# 中華民國第 58 屆中小學科學展覽會

# 作品說明書

---

高級中等學校組　電腦與資訊學科

052501

## 熱點上的 Mirai- IOT 之 DDoS 防制研究

學校名稱：新北市立鶯歌高級工商職業學校

| 作者： | 指導老師： |
| --- | --- |
| 職一　呂育帆<br>職一　陳曜廷<br>職一　傅建智 | 曾盛如 |


關鍵詞：AP、Firewall

# 摘要

隨著物聯網裝置的大量運用，惡意人士可以操控物聯網裝置的惡意程式 Mirai 發動 DDoS 攻擊，AP 熱點是物聯網裝置的上網窗口，如果能從第一層 AP 進行防禦，將 DDoS 消弭於來源，可避免進一步對外擴大攻擊的危害。

Mirai 控制數量龐大的 IoT bot(殭屍)，是一種新型態的 DDoS 攻擊手法。我們經由特徵之探討，以達到 Mirai 的各種攻擊模式防制。

同時針對常見的 DDoS 攻擊型態，紀錄對外伺服器基本服務封包通訊，如：TCP:ACK、Http:80、ICMP 等，進行統計、分析、學習，運算出存取控制列表 ACL，在 AP 攔阻 DDoS 攻擊封包，建立即時防禦機制。

# 壹、研究動機

鑑於 106 年度臺灣學術網路危機處理中心資安巡迴研討會之議題仍以「網路威脅手法暨 DDoS 分析」主題，可見此項威脅為校園的網站安全防護的重點，並且刻不容緩。

根據 iThome 報導：「駭客們在網路論壇上發布消息，想要出租以 Mirai 惡意軟體為主的殭屍網路，其殭屍網路所控制的裝置數量已達到四十萬個」。從這篇報導看來，受 Mirai 感染的裝置數量龐大，如果利用大量的 IoT 裝置攻擊某些目標，所造成的損失將難以估計。

IoT 裝置主要透過 WIFI AP 連線，如能在第一線的將 DDoS 防堵，並且經由伺服器自動設定各個 AP 之 ACL(Access Control List)，實現即時設定快速打擊。

# 貳、研究目的

建立 AP 熱點防禦，避免 IoT 危害。以封包分析數據找出異常 IoT bot，立即防禦，以達成以下目的：

一、 剖析 DDoS 特性及歸納因應對策。

二、 建置 WIFI AP 及防禦平台。

三、 分析相關封包紀錄，辨識異常。

四、 即時設定 ACL，無縫防禦。

# 參、研究設備及器材

## 一、 硬體設備與材料

主要的硬體項目及所使用相關設備與材料如下：

(一) 建立自製 AP 熱點：以適度擷取 IoT 封包紀錄。

(二) 架構伺服器分析演算平台。

(三) IOT 實驗組設置。

表 1 使用材料設備表

| | | |
|---|---|---|
|  |  |  |
| 網路伺服器-後端分析 | 筆記型電腦-前端呈現 | WiFi 無線基地台 |
|  |  |  |
| 平板電腦 - IOT | Raspberry Pi 3 B – AP | Wireless USB WiFi Adapter |
|  |  |  |
| 智慧手機-IOT | Smart 7688-IOT | NodeMCU Lua WIFI V3-IOT |

## 二、相關軟體

本研究在電腦軟體上的工作項目為：

(一) 規畫研究構想。

(二) 架設自製無線熱點環境。

(三) 設計伺服器分析演算平台。

(四) 提供實驗組將 Mirai 主控安裝及植入。

所需的相關軟體如下：

表 2 相關軟體表

| 相關軟體表 | | |
|---|---|---|
| **項次** | **軟體名稱** | **規格用途** |
| 1 | PHP+MySQL 實驗平台 | 數據分析、圖表繪製之平台 |
| 2 | FreeBSD RPI | Ap 熱點建立 |
| 3 | Mirai Sorce code | Mirai 主控及值入程式 |
| 4 | Adobe Dreamweaver | 演算分析及繪圖程式編寫 |
| 5 | Notepad++ | 程式碼及資料編修 |
| 6 | Packet Tracer | 防禦架構繪製，模[擬測試 |
| 7 | Putty | SSH、Telnet 連線操作 |

# 肆、研究過程或方法

首先建立 IOT AP 無線環境，並進行封包紀錄，以分析對象 Mirai。接著在 AP 中加入防火牆規則，以防制 IOT 上 Mirai 攻擊。透過統計演算找出適當 Entropy RU，以進行判別及回饋。所計畫研究程序如下圖 1 及研究架構如圖 2：

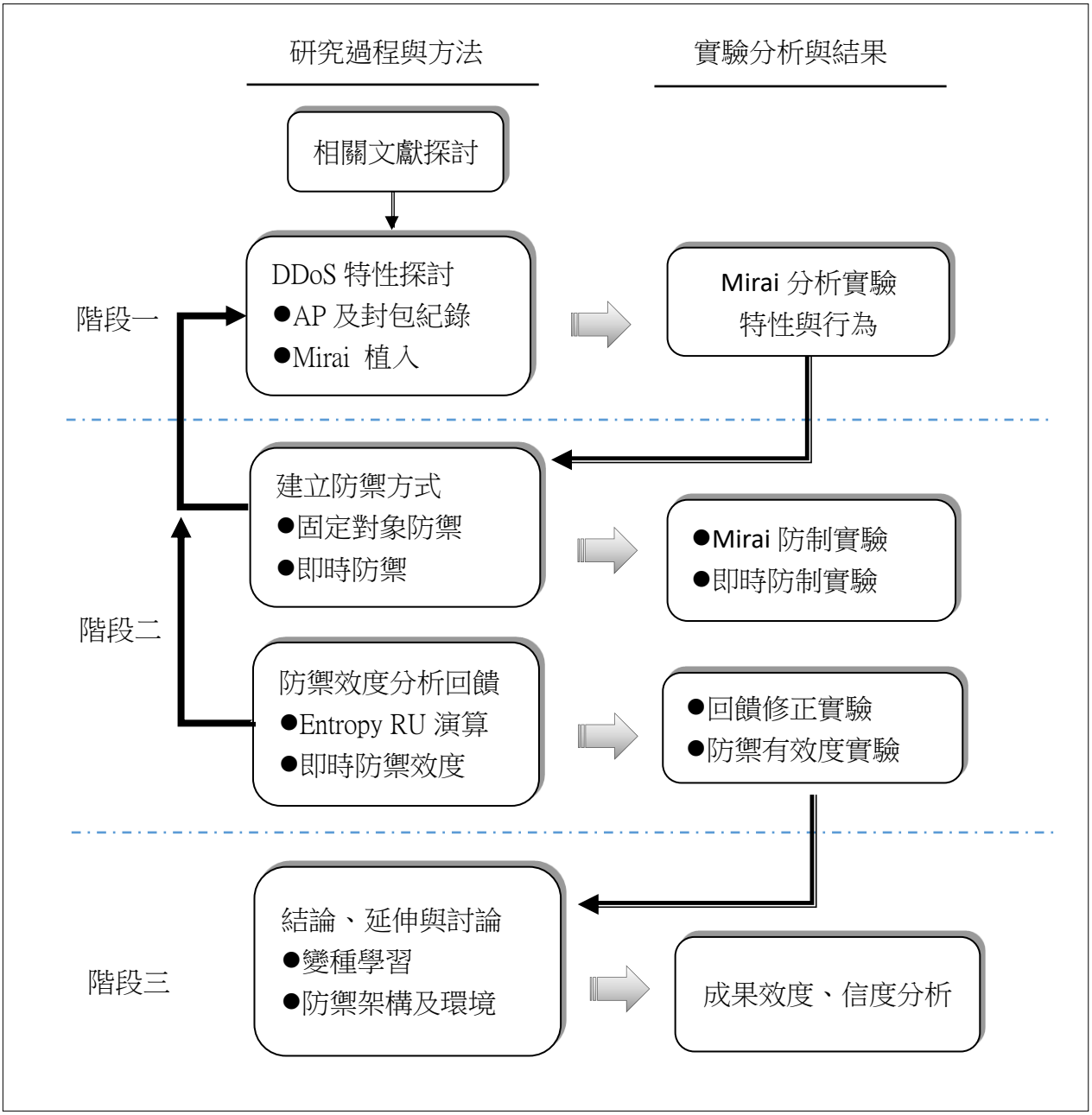



圖 1　AP 熱點之 DDoS 研究流程

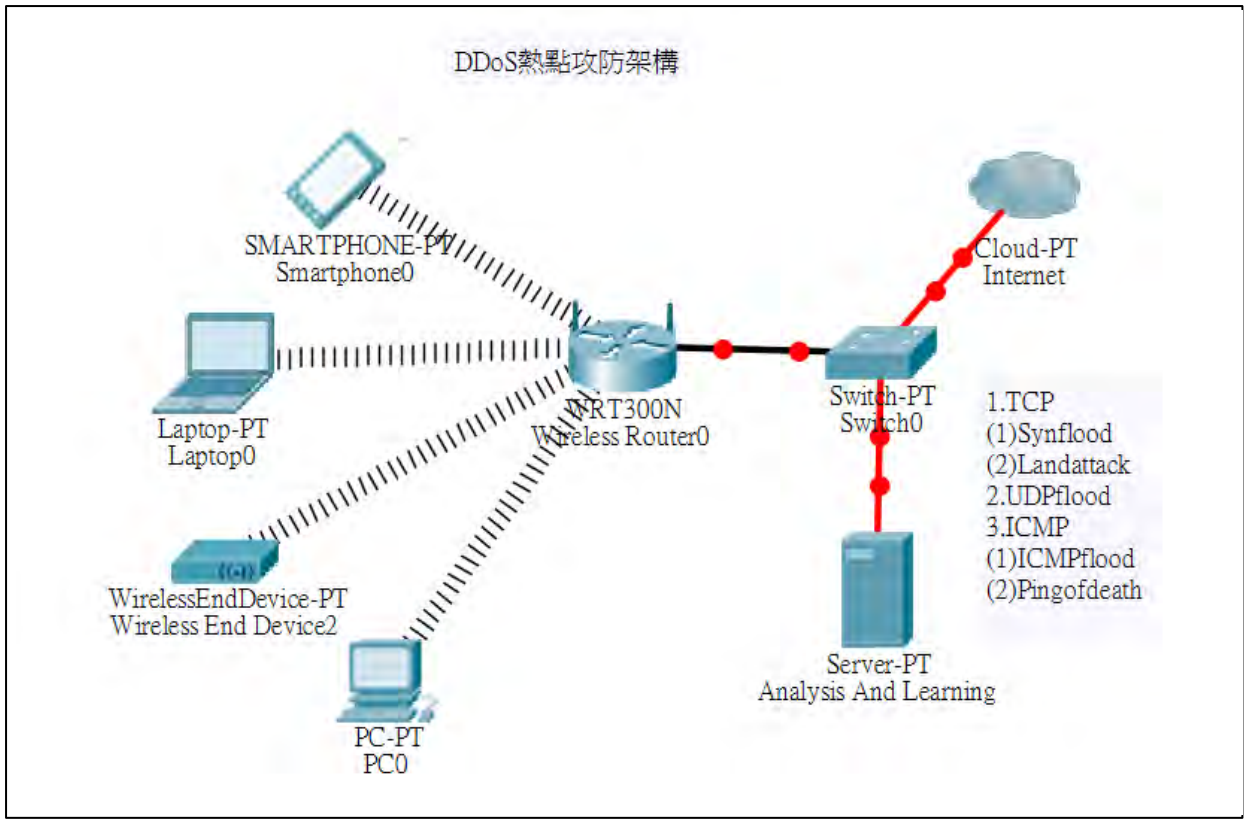



圖 2　AP 熱點之 DDoS 來源防禦架構

## 一、 DDoS 及 Mirai 特性探討

### (一) DDoS 攻擊特性

DDoS 攻擊目前最常發生在 OSI 第 3 (網路層)、4(傳輸層) 、 6(展示層) 和 7 層(應用層)，常見攻擊說明如表 2:

表 2 DDoS 攻擊方式表

| 編號 | OSI 層別 | 攻擊方式 | 備註 |
|---|---|---|---|
| 1 | 網路層 | flood UDP reply attack、ICMP flood | |
| 2 | 傳輸層 | SYN flood | |
| 3 | 展示層 | SSL session state attack、SSL negotiation attack、SSL vulnerability exploitation and compliance attack… | |
| 4 | 應用層 | SQL injection、Encrypted web attack、Encoding and Evasion、HTTPS/HTTP Floods… | |

以上分層可區分攻擊類型，通常不同攻擊類型，需要使用不同的技術來分析及防禦。在第 6、7 層的 DDos 攻擊，在偵測與防禦上較困難。第 3、4 層的攻擊是 DDos

較常見的類型，都是以大流量來癱瘓伺服器、防火牆、負載平衡器等，這種攻擊的特徵較明確，也較容易被檢測出來，在實作 DDos 防禦上較具可行性。

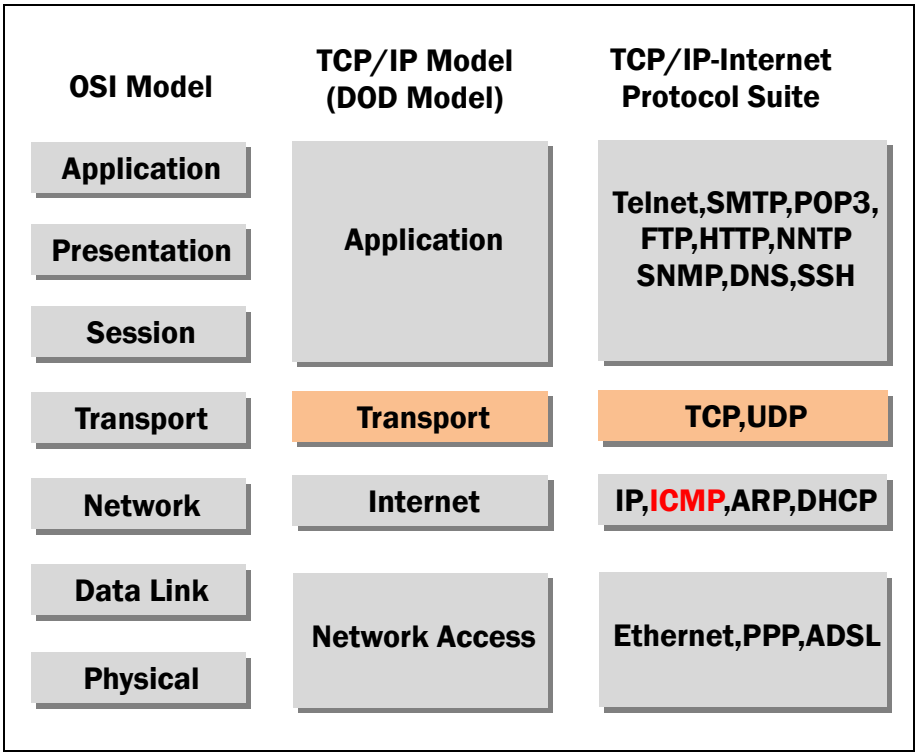



圖 3 TCP/IP DoD 模型中的結構圖

(二) Mirai Botnet 攻擊特性

1. Mirai 攻擊種類

從 GitHub Mirai 原始碼中 Mirai-Source-Code/mirai/bot/attack.h 內定義了攻擊種類如下表：

表 3 Mirai Botnet 的攻擊種類

| 編號 | 名稱 | 攻擊方式 | 說明 |
|---|---|---|---|
| 1 | UDP | Straight up UDP flood | *大量發送 UDP 封包 |
| 2 | VSE | Valve Source Engine query flood | *遊戲引擎平台 UDP 攻擊 |
| 3 | DNS | DNS water torture | *不存在域名的大量查詢 |
| 4 | SYN | SYN flood with options | TCP 交握 SYN 攻擊 |
| 5 | ACK | ACK flood | TCP 交握 ACK 攻擊 |
| 6 | STOMP | ACK flood to bypass mitigation devices | TCP 交握 ACK 攻擊，跳過緩衝設備 |
| 7 | GREIP | GRE IP flood | IP 通用路由封包攻擊 |

| | | | |
|---|---|---|---|
| 8 | GREETH | GRE Ethernet flood | 乙太通用路由封包攻擊 |
| 9 | UDP_PLAIN | Plain UDP flood optimized for speed | 無檔頭 UDP 封包攻擊 |
| 10 | HTTP | HTTP layer 7 flood | HTTP 網頁服務政擊 |

Mirai 攻擊方式種類有 10 項，而且其中 1、2、4、5、6 項攻擊埠口眾多，可針對 IOT 的來源之下的上表中單項攻擊進行分析，以確認受感染 IOT 設備，再將確認受感染 IOT 設備之可能性低的大量對外連線(Out bound)限制。

## 2. Mirai Botnet 攻擊方式

Mirai Botnet 是惡意人仕通過①C&C 伺服器向②Bot 發送攻擊指令，以形成 C&C 中心型殭屍網絡。同時，Bot 會掃描可感染的設備，將找到的設備信息發送給 Report 伺服器，由③downloader 植入 Bot 程序。如下圖：

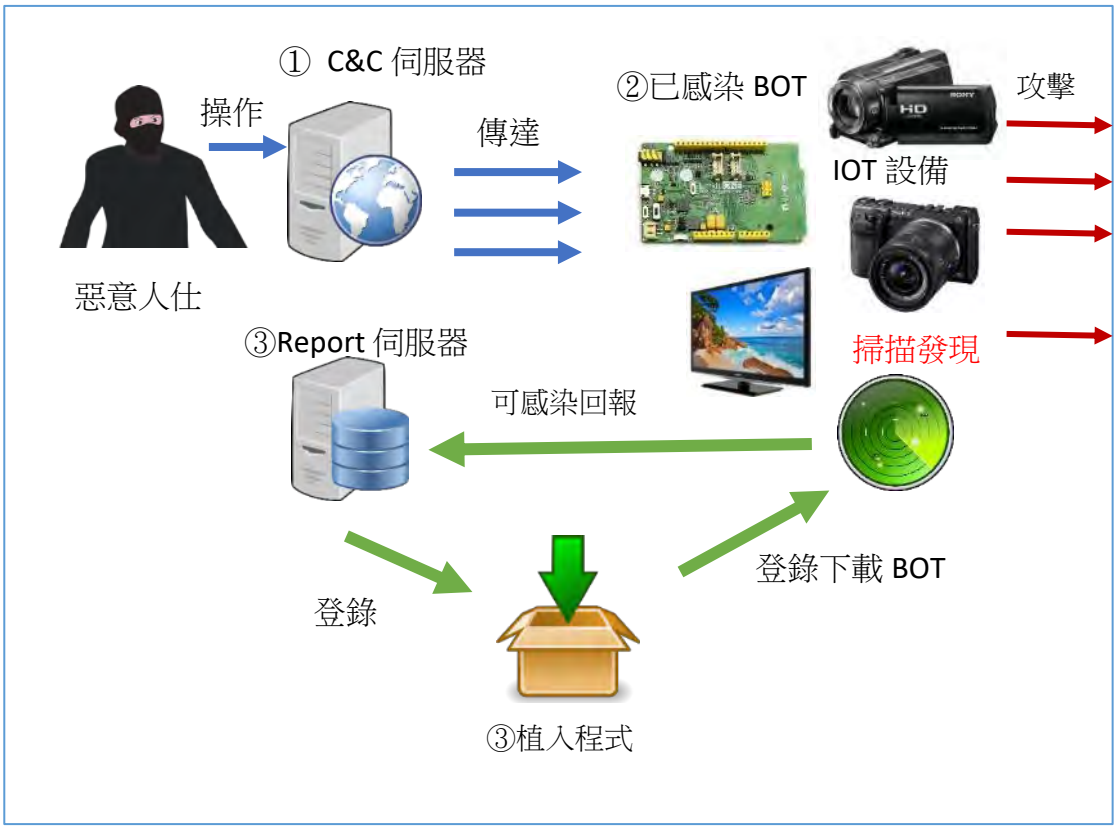



圖 4 Mirai Botnet 攻擊方式圖

## 二、 Mirai 攻擊行為實驗

(一) Entropy RU：相對不確性理論引用。

Shannon Entropy(熵)可以用來表示多事件不確定性，如果值愈大不確定性愈高。對於網路的異常攻擊其特性正好相反，極具規則性、連續性，因此算出來 Entropy 愈低，代表愈可能是異常攻擊。以下是為引用與推論：

1. 基本推演

(1)1 位元有 0,1，2 種不確定，$\log_2 2=1$

(2)2 位元有 4 種不確定，$\log_2 4=2$

(3)機率與隨機變量：如：兩胎中生男生女之熵 $\frac{1}{2}\times 1+\frac{1}{4}\times 2+\frac{1}{4}\times 2=\frac{3}{2}\text{bits}$

表 4 機率與隨機變量一例

| 組合 | 機率量(P) | 個別之隨機變量(I) |
|---|---|---|
| 男女 | $\frac{1}{2}$ | $\log_2 2=1$ |
| 男男 | $\frac{1}{4}$ | $\log_2 4=2$ |
| 女女 | $\frac{1}{4}$ | $\log_2 4=2$ |

2. 有限的樣本時，熵的公式

$$\mathrm{H(X)}=\sum_{\mathrm{i}\in\mathrm{X}} P(x_i)\,\mathrm{I}(x_i)$$

由上例：$\mathrm{I}(x_i)=\frac{1}{\log_{\mathrm{b}} P(x_i)}=-\log_{\mathrm{b}} P(x_i)$

P 為 X 的機率量函數，I(X)本身是個別之隨機變量

$$\mathrm{H(X)}=-\sum_{\mathrm{i}\in\mathrm{X}} P(x_i)\log_{\mathrm{b}} P(x_i)$$

X 熵值 H，當 b = 10，熵的單位是 Hart。

3. 熵的相對不確定（relative uncertainty）與絕對不確定（absolute uncertainty）

$$\text{相對不確定 RU}=\frac{\text{測量值}}{\text{絕對不確定 AU}}$$

$$\mathrm{RU(X)}=\frac{\mathrm{H(X)}}{\mathrm{H_{max}(X)}}=\frac{\mathrm{H(X)}}{\log min\{\mathrm{N_x}|m\}}$$

RU 可能最大值為 1

承上，兩胎中男女組合之$\mathrm{RU}(\frac{1}{2}) = \frac{\frac{3}{2}}{\log_2 3} = \mathbf{0.9464}$ ←4 事件佔 2 次（為中間值）。

