



密码编码学与网络信息安全 期末作业

题目：讨论网络与量子计算、可
信计算、深度学习报告

姓 名： 干皓丞

学 号： 2101212850

院 系： 信息工程学院

专 业： 计算机应用技术

研究方向： 通信及信息安全技术

导 师： 朱跃生 教授

二〇二二年六月

摘要

本作业为密码编码学与网络信息安全的期末个人报告，其内容包含了讨论 Wireshark 网络抓包分析与量子计算信息安全、可信计算技术、深度学习等在密码学现况的学习总结报告，其第一章说明讨论 Wireshark 工具的使用与 HTTP 跟 HTTPS 的分析，同时尝试运用 PHP 跟 XAMPP 的工具进行测试，另外还使用 APACHE 伺服器在 Windows 环境下进行证书的设定，最后对 WiFi 探针进行说明。第二章为量子信息在密码学与资讯安全等影响进行工作总结，而第三章则是可信计算技术与近来深度学习在密码学等工作梳理，最后进行报告的总结。

关键词：Wireshark 网络抓包分析、量子计算信息安全、可信计算技术、深度学习、密码学

目录

第一章 网络与信息流程分析.....	1
1.1 Wireshark 工具安装与说明.....	1
1.2 XAMPP 的 PHP 测试范例与凭证设定	3
1.2.1 XAMPP 说明.....	3
1.2.2 OpenSSL 说明	5
1.2.3 Git 说明.....	5
1.2.4 测试范例	7
1.2.5 PHP 的 Web Server 方案	14
1.2.6 凭证设定	15
1.3 移除凭证.....	16
1.4 进行 HTTP 与 HTTPS 分析	16
1.5 WiFi 探针原理与说明.....	17
1.5.1 何为 WiFi 探针	17
1.5.2 工作原理	18
1.5.3 深入了解	19
1.5.4 WiFi 探针工作	21
1.5.5 WIFI 探针能采集到的数据	22
1.5.6 数据释义	22
1.5.7 安全性.....	23
1.5.8 用何种设备.....	24
第二章 量子计算机与信息安全对密码学的影响.....	35
第三章 可信计算技术与近来深度学习对密码学的影响.....	37
第四章 工作总结	39
参考文献	41
致谢	43

主要符号对照表

x, y, m, n, t	标量, 通常为变量
K, L, D, M, N, T	标量, 通常为超参数
$x \in \mathbb{R}^D$	D 维列向量
(x_1, \dots, x_D)	D 维行向量
$(x_1, \dots, x_D)^T$ or $(x_1; \dots; x_D)^T$	D 维行向量
$x \in \mathbb{R}^{KD}$	(KD) 维的向量
\mathbb{M}_i or $\mathbb{M}_i(x)$	第 i 列为 $\mathbf{1}$ (或者 x), 其余为 $\mathbf{0}$ 的矩阵
$diag(\mathbf{x})$	对角矩阵, 其对角元素为 x
I_N or I	($N \times N$) 的单位阵
$A \in \mathbb{R}^{D_1 \times D_2 \times \dots \times D_K}$	大小为 $D_1 \times D_2 \times \dots \times D_K$ 的张量
$\{x^{(n)}\}_{n=1}^N$	集合
$\{(x^{(n)}, y^{(n)})\}_{n=1}^N$	数据集
$\mathcal{N}(x; \mu, \Sigma)$	变量 x 服从均值为 μ , 方差为 Σ 的高斯分布

① 本符号对照表内容选自邱锡鹏老师的《神经网络与深度学习》^[1]一书。

第一章 网络与信息流程分析

本章針對在深圳研究院的 WLAN 中，从请求接入开始，分别 HTTP 应用以及一种 HTTPS，[E]使用 Wireshark 网络抓包工具，进行抓包，并对抓取的信息流程进行分析，试论 Wi-Fi 探针的原理与应用。而本作業在過程中所面對的問題如下五個章節分[E]進行[E]明，首先講述在 Windows 平台上進行 Wireshark 工具安装，其二使用 XAMPP 框架使用 PHP 跟 APACHE 進行測試，其三是在 Mac 平台上使用 Wireshark 工具分析 HTTP 与 HTTPS 的網頁應用，最後在[E]明 WiFi 探针。

1.1 Wireshark 工具安装与说明

本作业此节说明 Wireshark 工具安装与使用，该节使用的平台为 Windows，在官网下载安装指定的 Windows 方案，其版本是 Wireshark 3.6.3 64 位元，此外在安装的程序中额外选择了 Npcap 1.55 和 USBPcap 1.5.4.0。安装后可看到有波动的网路流量，则是可以分析的目标。进入后则是可以看到整体个流量的状况，同时使用 Windows 平台下的命令提示字元去 Ping GitHub Page，来查看封包状况。

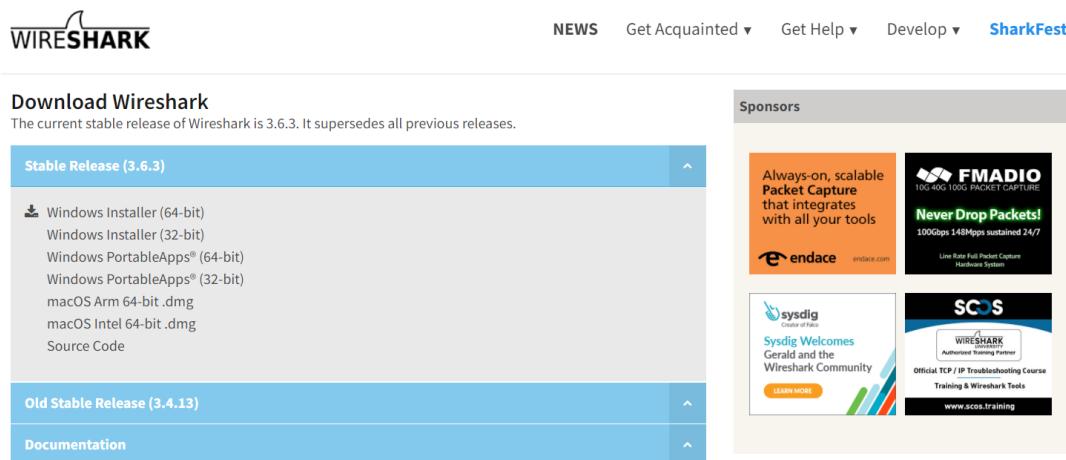


图 1.1 Wireshark 官方网站

Wireshark 最早名为 Ethereal，其溯源于 1997 年底的一位密苏里大学堪萨斯分校毕业生杰拉德·康姆斯 (Gerald Combs) 在一家小型的网际网路服务供应商上班，他因工作需求，迫切需要一个能够追踪网路流量的工具软体辅助其工作故而开始撰写。在经过几次中断开发的事件过后，于在 1998 年 7 月释出其第一个版本。此后康姆斯收到了来自全世界的修补程式、错误回报与鼓励信件。而 Ethereal 的发展就此而始。不久之后，Gilbert Ramirez 看到了这套软体的开发潜力并开始参予低阶程式的开发。1998 年 10 月，

来自 Network Appliance 公司的 Guy Harris 在寻找一套也是网路封包撷取 tcpview 更好的工具，于是他也开始参与 Ethereal 的开发工作。1998 年底，一位在教授 TCP/IP 课程的讲师 Richard Sharpe，看到了这套软体的发展潜力，而后开始参与开发与加入新协定的功能。在当时，新的通讯协定的制定并不复杂，因此他开始在 Ethereal 上新增的封包撷取功能，几乎包含了当时所有通讯协定。自此之后，数以千计的人开始参与 Ethereal 的开发，多半是因为希望能让 Ethereal 撷取特定的、尚未包含在 Ethereal 预设的网路协定的封包而参予新的开发，最后 2006 年因商标的问题，Ethereal 更名为 Wireshark。

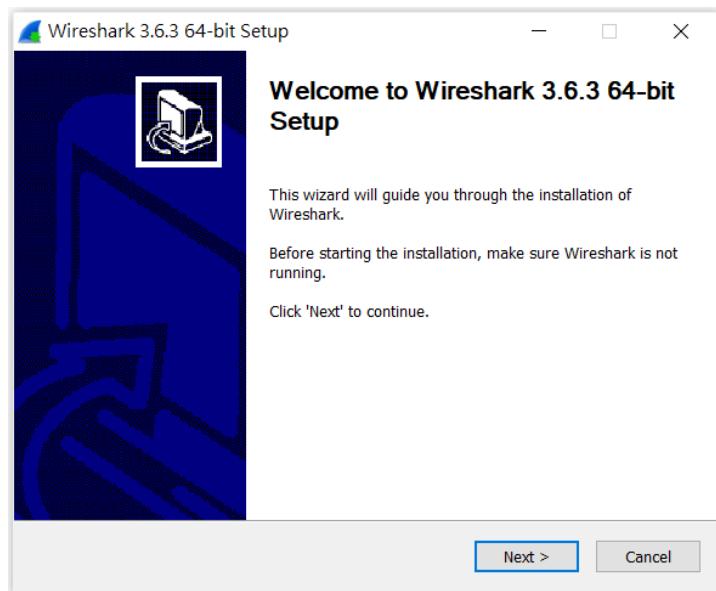


图 1.2 Wireshark Windows 应用程式安装流程

Npcap 是 Nmap 項目的用於 Microsoft Windows 的數據包捕獲和發送庫，其本身使用定制的 Windows 核驅動程序以及 Windows 構建的優秀 libpcap 庫來實現開放的 Pcap API。這允許 Windows 軟件使用簡單、可移植的 API 捕獲包括無線網絡、有线以太網、本地主機流量和許多 VPN 等原始網絡流量。Npcap 也允許發送原始數據包，因為 Mac 和 Linux 系統已經包含 Pcap API，因此 Npcap 允許流行的軟件，例如 Nmap 和 Wireshark 使用單個代碼庫在所有這些平台上運行。其 Npcap 最早始於 2013 年，作對現已停止的 WinPcap 庫的一些改進，但與之差別的是從那時起，前者已在很大程度上被重寫，經過數百個版本提高了 Npcap 的速度、可移植性、安全性和效率。

USBPcap 是可以让 Wireshark 来获得 Windows 的 USB 数据包，前者可以让后者能够对来自 USB 的封包进行分析，同样地该工具也是开源许可。

在本节可以看到 Windows 平台的 Wireshark 顺利的进行安装，同时对 GitHub Page 进行分析，所谓的 GitHub Page 是开放原始码的版本控制平台 GitHub，其所提供的静态页面服务，多用于该开源项目或者使用者进行项目的展示与说明之用。从分析过

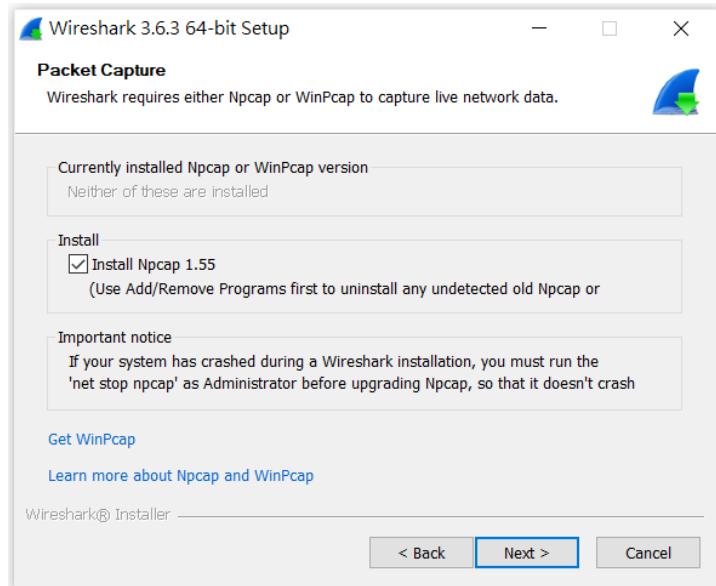


图 1.3 Install Npcap 1.55

程中可以看到 Wireshark 顺利的侦测到 GitHub Page 的状态，后面本作业会尝试说明 OpenSSL 与 CA 的设定。

```
> ping kancheng.github.io
```

1.2 XAMPP 的 PHP 测试范例与凭证设定

此节本作业除了说明 OpenSSL 等概念，也会尝试使用 XAMPP 与 PHP 分别进行凭证设定与范例，此节本作业会写程式范例同时也会提供 CA 设定档，根据写好的地区设定，在 Apache 伺服器中执行 CA 认证。

