Attack Surface of AI Applications

OWASP Stammtisch München

Clemens Hübner
inovex GmbH

04. Februar 2025

Euer Hintergrund?

Klassische Softwareentwicklung / Betrieb / Security

Data Science / Machine Learning / Datensicherheit

Clemens Hübner

Software Security @ inovex, Munich

Enabling teams to design, implement and test secure software

@clemens@infosec.exchange

/clemens-huebner

clemens.huebner@inovex.de

@inovexlife

blog.inovex.de

Using AI in software security

Builders

- Support modeling and documentation
- Generate secure code
- Answer questions about security
- Write/perform security tests

Defenders

- Classify vulnerabilities
- Improve incident detection / response
- Enhance monitoring (e.g. anomaly detection)

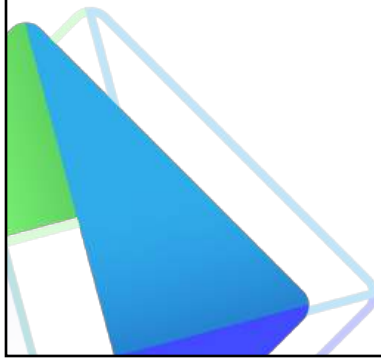
Breakers

- Generate payloads
- Develop exploits
- Improve human-directed attacks (e.g. phishing)

Questions for today

- What is the attack surface of an AI software system?
- What risks and weaknesses should be considered in AI software?
- Which measures and best practices exist for secure development of AI software?

What is the attack surface of an AI software system?



Definition AI software system

An "AI software system" is a computer program or application that utilizes artificial intelligence techniques and algorithms to perform tasks, make decisions, or analyze data to deliver intelligent functionality within software applications. *(BSI)*

Definition AI software system

An "AI software system" is a computer program or application that utilizes artificial intelligence techniques and algorithms to perform tasks, make decisions, or analyze data to deliver intelligent functionality within software applications. *(BSI)*

I can do programming!
I can do applications!

Definition AI software system

An "AI software system" is a computer program or application that utilizes artificial intelligence techniques and algorithms to perform tasks, make decisions, or analyze data to deliver intelligent functionality within software applications. *(BSI)*

I can do AI!
I can do data!

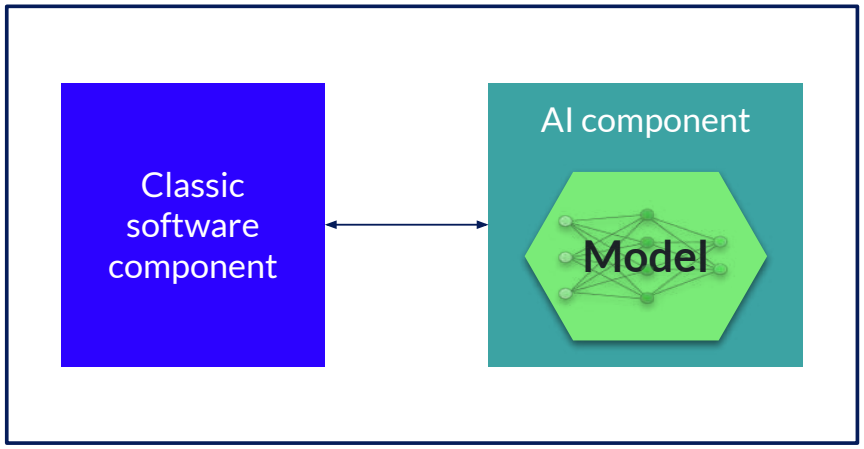
Definition AI software system

An "AI software system" is a computer program or application that utilizes artificial intelligence techniques and algorithms to perform tasks, make decisions, or analyze data to deliver intelligent functionality within software applications. *(BSI)*

An AI software system
is a software system
containing an AI component.

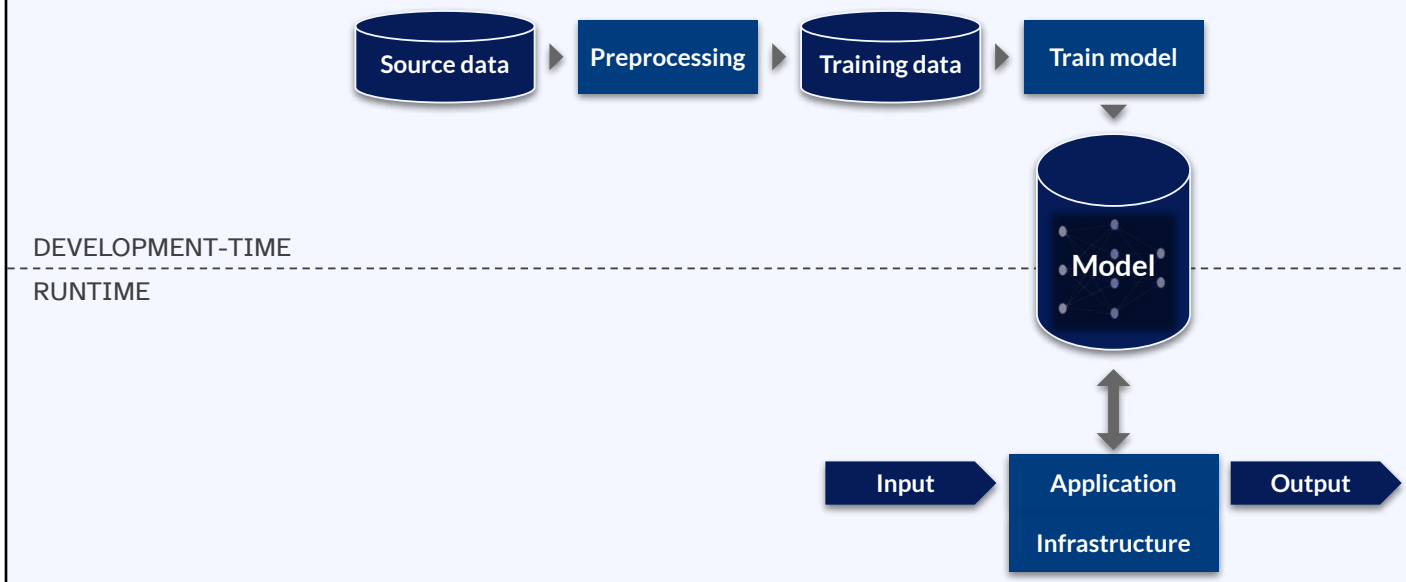
Definition AI software system

An AI software system is a software system containing an AI component.



