基於網路攝影機與路由器之物聯網裝置惡意攻擊研究

A Study on Penetration Testing of IoT Devices Based on Webcam and Router

王建凱
國立屏東大學資訊科學系
屏東市民生路4-18號
dabledark@gmail.com

楊政興
國立屏東大學資訊科學系
屏東市民生路4-18號
chyang@mail.nptu.edu.tw

翁麒耀
國立屏東大學資訊科學系
屏東市民生路4-18號
cyweng@mail.nptu.edu.tw

摘要

近年來，通訊及網路的相關技術進步使得許多設備都開始具備連網功能，這些數以萬計的物聯網設備已經逐漸融入我們的生活之中成為不可或缺的一環，雖然物聯網設備的處理效能或是訊息傳遞速度比起個人電腦或是智慧手機都遜色不少，但其數量龐大且使用者購買後通常不會再去監控設備狀況，因此駭客也將目標轉向數以萬計的物聯網設備。

物聯網裝置本身由於硬體效能受到成本或是耗電功率等等考量，而無法擁有完善的保護機制，例如：強健的編碼系統、完整的驗證機制、威脅即時通報機制，所以攻擊者也逐漸將攻擊目標從難以攻破的企業主機轉向保護較弱且數量龐大的物聯網裝置，常見的攻擊有阻斷式攻擊、勒索軟體、植入惡意挖礦軟體及竊取隱私或有利資訊等，因此本研究將針對目前常見的物聯網裝置攻擊手法與植入惡意程式之滲透測試過程加以探討，並提出改善之建議藉此提醒社會大眾對於物聯網裝置之安全重要性。

關鍵詞: 物聯網、惡意軟體、滲透測試。

Abstract

In recent years, communication and network technology makes many devices have networking functions. Tens of thousands of Internet equipment has gradually come into our life and become a necessity. Even though the Internet of things (IoT) devices have far lower processing efficiency or message transmission speed than that of a personal computer or smart phone, the hackers target to

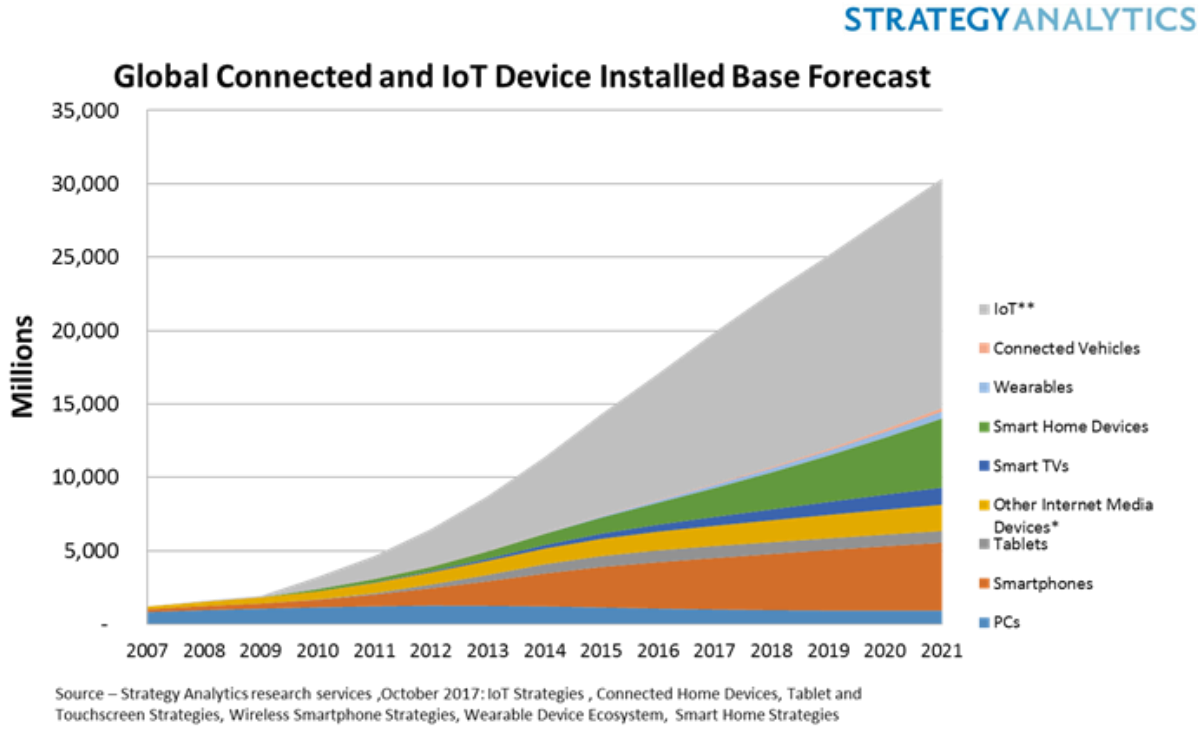
the IoT devices for the reason that the users usually won't monitor the equipment status after the equipment is set up

The IoT device itself is subject to cost or power consumption due to hardware efficiency, so it cannot have a complete protection mechanism, such as a robust coding system, a complete verification mechanism, or a immediate threat notification mechanism. Therefore, the attacker also gradually targets from the hard-to-break hosts of the enterprise to weak-protected and numerous IoT devices. The common attacks to IoT devices are denial of service, ransomware, malicious software, mining program, illegal steals of privacy or favorable information, and so on. In this study, we focus on the common attack techniques against to IoT devices and give some penetration tests for the embedded malicious programs. Finally, we provide some improving suggestions and remind the common users the importance for the IoT security.

Keywords: IOT, Malware, Penetration Testing

一、前言

根據strategy analytics[1]的調查，物聯網裝置於2017年已達全球200億台(如圖一)，並預估2020年會達到300億台的可觀數量，以2023年的預估人口總數80億來看，到2020年平均每個人都會有3台左右的物聯網裝置，如此可觀的數量也引來駭客的關注。



圖一 、物聯網裝置數量預估圖 (資料來源: strategy analytics)

由於物聯網設備數量龐大，因此攻擊者往往只要找到一個漏洞就能使同型號甚至是同廠商的設備都能利用此漏洞進行惡意行為，並且由於物聯網設備開發難度限制通常是採用廣泛應用在嵌入式系統的ARM的精簡指令集核心架構，少數為X86架構，並且大多

採用LINUX作為核心系統來進行開發，但由於硬體效能限制無法安裝功能較為完善安全的LINUX最新發行版，只能採用自行重新編寫的精簡LINUX系統，或是直接應用較舊版本的LINUX核心系統，因此部分舊版LINUX的漏洞或是早期不良設計之系統函數也會存在於嵌入式系統上。

