# Netflow 結合資安院黑名單建立全網域受感染者資料庫之實作

梁明章
財團法人國家實驗研究院國家高速網路與計算中心
liangmc@niar.org.tw

## 摘要

本文將說明 TWAREN NOC 利用國家資通安全研究院的黑名單自動部署服務每天自動下載黑名單結合 NOC 的 Netflow 演算體系查找有關的通聯記錄保存相關資料，便於長期分析可能受害淪陷者，形成資料庫以供研究查詢。

**關鍵詞**：網路流量紀錄，Netflow，TWAREN，國家資通安全研究院，黑名單，受感染者資料庫。

## 1. 前言

民國113年，「通訊保障及監察法」[1]的修正條文14-1規定「電信事業及設置公眾電信網路者經通訊監察之建置機關指定者，有保存及協助執行調取網路流量紀錄之義務」。而法條所稱「網路流量紀錄」，定義為「指用戶或電信使用人使用電信服務後，公眾電信網路所產生之通訊設備識別資料、網際網路位址與位置資訊、通信時間、連線用量與封包數量、網域名稱、應用服務類型及協定等未涉及通訊內容之紀錄。」，在實務上，Netflow 正是符合此定義的網路流量紀錄，Netflow 本來是 Cisco 公司提出的名詞，後來其他廠家設備另有 Cflow、Jflow 等等的格式與名稱，後來國際組織 IETF 還根據 Cisco Netflow Version 9[2] 擴充後制定了國際標準協議 IPFIX (Internet Protocol Flow Information Export)[3]，但都不如 Netflow 這稱呼來得通用且廣被人知，所以本文也以 Netflow 來代稱網路通信記錄，TWAREN 因為有多家廠商設備，其實並非全是 Netflow，但各家基本上會相容於 IPFIX 的規範。

TWAREN 從建立之始二十多年來從未中斷過 Netflow 的維運，堅持保存 Netflow 相關的維運技術並持續開發 Netflow 的應用，由於 Netflow 需要高昂的成本，原本只用來做使用分析統計報表、用量排行查詢等等，投資報酬率上效益就難以匹配成本，可是沒有 Netflow，前述功能就做不出來，前些年大資料平台開源之後，利用 ELK 的免費授權功能組建了一座大資料平台，更是擴大了 Netflow 的投入成本，幸而長官持續支持，才努力堅持下來。我們很快就利用自行開發的高速統計彙整排序程式結合大資料平台，終於達成骨幹等級大規模異常行為即時偵測的里程碑，後續結合路由器 FlowSpec (BGP Flow Specification)[4]功能達成可即時自動部署阻擋 IP 給骨幹路由器拋棄異常來源者 IP 的封包，達成快速應變的效果，在資安越來越受重視的年代，NOC 具備快速應變的骨幹大規模攻擊防堵機制也是有保護連線單位意義的。

而113年通保法的修正條文更是賦予我們更大的支持，因為 Netflow 的保存與提供查詢調取已變成是 NOC 必要的任務。

### 1.1 牛刀難以殺雞的問題

雖然我們已經達成即時偵測大規模異常惡意者並自動攔截的機制以保護骨幹，降低惡意封包到達使用者單位的衝擊，但這是一把牛刀，面對骨幹持續產生的海量 Netflow，想要即時運算偵測小規模異常，即使大資料平台再大千百倍也不可能達到，於是我們退而求其次，針對小規模的異常偵測不再追求「即時」，改成一天一次看看，結果也很艱難，目前每天的 Netflow 總筆數已經到了一百多億筆的量，若是對平台做簡單搜尋還沒問題，但若是要求平台進行統計與彙整，例如最簡單的，計算全日每個 IP 的用量，就可能會讓平台記憶體耗盡而崩潰，或是對一個月做動作，那就要面對三四千億筆資料的統計與彙整，大致上都會崩潰，主要原因是 Netflow 這資料的特性就是隨時隨地都在持續產生，目前的產生速度是平均每秒約十六萬筆，亦即大資料平台必須持續做資料索引與資料分散儲存與複製等工作，光是資料新增就持續占用不少記憶體、網路頻寬與演算力，還要同時應付查詢就會相當吃力，因此我們很難直接利用平台的高級或複雜的組合查詢能力來幫我們分析查找小規模的異常。

### 1.2 尋找受感染者的動機與研究

在現今資安設備越來越普及到各校的情況下，大規模異常對各校來說，除了可能吃完專線頻寬或是擊跨防火牆效能，最多就是阻斷公開的服務網站，對於一般使用者來說，不會造成直接的打擊，反而是被各種社交工程入侵的單位內鬼對各單位的網路環境威脅或許更大，所以我們一直摸索從骨幹 Netflow 查找這種已受感染的內鬼，或稱淪陷者、肉雞等等的存在，在爆發前先通知學校單位拔除掉內鬼就能降低未來風險。

