제안한 가상화 게시판과 클라우드 서버 간의 통신 기법

김용덕, 김상욱
경북대학교
전자전기컴퓨터학부
ydkim@woorisol.knu.ac.kr

A Communication technique between Cloud Server and Proposed Virtual Board

Yongdug Kim, Sangwook Kim
School of Electrical Engineering and Computer Science,
Kyungpook National University

요 약

클라우드 컴퓨팅 서비스는 정보의 접근을 위한 시간과 장소의 경계를 허물어 산업전반의 발전을 크게 이끌고 있지만 집중된 정보를 탈취하기 위한 해킹의 시도 또한 증가하고 있다. 특히 웹 기반의 클라우드 어플리케이션의 취약점을 통한 해킹이 주로 이루어진다. 가상화 게시판은 클라우드 서버에 등록되어 있는 프로시저 호출을 통해 동작하는 클라우드 가상화 어플리케이션이다. 본 논문은 가상화 게시판과 클라우드 서버의 통신을 위한 기술인 VaaS(Virtual as a Service)를 설계한다. 가상화 게시판을 통해 기술에 대한 설명과 검증결과를 제시한 후 클라우드 가상화 어플리케이션의 보안에 대한 기대효과에 대해 논의한다.

1. 서론

오늘날 다양한 클라우드 컴퓨팅 서비스들이 인터넷상에서 제공되고 있다.[1] 클라우드 내부로 정보의 집중화는 정보의 접근을 위한 시간과 장소의 경계를 허물어 산업전반의 발전을 크게 이끌고 있지만 집중된 정보를 탈취하기 위한 해킹의 시도 또한 증가하고 있다. 클라우드 서비스는 내부 구조, 처리방식이 외부로 공개되어있지 않기 때문에 서버로의 직접적인 해킹 시도가 어렵다. 따라서 해커는 사용자들이 클라우드 서비스를 이용할 수 있도록 개발된 클라우드 어플리케이션의 취약점을 이용하여 공격 대상 시스템만을 목표로 성공할 때 까지 지속적으로 이루어지는 하는 APT(Advanced Persistent Threat)[2] 방식의 해킹을 시도한다. 특히 웹 기반으로 제작된 클라우드 어플리케이션은 인젝션, 크로스 사이트 스크립트[3]와 같은 웹 공격에 의해 권한 탈취, 페이지 변조, 서비스 거부 등 타 플랫폼에 비해 해킹 성공을 위한 노력 대비 심각한 피해를 입힐 수 있어 해킹의 주요 대상으로 이용된다.

가상화 게시판은 게시글 읽기, 쓰기, 삭제, 수정을 웹 페이지 요청을 통해 수행하던 기존의 웹 게시판과 달리 클라우드 서버에 등록되어 있는 프로시저 호출을 통해 동작하는 클라우드 가상화 어플리케이션이다. 클라우드 서버는 가상화 게시판의 비즈니스 로직에 해당하는 모든 부분을 처리하며 가상화 게시판으로부터 수신된 인 바운드 패킷, 송신하는 아웃 바운드 패킷에 대한 무결성 검사를 실시하여 해킹 공격 탐지, 중요 데이터 유출 차단을 수행한다.

본 논문은 가상화 게시판과 클라우드 서버의 통신을 위한 기술인 'VaaS(Virtual as a Service)'를 설계한다. 가상화 게시판을 통해 기술에 대한 설명과 검증결과를 제시한 후 클라우드 가상화 어플리케이션의 보안에 대한 기대효과에 대해 논의한다.

