DNS 이중화 구성 및 보안설정

2024. 05

강사 : ㈜엔에스컨설팅 석원진

교육채널 안내

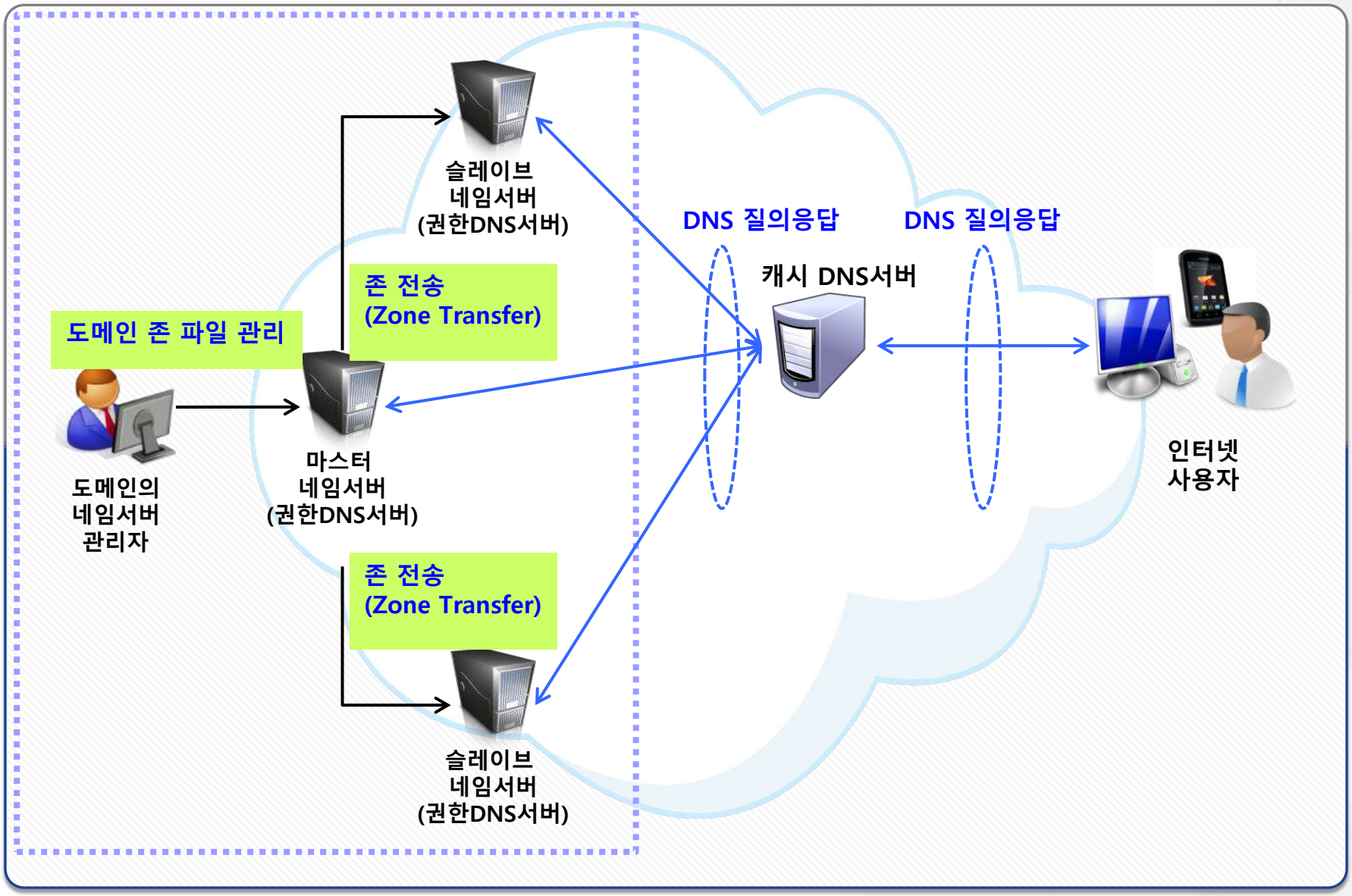
한국인터넷진흥원 인터넷주소기술팀 교육채널

- 한국인터넷진흥원 인터넷주소기술팀 주관 교육 전반 알림
- 등록된 이메일 주소로 교육 등록 안내 메일 발송(회원가입 필요)
  - https://event-us.kr/Ut2KSHw11cCG/event
  - https://www.onoffmix.com/ch/krnicwebnar
- DNS 기술교육 안내 홈페이지(회원가입 불필요)
  - https://krnic.or.kr/jsp/business/operate/dnsEdu.jsp

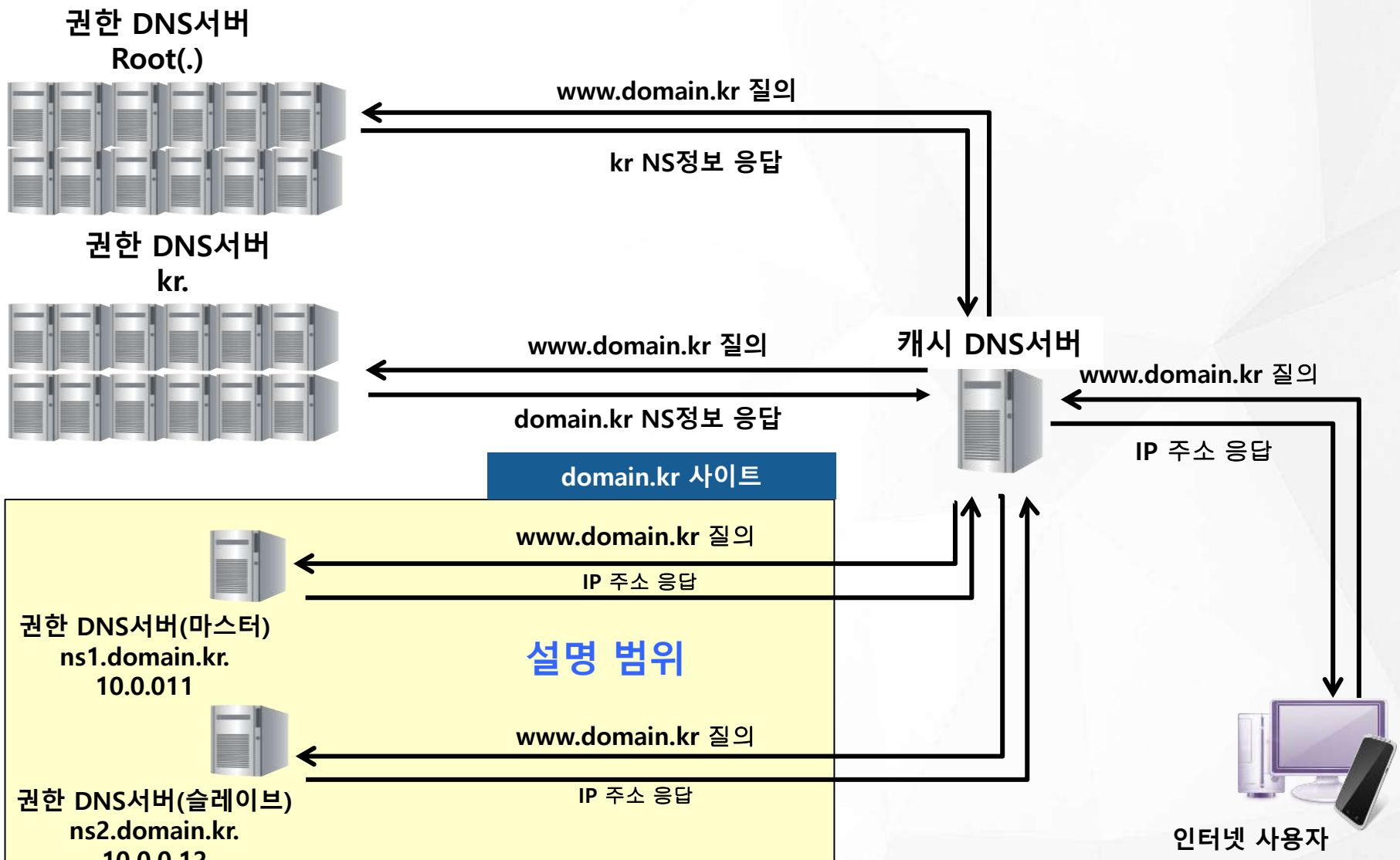
KISA

DNS 서버 기본 설치

DNS 서버 관리는 실제로 어떻게 이루어지나?



강의 개요


권한 DNS서버
Root(.)
www.domain.kr 질의
kr NS정보 응답
권한 DNS서버
kr.
www.domain.kr 질의
domain.kr NS정보 응답
캐시 DNS서버
www.domain.kr 질의
IP 주소 응답
domain.kr 사이트
www.domain.kr 질의
IP 주소 응답
권한 DNS서버(마스터)
ns1.domain.kr.
10.0.011
설명 범위
www.domain.kr 질의
IP 주소 응답
권한 DNS서버(슬레이브)
ns2.domain.kr.
10.0.0.12
인터넷 사용자

DNS 서버 기본 설치

설치 전제 조건

- O/S : Rocky Linux 8.x 혹은 9.x(Server 이상 설치, 패키지 선택 모두)
- DNS S/W : BIND 9.18.x
- 도메인네임 : domain.kr
- DNS 서버
  - 마스터 서버 : ns1.domain.kr (10.0.0.11)
  - 슬레이브 서버 : ns2.domain.kr (10.0.0.12)
- 웹 서버 : www.domain.kr (10.0.0.155)
  domain.kr (10.0.0.55)

Linux 환경에서의 BIND S/W 설치

BIND S/W 버전 선택

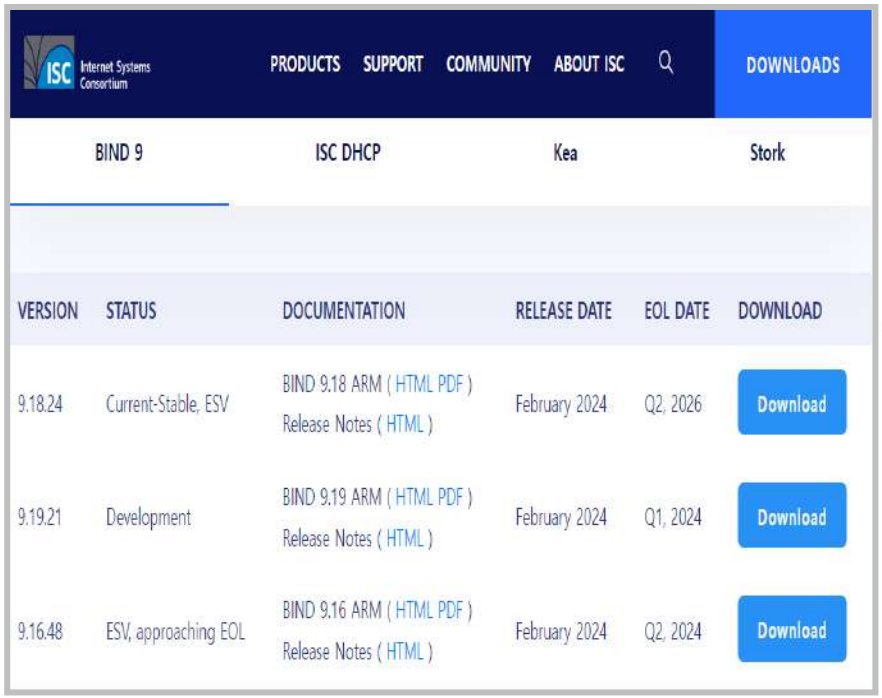
- BIND : - Berkeley Internet Name Domain의 약자
  - DNS를 구현한 소프트웨어의 하나
  - 거의 모든 플랫폼에 포팅 되었고, 가장 널리 사용.

