



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0068352
(43) 공개일자 2010년06월23일

(51) Int. Cl.

G06F 21/00 (2006.01) G06F 17/00 (2006.01)

(21) 출원번호 10-2010-0052042(분할)
(22) 출원일자 2010년06월01일
심사청구일자 2010년06월01일
(62) 원출원 특허 10-2008-0086918
원출원일자 2008년09월03일
심사청구일자 2008년09월03일

(71) 출원인

충남대학교산학협력단

대전광역시 유성구 공동 220번지 충남대학교

한신대학교 산학협력단

경기 오산시 양산동 한신대학교내

이형우

경기도 용인시 기흥구 중동 성산마을서해그랑블
3101동 501호

(72) 발명자

이형우

경기도 용인시 기흥구 중동 성산마을서해그랑블
3101동 501호

(74) 대리인

특허법인태동

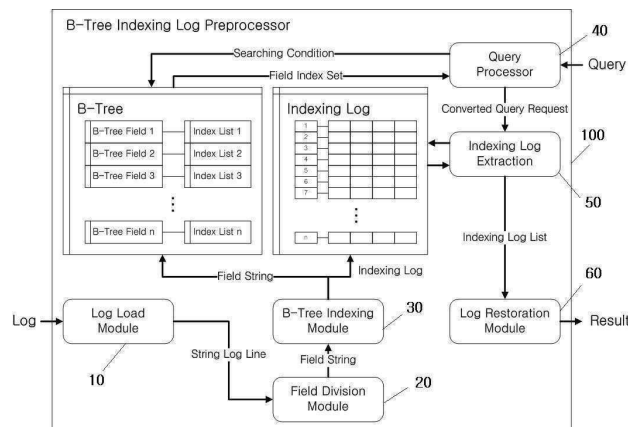
전체 청구항 수 : 총 1 항

(54) 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반 웹로그 복구방법

(57) 요약

본 발명은 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반의 웹로그 복구방법에 관한 것으로, 웹 아이디에스 시스템의 공격탐지를 위하여 일정한 포맷으로 구성된 비트리인덱스벡터기반의 구조를 가지는 웹 로그를 복구하는 방법에 있어서, 목록에서 해당하는 로그 인덱스 정보를 가져오는 제1 단계; 해당 필드의 비-트리에서 인덱스를 이용하여 문자열을 가져와서 통합하는 제2단계; 로그 텍스트를 복구하는 제3 단계;를 포함하여 이루어진 것을 특징으로 한다.

대표도 - 도12



특허청구의 범위

청구항 1

웹아이디에스 시스템의 공격탐지를 위하여 일정한 포맷으로 구성된 비트리인덱스벡터기반의 구조를 가지는 웹 로그를 복구하는 방법에 있어서,

목록에서 해당하는 로그 인덱스 정보를 가져오는 제1 단계;

해당 필드의 비-트리에서 인덱스를 이용하여 문자열을 가져와서 통합하는 제2단계;

로그 텍스트를 복구하는 제3 단계;를 포함하여 이루어진 것을 특징으로 하는 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반의 웹로그 복구방법.

명세서

기술분야

- [0001] 본 발명은 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반 웹로그 복구방법에 관한 것으로, 더욱 상세하게는 비-트리인덱스 기법을 통해 키워드의 비교를 통해 비-트리를 구성하고 웹 로그 정보내에 생성되는 정보에 대한 효율적인 인덱싱 구조를 실현함으로써 웹로그를 고속으로 검색하고 복구할 수 있도록 하는 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반 웹로그 복구방법에 관한 것이다.

배경기술

- [0002] 국내의 인터넷 이용률은 꾸준히 증가 추세에 있으며, 현재 국내 인터넷 사용자 수는 약 34,430천명을 넘고 있다. 2007년 6월 만 6세 이상 인터넷 이용률(최근 1개월 이내 인터넷 이용자의 비율)은 75.5%에 달하고 있다. 이처럼 국내 인터넷 이용자수 및 이용률은 꾸준히 증가 추세에 있으며, 사용자의 연령층 및 직업군 또한 다양해지고 있다.
- [0003] 인터넷 사용자의 증가에 따라 국내 주요 포털 웹 사이트에서 생성되는 웹 로그의 양은 평균 50GB/일 내외에 해당하는 대용량의 로그가 발생하고 있다. 이와 같이 웹 서비스 양적 증가와 더불어 웹 서비스에서의 취약점도 날로 급증하고 있어서 국내 주요 포털 사이트에 대한 웹 공격 시도 및 성공율도 함께 증가하고 있다.
- [0004] 또한 Web 2.0의 출현으로 웹 서비스는 더욱 확대되고 있으며 광고(ActiveX), 피싱, 악성코드 유포, 개인정보 관련 공격, 피싱, Service Hacking, XSS 공격, SQL Injection 및 script attack 등의 취약점을 이용한 웹 포털 사이트 공격은 급증하고 있다. 특히 웹 서비스에 SQL Injection / Parameter Injection 및 DoS 등의 공격을 통해 포털 사이트에 대한 악의적 공격을 시도하거나 웹 사용자에게 대한 정보를 불법적으로 유출하는 등의 취약점이 발견되고 있다.
- [0005] 따라서 이와 같은 형태의 웹 공격에 능동적으로 대응하기 위해서는 우선적으로 웹 사이트에서 생성되는 웹 로그(Web Log) 정보를 분석하여 공격 시도를 탐지하는 방법을 적용해야 한다. 웹 로그 정보는 인터넷을 주요사업 및 마케팅 수단으로 사용하는 기업들에게 소비자들의 성향 분석 및 보다 발전된 서비스 제공을 위한 도구로 사용된다. 그러나 수집된 웹 로그 데이터는 분석에 바로 이용될 수 있는 형태가 아니다. 웹 로그를 통해 사용자의 성향을 분석하고 웹 사이트의 통계적인 데이터를 추출하며 동시에 웹 공격 등을 사전에 차단하기 위해서는 웹 로그 정보를 대상으로 전처리 기법(Web Log Preprocessing)을 적용하여 포털 사이트로부터 생성되는 대용량 웹 로그 정보에 대한 고속 인덱싱 기법이 필요하다.
- [0006] 그러나 기존의 웹 IDS 시스템 및 웹 마이닝 기반 공격 탐지 기법은 대용량 웹 로그에 대한 고속 인덱싱 과정 없이 웹 서버에 대한 공격 탐지 과정을 수행하기 때문에 Web IDS/IPS 시스템 구동시 탐지율이 저하된다는 문제점을 갖고 있다.
- [0007] 기존의 IDS 시스템에서는 공격을 탐지하기 위해 IP 패킷을 대상으로 쿼리 데이터를 이용해 공격 여부를 탐지한다. 그러나 웹 서버의 공격 탐지를 위해서는 웹 서버에서 생성되는 웹 로그(Web Log) 데이터에 대한 분석을 통해 의

