

ECE/CS 230

Computer Systems Security

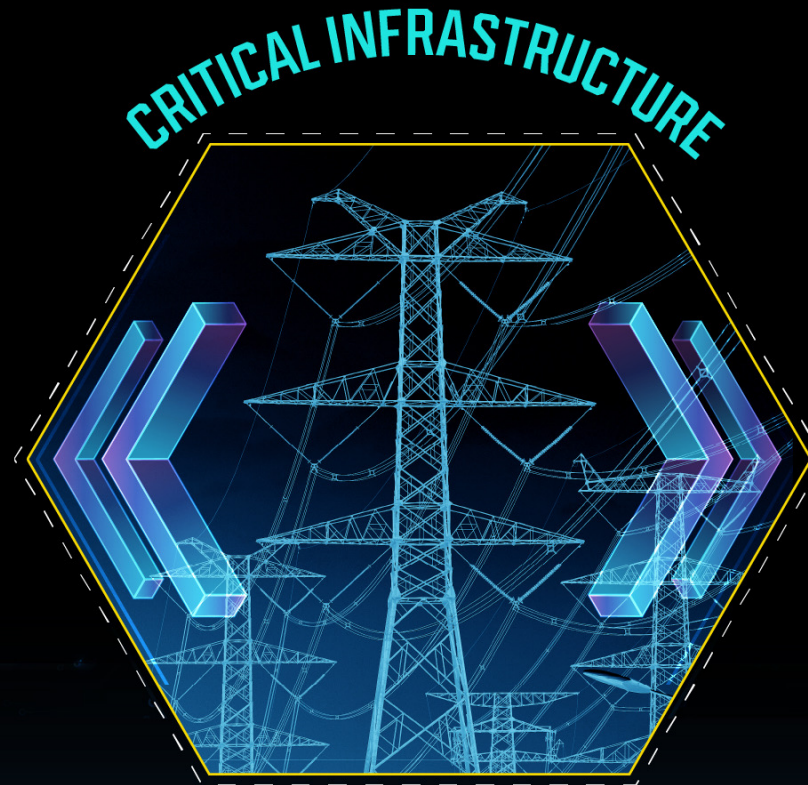
Charalambos (Harrys) Konstantinou

<https://sites.google.com/view/ececs230kaust/>

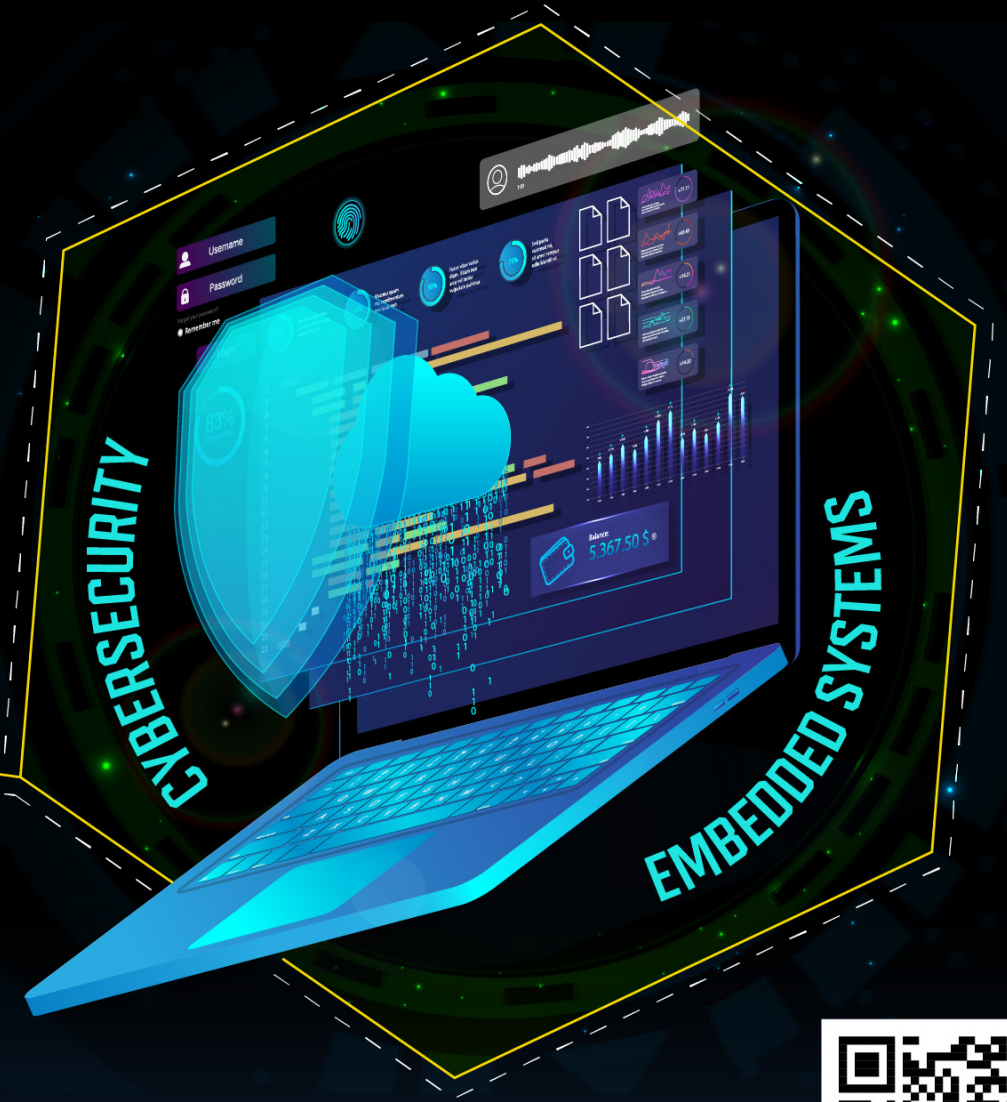
Logistics, Intro, Mindset, Ethics, Principles, Threat modeling

SENTRY

Secure Next Generation Resilient Systems Lab



CRITICAL INFRASTRUCTURE



CYBERSECURITY

EMBEDDED SYSTEMS



KAUST

RESILIENCE

Our mission: secure + resilient systems



1. Logistics

Resources

- Course information: slides, readings, assignments

<https://sites.google.com/view/ececs230kaust>

- Any other course material (discussions, announcements, etc.):

<https://blackboard.kaust.edu.sa/>

Course Description

Computer systems are essential in every part of our personal and professional life (e.g., online banking, social networking, etc.). These tasks can however expose the users to various security threats (e.g., credit card number theft, personal information leakage). Therefore, there is a need for designing secure computer systems. This course teaches both theoretical and practical concepts of cybersecurity.

The course will cover an introduction to the most important features of computer security, including topics such as symmetric ciphers, public key cryptosystems, digital signatures, hashes, message authentication codes, key management and distribution, authentication protocols, vulnerabilities and malware, access control, network security. The class will provide students with the necessary knowledge for designing secure computer systems and programs and methods for defending against malicious threats (e.g., viruses, worms, denial of service).

Objectives

- At the end of the course, the students will be able to (1) distinguish the broad set of technical aspects of cybersecurity, (2) be able to describe the vulnerabilities and threats posed by cyber-criminals to computer systems and supporting infrastructure, (3) explain common vulnerabilities in computer systems/programs, including buffer overflow vulnerabilities, time-of-check to time-of-use flaws, incomplete mediation, (4) know most theoretical concepts in the area of computer security including security principles, threat modeling, cryptography, access control), (5) apply theoretical concepts in practice by using a programming language to implement attacks and defenses in systems (e.g., operating systems, networks).

Prerequisites

Students are expected to enter this course with a background in computer systems and programming knowledge in C & Python and Linux systems, as well as basic knowledge of operating systems and data structures. Some knowledge of assembly and compilers will be helpful, but the relevant information will be covered in the course or in provided references.

Main References:

- **Instructor's lecture notes and handouts.**
- C. Pfleeger, S. Pfleeger, J. Margulies, “Security in Computing” 5th Edition, Prentice Hall, ISBN-13: 978-0134085043, Pearson; 5th edition 2015.
 - *Available as an e-book with your KAUST credentials [here](#).*
- The lectures may not be compatible with the textbook. Reading literatures coupling with the course content will be posted online, as well as the slides.

Office Hours

- Office hours:
 - Availability: By appointment
- TAs:
 - Rana Alahmadi & Li Zhou [TA hours: Tuesday 1-3pm (B1-L4)]
 - Read this: <https://sites.google.com/view/ececs230kaust/ta-policy> !!!
- Office:
Building 1, #4414

Evaluation & Assignments

Method of Evaluation

- 10% Online Active Participation
- 35% Assignments
- 20% Midterm exam
- 35% Final exam

Nature of Assignments

- Online Active Participation: A topic will be assigned weekly on Blackboard and students are required to participate in the discussion boards.
- Assignments: Assigned every other week and due one week after. They can be paper and pen questions and/or programming exercises. Some of the assignments will depend on previous assignments. Students will need to program in C and/or Python.
- Midterm exam: In-class open-note exam.
- Final exam: The final exam will be an open-notes exam, covering material from the whole semester, with emphasis on material covered since the last midterm.

Course Policies

- For the assignments students are expected to work independently. Offering and accepting solutions from others is an act of plagiarism, which will be penalized. Discussion among students is encouraged, but when in doubt, students should direct their questions to the professor or teaching assistant.
- A topic will be assigned weekly on Blackboard and students are required to participate in the discussion boards. Participation is: 1) answering questions posed in the topic description, 2) answering questions posed by other students or the instructor, 3) posting interesting/insightful summaries on articles that pertain to the weeks coursework but not necessarily have to be on the topic. Participation is not: 1) simple two sentence responses, 2) linking to articles, 3) copying and pasting.
- Weekly topic available every Monday 00:01 for one week (until next Sunday 23:59). Topic 1 will be available Monday Sep. 4.

