情報保護學會論文誌
第 21 卷 第 2 號, 2011. 4

인터넷 메신저의 통신내역 수집기술*

이 진 경,† 한 지 성, 이 상 진,‡
고려대학교 정보경영공학전문대학원

Method to Extract Communication History in Instant Messenger*

Jinkyung Lee,† Jisung Han, Sangjin Lee‡
Graduate School of Information Management and Security, Korea University

요 약

인터넷 메신저는 네트워크를 통해 대화를 나누거나 데이터 등을 주고받을 때 가장 널리 쓰이는 통신수단 중 하나이다. 그러므로 디지털 포렌식 관점에서 메신저의 통신내역을 확보하는 일은 전화, 휴대폰과 같이 전통적인 통신내역을 수집하는 작업처럼 매우 중요하다. 하지만 메신저의 통신내역은 사용자 컴퓨터에 암호화 되어 저장되거나 메신저 서버에 보존되기 때문에 의미 있는 데이터 수집이 쉽지 않다. 본 논문은 네이트온, 버디버디, 야후!, Mi3 메신저를 대상으로 사용자 컴퓨터에 저장된 통신내역을 복구하는 방법과 메신저 서버에 존재하는 통신내역을 열람하기 위한 인증우회기법을 제시한다.

ABSTRACT

Instant Messenger is one of the most popular communication service when translating message or data each other through Internet. For digital crime investigation, therefore, it is obviously important to obtain communication trace and contents derived from Instant Messenger. This is because that gathering traditional communication histories also have been important until now. However, extracting communication trace and contents are not easy because they are generally encrypted or obfuscated in local system, futhermore, sometimes they are located at server computer for Instant Messenger. This paper researches on extracting communication histories against NateOn, BuddyBuddy, Yahoo! messenger and Mi3 messenger, and obtaining user password or bypassing authentication system to Instant Messenger Service when a user use auto-login option.

Keywords: Digital Forensics, Instant Messenger

I. 서 론

사건수사 전반에 걸쳐서 수사대상의 통신내역을 확보하고 조사하는 일은 매우 중요하다. 수사관은 이런 통신흔적으로부터 용의자의 사건에 대한 의도나 생각을 알아낼 수 있으며, 이는 곧 수사를 진척시킬 수 있는 결정적인 정보로 사용된다. 전통적인 수사관점에서 통신내역은 전화라는 통신매체에 국한되었지만, 인터넷이 널리 보급된 현대사회에서는 인터넷을 통한 정보전달 수단을 고려하지 않을 수 없다. 인터넷 통신수단은 과거 이메일, 게시판, 채팅채널 등으로부터 인터넷 메신저, 인터넷 전화 등의 실시간성과 편리성을 함께 제공하는 고급 통신수단으로 발전하고 있다.

인터넷 메신저는 네트워크를 통해 대화 및 데이터 등을 실시간으로 주고받는 프로그램으로서 현재 이메일과 함께 가장 많이 사용되는 인터넷 통신수단이다

접수일(2010년 7월 13일), 수정일(2010년 11월 1일), 게재확정일(2010년 11월 5일)
* 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발 사업의 일환으로 수행하였음 [10035157, 실시간 분석을 위한 디지털 포렌식 기술 개발]
† 주저자, neobug@gmail.com
‡ 교신저자, sangjin@korea.ac.kr

[1]. 이메일은 정보전달에 있어서 상대방의 즉각적인 반응을 기대하기 어려운 반면 인터넷 메신저는 전화통화와 같은 실시간 정보통신을 가능하게 해준다. 비록 상호간의 통신을 위해서 같은 메신저 서비스를 이용해야 하지만, 이 문제는 특정 메신저 서비스들이 시장점유율을 장악함으로서 자연스럽게 해결되었다. 결과적으로 인터넷 메신저는 전화나 휴대폰과 같이 보편적인 통신수단의 하나가 되었다. 따라서 과거와는 달리 인터넷 메신저의 통신내역을 수집하는 기술이 요구된다.

디지털 포렌식 관점에서 인터넷 메신저의 통신내역을 확보하는 기술은 다양하다. 파일로 저장된 대화·쪽지·파일송수신 내역 등을 획득하여 분석하는 방법[2][3][4], 파일시스템의 비할당 영역에 존재하는 삭제된 통신 내역을 복구하는 방법[5], 아직 메모리에 남아있을 메신저 관련 데이터 등을 추출하는 방법 등[5] 수사에 도움이 되는 증거를 획득하기 위해서 여러 가지 방법이 제안되었다. 하지만 지금까지 제안된 통신내역 확보기술은 주로 해외에서 많이 사용되는 메신저를 대상으로 연구되었기 때문에 국내 인터넷 메신저로부터 통신내역을 확보하기에는 어려움이 따른다.

본 논문은 국내에서 인기 있는 메신저 서비스를 대상으로 통신내역 확보기술을 제시하고 있다. 먼저 인터넷 메신저의 통신내역은 서비스마다 저장위치와 방식이 다르기 때문에 이를 정리하였다. 또한 통신내용을 안전하게 저장하기 위한 암호화 과정은 서비스별로 알고리즘이 다르기 때문에 각각의 과정들을 분석하였다. 그리고 통신내역이 사용자 컴퓨터가 아닌 메신저 서버에 저장되는 경우를 고려해 서버로의 인증우회 기법을 연구하였다. 논문에서 다루는 메신저 서비스들은 네이트온(version 4.0), 버디버디(version 7), 야후!(version 8.1) 그리고 Mi3(version 5)이다.

II. 관련 연구

디지털 포렌식 관점에서 인터넷 메신저에 대한 연구는 2006년을 기점으로 활발히 진행되기 시작했다. Mike Dickson은 AOL사의 AIM 메신저, Microsoft사의 MSN(현재의 Windows Live), 그리고 야후! 메신저를 대상으로 디지털 포렌식 관점에서 의미있는 데이터들을 분류하고 획득하는 방법을 제시하였다[2][3][4]. AIM 메신저의 경우 전 세계적으로 1억 명 이상의 가입자를 보유하고 있는 업계 최고의 인기 메신저 서비스이며 MSN과 야후! 메신저 역시 이에 못지않은 규모의 시장을 가지고 있다[5].

Jessica Reust는 실제 수사 과정에서 AIM 메신저의 통신내역 및 친구목록 등을 확보하는 과정을 통해 사건의 실마리를 찾아내는 사례를 발표하였다[6]. 그 전까지의 연구 결과들이 프로그램 분석을 통해 의미 있는 데이터 수집 방법론을 제시했다면 이 논문은 실제 수사과정에서 지금까지 발표되었던 방법들이 어떻게 사용되었는지에 대해서 서술하고 있다.

