GLOC

Great Lakes Oracle Conference

Reducing The Surface Area Of Risk Using Data Masking

MAY 16 & 17, 2018

CLEVELAND PUBLIC AUDITORIUM,
CLEVELAND, OHIO

WWW.NEOOUG.ORG/GLOC

Who am I?

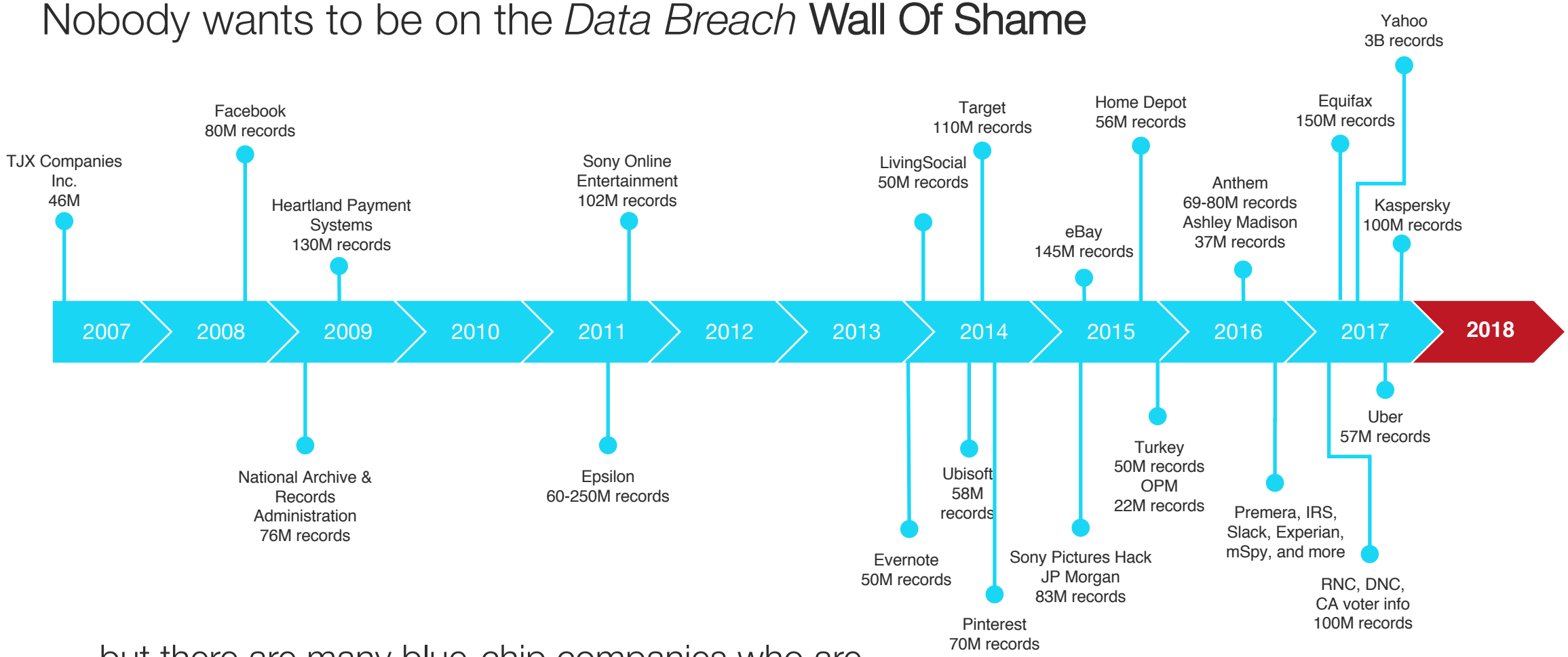
- DBA for 25+ years
  - “C” programmer prior to that
- Co-authored six Oracle books
  - Tech review on eight more
- Field services at Delphix
  - 16 years independent consultant prior to that
    - http://EvDBT.com/
- Married to @DBAKevlar
    - http://DBAKevlar.com/
  - That is our *new* home parked in front of our current home ➔ ➔ ➔
    - http://DancesWithWinnebagos.com

Agenda

1. Fear and loathing
2. External and internal threats
3. Data masking
4. Summary

Fear and Loathing

Nobody wants to be on the *Data Breach* Wall Of Shame



…but there are many blue-chip companies who are

Fear and Loathing

- The attitude of many is…

We have a firewall. We're good.

- In the 1930s, France built an enormous fortification known as the **Maginot Line**
  - It was designed specifically to prevent Germany from **ever** invading
  - Every military expert worldwide agreed that it was ***impregnable***

Nous avons la ligne Maginot! Que peuvent faire les Boche?

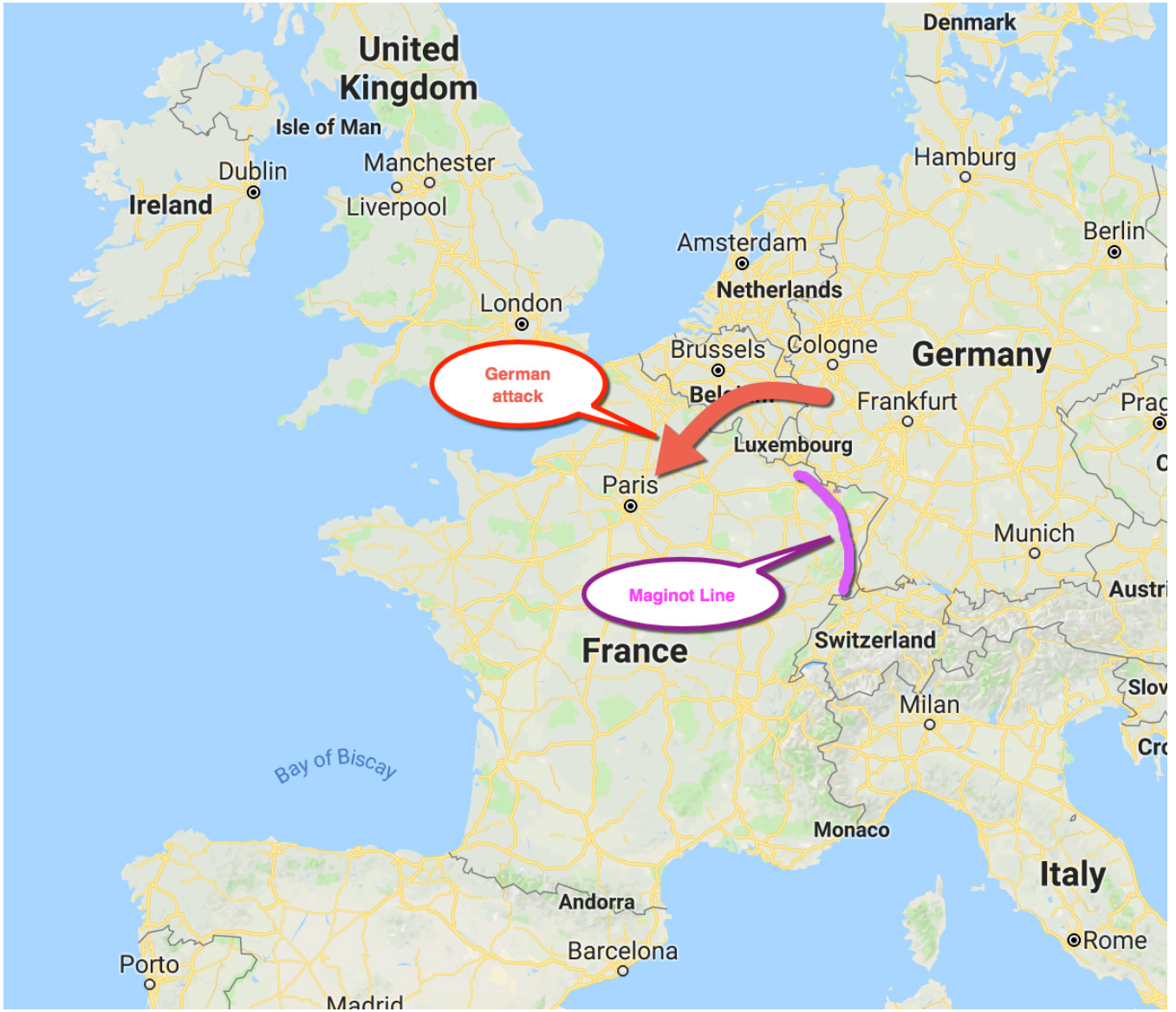
- In 1940, Germany conquered France in 6 weeks

