[네트워크 포렌식 기초] HTTP와 HTTPS 패킷 비교, 그리고 로그인 정보 추출

작성자: ssseo

1. 들어가며

디지털 포렌식에서 네트워크 분석은 필수입니다.
네트워크를 타고 흐르는 패킷에는 사용자 행위의 흔적이 그대로 남기 때문입니다.

하지만 프로토콜에 대한 제대로 된 이해가 선행되지 않은 채

실습부터 하게 된다면 막히는 부분이 많이 생길 수밖에 없습니다.

예를 들자면, 스푸핑이나 디도스 공격 실습 과정 중에 HTTP와 HTTPS의 차이를 잘 몰라 두 프로토콜을 혼돈하는 경우가 있을 수 있습니다. (사실 제 경험이기는 한데...)

이번 글에서는 HTTP와 HTTPS 프로토콜에 대해 알아보고, Wireshark를 활용해 HTTP와 HTTPS 패킷을 직접 비교하고, 로그인 정보가 어떻게 노출되거나 보호되는지 확인해보겠습니다.

2. HTTP, 왜 알아야 하나?

사실, "요즘 거의 다 HTTPS를 사용하는데, 꼭 HTTP에 대해 고려를 해야 하는 건가?" 라는 생각을 가지고 있는 분들도 많을 것입니다. 아니면 아직 HTTP와 HTTPS를 구분하지 못하는 분들도 계실 것 같습니다. 하지만 여전히 HTTP 때문에 일어나는 보안 상의 문제들이 존재하기 때문에, 한번 짚고 넘어갈 가치가 충분하다고 생각합니다.

아직 HTTP를 사용하는 환경들을 살펴보자면...

1) 내부 네트워크 환경

회사나 학교 내부에서만 쓰는 간단한 웹 애플리케이션, Iot 기기 설정 페이지 같은 경우는 HTTPS 인증서 발급/관리가 귀찮으니까 그냥 HTTP로 서비스하는 경우가 많습니다.

Ex) http://192.168.x.x 로 들어가는 공유기 설정창

2) 구형 시스템/레거시 서비스

HTTPS가 발표되기 전에 만들어져, 아직까지 업데이트가 안된 웹 서비스들이 HTTP를 유지하고 있는 경우가 있습니다. 특히! 정부나 공공기관의 옛날 자료실, 오래된 학교 사이트 등이 이 경우에 해당합니다.

앞 줄의 내용을 읽고, 정부나 공공기관의 보안이 이렇게 취약할 리가 없다고 생각하실

수 있지만 2020년에 공개된 기사를 살펴보면,

> 김영배 더불어민주당 의원이 이같은 문제를 지적했습니다. 당시 김 의원은 공공기관 웹사이트 1천280개 중 583개는 HTTPS가 아닌 HTTP로 연결되는 사이트라며, 보안 위협이 있을 수 있다고 지적했죠.

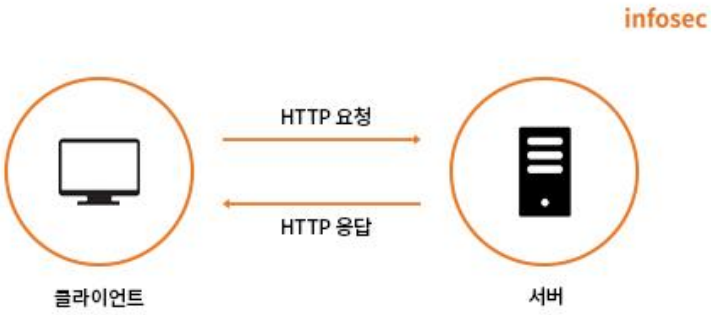
(출처: 김윤희, "HTTP 웹사이트 방문자, 이렇게 계정 털린다", ZDNet Korea, 2020)

2020년까지 절반에 가까운 공공기관 웹사이트 들이 HTTP를 여전히 사용하고 있었음을 알 수 있습니다. 더 최근으로 와보자면 2023년 한국 병원 웹사이트 중 약 25%가 평문 HTTP를 사용했다는 연구 결과 (https://arxiv.org/abs/2304.13278?utm_source=chatgpt.com)

가 있습니다. 생각보다 아직까지 많은 곳에서 HTTP가 사용되고 있다는 사실에 저는 꽤나 놀랐습니다.

그 외에도 공공 와이파이 이용과정에서도 HTTP가 이용되는 점 등이 HTTP 프로토콜에 대해서 우리가 아직 방심하면 안 된다는 것을 보여줍니다.

