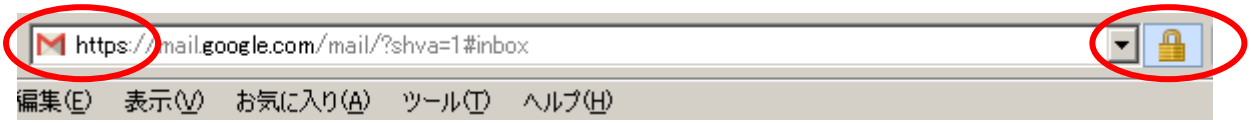
Web サーバ設定その 2 （SSL：暗号化通信）

皆さんはホームページにおいて個人情報を送る時、盗聴の心配をしていませんか？

以前、紹介した WireShark 等のパケットキャプチャソフトを使えば簡単に中身を見ることが出来ます。そこで、個人情報を入力するホームページでは安全のために暗号化通信(SSL)が施されています。皆さんも「金色の鍵」のマークが付いているホームページ(https の表示：下図)を見たことが有ると思います。



SSL は公開鍵暗号や秘密鍵暗号、デジタル証明書、ハッシュ関数などのセキュリティ技術を組み合わせ、データの盗聴や改ざん、なりすましを防ぐことができます。OSI 参照モデルではセッション層(第 5 層)とトランスポート層(第 4 層)の境界で動作し、HTTP や FTP などの上位のプロトコルを利用するアプリケーションソフトからは、特に意識することなく透過的に利用することができます。現在は SSL 3.0 をもとに若干の改良が加えられた TLS 1.0 が RFC 2246 として IETF で標準化されています。

※　SSL 【 Secure Sockets Layer 】

Netscape Communications 社が開発した、インターネット上で情報を暗号化して送受信するプロトコル。現在インターネットで広く使われている WWW や FTP などのデータを暗号化し、プライバシーに関わる情報やクレジットカード番号、企業秘密などを安全に送受信することができる。

今回はサーバ独自の SSL 自己署名証明書を作成します。しかし独自に作成した証明書ではアクセス時に警告が出るため、商用で使う場合はベリサイン社等の正規の認証局が発行する証明書を利用した方がよいでしょう。

(1)SSL 設定　(教科書 P138～)

SSL 経由での暗号化通信ができるよう設定

①　SSL のインストール

[hal@motobe ~]$ **rpm　-q　mod_ssl**
mod_ssl-2.4.53-11.el9.x86_64　　←インストールされています

※ネットからインストール出来ない時は下記の方法でインストール
[hal@motobe ~]$**sudo　yum　-y　install　mod_ssl**

②　httpd の設定(サーバー用秘密鍵・証明書作成)

[hal@motobe ~]# **cd　/etc/pki/tls/certs/**
[hal@motobe ~]$ **ls　/etc/pki/tls/certs**
ca-bundle.crt　　localhost.crt
ca-bundle.trust.crt　postfix.pem
※デフォルトの秘密鍵と証明書しかありません

証明書の作成(自己証明)

SSL/TLS を用いた信頼性の高い暗号化通信を構築するために必要な証明書を作成します。

当例では自己署名の独自証明書を作成します。 ただし、自己署名の証明書はアクセス時に警告が出るため、一般的にはその先には進んでもらえないでしょう。 よって、自己署名の証明書は、開発やテスト目的での内部ネットワーク利用に限り、 不特定多数からアクセスされるネットワーク上で利用する場合は、正規の認証局が発行する証明書を利用した方がよいでしょう。

[hal@motobe ~]$ **sudo vi /etc/ssl/openssl.cnf**

最終行に追記

[motobe.com]

subjectAltName = DNS:motobe.com,www.motobe.com

セクション名は任意

DNS:(自身のホスト名)

