Security
Threat Analysis

CS3524 Distributed Systems and Security

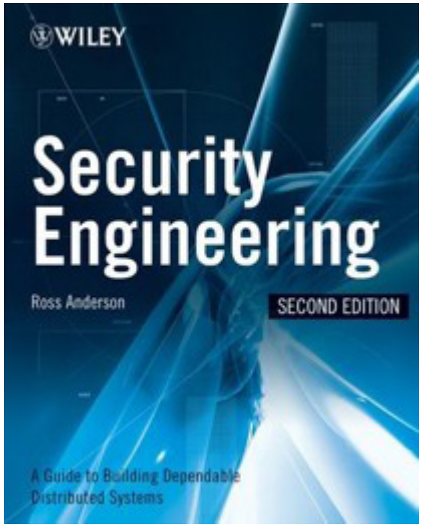
Lecture 17

Information about the Course: Tutorials

- For CS3517
  - Over the coming four weeks, there will be special 1 hour tutorials (instead of practicals):
    - 10 – 11am: FN 118
    - 2 – 3pm: FN 118
- For CS5551
  - Please see your course web site

Introduction to Security

- Threat Analysis
- Protocols



Reading:

Anderson, Security Engineering, chapters 1,2
See website for the book for these chapters:
http://www.cl.cam.ac.uk/~rja14/book.html

Reading:

Bruce Schneier, "Modeling Security Threats"
http://www.schneier.com/paper-attacktrees-ddj-ft.html

What is Security?

- Security is about ensuring that just the right agents have access to just the right resources at just the right time.
- This is a rather concise, yet rather precise statement.
- It is also rather general. That is a good thing.
- Important to understand how to apply it.
- So, what do all the words in it actually do?

Security is about ensuring that just the right
agents have access to just the right resources
at just the right times

Implicit in this is a notion of system. Also implicit is a notion of policy, which we will also explore later on. For now, let's assume we know what systems and policies are.

- Ensuring: so security is a process.
- Agents: can be people, organizations, computers, … 'legal entities'?
- Right: correct, appropriate, as intended by policy.
- Just: exactly the intended ones.
- Resources: the things consumed, created, manipulated within the system of interest.
- Times: identifiable periods during the system's execution.

More about Resources

- What is about (the system’s) resources that we care about?
- Core concepts (CIA) of information security:
  - **C**onfidentiality: the agreed/maintained degree of visibility of the resource
  - **I**ntegrity: the degree of correctness of the resource
  - **A**vailability: the degree of accessibility of the resource.
- We distinguish privacy from confidentiality. What is the difference?
- Read Anderson’s discussion of these issues.

Computer Security

- Confidentiality: access control to machines, to accounts on machines, to specific files, to networks; AV (etc.) software, firewalls.
- Integrity: avoid exposing files to overwriting, keep time-stamped back-ups, AV (etc.) software.
- Availability: redundant servers, file stores, network routing (internet robustness), AV (etc.) software.

If you don’t know how firewalls work, go and look up how malware attacks specific ports and how it is identified and blocked.

Computer Security

- Four Elements:
  - Policy/Protocol: regulations how to interact with systems, who is permitted to do what
  - Mechanisms: ciphers, access control (username / password, additional security questions), hardware that is tamper-resistant, etc
  - Assurance: how much you can rely on a particular security mechanism
  - Incentive: how to motivate personnel guarding and maintaining computer security to do their job properly (e.g. pay, and treat them well)

Network Security

- Access control again, firewalls again, of course.
- VPN, Virtual Private Networks: use SSH to create a ‘private extension’ of the target network
- Intrusion Detection Systems (**IDS**s): usually physical boxes that sit on a network and look for malware on machines or attacks in progress (e.g., DDOS attacks evidenced by abnormal flows of packets).
- Intrusion prevention systems (**IPS**s): Similar to IDSs, but can intervene in network traffic (e.g., block certain packets)

Vulnerabilities, Exploits, and Attacks

- It's a good idea to use these terms carefully.
- They apply to systems/security architectures:
  - A *vulnerability* is a potential way into a system
  - An *exploit* is way of using that way in
  - An *attack* implements the exploit and seeks to get into the system.
- An *attack vector* is the route or means by which an attack is delivered.
- The *attack surface* of a system refers to the part of the system that is (potentially) vulnerable to attacks.

Computer Security

- The goal of Software Engineering:
  - Ensure that “something is happening”: a software system should exhibit behaviour as specified
- The goal of Security Engineering”
  - Ensure that “something is not happening”: a software system should have defence mechanisms against malicious behaviour (access protection, etc)

Computer Security

- Computer security is about protection
  - Protect systems from malicious attacks
    - Avoid attacks on its operation
    - Avoid unauthorised access to data
- Issues
  - User authentication
  - Transaction integrity and accountability
  - Fault tolerance
  - Message secrecy
- Problem
  - Software designers often protect the wrong things or
  - They protect the right things in the wrong way
- How to get protection right?
  - What has to be protected?
  - How to protect it

Security-Critical Software Systems

Example: Bank

- Threats: mostly from bank personnel, operating dishonestly by using their access rights
  - account management / financial transactions / bookkeeping system
- Defence:
  - Policies: Large transfers have to be authorised by more than one person
  - Mechanisms
    - Bookkeeping procedures to make sure that money that is debited from one account, is credited to another – money only moves within a bank, never disappears
    - Alarm systems (e.g. Data mining, learning algorithms) to identify unusual volumes or patterns of transactions

