원격 키로깅 공격에 대응하기 위한 보안 메커니즘 연구*

김경이 ○ 조진성 전석희
경희대학교 컴퓨터공학과
kyungee.kim@gmail.com chojs@khu.ac.kr jeon@khu.ac.kr

Study of Security Mechanisms to Counter Remote Keylogging Attack on Search Engine Autocomplete

KyungE Kim ○ JinSung Cho SeokHee Jeon
School of Computer Science and Engineering, Kyung Hee University

요 약

원격 키로깅 공격(Remote Keylogging Attack)은 원격으로 컴퓨터의 모든 키스트로크를 캡처하는 공격으로써, 검색 엔진의 자동 완성 기능 수행 시 암호화 된 패킷에서도 프레임의 길이 및 키 스트로크 타이밍 등을 통해 검색 쿼리를 예측한다. 이는 사용자의 위치, 질병 등 개인의 민감한 정보를 노출시킬 뿐만 아니라, 지속적인 트래킹을 당할 수 있다. 이에 본 논문에서는 검색 엔진의 자동완성 기능 시 검색 쿼리를 암호화하고 더미 트래픽을 함께 보내는 방안을 제안하여 해당 원격 키로깅 공격에 방어할 수 있도록 한다.

1. 서 론

자동 완성(Autocomplete)은 사용자가 입력한 낱말 의 나머지 부분을 응용프로그램이 예측하는 기능으로써, 우리가 사용하는 대부분의 검색엔진(Search Engine)은 자동 완성 기능을 가지고 있다. 하지만 이 기능의 경우 사용자의 키 스트로크(Keystroke) 이벤트에 의해 검색 쿼리가 노출될 가능성이 있다. USENIX Security Symposium 2019의 한 논문 [1]에서는 사용자의 키 스트로크 패턴을 유추하여 단어의 길이를 예측하고 키 스트로크 타이밍에 따른 검색 쿼리 예상 리스트를 추출하는 공격 방법을 제안하였다.

자동 완성에 쓰이는 사용자의 검색 쿼리는 정치적 선호도, 개인 식별 가능 정보 등 민감한 개인정보를 포함할 수 있으며, 지속적인 트래킹(Tracking)을 통해 사용자에게 심각한 피해를 줄 수 있다.

따라서 본 논문에서는 위와 같은 공격에 대응하는 두 가지 방안을 제안한다. 첫 번째는 검색 쿼리를 일정한 길이로 암호화하여 공격자가 순수 키 스트로크 패턴을 추출하지 못하도록 한다. 두 번째는 키 스트로크 이벤트에 따른 통신 시 더미 트래픽(Dummy Traffic)을 추가하여 사용자가 실제로 보낸 요청(Request) 횟수를 알아내지 못하도록 한다. 이를 통해 공격자가 검색 쿼리의 길이 및 단어 구성을 알아내지 못하도록 유도함으로써 원격 키 로깅 공격을 방어한다. 이 연구는 구글(Google) 검색 엔진을 기준으로 작성되었다.

* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 사업의 연구결과로 수행되었음. (No. 2017-0-00093)

2. 기존 연구

2.1 HTTPS

HTTPS(Hypertext Transfer Protocol over Secure Socket Layer)는 WWW(World Wide Web, W3) 통신 프로토콜인 HTTP의 보안이 강화된 버전이다. 통신의 인증과 암호화를 위해 개발되었으며 전자상거래에 널리 쓰인다. 또한 SSL이나 TLS 프로토콜을 통해 세션 데이터를 암호화하여 데이터의 적절한 보호를 보장한다.

아래는 각각 HTTP와 HTTPS를 통해 전송된 패킷이다. HTTPS로 전송된 패킷은 데이터가 암호화 되어 패킷을 가로채더라도 해독이 불가하다.

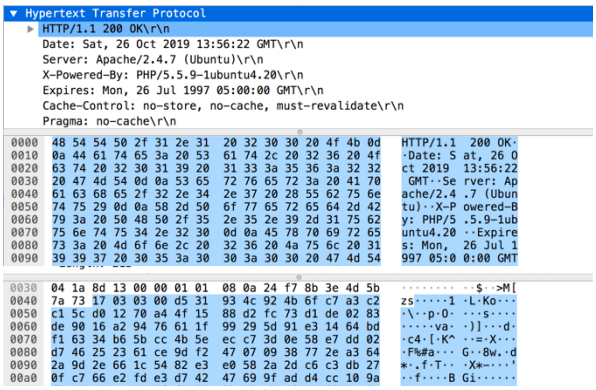

그림 1 HTTP와 HTTPS(TLS) 패킷

2.2 검색엔진의 자동 완성 기능을 통한 원격 키로깅 공격

[1]에서는 특정 검색 엔진(Google, Baidu)에서 자동 완성 기능 사용 시 HTTPS 통신을 사용하더라도 원격 키 로깅 공격이 가능하다는 것을 보여준다.

