

Lecture 9

Security (Part 1)

<lecturer, date>

Outline

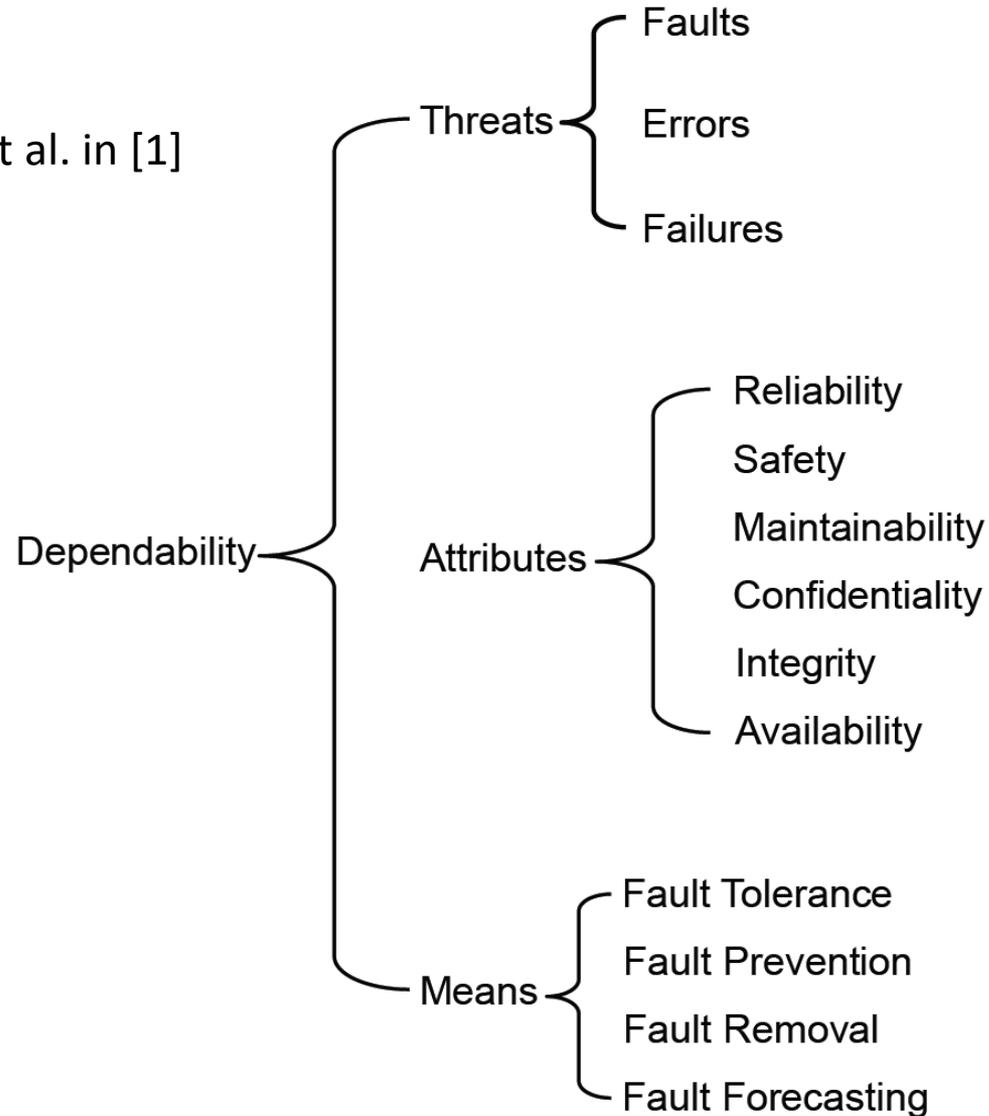
- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering
 - Denial of Service
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Introduction to dependability

- Definition: “*The ability of a computer system to deliver service that can justifiably be trusted.*”
- Example 1: consider the airbag system widely deployed in cars
 - The air bag should inflate in the event of a car crash
 - If the airbag does not inflate **always** whenever there is a crash, it cannot be termed as dependable.
- Example 2: consider Bob who has an email account
 - Only Bob should be able to read his email
 - If Alice is able to gain access to Bob’s email using some malicious means then the email system cannot be termed as dependable (or more precisely secure)

The dependability tree

Presented by Laprie et al. in [1]



Introduction to dependability

- Definition: “*The ability of a computer system to deliver service that can justifiably be trusted.*”
- Example 1: consider the airbag system widely deployed in cars
 - The air bag should inflate in the event of a car crash
 - If the airbag does not inflate **always** whenever there is a crash, it cannot be termed as dependable.

