CS 458 Notes

# **Module 6 – Database Security & Privacy**

1. Introduction to DB
   1. Relational DB

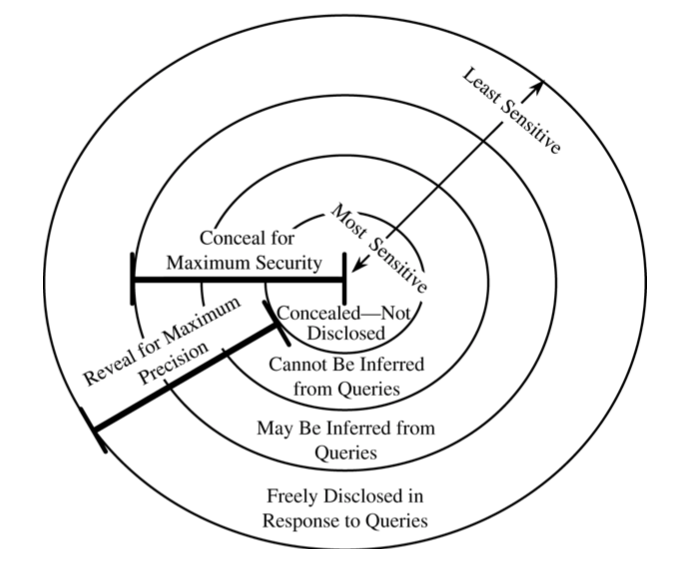
* Records – structured query-able collections of data
* Elements – fields in each record
* Schema – structure set by DB admin
* DBMS – database management system
  + Support for queries and management
  + Based on relational model
* DBMS stores record in one or multiple tables (relation)
  + Tables have name column (attributes) and rows (tuple)
  + Individual table can have relationships between them
  1. Queries
* Query language is SQL
* SELECT Address FROM NAME-ZIP WHERE (Zip = ’43210’) AND (Name = ’ADAMS’)
  + Prints address of family in relation NAME-ZIP whose zip code is 43210 and whose name is Adams
* SELECT Name, Airport FROM NAME-ZIP, ZIP-AIRPORT **WHERE** NAME-ZIP.Zip = ZIP-AIRPORT.Zip
  + Prints each family name and their airport by joining relations NAME-ZIP and ZIP-AIRPORT
* SELECT **COUNT**(Name) FROM NAME-ZIP WHERE City = ’Columbus’
  + Prints number of families in Columbus
  + Can also do other computations, like SUM, MIN, or AVG

1. Security requirements
   1. Database Integrity

* Logical and physical
* Protect against DB corruption
  + Only allow authorized people to perform updates
* Recover from physical problems
  + Power failures, disk crashes
  + Done by periodic backup
  + Keep transaction log to replay
  1. Element integrity
* Ensure correctness/accuracy of DB element
* Access control to limit who can update element
* Element check to validate correctness
  + E.g: DOB Year > 1900
  + Element must be numeric, in range
  + Not more than one employee can be president
  + Help against mistakes by authorised users
  + Enforced by triggers
    - Stored procedure, program will run, check and ensure constraint is met
  + Change log or shadow fields to undo erroneous changes
  + Error detection codes to protect against OS or hard disk problems
    - Hash DB with individual entry
    - Then hash with 2 entries
    - Then 4
    - nLogn size
    1. Two phase update
* For a set of operations, either all of them or none should be performed
  + Integrity violation if only some performed
    - E.g.: Money is withdrawn from an account but not deposit into another account
* First phase:
  + Gather information required for changes
  + Do not perform updates
  + Repeat if problem arises
* Second phase:
  + Make changes permanent
  + Repeat if problem arises
    1. Concurrency control
* Concurrent modifications can lead to integrity violation
  + 2 operation A & B read variable x
  + A update values of X
  + B then write new values of X
  + A’s update get lost
* Need to perform A & B as atomic operations
  1. Referential integrity
* Each table has primary key
  + Minimal set of attributes that uniquely identifies each tuple
  + E.g. User-ID
* A table might also have multiple foreign keys
  + Which are primary key from another table
* Referential integrity ensures that are no dangling foreign keys
  + E.g. each ZIP in Name-Zip table, there is another entry in Zip-airport
  1. Auditability
* Keep an audit of DB access
  + Both read/write
* Access control can be difficult
* Audit log allow retroactively identify users who accessed forbidden data
  + E.g. Police officer looking at someone criminal record
* Maybe combination of accesses resulted in disclosure
* Must decide about granularity of logging
  + Why is select query not logged – trade off with size
  1. Access Control
* More difficult than OS control
* Might have to control access at relation, record or even element level
* Many types of operations not jus read/write
  + Select, insert, update, create, drop
* Relationship between DB objects make it possible to learn sensitive information
  + No need direct accessing it
  + Aka Inference problem
* Efficiency problem in presence of thousands of records
* Each consisting of dozens of elements
* Access control might consider past queries
  + Current query, together with past
    - Could reveal sensitive information
  1. User authentication / availability
* Database might do its own authentication
* Additional check possible
* DB facilitate sharing
  + But availability can suffer if multiple users want to access the same record
  + Block access until other user finishes updating record