부로부터의 불법적인 접속을 탐지하거나 이상 탐지(Anomaly Detection) 기능을 제공해야 한다.

- [0008] 기존의 웹 IDS 시스템은 웹 로그를 기반으로 외부로부터의 공격이나 웹 시스템 내부의 부적절한 쿼리 전송 및 이상 접속 정보를 탐지하기 위해 공격 탐지 룰(Web Attack Rule) 정보를 사용한다.
- [0009] 하지만 기존의 웹 IDS 시스템은 대량으로 생성되는 웹 로그에 대한 별도의 전처리 과정 없이 웹 공격 탐지 룰과 비교하는 방식이므로 실시간으로 수행되는 웹 공격에 효율적으로 대처하지 못하고 있다.
- [0010] 또한 기존의 웹 IDS는 로그 파일의 전처리 과정이 없이 순차 검색을 통해 룰과의 비교를 수행하고 있다. 따라서 최악의 경우 웹 로그 생성 후 일정 시간이 경과된 이후 에서는 탐색 결과를 제시하게 된다는 단점이 있다.
- [0011] 따라서 웹 로그에 대한 전처리 과정과 함께 웹 로그 마이닝 기법을 접목하여 시스템에 적용할 경우 앞에서 제시한 악성코드 기반 공격, BotNet 등을 통한 DoS 공격 등 웹 서버에서의 공격 행위들에 대해 효율적으로 탐지/대응할 수 있을 것으로 예상된다.
- [0012] 웹 마이닝 기법은 크게 콘텐츠 마이닝, 이용정보 마이닝 및 구조 마이닝 기법으로 나눌 수 있다. 최근 Web2.0 기술이 부각되면서 동적 웹 사이트 구축이 급증하고 있다. 따라서 분석 대상이 되는 웹 로그 정보의 양 또한 급증하고 있어 이에 대한 대응 기술이 연구되어야 한다. 웹을 대상으로한 공격에 능동적으로 대응하기 위해서는 인덱싱 기법과 마이닝 기법이 결합되어야 할 것으로 예상된다. 대용량 웹 로그 정보에 대한 고속 인덱싱 기법을 통해 실시간 분석 기능을 제공할 수 있으며, 다양한 형태의 웹 공격을 판단할 수 있을 것이다.
- [0013] 도 1은 대용량 웹 정보를 대상으로 마이닝 기술을 적용하는 과정을 나타낸 개념도로서, 시스템 내부의 로그파일을 분석해 정의된 룰과 비교를 통해 공격을 탐지하는 과정을 나타낸 개념도이다.
- [0014] 도 2를 참조하면, 종래에는 대단위 웹 로그 정보에 대한 고속처리 기술이 미비하며, 로그 정보내 기록된 문자열 정보에 대한 처리기능 미비로 인해 공격 정보에 대한 효율적인 탐지 기능이 부족하다. 또한 종래기술은 대부분 웹 사용자의 이용 정보 등을 추출하기 위한 단계에 머무르고 있다. 종래 기술은 단지 웹 로그 정보로부터 사용자의 행위 정보를 예측하기 위한 과정으로 마이닝 기법을 사용하고 있다. 종래의 기술은 사용자가 앞으로 어떠한 페이지로 이동할 지에 대한 정보 등을 추출하기 위한 용도로 사용되고 있다.
- [0015] 하지만 웹 로그 마이닝 기술을 웹 로그내 공격 탐지 기술에 접목한다면 기존의 기법보다 더욱 개선된 탐지율을 보일 수 있게 된다. 따라서 이를 위해서는 대용량 웹 로그에 대한 고속 인덱싱 기술을 접목할 필요가 있다.
- [0016] 상기 설명한 바와 같이 웹 마이닝은 웹 로그 분석을 통해 웹 페이지 이용자의 접근 패턴을 자동으로 발견해내는 것을 말한다. 일반적으로 웹 서버에는 일정한 형태의 포맷을 구성하는 웹 로그가 생성된다. IIS 서버나 아파치 서버에서 생성되는 웹 로그는 W3C 포맷을 따르고 있다.
- [0017] 도 3은 종래기술에서 사용한 IIS 6.0의 W3C 확장 로그 포맷에 대하여 설명한다. 도 3의 여러 가지 로그 필드 중에서 웹 이용 마이닝에 이용되는 필드는 일부 필드들의 복합적인 연간 관계를 고려해 특정 필드의 정보만이 추출되어 사용된다. 따라서 대용량 웹 로그 정보로부터 이상탐지를 위한 방법을 도출하고 마이닝 정보를 추출하기 위해 도 4와 같은 로그 필드별 공격탐지 관련정보의 연관성을 도출하였다.
- [0018] 도 4와 같이 대용량 웹 로그 정보를 이용하여 웹 이상행위를 탐지하고 공격탐지 등을 적용하여 마이닝 기법과 연계하기 위해서 로그 필드 요소 추출과정이 수행되어야 한다. 그리고 관련 없는 로그 정보의 정제와 여과 과정, 이용자 분류, 분류된 이용자의 세션 분류, 분류된 세션을 고려해 웹 사이트에서의 이동경로 추출 등 사전에 웹 로그의 전처리 모듈이 개발되어야 한다.
- [0019] 종래기술의 웹 로그 전처리 방법은 도 5와 같이 크게 5단계로 구분할 수 있다.
- [0020] - 1단계 (Cleaning Log) - 분석에 필요하지 않은 로그 정보들을 삭제하는 전처리 과정이다. 일반적으로 jpg, jpeg, gif, swf 등 웹 페이지를 구성하는 멀티미디어 파일들은 사용자가 웹페이지 요청시 자동으로 요청되어 사용자의 의도와는 무관한 데이터들로 웹 이용 마이닝에 불필요한 정보들이다. 그러므로 삭제 대상이 된다. 이렇게 삭제된 로그는 1/20, 1/40의 효율을 볼 수 있다.
- [0021] - 2단계 (User Identification) - 웹 로그 정보를 이용해 사용자를 식별하는 단계. 일반적으로 c-ip와 c(User Agent)필드 정보를 이용해 사용자를 식별함. 그러나 프록시 서버를 사용할 때 단순히 c-ip와 user agent 정보만

