DDNS 是什么？

DDNS 英文全称 Dynamic Domain Name Server，中文含义是指动态域名服务。很多普通路由器或者智能路由器设置中，都可以找到 DDNS（动态 DNS）功能。

通俗的说，DDNS 是将用户的动态 IP 地址映射到一个固定的域名解析服务上，用户每次连接网络的时候客户端程序就会通过信息传递把该主机的动态 IP 地址传送给位于服务商主机上的服务器程序，服务器程序负责提供 DNS 服务并实现动态域名解析。

如果觉得这样还不好理解的话，可以简单这样理解。目前路由器拨号上网获得的多半是动态 IP,DDNS 可以将路由器变化的外网 IP 和固定的域名

国内使用较多的 DDNS 网站主要是花生壳

自己动手做 DDNS

用途

域名系统安全性扩展 (DNSSEC) 密钥生成工具。

语法

dnssec-keygen [-a algorithm] [-b keysize] [-n nametype] [-c class] [-e] [-f flag] [-g generator] [-h] [-k] [-p protocol] [-r randomdev] [-s strength] [-t type] [-v level] [name]

描述

dnssec-keygen 命令为 DNSSEC（安全 DNS）生成密钥。它还可以生成用于事务签名 (TSIG) 的密钥。

标志

项目 描述

-a algorithm 选择加密算法。algorithm 可以具有下列其中一个值：

RSAMD5

DSA

DH (Diffie-Hellman)

HMAC-MD5

这些值区分大小写。

注：

对于 DNSSEC，RSASHA1 是必须实现的算法，并且 DSA 是首选算法。

对于 TSIG，HMAC-MD5 是必需项。

HMAC-MD5 和 DH 会自动设置 -k 标志。

-b keysize 指定密钥位数。密钥大小的选项取决于所用的算法。RSAMD5 和 RSASHA1 密钥必须为 512 到 4096 位。DH 密钥必须为 128 到 4096 位。DSA 密钥必须为 512 到 1024 位，并且必须刚好是 64 的倍数。HMAC-MD5 密钥必须为 1 到 512 位。

-n nametype 指定密钥的所有者类型。nametype 的值必须为 ZONE（适用于 DNSSEC 区域密钥）、HOST 或 ENTITY（适用于与主机关联的密钥）、USER（适用于与用户关联的密钥）或 OTHER (DNSKEY)。 这些值不区分大小写。

-c class 指示包含密钥的域名服务器 (DNS) 记录必须具有指定类。如果未指定，那么使用类 IN。

-e 如果生成 RSAMD5 或 RSASHA1 密钥，那么使用大指数。

-f flag 在 KEY 或 DNSKEY 记录的 flag 字段中设置指定标志。唯一识别的标志是 KSK（密钥签名密钥）DNSKEY。

-g generator 如果生成 DH 密钥，那么使用此生成器。可接受的值为 2 和 5。如果未指定生成器，那么在可能的情况下将使用来自 RFC 2539 的已知质数；否则，缺省值为 2。

-h 显示 dnssec-keygen 命令的选项和参数的简短摘要。

-k 生成 KEY 记录而不是 DNSKEY 记录。

-p protocol 为生成的密钥设定协议值。protocol 是 0 到 255 之间的一个数。缺省值为 3 (DNSSEC)。

-r randomdev 指定随机源。如果操作系统不提供 /dev/random 文件或等价设备，那么随机缺省源为键盘输入。randomdev 参数指定字符设备的名称或包含要使用的随机数据的文件（而不使用缺省值）。特殊值键盘指示必须使用键盘输入。

-s strength 指定密钥的强度值。strength 参数的值是一个 0 到 15 之间的数，当前在 DNSSEC 中尚未定义此参数的用途。

-t type 指定密钥的使用。type 必须是 AUTHCONF、NOAUTHCONF、NOAUTH 或 NOCONF 的其中一项。缺省值为 AUTHCONF。AUTH 指的是认证数据的能力，而 CONF 指的是加密数据的能力。在密钥类型为 NOAUTHCONF 的情况下，不会为这些算法（DH、HMAC-MD5、

HMAC-SHA1 、 HMAC-SHA224 、 HMAC-SHA256 、 HMAC-SHA384 和 HMAC-SHA512）生成任何密钥。

-v level 设置调试级别。

参数

项目 描述

name 命令行上指定的密钥的名称。对于 DNSSEC 密钥，此名称必须与要为其生成密钥的区域的名称匹配。

生成的密钥

当 dnssec-keygen 命令成功完成时，它会在标准输出中显示格式为 Knnnn.+aaa+iiiii 的字符串。这是生成的密钥的标识字符串。

nnnn 是密钥名。

aaa 算法的数字表示。

iiiii 是密钥标识（或占地面积）。

dnssec-keygen 命令会创建名称基于所显示字符串的两个文件：Knnnn.+aaa+iiiii.key 包含公用密钥，而 Knnnn.+aaa+iiiii.private 包含专用密钥。

