DNSSEC Service Monitoring with Nagios

DNSSEC signatures expire if they are not replaced. This can make DNS names disappear for clients who are doing DNSSEC validation. We can avoid this kind of thing by monitoring our zones in software and sounding the alarm if signatures look like they are getting dangerously close to their expiry.

The goal is to find out that something bad might happen before it actually does. This is actually a useful goal for everything in life, but in this workshop we are mainly concentrating on DNSSEC.

OBJECTIVES

In this exercise we are going to:

- install Nagios3 (monitoring platform) with Apache2 (web server)
- install the world-famous NSRC DNSSEC signature validity plugin
- configure our lab hosts for monitoring
- configure DNSSEC validity monitoring for our signed zone
- play with the web page and extend as seems useful
- relax and luxuriate in the comfortable certainty that EVERYTHING is FINE

INSTALLING THE SOFTWARE

Since we are more interested in using the software in this workshop than understanding in enormous detail how it is built, we will use packages to make this part very easy.

We will install everything on the Client machine cli.grpXX.bw.te-labs.training.

```
sysadm@cli:~$
```

1. Update the local package repository so that it is current.

You probably don't need to do this in the lab, but it's a good idea if you're playing around on another server where the package repository might not be current. Feel free to do it, though. It does no harm.

```
sysadm@cli:~$ sudo apt-get update
```

2. Install Apache2 and Nagios4 from packages

```
sysadm@cli:~$ sudo apt-get install apache2
```

```
sysadm@cli:~$ sudo apt-get install nagios4-core nagios4-cgi
```

You may be asked some questions during the package installation process, depending on what packages are already installed.

- If postfix is installed as a dependency, select "Local only" for the general type of mail system this server will be.

- If you are asked to specify the default "mail name" for the server because postfix is being installed, just accept the default (the hostname of the machine).

- You will be asked for a password for a web user, "nagiosadmin". To make things easy in the workshop, use the same password as you have been using for the sysadm user.

3. Check that Apache and Nagios are running

Point a browser on your laptop at:

  http://cli.grpXX.bw.te-labs.training/nagios4

You should be challenged for a username and password. Use the username "nagiosadmin" and the password you specified during the nagios3 package installation.

If you clicked on that link and it didn't work, look at it again and realise that you must replace XX with your group number. Doing this before you ask for help will make the world a better place, and Simon will give you a lollipop.

4. Install DNSSEC signature validity Nagios plugin

The plugin we will use is this one:

  https://github.com/ableyjoe/checksig.sh

It's a simple shell script that you should feel free to extend and modify independently. If you have any good ideas about how it could be made better, you know who to contact!

We will download the script itself directly to the server:

```
  sysadm@cli:~$ cd
  sysadm@cli:~$ wget
'https://raw.githubusercontent.com/ableyjoe/checksig.sh/master/checksig.sh'
```

You can now copy the script to the Nagios plugins directory and make it executable:

```
  sysadm@cli:~$ sudo mv checksig.sh /usr/lib/nagios/plugins/
  sysadm@cli:~$ sudo chmod 755 /usr/lib/nagios/plugins/checksig.sh
  sysadm@cli:~$
```

This plugin needs a particular utility to be installed, "gawk". So make sure that it is installed:

```
  sysadm@cli:~$ sudo apt-get install gawk
```

5. Add workshop configuration to Nagios

We will add the various hosts in our group environment to Nagios, and

install some DNSSEC signature validity checks against all the servers.

Depending on how the workshop has been arranged this time, some of
your servers might not be running nameservers. Don't worry about
that -- they will show up as warnings or errors in Nagios, and
that's ok. Add them to the configuration anyway.

```
  sysadm@cli:~$ cd /etc/nagios4/conf.d
  sysadm@cli:/etc/nagios4/conf.d$
```

We will create a new file in this directory containing
our workshop-specific configurations. We shall call it
bw-workshop.cfg. Use your favourite editor to create the file
(i.e. replace EDITOR below with nano or vi or whatever):

```
  sysadm@cli:/etc/nagios4/conf.d$ sudo EDITOR bw-workshop.cfg
```

Adapt the following configuration and type it in. Don't just cut
and paste the following, or it won't work: you need to set addresses
and names that are appropriate to your group. Feel free to modify
things, especially if you are a Secret Nagios Ninja and know all
the Secret Tricks.

```
---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---

define host {
        host_name               cli.grpX
        alias                   GRPX Client Server
        address                 100.100.x.2
        check_command           check-host-alive
        check_interval          5
        retry_interval          1
        max_check_attempts      5
}

define host {
        host_name               soa.grpX
        alias                   GRP0 Master Server
        address                 100.100.x.66
        check_command           check-host-alive
        check_interval          5
        retry_interval          1
        max_check_attempts      5
}

define host {
        host_name               ns1.grpX
        alias                   GRPX Authoritative Server 1
        address                 100.100.X.130
        check_command           check-host-alive
        check_interval          5
        retry_interval          1
        max_check_attempts      5
}

define host {
        host_name               ns2.grpX
        alias                   GRPX Authoritative Server 2
        address                 100.100.X.131
        check_command           check-host-alive
        check_interval          5
```

```
        retry_interval          1
        max_check_attempts      5
}


define host {
        host_name               resolv1.grpX
        alias                   GRPX DNSSEC Validator
        address                 100.100.X.67
        check_command           check-host-alive
        check_interval          5
        retry_interval          1
        max_check_attempts      5
}

define host {
        host_name               resolv2.grpX
        alias                   GRPX DNSSEC Validator
        address                 100.100.X.68
        check_command           check-host-alive
        check_interval          5
        retry_interval          1
        max_check_attempts      5
}


define hostgroup {
        hostgroup_name  grpX-servers
        alias           GRPX Servers
        members      cli.grpX,soa.grpX,ns1.grpX,ns2.grpX,resolv1.grpX
}

define service {
        hostgroup_name          grpX-servers
        service_description     grpX.ls-sig-validity
        check_command           checksig-grpX
        max_check_attempts      5
        check_interval          5
        retry_interval          3
}

define command {
        command_name            checksig-grpX
        command_line            /usr/lib/nagios/plugins/checksig.sh
$HOSTADDRESS
$ grpX.ls 40m 20m
}

---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---8<---

6. Restart Nagios to activate the new configuration

  sysadm@soa:/etc/nagios4/conf.d$ sudo service nagios4 restart
  sysadm@soa:/etc/nagios4/conf.d$


7. Use your web browser to check your services

Same URL as before; look under "Services" on the left-hand-side
navigation panel. You should see signature validity checks for the
DNS servers you configured in step 6.
```