<http://blog.51cto.com/467754239/1558861>

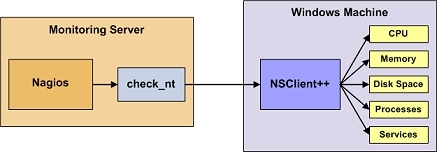
**一、简介**

1、获取更多的NSClient原理，请参考NSClient++官方站点

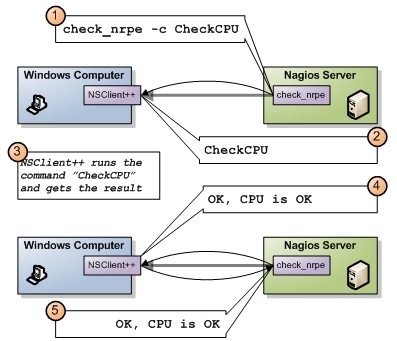
NSClient++官网：<http://www.nsclient.org/>

2、NSClient++与NRPE

NSClient++的工作原理

[](http://s3.51cto.com/wyfs02/M02/4A/E9/wKiom1QnYaewuJNPAACIXWqRWWw312.jpg)

NRPE的工作原理

[](http://s3.51cto.com/wyfs02/M00/4A/F0/wKiom1Qnc3zAjECMAAEUsOqv9kM145.jpg)

Nagios对Windows主机的监控主要有三种方法

第一种是NSclient++

第二种是NRPE

第三种是SNMP(不是很常用)

NSclient++与nrpe最大的区别就是:

1、被监控机上安装有nrpe,并且还有插件,最终的监控是由这些插件来进行的.当监控主机将监控请求发给nrpe后,nrpe调用插件来完成监控.

2、NSclient++则不同,被监控机上只安装NSclient++,没有任何的插件.当监控主机将监控请求发给NSclient++后,NSclient++直接完成监控,所有的监控是由NSclient++完成的。

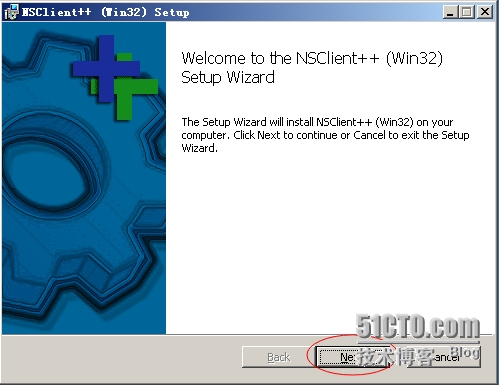
这也说明了NSclient++的一个很大的问题,不灵活,没有可扩展性.它只能完成自己本身包含的监控操作,不能由一些插件来扩展.好在NSclient++已经做的不错了,基本上可以完全满足我们的监控需要。

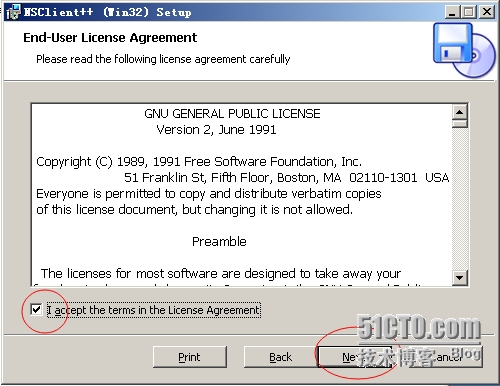
**二、check\_nt的方式监控windows主机**

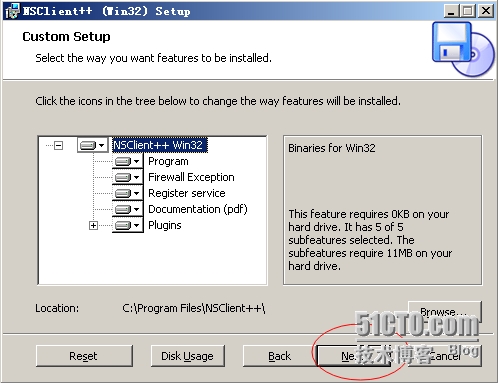
1、下载NSClient++

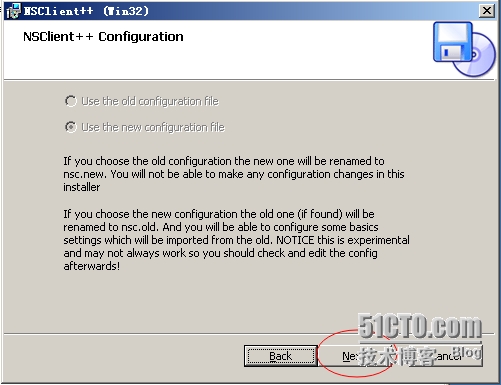
下载地址：<http://sourceforge.net/projects/nscplus/files/nscplus/>

