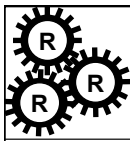


Information Security

An Introduction

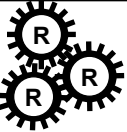
CSE870: Advanced Software Engineering: Security Intro



Acknowledgments

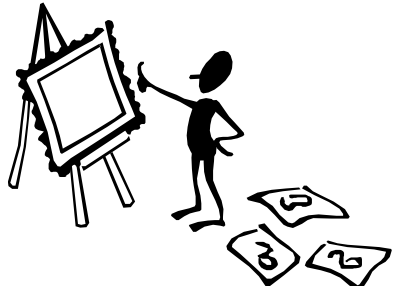
- Annie Anton
- Charles Pfleeger
- E. Spafford

CSE870: Advanced Software Engineering: Security Intro



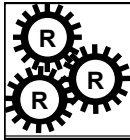
Outline

- Terminology
- Brief Introduction
- Security Planning
- Creating a Security Policy
- Threats, Attacks & Services
- Internet Privacy Policies



A stick figure stands next to a flipchart on an easel, pointing at it. Three pieces of paper with numbers 1, 2, and 3 are scattered on the floor nearby.

CSE870: Advanced Software Engineering: Security Intro



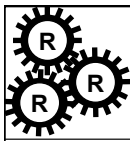
Terminology

- “A computer is **secure** if you can depend on it and its software to behave as you expect (intent).”
- ‘**Trust** describes our level of confidence that a computer system will behave as expected.’ (intended)



[Garfinkel & Spafford, Kasten]

CSE870: Advanced Software Engineering: Security Intro



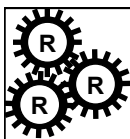
What is secure?

- Does not disclose information
- Does not allow unauthorized access
- Does not allow unauthorized change
- Maintains QoS despite input and load
- Preserves audit, authenticity, control
- No surprises!



[Spafford]

CSE870: Advanced Software Engineering: Security Intro

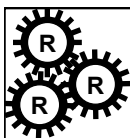


Why Worry?

- Information has value
 - when combined
 - when altered
 - when disclosed
- Resource use has value
 - unauthorized use
 - denial of service
- Damage to reputation
 - damage to your personal reputation
 - damage to your group
 - damage to your company
- Your system is not alone
 - other machines on the network
 - shared resources and files
 - indirect liability

[Spafford]

CSE870: Advanced Software Engineering: Security Intro

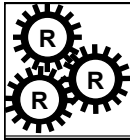


Three Common Failures

- Organization has no formal policy. Thus, personnel cannot *consistently* make necessary decisions.
- Organization has no reasonable response plans for violations, incidents, and disasters.
- Plans don't work when needed because they haven't been regularly tested, updated, and rehearsed. (E.g., failure of operational security)

[Spafford]

CSE870: Advanced Software Engineering: Security Intro

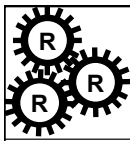


The Challenge

- “Without assurance that our systems will stay secure, we endanger our economies, our privacy, our personal safety and privacy, and our social institutions.” [Spafford]

[Spafford]

CSE870: Advanced Software Engineering: Security Intro



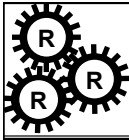
How do we get there?

- Understand the needs of the users
 - Narrow focus better than broad
- Understand basic tenets of security
 - Scarcity/rareness of programs and experts
- Capture requirements for design and validation
- Design with care using good tools and methods
- Validate & Verify



[Spafford]

CSE870: Advanced Software Engineering: Security Intro

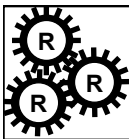


Understanding Security

- Good security means
 - Limiting what happens
 - Limiting who can make it happen
 - Limiting how it happens
 - Limiting who can change the system
- Users don't tolerate limits unless there is a paradigm shift
 - E.g.,
 - Mainframes to PCs/desktops
 - to laptops
 - to handhelds computers
 - to cellphones/blackberrys

[Spafford]