2007년 Microsoft사는 MSN 메신저를 대대적으로 업데이트함으로서 기능이 개선된 Windows Live라는 메신저를 발표하였다. 버전 상으로는 MSN 메신저를 계승하고 있지만 MSN 7.5 이하 버전과 8.0 이상 버전은 다른 프로그램이라고 생각할 수 있을 정도로 큰 차이를 보이고 있다. Wouter S. van Dongen은 새로운 Windows Live 메신저를 대상으로 디지털 포렌식 관점에서 의미 있는 정보들을 분류하였다. 그리고 메신저의 통신내역 등을 확보하기 위해 파일시스템의 비할당영역과 메모리영역까지 조사하는 연구를 발표하였다[5].

현재까지 발표된 인터넷 메신저 관련 논문은 대부분 글로벌 메신저를 대상으로 하고 있다. 분명 AIM 메신저나 야후! 메신저는 세계적인 관점에서 봤을 때는 가장 많이 사용되고 있는 서비스지만 국내에서는 상황이 다르다. 한국인터넷진흥원의 통계에 따르면 2008년 기준 네이트온 메신저의 이용자 수는 2000만 명을 상회하고 있다. 반면 글로벌 메신저의 선두격인 Windows Live는 700만 명의 사용자만을 확보하고 있을 뿐이다[7]. 그러므로 국내 상황에 맞는 메신저 서비스에 대한 연구가 수행되어야만 한다.

III. 통신내역 수집방안

대부분의 인터넷 메신저는 통신내역을 사용자 컴퓨터에 저장한다. 하지만 일부 메신저 서비스는 통신내역을 서버에만 저장하거나 혹은 양쪽 모두에 남겨놓기도 한다. 야후!, Mi3 메신저가 전자에 해당하며 네이트온, 버디버디가 후자에 해당한다. 다만 버디버디는 통신내역을 사용자 컴퓨터에 저장함과 동시에 그 일부인 쪽지내역만은 서버에 보존한다. 이처럼 통신내역을 서버에 저장하면 사용자는 어디에서든지 자신의 통신내역을 열람할 수 있게 되며, 이는 서비스 관점에서 봤을 때 강점이기 때문에 많은 메신저 서비스들이 지향할 가능성이 높다. 따라서 메신저의 통신내역을 확보하기 위한 방안은 사용자 컴퓨터에 저장된 데이터 복구뿐만 아니라 서버에 저장된 데이터 수집까지 모두

고려해야 한다.

사용자 컴퓨터에 저장되는 통신내역은 크게 포맷이 있는 데이터와 그렇지 않은 데이터로 구분된다. 포맷이 있는 데이터는 메신저 서비스만의 포맷을 갖거나 혹은 XML, SQLite 처럼 잘 알려진 포맷으로 구성되어 있으며, 포맷이 없는 데이터는 단순히 텍스트 평문의 형태를 가진다. 텍스트 평문 형태의 통신내역은 해당 파일을 생성한 응용프로그램이 무엇인지를 확인할 메타데이터가 없기 때문에 이 같은 파일을 발견했다고 해도 쉽게 메신저 통신내역이라고 정의하기는 어렵다. 하지만 포맷이 있는 데이터의 경우 각 메신저별로 저장위치가 일정하며 파일 안에 고유의 메타데이터를 가지고 있기 때문에 어떤 메신저의 것임을 가려낼 수 있다.

포맷 있는 통신내역은 일반적으로 그 내용을 보호하기 위해 인코딩 혹은 암호화가 되어있다. 따라서 의미 있는 통신내역을 확보하기 위해서는 암호화된 데이터부터 복구해야 한다. 복구를 위해서는 먼저 통신내역이 저장되는 위치를 파악해야 하며 또한 그 안에서 어떤 데이터가 암호화 되어 있는지를 알고 있어야 한다. 전자의 경우는 레지스트리나 파일에 존재하는 메신저 프로그램의 설정 데이터를 확인함으로서 파악가능하며, 후자는 통신내역 데이터의 자료구조를 분석함으로서 알 수 있다. 복구해야 할 암호데이터를 확보했다면 복호화를 위해 필요한 것은 사용된 암호 키와 암·복호화 과정이다. 이 두 가지 요소는 암호화를 수행한 메신저 프로그램에 내장되어 있으며 알고리즘과 데이터를 확보하기 위해서는 역공학 분석이 필요하다.

메신저 서버에 저장되는 통신내역을 열람하기 위해서는 크게 두 가지 방법을 제안할 수 있다. 하나는 서비스에 대한 인증절차를 거쳐 원격으로 데이터에 접근하는 것이다. 그리고 두 번째는 서버에 대한 압수수색 권한을 획득한 후 직접 데이터를 확보하는 방법이 있다. 후자의 방법이 적법하기는 하지만 권한 획득을 위해 일련의 절차를 거쳐야 하기 때문에 긴급 수사상황에서는 효과적이지 못하다. 반면 전자의 방법으로 획득한 정보는 수사의 참고자료로서만 사용될 수 있을 뿐 법적증거로서의 효력이 없다. 게다가 수사의 긴급성이 인정되지 않으면 개인정보침해의 문제를 야기할 수도 있다. 디지털 포렌식의 법적 절차 준수 및 개인정보침해 문제와 수사의 긴급성에 관한 논란은 정책연구에서 다루어져야 할 것이다.

서버에 있는 통신내역을 원격에서 획득하기 위해서는 서버로부터 사용자 인증을 받아야만 한다. 인증에는 보편적으로 ID와 패스워드를 필요로 하며 그 인증은 세션이 종료되는 순간까지만 유효하다. 때문에 서비스를 다시 이용하기 위해서는 ID와 패스워드를 재입력해야 하는 수고가 필요하다. 하지만 대부분의 메신저 서비스는 사용자 편의성을 위해 인증을 자동으로 수행해주는 자동로그인이라는 기능을 가지고 있다. 그리고 이 기능은 자동인증요소를 로컬 시스템에 저장하는 구조적 취약점을 내포하고 있기 때문에 인증우회의 한 수단으로 사용된다.

자동로그인 기능이 활성화되어 있는 메신저 서비스에 대해서 인증우회 기법을 적용하기 위해서는 먼저 자동인증요소를 확보해야 한다. 자동인증요소는 인증에 필요한 사용자 ID와 패스워드 역할을 하는 보안토큰을 의미하며 윈도우즈 운영체제의 경우 보편적으로 레지스트리나 파일에 저장되어 있다. 대부분의 보안토큰은 그 데이터가 외부로 유출되었을 때 남용되는 것을 막기 위해 생성과정에서 시스템의 MAC 주소를 사용하고 있다. 그러므로 획득한 보안토큰을 수사에 활용하기 위해서는 보안토큰이 생성되는 과정과 생성된 시스템의 MAC 주소를 알고 있어야 한다. 보안토큰 생성 알고리즘은 각 메신저 프로그램을 역공학함으로써 규명 가능하다.

IV. 주요 메신저별 분석

4.1 네이트온

과거 네이트온 메신저는 통신내역을 사용자 컴퓨터에 저장하였으나 2009년 10월 5일 이후로는 모든 대화내용과 쪽지내용을 메신저 서버에 저장하고 있다. 따라서 통신내역을 확보하기 위해서는 메신저 서버에 로그인해야 한다. 본 소절은 자동로그인 취약점을 이용한 네이트온 메신저 서버로의 인증 우회 기법을 기술한다.

