Apache’de HTTPS Yapma, SSL,TLS Güvensiz Sürümleri Deaktif Yapma, SSL,TLS Güvensiz Şifrelemeleri Deaktif Yapma

(+) Bu uygulama birebir denenmiştir ve başarıyla uygulanmıştır.

Gereksinimler

Ubuntu 18.04 LTS - Nmap SSL-ENUM-CIPHER Script’i (Ana Makine)
Kali 2019.3 VM - Apache // Örnek Bir Apache Makine

a. Apache’de HTTPS Açma

Apache web sunuculardа yapılandırmaya giderek ssl/tls özelliği açılabilir. Bu işlemler önce ssl/tls veri iletimi için public ve private anahtar oluşturma, sonra ssl/tls özelliğinin apache konfigurasyonundan aktifleştirilmesi ve anahtarların apache konfigurasyonda gösterilmesinden oluşur. Adımlar şu şekildedir:

- SSL/TLS Public ve Private anahtarlar oluşturulur.

Kali 2019.3 Terminal:

```
> sudo openssl req -new > new.cert.csr
> sudo openssl rsa -in privkey.pem -out new.cert.key
> sudo openssl x509 -in new.cert.csr -out new.cert.crt -req -signkey new.cert.key -days 3650
> sudo cp new.cert.crt /etc/ssl/certs/server.crt
> sudo cp new.cert.key /etc/ssl/private/server.key
```

- Apache’de SSL/TLS yapılandırması yapılır.

Kali 2019.3 Terminal:

```
> sudo service apache2 start
> sudo a2enmod ssl
> sudo service apache2 restart

> sudo nano /etc/apache2/sites-available/000-default.conf
```

(… İçerikte <VirtualHost *:443>… </VirtualHost> bloğu aşağıdaki gibi olmalıdır …)

```
NameVirtualHost *:443
NameVirtualHost *:80

<VirtualHost *:80>
        …
</VirtualHost>

<VirtualHost *:443>

        ServerName localhost
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
```

CustomLog ${APACHE_LOG_DIR}/access.log combined

 # SSL/TLS Etkinleştirme - EKLENEN BLOK
##
SSLEngine on
SSLOptions +StrictRequire
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
##
 # EKLENEN BLOK SON

