非c/s模式的puppet部署

huangmingyou@gmail.com

2013-04-11

c/s模式的弊端

puppet有可以有两种执行模式，一种是c/s模式，一种是单机模式。通常官方推荐在生产环境中用c/s模式部署。但是c/s模式有几个弊端，一是pupetmaster的安全性，二是客户端多了以后的效率问题，还有一个就是证书的管理比较繁琐。

1. puppet master的安全性： c/s模式中，puppet master和puppet agent是用https的方式交换信息，这只能保证信息在传送的过程中不被截取和修改。但是，如果puppet master本身就被黑客入侵，发布的代码本身就被修改过了呢？最坏的情况就是所有的puppet agent被破坏，例如代码里面给你加一条exec{ “evel”: command=>”mkfs.ext3 /dev/sda”;} 。 很多公司也许可以通过把agnet和master 作到同一个vpn内网里面来减少这个风险，但是，如果危险本身就来自内部呢？
2. 效率问题： 部署过上百台的agent以后，puppet master的效率问题就浮现出来了，说白了，master 就一个web服务器，如果agent过多，那问题的实质就成了如何部署一个高效率的web网站的问题了。但是，真的有必要投入大量的资源来部署过多的master吗？
3. 证书问题： 通常是puppet新手都会遇到的第一个问题，为了维护证书，你需要妥善的管理每个主机的主机名，极端的情况，还需要维护一个内部的dns。

解决方案:单机部署puppet

单机部署puppet,也能使用facter的变量，也可以使用template,基本上满足这两点，大部分的应用场景都可以应付了。 单机部署的思路是， 每台客户端直接通过网络下载puppet的代码（manifest)到本地执行。是的，就这么简单，但是稍微处理下，就能解决上面提到的三个问题。

首先，准备一对gnupg密钥对，把公钥分发到所有的客户端机器，私钥自己妥善保管到本地。极端的方法是保存到一个没有联网的机器上。每次修改了puppet代码，都用私钥对代码签名或者是签名+加密。 然后客户端从网络上下载到puppet代码以后，首先用公钥来解密和验证签名，如果签名不对，就不执行代码。 这样，通过http或者ftp下载代码，不需要部署master,也不会有性能问题。其次，因为没有了和puppaster的https交互，所以也不需要证书了，但是，如果你要区别不同主机的配置，你还是需要管理主机名。最后，因为对下载的代码进行了gpg的签名认证，能保证代码没有被篡改。不过前提是你要妥善保护你的gpg私钥。我现在是把私钥保存我在本地pc的加密磁盘上，就算有人把我硬盘拿走也不能得到私钥。另外私钥本身也有密码。同时我的pc是linux系统，没有监听任何的网络端口。当然，这都不能保证绝对的安全。 按照标

准的做法，gpg 私钥是不能保存在联网的机器上的。要从物理上隔离。不过，安全是无止境的，做到你认为安全的程度就够了。 说这么多，估计各位看官还是有点不太明白，那我贴一下我生产环境上客户端的一个执行脚本，这个脚本是用crontab来定时执行，从网络上用rsync来下载puppet代码，作gpg签名认证，然后执行代码。

```
#!/bin/bash
[ -f /nopuppet ]&&exit 0
# 如果有/nopuppet文件，说明本机不希望进行puppet配置

rm /opt/puppet/puppet* -rf
gpg --list-keys |grep B5D310A1||gpg --keyserver pgp.mit.edu --recv-key B5D310A1
# 如果是新机器，导入公钥
export RSYNC_PASSWORD=”rsync密码”
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
rsync -avz --delete --exclude='.svn' puppet@www.example.com::puppet/ /opt/puppet/

#用rsync下载代码

cd /opt/puppet/
mkdir /tmp/xy-puppet
tar zxf puppet.tgz -C /tmp/xy-puppet
gpg --list-keys |grep B5D310A1||gpg --import /tmp/xy-puppet/puppet/file/hmy.gpg
gpg --verify puppet.tgz.asc

#关键，签名认证
[ $? -ne 0 ]&&exit 0
# 如果签名不对，退出脚本，不执行后面的代码。
rsync -avz --delete /tmp/xy-puppet/puppet/ /etc/puppet/
puppet /etc/puppet/manifests/site.pp
```

通常，可以把puppet代码组织到名为puppet的目录下，用git或者svn管理起来，修改代码以后，把puppet打包成一个压缩文件，便于签名。下面是我的pc上的一个Makefile ,修改好代码以后，执行make,输入gpg密码后，就会把puppet代码发布到网上，等待客户端来更新。

```
all:

	@make clean
	@tar czpf puppet.tgz puppet
	@gpg -a -b -s puppet.tgz
	@upload-puppet
	@rm puppet.tgz puppet.tgz.asc -rf 2>/dev/null;echo "all done"
```

有什么疑问，email: haungmingyou@gmail.com。