안전한 도메인 네임 시스템에서의 인증기능을 가진 메시지 처리기 구현

○이만희, 임찬순, 석우진, 함영환, 변옥환
mhlee, csim, wjseok, yhham, ohbyeon@garam.kreonet.re.kr
한국전자통신연구원

An Implementation of Message Processing System for Authenticating in Secure DNS

○Man-Hee Lee, Chan-Soon Lim, Woo-Jin Seok, Young-Hwan Ham, Ok-Hwan Byeon
Electronics and Telecommunications Research Institute

요약

안전한 도메인 네임 시스템은 인터넷에서 호스트의 주소와 이름을 매핑해주는 분산 시스템인 도메인 네임 시스템에 보안 기능을 강화한 것이다. 본 연구에서는 구조체를 이용하여 도메인 네임 시스템에서 사용되는 메시지를 처리하는 동시에 이 메시지에 첨부되어 있는 인증 정보를 인증할 수 있는 메시지 처리 시스템을 구현하였다. 이 메시지 처리 시스템을 이용하여 안전한 도메인 네임 시스템의 구현이 쉬워질 것으로 예상된다.

1. 서론

도메인 네임 시스템(Domain Name System, DNS)은 인터넷에서 널리 사용되는 서비스이다. DNS는 시스템에 할당되어 있는 32비트 크기의 인터넷 주소와 암기하기 쉬운 호스트의 이름을 상호 매핑해주는 전세계적인 분산 시스템이다[7,8]. 그러나 인터넷 사용이 확대됨에 따라 DNS 서비스의 중요성 또한 증대되고 있지만, 오히려 인증기능이 없는 DNS의 보안 취약성을 이용한 해킹 사고가 증가하고 있다.

이를 막기 위해, IETF의 DNS Security 워킹 그룹은 인증기능을 가지는 Secure DNS(SecDNS)를 제안하였다. 이 모델은 DNS 서버와 네임 정보를 요청하는 프로그램인 리졸버간의 인증된 통신, DNS 서버간의 안전한 존 파일 전송, 공개키의 안전한 분배 기능 등이 정의되어 있다[1,2].

한편, DNS 서버로 가장 널리 쓰이는 프로그램은 BIND다. 이 프로그램은 초기 미국 버클리 대학에서 작성된 이후 현재는 Internet Software Consortium에서 개발을 관장하고 있다[9]. 그런데 BIND의 입,출력 DNS 메시지 처리에 있어서 그 메시지의 포인터를 거의 모든 함수에서 직접 사용 및 변경함으로써 프로그램의 복잡성을 더하고 있다. 이를 해결하기 위해 본 저자는 이미 구조체를 이용한 DNS 메시지 처리기를 제안하고 구현하였다[3].

본 연구에서는 SecDNS에서 보다 쉽고 효율적으로 인증처리를 하기 위해 이 DNS 메시지 처리기에 인증기능을 추가하였다. 이 처리기는 SecDNS 서버 메인 모듈과 외부 네트워킹 모듈 사이에 위치하여 SecDNS 메시지를 서버 메인 모듈에서 조작이 용이한 데이터 구조체로 변경하는 동시에 메시지에 첨부되어 있는 인증정보를 인출하여 인증처리를 수행한다. 또 서버 메인 모듈에서 작성한 데이터 구조체

를 DNS 메시지로 변경하면서 인증정보도 자동적으로 첨부한다. 본 처리기를 사용함으로써 서버 메인 모듈에서 메시지 처리 부분과 인증처리 부분이 분리되어 메인 모듈의 추가 및 변경이 매우 용이하게 된다. 그러므로 본 처리기는 안전한 네이밍 서비스를 위한 효율적인 메시지 처리기로 사료된다.

2. 안전한 도메인 네임 시스템

DNS Security 워킹 그룹은 안전한 네임 서비스를 보장하는 DNS 모델을 제안하였다[1]. 이 모델은 공개키 분배 기능, 데이터 인증 기능, 트랜젝션과 리퀘스트 인증기능을 제공한다. 인증 방법의 핵심은 SIG RR(Signature Resource Record)이다. DNS 는 프로토콜은 네이밍 정보의 단위로 RR 을 사용한다. 각 RR 은 한 호스트의 특정 정보를 나타내는데 예를 들면, hpnl.hpcnet.ne.kr 의 A RR 과 NS RR 은 각각 Address 인 134.75.30.249 와 Name server 인 ns.hpcnet.ne.kr 을 저장한다. 이를 네트워크 상에서 안전하게 전송하기 위해 각 RR 마다 부가의 SIG RR 을 작성한다. 인증 단계는 서버가 어떤 RR 을 DNS 응답으로 보낼 때, 해당 SIG RR 도 함께 응답 메시지에 포함시켜 보내고 응답을 받은 리졸버 또는 서버는 송신받은 RR 을 SIG RR 로 인증 검사를 한다(그림 1). 이때 사용되는 서버의 공개키 분배 방법에 대한 연구는 활발히 진행 중이다[4,5].

