Network Security Overview

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
Foundational Concepts in Security	1. Analyze the tradeoffs of balancing key security properties (Confidentiality, Integrity, and Availability). [Usage] 2. Describe the concepts of risk, threats, vulnerabilities and attack vectors (including the fact that there is no such thing as perfect security). [Familiarity] 4. Explain the concept of trust and trustworthiness. [Familiarity]

Dependencies

- A student has successfully completed a data structures course.
- A student has been introduced to the cybersecurity topics covered in Input Validation and Principles.
- 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 is covered at the same time that computer networks are introduced. All of the introductory material took approximately 2 lecture hours to cover.

Materials

What are the foundational security concepts?

According to ***Chapter 1 Introduction*** in [1], the foundational security concepts include the following:

- Confidentiality, integrity, and availability (CIA)
- Assurance, authenticity, anonymity (AAA), and non-repudiation

What do confidentiality, integrity and availability (CIA) mean?

The concepts associated with CIA tend to be called security goals. Note that all quotes come from [1].

- Confidentiality is “the avoidance of the unauthorized disclosure of information”.
- Integrity refers to “information (that) has not been altered in an unauthorized way”.
- Availability is “information (that) is accessible and modifiable in a timely fashion by those authorized to do so”.

What do assurance, authenticity, anonymity (AAA) and non-repudiation mean from a security perspective?

These concepts tend to be considered fundamental security concepts. Note that all quotes come from [1].

- Assurance “refers to how trust is provided and managed in computer systems”. Trust involves the interaction of:

  - Policies (i.e., behavioral expectations).
  - Permissions (i.e., behaviors allowed by agents).
  - Protections (i.e., mechanisms that enforce policies and permissions).
- Authenticity is “the ability to determine that statements, policies, and permissions issued by persons or systems are genuine”.
- Anonymity is a “property that certain records or transactions (are) not … attributable to any individual”.
- Non-repudiation is the assurance that someone cannot deny something; a user should be responsible for their actions and should not be able to deny what they have done.

How do the layers of the Internet Protocol Stack address these goals and concepts?

The acronym **Midc** in the table below means *mostly, if designed correctly*. This is used in the Application Layer to indicate that a protocol at this layer could address the security goal or concept if the protocol was designed correctly.

Security Goal or Concept	**Internet Protocol Layer**				
	Application	**Transport**	**Network**	**Link**	**Physical**
Confidentiality	Midc	No	No	No	No
Integrity	Midc	No	No	No	No
Availability	Maybe	Maybe	No	No	No
Assurance	Midc	No	No	No	No
Authenticity	Midc	No	No	No	No
Anonymity	Maybe	Yes	Yes	Yes	Yes
Non-repudiation	Midc	No	No	No	No

The three cells with a *maybe* require a little explanation:

- An application layer protocol may help improve availability of the service if it can quickly throw away (i.e., ignore) those messages that are invalid (for any reason). However, if a distributed denial of service (DDOS) attack has enough machines all sending invalid messages to the service it is likely that the service will not be able to keep up.
- The transport layer may improve availability of a service in that TCP includes flow control and congestion control. These two features allow the sending-side of a TCP connection to slow down while waiting for the receiving-side to catch up. This may help, but the network, link, and physical layers will still need to get involved in shutting down the DDOS attack.
- An application layer protocol may promote anonymity if this is designed into the protocol.

Is perfect (100%) security attainable?

When looking at the three security goals and four fundamental security concepts, it should be observed that it is impossible to achieve 100% (or perfect) security. A few examples illustrate this fact.

- A trade-off is often made that may improve confidentiality but reduce availability. A simple scenario that demonstrates this is two-factor authentication. This improves confidentiality as an individual must provide two factors that demonstrate they are who they say they are, but reduces availability since an individual may not always possess both factors. That is, a misplaced cell phone can result in lacking the second factor that allows someone to authenticate.

- Anonymity and non-repudiation are in direct conflict with each other. Allowing a user to be anonymous while using a system means that we do not record the actions that they've performed.
- A policy may exist that requires management approval for transactions over a certain threshold (e.g., a funds transfer of $1,000 or more requires approval from a second manager). This improves assurance (i.e., trust in the system) and integrity while reducing confidentiality, since more individuals know about the funds transfer.