另一方面是使用者不良行為導致的安全性問題，若是使用者本身沒有具備安全防護意識而將帳號密碼採用預設設定或是弱密碼，還有將設備之網路設定開放太多不必要之功能，或者選購設備時採用了含有已知漏洞的物聯網設備，或是購買後沒有定期更新和定期檢查安全設定，都會導致整體網路設備暴露於危險之中。本研究以路由器與網路攝影機為主要研究對象，針對信息嗅探、漏洞偵測、安全性設置等等方向逐一檢視其安全性，並嘗試進行滲透測試以模擬駭客的攻防手法，藉此展示物聯網安全的危害，以期達到強化使用者的安全性意識。

二、相關研究

2.1通訊協定攻擊

物聯網設備本身大多裝載了信號發送器與接收器，或是藉由各式不同媒介例如：網路、藍芽、zigbee…等來收發訊息，因此攻擊者可以藉由竊聽、偽造、訊息重送、劫持與重導方向等方法來收集有利訊息進行攻擊，在2016年，學者Amiza等人提出關於物聯網本身也有類似網路的7層協定，但變化成只有實體層、資料鏈結層、網路層、傳輸層、應用層等五層[2]，根據其不同層的協議也都有不同的攻擊方式，如表一所示。

表一：Amiza 等學者提出的基於協議的攻擊分類

層	攻擊	方法/攻擊策略
實體	干擾	在物聯網設備上產生訊號干擾和發射更大功率訊號阻塞通訊。
	篡改	攻擊其中某節點成功後利用此節點發送偽造訊息。
資料鏈結	碰撞	同時傳輸兩個相同頻率給節點。
	耗盡	重複碰撞節點。
	不公平	使用所有資料鍊結層之攻擊如：地址解析協議（ARP）攻擊、動態主機配置協議（DHCP）欺騙、生成樹協議（STP）攻擊…等。
網路	欺騙、更改或重放路由信息	創建路由循環，擴展或縮短原路由，從選擇的節點吸引其他節點或排斥部分網絡。
	有選擇性的轉發	選擇性的傳輸之前收集的信息。
	坑洞攻擊	使 DNS 解析錯誤位置或引導到惡意目標。
	女巫攻擊	入侵少數節點後偽造訊息影響更多節點
	蟲洞攻擊	使路由之間的跳數增加或縮短來

		達到控制路由表。
	你好泛洪	使用 HELLO 信息包對物聯網系統發起攻擊
	確認欺騙	欺騙網路層確認封包的機制。
傳輸	泛洪	重複請求一個新的連接，直到物聯網系統連接數達到飽和。
	去同步	中斷現有連接。
應用	可靠性攻擊和克隆攻擊:時鐘偏移、選擇性消息轉發、數據聚集失真	在物聯網系統中，對手通常會偽裝成正常的行為。攻擊者還可以在物聯網系統中選擇自己想要的信息，並發起自己的惡意活動。

2.2 殭屍網路感染研究

目前常見的攻擊物聯網設備手法最嚴重的就是利用殭屍網路感染，通常設備感染殭屍網路後會掃描同網域內的所有設備找尋下一個感染對象，因此擴散速度相當快，尤其物聯網設備幾乎很少被檢查，因此殭屍網路感染幾乎難以被偵測。

Mirai[3]在2016年首次被發現用來進行攻擊行為，專門感染LINUX系統的相關設備，主要攻擊對象是家用電腦與伺服器或是路由器與網路攝影機等可以連接網路的計算設備，Mirai搭載了最少60組以上常見的設備預設使用者或管理員帳號密碼，藉由掃描網路後找出有開啟遠端連線端口的設備進行遠端登入的暴力破解，由於大多數物聯網設備的使用者購買設備回家後幾乎不會修改其預設帳號密碼，因此Mirai的擴散速度極快，設備受到感染後使用者幾乎不會察覺異常，只會偶爾有卡頓的情形發生，Mirai在成功感染後會與C&C伺服器保持連接等待攻擊指令，由於感染對象數以萬計，發動DDOS阻斷服務攻擊的危害也高於一般攻擊的數千至數萬倍，幾乎能癱瘓任何網站，且此病毒還有多種變形版本用於不同用途，例如植入惡意程式像是勒索病毒或是惡意挖礦程式的變形版本，但其核心感染技術還是基於Mirai，尤其是Mirai的原始碼已被公開在網路上供任何人觀看下載，因此病毒進化速度與危害更為可怕。

至2018年為止，目前最新的變種病毒是Satori[5]，他是一個蠕蟲病毒具備自我複製能力，主要利用CVE資料庫中所公開含有遠端登入漏洞的設備，並針對市占率高的設備開採其漏洞，支持四種DDoS攻擊，包括udp_flood，syn_flood，tcp_ack_flood和gre_flood，並且還發現此病毒會將正在挖礦的設備修改其錢包地址，將地址改為自己的，利用這些設備來謀取利益，此病毒感染速度非常快，短短幾個月已感染數十萬台路由器。

2.3 逆向工程分析系統

目前針對物聯網設備的漏洞研究與通報或是安全機制分析最快速也最有效的方式就是藉由逆向工程，找出重要的系統檔案以後利用靜態分析或是動態分析方式來找出設備的漏洞，靜態分析是利用軟體將程式反編譯，將重要文件或是執行檔盡量回復成原始程式碼的形式，可幫助分析人員快速找出程式碼裡的結構性錯誤，函數設計上的漏洞等等；而動態分析則是實際讓程式執行，並在執行過程中適時地中斷程式，觀察程式在運

作過程中的動作與其存在記憶體中的值的變化，來分析程式到底做了哪些事情，還可以透過硬體模擬工具來對欲分析設備之韌體進行模擬並運行，使用虛擬化模擬技術來讓研究者不需購買設備也能實際運行設備系統，可以藉此找出系統運行時的各項功能與其參數傳遞，配合動靜態分析可以更快找出系統本身的問題關鍵點。