Course Policies

- Midterm and final exams will be open-notes meaning that students can consult the following course materials: slides, handouts (including labs and assigned readings), lecture notes. Anything else is not allowed (except linguistic dictionaries). Kindle-type readers without internet access are allowed to avoid printing slides, etc.
- All methods of evaluations are required. Students who do not show up for an exam or do not provide any assignment or participate in a discussion should expect a grade of zero on that item.
- Students will not receive extensions. **Late assignments (not exams) will be accepted. Students will be penalized 20% for every late day (day determined per the deadline of the submission time).**

Questions on logistics?

Who are you?

Besides responsible graduate students

What can I get from this course?

- From this course you should learn (in order of importance):
 - A basic understanding of the many facets of cybersecurity so that you can learn more about it in the future
 - How to think like a cybersecurity expert
 - The difference between security theory and security practice
 - How to differentiate between “hype” and “reality”
 - Definitions and terminology so you can converse with others about the topic

What will I NOT get from this course?

- By taking this course you will **not**:
 - become a security expert
 - learn how to write exploits and become a hacker/cracker
 - learn how to use security tools
 - get a list of best practices or things that you should do or can use
 - learn how to secure your computers at home, work or anywhere

i.e. - **NOT**

- Hack all things
- Be an 31337 h4x0r (“leet speak” for “elite hacker”)
- Save the world
- The answer is “black or white”

Student Responsibilities

- You are responsible for your grades
- Make sure all graded assignments, exams, etc. are correct
- I will not review all of your grades at the end of the semester if you think I made a mistake, this means you have to be proactive about ensuring that there are no mistakes in the grading of your exams as they are returned!
- Due dates are solid.

2. Introduction

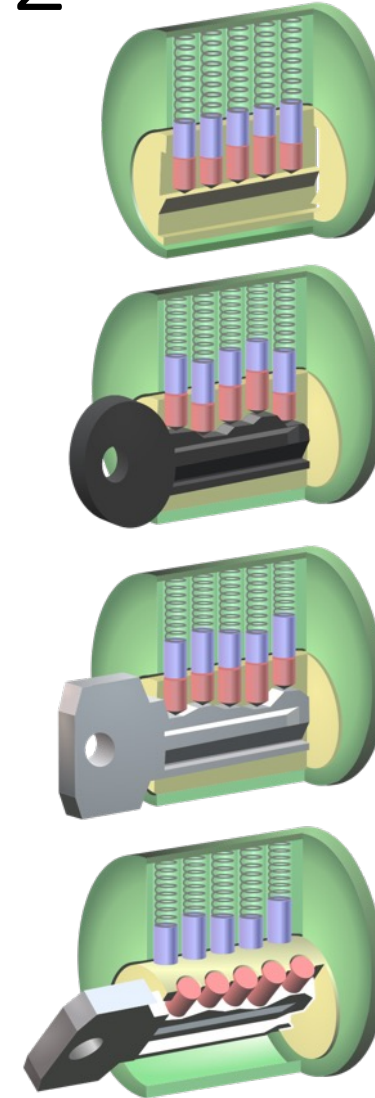
What is Information Security?

Security

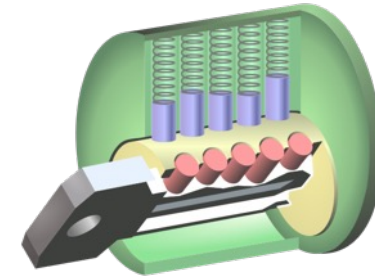
But first: Securing a building using locks

Pin Tumbler Locks – Theory 1/2

- A tumbler lock consists of:
 - A number of pin positions = n
 - A number of types of pins = m
- How many possible combinations?
 - m possibilities for each position
 - If we have 5 pin positions then
 - $m * m * m * m * m = m^5$
 - Generally: m^n
- How “secure” is the lock?



Pin Tumbler Locks – Theory 2/2



- How “secure” is the lock?
 - Lets take $m = 6$, $n = 6$
 - How many combinations? $6^6 = 46656$
 - If you have your house key, what is the likelihood that your neighbor’s lock uses the same key? $1/46656$
 - Is this secure?
 - Why?
 - Why not?
 - NYC has $\sim 3M$ households which means there are about 64 houses that uses the same key

Pin Tumbler Locks

Is it really secure?

- Although 46656 is not a large number, going from door to door to try each lock is difficult
 - It extremely time consuming: if it takes 1 minute to try the key at a single door...

*drunk man with a set of keys problem..

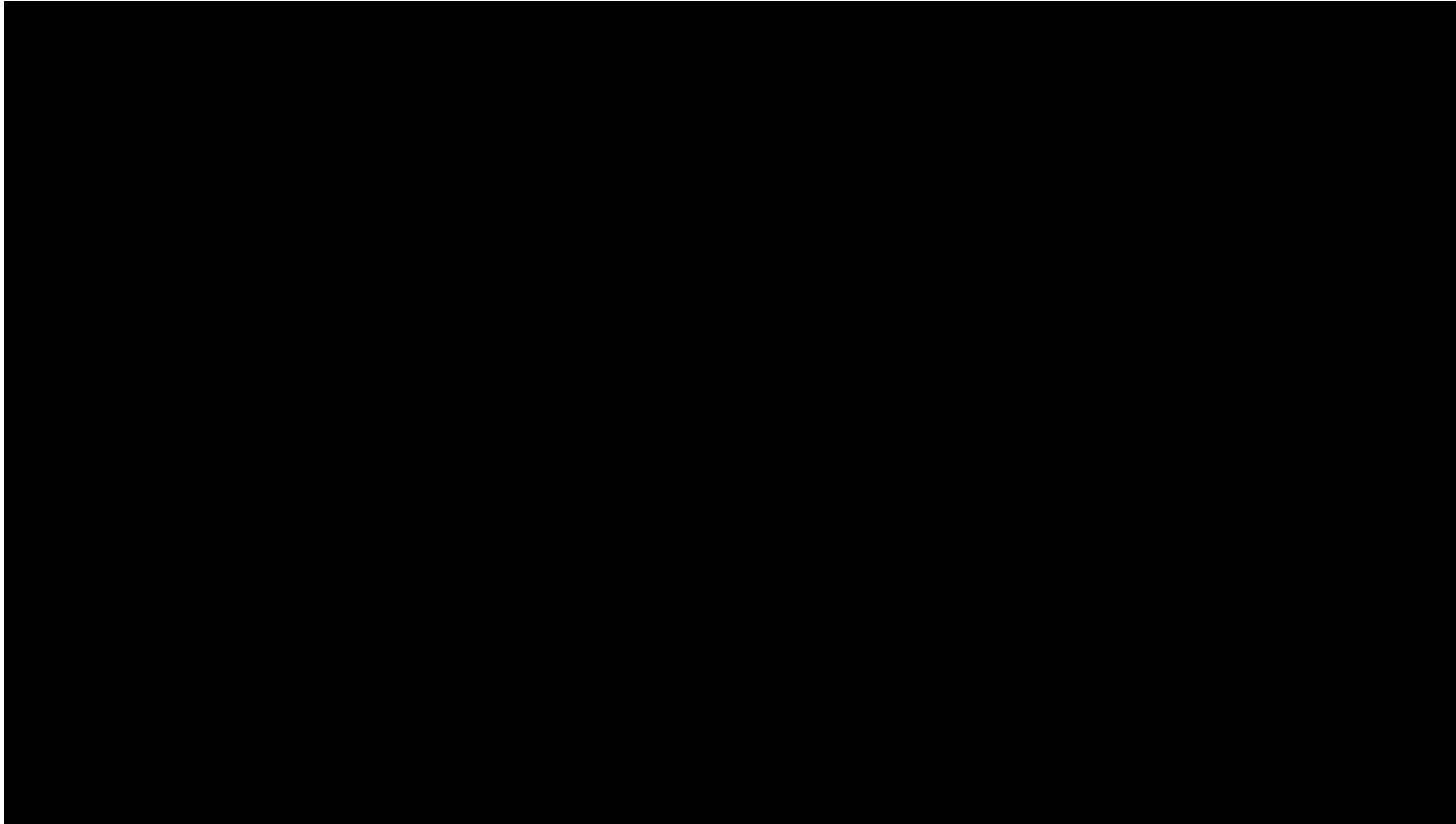
- Even if you had a month to do this: Going from door to door makes people suspicious – is this true? Maybe, maybe not. But that is not too important

Pin Tumbler Locks

What about making keys?

- Going from door to door is difficult
 - Even if I find a door that I can open, it might not be the one I want to open!
- Next question is: well can I just make keys?
 - Making keys is difficult as well
 - Making a key is time consuming
 - Given a door and the outside of the lock you don't know what key it needs
- So even if we can the lock maker can just adjust the variables: $m = 7$ would give use $7^6 = 117,649$

Lock Picking – An old video



<https://www.youtube.com/watch?v=vFXzibZhCV0>

<https://www.wired.com/2009/05/ff-keymaster>

Lock Picking video

So what did we see?

- Fastest time 1:55
- Slowest time 8:15
- Threshold = 10 minutes for “high security”
 - Notice the HUGE disparity between 10 minutes as 23 days!
- Some specialized tools were used

Pin Tumbler Locks - Reflection

- Theory and practice are drastically different
 - The same applies to security
 - We will thus cover theoretical concepts versus reality
- Locks are one of those things that we don't think about and assume is secure, but in reality it is not.
 - Knowledge of the internal mechanisms of the lock helps in understanding how to break it
 - Sometimes you just live with it – i.e. security doesn't trump everything else
- Despite the fact that locks are so easy to pick our houses are not getting robbed all the time! – Why?
 - Security is NOT just about the security of one single mechanism - the pin tumbler lock. Perhaps there is a neighborhood watch?
 - If the pin tumbler lock isn't secure enough, find another type of lock
 - If no locks are secure, then hire a guard
 - Sometimes we need to answer the question: so what if they can get in?
 - What will they get?
 - How will they get these things out?