How are risks, threats, vulnerabilities, and attack vectors related to each other?

This is discussed in the Principles of Information Security module, and summarized below.

- A *vulnerability* is a susceptibility or weakness in the system that can expose it to an attack.
- The people or adversaries who may violate a systems security by exploiting vulnerabilities are called *threats* or *attackers.*
- *Risk* is the expected damage of a vulnerability. Thus risk considers the likelihood of a vulnerability being exploited and the cost of the damage.
- An *attack vector* describes how an attacker was able to gain access to the system and carried out the attack and is typically used to describe malicious attacks rather than unintentional errors.

What are some examples of network protocols that are trustworthy?

Consumers that use the web for retail or financial transactions are accustomed to seeing a lock icon that signifies that HTTPS is being used. While consumers may not understand the technical details of HTTPS, most will have confidence that their transaction data is being protected.

Underlying HTTPS is SSL/TLS (Secure Sockets Layer / Transport Layer Security) protocols. These two protocols, with TLS being the newer version, require that a server authenticate to a client and that the server and client negotiate a secret key to be used for symmetric encryption of the data being transmitted between the two hosts. Other than the widely known OpenSSL Heartbleed attack [2,3], the latest TLS protocol has stood the test of time.

Assessment Methods

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

Why is the application layer so important with regarding to meeting security goals and applying security concepts?

a. Because the other layers satisfy many but not all of the goals/concepts.

b. Because the other layers were developed by someone else, and so we should be cautious about relying on these to satisfy our security goals/concepts.

c. All of the above.

d. None of the above.

Answer: d. None of the above.

Explain why the security goal of confidentiality is so important when developing a distributed application?

Answer: If the security goal of confidentiality is not considered when developing a distributed application, then an individual may be able to:

- *Gain access to persistent data that is in plaintext form (i.e., has not been encrypted).*
- *Gain access to data in transit that is in plaintext form (i.e., has not been encrypted).*
- *Gain access to data that they are not authorized to view, update, or delete.*

Identify two security risks/issues if a distributed software application does not support the security goal of confidentiality?

Sample answers:

If the security goal of confidentiality is not considered when developing a distributed application, then an individual may be able to:

- *Gain access to persistent data that is in plaintext form (i.e., has not been encrypted).*
- *Gain access to data in transit that is in plaintext form (i.e., has not been encrypted).*
- *Gain access to data that they are not authorized to view, update, or delete.*

This may lead to:

- *Trust in the organization's systems being decreased.*
- *User's stop using the app since the app is less safe; their trust is lowered.*
- *A malicious actor stealing one's identify.*
- *A malicious actor leaking personal data to the public.*

Explain why the security goal of integrity is so important when developing a distributed application?

Answer: If the security goal of integrity is not considered when developing a distributed application, then an individual may be able to:

- *Update or delete data even though they do not have the authority to do so.*

This may lead to a lack of trust in the distributed application.

What is the impact to the security of a distributed software application if its design does not consider the security goal of integrity?

Sample answers:

If the security goal of integrity is not considered when developing a distributed application, then an individual may be able to:

- *Create, update, or delete data even though they do not have the authority to do so.*
- *Create a fake file (i.e., spoofing) that includes malware. A user downloading this file may behn get infected with the malware.*

This may lead to:

- *A lack of trust in the distributed application*

Explain why the security goal of availability is so important when developing a distributed application?

Answer: If the security goal of availability is not considered when developing a distributed application, then an individual may be able to:

- *Cause a denial-of-service by sending many requests/packets to a service thus keeping it too busy to respond to valid requests.*

This would prevent information from being delivered in response to a valid request in a timely fashion.

Identify a reason why the security goal of availability is so hard to achieve from the perspective of developing a distributed software application?

