

hidden/unvisible hostgroups or service-sets - how to use

The "hidden hostgroup" easter egg implemented in Nagios 4 means you can

use hostgroups which have "register 0" to assign services to. It was

originally an idea by Christian Anton actually (waaaay back, afair),

but got implemented by accident during the Nagios 4.0 config revamp

as a sideeffect of the optimizations.

Basically, it means that you can create a config like this;

```
define hostgroup {
    hostgroup_name    _sset_pop3server
    # register 0 means this never shows up in the UI
    register          0
}

define service {
    use                some-random-service-template
    check_command     check_pop3!$host:_SSETPOP3WARN$!$host:_SSETPOP3CRIT$
    service_description POP3 service
}

define host {
    use some-random-host-template
    host_name pop3server
    hostgroups _sset_pop3server
    _SSETPOP3WARN 10
    _SSETPOP3CRIT 20
}
```

and things will just work.

Note that in order to get any kind of performance out of using this on a larger-than-pretty-small scale, it's necessary to implement some caching in Nagios so most command macros are only resolved once instead of every time. There's a plan for that, and it can be implemented pretty quickly.

The (very loose) idea is to add some sort of support for this in Nacoma. Possibly by converting existing "host profiles" or by using the already-coined term "service sets" and build basic monitoring for particular types of equipment (such as ms-nsclient, unix-nrpe, server-snmp, vmware-guest etc, etc).

The long-term idea is still to implement "callable" service sets, where one can do things like

```
define host {
    host_name MS Exchange 1
    service_set msexchange(warn=10,crit=20,sendqwarn=1000)
}

define serviceset {
    serviceset_name mail-server

    define service {
        service_description POP3 service
        check_command check_pop3!{%warn}!{%crit}
    }

    define service {
        service_description SMTP service
        check_command $PLUGIN_DIR$/check_smtp!{%warn}!{%crit}
    }
}

define serviceset {
    serviceset_name ms-exchange
    includes mail-server

    define service {
        service_description MExchange SendQ
        check_command $PLUGIN_DIR$/check_nsclient!{%sendqwarn}
    }
}
```

although it's possible we'll make a special prefix (sset_) for hosts to allow them to pass variables to service-sets as well.

If you want to make servicesets right now, please utilize the `_sset` prefix for custom variables for hosts. Use the same prefix for `hostgroup_names`, and use `_service_set` "somename" for services you want included in the set. That will allow us to automagically manage servicesets.

The ui honey on top of the Nagios syntax voodoo will most likely be delivered in 6.2 in some form or other, even if it appears as a fairly simple hack.