1. Data disclosure and inference
   1. Types of disclosure

* Exact data
* Bounds
  + Sensitive value is smaller than H
    - But bigger than L
  + Might iteratively decrease range (binary search)
* Negative result
  + Knowing a person does not have zero felony convictions is sensitive
  + Even if actual number is hidden
* Existence
  + Knowing of existence of some data can be sensitive
* Probable value
  + Sensitive data has value x with probability y
  1. Security vs precision
* Security:
  + Forbid queries that access sensitive data
  + Even if aggregated result is no longer sensitive
* Precision:
  + Aggregated result should reveal as much non-sensitive data as possible



* 1. Data inference
* Derivation of sensitive data from non-sensitive data
* Direct attack
  + Attacker issue query that directly yields sensitive data
    - SELECT SUM(salary) FROM staff WHERE lastname = ’Adams’ OR (sex = ’M’ AND sex = ’F’)
* Indirect attack
  + Infer sensitive data from statistical result
  + Tacker attack
  1. Statistical inference attack
* Sum
  + Leak sensitive data if sum covers only 1 record or attacker can control set of covered records
    - SELECT SUM(salary)
    - SELECT SUM(salary) WHERE lastname != “Adams”
* Count
  + Same as sum
* Mean
  + Sum = count \* mean
* Median
  + Intersecting medians might leak sensitive data
  1. Tracker attack
* Let q() be result of a query
  + E.g. COUNT or SUM

Let S be set of all records

* Using Venn diagrams
  + q(C) = q (C or T) + q (C or not T) – q(S)
    - use RHS for computing q(C) if q(C) matches fewer than k records
  + q(C) = 2 \* q(S) – q(not C or T) – q(not C or not T)
    - RHS to compute q(C) if q(C) matches more than N – k records
* In general
  + Simple logic or linear algebra might allow attacker to convert a forbidden query to multiple allowed queries
  1. Controls for statistical inference attack
* Seems difficult
* Suppression
  + Supress sensitive data from result
* Concealing
  + Answer is close to actual value, but not exactly
  + Modify
* n-item k% rule
  + for the set of records that were included in the result
  + there is a subset of n records that is responsible for > k%, omit n records from result
  + omission itself might leak information or omitted value could be derived with other means
* Combined results
  + Report set or range of possible values
* Random sample
  + Compute result on random sample of DB
  + Need use sample for equivalent queries
* Random data perturbation
  + Add/sub small random error to/from each value
  + Exception is that statistical properties are maintained
* Query analysis
  + Theorical
  + Maintain history of user queries and observe possible inferences
  + Costly, fails for colluding users
  1. Differential privacy
* Response to a query should not depend on an individual
* Being part of the dataset
* Query K has e-differential privacy if for all datasets, D and D’
  + Where D & D’ differ in at most 1 row
  + Probability of k(D) has a particular output is at most eepsilon \* probability of K(D’)
    - Output of (0 <= epsilon <= 1)
* Differential privacy is achieved by adding noise to result of a query
  1. Data aggregation
* Need to have pre-defined data
* Related to data inference
* Build sensitive result from less sensitive inputs
* Aggregation can take place outside DBMS
* Closely related to data mining
  + Information from different database is combined
  + Difficult *to control*

