



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년10월29일

(11) 등록번호 10-1564518

(24) 등록일자 2015년10월23일

(51) 국제특허분류(Int. Cl.)

H04L 12/26 (2006.01) H04L 12/813 (2013.01)

(21) 출원번호 10-2014-0185959

(22) 출원일자 2014년12월22일

심사청구일자 2014년12월22일

(56) 선행기술조사문헌

US20100284327 A1

KR1020070023408 A

(73) 특허권자

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

김명섭

세종특별자치시 조치원읍 도원로 16, 111동 902호 (푸르지오1차)

심규석

서울특별시 관악구 법원단지36길 22, 301호 (신림동)

윤성호

경상북도 포항시 북구 우창동로 157, 103동 702호 (창포동, 금호어울림아파트)

(74) 대리인

송인호, 민영준, 최관락

전체 청구항 수 : 총 10 항

심사관 : 전용해

(54) 발명의 명칭 네트워크 트래픽 탐지를 위한 규칙 자동 생성 방법 및 장치

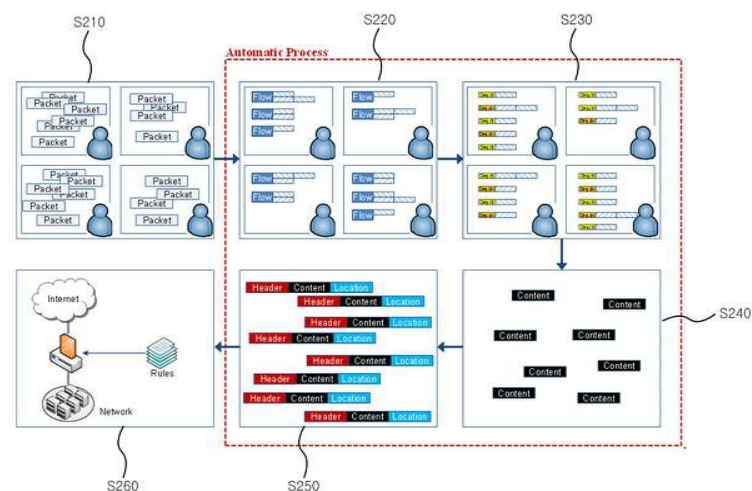
(57) 요약

본 발명은 순차 패턴 알고리즘을 이용하여 네트워크 트래픽을 자동으로 탐지하는 네트워크 트래픽 탐지를 위한 규칙 생성 방법 및 장치에 관한 것이다.

본 발명의 일실시예에 의한 규칙 자동 생성 방법은 패킷을 포함하는 트래픽을 수집하는 단계; 상기 트래픽에 포함된 복수의 패킷을 그룹핑하여 플로우를 구성하는 단계; 상기 플로우에서 식별자 및 페이로드를 추출하여 시퀀스를 구성하는 단계; 및 길이가 k-1인 콘텐츠 집합(L_{k-1})에 속하는 콘텐츠의 조합을 이용하여 길이가 k인 콘텐츠 집합(L_k)을 생성하여 후보 콘텐츠를 추출하는 단계- 상기 k는 2이상의 자연수이고, 길이가 1인 콘텐츠 집합(L_1)은

상기 구성된 시퀀스에서 추출됨-; 상기 후보 콘텐츠 간의 포함관계를 고려하여 최종 콘텐츠를 추출하는 단계; 및 상기 추출된 최종 콘텐츠를 이용하여 규칙을 생성하는 단계를 포함할 수 있다.

대표도 - 도2



명세서

청구범위

청구항 1

패킷을 포함하는 트래픽을 수집하는 단계;

상기 트래픽에 포함된 복수의 패킷을 그룹핑하여 플로우를 구성하는 단계;

상기 플로우에서 식별자 및 페이로드를 추출하여 시퀀스를 구성하는 단계; 및

길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에 속하는 콘텐츠의 조합을 이용하여 길이가 k 인 콘텐츠 집합(L_k)을 생성하여 후보 콘텐츠를 추출하는 단계- 상기 k 는 2이상의 자연수이고, 길이가 1인 콘텐츠 집합(L_1)은 상기 구성된 시퀀스에서 추출됨-;

상기 후보 콘텐츠 간의 포함관계를 고려하여 최종 콘텐츠를 추출하는 단계; 및

상기 추출된 최종 콘텐츠를 이용하여 규칙을 생성하는 단계를 포함하는 것을 특징으로 하는 규칙 자동 생성 방법.

청구항 2

제 1 항에 있어서, 상기 후보 콘텐츠 추출 단계는

상기 길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에서 설정된 최소 지지도를 만족하지 않는 콘텐츠를 삭제하는 단계를 포함하는 것을 특징으로 하는 규칙 자동 생성 방법.

청구항 3

제 2 항에 있어서, 상기 콘텐츠의 지지도는

호스트, 파일, 플로우 중 어느 하나를 기준으로 하여 산출되는 것을 특징으로 하는 규칙 자동 생성 방법.

청구항 4

제 2 항에 있어서, 상기 후보 콘텐츠 추출 단계는

길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에서 첫 문자를 제외한 길이 $k-2$ content와 마지막 문자를 제외한 길이 $k-2$ content가 동일한 콘텐츠 집합($L_{k-1} L_{k-1}$)에 속하는 콘텐츠들 간의 조합을 이용하는 단계를 포함하는 것을 특징으로 하는 규칙 자동 생성 방법.

청구항 5

제 2 항에 있어서, 상기 규칙 생성 단계는

상기 추출된 최종 콘텐츠에 매칭되는 패킷의 헤더 정보 및 상기 최종 콘텐츠의 위치 정보 중 적어도 하나를 이용하는 단계를 포함하는 것을 특징으로 하는 규칙 자동 생성 방법.

청구항 6

패킷을 포함하는 트래픽을 수집하는 수집부;

상기 트래픽에 포함된 복수의 패킷을 그룹핑하여 플로우를 구성하는 플로우 구성부;

상기 플로우에서 식별자 및 페이로드를 추출하여 시퀀스를 구성하는 시퀀스 구성부; 및

길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에 속하는 콘텐츠의 조합을 이용하여 길이가 k 인 콘텐츠 집합(L_k)을 생성하여 후보 콘텐츠를 추출하고, 상기 후보 콘텐츠 간의 포함관계를 고려하여 길이가 최종 콘텐츠로 추출하는 콘텐츠 추

출부- 상기 k는 2이상의 자연수이고, 길이가 1인 콘텐츠 집합(L_1)은 상기 구성된 시퀀스에서 추출됨-

상기 추출된 최종 콘텐츠를 이용하여 규칙을 생성하는 규칙 생성부;

상기 수집부, 상기 플로우 구성부, 상기 시퀀스 구성부, 상기 콘텐츠 추출부 및 상기 규칙 생성부를 제어하는 제어부를 포함하는 것을 특징으로 하는 규칙 자동 생성 장치.

