

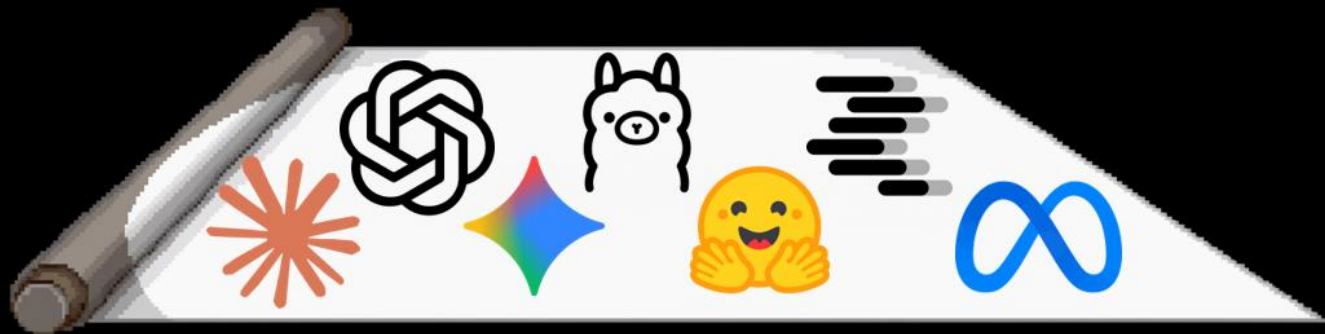
# 小蝦米闖AI江湖

北區ASOC維運中心

李承學

2026/9/2

# 小蝦米闖 AI 江湖



小蝦米初入 AI 江湖  
尋找自己的「AI 野球拳」  
招募群俠尋天書

# 小蝦米初入江湖

---

## 第一性原理 (First Principles)

最初由亞里斯多德提出，近代有馬斯克 (Elon Musk) 與黃仁勳 (Jensen Huang) 在訪問時提到此概念。

## LLM運作原理

LLM簡易運作原理探討。  
(為了方便快速理解，使用簡化後的流程與虛構數字進行舉例說明，僅作為概念示意用途。)

## 可能性與侷限性

理解原理後，LLM的可能性與侷限性是什麼？

# 第一性原理 (First Principles)



亞里斯多德提到：任何系統性研究，若要探究其本質、原因或基本要素，必須從掌握這些最底層的邏輯開始。

只有透徹理解了原理和核心要素，才算真正掌握了知識。



小結：

1. 第一性原理是將複雜問題拆解至最基本的事實與原理。
2. 重新檢視既有的假設與慣例，並從這些基本事實出發。
3. 重新推導適合的解決方案，而不是直接套用過去的經驗或既有答案。

馬斯克 (Elon Musk) 在訪談中提到：有人說歷史上電池成本每千瓦時 (度) 600 美元，未來也不會好多少，但電池是用什麼組成的？現貨市場價格多少？由鈷、鎳、鋁、碳、用於隔離的聚合物跟一個鋼殼。

材料角度分析，金屬交易所購買這些材料，每種材料價格是多少？大概每千瓦時 (度) 80 美元。

想出一些巧妙方法將這些材料組合成電池，就能製造出比任何人想像都要便宜的電池。

黃仁勳 (Jensen Huang) 在訪談中提到：創建一家公司，就像解決所有問題一樣，你必須從基本原理開始。

我們要創造的這台機器是什麼？

它的輸出是什麼？

它的輸入是什麼？

它所處的環境是什麼？

...

你想做什麼？

你想建構什麼？

我認為應該從這個角度來思考。

# LLM運作原理

自然語言輸入

I ate an apple

tokenizer

I/ ate/ an/ apple

embedding

I → [0.56, 1.57, ...]

⋮

apple → [1.07, 3.48, ...]

transformer

轉換

Query

Key

Value

輸出層

推測下一個最高機率的token

RAG

- 問題敘述轉為向量 (embedding)  
I ate a banana → [3.72, 0.37, ...]
- 向量進資料庫比對，找出接近的向量  
[3.72, 0.37, ...] 找到 [3.51, 0.29, ...]
- 接近向量對應的原始資料，當作prompt傳回去LLM  
[3.51, 0.29, ...] 對應原始資料為：香蕉含有維生素A、B1、B2、B6、菸鹼酸、葉酸、C，及礦物質鈣、磷、鐵、鎂、鉀、錳等

# LLM運作原理

I → [0.56, 1.57, ...]

⋮

apple → [1.07, 3.48, ...]

**transformer** 轉換  
Query  
Key  
Value

轉成向量後，會將每個token再轉成Q、K、V三個值

# 會依照模型的原始權重計算出每個token的初始QKV

# e.g.,

Q\_apple = apple向量 \* W<sub>q</sub>

K\_apple = apple向量 \* W<sub>k</sub>

V\_apple = apple向量 \* W<sub>v</sub>

Query: 我要找什麼?

Key: 標籤

Value: 內容

# LLM運作原理

transformer 轉換

Query

Key

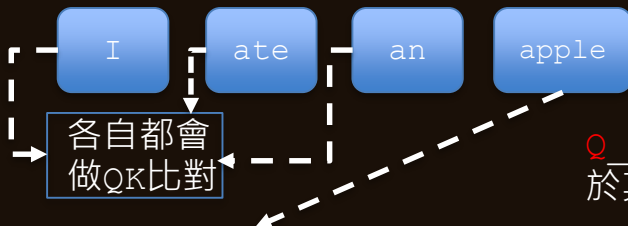
Value

Query: 我要找什麼?

Key: 標籤

Value: 內容

LLM為了理解語意，會透過數學計算找出各個token間的關係  
每個token都會進行QKV計算，以「apple」舉例說明



$Q_{apple}$  比對所有的K，得到 $Q_{apple}$ 對於其他token的關係權重

假設

$Q_{apple} = [1.0, 0.8]$

$K_I = [0.1, 0.0]$

$K_{ate} = [1.0, 0.9]$

$K_{an} = [0.4, 0.2]$

$K_{apple} = [0.8, 0.7]$

$Q_{apple}$  對  $K_I$  的權重分數：

$[1.0, 0.8] * [0.1, 0.0] = 0.1$

$Q_{apple}$  對  $K_{ate}$  的權重分數：

$[1.0, 0.8] * [1.0, 0.9] = 1.72$

$Q_{apple}$  對  $K_{an}$  的權重分數：

$[1.0, 0.8] * [0.4, 0.2] = 0.56$

$Q_{apple}$  對  $K_{apple}$  的權重分數：

$[1.0, 0.8] * [0.8, 0.7] = 1.36$

# LLM運作原理

**transformer** 轉換

Query

Key

Value

Query: 我要找什麼?

Key: 標籤

Value: 內容

Q\_apple 對 K\_I 的權重分數:

$$[1.0, 0.8] * [0.1, 0.0] = 0.1$$

Q\_apple 對 K\_ate 的權重分數:

$$[1.0, 0.8] * [1.0, 0.9] = 1.72$$

Q\_apple 對 K\_an 的權重分數:

$$[1.0, 0.8] * [0.4, 0.2] = 0.56$$

Q\_apple 對 K\_apple 的權重分數:

$$[1.0, 0.8] * [0.8, 0.7] = 1.36$$

省略計算過程

Q\_apple對各個token的權重

| I     | 9%  |
|-------|-----|
| ate   | 45% |
| an    | 15% |
| apple | 31% |

# LLM運作原理

**transformer** 轉換

Query

Key

Value

Query: 我要找什麼?

Key: 標籤

Value: 內容

```
New_apple =  
[0.009, 0.000]  
+  
[0.450, 0.405]  
+  
[0.045, 0.030]  
+  
[0.248, 0.186]
```

重新計算apple的向量，透過各token的v值與剛得到的權重，進行加權計算。

假設各token的v值是：

$V_I = [0.1, 0.0]$

$V_{ate} = [1.0, 0.9]$

$V_{an} = [0.3, 0.2]$

$V_{apple} = [0.8, 0.6]$

```
New_apple =  
0.09 (權重) * [0.1, 0.0] ( $V_I$ ) +  
0.45 (權重) * [1.0, 0.9] ( $V_{ate}$ ) +  
0.15 (權重) * [0.3, 0.2] ( $V_{an}$ ) +  
0.31 (權重) * [0.8, 0.6] ( $V_{apple}$ )
```

```
New_apple =  
[0.752, 0.621]
```

Q\_apple對各個token的權重

| I     | 9%  |
|-------|-----|
| ate   | 45% |
| an    | 15% |
| apple | 31% |

# LLM運作原理

**transformer** 轉換

Query  
Key  
Value

Query: 我要找什麼?