1. Multilevel security database
   1. Intro

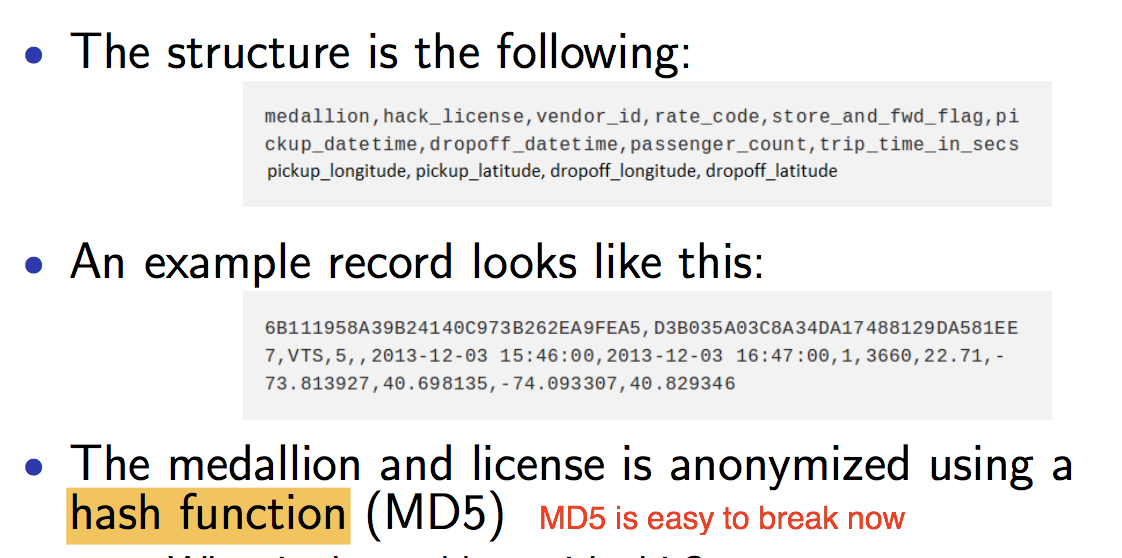
* Support classification /compartmentalisation of information
  + According to its confidentialit**y**
  + **E.g.**  2 sensitivity – sensitive and not sensitive
* At element level if necessary
  + Salary might be sensitive only for some employee
  + Other information in employee record might not be sensitive
* In an MLS
  + Each object has a sensitivity classification and maybe a set of compartments
    - Object can be element of aggregate, column or row
  1. \*-Property
* the property of no read up, no write down in an MLS DB
  + difficult
  + user doing a write up, when user cannot read the data having higher sensitivity (Blind writes)
  + write down need a sanitation mechanism
  + trusted processes that can do anything
* DBMS must read and write access at all level
  + Answer user queries
  + Perform backup
  + Optimizes database
  1. Confidentiality
* depending on user’s clearance
  + might get different answer for a query
  + less precision for low-clearance users
* Existence of a record could be confidential
* Keeping existence hidden
* Lead to having multiple records with the same primary key but different sensitivity (poly-instantiation)
  + E.g.
  + Admin notice there is no record for Bob and create one
  + However, Bob hill is a secret agent – there is already a record, which admin can’t see
  + DBMS must allow admin request; else admin would get suspicious
  1. Partitioning
* Having separate DB for each classificat*ion* level
* Simple
* Mightlead to data stored redundantly in multiple database
* Doesn’t address problem of a high-level user needing access to low level data combined with high level data
  1. Encryption
* Separate data by encryption with a key unique to its level
* Careful with scheme used
  + Encryption the same value in different records with the same key should lead to different ciphertext
* Processing of a query become expensive, many records might have to de decrypted.
  1. Integrity lock
* Provide both integrity and access control
* Each data item consists of
  + Actual data item
  + Integrity level (maybe concealed)
  + Cryptographic signature covering the above item attribute name and its record number.
* Signature protects against attacks on the above fields
  + Such as attack trying to modify the sensitively label
  + Attack trying to move/copy item in DB
* Does not protect against replay attacks
* Any untrusted DB can be used to store data items and their integrity locks
  + Can consume lots of space
* Trusted procedure handles access control and manages integrity locks
  + Update integrity level to enforce \*-property or re-computes signature after a write access = Expensive
* Must encrypt items and lock if there is other way to get access to data

