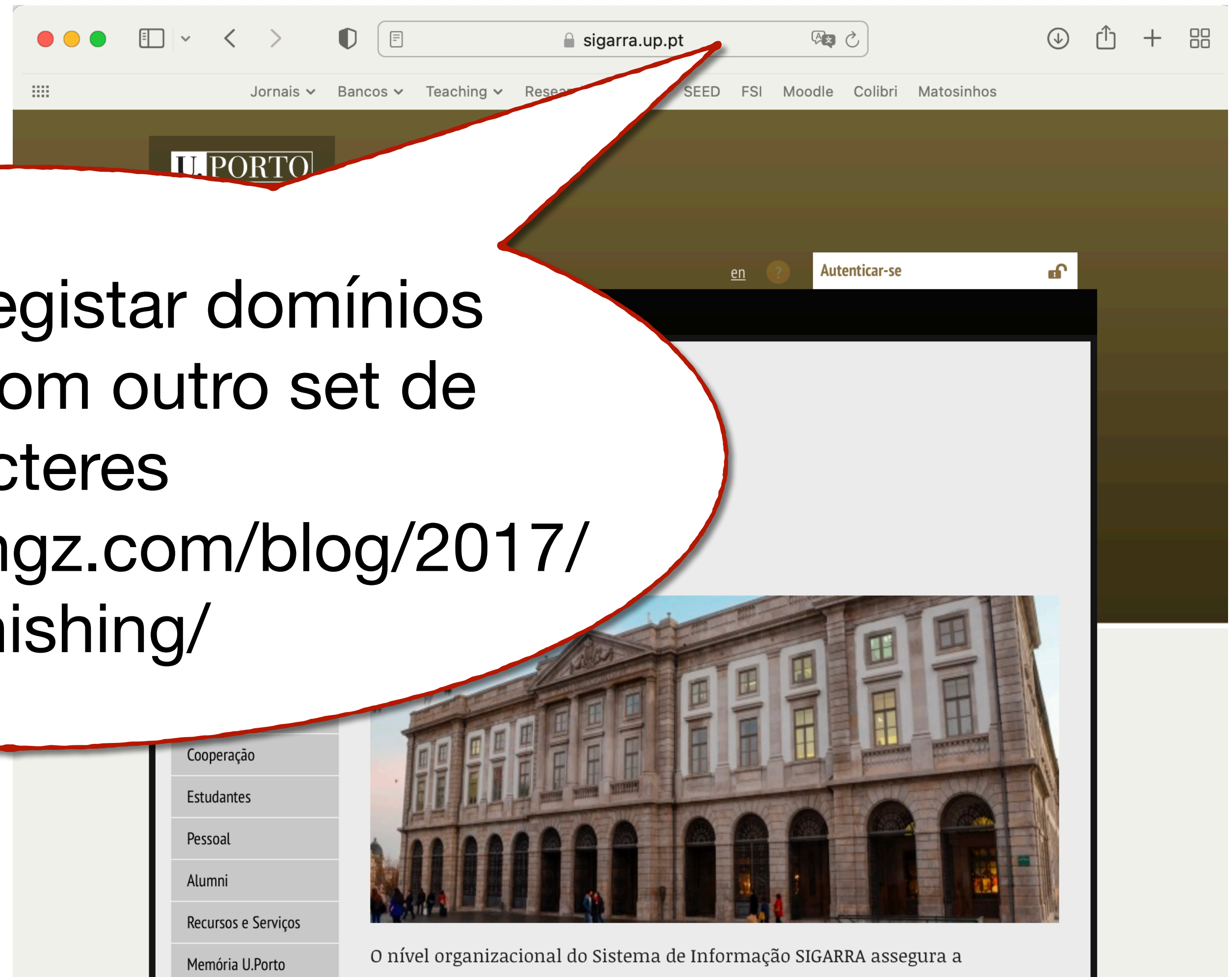
- Ligados a

Homoglyphs: registar domínios
parecidos ou com outro set de
caracteres

