

親愛的顧客 您好

感謝 貴公司對臺灣網路認證公司 TWCA SSL 憑證服務的支持。

提醒您：各大瀏覽器廠商將於 2020 年 3 月份起，停止對 TLS 1.0 與 TLS 1.1 傳輸協定之支援。

參考資料

[Safari] <https://webkit.org/blog/8462/deprecation-of-legacy-tls-1-0-and-1-1-versions/>

[Firefox] <https://blog.mozilla.org/security/2018/10/15/removing-old-versions-of-tls/>

[Edge] <https://blogs.windows.com/msedgedev/2018/10/15/modernizing-tls-edge-ie11/>

[Chrome] <https://security.googleblog.com/2018/10/modernizing-transport-security.html>

➤ 現況

Google Chrome 瀏覽器自版本 79 起(2019/12 推出) 已針對網站系統所提供之 TLS 傳輸協定版本進行檢查，若網站無支援 TLS 1.2 傳輸協定即會出現警告訊息(圖一)，Mozilla Firefox 瀏覽器亦有類似安排(圖二)。



(圖一、Chrome 79, 網站不支援 TLS v1.2 傳輸協定)



(圖二、Firefox 72, 網站不支援 TLS v1.2 傳輸協定)

➤ 即將實施

目前已知 Google Chrome 瀏覽器將於版本 81 (2020/03 推出)、Mozilla Firefox 瀏覽器將於版本 74 (2020/03 推出) 完全停止對 TLS 1.0 與 1.1 傳輸協定之支援，屆時 貴公司網站若無支援 TLS 1.2 傳輸協定，則使用最新瀏覽器之使用者將無法以 HTTPS 方式連線瀏覽 貴公司網站，造成網站使用上之巨大衝擊。

➤ 檢查方式

可利用 Qualys SSL Labs 提供之網站 SSL 檢查免費服務，

確認 貴公司網站有無支援 TLS 1.2 傳輸協定：

- 1、 連至以下網址 <https://www.ssllabs.com/ssltest/>
- 2、 於 Hostname 欄位輸入網址，以本公司網站為例，輸入 www.twca.com.tw，

輸入後請按 "Submit" 開始檢查。

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

Hostname:

- 3、 請稍等片刻，直到 "Grade" 出現英文字，代表檢查完成，再點選 "Server" 欄位

顯示之 IP，產生結果報告。

	Server	Test time	Grade
1	2001:b031:1306:ff00:0:0:0:1011 2001-b031-1306-ff00-0000-0000-0000-1011.hinet-ip6.hinet.net Ready	Wed, 19 Feb 2020 12:59:38 UTC Duration: 120.8 sec	A

- 4、 確認結果報告中 Protocols 章節之 "TLS 1.2" 檢查項目，

Configuration	
Protocols	
TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No
For TLS 1.3 tests, we only support RFC 8446.	

