KIPS Trans. Comp. and Comm. Sys.
Vol.9, No.7 pp.143~148 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2020.9.7.143

Log Management System of Web Server Based on Blockchain in Cloud Environment

Yong-Bum Son† · Young-Hak Kim††

ABSTRACT

Recently, web services have been expanded to various areas with the evolution of cloud environment. Whenever a user accesses a web service, the user's log information is stored in the web server. This log information is used as data to analyze the user's web service tendencies and is also used as important data to track the user's system access when a security problem in the system occurs. Currently, most web servers manage user log information in a centralized manner. When user log information is managed in a centralized manner, it is simple in the side of operation, but has a disadvantage of being very vulnerable to external malicious attacks. In the case of centralized management, user log information stored in the web server can be arbitrarily manipulated by external attacks, and in severe cases, the manipulated information can be leaked. In this case, it not only decreases the trust of the web service, but also makes it difficult to trace the source and cause of the attack on the web server. In order to solve these problems, this paper proposes a new method of managing user log information in a cloud environment by applying blockchain technology as an alternative to the existing centralized log management method. The proposed method can manage log information safely from external attacks because user log information is distributed and stored in blockchain on a private network with cloud environment.

Keywords : Blockchain, Web Site, Web Server, Access Log, Cloud

클라우드 환경에서 블록체인 기반의 웹서버 로그 관리 시스템

손 용 범† · 김 영 학††

요 약

최근에 클라우드 환경의 발전과 더불어 다양한 영역에서 웹을 통한 서비스가 확대되고 있다. 사용자가 웹 서비스에 접속할 때마다 웹 서버에 사용자의 로그 정보가 저장된다. 이러한 로그 정보는 사용자의 웹 서비스 성향을 분석하는 자료로 활용되며 또한 시스템에서 보안 문제가 발생 시에 사용자의 시스템 접속을 추적하기 위한 중요한 자료로 사용된다. 현재 대부분 웹 서버의 경우 사용자의 로그 정보를 중앙 집중 방식으로 관리한다. 사용자 로그 정보를 중앙 집중식으로 관리할 경우 운영 측면에서 단순하지만 외부의 악의적인 공격에 매우 취약한 단점을 갖는다. 중앙 집중 관리의 경우 외부 공격에 의해 웹 서버에 저장된 사용자 로그 정보가 임의로 조작될 수 있으며 심한 경우 조작된 정보가 유출될 수 있다. 이러한 경우 웹 서비스의 신뢰를 떨어뜨릴 뿐만 아니라 웹 서버의 공격에 대한 원인 발생지와 공격자의 추적이 어려워진다. 이러한 문제를 해결하기 위해 본 논문에서는 기존의 중앙 집중식 로그 관리 방법의 대안으로 블록체인 기술을 적용하여 클라우드 환경에서 사용자 로그 정보를 관리하는 새로운 방법을 제안한다. 제안된 방법은 사용자의 로그 정보가 클라우드 환경을 갖는 프라이빗 네트워크에 블록체인으로 분산 저장되기 때문에 외부 공격으로부터 안전하게 로그 정보를 관리할 수 있다.

키워드 : 블록체인, 웹사이트, 웹서버, 접근로그, 클라우드

1. 서 론

최근에 웹 애플리케이션은 인터넷의 사용이 일반화되고 클라우드 환경이 발전함에 따라 지속적으로 그 응용 영역이 확대되고 있다. 개인정보와 관련된 중요한 각종 문서 파일들이 클라우드 또는 웹 환경을 통하여 공유될 수 있다. 보통 웹 환경을 통하여 외부 공격이 발생하기 때문에 이러한 공격으로부터 시스템을 방어하는 것은 중요한 기술 중의 하나이다[1]. 2017년 발표된 웹 애플리케이션 보안 위협 랭킹에 따르면 인증과 세션 관리 외에도 전통적인 공격 방법인 SQL 삽입과 XSS(cross-site scripting) 공격이 1위와 3위를 차지하였다[2]. 웹서버 해킹은 전통적인 시스템 해킹기법에 비해 상대적으로 공략하기 쉬운 오픈소스 기반 웹 애플리케이션의 취약점을 이용한 애플리케이션 해킹으로 가고 있는 추세이다. 웹 애플리케이션 해킹은 웹서버에서 동작하는 애플리케이션의 구조, 논리, 코딩상의 문제점을 통해 공격하는 것이다. 일반

