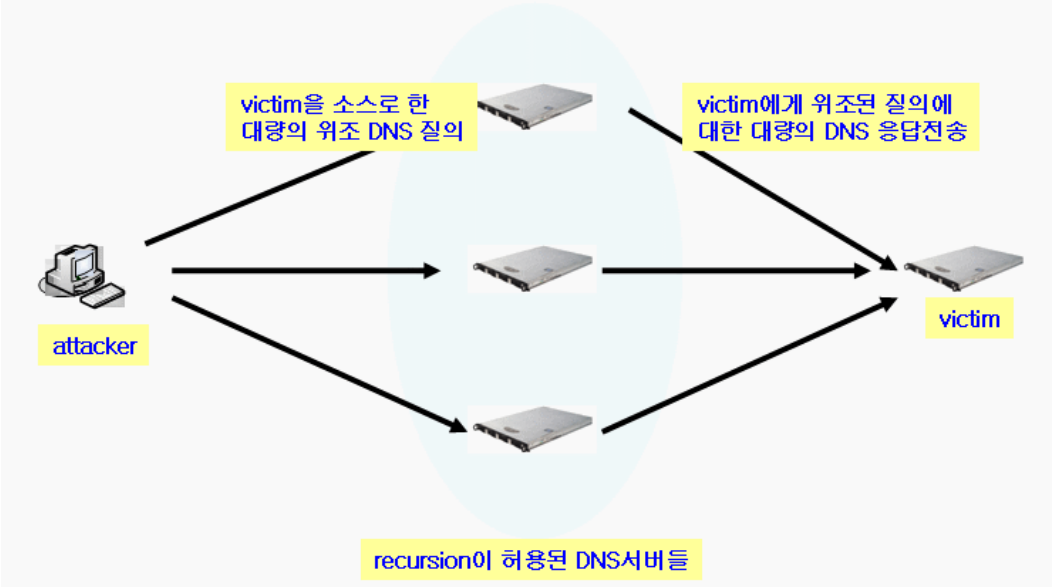
오늘과내일 홍석범(antihong@tt.co.kr)

보이지 않는 위협, recursion이 허용된 DNS

우리가 흔히 아무 생각 없이 설치해서 사용하고 있는 DNS와 관련하여 최근 새로운 위협이 조심스럽게 논의되고 있다. 이는 몇 년전까지 분산서비스거부공격(DDoS)으로 사용되어왔던 icmp 기반의 스머프(smurf) 공격과 유사한 방식으로, recursion이 허용(open)되어 있는 DNS서버를 일종의 증폭기(ampilfier) 또는 좀비(zombie)로 악용하여 특정 시스템 또는 네트워크에 대한 공격으로서 아래의 그림을 기초로 공격의 작동방식에 대해 설명해 보도록 하겠다.



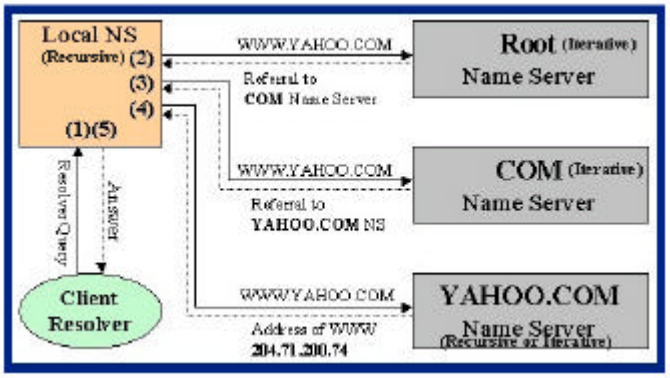
먼저, 공격자는 순환질의(recursive query)가 허용되어 있는 각각의 DNS서버를 목적지로 하여 도메인에 대한 많은 udp 질의를 한다. 단, 이때 소스ip는 공격자의 실제 ip가 아닌 victim의 ip로 위조하도록 하여야 한다. 참고로 udp나 icmp는 tcp와 달리 소스 ip 위조가 쉽고 대량의 패킷 생성이 가능하기 때문에 공격자의 입장에서는 유용한 프로토콜이 될 것이다. 이제 질의를 받은 각각의 DNS 서버는 요청받은 질의에 대해 여러 가지 경로로 적절한 답을 찾은 후 실제 attacker가 아닌 해당 질의의 소스 ip인 victim으로 일제히 응답하게 된다. 만약 공격자가 짧은 시간에 많은 질의를 하고, 또한 하나가 아닌 많은 DNS서버를 이용하면 할수록 그리고 응답 패킷의 사이즈를 최대(512byte)로 하도록 질의를 조작할 수 있다면 공격자 입장에서 공격의 효과는 더욱 클 것이다.