4.1.1 자동인증 데이터의 위치

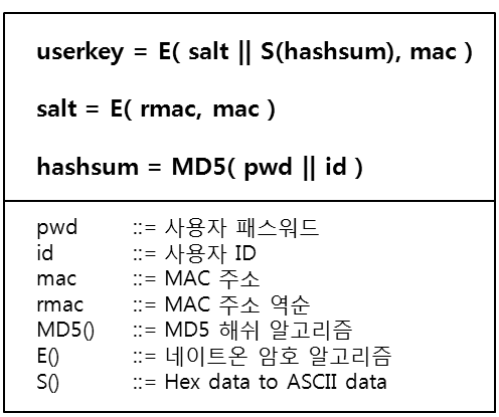
네이트온은 자동로그인 기능이 설정될 경우 사용자 ID와 보안토큰을 레지스트리에 저장한다. HKEY_CURRENT_USER\Software\SK Communications\Messenger\Settings\Save Pass 키 값은 자동로그인 기능의 활성화 여부를 나타내며, 활성화되어있을 경우 Y 그렇지 않을 경우 N으로 설정되어 있다. HKEY_CURRENT_USER\Software\SK

Communications\Messenger\Settings\UserID 키 값은 자동인증에 사용되는 메신저 ID를 저장하고 있다. 그리고 보안토큰은 HKEY_CURRENT_USER\Software\SK Communications\Messenger\Settings\UserKey 키 값에 16진수 데이터로 저장되어 있다.

4.1.2 보안토큰 생성과정

네이트온의 보안토큰은 사용자 패스워드와 ID를 MD5로 해쉬한 후 그 값을 네이트온 고유의 알고리즘으로 암호화 한 값이다. 이 데이터는 레지스트리의 특정위치에 저장되며 생성과정에서 시스템의 MAC 주소를 첨가하기 때문에 다른 시스템에서는 의미 없는 보안토큰이 된다. 즉, A 컴퓨터의 MAC 주소를 이용해 생성된 보안토큰은 MAC 주소가 다른 B 컴퓨터에서는 유효하지 못하다. 이는 수사대상의 하드디스크 이미지로부터 보안토큰을 획득해도 수사관의 컴퓨터에서는 이용할 수 없다는 의미이다. 따라서 A 컴퓨터에서 획득한 보안토큰을 다른 컴퓨터에서 활용하기 위해서는 보안토큰의 재구성이 필요하다.

〔그림 1〕은 네이트온 메신저의 보안토큰 userkey가 생성되는 과정을 수식으로 표현하고 있다. 먼저 사용자 패스워드는 이메일 형태의 ID와 문자열로서 합쳐진 후 1회의 MD5 해쉬 알고리즘을 거친다. 예를 들어, 패스워드가 '1234'이고 ID가 'neobug@korea.ac.kr'이라면 해쉬의 입력 값으로 '1234neobug@korea.ac.kr'이 선택되는 것이다. 다만 이메일 도메인이 'neobug@nate.com'일 경우는 '1234neobug'처럼 ID만 합쳐진다[8]. 시스템마다 고유의 보안토큰을 생성하기 위한 목적으로 사용되는 salt는



〔그림 1〕 네이트온 보안토큰 생성 과정

```
NateOn_Decode(UserKey, MAC[])
{
        index = 0    count = 0    neobug = 70

        LOOP index to UserKeySize
        {
                temp =  UserKey[index] + 254

                IF (temp-0xFE) < neobug
                        temp = (temp+255)-254-neobug
                ELSE
                        temp = temp-254-neobug

                IF (count%12) == 0
                        count = 0

                MACtemp = MAC[count]
                count = count + 1
                temp = temp+MACtemp

                IF temp > 255
                        temp = temp + 1

                result = temp
                neobug = UserKey[index];
                UserKey[index]=result;
                index = index + 1
        }
}
```

〔그림 2〕 네이트온 보안토큰 복호 알고리즘

MAC 주소와 〔그림 2〕의 네이트온 암호알고리즘으로 만들어진다. 평문은 MAC 주소를 역순으로 나열한 12자리 문자열이며 암호 키는 정순으로 나열한 MAC 주소가 사용된다. 최종적으로 보안토큰은 위 과정에서 생성된 salt와 아스키 형태로 치환한 MD5 해쉬 값을 암호화함으로서 생성된다. 사용되는 암호 알고리즘과 암호 키는 salt를 생성할 때 사용된 것과 동일하다.

4.1.3 인증우회 기법

수사관이 용의자의 하드디스크 이미지로부터 보안토큰을 입수하였다면 〔그림 1〕 과정과 〔그림 2〕의 복호 알고리즘을 이용하여 hashsum을 추출할 수 있다. hashsum 추출이 완료되면 인증우회를 할 시스템의 MAC 주소를 첨가해 새로운 보안토큰을 생성한다. 그리고 UserKey 레지스트리 키 값에 새로운 보안토큰을 설정하고, UserID 키 값에는 이미지로부터 획득한 사용자 ID를 설정한다. 마지막으로 Save Pass 키 값을 Y로 바꿔준 후 네이트온 메신저를 실행시키면 사용자 인증이 자동으로 수행된다.

4.2 버디버디

현재 버디버디 메신저는 통신내역을 사용자 컴퓨터와 메신저 서버에 동시 저장하고 있다. 이는 2009년 11월 12일 이후 적용된 사항이며 이전에는 오직 로컬

컴퓨터에만 통신내역을 저장하였다. 정확히는 서버에 저장되는 통신내역은 쪽지내역 뿐이며 3개월의 수명을 가진다. 반면 사용자 컴퓨터에는 대화·쪽지·파일송수신 내역 모두가 저장되며 사용자가 삭제를 하지 않는 이상 지속적으로 존재한다. 그러므로 버디버디 메신저의 통신내역을 확보하기 위해서는 로컬 컴퓨터와 서버 모두를 고려해야 한다.

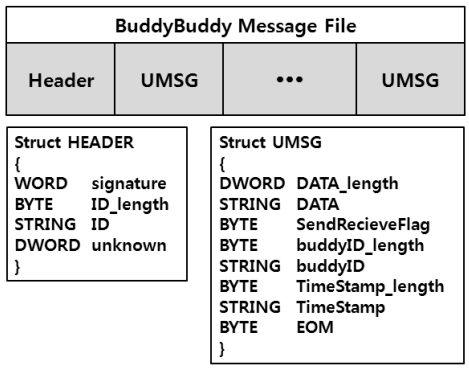
4.2.1 통신내역 및 자동인증 데이터의 위치

버디버디 메신저의 통신내역은 파일시스템의 고정된 경로에 저장된다. 메신저가 설치된 경로를 〈BuddyBuddy PATH〉라고 정의했을 때, 모든 대화·쪽지·파일송수신 내역은 〈BuddyBuddy PATH〉\BuddyBuddy\UMSG 밑에 저장된다. 대화내역은 이 경로에 사용자 ID별로 디렉터리를 생성하여 관리하며 각 디렉터리 밑에 대화상대별로 내용이 저장된다. 즉, 버디버디의 대화내역은 〈BuddyBuddy PATH〉\BuddyBuddy\UMSG\〈메신저ID〉\〈대화상대ID〉.btvU와 같은 다수의 파일로 저장되는 것이다. 반면 버디버디의 쪽지내역은 각 ID별로 〈BuddyBuddy PATH〉\BuddyBuddy\UMSG\〈메신저ID〉.bmvU라는 하나의 파일에 모든 송수신 내역을 저장한다. 파일송수신 내역은 대화내역과 같은 방법으로 저장된다.