1. Designs of secure databases
   1. Trusted front end

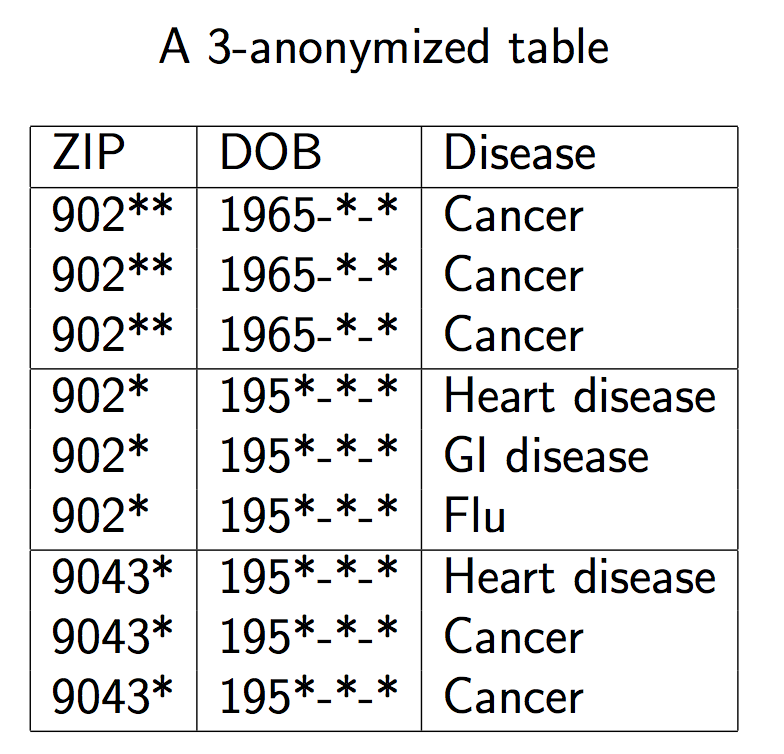
* Front end authenticates a user and forwards user query to old style DBMS
* Front end get result from DBMS and remove data items that user is not allowed to see
* Allows the use of existing DBMS and database
* Inefficient if DBMS return lots of items
  + Most of them are being dropped at the front end
  1. Commutative filters
* Front end re-writes user query according to a user’s classification.
  + Remove attributes that user is not allowed to se
  + Add constraint expressing user classification
* Benefits from DBMS superior query processing capabilities and discards forbidden data items early on
* Front end might still have to do some post processing
  1. Distributed/federated databases
* Based on partitioning
* Front end forward user query only to databases
  + That user can access based on classification
* Front end might have to combine results from multiple databases
  + Complex process, front end essentially and become a DBMS
* Don’t scale a lot of classification labels
  1. Views
* Supported by many DBMS
* View = logical DB that represent a subset of some other DB
  + CREATE VIEW foo AS SELECT \* FROM bar WHERE …
* Element in view can correspond to an element in underlying DB
  + Or be a combination of multiple elements
    - E.g. their sum
* View can be used for access control
  + User’s view of a DB consists of only the data that user can access
  + Hide attribute / row unless user can access at least 1 element, set to UNDEFINED any elements that user can’t access
  1. Truman vs non-Truman semantics
* Truman Semantics:
  + DBMS pretends data the user can access is all the data there is
  + All queries will succeed
    - Even if they return incorrect result
* Non-Truman Semantics:
  + DBMS can reject queries that ask for data the user is not allowed to access
    - Any queries that succeed will produce precise answers
    - Some queries will fail

