COMS3000/7003

Week 3

Authentication Protocols, Biometrics

David Ross

Assignment

The University of Queensland
School of Information Technology and Electrical Engineering

COMS3000/7003
Semester 2, 2017

Report (Weighting 20%)

This Assignment is due **4:00 pm Friday, 22/9/2017**

Auric Enterprises Threat and Vulnerability Analysis

[COMS3000: 20%, 20 marks (total)]
[COMS7003: 15%, 20 marks * (3/4), this section + 5 marks additional task]

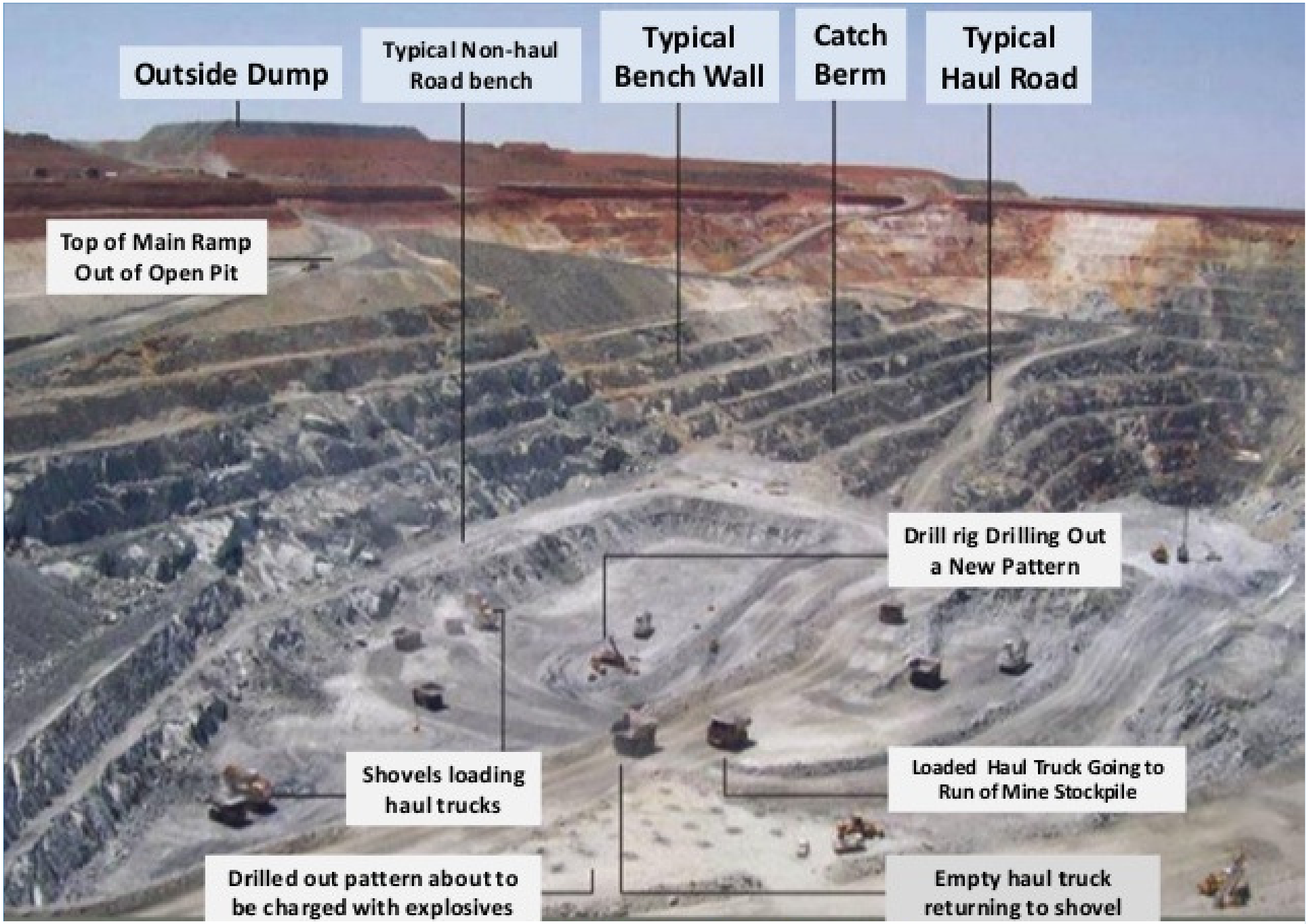


Image by Prof Hassan Z Harraz <Hassan.harraz@science.tanta.edu.eg>

Late Assessments (FAQ)

