(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년06월20일
(11) 등록번호 10-1158464
(24) 등록일자 2012년06월14일

(51) 국제특허분류(Int. Cl.)
G06F 11/30 (2006.01) *G06F 21/20* (2006.01)
(21) 출원번호 10-2010-0118617
(22) 출원일자 2010년11월26일
심사청구일자 2010년11월26일
(65) 공개번호 10-2012-0057059
(43) 공개일자 2012년06월05일
(56) 선행기술조사문헌
KR1020040082633 A
"블랙리스트 접근 트래픽 감시를 통한 봇 탐지 방법", KNOM Review, Vol.13, No. 1, pp.22-34 (2010. 7.)

(73) 특허권자
고려대학교 산학협력단

(72) 발명자
이희조
권종훈
이제현

(74) 대리인
특허법인엠에이피에스

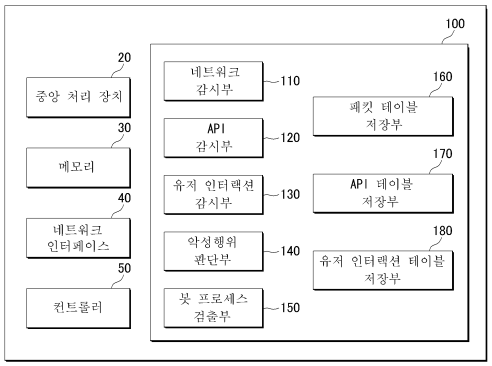
전체 청구항 수 : 총 13 항 심사관 : 이정은

(54) 발명의 명칭 **봇 프로세스 탐지 장치 및 방법**

(57) 요 약

본 발명은 사용자 단말에서 봇(Bot) 실행을 관리하는 프로세스의 탐지 방법에 관한 것으로, (a) 상기 사용자 단말에서의 패킷 발생 여부, API의 호출 여부 및 유저 인터랙션의 발생 여부를 모니터링하는 단계, (b) 상기 모니터링 결과에 따라 패킷 발생 정보, API의 호출 정보 및 유저 인터랙션 정보를 각각 패킷 테이블, API 테이블 및 유저 인터랙션 테이블에 저장하는 단계, (c) 상기 사용자 단말에서 패킷이 발생한 경우, 상기 유저 인터랙션의 발생에 따라 상기 패킷이 발생하였는지 여부에 기초하여 상기 발생한 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 단계 및 (d) 상기 봇에 의하여 발생한 것으로 분류된 패킷과 상기 API 테이블에 각 API의 호출 정보에 매칭되어 저장된 프로세스들 간의 상관도에 기초하여, 상기 봇에 의하여 발생한 패킷으로 분류된 패킷을 관리하는 프로세스를 봇 프로세스로서 검출하는 단계를 포함한다.

대 표 도 - 도1



10

이 발명을 지원한 국가연구개발사업

과제고유번호	WR080951M0211612
부처명	(재)서울특별시시정개발연구원
연구사업명	세계유수연구소유치지원사업
연구과제명	2차년도 [1-B] Blended Services Applications
주관기관	고려대학교 산학협력단
연구기간	2009.12.01 ~ 2010.11.30

특허청구의 범위

청구항 1

사용자 단말에서 봇(Bot) 실행을 관리하는 프로세스의 탐지 방법에 있어서,

(a) 상기 사용자 단말에서의 패킷 발생 여부, API의 호출 여부 및 유저 인터랙션의 발생 여부를 모니터링하는 단계,

(b) 상기 모니터링 단계의 수행 중에 발생된 패킷의 속성 정보, 상기 API를 호출한 프로세스에 대한 정보 및 상기 유저 인터랙션을 발생시킨 프로세스에 대한 정보를 각각 패킷 테이블, API 테이블 및 유저 인터랙션 테이블에 저장하는 단계,

(c) 상기 사용자 단말에서 패킷이 발생한 경우, 상기 유저 인터랙션과 무관하게 상기 패킷이 발생하였다면, 상기 발생한 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 단계 및

(d) 상기 봇에 의하여 발생한 것으로 분류된 패킷과 상기 API 테이블에 저장된 프로세스들 간의 상관도가 임계값 이상인 경우, 해당 패킷을 관리하는 프로세스를 봇 프로세스로서 검출하는 단계를 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 2

제 1 항에 있어서,

상기 (b) 단계는,

상기 패킷이 발생한 경우 발생된 패킷을 관리하는 프로세스 정보 및 패킷의 발생 시간 정보를 상기 패킷 테이블에 추가적으로 저장하는 것인 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 3

제 1 항에 있어서,

상기 (b) 단계는,

상기 API의 호출이 발생한 경우, 상기 API 호출의 발생 시간 정보를 상기 API 테이블에 추가적으로 저장하는 것인 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 4

제 1 항에 있어서,

상기 (b) 단계는,

상기 유저 인터랙션이 발생한 경우, 상기 유저 인터랙션의 발생과 관련된 프로세스 별로 유저 인터랙션의 발생 시간 정보를 상기 유저 인터랙션 테이블에 추가적으로 저장하는 것인 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 5

제 1 항에 있어서,

상기 (c) 단계는,

상기 패킷의 발생 시간과 가장 최근의 유저 인터랙션 발생 시간과의 차이가 임계값 이하인 경우 상기 패킷을

정상 패킷으로 분류하는 단계 및

상기 패킷의 발생 시간과 상기 가장 최근의 유저 인터랙션 발생 시간과의 차이가 임계값을 초과하는 경우 상기 발생한 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 단계

를 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 6

제 1 항에 있어서,

상기 패킷이 봇에 의하여 발생한 패킷으로 분류된 경우, 상기 패킷의 속성 정보에 포함된 소스 어드레스 정보가 상기 사용자 단말의 네트워크 어드레스 정보와 상이한 경우, 상기 패킷을 스푸핑(spoofing) 행위와 관련된 패킷으로 분류하는 단계를 더 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 7

제 1 항에 있어서,

상기 패킷이 봇에 의하여 발생한 패킷으로 분류된 경우, 송신 패킷의 양에 대한 수신 패킷의 양의 비율이 임계값 보다 작은 경우 DDoS 트래픽 발생행위와 관련된 패킷으로 분류하는 단계를 더 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 8

제 1 항에 있어서,

상기 패킷이 봇에 의하여 발생한 패킷으로 분류된 경우, 메일 전송 프로토콜을 이용한 단위 시간당 메일 발송 횟수가 임계값을 초과하면 스팸메일 발송행위와 관련된 패킷으로 분류하는 단계를 더 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 9

삭제

청구항 10

제 1 항에 있어서,

상기 (d) 단계에서 검출된 봇 프로세스의 실행을 차단하는 단계를 더 포함하는 봇 실행을 관리하는 프로세스의 탐지 방법.

청구항 11

삭제

청구항 12

삭제

청구항 13

청구항 1 내지 청구항 8 및 청구항 10 중 어느 한 항의 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터 판독 가능한 기록 매체.

