# Let's Encrypt SSL Certificate Setup Guide

## 1. Introduction

Let's Encrypt is a free, automated, and open certificate authority (CA) that provides SSL/TLS certificates. It enables HTTPS on websites, aiming to make the internet more secure.

## 2. Installing Certbot

Update system packages:  
sudo apt-get update  
  
Install Certbot:  
sudo apt-get install certbot  
  
Certbot helps automate the process of obtaining and renewing SSL certificates.

## 3. Requesting Certificate - Manual Mode

Use manual mode if you don't want Certbot to automatically modify your server.  
  
Command:  
sudo certbot certonly --manual -v  
  
Follow prompts to create challenge files for domain validation.

## 4. Requesting Certificate - Web Server Plugins (Nginx, Apache)

Certbot can automatically configure your web server if you use Nginx or Apache.  
  
For Nginx:  
sudo apt-get install python3-certbot-nginx  
sudo certbot --nginx  
  
For Apache:  
sudo apt-get install python3-certbot-apache  
sudo certbot --apache  
  
Certbot will automatically edit your configuration files and reload the server.

## 5. Requesting Certificate - DNS Challenge

DNS challenge is useful for wildcard certificates (e.g., \*.example.com).  
  
Command:  
sudo certbot certonly --manual --preferred-challenges=dns -d '\*.example.com' -d example.com  
  
You will be prompted to create specific DNS TXT records to prove domain ownership.

## 6. Requesting Certificate - Standalone Mode

Use standalone mode when no web server is running, or ports 80/443 are free.  
  
Command:  
sudo certbot certonly --standalone -d example.com  
  
Certbot will temporarily start a web server to answer ACME challenges.

## 7. Understanding Generated Files

After successful issuance, Certbot saves certificates here:  
/etc/letsencrypt/live/yourdomain/  
  
- cert.pem: Your domain's SSL certificate  
- chain.pem: Intermediate certificates  
- fullchain.pem: Combined cert + intermediates  
- privkey.pem: Private key  
  
fullchain.pem is most commonly used in server configurations.

## 8. Installing SSL on Servers

Example Nginx configuration:  
  
server {  
 listen 443 ssl;  
 server\_name example.com;  
  
 ssl\_certificate /etc/letsencrypt/live/example.com/fullchain.pem;  
 ssl\_certificate\_key /etc/letsencrypt/live/example.com/privkey.pem;  
}  
  
Reload Nginx:  
sudo systemctl reload nginx  
  
Example Apache configuration:  
SSLEngine on  
SSLCertificateFile /etc/letsencrypt/live/example.com/fullchain.pem  
SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem  
Then reload Apache:  
sudo systemctl reload apache2

## 9. Renewal Process

Certificates are valid for 90 days.  
  
For automated methods (Nginx/Apache/Standalone), use:  
sudo certbot renew  
  
Manual certificates require re-running the manual process.  
  
You can simulate a renewal test with:  
sudo certbot renew --dry-run

## 10. Important Notes

- Always secure privkey.pem.  
- Use strong permissions (chmod 600).  
- Validate SSL using tools like SSL Labs.  
- Monitor expiry dates to avoid certificate expiration.  
- Automate renewals when possible to avoid downtime.  
  
Official Website: https://letsencrypt.org/