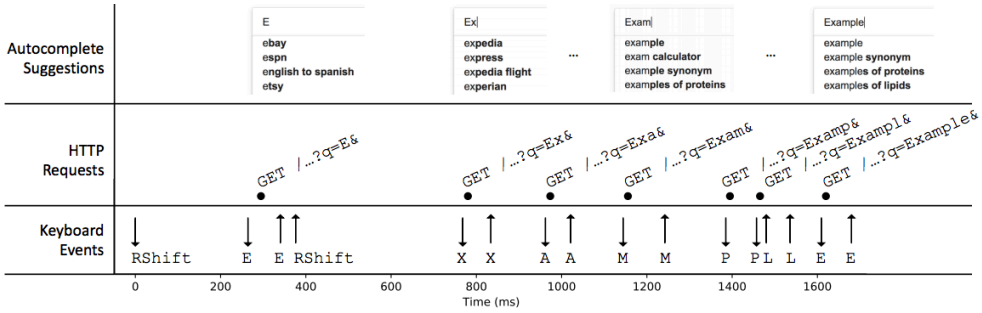

그림 2 자동 완성 기능 사용 시 사용자 키 스트로크에 따른 요청[3]

그림 2와 같이 사용자가 검색 쿼리를 입력할 때마다 키 스트로크 이벤트가 발생된다. 클라이언트에서는 검색 엔진의 자동 완성 기능에 의해 네트워크 트래픽을 유도한다. 이때 클라이언트와 서버 사이에 오가는 패킷 파일을 토대로 도착 시간과 프레임(Frame)의 길이를 통해 각종 패킷들 중 순수 키 스트로크 패킷만을 추출 할 수 있다. 이는 문자와 스페이스(Space) 키의 크기가 다르기 때문이다. 따라서 검색 쿼리 원문이 총 몇 글자 이며 그 중 띄어쓰기가 몇 번째에 위치해 있는 지에 대한 키 스트로크 패턴을 알아낼 수 있다.

그후, 사용자가 검색한 쿼리 리스트를 예측한다. 2개의 단일 문자 키의 키보드 위치 기반 상관관계(Mean, Std)가 명시되어 있는 바이그램(Bigram)을 통해 키 스트로크 타이밍에 따른 단어 개연성(Word Probabilities) 목록을 생성한다.[4] 그 후, 언어 모델(Language Model)과 생성한 단어 개연성 목록을 바탕으로 사용자 검색 쿼리로 예상되는 리스트를 추출할 수 있다[5].

3. 문제 정의

국내에서도 HTTPS와 같은 보안 채널을 사용할 것을 권고하듯이[2] 이미 HTTPS의 중요성은 널리 알려져 있다. 하지만 보안 채널을 통해 전송하여 암호화 된 패킷이더라도 자동 완성 기능 사용 시 원격 키로깅 공격을 통해 사용자 검색 쿼리를 예측 가능하다는 것을 [1]에서 보여주고 있다. 이 공격은 비단 자동 완성 기능에만 국한된 것이 아니다. 이외에도 사용자에 의해 키 스트로크가 발생하여 클라이언트에서 서버가 통신하는 모든 상황에서 해당 공격이 통할 수 있다.

원격 키로깅 공격에서 공격자가 검색 쿼리를 예측할 수 있는 이유는 크게 두 가지이다. 바로 사용자의 순수 키 스트로크 패킷을 추출할 수 있다는 점과 해당 패킷들의 순차적인 프레임 길이 증가이다. 따라서 본 논문에서는 원격 키로깅 공격에 대해 두 가지 방법으로 방어한다. 해당 공격은 입력 값을 복사하여 붙여 넣은 쿼리, 백 스페이스 및 딜리트 (Delete) 키와 같은 키 스트로크는 제외하므로 본 연구 또한 키 스트로크의 내용은 알파벳과 스페이스 키로 제한한다.