CSE870: Advanced Software Engineering: Security Intro



Psychological Acceptability

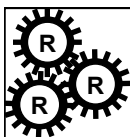
- Easy to use
 - Should be as easy to use as to not use
- False alarms should be avoided
- Frequent changes and updates are bad
- Should not require great expertise to get correct

...Doesn't match user population



[Spafford]

CSE870: Advanced Software Engineering: Security Intro



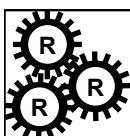
Patches

- Fixes for flaws that require an expert to install are not a good fix.
- Fixes that break something else are not a good fix.
- Frequent fixes may be ignored.
- Goal should be design, not patch



[Spafford]

CSE870: Advanced Software Engineering: Security Intro

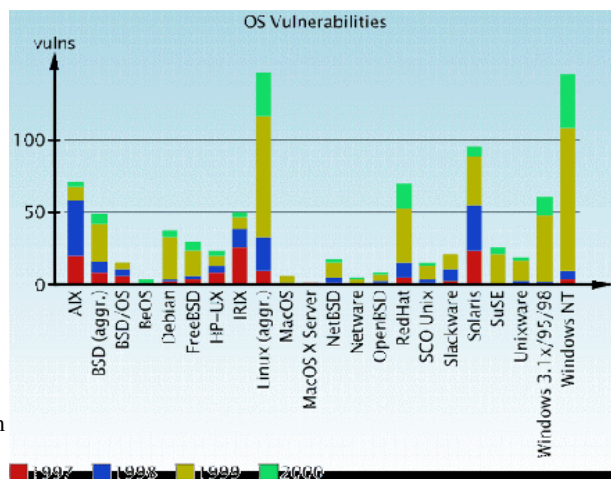


Source of Problems

About 30% are buffer overflows or unchecked data

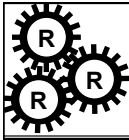
Over 90% are coding/design flaws.

Source:
Securityfocus.com



[Spafford]

CSE870: Advanced Software Engineering: Security Intro



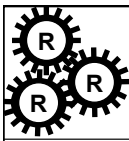
Quality as a Market Problem

- Good software engineers and security designers are scarce
- Productivity of coders varies:
 - Top 10% are at least 10x more productive than average coder.
 - Organizations should invest in raising skill level.
- That takes time and money, so there is a disincentive to improving quality



[Spafford]

CSE870: Advanced Software Engineering: Security Intro

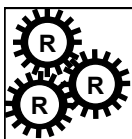


What can we do?

- Understand that there is no “average user”
- Understand balance between features and security
- Employ better testing
- Manage complexity and change
- Build in security from the start
- Understand policy differences.

[Spafford]

CSE870: Advanced Software Engineering: Security Intro

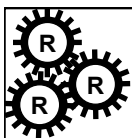


Security Planning

- Security needs planning
- Risk assessment
- Cost-benefit analysis
- Creating policies to reflect your needs
- Implementation
- Audit and incident response

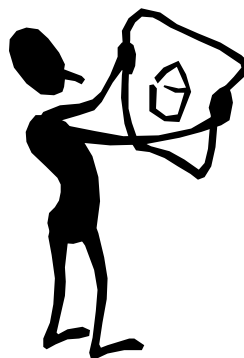


[Garfinkel & Spafford] CSE870: Advanced Software Engineering: Security Intro

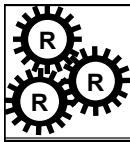


Planning Your Security Needs

- Confidentiality
- Data Integrity
- Availability
- Consistency
- Control
- Audit



[Garfinkel & Spafford] CSE870: Advanced Software Engineering: Security Intro

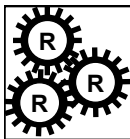


Critical Concerns for Various Industries?

- Banking environment?
- National defense-related system that processes classified information?
- University?
- E-Commerce?



