# Fundamentals of Data Engineering - Reis & Housley

## Part III. Security, Privacy, and the Future of Data Engineering

### Chapter 10 – Security and Privacy

* Security is vital to the practice of DE
* This should be blindingly obvious, but often DE’s view security as an afterthought
* **Security should be one of the first things, if not the first thing, a DE needs to think about in every aspect of their job + every stage of the DE lifecycle**
* You deal w/ sensitive data, information, + access daily
* Your organization, customers, + business partners expect these valuable assets to be handled with the utmost care and concern
* One security breach or a data leak can leave your business dead in the water, + your career and reputation are ruined if it’s your fault
* **Security is a key ingredient for privacy**
* Privacy has long been critical to trust in the corporate IT space, since engineers directly or indirectly handle data related to people’s private lives
* This includes financial information, data on private communications (emails, texts, phone calls), medical history, educational records, + job history
* A company that leaked this information or misused it could find itself a pariah when the breach came to light
* **Increasingly, privacy is a matter of significant *legal* importance**
* Ex: The Family Educational Rights and Privacy Act (FERPA) went into effect in the US in the 1970s, the Health Insurance Portability and Accountability Act (HIPAA) followed in the 1990s, GDPR was passed in Europe in the mid-2010s, + several US-based privacy bills have passed or will soon
* This is just a tiny sampling of privacy-related statutes (and just the beginning)
* Still, the penalties for violation of any of these laws can be significant, even devastating, to a business
* And b/c data systems are woven into the fabric of education, health care, and business, DE’s handle sensitive data related to each of these laws.
* **A DE’s *exact* security + privacy responsibilities will vary significantly between organizations**
* At a small startup, a DE may do double duty as a data security engineer
* A large tech company will have armies of security engineers + security researchers
* Even in this situation, DE’s will often be able to identify security practices + technology vulnerabilities w/in their own teams + systems that they can report + mitigate in collaboration w/ dedicated security personnel

#### People

* **The weakest link in security + privacy is *you***
* Security is **often compromised at the human level**, so conduct yourself as if you’re always a target
* Act as if a bot or human actor is trying to infiltrate your sensitive credentials + information at any given time
* This is reality, + it’s not going away
* **Take a defensive posture with everything you do, online *and* offline**
* Exercise the **power of negative thinking** and **always be paranoid**.

##### The Power of Negative Thinking

* American surgeon Atul Gawande wrote a 2007 op-ed in the NYT w/ a central thesis is that positive thinking can blind us to the possibility of terrorist attacks or medical emergencies + deter preparation
* ***Negative* thinking allows us to consider disastrous scenarios and act to prevent them**
* **DE’s should actively think through the scenarios for data utilization + collect sensitive data *only if there is an actual need downstream***
* **The best way to protect private + sensitive data is to avoid ingesting this data in the first place**
* **DE’s should think about the attack + leak scenarios w/ any data pipeline or storage system they utilize**
* **When deciding on security strategies, ensure that your approach delivers proper security + not just the *illusion* of safety.**

##### Always Be Paranoid

* **Always exercise caution when someone asks you for your credentials**
* When in doubt (*always be in extreme doubt when asked for credentials*), hold off + get second opinions from coworkers + friends
* Confirm w/ other people that the request is indeed legitimate
* A quick chat or phone call is cheaper than a ransomware attack triggered through an email click
* **Trust nobody at face value when asked for credentials, sensitive data, or confidential information, *including from your coworkers***
* **You are also the first line of defense in respecting privacy and ethics**
* Are you uncomfortable w/ sensitive data you’ve been tasked to collect?
* Do you have ethical questions about the way data is being handled in a project?
* Raise your concerns w/ colleagues + leadership
* **Ensure that your work is both legally compliant and ethical**

#### Processes

* When people follow regular security processes, security becomes part of the job.
* **Make security a habit, regularly practice *real* security, exercise the principle of least privilege, and understand the shared responsibility model in the cloud**