若檢查結果為 "Yes"，即代表貴公司網站已支援 TLS 1.2，將不受 3 月份瀏覽器更版之影響；

若檢查結果為 "No"，即代表貴公司網站尚未支援 TLS 1.2，請盡快安排網站之設定調整作業。

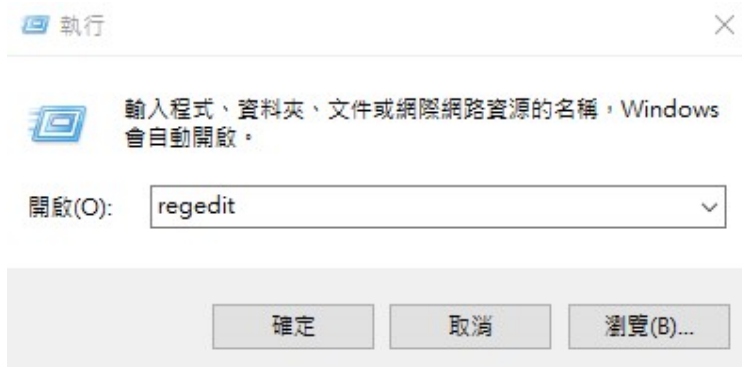
➤ 網站之設定調整

以下針對市面上三種常用的網頁伺服器(IIS、Apache、Tomcat)，說明如何設定支援 TLS 1.2 的操作步驟。

1、 IIS

