역공학을 이용한 페이스북 프로토콜 추론

정인식, 주홍택*
계명대학교 컴퓨터공학과
e-mail : {dlstlrdl77, juht}@kmu.ac.kr

Facebook Protocol Inference using Reverse Engineering

In-Sik Jung, Hong-Taek Ju
Dept. of Computer Engineering, Keimyung University

요 약

본 논문에서는, 패킷 모니터링을 이용하여 모바일 환경에서 페이스북 서버와 클라이언트의 어플리케이션간의 동작을 분석하고 페이스북 Graph API 를 사용하여 프로토콜을 분석하였다. 페이스북 프로토콜의 분석결과는 향후 다양한 플랫폼에서 페이스북 사용과 게이트웨이 서버와 페이스북 서버간의 통신 기능을 수행하는데 활용하고자 한다.

1. 서론

SNS 란, Social Network Service 의 약자로 “Six Degrees”이론을 기반으로 사람들간의 관계를 만들고 그 관계를 관리하는 것을 도와주는 서비스이다. “Six Degrees”이론은 어느 사회의 관계에서도 무작위의 두 사람 사이의 관계를 찾기 위해서는 6 단계만 거치면 된다는 것이다[1]. 이 이론을 따라 관계의 중요성이 부각 되었다. SNS 는 오프라인에서만 관리하던 관계를 물리적으로 제약을 받지 않고 온라인으로 관리할 수 있게 해주며 편리함으로 인해 SNS 의 인기가 높아졌다. SNS 의 종류에는 페이스북, 트위터, 미투데이 등 여러 종류가 있는데 가장 많은 사용을 하는 페이스북을 선정하여 분석하였다[5]. 페이스북은 모바일 어플리케이션을 제공하고 있는데 소셜 미디어 통계 사이트인 Social Bakers 에서는 전 세계 페이스북 사용자의 수 9 억 명 중 약 5 억 명이 모바일 페이스북을 이용 사용하는 것으로 나타났다[3]. 현재 모바일 페이스북의 편리함으로 보아 모바일 페이스북 사용자의 증가할 것으로 예상되며 그에 따라 모바일 단말과 페이스북 서버간의 트래픽도 점점 증가할 것이라고 예상된다. 페이스북 서버와 모바일 단말 사이의 트래픽 증가로 인해 망 과부하가 발생할 경우 모바일 어플리케이션을 사용에 불편을 겪을 것이다.

현재 페이스북은 공개된 프로토콜 규약을 가지고 있지 않다. 만약 역공학을 이용하여 페이스북의 메시지 전달과정에서 프로토콜들 규약을 추론하여 알아낼 수 있다면 페이스북의 활용도가 높아질 것이다.

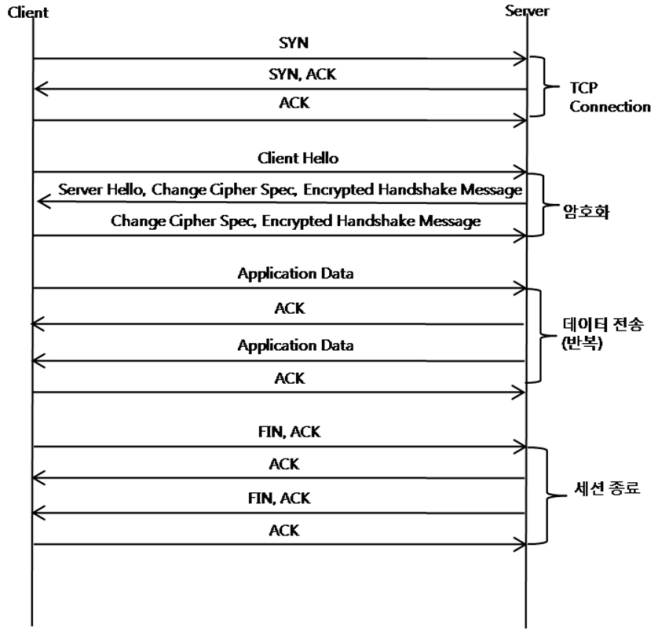
분석한 정보들을 이용하여 현재 페이스북이 지원하지 않는 다양한 플랫폼에서 페이스북 어플리케이션의 기능을 사용할 수 있을 것이다. 그리고 게이트웨이 서버를 만들어 트래픽 모니터링을 할 수 있을 것이고 페이스북 서버로부터 캐싱 기능을 구현하여 통신 향상을 위한 방안을 연구하는데 도움이 될 것이다.