※본 연구는 금오공과대학교 교수연구년지원으로 수행되었음.
† 준 회 원 : 한국정보화진흥원 정보보안팀 선임연구원
†† 종신회원 : 금오공과대학교 컴퓨터공학과 교수
Manuscript Received : March 18, 2020
Accepted : May 7, 2020
* Corresponding Author : Young-Hak Kim(kimyh@kumoh.ac.kr)

적으로 웹 애플리케이션은 운영체제 및 데이터베이스시스템과 상호 작용을 통해 서비스를 제공한다. 하지만, 의도하지 않은 운영체제 명령의 수행과 트랜잭션 정보와 같은 중요한 정보의 노출 등으로 피해가 발생할 수 있다. 특히, SQL 인젝션 공격은 웹 인터페이스를 통해 데이터베이스의 중요한 정보를 노출할 수 있는 위협으로 널리 알려져 있다[3].

인터넷 상에서 이용할 수 있는 정보의 양이 증가함에 따라 인터넷의 사용자가 지속적으로 증가하고 있다. 웹을 이용하는 사용자에 관한 정보를 분석하여 웹 서비스의 질을 향상시킬 필요성이 크게 대두되고 있다. 웹 사이트의 컨텐츠(contents)도 중요 하지만 컨텐츠 사용자의 편의성을 고려해야 한다. 기존에 구축된 웹사이트를 분석하고 재구성하는 것은 이제 웹사이트 관리에 필요한 주요 요소가 되어가고 있다. 웹사이트는 사용자 중심의 매체이므로 끊임없이 사용자요구를 파악하여 수정해나가야 한다. 그러므로 웹 사용자의 성향을 파악하고 분석하기 위한 중요한 자료로 웹서버의 로그 파일을 이용할 수 있으며 웹서버의 로그 파일을 분석하기 위한 많은 연구들이 진행되고 있다[4].

웹사이트는 개인정보, 문서, 사진, 컨텐츠와 같은 중요한 데이터를 저장하고 있기 때문에 불법적인 접근의 공격으로 인해 금전적인 피해가 발생한다. 웹상에서 공유되는 데이터가 불의의 공격을 받게 되면, 관리자는 공격자를 찾기 위해서 사용자 접속 로그 파일을 분석하여 공격자를 추적한다. 하지만, 공격자가 로그 파일을 삭제하거나 임의로 내용을 변경하면 문제 발생 근원지의 추적이 불가능해진다. 현재 대부분 웹서버는 사용자 로그 정보를 중앙 집중식 방법으로 관리한다. 이 방법은 관리 측면에서 매우 단순하지만 외부 공격에 의해 임의로 조작될 수 있고 심지어 조작된 정보가 유출될 수 있다. 따라서 웹 서비스의 신뢰성을 높이기 위해서 사용자 로그 정보가 안전하게 관리되어야 한다.

최근에는 블록체인 기술을 이용하여 웹 애플리케이션 및 웹서버 관련 연구가 활발하게 진행되고 있다. 하지만, 웹사이트를 안전하게 운영하기 위한 필수 요소인 웹서버 상태 정보를 나타내는 사용자 로그 정보 관리에 대해서는 현재 블록체인 기술이 적용되지 않고 있다. 본 논문에서는 웹서버에서 발생할 수 있는 오류와 악의적인 공격으로 발생되는 로그 정보를 안전하게 보관하는 기법으로 블록체인 기반의 사용자 로그 관리 시스템을 제안한다. 제안된 방법에서 사용자 로그 정보는 클라우드 환경의 프라이빗 네트워크에 블록체인으로 안전하게 분산되어 저장된다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구 및 기존 시스템에 대해서 설명한다. 3장에서는 블록체인 기반 시스템의 설계에 대해서 소개한다. 4장은 성능 평가를 수행하고, 5장에서는 본 논문에 대한 결론을 제시한다.

2. 관련 연구 및 기존 시스템

2.1 프라이빗 블록체인

프라이빗 블록체인은 기존의 퍼블릭 블록체인 시스템에서 채굴자의 역할이었던 거래 검증을 블록의 승인권한을 가진 시스템 관리 주체가 하도록 하는 시스템이다. 블록에 대한 검증 주체를 채굴자에서 중앙의 시스템 관리 주체로 이동하면서 블록체인 시스템의 개발 목적인 탈 중앙성과 개방성은 비교적 약해진다. 그러나 프라이빗 블록체인은 기존의 블록체인 시스템과 비교해 저장 정보의 다양성과 높은 처리 성능을 제공한다. 프라이빗 블록체인은 기존의 블록체인 시스템에 필수적인 채굴 과정을 생략하기 때문에 블록이 확정되는데 걸리는 시간이 비교적 짧다. 이는 블록체인의 처리 성능과 직결된다. 또한 프라이빗 블록체인은 블록체인에 참가하는 주체를 제한할 수 있다. 따라서 열람 권한에 제한을 둬야하는 정보를 저장하는데도 사용할 수 있다[5].

