APT 공격 탐지를 위한 호스트 기반 특징 표현 방법

Host based Feature Description Method for Detecting APT Attack

저자 (Authors)

문대성, 이한성, 김익균

Daesung Moon, Hansung Lee, Ikkyun Kim

출처 (Source)

정보보호학회논문지 24(5), 2014.10, 839-850 (12 pages)

Journal of the Korea Institute of Information Security & Cryptology 24(5), 2014.10, 839-850 (12 pages)

발행처 (Publisher)

한국정보보호학회

Korea Institute Of Information Security And Cryptology

URL

http://www.dbpia.co.kr/Article/NODE02492694

APA Style

문대성, 이한성, 김익균 (2014). APT 공격 탐지를 위한 호스트 기반 특징 표현 방법. 정보보호학회논문지, 24(5), 839-850.

Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.5, Oct. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.5.839

APT 공격 탐지를 위한 호스트 기반 특징 표현 방법*

문 대 성,† 이 한 성,‡ 김 익 균
한국전자통신연구원 네트워크보안연구실

Host based Feature Description Method for Detecting APT Attack*

Daesung Moon,† Hansung Lee,‡ Ikkyun Kim
ETRI, Network Security Research Team

요 약

3.20 사이버 테러 등 APT 공격이 사회적, 경제적으로 막대한 피해를 초래함에 따라 APT 공격을 방어하기 위한 기술적인 대책이 절실히 요구되고 있으나, 시그너쳐에 기반한 보안 장비로는 대응하는데 한계가 있다. 이에 본 논문에서는 기존 시그너쳐 기반 침입탐지 시스템의 한계를 극복하기 위해서 호스트 PC에서 발생하는 행위정보를 기반으로 악성코드를 탐지하는 방법을 제안한다. 먼저, 악성코드와 정상 실행파일을 구분하기 위한 39개의 특성인자를 정의하고, 악성코드 및 정상 실행파일이 실행되는 동안 발생하는 870만 개의 특성인자 데이터를 수집하였다. 또한, 수집된 데이터에 대해 각 특성인자의 발생빈도를 프로세스 ID 별로 재구성하여 실행파일이 호스트에서 실행되는 동안의 행위정보를 83차원의 벡터로 표현하였다. 특히, 자식 프로세스에서 발생하는 특성인자 이벤트의 발생빈도를 포함함으로써 보다 정확한 행위정보의 표현이 가능하였다. C4.5 결정트리 방법을 적용하여 악성코드와 정상파일을 분류한 결과 각각 2.0%의 오탐률과 5.8%의 미탐률을 보였다.

ABSTRACT

As the social and financial damages caused by APT attack such as 3.20 cyber terror are increased, the technical solution against APT attack is required. It is, however, difficult to protect APT attack with existing security equipments because the attack use a zero-day malware persistingly. In this paper, we propose a host based anomaly detection method to overcome the limitation of the conventional signature-based intrusion detection system. First, we defined 39 features to identify between normal and abnormal behavior, and then collected 8.7 million feature data set that are occurred during running both malware and normal executable file. Further, each process is represented as 83-dimensional vector that profiles the frequency of appearance of features. the vector also includes the frequency of features generated in the child processes of each process. Therefore, it is possible to represent the whole behavior information of the process while the process is running. In the experimental results which is applying C4.5 decision tree algorithm, we have confirmed 2.0% and 5.8% for the false positive and the false negative, respectively.

Keywords: Advanced Persistent Threat, APT, Anomaly Detection, HIDS

I. 서 론

컴퓨터, 인터넷 서비스 등 정보통신 기술이 급속히 발달함에 따라, 사용자들은 언제 어디에서나 원하는 정보에 접근하여 사용할 수 있게 되었다. 이러한 기술적인 발달은 사용자들에게 편리함을 제공할 뿐만 아니

접수일(2014년 7월 17일), 게재확정일(2014년 8월 3일)
* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 수행하였음.[13-921-06-002, 다중소스 데이터의 Long-term History 분석 기반 사이버 표적공격 인지 및 추적 기술개발]
† 주저자, daesung@etri.re.kr
‡ 교신저자, mohan@etri.re.kr(Corresponding author)

라, 기업에는 높은 업무 효율성을 제공하고 있으나 주요정보가 네트워크 환경에 연결되어 있기 때문에 악의적인 사용자에 의해 유출될 가능성이 항상 존재한다. 최는 발생한 SK 커뮤니케이션즈 해킹 사고, 농협 전산망 해킹 사고 등의 해킹 사고는 이러한 우려를 방증하고 있으며, 심각한 사회적인 문제로 대두되고 있다[1,2].

과거 해커들은 단순한 호기심 또는 자신의 능력을 과시하기 위해서 컴퓨터 바이러스, 인터넷 웜과 같은 악성코드를 제작하여 불특정 다수에게 전파한 후, 다른 사용자의 시스템을 해킹하는 악의적인 행동을 하였다. 2000년대 중후반에는 불특정 시스템들로부터 발생된 대량의 트래픽을 공격대상 홈페이지 또는 서버에 보내 해당 네트워크 및 시스템의 성능을 저하시켜 정상적인 서비스 제공이 불가능하게 하는 분산 서비스 거부 공격(Distributed Denial-of-Service, DDoS)이 출연하였다.

주로 특정 홈페이지를 공격하여 서비스를 마비시키는 것이 목적인 DDoS 공격과는 달리, 최근 명확한 목적과 대상을 상대로 시스템을 파괴하거나, 대량의 중요 정보를 유출시키는 지능형 지속 위협(Advanced Persistent Threat, APT) 공격이 등장하였으며, 원자력 발전소와 같은 중요한 산업기반시설이나 구글, 야후 같은 유명 인터넷업체, EMC RSA같은 대표적인 보안업체들이 잇달아 APT 공격에 속수무책으로 당하면서, 그 우려와 관심이 최근 무척 높아지고 있다[3-5].