為了協助尋找單位內鬼，近年來我們聚焦某單位的 Netflow 進行細緻的 Netflow 分析演算，因為面對的數量還可以承受，也有一些成果，過往 TANet 研討會上也報告過，但是 Netflow 的資料並未包含封包內容，我們只能從 Port / Packet / Octet / Protocol 等等去綜合判斷，即使有推測結論，卻無法驗證對錯，最嚴重的問題莫過於，僅靠 Netflow，很多情境下我們甚至無法確認惡意來源 IP 是真實或假冒的，這個地基站不住，往上搭疊的推理就如空中樓閣，無法驗證。

因此，我們發現若要繼續走下去，我們需要確認的黑名單，或是可以驗證的手段，網路上雖然有著名的網站能做查詢，但是我們系統運行下會產生很多嫌疑犯，沒有那個人力去天天一個一

個人工比對，我們需要一個能自動化連結 Netflow 運算體系的方法，而國家資通安全研究院正好推出一個自動化服務可以提供我們進一步的希望。

## 2. 實作方法說明

國家資通安全研究院已經將黑名單的提供方式做成可由使用者自動去下載的服務，詳情請見「黑名單自動部署服務系統」[5]，經過申請核可後，就能獲得專用的 URL 以自動下載黑名單，結合 NOC 的 Netflow 運算體系進行後續處理，一切皆可撰寫程式自動化運行。

### 2.1 第一階段：根據 IoC 清單查詢 ELK 取回相關 Netflow

黑名單下載可選擇 IP 或網域名稱兩種格式，以我們的需求而言自然是選擇 IP 清單，內容是每行一個 IP 位址的文字檔，由於國家資通安全研究院聲明每天更新一次黑名單資料，但並未明說更新時刻，故我們每天於午夜換日前一分鐘下載黑名單，然後逐行讀取每個 IP，每個 IP 皆以下列範例對 ELK 進行查詢。

```
curl "elk:port/twaren,tanet/_search" -d "
{
  \"size\":10000, #最多取回一萬筆 Netflow
  \"track_total_hits\":true, #請 ELK 統計符合的總數
  \"query\":{
    \"bool\":{
      \"should\":[
        {\"term\":{\"sourceIPAddress\":\"$1\"}},
        {\"term\":{\"destinationIPAddress\":\"$1\"}}
      ],
      \"minimum_should_match\":1
    }
  }
}"
```

上述查詢參數為何僅取回一萬筆即可，主要是 Netflow 非常多的 IoC 通常是被駭客命令進行廣域掃描探測或是參與 DDoS/DoS 的「棄子」(因為這樣顯眼的行為容易被網管發現並拔除)，被波及的大量目標對我們來說並無多大的紀錄與研究價值，因此僅取回 ELK 單次查詢預設的回覆筆數上限一萬筆，作為分析該 IoC 行為特徵的樣本材料即可。

至於「track_total_hits : true」是告訴 ELK 雖然我們最多只要一萬筆回覆，但還是請 ELK 把搜尋做完並統計最終符合條件的總筆數，因為我們要紀錄該 IoC 造成的規模。

目前黑名單已經膨脹到三千筆左右，未免造成 ELK 崩潰，採一次一個 IoC 循序方式查詢而非併發同時查詢，以 TWAREN NOC 的 ELK 平台算力，深夜約需半小時，白天則要超過一小時才能完成全清單查詢，主要看 ELK 當時的負擔，我們的 ELK 在白天要承擔 TWAREN 每秒 9 萬筆、TANet 出國口每秒5千筆、FOX 交換中心每秒四萬筆，加上 Syslog、防火牆、DNS、資安設備等一堆即時紀錄，相當於平均每秒都要吃進十六萬筆進行索引並儲存，可以說全系統壓力很重。還好國家資通安全研究院也有考量到各單位使用黑名單的設備能量，有說明目前會盡量把黑名單去蕪存菁，維護在五千筆之內。

根據我們目前的運行紀錄，發現近一周這三千左右的黑名單，大約有40%到50%在 TWAREN 全域以及 TANet 的出國口都查不到傳輸紀錄，可能是真的沒有波及到學術網路，但也有可能 IoC 並不在國外，而是在國內 ISP，走區網中心多點互連線路與 TANet 傳輸的就不會經過出國口，因此查不到。