VERSION	STATUS	DOCUMENTATION	RELEASE DATE	EOL DATE	DOWNLOAD
9.18.24	Current-Stable, ESV	BIND 9.18 ARM (HTML PDF) Release Notes (HTML)	February 2024	Q2, 2026	Download
9.19.21	Development	BIND 9.19 ARM (HTML PDF) Release Notes (HTML)	February 2024	Q1, 2024	Download
9.16.48	ESV, approaching EOL	BIND 9.16 ARM (HTML PDF) Release Notes (HTML)	February 2024	Q2, 2024	Download

Linux 환경에서의 BIND S/W 설치

DNS BIND 다운로드 - Unix용 소스 버전

- http://www.isc.org
- bind-9.x.x.tar.gz 소스파일 다운로드
  - 설치 시 최신버전을 다운로드



Linux 환경에서의 BIND S/W 설치

BIND S/W 버전 선택

- **BIND 소프트웨어 지원 정책 종류**

구 분	내 용
Development	- 소프트웨어 개발 중인 버전입니다. - 개발 버전은 24 개월 동안 지원합니다. - 보안패치 버전이 따로 존재하지 않습니다.
Stable	- 안정화된 최신 소프트웨어 버전입니다. - BIND의 안정적인 브랜치는 24 개월동안 사소한 기능 업데이트와 버그 수정을 받습니다.
Extended Support(ESV)	- Stable 버전의 확장 지원 버전입니다. - Stable 버전(24개월)→ ESV버전(24개월) 총 48개월 지원합니다. - 중요한 수정이나 보안패치만 지원합니다.

※ ISC 소프트웨어 지원 정책 참조(https://www.isc.org/downloads/software-support-policy)

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(1/6)

- **BIND 9.18 버전을 설치하기 위해서는 아래 4가지 프로그램을 시스템에 먼저 설치합니다.**
  - ① openssl-devel
  - ② libcap-devel
  - ③ jemalloc-devel(9.18 버전부터 의존) glibc가 제공하는 malloc 함수를 대체하는 라이브러리, 메모리 파편화 방지 등 메모리 할당관리 기능에서 보다 안정적인 기능 제공
  - ④ libuv-devel(1.37버전 이상 필요)

①②는 dnf 명령으로 간단히 설치,

① openssl-devel, ② libcap-devel 설치 (이미 설치되었을 경우, ③부터 설치)

```
[root@ns1 ~]# dnf  -y  install openssl-devel
 .. 생략

[root@ns1 ~]# dnf  -y  install libcap-devel
.. 생략
```

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(2/6)

③의 epel repository(Extra Packages for Enterprise Linux)은 Fedora Project 에서 제공되는 저장소로 각종 패키지의 최신 버전을 제공하는 community기반의 저장소 추가 필요

④는 PowerTools repository 활성화 후 설치

- epel repository 추가 방법은

  # dnf –y install epel-release.noarch ※ 설치 완료 후 repository 확인 가능

```
[root@ns1 ~]# dnf  –y install epel-release.noarch

설치 중:
 epel-release              noarch              8-18.el8              extras              24 k

================================================================================
설치  1 꾸러미
전체 내려받기 크기: 24 k
설치된 크기 : 35 k
꾸러미 내려받기 중:
epel-release-8-18.el8.noarch.rpm                                 4.3 kB/s |  24 kB     00:05

... 중략

설치되었습니다:
  epel-release-8-18.el8.noarch

[root@ns1 ~]# dnf  repolist
저장소 ID           저장소 이름
appstream           Rocky Linux 8 - AppStream
baseos              Rocky Linux 8 - BaseOS
epel                Extra Packages for Enterprise Linux 8 - x86_64
extras              Rocky Linux 8 - Extras
```

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(3/6)

③ jemalloc-devel 설치
dnf -y install jemalloc-devel

```
[root@ns1 ~]# dnf  -y install  jemalloc-devel
.... 중략...

Extra Packages for Enterprise Linux 8 - x86_64                  590 kB/s |  16 MB     00:27
================================================================================
 꾸러미                 구조                 버전                    저장소             크기
================================================================================
설치 중:
 jemalloc-devel          x86_64          5.2.1-2.el8          epel              88 k
종속 꾸러미 설치 중:
 jemalloc                x86_64          5.2.1-2.el8          epel             228 k
... 중략 ...

꾸러미 내려받기 중:
jemalloc-5.2.1-2.el8.x86_64.rpm                                 30 kB/s | 228 kB     00:07
--------------------------------------------------------------------------------
... 중략 ...

설치되었습니다:
  jemalloc-5.2.1-2.el8.x86_64                    jemalloc-devel-5.2.1-2.el8.x86_64

완료되었습니다!
```

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(4/6)

④ libuv-devel 설치
- Power Tools Repository 활성화(2가지 방법 모두 가능)

가. 명령어를 이용하는 방법
dnf config-manager --set-enabled powertools(명령어를 이용한 방법)

나. Rocky-PowerTools.repo 파일을 직접 수정하는 방법

```
# cd /etc/yum.repository.d
# vi Rocky-PowerTools.repo(직접 파일을 수정하는 방법)

... 생략...

[powertools]
name=Rocky Linux $releasever - PowerTools
mirrorlist=https://mirrors.rockylinux.org/mirrorlist?arch=$basearch&repo=PowerTools-$releasever
#baseurl=http://dl.rockylinux.org/$contentdir/$releasever/PowerTools/$basearch/os/
gpgcheck=1
enabled=1 ← 기존 '0'에서 '1'로 값 변경
countme=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rockyofficial
```

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(5/6)

④ libuv-devel 설치
- Power Tools Repository 활성화 확인

```
[root@NS-1 yum.repos.d]# dnf repolist

저장소 ID                 저장소 이름
appstream                 Rocky Linux 8 - AppStream
baseos                    Rocky Linux 8 - BaseOS
extras                    Rocky Linux 8 - Extras
powertools                Rocky Linux 8 – PowerTools


[root@NS-1 yum.repos.d]#
```

Linux 환경에서의 BIND S/W 설치

선수프로그램 설치(6/6)

④ libuv-devel 설치

dnf –y install libuv-devel

```
================================================================================
 꾸러미                 구조              버전                      저장소                크기
================================================================================
설치 중:
 libuv-devel           x86_64            1:1.41.1-1.el8_4          powertools            36 k
종속 꾸러미 설치 중:
 libuv                 x86_64            1:1.41.1-1.el8_4          appstream             155 k
================================================================================
설치  2 꾸러미
(1/2): libuv-devel-1.41.1-1.el8_4.x86_64.rpm                  6.4 kB/s |  36 kB     00:05
(2/2): libuv-1.41.1-1.el8_4.x86_64.rpm                         27 kB/s | 155 kB     00:05
--------------------------------------------------------------------------------------------
합계                                                  11 kB/s | 191 kB     00:17
.... 생략 .....
설치되었습니다:
  libuv-1:1.41.1-1.el8_4.x86_64                libuv-devel-1:1.41.1-1.el8_4.x86_64

완료되었습니다!
```

Linux 환경에서의 BIND S/W 설치

DNS BIND 설치

① 다운로드한 bind-9.x.x.tar.gz 소스파일의 압축해제
tar Jxvf bind-9.x.x.tar

② 압축이 해제된 디렉터리로 이동
cd bind-9.x.x

```
bash-3.00# ls -l
total 16272
-rw-r--r--          1  dns-edu        other        8322766    Aug  11    16:35     bind-9.18.24.tar.gz
bash-3.00# tar Jvxf  bind-9.18.24.tar.gz

.... 중략

bash-3.00# cd bind-9.18.24
bash-3.00#
```

KISA

Linux 환경에서의 BIND S/W 설치

DNS BIND 설치

③ 소스 컴파일 및 인스톨

```
# ./configure --prefix=설치위치 --sysconfdir=설정파일위치 --disable-doh
# make
# make install
```

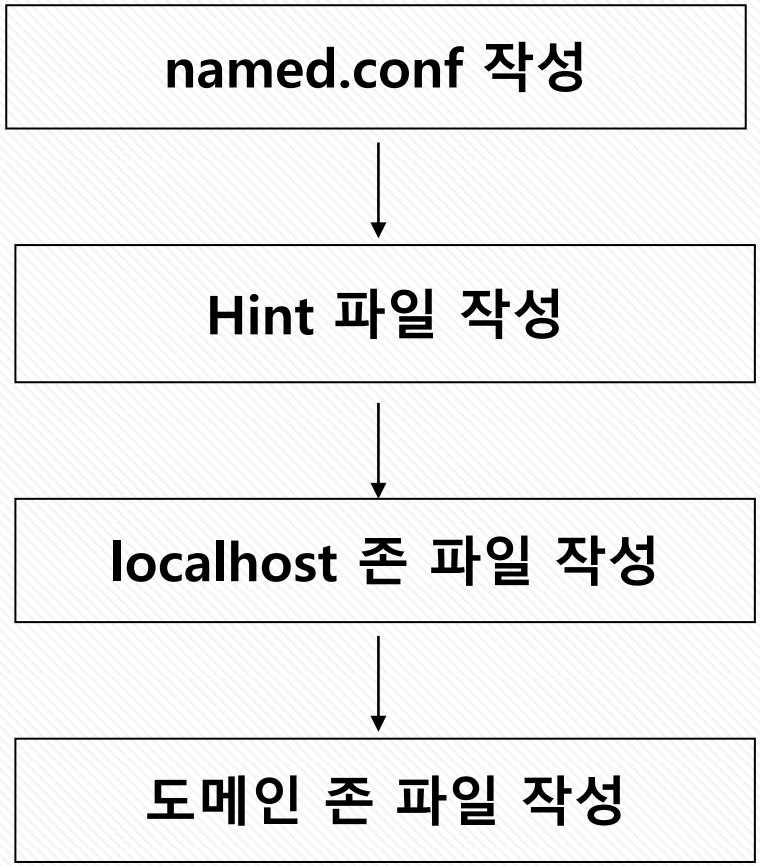
※ '—enable-threads', '—enable-ipv6', '—without-python' 등 옵션은 BIND 9.18에서 삭제됨

※ '—disable-doh' 옵션 추가 지정(DoH 서비스 불필요 시 추가)

```
bash-3.00# ./configure --prefix=/usr/local/bind --sysconfdir=/etc  --disable-doh
checking build system type...sparc-sun-solaris2.10
checking host system type...sparc-sun-solaris2.10
checking whether make sets $(MAKE) ... yes
checking for gcc... gcc

bash-3.00# make
making all in /home/dns-edu/supplies/bind-9.18.18/lib
make[1] : Entering directory '/home/dns-edu/supplies/bind-9.18.18/lib'
making all in /home/dns-edu/supplies/bind-9.18.18lib/isc
make[2] : Entering directory '/home/dns-edu/supplies/bind-9.18.18/lib'
making all in /home/dnsex/BIND/bind-9.16.21/lib/isc/include

bash-3.00# make install
```

DNS 서버 구축

BIND S/W 파일 구성 순서



DNS 서버 구축

named.conf 작성

```
options {
   directory "/var/named";
   dump-file "/var/tmp/named_dump.db";
   statistics-file "/var/tmp/named.stats";
   pid-file "/var/run/named.pid";
  dnssec-validation no;
   };

zone "." {
   type hint;
   file "named.cache";
};

zone "localhost"{
   type master;
   file "localhost.zone";
  };

zone "0.0.127.in-addr.arpa" {
    type master;
    file "localhost.rev";
};

zone "domain.kr"{
   type master;
   file "domain.kr-zone";
   };
```

- 영역 데이터(zone) 파일의 위치를 나타내는 지시자
- 루트 캐쉬와 영역 데이터가 dump 되는 경로를 지정
- 통계 파일이 생성되는 경로 지정
- BIND 작동시의 프로세스 ID가 저장되는 경로지정
- ROOT DNS 정보를 가지는 파일의 이름 및 위치지정
- 로컬호스트는 많은 어플리케이션에서 로컬 PC를 참조하는 축약형으로 사용. 항상 루프백 주소인 127.0.0.1(혹은 IPV6에서는 ::1)로 매핑
  불필요한 루트 서버 트래픽을 발생시키는 일 이외에도 어플리케이션의 속도를 현저하게 저하. 로컬호스트(그리고 그 역매핑)를 해석하기 위해 BIND named.conf 파일 안에 localhost zone 파일 필요
- zone 파일을 선언
- 해당 서버가 primary서버인지 secondary서버인지 표시
- 영역데이터(zone) 파일의 이름을 기술
  일반적으로 FQDN을 이용하여 이름을 지정

DNS 서버 구축

named.conf 작성

- **named.conf의 문법**
  - options이나 zone을 지정할 때는 "{" 와 "}" 사이에 원하는 내용을 넣습니다..
  - "{" 와 "}" 안에서는 각 라인이 끝날 때 ";" 표시를 합니다.
  - "}" 기호를 끝낼 때도 반드시 ";" 표시를 해야 합니다..