Fear and Loathing

- Germany simply *bypassed* the Maginot Line and conquered France in 6 weeks

Fear and Loathing

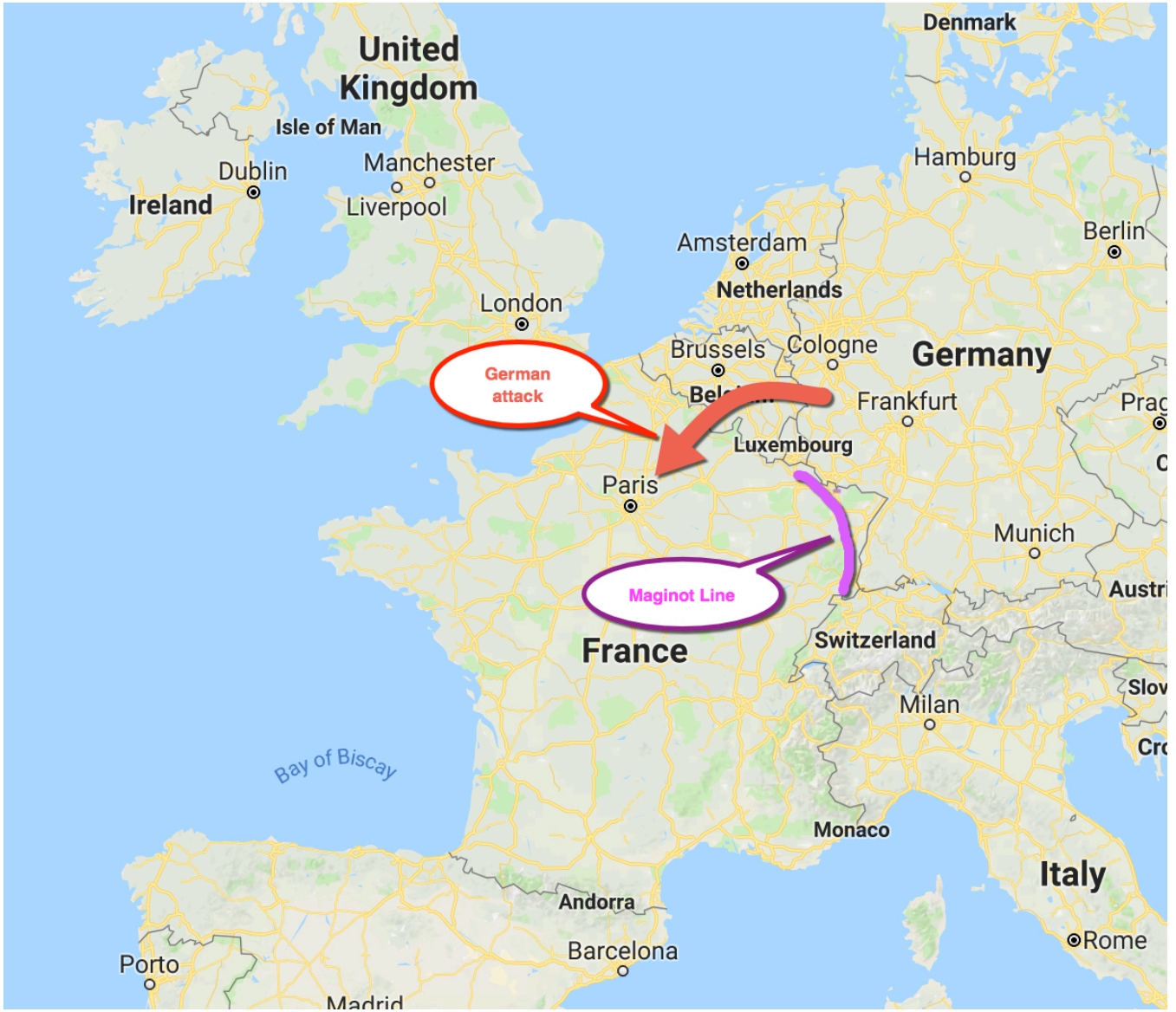
- Germany simply *bypassed* the Maginot Line and conquered France in 6 weeks



Fear and Loathing

- Germany simply *bypassed* the Maginot Line and conquered France in 6 weeks

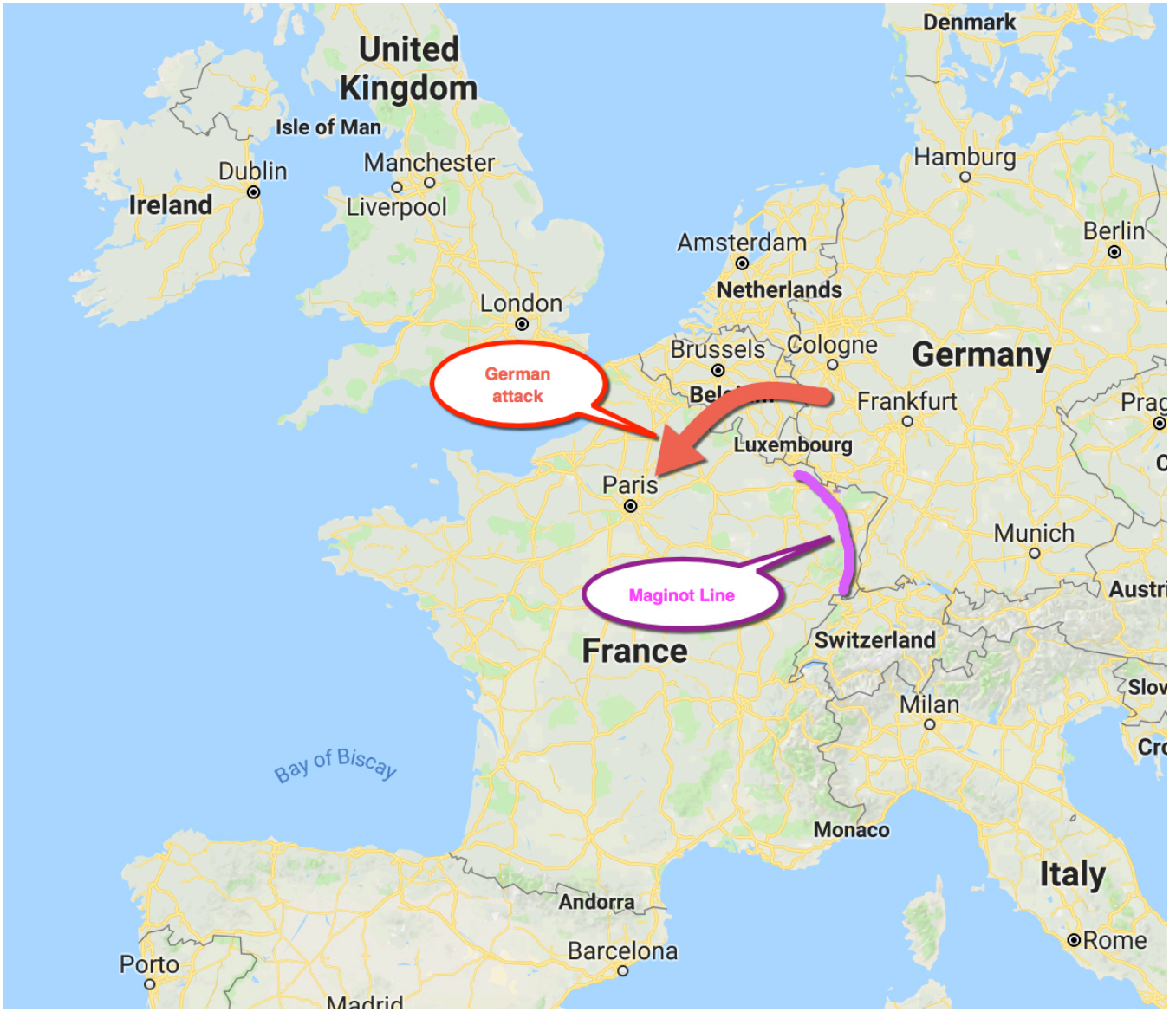
- *Lessons learned*:



Fear and Loathing

- Germany simply *bypassed* the Maginot Line and conquered France in 6 weeks

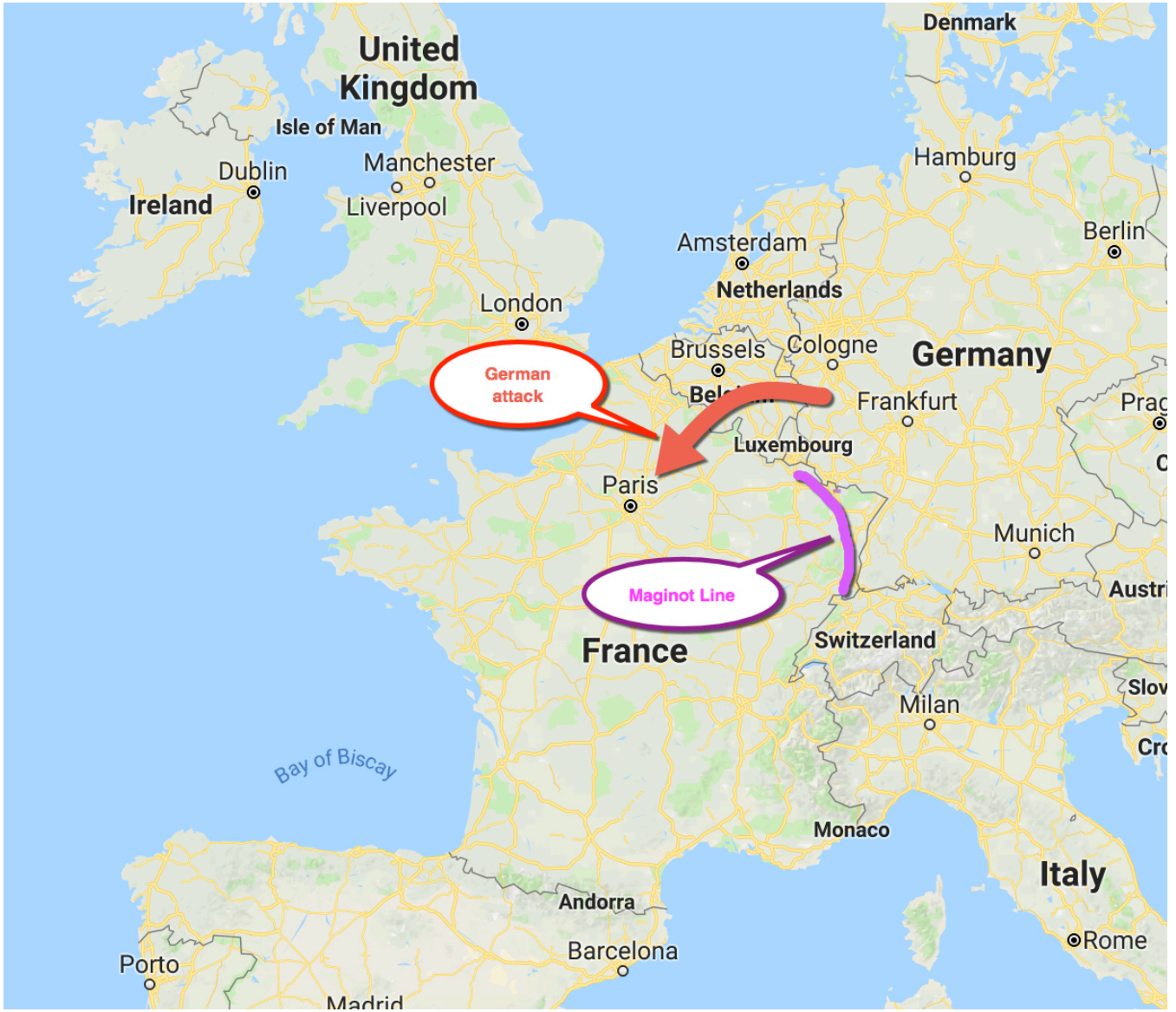
- *Lessons learned*:
  1. Use multiple layers of defense
     - *Do **not** rely on a single strong defense against a single threat*



Fear and Loathing

- Germany simply *bypassed* the Maginot Line and conquered France in 6 weeks

- *Lessons learned*:
  1. Use multiple layers of defense
     - *Do **not** rely on a single strong defense against a single threat*
  2. Create strongpoints and concentrate defenses within
     - *Impossible to defend **everything** equally, so **prioritize** and **focus***