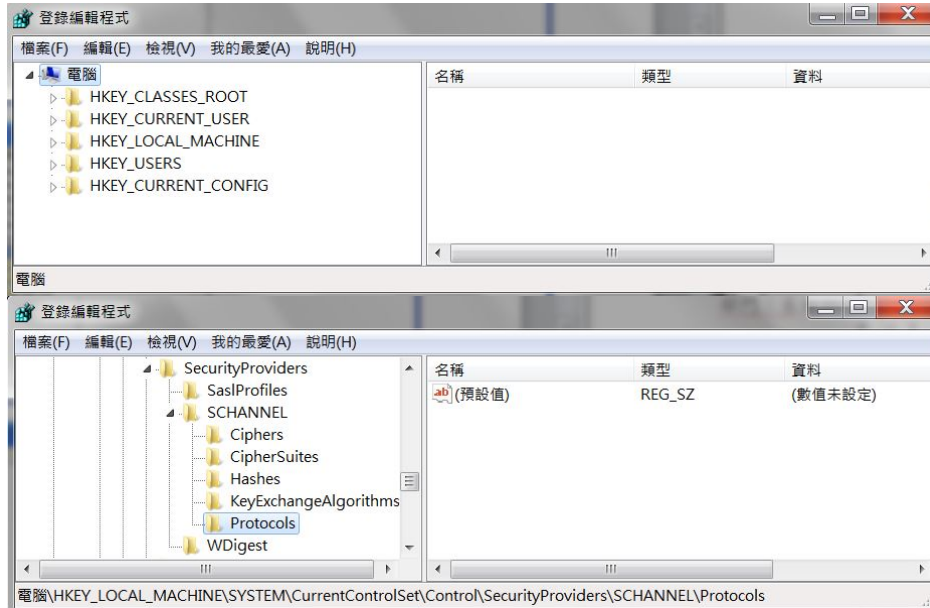
1.1 開啟登入編輯程式

於桌面左下角按滑鼠右鍵，點選”執行”，於搜尋對話方塊中輸入 `regedit` 後按”確定”，開啟「登入編輯程式」。



1.2 依序展開登入機碼

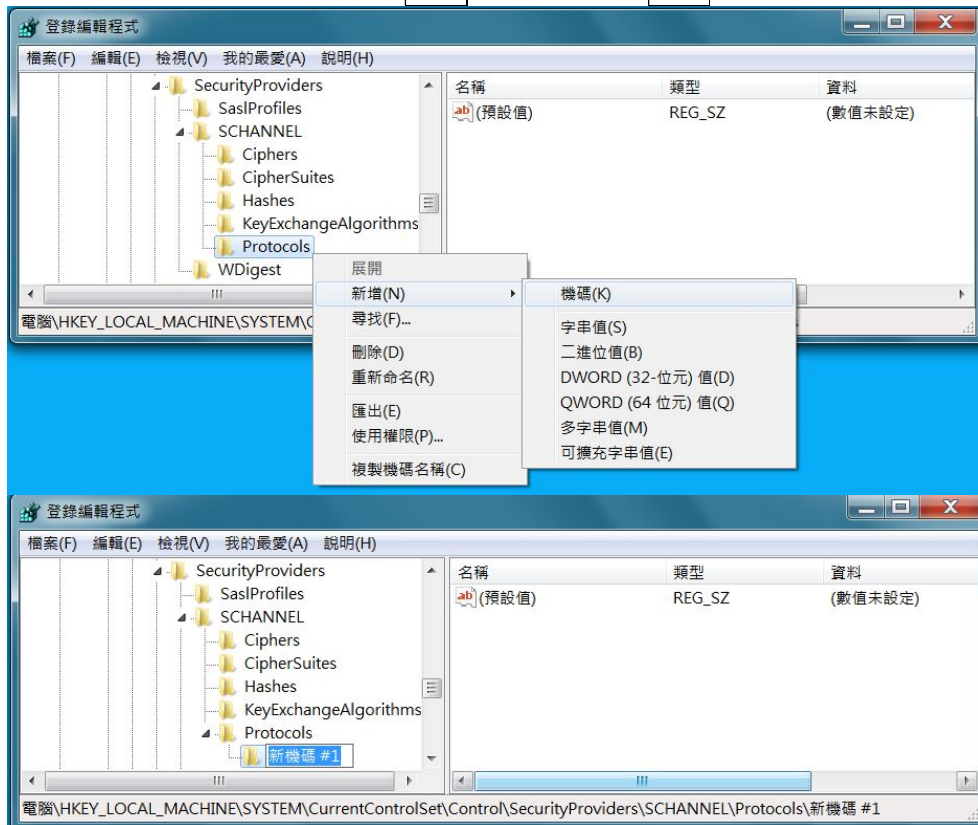
依序展開 `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols`



