Information Security CS 526

Topic 6: User Authentication

Reading for this Lecture

Wikipedia

- Password
- Password strength
- Salt_(cryptography)
- Password cracking
- Trusted path
- One time password

Important Takeaway Message

Thinking about security is to consider and weigh in different trade-offs

Understanding and proper usages of some basic terminologies are important

Three A's of Information Security

Authentication

vs.

Access Control

vs.

Audit

Authentication, Authorization, and Audit

- **Authentication**
  - It is the process of determining whether somebody is who he/she is claiming to be
- **Access control**
  - It is the process of determining whether an action is allowed with respect to some well-defined rules or **policies**
- **Audit**
  - Record everything to identify attackers after the fact

Authentication and Access Control (From Wikipedia)

- **Authentication** is the act of establishing or confirming something (or someone) as authentic, that is, that ***claims made by or about the subject are true***. This might involve ***confirming the identity of a person***, tracing the origins of an artifact, ensuring that a product is what its packaging and labeling claims to be, or ***assuring that a computer program is a trusted one***

- **Access control** is a system which enables an authority to ***control access*** to areas and ***resources*** in a given physical facility or ***computer-based information system***

Why Audit?

- Do not have enough information during decision making time to make a judgment whether an access request is valid

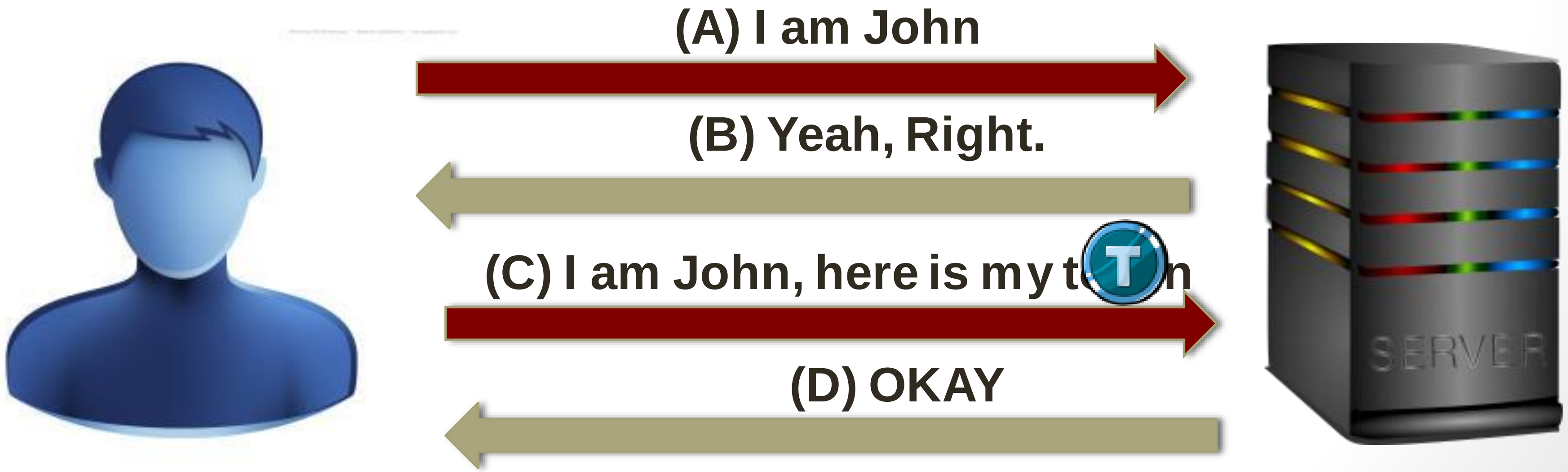
- It is difficult to weigh in all possible conditions of a valid access request

- Specially relevant when legitimacy of access request depends on contextual information

Our concentration today is *user authentication*

Scenarios Requiring User Authentication

- Logging into a local computer
- Logging into a remote computer
- Logging into a network
- Accessing websites



(A) I am John

(B) Yeah, Right.

(C) I am John, here is my token

(D) OKAY

Authentication Token

- Based on something the user know
  - **Example**: Passphrase, password
- Based on something the user possesses
  - **Example**: Smart card or token
- Based on something the user is
  - **Example**: Biometric

Proposals of Authentication Token

- Cryptography-based
- Others
  - ***Passwords***
  - Biometrics
  - Graphical passwords
  - 2-factor authentication
  - Out of band authentication

Cryptography-based Designs

- **One-time passwords**
  - Each password is used only once
  - Defend against adversary who can eavesdrop and later impersonate
- **Challenge-response**
  - Send a response related to the password and a challenge
- **Zero-knowledge proof of knowledge**
  - Prove knowledge of a value without revealing it (***Out of scope)***

One-Time Passwords (OTP)

- Two parties share a list of one-time passwords

- Time synchronized OTP
  - Example: $\mathbf{MAC_K(t)}$ where $\mathbf{t}$ is the current time

- Using a hash chain (Proposed by Lamport)
  - $\mathbf{H(s), H(H(s)), \ldots, H^{1000}(s)}$
  - Use these hash values in reverse order

Lamport's One-Time Password

- Setting: A wants to authenticate itself to B
- Initialization:
  - A selects an arbitrary value S, a hash function H(), and integer value t
  - A computes $W_0 = H^t(S)$ and sends W_0, and H() to B
  - B stores W_0
- Protocol: To authenticate to B at time i where $1 <= i <= t$
  - A sends to B: $A, i, w_i = H^{t-i}(S)$
  - B checks: $i = i_A, H(w_i) = w_{i-1}$
  - If both holds, $i_A = i_A + 1$

Challenge-Response Protocols

- **Goal**: one entity authenticates to other entity proving the knowledge of a secret, ‘challenge’
- ***How to design this using the crypto tool we have learned?***
- **Approach**: Use time-variant parameters to prevent replay, interleaving attacks, provide uniqueness and timeliness
  - Example: nonce (used only once), timestamps

Challenge-Response Protocols

- Unilateral authentication (timestamp-based)
  - A to B: $MAC_K(t_A, B)$
- Unilateral authentication (nonce-based)
  - B to A: r_B
  - A to B: $MAC_K(r_B, B)$
- Mutual authentication (nonce-based)
  - B to A: r_B
  - A to B: r_A, $MAC_K(r_A, r_B, B)$
  - B to A: $MAC_K(r_B, r_A)$

Public-key Cryptography

Cleverly use Digital Signature to authenticate to a party.