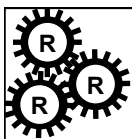
CSE870: Advanced Software Engineering: Security Intro



Risk Assessment

- Three questions to answer:
 - *What* am I trying to protect?
 - What do I need to *protect against*?
 - *How much* time, effort and money am I willing to expend to obtain adequate protection?
- Three key steps:
 - Identify assets
 - Identify threats
 - Calculate risks

[Garfinkel & Spafford] CSE870: Advanced Software Engineering: Security Intro

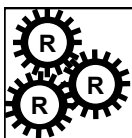


Risk Assessment Step 1: Identify Assets

- Tangibles
 - Computers, disk drives, proprietary data, backups and archives, manuals, printouts, commercial software distribution media, communications equipment & wiring, personnel records, audit records
- Intangibles
 - Safety & health of personnel, privacy of users, personnel passwords, public image & reputation, customer/client goodwill, processing availability, configuration information

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

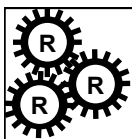


Risk Assessment Step 2: Identify Threats

- Illness of key people
- Loss of key personnel
- Loss of phone/network services
- Loss of utilities (phone water, electricity) for a short or prolonged time
- Lightening or flood
- Theft of disks, tapes, key person's laptop or home computer
- Introduction of a virus
- Computer vendor bankruptcy
- Bugs in software
- Subverted employees or 3rd party personnel
- Labor unrest
- Political terrorism
- Random "hackers"

[Garfinkel & Spafford]

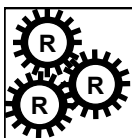
CSE870: Advanced Software Engineering: Security Intro



Broad Categories of Threats

- Interruption
- Interception
- Modification
- Fabrication

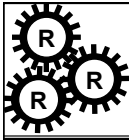
CSE870: Advanced Software Engineering: Security Intro



Interruption

- Asset becomes lost, unavailable, unusable
- Ex:
 - Malicious destruction of HW device
 - Erasure of program or data
 - Malfunction of OS (e.g., cannot find a file)

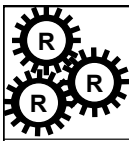
CSE870: Advanced Software Engineering: Security Intro



Interception

- Unauthorized party gained access to an asset
 - Outside party: person, program, system
- Ex:
 - Illicit copying of program/data files
 - Wiretapping to obtain data in network
- Loss may or may not be detected (i.e., leave no traces)

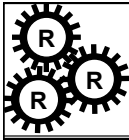
CSE870: Advanced Software Engineering: Security Intro



Modification

- Unauthorized access tampers with asset
- Ex:
 - Change values in database
 - Add computation to a program
 - Modify data during transmission
 - Modify hardware
- Detection may be difficult

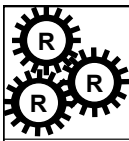
CSE870: Advanced Software Engineering: Security Intro



More Modification

- Trojan horse:
 - Overtly does one task, covertly does something else
- Virus:
 - example of trojan horse;
 - Spread infection from one computer to next
- Trapdoor: program has secret entry point
- Information leaks: (in program)
 - Make info accessible to unintended people/ programs

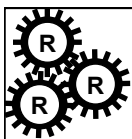
CSE870: Advanced Software Engineering: Security Intro



Fabrication

- Unauthorized party produce/generate counterfeit objects on computing system
- Ex:
 - Insert spurious transactions to a network
 - Add records to an existing database
- Detection and authentication are problems

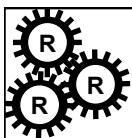
CSE870: Advanced Software Engineering: Security Intro



Risk Assessment Step 3: Quantify Threats

- Estimate likelihood of each threat occurring
- If an event happens on a regular basis, you can estimate based on your records
- Other sources:
 - Power company: official estimate of likelihood for power outage during coming year
 - Insurance company: actuarial data on probabilities of death of key personnel based on age & health
 - Etc.
- Example: Earthquake once in 100 years (1% of your list) vs. discovery of 3 serious bugs in sendmail during next year (300%)

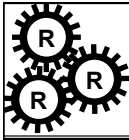
[Garfinkel & Spafford] CSE870: Advanced Software Engineering: Security Intro



Security Goals

- Computer security objective: Maintain 3 characteristics
- Confidentiality:
 - Assets are accessible only by authorized parties
 - Read-type access: read, view, print, existence
 - AKA: secrecy and privacy
- Integrity:
 - Modified only by authorized parties in authorized ways
 - Modification: write, change, change status, delete, create
- Availability:
 - Assets accessible to authorized parties
 - AKA: denial of service