방화벽(Firewall), 침임탐지시스템((Instruction Detection System, IDS), 침입차단시스템(Intrusion Prevention System, IPS)과 같은 네트워크 보안장비와, 바이러스 백신(Anti-Virus), 내부정보유출방지 기술(Data Loss Prevention, DLP) 등의 호스트 PC보안 기술들이 사이버 공격에 대응하기 위해 사용되고 있다. 이러한 기존 기술들은 대부분 알려진 공격에 대한 블랙리스트(blacklist)나 시그너쳐(signature)에 기반을 두고 있기 때문에 알려진 공격에 대해서는 효과적인 탐지 및 대응능력을 보여주고 있으나, 제로데이(Zero-day)취약점 및 신종/변종 악성코드를 지속적으로 이용하는 APT 공격을 대응하기에는 한계가 있다.

침입 탐지 시스템은 데이터가 수집되는 위치에 따라서 네트워크 기반 침입탐지 시스템(Network based IDS, NIDS)과 호스트 기반 침입탐지 시스템(Host based IDS, HIDS)으로 구분할 수 있다. 또한, 침입탐지 분석방법을 기준으로 오용 탐지(Misuse Detection)와 비정상행위 탐지(Anomaly Detection)로 나눌 수 있다[6-8].

본 논문에서는 호스트 기반 비정상행위 탐지 시스템(Host-based Anomaly Detection System)을 제안한다. 먼저, 호스트 PC에서 실행되는 프로세스가 정상 또는 비정상 인지를 구분하기 위해 7종류의 카테고리(프로세스, 쓰레드, 파일시스템, 레지스트리, 네트워크, 서비스, 기타정보 등)로 구분하여 총 39개의 특징들을 정의한 후, 호스트 PC에서 실행되는 프로세스로부터 발생되는 39개의 특징정보를 수집하여 데이터베이스를 생성하였다.

수집된 데이터베이스는 다시 프로세스별로 각 특징정보의 발생횟수를 표현하는 형태로 재구성하였다. 이때, 자식 프로세스가 존재할 경우 자식 프로세스에서 발생되는 특징정보의 발생횟수까지 함께 표현함으로써, 해당 프로세스의 시작 시점부터 종료 시점까지 발생되는 행위정보를 특징 벡터로 재구성 하였다.

마지막으로, 프로세스 별로 재구성된 프로세스 행위정보를 결정트리(Decision Tree) 방법에 입력으로 사용하여 악성코드와 정상 실행파일을 구분하는 실험을 수행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 APT 공격의 특징 및 침입탐지 시스템에 대해 설명한다. 3장에서는 본 논문에서 제안한 APT 공격을 탐지하기 위한 호스트 행위기반 특성인자의 표현(description) 방법에 대해 기술하고, 4장에서는 데이터 수집 및 실험결과에 의한 성능을 평가한다. 마지막으로 5장에서는 결론을 맺는다.

II. APT 공격

2.1 APT 공격 유형

APT 공격[9-11]에 대한 정의는 문헌들 마다 다양하게 정의하고 있으며, 미국 표준기술연구소(National Institute of Standards and Technology, NIST)에서는 APT 공격을 다음과 같이 정의하고 있다. APT 공격은 전문지식과 많은 자원을 가진 공격자가 여러 다른 공격 경로를 통해 그들의 목적을 달성하는 공격이다. 일반적으로 공격자들의 목적은 첫째,

중요정보를 지속적으로 탈취하기 위한 조직의 정보기술 인프라 내에서 기반을 마련하거나 확장하고, 둘째, 공격 대상의 임무나 계획 또는 조직 차체를 약화시키거나 방해하는 것이며, 셋째, 향후 이러한 목적을 달성하기 위해서 기반을 마련하는 것이다. 게다가, APT 공격은 방어자의 노력에 적응하면서, 목적을 실행하기 위해 필요한 수준의 상호 작용을 유지하면서, 오랜 기간 동안 반복적으로 목적 달성을 시도한다[9].

즉, 일반적으로 특정한 목표 대상에 대해 취약점을 파악하고 다양한 방법을 이용한 지속적인 공격활동으로 정보 탈취, 시스템 파괴 등의 손상을 입히려는 새로운 공격 형태를 의미한다. 위키피디아(Wikipedia)에서는 사이버공격이 지능형(Advanced), 지속성(Persistent), 위협성(Threat)이라는 특징적 요건들을 충족하면 APT 공격인 것으로 간주하고 있다[10].

APT 공격의 단계 또한 문헌마다 조금씩 다르게 정의하고 있으며, 본 논문에서는 Fig. 1과 같이 APT 공격을 사전준비, 내부망 침투, 내부활동, 목적달성의 4단계로 정의한다. 사전준비 단계는 공격자가 이후 단계를 성공하기 위해 준비하는 과정으로써, 공격대상에 관한 정보수집/분석, 웹페이지 변조, C&C 확보 등과 같은 작업이 이루어진다. 공격대상에 관한 정보수집/분석 작업은 공격대상의 홈페이지를 통해 내부 조직구조, 직원들의 연락처와 같은 다양한 정보를 수집할 수 있으며, SNS에서 수집된 자료의 분석을 통해서도 공격대상에 대한 다양한 정보(사용 SW 종류, 협력업체, 자주 접속하는 웹사이트/웹하드 등)를 획득할 수 있다. 웹페이지 변조 작업은 공격대상 정보수집/분석 작업을 통해 획득된 공격대상의 임직원들이 자주 접속하는 웹사이트/웹하드를 변조하여 악성코드를 업로드 시키는 등의 사전준비 작업이 이루어진다. 또한, 공격대상 내부에 악성코드로 감염된 좀비PC와 통신/제어하기 위한 C&C 서버 확보 등이 사전준비 단계의 작업이다. 이처럼, 사전준비 단계는 홈페이지, SNS 등 이미 공개된 자료를 이용할 뿐만 아니라, 공격대상의 외부에서 진행되기 때문에 사전준비 단계에서 APT 공격을 탐지하는 것은 불가능하다.

