FC채널을 이용한 단절된 네트워크기반의 웹서비스 구현

이미경, 민성기

고려대학교 컴퓨터정보통신대학원 정보통신공학과,
고려대학교 컴퓨터정보통신대학원

Web Service using Fiber Channel in Severance Networks

Mi-Koung Lee*, Sung-Gi Min**

*Graduate School of Computer & Information Technology, Korea University
**Dept. of Computer Science & Engineering, Korea University
E-mail : lmgjae@pmo.go.kr, sgmin@korea.ac.kr

요 약

현대에서 보안의 중요성은 나날이 증가하고 있는 가운데 내부 데이터베이스의 보안을 유지하기 위해 외부망과 단절된 업무망을 유지하는 곳이 많다. 인터넷의 발달에 따른 사용자의 편의성과 각종 어플리케이션의 통합과 이 기종 플랫폼에 따른 기업간 협업문제를 해결하는데 웹서비스가 이상적인 모델로 인정받고 있다.[1] 따라서 단절된 내부망과 외부망과의 연결이 불가피하다. 이에 따라 본 논문에서는 내부망의 보안을 위해 FC채널을 이용하여 네트워크의 연결 없이 웹 서비스가 가능한 모델을 설계하고 구현하였다.

1. 서론

인터넷이 발달한 현대에 있어 해킹이나 악성코드 등으로 인한 그 폐해나 날로 심각해지고 있어 각 기업 또는 기관에서는 외부 인터넷망과 단절된 네트워크 구조로 내부자료 보안에 힘쓰고 있다.

하지만 인터넷의 발달과 서비스의 통합 등으로 이기종 플랫폼 및 어플리케이션의 이상적인 모델로 떠오르는 웹서비스를 포기할 수 없는 현실이다.

웹서비스란 패키지 형태의 소프트웨어를 대체하는 새로운 개념의 소프트웨어 서비스를 말한다. 즉, 언제 어디서나 어떠한 기기를 통해서든지 인터넷에 접속하여 필요한 소프트웨어와 데이터 파일을 자유롭게 활용할 수 있도록 해주는 것이다.

웹 서비스는 다양한 이기종의 플랫폼을 뛰어넘는 표준 프로토콜을 사용함으로써 독립적인 웹들이 서로 '연결된' 환경에서 원활한 서비스를 가능하게 해준다[2]

이런 웹 서비스를 하기 위해서는 단절된 네트워크안의 내부자료와 외부 인터넷 서비스망과의 연결이 불가피하지만 보안의 취약점을 알고도 연결하기는 쉽지가 않다. 본 논문에서는 이런 보안문제를 해결하기 위해 망의 연결 없이 스토리지의 FC(Fiber Channel)을 이용한 자료교환보안시스템을 통하여 웹서비스를 설계 및 구현하고자 한다.

2. FC(Fiber Channel)를 이용한 웹 서비스 설계

2.1 웹 서비스 개요 및 설계

웹서비스는 XML(Extensible Markup Language), SOAP(Simple Object Access Protocol), WDSL(Web Services Description Language) UDDI(Universal Description Discovery and Integration)등의 표준기술에 기반하고 있다. 현재 W3C에서는 WSDL과 UDDI를 권고안으로 채택하였다.

웹 서비스는 이기종 플랫폼에 탑재된 서로 다른 애플리케이션들 간에 데이터 통신기능을 이용하여 작업을

자동화할 수 있는 서비스 통합 기술이다. 또한 기기와 프로그램에 상관없이 SOAP 프로토콜을 통해 컴퓨터에 구현된 기능을 마치 자신의 컴퓨터에 있는 기능처럼 사용할 수 있다.

웹 서비스 아키텍쳐는 서비스 제공자(service provider), 서비스 소비자(Service Consumer), 서비스 중개자(Service Broker)와 같은 세 가지 역할들의 상호작용을 기반으로 한다.

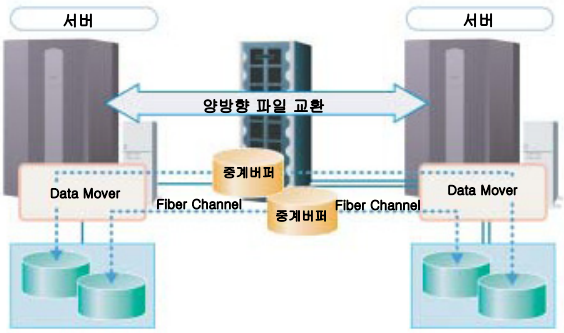
서비스 제공자는 비즈니스 관점에서 보면 서비스가 운영되는 플랫폼이다.

