F05-03-G0005 DNS안전운용가이드

1. 문서정보

작성자 : 공개SW 기술지원센터 이장원
작성일 : 2005년 10월

2. 목차

3. 개 요

이문서는 DNS를 안전하게 운용 하기위한 문서이다.

4. DNS 안전운용가이드 소개

DNS는 인터넷 기반 구조에 있어서 핵심적인 역할을 하기 때문에 안전한 운용을 요구하는 매우 중요한 요소이다. 그러나 예전부터 최근까지 DNS 관련 취약점들이 지속적으로 발견되고 있고, 최근에는 이러한 취약점을 공격하는 코드가 공개되어 있어 실제 관련 해킹사고도 많이 발생하고 있다. 또한 DNS는 각 조직의 필수적인 서비스로 Firewall에서 흔히 개방해놓는 포트이기 때문에 공격자로부터 주요 공격목표가 될 수 있어 더욱 주의가 필요한 서비스 이다.

본 가이드에서는 DNS 운용 시 발생할 수 있는 보안 취약점에 대하여 먼저 알아보고 이에 대한 해결책을 소개한다. 본 가이드에서 특별한 언급이 있지 않는 한 DNS라 함은 ISC(Internet Software Consortium, http://www.isc.org)의 BIND 8.x 버전을 의미하며 이 가이드를 읽는 독자는 기본적인 DNS 동작 원리와 서버 설정에 대해 이해 하고 있다고 가정한다.

BIND는 두 가지 버전으로(4, 8번대) 발전해 왔는데, 현재는 8번대 버전만이 지속적으로 기능이 확장되고 있다. 따라서 본 문서에 8버전대 이상의 BIND를 중심으로 설명한다.

5. DNS 보안 취약점

가. Zone Transfer

DNS 서버 관리에 있어서 가장 흔히 잘못 설정하는 것 중의 하나는 바로 불필요한 사용자에게 DNS Zone Transfer를 허용하는 것이다. 해당 도메인의 Zone에 대한 복사본을 얻기 위해 Primary Name Server로부터 Zone 데이터베이스를 끌어오는 작업을 Zone Transfer라 하는데, 이는 Primary Name Server가 다운될 경우 Secondary Name Server가 그 역할을 대신하게 되기 때문에 양쪽 서버간의 정보를 일관성 있게 유지시키기 위해 수행되는 작업이다. 이 작업은 주로 Secondary Name Server 측에서 이루어지며, 때때로 얼마나 많은 수의 호스트가 등록되어 있는지 혹은 Zone의 문법적 오류를 검사하기 위해 관리자가 수동으로 조작하기도 한다. Zone Transfer는 Authority를 갖는 DNS 서버에 직접 질의하여야 하므로, nslookup 상에서 해당 DNS 서버로 질의 서버를 변경한 후, ls 명령을 사용하면 된다. 일반적으로 Zone Transfer는 Primary Name Server와 Secondary Name Server의 Zone 정보를 일관성있게 유지하기 위해 이루어 지기 때문에 Second Name Server에서만 Zone Transfer를 할 수 있도록 하면 된다.

그러나 허가되지 않는 사용자에게 Zone Transfer를 허용할 경우 DNS 서버의 중요한 정보가 유출되게 된다. 즉, 공격자는 전송 받은 Zone 정보를 이용하여 호스트 정보, 네트워크 구성 형태 등의 많은 정보를 파악할 수 있게 된다. 대부분의 사이트에서 DNS 서버를 디폴트로 설치할 경우 임의의 사용자가 Zone Transfer 를 할 수 있도록 설정된다. 다음은 nslookup 명령을 이용하여 DNS 서버의 Zone 데이터를 수집하는 것을 보여준다.

```
[bash]$ nslookup
Default Server : xxx.co.kr
Address :   10.10.20.2
>> set type=any
>> ls –d xxx.co.kr >> /tmp/zone_out
[bash]$ more zone_out
   xxx.co.kr.                SOA     db.xxx.or.kr administrator.xxx.co.kr.
   (85 3600 600 86400 3600)
   xxx.co.kr.                NS      www.xxx.co.kr
   xxx.co.kr.                NS      db.xxx.co.kr
   xxx.co.kr.                MX      21      mail.xxx.co.kr
   db                            A       10.10.20.1
   intra                         CNAME oa.xxx.co.kr
   mail                          A       10.10.20.5
   mail                          MX      22      mail.xxx.co.kr
   monitor                       A       10.10.20.4
   ns                            A       10.10.20.1
   …
   xxx.co.kr.                SOA     db.xxx.co.kr administrator.xxx.co.kr.
   (85 3600 600 86400 3600)
```

(그림 1) Zone Transfer 예제

(그림 1)과 같이 공격자는 xxx.co.kr 도메인에 대한 Zone Transfer를 통해 내부 시스템에 대한 정보를 파악할 수 있게 된다(nslookup의 ls 명령 이외에도 dig에서 axfr 옵션을 통해서도 Zone Transfer를 요청할 수 있다). db 라는 호스트 이름은 아마도 데이터베이스 서버일 가능성이 높으며, intra 라는 이름의 서버는 사내 인트라넷 서버로 추측할 수 있을 것이다.

사실 DNS의 Zone 정보에 대한 접근을 제한하는 것은 가장 효과적인 해킹 예방 방법중의 하나이다. 대부분의 공격도구(주로 취약점 스캐너)는 공격하고자 하는 시스템의 IP 리스트를 얻기 위해 Zone Tranfer를 이용한다. 즉, 특정 사이트의 도메인에 대하여 Zone 정보를 수집해서 나온 호스트 IP에 대하여 공격을 시도하게 된다. 반대로 말하면 Zone 정보를 가능한 외부에 노출시키지 않으면 그만큼 공격을 당하지 않게 되는 것이다.