1. Data mining and data release
   1. Intro

* MLDB is not a commercial success
  + Mainly military clients
  + Finding possible inference is NP-complete
* Combination of sensitive information, stored in multiple (huge DB)
  + Raise similar concerns and gotten lot of attention recently
* Data miners actively gather additional data from 3th parties
  + Unlike single entity – have control of data
* Data mining tries to automatically find interesting patterns in data
  + Using plethora of technologies
    - Statistics, machine learning, pattern recognition
* Useful for security purpose
  + Learn information from logs
  1. Security problems of Data Mining
     1. Confidentiality
* Derivation of sensitive information
* Can be revealed from DM
* E.g.: supermarket can use loyalty card to learn who buys what kinds of product and sell this data, maybe to manufacturers’ competitors
  + 1. Integrity
* Mistakes in data
  + Can lead to wrong conclusion
  + Which can negatively impact individual
    - E.g. receiving irrelevant mail to being denied flying
  + Privacy calls for right of individuals to correct mistakes in stored data
  + False positives
    - Can also be false negative
    - Don’t blindly trust data mining applications
    1. Availability
* Incompatibility of different DB
* Mined DB are often created by different organisations
  + Different primary key, attribute semantics
* Make combination of DB difficult
* Must distinguish between inability to combine data and inability to find correlation.
  1. Privacy and Data mining
* Might reveal sensitive information about individuals
  + Based on aggregation and inference techniques
* Avoiding privacy violation is active research
* Data collection and mining is done by private companies
* And by governments
  + Tend to be secretive, no clear procedures
  1. Privacy preserving data release
* Anonymise data records before releasing them
  + Strip names, address, phone numbers
* E.g. Using MD5 - easy to break now



* 1. K-anonymity
* Ensure that each released record, there are at least k-1 other released records which
  + Cannot be distinguished (k >=2)
* E.g.
  + Health records, if release a record only of there are k -1 other records that have the same ZIP code, gender, DOB
    - Assumption: only one records for each individual
  + Due to 87% number
    - It won’t return many records, need some pre-processing of records
    - Remove ZIP, gender, DOB
      * Reduce granularity of ZIP code of DOB
      * Domain generalisation
    1. Limitations of k-anonymity



* 1. L diversity and t-closeness
* Homogeneity attack
  + If know bob, (902\*\*,1965-\*-\*)
  + Then bob has cancer
* Background knowledge attack
  + If you know Dave is in table
  + And risk for heart disease is low
    - Can conclude Dave has cancer
* L-diversity property
  + For any quasi-identifier, there should be at least l “well-represented” values of sensitive fields
* Possibly still not good enough: t –closeness
  + Ensure distributions of the values for any quasi-identifier are within t of the distribution for the whole table
  1. Value swapping
* Data perturbation is based on swapping values of some data fields for a subset of the released records
  + Swap addresses in subset of records
* Any linking done on the released records can no longer considered to be true
* Trade-off between privacy and accuracy
* Value swapping make strong correlations less strong and weak correlations might go away entirely
  1. Adding noise
* Data perturbation based on adding small positive or negative error to each error
* Given distribution of data after perturbation and distribution of added errors, distribution of underlying data can be determined
  + Not its actual values
* Protects privacy without sacrificing accuracy
  1. Sampling / synthetic data
* Release only a subset of respondents’ data
  + 1% with the following:
  + Geographic coarsening
    - Restrict geographic identifiers to regions containing at least a certain population
    - E.g. 100k people
  + Top/bottom-coding:
    - For example, if there are sufficiently few respondents over age 90, top coding would replace all ages >= 90 with the value 90
* Build a distribution model based on gathered data and use the model to generate synthetic data with similar characteristics to original data
  + Release one or a few sets of synthetic data