三、網路攝影機與路由器之滲透測試實驗

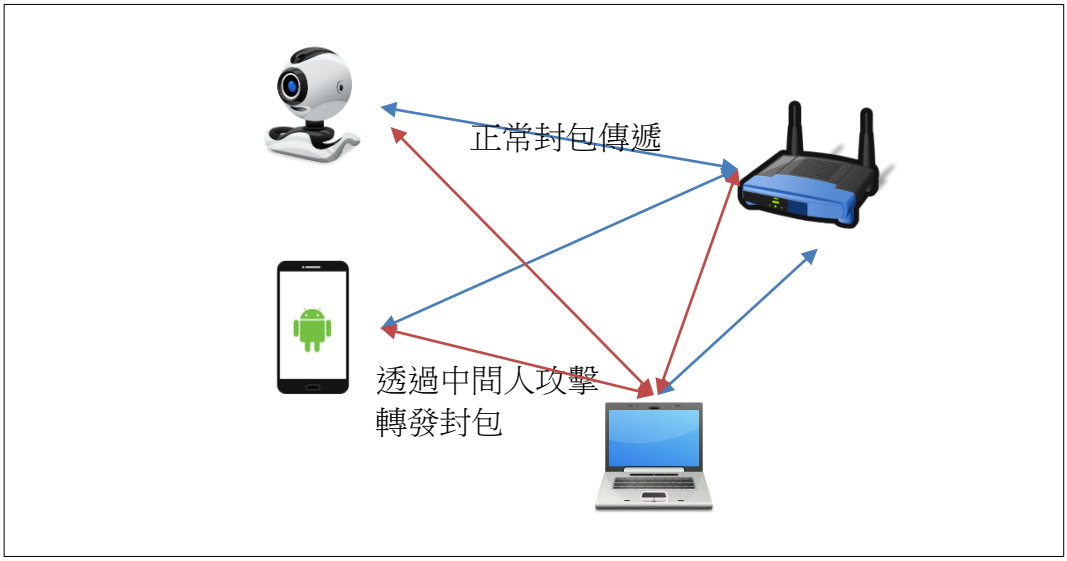
3.1 實驗環境與架構

本實驗針對網路攝影機與路由器的滲透過程中所需要用到的工具與方法做一個詳細的介紹，並會展示其攻擊流程圖，目前大部分的攻擊都與此流程圖的步驟相仿，此外本次實驗的攻擊主機系統是KALI LINUX，它是一個裝載許多攻擊工具的LINUX開源版本，本次研究所使用到的各種工具在此LINUX系統環境都已預先裝載好。

本實驗攻擊對象為市售某品牌的路由器與不同品牌的網路攝影機，實驗過程將嘗試對設備進行滲透測試並分析研究，相關設備之系統類型與區域網路內的對應IP表如表二，整體實驗環境架構圖如圖二，此次實驗環境設計攻擊者與受害設備皆為同一區域網路環境，若是需要從外部網路進行本實驗則需要利用破解WI-FI密碼後進入無線網路環境；或是成功攻入有對外開放服務之設備後利用此路由器做為跳板進入此內部網路後方能進行本實驗。

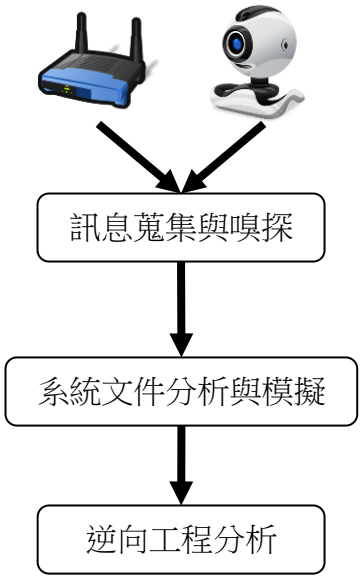
表二： 實驗設備規格

	虛擬機	網路攝影機	路由器	行動手機
運作系統	Linux	Linux	Linux	Android
網路環境	區域網路	區域網路	區域網路	區域網路
網路形式	實體線路	WI-FI	WI-FI	WI-FI
對應 IP	10.1.1.16	10.1.1.15	10.1.1.1	10.1.1.11 10.1.1.12



圖二： 攻擊環境圖 藍色為正常連線 紅色攻擊者攻擊路線

不管對於任何對物聯網設備的攻擊都會經過幾個步驟如圖三，分別是訊息蒐集與嗅探、系統文件分析與模擬、逆向工程分析。



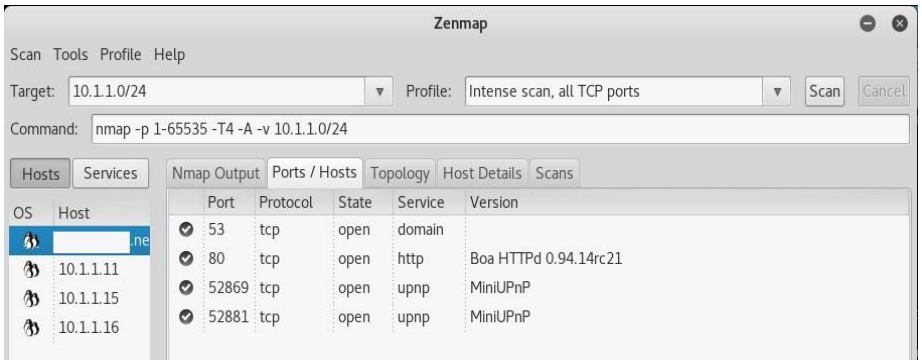
圖三：實驗架構圖

3.2 訊息蒐集與嗅探

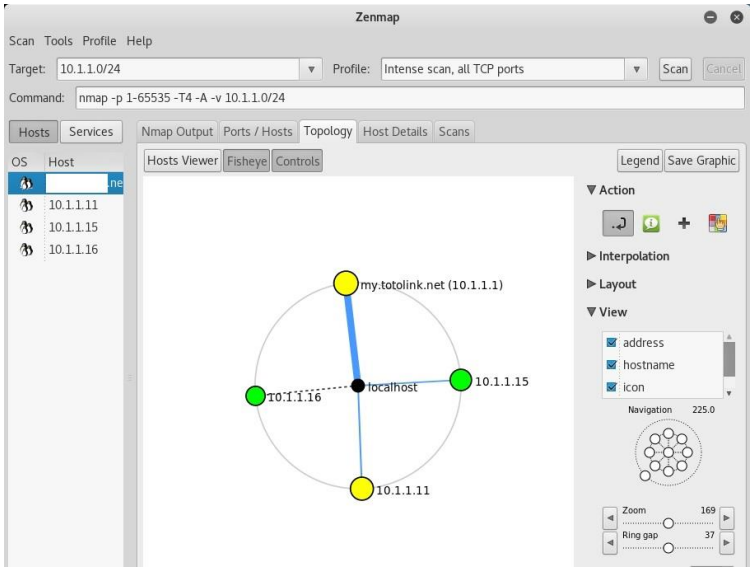
(1) 端口掃描

端口掃描可以說是物聯網攻擊或是任何網路攻擊最重要的前置步驟，藉由掃描欲查詢的目標端口，就可以藉由其開放的端口知道目標提供哪些服務，並且有些端口是危險性非常高的，只要開啟幾乎必受攻擊，因此我們可以利用一些工具例如:NMAP或是ZENNMAP等網路端口掃描工具，藉由掃描其端口與開放的服務類型可以有利於我們縮小攻擊目標。