나. Dynamic Updates

BIND-8 버전부터 지원되는 Dynamic Update기능은 관리자가 직접 해당 도메인 네임서버의 Zone 파일을 수정하지 않고도 동적으로 Zone 파일의 레코드를 원격 갱신할 수 있도록 해준다. 도메인 관리를 자동화 하거나, 사용자별로 접속 도메인을 실시간 변경하여 제공하거나, DHCP에서의 주소-IP 매칭과 같이 실시간으로 레코드를 변경 또는 갱신할 필요가 있는 서비스에 특히 유용한 기능이다. Dynamic Update는 보안상의 이유로 기본적으로 설정되어 있지 않기때문에 이를 허용할 도메인에 대해서는 allow-update 옵션을 추가해야 한다. Dynamic Updates 기능은 잘 사용할 경우 유용할 수 있지만, 그렇지 않을 경우에는 Updator에 의해 Zone 내부의 레코드 정보가 삭제되거나 변경될 수 있는 위험성이 있다.

다. DNS 코드 취약점

DNS 트래픽은 일반적으로 Firewall을 통과할 수 있도록 설정되어 있어 공격자는 충분히 이를 악용할 수 있다. 따라서 최근에는 이를 이용하는 공격방법이 많이 공개되고 있으며, 실제 해킹에 사용되고 있다. 리모트에서 공격자가 DNS 서버로 하여금 공격자가 원하는 코드를 실행시키거나 루트 쉘을 얻게 하는 일반적인 방법은 DNS 서버에게 교묘히 조작된 패킷을 보냄으로써 버퍼 오버플로우를 발생시키도록 하는 것이다. 이러한 공격의 예로는 최근에 보고된 BIND 8.2.3 이전 버전의 TSIG 취약점과 BIND 8.2/8.2.1 버전에 대한 NXT Record 취약점이 있다.

BIND 취약점 리스트 : http://www.isc.org/products/BIND/bind-security-19991108.html

BIND 8.2/8.2.1 버전의 NXT Record 취약점에 대한 공격은 (그림 2)~(그림 4)과 같이 이루어진다. (그림 2)에서는 먼저 공격하려는 DNS 서버에 대한 버전을 확인하며, (그림 3)에서는 nmap등으로 확인된 DNS 서버 호스트 정보를 이용하여 adm-nxt 공격 도구(DNS 포트 53번으로 바인딩 됨)를 실행시킨다. Adm-nxt 공격 도구는 NXT Record와 관련된 교묘히 조작된 패킷(공격자가 실행하기를 원하는 코드가 포함된 패킷)을 상대방 DNS 서버로 전송함으로써 해당 DNS 서버로 하여금 조작된 NXT Record 패킷의 검사 과정에서 버퍼 오버플로우를 일으키도록 한다. 그리고 이 때 공격자가 실행하기를 원하는 코드가 BIND 실행 권한으로 실행되게 된다. (그림 4)는 피해시스템의 DNS 서버가 공격자 호스트의 도메인 정보에 대해 질의를 하도록 하는 것이다. 즉 피해 시스템의 DNS서버가 공격자 시스템으로 질의를 보내면 53번 포트에서 대기하고 있는 adm-nxt 공격 도구는 그 응답으로서 교묘히 조작된 패킷을 보내어 공격하게 되는 것이다.

```
[tsunami]# dig@10.1.1.100 version.bind chaos txt
; <<>> Dig 8.1 <<>> @10.1.1.100 version.bind chaos txt
; (1 server found)
;; res options : init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:10
;; flags: qr aa rd ra; QUERY: 1 ANSWER: 1, AUTORITY: 0, ADDITIONAL: 0
;; QUERY SECTION:
;;          version.bind, type = TXT, class = CHAOS
;; ANSWER SECTION
VERSION.BIND.      0S  CHAOS  TXT  "8.2.1"
```

(그림 2) BIND 버전 확인

```
[tsunami]# adm-nxt
Usage: ./adm-nxt architecture [command]
Available architectures:
   1: Linux Redhat 6.x       - named 8.2/8.2.1 (from rpm)
   2: Linux SolarDiz's non-exec stack patch - named 8.2/8.2.1
   3: Solaris 7 (0xff)       - named 8.2.1
   ...
[tsunami]# adm-nxt 1
VERSION.BIND.      0S  CHAOS  TXT  "8.2.1"
```