으로 동일한 컴퓨터를 이용하는 서로 다른 사용자를 식별할 수 없다는 문제점을 가지고 있다.

- [0022] - 3단계 (Session Identification) - 세션이란 사용자의 로그인 유무를 나타내는 것이 아니다. 사용자가 브라우저로 웹 페이지를 이용할 때부터 세션 ID가 부여된다. 그러므로 사용자 개인의 웹 페이지 내에서 이동 경로 및 선호도 분석을 위해 가장 중요한 전처리 과정이다.
- [0023] - 4단계 (Path Completion) - 분류된 세션 정보를 이용해 웹 페이지 내에서의 사용자 이동경로를 추적하고, 일반적으로 back 또는 forward 버튼을 눌러 이동한 경우의 경로 연결 과정을 수행한다.
- [0024] - 5단계 (Formatting) - 웹 로그 분석에 적합한 정보의 형태로 포맷을 전환한다. 포맷 형태는 사용되는 분석 기법에 따라 결정한다.
- [0025] 위 단계를 거친 웹 로그는 웹 마이닝 기법에 유용한 자료로 사용된다. 그러나 현대 웹 환경의 발달 및 변화로 웹 로그의 데이터 형태가 더욱 복잡해지고 크기 또한 대용량화 되어가고 있어, 기존의 전처리 방식은 새로운 개선이 시급한 실정이다.
- [0026] 웹의 발달로 웹 로그의 크기는 대용량화되고 있다. Microsoft.com의 하루 평균 웹 로그의 양은 200~300GB이고, 최고 1TB의 웹 로그가 발생되기도 한다. 이처럼 대용량의 웹 로그를 기존의 방식대로 전 처리 하는 것은 비효율적인 방법이다. 기존 전처리 단계의 1단계 (Cleaning Log)를 수행하기 위해서는 로그 정보의 접미사(suffix)를 검사함으로써 가능하다. 예를 들어 파일 이름의 접미사가 'gif, jpeg' 등인 모든 로그 정보는 삭제될 수 있다. 그러나 기존 전처리 기법에서는 도 3에서의 전체 필드의 순차적인 검사를 통해 Cleaning Log 과정을 수행한다.
- [0027] 웹 로그 각각의 필드 정보를 이용해 새로운 정보를 생성하기 위해서는 필드별 분류 기법 및 특정 문자열 로그에 대한 결과 내 검색 기능이 필요하다. 예를 들어 웹 서버에 특정한 자원을 일정한 주기로 요청하는 로그 정보는 웹 스캔공격으로 레포팅 할 수 있다. 이 결과는 cs-uri-stem필드를 추출해 date 및 time필드의 결과 내 검색 기능으로 탐지할 수 있다. 그러나 기존의 웹 로그 전처리 기법은 필드 단위의 탐색 및 결과 내 검색 기능의 부재로 복잡한 로그 필드를 이용한 마이닝에 비효율적인 문제점이 있었다.