DNS を複数指定する場合はカンマ区切り

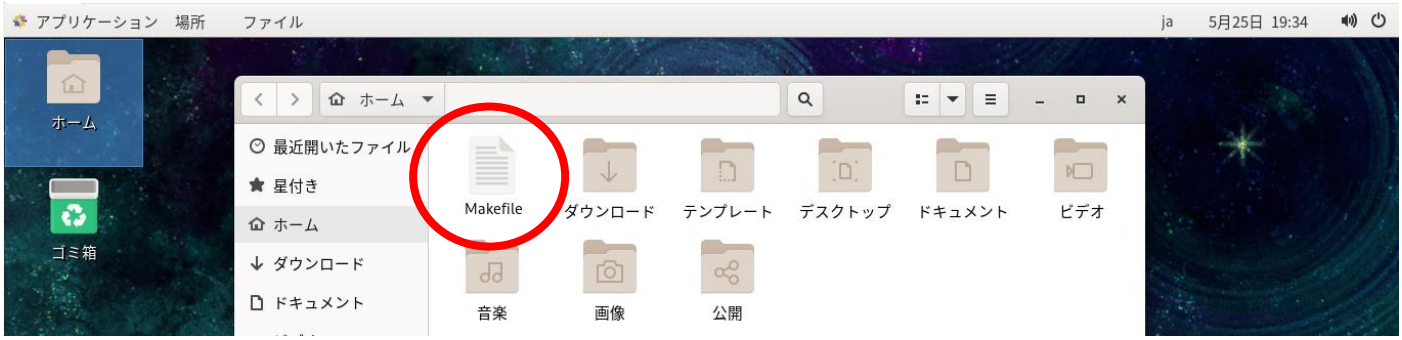
●秘密キー(server.key)と証明書(server.csr)の作成

素材サーバや Tearms にある「Makefile」(秘密キーと証明書の作成プログラム)を Linux のホームディレクトリにコピペ

・VMware と UTM : Linux のホームディレクトリに直接ドラックアンドドロップ

・VirtualBox : SCP を用いてコピー

※「NT43 Linux 資料 08 SSH(5 月 22 日)」の(5) SSH(SCP)を用いたファイル転送 参考



[hal@motobe ~]$**pwd**

/home/hal

[hal@motobe ~]$ **sudo cp Makefile /etc/pki/tls/certs**

[hal@motobe ~]$ **cd /etc/pki/tls/certs/**

[hal@motobe certs]$ **ls**

Makefile ca-bundle.crt ca-bundle.trust.crt postfix.pem

※「Makefile」で秘密キーと証明書を作成します

[hal@motobe certs]$ **sudo sed -i 's/365/3650/g' Makefile**
※証明書の有効期限を 1 年から 10 年に変更

[hal@motobe certs]$ **sudo make server.crt** ← サーバ用秘密鍵・証明書作成
umask 77 ; ¥
/usr/bin/openssl genrsa -aes128 2048 > server.key
Enter pass phrase: ←任意のパスワード(今回は **halhal**)を応答※表示はされない
Verifying - Enter pass phrase: ←任意のパスワード(今回は **halhal**)を応答(確認)※表示はされない
umask 77 ; ¥
/usr/bin/openssl req -utf8 -new -key server.key -x509 -days 3650 -out server.crt
Enter pass phrase for server.key:←上記で応答したパスワード(今回は **halhal**)を応答※表示はされない
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) [XX]:**jp** ← 国名応答
State or Province Name (full name) []:**Osaka** ← 都道府県名応答
Locality Name (eg, city) [Default City]:**Osaka** ← 市区町村名応答
Organization Name (eg, company) [Default Company Ltd]:**Motobe.com** ←サイト名応答(なんでもいい)
Organizational Unit Name (eg, section) []: ← **空 ENTER**
Common Name (eg, your name or your server's hostname) []:**www.motobe.com** ←Web サーバ名応答
Email Address []:**root@mail.motobe.com** ← 管理者メールアドレス応答
※青の部分は各自の設定に合わせてください

[hal@motobe certs]$ **ls**
Makefile ca-bundle.trust.crt **server.crt**
ca-bundle.crt postfix.pem **server.key**
※server.kye：秘密キー、server.crt：証明書

[hal@motobe certs]$ **sudo openssl rsa -in server.key -out server.key**
← サーバ用秘密鍵からパスワード削除
Enter pass phrase for server.key:
← サーバ用秘密鍵・証明書作成時のパスワード(今回は **halhal**)応答※表示はされない
writing RSA key
※パスワードを削除するのは、Web サーバ起動時にパスワードを要求されないようにするため

(注意！)作成エラーが出たら

(1)失敗した時は上記設定の再確認。どうしても無理の時は「server.key」、「server.crt」を削除してやり直し！
保存先は ssl.conf　に書いてあります

/etc/pki/tls/certs/server.crt
/etc/pki/tls/certs/server.key

※削除はコマンドでも GUI(マウスを用いて)でも構いません。　←出来ればスマートにコマンドで消して欲しいですが…(^_^;)

