釣魚網站整合並實踐於社群軟體

Integrate phishing site and execute on the browser and social media

指導教授: 蔡孟勳老師

組員: 林鴻逸

目錄

Phishing check site

Try the new URL Reputation API by APIVoid.

Need to scan an IP address? Try IPVoid

Enter website or URL here | Scan Website

Data submitted here is shared with security companies (terms of use).

- 背後有龐大的資料庫
- 隨時線上查詢

- 不定時更新
- 有些太老舊
- 不同國家資料不同

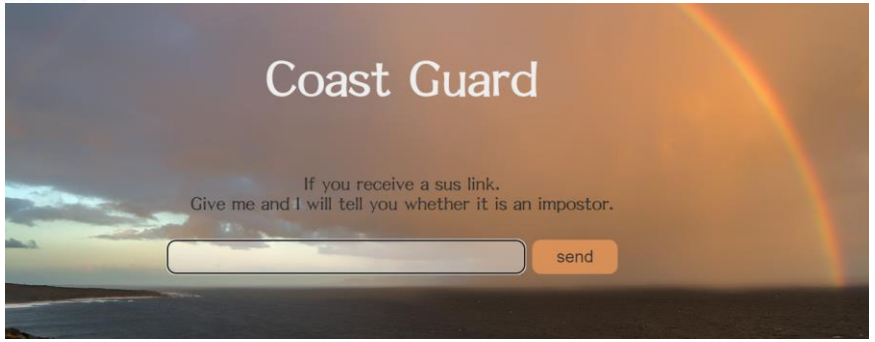
目標 & 動機

- 解決不同國家的問題
- 包攬舊的又承接新的
- 銜接到社群軟體
- 解決要連到那些網站的麻煩

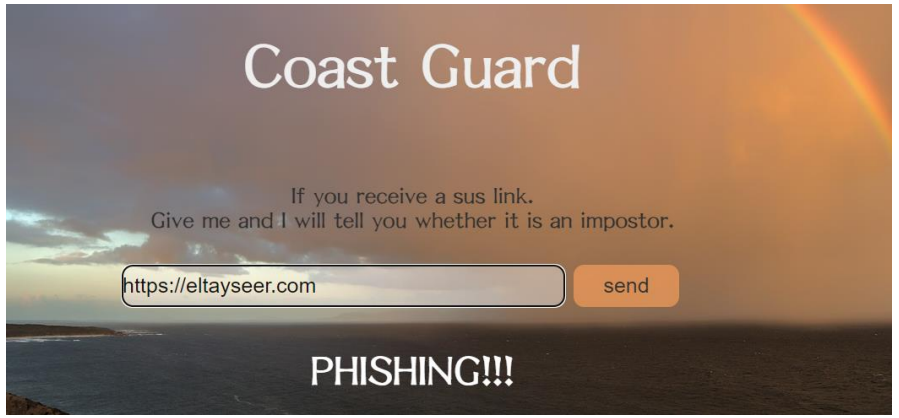
使用工具

- Nodejs： Central System 架設
- MySQL： Danger link 儲存
- Javascript： 撰寫 API 與串聯各架構
- LINE Developer： 建立 LINE Bot 使用
- Discord API： 建立 Discord Bot 使用
- Google Chrome Extension： 建立 Chrome 擴充程式