2、安装NSClient++

[](http://s3.51cto.com/wyfs02/M01/4A/B2/wKiom1QmswLChu7BAAF5ugAoQLo820.jpg)

[](http://s3.51cto.com/wyfs02/M02/4A/B4/wKioL1QmsyqyCXr-AAHsG6bmgZw306.jpg)

[](http://s3.51cto.com/wyfs02/M02/4A/B2/wKiom1QmswOwzMYBAAGsdr9E0Pc403.jpg)

[](http://s3.51cto.com/wyfs02/M00/4A/B4/wKioL1QmsyqTfejsAAGRfdnH-xU473.jpg)

Allowed hosts：(this is the IP of the nagios (or other)server)

允许的主机地址：Nagios服务器端的IP地址

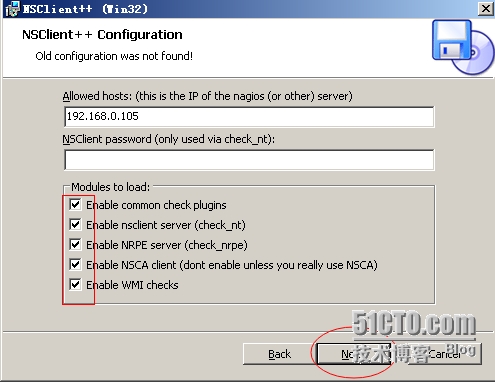
**NSClient password（only userd via check\_nt）**

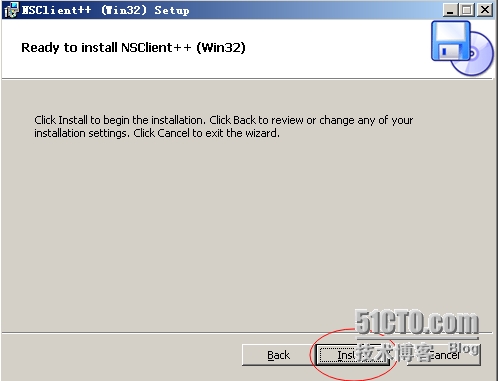
NSClient的密码：填写Nagios跟NSClient++进程通信的密码，可以不设置

Modules to load：

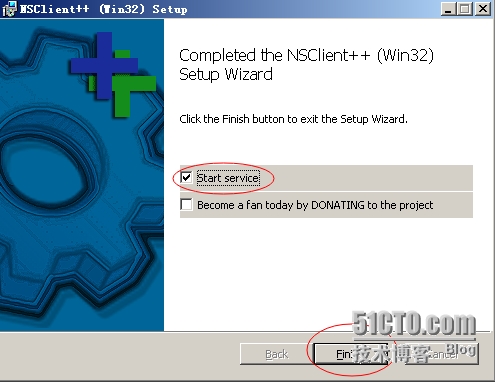
安装并加载相应的模块：NSClient++自带的有check\_plugins插件，check\_nt,check\_nrpe,NSCA,WMI

在这里我们全部选择，以后会有用到的。

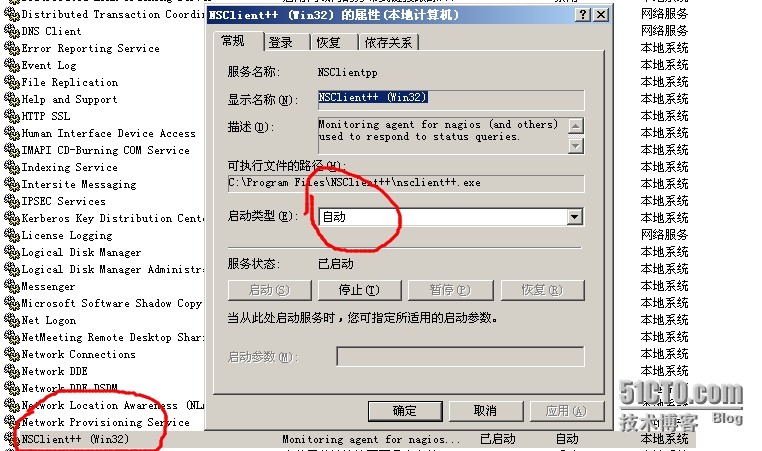
[](http://s3.51cto.com/wyfs02/M00/4A/B2/wKiom1QmswOTN7ZdAAF8CMhGqWs779.jpg)

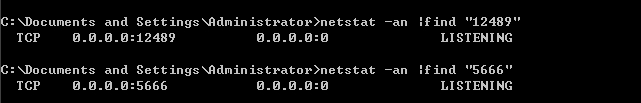
[](http://s3.51cto.com/wyfs02/M02/4A/B4/wKioL1Qmsyvjmrj1AADuL2UQatY830.jpg)

点击【Finsh】，表示安装NSClient++完成

[](http://s3.51cto.com/wyfs02/M01/4A/B4/wKioL1QmsyujK89FAAGCBr_MNf0266.jpg)