Key: 標籤

Value: 內容

加權計算後得到apple的新向量。

apple = [1.07, 3.48]

New\_apple = [0.752, 0.621]

[0.752, 0.621]

裡面已包含「I ate an apple」的關係向量，LLM透過這個機制了解此時的「apple」應該代表什麼意思。

LLM會知道「apple」代表吃的蘋果，而非蘋果電腦。



# LLM運作原理

輸出層 推測下一個最高機率的token

New\_apple = [0.752, 0.621]

計算機率

token表

|           |     |
|-----------|-----|
| 「. 」      | ... |
| 「pie 」    | ... |
| 「laptop 」 | ... |
| 「and 」    | ... |
| ...       | ... |

假設

「. 」 = [2.0, 2.0]

「pie 」 = [1.0, 2.5]

「laptop 」 = [2.0, -1.0]

「and 」 = [1.5, 1.2]

「. 」  
= [0.752, 0.621] \* [2.0, 2.0]  
= 0.752 \* 2.0 + 0.621 \* 2.0  
= 1.504 + 1.242  
= 2.746

「pie」  
= [0.752, 0.621] \* [1.0, 2.5]  
= 0.752 \* 1.0 + 0.621 \* 2.5  
= 0.752 + 1.5525  
= 2.3045

「laptop」  
= [0.752, 0.621] \* [2.0, -1.0]  
= 0.752 \* 2.0 + 0.621 \* -1.0  
= 1.504 - 0.621  
= 0.883

「and」  
= [0.752, 0.621] \* [1.5, 1.2]  
= 0.752 \* 1.5 + 0.621 \* 1.2  
= 1.128 + 0.7452  
= 1.8732

# LLM運作原理

輸出層 推測下一個最高機率的token

New\_apple = [0.752, 0.621]

計算機率

token表

「.」  
「pie」  
「laptop」  
「and」  
...

歐拉數  $e \approx 2.72$

機率公式為：

$e^{\text{分數}}$

「.」                      「.」  
 $\approx 2.72^{2.746} \approx 15.607 / 34.577$   
 $\approx 15.607 \approx 45.14\%$

「pie」                      「pie」  
 $\approx 2.72^{2.3045} \approx 10.034 / 34.577$   
 $\approx 10.034 \approx 29.02\%$

「laptop」                      「laptop」  
 $\approx 2.72^{0.883} \approx 2.419 / 34.577$   
 $\approx 2.419 \approx 7.00\%$

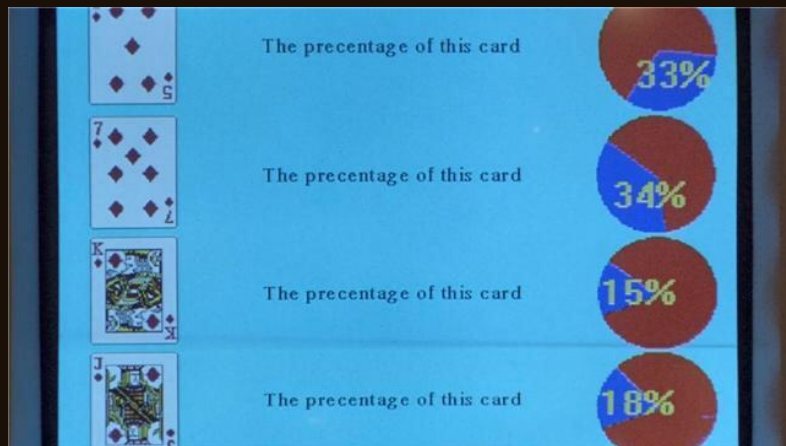
「and」                      「and」  
 $\approx 2.72^{1.8732} \approx 6.517 / 34.577$   
 $\approx 6.517 \approx 18.85\%$

New\_apple與token表的  
各token分數

|        |        |
|--------|--------|
| .      | 2.746  |
| pie    | 2.3045 |
| laptop | 0.883  |
| and    | 1.8732 |

# LLM運作原理

輸出層 推測下一個最高機率的token



每個模型選擇機率的使用方式不太一定



| .      | 45.14% |
|--------|--------|
| pie    | 29.02% |
| laptop | 7.00%  |
| and    | 18.85% |

「I ate an apple」  
最後高機率會接「.」，  
變為「I ate an  
apple .」。

# 可能性與侷限性

預測下一個token，可以算是人工智慧嗎？

我認為某些層面可以算是...

人類思想的本質也是一種預測，

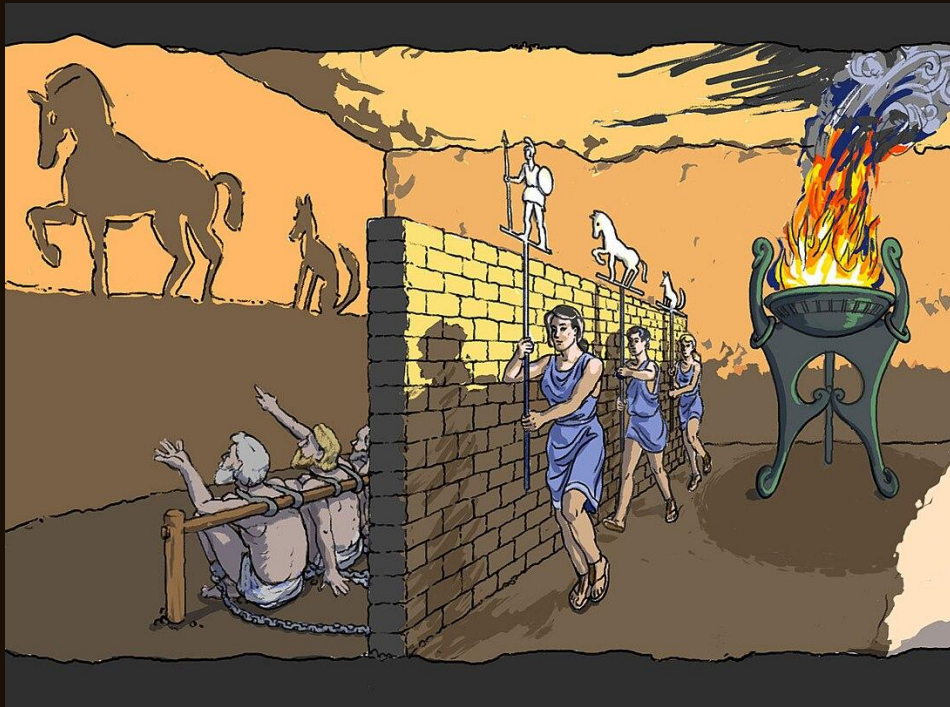
1. 一顆飛來的球，我們可以預測它的落點。
2. 某人講話到一半，我們可以預測他接下來想講什麼。
3. 醫生看病人症狀，依據醫學知識推斷病情。



# 可能性與侷限性

但人類預測真實世界，LLM預測文字的描述。

LLM有點類似柏拉圖的洞穴預言，它看到的是真實世界的影子(人類文字描述的世界)，它透過影子進行預測，但從來沒有走出洞穴。

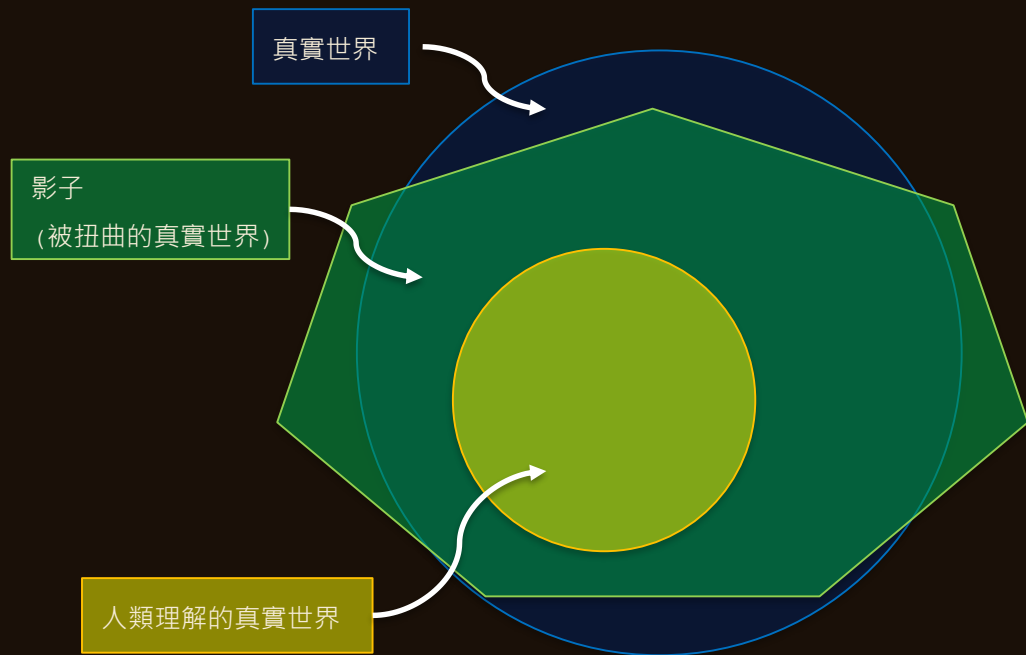


圖片來源：<https://zh.wikipedia.org/zh-tw/%E5%9C%B0%E7%A9%B4%E5%AF%93%E8%A8%80>

# 可能性與侷限性

影子是真實世界的部分投影，受到文字描述的偏差與空洞影響。

1. 文字受到**人類偏見**與**書寫空洞**影響 (書籍遺失或瑣碎小事不記錄)，影子是被扭曲過的真實世界。
2. 但部分是真實的。LLM透過文字 (影子) 這個**有損的中間層**，**間接、部分地重建了世界**。
3. 人類可以理解的真實世界有限，可以透過影子 (LLM) 輔助，讓我們快速學習或提高效率。