프로토콜 분석 방법은 Passive Monitoring 을 이용한 페이스북 프로토콜 절차 분석을 하고 Open API 와 페이스북 SDK 2.0 를 이용해 데이터 전달의 포맷을 분석을 한다.

2. Passive Monitoring 을 이용한 추론

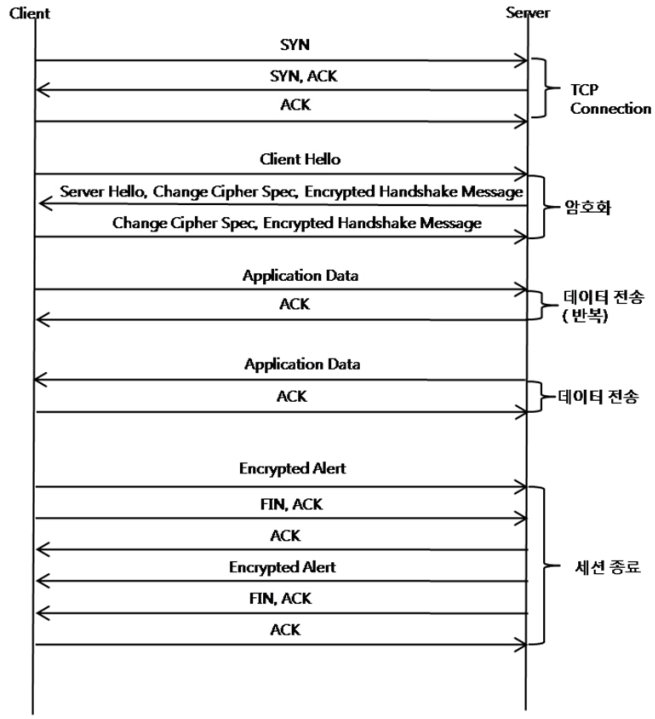
모바일 페이스북의 데이터 처리과정과 서비스 절차를 분석하기 위하여 다음과 같은 실험환경을 구성한다. 먼저 안드로이드 공식 페이스북 어플리케이션을 이용하고 3G 와 Wi-Fi 의 두 가지 환경에서 각각 실험한다. 모바일 단말으로는 SHW-M180S 기기(OS 2.3.4 Gingerbread)를 사용하고 모바일 환경에서 패킷을 캡쳐하기 위해서 Wire Shark For Root 어플리케이션을 사용한다. 안드로이드 환경에서 직접적인 네트워크 통신을 보기 위해서는 안드로이드의 루트권한이 필요하기 때문에 단말을 안드로이드 루팅을 하여 사용한다. Wi-Fi 환경에서 패킷을 캡쳐할때는 노트북을 이용해 Wi-FI 망을 만든 후 tcpdump, WireShark 같은 패킷 캡쳐 프로그램을 사용하여 실시간으로 패킷을 보며 분석할 수 있지만 3G 환경과 동일한 환경에서 실험을 하기 위해서 모바일 단말에서 패킷을 캡쳐 하고 단말에서 PC 로 캡쳐한 파일을 옮겨 WireShark 를 사용하여 분석하였다.

캡쳐한 패킷들을 이용하여 모바일 페이스북의 특정 서비스가 실행 될 때 페이스북 서버와 단말 사이에서 통신순서나, 패킷의 크기 같은 다양한 정보를 얻을 수 있다. 먼저 3G 와 Wi-Fi 상태에서 페이스북 어플리케이션을 사용하여 담벼락에 영어로 된 5 글자를 전송하고 분석을 하였다.



(그림 1) Wi-Fi, 3G 환경에서 글쓰기

3G, Wi-Fi 환경 모두 DNS 를 통해 도메인 Graph.facebook.com 을 검색하여 접속하였고 전송되는 패킷이나 순서가 같았다. 데이터 전달과정은 그림 1 과 같다. 먼저 TCP 3way Handshake 를 사용하여 세션이 성립되고 TLSv1 을 사용한 TLS handshake 가 이루어 지면서 안전한 통신기반을 구축한다. 암호화된 데이터를 주고 받으면서 데이터를 통신하고 FIN, ACK 을 통해 세션을 종료시킨다. 이처럼 전송되는 패킷들의 순서를 볼 수 있지만 전송되는 데이터의 세부정보는 TLS 를 통해 암호화되어 있어서 알 수가 없다.