# **Module 7 – Non-technical Aspect**

1. Administering security
   1. Security planning

* Many users do not appreciate the security and privacy risks in using computers
* Security plan
  + A document put together by an organisation
  + Explains what the security goals are
  + How they are met
  + How they stay met
  1. Contents of security plan
* Both a description of current state of security of an organisation
* As well as plan for improvement
* Has 7 parts
  + 1. Policy
* High-level statement of purpose and intent
* Should specify
* Goal
  + Relative importance of confidentiality, availability and integrity
  + Which has higher priority: secure data or serving customer
* Responsibility
  + Whose job is getting security right
* Commitment
  + Institutionally, who provides security support for staff?
  + Where does security fit into the organisation chart
    1. Current State
* Contain risk analysis describing status of system
  + Assets and controls implemented
  + What might go wrong?
  + What vulnerabilities are currently exposed?
* List the limits of security responsibility
  + Who is responsible for security of the internet uplink router to company ISP?
* How is people privacy affected?
  + Perform Privacy Impact Assessment (PIA)
    1. Requirements
* What needs does the organisation have
  + Who is allowed and what do what?
  + What log should be kept
  + Do you need to be able to measure the ongoing effectiveness of security controls?
* Not anything to do with mechanism
  + Does not say anything about how to accomplish the listed goals
  + Should be technology neutral
  + Employees should be allowed to access their mail while travelling
    - Should not say any of the words
      * VPN, ssh, TLS, IPsec
    1. Recommended controls
* List of mechanisms to control vulnerabilities identified in “current state”
* Satisfy needs in “requirement section”
* Taking account, the priorities in the “policy section”
  + 1. Accountability
* Who is accountable if security controls aren’t implemented
  + or implemented properly or fail?
* Probably different people will account for different pieces of the plan
  + 1. Timetable
* Any reasonably sized security plan will be too big to implement at once
  + Obtaining new hardware / software
  + Configuring / installing it
  + Training users
* Timetable section of a security plan lists how and when elements of the plan will be performed
  + What order, noting dependencies
* Include milestone to track progress along the way
  + 1. Continuing attention
* State of organisation is not static
* State of world is also not static
* There will be new vulnerabilities
* Existing controls will become ineffectual
  + Should list a process for periodic review and updating of the plan
  1. Who writes security plan?
* Security planning team should be represented from several different group
  + Upper management / CTO / CIO
  + IT (hardware group, sysadmin), data entry personnel
  + Systems and application programmers, DB admins
  + Physical security personnel.
  1. Business continuity plans
* Another kind of security plan
* Focus on availability
* Explains what will organisation do if it encounters a situation that is:
  + 1. Catastrophic failures
* Large part of computing capability is unavailable
  + Failures include
    - Fire / earthquakes – destroy data
    - Flood = prevent operations staff being able to reach office
    - Pandemic outbreak of virus – keep staff home sick

* 1. Advanced Planning
* Actual plan that include
  + Who is in charge when a catastrophe occurs
    - Person will be one to declare when emergency is over and things can get back to normal
  + What needs to be done
    - To deal with keeping business going
  + Who will do it
* Require doing:
  + Acquire redundant equipment
  + Arrange for regular data backup
  + Stockpile supplies
  + Train employee to know how to react
  1. Incident response plans
* Need to consider
  + Legal issues
    - Incident has legal ramifications
    - Under what event should law enforcement get involved
  + Preserving evidence
