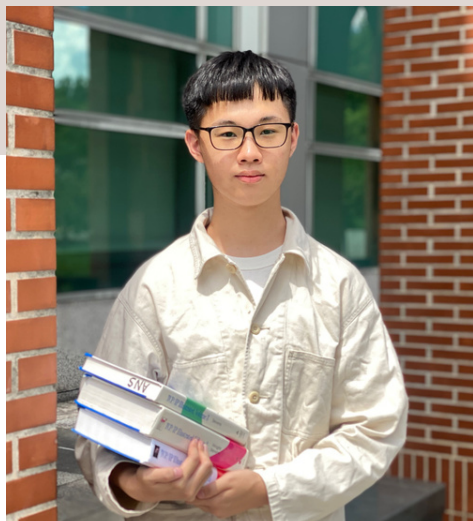




# 林宥熙(Raymond)

業餘漏洞研究員

kernel / 底層開發 愛好者

## 👤 簡介

我是來自南山高中的林宥熙，是一位業餘的漏洞研究員。除了投入漏洞研究，我也積極參與開源專案，貢獻軟體安全的知識，幫助提升專案的安全性。

資安領域之外，我還是一名正在探索 kernel 和底層開發的學生，對作業系統的運作原理充滿興趣。

我對未知領域保持強烈好奇心並願意深入學習。喜歡追根究底、長時間專注研究技術，並且不輕易放棄，這些特質讓我的學習和研究更有效率。

我期望能成為一位有扎實學理背景的軟體工程師和專業漏洞研究員，投入資訊產業和社群發揮所學。

## 🕒 活動經歷

- AIS3 2022 最佳專題獎
- AIS3 EOF 2024 Finals #8
- MyFirst CTF 2022 #15
- AIS3 EOF 2021、2022
- AIS3 Junior 第 1~3 屆 (2022 ~ 2024)
- 台灣好厲駭第七屆 (2022) 結業學員
- 台灣好厲駭第八屆 (2023) 結業學員
- 台灣好厲駭第九屆 (2024) 學員
- HITCON CMT 台灣駭客年會 2024 會眾
- QRACON 學生量子計算機年會 2024 會眾

## ⊕ 校內經歷和服務經驗

- 高中 1~3 年級 數理資優班
- 多元選修：C 語言 / TQC C+ 認證
- 國中一年級至高中一年級擔任衛生糾察隊員，累積 150 小時以上服務時數

## 聯絡方式

✉ [linraymond2006@gmail.com](mailto:linraymond2006@gmail.com)

🔗 [linraymond2006](https://github.com/linraymond2006)

📍 新北市私立南山高中

## 漏洞列表

CVSS 6.5 CVE-2024-6044

CVSS 8.8 CVE-2024-6045 **RCE**

CVSS 9.8 CVE-2024-45694 **RCE**

CVSS 9.8 CVE-2024-45695 **RCE**

CVSS 8.8 CVE-2024-45696 **RCE**

CVSS 9.8 CVE-2024-45697 **RCE**

CVSS 8.8 CVE-2024-45698 **RCE**

中度風險 ZD-2024-00904 **RCE**

## 核心技能

- C (ISO/IEC 9899:1999)
- x86 組合語言
- 逆向工程 / ghidra 框架
- 漏洞挖掘 / 利用程式開發

## 實驗性專案

- 實做 x86 MBR bootloader
- 摸索 x86 kernel

## 證照和檢定

- Certified Network Defender
- IELTS Academic band score 6 (CEFR B2)

# 目錄

## 個人資料

### 自傳

自傳和學習過程

技能組

個人特質和申請動機

讀書計畫

成績單和科學班就讀證明

## 其他有利審查資料

### 培訓和 活動心得

AIS3 2022 最佳專題

AIS3 EOF 2024 決賽

台灣好厲駭培訓

AIS3 Junior 營隊

### 作品、 成果和證照

漏洞列表

實驗性專案

CND 網路防禦專家認證

# 自傳

# 自傳和學習過程

## 國中時期：種下興趣種子、技術啟蒙

在接觸電腦前，我的興趣是曲棍球，但自從國中一年級接觸到 scratch 語言後，便開始對電腦程式產生興趣。隨後我透過 Udemy 自學了 python 3，並深深著迷於電腦用可程式化邏輯操控的運作方式。

