SSL 패킷 분석을 통한 HTTPS 통신 흐름 이해

요약

Wire shark 를 통해 수집한 HTTPS 패킷을 분석

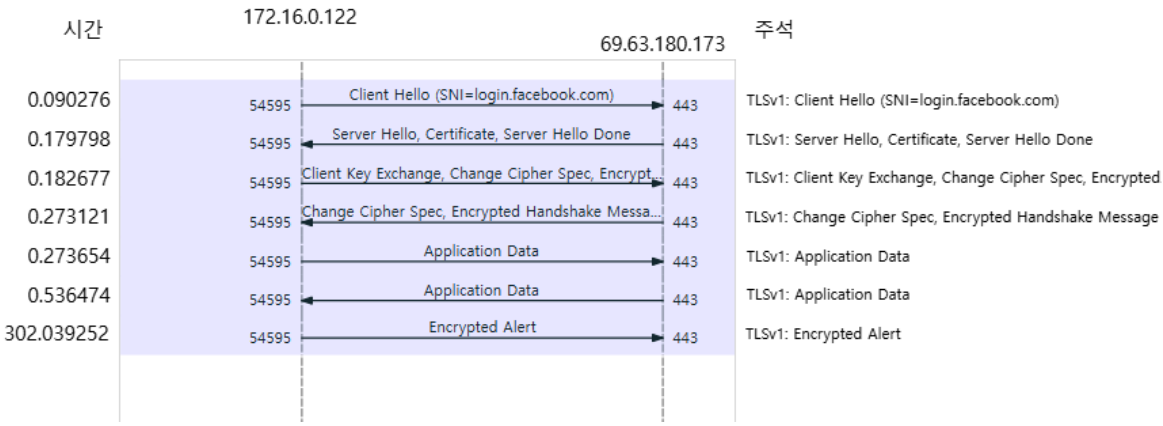
26AI 조영규

목차

1. SSL 패킷 필드 분석

- 각 패킷의 필드 구성 및 패킷의 역할에 대해 분석한다.

1.0.1 TLS 패킷의 Flow Graph



- 기본적인 흐름 설명

1. Client Hello 패킷을 통해 TLS Handshake 요청을 서버에게 보낸다.
2. 서버는 Server Hello 패킷을 클라이언트에게 전송하여 보안 설정을 확정한다.
3. 서버가 Certificate 패킷을 통해 자신의 인증서를 클라이언트에게 전달한다.
4. Server Key Exchange 패킷을 통해 서버의 임시 공개키를 전달하는데, RSA 방식에서는 생략이 가능하다.
5. Certificate Request 패킷을 전송하여 클라이언트의 인증서를 요구할 수 있다. 마찬가지로 생략 가능하다.
6. Server Hello Done 패킷은 서버가 Handshake Message 를 모두 보냈음을 알린다.
7. Certificate Request 가 생략되어 클라이언트의 Certificate 패킷은 생략된다.
8. Client Key Exchange 패킷은 클라이언트가 Pre-Master Secret 을 서버에게 전달. Pre-Master Secret 는 세션 키 생성에 사용한다.
9. Certificate Verify 패킷도 Certificate 패킷과 마찬가지로 생략된다.
10. Change Cipher Spec 패킷을 통해 클라이언트가 암호화 통신의 시작을 알린다. 이후의 메시지는 암호화된 상태로 전송된다.

11. Finished 패킷으로 클라이언트가 SSL/TLS Handshake 의 마지막 단계로 정상적으로 완료되었음을 서버에게 알리는 암호화된 메시지이다. 무결성을 검증하는 메시지를 암호화해 전송한다.

12. 서버도 Change Cipher Spec 패킷으로 암호화 패킷을 전송한다.
13. 서버 Finished 도 클라이언트와 동일한 메시지를 암호화해 응답한다.
14. 이후 Application Data 패킷으로 실제 HTTP 요청/응답이 암호화되어 전송됨
15. 마지막으로 서버 또는 클라이언트가 Encrypted Alert 패킷으로 세션 종료나 오류 메시지를 전송한다.

