中間人攻擊

B073040047 楊志璿　　M093040096 李天朗　　M093040094 王文濤

January 18, 2022

Contents

1 摘要

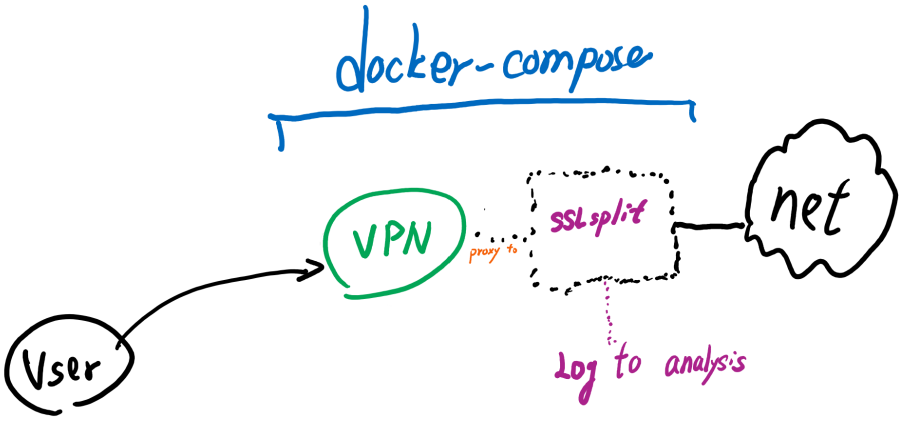
這是一篇關於中間人攻擊的期末報告，題目要求必須要實做 HTTPS 連線之竊取帳號密碼。本篇報告介紹本期末專案的：HTTPS 中間人攻擊、即時密碼測錄、實做成果與中山大學網路大學之登入安全探討。

本專案開放原始碼於:https://github.com/25077667/Man-in-the-Middle-Attack

2 本文

建議具有中間人攻擊之相關背景知識，再來閱讀此篇報告。關於中間人攻擊之相關內容，礙於篇幅，本文將不再贅述。
藉由無知使用者，在不檢查憑證之情況下，實現隱形中間人攻擊。

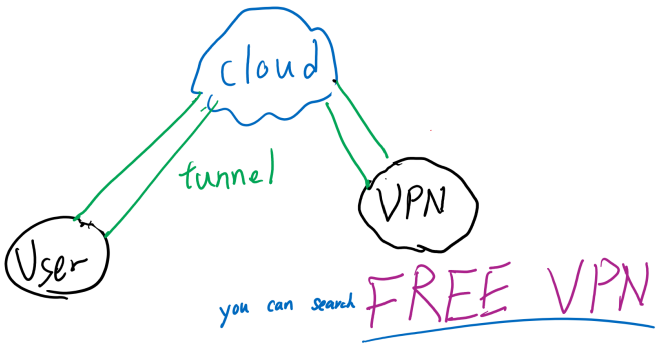
2.1 架構



docker-compose 部份，僅用於「重現」方便，沒有限制必須使用 docker 相關技術。

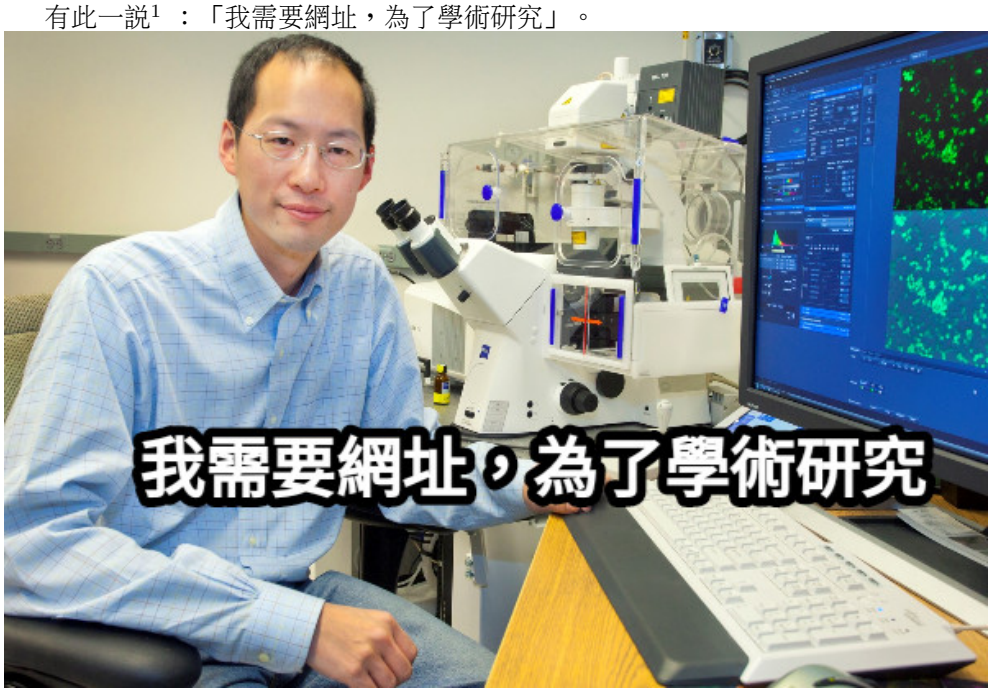
在使用者連上 VPN 之情況下，可以使用 iptables 於VPN 後端，將 VPN 所接收之資料，交給 sslsplit 代理轉發。實際上，使用者是無法知道 sslslpit 之存在的，因為 VPN 本身即是一種代理(proxy)。

