carpedm20

Machine Learning/Data Visualization/Computer Security

2013년 9월 12일 목요일

LINE(라인) protocol analysis [1] : let's send a message

*** 2014.08.02 update ***

If you need a python code right away, then please keep in touch with https://github.com/carpedm20/LINE

After I analyzed *LOCO protocol* of KakaoTalk, I've been curious about the operation of other messaging applications.

Like KakaoTalk, LINE is the instant messaging application on smartphones and PCs.

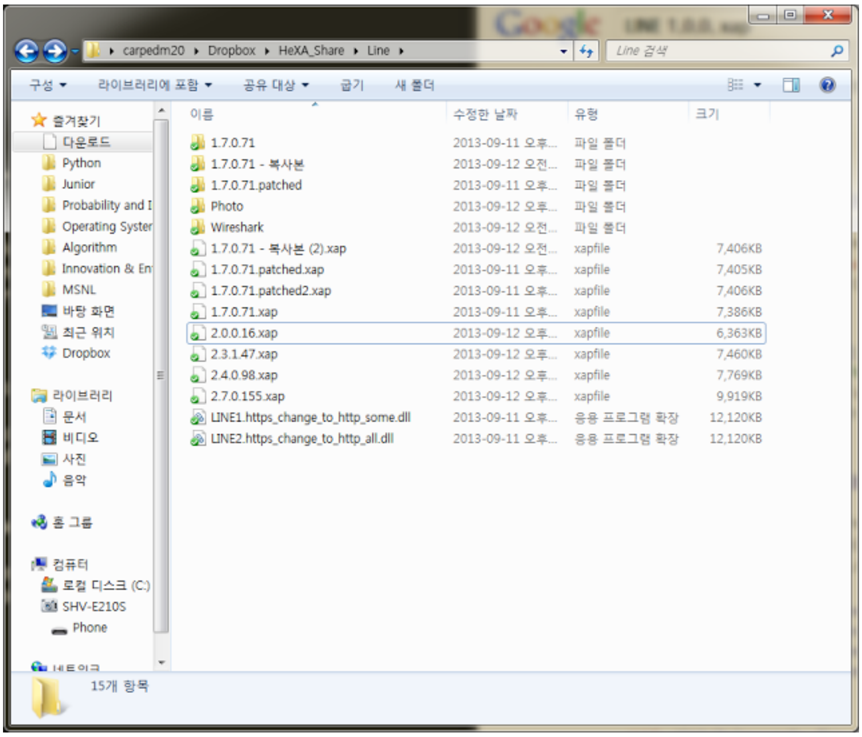
LINE is not popular in Korea, but media currently said that LINE is one of the most popular messaging app in Japan.

So, I decided to analyze the protocol of LINE and I'll record the steps that I followed in this post.

My final goal is to implement the LINE protocol in python, especially sending and receiving messages.

1. Download xap file

First of all, I needed a xap file of LINE windows mobile application, so I searched it on Google.



이름	수정한 날짜	유형	크기
1.7.0.71	2013-09-11 오후...	파일 폴더	
1.7.0.71 - 복사본	2013-09-12 오전...	파일 폴더	
1.7.0.71.patched	2013-09-11 오후...	파일 폴더	
Photo	2013-09-12 오후...	파일 폴더	
Wireshark	2013-09-12 오전...	파일 폴더	
1.7.0.71 - 복사본 (2).xap	2013-09-12 오전...	xapfile	7,406KB
1.7.0.71.patched.xap	2013-09-11 오후...	xapfile	7,405KB
1.7.0.71.patched2.xap	2013-09-11 오후...	xapfile	7,406KB
1.7.0.71.xap	2013-09-11 오후...	xapfile	7,386KB
2.0.0.16.xap	2013-09-12 오후...	xapfile	6,363KB
2.3.1.47.xap	2013-09-12 오후...	xapfile	7,460KB
2.4.0.98.xap	2013-09-12 오후...	xapfile	7,769KB
2.7.0.155.xap	2013-09-12 오후...	xapfile	9,919KB
LINE1.https_change_to_http_some.dll	2013-09-11 오후...	응용 프로그램 확장	12,120KB
LINE2.https_change_to_http_all.dll	2013-09-11 오후...	응용 프로그램 확장	12,120KB

