Software Testing In Age of Data Privacy: A Balancing Act

Mark Grechanik, Chen Fu, Qing Xie

Testing Database-centric applications (DCAs)

Database-centric applications (DCAs) are common in enterprise computing, and they use nontrivial databases

- Testing of DCAs is increasingly outsourced to testing centers in order to achieve lower cost and better quality
- Databases should also be made available to test engineers, so that they can test using real data

However, the databases contain sensitive information, and they often cannot be disclosed to anyone outside the organizations that own the DCAs

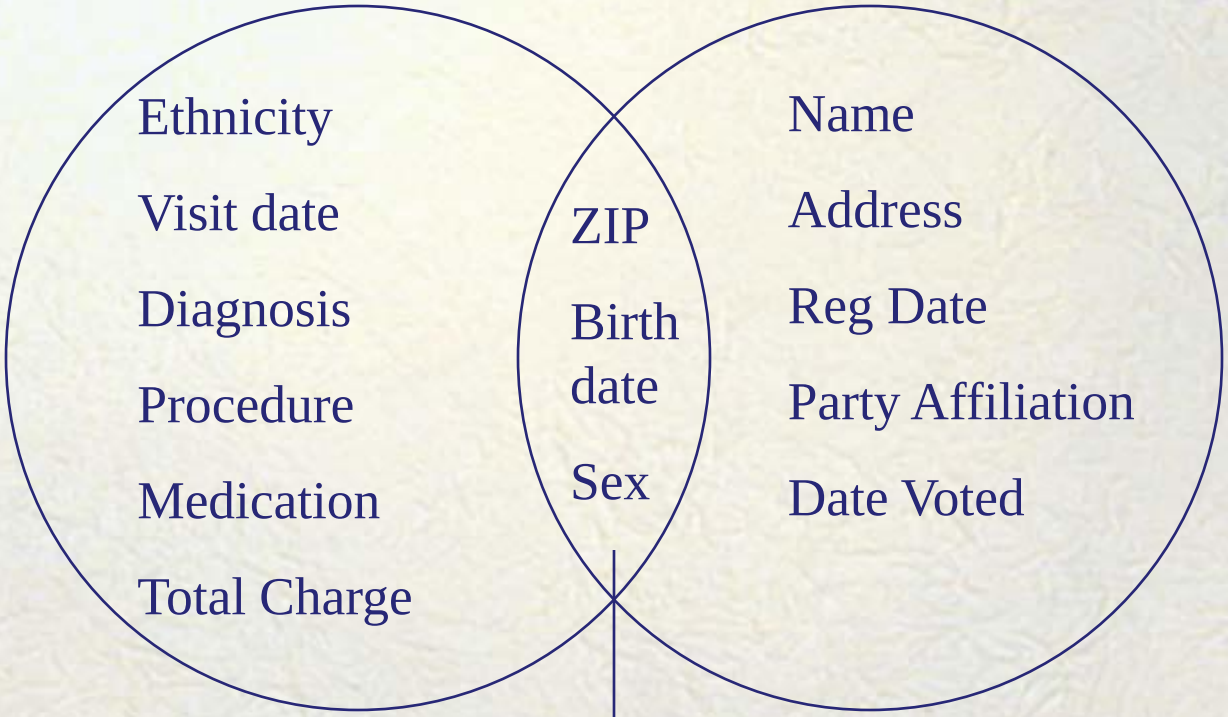
- Currently, testing is either performed with fake data or real data that is exposed to external testing organizations

Just Remove Names

- And SSN, addresses and phone numbers maybe.

- Really?

Mastercard Commercial



"Anonymized" State Employee Health Data

$0

Voters List

$20

Governor of Massachusetts William Weld's Health Record

Priceless

source: k-anonymity is summarised in a paper by LATANYA SWEENEY(k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY)

↓

Governor of Massachusetts William Weld's Health Record

Priceless

Medical Insurance DCA

Rec	Age	ZipCode	Nationality	Disease
1	42	52000	American	Ulcer
2	47	53000	Palauan	Viral
3	51	32000	American	Heart disease
4	**55**	**32000**	**Japanese**	**Gastritis**
5	62	51000	Palauan	Dyspepsia
6	67	35000	American	Dyspepsia

Quasi-Identifiers (QIs)

The individual is a 55-year old Japanese who lives in zip code 32000. If we know that there is a single 55-year old Japanese who lives in this zip code, we can infer that this person suffers from gastritis.