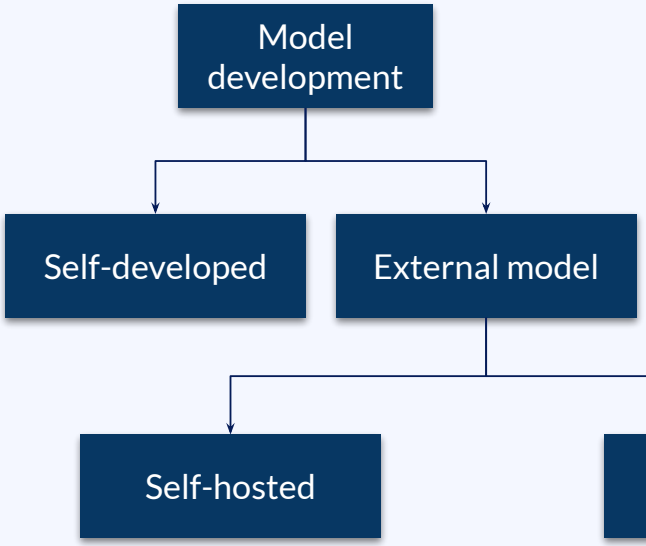
AI software system

AI Development Process



inovex

Model development



- Developing an own model needs expertise
- Well-thought design

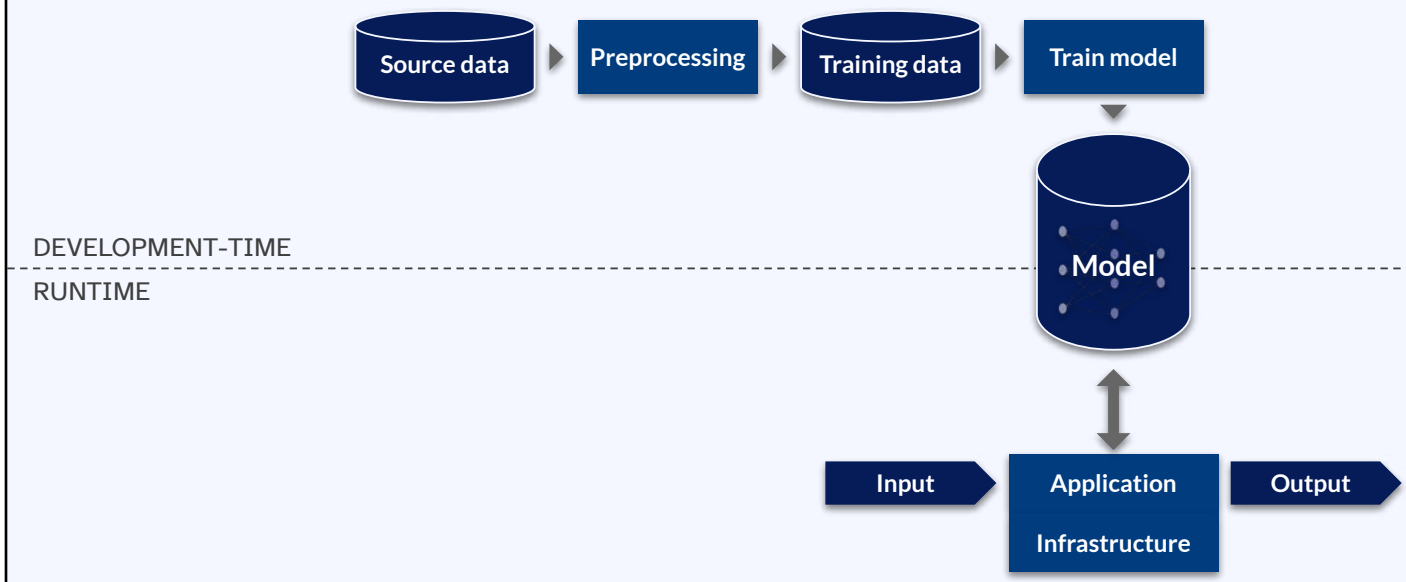
- Usage of model developed by other providers

- Model hosted in own environment
- Use as-is or extend the model with domain data

- Model API called from another provider
- Added dependency on external system environment

inovex

AI Development Process



inovex

Morris II AI worm can steal your confidential data and infect ChatGPT and Gemini

Google Brain researchers demo method to hijack neural networks

National Cyber Security Centre

Exercise caution when building off LLMs

30 August 2023

ChatGPT Continues to Fail in Fight Against Malicious Content

by Vishwa Pandagle — February 8, 2023 - Updated on May 4, 2023

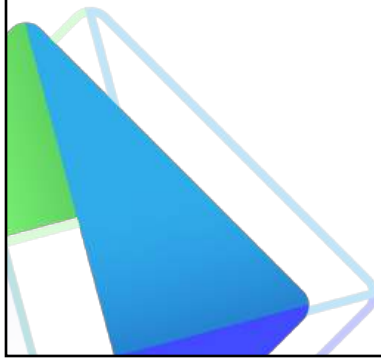
ADVENTURES IN 21ST-CENTURY HACKING —

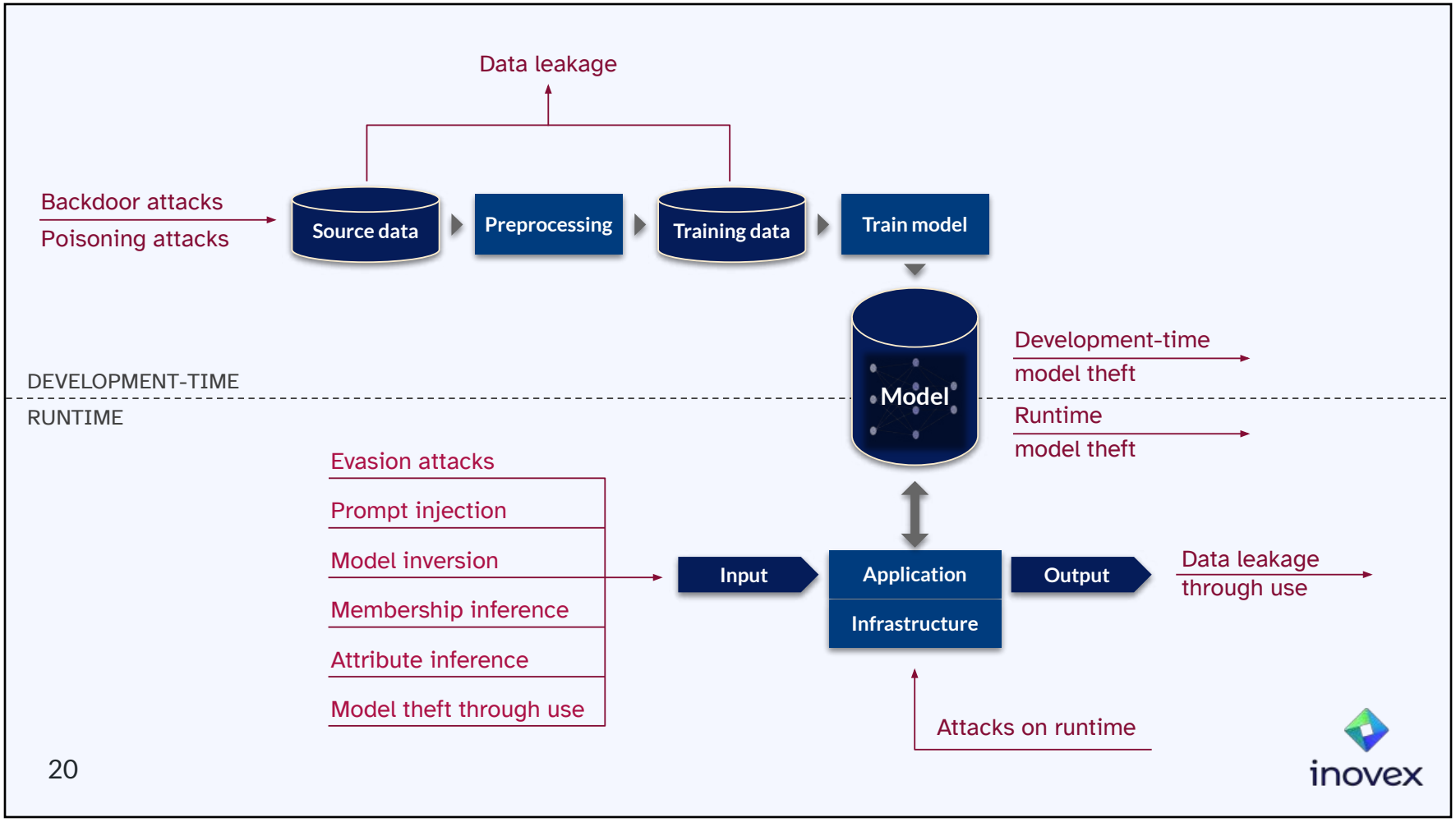
AI-powered Bing Chat spills its secrets via prompt injection attack

BENJ EDWARDS - 2/10/2023, 8:11 PM

inovex

What risks and weaknesses should be considered in AI software?




