Compiling OpenSSL 3.0 with PHP

Why Integrate OpenSSL 3.0 with PHP?

- Access the latest security patches and cryptographic algorithms from OpenSSL 3.x.
- Ensure compatibility with modern security protocols like TLS 1.3.
- Leverage advanced cryptographic functions directly within PHP applications.
- Gain fine-grained control over build configurations for specific environments.
- Address potential security vulnerabilities present in older OpenSSL versions.

Prerequisites and OpenSSL Compilation

- **System Requirements:**
  - Linux or macOS environment.
  - C compiler (e.g., GCC, Clang) and build tools (make).
  - Development libraries (e.g., zlib, perl).
- **Download OpenSSL 3.0 Series:**
  - Obtain the source code from the official OpenSSL website (ssl.com).
- **OpenSSL Compilation Steps (Example):**
  - `./config [options]` (e.g., `./config --prefix=/usr/local/openssl-3.0.0`)
  - `make`
  - `sudo make install` (or install to the specified prefix)
- Note: Compiling OpenSSL separately is often optional if a suitable system-wide version is available, but recommended for specific version control or custom builds.

PHP Compilation: Integrating OpenSSL

- **Download PHP Source:** Obtain the desired PHP version source code.
- **Configure PHP with OpenSSL Options:**
  - `--with-openssl`:
    - Enables the OpenSSL extension for PHP, allowing it to use SSL/TLS capabilities.
  - `--with-openssl-dir=/path/to/openssl/installation`:
    - Specifies the base directory where OpenSSL was installed (headers and libraries). Essential if OpenSSL is not in standard system paths.
  - `--with-ssl=/path/to/openssl/installation`:
    - An older or alternative way to specify the OpenSSL directory, often functionally similar to `--with-openssl-dir`.
  - `--enable-openssl`:
    - Ensures that the OpenSSL extension is compiled into PHP. Often implicitly enabled by `--with-openssl`.
  - `--with-libdir=lib64`:
    - May be required on some 64-bit systems to help the configure script locate OpenSSL

Verification and Final Steps

- **Verify OpenSSL Version in PHP:**
  - Create a `phpinfo()` file and check the 'openssl' section for version details and enabled features.
  - Use the command line: `php -i | grep 'OpenSSL version'`
- **Common Configuration Scenarios:**
  - **System OpenSSL:** `./configure --with-openssl` (if OpenSSL is in standard system paths)
  - **Custom OpenSSL Path:** `./configure --with-openssl --with-openssl-dir=/usr/local/openssl-3.0.0`
- **Considerations for Specific Builds:**
  - The actual OpenSSL library version used by PHP will be determined by the successful configuration and linking.
  - Always ensure you are compiling against a secure and stable OpenSSL 3.0.x release.
  - Regularly update both PHP and OpenSSL for ongoing security.