4. 異常封包 RU 分析：

下表為封包偵測一例，前 10 項 RU 值計算，RU=0.70387 < 0.9464，因為值低則最大項異常可能性高，如表 5：

表 5 異常封包 RU 值

| 編號 | 次數 | 機率 P | P(x)*I(x) | RU |
|---|---|---|---|---|
| 1 | 40 | 0.48193 | 0.50752 | 0.70387 |
| 2 | 16 | 0.19277 | 0.45784 | |
| 3 | 9 | 0.10843 | 0.34754 | |
| 4 | 7 | 0.08434 | 0.30089 | |
| 5 | 2 | 0.02410 | 0.12952 | |
| 6 | 2 | 0.02410 | 0.12952 | |
| 7 | 2 | 0.02410 | 0.12952 | |
| 8 | 2 | 0.02410 | 0.12952 | |
| 9 | 2 | 0.02410 | 0.12952 | |
| 10 | 1 | 0.01205 | 0.07681 | |
| | 83 | 1 | 2.333820 | <-Entropy |

最大項異常攻擊確認後，可繼續計算後其他項的可能性，如表 6：

表 6 前項確認移除後之 RU 值

| 編號 | 次數 | 機率 P | P(x)*I(x) | RU |
|---|---|---|---|---|
| 1 | 40 | 0.372093 | 0.5307032 | 0.70387 |
| 2 | 16 | 0.209302 | 0.4722572 | 0.81543 |
| 3 | 9 | 0.162791 | 0.4263342 | 0.86668 |
| 4 | 7 | 0.046512 | 0.2058728 | 0.89857 |
| 5 | 2 | 0.046512 | 0.2058728 | 0.98661 |
| 6 | 2 | 0.046512 | 0.2058728 | 0.98239 |
| 7 | 2 | 0.046512 | 0.2058728 | : |
| 8 | 2 | 0.046512 | 0.2058728 | : |
| 9 | 2 | 0.023256 | 0.1261922 | : |
| 10 | 1 | 0.01205 | 0.07681 | : |
| | 83 | 1 | 2.5848506 | <-Entropy |

上面的推演結果以程式實現，主要的演算法程式如圖 5：

```
$counts=array(40,16,9,7,2,2,2,2,2,1);     //存取次數
$num=count($counts);          //總項目
$sum=array_sum($counts);;     //次數總合
foreach($counts as $ci){
    $p=$ci/$sum;              //個別機率
    $en_all[]=$p * -log( $p,2);     //個別熵 P(x)*I(x)
}
$entropy=array_sum($en_all);     //計算 Entroy
$RU=$entropy/log($num,2);          //計算 RU
```

圖 5 RU 演算程式

## (二)分析架構與平台

1. AP 實驗環境：

(1)硬體：Raspberry Pi 3 B、創見 UHS-I 600x 高速記憶卡、 EDIMAX EW-7811Un WIFI USB

(2) 軟體：FreeBSD12 RPI、內建 AP(hostapd)、內建 PF(packet filter)、MySQL Client

(3) 運作說明：如圖 5 所示，安裝架設 AP，並取得 AP 紀錄如圖 6。

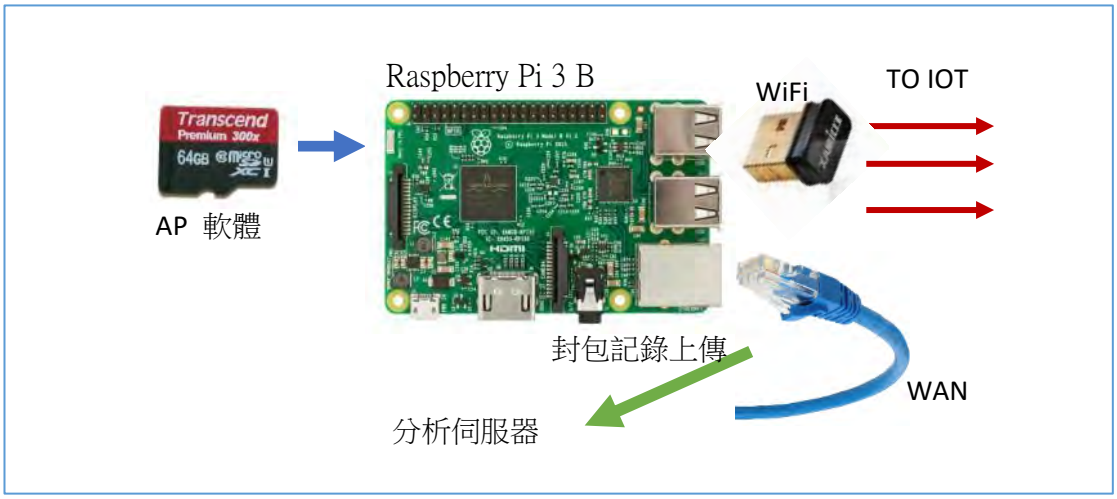



圖 6 建立可規劃之 AP


圖 7 自行架設 AP 運作及紀錄

2. 伺服器分析平台：

(1) 硬體：DELL PowerEdge R730 伺服器、UNIFOSA 2U 機架式磁碟陣列(SSD)

(2) 軟體：FreeBSD12、Apache、PHP、MySQL Server

(3) 運作說明：接收各 AP 之封包紀錄，分析並決定處理策略。

| pl_id | pl_out_time | pl_time | pl_rule | pl_action | pl_direct | pl_if | pl_src_addr | pl_src_port | pl_dst_addr | pl_dst_port | pl_proto | pl_count | pl_len |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1363132082 | 1363131994.440217 | 33 | 1 | 1 | em0 | 2736038777 | 1112 | 1876774285 | 40038 | 0 | 1 | 0 |
| 2 | 1363132082 | 1363132000.003976 | 33 | 1 | 1 | em0 | 2736038687 | 56180 | 1249714045 | 5222 | 0 | 1 | 0 |
| 3 | 1363132440 | 1363132047.852354 | 33 | 1 | 1 | em0 | 2736038777 | 1118 | 2736596250 | 3939 | 0 | 3 | 0 |
| 4 | 1363132440 | 1363132052.596841 | 33 | 1 | 1 | em0 | 2736038773 | 60164 | 1136901070 | 5050 | 0 | 1 | 0 |
| 5 | 1363132440 | 1363132068.460351 | 33 | 1 | 1 | em0 | 2736038773 | 60182 | 1136900797 | 5050 | 0 | 1 | 0 |
| 6 | 1363132440 | 1363132120.184468 | 33 | 1 | 1 | em0 | 2736038687 | 49171 | 1249714045 | 5222 | 0 | 2 | 0 |
| 7 | 1363132440 | 1363132170.073283 | 33 | 1 | 1 | em0 | 2736038880 | 61505 | 2736038705 | 25864 | 0 | 1 | 0 |
| 8 | 1363132440 | 1363132175.153073 | 33 | 1 | 1 | em0 | 2736038880 | 61511 | 2736038705 | 15415 | 0 | 1 | 0 |
| 9 | 1363132440 | 1363132177.403386 | 33 | 1 | 1 | em0 | 2736038880 | 61514 | 2736038705 | 56928 | 0 | 1 | 0 |
| 10 | 1363132440 | 1363132178.867907 | 33 | 1 | 1 | em0 | 2736038880 | 61516 | 2736038705 | 59165 | 0 | 1 | 0 |

資料庫紀錄

圖 8 伺服器接收 AP 上傳之封包原始記錄

| No. | Time | Duration | Source addr | Protocol | Counts | Commet |
|---|---|---|---|---|---|---|
| 5616 | Jan 19 00:00:00 | Day | 10.1.1.10 | tcp | 90 | |
| 5614 | Jan 19 00:00:00 | Day | 10.1.1.10 | udp | 61 | |
| 5632 | Jan 25 00:00:00 | Day | 10.1.1.10 | tcp | 34 | |
| 5606 | Jan 11 00:00:00 | Day | 10.1.1.10 | tcp | 11 | |
| 5618 | Jan 19 00:00:00 | Day | 10.1.1.10 | other | 10 | |
| 5610 | Jan 11 00:00:00 | Day | 10.1.1.10 | udp | 2 | |
| 5630 | Jan 25 00:00:00 | Day | 10.1.1.10 | icmp | 2 | |
| 5584 | Jan 08 00:00:00 | Day | 10.0.1.10 | udp | 2 | |
| 5588 | Jan 08 00:00:00 | Day | 10.10.10.10 | icmp | 2 | |

累計次數

圖 9 伺服器上之封包記錄統計

## (三)Mirai 攻擊分析：

受 Mirai 感染的裝置，將持續在網際網路上掃描物聯網裝置的 IP 位址和連接埠(先從相同網段開始，之後擴張到其他網段)。掃描到其他未受感染裝置的 IP 位址和連接埠之後，Mirai 會通過多種預設帳號和密碼嘗試登入該裝置，如果可以登入該裝置，隨即開始安裝 Mirai。

### 1.攻擊封包特徵

(1) 封包特徵：每一連線封包是由協定、來源位址、目的位址、來源埠、目的埠、及封包大小等基本六項；TCP 另有 8 種三方交握旗號，如表 5。

(2) 對於 IOT 的攻擊是由內而外攻擊，可以只考慮對外連線封包。並且是由駭客下命令

表 7 TCP 三方交握旗號

**TCP flags**

Matching TCP packets based on flags is most often used to filter TCP packets that are attempting to open a new connection. The TCP flags and their meanings are listed here:

- **F** : FIN - Finish; end of session
- **S** : SYN - Synchronize; indicates request to start session
- **R** : RST - Reset; drop a connection
- **P** : PUSH - Push; packet is sent immediately
- **A** : ACK - Acknowledgement
- **U** : URG - Urgent
- **E** : ECE - Explicit Congestion Notification Echo
- **W** : CWR - Congestion Window Reduced

摘自：OpenBSD PF FAQ

攻擊特定幾個目標，可以針對目標 IP 分析（在時段內）。

(3) 受攻擊目標因承受短時間大量封包，回應時間將大幅增加，甚至 Time out。

(4) Mirai 攻擊方式種類有 10 項，並經由 CnC 伺服器命令攻擊其中一項。

2. Entropy 程式流程：

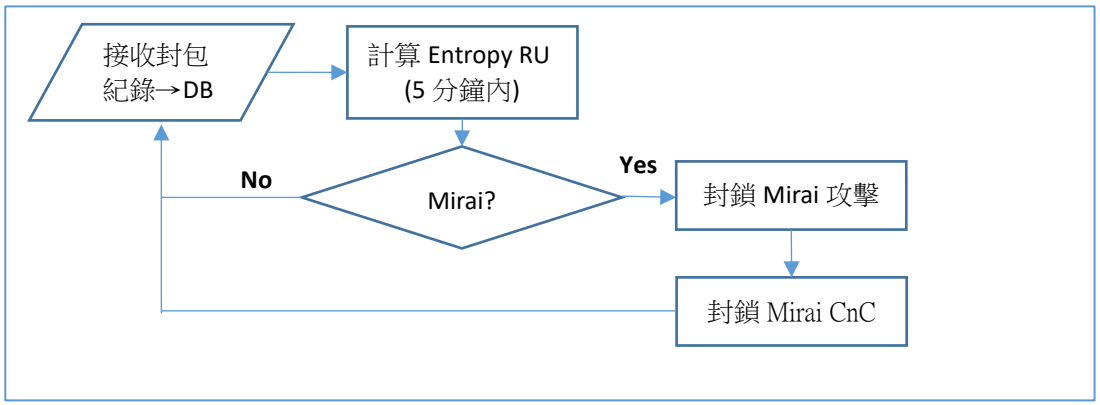



圖 10　Entropy 程式處理流程

3. **Mirai 攻擊偵測實驗**：

Mirai 攻擊雖然有 10 種，但是在 OSI 第四層均屬 UDP/TCP。以下針對這二項進行實驗偵測:：

表 6　Mirai 攻擊種類次數統計

| Attack Type | Attacks | Targets | Class |
|---|---|---|---|
| HTTP flood | 2,736 | 1,035 | A |
| UDP-PLAIN flood | 2,542 | 1,278 | V |
| UDP flood | 2,440 | 1,479 | V |
| ACK flood | 2,173 | 875 | S |
| SYN flood | 1,935 | 764 | S |
| GRE-IP flood | 994 | 587 | A |
| ACK-STOMP flood | 830 | 359 | S |
| VSE flood | 809 | 550 | A |
| DNS flood | 417 | 173 | A |
| GRE-ETH flood | 318 | 210 | A |