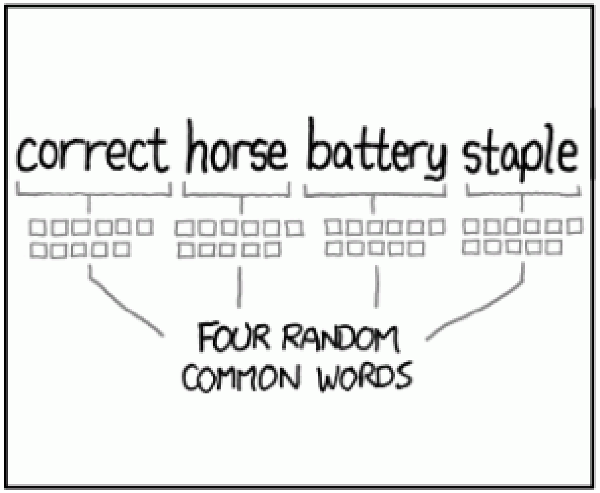
(*This will be covered later*)

Passwords

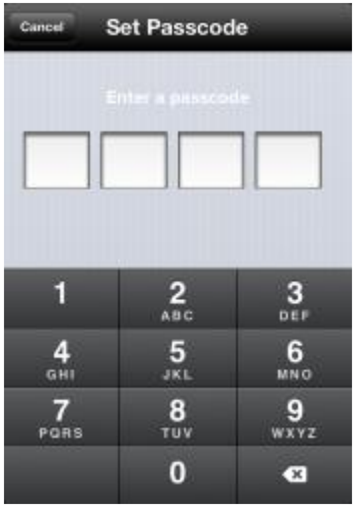
- Oldest and most common form of authentication token due to its ease of deployment
- 1961 Compatible Time-Sharing System at MIT was most likely the first deployment of passwords
- Password was deployed in traditional computer systems like MULTICS and Unix in the 1970

Variations of Passwords

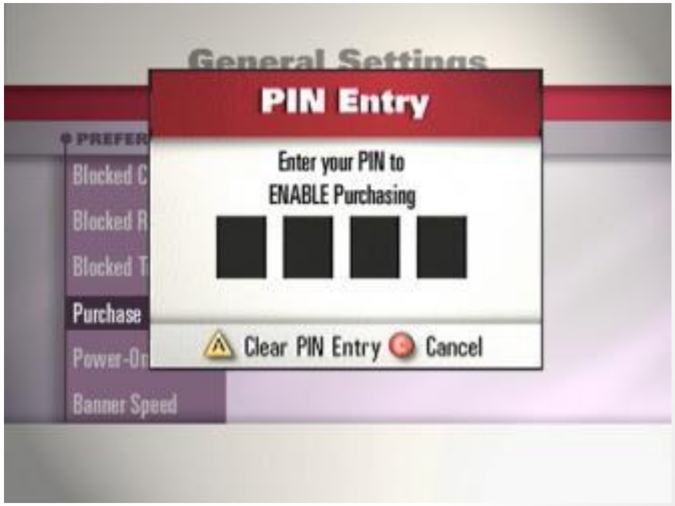
- Passphrase
  - A sequence of words or other text used for similar purpose as password



- Passcode



- Personal Identification Number (PIN)



Enter your PIN to ENABLE Purchasing

Attractive Properties of Password

- Easily deployable
  - No need for additional hardware
- Customizable
  - Choose your own password
- Convenient to replace
- Ease of use

Problems with Passwords

- For security, it is desirable for passwords to be unpredictable
- However, it is difficult to remember highly random things
- Recent survey showed, an individual on average has 106 online accounts
- It is desired for individuals to not have the same password for all accounts

Problems with Passwords

There is an inherent tension between security and usability of passwords

I changed all my passwords to "incorrect".
So whenever I forget, it will tell me "Your password is incorrect."
WeirdNutDaily.com

Usability Metrics

- Sentiment
  - Creation difficulty, recall difficulty
- Time
  - Password creation and recall
- Memorability
  - Recall attempts, password writedown

Human Memory

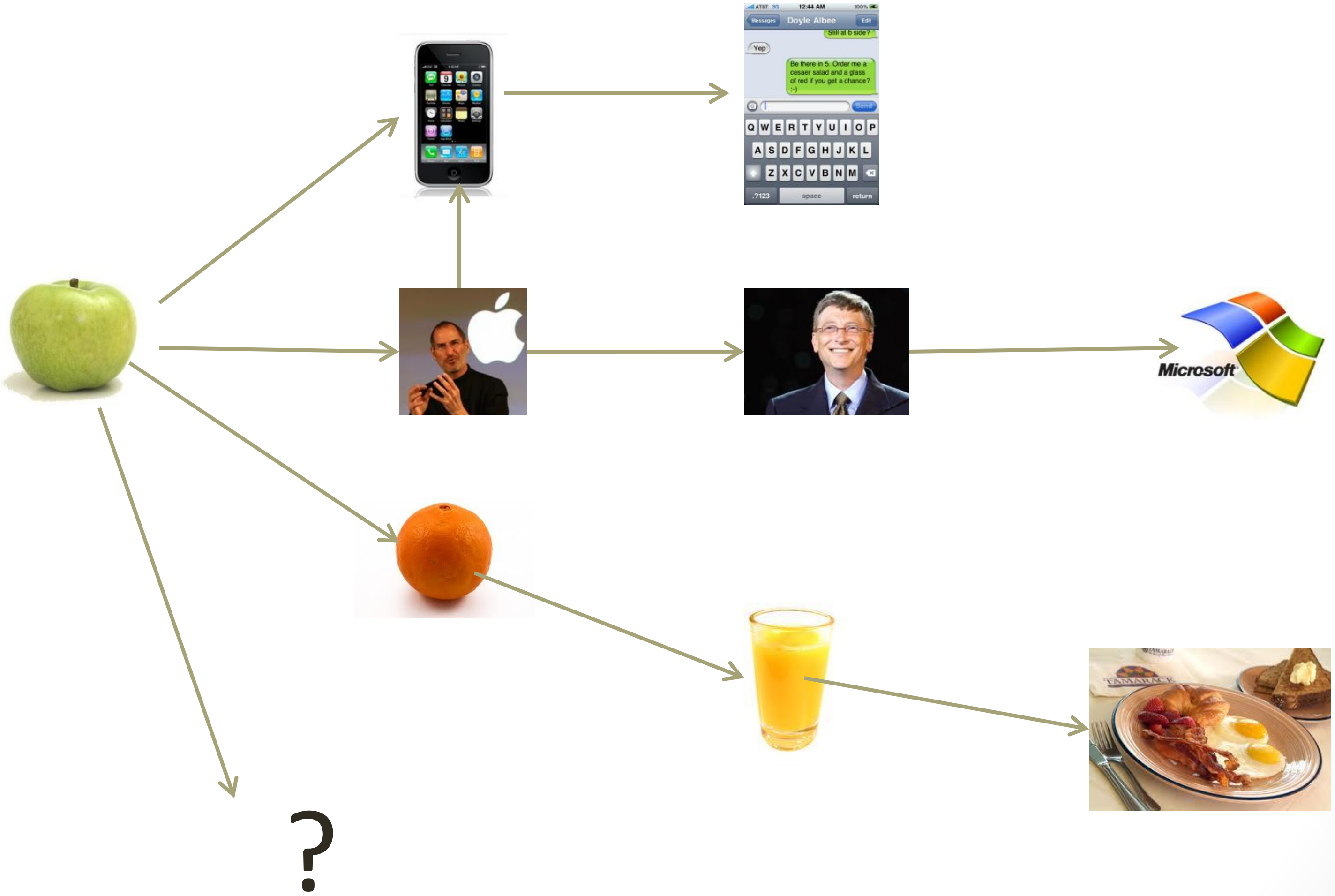
- Human Memory is semantic
- Human memory is associative
- Human memory is lossy

Human memory is Semantic

- Memorize: nbccbsabc
- Memorize: tkqizrlwp
- 3 Chunks vs. 9 Chunks!
- **Usability Goal**: Minimize Number of Chunks

Source: *The magical number seven, plus or minus two* [Miller 56]

Human memory is Associative



Still at b side?

