Network Security Concepts

Target Course

Networks

Learning Goals

A student shall be able to:

1. Describe foundational security concepts in securing networks and systems.

IAS Outcomes

IAS Knowledge Topic	Outcome
Cryptography	1. Describe the purpose of cryptography and list ways it is used in data communications. [Familiarity] 2. Define the following terms: cipher, cryptanalysis, cryptographic algorithm, and cryptology, and describe the two basic methods (ciphers) for transforming plain text in cipher text. [Familiarity] 4. Explain how public key infrastructure supports digital signing and encryption and discuss the limitations/vulnerabilities. [Familiarity]
Principles of Secure Design	2. Summarize the principle of fail-safe and deny-by-default. [Familiarity] 3. Discuss the implications of relying on open design or the secrecy of design for security. [Familiarity] 4. Explain the goals of end-to-end data security. [Familiarity] 5. Discuss the benefits of having multiple layers of defenses. [Familiarity] 8. Describe the concept of mediation and the principle of complete mediation. [Familiarity] 9. Describe standard components for security operations, and explain the benefits of their use instead of reinventing fundamentals operations. [Familiarity] 11. Discuss the importance of usability in security mechanism design. [Familiarity]

Dependencies

- Assumes no pre-requisite knowledge.
- Should be covered at the same time as network topics are being introduced.

Summary

Introduce foundational security concepts and how they apply to a network environment.

Estimated Time

This module took about 1 lecture hour to cover. Please note that students had seen the fifteen security design principles in other upper-level courses. So this portion of the module was used as a review.

Materials

Are there additional foundational security concepts that apply specifically to networks?

- According to [1]:
  - Threats and Attacks
  - Security Principles
  - Access Control Models (covered in operating systems course)
  - Cryptographic Concepts
  - Implementation and Usability Issues
- According to [2]:
  - Protocols
  - Distributed systems

- Economics (not covered in this module)

What are some of the common threats and attacks?

This is discussed in module ***Common Attack Types***.

What are the security principles that should be adhered to when designing and implementing a network-based software system?

The first ten principles come from a paper written by Saltzer and Schroeder in 1975 [3].

1. Economy of mechanism

   Keep your design as simple as possible (aka KISS-Keep It Simple Stupid). This allows quality assurance methods (e.g., formal reviews, design walkthroughs) to have the greatest chance of finding security vulnerabilities.

2. Fail-safe defaults

   The default setting/action should be to favor security over usability. When in doubt, deny access. That is, your design should base access to data on permission rather than exclusion. Only when the protection scheme identifies conditions to permit access should the data be accessible. In contrast, creating a scheme that describes the conditions for refusing access presents the wrong psychological perspective for a secure software design. To state another way, the rules needed to express permission are likely to be simpler to understand than the rules needed to refuse permission.

3. Complete mediation

   Every access request should be checked for adherence to a protection scheme. This implies that a foolproof method of identifying the source of each request must be devised and suggests that ideas about improving performance by remembering the result of a previous authority check be examined skeptically. Care must be taken when a change in authority occurs, to ensure that the remembered results are systematically updated.

4. Open design

   Publish your design for anyone to review. This allows reviewers to comment on the security mechanisms being used while protecting the keys or passwords that are used by the mechanisms. You should assume that your design is not a secret.

5. Separation of privilege

   Originally defined as requiring multiple conditions to access a restricted resource or to perform some action (e.g., require two or more keys to unlock a protection mechanism). More recently, this has been defined as separating components of a system to reduce damage when a security breach occurs in any one component.

6. Least privilege

   Each user and program should operate with the minimum set of privileges necessary to accomplish the job. When software must access an information asset, it should ideally be granted this access only for the moments in time when it is using the information asset. This limits the damage that may result from an accident or error.

7. Least common mechanism

   Minimize the number of resources being shared/used by more than one user or system. Each security mechanism that is shared among most/all users or is shared among systems, especially when shared variables are used, represents a potential information path that may unintentionally compromise security. Do not share objects and protection mechanisms, instead create separate instances for each user or system interface.

8. Psychological acceptability

   User interfaces related to security mechanisms should be designed based on what a user expects. A well designed HCI will match the protection mechanism to the user's mental image of their protection goals.

9. Work factor

   The cost of compromising a security mechanism should be compared with the resources of an attacker when designing a security scheme. When a likely attacker has limited resources, the system may require less sophisticated defensive mechanisms.

10. Compromise recording

    It may be more desirable to record the details of an intrusion rather than designing more sophisticated defensive or prevention mechanisms.

In 2013, Gary McGraw [4] expanded on the security principles identified by Saltzer and Schroeder by adding five more principles[1].

11. Secure the weakest link

    The suite of security mechanisms being used are only as good as the weakest security mechanism being used. The analogy often used is that a chain is only as strong as its weakest link. Likewise, a system is only as secure as its weakest security mechanism.

