**Защита сервера DNS - Настройка безопасности**

Денис Колисниченко

Конфигурируя сервер, администраторы часто забывают правильно настроить службу DNS. После такой настройки служба DNS работает корректно: IP-адреса разрешаются в имена компьютеров, а символьные имена без проблем преобразуются в IP-адреса. На этом большинство администраторов и останавливаются: главное, чтобы работало. Работать-то оно работает, но неправильно настроенный сервер DNS может стать огромной дырой в системе безопасности компании. Одно дело, когда сервер DNS обслуживает локальную сеть без выхода в Интернет: даже, если кто-то и попытается "взломать" сервер, то вычислить "хакера" довольно просто. А вот, если сеть предприятия подключена к Интернет, то узнать, кто же пытался взломать (или взломал) вашу сеть довольно сложно. Ущерб от взлома может обойтись компании в кругленькую сумму.

Прежде, чем приступить к взлому сети или отдельной системы злоумышленник (или группа злоумышленников) пытается собрать как можно больше информации: имена компьютеров сети, имена пользователей, версии установленного программного обеспечения. Целой кладовой полезной для взломщика информации станет неправильно настроенная служба DNS (BIND). Рассмотрим небольшой пример: запустите программу nslookup и введите команду:

ls server.com

Если администратор забыл правильно настроить трансфер зоны, то кто угодно получит список компьютеров нашей сети:

[comp2.server.com] server.com. 323.111.200.2

server.com. server = comp1.server.com server.com. server = comp2.server.com server.com.

server = comp3.server.com mail 323.111.200.17 gold 323.111.200.22 www.ie 323.111.200.11

jersild 323.111.200.25 comp1 323.111.200.1 comp3 323.111.200.3 parasit3 323.111.200.20

www.press 323.111.200.30 comp1 323.111.200.1 www 323.111.200.2

Примечание. Чтобы не было недоразумений, указаны несуществующие IP-адреса

Дабы не случилось непоправимого, разрешите передачу зону только одному компьютеру - вторичному серверу DNS вашей компании, если такой, конечно, имеется. В файле конфигурации сервиса named - /etc/named.conf - измените секцию options следующим образом:

options{

allow-transfer { 192.168.1.2; }; };

Вторичный сервер DNS, как правило, не передает никакой информации о зоне, поэтому обязательно укажите следующую строку в его файле конфигурации /etc/named.conf (в секции options):

allow-transfer

{ none; }

Если у вас нет вторичного сервера DNS, добавьте вышеуказанную строку в файл конфигурации основного сервера DNS.

Как любой хороший администратор, вы хотите, чтобы ваш сервер DNS быстро обслуживал запросы клиентов. Но к вашему серверу могут подключаться пользователи не из вашей сети, например, из сети конкурирующего провайдера. Тогда вас сервер будет обслуживать "чужих" клиентов. Непорядок! Опция allow-query позволяет указать адреса узлов и сетей, которым можно использовать наш сервер DNS:

allow-query { 192.168.1.0/24; localhost; };

В данном примере мы позволяем использовать наш сервер узлам из сети 192.168.1.0 и узлу localhost. Целесообразно разрешить рекурсивные запросы только из сети 192.168.1.0 и узлу localhost:

allow-recursion { 192.168.1.0/24; localhost; };

Обычно взлом любой сети начинается со сбора информации - о структуре сети, об установленном программном обеспечении и версиях этого ПО и т.д. Мы можем заставить сервер DNS сообщать не номер своей версии, а произвольное сообщение:

version

"Made in USSR";

Все вышеперечисленные опции должны быть указаны в секции options файла конфигурации named.conf:

options { allow-query

{ 192.168.1.0/24; localhost; }; allow-recursion { 192.168.1.0/24; localhost; };

allow-transfer { 192.168.1.2; }; version "Made in USSR"; }

Из соображений безопасности рекомендуется запускать все сетевые сервисы в так называемом chroot-окружении. Сейчас поясню, что это такое. Создается файловая система, повторяющая структуру корневой файловой системы, но на этой файловой системе будут только те файлы, которые необходимы для запуска нашего сетевого сервиса. Взломав сетевой сервис и получив доступ к корневой файловой системе, злоумышленник не сможет повредить всей системе в целом, поскольку он получит доступ только к файлам, которые принадлежат данному сетевому сервису. Одни сетевые сервисы могут работать в chroot-окружении, а другие - нет. Сервис BIND как раз относится к первой группе. Теперь разберемся, как все это организовывается. Вам не нужно создавать отдельный раздел на диске для каждого сетевого сервиса: нужно только создать каталог, например, root-dns, в который вы скопируете все файлы, необходимые для запуска сервера DNS. Потом, при запуске сервиса, будет выполнена команда chroot для этого сервиса, которая подменит файловую систему. А так как в каталоге root-dns, который станет каталогом /, имеются все необходимые файлы для работы bind, то для сервиса запуск и работа в chroot-окружении будет совершенно прозрачным.

Сразу нужно сказать, что настраивать chroot-окружении мы будем для девятой версии BIND, поскольку это значительно проще, чем для восьмой версии. В отличие от восьмой версии, где для настройки chroot-окружения нужно было копировать все бинарные файлы или библиотеки, необходимые для запуска BIND, для работы девятой версии достаточно скопировать только файлы конфигурации и зон, обслуживаемых сервером.