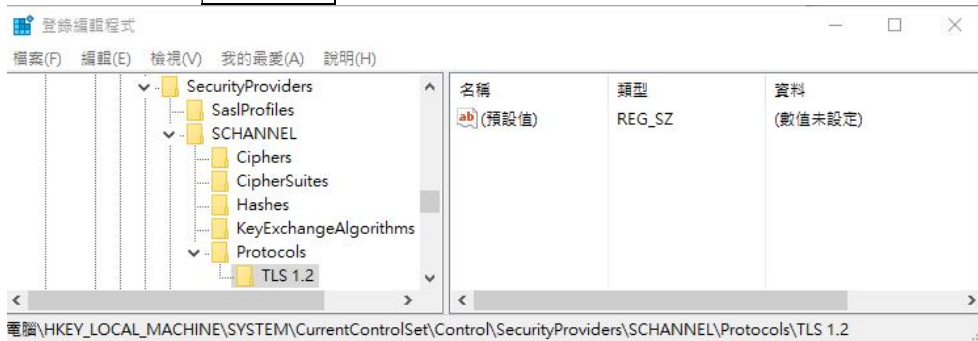
1.3 新增 TLS 1.2 機碼

※ 如系統本身已有 TLS 1.2 機碼則無需執行，反之若無，請參照以下步驟新增。

1.3.1 於 **Protocols** 按右鍵選擇 **新增**，資料類型為 **機碼**。



1.3.2 將新增之 **新機碼#1** 重新命名成 **TLS 1.2**。



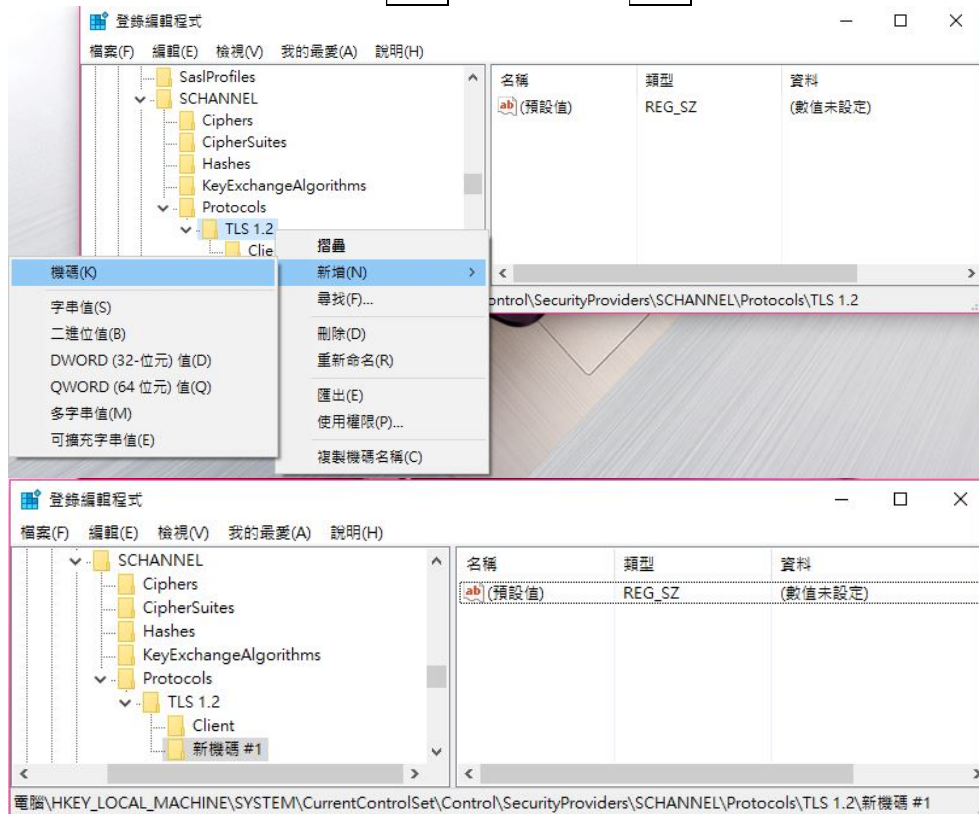
1.3.3 於 **TLS 1.2** 按右鍵選擇 **新增**，資料類型為 **機碼**。