UDP
TCP

摘自：Understanding the Mirai Botnet - Research at Google

(1) TCP 攻擊偵測實驗：


圖 11　TCP 的攻擊統計

(2) UDP 攻擊偵測實驗：

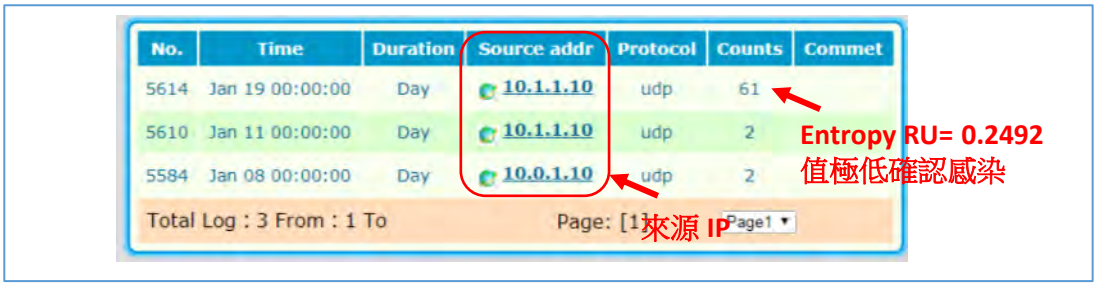



圖 12　UDP 的攻擊統計

## 三、 Mirai 即時防制

對於確定受感染之 IOT Bot，可在 AP 端封鎖其攻擊封包。

(一) 封鎖攻擊封包

表 8 – tcp、udp、icmp 封鎖攻擊之規則

```
block in on { $int_if } inet proto tcp from $lan_net to any flags S/SA no state
block in on { $int_if } inet proto udp from $lan_net to any no state
block in on { $int_if } inet proto icmp from $lan_net to any no state
```

(二) CnC 伺服器命令封鎖

由 mirai/bot/main.c，可知 IOT Bot 是主動向 CnC 溝通，並且透過 telnet(port 23)，可關閉 Bot 對外的 Tcp port 23，以斷絕後續攻擊命令。

```
315             // Try to read in buffer from CNC
316             errno = 0;
317             n = recv(fd_serv, rdbuf, len, MSG_NOSIGNAL | MSG_PEEK);
318             if (n == -1)
319             {
320                 if (errno == EWOULDBLOCK || errno == EAGAIN || errno == EINTR)
321                     continue;
322                 else
323                     n = 0;
324             }
```

Bot 接收 CnC 命令

圖 13　IOT Bot 向 CnC 命令

# 伍、研究結果

## 一、 偵測 Mirai 攻擊封包之負荷：

主要瓶頸在於 AP 上 SD 卡封包紀錄寫入速度，隨封包數增加而延遲處理時間。

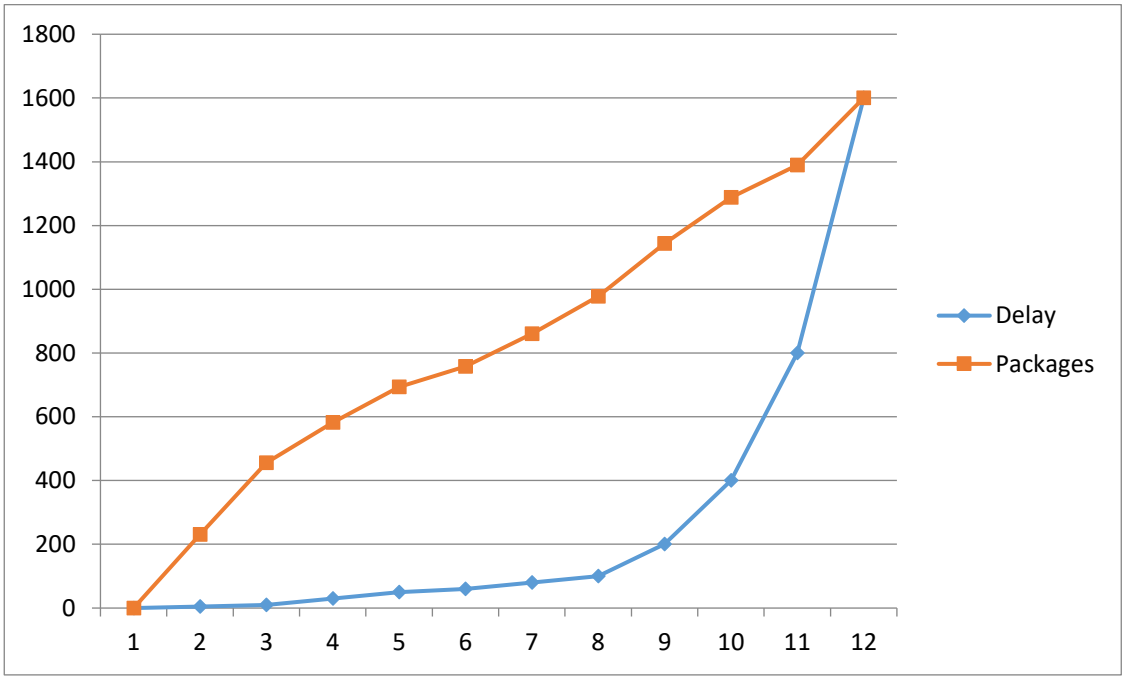



圖 14 封包紀錄寫入 SD 卡次負荷(秒)

## 二、 Mirai 防禦之負荷：

IOT BOT 由 CnC 命令設定每次攻擊方式及數量， AP 端對頻寬造成重大負擔。

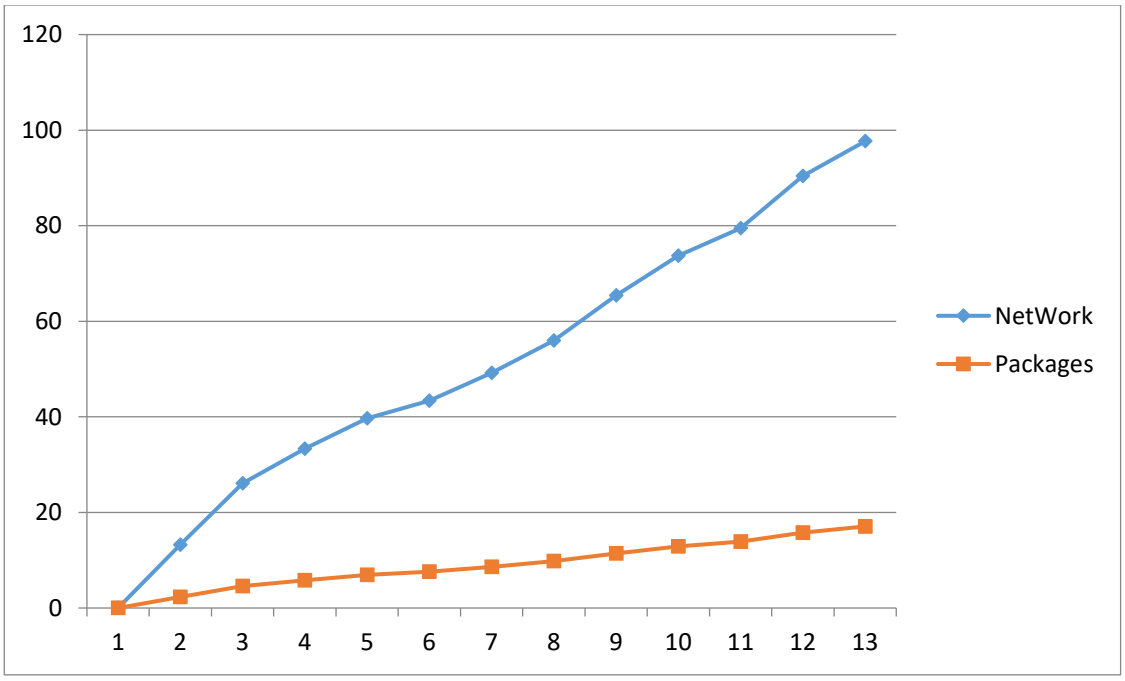



圖 15 封鎖數量之 AP 網路負荷(單位百筆)

## 三、 Mirai 即時防制成果

(一) AP 防制處理反應時間：以確認感染之 Bot，第一筆攻擊算起。

表 9 Mirai Botnet 防制處理反應時間

| 編號 | 攻擊方式 | 平均反應時間(分) | 備註 |
|---|---|---|---|
| 1 | UDP | 8.63 | |
| 2 | VSE | 9.31 | |
| 3 | DNS | 7.42 | |
| 4 | SYN | 12.23 | |
| 5 | ACK | 11.74 | |
| 6 | STOMP | 10.63 | |
| 7 | GREIP | 12.82 | |
| 8 | GREETH | 9.82 | |
| 9 | UDP_PLAIN | 6.44 | |
| 10 | HTTP | 10.91 | |

(二) CnC 封鎖後的 AP 網路負荷：

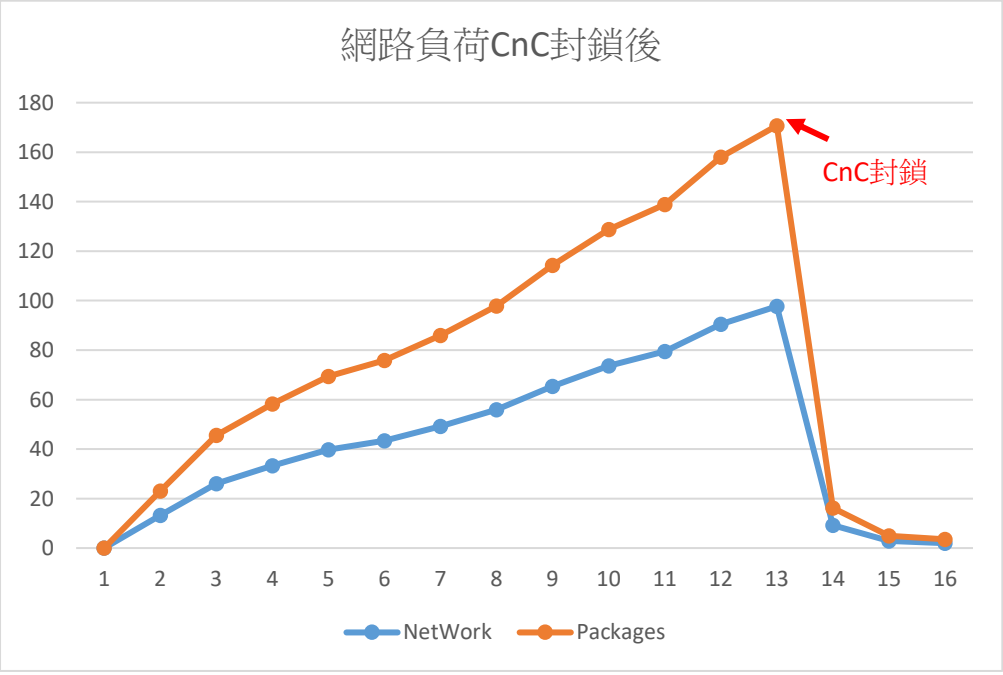



圖 16　AP 網路負荷 CnC 封鎖(單位十筆)