4. 해결 방안

4.1 검색 쿼리 암호화

	src	dst	frame_time	frame_length	protocol
9	192.168.116.128	172.217.25.196	1.571946e+12	153	6
16	192.168.116.128	172.217.25.196	1.571946e+12	155	6
23	192.168.116.128	172.217.25.196	1.571946e+12	156	6
30	192.168.116.128	172.217.25.196	1.571946e+12	156	6
38	192.168.116.128	172.217.25.196	1.571946e+12	157	6
45	192.168.116.128	172.217.25.196	1.571946e+12	159	6
52	192.168.116.128	172.217.25.196	1.571946e+12	160	6
59	192.168.116.128	172.217.25.196	1.571946e+12	160	6
67	192.168.116.128	172.217.25.196	1.571946e+12	161	6

그림 4 사용자 키 스트로크 패킷 추출

구글 검색엔진에서는 하나의 키 스트로크당 0~2만큼의 프레임 길이가 증가한다. +0과 +1은 알파벳이며 +2는 스페이스 키이다. 여기서 프레임 길이의 증가율이 단일문자임에도 +0인 것과 스페이스 키의 URL 인코딩(Encoding) 값이 '%20'임에도 불구하고 +2만 증가하는 것은 HTTP2 헤더 압축(Header Compression)의 결과이다[6]. 이를 통해 공격자는 검색 쿼리에서 문자와 띄어쓰기의 구성을 알아낼 수 있다.

그림 4는 서버와 클라이언트 간 통신하며 주고받은 모든 패킷 중 자동 완성 관련 패킷만을 추출한 결과이다. 검색 쿼리 원문은 'I can see' 였으며, 16번과 45번이 이전 패킷 프레임의 길이와 비교해 2씩 증가했으므로 스페이스 키 패킷임을 알 수 있다.

Name	Status	Type
search?q=iOo%2FK7rTjJAXYvgjUcZmkQ%3D%3D&cp=...	200	xhr
search?q=X9jhJlwtP8ejXfW9mDY%2F6A%3D%3D&cp=...	200	xhr
search?q=F3eNGPxd%2BkbhQ97HHKO0OA%3D%3D...	200	xhr
search?q=vLMaH16Sx5aJfnQehNFSbQ%3D%3D&cp=2...	200	xhr
search?q=4F6O6MyNohM%2ByLoYoTi6uA%3D%3D&c...	200	xhr
search?q=Enzr1zQKN4VN54p5pM4dOA%3D%3D&cp=...	200	xhr
12 requests | 1.4 KB transferred | 659 B resources		

그림 5 암호화를 적용한 자동완성 요청

따라서 본 연구에서는 프레임 길이의 순차적인 증가를 없애기 위하여 원문 쿼리를 AES 암호화하여 통신한다. 그림 5에서 볼 수 있듯 사용자가 키 스트로크를 할 때마다 암호화 된 검색 쿼리와 대칭 키를 함께 서버로 전송한다. 검색 쿼리는 항상 동일한 크기로 인코딩 되기 때문에 패킷의 프레임 길이 또한 동일하다. 따라서 순수 키 스트로크 패킷만을 추출하는 것은 불가능하다. 만약 추출하더라도 문자와 띄어쓰기 구성을 알아낼 수 없기 때문에 공격에 성공하는 것은 불가능에 가깝다.

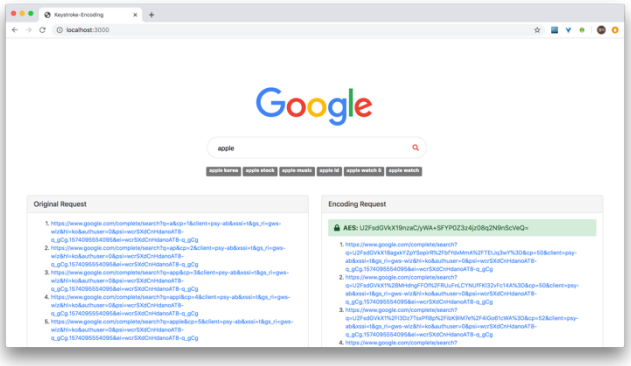

그림 6 검색 엔진(google) 환경 구성

그림 6은 특정 검색엔진 환경과 동일하게 구현한 서버이다. 좌측은 현재 해당 검색 엔진에서 자동 완성 기능을 위해 주고받는 요청이고 우측은 AES 암호화된 요청이다. 기존 방식에서 한 글자씩 증가되는 쿼리를 URL 자체에서 확인 가능하나, 암호화된 방식의 경우에는 항상 같은 길이를 유지하는 것을 볼 수 있다.

	src	dst	frame_time	frame_length	protocol
14	104.17.64.4	172.16.251.175	1.574252e+12	1460	6
27	23.111.9.35	172.16.251.175	1.574252e+12	1460	6
	src	dst	frame_time	frame_length	protocol
206	172.217.24.138	172.16.251.175	1.574252e+12	215	6
207	172.217.24.138	172.16.251.175	1.574252e+12	231	6

그림 7 암호화 적용한 패킷 추출

그림 7은 기존 검색 엔진과 동일하지만 암호화를 적용한 환경에서 서버와 클라이언트가 주고 받은 모든 패킷들 중 자동완성 패킷을 추출한 결과이다. 3.1을 적용한 결과 순수 키 스트로크 패킷을 추출하지 못하는 것을 볼 수 있다. 이 방안은 공격자로 하여금 검색 쿼리에서 문자와 띄어쓰기 구성을 알아내지 못하게 함으로써 원문 검색 쿼리를 보호 하는 데에 의의가 있다.