- **See course profile**: As given in the first lecture:
- "The submission of progressive assessment material on the due date as set out in this Electronic Course Profile is the sole responsibility of the student."
- "Unless advised in the Course Profile, assessment items received after the due date will receive a zero mark unless you have been approved to submit the assessment item after the due date.
- However, if there are medical or exceptional circumstances that will affect your ability to complete an assessment by the due date, then you can apply for an extension via the following methods:"

“5.3 Late Submission” (ECP)

- ***Mid-Semester Examinations (includes Oral Presentation, Written Examination or Laboratory Practical held during the teaching weeks of semester):***
- All applications for deferred mid-semester examinations must be submitted online via mySI-net > myRequests.
- Hard copy application forms or requests received via email will not be considered.

“5.3 Late Submission” (ECP)

- ***“Other Assignments:”***
- “You can find further information and the relevant forms online. The ***Application for Extension of Assessment Due Date*** form and supporting documentation (e.g. medical certificate) can be submitted by email to *enquiries@itee.uq.edu.au* or in person to the School office (General Purpose South [78], level 4 Coursework Studies Office).”

“5.3 Late Submission” (ECP)

- “An extension application granted on medical grounds will be approved for the number of calendar days the medical certificate indicates you were unfit for study. Students who are ill for more than 14 days should consider applying for withdrawal without academic penalty.
- Requests must be made at least 48 hours prior to the submission deadline, unless the medical or other circumstances are such that you could not reasonably be expected to have applied by then.”

“5.3 Late Submission” (ECP)

- “Requests for extensions which are received on or after the due date may not be able to be considered.
- Extensions may not be possible for some pieces of assessment (such as assignments for which solutions are posted immediately after the submission deadline or in the case of group work). Where an extension cannot be granted for such reasons, the Course Coordinator may propose equivalent assessment.
- The School will issue a notification of the outcome to your student email account.”

“5.3 Late Submission” (ECP)

- Please note: While a scanned copy or clear photographic image of the supporting documentation is acceptable, you must retain the original documentation for a minimum period of six (6) months to provide as verification should you be requested to do so. Failure to produce the original documentation for verification may result in the approval of your extension being rescinded.
- For assignments that require both a hard copy as well as an electronic submission, the assignment is considered as submitted only when BOTH the hard copy AND the electronic version have been submitted. If the two versions are submitted at a different time or day, the later submission time and day will be considered.”

Today's Lecture

- Quick Recap
  - Access Control, Authentication
  - Cryptographic one-way hash functions
    - In the following, I will often refer to them simply as 'cryptographic hash function' or 'hash function'
- Authentication Protocols
- Multi-factor Authentication
- Introduction to Biometrics

Recap

Access Control

- Three steps of Identity-based Access Control
  - Identification
    - “Tell me who you are”
  - Authentication
    - “Prove that you are who you claim to be”
  - Authorisation
    - Determine what access is granted for this principal (person/entity) for this resource (asset)

Authentication

- Authentication
  - “The process of verifying a claimed identity.”
- Three basic methods of Authentication
  - With something you **know**
  - With something you **have**
  - With something you **are (or do)**

Back to Passwords
Unix Password Process

- Unix (and Windows) doesn't store the user password *p*.
  - It stores a one-way hash *h(p)* of the password.
    - Some people call this 'encrypted passwords', even though this is technically not quite correct.
- From knowledge of secure hash functions, we know:
  - It is practically 'impossible' to calculate *p* given *h(p)*.
- How does a login procedure work?
- When a user offers a password p_o at login, the OS calculates the hash $h(p_o)$, and checks whether $h(p_o) = h(p)$
- If so, it is almost certain that $p_o = p$, and the user is authenticated.
  - Remember: Collisions are extremely rare
- An attacker learning *h(p)* cannot find *p*, due to the one-way nature of h

UNIX Password File

- Traditionally, Unix stored 'encrypted' password in the file */etc/passwd*, which is world-readable
- Example:

```
root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
uucp:OORoMN9FyZfNE:4:4::/var/spool/uu:/usr/lib/uucp/uucico
rachel:eH5/.mj7NB3dx:181:100:Rachel Cohen:/u/rachel:/bin/ksh
arlin:f8fk3j1OIf34.:182:100:Arlin Steinberg:/u/arlin:/bin/csh
```

