IERG4210 Web Programming and Security

Course Website: https://course.ie.cuhk.edu.hk/~ierg4210/
Live FB Feedback Group: https://fb.com/groups/ierg4210.2014spring/

Course Overview

Dr. Adonis Fung
phfung@ie.cuhk.edu.hk

Information Engineering, CUHK
Product Security Engineering, Yahoo!

Copyright. All Rights Reserved.

Teaching Team

- Instructor: Dr. Adonis FUNG (Adon)
  - Office Location: Yahoo! 15/F Caroline Centre, Causeway Bay, HK
  - Office Hours: Preferably by email appointment
    Q&As most encouraged thru the Facebook Group
  - Email: phfung@ie.cuhk.edu.hk

- TA: Wenrui Diao
  - Office Location: SHB801
  - Email: dw013@ie.cuhk.edu.hk
- TA: Benedict Mak
  - Office Location: SHB803
  - Email: mlt014@ie.cuhk.edu.hk

- TA: pending
  - Office Location: TBD
  - Email: TBD

Copyright. All Rights Reserved.

Instructor Introduction

- Full-time Security Engineer at Yahoo!
  - Re-inventing web vulnerability scanning infrastructure
  - Deploying security measures across the globe
- Part-time Instructor at CUHK
  - My honor to teach IERG4210 the 3rd time
  - Thanks to Prof. DM Chiu and Prof. KW Cheung
- Researches
  - Interesting discovery reported in news headline in 2009
  - SSL Enforcement, Web Vuln Scanning, Profile Pollution
  - Awarded Fellowship, Grant, Outstanding TA twice
  - Visited Georgia Tech. during 2013

Copyright. All Rights Reserved.

Course Description

- Web Programming and Security:
  - The programming languages for both client- and server-side will be introduced, with security design principles and common vulnerabilities highlighted early on
  - Open standards and real-world security case studies will be used for illustrations
  - Optimization and performance issues will also be covered
- This course also extends to the security threats confronting web browsers, transport protocols and web servers, as well as optionally the mobile and cloud computing.
- Being security-conscious throughout the development cycle, students will have the opportunity to practice with web programming assignment.

"Economic prosperity in the 21st century will depend on cybersecurity"
- U.S. President Obama, 2009

Copyright. All Rights Reserved.

Topics to be Covered (1/2)

- Web Architecture
  - HTTP, URL, etc

- Web Dev. Languages
  - HTML, CSS, PHP, (No)SQL
  - JavaScript heavy
- Web Dev. Components
  - User Interface Design
  - Forms Handling
  - Database Management
  - Session Management & Auth

- Web App. Security:
  - 8 Security Principles
  - Security Goals: Confidentiality, Integrity, Availability, Auth, Non-repudiation
  - Browser Security Model: SOPs
  - Mashup Devel and Security / Cross-origin Communications
  - Top Application Security Risks

Copyright. All Rights Reserved.

Topics to be Covered (2/2)

- Transport Layer and Browser Security
  - TLS/SSL, PKI, Certificates, Digital Signatures, SSH
  - Cert Pinning, 2FA, XSS Audits, Content Security Policy, Extensions, etc
- Security Testing
  - Penetration Testing
  - Web App Crawling and Scanning

- Building Fast and Scalable WebApp, plus Optimizations
  - Scalability Concerns/Solutions
  - Using Cloud Resources
  - Settings and Code Tweaks
  - Search Engine Optimizations

Scared off? Schedule to be adjusted according to your learning curve … :)

Copyright. All Rights Reserved.

Teaching Schedule

The most current version will be always posted at:
https://ierg4210.github.io/web/schedule.html

Schedule is subject to adjustment as needed

Time, class size, students' background, etc…

Copyright. All Rights Reserved.

Learning Outcomes

- Demonstrate understanding of the principles and techniques in the design and development of secure web applications
- Appraise and be inspired on how the web, ranging from browsers to servers, can be attacked and better secured
- Raise security awareness throughout the development of web applications as well as other engineering practices

- or additionally, :)
  - grasp "the tools to become fabulously wealthy" (quoted from: Stanford CS142 Web Applications course)
  - "the students will be able to find a good job and earn a good living :-)" (quoted from Prof. K.W. Cheung, IE)
  - Be a security practitioner/engineer/researcher

Copyright. All Rights Reserved.

News: Internet Security News in 2014

- Dec 28, 2014
  "US and British intelligence agencies undertake every effort imaginable to crack all types of encrypted Internet communication. The cloud, it seems, is full of holes."
  http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html
- Sept 11, 2014
  "Yahoo said the government threatened to fine the company US$250,000/day if it did not comply with the surveillance"
  http://www.cbsnews.com/news/yahoo-waged-court-fight-with-u-s-government-over-surveillance/
- Aug 26, 2014
  "At no time in history has there been a greater need to hire security professionals to protect and defend infrastructures from attacks."
  http://www.zdnet.com/article/cybersecurity-hiring-crisis-rockstars-anger-and-the-billion-dollar-problem/

Copyright. All Rights Reserved.

Learning Activities

- Weekly Lectures — Tues 3:30-6:15pm
  - teaching (~2.5 hours)
  - leave your feedback at the live FB Group
- Readings — Some book chapters/pages (30 min)
  - reinforcing your learning
- Tutorials — Time and Venue TBD (1 hr)
  - assignment guidance
  - reviews on important topics
- Assignments — Web development and hacking (7 x 7hrs)
  - all about practising what you learned

Student/Faculty Expectations on Teaching and Learning
http://www.erg.cuhk.edu.hk/Student-Faculty-Expectations

Copyright. All Rights Reserved.

