CAN304 W6

Basic authentication mechanisms

Authentication mechanisms (三类):

- Something you know
  - Passwords, challenge/response
- Something you have (token)
  - Smart cards, electronic keycard, physical key
- Something you are (static biometrics)
  - Fingerprint, retina, face
- Something you do (dynamic biometrics)
  - Voice pattern, handwriting, typing rhythm

Something you know

Passwords

最古老和最常用的安全机制之一，通过要求用户生成机密来对用户进行身份验证，机密通常只有用户和认证者知道。

Problems with passwords:

- 它们必须是不可猜测的，但人们很容易记住
- 如果网络将远程设备连接到计算机，则容易受到密码嗅探器 (password sniffers) 的影响
- 除非很长，否则暴力攻击通常会对它们起作用

Proper use of passwords:

密码应该足够长；密码应包含非字母字符；密码应不可猜测；密码应经常更改；密码不应被写下来；密码不应被共享；难以同时实现所有这些。

Passwords and single sign-on:

许多系统只要求输入一次密码，生成的身份验证将持续整个“会话”。不过其他人可以使用经过身份验证的机器。

Handling passwords

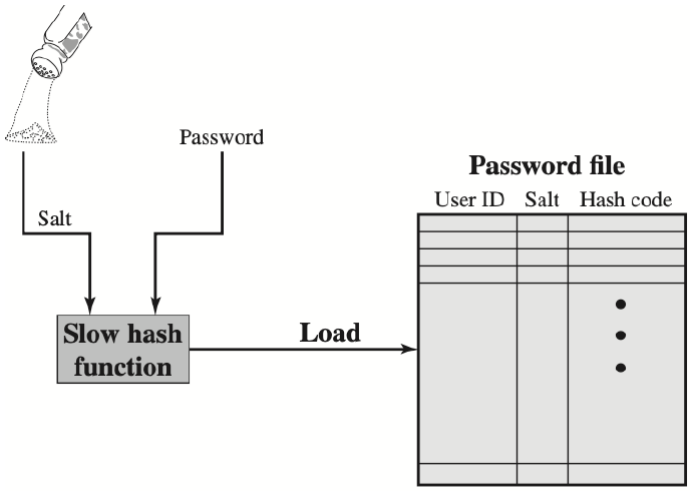
操作系统必须能够在用户登录时检查密码，但操作系统不一定要存储密码。操作系统可以储存密码的 hash 版本，然后进行对比即可。

不过这样也不一定安全。假如攻击者有一个密码本，里面记载了所有可能密钥的 hash value，他可以根据密码本找到对应的真实密码 (dictionary attacks)。

