

CISC 3325 - INFORMATION SECURITY

Security Principles



Adapted from *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

COMPUTER SECURITY QUIZ

What is penetration testing?

- A. A procedure for testing libraries or other program components for vulnerabilities
- B. Whole-system testing for security flaws and bugs
- C. A security-minded form of unit testing that applies early in the development process
- D. All of the above



What is penetration testing?

- A. A procedure for testing libraries or other program components for vulnerabilities
- ✓ - B. Whole-system testing for security flaws and bugs
- C. A security-minded form of unit testing that applies early in the development process
- D. All of the above

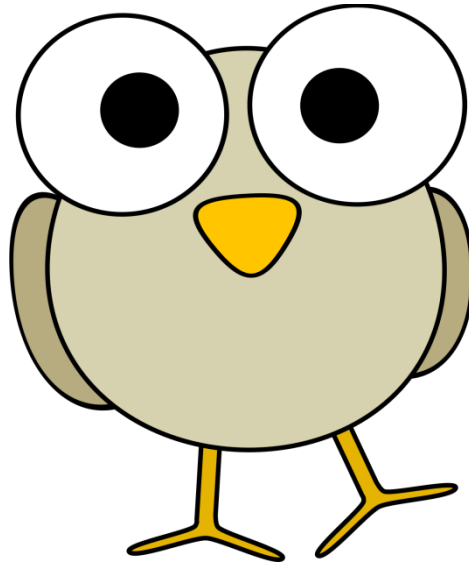
Which of the following are benefits of penetration testing?

- A. Results are often reproducible
- B. Full evidence of security: a clean test means a secure system
- C. Compositionality of security properties means tested components are secure even if others change
- D. They specifically consider adversarial thinking, which is not usually necessary for normal tests

Which of the following are benefits of penetration testing?

- 
- A. Results are often reproducible
 - B. Full evidence of security: a clean test means a secure system
 - C. Compositionality of security properties means tested components are secure even if others change
 - D. They specifically consider adversarial thinking, which is not usually necessary for normal tests
- 

- Questions?



??



SECURITY PRINCIPLES

Making decisions about technology in an uncertain world

<https://criticaluncertainties.com/category/security/saltzer-and-schroeder-principles/>

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Security Principles

- How to design of secure software systems?
- Saltzer and Schroeder's design principles
 - Published in The Protection of Information in Computer Systems [1975]

Design Principles for Security

- Least privilege
- Economy of mechanism
- Fail-safe defaults
- Open design
- Complete mediation
- Separation of privilege
- Least common mechanism
- Psychological Acceptability - Ease of use

Least privilege

- Each program and user of a computer system should operate with the bare minimum privileges necessary to function properly.
 - If this principle is enforced, abuse of privileges is restricted, and the damage caused by the compromise of a particular application or user account is minimized.
 - The military concept of need-to-know information is an example of this principle.

Fail-safe defaults

- Default configuration of a system should have a conservative protection scheme
 - E.g., when adding a new user to an operating system
 - the default group of the user should have minimal access rights to files and services.
 - Unfortunately, operating systems and applications often have default options that favor usability over security.
 - This has been historically the case for a number of popular applications
 - such as web browsers that allow the execution of code downloaded from the web server.

Economy of mechanism

- This principle stresses simplicity in the design and implementation of security measures.
 - simplicity is especially important in the security domain
 - since a simple security framework facilitates its understanding by developers and users
 - enables the efficient development and verification of enforcement methods for it.
 - This is also applicable to most engineering projects

Open design

- The security architecture and design of a system should be made publicly available.
 - Security should rely only on keeping crypto keys secret.
 - Open design allows for a system to be scrutinized by multiple parties
 - leads to the early discovery and correction of security vulnerabilities caused by design errors.