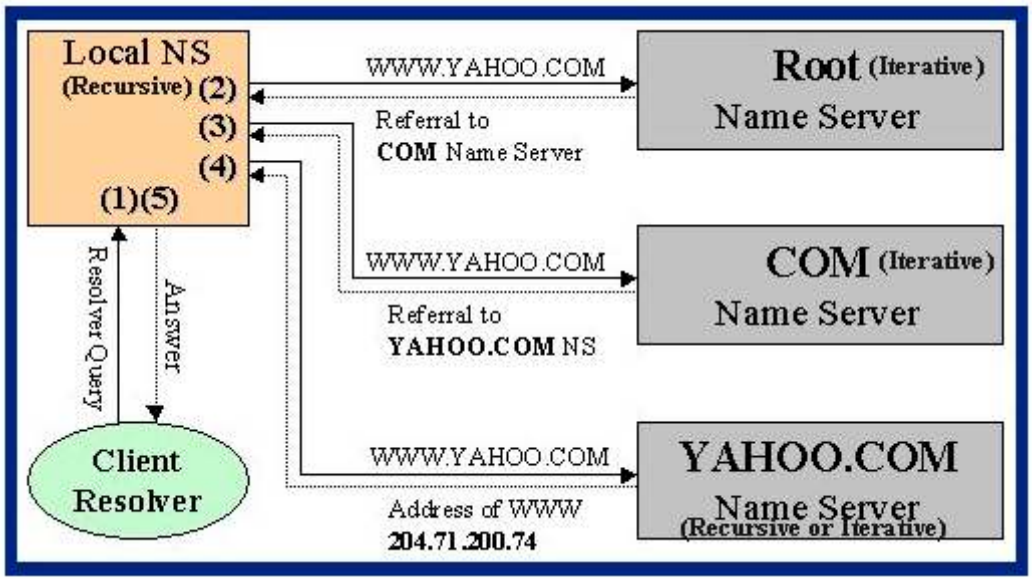
(그림 3) adm-nxt BIND 8.2/8.2.1 NXT Record 취약점 공격 도구


```
[quake]# nslookup
Default Server: localhost.attackers.org
Address: 127.0.0.1
> server 10.1.1.100
Default Server: dns.victim.net
Address: 10.1.1.100
>hash.attackers.org
server: dns.victim.net
address: 10.1.1.100
```

(그림 4) dns.victim.net 호스트에서 attackers.org 호스트로의 정보 요청

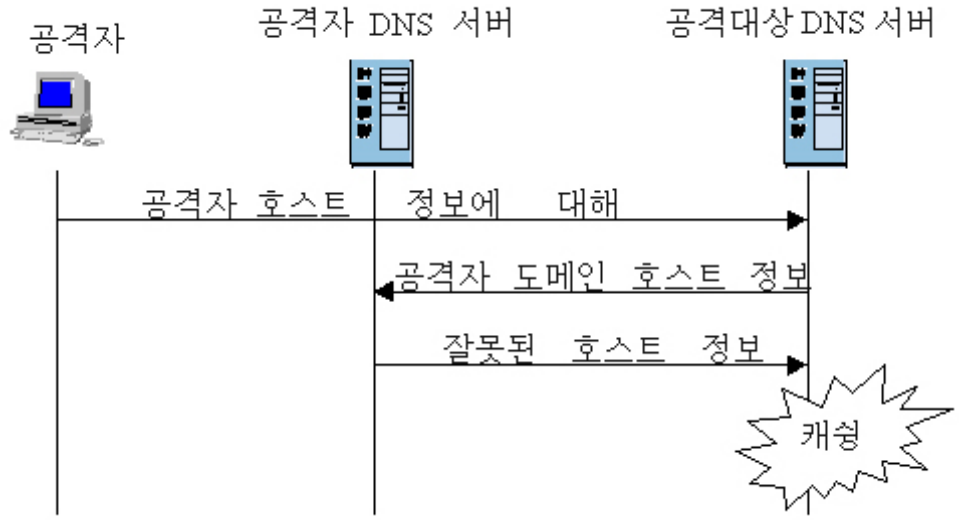
라. Address Spoofing

DNS 쿼리는 Recursive 방식과 Iterative 방식으로 전달될 수 있다. DNS 서버가 Recursive 모드로 동작할 때에는, (그림 5)와 같이 클라이언트(Resolver)의 요청에 대해 Namespace를 검색한 후 결과를 전달하고 성능상의 이유로 인해(동일 요청에 대한 재검색 오버헤드를 줄이기 위해) 검색 결과를 캐쉬한다. 하지만 Iterative 모드에서는 알 수 없는 질의(자신이 관리하지 않는 도메인에 대한 요청)에 대해, 응답 가능한 DNS 서버의 목록을 전달한다. 대부분의 DNS 서버는 Recursive 모드로 동작하며, Iterative 모드는 루트 서버와 같이 DNS 서버를 위한 DNS 서버(DNS 서버간의 통신에는 Iterative 모드가 사용됨)에서 과다한 트래픽을 막기 위해 사용한다. 또한, 클라이언트는 Iterative 모드로 설정된 DNS 서버를 사용할 수 없으므로, DNS 서버 목록(예:resolv.conf, 윈도우의 DNS 찾기목록)에 추가해서는 안 된다.



(그림 5) Recursive 모드와 Iterative 모드 예

Address Spoofing의 경우에는 검색 결과를 캐쉬하고 있는 Recursive 모드의 DNS 서버에서 주로 발생할 가능성이 있다. (그림 6)과 같이 공격자는 nslookup을 실행시키고 질의 대상 DNS 서버를 Recursive 모드로 동작하고 있는 DNS 서버로 설정한다. 이후 공격자는 공격대상 DNS 서버로 하여금 미리 잘못 설정해둔 자신의 DNS 서버로 호스트 정보를 검색하도록 하고, 공격자는 그 응답으로 거짓 정보를 제공한다. 그러면 공격대상 DNS 서버는 검색 결과를 캐쉬하게 되며, 이후 해당 도메인의 호스트들은 위조된 캐쉬 결과를 사용하여 인터넷을 사용하게 된다.