Finally, I found the old version of LINE xap file (version : 1.7.0.71) :)

About me

http://carpedm20.github.io/

HeXA

블로그 검색

검색

블로그 보관함

- ► 2015 (4)
- ► 2014 (17)
- ▼ 2013 (44)
  - ► 12월 (4)
  - ► 11월 (4)
  - ► 10월 (2)
  - ▼ 9월 (4)
    - LINE(라인) protocol analysis [4] : Registration
    - LINE(라인) protocol analysis [3] : session key
    - LINE(라인) protocol analysis [2] : HTTP(S) data
    - LINE(라인) protocol analysis [1] : let's send a mess...
  - ► 8월 (5)
  - ► 7월 (7)
  - ► 5월 (6)
  - ► 4월 (11)
  - ► 3월 (1)
- ► 2012 (30)

가장 많이 본 글

- 해킹 공부 자료
- [카카오톡] 헥사봇
- 해킹의 분류 [1] - 시스템 해킹
- 해킹의 분류 [4] - 무선랜 해킹
- [C#] HttpWebRequest 로 http GET, POST 전송 및 처리
- 해킹의 분류 [3] - 리버싱
- LINE(라인) protocol analysis [1] : let's send a message
- 스타크래프트 시디키 알고리즘 분석 [1]
- [python] Scrapy - 네이버 영화 파싱
- 해킹의 분류 [2] - 네트워크 해킹

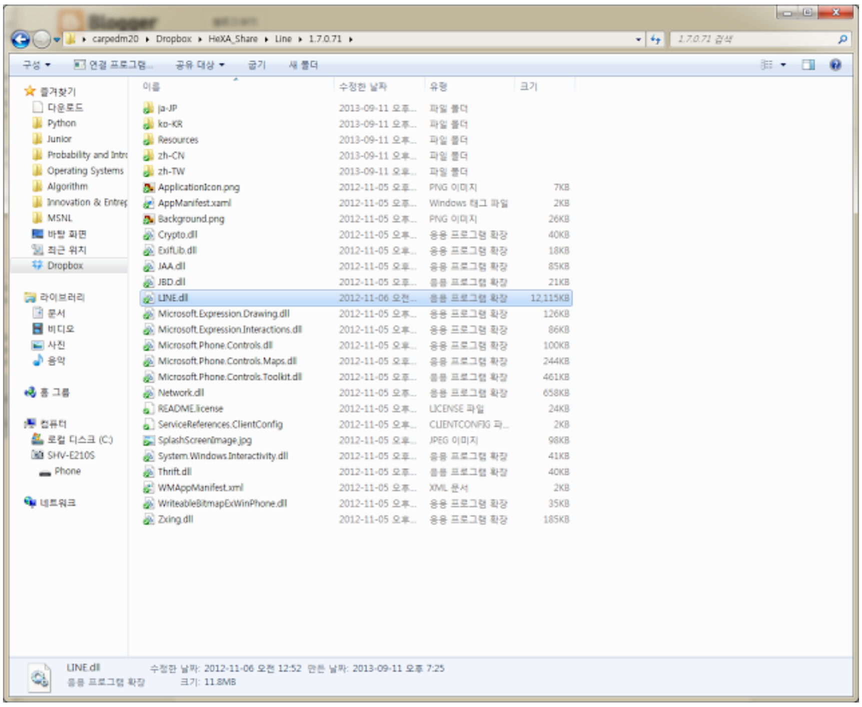
태그

python github 소스 예제 정리 리버싱 product 윈도우 시스템 프로그래밍 어셈블리 프로젝트 디버깅 백트랙 C# Django LINE 메타스플로잇 악성코드 web 해킹 flask javascript 명령어 카카오톡 android ctf php visualization 공부 네트워크 팁 Network VMware Wireshark angularjs between bot chrome hacking html java jquery movie parsing pcap ppt scrapy smali twitter ubuntu ultrasurf watcha write-up 개발 레지스터 무비덕 무선랜 서비스 설정 시스템 프로그래밍 시스템 코드 크롬 확장 프로그램