…

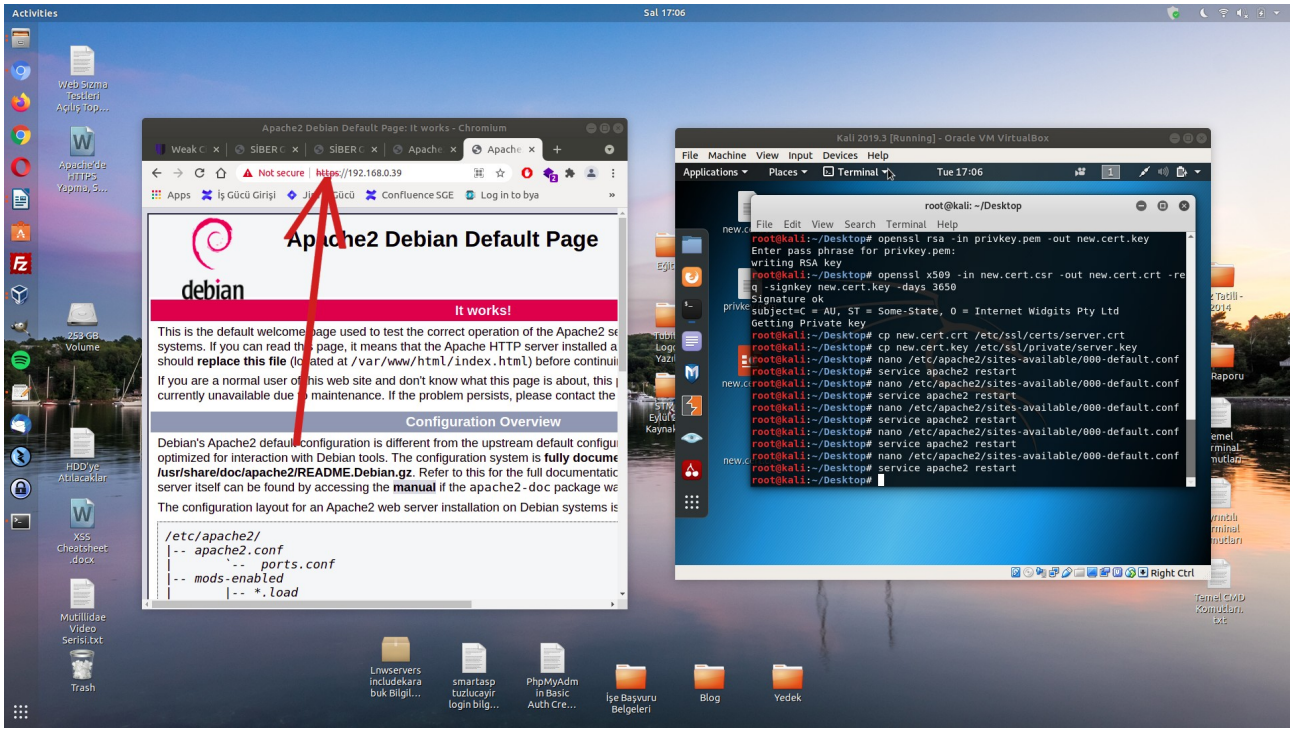
</VirtualHost>

- Apache servisi restart'lanır.

Kali 2019.3 Terminal:

> service apache2 restart

Bu şekilde önce ssl/tls veri iletimi için public ve private anahtar oluşturulur ve sistem ilgili dizinine yerleştirilir. Sonra apache konfigurasyon dosyasından ssl/tls aktifleştirme direktifi girilir ve public, private anahtarların yerleştiği konum gösterilir. Böylece servis yeniden başlatılarak https erişimi açılır.



b. Güvensiz SSL/TLS Protokolleri Deaktif Yapma

Öncelikle https olan apache web sunucuda aktif olan tüm ssl/tls protokollerini görelim.

Ubuntu 18.04 LTS Ana Makine:

> nmap --script ssl-enum-ciphers -p 443 192.168.0.39 // Kali IP'si

Çıktı:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-07-20 17:10 +03
Nmap scan report for 192.168.0.39
Host is up (0.00025s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: client
|   TLSv1.1:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: client
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|
```

```
|     TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|     TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (rsa 2048) - A
|     TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_AES_128_CCM (rsa 2048) - A
|     TLS_RSA_WITH_AES_128_CCM_8 (rsa 2048) - A
|     TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_CCM (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_CCM_8 (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|     TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (rsa 2048) - A
|   compressors:
|     NULL
|   cipher preference: client
|_  least strength: A
MAC Address: 08:00:27:93:B2:2C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.10 seconds
```

Bu kullanılan protokollerden güvensiz olanları web sunucuda kapatalım. Adımlar şu şekildedir:

- Sadece güvenli TLSv1.2 sürümünü etkin bırakmak için;

i) Apache 000-default.conf dosyası açılır.

> sudo nano /etc/apache2/sites-available/000-default.conf

İçerisindeki SSL etkinleştirme bloğuna şu direktif ilave edilir.

```
NameVirtualHost *:443
NameVirtualHost *:80

<VirtualHost *:80>

        …

</VirtualHost>

<VirtualHost *:443>

        ...

        # SSL/TLS Etkinleştirme
        ##############################################
        SSLEngine on
        SSLOptions +StrictRequire
        SSLCertificateFile /etc/ssl/certs/server.crt
        SSLCertificateKeyFile /etc/ssl/private/server.key

        # Sadece SSL/TLS Güvenli Sürümünü Etkin Kılma Direktifi
        SSLProtocol +TLSv1.2
        ##############################################
```

…

</VirtualHost>

ii) Apache servisi yeniden başlatılır.

> service apache2 restart

Sistem restart'laması sonrası değişikliklerin geçerli olup olmadığını görmek için apache web sunucudaki aktif ssl/tls protokollerini tekrar denetleyelim.

Ubuntu 18.04 LTS Ana Makine:

> nmap --script ssl-enum-ciphers -p 443 192.168.0.39 // Kali IP'si

Çıktı:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-07-20 17:20 +03
Nmap scan report for 192.168.0.39
Host is up (0.00030s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CCM (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CCM_8 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CCM (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CCM_8 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
```

```
|      TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (rsa 2048) - A
|    compressors:
|      NULL
|    cipher preference: client
|_  least strength: A
MAC Address: 08:00:27:93:B2:2C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.34 seconds
```

Görüldüğü gibi aktif ssl/tls protokolleri sadece güvenli olan TLSv1.2 şeklinde olmuştur.

c. SSL/TLS Protokollerindeki Güvensiz Şifrelemeleri Deaktif Yapma

Güvensiz ssl/tls protokolleri deaktif edildikten sonra geriye kalan güvenli tls protokolü TLSv1.2 için bu protokolde kullanılan güvensiz şifreleme algoritmaları da deaktif edilmelidir. Böylece tam güvenli bir ssl/tls sertifikasyon yapılandırılmasına gidilmiş olacaktır.