Security-Critical Software Systems Example: Bank

- Automatic Teller Machines
  - Is a public interface to the bank, high vulnerability
  - Threat:
    - From inside the bank as well as from outside: e.g. “phantom withdrawals” by bank staff, utilising loop holes in security systems and procedures
  - Defence:
    - Policy: withdrawal limit
    - Mechanisms:
      - Customer ID card and personal pin number
      - Cryptography: ATM’s were the first large scale commercial use of cryptography, helped establish a number of crypto standards
      - relatively small reward, for a large risk

Security-Critical Software Systems Example: Bank

- Bank web site
  - Is a public interface to the bank, high vulnerability
  - Threat:
    - “phishing”: bogus web sites pose as a bank’s web site and ask customers for username / password
      - This is an interesting attack method, as it attacks customers and not the bank directly, it renders security technology put in place in the bank rather ineffective
  - Defence
    - “phishing” poses an interesting problem for security engineering, as a defence strategy has to take into account a mix of elements:
      - Authentication: more sophisticated authentication apart from password protection, such as hardware-based solutions, etc.)
      - Usability: customer convenience must be retained
      - Psychology: analyse behaviour patterns of customers and attackers, educate customers
      - Economics: how much to invest in defence mechanisms, or just cover the monetary loss?

Security in Information Systems

- Key points:
  - Computer Security: How much security is enough? What should be the balance of focus?
  - Threat analysis: understanding where vulnerabilities are and how they might be exploited
  - Protocols: security procedures, algorithms
    - These have to evolve as attackers get ever more sophisticated in their methods

The Dilemma

- Computer Security has to evolve and become more sophisticated:
  - Attacks get more sophisticated to deal with defences
  - Defences get more sophisticated to deal with new forms of attacks, ...
- How can we introduce security

Threat Analysis

> *Clearly, what we need is a way to model threats against computer systems. If we can understand all the different ways in which a system can be attacked, we can likely design countermeasures to thwart those attacks. And if we can understand who the attackers are -- not to mention their abilities, motivations, and goals -- maybe we can install the proper countermeasures to deal with the real threats*

Bruce Schneier, “Modeling Security Threats”
http://www.schneier.com/paper-attacktrees-ddj-ft.html

- Purpose
  - To gain an understanding of the threats and their severity
  - To appreciate from where attacks are likely to occur
  - To make decisions about defensive actions

Threat Trees / Attack Trees

- Provide a means to model security threats
  - Is a way to classify and organise threats
  - Many variations exist

Reading:

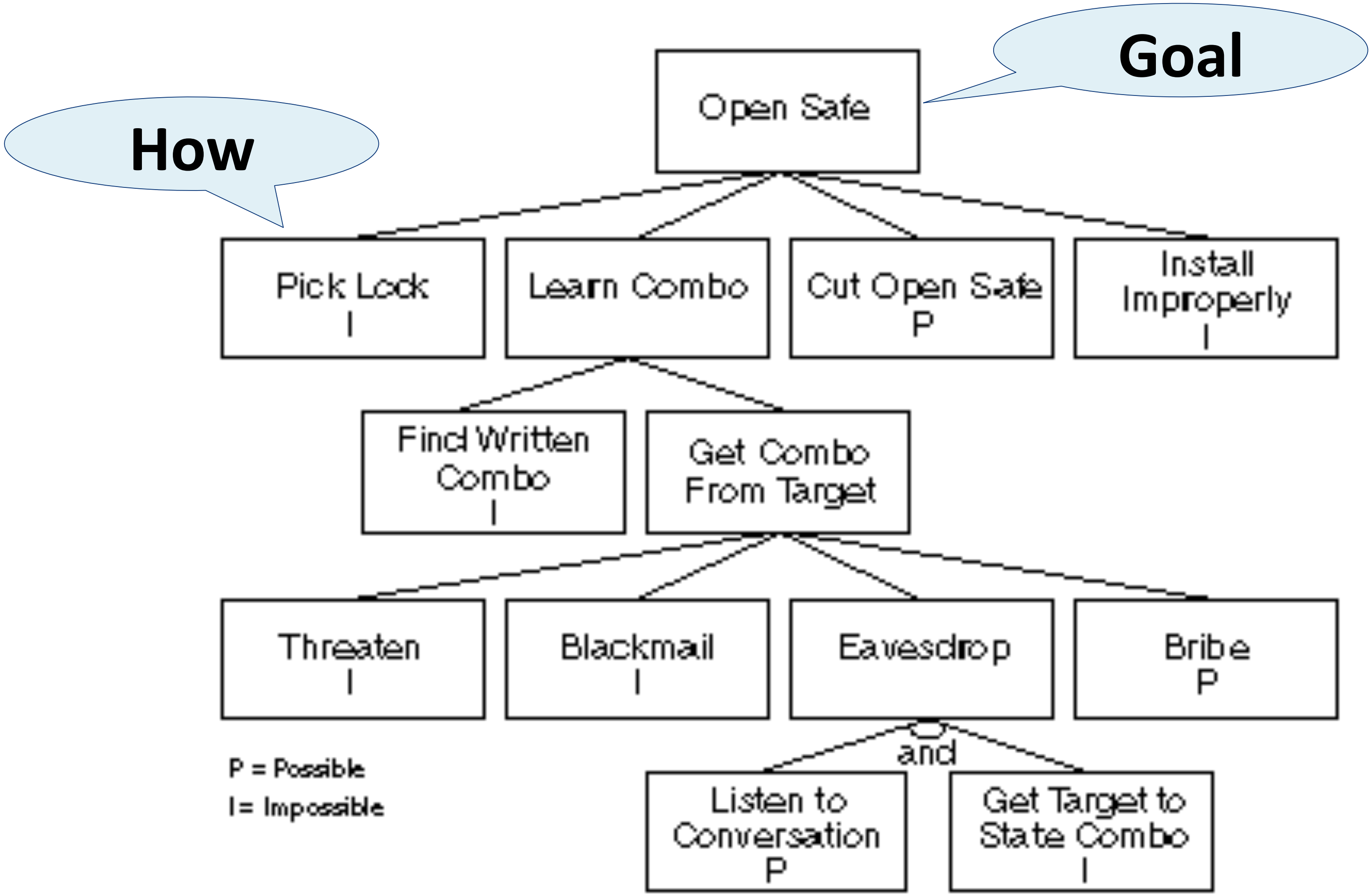
Bruce Schneier, “Modeling Security Threats”