방문자

The latest version of windows LINE application is 2.7.0.155

2. Unzip xap file



이름	수정한 날짜	유형	크기
ja-JP	2013-09-11 오후...	파일 폴더	
ko-KR	2013-09-11 오후...	파일 폴더	
Resources	2013-09-11 오후...	파일 폴더	
zh-CN	2013-09-11 오후...	파일 폴더	
zh-TW	2013-09-11 오후...	파일 폴더	
ApplicationIcon.png	2012-11-05 오후...	PNG 이미지	7KB
AppManifest.xaml	2012-11-05 오후...	Windows 태그 파일	2KB
Background.png	2012-11-05 오후...	PNG 이미지	26KB
Crypto.dll	2012-11-05 오후...	응용 프로그램 확장	40KB
ExifLib.dll	2012-11-05 오후...	응용 프로그램 확장	18KB
JAA.dll	2012-11-05 오후...	응용 프로그램 확장	85KB
JBD.dll	2012-11-05 오후...	응용 프로그램 확장	21KB
LINE.dll	2012-11-06 오전...	응용 프로그램 확장	12,115KB
Microsoft.Expression.Drawing.dll	2012-11-05 오후...	응용 프로그램 확장	126KB
Microsoft.Expression.Interactions.dll	2012-11-05 오후...	응용 프로그램 확장	86KB
Microsoft.Phone.Controls.dll	2012-11-05 오후...	응용 프로그램 확장	100KB
Microsoft.Phone.Controls.Maps.dll	2012-11-05 오후...	응용 프로그램 확장	244KB
Microsoft.Phone.Controls.Toolkit.dll	2012-11-05 오후...	응용 프로그램 확장	461KB
Network.dll	2012-11-05 오후...	응용 프로그램 확장	658KB
README.license	2012-11-05 오후...	LICENSE 파일	24KB
ServiceReferences.ClientConfig	2012-11-05 오후...	CLIENTCONFIG 파...	2KB
SplashScreenImage.jpg	2012-11-05 오후...	JPEG 이미지	98KB
System.Windows.Interactivity.dll	2012-11-05 오후...	응용 프로그램 확장	41KB
Thrift.dll	2012-11-05 오후...	응용 프로그램 확장	40KB
WMAppManifest.xml	2012-11-05 오후...	XML 문서	2KB
WriteableBitmapExWinPhone.dll	2012-11-05 오후...	응용 프로그램 확장	35KB
Zxing.dll	2012-11-05 오후...	응용 프로그램 확장	185KB

LINE.dll 수정한 날짜: 2012-11-06 오전 12:52 만든 날짜: 2013-09-11 오후 7:25
응용 프로그램 확장 크기: 11.8MB

The first thing that attracted me was 'Line.dll' file and I guessed it may have core functions for the chat protocol.

And also, I could see 'Thrift.dll' which is the library for Thrift framework.

After I searched Google for a moment, I found that Thrift is an open source project for cross-language service built by Apache.

Now, I knew LINE uses Thrift library for network communication, which is not their own protocol, so I thought it might be easy to implement LINE chat system (compare to LOCO protocol...).