버디버디 메신저의 자동인증요소는 레지스트리에 존재한다. 메신저 ID는 레지스트리 키 값 HKEY_LOCAL_MACHINE\SOFTWARE\BuddyBuddy\BuddyBuddy\LID에 기록되며, 인증에 사용되는 보안토큰은 HKEY_LOCAL_MACHINE\SOFTWARE\BuddyBuddy\BuddyBuddy\PEDU에 저장된다. 자동로그인 설정 여부에 대한 판단은 PEDU 값의 존재여부로 판단한다. (2010년 11월 1일 기준 버디버디의 보안토큰은 HKEY_LOCAL_MACHINE\SOFTWARE\BuddyBuddy\BuddyBuddy\PEDN에 저장되는 것으로 변경 되었다.)

4.2.2 통신내역 파일 구조

버디버디의 대화·쪽지·파일송수신 내역은 〔그림 3〕과 같은 데이터 구조를 가진다. 항상 파일의 처음에는 헤더 구조체가 존재하며 그 뒤를 이어 다수의 메시지 구조체가 연속으로 나열되어 있다. 하나의 메시지 구조체는 쪽지의 경우 상대방에게 전송하거나 받은 한



〔그림 3〕 버디버디 통신내역 파일 구조

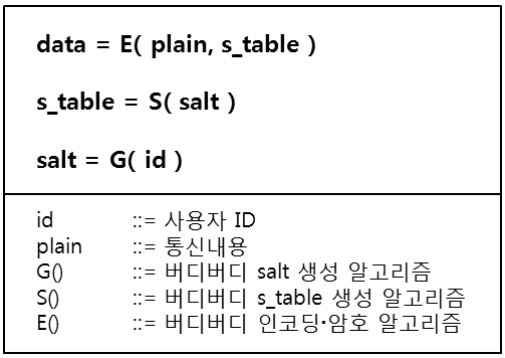
번의 통신내용만 담고 있고, 대화 및 파일송수신의 경우 세션이 시작한 시점부터 종료될 때까지의 모든 내용을 담고 있다. 메시지 구조체는 통신을 완료한 시간 순으로 기록된다.

헤더 구조체는 signature, ID_length, 그리고 ID 등으로 구성되어 있다. signature는 파일의 시작을 알리는 값으로 항상 FFFE 값을 유지한다. ID는 해당 파일의 주인이 누구인지를 뜻하고 있으며 유니코드 방식으로 저장된다. 즉, ID_length 값이 6이라면 실제 ID의 크기는 12바이트가 되는 것이다.

메시지 구조체는 암호화 된 통신내용과 통신이 이루어진 시간 그리고 누구와 통신을 시도하였는지를 기록하고 있다. DATA_length는 통신내용의 크기를 나타내며, SendReceiveFlag는 통신내용이 헤더의 ID 기준으로 송신 혹은 수신되었는지를 알려준다. 이 값이 0이면 송신 데이터이고 1이면 수신 데이터이다. buddyID는 누구와 통신을 하였는지를 기록하고 있으며, TimeStamp 값은 통신이 이루어진 시간을 나타낸다. 여기서 buddyID와 TimeStamp 값은 유니코드 형식 문자열로 저장되기 때문에 각각의 크기를 알려주는 length 값들이 존재한다. 마지막으로 한 개의 메시지 구조체가 끝날 때마다 1바이트의 0x00 값이 종결자로서 추가된다.

4.2.3 통신내역 암호화 과정

버디버디의 메시지 구조체는 암호화 된 통신내용을 저장하고 있으며 그 생성과정은 〔그림 4〕와 같다. 먼저 알고리즘 E는 분석결과 인코딩 알고리즘에 가깝지만, 인자 값 s_table에 의해 암호화 속성도 가지고 있음을 확인하였다. s_table은 사용자 ID로부터 파생



〔그림 4〕 버디버디 통신내용 암호화 과정

된 64바이트 크기의 문자열로 알고리즘 E에서는 암호 키로 사용된다. 알고리즘 S는 암호 키 s_table을 직접적으로 생성하는 알고리즘으로서 고정된 64바이트 문자열을 일정한 규칙대로 섞어주는 역할을 한다. 이 때 일정한 규칙에 변화를 가져오는 요소로서 salt가 사용되는데 이 salt는 알고리즘 G에 의해 생성되는 4바이트 데이터이다. 그리고 salt는 사용자 ID로부터 생성된다. 즉, 알고리즘 G와 S는 패스워드 id로부터 암호 키 s_table을 생성하기 위한 키 스케줄링 과정이라 볼 수 있다.

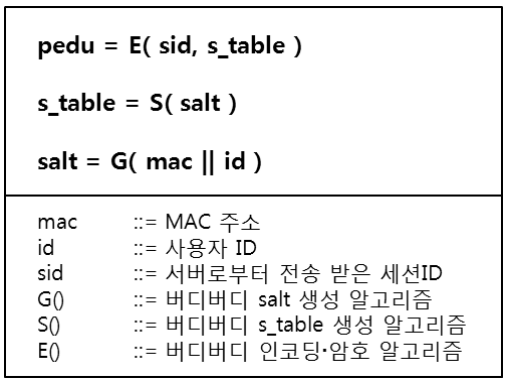
버디버디 메신저의 암호화 된 통신내용을 복구하기 위해서는 통신내역이 저장된 파일과 위에서 언급한 세 가지 알고리즘을 확보하고 있어야 한다. 통신내역 파일은 파일 시스템의 지정된 위치에서 발견할 수 있으며 삭제가 되었다면 파일복구를 통해서도 확보할 수 있다. 버디버디 고유의 알고리즘들은 메신저 프로그램을 분석함으로서 알아낼 수 있다. 알고리즘 E는 암호화를 수행하는 알고리즘이지만 이를 역으로 이용하면 복호화 과정을 산출할 수 있다. 그리고 통신내역 파일 안에 있는 사용자 ID를 기반으로 암호 키인 s_table을 생성하면 모든 통신내용을 복구할 수 있다.

