안티 포렌식 행위 탐지를 위한 퍼지 전문가 시스템☆

Fuzzy Expert System for Detecting Anti-Forensic Activities

김 세 령* 김 휘 강**
Se Ryoung Kim Huy Kang Kim

요 약

최근 사이버 범죄의 증가와 그 대상 시스템의 다양화로 인하여 디지털 포렌식의 중요성이 커지고 있다. 일부 시스템들은 전원이나 네트워크를 차단하지 않고 수사하는 live forensic의 방법을 채택하고 있는데, 인터넷 사용이 일반화됨에 따라 live forensic 방법이 채택되는 횟수가 증가하고 있다. 그러나 live forensic 기술이 상당한 발전을 거듭하였음에도 불구하고 원격으로 접근하여 행해지는 Anti-forensic 행위에는 여전히 취약한 실정이다. 이와 같은 문제를 해결하기 위하여 첫 번째로 우리는 Anti-forensic 행위를 5개의 계층으로 분류하고 각 계층별로 가능한 Anti-forensic 행위의 시나리오를 생성하는 방법을 제안하였다. 두 번째로 fuzzy 전문가 시스템을 제안하여 효과적으로 Anti-forensic 행위를 탐지할 수 있도록 하였다. 몇몇 Anti-forensic 행위에 사용되는 명령어들은 일반적인 시스템 관리를 위하여 사용되는 명령어와 매우 유사하다. 따라서 우리는 fuzzy logic을 사용하여 모호한 데이터를 다룰 수 있도록 하였다. 미리 정의된 시나리오에서 명령어와 옵션 및 인자 값을 이용하여 룰을 생성하고 fuzzy 전문가 시스템에 이 룰을 학습하도록 하여 유사한 행위가 탐지되었을 때 추론을 통하여 수사관에게 얼마나 위험한 행위인지 알려준다. 이 시스템은 live forensic 수사가 진행될 때 발생할 수 있는 Anti-forensic 행위를 실시간으로 탐지할 수 있도록 하여 증거 데이터의 무결성을 유지하도록 한다.

ABSTRACT

Recently, the importance of digital forensic has been magnified because of the dramatic increase of cyber crimes and the increasing complexity of the investigation of target systems such as PCs, servers, and database systems. Moreover, some systems have to be investigated with live forensic techniques. However, even though live forensic techniques have been improved, they are still vulnerable to anti-forensic activities when the target systems are remotely accessible by criminals or their accomplices. To solve this problem, we first suggest a layer-based model and the anti-forensic scenarios which can actually be applicable to each layer. Our suggested model, the Anti-Forensic Activites layer-based model, has 5 layers – the physical layer, network layer, OS layer, database application layer and data layer. Each layer has possible anti-forensic scenarios with detailed commands. Second, we propose a fuzzy expert system for effectively detecting anti-forensic activities. Some anti-forensic activities are hardly distinguished from normal activities. So, we use fuzzy logic for handling ambiguous data. We make rule sets with extracted commands and their arguments from pre-defined scenarios and the fuzzy expert system learns the rule sets. With this system, we can detect anti-forensic activities in real time when performing live forensic.

☞ keyword : 안티 포렌식(Anti-forensic), 안티 포렌식 행위 모델(Anti-forensic activity model), 라이브 포렌식(live forensic), 퍼지 로직(fuzzy logic), 전문가 시스템(expert system)

* 정 회 원 : 고려대학교 정보보호대학원 석사과정
seryoung82@korea.ac.kr
** 종신회원 : 고려대학교 정보보호대학원 조교수
cenda@korea.ac.kr (교신저자)
[2011/04/15 투고 - 2011/05/02 심사 - 2011/08/04 심사완료]
☆ 본 연구는 지식경제부 및 정보통신산업진흥원의 “대학 IT연구센터 육성·지원사업”의 연구결과로 수행되었음” (NIPA-2011-C1090-1001-0004)
☆ A preliminary version of this paper appeared in ICONI/APIC-IST 2010, Dec 16-20, Mactan Island, Philippines.This version is improved considerably from the previous version by including new results and features.

1. 서 론

최근 수사관들은 포렌식을 할 때, live forensic 기법을 선택해야만 하는 경우가 증가하고 있다. 예를 들면, 조사대상 시스템이 핵심적인 서비스에 이용되고 있어 시스템을 정지시킬 수 없는 경우, 더불어 이 시스템을 조사하기 위해 시스템을 정지시킴으로 인해 발생하는 손해비용이 충분히 큰 경우, 또는 약관이나 SLA(Service Level Agreement)에 명

시된 서비스가용성을 준수해야만 하여 시스템 가동을 중지시킬 수 없는 경우, 수사에 큰 제약이 되고 있다.

이와 같은 환경의 변화에 따라, live forensic의 중요성과 활용 빈도는 점차 높아지고 있으나, live forensic 환경은 용의자들이 증거를 없애기에 좋은 환경이기도 하다. live forensic 환경에서 수집되는 정보들은 일반적으로 메모리에 로드 되어 있는 휘발성 데이터들이다. 수사관들이 live forensic을 수행하는 동안, 조사대상 시스템에 원격지에서 잠입하여 증거 데이터를 손상 및 훼손하는 행위를 알아내는 것이 결코 쉽지 않다.

지금까지의 live forensic 환경에서는 방화벽이나 침입탐지 시스템의 로그 분석을 통한 네트워크 포렌식 방법이 주로 수행되어 왔다. 그러나 네트워크 장비들에 대한 로그 분석은 시간 소요가 크고 즉각적으로 대응하기도 어렵다. 장비들이 내어놓는 로그의 양은 방대하고 형식도 제각각 이기 때문이다[2]. 뿐만 아니라 실제 live forensic에서 발생하는 행위들이 범죄행위를 숨기기 위한 목적으로 수행되는 증거 데이터의 삭제, 암호화, 덮어쓰기 등과 같은 행위임에도 기존의 네트워크 포렌식 연구들은 DARPA dataset과 같은 데이터를 이용하여 수행되어 왔다[2, 3, 4, 18, 19, 20].

하지만 침입탐지 시스템은 본래 Anti-forensic 행위를 탐지하기 위한 도구로 만들어진 것이 아니어서 시스템 관리자 권한으로 정상적인 명령어 수행을 통해 이루어지는 Anti-forensic 행위를 탐지하는 데에는 한계가 있었다.

본 논문에서는 이 같은 문제를 해결하기 위하여 다음과 같은 시스템을 제안하였다. 먼저 조사 대상 시스템의 네트워크 환경을 자동으로 탐색하여 가장 적합한 모니터링 환경을 제공하는 도구인 NFST(Network Forensic Support Tool)를 구현하였고, 미리 발생 가능한 Anti-forensic 행위를 layer기반으로 modeling하고 이를 토대로 Anti-forensic 행위 시나리오를 구성하였다. 이 시나리오는 ruleset으로 가공하여 fuzzy 기반의 전문가 시스템에 적용하였다. 이를 통해 실시간으로 Anti-forensic 행위를 탐지하고 즉각적인 대응이 가능하도록 하였다.

2. 관련 연구

2.1 Live forensic

live forensic은 휘발성 데이터 수집과, 분석의 두 단계로 이루어진다. 더 세부적으로 보면, 수집의 방법은 사용자 인터페이스를 이용하는 방법, imported utility를 사용하는 방법, modified system을 이용하는 방법, 추가적인 하드웨어를 사용하는 방법으로 나뉜다[27].

