CS431

Computer and Network Security

Principles for Designing Secure Systems

Abhishek Bichhawat

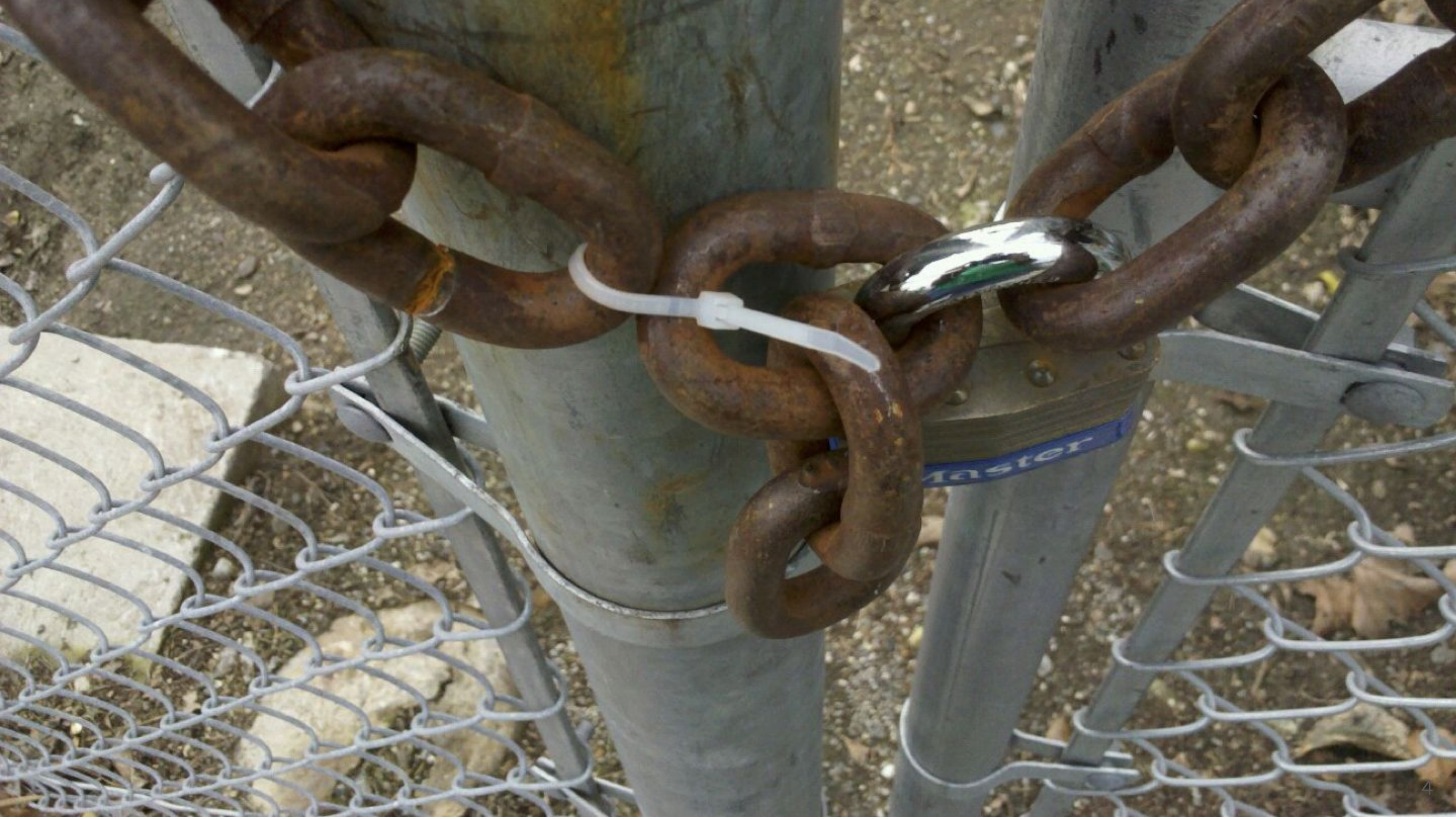
11/01/2024

Building Secure Systems Is Hard

- **Example**: Your grade files stored on IMS
  - **Policy:** only TAs should be able to read and write the grades file.
  - Easy to implement the *positive* aspect of the policy:
    - There just has to be one code path that allows a TA to get at the file.
  - But security is a *negative* goal:
    - We want no tricky way for a non-TA to get at the file.
  - There are a huge number of potential attacks to consider!
    - Exploit a bug in the server's code.
    - Guess a TA's password.
    - Steal a TA's laptop, maybe it has a local copy of the grades file.
    - Intercept grades when they are sent over the network to the registrar.
    - Break the cryptographic scheme used to encrypt grades over the network.
    - Trick the TA's computer into encrypting grades with the attacker’s key.
    - Get a job in the registrar's office, or as a TA.