- Is the problem of password storage security solved?
- Not quite. An attacker can launch an ***off-line*** password cracking attack
  - Try different passwords, hash them and compare result with any of the entries in the password file. Continue until there is a match.
- → 'Encrypted' or hashed passwords need to be hidden
  - "Shadow passwords"
  - Store password hashes in a separate file with restricted access, e.g. */etc/shadow*
  - *For practical reasons, access to /etc/passwd file cannot be restricted. A lot of utilities rely on other information stored there, e.g. user IDs, Groups IDs, …*

Password Cracking

- What type of attack is possible if an attacker has access to the password file (hashed passwords)?
- Dictionary cracking attack
  - For ‘common’ words p, calculate h(p) and see if it matches any entry in the password file
- Brute-force attack
  - Try all possible passwords
  - For every possible word *p* up to a certain length, calculate h(*p*) and see if there is a match in the password file
- Hybrid
  - Use dictionary words combined with numbers etc.
- These are **off-line** attacks
  - Attacker has access to password hashes, and plenty of time

Password Cracking Tools

- There are many …
- For example, there are more than 30 password cracking tools in Kali Linux
  - Kali Linux – Penetration Testing Distribution
  - https://www.kali.org/

Brute Force Attack

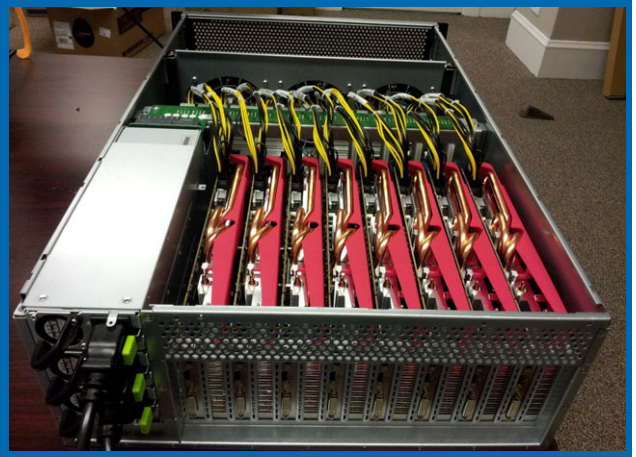
- Brute force attack: Trying every possible password
- We assume
  - Password length of 8 characters
  - We assume that passwords are chosen randomly
  - We assume an attacker can do 10,000,000 guesses per second

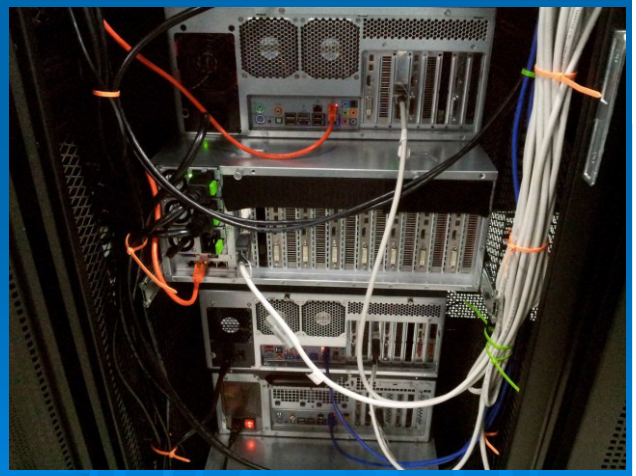
- Case 1:
  - Only lowercase alphanumerical letters, a,b,c, …z (26)
  - Number of passwords?
    - $26^8 \approx 2*10^{11}$ → An attack takes on average 2.75 hours
    - (We assume that we have only one password hash. On average, we find the result after trying half of the possible passwords.)