발명의 내용

해결하려는 과제

- [0028] 본 발명의 목적은 B-트리 인덱스 벡터(B-Tree Index Vector) 구조를 이용하여 대량의 웹 로그 정보를 고속으로 검색하고 복구하기 위한 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반 웹로그 복구방법을 제 공함에 있다.

과제의 해결 수단

- [0029] 상기한 바와 같은 목적을 달성하기 위한 본 발명의 바람직한 실시예에 따르면,
- [0030] 웹아이디에스 시스템의 공격탐지를 위하여 일정한 포맷으로 구성된 비트리인덱스벡터기반의 구조를 가지는 웹 로그를 복구하는 방법에 있어서,
- [0031] 목록에서 해당하는 로그 인덱스 정보를 가져오는 제1 단계;
- [0032] 해당 필드의 비-트리에서 인덱스를 이용하여 문자열을 가져와서 통합하는 제2단계;
- [0033] 로그 텍스트를 복구하는 제3 단계;를 포함하여 이루어진 것을 특징으로 하는 대용량 웹로그마이닝 및 공격탐지를 위한 비트리인덱스벡터기반의 웹로그 복구방법이 제공된다.

발명의 효과

- [0034] 상기 설명한 바와 같이, 본 발명에 따른 비트리인덱스벡터기반 웹로그 복구방법에 의하면, cs-uri-stem필드를 추출해 중복된 로그 정보를 인덱스화하여 Cleaning Log 과정을 수행함으로써 종래에 비해 효율적이다.
- [0035] 본 발명에 따르면, 웹 로그 정보에 대한 필드 단위 파싱 및 효율적인 인덱싱 처리 과정을 수행하며 마이닝 알고리즘을 기반으로 대용량의 로그에 대해 공격 탐지 및 실시간 대응 성능을 향상시킬 수 있을 것으로 예상되어 개선된 웹 IDS/IPS 시스템 구축이 가능한 효과가 있다.

[0036] 본 발명에 따르면, 로그 정보를 로드해 필드별 파싱 기법과 중복 문자열을 고려한 필드 단위 인덱스 기법을 이용해 로그 정보의 크기가 혁신적으로 줄어들게 되는 효과가 있고, 또한 B-트리 구조로 필드 단위의 인덱스 정보를 구성해 기존 전처리 과정의 성능을 향상시킬 수 있는 효과가 있다.

[0037] 또한, 본 발명에 따르면, 대용량 웹 로그 정보에 대해 멀티쓰레드 기반 고속 전처리 및 검색 기능을 제공하였기 때문에 웹 로그 정보내 이상탐지 및 마이닝 기법과 연계할 경우 전체적인 시스템의 성능을 높일 수 있는 효과가 있다.

도면의 간단한 설명

- [0038] 도 1은 종래의 대용량 웹 로그 전처리방법의 일예를 나타내는 개념도이다.
- 도 2는 종래의 웹 로그 정보 기반 마이닝방법의 일예를 나타내는 개념도이다.
- 도 3은 종래기술(IIS 6.0)의 W3C확장 로그 포맷을 나타낸 것이다.
- 도 4는 도 3의 각 필드별 이상행위 고려사항을 나타낸 것이다.
- 도 5는 종래의 웹 로그 전처리방법의 일예를 나타낸 개념도이다.
- 도 6은 본 발명에 따른 웹 로그 전처리방법의 개략적인 개념도이다.
- 도 7은 본 발명에 따른 본 발명에 따른 웹 로그 전처리방법에서의 멀티쓰레드 기반 웹 로그 인덱싱과정을 나타낸 개념도이다.
- 도 8은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법의 일예를 나타낸 흐름도이다.
- 도 9는 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서 비-트리 구조를 나타낸 것이다.
- 도 10은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 웹로그 복구방법을 나타낸 개념도이다.
- 도 11은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 고속검색방법의 일예를 나타낸 도면이다.
- 도 12는 본 발명에 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 비-트리 인덱싱 로그프로세서의 개략적인 블록구성도이다.
- 도 13은 도 12의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0039] 이하 본 발명에 따른 비트리인덱싱벡터기반 웹로그 복구방법을 첨부도면을 참조로 상세히 설명한다.
- [0040] 도 6은 본 발명에 따른 웹 로그 전처리방법의 개략적인 개념도이고, 도 7은 본 발명에 따른 본 발명에 따른 웹 로그 전처리방법에서의 멀티쓰레드 기반 웹 로그 인덱싱과정을 나타낸 개념도이고, 도 8은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법의 일예를 나타낸 흐름도이고, 도 9는 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서 비-트리 구조를 나타낸 것이고, 도 10은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 웹로그 복구방법을 나타낸 개념도이고, 도 11은 본 발명에 따른 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 고속검색방법의 일예를 나타낸 도면이고, 도 12는 본 발명에 비-트리기인덱싱 벡터기반 웹로그 고속검색방법에서의 비-트리 인덱싱 로그프로세서의 개략적인 블록구성도이고, 도 13은 도 12의 흐름도이다.
- [0041] 도 6을 참조하면, 본 발명에 따른 비트리인덱싱벡터기반 웹로그 고속검색방법은 대용량 웹 로그내 공격 탐지의 효율성을 높이기 위해 웹 로그 정보에 대한 고속 전처리/인덱싱 과정을 수행한다. 또한 실시간 공격 대응 기술을 제공하기 위해 로그 인덱싱 기반 마이닝 기법과 연계할 수 있는 방안을 제시하고자 한다. 제시한 모듈별 상세 구조는 다음과 같다.
- [0042] IIS 로그는 하나의 거대한 크기의 텍스트 정보이다. 하지만 하나의 거대한 정보를 각 라인으로 나눌 수 있고, 각 라인은 다시 필드들로 나눌 수 있다. 라인마다 필드들로 이루어져 있고, 필드마다 각각의 해당 정보를 가지고 있기 때문이다. 그러므로 필요한 정보를 거대한 크기의 텍스트 정보에서 바로 검색하는 것보다는 문제를 분