(그림 2) Wi-Fi, 3G 환경에서 사진 올리기

이어 같은 실험 환경에서 페이스북 담벼락에 99KB 의 사진을 전송하여 분석 하였다. 3G, Wi-Fi 두 환경 모두 동일한 과정이 나타났다. 사진을 전송할 때는 2 개의 서버와 연속적으로 통신을 하는데 그림 2 는 첫 번째 서버와 통신하는 것을 보여준다. 앞의 실험과 같이 세션을 성립하고 TLS handshake 후 데이터 전송을 하고 세션을 종료한다. 이 데이터 전송이 끝나면 앞에서 실험한 글을 쓰는 서버에 접속하여 데이터를 전송되는 것을 볼 수 있었다. 하지만 암호화로 인해 전송되는 데이터의 세부정보는 알 수 없었다.

글쓰기와 사진을 전송하는 과정은 모두 암호화되어 있어 전송되는 데이터의 내용을 알 수 없었다. 그러므로 확실한 프로토콜 확인을 하기 위해 페이스북 Open Graph API 과 Facebook SDK 2.0 를 이용하여 어플리케이션을 직접 작성하여 글과 사진 쓰기 과정에서 발생하는 결과를 이용해 프로토콜들을 분석한다.

3. 공개 API 와 SDK 2.0 를 이용한 추론

3.1 HTTP Request Message Format

SDK 를 통해 페이스북을 살펴보면 페이스북에서는 담벼락에 글이나 사진을 전송할 때 HTTP 를 사용한다는 것을 알 수 있다. HTTP messages 는 클라이언트에서 서버로 요청하는 Request Message 와 서버에서 클라이언트로 응답 오는 Response Message 로 나뉜다[9]. HTTP 의 Request Message 형식은 Request line, General Headers, Request Headers, Entity Headers, Empty Line, Message Body 로 이루어져 있다.

각 라인을 간략하게 설명하면 Request Line 은 세 가지의 목적을 가진다. 클라이언트가 원하는 명령이나 작업을 지정하고, 서버로부터 작업이 수행되어야 할 자원을 지정하고, 클라이언트가 사용하는 HTTP 의 버전을 나타낸다(<method>, <request URI>, <HTTP version>)[10]. General Headers 는 클라이언트 요청과 서버 응답을 할 때 모두 쓰이는데 메시지 처리를 제어하거나 수신자에게 정보 제공에 사용된다. Request Headers 에서는 클라이언트의 요청에 대해 자세한 것들을 서버로 전달하고 클라이언트의 요청을 처리하는 방법을 제어한다. Entity Headers 는 메시지를 통해 전달되는 내용을 설명한다.

3.2 Graph API

페이스북은 현재 Graph API 를 사용한다. Graph API 의 특징은 사용자, 담벼락 글, 댓글, 좋아요 등 각각의 컨텐츠를 하나의 개체로 정의한다[2]. 또한 개체들을 검색, 삭제, 게시, 업데이트를 하는데도 이용된다. 각각의 오브젝트들은 자신만의 고유한 개체 ID 를 가지고 있고 https://graph.facebook.com/<페이스북 ID> 을 통해서 개체의 정보에 대해 질의를 하면 페이스북 ID 의 개체 정보를 응답 받을 수 있다[4]. 예를 들어 필자의 페이스북 ID 로 https://graph.facebook.com/insikchenje 를 요청하게 되면 표 1 과 같은 JSON(Java Script Object Notation)형식[6]의 응답을 할 것이다. 물론 자세한 개체 정보들을 알기 위해서는 페이스북의

ACCESS_TOKEN 을 발급받아야 한다. 페이스북의 각 컨텐츠를 사용하기 위해서는 각각의 권한이 필요하다. 페이스북 어플리케이션을 사용할 때 그 어플리케이션에서 사용할 컨텐츠들에대한

<표 1> Response 되는 JSON 형식의 ID 정보(권한 미포함)

```
{
 "id": "100003300528513",
 "name": "정인식",
 "first_name": "인식",
 "last_name": "정",
 "username": "insikchenje",
 "gender": "male",
 "locale": "ko_KR"
}
```