청구항 14

사용자 단말에서 봇(Bot) 실행을 관리하는 프로세스의 탐지 장치에 있어서,

상기 사용자 단말에서의 패킷 발생 여부를 모니터링하는 네트워크 감시부,

상기 사용자 단말에서의 API의 호출 여부를 모니터링하는 API 감시부,

상기 사용자 단말에서의 유저 인터랙션 정보의 발생 여부를 모니터링하는 유저 인터랙션 감시부,

상기 패킷이 발생한 경우, 발생된 패킷의 속성 정보, 패킷을 관리하는 프로세스 및 패킷의 발생 시간에 대한 정보가 저장되는 패킷 테이블,

상기 API의 호출이 발생한 경우, 상기 API 호출을 발생시킨 프로세스 및 API 호출의 발생 시간에 대한 정보가 저장되는 API 테이블,

상기 유저 인터랙션이 발생한 경우, 상기 유저 인터랙션의 발생과 관련된 프로세스 별로 유저 인터랙션의 발생 시간 정보가 저장되는 유저 인터랙션 테이블,

상기 패킷의 발생시간과 가장 최근의 유저 인터랙션 발생 시간의 차이가 임계값보다 큰 경우 상기 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 패킷 분류부 및

상기 봇에 의하여 발생한 패킷으로 분류된 패킷과 상기 API 테이블에 저장된 프로세스들 간의 상관도가 임계값 이상인 경우, 해당 패킷을 관리하는 프로세스를 봇 프로세스로서 검출하는 봇 프로세스 검출부

를 포함하는 봇 실행을 관리하는 프로세스의 탐지 장치.

청구항 15

제 14 항에 있어서,

상기 봇 프로세스 검출부는 상기 검출된 봇 프로세스의 실행을 차단시키는 봇 실행을 관리하는 프로세스의 탐지 장치.

청구항 16

제 14 항에 있어서,

상기 패킷 분류부는,

상기 패킷의 속성 정보에 포함된 소스 어드레스 정보가 상기 사용자 단말의 네트워크 어드레스 정보와 상이한 경우, 상기 패킷을 스푸핑(spoofing) 행위와 관련된 패킷으로 분류하고,

송신 패킷의 양에 대한 수신 패킷의 양의 비율이 임계값 보다 작은 경우 DDoS 트래픽 발생행위와 관련된 패킷으로 분류하고,

메일 전송 프로토콜(SMTP)을 이용한 단위 시간당 메일 발송 횟수가 임계값을 초과하면 스팸메일 발송행위와 관련된 패킷으로 분류하는 것인 봇 실행을 관리하는 프로세스의 탐지 장치.

명 세 서

기 술 분 야

[0001] 본 발명은 봇 프로세스를 탐지하는 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 최근 들어, 봇넷(botnet)에 의하여 발생되는 여러 종류의 공격행위는 인터넷 시스템의 안정성을 위협하는 중대한 문제가 되고 있다. 봇넷이란 스팸메일이나 악성코드 등을 전파하도록 하는 악성코드 봇(Bot)에 감염되

어 해커가 마음대로 제어할 수 있는 좀비 PC들로 구성된 네트워크를 말한다. 일단 봇에 감염되면 실제 PC 사용자들은 자신의 컴퓨터가 감염된 줄 모르는 경우가 많고, 해커는 수십에서 수만 대의 시스템에 명령을 전달해 특정 인터넷 사이트에 대량의 접속 신호를 보내 해당 사이트를 다운시키는 등의 방식으로 대규모 네트워크 공격을 수행할 수 있다.

[0003] 봇(Bot)은 컴퓨터에 감염되는 악성코드의 한 종류로, 컴퓨터 사용자의 의견과 상관없이 봇 마스터(Bot master)라 불리는 공격자의 명령을 받아 악성행위를 수행한다. 이러한 봇은 분산 서비스 거부 (Distribute Denial of Service, DDoS) 공격, 스팸 메일, 부정 클릭(Click fraud), 개인 정보 탈취 등 다양한 기능을 탑재하고 명령을 통해 제어 가능한 면에서 높은 확장성을 가졌다.

[0004] 봇 탐지 관련 기술은 크게 네트워크 기반 탐지와 호스트 기반 탐지 기술로 나뉜다.

[0005] 네트워크 기반 탐지 기술은 봇 제어를 위한 명령 트래픽이나 봇이 발생시키는 악성 트래픽의 중앙 집중형 구조를 이용한다. 봇은 여러 대가 네트워크를 형성하고, 소수의 명령 제어 서버(Command and control server, C&C)에 의해 제어되므로 중앙 집중형 구조를 가진다. 하지만 최근 봇은 분산형 구조를 보이고 있다.

[0006] 호스트 기반 탐지 기술은 컴퓨터에서 메모리 감시를 통해, 특정 정보가 사용자의 인가 없이 전파되는 과정을 추적하는 것이 일반적이다. 하지만 모든 정보의 흐름을 모두 추적, 기록하는 것은 한계가 있다.

[0007] 최신 봇은 HTTP, P2P 혹은 융합(Hybrid) 프로토콜을 이용하여 특징을 다양화 하고, 비동기식 통신을 이용하며, 통신의 암호화를 이루는 등 기존의 연구로는 탐지가 어려운 지경에 있다.

[0008] 본 발명에서는 봇의 특성인 컴퓨터 사용자와의 비동기성을 이용하여, 봇을 효과적으로 탐지할 수 있는 새로운 방식을 제안하고자 한다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 일부 실시예는 봇 고유의 특성을 이용하여, 사용자의 인터랙션 정보와 API 호출 정보에 기초하여 봇을 관리하는 프로세스를 탐지하는 방법 및 장치를 제공한다.

과제의 해결 수단

[0010] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 제 1 측면은 사용자 단말에서 봇(Bot) 실행을 관리하는 프로세스의 탐지 방법에 관한 것으로, (a) 상기 사용자 단말에서의 패킷 발생 여부, API의 호출 여부 및 유저 인터랙션의 발생 여부를 모니터링하는 단계, (b) 상기 모니터링 결과에 따라 패킷 발생 정보, API의 호출 정보 및 유저 인터랙션 정보를 각각 패킷 테이블, API 테이블 및 유저 인터랙션 테이블에 저장하는 단계, (c) 상기 사용자 단말에서 패킷이 발생한 경우, 상기 유저 인터랙션의 발생에 따라 상기 패킷이 발생하였는지 여부에 기초하여 상기 발생한 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 단계 및 (d) 상기 봇에 의하여 발생한 것으로 분류된 패킷과 상기 API 테이블에 각 API의 호출 정보에 매칭되어 저장된 프로세스들 간의 상관도에 기초하여, 상기 봇에 의하여 발생한 패킷으로 분류된 패킷을 관리하는 프로세스를 봇 프로세스로서 검출하는 단계를 포함한다.

[0011] 또한, 본 발명의 제 2 측면은 사용자 단말에서 봇(Bot)에 의하여 발생한 패킷을 분류하는 방법에 관한 것으로, (a) 상기 사용자 단말에서의 패킷 발생 여부 및 유저 인터랙션의 발생 여부를 모니터링하는 단계, (b) 상기 모니터링 결과에 따라 패킷 발생 정보 및 유저 인터랙션 정보를 각각 패킷 테이블 및 유저 인터랙션 테이블에 저장하는 단계, (c) 상기 사용자 단말에서 패킷이 발생한 경우, 상기 유저 인터랙션 테이블에 저장된 가장 최근의 유저 인터랙션 발생 시간과 상기 패킷의 발생 시간과의 차이에 기초하여 상기 발생한 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 단계를 포함한다.

