「정보학연구」 제10권 제4호 2007년 12월(pp.41-56)
The Journal of Information Technology, Vol.10, No.4

컴퓨터 포렌식스 지원을 위한 시스템 로그 및 휘발성 정보 수집에 관한 연구

A Study of System Log and Volatile Information Collection for Computer Forensics

고 은 주 (Gho, Eun Ju)* · 오 세 민 (Oh, Se Min)** · 장 은 겸 (Jang, Eun Gyeom)*** ·
이 종 섭 (Lee, Jong Sub)***** · 최 용 락 (Choi, Yong-Rak)*****

목 차

Abstract

In Digital Computing Environment, volatile information such as register, cache memory, and network information are hard to make certain of a real-time collection because such volatile information are easily modified or disappeared. Thus, a collection of volatile information is one of important step for computer forensics system on ubiquitous computing. In this paper, we propose a volatile information collection module, which collects variable volatile informa-

* 대전대학교 컴퓨터공학과
** 대전대학교 컴퓨터공학과
*** 대전대학교 컴퓨터공학과
**** 대전대학교 컴퓨터공학과
***** 대전대학교 컴퓨터공학과

tion of server system based on memory mapping in real-time.

Key words: Digital Computing Environment, collects variable, server system

Ⅰ. 서론

디지털 컴퓨팅 환경에서의 서비스는 외부의 의도적인 공격 및 오작동에 의해 침해될 수 있다. 특히, 서비스를 제공하는 서버 시스템의 침해사고는 지난 2003년 125대란에서 보여준 것처럼 엄청난 피해를 초래한다. 기존의 일반적인 컴퓨팅 환경에서의 보안 대비책은 다방면에서 적용되고 있고 유비쿼터스 환경에서의 침해사고에 대한 예방과 대응은 아직까지 초기단계로서 부분적으로만 적용되어 많은 부분에서 미흡하다. 따라서, 유비쿼터스를 포함한 컴퓨팅환경은 보안 침해 및 시스템 오작동에 의한 사고로부터 기업 또는 공공기관의 업무적, 재산적, 지재권의 피해를 일으킬 수 있는 요지로 남아있다. 따라서 침해사고에 대한 예방 대책뿐만 아니라 만일의 침해사고에 대비하여 그 사고 이후에 이를 정확히 탐지, 분석, 대응하기 위하여 관련 증거의 수집이 선행되어야 한다[1].

특히, 침해사고 당시의 사라지기 쉬운 휘발성 정보는 침해사고 이후에도 보존되는 일반적인 로그 파일(시스템 로그 파일, 애플리케이션 로그 파일)들과 다르게 시간이 지나면 사라지기 때문에 그 정보를 획득하는 것이 매우 어렵다.

그러나 기존 휘발성 정보 수집 도구들은 수집 절차마다 수사관의 직접적인 제어가 요구하며 쉘 명령어 기반의 단편적인 휘발성 정보 수집만을 지원하고, 자동화된 실시간 수집 기능과 메모리에 저장된 정보의 수집이 지원되지 않기 때문에 침해사고 이후 변경되기 쉬운 메모리 정보들과 휘발성 정보들에 대한 자동화된 실시간 수집이 필요하다.

따라서 본 논문에서는 유비쿼터스를 포함한 컴퓨팅 환경에서 침해사고에 대응할 수 있는 컴퓨터 포렌식스를 지원하는 증거수집 시스템의 휘발성 정보 수집 기법을 제안한다. 본 연구를 위해 휘발성 정보수집 도구를 활용하여 증거수집 정보를 분석하여 컴퓨터 포렌식스 지원 로그를 생성하였다. 또한, 이상 징후 발생 시에도 자동적으로 휘발성 정보를 수집하여 메모리에 남아있는 정보를 수집할 수 있는 기능 모듈 설계하여 시스템 침해 발생시 공격 시나리오를 작성할 수 있도록 하였다.

본 논문의 구성은2장에서 본 연구의 관련 연구로써 휘발성 정보 수집 절차를 알아보고 TCT의 휘발성 정보 수집 기능을 분석한다. 그리고 서론에서의 기존 도구의 문제점과 관련연구를 바탕으로 컴퓨터 포렌식스를 지원을 위한 휘발성정보수집 요구사항을 분석하였다.

3장에서는 도출된요구사항을 바탕으로 휘발성 정보 수집 모듈의 구성을 설계하고 각 세부모듈의 기능에 대해 설명한다. 4장에서는 기존 휘발성 정보 수집 도구와의 기능 평가한 내용을

분석하고 마지막 5장에서는 본 연구의 결과와 향후 과제에 대하여 설명한다.