4.2.4 보안토큰 생성 과정

버디버디의 보안토큰은 메신저 서버로부터 전송받은 세션ID를 암호화 한 값이다. 세션ID는 사용자가 인증 과정을 거친 후 서버로부터 전달받는 일련의 데이터로서, 클라이언트는 이 값을 통해 서버로부터 자신의 개체성을 인정받는다. 또한 세션ID에는 유효기간이 존재하는데 일반적으로 서버와 클라이언트의 연결이 종료되는 시점까지이다. 따라서 보안토큰 역시 세션ID의 영향을 받아 일정한 생명주기를 가지게 된다. 메신저 사용자가 로그인을 성공함으로서 최신의 보안토큰을 생성했다면, 이 보안토큰은 오직 다음 로그인이 성공되는 시점까지만 유효하다. 예를 들어, 사건 용의자의 보안토큰을 입수하였다 해도 그 용의자가 메신저에 로그인을 한번이라도 시도하면 그 시점에서 입수한 보안토큰은 무용지물이 된다는 것이다. 따라서 버디버디 메신저의 인증우회 기법은 시간적 제약이 존재한다.

버디버디의 보안토큰은 레지스트리 키 값으로 저장되기 때문에 상황에 따라 쉽게 노출될 수 있다. 그러므로 보안토큰 자체에 아무런 보안장치도 존재하지 않는다면 이 데이터는 쉽게 남용될 것이다. 예를 들어, 획득한 보안토큰을 단순히 복사만 함으로서 다른 컴퓨터에서도 메신저 서버로의 자동로그인이 가능해질 수 있다. 하지만 버디버디 메신저는 이런 문제를 방지하기 위해 보안토큰 생성과정에서 시스템에 의존적인 MAC 주소를 첨가하고 있다. 〔그림 5〕에 따르면 알고리즘 G는 MAC 주소와 사용자 ID를 입력 값으로 salt를 생성하고 있다. 이 salt는 앞 소절에서 언급한 것처럼 알고리즘 E의 암호 키를 생성하는데 사용되며 결과적으로 함수의 출력 값(보안토큰)에 큰 영향을 미친다. 즉, 버디버디 보안토큰은 MAC 주소 자체를 보정하지 않는 이상 해당 보안토큰을 생성한 시스템이 아니면 사용할 수 없도록 설계되어 있다. 그러므로 어떠한 지역적 제약 없이 서비스로의 사용자 인증을 성공시키기 위해서는 보안토큰의 재구성은 필수적이다.

〔그림 5〕는 버디버디 메신저의 보안토큰 pedu가 생성되는 과정을 수식으로 보여주고 있다. pedu의 생성과정은 앞 소절에서 설명한 통신내용 암호화 과정과 거의 동일하다. 다만 salt를 생성하는 알고리즘 G의 입력 형태가 조금 다른 것을 알 수 있다. 수식에는



〔그림 5〕 버디버디 보안토큰 생성 과정

MAC 주소와 사용자 ID의 병합 데이터가 입력되는 것으로 되어있지만, 정확하게는 10진수로 표기한 MAC 주소 문자열과 사용자 ID를 합친 데이터가 입력된다. 예를 들어, MAC 주소가 '00-22-1A-8F-94-03'이고 사용자 ID가 'neobug'라면 알고리즘 G의 입력 값은 '000034026143148003neobug'가 된다. 알고리즘 G의 출력 값은 4바이트 salt이고 이 값에 의해 알고리즘 E의 암호 키 s_table이 생성된다. 버디버디 보안토큰 pedu는 서버로부터 전송된 세션 ID를 이 암호 키로 암호화 한 값이다.

4.2.5 인증우회 기법

버디버디 메신저 서비스에 대한 인증우회를 성공시키기 위해서는 시간적·지역적 제약을 극복해야 한다. 하지만 시간적 제약에 대해서는 해결방법이 없기 때문에 지역적 제약을 극복하는 관점에서만 기술한다.

증거수집 과정에서 용의자의 하드디스크 이미지와 활성데이터를 모두 확보하였다면 수사관은 이미지로부터 레지스트리 하이브를 추출할 수 있다. 레지스트리 하이브는 모든 레지스트리 정보를 저장하고 있는 데이터베이스이다. 레지스트리 하이브에서 사용자 ID와 보안토큰을 추출하였다면 〔그림 6〕의 복호 알고리즘과 MAC 주소를 이용해 보안토큰으로부터 세션ID를 확보한다. 세션ID를 성공적으로 추출하였다면 인증우회를 시도할 컴퓨터의 MAC 주소로 새로운 보안토큰을 생성할 수 있다. 이렇게 생성된 보안토큰을 앞서 언급한 레지스트리 키 값 PEDU에 설정하고 LID 레지스트리 키 값에 사용자 ID를 기입해준 후, 버디버디 메신저를 실행시키면 사용자 인증은 성공적으로 수행된다.

```
Buddy2_Decode(STable, encodedDATA)
{
        DATA[] = null
        index1, index2, pivot, insert = 0

        LOOP index1 to encodedDATASize
        {
        LOOP index2 to STableSize
        {
                IF STable[index2] == encodedDATA[index1]
                {
                buf[pivot] = index2;
                IF pivot == 3
                {
                        DATA[insert] = ((buf[1] >> 4) & 15) | (buf[0] << 2)
                        DATA[insert+1] = ((buf[2] >> 2) & 15) | (buf[1] << 4)
                        DATA[insert+2] = (buf[2] << 6) | buf[3]
                        insert = insert+3
                        pivot = 0
                        BREAK
                }
                pivot = pivot + 1
                }
                index2 = index2 + 1
        }
        index1 = index1 + 1
        }
        return DATA;
}
```

〔그림 6〕 버디버디 고유 복호 알고리즘

4.3 야후!

야후! 메신저의 대화내역은 오직 사용자 컴퓨터에만 저장되며 그 파일은 대화상대 및 시간 등의 메타정보와 대화내용을 포함하고 있다. 그리고 대화내용은 다른 사용자가 쉽게 열람하지 못하게 암호화가 되어있다. 따라서 야후! 메신저의 대화내역을 확보하기 위해서는 대화내역 파일의 구조와 데이터 복호화 과정을 분석해야 한다.

4.3.1 통신내역 데이터의 위치

야후! 메신저의 대화내역은 파일시스템의 고정된 위치에 저장된다. 메신저가 설치된 경로를 〈Yahoo! PATH〉라고 정의하면, 대화내역은 〈Yahoo! PATH〉\Messenger\Profiles 밑에 사용자 ID 별로 디렉터리를 생성하여 각각 저장된다. 즉, 한 사용자의 대화내역이 저장되는 위치는 〈Yahoo! PATH〉\Messenger\Profiles\〈메신저ID〉 디렉터리 아래이다.

