Building a DNS resolver for FedRAMP

Brian Somers & Prashanth Suvarna,
Cisco Umbrella/OpenDNS

Version 3.0

Agenda

- ➢ Overview
- ➢ OpenSSL and FIPS - Options + nuances
- ➢ Challenges with OpenSSL 3
- ➢ FIPS compliant DNSSEC, DoH & DoT, DNSCrypt
- ➢ Other considerations

Overview

- ➢ OpenDNS / Cisco Umbrella resolvers introduction
- ➢ Why FedRAMP?
  - ❖ Opens opportunity to US Gov and others requiring FedRAMP compliance
- ➢ New infrastructure on AWS separate from commercial/public fleet.
- ➢ Software running on our public fleet:
  - ❖ Written in C
  - ❖ Runs on Debian
  - ❖ Uses OpenSSL (libssl) 1.1.1 for ssl/crypto operations & libsodium for DNSCrypt
- ➢ What is needed for FedRAMP?
  - ❖ Make the resolver FIPS compliant

OpenSSL and FIPS – Options

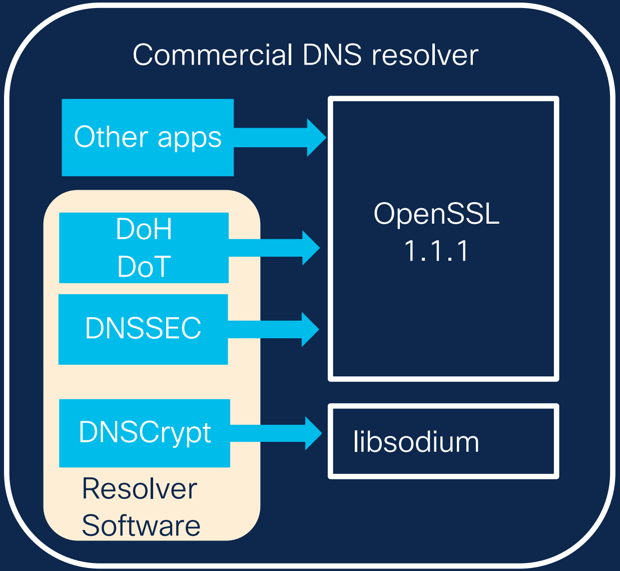
FIPS Module 2.0	FIPS module 3.0
➢ FIPS Certified	➢ FIPS certified
➢ Compatible with OpenSSL **1.0.1** & 1.0.2	➢ Compatible with OpenSSL **3.0** and **3.1**
➢ Not compatible with OpenSSL 1.1.1	
➢ OpenSSL not releasing updates	➢ OpenSSL actively maintains it

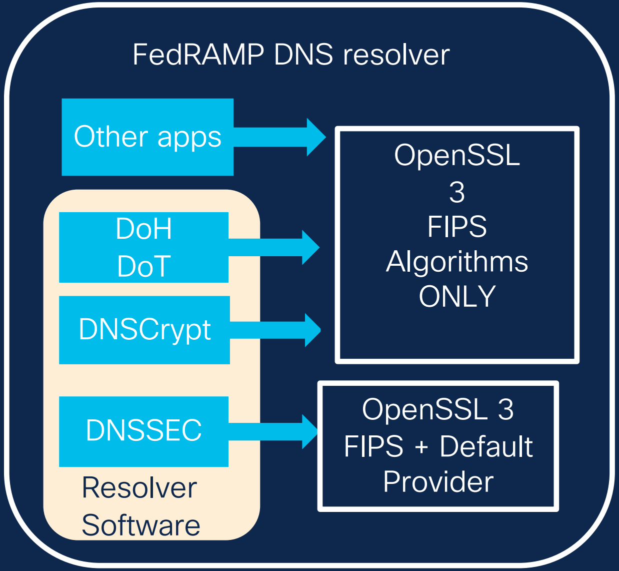
OpenSSL 3 nuances

- OpenSSL Providers concept
- Default Provider & FIPS Provider
- Several low-level APIs deprecated
- Strong push to use high-level EVP (i.e., Digital EnVeloPe) APIs

Challenges with OpenSSL 3

- ➢ OpenSSL 3 not available for Debian 10/11
- ➢ Build it with FIPS support and package it ourselves
- ➢ Porting of low-level APIs to higher-level EVP APIs
- ➢ Supporting both OpenSSL 1.1.1 and OpenSSL 3 from same codebase





