Admins - Anomalie #1264

DNS interne pour l'environnement de test

30/04/2013 11:44 - Quentin CHERGUI

Statut:	Fermé	**Début:**	30/04/2013
Priorité:	Normale	**Echéance:**	
Assigné à:	Quentin CHERGUI	**% réalisé:**	100%
Catégorie:	Task	**Temps estimé:**	0.00 heure
Version cible:	Juin 2013	**Temps passé:**	16.50 heures
Difficulté:	3 Moyen		

Description

Lors de l'écriture des tests dans April-Ci, le problème de la résolutions des noms externes (comme april.org) dans l'environnement de test est apparu (par exemple, les scénarios Cucumber ont besoin d'avoir un FQDN indiqué).

Pour que ces noms externent pointent vers l'environnement de test, il faut un résolveur DNS interne qui fasse correspondre chaque nom externe à son équivalent dans l'environnement de test (en .novalocal).

- each instance runs a spoof_dns instance that overrides some DNS entries

```
fqdn2fqdn = {
    'foo.com': 'foo.me',
    'bar.com': 'bar.me',
}
```

- the spoof_dns forwards the novalocal zone to the OpenStack provided resolver found in the resolv.conf of the instance at boot time
- the dhclient is configured to prepend unbound to /etc/resolv.conf so that it is always used as a resolveer

```
/etc/dhcp/dhclient.conf:prepend domain-name-servers 127.0.0.1;
```

- a script parses the manifest files and creates fqdn2fqdn entries to *.novalocal for each node name, using the spoof_dns syntax
- when in a test environment (determined by the is_april_ci varilable found in the params file used to store password) the april_ci_dns class is activated to run spoof_dns on the instance

Révisions associées

Révision 88efdddb - 02/05/2013 18:25 - Loïc Dachary

Move april_params.pp up in the list of inclusion in site.pp so that it is first, so that the variables it contains can be used to decide for the inclusion of other manifest files. Add the $is_april_ci boolean to april_params.pp to distinguish between the production environment and the continuous integration environment. An example of action that only makes sense in the continuous integration context is when the DNS is configured to redirect all top level domain names so that they resolve into private IPs within the continuous environment. It would not make sense to do the same in the production environment.
refs #1264

Révision e34a8262 - 03/05/2013 16:24 - Loïc Dachary

implement a DNS spoofing daemon with associated tests. The map translating a FQDN into another is to be defined in the spoof_map.py file and as follows:
fqdn2fqdn = {
'foo.com': 'foo.me',
'bar.com': 'bar.me',
}
refs #1264

Révision b162b3a3 - 03/05/2013 18:13 - Loïc Dachary

use addBoth instead of addCallback to raise an error if the domain resolves refs #1264

Révision b7703280 - 05/06/2013 15:33 - Quentin CHERGUI

Add a lying DNS resolver machine on april-ci infrastructure, based on BIND9 with RPZ-policy. Aim is redirect production domains to test environnement. In this revision, /etc/resolv.conf is not changed on test VMs. refs #1264

Révision faacb31c - 10/06/2013 18:27 - Quentin CHERGUI

april_ci_dns : add forwarder to Bind9 configuration, based on original /etc/resolv.conf file refs #1264

Révision 7c41fae2 - 10/06/2013 19:04 - Quentin CHERGUI

Force write /etc/resolv.conf when dhclient does not run properly
For unknow reasons, dhclient eth0 doesn't works for re-run dhcp.
In this case, we need to rewrite manually /etc/resolv.conf
/!\ DIRTY HACK refs #1264

Révision 754e704d - 12/06/2013 16:38 - Quentin CHERGUI

Change dhclient options for have only-one DNS resolver, defined by Puppet, and restart dhclient refs #1264

Révision e5e22ed5 - 12/06/2013 17:27 - Quentin CHERGUI

Deplace puppet agent execution on puppetmaster : must be runned after deployment of dns-ci refs #1264