http://www.schneier.com/paper-attacktrees-ddj-ft.html

Threat Trees

- Provide a methodical way of describing the security of systems, based on threats
- Threats are ordered into a tree-like structure
  - The root node of this tree represents the overall goal of an attacker
  - Intermediary nodes represent sub-goals for the attacker to achieve in order to achieve the overall goal
  - The leaf nodes represent the different ways or possible attacks how to achieve this goal

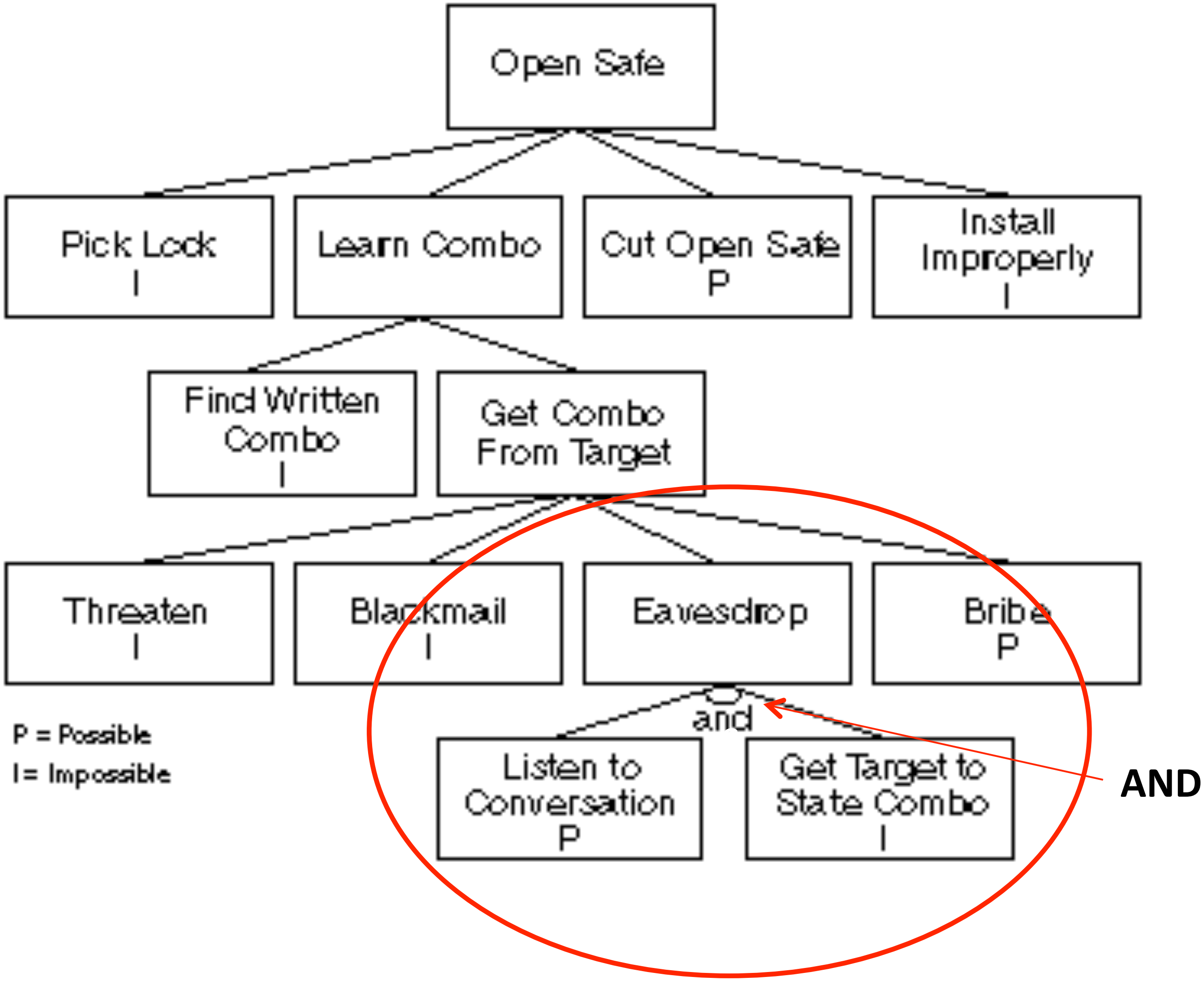
Threat Tree: Opening a Physical Safe



Threat Trees

- Threat trees have two kinds of nodes:
  - OR nodes: represent alternative attacks
  - AND nodes: represent the need that all steps (attacks) have to be taken for achieving a higher-level goal
- In the safe example:
  - AND node "eavesdrop": attackers have to listen in to conversations and get safe owner to actually say the combination

Threat Tree: AND and OR nodes



Threat Trees

- We can also describe Threat Trees more conveniently in outline form (in particular, if they become very large):

Goal: Open Save

