Computer Security

CSE 550 - Computer Systems

Austin Gebauer

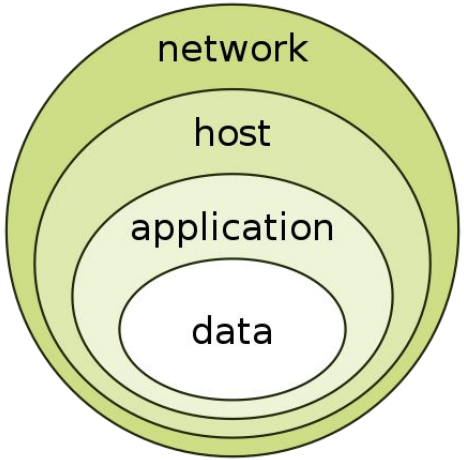
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY of WASHINGTON

Outline

- The context of security
- Threat models
- Summary of the key concepts from the papers
  - The Protection of Information in Computer Systems; Saltzer, Schroeder [1975]
  - Computer Security in the Real World; Lampson [2000]
- Distillation of concepts
- The context of cryptography
- Security principles
- Standards-based interoperability using open protocols
- Security in the Cloud
- Highlight of a recent security breach and following industry trends
- What is the cost of security? Evaluation of tradeoffs

The Context of Security

- Computer security is the protection of computer systems and information from harm, theft, unauthorized use, and disruption of service
- Often said that a security system is only as strong as its weakest link.
  - The links include people and physical security
- Security concerns exist at every layer
  - Hardware
    - CPU, Memory, Disk
    - Fabrication/Manufacturing
  - Software
    - Operating Systems
    - Applications
    - Runtime Environments
  - Networks
    - Local/Wide
    - Cloud



Threat Model

- Security is a spectrum and is highly dependent on context
- Being secure is dependent on your **threat model** and definition of what is safe
- Important to **know what you are trying to protect**, and **against whom** you wish to protect it.
  - What are the assets of value?
  - What are the threats?
- Since there's really no such thing as a perfect security, when we say that a system is "secure", what we are really saying is that it provides a sufficient level of security for our asserts of interest against certain classes of threat.
- We need to address the security of a system under the designated thread model

Paper 1: The Protection of Information in Computer Systems

Summary of considerations surrounding information protection:

- The application of computers to information handling problems produces a need for a variety of **security mechanisms**
- At least four levels of **functional goals** for a protection system were identified
  - At all levels, the provisions for dynamic changes to authorization for access are desired
- Makes the claim that no one knows how to build a secure system without flaws.
  - The paper proposes to rely on a set of **security principles**
- Introduces **authentication** as a system that verifies a user's claimed identity
- Introduces **authorization** as giving a user access to some object
- An **audit** trail can be established using compromise recording

Paper 1: The Protection of Information in Computer Systems

Functional levels/goals described in the paper:

- **All-or-nothing systems**
  - Provide isolation of users, sometimes moderated by some information sharing. If only isolation provided, the user of such a system might as well be using a private computer
- **Controlled sharing**
  - Control explicitly who may access each object stored in the system
- **User-programmed sharing controls**
  - User-defined protected objects and subsystems
  - Protected subsystem is a collection of programs and data with the property that only the programs of the subsystem have direct access to the protected objects
- **Putting strings on information**
  - Control information *after* it's been released

Paper 1: The Protection of Information in Computer Systems

- The paper took a bottom-up approach to designing an information protection mechanism for memory segments in a multiuser system
- Mechanisms weren't intended so much to explain the particular systems as they are to explain the underlying concepts of information protection
- Generalizes the mechanics of access using two different models
  - Access Control Lists
  - Capabilities
- Access models are appropriate to protect other objects provided by the hardware and software

TABLE I
TYPICAL SYSTEM-PROVIDED PROTECTED OBJECTS

Object	Typical Separately Permittable Operations
Data segment	READ data from the segment WRITE data into the segment Use any capability found in the segment Use any READ capability found in the segment WRITE a capability into the segment
Access controller	READ access control list Modify names appearing on an access control list Modify permissions in access control list entries Destroy object protected by this access controller
FIFO message queue	Enqueue a message Dequeue a message Examine queue contents without dequeueing
Input/Output device	READ data WRITE data Issue device-control commands
Removable recording medium (e.g., magnetic tape reel)	READ data WRITE over data WRITE data in new area

Paper 1: The Protection of Information in Computer Systems

State of the art research directions at the time the paper was published in 1975:

1. Certification of the correctness of protection system designs and implementations
2. Invulnerability to single faults
3. Constraints on use of information after release
4. Encipherment of information with secret keys
5. Improved authentication schemes

Paper 2: Computer Security in the Real World

Let's discuss some quotes from the paper:

> "On the other hand, not much harm is actually being done by attacks on these insecure systems."