II. 관련 연구

1. 휘발성 정보 수집 절차

1) 휘발성 정보 수집 우선순위

휘발성 정보는 운영 중인 프로세스들, 활성화된 TCP/UDP 포트들, 컴퓨터 메모리에서 동작중인 프로그램 이미지, 버퍼 안의 내용들, 접속 요청들의 큐 내용들, 개체 인터페이스들, 그리고 커널을 위해 예약된 가상 메모리의 내부로 적재된 모듈 같은 것들이 있다. RFC 3227 "Guidelines for Evidence Collection"의 『증거 수집 과정에 관한 운영지침』에서는 "휘발성이 있는 것을 먼저 진행하고 그렇지 않은 것을 나중에 진행한다."라고 명시되어 있다. 또 휘발성의 순서(Order of Volatility: OOV) 원칙에서는 사라지기 쉬운 휘발성 정보의 순서를 다음과 같은 예를 들어 보여주고 있다[2].

① registers, cache
② routing table, arp cache, process table, kernel statistics, memory
③ temporary file systems
④ disk
⑤ remote logging and monitoring data that is relevant to the system in question
⑥ physical configuration, network topology
⑦ archival media

2) 증거 수집 절차

침해 증거물 수집은 침해사고 대응에서 가장 중요하며 최우선으로 수행해야 한다. 증거수집은 일정한 수집 절차를 기반으로 진행되어야 한다. 정해진 일련의 순서에 의해 행해지는 증거수집 절차는 휘발성의 특성이 높은 것부터 낮은 우선순위에 의해 순서대로 로그정보를 수집해야만 사라지는 증거를 최소화할 수 있다. 운용 중인 시스템(유닉스/리눅스 시스템)에서의 자료수집 단계에 따른 포렌식스 도구의 사용 절차는 아래와 같은 단계로 나눈다[3][4].

① 공격받은 시스템의 화면을 캡쳐: 스크린 샷의 일종으로 디지털 카메라를 사용하는 단순한 단계이다.
② media mount: 정보를 수집할시스템 안으로 외부 미디어를 mount 한다.

③ 수집 시작 시간: 정보 수집 시점에 대한 정보를 수집하는 단계이다.

④ cache table: cache table로부터 정보를 수집하는 단계로서 수명이 매우 짧은 정보들이 포함되어 있기 때문에 가장 먼저 수집하는 정보이다. arp와 routing table로부터 데이터를 수집한다.

⑤ 현재 접속 상태가 유지중인 연결들과 열려진 TCP/UDP 포트들: 모든 활성화된 raw소켓을 수집하기 전에 현재 접속들과 열려진 TCP/UDP 포트들에 대한 정보를 수집하는 단계이다.

⑥ 물리적 메모리 이미지: 전체 시스템의 메모리를 복사하는 단계이다. /dev/mem 장치를 복사하거나 또는 kcore 파일(리눅스 운영체제에서 RAM을 나타냄)을 복사함으로써 물리적 메모리에 접근한다.

⑦ 운영체제의 커널 메모리에 적재되어 있는 모듈들의 목록: 수집된 데이터가 완벽하다는 것과 'netstat'나 'lsof' 명령어들의 결과가 커널 레벨에서 변경되지 않았다는 것을 확인하는 단계로 현재 메모리에 적재된 모듈을 확인한다.

⑧ 활동 중인 프로세스의 목록: 모든 프로세스들, 열린 포트들과 파일들에 대한 정보를 수집하는 단계이다.

⑨ 의심스러운 프로세스 목록: 프로세서에 의해 할당된 전체 메모리를 복사하는 단계로 pcat 도구를 사용한다.

⑩ 공격받은 시스템에 대한 유용한 정보: 공격받은 시스템으로부터 일부 유용한 정보를 모으는 단계이다.

⑪ 수집 종료 시간: 마지막 단계로서 3 과 마찬가지로 모든 정보의 수집을 마친 시점에 대한 정보를 기록한다.

증거 수집을 위해 nc, dd, datacat, pcat, Hunter.o, insmod, netstat, arp, route, dmesg 등의 도구들을 활용한다. 그리고 각 단계에서 수집한 정보들은 MD5 등의 도구들에 의해 암호학적 점검 값들을 기록하며 수집된 모든 정보들은 netcat(nc)과 같은 도구를 이용하여 원격지로 전송하여 관리한다[5].

2. TCT의 휘발성 정보 수집 기능 분석

TCT(The Coroner's Toolkit)는 Unix 계열 시스템에서 수행되는 컴퓨터 포렌식스 도구이다. TCT에서 중요한 도구는 4가지로 Grave-robber, mactime, unrm, lazarus 등이 있다[6]. 이러한 도구 중에서 Grave-robber는 다양한 휘발성 데이터를 캡처하고 수집된 개별 증거의 무결성 확보를 위해 MD5 해쉬 값을 만든다. TCT는 휘발성의 순서 원칙에 따라 작성되었다. 이는 휘발성이 강한 정보를 항상 우선적으로 수집해야 함을 의미한다. Grave-robber가 수집 하는 기본

정보는 다음과 같다[6][7].

