

Activity X: Computer Forensics

By Saenyakorn Siangsanoh 6232035721 and Poravee Binhayearason 6230314421

สามารถดู Resource เต็ม ๆ ได้ที่ [2110413-COMP-SECURITY Activity 11](#)

Table of Contents

- [1](#)
 - [Answer](#)
- [2](#)
 - [Answer](#)
- [3](#)
 - [Answer](#)
- [Assets](#)

1

How many words do we need to correctly identify the person?

Answer

เราได้ทำการเขียนโปรแกรมทำการสร้าง digraph โดยเชื่อว่าเป็น biometric ของบุคคลที่แตกต่างกัน ประกอบด้วย

code ของ การสร้างคำเพื่อใช้ทดสอบ

```
1 def clean_text(text: str) -> str:
2     new_text = []
3     for c in text:
4         if c.isalpha() or c.isspace():
5             new_text[-1] = new_text[-1] + c.lower()
6     return " ".join(new_text)
```

code ของ การสร้าง digraph

```
111 t1 = time()
112 prev_char = None
113 buffer = []
114 sub_result = []
115 while True:
116     print("".join(buffer), end="\r")
117     typed_char = readchar.readchar()
118     t2 = time()
119     if typed_char == "\r" or ord(typed_char) == 10:
120         break
121     elif typed_char == "\b":
122         if len(buffer) >= 1:
123             buffer.pop()
124             if len(sub_result) >= 1:
125                 sub_result.pop()
126
127             prev_char = None
128     else:
129         buffer.append(typed_char)
130         if prev_char != None:
131             sub_result.append((prev_char+typed_char, t2-t1))
132         prev_char = typed_char
133
134     t1 = t2
135
136 for k, v in sub_result:
137     if k not in result:
138         result[k] = (v, 1)
139     else:
140         result[k] = ((result[k][0]*result[k][1] + v) /
141                     (result[k][1]+1), result[k][1]+1)
```

code ของการ load, store digraph ที่ได้

```

1  import json
2
3
4  def store_result(result: dict, name: str) -> None:
5      with open(f'{name}.json', 'w') as f:
6          json_string = json.dumps(result)
7          f.write(json_string)
8
9
10 def load_result(name: str) -> None:
11     with open(f'{name}.json', 'r') as f:
12         json_string = f.read()
13         result = json.loads(json_string)
14         return result
15
16
17 def remove_result(name: str) -> None:
18     import os
19     os.remove(f'{name}.json')

```

code ของการตรวจสอบ similarity

```

202 # TODO: identify the user (loop over all user in database -> assume we have 2 users in database)
203 ree_result = manage_result.load_result("ree")
204 yu_result = manage_result.load_result("yu")
205 user_results = [ree_result, yu_result]
206 # We define the most similar user as the user with the smallest distance
207 user_score = [0]*len(user_results)
208 for idx, result in enumerate(user_results):
209     # Get distance for all typing pattern
210     for k, v in unk_result:
211         if k in result:
212             user_score[idx] += abs(result[k][0] - v[0])
213
214 # Get min score
215 pos = np.argmin(user_score)
216 if pos == 0:
217     print("\nYou are ree")
218 else:
219     print("\nYou are yu")

```

เพื่อที่จะให้มี digraph ที่มีขนาดใหญ่พอที่จะใช้เปรียบเทียบ จำเป็นจะต้องใช้อย่างน้อย 10 คำ ที่ต้องมีคู่ของตัวอักษรไม่ซ้ำกัน

2

Do you think this method is scalable? (to thousand persons) for either recognition system or identification system. Please provide your analysis.

Answer

ไม่ เพราะ มีหลายคนที่มีวิธีการเขียนที่เหมือนกันมาก เช่น พวกนักแข่งพิมพ์จะพิมพ์กันได้ยารวดเร็วทำให้ error ที่ได้น้อยจนไม่สามารถระบุได้ว่าใครเป็นใคร เมื่อเทียบกับคนธรรมดาที่พิมพ์ช้ากว่า

3

Will you use this kind of authentication in your system? Please also provide a reason

Answer

ไม่ใช้ เพราะเป็นการ authenticate ที่อ่อนเกินไป และความเป็นตัวตนน้อยเกินไป หมายความว่าอาจจะมีคนอื่นที่มีวิธีการพิมพ์เหมือนเราก็ได้

Assets

```
State: None
Enter a command (type "help" for explanation): authenticate
Enter a name: ree
Type the follwing sentence
automata were principle cosmologists have recently engaged with the capitalized
automata were principle cosmologists have recently engaged with the capitalized
Score: 0.1119392223355144
Authentication successful
█
```