##### Security Theater Vs. Security Habit

* W/ corporate clients, there’s a pervasive focus on compliance (w/ internal rules, laws, recommendations from standards bodies, etc.), but **not enough attention to potentially bad scenarios**
* Unfortunately, this **creates an *illusion* of security but often leaves gaping holes that would be evident with a few minutes of reflection**
* **Security needs to be simple +effective enough to become habitual throughout an organization**
* There are a number of companies w/ security policies in the hundreds of pages that nobody reads, along w/ the annual security policy review that people immediately forget, all in checking a box for a security audit
* This is **security theater, where security is done in the letter of compliance** (SOC-2, ISO 27001, and related) **without real *commitment***
* Instead, **pursue a spirit of genuine + habitual security, + bake a security mindset into your culture**
* **Security doesn’t need to be complicated**
* Ex: Run security training + policy review at least once a month to ingrain this into your team’s DNA + update each other on security practices you can improve
* **Security must not be an afterthought for your data team**
* ***Everyone* is responsible + has a role to play**
* **It *must* be the priority for you and everyone else you work w/**

##### Active Security

* Returning to the idea of negative thinking, **active securityentails thinking about + researching security threats in a dynamic + changing world**
* Rather than simply deploying scheduled simulated phishing attacks, take an active security posture by researching successful phishing attacks + thinking through your organizational security vulnerabilities
* Rather than simply adopting a standard compliance checklist, think about internal vulnerabilities specific to your organization + incentives employees might have to leak or misuse private information
* *This is discussed more later*

##### The Principle of Least Privilege

* The **principle of least privilege**means that **a person or system should be given only the privileges and data they need to complete the task at hand *and nothing more***
* Often, one can see **an antipattern in the cloud: a regular user is given admin access to *everything*, when that person may need just a handful of IAM roles to do their work**
* **Giving someone carte blanche admin access is a huge mistake + should *never* happen under the principle of least privilege**
* Instead, **provide the user (or group they belong to) the specific IAM roles they need *when they need them***
* **When these roles are no longer needed, take them away**
* ***The same rule applies to service accounts***
* Treat humans and machines the same way: give them only the privileges + data they need to do their jobs, + *only for the timespan when needed*
* Of course, **the principle of least privilege is also critical to privacy**
* Your users + customers expect that people will look at their sensitive data *only when necessary*, so make sure that this is the case
* **Implement column, row, + cell-level access controls around sensitive data**, consider **masking PII + other sensitive data**, + **create views that contain *only* the information the viewer needs to access**
* **Some data must be retained but should be accessed only in an emergency**
* Put this data behind a **broken glass process: users can access it only after going through an emergency approval process to fix a problem, query critical historical information, etc.**
* Access is revoked *immediately* once the work is done

##### Shared Responsibility in the Cloud

* **Security is a *shared* responsibility in the cloud**
* The **cloud vendor is responsible for ensuring the physical security of its data center + hardware**
* At the same time, ***you* are responsible for the security of the applications + systems you build and maintain in the cloud**
* **Most cloud security breaches continue to be caused by end users, not the cloud**
* Breaches occur because of unintended misconfigurations, mistakes, oversights, and sloppiness

##### Always Back Up Your Data

* Data disappears, sometimes it’s a dead hard drive or server, + in other cases, someone might accidentally delete a database or an object storage bucket
* A bad actor can also lock away data, + ransomware attacks are widespread these days
* Some insurance companies are reducing payouts in the event of an attack, leaving you on the hook both to recover your data *and* pay the bad actor who’s holding it hostage
* **You need to back up your data regularly, both for disaster recovery *and* continuity of business operations (if a version of your data is compromised in a ransomware attack)**
* Additionally, **test the restoration of your data backups on a regular basis**
* Data backup doesn’t *strictly* fit under security + privacy practices, + instead it goes under the larger heading of **disaster prevention**, but it’s **adjacent to security**, especially in the era of ransomware attacks