3. HTTP



HTTP는 서버/클라이언트 모델을 따르며 요청(Request)와 응답(Response) 형태로 구성되어 있습니다. 클라이언트는 웹 브라우저를 통해 웹 서버로 요청을 전송하고 웹 서버는 이에 대한 응답을 클라이언트에게 전송합니다.

- HTTP 요청

HTTP 요청은 클라이언트가 서버에게 특정 동작을 요청하기 위해 전송하는 메시지이며 요청 페이지와 함께 서버에 전달하는 클라이언트의 정보를 포함하고 있습니다.

1) HTTP 요청 헤더 구성

```
GET /business/expert/eqst.do HTTP/1.1
Host: infosec.adtcaps.co.kr
Cookie: JSESSIONID=0E02DE7F0B79B7EBD6DEE9338CE17EE7; _ga=GA1.3.306907063.1646632323; _gid=GA1.3.
Cache-Control: max-age=0
Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://www.google.com/
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close
```

· GET /business/expert/eqst.do HTTP/1.1 : 요청 URL 정보 및 HTTP 버전

· Host : 요청 도메인

· Cookie : 클라이언트 측에 저장된 사용자 상태 정보

· User-Agent : 사용자의 웹 브라우저 종류

· Accept : 요청 데이터 타입

· Referer : 요청을 보낸 페이지의 URL

2) HTTP 요청 메소드

HTTP 요청 헤더 중 요청 메소드를 통해 클라이언트가 웹 서버에게 요청의 목적과 종류를 알립니다. 주로 GET, POST 방식으로 자원을 요청합니다. (TRACE, PUT, DELETE와 같은 메소드는 사용자가 웹 서비스를 이용할 때 필요하지 않기 때문에, 설정되어 있을 경우 취약점이 되기도해서 주의해야 합니다!).

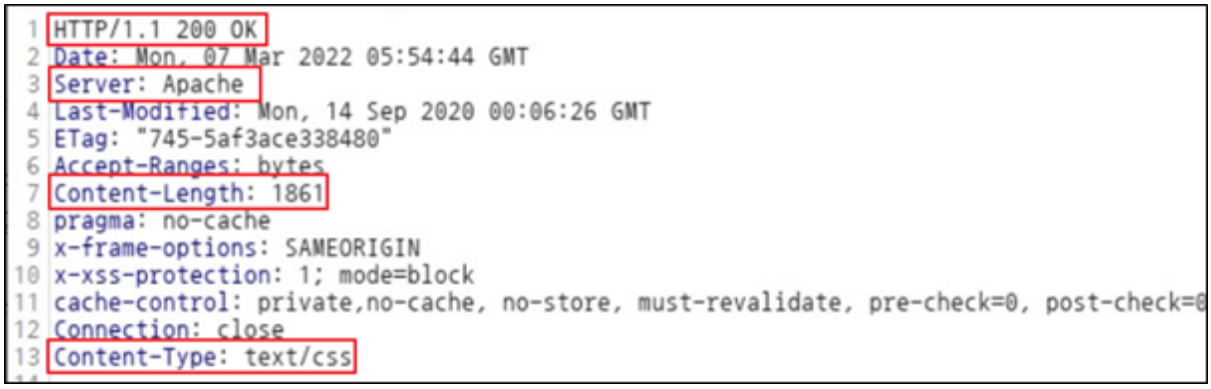
infosec

Method	의미
GET	서버 측에 자원 요청
POST	서버로 자원 전송
HEAD	HTTP 헤더 정보만 수신
TRACE	원격지 서버에 루프백 테스트
PUT	요청된 자원 갱신
DELETE	요청된 자원 삭제
OPTIONS	응답 가능한 HTTP 메소드 요청

- HTTP 응답

서버는 클라이언트로부터 요청이 오면 응답 헤더의 정보와 바디의 데이터를 포함하여 요청에 대한 응답을 합니다.

1) HTTP 응답 헤더 구성



· HTTP/1.1 200 OK : HTTP 버전과 응답 코드

· Server : 웹 서버 정보

· Content-Length : 응답 패킷의 길이

· Content-Type : MIME 타입

2) HTTP 응답 코드(상태 코드)

서버는 클라이언트가 보낸 HTTP 요청에 대한 응답 코드를 보내는데, 이를 보고 요청의 성공과 실패 여부와 같은 서버의 상태를 판단할 수 있습니다. 응답 코드는 100번대부터 500번대까지의 세 자리 숫자로 구성되며, 이 중 **클라이언트 오류를 나타내는 400번대 코드**와 **서버 오류를 나타내는 500번대 코드**를 주의 깊게 봐야 합니다. (아래의 표는 자주 볼 수 있는 HTTP 응답 코드의 몇 가지 예시입니다.)