2.2 攻擊情境



使用者想要使用免費之中山大學學術網路服務，中山大學之 ip 有付費訂閱許多期刊，以利學術研究。於是使用者發現有一個不用中山大學學生證的「免費」 VPN 即可以使用中山大學之 ip，於是使用者便直接陷入本次實驗之中間人攻擊陷阱。

有此一說[1] ：「我需要網址，為了學術研究」。



因此本專案作為學術用途，連線上中山大學之學術網路，以利學術研究。或者可以支持「開放學術資源[2]」、「開放知識運動[3] [4]」，學術知識為全人類所共有，並非少數研究機構所把持。

2.3 即時檢視受害者資訊與實做

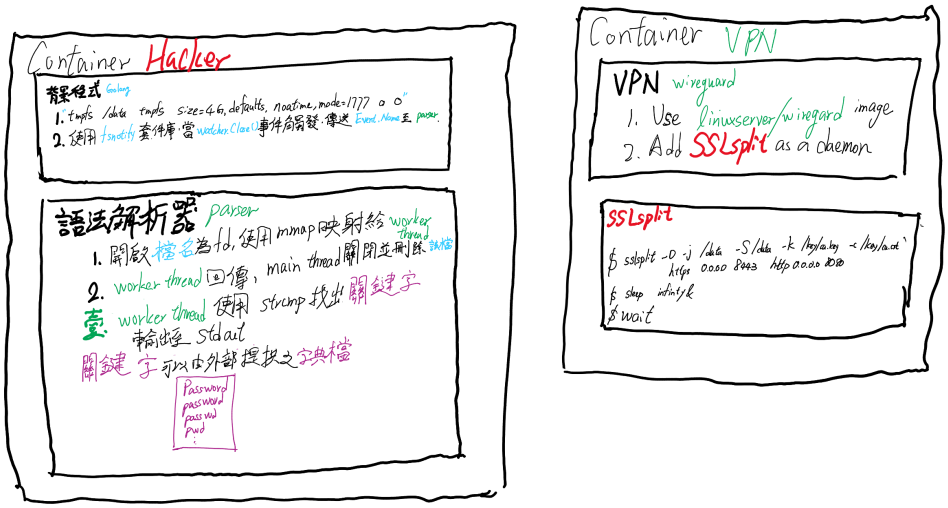
本篇研究報告，提出此機制可以即時檢視受害者資訊，而不用處理事後 pcap 封包紀錄擋。因為通常網路流量都相當大，將所有封包資訊儲存並分析，是不切實際的作法。

sslsplit[5] 攻擊工具會一個輸出目錄，將其經過之流量，解密後輸出至指定目錄檔案。我們可以透過 tmpfs 映射一塊記憶體空間，並且掛載(mount)於給定目標(例如/data)；並且由駭客所撰寫背景程式監聽[6] 此記憶體空間之檔案關閉事件，進而送出該檔案給語法解析器(parser)解析封包內容。

此機制之完整實做並未出現在期末上台報告中，因為正當我們測試本校網路大學時 HTTPS 加密連線，我們應該將上台報告改變為如下3.1 之弱點分析。但是，此機制將在此以圖例與 Python 程式碼呈現。另外我們可以用 docker-compose 保證 Hacker container "create" before VPN container.