Révision 81b73bb3 - 12/06/2013 18:21 - Quentin CHERGUI

Clean unused files on april_ci_dns module
A lot of old files was present on this modules, used during tests for find a solution to the DNS problem on test environnement. You can see this differents test on previous commits, or on attached Redmine ticket (#1264 on agir.april.org).
refs #1264

Révision 40bf91f5 - 17/06/2013 12:33 - Quentin CHERGUI

Keep the dns-ci instance alive when tests are finished refs #1264

Révision cf99a9f3 - 18/06/2013 13:27 - Quentin CHERGUI

Remove deletion and reconstrucion of dns-ci un setupdns() function, and add puppet agent call for update it refs #1264

Révision 1c626c7c - 18/06/2013 14:04 - Quentin CHERGUI

Remove deletion and reconstrucion of dns-ci un setupdns() function, and add puppet agent call for update it refs #1264

Révision 56e2c770 - 19/06/2013 17:18 - Quentin CHERGUI

Correct the tests for generate_zpz_zone script in module april_ci_dns refs #1264

Historique

#1 - 02/05/2013 11:09 - Loïc Dachary

nsswitch.conf n'est pas utilisable pour ça. L'option db pourrait servir pour les hosts mais c'est pratiquement jamais utilisé donc les chances pour que ça marche sont faibles.

#2 - 02/05/2013 12:24 - Loïc Dachary

Faire un script a base de http://notmysock.org/blog/hacks/a-twisted-dns-story.html

#3 - 02/05/2013 17:13 - Loïc Dachary

```
# twistd -y dns.py
import socket
from twisted.internet.protocol import Factory, Protocol
from twisted.internet import reactor
from twisted.names import dns
from twisted.names import client, server
public2private = {
    'foo.com': 'foo.me',
    'bar.com': 'bar.me',
}
class DNSServerFactory(server.DNSServerFactory):
    def handleQuery(self, message, protocol, address):
        query = message.queries[0]
        if query.name.name in public2private:
            remapped = query.name.name
            query.name.name = public2private[query.name.name]
```

```
        else:
            remapped = False

        return self.resolver.query(query).addCallback(
            self.gotResolverResponse, remapped, protocol, message, address
            ).addErrback(
            self.gotResolverError, remapped, protocol, message, address
            )
    def gotResolverResponse(self, (ans, auth, add), remapped, protocol, message, address):
        if remapped:
            message.queries[0].name.name = remapped
            ans[0].name.name = remapped
        args = (self, (ans, auth, add), protocol, message, address)
        return server.DNSServerFactory.gotResolverResponse(*args)
    def gotResolverError(self, failure, remapped, protocol, message, address):
        if remapped:
            #type(message.queries[0].name.name)
            message.queries[0].name.name = remapped
        args = (self, failure, protocol, message, address)
        return server.DNSServerFactory.gotResolverError(*args)
verbosity = 0
resolver = client.Resolver(servers=[('8.8.8.8', 53)])
factory = DNSServerFactory(clients=[resolver], verbose=verbosity)
protocol = dns.DNSDatagramProtocol(factory)
factory.noisy = protocol.noisy = verbosity
reactor.listenUDP(53, protocol)
reactor.listenTCP(53, factory)
reactor.run()
```

#4 - 02/05/2013 19:07 - Loïc Dachary

Ettercap is a comprehensive suite for man in the middle attacks est peut être exactement ce dont on a besoin. A tout les point de vues on a les même problèmes, juste pour des objectifs différents. Le seul problème c'est que ça ne permet pas non plus de remapper un FQDN dans un autre FQDN, comme l'explique le fichier de configuration du module dns_spoof

#5 - 02/05/2013 19:40 - Loïc Dachary

@quentin : "#FIXME : il manque un gotResolverError pour éviter les "Question section mismatch" quand le nom n'existe pas." je ne comprend pas comment reproduire ce problem ?

