DNS Concepts for sipXcom

This particular document isn't meant to be a tutorial on how to create DNS zones or add records to them. It is meant to help the reader understand the issues surrounding DNS configurations for sipXcom and how to leverage DNS in different scenarios. DNS configuration seems to be the most difficult concept for users new to the system to grasp.

DNS setup can be tricky for a system setup where a user needs to be able to roam from inside the office to outside and then maybe even be on a VPN connection. To make this all work seamlessly, the PBX must be able to be referenced the same way no matter where the end-user is located.

It is very important that network administrators understand DNS. Wikipedia has a great reference to help get more comfortable with DNS. Even if you think you know everything about it, a refresher is always good.

There is a standalone xecsuser:document on setting up Microsoft DNS Services for use with sipXcom

DNS Records

It is important to understand the difference between A records and SRV records so let's get that out of the way first.

An A record is a typical host name pointer to an IP address. So, **sipXcom.example.com** might be an A record that points to **127.1.1.43**. In this example, sipXcom is the host name, **example.com** is the domain name and **sipXcom.example.com** is referred to as the fully qualified domain name (fqdn). We use A records because it's easier to remember computer names than a series of numbers (for most people anyway). Most Asterisk configurations the author has seen run with an A record setup for the PBX.

SRV records (service records) are used by newer Internet protocols to locate a type of service that might be available for a domain. MX records (mail exchange) work similarly to SRV records but are used specifically for mail. SRV records can be used for any number of services. A SIP service record will look like **_sip._udp.example.com**. In this example a SIP device can query DNS for domain **example.com** and find an IP address or host name for SIP services running on the UDP protocol. Similarly, **_sip._tcp.example.com** would return an IP address or host name for SIP services running on the TCP protocol. So, a phone registering to the **example.com** domain would lookup **_sip._udp.example.com** and have the host name **sipXcom.example.com** returned. Why not just cut to the chase and use the A record? You'll see why in a bit.

sipXcom PBX configurations are typically setup with SRV records (strongly recommended). SRV records are the method utilized to locate the sipXcom PBX in a clustered environment.

Host Records (A Records)

As described above, A Records are simply name pointers to IP addresses. While A record naming is not typically recommended for sipXcom installations, A records can be utilized to locate the pbx. On a clean sipXcom system (before you add any users), under **system administration**, go into the **System** menu and select the **Domain** menu item. Change the **domain name** to the fully qualified domain name that you would like to use, click **Apply** and then reboot.

Make sure your new A record is in your internal DNS server pointing to the internal IP address of the sipXcom pbx.

Internally DNS zone file for the **example.com** domain would have an A record setup pointing to the IP of the PBX:

```
sipXcom         A          192.168.10.2
```

Have your ISP (or if you can do it yourself with a self-managed DNS hosting provider) add an A record for your pbx pointing at the external IP address of your firewall/SBC.

Externally, DNS would have an A record setup pointing to the external IP which is mapped to the internal IP of the PBX:

```
sipXcom         A          127.1.1.43
```

Testing A Record Configuration

Verify that it all works properly by pinging the PBX by name from both inside the firewall and from outside the firewall using the same DNS domain name. For example:

```
ping sipXcom.example.com
```

Should return something like:

```
Pinging sipXcom.example.com [192.168.10.2] with 32 bytes of data:

Reply from 192.168.10.2: bytes=32 time=1ms TTL=64

Reply from 192.168.10.2: bytes=32 time=1ms TTL=64

Reply from 192.168.10.2: bytes=32 time=1ms TTL=64

Reply from 192.168.10.2: bytes=32 time=1ms TTL=64


Ping statistics for 192.168.10.2:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

From the Internet the results should be similar but the IP address will be the external IP address of the firewall/SBC (assuming of course that you are allowing ICMP traffic).

SRV Records

So, why are SRV records typically used by sipXcom? Well, they allow us to be contacted with the same address used to deliver e-mail and to provide server redundancy and load balancing.

Take for example the top salesperson for Example Company, Joe User. Joe's phone extension is 512 and he has an email address of joe.user@example.com. SRV records allow Joe User to be contacted at [] and if an alias is put on his user account he can also be called by phone at []. If **example.com** were using A records for their sipXcom installation, Joe's phone number and alias would have be [] and [] respectively (unless of course the addresses are transformed by the SBC, but then we get into the whole problem of inside the network and outside the network).

SRV records also allow the administrator to have redundant and load balanced SBC computers (sipXcom also uses this method for locating PBX's in a clustered/high availability configuration). SRV records hold the following information:

Service: the symbolic name of the desired service.

- **Protocol**: this is usually either TCP or UDP.
- **Domain name**: the domain for which this record is valid.
- **TTL**: standard DNS time to live field.
- **Class**: standard DNS class field (this is always IN).
- **Priority**: the priority of the target host.
- **Weight**: A relative weight for records with the same priority.
- **Port**: the TCP or UDP port on which the service is to be found.
- **Target**: the hostname of the machine providing the service.

An example SRV record might look like this in a DNS zone file:

```
_sip._udp.example.com 86400 IN SRV 0 5 5060 sipXcom.example.com.
```

This SRV record points to a server named **sipXcom.example.com** listening on **UDP** port **5060** for **SIP** protocol connections. The priority given here is **0**, and the weight is **5**.

The priority field is similar to an MX record's priority value. Clients always use the SRV record with the lowest priority value first, and only fall back to other records if the connection with this record's host fails. Thus a service may have a designated "fallback" server, which will only be used if the primary server fails. Only another SRV record, with a priority field value higher than the primary server's record, is needed.

If a service has multiple SRV records with the same priority value, clients use the weight field to determine which host to use. The weight value is relevant only in relation to other weight values for the service, and only among records with the same priority value.

In the following example, both the priority and weight fields are used to provide a combination of load balancing and backup service.

```
_sip._udp.example.com 86400 IN SRV 10 60 5060 sipx1.example.com.

_sip._udp.example.com 86400 IN SRV 10 20 5060 sipx2.example.com.

_sip._udp.example.com 86400 IN SRV 10 20 5060 sipx3.example.com.

_sip._udp.example.com 86400 IN SRV 20 0 5060 sipx4.example.com.
```

The first three records share a priority **10**, so the weight field's value will be used by clients to determine which host to contact. The sum of all three values is **100**, so **sipx1.example.com** will be used **60%** of the time. The other two hosts, **sipx2** and **sipx3**, will be used for **20%** of requests each. If **sipx1.example.com** is unavailable, these two remaining machines will share the load equally, since they will each be selected 50% of the time.

If all three hosts with priority **10** are unavailable, the record with the next highest priority value will be chosen, which is **sipx4.example.com**. This might be a machine in another physical location, presumably not vulnerable to anything that would cause the first three hosts to become unavailable.

Let's look at what these records look like on the internal DNS server and then on the external DNS server. We'll just concentrate on a single SBC and not a redundant configuration.

So, internally DNS zone file for the example.com domain would have an A record setup pointing to the IP of the PBX:

```
sipXcom         A         192.168.10.2
```

And then the SRV records would be setup as follows:

```
_sip._udp.example.com   86400     IN        SRV        10 100 5060        sipXcom.example.com

_sip._tcp.example.com   86400     IN        SRV        10          100 5060        sipXcom.example.com
```

Externally, DNS would have an A record setup pointing to the external IP of the PBX (which would have a 1:1 NAT mapping to the internal IP address of the PBX and be firewalled appropriately):

```
 sipXcom        A         127.1.1.43
```

And then the SRV records would be setup exactly as they were internally (note, if you are using OpenSBC it only needs the UDP record):

```
_sip._udp.example.com   86400     IN        SRV        10 100 5060        sipXcom.example.com

_sip._tcp.example.com   86400     IN        SRV        10          100 5060        sipXcom.example.com
```

Testing SRV Configuration

The dig command in Linux lets us test DNS.

At the command line enter:

```
dig --t SRV _sip._udp.example.com
```

The following will be displayed:

```
; <<>> DiG 9.3.4-P1 <<>> -t SRV _sip._udp.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48387
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;_sip._udp.example.com. IN      SRV

;; ANSWER SECTION:
_sip._udp.example.com. 0 IN SRV      0 0 5060 sipXcom.example.com.