.key 文件包含可插入（直接插入或使用 $INCLUDE 语句插入）到区域文件中的 DNSKEY 记录。.private 文件中包含特定于算法的字段。为了安全起见，此文件没有常规的读许可权。会为诸如 HMAC-MD5 之类的对称加密算法生成 .key 和 .private 文件，即使公用密钥和专用密钥相等。

示例

要为 example.com 域生成一个 768 位的 DSA 密钥，请输入以下命令：

dnssec-keygen -a DSA -b 768 -n ZONE example.com

nsupdate

是一个动态 DNS 更新工具.可以向 DNS 服务器提交更新记录的请求.它可以从区文件中添加或删除资源记录,而不需要手动进行编辑区文件.

下面是使用方法：

nsupdate [-d] [[-y keyname:secret] [-k keyfile]] [-v]

[filename]

-d 调试模式.

-k 从 keyfile 文件中读取密钥信息.

-y keyname 是密钥的名称,secret 是以 base64 编码的密钥.

-v 使用 TCP 协议进行 nsupdate.默认是使用 UDP 协议.

输入格式:

nsupdate 可以从终端或文件中读取命令.每个命令一行.一个空行或一个"send"命令,则会将先前

输入的命令发送到 DNS 服务器上.

命令格式:

server servername [port]

发送请求到 servername 服务器的 port 端口.如果不指定 servername,nsupdate 将把请求发送给

当前去的主 DNS 服务器.

如:

> server 192.168.0.1 53

local address [port]

发送 nsupdate 请求时,使用的本地地址和端口.

zone zonename

指定需要更新的区名.

class classname

指定默认类别.默认的类别是 IN.

key name secret

指定所有更新使用的密钥.

prereq nxdomain domain-name

要求 domain-name 中不存在任何资源记录.

prereq yxdomain domain-name

要求 domain-name 存在,并且至少包含有一条记录.

prereq nxrrset domain-name [class] type

要求 domain-name 中没有指定类别的资源记录.

prereq yxrrset domain-name [class] type

要求存在一条指定的资源记录.类别和 domain-name 必须存在.

update delete domain-name [ttl] [class] [type [data…]]

删除 domain-name 的资源记录.如果指定了 type 和 data,仅删除匹配的记录.

update add domain-name ttl [class] type data…

添加一条资源记录.

show

显示自 send 命令后,所有的要求信息和更新请求.

send

将要求信息和更新请求发送到 DNS 服务器.等同于输入一个空行.

nsupdate 示例:

```
# nsupdate

> server 127.0.0.1

> update delete www.centos.bz A

>

> update add www.centos.bz 80000 IN A 192.168.0.2

> update add 2.0.168.192.in-addr.arpa 80000 PTR A www.centos.bz

> send

> quit
```

DDNS 的安装与配置

1.

安装

2.

yum install bind bind-chroot

1.

配置正向解析

2.

listen-on port 53 { ip.xx.xx.xx; };

//listen-on-v6 port 53 { ::1; };

```
allow-query { any; };

... ...

zone "luck.me." IN {

        type master;

        file "luck.me.ndb";

};
```

luck.me.ndb

```
        $ORIGIN .

        $TTL 5 ; 5 seconds

        luck.me IN SOA luck.me. rname.invalid. (

        3 ; serial
```

86400 ; refresh (1 day)

3600 ; retry (1 hour)

604800 ; expire (1 week)

10800 ; minimum (3 hours)

)

NS luck.me.

A 127.0.0.1

$ORIGIN luck.me.

* A 127.0.0.1

1.

创建秘钥

2.

dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 128 -n HOST luck.me.

Kluck.me* .key .private

1.

配置动态 DNS

2.

named.conf

allow-update { key "luck_ddns"; };

include "/etc/luck_ddns.key";

key "luck_ddns" {

algorithm hmac-md5;

secret "xxxxxxxx===";

};

1.

查询验证

2.

dig @192.168.xx.xx A luck.me

dig @192.168.xx.xx A xxx.luck.me

nsupdate

> server 192.168.1.11

> zone luck.me

> key key_name xxxx---key---==

> update add t.luck.me 100 A 1.2.3.5

> send

> quit

添加动态解析

nsupdate

server 192.168.xx.xx