1.2.1 XAMPP 说明

XAMPP 是一个可以简单整合网页伺服器 Apache、伺服器端语言 PHP、程式语言 Perl 及资料库 MariaDB 的软体包，只要透过 XAMPP 就可以方便开发者快速架站，多数使用者在架站时需要透过网页伺服器让访客能够连到所开发的网站上，目前比较普遍的网页伺服器有简称 Apache 的 Apache HTTP 伺服器、Nginx、以及简称为 IIS 的 Microsoft Internet Information Server 等，基本上透过网页伺服器就能将网页包含图像、影音等各种档案提供给请求的使用者。

PHP 最初是由勒多夫在 1995 年开始开发的，现在 PHP 的标准由 the PHP Group 维护，该语言是一种开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入 HTML 中使用，其语法借鉴吸收 C 语言、Java 和 Perl 等流行计算机语言的特点，易于一般程

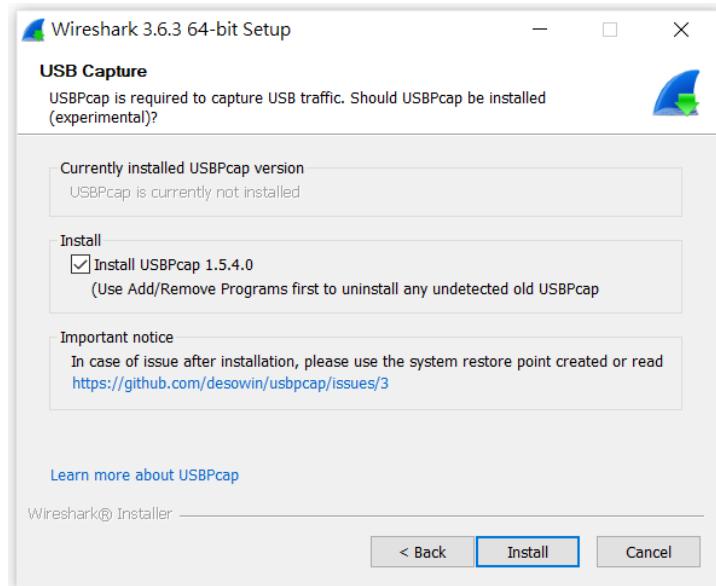


图 1.4 Install USBPcap 1.5.4.0

程序员学习。而 PHP 的主要目标是允许网络开发人员快速编写动态页面，但 PHP 也被用于其他很多领域。

另外当中的 MySQL 则是在 1995 年，Michael Widenius, David Axmark 及 Allan Larsson 创立了瑞典的 MySQL AB 公司，随之推出了现今最具知名度的同名产品 MySQL，作为关联式资料库的管理所使用。所以 MySQL 从字面上来理解就是一种关联式资料库管理系统 (Relational Database Management System; RDBMS)，同时也因为 MySQL 的问世，让 MySQL AB 曾经成为过去全球最大的开放源码公司。MySQL AB 运用双重许可，一方面 MySQL 属于 GPL 的开放协议，让软体在 GPL 的规范下可以无偿使用，但这样的规范对于部分公司可能不敷使用，例如必须使用到没有被开源的程式码或技术，那就只能靠付费的方式来获得。另一方面，MySQL AB 也靠着顾问服务以及认证的方式赚取收入，例如透过开班授课、培训的方式，来取的 MySQL 的 Certificate 等方式获取利润；通过就是卖服务，MySQL 可能不需要付费，但是如果要原厂支援的话则需要付费。而在 2008 年，升阳软体公司 (Sun Microsystems) 透过 10 亿美金的价格收购，而在 2009 年升阳也被甲骨文公司 (Oracle) 收购，于是乎 MySQL 成为 Oracle 旗下产品。但收购后 Oracle 大幅调涨其商业版的售价，让许多开源人士不再看好，担心有一天开源社群版最后就会被商业版取代，于是 Michael Widenius 以 MySQL 为基础，成立分支计划 MariaDB。而一些使用 MySQL 的开源专案逐渐也转向 MariaDB，例如维基百科就在 2013 年正式转换。不过因为 MariaDB 就是直接用 Fork 分流的方式从 MySQL 原始码开始开发，所以原使用者要转换到 MariaDB 上并不太困难，且外部连线函式库都可以共用。

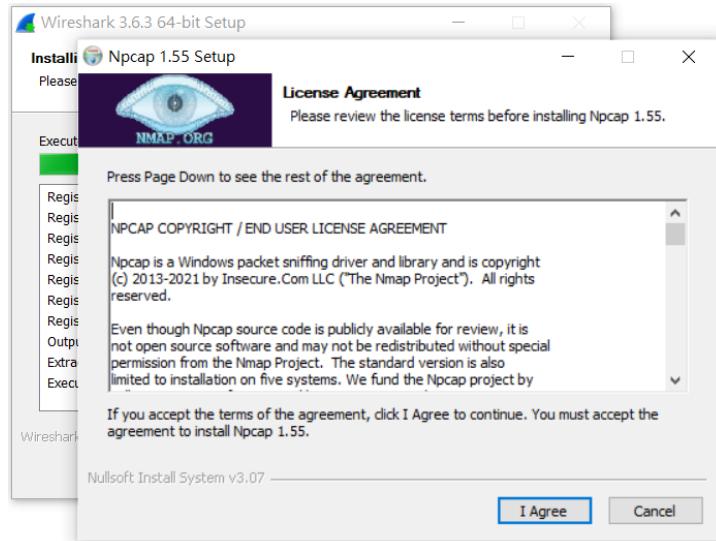


图 1.5 Npcap 1.55 流程

1.2.2 OpenSSL 说明

再来说说明 OpenSSL，在计算机网路上 OpenSSL 是一个开放原始码的软体函式库套件，计划于 1998 年开始，该项目的目标是发明一套自由的加密工具，使之在网际网路上使用。其 OpenSSL 以 Eric Young 以及 Tim Hudson 两人所开发的 SSLeay 为基础，但随着两人前往 RSA 公司任职后，SSLeay 与 1998 年 12 月停止开发，故因此在 1998 年 12 月，社群另外分支出 OpenSSL 继续开发。目前 OpenSSL 应用程式可以使用这个套件来进行安全通讯，来避免窃听，同时确认另一端连线者的身分，此套件广泛被应用在网际网路的网页伺服器上。另外该应用的主要函式库皆是以 C 语言所写成，实作了基本的加密功能，实作了 SSL 与 TLS 协定，此外 OpenSSL 可以运行在 OpenVMS、Microsoft Windows 以及绝大多数如 Solaris, Linux, Mac OS X 与各种版本的开放原始码 BSD 作业系统类 Unix 作业系统上。虽然此软体是开放原始码的，但其授权书条款与 GPL 有冲突之处，故如 Wget 等 GPL 软体使用 OpenSSL 时必须对 OpenSSL 给予例外。

1.2.3 Git 说明

Windows 的 OpenSSL 其本作业在此使用 Windows 平台版本中的 Git 内建的 OpenSSL，另外要说明的是 Git 是一个分散式版本控制软件，最初由 Linux 开发者林纳斯·托瓦兹创作，并于 2005 年以 GPL 授权条款释出，其最初目的是为了更好地管理 Linux 核心开发而设计。同时应注意的是，这与 GNU Interactive Tools 等类似 Norton Commander 界面的文件管理器有着根本上不同。Git 最初的开发动力来自于 BitKeeper 和 Monotone，其最初只是一个可以被其他如 Cogito 或 Stgit 前端包装的后端而开发，但后来 Git

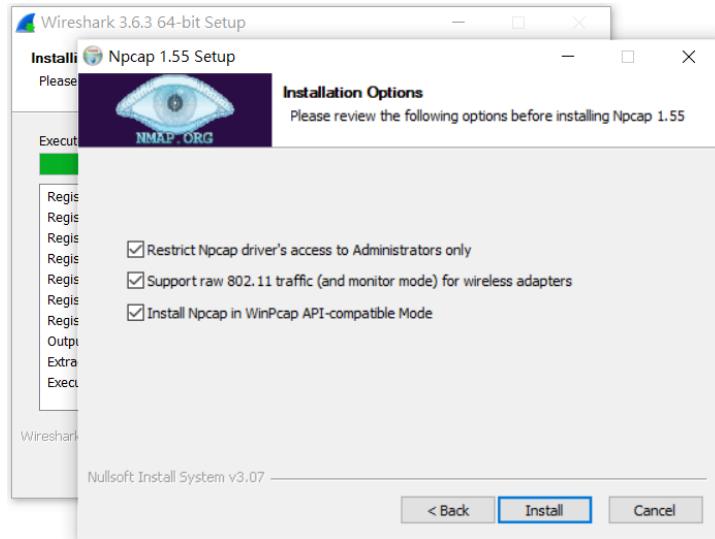


图 1.6 Npcap 1.55 设定

内核已经成熟到可以独立地用作版本控制，并被很多软体专案广泛使用其中包括 Linux 核心。

1.2.3.1 OpenSSL 的产生流程与指令结果

在此可以看到 OpenSSL 的产生流程，但本次作业考量到便捷性故考虑使用在 Windows 的 Git 所提供的命令介面。

1. 指令

```
openssl genrsa -out privkey.pem 4096
openssl ssh-keygen -t rsa -b 4096 -f privkey.pem
```

2. 结果

```
USER@Aspire-R7 MINGW64 /d/git-project/test
\$ openssl genrsa -out privkey.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)

USER@Aspire-R7 MINGW64 /d/git-project/test
\$ openssl req -new -x509 -key privkey.pem -out cacert.pem -days 1095
You are about to be asked to enter information that will be incorporated
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

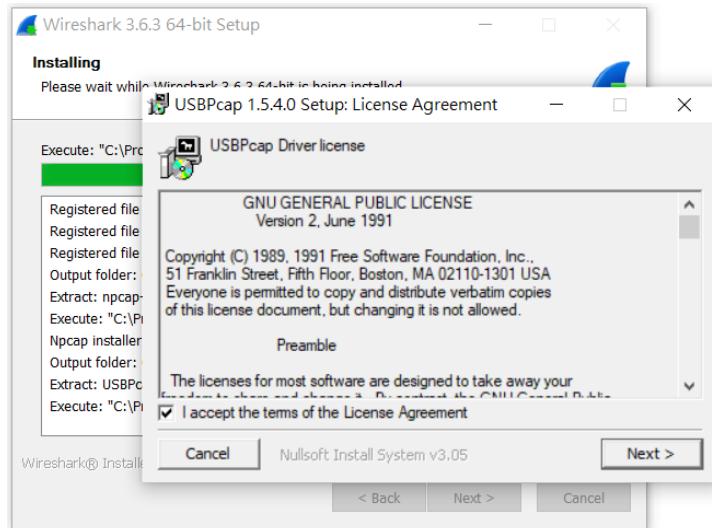


图 1.7 USBPcap 1.5.4.0 流程

```

For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:CN
Locality Name (eg, city) []:Shenzhen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PKU
Organizational Unit Name (eg, section) []:CS
Common Name (e.g. server FQDN or YOUR name) []:KAN
Email Address []:zz@pku.com

USER@Aspire-R7 MINGW64 /d/git-project/test

```

1.2.4 测试范例

在此本作业为此写了 PHP 测试范例，同时也为了 APACHE 准备了设定 CA 凭证的设定档案，当中设定档案的根据皆为 XAMPP 的 APACHE 目录下。在其原目录建立 crt 目录，并放入 cert.conf 与 make-cert.bat 档案，当中 BAT 档案为 Windows 的命令档案，后者会根据前者的设定产生 CA 凭证。最后会在其 crt 目录下看到所产生的 localhost 目录。凭证设定完成后要修改其 apache _ conf _ extra _ httpd-xampp.conf 档案进行设定，最后重启执行。从该目录可以看到产生的 CA 档案。另外 PHP 与 JS 档案则是根据 XAMPP 的预设目录，将其 PHP 与 HTML 档案放于 htdocs 目录下。

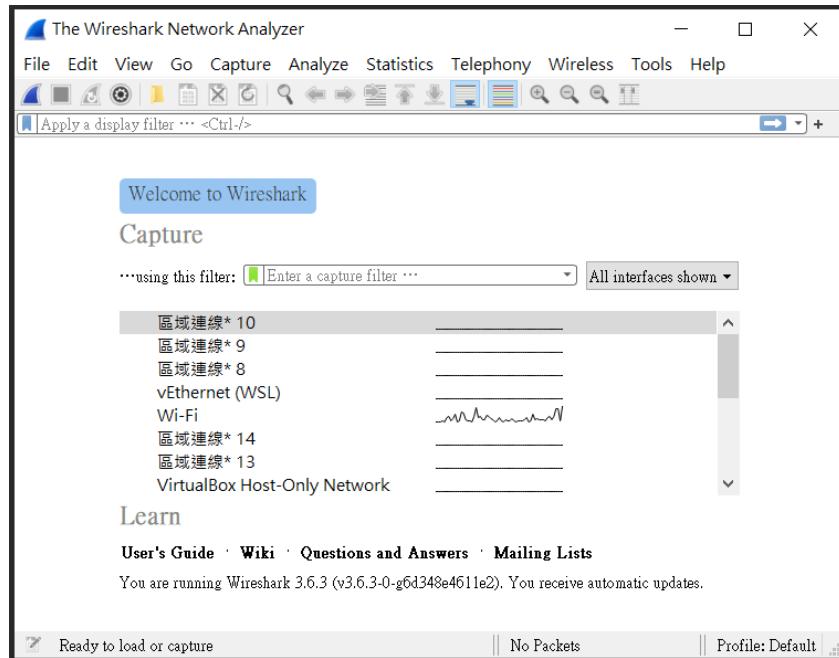


图 1.8 Wireshark 应用程式画面

1.2.4.1 cert.conf

为根据 XAMPP 的 APACHE 与台湾地区所考量的 CA 设定档案。与 BAT 命令档案同放于 APACHE。