야후! 메신저는 한명의 상대와 통신하는 1:1 대화와 한 번에 다수의 상대와 통신하는 1:N 대화를 구분하여 관리한다. 〈Yahoo! PATH〉\Messenger\Profiles\〈메신저ID〉\Archive 밑에는 두 가지의 디렉터리가 존재하는데 Messages와 Conferences이다. Messages는 1:1 대화내역을 저장하는 디렉터리이며 Conferences는 1:N 대화내역을 저장하는 디렉터리이다. 각각의 디렉터리는 다수의 하위 디렉터리를 생성한 후 그곳에 대화내역을 분산 저장하는데, 1:1 대화내역의 경우 대화상대 별로 디렉터리를 생성하여 구분하는 반면 1:N 대화내역은 방장의 ID로 디렉터리를 생성하여 관리한다. 즉, 1:1 대화내역 파일이 저장되는 경로는 〈Yahoo! PATH〉\Messenger\Profiles\〈메신저ID〉\Archive\Messages\〈대화상대ID〉 이고, 1:N 대화내역 파일이 저장되는 경로는 〈Yahoo! PATH〉\Messenger\Profiles\〈메신저ID〉\Archive\Conferences\〈방장ID〉 이다.

야후! 메신저의 대화내역 파일은 하루를 기본 단위로 다수의 파일을 생성하여 관리한다. 대화내역 파일

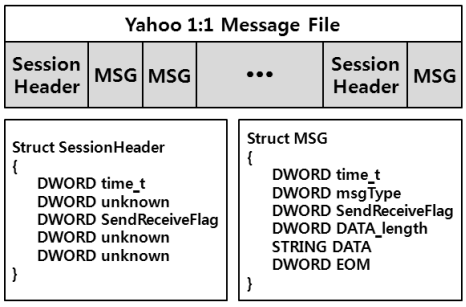
의 크기와는 관계없이 하루 전의 대화내역과 당일의 대화내역은 다른 파일로 구분된다. 생성된 파일의 이름은 '날짜-ID.확장자'의 형태를 유지한다. 예를 들어, 'neobug@ymail.com' ID를 가진 사용자가 2010년 1월 1일에 누군가와 통신한 대화내역은 '20100101-neobug@ymail.com.dat'의 파일 이름을 가진다.

4.3.2 통신내역 파일 구조

야후! 메신저의 대화내역 파일은 〔그림 7〕과 같이 하나의 세션헤더 뒤에 다수의 메시지 구조체가 연속되는 형태이다. 세션헤더는 대화가 시작된 시점의 정보로 구성되며 메시지 구조체는 송수신된 대화 하나하나를 의미한다. 즉, 사용자가 대화창을 열고 누군가와 대화를 시도하게 되면 그 순간 세션헤더 정보가 생성되는 것이며, 무언가를 전송하거나 받는 시점에서 메시지 구조체가 만들어진다. 이 메시지 구조체는 대화창이 종료되는 시점까지 연속적으로 이어지게 된다.

세션헤더 구조체는 항상 4바이트의 time_t 데이터를 가지고 시작한다. 이 데이터는 1970년 1월 1일을 기준으로 몇 초가 지났는지를 기록하고 있으며 그 시간 값은 대화창이 생성된 시점을 가리키고 있다. 세션헤더의 세 번째 필드인 SendReceviceFlag는 누구로부터 세션이 시작되었는지를 알려준다. 그 값이 0이면 사용자 자신이 대화창을 생성한 것이며 1이면 대화를 걸어온 상대가 세션의 주체임을 뜻한다.

메시지 구조체는 메시지가 전송된 시간, 대화방법의 종류, 송수신 주체 등의 메타데이터와 암호화 된 대화내용이 저장되어 있다. 첫 번째 필드인 time_t 데이터는 세션헤더의 그것과 동일한 표현방법을 사용한다. 그리고 그 시간 값은 대화내용이 전송된 순간을 가리킨다. 두 번째 필드인 msgType은 대화방법이



〔그림 7〕 야후! 메신저 통신내역 파일 구조

1:1 대화인지 1:N 대화인지에 따라 달라지는데, 그 값이 6이면 1:1 대화방법을 의미하고 25이면 1:N 대화를 의미한다. SendReceiveFlag는 세션헤더의 세 번째 필드처럼 전송주체를 확인해주는 데이터로 여기서는 대화내용이 전송객체가 된다. 메시지 구조체의 네 번째 필드는 대화내용이 저장되는 DATA필드의 크기를 기록하고 있으며, 마지막 필드인 EOM은 암호화 된 대화내용의 끝을 알려준다.

4.3.3 통신내역 암호화 과정

야후! 메신저는 파일에 저장시킬 대화내용을 암호화시키기 위해 XOR연산을 수행한다. 이때 사용되는 암호 키는 메신저의 사용자 ID이다. 예를 들어, 'neobug@ymail.com' 사용자가 대화를 나눈 후 그 내용을 저장하게 되면 원본 대화내용은 ID와 한 바이트씩 XOR를 순차적으로 수행하게 된다. 대화내용이 Hello라면 H와 n, e와 e, 그리고 l과 o가 차례대로 연산되어진다. 만약 ID의 마지막 글자인 m까지 사용했는데도 원본 대화내용이 남아있다면 다시 ID의 첫 번째 문자부터 순서대로 사용된다.

야후! 메신저의 대화내용을 복구하기 위해서는 대화내역 파일과 그 파일의 주체인 해당 ID가 필요하다. 대화내역 파일은 파일시스템의 고정된 경로에 저장되어 있거나 삭제된 영역으로부터 복구가 가능하다. 그리고 해당 ID는 그 파일의 이름에서 확보할 수 있다. 암호화에 수행되는 연산은 오직 한 번의 XOR연산이기 때문에 특별히 메신저 프로그램을 분석할 필요는 없다.

4.4 Mi3

Mi3 메신저는 실시간대화, 쪽지전송, 파일송수신 등 여타 메신저들처럼 다양한 통신서비스를 제공하고 있지만 서비스 제공업체는 쪽지전송 기능을 통신의 주력으로 삼고 있다. 일반 메신저들은 친구목록에서 친구를 선택하면 자동으로 대화창이 뜨는 반면 Mi3 메신저는 쪽지 전송창이 활성화된다는 점이 그 좋은 예이다. 확실히 Mi3 메신저는 증권관련 업무를 수행하는 사람들이 즐겨 쓰기 때문에 시간을 많이 뺏는 실시간대화보다는 간략하게 전할말만 전달하는 쪽지 기능이 더욱 부각될 수밖에 없다.

Mi3 메신저의 쪽지내역은 항상 컴퓨터의 고정된 경로에 저장되며 그 내용은 고유의 알고리즘으로 암호화 되어있다. Mi3 메신저의 자동로그인 요소는 레지

스트리가 아닌 파일에 저장되는데, 그 데이터 역시 메신저 고유의 알고리즘으로 암호화 되어있다. 따라서 Mi3 메신저의 쪽지내역을 복구하거나 자동로그인에 사용되는 ID와 패스워드를 확보하기 위해서는 사용된 암호알고리즘과 암호 키를 파악할 해야 한다.