從圖四我們可以看到ZENMAP掃描後的結果顯示了該 IP 開啟了4個端口，也顯示了該端口對應的服務，圖五是ZENMAP的網路拓樸畫面，不只能提供視覺化的查詢介面，還能提供圖形，若是透過完整查詢還能推測各拓樸的主機系統版本與型號，使攻擊者在擴散病毒時更有效率。若我們要嘗試進攻這台設備的話就可以利用此設備開啟的三種服務分別下手，埠號53是DNS服務端口而對攻擊過程幫助不大；埠號80是http 服務，可以參考許多Web security的攻擊方式嘗試攻擊，一般來說網路攝影機之Web security上主要的攻擊目標通常會使管理者之使用權限、檔案存取權限、網路存取權限…等，主要係因為植入惡意程式都需要上述權限，網路存取權限是為了竊取更有利資訊或是可以用來發送DDOS攻擊，因此攻擊者通常會在掃描出port後利用漏洞掃描工具或是網路上之現成工具進行掃描後針對工具顯示之漏洞進行攻擊；剩下的埠號52869和52881應該就是路由器系統本身自行定義的端口，除非進行逆向工程得知此服務內容否則無法進行攻擊；此外，若是設備有開啟埠號22，就代表此設備有開啟加密遠端連線SSH，針對SSH的加密遠端連線可以利用KALI LINUX的Hydra工具，此工具是專門用來破解多種協議的登錄密碼，破解成功後即可取得帳號密碼登入後取得控制權，不論是網路攝影機或是路由器皆能透過此方式取得控制權限。



圖四： ZENMAP 的視覺化掃描結果可顯示主機系統型號



圖五： ZENMAP 掃描設備後所展示的網路拓樸，含有設備對應 IP 與系統架構

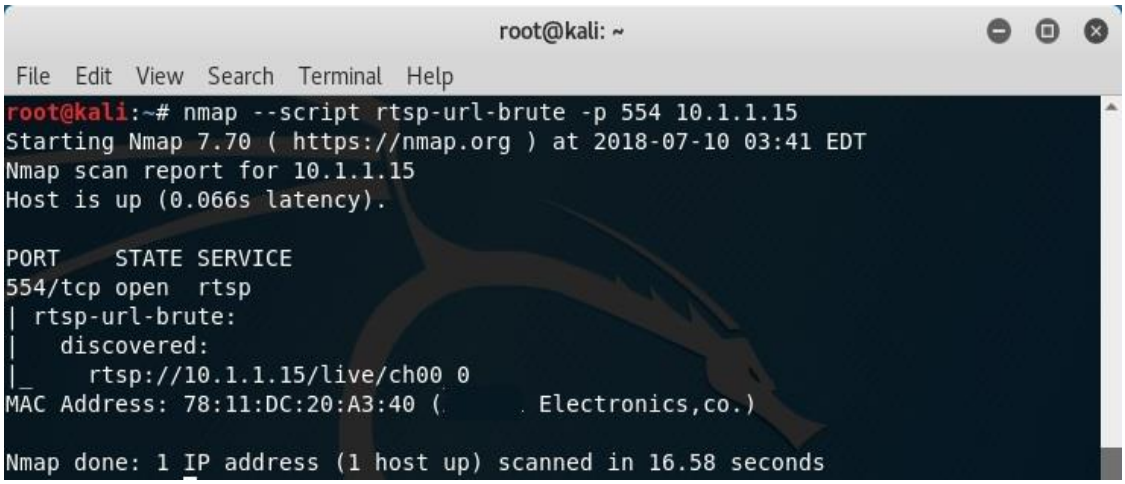
此外若是網路攝影機有開啟 554 port，就能利用 NMAP 裝載的能夠破解關於網路攝影機有開啟 RTSP 協定的專用 554 端口協定，破解成功後會開啟的特殊網址 RTSP-URL，只要開啟命令列並輸入

Nmap －script rtsp-url-brute -p 554 欲破解之 IP

就能進行腳本破解出來 RTSP-URL，如圖六，RTSP-URL 是一種特殊協定，只要在瀏覽器的網址列輸入此 RTSP-URL 特殊網址就能開啟 RTSP 協定，並將其網路攝影機的監視畫面同步傳送經由輸入網址後用網頁或是撥放器的方式顯示給使用者，並且不需要登入身分認證，危險性非常高，此次實驗有嘗試在網路攝影機開啟 554 端口後運行 RTSP 服務並成功破解出 RTSP-URL 如以下網址

Rtsp://10.1.1.15/live/ch00.0

將此網址輸入進瀏覽器後期顯示畫面如圖七，且只需要輸入網址完全不需要類似登入使用者的動作，十分危險。



圖六 利用 NMAP 的破解 RTSP 專屬網址畫面

圖七 輸入 RTSP-URL 後同步顯示出網路攝影機之畫面

目前得知網路攝影機主要攻擊目標是竊取畫面與取得管理者權限，路由器主要攻擊目標是取得管理者權限或是網路請求權限與檔案存取權限。

(2) 中間人攻擊

中間人攻擊[4]是蒐集訊息最好的方式之一，藉由工具輔助可以幫助竊取對話金鑰，使攻擊者可以有效的介入目標的訊息傳遞途徑，用以獲取有用訊息，這部分使用到的工具為 KALI LINUX 內建的 Arpspoof 工具與 ettercap 工具，此類工具攻擊原理都與 ARP 協定還有 DNS 查詢協定有關，ARP 攻擊是利用 ARP 協定中目標在查詢路由器 IP 位置時攻擊者發送大量假的 ARP 回覆，讓目標以為假的 ARP 目標就是路由器，此時就能讓目標的封包都經過攻擊者來轉發，就能達到劫持訊息的效果，DNS 欺騙攻擊也是相同原理。

如圖八所示，Arpspoof 工具在執行時需開啟路由轉發功能後就可以攔截目標到路由器之前的所有封包，攔截到的封包可以使用 Wireshark 工具來過濾並檢視封包內容，如圖九所示。