2. 관련연구

클라우드 컴퓨팅 서비스는 서비스 형태에 따라 인프라스트럭처 기반(IaaS), 플랫폼 기반(PaaS), 소프트웨어 기반(SaaS)으로 나뉜다. 구글의 GAE(Google App Engine), 아마존의 AWS(Application Web Service)는 대표적인 플랫폼 기반의 서비스로 사용자가 원하는 웹 서비스를 직접 프로그래밍 언어로 개발하여 이를 업로드 하는 방법으로 자신만의 소프트웨어 기반의 클라우드 어플리케이션을 제작 할 수 있는 환경을 제공해 준다.[4] 이와 같은 서비스는 시스템에 업로드 되는 데이터의 무결성 검사뿐만 아니라 업로드 이후에 동작하는 개별 서비스 상의 송·수신자 간의 패킷 데이터에 대한 무결성 검사가 추가로 필요하다. 클라우드 서버로의 접근은 클라우드 어플리케이션을 통해서 이루어지므로 어플리케이션 취약점을 통한 서버로의 비정상적인 접근 시도를 탐지 및 차단하기 위한 수단이 필요하다. 또한 어플리케이션을 서버와 분리된 공간에서 동작하도록 하여 어플리케이션의 취약점을 통한 해킹이 서버 내부로의 피해로 이어지지 않도록 해야 한다.

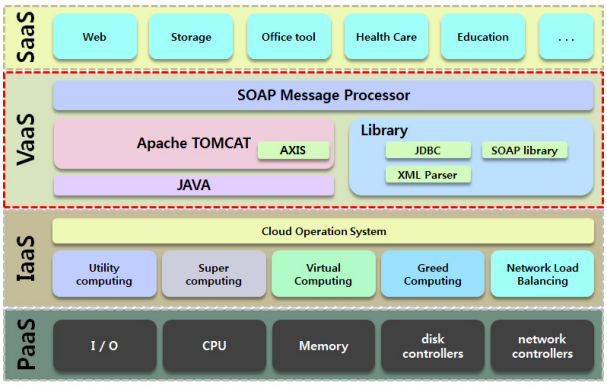
3. 가상화 게시판

클라우드 서비스의 발전으로 시설 투자비용 없이 누구나 쉽게 웹 사이트를 구축 할 수 있게 됐지만 화면 중앙에 배치된 게시판 어플리케이션의 취약점을 통한 해킹의 위협은 여전하다. 가상화 게시판은 클라우드 환경에서 웹 서비스를 이용하는 사용자들에게 보안에 특화된 게시판 어플리케이션을 제공한다. 게시글 읽기, 쓰기, 삭제, 수정을 처리하는 비즈니스 로직은 모두 서버에서 이루어지며 서버로 수신되는 인, 아웃 바운드 패킷에 대한 무결성 검사를 실시하여 해킹 공격 탐지, 중요 데이터 유출 차단을 수행한다.

기존의 웹 게시판은 게시글의 읽기, 쓰기, 삭제, 수정에 대한 요청을 PHP, JSP, ASP와 같은 서버 사이드 스크립트 언어로 작성된 웹 페이지 호출을 통해 이루어진다. 호출시 GET 또는 POST 방식으로 전달되는 변수와 값은 웹 페이지 내의 데이터베이스 질의 언어 구문에 들어간다. 이때 해커는 웹 페이지 호출 시 전달되는 변수와 값을 조작해 데이터베이스 내의 특정 테이블의 칼럼 데이터를 가져오거나 변조한다. 또는 자바스크립트를 게시글에 삽입하여 사용자 정보를 가지고 있는 쿠키를 가져오거나 악성 프로그램 설치를 유도한다. 이와 같은 해커의 악의적인 행위를 사전에 탐지하기 위해서는 웹 페이지를 요청 시 마다 전달되는 변수와 값에 대한 검증을 실시해야한다. 하지만 다수의 사용자에 대한 요청 처리, 검증을 해야 할 변수와 값 외에 HTTP 프로토콜로 전달되는 웹 소스를 전체를 수신 받아야 하기 때문에 서버의 퍼포먼스 저하로 인한 서비스 마미로 이어질 수 있다.