[0012] 또한, 본 발명의 제 3 측면은 사용자 단말에서 봇(Bot) 실행을 관리하는 프로세스의 탐지 장치에 관한 것으로, 상기 사용자 단말에서의 패킷 발생 여부를 모니터링하는 네트워크 감시부, 상기 사용자 단말에서의 API의 호출 여부를 모니터링하는 API 감시부, 상기 사용자 단말에서의 유저 인터랙션 정보의 발생 여부를 모니터링하는 유저 인터랙션 감시부, 상기 패킷이 발생한 경우, 발생된 패킷의 속성 정보, 패킷을 관리하는 프로세스 정보 및 패킷의 발생 시간 정보가 저장되는 패킷 테이블, 상기 API의 호출이 발생한 경우, 상기 API 호출을 발생시킨 프로세스 정보 및 API 호출의 발생 시간 정보가 저장되는 API 테이블, 상기 유저 인터랙션이 발생한 경우, 상기 유저 인터랙션의 발생과 관련된 프로세스 별로 유저 인터랙션의 발생 시간 정보가 저장되

는 유저 인터랙션 테이블, 상기 패킷의 발생시간과 가장 최근의 유저 인터랙션 발생 시간의 차이에 기초하여 상기 패킷을 상기 봇에 의하여 발생한 패킷으로 분류하는 패킷 분류부 및 상기 봇에 의하여 발생한 패킷으로 분류된 패킷과 상기 API 테이블에 저장된 프로세스들 간의 상관도에 기초하여, 상기 봇에 의하여 발생한 패킷으로 분류된 패킷을 관리하는 프로세스를 봇 프로세스로서 검출하는 봇 프로세스 검출부를 포함한다.

발명의 효과

[0013] 전술한 본 발명의 과제 해결 수단에 의하면, 사용자 단말 내에서 공격 트래픽을 발생시킨 봇 프로세스를 용이하게 탐지할 수 있으므로, 사용자 단말에서 발생하는 공격 트래픽을 용이하게 차단할 수 있다. 또한, 악성 봇에 의하여 정상 사용자의 서비스가 잘못 차단될 가능성을 감소시킨다. 특히, 본원 발명에서는 사용자의 인터랙션 정보와 API 호출 정보에 기초하여 봇을 관리하는 프로세스를 탐지하므로, 비교적 단순한 알고리즘으로 봇 프로세스를 탐지할 수 있다.

도면의 간단한 설명

[0014] 도 1은 본원 발명의 일 실시예에 따른 봇 프로세스 탐지 장치를 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따라 패킷 테이블 저장부(160)에 저장되는 패킷 테이블의 예시를 도시한 도면이다.

도 3은 사용자 인터랙션 정도에 따른 악성행위의 분류를 도시한 도면이다.

도 4는 본 발명의 일 실시예에 따른 유저 인터랙션 테이블을 도시한 도면이다.

도 5는 본 발명의 일 실시예에 따른 봇 프로세스 탐지 방법을 도시한 순서도이다.

도 6은 패킷의 특성과 유저 인터랙션과의 관계를 설명하기 위한 도면이다.

도 7은 API 호출과 악성 행위와 관련된 패킷의 상관도를 도시하는 예시적인 도면이다.

발명을 실시하기 위한 구체적인 내용

[0015] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0016] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0017] 도 1은 본원 발명의 일 실시예에 따른 봇 프로세스 탐지 장치를 도시한 도면이다.

[0018] 도시된 사용자 단말(10)은 봇 마스터에 의하여 감염될 수 있는 범용 컴퓨터로서 중앙 처리 장치(20), 메모리(30), 네트워크 인터페이스(40), I/O 컨트롤러(50) 및 기타 주변 장치를 포함한다. 사용자 단말(10)은 네트워크를 통해 인터넷 등에 접속할 수 있는 컴퓨터나 휴대용 단말기로 구현될 수 있다. 여기서, 컴퓨터는 예를 들어, 웹 브라우저(WEB Browser)가 탑재된 노트북, 데스크톱(desktop), 랩톱(laptop) 등을 포함하고, 휴대용 단말기는 예를 들어, 휴대성과 이동성이 보장되는 무선 통신 장치로서, PCS(Personal Communication System), GSM(Global System for Mobile communications), PDC(Personal Digital Cellular), PHS(Personal Handyphone System), PDA(Personal Digital Assistant), IMT(International Mobile Telecommunication)-2000, CDMA(Code Division Multiple Access)-2000, W-CDMA(W-Code Division Multiple Access), Wibro(Wireless Broadband Internet) 단말, 스마트 폰 등과 같은 모든 종류의 핸드헬드(Handheld) 기반의 무선 통신 장치를 포함할 수 있다.

[0019] 본원 발명에 따르면, 사용자 단말(10)에 설치된 봇 프로세스 탐지 모듈(100)을 통해 사용자 단말(10)이 봇에 감염되었는지 여부를 탐지할 수 있고, 봇 공격과 관련된 네트워크 트래픽이 발생하면 이를 비정상 트래픽 혹은 공격 트래픽으로 간주하고 이를 차단한다.

[0020] 봇 프로세스 탐지 모듈(100)은 네트워크 감시부(110), API 감시부(120), 유저 인터랙션 감시부(130), 패킷 분류부(140), 봇 프로세스 검출부(150), 패킷 테이블 저장부(160), APT 테이블 저장부(170) 및 유저 인터랙션 테이블 저장부(180)를 포함한다.

[0021] 참고로, 본 발명의 실시예에 따른 도 1에 도시된 구성 요소들은 소프트웨어 또는 FPGA(Field Programmable Gate Array) 또는 ASIC(Application Specific Integrated Circuit)와 같은 하드웨어 구성 요소를 의미하며, 소정의 역할들을 수행한다.

[0022] 그렇지만 '구성 요소들'은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니며, 각 구성 요소는 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다.

[0023] 따라서, 일 예로서 구성 요소는 소프트웨어 구성 요소들, 객체지향 소프트웨어 구성 요소들, 클래스 구성 요소들 및 태스크 구성 요소들과 같은 구성 요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다.

[0024] 구성 요소들과 해당 구성 요소들 안에서 제공되는 기능은 더 작은 수의 구성 요소들로 결합되거나 추가적인 구성 요소들로 더 분리될 수 있다.

[0025] 네트워크 감시부(110)는 사용자 단말(10)의 네트워크 인터페이스(40)를 통한 패킷의 전송 또는 수신 여부를 감시하고, 패킷의 속성 정보를 패킷 테이블 형태로 저장한다. 패킷의 속성 정보는 목적지 어드레스 및 포트 정보, 소스 어드레스 및 포트 정보를 포함한다. 즉, 네트워크 감시부(110)는 전송하는 패킷의 목적지 어드레스와 포트 정보를 저장하고, 수신하는 패킷의 소스 어드레스와 포트 정보를 저장하도록 한다.

[0026] 또한, 각 패킷의 전송 또는 수신과 관련한 프로세스를 확인하여, 각 프로세스 별로 시간대별로 패킷의 전송 또는 수신횟수를 저장한다. 이를 위해, 사용자 단말(10)의 중앙 처리 장치(20)에 의하여 결정되는 프로세스 아이디(PID)를 검출하고, 프로세스 아이디에 의하여 구별되는 각 프로세스에 대하여 패킷의 전송 또는 수신 횟수를 시간대별로 저장한다.