Overall

Security is about answering all of these questions and making informed decisions

Security = Risk Management

Imagine this

- Imagine what happens if it was possible to try a lock in .1 second instead of 1 minute
 - Then it will take 55.2 minutes
- Imagine also that you don't have to physically be in front of the lock to open it
 - Then our original assumption that it would be "suspicious" is incorrect
 - Which means the neighborhood watch won't work
- Imagine also that you don't need to be physically there to steal things
 - Which means the guard won't work either
- Or if physically there, no one knows or can ever know who you are

Computer Security

- The “what ifs” in the previous slide represent the advent of the computer
 - Today’s computers are extremely fast. They are even faster with grid computing!
 - <http://www.enigmaathome.net/> for an interesting read
 - Today’s computers are highly connected – physical access is not required
 - Anonymity is easy
- Computer security is therefore about securing that same building despite all of these problems
 - Attacks are faster
 - Attackers have easy access
 - Attackers can hide themselves at will

Back to the ~~Future~~ Security

- Security is a property (or more accurately a collection of properties) that hold in a given system under a given set of constraints
 - Where a system is anything from hardware, software, firmware, and information being processed, stored, and communicated.
 - and constraints define adversaries and their capabilities.
- Can also mean the measures and controls that ensure these properties
- Security is weird, as we don't explicitly study other properties
 - Correctness
 - Performance

3. Security mindset

Meet the Adversary

- “Security studies how systems behave in the presence of an adversary.”
- The adversary
 - a.k.a. the attacker
 - a.k.a. the bad guy
- *An intelligence that actively tries to cause the system to misbehave.



“Know your enemy”

- Motives?
- Capabilities?
- Degree of access?

Thinking Like an Attacker

- Look for weakest links – easiest to attack.
- Identify assumptions that security depends on.
 - Are they false?
- Think outside the box:
Not constrained by system designer's worldview.

Practice thinking like an attacker:

For every system you interact with, think about what it means for it to be secure, and image how it could be exploited by an attacker.

Security Fail

- User: Admin
- Password: Admin



Thinking Like as a Defender

- Security policy
 - What are we trying to protect?
 - What properties are we trying to enforce?
- Threat model
 - Who are the attackers?
 - What are their Capabilities? Motivations? Access?
- Risk assessment
 - What are the weaknesses of the system?
 - How likely?
- Countermeasures
 - Technical vs. nontechnical?
 - How much do they cost?

**Challenge is to think rationally
and rigorously about risk.
Rational paranoia.**

The Security Mindset

- Thinking like an attacker
 - Understand techniques for circumventing security.
 - Look for ways security can break, not reasons why it won't.
- Thinking like a defender
 - Know what you're defending, and against whom.
 - Weigh benefits vs. costs: No system is ever completely secure.
 - "Rational paranoia!"

Schneier's law

- “Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break.”
- “What is hard is creating an algorithm that no one else can break, even after years of analysis.”
- Replace “cryptographer” with “engineer” and “algorithm” with “system” and it still holds true

So far

- **Logistics**
- **Intro**
- **Mindset**


What's next

- **Ethics**
- **Design principles**
- **Threat modeling**

Ethics

Ethics

Ethics in business
moral principles
rules and regulation
of right conduct rec
values that guide t



Incident 1

Target to pay \$18.5M for 2013 data breach that affected 41 million consumers

Kevin McCoy, USA TODAY

Published 4:10 p.m. ET May 23, 2017 | Updated 6:42 p.m. ET May 23,



Incident n

- The list can go forever
- Thank goodness for security experts



Ethics – Consumers – Ethical Issues Behind Cybersecurity

- There can't be any delays in letting customers know that a data breach has occurred and their information/money may have been stolen.
- Target data breach
 - International Business Times reported that the retailer discovered the breach Dec. 13, 2013
 - It kept its 70 million affected customers in the dark until Dec. 19 – one day after the hack was revealed in a blog post by cybersecurity reporter Brian Krebs.
 - Target lost significant customer trust.
 - Net earnings dropped 46 percent from the previous year in 2013's fourth quarter, according to The New York Times.

Security “Research” to the Rescue!

- Researchers want to help, to benefit the community
- ...but oh, the temptations!
First to publish; do something new; show how 1337^a you are; fight for funding; ends justify the means
- ...and the conflicts
Affecting other research; impacting law enforcement investigations; thwarting mitigation efforts; protecting rights; helping the bad guys; less risky options?

^a1337, which stands for "leet", short for "elite hacker" and "leetspeak" in leetspeak. Leetspeak is a form of symbolic writing that substitutes various numbers and ASCII symbols for letters.

What are ethics?

- “The field of ethics (or moral philosophy) involves systematizing, defending, and recommending concepts of right and wrong behavior.”
- Normative ethics, is concerned with developing a set of morals or guiding principles intended to influence the conduct of individuals and groups within a population (i.e., a profession, a religion, or society at large).
 - Consequentialism: Consequences are the most important consideration
 - Deontology (duty-based ethics): Following rules is most important
 - Virtue ethics: An individual’s character is more important than either actions or consequences

Philosophy 101-level ethics problem

- Situation: You've been captured along with 10 other people and your captors give you a choice: Shoot one of the 10 people yourself and everyone else lives or shoot no one and your captors will kill all 10.
- Deontological (duty-based) ethics may have a rule, "do not kill" so the ethical thing to do is kill no one (but then 10 people die)
- Consequentialism may dictate that one dead person is a better outcome than 10 dead people so the ethical thing to do is to shoot

Computer Ethics

- “A typical problem in computer ethics arises because there is a policy vacuum about how computer technology should be used. Computers provide us with **new capabilities** and these in turn give us **new choices** for action. Often, either no policies for conduct in these situations exist or existing policies seem inadequate. A central task of computer ethics is to determine **what we should do** in such cases, i.e., to formulate policies to guide our actions.”

James H. Moor

Ethics and Law

Ethics != Law

- “Law can be defined as a consistent set of universal rules that are widely published, generally accepted, and usually enforced”
- Interrelated but by no means identical (e.g., legal but not ethical, ethical but not legal)
 - Adherence to ethical principles may be required to meet regulatory requirements surrounding academic research
 - A law may illuminate the line between beneficial acts and harmful ones.
 - If the computer security research community develops ethical principals and standards that are acceptable to the profession and integrates those as standard practice, it makes it easier for legislatures and courts to effectively perform their functions.

Professional Ethical Codes

- IEEE Code of Ethics (2006)
 - commits members “to the highest ethical and professional conduct”. Members agree to avoid conflicts of interest, be honest, engage in responsible decision making, accept criticism of work, etc.
- ACM Code of Ethics and Professional conduct (1992)
 - “contribute to society and human well-being”, “avoid harm to others”, along with six other principles (e.g., don’t discriminate, be honest, respect privacy).

Case Study: Botnets (1/3)

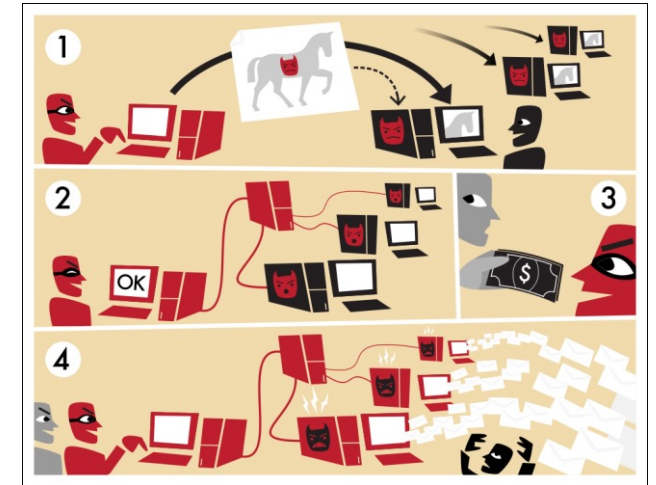
Botnets, briefly

Bots are compromised computers under the control of some 3rd party

Collection of bots comprise a botnet

Bots communicate with command & control servers which provide instructions, e.g., DDOS a host, send spam, find new machines to infect

(Almost) every major security incident today involves botnets



Case Study: Botnets (2/3)

- A researcher constructs a benign botnet out of compromised routers and uses it to measure the entire Internet; data released publicly and anonymously

Is this ethical?

What are the potential issues?

Case Study: Botnets (3/3)

- “While it probably was not technically legal, no harm was done and it produced something with educational value. The information was shared, the techniques and methods were shared... nothing was hidden (assuming everything is as it has been portrayed). Therefore, I **can't see anything unethical about this.**”
- “Objectively, the research gets a plus from me. It does however raise a small **red flag about our view of privacy on an ethical** and moral level.”
- “No matter how useful or interesting the data, is it ethical to use it? **I'm not sure.**”

1a. Definitions

Definitions

Threats, Vulnerabilities and Attacks

- A *threat* to a system is any potential occurrence, malicious or otherwise, that can have an adverse effect on the assets and resources associated with the system.
- A *vulnerability* of a system is some characteristic that makes it possible for a threat to occur.
- An *attack* on a system is some action that involves exploitation of some vulnerability in order to cause an existing threat to occur.

Types of threats

- Can be classified into four broad categories
 - Disclosure - unauthorized access to information
 - Deception - acceptance of false data
 - Disruption - interruption or prevention of correct operation
 - Usurpation - unauthorized control of some part of a system
- Examples include – snooping, sniffing, spoofing, delay, denial of service, theft of computational resources...