Sample answers:

The security goal of availability is so hard to achieve when developing a distributed application since:

- *A DDOS (distributed denial-of-service) attack is able to overwhelm the application with packets that are not valid, preventing the application from receiving packets that contain legitimate data. The design of the distributed software application cannot do anything to prevent the invalid packets from being received; other network protocol layers and devices must deal with this type of attack.*
- *The other goals may be in direct conflict with availability. For example, two-factor authentication will increase confidentiality by requiring a second method for proving who you are, but this second factor may not be working. This prevents user from accessing the application, resulting in the application being more secure but less available.*

Explain why the security goal of non-repudiation is so important when developing a distributed application?

Answer: If the security goal of non-repudiation is not considered when developing a distributed application, then an individual may be able to:

- *Alter some data (e.g., their salary) without any proof that this individual altered data they were not supposed to have access to.*

This would make attribution of an attack much harder to determine.

Explain why it is hard to develop a distributed software application that supports both nonrepudiation and anonymity?

Sample answer:

Developing a distributed software application that supports non-repudiation means that user actions would be recorded in a way that could be reviewed by others, which would prevent a user from performing application actions that are intended to be anonymous.

What does assurance mean as a foundational security concept?

Answer: Assurance is a belief an individual has that a system is trustworthy. This trust is managed by the policies, permissions, and protections the system uses.

Assurance is a belief an individual has that a system is trustworthy. This trust is managed by the policies, permissions, and protections the system implements.

a) Give an example of a policy, along with its appropriate permission(s) and/or protection(s), that will provide assurance to a user of a system.

b) Describe a risk that may affect the policy you described in 14.a, and identify how this risk affects the permission(s) and/or protection(s) associated with the policy.

Sample Answers:

- *A network policy that all data transmitted within an organization's internal network be encrypted using symmetric cryptography. An associated permission would be who (i.e., people and systems) has access to the symmetric key used to encrypt and decrypt the data. An associated protection would be how the entities that have permission to access to the symmetric key get authenticated to ensure they are who they say they are.*
  - *A risk of this policy is when the symmetric key is compromised, allowing anyone with the key to encrypt and decrypt data being transmitted on the internal network. This risk may be realized through a weak authentication system that is used to protect access and use of the key.*
- *Using HTTPS with a login screen. The user can only access their data after they login. If they try to use HTTP deny access.*
  - *If the certificate authority (for the SSL cert.) was compromised, the website may be easier to spoof, along with the certificate. Protection: alert the user to a breach.*
- *A company with a policy to require two-factor authentication and a promise to never disclose personal info to ad sites can assure users that their info on this company's systems is secure.*
  - *If a user gets their password broken or stolen, this breaks the policy of keeping user data confidential, because the malicious actor now has permission to access the account info. They can, if things aren't implemented correctly, reset or remove two-factor authentication and other protections on the account.*

What does authenticity mean as a foundational security concept?

Answer: Authenticity is a mechanism that may be used to provide evidence that an individual or system is who they say they are. An example of an authenticity mechanism is providing a valid username and password before gaining access to a system.

What does anonymity mean as a foundational security concept?

Answer: Anonymity may mean:

- *That an individual has the ability to use a system without first divulging any authentication credentials or data that will uniquely identify them.*
- *That a system is designed so that it does not attempt to identify its individual users. For example, the web was originally designed so that web servers did not know whom they were communicating with. However, cookies and other techniques are now used to allow web-based services to identify their users.*

As a designer of software applications, what types of questions should you be asking and discussing to help promote privacy of data?

Answers - general comments:

These questions should align with one or more of the security goals and concepts OR should be related to one or more of the 15 security design principles.

Answers - some sample questions:

- *Do we really need to obtain and either store or transmit this private data?*
- *Which of this data may be used maliciously to identify our users/customers?*
- *Which of this data should we encrypt while it is persistently stored?*
- *Which of this data should we encrypt while it is in transit?*
- *What type of cryptographic algorithm should we be used to encrypt/decrypt the data?*

References

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

[2] Semantic, (2017). Attack: OpenSSL Heartbleed CVE-2014-0160 3. Retrieved on November 26, 2017 from https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=27517.

[3] Wikipedia, (2017). Heartbleed. Retrieved on November 26, 2017 from https://en.wikipedia.org/wiki/Heartbleed.