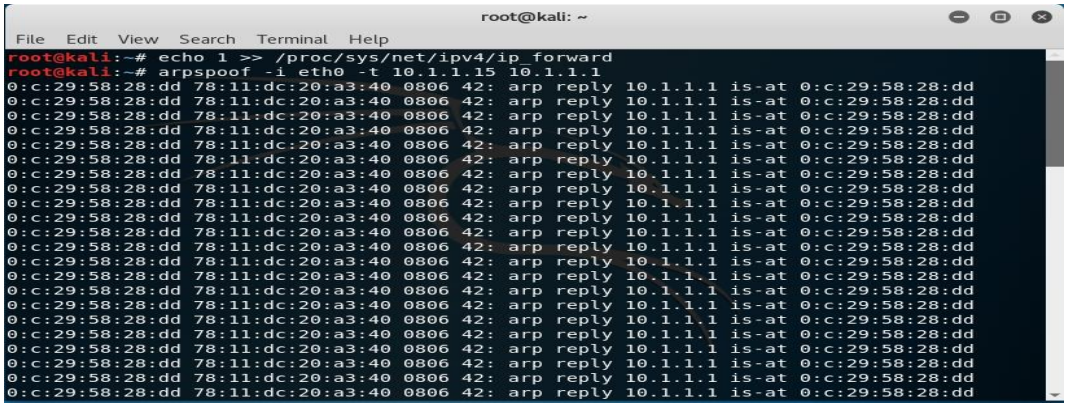
輸入下面指令後就能開啟路由轉發功能：

```
echo 1 >> /proc/sys/net/ipv4/ip_forward
```

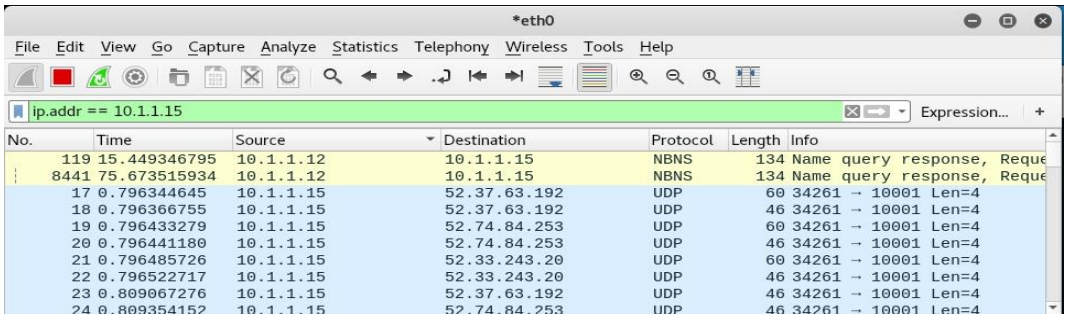
接著，輸入下面指令就可以啟用 Arpspoof 攻擊：

Arpspoof -I eth0 -t 目標一 IP 目標二 IP

指令使用時只需要知道欲啟用路由轉發功能的網路介面卡與欲轉發的兩個目標 IP 就可以開始進行攻擊，開啟 Arpspoof 即可使用 Wireshark 工具同步攔截封包查看其內容來獲得有利資訊，但由於本次實驗傳輸之封包皆透過加密傳輸，因此無法獲得有利資訊需要往逆向工程的方面著手。



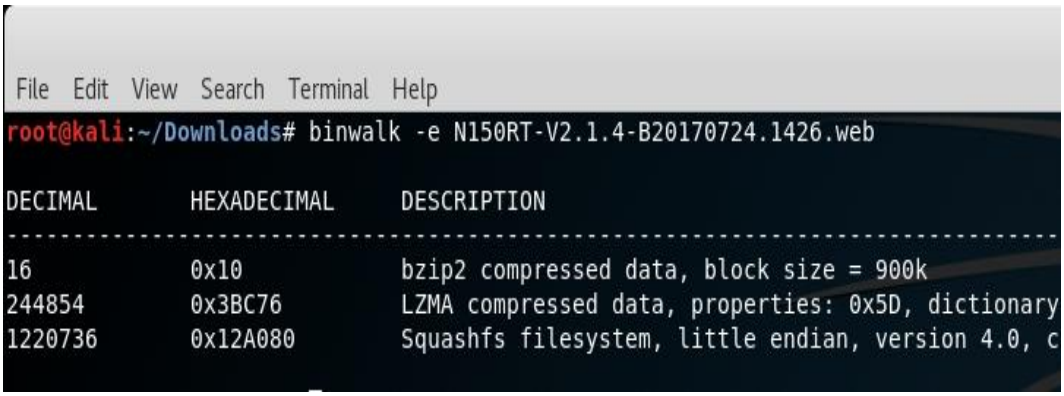
圖八 Arpspoof 執行封包轉發中



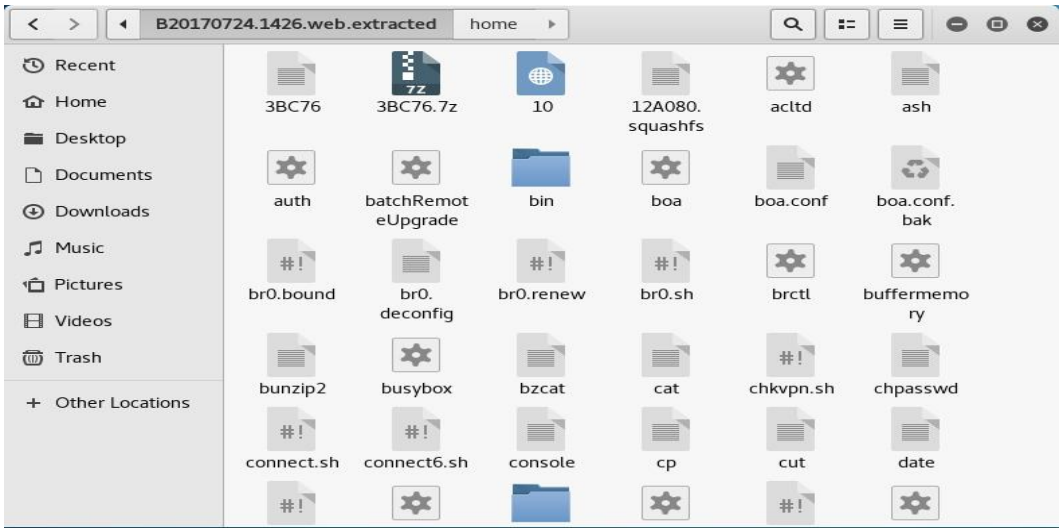
圖九 利用Wireshark撈取封包分析中

3.3 系統文件分析

檔案系統分析的部分目前的研究方式有兩種方式，第一種是購置相同款式之設備，利用修改其韌體檔案強制開啟遠端連線服務後，利用動態分析來分析記憶體數據與程式執行情況來找出漏洞，或者是將檔案打包出來進行靜態分析；第二種方式是搜尋網路上的官方文件或手冊與釋出的韌體檔，由於韌體檔通常是已經打包或是封裝完後的檔案，因此無法利用軟體直接查看，可以利用Binwalk工具來分析檔案，Binwalk是一個強大的檔案分析工具，能夠分析檔案架構還能過濾檔案雜訊，藉此還能提取文件，因此常被利用在文件分析中，如圖十，我們從官網下載回來的韌體檔案利用Binwalk分析後嘗試提取，啟用Binwalk指令後再加上-e 就可以啟用自動化提取檔案功能，並在圖九展示其提取出的所有檔案，以利進行下一步的分析。



