Security Engineering
Fall 2015

Lecture 12 – Authentication

Fabio Massacci

What is authentication?

- ***It is the process of verifying a claimed identity by r for a system entity***
- ***It consists of two main steps:***
  - Identification
    - Present an identifier to the security system
    - You annouce who you are
  - Verification
    - Presenting or generating authentication Information that provides evidence of the binding between the entity and the identifier
    - You prove who you are
- ***Remember: you are authenticating a stranger***

Means of Authentication

- ***Something the individual knows***
  - Password-based
- ***Something the individual owns***
  - Token-based
- ***Something the individual is***
  - Static biometric
- ***Something the individual does***
  - Dynamic biometrics
- ***Somewhere the individual is***
  - Location-based

Something You Know

- ***The user has to know some secret to be authenticated.***
  - password,
  - personal identification number (PIN),
  - personal information like home address, date of birth, name of mother maiden name (used e.g. by banks to authenticate customers on the phone)
- ***Password-based authentication***
  - user provides name/login and password
  - system compares password with that saved for specified login
  - authenticates ID of user wishing to log
  - AC starts from that user's ID

Password Authentication

- ***Typical issues that need to be addressed:***
  - how to get the password to the user,
  - forgotten passwords,
  - password guessing,
  - protection of the password file
- ***Dangers***
  - User accounts without passwords.
  - Unchanged default passwords.
  - Badly chosen passwords – dictionary/brute force attacks.
  - Passwords stored in the clear.
  - Passwords transmitted in the clear.
  - Users forget passwords
    - the infrastructure for re-issuing passwords can be quite expensive (if it has to be truly secure)

Password Choices

- ***users may pick short passwords***
  - e.g. 3% were 3 chars or less, easily guessed
  - system can reject choices that are too short
- ***users may pick guessable passwords***
  - so crackers use lists of likely passwords
  - e.g. one study of 14000 encrypted passwords guessed nearly 1/4 of them
  - would take about 1 hour on fastest systems to compute all variants, and only need 1 break!

Old Password Guessing Tools

- ***Hydra*** *http://www.thc.org*
  - guess all sorts of passwords, including HTTP, Telnet, and Windows logons
- ***TSGrinder*** *http://www.hammerofgod.com/download.htm*
  - for brute-force attacks against Terminal Services and RDP connections
- ***SQLRecon*** *http://www.sqlsecurity.com/DesktopDefault.aspx?tabid=26)*
  - for brute-force attacks against SQL authentication

Password Cracking

- ***dictionary attacks***
  - try each word then obvious variants in large dictionary against hash in password file
- ***rainbow table attacks***
  - precompute tables of hash values for all salts
  - a mammoth table of hash values
  - e.g. 1.4GB table cracks 99.9% of alphanumeric Windows passwords in 13.8 secs
  - not feasible if larger salt values used

Proactive Password Checking

- ***Rule enforcement plus user advice, e.g.***
  - 8+ chars, upper/lower/numeric/punctuation
  - may not suffice
- ***Password cracker***
  - time and space issues
- ***Markov Model***
  - generates guessable passwords
  - hence reject any password it might generate
- ***Bloom Filter***
  - use to build table based on dictionary using hashes
  - check desired password against this table

Password File Access Control

- ***can block offline guessing attacks by denying access to encrypted passwords***
  - make available only to privileged users
  - often using a separate shadow password file
- ***still have vulnerabilities***
  - exploit O/S bug
  - accident with permissions making it readable
  - users with same password on other systems
  - access from unprotected backup media
  - sniff passwords in unprotected network traffic

Limit validity of password

- ***Limit usage of easy passwords***
  - Set default password
  - Change default password to unique, unguessable value
- ***Limit password validity.***
  - Expiry dates for passwords forces users to change passwords regularly
  - Prevent users from reverting to old passwords, e.g. keep a list of the last ten passwords used.
- ***Limit attempts of testing password validity:***
  - Monitor unsuccessful login attempts and react by locking user account (completely or for a given time interval) to prevent or discourage further attempt
- ***Inform users***
  - display time of last login and number of failed login attempts since

Limitations impacts Usability

- ***Default passwords are "printed" in system manual***
  - Cannot be different for every system!
- ***Users are best at memorizing passwords they use regularly but not when used only occasionally***
  - Do not change passwords before weekends or holidays
- ***Limits apply to all users simultaneously → individual failures become massive failures***
  - Do not change all users passwords on the same day

Bootstrapping authentication

- ***Passwords are secrets shared between user and system***
  - "The" user is whoever knows the secret
- ***How do you bootstrap a system so that the password ends up in the right places, but nowhere else?***
  - In an enterprise, users can collect their password personally.
  - In Web applications you want to deal with remote users.