國中一、二年級我開始探索自己的興趣所在，嘗試了 Kotlin 和 Javascript 兩種語言，這也進一步激起了我對程式設計的熱情。

## 國中升高中：廣泛探索、發掘不同研究領域

國中三年級因緣際會下在網路上看到了「作業系統」這個主題，得知作業系統是驅動每一台電腦的軟體，好奇心發作就跳入了這個主題，一摸索就是兩年多。

也是這個機會，讓我學習到了原本沒有接觸到的知識：閱讀恐龍書、算盤本、自學 C 語言、在 youtube 上看影片學 8086 assembly（古早味 MASM），到後來看文件學習 i386、x86\_64 指令。不僅收穫了豐富的技術，還因為缺乏同儕間交流和老師指導，促使我自行閱讀技術文件、尋找資源和提問來解決困難，這些無形中的培養的問題解決能力也成為了我日後學習的重要助力。這段期間我幾乎每個平日都花二到三小時進行學習，無論是上學前的早晨、下課時間、甚至中午午休等握零碎時間，我都不放過任何趁機學習的機會。

學會了這些技術後，在爸爸的建議下參加了人生第一場解題式 CTF 比賽：AIS3 EOF 2021，開始學習逆向工程和 binary exploitation，並在隔年透過 MyFirst CTF 第十五名錄取了 AIS3 營隊，拿下了最佳專題獎。

## 高中二年級：探索資安、深入了解

在網路上看到了 CND 這張證照，激起了我對於藍隊未知領域的好奇心。我先上了 NINS（網路基礎架構與服務）後就報名 CND 課程，上課了之後才發現原來攻防是一體兩面的事情，也學會了企業內的藍隊基礎知識，並且得以以防守方的角度檢視攻擊面。

# 自傳和學習過程

## 高中二年級：轉折

在高中二年級時，我發現我的現有知識已經沒有辦法支撐 kernel 開發所需，在社群分享我的困擾後，一位工程師：廖先生 主動聯繫我。他用下班時間和我說明我應該要做什麼、可以從哪邊下手學習系統軟體。

聽到建議後我內心掙扎許久，畢竟那是我兩年來全心投入的研究，直到開始研究 Linux kernel，才發現原來我花了兩年在做白工。那時的我根本不清楚什麼是作業系統，什麼是核心。

受過這次幫助後，我希望我日後也可以像廖先生一樣，成為在他人有困難時能幫助他人的人，在有足夠的相關知識之後，我打算藉由回答論壇上的問題來解決其他人的困惑。

## 高中三年級：整合所學、踏上漏洞研究之路

今年初參加了 AIS3 EOF 決賽後，我才看清這兩年打 CTF 雖然幫助學會撰寫 exploit，但漏洞「挖掘」和「開發」的能力仍是不足的。

自我反省後我懷疑是 CTF 讓我看到的攻擊面不夠廣、或不貼近實際情況，舉例：binary exploitation 題目不外乎就是 stack / heap-based buffer overflow、format string、UAF / double free，比賽旨在引導參加者用漏洞寫出對應的利用程式，但是現實則不全然如此：邏輯漏洞和函式間協作的問題更像是趨勢。

由於 IoT 設備對於新手比較友善且容易取得，設備上運行的多個服務也提供了廣闊的攻擊面，於是我以 IoT 設備為起點。選擇台灣在地品牌 D-Link 為研究目標，從去年年底開始摸索，並於今年三月發現了人生第一個真實世界的漏洞，至今共累積七個 CVE，多數都是直接導致 RCE 的。目前我也從閉源的 IoT 設備慢慢轉向開源專案，並且在一些小專案中做出貢獻。

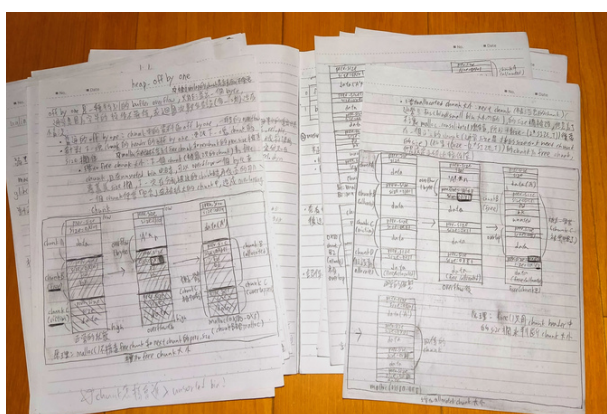
# 自傳和學習過程

## 國中至高中：實務外的理論知識學習過程

資訊工程和資訊安全是很重視實務的學問。但是沒有理論支撐的實做只是沒有價值的空殼，我在沒有足夠知識的情況下探索 kernel 失敗後，對於這一點感悟甚深。

所以我利用零碎時間（例如：下課、午休、睡前等）拿起紙本書籍補充理論知識。目前累積的閱讀多數是 漏洞利用和分析、逆向工程、Linux kernel、網路協定 等主題，也不乏少數實務書籍。