DNSSEC Algorithms

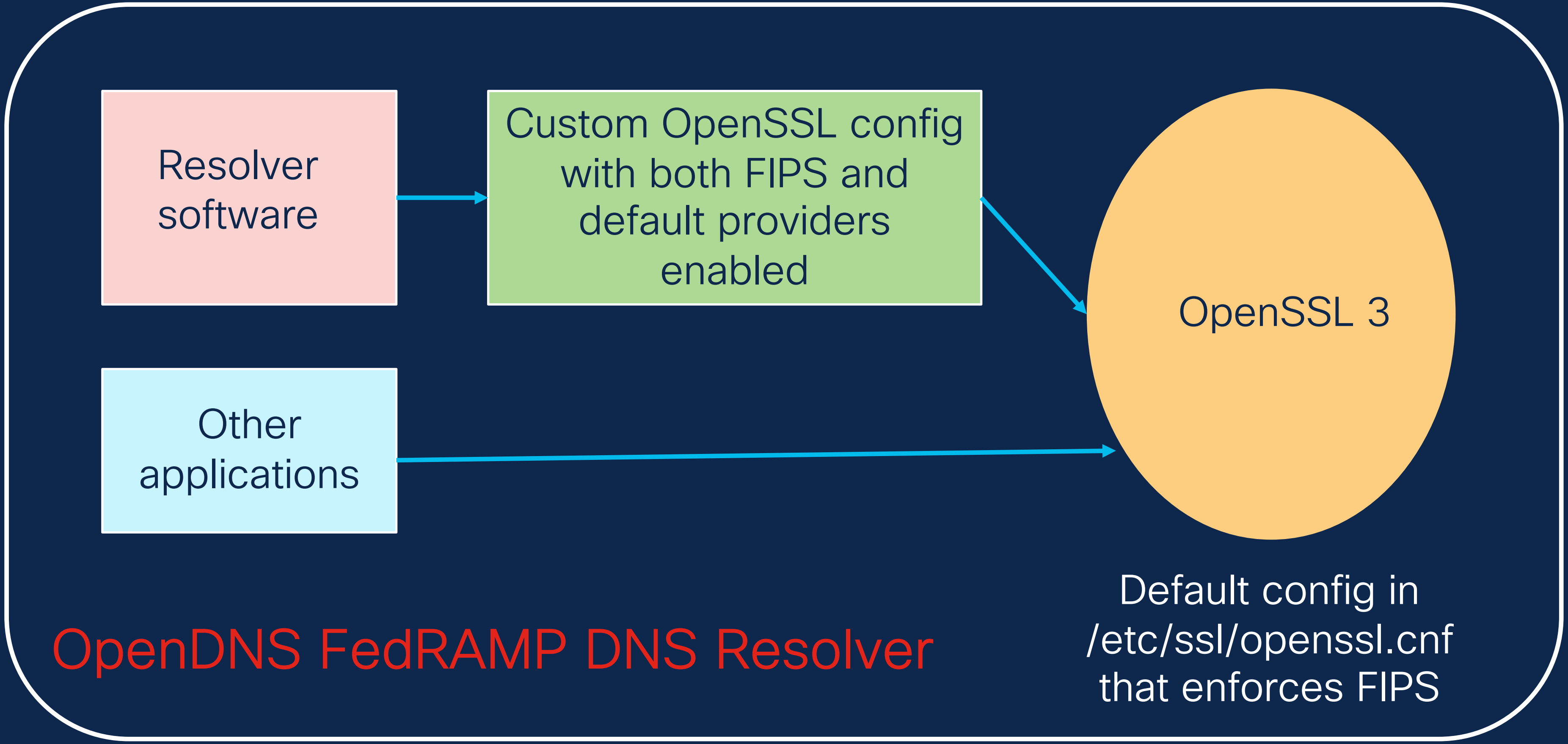
FIPS 👍	FIPS 👎
➢ RSA/SHA-1	➢ Ed25519
➢ RSASHA1-NSEC3-SHA1	➢ Ed448
➢ RSA/SHA-256	
➢ RSA/SHA-512	
➢ ECDSA Curve P-256 with SHA-256	
➢ ECDSA Curve P-384 with SHA-384	

* List of algorithms supported by OpenDNS/Umbrella resolvers as of 31-August-2023

FIPS Compliant DNSSEC

- ED25519 and ED448 are not yet FIPS compliant
- How to comply without compromising security?
- Enforce validation failures for above but treat successful validations as unsigned
- How to validate non-FIPS compliant algorithms on a FIPS system?