서비스 소비자는 비즈니스 관점에서 특정 서비스를 요구하는 비즈니스이며 아키텍쳐 관점에서는 서비스를 찾고 호출하는 애플리케이션이다. 서비스 중개자는 서비스 제공자가 출판하는 서비스 기술(Service Description)들을 관리하고 서비스 소비자에게 서비스 검색 서비스를 제공한다[3]

2.2 자료교환보안시스템의 개요 및 설계

자료교환보안시스템은 내부에서만 사용하는 단절된 사설망과 외부에 공개된 인터넷과 같은 네트워크로 분리된 시스템 간에 네트워크를 경유하지 않고 채널을 통해 상호간의 자료 교환을 실시간으로 수행하는 정보시스템이다.[4] 즉 통신망의 연결이 없이 서로 단절된 2개의 망에 있는 시스템 사이에 스토리지를 두고 FC(Fiber Channel)로 연결하여 자료를 주고받는 스토리지 공유방식을 이용한 자료교환방식이다. 스토리지의 FC를 이용한 자료교환보안시스템은 고속으로 대용량 자료처리에 적절하고 네트워크의 단절된 상태를 유지하기 때문에 보안성이 완벽하다. 당초 자료교환보안시스템은 국세청의 홈택스 시스템에 적용하기 위한 것이어서 일괄처리방식의 자료에 맞게 고안된 것으로 본 논문에서는 이 방식을 좀더 발전시켜 현 추세인 웹서비스에 적용하고자 한다.

FC를 이용한 자료교환보안시스템은 아래 그림과 같이 구성된다. 서로 다른 네트워크 사이에 디스크 어레이를 두며 각 서버에 버퍼를 할당하여 송수신 정보를 넣어두게 된다. 데이터전달을 위한 데몬이 상시 모니터링을 하다가 수신된 정보가 있을때 Data Mover라는 솔루션이 이를 두 서버의 공동 영역인 버퍼로 move하면 상대방의 데몬이 감지하여 서버로 전달하게 되는 것이다.

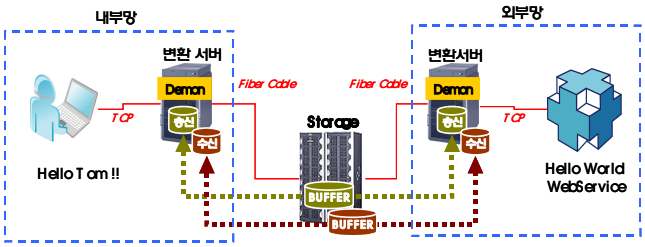


[그림 1 자료교환보안시스템 구성도]

3. FC(Fiber Channel)를 이용한 웹 서비스 구현

3.1 필요기술

단절 네트워크에서 웹서비스 구현은 HTTP 통신을 기반으로 하는데 Network 단절 구간에서는 특히 Storage를 연동한 File System을 이용한다. 즉 웹서비스의 기반이 되는 HTTP 통신을 흘러가는 패킷을 file로 변환하여 공유되는 디스크어레이의 FC통신을 할 수 있도록 역할을 하는 변환서버가 필요하게 된다. 이를 외부망과 내부망의 단절된 네트워크 구성에 접목하면 연결된 외부망의 경우, 변환서버 전단까지의 Protocol은 표준화된 HTTP Protocol을 사용하며, 변환서버단에서 이 표준 Protocol을 암호화된 File System으로 변환하여 전달하게 된다. 독립된 내부망의 경우 또한 외부망의 구성과 동일하게 가져간다. 사용자 정보 요청 형태의 종류는 Web Browser와 Web Server(또는 WAS)를 통한 방식과 직접 변환서버의 통신 Process로 요청을 하는 두 가지의 방식이 있을 수 있다.



[그림 2 자료교환보안시스템을 이용한 웹서비스의 전달]

변환서버는 외부망부와 폐쇄된 내부망부로 나뉘며, 물리적으로 양단에 각각 하나의 서버로 구축되며, 그 내부에 데이터 처리 Processor가 3가지 Type(Requester, Responser, DirectRequester)으로 운영된다.

