Perspective: Security

CSCE 742 - Lecture 18 - 11/13/2018

“Your personal identity isn’t worth quite as much as it used to be - at least to thieves willing to swipe it. According to experts who monitor such markets, the value of stolen credit card data may range from $3 to as little as 40 cents.

That’s down tenfold from a decade ago - even though the cost to an individual who has a credit card stolen can soar into the hundreds of dollars.”

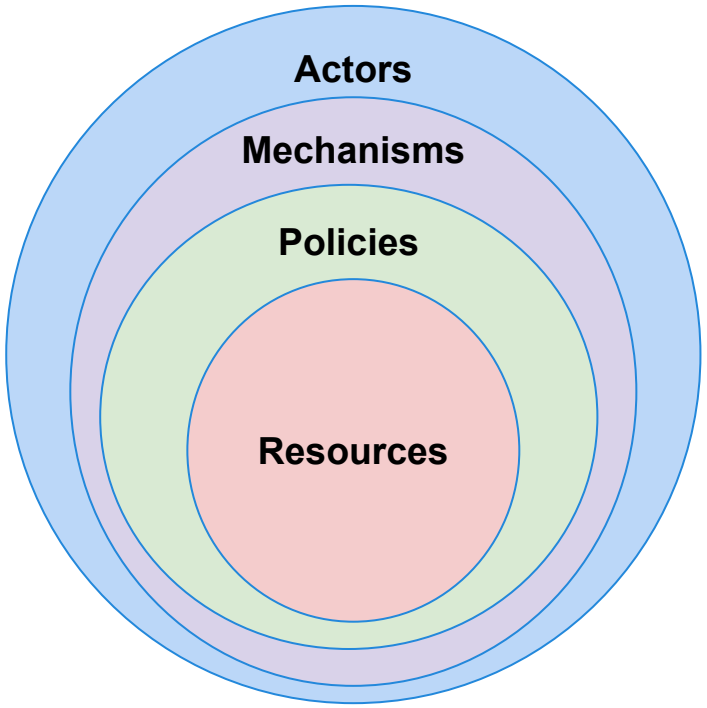
Taylor Buley, Forbes.com

Security

- The ability of the software to protect data and information from unauthorized access.
  - **While still providing access to people and systems that are authorized.**
- Can we protect software from attacks?
  - Any action taken against a computer system with the intent of causing harm.
  - Unauthorized access attempts.
  - Attempts to deny service to legitimate users.

Security

- Processes that allow owners of resources to control access.
  - Actors (systems or users).
  - Resources are sensitive elements, operations, and data of the system.
  - Policies define legitimate access to resourced.
  - Enforced by security mechanisms used by actors to gain access to resources.



Security Characterization (CIA)

- Confidentiality
  - Data and services are protected from unauthorized access.
    - A hacker cannot access your tax returns on an IRS server.
- Integrity
  - Data and services are not subject to unauthorized manipulation.
    - Your grade has not changed since assigned.
- Availability
  - The system will be available for legitimate use.
    - A DDOS attack will not prevent your purchase.

Supporting CIA

- Authentication
  - Verifies identities of all parties to a transaction.
    - Did the e-mail really come from the bank?
- Nonrepudiation
  - Guarantees that the sender of a message cannot deny sending the message, and the recipient cannot deny receiving the message.
    - You cannot deny ordering the book, and Amazon cannot claim you never ordered.
- Authorization
  - Grants user the privilege of performing a task.
    - The bank authorizes you to check balances.

Security Approaches

- Achieving security relies on:
  - Detecting attacks.
  - Resisting attacks.
  - Reacting to attacks.
  - Recovering from attacks.
- Objects being protected are:
  - Data at rest.
  - Data in transit.
  - Computational processes.

Security is Risk Management

- **Not a binary quality.**
  - All systems will be compromised.
  - Try to avoid attack, prevent damage, and quickly recover.
  - Balance risks against cost of guarding against them.
  - Set realistic expectations!