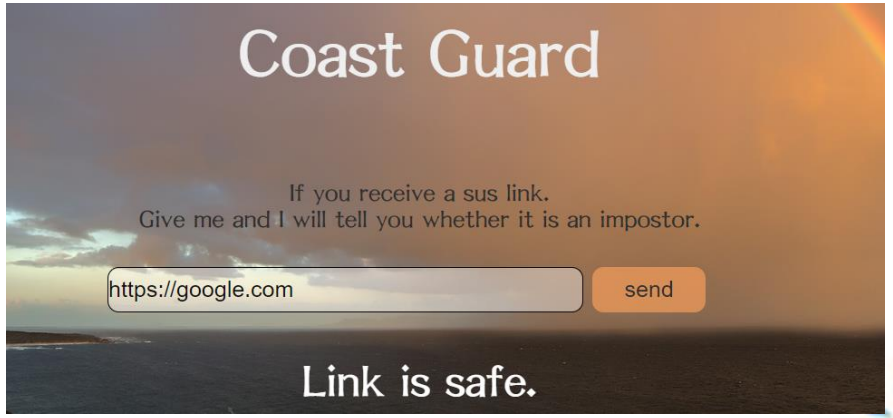
Web Portal



danger

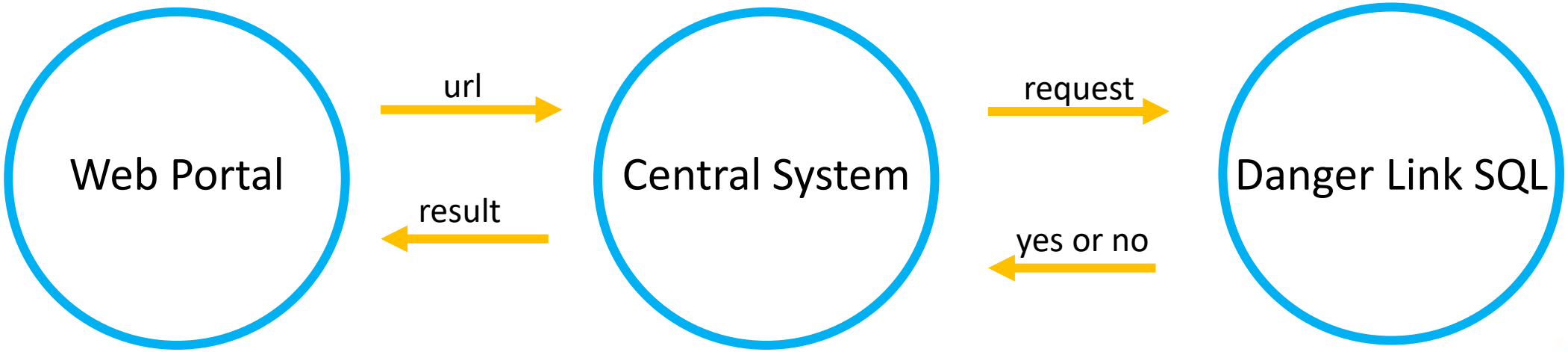


safe

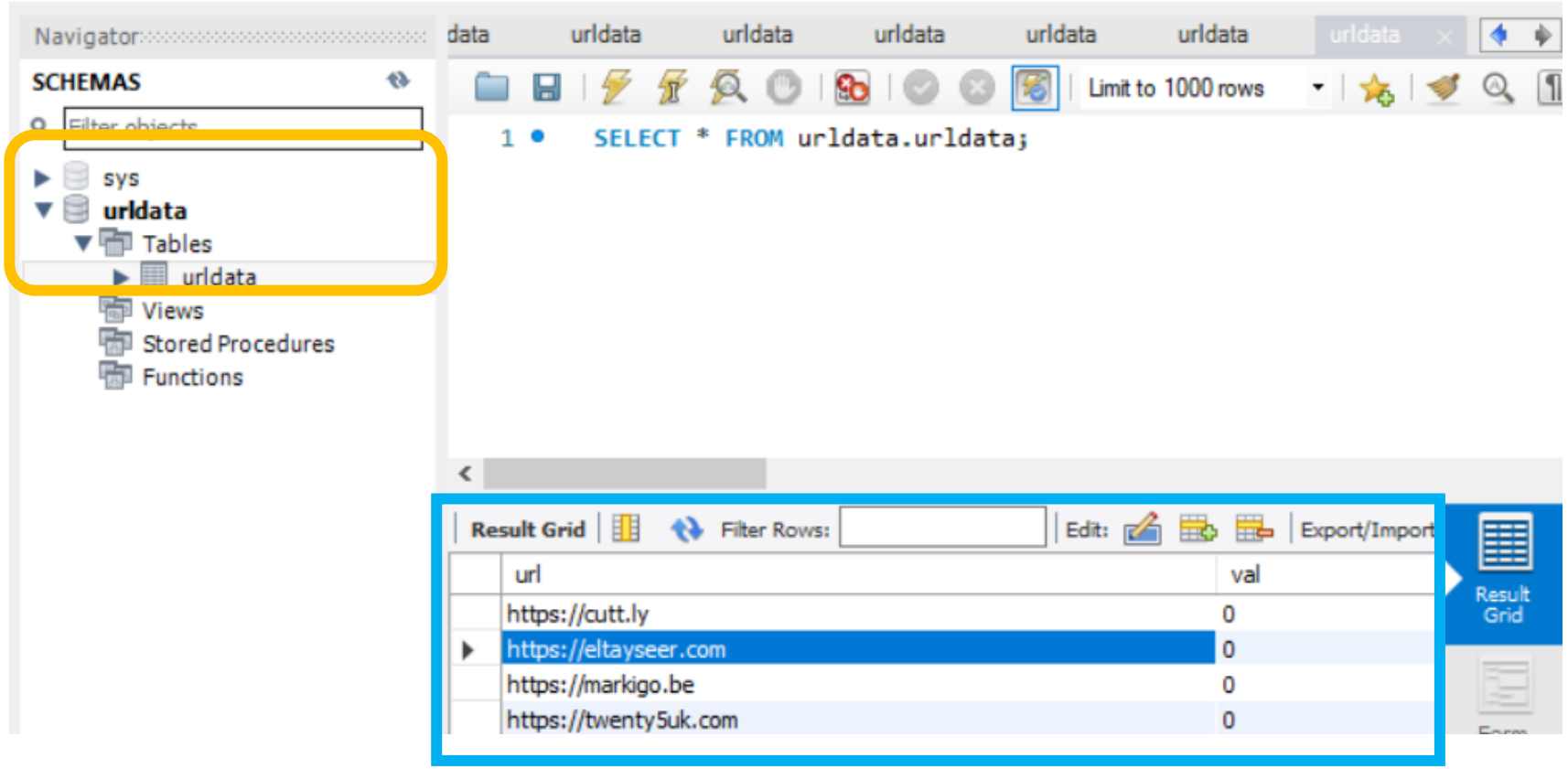


架設於實驗室的 30450 port
給使用者直接進行檢測

Web Portal


Web Portal
url
result
Central System
request
yes or no
Danger Link SQL

Danger Link data SQL

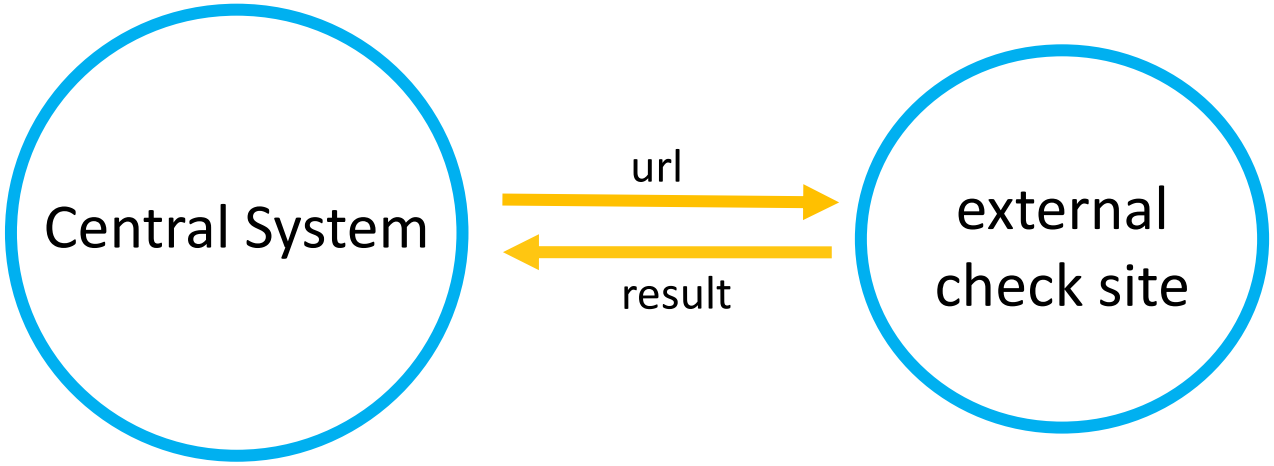


val = 0 => 危險 val = 1 => 有風險

Danger Link data SQL

