SNI 차단 및 암호화 연구

윤재웅* 김경호** 서화정**†

*한성대학교 컴퓨터공학부

**한성대학교 IT융합공학부

A Study of SNI Blocking and Encryption

Jae-Woong Yun*

*School of Computer Engineering, Hansung University.

Kyung-Ho Kim** Hwa-Jeong Seo**†

**Division of IT convergence Engineering, Hansung University.

요 약

정부는 지난 2월 고도화된 인터넷 접속차단 정책인 SNI 차단을 도입했다. 주요 논란점은 차단 과정에서 불법적으로 패킷의 일부분이 해킹된다는 부분이다. 이로 인해 TLS 1.3에서 제안된 SNI의 암호화인 ESNI가 주목받고 있다. 이에 본 논문에서는 이전부터 현재 SNI 차단까지 과정 및 TLS 1.3에서 제안된 ESNI의 특징 및 드래프트에서 Open Issue로 언급된 ESNI를 통한 0-RTT 가능성 대해서 알아본다.

I. 서론

지난 2월, 정부는 HTTPS(Hypertext Transfer Secure) 및 우회 접속 방식으로 불법 사이트를 제공하는 해외의 유해 웹사이트에 대한 접속 차단 기능을 고도화하기 위해 SNI(Server Name Indication)차단 시행을 발표했다[1].

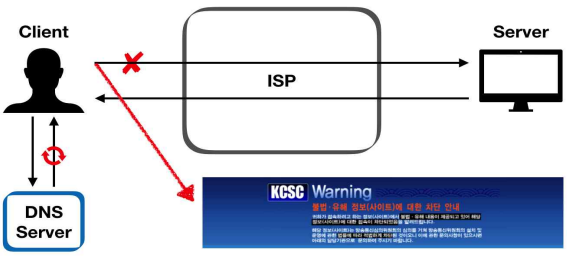
본 논문에서는 정부의 차단 방식과 이 과정에서 발생할 수 있는 보안상의 문제점을 해결할 수 있는 방법에 대해서 기술한다. 그리고 TLS(Transport Layer Security) 1.3에서 제안된 ESNI(Encrypted SNI)에 특징을 알아보고, 키 교환방식에 따른 기존 TLS 1.3에서 제안된 0-RTT(Round Trip Time) 과정을 드래프트의 Open Issue에서 언급한 Full Handshake 과정 없이 수행할 수 있는 가능성을 확인해본다.

구성으로 2장에서 과거부터 현재까지 진행된 차단방식들인 DNS, HTTP, HTTPS(SNI) 차단을 살펴보며, 3장에서는 SNI의 암호화인 ESNI의 특징 및 조건을 알아보고, 4장에서는 기존 제시된 방식과 ESNI를 통한 방식으로 0-RTT를 알아본다. 마지막으로 5장에서 결론으로 마무리한다.

II. 인터넷 차단

2.1 DNS 차단

DNS(Domain Name System)는 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있게 해준다. DNS에 주소를 요청할 때 해당 도메인 네임이 불법 사이트 목록에 등재된 것이라면 [그림 1]과 같이 실제 IP 주소가 아닌 “warning.or.kr"라는 경고 웹페이지로 리다이렉션 한다. 이러한 기술은 파밍(Pharming)이라고 부르며 불특정 다수에게 메일을 발송하여 위장된 웹페이지로 접속하게 한 후에 개인정보를 빼내가는 피싱(Finshing)의 발전된 기법이다.

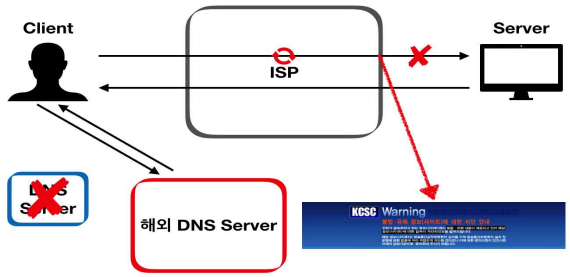


[그림 1] DNS 차단으로 리다이렉션

국내 인터넷 서비스 제공자(Internet Service Provider)가 운영하는 것이 아닌 구글 같은 해외 업체에서 제공하는 DNS를 이용하여 차단을 해결할 수 있다.