Kerckhoffs's principle

- A cryptosystem should be secure even if everything about the system, except the key, is public knowledge
 - Auguste Kerckhoffs, [19th century]

Kerckhoffs's principle

- Design principles for military ciphers:
 - System must be practically indecipherable
 - If not mathematically
 - Should not require secrecy
 - it should not be a problem if it falls into enemy hands
 - It must be possible to communicate and remember the key
 - Without written notes
 - Parties must be able to change or modify it at will;

Kerckhoffs's principle

- Design principles for military ciphers (cont.):
 - Should be portable and applicable to telegraph communications
 - should not require several persons to handle or operate;
 - System must be easy to use
 - should not require users to know and comply with a long list of rules.
 - given the circumstances in which it is to be used, the

Open design

- The security architecture and design of a system should be made publicly available.
 - Security should rely only on keeping crypto keys secret.
 - Open design allows for a system to be scrutinized by multiple parties
 - leads to the early discovery and correction of security vulnerabilities caused by design errors.
- The open design principle is the opposite of the approach known as security by obscurity
 - which tries to achieve security by keeping cryptographic algorithms secret
 - has been historically used without success by several organizations.

Complete mediation

- The idea behind this principle is that every access to a resource must be checked for compliance with a protection scheme.
 - => one should be wary of performance improvement techniques that save the results of previous authorization checks
 - since permissions can change over time.
 - For example, an online banking web site should require users to sign on again after a certain amount of time
 - say, 15 minutes, has elapsed.

Separation of privilege

- This principle dictates that multiple conditions should be required to achieve access to restricted resources
 - or have a program perform some action.

Least common mechanism

- In systems with multiple users, mechanisms allowing resources to be shared by more than one user should be minimized.
 - E.g., if a file or application needs to be accessed by more than one user, then these users should have separate channels by which to access these resources
 - to prevent unforeseen consequences that could cause security problems.

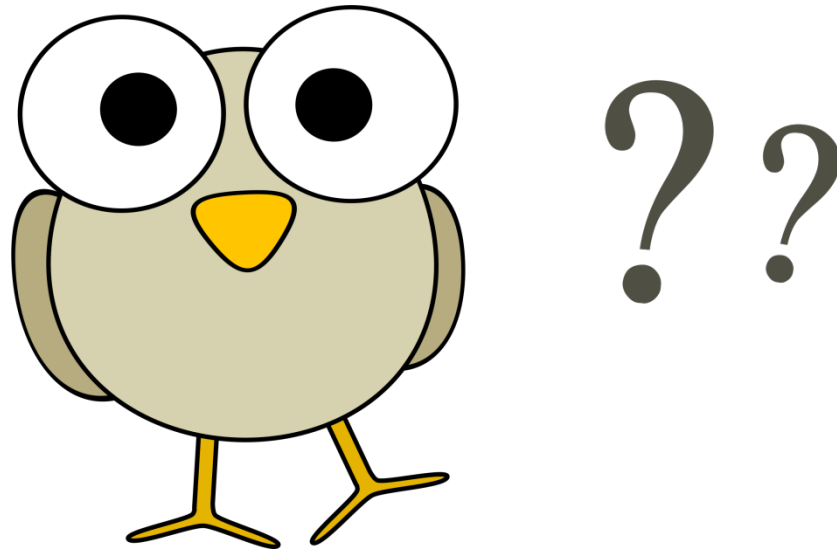
Psychological acceptability - Ease of use

- This principle states that user interfaces should be **well designed and intuitive**
- All security-related settings should adhere to what an ordinary user might expect.

Summary

- Malware can have a variety of harmful effects depending on its characteristics, including resource usage, infection vector, and payload
- Developers can use a variety of techniques for writing and testing code for security
- Following design principles helps protect systems against attacks

- Questions?



WHAT ABOUT INTERNET OF THINGS?

Internet of Things (IoT)

- A network of physical devices, vehicles, home appliances and other items
- Devices are embedded with electronics, software, sensors, actuators
- Network connectivity enables these objects to connect and exchange data
 - But can be a source of vulnerabilities

“Internet of Things” devices

- “Internet of things” devices are connected through the network
- Attackers may access device vulnerabilities through the network
- Devices are typically very cost sensitive
 - Therefore, very little support after purchase
 - User can not tell if they are secure

Example of IoT devices

- Wearable devices
- Medical devices
- Home automation
 - Refrigerators, stoves, etc.
- Automotive
- Etc.