- 메모리 사용과할당
- 비 할당된 파일시스템 공간
- netstat, route, ARP, 다른 네트워크 도구의 현재 결과
- 'ps'와 'lsof'명령어를 통한 모든 프로세스 데이터 등

또한, Grave-robber는 기본적인 수집 기능 이외에 모든 파일에 대해서 stat와 md5를 수행하며, 디렉터리와 구성정보에 대하여 'strings'를 실행한다[8]. Grave-robber는 크게 3개의 옵션(일반적인 옵션, 대량의 데이터를 수집하기 위한 옵션, 소량의 데이터를 수집하기 위한 옵션)으로 구분되어 있으며 피해 시스템의 네트워크 상태, 호스트 및 프로세스 등의 휘발성 데이터를 신속하게 수집하는 도구이다[8][9].

3. 메모리 정보 수집

일반적인 휘발성 정보 수집 절차에는 메모리 정보 수집이 포함되어 있지 않다. 하지만 메모리에는 기존에 수집되는 휘발성 정보에는 포함되지 않는 많은 정보들이 존재하며 법적 증거로써 유용한 정보들일 가능성이 매우 크기 때문에 반드시 획득해야 한다.

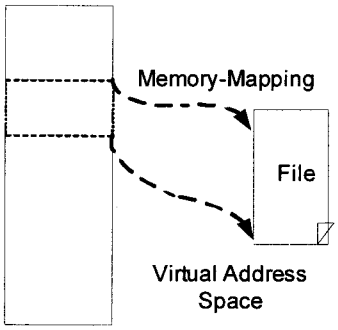
기존의 메모리 정보 수집 방법은 "윈도우 시스템에서 디지털 포렌식 관점의 메모리 정보 수집 및 분석 방법에 관한 고찰"에서 연구되었으며 그 중 Memory Mapped File에 대하여 설명한다.

가상 메모리와 유사하게 Memory Mapped File도 프로세스의 주소 공간에 특정 영역을 확보하고 이를 물리적으로 메모리에 매핑한다. 다른 점은 가상 메모리에서는 물리적 메모리가 페이징 파일로 한정되는 반면, Memory Mapped File은 유저가 지정하는 임의의 파일 자체라는 것이다. 이렇게 파일이 매핑되고 나면 파일을 마치 메모리인 것처럼 접근이 가능하다[10]. Memory Mapped File은 일반적으로 다음의 세 가지 경우에 사용된다.

실행 파일이나 동적 연결 라이브러리 파일들을 로딩: 운영체제가 실행 파일을 읽어 오고 실행하는 내부적인 방법도 바로 Memory Mapped File이다. 그림1은 Memory Mapped File 과 가상주소와의 관계를 보여준다.

- 디스크의 데이터파일에 접근: 디스크의 I/O를 줄이고 파일을 메모리에 버퍼링(buffering)할 수 있다. 이는 파일을 마치 메모리처럼 사용할 수 있으므로 간편한 파일 조작을 가능하게 한다.

· 프로세스간 통신: Memory Mapped File은 프로세스간 메모리를 공유하는 유일한 방법이기 때문에 시스템 상에서 수행중인 서로 다른 프로세스간의 데이터를 주고 받을 때 사용한다 [10].



〈그림 1〉 Memory Mapped File과 가상 주소와의 관계

Memory Mapped File을 이용하기 위해 다음의 세 단계의 절차를 갖는다.

① 디스크에서 Memory Mapped File로 사용할 파일을 식별하기 위하여 파일 커널 객체(file kernel object)를 생성한다.

② 시스템에게 파일의 크기와 어떻게 파일에 접근할 것인지를 알리기 위해 파일 매핑 커널 객체(file-mapping kernel object)를 생성한다.

③ 시스템에게 파일 매핑 커널 객체의 일부, 혹은 전부를 자신의 프로세스 주소 공간에 매핑 시킨다[10].

이와 같이 Memory Mapped File을 생성하고 다음의 세 단계에 의해 Memory Mapped File을 해제한다.

① 프로세스 주소 공간에 매핑되어 있는 파일 매핑 커널 객체(file-mapping kernel object)를 unmap한다.

② 파일 매핑 커널 객체를 close한다.

③ 파일 커널 객체(file kernel object)를 close한다[10].

4. 컴퓨터 포렌식스 지원을 위한 휘발성 정보 수집 요구사항

본 절에서는 서론에서 다룬 기존 도구들의 문제점과 제 2 장 관련 연구의 내용을 바탕으로 컴퓨터 포렌식스 지원을 위한 휘발성 정보 수집 요구사항을 도출한다. 다음은 도출된 요구 사항이다.