[1] https://games.yahoo.com.tw/meme-111554483.html
[2] https://en.wikipedia.org/wiki/Open_access
[3] https://en.wikipedia.org/wiki/Access_to_Knowledge_movement
[4] https://tw.okfn.org/
[5] https://www.roe.ch/SSLsplit
[6] https://man7.org/linux/man-pages/man7/inotify.7.html

Docker-compose 架構圖：



解析 HTTP Response 之Python 範例

```
#! python3
import sys
from io import BytesIO
from urllib3 import HTTPResponse
from http.client import parse_headers

rawresponse = sys.stdin.read().encode("utf8")
redirects = []

while True:
    header, body = rawresponse.split(b"\r\n\r\n", 1)
    if body[:4] == b"HTTP":
        redirects.append(header)
        rawresponse = body
    else:
        break

f = BytesIO(header)
# read one line for HTTP/2 STATUSCODE MESSAGE
requestline = f.readline().split(b" ")
protocol, status = requestline[:2]
headers = parse_headers(f)

resp = HTTPResponse(body, headers=headers)
resp.status = int(status)

if ('text/html' in resp.headers['Content-Type']):
    print("body")
    print(body.decode())
```

2.4 防禦方式

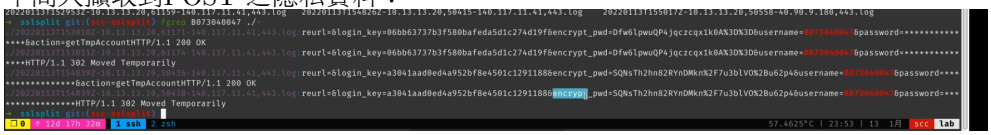
- 檢查憑證
- 憑證固定
- 網路治理

3 成果

受害者登入界面：

受害者成功登入：

中間人擷取到POST 之隱私資料：



3.1 國立中山大學網路大學之弱點分析

從上3 圖可見密碼欄位全為'*' 符號，我們猜測中山網路大學可能有考量到中間人攻擊的可能性，但是仔細觀察 login_key 與 encrypt_pwd 可以發現其中端倪。根據多年駭客經驗，該欄位是base64 編碼(encode)。因此翻閱網路大學 base64 編碼 encrypt_pwd 處可以發現:

```
48      var pwdmask = "**********";
49      /* NSYSU??s begin ?p??K?X?[?K?嗯(?]??????????s?K?X, ???ら|?€??}) */
50      // var md5key  = MD5(node.password.value);
51      // var cypkey  = md5key.substr(0,4) + node.login_key.value.substr(0,4);
52      var cypkey = node.login_key.value.substr(0,8);
53      /* NSYSU??s begin end */
54      node.encrypt_pwd.value = stringToBase64(des(cypkey, node.password.value, 1));
55      node.password.value    = pwdmask.substr(0,node.password.value.length);
56
57      return true;
```

發現加密方式如程式碼，為 DES 加密。此加密為已知弱「對稱式加密」，因此我們可以根據 DES 演算法，推導出此加密反函數，並且擁有該加密金鑰 login_key 鑲嵌於網頁原始碼中：

```
<form method="post" action="/login.php" id="loginForm" name="loginForm" onsubmit="return formSubmit();" autocomplete="off">
  <input type="hidden" name="reurl" value>
  <input type="hidden" name="login_key" value="3041aad0ed4a952bf8e4501c1291188">
  <input type="hidden" name="encrypt_pwd" value>
  <div id="loginBlockContainer">…</div> == $0
</form>
```

因此，此防禦對於 APT 攻擊**完全無效**，當駭客有實際如此觀察過網路大學之實做，完全可以透過 DES 反函數解密出受害者密碼。造成「無謂的加密」問題。

4 結論

本專案之中間人攻擊箸搭配 DNS 污染，於 VPN 端，使用者可以在完全沒看到「網頁不安全」之警示訊息於瀏覽器，但是實際上已經被中間人攻擊。唯一的檢查方式就是打開憑證資訊，確認此憑證是由權威的受信任的數字證書認證機構頒發。

否則，現行網路無法發覺您是否正在被中間人攻擊當中。

4.1 貢獻比例

姓名	貢獻比例
楊志璿	100%
李天朗	0%
王文濤	0%

M m093040094 <m093040094@nsysu.edu.tw> 2:07 AM

To: zxc25077667, TIANLANG LI Show details

Dear 志璿,

有重新進行fork（但不知道還有做什麼用），另外我們那天並沒有到場，因爲確實是沒有做什麼事情（很抱歉因爲我們也不常以這種方式協作，也不知道具體做什麼），不清楚學弟是否在禮拜五有進行報告，助教有寫信來問我們出場的情況，我們回復是"未按預期完成project，按正常給分即可"，很抱歉沒有合理和學弟分工，我知道不太能當作藉口，不過主因是我們另一門主修課程比較重要一些==，如果學弟有自行報告但願您能得到一個不錯的分數。

Regards,
WenTao

這是第10 頁

這是第11 頁

這是第12 頁

書面報告結束