1. Pick Lock
2. Learn Combo (OR)
   - 2.1 Find Written Combination
   - 2.2 Get Combination from Target Person (OR)
     - 2.2.1 Threaten
     - 2.2.2 Blackmail
     - 2.2.3 Eavesdrop (AND)
       - 2.2.3.1 Listen to Conversation
       - 2.2.3.2 Get Target Person to state Combination
     - 2.2.4 Bribe
3. Cut Open Safe
4. Install Safe improperly

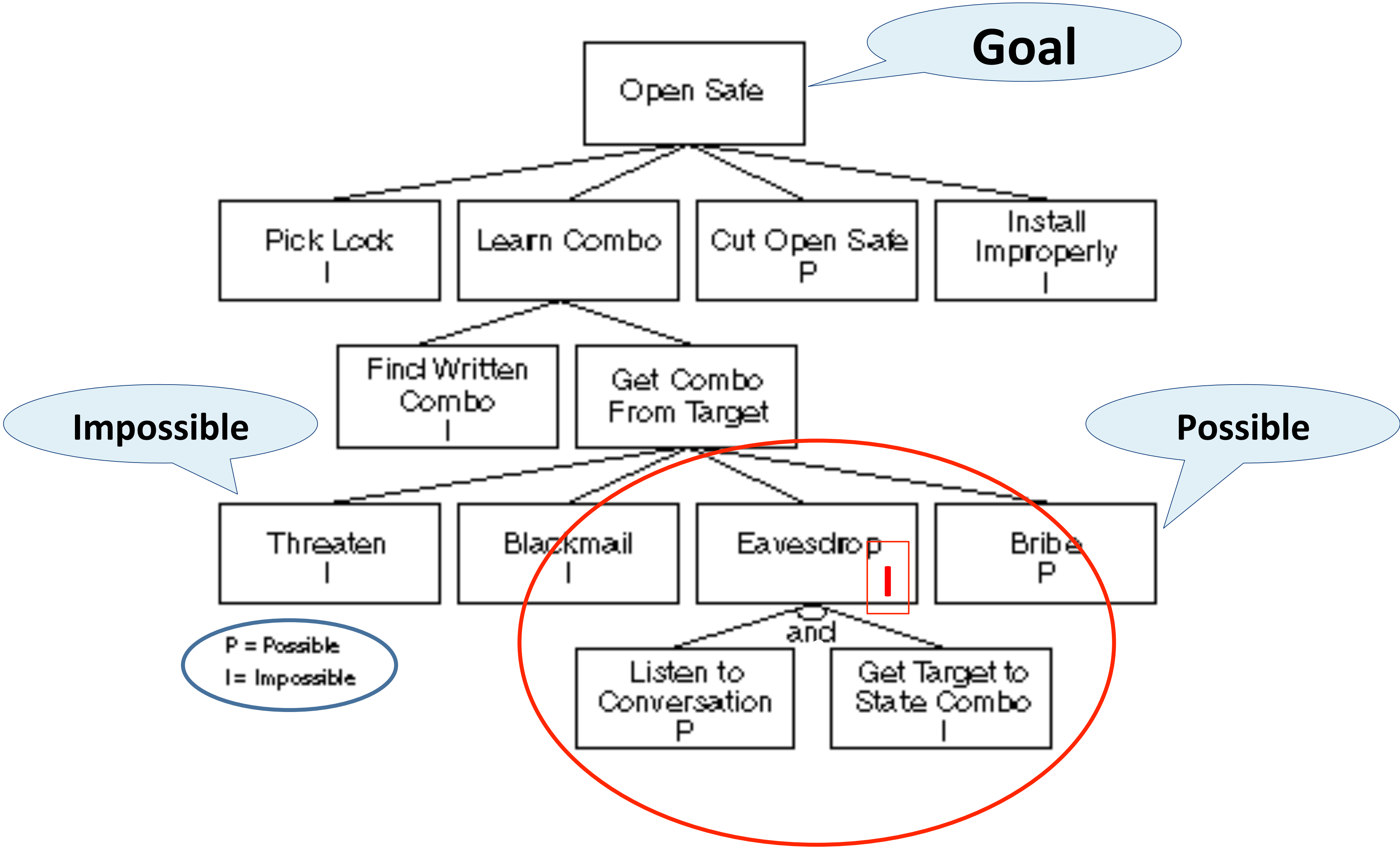
Threat Trees: Using them for Defence

- By creating such a Threat Tree, a company can analyse possible threats (courses of actions an attacker could take) and create defences against them
- Add annotations:
  - Leaf nodes in the Threat Tree are annotated with additional information
  - This information is propagated up the tree and aggregated according to AND and OR nodes
  - In our example
    - Nodes representing possible attacks are regarded by a company as either "possible" or "impossible"
    - Company has to take measures that, in particular, counteract possible kinds of attacks
    - They may ignore "impossible" attacks (saves money)
  - Other possible annotations:
    - Assumed costs, etc