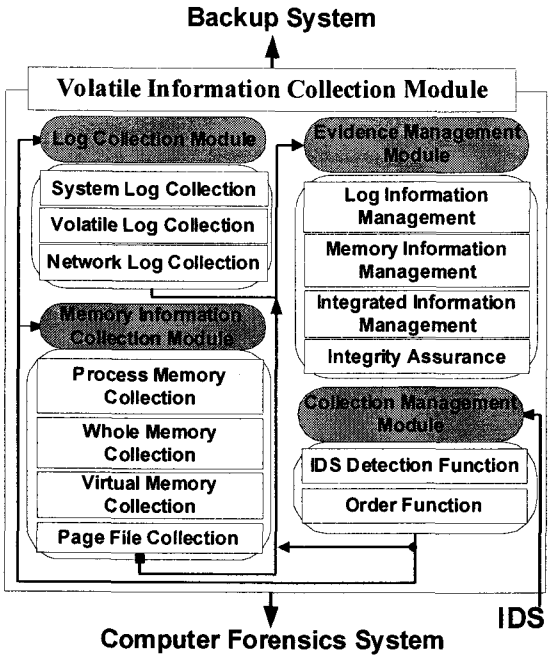
- 일관된 포렌식스 절차에 의한 휘발성 정보 수집: 절차상의 오류로 인하여 휘발성 정보가 범죄의 증거로 형성되지 못하는 것을 사전에 방지하는 것을 목표로 하여 적합한 절차에 의한 휘발성 정보 수집
- 시스템 포렌식스와 네트워크 포렌식스에서의 휘발성 정보 수집의 연동: 시스템 포렌식스에서의 획득하지 못하는 법적 증거로써의 휘발성 정보 수집을 위해 네트워크 포렌식스 도구에서의 휘발성 정보 수집과의 연동이 필요
- 획득 정보의 무결성 검증: 수집 모듈에서 수집한 휘발성 정보와 증거 관리 모듈에서 관리하고 있는 정보의 무결성 검증을 위해 원본과 분석본의 데이터 인증이 요구
- 획득 시스템의 인증: 컴퓨터 포렌식스 시스템에서 분석하고 있는 데이터가 피해 시스템으로부터 전송 받은 휘발성 데이터임을 검증하기 위한 증거물 획득 시스템 인증이 요구
- 메모리 기반의 휘발성 정보 수집: 프로세스 메모리, 시스템 전체 메모리, 가상 메모리, 페이지 파일 등에 포함된 정보들의 특성별로 수집하여 휘발성 정보와의 통합적인 관리가 필요
- 자동화된 증거 수집: 전문 수사관이 아닌 일반 관리자의 증거 수집을 위해서는 한번 혹은 그 이하의 명령으로 자동적으로 증거를 수집하는 기능이 필요

Ⅲ. 휘발성 정보 수집 모듈

1. 휘발성 정보 수집 모듈 구성도

본 논문에서는 디지털 컴퓨팅 환경에서 서버시스템으로 활용 될 수 있는 유닉스/리눅스 플랫폼에서 컴퓨터 포렌식스를 지원하는 휘발성 정보 수집 모듈을 2장에서 도출된 요구사항을 기반으로 시스템을 설계했다. 보안 침해사고 시 법적 대응을 위한 컴퓨터 포렌식스 시스템은 증거 수집 시스템, 증거 전송 시스템, 컴퓨터 포렌식스 시스템으로 구성될 수 있으며 이 중 본 논문에서는 증거 수집 시스템에서 휘발성 정보를 수집하기 위한 모듈을 제안한다.

휘발성 정보 수집 모듈은 Log Collection Module, Memory Information Collection Module, Evidence Management Module, Collection Management Module 의 4가지 세부 모듈로 나뉘며 구조는 그림 2와 같다.



〈그림 2〉 휘발성 정보 수집 모듈 구성도

Log Collection Module은 리눅스 시스템에 존재하는 일반적인 시스템 로그들을 수집하는 시스템 로그 수집 함수, 휘발성 정보를 수집하는 휘발성 로그 수집 함수, 네트워크 정보 캡쳐 도구Tcpflow를 이용해 세션별 네트워크 정보를 수집하는 네트워크 로그 정보 수집 함수로 이루어진다.

Memory Information Module은 시스템의 프로세스 메모리만을 수집하는 프로세스 메모리수집 함수, 시스템의 전체적인 메모리를 모두 수집하는 시스템 전체 메모리 수집 함수, 가상 메모리 정보를 수집하는 가상 메모리 수집 함수, 시스템의 페이지 파일 정보를 수집하는 페이지 파일 수집 함수로 구성된다.

Evidence Management Module은 로그 수집 모듈에서 수집한 로그 정보를 관리하는 로그 정보 관리 함수, 수집한 메모리 정보를 관리하는 메모리 정보 관리 함수, 각각의 모듈에서 수집한 로그 정보와 메모리 정보를 통합적으로 관리하는 통합 정보 관리 함수, 그리고 각 모듈에서 수집하고 관리하는 모든 정보들의 무결성을 검증하는 무결성 검증 함수로 구성된다.

마지막으로 Collection Management Module은 휘발성정보 수집 모듈 내의 전체적인 모듈의 기능을 제어하는 명령 전달 함수, 침입탐지시스템(IDS)로부터 탐지메시지를 수신하는 IDS 탐지 함수로 이루어진다.

2. 로그 수집 모듈

1) 시스템 로그 수집

시스템 로그는 utmp, wtmp, secure, lastlog, btmp, sulog, xferlog, history 등이 있으며, 이러한 로그 파일들은 독립적으로 분석하는 것이 아니라 서로 유기적인 관계를 갖고 분석해야 한다. 즉, 하나의 로그 파일에 침입 흔적이 발견되면 다른 로그 파일들에서 해당 IP나 해당 사용자와 관련된 로그를 분석해야 한다. 그러나 로그 정보는 공격자가 zap, wipe 등의 도구를 사용하여 쉽게 변경하거나 삭제할 수 있는 파일 관리상의 문제점이 있다.