Yep

Be there in 5. Order me a cesaer salad and a glass of red if you get a chance? :-)

Cues

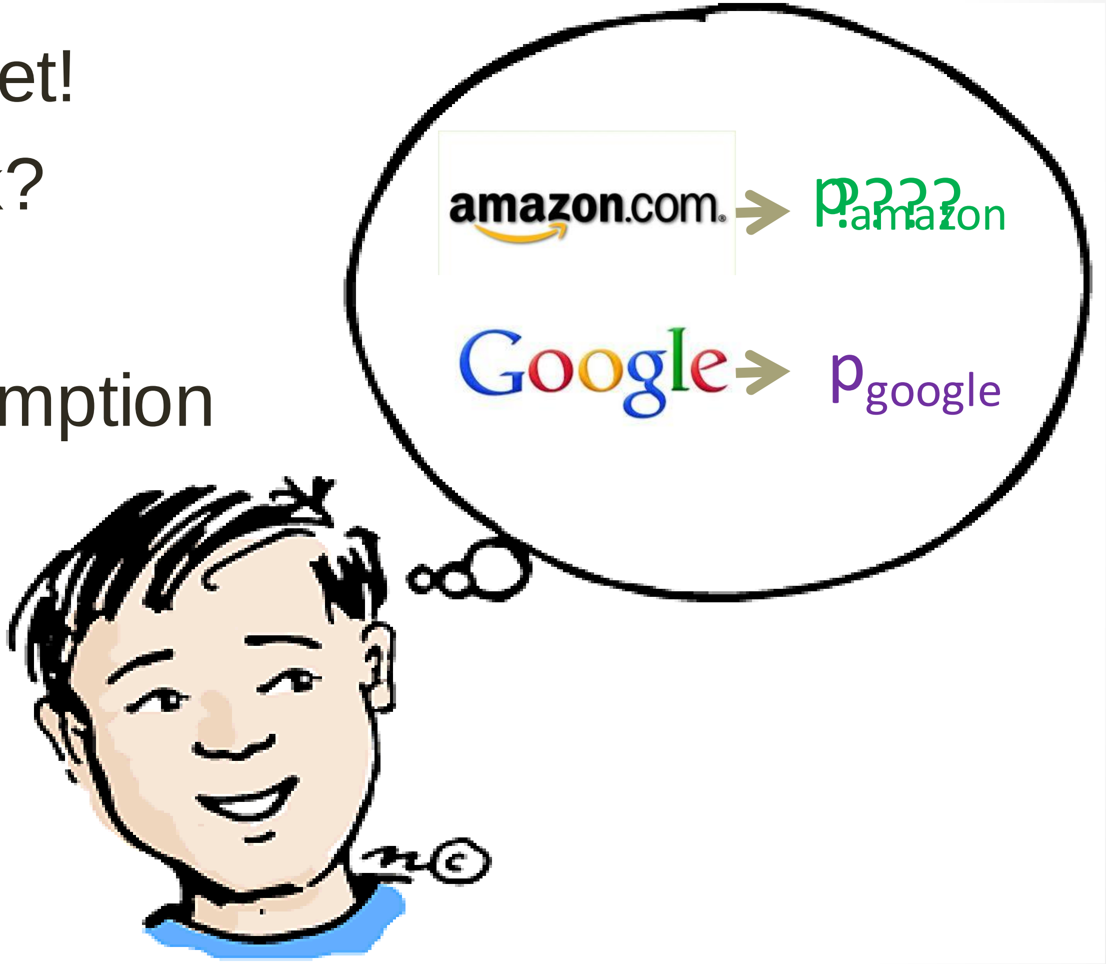
- Cue: context when a memory is stored

- Surrounding Environment
  - Sounds
  - Visual Surroundings
  - Web Site
  - ….

- As time passes we forget some of this context…

Human memory is Lossy

- Rehearse or Forget!
  - How much work?
- Quantify Usability
  - Rehearsal Assumption



Usability Question

- **Important Question**: Are human inherently bad at remembering random information?

- **Answer**: Not really, with proper training

- **Paper**: **Towards reliable storage of 56-bit secrets in human memory (USENIX-2014)**

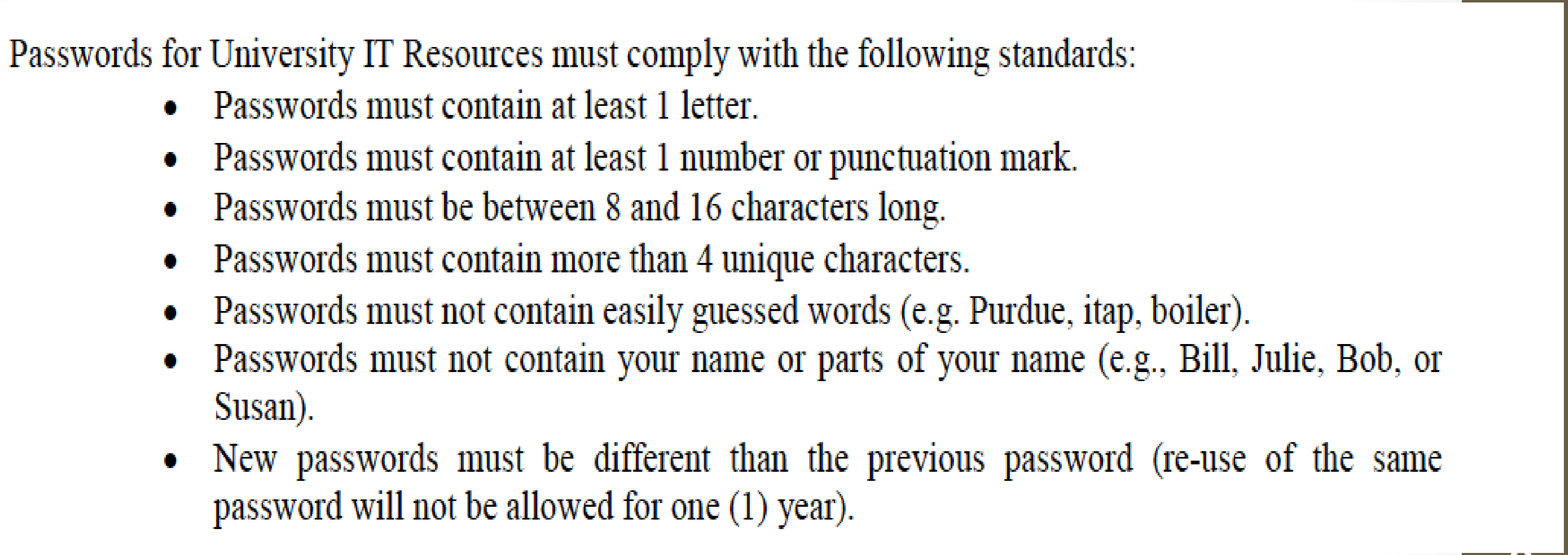
Example of Weak Passwords (Wikipedia)

- **Default passwords (as supplied by the system vendor and meant to be changed at installation time)**: *password*, *default*, *admin*, *guest*, etc.
- **Dictionary words**: *chameleon*, *RedSox*, *sandbags*, *bunnyhop!*, *IntenseCrabtree*, etc.
- **Words with numbers appended**: *password1*, *deer2000*, *john1234*, etc.,
- **Words with simple obfuscation**: *p@ssw0rd*, *l33th4x0r*, *g0ldf1sh*, etc.
- **Doubled words**: *crabcrab*, *stopstop*, *treetree*, *passpass*, etc., can be easily tested automatically.