한 조사에 의하면 전 세계 약 750만개의 DNS 서버가 공인IP로 연결되어 있는데 이 중 약 75%에 달하는 DNS 서버가 별도의 보안 설정 없이 recursive 질의가 허용되어 있다고 한

다. 따라서 공격자가 수 백만개의 recursion이 허용되어 있는 전 세계의 DNS서버를 일종의 zombie로 하여 특정 네트워크에 대한 분산서비스거부공격을 한다면 어렵지 않게 수 Gigabyte의 트래픽을 생성할 수 있게 되므로 그 피해는 엄청날 것이다.

참고 : recursion(순환질의)이란?

순환질의란 쉽게 말해서 자신의 PC에 설정해서 사용하는 DNS에서, 도메인에 대한 질의를 받아 응답을 찾아가는 과정이라고 생각하면 된다. 아래 그림에서는 자신의 PC에서 www.yahoo.com을 접속하기 위해 어떤 과정을 거쳐 답을 찾아오는지에 대해 설명하고 있는데, 최상의 root 네임서버에서부터 .com에 대한 위임권한을 가진 DNS서버를 거쳐 yahoo.com 도메인의 위임권한을 가진 DNS서버를 통해 www.yahoo.com의 IP를 찾아오고 있는 것을 알 수 있다.



위의 과정은 dig에서 제공하는 +trace 옵션을 이용하면 아래와 같이 간단히 확인 가능하다.

```
# dig yahoo.com +trace
; <<>> DiG 9.2.1 <<>> yahoo.com +trace
;; global options:  printcmd
.                       514596  IN      NS      F.ROOT-SERVERS.NET.
.                       514596  IN      NS      G.ROOT-SERVERS.NET.
.                       514596  IN      NS      H.ROOT-SERVERS.NET.
.                       514596  IN      NS      I.ROOT-SERVERS.NET.
.                       514596  IN      NS      J.ROOT-SERVERS.NET.
.                       514596  IN      NS      K.ROOT-SERVERS.NET.
.                       514596  IN      NS      L.ROOT-SERVERS.NET.
.                       514596  IN      NS      M.ROOT-SERVERS.NET.
.                       514596  IN      NS      A.ROOT-SERVERS.NET.
.                       514596  IN      NS      B.ROOT-SERVERS.NET.
.                       514596  IN      NS      C.ROOT-SERVERS.NET.
.                       514596  IN      NS      D.ROOT-SERVERS.NET.
```

```
.                       514596  IN      NS      E.ROOT-SERVERS.NET.
;; Received 356 bytes from 211.47.66.90#53(211.47.66.90) in 9 ms

com.                    172800  IN      NS      F.GTLD-SERVERS.NET.
com.                    172800  IN      NS      G.GTLD-SERVERS.NET.
com.                    172800  IN      NS      H.GTLD-SERVERS.NET.
com.                    172800  IN      NS      I.GTLD-SERVERS.NET.
com.                    172800  IN      NS      J.GTLD-SERVERS.NET.
com.                    172800  IN      NS      K.GTLD-SERVERS.NET.
com.                    172800  IN      NS      L.GTLD-SERVERS.NET.
com.                    172800  IN      NS      M.GTLD-SERVERS.NET.
com.                    172800  IN      NS      A.GTLD-SERVERS.NET.
com.                    172800  IN      NS      B.GTLD-SERVERS.NET.
com.                    172800  IN      NS      C.GTLD-SERVERS.NET.
com.                    172800  IN      NS      D.GTLD-SERVERS.NET.
com.                    172800  IN      NS      E.GTLD-SERVERS.NET.
;; Received 499 bytes from 192.5.5.241#53(F.ROOT-SERVERS.NET) in 6 ms

yahoo.com.              172800  IN      NS      ns1.yahoo.com.
yahoo.com.              172800  IN      NS      ns2.yahoo.com.
yahoo.com.              172800  IN      NS      ns3.yahoo.com.
yahoo.com.              172800  IN      NS      ns4.yahoo.com.
yahoo.com.              172800  IN      NS      ns5.yahoo.com.
;; Received 197 bytes from 192.35.51.30#53(F.GTLD-SERVERS.NET) in 165 ms

yahoo.com.              300     IN      A       216.109.112.135
yahoo.com.              300     IN      A       66.94.234.13
yahoo.com.              172800  IN      NS      ns1.yahoo.com.
yahoo.com.              172800  IN      NS      ns2.yahoo.com.
yahoo.com.              172800  IN      NS      ns3.yahoo.com.
yahoo.com.              172800  IN      NS      ns4.yahoo.com.
yahoo.com.              172800  IN      NS      ns5.yahoo.com.
;; Received 229 bytes from 66.218.71.63#53(ns1.yahoo.com) in 155 ms
```