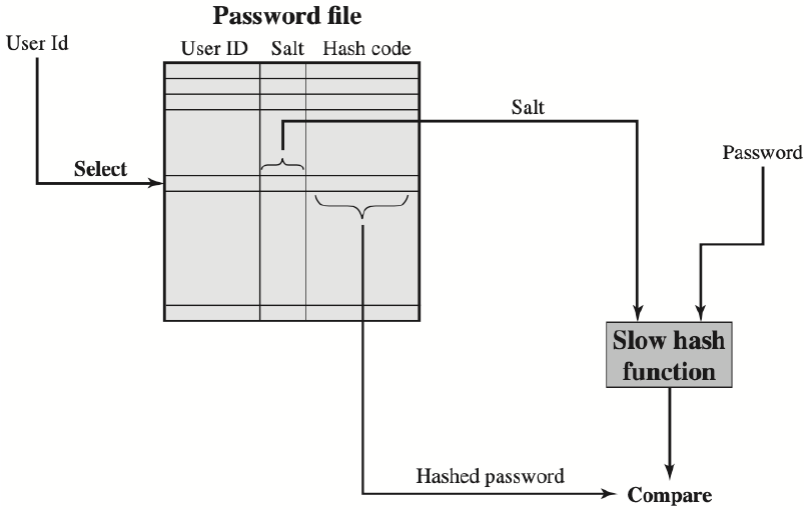
Salted passwords

将纯文本密码与随机数 (salt) 组合，这用于区分相同的密码 (这样生成的 hash value 就不同了)。

UNIX Password Scheme



(a) Loading a new password

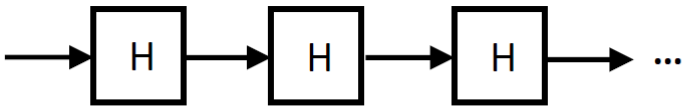


(b) Verifying a password

UNIX 直接在系统中给密码加 salt，验证密码时根据用户 ID 来取得 salt，和密码结合进行验证 (防止攻击者根据 hash value 找到密码)。

S/Key: One-Time Password

One-Time Password：一个密码就用一次，每次都使用不同的密码。



Server (verifier) generates

x, H(x), H^2(x), ., H^{n+1}(x)

Server only stores H^{n+1}(x) and discards others

User (prover) is given n one-time passwords:

H(x), H^2(x), ..., H^n(x)

User uses in the reverse order

服务器生成一个随机数 x，然后算出 H(x)，H(H(x)) = H^2(x)，...，H^{n+1}(x)，然后服务器只储存 H^{n+1}(x)。用户则有一系列 one-time passwords H(x)，...，H^n(x)，用户以逆序使用密码 (H^n(x)，...，H(x))。

例如：用户首先使用 H^n(x) 作为密码，计算 hash value 后为 H^{n+1}(x)，服务器验证自己储存的密码 H^{n+1}(x)，一致则通过；之后，密码 H^n(x) 被废弃，用户使用 H^{n-1}(x) 作为密码，服务器中的密码 H^{n+1}(x) 也被更新为 H^n(x) .....

***Compromise yields outdated passwords* (入侵会导致密码过时)。**

Other user authentication：双因素/多因素身份验证 (Two-factor/multi-factor authentication，密码+短信)，生物特征认证 (Biometric authentication，指纹)，其他用户认证。

Something you have

Identification devices：计算机可读的智能卡或其他硬件设备，通过向计算机提供设备进行身份验证 (数字银行使用的USB密钥)。

服务器如何确定远程用户的身份?

类似之前的 smart lock system，服务器发送给使用 identification devices 的设备一个 challenge (MAC tag 或 签名)，设备解决这个 challenge，并告知服务器。

通常用户必须输入密码才能激活卡。

Problems with identification devices

- 如果丢失或被盗，您无法对自己进行身份验证，但是拿到设备的其他人可以，因此通常与密码结合使用以避免此问题
- 除非做得巧妙，否则容易受到嗅探攻击 (sniffing attacks)
- 需要特殊硬件

Biometric authentication

Biometric authentication:

- 尝试根据独特的物理特征对个人进行身份验证
- 基于模式识别
- 与密码和 tokens 相比，技术上复杂且昂贵

使用的物理特性包括:

Facial characteristics，Fingerprints，Hand geometry，Retinal pattern，Iris，Signature，Voice。

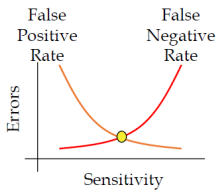
Biometric system：Enrollment -> Verification -> Identification (先登记，然后是训练，最后是识别)。

Problems with biometric authentication

- 通常需要非常特殊的硬件
- 可能并不像你想象的那么万无一失
- 许多物理特性在实际使用中变化太大
- 通常对程序或角色进行身份验证没有帮助
- 当它破裂时会发生什么？毕竟，您只有两个视网膜

Usability vs. security

- False positives (误报)
    - 在不应该匹配的时候进行匹配
- False negatives (漏报)
    - 在该匹配的时候没匹配上



The Crossover Error Rate (CER)

Generally, the lower the CER is, the better the system

But sometimes one rate more important than the other

Biometrics and usability

- 始终在误报 (false positives) 与漏报 (false negatives) 之间进行权衡
- 对于消费类设备，漏报非常非常糟糕 - 人们丢弃了不允许合法用户进入的设备