3、查看NSClient++服务是否已经启用

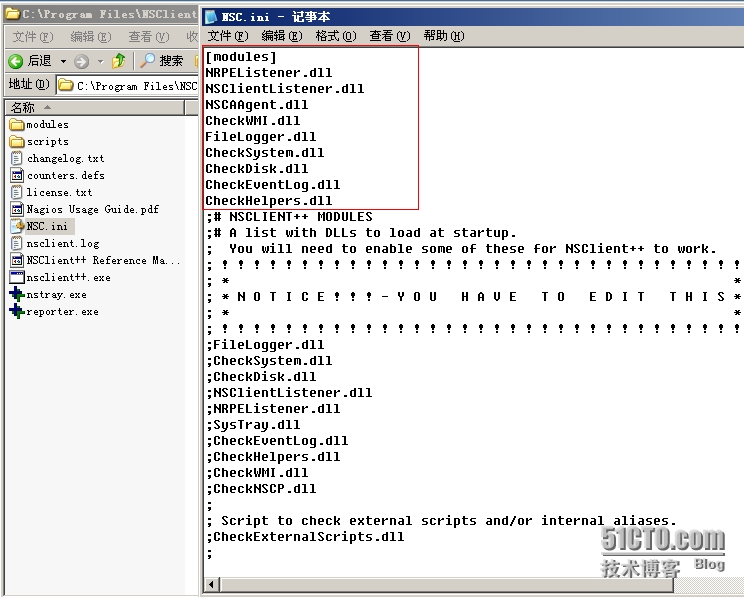
[](http://s3.51cto.com/wyfs02/M02/4A/E9/wKioL1QnXTmgMzAWAAQEAtOdHlA759.jpg)

[](http://s3.51cto.com/wyfs02/M01/4A/E9/wKiom1QnYYexKlUCAACG8_0iqhA152.jpg)

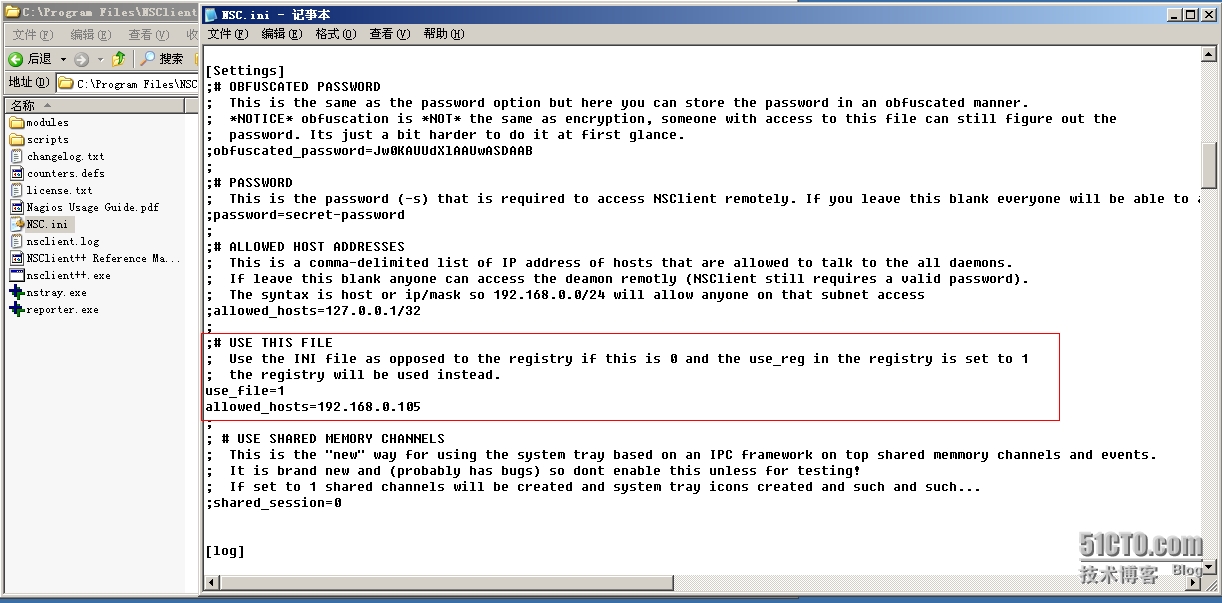
4、查看NSClient++的配置文件

默认是安装在C:\Program Files\NSClient++ 目录下，NSC.ini即为NSClient服务的配置文件，一般我们无需修改，但是当我们监控端的IP地址改变时，或者密码忘记，即可以在这里修改了。