(三) 受攻擊主機網路負荷變化：

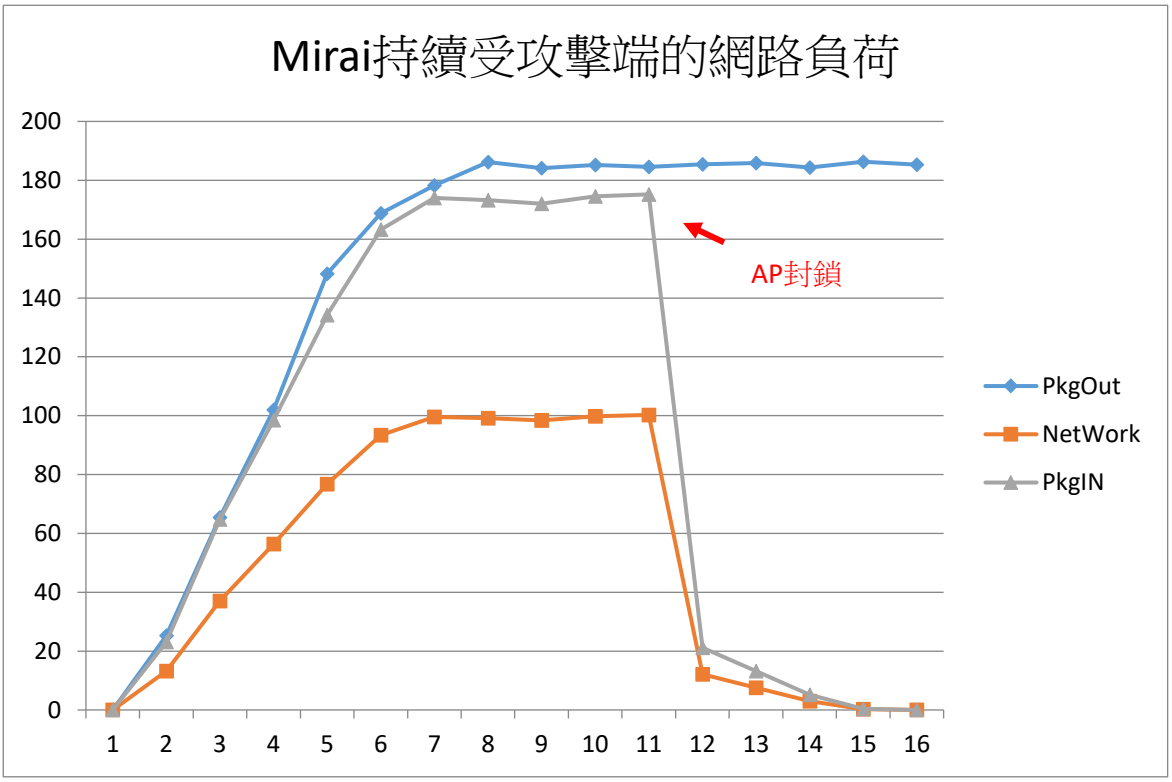



圖 17 受攻擊端網路負荷與防制封鎖後(單位十筆)

四、 **Entropy RU 偵測 Mirai 攻擊效率：** Mirai 攻擊方式雖然有 10 種，但每次進行單純一種攻擊，並且時間連續以形成攻擊，所計算 Entropy RU 明顯偏極低，有 100%偵測率。

# 陸、討論

Mirai bonet 如同其他 DDoS 有其變種，以 OMG 為例因為運作模式相似高，在 UDP 及 TCP 的 Entropy RU 判斷上亦可達成偵測：

一、 DDoS 的防禦：所有 DDoS 均有大量攻擊之特性，計算 Entropy RU 亦有極高準確度。

表 10　Entropy RU 偵測之準確度(The first Attack Mode )

| k | %True Positives | %False Negatives | %True Negatives | %True Positives | %Overall Accuracy |
|---|---|---|---|---|---|
| 3 | 99.65 | 0.35 | 100 | 0 | 99.82 |
| 5 | 99.65 | 0.35 | 100 | 0 | 99.82 |
| 7 | 99.59 | 0.41 | 99.2 | 0.8 | 99.40 |

摘自：用三階層式模組於偵測伺服器上的 DDoS 攻擊

二、 OMG Mirai bonet 變種偵測：OMG 新變種會在加入防火牆兩組規則，以允許兩個隨機埠口上的流量通過，以提供為惡意人士使用，其他攻擊與原來 Mirai 相同。

三、 MAC Address 精確定址：因為 AP 運作於 IOT 相同網段，使用 MAC 定址可更精確掌握內部網段的 IOT bot，在 OSI 第二層進行防禦。

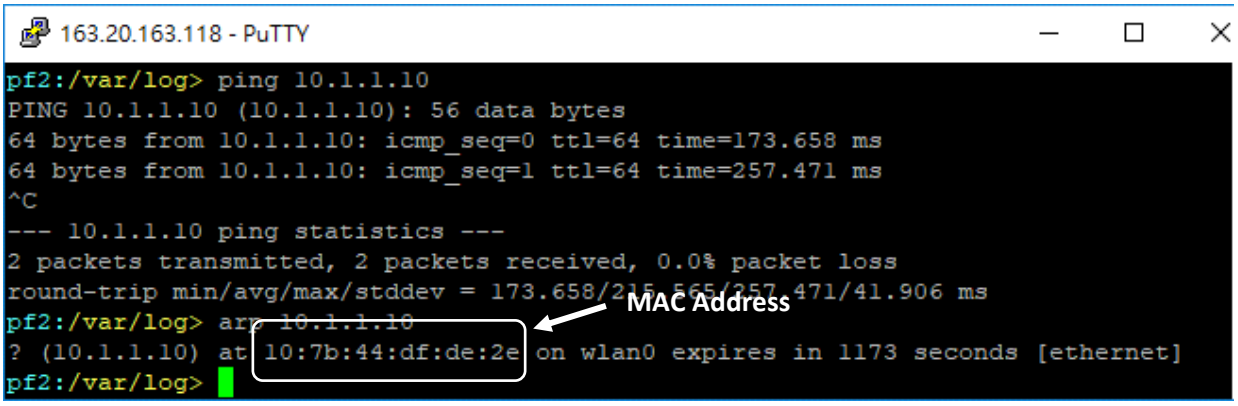



圖 18　IOT 設備之 Mac Address

# 柒、結論

AP 為 IOT 設備的前端上網設備，可做為 DDoS 攻擊第一道防線。以下是我們的各項結論：

一、 **熱點 AP 可有效防制 Mirai 攻擊：**可偵測出 Mirai bot，並即時封鎖防制，從來源隔離。

二、 **封鎖 Mirai CnC 可阻止後續攻擊：**IOT Bot 定時向 CnC 取得攻擊命令，封鎖可防再取得後續命令。

三、 **分析 UDP 及 TCP 封包同時可防制其它 DDoS 攻擊**：一般 DDoS 攻擊與 Mirai 相似，以大量攻擊為主，可透過類似方式防禦。

四、 **IOT 設備的安全預防：**定期更新 IOT 韌體，可增加系統安全。使用前需變更內部預設密碼，防止以密碼字典登入。

# 捌、參考資料及其他


政府機關 DDoS 防禦與應變作業程序(2016 簡報)。行政院國家資通安全會報技術服務中心

UDP DDoS 攻擊分析報告(2013)。TACERT 臺灣學術網路危機處理中心團隊製

臺灣學術網路危機處理中心資安巡迴研討會-網路威脅手法暨 DDoS 分析(2017)。臺灣學術網路危機處理中心

用三階層式模組於偵測伺服器上的 DDoS 攻擊之研究(2013)。台灣科技大學資訊工程所 黃冠錡、王子彥

Mirai 木马总结—从源码到反汇编。帝星捧月

1989 (51)熵(Entropy)(下) (52)熵(Entropy)(下)。中央研究院數學研究所數學傳播 李天岩

Mirai and IoT Botnet Analysis(2017 簡報)。Robert Graham

Understanding the Mirai Botnet(2017)。Research at Google

W.Richard Stens(2009)。TCP/IP Illuxtrated，Volumel 中譯本。台北市：和碩科技文化有限公司。

【Mirai 新變種 OMG 將物聯網設備變成代理伺服器】。

https://blog.trendmicro.com.tw/?p=54671

【熵 (Entropy)】Shannon 熵、Kolmogorov 熵、拓樸熵 (Topological Entropy)、Boltzmann 熵。

http://episte.math.ntu.edu.tw/articles/mm/mm_13_3_01/index.html

【治標更治本，如何從根源防護 DDoS 攻擊】https://www.pixpo.net/technology/0Hwd8Ajo.html

【機器學習: 異常偵測 (Machine Learning: Anomaly Detection) 】

http://murphymind.blogspot.tw/2017/07/anomaly.detection.html?m=1

【Mirai-Iot-BotNet/TUTORIAL.txt】

https://github.com/Screamfox/-Mirai-Iot-BotNet/blob/master/TUTORIAL.txt

【深度解析：「Mirai」原始碼的結構及其對策】 https://kknews.cc/tech/48yl5vv.html

【安全廠商與 ISP 聯手破獲 Mirai 變種 Satori 殭屍網路病毒，作者只是上網求教的業餘駭客】

https://www.ithome.com.tw/news/119922

【Mirai 惡意程式探討與防範】

http://www.cc.ntu.edu.tw/chinese/epaper/0043/20171220_4309.html

【利用 FreeBSD for Raspberry Pi 2 來製作無線網路基地台】

http://freebsdtaiwan.blogspot.tw/2017/04/freebsd-for-raspberry-pi-2.html

# 【評語】052501

概述：隨著物聯網裝置的大量運用，惡意人士可以透過操控物聯網裝置的惡意程式 Mirai 發動 DDoS 攻擊，而 AP 熱點是物聯網裝置的上網窗口，如果能從第一層 AP 來進行防禦，可避免進一步對外擴大攻擊的危害。Mirai 可以控制數量龐大的 IoTbot，是一種新型態的 DDoS 攻擊手法，此作品經由特徵之探討，以達到 Mirai 的各種攻擊模式防制。同時針對常見的 DDos 攻擊型態，記錄對外伺服器基本服務封包通訊，進行統計與分析，來建立即時防禦機制。

優點：

透過更改 AP 的機制來防止連線的物聯網裝置被用來進行 DDOS 攻擊。

可改進之處：

1. 所使用的演算法用計算 Entropy 值來分辨是否為惡意攻擊，但並沒有做大量的實驗來證明所取的判斷門檻值可以有效的判斷惡意攻擊，建議加強此部分的研究。
2. 在報告中可加強說明和分析為何在表 9 中的防制處理時間需要這麼長的時間才能完成。

# 作品海報

# 摘　要


隨著物聯網裝置的大量運用，惡意人士可以操控物聯網裝置的惡意程式Mirai發動DDoS攻擊，AP熱點是物聯網裝置的上網窗口，如果能從第一層AP進行防禦，將DDoS消弭於來源，可避免進一步對外擴大攻擊的危害。

Mirai控制數量龐大的IoT bot(殭屍)，是一種新型態的DDoS攻擊手法。我們經由特徵之探討，以達到Mirai的各種攻擊模式防制。

同時針對常見的DDoS攻擊型態，紀錄對外伺服器基本服務封包通訊，如：TCP:ACK、Http:80、ICMP等，進行統計、分析、學習，運算出存取控制列表ACL，在AP攔阻DDoS攻擊封包，建立即時防禦機制。