내부망 침투 단계는 공격자가 공격대상의 IT 인프라에 침투하는 단계로써, 악성 e-mail 발송, 변조된 웹하드/웹게시판 접속, 변조 업데이트 서버 접속 등의 방법을 통해 이루어진다. 메일 수신자의 관심분야, 사용 중인 웹하드/웹게시판, 사용 중인 SW 리스트 정보는 사전준비 단계에서 이미 수집/분석되었다. 악성 e-mail을 통한 내부망 침투의 경우 공격대상 조직의 인사 관련 내용, 사회적인 이슈와 관련된 내용, 카드 이용대금 명세서, 쇼핑몰 배송안내 등 사회공학적인 기법을 이용하여 공격대상 직원이 관심을 가지고 의심없이 메일의 첨부파일을 실행하도록 유도한다. 또한, 변조된 웹하드/웹게시판에 공격대상 직원이 접속하게 되면 악성코드가 자동으로 다운로드 되어 PC를 감염시킨다. 특히, 최근 발생한 대표적인 APT 공격 사례인 SK 컴즈 및 3.20 사이버 사고에서와 같이 목표대상에서 사용 중인 SW의 업데이트 서버를 통해 악성코드를 감염시킴으로써 각종 보안장비를 무력화 시키고 공격대상의 내부망에 침투할 수 있었다. 이처럼, 사회공학적인 기법 및 제로데이 취약점을 이용하여 침투하기 때문에 내부망 침투 단계에서 APT 공격을 탐지하는 것은 아주 어렵다.

내부활동 단계는 내부망 침투 단계에서 감염시킨 좀비PC를 거점화하여 최종 공격목적을 달성하기 위해 공격대상의 내부 IT 인프라에 대한 정보를 수집하는 단계이다. 내부활동 단계에서는 좀비PC가 C&C

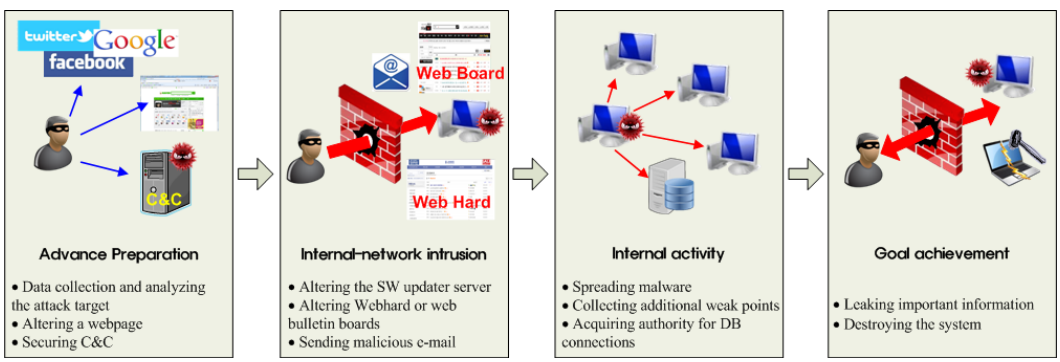


Fig. 1. Stage of APT Attack

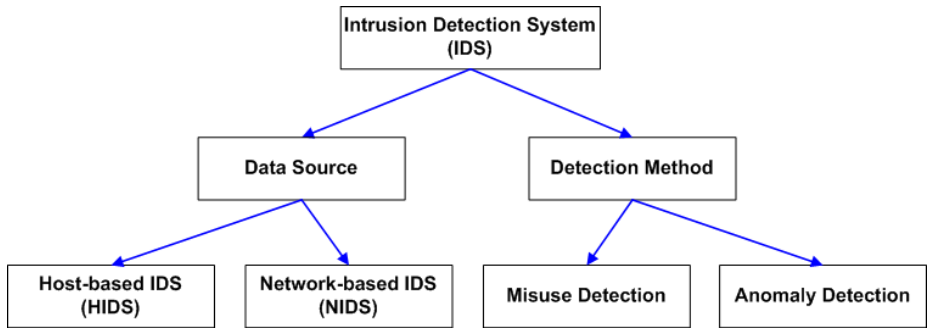


Fig. 2. Classification of Intrusion Detection System

서버와 연결되어 추가적인 악성코드를 다운로드하는 작업, 악성코드를 공격대상 조직의 다른 PC에 전파하는 작업, 추가 취약점을 수집하는 작업, 중요정보가 저장되어 있는 데이터베이스에 접속하기 위한 권한을 획득하는 등의 행위들이 다수의 악성코드에 의해 진행된다.

마지막, 목적달성 단계는 중요정보를 유출하거나, 공격대상 내부 IT 인프라를 파괴하는 등 APT 공격의 목적을 달성하기 위한 작업이 진행되며, 특정 행위를 실행하기 위한 다양한 악성코드가 사용된다. 추가적으로 공격자가 핵심정보를 지속적으로 유출시키기 위해 백도어 등의 프로그램을 설치하여 공격대상에 지속적으로 접근할 수 있도록 시도한다. SK 컴즈 사고에서는 대량의 고객정보가 유출되었으며, 3.20 사이버 테러에서와 같이 감염된 시스템의 MBR(Master Boot Record)을 손상시켜 시스템을 사용불가능하게 만들었다.

이처럼, 지능적인 APT 공격에 대응하기 위해서는 호스트, 네트워크 및 레거시(legacy) 장비 등에서 발생되는 다양한 이벤트 정보를 종합적으로 이용하여 분석해야 한다. 특히, 호스트 PC 관점에서 보면, APT 공격은 다수의 악성코드를 호스트 PC에 감염시킨 후 공격목적을 달성하기 때문에, 호스트 PC에서 다수의 악성코드 중 일부만 탐지해도 APT 공격을 막을 수 있다.

본 논문에서 제안하는 호스트기반 비정상행위 탐지 방법은 상대적으로 탐지가 불가능하거나 어려운 사전 준비 및 내부망 침투 단계가 아니라, 내부활동 및 목적달성 단계에서 APT 공격을 탐지하는 것에 초점을 둔다.

2.2 호스트 기반 비정상 행위 탐지 기존 방법

침입탐지 시스템은 데이터가 수집되는 위치에 따라서 네트워크 기반 침입탐지 시스템과 호스트 기반 침입탐지 시스템으로 구분할 수 있다. 직관적으로 알 수 있듯이 네트워크 기반 침입탐지 시스템은 네트워크 트래픽 데이터를 분석하여 침입여부를 탐지하며, 호스트 기반 침입탐지 시스템은 호스트 PC에 생성되는 실행파일의 시그너쳐를 분석하거나, 실행중인 프로세스의 행위정보를 분석하여 침입여부를 판단하게 된다.