- **dnssec-validation no; 옵션 추가 설정**
  - BIND 9.14 이상부터 dnssec-validation 옵션 default값이 auto로 전환됩니다..
  - DNSSEC 서명검증 동작이 기본동작이 됩니다.
  - 이로 인해 권한DNS 서버의 경우 권한DNS임에도 불구하고 루트 존의 서명키 관리 절차가 수행되어 외부 인터넷으로 루트 존 DNSKEY 질의 절차 진행, 외부 질의가 차단된 환경의 권한 DNS서버의 경우 오류 로그가 발생합니다.
  - 권한DNS 서버의 경우, 명시적으로 'dnssec-validation no;' 옵션을 설정해야 합니다.

DNS 서버 구축

Hint(named.cache)파일 작성

- **방법 : dig 명령을 이용해 파일 생성**

```
bash-3.00# mkdir  /var/named

bash-3.00# dig @a.root-servers.net  .  ns > /var/named/named.cache

bash-3.00# ls -l /var/named
```

- **상기의 방법을 통해 얻은 Hint 파일을 /var/named경로에 named.cache 파일 저장**

※ 최초 설치 시 /var/named 디렉터리는 별도의 명령으로 생성해 준다.

※ Hint 파일이란?
최상위 DNS(root 네임서버)의 정보가 기록되어 있는 파일이다. BIND가 시작될 때 named.cache 파일의 내용을 메모리에 로딩한다.

DNS 서버 구축

localhost 존 파일 작성(1/2)

- **/var/named/localhost.zone**

```
$TTL 1d ;
$ORIGIN localhost.
localhost.     IN     SOA localhost. hostmaster.localhost. (
                                2024040801 ; serial
                                    3H ; refresh
                                   15M ; retry
                                    1w ; expire
                                    3h ; minimum
      )
localhost.      IN   NS localhost.
localhost.      IN   A   127.0.0.1
```

DNS 서버 구축

localhost 존 파일 작성(2/2)

- **/var/named/localhost.rev**

```
$TTL 86400 ; 24 hours
$ORIGIN 0.0.127.in-addr.arpa.
@       IN      SOA     localhost. hostmaster.localhost.  (
                        2024040801 ; Serial number
                        3h      ; Refresh
                        15      ; Retry
                        1w      ; Expire
                        3h      ; Minimum
)
        IN      NS      localhost.
1       IN      PTR     localhost.
```

DNS 서버 구축

도메인 존 파일 작성

- **/var/named/domain.kr-zone**

```
$TTL 60       ; 1 minute                                        ← SOA영역
@          IN SOA        ns1.domain.kr. manager.domain.kr.(
                         2024101301; serial
                         60        ; refresh (6 hours)
                         30        ; retry (30 minutes)
                         1209600     ; expire (2 weeks)
                         60          ; minimum (1 minute)
                         )

                         IN      A          10.0.0.55          ← DNS서버 지정
                         IN      NS         ns1.domain.kr.
                         IN      NS         ns2.domain.kr.
ns1.domain.kr.           IN      A          10.0.0.11
ns2.domain.kr.           IN      A          10.0.0.12

$ORIGIN domain.kr.                                             ← A Record 설정
www                      IN      A          10.0.0.155
ftp                      IN      A          10.0.0.154
```

DNS 서버 구축

존 파일 설명 – SOA 영역 (1/2)

- **SOA(start of authority) Resource Record**

- 해당 도메인에 대해 DNS서버가 인증(authoritative)된 자료를 가지고 있음을 의미합니다.
- 자료가 최적의 상태로 유지되도록 관리해야 합니다.

```
$TTL 86400      ; 1 day
@       IN SOA      ns1.domain.kr. manager.domain.kr.(
```

$TTL	• DNS cache를 저장하는 기간 (초단위)
@	• zone데이터에 대한 도메인네임을 지칭 • '@' or FQDN
IN	• 클래스 이름, 일반적으로 IN(Internet)을 사용
SOA	• 리소스 레코드의 유형을 지정하는 필드
ns1.domain.kr.	• master DNS서버의 도메인네임 • 상위 도메인네임에다 기재한 글루레코드를 기술
manager.domain.kr.	• 관리자 E-mail 주소 • '@'를 '.'로 바꾸어 표기

KISA

DNS 서버 구축

존 파일 설명 – SOA 영역 (2/2)

```
2024102301; serial
21600       ; refresh (6 hours)
1800        ; retry (30 minutes)
1209600     ; expire (2 weeks)
86400       ; minimum (1 day)
)
```

serial	• 존 파일 레코드의 갱신여부를 숫자로 표기
refresh	• 2차DNS서버의 1차DNS서버 존 파일 갱신여부 체크 주기(초단위)
retry	• 갱신여부체크 실패 시 재시도까지 시간 간격(초단위)
Expire	• 갱신여부체크가 연속으로 실패시 2차DNS서버의 zone데이터를 Expire시키기까지의 시간간격(초단위)
mininum	• 존 파일의 TTL값을 지정

KISA

DNS 서버 구축

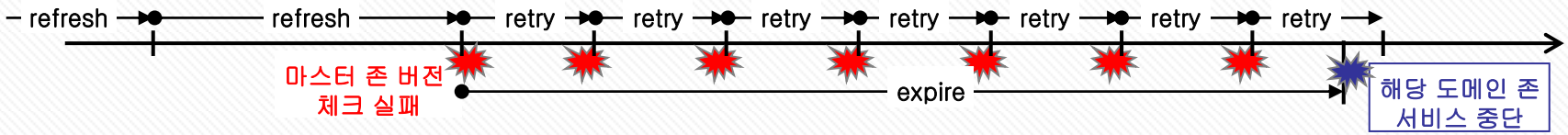
DNS서버 지정

```
                    IN    NS    ns1.domain.kr.
                    IN    NS    ns2.domain.kr.
ns1.domain.kr.  IN    A     10.0.0.11
ns2.domain.kr.  IN    A     10.0.0.12
```

IN	• 클래스 이름, 일반적으로 IN(Internet)을 사용
NS	• 응답 레코드 타입 지정, 여기서 NS는 Name Server
A	• 응답 레코드 타입 지정, 여기서 A는 IP Address

[참고] SOA 레코드의 주요 필드 역할

- **DNS서버에 의한 도메인 존 데이터 관리용 데이터**
  - 도메인 존 데이터의 동기화 관리 (슬레이브 DNS서버)
    - Refresh 필드 : 마스터 도메인 존 버전정보 체크 주기 지정
    - Retry 필드 : Refresh 값 주기에 의한 존 버전 체크 실패시 재시도 시간 간격
    - Expire 필드 : Retry 값에 의한 연속적 재시도 지속 반복 후 슬레이브 DNS서버에서 해당 도메인 데이터의 최종 서비스 중단처리 시점까지의 경과시간



  - 도메인의 마스터 DNS서버 네임 지정
    - Mname 필드 : 도메인 존 DNS서버 중 마스터 DNS서버의 도메인 네임 지정
      - DNS Notify, DNS Dynamic Update 등 확장 표준에서 사용
      - DNS Notify는 디폴트로 SOA mname 필드의 DNS서버(마스터 DNS서버)를 제외한 나머지 NS 레코드의 DNS서버로 Notify 메시지 송출

- 네거티브 캐싱(negative caching)
  - 도메인 존에 부재한 레코드의 캐싱 시간 지정
    - Minimum 필드 : 도메인 존에 정의되지 않은 레코드를 질의한 경우, 캐시 DNS서버 캐시에 이 부재 레코드 정보를 저장해 두는 캐싱 기간

KISA

[참고] SOA 레코드의 refresh, retry, expire 필드 설정

- Refresh, Retry, Expire 필드 설정 (참고값)
  - Refresh : 20분(1,200초) ~ 2일(172,800초)
    - ✓ DNS Notify 미적용 경우 : 20분(1,200초) ~ 2시간(7,200초)
  - Retry : refresh 보다 작은 값
  - Expire : 1주(604,800초)~약 6주(3,600,00초)
    - ✓ 단 expire 값은 (refresh x 7) 보다 큰 값으로 설정 권고

- 상기 값은 RFC1912(Info.) 문서 및 유럽지역 NIC들의 권고 설정 값 등을 종합 참조한 것입니다.
- 실제 설정 값은 사이트의 인터넷 서비스 성격, 네트워크 용량 등을 고려하여 결정해야 합니다.

예시 설정:

```
$ORIGIN domain.kr.
$TTL 300
 @     IN SOA ns1.domain.kr. dnsadm.domain.kr. (
              2024102301 ; serial
              1800       ; refresh (30 minutes)
              300        ; retry (5 minutes)
              3600000    ; expire (5 weeks 6 days 16 hours)
              300        ; minimum (5 minutes)
              )
                    IN NS  ns1.domain.kr.
                    IN NS  ns2.domain.kr.

ns1.domain.kr.   IN A   10.0.0.11
ns2.domain.kr.   IN A   10.0.0.12
```

[참고] SOA 레코드의 minimum 필드 설정

- minimum 필드 설정 (참고값)
  - minimum : 3분(180초)~2시간(7,200초)
    - 특별한 사유가 없다면, 1시간 ~ 2시간 사이 값 설정 권고 (RFC2308)
    - 너무 긴 시간 값을 설정하는 경우, 신규 레코드 추가 설정 시, 최악의 경우 해당 시간만큼 네거티브 캐싱이 소진되기를 기다려야 할 수 있습니다.
    - 너무 짧은 시간 값을 설정하는 경우, 부재 레코드 질의가 자주 유입될 수 있습니다.
  - 상기 값은 RFC2308(Std.) 문서를 참조한 것입니다.