;; Query time: 3 msec
;; SERVER: 172.16.1.254#53(172.16.1.254)
;; WHEN: Sat Jan 3 15:46:12 2009
;; MSG SIZE rcvd: 81
```

From the **ANSWER: 1** in the returned information and the **sipXcom.example.com** in the **ANSWER SECTION:** we know that the SRV record is working properly.

From a Windows workstation the **nslookup** command can be used:

At the command prompt enter:

```
nslookup
```

```
Default Server: UnKnown
Address: 172.16.1.254
> set querytype=SRV
> _sip._udp.example.com
Server: UnKnown
Address: 172.16.1.254

_sip._udp.example.com SRV service location:
          priority         = 0
          weight           = 0
          port             = 5060
          svr hostname     = sipXcom.example.com
```

Required DNS Records for a Single Server Environment

The following are the required records for a single server sipXcom system.

A DNS Domain that is equivalent to the SIP domain

A-Record (host record) for the server

SRV records for the SIP communications (port 5060 tcp & udp).

SRV record for the resource record (port 5070 tcp)

SRV record for XMPP client connections (port 5222 tcp)

SRV record for XMPP server connections (port 5269 tcp)

SRV record for XMPP client connections to XMPP conference (port 5222)

SRV record for XMPP servers connections to XMPP conference (port 5222)

Example Single Server DNS Zone file

A typical DNS zone file looks as follows:

```
; WARNING: Zone file configuration is a sipX automatically generated file.
;          Contents may be overwritten unless you set the named.conf DNS_MODE.
;
```

```
$TTL 1800
@       IN     SOA    ns1.ezuce.com. root.ezuce.com. (
                       2010090201 ; serial#
                       1800            ; refresh, seconds
                       1800            ; retry, seconds
                       1800            ; expire, seconds
                       1800 )          ; minimum TTL, seconds

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DNS Servers for 'ezuce.com'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; NS record for ezuce.com
;    server: openuc.ezuce.com
;
ezuce.com.            IN     NS     openuc.ezuce.com.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Call Routing for SIP domain 'ezuce.com'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; NAPTR record for SIP TCP ezuce.com
;     priority: 2  weight: 0
;     protocol: "SIP+D2T"  regex: ""  uri: _sip._tcp.ezuce.com
;
ezuce.com.           IN      NAPTR   2 0 "s" "SIP+D2T" "" _sip._tcp.ezuce.com.

; NAPTR record for SIP UDP ezuce.com
;     priority: 2  weight: 0
;     protocol: "SIP+D2U"  regex: ""  uri: _sip._udp.ezuce.com
;
ezuce.com.           IN      NAPTR   2 0 "s" "SIP+D2U" "" _sip._udp.ezuce.com.

; SRV record for domain SIP TCP ezuce.com
;     priority: 1  weight: 0  port: 5060  server: openuc.ezuce.com
;
_sip._tcp.ezuce.com. IN      SRV     1 0 5060 openuc.ezuce.com.

; SRV record for domain SIP UDP ezuce.com
;     priority: 1  weight: 0  port: 5060  server: openuc.ezuce.com
;
_sip._udp.ezuce.com. IN      SRV     1 0 5060 openuc.ezuce.com.

; SRV record for service SIP TCP rr.openuc.ezuce.com
;     priority: 1  weight: 0  port: 5070  server: openuc.ezuce.com
;
_sip._tcp.rr.openuc.ezuce.com. IN      SRV     1   0 5070 openuc.ezuce.com.

; SRV record for XMPP SERVER TCP ezuce.com
;     priority: 1  weight: 0  port: 5269  server: openuc.ezuce.com
;
_xmpp-server._tcp.ezuce.com. IN      SRV     1  0 5269 openuc.ezuce.com.

; SRV record for XMPP CLIENT TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;
_xmpp-client._tcp.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

; SRV record for XMPP SERVER CHAT ROOM TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;     NOTE: the XMPP client port is used here as this is the port used
;           by openfire to service multi-user chat requests.
;
_xmpp-server._tcp.conference.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

; SRV record for XMPP CLIENT CHAT ROOM TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;
_xmpp-client._tcp.conference.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
; IP Addresses
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; A record for openuc.ezuce.com
;
openuc.ezuce.com.       IN      A       192.168.5.2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Required DNS Records for a Multiple Server Environment

The following are the required records for a multiple server sipXcom system.

A DNS Domain that is equivalent to the SIP domain

A-Record (host record) for each of the servers

SRV records for the SIP communications (port 5060 tcp & udp) with priority and weight for each of the servers.