# 研 究 動 機

鑑於106年度臺灣學術網路危機處理中心資安巡迴研討會之議題仍以「網路威脅手法暨DDoS分析」主題，可見此項威脅為校園的網站安全防護的重點，並且刻不容緩。

根據 iThome報導：「駭客們在網路論壇上發布消息，想要出租以Mirai惡意軟體為主的殭屍網路，其殭屍網路所控制的裝置數量已達到四十萬個」。從這篇報導看來，受Mirai感染的裝置數量龐大，如果利用大量的IoT裝置攻擊某些目標，所造成的損失將難以估計。

IoT裝置主要透過WIFI AP連線，如能在第一線的將DDoS防堵，並且經由伺服器自動設定各個AP之ACL(Access Control List)，實現即時設定快速打擊。

# 研 究 目 的

建立AP熱點防禦，避免IoT危害。以封包分析數據找出異常IoT bot，立即防禦，以達成以下目的：

**一、剖析DDoS特性及歸納因應對策**

**二、建置WIFI AP及防禦平台**

**三、分析相關封包紀錄，辨識異常**

**四、即時設定ACL，無縫防禦**

# 研 究 設 備 與 器 材

表1 使用設備器材表

網路伺服器-後端分析

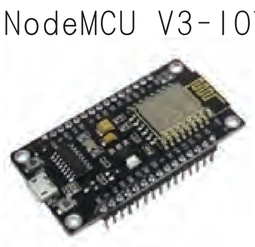
平板電腦-IOT

筆記型電腦-前端呈現

Raspberry Pi3B-AP

USB WiFi Adapter

NodeMCU V3-IOT



智慧手機-IOT

Smart7688-IOT

WiFi無線基地

表2 相關軟體表

| 相關軟體表 | | |
|---|---|---|
| 項次 | 軟體名稱 | 規格用途 |
| 1 | PHP+MySQL 實驗平台 | 數據分析、圖表繪製之平台 |
| 2 | FreeBSD RPI | Ap 熱點建立 |
| 3 | Mirai Sorce code | Mirai 主控及值入程式 |
| 4 | Adobe Dreamweaver | 演算分析及繪圖程式編寫 |
| 5 | Notepad++ | 程式碼及資料編修 |
| 6 | Packet Tracer | 防禦架構繪製，測試 |
| 7 | Putty | SSH、Telnet 連線操作 |

# 研 究 過 程 或 方 法

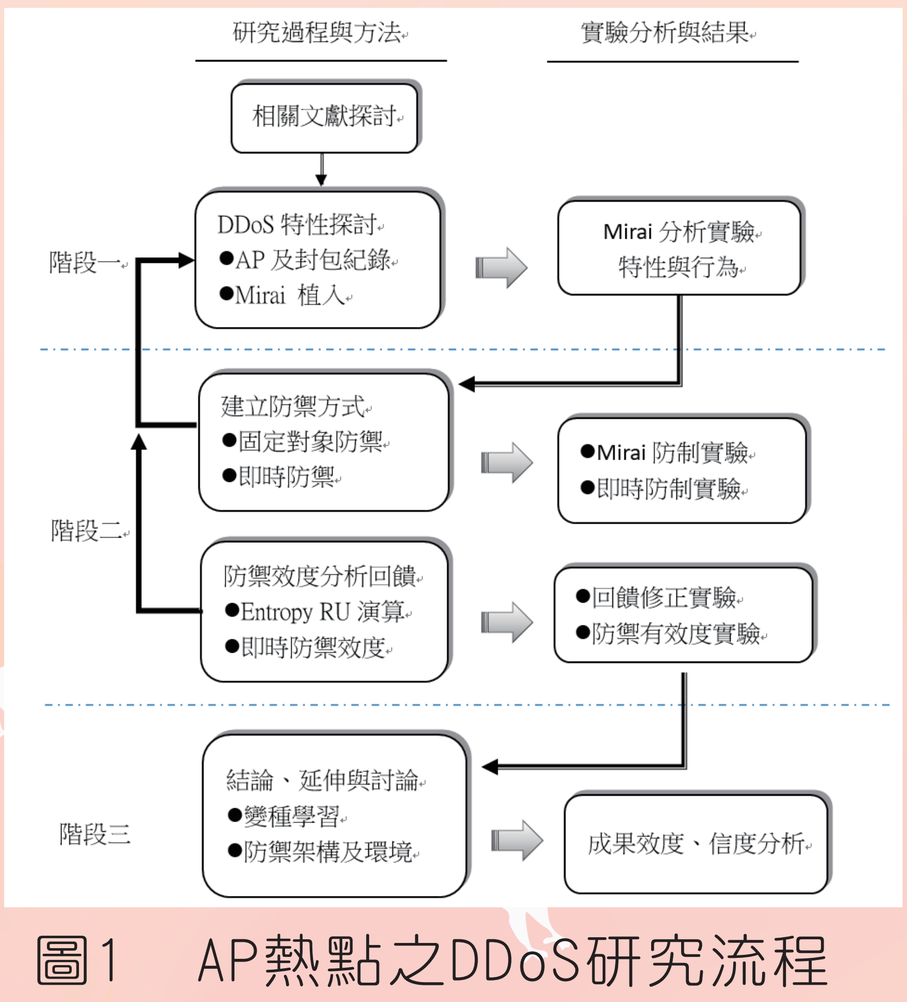



圖1　AP熱點之DDoS研究流程

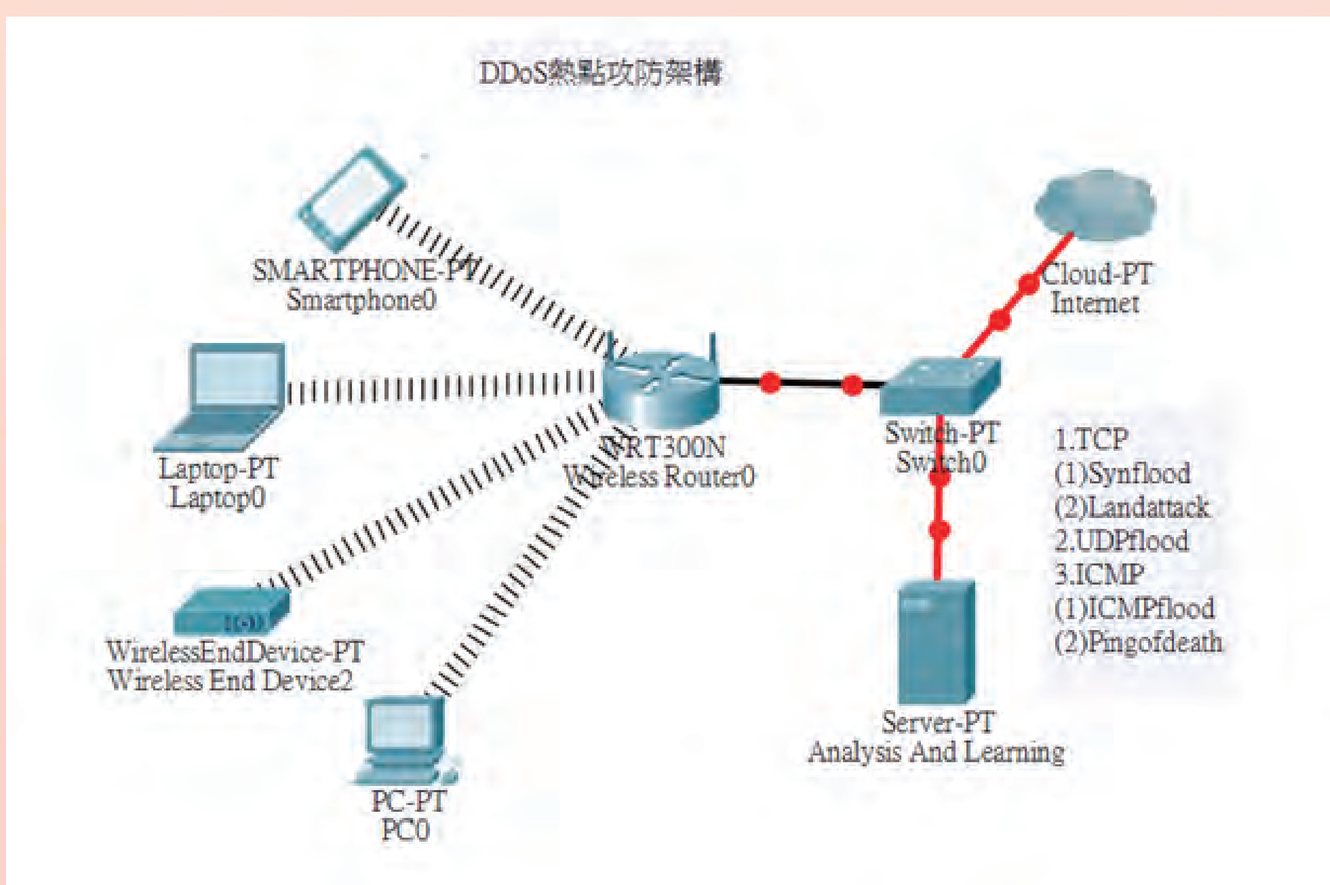



圖2　AP熱點之DDoS來源防禦架構

# 一、DDoS及Mirai特性探討

## （一）DDoS攻擊特性

DDoS 攻擊目前最常發生在OSI第 3（網路層）、 4(傳輸層) 、 6(展示層) 和 7 層(應用層)，常見攻擊說明如表2:

在不同層的攻擊類型，是使用不同的技術分析及防禦。在第6、7層的DDos攻擊，在偵測與防禦上較困難。第3、4層的攻擊是DDos較常見的類型，都是以大流量來癱瘓伺服器、防火牆、負載平衡器等，這種攻擊的特徵較明確，且容易被檢測，實作上較具可行性。

表2 DDoS 攻擊方式表

| 編號 | OSI 層別 | 攻擊方式 |
|---|---|---|
| 1 | 網路層 | flood UDP reply attack、ICMP flood |
| 2 | 傳輸層 | SYN flood |
| 3 | 展示層 | SSL session state attack、SSL negotiation attack、SSL vulnerability exploitation and compliance attack… |
| 4 | 應用層 | SQL injection、Encrypted web attack、Encoding and Evasion、HTTPS/HTTP Floods… |

## （二）Mirai Botnet攻擊特性

1. Mirai攻擊種類

從Mirai-Source-Code/mirai/bot/attack.h內定義了攻擊種類如下表：

表3 Mirai Botnet的攻擊種類

| 編號 | 名稱 | 攻擊方式 | 說明 |
|---|---|---|---|
| 1 | UDP | Straight up UDP flood | *大量發送 UDP 封包 |
| 2 | VSE | Valve Source Engine query flood | *遊戲引擎平台 UDP 攻擊 |
| 3 | DNS | DNS water torture | *不存在域名的大量查詢 |
| 4 | SYN | SYN flood with options | TCP 交握 SYN 攻擊 |
| 5 | ACK | ACK flood | TCP 交握 ACK 攻擊 |
| 6 | STOMP | ACK flood to bypass mitigation devices | TCP 交握 ACK 攻擊，跳過緩衝設備 |
| 7 | GREIP | GRE IP flood | IP 通用路由封包攻擊 |
| 8 | GREETH | GRE Ethernet flood | 乙太通用路由封包攻擊 |
| 9 | UDP_PLAIN | Plain UDP flood optimized for speed | 無檔頭 UDP 封包攻擊 |
| 10 | HTTP | HTTP layer 7 flood | HTTP 網頁服務政擊 |