또한, 침입탐지 분석방법을 기준으로 오용 탐지(Misuse Detection)와 비정상행위 탐지(Anomaly Detection)로 나눌 수 있다. 오용 탐지는 이미 알려진 악성코드 및 악성행위에 대해 행위 규칙(behavior rule), 시그너쳐 등의 규칙을 생성한 후, 규칙에 만족할 경우 악성코드로 분류하는 침입탐지 방법이다. 반면, 비정상행위 탐지는 악성코드가 아닌 정상행위에서 발생되는 네트워크 및 호스트 데이터를 수집하여 정상행위를 나타내는 모델을 생성한다. 생성된 정상행위 모델에 포함되지 않는 행위를 악성행위로 탐지하는 방법이며, 정상행위 모델 생성에는 데이터 마이닝, 통계적 방법 등의 방법들이 사용된다[6,7].

오용 탐지는 악성코드 및 악성행위를 기준으로 규칙을 생성하기 때문에, 알려진 악성코드에 대해 높은 탐지율과 빠른 탐지속도를 보장하지만, 제로데이 악성코드에 대한 대응이 불가능하고, 신종/변종 악성코드에 대한 실시간 갱신이 필수적이다. 비정상행위 탐지는 정상행위 및 정상 실행파일을 기준으로 탐지하기 때문에, 악성코드의 종류에 상관없이 탐지가 가능하며, 정상행위에서 벗어나는 제로데이 악성코드에 대해서도 탐지가 가능한 장점이 있는 반면, 높은 오탐률과 정상행위에 대한 데이터의 수집이 어려운 단점이 있다.

APT 공격과 같이 최근에 발생된 사이버 공격들은 제로데이 취약점 및 제로데이 악성코드를 사용하는 등 점차 지능화된 공격기법을 사용하기 때문에 오용 탐지만으로는 적절한 대응이 어렵다. 따라서, 본 논문에서는 정상행위에 포함되지 않는 악성코드 및 악성행위를 탐지하기 위한 비정상행위 탐지에 초점을 두고 있으며, 특히 호스트 기반 비정상행위 탐지에 관하여 기술한다.

기존 호스트 기반 비정상행위 탐지 연구들에서는 악성코드와 정상 실행파일의 행위를 구분하기 위해 OPCode 상에서 "open, read, close"처럼 시스템 호출 시퀀스(system call sequence)를 특징정보로 주로 이용하였으며, 수집된 특징정보를 그래프, HMM(Hidden Markov Model) 등의 데이터 마이닝 기법에 적용하는 호스트 기반 비정상행위 탐지 방법이 주로 연구되었다[6, 12-14]. Murtaza[15]는 시스템 호출 시퀀스를 사용하는 대신 특정 시간동안 발생하는 시스템 호출 시퀀스를 상태(state)라고 정의하고, File System, Kernel, Memory Management 등 8개의 상태 카테고리로 구분하였다. 특정 응용 프로그램에서 발생하는 상태 전이(state transition) 및 상태들의 상호연관성 패턴을 분석하여 정상행위와 비정상 행위를 구분하였다. S. S. Murtaza의 연구결과에서는 낮은 오탐율과 빠른 탐지시간이 가능함을 보였으나, 모든 정상 프로그램에 대해 상태정보의 패턴을 수집하기 힘든 문제점이 있다.

Kaur[16]는 호스트 기반 비정상 행위 탐지에 퍼지 로직과 유전자 알고리즘을 이용하였다. 수집된 시스템 로그 파일에 퍼지 규칙을 적용하여 정상 사용자의 행위를 정의하였고, 퍼지 규칙으로 표현된 사용자 행위를 보다 최적화하기 위해서 유전자 알고리즘을 적용하여 새로운 사용자 행위 규칙을 생성하였다.

Igor Santo[17]는 정상파일과 악성코드에서 발생되는 OPCode의 발생빈도와 OPCode 시퀀스 정보를 활용하였다. 실행파일에서 추출된 OPCode의 발생빈도로부터 생성된 OPCode의 가중치를 활용하여 의미 없는 OPCode를 잡음으로 제거하였다. 또한, 각각의 실행파일에 대해서 OPCode 시퀀스와 해당 시퀀스의 빈도수를 벡터형태로 표현하였기 때문에, 코사인 유사도를 이용하여 탐지대상 실행파일이 악성파일 또는 정상파일과 유사한지 특정하였다. 활용하여 두 개 파일의 유사도를 코사인 유사도를 화하고, 벡터들 간의 코사인 유사도를 도출하였다.

이와 같은 기존연구들은 실행파일에 역공학(reverse engineering) 기법을 적용하여 생성된 PE 파일에서 악성코드의 시스템 호출 패턴을 분석하는 정적분석이 주를 이루었다. 그러나, 최근 정적분석을 어렵게 하는 난독화 기법이 악성코드에 적용되어 악성코드 탐지를 어렵게 하고 있다.

III. 호스트 행위 기반 APT 탐지

APT 공격은 기존의 사이버 공격과 마찬가지로 다수의 악성코드를 호스트 PC에서 실행시켜 공격목적을 달성하기 때문에, 호스트에서 APT 공격을 방어하기 위해서는 악성코드의 탐지가 기본이다. 본 연구에서는 악성코드를 탐지하는 다양한 방법 중, 악성코드 시그너쳐를 이용하지 않고 악성코드가 호스트 PC에서 실행되는 동안 발생되는 다양한 행위 이벤트정보를 이용한 악성코드 탐지방법을 제안하고자 한다.