What can possibly go wrong?

- You find a USB stick on the floor
- What should you do?



Hacking with USB sticks:

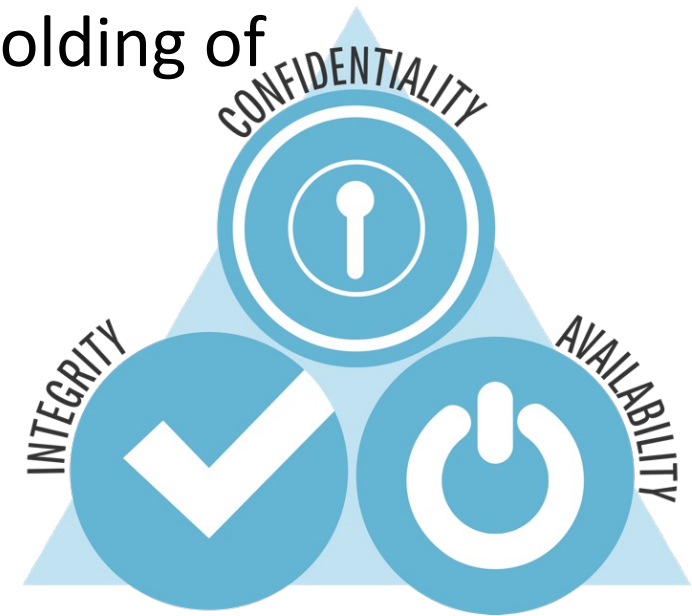
- Setup: Cyber-criminals "accidentally" leave infected USB sticks in the parking lot of the DSM (a Dutch company)
- USB sticks were infected with malware that recorded employee Username and passwords and then sent the information to remote IP addresses.

<https://it.slashdot.org/story/12/07/09/2317239/criminals-distribute-infected-usb-sticks-in-parking-lot>

Primary Issues

CIA Triad

- **Confidentiality**: prevention of unauthorized disclosure of information
- **Integrity**: prevention of unauthorized modification of information
- **Availability**: ability to withstand unauthorized withholding of information or resources



Security – Evolving Definitions

- Security: freedom from risk and danger.
- ~1980: physical security / confidentiality.
- ~1980-~2000: Integrity and access control
- ~2000-2008: Availability (DoS)
- Now: Hard to define...

Informal definitions on computer security

- Cheswik and Bellovin – “keeping anyone from doing things you do not want them to do, with, on, or from your computers or any peripheral devices.”
- Garfinkel and Spafford - “A computer is secure if you can depend on it and its software to behave as you expect ... This concept is often called trust; you trust the system to preserve and protect your data.”

Other security issues

- There are other issues that arise in the design of secure systems besides confidentiality, availability and integrity:
 - Accountability
 - Reliability
 - Access Control
 - Authentication
 - Non-repudiation
 - Privacy

Policy and Mechanism

- A *security policy* is a statement of what is, and is not, allowed.
- A *security mechanism* is a procedure, tool, or method of enforcing security policy.

Security Policy

- A security policy is a set of rules stating which actions are permitted and which are not.
- Can be informal or highly mathematical.
- If we consider a computer system to be a finite state automaton with state transitions then
 - A **security policy** is a statement that partitions the states of a system into a set of authorized or secure states and a set of unauthorized or non-secure states.
 - A **secure system** is a system that starts in an authorized state and cannot enter an unauthorized state.
 - A **breach of security** occurs when a system enters an unauthorized state.
- We expect a trusted system to enforce the required security policies.

Elements of a Security Policy

- A security policy considers all relevant aspects of:
 - Confidentiality
 - Integrity
 - Availability
- Confidentiality policy: Identifies information leakage and controls information flow.
- Integrity Policy: Identifies authorized ways in which information may be altered. Enforces separation of duties.
- Availability policy: Describes what services must be provided: example – a browser may download pages but no Java applets.

Security Mechanism

- A *security mechanism* is a procedure that enforces some part of a security policy.
- We will learn many mechanisms.

Goals of a security mechanism

- Given a policy that specifies what is “secure” and what is “non-secure” goal of security is to put in place mechanisms that provide:
 - Prevention
 - Detection
 - Recovery

Types of Security Mechanisms/controls

- Cryptography and cryptographic protocols.
- Software controls.
- Hardware controls.
- Physical controls.

Trust

- Security policies and mechanisms are based on assumptions and one trusts that these assumptions hold.
- Aspirin from drugstore is considered trustworthy. The basis of this trust is:
 - Testing and certification by FDA.
 - Manufacturing standard of company and regulatory mechanisms that ensure it.
 - Safety seal on the bottle.
- Similarly, for a secure system to achieve trust, specific steps need to be taken.

Operational Issues in Security

- Risk Analysis or Assessment
- Cost-Benefit Analysis
- Laws and Regulations
- Human Issues: usability

Some Questions

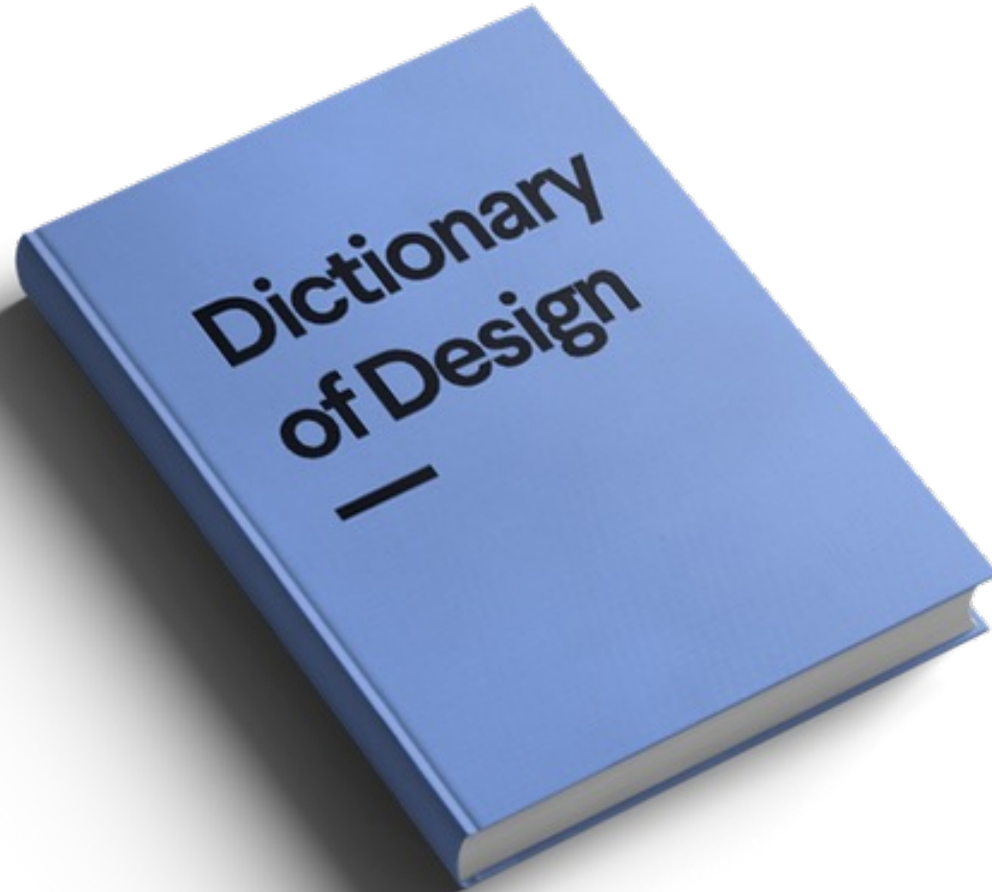
- University policy disallows cheating – copying or enabling copying of another student's homework. Student A posts her homework file online. Student B copies it. Has B violated the policy? Has A?
- Eve jams the wireless signal in the ECE department, what does she achieve?
- Your age is not public on Facebook. People can infer your age from the ages of your friends. Who has violated your privacy?