휘발성 데이터를 수집할 때에는 상대적으로 더 오래 지속되는 데이터를 나중에 수집하는 것을 원칙으로 하며 이와 관련한 정보는 RFC3227 '증거 수집과 저장에 대한 지침'에 나와 있다[1, 22]. 휘발성 정보에는 열려있는 파일목록, 네트워크 정보, 프로세스 정보, 메모리 정보 등이 있으며 수사관들은 주로 CLI(Command Line Interface) 기반의 자동화된 도구를 사용하여 이들 데이터를 수집한다. 일반적인 CLI기반 시스템 관리 명령어들도 사용할 수 있으며, 앞서 언급한 자동화된 도구들 역시 대부분이 명령어들에 기반하고 있다.

메모리 분석의 경우, 프로세스 메모리를 수집하는 방법과 물리 메모리 덤프의 방법이 있다. 수사관은 수사하는 동안 모든 프로세스 목록보다는 특정 프로세스나 소수의 프로세스에만 관심을 가지게 되는 경우도 있는데 이 경우에 가상 메모리나 페이지 파일에서 프로세스가 사용한 모든 메모리를 수집한다. userdump.exe의 경우처럼 어떤 프로세스에도 디버거를 연결하지 않고 프로세스 종료 없이 덤프를 생성하는 방법을 취한다.

물리 메모리 덤프는 메모리 전체를 덤프하는 것으로 현재 메모리 상에 존재하는 모든 정보를 덤프한다. 이 외에 시스템이 기본으로 제공하는 crash dump나 하드웨어 장치를 이용하는 방법 등이 있다.

문제는 live forensic 환경에서 수사관들이 사용하는 여러 가지 도구들이 Anti-forensic을 위해서도 사용할 수 있다는 점이다[23]. 이 도구들이 그러한 목

적으로 사용되는 경우 드러나지 않게 Anti-forensic 행위를 하는 것이 가능해진다. 그러나 수사관이 이 사실을 알기는 매우 어렵다. 실제로 Anti-forensic을 목적으로 수사관을 혼란스럽게 하기 위하여 접근한 것일 수 도 있고, 심지어는 해당 시스템이 조사 중인 것을 모른 채 접근한 것일 수 도 있다. 따라서 수사관은 수사 중 발생하는 모든 접근 행위들에 대해 인지하고 판단해야 할 필요가 있다.

2.2 Anti-forensic

Anti-forensic이란 포렌식 수사 시, 가치 있고 유용한 증거 데이터를 수집하지 못하도록 하는 모든 행위를 일컫는다[15]. Liu 등은 Anti-forensic 행위를 목적에 따라 다음의 4가지로 분류하였다[22, 23].

- 탐지 회피
- 정보 수집 방해
- 사건 처리에 필요한 시간 증가
- 수사 보고서나 법정 증거 자료에 대한 의문 제기

이 외에 포렌식 도구가 사용되었음을 공격자에게 알려주거나 수사관에 대한 공격을 시도하려는 목적을 가지기도 한다[23].

가장 일반적인 Anti-forensic 행위는 데이터를 훼손 및 삭제하는 것이며, 그 중 가장 오래된 방법은 덮어 쓰기 이다. 데이터를 덮어쓰는 도구들은 다음의 3가지 방법 중 하나를 선택한다.

- 저장 매체 전체 덮어쓰기
- 개별 파일에 대한 덮어쓰기
- 이미 삭제되었으나 여전히 저장 매체에 남아 있는 파일들에 덮어쓰기

최근 두드러지게 등장하는 Anti-forensic 기법 중 하나로는 packing기술이 있다. 본래는 실행 파일을 경량화 할 목적으로 등장하였으나 최근 악성코드와 같은 악의적인 목적으로 쓰이는 일이 증가하였다[26]. 알려져 있는 packing도구는 수십 가지에 이르며 암호화된 데이터와 유사하게 분석에 장시간이 소요된다.

이 외에 포렌식 소프트웨어의 crash를 유발하는 compression bombs나 메시지를 이미지 등에 숨겨 찾기 어렵게 하는 Steganography기법, generic data hiding기법 등이 Anti-forensic에 사용되고 있다[12, 13]. 이 중 generic data hiding은 현세대의 포렌식 도구들이 고려하지 않는 접근 불가능 영역에도 데이터를 숨길 수 있어 분석을 매우 어렵게 한다.

지금까지 언급한 방법 및 도구들은 공통적으로 한 가지 치명적인 단점을 가지고 있다. 사용 시 발견되기 쉽다는 것이다. 때문에 최근의 Anti-forensic은 흔적(Footprint)을 최소화 하는데 많은 노력을 기울인다. 실제 디스크에 설치되는 것이 아니라 rootkit처럼 메모리에 상주하도록 만들어진 도구들이나 Virtual Machine을 사용하는 방법, Live CD, 부팅 가능한 USB를 이용하는 방법 등이 그 예이다. 이들 방법을 사용하면 일련의 공격 작업 후 시스템 종료 시, 그 어디에서도 공격의 흔적을 찾을 수 없게 된다.

기존의 디지털 포렌식 연구에서는 static analysis나 네트워크 포렌식만을 고려하고 있기 때문에 live forensic 환경에서 위와 같은 도구들이 사용되는 경우 어떻게 대응할 것인가에 대한 논의는 부족하다. 이러한 이유로 본 논문에서는 live forensic 환경에서의 Anti-forensic 행위에 주목하고 효율적인 대처 방안을 모색하여, 최대한 무결성을 유지하면서 증거 데이터를 수집할 수 있도록 하는 환경을 제공하는데 중점을 두었다.

2.3 Fuzzy logic 기반의 침입 탐지

본 연구에서 제안하려는 시스템과 관련 있는 유사한 연구들은 네트워크 포렌식, 침입 탐지 시스템과 연관되어 진행되어왔다. Niandong 등[2]과 Kim 등[17], Saniee 등[18], Adel 등[19], Zaiqiang 등[20]은 모두 DARPA dataset과 KDD Cup99 dataset을 이용하고 있으며, fuzzy logic을 사용하여 비정상 트래픽 여부를 판단하는 방법론을 택하고 있다. 그러나 이

연구들은 DARPA dataset과 KDD Cup99 dataset을 학습 및 추론의 근거로 사용하고 있기 때문에 명령어 단위의 Anti-forensic 행위를 탐지 하는 데에는 어려움이 따른다. live forensic 환경에서는 DDoS나 Probing과 같은 트래픽뿐만 아니라, 외부로부터의 허가 되지 않은 명령어 및 명령어를 사용한 접근 시도 역시 탐지의 대상이 되므로 위 연구들에서 사용된 dataset만으로는 시스템의 무결성을 유지하기 어려워진다.

fuzzy logic은 사람이 사용하는 언어 표현과 같은 모호한 정보를 다루는데 특화된 알고리즘이다. 앞서 소개한 연구들은 모두 fuzzy logic을 사용하거나 fuzzy logic에 신경망이나 유전자 알고리즘과 같은 다른 알고리즘을 결합하여 사용하고 있다. 그러나 복잡한 연산과 학습이라는 과정이 필요하게 되어 탐지에 소요되는 시간이 늘어나고 구현에 어려움이 따른다. live forensic 및 실시간 대응이라는 환경적 특성을 고려한다면, 연산을 간단하게 하고 수행 속도를 높일 수 있도록 하는 경량화의 과정이 필요하다.

