

# Just for noting (<https://m157q.github.io>)

Just for noting (<https://m157q.github.io>)

[Categories \(/categories\)](/categories)   [Archives \(/archives\)](/archives)   [Tags \(/tags\)](/tags)

[About \(https://m157q.github.io/pages/about/\)](https://m157q.github.io/pages/about/)

🔍 Search

## HITCON 2015 IOT Wargame (<https://m157q.github.io/posts/2015/08/29/hitcon-2015-iot-wargame/>)

今年的 HITCON 的主題是 IoT，  
所以 Wargame 也配合了 IoT 的主題，  
用 Arduino Nano + Raspberry Pi 出題。

Wargame 的網址是 <http://iot.hitcon.org> (<http://iot.hitcon.org>)

**IOT Wargame**  
<http://iot.hitcon.org/>

Challs Challengers Register Login

NANO 1	NANO 2	NANO 3	NANO 4	-
1	1	1	-	-

**NANO 1**  
1 points

Desc  
解開摩斯電碼  
(Decrypt the Morse Code)

Link  
[nano-1.tgz](#)

Enter key Send

Announcement

- Raspberry Pi 挑戰賽規則請詳閱 [規則頁面](#)
- HITCON NANO 積分競賽規則請詳閱 [規則頁面](#)

# Arduino Nano 部份

Arduino Nano 的題目只有三題，  
(原本好像有打算出四題的感覺，但可能時間來不及的關係)  
而且難度蠻簡單的，  
其實蠻適合所有會眾入門，  
而且每位會眾報到後都會拿到已經把題目燒進去的 Arduino Nano。  
(還有附排針，但題目不會用到，原本一看到以為要現場焊接XD)  
但參與度好像不高，  
寫這篇文章的當下，  
總計是 243 位註冊帳號，  
其中 161 位至少答對一題，  
三題全部都做完的有 69 位。

我自己是在第一天議程結束後的晚上才開始看題目，  
然後大概花3個多小時才把三題都解出來，  
(GeoHot 花了一個多小時就全解完了QQ)  
以下寫一下自己的解法，  
供有興趣的人參考，  
有問題也歡迎回覆。  
  
其實題目真的不難，  
真的有在打 Wargame 或 CTF 的人應該覺得簡單到沒必要浪費時間寫 Write-up 吧XD  
但因為身邊還是有人不會寫，  
所以秉持著推廣一下 Wargame 跟 CTF 的精神寫一下這部份，  
Orange 出的 RPi 題目我就解不出來了 XDrz

## 1. 讓 Arduino Nano 能被你的電腦讀到

老實說這應該是最難的部份了，  
因為第一次接觸 Arduino，  
所以我在這上面花蠻多時間的Orz  
IRC 上也很多人在詢問，  
有看到有人在問 Mac, Windows  
就是沒看到有人問 Linux 的，  
(噢，好像有看到有人問 Ubuntu 的?有點忘記了)  
因為我本身用 Arch Linux，  
所以這篇文章會以 Arch Linux 為主，  
(順便偷偷推廣 [Arch Linux \(https://www.archlinux.org/\)](https://www.archlinux.org/) XD)

Windows 的使用者可以參考會眾 toby 寫的教學  
[HITCON 之 Windows 也要玩 Nano « Toby 'n Hack \(http://toby.logdown.com/posts/293552/hitcon-play-nano-on-windows\)](http://toby.logdown.com/posts/293552/hitcon-play-nano-on-windows)，  
仍然不行的話，可以看一下官方的 [Arduino - Windows \(https://www.arduino.cc/en/Guide/Windows\)](https://www.arduino.cc/en/Guide/Windows) 試試。  
  
Mac 的使用者可以參考這篇 [Arduino Nano no serial port for MacBook Air 2013 - Arduino Stack Exchange \(http://arduino.stackexchange.com/questions/5119/arduino-nano-no-serial-port-for-macbook-air-2013\)](http://arduino.stackexchange.com/questions/5119/arduino-nano-no-serial-port-for-macbook-air-2013) (但有人反應裝了還是讀不到)  
仍然不行的話，也可以看一下官方的 [Arduino - MacOSX \(https://www.arduino.cc/en/Guide/MacOSX\)](https://www.arduino.cc/en/Guide/MacOSX) 試試。  
  