Example of Weak Passwords (Wikipedia)

- **Common sequences from a keyboard row**: *qwerty*, *12345*, *asdfgh*, *fred*, etc.
- **Numeric sequences based on well known numbers** such as 911, 314159, or 27182, etc.,
- **IDs**: *jsmith123*, *1/1/1970*, *555–1234*, etc.,
- **Personal Info**: license plate number, SSN, telephone number, student ID, address, birthday, relative's or pet's names, etc.,
  - Can easily be tested automatically after a simple investigation of person's details.

Password Composition Policy



Passwords for University IT Resources must comply with the following standards:

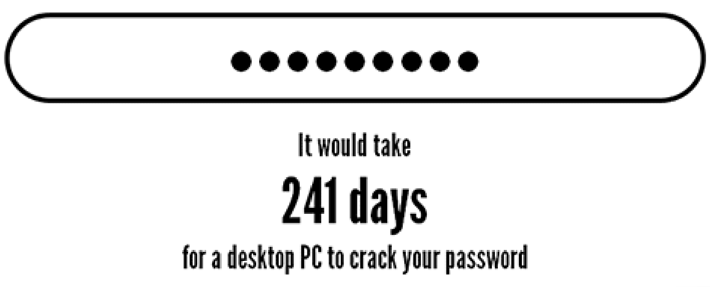
- Passwords must contain at least 1 letter.
- Passwords must contain at least 1 number or punctuation mark.
- Passwords must be between 8 and 16 characters long.
- Passwords must contain more than 4 unique characters.
- Passwords must not contain easily guessed words (e.g. Purdue, itap, boiler).
- Passwords must not contain your name or parts of your name (e.g., Bill, Julie, Bob, or Susan).
- New passwords must be different than the previous password (re-use of the same password will not be allowed for one (1) year).

Password Generated: P@ssw0rd1

Password Strength

- One possible approach of deterring users from creating weak passwords is to notify them whenever they have created a weak password

- Just this information is sometimes good enough to make the user create a stronger password

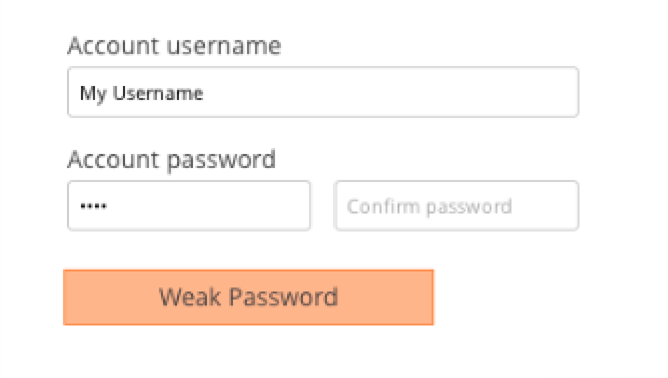


●●●●●●●●●●

It would take

241 days

for a desktop PC to crack your password

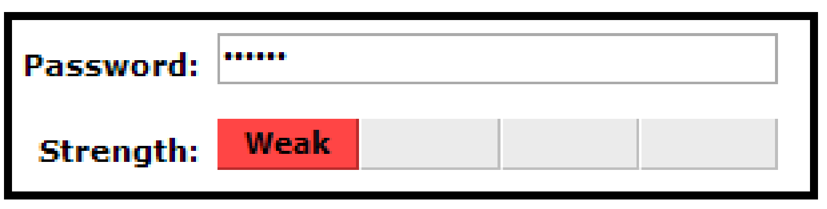


Account username

My Username

Account password

••••



Password: ••••••

Strength: Weak

Password Strength

- The average number of guesses the attacker must make to find the correct password
  - Determined by how unpredictable the password is, including how long the password is, what set of symbols it is drawn from, and how it is created.

- The ease with which an attacker can check the validity of a guessed password
  - Determined by how the password is stored, how the checking is done, and any limitation on trying passwords

Password Entropy

- The entropy bits of a password (also known as **guess entropy**), i.e., the information entropy of a password, measured in bits, is
  - The base-2 logarithm of the number of guesses needed to find the password with certainty
  - A password with, say, 42 bits of strength calculated in this way would be as strong as a string of 42 bits chosen randomly
  - Adding one bit of entropy to a password doubles the number of guesses required

Password Entropy Estimation

- People are bad at achieving sufficient entropy to produce satisfactory passwords
- NIST suggests the following scheme to estimate the entropy of human-generated passwords:
  - The entropy of the 1st character is 4 bits;
  - The entropy of the next 7 characters are 2 bits per character;
  - The 9th through the 20th character has 1.5 bits of entropy per character;
  - Characters 21 and above have 1 bit of entropy per character.
- This would imply that an 8 character human-selected password has about 18 bits of entropy.

Towards a Better Estimation of Password Entropy

- NIST suggestion fails to consider usage of different category of characters: Lower-case letters, digits, upper-case letters, special symbols
- **Orders also matter**: "Password123!" should have different entropy from "ao3swPd!2s1r"
- **State of art**: Variable-order markov chains to model probability of different strings as passwords: "**A Study of Probabilistic Password Models**" by Ma, Yang, Luo, Li in IEEE S&P 2014.
- **Fundamental challenge**: there are different attack strategies out there, which try passwords with different ordering
- .

Mechanisms to Avoid Weak Passwords

- Allow long passphrases, forbid short passwords
- Randomly generate passwords when appropriate
- Give user suggestions/guidelines in choosing passwords
    - Example: Think of a sentence and select letters from it, “It’s 12 noon and I am hungry” => “I’S12&IAH”
    - Using both letter, numbers, and special characters
- Check the quality of user-selected passwords
    - Run dictionary attack tools and other sanity checks
    - Evaluate strength of a password and explain the weaknesses
- **Active research area**

Password Entropy and Usability

- Forcing users to only use randomly generated password is bad
- The “**Weakest Link**” security principle applies:
  - Often times, guessing passwords is not the weakest link
  - One can use various ways to reduce adversary’s abilities to test password guesses
  - **Forgotten password**:
    - The recovering method either has low security, or costs lots of money
    - It creates a weaker link.

Relevant Security Principle

- **Psychological acceptability**:
  - It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors.
  - Taken from Saltzer & Schroeder: “**The Protection of Information in Computer Systems**”, which identifies 8 security principles, including the “open design” principle

Threats to Passwords

- Eavesdropping (insecure channel between client and server)
- Login spoofing (human errors), shoulder surfing, keyloggers
- Offline dictionary attacks
- Social engineering (human errors)
  - Pretexting: creating and using an invented scenario (the pretext) to persuade a target to release information or perform an action and is usually done over the telephone
- Online guessing (weak passwords)

Offline Dictionary Attacks

- With the frequent data breaches offline dictionary attacks have become a real worry for system designers and security experts

Company	Victims
Adobe	2.9 million
Evernote	50 million
Twitter	250,000
Living Social	50 million

Password Storage (UNIX)

- The file /etc/passwd stores H(password) together with each user's login name, user id, home directory, login shell, etc.
  - H is essentially an one-way hash function
  - **Roger Needham and Mike Guy in the 1960s proposed storing password hashes**
  - The file /etc/passwd must be world readable
- Brute force attacks possible
  - How to most effectively brute-force when trying to obtain password of any account on a system with many accounts?