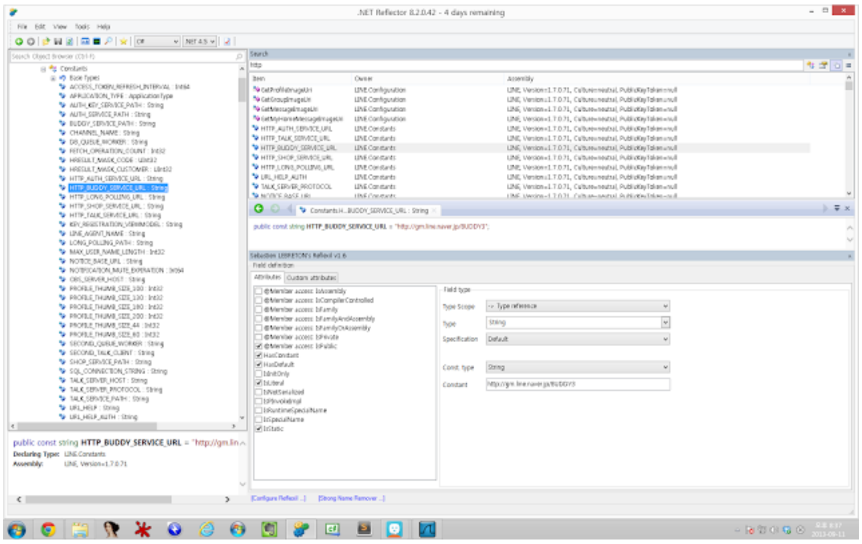
3. Packet Analysis

Before I did the static analysis, I used **windows mobile phone emulator** for the packet analysis.

Of course, the network between application and server was encrypted using **https**.

There were some packets which seem to be TCP protocol but I focused on the HTTP communication.

After looked over the packet, I used **.Net reflector** to see the real decompiled source code of applications.



```
Constants
  Base Types
  ACCESS_TOKEN_REFRESH_INTERVAL : Int64
  APPLICATION_TYPE : ApplicationType
  AUTH_KEY_SERVICE_PATH : String
  AUTH_SERVICE_PATH : String
  BUDDY_SERVICE_PATH : String
  CHANNEL_NAME : String
  DB_QUEUE_WORKER : String
  FETCH_OPERATION_COUNT : Int32
  HRESULT_MASK_CODE : UInt32
  HRESULT_MASK_CUSTOMER : UInt32
  HTTP_AUTH_SERVICE_URL : String
  HTTP_BUDDY_SERVICE_URL : String
  HTTP_LONG_POLLING_URL : String
  HTTP_SHOP_SERVICE_URL : String
  HTTP_TALK_SERVICE_URL : String
  KEY_REGISTRATION_VERINMODEL : String
  LINE_AGENT_NAME : String
  LONG_POLLING_PATH : String
  MAX_USER_NAME_LENGTH : Int32
  NOTICE_BASE_URL : String
  NOTIFICATION_MUTE_EXPIRATION : Int64
  OBS_SERVER_HOST : String
  PROFILE_THUMB_SIZE_100 : Int32
  PROFILE_THUMB_SIZE_120 : Int32
  PROFILE_THUMB_SIZE_180 : Int32
  PROFILE_THUMB_SIZE_200 : Int32
  PROFILE_THUMB_SIZE_44 : Int32
  PROFILE_THUMB_SIZE_60 : Int32
  SECOND_QUEUE_WORKER : String
  SECOND_TALK_CLIENT : String
  SHOP_SERVICE_PATH : String
  SQL_CONNECTION_STRING : String
  TALK_SERVER_HOST : String
  TALK_SERVER_PROTOCOL : String
  TALK_SERVICE_PATH : String
  URL_HELP : String
  URL_HELP_AUTH : String
```

```
public const string HTTP_BUDDY_SERVICE_URL = "http://gm.lin...
Declaring Type: LINE.Constants
Assembly:       LINE, Version=1.7.0.71
```

Search: http

Item	Owner	Assembly
GetProfileImageUrl	LINE.Configuration	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
GetGroupImageUrl	LINE.Configuration	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
GetMessageImageUrl	LINE.Configuration	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
GetMyHomeMessageImageUrl	LINE.Configuration	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
HTTP_AUTH_SERVICE_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
HTTP_TALK_SERVICE_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
HTTP_BUDDY_SERVICE_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
HTTP_SHOP_SERVICE_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
HTTP_LONG_POLLING_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
URL_HELP_AUTH	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
TALK_SERVER_PROTOCOL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null
NOTICE_BASE_URL	LINE.Constants	LINE, Version=1.7.0.71, Culture=neutral, PublicKeyToken=null

Constants.H...BUDDY_SERVICE_URL : String

```
public const string HTTP_BUDDY_SERVICE_URL = "http://gm.line.naver.jp/BUDDY3";
```