12. Defend in depth

    Your design should include redundancy and layers of defense. This approach looks to manage security risks by using a diverse set of security mechanisms that provide redundant capabilities or are provided by different software layers.

13. Be reluctant to trust

    Be skeptical of security protections that are not within your software system. The quote "trust but verify" often used to describe international agreements to limit nuclear weapons is a good motto to follow when it comes to placing your software within an operating environment. A cloud provider may claim to provide certain protections, but it is best to verify these as best you can.

14. Promote privacy

    Your design needs to consider the types of personal information you are collecting from a user. Do you really

[1] McGraw includes the first eight principles from Saltzer and Schroeder and adds five principles to get to a total of 13 principles. Thus, McGraw's article is titled "Thirteen Principles …".

	need the information you are requesting? Should the personal information be encrypted? Does this data really need to be persistently stored?
15. Use your resources	Nobody knows everything about what a good software security design looks like. Talk to others about the design choices you are making. Have experts with different backgrounds review your design.

What are the cryptographic concepts that should be understood and used?

• Cryptosystems	Encryption and decryption of data. Symmetric encryption uses the same secret key to encrypt and decrypt the data. Asymmetric encryption (aka public-key encryption) uses a public key to encrypt the data and a private key to decrypt the data.
• Digital signatures	Use public-key encryption to verify who sent you the data. The sender encrypts the data using their private key. Anyone receiving this data may use the sender's public key to decrypt the data. In theory, only the sender knows their private key and so this verifies that the data did come from the sender.
• Simple attacks on cryptosystems	See the module ***Common Attack Types***.
• Cryptographic hash functions	A type of checksum on a data value that has two important properties: the hash function is a *one-way* function and the checksum generally contains many fewer bits than the original data value. A one-way hash function produces a checksum given a data value. But it is hard to recreate the data value if all you have is a checksum value.
• Digital certificates	A statement from a certificate authority that combines a public key with identifying information about the entity that owns that public key. This is used to ensure that the public key being used is associated with the entity you want to communicate with.

What issues exist in correctly implementing and using computer security mechanisms?

• Efficiency and usability	Providing security mechanisms that are slow gives a user a disincentive to use these mechanisms. Using a security mechanism that is easily misunderstood by a user allows for greater potential in misuse, which may result in vulnerabilities.
• Passwords	A primary authentication mechanism. Ideally, passwords should be hard to guess but easy to remember.
• Social engineering	See the module ***Common Attack Types***.
• Vulnerabilities from programming errors	A design should provide clear instructions on how to implement security mechanisms and how to test a system against all security requirements.

What are protocols and why are they an important security concept?

Computer systems include different types of entities that interact with each other. These entities include people (i.e., end-users), organizations, software elements, and hardware devices. Instances of these entity types need to communicate with each other in order for a computer system to perform as intended. A protocol is a set of rules that govern the communication between one or more instances of these entity types.

A protocol should be designed and implemented with security as an integral part of the development process. While it may be too expensive to design a protocol that protects the system from all possible attacks, a protocol should be designed to mitigate the risks associated with likely threats.

A protocol may be extremely simple, like swiping an employee id through a reader to gain access to a building or entering a username and password to gain access to a software system. A protocol may be complex, like the Transmission Control Protocol (TCP) used to guarantee delivery of packets within the Internet protocol stack.

What additional concepts are important to consider when developing large distributed systems?

- Concurrency

  While concurrency is a topic most often attributed to operating systems, the popularity of distributed systems with web front-ends have resulted in large server farms that have raised the stakes in designing, implementing and deploying these systems. Concurrency issues that may influence security include:

  - Replicating data across many sites versus potential delay in using centrally controlled data.
  - Locking access to prevent inconsistent updates.
  - The order in which updates occur. How do you represent time in a distributed system used around the world?
  - The possibility of a distributed deadlock.
  - Can the system be designed where every transaction is atomic, consistent, isolated and durable (aka ACID)?
  - Is the notion of time secure?

- Failure recovery

  What are the fault tolerance requirements for the system?

  How will the system recover from a failure?

- Naming

  How does a distributed system identify its users?

  How many identities (i.e., usernames) does a user of distributed systems have?

  How does a user identify a distributed system?

Assessment Methods

Below are questions that have been used on quizzes and exams.

The security design principle of psychological acceptability means that the user interface for a security mechanism conforms to the user's expectations.

a. True.

b. False.

Answer: true.

The security design principle of complete mediation means that each request to access data/system should be checked for adherence to a protection scheme. This design principle should be designed into all systems, regardless of the impact to performance or usability.

a. True.

b. False.

Answer: false.

The security design principle of compromise recording satisfies the non-repudiation security concept since recording (i.e., logging) events will prevent a user from denying an action that they performed.

a. True.

b. False.

Answer: True.

Proprietary designs are inherently more secure than open designs since fewer people know about the proprietary designs.

a. True.

b. False.