2.2 클라우드 컴퓨팅

클라우드 컴퓨팅은 세 가지 서비스로 구분한다. 첫 번째 SaaS(Software as a Service)는 주로 개인과 기업을 대상으로 온라인을 통해 소프트웨어 서비스를 제공한다. 즉, 클라우드 환경에서 사용 가능한 응용소프트웨어를 사용자에게 제공하는 주문형 소프트웨어(On-demand software) 서비스로서 웹 브라우저만 있으면 전 세계 어디든 접속하여 사용가능하다. 두 번째로 PaaS(Platform as a Service)이다. PaaS는 주로 응용프로그램 개발환경을 제공하는 것이 목적이다. 최근에는 BaaS(Blockchain as a Service)로 블록체인을 서비스로 제공을 하고 있으며 대표적인 것으로 마이크로소프트 에저(Azure) 등이 있다. IaaS(Infrastructure as a Service)로 서버 운영에 필요한 서버자원, IP, Network, Storage, 전력 등 여러 인프라 서비스를 제공한다[6].

2.3 Hyperledger Fabric

Hyperledger는 2015년 12월 설립하고 2016년 1월에 정식 발촉한 Linux foundation 오픈소스 프로젝트로 프라이빗 블록체인 구현을 목표로 하고 있다. Hyperledger는 현재 IBM, 엑센추어, JP 모건 등 100여개 기업이 참여하고 있다. Hyperledger Fabric[7]은 Hyperledger의 코어 부분으로 2017년 7월에 버전 1.0의 정식 버전이 발표되었다. Hyperledger의 구조는 Membership 서비스, Application, Peer(Endorser, Committer), Chaincode, Ordering 서비스 등으로 나눌 수 있다. Membership 서비스는 신원이 확인된 참여자만 참여할 수 있도록 하는 CA(Certificate Authority) 기능을 포함하여 신원이 확인된 참여자에게 X.509 인증서를 발급해 주고 X.509인증서를 이용하여 신원을 확인한다. Application은 다양한 응용이 가능하며 Hyperledger SDK(Software Development Kit)를 통하여 Hyperledger 네트워크와 통신한다[8].

2.4 기존 로그 시스템

아파치 로그는 크게 에러 로그와 접근 로그로 구분할 수 있다. 에러 로그는 웹서버의 진단 정보와 요청을 처리하는 도중 발생한 오류를 파일에 기록한다. 서버가 기동하거나 동작하는데 문제가 발생하면 오류에 대한 내역이 에러 로그에 기

록되기 때문에 시스템 운영자는 가장 먼저 확인해야 한다. 접근 로그는 서버가 처리하는 모든 요청을 기록하며, 클라이언트가 제공한 정보는 로그 파일에 대부분 기록이 된다. 즉, 악의적인 의도를 가진 클라이언트가 로그 파일에 제어문자를 추가할 수 있으므로 로그를 이용할 때는 주의해야 한다. LogFormat을 통하여 사용자 로그에 포함된 정보를 확인할 수 있다. 접근 로그에 정보를 기록하는 것은 로그 관리의 기본이며, 정보를 분석하고 유용한 통계를 만들어 활용할 수 있다. 아파치의 로그 파일이 있는 디렉토리에 대해 root 권한이 있다면 서버를 실행하는 권한을 얻을 수 있기 때문에 안정한 관리가 필요하다. 리눅스 시스템에서는 syslog를 사용하여 다른 프로그램으로 전송이 가능하다. Fig. 1은 웹사이트 운영을 위한 필수 요소인 아파치 웹서버 로그 구조를 보여준다.