圖十： 利用 Binwalk 分析檔案並找出提取檔案的位元大小



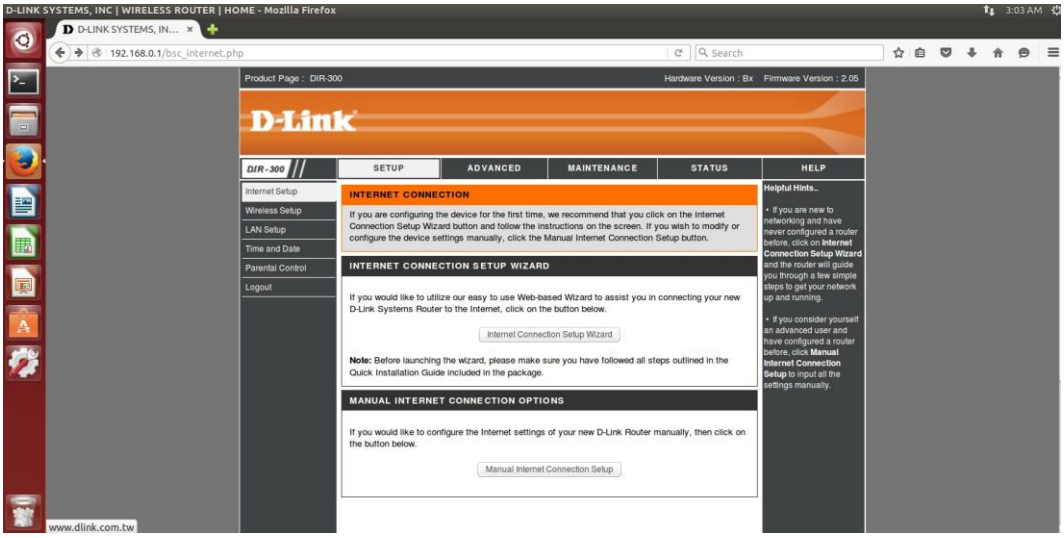
圖十一： 利用 Binwalk 分析韌體提取出來的檔案文件，含有非常多原始碼。

使用官方釋出的韌體檔案來進行分析還有另一種方式就是使用硬體模擬工具，硬體模擬工具搭載了非常多設備商所開發之設備之系統所需的大多數程式，運用類似虛擬機的模式模擬出設備本身實際運作之情形，我們的實驗使用firmware-analysis-toolkit這項工具來進行模擬，程式執行示意圖如圖十二，韌體模擬執行畫面如圖十三。韌體成功模擬運行之後我們可以利用Wireshark工具來攔截封包，查看設備在執行不同頁面時傳遞了哪些參數，這些參數都可以在逆向分析時幫助除錯加速破解。

```
                    By Attify - https://attify.com  | @attifyme

[?] Enter the name or absolute path of the firmware you want to analyse : DIR850LB1_FW210WWb03.bin
[?] Enter the brand of the firmware : dlink
[+] Now going to extract the firmware. Hold on..
[+] Firmware : DIR850LB1_FW210WWb03.bin
[+] Brand : dlink
[+] Database image ID : 1
[+] Identifying architecture
[+] Architecture : mipseb
[+] Storing filesystem in database
[!] Filesystem already exists
[+] Building QEMU disk image
[+] Setting up the network connection, please standby
[+] Network interfaces : [('br0', '192.168.0.1'), ('br1', '192.168.7.1')]
[+] Running the firmware finally
[+] command line : sudo /home/ec/firmadyne/scratch/1/run.sh
[*] Press ENTER to run the firmware...
```

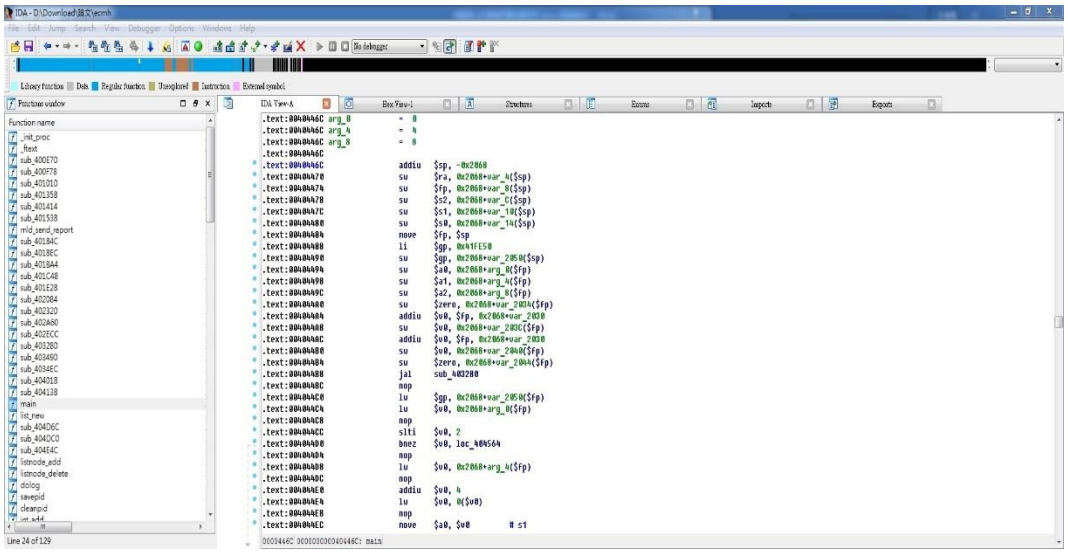
圖十二：firmware-analysis-toolkit執行畫面(資料來源：github/attify)