Medical Insurance DCA

Rec	Age	ZipCode	Nationality	Disease
1	42	52000	American	Ulcer
2	47	53000	Palauan	Viral
3	51	32000	American	Heart disease
4	55	32000	Japanese	Gastritis
5	62	51000	Palauan	Dyspepsia
6	67	35000	American	Dyspepsia

```
if( nationality==“Japanese” &&
    age > 40 && age < 60 ) {
        f(disease);
}
```

Medical Insurance DCA

Rec	Age	ZipCode	Nationality	Disease
1	42	52000	American	Ulcer
2	47	53000	Palauan	Viral
3	51	32000	American	Heart disease
4	55	32000	Japanese	Gastritis
5	62	51000	Palauan	Dyspepsia
6	67	35000	American	Dyspepsia

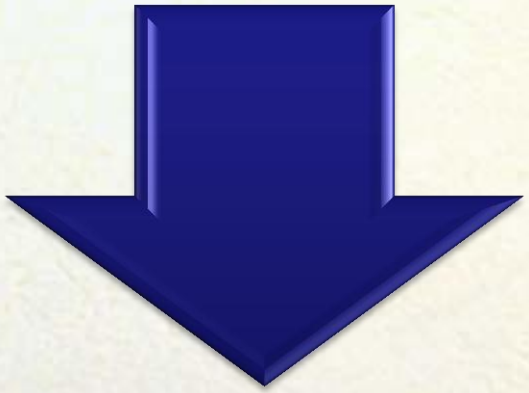
```
if( nationality==“Japanese” &&
    age > 40 && age < 60 ) {
        f(disease);
}
```

Medical Insurance DCA

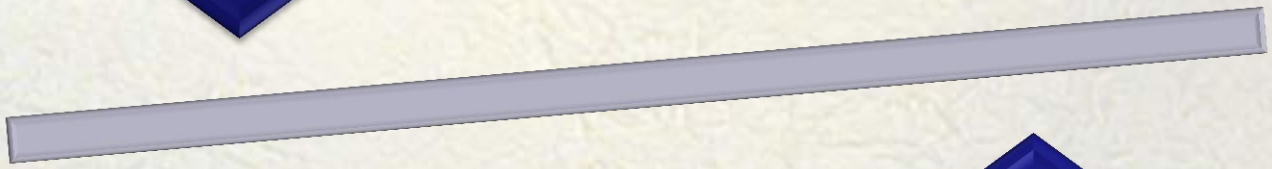
Rec	Age	ZipCode	Nationality	Disease
1	42	52000	American	Ulcer
2	47	53000	Palauan	Viral
3	51	32000	American	Heart disease
4	55	32000	Japanese	Gastritis
5	62	51000	Palauan	Dyspepsia
6	67	35000	American	Dyspepsia

```
if( nationality==“Japanese” &&
    age > 40 && age < 60 ) {
        f(disease);
}
```

Conflicting Goals



make testing as realistic as possible



protect real data from testers who need them to make testing realistic

Generate Fake Data

- To look "realistic"
  - Must capture complex structure of the original data
  - Must understand explicit and implicit connections between data

Example Of Generating Semantically Incorrect Data

- A test data generation tool for insurance application creates an entry in the database for a man who suffers from ***gestational diabetes***.
- Does it make sense?

Clean Room Testing

- Physically Restricted
- Security Clearance
- No internet
- No USB
- No CD
- No Phone
- No camera
- Personal search

A Textbook Approach

- Data Anonymization
  - Experts locate Quasi-identifiers
  - Select subset of quasi-identifiers to scramble, until privacy goal is reached.
  - Don't take application behavior into account.

Our Solution

With our solution for Testing Applications with Data Anonymization (TaDa), organizations can preserve test coverage while releasing DCAs to external test centers without disclosing sensitive information

- TaDa combines dynamic symbolic execution with data anonymization algorithms to determine how protecting values of database attributes affects test coverage, thereby enabling DCA owners to determine what parts of DCAs to test in-house before sanitizing and releasing DCAs to test centers

TaDa enables analysts to decide how to balance solutions for the conflicting goals of providing a minimum cost anonymization solution while preserving test coverage of DCAs

- Guide experts to scramble data attributes that hurts coverage the least.

Attributes Ranking

- Number of statements affected
  - Given a data attribute, we know:
    - how it is used in which branches
    - How many statements are control dependent on these branches
- Goal:
  - Selecting attributes that affects least number of statements to scramble

Different Qis Affect Test Coverage Differently