청구항 7

제 6 항에 있어서, 상기 콘텐츠 추출부는

상기 길이가 k-1인 콘텐츠 집합(L_{k-1})에서 설정된 최소 지지도를 만족하지 않는 콘텐츠를 삭제하는 것을 특징으로 하는 규칙 자동 생성 장치.

청구항 8

제 7 항에 있어서, 상기 콘텐츠의 지지도는

호스트, 파일, 플로우 중 어느 하나를 기준으로 하여 산출되는 것을 특징으로 하는 규칙 자동 생성 장치.

청구항 9

제 7 항에 있어서, 상기 콘텐츠 추출부는

길이가 k-1인 콘텐츠 집합(L_{k-1})에서 첫 문자를 제외한 길이 k-2 content와 마지막 문자를 제외한 길이 k-2 content가 동일한 콘텐츠 집합($L_{k-1}L_{k-1}$)에 속하는 콘텐츠들 간의 조합을 이용하는 것을 특징으로 하는 규칙 자동 생성 장치.

청구항 10

제 7 항에 있어서, 상기 규칙 생성부는

상기 추출된 최종 콘텐츠에 매칭되는 패킷의 헤더 정보 및 상기 최종 콘텐츠의 위치 정보 중 적어도 하나를 이용하는 것을 특징으로 하는 규칙 자동 생성 장치.

발명의 설명

기술 분야

[0001]

본 발명은 네트워크 트래픽 탐지를 위한 규칙 자동 생성 방법 및 장치에 관한 것으로, 보다 상세하게는 순차 패턴 알고리즘을 이용하여 네트워크 트래픽을 자동으로 탐지하는 네트워크 트래픽 탐지를 위한 규칙 생성 방법 및 장치에 관한 것이다.

배경 기술

[0002]

네트워크 관리의 목적은 네트워크 자원을 최대한 활용하고 비정상 공격으로부터 네트워크 장비를 보호하는 것이다. 이를 위해 네트워크 관리자들은 적절한 네트워크 정책을 수립하여 관리 대상 네트워크에 적용한다. 네트워크 정책은 특정 트래픽을 차단하거나 대역폭을 조절하는 방법으로 수행되기 때문에 트래픽의 발생 원천을 알아내는 트래픽 분석이 선행되어야 한다.

[0003]

트래픽 분석은 응용을 대표하는 고유한 특성(시그니처, 규칙)을 사용하여 트래픽을 발생시킨 응용을 판별하는 것으로써, 분석 결과는 네트워크 정책뿐만 아니라, capacity planning, network provisioning, traffic engineering, fault diagnosis 등과 같은 다양한 분야에서 활용된다. 관련된 선행문헌으로 대한민국 등록특허 제10-1156008호가 있다.

[0004]

Snort는 보편적으로 사용하는 트래픽 탐지 엔진이다. 실제 상용 회사에서도 Snort 엔진을 트래픽 분석 및 탐지 장비 개발에 활용할 만큼 네트워크 트래픽 분야에서 매우 보편적인 도구이다. Snort는 기 정의된 규칙(Rule)을 사용하여 대응되는 패킷이 탐지하고 규칙에 정의된 행위(action)를 수행한다. 규칙은 패킷 단위로 적용되며, 패킷의 헤더 정보(IP address, port, protocol)와 통계 정보(packet size), 그리고 페이로드 정보(content,

pcrc, offset) 등을 사용한다.

- [0005] 일반적으로 규칙을 생성하기 위해서는 분석 대상 트래픽을 전수 조사하여 해당 트래픽에서만 관찰되는 공통된 특징을 찾는 작업을 수행하였다.
- [0006] 하지만, 이러한 방법은 규칙 생성 시간이 많이 소요되고, 추출하는 사람의 능력에 따라 생성된 규칙의 정확도가 가변적이라는 한계점을 가진다.
- [0007] 따라서 수동이 아닌 자동으로 네트워크 탐지를 위한 규칙을 신속하게 생성하는 기술에 대한 연구가 필요한 실정이다.

발명의 내용

해결하려는 과제

- [0008] 본 발명의 목적은 신속하고 일괄적인 네트워크 트래픽 탐지 규칙을 자동으로 생성할 수 있는 규칙 자동 생성 방법 및 장치를 제공하는 데 있다.
- [0009] 본 발명의 목적은 순차 패턴 알고리즘을 이용하여 네트워크 트래픽을 자동으로 탐지하는 네트워크 트래픽 탐지를 위한 규칙 생성 방법 및 장치를 제공하는 데 있다.

과제의 해결 수단

- [0010] 상기 목적을 달성하기 위해 본 발명의 일실시예에 의하면, 패킷을 포함하는 트래픽을 수집하는 단계; 상기 트래픽에 포함된 복수의 패킷을 그룹핑하여 플로우를 구성하는 단계; 상기 플로우에서 식별자 및 페이로드를 추출하여 시퀀스를 구성하는 단계; 및 길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에 속하는 콘텐츠의 조합을 이용하여 길이가 k 인 콘텐츠 집합(L_k)을 생성하여 후보 콘텐츠를 추출하는 단계- 상기 k 는 2이상의 자연수이고, 길이가 1인 콘텐츠 집합(L_1)은 상기 구성된 시퀀스에서 추출됨-; 상기 후보 콘텐츠 간의 포함관계를 고려하여 최종 콘텐츠를 추출하는 단계; 및 상기 추출된 최종 콘텐츠를 이용하여 규칙을 생성하는 단계를 포함하는 규칙 자동 생성 방법이 개시된다.
- [0011] 상기 목적을 달성하기 위해 본 발명의 일실시예에 의하면, 패킷을 포함하는 트래픽을 수집하는 수집부; 상기 트래픽에 포함된 복수의 패킷을 그룹핑하여 플로우를 구성하는 플로우 구성부; 상기 플로우에서 식별자 및 페이로드를 추출하여 시퀀스를 구성하는 시퀀스 구성부; 및 길이가 $k-1$ 인 콘텐츠 집합(L_{k-1})에 속하는 콘텐츠의 조합을 이용하여 길이가 k 인 콘텐츠 집합(L_k)을 생성하여 후보 콘텐츠를 추출하고, 상기 후보 콘텐츠 간의 포함관계를 고려하여 길이가 최종 콘텐츠로 추출하는 콘텐츠 추출부- 상기 k 는 2이상의 자연수이고, 길이가 1인 콘텐츠 집합(L_1)은 상기 구성된 시퀀스에서 추출됨-; 상기 추출된 최종 콘텐츠를 이용하여 규칙을 생성하는 규칙 생성부; 상기 수집부, 상기 플로우 구성부, 상기 시퀀스 구성부, 상기 콘텐츠 추출부 및 상기 규칙 생성부를 제어하는 제어부를 포함하는 규칙 자동 생성 장치가 개시된다.