(그림 6) Recursive 모드 DNS 서버의 캐쉬 문제

이러한 공격을 통해 공격자는 유명 전자상거래 사이트와 똑같은 내용으로 자신의 사이트를 만들어 놓고, 일반 사용자가 접속하도록 유도한 뒤 사용자의 패스워드나 ID 등의 정보를 유출하는 공격을 수행할 수 있다. 사용자는 자신이 해커 사이트에 접속한지도 모른체 자신의 모든 거래정보를 입력하게 될 것이다.

6. DNS 보안 설정

DNS 서버를 안전하게 운영하기 위해서는 단순히 서버를 설치하는 것 이외에 보안과 관련된 많은 설정들을 필요로 한다. 다음은 DNS 서버의 안전한 설치 및 보안설정에 대하여 설명한다.

가. Zone Transfer 방지

BIND-8 버전에서는 "named.conf" 설정파일에서 다음과 같이 allow-transfer 설정을 사용하여 Zone 데이터를 획득할 수 있는 시스템을 제한할 수 있다. 다음은 111, 112번 호스트에서만 Zone Transfer 를 허용하도록 하는 설정이다.

```
options {
        allow-transfer { 10.10.10.111; 10.10.10.112;};
};

* 특정 도메인의 Zone에 대해서 제한할 경우에는
다음과 같이 설정할 수도 있음

zone "xxx.co.kr" {
      type master;
      file "db.xxx.co.kr";
      allow-transfer { 10.10.10.111; 10.10.10.112; };
```

(그림 7) allow-transfer를 이용한 Zone Transfer 제한 설정

Zone Transfer 를 제한하는데 있어 주의할 것은 Secondary 서버에서도 이를 설정해야 한다는 것이다. 이럴 경우 allow-transfer {none;} 과 같이 Zone 데이터 유출을 허용하지 않는 것이 바람직하다.

BIND-8.2 이상 버전에서는 Zone 데이터에 대한 인증과 검증을 위한 Transaction Signature (TSIG)라는 확장된 기능을 제공하는데, 이를 이용하여 Zone Transfer를 제한할 수도 있다.

TSIG 기능은 Primary master name server와 slave name server에 암호키를 설정하고, 이들 DNS 서버가 통신할 때 해당 키를 사용하여 데이터를 서명함으로써 서버간의 인증을 제공한다. (그림 8)은 Primary master name server가 10.10.10.178 주소를 가진 DNS 서버에 대해 응답이나 Zone Transfer를 할 때 huskymo-tornado라는 키를 사용하여 서명하도록 명시하고 있다. 또한 10.10.10.178로부터의 응답에 대한 서명도 요구하고 있다. 따라서 해당 키를 가지고 있지 않은 호스트로부터의 Zone transfer는 거부되게 된다.

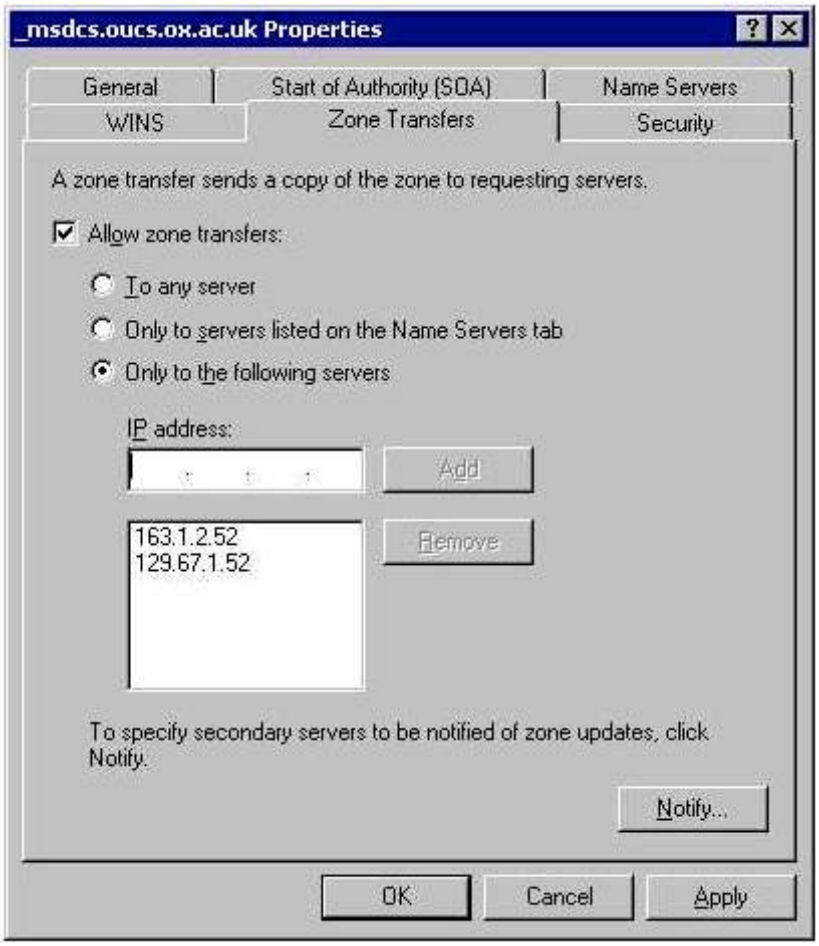
```
key huskymo-tornado. {
        algorithm hmac-md5;
        secret "mZiMNOUYQPMNwsDzrX2Enw==";
};
server 10.10.10.178 {
        transfer-format many-answers;
        keys { huskymo-tornado. ; };
};
zone "xxx.co.kr" {
        type master;
        file "db.xxx.co.kr";
        allow-transfer { 10.10.10.178; };
};
```