至於 Linux 的話，可以參考 Arduino 官方的這篇 [Arduino Playground - Linux \(http://playground.arduino.cc/Learning/Linux\)](http://playground.arduino.cc/Learning/Linux)。

而 Arch Linux 的話，當然就是看官方的 [Arduino - ArchWiki \(https://wiki.archlinux.org/index.php/Arduino\)](https://wiki.archlinux.org/index.php/Arduino)，從 AUR 上安裝 Arduino。

(但其實我在還沒安裝 AUR 的 Arduino 時，就已經出現 `/dev/ttyUSB0` 了，所以我覺得應該是有 driver 的，但我當時還不知道怎麼改 `ans.py` 的 port，所以還是裝了 AUR 的 arduino 套件，直到看到 [Arduino Playground - Python \(http://playground.arduino.cc/Interfacing/Python\)](http://playground.arduino.cc/Interfacing/Python) 才發現是 `port` 的值錯了才覺得一開始可能不用裝 `arduino` 也會動，但也懶的移除了。所以如果有人也有這樣的情況，然後也還沒裝 `arduino` 的話，可以先試試看直接改 `ans.py` 的 port，看能不能用，這樣就不用多裝 3xx MB 的 arduino 套件了。至於怎麼改 port，下面會提到)

如果沒有出現 `/dev/ttyUSB0` 或是試了不成功的話，就繼續試試

```
yaourt -S arduino
sudo gpasswd -a $user uucp
sudo gpasswd -a $user lock
sudo gpasswd -a $user tty
sudo modprobe cdc_acm
```

之後，登出再登入。

ArchWiki 上是說會出現 `/dev/ttyACMx` (x 為數字)

但我的是 `/dev/ttyUSB0`，一樣可以正常使用。

用 `screen /dev/ttyUSB0 115200` 確認有顯示以下內容就是有讀到了

```
HITCON 2015 NANO GAME

Nano$ [0] Morse
Nano$ [1] Maze
Nano$ [2] Calculator
Nano$ enter your choice:
```

## 2. 透過 pip 安裝 pyserial

在使用官方提供的 `ans.py` 之前，記得先安裝必要的套件 `pyserial` 使用 `pip install pyserial` 安裝完後，打開 `ans.py` 將第 67 行的 `port=3` 改成 `port='/dev/ttyUSB0'` (這邊的 `/dev/ttyUSB0` 視實際情況更改) 就可以開始修改 `ans.py` 進行 Python Wargame 了(?)

修改第 47 行的 choice 可以選擇要解的題目，

```
'0' == NANO 1
```

```
'1' == NANO 2
```

```
'2' == NANO 3
```

每個 \*.tgz 用 `tar zxvf $tgz_file` 解開

每個裡面的 readme.txt 基本上都一樣的，不太需要看

每題拿到 key 之後，

請複製該 key，

再到 <http://iot.hitcon.org> (<http://iot.hitcon.org>) 登入，

然後點選對應的題目，  
至右下角 `enter key` 欄位，  
貼上 key，之後按 send 送出。

以下開始講解各題：

## NANO 1

[NANO 1 原始檔 \(/files/hitcon-2015-iot-wargame/nano/nano-1.tgz\)](/files/hitcon-2015-iot-wargame/nano/nano-1.tgz)

這題會拿到一串摩斯密碼 (<https://zh.wikipedia.org/zh-tw/%E6%91%A9%E5%B0%94%E6%96%AF%E7%94%B5%E7%A0%81>)

```
.... . - .-. --- -. . - . - - - . - - - . - - - . - . . . .
```

隨便找個解摩斯密碼的網站 <http://morsecode.scphillips.com/translator.html>  
(<http://morsecode.scphillips.com/translator.html>)