예시 설정:

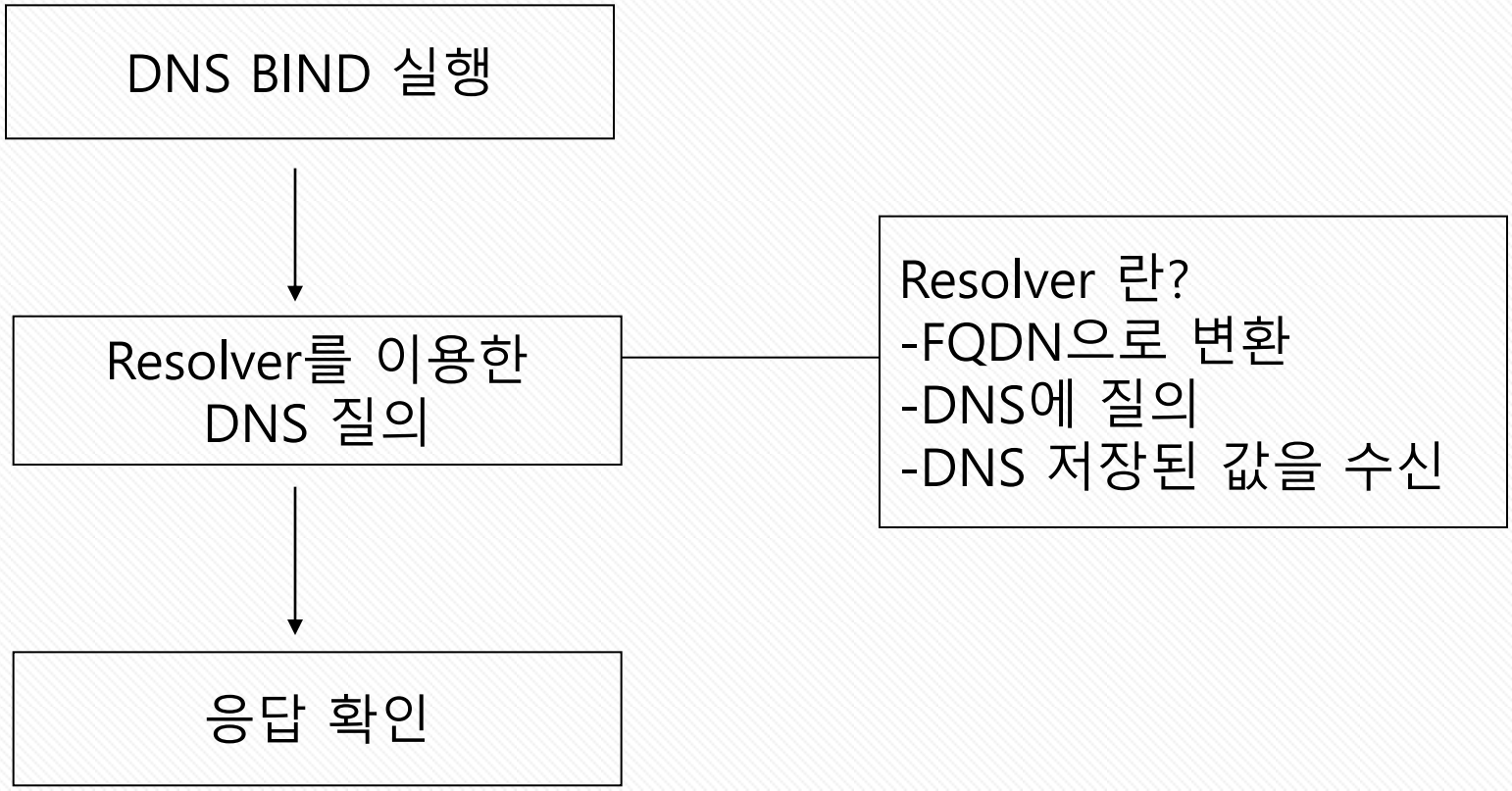
```
$ORIGIN domain.kr.
$TTL 300
 @     IN SOA ns1.domain.kr. dnsadm.domain.kr. (
             2024102301 ; serial
             1800       ; refresh (30 minutes)
             300        ; retry (5 minutes)
             3600000    ; expire (5 weeks 6 days 16 hours)
             300        ; minimum (5 minutes)
             )
       IN NS  ns1.domain.kr.
       IN NS  ns2.domain.kr.

ns1.domain.kr.   IN A   10.0.0.11
ns2.domain.kr.   IN A   10.0.0.12
```

부재 레코드 질의:

부재 도메인 네임 www.domain.kr의 A 타입 질의 경우,

→ 네거티브 캐싱 시, TTL 값이 없으므로, SOA minimum 필드값 사용 캐싱
→ "www.domain.kr. IN" 을 300초 동안 네거티브 캐싱

질의/응답 확인


DNS BIND 실행
Resolver를 이용한
DNS 질의
응답 확인
Resolver 란?
-FQDN으로 변환
-DNS에 질의
-DNS 저장된 값을 수신

질의/응답 확인

BIND 실행 및 확인(1/2)

- **BIND 실행**
  # /usr/local/bind/sbin/named

- **BIND 구동 확인**
  # ps –ef | grep named

```
[root@NS1 ~]# /usr/local/bind/sbin/named
[root@NS1 ~]# ps -ef |grep named
root     12445     1  1 20:10 ?        00:00:00 /usr/local/bind/sbin/named
root     12447  1673  0 20:10 tty1     00:00:00 grep named
[root@NS1 ~]#
```

질의/응답 확인

BIND 실행 및 확인(2/2)

- **BIND 구동 확인 시 정상 동작된 것이 확인 안될 때**
  # tail /var/log/messages

설정오류 예시)

```
[root@NS-1 named]# tail /var/log/messages
Mar 10 22:05:32 NS-1 named[67427]: using 1 UDP listener per interface
Mar 10 22:05:32 NS-1 named[67427]: DNSSEC algorithms: RSASHA1 NSEC3RSASHA1 RSASHA256 RSASHA512 ECDSAP256SHA256
Mar 10 22:05:32 NS-1 named[67427]: DS algorithms: SHA-1 SHA-256 SHA-384
Mar 10 22:05:32 NS-1 named[67427]: HMAC algorithms: HMAC-MD5 HMAC-SHA1 HMAC-SHA224 HMAC-SHA256 HMAC-SHA384
Mar 10 22:05:32 NS-1 named[67427]: TKEY mode 2 support (Diffie-Hellman): yes
Mar 10 22:05:32 NS-1 named[67427]: TKEY mode 3 support (GSS-API): yes
Mar 10 22:05:32 NS-1 named[67427]: loading configuration from '/etc/named.conf'
Mar 10 22:05:32 NS-1 named[67427]: /etc/named.conf:17: missing ';' before '}'
Mar 10 22:05:32 NS-1 named[67427]: loading configuration: failure
Mar 10 22:05:32 NS-1 named[67427]: exiting (due to fatal error)
```

질의/응답 확인

Checkconf, Checkzone

- **named-checkconf : named.conf 파일 문법 확인**
  - 사용법 : /usr/local/bind/bin/named-checkconf

    ※ 문법상 오류가 있을 경우, 해당 구문의 라인번호와 오류사항 표기

```
[root@ns-1 ~]# /usr/local/bind/bin/named-checkconf
/etc/named.conf : 17 : missing ';' before 'allow-update'
[root@ns-1 ~]#
```

- **named-checkzone : 존파일 문법 확인**
  - 사용법 : /usr/local/bind/bin/named-checkzone 도메인명 존파일 이름

    ※ 문법상 오류가 있을 경우, 해당 구문의 라인번호와 오류사항 표기

```
[root@ns-1 ~]# /usr/local/bind/bin/named-checkzone domain.kr /var/named/domain.kr-zone
zone domain.kr / IN: loaded serial 2024102301
OK
[root@ns-1 ~]#
```

질의/응답 확인

Resolver Tool을 이용한 질의/응답

- **사용법 : dig @(점검대상 IP주소) 도메인 이름 질의레코드_종류**
  예) dig @127.0.0.1 www.domain.kr a +norec

```
; <<>> DiG 9.3.2 <<>> @127.0.0.1 www.domain.kr A +norec
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; -]]HEADER[[- opcode: QUERY, status: NOERROR, id: 1295
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;domain.kr.              IN          A

;; ANSWER SECTION:
www.domain.kr.        60    IN          A       10.0.0.155

;; AUTHORITY SECTION:
domain.kr         60    IN          NS       ns1.domain.kr
domain.kr         60    IN          NS       ns2.domain.kr


;; ADDITIONAL SECTION:
ns1.domain.kr.    60    IN           A       10.0.0.11
ns2.domain.kr.    60    IN           A       10.0.0.12


;; Query time: 15 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Sep 21 14:47:27 2006
;; MSG SIZE  rcvd: 184
```

DNS 이중화 구성

마스터/슬레이브 DNS서버 구성

- **구성의 목적**
  - "도메인 존 데이터" 이중화 구성 구현
    - ✓ 도메인 존을 가지고 있는 DNS서버를 2식 이상으로 구성하여 이중화/다중화 구현
    - ✓ 일부 DNS서버 시스템의 장애 시에도 도메인 존 데이터는 언제나 가용 상태 유지
- **마스터/슬레이브 DNS서버의 구분 기준**
  - "도메인 존에 대하여 마스터 DNS서버, 슬레이브 DNS서버" 구분
    - ✓ BIND의 경우, named.conf 파일의 zone 설정에서 "**type master;**"이면 해당 도메인 존에 대해 이 DNS서버는 마스터 DNS서버 역할
- **마스터/슬레이브 DNS서버의 도메인 존 데이터 동기화 자동 관리**
  - 존 데이터의 SOA 레코드 필드 설정 값에 의존하여 존 데이터 자동 동기화
  - DNS서버 환경설정 파일에 의해 마스터 존/슬레이브 존 구분 관리

마스터/슬레이브 DNS서버 설정

- **마스터 DNS서버 환경설정에 "마스터 존" 지정 작성, 존파일 저장**
  - BIND의 경우, named.conf 파일에서 작성

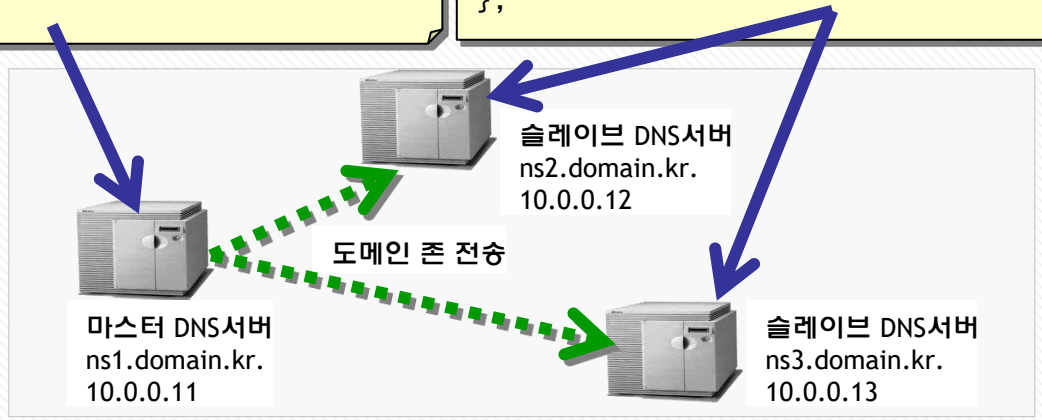
- **슬레이브 DNS서버 각각의 환경설정에 "슬레이브 존" 지정 작성**
  - 각 DNS서버에 대하여 아래 설정 예시 참조

도메인의 "마스터 존" 지정 설정

```
options {
    allow-transfer {10.0.0.12; 10.0.0.13;};
// 도메인의 슬레이브 DNS서버에 대해 존 전송 허용 설정

};

zone "domain.kr" IN {
    type master;                  // "마스터 존" 으로 지정
    file "my-domain.kr.zone"; // 도메인 존 파일명 지정
};
```

도메인의 "슬레이브 존" 지정 설정

```
options {
    allow-transfer { none; };
};

zone "domain.kr" IN {
    type slave;          // "슬레이브 존 "으로 지정
    masters { 10.0.0.11;};
        // 마스터 DNS서버 IP 주소 지정
};
```