(그림 8) Primary master name server의 TSIG 설정

```
key huskymo-tornado. {
        algorithm hmac-md5;
        secret "mZiMNOUYQPMNwsDzrX2Enw==";
};
server 10.10.10.250 {
        transfer-format many-answers;
        keys { huskymo-tornado. ; };
};
zone "xxx.co.kr" {
        type slave;
        file "bak.xxx.co.kr";
        allow-transfer { none; };
};
```

(그림 9) Slave name server의 TSIG 설정

MS Windows 2000 DNS Server의 경우 Microsoft Management Console에서 Services and Application₩DNS₩[server_name]₩Forward Lookup Zones₩[zone_name] | Properties에서 (그림 10)과 같이 설정한다. MS Windows 2000 DNS Server은 디폴트로 모든 서버에게 Zone Transfer를 허용하게끔 설정되어 있으므로 반듯이 다음과 같이Zone Transfer를 제한 시켜야 한다.



(그림 10) Microsfot DNS Zone Transfer 제한 설정

나. Dynamic Updates 설정

Dynamic Updates를 제한하기 위해서는 IP 주소를 이용하여 제한하거나 또는 TSIG Key를 이용하여 제한하는 방법이 있다. 만일 IP 주소를 사용한다면 먼저 라우터나 Firewall에서 IP Spoofing 공격을 차단하도록 설정하여야 한다. 왜냐하면 IP 주소를 이용한 인증은 언제든 spoofing 공격을 이용하여 우회할 수 있기 때문이다. Dynamic Updates 기능은 BIND-8 이상의 버전에서만 사용가능하며, 디폴트로 허용되지 않기 때문에 특별히 설정할 필요는 없으며, 필요할 경우에만 주의하여 설정하면 된다.

```
zone "xxx.co.kr" {
        Type master;
        File "db.xxx.co.kr";
        Allow-update { 10.10.10.178; };
};
```

(그림 11) allow-update 를 이용한 Dynamic Updates 허가 설정

Top

다. DNS 코드 취약점 보안

DNS는 인터넷의 기반이 되는 주요서버로서 매우 매력적인 공격대상 중의 하나이다. 현재 지속적으로 DNS 관련 취약점이 발견되고 있으며, 이를 이용한 침입이 이루어 지고 있다. 공격자는 취약점(버그)을 이용하여 원격에서 임의의 코드를 실행시키거나, 서비스거부공격을 하거나 또는 쉘을 획득하여 시스템에 침입하기도 한다. 그리고 무엇보다 다른 취약점이 언제든

발견될 수 있으므로 DNS 서버 운영환경을 안전하게 구성하여야 한다. 다음은 DNS 서버 자체의 취약점에 대비하여 안전하게 서버를 운영하는 방법에 관한 것이다.

1) DNS 서버를 사용하지 않는다면 이를 실행하지 않거나 아예 제거한다.

2) 항상 최신의 버전의 BIND 를 사용한다.

- http://www/isc.org

3) named를 실행할 때 관리자 권한(root 권한)이 아닌 별도의 사용자 계정으로 실행시킨다.

BIND에서는 이를 위한 실행 옵션을 가지고 있다. 다음과 같은 옵션으로 named를 실행하게 되면, 처음에 53번 포트로 바인딩할 때는 관리자 권한으로 실행되며, 이후에는 명시된 user_id와 group_id로 실행되게 된다. 만일의 경우 named 서버가 공격을 당하더라도 낮은 권한을 갖게 되므로 그만큼 피해를 줄일 수 있게 된다.

```
[dns]# named –u user_id –g group_id
```

(그림 12) named 실행 권한 변경 옵션

4) Chroot (Change Root Directory) 환경에서 named를 실행시킨다.

```
[dns]# named –u user_id –g group_id –t /chroot/named
```

(그림 13) named chroot 변경 옵션

위와 같은 chroot 환경에서 named 프로세스를 실행하게되면, named는 /chroot/named 디렉토리를 시스템의 루트 디렉토리로 여기게 되고 따라서 /chroot/named 하위 디렉토리에 대해서만 접근할 수 있게된다. 이는 만일 서버의 취약점으로 인하여 침입을 당한 경우, 제한 영역의 시스템 디렉토리에만 그 피해를 국한시키기 위한 메커니즘이다. (그림 14)~(그림17)까지는 BIND-8.2.3 버전의 named 프로세스를 chroot 환경(Jail, 감옥이라고도 부름)에서 실행시키기 위한 과정을 기술한 것으로서 세부 사항은 참고문헌[3]을 참고하기 바란다.

1. Creating a User (BIND 의 실행 계정)
/etc/passwd 파일 설정 추가 *named:x:200:200:NameServer:/chroot/named:/bin/false*
/etc/group 파일 설정 추가 *named:x:200*

2. Directory Structure (BIND 실행에 필요한 디렉토리 구조 생성)

```
/chroot
    +---named
        +---bin
        +---dev
        +---etc
            +---namedb
        +---lib
        +---var
            +---run
```

3. Placing the BIND Data (기존의 BIND 데이터 이동)

```
# cp –p /etc/named.conf /chroot/named/etc/
# cp –a /var/named/* /chroot/named/etc/namedb/
# chown –R named:named /chroot/named/etc/namedb
# chown named:named /chroot/named/var/run
```