발명의 효과

- [0012] 본 발명의 일실시예에 의한 네트워크 트래픽 탐지를 위한 규칙 자동 생성 방법 및 장치는 관리자의 수작업에 의존하지 않고도 자동으로 네트워크 트래픽 탐지 규칙을 신속하게 생성할 수 있다.
- [0013] 본 발명의 일실시예에 의하면, 전처리나 후처리 과정이 없이도 네트워크 트래픽 탐지 규칙을 생성할 수 있다.

도면의 간단한 설명

- [0014] 도 1은 본 발명의 일실시예와 관련된 규칙 자동 생성 장치를 나타내는 블록도이다.
- 도 2는 본 발명의 일실시예와 관련된 규칙 자동 생성 방법을 나타내는 흐름도이다.
- 도 3은 본 발명의 일실시예와 관련된 콘텐츠 추출 알고리즘을 나타낸다.
- 도 4는 본 발명의 일실시예와 관련된 후보 콘텐츠 알고리즘을 나타낸다.
- 도 5는 도 3 내지 도 4의 알고리즘을 수행하는 과정의 예시를 나타낸다.

도 6은 본 발명의 일실시예와 관련된 콘텐츠 정보만을 이용하여 생성된 규칙의 예를 나타낸다.

도 7은 본 발명의 일실시예와 관련된 콘텐츠 정보와 추가 정보를 이용하여 생성된 규칙의 예를 나타낸다.

도 8은 본 발명의 일실시예와 관련된 콘텐츠 위치 분석 알고리즘을 나타낸다.

도 9는 본 발명의 일실시예와 관련된 규칙 자동 생성 방법을 통해 생성된 snort content 규칙의 예를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0015] 이하, 본 발명의 일실시예와 관련된 네트워크 트래픽 탐지를 위한 규칙 자동 생성 방법 및 장치에 대해 도면을 참조하여 설명하도록 하겠다.
- [0016] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.
- [0017] 이하에서는 네트워크 트래픽 탐지를 Snort content 규칙을 네트워크 탐지 규칙의 예로 설명하도록 하겠다. Snort는 보편적으로 사용하는 트래픽 탐지 엔진이다. 실제 상용 회사에서도 Snort 엔진을 트래픽 분석 및 탐지 장비 개발에 활용할 만큼 네트워크 트래픽 분야에서 매우 보편적인 도구이다. Snort는 기 정의된 규칙(Rule)을 사용하여 대응되는 패킷이 탐지하고 규칙에 정의된 행위(action)를 수행한다. 규칙은 패킷 단위로 적용되며, 패킷의 헤더 정보(IP address, port, protocol)와 통계 정보(packet size), 그리고 페이로드 정보(content, pcre, offset) 등을 사용한다.
- [0018] 도 1은 본 발명의 일실시예와 관련된 규칙 자동 생성 장치를 나타내는 블록도이다.
- [0019] 도시된 바와 같이, 규칙 자동 생성 장치(100)는 수집부(110), 플로우 구성부(120), 시퀀스 구성부(130), 콘텐츠 추출부(140), 규칙 생성부(150), 저장부(160), 전송부(170) 및 제어부(180)를 포함할 수 있다.
- [0020] 수집부(110)는 최초 호스트 별로 분석 대상 응용 및 서비스, 혹은 악성 코드가 발생한 트래픽을 수집할 수 있다. 상기 트래픽은 패킷(packet)을 기본 단위로 한다.
- [0021] 플로우 구성부(120)는 수집된 패킷 집합 트래픽을 플로우(flow)로 구성할 수 있다.
- [0022] 이하, 실시예에서의 플로우는 5-tuple(source IP, source port, Destination IP, Destination port, Protocol)이 동일한 패킷의 집합이다.
- [0023] 시퀀스 구성부(130)는 단일 플로우에서 전송 방향이 같은 패킷들을 조합하여 하나의 시퀀스(sequence)를 구성할 수 있다. 예를 들어, 시퀀스 구성부(130)는 패킷의 구성요소 중 페이로드만을 조합하여 하나의 시퀀스(sequence)를 구성할 수 있다.
- [0024] 시퀀스(sequence)는 콘텐츠(content)를 추출하기 위한 순차 패턴 알고리즘의 입력으로 사용될 수 있다.
- [0025] 콘텐츠 추출부(140)는 순차 패턴 알고리즘을 이용하여 규칙에 적용될 콘텐츠를 추출할 수 있다. 순차 패턴 알고리즘은 입력 받은 sequence에서 길이가 1인 후보 content를 시작으로 길이를 증가시키면서 후보 content를 찾고 최종적으로 일정 수준이상의 지지도를 가지는 content를 추출할 수 있다. 본 명세서에서 content 길이는 해당 콘텐츠의 바이트 크기를 의미한다.
- [0026] 규칙 생성부(150)는 상기 콘텐츠 추출부(140)에서 추출된 콘텐츠를 이용하여 네트워크 탐지 규칙을 생성할 수 있다. 단순히 content만을 규칙으로 사용할 경우 오탐(탐지 대상이 아닌 트래픽을 탐지)의 가능성이 높기 때문에, 규칙 생성부(150)는 추가적인 정보를 분석하여 규칙에 기술할 수 있다. 예를 들어, 상기 추가 정보는 헤더 정보 및 content의 위치 정보를 포함할 수 있다.
- [0027] 저장부(160)는 생성된 규칙을 저장할 수 있다.
- [0028] 전송부(170)는 저장된 규칙을 특정 서버에 제공할 수 있다.
- [0029] 제어부(180)는 수집부(110), 플로우 구성부(120), 시퀀스 구성부(130), 콘텐츠 추출부(140), 규칙 생성부(150), 저장부(160) 및 전송부(170)를 전반적으로 제어할 수 있다.
- [0030] 도 2는 본 발명의 일실시예와 관련된 규칙 자동 생성 방법을 나타내는 흐름도이다.