# 可能性與侷限性



## ✓ 可能性 (語言的結構性高)

### 1. 摘要長文、會議記錄、翻譯、潤稿、改寫語氣

抓語言重點為LLM的強項。

### 2. 寫程式、debug、解釋程式碼、解讀log、電腦指令

程式語言結構性極高。

### 3. 解釋概念、腦力激盪、草擬企劃、角色扮演

重組相似語意也是LLM的強項，可以用類比方式解釋概念、產生新的觀點、模擬某個腳色等等。

## ✗ 侷限性 (答案在語言之外)

### 1. 預測樂透號碼

純隨機，條件與結果零關聯

### 2. 預測天氣或股票 (未串接工具資料)

LLM可能會提供看起來有道理的答案，但並沒有實際的指標或公式佐證。

### 3. 數學計算 (未串接工具資料)

LLM的算術，不是用演算法 (algorithm)，而是用啟發式 (heuristics) 逼近答案。

# 可能性與侷限性

---

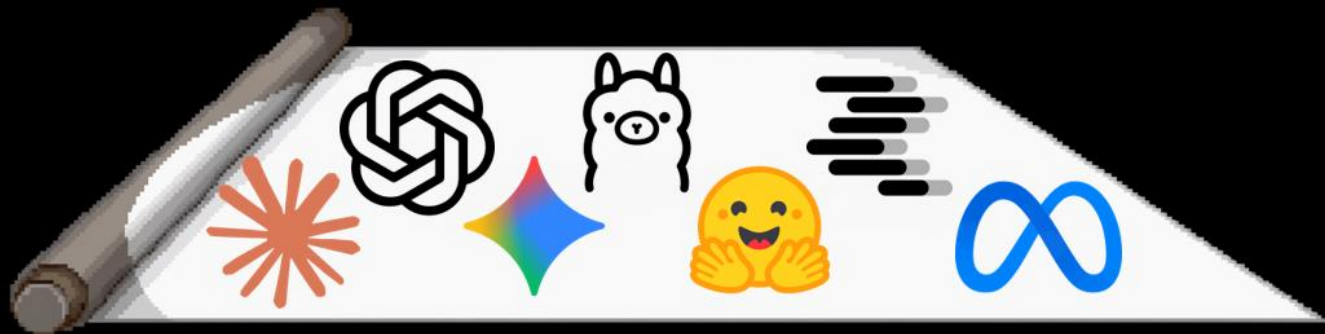
理解LLM的簡易運作原理後，可以更了解LLM的可能性和侷限性。

LLM可以預測未來嗎？ aka分析股市。

您偏向以下哪個說法？

1. 歷史是一個規律循環，分久必合，合久必分，太陽底下無新鮮事。
2. 現實是一個眾多意志的集合體，受到各種條件影響，未來不會出現相同的歷史。

# 小蝦米闖 AI 江湖



小蝦米初入 AI 江湖  
尋找自己的「AI 野球拳」  
招募群俠尋天書

# 尋找自己的AI野球拳

金庸群俠傳最高的血量為999。

練到10級的野球拳傷害有500以上，搭配周伯通的左右互搏（一個招式可以打兩下）...

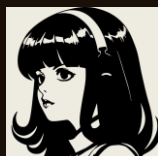
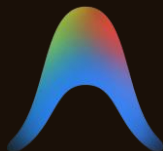


# 尋找自己的AI野球拳

野球拳從第1級到第8級，威力幾乎沒有變化，第9級有提升但不明顯。

但升到第10級後，威力猛然提升，直接稱霸武林。

在AI快速發展的時代，幾乎每幾天就有大量AI相關應用被推出，有時候會覺得沒持續關注的話就要被落後了。



認為可以比照小蝦米學習武功的方式，各路門派的武功有機會一樣學習，  
**但真正的武林絕學為野球拳。**

- 小蝦米若只學各路門派的武功，一樣可以破關，但需要針對敵人的特性，使用特定的武功對付。
- 小蝦米若精通10級野球拳，不論敵人怎麼變，敵人如何多，都可以用野球拳一拳打天下。

我會想將野球拳比喻為心法，只要掌握了心法，不論現代AI時代變化的多快，AI工具有多多，都可以透過自己領悟的心法行走AI江湖。

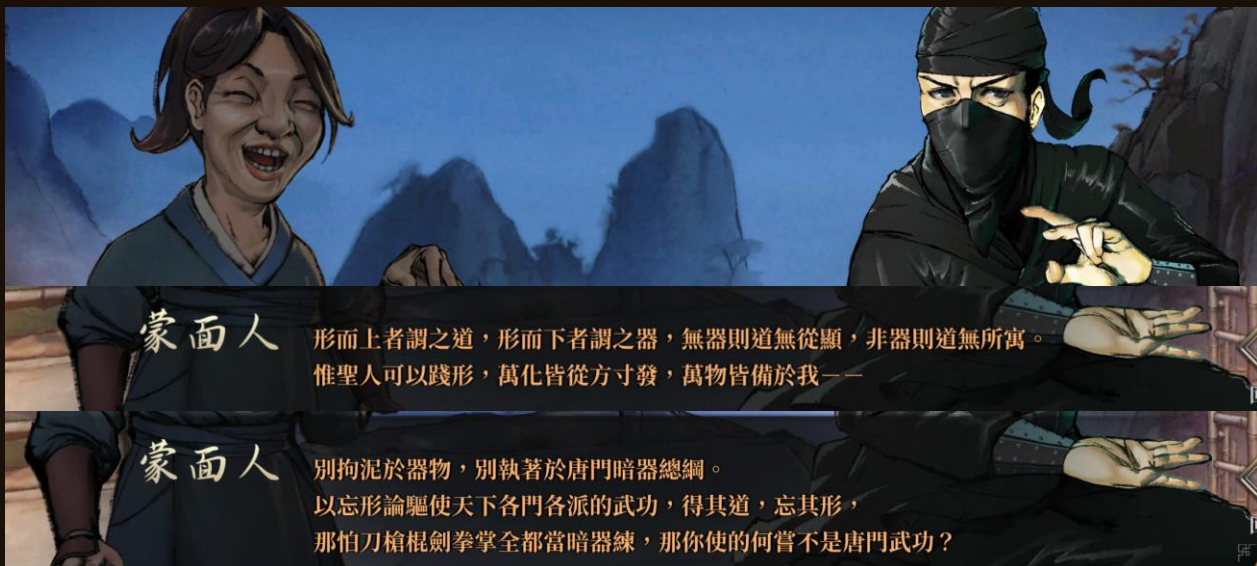
# 尋找自己的AI野球拳

## 唐門暗器總綱-忘形論

### 得其道，忘其形...

形而上者謂之道，形而下者謂之器，無器則道無從顯，非器則道無所寓。

惟聖人可以踐形，萬化皆從方寸發，萬物皆備於我——



蒙面人

形而上者謂之道，形而下者謂之器，無器則道無從顯，非器則道無所寓。  
惟聖人可以踐形，萬化皆從方寸發，萬物皆備於我——

蒙面人

別拘泥於器物，別執著於唐門暗器總綱。  
以忘形論驅使天下各門各派的武功，得其道，忘其形，  
那怕刀槍棍劍拳掌全都當暗器練，那你使的何嘗不是唐門武功？

# 尋找自己的AI野球拳



- 每幾天可能就有新的AI工具，但這個AI工具的功能真的是組織需要的嗎？  
(評估自身需求及是否確實可以解決問題，就算未導入也可了解其背後技術原理)
  - AI只是其中一個手段，解決痛點才是目的，用AI、用程式或用人工都沒有問題，  
考量時間、維護、知識交接成本，選一個最適合的即可。
1. 組織內真正的痛點是什麼？
  2. 找到痛點後再往更深一層思考，出現這個痛點的原因是什麼？
  3. 痛點的原因要如何解決？
  4. 解決方式可以100%解決嗎？若不能100%解決，有什麼其他的輔助機制？

# 尋找自己的AI野球拳

舉例：資安設備使用Snort Rules偵測網路惡意行為，但有時候轄下單位會回報此為正常網路行為，詳細分析後的確應為正常網路服務。

1. **真正的痛點**：Snort Rules有時候會誤判。

2. **痛點的原因**：Snort Rules是寫死的規則，有時合法的網路服務會在意外情況下符合偵測規則。

3. **痛點的原因要如何解決**：Snort Rules可能會有許多複雜的偵測條件，可將被觸發事件的Snort Rules偵測條件及觸發時的網路封包內容，交給LLM進行分析，分析是否有誤判可能。

4. **解決方式可以100%解決嗎？若不能100%解決，有什麼其他的輔助機制？**

無法100%解決，因LLM會依照目前條件(語意) 推論出最接近的答案，遇到複雜的問題容易產生幻覺問題，讓LLM的回答看起來正確及很有道理。

可透過prompt設定，讓LLM要從找到誤判的情況為前提，進行對抗式分析，找出誤判的可能性及原因，以及單一LLM可能會有偏見，可透過多方LLM先獨立分析，後續進行辯論產生共識，最後由人類專家去分析多方LLM產生的共識，分析確實是否有這個誤判可能性。

