Security Design Principles

System Security (CM0625, CM0631) 2024-25
Università Ca' Foscari Venezia

Riccardo Focardi
www.unive.it/data/persone/5590470
secgroup.dais.unive.it

Security design principles (1)

Simple vs. complex

Economy of mechanism: the design of security measures embodied in both hardware and software should be as simple and small as possible

- complex mechanisms are more vulnerable!
- complex mechanisms are hard to maintain and configure

Security design principles (2)

Permission vs. exclusion

Fail-safe default: access decisions should be based on permission rather than exclusion

- a mistake will tend to refuse permission (safe and easy to detect)
- access based on exclusion might permit unauthorised access that would be hard to notice

Security design principles (3)

Complete mediation: every access must be checked against the access control mechanism

Optimizations

- resource-intensive, but caching access decisions would **ignore** changes in access policy
- ***Example***: web applications should always check access to page/resources (e.g., do not base it on just the user ID)

Security design principles (4)

Open vs. closed design

Open design: the design of a security mechanism should be open rather than secret

- open design allows for expert reviews
- ***Example***: crypto algorithms are public and only the keys are kept secret

Security design principles (5)

Single vs. separated privileges

Separation of privilege: multiple privilege attributes are required to achieve a sensitive task

- ***Example 1***: separate privileges in organizations (e.g. role-based access control)
- ***Example 2***: multi-factor user authentication requires the use of multiple techniques
- Not to confuse with **least privilege**

Security design principles (6)

Min vs. max privileges

Least privilege: every process and every user of the system should operate at the least set of privileges necessary to perform the task

- mitigates attacks
- prevents accidental exposures

Security design principles (7)

Single vs. multiple protections

Layering: use of multiple, overlapping protection approaches

- failure of one protection will not leave the system unprotected
- multiple barriers between an adversary and protected information or services

⇒ *defense in depth*

Security design principles (8)

Usability

Psychological acceptability: the security mechanisms should not interfere with the work of users

- low **usability** might lead users to turn off mechanisms
- security mechanisms should be transparent when possible
- if the mechanisms are counterintuitive, users might make mistakes

Security design principles (9)

Isolated vs. connected

Isolation: physical or logical isolation of critical information/resources

Examples:

1. public access systems should be isolated from critical resources
2. processes/files of users should be isolated from one another
3. security mechanisms should be isolated from the rest of the system

Security design principles (10)

Modular vs. monolithic

Modularity: use of a modular architecture for mechanism design and implementation

- common security modules shared by applications that can be checked once and easily maintained
- mechanisms to protect security modules so to provide **Isolation**

Computer Security Strategy

- **Specification/policy**:
  What is the security scheme supposed to do?
- **Implementation/mechanisms**:
  How does it do it?
- **Correctness/assurance**:
  Does it really work?

Security Policy

Ease of use versus security: security involves penalties in usability

- Access control requires to remember passwords and perhaps perform other actions
- Firewalls reduce available transmission capacity
- Virus-checking software reduces available processing power
- ...

Cost of security versus cost of failure and recovery: security is not for free

- Cost of failure and recovery should be considered
- It depends on the asset value and on the **risk** (and cost) of attacks
- **business** decision influenced by **legal** requirements

Attack trees

Attack trees are a methodical way of describing the security of systems, based on varying attacks

Nodes are OR or AND

- OR is possible if one child is possible
- AND is possible if all children are possible