앞서 소개한 선행 연구들은 모두 네트워크 포렌식에 치중하고 있는 만큼 해당 트래픽이 얼마나 위험한가에 대한 판단보다 audit data와 얼마나 유사한가와 같은 결과를 보여준다. 따라서 수사관은 해당 트래픽이 얼마나 위험하고, 어디에 영향을 끼치며, 어떤 대응을 필요로 하는지 알기 어렵다. 외부로부터 허가되지 않은 접근이 있음에도 그 사실 외에 알 수 있는 정보가 없다는 것은 적합한 대응이 어려움은 물론 증거 데이터의 무결성마저 지킬 수 없다는 것을 의미한다.

본 논문에서 제안하는 시스템은 명령어 단위의 Anti-forensic 행위 및 접근을 탐지할 뿐만 아니라, 얼마나 위험한지, 어디에 영향을 끼칠 수 있는지에 대한 정보도 알려준다. fuzzy logic의 연산을 단순화하여 즉각적인 탐지와 대응이 가능하게 하였으며, 조사 대상 시스템의 네트워크 상태를 자동으로 검사하고 가장 적합한 트래픽 모니터링 방법을 제시하여 줄 수 있도록 하였다.

```
$ cat > .forward
| "cp /bin/sh /home/gk/my_shell ; chmod 755
/home/gk/my_shell"
$ cat .forward
| "cp /bin/sh /home/gk/my_shell ; chmod 755
/home/gk/my_shell"
$ echo hello chump | mail gk@targetsystem.com
```

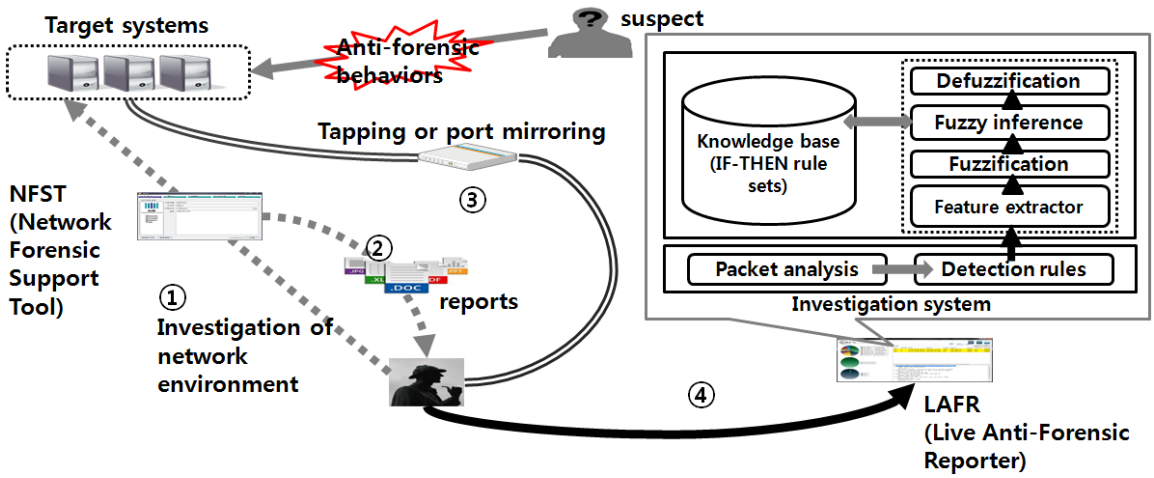
(그림 1) 모호한 명령어의 예

3. Framework

Anti-forensic 행위의 주목적은 수사관의 수사를 방해하는 것이다. live forensic이 수행되는 환경에서 수집되는 데이터들은 대부분 휘발성 데이터들이기 때문에 Anti-forensic 행위가 수사에 미치는 영향력은 매우 클 수밖에 없다. 만약 실시간으로 Anti-forensic 행위를 모니터링 하는데 실패한다면, 용의자는 수사관이 증거를 수집하기도 전에 데이터를 삭제할 수 도 있다.

용의자들은 대부분 범죄 행위를 숨기기 위해 로그를 삭제하거나 데이터를 훼손하는데, 이 때 사용되는 명령어나 도구들은 정상적인 시스템 운영을 위하여 사용되는 것들과 매우 유사하다. 만약 (그림 1)과 같은 명령이 수행된 경우 이 명령이 Anti-forensic 행위가 될지 여부는 누구도 정확히 이야기할 수 없을 것이다[6]. cat이나 cp, chmod등의 명령은 정상적인 시스템 관리 시에도 일상적으로 사용되는 명령어들이며, forward, /bin/sh등과 같은 파일 및 경로도 일반적으로 사용되는 것들이다. 그러나 메일의 forward파일에 자신이 만든 shell을 복사해 두고, 권한을 755로 설정 한 후, 메일로 보내는 행위는 그 흐름상 매우 의심스럽다. 만들어진 shell이 어떤 행위를 할 지 알 수 없기 때문이다. 이 같은 이유로 Anti-forensic 행위 탐지의 정확도를 높이기 위해서는 명령어뿐만 아니라 명령어에 따라 나오는 옵션, 접근하는 파일의 경로 및 시퀀스를 함께 분석할 필요가 있다.

live forensic 환경에서 신뢰성 있는 Anti-forensic 행위 탐지를 위해서는 수사 대상 시스템으로 유입



(그림 2) live forensic 환경에서의 anti-forensic행위 탐지 시스템 구조

되는 네트워크 packet을 모니터링 하여 확인해야 한다. 수사 대상 시스템은 반드시 PC가 아닐 수 도 있고, 같은 건물에 존재하지 않을 수 도 있으며, 원격지에 있어 직접적인 접근이 불가능 할 수 도 있다. 그러므로 packet을 어떤 방법으로 수집하고 감시할 것인지 결정하는 의사 결정 과정이 필요하다.

본 연구에서는 자동화된 네트워크 환경 분석 프로그램을 개발하여 수사관에게 가장 적합한 packet 모니터링 방법을 제시하여 주도록 하였다. 모니터링 된 packet은 분석과정을 통해 악의적인 의도가 담긴 것인지 여부를 검증 받는다. 확인 결과는 fuzzy 기반의 전문가 시스템에 입력되며, 어느 정도의 위험성을 가지는지를 결과로 보여준다. (그림 2)는 이와 같은 과정의 전반적인 흐름을 보여준다.

이전에 언급한 바와 같이 우리는 Anti-forensic 행위의 시나리오를 생성하여 미리 가능한 행위들을 예측할 수 있게 하는 것을 목적으로 한다. 이를 위하여 시나리오를 생성하기 전, 가능성 있는 행위들을 layer기반으로 모델링 하였다. 크게 Data layer, DB/Application layer, OS/Platform layer, Network layer, Physical layer의 5계층으로 분류하였으며, 이를 AFA(Anti-Forensic Activity)모델이라고 명명 하였다. 이 모델을 기반으로 live forensic 환경에서 발생할 수 있는 Anti-forensic 행위의 시나리오를 구성하여 효율성과 실효성을 도모하였다. (표 1)은 AFA 모델을 나타낸 것이다.

(표 2), (표 3), (표 4)는 Data layer, OS/Platform layer, Database layer에서 Anti-forensic 행위에 사용될 가능성이 있는 명령어들을 보여준다. 표에서 확인 할 수 있는 것과 같이 의심스러운 행위들에 사용되는 명령어들은 모두 정상적인 시스템 운용을 위해 사용할 수 있는 명령어들이다.