### 2.2 第二階段：轉換 Netflow 資料為適合未來處理的新 Netflow 並寫回 ELK 新 Index


```
{   轉換前
  "exporterName": "192.192.68.33",
  "sourceIPAddress": "163.26.171.228",
  "sourceIPv4Address": "163.26.171.228",
  "destinationIPAddress": "134.122.129.10",
  "destinationIPv4Address": "134.122.129.10",
  "ipClassOfService": 0,
  "protocolIdentifier": 6,
  "sourceTransportPort": 13216,
  "destinationTransportPort": 45,
  "icmpTypeCodeIPv4": 0,
  "ingressInterface": 683,
  "vlanId": 4,
  "sourceIPv4PrefixLength": 14,
  "destinationIPv4PrefixLength": 24,
  "bgpSourceAsNumber": 1659,
  "bgpDestinationAsNumber": 152194,
  "ipNextHopIPv4Address": "65.49.108.1",
  "tcpControlBits": 2,
  "egressInterface": 724,
  "minimumTTL": 122,
  "maximumTTL": 122,
  "flowEndReason": 2,
  "ipVer": 4,
  "bgpNextHopIPv4Address": "65.49.108.1",
  "flowDirection": 255,
  "dot1qVlanId": 0,
  "dot1qCustomerVlanId": 0,
  "fragmentIdentification": 0,
  "octetDeltaCount": 52,
  "packetDeltaCount": 1,
  "flowStartMilliseconds": 1757057835008,
  "_timestamp": 1757057835008,
  "flowEndMilliseconds": 1757057835008,
  "srcGeoip": {
    "country_code": "TW",
    "city": "Tainan_City",
    "longitude": 120.214798,
    "latitude": 22.991699,
    "coordinates": [120.214798, 22.991699]
  },
  "dstGeoip": {
    "country_code": "HK",
    "city": "Shatin",
    "longitude": 114.177902,
    "latitude": 22.3829,
    "coordinates": [114.177902, 22.3829]
  },
  "exporterIP": "192.192.68.33",
  "ipVersion": 4
}
```

```
{   轉換後
   "IocIP": "134.122.129.10",
   "RemoteIP": "163.26.171.228",
   "Exporter": "192.192.68.33",
   "TotalFlows": 1207,
   "RcvPackets": 1,
   "RcvOctets": 52,
   "SendPackets": 0,
   "SendOctets": 0,
   "IocPort": 45,
   "RemotePort": 13216,
   "Direction": "Rcv",
   "TimeStamp": 1757057835008,
   "IocAsName": "UnKnow",
   "IocAsNumber": 21122,
   "RemoteAsName": "UnKnow",
   "RemoteAsNumber": 1659,
   "IocGeoip": {
      "country_code": "HK",
      "city": "Shatin",
      "longitude": 114.1779,
      "latitude": 22.3829,
      "coordinates": [114.1779, 22.3829]
   },
   "RemoteGeoip": {
      "country_code": "TW",
      "city": "Tainan_City",
      "longitude": 120.2148,
      "latitude": 22.9917,
      "coordinates": [120.2148, 22.9917]
   },
   "TcpControlBits": 2,
   "Protocol": 6,
   "IpVersion": 4,
   "FromIndex": "tanet-2025.09.05",
   "Date": "2025-09-05"
}
```


由於每筆 Netflow 都是單向的，IoC 有可能是來源端也可能是目的端，因此原始 Netflow 資料在大平台系統做彙整查詢時，兩個方向混雜會造成困擾，連圖表跟即時儀表板都不好做。為了方便未來長時間的重複演算，以及便於 NOC 即時監控儀表板的製作，我們對每個 IoC 都建立一個主物件，處理從 ELK 取回的 Netflow 做轉換處理，以 IoC 的 IP 位址為主視角，將對端稱為 RemoteIP，Send 就表示 IoC 傳出給 RemoteIP，Rcv 就表示 RemoteIP 傳輸給 IoC，等於固定好雙方的座位，如此在使用大資料平台的查詢語言時，可以大大簡化複雜性，省掉中間轉換的過渡資料陣列，降低平台 JVM 崩潰的風險，在監控儀表板上也容易顯示，上面分別顯示原本 TWAREN 儲存的 Netflow 以及轉換後的 Netflow。

轉換後的 Netflow 納入 IoC 主物件中做成 Netflow 陣列，然後我們將每個 IoC 的新 Netflow 陣列都輸出為 JSON 格式，為了使用 Bulk RESTful API 匯進 ELK，程式先輸出成一個超大 JSON 檔案，雖然 ELK 的 Bulk 一次最大極限是100MB，但如此大包對於經手處理的節點們的 JVM 會造成很大負擔，網路上常見建議為10MB 少一些，根據每筆 JSON 的大小跟10MB 計算即可推估大約一個 Bulk 包多少筆 JSON 會比較適當，利用 Linux 常見的「split -l」指令（參數-l 表示幾行切一包）進行切割，然後上傳至 ELK。

