attack

June 27, 2024

1 Progetto di Internet Security

1.0.1 Autore: Giulio Pedicone

1.0.2 Matricola: 1000031068

1.0.3 Prima Fase: Visualizzazione del dataset non anonimizzato

In questa fase iniziale, procediamo alla visualizzazione di un dataset contenente informazioni mediche di cittadini americani. Il dataset è gratuito ed è stato scaricato dalla piattaforma Kaggle. Può essere trovato al seguente link e include le seguenti informazioni sui pazienti:

- **Name**: Nome del paziente associato al record sanitario.
- **Age**: Età del paziente al momento dell'ammissione, espressa in anni.
- **Gender**: Genere del paziente, può essere "Maschio" o "Femmina".
- **Blood Type**: Gruppo sanguigno del paziente, come "A+", "O-", eccetera.
- **Medical Condition**: Condizione medica primaria o diagnosi associata al paziente, come "Diabete", "Ipertensione", "Asma", e altre.
- **Date of Admission**: Data di ammissione del paziente presso la struttura sanitaria.
- **Doctor**: Nome del medico responsabile delle cure durante l'ammissione del paziente.
- **Hospital**: Identifica l'ospedale o la struttura sanitaria dove il paziente è stato ammesso.
- **Insurance Provider**: Fornitore dell'assicurazione del paziente, che può essere tra diversi opzioni come "Aetna", "Blue Cross", "Cigna", "UnitedHealthcare" e "Medicare".
- **Billing Amount**: Importo fatturato per i servizi sanitari forniti al paziente durante l'ammissione, espressa come numero decimale.
- **Room Number**: Numero della stanza dove il paziente è stato alloggiato durante l'ammissione.
- **Admission Type**: Specifica il tipo di ammissione, come "Emergenza", "Elettiva" o "Urgente", riflettendo le circostanze dell'ammissione.
- **Discharge Date**: Data di dimissione del paziente dalla struttura sanitaria, basata sulla data di ammissione e un numero casuale di giorni entro un intervallo realistico.
- **Medication**: Identifica una medicazione prescritta o somministrata al paziente durante l'ammissione, come "Aspirina", "Ibuprofene", "Penicillina", "Paracetamolo" e "Lipitor".
- **Test Results**: Descrive i risultati di un test medico eseguito durante l'ammissione del paziente. Possibili valori includono "Normale", "Anormale" o "Inconcludente".

Il dataset contiene un totale di 55,500 record.

[ ]:
```
import pandas as pd
```

```
pd.set_option("display.max_columns", None)  # Mostra tutte le colonne
file_path = "Allegati/healthcare_dataset.csv"

df = pd.read_csv(file_path)

display(df.head())  # Stampa solo i primi 5 record
```

```
            Name  Age  Gender Blood Type Medical Condition Date of Admission  \
0  Bobby JacksOn   30    Male         B-            Cancer        2024-01-31
1   LesLie TErRy   62    Male         A+           Obesity        2019-08-20
2    DaNnY sMitH   76  Female         A-           Obesity        2022-09-22
3   andrEw waTtS   28  Female         O+          Diabetes        2020-11-18
4  adrIENNE bEll   43  Female        AB+            Cancer        2022-09-19

             Doctor                     Hospital Insurance Provider  \
0     Matthew Smith             Sons and Miller         Blue Cross
1   Samantha Davies                      Kim Inc           Medicare
2  Tiffany Mitchell                     Cook PLC              Aetna
3      Kevin Wells  Hernandez Rogers and Vang,           Medicare
4    Kathleen Hanna                 White-White              Aetna

   Billing Amount  Room Number Admission Type Discharge Date   Medication  \
0    18856.281306          328         Urgent     2024-02-02  Paracetamol
1    33643.327287          265      Emergency     2019-08-26    Ibuprofen
2    27955.096079          205      Emergency     2022-10-07      Aspirin
3    37909.782410          450       Elective     2020-12-18    Ibuprofen
4    14238.317814          458         Urgent     2022-10-09   Penicillin

   Test Results
0        Normal
1  Inconclusive
2        Normal
3      Abnormal
4      Abnormal
```

2 Fase 2: Pseudonimizzazione del dataset

In questa fase, si vogliono proteggere le informazioni sensibili del dataset come il nominativo del paziente per rendere il dataset pseudo-anonimizzato. Utilizziamo la tecnica di sostituire il nominativo della persona creando una funzione che sostituisce il nome tramite la seguente operazione matematica:

$$

newName = _{i=1}^{n} ASCII(c_i) + Room Number

$$ dove:

- c_i sono i caratteri di oldName

- ASCII(c_i) rappresenta il valore ASCII del carattere c_i

Tronchiamo inoltre il dataset a soli 1000 valori per velocizzare le operazioni successive.