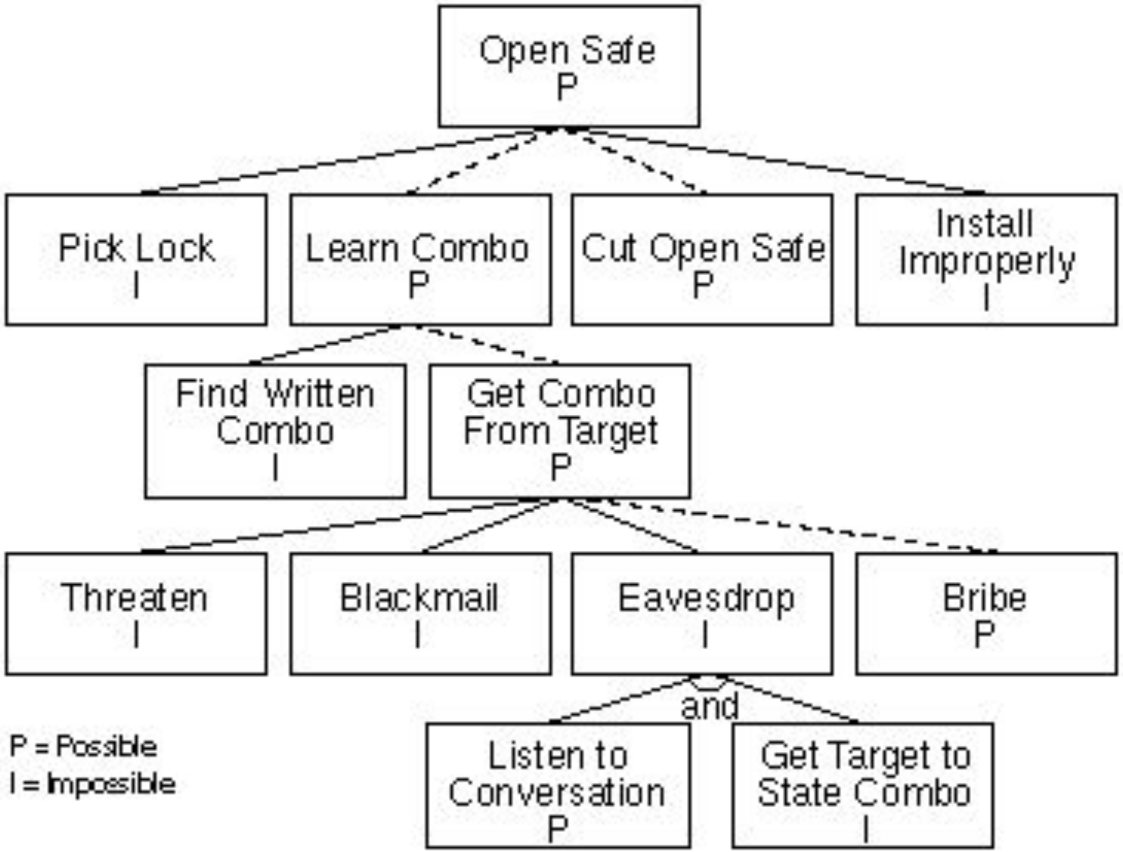

Figure 2: Possible Attacks.From https://www.schneier.com/

Attack trees

Values can be associated to the nodes

- Example: Cost

Values propagate from leaves up (parent gets the cheapest attack)

