(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년10월07일
(11) 등록번호 10-2162991
(24) 등록일자 2020년09월28일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 12/24* (2006.01)
H04L 12/26 (2006.01) *H04L 12/815* (2013.01)
(52) CPC특허분류
H04L 63/20 (2013.01)
H04L 41/28 (2013.01)
(21) 출원번호 10-2019-0058274
(22) 출원일자 2019년05월17일
심사청구일자 2019년05월17일
(56) 선행기술조사문헌
KR1020050081881 A*
KR1020150062136 A*
KR1020180129376 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
배재대학교 산학협력단
대전광역시 서구 배재로 155-40 (도마동)
(72) 발명자
정희경
대전광역시 서구 둔산로 155, 112동 1303호(둔산동, 크로바아파트)
양승의
대전광역시 유성구 엑스포로 448, 305동 303호(전민동, 엑스포아파트)
(74) 대리인
유병욱, 한승범

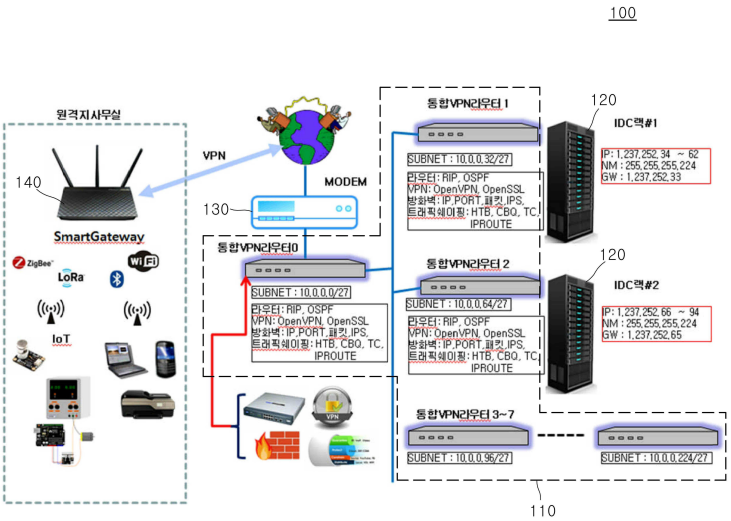
전체 청구항 수 : 총 9 항 심사관 : 박보미

(54) 발명의 명칭 **ＩＤＣ용 통합 보안 라우터 및 트래픽 쉐이핑과 ＩＰＳ의 융합 구현 기반의 통합 보안 서비스 방법**

(57) 요 약

본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터는 IDC(Internet Data Center) 서버의 펌웨어(Target Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정하는 룰셋 설정부; 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단하는 패턴 분석부; 및 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시키고, 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 허용을 하는 대신에 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리하는 의심 패킷 관리부를 포함한다.

대 표 도



(52) CPC특허분류
H04L 43/04 (2013.01)
H04L 43/0876 (2013.01)
H04L 47/22 (2013.01)
H04L 63/0263 (2013.01)
H04L 63/1408 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1425119765
부처명 중소벤처기업부
과제관리(전문)기관명 중소기업기술정보진흥원
연구사업명 산학연협력기술개발(R&D)
연구과제명 개방형 플랫폼 기반 데이터센터용 랙단위 통합 VPN 라우터 개발
기 여 율 1/1
과제수행기관명 행복을만드는집
연구기간 2018.06.01 ~ 2019.05.31

명 세 서

청구범위

청구항 1

IDC(Internet Data Center) 서버의 펌웨어(Target Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정하는 룰셋 설정부;

상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단하는 패턴 분석부; 및

상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시키고, 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 허용을 하는 대신에 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리하는 의심 패킷 관리부

를 포함하고,

상기 의심 패킷 관리부는

침입 방지 시스템에서 발생시킨 경보 메시지를 모니터링하여 상기 의심 패킷을 탐색하고, 상기 탐색된 의심 패킷을 대역폭이 제한된 큐에 넣은 후 해킹 체크 함수를 수행하며, 상기 해킹 체크 함수의 수행 결과에 따라, 상기 큐에 넣은 의심 패킷이 해킹 아님으로 판단되면 해당 패킷을 상기 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단되면 일정 시간 해당 소스의 모든 패킷을 드롭(Drop)시키며, 해킹인지 아닌지 판단 자체가 되지 않으면 상기 큐에 머물면서 후속 패킷에 대해 다시 판단을 위한 모니터링을 수행하는 것을 특징으로 하는 IDC용 통합 보안 라우터.

청구항 2

삭제

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 침입 방지 시스템은

소셜 엔지니어링(Social Engineering), 위장(Impersonation), 익스플로잇(Exploits), 전이적 신뢰(Transitive Trust), 데이터 드리븐(Data Driven), 기반 시설(Infrastructure), DoS(Denial of Service), 분산 DoS 중 적어도 하나를 포함하는 침입 수법에 대하여 IPS(Invasion Protection System) 기능을 지원하는 스노트(Snort)를 포함하는 것을 특징으로 하는 IDC용 통합 보안 라우터.

청구항 5

제1항에 있어서,

상기 의심 패킷 관리부는

다중 큐잉으로 구현된 VPN 터널링 구간의 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행하는 것을 특징으로 하는 IDC용 통합 보안 라우터.

청구항 6

제5항에 있어서,

상기 의심 패킷 관리부는

시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행하는 것을 특징으로 하는 IDC용 통합 보안 라우터.

청구항 7

제6항에 있어서,

상기 의심 패킷 관리부는

VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행하는 것을 특징으로 하는 IDC용 통합 보안 라우터.

청구항 8

IDC용 통합 보안 라우터의 룰셋 설정부가 IDC(Internet Data Center) 서버의 펌웨어(Target Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정하는 단계;

상기 IDC용 통합 보안 라우터의 패턴 분석부가 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단하는 단계;

상기 IDC용 통합 보안 라우터의 의심 패킷 관리부가 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시키는 단계;

상기 의심 패킷 관리부가 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 상기 의심 패킷에 대해 허용을 하는 대신에, 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리하는 단계;

상기 의심 패킷 관리부가 침입 방지 시스템에서 발생시킨 경보 메시지를 모니터링하여 상기 의심 패킷을 탐색하는 단계;

상기 의심 패킷 관리부가 상기 탐색된 의심 패킷을 대역폭이 제한된 큐에 넣은 후 해킹 체크 함수를 수행하는 단계; 및

상기 의심 패킷 관리부가 상기 해킹 체크 함수의 수행 결과에 따라, 상기 큐에 넣은 의심 패킷이 해킹 아님으로 판단되면 해당 패킷을 상기 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단되면 일정 시간 해당 소스의 모든 패킷을 드롭(Drop)시키며, 해킹인지 아닌지 판단 자체가 되지 않으면 상기 큐에 머물면서 후속 패킷에 대해 다시 판단을 위한 모니터링을 수행하는 단계

를 포함하는 것을 특징으로 하는 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법.

청구항 9

삭제

청구항 10

삭제

청구항 11

제8항에 있어서,

상기 트래픽 쉐이핑은

다중 큐잉으로 구현된 VPN 터널링 구간의 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 각 큐잉별 트래픽 부하량을 계산하는 단계; 및

상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 포함하여 실시간으로 수행되는 것을 특징으로 하는 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법.

청구항 12

제11항에 있어서,

상기 트래픽 쉐이핑은

시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하는 단계; 및

상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 더 포함하여 실시간으로 수행되는 것을 특징으로 하는 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법.

청구항 13

제12항에 있어서,

상기 트래픽 쉐이핑은

VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하는 단계; 및

상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계

를 더 포함하여 실시간으로 수행되는 것을 특징으로 하는 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은 IDC용 통합 보안 라우터 및 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법에 관한 것이다.

배 경 기 술

[0003] 전통적인 웹서버, 스트리밍, 인트라넷 그리고 최근의 빅데이터, 클라우드 등 많은 서비스가 서버 기반에 운영되고 있다. 또한, 사무실이 없는 온라인 비즈니스도 서버를 기반으로 개발, 영업, 기획, 마케팅, 판매 등이 이루어지고 있다. 이에 해킹에 안전하고 안정적인 서버운영이 사업성공의 필수적인 요소가 되고 있으며, 이를 지원해주는 기술서비스가 바로 인터넷 데이터 센터(IDC)이다.

[0004] IDC는 작은 공간에 많은 서버를 유치해야 하기 때문에 랙 단위로 라우터, 스위치 등 통신 장비와 20 여대의 슬림 서버를 슬롯 형태로 패키징 하여 운영한다. 이렇게 함으로써 좁은 공간에 효율적인 통신 인프라, 전원 관리 등을 제공하여 저렴한 비용에 고성능 서버를 운영할 수 있도록 한다.

[0005] 여기서 중요한 문제는 네트워크 단위 혹은 서버 단위, 좀 더 세부적인 서비스 단위의 해킹 및 공격을 모니터링하고 방어하는 서비스를 제공해야 하는데 있다. 또한, 특정 서버에 문제가 발생했을 경우 동일한 네트워크에 있는 다른 서버들에 피해를 주지 않도록 해야 한다.

[0006] 이에, 랙 단위의 소형 라우터를 기반으로 IPS 방화벽 그리고 네트워크별, IP별, 서비스 포트별로 세부적으로 트래픽을 제어할 수 있는 개방형플랫폼 기반 IDC용 랙 단위 통합 보안 라우터의 개발이 필요한 실정이다.