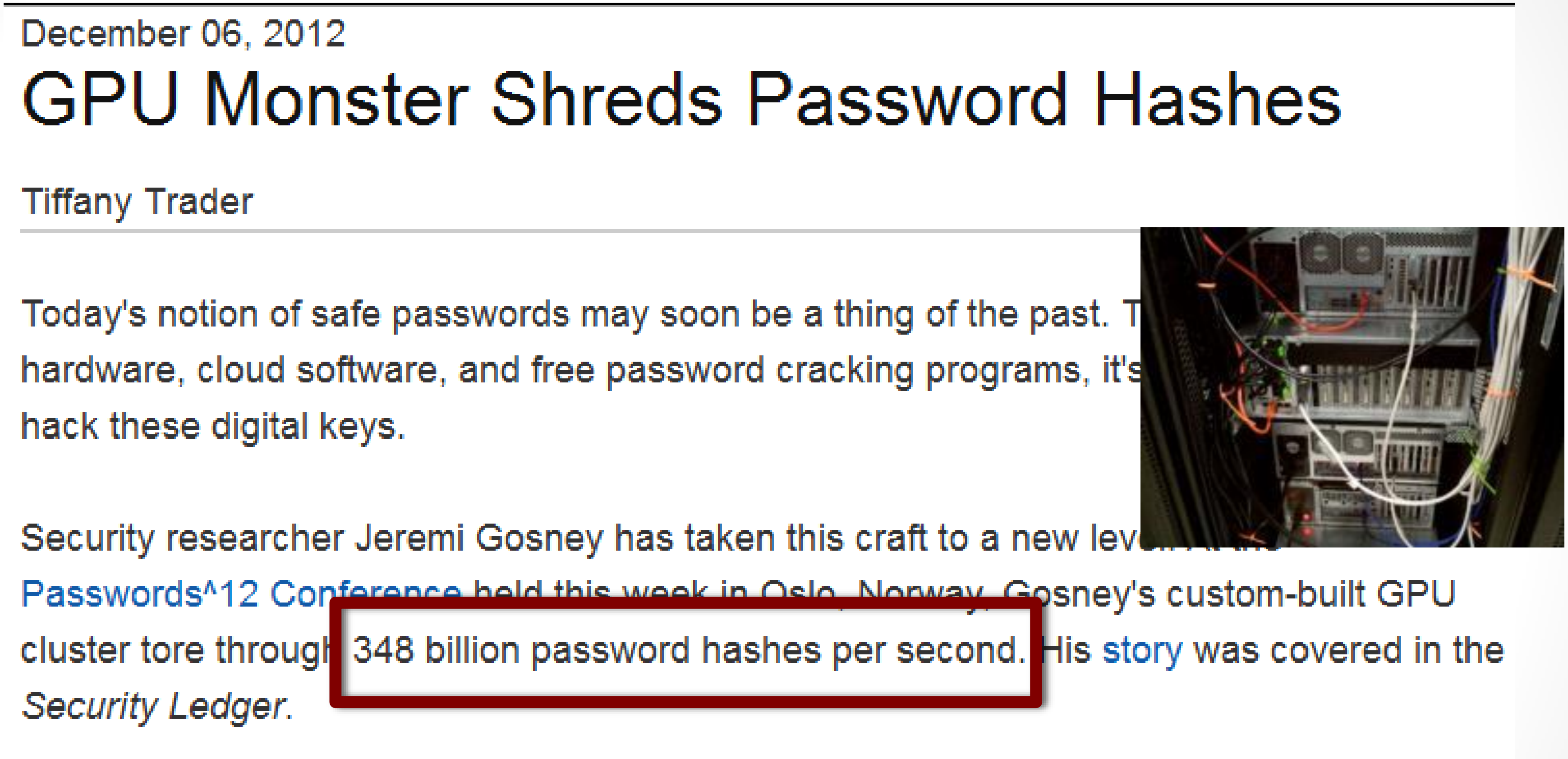
Password Salts

- More modern UNIX systems divide /etc/password into two files: /etc/password; and /etc/shadow (readable only by root)
- Store [r, H(password,r)] rather than H(password) in /etc/shadow
  - r is randomly chosen for each password
  - r is public, similar to IV in CBC & CTR modes
- ***Benefits***
  - Dictionary attacks much more difficult
    - Single account attack cost remains the same
  - Same password would have different hashes

Dictionary and Guessing Attacks

- Protect stored passwords with cryptography and access control
  - "***Defense in Depth***" principle is applicable:
    - Use multiple independent methods of defense, so that even if one layer fails, security is still not compromised
    - Example: Consider password dataset compromises
- Disable accounts with multiple failed attempts
- Require extra authentication mechanism

New Age of Offline Attacks



December 06, 2012

GPU Monster Shreds Password Hashes

Tiffany Trader

Today's notion of safe passwords may soon be a thing of the past. T
hardware, cloud software, and free password cracking programs, it's
hack these digital keys.

Security researcher Jeremi Gosney has taken this craft to a new level. At the Passwords^12 Conference held this week in Oslo, Norway, Gosney's custom-built GPU cluster tore through 348 billion password hashes per second. His story was covered in the *Security Ledger*.

New Age of Offline Attacks

- Attackers are building ASIC (Application Specific Integrated Circuits) for password cracking

- They are very efficient in calculating hash values, e.g., **355 million SHA2 hashes/s**

- Relies on Graphical Processing Units (GPUs)
- http://hashcat.net/oclhashcat/

Defenses against Offline Attacks

- Intentionally make the hash functions slow, e.g., bcrypt, scrypt

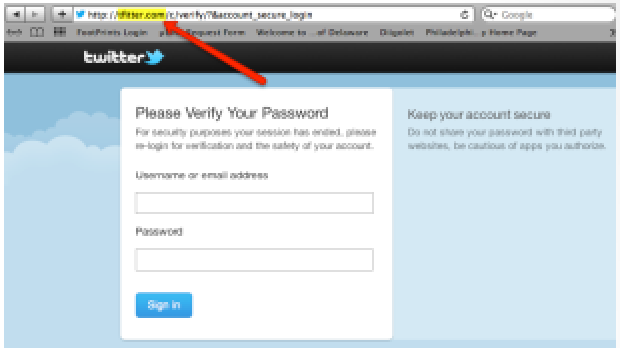
- Some on-going work on cost asymmetric hash function designs (CASH)
  - Easy to verify a hash
  - Difficult to carry out a dictionary/brute-force attack

Attack Strategy

Dumb Attacker	Smart Attacker
AAAAAA	password
AAAAAB	iloveyou
AAAAAC	monkey
AAAAAD	12345678
AAAAAE	password1
……	……

Login Spoofing

- **Login Spoofing Attacks**:
  - write a program showing a login window on screen and record the passwords
  - put su in current directory
- **Defense**: ***Trusted Path***
  - Mechanism that provides confidence that the user is communicating with the real intended server
    - Attackers can't intercept or modify whatever information is being communicated.
    - Defends attacks such as fake login programs
  - **Example**: Ctrl+Alt+Del for log in on Windows
    - Causes a non-maskable interrupt that can only be intercepted by the operating system, guaranteeing that the login window cannot be spoofed



Please Verify Your Password

For security purposes your session has ended, please re-login for verification and the safety of your account.