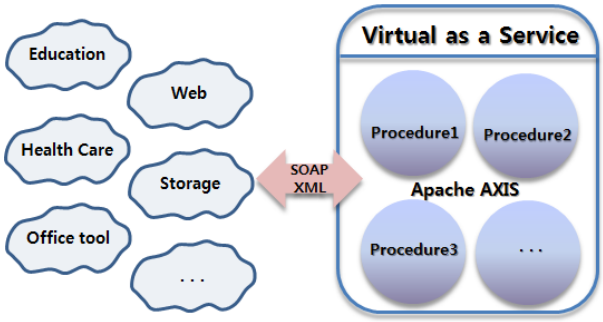
4. VaaS(Virtual as a Service)

VaaS(Virtual as a Service)는 <그림1>과 같이 클라우드 서비스 모델인 소프트웨어 서비스(SaaS)와 인프라 서비스(IaaS)의 사이에 위치하여 SaaS에서 운영되는 클라우드 어플리케이션이 IaaS에 저장되어 있는 클라우드 데이터에 접근하기 위한 기술이다.



<그림1> 클라우드 서비스 모델 내 VaaS

VaaS는 클라우드 어플리케이션을 클라이언트, VaaS내 Apache AXIS[5]를 서버로 하는 RPC(Remote Procedure Call) 환경을 제공하고 SOAP(Simple Object Access Protocol) XML 데이터를 통해 클라이언트와 서버 간 통신을 수행한다.



<그림2> 클라우드 어플리케이션과 VaaS의 통신

Apache AXIS는 JAVA언어로 작성된 프로시저를 SOAP 메시지를 이용해 클라우드 어플리케이션이 원격에서 호출 가능하도록 하는 웹 서비스를 제공한다. 프로시저는 특정 동작을 위한 함수로써 가상화 게시판의 경우 읽기, 쓰기, 삭제 수정에 관한 동작이 각각의 프로시저라고 할 수 있다. 클라우드 어플리케이션 프로그램 내에는 사용자 인터페이스만을 구현을 하고 원격지인 VaaS에 동작에 관한 프로시저를 등록함으로써 실제로는 분리된 공간에서 프로그램이 동작하지만 논리적으로는 하나인 가상화 어플리케이션이 구현된다.

4.1 클라우드 어플리케이션과 VaaS의 통신 방법

<그림 3>과 <그림 4>는 클라우드 어플리케이션과 VaaS의 통신을 위한 SOAP 메시지의 구조에 대해 설명한다. SOAP은 XML 스키마를 기본으로 하며 SOAP Body 내에 송·수신되는 데이터가 들어가며 문서 객체 모델(DOM, Document Object Model)로 표현된다. 클라우드 어플리케이션에서 VaaS에 등록된 프로시저를 호출시 SOAP 메시지는 프로시저 이름과 전달 할 변수, 값을 <그림 3>과 같은 구조로 제작하여 HTTP 프로토콜을 통해 VaaS로 전송한다.

```
POST /{VaaS URL} HTTP/1.0
Host: {VaaS Domain}
User-Agent: {User-Agent}
Content-Type: text/xml; charset=utf-8
SOAPAction: ""
Content-Length: {legnth}

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
 <{PROCEDURE} xmlns="http://system">
  <{argument1}>{value}</{argument1}>
  <{argument2}>{value}</{argument2}>
 </{PROCEDURE}>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<그림 3> VaaS로 프로시저 요청