[0027] 도 2는 본 발명의 일 실시예에 따라 패킷 테이블 저장부(160)에 저장되는 패킷 테이블의 예시를 도시한 도면이다.

[0028] 도시된 바와 같이, 수신 패킷($\check{P}$)과 송신 패킷($\hat{P}$)을 구분하고, 각 패킷의 송신 또는 수신과 관련한 프로세스 아이디에 기초하여, 패킷의 시간대(Ti)별 송신횟수 또는 수신횟수를 기록한다. 도시된 패킷 테이블의 예를 기준으로 설명하면, 제 1 수신 패킷($\check{P}_1, (PID_1)$)은 제 1 프로세스 아이디(PID1)에 의하여 식별되는 프로세스에 의해 관리되는 것으로, 제 1 시간구간(T1)에서는 3회 수신되었고, 제 2 시간구간(T2)에서는 4회 수신되었음을 확인할 수 있다.

[0029] 마찬가지로, 제 n 송신 패킷($\hat{P}_N, (PID_N)$)은 제 n 프로세스 아이디(PID_N)에 의하여 식별되는 프로세스에 의해 관리되는 것으로, 제 1 시간구간(T1)에서는 4회 송신되었고, 제 2 시간구간(T2)에서는 4회 송신되었음을 확인할 수 있다.

[0030] 한편, 모든 패킷에 대하여 프로세스 아이디를 검출할 수 있는 것은 아니다. 특히, 악성 행위에 사용되는 패킷의 경우 프로세스 아이디를 검출할 수 없는 경우가 있으며, 이러한 경우에는 프로세스 아이디 정보가 없이 패킷 테이블을 구성하게 된다. 즉, 송신 패킷의 시간대별 송신 횟수 또는 수신 패킷의 시간대별 수신 횟수만이 기록될 뿐, 해당 패킷에 대한 프로세스 아이디에 대한 정보는 저장되지 않을 수 있다.

[0031] 이와 같이, 네트워크 감시부(110)는 사용자 단말(10)에서의 패킷의 송수신 여부를 확인하고, 송수신되는 패킷의 정보를 패킷 테이블에 저장한다.

[0032] 다시 도 1을 참조하면, API 감시부(120)는 사용자 단말(10)에서 API(Application Programming Interface)의 호출 여부를 검출하고, 해당 API를 호출한 프로세스에 대한 정보를 API 테이블 형태로 저장한다.

[0033] 이때, API 후킹 기술을 이용할 수 있으며, 이를 위해 후킹 DLL(dynamic link library)등을 이용할 수 있으나, 반드시 이에 한정되는 것은 아니다. 바람직하게는, 사용자 단말(10)의 동작 중 패킷을 외부로 전송하는 처리

동작과 관련한 주요 API를 기준으로 하여 API의 호출 여부를 검출한다.

[0034] 예를 들어, send(), sendto(), InternetConnction(), InternetWrite-File(), HttpSendRequestEx()와 같은 API의 호출 여부를 감시한다.

[0035] API 테이블은 도 2에서 설명한 패킷 테이블과 유사한 형태로 각 API를 호출한 프로세스 아이디와 시간대별 호출횟수 정보를 포함한다. 즉, 해당 API를 호출한 프로세스의 프로세스 아이디 정보가 함께 매칭되어 저장된다. 또한, 해당 API의 시간대별 호출횟수 정보도 함께 저장된다. 이때, 패킷 테이블과는 달리 각 API에 대해서는 프로세스 아이디가 반드시 함께 매칭되어 저장된다. 즉, 패킷 테이블의 경우 악성행위에 사용되는 패킷에 대해서는 프로세스 아이디를 같이 저장할 수 없는 경우가 있으나, API는 각 API를 호출하는 프로세스를 모두 확정할 수 있기 때문에, 각 API에 대한 프로세스 아이디를 매칭시켜 저장할 수 있다.

[0036] 유저 인터랙션 감시부(130)는 사용자의 요청 행위 또는 사용자에 대한 보고 행위와 같은 유저 인터랙션(user interaction) 행위 발생 여부를 검출하고, 해당 유저 인터랙션을 발생시킨 프로세스에 대한 정보를 유저 인터랙션 테이블 형태로 저장한다.

[0037] 유저 인터랙션은 크게 요청 인터랙션(request interaction)과 보고 인터랙션(report interaction)으로 구분될 수 있다. 요청 인터랙션은 사용자 단말(10)에서 실행되는 각종 프로세스의 동작을 위해, 키보드, 마우스, 터치스크린 또는 마이크 등 각종 입력 장치를 통해 입력되는 사용자의 각종 입력 행위를 포함한다. 또한, 보고 인터랙션은 사용자에게 각종 프로그램의 상태를 알리기 위해 실행되는 모든 이벤트를 포함한다.

[0038] 사용자 인터랙션에 의하여 발생되는 패킷은 그렇지 않은 패킷에 비하여 안정성이 높은 것으로 판단할 수 있다. 그러나, 그렇지 않은 패킷의 경우 봇에 의하여 발생된 패킷으로 악성행위에 사용될 가능성이 높은 것으로 간주할 수 있다. 따라서, 본원 발명에서는 패킷과 사용자 인터랙션과의 관계에 기초하여 봇에 의하여 발생된 패킷인지 여부를 판단하도록 한다.

[0039] 도 3은 사용자 인터랙션 정도에 따른 악성행위의 분류를 도시한 도면이다.

[0040] 도시된 바와 같이, 사용자의 요청 인터랙션과 보고 인터랙션이 모두 관련된 처리행위는 웹 서핑, 문서 작성과 같이 위험도가 낮은 행위로 판단할 수 있다.

[0041] 또한, 요청 인터랙션과 관련된 행위나 보고 인터랙션과 관련된 행위는 위험도가 낮은 행위로 판단할 수 있다.

[0042] 그러나, 요청 인터랙션 및 보고 인터랙션과 관련되지 않은 처리행위는 DDoS 론칭, 스팸메일 송부, 정보 도난과 같이 위험도가 높은 행위로 판단할 수 있다.

[0043] 본원 발명에서는 이러한 인터랙션 정보에 기초하여 위험성을 판단하기 위해, 유저 인터랙션 테이블을 구성한다.

[0044] 도 4는 본 발명의 일 실시예에 따른 유저 인터랙션 테이블을 도시한 도면이다.

[0045] 도시된 바와 같이, 각 프로세스 별로 각 프로세스가 실행된 시점에 대한 정보를 유저 인터랙션 테이블에 저장한다. 또한, 각 프로세스가 실행될 때마다, 실행시간 정보를 업데이트하여 가장 최근의 실행 시간 정보가 저장되도록 한다.

[0046] 다시 도 1을 참조하면, 패킷 분류부(140)는 패킷이 송신 또는 수신되었을 때 해당 패킷과 사용자 인터랙션과의 관련성에 기초하여, 패킷이 악성행위를 위한 봇에 의하여 발생된 것인지 여부를 판단한다.