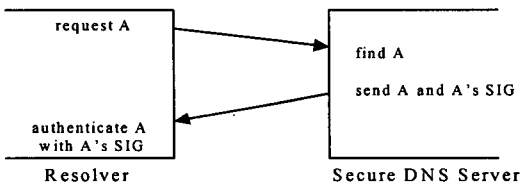

그림 1 안전한 도메인 네임 시스템

3. 인증 기능이 추가된 메시지 처리기 모델

DNS 의 메시지 형태는 헤더와 질의 부분을 제외한 질의에 대한 응답을 쓰는 부분인 answer 부분, answer 내용의 권한 및 네임 서버를 쓰는 authority 부분, authority 내 호스트의 주소등을 쓰는 additional 부분으로 나뉘어져 있다(그림 2).

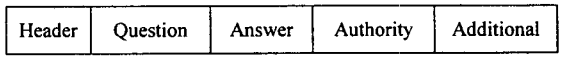

그림 2 DNS 메시지 포맷

이 메시지를 효율적으로 처리하기 위해 제안된 것이 구조체를 이용한 메시지 처리기로써 각 부분에 저장된 RR 을 구조체 형태로 저장하고 처리하는 모델이다[3]. 그리고 SecDNS 에서는 각 RR 에 대해 SIG RR 이 존재하므로 이를 메시지 처리기로 파싱하면 각 SIG RR 은 일반 RR 의 하위에 바로 연결된 형태로 나타난다(그림 3). 이때 인증 함수를 사용하여 SIG RR 에 포함되어 있는 signature 로 RR 의 유효성을 검사한다. 이로써 DNS 메시지 처리기를 통과한 메시지 정보들은 무결성이 보장된다.

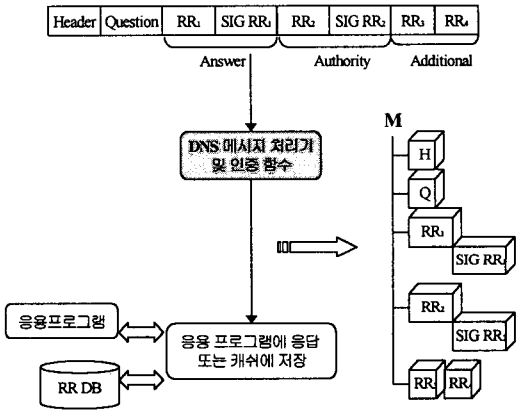

그림 3 인증기능이 추가된 구조체 메시지 처리기

4. 인증 기능을 가진 메시지 처리기 구현

본 절에서는 구조체를 이용한 인증 기능을 가진 메시지 처리기 구현 방법을 설명한다. 본 구현에 사용된 자료구조는 구조체 메시지 처리기를 위해 임찬순, 이만희에 의해 정의 및 구현되었다[3,6]. 다음 표 1 은 메시지 처리기에 인증기능을 추가하기 위해서 사용된 인증 관련 함수와 간추린 소스 코드이다.