1.1.2 4 번 프레임 Client Hello

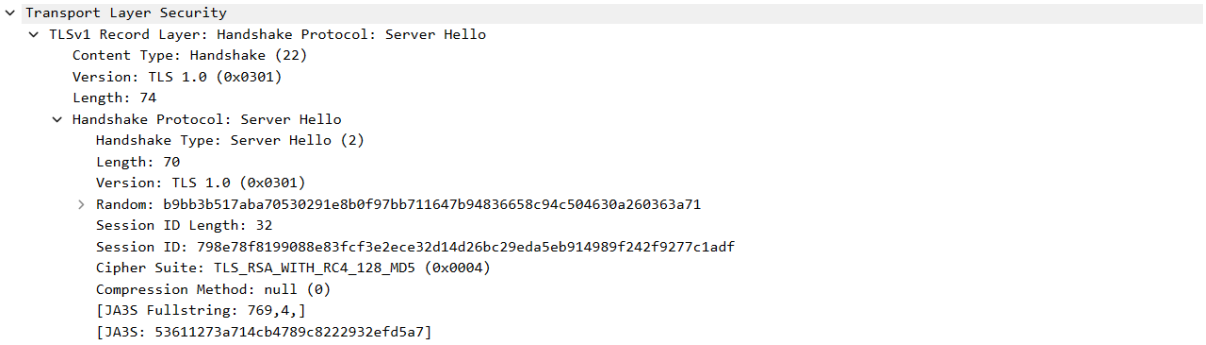
1.1.3 패킷 설명

- 이 패킷은 클라이언트가 서버에게 보안 통신을 사용하겠다고 알리는 메시지로, SSL/TLS Handshake 의 시작점이 되는 패킷이다.
- 서버에게 연결을 요청하는 패킷이다.

1.1.4 패킷 필드 분석

- Content Type: Handshake (22). 연결 요청임을 나타낸다.
- Version: TLS 1.0, TLS 1.0 버전 명시
- Length: 아래 Handshake 메시지 전체 길이 (164)
- Handshake Protocol Header
- Handshake Type: Client Hello (1)
- Length: Handshake 메시지 전체 바이트 수 (3바이트). 여기에 포함되는 건 Client Hello 메시지의 나머지 모든 필드들 전체 크기이다. (예: Cipher Suites, Compression Methods, Extensions 등 포함)
- Client Version: 클라이언트가 사용할 수 있는 TLS 버전이 1.0
- 랜덤 값: 32바이트 랜덤 값으로, 세션 키 생성 시 사용된다. 처음 4바이트는 유닉스 timestamp, 나머지는 랜덤 바이트
- 세션 ID: 이전 세션 재사용을 위한 ID 또는 빈 값
- Cipher Suites: 클라이언트가 지원하는 암호화 알고리즘 리스트로, 서버가 이 중에서 하나를 선택한다. (예시: TLS_RSA_WITH_AES_128_CBC_SHA)
- Compression Methods: 클라이언트가 지원하는 압축 방식으로, 대부분의 경우 00(null, 압축 안 함)만 포함된다. 오래된 SSL에는 압축이 존재하지만, TLS 대부분은 압축을 하지 않는다.
- Extensions: 여러 부가 설정 포함. SNI, ALPN 등

1.2.1 5 번 프레임 Server Hello



1.2.2 패킷 설명

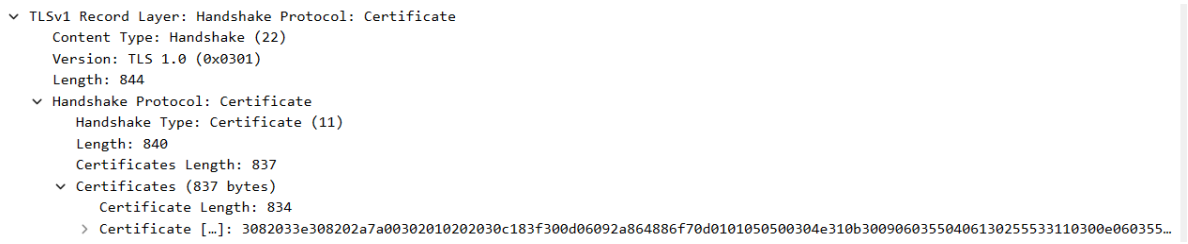
- 서버가 클라이언트 요청에 응답하는 메시지이다.