##### An Example Security Policy

* This section presents a sample security policy regarding credentials, devices, + sensitive information
* Notice we don’t overcomplicate things + instead give people a short list of practical actions they can take immediately
* **Example Security Policy (w/ ground rules)**
* **Protect your credentials at all costs**
* **Use a single-sign-on (SSO) for *everything***
* **Avoid passwords whenever possible**, + use **SSO as the default**
* **Use multifactor authentication with SSO**
* **Don’t share passwords or credentials** (includes client passwords and credentials)
* If in doubt, see the person you report to
* If that person is in doubt, keep digging until you find an answer
* **Beware of phishing and scam calls**
* Don’t ever give your passwords out (Again, prioritize SSO)
* **Disable or delete old credentials**
* *Preferably the latter*
* **Don’t put credentials in code**
* **Handle secrets as *configuration* and *never* commit them to version control**
* **Use a secrets manager where possible.**
* **Always exercise the principle of least privilege**
* **Never give more access than is required to do the job**
* This **applies to *all* credentials + privileges in the cloud *and* on premises**
* **Protect Your Devices**
* **Use device management for *all* devices used by employees**
* If an employee leaves the company or your device gets lost, the device can be remotely wiped
* **Use multifactor authentication for all devices.**
* **Sign into your device using your company email credentials.**
* **All policies covering credentials + behavior apply to your device(s).**
* **Treat your device as an extension of yourself**
* **Don’t let your assigned device(s) out of your sight**
* When **screen sharing**, be aware of *exactly* what you’re sharing to protect sensitive information + communications
* Share only single documents, browser tabs, or windows, + **avoid sharing your full desktop**
* **Share only what’s required to convey your point**
* Use “do not disturb” mode when on video calls
* This prevents messages from appearing during calls or recordings
* **Software Update Policy**
* Restart your web browser when you see an update alert.
* Run minor OS updates on company + personal devices.
* The company will identify critical major OS updates and provide guidance.
* Don’t use the beta version of an OS.
* Wait a week or two for new major OS version releases.
* These are some basic examples of how security can be simple and effective
* **Based on your company’s security profile, you may need to add more requirements for people to follow**
* And again, **always remember that people are your weakest link in security**

#### Technology

* After you’ve addressed security w/ people + processes, it’s time to look at how you leverage tech to secure your systems + data assets
* The following are some significant areas you should prioritize.

##### Patch and Update Systems

* **Software gets stale, + security vulnerabilities are constantly discovered**
* **To avoid exposing a security flaw in an older version of the tools you’re using, always patch and update OS’s + software as new updates become available**
* Thankfully, many SaaS + cloud-managed services automatically perform upgrades + other maintenance w/out your intervention
* **To update your *own* code and dependencies, either automate builds or set alerts on releases and vulnerabilities so you can be prompted to perform the updates manually**

##### Encryption

* **Encryption is not a magic bullet, as it will do little to protect you in the event of a *human* security breach that grants access to credentials**
* **Encryption is a *baseline* requirement for *any* organization that respects security and privacy**
* **It will protect you from *basic* attacks**, such as network traffic interception.
* Let’s look separately at **encryption at rest and in transit**

###### Encryption At Rest

* **Be sure your data is encrypted when it is at rest (on a storage device)**
* Company **laptops should have full-disk encryption enabled to protect data if a device is stolen**
* **Implement server-side encryption for all data stored in servers, filesystems, databases, + object storage in the cloud**
* **All data backups for archival purposes should also be encrypted**
* Finally, **incorporate application-level encryption where applicable**

###### Encryption Over the Wire

* **Encryption over the wire is now the default for current protocols**
* Ex: HTTPS is generally required for modern cloud APIs
* **DE’s should always be aware of how keys are handled, as bad key handling is a significant source of data leaks**
* In addition, **HTTPS does nothing to protect data if bucket permissions are left open to the public, another cause of several data scandals over the last decade**
* DE’s should also be aware of the **security limitations of older protocols**
* Ex: FTP is simply *not* secure on a public network
* While this may not appear to be a problem when data is already public, FTP is vulnerable to man-in-the-middle attacks, whereby an attacker intercepts downloaded data + changes it before it arrives at the client
* **It is best to simply avoid FTP**
* **Make sure everything is encrypted over the wire, even w/ legacy protocols**
* **When in doubt, use robust technology with encryption baked in**