Approaching Security

- Establish social norms
- Establish legal rules
- Make it “uneconomical”
- Establish defense


Master

“Security is a process, not a product”!

- Need to engineer security into a system
- During system design yields the best results
  - As more functionalities are added to the system, it becomes difficult to search and fix security bugs
  - Unfixed bugs lead to huge losses
- Security is not absolute
  - Cannot have full security
  - Tradeoff between attacker strength, cost and required security!

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary

"An impressive résumé, General, but remember—department-store security is different from national security."

Building Secure Systems

- For building secure systems, one should:
    - Understand the adversary
        - Make it hard enough for an adversary so that the adversary spends their energy elsewhere
        - Build a model of the adversary, their motives and their capabilities
        - Who can attack you and why?

Building Secure Systems

- For building secure systems, one should:
    - Understand the adversary (assumptions on the adversary)
        - Knows general information about systems and can interact
        - May try a brute-force approach
        - May collude to perform complex attacks
        - Has the resources required for the attack
        - Can obtain privileges

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
    - Limit the **trusted computing base (TCB)**
    - TCB are the components of a system that you trust
    - Security of a system is built on top of the TCB
    - Determine what components should be in the TCB
      - Should be secure and “unhackable”
      - “Old”-code is the most vulnerable
    - KISS principle

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs

Building Secure Systems

- For building secure systems, one should:
    - Understand the adversary
    - Understand the cost tradeoffs
        - Simply put, the cost of defense should be lesser than the cost of resource being secured
        - Security is directly proportional to costs
            - Example : Safes and lockers

Building Secure Systems

- For building secure systems, one should:
    - Understand the adversary
    - Understand the cost tradeoffs
    - Detect, if not prevent

Building Secure Systems

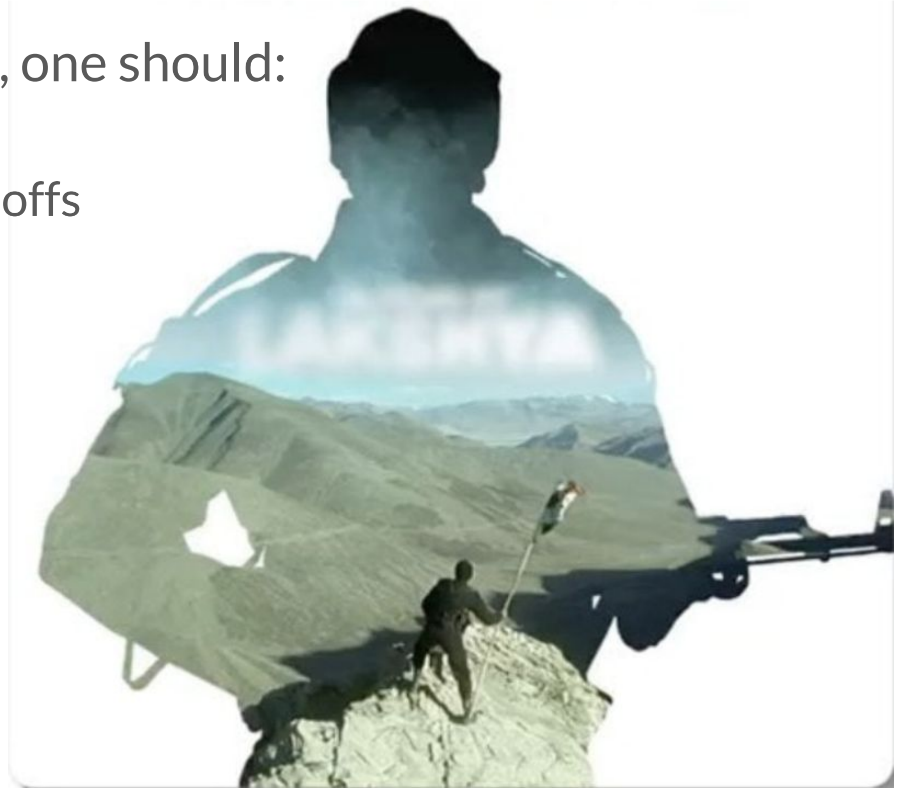
- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
    - Build systems to deter or prevent all attacks
    - If we can't stop an attack, we should be able to detect (and recover)
    - Prepare for the worst.
    - Do not rely on prevention; recovery is as important