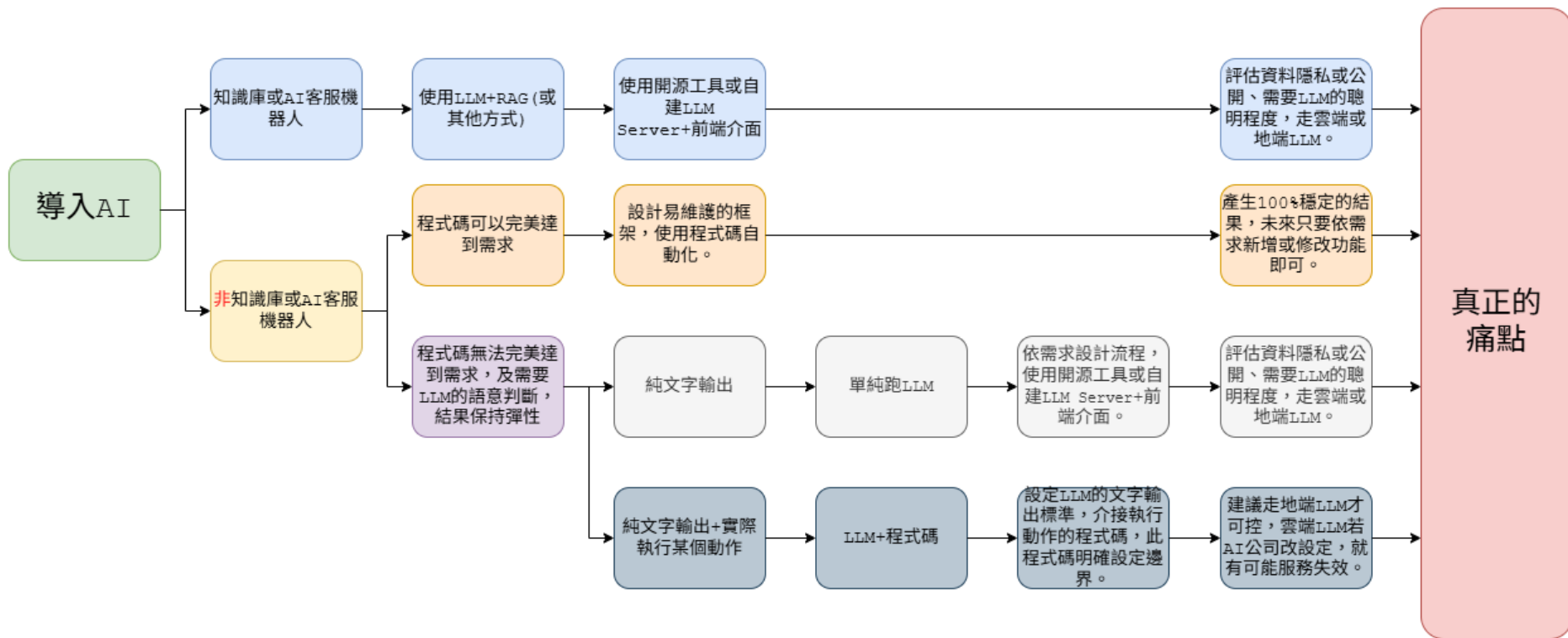
# 尋找自己的AI野球拳

長官期望導入AI，會產生兩個目標，

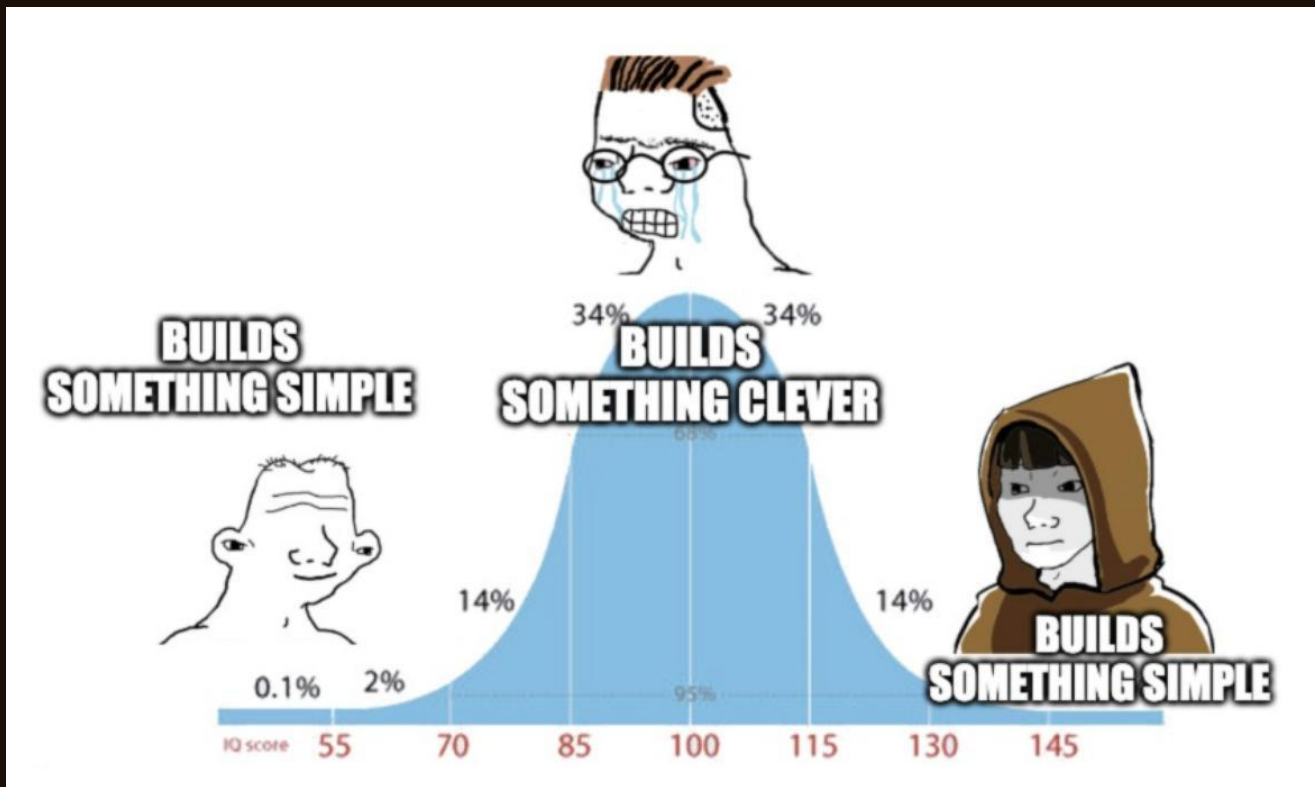
1. 導入AI
2. 組織內有哪些痛點可以透過AI解決？

|        | LLM                                                                                                                                    | 程式碼                                                   |
|--------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| 優點     | 透過語意去判斷，結果保持彈性，<br><b>例如駭客攻擊</b> ，就算駭客繞過規則偵測，LLM只要有能力去判讀LOG，就有機會抓到異常行為而告警。                                                             | 除非有bug，輸出結果穩定，不會有例外狀況。                                |
| 缺點     | 本質為推論，輸出結果可能會有意料之外狀況。                                                                                                                  | 較難處裡變種的惡意行為， <b>例如駭客攻擊</b> ，只要想辦法繞過規則偵測，程式碼(規則)就會失效了。 |
| 結合優點缺點 | LLM搭配程式碼執行實際動作，利用LLM的 <b>語意判斷</b> (若規則直接被繞過， <b>還有一道防線</b> )，執行實際動作時透過特定程式碼執行(限定只能下達設定好的指令)， <b>避免</b> LLM輸出結果有例外狀況造成 <b>系統被誤下指令</b> 。 |                                                       |

# 尋找自己的AI野球拳



# 尋找自己的AI野球拳



# 尋找自己的AI野球拳

---

## 總結

### 1. 得其道，忘其形

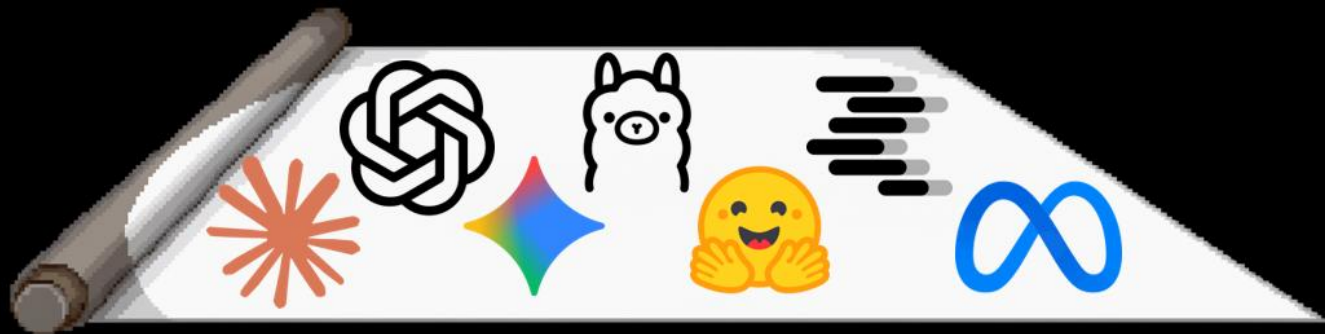
(專注解決痛點，各種AI工具、程式碼、人工方式均可搭配使用)