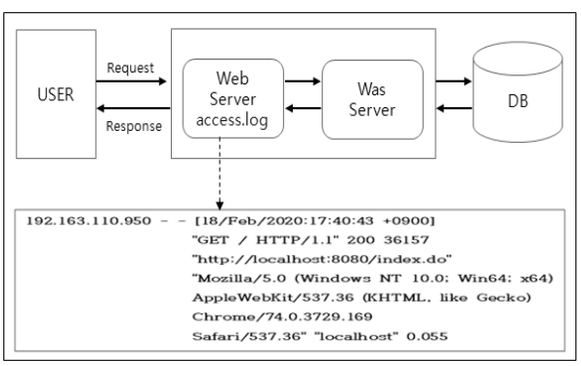

Fig. 1. Log Structure of Web Server

2.5 성능 평가 요인

블록체인에 관한 이전 연구에서 중앙집중식 서버 방식과 성능 비교를 위한 주요 요인으로 신뢰도, 보안성, 안전성 등의 항목을 사용하였다[9,11]. 본 논문에서는 제안된 블록체인 결과의 비교를 위해 신뢰도, 보안성, 안전성 이외에 분산관리, 공격 추적, 확장성 등의 항목을 기능별로 구분하여 비교 분석한다.

3. 블록체인 기반 시스템의 설계

3.1 로그 관리 시스템 설계

국가기관이나 기업체 웹사이트는 이용자들에게 유용한 정보를 제공해주는 역할을 한다. 웹사이트를 운영하기 위해서는 웹서버가 필요하며, 핵심요소인 접근 로그 파일은 사용자들이 웹사이트에 접근하는 경로 및 정보 이용에 대한 내역을 모두 포함한다. 대부분의 접근 로그 파일들은 Common log format을 따른다. 이 표준 형식에는 주로 사용자 IP주소, 접근시간, 요청방식, 접근한 페이지의 URL, 데이터 전송을 위해 사용된 프로토콜, 에러 코드, 전송된 데이터 크기 등의 정보를 포함한다. 웹서버의 접근 로그를 설정하기 위해서는 아파치 설정 파일에서 설정 가능하며, 지정된 위치에 날짜별로 생성된다. 본 논문에서는 웹서버에 악의적인 공격으로부터 로그 파일을 안전하게 관리하기 위해 프라이빗 블록체인 기술을 적용하여 Fig. 2와 같이 사용자 로그 관리 시스템을 설계한다.

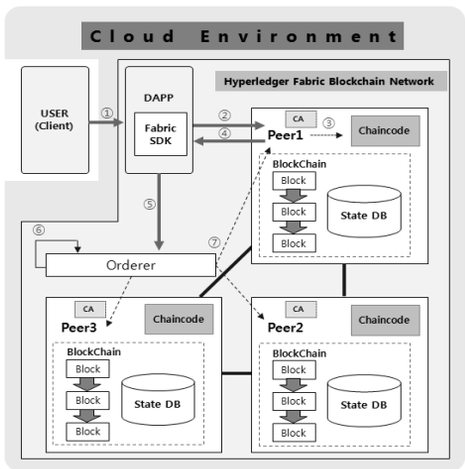

Fig. 2. Log Management System Based on Blockchain

Fabric SDK는 트랜잭션을 생성하고 체인코드 함수를 호출하는 역할을 한다. CA 노드는 디지털 인증서를 사용하여 PKI(Public Key Infrastructure) 방식으로 인증을 수행하게 된다. Peer는 원장(Ledger)의 상태를 관리하고, 체인코드는 분산원장에 데이터를 기록 및 읽기 위해서 사용한다. Orderer는 트랜잭션을 정렬한 후 최신 블록을 생성하고, State DB는 트랜잭션을 실행한 State 결과를 관리하는 저장소이다.

현재 공공기관의 웹사이트 또는 업무시스템 같은 경우 정부 지침에 따라 대부분 클라우드 환경을 통해 서비스를 제공하는 추세이다. 웹서버의 로그 파일은 일정한 기간이 경과하면 디스크 사용량이 늘어나게 되어 공간 부족 현상이 빈번하게 발생한다. 이러한 하드웨어 자원적인 문제점을 편리하게 해결할 수 있는 방법으로 클라우드 환경에서 실험을 진행하였으며, 시스템 수행 절차는 다음과 같다.

1) 웹사이트 정보 이용자로부터 시스템 접근을 통해 로그 데이터가 생성되고 트랜잭션을 요청한다.
2) Fabric SDK를 통해 Peer와 연결 요청하고, 체인코드를 실행 요청한다.
3) 로그 데이터에 대한 체인코드가 실행된다.
4) 보증 허가된(Peer 인증서) 로그 데이터를 반환한다.
5) 보증 허가된(Peer 인증서) 로그 데이터를 전달한다.
6) 보증 허가된(Peer 인증서) 로그 데이터를 최신 블록에 포함한다.
7) 로그 데이터가 기록된 최신 블록을 블록체인 네트워크의 Peer1, Peer2, Peer3에 전달하고 State DB 상태를 업데이트한다. Table 1은 Fig. 3의 수행 프로세스에 대한 기능별 역할을 보여준다.