Field type
Type Scope: -> Type reference
Type: String
Specification: Default
Const. type: String
Constant: http://gm.line.naver.jp/BUDDY3

□ @Member access: IsAssembly
□ @Member access: IsCompilerControlled
□ @Member access: IsFamily
□ @Member access: IsFamilyAndAssembly
□ @Member access: IsFamilyOrAssembly
□ @Member access: IsPrivate
☑ @Member access: IsPublic
☑ HasConstant
☑ HasDefault
□ IsInitOnly
☑ IsLiteral
□ IsNotSerialized
□ IsPInvokeImpl
□ IsRuntimeSpecialName
□ IsSpecialName
☑ IsStatic

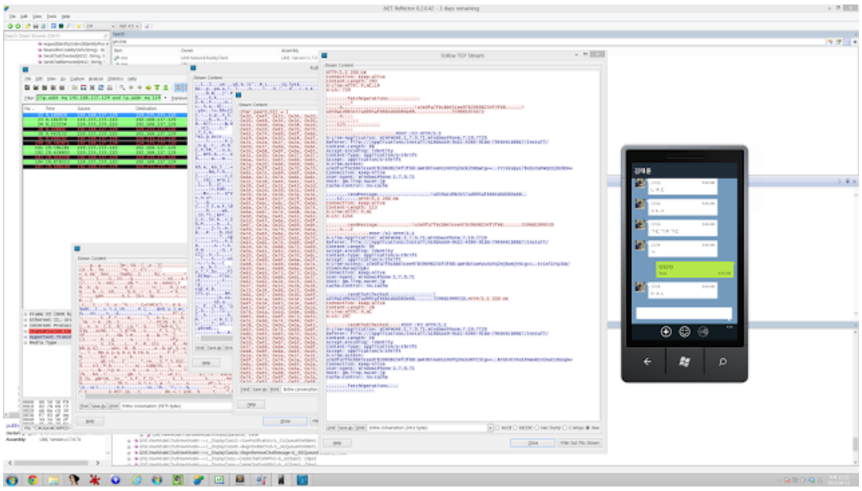
I searched 'https' as a string, changed them to 'http', and re-zipped the xap file.

At this point, I found out that the DNS of main server for chat communication was 'gm.line.naver.jp'

Especially, '**gm.line.naver.jp/S3**' is used for authorization and chat service for LINE.

# http://gm.line.naver.jp/api/v3/TalkService.do for talkSession

Then, I could see the plain chat communication between server and client in the packets.



I'm not sure that HTTP is LINE's main protocol, because LOCO protocol of KakaoTalk used their own packet structure which was encrypted with AES.

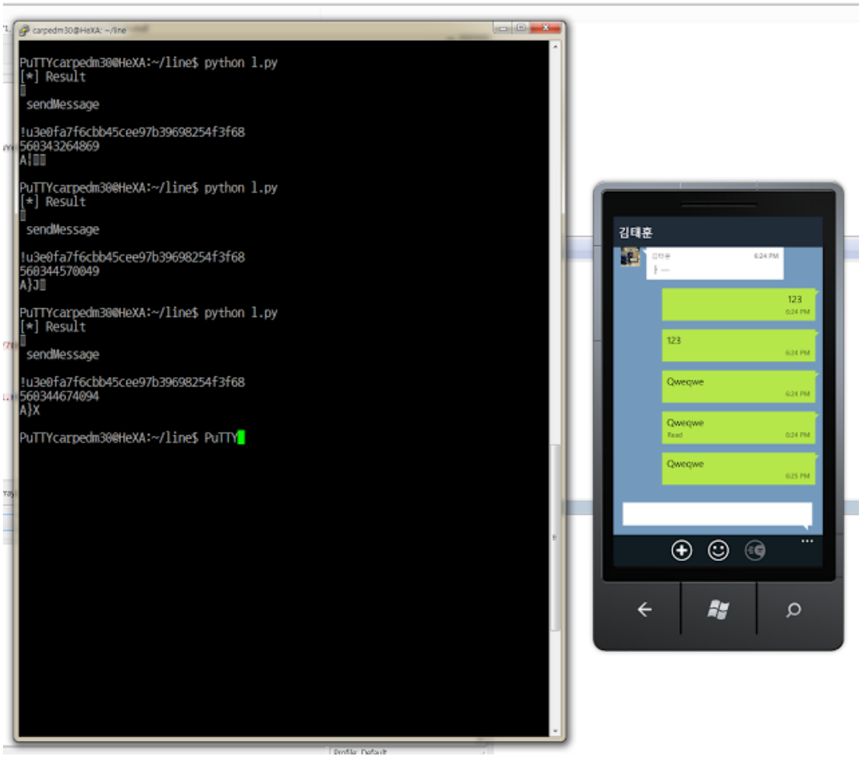
As you can see, LINE doesn't encrypt any messages, so I can see the **plain message** from packet.