Impact on Views

- Context:
  - Identify external connections that could have vulnerabilities, and protect them from malicious use.
- Functional:
  - Identify which functional elements need to be protected. The functional structure may be impacted by the need to implement security policies.
- Information:
  - Helps you see what sensitive data needs to be protected. Information models are modified as a result of security design (e.g., partitioning information by sensitivity).

Impact on Views

- Concurrency:
  - May need to isolate elements into processes. This will affect the system's concurrency structure.
- Development:
  - Identify constraints that developers need to be aware of to ensure that security policy is enforced.
- Deployment:
  - May need special hardware or software, or to change deployment arrangements to address risks.
- Operational:
  - Make responsibilities clear, so security can be reflected in operational processes.

Security Scenarios

Security Quality Scenarios

- Measure of the system's ability to protect data from unauthorized access while still providing service to authorized users.
- Scenarios measure response to attack.
  - Stimuli are attacks from external systems/users or demonstrations of policies (log-in, authorization).
- Responses include auditing, logging, reporting, analyzing.
  - Response measures include amount of data loss/compromise, time to detect/mitigate, % of attacks resisted, etc.

Generic Security Scenario

- **Overview:** Description of the scenario.
- **System/environment state:** The attack can come when the system is either online or offline, either connected to or disconnected from a network, either behind a firewall or open to a network, fully operational, partially operational, or not operational.
- **External Stimulus:** The source of the attack may be either a human or another system. It may have been previously identified or may be currently unknown. A human attacker may be from outside the organization or from inside the organization. The stimulus is an attack (unauthorized attempt to display data, change or delete data, access services, change the system's behavior, or reduce availability).

Generic Security Scenario

- **Required system behavior:** The system should ensure that transactions are such that:
  - Data/services are protected from unauthorized access
  - Data/services are not manipulated without authorization
  - Parties to a transaction are identified and cannot repudiate their involvement
  - Data, resources, and system services will be available for legitimate use.

  The system should also track activities by
  - Recording access or modification and attempts to access data, resources, or services
  - Notifying appropriate entities (people or systems) when an apparent attack is occurring.

Generic Security Scenario

- **Response Measure:** Measures of a system's response include:
  - How much of a system is compromised when a particular component or data value is compromised.
  - How much time passed before an attack was detected
  - How many attacks were resisted
  - How long it took to recover from a successful attack
  - How much data was vulnerable to a particular attack.

Example Security Scenario

Unsuccessful Modification Attempt

- **Overview:** A disgruntled employee at a remote location attempts to change their pay rate.
- **System/environment state:** The system is operating normally, without problems. 100 active users are logged into the system.
- **External Stimulus:** An employee has discovered the location of a configuration file storing all employee pay rates. They log in (using their credentials) and use a stolen passkey to open the locked file. They modify the file with a new rate and save changes.
- **Required system behavior:** The system maintains an audit trail. The user is able to modify the file, as they have the passkey. However, the log records the date, time, identify of user, and modification made. System administrators are informed of the modification.
- **Response measure:** The correct data is restored within a day and the source of tampering has been identified and reported.

Example Security Scenario

Unsuccessful Authentication

- **Overview:** A user attempts to authenticate but the authentication fails due to unrecognized auth token or due to system unavailability.
- **System/environment state:** There is a valve installed on the tap. There is a flow meter installed on the tap. There is a piezo buzzer installed on the Kegboard. Authentication hardware (RFID or one-wire) is installed on the Kegboard. There is no pour in progress. The system is operating normally, without problems.
- **External Stimulus:** A user presents an auth token to the authentication hardware on the Kegboard.
- **Required system behavior:** The auth token is unrecognized, and the valve is not opened. An audible sound is played from the piezo buzzer, indicating authentication failure.
- **Response measure:** No beer is dispensed.