- 이 패킷을 통해 서버는 어떤 암호화 방식(예: RSA)을 사용할지 결정한다.

1.2.3 패킷 필드 분석

- TLS에서 성능을 고려해 Server Hello부터 Server Hello Done까지 붙여서 전송
- Record Layer Header
- Content Type: Handshake (22)
- Version: TLS 1.0
- Length: 74
- Handshake Protocol: Server Hello
- Handshake Type: Server Hello (2)
- Length: 메시지 본문 길이 (70)
- Version: TLS 1.0
- Random: 서버가 생성한 32바이트 랜덤 값. 클라이언트 랜덤 값과 함께 세션 키 생성에 활용
- Session ID: 서버가 생성한 32바이트 랜덤 값. 클라이언트의 세션ID를 그대로 수락하거나 새로 생성. 이 패킷의 경우 새로 생성
- Cipher Suites: TLS_RSA_WITH_RC4_128_MD5 선택
- Compression Methods: 0 (압축 안함)

1.3.1 5 번 프레임 Certificate



1.3.2 패킷 설명

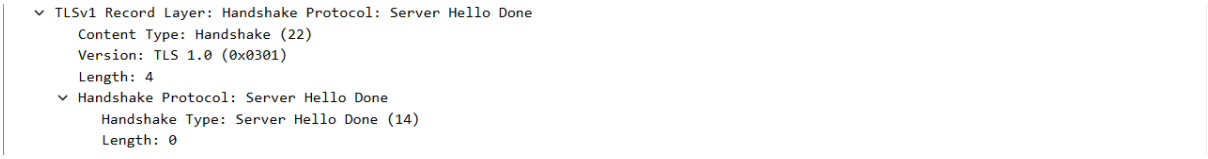
- 서버가 자신을 증명하기 위해 보내는 인증서 정보가 담긴 패킷이다.

- 서버의 공개키와 서버의 신원 정보, 서명 등이 들어있다.

1.3.3 패킷 필드 분석

- Content Type: Handshake (22)
- Version: TLS 1.0
- Length: 844
- Handshake Protocol: Certificate
- Handshake Type: Certificate (11)
- Length: 840
- Certificates Length: 837
- Certificates: 서버 인증서 부분

1.4.1 5 번 프레임 Server Hello done



1.4.2 패킷 설명

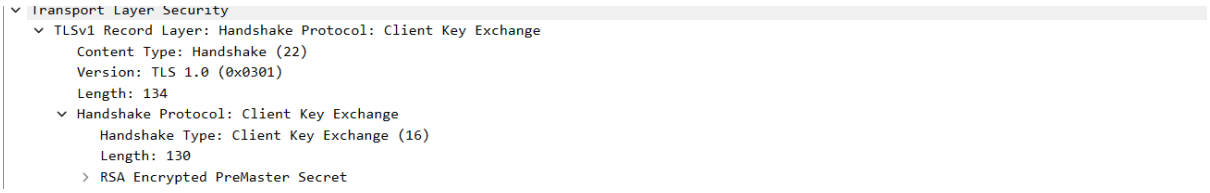
- 서버 측의 초기 설정 완료를 클라이언트에게 알린다.

- Server Key Exchange가 생략된 것을 확인할 수 있는데, 이는 서버 인증서가 RSA 키를 포함하고 있고, 키 교환도 RSA 기반으로 진행되어 클라이언트는 이 공개키로 Pre-master Secret를 암호화해서 보내면 되기 때문에 생략할 수 있다. .

1.4.3 패킷 필드 분석

- Content Type: Handshake (22)
- Version: TLS 1.0
- Length: 4
- Handshake Protocol: Server Hello Done
- Handshake Type: Server Hello Done (14)
- Length: 0

1.5.1 7 번 프레임 Client Key Exchange



1.5.2 패킷 설명

- 암호화 통신을 위한 Pre-Master Secret 을 서버에 전달한다.

- 이 Pre-Master Secret 이 세션 키 생성에 사용된다.

- Master Secret = Pre-Master Secret + Client Random + Server Random

- 이 Master Secret 을 기반으로 세션 키들을 생성한다.