Also, '**X-Line-Access**', which was included in the header, seems like a **session key**, so I was wonder whether the previous session can be used for communication or not.

So I quickly wrote a dirty python code which send the exactly same packet to the server...

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
__author__ = 'carpedm20'

import urllib2

def send():
    url = 'http://gm.line.naver.jp/S3'
    headers = { 'POST' : '/S3',
        'X-Line-Application' : 'WINPHONE.1.7.0.71.WindowsPhone.7.10.7720',
        'Referer' : 'file:///Applications/Install/???/Install/',
        'Accept-Encoding' : 'identity',
        'Content-Type' : 'application/x-thrift',
        'Accept' : 'application/x-thrift',
        'X-Line-Access' : '???',
        'Connection' : 'Keep-Alive',
        'User-Agent' : 'WindowsPhone 1.7.0.71',
        'HOST' : 'gm.line.naver.jp',
        'Cache-Control' : 'no-cache'}

data='\x80\x01\x00\x01\x00\x00\x00\x0b\x73\x65\x6e\x64\x4d\x65\x73\x73\x61\x67\x65\x00\x00\x00\x00\x08\x00\x01\x00\x00\x00\x00\x0c\x00\x02\x0b\x00\x02\x00\x00\x00\x21\x75\x30\x33\x39\x61\x31\x64\x39\x62\x33\x34\x35\x37\x61\x64\x39\x39\x35\x61\x66\x36\x36\x62\x34\x64\x64\x64\x30\x38\x30\x65\x36\x38\x0b\x00\x0a\x00\x00\x00\x06\x51\x77\x65\x71\x77\x65\x02\x00\x0e\x00\x00\x00'
    request = urllib2.Request(url, data, headers)
    response = urllib2.urlopen(request)

    print "[*] Result "
    data = response.read()
    print data
    #data = json.loads(data ,encoding='utf-8')

send()
```

It worked pretty well!



```
PuTTYcarpedm30@HeXA:~/line$ python 1.py
[*] Result
▯
 sendMessage

!u3e0fa7f6cbb45cee97b39698254f3f68
560343264869
A¦▯▯

PuTTYcarpedm30@HeXA:~/line$ python 1.py
[*] Result
▯
 sendMessage

!u3e0fa7f6cbb45cee97b39698254f3f68
560344570049
A}J▯

PuTTYcarpedm30@HeXA:~/line$ python 1.py
[*] Result
▯
 sendMessage

!u3e0fa7f6cbb45cee97b39698254f3f68
560344674094
A}X

PuTTYcarpedm30@HeXA:~/line$ PuTTY
```

김태훈

6:24 PM

123 6:24 PM

123 6:24 PM

Qweqwe 6:24 PM

Qweqwe Read 6:24 PM

Qweqwe 6:25 PM

Next : LINE protocol analysis [2] : HTTP(S) data

Google에 이 URL 추천

라벨: LINE, python, 리버싱

댓글 없음:

댓글 쓰기

댓글을 입력하세요...

작성자 ggyy (Google) 로그아웃

게시 미리보기 □ 알림

최근 게시물 홈 이전 게시물

피드 구독하기: 댓글 (Atom)

Blogger 제공.