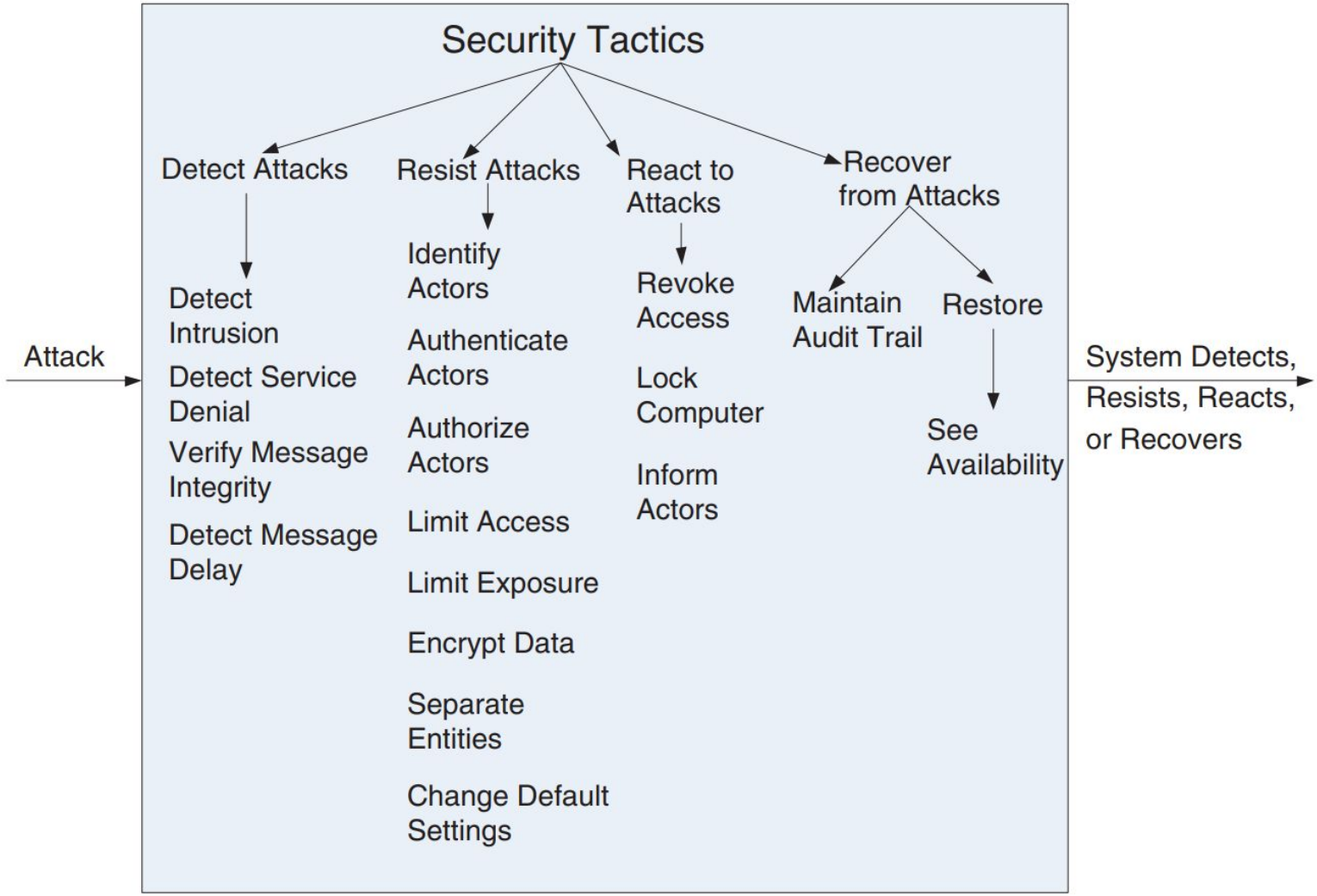
Security Tactics

Consider Physical Security

Secure buildings have:

- Limited access.
  - Security checkpoints, locked doors.
- Means of detecting intruders.
  - RFID badges.
- Deterrence mechanisms.
  - Armed guards.
- Reaction mechanisms.
  - Automatic locking of doors.
- Recover mechanisms
  - Off-site backup.

