Let’s Encrypt menyediakan SSL/TLS gratis, terotomatisasi, dan terbuka yang diharapkan memberi manfaat bagi publik dari segi keamanan berinternet. Sertifikat SSL ini disediakan oleh Internet Security Research Group (ISRG).

Sebelum menggunakannya berikut beberapa hal mengenai Let's Encrypt yang perlu kita ketahui:

-

Gratis: Setiap orang yang memiliki domain bisa menggunakan Let's Encrypt untuk memperoleh sertifikat SSL yang terpercaya secara gratis.

-

Otomatis: Untuk mendapatkan sertifikat SSL dari Let's Encrypt caranya sangatlah mudah dan cepat dikarenakan proses otomatisasi.

-

Aman: Dengan menggunakan SSL di server kita maka keamanan akan lebih terjamin. Apalagi Let's Encrypt mendukung hampir semua browser yang ada.

-

Transparan: Data dari semua SSL tercatat secara publik di database mereka yang memungkinkan semua orang untuk menelitinya.

-

90 hari: Masa berlaku dari sertifikat SSL gratisan ini hanya 90 hari tapi dikarenakan mendukung otomatisasi sehingga bukan menjadi masalah karena bisa auto-renew.

Instalasi dan Konfigurasi Let's Encrypt

Saya disini menggunakan CentOS (6 & 7) meskipun hampir sama untuk semua OS misal Debian ataupun Ubuntu.

Pertama kita menginstall Git dan mendownload repo dari Let's Encrypt ke server kita.

```
# Install git    yum install git    # Clone repo ke direktori /opt/letsencrypt    git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt    cd /opt/letsencrypt ./letsencrypt-auto --help
```

Buat folder /etc/letsencrypt dengan menjalankan perintah berikut:

```
mkdir /etc/letsencrypt
```

Setelah itu buat file konfigurasi di folder /etc/letsencrypt dengan ekstensi .ini misal domain.ini dan paste teks berikut:

```
# This is an example of the kind of things you can do in a configuration file.  # All flags used by the client can be configured here. Run Let's Encrypt with  # "--help" to learn more about the available options.    # Use a 4096 bit RSA key instead of 2048  rsa-key-size = 4096    # Uncomment and update to register with the specified e-mail address  email = foo@example.com    # Uncomment and update to generate certificates for the specified  # domains.  domains = example.com, www.example.com    # Uncomment to use a text interface instead of ncurses  # text = True    # Uncomment to use the standalone authenticator on port 443  # authenticator = standalone  # standalone-supported-challenges = tls-sni-01    # Uncomment to use the webroot authenticator. Replace webroot-path with the  # path to the public_html / webroot folder being served by your web server.  # authenticator = webroot  # webroot-path = /usr/share/nginx/html
```

Catatan:

- Ganti opsi email dengan email kamu untuk notifikasi ketika sertifikat akan expire.
- Untuk opsi domains isikan semua domain kamu yang ingin dibuatkan SSL-nya. Ingat untuk memisahkan dengan koma untuk tiap domainnya.

Kita tinggal membuat SSL untuk domain yang sudah kita tuliskan diatas dengan menjalankan perintah seperti berikut:

Matikan webservernya service httpd stop # Masuk ke folder /opt/letsencrypt cd /opt/letsencrypt # Generate SSL ./letsencrypt-auto certonly --standalone --agree-tos --renew-by-default --config /etc/letsencrypt/domain.ini # Nyalakan lagi webservernya service httpd start

SSL akan tercipta di folder /etc/letsencrypt/live dan untuk nama menggunakan domain pertama pada opsi domains di file konfigurasinya misal example.com.

Ada 3 buah file berekstensi .pem yang kita perlukan yaitu: cert.pem, chain.pem, dan privkey.pem.

Instalasi SSL ke Webserver

Saya menggunakan webserver Apache dan kita cukup menambahkan SSL yang ada ke virtual host. Contoh:

<virtualhost *:80> ServerName example.com ServerAlias example.com www.example.com Redirect permanent / https://www.example.com/ </VirtualHost> <virtualhost *:443> ServerName example.com ServerAlias example.com www.example.com ServerAdmin dan@example.com DocumentRoot /var/www/html/example.com ErrorLog /var/log/httpd/example.com_error.log CustomLog /var/log/httpd/example.com_access.log combined <Directory "/var/www/html/example.com"> Options FollowSymLinks Indexes AllowOverride All Order Allow,Deny Allow from all DirectoryIndex index.php </Directory> SSLEngine on SSLProtocol ALL -SSLv2 -SSLv3 SSLHonorCipherOrder On SSLCipherSuite ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS SSLCertificateFile /etc/letsencrypt/live/example.com/cert.pem SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem SSLCertificateChainFile /etc/letsencrypt/live/example.com/chain.pem </VirtualHost>

Catatan: Ganti example.com dengan domain SSL kamu.