4. System Support Files (시스템 파일 복사)

```
# cd /chroot/named/lib
# cp –p /lib/libc-2.*.so .
# ln –s libc-2.*.so libc.so.6
# cp –p /lib/ld-2.*.so .
# ln –s ld-2.*.so ld-linux.so.2
# cp /sbin/ldconfig /chroot/named/bin/
# chroot /chroot/named /bin/ldconfig –v
# mknod /chroot/named/dev/null c 1 3
# cp /etc/localtime /chroot/named/etc/
# echo `named:x:200:` > /chroot/named/etc/group
```

5. Logging (로그 디렉토리 설정)

```
daemon syslogd –m 0
->daemon syslogd –m 0 –a /chroot/named/dev/log
# /etc/rc.d/init.d/syslog stop
# /etc/rc.d/init.d/syslog start
```

(그림 14) Preparing the Jail

1. Modifying Paths(BIND 소스 파일 수정)
src/port/linux/Makefile.set 파일에서
DESTRUN=/var/run 을 DESTRUN=/chroot/named/var/run 으로 수정
Src/bin/named/named.h 파일에서
#include "pathnames.h" 아래에 #define _PATH_NDCSOCK "/var/run/ndc" 추가

2. Doing the Build (BIND 컴파일)

```
# make clean
# make depend
# make
```

(그림 15) Compiling BIND

```
1. Installing the Tools Outside the Jail(BIND 인스톨)
# make install
```

```
2. Installing the Binaries in the Jail(Chroot 된 디렉토리에 실행 파일 복사)
# cp src/bin/named/named /chroot/named/bin
# cp src/bin/named-xfer/named-xfer /chroot/named/bin
```

```
3. Setting up the Init Script (init 스크립트 설정, Redhat 6.0 의 경우)
.....
[ -f /chroot/named/bin/named ] || exit 0
[ -f /chroot/named/etc/named.conf ] || exit 0
# See how we were called.
case "$1" in
  start)
        # Start daemons.
        echo -n "Starting named: "
        daemon /chroot/named/bin/named -u named -g named -t /chroot/named
        echo
        touch /var/lock/subsys/named
.....
```

```
4. Configuration Changes (named.conf options 섹션에 디렉티브 추가 설정)
directory "/etc/namedb";
pid-file "/var/run/named.pid";
named-xfer "bin/named-xfer";
```

(그림 16) Installing Your Shiny New BIND

```
1. Launching BIND(BIND 실행)
# /etc/rc.d/init.d/named start
```

(그림 17) Launching BIND

라. Address Spoofing

DNS를 아무런 제한 없이 Recursive 모드로 동작하게 되면 spoofing 공격에 취약하게 된다. 따라서 DNS를 설정할 때, 다음과 같은 보안설정을 하여야 한다.

. recursion 모드를 해제한다.
. named가 응답할 query를 제한한다.
. named가 응답할 recursive query를 제한한다.

1) recursion 모드 해지

BIND 8.x 버전에서는 (그림 18)과 같이 recursion 모드를 해지할 수 있다. 이는 해당 DNS가 다른 DNS나 클라이언트를 대신해서 도메인에 대한 질의를 수행하지 않도록 한다. **하지만 이는 클라이언트의 요청을 지원하지 못하므로, 클라이언트에게 DNS 서비스를 제공하거나, forwarder로 사용될 경우에는 이러한 방식으로 설정하지 못한다.**

```
options {
        recursion no;
};
```

(그림 18) Iterative 모드 설정 방법

Microsoft DNS Server의 경우에는 레지스트리 에디터(regedit.exe)를 이용하여 (그림 19)과 같이 Iterative 모드로 설정할 수 있다.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Parameters
레지스트리 패스에서 NoRecursive 항목을 1로 설정한다.

(그림 19) MS Windows DNS Iterative 모드 설정 방법

만약 클라이언트에게 DNS 서비스를 제공해야 하는 경우(**대부분의 경우**이다)에는, Recursive 질의를 허용해야만 하는데, 이 경우에는 DNS 서버가 받아 들일 수 있는 요청을 제한할 수 있도록 설정하여 사용하면 된다. 즉, 특정 주소로부터 온 질의에 대해서만 DNS 서버가 응답을 하도록 설정하거나, 특정 Zone에 대한 요청에만 응답하도록 설정할 수 있다.

일반적으로 DNS 서버가 관리하는 Zone에 대한 요청은 외부나 내부 모두에게서 발생할 수 있으며, DNS 서버가 관리하지 않는 Zone에 대해서는 내부에서만 발생되게 된다. 이는 (그림 20)와 같이 "allow-query"를 사용하여 제한할 수 있다.

```
acl internal { 10.10.10.176/24; };
options {
        directory "/var/named";
        allow-query { internal; };
};
zone "xxx.co.kr" {
        type master;
        file "db.xxx.co.kr";
        allow-query { any; };
};
```

(그림 20) DNS 서버로의 질의 제한 설정

이는 내부 클라이언트(resolver)에게는 모든 DNS 질의를 허용하며, 해당 DNS 서버가 관리하는 "xxx.co.kr" Zone에 대한 질의는 내부 외부 모두에게 허용한다는 설정이다. 이러한 설정은 BIND 8 버전 대에서만 가능하다.