我也很感謝爸爸願意把辛苦賺來的錢投資在我的教育上，而且總是支持我追求我喜愛的技術，讓我可以繼續學習和進步。



binary exploitation 手寫筆記

筆記

07/08/2023 22:22

- 如果該object被初始化，則會創建object/instance (實例)
- 若沒有指定初始化的值，則新建的instance的初始值是undetermined，而如果有初始化的動作，對於每個新的instance都會
- 至於variable length array的lifetime是從該array的宣告開始，直到離開宣告的那個scope
- 變量和初始化的規定和variable length array一樣
- 對於volatile物件的值，最後儲存的值不一定是顯性 (explicit) 的，因為它可能會被外部因素改變
- constant address：兩個指向同一物件的指針，不論在何時出現，他們比較的值也會相等，但這樣不代表該地址在相同process的會變
- 型別 (types)
  - 一個物件或是程式的回傳值是由型別來決定要如何存取的
  - types的分類
    - object types
      - 形容一個物件的型別
    - function types
      - 形容一個function的型別
    - incomplete types
      - 可以形容一個object，但是需要提供更多資訊才能決定該型別的大小 (寬度)
  - 種類：
    - \_Bool：足以存放0和1的型別
    - char：足以存放任何一個basic "execution" character set中成員的型別
      - 如果一個basic execution character set的成員放在char，他的值規定不為負 (nonnegative)
      - 其他的character set members存放在此型別時是implementation-defined behavior，但是就是不能超出char可以表示的範圍
      - 一個char型的最小值在c標準中由宏定義，他的值一定是0或更小的CFR-MIN (lowest char minimal) 型別一個

ISO/IEC 9899:1999 (C99 規格書) 閱讀筆記



國中至現在累積的紙本書籍閱讀 (附件：我的閱讀書單)

# 技能組

- 可實際運用 / 持續精進中
- 有基本概念 / 正在學習
- 開始學習 / 正在探索
- 個人特質 / 軟實力

## 經驗 / 特質 / 軟技能

自學能力 / 耐挫力

技術文件閱讀能力

駭客道德 / 正面價值觀

學習動機 / 好奇心 / 求知慾

追根究底的精神

團隊協作能力

## 系統軟體 / kernel 開發 相關

C (ISO/IEC 9899:1999)

x86 組合語言

linux kernel 知識

## 資訊安全 / 軟體安全 相關

逆向工程 / ghidra 框架

binary exploitation (x86 / arm / mips)

自動化 exploit 撰寫

security code review

韌體分析

## 其他

密碼學

密碼學：數學原理

TCP/IP協定組

# 個人特質和申請動機

## 為什麼選擇特殊選才？為什麼選擇資工系？

- **摸索後的結果：**從國中起長時間投入資訊和資安領域的學習後，我逐漸對資訊工程充滿好奇心和熱忱，發覺這才是我所喜愛的學問。
- **術業有專攻：**‘You can do anything, but not everything’，我的專業能力在通識教育中無法完整的發揮，希望藉由特殊選才進入大學繼續學習，讓興趣真正轉變成專業。
- **構建知識面：**自學過程中累積了許多分散的知識點，希望透過特殊選才管道進入大學，接受更系統化的學術訓練，並將這些技術整合成一個通用的知識體系。

## 適合資工系的優勢特質？

- **對電腦的熱愛、強烈的學習動機：**我投入大量時間在學習與電腦相關的技術，將其視為專業與興趣的結合。這讓我不論是假日還是平日的凌晨都會不分晝夜地鑽研技術。我也充分利用零碎的原子時間，在學校下課或等待時繼續學習。
- **自學能力強：**對於需要的知識會自己尋找，而不是被動接收。
- **對技術的執著：**對於技術的追求不中斷，持續學習最新的知識。在執著驅使下，我非常願意廢寢忘食的投入長時間研究。
- **對未知領域保有好奇心：**我願意探索尚未學習的知識，這種特質讓我能夠適應變化快速的資訊世界。
- **了解學習意義：**我很清楚學習的目的和意義不是為了應付考試，而是為了實際應用做的預演。
- **誠實面對自己的弱點並修正：**若檢驗出知識盲點會立即修正並尋找資源學習，而不是視而不見或不懂裝懂。
- **實作和理論配合：**在實作發現理論知識不夠時，我會補足理論知識；理論學習到一定程度後，我會動手實做，把所學具體化。