4. 탐지 룰의 생성과 NFST의 구현

Anti-forensic 행위의 주 목적은 수사관의 수사를 방해하는 것이다. live forensic이 수행되는 환경에서 수집되는 데이터들은 대부분 휘발성 데이터들이기 때문에 Anti-forensic 행위가 수사에 미치는 영향력은 매우 클 수밖에 없다. 만약 실시간으로 Anti-forensic 행위를 모니터링 하는데 실패한다면, 용의자는 수사관이 증거를 수집하기도 전에 데이터를 삭제할 수 도 있다.

live forensic 환경에서의 수사는 시스템에 대한 직접적인 접근을 통한 방법 보다는 간접적인 방법을 이용할 필요가 있다. 이는 수사 중에도 의도하지 않게 증거 데이터의 무결성을 훼손하게 되는 상황이 발생 할 수 있기 때문이다. 본 연구에서는 위

(표 1) AFA model

Layer	Scenarios	Related protocols or commands
Data	Anti-forensics activities targeted on data deleting, hiding and modifying **e.g.** disk encryption tool, erase tool, disk utilities	OS commands, encryption tool
Database/ Application	Anti-forensics activities targeted on application and Database system **e.g.** deleting or modifying DBMS audit log, DB tables, changing DBA account password, deleting app. log	OS commands, SQL query
OS/ Platform	Anti-forensics activities targeted on OS system **e.g.** Changing administrator's password, deleting or modifying system log	OS commands, telnet, ssh, etc
Network	Anti-forensics activities targeted on network devices and network security system **e.g.** deleting network router's log., DOS attack to network monitoring system	TCP/IP, dial-up modem
Physical	Anti-forensics activities targeted on physical changes **e.g.** Changing BIOS password, changing RAID disk configuration in the BIOS menu	Physical access or KVM over IP

(표 2) Data layer의 모호한 명령어 예시

Scenario category	Related commands in Linux/Unix	Related commands in Windows
Explore and modify	find, vi, cat	dir /d, edit
delete	rm –rf, rmdir	del, rd, format
Change permission	chmod, umask	Icacls
Change or erase disk	fdisk, newfs	diskpart, manage -bde
Encrypt file	openssl, crypt, vi -x	Truecrypt

(표 3) OS/Platform layer의 모호한 명령어 예시

Scenario category	Related commands in Linux/Unix	Related commands in Windows
Obtain root account	su, sudo	N/A
Modify password	passwd	PsPasswd, netuser
Delete system log	./zap2 webmaster	Forfiles, PsLogList
Modify the system time	date, rdate	time, date
Modify or delete registry	N/A	reg add, reg del, reg export

(표 4) Database layer의 모호한 명령어 예시

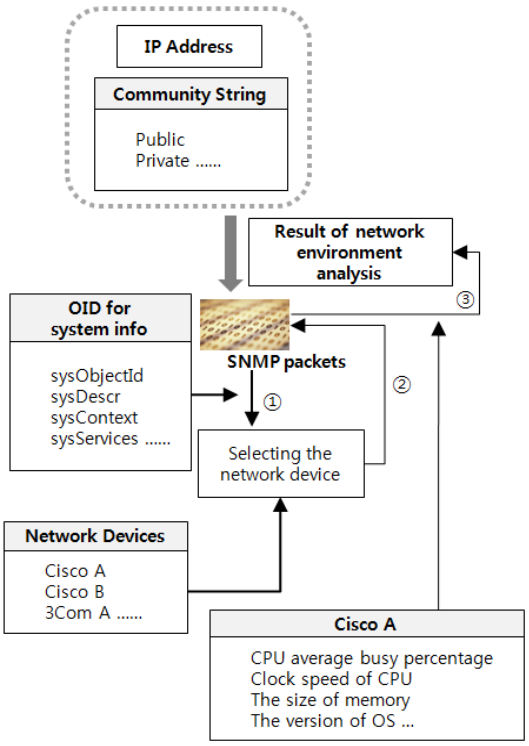
Scenario category	Related commands in Windows
Modify DBA's passwd	SET PASSWORD FOR .....
Delete account	DELETE FROM user WHERE .....
Delete table	DELETE FROM table_name TO .....
Delete record	DELETE FROM table_name WHERE .....
Delete DB	DROP DATABASE.....
Modify or delete log	DBCC SHRINKFILE (...) Exec sp_configure 'default trace enabled', 0

의 조건을 만족시키기 위하여 수사 대상 시스템으로 으로 inbound되는 네트워크 packet을 수집, 분석할 수 있는 도구로 ruleset을 최적화한 IDS(Intrusion Detection System)를 활용하였다. 또한 대상 시스템에 간접적으로 접근할 때 최적의 조건을 찾아 수사관에게 알려주는 네트워크 환경 분석 도구인 NFST를 구현하였다.

4.1 NFST의 구현

수사관은 본격적인 수사에 앞서 조사 대상 시스템의 환경을 분석할 필요가 있다. 이를 위하여 관리자와 인터뷰를 하거나 용의자에게 직접 묻는 방법, 직접 조사 하는 방법 등을 동원한다. 그러나 이와 같은 조사 방법은 상당히 많은 시간을 필요로 하며, 수사관을 지치게 하여 수사 하고자 하는 의욕을 떨어뜨린다. 만약 조사 대상 시스템의 관리자가 없거나 용의자가 묵비권을 행사하고 있다면 조사 대상 시스템의 환경을 조사하는데 소모되는 시간은 더 늘어나게 될 것이다. 우리는 이와 같이 의사결정에 소요되는 시간을 줄이고 가장 효율적인 방법으로 대상 시스템에 접근하여 수사를 진행 할 수 있도록 도와주는 NFST(Network Forensic Support Tool)를 구현하였다.

SNMP는 네트워크 관리 프로토콜이며 시스템이나 네트워크 관리자로 하여금 원격으로 네트워크 장비를 모니터링 하고 환경 설정 등의 운영을 가능하게 하는 프로토콜이다. NFST는 바로 이 SNMP를 이용하여 대상 네트워크 내에 어떠한 장비가 있는지, 현재 메모리 상태나 CPU상태는 어떠한지, 여유 포트는 있는지, tap장비를 필요로 하는지 등의 정보를 수집한다. SNMP가 이들 정보를 수집하기 위해서는 sysObjectID라는 고유 식별자를 이용하는데, 이것은 네트워크 상에 연결되어 있는 모든 디바이스에 부여되어 있다. 따라서 이 식별자를 이용하면 해당 디바이스에 대한 상세 정보를 획득할 수 있다. 이렇게 획득한 정보를 토대로 시스템의 상태를 판단하여 최적의 트래픽 모니터링 및 capture방법을 수사관에게 보고서 형태로 출력하여 보여준다. (그림 3)은 이와 같은 NFST의 의사 결정과정을 그림으로 나타낸 것이다. (표 5)와 (표 6)은 조사 대상 장비가 3Com스위치에 연결되어 있는 경우에 sysObjectID를 이용하여 획득한 CPU와 메모리의 정보들을 나타낸다. 3Com 스위치 장비의 경우 sysObjectID가 1.3.6.1.4.1번으로 시작되며 이 번호 하위에 있는 번호들로 해당 장비를 구성하고 있는 장치들에 대한 상세 정보를 얻어 올 수 있다.



