

Lab 7.1 Dynamic DNS

IN715 Networks Three

September 2, 2014

Introduction

DHCP presents a problem for DNS services, since we can't always be sure what IP address a particular machine will be assigned. For servers we can get around this using static IP configuration or reserved addresses in DHCP, but once we have a large number of machines involved this process becomes more difficult. Dynamic DNS updating can solve this problem.

With dynamic DNS we configure our DNS server to receive zone updates from other machines. This opens up a significant security risk, since a malicious attacker could "update" the address of a critical server. This risk is significant enough that OpenBSD's default DHCP server doesn't support this functionality. We will install the standard ISC server and then reduce the risk by using a security mechanism called *TSIG* to allow only our DHCP servers to submit DDNS updates.

You must have properly functioning DHCP and DNS servers running before attempting this lab.

1 Install the ISC dhcp server

Install the current ISC dhcpd with the command

```
pkg_add -r ftp://mirror.es7.net/pub/OpenBSD/5.5/packages/i386/isc-dhcp-server-4.2.5.1p0.tgz
```

Then stop the current server with the command

```
/etc/rc.d/dhcpd stop
```

2 Generate the TSIG key

We will use TSIG to authenticate DNS updates from our DHCP server. TSIG uses a symmetric encryption key that we generate with the command

```
dnssec-keygen -a HMAC-MD5 -b 128 -r /dev/urandom -n USER DDNS_UPDATE
```

This will produce two files: `Kddns_update.*.key` and `Kddns_update.*.private`. Copy the key from the private file. Look for a line like

```
Key: 8aEQAk+HBZVFUaZ+Bk.yQ==
```

Everything after the `Key:` is the key value.

Create a file called `ddns.key` containing the following:

```
key DDNS_UPDATE {
    algorithm HMAC-MD5.SIG-ALG.REG.INT;
    secret "<your key value>";
};
```

Set the permissions on the new file

```
chmod 640 ddns.key
```

and then copy the file to the appropriate locations and set the ownership appropriately

```
cp ddns.key /var/named/etc/
chown root:named /var/named/etc/ddns.key
cp ddns.key /etc/
```

3 Modify DNS configuration

The DNS configuration is very simple. In the zone sections of `/var/named/etc/named.conf`, add the following line:

```
allow-update { key DDNS_UPDATE; };
```

This should go in your `nw3.sqrawler.com` zone and in the reverse zones you have set up on your network.

You will also need to include the key file in your config by adding the line

```
include "etc/ddns.key"
```

to your `named.conf` file, right above the zone sections.

4 Modify DHCP configuration

Add the following global options to your `/etc/dhcpd.conf` file:

```
ddns-updates on;
ddns-update-style interim;
include "/etc/ddns.key";
```

Finally, create blocks for the zones for which you will perform dynamic updates like the following:

```
zone nw3.sqrawler.com. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}

zone 5.16.172.in-addr.arpa. {
    primary 127.0.0.1;
    key DDNS_UPDATE;
}
```

Note that, since our DHCP server and our DNS server run on the same machine, we can use the loopback addresses for our primary DNS server. If this were not the case, we would need to supply the DNS server's address.

5 Test your setup

Once your configuration is complete, restart your DNS and DHCP servers. To start your alternative DHCP server, use the command

```
/usr/local/sbin/dhcpd
```

Then obtain a DHCP lease for one or more of your client machines, then use `dig` to lookup the corresponding A record. The name in the A record should match the hostname of the client machine.