Data leakage
Backdoor attacks
Poisoning attacks
Source data
Preprocessing
Training data
Train model
Model
Development-time
model theft
Runtime
model theft
DEVELOPMENT-TIME
RUNTIME
Evasion attacks
Prompt injection
Model inversion
Membership inference
Attribute inference
Model theft through use
Input
Application
Infrastructure
Output
Data leakage
through use
Attacks on runtime

inovex

Demo software: Credit Score Service

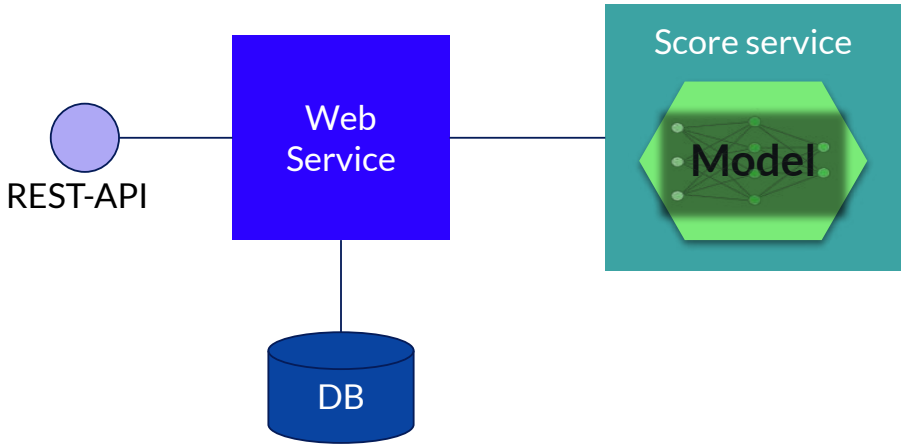
Use Case: Calculate creditworthiness of applicants

Input:

- Demographics
- Payment History
- ...

Output:

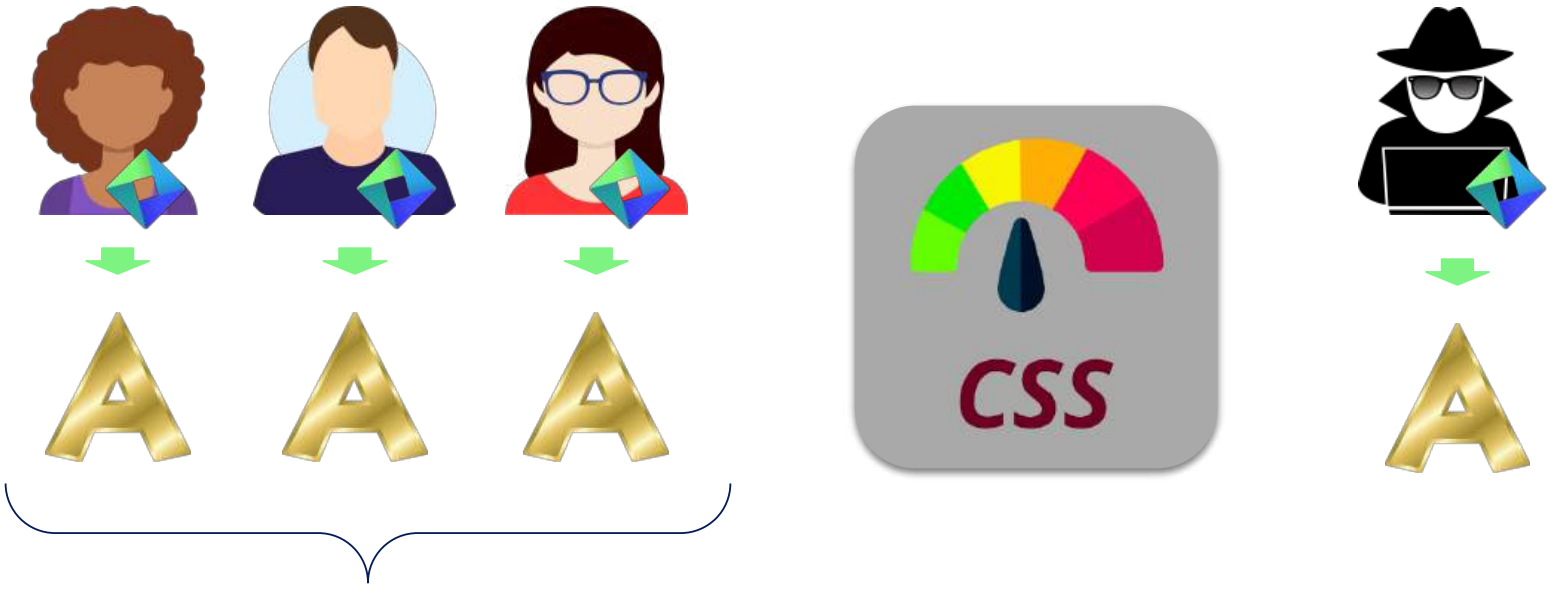
- Credit Score



Backdoor Attacks

Training data is manipulated in a way an attacker can obtain wrong results later.

Backdoor Attacks



Prevent Backdoor Attacks

Never trust user input!

- question training data, handle untrusted data carefully
- validate/sanitize input

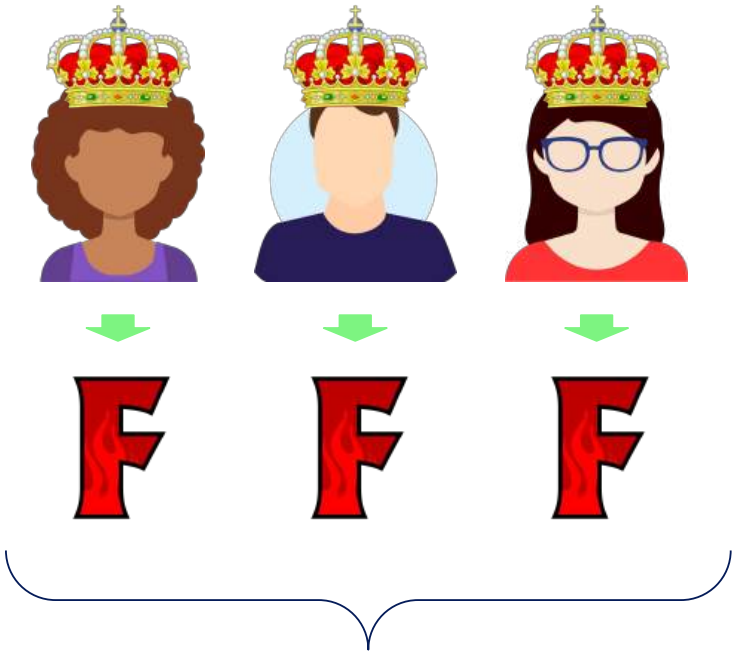
Never trust data quality!

- perform quality control on train data
- train decentral, maybe even federated
- distort train data
- prevent overfitting

Poisoning Attacks

Training data is manipulated so the attacker reduces the results of the model, e.g. its efficiency or correctness.

Poisoning Attacks



Train set

Prevent Poisoning Attacks

Never trust user input!

- question training data, handle untrusted data carefully
- validate/sanitize input
- handle data as part of supply chain

Never trust data quality!