OpenSSL 3 interactions



Resolver software

Custom OpenSSL config with both FIPS and default providers enabled

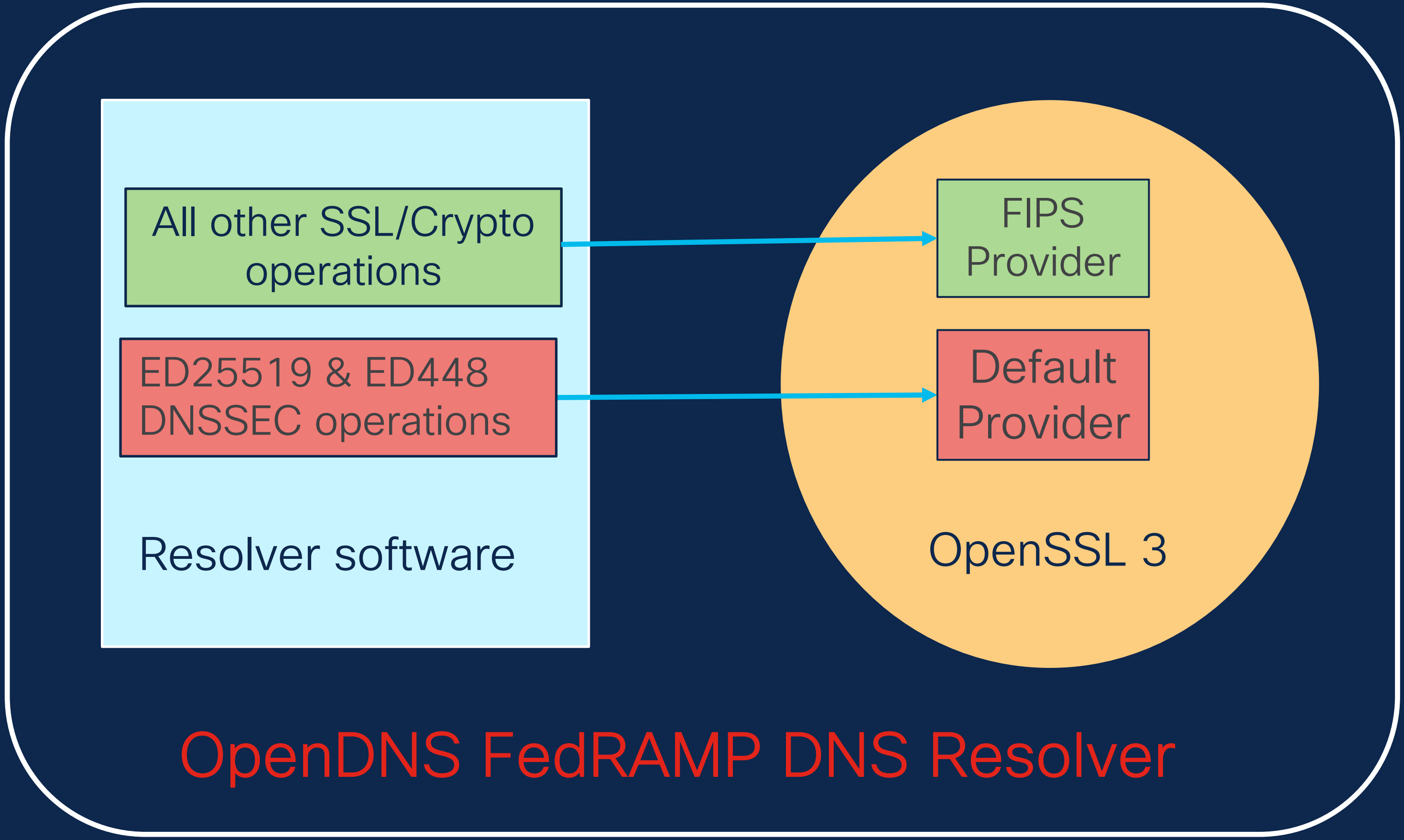
OpenSSL 3

Other applications

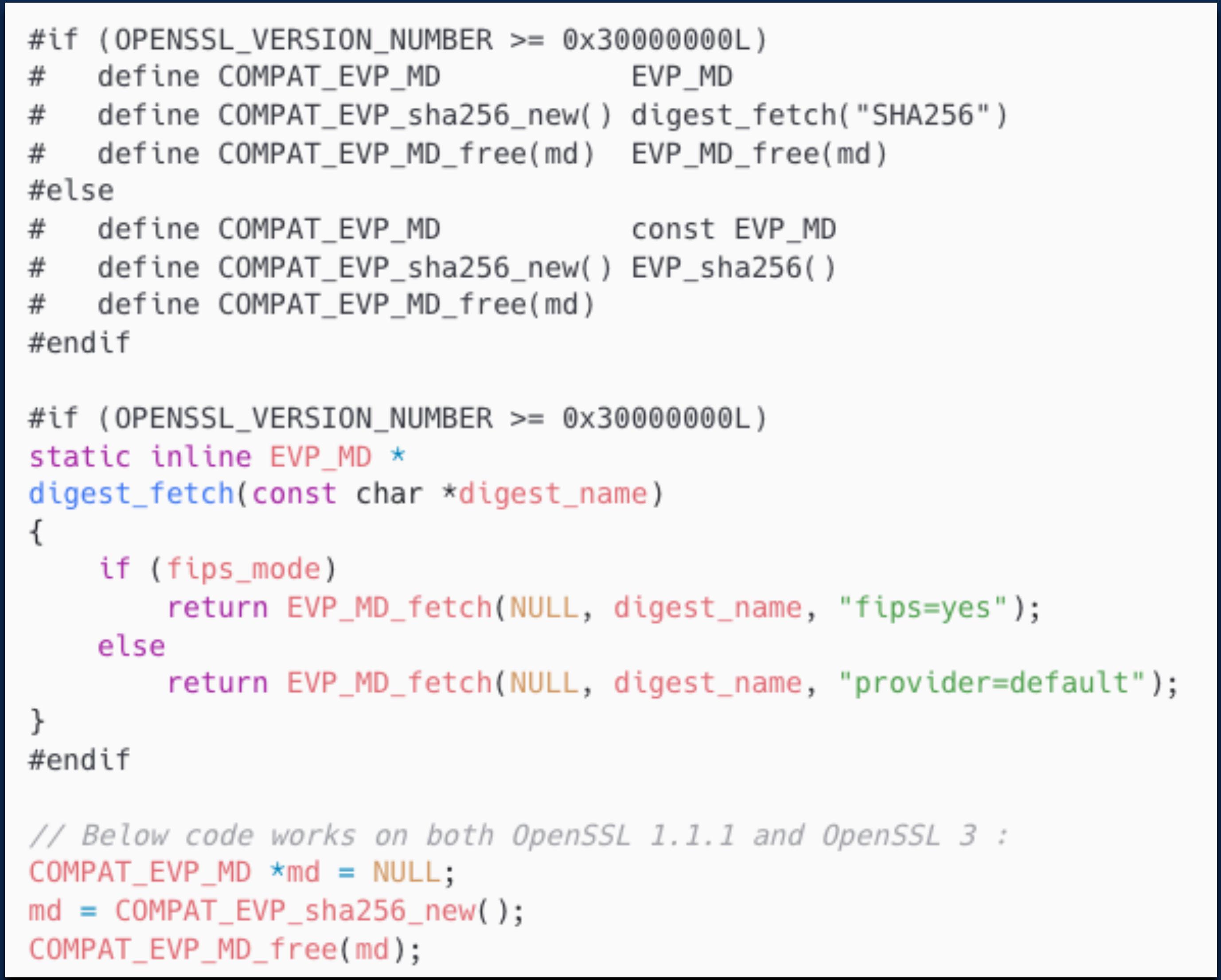
Default config in /etc/ssl/openssl.cnf that enforces FIPS

OpenDNS FedRAMP DNS Resolver

Resolver interactions – Deeper look


All other SSL/Crypto operations
ED25519 & ED448 DNSSEC operations
Resolver software
FIPS Provider
Default Provider
OpenSSL 3
OpenDNS FedRAMP DNS Resolver