[0047] 추가적으로, 패킷 분류부(140)는 사용자 단말(10)에서 송신되는 패킷이 스푸핑(spoofing)을 위한 것인지를 판단한다. 스푸핑은 봇이 자신의 위치를 은닉하기 위해 패킷에 기록되는 소스 어드레스 정보를 봇이 설치된 사용자 단말(10)의 어드레스가 아닌 제 3 자의 어드레스 정보로 교체하는 행위를 의미한다. 따라서, 외부로 송신되는 패킷의 소스 어드레스가 사용자 단말(10)의 어드레스와 다른 경우에는 스푸핑 행위와 관련된 것으로 간주한다.

[0048] 추가적으로, 패킷 분류부(140)는 송신되는 패킷의 양과 수신되는 패킷의 양을 비교하여, 대칭성이 떨어지는 경우 DDoS 트래픽 발생행위로 판단한다. 정상적인 통신의 경우, 송신되는 패킷의 양과 수신되는 패킷의 양은 대체로 같은 비율을 유지한다. 그러나, DDoS 트래픽과 관련된 송신 패킷의 경우 그 데이터 양이 상당히 큰데 반하여, 수신 패킷은 매우 작게 된다. 따라서, 송신 패킷에 비하여 수신 패킷의 양이 매우 작은 경우에는 DDoS 트래픽 발생행위로 판단한다.

[0049] 추가적으로, 패킷 분류부(140)는 스팸메일의 발송 여부를 판단한다. 메일 전송 프로토콜(SMTP)을 이용한 단

위 시간당 메일 발송 횟수가 임계값을 초과하면 스팸메일 발송행위로 판단한다.

[0050] 봇 프로세스 검출부(150)는 패킷 분류부(140)를 통해 악성행위의 발생 여부가 판단된 경우, 해당 악성행위를 유발시킨 프로세스를 검출한다. 통상적인 경우, 특정 패킷과 관련한 프로세스를 패킷 테이블 저장부(160)를 통해 확인할 수 있다. 그러나, 악성행위와 관련된 패킷의 대부분은 포트 정보 등을 은닉하기 때문에, 패킷의 발생과 관련한 프로세스를 확인할 수 없는 경우가 대부분이다. 따라서, 패킷 테이블 저장부(160)에는 프로세스 아이디가 저장되지 않은 패킷 테이블이 생성될 수 있음은 앞에서도 설명한 바와 같다.

[0051] 본원 발명에서는 API 호출과 관련한 프로세스의 정보는 필수적으로 저장된다는 특성을 이용하여 악성행위와 관련된 프로세스를 확인하고자 한다.

[0052] 이를 위해 상관 계수수를 이용하여, 프로세스 아이디를 확인한다. 즉, 패킷 테이블에 저장된 패킷과 API 테이블에 저장된 프로세스간의 상관 계수를 산출하고, 상관도가 높은 것으로 판단된 경우, API 테이블에 저장된 프로세스가 패킷 테이블에 저장된 패킷을 관리하는 것으로 확정할 수 있다.

[0053] 본원 발명에서는 피어슨 상관 계수를 산출하여 패킷 테이블에 저장된 패킷과 API 테이블에 저장된 프로세스 간의 상관계수를 산출한다. 상세한 수학식은 아래 수학식 1과 같다.

[0054] [수학식 1]

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

[0055]

[0056] 이때, X는 $X = AT\{PID, T_N\}$와 같이 정의되고, API 테이블에 저장된 특정 프로세스를 나타낸다. Y는 $Y = PT\{(P, PID), T_N\}$ 와 같이 정의되고, 패킷 테이블에 저장된 특정 패킷을 나타낸다.

[0057] 상기 수학식 1에 의하여 산출된 상관도에 따라 어떠한 프로세스가 특정 패킷을 관리하는지 여부를 확인할 수 있다.

$$\rho_{X,Y} = \begin{cases} 1 & \text{positive correlated} \\ 0 & \text{uncorrelated} \\ -1 & \text{negative correlated} \end{cases}$$

[0058]

[0059] 즉, 상관도가 1인 경우에는 관련성이 매우 높은 것으로 보고, 0인 경우에는 관련성이 없는 것으로 본다.

[0060] 이제, 본 발명을 이용한 봇 공격과 관련한 프로세스를 탐지하는 방법을 설명하기로 한다.

[0061] 도 5는 본 발명의 일 실시예에 따른 봇 프로세스 탐지 방법을 도시한 순서도이다.

[0062] 먼저, 네트워크 감시부(110)를 통해, 패킷의 발생여부를 모니터링하고(S510), 패킷의 발생여부를 확인한다(S512). 이때, 모니터링 대상이 되는 패킷은 송신 패킷과 수신 패킷을 포함한다. 따라서, 외부로 전송할 패킷이 발생된 경우 또는 외부로부터 패킷을 수신한 경우 이를 감지한다.

[0063] 다음으로, 발생된 패킷의 속성 정보, 해당 패킷을 관리하는 프로세스 정보, 패킷의 발생 시간 정보 등을 확인하고(S514), 패킷 테이블을 갱신한다(S516). 패킷 테이블의 구성에 대해서는 도 2에서 설명한 바와 같다. 이때, 프로세스 정보는 사용자 단말(10)에서 부여한 프로세스 아이디 정보를 포함하며, 악성 행위와 관련된 패킷의 경우 프로세스 아이디를 확인할 수 없을 수 있다.

[0064] 이와 같이, 상기 단계(S510~S516)를 통해 패킷의 발생여부를 모니터링하고, 모니터링 결과에 따라 패킷 테이블을 갱신한다.

[0065] 패킷 모니터링 동작과 병렬적으로, API 감시부(120)를 통해 API의 호출 여부를 모니터링 하고(S520), API의 호출여부를 확인한다(S522). 이때, 모니터링 대상이 되는 API는 사용자 단말(10)의 동작 중 패킷을 외부로 전송하는 처리 동작과 관련한 주요 API를 포함한다.

[0066] 다음으로, 호출된 API를 관리하는 프로세스 정보, API의 호출 시간 정보 등을 확인하고(S524), API 테이블을 갱신한다(S526). API 테이블의 구성에 대해서는 앞서 설명한 바와 같다. 이때, 프로세스 정보는 사용자 단

말(10)에서 부여한 프로세스 아이디 정보를 포함한다.

[0067] 이와 같이, 상기 단계(S520~S526)를 통해 API의 호출여부를 모니터링하고, 모니터링 결과에 따라 API 테이블을 갱신한다.

[0068] 또한, 패킷 모니터링 및 API 호출 모니터링과 병렬적으로, 유저 인터랙션 감시부(130)를 통해 유저 인터랙션의 발생 여부를 모니터링하고(S530), 유저 인터랙션의 발생을 확인한다(S532). 이때, 모니터링 대상이 되는 유저 인터랙션은 요청 인터랙션과 보고 인터랙션을 포함한다.

[0069] 다음으로, 발생된 유저 인터랙션을 관리하는 프로세스 정보 및 유저 인터랙션의 발생 시간 정보 등을 확인하고(S534), 유저 인터랙션 테이블을 갱신한다(S536). 유저 인터랙션 테이블의 구성에 대해서는 앞서 설명한 바와 같다. 이때, 프로세스 정보는 사용자 단말(10)에서 부여한 프로세스 아이디 정보를 포함한다.

[0070] 이와 같이, 상기 단계(S530~S536)를 통해 유저 인터랙션의 발생여부를 모니터링하고, 모니터링 결과에 따라 유저 인터랙션 테이블을 갱신한다.