[0031] 수집부(110)는 결정된 탐지 대상에 근거하여 트래픽을 수집할 수 있다(S210). 트래픽 수집은 규칙 생성을 위한 첫 단계라 할 수 있다. 탐지 대상은 응용, 서비스, 공격, 악성 코드 등 네트워크 관리 목적에 따라 매우 다양할 수 있다. 수집부(110)는 트래픽을 발생시키는 호스트에서 직접 트래픽을 수집 할 경우, wireshark, tcpdump와 같은 수집 도구를 사용하고, 네트워크 전체 트래픽을 수집할 경우, 스위치의 미러링 기능이나 탭 장비를 사용하여 수집할 수 있다.

[0032] 하기 수학적 식 1 및 수학적 식 2는 수집한 패킷의 형태를 나타낸다.

수학적 식 1

$$[0033] \text{PacketSet} = \{P_1, P_2, \dots, P_p\}$$

수학적 식 2

$$[0034] P_i = \left\{ \begin{array}{l} \text{host}_{id}, \text{srcIP}, \text{srcPort}, \text{L4Prot}, \text{dstIP}, \text{dstPort}, \\ \text{payload} = \langle a_1 a_2 a_3 \dots a_a \rangle \end{array} \right\}$$

[0035] PacketSet은 패킷들의 집합을 의미하고 단일 패킷 P는 호스트 ID, source IP address, source port, Layer4 protocol, destination IP address, destination port, 그리고 payload로 구성된다. 특히, payload는 연속된 문자들로 구성되며, 본 실시예에서 자동 생성하는 content는 payload의 부분 문자열을 의미한다.

[0036] 규칙 생성을 위한 트래픽 수집에서는 탐지 대상 트래픽만을 수집해야 하기 때문에 개별 호스트에서 트래픽을 수집하는 것이 생성된 규칙의 정확성을 높이는 측면에서 권장될 수 있다.

[0037] 한편, 본 발명의 일실시예에서 사용하는 지지도는 입력 트래픽을 발생시킨 호스트를 기준으로 하기 때문에 최소 2개 이상의 호스트에서 트래픽을 수집해야 한다. 하지만, 실제 트래픽 수집 환경에서 여러 호스트의 트래픽을 수집하는 것은 매우 번거롭고 불가능한 경우가 있기 때문에 동일한 호스트에서 수집한 트래픽을 여러 파일에 나누어 저장하고 지지도를 호스트 기준이 아닌 입력 파일 기준으로 계산하는 방법도 적용할 수 있다. 즉, 지지도를 계산하는 기준으로 트래픽 수집의 환경에 따라 변화할 수 있다. 자세한 설명은 후술하도록 하겠다.

[0038] 플로우 구성부(120)는 수집된 패킷 집합 트래픽을 플로우로 구성할 수 있다(S220). 본 실시예에서 사용한 플로우는 수학적 식 3 및 수학적 식 4와 같이 5-tuple(source IP, source port, Destination IP, Destination port, Protocol)이 동일한 패킷의 집합이다.

수학적 식 3

$$[0039] F = \{F_1, F_2, \dots, F_f\}$$

수학식 4

$$F_i = \left\{ \begin{array}{l} host_id, srcIP, srcPort, L4Prot, dstIP, dstPort, \\ forward = \{P_1, P_2, \dots, P_x | P_{1.5tupe} \dots = P_{x.5tupe}\}, \\ backward = \{P_1, P_2, \dots, P_y | P_{1.5tupe} \dots = P_{y.5tupe}\} \\ |forward.srcIP = backward.dstIP, \\ forward.srcPort = backward.dstPort, \\ forward.L4Prot = backward.L4Prot, \\ forward.dstIP = backward.srcIP, \\ forward.dstPort = backward.srcPort \end{array} \right\}$$

[0040]

[0041]

한편, 본 발명의 일실시예에 의하면, source 측과 destination측이 대칭되는 플로우는 하나의 플로우로 구성하고 각 패킷 집합에 전송 방향(forward, backward)을 기입할 수 있다. 즉, 본 발명의 일실시예에서 정의한 플로우는 5-tuple이 동일한 패킷 집합과 이와 대칭 패킷 집합을 포함하는 양방향 플로우가 될 수 있다.

[0042]

패킷을 플로우로 구성하는 이유는 비록 snort가 패킷 단위에 적용되기는 하나 네트워크의 특성상 단일 메시지가 여러 패킷으로 나누어 전송되는 경우(패킷 단편화)가 발생한다. 따라서 단일 플로우를 구성하는 패킷들을 전송 방향 별로 구분하여 페이로드를 합치면 메시지의 끊김 없이 실제 전송된 페이로드 메시지를 확인할 수 있다.

[0043]

시퀀스 구성부(130)는 플로우의 순방향(forward), 역방향(backward)으로 구분된 패킷들의 페이로드만을 추출하여 하나의 sequence를 만들 수 있다(S230). 만약, 플로우가 양방향 통신 패킷들로 구성되었다면, 2개의 sequence가 생성되고 단 방향 통신 트래픽이면 1개의 sequence가 생성된다.

[0044]

수학식 5 및 수학식 6은 시퀀스를 나타내는 수식이다.

[0045]

SequenceSet은 수학식 5와 같이 여러 sequence(S)들로 구성되며, 하나의 sequence는 수학식 6과 같이 호스트 ID와 문자열 $\langle a_1 a_2 a_3 \dots a_n \rangle$ ($a_1 a_2 a_3 \dots a_n$)로 구성될 수 있다.