> "Many vendors of security have learned to their regret that although people complain about inadequate security, they won't spend much money, sacrifice many features, or put up with much inconvenience in order to improve it."

> "We don't have "real" security guarantees to stop bad things from happening, and the main reason is that people don't buy it. They don't buy it because the danger is small, and because security is a pain."

> "While we await a catastrophe, simpler setup is the *most important step* toward better security."

1. **Have these quotes aged well? Which ones have / haven't?**

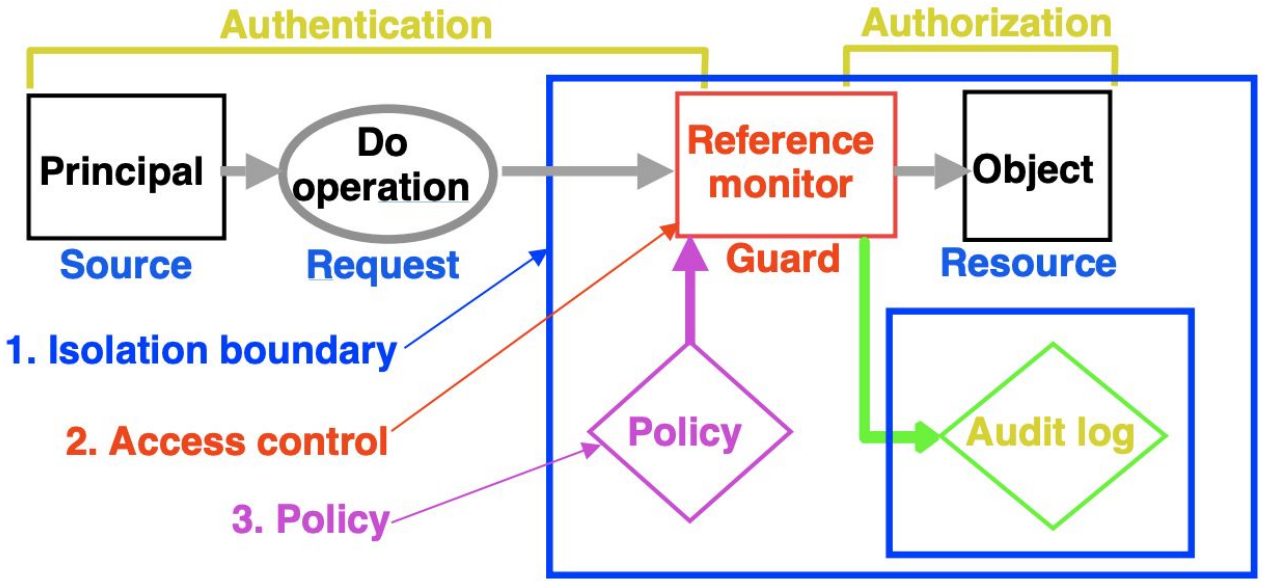
https://tinyurl.com/4cc6vty6

Paper 2: Computer Security in the Real World

Three basic mechanisms for implementing security:

- **Authenticating** principals
  - Answers the question “Who said that?”
  - Principal - abstraction of “who” or “identity”
  - Principals are people, services, groups, keys
- **Authorizing** access
  - Answers the question “Who can do which operations on some object?”
- **Auditing** the decisions of the guard
  - So that later it’s possible to figure out what, when, and why something happened
  - Also “who” since a principal is associated with the operation

Paper 2: Computer Security in the Real World



Paper 2: Computer Security in the Real World

- **Local** access control
  - Exists in most systems
  - Typically a similar model including the means for
    - Authentication
    - Authorization via Policy, Groups, ACLs
- **Distributed** access control
  - “Everyone needs a uniform way to do end-to-end authentication and authorization across the entire Internet.”
  - Chains of responsibility for a “speaks for” relationship
  - Delegation of trust
  - Principal A speaks for B: A ⇒ B
    - A is **as powerful as** B, or **trusted like** B
    - Key #123 ⇒ Alice (key for Alice), Alice ⇒ Atom (group membership)