    - How can you recover from the incident while maintaining as much forensic evidence as possible?
  + Records
    - Keep careful track of everything you do once you notice the breach
  + Public relations
    - Speak with one voice
    1. After the incident
* Hold a review to ask
* Is any security control action to be taken?
  + Any breach occur, is it patched?
* Did the incident response plan work?
  + Did everyone know whom to notify?
  + Was response time fast enough.
  1. Risks
* Potential problem that a system or its users may experience
* 2 important characteristics
  + Probability:
    - Between 0 and 1 that the risk will occur
    - Risk it will turn into a problem
  + Impact
    - If risk occur, what harm will happen?
    - Cost, damage lost, etc
* Risk exposure = probability \* impact
  + This probability and impact risk change over time
  1. Risk Analysis
* Impossible the eliminate risk
* No = all system is not secure
* Not just specific to security and privacy issues
  + 1. Identify assets
* 6 main assets to protect
  + hardware, software, data
  + people – skills to run system, network
  + documentation – on hardware and software, but also the security plan
  + supplies –supporting role such as printer, forms, etc.
    1. Determine vulnerabilities
* Apply best knowledge
* Think creative – like an attacker
* Come up with as many attacks on your own system (technical and non-technical)
  + Confidentiality, integrity and availability
  + Privacy
    1. Estimate like-hood of exploitation
* Hardest step
  + Difficult to estimate the probability of each risk
* Use frequency analysis
  + How often has this risk been a problem in a year?
* Consider existing controls and their own probabilities of failure
  + 1. Compute risk exposure
* Identify impact of risk, also a tricky step
* Good examples
  + Legal obligations to converse confidentiality or integrity.
  + Penalties for failing to provide a service.
  + Cost of delays or outsource data processing if system is not available
    1. Survey applicable controls
* Each risk = think of different ways to control vulnerability
  + Technical and non-technical mean
* Classify each control to how well it protects against each vulnerability
  + Note a control that protects against one
  + Watch out for interactions among different controls
    1. Project savings due to control
* Expected cost of not controlling the risk is just risk exposure
* Each control, find the cost of control (direct cost)
  + E.g. buying network monitor equipment, training, etc.
* Plus, the exposure of the controlled risk
* IMPORTANT
  + Savings = risk exposure – cost of control
  + Go for positive
  1. Physical Security
* Firewall in the world won’t help you defend against an attacker who physically steals your laptop off your desk
* Need to protect the physical machines
  + as well software and data
  1. Physical Threats
* 2 major class of physical threats
* Nature
  + Fire, Flood, Blackouts
* Human
  + Vandals thieves targeted attackers
  1. Physical controls against humans
* Additional measures to protect against human
* Need to not only recover the loss
  + But also deal with release of potentially sensitive data
    1. Vandals
* Not just after the data
  + 1. Thieves
* Most thefts are after software, hardware and data
  + 1. Targeted attackers
  1. Protecting offline data
* Good sense of how protect data on active machine hooked up to network
* Harder for a network-based attacker to get data
* Easy for thief and insider
* To safely dispose of data
  + Paper – shredder
  + Optical media – overwrite with a standard
  1. Tiger team
* Group of people that try to crack and test your security team
  + Security professionals
* Hire them to try break your system
  + Will report to you what is wrong
  1. Legal Protections
* How to defend against a threat
  + Prevent: block attack
  + Deter: make attack harder or more expensive
  + Deflect: make less attractive to attacker
  + Detect: notice that attack is occurring
  + Recover: mitigate the effects of attack
* Use technological defences, we can also use legal defences
  + Find an attorney that pursue.