Instalasi SSL pada Zpanel/Sentora

Jika kalian menggunakan Zpanel/Sentora ada beberapa hal yang perlu disiapkan terlebih dahulu.

Pastikan sudah menginstall mod_ssl:

```
yum install mod_ssl
```

Buka /etc/httpd/conf.d/ssl.conf dan kasih tanda pagar pada Listen 443 menjadi:

```
# # When we also provide SSL we have to listen to the # the HTTPS port in addition. # # Listen 443
```

Masuk ke panel kalian, klik Admin -> klik Module Admin -> Apache Config -> Scroll ke bawah dan pada Override a Virtual Host Setting pilih domain yang akan di konfigurasi.

Selanjutnya pada Port Override isikan dengan 443 dan centang Forward Port 80 to Overriden Port. Isikan konfigurasi berikut pada Custom Entry:.

```
SSLEngine on SSLProtocol ALL -SSLv2 -SSLv3 SSLHonorCipherOrder On SSLCipherSuite ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS SSLCertificateFile /etc/letsencrypt/live/example.com/cert.pem SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem SSLCertificateChainFile /etc/letsencrypt/live/example.com/chain.pem # Keeping bellow for future upgrades. # Requires Apache >= 2.4 # SSLCompression off
```

Catatan: Ganti example.com dengan domain SSL kamu.

Paksa supaya Zpanel/Sentora melakukan perubahan segera:

```
# Zpanel php -q /etc/zpanel/panel/bin/daemon.php # Sentora php -q /etc/sentora/panel/bin/daemon.php # Restart Apache service httpd restart
```

Lakukan ke semua domain yang ingin dipasangi SSL dan kita tidak perlu mengganti nama

SSL-nya.

Auto-renew Let's Encrypt

Masa berlaku dari SSL-nya hanya 90 hari jadi solusi yang paling tepat adalah kita membuat script bash yang dipadukan dengan cron untuk memperpanjang SSL secara otomatis jika sudah akan expire.

Pertama buat script bash-nya:

```
vi /etc/letsencrypt/le-renew
```

Isikan dengan script berikut:

```
#!/bin/bash web_service='httpd' config_file="/etc/letsencrypt/domain.ini" le_path='/opt/letsencrypt' exp_limit=30; if [ ! -f $config_file ]; then echo "[ERROR] config file does not exist: $config_file" exit 1; fi domain=`grep "^s*domains" $config_file | sed "s/^s*domainss*=s*//" | sed 's/(s*)|,.*$//'` cert_file="/etc/letsencrypt/live/$domain/fullchain.pem" if [ ! -f $cert_file ]; then echo "[ERROR] certificate file not found for domain $domain." fi exp=$(date -d "`openssl x509 -in $cert_file -text -noout|grep "Not After"|cut -c 25-`" +%s) datenow=$(date -d "now" +%s) days_exp=$(echo ( $exp - $datenow ) / 86400 |bc) echo "Checking expiration date for $domain..." if [ "$days_exp" -gt "$exp_limit" ] ; then echo "The certificate is up to date, no need for renewal ($days_exp days left)." exit 0; else echo "The certificate for $domain is about to expire soon. Starting webroot renewal script..." echo "Stopping $web_service" service $web_service stop $le_path/letsencrypt-auto certonly --standalone --agree-tos --renew-by-default --config $config_file echo "Starting $web_service" service $web_service start echo "Renewal process finished for domain $domain" exit 0; fi
```

Setelah disimpan pastikan script tersebut bisa dieksekusi:

```
chmod +x /etc/letsencrypt/le-renew
```

Sekarang kita hanya perlu membuat cron yang berjalan satu minggu sekali untuk mengecek apakah SSL sudah mau expire dan jika SSL akan expire kurang dari 30 hari maka otomatis script bash akan me-renew sertifikatnya.

```
# Membuka crontab crontab -e # Isikan baris berikut 30 2 * * 1 /etc/letsencrypt/le-renew > /var/log/le-renew.log 2>&1
```

Untuk mengetestnya kita eksekusi saja script bashnya. Contoh outputnya sebagai berikut:

Checking expiration date for example.com... The certificate is up to date, no need for renewal (89 days left).