- perform quality control on train data
- broaden train data, use federated learning
- use *golden dataset* for stability checks

Evasion Attacks

The attacker manipulates the input to the model to influence its results

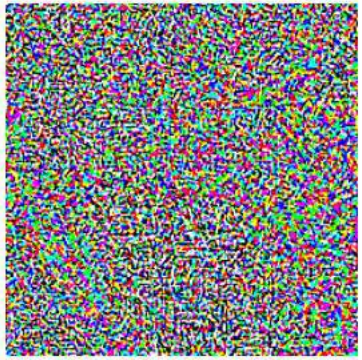
Evasion Attacks

Based on the attackers possibilities, we differentiate between

- Whitebox attacks, where the attacker has access to the model itself
- Blackblox attacks, where the attacker has no access to the model

White Box Adversarial Attacks

$+ .007 \times$



$=$

$\boldsymbol{x}$

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES
(Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy, Google Inc.)

Black Box Adversarial Attacks

Robust Physical-World Attacks on
Deep Learning Visual Classification
(Kevin Eykholt et al., 2018)

Evasion Attacks

- Small changes to applicants data might cause bigger changes in model output
- The more control the attacker has over the input, the easier attacks are

Prevent Evasion Attacks

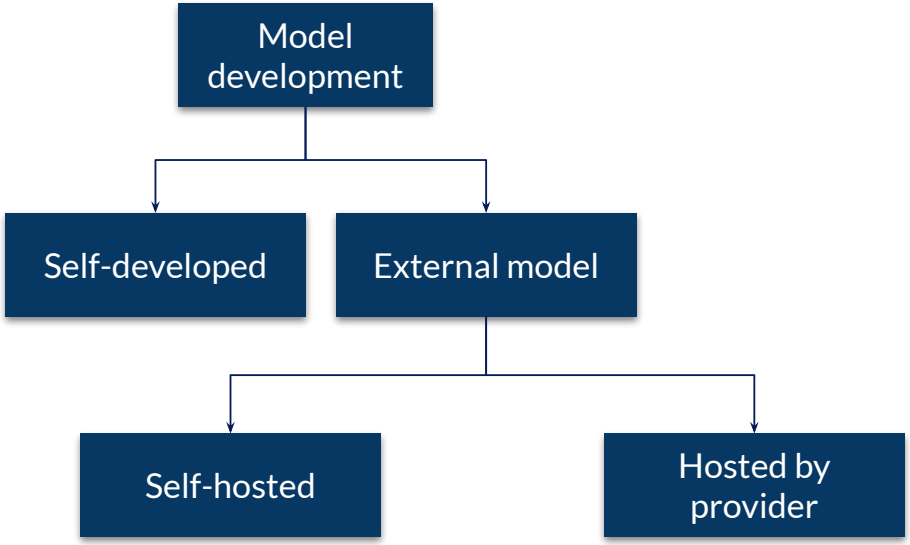
Expect users to be attackers!

- monitor usage, especially inputs
- restrict access
- sanitize inputs and outputs

Aim for a robust model!