使用外部網站的資料庫

去串聯各個外部網站讓它們判斷網址的安全



Central system

接收 url 並且作處理

https://xxx.ooo.com/aaa/bbb.php

https:// xxx.ooo.com /....

https://xxx.ooo.com

or http://xxx.ooo.com

Central system

voidurl.com 的檢測結果

Report Summary

Website Address	Google.com
Last Analysis	7 hours ago | Rescan
Detections Counts	0/44
Domain Registration	1997-09-15 | 25 years ago

Central system

根據不同網站的 fetch 結果判斷是否 phishing

- 使用 fetch 工具傳送 https request 給外部網站並接收 response (html 格式)
- 分析 html 並根據關鍵字做切割分段，如 voidurl 的 “ Detection ”
- 取出需要的數據以此獲得這個網頁的判斷結果

```
<tr><td width="260"><span class="font-bold">Website Address</span></td><td><strong>Google.com</strong></td></tr>
<tr><td><span class="font-bold">Last Analysis</span></td><td>7 hours ago  |  <i class="fa fa-refresh" aria-hidden="true"></i> <a hre
<tr><td><span class="font-bold">Detections Counts</span></td><td><span class="label label-success">0/44</span></td></tr>
```

Central system

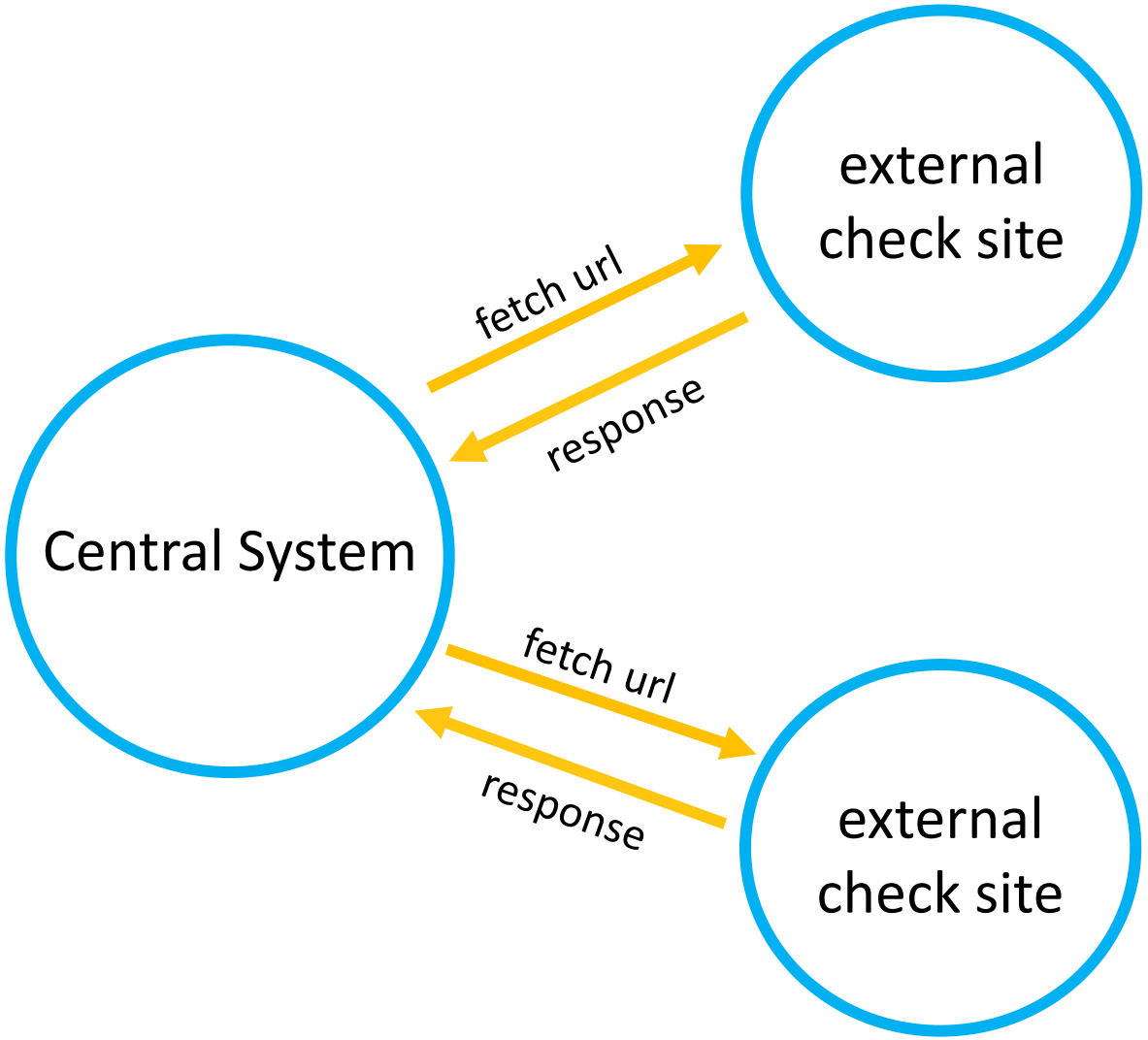
```
const fetch = require("node-fetch");
fetch( "目標網站的 url" , {
    "header": {
        "..." : "..."
    },
    "body": "",
    "method": "post"
})
.then(res =>{
    return res.text();// res.text() is response
})
```