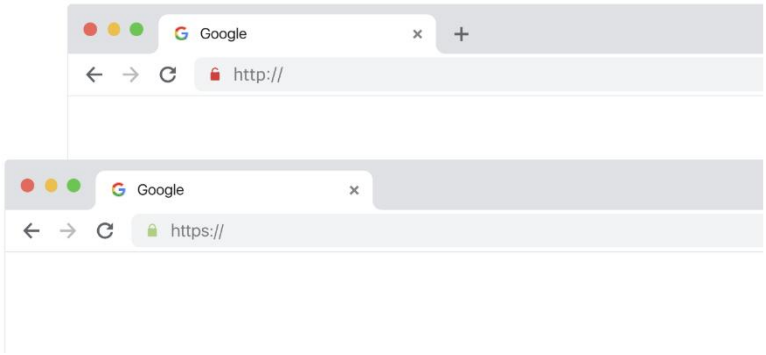
infosec

구분	응답 코드	응답 메시지	의미
1xx: 정보	100	Continue	진행 중
2xx: 성공	200	OK	요청에 대한 성공
	201	Created	요청 자원이 정상적으로 생성이 됨
3xx: 리다이렉션	301	Moved Permanently	요청한 자원이 새 URL에 존재
	302	Found	임시적으로 주소가 바뀌었을 경우
4xx: 클라이언트 오류	400	Bad Request	잘못된 요청
	401	Unauthorized	권한 없는 요청
	403	Forbidden	서버에서 해당 자원에 대한 접근 금지
	404	Not Found	요청 자원이 서버에 존재하지 않음
5xx: 서버 오류	500	Internal Server Error	내부 서버 오류
	502	Bad Gateway	게이트웨이로부터 잘못된 응답 수신
	503	Service Unavailable	현재 서버를 사용할 수 없음
	504	Gateway Timeout	게이트웨이가 응답을 받지 못함

(CS를 아예 모르는 사람이더라도 **404오류**는 겪어본 적이 있을 것입니다. 방금 알아본 것을 토대로 분석해보자면~ 오류가 떴는데 코드 번호가 404라면, 우선 서버가 아닌 클라이언트 쪽의 오류이고 내가 찾으려는 서버가 아예 존재하지 않아서 요청이 실패되었음을 알 수 있습니다.)

4. HTTP vs HTTPS

사실, CS에 대해 관심이 있는 사람이라면, HTTPS 가 HTTP보다 보안성이 더 뛰어난 프로토콜이라는 것은 알고 있을 것입니다. 그렇다면, 정확히 어떤 차이 때문에 HTTPS 가 더 안전한지 알고 계실까요?



HTTP (HyperText Transfer Protocol)는 광범위한 데이터의 빠르고 쉽고 안정적인 이동을 위한 기본적인 프로토콜 인터페이스입니다. 또한 HTTP는 **대량의 작은 포켓 전송**이

가능하고, **헤더 섹션이나 메세지에 크기 제한이 없습니다**.

HTTP 프로토콜은 **TCP/IP 기반**으로 동작하며, 전송 과정에서 장치 간 데이터가 손상되지 않도록 **안정적인 통신**을 가능하게 합니다. 그리고 데이터는 IP주소와 URL들을 기반으로 전송됩니다. HTTP 연결은 **동적**이며, **엑세스가 이루어 질때마다 종료**되는 것이 특징입니다.

HTTPS(HyperText Transfer Protocol over Secure Socket Layer)는 **SSL 및 TLS**를 통해 HTTP를 실행하는 프로토콜로, 네트워크 통신을 보호하기 위해 **암호화된 양방향 통신 채널**을 설정할 수 있습니다. 암호화/복호화 과정이 포함되어 있기 때문에 HTTP보다 데이터 통신은 느립니다.

하지만 신뢰할 수 있는 인증기관에서 서명한 인증서를 클라이언트와 서버에 제시함으로써 신원인증이 보장됩니다. HTTPS는 **실시간 통신에는 적합하지 않지만**, 사용자의 정보나 조직의 보안을 보호하기 위해 사용됩니다.