Username or email address

Password

Keep your account secure

Do not share your password with third party websites, be cautious of apps you authorize.

Spoofing Attack on the Web

- Phishing attacks
  - Attempting to acquire sensitive information such as usernames and passwords details by masquerading as a trustworthy entity in electronic communication.
- Website forgery
  - Set up fake websites that look like e-commerce sites and trick users into visiting the sites and entering sensitive info
- ***Defenses***
  - Browser filtering of known phishing sites
  - Cryptographic authentication of servers
  - User-configured authentication of servers
    - To ensure that the site is the one the human user has in mind
    - E.g., site key, pre-selected picture/phrases

KeyLogging

- Threats from insecure client side
- ***Keystroke logging*** is the action of logging the keys typed on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored.
- ***Software-based***: key-stroke events, grab web forms, analyze HTTP packets
- ***Hardware-based***: Connectors, wireless sniffers, acoustic based
- ***Defenses***: Anti-spyware, network monitors, on-screen soft keyboard, automatic form filler, etc.
- In general difficult to deal with once on the system

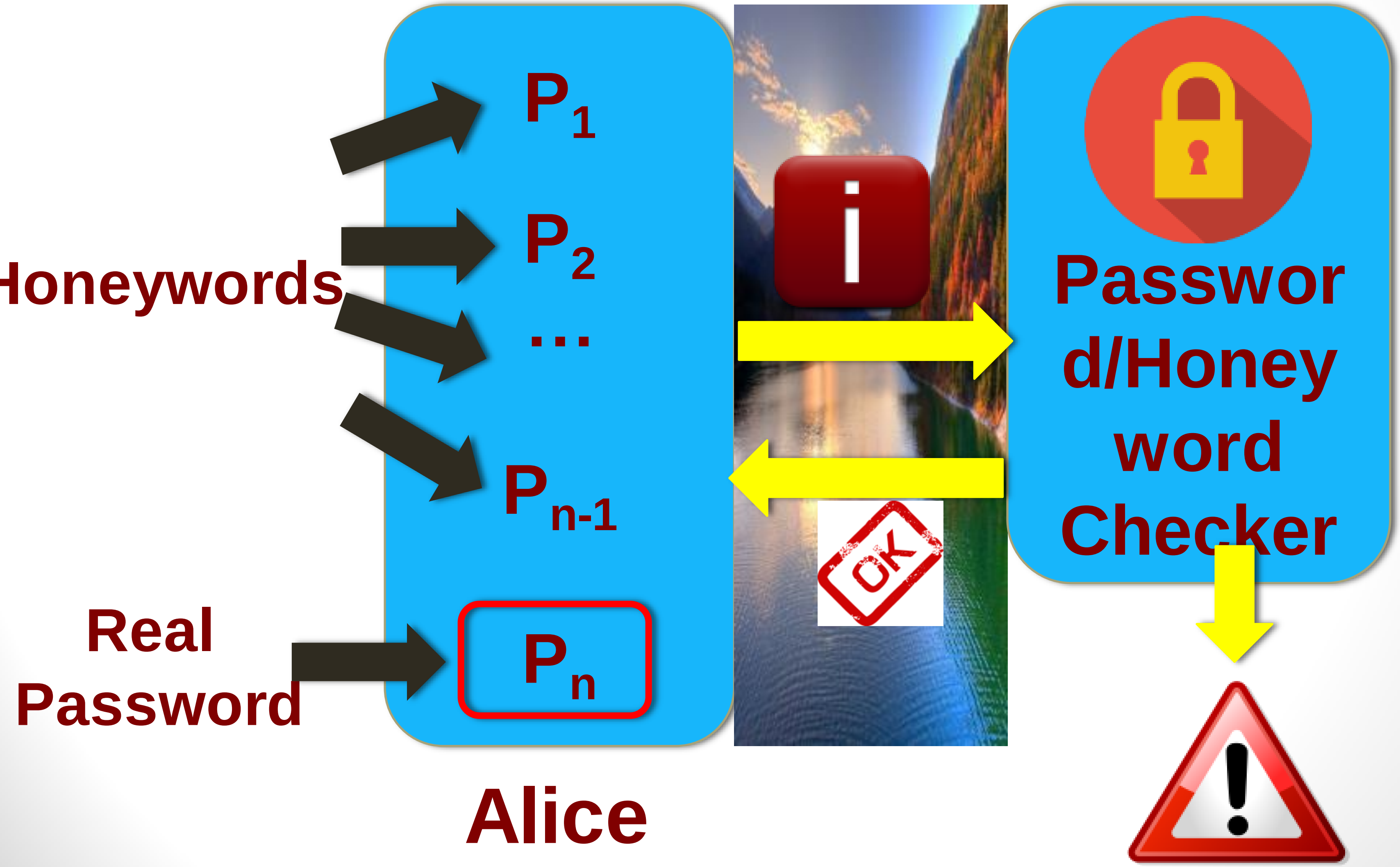
Recent Study by Us

How different people create the same passwords?

31.5%

Alternative Proposals

Honeywords: Making Password Cracking Detectable



Honeywords

P_1

P_2

…

P_{n-1}

Real Password

P_n

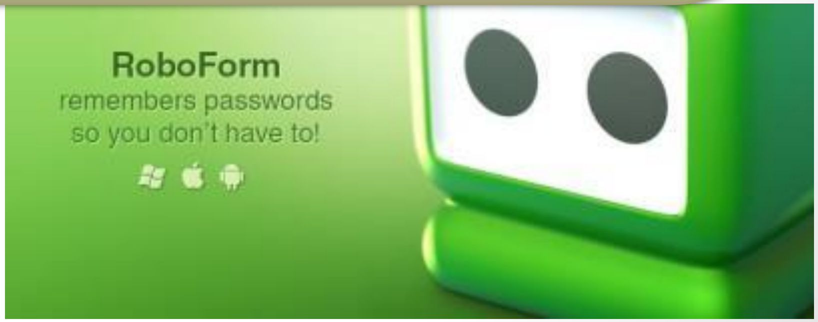
Alice

Password/Honeyword Checker

Password Managers

- A software that generates random

What is the problem with having a password manager manage your passwords?



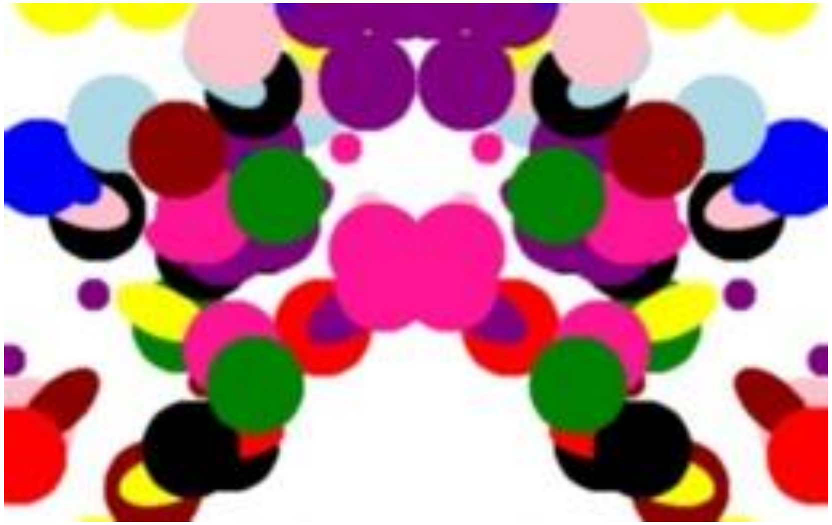
Password Managers

One single point of failure.