Security Tactics



Detect Attacks

- Detect intrusion:
  - Comparison of network traffic or service request patterns to known patterns of malicious behavior.
  - Can be based on protocol, TCP flags, payload sizes, applications, source/destination address, or port number.
- Detect service denial:
  - System is flooded with too many requests or carefully timed requests to prevent others from accessing data or services.
  - Comparison of pattern or signature of network traffic coming into a system to historic profiles of known denial-of-service attacks.

Detect Attacks

- Verify message integrity:
  - Use checksums or hash values to verify the integrity of messages or files.
  - Checksum: Maintain a piece of information about a file, check whether the file matches the stored information when used.
  - The checksum is often a **hash**, a unique string generated based on a piece of content.
  - Any change to the files or messages will cause a major change in the hash value.

Detect Attacks

- Detect message delay:
  - Detects man-in-the-middle attacks, where someone intercepts and modifies messages.
  - By checking the time it takes to deliver a message, we can detect suspicious timing behavior.
    - Too long of a delay: messages may be being intercepted or modified.
    - Variable timing: some messages may have been intercepted. Timestamps could have been edited.

Identifying Threats

- Threat models identify the threats you feel the system is subject to, their impact, and their likelihood.
  - Who is likely to try to infringe the security policy?
  - How will they try to do so?
  - What are the attacker's main characteristics?
    - Sophistication, motivation, resources.
  - What are the consequences of the policy being breached in this way?
  - Have security specialists review the model.

Attack Trees

- Structured notation for categorizing threats and their probability.
  - Represented visually as a tree as a nested list.
  - Root of the tree shows the goal of the attack.
  - Branches classify the different types of attacks that could be attempted.
  - Create a tree for each goal an attacker may have. Can be used to analyze security policies.

Goal: Obtain customer credit card details.

1. Extract details from database.
   1.1 Access database directly.
      1. Crack/guess database passwords.
      2. Crack/guess OS passwords that bypass database security.
   1.2 Access via a member of the administration staff.
      1. Bribe a database administrator (DBA).
      2. Conduct social engineering by e-mail to trick the DBA into revealing details

Attack Tree Example

2. Extract details from Web interface.
   - 2.1. Set up a dummy Web site and e-mail users the URL to trick them into entering credit card details.
   - 2.2. Crack/guess passwords for user accounts and extract details from the GUI.
   - 2.3. Send users a program by e-mail to record keystrokes.
   - 2.4. Attack the domain name server to hijack domain name and attack 2.1.
   - 2.5. Attack the server software directly to try to find loopholes in its security.

3. Find details outside the system.
   - 3.1. Conduct social engineering by phone/e-mail to get customer services staff to reveal card details.
   - 3.2. Direct a social-engineering attack on users by using public details from the site to make contact.

Designing Security Implementation

- In considering security design, identify system-wide policies and infrastructure to enforce policies.
  - Single-sign-on, firewalls, SSL encryption, policy management systems.
- For the attack tree:
  - Isolate databases and security-sensitive elements from public networks using firewalls.
  - Analyze paths into your system to check them for possible vulnerabilities.
  - Arranging penetration testing to see if experts can find ways into your system.

Security Implementation

- For the attack tree:
  - Identify an intrusion detection strategy that would allow security breaches to be recognized.
  - Train staff to avoid social-engineering attacks and to abide by strict privacy protection procedures.
  - Design site so that a minimal amount of user information (ideally, none) is publicly viewable.
  - Design site so sensitive information is never shown in full (e.g., display just the last four digits of a card number to allow legitimate users to identify their cards in lists).
  - Apply security-related software updates to all third-party software used in the system.
  - Remind users of security precautions they should take (e.g., not revealing passwords to anyone, including your staff; checking URLs before entering information)

Resist Attacks

- Identify actors:
  - What are the sources of external input?
    - (users, programs)
  - Can identify through ID of users, IP addresses of requests, protocols, ports.
  - Can restrict access to known IDs (white list).
  - Can at least log actions and associate with IDs.
- Authenticate actors:
  - Ensure any actor is actually who they say they are.
  - Passwords, one-time codes, certificates, biometrics.
  - Two-factor authentication pairs a password with a physical item or one-time code.