## 為什麼選擇臺灣師範大學資訊工程學系？

- 國際化，有多個國外姊妹校
- 強大師資、優秀同學和學長；強調教學、研究、創作共同發展
- 和臺灣大學、台灣科技大學 組成三校聯盟，可跨校選課或雙主修
- 離住家只有兩個捷運站的距離，對於日後穩定求學有很大幫助

# 讀書計畫和說明

| 學習階段   | 計畫                                                                                                                                                                                                                      |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 特殊選才後  | <ul style="list-style-type: none"><li>• 繼續漏洞挖掘，嘗試影響力更大的專案</li><li>• 學習惡意軟體開發和偵測技術</li><li>• 研讀數學</li><li>• 補強資料結構</li></ul>                                                                                             |
| 大學時期   | <ul style="list-style-type: none"><li>• 加入大學社團</li><li>• 重新開始學習 kernel</li><li>• 利用 MOOC 平台拓展專業領域</li><li>• 大量閱讀書籍</li><li>• 搭建部落格撰寫技術文章</li><li>• 繼續學習英文並重新參加 IELTS 考試</li><li>• 進入資安企業實習</li><li>• 投入自由軟體開發</li></ul> |
| 大學畢業後  | <ul style="list-style-type: none"><li>• 考取研究所</li></ul>                                                                                                                                                                 |
| 研究所時期  | <ul style="list-style-type: none"><li>• 追蹤各大頂尖會議並大量閱讀論文</li><li>• 爭取期刊論文發表機會</li><li>• 將部落格文章整理成系列教學文</li><li>• 為工作累積實力</li></ul>                                                                                       |
| 研究所畢業後 | <ul style="list-style-type: none"><li>• 應徵工作（資安、研發）<ul style="list-style-type: none"><li>◦ 藍隊</li><li>◦ 紅隊</li><li>◦ RD 工程師</li></ul></li><li>• 蒐集題材，為攻讀博士做準備</li></ul>                                                 |

# 讀書計畫和說明

## 特殊選才後

- 繼續漏洞挖掘：應用進階漏洞利用技術，拓廣認識的攻擊面。
- 學習惡意軟體開發及偵測技術：購買 Maldev Academy 的課程學習惡意軟體技術。並嘗試學習撰寫過濾 pattern。
- 研讀數學：研讀大學需要用到的數學，並在學習後將之前看不太懂的書 *Concrete Mathematics* 看熟。
- 補強資料結構：對於資料結構的操作不夠熟悉，在這時候補強。

## 大學時期規劃

- 加入大學社團：由於高中時過度執著於研究技術，且作業系統不是同齡中熱門的話題，導致幾乎沒有參加社群討論，因此我希望可以在大學時彌補遺憾。
- 重新開始學習 kernel：之前失敗的 kernel 研究，在大學研究過相關知識後重新開始。
- 利用 MOOC 平台拓展專業領域：使用 coursera 等平台，探索可以學習的專業領域，加深技術的同時也開拓視野。
- 大量閱讀書籍：實做能力是理論基礎的擴展，我將延續高中時期大量閱讀的習慣，向理論和實務書籍學習。
- 搭建部落格撰寫技術文章：高中時只有記錄筆記的習慣，大學之後我計畫在部落格上撰寫技術文章並分享所學。
- 繼續學習英文並重新參加 IELTS 考試：用更嚴謹的方式學習英文，並取得未來國際交流的語言資格。
- 進入企業實習：我打算在大學三年級的時候開始應徵實習生，為之後工作準備，並擬定研究所研究方向。
- 投入自由軟體開發：向社群貢獻自己的資訊和資安知識。

## 大學畢業後規劃

- 考取研究所：在大學時期規劃好的企業實習中，我希望可以看到資訊產業的需求，並且以更貼近現實的角度進行研究。

# 讀書計畫和說明

## 研究所時期規劃

- **追蹤各大頂尖會議並大量閱讀論文：**追蹤 USENIX 等各種主題的會議論文，觀摩前輩的研究結果同時[提高自己的上限](#)。
- **爭取期刊論文發表機會：**整合所學理論知識並尋找研究議題，[協同教授指導](#)，嘗試投稿期刊論文累積學習成果。
- **將部落格文章整理成系列教學文：**把大學累積的心得文章和技術文章整理，製作成有助於後輩學習的材料。
- **為工作累積實力：**不斷累積技術和 [Domain know-how](#) 之外也[學習學習的能力](#)。在未來進入職場後能夠不斷吸收新知，而非停滯不前。