(그림 3) 네트워크 환경 분석 과정

4.2 최적화된 탐지 rule의 생성

침입 탐지 시스템은 본래 포렌식을 위한 도구로 개발된 것이 아니므로 live forensic 환경에서 그대로 사용하는 것은 적합하지 않은 방법이다. 따라서 Anti-forensic 행위를 제대로 탐지해 낼 수 있도록 ruleset을 최적화 할 필요가 있다. 본 연구에서는 모두 36가지의 시나리오를 생성하였으며, AFA모델에 기반을 두었다. Anti forensic 행위에 사용한 명령어

(표 5) sysObjectID를 이용하여 수집한 CPU정보

sysObjectID	information
.1.3.6.1.4.1.2011.2.2.4.12	The system CPU average busy percentage in the last 5 seconds period.
1.3.6.1.4.1.2011.10.2.6.1.1.1.1.7	The threshold for the CPU usage
.1.3.6.1.4.1.2011.10.2.75.2.1.5.1.7	Represents the CPU type of AP
.1.3.6.1.4.1.2011.10.2.6.1.1.1.1.24	The CPU frequency of entity
.1.3.6.1.4.1.2011.10.2.75.2.1.5.1.8	The clock speed of CPU

(표 6) sysObjectID를 이용하여 수집한 메모리정보

sysObjectID	information
.1.3.6.1.4.1.2011.2.2.5.1	The use size of the memory
.1.3.6.1.4.1.2011.10.9.1.2.1.11	The threshold for the Memory usage
.1.3.6.1.4.1.2011.10.2.6.1.1.1.1.10	The size of memory for the entity
.1.3.6.1.4.1.2011.10.2.6.1.1.1.1.28	The memory type of entity

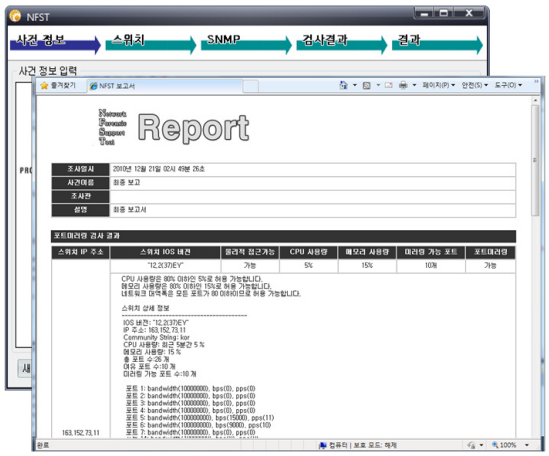
를 가장 첫 번째 탐지 대상으로 하여 추가적으로 명령어의 옵션 및 대상 경로를 함께 탐지할 수 있도록 하였다. 이와 같은 접근 방식은 결국 audit data와 얼마나 유사한가를 보기보다 해당 명령어가 얼마나 위험한 것인가로 판단하게 하는 기초가 된다.

(표 7)은 live forensic 환경에서 Anti-forensic 행위를 탐지하는데 최적화 시켜 생성한 ruleset의 예이다.

5. Fuzzy 전문가 시스템

5.1 전문가 시스템

전문가 시스템은 주어진 질문에 대하여 지식 베이스에 저장된 정보를 근거로 추론 후 답을 제공하는 시스템이다[7]. 전문가 시스템은 특별히 신뢰할



(그림 4) NFST (Network Forensic Support Tool)

수 없는 데이터, 불완전한 전제로부터 결과를 추론해 낼 수 있다. 전문가 시스템의 지식 베이스는 전문가의 자문에 의하여 이미 정답이 알려진 데이터들로 구성된다. 이 데이터들은 rule의 형태를 가지며 학습되지 않은 질문에 대하여 답을 할 때 추론의 근거가 된다. 추론에 사용되는 알고리즘은 매우 다양하게 존재하는데, 몇 가지 예로 인공 신경망 알고리즘, 유전자 알고리즘, fuzzy logic 등을 들 수 있다. 본 연구에서는 모호한 데이터들에 대한 판단을 필요로 하므로 그에 가장 적합한 fuzzy logic을 선택하였다.

5.2 Fuzzy logic

fuzzy logic은 1965년 Lotfi A. Zadeh가 처음 제안하였다. 연구의 출발은 그의 아내가 예쁜 정도를 컴퓨터가 표현하도록 하려면 어떻게 해야 하는가 하는 질문에서 시작되었다. 실제로 fuzzy logic은 명확한 판단을 내리기 어려운 문제나 주관적 판단에 바탕을 둔 모호성을 대상으로 한다. 이는 인간의 언어 및 사고와 관련한 모호함을 의미한다. fuzzy logic은 확률론을 기반으로 하여 각 대상이 어떤 집합에 대하여 속한다, 속하지 않는다 라는 논리로부터 각 대상을 그 모임에 속하는 정도로 이해하여 일반화한다.

(표 7) 실시간 Anti-forensic행위 탐지에 최적화된 ruleset

.Rules	Alert log	Meaning
alert tcp any any -> any any (msg:"VNC server response"; flow:established; content:"RFB 0"; depth:5; content:".0"; depth:2; offset:7; classtype:misc-activity; sid:1000170;)	[**] [1:1000170:0] VNC server response [**][Classification: Misc activity] [Priority: 3]09/07-19:13:27.882616 0:50:56: C0:0:8 -> 0:C:29:54:72:94 type:0x80 len:0x42 192.168.88.1:58522 -> 192.168.88.130:5900 TCP TTL:128 TOS:0x0 ID:3611 IpLen:20 DgmLen:52 DF ***AP*** Seq:0x2C151E93 Ack:0x79630D80 Win:0xFAE4 TcpLen:20	If protocol is TCP, port number is any and packet is for VNC response. Alert log shows the consequence of detection. The VNC response packet is detected on 7 Sep. Source ip is 192.168.88.1 and destination ip is 192.168.88.130
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"3_NETWORK_COMMAND / confirm network information"; content:"net sessions"; sid:1000160;)	[**] [1:1000165:0] network information check [**][Priority: 0] 09/07-18:24:27.566548 0:50:56:C0:0:8 -> 0:C:29:54:72:94 type:0x800 len:0xF8 192.168.88.1:58151 -> 192.168.88.130:23 TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:234 ***AP*** Seq: 0x7435778D Ack: 0xBD279BE9 Win: 0x1700 TcpLen: 20	If protocol is TCP, port number is any and command is 'net sessions'. Alert log shows the result of detection. The 'net sessions' command is detected on September 7. Source ip is 192.168.88.130 and destination ip is 192.168.88.1.

(표 8) Categories of commands

Category	Description	Examples
Caution	Commands related to deletion, encryption, permission, shutdown and account	rm –rf chmod net user shutdown init
Attention	Commands related to changing time, confirmation about system information, starting or shutdown service (process), editing file and exploring files.	touch taskkill ipconfig find vi finger gcc
Common	Commands that are excluded in Caution and Attention categories.	ls cd

앞 절에서 언급하였던 바와 같이 live forensic 환경에서의 Anti-forensic 행위는 정상적인 시스템 관리를 위하여 사용되는 도구 및 명령어와 매우 유사하다. 그러므로 탐지된 Anti-forensic 행위가 얼마나 수사 환경에 해를 끼칠 수 있는 것인지 판단하는데 fuzzy logic은 매우 적합한 알고리즘이라고 할 수 있다[7].

fuzzy logic에서는 집합의 개념을 사용하는데, 이를 fuzzy 집합 이라고 한다. fuzzy 집합은 어떤 membership에 속한 정도의 값(grade of membership)을 가지는 대상의 집합이다[28]. 확률에 기반하고 있으므로 그 범위는 [0, 1]이다. 1에 가까운 값을 가질수록 grade of membership이 높음을 의미한다. 이 집합은 membership에 의해 그 성격이 결정된다. 집합의 값을 좌표축에 대응 시키면 함수를 얻을 수 있는데 이 함수를 membership함수라고 한다.