```
[ req ]
default_bits      = 2048
default_keyfile   = server-key.pem
distinguished_name = subject
req_extensions    = req_ext
x509_extensions   = x509_ext
```

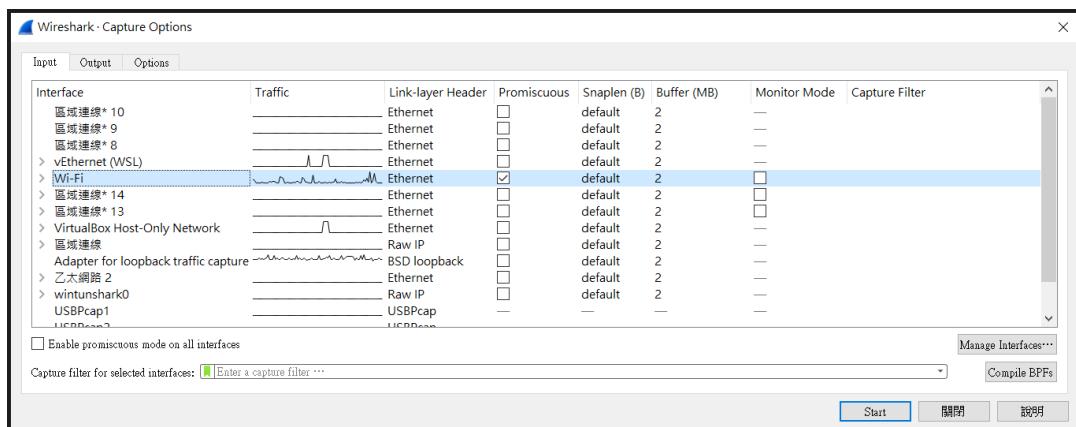


图 1.9 Wireshark 封包选项

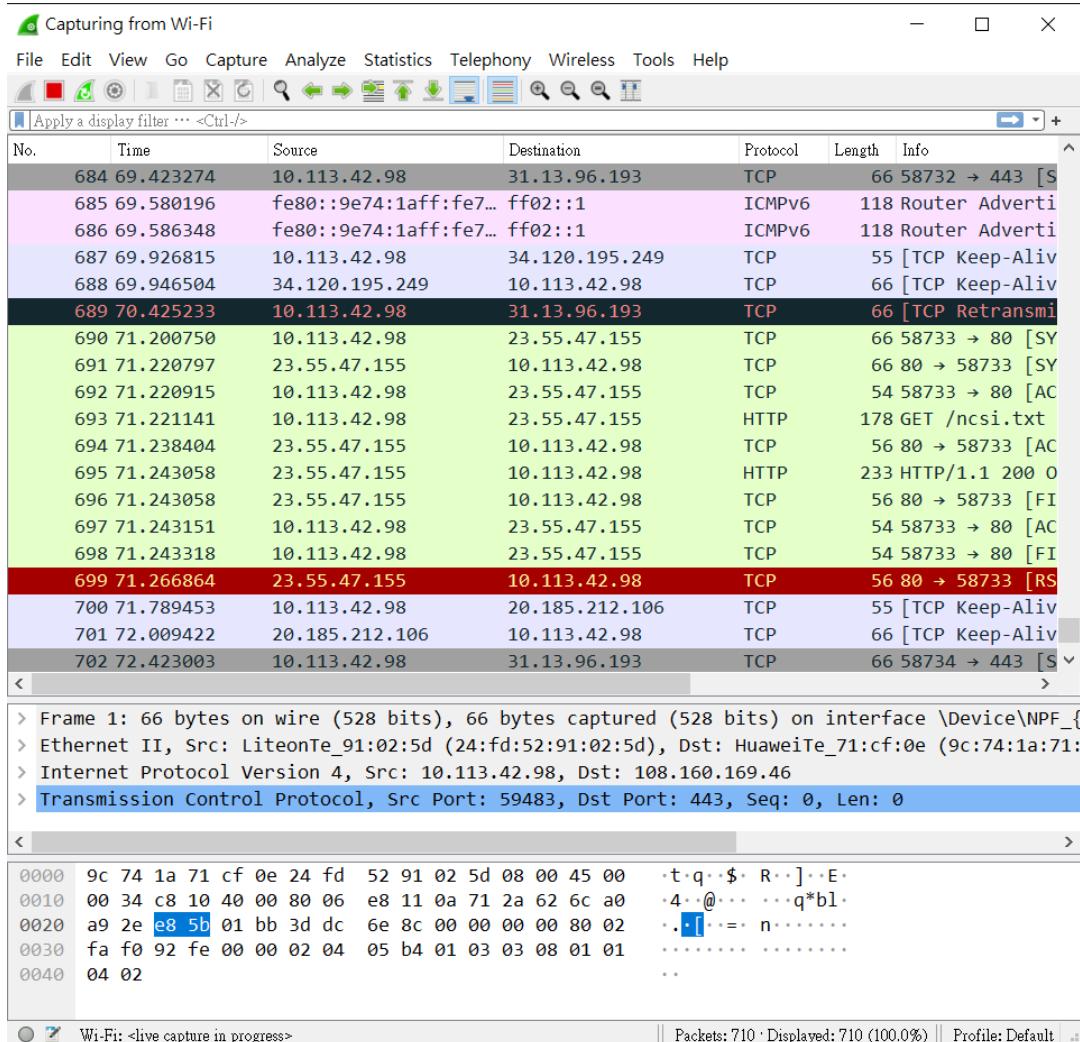


图 1.10 WiFi 追踪

```

string_mask          = utf8only

[ subject ]

countryName          = Country Name (2 letter code)
countryName_default  = TW

stateOrProvinceName  = State or Province Name (full name)
stateOrProvinceName_default = Taiwan

localityName         = Locality Name (eg, city)
localityName_default = Taipei

organizationName      = Organization Name (eg, company)
organizationName_default = Personal Reserach
    
```

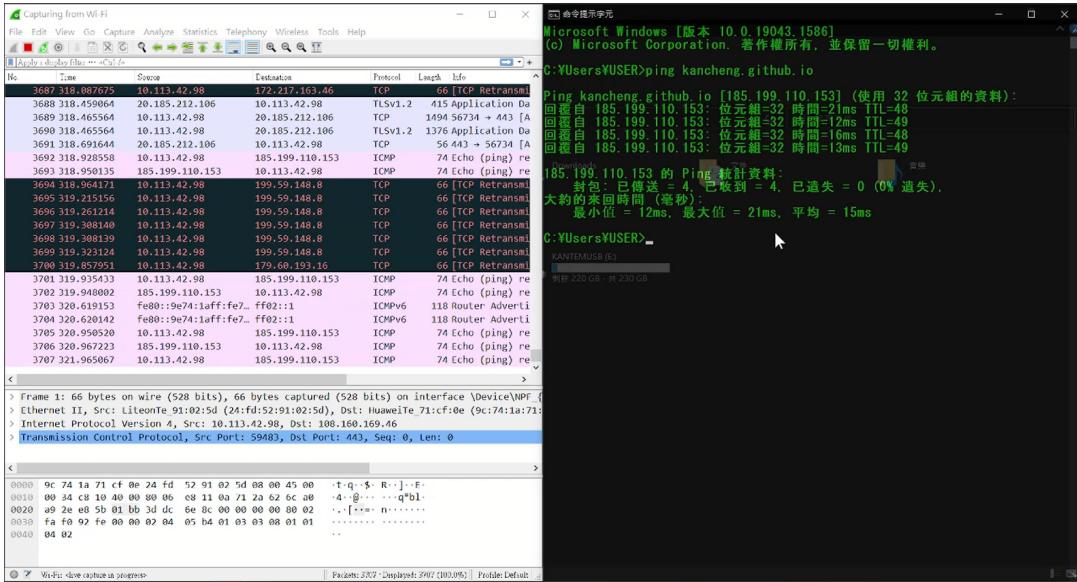


图 1.11 使用 GitHub Page 测试

```

commonName          = Common Name (e.g. server FQDN or YOUR name)
commonName_default = localhost

emailAddress        = Email Address
emailAddress_default = test@example.com

[ x509_ext ]

subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer

basicConstraints     = CA:FALSE
keyUsage              = digitalSignature, keyEncipherment
subjectAltName        = @alternate_names
nsComment             = "OpenSSL Generated Certificate"

[ req_ext ]

subjectKeyIdentifier = hash

basicConstraints     = CA:FALSE
keyUsage              = digitalSignature, keyEncipherment
subjectAltName        = @alternate_names
nsComment             = "OpenSSL Generated Certificate"

[ alternate_names ]

```

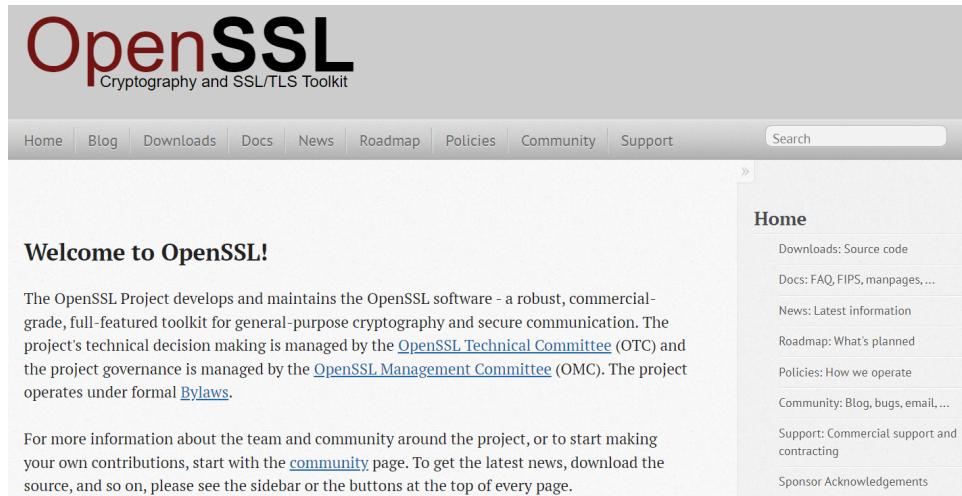


图 1.12 OpenSSL 官方页面

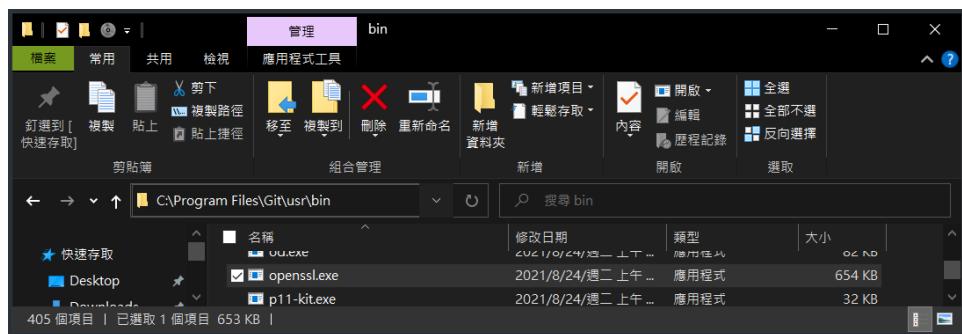


图 1.13 Git 的 Windows 版本内建 OpenSSL

DNS.1 = localhost

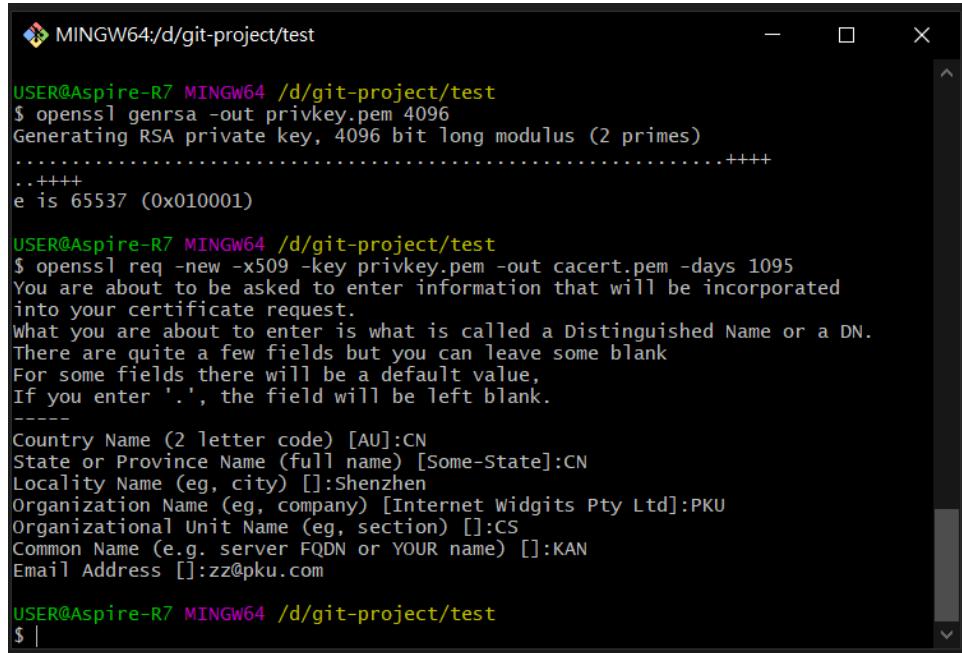
1.2.4.2 make-cert.bat

为对应 CONF 的设定档案所设定的命令档案，在 Windows 平台上使用 PowerShell 进行执行，最后产生 CA 档案。