由於我們是每天一次處理昨天的資料，且產生一個新 Index（ELK 的名詞，類似於關聯式資料庫的 Table 資料表的地位），此 Index 匯完資料後就不會再有更改，因此可對 Index 做下列設定，要求 Index 不要一邊收資料一邊做索引樹，建立為手動索引狀態，可等資料全匯入後，再下令建立索引樹，如此可盡量減少索引樹的數量，增加未來查詢的速度。

```
curl -s -XPUT "elk:port/IoC/_settings" -d '{
      "index.refresh_interval": "-1",
      "index.translog.flush_threshold_size": "512mb"
 }'
資料全匯入後，再下令索引
curl -s -XPUT "elk:port/IoC/_refresh"
```

到此，我們就把 IoC 有關的 Netflow 細節留存到 ELK 中，這些資料量相比整體骨幹的量只是滄海一粟，即使用來做超長時區間的查詢也不必擔心 ELK 崩潰。

### 2.3 第三階段處理：濾除廣域探測類及服務阻斷攻擊類的 IoC

程式運行到此，所有 IoC 物件都包含修改過的 Netflow 陣列，程式會開始針對幾個我們感興趣的樣態進行分析，我們先把「TotalFlows」遠超過一萬以上的 IoC 註記不進行下階段運算，雖然粗略，但目的主要是減少後續算力需求，照目前觀察的數據，可減少約4%~7%的 IoC，加上前面提到45%查無資料的 IoC，已經可略過一半的 IoC。

剩下的 IoC 就是全日下來被記錄到的 Netflow 不足一萬筆，但是我們仍須再次濾掉廣域探測的，因為有可能某些 IoC 只是因為今天的掃瞄範圍僅波及一點點學術網域，所以 Netflow 量少，因此我們再對每個 IoC 的 Netflow 陣列以 RemoteIP 做排序，如果每個 RemoteIP 僅有一兩筆，加上 Port / Packet / Octet 結合判斷可推測為單純的掃描探測，可註記為廣域，但如果都是 RemoteIP 單向丟一兩封包給 IoC，那就有很大問題，或許是個肉雞眾多的大家族在每天請安簽到。那這個 IoC 就需要關注，對他簽到的 RemoteIP 也必須註記關注。

當我們再度清除一次廣域 IoC 之後，再濾除阻斷服務攻擊的 IoC，這倒是挺好判斷，本文不再贅述。接著進入下階段。

### 2.4 第四階段處理：處理分析 RemoteIP 端

前文已說過，本文關心的重點其實在於已淪陷的受害者，也就是 RemoteIP 這端，然而 RemoteIP 雖然有被 IoC 碰觸，卻未必淪陷，是否有問題，需要長期觀察，因此我們也需要對這些疑似受害者留存有利於搜尋的資料。

我們的程式會處理每個 IoC 的 RemoteIP，針對每個 RemoteIP 建立主物件，反向紀錄與他有牽扯的 IoC，相關 Netflow 也把參考掛到本身的 Netflow 陣列內，不同 IoC 內重複出現的 RemoteIP 則併為同一個，由於 Netflow 已經在前面階段紀錄回 ELK，所以本階段不再把 RemoteIP 物件內的 Netflow 重複寫去 ELK，主要是紀錄彙整資訊，像是每個 RemoteIP 牽涉到的 IoC 陣列（一個 RemoteIP 可能被不只一個 IoC 碰過），以及 Port / Packet / Octet / Protocol 等等特徵。

最後，RemoteIP 物件陣列與 IoC 物件陣列（不含 Netflow，因為第二階段已記錄）都會寫去 ELK 的彙整 Index，作為長期演算資料來源。

這個程式每天清晨運行一次並記錄前述結果回大資料平台，再來我們以另外一隻程式每天一次針對前述幾階段紀錄資料做大範圍綜合分析，由於前述的 IoC-Netflow 與 IoC & RemoteIP 彙整 Index 都是屬於「小量」的資料，因此我們對 ELK 的查詢時區間可以含括極大的跨度，例如月或年，也不用擔心造成大平台的崩潰。

第一種演算，就是對 RemoteIP 的 Index 以月範圍統計每個 RemoteIP 的出現次數並排序，由於 RemoteIP 每天只會紀錄一篇，出現次數越高的 RemoteIP 就表示出現越多天，前文說過在第三階段已經略過廣域及阻斷類的彙整，因此這些 RemoteIP 通常不是被無辜波及的對象，而有可能是 IoC 刻意連線的對象，得出排序列表之後，程式再從排序最高的 RemoteIP 開始，一個個對 IoC-Netflow Index 做精準的細緻查詢彙整與綜合判斷。