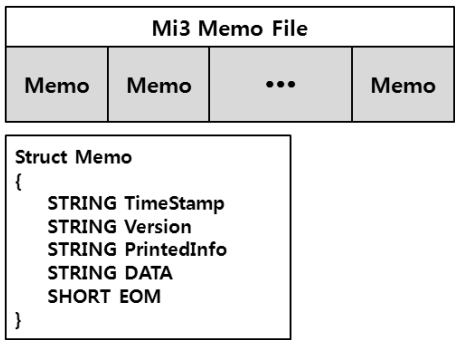
4.4.1 통신내역 및 자동인증 데이터의 위치

Mi3 메신저의 쪽지내역은 항상 고정된 경로에 저장된다. 메신저가 설치된 경로를 〈Mi3 PATH〉라고 하면, 모든 쪽지내역은 〈Mi3 PATH〉\Data 경로 아래에 사용자 별로 저장된다. 즉, 쪽지내역 파일이 저장되는 최종적인 위치는 〈Mi3 PATH〉\Data\〈메신저ID〉 디렉터리이다. 쪽지내역은 2개의 파일로 나누어 저장되는데 하나는 사용자가 송신한 쪽지내역을 담고 있는 SendMsg.ezd 파일이다. 그리고 다른 하나는 ReceiveMsg.ezd파일로서 사용자가 수신 받은 쪽지내역을 저장하고 있다.

Mi3 메신저의 자동인증 요소는 다른 메신저들과 달리 레지스트리가 아닌 파일에 저장된다. 해당 파일의 경로는 〈Mi3 PATH〉\Data\Config.ezd이며 자동인증요소 외에 메신저의 많은 설정데이터들이 저장되어있다. 이 파일에서 SAVEPASSWORD 키워드는 자동인증 기능의 활성화 여부를 알려준다. 그 값이 0이면 자동인증 기능이 비활성화 상태인 것이고 1이면 활성화 상태임을 의미한다. ID 키워드는 자동인증에 사용될 메신저 ID를 저장하고 있으며, 보안토큰은 PASSWORD 키워드에 기록되어 있다.

4.4.2 통신내역 파일구조 및 암호화 과정

Mi3 메신저의 쪽지내역 파일은 〔그림 8〕과 같이 다수의 메모 구조체로 구성되어있다. 각각의 메모 구조체는 송수신된 쪽지 하나하나를 의미하며, 오래된 쪽지부터 최근의 쪽지 순으로 순차 저장된다.



〔그림 8〕 Mi3 메신저 통신내역 파일구조

plain[i] = (cipher[i+1] ⊕ keyA[i]) ⊕ table[cipher[0]+i]

keyA = S(key)

key = R(IV, id)

IV	::= Mi3 키 생성을 위해 사용되는 고정 값
id	::= 사용자 ID
key	::= Mi3 쪽지내역 암호 키 (16bytes)
keyA	::= Mi3 쪽지내역 아스키 형태 암호 키 (32bytes)
table	::= 암·복호화에 사용되는 고정 테이블 (256bytes)
cipher	::= Mi3 쪽지내역 (암호문)
plain	::= Mi3 쪽지내역 (평문)
R()	::= 암호 키 생성 라운드 함수
S()	::= Hex data to ASCII data

〔그림 9〕 Mi3 메신저 통신내용 복호화 과정

메모 구조체는 암호화 된 쪽지내용, 쪽지의 송수신 시간 그리고 송수신 대상자를 아스키 문자열 형태로 저장하고 있다. TimeStamp는 쪽지가 송수신된 시간을 기록한 값으로 항상 20byte의 크기를 가진다. 쪽지의 송수신 시간은 PrintedInfo에서도 발견할 수 있는데, TimeStamp와 다른 점은 저장된 데이터의 표현이 프로그램이 아닌 사람을 위한 형태라는 점이다. 또한 PrintedInfo는 쪽지가 누구로부터 혹은 누구에게 전송되었는지도 기록하고 있다. EOM은 개행문자를 사용하고 있다.

Mi3 메신저의 쪽지내용 DATA는 〔그림 9〕와 같은 과정으로 복호화 된다. 먼저 암·복호화에 사용되는 키를 생성하는데 4가지의 라운드 함수 R이 사용되는데, 각 라운드 함수는 16번씩 수행되어 총 64번의 함수가 호출된다. 이때 입력되는 데이터는 16바이트 크기의 고정 데이터와 메신저의 ID이다. 라운드 함수 R에 의해 생성된 16바이트 결과 값 key는 다시 16진수 값을 그대로 아스키 데이터로 전환하여 32바이트 크기의 keyA를 생성한다. 그리고 이렇게 생성된 keyA를 가지고 Mi3 쪽지내역을 암·복호화 하게 된다. table은 Mi3 메신저 프로그램에 하드코딩 되어 있는 256바이트 크기의 데이터로서 암호문의 첫 번째 바이트를 오프셋으로 한 해당위치의 데이터를 복호화 과정에 사용한다.

4.4.3 보안토큰 생성 과정

Mi3 메신저의 보안토큰은 〔그림 10〕과 같이 로그인 패스워드에 대해 vector값을 XOR 연산함으로서 생성된다. vector의 시작 값은 항상 0x6525이고 한

vector = factor2 + 0x58BF

factorB = factorA * 0xCE6D

factorA = token[i] + vector

token[i] = password[i] ⊕ vector[0]

password ::= Mi3 메신저 사용자 패스워드
vector ::= CBC 모드와 같은 효과를 주는 데이터(IV=0x6525)
token ::= 보안토큰

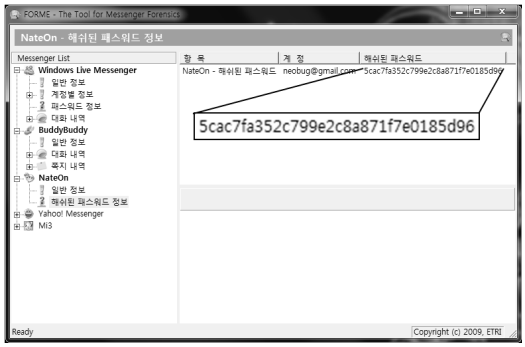
〔그림 10〕 Mi3 메신저 보안토큰 생성 과정

바이트의 보안토큰을 생성할 때마다 업데이트되어 마치 CBC 모드처럼 두 번째 바이트 보안토큰을 생성하는데 영향을 미친다. 예를 들어, 로그인 패스워드가 neo222이라면 보안토큰 생성과정은 다음과 같다. n의 16진수 값 0x6E와 IV의 첫 번째 데이터 0x65를 XOR한다. 그 결과는 0x0B이며 이 데이터가 보안토큰의 첫 번째 데이터가 된다. 보안토큰의 두 번째 데이터는 e의 16진수 값 0x65와 업데이트 된 vector 값을 XOR 해야 한다. 〔그림 10〕 수식대로 vector를 업데이트하면 0x6525는 0x0E2F가 되고 결국 0x65와 0x0E의 XOR 연산이 보안토큰의 두 번째 데이터가 되는 것이다. 이렇게 보안토큰 생성을 완료하게 되면 바이트 단위로 16진수 데이터를 10진수로 전환하면서 보안토큰의 생성을 마치게 된다. 이 10진수 문자열 데이터는 앞 소절에 언급한 파일에 기록된다.