- train adversarial examples
- distort input
- adversarial-aware distillation

Model development

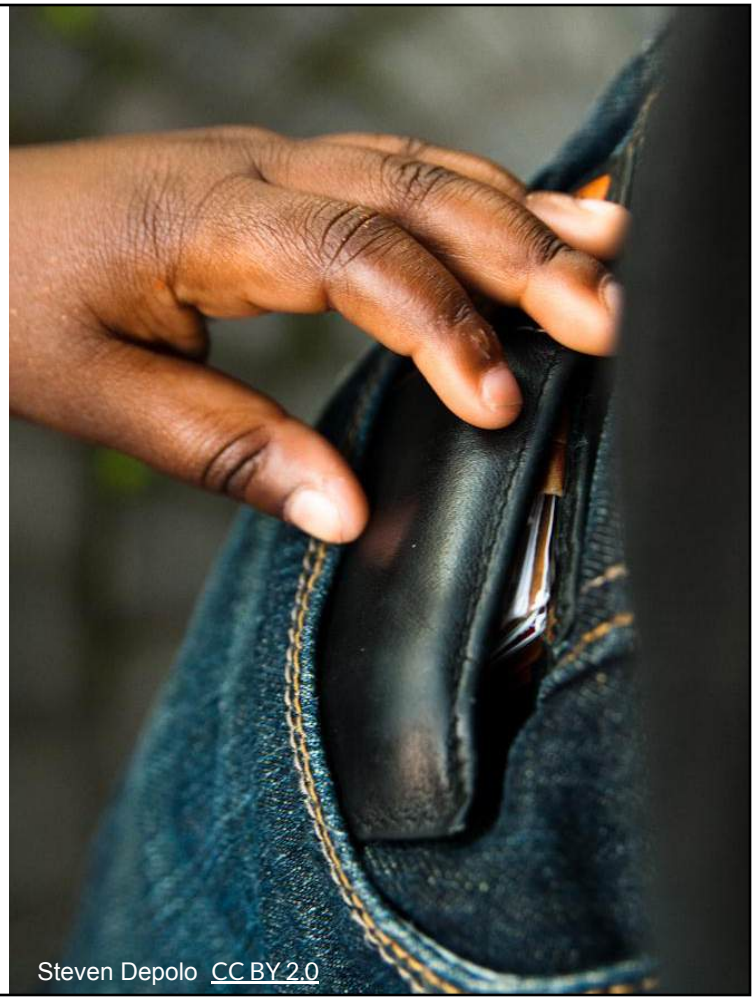


inovex

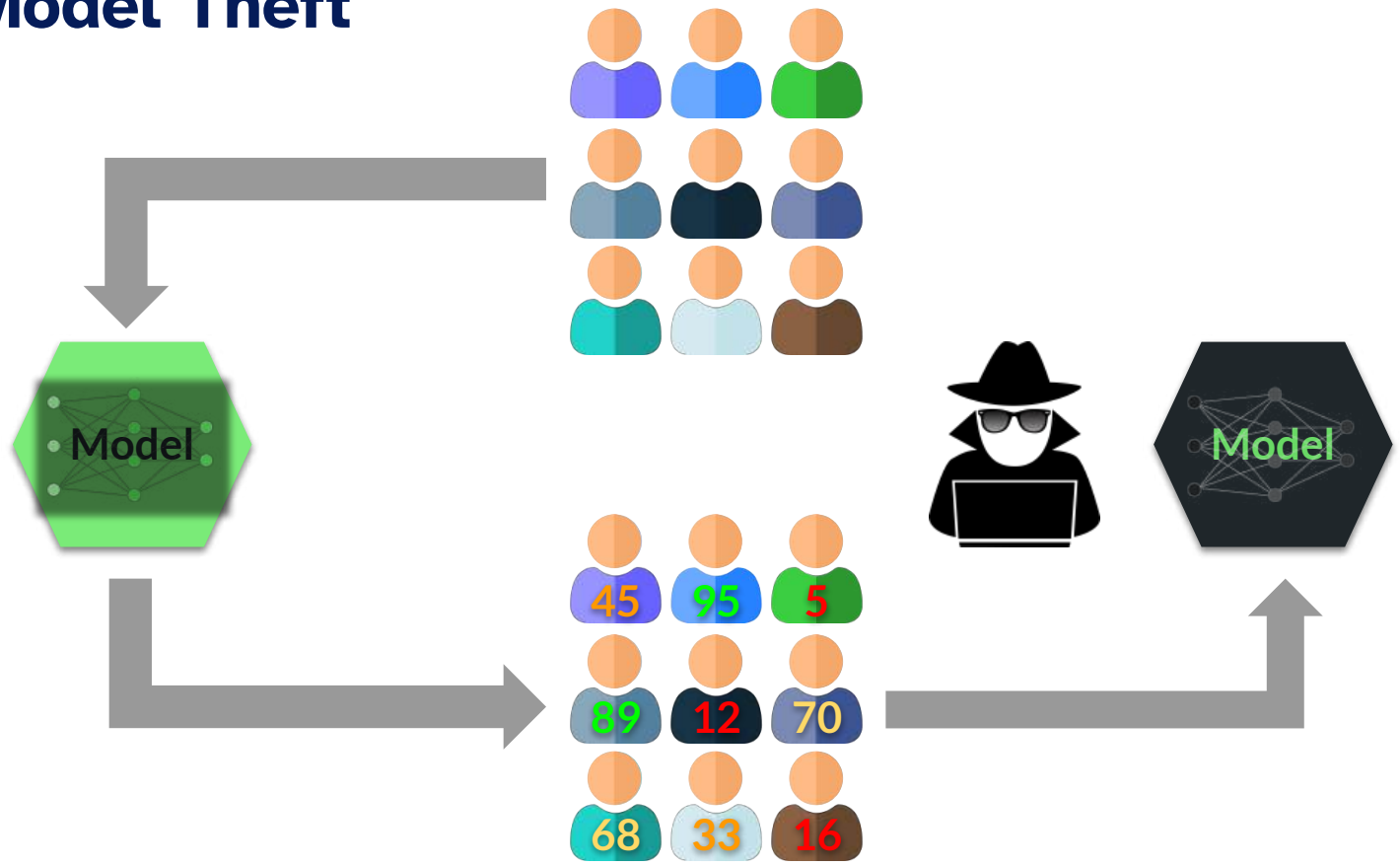
Model Theft

The attacker uses his access to the trained model as an oracle:

Classifying his own data set using the oracle allows him to train his own model



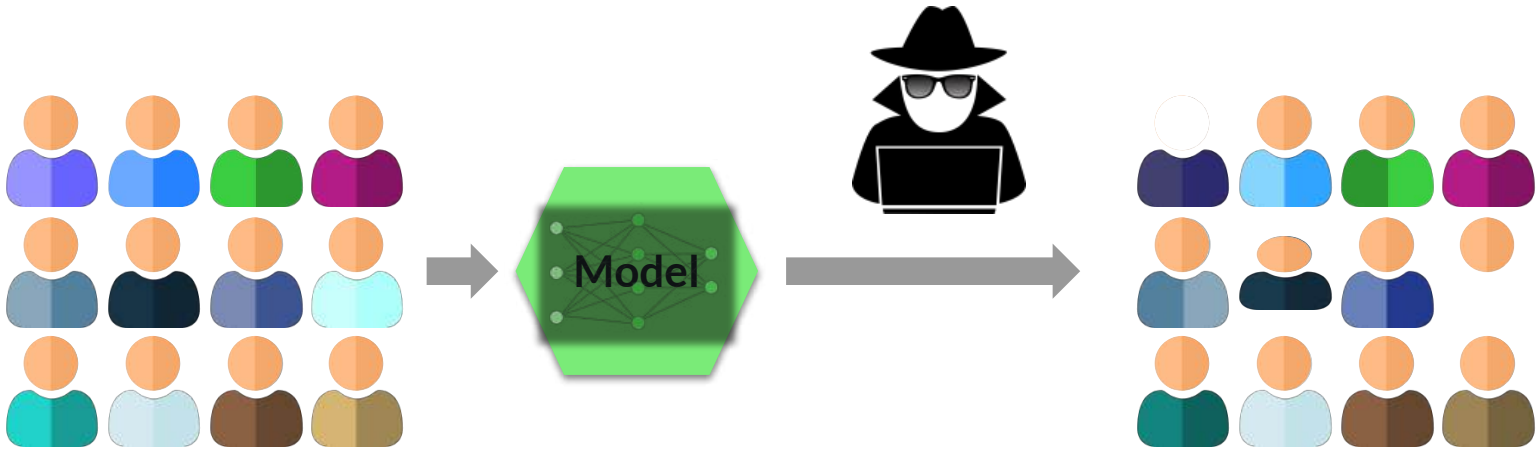
Model Theft



Model Inversion

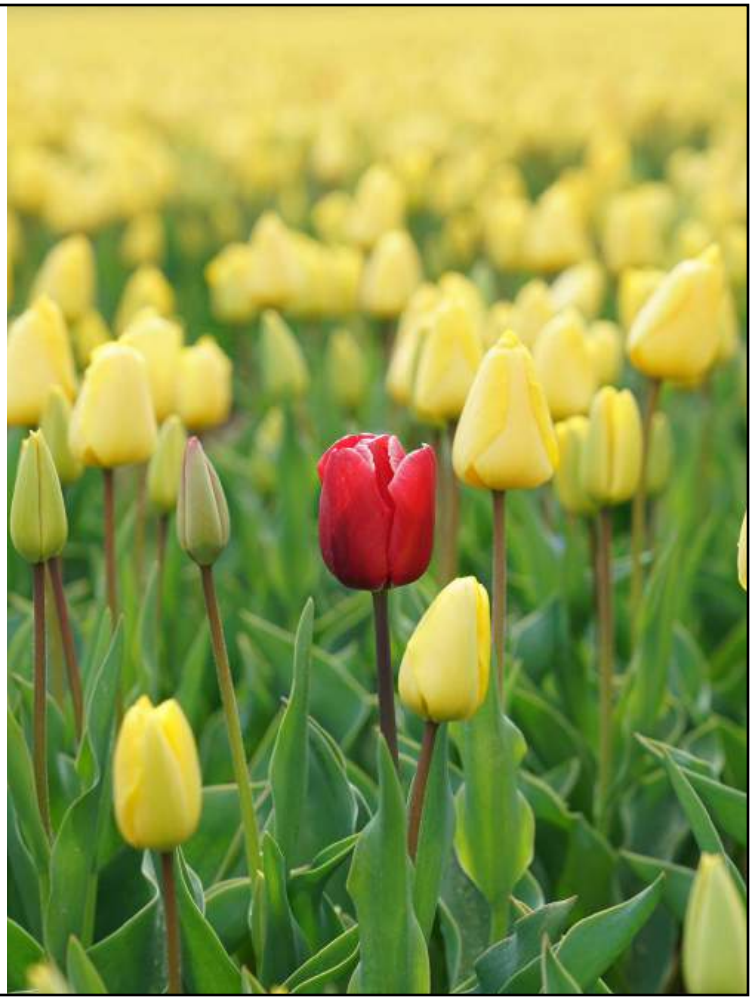
The attacker uses his access to the model to get information about the source data.

Model Inversion

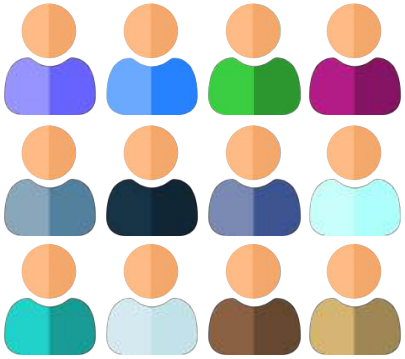


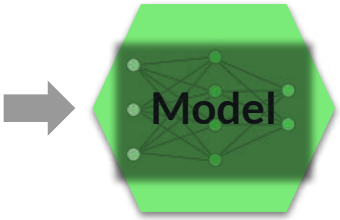
Membership Inference

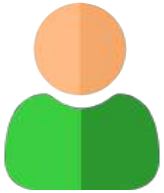
The attacker can obtain the information if a single piece of data was part of the training set.



Membership Inference







?

Prevent Model Theft / Inversion

Limit access to the system!