[0071] 다음으로, 패킷의 발생시간과 유저 인터랙션 시간의 차이값을 산출한다(S540).

[0072] 다음으로, 산출된 차이값이 임계값보다 작은 경우에는, 발생된 패킷이 유저 인터랙션에 의하여 발생된 것으로 판단하고, 봇에 의하여 발생된 패킷이 아닌 것으로 판단한다(S542).

[0073] 그러나, 산출된 차이값이 임계값보다 큰 경우에는, 패킷이 사용자에 의하여 발생된 것이 아닌, 봇에 의하여 발생된 패킷인 것으로 판단한다. 바람직하게는, 임계값은 1초로 설정한다. 통상적인 사용자 단말의 컴퓨팅 성능을 고려할 때, 유저 인터랙션이 발생한 이후 수 밀리 초(ms) 내에 프로세스가 실행되고, 패킷이 발생하게 된다. 다만, 상기 임계값은 사용자 단말의 통상적인 컴퓨팅 능력을 고려하여 다양한 값으로 변경될 수 있다.

[0074] 다음으로, 산출된 차이값이 임계값 보다 큰 경우에는 발생된 패킷이 어떠한 악성행위에 관한 것인지 여부를 판단한다(S544, S546, S548). 본 판단 단계는 악성행위의 종류에 따라 추가될 수도 있고, 간단을 위해 생략될 수 도 있다. 즉, 상기 단계(S542)만을 기준으로 악성행위에 해당되는 것으로 결정할 수도 있다. 또한, 각 단계(S544, S546, S548)의 실행 순서는 변경될 수 있다.

[0075] 단계(S544)에서는 스푸핑 행위와 관련된 것인지 여부를 확인한다. 스푸핑은 봇이 자신의 위치를 은닉하기 위해 패킷에 기록되는 소스 어드레스 정보를 봇이 설치된 사용자 단말(10)의 어드레스가 아닌 제 3 자의 어드레스 정보로 교체하는 행위를 의미한다. 따라서, 외부로 송신되는 패킷의 소스 어드레스가 사용자 단말(10)의 어드레스와 다른 경우에는 스푸핑 행위와 관련된 것으로 간주한다.

[0076] 추가적으로, 단계(S546)에서는 DDoS 트래픽 발생 행위를 판단한다. 즉, 송신되는 패킷의 양과 수신되는 패킷의 양을 비교하여, 대칭성이 떨어지는 경우 DDoS 트래픽 발생행위로 판단한다. 정상적인 통신의 경우, 송신되는 패킷의 양과 수신되는 패킷의 양은 대체로 같은 비율을 유지한다. 그러나, DDoS 트래픽과 관련된 송신 패킷의 경우 그 데이터 양이 상당히 큰데 반하여, 수신 패킷은 매우 작게 된다. 따라서, 송신 패킷에 비하여 수신 패킷의 양이 매우 작은 경우에는 DDoS 트래픽 발생행위로 판단한다.

[0077] 구체적으로는 아래 수학식 2를 통해 패킷의 대칭성을 판단한다.

[0078] [수학식 2]

$$R_{IO} = \frac{Incoming\ packets}{Outgoing\ packets} = \frac{PT\{\check{P}_i, T\}}{PT\{\hat{P}_i, T\}}$$

[0079]

[0080] 즉, 수신 패킷을 송신 패킷으로 나누고, 그 값이 0에 가까운 경우에는 DDoS 트래픽 발생행위로 판단한다. 바람직하게는, 임계값으로는 0.5를 사용한다. 즉, 산출된 값(R_{IO})이 0.5 보다 작은 경우에는 DDoS 트래픽 발생행위로 판단한다.

[0081] 추가적으로, 단계(S548)에서는 스팸메일의 발송 여부를 판단한다. 메일 전송 프로토콜(SMTP)을 이용한 단위 시간당 메일 발송 횟수가 임계값을 초과하면 스팸메일 발송행위로 판단한다.

[0082] 다음으로, 상기 단계(S542~S548)를 통해 악성행위와 관련된 패킷이 어떠한 프로세스에 의하여 관리되는지 여부를 확인한다. 이를 위해, 악성행위와 관련된 패킷과 API 테이블에 저장된 프로세스간의 상관 계수를 산출

한다(S550). 앞서 설명한 피어슨 상관 계수 산출식을 이용하여, 상관 계수를 산출할 수 있다.

[0083] 다음으로, 상관 계수에 기초하여, 악성행위와 관련된 패킷을 발생시킨 프로세스를 확정한다(S552). 상기 산출된 상관 계수에 기초하여 산출하며, 상관 계수가 높은 프로세스를 패킷 발생 프로세스로 확정한다. 이렇게 확정된 프로세스의 실행을 차단하면, 악성행위와 관련된 패킷의 발생을 차단할 수 있다.

[0084] 이와 같이, 본원 발명에서는 사용자의 인터랙션 여부에 기초하여 패킷의 악성 행위 관련 여부를 판단한다. 또한, 악성 행위와 관련된 패킷을 발생시킨 프로세스를 확인하기 위해, API 테이블을 참조한다. 패킷 테이블을 직접 참조하여 프로세스를 확인할 수 있으나, 대부분의 악성 행위와 관련된 패킷은 직접적으로 프로세스를 확인하기 어려우므로, API 테이블에 저장된 프로세스와 패킷의 상관 계수에 기초하여 악성 행위와 관련된 패킷을 발생시킨 프로세스를 확정한다.

[0085] 도 6은 패킷의 특성과 유저 인터랙션과의 관계를 설명하기 위한 도면이다.

[0086] (a)~(d)에 도시된 바와 같이, 인터넷 접속, 메일 전송, FTP 파일전송, P2P 파일전송과 같은 통상적인 프로세스에 의하여 발생된 패킷의 경우, 유저 인터랙션(UI_{PID})과 패킷 발생이 거의 동시에 이루어지는 것을 확인할 수 있다.

[0087] 그러나, (e), (f)에 도시된 바와 같이, DDoS 공격이나 스팸 메일 전송과 같은 악성 행위와 관련된 프로세스에 의하여 발생된 패킷의 경우, 유저 인터랙션과 무관하게 패킷이 발생함을 확인할 수 있다.

[0088] 도 7은 API 호출과 악성 행위와 관련된 패킷의 상관도를 도시하는 예시적인 도면이다.

[0089] (a), (b)에 도시된 바와 같이, 악성 행위와 관련된 특정 패킷(Flood packet)을 확인할 수 있고, 이 패킷의 발생시에 호출된 여러 API의 호출 패턴을 확인할 수 있다. 이 중에서 상관관계에 기초하여 판단할 때, 특정 API(SDbot API)와 상관관계가 가장 큼을 확인할 수 있고, 이와 관련된 프로세스의 동작을 차단하도록 한다.

[0090] 또한, (c)에 도시된 바와 같이, 악성 행위와 관련된 특정 패킷(SMTP packet)을 확인할 수 있고, 이 패킷의 발생시에 호출된 특정 API(Hellbot API)와 상관관계가 가장 큼을 확인할 수 있다. 따라서, 이와 관련된 프로세스의 동작을 차단하도록 한다.

[0091] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

[0092] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍쳐를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.