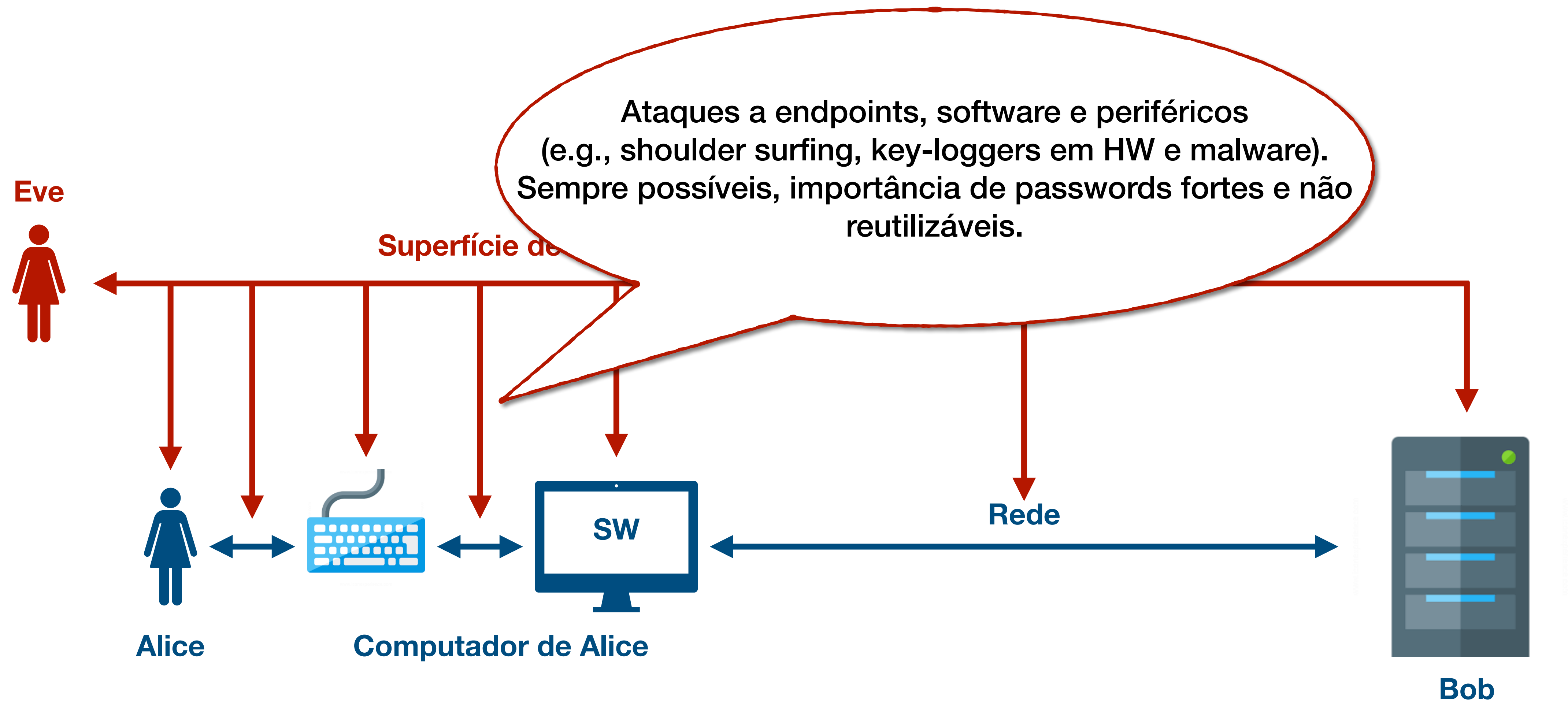
- Pressupõe
verifica DNS
- <https://www.xudongz.com/blog/2017/idn-phishing/>