[0007] 관련 선행기술로는 한국공개특허 제10-2004-0039909호(발명의 명칭: 전송계층 터널링을 이용한 가상사설망에서의 통신품질향상방법, 공개일자: 2004.05.12.)가 있다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 일 실시예는 랙 단위의 소형 라우터를 기반으로 IPS 방화벽 그리고 네트워크별, IP별, 서비스 포트별로 세부적으로 트래픽을 제어함으로써 서비스 단위의 해킹 및 공격을 모니터링하고 방어하는 서비스를 제공하고 특정 서버에 문제가 발생했을 경우 동일한 네트워크에 있는 다른 서버들에 피해를 주지 않도록 할 수 있는 IDC용 통합 보안 라우터 및 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법을 제공한다.

[0011] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0013] 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터는 IDC(Internet Data Center) 서버의 펌웨어(Target Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정하는 룰셋 설정부; 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단하는 패턴 분석부; 및 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시키고, 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 허용을 하는 대신에 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리하는 의심 패킷 관리부를 포함한다.

[0014] 상기 의심 패킷 관리부는 침입 방지 시스템에서 발생시킨 경보 메시지를 모니터링하여 상기 의심 패킷을 탐색하고, 상기 탐색된 의심 패킷을 대역폭이 제한된 큐에 넣은 후 해킹 체크 함수를 수행할 수 있다.

[0015] 상기 의심 패킷 관리부는 상기 해킹 체크 함수의 수행 결과에 따라, 상기 큐에 넣은 의심 패킷이 해킹 아님으로 판단되면 해당 패킷을 상기 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단되면 일정 시간 해당 소스의 모든 패킷을 드롭(Drop)시킬 수 있다.

[0016] 상기 침입 방지 시스템은 소셜 엔지니어링(Social Engineering), 위장(Impersonation), 익스플로잇(Exploits), 전이적 신뢰(Transitive Trust), 데이터 드리븐(Data Driven), 기반 시설(Infrastructure), DoS(Denial of Service), 분산 DoS 중 적어도 하나를 포함하는 침입 수법에 대하여 IPS(Invasion Protection System) 기능을 지원하는 스노트(Snort)를 포함할 수 있다.

[0017] 상기 의심 패킷 관리부는 다중 큐잉으로 구현된 VPN 터널링 구간의 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0018] 상기 의심 패킷 관리부는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0019] 상기 의심 패킷 관리부는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0020] 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법은 IDC용 통합 보안 라우터의 룰셋 설정부가 IDC(Internet Data Center) 서버의 펌웨어(Target Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정하는 단계; 상기 IDC용 통합 보안 라우터의 패턴 분석부가 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단하는 단계; 상기 IDC용 통합 보안 라우터의 의심 패킷 관리부가 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시키는 단계; 및 상기 의심 패킷 관리부가 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 상기 의심 패킷에 대해 허용을 하는 대신에, 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리하는 단계를 포함한다.

[0021] 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법은 상기 의심 패킷 관리부가 침입 방지 시스템에서 발생시킨 경보 메시지를 모니터링하여 상기 의심 패킷을 탐색하는 단계; 및 상기 의심 패킷 관리부가 상기 탐색된 의심 패킷을 대역폭이 제한된 큐에 넣은 후 해킹 체크 함수를 수행하는 단계를 더 포함할 수 있다.

[0022] 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법은 상기 의심 패킷 관리부가 상기 해킹 체크 함수의 수행 결과에 따라, 상기 큐에 넣은 의심 패킷이 해킹 아님으로 판단되면 해당 패킷을 상기 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단되면 일정 시간 해당 소스의 모든 패킷을 드롭(Drop)시키는 단계를 더 포함할 수 있다.

[0023] 상기 트래픽 쉐이핑은 다중 큐잉으로 구현된 VPN 터널링 구간의 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 각 큐잉별 트래픽 부하량을 계산하는 단계; 및 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 포함하여 실시간으로 수행될 수 있다.

[0024] 상기 트래픽 쉐이핑은 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하는 단계; 및 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 더 포함하여 실시간으로 수행될 수 있다.

[0025] 상기 트래픽 쉐이핑은 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하는 단계; 및 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석하는 단계를 더 포함하여 실시간으로 수행될 수 있다.

[0027] 기타 실시예들의 구체적인 사항들은 상세한 설명 및 첨부 도면들에 포함되어 있다.

발명의 효과

[0029] 본 발명의 일 실시예에 따르면, 랙 단위의 소형 라우터를 기반으로 IPS 방화벽 그리고 네트워크별, IP별, 서비스 포트별로 세부적으로 트래픽을 제어함으로써 서비스 단위의 해킹 및 공격을 모니터링하고 방어하는 서비스를 제공하고 특정 서버에 문제가 발생했을 경우 동일한 네트워크에 있는 다른 서버들에 피해를 주지 않도록 할 수 있다.

도면의 간단한 설명

[0031] 도 1은 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 네트워크 구성을 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 상세 구성을 도시한 블록도이다.

도 3은 도 2의 의심 패킷 관리부의 상세 구성을 설명하기 위해 도시한 블록도이다.

도 4 내지 도 8은 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 개발 과정에 대해 설명하기 위해 도시한 도면이다.

도 9는 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 성능을 설명하기 위해 도시한 도면이다.

도 10은 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법을 설명하기 위해 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 본 발명의 이점 및/또는 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다.

[0033] 또한, 이하 실시되는 본 발명의 바람직한 실시예는 본 발명을 이루는 기술적 구성요소를 효율적으로 설명하기 위해 각각의 시스템 기능구성에 기 구비되어 있거나, 또는 본 발명이 속하는 기술분야에서 통상적으로 구비되는 시스템 기능 구성은 가능한 생략하고, 본 발명을 위해 추가적으로 구비되어야 하는 기능 구성을 위주로 설명한다. 만약 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면, 하기에 도시하지 않고 생략된 기능 구성 중에서 종래에 기 사용되고 있는 구성요소의 기능을 용이하게 이해할 수 있을 것이며, 또한 상기와 같이 생략된 구성 요소와 본 발명을 위해 추가된 구성 요소 사이의 관계도 명백하게 이해할 수 있을 것이다.

[0034] 또한, 이하의 설명에 있어서, 신호 또는 정보의 "전송", "통신", "송신", "수신" 기타 이와 유사한 의미의 용어는 일 구성요소에서 다른 구성요소로 신호 또는 정보가 직접 전달되는 것뿐만이 아니라 다른 구성요소를 거쳐 전달되는 것도 포함한다. 특히 신호 또는 정보를 일 구성요소로 "전송" 또는 "송신"한다는 것은 그 신호 또는 정보의 최종 목적지를 지시하는 것이고 직접적인 목적지를 의미하는 것이 아니다. 이는 신호 또는 정보의 "수신"에 있어서도 동일하다.

[0036] 이하에서는 첨부된 도면을 참조하여 본 발명의 실시예들을 상세히 설명하기로 한다.

[0037] 도 1은 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 네트워크 구성(100)을 도시한 도면이다.

[0038] 도 1을 참조하면, 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터(110)는 물리적으로 떨어져 있는 원격지의 네트워크(사무실)를 가상의 네트워크를 이용하여 마치 하나의 네트워크처럼 보이게 구성하는 VPN 시스템에 구현하여 VPN 터널링 시 최적의 터널링 속도를 유지하여 우수한 인터넷 회선 품질을 제공할 수 있다.

[0039] 여기서, 상기 VPN 시스템은 상기 IDC용 통합 보안 라우터(110)에 연결된 모뎀(130)과 원격지 사무실에 위치한 스마트 게이트웨이(140) 사이에 구현될 수 있다. 여기서, 상기 스마트 게이트웨이(140)는 원격지 사무실에 배치된 IoT 디바이스들과 연결되어 IoT 서비스를 제공할 수 있다.

[0040] 상기 VPN 시스템은 OpenVPN Client Gateway 제어 시스템(미도시)를 통해 OpenVPN Client Gateway(미도시)의 동작을 제어할 수 있다. 상기 스마트 게이트웨이(140)는 상기 OpenVPN Client Gateway 및 상기 OpenVPN Client Gateway 제어 시스템을 포함하여 구현될 수 있다. 여기서, 상기 OpenVPN Client Gateway 제어 시스템은 상기 OpenVPN Client Gateway를 포함하여 구현될 수도 있고, 상기 OpenVPN Client Gateway와는 별개로 구현될 수도 있다. 상기 OpenVPN Client Gateway가 유무선 공유기로 구현되는 경우, 상기 OpenVPN Client Gateway 제어 시스템은 응용 프로그램 형태로 유무선 공유기에 탑재되어 일체로 형성될 수도 있다.

[0041] 상기 IDC용 통합 보안 라우터(110)는 랙(120) 단위를 기반으로 IPS 방화벽 그리고 네트워크별, IP별, 서비스 포트별로 세부적으로 트래픽을 제어할 수 있다. 이와 같은 IDC용 통합 보안 라우터(110)는 랙(120) 단위 통합 네트워크 장비로서 라우터, VPN, 방화벽은 물론 트래픽 쉐이핑과 IPS까지 모두 지원하는 장비로 구현될 수 있다.

[0042] 상기와 같이 네트워크 장비를 통합해서 얻을 수 있는 가장 큰 장점은 경제성이다. 5개의 기능을 하나로 만들면 비용을 1/5 정도로 줄일 수 있다. 또한, 각 장비에서는 목적에 따라 네트워크 패킷을 레이어 2~5까지 필요한 레벨로 분리해서 분석하고 재조합하기 때문에 5개의 장비에서 중복 분석하는 것보다 하나의 장비에서 하는 것이 효율적이다.

