Concepts and Responsibility

Cloud Security

CSE4059

An example: LabLink

- **LabLink**
  - A fictional university service (entirely made up, used as our running example)
  - Students submit project files; instructors give feedback; "**it runs on cloud infrastructure**"

- Story
  - 10pm, the night before a deadline.. a TA can't open one student's file; a permission error
  - The TA changes a storage *permission* to unblock that one student. It works!
  - The next morning: *anyone* holding the link to that storage location can read *multiple* students' submissions..
  - The cloud provider's infrastructure worked exactly as documented the entire time

What Makes This "Cloud"? Five Characteristics

1. On-demand self-service

- Resources can be provisioned without a human at the provider approving it

2. Broad network access

- Reachable over standard protocols from anywhere

3. Resource pooling (multi-tenancy)

- The same hardware serves many customers, separated logically

4. Rapid elasticity

- Capacity expands or contracts automatically, commensurate with demand

5. Measured service

- Usage is metered, and that usage data is itself observable

Each Characteristic Raises a New Security Question

Characteristic	Old assumption it breaks	New security question
Self-service	Only IT staff create infrastructure	Who is authorized to create resources, and how would we know?
Broad access	Sensitive systems only reachable from inside a building	Is every public interface public on purpose?
Pooling	“My hardware” is physically isolated	What guarantees isolation between tenants?
Elasticity	The set of running systems changes slowly	Can monitoring keep up with auto-appearing resources?
Measured service	-	Usage data itself can reveal patterns; who can see it?

Service Models: IaaS, PaaS, SaaS

- **Infrastructure as a Service (IaaS)**
  - Provider operates facility, hardware, virtualization – You get a virtual machine
  - You operate the Operating System (OS), runtime, application, and data
- **Platform as a Service (PaaS)**
  - Provider also operates the OS and runtime
  - You deploy code and manage data
- **Software as a Service (SaaS)**
  - Provider operates *everything* through the application layer
  - You control your data, and whatever user-specific configuration the app chooses to expose
- LabLink could be built as any of the three
  - They'd look identical to a student uploading a file

Who Operates Each Layer?

	IaaS	PaaS	SaaS
Facility	Provider	Provider	Provider
Hardware	Provider	Provider	Provider
Virtualization	Provider	Provider	Provider
OS	Customer	Provider	Provider
Runtime	Customer	Provider	Provider
Application	Customer	Customer	Provider
Data	Customer	Customer	Customer

Orange = Customer-operated
Blue = Provider-operated

Assets, Actors, and Trust Boundaries

- **Asset**
  - Anything of value that could be harmed (LabLink's files, credentials, availability)
- **Actor**
  - Any human or automated entity that interacts with the system
  - Ex) student, admin, TA, external attacker, the cloud provider itself
- **Trust boundary**
  - A line where the level of trust or verification changes

- **Management plane** (configuring cloud resources) vs. **data plane** (the app's normal work)
  - LabLink? the TA's fix was a MANAGEMENT-plane action → Its consequence showed up on the DATA plane

Five Threat Patterns That Recur Constantly

- **Stolen or leaked credentials**
  - A single leaked key can reach far more than a stolen keycard ever could
- **Exposed services and management interfaces**
  - Cloud resources default to reachable-from-anywhere unless restricted
- **Misconfiguration**
  - The single most common root cause of real cloud incidents
- **Insecure automation**
  - A script that embeds a secret or grants broad access, run without review
- **Multi-tenancy / isolation failure**
  - Rare among major providers, but conceptually important

Example: A Small Mistake, an Automated Attacker

- A deployment script accidentally commits a cloud access key to a public code repository (ex. GitHub)
  - Insecure automation + Credential exposure
- Within minutes, automated scanners operated by attackers can find and use it
- This combination could be real cloud breaches

LabLink Incident – Which Pattern Was It?

- Not stolen credentials, not an exposed interface, not insecure automation, not multi-tenancy failure.

- It was misconfiguration
  - A storage permission left broader than intended...

- The most common pattern is also the most easily overlooked as "not a real threat”

Resolving Tonight's Puzzle

- The cloud provider's storage service worked **exactly** as documented, the whole time
- The provider met its side of what's called the shared responsibility model
- The breach happened entirely within the customer's side
  - LabLink's own team was responsible for correct configuration
- "The provider secures its infrastructure" was never a complete answer by itself

Shared Responsibility

Who is responsible for what

The Shared Responsibility Model

- In any cloud arrangement
  - Some controls are the provider's, some are the customer's, some are shared
  - This division is a property of the architecture and the contract – Not automatic

- Security ***of*** the cloud
  - Facilities, hardware, network, virtualization (provider)

- Security ***in*** the cloud
  - Configuration, access control, application logic, data (customer)

- Misconception?
  - A secure provider does no make your deployment automatically secure

Two Sides of One System

- Cloud systems have two sides of responsibility:
  - What the provider promises and operates
  - What the customer must configure and operate
  - Neither side substitutes for the other

- A perfect provider cannot rescue a careless customer configuration
- A careful customer cannot compensate for a provider that fails its own side

- Two sides are not symmetric in practice
  - Provider side: broad, but standardized, audited, and done once for everyone
  - Your side: narrower, but bespoke, unaudited, and done by you

Responsibility Across IaaS, PaaS, SaaS

	IaaS	PaaS	SaaS
Physical / data center	Provider	Provider	Provider
Hardware & virtualization	Provider	Provider	Provider
OS patching	Customer	Provider	Provider
Network configuration	Customer	Shared	Provider
Identity & access config	Customer	Customer	Customer
Application logic	Customer	Customer	Provider
Data classification & handling	Customer	Customer	Customer

Customer responsibility shrinks toward SaaS, but never reaches zero

Less responsibility =/= Less risk

One Control, Two Models: Who Is Responsible?

- Q) Who is responsible for making sure the submissions database is not publicly readable?
  - [Case A] LabLink on IaaS
    - A database you installed on a virtual machine you rent
  - [Case B] LabLink on SaaS
    - A managed service where you never touch a server at all
- Same question, two different amounts of machinery underneath

Both Times, It Is You

- Both times, you

- [Case A] Under IaaS
  - You configure the database, the host, and the network in front of it
- [Case B] Under SaaS
  - You configure far less; almost everything belongs to the provider

- But "who may read this" is still one of the few things left to you

- Why?
  - No provider can know who is supposed to see your students' submissions

A Certified Provider, an Insecure Customer

- Suppose LabLink's cloud provider holds a well-known, widely respected security certification
- LabLink's team still leaves an access permission policy far broader than necessary

- The provider's certification is still true and still meaningful
  - For the provider's own scope
- But it says nothing about LabLink’s access permissions

- Both statements are true at once
  - No contradiction between '*the provider is certified*' and '*the customer is insecure*'

Summary

- Cloud characteristics raise new security questions
  - Self-service, access, pooling, elasticity, metering
- IaaS / PaaS / SaaS divide who operates and, separately, who is responsible for
- Shared responsibility
  - Provider and customer each own different layers – neither substitutes for the other
- A provider's certification proves its own scope – not your configuration