回傳一段 html code => 用關鍵字去切割跟分析

需要處理的 header 跟 body 會不一樣

每個網站也要用不同方式去設計切割方法

Central system



LINE Bot

設計一個收發訊息的 LINE Bot

把收到的網址傳到 Central System

Central System 判斷完傳給 bot

LINE Bot 把結果顯示給使用者

LINE Bot

Webhook settings

Webhook URL ⓘ https://coastguard.imslab.org/linewebhook

Verify Edit

nginx 把 https://coastguard.imslab.org 轉到 LINE Bot 的接收 port

LINE Bot 再把網址丟給 Central System

LINE Bot

加入好友即可送訊息

傳送後約一到兩秒就會收到回覆

Discord Bot

與 LINE類似的原理

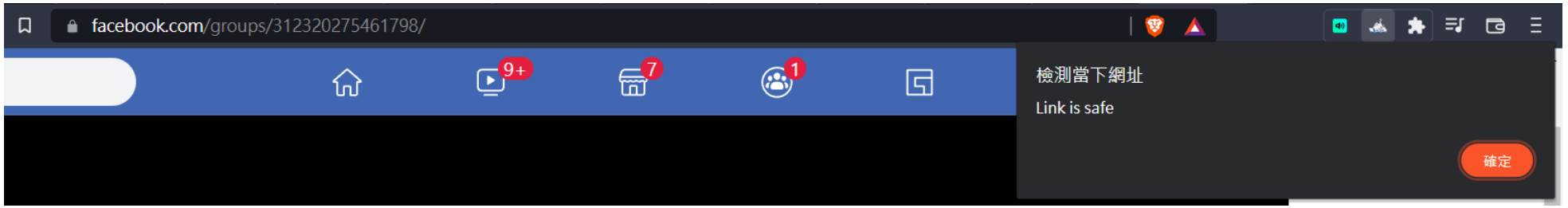
一樣是用 Discord Bot 接收訊息

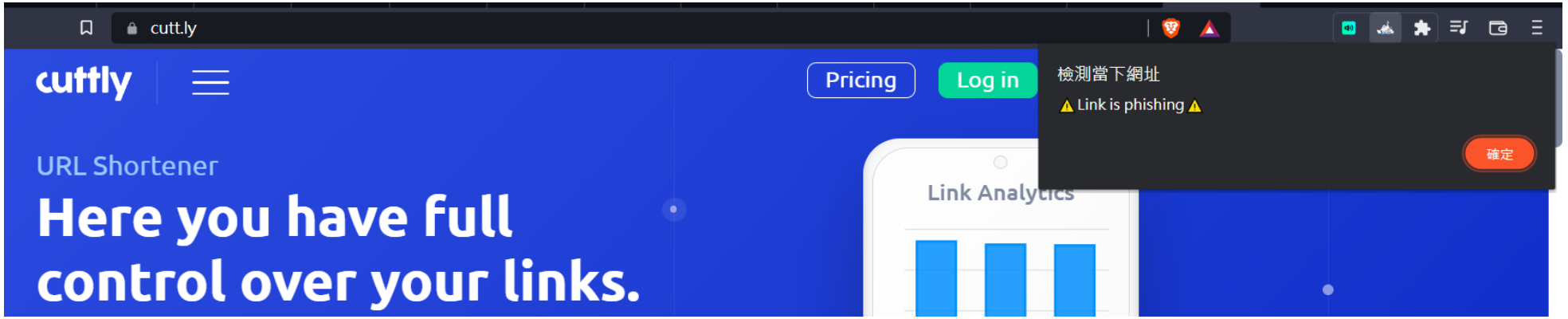
傳給 Central System 判斷

再把結果顯示給使用者

Discord 在開發上比較方便，只需要 token 即可使用
client.login(token) 就可以讓 Bot 上線運行