마스터/슬레이브 DNS서버 설정

● 도메인의 마스터/슬레이브 DNS서버 설정

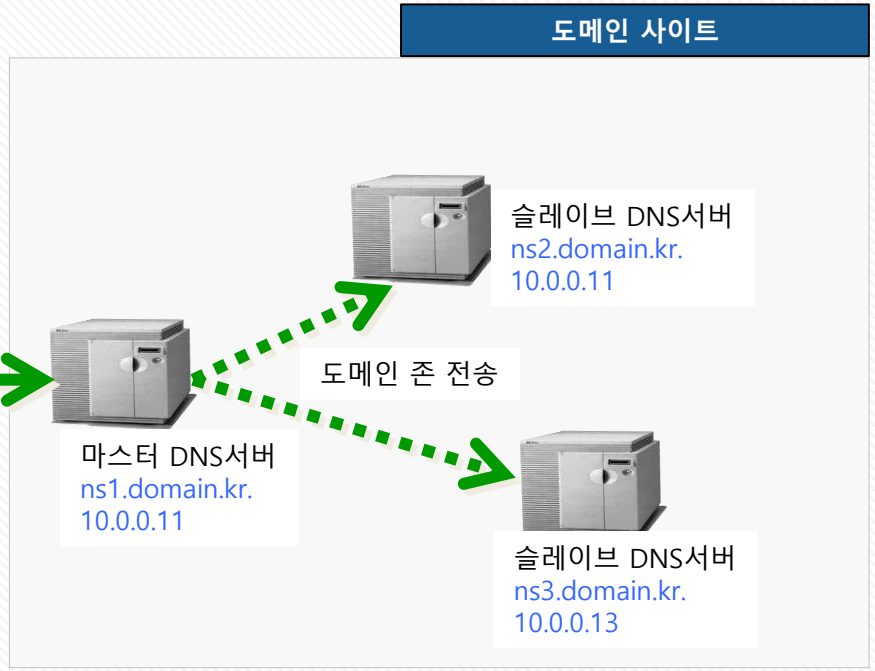
- 도메인 존의 마스터 DNS서버와 슬레이브 DNS서버 리스트 확정
- DNS서버 리스트를 모두 반영하여 도메인 존 파일 작성
- 마스터 DNS서버 환경설정에 "마스터 존" 지정 작성, 존파일 저장
- 슬레이브 DNS서버 각각의 환경설정에 "슬레이브 존" 지정 작성
- 마스터 DNS서버에 구성 사항 반영, 슬레이브 DNS서버에 반영 적용
- 각 DNS서버의 정상응답 동작 점검

도메인 존 파일

```
$ORIGIN domain.kr.
$TTL 300
 @     IN SOA ns1.domain.kr. dnsadm.domain.kr. (
              2024052301 ; serial
              1800       ; refresh (30 minutes)
              300        ; retry (5 minutes)
              3600000    ; expire (5 weeks 6 days 16 hours)
              300        ; minimum (5 minutes)
              )
      IN  A    10.0.0.55

      IN NS   ns1.domain.kr.
      IN NS   ns2.domain.kr.
      IN NS   ns3.domain.kr.

ns1.domain.kr.   IN A    10.0.0.11
ns2.domain.kr.   IN A    10.0.0.12
ns3.domain.kr.   IN A    10.0.0.13


www    IN A    10.0.0.155
ftp    IN A    10.0.0.56
```

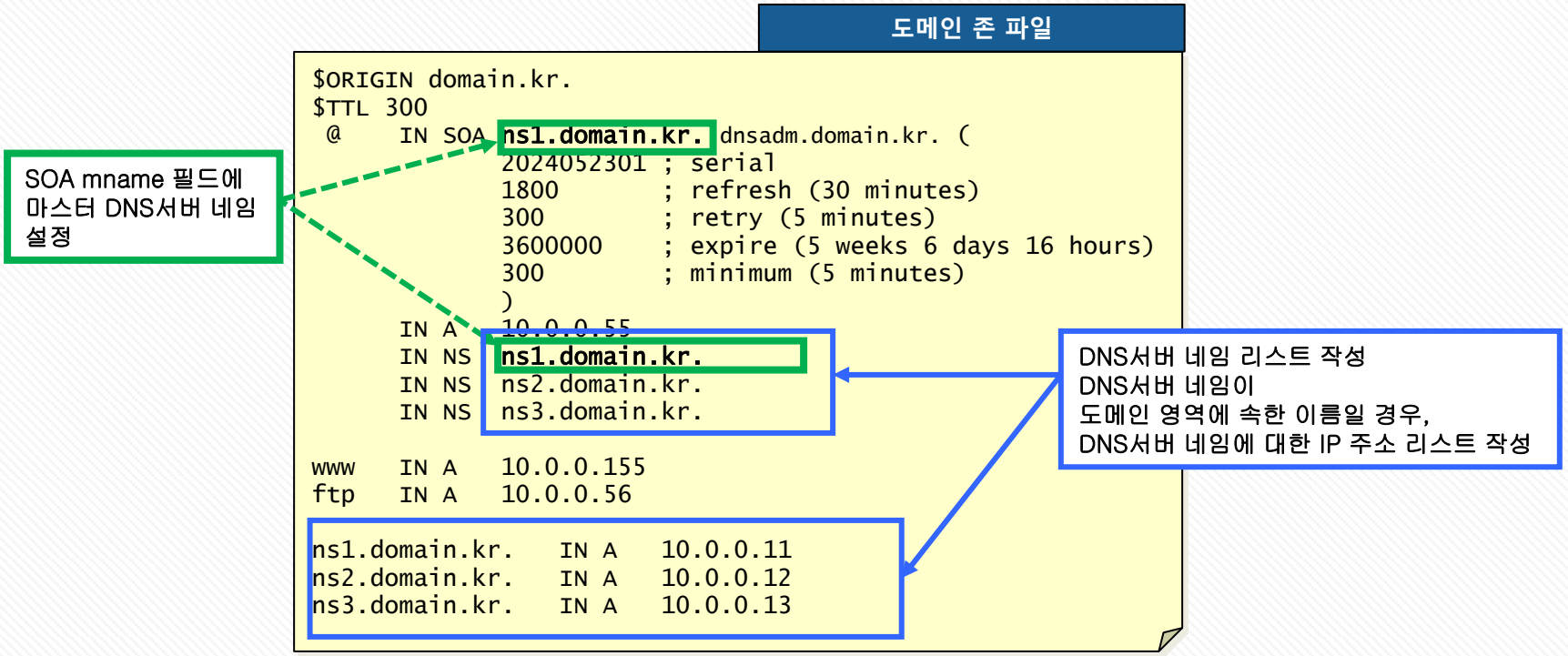
존 데이터 적용

도메인 사이트



마스터/슬레이브 DNS서버 설정

- **DNS서버 리스트를 모두 반영하여 도메인 존 파일 작성**
  - 도메인의 DNS서버 리스트 (도메인 네임, IP 주소) 반영 존 파일 작성
  - SOA 레코드의 mname(마스터 DNS서버 네임) 필드에 마스터 DNS서버 네임을 기입 작성
    - SOA mname에 마스터 DNS서버 네임을 기입함으로써, DNS 확장 기능이 정상 동작할 수 있도록 준비 (DNS Notify, Dynamic Update 등)
    - SOA 기타 필드의 설정값은 SOA 레코드 설정 관련 부분에서 상세 설명



마스터/슬레이브 DNS서버 설정

- **마스터 DNS서버에 구성 사항 반영, 슬레이브 DNS서버에 반영 적용**
  - 마스터 DNS서버에 먼저 반영 적용한 후, 슬레이브 DNS서버 반영 적용
  - BIND 경우, "rndc reconfig" 실행 DNS서버 반영 적용

- **각 DNS서버의 정상응답 동작 점검**
  - 각 DNS서버에 대해 설정한 도메인에 대해 any 타입으로 iterative 질의
    - 캐시 DNS서버 입장에서 질의 필요 (iterative 질의)
  - dig 명령 (iterative 질의 = +norec 옵션)
    - dig @10.0.0.11 domain.kr a +norec
    - dig @10.0.0.12 domain.kr a +norec
    - dig @10.0.0.13 domain.kr a +norec
  - 응답결과 체크 포인트
    - 각 DNS서버의 응답이 서로 동일
    - 응답 메시지 헤더에 반드시 AA 플래그 설정 응답
    - Answer Section 데이터 정상 응답 확인

※ 마스터/슬레이브 DNS서버는 도메인 존 관리상의 구분일 뿐, 외부의 캐시 DNS서버 입장에서는 모두 authoritative 응답을 하는 authoritative DNS서버

네임서버 기본 보안설정

DNS 서버 보안

DNS서버 보안 기본사항

- **네임서버에 대한 접근 통제**
  - 방화벽 UDP/TCP 53포트 허용, 인가된 사람만 접근
- **최소한의 서비스 운영**
- **최소 사용자 및 관리자 계정 유지 및 암호 관리 철저**
- **물리적 접근 통제**
  - 서로 다른 네트워크에 Master, Slave DNS 서버 설치
  - 서로 다른 건물에 Master, Slave DNS 서버 설치
  - 서로 다른 OS에 Master, Slave DNS 서버 설치

DNS 서버 보안

DNS 서버 S/W 보안 설정 종류

- **DNS 서버 용도에 따른 네트워크 구성**
  - Master, Slave DNS 서버 이중화/다중화 구성
- **DNS 서버 운영에 다른 구성**
  - 권한(Authority) DNS 와 캐시(Caching) DNS를 구분하여 설치 운영
- **DNS 서버 보안 설정**
  - BIND 버전 정보 유출 제한(/etc/named.conf 파일 수정)
    - ☞ options { version "Unknown"; };
  - 캐시 DNS 서버가 아니라면 Recursion 질의 제한
    - ☞ options { recursion no; };
  - Zone-Transfer 제한
    - ☞ options { allow-transfer {address_match_list}; };
  - Dynamic Update 제한
    - ☞ options { allow-update {address_match_list}; };
  - Notify 제한
    - ☞ options { allow-notify {address_match_list}; };
- **TSIG를 이용한 보안 강화**
  - Zone Transfer, notify, 순환질의, Dynamic update 시 암호화
- **DNS 서버 프로그램 최신 버전 사용**

KISA

네임서버 버전정보 노출방지 설정

BIND 버전 정보 확인

- **BIND는 버전에 따른 취약점을 가지고 있습니다.**
- **버전을 숨기는 것만으로도 취약점에 대한 정보를 숨길 수 있습니다.**

• dig @10.0.0.11 txt chaos version.bind

```
; <<>> DiG 9.18.24 <<>> @10.0.0.11 txt chaos version.bind
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19156
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "9.18.24"

;; AUTHORITY SECTION:
version.bind.           0       CH      NS      version.bind.

;; Query time: 7 msec
;; SERVER: 10.0.0.11#53(10.0.0.11)
;; WHEN: Wed Mar  7 03:54:48 2024
;; MSG SIZE  rcvd: 72
```

네임서버 버전정보 노출방지 설정

● 네임서버 버전정보 노출방지

- 네임서버 S/W 버전정보 제공 방식
  - BIND DNS의 경우, "version.bind"의 CHAOS 도메인으로 버전정보 기본 제공합니다.
  - BIND DNS 외에도 NSD, Nominum ANS, PowerDNS 등의 네임서버가 동일한 방식으로 S/W 버전정보를 기본 제공합니다.

```
$ dig @211.182.233.3 version.bind ch txt +short
"BIND 9.18.3"
$ dig @193.0.14.129 version.bind ch txt +short
"NSD 4.10.7"
$ dig @152.99.1.10 version.bind ch txt +short
"Nominum ANS 2.6.0.1"
$ dig @222.239.76.130 version.bind ch txt +short
"Served by POWERDNS 2.9.21 $Id: packethandler.cc 1036 2024-04-19 20:43:14Z ahu $"
```

- 네임서버 S/W 버전정보 노출 방지의 필요성
  - 네임서버 S/W 버전별 공지된 보안 취약점 이용, 해당 버전의 네임서버 리스트 파악 공격 시도의 위험이 있습니다.
  - 예: '09년 7월말 공지 BIND9 Dynamic Update를 사용한 원격 DoS 공격 취약점
    - ✓ 취약점 버전의 네임서버를 파악하여, 원격에서 서비스 중단시킬 수 있습니다.
  - 네임서버 S/W 취약점 가능성에 대비 사용 중인 S/W 버전정보 노출방지가 필요합니다.