3.1 호스트 기반 특성인자 정의

호스트 행위 기반 APT 탐지를 위한 첫 번째 단계로, 악성코드와 정상 실행파일의 구분을 위한 행위 이벤트로부터 특징정보를 정의했으며 특성인자라 명명한다. 본 연구에서 정의한 특성인자는 Table 1에 정의하였으며, Windows 프로그램들이 Windows API를 호출할 때 호스트에서 발생하는 다양한 이벤트 중에서 프로세스, 쓰레드, 파일시스템, 레지스트리, 네트워크, 서비스, 기타정보 등 7종류의 카테고리로 구분하여 총 39개의 특징들을 정의하였다. 39개의 특성인자들은 특정 악성코드의 행위에 의존적인 내용이 아니며, 정상 실행파일에서도 빈번하게 발생하는 특성인자들로 구성되어 있다.

프로세스 카테고리는 프로그램의 프로세스 생성 및 삭제, 프로세스 메모리 접근 등과 관련된 행위를 정의하는 7개의 특성인자로 구성되며, 쓰레드 카테고리는 프로그램의 쓰레드 생성 및 삭제, 쓰레드 제어와 관련된 행위를 정의하는 7개의 특정인자로 구성된다. 파일 카테고리는 프로그램의 파일 생성 및 삭제, 복사, 검색, 입출력 등과 관련된 8개의 특성인자로 정의하였으며, 레지스트리 카테고리는 프로그램의 레지스트리 생성 및 삭제, 읽기, 쓰기 등 5개의 특정인자로 정의하였다. 네트워크 및 서비스 카테고리는 호스트 컴퓨터에서 동작하는 해당 프로그램이 네트워크 및 인터넷

Table 1. Definition of Characteristic Parameters for Host based Anomaly Detection

Category	Index	Characteristic Parameter	Category	Index	Characteristic Parameter
Process	1	CreateProcess	File	21	WriteFile
	2	ExitProcess		22	SearchFile
	3	TerminateProcess	Registry	23	CreateRegistry
	4	OpenProcess		24	DeleteRegistry
	5	ReadProcessMemory		25	OpenRegistry
	6	WriteProcessMemory		26	ReadRegistry
	7	SearchProcess		27	WriteRegistry
Thread	8	CreateLocalThread	Network	28	Connect
	9	CreateRemoteThread		29	Listen
	10	ExitThread		30	Send
	11	TerminateThread		31	Recv
	12	OpenThread		32	Download
	13	SuspendThead	Service	33	CreateService
	14	ResumeThead		34	DeleteService
File	15	CreateFile		35	OpenService
	16	CopyFile		36	StartService
	17	MoveFile	Misc	37	CreateMutex
	18	DeleteFile		38	OpenMutex
	19	OpenFile		39	LoadLibrary
	20	ReadFile			

Table 2. Information of Collected Characteristic Parameters

Collected information	Explanation
index	Index of characteristic parameters event-occurrence
HostID	Host-PC ID
UserID	User ID
Time	Time of characteristic parameters event-occurrence
Filename	Executable filename
PID	Process ID
TID	Thread ID
Feature Index	Index of characteristic-parameter
Result	Return value of the API
Parameter1	Parameter information for the API

서비스를 사용할 때 발생할 수 있는 행위들로 각각 5개와 4개의 특정인자로 구성된다. 마지막으로, 기타 카테고리는 특정 분류로 정의하기 어려운 CreateMutex, OpenMutex, LoadLibaray의 3개의 특성인자로 정의하였다. 각각의 특성인자는 동일한 행위 이벤트라도 서로 다른 여러 가지 Windows API가 호출될 때 발생할 수 있기 때문에, 호스트에서 발생하는 모든 특성인자를 수집하기 위해서는 39개보

Table 3. Data Structure for Feature Description Vector

Index	Data type	Parameter name	Explanation	Characteristic-parameter index
1	char *	cUserID	User ID	-
2	int	iHostID	Host ID	-
3	char *	cFilename	Executable filename	-
4	int	iPid	Process ID	-
5	int	iCreateProcess	Number of CreateProcess event	1
6	int	iCreateProcess_CP	Number of CreateProcess event of child process	1
7	int	iExitProcess	Number of ExitProcess event	2
8	int	iExitProcess_CP	Number of ExitProcess event of child process'	2
. . .	. . .	. . .	. . .	. . .
79	int	iOpenMutex	Number of OpenMutex event	38
80	int	iOpenMutex_CP	Number of OpenMutex event of child process'	38
81	int	iLoadLibrary	Number of LoadLibrary event	39
82	int	iLoadLibrary_CP	Number of LoadLibrary event of child process'	39
83	int	type	Process attribute (normal/abnormal)	-

다 많은 Windows API를 모니터링하여야 한다. 예를 들어, SearchProcess 특성인자의 경우에는 FindWindowA(), FindWindowW(), FindWindowExA(), FindWindowExW() 등 다양한 Windows API 호출에 의해 해당 이벤트가 발생된다.

악성코드와 정상 실행파일이 호스트 PC에서 실행되는 동안 특성인자에 해당하는 이벤트가 발생하면 관련 데이터를 수집하게 되며, 하나의 이벤트가 발행하였을 때 Table 2와 같은 구조로 수집된다. 해당 프로그램이 동작하는 Host ID, 실행중인 프로그램에 해당하는 User ID, 프로세스 ID, 이벤트 발생시간, 실행파일 이름, Table 1에서 정의된 특성인자(Feature Index) 및 API 호출에서 발생하는 파라미터 등 다양한 정보가 함께 수집된다.

