

L7 Reverse Proxy Web Server @ 內網

臺灣大學計資中心

網路組

游子興

大綱

- * L7 Reverse Proxy HTTPS to HTTP
 - * 導入 HTTPS 加密協定: 使用 Let's Encrypt 免費憑證
 - * 導入 Zero Trust 架構
 - * 導入 HTTP Basic Authentication
 - * ngx_http_auth_basic_module
- * Transparent Proxy Mode

L7 Reverse Proxy HTTPs to HTTP

快速導入 HTTPS 協定
使用 Let's Encrypt 免費憑證 (Certbot)

L7 Reverse Proxy

HTTPs to HTTP

- * 應用情境

- * 原 Web Server 無法導入 HTTPS 協定
 - * 例如. 廠商硬體設備、OS/Web Server 過舊或不支援 HTTPs 協定

- * 優點

- * 快速導入 HTTPS + Let's Encrypt 免費憑證 (Certbot)

L7 Reverse Proxy

HTTPS to HTTP



使用者連線:

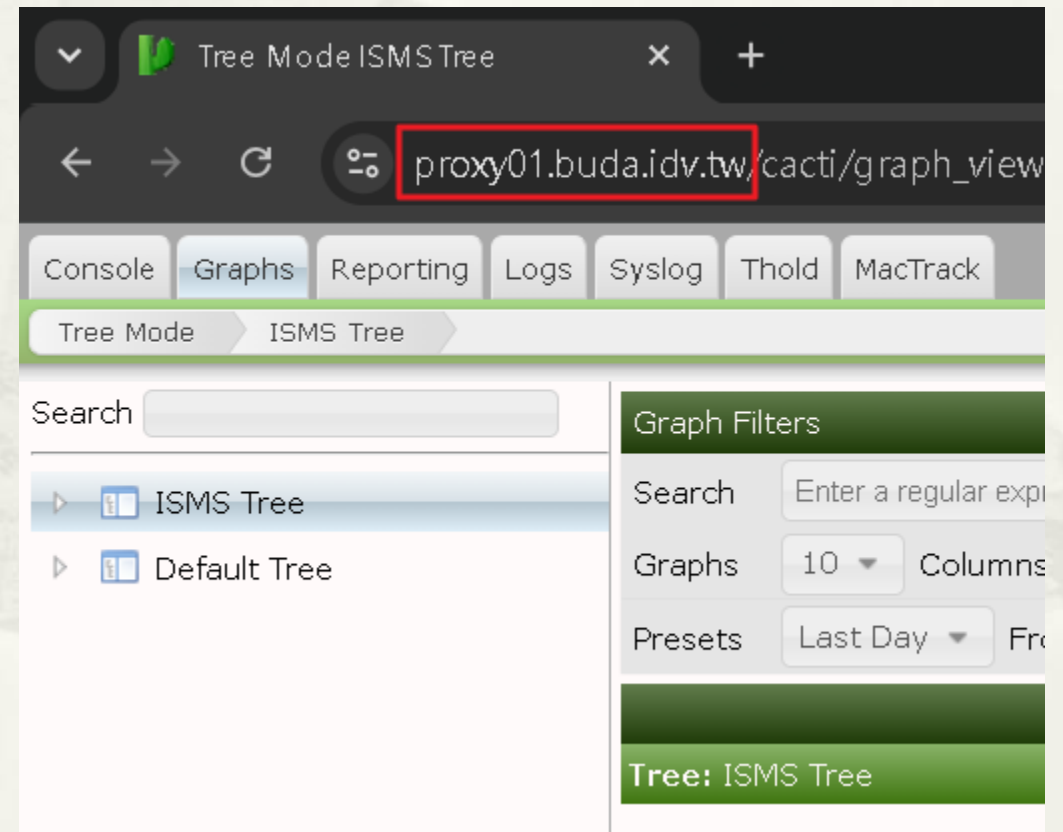
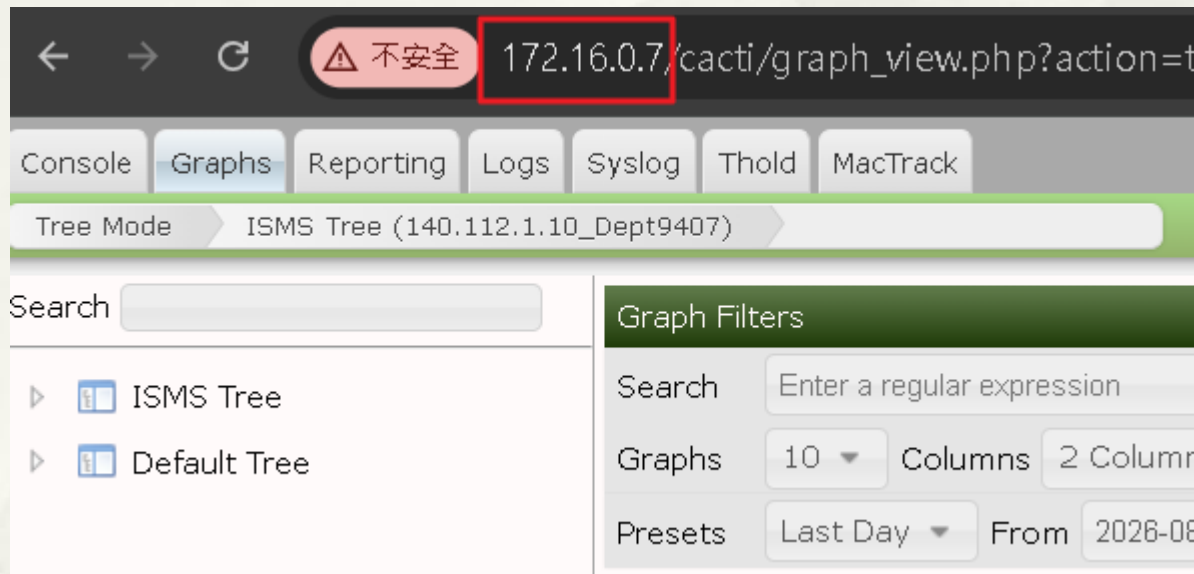
<https://proxy01.buda.idv.tw/cacti>

```
* /etc/nginx/sites-enabled/default
server {
    listen 443 ssl;
    server_name proxy01.buda.idv.tw; #Virtual Host
    ssl_certificate /etc/letsencrypt/live/proxy01.buda.idv.tw/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/proxy01.buda.idv.tw/privkey.pem;
    location / {
        proxy_pass http://172.16.0.7;
    }
}
```