```
@echo off
::set /p domain="Enter Domain: "
set domain="localhost"
set OPENSSL_CONF=../conf/openssl.cnf

if not exist .%\domain% mkdir .%\domain%

..\bin\openssl req -config cert.conf -new -sha256 -newkey rsa:2048 -nodes -keyout\
%\domain%\server.key -x509 -days 3650 -out %domain%\server.crt
```



A screenshot of a terminal window titled "MINGW64:/d/git-project/test". The terminal is executing an OpenSSL command to generate an RSA private key and a certificate request. The output shows the generation of a 4096-bit modulus with two primes, the selection of an encryption exponent (e) as 65537 (0x010001), and the creation of a certificate request with fields for country, state/province, locality, organization, organizational unit, common name, and email address. The user is prompted to enter information for these fields, with some being left blank or having default values.

```
USER@Aspire-R7 MINGW64 /d/git-project/test
$ openssl genrsa -out privkey.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)

USER@Aspire-R7 MINGW64 /d/git-project/test
$ openssl req -new -x509 -key privkey.pem -out cacert.pem -days 1095
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:CN
Locality Name (eg, city) []:Shenzhen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PKU
Organizational Unit Name (eg, section) []:CS
Common Name (e.g. server FQDN or YOUR name) []:KAN
Email Address []:zz@pku.com

USER@Aspire-R7 MINGW64 /d/git-project/test
$ |
```

图 1.14 OpenSSL 执行

```
echo.
echo -----
echo The certificate was provided.
echo.
```

1.2.4.3 http-s-index.php

为测试 HTTP 和 HTTPS 所特地准备的档案，当中的 PHP 根据路径与资讯进行判断。

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>TEST - HTTP & HTTPS</title>
    <style type="text/css">
        body {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>TEST - HTTP & HTTPS</h1>
    <div>
```

```

<?php
    if (!empty($_SERVER['HTTPS']) && ('on' == $_SERVER['HTTPS'])) {
        $uri = 'https://';
        echo " 目前是 HTTPS";
    } else {
        $uri = 'http://';
        echo " 目前是 HTTP";
    }
    $uri .= $_SERVER['HTTP_HOST'];
    phpinfo();
?>
</div>
</body>
</html>

```

1.2.4.4 http-s-index.html

为测试 HTTP 和 HTTPS 所特地准备的档案，当中的 JS 根据路径与资讯进行判断。

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>TEST - HTTP & HTTPS</title>
<style type="text/css">
    body {
        text-align: center;
    }
</style>
</head>
<body>
    <h1>TEST - HTTP & HTTPS</h1>
    <div>
        <span id="show"></span>
    </div>
</body>
<script type="text/javascript">
    var ishttps = 'https:' == document.location.protocol ? true: false;
    if(ishttps){
        // alert(" 这是一个 HTTPS 请求");
        document.getElementById("show").textContent=" 这是一个 HTTPS 请求";
    }else{
        // alert(" 这是一个 HTTP 请求");
    }
</script>

```

```
document.getElementById("show").textContent=" 这是一个 HTTP 请求";
}
</script>
</html>
```

1.2.4.5 httpd-xampp.conf

安装完凭证后，需要在 APACHE 对 httpd-xampp.conf 进行设定，最后重启 APACHE。

```
## localhost
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost

    ServerAlias *.localhost

    RewriteEngine On
    RewriteCond \ %{HTTPS} off
    RewriteRule (.*) https://\ %{SERVER_NAME}/\$1 [R,L]
</VirtualHost>
<VirtualHost *:443>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost

    ServerAlias *.localhost
    SSLEngine on
    SSLCertificateFile "crt/localhost/server.crt"
    SSLCertificateKeyFile "crt/localhost/server.key"
</VirtualHost>
```

1.2.5 PHP 的 Web Server 方案

此外本作业针对 XAMPP，同时也准备了备案，其 PHP 内建有 Web Server 的方案，其作用是提供使用者一个方便的测试用途。本作业根据此作为测试的备案。从 PHP 的指令中可以看到，其参数为大写的 S，最后加上 IP 位址与 Port 号。

1. 指令

```
> php -S 127.0.0.1:9988
```

2. PHP 帮助

```
Usage: php [options] [-f] <file> [--] [args...]
php [options] -r <code> [--] [args...]
```

```

php [options] [-B <begin_code>] -R <code> [-E <end_code>] [--] [args...]
php [options] [-B <begin_code>] -F <file> [-E <end_code>] [--] [args...]
php [options] -S <addr>:<port> [-t docroot] [router]
php [options] -- [args...]
php [options] -a

-a           Run as interactive shell
-c <path>|<file> Look for php.ini file in this directory
-n           No configuration (ini) files will be used
-d foo[=bar] Define INI entry foo with value 'bar'
-e           Generate extended information for debugger/profiler
-f <file>   Parse and execute <file>.
-h           This help
-i           PHP information
-l           Syntax check only (lint)
-m           Show compiled in modules
-r <code>   Run PHP <code> without using script tags <?..?>
-B <begin_code> Run PHP <begin_code> before processing input lines
-R <code>   Run PHP <code> for every input line
-F <file>   Parse and execute <file> for every input line
-E <end_code> Run PHP <end_code> after processing all input lines
-H           Hide any passed arguments from external tools.
-S <addr>:<port> Run with built-in web server.
-t <docroot> Specify document root <docroot> for built-in web server.
-s           Output HTML syntax highlighted source.
-v           Version number
-w           Output source with stripped comments and whitespace.
-z <file>   Load Zend extension <file>.

args...      Arguments passed to script. Use -- args when first argument
            starts with - or script is read from stdin

--ini        Show configuration file names

--rf <name>  Show information about function <name>.
--rc <name>  Show information about class <name>.
--re <name>  Show information about extension <name>.
--rz <name>  Show information about Zend extension <name>.
--ri <name>  Show configuration for extension <name>.

```

1.2.6 凭证设定

根据本作业前几个小节，可以知道其事前工作的准备。同时可以看到此小节执行 BAT 与 APACHE 产生凭证后，对其进行安装，最后于 Windows 平台检视其结果。同时

也看到测试范例显示的资讯。

```
(base) PS C:\xampp\apache\crt> .\make-cert.bat
Generating a RSA private key
-----+
writing new private key to 'localhost\server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [TW]: Taiwan
State or Province Name (full name) [Taiwan]:
Locality Name (eg, city) [Taipei]:
Organization Name (eg, company) [Personal Reserach]:
Common Name (e.g. server FQDN or YOUR name) [localhost]:
Email Address [test@example.com]