따라서 피해 시스템에서는 MD5를 사용하여 시스템 로그의 신뢰성을 유지하면서 네트워크를 통하여 안전한 로그 관리 시스템에 저장해야 한다. 그리고 원래의 로그는 새로운 파일을 위한 공간을 확보해주기 위해 대상 시스템에서 삭제되는 로그 순환을 한다.

2) 휘발성 로그 수집

공개된 도구인 TCT에 포함된 Grave-Robber의 기능을 기반으로 중복되는 정보를 제거하며 자동으로 휘발성 로그를 수집할 수 있도록 수정하였다. 휘발성 로그의 생성과 동시에 로그 무결성 보장을 위해 수집된 모든 휘발성 로그에 대한 MD5 해쉬 값을 생성한다. 휘발성 로그 수집도 시스템 로그 수집과 마찬가지로 네트워크를 통하여 안전한 로그 머신에 저장되며, 새로 생성되는 파일의 공간 확보를 위해 로그 순환을 한다.

3) 네트워크 로그 수집

네트워크 패킷 정보를 세션별로 수집하여 파일로 생성하기 위해 공개도구인 Tcpflow를 활용한다. Tcpflow는 TCP 연결흐름 부분에서 전송되는 데이터를 캡쳐하고, 프로토콜을 분석하거나 디버깅하기 쉽게 데이터를 저장하는 프로그램이다. Tcpflow가 수집하는 데이터는 패킷 데이터의 종류에 따라 다양한 포맷으로 수집되며 주요 내용은 다음과 같다.

- Telnet을 통한 원격접속: 세션의 시작시점부터 종료시점까지 사용자가 입력한 모든 키 입력 정보를 수집한다.
- Ftp 파일 업로드 및 다운로드: 세션 시작시점부터 종료시점까지 ftp 접속 정보와 파일의 업로드 및 다운로드 정보를 수집한다.
- 웹 서비스: 클라이언트와 서버 간에 전송되는 웹 페이지의 소스코드 정보를 수집한다.

그러나, Tcpflow는 침해사고 분석에 필요한 시간정보와 데이터의 무결성을 위한 MD5 CheckSum을 생성하지 못하므로 Tcpflow에 의해 수집한 패킷 데이터의 헤더 정보에 시간정보와 MD5 CheckSum을 생성할 수 있는 기능을 추가 하였다. 시간 기록은 어떤 사건 보고의 원

자성과 로그의 상호연관 작업 수행에 절대적으로 필요하기 때문에 시간정보 데이터를 추가하였다.

3. 메모리 정보 수집 모듈

메모리 정보 수집 모듈은 정보 수집에 걸리는 시간 때문에 실시간 수집이 어려워 시스템 환경이나 관리자의 필요에 따라 임의의 주기로 수집된다. 그리고 침입 탐지 시스템에서 송신된침입 탐지 메시지를 IDS 탐지 함수가 수신한 경우에는 휘발성 정보 수집 모듈이 실행된 후에 마지막 침입 메시지를 수신한 이후 실행되는 모듈로 침입 탐지가 종료된 시점의 메모리 정보를 수집한다.

메모리 정보 수집 모듈은 프로세스 메모리 수집 함수, 시스템 전체 메모리 수집 함수, 가상 메모리 수집 함수, 페이지 파일 수집 함수로 이루어진다.

1) 프로세스 메모리 수집

메모리 정보는 크게 시스템 메모리와 프로세스 메모리로 구분할 수 있다. 먼저 프로세스 메모리를 획득하는 프로세스 메모리 수집 함수에 대해 설명한다. 메모리 덤프 파일에는 해당 프로세스가 사용했던 정보들이 있기 때문에, 사용자가 입력한 데이터 혹은 프로세스가 생성한 데이터들이 존재한다. 프로세스 메모리 수집 함수는 웹 브라우저의 접속관련 프로세스, 인터넷 뱅킹시의 인터넷 뱅킹 프로그램 등의 웹 인증 관련 프로세스 정보를 수집하는 함수이다.

메모리 덤프 파일에는 바이너리 값과 문자열이 섞여 있어 수사관이 분석하기 힘들기 때문에 가독성이 변한 형태로 변환한다. 이것은 오브젝트 파일이나 실행 파일에서 ASCII 코드나 유니코드 문자열을 검색해서 출력해주는 'strings' 명령의 기능을 기반으로 한다.

(1) 웹 브라우저

웹 브라우저를 이용해서 특정 웹사이트에 보안 접속을 통한 로그인을 하고 그 메모리를 덤프 한다. 클라이언트가 웹 서버에 보안접속 과정을 거쳐 로그인을 하더라도 ID와 패스워드는 반드시 메모리에 적재되었다가 암호화되어 통신하기 때문에 메모리에 암호화되기 전의 평문 상태가 그대로 남아 있게 된다.