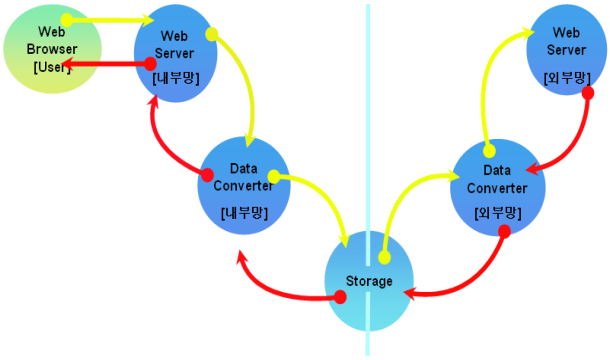
첫번째로 Type A(Requester)의 경우 외부망측

WAS와 항시 Connection을 유지하며 Async방식으로 데이터를 처리하며, 요청 데이터를 암호화된 파일의 형태로 Storage에 저장을 하고, 같은 패턴의 응답 파일이 들어 올때까지 대기하다가 응답(Responser)을 하게된다.

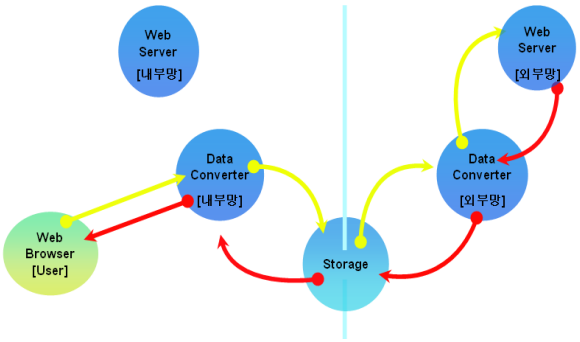
TypeC(DirectRequester)의 경우도 Type A(Requester)의 경우와 동일한 방식의 동작을 하지만, 접속 가능한 사용자의 IP Address를 등록하고 이것으로 인증받은 사용자만 접속을 허용하게 되는 부분만 차이가 있다.

Type B(Responser)의 경우는 Type A(Requester)와 TypeB(Responser)의 요청을 모두 처리하게 된다.

Storage부분에서의 데이터 전송의 형태는 Data Mover에 의해서 이루어지며, 그 방식은 다른 영역의 Partition으로의 Copy가 아니라 Cut & Paste (Move)의 방식을 기본으로 한다. 이 Move의 동작은 완전 자동으로 이루어지는 것이 아니라, Application 단에서 전송할 파일데이터 작업을 완전히 끝낸 다음에 특정 Command를 실행함으로써 동작하게 된다.



[그림 3 Requester와 Responser]



[그림 4 Direct Requester와 Responser]

3.2 구성환경

- WAS서버 : SUN unix system, Jeus
- 변환서버 : Fujitsu unix system, 3Ghz, 4G memory 변환SW(XL Data Mover)
- 디스크어레이 : 146 GB x 5, FC 4 Channel

3.3 메시지 형식

3.3.1 요청 메시지

```
GET / HTTP/1.1↓
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*↓
Referer:
http://web.xxx.com/search.xxx?sm=tab_hty&where=web kr&query=http%C↓
Language: ko↓
Accept-Encoding: gzip, deflate↓
User-Agent: Mozilla/4.0 (compatible: MSIE 6.0; Windows NT 5.1; SV1;
InfoPath.1)↓
Host: www.xxx.com↓
Connection: Keep-Alive↓
Cookie: page_uid=e3A5gloQsC0ssujLLxNsss--518761; ……↓
```

HTTP의 Request 형식은 아주 간단하다. 첫 번째 줄 처음에 서버의 어떤 기능을 이용하려는지 지정을 하며 이것이 method이다. 가장 일반적으로 쓰이고 있는 것은 GET 이다. 이것은 브라우저가 서버에게 문서를 보내달라고 요청하는 것이다. 그 다음에는 화일 이름과 위치하는 디렉토리 이름 등이 들어가는 URI를 지정하고 현재 쓰이고 있는 HTTP 프로토콜의 버전을 지정한다. 이 다음에는 MIME 형식으로 표현되는 일련의 지정사항들을 덧붙일 수가 있는데, 예를 들어 브라우저의 종류 같은 것이다. 아래와 같은 예를 볼 수 있으며, Request chain이란 여러 헤더로 구성되어 있는 요구 메시지를 일컫는 말이다

3.3.2 응답 메시지

```
HTTP/1.1 200 OK↓
Date: Mon, 31 Dec 2007 06:39:01 GMT↓
Server: Apache↓
Cache-control: no-cache, no-store, must-revalidate↓
Pragma: no-cache↓
P3P: CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN
Connection: close↓
Transfer-Encoding: chunked↓
Content-Type: text/html; charset=euc-kr↓
```

HTTP에서의 응답 형식도 아주 간단하게 구성되어 있다. 서버에서 쓰이고 있는 프로토콜 버전, Request에 대한 실행 결과 코드 및 설명문이 있으며, 전달해줄 데이타의 형식, 데이타 길이 등과 같은 추가적인 정보가 MIME 형식으로 표현되어 있다. 이어서 마지막에 헤더 정보의 끝을 나타내는 빈줄이 들어가고, 뒤이어 실제 데이타가 전달

된다. 데이타 전달이 끝나면 서버는 연결을 끊는다. Response chain이란 여러 가지 헤더로 구성되어 있는 응답 메시지를 일컫는 말이다.