[0043] 한편, IDC 센터를 운영하기 위해서 가장 필요한 기술 중 하나가 바로 트래픽 쉐이핑 기술이다. 다양한 서비스를 하는 고객들의 서버 시스템을 IDC에 집중해 놓고 운영하다 보면 서버 문제, 해킹 문제, 트래픽 문제, 바이러스 문제 등이 다양하게 일어날 수 있다. 이는 문제를 유발시킨 서버는 물론 같은 네트워크에서 운영되는 다른

서버들에게까지 피해를 주게 되어 심각한 서비스 품질 저하를 일으킬 수 있다. 이에 본 발명의 일 실시예에서는 실시간 트래픽 쉐이핑 기능을 통해 네트워크 트래픽 상황을 네트워크별, IP별, 서비스 포트별로 INBOUND, OUTBOUND 모두 모니터링 하고 대역폭을 조절할 수 있게 하여 트래픽 문제가 발생하더라도 전체적인 서비스 품질은 보장하도록 할 수 있다.

[0044] 또한, IDC 센터를 운영하기 위해서는 라우터 및 IPS 방화벽이 필요하다. 랙 단위로 라우터를 구성하여 다양한 고객의 다양한 서비스(100M 공유, 10M 전용 등) 요구에 맞추어 네트워크를 분리 운영하여 효율적이고 경제적인 IDC 운영이 되도록 할 수 있다. 또한 네트워크 침입 시도에 대해서는 패킷 분석을 통해 라우터 단에서 패킷을 차단할 수 있도록 하여 서버의 피해를 줄일 수 있게 한다.

[0045] 또한, IDC 센터를 운영하기 위해서는 VPN이 필요하다. 원격지 근무나 본사와 지사 간 안전한 네트워크 공유를 위한 VPN 서비스 요구는 꾸준히 커지고 있다. 또한, 사물인터넷의 확산과 함께 기업내부의 센서 및 제어장치들을 안전하게 모니터링 하고 접근할 수 있는 서비스의 요구가 많아지고 있다. 이에 대한 서비스를 네트워크 VPN으로 제공할 수 있다.

[0046] 또한, IDC 센터를 운영하기 위해서는 OpenWRT 플랫폼이 필요하다. 이러한 통합 네트워크 장비를 개발하기 위해서는 하드웨어 및 소프트웨어 플랫폼이 중요하다. 지원 타겟 하드웨어가 다양하여 특정 타겟에 종속되지 말아야 하고, RIP, OSPF, OpenVPN, SSL, 다중 큐잉, IPS 등 관련 커널 및 프로토콜 스택의 구현에 문제가 없어야 한다. 이를 만족시키는 우수한 플랫폼 중 하나가 OpenWRT이다.

[0048] 도 2는 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 상세 구성을 도시한 블록도이다.

[0049] 도 2를 참조하면, 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터(110)는 룰셋 설정부(210), 패턴 분석부(220), 의심 패킷 관리부(230), 및 제어부(240)를 포함하여 구성될 수 있다.

[0050] 상기 룰셋 설정부(210)는 IDC 서버의 펌웨어(Firmware) 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋(Rule-Set)을 설정할 수 있다.

[0051] 상기 패턴 분석부(220)는 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초하여 패턴 분석을 실시하고, 상기 패턴 분석을 통해 상기 데이터 트래픽이 침입 공격 패킷의 트래픽인지 또는 의심 패킷의 트래픽인지 여부를 판단할 수 있다.

[0052] 즉, 상기 패턴 분석부(220)는 상기 데이터 트래픽을 상기 룰셋에 포함된 복수의 룰과 비교하여 그 패턴의 유사성 또는 동일성 여부를 분석하고, 그 패턴 분석의 결과 동일하거나 유사하면 상기 데이터 트래픽을 침입 공격 패킷의 트래픽으로 판단할 수 있다. 반면, 상기 패턴 분석부(220)는 그 패턴 분석의 결과 동일하거나 유사하지 않으면 상기 데이터 트래픽을 의심 패킷의 트래픽으로 판단할 수 있다.

[0053] 상기 의심 패킷 관리부(230)는 상기 패턴 분석부(220)의 판단 결과에 따라 상기 데이터 트래픽을 차단하거나 허용할 수 있다. 즉, 상기 의심 패킷 관리부(230)는 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우에는 상기 침입 공격 패킷을 즉시 차단시킬 수 있다.

[0054] 반면, 상기 의심 패킷 관리부(230)는 상기 데이터 트래픽이 상기 의심 패킷의 트래픽인 것으로 판단된 경우에는 상기 의심 패킷에 대해 허용을 할 수 있다. 단, 상기 의심 패킷에 대해 허용을 하는 경우에 한해서, 상기 의심 패킷 관리부(230)는 트래픽 쉐이핑(Traffic Shaping)을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리할 수 있다.

[0055] 상기 의심 패킷 관리부(230)는 침입 방지 시스템에서 발생시킨 경보 메시지를 모니터링하여 상기 의심 패킷을 탐색하고, 상기 탐색된 의심 패킷을 대역폭이 제한된 큐에 넣은 후 해킹 체크 함수를 수행할 수 있다. 상기 의심 패킷 관리부(230)는 상기 해킹 체크 함수의 수행 결과에 따라, 상기 큐에 넣은 의심 패킷이 해킹 아님으로 판단되면 해당 패킷을 상기 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단되면 일정 시간 해당 소스의 모든 패킷을 드롭(Drop)시킬 수 있다.

[0056] 여기서, 상기 침입 방지 시스템은 소셜 엔지니어링(Social Engineering), 위장(Impersonation), 익스플로잇(Exploits), 전이적 신뢰(Transitive Trust), 데이터 드리븐(Data Driven), 기반 시설(Infrastructure), DoS(Denial of Service), 분산 DoS 중 적어도 하나를 포함하는 침입 수법에 대하여 IPS(Invasion Protection System) 기능을 지원하는 스노트(Snort)를 포함할 수 있다.

[0057] 상기 의심 패킷 관리부(230)는 다중 큐잉으로 구현된 VPN 터널링 구간의 각 큐잉별 드롭(drop)율 및 상기 각 큐

잉별로 할당된 밴드폭에 기초하여 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0058] 또는, 상기 의심 패킷 관리부(230)는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0059] 또는, 상기 의심 패킷 관리부(230)는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석함으로써 상기 트래픽 쉐이핑을 실시간으로 수행할 수 있다.

[0060] 상기 제어부(240)는 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터(110), 즉 상기 룰셋 설정부(210), 상기 패턴 분석부(220), 상기 의심 패킷 관리부(230) 등의 동작을 전반적으로 제어할 수 있다.

[0062] 도 3은 도 2의 의심 패킷 관리부(230)의 상세 구성을 설명하기 위해 도시한 블록도이다.

[0063] 도 3을 참조하면, 상기 의심 패킷 관리부(230)는 다중 큐(310), 큐 제어부(320), 패킷 할당부(330), 및 모니터링부(340)를 포함하여 구성될 수 있다.

[0064] 상기 다중 큐(310)는 원격지의 네트워크를 가상 사설망(VPN)을 통해 연결하기 위한 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 잠시 버퍼링하였다가 사전에 정의된 우선순위에 따라 출력할 수 있다. 다시 말해, 상기 다중 큐(310)는 패킷을 잠시 버퍼링하였다가 사전에 정의된 트래픽 제어 룰에 따라 패킷을 출력하는 저장소 역할을 한다.

[0065] 상기 다중 큐(310)는 실제 패킷을 저장하는 메모리 공간이 아니라, 패킷이 대기하고 있는 메모리의 위치를 알려주는 포인터 정보를 연속적으로 가지고 있는 것으로 볼 수 있다. 큐잉(queuing)은 이러한 다중 큐(310)를 이용하여 아웃바운드 또는 인바운드 인터페이스로 내보내기 위해 대기하고 있는 패킷들의 포인터 주소를 기 정의된 스케줄링에 따라 나열하는 작업을 말한다.

[0066] 상기 다중 큐(310)는 ToS(Type of Service)와 우선순위에 따라 패킷을 출력하는 SFQ(Stochastic Fair Queuing) 로직을 갖는 큐, 및 대역폭의 조절을 통해 패킷의 유효 속도 또는 최대 속도를 제한하는 HTB(Hierarchical Token Bucket) 로직을 갖는 큐를 포함할 수 있다.

[0067] 상기 다중 큐(310)는 상기 SFQ 로직 또는 상기 HTB 로직을 기반으로 상기 VPN 터널링 구간에서 인바운드 또는 아웃바운드하는 패킷을 다중 큐잉(Multi-Queuing)할 수 있다.

[0068] 상기 큐 제어부(320)는 상기 다중 큐(310)의 개수, 우선순위 및 대역폭 중 적어도 하나를 조절 또는 변경할 수 있다. 이를 위해, 상기 큐 제어부(320)는 상기 SFQ 로직 및/또는 상기 HTB 로직을 이용하여 상기 다중 큐(310)를 제어할 수 있다.

[0069] 즉, 상기 큐 제어부(320)는 네트워크 상태 또는 시스템 자원의 점유율에 따라 상기 다중 큐(310)의 개수, 우선순위, 대역폭 등을 제어할 수 있다. 여기서, 상기 다중 큐(310)의 개수는 많을수록 패킷 전송률이 우수하고 패킷 드롭(drop)율이 낮은 장점이 있지만, 버퍼링의 딜레이가 커지고 이는 곧 시스템 부하가 높아지게 되는 단점도 있게 된다. 그러므로, 상기 큐 제어부(320)는 네트워크 상태 또는 시스템 자원의 점유율에 따라 최적의 다중 큐잉이 가능하도록 제어할 수 있다.

[0070] 상기 큐 제어부(320)는 서비스 포트별, IP 주소별로 상기 다중 큐(310)를 개별 큐로 구분할 수 있으며, 상기 구분된 개별 큐의 우선순위 또는 대역폭을 조절할 수 있다. 여기서, 상기 서비스 포트는 ftp, htb, ssh, telnet, www, tcmp 프로토콜을 통해 접속되는 포트를 포함할 수 있다.