- Browsers procuram tornar isso
mais fácil

- Mesmo assim é possível tentar
enganar o utilizador



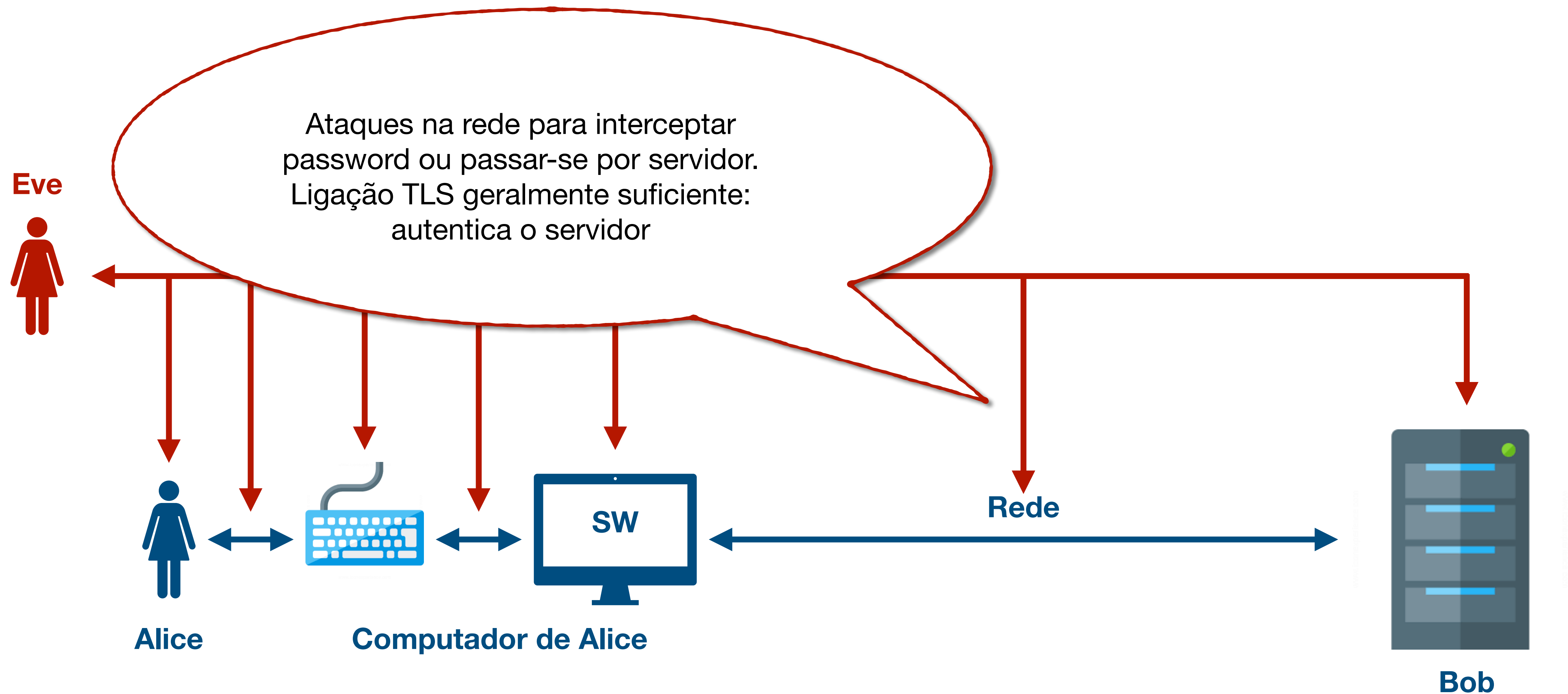
Ataques a Passwords: Endpoint



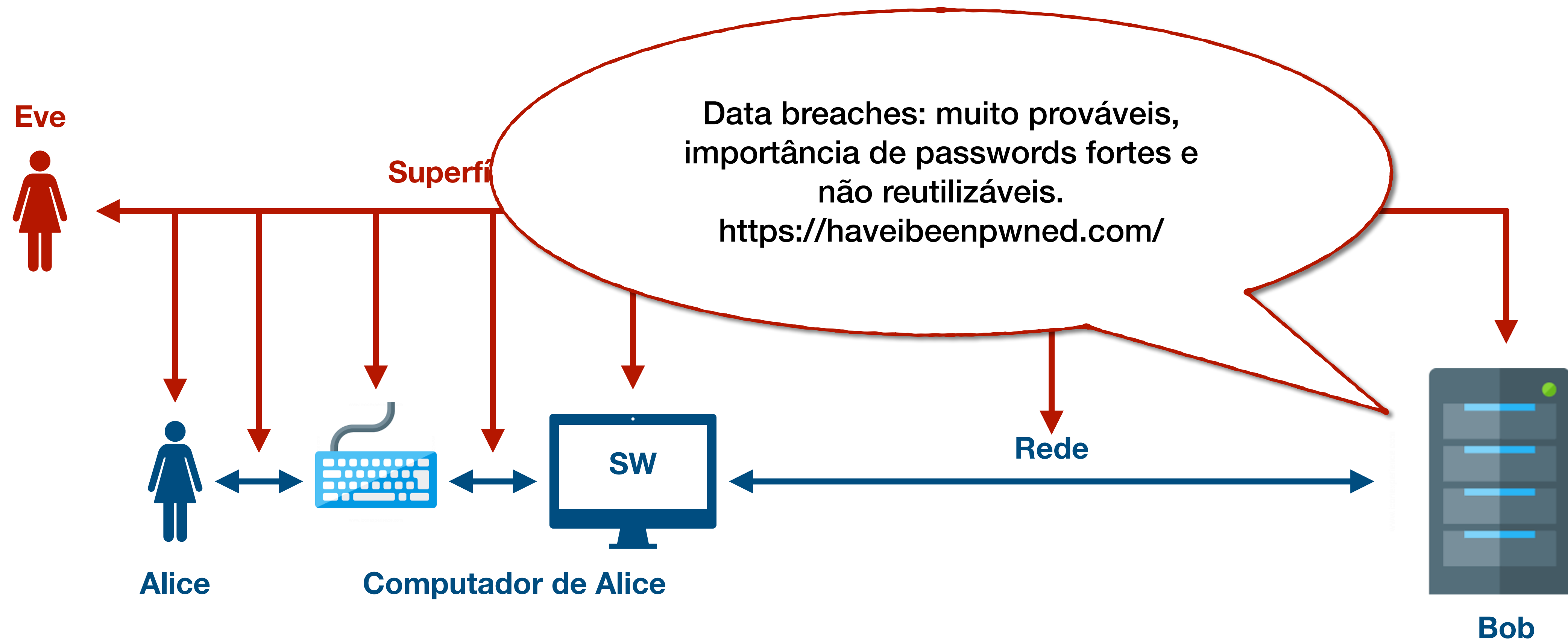
Ataques a Passwords: Endpoint

- HW keyloggers: dispositivos colocados entre teclado e computador
- SW keyloggers:
 - malware que intercepta keystrokes
- Outros tipos de malware:
 - passwords em memória, e.g., clipboard
 - passwords armazenadas, e.g. passwords.txt, cache dos browsers, etc.