IoT Devices



<https://www.pentasecurity.com/blog/10-smartest-iot-devices-2017/>

“Internet of Things” devices

- Device only communicates through a central service
- Most of the companies running the service are “Data Asset” companies
 - Make their money from advertising, not from the product itself
 - Product may actually be subsidized considerably
 - Companies include Google, Amazon, Salesforce, etc...
 - Apple HomeKit provides a higher level of security
 - But you still need to trust that it does not report to a third party

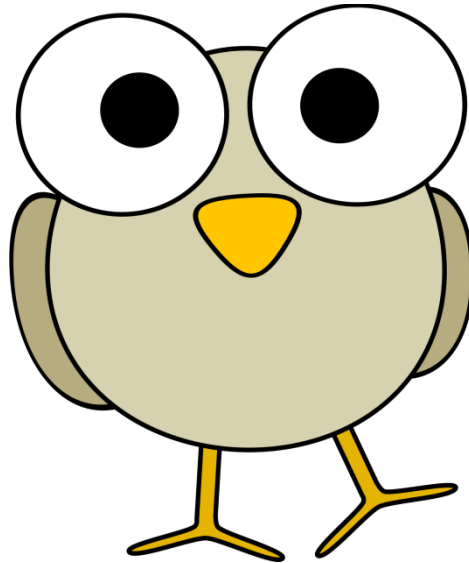
Risks in using IoT devices

- Devices collect information about user via device
- Many devices do not require strong passwords (or any passwords)
- Devices may use unencrypted network services
- User interface of applications may be vulnerable to different attacks
 - Shoulder surfing, etc.

“Internet of Things” devices

- Goal: Increase security of devices
- Possible defenses:
 - Develop applications with security in mind
 - Consider security throughout the entire development lifecycle of IoT devices
 - Not treat it separately or as an “add-on”
 - Integrate human understanding and algorithms
 - Take in account the “human factor”

- Questions?