假設我們現在針對 RemoteIP-1號做分析，我們以此 IP 為條件從 IoC-Netflow 查詢整個月的範圍，同時以 IoCIP 為第一層區分、IoC-Port / Remote-Port 為第二層區分，Packets & Octets 等作第三層統計，計算分類樣態有幾種，樣態種類越少代表每次行為相似度越高，然後判斷 Client / Server 方向（正常系統的 Socket 呼叫會給 Client 端配置高於32768的 Port 號碼，因此正經的 Server 端也不會選擇高於32768的 Port 號來開門），以方向來區分，若 IoC 端類似 Server 且 Port 趨同，則可能是中繼站，RemoteIP 是在向中繼站簽到。如果 RemoteIP 這端類似 Server 且 Port 趨同，那麼這可能是 IoC 在進行 APT 攻擊（進階持續性威脅 Advanced Persistent Threat），於是我們就能將 RemoteIP 的受害屬性加上分類，並且在分類上累加分數。同樣的，IoCIP 也有類似操作。這裡所形成的綜合判斷結果，會寫入綜合判斷結果的 Index，這個 Index 內每個 IoCIP 或 RemoteIP 都只會有一篇文件，跟前述的 IoCIP / RemoteIP 每天產生新文件的彙整 Index 不同，綜合判斷結果寫回 Index 時，會以更新文件的方式，保持同一個 IP 僅有一篇最新的有效文件，因為同一個 IP 的屬性資料分數會每天累加。

如此每天進行一次整月或數月或是整年範圍的 RemoteIP 分析後對關聯式資料庫那邊的屬性與分數做調整，分數越高的嫌疑度就更深，至於臨界值該定多少，我們還需要研究，因為現階段累積還太少。

這階段的程式我們會持續研究改善，添加更多分析演算，未來也可能有越來越多新的結果欄位被記錄到綜合判斷彙整 Index，然而變動的欄位也正是文件型資料庫所擅長的。

## 3. 結論與未來工作

本文使用國家資通安全研究院維護的黑名單自動部署服務自動下載 IoC 清單整合 NOC 開發多年的 Netflow 運算體系做受感染者評估運算，做為比較有可信度的惡意來源清單，可彌補我們以往光靠 Netflow 難以確認小規模型的惡意來源 IP 是否有意義這項弱點。

反過來看，國家資通安全研究院的黑名單來自各方資安單位與設備的通報，並不帶有受害者資訊，而 NOC 掌握網域通聯記錄（Netflow），卻可以利用這些點狀分布蒐集的黑名單，擴及到骨幹領域全面的查找以試圖找出全網已淪陷的受害者，若是政策與法律許可（我們不是法律專家，公開是否違法需研議），甚至可以開放網頁查詢，或是僅開放給各校網管來查詢（這或許比較可行），就能替國內學研單位拔除一些隱藏未爆發的受害者。

除了拔除未爆雷之外，未來研究或許能取得更進一步的成果，舉個例子，RemoteIP 每天簽到一下，不見得已淪陷，但如果同樣這群 RemoteIP 長期下來天天都簽到一下，那就頗有家族的可能性，掌握了這樣已淪陷的 RemoteIP 群，哪天突然整群都沒查到新的簽到記錄，我們甚至可以反過來清查這群 RemoteIP 的新連線紀錄找出新的同特徵交集 IP，說不定就是轉移過的新中繼點，再把新中繼點當作 IoC 再做前面的那些步驟，是否與舊行為類似，說不定就能比各通報網站更早發現新的 IoC。

還有許多點子，等著我們未來發掘，也是我們未來的工作，而先把可能有用的資料萃取保存下來，則是我們已經在做的工作，我們也期待未來有一天能夠開放綜合彙整 Index 給需要的網管人員查詢。

## 參考文獻

[1] 通訊保障及監察法, 全國法規資料庫, https://law.moj.gov.tw/LawClass/LawAll.aspx?pcode=K0060044

[2] Cisco Netflow Version 9, CISCO, https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html

[3] IP Flow Information Export (IPFIX), IETF, https://datatracker.ietf.org/wg/ipfix/documents/

[4] BGP Flow Specification, https://datatracker.ietf.org/doc/draft-ietf-idr-flowspec-v2/

[5] 黑名單自動部署服務系統, 國家資通安全研究院, https://ironcloak.nics.nat.gov.tw/index