<표 2> Response 되는 JSON 형식의 ID 정보(권한 포함)

```
{
 "id": "100003300528513",
 "name": "정인식",
 "first_name": "인식",
 "last_name": "정",
 "link": "http://www.facebook.com/insikchenje",
 "username": "insikchenje",
 "birthday": "01/03/1988",
 "gender": "male",
 "email": "insikchenje@hanmail.net",
 "timezone": 9,
 "locale": "ko_KR",
 "verified": true,
 "updated_time": "2012-08-07T02:11:48+0000"
}
```

<표 3> Response 되는 JSON 형식의 친구 정보(권한 포함)

```
{
 "id": "100003300528513",
 "name": "정인식",
 "first_name": "인식",
 "last_name": "정",
 "link": "http://www.facebook.com/insikchenje",
 "username": "insikchenje",
 "birthday": "01/03/1988",
 "gender": "male",
 "email": "insikchenje@hanmail.net",
 "timezone": 9,
 "locale": "ko_KR",
 "verified": true,
 "updated_time": "2012-08-07T02:11:48+0000"
}
```

허가가 있어야 데이터를 가져올 수 있다. 만약 자신의 정보를 요청 할 수 있는 user_data_permmision 권한을 얻고 나면 표 2 와 같은 더욱 자세한 정보를 알 수 있다[7].

또 개체들을 연결시켜주고 그 관계를 사용하는 것이 페이스북 Graph API 의 특징인데 https://graph.facebook.com/<ID>/<CONNECTION_TYPE>?access_token=<ACCESS_TOKEN>으로 원하는 관계에 있는 개체들을 확인할 수 있다. CONNECTION_TYPE 은 friend list, likes 와 같은 관계를 ACCESS_TOKEN 는 로그인시 발급받아 어플리케이션에서 사용할 수 있는 권한을 의미한다. 만약 사용자의 친구를 보고 싶다면 http://graph.facebook.com/<ID>/friends>?access_token=<ACCESS_TOKEN>를 하게 되면 표 3 과 같이 볼 수 있다. 이처럼 페이스북의 개체들은 관계로 표현된다.

3.3 Facebook SDK 2.0

페이스북의 정확한 프로토콜을 알기 위해서 Graph API 정보를 기초로 페이스북 SDK 2.0 를 사용하여 페이스북 어플리케이션을 만들어 분석하였다. 만든 어플리케이션에는 페이스북의 로그인, 담벼락에 글쓰기, 담벼락에 사진 올리기, 로그아웃 기능을 수행할 수 있다. 직접 작성한 어플리케이션을 페이스북 Developer 에 등록하고 고유의 APP_ID 를 받아서 사용하면 만든 어플리케이션이 정상적인 동작이 가능하다.

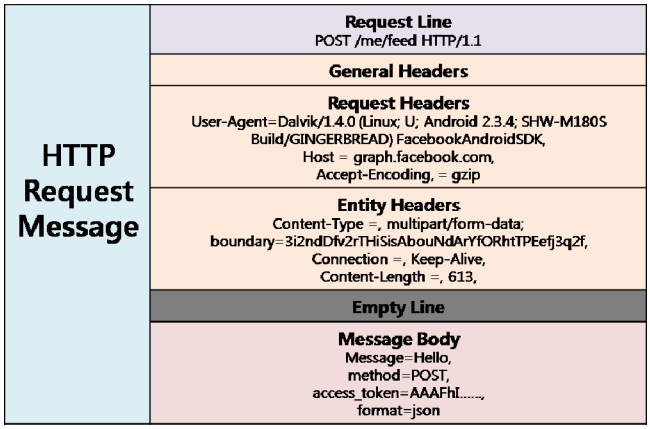
분석을 하는 과정은 모바일 단말에서 떠난 패킷은 암호화 되어서 전송되기 때문에 SDK 2.0 의 각각의 메소드에서 중단점을 걸어 변수 값을 보거나 로그를 남겨서 분석한다.