네임서버 버전정보 노출방지 설정

<버전정보 문자열 TXT 레코드 제거 설정>

```
options {
    ... 기존 설정 생략 ...

    version none;
        // 버전정보 문자열 레코드 제거
};
```

```
$ dig @localhost version.bind ch txt

; <<>> DiG 9.18.24<<>> @localhost version.bind ch txt
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1146
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL:
0

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; AUTHORITY SECTION:
version.bind.           86400   CH      SOA     version.bind.
hostmaster.version.bind. ...

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jun 12 19:09:52 2024
;; MSG SIZE  rcvd: 77
```

네임서버 버전정보 노출방지 설정

<의미 없는 문자열로 대체 설정>

```
options {
    … 기존 설정 생략 …

    version “UNKNOWN”;
        // 의미없는 문자열로 대체
};
```

```
$ dig @localhost version.bind ch txt

; <<>> DiG 9.18.24 <<>> @localhost version.bind ch txt
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 953
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
0

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "UNKNOWN"

;; AUTHORITY SECTION:
version.bind.           0       CH      NS      version.bind.

;; Query time: 10 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jun 12 19:10:35 2024
;; MSG SIZE  rcvd: 63
```

질의제한

모든 질의 제한

- BIND 9.x에서는 allow-query 서브 구문에서 질의에 대한 IP주소 기반의 액세스 리스트를 만들 수 있습니다.
- 이 액세스 리스트를 특정영역에 적용하거나 DNS서버가 질의하는 모든 질의에 적용할 수 있습니다.
- 다음은 allow-query 서브 구문의 전체적인 형식입니다.

```
Options {
        allow-query { address_match_list; };
};

acl "my-domain-net" { 10.0.0.0/8; }; ←네트워크 대역(특정 IP주소도 가능)
```

질의제한

특정영역의 질의 제한

- 액세스 제어 리스트(ACL)을 특정영역에 적용할 수 있습니다.
- 이런 경우에는 보호하려는 영역에 대한 존 구문에 allow-query 서브 구문을 이용합니다.

```
acl “my-domain-net” { 10.0.0.0/8; } ←네트워크 대역

zone “domain.kr” {
type master;
file “domain.kr-zone”;
allow-query { “my-domain-net” ; };
};
```

도메인 존 전송 제한 설정

도메인 존 정보 전송 이란?

- **존 정보 전송 설정은 마스터와 슬레이브 간의 DNS 존데이터 관리의 편의성을 위해 설정합니다.**

  (마스터서버에서 존 데이터를 변경 시 슬레이브 서버에 자동 업데이트 설정)

- **nsupdate를 사용하여 존 데이터를 수정해야 바로 적용이 됩니다.**
- **직접 존 데이터 파일을 수정 시 데이터 적용이 이루어 지지 않으므로 주의해야 합니다.**

- **존 정보 전송에는 2가지 타입의 전송 방식이 있습니다.**
  - 전체 전송 (AXFR) : 존 데이터 전부 전송
  - 증분 전송 (IXFR) : 바뀐 데이터만 전송

도메인 존 전송 제한 설정

도메인 존 정보 전송 제한 설정법 (마스터 서버)

- 마스터 서버에서는 존 정보 전송 할 대상을 지정합니다

```
options {
    directory "/var/named";
    dump-file "/var/tmp/named_dump.db";
    statistics-file "/var/tmp/named.stats";
    pid-file "/var/run/named.pid";
    recursion no;
    allow-recursion {OOO.OOO.OO.OO;};
    version "Internet Server";
    allow-transfer { OOO.OOO.OO.OO; }; ← 변경할 부분,  Slave DNS서버의 IP주소
};
zone "." {
    type hint;
    file "named.cache";
};
zone "domain.kr" {
    type master;
    file "domain.kr-zone";
     allow-update {127.0.0.1; }; ← 해당 존파일의 nsupdate를 허용할 IP
```

도메인 존 전송 제한 설정

도메인 존 정보 전송 제한 설정법 (슬레이브 서버)

- 슬레이브 서버에서는 존 데이터의 타입과 마스터서버의 IP주소를 지정합니다.

```
options {
    directory "/var/named";
    dump-file "/var/tmp/named_dump.db";
    statistics-file "/var/tmp/named.stats";
    pid-file "/var/run/named.pid";
    recursion no;
    allow-recursion {OOO.OOO.OO.OO;};
    version "Internet Server";
    allow-transfer { none; }; ← 존 정보 제한 설정
};
zone "." {
    type hint;
    file "named.cache";
};
zone "domain.kr" {
    type slave;
    file "domain.kr-slave";
    masters { XXX.XXX.XXX.XXX; }; ← Master DNS서버의 IP주소
};
```

Open 캐시 DNS 보안설정

Open 캐시 DNS 보안 설정

Open 캐시 DNS설정 및 취약점

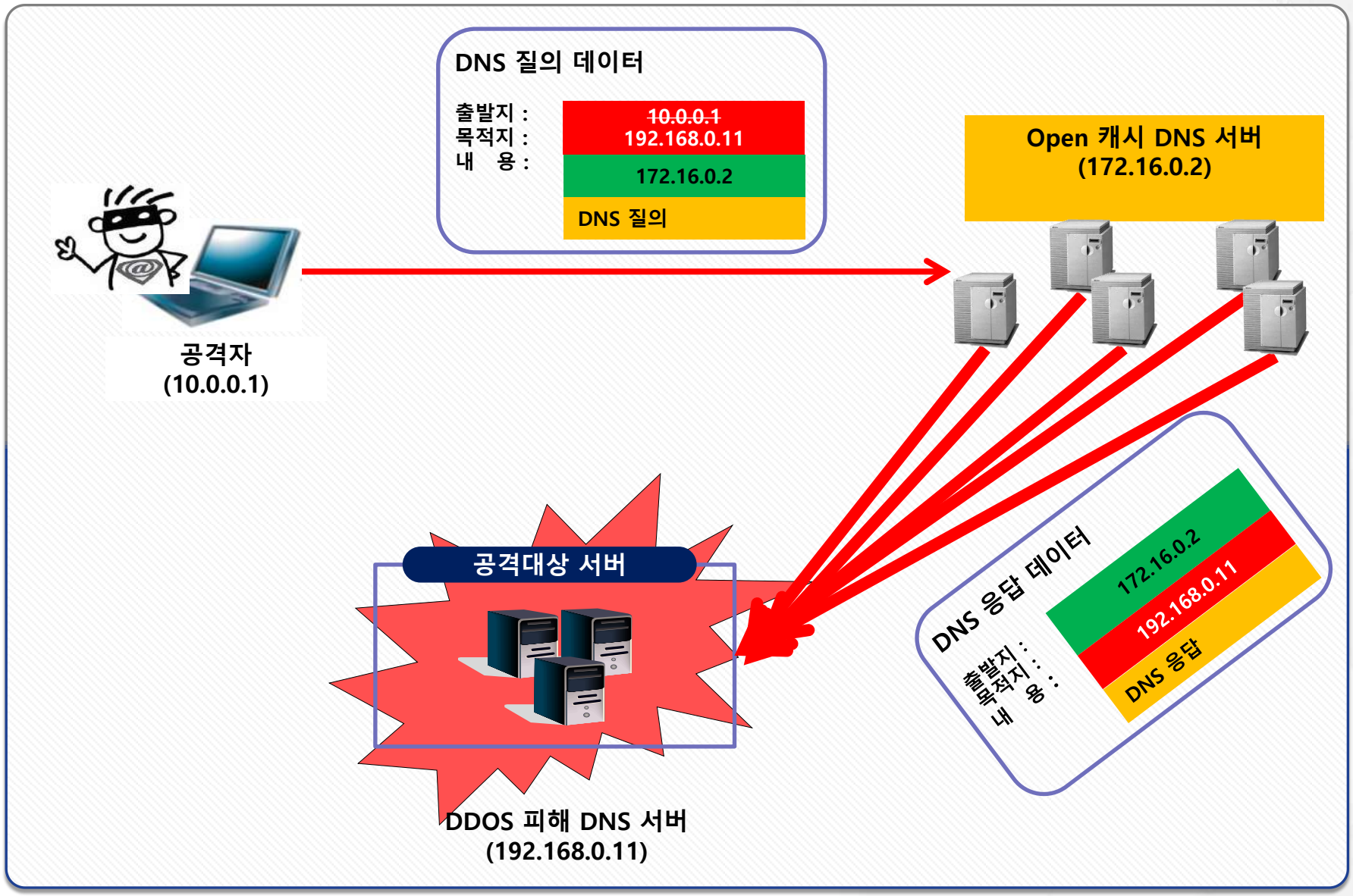
- Open 캐시 DNS는 모든 DNS질의에 응답을 해주는 상태의 DNS 서버를 말합니다.
- Open 캐시 DNS 설정은 별도의 제한 설정을 하지 않을 경우 자동으로 동작상태 설정됩니다.

- Open 캐시 DNS는 다음 공격에 취약합니다.
  - DDoS / 증폭공격
    - Open 캐시 DNS는 네임서버로 들어오는 모든 쿼리에 대해 검색을 시도합니다.
    - 수많은 질의가 한꺼번에 보내는 공격(DDoS)을 한다면 해당 네임서버는 정상적인 요청을 처리하지 못하는 상태에 빠질 수 있습니다.
    - 제 3의 사이트 공격을 위해 4,000 바이트 이상의 응답 패킷을 공격대상 사이트로 발생하도록 조작하여 DNS 증폭공격에 이용 될 수 있으므로 취약합니다.

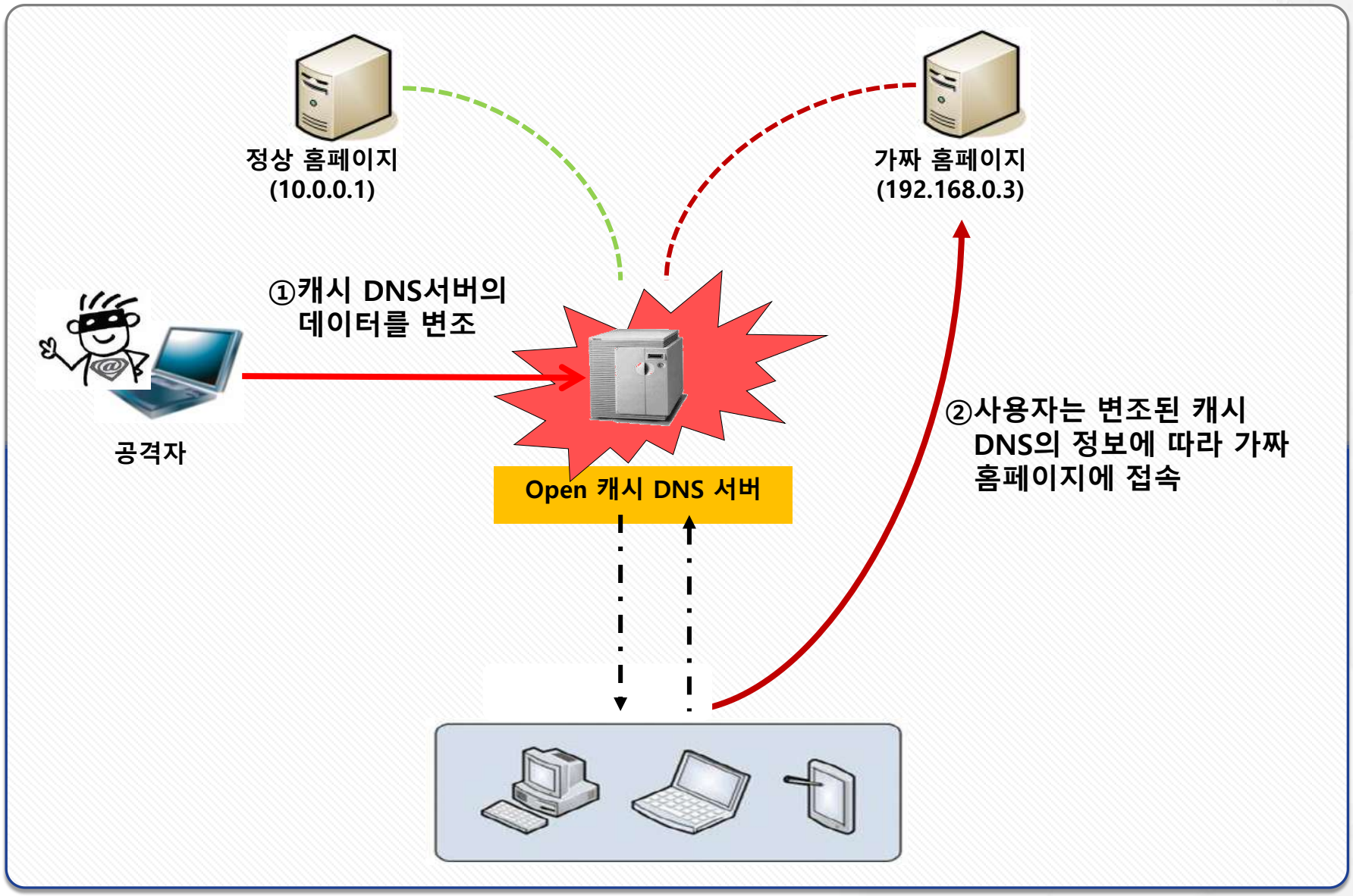
  - **Cache Poisoning**
    - Open 캐시 DNS서버에 자신의 도메인을 비롯하여 특정 도메인의 잘못된 값을 저장해 두고 무작위 클라이언트가 자신의 도메인에 접속하도록 유도합니다.
    - Cache poisoning 공격이 위험한 이유는 엉뚱한 사이트로 접속하게 하고 사적인 정보를 훔치는 Pharming(파밍) 공격을 유도할 수 있기 때문에 위험합니다.