Assessment Scheme

- Assignments (shopping cart in 6 phases) 40%
  1. HTML and CSS
  2. Form Handling and Image Upload
  3. Authentication
  4. AJAX Shopping List
  5. Payment Gateway Integration
  6. Free-style Features: Password Reset/Mashup
  7. Vulnerability Discovery and Peer-hacking
- Quizzes (mostly online, one during class) 10%
- ~~Computer-based Midterm Test~~
- Final Examination 50%

Scared off…? yeh, this course is very harsh. But you'll be rewarded by what to learn

Copyright. All Rights Reserved.

Grading Strategies (1/3)

- Assignments (40%)
  - To provide you with chances to code and practice what you learnt
  - An e-commerce website ($$) deserves protection far more than any other examples. Although source code for shopping carts is well-known everywhere (books/web), it may not be secure
  - >6 phases (and deadlines, approx. biweekly) towards a final product, i.e. shopping cart (example: amazon.com, walmart.com).
    - TAs will track students' progress
    - Invite weaker students to attend an interim demo, which is compulsory for students lagged behind in first 4 phases
      - To ensure students are capable of completing the basic parts
      - To forfeit the penalty of late submissions
  - The last phase will be peer-hacking

Copyright. All Rights Reserved.

Grading Strategies (2/3)

- Assignments (40%, continued)
  - Final demo: all students will be graded at the end of semester
    - 0% - 100% Result-oriented
      - Based on completion of the required features
      - No restrictions on the use of languages/libraries (Ruby and jQuery? fine!)
    - -75% Challenge-response Q&A
      - Be challenged in person on your understanding of code
      - Marks to be deducted if coded something you don't understanding

- Revision Quizzes (10%)
  - Picked up from some assigned readings (good for you!)
  - If online, 10-15 questions after class, count on 2 trials
  - One MC-style quiz during class in midterm (5%)
  - To ensure you understand what was taught in lectures

Copyright. All Rights Reserved.

Grading Strategies (3/3)

- Final Examination (50%)
  - Question types: less coding, more conceptual questions (tentative)
  - To assess your understanding on programming and security in general

MOST IMPORANTLY!!!!

- Testimonies of last semester: IERG4210 is exceptionally demanding!!
- BUT, in return, you learn the most needed skills from the job market
- AND, be inspired on the way you code and think!!

Copyright. All Rights Reserved.

Failed

- FAIL rate in 2012 Spring: >9%
  - No justification needed
  - May be MORE this term!
  - If you decided to be here, pls. be hardworking!!

9%

Otherwise, why not IERG4130 first?
(FYI, fail rate of other courses: <3%)

Copyright. All Rights Reserved.

Excell

- A-/A rate in 2012 Spring: ~ %
  - Smart and hardworking students should be awarded
  - RGS didn't complain so far :)

(FYI, rate of other courses: at most 30%)

%

Copyright. All Rights Reserved.

Assignment and Project Policies

- Submission Policies for Assignments and Project
  - Approximately two weeks of time/phase
  - Unfortunately, firm deadline and no extension :(
  - BUT, Early Submission Reward: Extension for future phases :)
    e.g. For every 48-hour in advance → 24-hour extension

Make good use of your time. Plan early. Start early.

- Honesty in Academic Work: the university policy
  - CUHK places very high importance on honesty in academic work submitted by students, and adopts a policy of *zero tolerance* on cheating in examinations and plagiarism. http://www.cuhk.edu.hk/policy/academichonesty/
- Ethical Hacking
  - You need to learn how to hack, or you don't know what to protect. BUT, apply HACKING SKILLS in a LEGAL and ETHICAL way

Copyright. All Rights Reserved.

Learning Resources

- Textbooks (FREE e-Books! via CU Library)
  - N. Daswani, C. Kern, and A. Kesavan, "Foundations of Security: What Every Programmer Needs to Know," 2007
  - D. Stuttard, and M. Pinto, "The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws," 2008
  - J. Resig, "Pro JavaScript Techniques," 2007
  - W. J. Gilmore, "Beginning PHP and MySQL: From Novice to Professional," 2010
  - D. Oehlman and S. Blanc, "Pro Android Web Apps: Develop for Android Using HTML5, CSS3 & JavaScript," 2011

- and, web resources are rich too…

No worries. Specific chapters to be assigned for bedtime reading. :)

  - Stanford University: CS142 Web Applications
  - CUHK: CSCI 4140 Open Source Software Project Development
  - Self-learning motto: "Google before asking us questions"

Copyright. All Rights Reserved.

- Internet Components
  - URL
  - Domain Name
  - IP Address
  - World Wide Web
- Evolution of the Web