(2)エラー表示が出てそれ以上進まなくなった時は「Makefile」が壊れているかもしれません。ホームディレクトリにコピーした「Makefile」を使用してもう一度、最初「make　server.crt」からやり直す

③SSL 設定

[hal@motobe ~]$ sudo　cp　/etc/httpd/conf.d/ssl.conf　/etc/httpd/conf.d/ssl.conf.old
※あらかじめ、設定ファイルのバックアップ(ssl.conf.old)を取っておく

[hal@motobe ~]$ ls　/etc/httpd/conf.d/
README　　manual.conf　ssl.conf.old
autoindex.conf　php.conf　　userdir.conf
fcgid.conf　　ssl.conf　　welcome.conf.old

[root@motobe certs]# vi　/etc/httpd/conf.d/ssl.conf　　←　ApacheSSL 設定ファイル編集

#43 行目
DocumentRoot "/var/www/html"　　←　#を削除(コメント解除)

#44 行目：＃を削除し、サーバ名指定
ServerName　www.motobe.com:443

85 行目 ： 取得した証明書を指定
SSLCertificateFile /etc/pki/tls/certs/server.crt

93 行目 ： 取得した鍵ファイルを指定
SSLCertificateKeyFile /etc/pki/tls/certs/server.key

④web サーバの再起動と FW 設定

[root@motobe ~]#**sudo systemctl restart httpd**
※ここでエラーが出たら、英語のメッセージをよく見て下さい！「line XX(数字)がダメ」、「file がダメ」など

(エラー例)

[hal@motobe ~]$ **sudo systemctl restart httpd**
Job for httpd.service failed because the control process exited with error code.
See "**systemctl status httpd.service**" and "journalctl -xeu httpd.service" for details.
[hal@motobe ~]$ sudo systemctl status httpd
× httpd.service - The Apache HTTP Server
(省略)
5 月 24 18:31:57 motobe.com systemd[1]: Starting The Apache HTTP Server...
5 月 24 18:31:57 motobe.com httpd[71752]: AH00526: Syntax error on line 85 of /etc/httpd/conf.d/ssl.conf:
5 月 24 18:31:57 motobe.com httpd[71752]: SSLCertificateFile: file '/etc/pki/tls/certs/server.crt' does not exist or is empty
※上記例では「server.crt ファイル」が存在しない！ってエラーが出てますね！

Firewalld を有効にしている場合は、HTTPS サービスの許可が必要です。なお、HTTPS は 443/TCP を使用します。
[root@motobe certs]# **firewall-cmd --add-service=https --permanent**
success
※https の追加

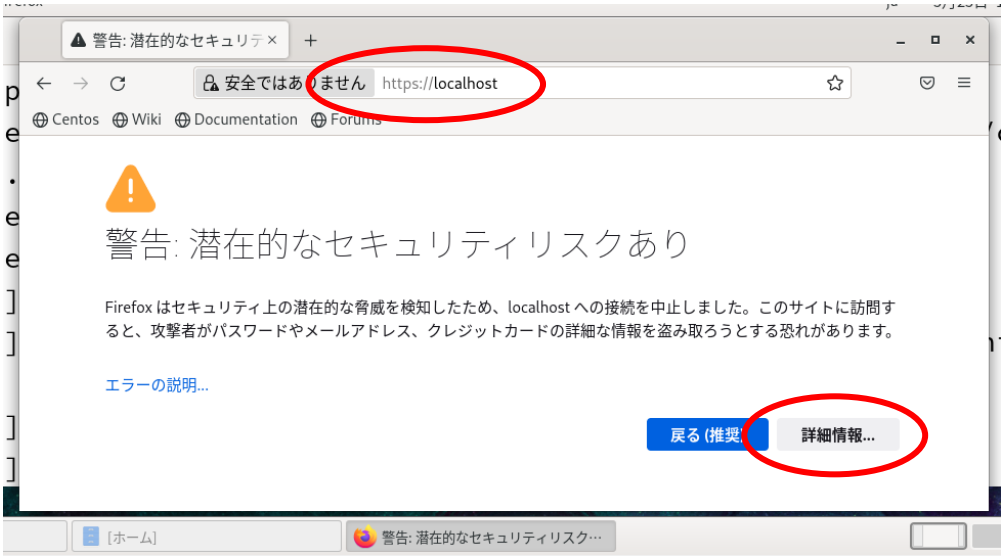
[root@motobe certs]# **firewall-cmd --reload**
success
※FW の再起動