Some Hard Questions

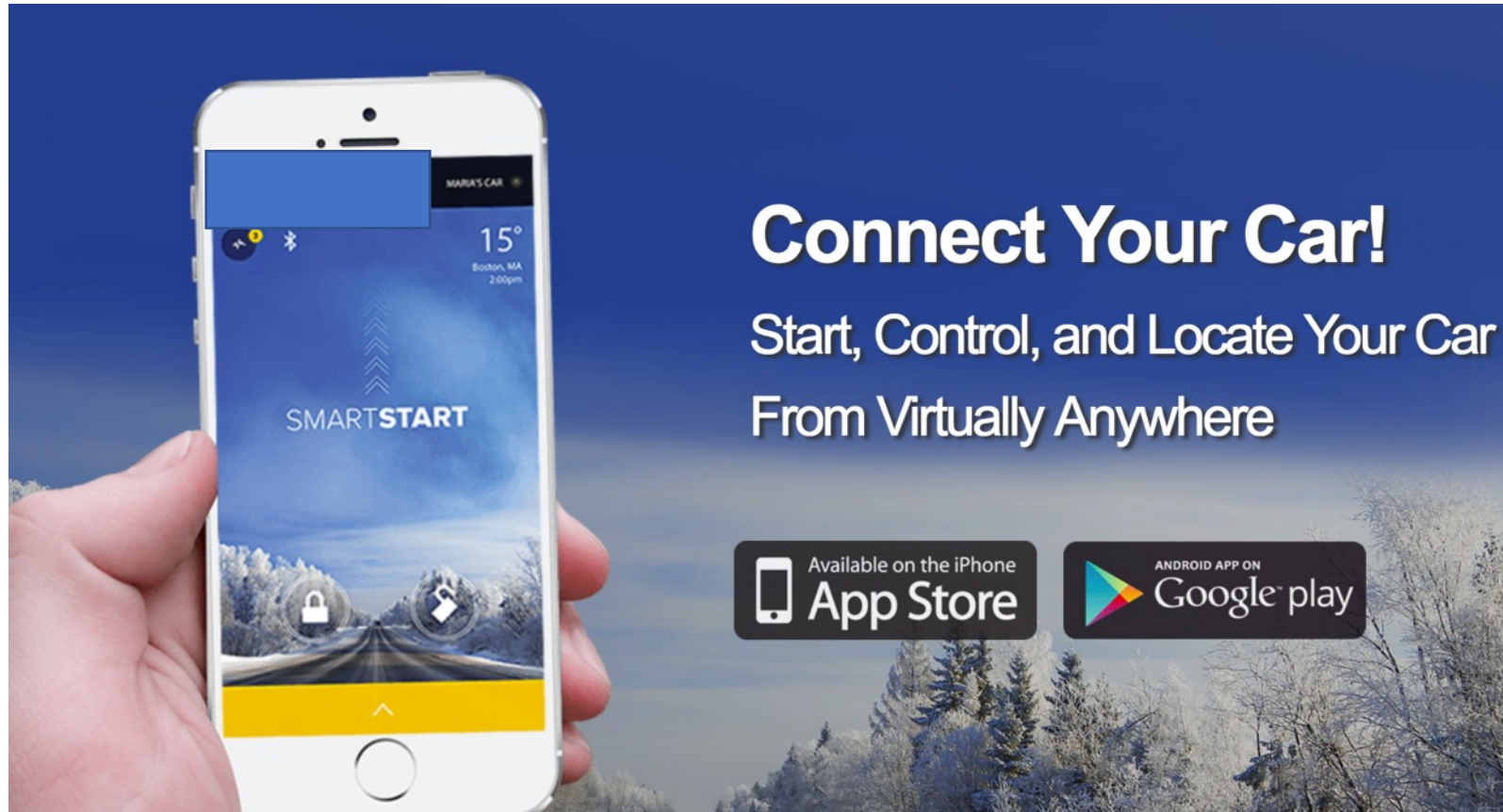
- Is it okay for me to access the Internet via your wireless access point? Does it matter if it is open vs using easily crackable encryption?
- Microsoft releases a patch for a previously unknown bug. Attackers then discover and attack the flaw. Did patching the bug improve or harm security?

2. Design principles

Design Principles



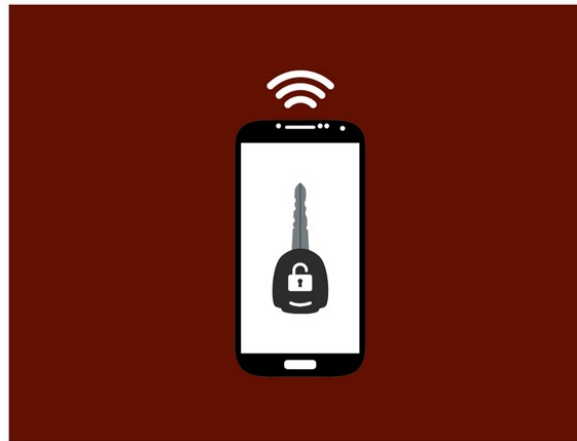
What could possibly go wrong?



What could possibly go wrong?

ANDY GREENBERG SECURITY 02.16.17 05:30 PM

ANDROID PHONE HACKS COULD UNLOCK MILLIONS OF CARS



WIRED

IN THE ERA of the connected car, automakers and third-party developers compete to turn smartphones into vehicular remote controls, allowing drivers to locate, lock, and unlock their rides with a screen tap. Some apps even summon cars and trucks in *Knight Rider* fashion. But phones can be hacked. And when they are, those car-connected features can fall into the hands of hackers, too.

<https://www.wired.com/2017/02/hacked-android-phones-unlock-millions-cars/>

Design Principles for Secure Systems

- Two basic themes:
 - Simplicity – KISS¹
 - Makes design and interactions easy
 - Easy to prove its safety
 - Restriction
 - Minimize the power of entities
 - There are no “laws” of security
 - Know the basic ideas
 - Use these to help you reason about security
- ¹KISS is an acronym for “Keep it simple, stupid” as a design principle noted by the U.S. Navy in 1960.

Principles of design

1. Principle of least privilege
2. Principle of fail-safe defaults
3. Principle of economy of mechanism
4. Principle of complete mediation
5. Principle of open design
6. Principle of separation of privilege
7. Principle of least common mechanism
8. Principle of psychological acceptability

Principle of least privilege

- Entity should be given only the information / privileges needed to finish a task
 - Temporary elevation of privilege should be relinquished immediately
 - Granularity of privileges
 - Append permission only for logging process.
 - Strong privacy implications.

Principle of fail-safe defaults

- Use sane defaults. The default should be secure.
 - Default access to an object is none
 - Access Control Lists (ACLs), firewall examples.
 - Restricting privileges at the time of creation
 - What if the attacker's goal is to cause denial- of-service?
- “Fail-closed” (as opposed to "fail-open")

Principle of economy of mechanism

- Security mechanisms should be as simple as possible.
 - Fewer errors
 - Testing and verification is easy
 - Assumptions are less
- “Minimizing the Trusted-Computing Base”

Principle of complete mediation

- All accesses to objects should be checked to ensure they are allowed.
 - UNIX file descriptor
 - DNS cache poisoning.
 - Restrict caching policies
 - Security vs. performance issues

Principle of open design

- Security of a mechanism should not depend upon secrecy of its design or implementation (why not?)
 - Secrecy \neq security
 - Complexity \neq security
 - “Security through obscurity”
 - Cryptography and openness

Principle of separation of privilege

- System should not grant permission based on single condition
 - Company checks over \$75,000 to be signed by two officers.
 - Example: “su” on BSD requires
 - User be in group “wheel”
 - User knows root password
 - Restrictive because it limits access
- “Don't put all of your eggs in one basket”

Principle of least common mechanism

- Mechanisms used to access resources should not be shared
 - Shared resources need resource isolation to prevent becoming a denial-of-service target
 - Restrictive because it limits sharing

Principle of psychological acceptability

- Security mechanism should not make the resource difficult to access
- Recognizes the most important element in security: **HUMAN**
- “Usability vs security”

Example

- Viruses cause havoc on PCs because, any program or script that is downloaded or received as email attachment, runs with the privileges of the user that runs them. Or worse the privileges of the administrator.

- What is the problem?
- What design principles are being exploited?



Principle of least privilege

Principle of fail-safe defaults

Principle of economy of mechanism

Principle of complete mediation

Principle of open design

Principle of separation of privilege

Principle of least common mechanism

Principle of psychological acceptability

Example

- Openssl has >0.5 million lines of code. This code is **trusted**, largely for performance reasons and to make the standard library programmers' lives easier.

- What design principles are in play?



Result

What is the Heartbleed bug, how does it work and how was it fixed?

The mistake that caused the Heartbleed vulnerability can be traced to a single line of code in OpenSSL, an open source code library. Here's what you need to know now.

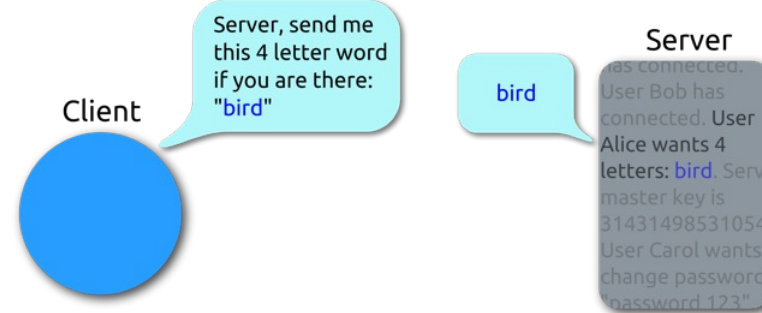


By **Josh Fruhlinger**

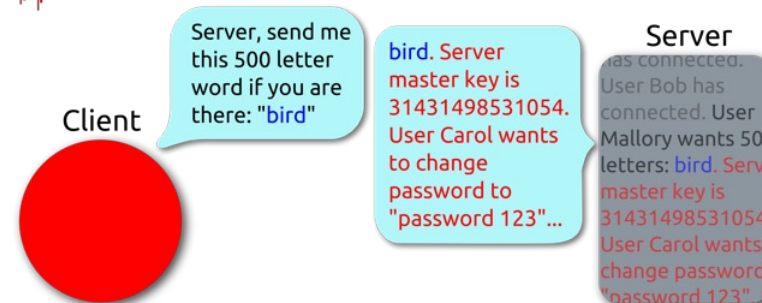
CSO | SEP 13, 2017 2:53 AM PT



Heartbeat – Normal usage



Heartbeat – Malicious usage



Example

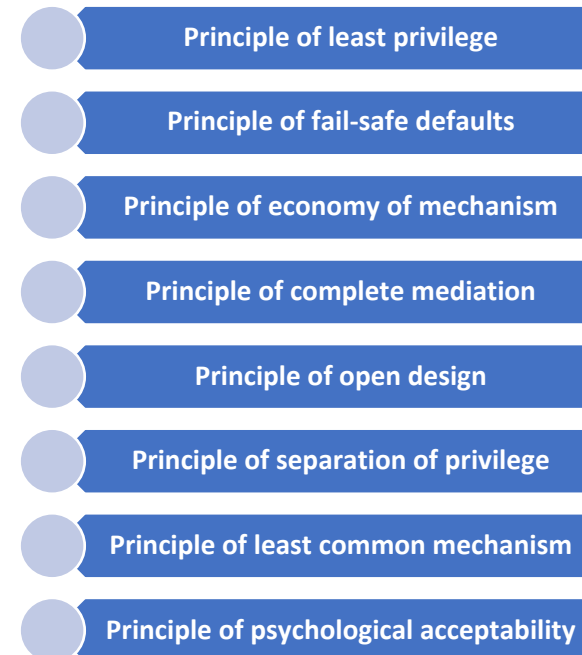
- In Python, if you use the default library to open an HTTPS connection, by default it does not check that the server certificate possesses a chain of signatures to a root- of-trust.
- What principle is being violated?