[0071] 상기와 같은 큐 제어부(320)의 동작을 통해, 상기 다중 큐(310)에는 ToS, QoS에 따른 우선순위가 지정될 수 있지만, IP 주소별, 서비스 포트별로 배치시킨 각 큐에 대해서도 우선순위가 할당되고 각 큐의 대역폭도 네트워크 상태, 시스템 자원 점유율 등에 따라 설정 및 변경될 수 있다.

[0072] 상기 패킷 할당부(330)는 후술하는 모니터링부(340)에 의한 네트워크 상태의 분석 결과에 기초하여 상기 인바운드 또는 아웃바운드하는 패킷을 상기 다중 큐(310)에 적응적으로 할당할 수 있으며, 이를 통해 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0073] 또한, 상기 패킷 할당부(330)는 상기 패킷을 서비스 포트별, IP 주소별 등 세부적으로 구분하여 할당할 수 있다. 즉, 상기 패킷 할당부(330)는 상기 패킷을 서비스 포트별, IP 주소별로 구분하고, 상기 구분된 패킷을 상기 네트워크 상태의 분석 결과에 기초하여 상기 다중 큐에 적응적으로 할당할 수 있다.

[0074] 이때, 상기 VPN 터널링 구간에 걸리는 네트워크 상태는 모니터링부(340)를 통해 분석될 수 있으며, 상기 패킷 할당부(330)는 상기 모니터링부(340)를 통해 분석된 네트워크 상태에 따라 최적화된 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘에 따라 패킷을 할당할 수 있다.

[0075] 상기 패킷 할당부(330)는 상기 트래픽 쉐이핑 알고리즘의 선택을 패킷 할당이 필요할 때마다 수행할 수 있지만, 패킷 할당 시점과는 상관없이 상기 모니터링부(340)에서 상기 네트워크 상태를 분석하는 일정 주기마다 수행하고 주기 동안에는 상기 선택된 트래픽 쉐이핑 알고리즘으로 동작하도록 할 수 있다.

[0076] 상기 트래픽 쉐이핑 알고리즘은 다중 큐잉에 따라 다양한 쉐이핑이 가능하다. 상기 트래픽 쉐이핑 알고리즘은 IDC(Internet Data Center)에서 사용하는 수준의 인바운드, 아웃바운드 트래픽 제어는 물론, 각 서비스 포트별, IP 주소별, IP 네트워크별, 또는 이더 패킷(ether packet) 수준의 트래픽 제어까지 구현할 수 있다. 또한, 상기 트래픽 쉐이핑 알고리즘은 패킷 큐잉을 통해서 TAP(Test Access Port) 장비에서 사용하는 패킷 인스펙션까지 구현할 수 있기 때문에 점점 지능화 되어가고 있는 해킹 공격에 대해서도 트래픽 제어가 가능하다.

[0077] 상기 모니터링부(340)는 상기 VPN 터널링 구간에 걸리는 트래픽 부하 및 상기 다중 큐의 각 큐잉별 트래픽 부하를 포함하는 네트워크 상태를 모니터링하고 분석할 수 있다.

[0078] 이를 위해, 일 실시예로서, 상기 모니터링부(340)는 시스템 전체에 대한 CPU 사용 시간과 VPN 터널링 데몬의 사용 시간을 바탕으로 상기 VPN 터널링 데몬의 부하량을 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산하고, 상기 CPU의 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0079] 이때, 상기 모니터링부(340)는 하기 수학식 1에 기초하여, 상기 시스템 전체에 대한 CPU 사용 시간과 상기 VPN 터널링 데몬의 사용 시간을 일정시간(1초~10초) 간격으로 계산하여 상기 VPN 터널링에 사용되는 CPU의 부하량을 계산할 수 있다.

[0080] [수학식 1]

[0081]
$$user\_util = 100 * (utime\_after - utime\_before) / (time\_total\_after - time\_total\_before);$$
$$sys\_util = 100 * (stime\_after - stime\_before) / (time\_total\_after - time\_total\_before);$$

[0082] 여기서, utime은 유저 모드 지피스(user mode jiffies), utime_before 및 utime_after는 유저 모드 지피스가 발생하는 전/후 시간, user_util은 시스템 전체에 대한 CPU 사용 시간이고, stime은 커널 모드 지피스(kernel mode jiffies), stime_before 및 stime_after는 커널 모드 지피스가 발생하는 전/후 시간, sys_util은 커널 모드 사용 시간임.

[0083] 이에 따라, 상기 패킷 할당부(330)는 상기 CPU의 부하량에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0084] 다른 실시예로서, 상기 모니터링부(340)는 VPN 터널링 디바이스의 패킷 정보에 기초하여 전체 패킷 드롭(drop)율을 계산하고, 상기 전체 패킷 드롭율에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0085] 이때, 상기 모니터링부(340)는 하기 수학식 2에 기초하여, 상기 VPN 터널링 디바이스의 패킷 정보를 일정시간(1초~10초) 간격으로 계산하여 상기 전체 패킷 드롭율을 계산할 수 있다.

[0086] [수학식 2]

[0087]
$$rx\_drop\_rate=100*(rx\_drop\_after - rx\_drop\_before)/(rx\_pkt\_tot\_after - rx\_pkt\_tot\_before);$$
$$tx\_drop\_rate=100*(tx\_drop\_after - tx\_drop\_before)/(tx\_pkt\_tot\_after - tx\_pkt\_tot\_before);$$

[0088] 여기서, rx_drop은 상기 VPN 터널링 디바이스의 송신 측에서 드롭된 패킷 수, tx_drop은 상기 VPN 터널링 디바이스의 수신 측에서 드롭된 패킷 수, rx_pkt_tot는 상기 VPN 터널링 디바이스의 송신 측 패킷 수, tx_pkt_tot는

상기 VPN 터널링 디바이스의 수신 측 패킷 수임.

[0089] 이에 따라, 상기 패킷 할당부(330)는 상기 전체 패킷 드롭율에 기초하여 해당 네트워크 상태에 맞는 트래픽 쉐이핑 알고리즘을 선택하고, 상기 선택된 트래픽 쉐이핑 알고리즘을 이용하여 상기 패킷을 상기 다중 큐에 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0090] 또 다른 실시예로서, 상기 모니터링부(340)는 상기 각 큐잉별 드롭(drop)율 및 상기 각 큐잉별로 할당된 밴드폭에 기초하여 상기 각 큐잉별 트래픽 부하량을 계산하고, 상기 각 큐잉별 트래픽 부하량에 기초하여 상기 VPN 터널링 구간에 걸리는 트래픽 부하를 모니터링하고 분석할 수 있다.

[0091] 이에 따라, 상기 패킷 할당부(330)는 상기 각 큐잉별 트래픽 부하량에 기초하여 다중 큐잉 중 부하가 상대적으로 적은 큐잉을 찾아서 밴드폭을 줄이고, 상대적으로 부하가 큰 다른 큐잉에 밴드폭을 더 할당하여 상기 VPN 터널링 구간의 트래픽을 제어할 수 있다.

[0092] 즉, 상기 패킷 할당부(330)는 상기 모니터링부(340)를 통해 다중 큐잉에서 큐잉별로 패킷량, drop량이 계산되면, 그 계산 결과를 토대로 하여 트래픽 유발 큐잉을 찾아 해당 트래픽에 대하여 우선순위, 속도를 최적화할 수 있도록 제어할 수 있다.

[0093] 앞서 수학식 1, 2를 통해 설명한 CPU 부하량과 패킷 drop율은 리눅스 커널에서 실시간 시스템 정보를 가지고 계산하는 것이다. 즉, 추가 부하 없이 네트워크 트래픽 부하를 계산해내는 것이다.

[0094] 여기서, 패킷 drop율은 전체 모든 종류의 패킷에 대한 drop을 보는 것이다. 이때 문제점은 네트워크 패킷이 과부하 상태라 drop 하기는 하는데 어떤 서비스 때문에 drop을 하는지 모르는데 있다.

[0095] 이를 해결하고자 네트워크 트래픽을 세부적인 서비스별로 각각 다중 큐잉을 하여 각 큐잉별, 즉 서비스별로 패킷의 drop을 파악하여 어떤 서비스가 지정된 큐잉 폭보다 더 부하가 큰가를 파악하는 것이다. 즉, 큐잉별 패킷 drop을 계산하고 할당된 큐잉 밴드폭을 계산하면 각 서비스별로 세부적으로 부하량을 계산할 수 있다.

[0096] 이러한 계산은 큐잉별 실시간에 가능하고 이를 바탕으로 여러 큐잉 중 부하가 적은 큐잉을 찾아서 밴드폭을 줄이고, 이렇게 줄인 만큼 부하가 큰 큐잉에 밴드폭을 더 할당해 줄 수 있도록 하는 방식이다.

[0097] 이 방법은 전체 패킷 drop율만 가지고 네트워크 부하를 측정하는 경우 추가로 부하가 걸리지 않는 것과 동일한 장점을 그대로 가지고 있고, 여기에 어떤 종류의 서비스 패킷이 원인인가까지 찾아낼 수 있는 방법이다.

[0098] 따라서, 상기의 방법에 의하면 실시간에 다중 큐잉의 내부 트래픽을 세부적으로 계산하여 다중 큐잉 밴드폭을 효율적으로 실시간에 최적의 상태로 변화시켜 줄 수 있다. 즉, 미리 설계된 다중 큐잉 시나리오 없이 최적의 상태로 운영이 가능하다. 다시 말해, 미리 정해진 큐잉 알고리즘을 단순히 적용하여 발생하는 속도 제어 오류를 방지하고 실시간 해당되는 큐잉 및 관련 큐잉에 대해서 세부적으로 제어할 수 있다.

[0100] 이하에서는 도 4 내지 도 8을 참조하여 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 개발 과정에 대해 서술한다.