할하여 해결하는 분할 정복 방법을 적용하고 멀티쓰레드를 이용하여 고속 인덱싱 기술을 제공하였다(도 7 참조).

- [0043] 제안한 방법은 각 필드의 정보를 인덱싱하고 검색에 최적화된 B-트리로 생성하는 것이다. 웹 로그를 분석하기 위해서는 웹 로그를 로드하는 시점에서 바로 B-트리로 변환하는 것이 필요하다. 그러므로 웹 로그를 불러옴과 동시에 위 작업을 수행한다.
- [0044] 중복 문자열의 처리는 웹 로그의 로드 기능이 수행되면서 자동으로 실행된다. STL(Standard Template Library)의 map 모듈은 키워드와 데이터로 이루어져 있으며, 키워드의 비교를 통해 B-트리로 구성한다.
- [0045] 대용량 웹 로그 정보 내에는 시간정보, 송신자 IP 및 수신자 IP 정보 그리고 사용자의 접속 환경과 접속 URI 정보 등을 포함하고 있다. 따라서 대형 포털 사이트인 경우 하루에도 수십 GB 정도로 생성되는 웹 로그 정보를 효율적으로 저장/관리하고 공격탐지 및 웹 마이닝 등에 적용하기 위해서는 효율적인 자료 구조를 사용해야 한다.
- [0046] 따라서 본 발명에서는 B-트리 구조를 적용하여 로그 정보내 생성되는 정보에 대한 효율적 인덱싱 구조를 구현하게 되었다.
- [0047] 키워드는 필드의 정보로 하고 데이터는 필드 정보의 인덱스로 지정하였다. 만약 'Key5'의 정보가 map에 들어가게 되면 B-트리 내에 이미 인덱싱 정보가 있을 경우 추가적으로 인덱싱 과정을 수행하지 않고 이미 부여된 인덱싱 값을 이용하여 로그 정보를 저장/관리할 수 있다.
- [0048] 본 발명에서는 도 8과 같은 B-트리 인덱싱 벡터 기반 웹로그 정보 전처리방법이 수행된다. 대용량 웹 로그 정보를 B-트리 기반의 인덱스 벡터(B-Tree Index Vector) 값으로 변환하게 되면 전체적인 저장 용량이 감소하게 될 것이다. 또한 웹 로그 정보내 검색 과정에서도 문자열에 대한 검색보다 인덱스 값에 의한 검색 과정을 수행한다면 더욱더 빠른 결과를 제시할 수 있을 것이다. 따라서 본 발명에서는 IIS 웹 로그 정보에 대해 B-트리 기반 인덱스 값을 부여하여 성능 개선을 가져올 수 있었다. 결국 웹 로그 정보는 다음과 같이 인덱스 기반 전처리처리 과정을 수행한 후에 저장되기 때문에 전체 전처리 시간과 파일 크기를 최소화 할 수 있었다.
- [0049] 웹 로그에 대해 도 9와 같은 인덱스 키 값 리스트 정보와 B-트리 구조를 갖는다. 인덱스 키 값 리스트는 웹 로그 필드별 고유 키 값 정보를 저장하고 있으면서 고속으로 웹 로그 원본 텍스트로 복구하는 기능도 제공한다.
- [0050] 도 10과 같이 각각의 웹 로그 정보를 인덱스화하여 축약된 형태로 변형 저장하였기 때문에 웹 로그 원본 텍스트 값으로 복구하고자 하는 경우 해당 로그 인덱스 벡터 값을 보고 B-트리내에 저장된 인덱스 키 값을 찾아서 해당 필드 값으로 조합하여 구성하게 되면 원본 웹 로그 정보를 재구성할 수 있다.
- [0051] 각 필드별로 인덱스 기반 B-트리 정보를 가지고 있기 때문에 해당 필드별로 저장된 정보를 검색하여 멀티쓰레드 방식으로 재구성하는 과정을 수행하도록 하여 재구성시 소요되는 시간 지연을 최소화 할 수 있었다.
- [0052] 이와 같은 구조를 사용하였기 때문에 대용량의 웹 로그 정보에 대한 검색을 수행할 경우에도 성능을 최대화할 수 있다.
- [0053] 도 11과 같이 웹 로그 인덱스 벡터 값을 이용해서 필드의 인덱스 번호를 알고 있을 시에 인덱스 번호를 이용하여 직접적으로 필드의 정보에 고속으로 접근할 수 있는 구조를 제공한다.
- [0054] 검색을 시작할 때는 기본적으로 필드의 정보를 가지고 검색을 한다. 그럴 때는 탐색의 최적화된 B-트리에서 정보를 찾아 그에 대한 인덱스 값을 구한다. 그리고 그 이후에 모든 작업은 구해진 인덱스 번호만으로 처리를 한다. 인덱스 번호를 구한 이후에 또다시 B-트리에서 검색할 필요가 없이 인덱스 번호만 가지고 값을 참조할 수 있다.
- [0055] 도 12를 참조하면, 비-트리기반 인덱싱로그 프로세서(100)는 입력되는 웹로그를 로딩하는 로그 로드모듈(10)과, 상기 로그 로드모듈(10)로부터 전달된 웹로그를 필드별로 분할하는 필드분할모듈(20)과, 상기 필드분할모듈(20)에 의해 분할된 필드를 통해 비-트리를 구성함과 아울러 인덱스 리스트를 작성하고 인덱싱 로그를 생성하는 비-트리인덱싱모듈(30)과, 외부 쿼리(query)에 대하여 비-트리를 통해 검색조건에 맞는 비-트리를 검색하고 필드 인덱스 세트를 추출하고 변형된 쿼리를 요청하는 쿼리프로세서(40)와, 상기 쿼리프로세서의 요청에 따라 인덱싱 로그를 추출하는 인덱싱로그추출부(50) 및 추출된 인덱싱로그 리스트를 통해 로그를 복구하고 출력하는 로그복구모듈(60)을 포함하여 이루어진다.
- [0056] 도 13을 참조하면, 먼저 로그가 로딩되면(S1), 로그 라인을 추출하고(S2), 이후 필드 정보가 있는지를 판단하여(S3), 만약 필드 정보가 없으면 필드 정보를 생성한다(S4). 상기 단계 S3에서 만약 필드 정보가 있으면 필드 분