수학식 5

$$SequenceSet = \{S_1, S_2, \dots, S_s\}$$

[0046]

수학식 6

$$S_i = \{host_id, \langle a_1 a_2 a_3 \dots a_n \rangle\}$$

[0047]

[0048]

시퀀스 구성부(130)는 수학식 5 및 수학식 6과 같이 구성된 sequence에 호스트 ID를 기입할 수 있다. 이는 후술할 지지도(support) 계산을 위해 사용될 수 있다. 만약 지지도 계산을 파일 기준으로 계산할 경우 시퀀스 구성부(130)는 파일 ID를 기입할 수 있다.

[0049]

컨텐츠 추출부(140)는 sequence 집합과 최소 지지도를 입력 받아 최소 지지도를 만족하는 content를 추출할 수 있다(S240).

[0050]

도 3은 본 발명의 일실시예와 관련된 컨텐츠 추출 알고리즘을 나타내고, 도 4는 본 발명의 일실시예와 관련된 후보 컨텐츠 알고리즘을 나타낸다. 도 3의 알고리즘 및 도 4의 알고리즘은 순차 패턴 알고리즘으로, 입력 받은

sequence 집합으로부터 기 정의된 최소 지지도를 만족하는 content 집합을 출력하는 방법을 나타낸다.

하기 수학적 7 및 수학적 8은 추출된 콘텐츠를 나타낸다.

수학적 7

$$ContentSet = \{C_1, C_2, \dots, C_c\}$$

수학적 8

$$C_i = \{\langle a_x a_{x+1} \dots a_y \rangle | 1 \leq x \leq y \leq n, \}$$

도 3을 참조하면, 콘텐츠 추출부(140)는 입력 받은 sequence 집합의 모든 sequence에서 길이 1인 content를 추출하여 길이 1 content 집합 $L_1 L_1$ 에 저장한다(도 3 알고리즘 Line: 1~5). 콘텐츠 추출부(140)는 길이 1인 content를 시작으로 길이를 1씩 늘려가며 모든 길이의 content를 추출하여 자신의 길이 content 집합 $L_k L_k$ 에 저장한다(도 3 알고리즘 Line: 6~20). 단, 콘텐츠 추출부(140)는 새로 생성된 집합 $L_{k-1} L_{k-1}$ 의 모든 content 중, 입력받은 최소 지지도를 만족하지 않는 content는 삭제한다(도 3 알고리즘 Line: 8~17). 최소 지지도를 만족하지 못하는 content는 content 추출 자격을 만족하지 못할 뿐만 아니라 해당 content를 확장한 content에서도 최소 지지도를 만족하지 않기 때문이다.

수학적 9는 지지도를 계산하는 수식이다.

수학적 9

$$Support = \frac{\text{Number of support hosts}}{\text{Total number of hosts}}$$

도 3의 알고리즘에서는 content의 지지도는 수학적 (9)와 같이 전체 호스트 수 중 해당 content를 발생시킨 호스트의 비율로 추정될 수 있다(도 3 알고리즘 Line: 9~13). 예를 들어 특정 content가 모든 호스트에서 발생된 트래픽에서 관찰될 경우 지지도는 1, 그렇지 않을 경우 1 보다 작은 값을 가진다.

만약, 제한된 트래픽 수집 환경에서 단일 호스트에서 수집된 트래픽을 본 알고리즘에 적용할 경우, 호스트 기준의 지지도 계산을 파일이나 플로우 기준으로 변경하여 사용할 수 있다. 최소 지지도를 만족하지 못하는 content가 삭제된 집합 $L_{k-1} L_{k-1}$ 의 content는 집합 $L_k L_k$ 을 생성하는 데 사용될 수 있다(도 3 알고리즘 Line: 18).

이때 사용하는 방법은 도 4의 알고리즘에 기술된 방법이다. 콘텐츠 추출부(140)는 입력받은 집합 $L_{k-1} L_{k-1}$ 의 content들을 비교하여 집합 $L_k L_k$ 의 content를 생성할 수 있다. 집합 $L_{k-1} L_{k-1}$ 의 content를 조합하여 집합 $L_k L_k$ 의 content를 생성하는 것은 첫 문자를 제외한 길이 k-2 content와 마지막 문자를 제외한 길이 k-2 content가 동일한 집합 $L_{k-1} L_{k-1}$ 의 content들끼리 가능하다(도 4 알고리즘 Line: 1~7).

예를 들어, 콘텐츠 추출부(140)는 집합 $L_4 L_4$ 의 content인 "abcd"와 "bcde"는 "a"를 제외한 "bcd"와 'e'를 제외한 "bcd"가 같기 때문에 집합 $L_5 L_5$ 의 content "abcde"를 생성할 수 있다.

위와 같은 방법으로 길이를 1씩 증가 시키면서 더 이상 새로운 content가 추출되지 않을 때까지 content 추출과

지지도 미만 content 삭제를 반복한다. 추출의 마지막 단계로써 추출된 모든 길이의 content의 포함 관계를 확인하고, 만약 포함 관계에 있는 content가 발견되면 해당 content를 집합에서 삭제한다(도 3 알고리즘 Line: 22). 콘텐츠 추출부(140)는 최종적으로 생성된 content 집합을 다음 단계인 규칙 생성 단계로 전달할 수 있다.

[0062] 도 5는 도 3 내지 도 4의 알고리즘을 수행하는 과정의 예시를 나타낸다.

[0063] 도 5의 예시에서는 3명의 호스트에서 발생한 트래픽으로 구성된 SequenceSet과 최소 지지도 0.6을 입력받는다. SequenceSet은 4개의 sequence로 구성되어 있다. 최소 지지도가 0.6이라는 의미는 전체 호스트 수가 3이기 때문에 최소 2명의 호스트에서 발생한 트래픽에 해당 content가 관찰되어야 한다는 의미이다.

[0064] 도 3 및 도 4의 알고리즘이 수행되면 길이 1인 모든 content를 추출한다. 길이 1인 content($\langle C \rangle \langle C \rangle$, $\langle M \rangle \langle M \rangle$, $\langle D \rangle \langle D \rangle$) 모두 최소 지지도 0.6을 만족하기 때문에 길이 2 content 생성에 사용된다. 길이 2 content 생성 후, 최소 지지도를 계산하고 만족하지 못하는 content들은 삭제한다. 콘텐츠 추출부(140)는 최소 지지도를 만족하는 길이 2 content($\langle CM \rangle \langle CM \rangle$, $\langle MD \rangle \langle MD \rangle$)를 사용하여 길이 3 content를 추출한다. 최소 지지도를 만족하는 길이 3 content의 개수가 1이기 때문에 더 이상 content의 길이를 늘리는 것은 불가능하다. 따라서 콘텐츠 추출부(140)는 content 추출을 종료한다. 추출 종료 후, 포함관계가 있는 $\langle C \rangle \langle C \rangle$, $\langle M \rangle \langle M \rangle$, $\langle D \rangle \langle D \rangle$ 는 contentSet에서 삭제되고 최종적으로 $\langle CMD \rangle \langle CMD \rangle$ 가 다음 단계인 규칙 생성 단계로 전달될 수 있다.