5.3 Feature selection

앞서 작성한 시나리오에서와 같이 Anti-forensic 행위들이 드러나지 않게 동작하기에 유리한 CLI (Command Line Interface)기반의 명령어나 도구들로 행해진다고 알려져 있다. 이들은 명령어와 해당 인자 값으로 구성되어 있다. 따라서 입력된 명령어와 인자 값을 확인하여 위험한 정도를 추론해 낼 수 있도록 3가지 변수를 정의하였다. (1)은 3가지 변수의 집합을 나타낸 것이다. (표 8), (표 9)는 각 변수의 범위와 그 예를 나타낸다.

(표 9) Categories of command arguments

Category	Description	Examples
Very Suspicious	System configuration files, system log files and configuration files related to account, logging and database	/etc/passwd /etc/sysconfig /network /var/log /var/admin username (followed by admin command)
Suspicious	Application related files except database files and source files	shellcode, utilities that can be used by users.
Possibly Suspicious	The files that generated by user	document file

$$FZ_{command} = \{Caution, Attention, Common\}$$
$$FZ_{argument} = \{Very\ suspicious, Suspicious, Possibly\ suspicious\}$$
$$FZl_{evelofdanter} = \{Very\ high, High, Middle, Low\} \quad (1)$$

Anti-forensic 행위에 사용되는 명령어 및 옵션들이 나타내는 위험의 정도는 정규분포를 따른다. 예를 들어 데이터 조회, 수정 및 생성 명령어인 vi는 어떻게 사용되느냐에 따라서 아주 위험한 명령어 군에 속할 수 도 있고 주의해야 할 명령어 군에 속할 수 도 있다. live forensic환경에서 시도 되는 명령어는 어떤 식으로든 시스템의 무결성을 유지할 수 없도록 만들기 때문에 임의의 파일을 조회하거나 경로를 탐색하는 것과 같은 명령어들을 제외한 거의 모든 명령어들이 위험하거나 매우 위험한 정도에 속하게 된다. 결국 명령어 및 옵션의 위험도를 표현하는 fuzzy 집합에서 매우 위험하거나 위험함을 나타내는 집합에 속하는 비율이 위험하지 않음을 나타내는 집합에 속하는 비율 보다 훨씬 많아지게 되는 것이다. 본 연구에서는 이와 같은 분포를 정확하게 나타낼 수 있고 upper limit값과 lower limit값 그리고 mode값을 알고 있는 경우 가장 적합한 형태인 삼각형 모양의 membership함수를 선택하였다[29]. fuzzy logic의 범위는 [0. 1]이므로, upper limit값은 1이 되고 lower limit값은 0이다. mode값은 upper limit 값과 동일한 1이 된다. (그림 5)는 명령어와 인자 값, 위험도에 대한 membership함수의 그래프 및 fuzzy 추론의 결과를 나타내는 그래프를 보여준다.

5.4 지식 베이스

fuzzy logic에서의 지식 베이스는 IF-THEN rule을 따른다. 명령어에 해당하는 조건이 3가지, 인자 값에 해당하는 조건이 3가지, 결과인 위험도에 해당하는 조건이 4가지로 총 36개의 rule이 생성가능하며, 그 중 12개의 rule을 학습에 사용하였다. 다음은 본 연구에서 사용한 rule의 일부이며, X는 명령어를 Y는 인자 값을, Z는 위험도를 나타낸다.

R1 : If X is less than 3(Caution) and Y is less than 3(very suspicious) then Z is Very High

R3 : If X is less than 3(Caution) and Y is greater than 5 and less than 7(possibly suspicious or suspicious) then Z is Middle

R4 : If X is less than 3(Caution) and Y is greater than 7(suspicious) then Z is Low

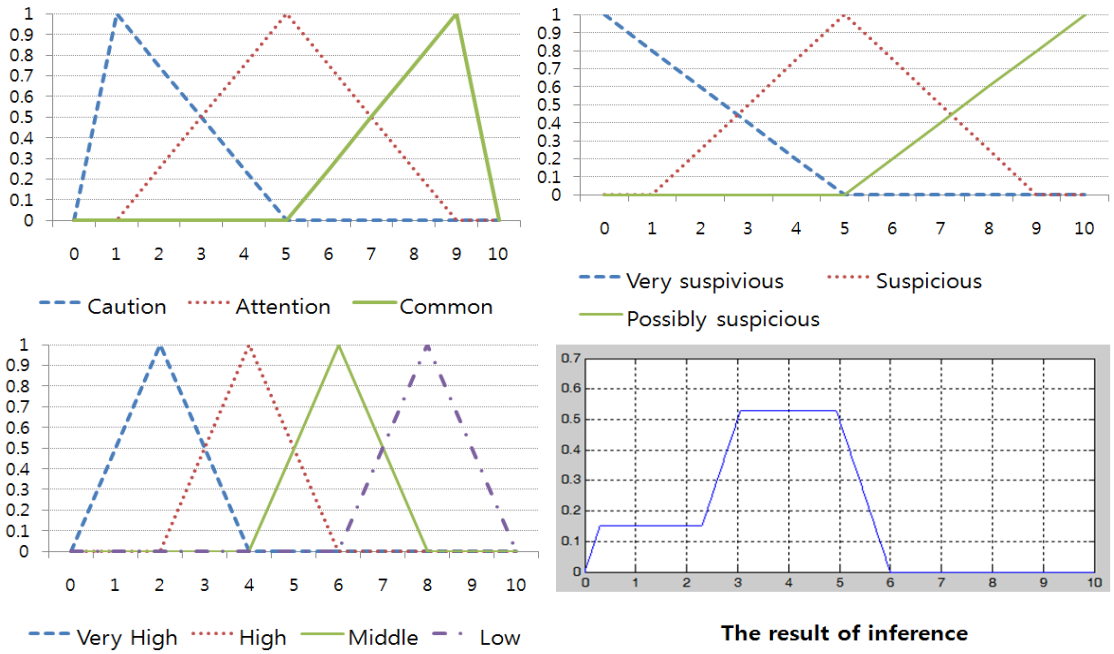
R5 : If X is greater than 3 and less than7(Caution or Attention) and Y is less than 3(very suspicious) then Z is Very High

R7 : If X is greater than 3 and less than7(Caution or Attention) and Y greater than 5 and Y is less than 7(possibly suspicious or suspicious) then Z is Middle

R8 : If X is greater than 3 and less than7(Caution or Attention) and Y is greater than 7(suspicious) then Z is Low

R9 : If X is greater than 7(Common) and Y is less than 3(very suspicious) then Z is Very High

R11 : If X is greater than 7(Common) and Y is greater than 5 and less than 7(possibly suspicious or suspicious) then Z is Middle



(그림 5) membership 함수와 추론 결과 그래프

R12 : : If X is greater than 7(Common) and Y is greater than 7(suspicious) then Z is Low

rule에 사용된 숫자는 membership함수의 값을 나타내며 X값에 대하여 각 fuzzy membership함수의 어떤 결과 값과 대응되는가에 의해 Z값이 결정된다. 이러한 과정을 추론이라 한다.