Uyarı :

Apache web yazılımı varsayılan olarak A notlu güçlü şifrelemeleri kullanmakta.

> nmap --script ssl-enum-ciphers -p 443 192.168.0.39 // Kali IP’si

Çıktı:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-07-20 17:20 +03
Nmap scan report for 192.168.0.39
Host is up (0.00030s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (rsa 2048) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CCM (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CCM_8 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
```

```
|     TLS_RSA_WITH_AES_256_CCM (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_CCM_8 (rsa 2048) - A
|     TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|     TLS_RSA_WITH_ARIA_128_GCM_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_ARIA_256_GCM_SHA384 (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (rsa 2048) - A
|   compressors:
|     NULL
|   cipher preference: client
|_  least strength: A
MAC Address: 08:00:27:93:B2:2C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.34 seconds
```

Fakat eski sürümlerinde zayıf şifrelemeler var olabileceğinden bu başlıkta anlatılan adım uygulanabilir.

- Apache sunucuda güvensiz şifreleme algoritmalarını kapama için;

i) Apache 000-default.conf dosyası açılır.

> sudo nano /etc/apache2/sites-available/000-default.conf

İçerisindeki SSL etkinleştirme bloğuna şu direktif ilave edilir.

```
NameVirtualHost *:443
NameVirtualHost *:80

<VirtualHost *:80>

        …

</VirtualHost>

<VirtualHost *:443>

        ...

        # SSL/TLS Etkinleştirme
        ###############################################
        SSLEngine on
        SSLOptions +StrictRequire
        SSLCertificateFile /etc/ssl/certs/server.crt
        SSLCertificateKeyFile /etc/ssl/private/server.key

        # Sadece SSL/TLS Güvenli Sürümünü Etkin Kılma Direktifi
        SSLProtocol +TLSv1.2

        # SSL/TLS Güvenli Sürümündeki Güvensiz Şifreleme Algoritmalarını
        # Kapama Direktifi - Netsparker'ın Önerdiği Önlem Bu Şekildedir.
        SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
        ###############################################

        …
```

</VirtualHost>

ii) Apache servisi yeniden başlatılır.

> service apache2 restart

Sistem restart'laması sonrası değişiklikler geçerli olacaktır. Görüldüğü gibi aktif ssl/tls protokolleri sadece güvenli olan TLSv1.2 şeklinde ve TLSv1.2'nin apache varsayılan ayarlarından kaynaklı kullandığı şifreleme algoritmaları not seviyesi A şeklinde. Eğer güvensiz şifreleme algoritmaları varsa düzenleme sonrası deaktif duruma gelecektir. Bu şekilde tam güvenli bir ssl/tls sertifikasyonu yapılandırması uygulanabilir.

Benim Not:

Apache'de SSL/TLS yapılandırması Ubuntu 18.04 LTS ana makinede de uygulanmıştır ve ana makinede localhost'ta https erişimi açılmıştır. Yani yapılandırma apache genel konfigurasyonudur.

Kaynaklar

https://www.linux.com/training-tutorials/creating-self-signed-ssl-certificates-apache-linux/

https://serverfault.com/questions/920461/why-openssl-ignore-days-for-expiration-date-for-self-signed-certificate

https://www.google.com/search?ei=OgaSX-XGBZGSjLsPg9ycqAU&q=10+years+equal+to+as+day+number&oq=10+years+equal+to+as+day+number&gs_lcp=CgZwc3ktYWIQAzIHCCEQChCgATIHCCEQChCgAToECAAQEzoICAAQFhAeEBM6CAghEBYQHRAeOgUIIRCgAToECCEQFUoFCAgSATFQ3B5Y8Dlg2DpoAXAAeACAAaECiAGhFpIBBjAuMTAuNZgBAKABAaoBB2d3cy13aXrAAQE&sclient=psy-ab&ved=0ahUKEwjly4-8nsnsAhURCWMBHQMuB1UQ4dUDCA0&uact=5

https://askubuntu.com/questions/328162/how-to-create-shortcut-for-a-command-in-terminal

https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/insecure-transportation-security-protocol-supported-sslv3/

https://stackoverflow.com/questions/7580508/getting-chrome-to-accept-self-signed-localhost-certificate/43666288#43666288

https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/weak-ciphers-enabled/