#6 - 03/05/2013 00:42 - Quentin CHERGUI

Pour reproduire le "Question section mismatch" :
1) Ajouter un alias vers un nom n'existant pas (par exemple www.april.org => thisnamedontexist.example.novalocal)
2) Interroger le serveur DNS (avec dig par exemple). La question est interceptée, réécrite, une tentative de résolution se fait, mais elle revient en erreur (logique). La réponse étant une erreur (NXDOMAIN), elle n'est pas interceptée par gotResolverResponse et n'est pas réécrite.
3) Dig reçoit une réponse a une question qu'il n'a pas posé, et il n'aime pas ça. Il retente la requête plusieurs fois (6 chez moi), puis il abandonne. Au final, on doit perdre une 20aine de secondes.

Solution : redéfinir gotResolverError de la même manière que l'on a redéfini gotResolverResponse, pour obtenir un NXDOMAIN avec la question posée initialement.

#7 - 03/05/2013 18:33 - Loïc Dachary

Minimal DNS spoofing daemon

#8 - 07/05/2013 11:18 - Loïc Dachary

- Version cible changé de Avril 2013 à Mai 2013

#9 - 07/05/2013 11:19 - Loïc Dachary

DNSChef is a highly configurable DNS proxy for Penetration Testers and Malware Analysts

```
 sudo python dnschef.py --fakedomains=foo.com --fakealias=foo.me
          _                 _           __
         | | version 0.2  | |         / _|
       __| |_ __  ___  ___| |__   ___| |_
      / _` | '_ \/ __|/ __| '_ \ / _ \  _|
     | (_| | | | \__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                   iphelix@thesprawl.org
[!] You have forgotten to specify which IP to use for fake responses
```

it's designed to intercept about everything and redirect to another domain.

#10 - 07/05/2013 11:53 - Loïc Dachary

dnsspoof is restricted to IPs and not domain names

#11 - 10/05/2013 00:11 - Loïc Dachary

The

```
stocks.yahoo.com    CNAME    www.google.com.
```

example from How to configure your BIND resolvers to lie using Response Policy Zones does exactly what we need. The only file to create is the RPZ zone file, including one line per rewrite, such as:

```
april.org CNAME april.novalocal
spip.libre-en-fete.org CNAME libre-en-fete.novalocal
```

It will probably make sense to setup bind as a separate instance because RPZ is only implemented in bind9 >= 9.8.0 and squeeze has a lower version.

#12 - 10/05/2013 00:12 - Loïc Dachary

- % réalisé changé de 0 à 70

#13 - 17/05/2013 12:14 - Quentin CHERGUI

DNS RPZ successfully tested on Bind 9.8 on Debian Wheezy.
(Bind 9.8 is backported on Debian Squeeze)

#14 - 17/05/2013 13:32 - Quentin CHERGUI

Others docs for DNS RPZ :
http://www.isc.org/software/rpz
http://www.bortzmeyer.org/rpz-faire-mentir-resolveur-dns.html