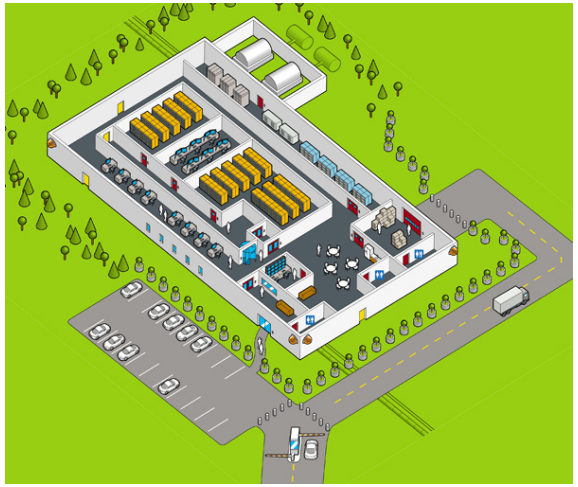
Agenda

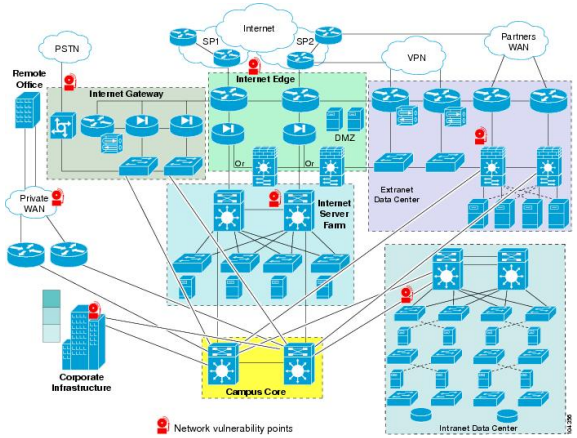
1. Fear and loathing
2. External and internal threats
3. Data masking
4. Summary

External and internal threats

How do we apply these lessons to prevent data breaches?

1. Layered defense
   a) Physical security of data center
   b) Network security (firewalls)
   c) Strong authentication to services and servers
   d) Centralized rule-based authorization to services and servers
   e) Encryption of data in-flight
   f) Encryption of data at-rest
2. Reduce the surface area of risk
   a) Prioritize and focus protection efforts on production systems
   b) Mask (obfuscate) sensitive/confidential data in non-production systems

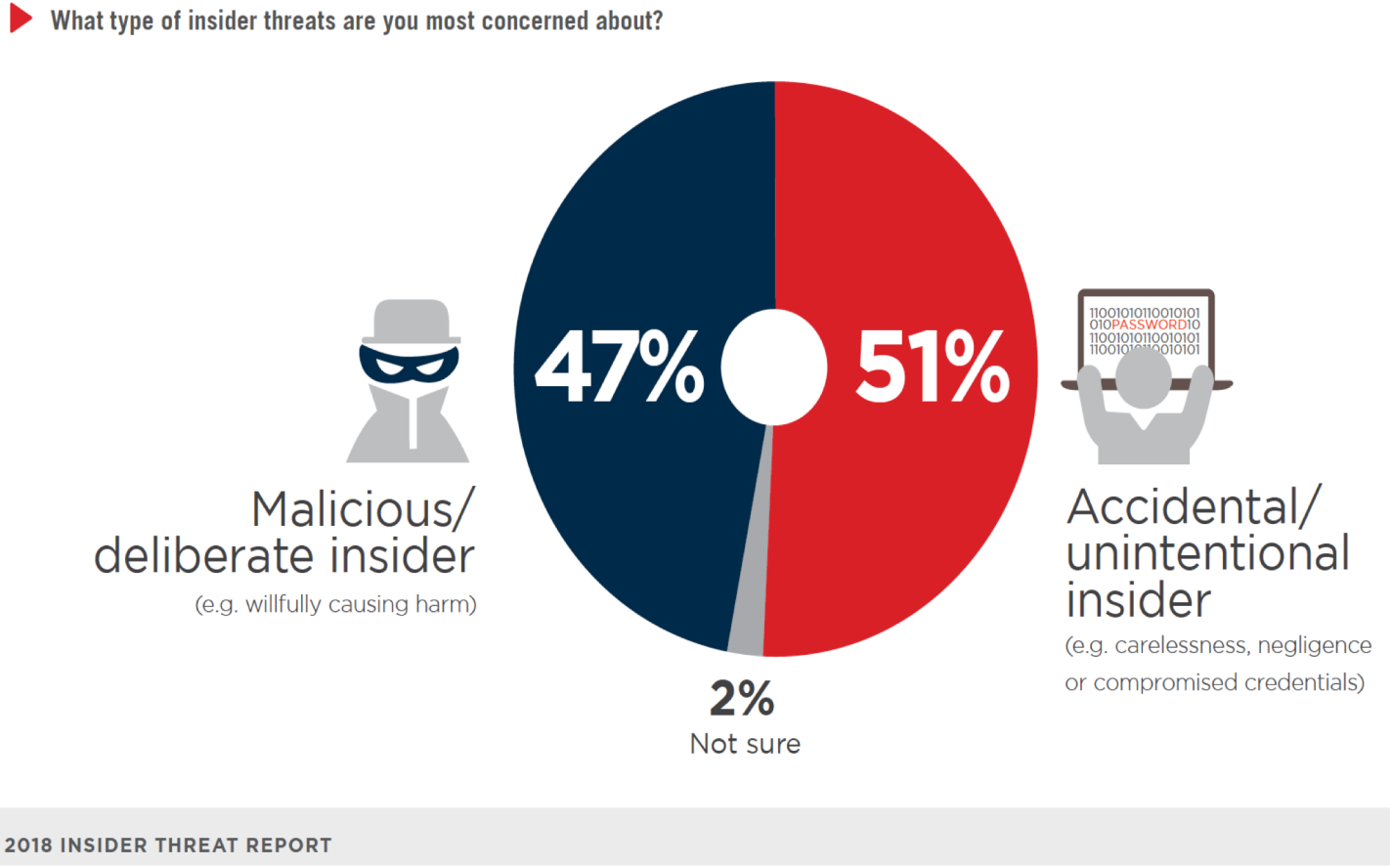




External and internal threats

In addition to attacks from **external vectors**, there is growing realization about the nature of **insider threats**

- **90%** of organizations *feel vulnerable* to insider attack
  - The main enabling risk factors include…
    - too many users with excessive access privileges (**37%**)
    - an increasing number of devices with access to sensitive data (**36%**)
    - increasing complexity of information technology (**35%**)
- **53%** *confirmed* insider attacks against their organization in the **previous 12 months**
  - Typically fewer than 5 attacks, but **27%** say insider attacks have become more frequent

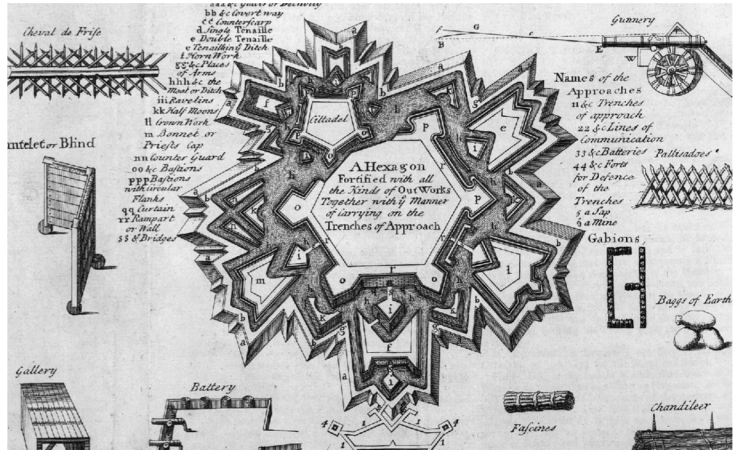


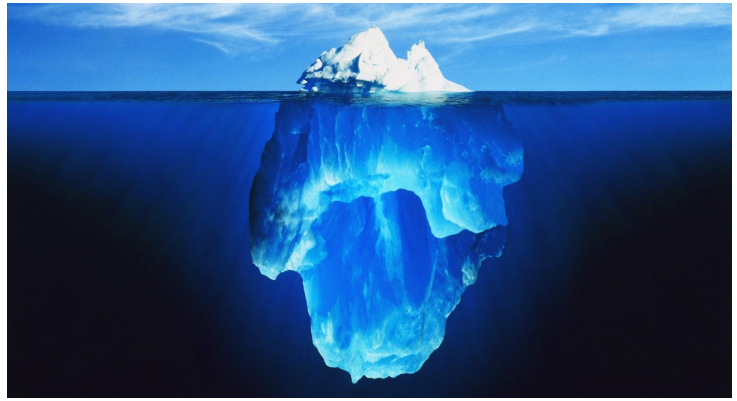
Courtesy of: *2018 Insider Threat Report – Cybersecurity-Insiders.com and Crowd Research Partners*

External and internal threats

How do we apply these lessons to prevent data breaches?

1. Layered defense
   a) Physical security of data center
   b) Network security (firewalls)
   c) Strong authentication to services and servers
   d) Centralized rule-based authorization to services and servers
   e) Encryption of data in-flight
   f) Encryption of data at-rest
2. Reduce the surface area of risk
   a) Prioritize and focus protection efforts on production systems
   b) Mask (obfuscate) sensitive/confidential data in non-production systems





