**Base rate fallacy:** Here is the calculations for Question 5 of Quiz 3:

X: being accepted by the system Y: being a legitimate user  
🡺 Pr(X | not Y) + Pr(not X | not Y) = 1

🡺 Pr(not X | Y) + Pr(X | Y) = 1

🡺 Pr(X | not Y) + Pr(X | Y) = 1

Pr(X| not Y) = FAR = 4% 🡺 Pr(not X | not Y) = 1 - 4% = 96%  
  
Pr(not X | Y) = FRR = 15% 🡺 Pr(X | Y) = 1 - 15% = 85%  
  
Pr(Y) = 95% 🡺 Pr(not Y) = 1 - 95% = 5%  
  
1) Pr(X) = Pr( X | not Y) Pr( not Y) + Pr( X | Y) Pr(Y) 🡺 4% \* 5% + 85%\* 95% = 80.95%

🡺 Pr(not X) = 1 - 80.95% = 19.05%

2) Pr(not X) =  Pr( not X | not Y) Pr( not Y) + Pr( not X | Y) Pr(Y)

🡺 96% \* 5% + 15% \* 95% = 19.05%

A) The probability of the person being an intruder if they are rejected is Pr(not Y | not X)  
  
Pr(not Y | not X) = Pr(not X | not Y) Pr( Not Y) / Pr(Not X) = 96% \* 5% / 19.05% = ~25.19%  
  
B) The probability of the person being a legitimate user if they are accepted is Pr(Y | X)  
  
Pr( Y | X ) = Pr (X | Y) Pr(Y) / Pr(X) = 85% \* 95% / 80.95% = ~99.75%

**K-anonymization and l-diversity:**

Make 3-anonymus and find the largest value of l for which the table is l-diverse.

|  |  |  |
| --- | --- | --- |
| **ZIP** | **DoB** | **Disease** |
| 50232 | 1965-07-11 | Covid19 |
| 50211 | 1965-08-01 | Heart Disease |
| 50233 | 1965-07-12 | Covid19 |
| 50255 | 1950-12-12 | Cancer |
| 50512 | 1954-01-01 | Covid19 |
| 31403 | 2020-08-01 | No Teeth |
| 31402 | 2001-01-01 | Covid19 |
| 31401 | 2012-01-01 | Flu |
| 31401 | 2001-01-01 | Heart Disease |

In this case, we are considering the values in the Disease field as the sensitive data and ZIP and DoB as quasi-identifiers. We are trying to reduce granularity of the quasi-identifiers, just enough to get groups of 3 or more records with equal quasi-identifiers. To do so, we replace symbols with \*. We start with the least significant digits of ZIP code and values in DoB and move towards the most significant digits or values. The result after the 3-anonymization looks as follows:

|  |  |  |
| --- | --- | --- |
| **ZIP** | **DoB** | **Disease** |
| 50\*\*\* (or 50\*) | 19\*\*-\*\*-\*\* (19\*-\*-\*) | Covid19 |
| 50\*\*\* | 19\*\*-\*\*-\*\* | Heart Disease |
| 50\*\*\* | 19\*\*-\*\*-\*\* | Covid19 |
| 50\*\*\* | 19\*\*-\*\*-\*\* | Cancer |
| 50\*\*\* | 19\*\*-\*\*-\*\* | Covid19 |
|  |  |  |
| 3140\* | 20\*\*-\*\*-\*\* | No Teeth |
| 3140\* | 20\*\*-\*\*-\*\* | Covid19 |
| 3140\* | 20\*\*-\*\*-\*\* | Flu |
| 3140\* | 20\*\*-\*\*-\*\* | Heart Disease |

Then for each quasi identifier we count the distinct values of sensitive data:

For ZIP= 50\*\*\* and DoB= 19\*\*-\*\*-\*\*, Covid19, Heart Disease, and Cancer are possible, hence 3 distinct values.

For ZIP=3140\* and DoB = 20\*\*-\*\*-\*\*, No Teeth, Covid19, Flue, and Heart Disease are possible, i.e., 4 distinct values.

The minimum of these distinct values, that is 3, is the maximum value of l for which this table is l-diverse.

**Buffer overflow:** An attacker can make the following function to return 1, without knowing that the password is “password123”.

**bool authenticate(){**

**bool flag = true;**

**char password[20];**

**gets(password);**

**if(! strcpy(password, “password123”)){**

**flag = false;**

**}**

**return flag;**

**}**

All the attacker needs to do is to enter a string of length 21. Doing that, all the memory bytes in the character array password is written over, and the last character and the null terminator are written in the memory bytes of the variable flag. If the last character of the string is not ‘\0’, the value of flag is non-zero which will evaluate to true;