- restrict and monitor access
- rate limiting

Control creation and content of model

- prevent overfitting
- reduce model output

Attribute Inference

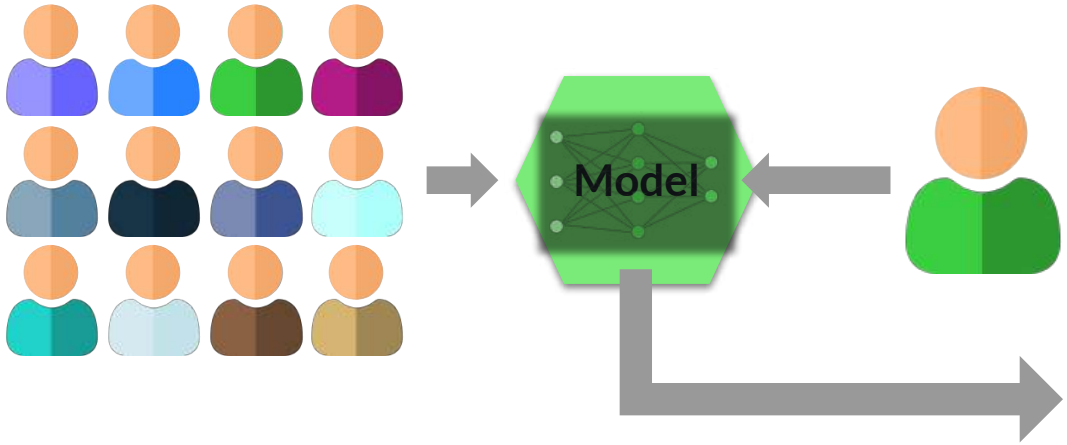
The attacker has a set of attributes related to a piece of input data

By attribute inference, he can get information about further, private attributes

Attribute Inference

- The attacker has a set of attributes related to a piece of input data
- By attribute inference, he can get information about further, private attributes

Attribute Inference



last name: Hübner
first name: Clemens
age: 30
profession: Security Engineer
location: Munich
married: ***???***

married: no

Prevent Attribute Inference

Secure access to user data

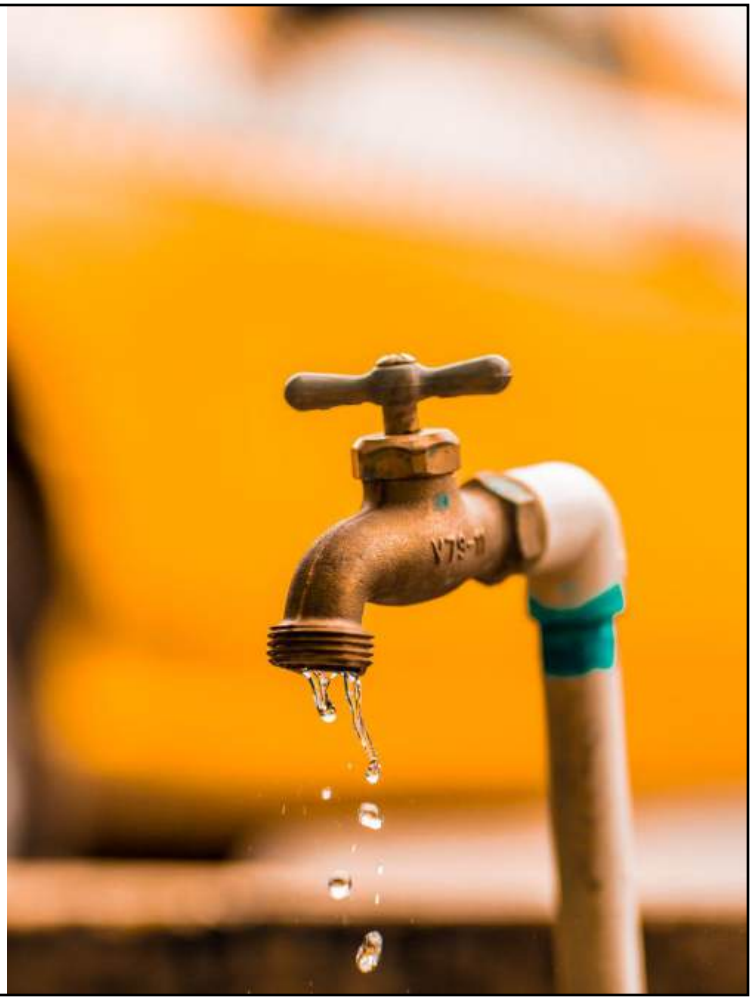
- restrict access
- monitor output

Take privacy into account

- preprocess data, e.g. obfuscate sensitive data in trainings data
- use differential privacy
- in general evaluate model privacy