OVERVIEW OF THE INTERNET

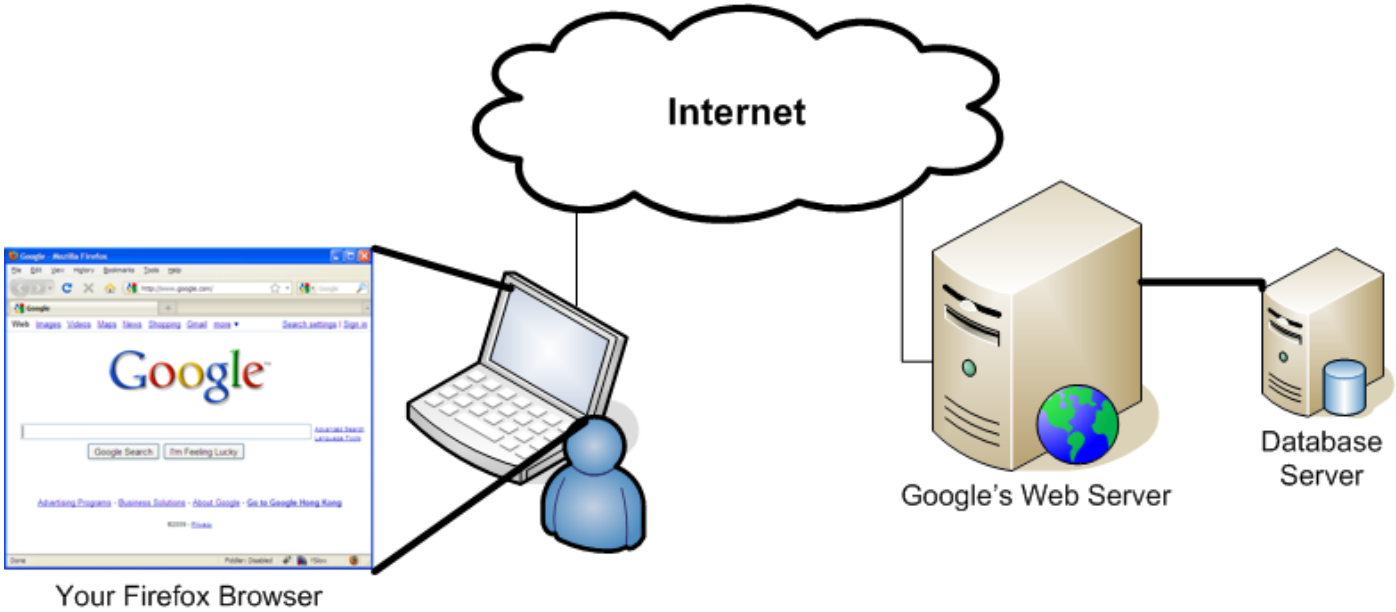
Copyright. All Rights Reserved.

Journey of Web Browsing

- To introduce the Web, let's begin from the user's perspective
  - What is being done… once you open a website with your browser?

To be briefly illustrated by heading all the way from URL to HTML…

URL > *Domain* > *IP Address* > *HTTP* > *HTML*



Copyright. All Rights Reserved.

Universal Resource Locator (URL)

URL	>	Domain	>	IP Address	>	HTTP	>	HTML

- URL is a string that references an Internet resource. For example:

http:	//	www . cuhk . edu . hk	: 80	/	english / index.html	?a=1&b=1	#top
protocol		domain name	port	folder	file	query string	fragment id
				resource path			

where 80 is the default port number for HTTP server (optional if using default)

- More examples: Can you tell the component names?
  - https://www.ebanking.hsbc.com.hk:443/1/2/logon
  - ftp://ftp.cuhk.edu.hk:21/

Copyright. All Rights Reserved.

Domain Name

URL	>	**Domain**	>	IP Address	>	HTTP	>	HTML

- Domain Name System (DNS) server resolves domain name to IP address(es) for ease of memorizing, or vice versa
  - Reference: wikipedia.org/wiki/Domain_Name_System

So, do you know when is reverse DNS lookup employed?

- Domain Name Examples:

- For `www.google.com`,
  - Top-level: `com`
  - Second-level: `google`
    - ~US$10/year
  - Subdomain: `www`

- For `www.cuhk.edu.hk`,
  - Country-coded top-level: `hk`
  - Second-level: `edu`
  - Third-level: `cuhk`
  - Subdomain/4th-level: `www`

- Command Shell (Demo):
  `nslookup www.cuhk.edu.hk`

```
C:\Users\user>nslookup www.cuhk.edu.hk
Server:  UnKnown
Address:  10.0.255.244

Non-authoritative answer:
Name:    www.cuhk.edu.hk
Address:  137.189.11.73
```


Copyright. All Rights Reserved.

IP Address

- IP Address is a numerical address that references a device connecting to a computer network using the Internet Protocol.
  - Take IERG3831/3841 for more information on routing protocols
  - Reference: http://en.wikipedia.org/wiki/IP_address
- Example: how about the DNS record of www.google.com?
  - Try to keep querying multiple times…
  - Round-robin DNS: A domain name is resolved to multiple IP addresses, to be swapped after each query
  - Achieving load balancing, often employed by high-traffic websites
  - Domain can be one-to-many mapping

```
C:\Users\user>nslookup www.google.com
Server:  UnKnown
Address:  10.0.255.244

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  2404:6800:8005::67
          74.125.71.147
          74.125.71.99
          74.125.71.103
          74.125.71.104
          74.125.71.105
          74.125.71.106
Aliases:  www.google.com
```


Copyright. All Rights Reserved.

Brief History of World Wide Web

- Wide Wide Web is the point-and-click system of navigating through information shared over the Internet by using hypertext
- Invented by a physicist Tim Berners-Lee at CERN for sharing physics findings
- In 1990, the HyperText Transfer Protocol (HTTP), first HTTP Server and Browser (WorldWideWeb, a HTML renderer) were all born
- Founders Chair at MIT

References:
http://www.w3.org/People/Berners-Lee/
http://info.cern.ch/

mostly history… tech details to be later covered

Copyright. All Rights Reserved.