L7 Reverse Proxy

HTTPs to HTTP

* 支援使用 https:// 連線



L7 Reverse Proxy HTTPs to HTTP Web Server@內網

導入 Zero Trust 架構

導入 Zero Trust 架構

- * SSH login to Web Server (172.16.0.7)
- * Ping 8.8.8.8 Ok

```
root@ubuntu20:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=1.57 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=1.62 ms
```

- * 移除 Gateway IP 設定

- * ~# nano /etc/netplan/00-installer-config.yaml

```
network:
  version: 2
  ethernets:
    ens160:
      dhcp4: no
      addresses:
        - 172.16.0.7/24
      # routes:
      #   - to: default
      #     via: 172.16.0.1
      nameservers:
        addresses: [168.95.1.1, 8.8.8.8]
```

- * ~# netplan apply
- * Ping 8.8.8.8 Failed

```
root@ubuntu20:~# ping 8.8.8.8
ping: connect: Network is unreachable
```

- * Zero Trust

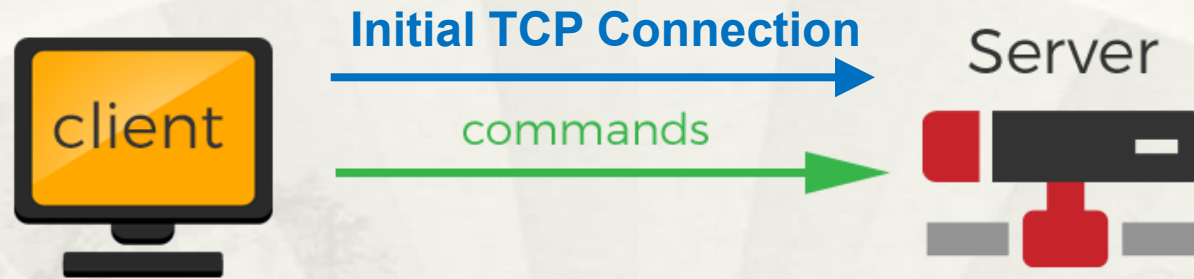
- * Web Server 僅能進行內網連線
- * 可避免未知後門/木馬(如: Reverse Shell) 殭屍網絡等 持續運作.

※仍無法阻擋 Web Shell (需導入 WAF 才能阻擋)

補充

Normal Shell vs. Reverse Shell

Normal shell



Reverse shell



圖片來源: <https://medium.com/@rietesh/python-reverse-shell-hack-your-neighbours-552561336ca8>

L7 Reverse Proxy HTTPs to HTTP Web Server@內網

導入 HTTP Basic Authentication
(ngx_http_auth_basic_module)

導入 HTTP Basic Authentication

- * 應用情境

- * 原 Web Server 無法提供"帳號密碼登入驗證"
- * 原 Web Server 無導入 HTTPS 協定
 - * HTTP Basic Authentication 密碼使用 base64 傳輸, 應啟用 HTTPS 加密傳輸

- * 使用 ngx_http_auth_basic_module

- * Ref. https://nginx.org/en/docs/http/ngx_http_auth_basic_module.html

/etc/nginx/sites-enabled/L7_proxy-auth.conf

```
server {  
    listen 443 ssl;  
    server_name proxy-auth.buda.idv.tw;  
  
    ssl_certificate /etc/letsencrypt/live/proxy-auth.buda.idv.tw/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/proxy-auth.buda.idv.tw/privkey.pem;  
  
    location / {  
        auth_basic "closed site"; # used as a realm  
        auth_basic_user_file basic_passwd;  
        proxy_pass http://172.16.0.7;  
    }  
}
```

/etc/nginx/basic_passwd

- * put at /etc/nginx/
- * 格式 user:password
- * /etc/nginx/basic_passwd
 - user1:A5F0pZqTMG1ks
 - user2:\$1\$Tx8uxb20\$qjj/olp4q2.VRW6W7bMkf1
 - user3:\$apr1\$6f41RlpJ\$mMhTDwY2SkQQuSibbRBFZ/
 - user4:\$5\$VUWHLFZriMwp7xr\$aVS8PQIWviBEo3EPGRGm80QovtLpvM20v/kG3runoZ.
 - user5:\$6\$yb1.q/Svs4QN.rQn\$7cDtRJczHXbgnVA9uBeSanoC57GiUd1nu.slkX.UbmE2DB4.tOGmgXREU3kJhYnlQ4zvUvDJZBCLX86FbeunY/
- * openssl passwd 12345
- * openssl passwd -apr1 12345
 - ※四種演算法皆支援
 - 6 SHA512-based password algorithm
 - 5 SHA256-based password algorithm
 - 1 MD5-based password algorithm(預設)
 - apr1 MD5-based password algorithm, Apache variant

導入 HTTP Basic Authentication

- * 需輸入"帳號密碼" 進行登入驗證

The screenshot shows a web browser window with the address bar displaying 'proxy-auth.buda.idv.tw'. A dark-themed login dialog is overlaid on the page. The dialog contains the following elements:

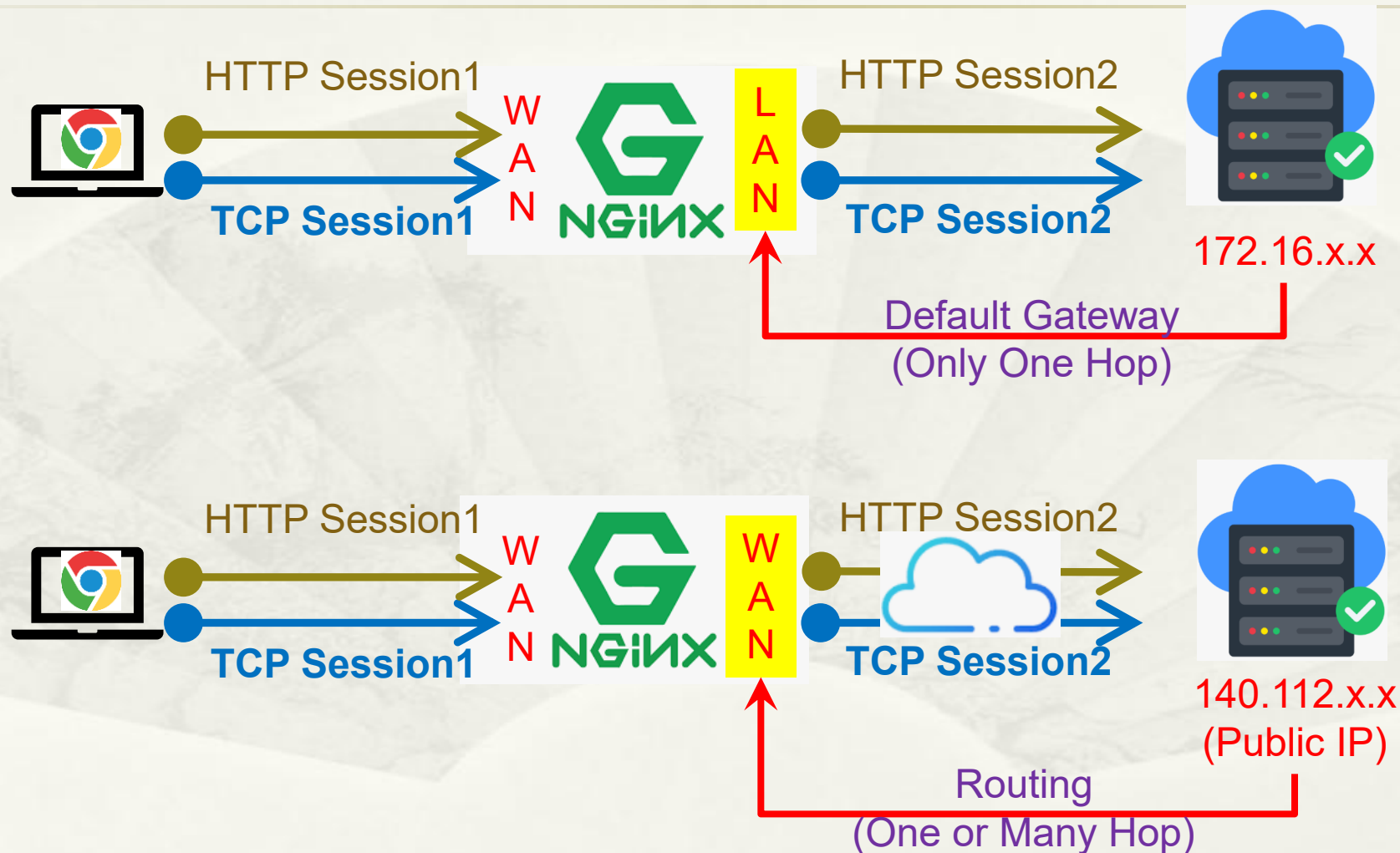
- 登入** (Login) header
- URL: `https://proxy-auth.buda.idv.tw`
- 使用者名稱** (Username) field: `user1`
- 密碼** (Password) field: masked with five dots
- 登入** (Login) button (light blue)
- 取消** (Cancel) button (dark blue)