Building Secure Systems

- For building secure systems, one should:
    - Understand the adversary
    - Understand the cost tradeoffs
    - Detect, if not prevent
    - Design in-depth defense

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense



Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense

Murud-Janjira Fort

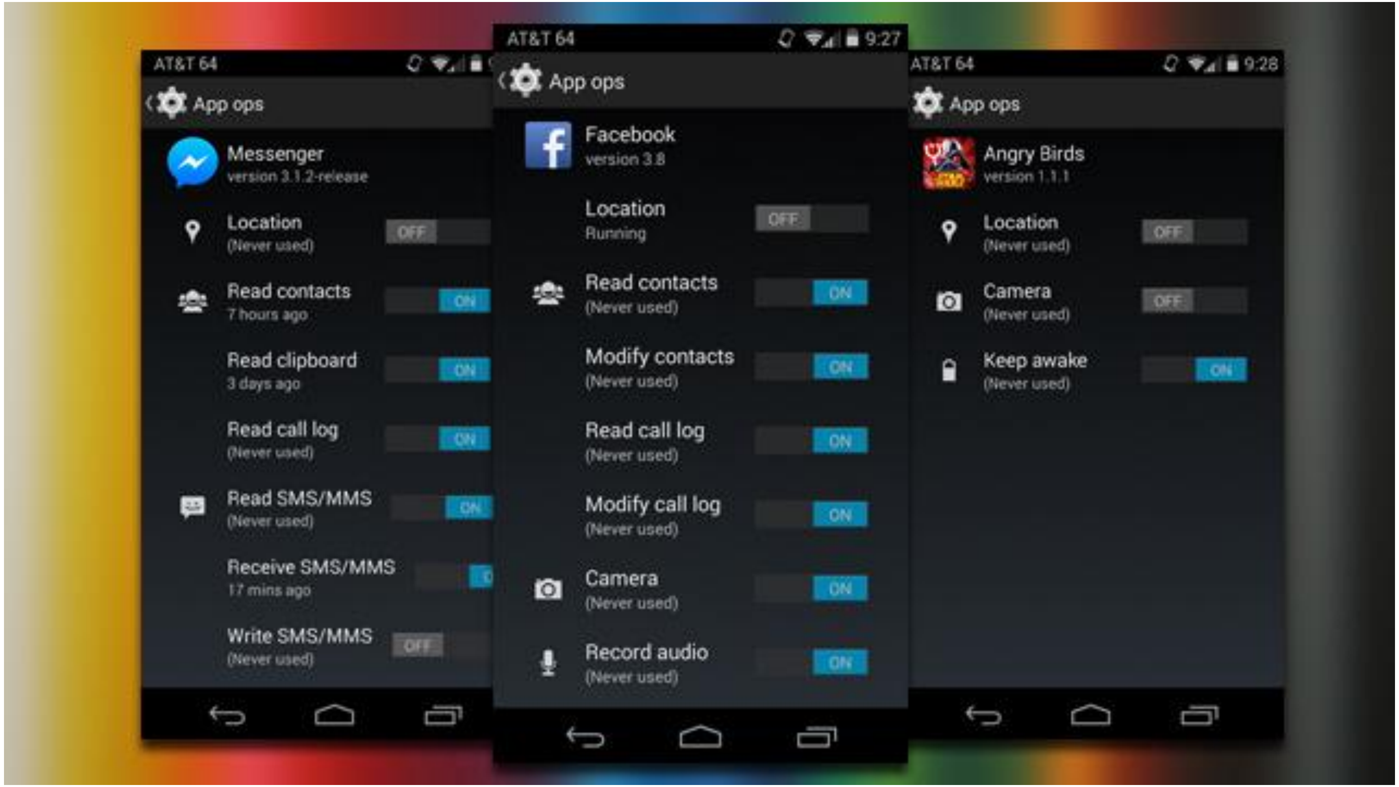
Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense
    - Multiple types of defenses should be layered together
      - Attacker should have to breach all defenses
    - Remember the costs

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense
  - Provide least privilege

Building Secure Systems



AT&T 64

App ops

Messenger
version 3.1.2-release

Location
(Never used)
OFF

Read contacts
7 hours ago
ON

Read clipboard
3 days ago
ON

Read call log
(Never used)
ON

Read SMS/MMS
(Never used)
ON

Receive SMS/MMS
17 mins ago

Write SMS/MMS
(Never used)
OFF

AT&T 64 9:27

App ops

Facebook
version 3.8

Location
Running
OFF

Read contacts
(Never used)
ON

Modify contacts
(Never used)
ON

Read call log
(Never used)
ON

Modify call log
(Never used)
ON

Camera
(Never used)
ON

Record audio
(Never used)
ON

AT&T 64 9:28

App ops

Angry Birds
version 1.1.1

Location
(Never used)
OFF

Camera
(Never used)
OFF

Keep awake
(Never used)
ON

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense
  - Provide least privilege
    - Do not grant unnecessary permissions
    - Devise methods to control access