[ ]:
```
output_file = "Allegati/truncated_dataset.csv"
df.head(1000).to_csv(output_file, index=False)  # Tronca il dataset a soli 1000 valori

new_dataframe = pd.read_csv(output_file)

def pseudonimize(name, room):
    name_ascii_sum = sum(ord(char) for char in name)  # Somma dei valori ASCII dei caratteri in 'name'
    return name_ascii_sum + room

# Modifica la colonna "Name" nel DataFrame utilizzando la funzione pseudonimize
for index, row in new_dataframe.iterrows():
    new_name = pseudonimize(row["Name"], row["Room Number"])
    new_dataframe.loc[index, "Name"] = new_name

# Visualizza le prime righe del DataFrame con i nomi pseudonimizzati
display(new_dataframe.head())
```

```
   Name  Age  Gender Blood Type Medical Condition Date of Admission  \
0  1535   30    Male         B-            Cancer        2024-01-31
1  1341   62    Male         A+           Obesity        2019-08-20
2  1164   76  Female         A-           Obesity        2022-09-22
3  1590   28  Female         O+          Diabetes        2020-11-18
4  1551   43  Female        AB+            Cancer        2022-09-19

             Doctor                    Hospital Insurance Provider  \
0     Matthew Smith             Sons and Miller         Blue Cross
1   Samantha Davies                      Kim Inc           Medicare
2  Tiffany Mitchell                    Cook PLC              Aetna
3       Kevin Wells  Hernandez Rogers and Vang,           Medicare
4    Kathleen Hanna                 White-White              Aetna

   Billing Amount  Room Number Admission Type Discharge Date   Medication  \
0    18856.281306          328         Urgent     2024-02-02  Paracetamol
1    33643.327287          265      Emergency     2019-08-26    Ibuprofen
2    27955.096079          205      Emergency     2022-10-07      Aspirin
3    37909.782410          450       Elective     2020-12-18    Ibuprofen
4    14238.317814          458         Urgent     2022-10-09   Penicillin

   Test Results
0        Normal
1  Inconclusive
2        Normal
```

```
3      Abnormal
4      Abnormal
```

3 Fase 3: Creazione di un meccanismo di recupero

Quando pseudonimizziamo un dataset e abbiamo bisogno successivamente di risalire al nominativo originale del paziente, è necessario creare una tabella di mappatura tra il nominativo del paziente e il suo pseudonimo. Questa tabella deve essere conservata in modo sicuro e protetto poiché contiene tutte le coppie tra nominativo e pseudonimo presenti nel dataset pseudonimizzato.

[ ]:
```
df_nomi = df['Name'].head(1000)
new_df_nomi = new_dataframe['Name'].head(1000)

# Creiamo un nuovo dataframe con i nomi ottenuti
nomi_dataframe = pd.DataFrame({
    'Nome': df_nomi,
    'Nome Pseudonimizzato': new_df_nomi
})

# Scriviamo il dataframe in un file CSV
nomi_dataframe.to_csv('Allegati/tabellaMappatura.csv', index=False)

display(nomi_dataframe.head())
```

```
            Nome Nome Pseudonimizzato
0  Bobby JacksOn                 1535
1   LesLie TErRy                 1341
2    DaNnY sMitH                 1164
3   andrEw waTtS                 1590
4  adrIENNE bEll                 1551
```

4 Fase 4: Creazione della nuova tabella pseudonimizzata

Abbiamo ora pseudonimizzato il dataset. Salviamo il nuovo dataset pseudonimizzato all'interno di un file chiamato `pseudonymized.csv`.

[ ]:
```
new_dataframe.to_csv('Allegati/pseudonymized.csv', index=False)
```

5 Fase 5: Inizio dell'attacco di linking

Durante questa fase, l'attaccante dispone di due dataset cruciali:

- Il file `pseudonymized.csv` che contiene informazioni sanitarie di vari pazienti, con i loro nominativi pseudonimizzati.
- Il file `blood_donation_dataset.csv` che include dettagli su donatori di sangue, specificamente:
  - Nome: Il nome completo del donatore di sangue.
  - Età: L'età del donatore al momento della donazione.