[0065] 규칙 생성부(150)는 추출된 최종 콘텐츠를 이용하여 규칙을 생성할 수 있다.

[0066] 추출된 최종 content 정보만을 사용하여 Snort 규칙을 작성할 경우, 오탐의 가능성이 높다. 즉, 추출된 content의 길이가 너무 짧을 경우, 탐지 대상이 아닌 트래픽에서 해당 규칙이 적용될 수 있다.

[0067] 도 6은 본 발명의 일실시예와 관련된 콘텐츠 정보만을 이용하여 생성된 규칙의 예를 나타내고, 도 7은 본 발명의 일실시예와 관련된 콘텐츠 정보와 추가 정보를 이용하여 생성된 규칙의 예를 나타낸다.

[0068] 도 6 및 도 7과 같이 content 정보만을 사용한 규칙과 그렇지 않은 규칙은 큰 차이를 보인다. 도 6의 예는 패킷 페이로드 전체를 검사하면서 해당 content가 존재하는지를 검사한다. 하지만, 도 7은 TCP 프로토콜을 사용하고, Destination IP가 111.222.333.0/24, 포트 번호는 80을 사용하여 전송하는 패킷 중 해당 content가 페이로드의 4번째 바이트부터 20번째 바이트 사이에 존재하는지를 검사한다.

[0069] 따라서 규칙 생성부(150)는 content의 위치 정보와 헤더 정보 등의 추가 정보를 분석하고, 추출된 최종 콘텐츠 외에 분석된 정보를 포함시켜 규칙을 생성할 수 있다(S250). 상기와 같이 분석된 추가 정보를 규칙에 포함시킴으로써 규칙의 오탐 가능성을 낮출 수 있을 뿐만 아니라, 상대적으로 수행 오버헤드가 큰 페이로드 검사량을 줄일 수 있기에 트래픽 탐지 시스템의 성능이 향상될 수 있다.

[0070] 규칙 생성부(150)는 추출한 content의 위치 정보를 분석하기 위해 트래픽 수집 단계에서 생성한 패킷 데이터를 사용할 수 있다. Snort는 패킷 단위로 동작하기 때문에 content의 위치는 sequence가 아닌 실제 패킷 페이로드 내의 위치로 분석해야 한다.

[0071] 도 8은 본 발명의 일실시예와 관련된 콘텐츠 위치 분석 알고리즘을 나타낸다.

[0072] 도 8의 알고리즘은 content와 packetSet이 주어졌을 때, content의 위치 정보를 분석하는 과정을 나타낸다. 도 8의 알고리즘의 출력인 offset은 해당 content가 packetSet의 패킷에 매칭 될 때, 매칭 시작 위치의 최소 바이트 위치를 의미하고 depth는 매칭 종료 위치의 최대 바이트 위치를 의미한다. 즉, 해당 content가 패킷에 매칭 될 때, 페이로드의 offset과 depth 사이에서만 매칭된다는 의미이다.

[0073] 도 8을 참조하면, 규칙 생성부(150)는 최초 offset은 패킷의 최대 크기로 depth는 0으로 초기화 한다(도 8의 알고리즘 Line: 1~2). 그리고 규칙 생성부(150)는 packetSet의 모든 패킷을 순회하며 offset과 depth를 조정한다. 규칙 생성부(150)는 입력으로 받은 content와 패킷의 매칭 여부가 확인하고, 만약 매칭이 된다면 시작 바이트 위치를 얻어와 현재 offset과 비교한다. 만약, 현재 offset 보다 작은 값이면, 규칙 생성부(150)는 해당 값은 현재 offset으로 변경한다. depth의 경우 종료 바이트 위치를 얻어와 현재 depth 보다 큰 값이면, 규칙 생성부(150)는 해당 값을 현재 depth로 변경한다(도 8의 알고리즘 Line: 4~6).

[0074] 규칙 생성부(150)는 최종 결정된 offset과 depth값을 content 규칙에 추가한다.

[0075] 규칙 생성부(150)는 추출한 content의 헤더 정보를 분석하기 위해 앞서 설명한 위치 정보 분석 단계와 유사한 과정을 수행할 수 있다. 규칙 생성부(150)는 packetSet의 모든 패킷을 순회하며, 해당 content와 매칭 여부를

확인한다. 만약, 매칭이 된다면 규칙 생성부(150)는 해당 패킷의 헤더 정보를 저장한다. 모든 패킷을 검사한 후, 저장된 헤더 정보가 고유한 한가지 값을 가지는 경우, 규칙 생성부(150)는 해당 헤더 정보를 content 규칙에 추가한다. 규칙 생성부(150)는 IP의 경우 CIDR 값을 32, 24, 16 순으로 감소시키면서 고유한 값이 추출될 때까지 반복한다. 즉, 규칙 생성부(150)는 CIDR 값이 32인, D 클래스 IP로 고유한 값을 찾는 것을 시도하고, 만약 찾지 못하면, CIDR 값을 24로 적용하여 C 클래스 IP를 찾는다. 예를 들어 해당 content가 매칭 되는 Destination IP가 CIDR 32로 "111.222.333.1/32"과 "111.222.333.2/32"가 추출되면, CIDR 24로 설정하고 "111.222.333.0/24"를 추출한다. 규칙 생성부(150)는 CIDR 16을 적용해도 고유한 IP를 추출하지 못할 경우 "any"로 추출할 수 있다.

[0076] 규칙 생성부(150)는 port의 경우에도 고유한 값이 추출되면 해당 값을 사용하고 고유하지 않은 여러 값이 추출되면 "any"로 추출할 수 있다.

[0077] 상술한 콘텐츠의 위치 정보, 헤더 정보 등의 추가 정보가 추가되어 규칙이 생성되면, 상기 생성된 규칙은 저장부(160)에 저장되고, 외부(개인 또는 단체)로 전송될 수 있다(S260).