-----
The certificate was provided.
```

图 1.15 执行 BAT 范例

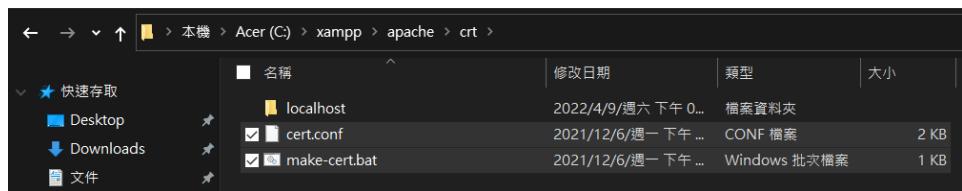


图 1.16 APACHE 结果

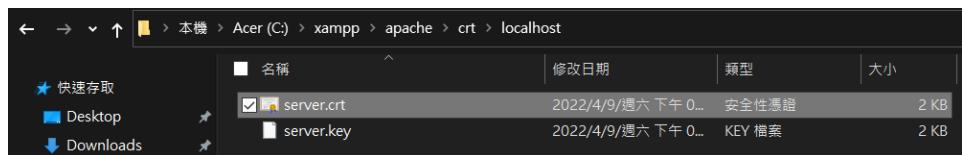


图 1.17 证书

1.3 移除凭证

本节接续前一小节说明在 Windows 平台安装后的凭证移除，首先找 MMC，进入主控台进行移除的设定，同时将凭证选项加入管理，同时设定帐户的权限范围，最后对成功的证书进行移除，移除成功后根据上一节的状况，要将 APACHE 的设定移掉，最后重启 APACHE。

1.4 进行 HTTP 与 HTTPS 分析

本作业此节使用 Mac 的 Wireshark，对 HTTPS 与 HTTP 两个现有的网路服务进行分析，同时使用 Mac 平台的 Wireshark 工具进行判断与分析，从结果来说，很明显得可以从各自的 TCP 串流中可以看到安全性问题。

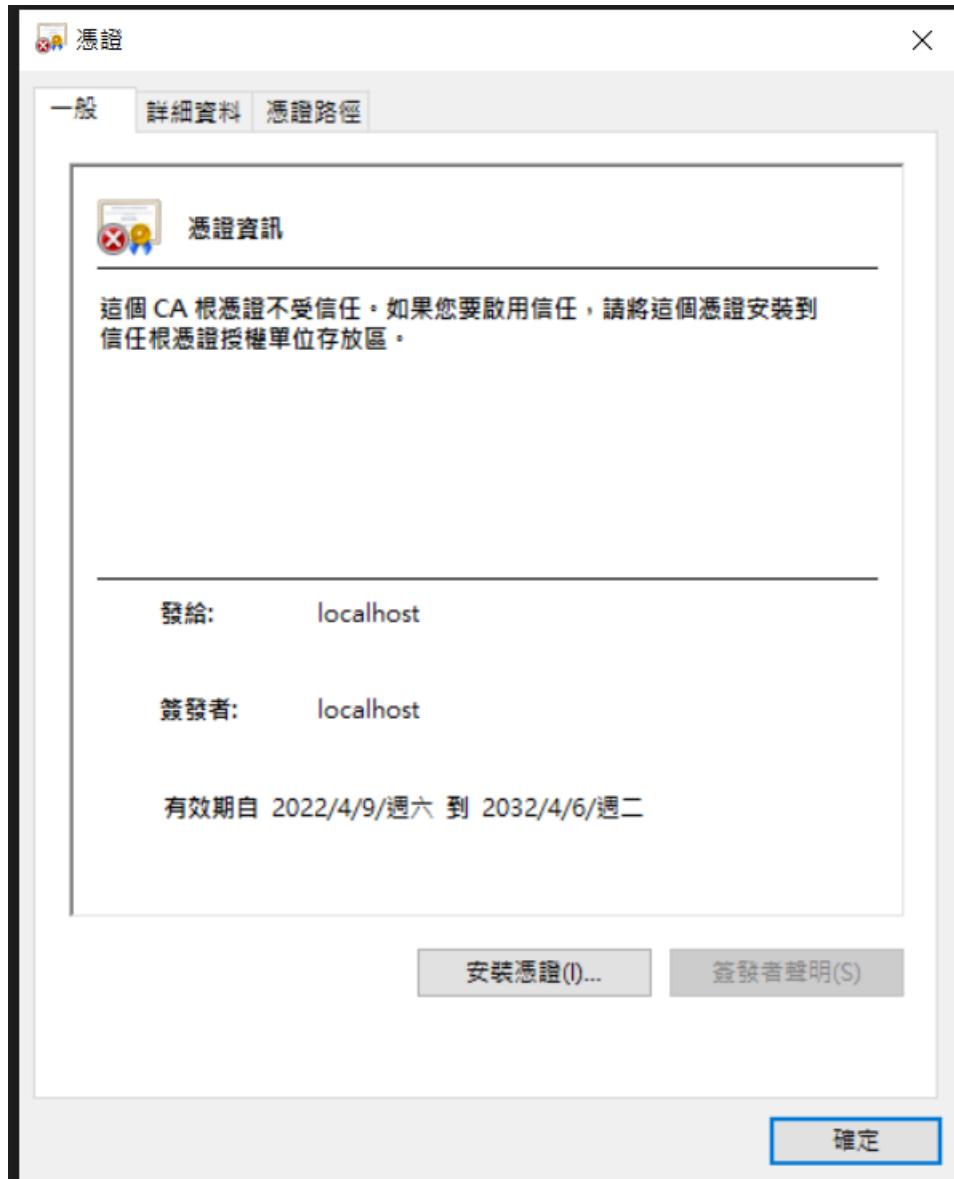


图 1.18 安装证书

1.5 WiFi 探针原理与说明

1.5.1 何为 WiFi 探针

所谓的 WiFi 探针技术是根据 WiFi 探测技术来识别用途名为 AP 的无线访问接入点，附近已开启 WiFi 的智能手机或者类似于笔记本、平板电脑等 WiFi 终端，其无需使用者接入 WiFi，WiFi 探针就能够识别使用者的信息。而当使用者走进探针信号覆盖区域内且使用者的 WiFi 设备开启时，其的设备就能被探针探测并发现，所以无论是 Apple 的 IOS 或者是 Google 安卓系统都能轻易检测，并且获取设备的 MAC 地址与相关资讯。其特点如下所示：

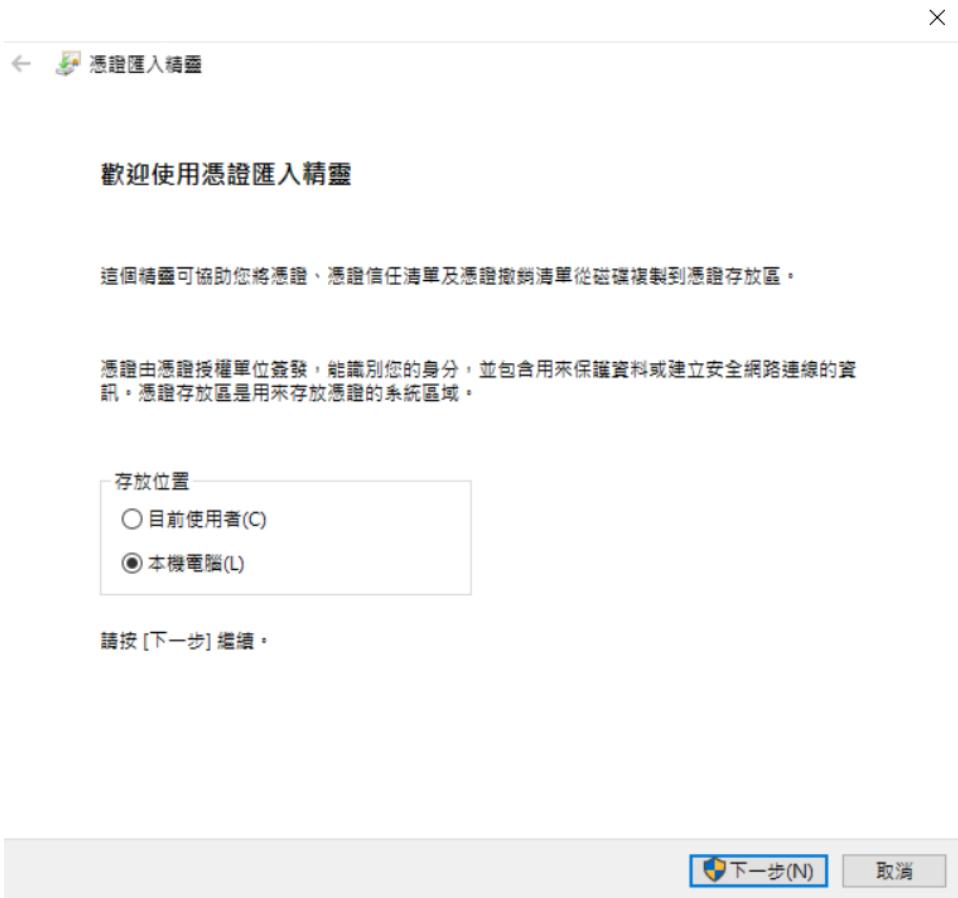


图 1.19 汇入流程

- 用户无需参与，无需连接到网络
- Android, IOS 全兼容
- 自动探测区域内手机 MAC 地址
- 手机，平板均能探测

1.5.2 工作原理

WiFi 是基于 IEEE 802.11a/b/g/n 协议而成，在标准协议中，定义了名为 AP 的无线接入点和名为站或客户端的 STA 的两种工作模式，同时协议中规定了 BEACON、ACK、DATA、PROBE 等多种无线数据帧类型，在站连接到无线接入点时进行交互的就是数据帧和应答帧、同时 AP 周期性发送 BEACON。在站点没有连接到无线接入点上，手机客户端等站点也会发送 PROBE 帧进行探测询问哪个 AP 是可以连接使用。其 WiFi 探针就是基于各种无线数据帧来抓获手机等 WIFI 客户端的 MAC 地址信息。每个 AP 每隔几十毫秒到几秒不等一定时间会向周围的 STA 和 AP 广播 BEACON 帧，就是告诉周围的 STA 和其他的 AP，自己是 xxxx 的 bssid，发出快来连接我的请求，同时每个

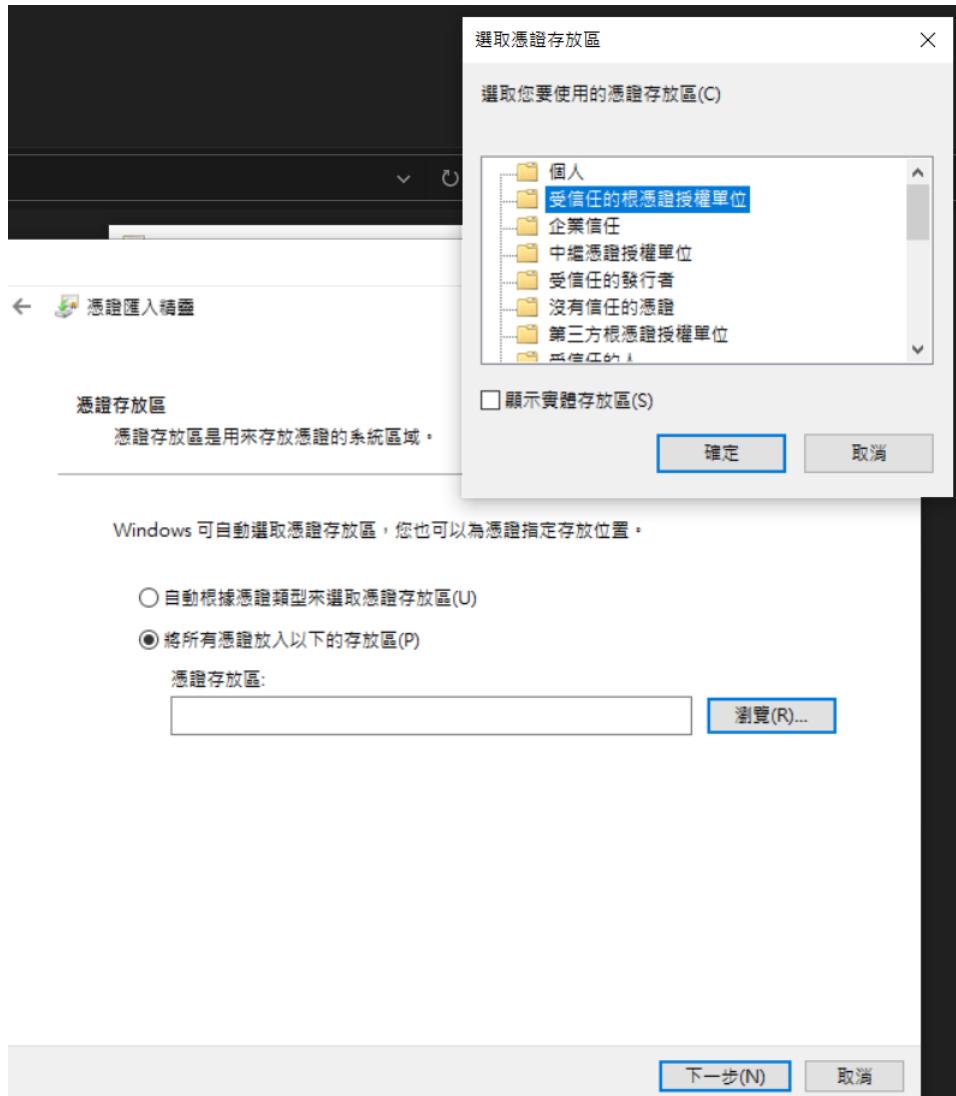


图 1.20 受信任的跟凭证授权单位

STA，如手机、笔记本等除了默默监听周边 AP 发送的 BEACON 帧以外，还会偷偷发送 PROBE 帧，类似于我是 xxxx 的 MAC 地址，我能够连结的请求。

1.5.3 深入了解

要深入了解 WiFi 探针技术，首先先认识 WiFi 所使用的网络协议，WiFi 采用的是 IEEE802.11 协议集，此协议集包含许多子协议。其中按照时间顺序发展，主要有：(1) 802.11a,(2) 802.11b,(3) 802.11g,(4) 802.11n。同时在网络通信中，数据被封装成了帧，而帧就是指通信中的一个数据块。但是帧在数据链路层传输的时候是有固定格式的，不是随意的封装和打包就可以传输，大小有限制，最小 46 字节，最大 1500 字节，所以必须按照此规则来封装。下面为 802.11 的帧结构，且从上面的结构可以知道，前两个字节为：帧控制字段，控制字段的前 2 BIT 节为协议类型，且目前此值为：0。