KISA

[참고] Open 캐시 DNS 서버를 이용한 DDoS 공격 원리

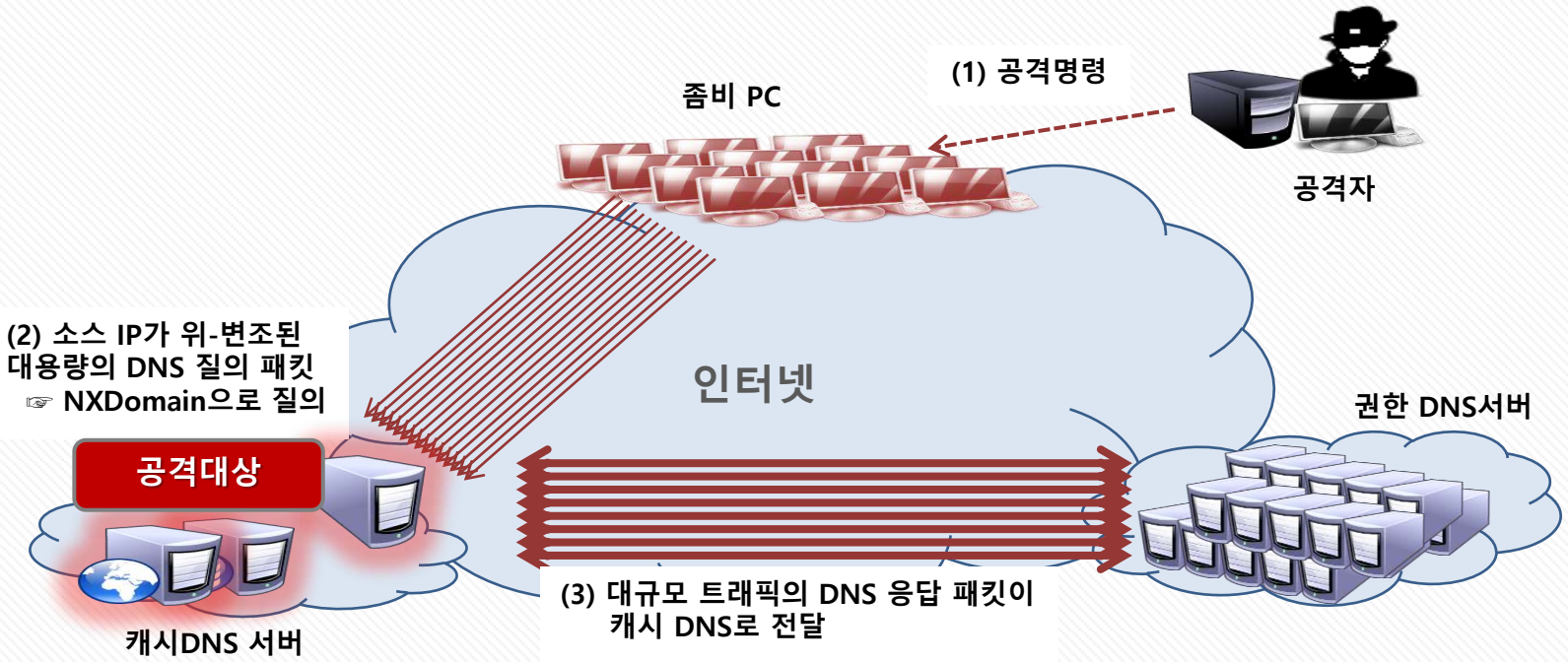


[참고] Open 캐시 DNS 서버를 이용한 캐시 포이즈닝 원리



[참고] DNS 기반 DDos 공격분류 및 방법

1. Basic NXDomain Attack



※ 출처 : https://community.infoblox.com/blogs/2014/10/02/dns-based-ddos-attacks-what's-name

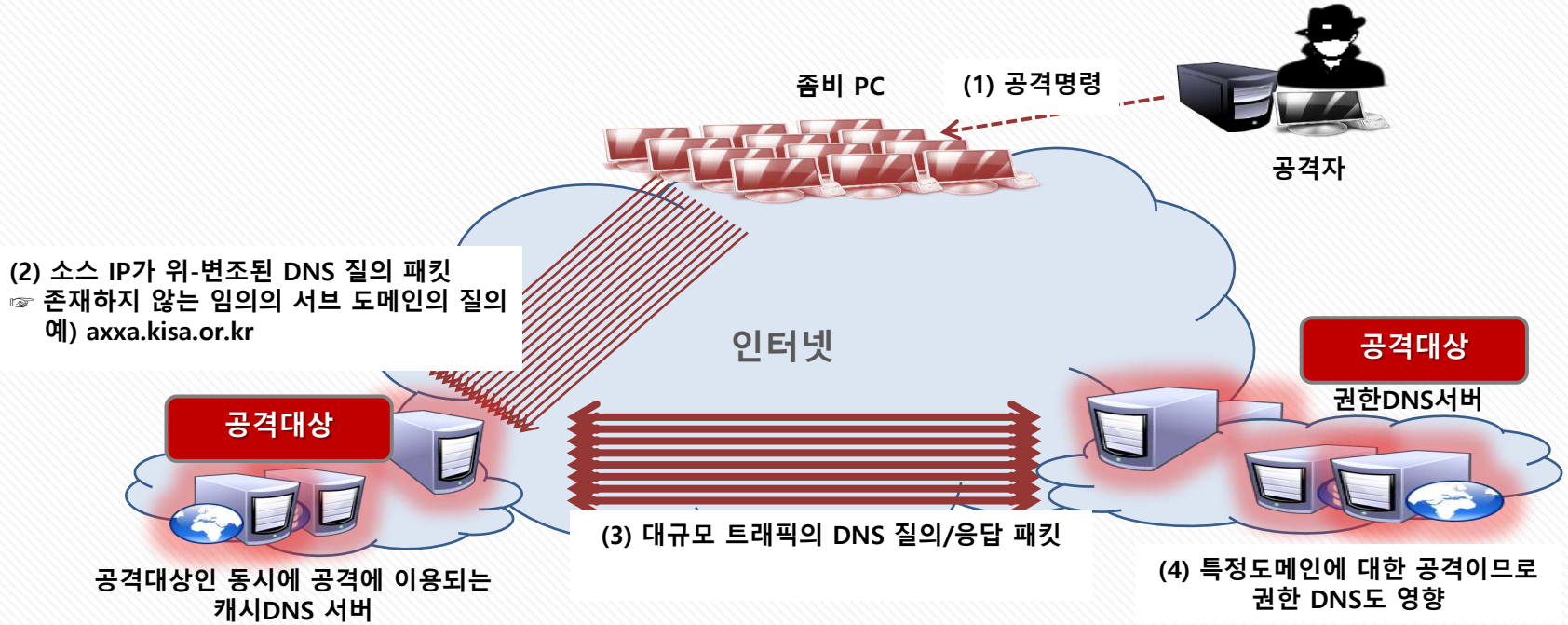
공격대상	영향	권장사항
캐시 DNS서버	NXDomain 응답결과 값이 캐시에 저장되어 자원이 소모가 됨. 캐시 DNS 서버가 느리게 응답	캐시 DNS서버를 외부에 Open 되지 않도록 설정

※ NXDomain : Not eXist Domain, 실제로 존재하지 않는 도메인네임. 존 파일에 없는 도메인 네임

KISA

[참고] DNS 기반 DDos 공격분류 및 방법

2. Random Sub-domain attacks on Legitimate Domains



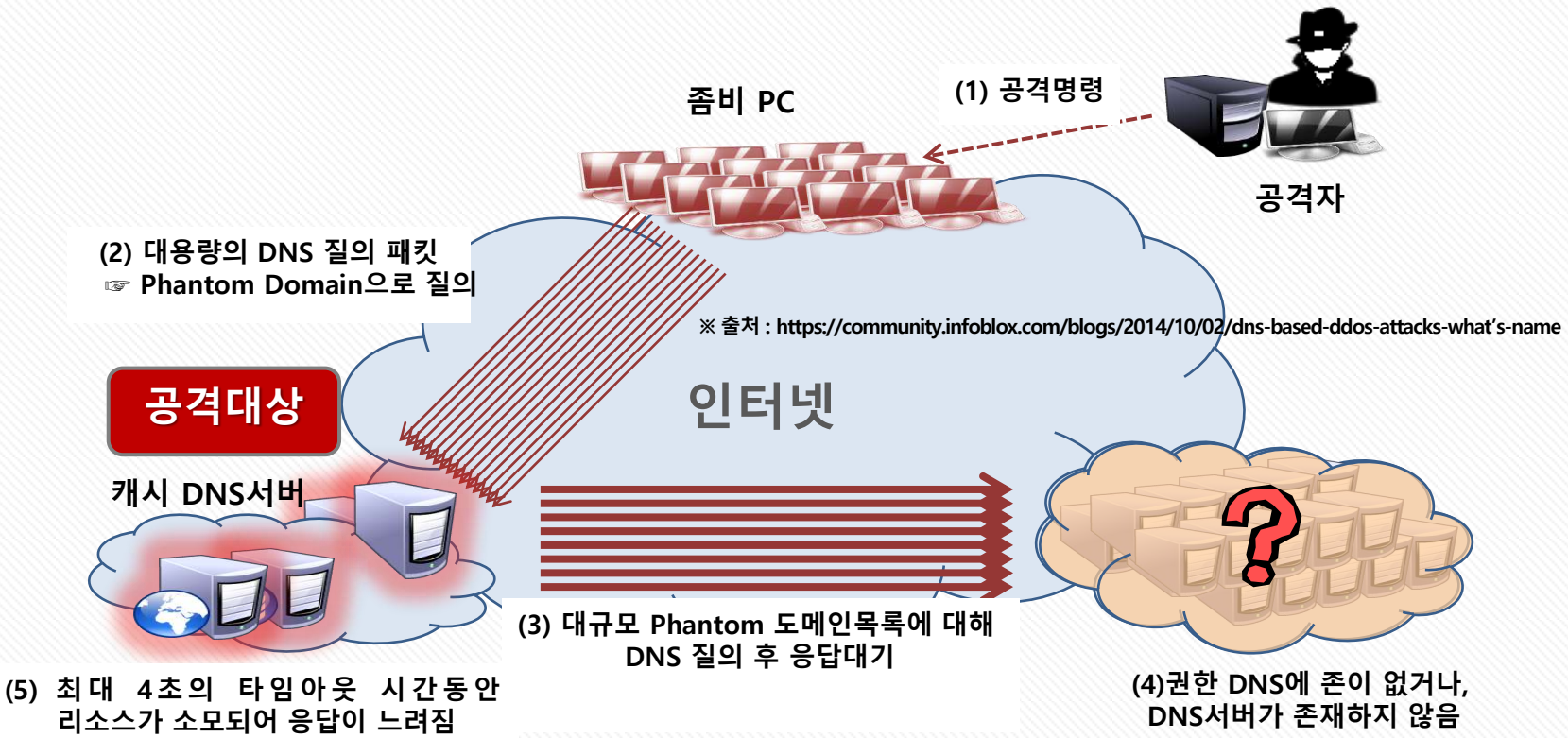
※ 출처 : https://community.infoblox.com/blogs/2014/10/02/dns-based-ddos-attacks-what's-name