할을 수행하고(S5), B-트리 인덱싱을 수행한다(S6). 이후 새로운 키워드 인지를 판단하여(S7), 만약 새로운 키워드인 경우에는 키워드를 삽입하고 인덱스를 생성하고(S8) 인덱스를 획득한다(S9). 인덱스 생성시 B-트리 필드에서 키워드를 찾고(S24), 찾은 인덱스는 전달된다(S9). 만약 새로운 키워드가 아니면 인덱스를 획득하고(S9), 상기 단계 S9이후에, 인덱싱 로그를 생성한다(S10). 생성된 인덱싱 로그는 인덱싱 로그필드에 저장된다(S25).

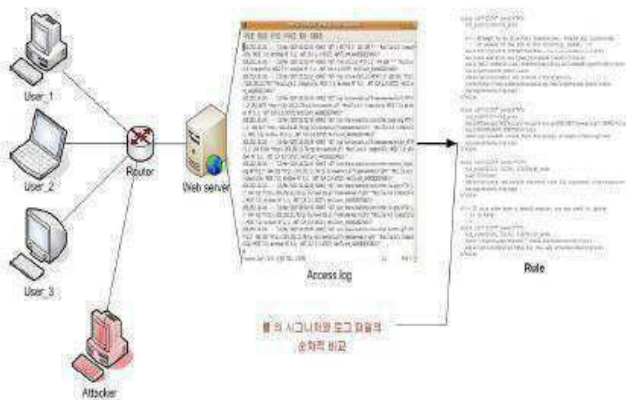
[0057] 한편, 쿼리 프로세서는 쿼리 입력시(S21) 입력된 쿼리를 분석하여 B-트리와 인덱싱 로그필드와 연계하여 인덱싱 로그를 추출하고(S23) 로그를 복구한다(S26).

부호의 설명

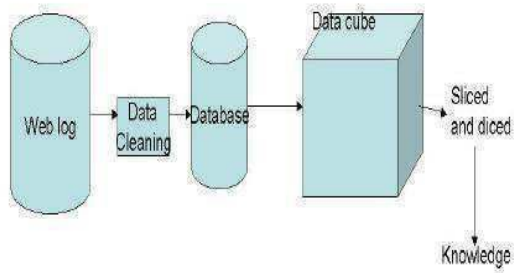
- [0058]
- 10: 로그 로드모듈
 - 20: 필드분할모듈
 - 30: 비-트리인덱싱모듈
 - 40: 쿼리프로세서
 - 50: 인덱싱로그추출부
 - 60: 로그복구모듈
 - 100: 비-트리기반 인덱싱로그 프로세서

도면

도면1



도면2



도면3

필드	필드내 주요 정보	설정
date	activity 발생 날짜	Y
time	activity 발생 시간	Y
c-ip	서버에 접속한 클라이언트의 IP 주소	Y
cs-username	서버에 접속한 인증된 사용자 ID (누구나 접속을 허용할 시 하이픈"-")	Y
s-sitename	클라이언트가 접속한 인터넷 사이트	N
s-computernam e	로그가 생성된 서버의 컴퓨터	N
s-ip	로그가 생성된 서버의 IP 주소	Y
s-port	클라이언트가 접속한 서버의 포트	Y
cs-method	요청된 action, (예를 들어 GET method)	Y
cs-uri-stem	사용자가 접속한 웹 서버 자원 (예를 들어 Default.htm)	Y
cs-uri-query	클라이언트 쿼리 정보	Y
sc-status	HTTP, FTP에서의 action 상태	Y
sc-win32-statu s	윈도우 운영체제의 action 상태	N
sc-bytes	서버가 전송한 바이트	N
cs-bytes	서버가 수신한 바이트	N
time-taken	action이 지속된 시간 (단위 milliseconds)	N
cs-version	프로토콜 버전	N
cs-host	호스트 예서의 콘텐츠	N
cs(User-Agent)	클라이언트의 브라우저 정보	Y
cs(Cookie)	쿠키 정보	N
cs(Referer)	현재 사이트 이전의 참조 사이트 정보	N
sc-substatus	에러 코드의 세부 정보	Y