5.5 추론

fuzzy logic의 추론 알고리즘은 직접법과 간접법으로 나뉜다. 직접법은 다시 mamdani추론 법, 선형 추론 법, 간략 추론 법, 변형된 선형(혼합)추론 법으로 나뉜다. 본 연구에서는 명령어와 인자 값에 해당하는 membership함수에서 겹치는 영역이 있는 경우 작은 값을 선택하여 위험도를 산출하는 방식을 선택하였다. 이와 같은 연산에 가장 적합하도록 고안된 것이 mamdani 추론법이다[8, 9]. 이 추론법을 IF-THEN rule에 적용하는 방법은 다음과 같다. IF-THEN rule이 (2)와 같다고 가정한다. Rule1과 Rule2는 실행된 명령어의 시퀀스로 정의할 수 있으며, X는 명령어, Y는 명령어의 옵션, Z는 해당 명령어와 옵션의 전체적인 위험도를 의미한다. A, B, C는 각각 그 위험도를 수치로 나타낸 값이라고 볼 수 있다. (5.4의 rule참조) Rule1과 Rule2가 반드시 상관관계를 가져야 하는 것은 아니다. 이제(3)과 같은 방법으로 각 rule의 적합도를 구한다. (3)의 식에서 μ는 membership function을 의미하며 아래 첨자가 A인 것은 명령어에 대한 memebership function을, B인 것은 argument의 membership function을 의미한다. (4)에서는 (3)에서 구한 적합도를 THEN절의 fuzzy 집합에 반영하여 추론의 결과를 구한다. 최종적인 추론의 결과는 (5)와 같이 구해지는데, 이 값은 fuzzy한 값이므로 다시 크리스프(crisp)값으로 변경해 주어야 한다. (6)과 같은 무게 중심법(center of gravity)을 사용하여 구한다[8, 9].

Rule1 : IF X is A1 and Y is B1 THEN Z is C1
Rule2 : IF X is A2 and Y is B2 THEN Z is C2

(2)

$$W_1' = [\max_{(x)} \mu_{A1}(x) \wedge \mu_{A'}(x)] \wedge [\max_{(x)} \mu_{B1}(x) \wedge \mu_{B'}(x)]$$
$$W_2' = [\max_{(x)} \mu_{A2}(x) \wedge \mu_{B'}(x)] \wedge [\max_{(x)} \mu_{B2}(x) \wedge \mu_{B'}(x)] \quad (3)$$

$$\mu_{c'1}(z) = W_1' \wedge \mu_{c1}(z), \quad \forall z \in Z$$
$$\mu_{c'2}(z) = W_2' \wedge \mu_{c2}(z), \quad \forall z \in Z \quad (4)$$

$$\mu_c(z) = \mu_{c'1}(z) \vee \mu_{c'2}(z) \quad (5)$$

$$z_0 = \frac{\int \mu_c(z) \cdot z dz}{\int \mu_c(z) dz} \quad (6)$$

5.6 Experimental results

지금까지 논의한 원리들을 바탕으로 실제 시나리오들을 테스트 해 보았다. 전문가 시스템 이므로 해당 명령어와 옵션에 대한 위험도는 미리 정해 두었다. 위험도는 파일의 손상 및 훼손의 정도와 비례한다. 파일에 대한 정보를 변경하거나 시스템의 정보를 변경하는 행위도 데이터 훼손에 포함시켰다. 12개의 rule을 이용하여 추론해낸 위험도가 미리 지정해 둔 위험도와 얼마나 일치하였는가 하는 것으로 정확도를 계산한다.

첫 번째 시나리오는 용의자가 데이터베이스에 접근하여 회사의 기밀 데이터를 추출하고 침입의 흔적을 지우는 것이다.

```
$vi su.c
$gcc –o exe_su exe_su.c
$./exe_su
#who
#mysql_safe –skip-grant &
Mysql> show databases;
Mysql> use employees
Mysql> select name from employees where
salary > 300
#./zap2 guest
#./zap2 root
```

두 번째 시나리오는 침입 탐지 시스템을 kill하고, root권한의 사용자를 추가하는 shellcode를 사용한 경우이다. 해당 파일을 컴파일하고 실행 한 후, 파일을 숨기고 있다. 그러나 어떤 내용을 담고 있는지 겉으로 보이지 않기 때문에 의심스러운 행위를 하고 있음에도 불구하고 정확하게 침입 행위인지 판단하기 어렵다.

```
$gcc executable
$./executable
$mv executable  .executable
```

세 번째 시나리오는 용의자가 조사 대상 시스템을 감시 하면서 필요한 정보들을 수집할 수 있게 하는 경우이다. 조사관이 어떤 명령어를 사용하고 있는지 어떤 프로그램을 사용하고 있는지 등을 볼 수 있다. 이 경우는 직접적으로 시스템에 해악을 끼치지는 않지만 조사관이 어떤 작업을 수행하고 있는지 감시가 가능하기 때문에 눈에 띄지 않게 원하는 작업을 수행할 수 있는 환경을 제공 받을 수 있게 된다[6].

```
$ xscan quake
$ tail –f KEYLOG.quake:0.0
# xlswins –display quake:0.0 | grep –I snort
# xwatchwin quake –w 0x1000561
[quake]$ xwd –root –display localhost:0.0 >
dump.xwd
```

(표 10), (표 11), (표 12)는 위의 시나리오들을 테스트한 결과이다.

6. 결 론

본 논문에서 는 계층기반의 AFA모델을 제안하고 그에 따라 Anti-forensic 행위 시나리오를 생성하였다. NFST를 구현하여 수사 대상 시스템에 대한 최적의 수사 환경을 제공하도록 하였으며, 침입 탐지 시스템과 fuzzy 전문가 시스템을 결합하여 실시간으로 Anti-forensic 행위를 탐지 할 수 있게 하였다. 우리가 제안한 시스템의 이점은 다음과 같다.

(표 10) 첫 번째 시나리오에 대한 테스트 결과

Sequence of the commands	mark	Command argument	mark	Degree of danger
vi → gcc	Caution	-o	Suspicious	High
gcc → exe_su	Attention	-o ,exe_su, exe_su.c	Suspicious	High
exe_su → who	Attention	N/A	Possibly suspicious	**High**
who → mysql_safe	Caution	--skip-grant &	Very suspicious	**Low**
mysql → show	Attention	N/A	Possibly suspicious	Low
show→ use	Attention	N/A	Possibly suspicious	Low
use → select	Attention	From, where	Possibly suspicious	Low
select → zap2	Caution	Guest	Very suspicious	**Middle**
zap2→ zap2	Caution	guest, root	Very suspicious	Very High

(표 11) 두 번째 시나리오에 대한 테스트 결과

Sequence of the commands	mark	Command argument	mark	Degree of danger
gcc → executable	Attention	Executable	Possibly suspicious	**High**
executable → mv	Attention	executable, .excutable	Possibly suspicious	**High**

(표 12) 세 번째 시나리오에 대한 테스트 결과

Sequence of the commands	mark	Command argument	mark	Degree of danger
xscan → tail	Caution	quake, -f	Very suspicious	Very High
tail → xlswins	Caution	-f, -display	Very suspicious	Very High
xlswins → xwatchwin	Caution	-display, -w	Very suspicious	**Middle**
xwatchwin → xwd	Attention	-w, -root, -display	suspicious	Middle

첫 번째, 가변적인 수사 환경에서 가장 적합한 방법으로 Anti-forensic 행위를 탐지할 수 있도록 자동화 하였다. NFST프로그램은 기본적인 수사 정보 입력 과정만 거치면 수사 대상 시스템의 네트워크 환경을 자동으로 조사하여 그 결과를 알려준다.