Answer: false.

Which of the following is the best description for the security design principle complete mediation?

a. Each request to access data should be checked for adherence to a protection scheme.

b. Each request to access data should be checked for adherence to an authentication scheme.

c. Each request to access data should be checked for adherence to an authorization scheme.

d. None of the above.

Answer: a. Each request to access data should be checked for adherence to a protection scheme.

An application layer protocol design that was included on a take-home final exam.

Develop Application-Layer Protocol (20 points)

You will develop a command-reply application layer protocol that could be used by autonomous vehicles that are operating on a highway.

Introduction

Automobile manufacturers have been doing research and development on autonomous vehicles for a number of years. The ultimate goal is to have automobiles that require no humanintervention while driving. While there are many forms this technology takes, the focus of this application-layer protocol will be on having autonomous vehicles form a convoy while on a highway. The term autonomous highway convoy (AHC) is defined as two or more vehicles that have successfully communicated with each other to form a convoy. All vehicles that are in an AHC travel at the same speed, are in the same highway lane, and have a fixed spacing of six feet between each vehicle in the convoy. For purposes of this exam, vehicles can only be added to an AHC by joining the front or back of the convoy.

Autonomous Vehicle Assumptions

Each vehicle that participates in an autonomous highway convoy shall have two cameras, a dedicated CPU for AHC operation, and a wireless communication link capable of communicating with other vehicles that are within fifty feet of the vehicle.

- One camera shall be in the front of each AHC vehicle. It is used to obtain the license plate number of the automobile immediately in front of the AHC vehicle. This automobile may or may not be part of the AHC.
- One camera shall be in the back of each AHC vehicle. It is used to obtain the license plate number of the automobile immediately in back of the AHC vehicle. This automobile may or may not be part of the AHC.
- Each vehicle within an AHC shall directly communicate with at most two other AHC vehicles - the vehicle directly in front and the vehicle directly behind.
    - The AHC vehicle in the front of the convoy shall allow a non-AHC vehicle to join the convoy in the front.
    - The AHC vehicle in the back of the convoy shall allow a non-AHC vehicle to join the convoy in the back.
    - No vehicle may join the AHC in between two existing AHC vehicles.
- Each vehicle license plate number contains at least six and no more than seven alphanumeric characters. (An alphanumeric character is in the range [A-Z,0,9]. All letters are uppercase.)

AHC Protocol Scope

The following describes the scope of the autonomous highway convoy protocol that you must describe.

- Assumptions:
    - There are (at least) two vehicles already in an AHC. Your protocol does not need to include command-reply messages that allow vehicles to create an AHC.
    - There are always (at least) two vehicles in an AHC. Your protocol does not need to include command-reply messages that may result from an AHC no longer existing.
- Your task is to develop a protocol to allow vehicles to be added to or removed from an existing AHC.
    - The protocol must have a command-reply message sequence that allows a non-AHC vehicle to join the front of the convoy by communicating with the AHC vehicle currently at the front of the convoy. Only the AHC vehicle currently at the front of the convoy and the non-ACH vehicle looking to join the front of the convoy will participate in this message sequence. (2.5 points)
    - The protocol must have a command-reply message sequence that allows an AHC vehicle to leave the front of the convoy by communicating with the AHC vehicle immediately behind it in the convoy. Only the first two AHC vehicles at the front of the convoy will participate in this message sequence. (2.5 points)
    - The protocol must have a command-reply message sequence that allows a non-AHC vehicle to join the back of the convoy by communicating with the AHC vehicle currently

at the back of the convoy. Only the AHC vehicle currently at the back of the convoy and the non-AHC vehicle looking to join the back of the convoy will participate in this message sequence. (2.5 points)
    - The protocol must have a command-reply message sequence that allows an AHC vehicle to leave the back of the convoy by communicating with the AHC vehicle immediately in front of it in the convoy. Only the last two AHC vehicles at the back of the convoy will participate in this message sequence. (2.5 points)

- Security is a significant concern for this protocol.
    - Develop an authentication scheme that is used whenever a vehicle wants to join an AHC.
        - Provide a concise explanation for how the authentication scheme works. (2.5 points)
        - Provide an explanation of the risks that are being mitigated by use of your authentication scheme. (2.5 points)
    - Develop an encryption scheme that is used by the AHC.
        - Provide a concise explanation for how the encryption scheme works. (2.5 points)
        - Provide an explanation of the risks that are being mitigated by use of your encryption scheme. (2.5 points)

References

[1] M.T. Goodrich & R. Tamassia, (2011). *Introduction to Computer Security*. Addison Wesley.

[2] R. Anderson, (2008). *Security Engineering, Second Edition*. Wiley.

[3] J.H. Saltzer & M.D. Schroeder, (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63(9), 1278-1308.

[4] G. McGraw, (2013). Thirteen principles to ensure enterprise system security. Retrieved on July 28, 2015 from searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security.