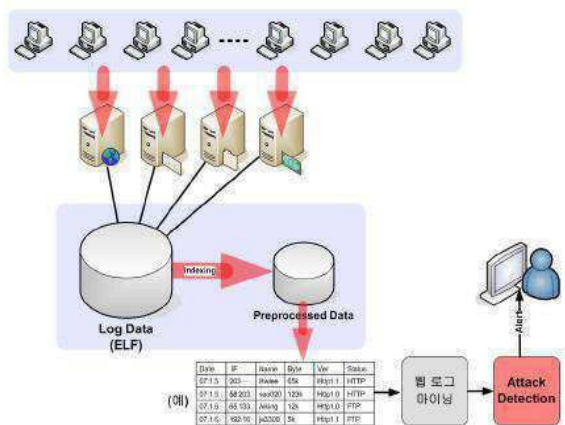
도면4

필드	필드별 이상행위
IP	한 IP에서 많이 요청되었는가?
	사용된 IP가 정상 등록된 IP인가?
date_time	특정시간에 집중되었는가?
	주기적으로 요청되었는가?
UserAgent	한 IP에서 UA의 종류가 많은가?
uri-stem	특정페이지만을 집중적으로 요청했나?
	요청된 페이지가 대중적인가?
	요청된 페이지에 취약성이 존재하는가?
uri-query	요청된 쿼리에 취약성 공격이 있는가?
sc_status	특정 에러 코드에 집중되는가?
tiometaken	처리시간이 편차가 심한가?
Bytes	전송 Byte의 편차가 심한가?
cs(referer)	정상적인 referer에서 요청되었는가?