1.3.4 將新增之新機碼#1重新命名成 Client。



1.3.5 於 TLS 1.2 按右鍵選擇新增，資料類型為機碼。

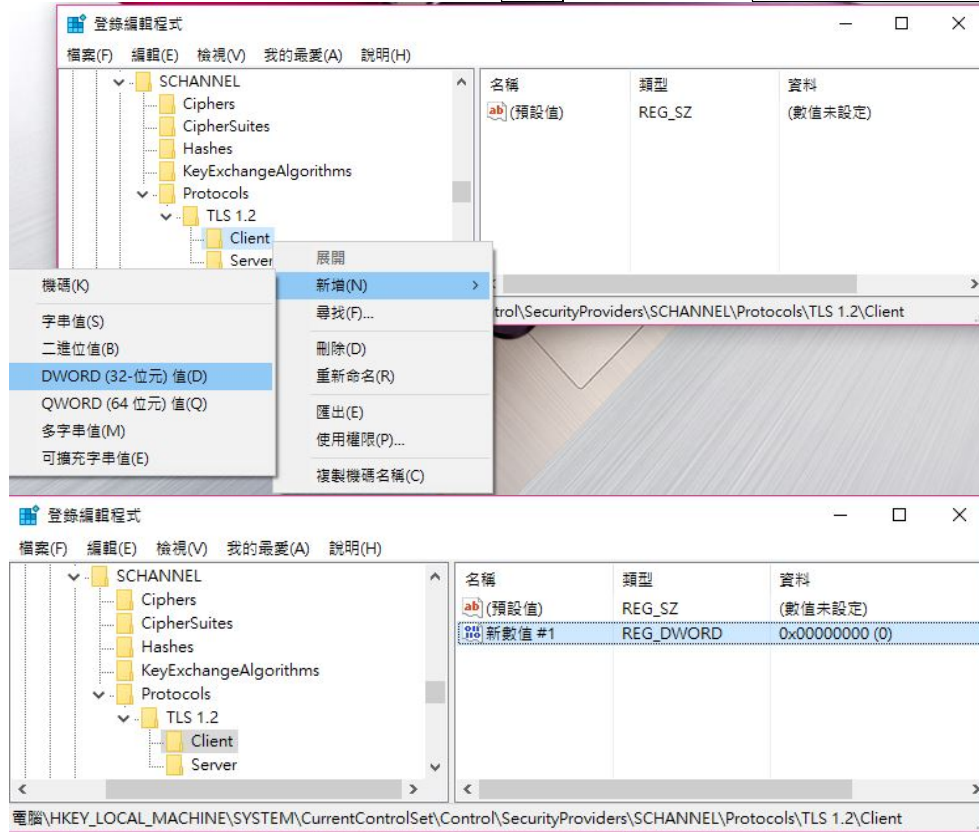


1.3.6 將新增之新機碼#1重新命名成 Server。

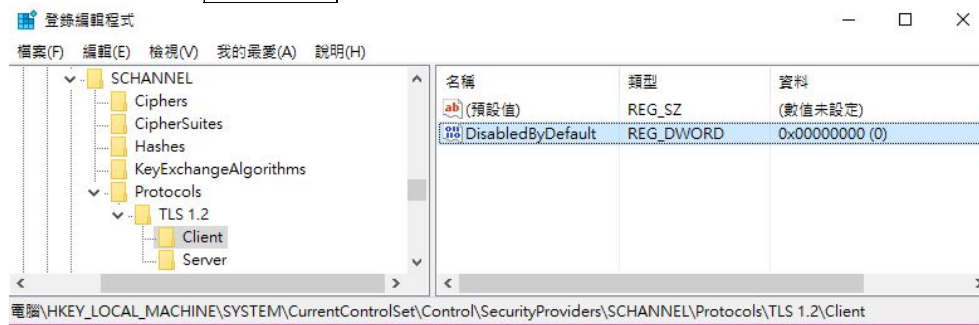


1.4 修改 TLS 1.2 機碼-Client

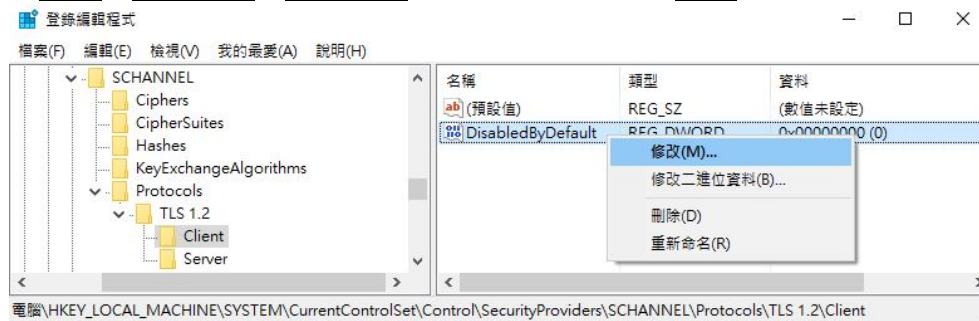
1.4.1 於 **TLS 1.2** 的 **Client** 按右鍵選擇 **新增**，資料類型為 **DWORD(32-位元)值**。