- Case 2:
  - All printable characters: 95
  - Number of passwords?
    - $95^8 \approx 6.6*10^{15}$
    - → An attack takes on average about 1 year

Password Cracking Speed

- Our assumption of password cracking speed on the previous slide was rather conservative
- 25 GPU Cluster, built in 2012, achieves a speed of
  - 350 billion-guesses-per-second
  - Can ‘brute force’ 8 character passwords (case 2) in less than 6 hours (instead of 1 year)





- https://arstechnica.com/security/2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours/

- Long, random (high entropy) passwords are still secure
  - We will talk about this in another lecture

Brute Force Attack – Time-memory trade off

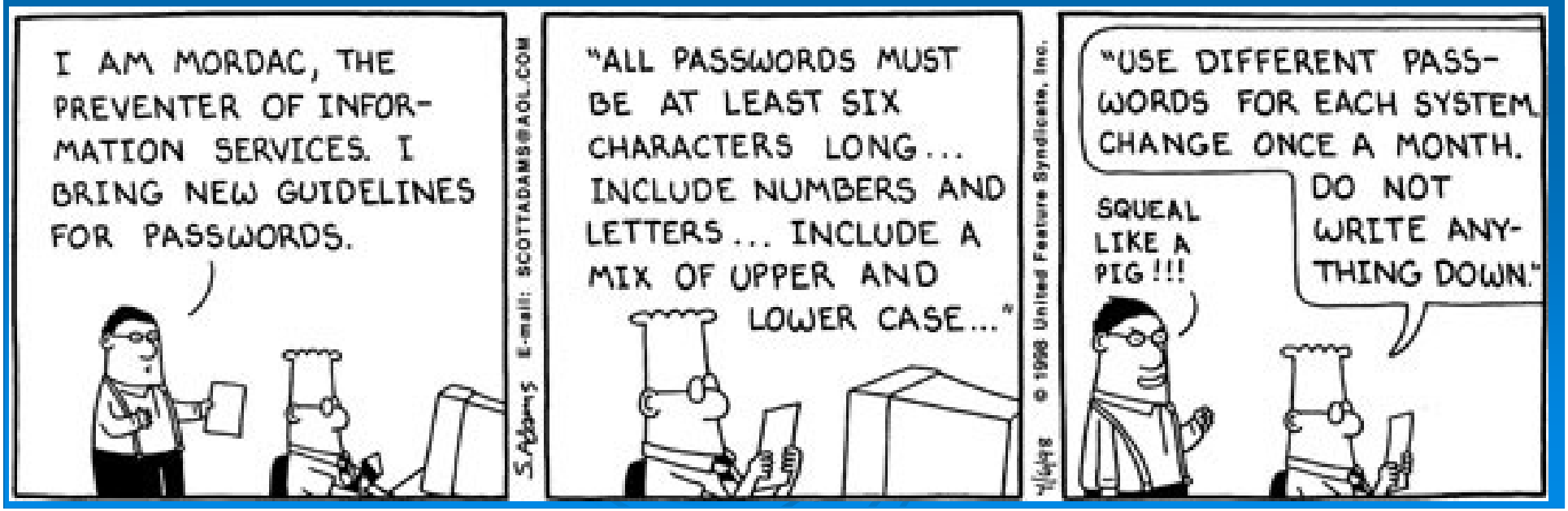
- To increase the password cracking speed even further, could an attacker pre-compute and store all possible password hashes, e.g. for all passwords of up to 8 characters?
- Required memory
  - $95^8 \approx 6.6*10^{15}$ hashes ≈ 105,600 TB
  - Not practical
- But, can pre-compute hashes of a subset of passwords, and thereby save some time
  - 'Time-memory trade off'
  - 'Rainbow tables'
    - https://en.wikipedia.org/wiki/Rainbow_table
    - http://project-rainbowcrack.com/
  - There are many (large) rainbow tables available for download

Password Selection

- Even seemingly random passwords can surprisingly easily be cracked
  - https://www.schneier.com/blog/archives/2013/06/a_really_good_a.html
  - Use of large dictionaries, with clever rules for ‘tweaking’ and combining
  - → Brute force attack is rarely needed!