[0093] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0094] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0095]

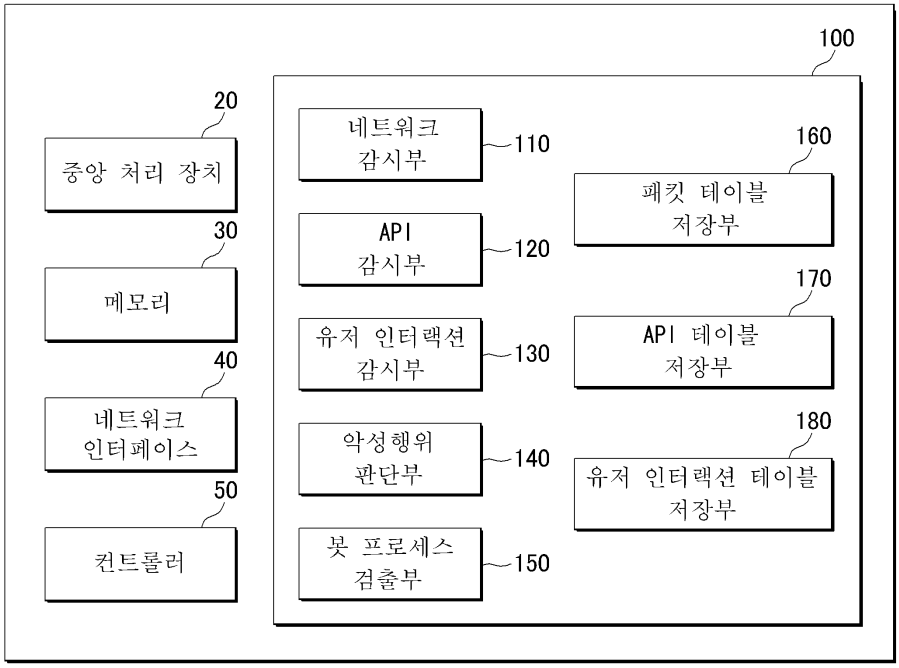
100: 봇 프로세스 탐지 모듈	110: 네트워크 감시부
120: API 감시부	130: 유저 인터랙션 감시부
140: 패킷 분류부	150: 봇 프로세스 검출부

160: 패킷 테이블 저장부 170: APT 테이블 저장부

180: 유저 인터랙션 테이블 저장부

도면

도면1



10

도면2

$$PT = \begin{array}{c} \\ \check{P}_1, (PID_1) \\ \hat{P}_1, (PID_1) \\ \check{P}_2, (PID_2) \\ \cdots \\ \hat{P}_N, (PID_N) \end{array} \begin{array}{c} T_1\ T_2\ \ldots\ T_i \\ \left[\begin{array}{c} 3,\ 4,\ \ldots,\ 1 \\ 4,\ 5,\ \ldots,\ 1 \\ 17, 6,\ \ldots,\ 9 \\ \cdots \\ 4,\ 4,\ \ldots,\ 5 \end{array}\right] \end{array}$$

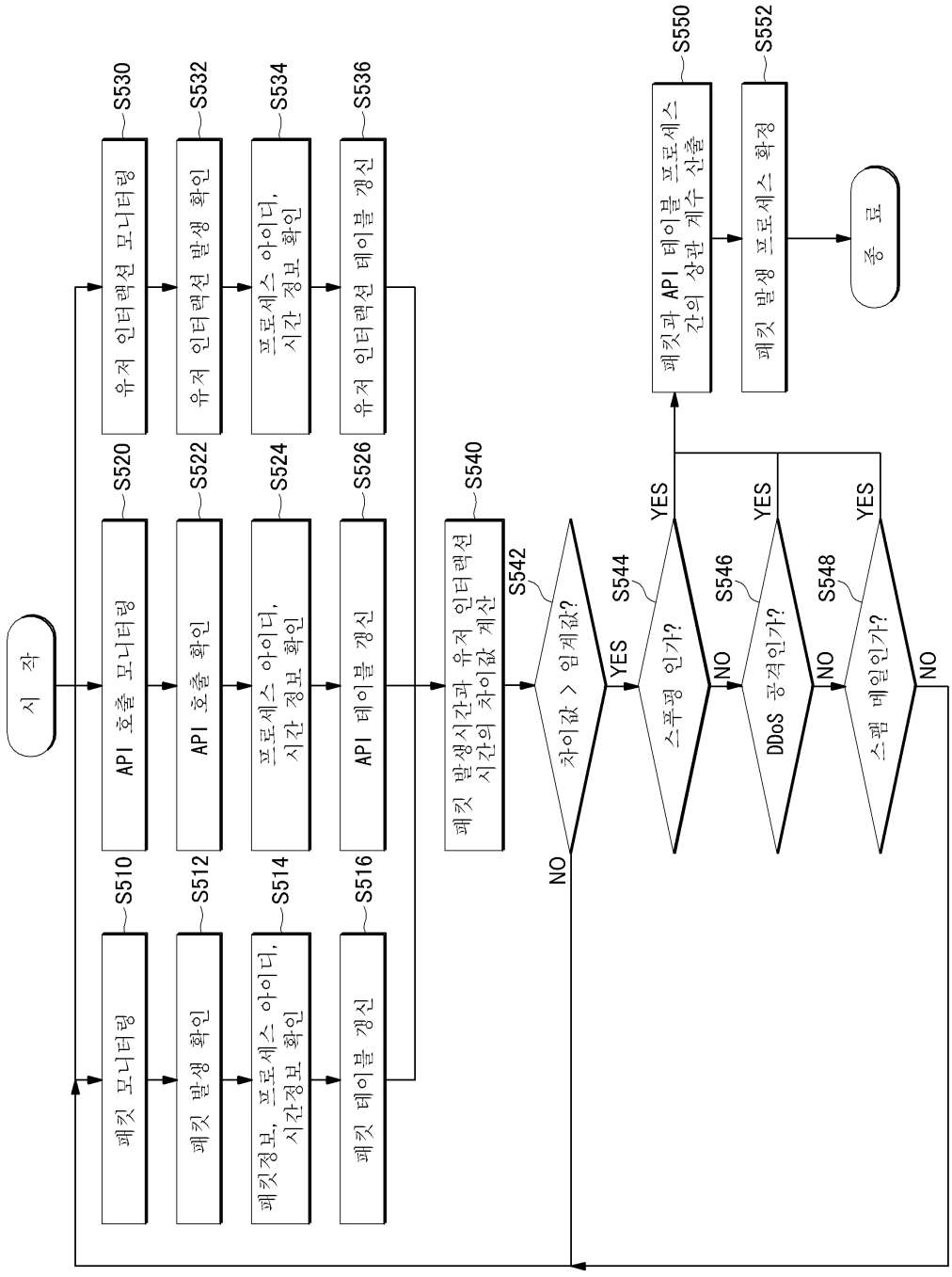
도면3

UL_{RQ}	UL_{RP}	분류	관련 행위	위험도
O	O	사용자 상호 작용 서비스 (User interactive service)	웹 서핑, 문서 작성 작업, 이메일 송부, 게임, 멀티미디어 실행	낮음
O	×	트리거 서비스 (Triggered service)	히스토리 로깅 (History logging)	중간
×	O	리포트 서비스 (Report service)	자동 업데이트 예약 작업 (Reserved work)	낮음
×	×	백그라운드 서비스 (Background service)	DDoS 론칭 스팸 메일 송부 정보 도난(Information theft)	높음