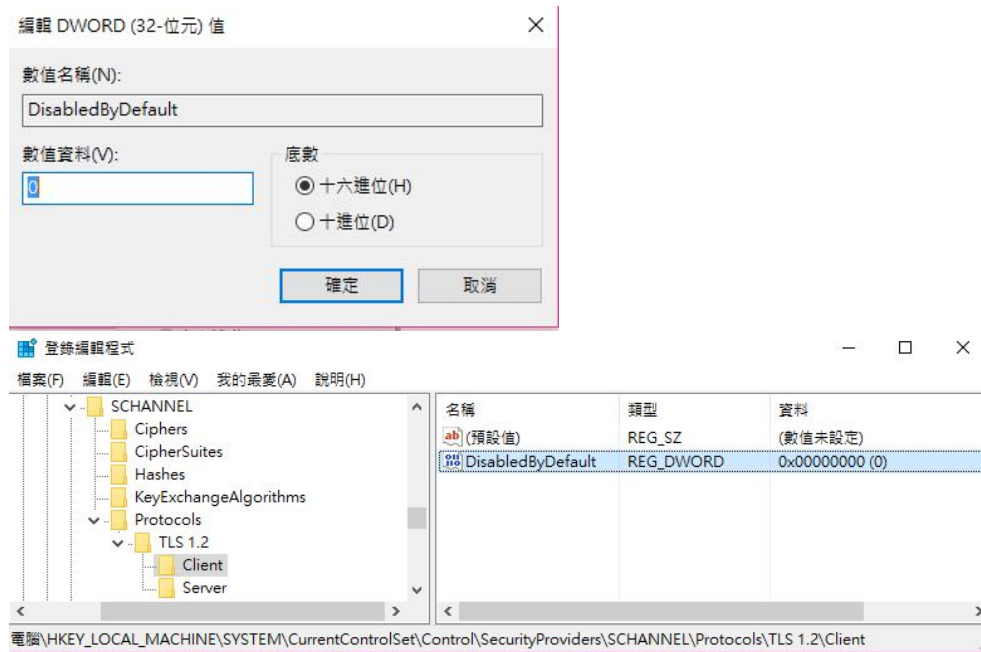


1.4.2 將新增之 **新數值#1** 重新命名成 **DisabledByDefault**。



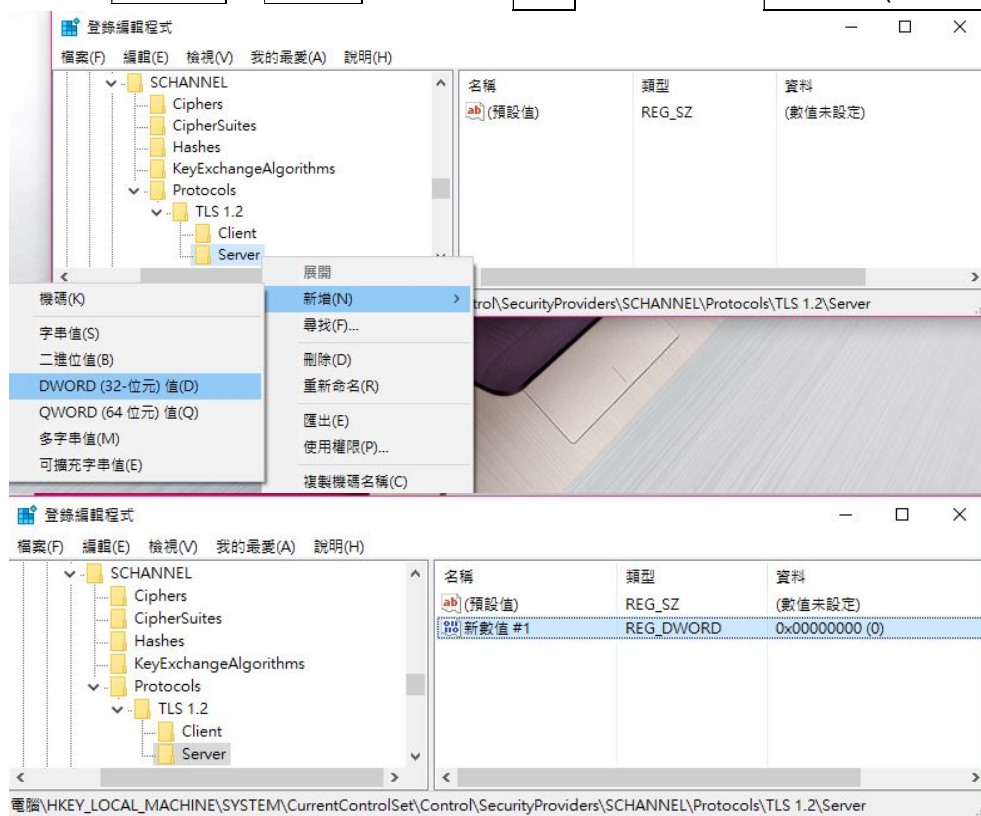
1.4.3 於 **DisabledByDefault** 按右鍵選擇 **修改**，將底數為 **十六進位** 之 **數值資料**，數值設定為 **0**，按 **確定**。