BUT, they have all evolved…

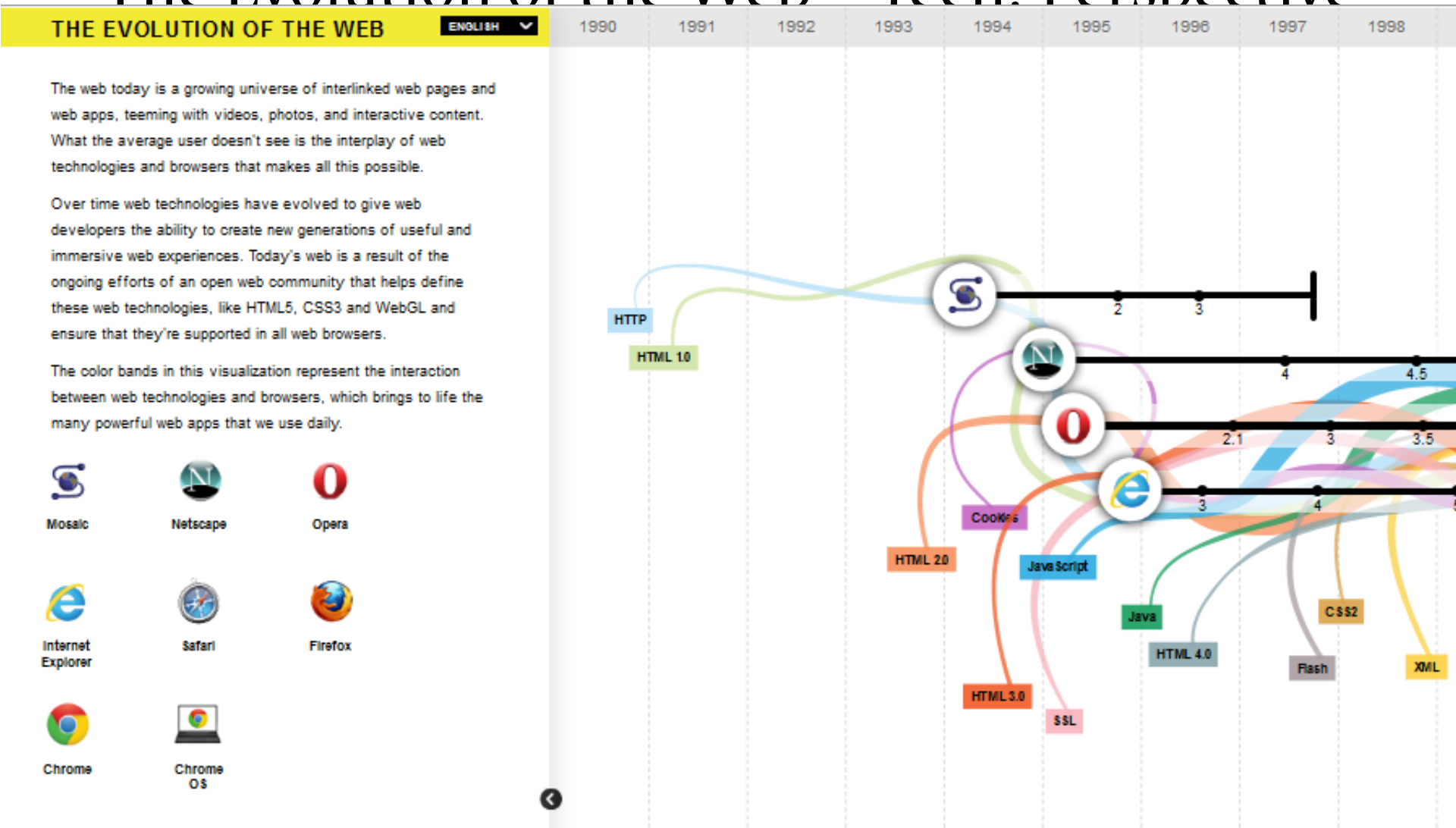
URL > *Domain* > *IP Address* > *HTTP* > *HTML*

	1990s	2010s
URL	example.com/?page=index	example.com/#!page=index ➔ Use of fragment id ➔ favors AJAX and browser history
Domain	longdomainname.com	goo.gl, bit.ly ➔ Use of URL Shortener
IP Address	137.189.11.73 (IPv4 addr of www.cuhk.edu.hk)	2405:3000:3:b0:137:189:11:71 (IPv6 addr of www.cuhk.edu.hk)
HTTP	HTTP/1.0	HTTP/1.1 Use of HTTPS, the cryptographic version of HTTP
HTML	HTML 1.0	JavaScript, HTML 5, XML, CSS 3, etc

- What's even more amazed: Browsers, Cloud, Mobile, etc…

Copyright. All Rights Reserved.