Distillation of Concepts

- Authentication
  - Who are we talking to?
    - Identity
      - Humans and machines
- Authorization
  - What is the access policy?
    - Access Control List
    - Permissions/Privileges
      - Read, Write, Execute on files
      - HTTP verbs for network services
      - Group Membership
      - Capabilities
- Delegation
- Auditing

The Context of Cryptography

"Computer security has been regarded as an offshoot of communication security, which is based on cryptography. Since cryptography can be nearly perfect, it's natural to think that computer security can be as well."

- *Lampson*, *Computer Security in the Real World*

- Where does cryptography fit into the picture of secure systems?
- What problems does it solve?

The Context of Cryptography

- Cryptography *alone* is fairly useless
  - Think of a physical lock. It's pretty useless on its own.
  - It needs to be a part of a larger system
  - Larger system can be a door on a building, a chain, a safe
  - System extends to to people who are using the lock
  - They need to remember to actually lock it and not leave the key around for others to find
- The same goes for cryptography -- it's a part of a much larger security system
- Cryptography takes on the role of the lock
  - It has to distinguish between "good" access and "bad" access
  - Plays a role in both authentication and authorization
- It is not *the security solution*, but *a part of the solution*

The Context of Cryptography

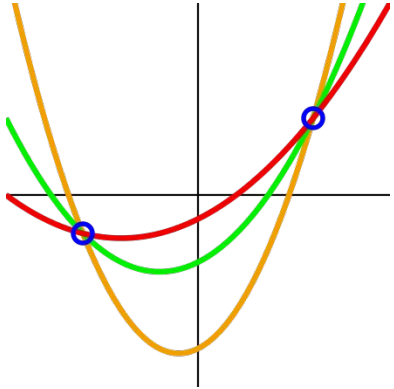
- Authentication and authorization are not always enough
  - Cryptography provides secure communication in the presence of adversarial behavior
  - Prevent third parties or the public from reading private messages
- Information security concepts are central to cryptography
  - Confidentiality
    - Encrypt the information to make it unintelligible to everyone but those who are authorized to view it
  - Integrity
    - Assurance that data has not been modified in an unauthorized manner after it was created, transmitted or stored
  - Authentication
    - Can be used to verify the identity of who created the information, such as a user or system

Cryptographic Keys

- Cryptographic techniques use “keys” to protect access (lock example)
- Symmetric
  - Same key performs encryption and decryption
- Asymmetric
  - Mathematically related key pairs (public and private)
  - Decryption, Signature generation use the private portion
  - Encryption, Signature verification use the public portion
- Think about the lock example
  - The keys are a secret
  - What should we do with the keys?

Securing Cryptographic Keys

- Difficult problem
- Key management and key storage is **crucial** to any cryptographic system.
- Cryptographic Key Management Systems
  - System includes generation, exchange, storage, use, destruction and replacement of cryptographic keys
- Secret Sharing
  - Split keys into “shares”
  - Majority of shares needed to recover the key
  - Shamir’s Secret Sharing algorithm
  - Used to secure a secret in a distributed way
  - Most often to secure other encryption keys



Security Principles

Both of the papers made the claim that secure systems aren't provably secure. Security principles (and understanding the threat model) help us to build more secure systems.

- **Least privilege**
  - Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
- **Separation of privilege**
  - Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

Security Principles

- **Economy of mechanism**
  - Keep the design as simple and small as possible
- **Least common mechanism**
  - Minimize the amount of mechanism common to more than one user and depended on by all users
- **Open design**
  - The design should not be secret

Open Security Protocols

- Open Security Protocols provide **standards-based interoperability** between computer systems
- Often reviewed in a public arena
- Provider less bespoke workflows
- Let's take a closer look at an open protocol for **authentication** and **authorization** that's commonly used on the web
  - OpenID Connect (OIDC)