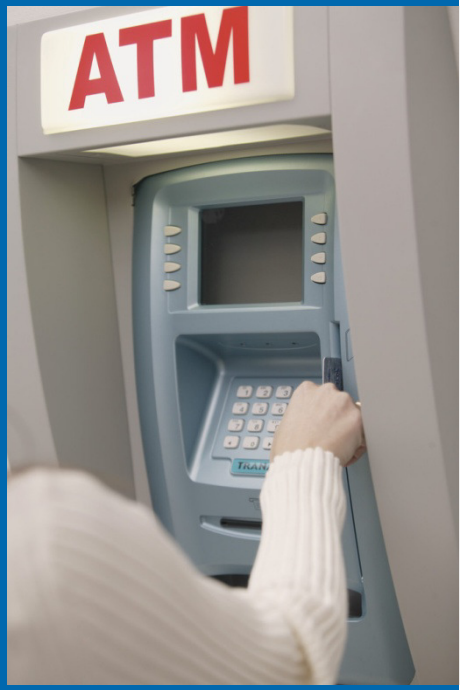
Password Selection Dilemma
Usability vs. Security

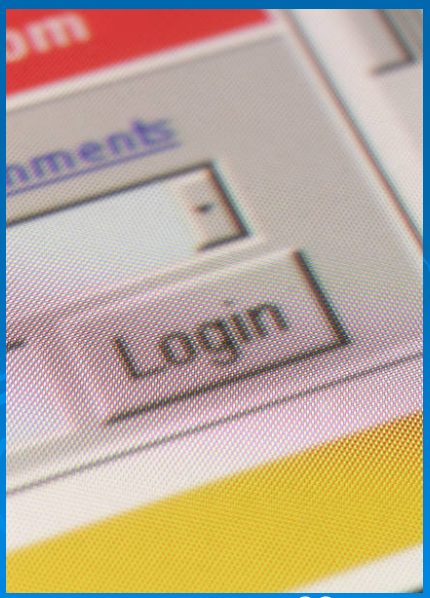
- Conflict between usability and security
  - Trade-off
- *Typical Password selection Rules:*
  - *"The password must be impossible to remember and never written down"*, or something along those lines.
- Important is the "Entropy" or "Randomness" of password
  - More on this in another lecture



Online Password Attack

- We have talked a lot about offline password cracking attacks
- What is a online password attack?
- For example
  - Someone trying a PIN with a stolen ATM card
  - Someone, who knows your login, trying to login to your web bank or computer account by guessing and trying different passwords
- What is the key difference to the offline attack?
  - Attacker talks to a live system
  - Much harder
  - Time and/or number of tries is limited





Security of Passwords

- Is a 4 digit PIN code for an ATM secure?
- What about brute force attack?
  - Only 10'000 different PINs possible
- The security of passwords depends on the user interface.
- ATM
  - Hard to automate. Tries are manual → slow
  - After 3 unsuccessful tries, the machine 'eats' the card
- In this context, a 4 digit PIN is reasonably secure!

Password Managers

- Help users deal with managing large number of password protected accounts
- Typically, users need to remember only one ‘master password’, which then unlocks access to ‘site specific’ passwords
  - Either encrypt site-specific password chosen by user, or automatically generate site-specific passwords, e.g. from URL + Master password
  - Online, offline
  - Many commercial and free versions available
    - LastPass, Dashlane, RoboForm, 1Password, KeePass …
    - http://www.asecurelife.com/dashlane-vs-lastpass-vs-1password-vs-roboform-vs-keepass/
    - https://www.lifehacker.com.au/2015/02/lifehacker-faceoff-the-best-password-managers-compared/
    - http://forums.whirlpool.net.au/archive/2481014 …

Any questions so far?

AUTHENTICATION PROTOCOLS

Password-based Authentication Protocols

- Authentication over the network
  - e.g. remote login, authentication to web server
- How can we do this in the most basic way?
  - Alice sends her password p_A to the server (Bob)
    - Illustration
      - https://outbox.eait.uq.edu.au/uqmportm/coms3000/SimplePasswordReplay.html
    - Eve (Dogbert) can eavesdrop and learn password
- Example protocol:
  - PAP (Password Authentication Protocol, RFC 1334)
- Alternative?
  - Alice sends a hash of her password, $h(p_A)$
    - Eve cannot find p_A, but?
    - Attacker does not need password, but can simply resend hashed password → Replay attack
- So what is a better approach?