### 2. 盡量保持簡單架構以方便維護

模組化設計 (Modular Design)

低耦合 (Low Coupling)

# 小蝦米闖 AI 江湖



小蝦米初入 AI 江湖  
尋找自己的「AI 野球拳」  
招募群俠尋天書

# 招募群俠尋天書

---

1. **CSI** (Channel State Information) 偵測空間異動

2. **AI Mini EDR**

# 招募群俠尋天書

---

## CSI (Channel State Information) 偵測空間異動

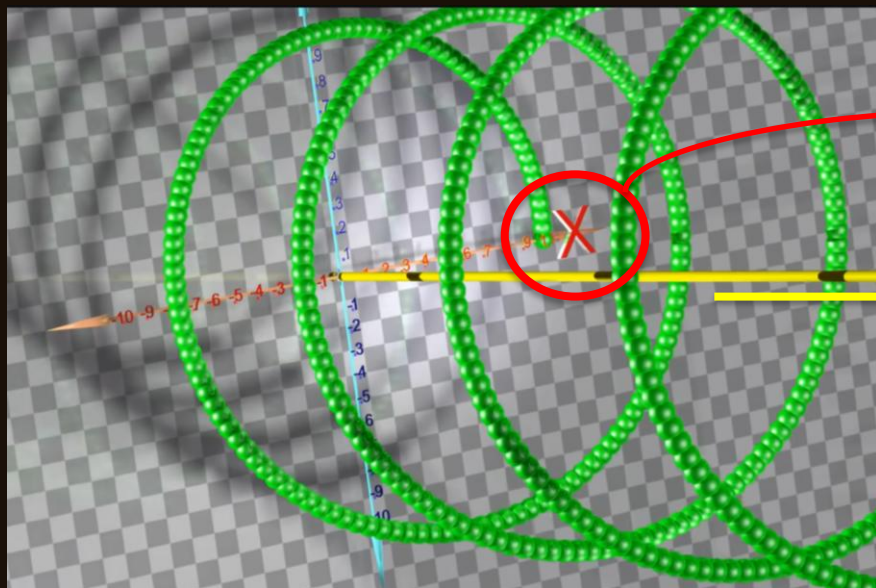
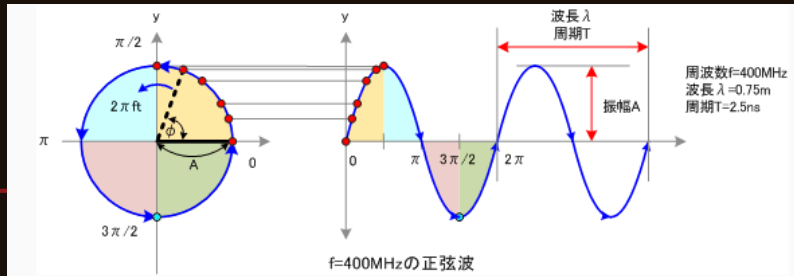
1. 透過WIFI的頻道偵測某個空間，被動式監聽。
2. 利用網卡接收WIFI封包，再從封包抽取CSI分析。
3. CSI原始儲存為複數形式。
4. 複數轉為 Amplitude (震幅) & Phase (相位)。
5. 設定 baseline 當空間基準，以此偵測空間異動。

# 招募群俠尋天書

WIFI發送端送出固定的時域波形 ->

固定的時域波形 -> 環境變化擾動 -> 不定型的時域波形

WIFI接收端收到不定型的時域波形 -> 傅立葉轉換 -> 取得複數 (a, b)



複數 (a, b) 就可代表此頻率

時間線

參考資料:

<https://hackmd.io/@sysprog/fourier-transform>

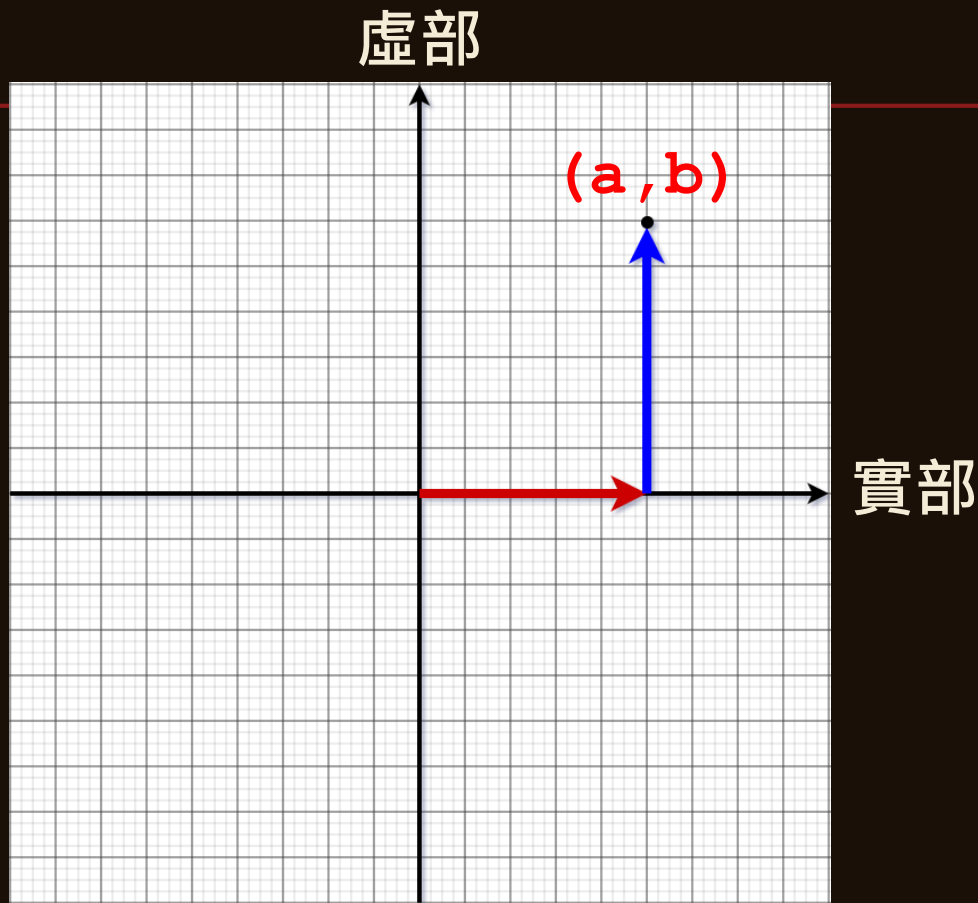
<https://www.youtube.com/watch?v=r18Gi81SkfM>

# 招募群俠尋天書

CSI原始儲存為複數形式

實部

虛部

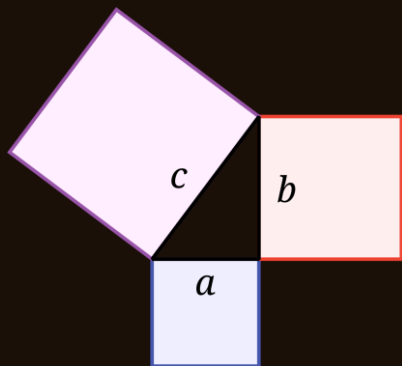


# 招募群俠尋天書

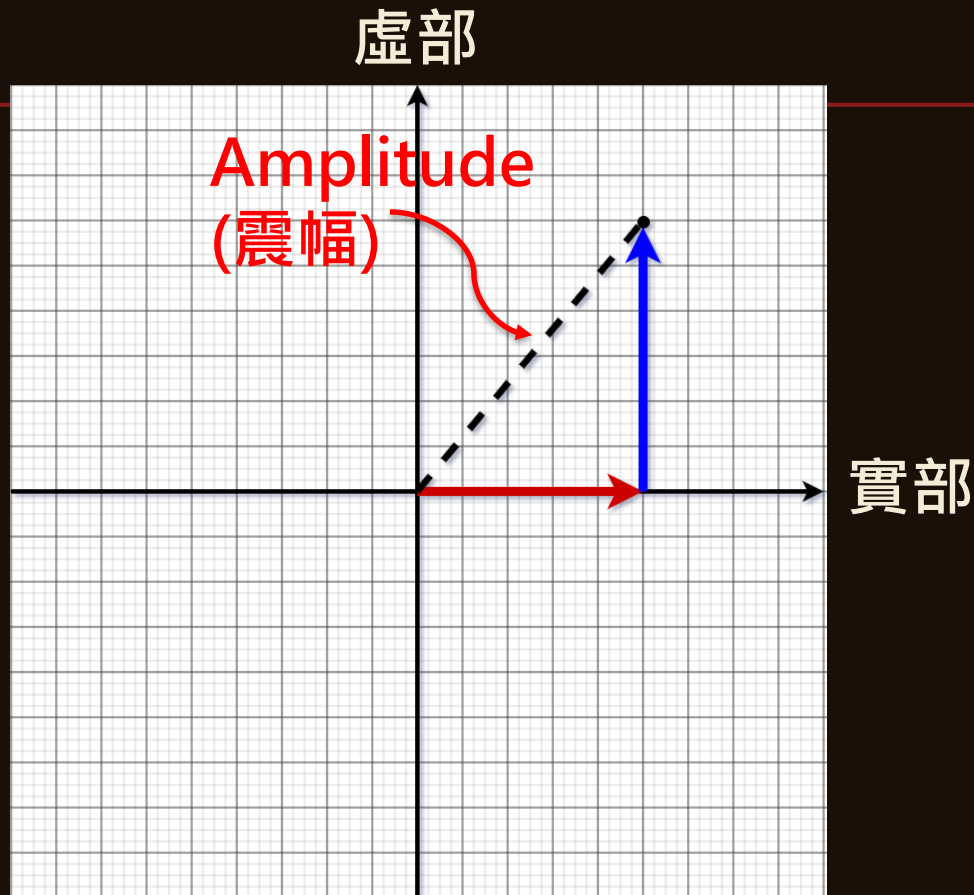
Amplitude (震幅) & Phase (相位)