Example

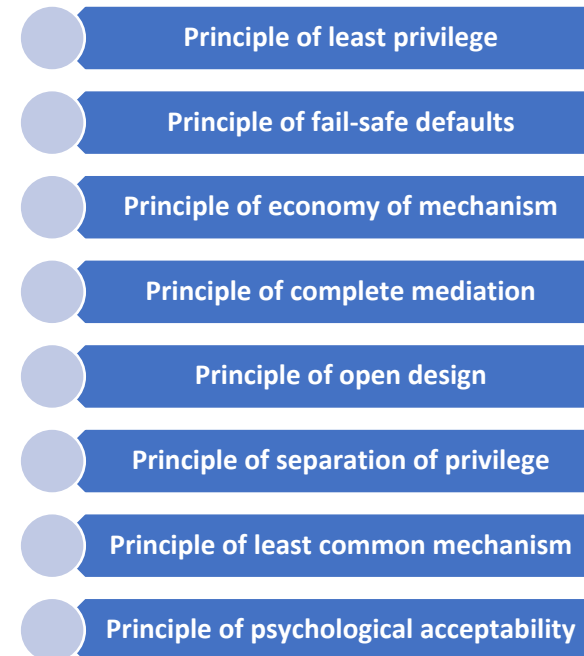
- TLS defines a mandatory server side certificate and an optional client side certificate. Though highest level of security is achieved using client and server side certificates, client side keys did not become very popular because of administrative overhead (Installation, expiration of client side certificates).

- What principle is being violated?



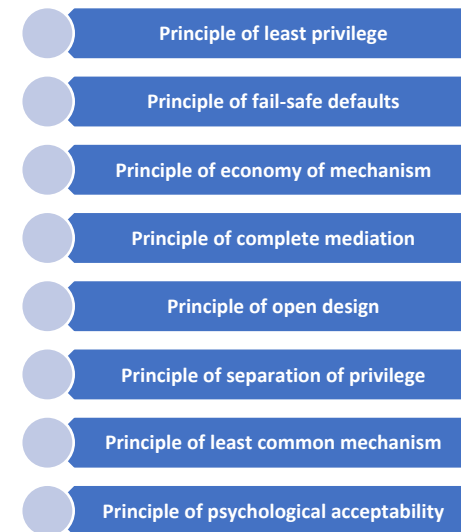
Example

- The Stork package manager shares immutable copies of installed packages across OS VMs. It reduces duplicate package downloads between VMs and saves disk space, network bandwidth, and even memory.
- Which of the above principles does Stork follow or violate?



Example

- The Stork package manager shares immutable copies of installed packages across OS VMs. It reduces duplicate package downloads between VMs and saves disk space, network bandwidth, and even memory.
- Which of the above principles does Stork follow or violate?
- Stork violates the principles of least common mechanism and least privilege to prevent duplicate downloads. How would this impact the threat from a man-in-the-middle attacker?



3. Threat modeling

Security Life Cycle

- So far what we have learnt helps us in design, specification and implementation mainly.
 - What about others?
 - We start with threat analysis/modeling.



Why Threat Modeling

- Helps you understand your application better.
- Discover potential design flaws and vulnerabilities
- Prioritize security analysis
- Understand overall security risk
- Develop mitigating strategies
- Provide more complete analysis

Why Threat Modeling

- “My house is secure” is almost meaningless
 - Against a burglar? Against a meteor strike? A thermonuclear device?
- “My system is secure” is almost meaningless
 - Against what? To what extent?
- Threat modeling is a process to define the goals and constraints of a security solution
 - Translate user requirements to security requirements

Threat Modeling

- Threats and assets are key – vulnerabilities and attacks are only concerns if there is a threat to an asset to be concerned about.
- How do we identify and evaluate threats?
 - Arbitrary Threat or Attack Lists
 - Random and unstructured
 - Dubious completeness
 - Threat Trees or Attack Trees
 - More structured
 - Modular and Re-usable
 - Currently favored approach

Threat Modeling

- Start with questions like the following:
 - Who are my potential adversaries?
 - What is their motivation, and what are their goals?
 - How much inside information do they have?
 - How much funding do they have?
 - How averse are they to risk?
 - [Be paranoid: do not underestimate the attacker's capability; do not also ignore easy/dumb attacks]
- Then enumerate threats by stepping through each of the system's assets, reviewing a list of attack goals for each asset. Assets and threats are closely correlated.

Threat Modeling – main steps

1. Understand your system
2. Understand what assets/resources need to be protected
3. Predict who the potential attackers are against a particular asset and what are the possible (known) attacks
4. Perform risk assessment
 1. Determine what is the expected risk (quantitative or qualitative) because of an attack
5. Perform risk management: Employ security mechanisms (mitigation), if needed
 1. Determine if they are cost effective

Defining, using a threat model

- A Threat Model (TM) defines the security assertions and constraints for a product
 - Assets: What we're protecting
 - Threats: What we're protecting it against
 - Mitigations: How we're protecting our Assets
- Use TM to narrow subsequent mitigation efforts
 - Don't secure review, fuzz test all interfaces
 - Select the ones that are critical
- TM is part science, part art, part experience, part nuance, part preference
 - Few big assets vs lots of focused assets

Types of threats – Remember?

- Can be classified into four broad categories
 - Disclosure - unauthorized access to information
 - Deception - acceptance of false data
 - Disruption - interruption or prevention of correct operation
 - Usurpation - unauthorized control of some part of a system
- Examples include – snooping, sniffing, spoofing, delay, denial of service, theft of computational resources...

STRIDE Model

- In general, threats can be classified into six classes based on their effect :
 - Spoofing - Using someone else's credentials to gain access to otherwise inaccessible assets.
 - Tampering - Changing data to mount an attack.
 - Repudiation - Occurs when a user denies performing an action, but the target of the action has no way to prove otherwise.
 - Information disclosure - The disclosure of information to a user who does not have permission to see it.
 - Denial of service - Reducing the ability of valid users to access resources.
 - Elevation of privilege - Occurs when an unprivileged user gains privileged status.

Ranking Threats

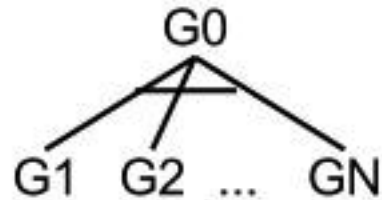
- Used for prioritizing work
- One methodology for ranking threats is the use of DREAD (used by Microsoft!)
 - Damage Potential
 - Reproducibility
 - Exploitability Cost (or cost and ease of performing attack)
 - Affected Users
 - Discoverability
- DREAD rating is calculated by adding the rating for each component
 - For example, 3: High, 2: Medium, 1: Low
 - For a particular threat, we might have
 - Damage Potential = 3
 - Reproducibility = 3
 - Exploitability Cost (or cost and ease of performing attack) = 2
 - Affected Users = 2
 - Discoverability = 2
 - Total Rating: 12, which might be regarded as High, since one can set 12–15 as High, 8–11 as Medium, and 5–7 as Low risk.

Attack Trees

- Data structure to represent an attack
- Look at system from attackers point of view.
- The root node of the tree is the global goal of the attacker
- Children are refinements of this goal
- Nodes can be conjunctive (AND) or disjunctive (OR)

Notations for nodes

- Can be represented graphically or textually
- Conjunctive (AND) node
 - To achieve G_0 , you must achieve G_1 AND G_2 ... AND G_N

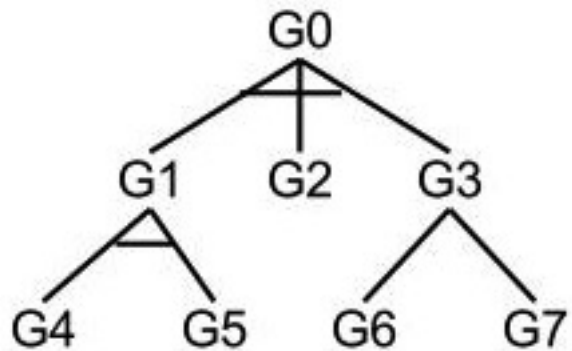


- Disjunctive (OR) node
 - To achieve G_0 , you must achieve G_1 OR G_2 ... OR G_N



Attack Trees

- Attack trees consist of any combination of conjunctive and disjunctive nodes.
- Individual intrusion scenarios are created by depth first traversal.