2.2 HTTP 차단

HTTP(HyperText Transfer Protocol)는 웹상에서 동작되는 프로토콜로 주로 HTML 문서를 주고받는 데에 쓰인다. HTTP 통신은 평문으로 수행되어 보안에 취약한 문제점을 가지고 있다. [그림 2]와 같이 DNS로부터 IP 주소를 수신한 웹브라우저가 국내 ISP(Internet Service Provider)에게 해당 주소로의 연결을 요청하는 순간, 이것이 불법 사이트 목록에 등재된 것일 경우 이를 차단하는 방식이다.



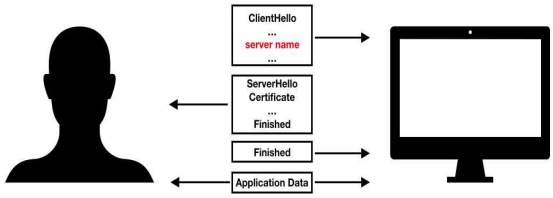
[그림 2] HTTP 차단으로 리다이렉션

TLS 프로토콜로 HTTP 통신을 암호화한 HTTPS를 통하여 클라이언트와 웹 서버 간 통신을 암호화하여 이를 해결할 수 있다.

2.3 HTTPS(SNI) 차단

SNI(Server Name Indication)는 TLS 프로토콜의 확장이며 이를 이용하여 동일한 IP 주소에 여러 개의 인증서를 사용할 수 있다. 따라서 동일 IP의 여러 호스트 인증이 가능하게 된다.

HTTP 통신의 암호화를 위해 TLS를 사용하는데, 이 경우 인증이 완료된 클라이언트와 웹 서버 사이의 통신은 암호화되는 반면 인증 과정에서의 통신은 암호화되지 않고 진행된다. [그림 3]과 같이 인증 과정 초기에 SNI 패킷을 주고받는데, 클라이언트와 웹 서버 간의 TCP+TLS Handshake 과정에서 아직 암호화 키를 전달받기 전 이기 때문에 ClientHello에서 SNI는 암호화되지 않은 평문으로 전달된다.



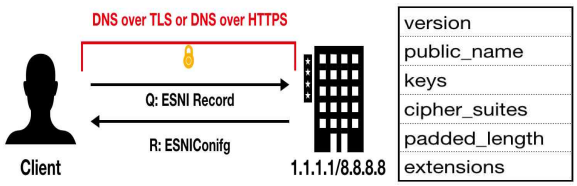
[그림 3] 평문으로 전달되는 SNI

SNI 정보는 네트워크상의 트래픽 모니터링은 물론 장애 문제를 해결하고자 하는 네트워크 관리자에게 매우 중요한 도구이며 네트워크에 가시성을 제공한다. 하지만 SNI 패킷에 포함된 도메인 정보를 확인하여 특정 웹사이트를 차단하는 것이 가능하다. 이로 인해 제 삼자에게 쉽게 노출이 되어 보안 문제가 생길 수 있다[2]. 다시 말해서, SNI 차단 방식은 HTTPS 통신을 하더라도 암호화가 시작되기 전에 SNI 필드에 호스트 이름이 평문으로 표시되는 것을 이용하여 ISP가 웹사이트를 차단하는 방식이다.

III. ESNI

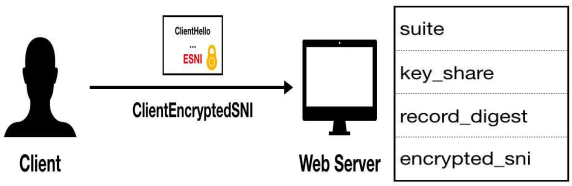
3.1 SNI 암호화를 위한 키 교환

ESNI는 TLS 1.3에서 제안된 SNI 암호화이다. 암호화를 하기 위해서 웹 서버는 신뢰할 수 있는 DNS 서버에 공개키를 미리 올려놓는다. 그리고 클라이언트가 DNS에 요청 시 TXT 레코드를 통해 ESNIConifg 구조체를 받는다. 이 구조체는 [그림 4]의 값들을 갖고 있다.