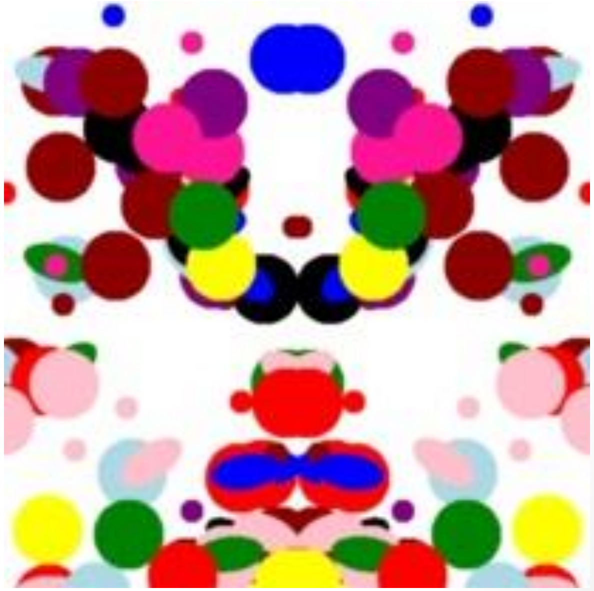
One master password protecting all your passwords.

GOTCHA

- Shows random computer generated inkblot images
- Depends on human imagination to give it an interpretation



Steroid Cow



Evil Clown

Naturally Rehearsing Password

Public

Private

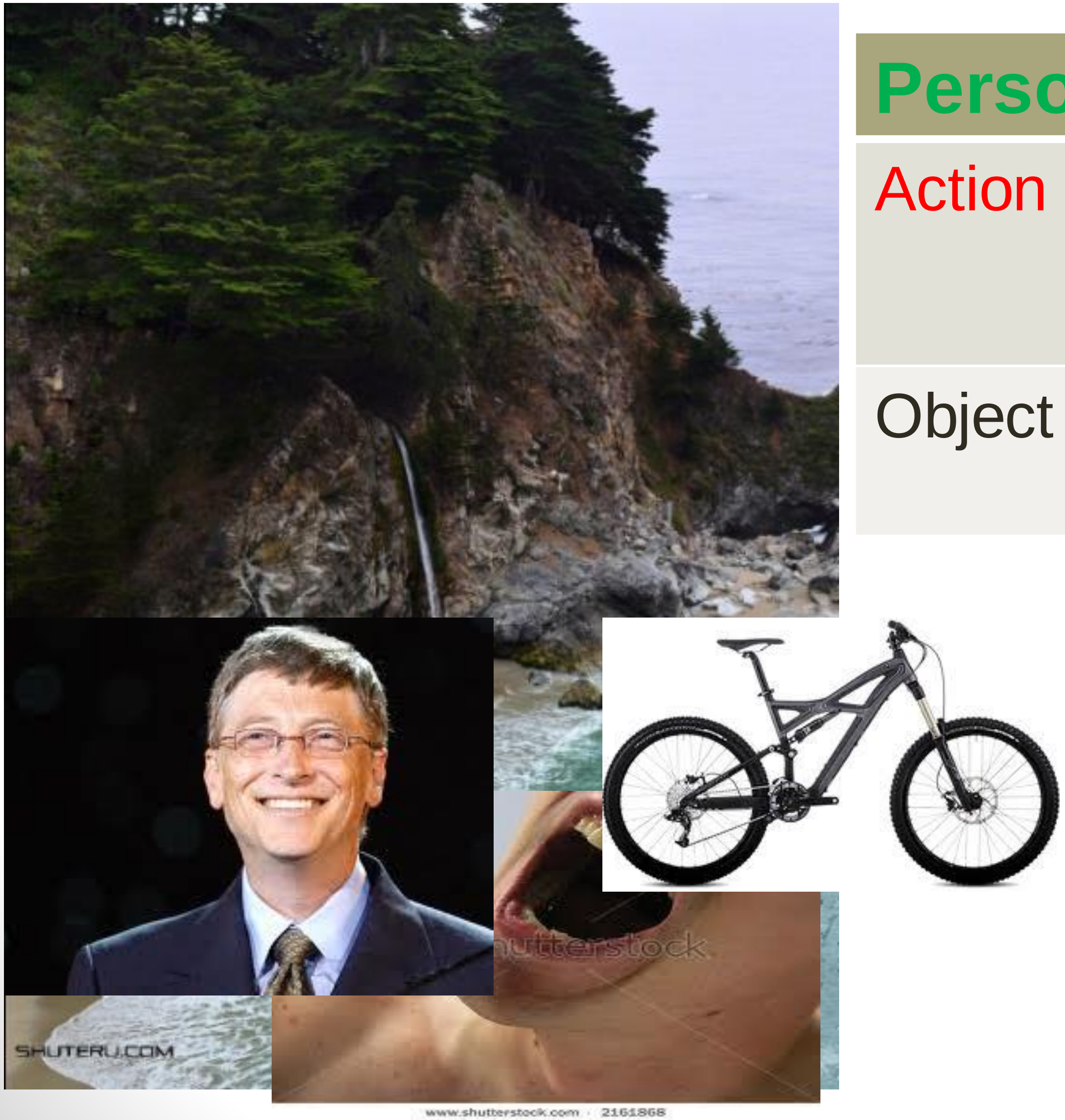
Action: kicking

Object: penguin

Naturally Rehearsing Password

Person	Alan Turing
Action	Kissing
Object	Piranha

Naturally Rehearsing Password

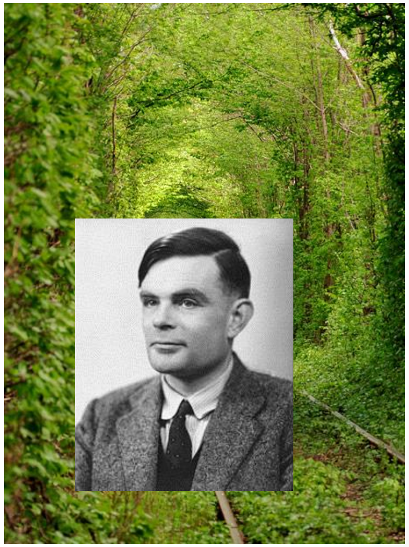


Person	Bill Gates
Action	swallowing
Object	bike

Using the Password

PayPal

…

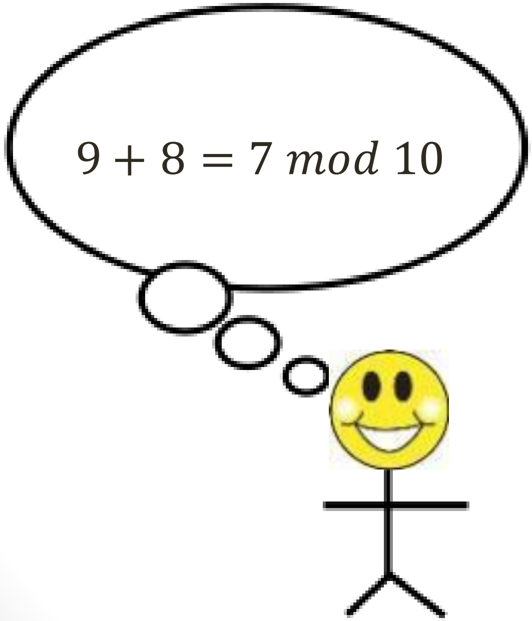


Pwd

Kic+Pen + … + Kis+pir

Human Computable Password

- Restricted
  - Simple operations (addition, lookup)
  - Operations performed in memory (limited space)