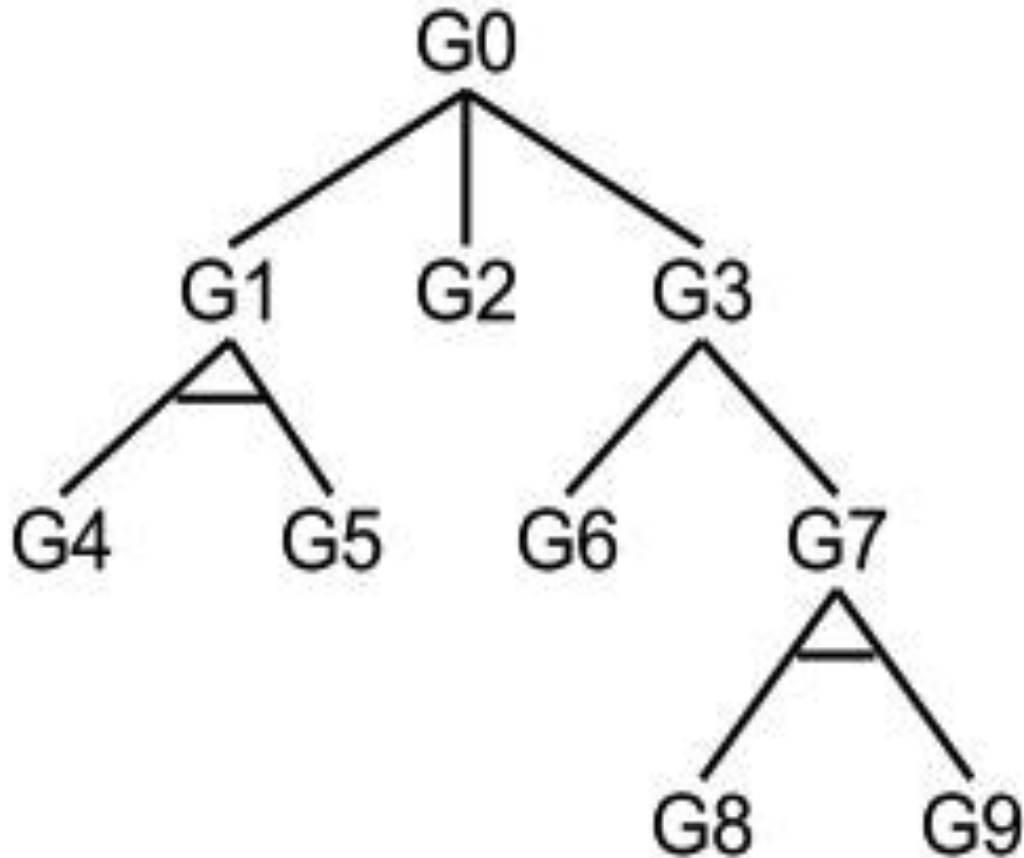
So the tree to the left leads to the attack scenarios:

<G4, G5, G2, G6>

<G4, G5, G2, G7>

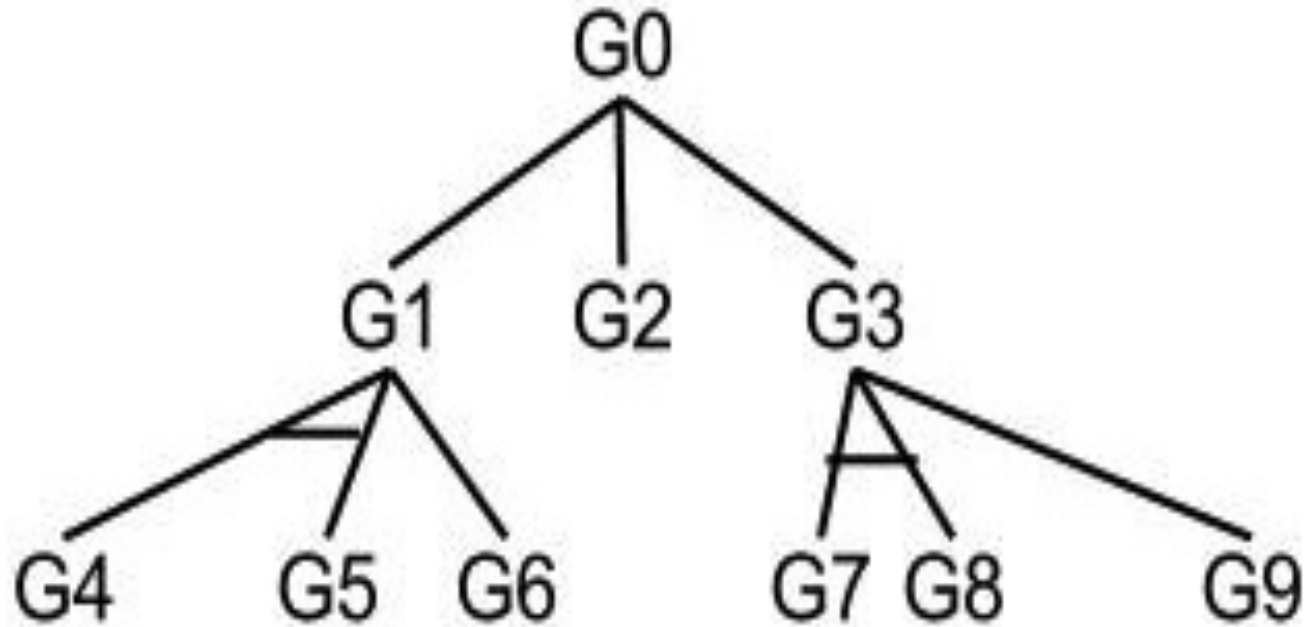
Another Example

- What are the attack scenarios for the tree below?

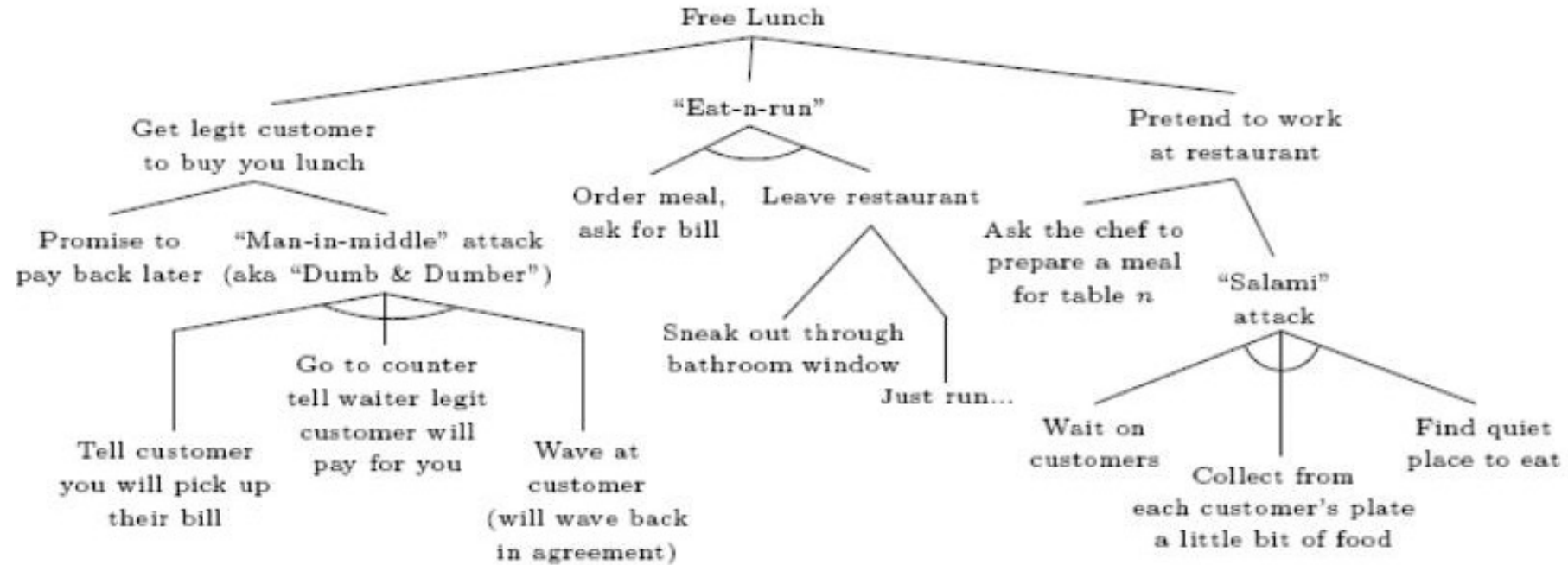


Yet another...

- What are the attack scenarios for the tree below?

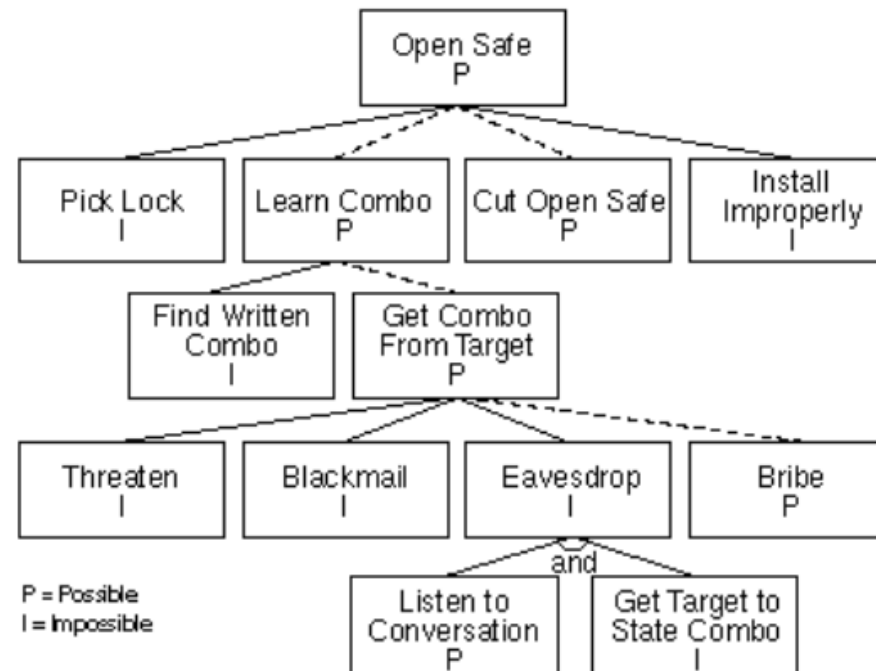


Attack Trees – a funny example



Attributes: Boolean

- You can assign attributes to nodes in the tree to help you reason about them
 - Can be useful in understanding what sorts of attackers can launch certain attacks
- “Possible” and “Impossible” are one way to assign attributes to the tree