Amplitude (震幅)

畢氏定理  $a^2 + b^2 = c^2$



$$\text{Amplitude (震幅)} = \sqrt{a^2 + b^2}$$



# 招募群俠尋天書



**Amplitude (震幅) & Phase (相位)**

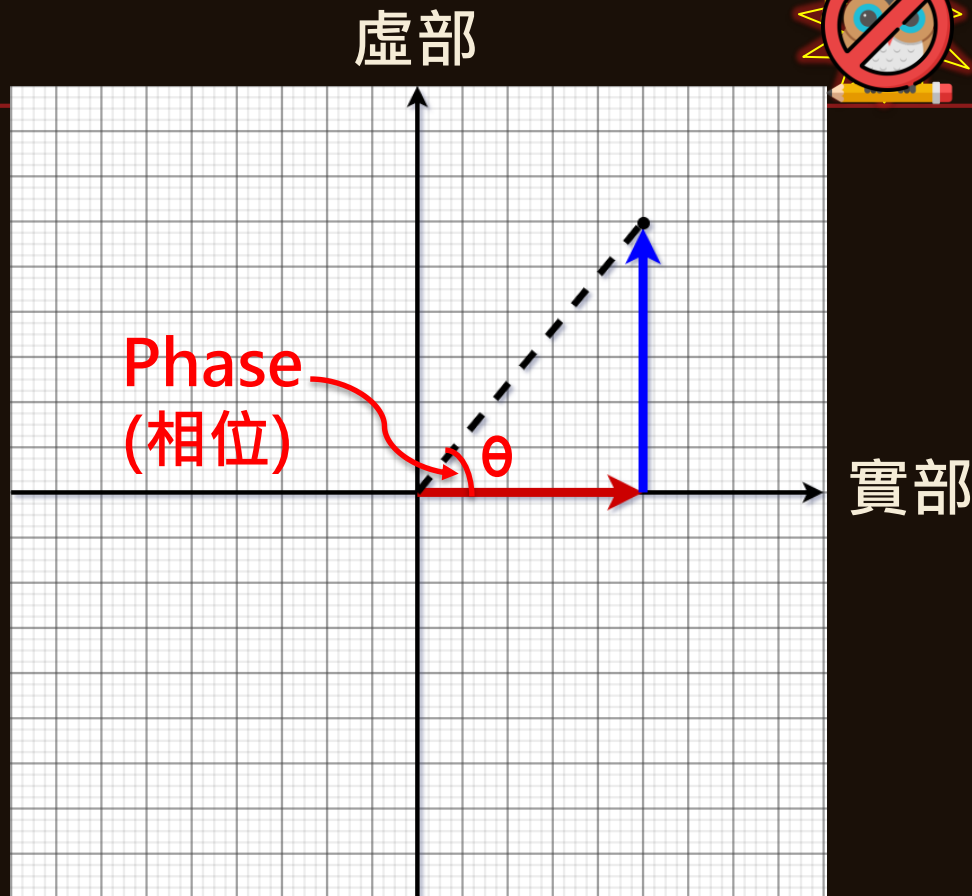
Phase (相位)

三角函數

$\sin\theta$ 、 $\cos\theta$ 、 $\tan\theta$

反三角函數

$\sin^{-1}$ 、 $\cos^{-1}$ 、 $\tan^{-1}$



|              |                       |
|--------------|-----------------------|
| $\sin\theta$ | 對邊 / 斜邊               |
| $\cos\theta$ | 鄰邊 / 斜邊               |
| $\tan\theta$ | 對邊 / 鄰邊               |
| $\sin^{-1}$  | $\sin^{-1}$ (對邊 / 斜邊) |
| $\cos^{-1}$  | $\cos^{-1}$ (鄰邊 / 斜邊) |
| $\tan^{-1}$  | $\tan^{-1}$ (對邊 / 鄰邊) |

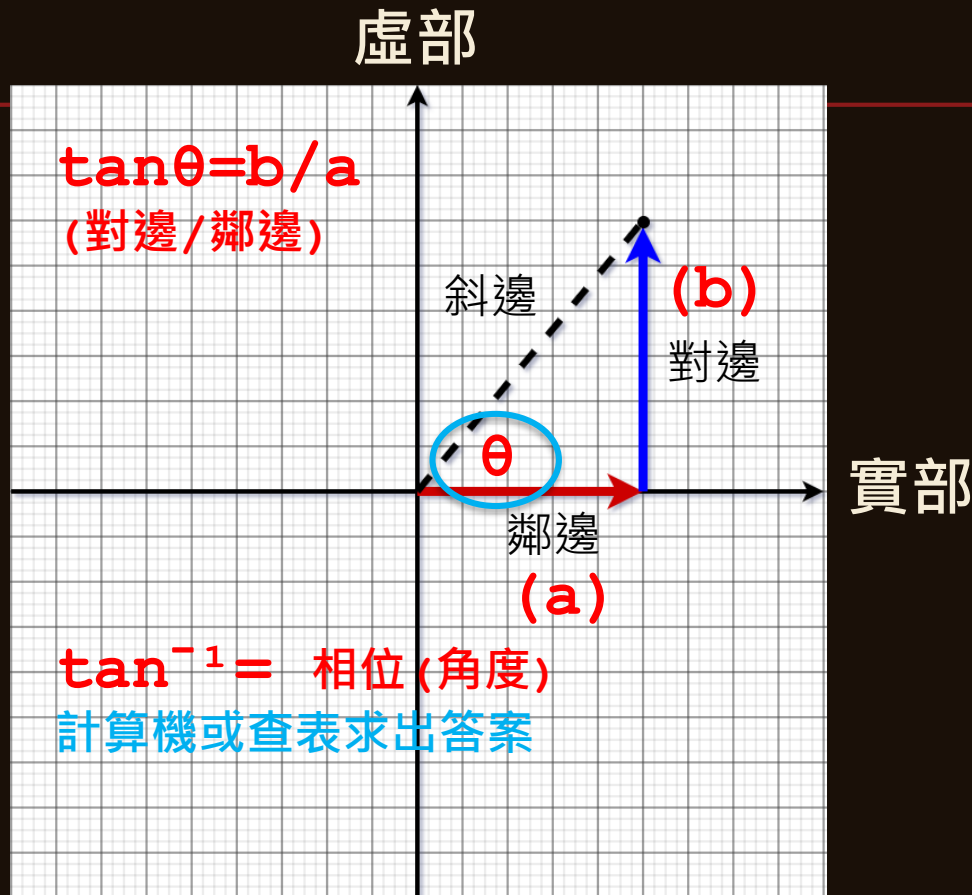
# 招募群俠尋天書

**Amplitude (震幅) & Phase (相位)**

**Phase (相位)**

有直角三角形的三邊長，就可以透過反三角函數推算出相位 (角度)。

|              |                       |
|--------------|-----------------------|
| $\sin\theta$ | 對邊 / 斜邊               |
| $\cos\theta$ | 鄰邊 / 斜邊               |
| $\tan\theta$ | 對邊 / 鄰邊               |
| $\sin^{-1}$  | $\sin^{-1}$ (對邊 / 斜邊) |
| $\cos^{-1}$  | $\cos^{-1}$ (鄰邊 / 斜邊) |
| $\tan^{-1}$  | $\tan^{-1}$ (對邊 / 鄰邊) |



# 招募群俠尋天書



2.4G

1

...