##### Logging, Monitoring, and Alerting

* Hackers + bad actors typically don’t *announce* that they’re infiltrating your systems
* **Most companies don’t find out about security incidents until well after the fact**
* **Part of DataOps is to observe, detect, + alert on incidents**
* **As a DE, you should set up automated monitoring, logging, + alerting to be aware of peculiar events when they happen in your systems**
* **If possible, set up automatic anomaly detection.**
* Here are some areas you should monitor:
* **Access**
* **Who’s accessing what, when, and from where?** What **new accesses** were granted?
* Are there **strange patterns** w/ current users that might indicate an account is compromised, such as trying to access systems they don’t usually access or shouldn’t have access to?
* Do you see new unrecognized users accessing your system?
* **Be sure to regularly comb through access logs, users, + their roles to ensure that everything looks OK**
* **Resources**
* **Monitor your disk, CPU, memory, + I/O for patterns that seem out of the ordinary**
* Did your resources suddenly change?
* If so, this might indicate a security breach
* **Billing**
* **Especially w/ SaaS + cloud-managed services**, you need to **oversee costs**
* Set up **budget alerts** to make sure your spending is w/in expectations
* If an unexpected spike occurs in billing, this might indicate someone or something is utilizing your resources for malicious purposes.
* **Excess permissions**
* Increasingly, vendors are providing **tools that monitor for permissions that are *not* utilized** by a user or service account over some time
* These **tools can often be configured to automatically alert an admin or remove permissions** after a specified elapsed time
* Ex: Suppose a particular analyst hasn’t accessed Redshift for 6 months
* These permissions can be removed, closing a potential security hole
* If the analyst needs to access Redshift in the future, they can put in a ticket to restore permissions
* It’s best to **combine these areas in your monitoring to get a cross-sectional view of your resource, access, + billing profile**
* Try setting up a dashboard for everyone on the data team to view monitoring + receive alerts when something seems out of the ordinary
* Couple this w/ an effective incident response plan to manage security breaches when they occur, + run through the plan on a regular basis so you are prepared

##### Network Access

* You can see DE’s sometimes doing pretty wild things regarding **network access**
* In several instances, there have been publicly available Amazon S3 buckets that housed lots of sensitive data, or Amazon EC2 instances w/ inbound SSH access open to the whole world for 0.0.0.0/0 (all IPs), or databases with open access to all inbound requests over the public internet
* These are just a few examples of terrible network security practices
* **In *principle*, network security should be left to security experts at your company**
* **In *practice*, you may need to assume significant responsibility for network security in a small company**
* **As a DE, you will encounter databases, object storage, + servers so often that you should at least be aware of simple measures you can take to make sure you’re in line w/ good network access practices**
* **Understand what IPs + ports are open, to whom, + why**
* **Allow the incoming IP addresses of the systems + users that will access these ports (AKA whitelisting IPs) + *avoid broadly opening connections for any reason***
* **When accessing the cloud or a SaaS tool, use an encrypted connection**
* Ex: Don’t use an unencrypted website from a coffee shop
* **For hosting on-prem servers**, recall Chapter 3 + the difference between a **hardened perimeter** and **zero-trust security**
* **The cloud is generally closer to zero-trust security (every action requires authentication)**
* **The cloud is an arguably more secure option for most organizations *b/c* it imposes zero-trust practices + allows companies to leverage the army of security engineers employed by the public clouds**
* ***However*, sometimes hardened perimeter security still makes sense**
* Ex: Nuclear missile silos are air gapped (not connected to any networks)
* **Air-gapped servers are the ultimate example of a hardened security perimeter**
* **Just keep in mind that even on-prem, air-gapped servers are vulnerable to *human* security failings**

##### Security for Low-Level Data Engineering

* **For DE’s who work in the guts of data storage + processing systems, it is critical to consider the security implications of *every* element**
* **Any software library, storage system, or compute node is a potential security vulnerability**
* **A flaw in an obscure logging library might allow attackers to bypass access controls or encryption**
* **Even CPU architectures + microcode represent potential vulnerabilities, since sensitive data can be vulnerable when it’s at rest in memory or a CPU cache**
* **No link in the chain can be taken for granted**
* We’ve been talking principally about *high-level* DE (stitching together tools to handle the entire DE lifecycle)
* Thus, the gory technical details are left to your own research

###### Internal Security Research

* We discussed the idea of **active security****for *processes***, but it’s also highly recommended to adopt an **active security approach to *technology***
* Specifically, this means that ***every* technology employee should think about security problems**
* ***Why is this important?***
* **Every technology contributor develops a domain of technical expertise**
* **Even if your company employs an army of security researchers, DE’s will become intimately familiar w/ specific data systems + cloud services in their purview**
* **Experts in a particular technology are well positioned to identify security holes in this technology**
* **Encourage every DE to be *actively* involved in security**
* **When they identify potential security risks in their systems, they should think through mitigations + take an active role in deploying these**

#### Conclusion

* **Security needs to be a *habit* of mind *and* action**, so treat data like a wallet or smartphone
* Although you won’t likely be in charge of security for your company, **knowing basic security practices + keeping security top of mind will help reduce the risk of data security breaches at your organization**

#### Additional Resources

* *Building Secure and Reliable Systems* by Heather Adkins et al. (O’Reilly)
* Open Web Application Security Project (OWASP) publications: <https://owasp.org/>
* *Practical Cloud Security* by Chris Dotson (O’Reilly)