두 번째, 정상적인 시스템 운용 명령어와 구분이 명확하게 지어지지 않는 모호한 명령어들까지 탐지가 가능하도록 하였다. 또한 명령어와 해당 명령어의 인자 값에 대한 다음 시퀀스까지 분석 하여 탐지의 정확도를 높일 수 있도록 하였다. 뿐만 아니라 실제 live forensic 환경에서 발생 할 수 있는 가능한 모든 Anti-forensic 행위를 탐지 할 수 있도록 하였다. 이는 실제 현업에서 우리가 제안한 시스템을 바로 적용할 수 있음을 시사한다.

세 번째, 비교적 간단한 fuzzy 연산을 사용하여 빠른 연산을 가능하게 하였다. 이는 신속하고 효율적인 모니터링을 가능하게 한다. 모니터링의 결과는 보고서로 출력하여 수사관이 바로 확인 하는 것이 가능하다.

앞으로의 연구는 유포중인 악성코드 탐지나 악의적인 스크립트 코드 업로드 등과 같이 사람이 수행하는 행위가 아닌, 네트워크 상에서 자동적으로 수행되는 행위도 함께 탐지 할 수 있도록 하는 방향으로 진행되어야 할 것이다.

참 고 문 헌

[1] Harlen Carvey, “Windows Forensic Analysis”, Syngress Publishing, ISBN 1597494224, 2009

[2] Niandong Liao, Shengfeng Tian, Tinghua Wang, “Network forensic based on fuzzy logic and expert system”, Computer Communications, vol.32, issue17, pp.1881-1892, 2009.

[3] Emmanuel S.Phill, R.C.Joshi, Rajdeep Niyogi, “A Generic Framework for Network Forensic”, International Journal of Computer Applications, vol.1. no.11, 2010.

[4] Christian S.J. Peron, Michael Legary, “Digital Anti-Forensic: Emerging trends in data transformation techniques”

[5] Frank Adelstein, “Live Forensic, Diagnosing your system without killing it first, Communications of the ACM, vol.49, no.2, 2006.

[6] Stuart McClure, Joel Scambray, George Kurtz, “Hacking Exposed : Network Security Secrets and Solutions”, Third Edition, McGraw-Hill, ISBN 007149426X, 2007

[7] L.A. Zadeh, “The role of fuzzy logic in the management of uncertainty in expert system” Fuzzy Sets and Systems, vol.11, issue1-3, pp.197-198, 1983.

[8] E.H. Mamdani, S. Assilian, “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller”, International Journal of Man-Machine Studies, vol.7, issue1, pp.1-13, 1975

[9] E.H. Mamdani, “Advances in The Linguistic Synthesis of Fuzzy Controllers”, International Journal of Man-Machine Studies, vol.8, issue6, pp.669-678, 1976

[10] Srinivas Mukkamala, Andrew H. Sung, “Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Technologies” International Journal of Digital Evidence, vol.1, issue4, 2003.

[11] Zaiqiang Liu, Dengguo Feng, “Incremental Fuzzy Decision Tree-Based Network Forensic System”, Computational Intelligence and Security, vol.3802, pp.995-1002, 2005.

[12] http://computer-forensics2.sans.org/blog/2008/11/19/

[13] Nicolas Ruff, “Windows Memory Forensics”, Journal in Computer Virology, vol.4, no.2, pp.83-100, 2007.

[14] Bryan Sartin, “Anti-Forensics – Distorting the Evidence”, Computer Fraud & Security, vol.2006, issue5, pp.4-6, 2006.

[15] Paul A. Henry, “Anti-Forensics”, Secure computing

[16] Ryan Harris, “Arriving at an anti-forensic consensus: Examining how to define and control the anti-forensics problem”, Digital Investigation, vol.3, supplement1, pp.44-49, 2006.

[17] Jung-Sun Kim, Dong-Geun Kim, Bong-Nam Noh, “A Fuzzy Logic Based Expert System as a Network Forensic”, 2004 IEEE International Conference on, vol.2, pp.879-884, 2004.

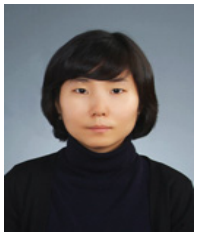
[18] M.Saniee Abadeh, J.Habibi, C.Lucas, “Intrusion Detection using a Fuzzy genetics-based Learning Algorithm”, Journal of Network and Computer Applications, vol.30, issue1, pp.414-428, 2007.

[19] Adel Nadjaran Toosi, Mohsen Kahani, “A New Approach to Intrusion Detection Based on an Evolutionary Soft Computing Model Using Neuro-Fuzzy Classifiers”, Computer Communications, vol.30, issue10, pp.2201-2212, 2007.

[20] Zaiqiang Liu, Dengguo Feng, “Incremental Fuzzy Decision Tree-Based Network Forensic

System", Computational Intelligence and Security, vol.3802, pp.995-1002, 2005.
[21] http://datatracker.ietf.org/doc/rfc3227/
[22] Simon Garfinkel, "Anti-Forensics : Techniques, Detection and Countermeasures", 2nd InternationalConferenceoni-WarfareandSecurity, pp.77-84
[23] Liu, Brown, "Bleeding-Edge Anti-Forensics", Infosec World Conference and Expo, MIS Training Institute
[24] http://support.microsoft.com/kb/223316
[25] http://securotyfocus.com/archive/1/348638/2003-12-29/2004-01-04/0
[26] http://www.shadowserver.org/wiki/pmwiki.php/Stats/PackerStatistics
[27] Brian Hay, Kara Nance, Matt Bishop, "Live forensic Progress and Challenges", IEEE Security and Privacy, vol.7, issue2, pp.30-37, 2009.
[28] L.A. Zadeh, "Fuzzy Sets", Information and Control, pp.338-353, 1965.
[29] http://en.wikipedia.org/wiki/Triangular_distribution

◐ 저 자 소 개 ◑



김 세 령 (Se Ryoung Kim)
2005년 서울여자대학교 컴퓨터 공학과 (학사)
2010년~현재 고려대학교 정보보호대학원 정보보호학과 석사과정
관심분야 : 침입 탐지, 전문가 시스템, 역공학
E-mail : seryoung82@korea.ac.kr

김 휘 강 (Huy Kang Kim)
1998년 KAIST 산업경영학과 (학사)
2000년 KAIST 대학원 산업공학과 (석사)
2009년 KAIST 대학원 산업 및 시스템 공학과 (박사)
2010년~현재 고려대학교 정보보호대학원 정보보호학과 조교수
관심분야 : 온라인 게임 보안, 침입 탐지, 데이터 마이닝, 전문가 시스템
E-mail : cenda@korea.ac.kr