[0101] 1. OpenWRT 플랫폼 구축

[0102] 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터는 VPN과 라우터 프로토콜은 물론 IPS, 트래픽 쉐이핑 등 핵심 프로토콜들이 모두 구현되어야 한다. 자칫 메모리, CPU 부하가 커질 수 있기 때문에 효율적인 커널 튜닝이 필요하고 CPU 및 메모리 성능도 만족할 만한 타겟을 선정해야 한다.

[0103] 2. OpenWRT 타겟 하드웨어 플랫폼 구현

[0104] 게이트웨이를 구현하기 위한 타겟 하드웨어는 양산비용 측면에서 다양한 선택이 가능하다. 본 발명의 일 실시예에서는 수급이 원활하며 가성비가 우수한 타겟 시스템을 선택하였다. 소형 통합 VPN 라우터로 양산비용이 저렴하고 성능은 snort 및 트래픽 쉐이핑까지 가능하도록 구현하였다. 타겟 시스템에 대한 하드웨어/소프트웨어 스펙은 도 4에 도시된 바와 같다.

[0105] 3. OpenWRT 타겟 소프트웨어 플랫폼 구현

[0106] Linux에서 버전의 중요성은 커널 및 지원 디바이스 그리고 관련 프로토콜의 성능과 호환성 그리고 안정성 보장 등에 있다. OpenWRT도 리눅스 배포본 중에 한 종류이며 특히 네트워크 장비의 임베디드 플랫폼으로 유명한 것 중 하나이다. 데스크탑이나 서버용 Linux의 업그레이드도 큰 노력이 수반되지만 임베디드 시스템에서의 버전 업그레이드는 보안, 성능, 안정성은 물론 새로운 프로토콜 지원, 새로운 하드웨어 타겟 플랫폼 및 모듈 지원 등

그 중요성은 일반 Linux의 경우보다 훨씬 크다.

[0107] 본 발명의 일 실시예에서 사용한 최신의 Chaos Calmer 15.05의 특징은 커널 업그레이드, 시스템 안정성 향상, 버그 패치, 새로운 하드웨어 모듈 지원 등이 있고, 반면에 많은 프로토콜 및 패키지들이 업그레이드 되었기 때문에 하위 호환에 주의를 해야 한다. 본 발명의 일 실시예에 필요한 RIP, OSPF, VPN, IPS 등의 프로토콜은 이전 버전부터도 안정적인 동작이 가능하기 때문에 큰 문제는 없었다. 다만 실시간 트래픽 쉐이핑을 위한 커널 튜닝과 타겟 플랫폼의 세밀한 설정과 컴파일은 여전히 중요한 부분이라 할 수 있다.

[0108] 4. 관련 프로토콜 스택 구현

[0109] 본 발명의 일 실시예에서 구현된 프로토콜 기술로서 필수 프로토콜을 기술하면 다음과 같다.

[0110] - VPN 터널링은 OpenVPN 기반에 네트워크 계층3에서 구현되는 OpenSSL로 구현하였다.

[0111] - 다중큐잉을 통한 트래픽쉐이핑

[0112] ToS, QoS를 단일 큐잉에서 구현하는 기존의 네트워크 장비들과 달리 본 발명의 일 실시예에서는 다중 큐잉 지원 커널을 지원하며, IPS와 융합 구현을 할 것이다.

[0113] 다중 큐잉을 지원하기 위해서는 OS의 네트워크 커널에서 CBQ, SFQ, HTB 등을 지원하도록 커널 수정을 해야 하기 때문에 안정성 및 성능을 보장하기 위해서는 적절한 튜닝 기술을 적용해야 한다. 본 발명의 일 실시예에서는 다중 큐잉과 터널링을 지원하도록 커널을 수정하고 프로그램을 구현하였다. 또한, 도 5에서 보듯이 서로 다른 특성의 CBQ, SFQ, HTB 큐를 구성하고 각 큐에 서비스 포트별, IP 주소별로 할당하여 동작할 수 있도록 하였다.

[0114] 5. RIP, OSPF 구현

[0115] 도 6은 라우팅 프로토콜에 대해 보여주고 있다. 도 6을 참조하면, 대표적인 라우팅 프로토콜은 RIP(Routing Information Protocol)와 OSPF(Open Shortest Path First)가 있다. RIP와 OSPF는 대표적인 IGP(Interior Gateway Protocol)로서 하나의 AS(Autonomous System) 내의 라우터들끼리 라우팅 정보 교환을 위한 라우팅 프로토콜이다. 또한 AS들 간의 라우팅 정보 교환을 위해서는 EGP(Exterior Gateway Protocol)가 있고 EGP에는 BGP(Border Gateway Protocol)가 대표적인 프로토콜이다.

[0116] 도 7은 zebra의 실행 구조를 보여주고 있다. 도 7을 참조하면, OpenWRT에서 RIP, OSPF를 구현하기 위해서는 라우팅 매니저인 zebra 패키지를 준비해야 한다. zebra 패키지를 다운로드하여 설치하면 ripd(RIPv1,2), ospfd(ospfv2), bgpd, ripngd(IPv6), ospfd6d(IPv6) 라우팅 프로토콜을 지원하는 데몬(daemon)들이 설치된다. 이 데몬들은 반드시 zebra와 연동되어야 하며, 단독으로 실행될 수 없다. zebra의 중요한 특징 중의 하나는 CLI(Command Line Interface)를 지원하는 것인데 이 CLI는 VTYSH라는 쉘(Shell)을 이용함으로써 사용 가능하다. VTYSH는 putty를 이용하여 연결하게 되며 연결 하고자 하는 포트번호는 사용자가 /etc/service 파일을 수정함으로써 접근 가능하다. VTYSH를 이용하면 시스코라우터 설정과 동일하게 외부에서 putty을 이용하여 ripd, ospfd, zebra의 환경 설정이 가능하다.

[0117] zebra는 현재 quagga로 버전이 업그레이드 되었고, Network->Routing and Redirection->quagga-> 로 이동하여 quagga-ospfd, ripd, vtysh, zebra 등을 선택하여 커널 설정이 가능하다. 커널 컴파일이 성공적이라면 타겟 펌웨어를 업그레이드하고 구성하려는 라우팅 구조에 맞게 zebra.conf, ripd.conf, ospf.conf 환경 설정 파일을 작성하면 최대 15 HOP까지 라우팅을 구성할 수 있다.

[0118] 6. IPS 구현

[0119] IPS는 방화벽에 이은 진화한 보안 솔루션이다. 네트워크 공격에 대한 방화벽의 차단이 실패하였을 경우에도 피해를 최소화하고 관리자의 부재 시에도 해킹에 적절히 대응해 주는 보안 솔루션이다. 즉, IPS는 방화벽이 단순한 룰에 따라 불법 침입을 차단하는데 따른 보안상의 한계점을 극복할 수 있게 한다.

[0120] IPS에는 크게 2가지 방법이 있다. 첫 번째, 오용침입탐지라고 불리는 Misuse IPS는 기본적으로 일어나서는 안 될 행위 패턴을 설정하고 패턴과 일치하는가를 확인하는 방법이다. 두 번째, 일어날 수 있는 극히 정상적인 사용에 대해서 사용자나 그룹, 프로그램, 시스템 리소스에서 비정상적인 행위가 일어나는 지를 탐지하는 Anomaly IPS 방법이 있다. 정확한 동작을 위해서는 다양한 침입패턴, 오용패턴코드를 적용하여 실시간 공격형태분석과 실제로 침입탐지와 오용탐지까지 찾아내는 것이 중요하다. 여기서 생각해봐야 하는 중요한 트레이드 오프는 바로 오탐에 대한 다음 2가지 시각이다.

[0121] false positive는 실제 공격이 아닌데 IPS가 공격으로 잘못 판단하는 경우를 가리키고, false negative는 실제 공격이 일어났지만 IPS가 이를 공격으로 생각하지 않는 경우를 가리키는데, 이러한 2가지 오탐은 모두 문제가 있다. false positive의 경우 정상적인 사용자까지 불편을 야기하고, false negative의 경우 실제 공격에 대응을 못하는 문제를 가지고 있다. 여기서 어떤 방법을 선택해야 할까 하는 것은 어려운 고민거리다. 이에 대한 해결 방안은 아래의 트래픽 쉐이핑과 IPS의 융합 구현에 관한 실시예에서 자세히 설명한다.

[0122] 침입을 탐지하기 위해서는 나의 취약점과 침입 유형을 파악하고 있어야 한다. 침입이란 일반적으로 컴퓨터가 사용하는 자원의 무결성, 기밀성, 유효성을 저해하는 일련의 행위나 컴퓨터 시스템의 보안 정책을 파괴하는 행위를 침입이라고 정의한다. 도 8은 일반적인 침입 수법의 분류를 나타낸다. 도 8을 참조하면, 상기 침입 수법은 소셜 엔지니어링(Social Engineering), 위장(Impersonation), 익스플로잇(Exploits), 전이적 신뢰(Transitive Trust), 데이터 드리븐(Data Driven), 기반 시설(Infrastructure), DoS(Denial of Service), 분산 DoS 등을 포함할 수 있다.

[0123] OpenWRT에서 지원하는 대표적인 IPS는 snort이다. snort는 위에서 언급한 침입 수법에 대한 IPS 기능을 지원한다. 구현은 Network -> Firewall -> snort, snort-mysql 등을 선택하여 커널 설정이 가능하다. 커널 컴파일이 성공적이라면 타겟 펌웨어를 업그레이드하고 snort.conf, local.rules 등 환경 설정 파일에 IPS 관련 로그, 패턴, 룰셋 등을 설정하면 원하는 동작을 구성할 수 있다.