Resist Attacks

- Authorize actors:
  - **Only** authenticated actors have the right to access and modify data and services.
  - Provide an access control mechanism, based on actors, classes of actors, and actor roles.
- Limit access:
  - Control what and who may access the system.
  - White list (only those on the list can access)
  - Black list (those on the list cannot access)
  - Can limit access to CPUs, memory, network.
  - Achieved through process management, memory protection, blocking hosts, closing ports, rejecting protocols.

Limit Access

- Grant least privilege possible:
  - Only give users what they absolutely need.
  - Vary set of privileges over time.
- Firewalls:
  - Firewall is a single point of access to intranet.
  - A demilitarized zone is a subnet between internet and intranet, protected with firewalls on each side.
    - Used to let external users access services outside of the intranet.
  - Minimizes open ports in the internal firewall.
  - Limits access by identifying, authenticating, and authorizing all users.

Resist Attacks

- Limit exposure:
  - Concealing facts about a system.
    - (security by obscurity)
    - Ex: Hide how many entry points a system has.
  - Distributing critical resources so one exploit cannot fully compromise a resources.
    - Ex: Store data in multiple data centers.
- Encrypt data:
  - “Scramble” both data and communication.
  - Implemented by a virtual private network (VPN) or Secure Sockets Layer (SSL).
  - Can be symmetric (both parties use same key) or asymmetric (public/private keys).

Resist Attacks

- Separate entities:
  - Store elements and data on different physical machines (physical separation).
  - Virtual machines provide a sandboxed environment (software separation).
  - Keep elements as independent as possible.
  - Separate sensitive from nonsensitive data.
- Change default settings:
  - Do not use default passwords, port settings, IP addresses!
  - These are all public pieces of information.

Resist Attacks

- Defend in depth:
  - Use more than one security measure!
  - A series of defenses provides a greater level of security than a single one could.
  - Layer defenses in case any one fails.
- Keep security design simple:
  - The best security design is one you have analyzed for flaws.
  - Complex designs are hard to verify.
  - Make sure you can verify all security requirements.

React to Attacks

- Revoke access.
  - Lock access to sensitive resources for all users, legitimate or not.
  - Revoke rights for suspected user until cleared.
- Lock account or CPU.
  - If repeated failed login attempts, limit access to that machine or the account.
  - Can be time-based or until manual intervention.
- Inform actors.
  - Ongoing attacks may require action from administrators or systems.
  - Notify them of a detected attack.

Recover from Attacks

- If a system has been attacked, it needs to recover (restoration of data or services).
  - Additional servers or network connections may be kept in reserve for this purpose.
  - Need to maintain an audit trail.
    - Record of user and system actions and effects.
    - Helps trace actions of attacker.
    - Helps identify and prosecute the attacker.
    - Helps identify what data needs to be restored.
    - Helps build better defenses in the future.

Security Design

Allocation of Responsibilities

- Determine which system responsibilities need to be secure. For each, ensure that capability exists to:
  - Identify, authenticate, and authorize the actor.
  - Grant or deny access to data or services.
  - Record attempts to access/modify data or services.
  - Encrypt data.
  - Recognize reduced availability and inform appropriate personnel and restrict access.
  - Recover from an attack.
  - Verify checksums and hash values

Coordination Model

- Determine mechanisms required to communicate and coordinate with other systems or users.
  - For these, ensure that mechanisms are in place for authenticating and authorizing the actor or system, and encrypting data for transmission.
  - Ensure that demand for resources or services can be monitored and that unexpectedly high demands result in restricting or terminating the connection.