解碼後得到 `HITCONNANOGAMEMORSE`

把 `game0()` 的 `answer = ''` 改成 `answer = 'HITCONNANOGAMEMORSE'`

再執行一次 `ans.py` 便可得到 key

```
Nano$ key is 457E1A295B9D1C48
```

## NANO 2

[NANO 2 原始檔 \(/files/hitcon-2015-iot-wargame/nano/nano-2.tgz\)](/files/hitcon-2015-iot-wargame/nano/nano-2.tgz)

這是三題裡面最難的一題，也是三題中最少人解出來的。

這題要在九秒內走出 `14*14` 的迷宮，

從 `(0, 0)` 走到 `(14,14)`

(除非你手速夠快，不然以本題的設計基本上九秒走不完)

'O' 代表人、'.' 代表路、'+' 代表牆、'X' 代表出口  
'w' 往上走、'a' 往左走、's' 往下走、'd' 往右走

必須修改 `game1()` 裡頭的內容，  
把走迷宮的演算法寫在裏面。

我用的方法其實是半自動的，在必要的時候需要手動，也不保證每次成功。

(要全自動的話可能需要用 BFS ([https://en.wikipedia.org/wiki/Breadth-first\\_search](https://en.wikipedia.org/wiki/Breadth-first_search))，但我太廢，覺得這樣已經能解出題目就夠了。)

程式每次會呈獻 3x3 的現在位置地圖，

因為 'O' 每次的位置不固定，

所以先找到 'O' 的現在位置後，

將其紀錄為 `pos_x` 和 `pos_y`，

方便之後判斷其上下左右是否可走。

再來因為我們要往右下方行走才能到達出口，

所以就判斷如果 'O' 的右邊(`lines[pos_y, pos_x+1]`) 是 '.'(路) 的話就往右走，

已右邊舉例的話，如果有下列情況的話，就改換下一個優先的方向檢查：(按照 右、下、左、上 的順序)

- 邊界(`pos_x == 2`)(`pos_x+ 1` 就超出邊界了)
- 'O' 的右邊是 '+' (牆)
- 上一步是往左走(往反方向走)
  - 避免左右左右、上下上下這種不會前進的狀況發生
  - 造成了另一個問題，每當走入三面都是牆的洞的時候，就會卡住。
  - 這時就會需要手動移動
  - 因為要得知上一步，所以我建立了 `steps = [None]` 紀錄每一步的順序
  - 把每次的移動利用 `steps.append()` 紀錄起來
  - 並在每次透過 `steps[-1]` 得知上一步

依此類推，照著 右、下、左、上、手動 的先後順序，寫出程式碼。  
開始執行後，遇到卡住的狀況就手動控制，試個幾次很快就可以走出迷宮

```
Nano$ key is 3D52CB746F9E6C83
```

## NANO 3

[NANO 3 原始檔 \(/files/hitcon-2015-iot-wargame/nano/nano-3.tgz\)](/files/hitcon-2015-iot-wargame/nano/nano-3.tgz)

這題是必須在一秒內回答四則運算的結果，  
看一下題目，可以發現題目的字串長的像這樣

```
Nano$ 7 * 10 - 14 + 6 - 5 - 19 = ?
```

所以可以透過 Regular Expression

[7.2. re — Regular expression operations — Python 2.7.10 documentation](https://docs.python.org/2/library/re.html)

<https://docs.python.org/2/library/re.html>

把 `7 * 10 - 14 + 6 - 5 - 19` 取出來後

丟給 `eval()` (<https://docs.python.org/2/library/functions.html#eval>) 算出結果，

再把值 assign 給 `answer` 回答。

(`eval()`) 可以把字串當作 expression 進行求值，雖然很方便，但真正在開發程式的時候最好不要亂用，尤其是不要把使用者輸入的資料丟給 `eval()`，因為你永遠不知道使用者會輸入什麼奇怪的東西)