[root@motobe certs]# **firewall-cmd --zone=public --list-services**
dhcpv6-client http **https** ssh
※https の登録確認

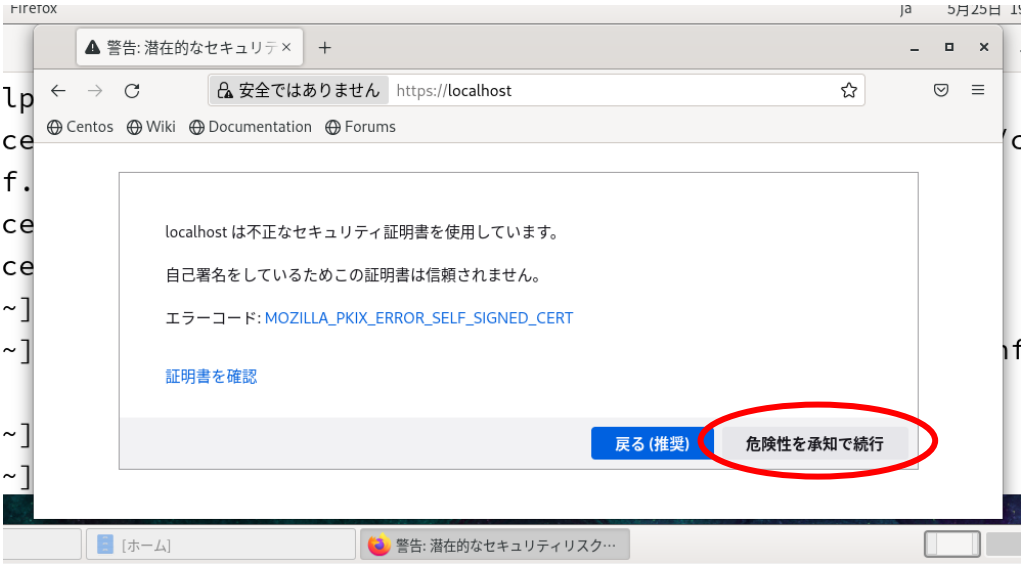
⑤確認

●サーバ側

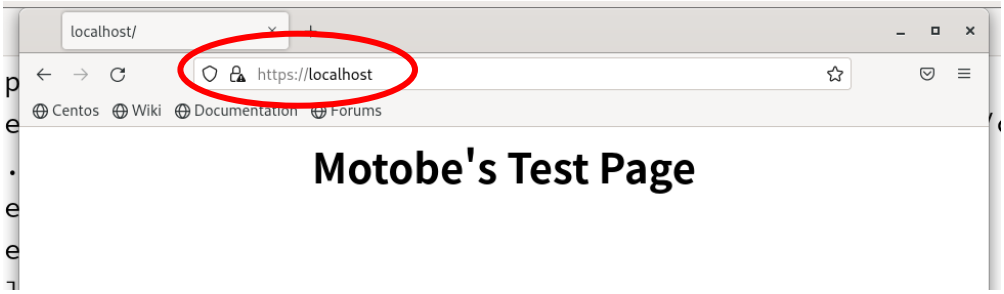
Linux のブラウザ(Firefox)で「**https://localhost(もしくは web サーバの IP アドレス)**」