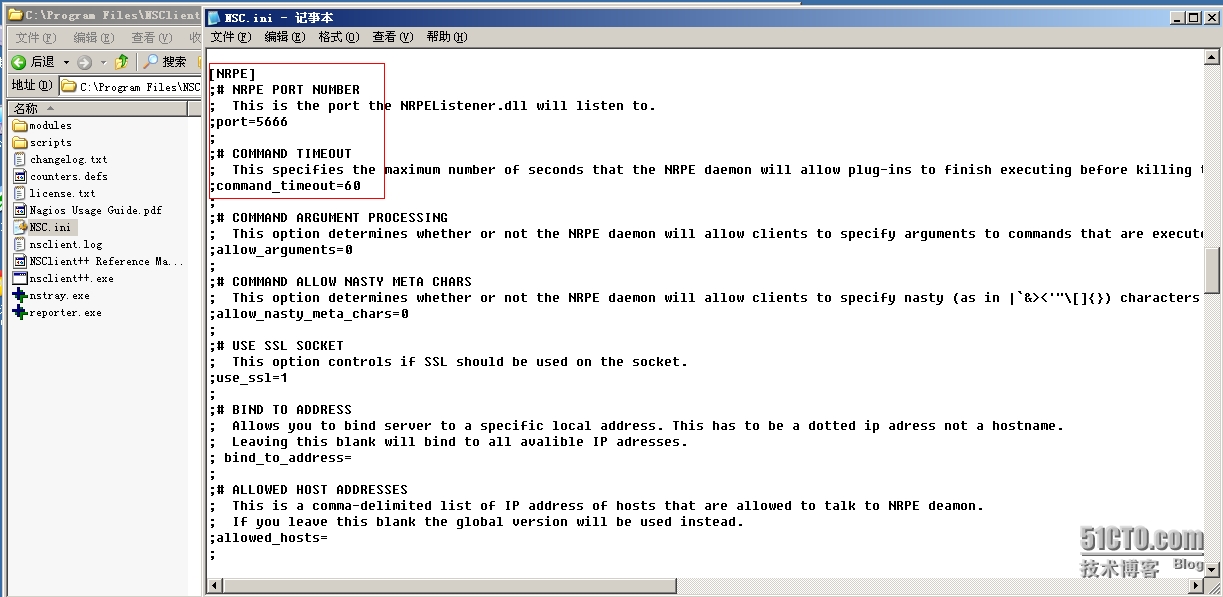
安装时加载的模块

[](http://s3.51cto.com/wyfs02/M00/4A/E9/wKioL1QnXLeisp6DAAOmUPyr8zQ318.jpg)

设置允许连接的地址：为Nagios的IP地址

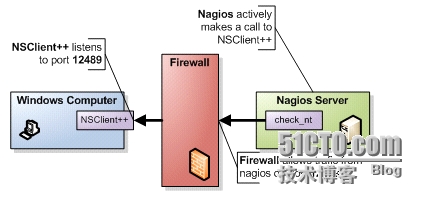
[](http://s3.51cto.com/wyfs02/M00/4A/E7/wKiom1QnXJDxt5BDAAZlbauD00Y308.jpg)

NRPE的默认端口号

[](http://s3.51cto.com/wyfs02/M01/4A/E9/wKioL1QnXLnCQe-sAATIjXBJX2M538.jpg)

三、NSClient应用监控

NSClient++与Nagios服务器通信，主要使用Nagios服务器的check\_nt插件。原理图如下

[](http://s3.51cto.com/wyfs02/M02/4A/E8/wKiom1QnXc7Rghy-AAC9NsiJ0jA271.jpg)