3.4 데이터 속성

기본적인 전송 Data Format은 HTTP Protocol이며, Data Packet 과 Data File의 형태로 양방향 변형이 일어난다. Storage 연동부에서는 암호화된 File의 형태로 전달되며,

전송된 이후에는 중복 Data 또는 보안상의 이유로 모두 삭제가 이루어지기 때문에, 사실상 Storage 상에서 File의 형태로 존재하는 시간은 극도로 짧다.

또한, HTTP 통신 메커니즘은 Web Browser에서 접속을 시도하고, 요청한 데이터에 대한 응답을 받으면, 접속을 종료하기 때문에, 서버단에서 사용자에게로 Data를 전송할 수 있는 방법이 없다. 따라서 Data 보관 또는 재전송 무의미하다.

(1) 자료 요청 / 응답 및 오류 검증

① 요청/응답/오류 메시지를 기본적으로 Console Window에 보여주게 되어있다.

② 요청/응답 메시지는 해당 일자의 Packet Log File에 저장이되며, 확인이 가능하다

③ 오류 메시지의 경우, Process의 Stack Trace 메시지를 화면과 Log File에 자동 기록되며, 확인 가능하다.

3.5 성능

Data 변환 및 전송 서버의 경우, 해당 서비스 요청 Data를 수신한 후, File의 형태로 Packet을 변환하여 전송하고, File 형태의 응답을 각 Thread 마다 모니터링하며 대기해야하는 구조이므로, 접속자 및 서비스 요청 증가에 따른 Thread가 증가하게 된다.

통신량과 스레드 증가로 인한 부하는 L4 Switch를 이용하여 네트워크의 부하를 분산시키고, 그 다음으로 변환서버 데몬상에서의 Job 분배로 부하를 분산시킬 수 있다.

서비스 증가에 따른 통신량 및 Thread 증가의 경우, Hardware 증설과 L4 Switch를 이용한 Load Balancing 접속자 및 요청 Data를 분산하고 , L4 Switch를 이용한 환경에서 변환서버 자체적으로는 파일의 이름을 가지고 분산 처리 할 수 있도록 구성할 수 있다. 이때 같은 네트워크 내의 변환서버 데몬은 동일한 이름의 요청파일 이름을 사용할 수 없다.

각 패턴의 파일이름에는 프로세서 번호가 부여되고 (RQRA01.packet), 응답시에도 이 요청파일에 대응되는 파일이름 (RSRA01.packet) 으로 응답을 받도록 한다.

아래 그림에서 보듯이, 현재 변환서버와 스토리지의 환경이 변경되어야 하며, 요청시에는 각 데몬마다 각기 틀린 이름의 파일을 사용하며, 응답처리시에도 설정파일에 등록되어있는 패턴의 파일이름만을 가져와 처리하도록 구성하면, 여러요청을 분산하여 처리할 수 있게 된다.

4. 결론

본 논문에서는 네트워크가 단절된 환경에서 보안을 유지하면서 웹 서비스를 구현 방안을 도출하고 실제 구현해 보았다. 이는 웹 서비스의 기반 기술인 HTTP 서비스의 전송 패킷을 파일로 변환하는 부분의 개발이 필요하며 기관이나 기업의 특성 상 보안을 위해 외부망와 단절된 환경에서도 현재 개발 추세인 웹 서비스가 가능할 수 있었다. 하지만 디스크 어레이의 FC를 이용함으로써 서버 대 서버 통신이 아닌 사용자가 변환서버에 직접 요청하는 Type C(Direct Request)의 경우 속도가 많이 늦는 단점이 있어 이는 향후 연구 해야 할 과제이다

[참고문헌]

[1] 전자정부 웹 서비스 상호운용성 프레임워크 방안 연구 한국전산원 2004.9

[2] 웹 서비스를 이용한 공공기관 우편번호 지원시스템 구현 이광영 한밭대학교 2006.2

[3] 모바일 환경에서 인증과 음성인식을 위한웹 서비스 구현 고유정 한밭대학교 2004.2

[4] 스토리지를 이용한 네트워크단절기반의 자료교환 보안시스템 정영삼 숭실대 정보과학대학원, 2005.2