```
HTTP/1.1 200 OK
Server: {Apache-AXIS}
Content-Type: text/xml;charset=utf-8
Date: {Date}
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
 <{PROCEDURE}Response xmlns="http://system">
  <{PROCEDURE}Return>{value}</{PROCEDURE}Return>
  <{PROCEDURE}Return>{value}</{PROCEDURE}Return>
 </{PROCEDURE}Response>
</soapenv:Body>
</soapenv:Envelope>
```

<그림 4> VaaS로부터 수신된 프로시저 응답 메시지

VaaS는 클라우드 어플리케이션으로부터 수신된 SOAP 메시지를 읽어와 프로시저 이름, 변수, 값을 분리 한 후 해당 프로시를 호출한다. 호출된 프로시저의 결과 값을 다시 SOAP 메시지로 제작해 클라우드 어플리케이션으로 전송한다. <그림 4>는 VaaS로부터 클라우드 어플리케이션이 수신 한 프로시저 응답 메시지이다.

4.2 보안 무결성 검사

SOAP 메시지를 이용한 원격 프로시저 호출 방식은 기존의 웹 페이지 호출을 통한 기능 수행 방식보다 송·수신되는 데이터에 대한 효과적인 보안 무결성 검사를 수행할 수 있다.

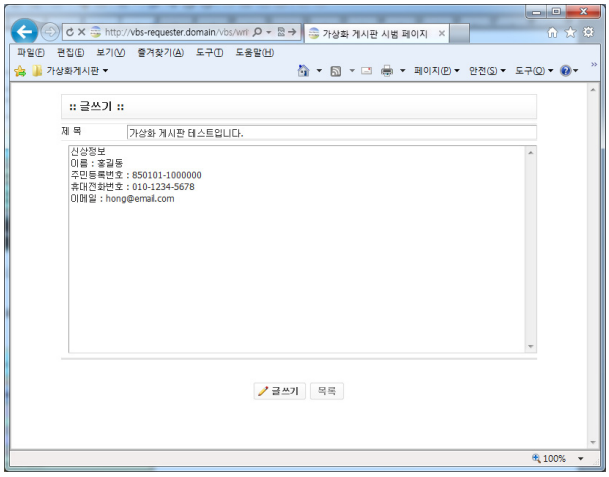
```
<html>
...
<?
validate_requestData($reqData);
..
validate_responseData(_POST['argument1']);
...
?>
...
<?
function validate_requestData($str) {
 $tag = array(",", "--", "'", "#", "&", "%", "=", "*", "/", "||", ";", ":", "<", ">");
 ...
 str_replace($tag, "", $str);
}
function validate_responseData($str) {
 $sid = "(([0-9]{6})([-|\||[:space:]]+)([0-9]{7}))|([0-9]{13}))";
 $sid_replace = "{주민등록번호 차단}";
 ...
 $str = ereg_replace($sid, $sid_replace, $str);
 ...
}
?>
...
```

<그림 5> 웹 페이지 호출 방식의 송·수신 데이터 무결성 검사

<그림 5>는 웹 페이지 호출 방식에서의 송·수신 되는 데이터 무결성 검사를 위한 PHP 웹 프로그램 소스의 일부분이다. validate_requestData()는 페이지 요청 시 데이터에 악의적인 행위를 할 수 있는 스크립트가 포함되어있는지 여부를 판단하고 제거하는 기능을 수행하며 validate_responseData()는 수신된 데이터가 외부로 유출되지 않아야 하는 데이터인지 여부를 판단, 차단하는 기능을 수행한다. 서버 환경에 따라 개발언어의 차이가 있을 뿐 프로그램의 알고리즘과 동작 결과도 같다. 하지만 데이터 무결성 검증을 위해 검증에 불필요한 html 프로그램 소스 까지 인터프리팅(interpreting) 한다는 점은 다수의 사용자가 서비스를 이용 시 퍼포먼스 저하를 가져온다. 이에 반에 VaaS의 SOAP 메시지 처리를 통한 원격 프로시저 호출 방식은 프로시저 이름, 변수, 값에 해당하는 경량의 XML 데이터를 송·수신 하므로 다수 요청에 따른 퍼포먼스 저하 현상을 해결 할 수 있다.