#15 - 14/06/2013 17:48 - Loïc Dachary

les tests de generate_rpz_zone ne fonctionnent pas

```
+ set -o functrace
+ PS4=' ${FUNCNAME[0]}: $LINENO: '
 : 117: test_nodes2unbound
 test_nodes2unbound: 103: nodes='foo.bar foo.novalocal
tia.bar tia.novalocal tia.april.org'
 test_nodes2unbound: 106: expected='local-data: "foo.bar IN CNAME foo.novalocal"
local-data: "tia.bar IN CNAME tia.novalocal"
local-data: "tia.april.org IN CNAME tia.novalocal"'
  test_nodes2unbound: 108: echo 'foo.bar foo.novalocal
tia.bar tia.novalocal tia.april.org'
  test_nodes2unbound: 108: nodes2unbound
  nodes2unbound: 42: read nodes
  nodes2unbound: 44: for node in '$nodes'
  nodes2unbound: 46: echo foo.bar
  nodes2unbound: 46: grep '.novalocal$'
  nodes2unbound: 44: for node in '$nodes'
  nodes2unbound: 46: echo foo.novalocal
  nodes2unbound: 46: grep '.novalocal$'
  nodes2unbound: 48: mainname=foo.novalocal
  nodes2unbound: 51: for node in '$nodes'
  nodes2unbound: 53: '[' foo.bar '!=' foo.novalocal ']'
  nodes2unbound: 59: echo 'foo.bar IN CNAME foo.novalocal.'
  nodes2unbound: 51: for node in '$nodes'
  nodes2unbound: 53: '[' foo.novalocal '!=' foo.novalocal ']'
  nodes2unbound: 42: read nodes
  nodes2unbound: 44: for node in '$nodes'
  nodes2unbound: 46: echo tia.bar
  nodes2unbound: 46: grep '.novalocal$'
  nodes2unbound: 44: for node in '$nodes'
  nodes2unbound: 46: echo tia.novalocal
  nodes2unbound: 46: grep '.novalocal$'
  nodes2unbound: 48: mainname=tia.novalocal
```

```
  nodes2unbound: 44: for node in '$nodes'
  nodes2unbound: 46: echo tia.april.org
  nodes2unbound: 46: grep '.novalocal$'
  nodes2unbound: 51: for node in '$nodes'
  nodes2unbound: 53: '[' tia.bar '!=' tia.novalocal ']'
  nodes2unbound: 59: echo 'tia.bar IN CNAME tia.novalocal.'
  nodes2unbound: 51: for node in '$nodes'
  nodes2unbound: 53: '[' tia.novalocal '!=' tia.novalocal ']'
  nodes2unbound: 51: for node in '$nodes'
  nodes2unbound: 53: '[' tia.april.org '!=' tia.novalocal ']'
  nodes2unbound: 59: echo 'tia.april.org IN CNAME tia.novalocal.'
  nodes2unbound: 42: read nodes
 test_nodes2unbound: 108: output='foo.bar IN CNAME foo.novalocal.
tia.bar IN CNAME tia.novalocal.
tia.april.org IN CNAME tia.novalocal.'
 test_nodes2unbound: 109: '[' '!' 'foo.bar IN CNAME foo.novalocal.
tia.bar IN CNAME tia.novalocal.
tia.april.org IN CNAME tia.novalocal.' = 'local-data: "foo.bar IN CNAME foo.novalocal"
local-data: "tia.bar IN CNAME tia.novalocal"
local-data: "tia.april.org IN CNAME tia.novalocal"' ']'
 test_nodes2unbound: 111: echo 'Error : foo.bar IN CNAME foo.novalocal.
tia.bar IN CNAME tia.novalocal.
tia.april.org IN CNAME tia.novalocal. instead of local-data: "foo.bar IN CNAME foo.novalocal"
local-data: "tia.bar IN CNAME tia.novalocal"
local-data: "tia.april.org IN CNAME tia.novalocal"'
Error : foo.bar IN CNAME foo.novalocal.
tia.bar IN CNAME tia.novalocal.
tia.april.org IN CNAME tia.novalocal. instead of local-data: "foo.bar IN CNAME foo.novalocal"
local-data: "tia.bar IN CNAME tia.novalocal"
local-data: "tia.april.org IN CNAME tia.novalocal"
 test_nodes2unbound: 112: return 2
 : 117: exit 1
```

#16 - 14/06/2013 17:52 - Loïc Dachary

Je pense qu'il serait plus simple dans change-resolver.sh de se contenter de décommenter / substituer la ligne "prepend domain-name-servers 127.0.0.1;" qui existe déjà dans le fichier par défaut, au lieu d'utiliser un template.

#17 - 14/06/2013 17:54 - Loïc Dachary

Il serait mieux de mettre 512MB de RAM

#18 - 17/06/2013 18:17 - Quentin CHERGUI

- Version cible changé de Mai 2013 à Juin 2013

- % réalisé changé de 70 à 90

Reste à faire :

- changer les tests
- Corriger setupdns pour qu'il ne démarre la VM que si elle est déjà démarrée

#19 - 19/06/2013 18:04 - Quentin CHERGUI

- Statut changé de En cours de traitement à Résolu

- % réalisé changé de 90 à 100

Les tests sont bon, le module est terminé.

#20 - 29/05/2019 12:19 - Quentin Gibeaux

- Statut changé de Résolu à Fermé