Example: Fetching a digest implementation

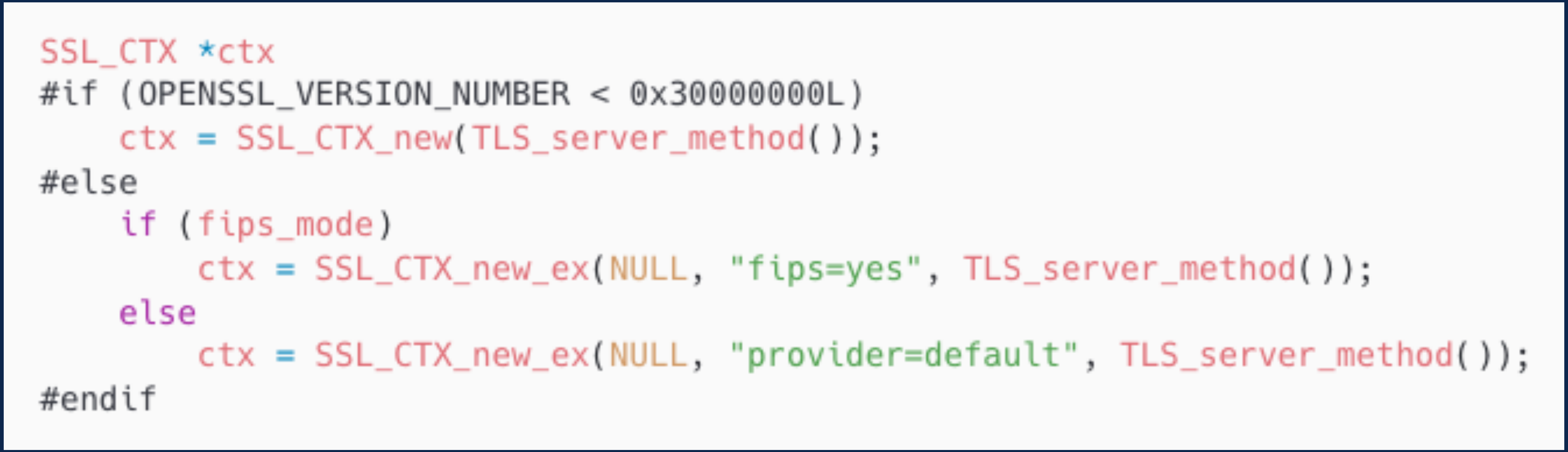


```
#if (OPENSSL_VERSION_NUMBER >= 0x30000000L)
#   define COMPAT_EVP_MD              EVP_MD
#   define COMPAT_EVP_sha256_new() digest_fetch("SHA256")
#   define COMPAT_EVP_MD_free(md)  EVP_MD_free(md)
#else
#   define COMPAT_EVP_MD              const EVP_MD
#   define COMPAT_EVP_sha256_new() EVP_sha256()
#   define COMPAT_EVP_MD_free(md)
#endif

#if (OPENSSL_VERSION_NUMBER >= 0x30000000L)
static inline EVP_MD *
digest_fetch(const char *digest_name)
{
    if (fips_mode)
        return EVP_MD_fetch(NULL, digest_name, "fips=yes");
    else
        return EVP_MD_fetch(NULL, digest_name, "provider=default");
}
#endif

// Below code works on both OpenSSL 1.1.1 and OpenSSL 3 :
COMPAT_EVP_MD *md = NULL;
md = COMPAT_EVP_sha256_new();
COMPAT_EVP_MD_free(md);
```

FIPS compliant DoH and DoT

- FIPS module restricts the supported TLS Ciphers for DoH and DoT
- TLS_CHACHA20_POLY1305_SHA256 and ECDHE-RSA-CHACHA20-POLY1305 unsupported

Example 2: Creating an SSL_CTX object



```
SSL_CTX *ctx
#if (OPENSSL_VERSION_NUMBER < 0x30000000L)
    ctx = SSL_CTX_new(TLS_server_method());
#else
    if (fips_mode)
        ctx = SSL_CTX_new_ex(NULL, "fips=yes", TLS_server_method());
    else
        ctx = SSL_CTX_new_ex(NULL, "provider=default", TLS_server_method());
#endif
```

FIPS compliant DNSCrypt

- RFC work in progress
- ES3 is the new FIPS compliant ECDHE-ECDSA-AES128-GCM-SHA256 encryption system.
- ES1 & ES2 won't be supported in FIPS mode

Other considerations...

- Provide both OpenDNS/Umbrella protection & support Protective DNS requirements for government agencies
- Other tooling/environment changes
- Security and compliance

References

- OpenSSL 3.0 Design:

  https://www.openssl.org/docs/OpenSSL300Design.html
- OpenSSL 3.0 FIPS Certification:

  https://www.openssl.org/blog/blog/2022/08/24/FIPS-validation-certificate-issued/
- OpenSSL FIPS provider compliance related information:

  https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp4282.pdf
- FIPS certified algorithms:

  https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf
- DNSCrypt RFC (draft)

  https://datatracker.ietf.org/doc/draft-denis-dprive-dnscrypt/

Closing comments

Thank you!