Building Secure Systems (story so far)

- For building secure systems, one should:
  - Understand the adversary
    - Who can attack you and why?
  - Understand the cost tradeoffs
    - Security is directly proportional to costs
  - Detect, if not prevent
    - If we can’t stop an attack, we should be able to detect (and recover)
  - Design in-depth defense
    - Multiple types of defenses should be layered together
  - Provide least privileges
    - Do not grant unnecessary permissions

Building Secure Systems

- For building secure systems, one should also:
    - **Completely mediate access**
        - Examples: Network firewall, airport security, the doors to hostel rooms!

Building Secure Systems

- For building secure systems, one should also:
  - **Completely mediate access**
    - Examples: Network firewall, airport security, the doors to hostel rooms!

Building Secure Systems

- For building secure systems, one should also:
  - **Completely mediate access**
    - Monitor and secure all access points
    - Make sure this is “sound” and “complete”

Building Secure Systems

- For building secure systems, one should also:
  - **Completely mediate access**
    - Monitor and secure all access points
    - Make sure this is “sound” and “complete”
      - Completeness (can’t be bypassed)
      - Security (can’t be tampered with)

Building Secure Systems

Consider the following pseudocode:

```
function withdraw_money (user, withdraw_amount)
    let balance := getBalance (user)
    if balance < withdraw_amount then abort
    setBalance (user, balance - withdraw_amount)
    return withdraw_amount to user
```

Building Secure Systems

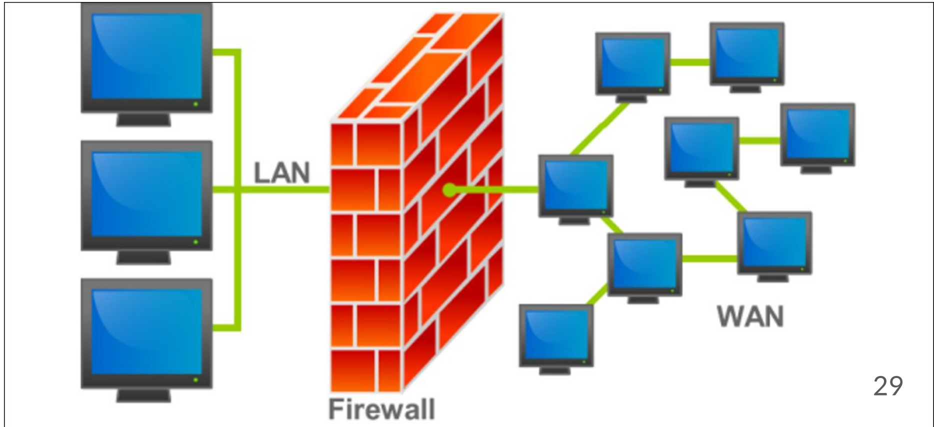
- For building secure systems, one should also:
  - Completely mediate access
  - Distribute responsibility when providing privilege/access
    - Multiple parties to exercise that responsibility

Building Secure Systems

- For building secure systems, one should also:
    - Completely mediate access
    - Distribute responsibility when providing privilege/access
    - **Consider faults and failures**

Building Secure Systems

- For building secure systems, one should also:
  - Completely mediate access
  - Distribute responsibility when providing privilege/access
  - **Consider faults and failures**
    - Employ fail-safe defaults

