

МИНИМУМ УСИЛИЙ

НА ЗАЩИТУ

DNS



СЕРГЕЙ РОПЧАН

По статистике (некоторых security teams) из 100 % серверов – 63% работает с неправильно настроенной службой DNS, в контексте сетевой безопасности, конечно. Узнав столь печальную статистику, я решил провести свой собственный анализ.

```
[main.target.com]
$ORIGIN target.com.
@      1D      IN      SOA      ns root (
                2001081109      ;serial
                8H      ;refresh
                2H      ;retry
                1W      ;expiry
                1D )      ;minimum

                1D      IN      NS       ns
                1D      IN      NS       r1.ns.net.
                1D      IN      NS       r2.ns.com.
                1D      IN      MX       20 m1.ns.net
                1D      IN      MX       10 m2.ns.com
                1D      IN      MX       10 main
c4      1D      IN      A        192.5.62.78
admin   1D      IN      A        192.5.62.74
localhost 1D      IN      A        127.0.0.1
mail    1D      IN      CNAME    main
proxy   1D      IN      CNAME    main
www     1D      IN      CNAME    main
c1      1D      IN      A        192.5.62.75
c2      1D      IN      A        192.5.62.76
c3      1D      IN      A        192.5.62.77
ns      1D      IN      A        192.5.62.73
ftp     1D      IN      CNAME    main
@      1D      IN      SOA      ns root (
                2001081109      ;serial
                8H      ;refresh
                2H      ;retry
                1W      ;expiry
                1D )      ;minimum
```

И вот что у меня вышло – из 100 серверов, выбранных мною наугад из различных сегментов Интернета, 57 серверов оказались с неправильно настроенной службой имен.

Под словом «неправильно» я подразумеваю, например, получение файла пересылки (transfer) зоны (эта информация является наиболее важной в аспекте безопасности DNS), на основе которого, как известно, можно построить схему внутренней сети исследуемой системы, так как мы получаем полную информацию о инфраструктуре внутренней сети (имена хостов, ip-адреса, почтовые сервера, вышестоящие сервера имен, псевдонимы хостов и т. д.). Вот наглядный пример: выбираем цель, например, www.target.ru; как известно, для работы с DNS подходит nslookup, которая входит в состав большинства ОС. Итак:

```
fenix# nslookup
Nameserver 192.145.45.1

>server www.target.com
>ls -d target.com
```

и если в ответ мы получаем что-то похожее на:

(данная информация является файлом пересылки зоны), то это означает, что администратор системы не предпринял никаких усилий, хотя бы минимальных, (которые я опишу в этой статье) для защиты DNS то ли от незнания, то ли от неумения. Но это все неважно, так как речь в этой статье будет идти не о причинах, побудивших или не побудивших администратора к такому безразличию, которое в свою очередь может привести к плачевным результатам.

Я хочу попытаться показать, как хотя бы «немного», (если это слово можно считать корректным в контексте сетевой безопасности) защитить свой DNS-сервер. Я ни в коем случае не претендую на полноту изложенной здесь информации, так как этот вопрос требует намного более пристального внимания со стороны администраторов и выходит за пределы этой статьи; более детальную информацию о защите DNS можно получить в списке дополнительных материалов в конце этой статьи.

Итак, начнем. Прежде всего следует сказать, что все описанное ниже будет относиться к настройке bind 8.x.x. Работать мы будем непосредственно с файлами конфигурации bind, а если быть точным, то в основном с named.conf (/etc/namedb/named.conf).

Первое, что необходимо сделать в `named.conf` – это определить список доступа `acl`, в который будут входить «доверенные» сети и хосты (так же я советую включить в этот список `ip`-адреса первичных DNS-серверов, которые делегировали на вас управление каким-либо доменом). Создаем секцию `acl` в самом начале файла (`named.conf`):

```
acl "trusted" {
    localhost;
    192.168.3.0;
};
```

таким образом, «доверенными» являются `localhost` и сеть (`192.168.3.0`), хосты которой пользуются услугами нашего сервера имен.

Далее в секцию `options` данного файла вносим:

```
options {
    ...
    #определения прав доступа по умолчанию
    allow-query { trusted };          ## разрешить запросы группе
                                     ## «доверенных» хостов
    allow-transfer { none };         ## запретить пересылку
                                     ## файла зоны кому-либо
    allow-recursion { trusted };     ## разрешить рекурсивные
                                     ##запросы группе «доверенных» хостов
    ...
};
```

Данные директивы определяют поведение `bind` по умолчанию (запросы разрешены только для «доверенных» хостов, трансфер зон запрещен, рекурсивные запросы разрешены только для «доверенных» хостов), эти директивы можно переопределить для каждой зоны в соответствующей секции `zone`. Тут я советую следовать такому «правилу»: зонам, для которых наш сервер имен является вторичным (`slave`), необходимо явно разрешить запросы с любым `ip`-адресом источника, для первичных зон (`master`), кроме (`0.0.127.in-addr.arpa`), дополнительно к этому необходимо разрешить трансфер зонных файлов, то есть вот что у нас выходит:

```
zone "example.com" {
    type master;
    file "primary/example.com";

    #переопределение прав доступа, заданных по умолчанию в options
    allow-query { any; };
    allow-transfer { localhost; 192.168.3.0; };
};

zone "3.168.192.in-addr.arpa" {
    type slave;
    file "secondary/0.126.in-addr.arpa";
    masters { 192.168.3.1; };

    #переопределение прав доступа, заданных по умолчанию в options
    allow-query { any; };
    allow-transfer { localhost; };
};
```

В том случае если ваш сервер является, например, сервером имен (единственным) для локальной сети (`intranet`), т. е. не является шлюзом во внешний мир, то можно настроить `bind` таким образом, чтобы он разрешал запросы только с «доверенных» хостов. Для этого

необходимо в `named.conf` внести описание зоны `bind`:

```
zone "bind" chaos {
    type master;
    file "primary/bind";
    allow-query { trusted };
    allow-transfer { none };
};
```

и создать, собственно, файл описания зоны:

```
$TTL 3600
$ORIGIN bind.
@ 1D      CHAOS      SOA localhost. root.localhost. {
    1      ;serial
    3H     ;refresh
    1H     ;retry
    1W     ;expire
    1D )   ;minimum
CHAOS NS  localhost/
;
```

Доступ мы ограничили; я думаю, со мной многие согласятся, что «защита» без ведения логов – это не защита, так как лучше знать своего врага в лицо. Разработчики `bind`'а позаботились и об этом: существует возможность ведения лог-файлов. Вот основные приемы (создаем новую секцию `logging`):

```
logging {

    # определяем канал по умолчанию для ведения
    # логов запросов к bind'у
    channel default_ch {
        file "/var/log/named.log";
        serverity info;          #уровень важности регистра
                                 #ционной информации
        print-time yes;         #регистрировать время
        print-category yes;     #регистрировать категорию
    };

    channel security_ch {

    #определяем канал для ведения логов по защите
        file "/var/log/security.log";
        serverity info;
        print-time yes;
        print-category yes;
    };

    ##инициализация каналов
    category default { default_ch; };
    category security { security_ch; };
};
```

Вот и все. Как было сказано выше, эта информация не является исчерпывающей по данному вопросу, так как полная настройка системы защиты DNS строится на создании для службы DNS `chrooted environment`, создание для нее `sandbox`'а и т. д., вплоть до защиты системы в целом.

Рекомендуемые материалы:

1. М. Канавалова. *Securing Bind*.
2. RFC по DNS.
3. *man bind*.
4. Criag H. Rowland. *Securing Bind*.