Threat Trees: Annotations

- Add annotations to each node:
  - Determine, whether a particular attack is "possible" or "impossible"
  - What it means for OR nodes
    - the attack represented by an OR node is possible if ***any*** of its child nodes are possible
    - The attack represented by an OR node is impossible if ***all*** child nodes are annotated as being impossible
  - What it means for AND nodes
    - The attack represented by an AND node is possible only if ***all*** child nodes are annotated as being possible
    - The attack represented by an AND node is impossible if ***any*** of its child nodes are impossible

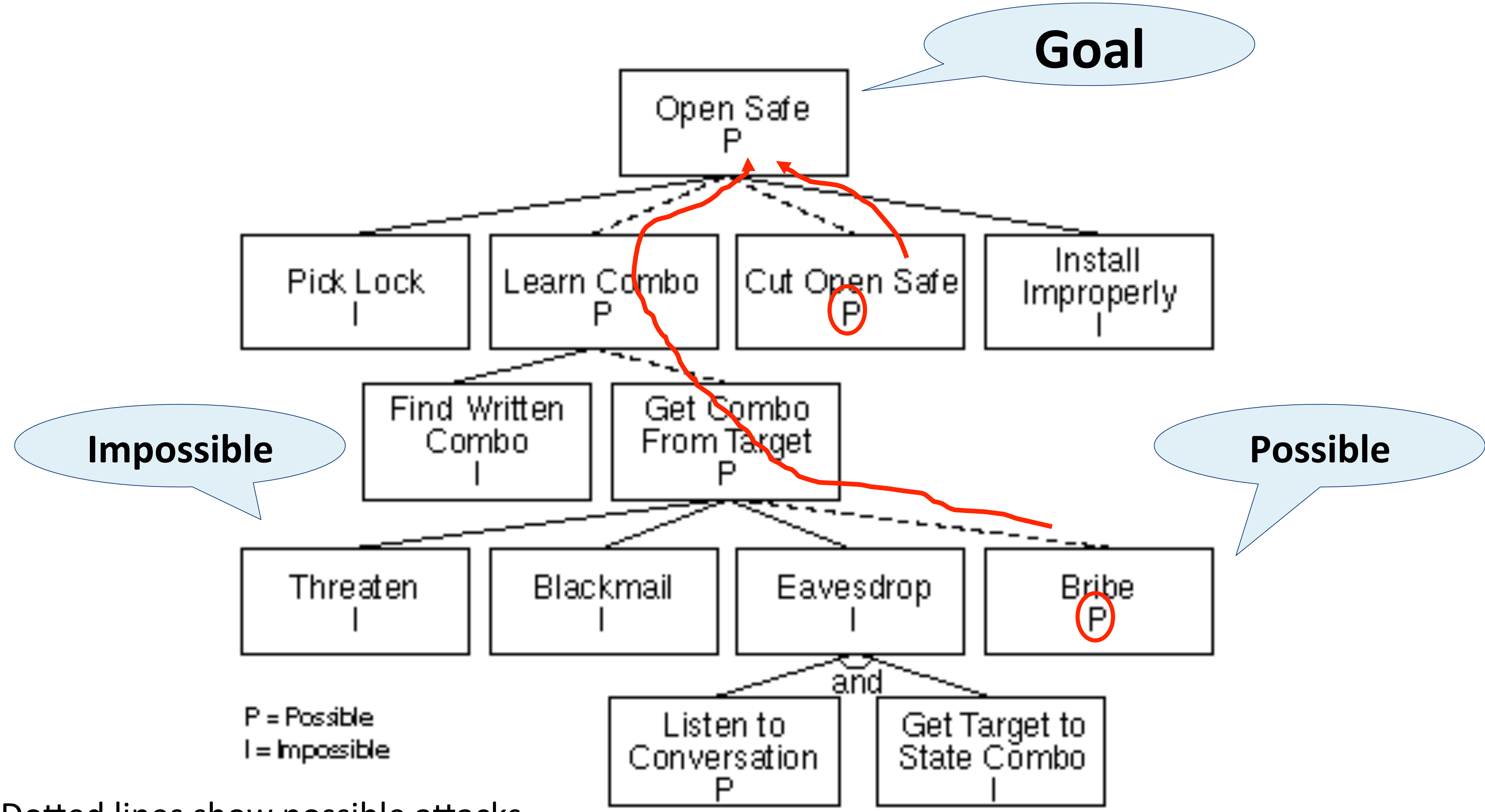
Threat Trees: Annotations



Threat Trees: Using them for Defence

- Interpreting an annotated Threat Tree
  - Annotations in our example
    - “possible” / “impossible”
  - Analysis of example results in insight that
    - Attackers may cut open safe, or
    - Learn the combination by bribing
  - Company has to take measures that, in particular, counteract these kinds of attacks
  - They may ignore “impossible” attacks (saves money)
- Additional annotations (e.g. Assumed costs for attacks) may give further insights how to create defences and how much to invest against what type of attack