Начнем настраивать chroot-окружение для нашего сервера DNS. Создадим каталоги корневой файловой системы сервера DNS - root-dns:

mkdir -p /root-dns mkdir -p /root-dns/etc mkdir -p /root-dns/var/run/named

mkdir -p /root-dns/var/named

Остановим сервер DNS, если он запущен:

service named stop

Переместим файл конфигурации named.conf и файлы зон в каталог /root-dns:

mv /etc/named.conf /root-dns/etc/

mv /var/named/\* /root-dns/var/named/ chown named.named /chroot/etc/named.conf

chown -R named.named /root-dns/var/named/\*

Нам еще понадобится файл localtime для правильной работы сервера DNS со временем:

cp /etc/localtime

/root-dns/etc/

Защитим от редактирования и удаления файл конфигурации named.conf:

chattr +i /root-dns/etc/named.conf

Примечание. Не забудьте снять атрибут "i" перед редактированием файла конфигурации (chattr -i /root-dns/etc/named.conf)

Удалим каталоги /var/named и /var/run/named - они нам больше не нужны:

rm -rf /var/named/ rm -rf /var/run/named/

Добавим в файл /etc/sysconfig/named строку:

ROOTDIR="/root-dns/"

Все, теперь можно запустить сервер named:

service named start

Выполните команду ps -ax | grep named. Если вы увидите примерно следующее:

5380

? S 0:00 named -u named -t /root-dns/ 5381 ? S 0:00 named -u named -t /root-dns/

5382 ? S 0:00 named -u named -t /root-dns/ 5383 ? S 0:00 named -u named -t /root-dns/

значит, вы все сделали правильно.

Мы запустили сервер DNS в chroot-окружении, но на этом данная статья не заканчивается. В девятой версии BIND появилась возможность создавать подписи транзакций (TSIG - Transaction SIGnatures). Механизм TSIG работает так: сервер получает сообщение, подписанное ключом, затем подпись проверяется, если она "правильная", сервер отправляет ответ, подписанный тем же ключом.

Механизм TSIG очень эффективен при передаче информации о зоне, уведомлений об изменении зоны и рекурсивных сообщений. Согласитесь, проверка подписи надежнее, чем проверка IP-адреса. Злоумышленник может вывести вторичный сервер DNS банальной атакой на отказ, и, пока администратор будет "подымать" вторичный сервер, он заменит свой IP-адрес адресом вторичного сервера. При использовании TSIG задача злоумышленника значительно усложняется: ведь ему придется "подобрать" 128-битный MD5-ключ, а вероятность такого подбора стремится к нулю.

Итак, приступим к настройке. Остановим сервис named, если он запущен:

service

named stop

Сгенерируем общие ключи для каждой пары узлов. Общие ключи используются при "общении" первичного и вторичного серверов DNS.

[root@dns]# dnssec-keygen -a hmac-md5 -b 128 -n HOST ns1-ns2 Kns1-ns2.+157+49406

Мы используем алгоритм HMAC-MD5, 128-битное шифрование, ns1-ns2 - это имя ключа. После выполнения этой команды будет создан файл Kns1-ns2.+176+40946.private. Откройте его в любом текстовом редакторе. Вы увидите примерно следующее:

Private-key-format:

v1.2 Algorithm: 157 (HMAC\_MD5) Key: ms7dfts87Cjhj7FD9lk7a3==

Ключ "ms7dfts87Cjhj7FD9lk7a3==" и будет тем секретом, который будет передаваться между серверами. Запишите данный ключ на бумаге (которую потом нужно будет уничтожить) и удалите файлы:

rm -f Kns1-ns2.+157+49406.key rm -f Kns1-ns2.+157+49406.private

Добавим в файл /etc/named.conf (или /root-dns/etc/named.conf) первичного и вторичного серверов DNS следующие строки:

key ns1-ns2 { algorithm hmac-md5;

secret "ms7dfts87Cjhj7FD9lk7a3=="; };

Укажем серверу BIND использовать ключ ns1-ns2. Для этого в файле named.conf первичного сервера DNS добавим опцию server:

Листинг 1. Фрагмент файла named.conf первичного сервера DNS

key ns1-ns2 { algorithm hmac-md5; secret "ms7dfts87Cjhj7FD9lk7a3=="; }; # прописываем

вторичный сервер DNS - 192.168.1.2: server 192.168.1.2 { keys { ns1-ns2; }; };

options { # разрешаем передачу зоны вторичному серверу DNS allow-transfer { 192.168.1.2;

}; ? }; ?. Листинг 2. Фрагмент файла named.conf ВТОРИЧНОГО сервера DNS key ns1-ns2

{ algorithm hmac-md5; secret "ms7dfts87Cjhj7FD9lk7a3=="; }; # прописываем первичный

сервер DNS - 192.168.1.1: server 192.168.1.1 { keys { ns1-ns2; }; }; options {

# никому не передаем зону allow-transfer { none }; ? }; ?.

Можно также настроить передачу зоны "по ключу". Для этого в файле конфигурации первичного сервера DNS замените строку

allow-transfer { 192.168.1.2; };

строкой

allow-transfer { key ns1-ns2; };

Нам осталось только "спрятать" файлы конфигурации обоих серверов DNS от посторонних глаз - ведь они содержат ключи в открытом виде.

chmod 600 named.conf

Запускаем сервис named:

service named start

Теперь о безопасности вашего BIND позаботится TSIG.