[그림 4] ESNIConfig 구조체

클라이언트는 웹 서버의 공개키와 자신의 공개키로부터 대칭키를 유도한다. 이 대칭키를 통해 [그림 5]의 ClientEncryptedSNI 구조체로 암호화한다. 그리고 클라이언트의 공개키와 함께 ClientHello 패킷을 통해 보낸다. 웹 서버는 자신의 개인키와 클라이언트로부터 받은 공개키를 사용하여 동일한 대칭키를 유도하여 ClientEncryptedSNI 구조체를 복호화한다. 여기서 대칭키 유도는 DH(Diffie-Hellman) 방식을 사용한다.



[그림 5] ClientEncryptedSNI 구조체

키 교환 시 고려해야 할 문제점은 HTTPS 기반의 웹 서버와 DNS 서버 상호 간에 동기화가 동시에 이루어져야 한다는 것이다. SNI 암호화에 사용되는 키의 만료기간으로 인한 갱신이 요구되는데 이러한 갱신이 이 두 서버에 동시에 반영되어야 한다. 일반적으로 웹 서버의 경우 ISP가 직접 관리하지만, DNS 서버는 ISP 제공 서버 혹은 외부의 네임서버를 사용하는 경우가 많기 때문에 어려움이 예상된다.

시험적으로 ESNI를 시행하고 있는 Cloudflare의 경우 웹 서버는 전방 익명성을 향상하기 위해 매 시간 키 갱신을 하며, 수 시간 동안의 키를 갖고 있는다[3]. 웹 서버는 키 갱신 문제를 감지하여 클라이언트에게 새롭게 갱신된 ESNI 레코드를 제공한다. 이때 기존의 ESNI를 완전히 비활성하지 않고 새로 갱신된 레코드를 통해 재연결하도록 한다.

하지만 ESNI(Encryption SNI)를 이용하고 평문으로 통신하는 DNS를 사용한다면 SNI를 암호화한 의미가 무색해진다. 중간 경로의 관찰자는 암호화된 SNI의 사용여부와는 관계없이 클라이언트가 보내는 평문 DNS 질의를 관찰하는 것만으로 사용자가 어떤 웹사이트에 접속하는지 알 수 있다.

즉 ESNI는 DNSSEC, DoT 또는 DoH와 같은 DNS 인증 및 암호화를 이용해야 인터넷상의 감시와 검열 문제 해결에 효과가 있다[4].

3.2 DNSSEC

DNSSEC(DNS Security Extension)은 DNS 데이터 대상의 “데이터 위조-변조 공격”을 방지하기 위한 인터넷 표준기술이다. DNS의 위조-변조 가능성을 원천적으로 차단하기 위해 DNSSEC의 공개키 암호화 방식의 전자서명 기술을 사용한다.

클라이언트가 얻은 DNS 정보가 실제로 맞는 정보인지에 대한 검증은 제공하지만, DNS 레코드는 물론 DNSSEC에 의해 서명된 레코드에 대해서는 암호화를 제공하지 않는다.

3.3 DNS-over-TLS

DoT(DNS-over-TLS)는 클라이언트와 DNS 서버 간의 TLS 프로토콜 기반 암호화된 통신을 통해 DNS 요청을 근본적으로 감청할 수 없게 만들 뿐만 아니라 중간자 공격을 차단함으로써 DNS 스푸핑 공격의 가능성을 줄인다.

3.4 DNS-over-HTTPS

DoH(DNS-over-HTTPS)는 DNS 요청을 평문이 아닌 HTTPS를 통해 DNS 정보에 접근한다. 이를 통하여 DoH는 DNS 요청을 HTTP 요청인 것처럼 위장하기 때문에 높은 보안성 효과 및 확장성을 제공한다.

3.5 DoT vs DoH

DoT는 TCP를 기본 연결 프로토콜로 사용하고 TLS 암호화 및 인증을 통해 계층화하는 반면 DoH는 HTTPS 사용하여 연결한다.

DoT는 자체 전용 포트인 853을 사용하며 DNS 요청이 암호화되기 때문에 내용을 알 수는 없지만 자체 전용 포트 번호로 인해 DoT를 사용하고 있다는 것을 알 수 있다. 반면, DoH는 기존 HTTPS 트래픽의 표준 포트인 443을 사용하므로 HTTPS 통신과 DoH 통신을 구분할 수 없다. 즉, DoT와는 다르게 DoH는 사용하고 있다는 것을 숨길 수 있다[5].