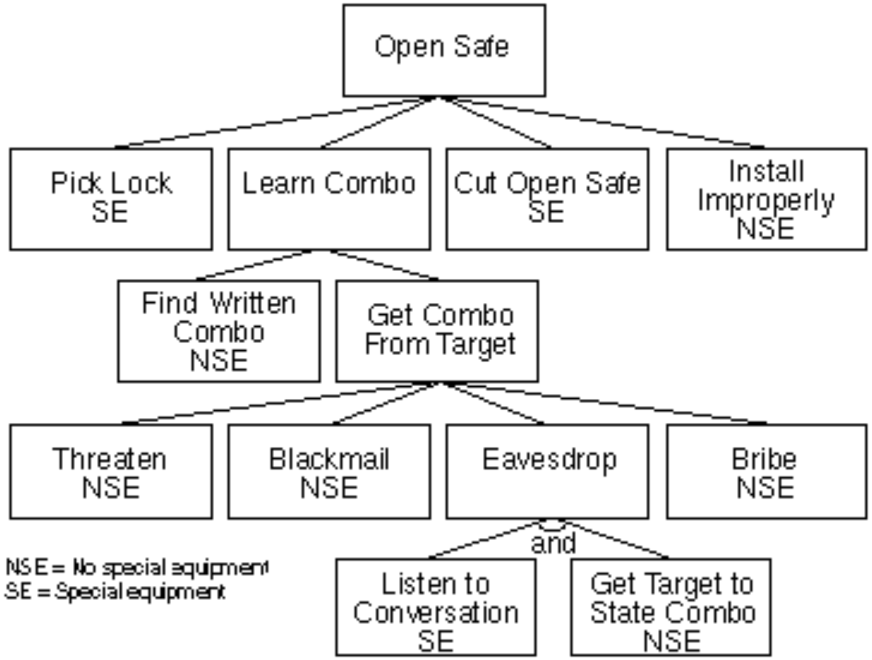
Threat Trees: Annotations



Dotted lines show possible attacks

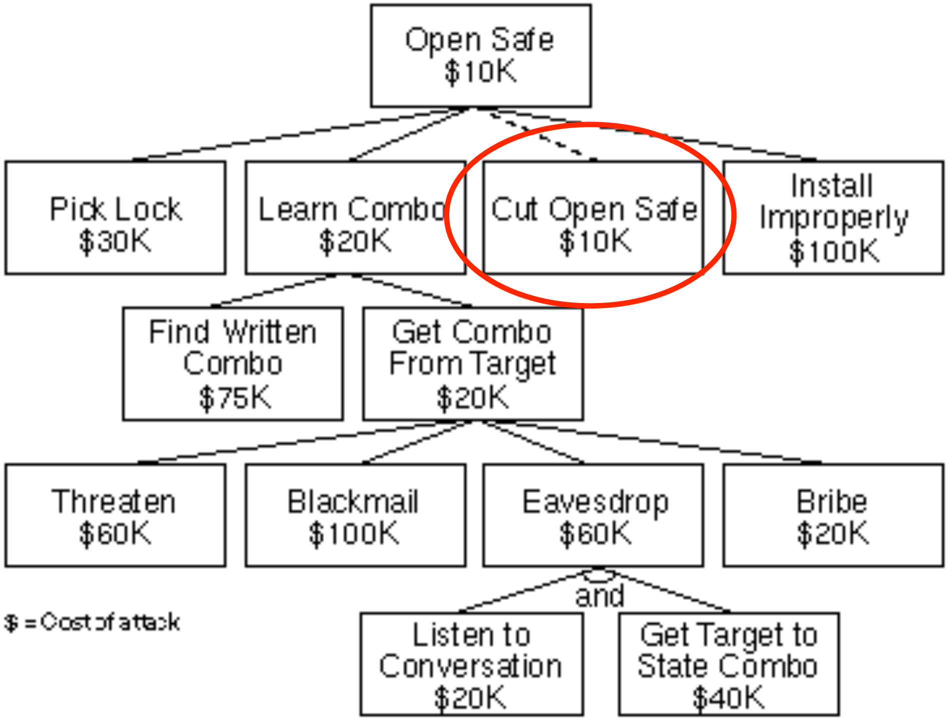
Threat Tree: Other possible Annotations

- Threat trees can have other annotations as well, such as
  - “easy” versus “hard”
  - “expensive” versus “inexpensive”
  - “intrusive” versus “nonintrusive”
  - “legal” versus “illegal
  - “special equipment” versus “no special equipment