(Weak) authentication of a remote user

- ***For remote users, passwords could be sent by mail, email, or phone, or entered by the user on a web page.***
- ***"Normally" your forgotten password is sent to your email address.***
  - Ability to reading an email is a proxy for your ability to know the password → to read the email you must know a password
  - How secure is that?
- ***You have to consider who might intercept the message and who might actually pick it up.***
  - E.g., a letter containing the password for an online bank account might be stolen or an impersonator may phone in asking for another user's password.

(Stronger) Authention of a Remote User

- ***Send passwords that are valid only for a single log-in request so that the user has to change immediately to a password not known by the sender***
  - Assume attacker does not control server's email, backbone network, local network, local email
- ***Request confirmation on a different channel to activate user account,***
  - Enter password on a webpage and send confirmation by SMS.
  - Send mail by courier with personal delivery.
- ***In an organisation:***
  - Don't give password to caller but call back an authorized phone number, e.g. from an internal company address book.
  - Call back someone else, e.g. caller's manager or local security officer.
- ***More details later when we discuss application authentication***

Resetting Passwords

- ***When setting up a new user account some delay in getting the password may be tolerated.***
- ***If you have forgotten your password but are in the middle of an important task you need instant help.***
- ***The procedures for resetting a password are the same as mentioned previously, but now instant reaction is desirable.***
  - In global organisations a hot desk has to be available round the clock.
  - Proper security training has to be given to personnel at the hot desk → e.g. call back

Spoofing Attacks

- ***When the user cannot check who will receive the password, spoofing attacks are possible:***
  - Attacker starts a program that presents a fake login screen and leaves the computer.
  - Next user coming to this machine enters username and password; these are stored by the attacker.
  - Login is aborted with a (fake) error message and the spoofing program terminates.
  - Control is returned to the operating system which now prompts the user with a genuine login request.

Is this just theory?

- ***Zeus – "Man in the Browser" attack on e-banking authentication system***
- ***Bank requires***
  - User password to log in on the system
  - one time password to make a bank transfer
- ***How Zeus managed to bypass that?***
- ***Which solutions the bank devised?***

Countermeasures

- ***Mutual authentication***
  - The system has to authenticate itself to the user.
  - Easier to do if the "user" is not a human but a program working on behalf of the user
- ***Trusted path***
  - guarantees that user communicates with system (e.g. the operating system and not with a spoofing program
    - E.g. secure attention key CTRL+ALT+DEL in Windows invokes the operating system logon screen.
  - Again easier to do if the "user" is in reality a program → see network lectures
- ***Log monitoring***
  - Displaying number of failed (or successful) logins may tell the user that something he didn't intended has happened.

Key Observations

- ***A password does <u>not</u> authenticate a person.***
  - Successful authentication only implies that the user knew a particular secret.
  - There is no way of telling the difference between the legitimate user and an intruder who has obtained that user's password.
- ***There is a case of computer misuse where somebody has logged in using your username and password.***
  - Can you prove your innocence?
  - Can you prove that you have not divulged your password?
- ***You cannot log in for some reason but there is an important task to do that requires authentication***
  - Can your secretary can log in for you and do all boring tasks as if he was you?
  - If you are wounded in combat can you pass the password to the second in command so he can take your place?

Why passwords are so resilient?

- ***Lot of research to replace passwords but no successful alternative yet***
  - Pass-phrases, pass-faces (very bad for male users), pass-signs etc.
  - What is the reason?
- ***Bugs***
  - .
  - .
- ***Features***
  - .
  - .
  - .

Why passwords are so resilient?

- ***Lot of research to replace passwords but no successful alternative yet***
  - Pass-phrases, pass-faces (very bad for male users), pass-signs etc.
  - What is the reason?
- ***Bug***
  - You only need a keyboard to generate your secrete
  - Anybody who obtains your secret is "you".
  - You leave no trace if you pass your secret to somebody else.
- ***Feature***
  - You only need a keyboard to generate your secret
  - Anybody who obtains your secret is "you".
  - You leave no trace if you pass your secret to somebody else.