Transparent Proxy Mode

Transparent Proxy Mode

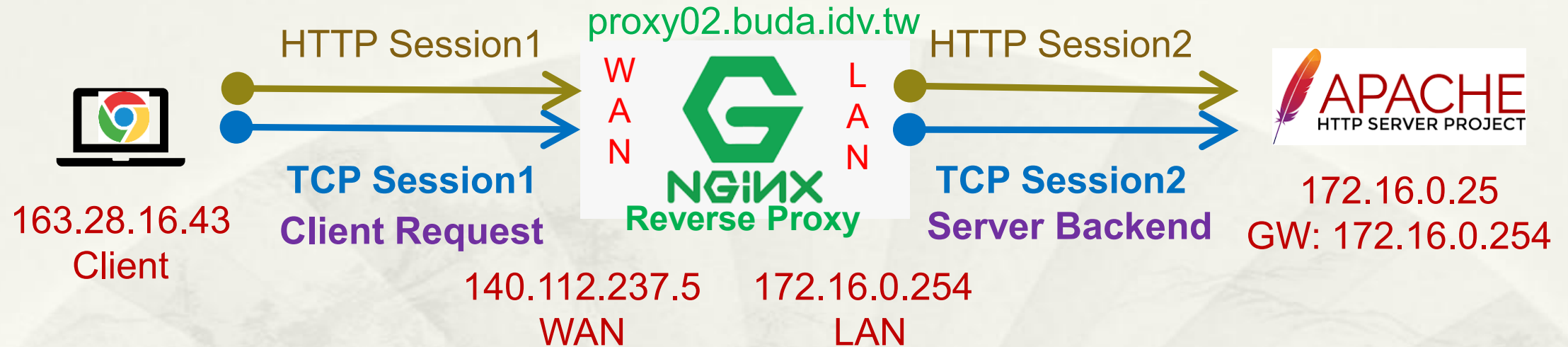
- * Browser Client Request
 - * 忠實傳遞 Client 資訊 to Backend Server
 - * Src IP/Port
 - * HTTP Header: Host
- * Backend Server Response
 - * 忠實傳遞 Server 資訊 to Client
 - * HTTP Header: Server

Transparent vs. Non-Transparent Mode



Transparent Proxy Mode

L7 Reverse Proxy HTTP to HTTP



使用者連線:

<http://proxy02.buda.idv.tw/>

環境準備

@ Reverse Proxy

- * **Enable IP Forward(Routing):** 暫時(重開機會失效):
~# echo "1" > /proc/sys/net/ipv4/ip_forward
Or
~# sysctl -w net.ipv4.ip_forward=1
- * **Enable NAT:** 暫時(重開機會失效)
~# iptables -t nat -A POSTROUTING -s 172.16.0.0/16 -o ens19 -j MASQUERADE
- * **Interception packets coming from the backend server:** 暫時(重開機會失效)
Mark packets coming from the backend
~# iptables -t mangle -A PREROUTING -p tcp -s 172.16.0.0/24 -j MARK --set-xmark 0x1/0xffffffff

Create a routing rule for marked packets
~# ip rule add fwmark 1 lookup 100

Route those packets to the local loopback (handling them internally)
~# ip route add local 0.0.0.0/0 dev lo table 100

/etc/nginx/sites-enabled/L7_proxy02.conf

```
server {
```

```
# SrcIP= Client's IP, SrcPort= Client's Port
```

```
proxy_bind $remote_addr:$remote_port transparent;
```

```
# By default, "Host" and "Connection" are not passed to proxied server.
```

```
# NGINX use "Connection= close" to proxied server.
```

```
proxy_set_header Host $host;
```

```
proxy_set_header Connection "keep-alive";
```

```
# By default, nginx does not pass "Date" "Server" "X-Pad" "X-Accel-" from a proxied server to a client.
```

```
proxy_pass_header Server;
```

```
proxy_pass_header Date;
```

```
proxy_pass_header X-Pad;
```

```
proxy_pass_header X-Accel-;
```

```
}
```