[0125] 트래픽 쉐이핑과 IPS의 융합 구현에 관한 실시예

[0127] Snort의 기능은 막강하다. 유사한 솔루션 중 snort가 제일 오래되고 다양한 플러그인이 가능하기 때문에 IDS, IPS 보안솔루션과 접목하여 지능형 방화벽 개발이 가능하다. 물론 여기서 가장 중요한 부분은 룰셋이다. 해킹 및 공격의 방법은 나날이 새롭게 진화하기 때문에 IDS, IPS 방화벽은 여기에 발 빠르게 쫓아 갈 수 있어야 한다. 실제로 이제까지 나타난 다양한 공격 방법에 대한 룰셋이 공개되어 있고 이를 적용할 수 있다. 하지만 룰셋 자체를 이해하는 데에는 많은 시간과 실험과정이 필요하다. snort 및 관련 룰셋에 대한 매뉴얼만 봐도 방대한 기술에 대한 이해가 필요하기 때문에 본격적인 구현을 위해서는 더욱 깊이 있는 분석이 필요하다.

[0128] 하지만 아무리 snort가 막강하고 룰셋을 잘 적용했다고 하더라도 앞서 언급한 2가지 오탐인 false positive와 false negative는 발생할 수밖에 없다. 룰셋을 엄격하게 적용하면 정상적인 사용까지 차단시키는 문제가 발생하고 느슨하게 적용하면 침입을 허용하는 문제가 발생한다. 또한, 룰셋을 너무 엄격하게 적용할 경우 속도 딜레이가 발생하여 응답성이 떨어질 수도 있다.

[0129] 본 실시예에서는 이를 해결하고자 룰셋을 느슨하게 적용하여 빠른 검출이 가능한 패턴은 즉시 차단시키고 의심스러운 침입 공격에 대해서는 허용을 하는 대신 트래픽 쉐이핑을 통해서 전체 네트워크에 문제를 일으키지 못하게 하는 방법을 제공한다. 또한, 이런 의심 패킷의 트래픽에 대해서는 별도 모니터링을 하면서 공격으로 판단이 되면 차단시키도록 하여 룰셋을 느슨하게 하면서도 트래픽을 제어하면서 내부 문제를 발생시키지 않고 천천히 확실히 잡아낼 수 있는 방법을 제공한다.

[0131] 1. 실시간 트래픽 쉐이핑 및 IPS 계산

[0132] 실시간 트래픽 쉐이핑 계산은 앞서 언급한 바와 같이 CPU 부하량 계산과 패킷 DROP율 계산 그리고 다중 큐잉에서 큐잉당 트래픽 계산에서 제시한 방법을 적용하며 여기서는 구체적인 설명을 생략한다. 여기에 IPS 룰셋에서 다음과 같이 확장 적용하여 의심 패킷에 대해서는 별도의 제한된 큐잉에 보내고 모니터링 하면서 공격 확인이 되면 차단시키는 방법을 적용한다.

[0133] 1) IPS룰셋

[0134] IPS와 트래픽 쉐이핑의 상호 동작을 확인하기 위해서 다음의 공격을 가정한다. 보통 공격의 전초 단계인 포트 스캔이나 DOS 공격의 하나로 대표적인 TCP SYNC-FIN 공격의 예를 들면 다음과 같다.

[0135] - 룰 설정

[0136] #vi local.rules

[0137] .....

[0138] alert tcp any any -> 192.168.219.0/24 any (flags: SF; ack: 0; msg:"SYNC-FIN packet detected"; sid:2000013;)

[0139] .....

[0140] - 의미 분석

[0141] 위의 예는 공격에 대해서 alert 메시지를 발생시키라는 룰셋이다. 발생 조건은 tcp 프로토콜에 대하여, 발생지는 모든 주소에서 모든 TCP 서비스포트에 대하여, 목적지는 C 클래스192.168.219.0 네트워크로 가는 모든 TCP 프로토콜에 대하여, flag가 SF(SYNC-FIN)이면서 ACK 값이 0인 패킷이다. 이런 조건이 만족되면 "SYNC-FIN packet detected" 라는 메시지를 로깅하게 된다. TCP flag가 SYNC-FIN 이면서 ACK 값이 0인 경우는 보통의 경우 발생하지 않는다. 이런 경우는 해킹툴의 앞단에서 실행되는 TCP 포트 스캐너가 동작할 경우 나타나는 현상이다.

[0142] - 해당되는 패킷 생성방법

[0143] 외부 PC(192.168.1.66)에서 다음 명령으로 TCP 80번 포트면서 flag SYNC-FIN 인 패킷을 생성시킨다.

[0144] $sudo nping --tcp -p 80 --flags syn,fin 192.168.219.122

[0145] Starting Nping 0.6.40 (http://nmap.org/nping) at 2015-10-17 22:44 KST

[0146] SENT (0.0509s) TCP 192.168.1.66:33239 > 192.168.219.122:80 SF ttl=64 id=48684 iplen=40 seq=1932467804 win=1480

[0147] .....

[0148] - alert 메시지 로깅 확인

[0149] $sudo tail -f /var/log/*

[0150] Oct 17 13:44:35 192.168.1.1 snort: message repeated 4 times: [[1:2000013:0] SYNC-FIN packet detected {TCP} 192.168.1.66:33239 -> 192.168.219.122:80]

[0151] .....

[0152] 위와 같이 IPS 룰셋을 설정하고, nping으로 패킷을 룰셋에 걸리도록 발생시키고, tail 명령으로 로그 데이터를 모니터링하면, snort 룰셋에 정한대로 alert 메시지를 확인할 수 있다. 공격이 확실하면 DROP이나 REJECT 할 수 있지만 우선은 alert 로그만 남기고 다음 의심 패킷 관리에서 처리하도록 한다.

[0153] 2) 의심 패킷 관리

[0154] 별도의 프로세스를 통해서 snort에서 발생시킨 alert 메시지를 모니터링 한다. 의심 패킷을 찾으면 우선 대역폭이 제한된 큐에 넣고 해킹 체크 함수가 판단하게 한다. 해킹 아님이 판단되면 큐에서 빼내서 정상 패킷으로 돌리고, 해킹으로 판단이 되면 일정시간 해당 소스의 모든 패킷은 DROP(or REJECT) 시키고, 판단되지 않으면 큐에 머물면서 후속 패킷에 대해 다시 판단을 받으며 모니터링할 수 있도록 한다.

[0155] 이렇게 함으로써 false positive 오탐율은 0%까지 낮출 수 있고, false negative 오탐율은 40%까지 상승할 수 있지만, 제한된 트래픽 큐잉에서 의심 패킷 관리를 받으며 정확한 판단을 할 수 있는 점과 특히 다른 장비의 대역폭 피해를 받지 않는 장점이 있다.

[0156] 여기서는 INBOUND에 대한 테스트만 했지만 트래픽 쉐이핑은 양방향 모두 가능하기 때문에 외부 공격 외에도 내부 서버의 바이러스 감염에 의한 OUTBOUND 트래픽에 대한 큐잉도 가능하기 때문에 실질적인 트래픽 쉐이핑이 큰 효과를 발휘할 수 있다.

[0157] 2. 성능 분석

[0158] 본 실시예의 성능은 도 9와 같다. 도 9를 참조하면, false negative 오탐의 경우 1차에서는 40% 이지만 모두 제한된 대역폭의 큐에 존재하기 때문에 전체 네트워크의 속도 저하는 없다. 그리고 2차 탐지를 거치면 10% 이내의 오탐만 남게 된다. 여기서, 오탐율 0%, 10%, 40%는 모두 설정을 통해 조정할 수 있다. 중요한 점은 다른 장비에 회선 속도 저하를 유발하지 않는다는 것이며, 장비의 성능과 운용 전략에 맞추어 오탐율을 조정할 수 있다는 것이다.

[0160] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어,

프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0161] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0163] 도 10은 본 발명의 일 실시예에 따른 IDC용 통합 보안 라우터의 트래픽 쉐이핑과 IPS의 융합 구현 기반의 통합 보안 서비스 방법을 설명하기 위해 도시한 흐름도이다.

[0164] 여기서 설명하는 방법은 본 발명의 하나의 실시예에 불과하며, 그 이외에 필요에 따라 다양한 단계들이 부가될 수 있고, 하기의 단계들도 순서를 변경하여 실시될 수 있으므로, 본 발명이 하기에 설명하는 각 단계 및 그 순서에 한정되는 것은 아니다.

[0165] 도 2 및 도 10을 참조하면, 단계(1010)에서 IDC용 통합 보안 라우터(110)의 룰셋 설정부(210)는 IDC 서버의 펌웨어 관련 환경 설정 파일에 복수의 룰을 포함하는 룰셋을 설정할 수 있다.

[0166] 다음으로, 단계(1020)에서 상기 IDC용 통합 보안 라우터(110)의 패턴 분석부(220)는 상기 IDC 서버에 의해 수집된 데이터 트래픽에 대하여 상기 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단할 수 있다.

[0167] 이때, 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽인 것으로 판단된 경우(1030의 "예" 방향), 단계(1040)에서 상기 IDC용 통합 보안 라우터(110)의 의심 패킷 관리부(230)는 상기 침입 공격 패킷을 즉시 차단시킬 수 있다.

[0168] 반면, 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽이 아니고(1030의 "아니오" 방향), 의심 패킷의 트래픽인 것으로 판단된 경우(1050의 "예" 방향), 단계(1060)에서 상기 IDC용 통합 보안 라우터(110)의 의심 패킷 관리부(230)는 상기 의심 패킷에 대해 허용을 하는 대신에 트래픽 쉐이핑을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리할 수 있다.

[0169] 한편, 상기 데이터 트래픽이 상기 침입 공격 패킷의 트래픽이 아니고(1030의 "아니오" 방향), 의심 패킷의 트래픽도 아닌 것으로 판단된 경우(1050의 "아니오" 방향), 본 실시예는 종료된다.