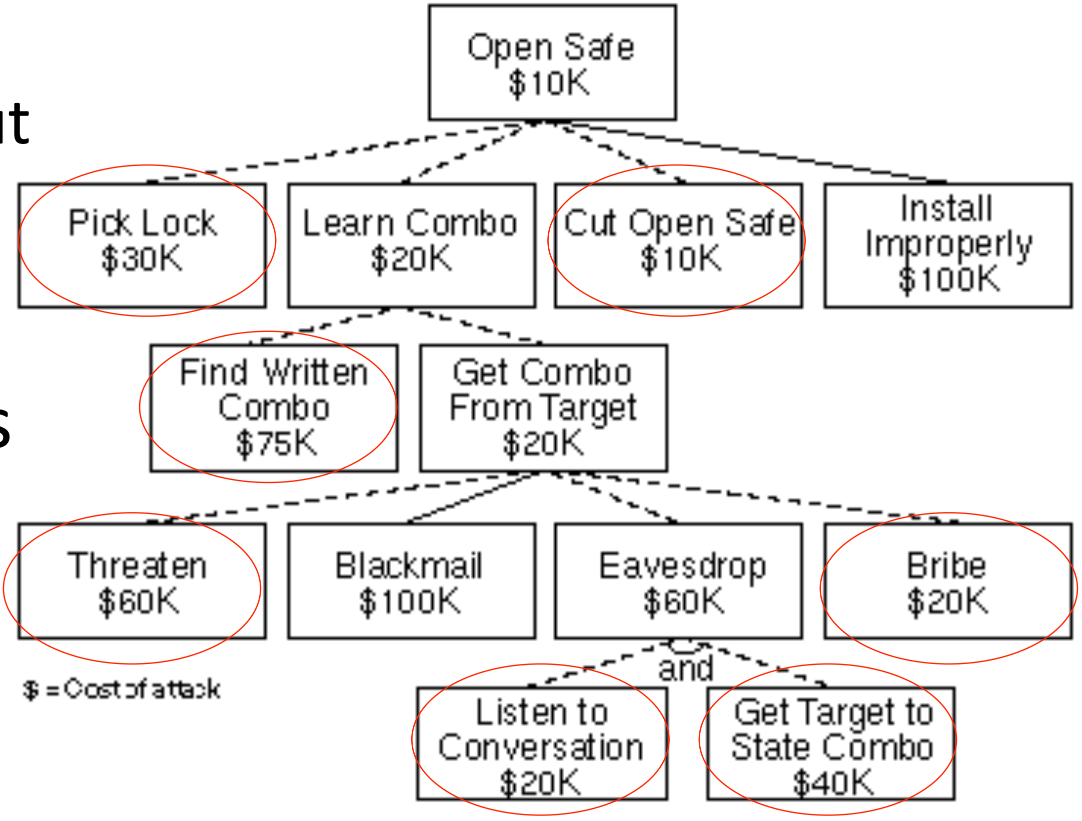
Threat Trees: Assigning Costs

- A company can also analyse the possible costs of attacks:
  - OR nodes will be annotated by the cost of their cheapest child node
  - AND nodes will be annotated with the sum of the costs of all child nodes
- Find cheapest attack and defend against it in particular



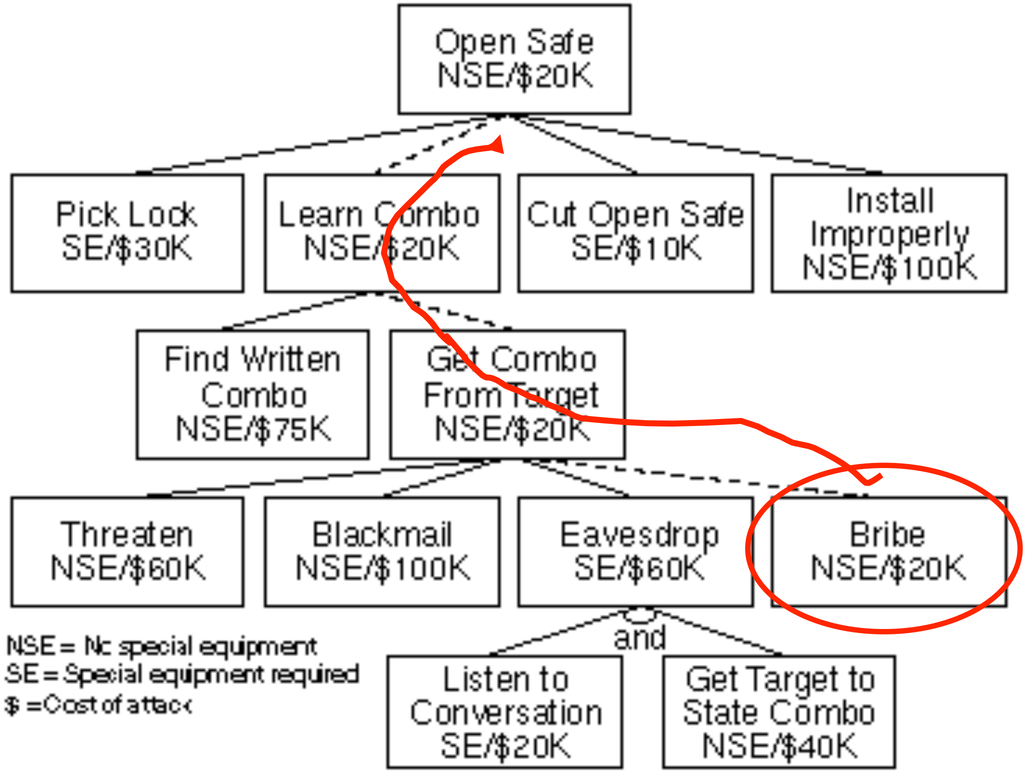
Threat Trees: Assign Costs

- E.g.: find all attacks that cost less than $100K
  - Dotted lines point out all these attacks/ threats
- Invest in defences against these threats