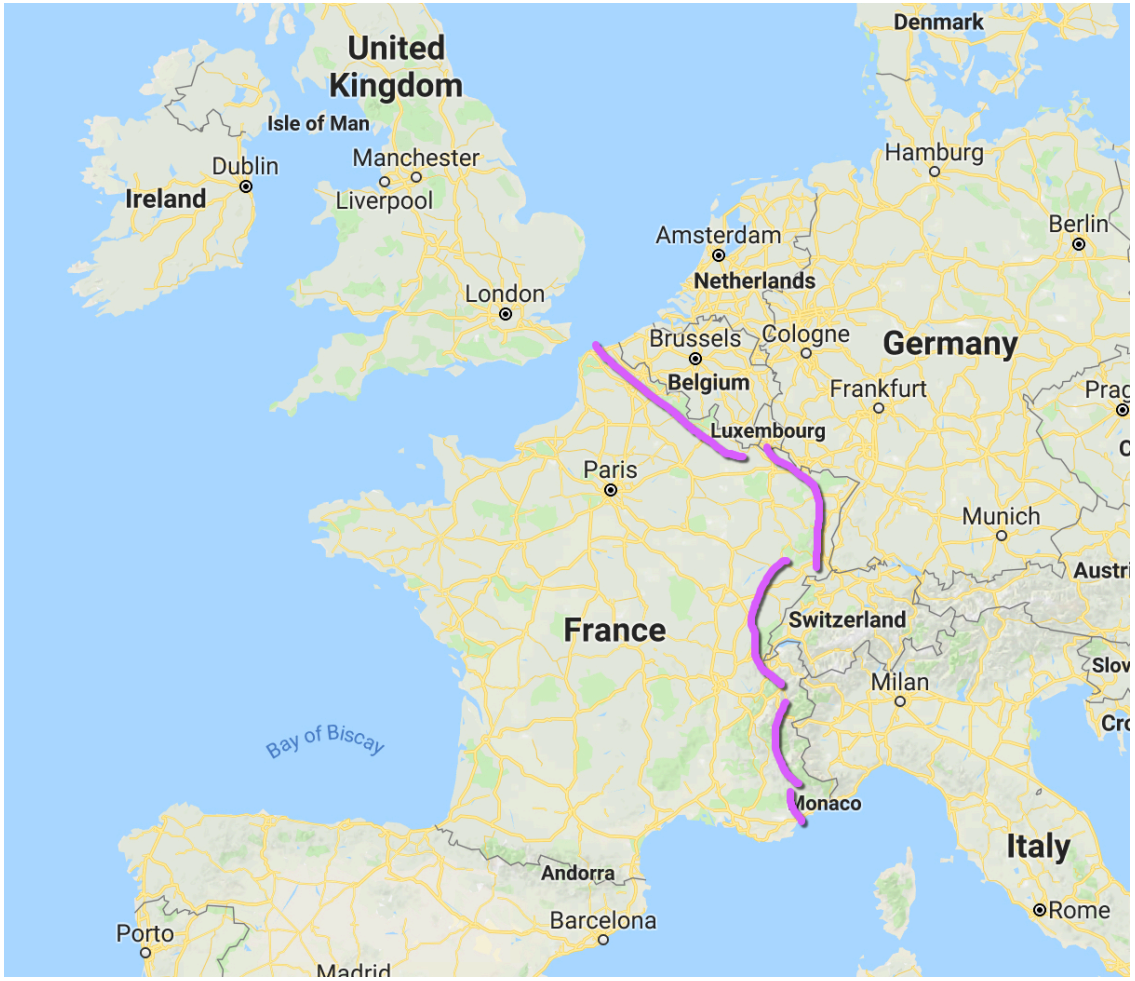
External and internal threats

External and internal threats

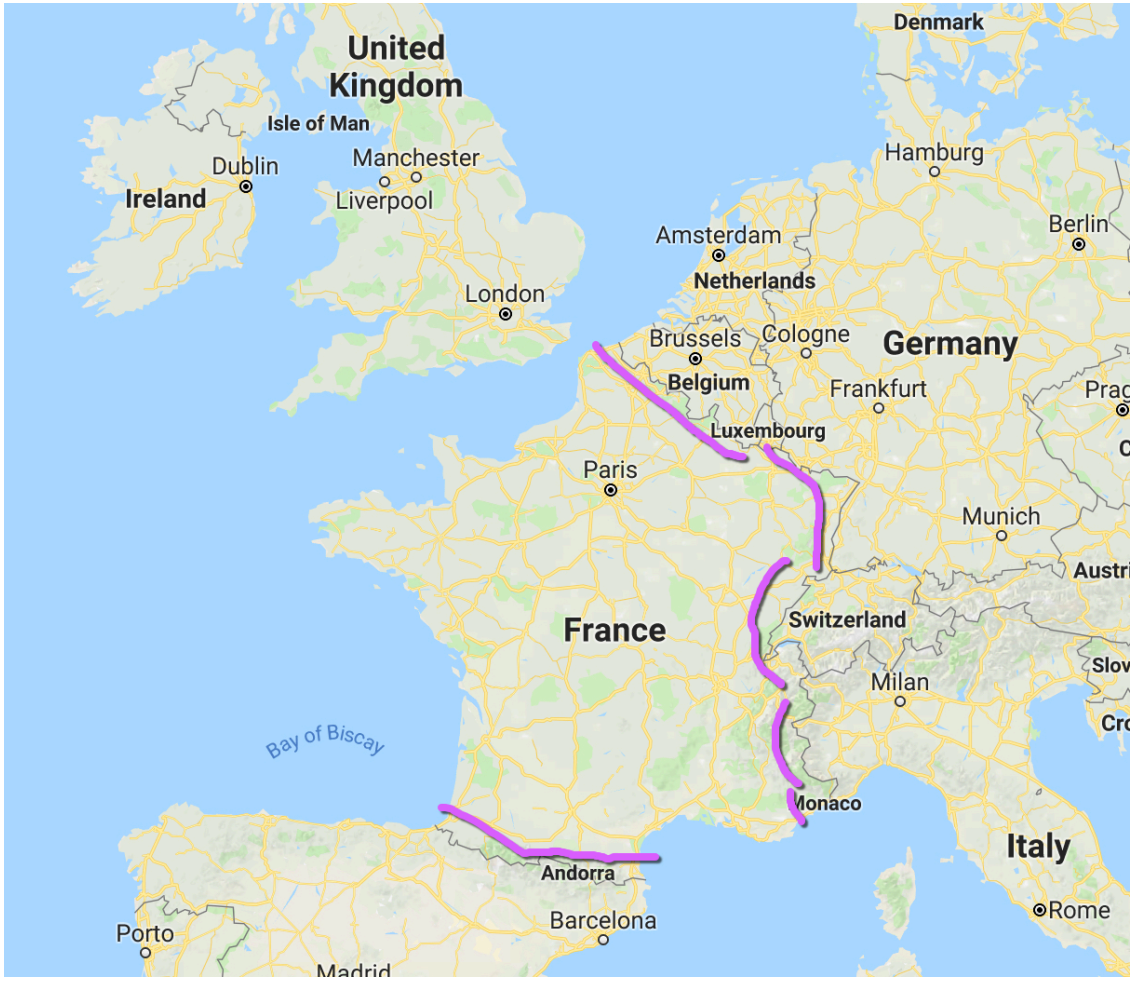
External and internal threats

External and internal threats

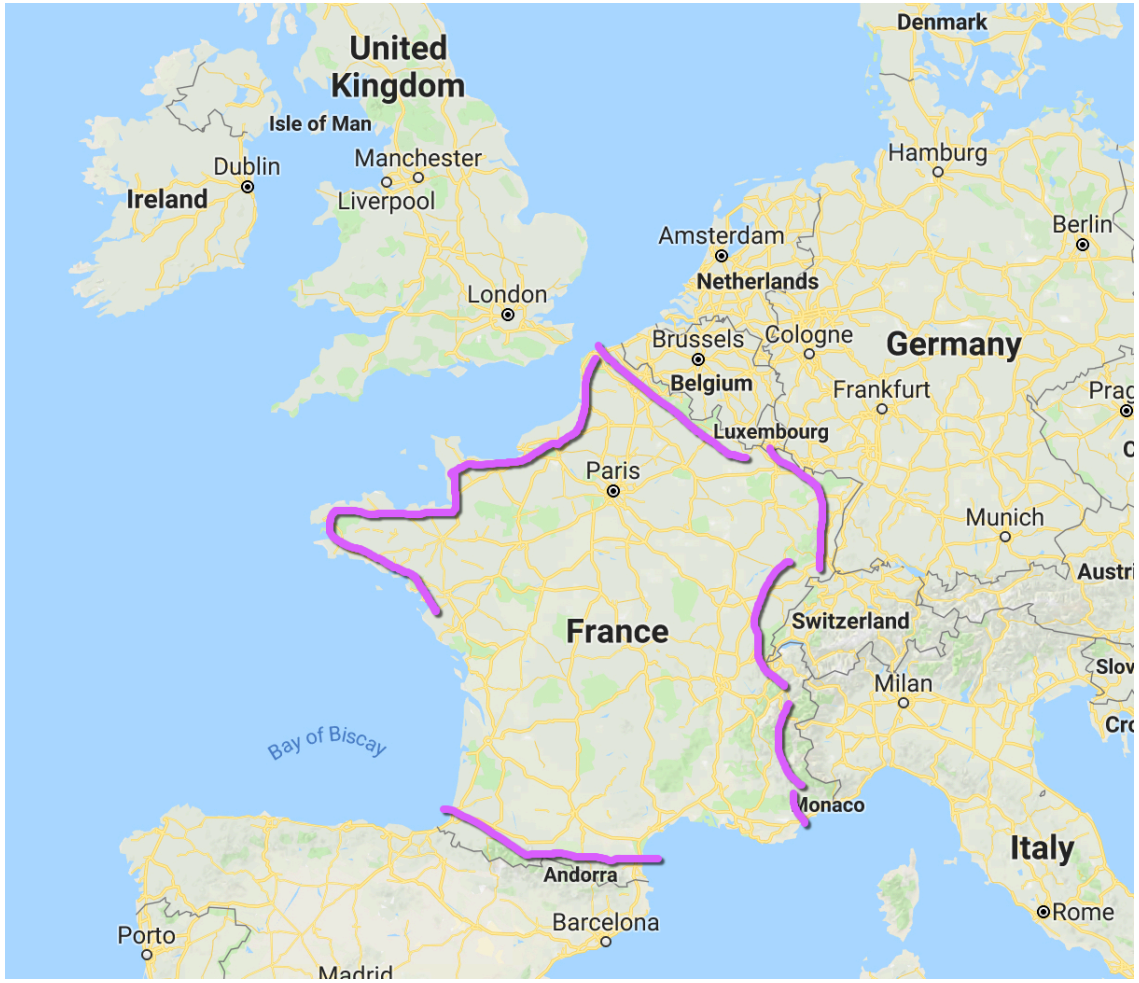
External and internal threats



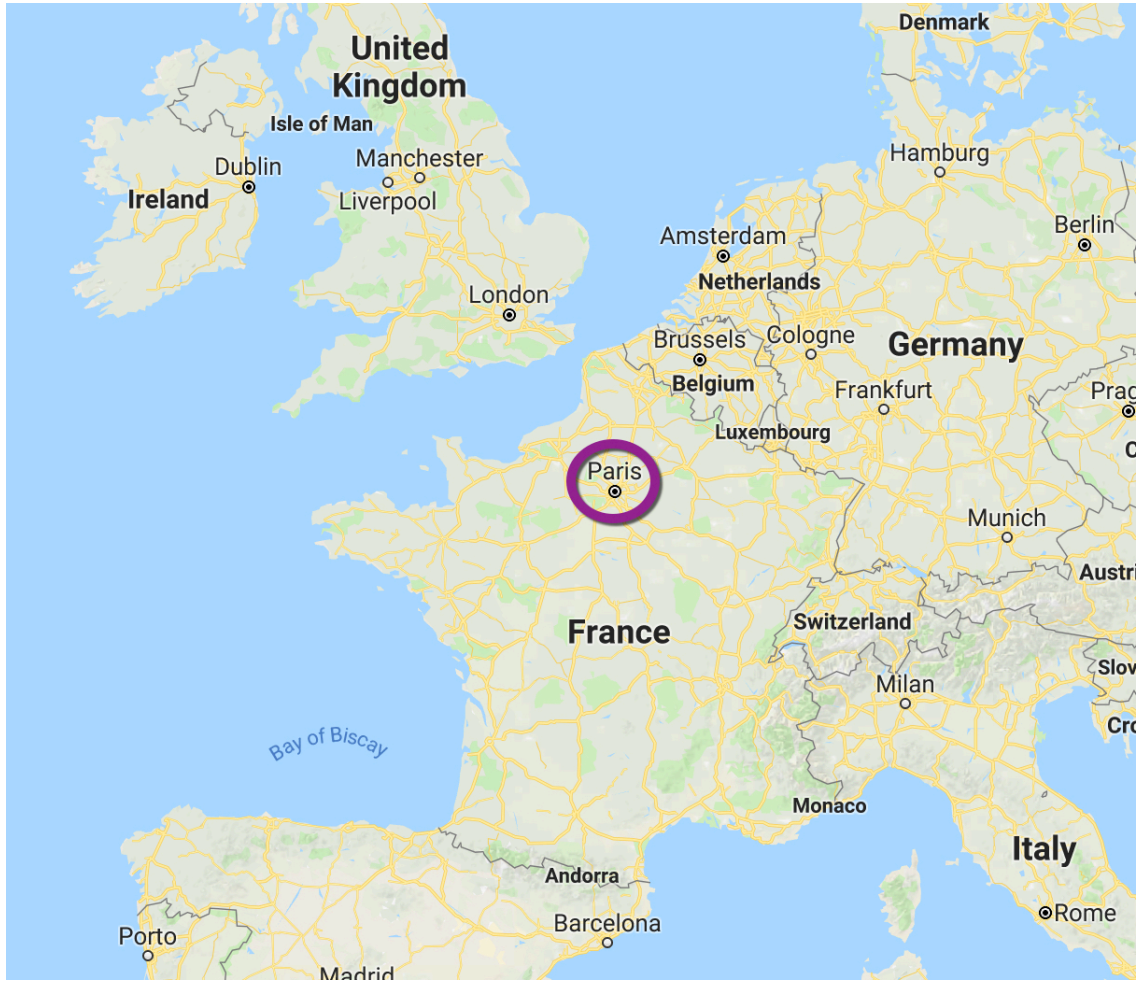
External and internal threats



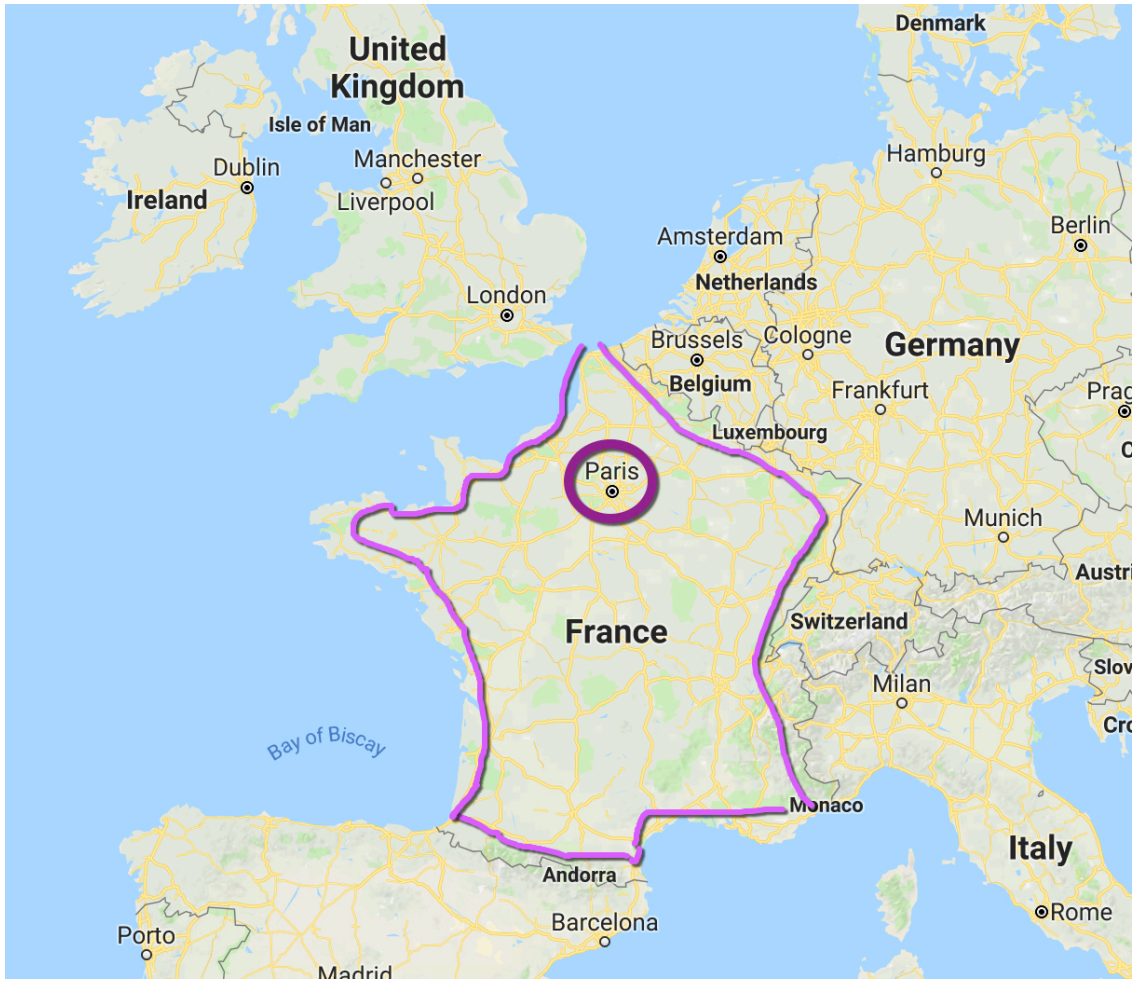
External and internal threats



External and internal threats

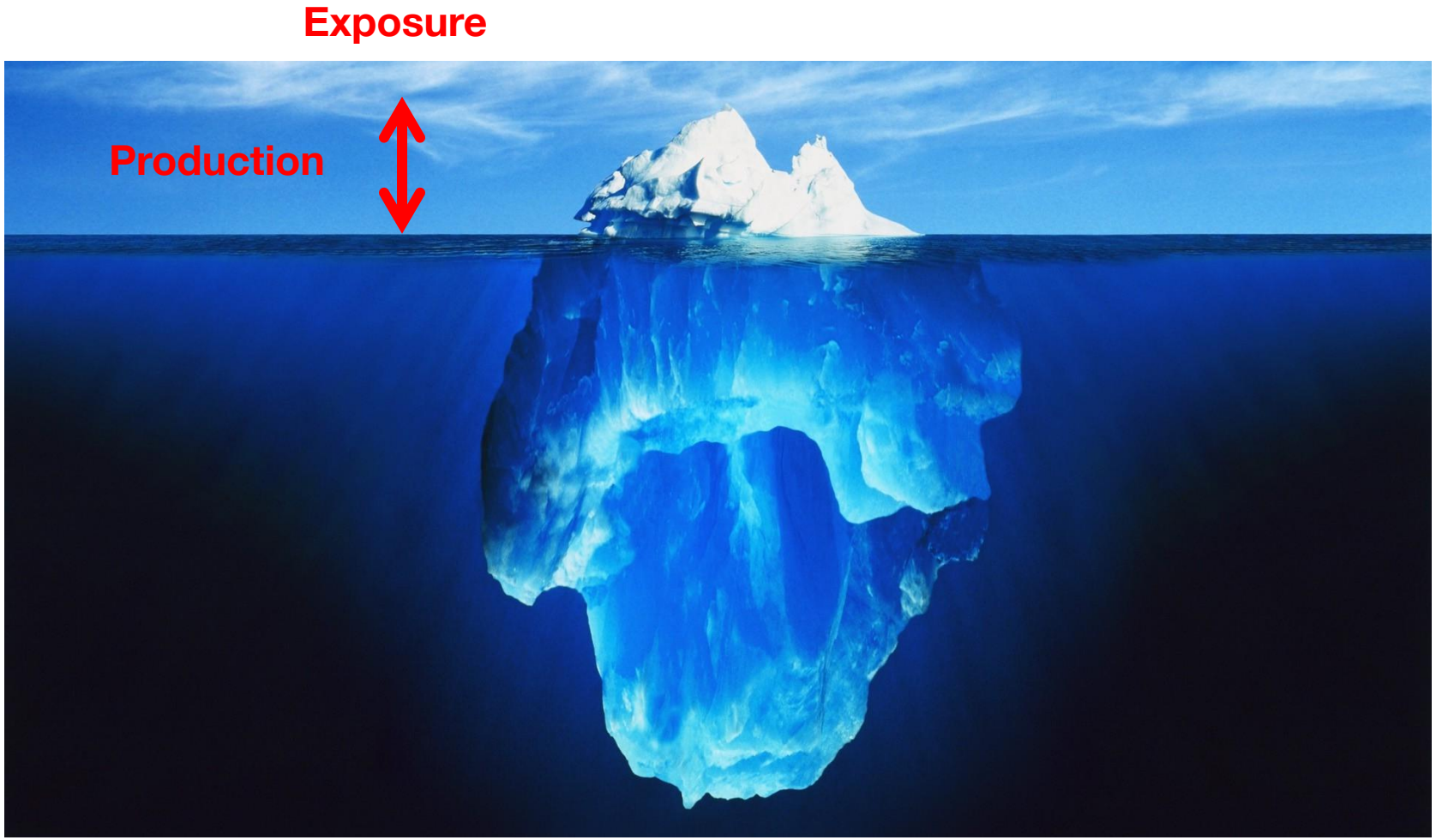


External and internal threats



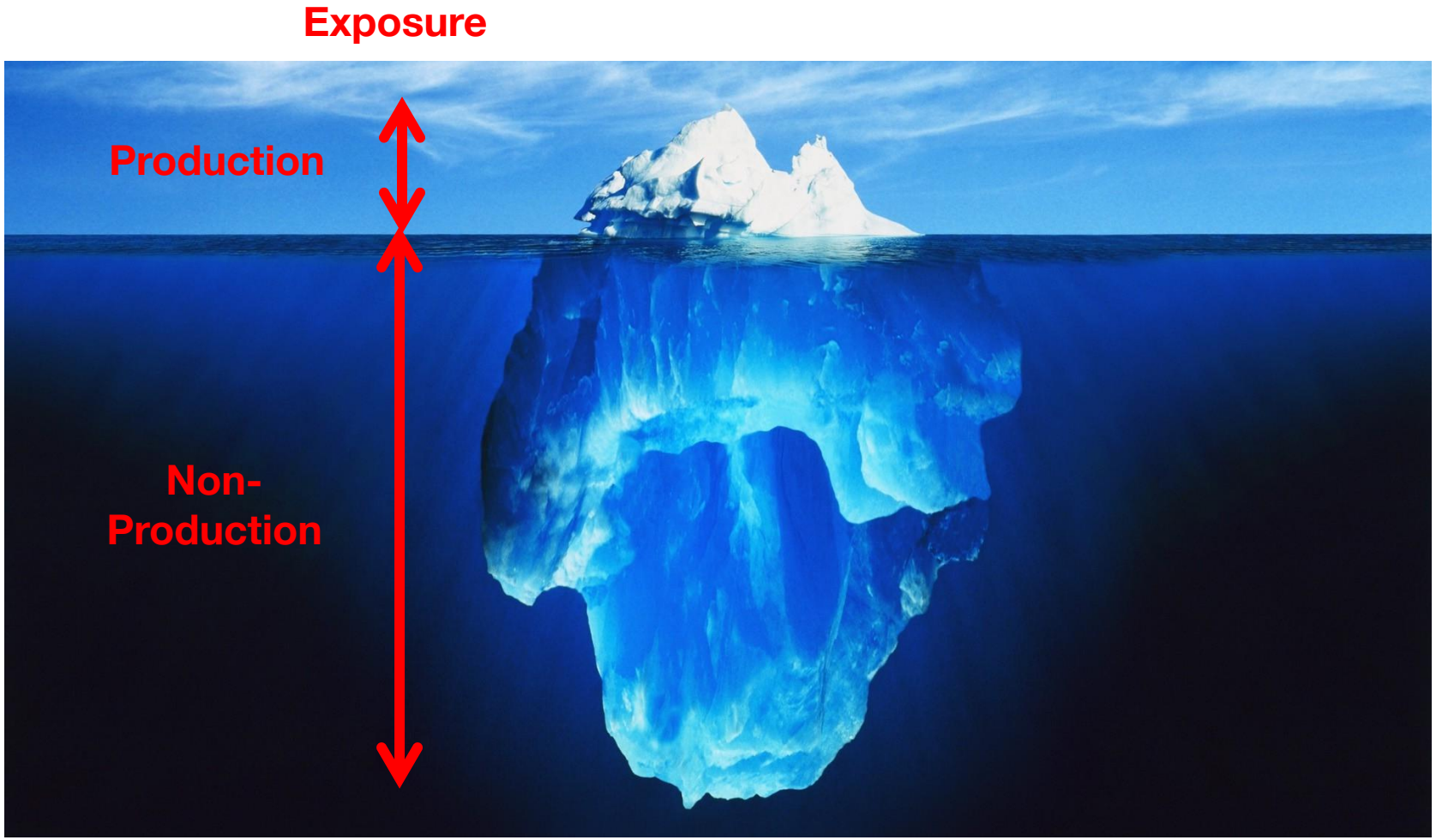
External and internal threats