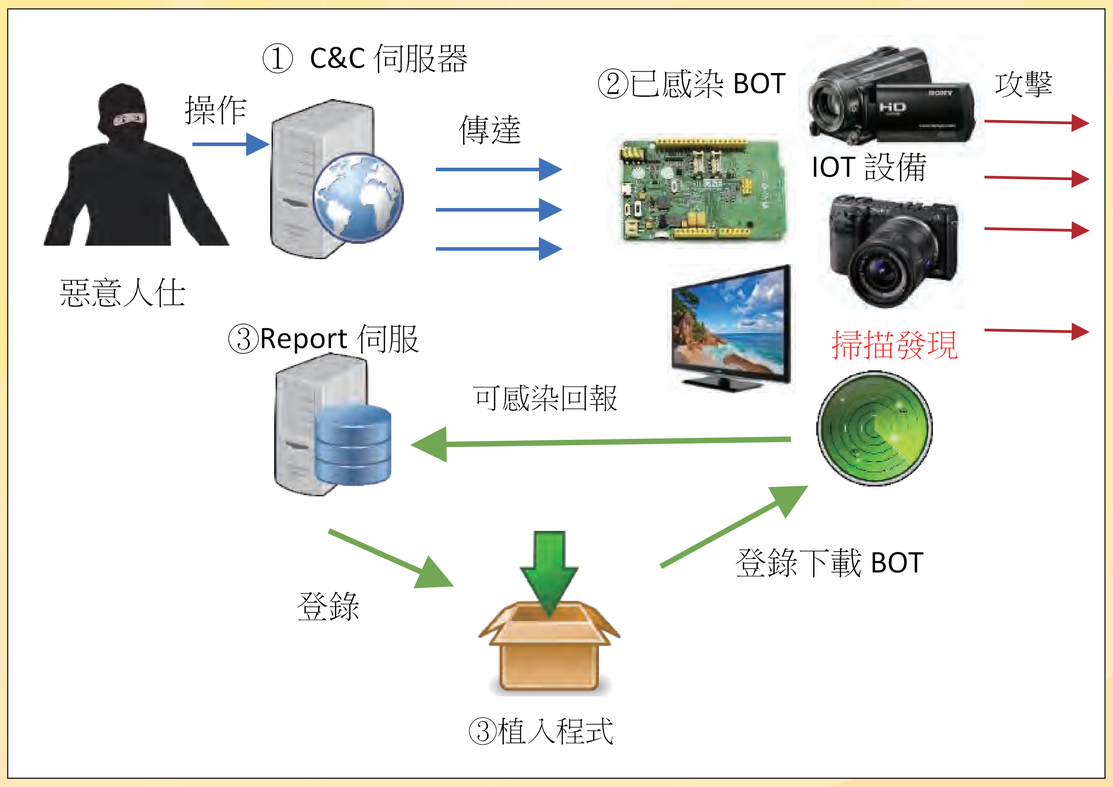



圖3 Mirai Botnet攻擊方式圖

2. Mirai Botnet攻擊方式

Mirai Botnet是惡意人仕通過①C&C伺服器向②Bot發送攻擊指令，以形成C&C為中心的殭屍網路。同時，Bot會掃描可感染的設備，將找到的設備信息發送給Report伺服器，由③down-loader植入Bot程序，如圖3。

# 二、Mirai 攻擊行為實驗

Shannon Entropy(熵)是表示多事件時之不確定性，如果值愈大不確定性愈高。對於網路的異常攻擊其特性正好相反，極具規則性、連續性，因此算出來Entropy愈低，代表愈可能是異常攻擊。

## (一)Entropy RU：相對不確性理論引用。

1. 有限的樣本時，熵H(X)的公式，P(X)為個別機率量，I(X)為取對數之機率變量

$$H(X) = \sum_{i \in X} P(x_i)\, I(x_i)$$

$$H(X) = -\sum_{i \in X} P(x_i) \log_b P(x_i)$$

$$RU(X) = \frac{H(X)}{H_{max}(X)} = \frac{H(X)}{\log min\{N_x | m\}}$$

2. 熵的相對不確定(relative uncertainty):RU值≦1, 比對時較明確，如上RU公式。

3. 攻擊封包RU分析：如表4, RU=0. 70387 < 0. 9464，因為值低則最大項異常可能性高，其演算程式如圖4。

```
$counts=array(40,16,9,7,2,2,2,2,2,1);//存取次數
$num=count($counts);          //總項目
$sum=array_sum($counts);;      //次數總合
foreach($counts as $ci){
    $p=$ci/$sum;              //個別機率
    $en_all[]=$p * -log( $p,2);//個別熵 P(x)*I(x)
}
$entropy=array_sum($en_all);     //計算 Entroy
$RU=$entropy/log($num,2);         //計算 RU
```

圖4 RU演算程式

表4 異常封包RU值

| 編號 | 次數 | 機率 P | P(x)*I(x) | RU |
|---|---|---|---|---|
| 1 | 40 | 0.372093 | 0.5307032 | 0.70387 |
| 2 | 16 | 0.209302 | 0.4722572 | 0.81543 |
| 3 | 9 | 0.162791 | 0.4263342 | 0.86668 |
| 4 | 7 | 0.046512 | 0.2058728 | 0.89857 |
| 5 | 2 | 0.046512 | 0.2058728 | 0.98661 |
| 6 | 2 | 0.046512 | 0.2058728 | 0.98239 |
| 7 | 2 | 0.046512 | 0.2058728 | : |
| 8 | 2 | 0.046512 | 0.2058728 | : |
| 9 | 2 | 0.023256 | 0.1261922 | : |
| 10 | 1 | 0.01205 | 0.07681 | : |
| 總合 | 83 | 1 | 2.5848506 | <-Entropy |

## (二) 分析架構與平台

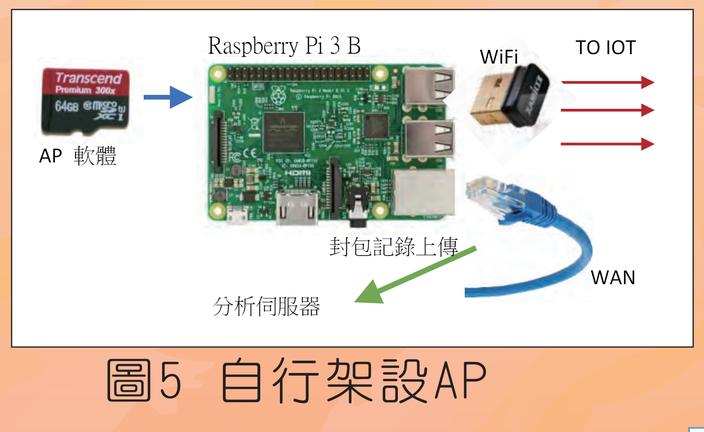



圖5 自行架設AP


圖6 自行架設AP運作及紀錄

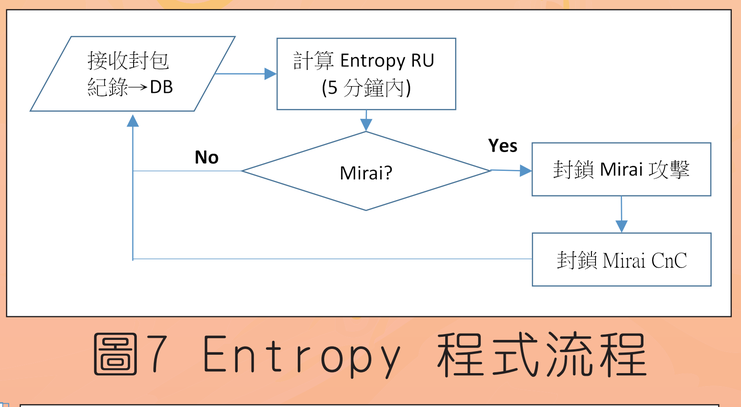



圖7 Entropy 程式流程

## (三) Mirai 攻擊分析：

1. 封包特徵：每一連線封包是由協定、來源位址、目的位址、來源埠、目的埠、及封包大小等基本六項


| pl_id | pl_out_time | pl_time | pl_rule | pl_action | pl_direct | pl_if | pl_src_addr | pl_src_port | pl_dst_addr | pl_dst_port | pl_proto | pl_count | pl_len |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1363132082 | 1363131994.440217 | 33 | 1 | 1 | em0 | 2736038777 | 1112 | 1876774285 | 40038 | 0 | 1 | 0 |
| 2 | 1363132082 | 1363132000.003976 | 33 | 1 | 1 | em0 | 2736038687 | 56180 | 1249714045 | 5222 | 0 | 1 | 0 |
| 3 | 1363132440 | 1363132047.852354 | 33 | 1 | 1 | em0 | 2736038777 | 1118 | 2735596250 | 3939 | 0 | 3 | 0 |
| 4 | 1363132440 | 1363132052.596841 | 33 | 1 | 1 | em0 | 2736038773 | 60164 | 1136901070 | 5050 | 0 | 1 | 0 |
| 5 | 1363132440 | 1363132068.460351 | 33 | 1 | 1 | em0 | 2736038773 | 60182 | 1136900797 | 5050 | 0 | 1 | 0 |
| 6 | 1363132440 | 1363132120.184468 | 33 | 1 | 1 | em0 | 2736038687 | 49171 | 1249714045 | 5222 | 0 | 2 | 0 |
| 7 | 1363132440 | 1363132170.073283 | 33 | 1 | 1 | em0 | 2736038880 | 61505 | 2736038705 | 25864 | 0 | 1 | 0 |
| 8 | 1363132440 | 1363132175.153073 | 33 | 1 | 1 | em0 | 2736038880 | 61511 | 2736038705 | 15415 | 0 | 1 | 0 |
| 9 | 1363132440 | 1363132177.403386 | 33 | 1 | 1 | em0 | 2736038880 | 61514 | 2736038705 | 56928 | 0 | 1 | 0 |
| 10 | 1363132440 | 1363132178.867907 | 33 | 1 | 1 | em0 | 2736038880 | 61516 | 2736038705 | 59165 | 0 | 1 | 0 |

圖8 DB封包原始記錄


| No. | Time | Duration | Source addr | Protocol | Counts | Commet |
|---|---|---|---|---|---|---|
| 5616 | Jan 19 00:00:00 | Day | 10.1.1.10 | tcp | 90 | |
| 5614 | Jan 19 00:00:00 | Day | 10.1.1.10 | udp | 61 | |
| 5632 | Jan 25 00:00:00 | Day | 10.1.1.10 | tcp | 34 | |
| 5606 | Jan 11 00:00:00 | Day | 10.1.1.10 | tcp | 11 | |
| 5618 | Jan 19 00:00:00 | Day | 10.1.1.10 | other | 10 | |
| 5610 | Jan 11 00:00:00 | Day | 10.1.1.10 | udp | 2 | |
| 5630 | Jan 25 00:00:00 | Day | 10.1.1.10 | icmp | 2 | |
| 5584 | Jan 08 00:00:00 | Day | 10.0.1.10 | udp | 2 | |
| 5588 | Jan 08 00:00:00 | Day | 10.10.10.10 | icmp | 2 | |

圖9 伺服器上之封包記錄統計

2. 對於IOT的攻擊是由內而外攻擊，可以只考慮對外連線封包。並且是由駭客下命令攻擊特定幾個目標，可以針對目標IP分析（在時段內）。

3. Mirai攻擊方式種類有10項，並經由CnC伺服器命令其中一項攻擊類型。

## 三、Mirai即時防制

對於確定受感染之IOT Bot，可在AP端防制封鎖其攻擊封包。

(一)封鎖攻擊封包

表5 - tcp、udp、icmp封鎖攻擊之規則

```
block in on { $int_if } inet proto tcp from $lan_net to any flags S/SA no state
block in on { $int_if } inet proto udp from $lan_net to any no state
block in on { $int_if } inet proto icmp from $lan_net to any no state
```

