NA HW1 Solution

Wireshark

NANI? Is my uplink as evil as Q-mao? (15 pts)

1. 在wireshark內filter打http，就可以找到你的POST data。
2. https不能直接看到emil & password。
    - (Original) 不一定，因為有可能網路架構是這樣的：

      client <-(HTTPS)-> proxy server <-(HTTP)-> web server

      將https簽章和加密等留給proxy server做，此時即使proxy server和web server是明文的http，在client眼裡看還是https。
    - (Modified) 第一次連線還是可以，因為只要在public key交換的過程中攔截，並傳送middle man自己的public key給對方，這樣就可以達到雙方以為互相建立了祕密管道，其實都是跟middle man建立管道。(之後middle man用自己的private key就可以還原傳送內容。)
3. 例如你去隨便一家咖啡廳連上他們的wifi，中間的內容可能都會被咖啡廳的網路掌管者窺視或竄改。常見的攻擊也可以參見Evil Twin Attack，它是利用同樣的SSID騙取別人連上他的wifi。

In the vast sea, I see no body but flag (10 pts)

Find a packet(放大鏡) -> 以String搜尋 -> 搜尋packet bytes -> 打入flag。這樣你就可以拿到這題的flag。

> flag{_w0w!_Mr._wir3sh4rk!_}

Piepie! The world needs you! (10 pts)

Little Observation

自己試試看網站，再用wireshark或者直接看javascript，就可以知道要把你的答案送出去是靠answer這個POST的key。

而按F12或者wireshark觀察，就可以知道，每次你的回答都會送到server去，再由server告訴你下一個問題是什麼。

而我們又知道http是一種stateless的協定，那它到底怎麼紀錄你到底回答了什麼呢？通常都會是cookie。但是我們這裡偷懶用了javascript的參數來回傳送來達到相同的效果。在仔細觀察一番會發現是用token在互傳的。簡單來說就是

Client	Server
發送 token = ""	收到後， 丟回token A 和question 1
收到token A ，發送token A + answer 1	收到後， 丟回token B 和question 2

- token是hex(AES.CBC)，所以基本上你沒有secret key你也不知道我到底給的token是什麼意思。
- 偷偷告訴你，token最重要的就是有包當下回傳的時間。

No!! piepie!

已知目標是flag，所以用搜尋功能就可以找到有flag的封包。（有複數個，要找到有意義的那個。）

找到這個封包的時候(No. 20472)，我們得知server(src)是140.112.30.26:80(也可以用dns的filter去觀察出ip。)，對應到的client(dst)是192.168.0.17:36912

這個時候我們就可以用filter: `http && tcp.port == 36912` 或者用follow http stream 找到發request的是在哪裡。發現在(No. 19976)這個封包。需要注意的一點是不能單純按照時間順序往前一個一個找，因為TA(出題者/我)有故意在同時間發送很多垃圾封包給server。

觀察No. 19976這個封包就會知道，它會送出answer="A PIEce of PIE" & token = "e03d2e04900..."，因此根據上面的機制，再回去找token = "e03d2e0490...."的封包在哪裡就好。得到(No 18361)，Question是"Piepie or pie?"

- 有個細節是因為是AES CBC，加上iv＆key固定，所以只用一些prefix找到很多很像的封包是正常的。善用copy唷！

一路順著找token，找port的方向，你就可以慢慢的把一整串挖掘出來。以下列出你應該要找到的順序。

No.	Who sends?	Client Port	Clues
20472	Server	36912	假的flag
19976	Client	36912	ans: A PIEce of PIE
18361	Server	36695	question: Piepie or pie?
16952	Client	36695	ans: Pieapple
14593	Server	36424	question: What is your favorite fruit?
10396	Client	36424	ans: To pie or not to pie
5391	Server	36184	question: What is the question?
3667	Client	36184	ans: Don McLean
2561	Server	36016	question: Who is your favorite pie maker?
2128	Client	36016	ans: PIEr than a pied pieman
1099	Server	35876	question: How pie are you?
888	Client	35876	ans: yes

因此對照著問題，重複在網站上打一次，就可以拿到flag了。

> **flag{W0w! Y0u 4r3 !}**

Heresy

- 其實我混淆的封包只會傳第一次。所以你去grep answer ?就可以找出一系列的問題答案，不用這麼麻煩。
- 垃圾回答沒有空格。你可以用regex快速找到所有答案，在按照題意對應也行。

Internet Protocol Stack: 5-layer Model

1. (10 pts)

- Physical Layer: 沒有header(你用wireshark看第一行會發現整個資料都會被反白)。目的在於硬體之間的資料傳送。
- Data Link Layer: 可能會紀錄MAC等資訊。目的在於這個網路下的傳送通道。
- Network Layer: 主要紀錄IP。目的是提供**主機**之間的通道。
- Transport Layer: 會紀錄port/checksum等。目的是提供**程式**之間的連線通道。
- Application Layer: 紀錄的東西因app而異。目的是提供**使用者**服務。

2. (10 pts)

- Yes, 利用RUDP，或是將TCP的功能實做在application layer都是一種方法。