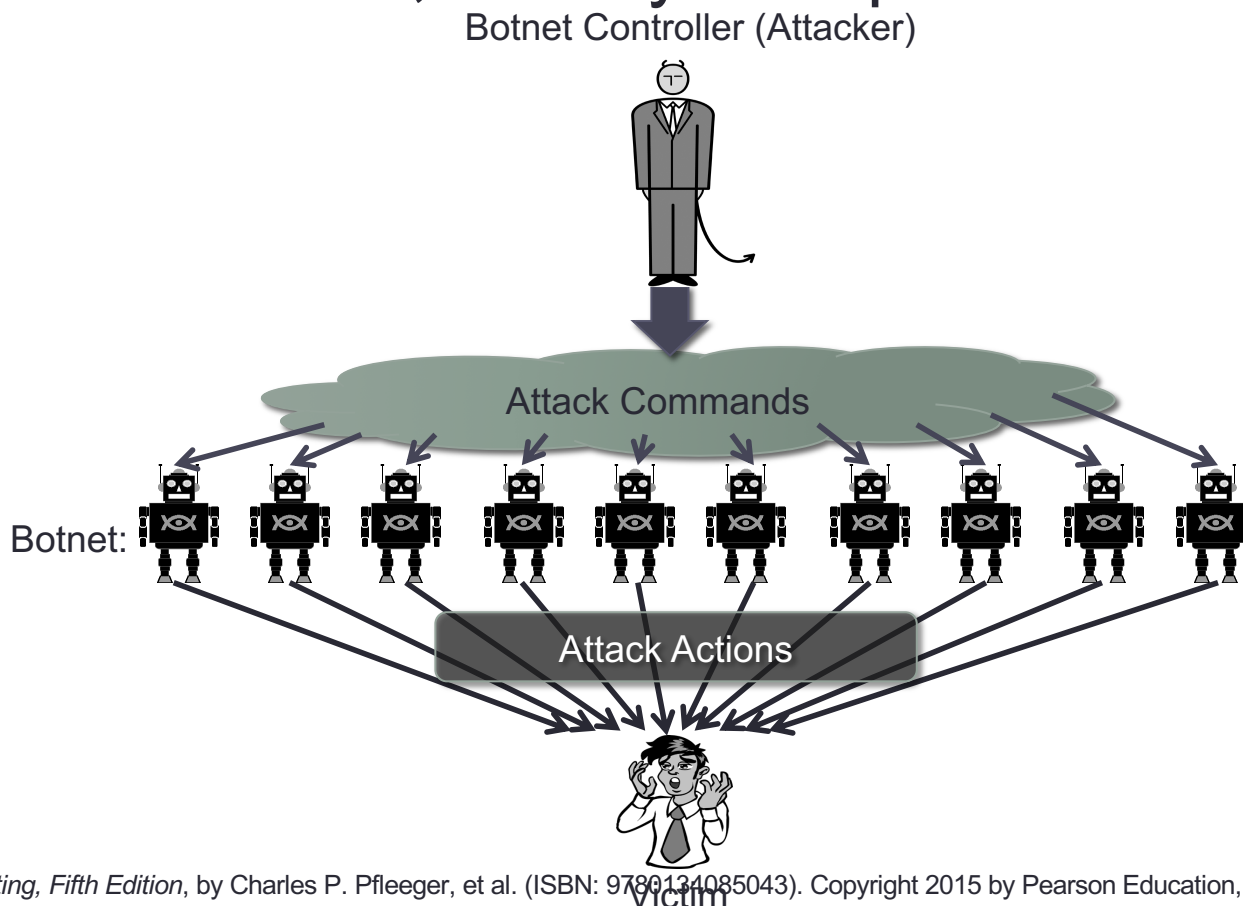


??

ADDITIONAL MALWARE

Malware Zombies

- Malware can turn a computer in to a **zombie**, which is a machine that is controlled externally to perform malicious attacks, usually as a part of a **botnet**.





Insider Attacks

- An **insider attack** is a security breach that is caused or facilitated by someone who is a part of the organization
 - The organization that controls or builds the asset that should be protected
- An insider attack refers to a security hole that is created in a software system
 - by one of its programmers.

<https://www.observeit.com/blog/top-motivating-factors-inside-threats/>



Backdoors

- A **backdoor**, A.K.A. a **trapdoor**, is a hidden feature or command in a program
- A Backdoor allows a user to perform actions he or she would not normally be allowed to do.
- When used in a normal way, this program performs completely as expected and advertised.
- But if the hidden feature is activated, the program does something unexpected
 - often in violation of security policies, such as performing a privilege escalation.

<https://www.computerhope.com/jargon/b/backdoor.htm>

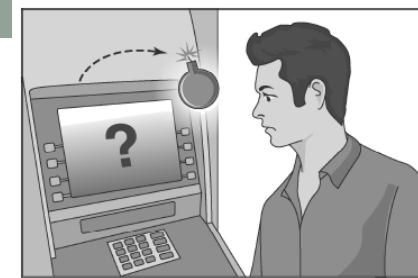


Backdoors

- A **backdoor**, A.K.A. a **trapdoor**, is a hidden feature or command in a program
- A Backdoor allows a user to perform actions he or she would not normally be allowed to do.
- Benign example: **Easter Eggs** in DVDs and software

<https://www.computerhope.com/jargon/b/backdoor.htm>

Logic Bombs



- A **logic bomb** is a program that performs a malicious action as a result of a certain logic condition.
- The classic example of a logic bomb is a programmer coding up the software for the payroll system
 - The programmer puts in code to make the program crash
 - should it ever process two consecutive payrolls without paying him.
- Another classic example combines a logic bomb with a backdoor:
 - a programmer puts in a logic bomb that will crash the program on a certain date.

The Omega Engineering Logic Bomb



- An example of a logic bomb that was actually triggered and caused damage:
 - Programmer Tim Lloyd was convicted of using on his former employer, Omega Engineering Corporation.
 - On July 31, 1996, a logic bomb was triggered on the server for Omega Engineering's manufacturing operations
 - Ultimately cost the company millions of dollars in damages, led to lay-off of many of its employees.