(2) 인터넷 뱅킹 프로세스

인터넷 뱅킹을 사용하기 위해 특정 은행 사이트에 공인 인증서를 통한 로그인을 하고 클라이언트 프로그램과 웹 브라우저 프로세스 메모리를 덤프한다. 덤프 파일에는 다음과 같이 인증서 패스워드, 계좌 번호, 해당 계좌의 비밀 번호가 평문 형태 그대로 있는 것을 확인하였다.

2) 시스템 전역 메모리 정보 수집

앞에서 살펴본 프로세스 메모리 덤프 방법을 사용해서는 시스템 메모리를 획득할 수 없다. 하지만 운영체제의 최대 절전 모드를 사용한다면 시스템의 전체 메모리를 손쉽게 획득할 수 있다.

리눅스에서는 시스템의 전원이 차단될 때 메모리의 모든 내용을 하드 디스크에 저장하는데 시스템 전체 메모리 수집 함수는 ntsysv 유틸리티를 통해서 특정 파일에 저장한다. 시스템이 설치될 때 운영체제에서 기본적으로 RAM 용량과 같은 크기의 공간을 확보한다. 최대 절전 기능으로 ntsysv 유틸리티를 통해 저장된 파일을 획득한다면 현재 메모리 전체 내용을 획득할 수 있다.

3) 가상 메모리 정보 수집

현재 사용되고 있는 대부분의 컴퓨터 시스템은 가상 메모리 기법을 이용한다. 가상 메모리(Virtual Memory)는 운영체제에 의해 구현되는 개념으로서 프로그래머에게 큰 용량의 메모리나 데이터 저장 공간을 사용할 수 있도록 허용하는 것을 말한다.

가상 메모리 수집 함수에서는 주기억 장치와 보조기억 장치 사이의 데이터 교환인 스와핑(swapping)기능을 기반으로 가상 메모리를 수집하며 그 절차는 다음과 같다.

① 프로그램 실행
② 메모리에서 프로그램이 프로세스를 선택
③ 하드디스크의 스왑 영역으로 이동
④ 하드디스크에 저장된 메모리 정보를 수집하여 증거 관리 모듈로 전송

4) 페이지 파일 정보 수집

페이지 파일에 기록되어 있는 정보들은 운영체제에서 기본적으로 시스템이 종료될 때 삭제하지 않게 되어 있다. 페이지 파일 수집 함수는 이러한 정보들을 수집하는 함수이다. 운영체제나 다른 기타 어플리케이션들이 설치된 경우 시스템에서 페이지 파일의 영역을 하드 디스크상 임의의 공간에 할당한다. 이렇게 할당된 페이지 파일에서 하드 디스크의 이전 사용 흔적을 찾을 수 있다. 왜냐하면 하드 디스크의 데이터를 삭제하고 포맷을 하더라도 실제 데이터는 지우지 않기 때문에 시스템을 설치하기 이전의 데이터가 남아 있기 때문이다. 여기서 수집된 데이터는 증거 관리 모듈로 전송된다.

4. 수집 정보 관리 모듈

수집 정보 관리 모듈은 휘발성 정보 수집 모듈의 모든 모듈들이 정보를 수집하는 시점과 전

송하는 시점을 제어하는 모듈이다. IDS 탐지 함수와 명령 함수로 이루어져 있으며, 다음의 경우에 각 모듈에서 정보의 수집 또는 증거 관리 모듈로의 전송이 이루어진다.

(1) 정보 수집 시점: 일반적인 상태에서 실시간으로 휘발성 로그 수집 모듈이 실행되어 휘발성 로그를 주기적으로 수집한다. 그리고 메모리 수집 모듈은 임의의 시간 주기로 수집한다.
(2) 정보 전송 시점: 메모리 정보 수집 모듈의 수집 지연 시간을 고려하여 임의의 시간 주기마다 수집된 정보들이 전송된다, 휘발성 로그 수집 모듈에서 수집된 정보와 메모리 수집 모듈에서 수집된 정보가 증거 관리 모듈로 전송된다.
(3) 침입탐지시스템으로부터 침입정보 메시지를 수신한 경우: 침입탐지시스템이 이상 징후를 알리면 휘발성 정보 수집 모듈(System Log, Volatile Log, Network Log)과 메모리 정보 수집 모듈이 동시에 실행되며 생성된 정보들이 증거 관리 모듈로 전송된다. 여기서 메모리 정보 수집 모듈은 마지막으로 침입 탐지 메시지를 수신한 이후에도 실행된다.
(4) 보안 관리자의 요청에 의한 경우: (3)의 경우와 마찬가지로 모든 수집 모듈들(Volatile Log Collection Module, Memory Information Collection Module)이 실행되며 생성된 정보들이 전송된다.

따라서, 평상시에는 실시간 휘발성 로그 수집 모듈의 수집과 결과물의 전송, 5분 주기의 메모리 정보 수집 모듈의 수집과 결과물의 전송이 이루어지며, 침입탐지시스템이나 보안 관리자의 요청에 의한 경우에는 그 즉시 모든 수집 모듈(휘발성 정보 수집 모듈, 메모리 정보 수집 모듈)의 수집과 전송을 수행시킨다..