Threat Trees: Combine Annotations

- E.g: “Find cheapest attack that requires no special equipment”



Creating a Threat Tree

- Procedure:
  1. Identify possible attack goals
     - Each of these goals is the starting point for a separate Threat Tree or a sub-tree of a single Threat Tree
  2. Identify possible attacks and sub-goals to achieve a top-level goal, add concrete attacks as leaf nodes, add sub-goals to the tree
  3. Repeat 2. for each added sub-goal, until only leaf nodes (concrete attacks) are added to the tree
  4. Assign annotations to each leaf node, propagate and aggregate annotations up the tree
  5. Keep tree updated, if values are changed (company analyst changes values to ask “what-if” questions)

Using a Threat Tree

- Create the threat tree and determine which attack is most likely:
  - Depends on attacker
    - Skill levels, insider knowledge – profiling of possible attackers is needed
    - Depends on circumstances
- Defend against most likely attacks, don't spend money / time to defend against unlikely attacks
- Calculate whether it is worth at all to invest in defences!

That’s it for Today!

- Rest of the slides are of benefit for you to read
- Have a look on: http://homepages.abdn.ac.uk/m.j.kollingbaum/pages/teaching/CS3517/lectures/

Annual Loss Expectancy

- The calculation of Annual Loss Expectancy (ALE):
  - – Which threat is the most serious in terms of economic loss
  - – Calculation: ALE = Damage x Likelihood

Threat	Damage	Frequency	ALE
SWIFT fraud	50M	.005/yr	250.000
ATM fraud	20.000	1/yr	20.000
Stolen by Staff	3.000	200/yr	600.000

Annual Loss Expectancy

- The main problem is getting data
  - Historical data works for common attacks (high frequency) that don’t change much (e.g. Theft by staff members)
  - Historical data is less informative for rare attacks that also may change with technology (e.g. SWIFT)
  - Historical data is useless for unexpected attacks

Annual Loss Expectancy

- Usage
  - Provides insight into which threat is the biggest danger
  - Also, provides information how much money should be spent on security
    - There is no point to spend 100.000 on ATM security, if the ATM ALE is only 20.000

Defence: Security Protocols

- Procedures / rules that govern exchange of information
- Example Security protocols
  - Identify a human to another human
  - Identify a human to a computer
  - Identify a computer to another computer

Defence: Human-Human Identification

- Example: “Unknown person U from known medical institution M wants confidential information about patient P”
  - Person U may pose as an employee of the institution M, tries to gain information
- Defence
  - Regulation/proper protocol may demand
    - Receiver of such a phone call has to check that request is reasonable, if true (domain knowledge)
    - that receiver phones back institution M, using an official number from an NHS directory and asks to speak to person U

Defence: Human-Computer Identification

- This is much harder, actually a key problem in security
  - Passwords are far from ideal!
    - Problems with password strength, can be guessed, not changed on critical systems (human error)
  - Alternatives:
    - Additional hardware solutions: fingerprint, iris scan, voice recognition

Defence: Computer-Computer Identification

- Example: Car remote control – user presses button on key to unlock door remotely
- How to secure such an interaction? How to defend against attempts from other people to open the car?
- Version 1: broadcast a password to the car
  - Vulnerability:
    - Thief can snoop and record password signal, and replay it with a sending device to open a car
    - If there is only a small number (64K) of possible passwords that can be set on the key, then thief can sit in a car park and try them all
- Any ideas for better protocols for computer-computer identification?

Defence: Computer-Computer Identification

Cryptography

- Encrypt plaintext message
  - "Please send money" → 3264hds7ydt6he8
- Decrypt ciphertext
  - 3264hds7ydt6he8 → "Please send money"
- Is a good defence – encryption and decryption require special knowledge, which a thief lacks
  - This is usually a "key"

Defence: Computer-Computer Identification

Simple Authentication

- Example: Car remote control sends (a) the encrypted version of the ID, and (c) a so-called "nonce" number N
  - N is a "number used once"
  - Car decrypts, checks that it has not seen N before, so a thief cannot just record the signal emanating from the car key and replay it
  - The key for decryption is known to the car key and the receiving car, but is not transmitted
- Vulnerability
  - If car doesn't remember all past N's, a thief can simply record and wait
  - A thief can jam the car, snoop signal from the remote and then replay

Defence: Computer-Computer Identification

Challenge Response

- Another possibility: car sends “nonce” to remote control, which must return encrypted ID + nonce
- Vulnerability: “Man in the Middle”
  - Get user to press remote when not by car
  - Transmit remote signals to/from car

Lesson: People are the Weak Link

- Most of the attacks trick people or exploit people's carelessness
  - Valuable info in rubbish
  - Convince secretaries you're a doctor
  - Bribe someone
- Most real-world attacks are partially or mostly based on human weakness