The Evolution of the Web – Tech. Perspective

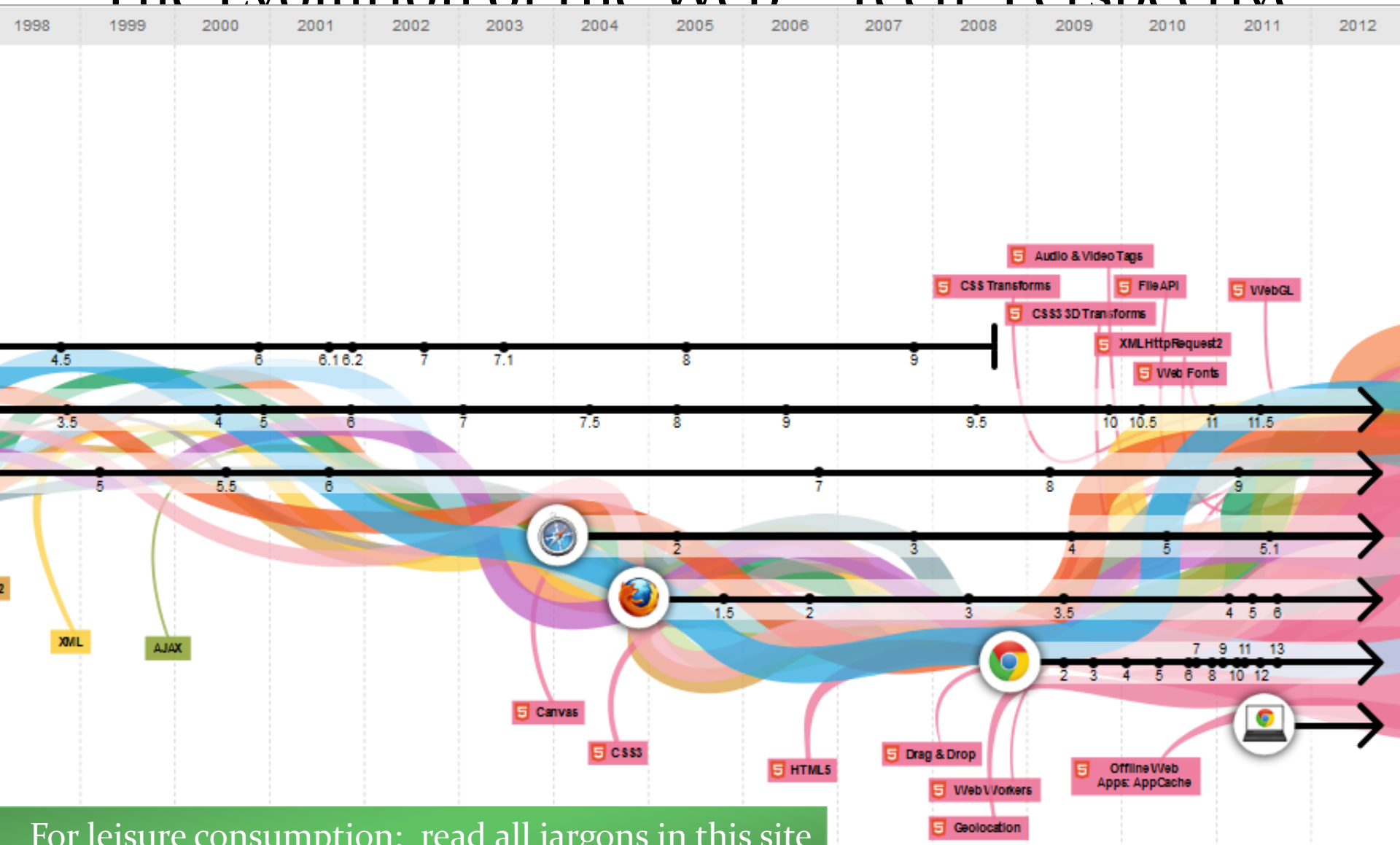


Take home assignment: read all jargons in this site

- Reference: http://evolutionofweb.appspot.com/

Copyright. All Rights Reserved.

The Evolution of the Web – Tech Perspective



For leisure consumption: read all jargons in this site

- Reference: http://evolutionofweb.appspot.com/

Copyright. All Rights Reserved.

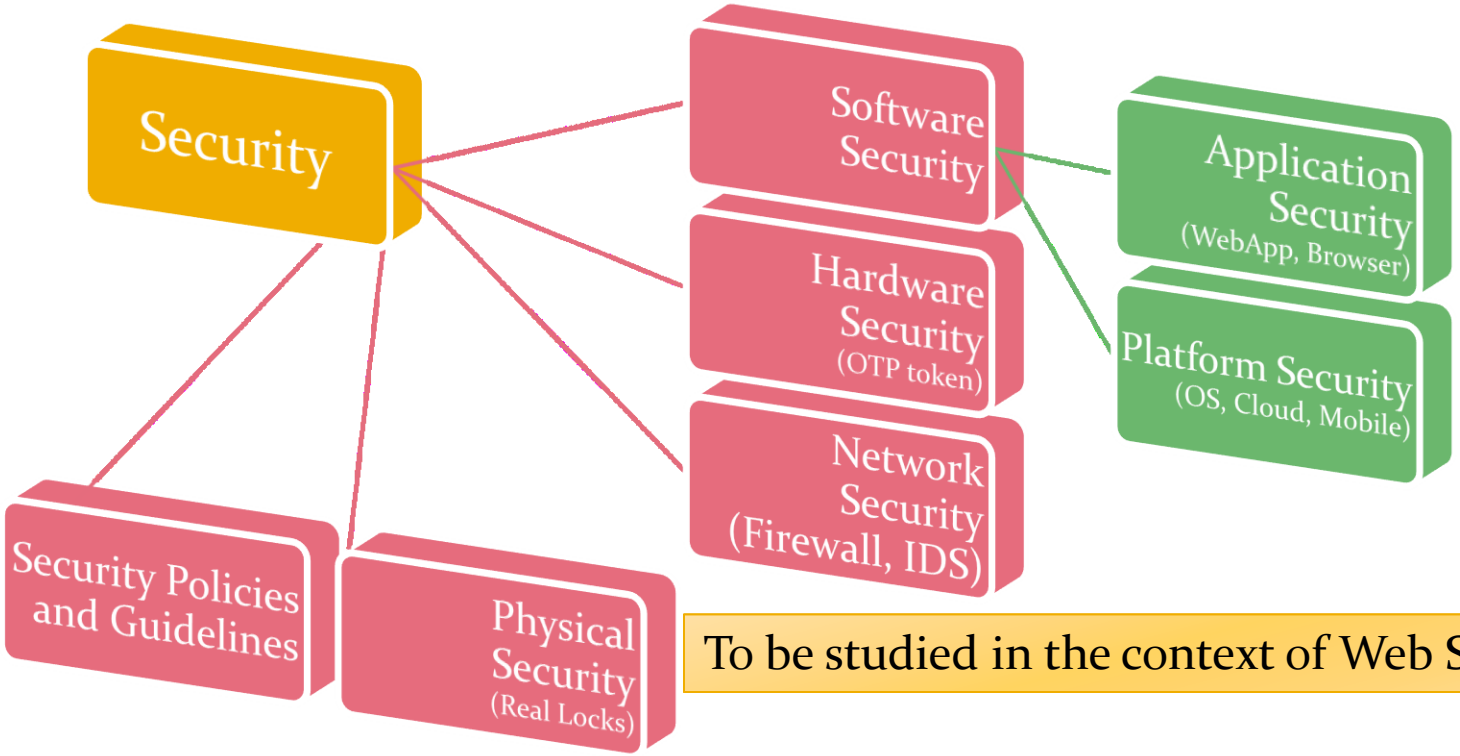
- Security is Holistic
- Secure Design Principles
- Information Security Goals
- Reflections on Security