## 研究所畢業後規劃

- **應徵工作：**最終選擇取決於大學和研究所的主要研究方向。
  - **資安職位：**除了投入攻擊方的陣線之外，我也考慮[加入防禦方](#)。
    - **藍隊（例如：SoC 分析師）：**從事[網路防禦和威脅分析等工作](#)，休閒時間為自由軟體[貢獻程式碼](#)的同時也以獨立漏洞研究員的身分[發表資安研究](#)。
    - **紅隊（例如：漏洞研究員）：**主要進行[漏洞研究](#)，向軟體貢獻出安全相關的修補。
  - **研發工程師：**工作研究和開發之餘[貢獻程式碼至開源專案](#)，並且融入工作專業和資安知識，在網路論壇或社群中[引導和幫助需要幫助的學習者](#)。
- **摸索博士學位研究主題：**以博士學位作為遠期目標，在[工作之餘持續觀察產業趨勢](#)，[找出待解決的問題](#)當作研究主題，為突破現存的問題做準備。（讀博士的意義參考 [The illustrated guide to a Ph.D.](#)）

# 成績單和科學班證明

高中歷年成績單（附件：掃描檔案）

科學班就讀證明（附件：掃描檔案）



# 培訓和活動心得

---

# AIS3 2022 最佳專題

透過 MyFirstCTF 的名次取得了進入為期七天的 AIS3 暑期營隊的資格。透過營隊的活動內容，我了解自己技術能力及未來努力的方向。

我的習慣是每一次參加活動後都會把需要加強的主題紀錄起來，並且回家買書自我精進（或補救）。這一次我看到了一群比我厲害很多的大學生和研究生，可以和他們一起進行活動真的是少有的契機。

我們最後決定挖掘無線路由器的漏洞。在分組專題中，我與組員互相支援合作，我負責尋找目標、拆開韌體、架設模擬環境以縮小組員的目標範圍。我們找到二個 buffer overflow 和 broken access control 漏洞，並獲得最佳專題獎，讓我不僅學到技術，更增進了團隊合作和協調能力。



宗成教授（右一）在 AIS3 2022 向我的組員和我（左一）頒發最佳專題獎。



合格證明書



最佳專題獎

# EOF 2024

經過三年後，我終於進入了 2024 年的 AIS3 EOF 的決賽。在決賽中，我負責 binary exploitation 和逆向工程相關題目，可惜在第一天晚上在被賽所以完全沒有睡覺，導致 binary exploitation 表現失常，結果不盡人意。

我在這一次的比賽後深刻反省為什麼三年過去了，看到的攻擊面卻這麼少？我得到的結論是解題式的 CTF 比賽限制了想像，讓很類似真實世界的漏洞在眼前卻無法識別出來。

我在這一次的比賽後開始投入真實世界漏洞的研究，從今年初至現在，已經累積了 7 個 CVE，讓我知道那時候選擇的方向是正確的，而且正在那一條正確的道路上持續前進。



我（左一）與隊友們在 2024 年度 AIS3 EOF 決賽後合影



決賽參賽證明

# 台灣好厲駭

我在 AIS3 2022 之外還參加了三屆的台灣好厲駭。其中影響我最大的是好厲駭的培訓，在培訓中有各式的主題的講座和課程，這些課程除了學到了各種攻擊、防禦和管理技術外，也點明了之後學習的方向。而且每個學年都會有自評表填寫，也提供了追蹤自己變化的機會。

另外，我在這些課程中也看到了許多比我優秀的同齡人和學長，透過不斷的提高自己看到的上限，讓我可以訂定出更明確的目標。



# AIS3 Junior

我認為參加 AIS3 Junior 讓我認識同齡的資訊安全愛好者，也是一個互相切磋交流的好機會。在營隊的最後也都有專題發表，是展現分工合作的時候，也可以觀摩其他組別製作的簡報是否有可以學習之處。

每次參加營隊後，我會列出一張需要補強的清單，例如在 AIS3 Junior 2023 年，製作的專題主題是中間人攻擊，但是由於知識不足沒有辦法完整做出，於是我回家之後研讀了基礎的密碼學（當時看的是 Cryptography and Network Security: Principles and Practice）和 TCP/IP 協定組（當時看的是 TCP/IP Illustrated）等等，才知道當時的想法根本是天馬行空、缺少技術細節的空談。