Delegated Authentication and Authorization

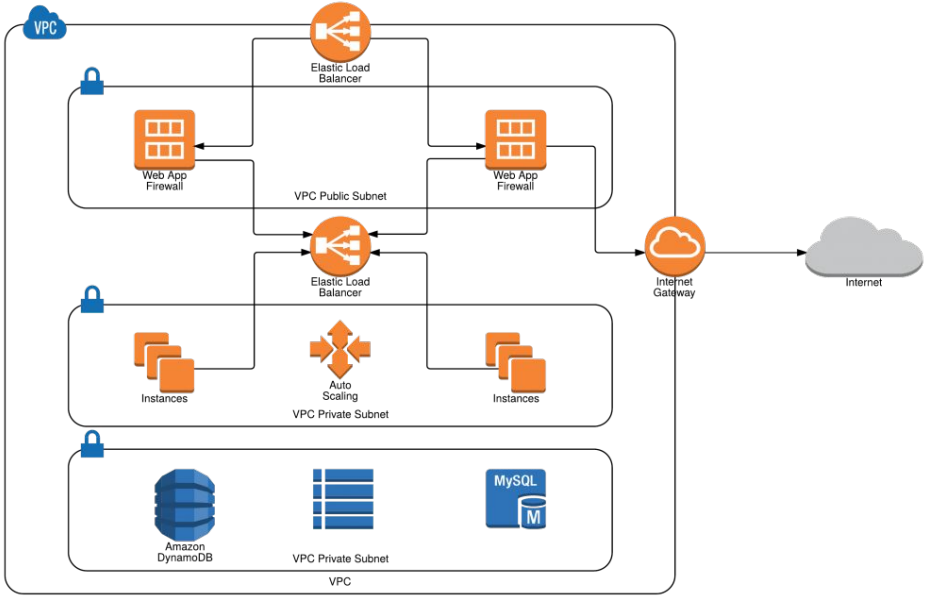
- How many have authenticated using something like this before?

Authentication – OpenID Connect (OIDC)

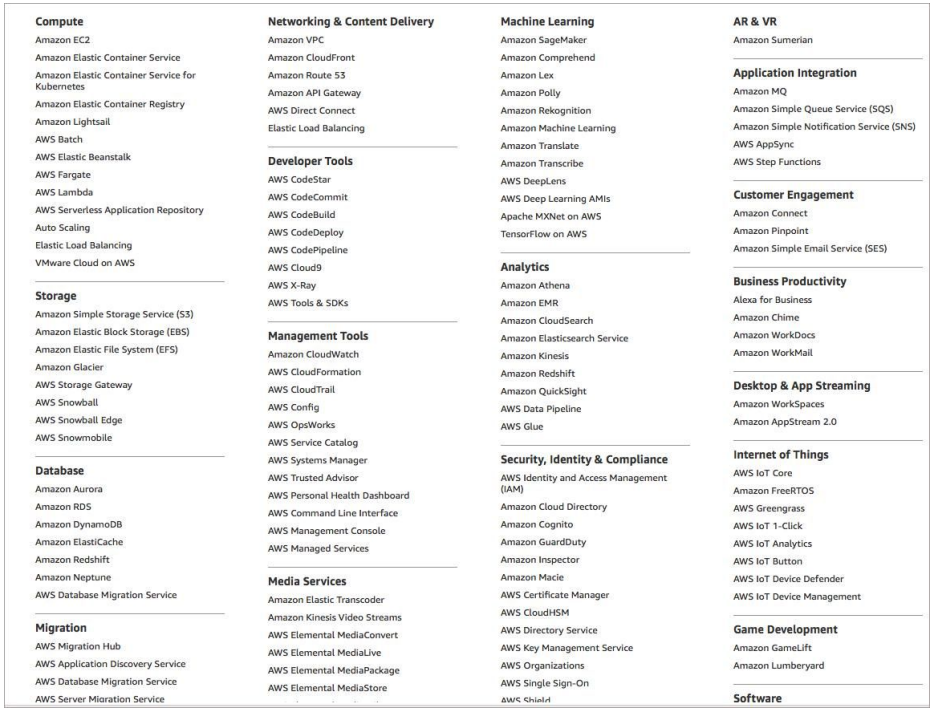
- OpenID Connect (OIDC) is an identity layer on top of the OAuth 2.0 protocol.
- OAuth 2.0 is an industry-standard protocol for authorization.
  - Focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices.
- It allows clients to verify the identity of the end-user based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the end-user in an interoperable manner.

Cloud Security

- The scale of cloud providers presents unique security challenges
- Arbitrarily complex environments
- How do we evaluate the threat model on cloud infrastructure?