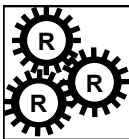
CSE870: Advanced Software Engineering: Security Intro



Vulnerabilities

- Reverse the 3 security objectives
- Major assets:
 - Hardware
 - Software
 - Data
- Their interconnection is also an asset

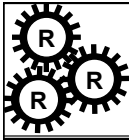
CSE870: Advanced Software Engineering: Security Intro



Threats to Hardware

- Physical device is visible – easy target
- “Involuntary computer-slaughter”
 - Accidental acts not intended to do harm
 - Ex: natural acts, human-oriented accidents (spilling of food/drink), dust, smoke, physical abuse
- “Voluntary computer slaughter” – machinicide:
 - Shoot or stab machines, bombs/fires/collisions, short out circuit boards (pens, knives, etc.), stolen
 - Theft and destruction major mechanisms for attack

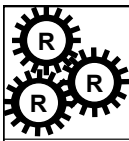
CSE870: Advanced Software Engineering: Security Intro



Threats to Software

- Computing Equipment worthless without software
- Deletion: easy to delete
 - Motivate need for configuration management
- Modification:
 - **Trojan horse**: overtly does one task, covertly does something else
 - **Virus**: type of Trojan horse; spread infection from one computer to another
 - **Trapdoor**: program has secret entry point
 - **Information leaks**: makes information accessible to unintended people/programs
- Theft: unauthorized copying of SW

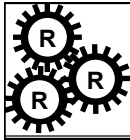
CSE870: Advanced Software Engineering: Security Intro



Threats to Data

- Printed data can be readily interpreted by general public
- Data attack more widespread than either HW or SW
- Data has cost:
 - Confidential data has value to competitors
 - Incorrectly modified data lead to loss of human life
 - Poor security can lead to financial liability
 - Personal data is leaked to public
- Data may have short life:
 - High value: (e.g., economic data and effect on stock market)

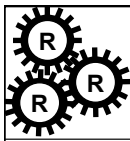
CSE870: Advanced Software Engineering: Security Intro



Threats to Data

- **Principle of Adequate Protection:**
 - Computer items must be protected only until they lose their value. They must be protected to a degree consistent with their value. [C. Pfieger 2000]

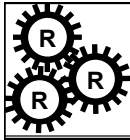
CSE870: Advanced Software Engineering: Security Intro



Threats to Data

- **Confidentiality:**
 - Preventing unauthorized disclosure
 - **Problems:** wiretapping, bugs in output devices, monitoring electromagnetic radiation, bribing key employees. (Data is often human readable.)
- **Integrity:**
 - Preventing unauthorized modification
 - **Problems:** malicious programs, erroneous file system utilities or flawed communication systems
 - Salami attack
 - Example?: Office Space
- **Availability:**
 - Preventing denial of authorized access
 - **Problems:** denial of service attacks, (flood server)

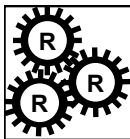
CSE870: Advanced Software Engineering: Security Intro



Other threatened entities

- Storage media
 - Need backups of data and physical protection of backups
- Networks:
 - Involve HW, SW, and data
- Access: access to computing equipment (unauthorized use of processing cycles, network, etc.)
- Key People
 - Crucial weak points

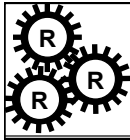
CSE870: Advanced Software Engineering: Security Intro



People Involved

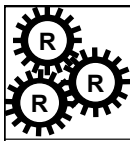
- **Amateurs:**
 - Observed flaw in security
 - Normal/regular employees
 - Exploit system (innocently?)
- **Crackers:**
 - Students who attempt to access facilities
 - “victimless” crime?
 - Serious offense: caused millions of dollars in damage
- **Career Criminals:**
 - Start as computer professionals who engage in computer crime and have good payoffs
 - Electronic spies
 - Response: lack of criminal prosecution trend

CSE870: Advanced Software Engineering: Security Intro



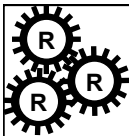
Methods of Defense

- Controls:
 - **Encryption**: transform data to unintelligible format to outside observers.
 - **SW controls**:
 - Internal program controls: parts of program enforce security restrictions (e.g., access limits)
 - Operating system controls: limitations enforced by OS to protect users from each other
 - Development controls: quality standards for design, code, test, and maintenance.
 - **May use HW components, encryption, or info collection**.
 - Affect users directly, so is usually first solution considered
 - Care must be taken in design because it affects the way systems are used
 - Balance between ease of use and effectiveness.