我（左三）與助教（右二）、隊友們在 AIS3 Junior 2024 合影



# 作品、成果和證照

---

# 漏洞列表：簡述

我回報的 CVE 中，有 3 個被歸類為 critical、3 個被歸類為 high、另一個則是 medium (FIRST CVSS 3.1)。除了 CVE-2024-6044 之外全部都是 [pre-auth RCE](#)，影響全球所有使用該機型的使用者。

為什麼會開始漏洞挖掘呢？主要因為在 AIS3 EOF 決賽的時候發現自己看到的攻擊面太少，更具體來說，[很多顯而易見的漏洞就這樣子在比賽中和我擦身而過](#)，這讓我反省自己的學習方式是否出了問題。

以我熟悉的逆向工程和 binary exploitation 來說，比賽題目範圍相當有限，而且[會利用漏洞真的代表找得到漏洞嗎？](#)我的答案是否定的。解題式 CTF 的 pwn 題可能只有幾百行，相對於編譯後有數 MB 的真實世界的程式來說真的很小。

為了挖掘漏洞，我在學校晚自習結束後回家，就馬上開始進行逆向工程。我藉由[壓縮睡眠時間來投入漏洞研究](#)、常常持續到凌晨一、兩點。這些[不中斷的努力](#)最後讓我成功踏入漏洞研究的領域。

除了技術上的進步，我也學會如何評估漏洞的嚴重性，撰寫完整的報告，並站在廠商的角度提供修補意見。我也[遵循 responsible vulnerability disclosure](#) 的標準，沒有在網路上公開發布 exploit 和細節，以免增加終端使用者被利用的風險。



The screenshot shows the iThome website's news section. The header includes the iThome logo and navigation links for News, Products & Technology, Special Topics, AI, Cloud, Medical IT, Security, Conferences, Communities, and IT Experience. The main article is titled 'D-Link修補Wi-Fi路由器高風險漏洞' (D-Link patches high-risk vulnerabilities in Wi-Fi routers). The sub-headline states: 'D-Link公布旗下3款路由器的5項漏洞，這些漏洞涉及記憶體緩衝區溢位、隱藏功能，以及作業系統命令注入' (D-Link announced 5 vulnerabilities for 3 of its routers, involving memory buffer overflows, hidden functions, and OS command injection). The author is '文/ 周峻佑' and the date is '2024-09-18 發表'. The article content lists the following CVEs: 'DIR-X4860 / DIR-X5460 / COVR-X1870 :: TWCERT - TVN-202409021 / TVN-202409022 / TVN-202409023 / TVN-202409024 / TVN-202429025 Vulnerabilities reports'.

我的研究成果登上 iThome 的資安新聞了！

# 漏洞列表

## CVE-2024-{6044, 6045} 起源

破解 D-Link EAGLE PRO 系列韌體加密機制的故事

# 漏洞列表

CVE-2024-6044 (Arbitrary file read)  
path traversal / improper input validation

# 漏洞列表

CVE-2024-45694 / CVE-2024-45695 (RCE)  
stack-based buffer overflow

# 漏洞列表

CVE-2024-45697 (RCE)

misconfigured iptables rule

# 漏洞列表

CVE-2024-45698 (RCE)

command injection / hidden functionality

CVE-2024-6045 / CVE-2024-45696 (RCE)

hard-coded credentials / b4ckd00r 📄

# 漏洞列表

## ZD-2024-00904 (sandbox escape)

zerojudge.tw sandbox escape

minidlna: blind ssrf

## toybox: path traversal (Arbitrary file creation)

由於簡化路徑的邏輯錯誤，導致用 wget 發送請求時，攻擊者如果傳送惡意的回應，可以建立任意檔案。

由於不是重大安全性問題，因此這一次回報用 [pull request](#) 的形式直接提供 patch，也引用了多個 RFC 來說明不合規於協定的行為。

If a Content-Disposition header is received and --output-document (or simply -O) is not specified, the creation of an arbitrary file can be triggered (not an overwrite of any existing files).

I think it's more like an undesirable behavior, rather than a vulnerability, since:

1. it requires user interaction and a request to an attacker-controlled server.
2. it does not overwrite any file, and without much information about the victim, it's very unlikely to pose a serious threat to integrity of victim's machine.
3. executable bit of the output file is not set, there's no direct way for the user to execute the downloaded file.
4. I did see an attempt to validate the Content-Disposition variable (which seems not functioning). So instead of returning an error when a `/` is encountered, the patch ignores characters before last `/` (if any).