- Example 2: consider Bob who has an email account

Security Issue

Only Bob should be able to read his email

- If Alice is able to gain access to Bob’s email using some malicious means then the email system cannot be termed as dependable (or more precisely secure)

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering
 - Denial of Service
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Introduction to security

- Example 2: consider Bob who has an email account
 - Only Bob should be able to read his email
 - If Alice is able to gain access to Bob's email using some malicious means then the email system cannot be termed as dependable (or more precisely secure)



Alice gets hold of Bob's computer in which his password is saved.

Alice makes use of her knowledge to "hack" into the email system e.g., using *code-injection*.

Introduction to security

- Example 2: consider Bob who has an email account
 - Only Bob should be able to read his email
 - If Alice is able to gain access to Bob's email using some malicious means then the email system cannot be termed as dependable (or more precisely secure)

Alice gets hold of Bob's computer in which his password is saved.

The need for **physical security**

Alice makes use of her knowledge to "hack" into the email system e.g., using *code-injection*.

The need for **computer security**

Introduction to security

- Example 2: consider Bob who has an email account
 - Only Bob should be able to read his email
 - If Alice is able to gain access to Bob's email using some malicious means then the email system cannot be termed as dependable (or more precisely secure)

Alice gets hold of Bob's computer in which his password is saved.