Challenge-Response Protocol

- Server gives client a "challenge", *c*
  - *Often called a 'nonce' (number used once)*
- Client calculates a response, which is a cryptographic one-way hash of *c* and password *p*
  - *r = h(c || p)* or *r = h(c XOR p)*
  - *'||' means concatenation*
  - Client sends *r* back to server
- Server can check calculation
- Eavesdropper can see *c* and *r*,
  - but cannot derive *p*
  - *Replay attack is not possible*
- *Animation:*
  - https://outbox.eait.uq.edu.au/uqmportm/coms3000/SimpleChallengeResponse.html
- *Example Protocol:*
  - *Challenge-Handshake Authentication Protocol (CHAP), RFC1994*

Choice of Challenge

- Consider a challenge response protocol where the server selects the challenge deterministically and predictably
- For example:
  - c1 = 1
  - c2 = 2
  - c3 = 3
  - …
- Response
  - r = h(ci || p)
- Problem?
- Doesn't the challenge have to be random and unpredictable?
- The main requirement is that *c* is not being reused (i.e. a nonce)
- The ability to predict *c* does not give an attacker an advantage
  - Only if c is reused can attacker launch replay attack

Practical Example
HTTP Authentication

- HTTP (Hypertext Transfer Protocol)
  - Simple request/response protocol between web clients and servers
- HTTP provides a simple access control and authentication mechanism
  - e.g. limit access to certain web pages to specific users
- Defined in RFC 2617
  - RFCs (Request for Comments) Standard documents for Internet protocols issued by the IETF (Internet Engineering Task Force)
    - https://www.ietf.org/rfc/rfc2617.txt
- Two types of authentication
  - "Basic" and "Digest"

HTTP Basic Authentication

- Browser sends a ‘GET’ request to a web server:
  - “Please give me this file”
- Server says:
  - “Not authorised for realm XYZ”
    - Realm = protected files or directories
  - Require “Basic Authentication”
- Browser asks user for username and password for realm XYZ
  - → pop-up window
- Browser sends new GET request:
  - “Please give me this file. Authentication details are:
    - Basic Authentication, username:password”
  - Server checks details, and responds.
- Problem?
  - Password sent in cleartext, can be eavesdropped
  - Basic HTTP Authentication is NOT secure!