Data Leakage

Data might get stolen and published, causing material or immaterial damage



Data Leakage

HDFC Bank's NBFC arm confirms data leak of customers

2 min read • Arti Singh

07 Mar 2023, 09:01 PM IST

Data breach confirmed by Ray-Ban after leak of over 70M customers' records

SC Staff May 23, 2023

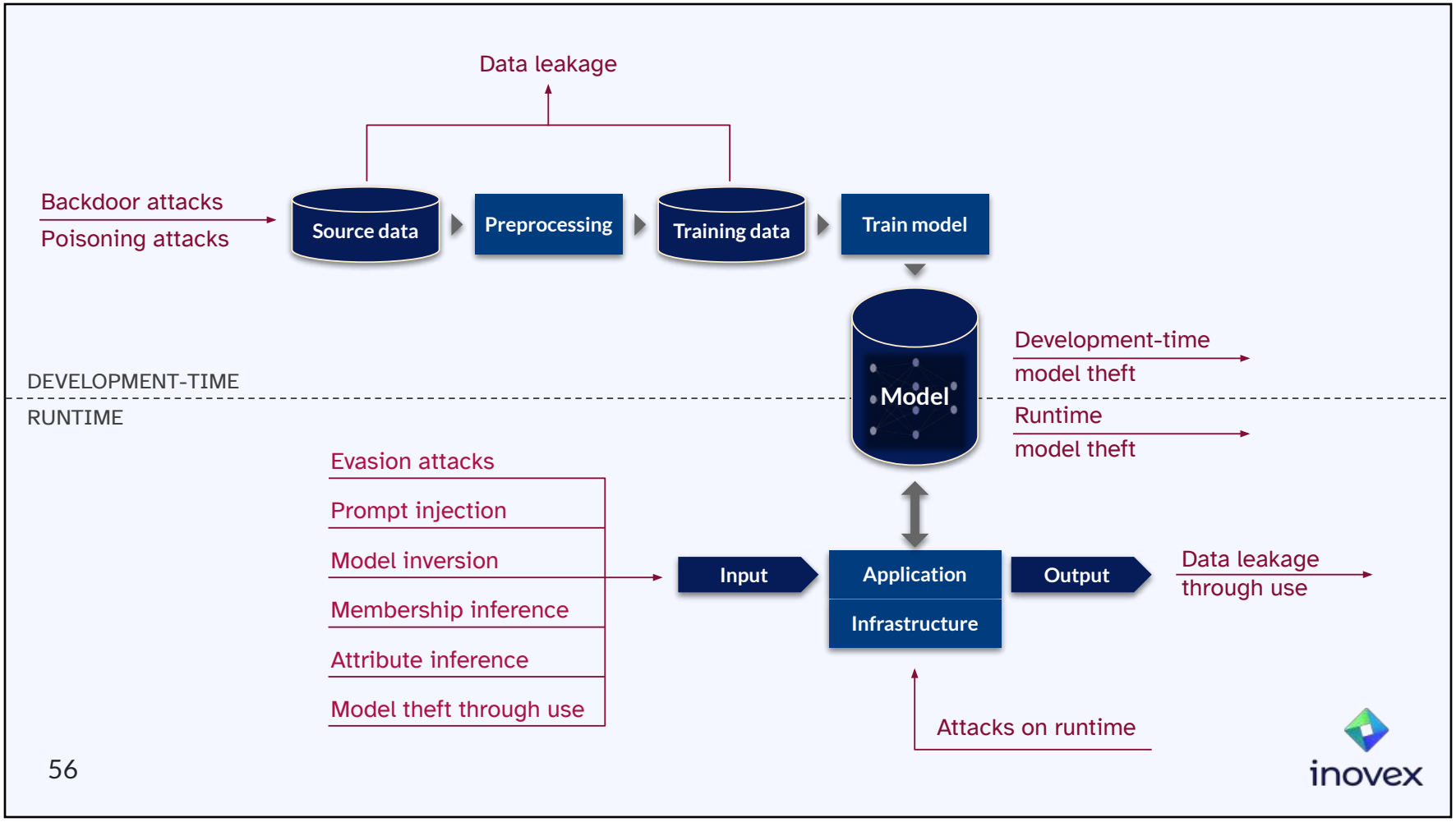
Prevent Data Leakage

Secure operation infrastructure

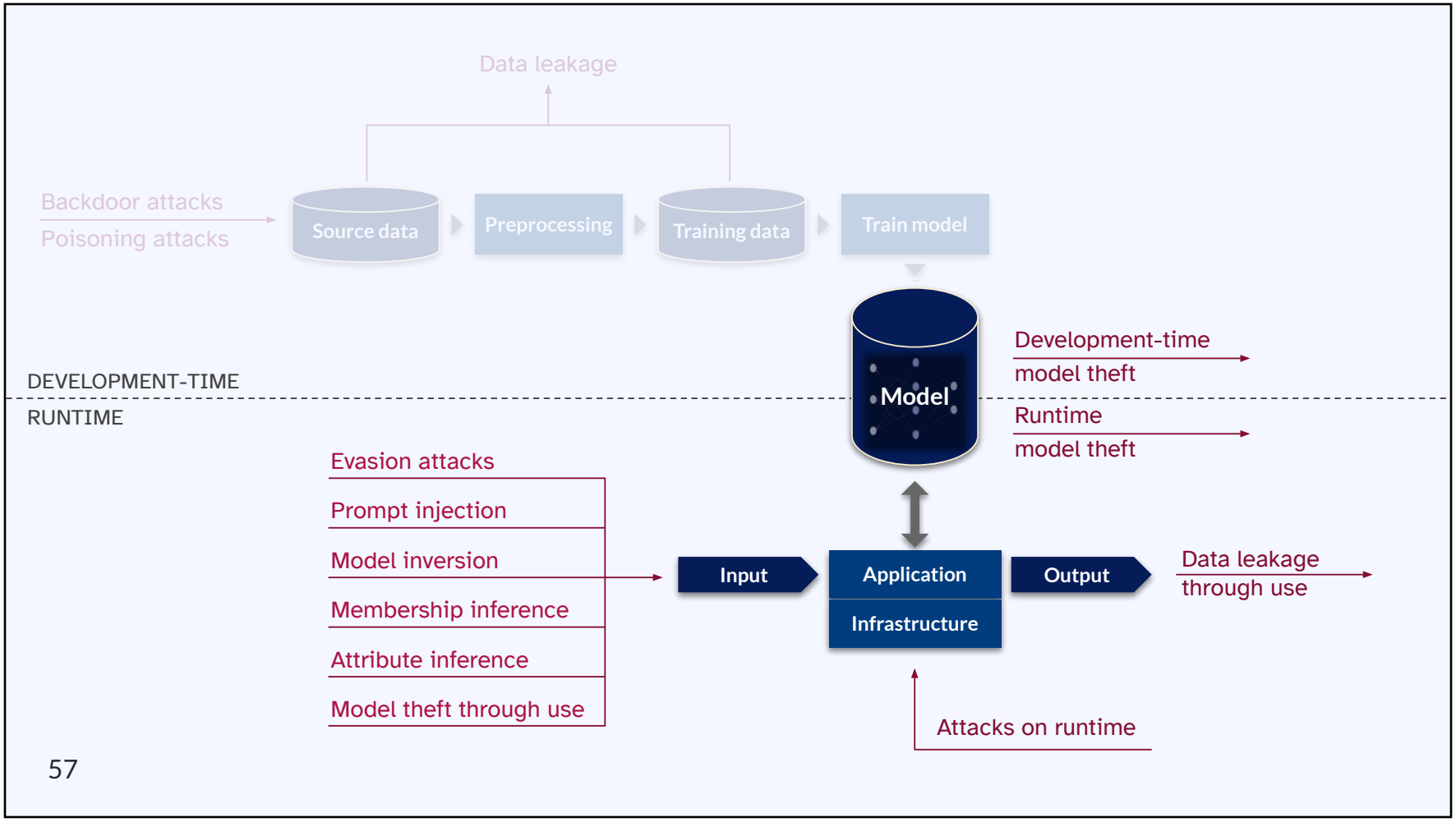
- keep environments separated
- restrict access
- defense in depth to mitigate effect of flaws