Data Model

- Determine the sensitivity of data fields.
  - Ensure that data of different sensitivity is separated.
  - Ensure that data of different sensitivity has different access rights and that access rights are checked prior to access.
  - Ensure that access to sensitive data is logged and that the log file is suitably protected.
  - Ensure that data is suitably encrypted and that keys are separated from the encrypted data.
  - Ensure that data can be restored if it is inappropriately modified.

Mapping Across Elements

- Determine how alternative mappings of elements could change:
  - How a user or system reads, writes, or modifies data
  - How services or resources are accessed.
  - How availability changes.
  - How logging and auditing is performed
- For all mappings, ensure that we can:
  - Identify, authenticate, and authorize an actor.
  - Grant or deny access to data or services.
  - Record attempts to access/modify data or services.
  - Encrypt data.
  - Recognize and recover from an attack.

Resource Management

- Determine how to identify and monitor a system or user.
  - (internal or external, authorized or not, with access to some or all resources).
- Determine how to authenticate, grant or deny access, notify entities, record actions, encrypt data, recognize and act on attacks.

Resource Management

- For each resource:
  - Can an external entity access or exhaust it?
  - Can we manage and log resource utilization?
  - Can we ensure that there are sufficient resources to perform the necessary security operations?
  - Can we ensure that contaminated elements can be prevented from contaminating other elements?
  - Can we ensure that shared resources are not used for passing sensitive data from an actor with access rights to that data to an actor without access rights?

Food for Thought

- Have you identified the sensitive resources contained in the system?
- Have you identified the actors who need access to the resources?
- Have you identified the system's needs for information integrity guarantees?
- Have you identified the system's availability needs?
- Have you established a security policy to define the security needs for the system, including which actors are allowed to perform which operations on which resources and where information integrity needs to be enforced?

Food for Thought

- Is the security policy as simple as possible?
- Have you worked through a formal threat model to identify the security risks your system faces?
- Have you considered insider as well as outsider threats to the system?
- Have you considered how the system's deployment environment will alter the threats to the system?
- Have you worked through example scenarios with your stakeholders so that they understand the planned security policy and the security risks the system runs?
- Have you reviewed your security requirements and design with external experts?

Food for Thought

- Have you addressed each threat identified in the threat model to the extent required?
- Have you considered all standard security principles when designing your security infrastructure?
- Is your security infrastructure as simple as possible?
- Have you defined how security breaches will be identified and how to recover from them?
- Have you applied the results of the Security perspective to all of the affected views?

Key Points

- Attacks against a system are attacks against the confidentiality, integrity, or availability of a system or its data.
  - Confidentiality: Keeping data away from those who shouldn’t have it.
  - Integrity: No unauthorized modifications or deletion of data.
  - Availability: System is accessible to authorized users.

Key Points

- Identifying, authenticating, and authorizing actors are how we determine who is entitled to access the system.
- No tactic is foolproof. Systems will be compromised.
- Tactics detect attacks, limit their spread, react, and recover from attacks.

Key Points

- Security is important!
  - Cannot cover it in one class. A whole semester may not cut it!
  - CSCE 548: Secure Software Development
    - A good start!
  - Also, classes on penetration testing, data security, etc.
    - You can get a whole degree in security.
- Pay attention, take precautions, fix bugs, keep up to date.

Next Time

- Perspective: Availability
  - Sources: Rozanski & Woods, Ch. 27
  - Bass, Clements, & Kazman, Ch. 5
- Homework:
  - Project, Part 3 - Due on Nov 18
  - Reading Assignment 3 - November 27th
    - Beaver et. al, “Finding a needle in Haystack: Facebook’s photo storage”
      - Summarize the system being developed.
      - Summarize the quality properties of interest to the developers, and how the design achieves them.
      - Identify two viewpoints that would be of interest for the stakeholders, and explain their importance.