1、check\_nt插件的使用说明

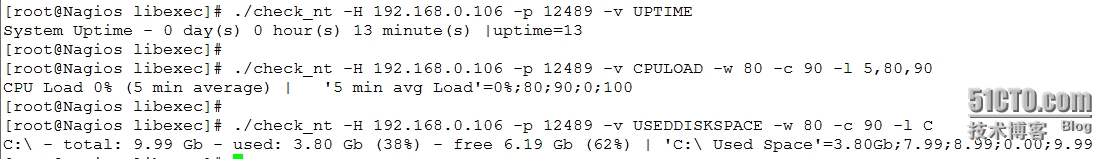
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115 | [root@Nagios ~]# cd /usr/local/nagios/libexec/  [root@Nagios libexec]# ./check\_nt -h   #查看check\_nt的语法  check\_nt v2.0.3 (nagios-plugins 2.0.3)  Copyright (c) 2000 Yves Rubin (rubiyz@yahoo.com)  Copyright (c) 2000-2014 Nagios Plugin Development Team      <devel@nagios-plugins.org>    This plugin collects data from the NSClient service running on a  Windows NT/2000/XP/2003 server.      Usage:  check\_nt -H host -v variable [-p port] [-w warning] [-c critical]  [-l params] [-d SHOWALL] [-u] [-t timeout]    Options:   -h, --help      Print detailed help screen   -V, --version      Print version information   --extra-opts=[section][@file]      Read options from an ini file. See      https://www.nagios-plugins.org/doc/extra-opts.html      for usage and examples.  Options:   -H, --hostname=HOST     Name of the host to check   -p, --port=INTEGER     Optional port number (default: 1248)  #默认端口号   -s, --secret=<password>                 #指定的密码     Password needed for the request   -w, --warning=INTEGER     Threshold which will result in a warning status   -c, --critical=INTEGER     Threshold which will result in a critical status   -t, --timeout=INTEGER     Seconds before connection attempt times out (default:  -l, --params=<parameters>     Parameters passed to specified check (see below) -d, --display={SHOWALL}     Display options (currently only SHOWALL works) -u, --unknown-timeout     Return UNKNOWN on timeouts10)   -h, --help     Print this help screen   -V, --version     Print version information   -v, --variable=STRING     Variable to check    Valid variables are:   CLIENTVERSION = Get the NSClient version    If -l <version> is specified, will return warning if versions differ.   CPULOAD =    Average CPU load on last x minutes.    Request a -l parameter with the following syntax:    -l <minutes range>,<warning threshold>,<critical threshold>.    <minute range> should be less than 24\*60.    Thresholds are percentage and up to 10 requests can be done in one shot.    ie: -l 60,90,95,120,90,95   UPTIME =    Get the uptime of the machine.    -l <unit>    <unit> = seconds, minutes, hours, or days. (default: minutes)    Thresholds will use the unit specified above.   USEDDISKSPACE =    Size and percentage of disk use.    Request a -l parameter containing the drive letter only.    Warning and critical thresholds can be specified with -w and -c.   MEMUSE =    Memory use.    Warning and critical thresholds can be specified with -w and -c.   SERVICESTATE =    Check the state of one or several services.    Request a -l parameters with the following syntax:    -l <service1>,<service2>,<service3>,...    You can specify -d SHOWALL in case you want to see working services    in the returned string.   PROCSTATE =    Check if one or several process are running.    Same syntax as SERVICESTATE.   COUNTER =    Check any performance counter of Windows NT/2000.      Request a -l parameters with the following syntax:      -l "\\<performance object>\\counter","<description>      The <description> parameter is optional and is given to a printf    output command which requires a float parameter.    If <description> does not include "%%", it is used as a label.    Some examples:    "Paging file usage is %%.2f %%%%"    "%%.f %%%% paging file used."   INSTANCES =    Check any performance counter object of Windows NT/2000.    Syntax: check\_nt -H <hostname> -p <port> -v INSTANCES -l <counter object>    <counter object> is a Windows Perfmon Counter object (eg. Process),    if it is two words, it should be enclosed in quotes    The returned results will be a comma-separated list of instances on     the selected computer for that object.    The purpose of this is to be run from command line to determine what instances     are available for monitoring without having to log onto the Windows server      to run Perfmon directly.    It can also be used in scripts that automatically create Nagios service     configuration files.    Some examples:    check\_nt -H 192.168.1.1 -p 1248 -v INSTANCES -l Process   #check\_nt的语法    Notes:   - The NSClient service should be running on the server to get any information     (http://nsclient.ready2run.nl).   - Critical thresholds should be lower than warning thresholds   - Default port 1248 is sometimes in use by other services. The error     output when this happens contains "Cannot map xxxxx to protocol number".     One fix for this is to change the port to something else on check\_nt     and on the client service it's connecting to.    Send email to help@nagios-plugins.org if you have questions regarding use  of this software. To submit patches or suggest improvements, send email to  devel@nagios-plugins.org |

2、check\_nt命令的使用

check\_nt参数解释

-w：警告比例

-c：紧急比例

-l：过去5分钟的平均值，80%的警告  90%紧急[](http://s3.51cto.com/wyfs02/M01/4A/B4/wKiom1QmuKqBRFWJAAJvlnFCVQ4764.jpg)

四、定义命令、主机、服务

1、定义命令

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | [root@Nagios ~]# vim /usr/local/nagios/etc/objects/commands.cfg  # 'check\_win' command definition  define command{          command\_name    check\_win          command\_line    $USER1$/check\_nt -H $HOSTADDRESS$ -p 12489 -v $ARG1$ $ARG2$          }  注释：  $..$    表示系统内置的宏，也就是所谓的变量  $USER1$ 表示插件所在的目录  -H      指定主机地址  $HOSTADDRESS$  应用到哪个主机，就用哪个主机的地址  $ARG1$  传递的参数,形参  -s      指定密码 默认为空 |

2、定义主机和服务

|  |  |
| --- | --- |
| 1  2  3 | [root@Nagios objects]# cp windows.cfg windows106.cfg  [root@Nagios objects]# sed -i 's/winserver/Windows106/g' windows106.cfg  [root@Nagios objects]# sed -i 's/192.168.1.2/192.168.0.106/' windows106.cfg |