SECURITY PRINCIPLES

Copyright. All Rights Reserved.

Security is Holistic

- Securing the system as a whole for greatest secruity
  - "Securing the system": prevent, detect, and respond
  - "as a whole": focusing not on any particular aspect but ALL aspects
  - "greatest security": usable, fault-tolerable, risk manageable



To be studied in the context of Web Security

Copyright. All Rights Reserved.

8 Secure Design Principles (1/6)

1. Securing the Weakest Link
   - Security is like a chain; a system is only as strong as its weakest link, which is most likely attacked
   - Risk Management – Nothing is perfectly secure unfortunately!!
     - Start by addressing the most serious risk (weakest) in your list, instead of those that is easiest to mitigate
     - Stop until all components are under a manageable level of security
   - Examples: (1) Attackers often bypass crypto, instead of breaking it
     (2) Un-educated users often click "yes" to everything

Copyright. All Rights Reserved.

8 Secure Design Principles (2/6)

2. Defense-in-Depth
   - Deploy holistic and diverse defense strategies so that even if one fails, the others can hopefully prevent a complete failure
   - Layers of defense. Promote redundancy. NO single point-of-failure!
   - Examples: (1) prevent, detect, contain, and recover
     (2) One-Time Password as 2nd defense if password is stolen

3. Secure Failure
   - A complex system can unavoidably break down, but it should not revert to insecure behavior
   - Promote the use of try/catch blocks in programming
   - Examples: (1) Disclose sensitive info in error or debugging messages?
     (2) A lift must not fall even if it fails to work

Copyright. All Rights Reserved.

8 Secure Design Principles (3/6)

4. Least-privilege
   - A user or program is granted the minimum amount of privilege (access and time granted) that is just sufficient to complete a task
   - Examples: (1) Block unneeded port numbers to reduce attack surfaces
     (2) Permissions are restricted even if an app is compromised

5. Compartmentalization / Separation of Privilege
   - Break the system up into as many isolated units as possible to minimize potential damage
   - Examples: (1) Run a forum on a server that stores credit card numbers?
     (2) Sandboxing tasks to run with its own privileges

Copyright. All Rights Reserved.

8 Secure Design Principles (4/6)

6. Simplicity
   - KISS (Keep it simple, stupid!): Keep a system design and implementation as clean and straightforward as possible
     - Reuse Off-the-shelf libraries whenever possible to stay simple
     - "Security by Obscurity" is not secure << reverse engineering
       - Crypto is secure not by hiding its algorithm but only a secret key
     - Example: complex system is always hard to maintain and analyze

Copyright. All Rights Reserved.

8 Secure Design Principles (4/6)

6. Simplicity
   - Usability: Design an idiot-proof and usable system as users (1) are lazy, (2) never read manual, and (2) ignore security if given a choice.
     - Considerations on trade-off security with convenience
     - Example: "You're going to lose money?" Users click "yes" to everything.
   - Secure Defaults: Deploy applications that are with more secure configurations by default. Let advanced users to relax them.
     - Example: Firewall should be defaulted to deny all traffic

Copyright. All Rights Reserved.

8 Secure Design Principles (4/6)

7. Promote Privacy
   - Privacy to be collected and used fairly as agreed by users (FB?)
   - A system should keep users' sensitive information confidential
   - Example: Stealing credit card numbers and IDs from insecure database

8. Don't extend trust easily. Be skeptical!
   - Be reluctant to trust your clients, who may not use a system as intended. (i.e. always validate and sanitize user inputs – one of the key problems for web application security)
   - Be reluctant to trust external components, which may not be built by security experts
   - Be reluctant to trust yourself, who may think what you built is perfect

Reference: IBM developerWorks, "Software Security Principles by G. McGraw, and J Viega," 01 Oct, 2000. [website removed. included as your reading]

Copyright. All Rights Reserved.