WAN

No.	Time	TTL	Source	Src Port	Destination	Dest Port	Protocol	Length	Info
1	0.000000	123	163.28.16.43	11977	140.112.237.5	80	TCP	66	11977 → 80 [SYN] Seq=0
2	0.000049	64	140.112.237.5	80	163.28.16.43	11977	TCP	66	80 → 11977 [SYN, ACK]
3	0.002012	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=
4	0.002334	123	163.28.16.43	11977	140.112.237.5	80	HTTP	137	GET / HTTP/1.1
5	0.002352	64	140.112.237.5	80	163.28.16.43	11977	TCP	54	80 → 11977 [ACK] Seq=
6	0.002955	64	140.112.237.5	80	163.28.16.43	11977	TCP	7354	80 → 11977 [PSH, ACK]
7	0.002960	64	140.112.237.5	80	163.28.16.43	11977	HTTP	3993	HTTP/1.1 200 OK (text
8	0.005339	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=
9	0.005339	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=
10	0.005415	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=
11	0.005551	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=
12	0.005969	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [FIN, ACK]
13	0.005995	64	140.112.237.5	80	163.28.16.43	11977	TCP	54	80 → 11977 [FIN, ACK]
14	0.007424	123	163.28.16.43	11977	140.112.237.5	80	TCP	60	11977 → 80 [ACK] Seq=

LAN

No.	Time	TTL	Source	Src Port	Destination	Dest Port	Protocol	Length	Info
1	0.000000	64	163.28.16.43	11977	172.16.0.25	80	TCP	74	11977 → 80 [SYN] Seq=0
2	0.000106	64	172.16.0.25	80	163.28.16.43	11977	TCP	74	80 → 11977 [SYN, ACK]
3	0.000125	64	163.28.16.43	11977	172.16.0.25	80	TCP	66	11977 → 80 [ACK] Seq=1
4	0.000149	64	163.28.16.43	11977	172.16.0.25	80	HTTP	173	GET / HTTP/1.0
5	0.000206	64	172.16.0.25	80	163.28.16.43	11977	TCP	66	80 → 11977 [ACK] Seq=1
6	0.000462	64	172.16.0.25	80	163.28.16.43	11977	TCP	7306	80 → 11977 [PSH, ACK]
7	0.000462	64	172.16.0.25	80	163.28.16.43	11977	HTTP	4097	HTTP/1.1 200 OK (text
8	0.000480	64	163.28.16.43	11977	172.16.0.25	80	TCP	66	11977 → 80 [ACK] Seq=1
9	0.000487	64	163.28.16.43	11977	172.16.0.25	80	TCP	66	11977 → 80 [ACK] Seq=1
10	0.000554	64	163.28.16.43	11977	172.16.0.25	80	TCP	66	11977 → 80 [FIN, ACK]
11	0.000591	64	172.16.0.25	80	163.28.16.43	11977	TCP	66	80 → 11977 [FIN, ACK]
12	0.000597	64	163.28.16.43	11977	172.16.0.25	80	TCP	66	11977 → 80 [ACK] Seq=1

Transparent Proxy Mode

Payload(L7)

WAN

LAN

The image displays two side-by-side Wireshark packet capture windows. The left window, titled 'Wireshark · Follow TCP Stream (tcp.stream eq 0) · wan080505.pcap', shows the WAN interface. The right window, titled 'Wireshark · Follow TCP Stream (tcp.stream eq 0) · lan080505.pcap', shows the LAN interface. A red arrow points from the WAN window to the LAN window, and a blue arrow points from the LAN window back to the WAN window, indicating the flow of traffic through the proxy.