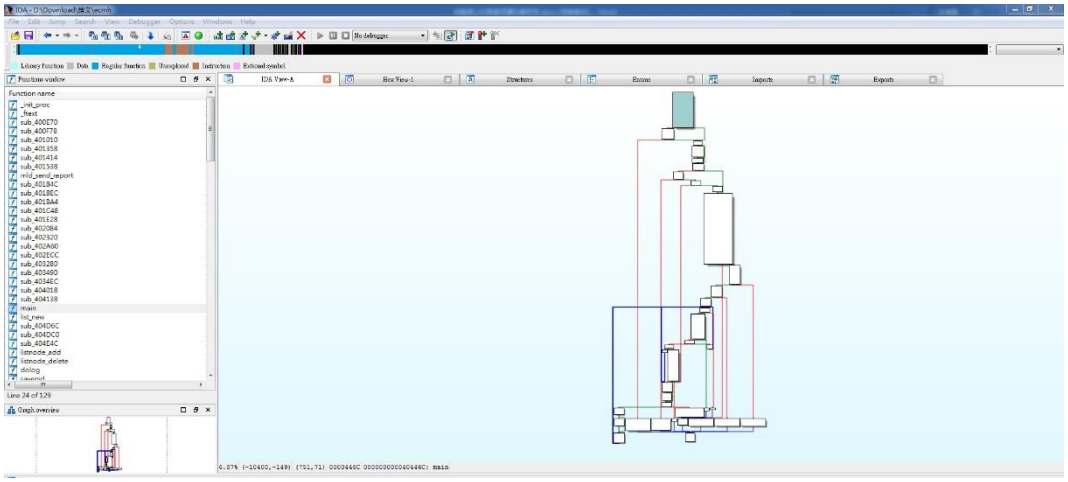
圖十三：韌體模擬路由器後台執行畫面，以 Dlink-DIR300 為例

3.4 系統逆向工程分析

針對提取出來的檔案中有些包含執行檔的部分，或是其他無法以軟體開啟的檔案也都可以嘗試用靜態分析工具IDA PRO來進行反編譯，利用反編譯我們可以看到程式模擬出來的執行檔轉換成組合語言的畫面，如果檔案支援的話還可以反編譯成C語言的形式就更能增加可讀性，如圖十四就是利用IDA PRO對韌體中包含的執行檔做反編譯的靜態分析畫面，還可以開啟右鍵選單選擇圖形化呈現程式分支圖，如圖十五。利用程式分支圖可以令我們鎖定目標功能，如需要檢視登入頁面執行哪些指令與引用哪些參數，如需查詢函數內容也可至圖十六顯示的函式位置圖，可幫助鎖定目標的功能，快速理解不同函數運作時所需之參數與運作結構，利於找出漏洞或是可以利用之指令來對設備進行更深一層的滲透。



圖十四： IDA PRO 反編譯程式畫面



圖十五： 圖形化顯示程式分支與跳轉，可幫助除錯。

圖十六： 函數儲存位置與型態與名稱

```
LOAD:0058C55C                   .byte    0
LOAD:0058C55D                   .byte    0
LOAD:0058C55E                   .byte    0
LOAD:0058C55F                   .byte    0
LOAD:0058C560 function_table:   .word aSetmultipleact    # DATA XREF: handle_HNAP1+150↑o
LOAD:0058C560                                            # "SetMultipleActions"
LOAD:0058C564                   .word sub_433768
LOAD:0058C568                   .word aGetdevicesetti_4  # "GetDeviceSettings"
LOAD:0058C56C                   .word sub_432D28
LOAD:0058C570                   .word aGetoperationmo    # "GetOperationMode"
LOAD:0058C574                   .word sub_433F70
LOAD:0058C578                   .word aGetsmartconnec_4  # "GetSmartconnectSettings"
LOAD:0058C57C                   .word sub_464DD4
LOAD:0058C580                   .word aGetuplinkinter    # "GetUplinkInterface"
LOAD:0058C584                   .word sub_433F48
LOAD:0058C588                   .word aLogin_4           # "Login"
LOAD:0058C58C                   .word sub_42ACB0
LOAD:0058C590                   .word aGetwlanradiose_6  # "GetWLanRadioSettings"
LOAD:0058C594                   .word sub_46462C
LOAD:0058C598                   .word aGetclientinfo     # "GetClientInfo"
LOAD:0058C59C                   .word sub_447B94
LOAD:0058C5A0                   .word aSetclientinfo_0   # "SetClientInfo"
LOAD:0058C5A4                   .word sub_447F58
LOAD:0058C5A8                   .word aUpdateclientin_4  # "UpdateClientInfo"
LOAD:0058C5AC                   .word sub_448B70
LOAD:0058C5B0                   .word aGetwlanradiose_7  # "GetWLanRadioSecurity"
LOAD:0058C5B4                   .word sub_463C90
LOAD:0058C5B8                   .word aSetdevicesetti_9  # "SetDeviceSettings"
LOAD:0058C5BC                   .word sub_4346EC
LOAD:0058C5C0                   .word aGetapclientset    # "GetAPClientSettings"
LOAD:0058C5C4                   .word sub_433EF8
LOAD:0058C5C8                   .word aSetapclientset_1  # "SetAPClientSettings"
LOAD:0058C5CC                   .word sub_433F20
LOAD:0058C5D0                   .word aSetwlanradiose_11  # "SetWLanRadioSettings"
LOAD:0058C5D4                   .word sub_46423C
0017C560 0058C560: LOAD:function_table (Synchronized with Hex View-1)
```

四、物聯網安全分析與建議

本研究針對路由器與網路攝影機之植入惡意挖礦程式的攻擊手法與一連串的滲透測試過程進行分析，我們可以發現從連線開始到訊息傳遞再到程式執行過程都有著無數的攻擊機會，如[6][7][8][9]都揭示了從連線的協議、檔案傳輸的協議都需要在設計過程中就需要有非常完善的設計與使用，不然等到設備生產完成之後就無法對硬體進行變更，使用者也不一定願意更新韌體，因此每個環節都需要重視不容輕忽才能夠有效預防惡意挖礦攻擊或是其他攻擊方式，在[10]的文獻中顯示，不安全的物聯網設備甚至會洩漏個人隱私或是對生命與財產造成危害，如：攻擊電力系統使其癱瘓等，因此本章節將針對前一章節提出的攻擊流程來逐一提出預防技巧，希望能提升整體物聯網的安全性。