14

5G

36

44

40

48

52

56

60

64

100

104

108

112

116

120

149

153

157

161

165

Subcarrier 頻率

$A_n$

$B_n$

$C_n$

$\vdots$

$X_n$

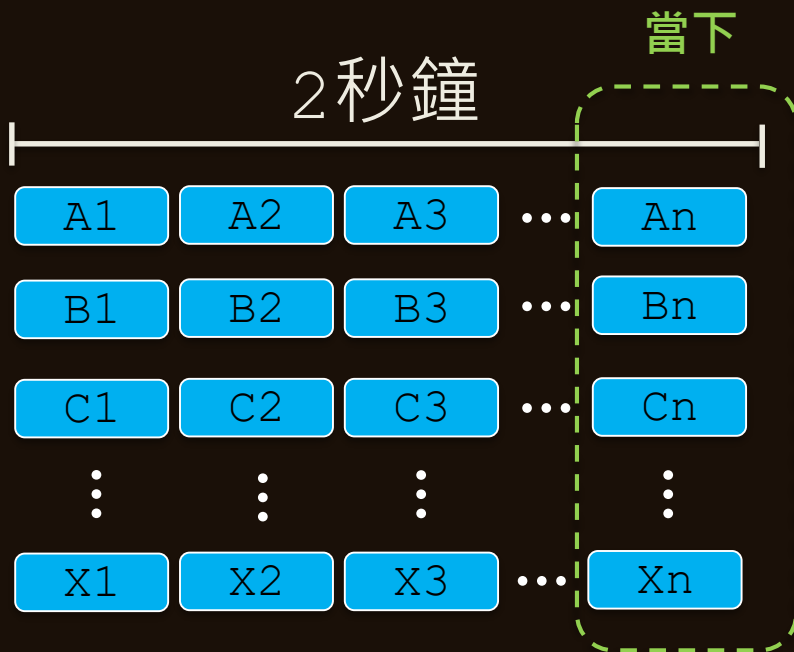
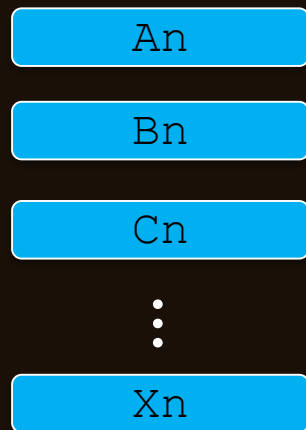
**DFS (Dynamic Frequency Selection)**

- 氣象雷達、軍用雷達共用，WIFI裝置通常會避開此頻道。
- 若指定此頻道，一偵測到氣象、軍用雷達等等，會自動讓出此頻道。

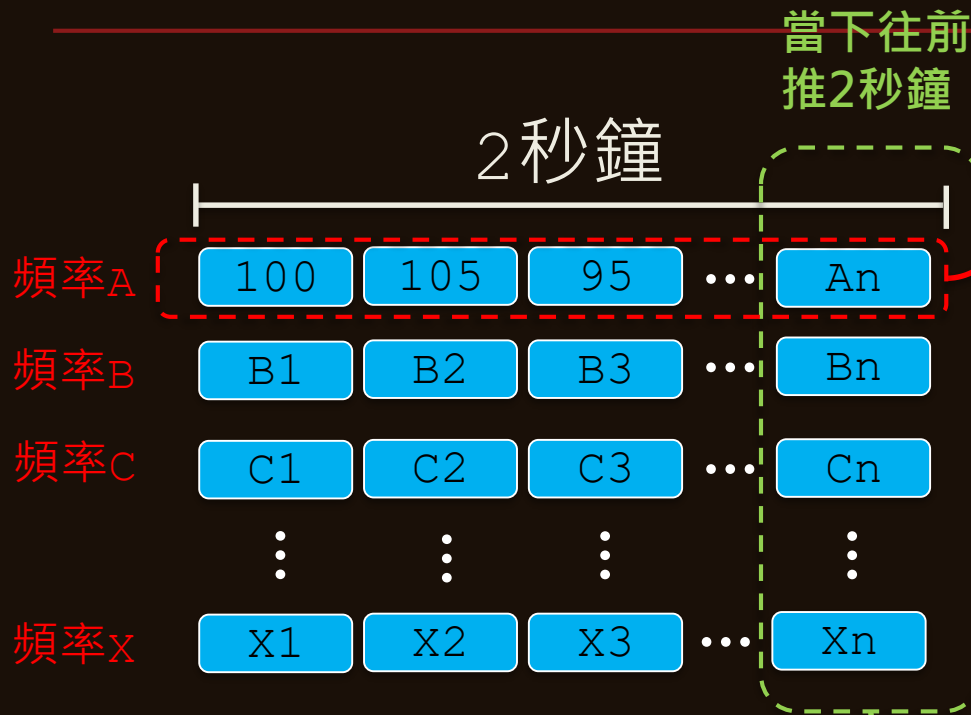
計算分數進行比較  
 $CV = \text{std}/\text{mean}$

# 招募群俠尋天書

Subcarrier 頻率



# 招募群俠尋天書



假設頻率A當下時間，往回推2秒鐘內收到3個震幅，分別為100、105、95。

平均 (mean) 為  $(100+105+95) / 3 = 100$

標準差 (std) 為

$$\sqrt{((100 - 100)^2 + (105 - 100)^2 + (95 - 100)^2) / 3}$$
$$= \sqrt{(50/3)} \approx 4.08$$

變異係數 (CV) =  $4.08 / 100 = 0.0408$

當下的最終動作分數為 (取平均數)：

$(\text{頻率A的CV} + \text{頻率B的CV} + \text{頻率C的CV} + \dots + \text{頻率X的CV}) / \text{頻率數量}$

# 招募群俠尋天書

- 計算baseline做比較，以此偵測是否空間有異常。
- 需要錄一段空房間的CSI，以此當基準做比較。

空房間CSI的每個當下動作分數

透過每個當下的動作分數，計算出baseline，

0.22170

0.26995

0.31820

...

n

baseline計算公式：平均(mean) + k \* 標準差(std)

平均(mean) 為  $(0.22170 + 0.26995 + 0.31820) / 3 = 0.26995$

標準差(std) 為

$$\sqrt{((0.22170 - 0.26995)^2 + (0.26995 - 0.26995)^2 + (0.31820 - 0.26995)^2) / 3}$$

$$= \sqrt{(0.0046562/3)} = 0.03940$$

$$\text{baseline} = 0.26995 + k * 0.03940$$

# 招募群俠尋天書

baseline

當下動作分數

**RSSI :**  
有封包在傳送的波動

**Noise :**  
未有封包傳送的波動

最新分數

0.3612

門檻值

0.3882

RSSI

-59 dBm

Noise floor

-62 dBm

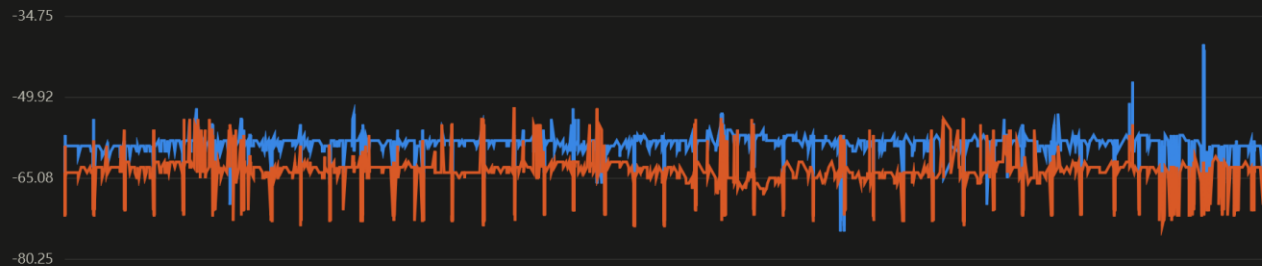
累積收到 frame 數

193955

動作分數 (圖表只顯示最近 1800 筆 · 總數已累積 193955 筆)



RSSI / Noise floor (dBm)