4.2 더미 트래픽 추가

공격자가 원문 쿼리를 알아내지 못하도록 방어하는 또 하나의 방법은 더미 트래픽을 추가하는 것이다. 사용자 키 스트로크 시 보내는 요청의 프레임 길이와 동일하거나 +1~2한 더미 패킷을 전송한다.

	src	dst	frame_time	frame_length	protocol
13	192.168.0.2	216.58.197.196	1.573403e+12	167	6
27	192.168.0.2	216.58.197.196	1.573403e+12	168	6
41	192.168.0.2	216.58.197.196	1.573403e+12	169	6
54	192.168.0.2	216.58.197.196	1.573403e+12	170	6
98	192.168.0.2	216.58.197.196	1.573403e+12	170	6
113	192.168.0.2	216.58.197.196	1.573403e+12	171	6

```
[7] Done: keystroke_timing(bigrams, keystrokes, word_lists)
[8] Done: predict_phrases(word_probas, language, k=k, alpha=a
really worth about
really nerve cells
really worth every
really offer other
really serve under
```

그림 8 더미 트래픽 추가한 결과

이는 키 스트로크 패킷과 비슷한 크기의 패킷을 전송함으로써 공격자가 키 스트로크 패턴을 찾아내는 데에 혼란을 줄 수 있다. 또한 공격자가 키 스트로크 패턴을 찾아내더라도 추출한 패킷에 더미 트래픽 포함되므로 사용자의 원문 검색 쿼리 길이를 유추하지 못한다. 그림 8은 더미 트래픽을 추가한 결과이다. 공격자는 더미 트래픽을 순수 키 스트로크 패킷으로 오인하여 원문 글자 수를 더 큰 값으로 생각할 수 있으며 +2한 패킷의 경우 띄어쓰기 패킷으로 잘못 파악할 수도 있다. 그림 8에서는 3개의 더미 패킷을 알파벳으로 인식하여 원문 쿼리와 비교해 각 단어마다 1글자씩 추가된 추측 리스트를 추출한 것을 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 검색 엔진의 자동 완성 기능에서의 원격 키로깅 공격에 대한 보안 메커니즘을 제안하였다.

3.1을 통해 프레임의 길이를 항상 동일하게 하여 순수 키 스트로크 패킷을 추출하지 못하도록 하였다. 또한 패킷을 추출하더라도 문자와 띄어쓰기 구성을 모르게 하여 원문 쿼리를 방어할 수 있다. 3.2에서는 더미 트래픽을 추가하여 공격자가 사용자의 순수 키 스트로크만을 추출하지 못하도록 하였다. 추출 패킷에 더미 트래픽이 섞이게 하거나 문자와 띄어쓰기를 오인하게 하여 원문 쿼리를 방어할 수 있다.

본 논문에서 제안한 보안 메커니즘의 경우 키 스트로크 패턴을 감지하지 못하게 하여 원문 쿼리를 예측하지 못하게 방어하는 것이 핵심이다. 이를 통해 더욱 더 보안이 강화된 통신을 할 수 있을 것이며, 개인의 프라이버시를 지킬 수 있을 것이다.

향후 연구로는 자동완성 특성상 서버와 클라이언트 간 빠른 통신이 이루어져야 하는 만큼 더욱 더 최적화 된 암호화 알고리즘을 연구할 계획이다.

참 고 문 헌

[1] John V. Monaco, “What Are You Searching For? A Remote Keylogging Attack on Search Engine Autocomplete” in 28th USENIX Security Symposium, 2019

[2] 한국인터넷진흥원(KISA), SW개발보안가이드, 2017

[3] John V. Monaco, “Feasibility of a Keystroke Timing Attack on Search Engines with Autocomplete” in IEEE Symposium on Security and Privacy Workshops, 2019

[4] John V. Monaco, “SoK: Keylogging Side Channels” in IEEE Symposium on Security and Privacy Workshops, 2018

[5] John V. Monaco, KREEP(Keystroke Recognition and Entropy Elimination Program), 2019
https://github.com/vmonaco/kreep

[6] R. Peon and H. Ruellan. HPACK: Header compression for HTTP/2. Technical report, 2015.