5. 증거 관리 모듈

증거 관리 모듈은 수집된 증거의 안전한 저장과 차후 분석에 용이하도록 데이터를 축약하여 관리하는 모듈이며, 로그 정보 관리 함수, 메모리 정보 관리 함수, 통합 정보 관리 함수, 무결성 검증 함수로 구성된다.

1) Log Information Management

상이한 형식으로 수집된 정보는 사건 통합을 위해 정규화 과정이 필요하다. 정규화는 원래 데이터의 무결성을 파괴하지 않으면서도 여러 형태의 로그와 연관시켜 로그파일의 원래 형식에서 데이터를 추출하는 상호연관 능력을 의미한다. 그리고 수집한 증거의 저장 공간을 줄이기 위해 중복되는 부분의 데이터 축약이 필요하다. 데이터 축약은 일련의 연관 사건들을 식별하고, 선택하는 기준에 따라서 상호연관 관계를 형성하도록 하기 위해 데이터를 추출하는 절차이다. 로그 정보 관리 함수는 다음의 규칙을 가지고 축약되어 통합 정보 관리 함수로 전송된다.

· 시스템 로그 정보

S = { *User, IP Address, Time, Port, LogInterval, FailDate, SuDate, SuTime, Daemon*}

· 휘발성 로그 정보

V = { *User, IP Address, MAC Address, Time, Port,Host, Protocol, State, Directory*}

· 네트워크 로그 정보

N = { *User, IP Address, MAC Address, Time, Port, OS, Execute Command*}

2) Memory Information Management

수집된 프로세스 메모리, 시스템 전체 메모리, 가상 메모리, 페이지 파일은 소유자(사용자 ID 혹은 프로세스), 침해 시점, 메모리 데이터로 정규화하여 저장할 수 있다. 수집된 정보는 통합 정보 관리 함수로 전송된다.

· 메모리 파일 정보

M = { *User* $\cup$ *Process, Time, Memory Data*}

3) Integrated Information Management

· 데이터 축약

$S \cap V \cap N \mid\mid M$ = {*User, IP Address, Time, Port, Memory Data* }

$S \cup V \cup N \mid\mid M$ = { *User, IP Address, Time, Port, Protocol, MAC Address, Memory Data* }

통합 정보 관리 함수에서는 로그 관리 함수에서 1차적으로 축약한 정보를 위와 같이 침해사고와 관련된 3가지 휘발성 정보를 상호 연관시키기 위해 동일한 정보를 정규화 하여 일정한 형식으로 표현한다. 그리고 데이터 중복을 줄이고 비일관성의 기회를 최소화하기 위해 공통되는 사용자 ID, IP 주소, 시간 정보를 기반으로 데이터를 축약하여 통합 정보 관리 데이터베이스에 저장한다. 그리고 메모리 정보 수집 함수에서 전달된 4가지 정보들은 사용자와 침해 시점을 기반으로 링크되어 저장된다.

4) Integrity Assurance

각각의 모듈에서 수집한 로그 정보와 메모리 정보들은 정보의 관리를 위해 증거 관리 모듈로 전송되어 각각의 관리 함수로 전달이 되어야 한다. 여기서 netcat(nc)를 이용하여 전송하게 된다. 그러나 전송 과정 후에 발생될 수 있는 파일 관리상의 무결성 문제가 발생할 수 있다. 따라서 수집 정보의 안전한 전송을 위해서는 Log Collection Module이 생성한 로그 파일들과

각 로그 파일의 MD5 해쉬 값 파일들을 netcat으로 전송하기 전에 cryptcat을 사용하여 암호화 하여 증거 관리 모듈로 전송해야 한다. 그리고 수신 후에는 복호화 후 로그 파일들의 MD5 해쉬 값을 비교하여 무결성 문제를 해결할 수 있다.

Ⅳ. 결과 분석

휘발성 정보 수집 모듈의 기능 평가는 3.1절의 컴퓨터 포렌식스 시스템 구성도에 나타난 시스템 환경을 고려하여 이루어졌으며2장에서 도출된 요구사항을 바탕으로 대표적인 휘발성 정보 수집 도구인 TCT(The Coroner's Toolkit)와 범용적으로 사용하는 컴퓨터 포렌식스 도구인 Guidance Soft사의 EnCase를 제안 모듈과 비교하였다. 각 요구 사항에 대한 지원 현황은 표 2와 같다.