Methods of Defense (cont'd)

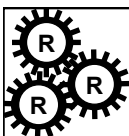
- **Hardware Controls**:
 - HW or smartcard implementations of encryption
 - Locks limiting access
 - Circuit boards that control access to disks in PCs
- **Policies**:
 - Added HW or SW features
 - Frequent changes of passwords
 - Must have training and administration
 - Legal and ethical controls (lack of understanding and standards for both)
- **Physical Controls**:
 - Locks on doors, guards at entry points,
 - backup copies of important artifacts,
 - physical site planning to avoid natural disasters



Effectiveness of Controls

- Awareness of problem
 - People using controls must understand the need
- Likelihood of Use:
 - *Principle of Effectiveness: Controls must be used to be effective. They must be efficient, easy to use, and appropriate.*
- Overlapping Controls:
 - Security for a PC may involve security for access to data, physical access to machine/storage media, and file locking mechanisms.
- Periodic Review:
 - Few controls are permanently useful.
 - Need to review and update.

CSE870: Advanced Software Engineering: Security Intro



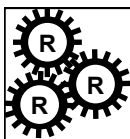
Cost Benefit Analysis

- Cost of Loss
 - Assigning cost range is sufficient
- Cost of Prevention
 - Cost of preventing each loss
- Adding up the Numbers
 - Matrix w/ assets, risks, possible losses
 - Includes: probability, the predicted loss, \$ required to defend against the loss
- Convincing Management
 - Risk assessment helps you make proper justifications for management



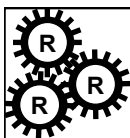
[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro



Security Policy

CSE870: Advanced Software Engineering: Security Intro

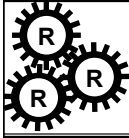


Creating Policy

- Defines what you consider to be valuable and what steps should be taken to safeguard those assets.
- General Policy
- Policy for Different Sets of Assets
 - Email, personnel data, etc.

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

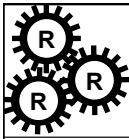


The Role of Policy

- Makes clear what is being protected and why
- States the responsibility for that protection
- Provides grounds upon which to interpret and resolve any later conflicts that might arise
- Should be general and change little over time
- Should *not* list specific threats, machines or individuals by name

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

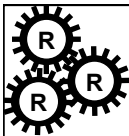


Policy Example

- “Information to be protected is any information discovered, learned, derived, or handled during the course of business that is not generally known outside of company X. This includes trade secret information (ours, and that of other organizations), patent disclosure information, personnel data, financial information, information about business opportunities, and anything else that conveys an advantage to company X so long as it is not disclosed. Personnel information about employees, customers and vendors is also to be considered confidential and protectable.

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

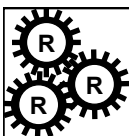


Standards

- Standards codify successful practice of security in an organization.
- Generally phrased in terms of “shall”
- Platform independent
- Imply a metric to determine if they have been met
- Developed to support policy
- Change slowly over time

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

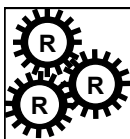


Example: Standard for Backups

- Backups *shall* be made of all online data and software on a regular basis. In no case will backups be done any less often than once every 72 hours of normal business operation. All backups should be kept for a period of at least six months; the first backup in January and July of each year will be kept indefinitely at an off-site, secured storage location. At least one full backup of the entire system *shall* be taken every other week. All backup media will meet accepted industry standards for its type, to be readable after a minimum of five years in unattended storage.

[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

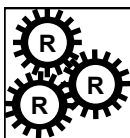


Guidelines

- “Should” statements in policies
- Interpret standards for a particular environment
- Guidelines may be violated
- Guide behavior
- Example:
 - Once per week, the administrator will pick a file at random from some backup made that week. The operator will be required to recover that file as a test of the backup procedures.