표 1 인증을 위한 구조체 메시지 처리 함수

함수	기능
RR_Extract	SIG 를 포함한 각 RR 을 type 별로 구조체로 변경
Write_SIG	SIG RR 을 DNS 메시지로 쓰기
Serialize_Rdata	각 RR 을 인증하기 위해 RR 과 SIG RR 을. 연결하여 스트링화
RR_Verify	Serialize_Rdata 를 호출하여 SIG RR 과 RR 을 스트링화하고 SIG RR 을 생성한 signer 의 공개키로 인증 검사

```
int RR_Extract( R, Rtype, msg, eom, cp, length)
{
        u_char *oldcp;
        int     n,len;
        char tempbuf[PACKETSZ],tempbuf1[PACKETSZ];
        oldcp = cp;
        switch(Rtype) {
                :
          case SIG:
            R->Rdata_sig  =  (  Rdata_SIG  *)malloc(  sizeof
(Rdata_SIG) );
            GETSHORT(R->Rdata_sig->type, cp);
            R->Rdata_sig->algorithm = *cp++;
            R->Rdata_sig->labels = *cp++;
            GETLONG(R->Rdata_sig->ori_ttl, cp);
            GETLONG(R->Rdata_sig->sig_expire, cp);
            GETLONG(R->Rdata_sig->sig_inception, cp);
            GETSHORT(R->Rdata_sig->key_tag, cp);
            n = dn_expand(msg, eom, cp, tempbuf, sizeof tempbuf);
            R->Rdata_sig->signer_name = strdup(tempbuf);
            cp += n;
            strncpys(tempbuf, cp, length-(cp-oldcp));
            tempbuf[length-(cp-oldcp)] = '\0';
            len = decode64(tempbuf,tempbuf1,88);
            R->Rdata_sig->signature = strdup(tempbuf1);
            cp += length-(cp-oldcp);
            return cp - oldcp;
                :
}}
int RR_Verify(Rdata *R)
{
        int seriallen,namelen,len,ret;
        char serial[PACKETSZ],vbuf[PACKETSZ],fullname[128];
        KEY *zkey=NULL;

        if (R->Rdata_sig == NULL) return 0;
        seriallen = Serialize_Rdata(serial,PACKETSZ,R);
        strcpy(fullname,R->rname->name);
        namelen = strlen(R->rname->name);
        zkey = dnssec_readpubkey(R->Rdata_sig->signer_name);
        if ((ret = dnssec_verify (SIG_MODE_INIT, fullname, namelen,
                zkey, R->Rdata_sig->signature, 64)) ||
           (ret = dnssec_verify (SIG_MODE_FINAL, serial, seriallen,
                zkey, R->Rdata_sig->signature, 64)))  {
            return -2; /*Verify failed*/
        }
        return 1;
}
```

그림 4 RR_Extract, RR_Verify 함수 소스

그림 4 에서 DNS 서버 또는 리졸버는 응답 메시지 처리시 RR_Extract 를 호출하여 각 RR 을 구조체 형태로 읽어 들인다. 이때 SIG RR 이 읽혀지면 RR_Verify 를 호출하여 RR 과 SIG RR 쌍의 무결성을 검사한다. Serialize_Rdata 는 인증함수의 입력으로 사용하기 위해 RR 와 SIG RR 을 연결하는 함수로써 도메인 네임에 압축하지 않은 형태로 대문자를 쓰지 않고 RR 과 SIG RR 을 연속적으로 스트링화한다[1]. 이 함수들을 사용하여 서버, 리졸버는 단순히 메시지 처리기를 호출하므로써 무결성이 보장된 DNS 메시지의 구조체를 넘겨 받을 수 있으므로 인증을 위한 추가의 모듈이 필요 없어지는 장점이 있다.

5. 결론

본 연구에서는 기존 BIND 의 복잡성의 원인이었던 메시지 처리 방법을 개선한 동시에 인증 기능이 추가된 구조체 메시지 처리기를 구현하였다. 이 방법은 DNS 메시지를 구조체로 저장하여 RR 처리를 쉽게 하고 SIG RR 을 파싱하면서 SIG RR 의 정보를 이용하여 무결성 검사를 하므로써 이 메시지를 사용하는 서버 또는 리졸버에 무결성이 보장된 정보를 제공한다. 차후 이 처리기는 안전한 네임 서버와 리졸버에 사용될 예정이여 향후 연구 과제로는 네임 서버를 사용한 Public Key Infrastructure 구현과 IPv6 지원을 위한 메시지 처리기를 구현하는 것이다.

참고문헌

[1] D. Eastlake, "Domain Name System Security Extensions", RFC2535, March 1999.

[2] D. Eastlake, "Secure Domain Name System Dynamic Update", RFC2137, April 1997.

[3] M.H Lee, C.S Lim, W.J Seok, Y.H Ham, and O.H Byeon, "An Implementation of Message Processing System using the Structure in Secure DNS", Proceedings of The 11th KIPS Conference, 1999.

[4] C.S Lim, O,H Byeon, and Y.C Sim, "Using DNS as a Certificate Repository in the Public key Infrastructure", Proceedings of IASTED International Conference on Parallel and Distributed Computer and Networks, 1998.

[5] C.S Lim, O,H Byeon, and Y.C Sim, "A Public Key Infrastructure based on the Secure DNS", Proceedings of International Computer Symposium, 1998.

[6] C.S Lim, H.W Park, O.H Byeon, and S.Y Park, "A Design and Implementation on Zone file Editor for Secure Naming Service", Proceedings of The 10th KIPS Conference, 1998.

[7] P. Mockapetris, "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

[8] P. Mockapetris, "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[9] http://www.isc.org/