[0078] 생성된 규칙은 다음과 같이 활용될 수 있다.

[0079] 첫째, 상기 생성된 일정 금액을 지불한 구독자에게 정기적으로 배부될 수 있다. 실제 www.snort.org에서도 일정 금액을 지불하는 개인 및 단체에게 최신 규칙을 제공한다. 둘째, 규칙 자동 생성 서비스를 제공하는 웹사이트를 운영할 수 있다. 사용자가 업로드한 트래픽 데이터에서 규칙을 자동 생성하고 생성된 결과를 실시간으로 확인할 수 있는 서비스를 제공할 수 있다. 마지막으로 본 시스템을 활용하기 원하는 네트워크 장비 개발 회사에 본 시스템을 설치하고 유지 관리 서비스를 제공할 수 있다.

[0080] 도 9는 본 발명의 일실시예와 관련된 규칙 자동 생성 방법을 통해 생성된 snort content 규칙의 예를 나타낸다.

[0081] 도식된 규칙의 예는 대표적인 인터넷 응용, 서비스 7종을 선정하여, 해당 응용을 사용할 때 발생하는 트래픽 수집을 통해 생성된 규칙이다.

[0082] 전술한 바와 같이, 본 발명의 일실시예에 의한 네트워크 트래픽 탐지를 위한 규칙 자동 생성 방법 및 장치는 관리자의 수작업에 의존하지 않고도 자동으로 네트워크 트래픽 탐지 규칙을 신속하게 생성할 수 있다.

[0083] 본 발명의 일실시예에 의하면, 전처리나 후처리 과정이 없이도 네트워크 트래픽 탐지 규칙을 생성할 수 있다.

[0084] 상술한 규칙 자동 생성 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터로 판독 가능한 기록 매체에 기록될 수 있다. 이때, 컴퓨터로 판독 가능한 기록매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 한편, 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0085] 컴퓨터로 판독 가능한 기록매체에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광매체(Magneto-Optical Media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.

[0086] 한편, 이러한 기록매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다.

[0087] 또한, 프로그램 명령에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0088] 상기와 같이 설명된 규칙 자동 생성 방법 및 장치는 상기 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

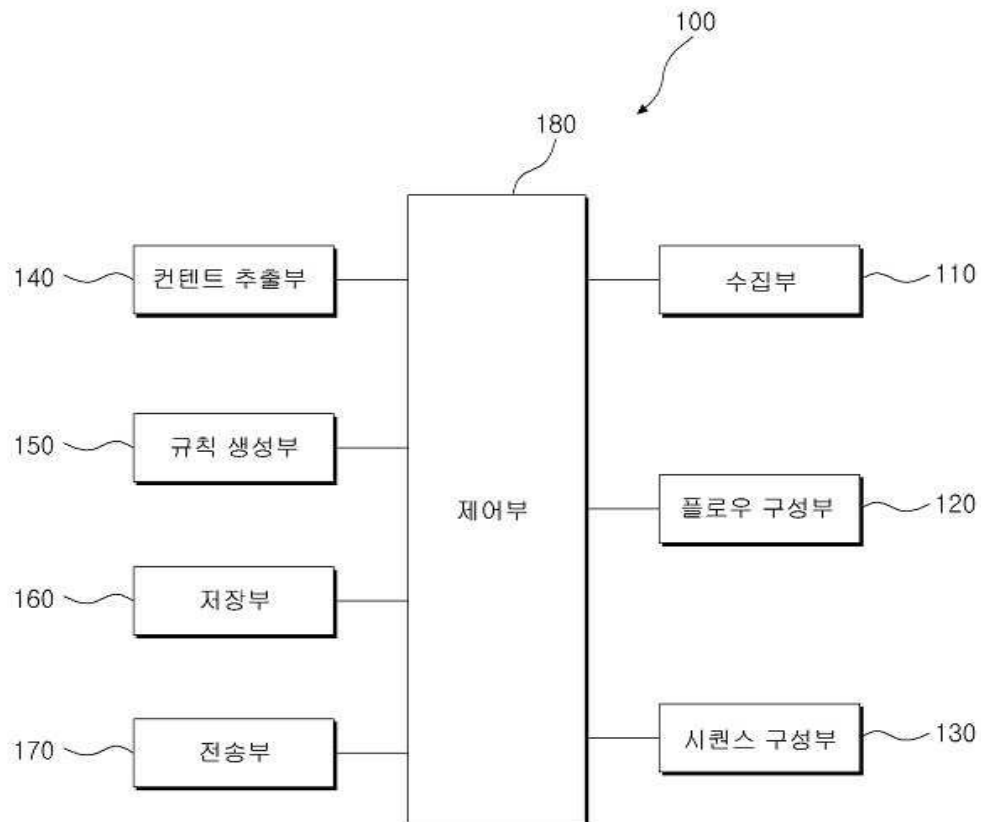
부호의 설명

[0089] 100: 규칙 자동 생성 장치

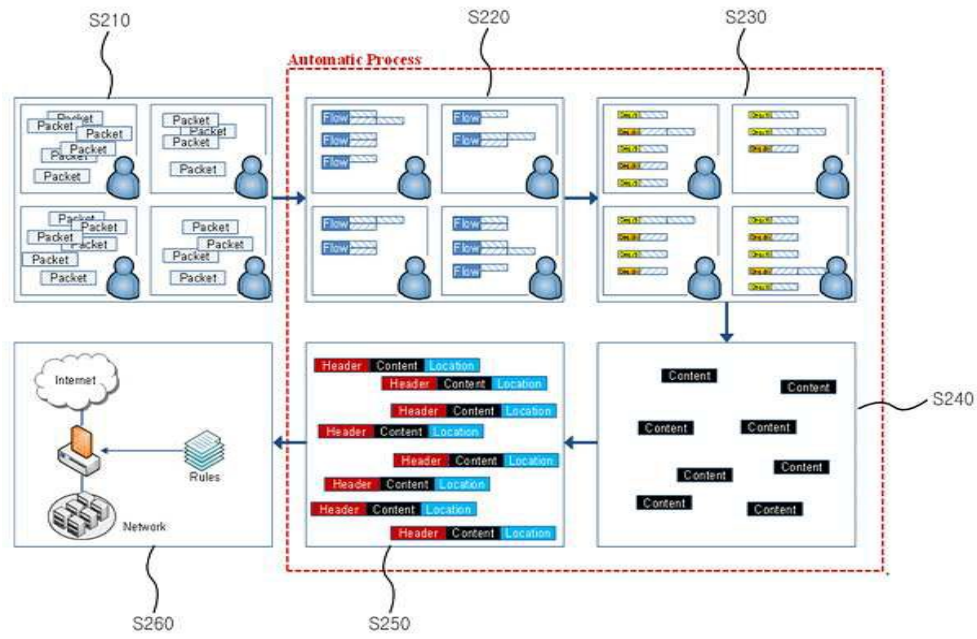
- 110: 수집부
- 120: 플로우 구성부
- 130: 시퀀스 구성부
- 140: 콘텐츠 추출부
- 150: 규칙 생성부
- 160: 저장부
- 170: 전송부
- 180: 제어부