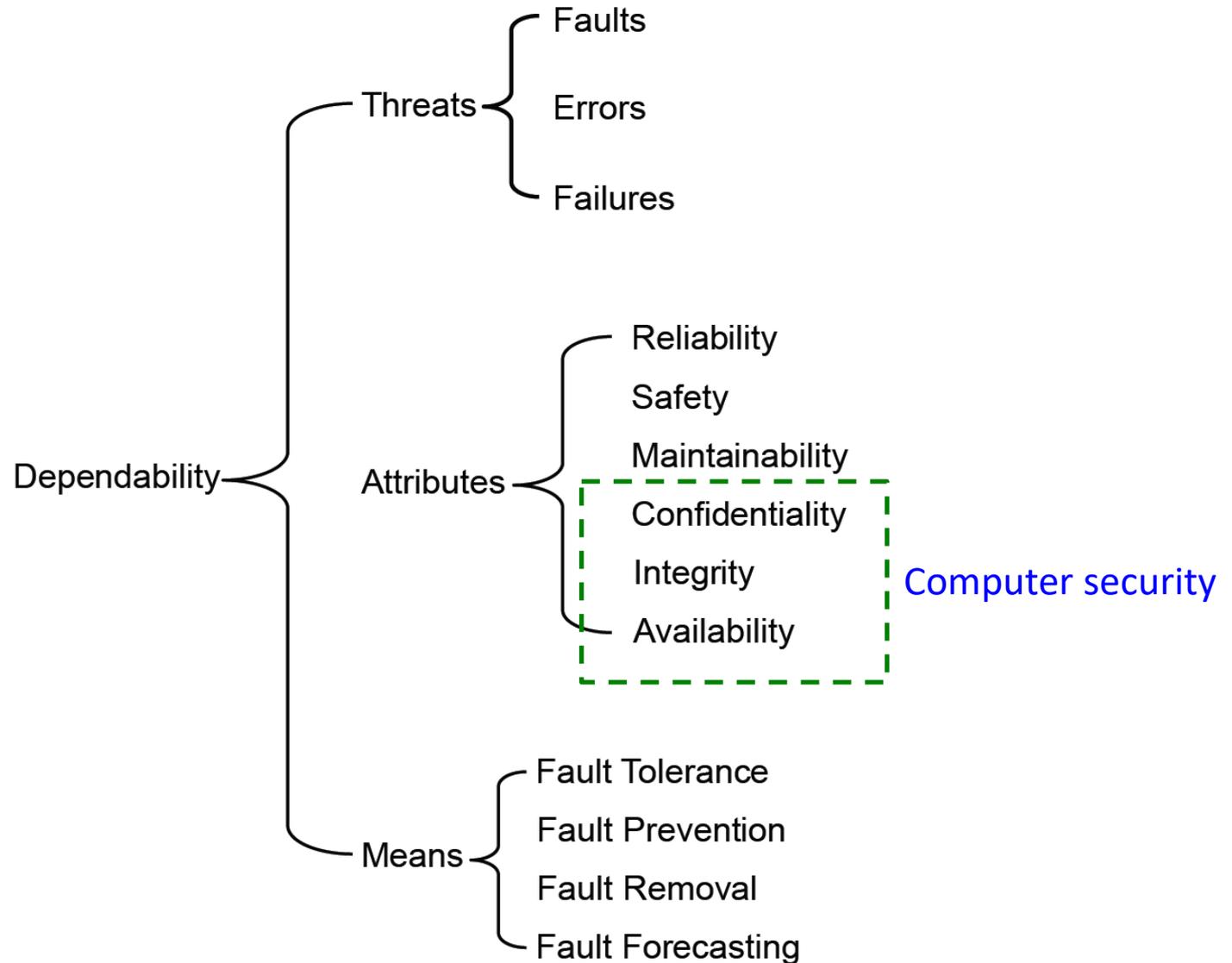
The need for **physical security**

Our focus

Alice makes use of her knowledge to "hack" into the email system e.g., using *code-injection*.

The need for **computer security**

The big picture



Computer security

- Confidentiality is defined as the absence of **unauthorized disclosure** of information
 - e.g., compromised if a hacker gets hold of the decryption key
- Integrity is defined as the absence of **improper system alterations**
 - e.g., compromised in case viruses infect computers
- Availability is defined as the **readiness** for a correct service
 - e.g., compromised if there is a denial-of-service attack

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering
 - Denial of Service
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Threat modeling

- Vulnerability is defined as a **weakness in the computer system** that can be exploited to gain unauthorized access.
- Threat is the existence of a possibility that **vulnerabilities** in the computer system could be **exploited** in order to **compromise** confidentiality, integrity or availability.
- Threat modeling is the **structured activity** of identifying computer security threats and vulnerabilities.
- Commonly expressed using an **attacker-centric** model
 - Who wants to attack?
 - What are their goals?
 - How do they perform the attack?

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering
 - Denial of Service
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Social engineering

- Who wants to attack?
 - con experts, hackers, fraudsters, enthusiasts
 - Examples:
 - credit card thieves
 - hackers wanting unauthorized access to the networks of e.g., large companies
- What are their goals?
 - get hold of confidential information
 - Examples:
 - credit card or bank account numbers
 - company account passwords
- How do they perform the attack?
 - usually by psychological manipulations
 - Examples:
 - Lottery scams
 - Phishing
 - Demo
 - one of you please enter your email password here:

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering ✓
 - Denial of Service
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Denial of service (DoS)

- Who wants to attack?
 - Mostly people who wants to make a statement, or intending harm
 - Examples:
 - terrorists
 - groups such as Anonymous
- What are their goals?
 - cause financial damage
 - as a means of protest (though illegal)
- How do they perform the attack?
 - send massive requests such that the service (e.g., a server) is overwhelmed
 - Examples:
 - The DoS that exploited the vulnerabilities in NTP servers (see reference 8)
 - Morris worm (see reference 10)

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering ✓
 - Denial of Service ✓
 - Code injection
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Code injection

- Who wants to attack?
 - Mostly people intending harm
 - Examples:
 - terrorist, thieves, spies
 - groups such as Anonymous
- What are their goals?
 - espionage
 - financial gains
 - as a means of protest (though illegal)
- How do they perform the attack?
 - “Inject” a malicious code into the vulnerable application
 - Usually performed by crafting URLs
 - Examples:
 - SQL injection (see reference 11)

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering ✓
 - Denial of Service ✓
 - Code injection ✓
 - Spoofing
- Examples
 - Stuxnet
- Conclusion