HTTP Request Header (WAN):

```
GET / HTTP/1.1
Host: proxy02.buda.idv.tw
User-Agent: curl/8.21.0
Accept: */*
```

HTTP Response Header (WAN):

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 10960
Connection: keep-alive
Date: Wed, 05 Aug 2026 08:22:11 GMT
Server: Apache/2.4.66 (Ubuntu)
Last-Modified: Mon, 11 May 2026 07:00:29 GMT
ETag: "2ad0-651854d4ebbac"
Accept-Ranges: bytes
Vary: Accept-Encoding
```

HTTP Request Header (LAN):

```
GET / HTTP/1.0
Host: proxy02.buda.idv.tw
Connection: keep-alive
User-Agent: curl/8.21.0
Accept: */*
```

HTTP Response Header (LAN):

```
HTTP/1.1 200 OK
Date: Wed, 05 Aug 2026 08:22:11 GMT
Server: Apache/2.4.66 (Ubuntu)
Last-Modified: Mon, 11 May 2026 07:00:29 GMT
ETag: "2ad0-651854d4ebbac"
Accept-Ranges: bytes
Content-Length: 10960
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

The body of the response in both windows is identical, showing the HTML content of the page, including the DOCTYPE declaration, XML namespace declarations, and the main content area.

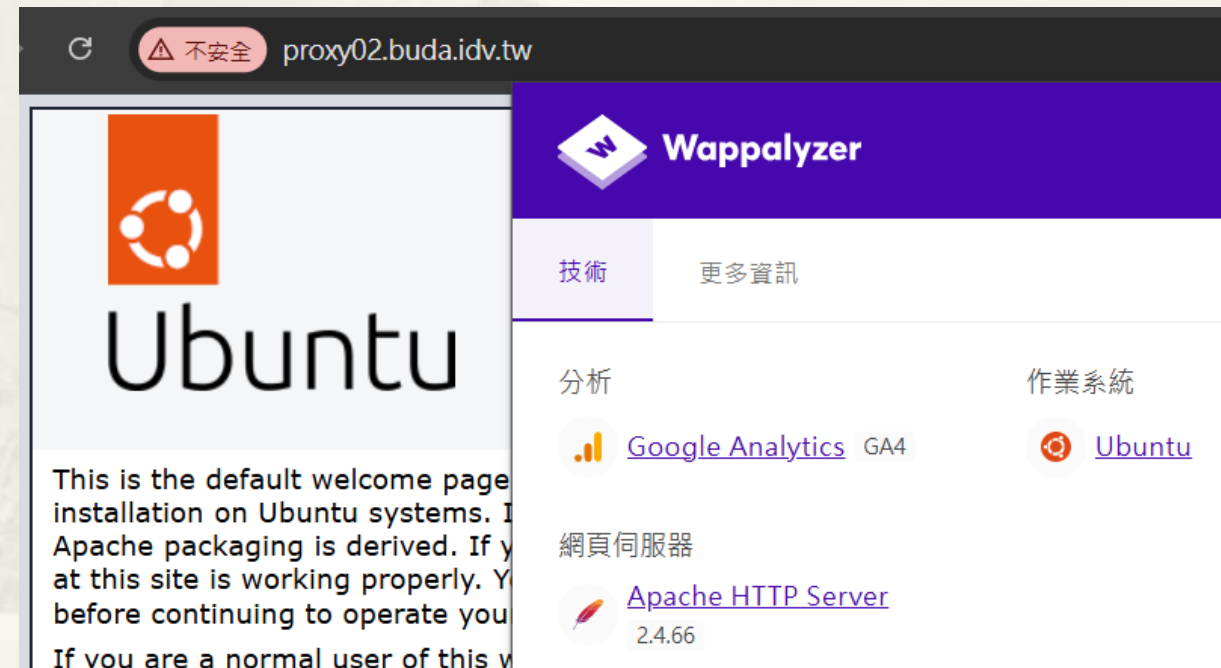
Transparent Proxy Mode

直接顯示原 Web Server & OS 版本

原始網站



L7 Reverse Proxy





簡報完畢
謝謝