- Non-production environments represent an enormous increase in the *surface area of risk* for exposure of sensitive production data



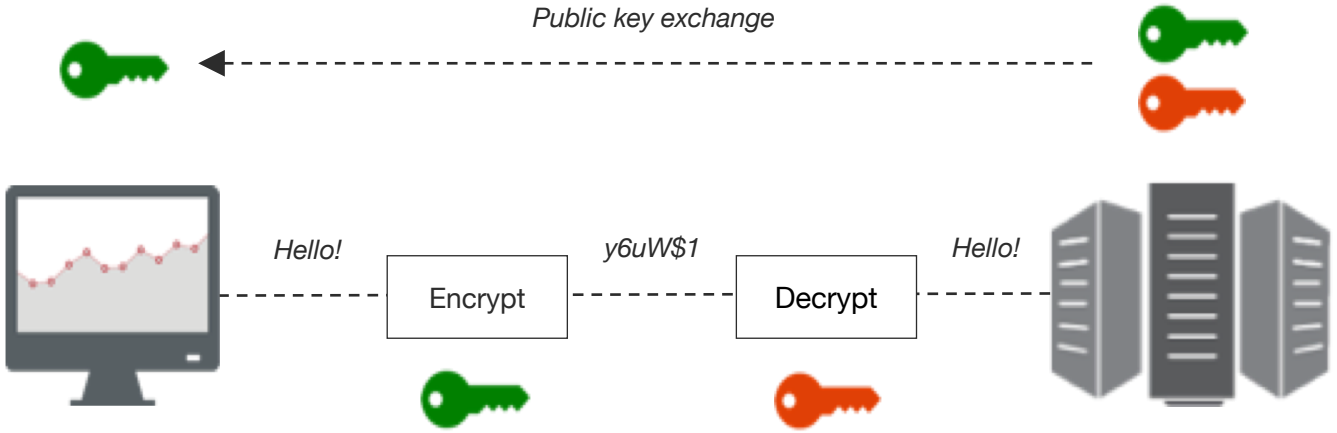
External and internal threats

- Non-production environments represent an enormous increase in the *surface area of risk* for exposure of sensitive production data



External and internal threats

- **Encryption** is the process of encoding data in such a way that only authenticated and authorized parties can decrypt it
- Decryption = ***reversible*** obfuscation



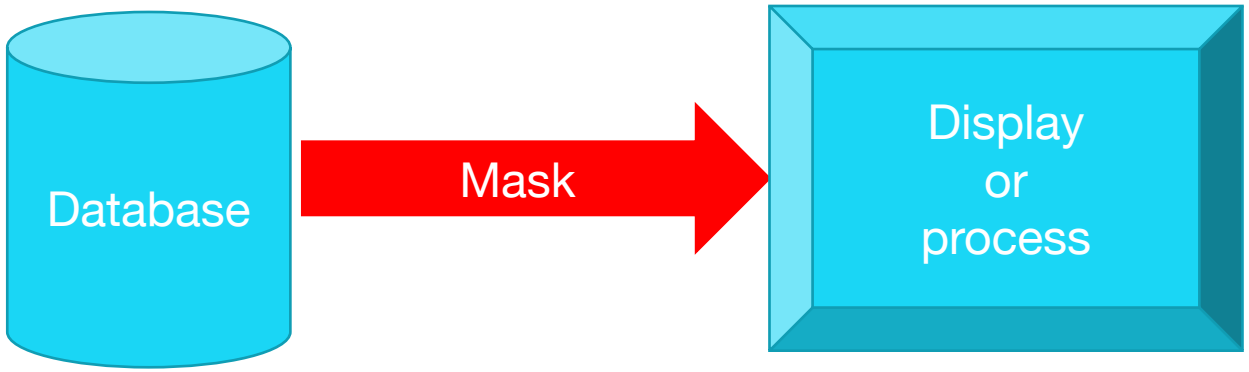
ADVANTAGES