SRV record for the resource record (port 5070 tcp) for each of the servers. This record is used for replication services to locate other servers for replication.

SRV record for XMPP client connections (port 5222 tcp) for the IM server

SRV record for XMPP server connections (port 5269 tcp) for the IM server

SRV record for XMPP client connections to XMPP conference (port 5222) for the IM server

SRV record for XMPP servers connections to XMPP conference (port 5222) for the IM server

Example DNS Zone file

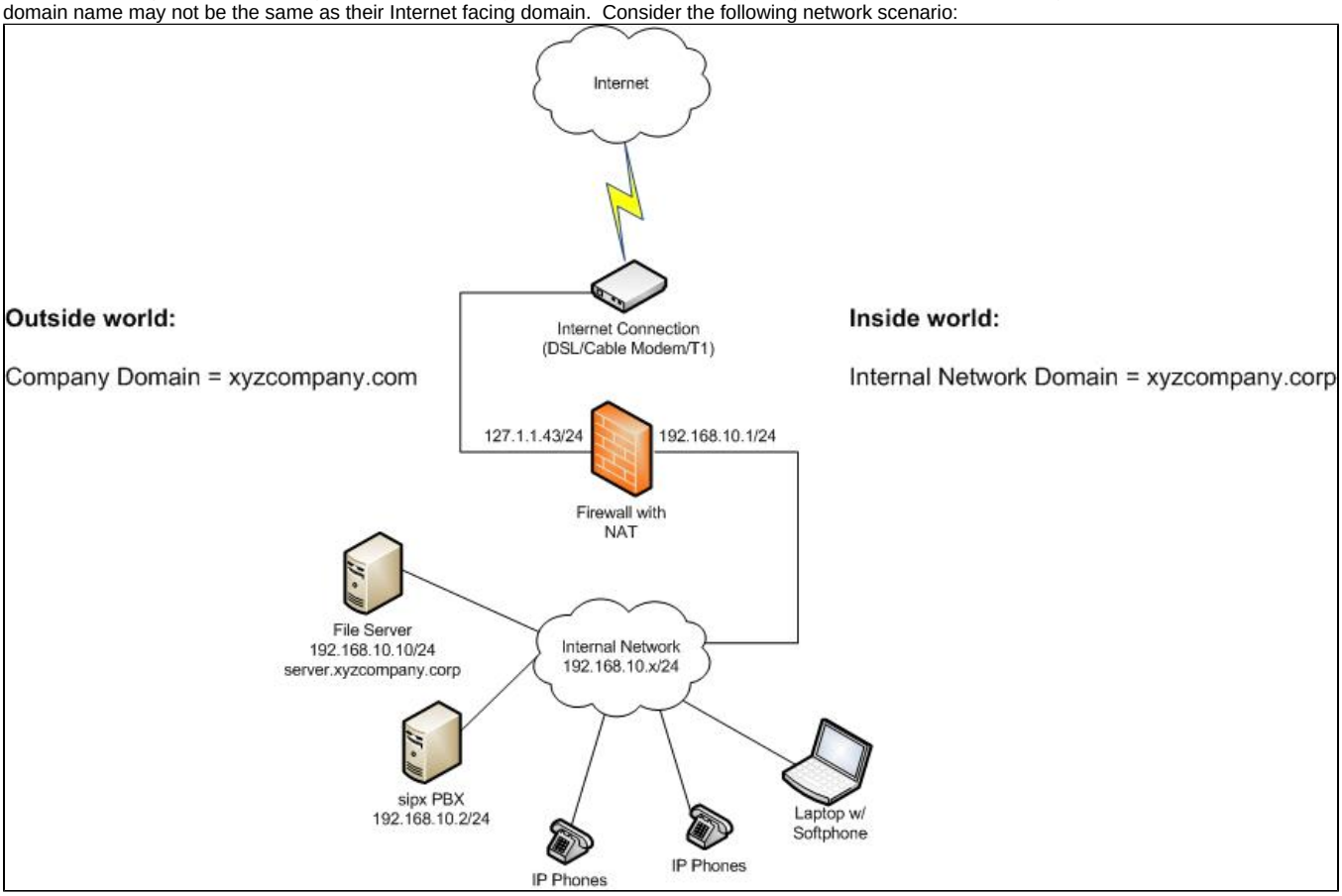
A typical DNS zone file for a multiple server configuration looks as follows:

```
; WARNING: Zone file configuration is a sipX automatically generated file.
;          Contents may be overwritten unless you set the named.conf DNS_MODE.
;
$TTL 1800
@       IN     SOA    ns1.ezuce.com. root.ezuce.com. (
                       2010090201 ; serial#
                       1800            ; refresh, seconds
                       1800            ; retry, seconds
                       1800            ; expire, seconds
                       1800 )          ; minimum TTL, seconds

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DNS Servers for 'ezuce.com'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; NS record for ezuce.com
;    server: openuc.ezuce.com
;
ezuce.com.            IN     NS     openuc.ezuce.com.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Call Routing for SIP domain 'ezuce.com'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; NAPTR record for SIP TCP ezuce.com
;     priority: 2  weight: 0
;     protocol: "SIP+D2T"  regex: ""  uri: _sip._tcp.ezuce.com
;
ezuce.com.           IN      NAPTR   2 0 "s" "SIP+D2T" "" _sip._tcp.ezuce.com.

; NAPTR record for SIP UDP ezuce.com
;     priority: 2  weight: 0
;     protocol: "SIP+D2U"  regex: ""  uri: _sip._udp.ezuce.com
;
ezuce.com.           IN      NAPTR   2 0 "s" "SIP+D2U" "" _sip._udp.ezuce.com.
```

```
; SRV record for domain SIP TCP ezuce.com
;     priority: 1  weight: 0  port: 5060  server: openuc.ezuce.com
;
_sip._tcp.ezuce.com. IN      SRV     1 0 5060 openuc.ezuce.com.
_sip._tcp.ezuce.com. IN      SRV     1 0 5060 openuc2.ezuce.com.

; SRV record for domain SIP UDP ezuce.com
;     priority: 1  weight: 0  port: 5060  server: openuc.ezuce.com
;
_sip._udp.ezuce.com. IN      SRV     1 0 5060 openuc.ezuce.com.
_sip._udp.ezuce.com. IN      SRV     1 0 5060 openuc2.ezuce.com.

; SRV record for service SIP TCP rr.openuc.ezuce.com
;     priority: 1  weight: 0  port: 5070  server: openuc.ezuce.com
;
_sip._tcp.rr.openuc.ezuce.com. IN      SRV     1   0 5070 openuc.ezuce.com.
_sip._tcp.rr.openuc.ezuce.com. IN      SRV     2 100 5070 openuc2.ezuce.com.

_sip._tcp.rr.openuc2.ezuce.com. IN      SRV     1   0 5070 openuc2.ezuce.com.
_sip._tcp.rr.openuc2.ezuce.com. IN      SRV     2 100 5070 openuc.ezuce.com.


; SRV record for XMPP SERVER TCP ezuce.com
;     priority: 1  weight: 0  port: 5269  server: openuc.ezuce.com
;
_xmpp-server._tcp.ezuce.com. IN      SRV     1  0 5269 openuc.ezuce.com.

; SRV record for XMPP CLIENT TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;