공격대상	영향	권장사항
캐시 DNS서버	NXDomain 응답결과 값이 캐시에 저장되어 자원이 소모가 됨. 캐시 DNS 서버가 느리게 응답	캐시 DNS서버를 외부에 Open 되지 않도록 설정
권한 DNS서버	권한 DNS의 캐시에 NXDomain 정보가 저장되어 자원이 소모 됨	

KISA

[참고] DNS 기반 DDos 공격분류 및 방법

3. Phantom Domain Attacks



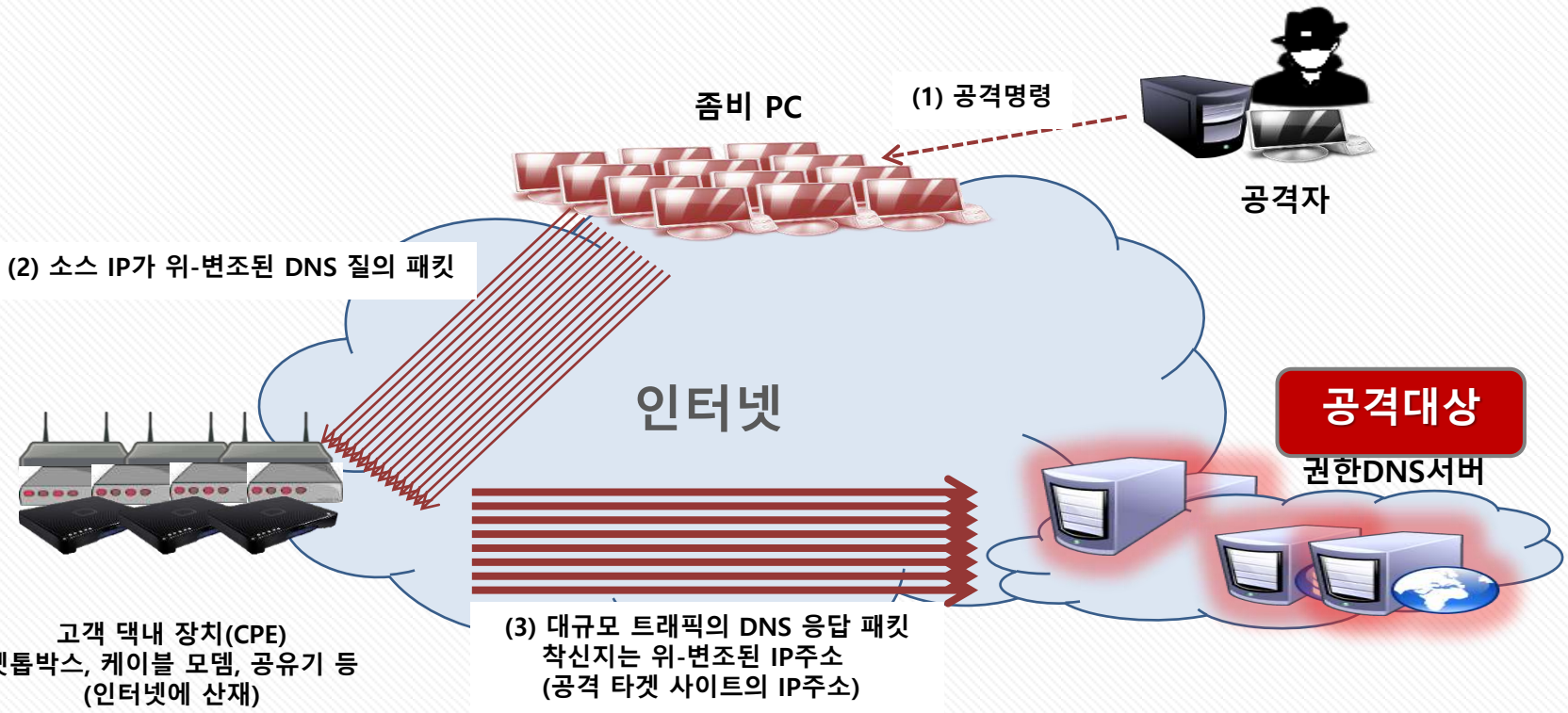
공격대상	영향	권장사항
캐시 DNS서버	NXDomain 응답결과 값이 캐시에 저장되어 자원이 소모가 됨. 캐시 DNS 서버가 느리게 응답	캐시 DNS서버를 외부에 Open 되지 않도록 설정

※ Phantom 도메인 : 상위 권한 DNS에 위임설정은 되어 있으나 실제로 네임서버를 운영하지 않거나 존이 없는 도메인. 해당 도메인에 DNS질의를 하면 최대 4초까지 타임아웃이 발생할 수 있음

KISA

[참고] DNS 기반 DDos 공격분류 및 방법

4. CPE-driven DDoS attacks in the ISP network



※ 출처 : https://community.infoblox.com/blogs/2014/10/02/dns-based-ddos-attacks-what's-name

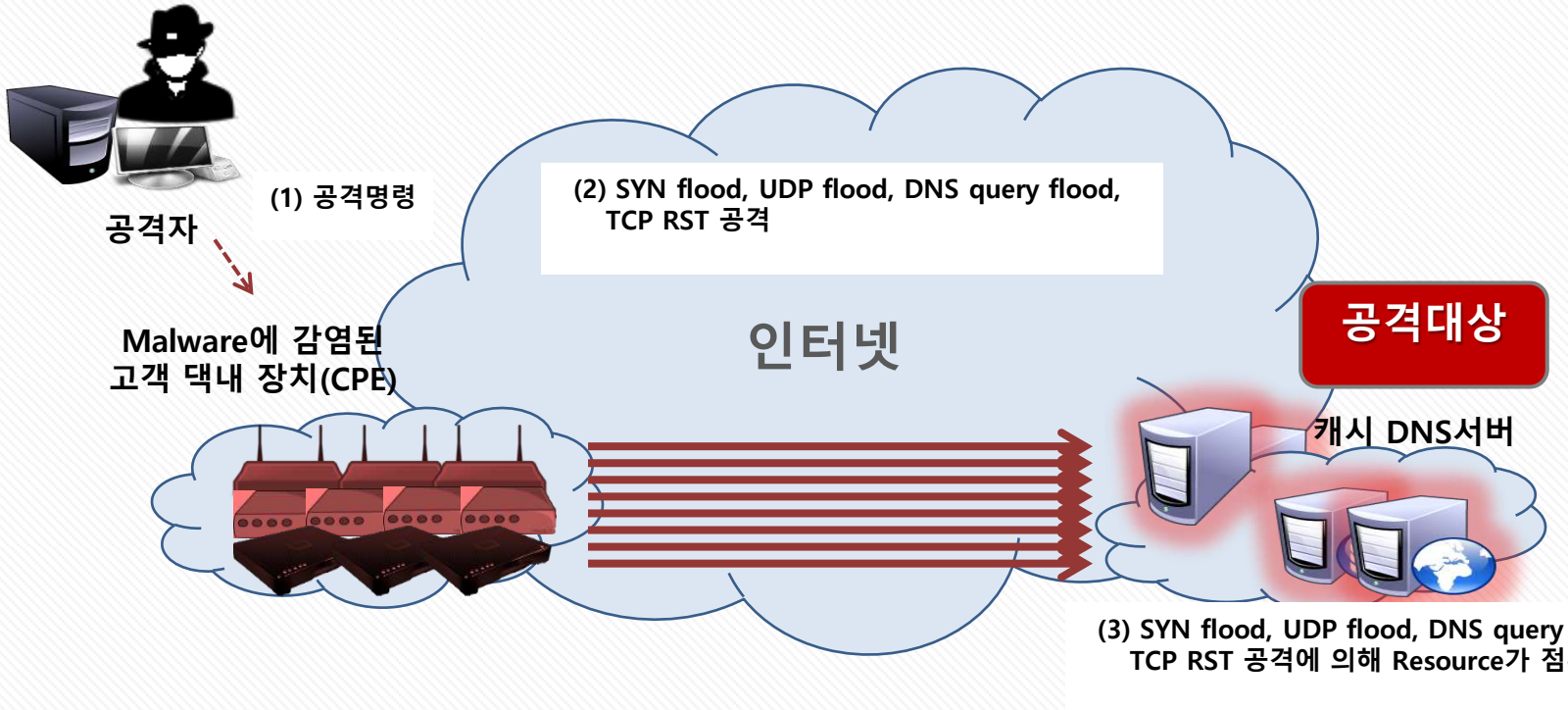
공격대상	영향	권장사항
캐시 DNS서버	인터넷을 통해 질의한 DNS응답 결과값이 공격대상으로 전달되어 공격대상 서버를 느리게 하거나 다운 시킴	CPE 장치에 반드시 암호를 설정하도록 권고

※ 고객 댁내 장치(customer premises equipment) : CPE란 통신 서비스 제공 회사가 공급하여 해당 회사의 네트워에 연결되어 있는 종단 장치를 말하는데, 이러한 예로는 단말기, 셋톱박스, 케이블 모뎀 및 ADSL 모뎀 등이 있다. 포트 포워딩, 캐시전용 DNS서버, DNS프록시 설정에 open된 장비 특징이 있다

KISA

[참고] DNS 기반 DDos 공격분류 및 방법

5. DDoS attacks using Malware-infected CPE devices



※ 출처 : https://community.infoblox.com/blogs/2014/10/02/dns-based-ddos-attacks-what's-name

공격대상	영향	권장사항
캐시 DNS서버	SYN flood, UDP flood, DNS query flood, TCP RST 공격 등으로 캐시 DNS 리소스가 소모됨	CPE 장치에 반드시 암호를 설정하도록 권고

KISA

OPEN 캐시 DNS 이용제한 설정

- **리커시브 응답제공 제한 설정**
  - 리커시브 응답제공 허용 호스트 리스트 정의
  - 정의된 호스트에 한하여 리커시브 응답 제공 허용 설정
    - 디폴트로 리커시브 기능 해제 상태로 동작
- **BIND DNS에서의 리커시브 응답제공 제한 설정 방법**

```
<named.conf>

// 내부 호스트의 서브 네트워크 지정
acl internal-hosts {
    localhost;
    localnets;
    192.168.1.0/24;
    192.168.5.0/24;
};

options {
      ...
      allow-query { any; };
          // authoritative 네임서버 경우, 명시적 설정
      allow-recursion { internal-hosts; };  // 리커시브 서비스 제한 설정
};
```

OPEN 캐시 DNS 이용제한 설정

allow-recursion 설정

- recursion 옵션은 설정된 대상에게만 리커시브 동작을 수행합니다.
- 리커시브 동작은 자신이 소유한 도메인 외의 DNS질의가 들어오면 해당 질의에 대한 답을 주기 위해 DNS 질의과정을 수행 하는 것을 말합니다.

```
options {
    directory "/var/named";
    dump-file "/var/tmp/named_dump.db";
    statistics-file "/var/tmp/named.stats";
    pid-file "/var/run/named.pid";
    allow-query { any; };                    // authoritative 네임서버 경우, 명시적 설정
    allow-recursion {OOO.OOO.OO.OO;};  // 질의를 허용할 네트워크 대역 또는 IP 주소
};

zone "." {
    type hint;
    file "named.cache";
};


... 후략 ...
```

Q & A

기타 문의 : in_chk@nic.or.kr

감사합니다