Table 1. Role of Each Function in Fig. 3

Function	Role
USER	Users of the website
DAPP	Distributed applications used by users
Orderer	Collect transactions, create blocks, and send them to peers
Peer	Blockchain network configuration, blockchain storage and management

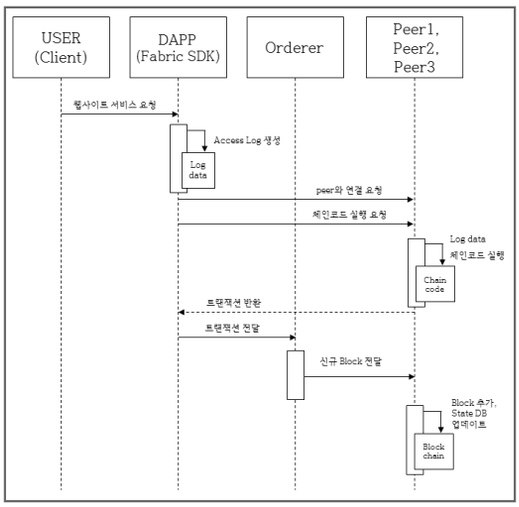

Fig. 3. Execution Process of Log Management System

Fig. 3은 제안 방식의 프라이빗 블록체인 기반 로그 관리 시스템의 수행 과정을 도식화한 것이다. 사용자가 USER를 통해 정보 이용을 위해 웹사이트에 접속하면 DAPP 분산 애플리케이션에 접근 로그가 생성되며, Fabric SDK를 통해 Peer와 연결 요청을 한다. 연결이 완료되면 해당 Peer에게 체인코드를 실행 요청하게 되고 처리된 트랜잭션이 Fabric SDK에 반환된다. Fabric SDK는 반환된 트랜잭션을 Orderer에 전달하고, Orderer는 전달된 트랜잭션을 확인하여 새로운 블록을 생성한다. Orderer는 새로 생성된 블록을 블록체인 네트워크의 각 Peer에 전달하며, 각 Peer는 블록의 검증을 통해 이상이 없는 경우 블록체인에 블록을 추가하고 State DB 상태를 업데이트한다.

3.2 로그 관리 시스템 구현

본 연구의 프라이빗 블록체인 기반의 로그 관리 시스템의 실험을 위해 Hyperledger Fabric 플랫폼을 사용하여 구현하며 개발 환경은 Table 2와 같다.

블록체인 네트워크 구성을 위해 Organization 1, Peer 3, Orderer 1, CLI, State DB와 CA 노드를 사용하여 실험 시스템을 구축하였다. 그리고 체인코드 작성을 위해 Go 언어를 사용하여 로그 데이터를 저장(set())하고, 출력(getAll()) 함수를 〈Algorithm 1〉과 같이 작성하였다.

본 연구의 테스트 데이터(로그) 생성을 위한 웹사이트는 Fig. 4와 같다. 사용자가 웹사이트에 접속하여 해당 메뉴 및 링크를 클릭할 때마다 로그 데이터가 지정된 경로에 생성된다.

실험을 위한 선행 작업으로 블록체인 네트워크를 구동한다. Fabric SDK를 통해 로그 데이터를 Endorsing Peer에 전달하고 체인코드를 수행한다. 만일 수행된 체인코드가 정확한 결과이면 자신의 디지털 인증서 값 Read/Write Set을 Orderer 노드에게 전달한다. Orderer 노드는 전달받은 트랜잭션을 정렬하여 새로운 블록을 생성하고 각 Peer에 전달한다. 이러한 과정을 통하여 Fig. 4에서 로그 데이터가 생성되며 Fig. 5와 같이 실행 결과가 각 Peer 노드에 저장되는 것을 확인할 수 있다.