The Omega Bomb Code

- The Logic Behind the Omega Engineering Time Bomb included the following strings:
- 7/30/96
 - Event that triggered the bomb
- F:
 - Focused attention to volume F, which had critical files
- F:\LOGIN\LOGIN 12345
 - Login a fictitious user, 12345 (the back door)
- CD \PUBLIC
 - Moves to the public folder of programs
- FIX.EXE /Y F:*.*
 - Run a program, called FIX, which actually deletes everything
- PURGE F:\ALL
 - Prevent recovery of the deleted files

Defenses against Insider Attacks

- Avoid single points of failure.
- Use code walk-throughs.
- Use archiving and reporting tools.
- Limit authority and permissions.
- Physically secure critical systems.
- Monitor employee behavior.
- Control software installations.

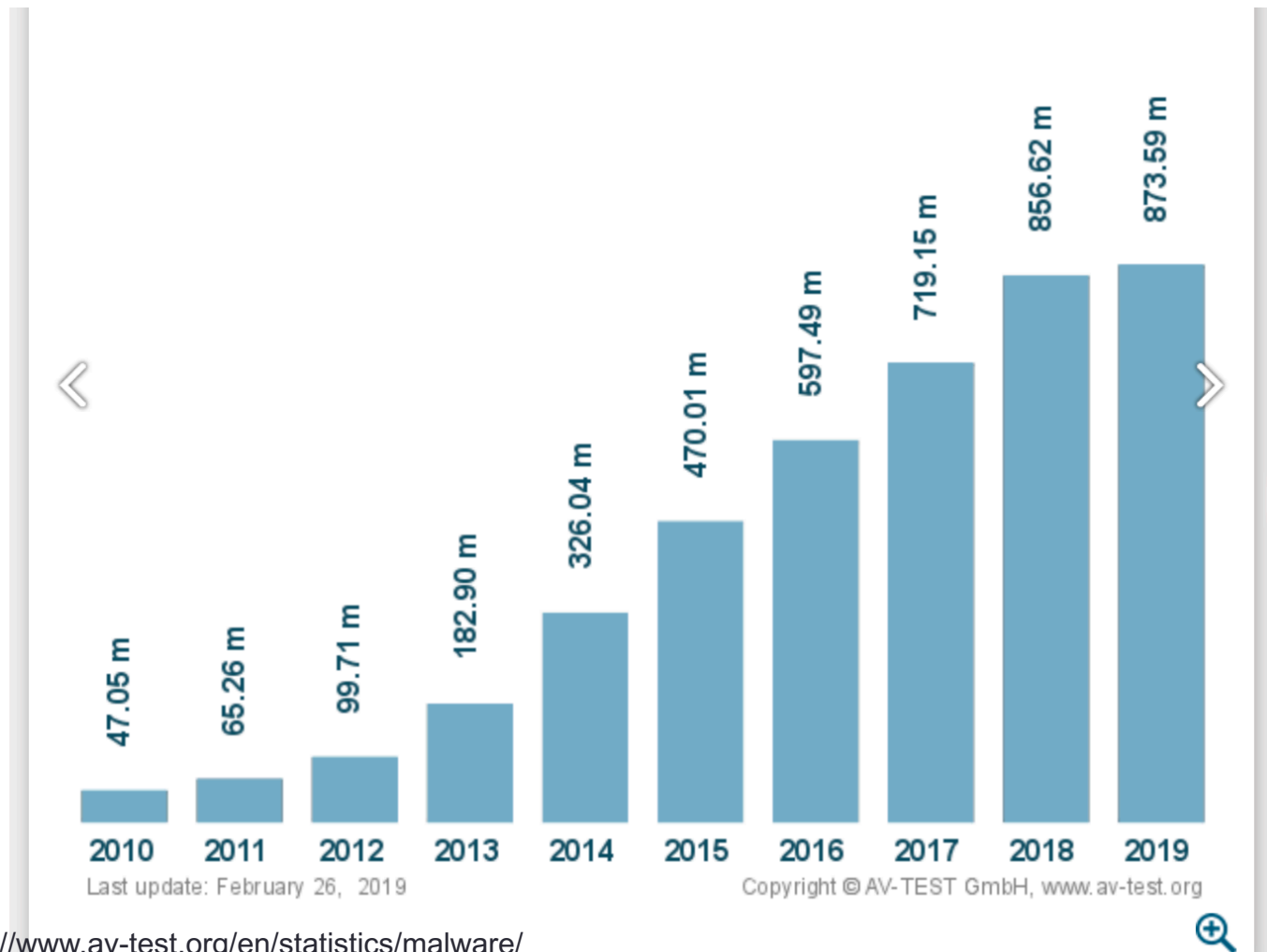


<https://www.cpni.gov.uk/insider-risk-assessment>

Financial Impact

- Malware often affects a large user population
- Significant financial impact, though estimates vary widely, up to \$100B per year (mi2g)
- Examples
 - LoveBug (2000) caused \$8.75B in damages and shut down the British parliament