照著這個想法修改一下 `game2()`，這題應該就解決了

```
Nano$ key is 273A9C1E2D380B48
```

極度建議自己寫程式碼，

真的真的想不出來的話再點下面的連結吧！

不然是不會進步的。

程式碼請見：<https://gist.github.com/M157q/164f9ba5bd8ae0b18afe>

<https://gist.github.com/M157q/164f9ba5bd8ae0b18afe>

然後看到 IRC 有人提到，

他把 Nano 上的 binary dump 到電腦上，

直接用 `strings` 拿到 3 個 key，

不知道怎麼做到的

查了一些關鍵字，會是 `avr-objdump` 嗎？

補上直接 dump 的 write-up [HITCON Knowledge Base — 秒解 Hitcon Nano 題](#)

(<http://kb.hitcon.org/post/128246307977/%E7%A7%92%E8%A7%A3-hitcon-nano-%E9%A1%8C>)

## Raspberry Pi 部份

有分四題：R0, R1, R2, R4 的題目都不同，都是由 Orange 出題。

我只有碰 R0 跟 R1 的題目，有些想法，但都沒解出來，以下簡單紀錄。

### R0

Web 那邊會拿到一個 [ARM64 的 binary 執行檔 \(/files/hitcon-2015-iot-wargame/rpi/r0/forkyou\)](/files/hitcon-2015-iot-wargame/rpi/r0/forkyou)，

之後用 `qemu-aarch64` 的環境就可以執行，

聽說還蠻簡單的，可是我還是解不出來QQ

似乎有在某處看到 `/bin/sh -c` 之類的呼叫，

應該是利用 BOF 把 EIP 指到那邊就可以拿到 shell 了？

補上 R0 write-up [HITCON Knowledge Base — HITCON 2015 IoT Wargame – R0 挑戰題](#)

(<http://kb.hitcon.org/post/127947378507/hitcon-2015-iot-wargame-r0-%E6%8C%91%E6%88%B0%E9%A1%8C>)

### R1

連進去是 Discuz! X3.2

查了一下，有 remote shell execute 漏洞：

[DiscuzX系列命令执行分析公开（三连弹） | WooYun知识库 \(http://drops.wooyun.org/papers/4611\)](http://drops.wooyun.org/papers/4611)

照著做之後，

找不到在 `data/attachment` 底下符合權限的圖片來編輯，

(最近重灌，還沒裝 DirBuster 或是 Burp Suite，所以沒暴力掃底下有哪些東西...)

無法觸發 picwidth 的 Remote Shell Execution 漏洞，

感覺方向應該是對的吧？

這題是四題裏面沒被人解出來的。

(GeoHot 把其他三題都解掉了...)

---

這次的 Wargame 大概就是這樣吧，

果然實力還是不夠，

HST 的 Wargame 拿到 reversed.txt 裡頭一串 ...---- 後也沒啥想法。

會把 Nano 的部份寫的這麼詳細的原因，

是因為想要推廣大家玩一下 Wargame 吧！

(覺得有趣的話就可以來打 CTF 了)

畢竟上面也講了，這次 Wargame 真的不算難，

而將近 1000 人的會眾，只有不到 300 人註冊，只有 161 人有拿到分，

這比例實在有點少。

再加上幾天前在 Facebook 上看到有人說 Nano 一片一片燒，燒到手快斷了，

覺得這麼少人玩實在有點可惜。

處理好 Arduino Nano 跟電腦連接之後，

基本上就是寫 Python Code，

所以我才戲稱是 Python Wargame，

但其實沒學過 Python 的人藉著這個機會學一下 Python 也不錯，

畢竟因為 Python 的方便性，很多 exploit 都用 Python 寫了。


總之，這篇就是一個小廢物的流水帳。


有問題歡迎留言討論，但我不一定會就是T\_T。

---

Share on:  (<https://twitter.com/home?status=HITCON%202015%20IOT%20Wargame%0Ahttps%3A//m157q.github.io/posts/2015/08/29/hitcon-2015-iot-wargame/%0Avia%20%40M157q%0A>)

 (<https://www.facebook.com/sharer/sharer.php?s=100&p%5Burl%5D=https%3A//m157q.github.io/posts/2015/08/29/hitcon-2015-iot-wargame/>)