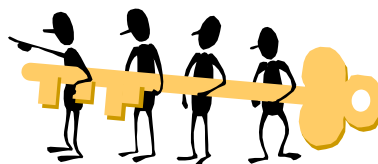
[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro



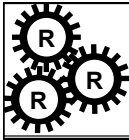
Keys to Developing Policy

- Assign an owner
- Be positive
 - People respond better to positive statements than to negative ones
- Remember that employees are people too
- Concentrate on education
- Have authority commensurate with responsibility
- Pick a basic philosophy
 - Be consistent
- Defend in depth



[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

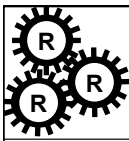


Goals for Security Policies

- Ensure only authorized users have access
- Prevent unauthorized users from gaining access
- Protect sensitive data from unauthorized access
- Prevent accidental damage to HW or SW
- Prevent intentional damage to HW or SW
- Create an environment that can recover quickly
- Communicate employee responsibilities

[J.B. Earp]

CSE870: Advanced Software Engineering: Security Intro

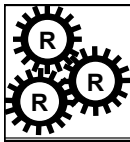


How to Attain the Goals?

- Form a committee
- Who should be involved?
- Decision-making people
- Security coordinator

[J.B. Earp]

CSE870: Advanced Software Engineering: Security Intro

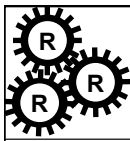


Security Policy Content

- Password policy
- S/W installation policy
- Confidential and sensitive data policy
- Network access policy
- Email use policy
- Internet use policy
- Modem use policy
- Remote access policy
- Policies for connecting to remote locations
 - Internet
 - Customer's networks
 - Vendor's networks
- Policies for use of laptops and loaner machines
- Computer room access policy

[J.B. Earp]

CSE870: Advanced Software Engineering: Security Intro

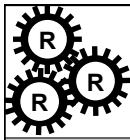


Response Policy

- Response team identified in policy
 - Dispatcher
 - Manager
 - Technical support specialist
 - Public relations specialist

[J.B. Earp]

CSE870: Advanced Software Engineering: Security Intro



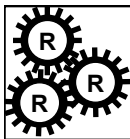
Four Easy Steps to a More Secure Computer

1. Decide how important security is to your site
2. Involve and educate your user community
3. Devise a plan for making and storing backups of your system data
4. Stay inquisitive and suspicious



[Garfinkel & Spafford]

CSE870: Advanced Software Engineering: Security Intro

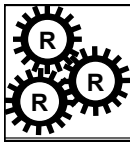


Threat Categories

- Data disclosure
 - Unauthorized access to an IS containing sensitive data (e.g., attacks resulting in data disclosure - eavesdropping)
- Fraud
 - Misrepresentation of identities (need to authenticate credit cards, etc.)
- Data insertion, removal, and modification
 - If it is possible to modify the data during transit, then it is possible to alter the financial transactions.

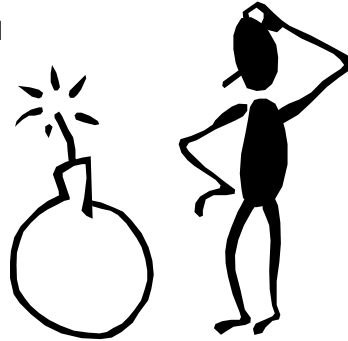
[Cyganski]

CSE870: Advanced Software Engineering: Security Intro



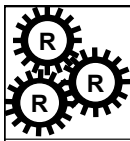
Attack Methods

- DoS (Denial of Service)
 - attacks involve restricting a shared resource from privileged users
 - maliciously causing a Net server to go down
 - unlawful under state and federal laws
- E-mail bombs (e.g., Spam)
 - series of mail messages sent as an annoyance.
- Viruses
- Spoofing
 - impersonation to gain unauthorized access



[J.B. Earp]

CSE870: Advanced Software Engineering: Security Intro

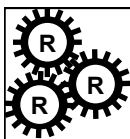


Security Services - 1

- Privacy
 - protect against unauthorized access to data.
- Authentication
 - positively identify an object or identity.
- Access Control
 - restrict access to an object or resource to only privileged identities.