 - In 2004, 8% of emails infected by W32/MyDoom.A at its peak
 - In February 2006, the Russian Stock Exchange was taken down by a virus.

Malware Growth



<https://www.av-test.org/en/statistics/malware/>

Professional Malware

- Growth in professional cybercrime and online fraud has led to demand for professionally developed malware
- New malware is often a custom-designed variations of known exploits
 - => malware designer can sell different “products” to his/her customers.
- Like every product, professional malware is subject to the laws of supply and demand.
 - Recent studies put the price of a software keystroke logger at \$23 and a botnet use at \$225.

Image by User:SilverStar from <http://commons.wikimedia.org/wiki/File:Supply-demand-equilibrium.svg>
used by permission under the *Creative Commons Attribution ShareAlike 3.0 License*

Professional Malware

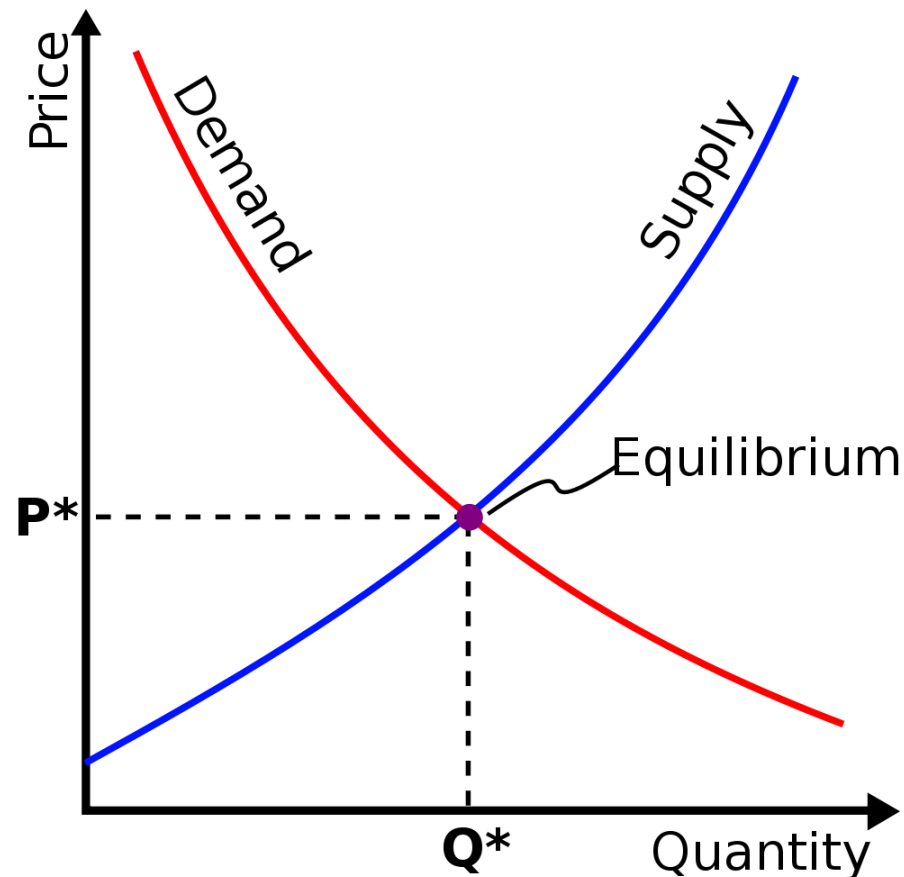


Image by User:SilverStar from <http://commons.wikimedia.org/wiki/File:Supply-demand-equilibrium.svg>
used by permission under the *Creative Commons Attribution ShareAlike 3.0 License*

- Questions?