Google Extension





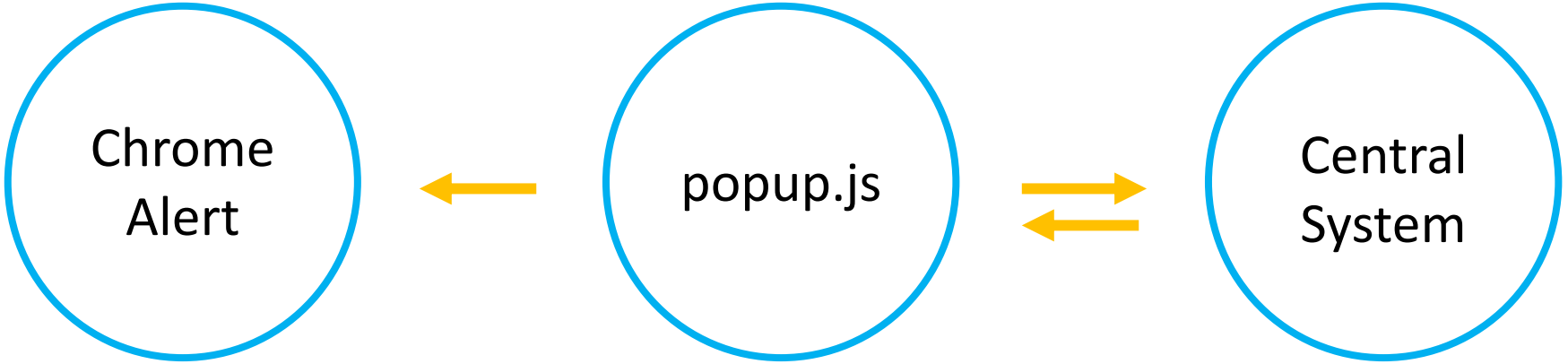
擷取當下的網址去做判斷

Google Extension

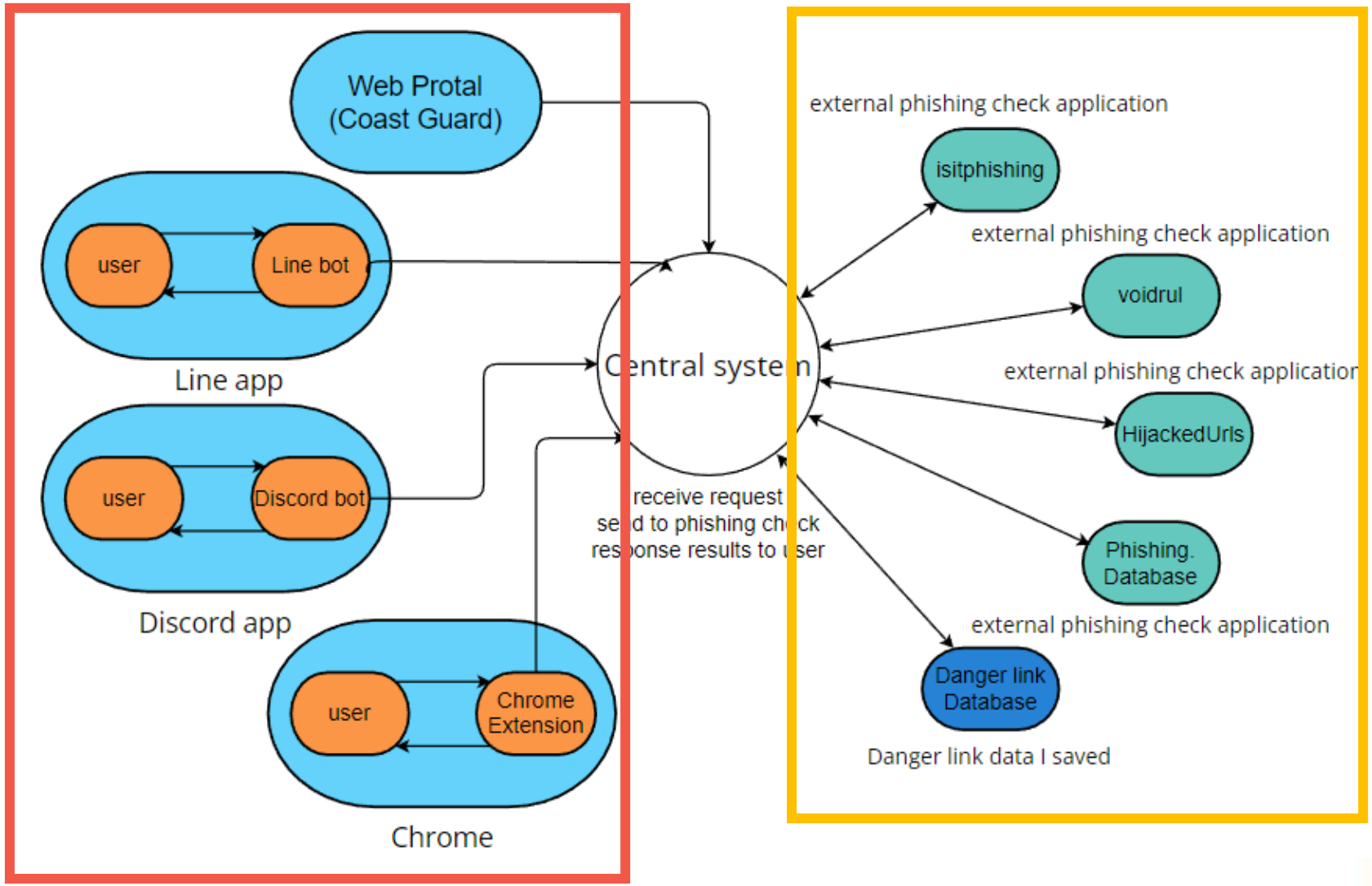
Google Extension

- manifest.json
- popup.js
- icon 

Google Extension



結論


Web Protal
(Coast Guard)
user
Line bot
Line app
user
Discord bot
Discord app
user
Chrome
Extension
Chrome
Central syste n
receive request
se d to phishing ch ck
res onse results to u ser
external phishing check application
isitphishing
external phishing check application
voidrul
external phishing check application
HijackedUrls
Phishing.
Database
external phishing check application
Danger link
Database
Danger link data I saved