IV. 0-RTT

4.1 TLS 1.2 재연결

HTTP를 암호화하는 과정을 매번 Full Handshake를 하는 것은 비효율적이다. 그래서

이미 한번 과정을 거친 통신을 간소화하는 두 가지 방법으로 Session ID, Session Ticket이 있다.

Session ID와 Session Ticket 모두 초기 Full Handshake 과정에서 서버로부터 값을 할당받는다. 이 값으로 다음 Handshake 과정부터는 상호 인증을 간소화하여 수행할 수 있게 된다.

RFC 5077에 의하면 Session ID보다 Session Ticket의 사용을 더 권장한다. 그 이유는 Session ID는 서버가 발행한 값들을 모두 메모리에 기억해야 하지만 Session Ticket은 클라이언트가 해당 내용을 기억하고 서버로 전달하는 방식이라 메모리 자원에 영향이 적기 때문이다. 그러나 클라이언트나 서버 둘 중 하나라도 Session Ticket을 지원하지 않는 경우에는 Session ID를 사용하게 된다.

4.2 TLS 1.3의 0-RTT

TLS 1.3의 0-RTT의 시범적 도입으로 기존 방법보다 빠르게 처리할 수 있는 가능성을 제시했다. 0-RTT는 기존의 Session ID, Session Ticket과 같은 역할을 수행하는 PSK(Pre Shared Key)를 클라이언트의 ClientHello 단계에서 요청할 Application Data와 함께 미리 암호화를 미리한 후 전송하는 방식이다. 서버는 ClientHello에서 PSK를 확인하고 검증을 수행고 오류가 없다고 판단되면 ServerHello에 암호화된 Application Data를 전송하게 된다[7].

4.3 ESNI를 통한 0-RTT

ESNI 드래프트의 Open Issue에서 언급된 것으로 가능성을 확인해 본다. SNI를 암호화하기 위해 웹 서버의 공개키를 사전에 신뢰할 수 있는 DNS 서버에 올린다. 이 과정을 통해 클라이언트는 웹 서버와 통신하기 전 이미 자신과 웹 서버의 공개키를 DH(Diffie-Hellman) 방식을 통해 ESNI-PSK로 유도할 수 있다. 이는 기존의 TLS 1.3의 0-RTT 방식인 PSK를 DNS 프로토콜을 통해 더욱 이전 과정에서 교환한 것이다. 그러므로 기존 PSK를 통한 0-RTT를 더 이른 과정에서 초기 Handshake 과정 없이 수행할 수 있다는 가능성이다[6].

V. 결론

본 논문에서는 인터넷 차단 방식에 대해서 과거부터 현재까지 살펴보았고, SNI의 암호화 방법을 알아보았다. 또한 ESNI와 DoH, DoT, DNSSEC의 조합을 통하여 통신의 전체적인 암호화에 도달 할 수 있으며, 기존의 TLS 1.3의 0-RTT를 Open Issue에서 언급한 ESNI를 통하여 초기 과정 없이 사용할 수 있는 가능성을 확인했다.

[참고문헌]

[1] 김평수. (2019). 안전한 웹사이트 접속을 위한 IETF 표준 기술 동향 분석. 한국통신학회지(정보와통신), 36(6), 32-40.

[2] A. Ghedini, Encrypt it or lose it: how encrypted SNI works, Cloudflare, September 2018.

[3] A. Ghedini, Encrypt that SNI: Firefox edition, Cloudflare, October 2018.

[4] A. Ghedini, Encrypt it or lose it: how encrypted SNI works, Cloudflare, September 2018.

[5] P. Nohe, What is the difference between DNS over TLS & DNS over HTTPS?, The TLS Store, December 2018.

[6] E. Rescorla RTFM, Inc. K. Oku Fastly N. Sullivan Cloudflare C.Wood Apple, Inc. Encryption Server Name Indication for TLS 1.3 draft-ietf-tls-esni-05, November 04, 2019.

[7] N.Sulliva, Introducing Zero Round Trip Time Resumption(0-RTT), Cloudflare, March. 2017.