Information Security Goals

- CIA Core Goals: Confidentiality, Integrity and Availability
  - Confidentiality and Integrity depends on Authentication and Authorization

Confidentiality

- Information be revealed to only authorized entities (keep things secret to auth people)

Integrity

- Information be protected from unauthorized modification (prevent unauth data tampering)

Authentication + Authorization

= Ensures who and what are authorized

Availability

- Information be accessible when required (mitigation of Denial-of-Service attacks)

Accountability
(maintain audit log)

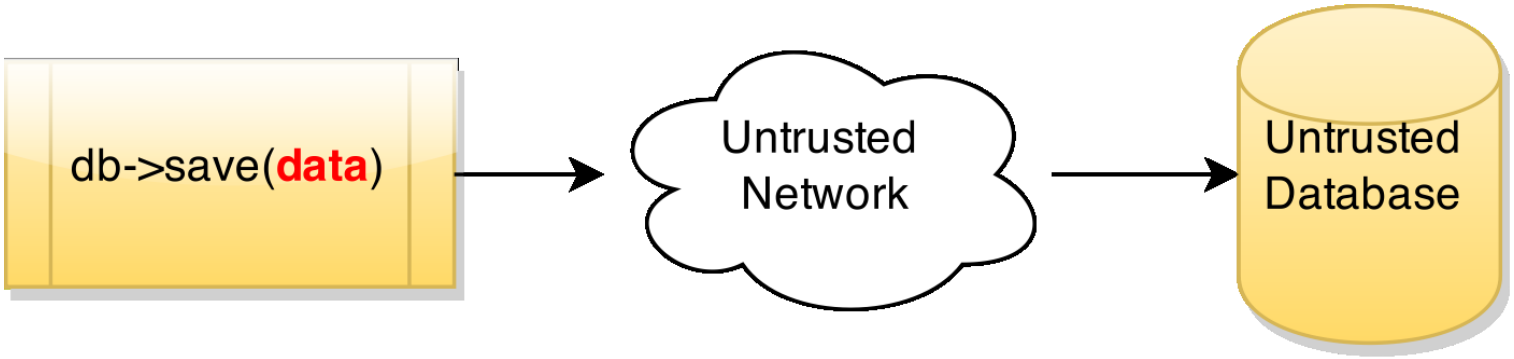
Non-repudiation
(prevent one to deny)

Covered in IERG4130. Check out the readings for revisions.

Copyright. All Rights Reserved.

A Taste of Solving Security (1/2)

- Think about outsourcing data to an untrusted DB at Cloud

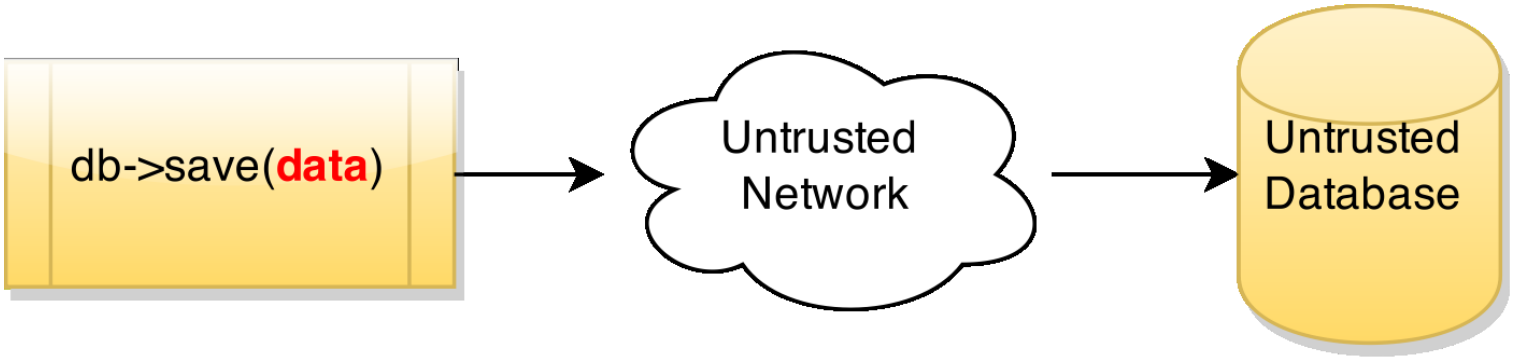


- Intuitively, need encryptions to protect **data**
  - Security goals?
  - At rest v.s. in transit? What technologies? Pros and Cons?
  - Which Security Principles?
  - Threat Modeling? Adversary's capabilities
  - More: Performance, Usability, Searching over encrypted data?

Copyright. All Rights Reserved.

A Taste of Solving Security (1/2)

- Think about outsourcing data to an untrusted DB at Cloud



- Some security solutions:

SSL/TLS encryption or SSH tunneling
- Preventing Sniffing/Man-in-the-middle
? Authenticity on server/client

Encryption at App/Proxy
- Encrypting non-sensitive data before it leaves
? Key Storage/Distribution, leakage after decrypted, etc…

Disk-level encryption
- Recovering raw bits from lost/disposed HDD impossible
? Adversary with (root) access

Copyright. All Rights Reserved.

Reasoning Security (1/2)

INTERNET BANKING LOGON

Identity Authentication

Security Tips

- When using dual password mode to logon to HSBC Internet Banking, HSBC will only ask for your partial second password and never ask for your full second password.
- HSBC Internet Banking will not ask you to provide excessive information such as name, date of birth, HK ID/passport number, credit card number, CVV/CVC code and the expiry date.
- If you find any unusual pop-up screen or the computer response is unusually slow, please do not provide your personal details and/or credit card information and report immediately to our Customer Services Hotline at (852) 2233 3000.

Enter Password:

Forgot your password(s)?