- Effective for sending data such as emails or files between two secured locations (*data in-flight*)
- Effective for protecting data in a production application (*data at-rest*)

- In non-production, developers and testers must be authorized to decrypt data to do their jobs
- What if they aren't really authorized to view sensitive data?

External and internal threats

- **Masking data in-flight** is the obfuscation of data **after** it has been retrieved from storage **at-rest**
- Masking = ***non-reversible*** obfuscation



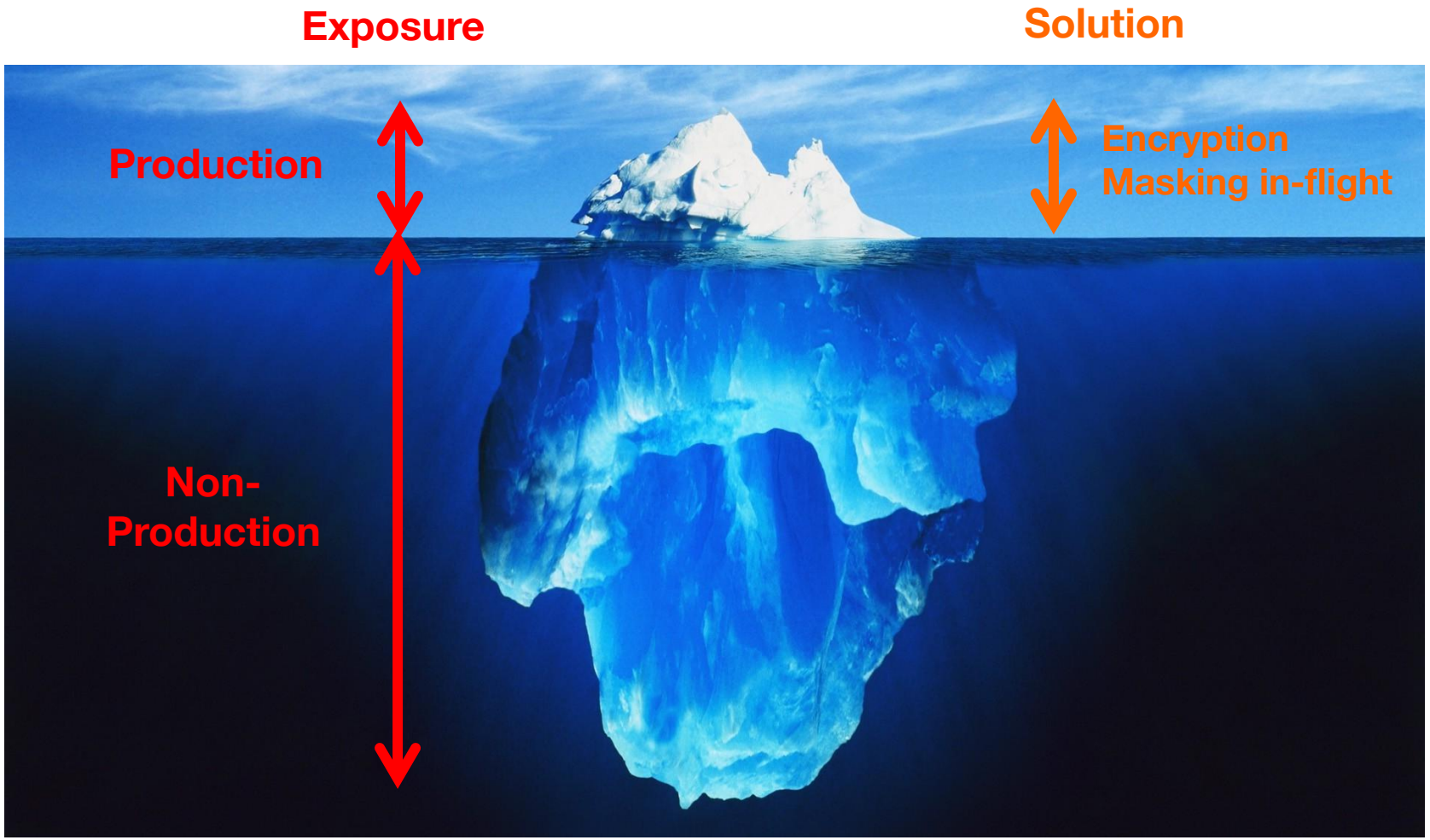
ADVANTAGES

- Effective for obfuscating data in production systems by not changing data at-rest

