Autenticação 1

Challenge-Response (Desafio-Resposta)

Garante a autenticação em tempo real.

- 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 a assinatura/MAC potencialmente dentro de um limite.

Propriedades

- O desafio tem de ser imprevisível, assim já não é possível fazer replay.
- A assinatura digital e MAC possuem chaves autênticas que são impossíveis de falsificar.

Autenticação de utilizadores

Pode ser feito com base na abordagem de desafio-resposta, mas usamos mais atualmente:

- Algo que se sabe/conhece (e.g password)
- Pode-se utilizar algo que possui (e.g telemovel, smartcard, etc)
- Algo que é intrinseco (e.g biometria)

Estas abordagem pode ser usada isoladamente, e.g password, ou pode ser utilizada em conjunto. Esta última é o que chamamos de autenticação multi-factor.

> Mesmo que muitas vezes estes fatores não sejam requisitados explicitamente, outras operações que podem utilizar, por exemplo,

> inteligência artificial, estão a decorrer no background para definir se é provável que seja o utilizador a fazer o acesso ou não.

Passwords

- **Vantagem:** é muito simples.
- **Problemas:** um atacante passivo pode passar a conhecer a password. Ou um atacante ativo pode fazer-se passar pelo servidor e assim a Alice precisa saber que está a falar com o servidor com certeza.

Phishing

Convencer a pessoa a dar password num site diferente. Faz-se o site completamente igual ao original e o url é muito semelhante. Com isso, o user pode pensar que está a aceder ao site correto, quando na verdade está num servidor malicioso.

> Na web uma rede com TLS é o suficiente para nos proteger.

Armazenar passwords

Uma solução naive é fazer uma lista de password/username.

Uma databreach revela toda a informação.

O servidor **não precisa saber a password** ele apenas precisa reconhecê-la. Uma das soluções é guardar apenas um hash da password, o que é difícil de inverter. Para a hash ser difícil de inverter, temos de ter uma grande entropia.

Ataques de dicionário

O dicionário possui uma coleção de passwords possíveis/prováveis. Ele tenta todas as possibilidades no dicionário até encontrar a correta.

Ainda há a pré-computação. Há uma tabela gigantesca onde já calculamos previamente a hash de várias passwords. Depois quando encontramos a hash de uma

password, basta verificar se ela já está na hash table.

Qual o tamanho que uma password deve ter?

Se passwords com apenas letras e digitos, temos 26+26+10 ⇒ 64 opções.

Portanto, temos 64^n = 2^6n passwords de tamanho n.

Se n = 6, temos 2^36 passwords possíveis: todas! Isto é uma tabela com 1 tb.

Como dificultar este ataque

Podemos usar Salt: um valor aleatório r.

O valor do r vai estar armazenado junto com o hash. Depois de uma databreach, sabe-se qual o salt.

Mesmo com salt, o ataque do dicionário ainda é possível, mas evitamos duas coisas:

- Utilizar uma pré computed table para guardar hash de passwords
- Comparar hashes vazadas de outros servidores

Podemos ainda usar outras funções de hash que são mais pesadas (tempo, recursos). Não tem um efeito muito significativo no servidor, mas faz diferença quando se tenta fazer o ataque do dicionário.

Perguntas (by Diana Cristina)

1) Diga porque razão é preciso soluções diferentes para a autenticação de origem de mensagens e de entidades.

▼ Resposta

Queremos aplicar o conceito de defesa em profundidade. Perante a falha de um dos meios de proteção teremos outras formas de garantir a segurança.

2) Qual a solução criptográfica usada para autenticação de entidades? Explique-a.

▼ Resposta

A solução é utilizar o desafio-resposta. Isto é utilizado em protocolos MAC, por exemplo.

3) Como é feita a autenticação de utilizadores humanos?

▼ Respota

Pode ser feita de várias formas: passwords, smartcards, etc. Pode ser algo que a pessoa possua, algo que ela conheça um algo intríseco como a biometria.

4) O que é phishing?

▼ Resposta

É quando alguem convence a pessoa a dar a fornecer credenciais. Um jeito interessante (baseado em fatos reais) é alguem lhe enviar um link do discord para ter conta **nitro** (premium). O link redireciona a pessoa a um site exatamente igual a página de login do discord. A pessoa então iria tentar fazer o login quando na verdade está a fornecer as credenciais para o atacante.

5) Enumere possíveis ataques a passwords.

▼ Resposta

- phishind
- Ataque do dicionario
- Dicionário hashed

6) É suficiente armazenar a hash em vez da password original?

▼ Resposta

Não é o suficiente. Abre a possibilidade para o atacante criar um dicionário com passwords já hashed. Assim a eficiência do ataque torna-se muito grande, pelo que basta verificar se password está contida no dicionário. Além disso o atacante pode identificar passwords iguais em base de dados diferentes.

7) Como se pode tornar mais difícil ataques por dicionário?

▼ Resposta

Pode-se impedir o ataque de passwords pré computadas utilizando salt. E esta contramedida ainda deixa o ataque do dicionário mais pesado, pelo que além de se computar a hash da password, deve-se ter em consideração o salt.