EPFL

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

Computer Security (COM-301)

Authentication – Passwords & Biometrics

Interactive Exercises

Carmela Troncoso
SPRING Lab
carmela.troncoso@epfl.ch

Authentication aboard the Enterprise

Consider the following authentication exchange in which Spock uses his password 'LongAndProsper' to prove his identity to Kirk:

```
Spock ---- (Spock, 'I want to login') ----> Kirk
Spock <--- Hash(Spock) ----- Kirk
Spock ---- Enc('LongAndProsper', Hash(Spock)) ----> Kirk
```

Which of the following statements is correct?

(a) Hash(Spock) is not a good challenge because it will be used every time
(b) Hash(Spock) is not a good challenge because anyone can compute it
(c) The protocol is bad because the login is sent on the first message
(d) Hash(Spock) is a good challenge because hashes output random numbers

The correct statement is (a).

Facebook password onion

```
$cur  = 'password'
$cur  = md5($cur)
$salt = randbytes(20)
$cur  = hmac_sha1($cur, $salt)
$cur  = remote_hmac_sha256($cur, $secret)
$cur  = scrypt($cur, $salt)
$cur  = hmac_sha256($cur, $salt)
```

Why does Facebook use this onion?

Source: Tom Ristenpart

4

Coping with legacy!

Imagine you have stored md5('password'); to transform it into sha256('password') you need to know password
They cannot guarantee that they will be able to have all users re-introduce their passwords that they had stored as insecure md5 to store them as something else.
This onion enables them to change what is stored, without the need to have the password in the clear

Knock, knock, knocking

Cersei and Jaime meet secretly every week in the crypt. To make sure that they are each other, before opening the door they have a protocol in which:

1) Jaime knocks a particular sequence (toc, toc, toctoc)

2) Cersei replies with another sequence (toctoc, toc, toctoc)

3) Jaime replies with just (toc).

Is this safe against an eavesdropping Tyrion hidden behind the bushes? If yes, justify. If not, explain how to fix it.

Tyrion can hear and learn Jamie and Cersei's sequences, and reproduce them.

Solution: Before leaving, Jaime and Cersei agree on a new sequence for the next day, so that the sequence is never repeated (similar to including R in the message in a challenge-response protocol).

Note that "encrypting the message" or "hashing the message" are not possible solutions in this case. One cannot encrypt knocking sounds!

Securing grades

Agree or disagree with the following statement and justify your answer:

"When configuring biometrics to be used as an authentication function to secure access to students' exams grades, it is important that the system has a low false negative rate even if the system finds many false positives"

Both agreeing and disagreeing are valid. What is important is the justification according to the importance the respondent gives to having false positives (violation of privacy = other students can read grades) and false negatives (bad usability, students need to retry login many times to see their grades)

Good Biometrics

These are common attributes for an authentication mechanism

(a) Universality: everyone has them

(b) Uniqueness: everyone has a different biometric

(c) Permanence: they do not change over time

(d) Secret: they are only known to the user

(e) Unpredictable: given a biometric trait, other trait cannot be predicted

Which ones are desired for biometric authentication? Of the desirable ones, do they have any downside?

[Recall that these answers are not the only possible ones]

(a) Universality is desired to avoid discrimination
(b) Uniqueness is desired to avoid security problem in which users can log in in place of other users ("similar" to knowing the password)
(c) Permanence is desired to avoid that biometrics need to be re-enrolled
(d) Secrecy is not needed. If biometrics are not secret, one would need a liveness detection algorithm
(e) Unpredictability is not desired in principle, but can be necessary if a given biometric can be recreated from the template

Uniqueness can be undesired as it enables to link users across databases, impeding that users cannot choose to have different identities for different services
Permanence can be undersired as it prevents revocation