University of BRISTOL

COMS20012: Basic Systems Security Concepts

Joseph Hallett

bristol.ac.uk

Welcome to Computer Systems B!

- In Computer Systems A you covered, *concurrency* and *distributed systems*…
- In Computer Systems B we cover *operating systems* and *security*!

Who am I?

Joseph Hallett

- I do developer-centered security

Other lectures by *Alma Oracevic*, *Sridhar Adeptu*, *Sana Belguith*.

- We are all lecturers in the Bristol Cyber Security Group (BCSG).

What is all this about then?

Computers do not exist in isolation

- We connect them to other systems via *networks*.

Software is unreliable:

- It's buggy,
- You can't trust who wrote it.

So how are we going to fix all this?

Operating systems!

- Mediate access to resources,
- Enforce isolation,
- Provide security boundaries,

More generally, provide *security*.

(Get ready for a lot of definitions…)

What is *Security*?

The act of protecting the **security properties** of system from harm:

- against a specific adversary with known (or unknown) powers
- in a specific environment
- for a specific period of time
- and having a plan for what needs to happen when it all goes wrong

Security Properties (CIA-triad)

Top-level properties that we want to maintain in a system (variations exist!)

Confidentiality

avoiding unauthorized disclosure of information

Integrity

avoiding unauthorized modification of data

Availability

avoiding periods where authorized users can't use a system

Tools for Confidentiality

To avoid unauthorized disclosure:

Encryption If you can't read it you can't disclose it.

Access control If the OS won't let you open the file you can't disclose whats inside it.

Authentication Mechanisms for determining if you're authorized to complete an action

> *Something you know, something you have, something you are.*

Physical security Lock the computer in a Faraday cage in the basement.

Tools for Integrity

To spot unauthorized modification:

Backups Just go check some data hasn't changed.

Checksums / Error correction codes Small values that change if the data changes.

To prevent unauthorized modification:

Encryption You can't modify what you can't write.

Access control You can't modify what the OS won't let you modify.

Physical security You can't modify what you can't physically get to.

Tools for Availability

To make sure you can always use the computer when you need it:

Redundancy

Duplicate systems/disks to deal with busy periods/failures.

Limits

Limit the amount of resource any single person/process can use

There are other security properties too!

AAA Authentication, Authorization and Accountability

- Tools include: identity management, access control and logging.

AAA Assurance, Authenticity and Anonymity

- Tools include: access control (policies), digital signatures, differential privacy, proxies and pseudonyms.

So how are we attacking these security properties?

Security isn't absolute!

- You protect the properties of a system *from* something

So what are the things that we want to protect against?

STRIDE

Framework developed at Microsoft for identifying threats to a system:

Spoofing: Can someone pretend to be someone else? (Attacks *authenticity*).

Tampering: Can someone modify things without authorization? (Attacks *integrity*).

Repudiation: Can someone deny authorship? (Attacks *accountability/assurance*).

Information Disclosure: Can someone learn something they're not supposed to? (Attacks *confidentiality*).

Denial of Service: Can someone attack the availability of a system?

Elevation of Privilege: Can someone do more than they should be able to (Attacks *authorization*).

Games to help you think about threats…

Analyzing a system for weaknesses is part of a larger security process called *threat modelling* that is the basis for all modern security engineering.

There are games that can help you learn to do this!

For example:

Elevation of Privilege

https://github.com/adamshostack/eop

Decisions and Disruptions

https://www.decisions-disruptions.org

Threats vs Weaknesses vs Vulnerabilities

(This is my pet hate…)

- Just because a system is under attack **does not** mean that a security property will be violated (or that we care).
- Just because a system is poorly designed **does not** mean that it will be possible to violate a security property by exploiting that weakness (defence in depth).

But…

- If there is a threat…
- …And there is a weakness…
- …And that weakness is exploitable…

Sometimes there is a vulnerability

Further reading

Common Weakness Enumeration

https://cwe.mitre.org

Common Vulnerability Enumeration

https://cve.org