_xmpp-client._tcp.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

; SRV record for XMPP SERVER CHAT ROOM TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;     NOTE: the XMPP client port is used here as this is the port used
;           by openfire to service multi-user chat requests.
;
_xmpp-server._tcp.conference.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

; SRV record for XMPP CLIENT CHAT ROOM TCP ezuce.com
;     priority: 1  weight: 0  port: 5222  server: openuc.ezuce.com
;
_xmpp-client._tcp.conference.ezuce.com. IN      SRV     1  0 5222 openuc.ezuce.com.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; IP Addresses
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; A record for openuc.ezuce.com
;
openuc.ezuce.com.       IN      A       192.168.5.2
openuc2.ezuce.com.      IN      A       192.168.5.3

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Scenario 1 – sipXcom PBX on the Data Network

Ok, so most companies aren't hiding under rocks and they are already using DNS internally on their networks. Like many, the internal network DNS domain name may not be the same as their Internet facing domain. Consider the following network scenario:



With our fictitious company, the internal computer network has an existing DNS domain of example.corp which is being used by Active Directory. DHCP and DNS are being handled by the file server at 192.168.10.10. The simplest approach to integrating sipXcom into this network is to leave DNS and DHCP on the file server and add the required SRV records and DHCP options to it.

There are a few problems with this scenario:

1. The operation of your phone system is dependent on DNS working. If DNS or your File Server has problems, your phone system is also down.
2. Mobile users that need to roam to different networks outside of the organization will not be able to resolve the example.corp domain from the internet.
3. Users outside of your organization can not call your system via a SIP URL (sip:ext@example.com).

Problems 2 and 3 can be worked around, but problem 1 is the bigger issue. Making sure that you have a robust DNS configuration is important to your computer network and your phone network.

> ⚠ **Important Tip:**
>
> In this scenario DNS and DHCP should be configured properly BEFORE installing sipXcom.

Configure DNS

There are four DNS records that need to be configured for sipXcom to function properly. They are:

- A Record for sipXcom PBX host name
- SRV Record for SIP UDP Signaling Traffic
- SRV Record for SIP TCP Signaling Traffic
- SRV Record for sipXcom PBX Resource locating (Resource Record)

Add a new host record for the PBX (in the above example it would be something like sipXcom.example.corp) pointing to the IP address of the PBX.

Add SRV records for **_sip._udp.example.corp**, **_sip._tcp.example.corp** and **_sip._tcp.rr.sipXcom.example.corp** all pointing to **sipXcom.example.corp** (if you are doing this in a Microsoft Windows environment, see document mentioned in beginning of this whitepaper).

In a bind (Linux DNS) configuration file these records would look something like this:

```
_sip._tcp.example.corp.         IN SRV 1 0 5060 sipXcom.example.corp.
_sip._udp.example.corp.         IN SRV 1 0 5060 sipXcom.example.corp.
_sip._tcp.rr.sipXcom.example.corp. IN SRV 1 0 5070 sipXcom.example.corp.
sipXcom.example.corp.              IN A           192.168.10.2
```

Configure DHCP

DHCP is used by phones and PC's alike in this scenario to get IP addresses as well as other information needed about the network to operate properly.

In addition to an IP address, to operate properly phones need a Default Gateway (DHCP option 3), DNS Domain Name (DHCP option 15), DNS Server IP address (DHCP option 6) and a TFTP server address (DHCP option 66). If you are using a different provisioning method you're on your own here.

A typical DHCP configuration file from a Linux system would look like:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
        range 192.168.10.20 172.168.10.254;  #IP Range
        default-lease-time 21600;
        max-lease-time 43200;
        option routers 192.168.10.1;  # Default gateway
        option subnet-mask 255.255.255.0;  # Subnet mask
        option domain-name "example.corp";  #DNS Domain Name
        option domain-name-servers 192.168.10.10;  #DNS Server IP
        option time-offset 18000;  # Eastern Standard Time
        option tftp-server-name "sipXcom.example.corp";  #phone provisioning
        option ntp-servers 192.168.10.10;  #get time from file server
        }
```

Testing

It is important to test your configuration and verify that it is operating as it should before you install your sipXcom system. Refer to the testing section above.

Boot a computer and make sure it receives an IP address just as a phone would. Check that it is receiving the proper DNS domain name (on a Windows machine you can use **ipconfig /all** from the command prompt to verify this).

Make sure the computer can Ping the PBX by its host name and us **NSLOOKUP** to verify that the SRV records are working as they should.

Once DNS and DHCP are working properly you are ready to install your sipXcom system.

Remote Users

Assuming now that your sipXcom system is up and running correctly, the next challenge with DNS is configuring it so that remote users can connect to your PBX. If those users are connecting via a VPN tunnel or wide area network, simply configure DNS on the far end to have the same DNS records and DHCP records we setup above.

If users are connecting from the Internet they are never going to be able to resolve **example.corp** because it isn't a valid DNS domain name. Your organization more than likely has something like **example.com** that is registered with an organization like Network Solutions and hosted either there or at another DNS hosting provider. The following host record and SRV records must be configured at your DNS hosting provider:

```
_sip._tcp.example.com.        IN SRV 1 0 5060 sipXcom.example.com.
_sip._udp.example.com.        IN SRV 1 0 5060 sipXcom.example.com.
sipXcom.example.com.             IN A           127.1.1.43  # Change to outside IP of your FW
```

Configure your firewall to allow and NAT ports **5060 udp**, **5060 tcp** and ports **30000 - 31000 udp** from **127.1.1.43** (again, this would be changed to **YOUR external IP address**) to **192.168.10.2** (this would be changed to the **internal IP address of YOUR PBX**).