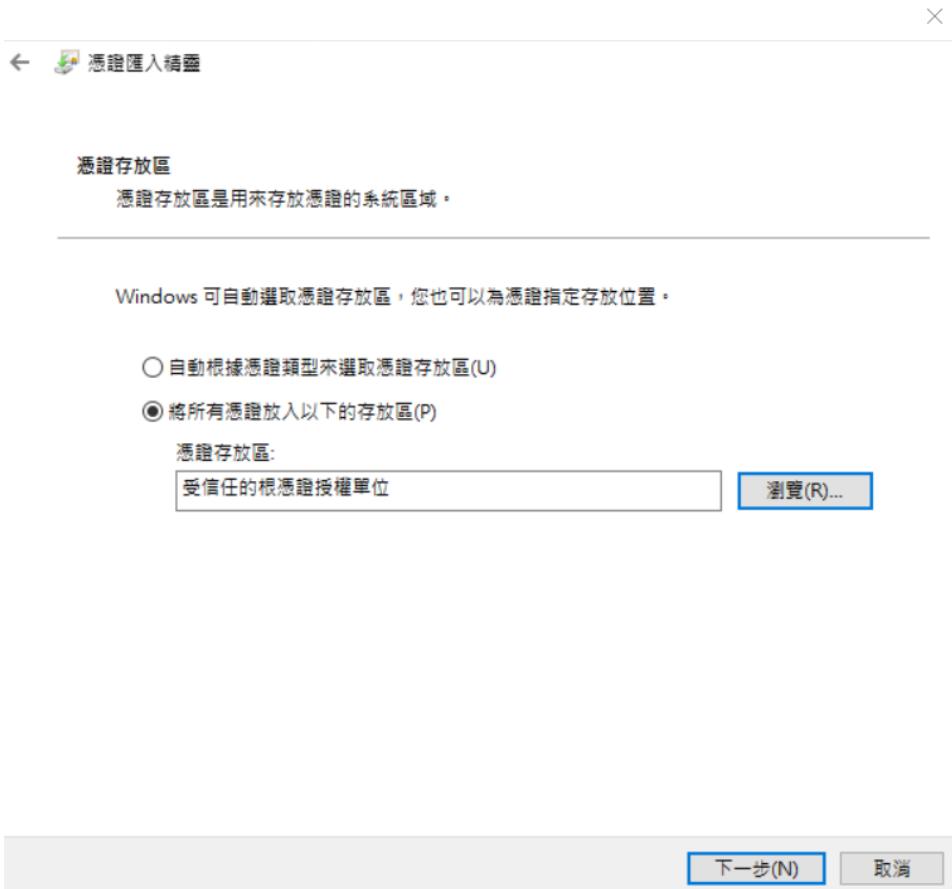


图 1.21 设定存放

- 控制帧 (Control Frame)：如 RTS 帧、CTS 帧、ACK 帧用于竞争期间的握手通信和正向确认、结束非竞争期等。
- 管理帧 (Management Frame)：如 Beacon 帧、Probe Request 帧，主要用于 STA 与 AP 之间协商、关系的控制，而控制行为如关联、认证、同步等。
- 数据帧 (Data Frame)：为承载数据的载体，用于在竞争期和非竞争期传输数据。

1.5.3.1 信标帧

信标帧 (BeaconFrame) 是相当重要的维护机制，主要来宣告某个 AP 网络的存在，同时会定期发送的信标，可让移动 WiFi 设备得知该网络的存在，从而调整加入该网络所必要的参数。而在基础网络里，AP 必须负责发送 Beacon 帧，而 Beacon 帧的所及范围即为基本服务区域。同时在基础型网络里，所有沟通都必须通过接入点，因此 WiFi 设备不能距离太远，否则无法接收到信标。下图可以看到帧格式的说明。



图 1.22 完成凭证

1.5.3.2 管理帧

管理帧 (Probe Request) 为探测请求帧，WiFi 设备将会利用 Probe Request 帧，扫描所在区域内目前有哪些 802.11 网络。下图为其帧格式的说明。

1.5.3.3 数据帧

数据帧 (Data) 为当接入点要送出一个帧给 WiFi 设备但是不必确认之前所传送的信息时，就会使用标准的数据帧。其标准的数据帧并不会征询对方是否有数据待传，因此不允许接收端传送任何数据。无竞争周期所使用的纯数据 (Data-Only) 帧和无竞争周期所使用的数据帧完全相同。

1.5.4 WiFi 探针工作

如图中描述的一样，其的 WiFi 探针其实就是一个 AP，它会定时的向自己的四周广播发送 Beacon 帧，用来通知附近的 WiFi 设备，AP 是存在，就好比它一直在向周围喊着，我在这里，大家快来连接我啊。此时 WiFi 设备如手机，平板电脑等，也会不停

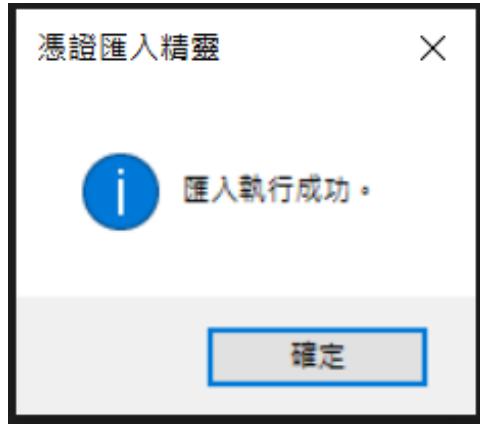


图 1.23 确定凭证

的发送着 probe 帧，去寻找附近可用的 AP。而在 probe 帧的介绍中就可以看到 probe 帧包含了设备的 MAC 地址，而当的 AP 接收到 probe 帧之后就获取了此设备的 MAC 地址，而该 AP 就是我们的 WiFi 探针。因此只要在 WiFi 探针覆盖区域内的设备打开着 WiFi，探针就能收集到他的 MAC 地址。

1.5.5 WIFI 探针能采集到的数据

- 设备 MAC 地址
- WiFi 信号强度
- WiFi 信号频道
- 信号帧类型

此外从采集数据图例中可以看到，其记录格式，探针从 MAC 抓取的设备 MAC 设备发送的 WiFi 包的的类型、子类型、信号强度与时间戳。

1.5.6 数据释义

- 探针 MAC : 就是探针本身的 MAC 地址。
- 抓取的设备 MAC : 指探针抓取到的 WiFi 信号的发射设备的 MAC 地址，一般为手机。
- 信号强度: 指探针抓取到的 WiFi 信号的强度，其最小值为"-100"，一般来说其值越大表示发射设备离探针越近。
- 设备发送的 WiFi 包的类型: 指探针抓取到的 WiFi 信号的类别，其末位数的值为 0、4、8 时，分别表示抓取到的 WiFi 信号为管理帧、控制帧、数据帧。
- 时间戳: 指探针抓取到 WiFi 信号的时间，如果探针在区域网路内使用而没有接入广域网的话，时间戳可能是不准确的。

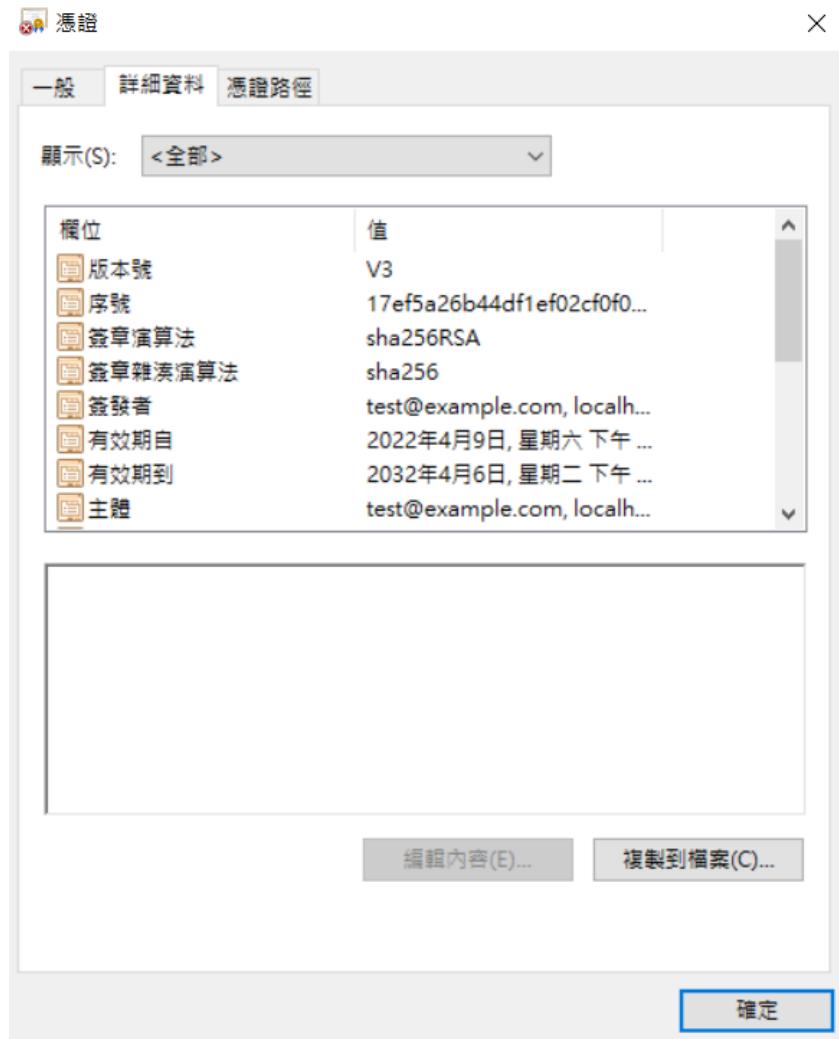


图 1.24 检视凭证资讯

1.5.7 安全性