Something You Hold

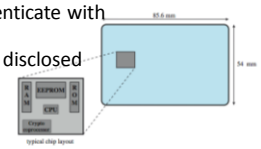
- ***The user has to present a physical token to be authenticated.***
  - In the past: keys (for self-access), seals (for access monitored by humans)
  - Today: Cards or identity tags (access to buildings), smart cards.
- ***Feature + Bug***
  - Anybody who is in possession of the token has the same rights as the legitimate owner.
  - Physical tokens can be lost or stolen without the user's cooperation
- ***To increase security, physical tokens are often used in combination with something that cannot be stolen***
  - bank cards come with a PIN or with a photo of the user.

Memory Card

- ***store but do not process data***
- ***magnetic stripe card, e.g. bank card***
- ***electronic memory card***
- ***used alone for physical access***
- ***with password/PIN for computer use***
- ***drawbacks of memory cards include:***
  - need special reader
  - loss of token issues
  - user dissatisfaction

Smartcard

- ***Smartcard has own processor, memory, I/O ports***
  - wired or wireless access by reader
  - may have crypto co-processor
  - ROM, EEPROM, RAM memory
- ***Can store secrets***
  - executes protocol to authenticate with reader/computer
  - secrets are "used" but not disclosed
  - secrets are tamperproof
- ***Alternative: USB dongles***

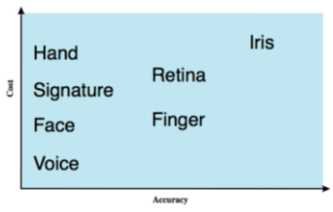


Who You Are

- ***Biometric schemes use unique physical characteristics (traits, features) of a person***
  - face,
  - finger prints,
  - iris patterns,
  - hand geometry
- ***Biometrics may seem to offer the most secure solution for authenticating a person***
  - Very good for specialized/limited access → e.g. access to ACC may require biometric authentication
- ***Little experience from large scale field trials on the performance of biometrics***
  - So far only large scale is biometric on mobile devices, but not know if most people actually turned that on

Biometric Authentication

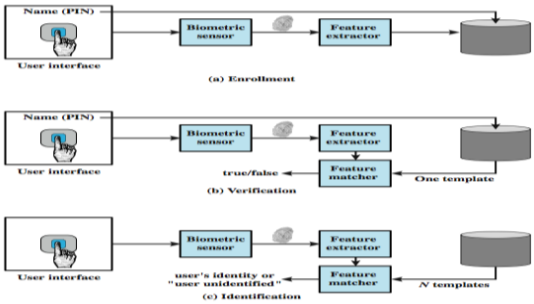
- ***authenticate user based on one of their physical characteristics***



Biometrics

- ***Use physical traits unique for each individual:***
  - Fingerprints
  - Iris patterns
- ***Biometric authentication (1:1 comparison, also called verification):***
  - Register biometric sample (fingerprint).
  - For authentication, compare new biometric sample with the user's registered reference value.
- ***Biometric identification (1:n comparison):***
  - Take biometric sample and compare against a database of samples to find out who the user is.

Operation of a Biometric System

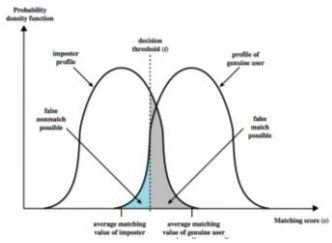


Enrolment and Authentication

- ***Enrolment:***
  - A reference template of the user's fingerprint is acquired at a fingerprint reader.
  - Templates are stored in a secure database.
- ***Failure-to-enrol (FTR):***
  - not every person has usable fingerprints.
  - For higher accuracy, several templates may be recorded, possibly for more than one finger.
- ***Authentication***
  - When the user logs on, a new reading of the fingerprint is taken and compared against the reference template.