〈표 1〉 제안 시스템 평가 (지원가능○, 지원불가×)

요구사항		TCT	EnCase	제안 시스템
일관된 포렌식스 절차에 의한 휘발성 정보 수집		○	○	○
자동화된 증거 수집		×	×	○
시스템 포렌식스와 네트워크 포렌식스에서의 휘발성 정보 수집의 연동		×	×	○
획득 정보의 무결성 검증		○	○	○
획득 시스템의 인증		○	×	○
메모리 기반 휘발성 정보 수집	프로세스 메모리 수집	×	×	○
	시스템 전체메모리 수집	○	×	○
	가상 메모리 수집	×	×	○
	페이지 파일 수집	×	×	○

제안한 휘발성 정보 수집 모듈은 기존의 포렌식스 도구들의 휘발성 수집 기능이 가지는 문제점인 자동적인 정보 수집의 어려움, 일관된 포렌식스 절차의 휘발성 정보 수집 지원 불가, 시스템 포렌식스와 네트워크 포렌식스에서의 휘발성정보 수집의 연동 불가, 메모리 정보 수집 기술 불가 등을 해결할 수 있도록 설계 되었다.

각 수집 모듈들의 수집시점과 전송시점을 결정하는 수집 관리 모듈에서 매초 혹은 매분 마다 주기적으로 데이터를 수집 및 전송을 제어하고 관리하여 침해사고 시점의 정보들을 수집하기 때문에 자동화된 증거 수집과 일관된 포렌식스 절차에 의한 휘발성 정보 수집이 가능하다.

로그 수집 모듈에서 시스템 로그 정보, 휘발성 로그 정보, 네트워크 로그 정보를 서로 다른

함수에서 유연하게 수집하여 시스템 포렌식스와 네트워크 포렌식스에서의 휘발성 정보 수집의 연동에 필요한 정보를 수집한다. 무결성 검증 함수는 수집한 정보를 전송하기 전에 MD5 해쉬 값을 적용한 후 전송하여 수신 서버에서 MD5 해쉬 값을 비교하여 무결성 검증과 획득 시스템의 인증을 가능하게 한다.

메모리정보 수집 모듈에서는 프로세스 메모리, 시스템 전체 메모리, 가상 메모리, 페이지 파일을 수집하여 기존에는 수집하지 못한 상세한 휘발성 정보를 수집할 수 있었다. 이렇게 정보의 종류별로 수집하는 모듈을 분류하여 수집함으로써 이후의 유비쿼터스 기반의 컴퓨터 네트워크 통합 포렌식스를 위한 휘발성 정보 분석 및 대응을 효과적으로 수행할수 있다.

다른 휘발성 정보 수집 도구와의 기능 평가 결과 본 연구에서의 휘발성 정보 수집 모듈은 기존의 휘발성 정보 수집 도구가 가지는 모든 기능을 포함하며 추가적으로 기존 도구에서 수집하지 못하는메모리 기반의 휘발성 정보 수집과 시스템 포렌식스와 네트워크 포렌식스에서의 휘발성정보 수집의 연동이 가능하며 법적 증거로서의 유용한 정보를 수집할 수 있다. 그리고 자동화된 증거 수집 기능을 지원하여 전문 수사관이 아닌 일반 사용자도 손쉽게 증거를 수집할 수 있다.

V. 결론

본 연구에서는 휘발성정보 수집 도구와 메모리 수집 기법을 활용하여 유비쿼터스 환경을 포함한 컴퓨팅 환경에서 서비스를 제공하는 서버시스템에서 활용할 수 있는 메모리 기반의 휘발성 정보를 수집하는 방법을 제안하였다.

제안한 모듈의 기능평가 결과 2장의 요구사항을 모두 만족하였으며, 비교 대상인 기존의 휘발성 정보 수집 도구에서는 문제점을 해결하였다. 특히, 지원하지 않는 메모리 기반의 휘발성 정보의 수집, 시스템 포렌식스와 네트워크 포렌식스의 연동 그리고 자동화된 증거 수집이 가능하였다.

그러나 수많은 서버 간의 시스템 시간이 상이할 경우 수집 시점에 달라져 수집되는 정보의 신뢰성에 문제가 발생할 수 있으며 정보의 관리가 어려운 한계점들이 있다. 때문에 향후 휘발성 정보 수집 모듈은 각각의 서버와 통합 관리 서버 간의 시간 동기화 방안에 대해 연구하고자 한다.

참고문헌

[1] Warren G, Kruse II, Jay G. Heiser, *COMPUTER FORENSICS: Incident Response Essenials,* Addison Wesley, 2001.

[2] RFC 3227 Guidelines for Evidence Collecting and Archiving.

[3] Kevin Mandia, Chris Prosise, Matt Pepe, *Incident response and computer forensics, Second Edition,* McGraw-Hill, 2003.

[4] Mariusz Burdach, "Forensic Analysis of a Live Linux System, Pt. 1," *Security Focus,* 2004.

[5] Mariusz Burdach, "Forensic Analysis of a Live Linux System, Pt. 2," *Security Focus,* 2004.

[6] Farmer, Dan, *Forensic discovery,* Addison-Wesley, 2005.

[7] Wistes Venema, Dan Farmer, *TCT(The Coroner's Toolkit),* 1998.

[8] Matt Frye, *TCT Feature Story,* 2005.

[9] Carnegie Mellon University, *Using The Coroner's Toolkit : Harvesting information with grave-robber,* 2001.

[10] Seok-Hee Lee, *A Study of Memory Information Collection and Analysis in a view of Digital Forensics in Window System,* Center for Information Technologies, Korea University, 2006.