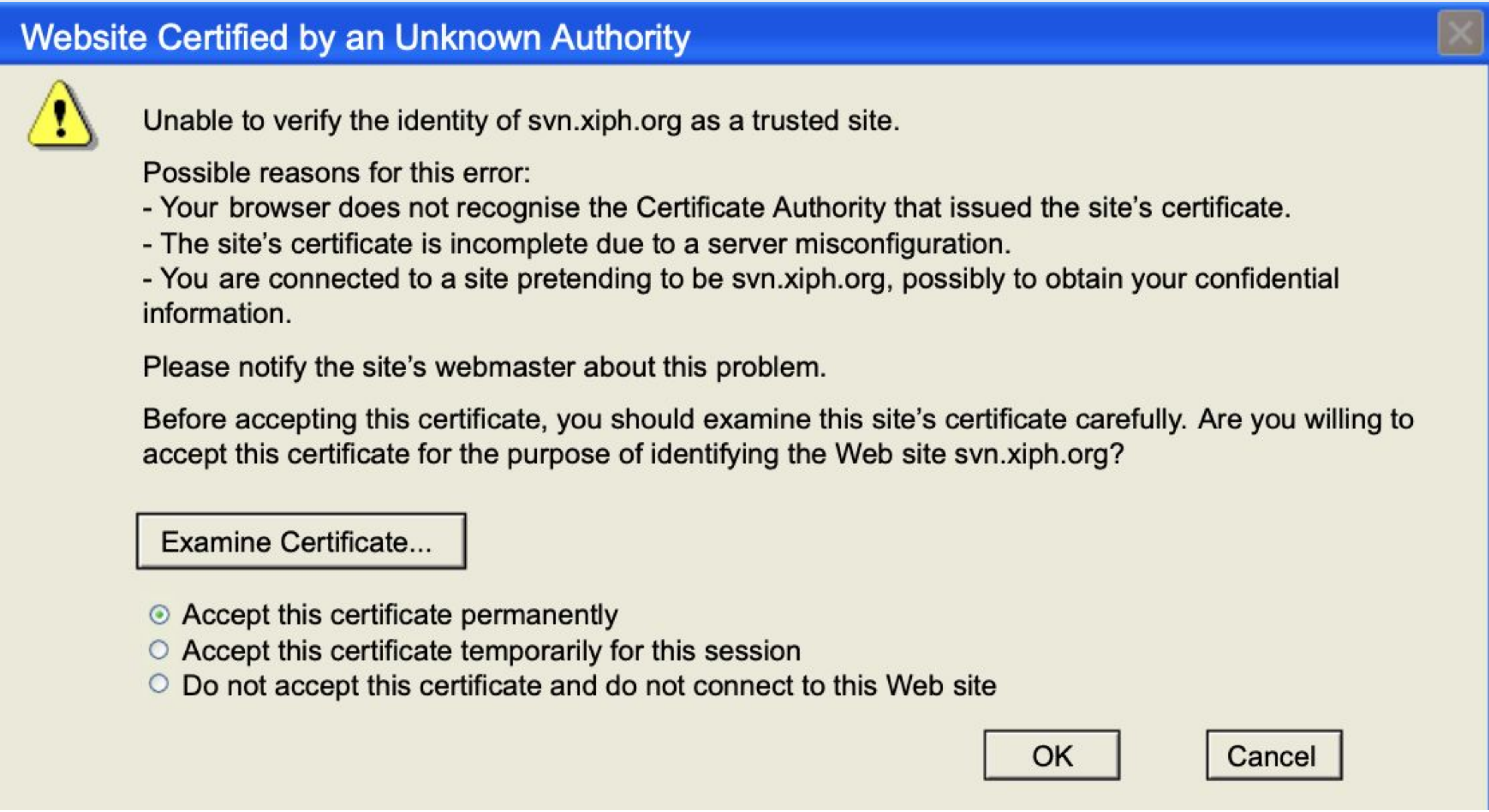


Building Secure Systems

- For building secure systems, one should also:
    - Completely mediate access
    - Distribute responsibility when providing privilege/access
    - Consider faults and failures
    - **Consider human factors, aspects and behavior**
        - Users build systems; make errors; account for them

Building Secure Systems

- For building secure systems, one should also:
  - Completely mediate access
  - Distribute responsibility when providing privilege/access
  - Consider faults and failures
  - **Consider human factors, aspects and behavior**
    - System should be used by end-users



Website Certified by an Unknown Authority

Unable to verify the identity of svn.xiph.org as a trusted site.

Possible reasons for this error:
- Your browser does not recognise the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be svn.xiph.org, possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to accept this certificate for the purpose of identifying the Web site svn.xiph.org?

● Accept this certificate permanently
○ Accept this certificate temporarily for this session
○ Do not accept this certificate and do not connect to this Web site

Building Secure Systems

- For building secure systems, one should also:
    - Completely mediate access
    - Distribute responsibility when providing privilege/
    - Consider faults and failures
    - **Consider human factors, aspects and behavior**
        - Social engineering (beware of your partners)

Building Secure Systems

- For building secure systems, one should:
  - Understand the adversary
  - Understand the cost tradeoffs
  - Detect, if not prevent
  - Design in-depth defense
  - Provide least privileges
  - Completely mediate access
  - Distribute responsibility when providing privilege/access
  - Consider faults and failures
  - Consider human factors, aspects and behavior
- DESIGN SECURITY FROM START