Table 2. Experimental Environment

Virtualized Environment	VirtualBox 6.0
Operating System	Ubuntu 16
Development Tools	cURL 7.5
	Docker / Docker Compose 1.22
	Go Language 1.11
	Python 2.7
	Node.js 8.16 / npm 5.6
	Hyperledger Fabric 1.3
System Specification	Intel(R) Core(TM) i5-2430M CPU 2.4GHz

Algorithm 1: Chaincode Execution Function

```
// Set stores the logs on the ledger.
func set(stub shim.ChaincodeStubInterface, args []string)
    (string, error) {
    if len(args) != 2 {
        return "", fmt.Errorf("Incorrect arguments.
                    Expecting a key and a value")
    }
    err := stub.PutState(args[0], []byte(args[1]))
    if err != nil {
        return "", fmt.Errorf("Failed to set asset: %s",args[0])
    }
    return args[1], nil
}

// Get returns the value of all logs
func (t *SimpleAsset)getAll(stub shim.ChaincodeStubInterface)
    (peer.Response) {
    startKey := "Logs1"
    endKey := "Logs99"
    resultsIterator,err:=stub.GetStateByRange(startKey, endKey)
    if err != nil {
        return shim.Error(err.Error())
    }
    defer resultsIterator.Close()
    var buffer bytes.Buffer
    buffer.WriteString("[")
    bArrayMemberAlreadyWritten := false
    for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()
        if err != nil {
            return shim.Error(err.Error())
        }
        if bArrayMemberAlreadyWritten == true {
            buffer.WriteString(",")
            buffer.WriteString("\n")
        }
        buffer.WriteString("{")
        buffer.WriteString("\"")
        buffer.WriteString(queryResponse.Key)
        buffer.WriteString("\"")
        buffer.WriteString(":")
        buffer.WriteString(string(queryResponse.Value))
        buffer.WriteString("}")
        bArrayMemberAlreadyWritten = true
    }
    buffer.WriteString("]")
    fmt.Printf("- queryAllLogs:\n%s\n", buffer.String())
    return shim.Success(buffer.Bytes())
}
```

최신 블록 검증 결과는 Orderer를 통해 Committing Peer로 전달된다. 최신 블록을 받은 Committing Peer는 해당 블록을 검증하기 위해 VSCC(Validation System Chaincode) 시

Fig. 4. Web Site for Experiment

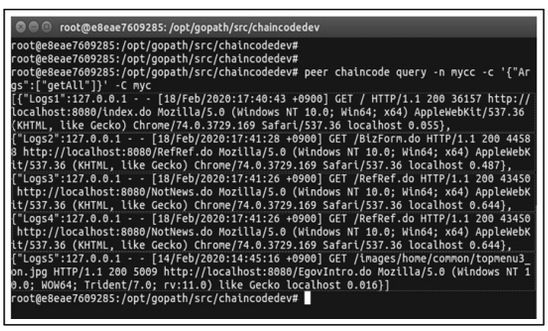

Fig. 5. Execution Result in Each Peer Node

스템 체인코드를 두 단계로 실행한다. 첫 번째는 최신 블록을 전달받은 Peer는 블록에 포함된 각각의 트랜잭션마다 보증 정책에 부합하는 Endorsing Peer의 디지털 인증서가 존재하는지 확인한다. 만약 보증 정책을 만족시키지 않는다면 해당 트랜잭션은 부적합(invalid) 판정하고 최신 블록에 트랜잭션 내용을 반영하지 않는다. 두 번째는 블록에 포함된 각 트랜잭션마다 Read/Write set 결과 값을 확인한다. Read set에 포함된 키값이 사용될 때 State DB의 레코드 버전(블록 버전)을 확인한 후, 현재 블록체인의 버전과 일치하는지 확인한다. 만약 일치하지 않으면 해당 트랜잭션은 부적합 판정을 받고 최신 블록에 트랜잭션 내용을 반영하지 않는다. 이로써 Committing Peer는 트랜잭션의 보증 여부에 따라서 트랜잭션마다 유효(valid) 혹은 무효(invalid) 태그를 표시하는 작업을 수행한다. 최신 블록 검증을 통과하면 Peer는 자신의 로컬 저장소에 저장되어 있는 블록체인에 최신 블록을 추가하게 된다. 또한, 유효 태그를 가진 트랜잭션의 내용만을 State DB에 업데이트한다[10].

4. 성능 평가

본 장에서는 기존의 중앙 집중식 로그 관리 기법과 제안된 프라이빗 블록체인 방식의 로그 관리 시스템을 비교 분석한다.