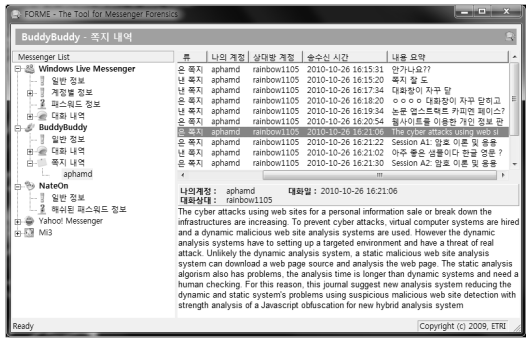
Mi3 메신저의 보안토큰은 패스워드 자체를 노출시키기 때문에 서버로의 인증이 필요할 때는 ID와 보안토큰으로부터 획득한 패스워드를 직접 사용한다. 또한 패스워드 자체는 수사대상으로부터 압수한 디지털객체들을 분석할 때 큰 도움이 될 수도 있다. 예를 들어, 암호화 되어있는 문서를 발견했을 때 메신저로부터 획득한 패스워드를 적용시켜볼 수도 있을 것이다.

V. 구 현

본 논문에서 제안한 메신저 통신내역 수집방안은 사용자 컴퓨터에 저장된 데이터를 직접 해독하는 방법과 인증우회를 통해 메신저 서버에 저장된 데이터를 확인하는 방법으로 구분된다. 4장에서는 각 방안이 실사례에 적용가능하다는 것을 보이기 위해 네이트온, 버디버디, 야후!, Mi3 메신저와 같이 실제 널리 쓰이는 메신저에 대한 대응 알고리즘을 기술하였다. 그리고 본 장에서는 앞서 기술된 알고리즘을 기반으로 구현한 인터넷 메신저 통신내역 수집도구인 FORME



〔그림 11〕 FORME - 네이트온 보안토큰 해체 기능



〔그림 12〕 FORME - 버디버디 통신내역 해석 기능

(The Tool for Messenger Forensic)를 보임으로서 논문에서 제안한 방안의 유효성을 증명한다.

〔그림 11〕은 프로그램 FORME가 네이트온 메신저의 보안토큰으로부터 추출한 패스워드+ID의 해쉬값을 출력하고 있는 화면이다. 이 해쉬 값은 패스워드 '1qaz2wsx'에 'neobug@gmail.com' 계정이 합쳐져 생성된 MD5 해쉬 값으로 인증우회 시 보안토큰 재구성의 핵심요소가 된다.

〔그림 12〕는 사용자 컴퓨터에 저장된 버디버디의 쪽지내역을 출력하고 있는 화면이다. 버디버디의 쪽지내역은 내용요약 필드와 내용 필드의 데이터가 암호화되어 있기 때문에 복구 전에는 그 내용을 알 수 없다. 하지만 FORME는 해당 데이터를 인식 가능한 문자열로 복호화 하는 기능을 가지고 있다.

VI. 결 론

사건수사에 있어서 전화나 휴대폰과 같이 전통적인 매체의 통신내역을 수집하는 일은 기본적이면서도 매우 중요한 일이다. 통신내역은 수사를 진척시킬 수 있는 참고자료 혹은 수사를 결정지을 수 있는 법적증거

자료가 되기 때문이다. 마찬가지로 인터넷 메신저의 통신내역을 확보하는 일은 디지털 증거획득 과정에서 필수적이다. 인터넷에는 이메일, 인터넷 전화 그리고 SNS 등 다양한 정보전달 수단이 존재하지만, 인터넷 메신저만큼 실시간 정보전달이 가능하고 다양한 기능과 편리한 인터페이스를 제공하는 복합적 통신수단을 찾아보기는 힘들다. 따라서 인터넷 메신저의 통신내역 및 사용흔적을 탐색하고 확보하는 일은 계속 연구되어야 한다.

본 논문에서는 파일시스템의 할당영역에 존재하는 데이터를 기반으로 통신내역을 확보하는 기술을 제안하였다. 또한 원격지에 저장되어있는 통신내역을 열람하기 위한 방안도 제시하였다. 하지만 인터넷 메신저는 정보전달 외에도 다양한 기능을 가지고 있기 때문에 다각도에서 분석될 필요가 있다. 예를 들어, 인터넷 메신저에서 서비스하고 있는 다양한 플러그인 기능들에 대한 디지털 포렌식 관점의 연구를 들 수 있다. 또한 비 할당영역에 존재하는 통신내역을 복구하기 위한 방법 등도 논의되어야 할 것이다.

참고문헌

〔1〕 한국인터넷진흥원, 2009 한국인터넷백서, 한국인터넷진흥원, pp. 190-193, 2009.

〔2〕 Mike Dickson, "An examination into AOL Instant Messenger 5.5 contact identification," Digital Investigation, vol. 3, no. 4, pp. 227-237, Dec. 2006.

〔3〕 Mike Dickson, "An examination into Yahoo Messenger 7.0 contact identification," Digital Investigation, Vol. 3, no. 3, pp. 159-165, Sep. 2006.

〔4〕 Mike Dickson, "An examination into MSN Messenger 7.5 contact identification," Digital Investigation, Vol. 3, no. 2, pp. 79-83, Jun. 2006.

〔5〕 Wouter S. van Dongen, "Forensic artefacts left by Windows Live Messenger 8.0," Digital Investigation, Vol. 4, no. 2, pp. 73-87, Jun. 2007.

〔6〕 Jessica Reust, "Case study: AOL instant messenger trace evidence," Digital Investigation, Vol. 3, no. 4, pp. 238-243, Dec. 2006.

〔7〕 한국인터넷진흥원, 2008 한국인터넷백서, 한국인터넷진흥원, pp. 230-233, 2008.

〔8〕 신동휘, 최윤성, 박상준, 김승주, 원동호, "네이트온 메신저의 사용자 인증 메커니즘에 대한 취약점 분석," 정보보호학회논문지, 17(1), pp. 67-80, 2007년 2월.

〈著者紹介〉

이 진 경 (Jinkyung Lee) 학생회원
2008년 2월: 한림대학교 컴퓨터공학 졸업
2010년 2월: 고려대학교 정보경영공학전문대학원 석사수료
2010년 7월~현재: 한국인터넷진흥원(KISA) 주임연구원
〈관심분야〉 디지털 포렌식, 역공학 분석, 악성코드

한 지 성 (Jisung Han) 학생회원
2009년 2월: 경원대학교 전자거래학 학사
2009년 3월~현재: 고려대학교 정보경영공학전문대학원 석사과정
〈관심분야〉 라이브 포렌식, 운영체제, 역공학

이 상 진 (Sangjin Lee) 종신회원
1987년 2월: 고려대학교 학사 졸업
1989년 2월: 고려대학교 석사 졸업
1994년 2월: 고려대학교 박사 졸업
1989년 10월~1999년 2월: ETRI 선임연구원
1999년 10월~현재: 고려대학교 정교수
〈관심분야〉 디지털 포렌식, 모바일 포렌식