Enter Second Password:

2nd 4th ... 2nd last

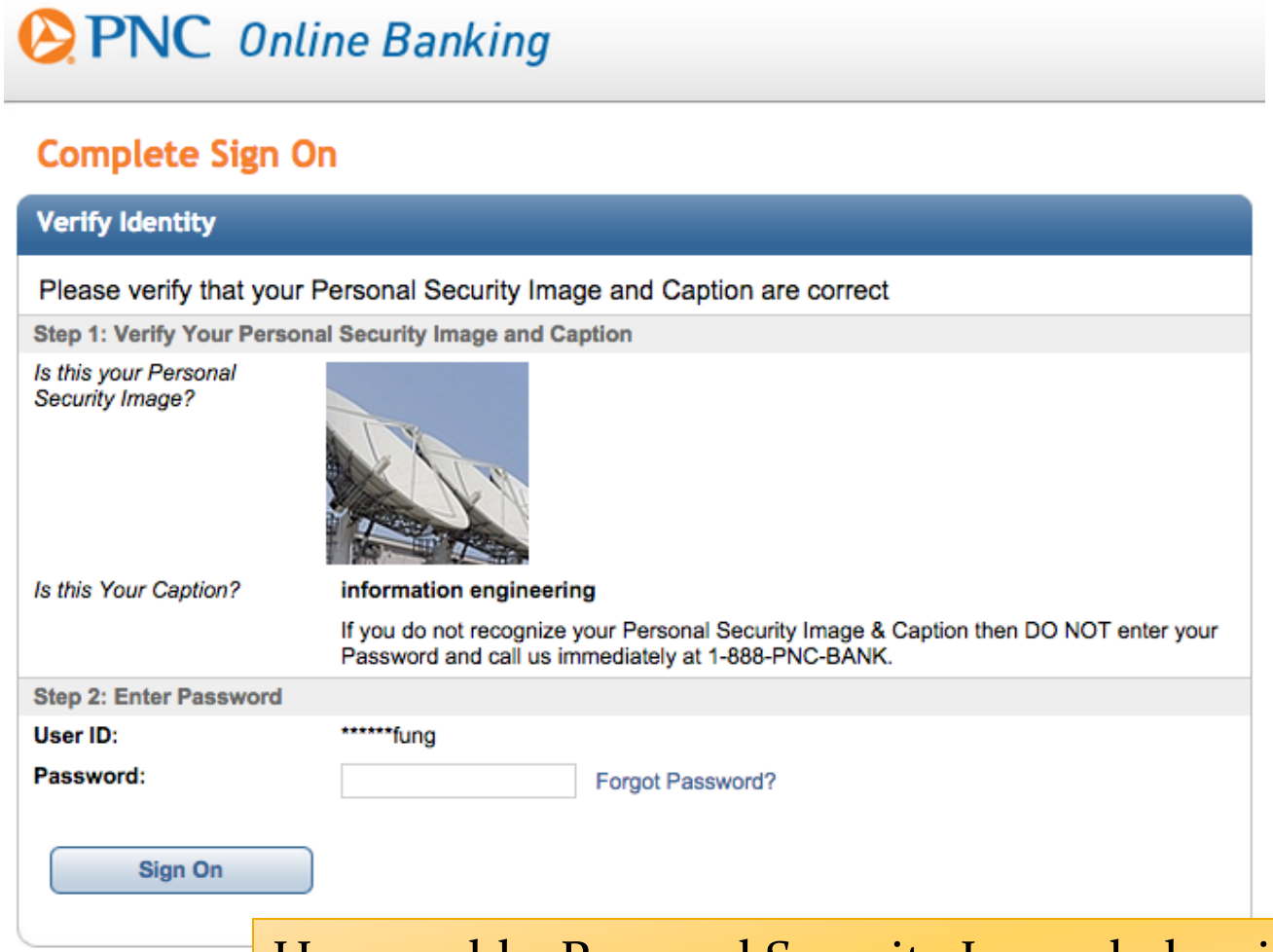
LOGON

Why need the Second Password?

Screen captured from hsbc.com.hk

Copyright. All Rights Reserved.

Reasoning Security (2/2)



How could a Personal Security Image help with Security?

Copyright. All Rights Reserved.

Reflections on Security (1/2)

Systems break; vulnerabilities get reported and fixed endlessly

WHY DOES IT TURN OUT TO BE LIKE THIS?

- No one is paying attention
  - Most products are not designed by anyone with security expertise
  - Security cannot be functionality tested - no amount of beta testing will uncover security flaws - so the flaws end up in end products
  - The buying public has no way to differentiate real security from bad security

- Secure-by-design is important!!

Reference: http://www.schneier.com/crypto-gram-0005.html

Copyright. All Rights Reserved.

Reflections on Security (2/2)

Systems break; vulnerabilities get reported and fixed endlessly

CAN WE COMPLETELY ELIMINATE SECURITY PROBLEMS?

- The only solution is to look for security processes
  - There's no such thing as perfect security
  - People don't understand the risks.
    Products alone cannot solve security problems.
  - There is some amount of risk you can accept,
    and some amount you can't.

- After Secure-by-design,
  it's then about how to avoid risk = likelihood x impact

Reference: http://www.schneier.com/crypto-gram-0005.html

Copyright. All Rights Reserved.

Take-home Readings and Quiz

- Read the following book chapters/pages:
  - 01-reading-SoftwareSecurityPrinciples
  - 01-reading-Daswani07-01SecurityGoals

- Visit the following websites:
  - Learn what the jargons like Javascript, AJAX, HTML 5 are: http://evolutionofweb.appspot.com/

- 1st DEADLINE: Online revision quiz, before Jan 13, 2015

Copyright. All Rights Reserved.

Some Logistics…

- More Coding this year/term…
  - You may like to come with your laptops
  - Try! Try! Try! You won't learn unless you code it yourself
  - Ask questions in-person when you run into troubles

- Tutorial Timeslots TBD

Copyright. All Rights Reserved.