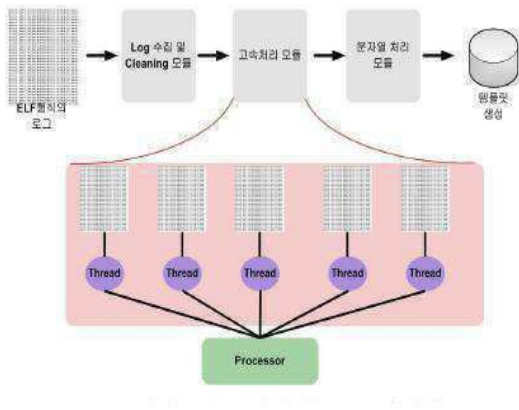
도면5



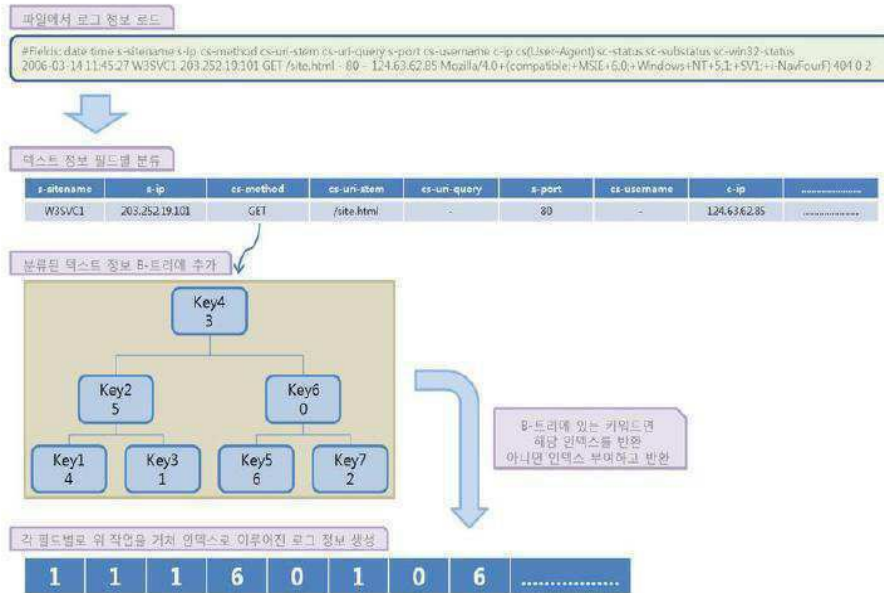
도면6



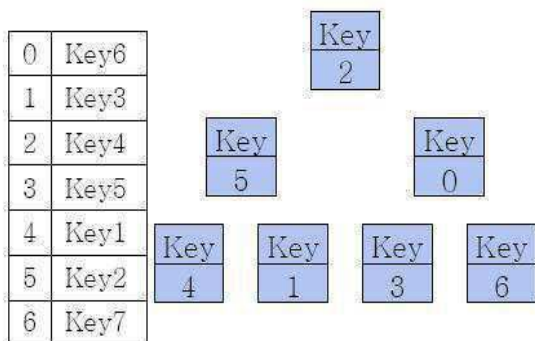
도면7



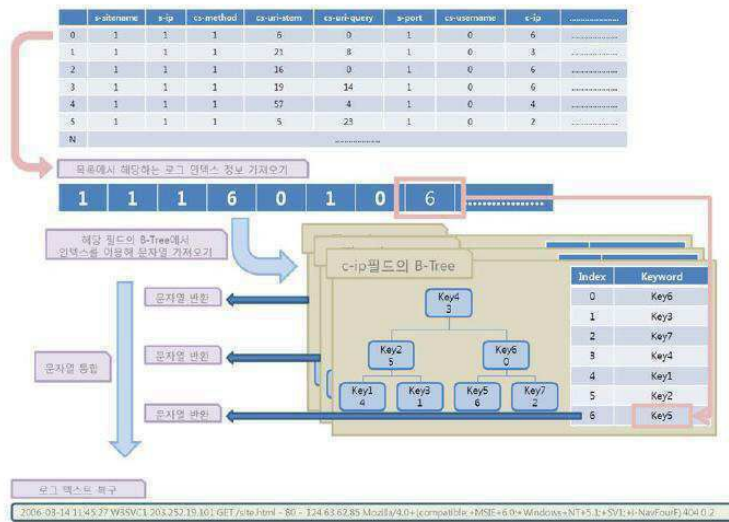
도면8



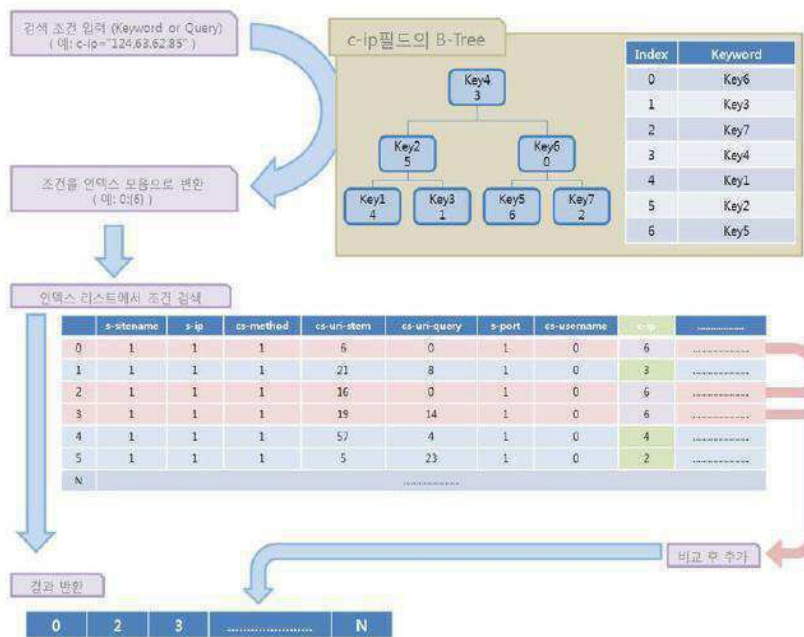
도면9



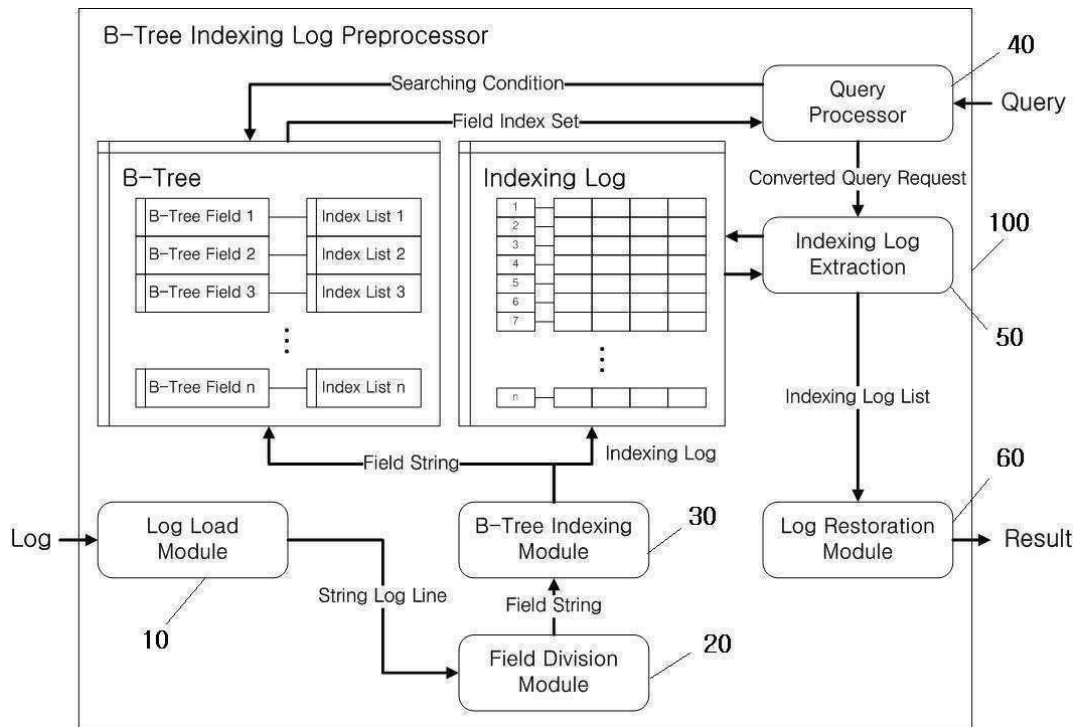
도면10



도면11



도면12



도면13