Cloud Security



Compute
Amazon EC2
Amazon Elastic Container Service
Amazon Elastic Container Service for Kubernetes
Amazon Elastic Container Registry
Amazon Lightsail
AWS Batch
AWS Elastic Beanstalk
AWS Fargate
AWS Lambda
AWS Serverless Application Repository
Auto Scaling
Elastic Load Balancing
VMware Cloud on AWS

Storage
Amazon Simple Storage Service (S3)
Amazon Elastic Block Storage (EBS)
Amazon Elastic File System (EFS)
Amazon Glacier
AWS Storage Gateway
AWS Snowball
AWS Snowball Edge
AWS Snowmobile

Database
Amazon Aurora
Amazon RDS
Amazon DynamoDB
Amazon ElastiCache
Amazon Redshift
Amazon Neptune
AWS Database Migration Service

Migration
AWS Migration Hub
AWS Application Discovery Service
AWS Database Migration Service
AWS Server Migration Service

Networking & Content Delivery
Amazon VPC
Amazon CloudFront
Amazon Route 53
Amazon API Gateway
AWS Direct Connect
Elastic Load Balancing

Developer Tools
AWS CodeStar
AWS CodeCommit
AWS CodeBuild
AWS CodeDeploy
AWS CodePipeline
AWS Cloud9
AWS X-Ray
AWS Tools & SDKs

Management Tools
Amazon CloudWatch
AWS CloudFormation
AWS CloudTrail
AWS Config
AWS OpsWorks
AWS Service Catalog
AWS Systems Manager
AWS Trusted Advisor
AWS Personal Health Dashboard
AWS Command Line Interface
AWS Management Console
AWS Managed Services

Media Services
Amazon Elastic Transcoder
Amazon Kinesis Video Streams
AWS Elemental MediaConvert
AWS Elemental MediaLive
AWS Elemental MediaPackage
AWS Elemental MediaStore

Machine Learning
Amazon SageMaker
Amazon Comprehend
Amazon Lex
Amazon Polly
Amazon Rekognition
Amazon Machine Learning
Amazon Translate
Amazon Transcribe
AWS DeepLens
AWS Deep Learning AMIs
Apache MXNet on AWS
TensorFlow on AWS

Analytics
Amazon Athena
Amazon EMR
Amazon CloudSearch
Amazon Elasticsearch Service
Amazon Kinesis
Amazon Redshift
Amazon QuickSight
AWS Data Pipeline
AWS Glue

Security, Identity & Compliance
AWS Identity and Access Management (IAM)
Amazon Cloud Directory
Amazon Cognito
Amazon GuardDuty
Amazon Inspector
Amazon Macie
AWS Certificate Manager
AWS CloudHSM
AWS Directory Service
AWS Key Management Service
AWS Organizations
AWS Single Sign-On
AWS Shield

AR & VR
Amazon Sumerian

Application Integration
Amazon MQ
Amazon Simple Queue Service (SQS)
Amazon Simple Notification Service (SNS)
AWS AppSync
AWS Step Functions

Customer Engagement
Amazon Connect
Amazon Pinpoint
Amazon Simple Email Service (SES)

Business Productivity
Alexa for Business
Amazon Chime
Amazon WorkDocs
Amazon WorkMail

Desktop & App Streaming
Amazon WorkSpaces
Amazon AppStream 2.0

Internet of Things
AWS IoT Core
Amazon FreeRTOS
AWS Greengrass
AWS IoT 1-Click
AWS IoT Analytics
AWS IoT Button
AWS IoT Device Defender
AWS IoT Device Management