HTTP Digest Authentication

- Challenge-response mechanism
  - Browser sends HTTP GET request
  - Server replies with:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm="Demo Content"
nonce="1011565422321"
```

  - Server reply includes the challenge (nonce)
  - Browser prompts user for username and password
  - Browser computes response *r* as hash (or "digest") of:
    - (username, password, realm, nonce, URL)
  - This hash is sent back to the server, which verifies it
- Secure?

HTTP Digest Authentication

- Digest Authentication solves the most severe security problem of Basic Authentication
  - Password is never sent in cleartext (cleartext = unencrypted)
    - and cannot be computed from response
  - Replay attack not possible
- Limitations:
  - Still password based
    - 'Plaintext' password stored at server
  - Uses MD5 hash, which is considered broken
  - Vulnerable to Man-in-the-middle attacks
    - More on this in the tutorial
- OK for low security applications
- What if we need a higher level of security?
  - Certificate-based authentication, TLS, HTTPS
  - Discussed later when we talk about public-key cryptography

SSH Authentication

- Secure Shell (SSH) Authentication Protocol
  - https://www.ietf.org/rfc/rfc4252.txt
  - This SSH Version 1 is vulnerable to man-in-the-middle attacks.
- Supports several methods
  - Password
  - Public Key
  - …
- Password Authentication
  - SSH establishes an encrypted channel between client and server
  - Then, the password (not hash) is sent across this secure channel
  - Limitation
    - Vulnerable to Man-in-the-middle attack
    - Problem: lack of authentication when establishing secure channel
- Public Key Authentication
  - Not vulnerable to Man-in-the-middle attack
  - Will discuss Public key crypto later

SSH-2 Authentication

- SSH-1 is vulnerable to man-in-the-middle attacks
- SSH-2 uses stronger key exchange:
  - Diffie-Hellman Group Exchange for the SSH Transport Layer Protocol
  - RSA Key Exchange for the SSH Transport Layer Protocol
  - Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the SSH Protocol
  - AES Galois Counter Mode for the SSH Transport Layer Protocol
  - Elliptic Curve Algorithm Integration in the SSH Transport Layer
  - SHA-2 Data Integrity Verification for the SSH Transport Layer Protocol
- Supports several authentication methods
  - Password
  - Public Key
  - keyboard-interactive (RFC 4256): e.g. S/Key or SecurID
  - GSSAPI

Lamport's Hashed Password Scheme

- Application of cryptographic one-way hash function for secure Authentication
- Idea: Create a sequence or chain of ***one-time*** passwords using a **one-way** hash function
  - Leslie Lamport, "Password Authentication with Insecure Communication", Communications of the ACM, November 1981 (short paper, only 2 pages)
  - http://lamport.azurewebsites.net/pubs/password.pdf
- Used in "S/Key" scheme, by Bellcore 1994
  - Defined in RFC1760 and RFC2289
  - Available for most Operating Systems, but less often used these days
- How do one time passwords (OTPs) work?
  - Each password is only used once, no reply attack possible
  - https://en.wikipedia.org/wiki/One-time_password
- How could one-way hash chains be used to implement this?

Lamport's Hashed Password Scheme

- Client selects initial secret password p_0 (seed value)
- Compute sequence of hash values
  - $p_1 = h(p_0)$
  - $p_2 = h(p_1) = h(h(p_0))$
  - $p_3 = h(p_2) = h(h(h(p_0)))$
  - ..
  - $p_n = h(p_{n-1})$
  - $p_{n+1} = h(p_n)$
- Store p_{n+1} at server with counter *n*
- When user logs in, server sends *n* as a challenge
- User sends p_n, server calculates $p_{n+1} = h(p_n)$ and compares it with stored value p_{n+1}
  - If correct, user is authenticated
- Server decrements *n* by one
- Next time, user needs to send password higher up in the list
- Eavesdropper observing p_n cannot compute p_{n-1}, required for next authentication
  - One-way nature of hash function
- No need to store password at the server!

Lamport's Hashed Password Scheme

- Animation:
  - https://outbox.eait.uq.edu.au/uqmportm/coms3000/SKeyPassword.html
- Possible Attack:
  - Attacker pretends to be server and gets valid password to be used with real server
    - Man-in-the-Middle Attack (see tutorial)
- Practical Problem:
  - Need to recalculate and re-distribute hash once we reach index 0

Authentication Protocols

- More on Authentication Protocols in a few weeks, when we have covered cryptography, and public-key cryptography in particular.

Back To General Authentication Mechanisms

Multi-factor Authentication

- Authentication: based on something you are, know or have
- If a higher level of security is required, two or three mechanisms are combined. Most commonly, two factors are used:
  - Two-factor authentication
- Something you know **and** something you are