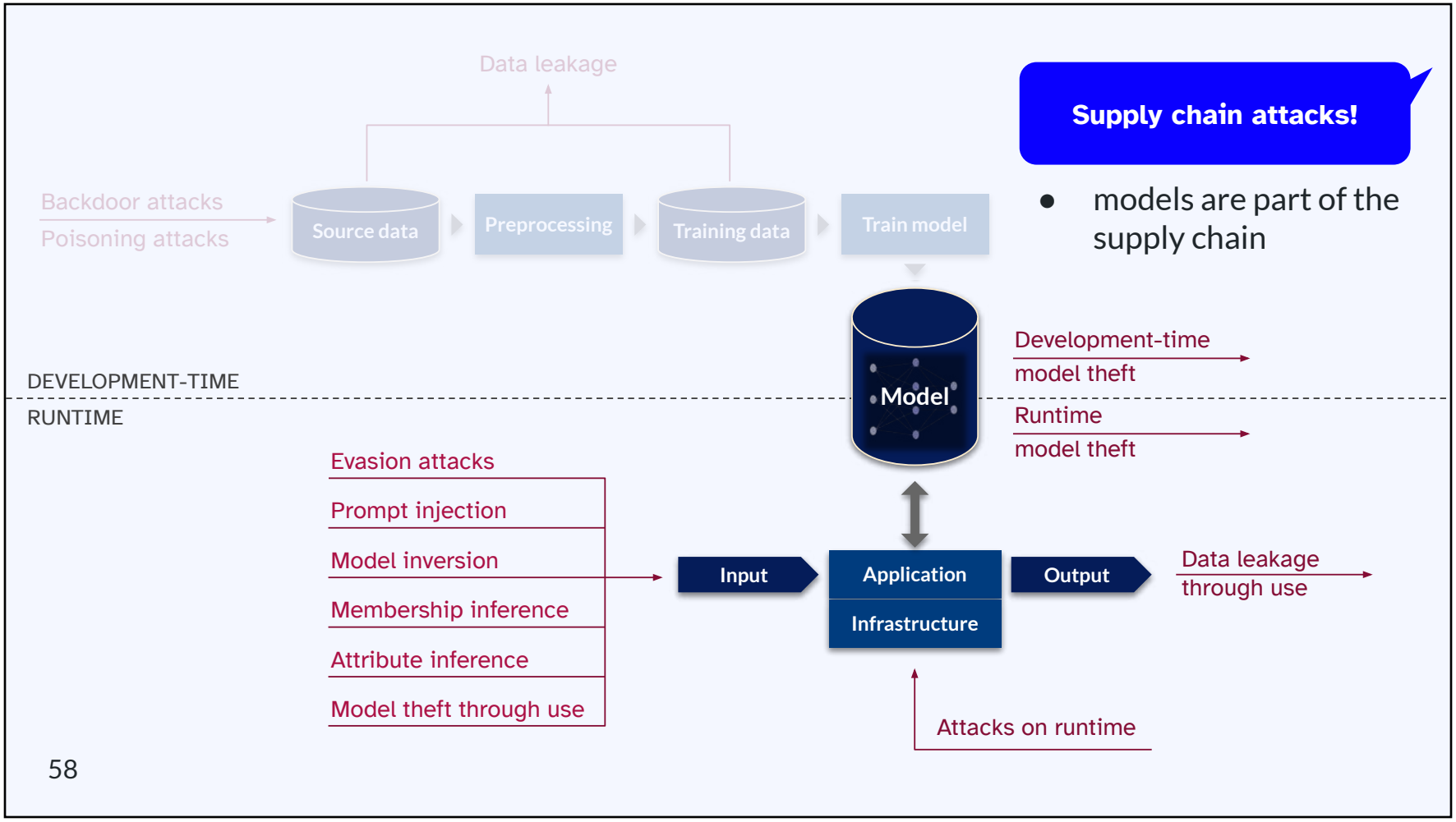
Secure training infrastructure

- minimize data usage, anonymize data
- reduce retention of data
- encrypt & restrict access


Data leakage
Backdoor attacks
Poisoning attacks
Source data
Preprocessing
Training data
Train model
Model
Development-time
model theft
Runtime
model theft
DEVELOPMENT-TIME
RUNTIME
Evasion attacks
Prompt injection
Model inversion
Membership inference
Attribute inference
Model theft through use
Input
Application
Infrastructure
Output
Data leakage
through use
Attacks on runtime

inovex


Data leakage
Backdoor attacks
Poisoning attacks
Source data
Preprocessing
Training data
Train model
Model
Development-time
model theft
Runtime
model theft
DEVELOPMENT-TIME
RUNTIME
Evasion attacks
Prompt injection
Model inversion
Membership inference
Attribute inference
Model theft through use
Input
Application
Infrastructure
Output
Data leakage
through use
Attacks on runtime


Data leakage
Supply chain attacks!
Backdoor attacks
Poisoning attacks
Source data
Preprocessing
Training data
Train model
models are part of the supply chain
Model
Development-time model theft
DEVELOPMENT-TIME
RUNTIME
Runtime model theft
Evasion attacks
Prompt injection
Model inversion
Membership inference
Attribute inference
Model theft through use
Input
Application
Infrastructure
Output
Data leakage through use
Attacks on runtime

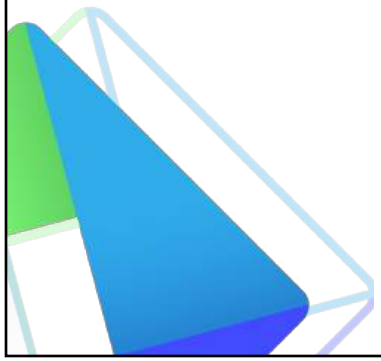
Don’t forget the traditional part

Typical attacks or security risks also occur here

- Supply Chain Attacks
- Attacks on authentication and authorization
- Logic and design flaws
- Security Misconfiguration
- Missing Logging/Monitoring/Alerting

+ Integration into development and business processes

Which best practices exist for secure development of AI software?



Guarantee data protection and security

Data origin	• Create reliance with data lineage • Ensure data cannot be changed
Privacy	• Prevent traceability of sensitive data • Work with anonymized data
Data storage	• Separate data and development environments • Ensure data is encrypted • Minimize data usage - also by time
Data quality	• Conduct regular evaluations • Monitor data drift

inovex

Avoid attacks on model development I

Developing your own model: Model design and parameters as well as data are in your own hands and your custom environment

Training	• Train decentrally • Ensure reproducibility through “Golden Dataset” • Keep design decisions transparent
Inference	• Avoid overfitting • Robustness through noise/adversarial examples • Reduce model output to the minimum

inovex

Avoid attacks on model development II

Usage of external models: Model is

- open source and self-hosted, i.e. model and data remain in its own environment
- closed source, i.e. data is sent to another environment

Regular updates	• open-source: regularly check dependencies, audits, source code • closed-source: Establish procedures for dealing with security incidents
Control relies with the provider	• How is the data processed and is it passed on? • How is the API secured? • What data was used for training?

inovex

Ensure a secure deployment

Secure Infrastructure	• Model transfer must be secure • Implement secure communication techniques • Integrate model surveillance
Separate production application	• Completely separate the production environment from all other environments • Separate environment for model training (avoid local training)
Access restriction	• Identify groups of people and interests • Access controls and authorization management • Restrict usage with rate limiting

inovex

Prevent pperation vulnerabilities

Feedback loops	• Avoid direct feedback to influence model adaptations • Manual evaluations are particularly important in early stages
Monitoring	• Restrict usage by stages • Evaluate which metrics for quality (accuracy, robustness), fairness (group fairness), .. meet your needs • Keep in mind: good metrics != good operations
Best practises	• Logging should be automated & centered • Errors should be comprehensible and attacks should be reproducible

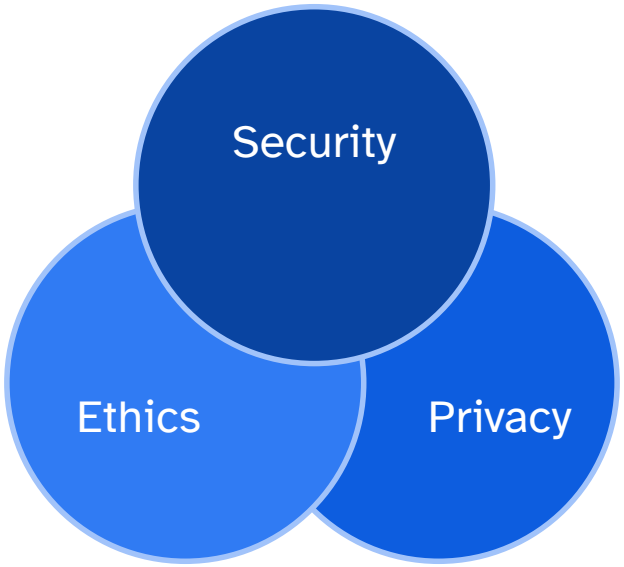
inovex

General AI best practices

→ Apply known best practices

- **Transparency**: documentation of model, data processing, feature extraction, potential bias and consequences
- **Traceability**: document development decisions
- **Explainability**: outputs and results should be explainable even when the model is a black-box model itself
- **Quality assurance:** check the code quality regularly to avoid vulnerabilities and risks

Connected disciplines



- develop risk scenarios
- check for bias and misrepresentation in data

- special care required when processing PII
- minimize data, limit storage
- use anonymization

Further Resources

- OWASP
  - AI Exchange
  - OWASP ML Top Ten
  - OWASP Top Ten for LLMs
- BSI Leitfaden
  - AI Security Concerns in a Nutshell
  - Provision or Use of External Data or Trained Models
  - together with international partners: Engaging with Artificial Intelligence
-

Takeaways

AI software is also software - known methods and measures remain useful and important

New threats and attacks emerge and need to be covered

Transfer existing knowledge accordingly and adapt threat model

Vielen Dank!

inovex Security Meetup
Mastering the Security Maze
27.02.2025 - München

Heise devSec-Thementag
KI & Security
08.04.2025 - online

@clemens@infosec.exchange

/clemens-huebner

clemens.huebner@inovex.de

@inovexlife

blog.inovex.de