 (<https://plus.google.com/share?url=https%3A//m157q.github.io/posts/2015/08/29/hitcon-2015-iot-wargame/>)

 (<mailto:?subject=HITCON%202015%20IOT%20Wargame&body=https%3A//m157q.github.io/posts/2015/08/29/hitcon-2015-iot-wargame/>)

---

## Related content

- [\[Python\] Sort dictionary by key or value \(https://m157q.github.io/posts/2013/05/10/python-sort-dictionary-by-key-or-value/\)](https://m157q.github.io/posts/2013/05/10/python-sort-dictionary-by-key-or-value/)
- [pip install lxml fail on MacOS \(https://m157q.github.io/posts/2015/02/04/pip-install-lxml-fail-on-macos/\)](https://m157q.github.io/posts/2015/02/04/pip-install-lxml-fail-on-macos/)
- [Magic of Python list slicing \(https://m157q.github.io/posts/2015/05/25/magic-of-python-list-slicing/\)](https://m157q.github.io/posts/2015/05/25/magic-of-python-list-slicing/)



- [Note] Learn Python! 開放源碼的動態程式設計語言體驗營 (<https://m157q.github.io/posts/2013/04/09/note-learn-python-kai-fang-yuan-ma-de-dong-tai-cheng-shi-she-ji-yu-yan-ti-yan-ying/>)
- SITCON 2015 - Android Repackaged App Detection System (<https://m157q.github.io/posts/2015/03/11/sitcon-2015-android-repackaged-app-detection-system/>)

0 Comments    m157q-blog

1 Login ▾

♥ Recommend    ↗ Share

Sort by Newest ▾



Start the discussion...

Be the first to comment.

ALSO ON M157Q-BLOG

WHAT'S THIS?

## OCTF 2015 Note

1 comment • 6 months ago



Alvaro Muñoz — Do you mind sharing the final padding attack script. Thanks!

## FOSS week1

4 comments • 7 months ago



jserv — LLVM 的授權是 BSD License，沒有一定要 GPL 形式的強制釋出原始碼條款，但 llvm-gcc 實際上是一種「掏空」GPL 授權的 gcc 的方式，也就是讓 BSD ...

✉ Subscribe

🗨 Add Disqus to your site

🔒 Privacy

📅 Sat 29 August 2015 (2015-08-29T21:11:33+08:00)

✍ Thu 03 September 2015 (2015-09-03T21:32:19+08:00)

👤 By m157q (<https://m157q.github.io/author/m157q.html>)

Write-up (<https://m157q.github.io/category/write-up/>)

Python (/tag/python/)

Arduino (/tag/arduino/)

Raspberry Pi (/tag/raspberry-pi/)

HITCON (/tag/hitcon/)

## Social

GitHub (<https://github.com/M157q>)

Twitter (<https://twitter.com/M157q>)

Google+ (<https://plus.google.com/u/0/+SYJheng/posts>)

## Links

[Pelican \(http://getpelican.com/\)](http://getpelican.com/)

[Python.org \(http://python.org/\)](http://python.org/)

[Jinja2 \(http://jinja.pocoo.org/\)](http://jinja.pocoo.org/)

## Browse content by

 [Categories \(https://m157q.github.io/categories/index.html\)](https://m157q.github.io/categories/index.html)

 [Dates \(https://m157q.github.io/archives/index.html\)](https://m157q.github.io/archives/index.html)

 [Tags \(https://m157q.github.io/tags/index.html\)](https://m157q.github.io/tags/index.html)

 [Feed \(https://m157q.github.io/feeds/write-up.atom.xml\)](https://m157q.github.io/feeds/write-up.atom.xml)

## Copyright notice

© Copyright 2013-2015 m157q.

## Disclaimer

All opinions expressed in this site are my own personal opinions and are not endorsed by, nor do they represent the opinions of my previous, current and future employers or any of its affiliates, partners or customers.

[↑ Back to top](#)

Site generated by Pelican (<http://getpelican.com/>).

Plumage (<https://github.com/kdeldycke/plumage>) theme by Kevin Deldycke (<http://kevin.deldycke.com>).