Biometric Accuracy

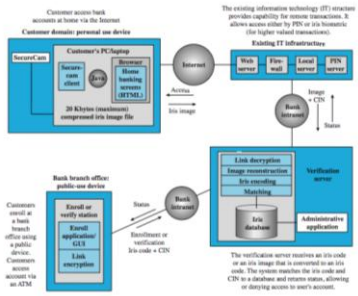
- ***never get identical templates***
- ***problems of false match / false non-match***



Biometrics

- ***Current and registered sample will never match perfectly → a user will be accepted if the match scores above a given threshold***
- ***False acceptance rate (FAR):***
  - wrong user accepted; problem in sensitive areas
- ***False rejection rate (FRR):***
  - user wrongly rejected; problem in commercial areas
- ***Equal error rate (EER):***
  - threshold set so that FAR= FRR
  - Best EER for fingerprint systems 1-2%; iris recognition has better performance.
- ***In practice threshold depends on application***

Practical Application



Biometrics

- ***Biometric traits are unique identifiers but no secrets!***
  - You leave your fingerprints in many places and fingers can be "forged" quite effectively.
  - Recall the US Social Security Number mistake!
- ***Local check (e.g. border control in Frankfurt):***
  - one can take measures to ensure a proper sample is taken.
- ***Remote check (Internet):***
  - if you cannot control how samples are taken, biometrics identify rather than authenticate individuals.

Biometrics – change control

- ***Identity theft:***
  - How to react if someone else misuses your fingerprint?
- ***If there is fraud on your credit card,***
  - you can be re-issued with a new card and PIN
  - If you have more than one card, the other cards are not affected.
- ***If you have burnt your finger, is there a back-up system for getting access?***
- ***What happens with a person that does not have the required biometric trait?***

What You Do

- ***People perform mechanical tasks in a way that is both repeatable and specific to the individual.***
  - Handwriting experts look at the dynamics of written documents to detect forgeries.
  - The way you raise the phone when you answer a call
- ***Example***
  - Let users sign on a special pad that measures attributes like writing speed and writing pressure.
  - On a keyboard, typing speed and key strokes intervals can be used for user authentication.
- ***Remote authentication needs trusted path from device capturing dynamic behavior to server.***

Where You Are

- ***Some operating systems grant access only if you log on from a certain terminal.***
  - A system manager may only log on from an operator console but not from an arbitrary user terminal.
  - Users may be only allowed to log on from a workstation in their office.
- ***Decisions of this kind will be even more frequent in mobile and distributed computing.***
- ***Global Positioning System (GPS) might be used to established the precise geographical location of a user during authentication BUT***
  - GPS is military and operated by the US
  - Galileo is an alternative program by the EU but still long way to go

Remote User Authentication

- ***authentication over network more complex***
  - problems of eavesdropping, replay
- ***generally use challenge-response***
  - user sends identity
  - host responds with random number N
  - user computes some function with N that only user can generate and sends back
  - host compares value from user with own computed value (or other similar function), if match user authenticated
- ***protects against a number of attacks***
- ***More of this in the application security part***

Challenge Response - II

- ***The Simplest protocol***
  - A & B agrees on some parameters off line
  - A → B: I'm A
  - B → A: Nonce = random number
  - B → A: f(B,Nonce) = function that only B can make but that A can check
  - A : ok
- ***Example instantiation***
  - A, B share secret S
  - …
  - B → A: Hash(S.B.A.Hash(S.Nonce)) provided Hash is a function that is easy to compute but hard to invert.
    - Why this is better than sending H(S.B.A.N)?
    - Why this better than sending H(S.B.N)? Why H(S.A.N) is worst of all?
    - When Hash(Hash(S).B.A.Hash(Hash(S).N)) would be desirable?
- ***Another example: send secure code via phone***
  - How really secure is that?

Beware: Security mechanisms fail

- ***Two equally important worries:***
  - failures that wrongly permit an action
  - failures that wrongly deny access
- ***Forgotten passwords, lost token, false biometric rejection, too frequent re-authentication, etc. etc.***
  - If not adequately addressed → system not available to legitimate users
  - If you believe your technology is perfect (or forget about this issue) → your system <u>will</u> fail

UNIVERSITY OF TRENTO - Italy

eit Digital MASTER SCHOOL

Reading Material

- ***Chapter 2. Stallings & Brown. Computer Security Principles and Practice.***
- ***Papers on password studies by***
  - Angela Sasse at UCL on what doesn't work
  - Frank Stajano at Cambridge on large studies and possible alternatives
- ***More sophisticated methods based on credentials***
  - See OpenAuth white paper