특히, BIND-8.2.1 이상 버전에서는 기본적으로 Iterative 모드로 동작하고 있는 상태에서 특정 IP 주소에 한하여 Recursive 질의를 받아들일 수 있도록 설정할 수 있다. 이는 "allow-recursion"을 사용하여 다음과 같이 보다 간편하게 설정할 수 있다.

```
acl internal { 10.10.10.176/24; };
options {
        directory "/var/named";
        allow-recursion { 10.10.10.176/24; };
};
zone "xxx.co.kr" {
        type master;
        file "db.xxx.co.kr";
};
```

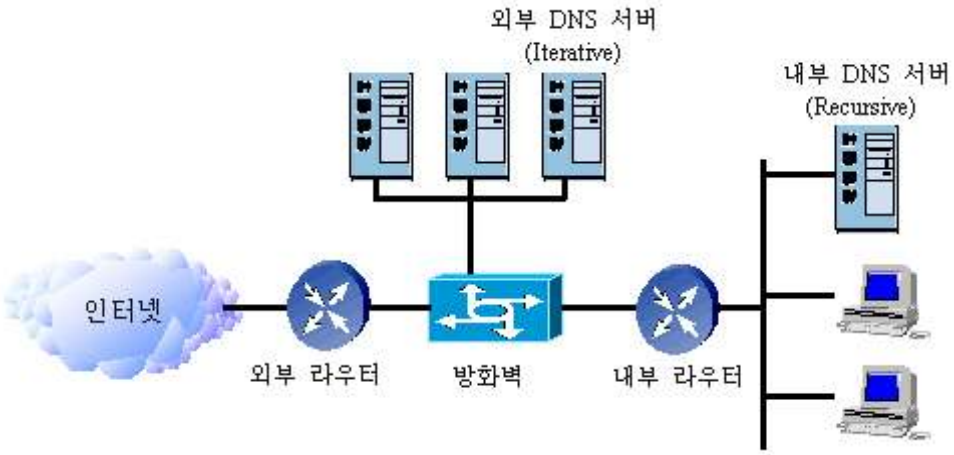
(그림 21) Recursive 질의 제한 설정

7. 안전한 DNS 구성

각 기관의 환경에 따라 내부의 DNS 데이터가 외부의 모든 사람에게 공개하기에는 다소 민감한 경우가 있을 수 있으며, 내부 호스트와 외부 호스트를 구분하여 제공하는 정보의 수준 및 내용을 차별화 하기를 원하는 경우, 또는 내부 DNS 서버만으로는 제대로 운용하기 어렵고 외부에 좀 잘 유지된 정보를 제공하기를 원하는 경우에는 보통 2개의 DNS 서버를 사용하게 된다.

내부 DNS 서버는 Recursive 모드로 동작하면서 내부 클라이언트들의 질의에 서비스를 제공하고, 외부 DNS 서버는 Iterative 모드로 동작하면서 각 기관이 관리하는 Zone에 대해서만 외부 DNS로부터의 질의 요청에 서비스를 제공하도록 설정한다. 이러한 설정은 외부에서 Zone 정보를 가져간다 하더라도, 외부에 서비스하는 정보만을 유출시키며, 내부 시스템에 대한 정보는 획득할 수 없도록 한다.

또한 Firewall에서 내부와 외부 DNS 각각에 대하여 접근제어를 함으로서 한층 더 보안을 강화할 수 있게 된다. 즉, 외부 DNS 서버가 침입을 당하더라도 내부시스템까지는 공격을 당하지 않도록 설정하는 것이다. 하나의 DNS만을 운영하게 되면, 해당 DNS 서버에 내부시스템에 대한 정보가 포함될 수 밖에 없으며, 시스템이 침입당했을 경우, 내부시스템까지 위험하게 되는 것이다. 사실 2개의 DNS 서버를 이용하기 위해서는 firewall의 기능과 함께 통합되어야 한다. 보다 자세한 내용은 "Building Intenet Firewall(인터넷 방화벽 구축하기)" 서적을 참고하기 바란다.



(그림 22) 2개의 DNS 운용 형태

```
acl slaves { 10.10.10.3; 10.10.10.1 };
options {
        directory "/var/named";
        recursion no;
        fetch-glue no;
        allow-query {any; };
};
zone "xxx.co.kr" {
      type master;
      file "db.xxx.co.kr";
      allow-transfer { slaves; };
};
```

(그림 23) 외부 DNS 서버 설정 예


```
acl internals { 10.10.10.168/24; };
options {
        directory "/var/named";
        recursion yes;
        allow-query {internals; };
};
zone "." {
        type hint;
        file "db.cache";
};
zone "xxx.co.kr" {
    type slave;
    masters { 10.10.10.2; };
    file "bak.xxx.co.kr";
    allow-transfer { internals; };
};
```

(그림 24) 내부 DNS 서버 설정 예

8. DNS의 확장된 보안기능

DNS의 확장으로서 제공되는 DNSSEC(DNS Security) 기능은 DNS 메시지에 공개키 기반의 전자서명 기능을 제공한다. 이는 DNS 데이터에 대한 응답이 누구로부터 왔는가에 관련한 인증 메커니즘을 제공한다.