도면

도면1



도면2



도면3

Input : $SequenceSet = \{S_1, S_2, \dots, S_s\}$, $minsupp$ ^o
Output : $ContentSet = \{C_1, C_2, \dots, C_c\}$ ^o

contentExtractor($SequenceSet, minsupp$)^o

```

1:  foreach sequence  $S$  in the  $SequenceSet$  doo
2:      foreach character  $a$  in the sequence  $S$  doo
3:           $L_1 = L_1 \cup a$ ;    //extract length-1 contentso
4:      endo
5:  endo
6:   $k = 2$ ;o
7:  while  $L_{k-1} \neq \emptyset$  doo
8:      foreach content  $c$  in the  $L_{k-1}$  do //delete under  $minsupp$ o
9:          for  $i = 1$  to  $s$  doo
10:             if ( $S_i$  include  $c$ ) theno
11:                  $count = count + 1$ ;o
12:             endo
13:          endo
14:          if ( $(count/s) < minsupp$ ) theno
15:               $L_{k-1} = L_{k-1} - c$ ;o
16:          endo
17:      endo
18:       $L_k = \text{candidate\_gen}(L_{k-1})$  //extract length- $k$  contentso
19:       $k++$ ;o
20:  endo
21:   $ContentSet = \cup_k L_k$ o
22:  deleteSubset( $ContentSet$ );o
23:  return  $ContentSet$ o

```

^oNotation: L_k : length k content set^o

도면4

Input : L_{k-1}
Output : L_k

candidate_gen(L_{k-1})

```

1:  foreach content  $p$  in  $L_{k-1}$  do
2:      foreach content  $q$  in  $L_{k-1}$  do
3:          if(( $p.a_2 = q.a_1$ ) && ( $p.a_3 = q.a_2$ ) ... && ( $p.a_{k-1} = q.a_{k-2}$ )) then
4:               $L_k = L_k \cup \langle p.a_1, p.a_2, \dots, p.a_{k-1}, q.a_{k-1} \rangle$ ;
5:          end
6:      end
7:  end
8:  return  $L_k$ 

```

*Notation: L_k : length k content set

도면5

SequenceSet = { S_1, S_2, S_3, S_4 }		$minsupp$
$S_1 = \{host_1, \langle \dots \dots CMD \dots \dots \rangle\}$		0.6
$S_2 = \{host_1, \langle \dots CD \dots \dots \rangle\}$		
$S_3 = \{host_2, \langle \dots \dots \dots CMD \dots \dots \rangle\}$		
$S_4 = \{host_3, \langle CM \dots \dots \rangle\}$		

L_1		L_2		L_3	
Length 1 content	Support	Length 2 content	Support	Length 3 content	Support
$\langle C \rangle$	1.00	$\langle CC \rangle$	0.00	$\langle CMD \rangle$	0.67
$\langle M \rangle$	1.00	$\langle CM \rangle$	1.00		
$\langle D \rangle$	0.67	$\langle CD \rangle$	0.34		
		$\langle MC \rangle$	0.00		
		$\langle MM \rangle$	0.00		
		$\langle MD \rangle$	0.67		
		$\langle DC \rangle$	0.00		
		$\langle DM \rangle$	0.00		
		$\langle DD \rangle$	0.00		

도면6

alert any any any → any any (content:"CMD":)
--

도면7

alert tcp any any → 111.222.333.0/24 80 (content:"CMD"; offset:4; depth:20;)

도면8

Input : content, packetSet = { $P_1, P_2, \dots, P_p$ }

Output : offset, depth

analysisContentLocation(content, packetSet)

- 1: offset = MAX_PACKET_SIZE;
- 2: depth = 0;
- 3: **foreach** packet t in packetSet **do**
- 4: **if**($t.isMatchContent(content)$) **then**
- 5: offset = $\min(offset, t.getStartMatch(content))$;
- 6: depth = $\max(depth, t.getEndMatch(content))$
- 7: **end**
- 8: **end**
- 9: **return** offset, depth;

*Notation: offset: min byte point beginning match, depth: max byte point ending match

도면9

Name	Rule Example
facebook	alert tcp 173.194.127.0/24 443 → any any (content:"00 c0=00 00(00 00 00 00 ff 01 00 01 00 00 0b 00 04 03 00 01 02 00=00 00;uP(00 00 00 10 00 0b 00 09 08;spdy:3.116 03 03 0e;N0b 00 0e;J00 0e"; offset: 43; depth: 100;);
youtube	alert tcp any any → any 80 (content:"GET/20/videoPlayback?c=web&cLen="; offset: 0; depth: 30;);
dropbox	alert tcp any any → 108.160.0.0/16 80 (content:"GET/20/subscribe?host_int=2942139721&ns_map=14199289_"; offset: 0; depth: 51;);
line	alert tcp any any > 23.76.153.88 80 (content:"GET/20/client/win/update_remote_info.xml/20/HTTP/1.1(0d 0a/Accept:(20/*/*(0d 0a/Accept-Encoding:(20/gzip,(20/deflate(0d 0a/User-A"; offset: 0; depth: 100;);
skype	alert tcp any any > 111.221.29.30 443 (content:"16 03 01 00u01 00 00q 03 01/T"; offset: 0; depth: 12;);
torrent	alert udp any any > any any (content:"d1.ad2.id20:"; offset: 0; depth: 12;);
typot	alert tcp any any > 110.45.243.174 80 (content:"GET/20/api/open/live/v1_0/Get_ads_info.xml?prgmid=11"; offset: 0; depth: 50;);