# 招募群俠尋天書

空間有明顯變化時，  
動作分數明顯超出  
baseline。

揮動手臂，嘗試讓空間  
有較明顯的變化。

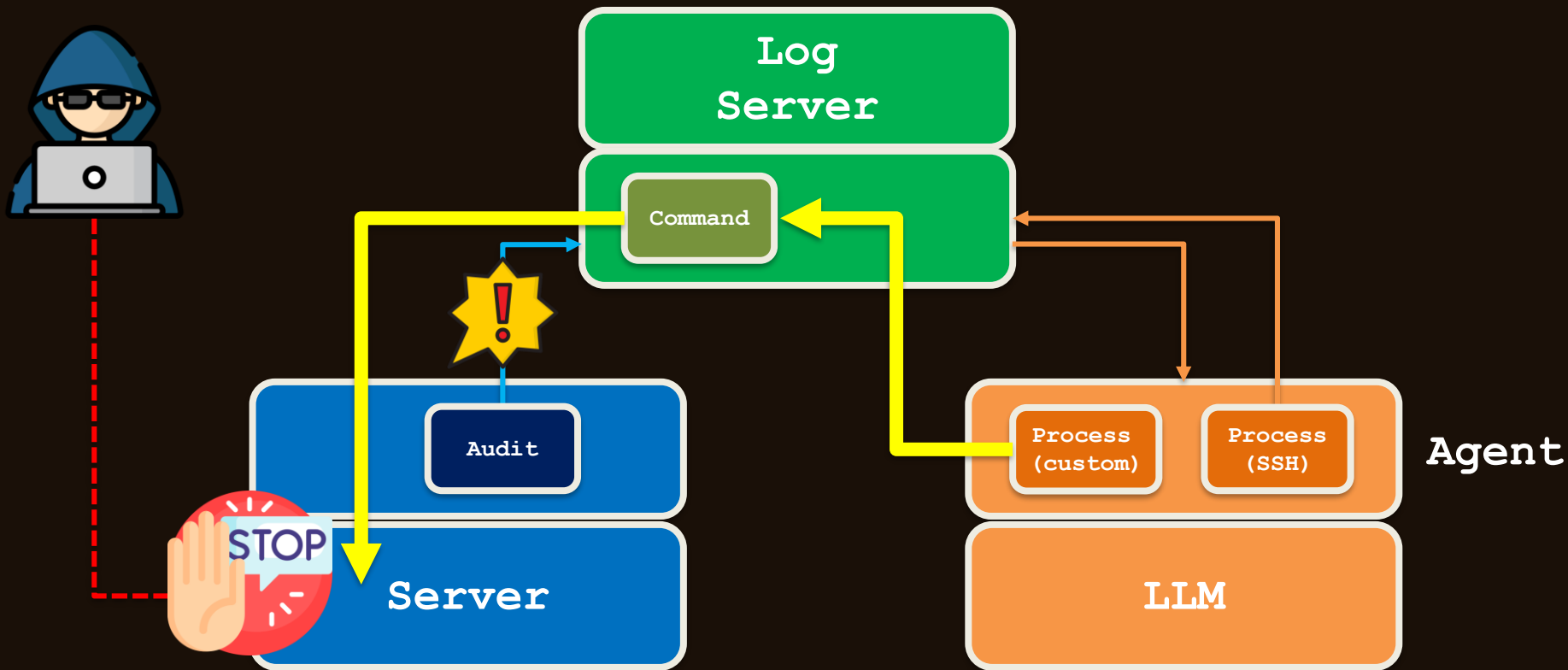
# 招募群俠尋天書

---

## AI Mini EDR

1. Audit服務收集核心行為，儲存觸發規則的LOG。
2. AI透過語意分析相關LOG，判斷是否為惡意行為。
3. 希望AI要能先初步處理，而不是單純通知。
4. 若AI誤判或幻覺而執行動作怎麼辦？
5. 設定邊界，只能處理白名單內的動作。

# 招募群俠尋天書



# 招募群俠尋天書

## Setp1.

### CVE-2026-46333攻擊測試

記憶體管理資訊 (mm) 已釋放，但檔案描述符 (FD) 尚未完全關閉的狀態，導致權限檢查可能被繞過。

## Setp2.

### LLM判斷LOG及執行動作

觸發Audit規則產生LOG，LLM讀取這些LOG，判斷為惡意行為，進行指定動作。

## Step3.

### 攻擊者連線被關閉

(測試的防禦動作)

```
*** System restart required ***
Last login: Wed Jun 17 05:50:51 2026 from 192.168.193.1
ubuntu-2@ubuntu-2:~$ cd ssh-keysign-pwn/
ubuntu-2@ubuntu-2:~/ssh-keysign-pwn$ date
Wed Jun 17 05:55:37 AM UTC 2026
```

```
ubuntu-2@ubuntu-2:~/ssh-keysign-pwn$ ./test20260617
uid=1000 target=/usr/lib/openssh/ssh-keysign
fd 4 -> /etc/ssh/ssh_host_ecdsa_key (round=0 try=445)
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktZjEAAAAABG5vbmUAAAAAEbm9uZQAAAAAAAAABAAAAABNLY2RzYS
1zaG8EYLW5pc3RwMjU2AAACG5pc3RwMjU2AAACQQRrKwaSB+2iPDJPExac3qrLKY9F+/d
q31NIF79U6rHsExkgEYcrJKz6MoIKQNTAmW1lGu+gnpkxsraytLBjhcaAAAG02UA5jt1
A0AAAAE2VjZHNhLXNoYTIibmLzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGsrBpIH7aIMK8
TFpzeqsuRj0K792rFu0gXv1TqseWT6SARhyskrPoyggpA1MCZbWUa76CemT6ytrK0sG0Fx
oAAAAhAlwvrclK+eViND8iaFIHoPyFJmW6AU1xIK9gTy00TAQAAADXJvb3R3ADWJ1bnR1
LTIBAg==
-----END OPENSSH PRIVATE KEY-----
```

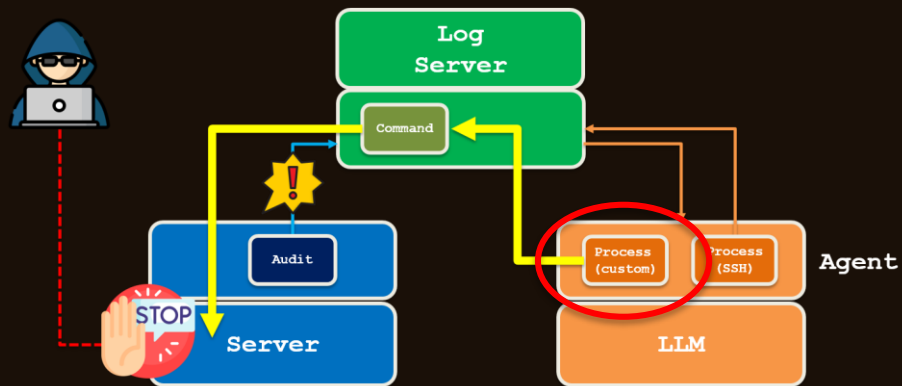
```
ubuntu-2@ubuntu-2:~/ssh-keysign-pwn$ Connection to 192.168.193.130 closed by remote
host.
Connection to 192.168.193.130 closed.
hsueh@hsuehdeMacBook-Air ~ %
```

```
*** 規則命中: privesc_setuid *** [EOE] ssh-keysign (exe:/usr/lib/openssh/ssh-keysign)
*** 規則命中: fdtheft *** [EOE] test20260617 (exe:/home/ubuntu-2/ssh-keysign-pwn/test20260617)
*** 規則命中: fdtheft *** [EOE] test20260617 (exe:/home/ubuntu-2/ssh-keysign-pwn/test20260617)
---
[2026-06-17 13:56:22,152] INFO HTTP Request: POST http://localhost:11434/api/chat "H
TTP/1.1 200 OK"
[2026-06-17 13:56:22,152] WARNING [ALERT] ubuntu-2 | 9 | 偵測到異常執行檔執行權限
提升與檔案描述符竊取行為 | 使用者登入後立即執行位於非標準路徑的惡意程式，且該程式觸
發了 privesc_setuid 與 fdtheft 規則，顯示正在嘗試利用 ssh-keysign 進行權限提升與檔案
描述符劫持攻擊。
[分析] ubuntu-2 -> ALERT: 9 | 偵測到異常執行檔執行權限提升與檔案描述符竊取行為 | 使用
者登入後立即執行位於非標準路徑的惡意程式，且該程式觸發了 privesc_setuid 與 fdtheft
規則，顯示正在嘗試利用 ssh-keysign 進行權限提升與檔案描述符劫持攻擊。
[2026-06-17 13:56:22,152] WARNING [RESPOND] notify | ubuntu-2 | 命中規則: fdtheft, n
et_connect, privesc_setuid, privesc_suid
[2026-06-17 13:56:22,152] WARNING [RESPOND] kill_session | ubuntu-2 | ses=1307 auid=
1000
[2026-06-17 13:56:22,163] INFO Connected (version 2.0, client OpenSSH_9.6p1)
[2026-06-17 13:56:22,420] INFO Authentication (publickey) successful!
[2026-06-17 13:56:22,471] INFO Connected (version 2.0, client OpenSSH_9.6p1)
```

攻擊者

LLM

# 招募群俠尋天書

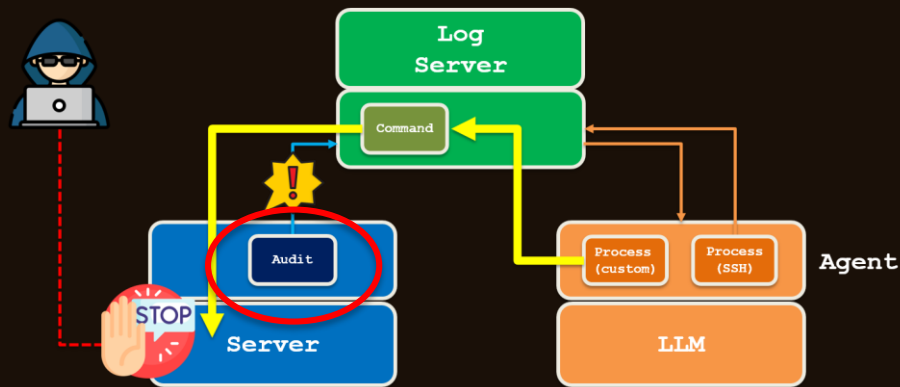


LLM判斷惡意行為，實際執行動作時：

1. LLM只能執行設定好邊界的程式。
2. 任何跳脫這個程式的指令都會**無效**。



# 招募群俠尋天書



Audit規則設定：

1. Audit可偵測Linux核心行為。
2. 規則設定相當有難度。
3. 有太多合法行為會觸發你預期的惡意行為。
4. 需要排除某些程式及要有一段觀察期。

- sshd使用許多函式，容易誤判。
- sudo會使用許多提權函式，容易被誤判惡意提權。
- git容易被誤判勒索行為。
- 但觸發的LOG給LLM看，就有高機率判定為正常行為。
- 若想要減少誤判的LOG，需要手動排除某些程式或行為 (Linux核心專家+AI輔助)。

# 小蝦米闖AI江湖

---

理解LLM可能性與侷限性，發揮LLM的優勢。

得其道，忘其形

結合興趣或工作需求開發測試，未使用也可學到知識及經驗。

謝謝聆聽