Table 1. Comparison of HTTP and HTTPS

	HTTP	HTTPS
Security	X	O
Encryption	X	O
Certificate	X	O
Port	80	443

 여기서 질문!

Q. HTTP는 안정적이라고 하고, HTTPS는 안전하다고 하는데 안전하지 않은 HTTP가 어떻게 안정적일 수 있나요?

A. HTTP가 "안정적" 이라고 하는 이유는 프로토콜이 단순하고 오랫동안 널리 쓰여서 호환성이 뛰어나다는 의미입니다. 별도의 암호화 과정이 없으니 성능 저하도 적고, 환경에 상관없이 "잘 동작한다"는 의미에서의 안정적임을 말하는 것입니다.

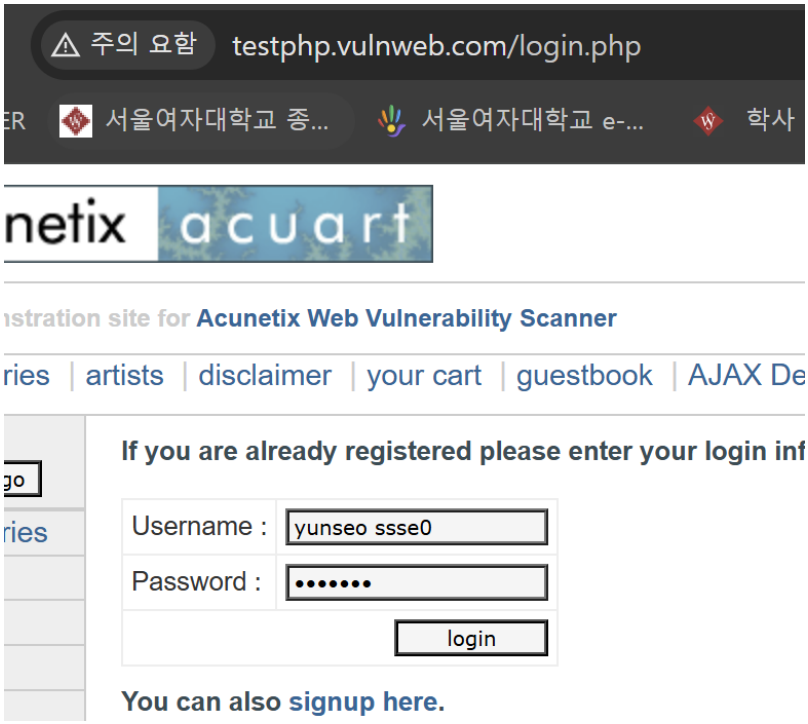
두 프로토콜 사이에 가장 큰 차이점은 바로 **SSL 인증서**입니다. SSL 인증서는 사용자가 사이트에 제공하는 정보를 암호화하는데, 이렇게 전송된 데이터는 중간에서 누군가 훔쳐낸다고 하더라도 해독할 수 없습니다. 그 외에도 HTTPS는 **TLS(전송 계층 보안) 프로토콜**

을 통해서도 보안을 유지합니다. TSL은 **데이터 무결성을 제공**하기 때문에 데이터가 전송 중에 수정되거나 손상되는 것을 막고, 사용자가 자신이 의도하는 웹사이트와 통신하고 있음을 입증하는 인증 기능도 제공하고 있습니다.