4、检测配置文件是否有语法错误

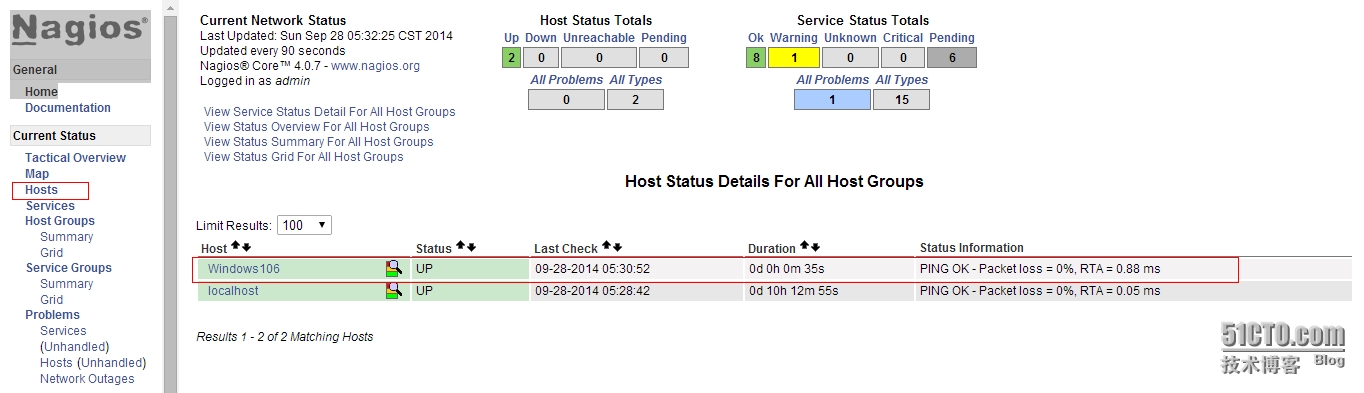
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | [root@Nagios ~]# service nagios configtest    Nagios Core 4.0.7  Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  Copyright (c) 1999-2009 Ethan Galstad  Last Modified: 06-03-2014  License: GPL    Website: http://www.nagios.org  Reading configuration data...     Read main config file okay...     Read object config files okay...    Running pre-flight check on configuration data...    Checking objects...      Checked 8 services.      Checked 1 hosts.      Checked 1 host groups.      Checked 0 service groups.      Checked 1 contacts.      Checked 1 contact groups.      Checked 25 commands.      Checked 5 time periods.      Checked 0 host escalations.      Checked 0 service escalations.  Checking for circular paths...      Checked 1 hosts      Checked 0 service dependencies      Checked 0 host dependencies      Checked 5 timeperiods  Checking global event handlers...  Checking obsessive compulsive processor commands...  Checking misc settings...    Total Warnings: 0  Total Errors:   0    Things look okay - No serious problems were detected during the pre-flight check  Object precache file created:  /usr/local/nagios/var/objects.precache |

5、重启nagios服务

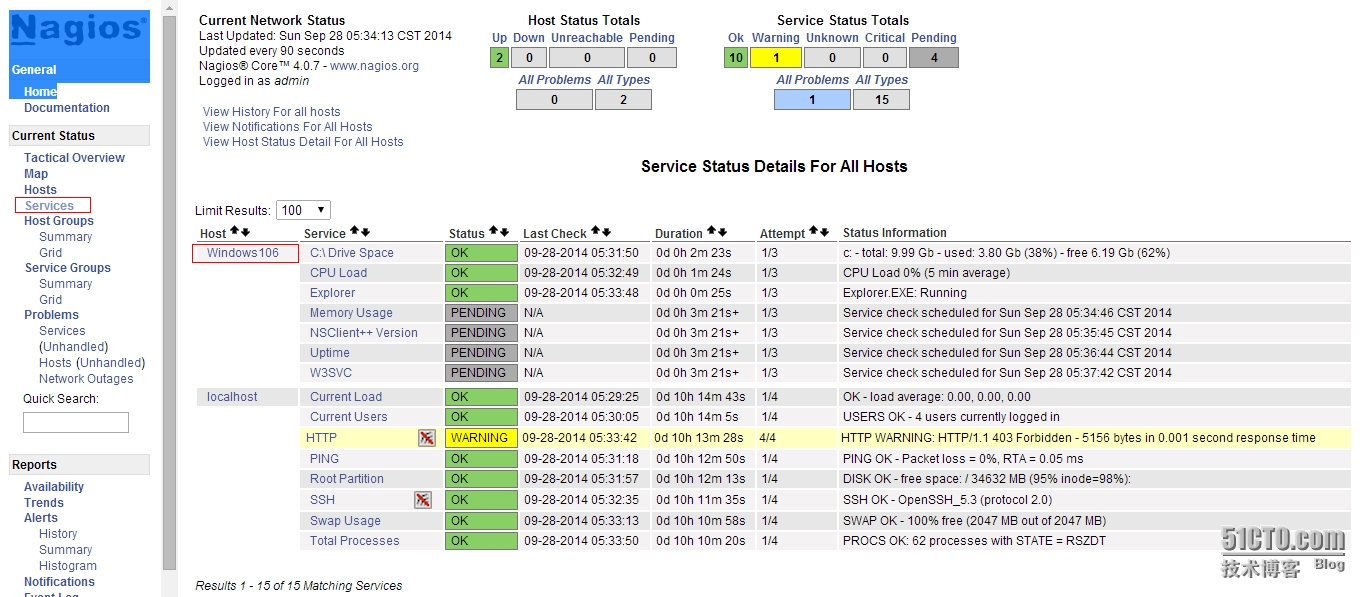
|  |  |
| --- | --- |
| 1  2  3  4 | [root@Nagios objects]# service nagios restart  Running configuration check...  Stopping nagios: .done.  Starting nagios: done. |