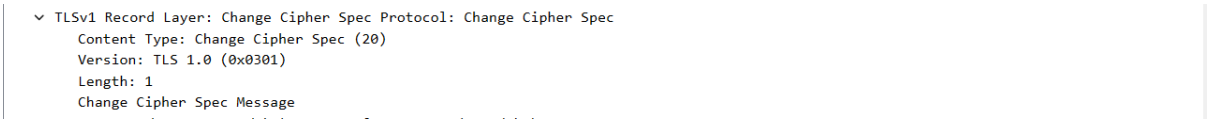
- 서버에서 클라이언트 Certificate Request 가 없었기 때문에 클라이언트 Certificate 패킷은 생략되었다.

- Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message 패킷이 7번 프레임에 통합되어 있다.

1.5.3 패킷 필드 분석

- Content Type: Handshake (22)
- Version: TLS 1.0
- Length: 134
- Handshake Protocol: Client Key Exchange
- Handshake Type: Client Key Exchange (16)
- Length: 130
- RSA Encrypted Pre-Master Secret: 이 암호화된 Pre-Master Secret은 서버의 개인키로 복호화 되어 이후 클라이언트와 서버가 동일한 세션 키를 생성하는 데 사용된다
- Encrypted Pre-Master Length: 128
- Encrypted Pre-Master [...]: 서버의 공개키로 암호화 한 값

1.6.1 7 번 프레임 Change Cipher Spec



1.6.2 패킷 설명

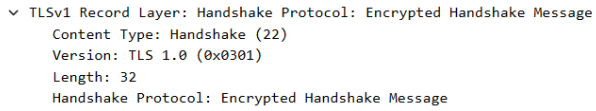
- 암호화된 통신으로의 전환을 알린다. (클라이언트가 서버에게)

- 세션 키가 설정된 다음에 이 패킷을 주고받는다.

1.6.3 패킷 필드 분석

- Content Type: Cipher Spec (20)
- Version: TLS 1.0
- Length: 1
- Change Cipher Spec Message: 암호화 전환의 신호 역할을 한다. 이 메시지를 통해 클라이언트 또는 서버는 지금부터 암호화 통신을 시작하겠다는 의사를 알린다.

1.7.1 7 번 프레임 Encrypted Handshake Message



1.7.2 패킷 설명

- 처음으로 암호화된 메시지를 전송하는 패킷이다 (클라이언트가 서버에게).

- Finished 메시지를 암호화한 형태로 Handshake 가 성공적으로 완료됨을 알린다.

1.7.3 패킷 필드 분석

- Content Type: Handshake (22)
- Version: TLS 1.0
- Length: 32
- Encrypted Handshake Message: 암호화된 메시지 (Finished)

1.8.1 8 번 프레임 Change Cipher Spec

```
Transport Layer Security
  TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.0 (0x0301)
      Length: 1
      Change Cipher Spec Message
```


1.8.2 패킷 설명

- 서버가 클라이언트에게 Change Cipher Spec 패킷을 전송하여 서버 측에서도 암호화 전환을 완료했음을 알린다.

1.8.3 패킷 필드 분석

- Content Type: Cipher Spec (20)
- Version: TLS 1.0
- Length: 1
- Change Cipher Spec Message: 암호화 전환

1.9.1 8 번 프레임 Encrypted Handshake Message

```
TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 32
    Handshake Protocol: Encrypted Handshake Message
```

1.9.2 패킷 설명

- 클라이언트와 서버가 암호화된 상태로 서로의 무결성을 검증하는 데 사용한다.

- Finished 메시지로, TLS Handshake 의 마지막 단계까지 성공적으로 검증하고 보안 연결을 성립했다는 것을 의미한다.