### reference

[rfc 2183 section 5](#)

[rfc 2616 section 15.5](#)

[rfc 2616 section 19.5.1](#)

pull request 以攻擊者的角度提供了可能造成的危害，並以開發者的角度提供 patch

# 漏洞挖掘：下一步？

由於 IoT 設備程式碼太過鬆散，因此下一步我將嘗試不同的目標進行漏洞開發的學習，計畫如下：

## 近期：間歇的學習-練習-修正 循環

- 學習後練習，練習後學習：知識和經驗可以同時間增加
- 理解更多程式語言：熟悉更多程式語言，可以檢閱的弱點也會更多
- 站在不同角度看攻擊面：站在開發者或攻擊者的角度看攻擊面

## 近期至中期：嘗試大型開源專案

- 嘗試較大型的開源專案
  - 各種 service daemon（例如：IRC、FTP ...）
  - 日常使用的小工具
- 在學習之餘，也做出貢獻

## 中期：學習 kernel exploitation 技術

- 搭設漏洞復現環境，由已知漏洞學習
- 完成 [pwn.college](http://pwn.college) 的 system exploitation 類別
- 了解不同架構的 kernel exploitation

## 中期：了解各種主流平台利用手法

- 了解 Windows 系統和 API: *Windows via C/C++, Windows internals*
- 了解 OS X 系統架構: *Mac OS X internals*

## 中期至遠期：深入了解駭客文化

- 閱讀各種駭客文學作品
  - *PoC // GTFO*
  - *Underground: Tales of hacking, madness and obsession on the electronic frontier*
  - ...



## 可解析 FAT32 的 MBR bootstrap code

在 400+ bytes 的限制下用純組合語言寫成的 MBR

BIOS上電自檢（POST）後，便會把 sector #0 載入到 0x7c00 並轉交控制權，而 MBR 就是位於 sector 0 的開機程式（stage 0 loader）。此段程式雖然只有 500 上下個位元組，卻要完成載入 stage 1 loader 的任務（stage 1 loader 會再載入 kernel 執行）。

由於教學通常都是用 FAT 12/16 做範例，因此我挑戰可以解析 FAT32 的 MBR。為此，我查看了微軟對於 FAT32 規格的文件（在 Internet Archive 上找到的），並詳細閱讀 BIOS 資料區的記憶體布局。這一次的試驗不僅學到如何精簡組合語言，也強化了技術文件閱讀的能力，算是一次非常難得的嘗試！

## 探索 kernel!

費時兩年的摸索

[source code](#)

從國中三年級到高中二年級的這段期間，我把大部分的時間投入在摸索 kernel，這是一個成長的過程。我摸索過的部份較多是硬體界面，包括：

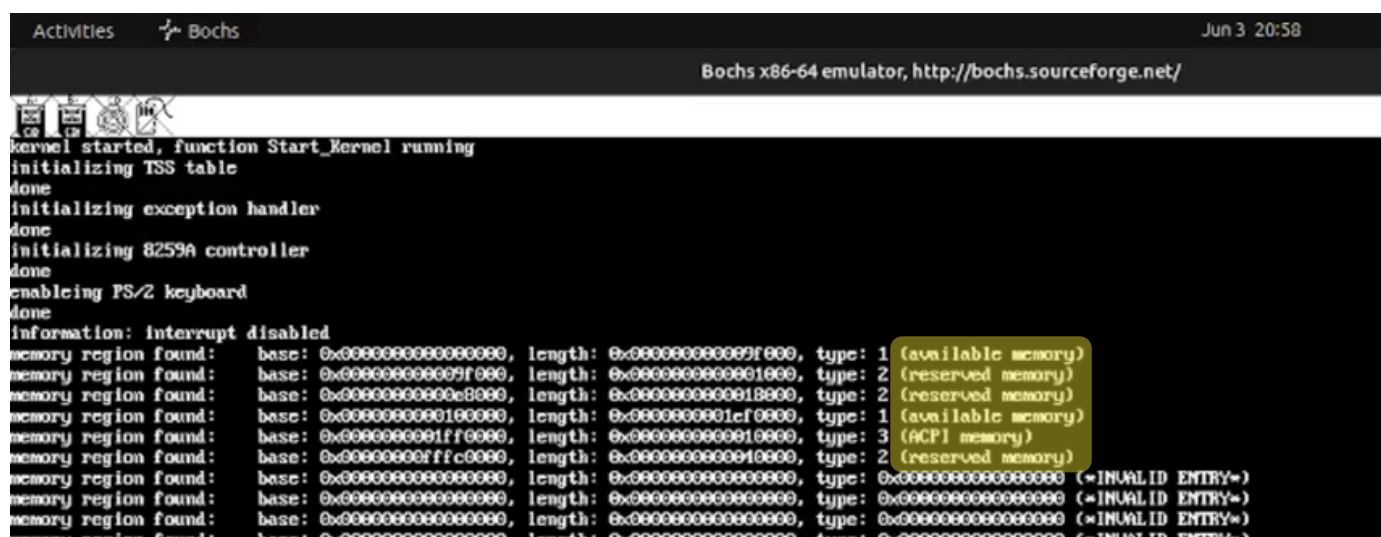
- 中斷處理
- 鍵盤解析程式
- SVGA 螢幕輸出（字體和類似 printk 的函式）
- 記憶體區塊偵測