四、浏览器查看监控信息

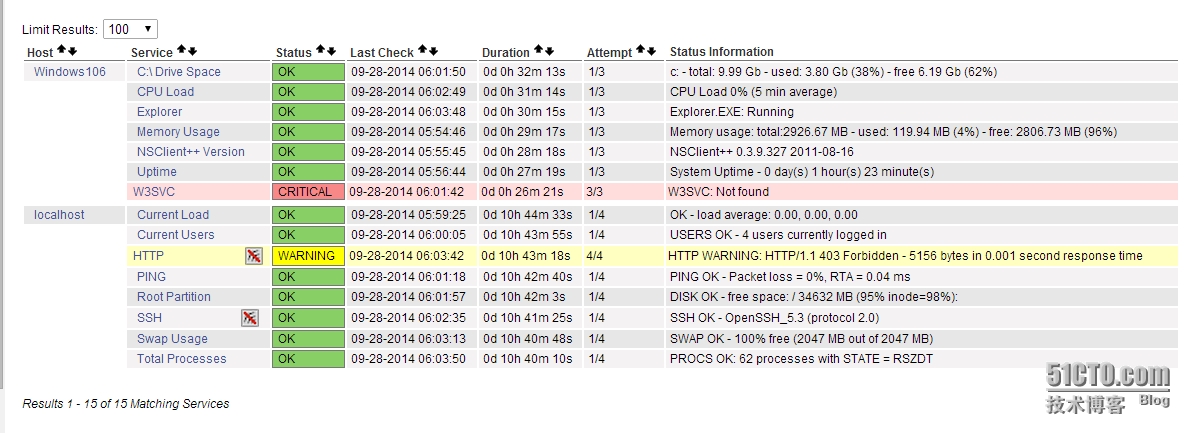
1、登陆后点击【Hosts】，查看新监控的Windows主机信息

[](http://s3.51cto.com/wyfs02/M00/4A/BB/wKioL1Qmw93SnhVOAAOI_HKVwgw436.jpg)

2、点击【Services】，查看Windows服务的状态信息

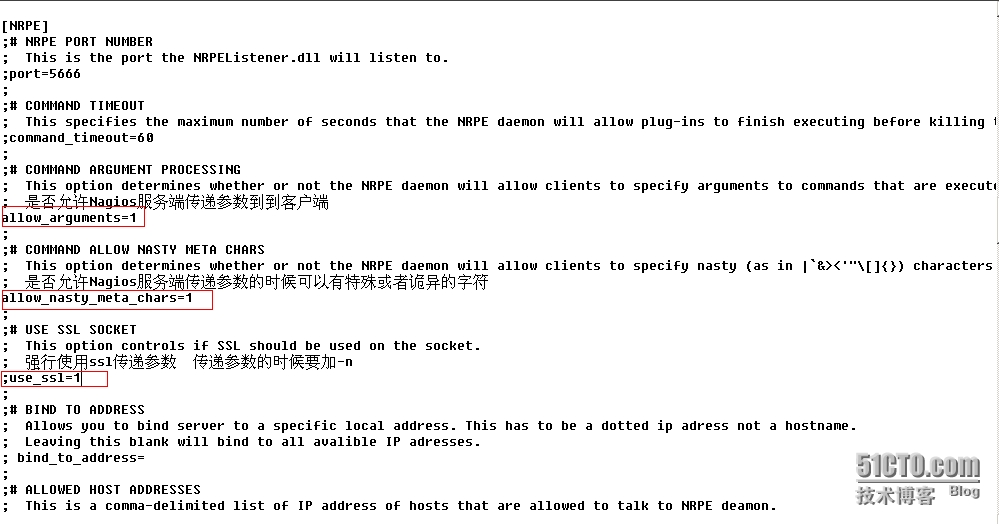
[](http://s3.51cto.com/wyfs02/M00/4A/B9/wKiom1Qmw7awxW6cAAfMmerFYOc506.jpg)