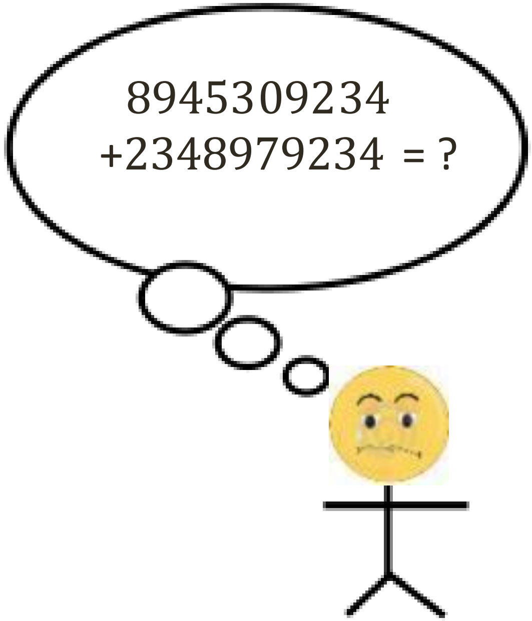


Image to Digit Mapping

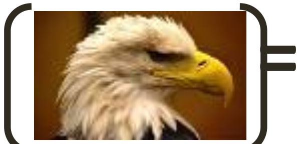
Image I			…	
□(I)	9	3	…	6

Initialization:

User Memorizes Random Mapping

Example: n=30 images

Mnemonics

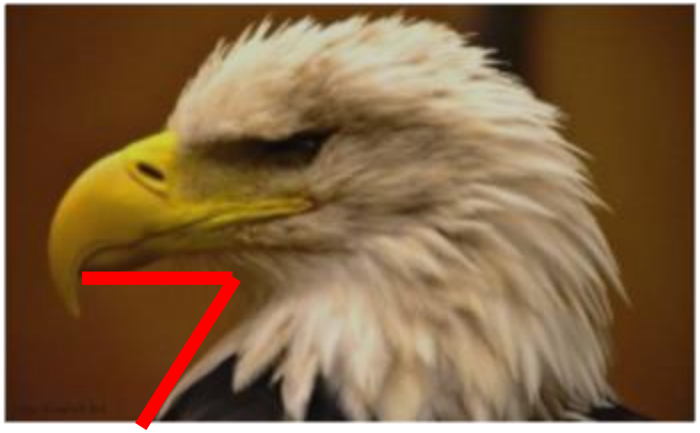
□ () = 4

Instruction: Remember that the eagle has a gold beak. There are four letters in “gold” and “beak”.

Mnemonics

- ( ) = **7**

Instruction: Trace the eagles body from the bottom of the eagle’s beak down to the bottom of the picture. It looks like the number 7.



Challenge Response

Response:

□ (**) + □(** **) mod**

10

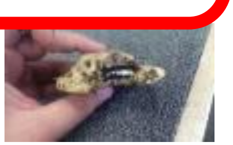
= 9+3 mod 10 = 2

0

1

2

3



4

5

6

7

8

9

Biometrics

- Your fingerprint is your ID!

- Your fingerprint is pretty unique

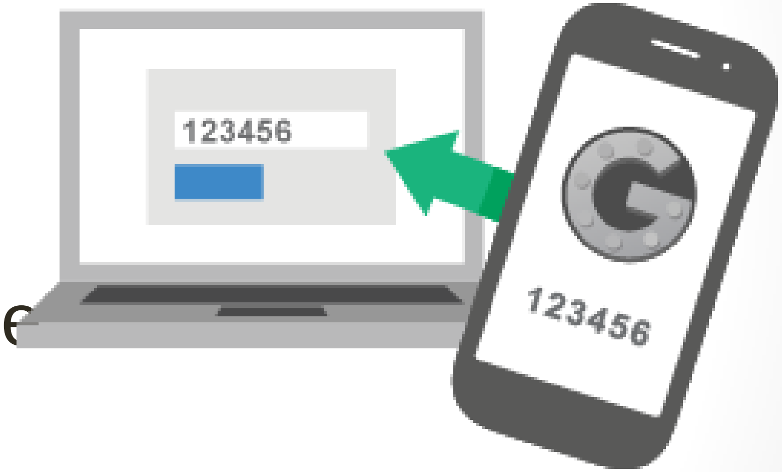
- Your fingerprint is convenient to carry.

Biometrics

- Your fingerprint is your ID!
  - Your fingerprint is a lot more valuable to other people than it used to be
- Your fingerprint is pretty unique
  - You have limited number of biometrics, so if Google and Microsoft use the same biometric, they can authenticate as you to each other
- Your fingerprint is convenient to carry
  - Unfortunately, biometric readers are a lot less convenient to deploy. They generally require special hardware

Two Factor Authentication

- Somebody steals your password, you can steal be safe
- Requiring two factor authentication all the time is not very usable



Additional Materials

- TED talk by Bruce Schneier on trade-off: http://www.ted.com/talks/bruce_schneier#t-625467
- TED talk by Lorrie Faith Cranor on passwords: http://www.ted.com/talks/lorrie_faith_cranor_what_s_wrong_with_your_pa_w0rd#t-764198
- Famous XKCD comic on password strength: https://xkcd.com/936/
- News article explaining password cracking strategy: http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-out-of-your-passwords/2/
- Hashcat password cracking website: http://hashcat.net/oclhashcat/

Interesting Papers

- Towards Reliable Storage of 56-bit Secrets in Human Memory: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-bonneau.pdf
- GOTCHA Password Hackers!: http://www.cs.cmu.edu/~jblocki/papers/aisec2013-fullversion.pdf
- Naturally-Rehearsing Passwords: http://www.cs.cmu.edu/~jblocki/crypto2013.pdf
- The Magical Number Seven, Plus or Minus Two: http://www.psych.utoronto.ca/users/peterson/psy430s2001/Miller%20GA%20Magical%20Seven%20Psych%20Review%201955.pdf
- A Study of Probabilistic Password Models : http://www.ieee-security.org/TC/SP2014/papers/AStudyofProbabilisticPasswordModels.pdf
- The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes: http://www.cl.cam.ac.uk/~fms27/papers/2012-BonneauHerOorSta-password--oakland.pdf
- Passwords and the Evolution of Imperfect Authentication: http://research.microsoft.com/pubs/250408/passwordsAndImperfectAuth.pdf

Acknowledgement

Most of the materials of this slide deck is taken from the slides of

Ninghui Li,

Lorrie Faith Cranor, and

Jeremiah Blocki

Next Topic

Continuation of Public Key Cryptography