Game Development
Amazon GameLift
Amazon Lumberyard

Software

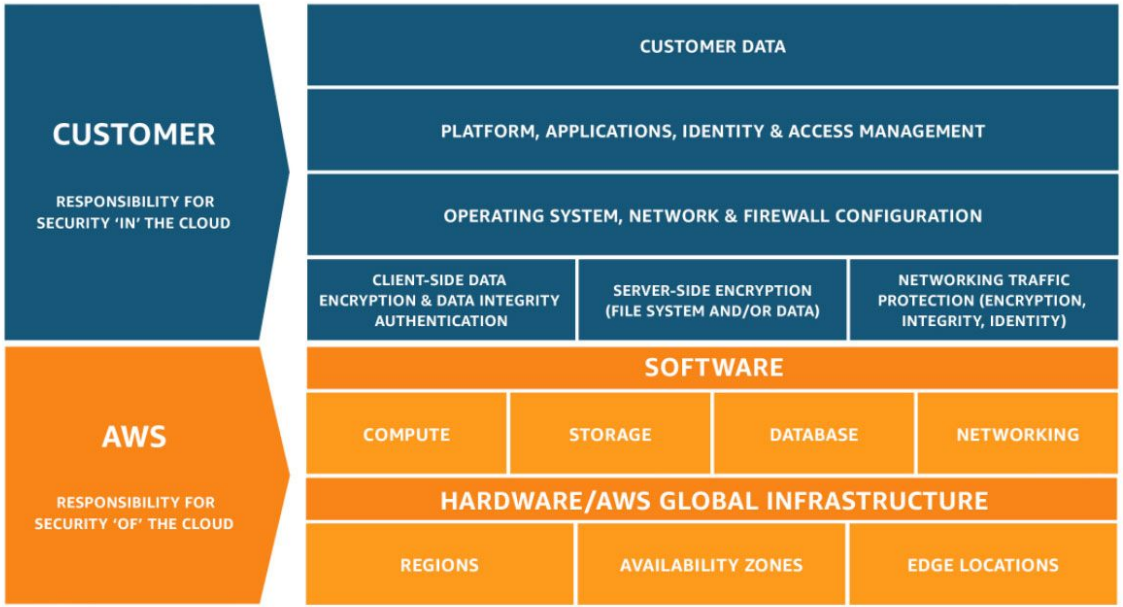
Cloud Security

Discussion

1. Who is responsible for security in the cloud? Think large cloud providers like Microsoft, Amazon, and Google.
2. Do you think security is a shared responsibility in the cloud?
   a. What resources are customers responsible for the security of?
   b. What resources are cloud providers responsible for the security of?
3. What are some security advantages/disadvantages to operating in the cloud?

https://tinyurl.com/4cc6vty6

Cloud Security – Shared Responsibility Model



https://aws.amazon.com/compliance/shared-responsibility-model/

Cloud Security

Discussion Continued...

1. Any surprises from the shared responsibility model published by AWS?
2. Do you agree with how resources are segmented? If not, what should be changed and why?

Recent Security Breach

- **Colonial Pipeline Ransomware Attack**
  - Security breach that took down the largest fuel pipeline in the U.S.
  - Led to fuel shortages across the East Coast
- Used a compromised password to get VPN access
  - Remote access to the company’s private network
- Once inside of the VPN, the attacker was able to control the pipeline management system
- Ransomware attack, which means that it threatens to publish the victim's personal data, block access and availability to systems unless a ransom is paid
- Should we assume the network is secure?
  - Wrong assumptions about the trusted computing base?

Zero Trust Networks

Federal Zero Trust Strategy

The Office of Management and Budget (OMB) is releasing a draft Federal Zero Trust Strategy in support of Executive Order 14028, "Improving the Nation's Cybersecurity", to adapt civilian agencies' enterprise security architecture to be based on zero trust principles.

Zero Trust Networks

- Calls for agencies to shift thinking so that they no longer assume that any networks or tools are — or will remain — secure.
- IP-based perimeters and access are replaced by ephemeral IP addresses and a constantly changing workforce with the need to access shared resources.
- IP-based access brittle in today's dynamic environments
- De-perimeterized, identity-based security
  - Human to machine
  - Machine to machine

Tradeoffs – What does security cost?

Discussion

1. What are some costs of increasing security in a system?
2. What can be gained by decreasing security in a system?

Security Versus Features

- Complexity can arise as features continue to be added to products and services
- Complexity is at odds with security
- Tradeoff with feature velocity
- Increasing security can have an opportunity cost (business perspective)

Security Versus Performance

- Security comes with an overhead
  - Authentication
  - Authorization
  - Cryptographic operations
- Bitcoin paper
  - Proof-of-work solves the problem of determining representation in majority decision making
  - Tradeoff performance for other security/system characteristics
- Performance can increase complexity, which decreases security

Security Versus Evolving Systems

- Large problem for security is that the full system continues to evolve even after the underlying security mechanisms are put in place
- Designer of the security mechanisms needs not only consider a wide range of attackers, but also to anticipate for future uses of the system
- Threat models change as the system evolves
- Evolving systems are at odds with security
- Principle of “least common mechanism” and “economy of mechanism” apply to evolving systems