Subject: Re: domain names / virtual machines
Posted by illc0mm on Fri, 29 Feb 2008 15:38:04 GMT
View Forum Message <> Reply to Message

If you want them in separate environments, I'd suggest something like the following. This is an over simplified setup but it should get the idea across.

Host Node:
IP 1.1.1.1 (external IP)
IP 192.168.0.1 (internal IP)

Firewall Rules:
#Port Forward TCP 25,21,80,110,143,443,whatever else from 1.1.1.1 to 192.168.0.101 (VE 101)
iptables -t nat -A PREROUTING -p tcp -d 1.1.1.1 --dport 25 -i eth0 -j DNAT --to-destination 192.168.0.101:25
iptables -t nat -A PREROUTING -p tcp -d 1.1.1.1 --dport 80 -i eth0 -j DNAT --to-destination 192.168.0.101:80
# etc...

# SNAT statement to allow VE nodes Internet access through the Host Node
iptables -A POSTROUTING -s 192.168.0.0/255.255.255.0 -o eth0 -j SNAT --to-source 1.1.1.1

VE Node 101 (your "proxy" node):
IP 192.168.0.101

apps:
 Perdition POP/IMAP proxy
 sshproxy-project
 pound proxy (http)
 postfix (relay)

VE Node 102 ("servera"):
IP 192.168.0.102

apps:
 apache
 ssh
 postfix (mta)
 courier-imap (pop/imap)
 whatever else...

VE Node 103 ("serverb"):
IP 192.168.0.103

apps:
 apache
 ssh
 postfix (mta)

courier-imap (pop/imap)
whatever else...

The only thing that is truly "name" aware will be your http connections since HTTP v1.1 provides a name based virtual facility. Nothing else, yet, can identify a server by name during the initial connection so you have to be more creative for those. IP packets don't contain name information so you have to resort to an application aware proxy. Usually you can do some sort of lookup on the user name or something to that effect to find out where to proxy the connection.

More about what runs on VE 101 (proxy node):

POP3/IMAP:
Perdition Mail Retrieval Proxy
http://www.vergenet.net/linux/perdition/

This will handle SSL and NON SSL connections and keys of the user account to find which server the user is going to. User accounts usually become the mailbox's full email address "user@domain.com". Perdition can then use several lookup methods (sql, hash, nis, regex, etc) to then proxy the connection to the correct server.

SMTP:
Postfix MTA
http://postfix.org

Postfix can be setup to accept messages for multiple domains, then relay those messages to the appropraite VE. Like perdition it can get that information from sql, hash, nis, regex, you name it. Makes it very flexiable. It's one of the more secure MTAs out there and it's pretty easy to configure.

HTTP/HTTPS:
http://www.apsis.ch/pound/

This can proxy both http and https (the back end communication will be http since it has to be decrypted in pound)

DNS:
Since everything is going to resolve to the same IP, this is pretty simple. I'd highly recommend maradns for this task, since it can use templates and make provisioning a snap, and it has an awesome security record. It has a low resource overhead and is easy to configure.

http://www.maradns.org/

SSH:
For this, you could use sshproxy-project

http://sshproxy-project.org/about/

You'd need to change the default port it runs on (port 2242 is the default) to port 22. At that point

I'd also change the port my Host Node runs on to something else (like 22222 or whatever) so it will not conflict. That way, your users will have a seamless experience. sshproxy-project wasn't really designed for this purpose, but it should work fine.

Jabber:
Not sure on this one, never done it but I'm sure the people on Jabber support forum might have a clue on that one.

Sybase:
I would assume this would be internal and not exposed to the Internet. I'm not sure what sybase web tools exist out there (like phpmyadmin) but I would imagine that's what you'd want to do. Otherwise, this is where a VPN or port tunneling over SSH would come into play.

SVN:
You'll have to research that on your own, I imagine it's possible.

As far as what you run on server a and b, it doesn't matter since the proxy node just needs to hand it off to something.

Hope that helps.

-illc0mm

---