1. Legal and ethical issues
   1. Overview of IP

* IP = intellectual property
* Differs in important ways:
  + Its non-depletable
  + It is replicable
  + It has minimal marginal cost
* Laws for IP differ from the laws for real property
  + More complicated
* 4 kinds of IPs
  + they cover different kinds of intangibles
  + convey different rights
  + have different durations
  + have different registration requirements
    1. Trade Secrets
* Simplest kind of IP
* Want to protect some secret information
  + E.g. Formula for Coca-Cola
* Just don’t tell anyone, call it a trade secret
  + But need to tell someone about this secret, or its not useful.
  + Get legal protection if that person pass on
    - 1. Reverse engineering
* Process of taking a finished product, taking it **apart**
  + To find out how it works
  + If successful = lost trade secret
  + E.g. RC4 used to be a trade secret, but reversed in 1994
* General rule for trade secret: must be secret
  + 1. Trademarks
* Protect the names, brands, logo
  + E.g. RC4
* To get one, make a legal filing showing that you are using the name in commerce
  + Let you sue others who use that name in confusing manner
* Domain names are often protected under trademark law
  + 1. Patents
* Applies to inventions
  + Novel
  + Useful
  + Non-obvious
* The bargain is that
  + Tell others how it works
  + In exchange for monopoly for 20 years
* Most difficult form of IP to obtain
  + - 1. Cryptography patents
* E.g. of cryptographic algorithms that are patented
  + Diffie-Hellman (expired in 1997)
  + RSA
  + ECG
* Since 2000, good unpatented example of each type of crypto
  + 1. Copyrights
* Most well-known kind of IP
* Protects expression of ideas
  + In tangible medium
* No filing requirement
  + But get additional benefits if you do file
* Last a limited time
  + 70 years
* copyright holder has monopoly rights over certain use of the work
  + primarily, making copies
    - 1. Legal copying
* Rights granted to copyright holder aren’t absolute
  + Anyone can copy a work without permission is some manner
* In USA, the manners are broad but loosely defined
  1. Fair use in US
* These exceptions are called fair use
  + For purposes such as criticism, comment, news, reporting, testing, or research
  + Four test
    - Purpose and character of use
      * Include if use is of a commercial nature or is for non-profit educational purposes
    - Nature of the copyrighted work
    - Amount and substantiality of portion used in relation to copyrighted work as a whole
    - Effect of the use upon potential market for or value of the copyrighted work
  1. Fair dealing in Canada
* Exception to copyright law is defined more narrowly
* Applies to private study, research, criticism, review, news, education…
* Similar set of tests as in the US
* Time shifting, format shifting, backup copies are also legal
  + 1. Private copying of sound recordings
* Does not cover private copying of sound recording
* However,
  + Allowed to copy a sound recording onto an audio recording medium for private use
  + In exchange, pay a levy of on blank audio recording media like CDs
* Makes downloading of songs over a p2p network legal in Canada
  + Not for uploading.
  1. Para-copyright
* 1998, US passed Digital Millennium Copyright Act (DMCA)
  + didn’t make any additional acts of making copies illegal
  + made illegal circumvention of a technology copy protection mechanism that might be in place
* Problems
  + Applies even when copy protection mechanism is broken to make a “fair use” copy
  + Also, made illegal the manufacture, selling or “traffic” of devices that might help you circumvent such methods
  1. Computer crime
* Laws about unauthorized access of computers, networks, or services differ from those about physical trespass
  + 1. Rule of evidence
* Problem when prosecuting computer crime is producing the evidence admissible in court
* Should log files of the machine that was broken into be admissible?
* How should you preserve electronic evidence from time of intrusion to time of trial?
  + Computer forensics replace forensics
    1. Cybercrime treaty
* Cybercrime are often intentional
* Rule of evidence, police powers, in one country don’t carry over to another
* Council of Europe cybercrime treaty states
  + Member countries should pass law making it easier for law enforcement to access telecommunication traffic
    - Voice, data, and internet
    1. Bill C-13 “Cyberbullying law”
* Passed in 2014
* Any public officer can demand that any computer data in a person control not be deleted
* Lower standard for seizing of computer data
* Transmission data, and tracking data to “reasonable grounds”
* Provide immunity to ISPs that “voluntarily” hand over customer data to government.
  1. Dangers of building back doors
* Ability for anyone to avoid attention and intercept communication is designing a weakness
* Dangers have been known since Clipper Chip
  + Before turn of the century
* Remember that technology can’t tell whether any oversight requirements are met
* If there is a back door built into the system
  + Bad guys will find a way to use it
    - Technical means (hacking)
    - Social engineering
    1. Case Study: Greek Wiretapping scandal
* Greek government had copies of their calls routed to a bank of throwaway phones
  + People used CALEA interception capabilities built into phone switches to comply with US Law
  1. Harmonisation
* Only build one thing, not one thing for each jurisdiction
* Often tend to end up with the biggest “back door” required anywhere
  + 1. Regulation of investigatory power
* Example of big back door required
  + In UK, Part 3 of RIP act went into effect on Oct 2007
  + Served notice to
    - Encrypted data
    - Hand over decryption keys
    - Don’t tell this has happened
  + Propose update in 2015
    - Protect MP’s communication more than ordinary citizens
  1. Redress for software failures
* Flaws are discovered in most products you buy
  + Get new one
  + Repaired the flaw
  + Get refund
* Embedded software does not have this problem
  + Flaws in embedded software are fixed by manufacturers
  1. Reporting flaws and failures
* Vendors prefer you report to them and not anyone else
* Some vendor back up this preference by suing you if you publicly disclose a security flaw in products
  + 1. Full disclosure
* Some prefer full disclosure
  + When you find a problem
    - Post it to a full disclosure mailing list of securities professional
  + Reasoning
    - Found the problem (you) = bad guys should have found it too
    - And may be actively exploiting it
    - Need to plug the hole ASAP
      * Until an official fix
    1. Responsible disclosure
* If you find a flaw
  + Report to vendor
  + Do not disclose for 30 days
* If vendor have not announced the flaw
  + They credit you
  1. Code of professional ethics
* As computer security, professional
  + Expected to uphold certain ethical standards
  + Ethics != law
* Member of one or more professional societies
  + Have their codes of professional ethics
  + E.g. CIPS
    - Canadian Information Processing Society
    - High level bullets from CIPS code
      * Protect public interest and maintain integrity
      * Demonstrate competence and quality of service
      * Maintain confidential information and privacy
      * Avoid conflicts of interest
      * Uphold responsibility to IT profession