Encrypted Handshake Message. (서버가 클라이언트에게 전송)

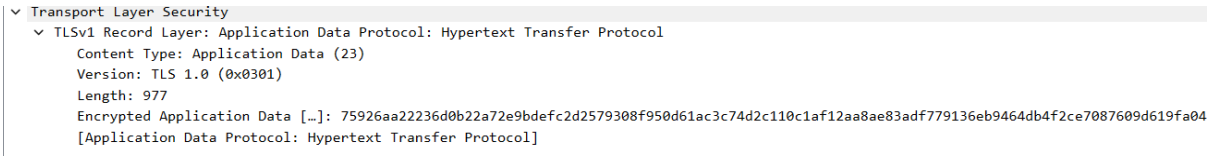
Content Type: Handshake (22)

Version: TLS 1.0

Length: 32

Encrypted Handshake Message: 암호화된 메시지 (Finished)

1.10.1 9 번 프레임: Application Data



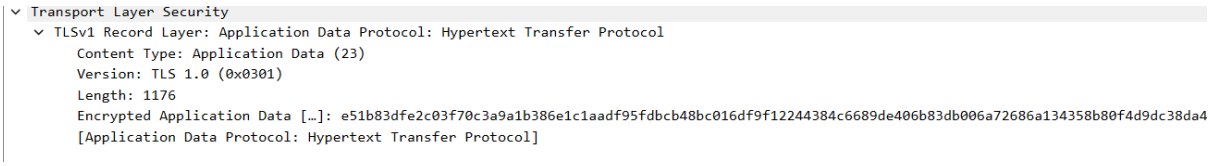
1.10.2 패킷 설명

- Hypertext Transfer Protocol: Handshake 연결 이후 클라이언트가 서버에게 암호화된 HTTP 요청을 보내는 패킷이다.

1.10.3 패킷 필드 분석

- Content Type: Application Data (23)
- Version: 1.0
- Length: 977
- Encrypted Application Data: 암호화된 HTTP 요청

1.11.1 11 번 프레임

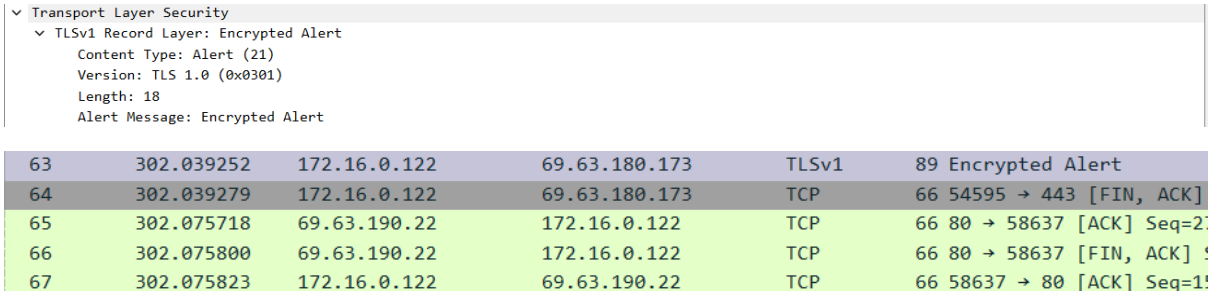


1.11.2 패킷 설명

- Hypertext Transfer Protocol: Handshake 연결 이후 서버가 클라이언트에게 암호화된 HTTP 응답을 보내는 패킷이다.

- Content Type: Application Data (23)
- Version: 1.0
- Length: 1176
- Encrypted Application Data: 암호화된 HTTP 응답

1.12.1 63 번 프레임



63	302.039252	172.16.0.122	69.63.180.173	TLSv1	89 Encrypted Alert
64	302.039279	172.16.0.122	69.63.180.173	TCP	66 54595 → 443 [FIN, ACK]
65	302.075718	69.63.190.22	172.16.0.122	TCP	66 80 → 58637 [ACK] Seq=2
66	302.075800	69.63.190.22	172.16.0.122	TCP	66 80 → 58637 [FIN, ACK]
67	302.075823	172.16.0.122	69.63.190.22	TCP	66 58637 → 80 [ACK] Seq=1

1.12.2 패킷 설명

- Encrypted Alert: 암호화된 경고 메시지를 서버에게 전달하는 패킷으로, 세션 종료나 오류 상황을 알린다.

- 63번 패킷 이후로 TCP 세션 종료 과정을 확인할 수 있다.
- Content Type: Alert (21)
- Version: 1.0
- Length: 18
- Alert Message Encrypted Alert: 암호화되어 내용은 확인 불가능한 메시지