3.2 호스트 기반 특성인자 표현 방법

3.1절에서 정의된 각각의 특성인자 데이터들은 정상파일에서도 발생하는 이벤트들이기 때문에 하나의 이벤트로부터 수집된 특성인자로는 악성코드에 의해 발생한 비정상 행위 중 하나인지 또는 정상파일에 의해 발생한 행위인지를 구분할 수 없다. 예를 들어, Table 1의 첫 번째 특성인자 이벤트에서처럼 실행파일이 다른 프로세스를 생성(Feature Index :1, CreateProcess)하는 행위로는 악성행위인지 비정상행위인지를 판단할 수 없다. 따라서, 수집된 특성인자 이벤트 정보를 비정상행위 (악성행위) 또는 정상행위로 구분할 수 있는 형태로 재구성하여 사용하여야 한다. 수집된 특성인자 이벤트 정보들을 재구성하기 위한 방법에는 특성인자의 발생 순서를 패턴화 하는 방법[17]과 특정 특성인자의 발생 빈도[13]를 이용하는 방법 등 다양한 재구성 방법이 사용 가능하다. 본 연구에서는 악성코드와 정상파일에 의해 발생된 행위를 표현하기 위하여 수집된 특성인자 이벤트 정보를 프로세스 ID별로 재구성하였다. 즉, 프로그램이 실행되는 동안 발생되는 모든 행위의 패턴을 나타내기 위해서 동일한 특성인자 이벤트가 발생한 빈도수를 누적하여 프로세스 ID 별로 재구성하였다. 일반적으로 하나의 실행 프로그램(정상 실행파일 또는 악성 코드)은 여러 개의 자식 프로세스를 생성할 수 있다. 특히, 악성 코드의 경우는 하나의 악성 코드가 여러 가지 악성행위를 수행하며, 각각의 악성행위를 담당하는 자식 프로세스를 생성하는 경우가 발생한다. 따라서, 특정 프로그램의 실행 중 행위를 보다 정확히 표현하기 위해서

Table 4. Example of Collected Characteristic Parameters Data

index	HostID	UserID	Time	Filename	PID	TID	Feature Index	Result	Parameter1
30937	1	admin	2014-01-22 14:57:39,219	firefox.exe	3796	2068	8 (ThreadCreate)	SUCCESS	Thread ID: 3848
31031	1	admin	2014-01-22 14:57:39,288	firefox.exe	3796	3848	39 (LoadImage)	SUCCESS	ImagePath: C:\...\firefox.exe
...	...	...	...	...	...	...	...	...	...
288132	1	admin	2014-01-22 14:59:44,448	FlashPlayerUpdateService.exe	720	1440	2 (ProcessExit)	SUCCESS	Exit Status: 0
288133	1	admin	2014-01-22 14:59:44,450	install_flash_player.exe	2588	2396	19 (OpenFile)	SUCCESS	TargetFile: C:\...\FlashPlayerUpdateService.exe
...	...	...	...	...	...	...	...	...	...
290857	1	admin	2014-01-22 14:59:46,243	install_flashplayer12x32_mssd_aaa_aih.exe	4052	0	30 (TCPSend)	SUCCESS	root-9b86572035:3789 -> 192.168.6.1:2042
290858	1	admin	2014-01-22 14:59:46,244	cmd.exe	3112	1388	39 (LoadImage)	SUCCESS	ImagePath: C:\...\ntdll.dll
...	...	...	...	...	...	...	...	...	...
348427	1	admin	2014-01-22 15:00:59,380	plugin-container.exe	2040	0	30 (TCPSend)	SUCCESS	root-9b86572035:3974 -> 192.168.6.1:2042
348428	1	admin	2014-01-22 15:00:59,381	firefox.exe	4024	0	30 (TCPSend)	SUCCESS	root-9b86572035:3948 -> 192.168.6.1:2042
...	...	...	...	...	...	...	...	...	...
351048	1	admin	2014-01-22 15:01:00,526	plugin-container.exe	2040	3788	3 (ProcessTerminate)	SUCCESS	Exit Status: 2
351049	1	admin	2014-01-22 15:01:00,557	firefox.exe	4024	3832	2 (ProcessExit)	SUCCESS	Exit Status: 0

는, 해당 프로세스 ID에 대한 특성인자 이벤트의 재구성 뿐만 아니라 자식 프로세스들의 특성인자 이벤트에 대한 재구성이 반드시 요구된다.

본 연구에서는 해당 프로세스와 자식프로세스들에서 발생한 특성인자 이벤트의 빈도수를 이용하여, 모니터링하고자 하는 프로세스에 대한 보다 구체적인 행위를 표현하도록 특징 값을 정의하였다. 하나의 실행 프로세스는 여러 개의 자식 프로세스를 표현하고 있기에, 해당 프로세스에 의해 발생한 각 특성인자 이벤트의 빈도수와 자식 프로세스들에 의해 발생한 각 특성인자 이벤트 빈도수의 누적 합을 이용하여 각 실행 프로세스의 행위를 특징 값으로 기술하였다. Table 3에 본 연구에서 제안하는 프로세스의 행위를 기술하기 위한 특징 값을 제시하였다. 제안하는 특징 값은 Table 2에서 기술한 UserID, HostID, Filename, ProcessID의 4차원과 모니터링 하고 있는 프로세스의 특성인자 이벤트의 빈도수 39차원, 자식 프로세스들의 특성인자 이벤트 빈도수의 누적 합 39차원, 마지막으로 테스트를 위한 라벨(정상 또는 악성) 1차원을 포함 총 83차원으로 정의된다. 각 특징 값의 속성 변수명은 〈자료형 type+특성인자 명+(_CP)〉로 구성하였다. 접미어 _CP는 자식 프로세스들의 특성인자 값을 의미한다. 예를 들어, iCreateProcess는 해당 프로세스(iPid)가 타 프로세스를 생성한 빈도수이며, iCreateProcess_CP는 해당 프로세스의 자식 프로세스들이 타 프로세스를 생성한 빈도수의 누적 합을

Table 5. Example of Feature Description Vectors for Each Process

cUserID	iHostID	cFilename	iPid	iCreateProcess	iCreateProcess_CP	iExitProcess	iExitProcess_CP	…	iOpenMutex	iOpenMutex_CP	iLoadLibrary	iLoadLibrary_CP	type
admin	1	gtbcheck.exe	2052	0	0	1	0	…	0	0	18	0	Normal
admin	1	FlashPlayerUpdateService.exe	720	0	0	1	0	…	0	0	23	0	Normal
admin	1	FlashPlayerUpdateService.exe	288	0	0	1	0	…	0	0	36	0	Normal
admin	1	cmd.exe	3112	0	0	1	0	…	0	0	29	0	Normal
admin	1	install_flash_player.exe	2588	3	0	1	3	…	0	0	63	88	Normal
admin	1	cmd.exe	3492	0	0	0	0	…	0	0	17	0	Normal
admin	1	install_flashplayer12x32_mssd_aaa_aih.exe	4052	6	4	1	8	…	15	26	93	367	Normal
admin	1	firefox.exe	3796	1	11	1	10	…	9	43	88	512	Normal
admin	1	plugin-container.exe	2040	0	0	0	0	…	1	0	66	0	Normal
admin	1	firefox.exe	4024	1	0	1	0	…	8	1	74	66	Normal

Table 6. Experimental Results using C4.5 on Collected Dataset

	Number	# of TP	# of FP	TP Rate	FP Rate	Precision	Recall
Malware	3133	3070	63	98.0 %	5.5 %	98.1 %	98.0 %
Normal code	1049	991	58	94.5 %	2.0 %	94.5 %	94.5 %
Total (weighted average)	4182	4061	121	97.1 %	4.6 %	97.1 %	97.1 %

의미한다.