  - e.g. Password and fingerprint
- Something you know **and** something you have
  - e.g. Password and physical key
- Something you are **and** something you have
  - e.g. Fingerprint and a physical key
- Do you know a practical example of two-factor authentication, something everybody uses on a regular basis?
  - ATM

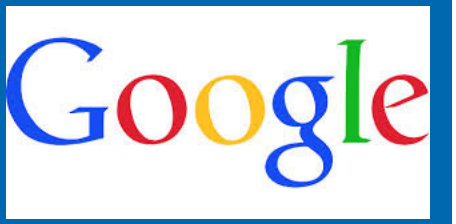
    You need the card and you need to know the PIN

Two-factor Authentication - Example

- Two factors:
  - Normal password
  - SecureID token
- SecureID runs pseudo random number generator
  - e.g. creates a new number every few minutes
  - A secure algorithm guarantees that even after observing a long sequence of numbers, numbers are still unpredictable for an attacker
- Same algorithm runs on server
  - Loosely time synchronised

Other Examples of Two-factor Authentication

- Google '2-step verification'
  - http://www.google.com.au/landing/2step/



- PayPal 'Security Key'
  - https://www.paypal.com/au/securitykey

- Twitter 'login verification'
  - https://blog.twitter.com/2013/getting-started-login-verification
  - https://www.cnet.com/au/how-to/how-to-enable-twitters-two-factor-authentication/

Authentication, Identification Reading

- Leslie Lamport, “Password Authentication with Insecure Communication”, Communications of the ACM, November 1981 http://lamport.azurewebsites.net/pubs/password.pdf
- The S/KEY One-Time Password System, RFC1760 https://www.ietf.org/rfc/rfc1760.txt
- HTTP Authentication, RFC2617 https://www.ietf.org/rfc/rfc2617.txt

Any questions so far?

BIOMETRICS

Biometrics in the News

- The Australian, 20 August 2015
  - http://www.theaustralian.com.au/life/personal-technology/windows-hello-can-identical-twins-fool-microsoft-and-intel/story-e6frgazf-1227490164701

Windows 10, released last month by Microsoft, replaces the hackable password system with biometric recognition. You log in using your fingerprints, and with eye and face recognition.

The new feature is called Windows Hello. If you have an iPhone or recent Samsung smartphone, you will know how convenient fingerprint recognition is, and it has proved consistent and reliable.

But a large number of notebooks coming on to the market with Windows 10 offer face recognition as an alternative to passwords for accessing your account.

The face recognition process involves a RealSense camera made by Intel, which sits embedded above the display. Three cameras — featuring an infra-red lens, a regular lens and a 3-D lens — use photographic analysis, heat detection and depth detection to decide who is at your computer display.

Personally I found face recognition worked a treat. The Lenovo Thinkpad Yoga 14 we used quickly identified who I was among several account holders, and in a flash logged me in.

Biometrics

- What are Biometrics?
  - *biometric* comes from the Greek words *bios* (life) and *metrikos* (measure).
- Authentication based on something you are
  - Characteristics of the human body, behaviour
- "Biometrics is the set of automated methods to recognize a person based on physiological or behavioural characteristics"

  Biometric Consortium
- Increasingly relevant
  - e.g. many countries have Biometric Passports
    - e.g. facial, fingerprint, iris
  - Smartphones have biometrics built in
  - Windows 10
  - …

Class Exercise

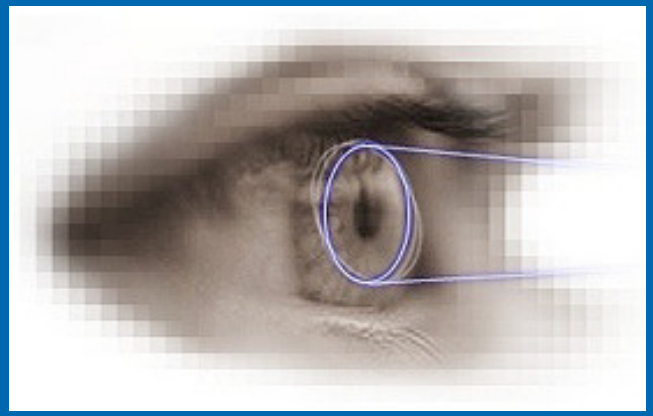
- In groups of 2 – 4, discuss different Biometric authentication methods.
- Write down your results.

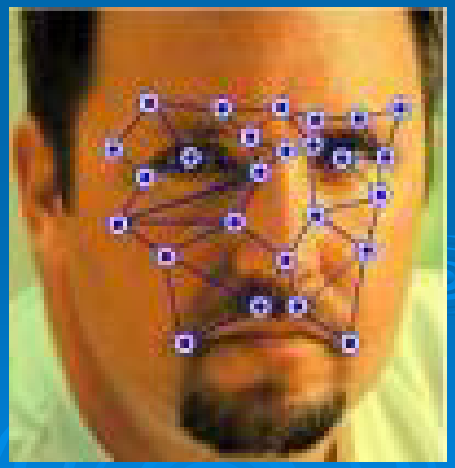
- Example of biometrics.
- What is a good biometric?
  - Are ‘shoe size’ or ‘hair colour’ good biometrics?
    - Why or why not?
- What are the important criteria for comparing different Biometrics?

“Joe Smith”

Biometrics

- Fingerprints
- Voice
- Iris
- Retina
- Hand-geometry
- Gait
- Signature
- Face
- DNA
- Odor
- …





Criteria for a good Biometric

- Universality
  - Each person should have the characteristic
    - e.g. bald people don't have hair colour
- Distinctiveness
  - Any two persons should be sufficiently different in terms of this characteristic
    - e.g. shoe size is not very distinctive
- Permanence
  - The characteristic should be sufficiently invariant over a period of time
    - e.g. Hair colour might change frequently
- Performance
  - Recognition accuracy (and speed)
- Acceptability
  - Extent to which people are willing to accept Biometric in their daily live
    - e.g. Retinal scan might be considered intrusive (can reveal information about health)
    - e.g. hand geometry reader might be considered unhygienic
- Circumvention
  - How easily the system can be tricked
    - e.g. "Gummy Fingers", i.e. artificial copy of real finger
      - Mythbusters episode: https://www.youtube.com/watch?v=MAfAVGES-Yc
      - https://www.businessinsider.com.au/biometric-fingerprint-password-hacking-2015-1