[0171] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CDROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도

록 구성될 수 있으며, 그 역도 마찬가지이다.

[0172] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0173] 그러므로, 다른 구현들, 다른 실시예들 및 청구범위와 균등한 것들도 후술하는 청구범위의 범위에 속한다.

부호의 설명

[0175] 110: IDC용 통합 보안 라우터

210: 룰셋 설정부

220: 패턴 분석부

230: 의심 패킷 관리부

240: 제어부

310: 다중 큐

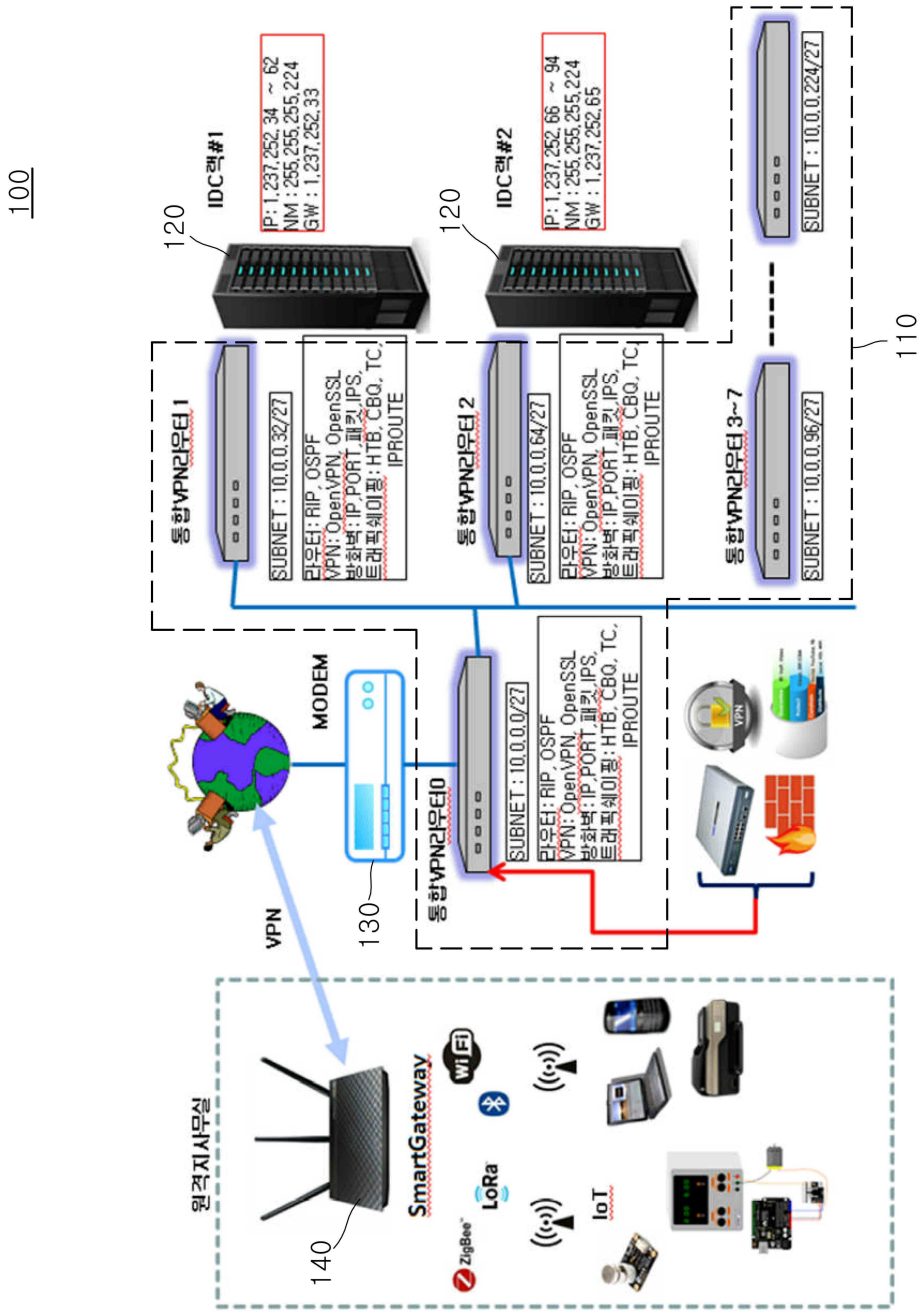
320: 큐 제어부

330: 패킷 할당부

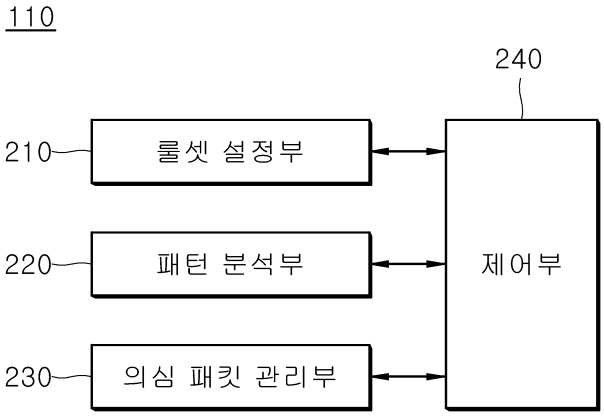
340: 모니터링부

도면

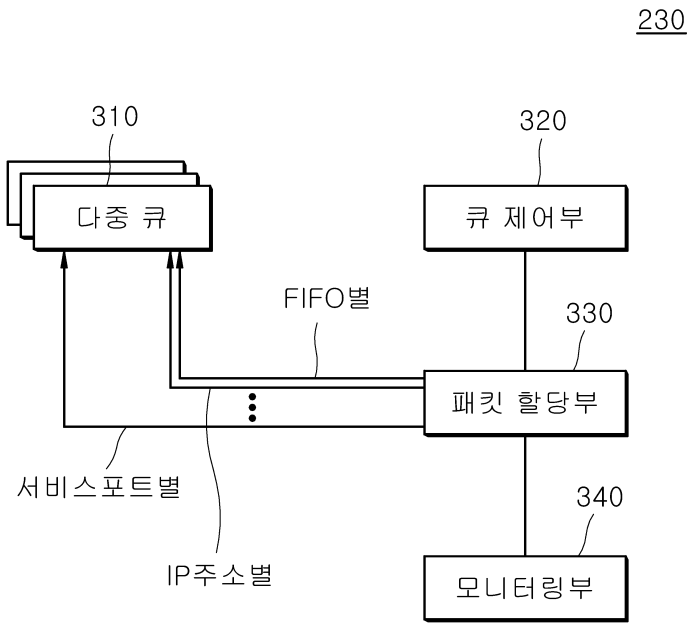
도면1


100
120
IDC객#1
IP: 1.237.252.34 ~ 62
NM : 255.255.255.224
GW : 1.237.252.33
120
IDC객#2
IP: 1.237.252.66 ~ 94
NM : 255.255.255.224
GW : 1.237.252.65
110
통합VPN라우터 1
SUBNET : 10.0.0.32/27
라우터: RIP, OSPF
VPN: OpenVPN, OpenSSL
방화벽: IP PORT 패킷 IPS,
트래픽쉐이핑: HTB, CBQ, TC,
IPROUTE
통합VPN라우터 2
SUBNET : 10.0.0.64/27
라우터: RIP, OSPF
VPN: OpenVPN, OpenSSL
방화벽: IP PORT 패킷 IPS,
트래픽쉐이핑: HTB, CBQ, TC,
IPROUTE
통합VPN라우터 3~7
SUBNET : 10.0.0.96/27
SUBNET : 10.0.0.224/27
MODEM
130
통합VPN라우터0
SUBNET : 10.0.0.0/27
라우터: RIP, OSPF
VPN: OpenVPN, OpenSSL
방화벽: IP PORT 패킷 IPS,
트래픽쉐이핑: HTB, CBQ, TC,
IPROUTE
VPN
원격지사무실
140
SmartGateway
WiFi
LoRa
ZigBee
IoT