[Cyganski]

CSE870: Advanced Software Engineering: Security Intro

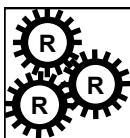


Security Services - 2

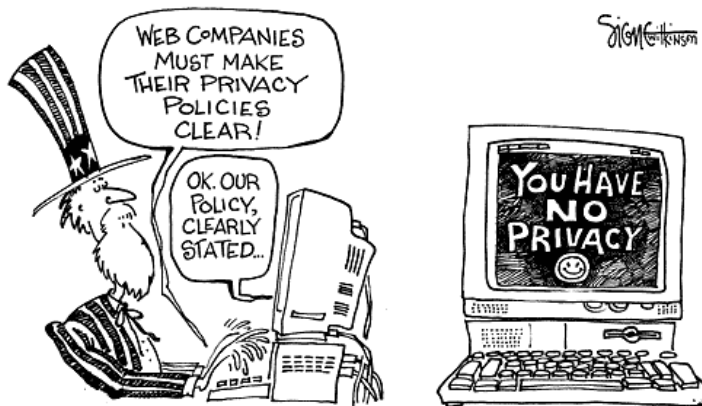
- Integrity
 - ensure that the data has not been altered since its creation.
- Non-repudiation:
 - Origin: message sender cannot deny being source msg
 - Submission: a provider can't deny submitting an order (time)
 - Delivery: can't deny receiving an item (for a customer)
 - Receipt: can't deny receiving a message/order
- Replay Prevention
 - ensure that data previously deemed valid cannot be resent by an attacker and mistakenly validated by a system a second time.

[Cyganski]

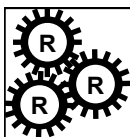
CSE870: Advanced Software Engineering: Security Intro



What is the Policy?

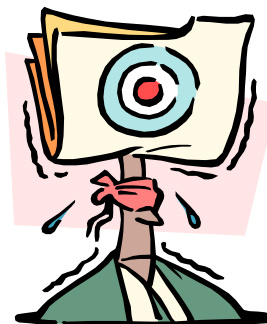


CSE870: Advanced Software Engineering: Security Intro

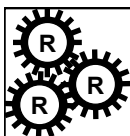


User Anxiety & Perceptions

- **Oblivious**
 - “Privacy Policy? What’s a privacy policy?”
- **Paranoid**
 - Doesn’t accept *any* cookies
 - Feels like a target
- **Misinformed**
 - “If there’s a seal, my personally identifiable information is safe”
 - “If there’s a privacy policy posted, I need not worry”
- **Informed**
 - Guards PII & ensures transactions w/ trusted source
 - PII: Personally Identifiable Information



CSE870: Advanced Software Engineering: Security Intro

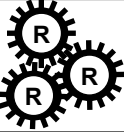


Internet Privacy Policies


- Beware of the short & sweet policies
 - Toysmart
- Beware of the long & legalese laden policies
- Trust seals are misleading to many customers
 - TRUSTe, BBBOnline, PrivacyRatings.com
- Policies often do not reflect actual site practices



CSE870: Advanced Software Engineering: Security Intro




TRUSTe



- Monitors licensees for compliance with posted privacy practices through a variety of measures
- A TRUSTe licensee's privacy policy must disclose:
 - what personal information is being gathered;
 - how the information will be used;
 - who the information will be shared with;
 - the choices available regarding how collected information is used;
 - safeguards in place to protect personal information from loss, misuse, or alteration;
 - and how individuals can update or correct inaccuracies in information collected about them

CSE870: Advanced Software Engineering: Security Intro



In-class exercise

- Design a security policy for an IVS (intelligent vehicle system).
- Hint: first perform risk assessment
- Determine whether you plan to have:
 - General policy
 - Asset-specific policy
 - Combination of both
- Identify your personnel (e.g., stakeholders, security-specific personnel, etc.)
- Present your policy to the class
 - Explain the rationale for your policy

CSE870: Advanced Software Engineering: Security Intro