是到了後來才知道 kernel 和 OS 的差異，因此當時取名為 system\_project，而不是 kernel\_project。回顧過去所寫的程式，才發現一路走來所累積到的點滴，已將現在的我推到不同的高度！

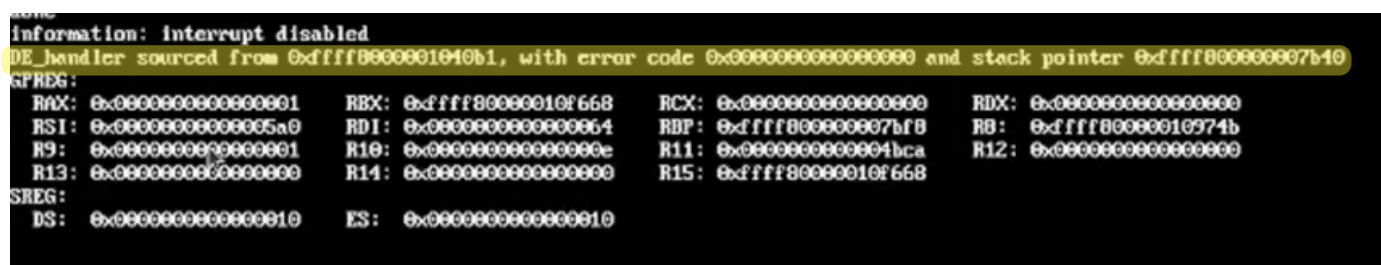
目前我已經轉成學習 Linux kernel module 開發和 Linux kernel 內部設計，目前還在摸索階段。我規劃之後將會融合我的軟體安全專長，投入 kernel 的軟體安全研究。



控制權轉交給 kernel



用 BIOS 提供的 e820 中斷偵測可用的記憶體區塊



Oops! 除零錯誤



可以從鍵盤讀取輸入了，圖中正在輸入 Hello, World

# 下一步？

## 研究 Linux kernel 並開發 kernel module

研究最被廣泛使用的核心

- 理解設計的巧思
- 了解 scalability 的各種考量
- 透過修改 kernel 學習

## 研讀經典書籍和教材

按部就班的學習

向一些非概念性的經典 課程/書籍 學習

- xv6
- *Operating Systems Design and Implementation*
- *Operating System Design: The Xinu Approach*

## 設計 kernel 各部件

畫出藍圖

- 確定 kernel 的主要部件
- 設計各部件的交互方式
- 如何有效的管理資料結構

## 從 CISC 轉向 RISC

著重 kernel 學習，而非硬體界面

- x86 有一些 legacy problem 和極度複雜的硬體界面，學習門檻較高
- RISC 架構的硬體介面相對簡單，是適合學習 kernel 運作原理的平台
- 例如：Arm / AArch64 甚至更精簡的 RISC-V

## 研究 RTOS

除了 GPOS 之外的研究

- 了解如何把作業系統技術整合至各種對延遲有高度要求的場景
- 學習 hard real-time OS 的設計巧思和真實應用

# CND 網路防禦專家認證

我對於攻擊技術比較有興趣，但是以前從來都不知道「防禦」是怎樣的**概念**，這也是我為什麼選擇這一門課的主因。

對我來說，這一次的訓練只是一個開始，更細節的部份還要再繼續學習之後才會了解，藉由這一次的考試分析後發現**我比較擅長技術面，而管理面的知識還要加強**，畢竟人也是很大的攻擊面（ID-10T 問題）。

攻防本來就是一體兩面的事，**攻擊者只需要找到一個點，而防禦者要防禦整個攻擊面**，所以我現在如果能夠看到更多的攻擊面，增加自己的見識，就有機會更全面的檢閱攻擊者可能的進入點。