3、等待几分钟后状态就正常了，如下图所示

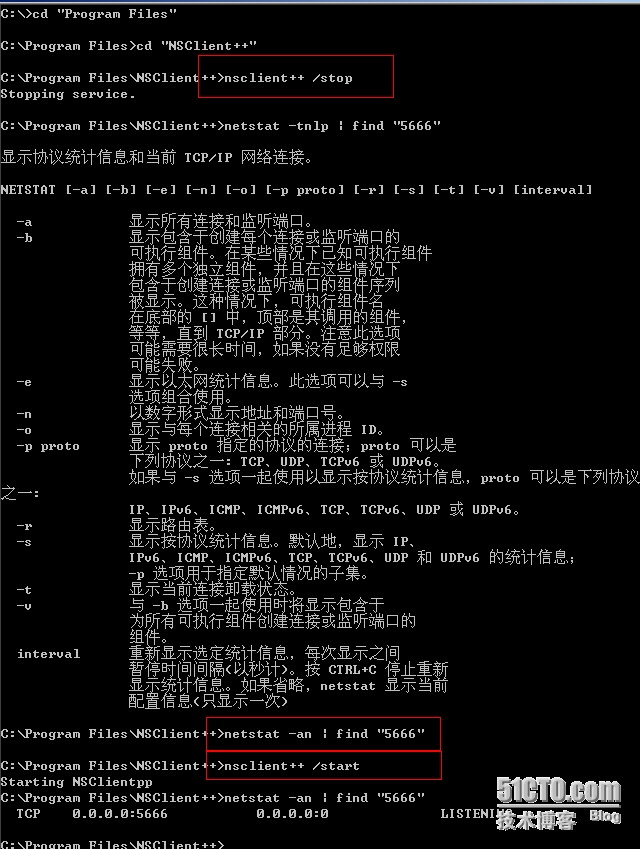
[](http://s3.51cto.com/wyfs02/M02/4A/BC/wKiom1Qmys7Sl-qRAAVcoYMMwSw515.jpg)

**三、NRPE的方式监控windows主机**

1、修改NSClient++的配置文件

[](http://s3.51cto.com/wyfs02/M02/4A/EE/wKioL1QnawbRvD3RAAO2J91L10Q919.jpg)

2、重新启动NSClient++服务

[](http://s3.51cto.com/wyfs02/M00/4A/EE/wKioL1Qna8GTPNRKAASz4_IEazg262.jpg)

3、Nagios服务端测试NRPE命令

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61 | [root@Nagios ~]# cd /usr/local/nagios/libexec/  [root@Nagios libexec]# ./check\_nrpe -h    NRPE Plugin for Nagios  Copyright (c) 1999-2008 Ethan Galstad (nagios@nagios.org)  Version: 2.15  Last Modified: 09-06-2013  License: GPL v2 with exemptions (-l for more info)  SSL/TLS Available: Anonymous DH Mode, OpenSSL 0.9.6 or higher required    Usage: check\_nrpe -H <host> [ -b <bindaddr> ] [-4] [-6] [-n] [-u] [-p <port>] [-t <timeout>] [-c <command>] [-a <arglist...>]    Options:   -n         = Do no use SSL   -u         = Make socket timeouts return an UNKNOWN state instead of CRITICAL   <host>     = The address of the host running the NRPE daemon   <bindaddr> = bind to local address   -4         = user ipv4 only   -6         = user ipv6 only   [port]     = The port on which the daemon is running (default=5666)   [timeout]  = Number of seconds before connection times out (default=10)   [command]  = The name of the command that the remote daemon should run   [arglist]  = Optional arguments that should be passed to the command.  Multiple                arguments should be separated by a space.  If provided, this must be                the last option supplied on the command line.    Note:  This plugin requires that you have the NRPE daemon running on the remote host.  You must also have configured the daemon to associate a specific plugin command  with the [command] option you are specifying here.  Upon receipt of the  [command] argument, the NRPE daemon will run the appropriate plugin command and  send the plugin output and return code back to \*this\* plugin.  This allows you  to execute plugins on remote hosts and 'fake' the results to make Nagios think  the plugin is being run locally.      check\_nrpe语法：  check\_nrpe ... -c <command> [-a <argument> <argument> <argument>]      check\_nrpe的内置命令：  · CheckAlwaysCRITICAL (check)  · CheckAlwaysOK (check)  · CheckAlwaysWARNING (check)  · CheckCPU (check)  · CheckCRITICAL (check)  · CheckCounter (check)  · CheckEventLog/CheckEventLog (check)  · CheckFile (check)  · CheckFileSize (check)  · CheckMem (check)  · CheckMultiple (check)  · CheckOK (check)  · CheckProcState (check)  · CheckServiceState (check)  · CheckTaskSched/CheckTaskSched (check)  · CheckUpTime (check)  · CheckVersion (check)  · CheckWARNING (check)  · CheckWMI/CheckWMI (check)  · CheckWMIValue (check) |
| 1  2 | [root@Nagios libexec]# ./check\_nrpe -H 192.168.1.142 -p 5666 -c CheckCPU -a warn=80 crit=90 time=20m time=10s time=4  OK CPU Load ok.|'20m'=0%;80;90 '10s'=0%;80;90 '4'=0%;80;90 |

©著作权归作者所有：来自51CTO博客作者zys467754239的原创作品，如需转载，请注明出处，否则将追究法律责任