在此讨论 WiFi 探针会不会侵犯个人隐私的问题，探针所收集的数据内容我们来看看 WiFi 探针设备究竟会收集什么信息，而从前面的分析已经看到，在不连接 WiFi 的情况下，移动设备只会发送 probe 帧，此时我们并不能通过探针访问网络进行数据传输，探针仅仅只能接收到 WiFi 设备发送的 probe 帧，收集 probe 帧携带的 MAC 地址，所以此时收集到信息是绝对无关用户个人信息和设备上其他信息。同时探针的数据处理由于探针本身设计仅仅是探测周边有些什么设备，因此并不产生大量数据，设计的时候就不会将收集到的数据存储在本身，而是通过有线连接直接发送到中心服务器上，这样即使有恶意的人将探针取走，也不能获得探针收集到的信息。同时有线连接也保证数据传输过程不容易通过电磁波的形式被监听和窃取。中心服务器一般都是在 IDC 机房里，而要进入 IDC 机房是需要经过 IDC 层层许可。因而不论是数据的传输还是存



图 1.25 执行画面

储，探针的数据都是安全的。

1.5.8 用何种设备

WiFi 探针设备和普通的无线路由器很接近，实际普通路由器就能做，只修改无线路由器的驱动部分，直接在驱动中抓取周边手机的信息。同时许多厂商也推出了专用的 WiFi 探针设备，集成为管理平台，更方便的管理收集的信息等特性。

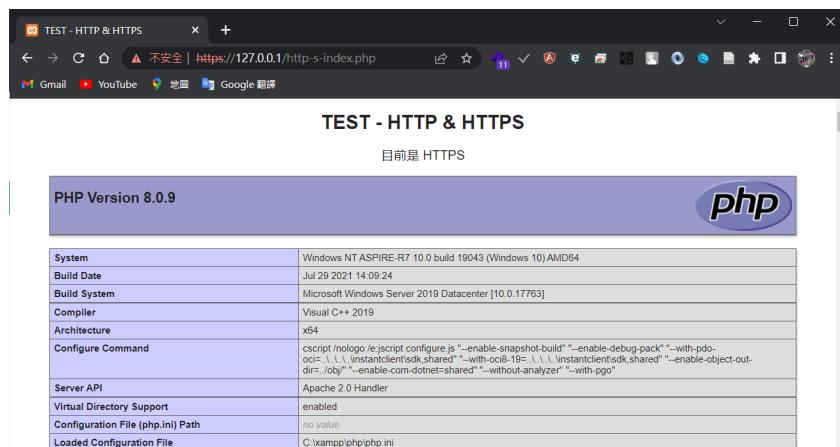


图 1.26 PHP 版本



图 1.27 JS 版本

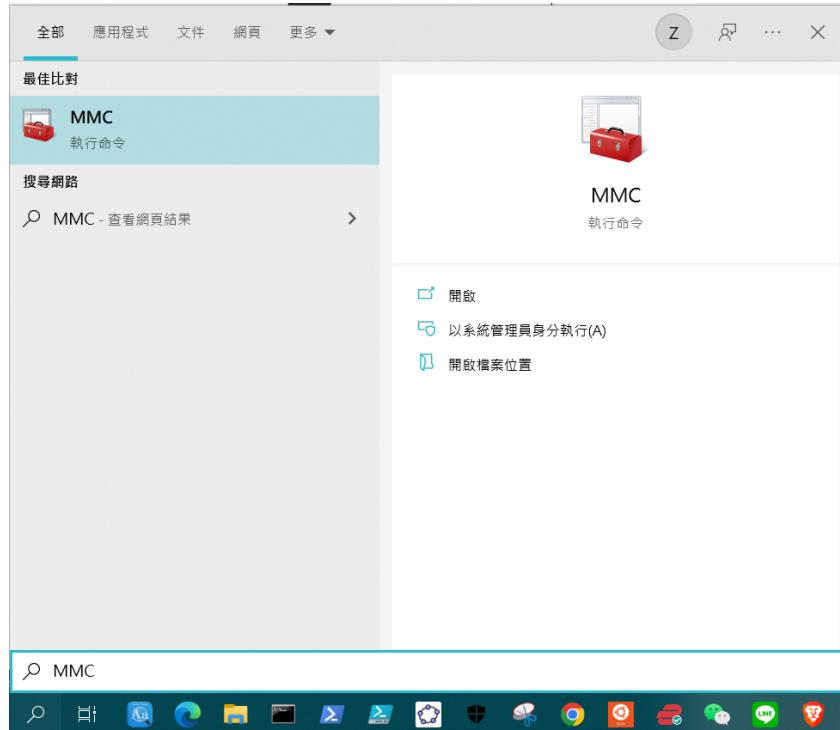


图 1.28 找 MMC



图 1.29 主控台移除

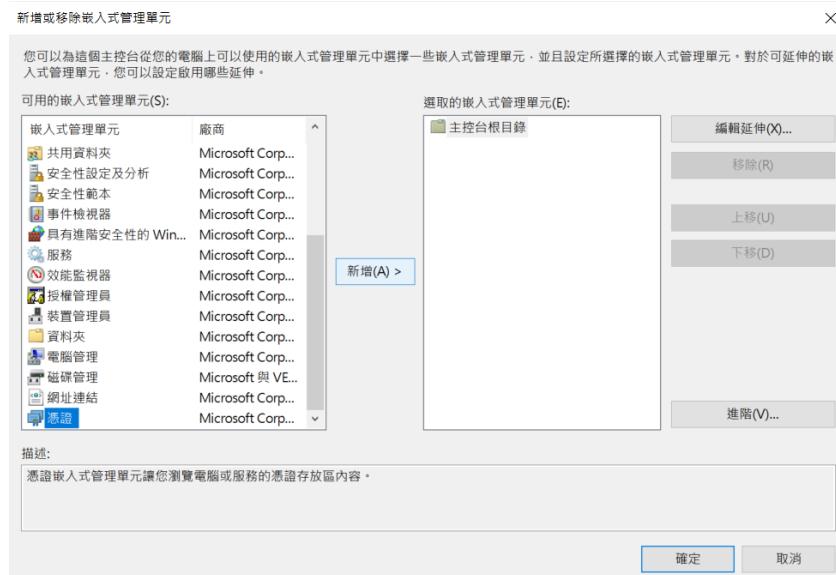


图 1.30 将凭证加入管理



图 1.31 设定帐户

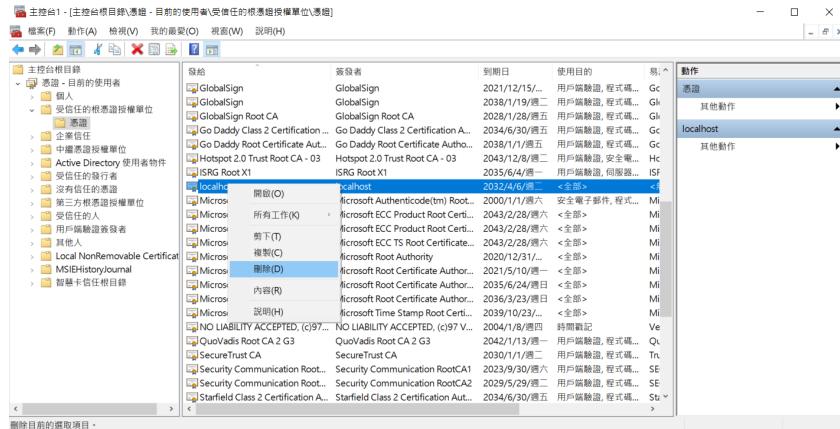


图 1.32 移除

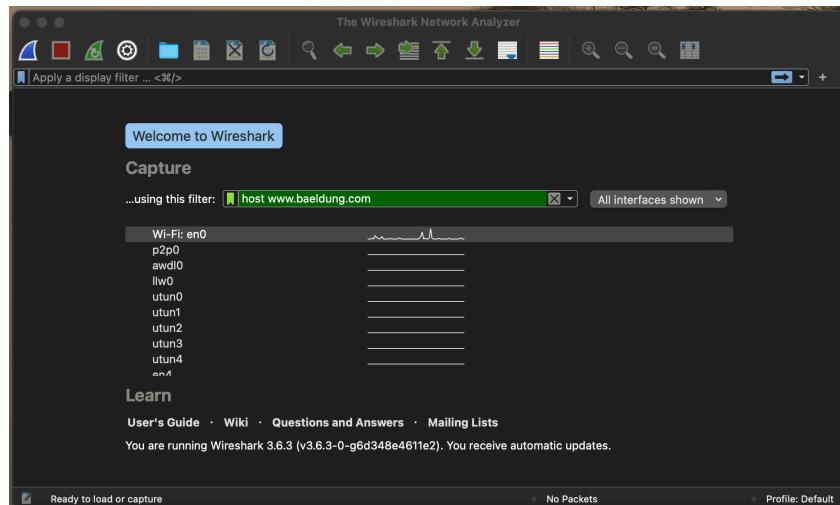


图 1.33 Mac 的 Wireshark

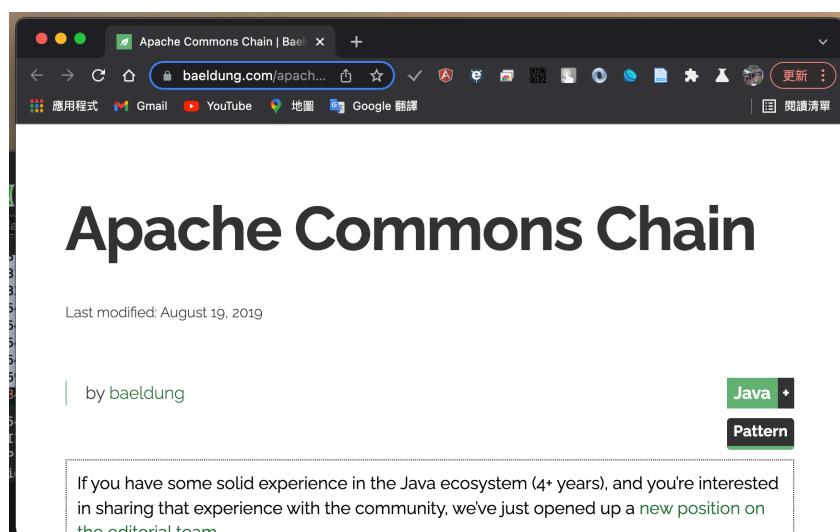


图 1.34 HTTPS 服务范例

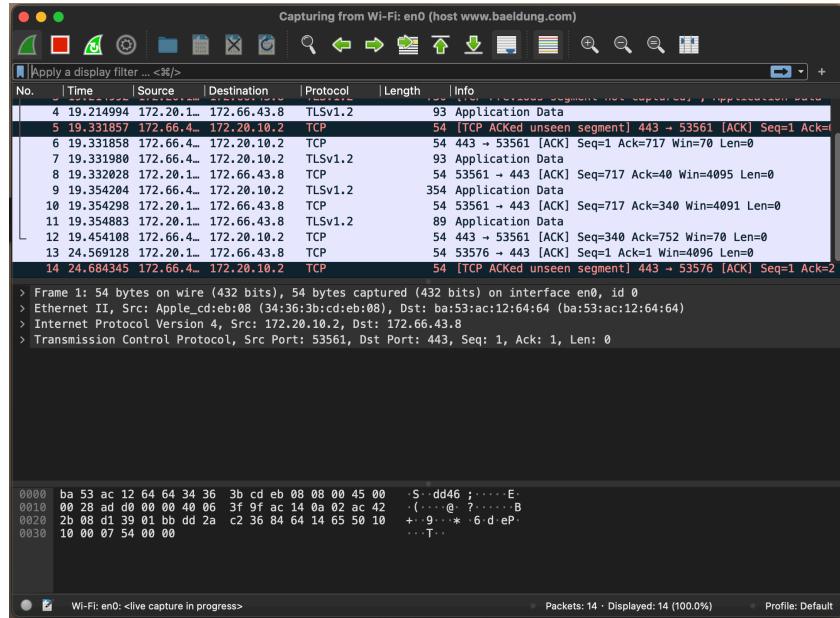


图 1.35 Wireshark 判断 HTTPS

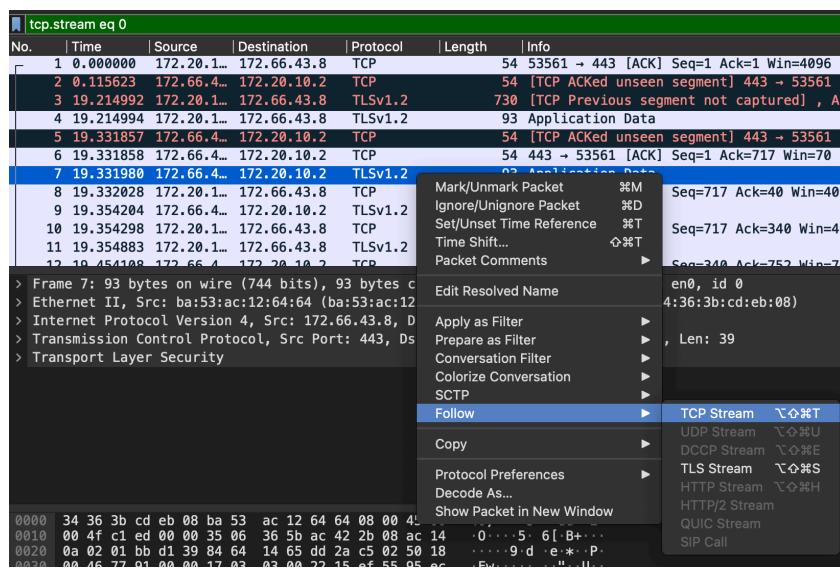


图 1.36 找 TCP 串流

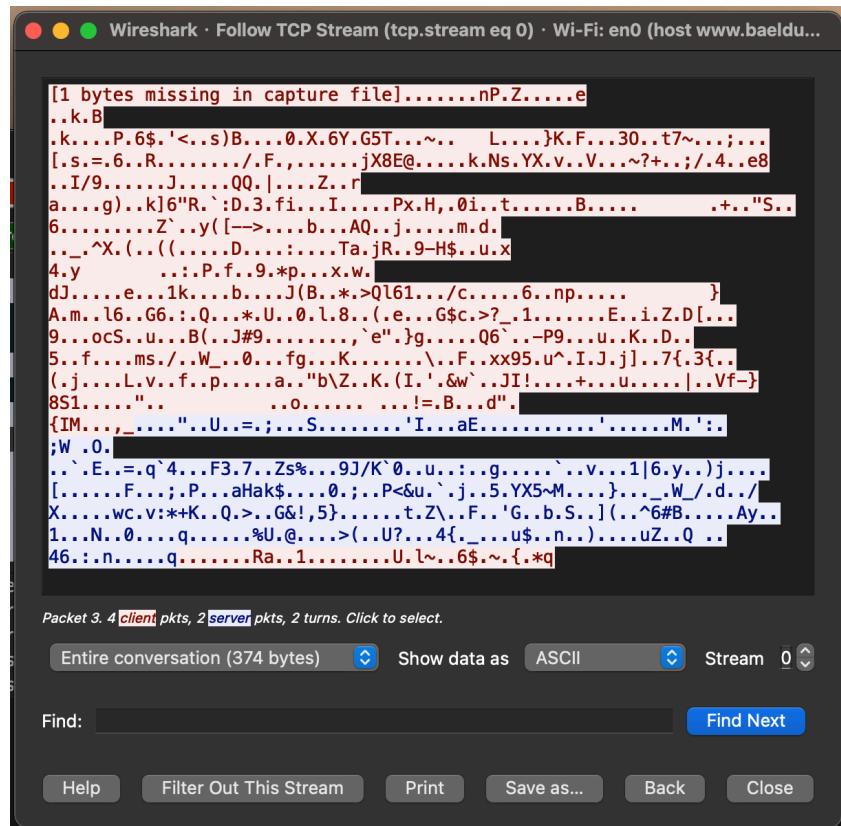


图 1.37 HTTPS 结果

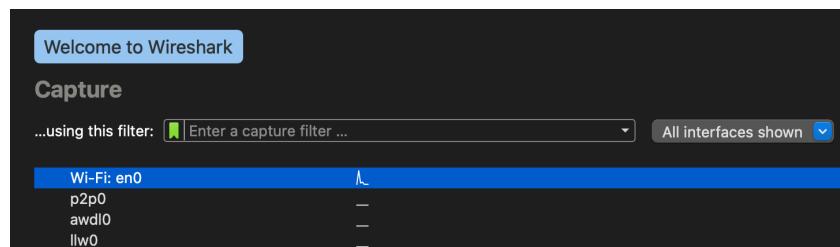


图 1.38 测试 HTTP

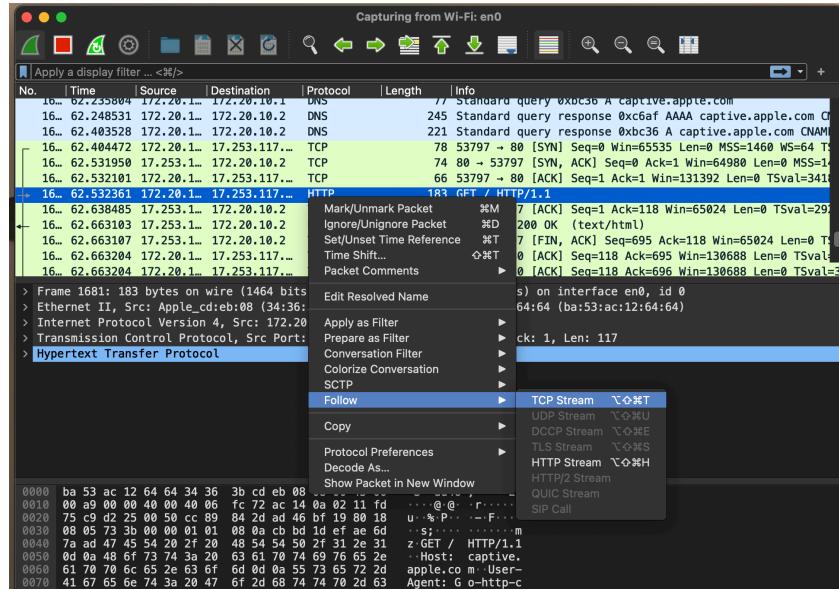


图 1.39 Wireshark 判断 HTTP

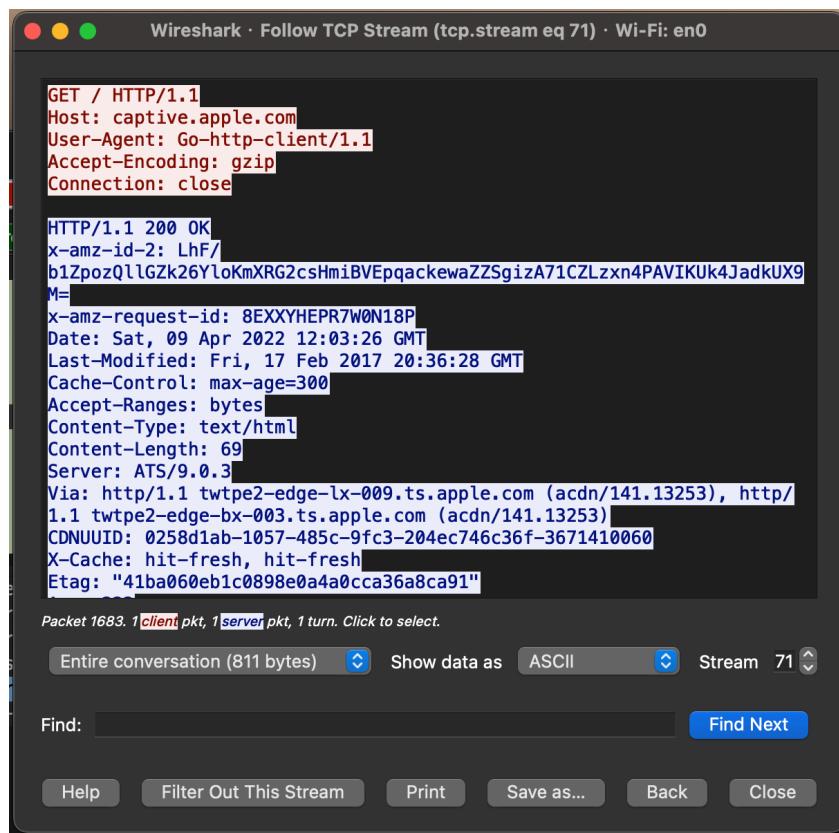


图 1.40 HTTP 结果

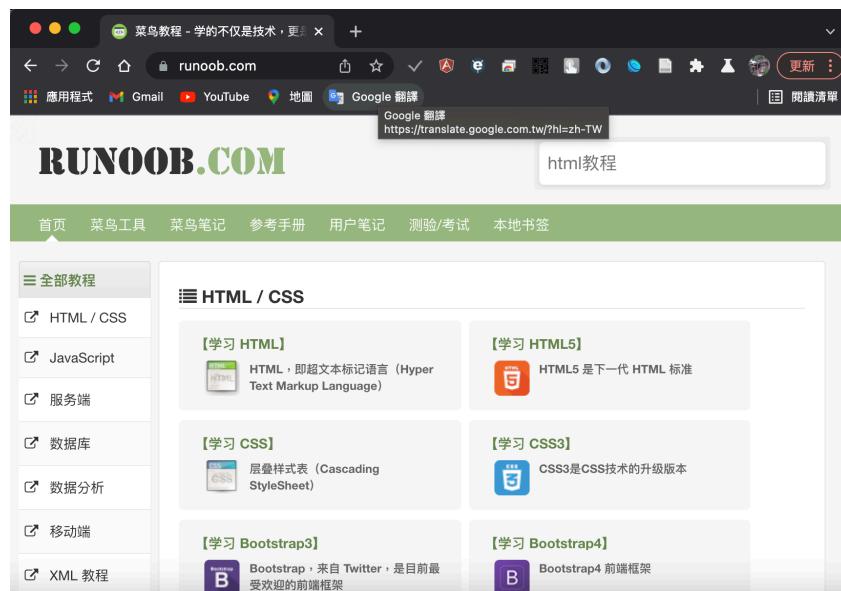


图 1.41 HTTP 服务范例

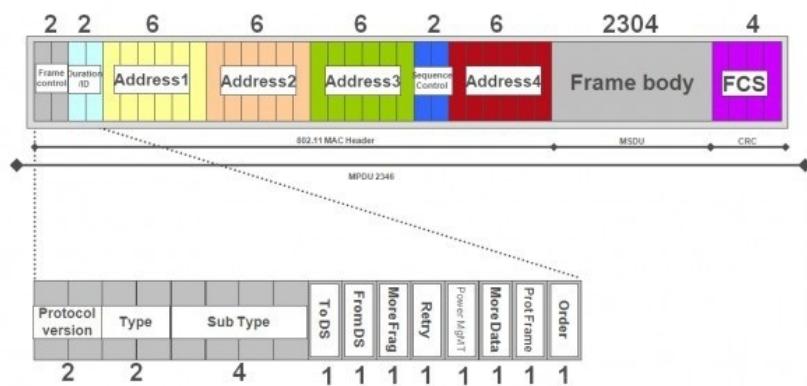


图 1.42 802.11 的帧

字段名	描述	长度(字节)
Frame Control(帧控制)字段	描述与控制MAC帧相关信息	2
Duration字段	计算帧持续时间的作用	2
Destination Address	MAC帧的目的地址	6
Source Address	MAC帧的源地址	6
BSSID(基本服务集ID)	用于过滤收到的MAC帧(在基础型网络里为工作站所关联的接入点的MAC地址)	6
Sequence Control(顺序控制字段)	用来重组帧片段以及丢弃重复帧	2
帧主体	用以传递上层信息	0-2312
FCS(帧校验序列)	验证传来的帧是否有误	4

图 1.43 帧控制字段

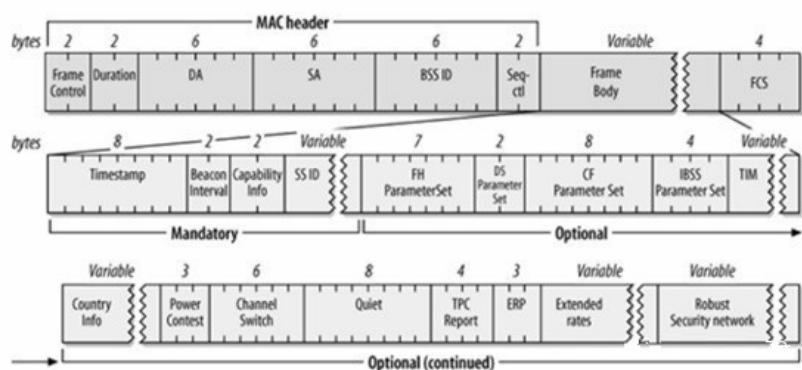


图 1.44 信标帧

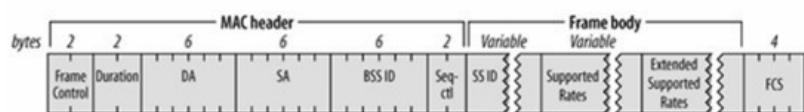


图 1.45 管理帧



图 1.46 WiFi 探针工作