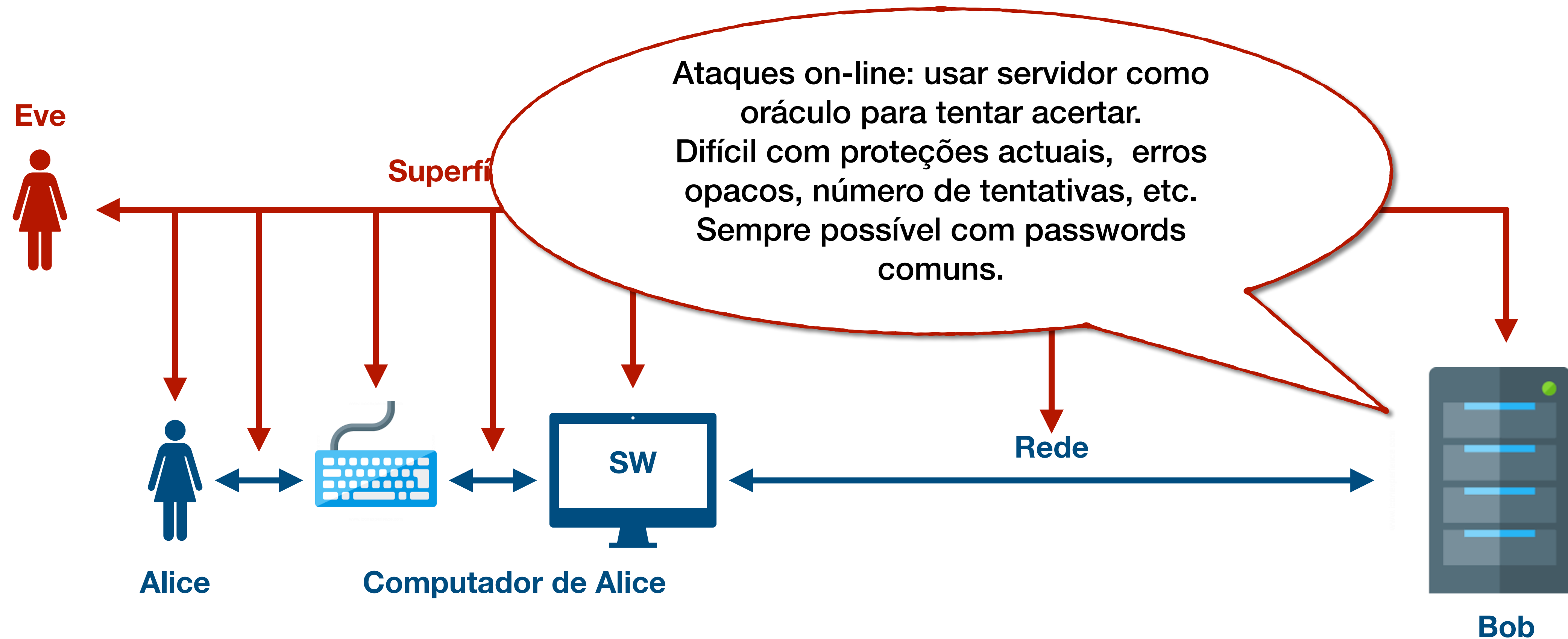
Ataques a Passwords: Rede




Ataques a Passwords: Servidor



Ataques a Passwords: Servidor



Armazenamento de passwords: server-side

- Qual o impacto de uma data breach?
 - que informação revela sobre passwords armazenadas num servidor?
 - depende da forma como são geridas/armazenadas
- solução *naive*:
 - lista de password/username
 - data breach revela toda a informação
- se passwords reutilizadas noutro sistema => 

Armazenamento de passwords: server-side

- Importante: o servidor não precisa de saber a password!
 - apenas precisa de ser capaz de reconhecer a password correta
- O servidor pode, e.g., guardar apenas $H(pw)$
 - se H for uma função de hash criptográfica
 - tem geralmente uma propriedade \Rightarrow difícil encontrar pré-imagens:
 - difícil encontrar pw dado $H(pw)$
 - se pw for difícil de adivinhar (porquê?)

Ataques de Dicionário

- Armazenar $H(\text{pw})$ ainda não é a solução ideal
 - dado $H(\text{pw})$ um atacante pode verificar os seus palpites
 - passwords iguais são armazenadas da mesma maneira
 - em servidores diferentes, entre dois utilizadores
- Ataques de dicionário:
 - dicionário \Rightarrow coleção de passwords possíveis/prováveis
 - tentar todas as possibilidades até encontrar a correta, ou ...
 - pré-computar todos os $H(\text{pw})$ no dicionário e construir uma hash-table:
 - dado $H(\text{pw}^*)$ para pw^* desconhecido basta procurar na hash-table!