- SQL Server Dynamic Data Masking (DDM) is an example

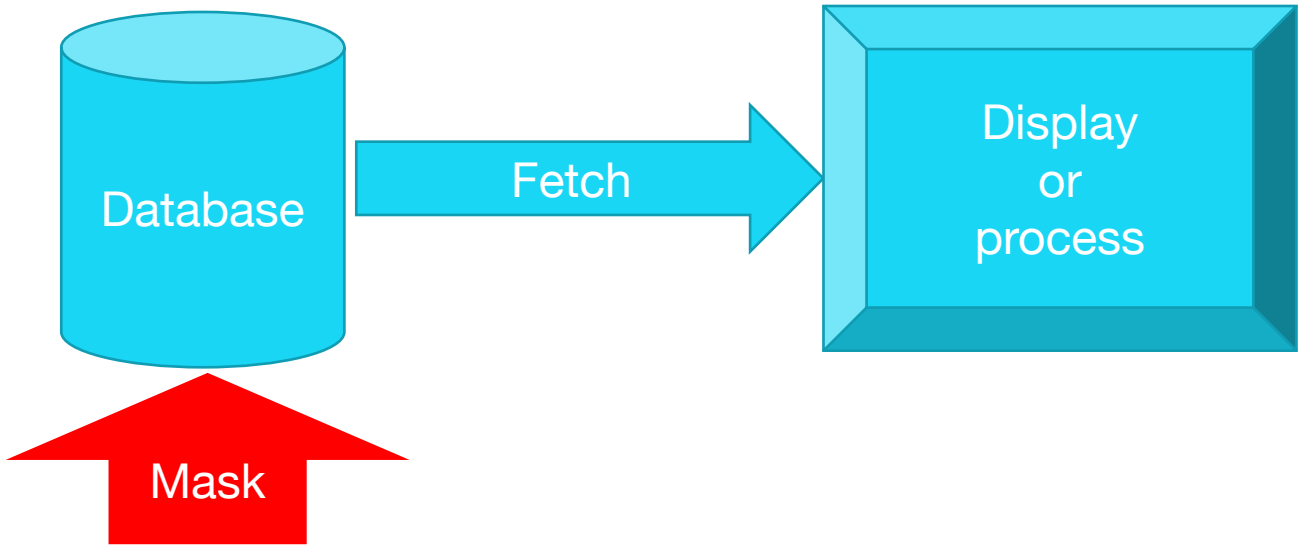
External and internal threats

- **Encryption** is the appropriate solution in **production** systems
  - *obfuscation* which is *reversible* upon *authorization*



External and internal threats

- **Masking data at-rest** is the obfuscation of data within the database using SQL statements
- Masking = ***non-reversible*** obfuscation



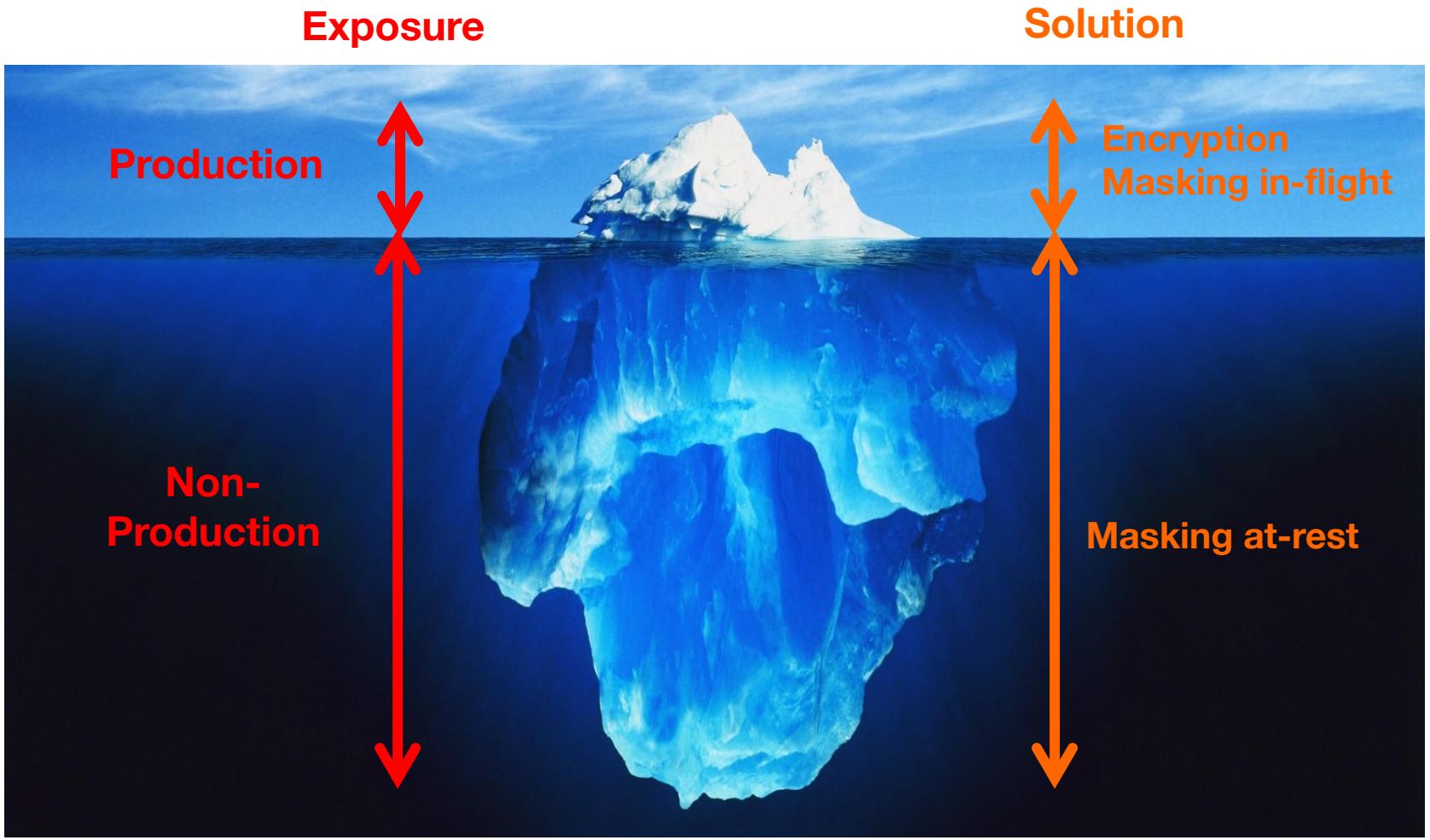
ADVANTAGES

- Effective for obfuscating data in non-production systems by changing data at-rest
- Allows provisioning non-production systems outside of secured authorized environments

- Delphix, IBM Optim, Informatica data masking are examples

External and internal threats

- **Encryption** and **masking in-flight** are appropriate solutions in **production** systems
  - *obfuscation* which is *reversible* upon *authorization*
- **Masking at-rest** is the appropriate solution in **non-production** systems
  - *obfuscation* which is *never reversible*