현재 웹서버 로그 파일의 경우 외부 공격자로부터 불법적인 목적으로 로그 파일을 수정하거나 삭제를 한다면, 어떤 문제가 발생된 근원지를 추적 할 수 있는 방법이 없다. 실제로 불법적인 의도를 갖고 웹서버에 공격한다면 치명적인 피해와 서비스 불가에 대한 막대한 피해가 발생하며, 웹서버의 복구와 공격자를 추적하는데 어려움이 따른다. 이러한 문제점을 해결하기 위해 본 논문에서는 웹서버의 접근 로그 데이터를 프라이빗 블록체인 기술을 통해 관리하는 기법을 제안하였다. 따라서 시스템 운영자는 블록체인을 통해 관리되는 로그 정보를 이용하여 시스템에 대한 상태를 상세히 알 수 있으며 관리할 수 있다. 또한, 블록체인의 보안성으로 인해 로그 파일의 접근 및 조작이 어려워 신뢰성 있는 정보를 제공할 수 있다. 본 논문에서 제안하는 블록체인 기반 로그 데이터의 경우 분산 저장되어 암호화 처리되기 때문에 의도적인 조작이 불가능하다.

Table 3. Comparison Result of Previous Log Management Method and Proposed Method

Classification by Function	Comparison Item	Existing Method	Proposed Method
Management Aspect	Reliability	LOW	HIGH
	Distributed Management	X	○
Security Aspect	Security	X	○
	Attack Tracking	LOW	HIGH
Expansion Aspect	Extensibility	LOW	HIGH
	Safety	LOW	HIGH

Table 3에서 기존방법은 중앙 집중식 로그 관리 방법을 의미하며 이 방법과 본 연구에서 제안한 방법의 비교 결과를 보여준다. Table 3의 비교 항목 중에서 신뢰도 및 안정성 항목은 [11]에서 언급한 결과를 참고하였다.

기존 로그 관리 기법은 중앙 시스템의 운영 서버에서 로그 파일을 관리하기 때문에 데이터 조작 등의 보안에 취약하여 불법적인 공격의 대상으로 인한 문제가 발생시, 로그 파일 분석과 추적이 어려워 웹서버 복구 시간 지연과 데이터 유실에 따른 웹사이트 신뢰도가 저하된다. 프라이빗 블록체인을 적용한 제안된 시스템의 경우 분산 방식의 로그 파일 관리로 인해 유지비용이 적고, 로그 파일의 강력한 보안으로 인한 웹사이트의 신뢰성을 향상 시킬 수 있다. 또한, 운영 중인 웹서버에 불법적인 공격을 받을 경우 블록체인에 의해 문제 발생의 근원과 공격자의 위치, 피해 내역 분석을 역추적 방법으로 관련 사항들을 확인 할 수 있다. 기존 운영 중인 웹서버 로그 관리 기법의 경우, 관리자의 관리를 통해 시스템이 운영된다. 웹사이트를 통해 정보를 이용하는 모든 사용자들의 내역들은 로그 파일에 기록되어 저장되기 때문에 일정한 기간이 지나면 로그 파일의 용량이 늘어나게 된다. 시스템 관리자는 주기적으로 운영 서버의 디스크 사용량을 점검하고 오래된 로그 파일을 제거하는 작업이 필요하다. 제안 기법에서는 이를 해결하기 위하여 클라우드 환경의 자원을 통해 디스크 용량을 유동적으로 확보함으로서 보다 안정적인 서비스를 운영할 수 있다. 본 연구에서는 프라이빗 블록체인을 통해 허가된 내부 블록체인에 로그 파일을 암호화하여 블록으로 연결하기 때문에 문제 발생시 로그 파일 분석을 통한 추적이 쉽고, 보안 측면에서 우수하다.

5. 결 론

4차 산업혁명의 핵심 기술인 인공지능과 함께 클라우드 컴퓨팅[12]은 끊임없이 변화와 발전을 이어가고 있다. 클라우드 컴퓨팅은 네트워크를 통해 인터넷에 존재하는 데이터를 다양한 장치와 브라우저를 통해서 이용한다. 다양한 장치와 브라우저를 이용하여 클라우드 환경에 존재하는 개인정보와 같은 중요한 데이터를 사용하기 때문에 제3자로부터 악용될 수 있는 위험이 존재한다. 개인정보와 관련된 중요한 정보와 기업의 핵심 정보도 클라우드 환경의 웹사이트를 통해 공유되고 있다. 웹서버의 취약한 부분을 이용한 공격이 빈번히 발생하기 때문에 불법적인 공격으로부터 방어하는 것은 중요하다. 본 논문에서는 기존 운영되고 있는 웹사이트 로그 파일의 문제점을 분석하여 로그 파일의 불법적인 접근자로부터 조작을 막는다. 또한, 클라우드 환경의 자원 활용으로 로그 파일의 저장 공간 관리의 수월함과 웹서버에 불의의 문제가 발생시 빠른 추적과 복구 및 대처가 가능하며, 웹서버를 안정적으로 관리할 수 있다. 프라이빗 블록체인의 기술을 이용하여 접근 로그 데이터를 암호화하여 불변한 로그 데이터는 악의적인 공격으로부터 빠른 대응이 가능하다. 불법적인 공격자로부터 개인정보 유출로 인해 악용하는 피해를 사전에 차단함으로서 2차 사고를 막을 수 있다. 또한, 프라이빗 블록체인 모델은 웹서버 로그뿐만 아니라 DB 로그 등 다양한 로그 관리 방안으로 활용될 수 있을 것이다.