Spooftng

- Who wants to attack?
 - Mostly people intending harm
 - Spies
 - Con artists
- What are their goals?
 - espionage
 - financial gains
 - Perform further attacks after concealing identity
 - Spamming
- How do they perform the attack?
 - Creation of network traffic or applications that masquerades as somebody else
 - Examples:
 - Email spoofing: modifying email headers to pretend as somebody else
 - IP address spoofing: modifying network packets to masquerade as legitimate user

Outline

- Introduction to dependability
 - The dependability tree
- Introduction to security
 - Physical security
 - Information security
 - Confidentiality
 - Integrity
 - Availability
- Threat modeling
- Common threats
 - Social engineering ✓
 - Denial of Service ✓
 - Code injection ✓
 - Spoofing ✓
- Examples
 - Stuxnet
- Conclusion

Example of attacks on industrial systems

The [stuxnet](#) is a computer worm that was designed to infect programmable logic controllers (PLCs) that is typically used for industrial automation.

- Exploited 4 zero-day flaws (i.e., previously unknown vulnerabilities)
- Targets Microsoft Windows OS and propagates through the network
- Scans for a particular Siemens software on computers controlling PLCs
- On finding such a computer it infects the PLC and executes malicious commands, else it becomes dormant
- Causes damage to the physical components, fast-spinning centrifuges in this case

- Allegedly designed to sabotage Iranian nuclear program

Conclusions

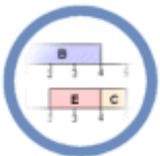
- Different types of vulnerabilities exist in computer systems making them prone to cyber attacks
- Securing them requires a multi-level strategy
- Developers need to understand the security risks involved and its implications to the user
- Developers need to be aware of the common threat models
- End users must be educated about the common types of attacks

References

- 1) Basic concepts and taxonomy of dependable and secure computing, Avizienis, A. ; Laprie, J.-C. ; Randell, B. ; Landwehr, C., IEEE Transactions on Dependable and Secure Computing, 2004
- 2) Threat Modeling: A Process To Ensure Application Security, <http://www.sans.org/reading-room/whitepapers/securecode/threat-modeling-process-ensure-application-security-1646>
- 3) Template Sample: Web Application Threat Model, <https://msdn.microsoft.com/en-us/library/ff649779.aspx>
- 4) Understanding Denial-of-Service Attacks (Security Tip (ST04-015)), <https://www.us-cert.gov/ncas/tips/ST04-015>
- 5) Anonymous, http://en.wikipedia.org/wiki/Anonymous_%28group%29
- 6) The Real Story of Stuxnet, <http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/>
- 7) Building a Cyber Secure Plant, <http://www.totallyintegratedautomation.com/2010/09/building-a-cyber-secure-plant/>

References

- 8) World's largest Denial of Service attack caused by vulnerability in the infrastructure of the web, <http://www.independent.co.uk/life-style/gadgets-and-tech/worlds-largest-denial-of-service-attack-caused-by-vulnerability-in-the-infrastructure-of-the-web-9122200.html>
- 9) Denial-of-Service Attacks, <http://www.cert.org/historical/advisories/ca-1997-28.cfm>
- 10) The Helminthiasis of the Internet, <ftp://ftp.isi.edu/in-notes/rfc1135.txt>
- 11) SQL Injection, <https://technet.microsoft.com/en-us/library/ms161953%28v=SQL.105%29.aspx>
- 12) Social Engineering: Manipulating the Source, <http://www.sans.org/reading-room/whitepapers/engineering/social-engineering-manipulating-the-source-32914>



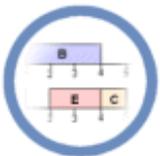
Lab 9

Security (Part 1)

<lecturer, date>

Description

- Download Webgoat: <http://webgoat.github.io/>
- Install it using the instructions at:
https://www.owasp.org/index.php/WebGoat_Installation
- Solve at least 4 assignments



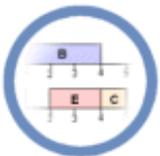
Seminar 9

Security (Part 1)

<lecturer, date>

Description

- Discuss the solutions of the lab
- Using the STUXNET incident as example, discuss the possible strategies that can be adopted minimize industrial security threats
 - The Real Story of Stuxnet,
<http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/>



Mini-project 9

Security (Part 1)

<lecturer, date>

Description

- Write a 4 page report on how you can apply security concepts for embedded systems
 - Merits
 - Demerits
 - Some recent advances
 - Open problems