IV. 실험결과

4.1 데이터 수집 및 프로세스 프로파일링

호스트 PC에서 발생하는 행위정보를 기반으로 악성코드를 탐지하기 위해서는 악성코드와 정상파일의 특성인자에 대한 데이터 수집이 가장 중요한 부분 중 하나이다. 본 논문에서는, Table 1과 같이 정의된 특성인자를 수집하기 위해서 가상머신을 설치한 후 3133개의 악성코드와 1049개의 정상파일을 실행하여 특성인자 이벤트가 발생할 때마다 수집기를 통하여 수집하였다. 특성인자 수집기는 쿠쿠샌드박스[18]와 프로세스 모니터[19]를 이용하였으며, 수집기를 통해 수집된 특성인자 이벤트 정보는 Table 4와 같다.

악성코드의 경우 malshare[20]에서 수집된 11,000개의 악성코드 중에서 Adware로 분류되는 악성코드를 제외한 나머지 2663개와 자체적으로 수집한 470개의 악성코드를 한 번에 하나씩 실행하면서 발생되는 특성인자 이벤트 정보를 수집하였다. 정상파일의 경우 메모장(Notepad) 등 윈도우즈에서 제공하는 기본 프로그램 뿐 만 아니라, 일반 사용자

가 주로 사용하는 웹 브라우즈, 워드프로세스, 파일 전송 프로그램, 압축 유틸리티 등의 프로그램을 실행하면서 수집하였다. 3133개의 악성코드를 실행하는 동안 대략 400만 건의 특성인자 이벤트가 수집되었으며, 정상파일을 실행하는 동안 470만 건의 특성인자 이벤트가 수집되었다.

Table 2에서 정의된 구조로 수집된 특성인자 데이터의 일부분을 Table 4에 제시하였다. firefox.exe, FlashPlayerUpdateService.exe 등의 실행파일에서 8번, 39번, 2번등의 특성인자 이벤트가 발생하여 수집된 결과이며, parameter1 항목에 해당 API가 호출될 때 사용한 다양한 부가정보가 저장되고 있음을 보여준다.

Table 5는 Table 4의 특성인자 데이터가 프로세스 별로 재구성된 최종 결과를 보여준다. Table 5에서 83번째 필드의 type은 해당 프로세스가 정상파일인지 악성파일인지를 나타내는 것으로써, 분석 알고리즘 개발에서 교사학습(Supervised Learning)을 이용할 경우 활용된다. 수집된 특성인자를 Table 5와 같이 표현함으로써 해당 프로세스와 자식 프로세스에서 발생된 이벤트의 패턴을 분석 할 수 있는 형태로 변환하였으며, 적절한 전처리 과정을 거쳐 기계학습, 패턴인식, 데이터 마이닝 등의 다양한 분석 알고리즘의 입력으로 사용할 수 있다.

4.2 결정트리를 이용한 악성코드 분류

Table 4과 같이 수집된 특성인자 이벤트 정보를 프로세스 ID 별로 재구성 한 결과(Table 5 참조)는 다양한 데이터 마이닝 알고리즘에 적용하여 대상 프로세스의 행위 패턴이 악성코드에 의한 비정상행위 인지를 판단하게 된다. 본 논문에서는 결정트리 알고리즘을 통해 4182개의 프로세스 행위정보를 악성코드와 정상파일로 분류하는 실험을 수행하였다. 데이터 마이닝 도구인 WEKA[21]를 이용하였으며, 결정트리 방법 중에서 가장 널리 알려진 C4.5 알고리즘을 적용하였다. 또한, 수집된 특성인자 데이터를 학습 데이터와 테스트 데이터로 나뉘지 않고 10-pold 방식을 통해 테스트를 수행하였다.

Table 6은 C4.5 알고리즘에 의해 수행된 악성코드 분류결과를 보여준다. Table 6에서 보는 바와 같이 수집된 데이터베이스에 C4.5 결정트리 방법을 적용한 실험에서 2.0%의 오탐률(False Positive)과 5.8%의 미탐률(False Negative)을 보였다. 이는, 본 논문에서 정의한 39개의 특성인자 정보가 악성코드와 정상파일의 행위정보를 적절히 구분할 수 있음을 의미한다. 뿐만 아니라, 해당 프로세스에 의해 발생한 각 특성인자 이벤트의 빈도수와 자식 프로세스들에 의해 발생한 각 특성인자 이벤트 빈도수의 누적 합을 프로세스 별로 재구성한 표현방법은 결정트리와 같은 데이터 마이닝 기법에 적절함을 알 수 있다.