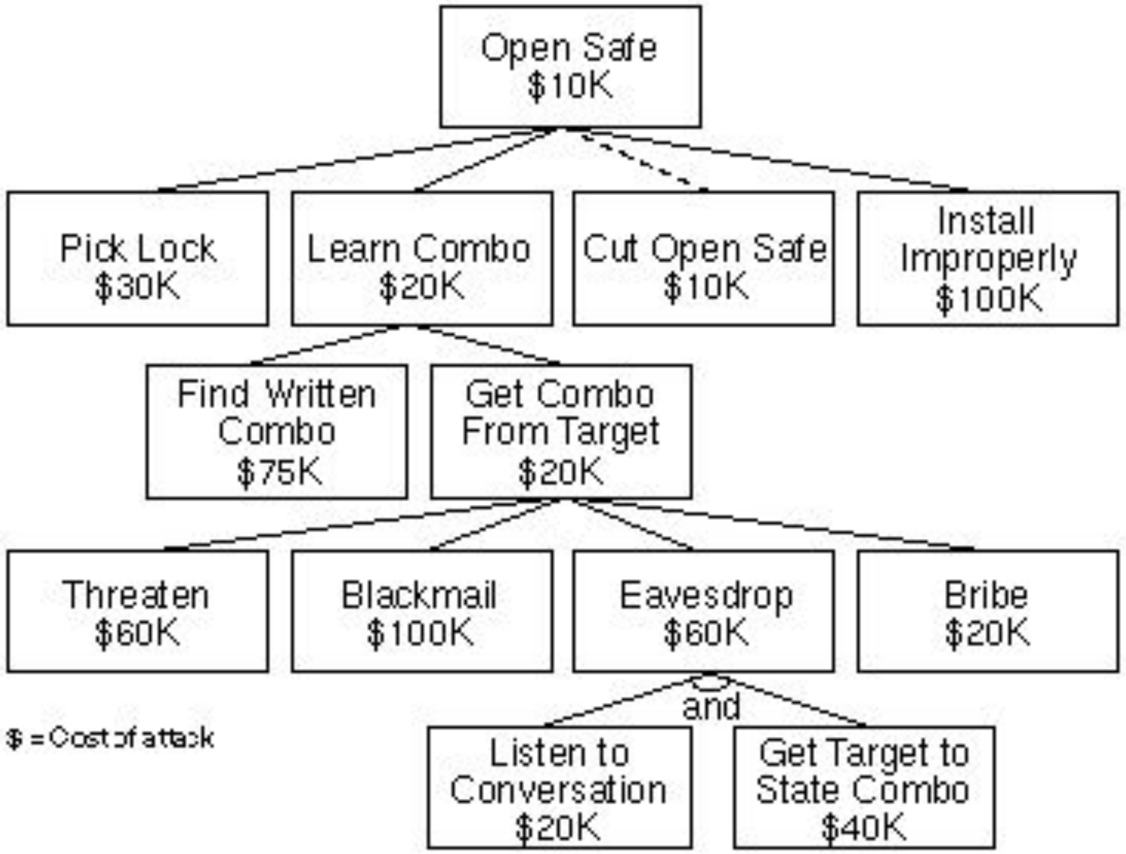

Figure 4: Cost of Attack. From https://www.schneier.com/

Attack trees

Evaluation

- Example : All attacks less that 100K $

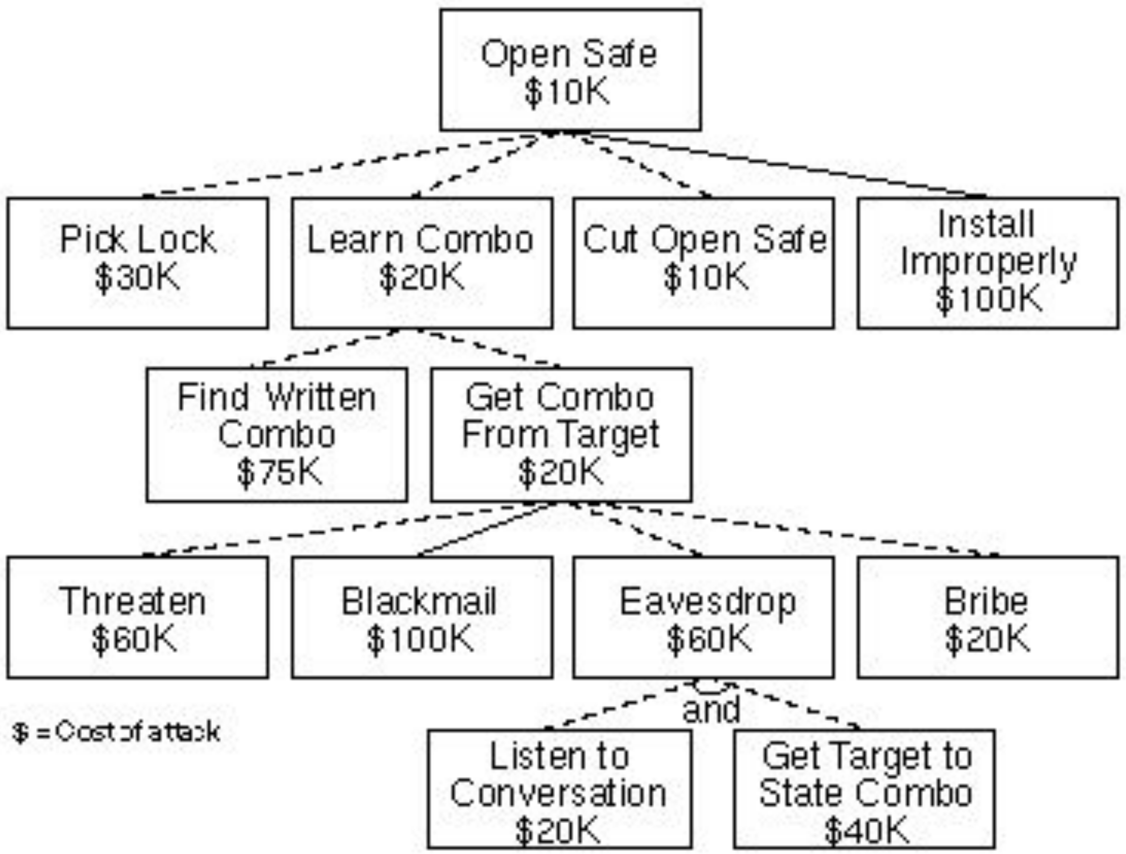

Figure 5: Attacks Less than $100,000. https://www.schneier.com/

Attack trees

Values can be associated to the nodes

- Example: Special equipment required

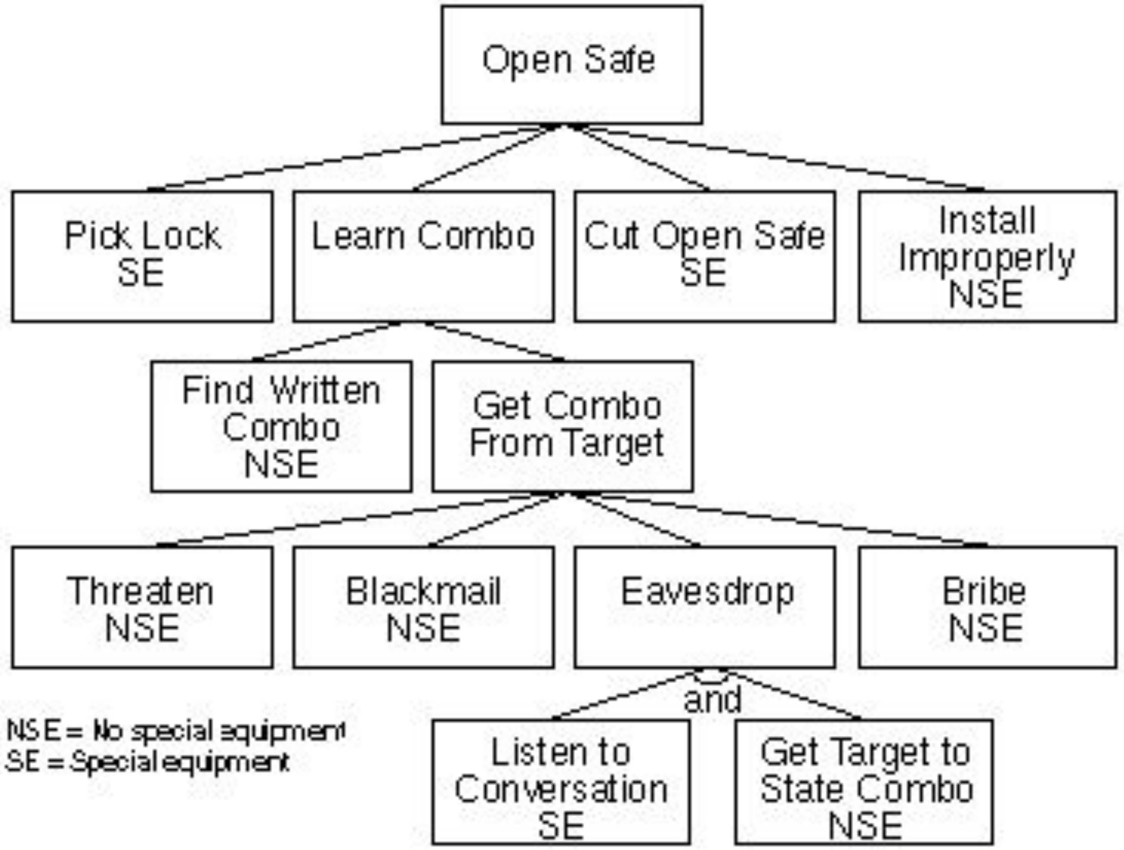

Figure 3: Special Equipment Required. https://www.schneier.com/

Attack trees

Evaluation

- Example: Cheapest requiring no special equipment

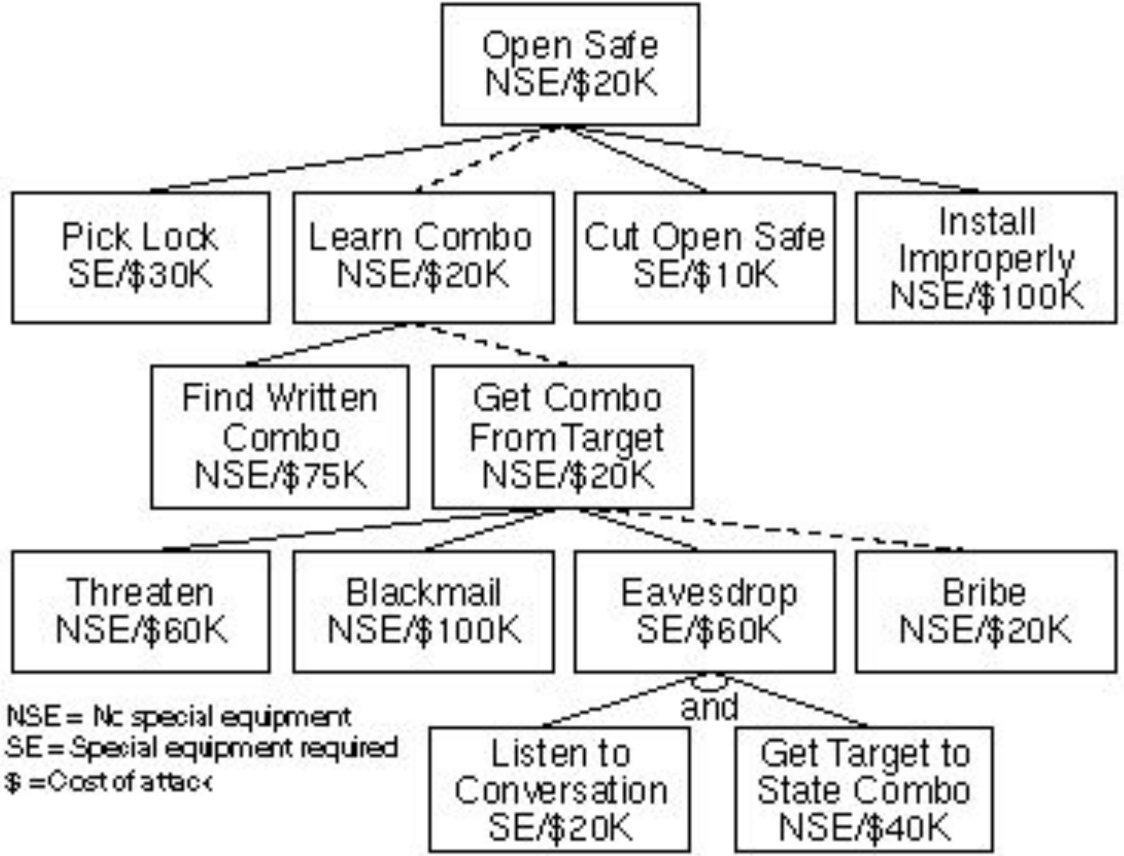

Figure 6: Cheapest NSE. https://www.schneier.com/

Computer Security Strategy

- **Specification/policy**:
  What is the security scheme supposed to do?
- **Implementation/mechanisms**:
  How does it do it?
- **Correctness/assurance**:
  Does it really work?

Security Implementation

Prevention: ideal security scheme in which no attack is successful

- Not always practical
- There might be vulnerabilities

Detection: when absolute protection is not feasible, it is still practical/useful to detect security attacks

- **Example**: Intrusion Detection System (IDS)

Response: the system responds in such a way as to halt the attack and prevent further damage

- **Example**: blacklisting IPs

Recovery: recover the system prior to the attack

- **Example**: backups

Correctness

Assurance: confidence that the system operates such that the system's security policy is enforced

1. Does the security system design meet its requirements?
2. Does the implementation meet its specifications?

⇒ Formal analysis can help

Evaluation: process of examining a computer product or system with respect to certain criteria

- development of evaluation criteria that can be applied to any security system (e.g. Common Criteria)

⇒ comparison of different solutions/products