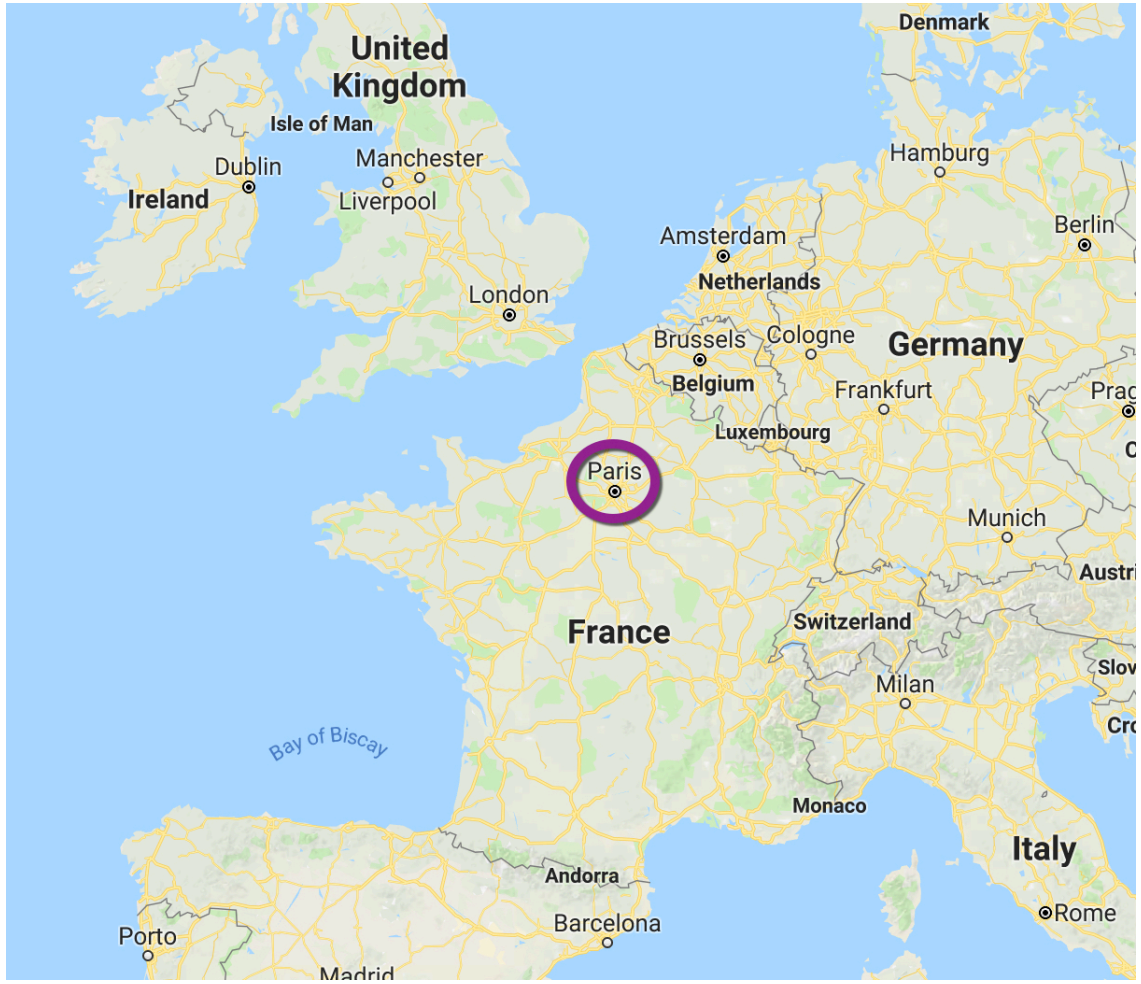


External and internal threats

- Database virtualization
  - For decades, non-production databases have been created using…
    - Database copies from production
    - Newly-created databases with generated data
  - Data virtualization technologies are now available
    - Thin-clone copies of databases sourced from production presented via network-attached storage
    - Allows DBAs to create TB-sized database copies in less than 10 minutes
      - Delphix, Windocks, Red Gate, Rubrik, Actifio, etc
  - So, by cloning production to create dozens or hundreds of copies for non-production…

…somewhere a security administrator is writhing in agony

External and internal threats



Agenda

1. Fear and loathing
2. External and internal threats
3. **Data masking**
4. Summary

Data masking

- Data masking at-rest is the *permanent irreversible* obfuscation of data
  - Obfuscation **does not** always mean *scrambling* or *randomizing*
    - Scrambled data looks *awful*, so awful that it is distracting
  - Obfuscation **does** involve the use of sophisticated algorithms to irreversibly obfuscate data
    - So that the data looks *useful*, but has no relation to the original values
- Many sensitive data items have embedded encoding
  - Simply scrambling these fields will cause applications to break
- Many sensitive data items are related groupings
  - Algorithms must be available to mask groups of fields
- Many data items can be inferred from other data items
  - Suppose there are only nine of 5000 patients standing taller than 2 meters in the original data
    - Having one or more known items of data can permit the inference of identity

Data masking

1	• Masking must not be reversible
2	• The results must be representative of the data source
3	• Referential integrity must be maintained
4	• Only mask non-sensitive data if it can be used to recreate sensitive data
5	• Masking must be a repeatable process

According to Rich Mogull, Securosis

Data masking

HOME GROWN SCRIPTS

- Build (vs Buy) has hidden costs
- Test data quality is low

KEY POINTS

- How much in-house expertise do we really have for obfuscating data?
- Are the scripts reusable? Across database platform? Across platform? Across all documents?
- How well has it been tested?
- Are we really serious about protecting confidential data? Or just checking off an item from a task list?

Data masking

HOME GROWN SCRIPTS	• Build (vs Buy) has hidden costs • Test data quality is low
STORED PROCEDURE	• Hard to mask data consistently due to specific code per source

KEY POINTS

- More formal than scripts…
- *Stored procedures?* How many database platforms are covered?
- *Stored procedures?* Does this work on documents too?
- How well does this scale? Has it been tested for performance?
- How is referential integrity managed?

Data masking

HOME GROWN SCRIPTS	• Build (vs Buy) has hidden costs • Test data quality is low
STORED PROCEDURE	• Hard to mask data consistently due to specific code per source
ETL REPURPOSING	• Point solution focused on data manipulation

KEY POINTS

- Obfuscating data is about *extracting* unmasked data, *transforming* data, and then *loading* masked data back, so this seems a natural evolution, but…
- Is there a “discovery” function embedded to ensure that all confidential data is identified?
- Is there an “audit” function to track when data was audited and if new data has been added since?

Data masking

HOME GROWN SCRIPTS	• Build (vs Buy) has hidden costs • Test data quality is low
STORED PROCEDURE	• Hard to mask data consistently due to specific code per source
ETL REPURPOSING	• Point solution focused on data manipulation
ENTERPRISE DATA MASKING	• Comprehensive • Profile, secure, and audit

KEY POINTS

- Dozens of data domains, and obfuscation algorithms for each domain pre-defined
- Custom domains, algorithms can be added
- Not a repurposed tool, nor designed to create a large services contract
- Consistently mask data on-premise; across data centers; the cloud; for files, RDBMS, or Mainframe
- Horizontally scales
- Integrated with the provisioning step of data virtualization
- Comprehensive solution

Data masking

PROFILE

- IDENTIFY sensitive data across sources
- ASSIGN masking algorithms to match data using domains
- REPORT risk profile across the enterprise

Data masking

PROFILE

- IDENTIFY sensitive data across sources
- ASSIGN masking algorithms to match data using domains
- REPORT risk profile across the enterprise

SECURE

- MASK without any programming
- MAINTAIN usability with fictitious, but realistic data
- APPLY masking with consistency, repeatability

Data masking

PROFILE

- » IDENTIFY sensitive data across sources
- » ASSIGN masking algorithms to match data using domains
- » REPORT risk profile across the enterprise

SECURE

- » MASK without any programming
- » MAINTAIN usability with fictitious, but realistic data
- » APPLY masking with consistency, repeatability

AUDIT

- » VERIFY all sensitive data is masked
- » admins if vulnerabilities are Ided
- » assessment to auditors

Data masking

- Profiling
  - Most application administrators know what tables, columns, and documents contain about **80%** of sensitive and confidential data
    - DBAs and programmer/analysts can probably identify another 10-15% of additional tables, columns, and documents
    - But it is the remaining 5% that goes overlooked that presents a problem
  - Profiling is functionality which scans data dictionaries and data seeking probable sources of sensitive and confidential data
    - Using regular expressions and text patterns
  - The end result of a profiling project is an inventory of sensitive data
    - Manual review and selection is needed to refine the inventory

Data masking

- Secure Lookup Algorithm
  - One of eight (8) data transformation frameworks pre-built into the Delphix masking engine
    - Patented proprietary encrypt / hash / modulus lookup algorithm, repeatable yet unbreakable
  - Used to assign a realistic value from a value selected from a pre-defined lookup table
    - The algorithm is irreversible and purposely creates collisions in the output values for added security
- Example
  1. Starting with original column value of "`XYZ Holdings`"
  2. original table has about 1000 distinct data values in the column
     - lookup table can be defined with 500 distinct data values
  3. Encrypt original value using AES 256 to "*`lGql159bm7aX2C3bBVMJ3uIg%=`*"
  4. MD5 Hash of the encrypted result = "`428618117`"
  5. 428618117 mod 500 = `117`
  6. Value within lookup table at entry 117 is "`Standard Oil`"

Data masking

The other seven prebuilt algorithms…

2. **Binary Lookup** – Much like **Secure Lookup**, but used when entire files are stored in a specific column
3. **Mapping** – Sequentially maps original data values to masked values that are pre-populated to a lookup table in the masking utility
4. **Segmented Mapping** – Replaces data values based on segment definitions. For example, an ACCOUNT NUMBER algorithm might keep the first segment of an account number but replace the remainder or remaining segments with a random number
5. **Min/Max** - This algorithm allows you to make sure all the values in the database are within a specified range. They prevent unique identification of individuals by characteristics that are outside the normal range, such as age over 99
6. **Data Cleansing** – If the target data needs to be put in a standard format prior to masking, you can use this algorithm. For example, Ariz, Az, Arizona can all be cleansed to AZ
7. **Free-Text Redaction** – This algorithms masks or redacts free text columns of files. It uses either a Whitelist or Blacklist to determine what words are masked or not masked. This algorithm may require additional configuration to work in the manner you desire
8. **Tokenization** – Replaces the data value with an algorithmically generated token that can be reversed

Data masking

Auditing

- As with anything in life, masking is an iterative process
- Despite best efforts during the profiling phase, things may be missed
- New functionality and new fixes might expose sensitive data inadvertently
- Human error (missed masking jobs, etc)
- Auditing processes are able to differentiate already-masked data from unmasked data
- From a liability standpoint, the organization needs a way to verify

Data masking

Gotchas

- Masking at-rest is performed using SQL transactions
  - Just like any other SQL-based application
- SQL Server, like other databases, has built-in data recovery mechanisms for SQL transactions
  - System-versioned temporal tables
    - Consider setting SYSTEM_VERSIONING to OFF during masking operations
  - Change data capture or change tracking
    - Consider running SYS.SP_CDC_DISABLE_DB and/or DISABLE CHANGE TRACKING during masking operations
  - Trigger logic
    - Custom audit trails
  - Recovery model
    - Consider switching to simple recovery model during masking operations
  - Other gotchas?

Agenda

1. Fear and loathing
2. External and internal threats
3. Data masking
4. **Summary**

Summary

1. Understand the different choices and their use-cases…
    1. **Encryption** and **masking in-flight** are good obfuscation solutions for **production** environments
        - Where all users are authenticated and authorized by the application
        - Where sensitive data can only be temporarily obfuscated
        - SQL Server Dynamic Data Masking is masking in-flight
    2. **Data masking at-rest** is the right solution for non-production environments
        - Irreversibly make sensitive data *inconsequential* from a security perspective
        - Remove the value from the asset
2. Data masking at-rest products…
    - Delphix DMSuite, IBM Optim, Informatica Data Masking, Red Gate Data Masker
    - Don't forget about database features that can trip you up
        - System-versioned temporal tables, use simple recovery mode during masking, etc

Summary

- Job titles/descriptions that didn't exist in 2016…
  - Data masking specialist
  - Data protection and vulnerability management specialist

…but they do now…

Summary

- If there is time and connectivity…

Virtualization and Masking demo *(about 10 minutes)*

Q & A


Tim.Gorman@Delphix.com


@TimGormanTech


https://www.linkedin.com/in/TimGorman/