※「詳細情報」選択



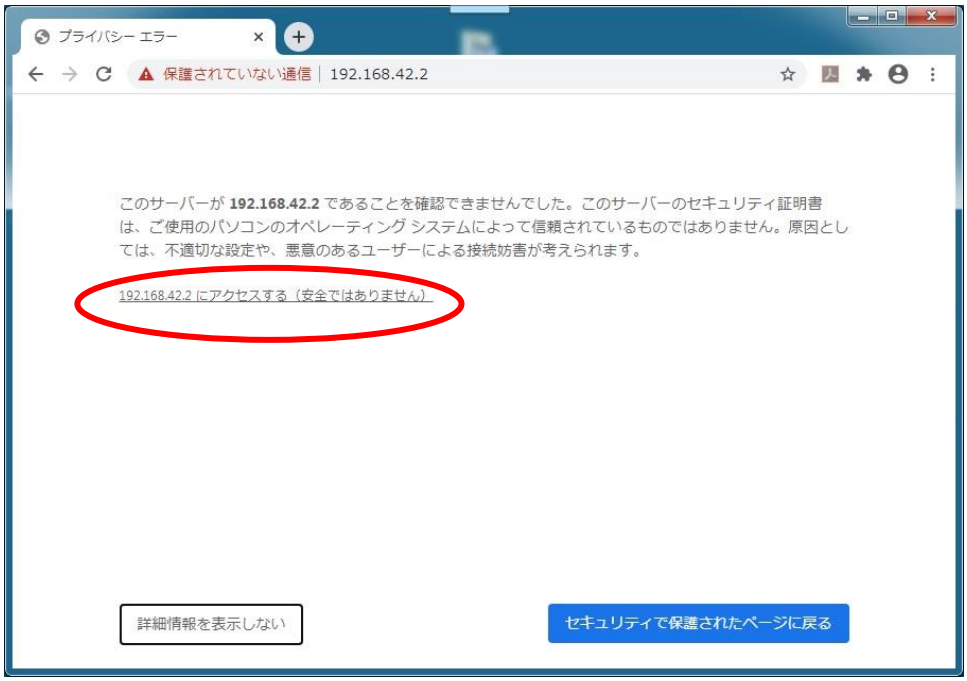
※「危険を承知で続行」選択

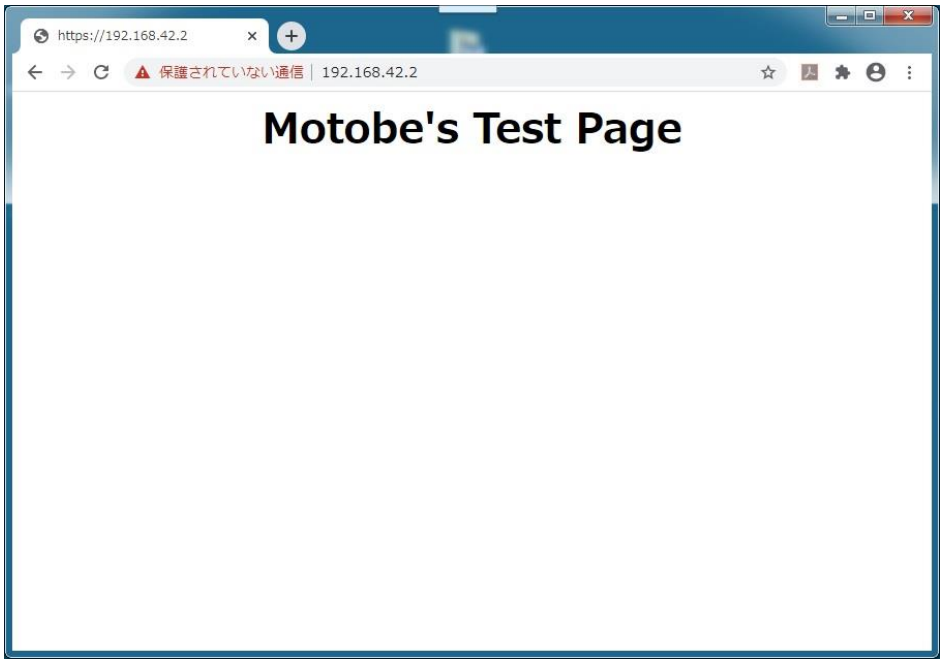


●クライント側(Windows、MAC)

クライント(Window)のブラウザで「**https://web サーバの IP アドレス**」

Web ブラウザでテストページに **https://** でアクセスしてみます。すると独自証明書なので警告が出ます。独自証明書であって、PC にインストールされていないからという理由で警告がでますが、 特に問題ないので「詳細設定」をクリックして続行します。





アクセスできました！

(2)暗号化確認

各自、**WireShark を起動**して、**通信が暗号化(中身が見えない)**されていること確認

①通常通信の確認　　←暗号化なし

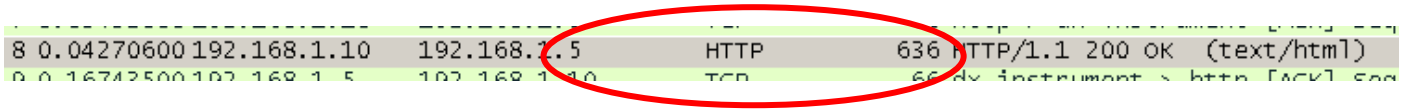
先ずクライント(Window、MAC)のブラウザで「**http://web サーバの IP アドレス**」