먼저 담벼락에 글을 쓸 때 전송되는 데이터에 대해 분석 하였다. 각각의 메소드에서 추가되는 정보들은 andoird.os 의 Bundle 타입의 변수에 저장된다. Bundle 타입은 Key 와 Value 를 가지는 타입이다 ([Key, Value]) [8]. 먼저 전송되기를 원하는 데이터를 생성한 Bundle 변수에 ([“message”, 전송될 글])을 저장한다. 그 후 페이스북 SDK 2.0 메소드 중 하나인 Facebook.java 에서 Request 메소드를 통해서 ([“format”, “json”]) 와 로그인 과정에서 얻은 ACCESS_TOKEN ([“access_token”, ACCESS TOKEN])을 Bundle 변수에 저장한다.

그 후 페이스북의 SDK 2.0 인 Util.java 의 메소드 openUrl 에서 ([“method”, 전송방법(예: post, get)])을 저장한다. 이런 과정을 거치고 나면 표 4 와 같은 데이터가 Message Body 에 저장 된다.

<표 4> Message Body 에 저장되는 데이터

```
message=hello,
method=POST,
access_token= AAAFhI…..,
format=json
```



(그림 3) 페이스북 데이터 포맷

동시에 전송되는 HTTP header 옵션들도 저장이 되는데 HTTP header 옵션은 페이스북 SDK 2.0 의 Util.java 에서 모두 결정된다. 처음 대상 시스템에서 호출한 User-Agent 를 저장한다. 그리고 Content-Type 을 multipart/form-data 로 지정하고 Connection 을 Keep-Alive 로 설정한다. 다른 헤더 옵션은 httpconnection 을 연결할 때 설정된다. 쓰기 과정 중 이클립스 중단점을 설정하고 디버깅을 해서 httpsEngine-RequestHeader-props 의 값을 살펴보면 완성된 Request Headers 를 확인할 수 있다. Request Headers 에서 알아낸 정보와 위에서 알아본 Message Body 를 HTTP Request Message 형식에 적용을 시켜보면 그림 3 과 같은 형식이 완성된다.

사진을 전송할 때는 Request URI 를/me/photos 로 해서 전송 하는 것 외에는 큰 차이가 없다.

4. 결론 및 향후 연구.

본 논문에서는 페이스북 API 와 SDK 2.0 를 이용하여 페이스북 프로토콜을 분석하는 방법을 제안하고 분석한 결과를 나타냈다. HTTP 를 사용하며 전송 할 때 어떠한 형식으로 전송 되는지 알아볼 수 있었다. 제안한 방법으로 댓글이나 좋아요 등 추가적인 기능들을 유추해 낼 수 있을 것이다. 뿐만 아니라 여러 플랫폼에서 페이스북의 주요 기능들을 사용하는데 도움이 될 것이다.

향후, 캐싱 기능이 있는 게이트웨이 서버를 구성할 때 알아낸 프로토콜을 사용할 수 있다. 게이트웨이 서버는 페이스북 서버와 모바일 단말 사이에 두고 게이트웨이 서버에서 페이스북 서버에 데이터를 전송할 때 위 실험에서 알아낸 헤더 정보들을 이용할 것이다.

Acknowledgement

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임(2012H1B8A2025942) *교신저자 : 주홍택

참고문헌

[1] Han Jing, Geng Qingjun, "Prospect Application in the Library of SNS," 2011 Third Pacific-Asia Conference, 2011/07/17 – 2011/07/18

[2] Graph Api, http://developers.facebook.com/docs/reference/api/, 2012/08/07

[3] Jeff Bullas, Facebook Approaches 500 Million Mobile Users – Inforgraphic, http://www.jeffbullas.com/2012/05/18/facebook-approaches-500-million-mobile-users-infographic/, 2012/05/18

[4] Muller, F., Thiesing, F., Computational Aspects of Social Networks (CASoN), Social Networking APIs for companies, 2011/10/19 – 2011/10/21

[5] Experian Hitwise US, Social Media Trends, http://www.experian.com/hitwise/online-trends-social-media.html, 2012/10/6

[6] Introducing Json, http://www.json.org, 2012/08/04

[7] Graph API Explorer , http://developers.facebook.com/tools/explorer, 2012/08/07

[8] Introducing Bundle Class, http://developer.android.com/reference/android/os/Bundle.html, 2012/08/01

[9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, HTTP Message, RFC 2616, http://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html#sec4, 2012/09/12

[10] Charles M. Kozierok, HTTP Request Format, http://www.tcpipguide.com/free/t_HTTPRequestMessageFormat-2.htm, 2012/09/12