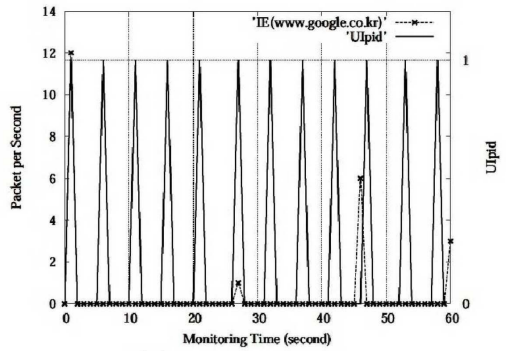
도면4

프로세스 아이디	실행 시간	
PID 1	08 : 31 : 20	
PID 2	09 : 01 : 11	
⋮	⋮	
PID n	09 : 05 : 21	

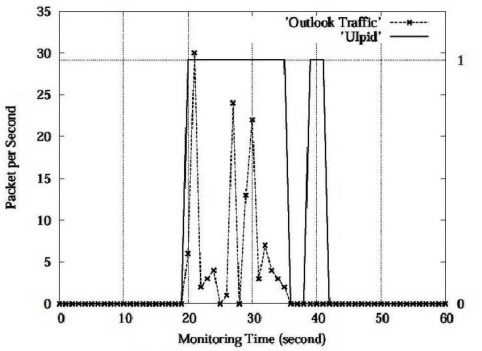
도면5


시 작
패킷 모니터링
S510
API 호출 모니터링
S520
유저 인터랙션 모니터링
S530
패킷정보, 프로세스 아이디, 시간정보 확인
S512
API 호출 확인
S522
유저 인터랙션 발생 확인
S532
패킷 발생 확인
S514
프로세스 아이디, 시간 정보 확인
S524
프로세스 아이디, 시간 정보 확인
S534
패킷 테이블 갱신
S516
API 테이블 갱신
S526
유저 인터랙션 테이블 갱신
S536
패킷 발생시간과 유저 인터랙션 시간의 차이값 계산
S540
차이값 > 임계값?
S542
YES
NO
스푸핑 인가?
S544
YES
NO
DDoS 공격인가?
S546
YES
NO
스팸 메일인가?
S548
YES
NO
패킷과 API 테이블 프로세스 간의 상관 계수 산출
S550
패킷 발생 프로세스 확정
S552
종 료

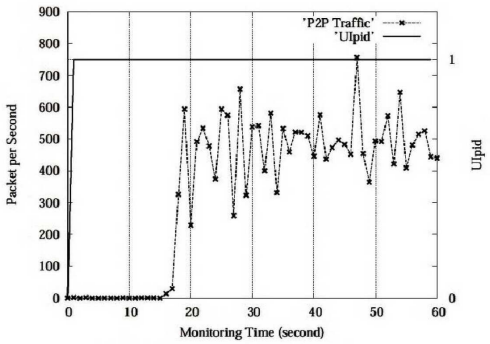
도면6



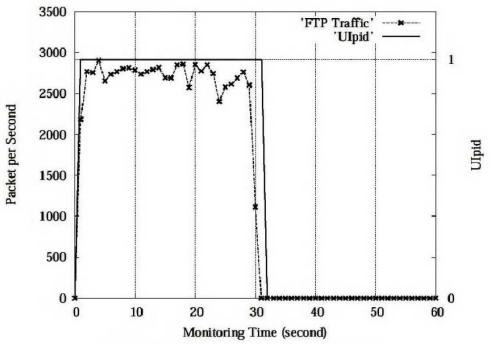
(a) Web connection



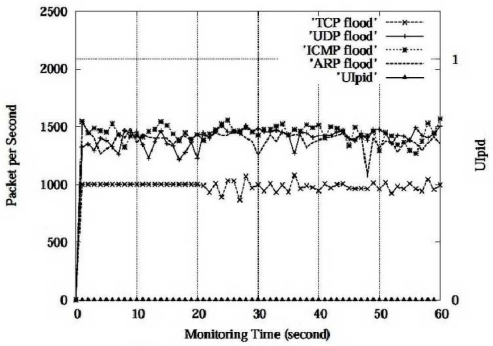
(b) Mail transfer



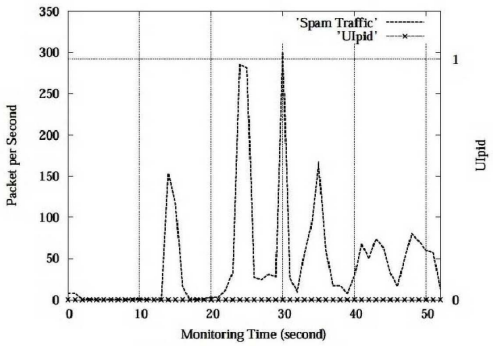
(c) FTP file transfer



(d) P2P file transfer

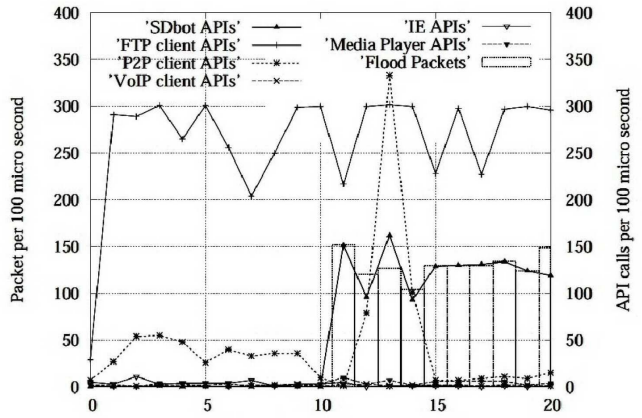


(e) DDoS attacks



(f) Spam attack

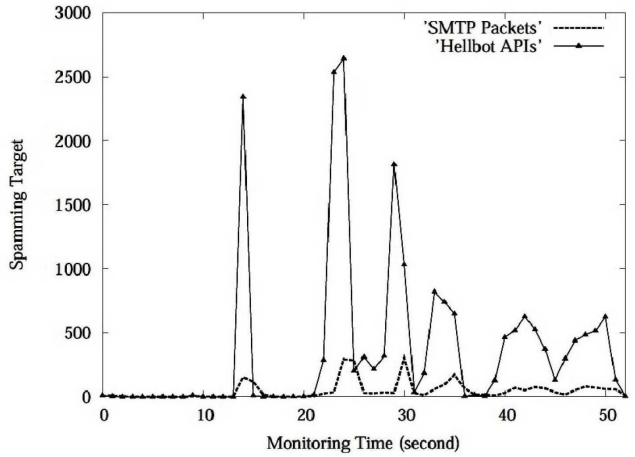
도면7



(a) Flood packets

Process	$\rho_{PID,P}$	Result	Process	$\rho_{PID,P}$	Result
SDbot	0.926548	Malicious	VoIP client	-0.06426	Normal
FTP client	0.191231	Normal	Internet Explorer	-0.16704	Normal
P2P client	0.022686	Normal	Media player	0.315773	Normal

(b)



(c) SMTP packet