- Genere: Il genere del donatore, indicato come “Maschio” o “Femmina”.
- Gruppo Sanguigno: Il tipo di sangue del donatore.
- Data della Donazione: La data in cui è stata effettuata la donazione.

L’obiettivo dell’attacco è ricondurre all’identificazione dei pazienti all’interno del dataset pseudonimizzato, sfruttando le informazioni disponibili nel dataset delle donazioni di sangue che contiene i nomi completi in chiaro.

[ ]:
```
file_path = "Allegati/blood_donation_dataset.csv"
blood_donation = pd.read_csv(file_path)

display(blood_donation.head())  # Stampa solo i primi 5 record
```

```
             Name  Age  Gender Blood Type Date of Donation
0   AMANDa DURhAm   46  Female         O+       2021-06-16
1     Erin oRTEga   43    Male        AB-       2023-05-24
2   BETh sChwaRTZ   58    Male        AB-       2024-03-29
3   JASmine sHort   40  Female         O-       2021-02-23
4  dAnIElle lOPeZ   37    Male         O-       2020-11-20
```

Supponiamo adesso di voler trovare il nominativo associato alla entry **n°546** del nostro dataset anonimizzato

[ ]:
```
file_path = "Allegati/pseudonymized.csv"
pseudonymized = pd.read_csv(file_path)

indiceDaCercare = 546 # Modificare questo valore per cercare un altro utente

display(pseudonymized.iloc[indiceDaCercare].to_frame().T)
```

```
     Name Age  Gender Blood Type Medical Condition Date of Admission  \
546  1623  32  Female         B+            Asthma        2020-02-20

            Doctor      Hospital Insurance Provider Billing Amount  \
546  Joseph Miller  Snyder-Perry           Medicare    17862.69225

    Room Number Admission Type Discharge Date   Medication  Test Results
546         482       Elective     2020-02-21  Paracetamol  Inconclusive
```

Dalla seguente voce, possiamo estrarre informazioni utili come:

- Età: 32 anni
- Genere: Femmina
- Gruppo Sanguigno: B+

Queste informazioni sono fondamentali per condurre un attacco di collegamento tramite il nostro dataset `blood_donation_dataset.csv`. Ora cerchiamo un record che soddisfi tutte e tre queste condizioni.

[ ]:
```
def search(age, gender, bloodType):
    for index, row in blood_donation.iterrows():
```

```
        if (
            row["Age"] == age
            and row["Gender"] == gender
            and row["Blood Type"] == bloodType
        ):
            return index

age = pseudonymized.iloc[indiceDaCercare]["Age"]
gender = pseudonymized.iloc[indiceDaCercare]["Gender"]
bloodType = pseudonymized.iloc[indiceDaCercare]["Blood Type"]

result = search(age, gender, bloodType)
display(blood_donation.iloc[result].to_frame().T)
```

```
              Name Age  Gender Blood Type Date of Donation
367  Connor BARTOn  32  Female         B+       2020-02-20
```

6 Fase 6: Conclusione dell'attacco di linking

L'attacco di linking ha identificato un record di una donatrice chiamata "Connor BARTOn" che sembra corrispondere al paziente cercato. Supponiamo che l'attaccante conosca l'algoritmo utilizzato per la pseudonimizzazione.

Per verificare la corrispondenza, sottraiamo il numero della stanza dal nome pseudonimizzato del paziente, ottenendo la somma dei caratteri ASCII del nome. Confrontiamo poi questa somma con quella del nome trovato nel dataset originale.

[ ]:
```
# Somma dei caratteri ASCII del nome nel database pseudonimizzato
check = pseudonymized.iloc[indiceDaCercare]["Name"] - pseudonymized.
 ↪iloc[indiceDaCercare]["Room Number"]

# Nome trovato nel DataFrame blood_donation
nomeTrovato = blood_donation.iloc[result]["Name"]

# Calcolo dell'ASCII dei nomi trovati
nomeTrovatoASCII = sum(ord(char) for char in nomeTrovato)

# Confronto degli ASCII
if nomeTrovatoASCII == check:
    print("Corrispondenza trovata")
else:
    print("Corrispondenza non trovata")
```

```
Corrispondenza trovata
```