・通常通信の中身を確認

プロトコル：http、info：HTTP～(text/html)　　をクリック

パケットの中身(Line－based　text　data：text/html)を確認して下さい

→**中身(html データ)が丸見え**になっています！




```
⊞ Hypertext Transfer Protocol
⊟ Line-based text data: text/html
    <html>\n
    <head>\n
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> \n
    <meta http-equiv="Content-Type" content="text/html; charset=euc-jp"> \n
    <title>\343\203\206\343\202\271\343\203\210</title>\n
    </head>\n
    <body>\n
    [truncated] <p>.htaccess\343\201\253\343\202\210\343\202\213web\343\202\265\
```

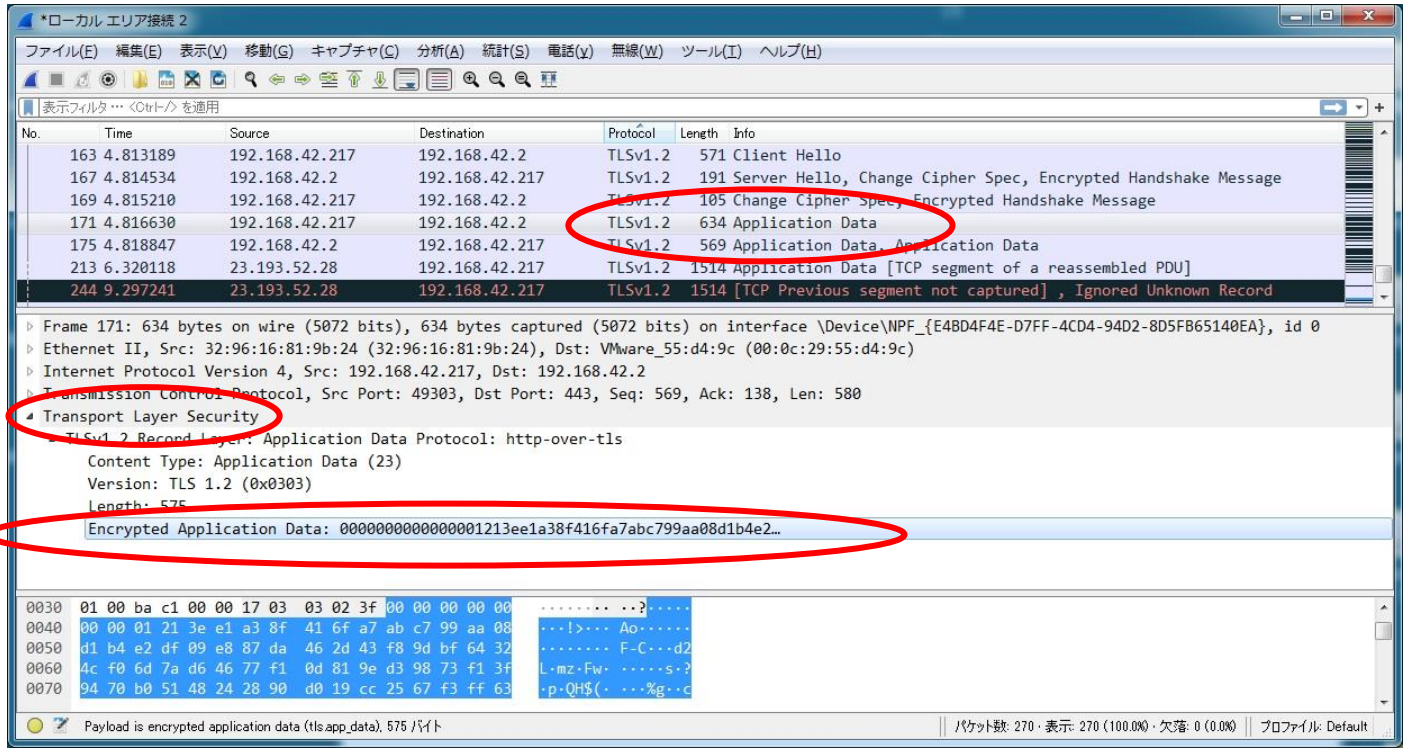
②SSL 通信の確認　←暗号化

クライント(Window)のブラウザで「**https://web サーバの IP アドレス**」

・暗号化通信の中身を確認

プロトコル：TLSV1.2、info：Application Data　　をクリック

パケットの中身(Transport Layer Security)を確認して下さい　←この時点で暗号化が分かる



※encrypted が暗号化している事を表している

SSL は非常に大切な技術ですので、しっかり押さえて下さい

以上！