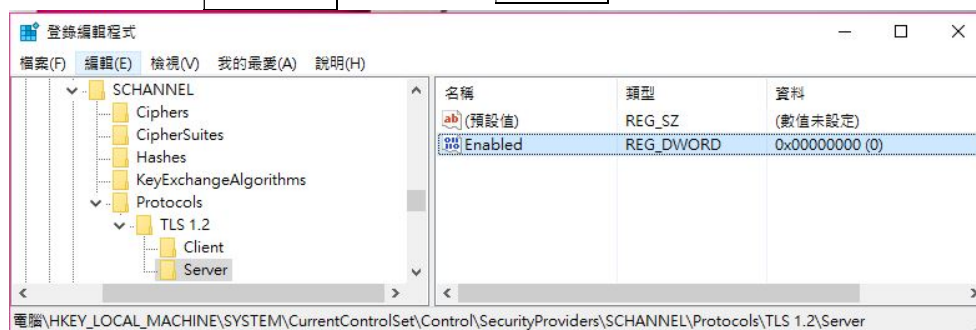


1.5 修改 TLS 1.2 機碼-Server

1.5.1 於 **TLS 1.2** 的 **Server** 按右鍵選擇 **新增**，資料類型為 **DWORD(32-位元)值**。

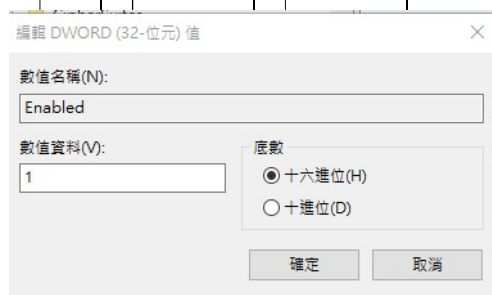


1.5.2 將新增之新數值#1 重新命名成 **Enabled**。



1.5.3 於 **Enabled** 按右鍵選擇 **修改**，

將底數為十六進位之數值資料，數值設定為 **1**，按 **確定**。



1.6 設定完成後，請重新啟動主機，使設定生效。

2、 Apache

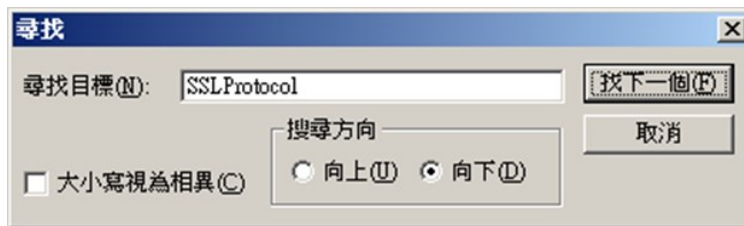
2.1 開啟 Apache 設定檔

Apache SSL 協定設定值儲存於 Apache 設定檔內

(依 Apache 版本不同可能存放於 httpd.conf、httpd-ssl.conf 或 ssl.conf)



2.2 搜尋 SSLProtocol 參數



2.3 修改 SSLProtocol 參數內容

設定 SSL 協定。可以用『+、-』分別表示『允許、拒絕』某些特定的協定。

SSLProtocol -all +TLSv1.2 (即代表允許 TLS 1.2，並拒絕其他較過時的協定)

```
SSLProtocol -all +TLSv1.2
```

備註：如需同時允許多個協定，請在最後面加上協定名稱(每個協定要用空格間隔)

```
SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2
```

(代表同時允許 TLS 1.0、TLS 1.1、TLS 1.2 傳輸協定)

2.4 設定完成後請儲存檔案，並重新啟動 Apache 服務，使設定生效。

3、 Tomcat

3.1 開啟 Tomcat 設定檔

Tomcat SSL 協定設定值儲存於 Tomcat 設定檔內

(預設路徑為\$TOMCAT_HOME \conf 目錄下的 server.xml 檔案)

3.2 開啟 Tomcat 設定檔 server.xml，並搜尋「sslProtocol」字串，可找到其中跟 SSL 相關的設定值



```
server.xml - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

<!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the JSSE configuration, when using APR, the
      connector should be using the OpenSSL style configuration
      described in the APR documentation -->

<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="c:\mykeystore.jks"
           keyAlias="keyname"
           keystorePass="xxx" />
```

3.3 如為 Tomcat 5 或 Tomcat 6 (6.0.38 前版本)

請新增「sslProtocols」設定，sslProtocols="TLSv1.2"



```
server.xml - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

      This connector uses the JSSE configuration, when using APR, the
      connector should be using the OpenSSL style configuration
      described in the APR documentation -->

<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="c:\mykeystore.jks"
           keyAlias="keyname"
           keystorePass="xxx"
           sslProtocols="TLSv1.2" />
```

備註：如需同時允許多個協定，請在最後面加上協定名稱(每個協定要用,間隔)

sslProtocols="TLSv1,TLSv1.1,TLSv1.2"

(代表同時允許 TLS 1.0、TLS 1.1、TLS 1.2 傳輸協定)

3.4 如為 Tomcat 6 (6.0.38 及之後版本) 或 Tomcat 7、Tomcat 8

請新增「sslEnabledProtocols」設定，sslEnabledProtocols="TLSv1.2"



```
server.xml - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation -->

<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="c:\mykeystore.jks"
keyAlias="keyname"
keystorePass="xxx"
sslEnabledProtocols="TLSv1.2" />
```

備註：如需同時允許多個協定，請在最後面加上協定名稱(每個協定要用,間隔)

sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"

(代表同時允許 TLS 1.0、TLS 1.1、TLS 1.2 傳輸協定)

3.5 設定完成後請儲存檔案，並重新啟動 Tomcat 服務，使設定生效。