본 논문에서는 웹서버의 접근 로그에 한정하여 프라이빗 블록체인을 이용하여 보안성을 높일 수 있도록 구현하였다. 향후에는 Dapp(Decentralized application)을 추가로 구현하여 웹사이트를 이용하는 사용자 접속에 대한 로그 트랜잭션을 자동으로 체인코드를 수행하여 일정한 개수를 모아 블록을 구성하는 기법에 대해서 추가로 연구를 진행할 계획이다.

References

[1] H. Mac, D. Truong, L. Nguyen, H. Nguyen, A. Tran, and D. Tran, "Detecting Attacks on Web Applications using Autoencoder," *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pp.416-421, 2018.

[2] H. Lee and K. Kim, "Novelty Detection on Web-server Log Dataset," *Journal of the korea Institute of Information and Communication Engineering*, Vol.23, No.10, pp.171-181, 2019.

[3] Y. Yun, J. Ryou, S. Park, and J. Park, "Profile based Web Application Attack Detection and Filtering Method," *The KIPS Transactions: Part A*, Vol.13, No.1, pp.19-26, 2006.

[4] G. Jung and N. Park, "A Study on using the Web Service by analyzing Web Server's Log File," *Journal of the Korea Society of Computer and Information*, Vol.5, No.1, pp. 35-42, 2000.

[5] J. Yun and M. Kim, "Private Blockchain and Smart Contract Based High Trustiness Crowdsensing Incentive Mechanism," *Journal of the Korea Institute of Information Security & Cryptology*, Vol.28, No.4, pp.999-1007, 2018.

[6] J. Moon, "Cloud Computing Trend and Future Directions," *The Korea Contents Association Review*, Vol.17, No.1, pp.23-26, 2019.

[7] Hyperledger Fabric [Internet], https://hyperledger-fabric.readthedocs.io/en/release

[8] S. Jung, "HyperCerts : Privacy-Enhanced OTP-Based Educational Certificate Blockchian System," *Journal of the Korea Institute of Information Security & Cryptology*, Vol.28, No.4, pp.987-997, 2018.

[9] J. Yun and M. Kim, "Private Blockchain and Smart Contract Based High Trustiness Crowdsensing Incentive Mechanism," *Journal of The Korea Institute of Information Security & Cryptology*, Vol.28, No.4, pp.999-1007, 2018.

[10] D. Yoon, Blockchain Learning with Hyperledger Fabric, Jpub, 2018.

[11] Y. Son and Y. Kim, "Design of Management System for Registering Agricultural Machine Using Blockchain," *Journal of The Korea Contents Association*, Vol.19, No.12, pp.18-27, 2019.

[12] S. Park and Y. Kim, "An Analysis of the Interaction Effect of Benefit and Cost on Cloud Computing Service," *KIPS Transactions on Computer and Communication Systems*, Vol.2, No.1, pp.27-34, 2013.

손 용 범

https://orcid.org/0000-0003-3249-1476
e-mail : sybum@nia.or.kr
2010년 금오공과대학교 컴퓨터공학과(학사)
2012년 금오공과대학교 컴퓨터공학과(석사)
2016년 ~ 현 재 금오공과대학교 컴퓨터공학과 박사과정
2019년 ~ 현 재 한국정보화진흥원 정보보안팀 선임연구원
관심분야 : 블록체인, 분산처리, 웹서버

김 영 학

https://orcid.org/0000-0003-4232-4612
e-mail : kimyh@kumoh.ac.kr
1984년 금오공과대학교 전자공학과(학사)
1989년 서강대학교 전자계산학과(석사)
1997년 서강대학교 전자계산학과(박사)
1999년 ~ 현 재 금오공과대학교 컴퓨터공학과 교수
관심분야 : 병렬알고리즘, 분산처리, 블록체인