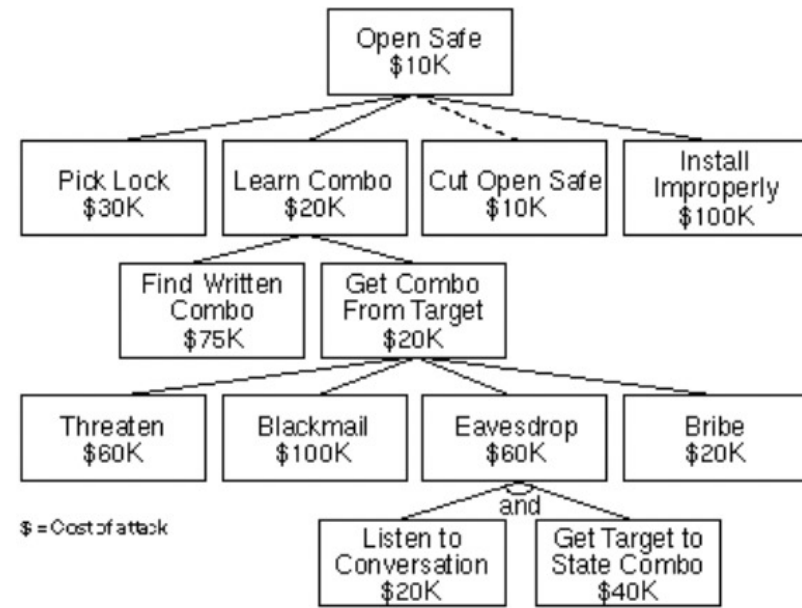


Attributes: Boolean

- “Possible” and “Impossible” are only one way to assign attributes to the tree
- Any Boolean value can be assigned to the leaf nodes and then propagated up the tree structure: AND/OR of the children node values
 - Easy vs. hard
 - Expensive vs. inexpensive
 - Legal vs. illegal
 - Special equipment Vs no special equipment

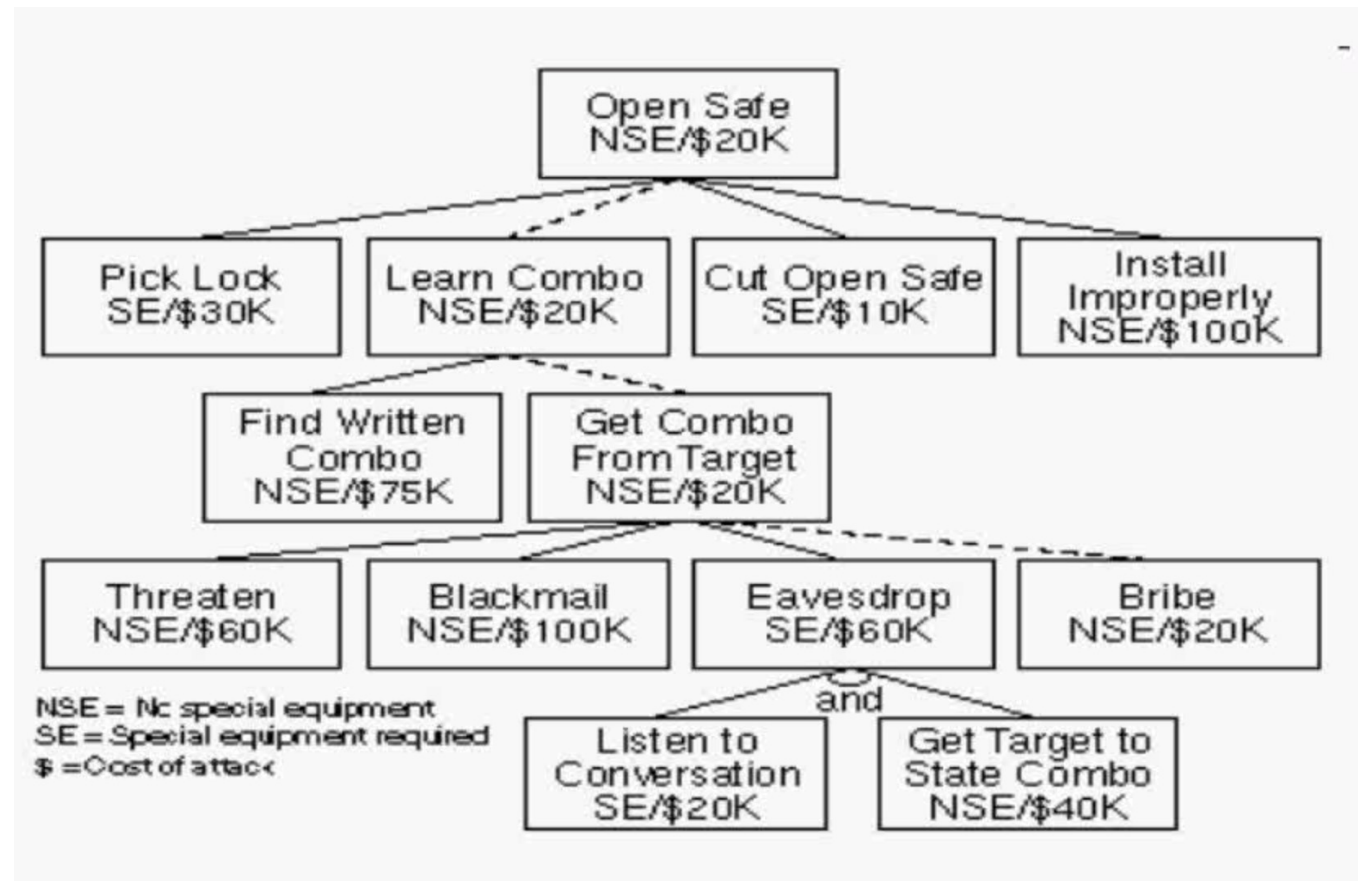
Attributes: Continuous

- Expensive vs. Inexpensive is fine, but good to say the amount, e.g.
- Continuous values can also be assigned to the nodes of the attack tree, and can be propagated up the tree
 - OR nodes have the value of their cheapest child
 - AND nodes have the value of the sum of their children



Combination of attributes

- Cheapest Attack with no Special Equipment



Risk Assessment and Management



Risk

- Risk: What (adverse) happens if a threat occurs?
 - Risk can exist when there is a known issue that increases the attack surface. Risk can also exist when there are non-specific issues, unexplored threat areas, or lack of depth-of- knowledge.
- An essential component of cybersecurity is risk analysis and risk management.

General Concept of Risk Assessment and Management

- A risk consists of something of value (an “asset” at risk) which may lose value if a negative event occurs.
 - Example: a car and its passengers are at risk in the event of an auto accident. Other people, cars, and roadside objects are also at risk
 - Example: Money invested in a stock is at risk in the event that the price of the stock goes down and the owner has to sell
- Risk analysis/assessment is the process of
 - Identifying the assets at risk (cost of asset – cost of most expensive attack)
 - Putting quantitative (e.g., dollars) or qualitative (e.g. low/medium/high) measures on the potential loss (impact)
 - Putting quantitative (i.e., the probability) or qualitative (e.g. low/medium/high) measures on the likelihood of the event happening
- Risk Management is a process for planning on how to control those risks

Example: Driving risk

- Assets at risk: people's lives and health, the automobile, other property
- Negative event: auto accident
- Risk Management:
 - Risk reduction (reduce likelihood): Following laws, defensive driving techniques, ABS, driving slow or just not driving on snowy days
 - Risk mitigation (reduce damage): Seat belts, air bags, "crumple zones" in auto design
 - Risk transfer (reduce cost): insurance
 - Risk acceptance: residual risk of injury, deductible on insurance

Example 1

- Example: Given a battlefield communications system. The related CIA asset is the _____ of the system, and the impact of a failure is _____.
- Example: Given a battlefield communications system. The related CIA asset is the availability and integrity of the system, and the impact of a failure is loss of life.

Example 2

- Example: Given a system that uses personal information such as name, SSN, etc. The related CIA asset at risk is the _____ of that information, and the impact of a compromise is the potential for _____.
- Example: Given a system that uses personal information such as name, SSN, etc. The related CIA asset at risk is the confidentiality of that information, and the impact of a compromise is the potential for identity theft.

Risk Assessment

- Assessment: measures of the impact of an event, and the probability of an event (threat agent exploiting a vulnerability)
- Quantitative (objective) and Qualitative (subjective) approaches both used.
- Quantitative approach:
 - Compute expected monetary value (impact) of loss for all “events”
 - Compute the probability of each type of expected loss
- Qualitative approach: use Low, Medium, High; ratings; other categorical scales

Risk Management

- Once you have risk computed for each threat you can prioritize them and for each do one of the following:
 - Accept the risk - The risk is so low or so costly to mitigate that it is worth accepting.
 - Transfer the risk - Transfer the risk to somebody else via insurance, warnings etc.
 - Reduce the risk - Remove the system component or feature associated with the risk if the feature is not worth the risk.
 - Mitigate the risk - Reduce the risk with countermeasures.
- The understanding of risks leads to policies, specifications and requirements.
- Appropriate security mechanisms are then developed and implemented, and then deployed

Threat Modeling Summary

- Enumerate assets
- Determine the threats to the system
- Perform risk assessment
- Perform risk management
 - If needed, perform risk mitigation by developing cost-effective security mechanisms

The End