Just as you tested your internal DNS, make sure you test the external DNS from outside your network.

One last step needs to be completed. sipXcom allows for domain name aliases (**System Menu -> Domains**). Add a domain alias for **example.com**. An alias allows the sipXcom system to respond to requests made to domains other than the domain it is setup with. It thus should be possible to setup your mobile users differently than your fixed position hard phone users and do a translation of **example.com** to **example.corp**.

Dynamic DNS

Wondering what about the case where the external IP address may change like with a Cable Modem or DSL connection? Usually the only way you will be able to deal with SRV records is by owning your own domain. Drop $20 a year with a hosting provider like GoDaddy.com or similar (just make sure they let you have SRV records) and get yourself a domain name.

Once you have an domain name, get setup with DynDNS or one of the other dynamic DNS providers (I use DynDNS because it works with Vyatta firewall). If you don't have a firewall that does dynamic DNS updates, you can usually run software on an internal machine that helps the dynamic DNS provider figure out your external IP address.

The dynamic DNS provider will let you determine your own host name and tag it to one of their domain names. For instance, **examplecorp.dyndns.net** might be a host name you could specify. We can then point to this dynamic DNS name from our own domain.

For pointing a host name at another host name we'll use a **CNAME** record (canonical name). So, externally the DNS would have a CNAME record setup pointing to the dynamic DNS name:

```
sipXcom               CNAME examplecorp.dyndns.net
```

And then the SRV records would be setup also pointing to the dynamic DNS name as follows:

```
_sip._udp.example.com 86400 IN SRV 10 100 5060 examplecorp.dyndns.net
_sip._tcp.example.com 86400 IN SRV 10 100 5060 examplecorp.dyndns.net
```

Notes on DNS & SRV Records with Polycom Phones

A flaw in Polycom's SRV record use has been identified. Polycom phones don't honor the DNS SRV weight, but they do honor the priority. What this means is that they don't resort the list of records when they get it from DNS. Most DNS servers by default produce the SRV records in cyclical order. This means that for a 3 server cluster there are 3 combinations that the records will show up.

A suggested work-around for this problem is to use the bind (9.6 & later) command rrset-order.

```
rrset-order { type SRV order random; };
```

Documentation for rrset-order here: http://www.zytrax.com/books/dns/ch7/queries.html

> *rrset-order*
>
> *rrset-order { order_spec ; [order_spec ; ...]*
> *rrset-order defines the order in which multiple records of the same type are returned. This works for any record type in which the records are similar not just A or AAAA RRs and covers results in the ANSWER SECTION and the ADDITIONAL SECTION. The default is cyclic (round-robin).*
>
> *The full specification of rrset-order is shown below. An 'order_spec' is defined as:*
>
> *class class_name][type type_name][name "domain_name"] order ordering;*
> *Where 'class_name' is the record class, for example, IN (default is 'any'), type is the Resource Record type (if none specified defaults to 'any'), domain_name limits the statement to a specific domain suffix and defaults to root (all domains), order is a key word and ordering may take one of the following values:*
>
> *fixed - records are returned in the order they are defined in the zone file*
> *random - records are returned in a random order*
> *cyclic - records are returned in a round-robin fashion*
> *Note: For reasons best known to the ISC (BIND's author) the fixed value is now (BIND 9.6+) only available if the configure option --with-fixed-rrset is used in the build. Neither BSD nor Debian standard packages use this option. This is likely to be true for Fedora and other RPMs but has not been verified (use named -V to check). For practical purposes only cyclic and random are the available choices.*
>
> *Examples*
>
> *Defines that all equal records for all domains will be returned in random order.*
>
> ```
> rrset-order {order random;};
> ```
>
> *Defines that all equal MX records for example.com will be returned in random order all others in cyclic order.*
>
> ```
> rrset-order {type MX name "example.com" order random; order cyclic};
> ```
>
> *This statement may be used in a view or a global options clause.*

Advanced DNS configuration

In addition to the above scenario an advanced location/subnet based DNS lookup system can be deployed using BIND Views. More information can be found here: Location based DNS views for sipXcom using BIND