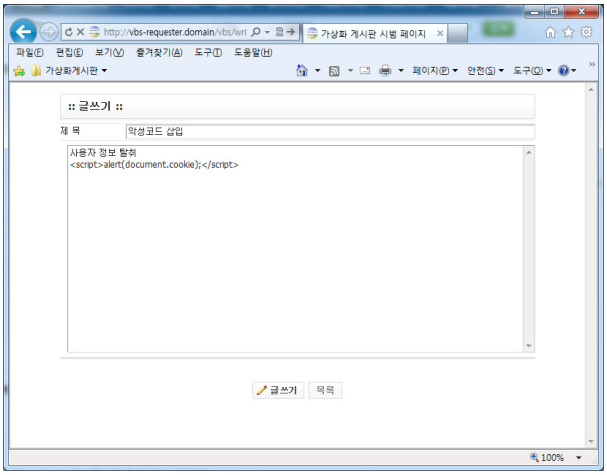
4.3 가상화 게시판을 통한 VaaS의 동작 실험

<그림 6>과 <그림 7>은 게시판에 주민등록번호, 휴대전화번호, 이메일과 같은 개인정보를 쓸 때 VaaS의 보안 무결성 검사를 통해 차단되는 과정을 보여준다.

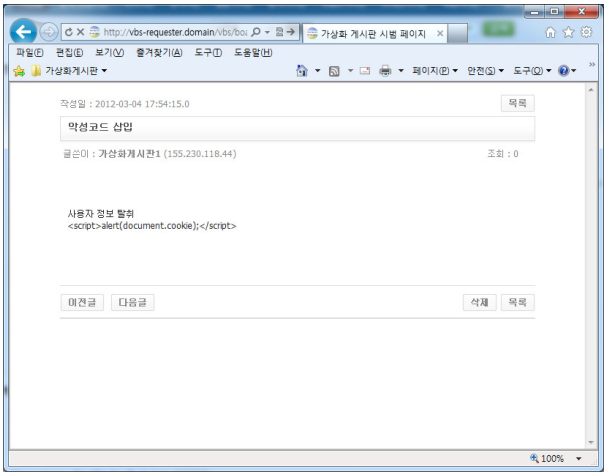


<그림 6> 중요 정보를 포함한 글

<그림 7> 중요 정보 노출 차단



<그림 8> 악성 코드 삽입 시도



<그림 8> 악성 코드 차단

<그림 7>과 <그림 8>은 게시판에 자바스크립트로 작성된 악성코드를 게시글에 포함했을 때 데이터 무결성 검사를 통한 차단이 이루어지는 과정을 보여준다. 자바스크립트는 스크립트는 일반 문자열로 치환되어 실행되지 않는다.

5. 논의

VaaS를 활용한 클라우드 가상화 어플리케이션의 효용성을 가상화 게시판을 개발을 통해 검증했다. 클라우드 서비스는 외부로부터 시스템 내부 구조가 노출 되지 않도록 사용자 환경에서의 가상화가 필요하다. 클라이언트 어플리케이션의 취약점은 접근 인터페이스에서 문제점에서 종결되어야 하며 시스템의 비즈니스 로직으로 피해가 이어지는 것을 막아야한다. VaaS를 통해 게시판 클라우드 어플리케이션 개발 외에도 오피스 작업, 교육, 건강 전반에 걸친 서비스를 개발 할 수 있다. VaaS를 활용한 원격 프로시저 호출 방식의 가상화 클라우드 어플리케이션 개발은 늘어나는 클라우드 웹 서비스를 대상으로 하는 해킹 공격에 대해 침입에 대한 탐지 및 차단, 정보 누출에 대한 탐지 및 차단을 퍼포먼스 저하 없이 수행할 수 있는 대안이다.

참고문헌

[1] “Above the Clouds: A Berkeley View of Cloud Computing,” UC Berkeley TR 2009, Feb. 2009.

[2] Smith, A. and Toppel, N., “Case Study: Using Security Awareness to Combat the Advanced Persistent Threat” Proceedings of the 13th Colloquium for Information Systems Security Education, June 2009

[3] OWASP, https://www.owasp.org/

[4] A. Lenk, M. Klems, J. Nimis, et al. What’s Inside the Cloud? An Architectural Map of Cloud Landscape. Proc. ACM/IEEE Symposium on Cloud Computing Challenges, 23-31, Vancouver, 2009.

[5] http://ws.apache.org/axis/