Ataques de Dicionário

- Quantas passwords pode um atacante testar?
 - passwords apenas com letras e dígitos
 - cada posição => 26 + 26 + 10 => 64 opções
 - $64^n = 2^{6n}$ passwords de tamanho n
 - Se $n = 6$, 2^{36} passwords possíveis (todas!)
 - tabela de 1TB com hashes de 160 bits

HACKERS RECENTLY LEAKED **153 MILLION** ADOBE USER EMAILS, ENCRYPTED PASSWORDS, AND PASSWORD HINTS. ADOBE ENCRYPTED THE PASSWORDS IMPROPERLY, MISUSING BLOCK-MODE 3DES. THE RESULT IS SOMETHING WONDERFUL:

USER	PASSWORD	HINT	
4e18acc1ab27a2d6		WEATHER VANE SWORD	<input type="text"/>
4e18acc1ab27a2d6		NAME1	<input type="text"/>
4e18acc1ab27a2d6	a0a2876eb1ea1fca	DUH	<input type="text"/>
8bab629e06eb6d		57	<input type="text"/>
8bab629e06eb6d	a0a2876eb1ea1fca	FAVORITE OF 12 APOSTLES	<input type="text"/>
8bab629e06eb6d	85e9da81a8a78adc	WITH YOUR OWN HAND YOU HAVE DONE ALL THIS	<input type="text"/>
4e18acc1ab27a2d6		SEXY EARLOBES	<input type="text"/>
1ab29ae86da6e5ca	7a2da0a2876eb1e	BEST TOS EPISODE	<input type="text"/>
		SUGARLAND	<input type="text"/>
		NAME + JERSEY #	<input type="text"/>
		ALPHA	<input type="text"/>
		OBVIOUS	<input type="text"/>
		MICHAEL JACKSON	<input type="text"/>
		HE DID THE MASH, HE DID THE PURLOINED	<input type="text"/>
		FAV. LATER-3 POKEMON	<input type="text"/>

THE GREATEST CROSSWORD PUZZLE
IN THE HISTORY OF THE WORLD

Ataques de Dicionário

- Como tornar os ataques de dicionário mais difíceis
- Observação para o caso $H(pw)$:
 - uma tabela serve para todos os utilizadores e todos os servidores!
- Salt!
 - armazenar $(r, H(r||pw))$ em que r se chama *salt* (como se verifica? \Rightarrow recalculando $H(r||pw)$)
 - aleatório, idealmente para cada utilizador
 - agora é precisa uma tabela para cada valor de salt possível
 - já não é realista pré-computar \Rightarrow tempo de trabalho *depois* de entrar no servidor
 - data breaches não ajudam a pré-computar tabelas atacar outros servidores