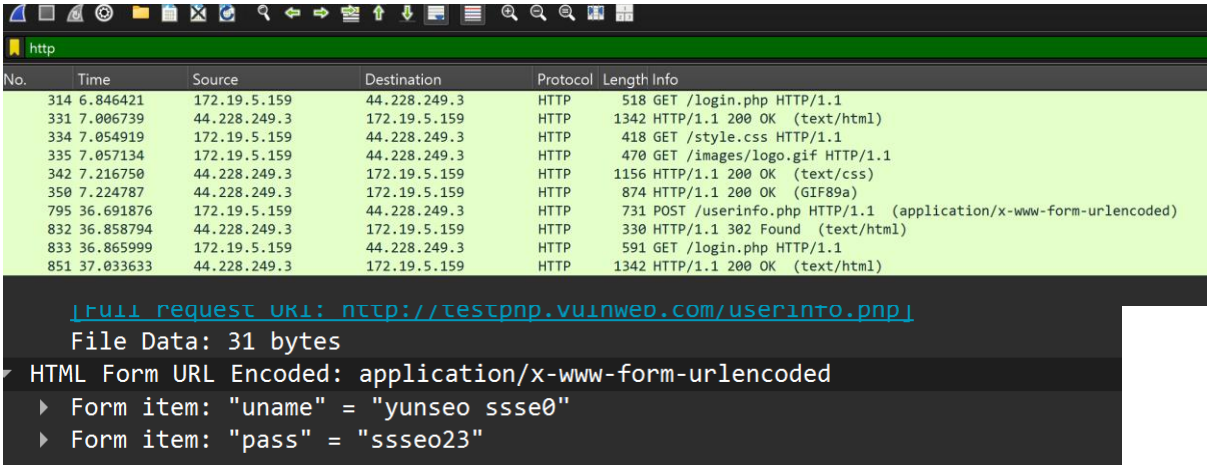
이러한 이유들로 2014년 구글에서는 HTTP를 HTTPS로 바꾸라고 권고하기도 하였는데, 그전까지는 전자상거래, 개인정보 데이터베이스를 다루는 웹 사이트에서만 다소 번거로운 HTTPS를 사용하고 있었습니다. 하지만 구글은 HTTPS로 전환을 장려하기 위해서 구글에서 HTTPS를 사용하는 웹 사이트에 대해서는 검색 순위 결과에 이익을 주겠다는 발표도 했다고 합니다..! (정말 HTTPS의 상용을 위해서 구글이 이렇게까지 했구나)

5. HTTP 패킷 분석

1. Wireshark 실행 → 인터페이스 선택 후 캡처 시작
2. 브라우저에서 http://testphp.vulnweb.com/login.php (Acunetix에서 제공하는 취약점 학습용 사이트, 로그인 가능) 접속
3. 임의의 아이디/비밀번호 입력



4. 와이어샤크에서 패킷 캡처를 중지시키고 http 필터를 적용해 캡처 된 패킷들을 확인합니다.



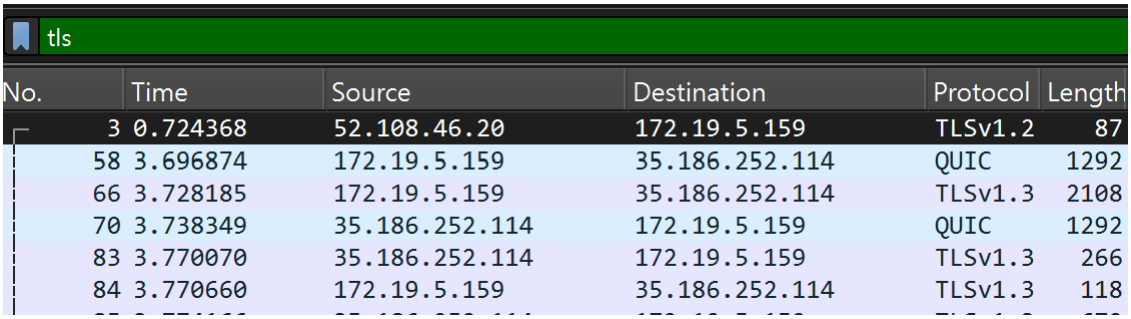
"login"이라는 텍스트가 보이는 패킷에 들어가보면 ... 제가 아까 입력한 아이디와 패스워드가 그냥 보입니다...

6. HTTPS 패킷 분석

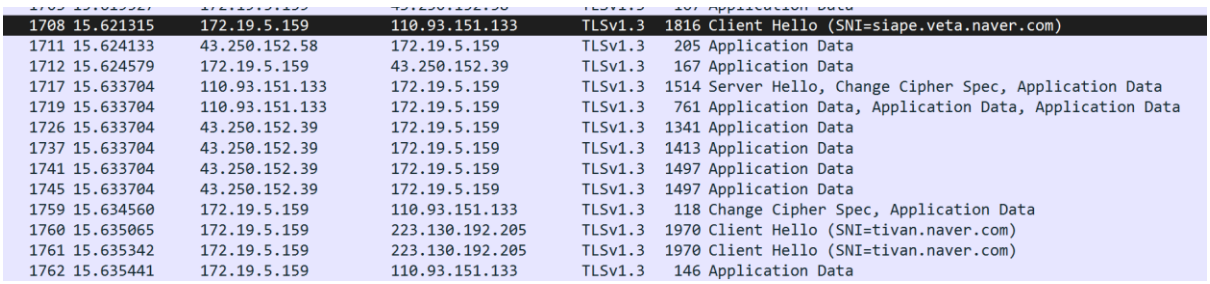
와이어샤크 실행은 5번과 동일하게 진행하고, 다만 웹에서 https:// 로 시작하는 아무 사이트나 접속해줍니다.