따라서, DNS 서버를 운영하는 관리자라면 자신의 서버가 recursion이 허용되어 있지는 않은지 확인해 보아야 한다. 참고로 recursion이 허용되어 있다면 DNS cache poisoning 공격에도 취약하다 할 수 있다.

그렇다면, 자신의 DNS가 recursion이 허용되어 있는지 여부는 어떻게 알 수 있을까? 간단히 해당 DNS 서버에 임의의 도메인을 질의하였을 때 정상적인 응답이 오는지 여부를 확인하면 되는데 아래 두 질의의 차이점을 각각 확인해 보기 바란다.

```
# nslookup www.yahoo.co.kr nis.dacom.co.kr

Server:         nis.dacom.co.kr
Address:        164.124.101.31#53

Non-authoritative answer:
*** Can't find www.yahoo.co.kr: No answer
```

위의 질의는 nis.dacom.co.kr DNS서버에 www.yahoo.co.kr에 대한 정보를 질의한 것으로 위임권한이 있는 도메인이 아니기 때문에 nis.dacom.co.kr에서 클라이언트가 요구한 응답을 하지 않는 것을 알 수 있다.

```
# nslookup www.yahoo.co.kr ns.dacom.co.kr

Server:         ns.dacom.co.kr
Address:        164.124.101.2#53

www.yahoo.co.kr canonical name = yahoo.co.kr.
Name:   yahoo.co.kr
Address: 222.231.19.227
Name:   yahoo.co.kr
Address: 202.43.214.151
```

하지만 위의 경우는 질의한 도메인에 대한 정보를 찾아서 알려주고 있다는 것을 알 수 있다. 즉, ns.dacom.co.kr은 recursion이 허용된 DNS라는 것을 알 수 있으며 이때의 응답은 위와 같이 단순한 IP 뿐만 아니라 MX나 SOA등 질의에 따르는 다양한 정보일 수 있을 것이다.

참고 : 자신의 DNS서버가 recursion이 허용되어있는지 여부의 확인방법

http://dns.measurement-factory.com/cgi-bin/openresolvercheck.pl 에 접속하여 테스트하여 보기 바란다. 검색하고자 하는 DNS 서버의 IP를 입력하여 쿼리전송을 클릭하면 되는데, 결과가 closed 라고 나온다면 보안 설정이 잘 된 것이고, open 이라고 나온다면 허용되어 있는 것이므로 주의가 요망된다.

그렇다면 recursion을 어떻게 적절히 제한하여야 할까? 가장 많은 사용자 층을 확보하고 있는 bind dns의 경우 설치를 하면 기본적으로 recursion이 허용되어 있는데, 이를 금지 또는 제한하는 방법은 매우 간단하다. 이는 주 설정 파일인 /etc/named.conf에서 다음과 같이 두 가지 방법으로 설정할 수 있다.

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

또는

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

bind DNS를 설치하면 recursion 기능이 기본적으로 모두 허용되어 있는데, 위와 같이 차단하게 되면 모든 recursion을 거부하게 된다. 만약 특정 IP 대역에 대해서만 recursion을 허용하도록 하려면 다음과 같이 설정 파일인 named.conf에서 허용할 IP 또는 IP 대역을 정의하면 된다. 아래의 경우 127.0.0.1 과 192.168.2.0/24 대역에서만 recursion을 허용하도록 제한하는 예를 보여주고 있다.

```
options {
                allow-recursion {127.0.0.1; 192.168.2.0/24; };
};
```

따라서 각 dns 서버의 용도와 서비스 내용을 잘 파악하여 recursion을 거부하거나 만약 필요하다면 특정 ip 대역에서만 허용하도록 설정하여 혹시나 모를 잠재 위협에 대응하도록 보안 설정을 하는 것이 좋다.