도면2



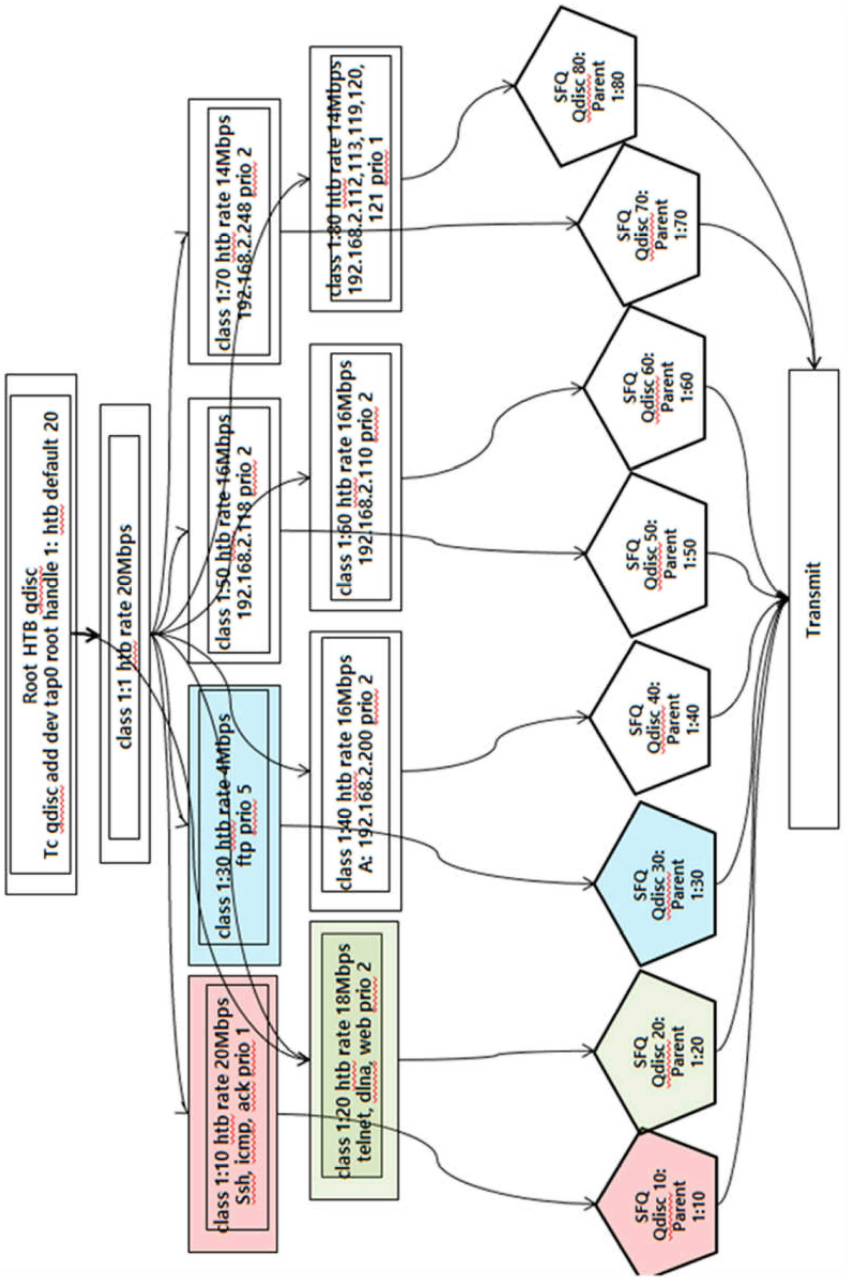
도면3



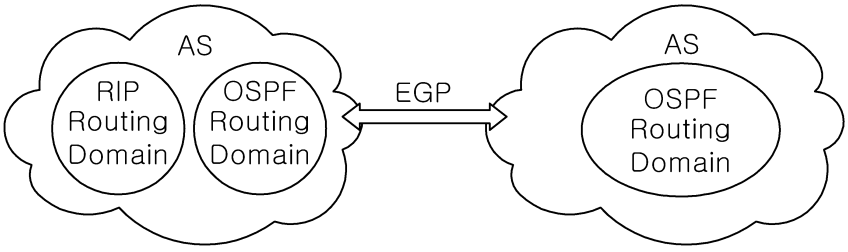
도면4

제품	H/W Spec	S/W Spec
MT7621	-DualCore 880Mhz -Flash:16MB -RAM:128MB -LAN/WAN: 1G -WIFI: 802.11AC /N/G/B/A 2.4G/5G -USB3.0, PCI-E, MicroSD	-OpenWRT 14.07 / 15.05 - RIP, OSPF, Zebra, OpenVPN, iproute, snort, LuCi

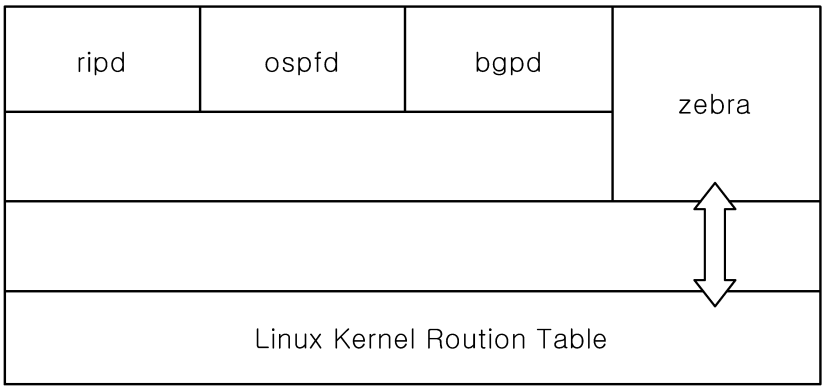
도면5


Root HTB qdisc
Tc qdisc add dev tap0 root handle 1: htb default 20
class 1:1 htb rate 20Mbps
class 1:10 htb rate 20Mbps
Ssh, icmp, ack prio 1
class 1:20 htb rate 18Mbps
telnet, dlna, web prio 2
class 1:30 htb rate 4Mbps
ftp prio 5
class 1:40 htb rate 16Mbps
A: 192.168.2.200 prio 2
class 1:50 htb rate 16Mbps
192.168.2.110 prio 2
class 1:50 htb rate 16Mbps
192.168.2.118 prio 2
class 1:70 htb rate 14Mbps
192.168.2.248 prio 2
class 1:80 htb rate 14Mbps
192.168.2.112,113,119,120,
121 prio 1
SFQ Qdisc 10: Parent 1:10
SFQ Qdisc 20: Parent 1:20
SFQ Qdisc 30: Parent 1:30
SFQ Qdisc 40: Parent 1:40
SFQ Qdisc 50: Parent 1:50
SFQ Qdisc 60: Parent 1:60
SFQ Qdisc 70: Parent 1:70
SFQ Qdisc 80: Parent 1:80
Transmit

도면6


AS
RIP Routing Domain
OSPF Routing Domain
EGP
AS
OSPF Routing Domain

도면7



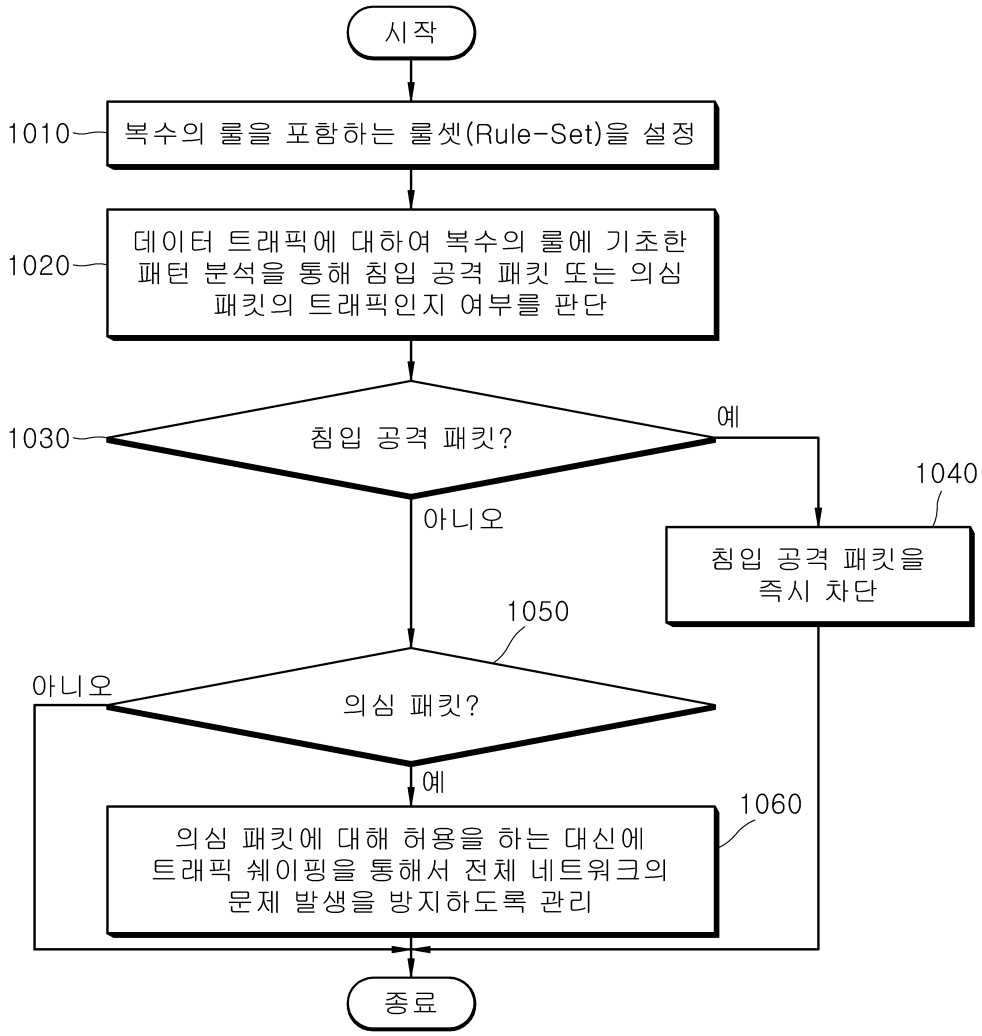
도면8

Name	Attack Rule
Social Engineering	관리자나 사용자 속이기
Impersonation	일반 사용자의 권한 뺏기
Exploits	시스템 보안 취약점 이용
Transitive Trust	신뢰하는 호스트나 네트워크 위장
Data Driven	attack program, trojan, backdoor, virus
Infrastructure	protocol/system 기본 기능 취약점
Denial of Service	system 의 정상 동작 방해
Distributed Denial Of Service	분산 DoS

도면9

번호	항목	연구결과
1	false positive 오탐율	0%
2	false negative 오탐율 1차	40%
3	false negative 오탐율 2차	10%
4	해킹공격시 및 오탐시 네트워크 속도저하	없음

도면10


시작
1010
복수의 룰을 포함하는 룰셋(Rule-Set)을 설정
1020
데이터 트래픽에 대하여 복수의 룰에 기초한 패턴 분석을 통해 침입 공격 패킷 또는 의심 패킷의 트래픽인지 여부를 판단
1030
침입 공격 패킷?
예
1040
침입 공격 패킷을 즉시 차단
아니오
1050
의심 패킷?
아니오
예
1060
의심 패킷에 대해 허용을 하는 대신에 트래픽 쉐이핑을 통해서 전체 네트워크의 문제 발생을 방지하도록 관리
종료