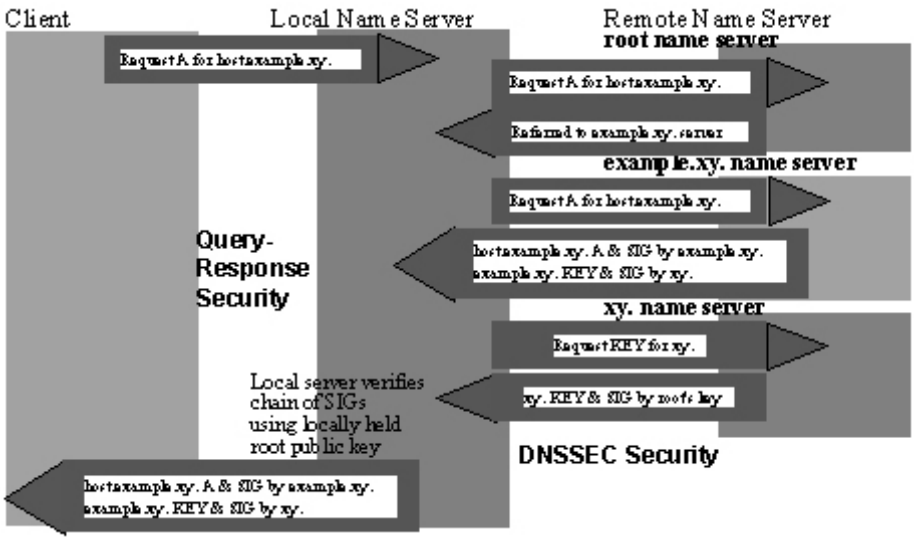
DNSSEC은 이와 같은 인증을 위해 2개의 새로운 레코드 타입인 KEY와 SIG 레코드 타입을 추가했다. KEY 레코드는 DNS 서버가 관리하고 있는 Zone을 위한 공개키를 저장하고 있으며 SIG 레코드는 Zone 내부의 각각 레코드에 대한 서명 결과를 (그림 25)에서와 같이 가지게 된다. 서명된 Zone에 대한 질의는 (그림 26)와 같은 과정으로 처리된다.

DNSSEC의 경우 IETF RFC 및 Draft로 표준화 작업이 진행 중이며 BIND-9버전 대 이후에 잘 구현되어 있다. 그러나 현재 널리 사용되고 있는 BIND-8 버전 대와의 공존 상황에서 DNSSEC 기능은 제한적으로 사용되고 있는 실정이다.

```
@ 14400 IN    SOA  test.netsec.tislabs.com. lewis.tislabs.com. 2000020701 1D 1H 1W 4H
  14400 IN    SIG   SOA 3 3 14400 20000306184745 20000207184745 48320 tise.cairn.net.(
              AIrF9Hb/BzoZ3xirlK8lxLzprIEVBFZLEE6Sqy8HlaU5r3ux
              VfEbcTA= )
  14400 IN    KEY   0x4100 3 3 (
                 ALb/qZQ/oVHyotuSbBWI1N+OYwRLv5RMc7XOb0wYE/tY02qF
                 Uf+9czS0B7pU2jYppF7RxL8b/OcWG3iAzaztzq6S0ZoQIh8J
                 M5LummzJiNl3agdbxUZH6pvmPNuiMbGl++2tUks+MAallpUz
                 4tEJPeBF+Zj8boYvWhQDaV6nwUY6kIrqRqhvAmOZHqtqzFT6
                 SdA07usEZEzZkXXS6PIg6JcN7mNhUa0qkDSNTIkrHWNCh++G
                 56dtKNsk4qn3ESreg/S2BRGWQ2/7X0PjMyBkDefvdIsw )
  14400 IN    SIG   KEY 3 3 14400 20000225145656 20000128145656 48320 tise.cairn.net.(
                 AKM6fdJmcV3Wec7sYKR5ktX2C3kWTLTcITD4iBP2rJVSFlKx
                 nsi3bRI= )

active 14400 IN CNAME active.netsec.tislabs.com.
  14400 IN SIG   CNAME 3 4 14400 20000225145656 20000128145656 48320 tise.cairn.net.(
           ACqtgIY8TkwTw83rQmt3fOPOx+TmpcCtCzl+EsFmYybcSYOlhP2Nht4= )
.....
```

(그림 25) DNSSEC에서의 KEY 레코드와 SIG 레코드



(그림 26) DNSSEC을 사용했을 때의 쿼리 처리 과정

9. 기타 유용한 문서

1. Cricket Liu, “Securing an Internet Name Server” http://www.acmebw.com/papers/securing.pdf
2. Rob Thomas, “Secure BIND Template Version 2.0”, http://www.cymru.com/~robt/Docs/Articles/secure-bind-template-20.html
3. Scott Wunsch, “Chroot-BIND HOWTO” http://www.losurs.org/docs/howto/Chroot-BIND.html
4. DNSSEC FAQ http://www.nominum.com/resources/faqs/dnssec-faq.pdf
5. Brian Wellington, “Network Security, Domain Name System(DNS) Security" http://www.pgp.com/research/nailabs/network-security/an-introduction.asp
6. Matt Larson, Cricket Liu, “Using BIND: Don’t get spoofed again” http://www.sun.com/sunworldonline/swol-11-1997/swol-11-bind.html
7. Edward Lewis, “DNS Security Extensions” http://download.nai.com/products/media/pgp/ppt/RIPE37_9122000_Intro.ppt
8. RFC 2535, “Domain Name System Security Extensions”
9. 김승영, “Powered by DNS” http://www.kr.freebsd.org/doc/PoweredByDNS/PoweredByDNS-3.4.1.html
10. Joel Scambray, Stuart McClure, George Kurtz, “Hacking Exposed, 2nd”, Osborne/McGraw-Hill, 2000
11. Zwicky Cooper, Chapman, “Building Internet Firewalls 2nd”, O’Reilly, 2000
12. W. Richard Stevens, “TCP/IP Illustrated, Voulme1”, Addison-Wesley, 1994
13. http://securityportal.com/cover/coverstory19990621.html
14. http://www.nominum.com/resources/faqs/bind-faq.html
15. http://www.isc.org/products/BIND/bind-security-19991108.html