(二)CnC伺服器命令封鎖

```
315             // Try to read in buffer from CNC
316             errno = 0;
317             n = recv(fd_serv, rdbuf, len, MSG_NOSIGNAL | MSG_PEEK);
318             if (n == -1)
319             {
320                 if (errno == EWOULDBLOCK || errno == EAGAIN || errno == EINTR)
321                     continue;
322                 else
323                     n = 0;
324             }
```

Bot 接收 CnC 命令

圖10 IOT Bot向CnC命令

由mirai/bot/main.c，可知IOT Bot是主動向CnC溝通，並且透過telnet(port 23)，可關閉Bot 對外的Tcp port 23，以斷絕後續攻擊命令。

# 研 究 結 果

## 一、Mirai攻擊之偵測及防禦負荷：

主要瓶頸在於AP上SD卡封包紀錄寫入速度，隨封包數增加而延遲處理時間。

IOT BOT由CnC命令設定每次攻擊方式及數量， AP端對頻寬造成重大負擔。

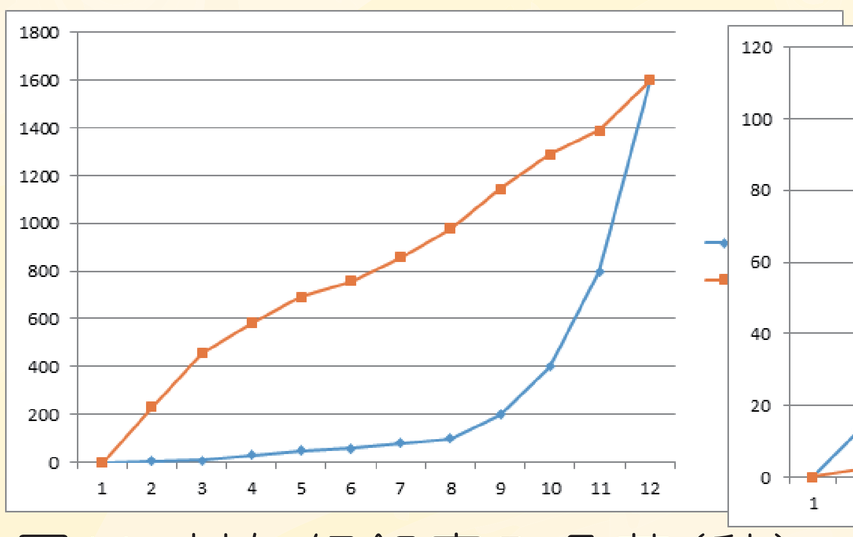



圖11 封包紀錄寫入負荷(秒)

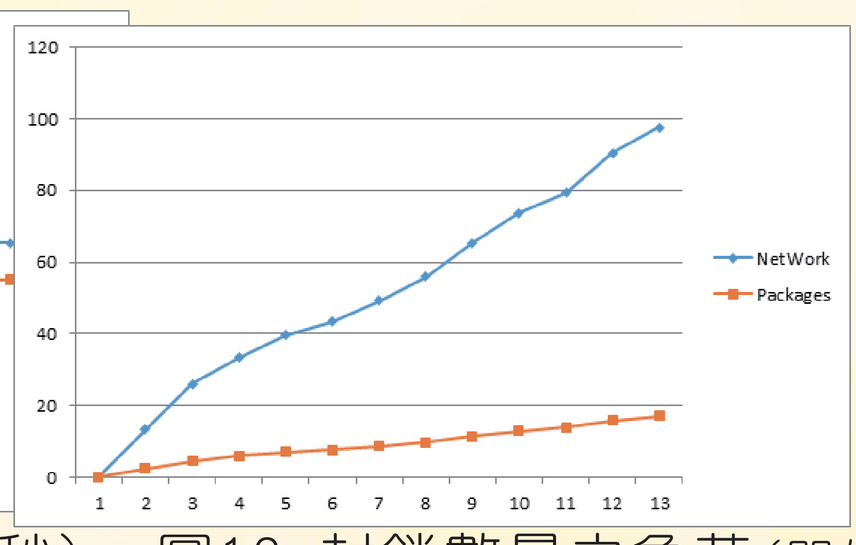



圖12 封鎖數量之負荷(單位百筆)

## 二、Mirai 即時防制成果

(一)AP防制處理反應時間：以確認感染之Bot，第一筆攻擊算起。 表6 處理反應時間

| 編號 | 攻擊方式 | 平均反應時間(分) |
|---|---|---|
| 1 | UDP | 8.63 |
| 2 | VSE | 9.31 |
| 3 | DNS | 7.42 |
| 4 | SYN | 12.23 |
| 5 | ACK | 11.74 |
| 6 | STOMP | 10.63 |
| 7 | GREIP | 12.82 |
| 8 | GREETH | 9.82 |
| 9 | UDP_PLAIN | 6.44 |
| 10 | HTTP | 10.91 |

(二)封鎖後AP及主機之網路負荷

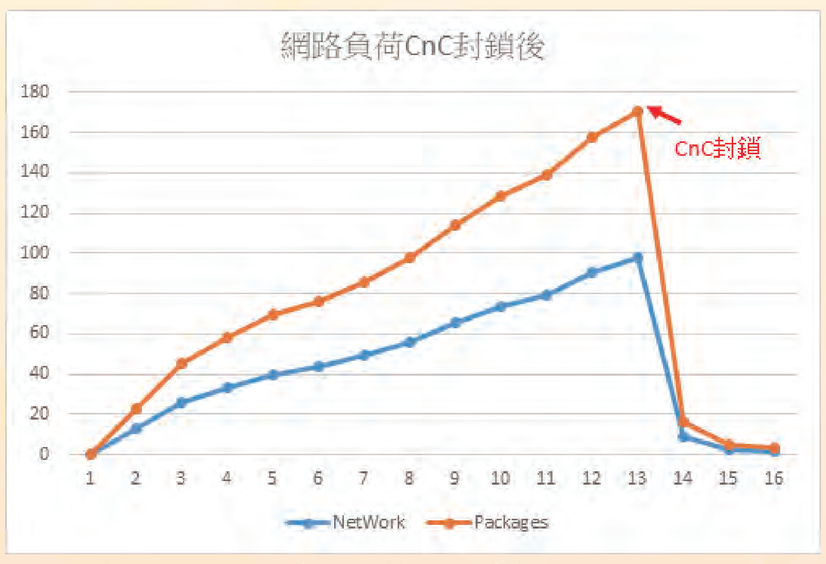



圖13 AP網路負荷(單位十筆)

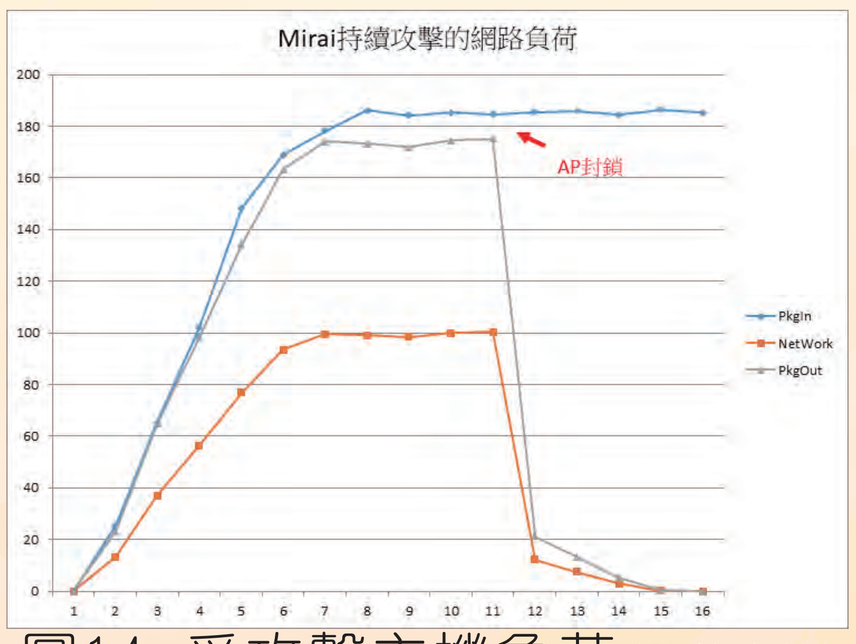



圖14 受攻擊主機負荷(單位十筆)

Entropy RU 偵測Mirai攻擊的效率極高，Mirai攻擊方式雖然有10種，但每次進行單純的一種攻擊，並且時間連續以形成攻擊，所計算Entropy RU明顯偏極低，因此偵測率更高。

# 討 論

Mirai bonet如同其他DDoS有其變種，以OMG為例，因為運作模式相似高，在UDP及TCP的Entropy RU判斷上亦可達成偵測：

**一、DDoS的防禦：**所有DDoS均有大量攻擊之特性，計算Entropy RU亦有極高準確度。

表7 Entropy RU偵測之準確度(The first Attack)

| k | %True Positives | %False Negatives | %True Negatives | %True Positives | %Overall Accuracy |
|---|---|---|---|---|---|
| 3 | 99.65 | 0.35 | 100 | 0 | 99.82 |
| 5 | 99.65 | 0.35 | 100 | 0 | 99.82 |
| 7 | 99.59 | 0.41 | 99.2 | 0.8 | 99.40 |

摘自：用三階層式模組於偵測伺服器上的DDoS攻擊

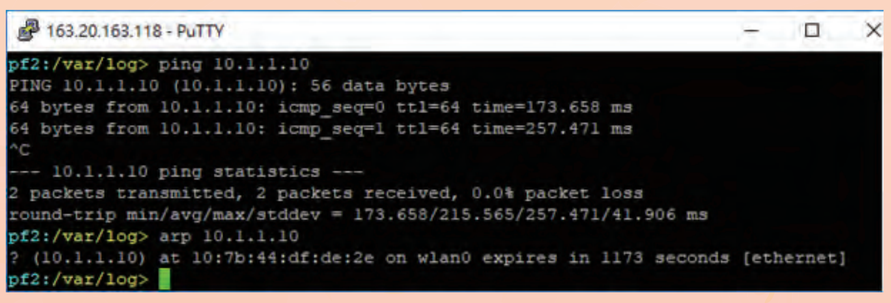



圖15 IOT設備之Mac Address

**二、OMG Mirai bonet變種偵測：**OMG新變種會在加入防火牆兩組規則，以允許兩個隨機埠口上的流量通過，以提供為惡意人士使用，其他攻擊與原來Mirai相同。

**三、MAC Address精確定址：**因為AP運作於IOT相同網段，使用MAC定址可更精確掌握內部網段的IOT bot。

# 結 論

AP為IOT設備的前端上網設備，可做為DDoS攻擊第一道防線。以下是我們的各項結論：

**一、熱點AP可有效防制Mirai攻擊：**可偵測出Mirai bot，並即時封鎖防制，從來源隔離。

**二、封鎖Mirai CnC可阻止後續攻擊：**IOT Bot定時向CnC取得攻擊命令，可阻止命令取得。

**三、分析UDP及TCP封包同時可防制其它DDoS攻擊：**一般DDoS攻擊與Mirai相似，以大量攻擊為主，可透過類似方式防禦。

**四、IOT設備的安全預防：**定期更新IOT韌體，可增加系統安全。使用前需變更內部預設密碼，防止以密碼字典登入。