```
COM3 - PuTTY
5c:cf:7f:dc:7e:58|60:01:94:0F:BD:EC|28:6C:07:5A:91:16|-65|0
5c:cf:7f:dc:7e:58|54:14:73:72:84:B4|28:6C:07:5A:91:16|-76|1
5c:cf:7f:dc:7e:58|54:EE:75:60:4C:98|28:6C:07:5A:91:16|-49|0
5c:cf:7f:dc:7e:58|60:01:94:0F:BD:EC|28:6C:07:5A:91:16|-55|0
5c:cf:7f:dc:7e:58|7C:7D:3D:B6:76:57|28:6C:07:5A:91:16|-101|0
5c:cf:7f:dc:7e:58|60:01:94:0F:BD:EC|28:6C:07:5A:91:16|-65|0
5c:cf:7f:dc:7e:58|3C:46:D8:DA:43:79|28:6C:07:5A:91:16|-57|0
5c:cf:7f:dc:7e:58|A4:71:74:4B:F2:0F|28:6C:07:5A:91:16|-87|1
5c:cf:7f:dc:7e:58|3C:46:D8:DA:43:79|28:6C:07:5A:91:16|-56|0
5c:cf:7f:dc:7e:58|74:D4:35:20:86:A1|D0:C7:C0:3E:26:62|-92|0
5c:cf:7f:dc:7e:58|74:27:EA:F6:43:55|D0:C7:C0:3E:26:62|-86|0
5c:cf:7f:dc:7e:58|5C:E0:C5:1C:55:25|C8:E7:D8:D8:B3:82|-54|0
5c:cf:7f:dc:7e:58|B8:86:87:5E:29:C7|FF:FF:FF:FF:FF:FF|-88|0
5c:cf:7f:dc:7e:58|7C:7D:3D:B6:76:57|28:6C:07:5A:91:16|-99|0
5c:cf:7f:dc:7e:58|60:01:94:0F:BD:EC|28:6C:07:5A:91:16|-67|0
5c:cf:7f:dc:7e:58|74:D4:35:20:86:A1|D0:C7:C0:3E:26:62|-94|0
5c:cf:7f:dc:7e:58|74:27:EA:F6:43:55|D0:C7:C0:3E:26:62|-82|0
5c:cf:7f:dc:7e:58|68:3E:34:64:11:20|28:6C:07:5A:91:16|-89|1
5c:cf:7f:dc:7e:58|5C:E0:C5:1C:55:25|C8:E7:D8:D8:B3:82|-49|0
5c:cf:7f:dc:7e:58|68:3E:34:64:11:20|28:6C:07:5A:91:16|-74|0
5c:cf:7f:dc:7e:58|3C:46:D8:DA:43:79|28:6C:07:5A:91:16|-55|0
5c:cf:7f:dc:7e:58|7C:7D:3D:B6:76:57|28:6C:07:5A:91:16|-97|1
5c:cf:7f:dc:7e:58|54:A0:50:7D:02:5F|28:6C:07:5A:91:16|-52|0
```

图 1.47 WIFI 探针能采集到的数据

第二章 量子计算机与信息安全对密码学的影响

本章尝试整理近来量子计算机的发展对信息安全领域的影响，包括带来的挑战与对策。

- IEEE

第三章 可信计算技术与近来深度学习对密码学的影响

第四章 工作总结

参考文献

- [1] 邱锡鹏. 神经网络与深度学习[M/OL]. 北京: 机械工业出版社, 2020. [https://nndl.github.io/.](https://nndl.github.io/)

致谢

非常感谢我的导师朱跃生教授，在 XXX 课让学生上充分实作了 LaTeX 的各类模板与写法，该流程也改善了自己目前的开发与研究工作模式，同时也演练过名为 Zotero 的开源授权的文献管理软体，同时也将此流程在其他课程的作业上进行测试获得良好的回馈。最后感谢在这一年来一起寒窗苦读得同学与所有老师，还有默默在开源社群与前沿研究奉献的技术人员跟研究者们。