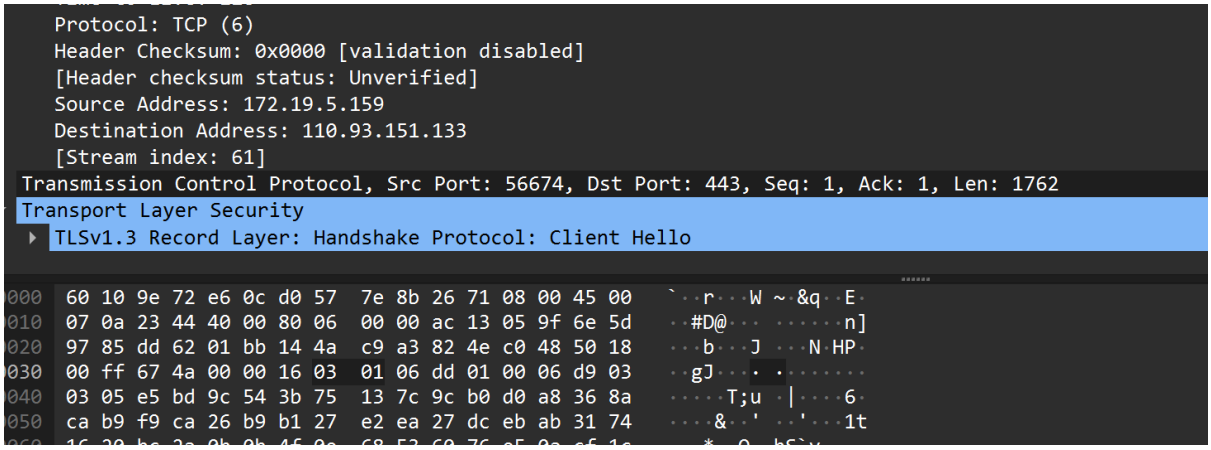
저는 naver.com으로 접속해 주었습니다.

동일하게 패킷 캡처를 마친 뒤, 이번에는 (HTTPS는 TLS 프로토콜을 거친다고 했으니)"tls" 필터를 적용해줍니다.



네이버에 접속한 기록이 보이는 패킷들을 볼 수 있고, 중간중간에 HTTP에서는 안 보이던 Application Data 라는 텍스트의 패킷들이 많이 보이는데... 이 패킷들이 암호화되어 전송된 패킷들입니다.





로그인 정보가 담길만한 패킷들을 열어보았지만 암호화되어 보이지 않았습니다.

6. 마무리

최신 통계에 따르면, 2025년 기준 미국 내 인터넷 트래픽 중 약 98%가 HTTPS를 사용하고 있다고 합니다. 국가별 차이는 있지만, 전 세계적으로도 HTTPS로의 전환이 거의 완료된 상태입니다. (https://securityscorecard.com/blog/https-vs-http-why-secure-connections-matter-in-2025/?utm_source=chatgpt.com)

그런 가운데 한국의 공공기관이 보안에 뒤쳐지고 있다는 느낌이 들어, 한국의 보안불감증에 대해 걱정이 되었습니다. 또한 이제는 HTTPS 프로토콜을 사용한다고 해도 여러 보안 취약점이 드러났기 때문에 머지않은 미래에 또 새로운 강력한 암호화 프로토콜이 나올 수도 있지 않을까? 라는 생각도 들었습니다.

참고자료

Hong, S., Kang, J., & Kwon, S. (2023). Performance Comparison of HTTP, HTTPS, and MQTT for IoT Applications. *International Journal of Advanced Smart Convergence*, *12*(1), 9–17. https://doi.org/10.7236/IJASC.2023.12.1.9

이준하, 배은정, 기환종, 김철영, 임선영. (2022-12-20). HTTP통신 방법과 주요 특징. 한국정보과학회 학술발표논문집, 제주.

한국재정정보원 사이버안전센터, [Special Report] 웹 취약점과 해킹 매커니즘, 2022 https://www.fis.kr/ko/major_biz/cyber_safety_oper/attack_info/security_news?articleSeq=2504

김윤희, “HTTP 웹사이트 방문자, 이렇게 계정 털린다” , ZDNet Korea, 2020, https://zdnet.co.kr/view/?no=20201109101050