Ataques de Dicionário

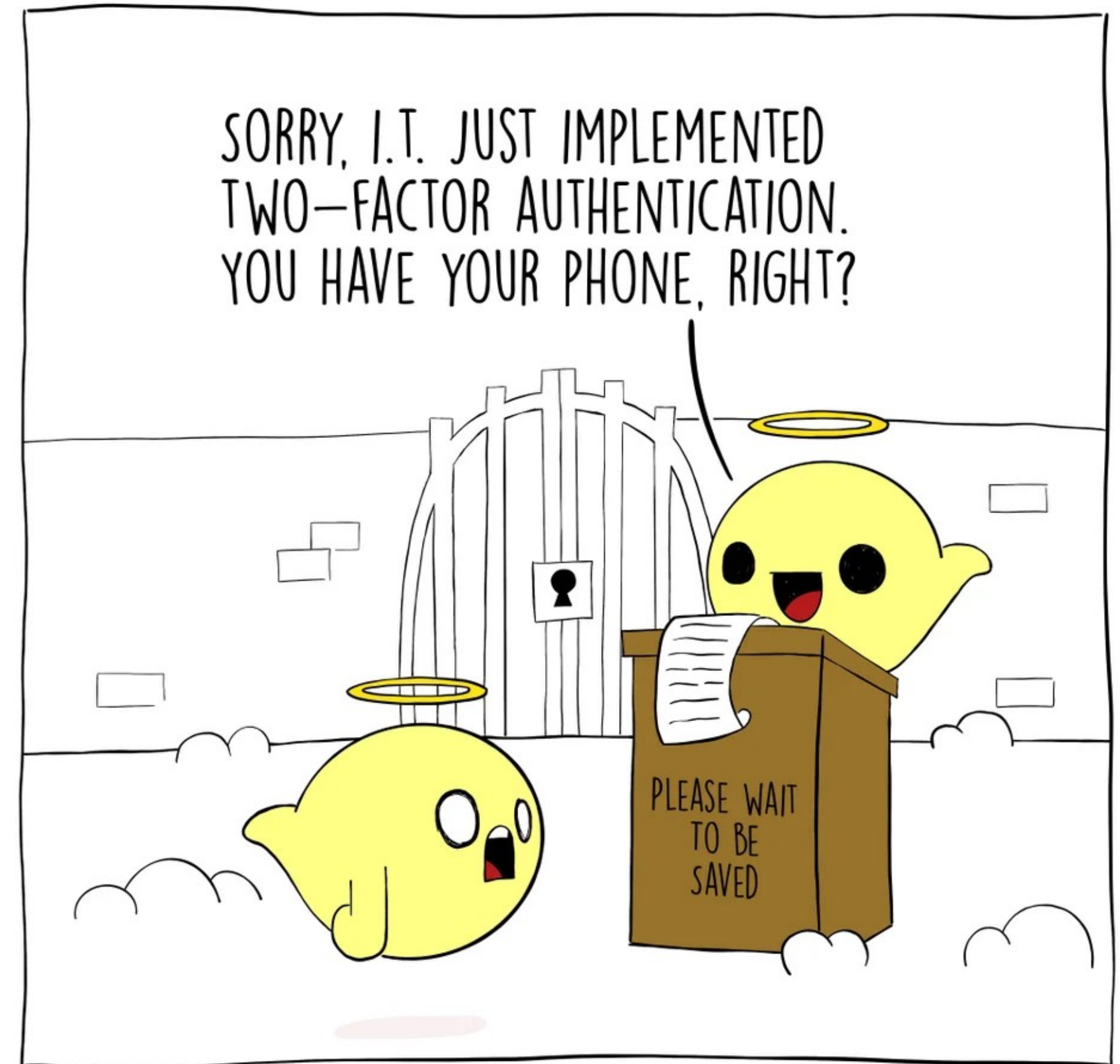
- A solução anterior pode não ser ainda suficiente:
 - hardware especializado para realizar ataques por dicionário
 - atacantes poderosos (estados) podem construir HW dedicado
- Em 2012 uma implementação documentada:
 - 2^{34} hashes por segundo (NTLM hash => windows)
 - 2^{33} hashes por segundo (MD5), 2^{31} hashes por segundo para SHA-1

Ataques de Dicionário

- Medida de mitigação adicional:
 - utilizar funções de hash especialmente pesadas (tempo, recursos)
 - não afecta o servidores significativamente
 - impacto significativo em ataques de dicionário
 - PBKDF2, bcrypt, scrypt

Multi-factor

- Autenticação multi-factor
 - Defesa em profundidade
 - Password vai ser quebrada
 - Preparar sistema para utilizações suspeitas/operações críticas
- Utilizar fatores adicionais para confirmação e/ou deteção de problemas



@THEIMMORTALGRIND