4.1 信息嗅探與蒐集

本次實驗所使用之網路攝影機其韌體在最初版本有開放22、554 port，因此可以進行遠端連線存取系統與RTSP連線監看攝影機畫面，若是遭受有心人攻擊就有可能破解遠端連線取得管理者權限，就能夠執行惡意程式，例如：惡意挖礦軟體、或是發動DDOS網路攻擊軟體，或是偽造網路請求來癱瘓整體家用網路與監控當前網路環境，若是破解RTSP協定也能不需登入就取得監視器畫面，有極大可能會洩漏個人隱私或是讓有心人紀錄使用者作息進而闖空門等事件。所幸最新版本之韌體更新之後已封鎖所有端口，只開放514 port，讓所有對攝影機支存取須透過手機APP來連線廠商之伺服器，透過伺服器來傳輸指令，就可以隨時監控指令傳送內容與避免設備遭受攻擊。路由器之部分則是如3-2節所演示，需要對其進行更深之逆向工程就可能發掘出漏洞函式，因此使用者須注意在設定路由器之設定時需更換預設管理者帳號密碼與避免開啟使用不到之功能，若有需要也須做好最完善之設定才能開啟服務，就能避免多數攻擊，若是路由器遭受攻擊，其危害遠大於網路攝影機，不管是發動DDOS攻擊之封包數量與網路環境遭受控制之程度都相當可怕，攻擊者甚至能存取任何與此路由器連線之設備，不可不防。

因此不管是殭屍網路感染或者是其他攻擊，都能夠發現網路端口掃描是其中的重要關鍵，所有的攻擊手法都需要根據開放的服務不同而有所改變，因此最好的預防方法就是盡量將不使用到的功能與端口關閉，開放越多端口代表著開放越多的攻擊機會，再來就是已開啟的服務需要定期進行安全性檢查，定期檢查網路設備的存取紀錄LOG檔案，檢視有無被掃描過的痕跡，或是異常的IP連線也需要注意，若是能夠定期重新啟動設備也能藉由重新開機清除其快取記憶體使其初始化就能避免惡意程式的留存。

中間人攻擊的手法主要是依靠大量偽造假的封包藉以欺騙使用者，因此可以使用幾種方式來避免此種情形發生；第一種是利用多方認證機制，使雙方在傳遞金鑰時可以藉由其他應用程式加以認證，讓攻擊者使用未經過認證的金鑰來偽造時能夠有效的被識破；第二種方式是在金鑰交換時加上時間戳記，例如當雙方在傳遞訊息時原本只需要40秒，但是某天在無網路延遲的情況下卻變成60秒，就可證明傳輸路徑中有中間人攻擊發生；第三種方式是使用一次性金鑰，每次建立連線皆使用不同的金鑰，這樣使得攻擊者在偽造大量重複的封包時能夠有效的被識破；此外使用的金鑰也能夠參照密碼學設計較強的密碼金鑰，較難以被複製，例如本次實驗過程藉由中間人攻擊或取之封包皆經過廠

商之特殊加密，除非藉由逆向工程破解其加密技術，否則無法從中獲取有利資訊，消費者在挑選產品時可以針對此點進行安全性評估。

4.2 檔案系統分析

影響系統檔案分析的安全性因素其實是環環相扣的，設計人員在構思部分功能的函數時，如果沒有全面考量因而導致攻擊者能夠不經過身分認證就引用函數，或是在傳遞引數時沒有考量到溢位的發生，導致攻擊者能利用溢位來覆蓋指令進而奪取控制權，都是危險性相當大的情況，要能有效的避免這些情形發生，最好的辦法就是要定期做代碼審計，藉由不同設計人員或是不同領域專長的研究人員用不同的面向來審視整體程式碼，找出其可能有漏洞的地方加以改善。

另外防止韌體文件檔案被任意的提取，最好的方式是做代碼混淆，利用代碼混淆來將原始程式碼改變成亂碼形式但是程式本身又能正確判讀，就能藉此躲避攻擊者的分析，就算程式碼本身有問題也不易被察覺，或是要將檔案加密也可以，但須考量物聯網設備的系統資源能否及時處理大量的加密文件。

若是使用者能夠定期更新物聯網設備之韌體絕對是最有效的防禦手段，因為攻擊者在進行逆向工程與檔案系統分析時，非常曠日廢時，經常需要測試數千數萬行之函式結構與參數傳遞過程，但經過使用者更新韌體後，所有設備中的變數或是參數皆重新洗牌，運作機制可能也不盡相同，攻擊者好不容易發掘之漏洞皆無法利用之，但若使用者無此安全意識，則攻擊者若成功發掘漏洞，此廠商所開發的數以萬計同型號之設備皆適用此漏洞，影響層面非常廣大，因此使用者務必培養定期更新之習慣避免遭受攻擊。

五、結論

本論文針對物聯網設備常見的攻擊手法與攻擊途徑做出整理，希望能夠讓社會大眾體悟到在我們生活環境中眾多的物聯網設備，所存在的安全問題，若是不加以防範的話會對我們的生活造成多大的危害。無法避免的趨勢，我們生活中的物聯網設備將越來越多，希望藉由此本論文的介紹和紕漏，能夠提升社會大眾對於物聯網設備安全的重視，進而避免人身財物的損失。未來，能夠有更多的人員投入物聯網安全的研究，找出物聯網設備的不安全設計與相關漏洞，讓貼近我們生活的物聯網更加安全。

參考文獻

[1] strategy:
<https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2017/10/26/smart-home-will-drive-internet-of-things-to-50-billion-devices-says-strategy-analytics#.WfKej1WWaUk> (Retrieved May 2018)

[2] Amiza, Naimah, “Internet of Things (IoT): Taxonomy of Security Attacks,” *3rd International Conference on Electronic Design (ICED)*, 2016:pp.321-326.

[3] Hamdija Sinanović ,Sasa Mrdovic “Analysis of Mirai malicious software” *2017 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*,2017,pp 1-5.

[4] Bhargav Pingle,Aakif Mairaj,Ahmad Y. Javaid” Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use”, *2018 IEEE International Conference on Electro/Information Technology (EIT)*,2018,pp 192-197.

[5] Satori:< https://www.ithome.com.tw/news/123999>(Retrieved May 2018)

[6] Mahmoud Ammara, Giovanni Russellob, Bruno Crispoa, “Internet of Things: A survey on the security of IoT frameworks” *2018 Journal of Information Security and Applications Volume 38*, February, pp 8-27.

[7] Jianwei Hou, Leilei Qu , Wenchang Shi, “A survey on internet of things security from data perspectives”,*Computer Networks Available online 28 November 2018.*

[8] Chiara Bodeia, Stefano Chessaa, Letterio Galletta “Measuring security in IoT communications”, *2018 Theoretical Computer Science Available online 5 December*.

[9] Ashok Kumar Das, Sherali Zeadally , Debiao He “Taxonomy and analysis of security protocols for Internet of Things”, *2018 Future Generation Computer Systems Volume 89,* December 2018, pp 110-125.

[10] Fadele Ayotunde Alaba , Mazliza Othman , Ibrahim Abaker Targio Hashem , Faiz Alotaibi “Internet of Things security: A survey”, *2017 Journal of Network and Computer Applications Volume 88*, 15 June 2017, pp 10-28