V. 결 론

최근 APT 공격이 지속적으로 발생하여 사회적인 이슈가 되고 있으며, 이에 대한 기술적인 방어 대책이 요구되고 있다. 그러나, 시그너쳐 기반 탐지 방법이 주를 이루는 기존 보안장비로는 제로데이 악성코드를 이용하는 등 지능적인 APT 공격에 대한 대응이 어려운 문제점이 있다. 본 논문에서는 악성코드가 호스트 PC에서 동작하는 동안 발생하는 행위정보를 이용하여 악성코드를 탐지하는 비정상행위 기반 탐지 방법을 제안하였다. 먼저, 호스트 PC에서 발생하는 다양한 행위정보 중에서 악성코드와 정상파일을 구분하기 위한 특성인자를 정의하고 각 프로세스에서 발생한 행위정보를 특성인자 발생빈도로 표현하였다. 특히, 특정 프로세스의 자식 프로세스에서 발생하는 특성인자 정보를 포함하여 특정 프로세스에 의해서 발생되는 모든 행위정보를 표현할 수 있었다. 본 논문에서 정의한 39개의 특성인자를 수집하기 위해 실제로 가상머신 환경에서 4000여 개의 악성 및 정상파일에 대해 870만개의 특성인자 데이터를 수집한 후, 프로세스 별로 특성인자의 발생빈도로 표현하였다. 발생빈도로 표현된 4000여 개의 데이터에 C4.5 결정트리 알고리즘을 적용하여 악성코드와 정상파일을 분류하는 실험에서 2.0%의 오탐률과 5.8%의 미탐률을 보였다. 보다 중요한 것은 SVM(Support Vector Machine), 신경망 등의 분류(classification) 알고리즘 또는 군집화(clustering) 알고리즘에 적용하기 위해서 수집된 870만개의 특성인자 이벤트 데이터를 다양한 형태로 가공하여 사용할 수 있다는 것이다.

References

[1] NSHC, "3.20 South Korea Cyber Attack, Red Alert Research Report," http://training.nshc.net/KOR/Document/virus/20130321_320CyberTerrorIncidentResponseReportbyRedAlert(EN).pdf, 2013

[2] Command Five. "SK Hack by an Advanced Persistent Threat," http://www.commandfive.com/papers/C5_APT_SKHack.pdf

[3] C. Tankard, "Persistent threats and how to monitor and deter them," *Network security*, Vol. 2011, No. 8, pp. 16-19, Aug. 2011.

[4] Symantec, "Symantec Internet Security Threat Report," *Symantec,* Vol. 17, Apr. 2011.

[5] A. W. Coviello. Open letter to RSA customers. www.rsa.com/node.aspx?id=3872, 2011.

[6] Jiankun Hu, "Host-Based Anomaly Intrusion Detection," *Handbook of Information and Communication Security*, Springer, pp 235-255, 2010.

[7] A. S. Ashoor and S. Gore, "Intrusion Detection System: Case study," *Proc. of International Conference on Advanced Materials Engineering*, vol. 15, Singapore, pp. 6-9, Oct. 2011.

[8] Kyungho Son, Taijin Lee, Dongho Won, "Design for Zombie PCs and APT Attack Detection based on traffic analysis," *Journal of The Korea Institute of Information Security & Cryptology*, Vol.24, No.3, pp. 491-498, Jun. 2014

[9] NIST, Special Publication 800-30 Revision 1, "Guide for Conducting Risk Assessments," http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

[10] "Advanced Persistent Threat", Wikipedia, http://en.wikipedia.org/wiki/Advanced_persistent_threat

[11] Verizon, "Threats on the horizon - the rise of the advanced persistent threat."

[12] G. Tandon, "Machine Learning for Host-based Anomaly Detection," Florida Institue of Technology, Melbourne, Florida, USA, Ph.D. thesis, 2008.

[13] W. Wang, X. H. Guan, and X. L. Zhang, "Modeling program behaviors by hidden Markov models for intrusion detection," *Proc. of International Conference on Machine Learning and Cybernetics*, pp. 2830-2835, Aug. 2004.

[14] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models," *Proc. of IEEE Symposium on Security and Privacy*, Oakland, USA, pp. 133-145, May. 1999.

[15] S. S. Murtaza, et al., Mario Couture, "A host-based anomaly detection approach by representing system calls as states of kernel modules," *Proc. of 24th Intl. Symposium on Software Reliability Engineering(ISSRE)*, pp. 431-440, Nov. 2013.

[16] H. Kaur and N. Gill. "Host based Anomaly Detection using Fuzzy Genetic Approach (FGA)," *International Journal of Computer Applications*, Vol. 74, No. 20, pp.5-9, Jul. 2013.

[17] I. Santos, et al., "Idea: Opcode-sequence-based malware detection," *Proc. of the 2nd International Symposium on Engineering Secure Software and Systems (ESSoS 2010), Lecture Notes in Computer Science*, Vol. 5965, pp. 35-43, Feb. 2010.

[18] Cuckoo sandbox, www.cuckoosandbox.org

[19] Process monitor, http://technet.microsoft.com/ko-kr/sysinternals/bb896645

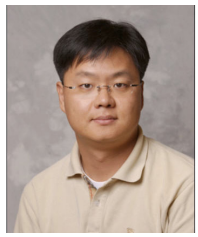
[20] Malshare, http://malshare.com/

[21] WEKA Open Sources tools for Data Mining, http://www.cs.waikato.ac.nz/ml/weka/

〈저 자 소 개〉

문 대 성 (Dae-Sung Moon) 정회원
1999년 2월: 인제대학교 전산학과 졸업
2001년 2월: 부산대학교 컴퓨터공학과 석사
2007년 2월: 고려대학교 전산학과 박사
2000년 12월~현재: 한국전자통신연구원 네트워크보안연구실 선임연구원
관심분야: 네트워크 보안, 데이터마이닝, 영상처리, 지능형비디오감시, 바이오인식

이 한 성 (Hansung LEE) 정회원
1996년 2월: 고려대학교 전산학과 학사
1996년 8월~1999년 7월: (주) 대우엔지니어링 근무
2002년 2월: 고려대학교 전산학과 석사
2008년 2월: 고려대학교 전산학과 박사
2009년 11월~현재: 한국전자통신연구원 선임연구원
관심분야: 패턴인식, 기계학습, 컴퓨터 비전, 데이터마이닝, 빅데이터



김 익 균 (Kim, Ikkyun) 정회원
1994년 2월: 경북대학교 컴퓨터공학과 졸업(공학사)
1996년 2월: 경북대학교 컴퓨터공학과졸업(공학석사)
2009년 2월: 경북대학교 컴퓨터공학과 졸업(공학박사)
2004년~2005년: Purdue University 초빙 연구원.
1996년~현재: 한국전자통신연구원 네트워크보안연구실 실장/책임연구원.
관심분야: 네트워크 보안, 컴퓨터 네트워크, 클라우드보안, 빅데이터 분석