American Journal of Networks and Communications
2013; 2(6): 140-148
Published online December 30, 2013 (http://www.sciencepublishinggroup.com/j/ajnc)
doi: 10.11648/j.ajnc.20130206.11

Working with TCP\IP based network monitoring system using Linux

Diponkar Paul[1, *], Shamsuddin Majamder[2]

[1]Department of Electrical and Electronic Engineering, Prime University, Dhaka, Bangladesh
[2]Department of Electrical and Electronic Engineering, World University of Bangladesh, Dhaka, Bangladesh

Email address:
dipo0001@ntu.edu.sg (D. Paul)

To cite this article:
Diponkar Paul, Shamsuddin Majamder. Working with TCP\IP Based Network Monitoring System Using Linux. *American Journal of Networks and Communications.* Vol. 2, No. 6, 2013, pp. 140-148. doi: 10.11648/j.ajnc.20130206.11

Abstract: Nagios is a stable, scalable and extensible enterprise-class network and system monitoring tool which allows administrators to monitor network and host resources such as HTTP, SMTP, POP3, disk usage and processor load. Originally Nagios was designed to run under Linux, but it can also be used on several UNIX o132perating systems. This chapter covers the installation and parts of the configuration of Nagios. The purpose of this paper is not only to introduce to everyone the concept of distributed monitoring with Nagios but capturing the beauty of it to improve the security of computer networks. Firstly, an introduction to Nagios will be discussed to provide readers a brief overview of what Nagios is. Next, it will discuss how distributed network monitoring is an essential part to information security. It will then proceed to introducing the requirements needed to build a distributed Nagios network monitoring environment and demonstrate how Nagios can be configured to construct a distributed monitoring environment that helps improve the state of security of distributed networks. In essence, companies hould be aware of the need for hiring specialized security analysts to perform round-the-clock systems monitoring to secure their resources.

Keywords: Nagois, CPAN, NET-SNMPD, CGI etc

1. Introduction

The document aims to assist users in installing, configuring, extending, troubleshooting and generally getting themost out of your Nagios system. It must be said that the documentation that ships with Nagios is excellent, and there are many of good tutorials that already exist. This book was written to complement the existing documentation that is already out there, and even go a litCtle further in some areas - especially for the beginners. For the true beginners in the audience, you'll find I have created a nice set of step by step instructions, walking you through the process of building a server from scratch, installing Nagios and configuring Nagios to monitor a typical network. Arguably the most useful source of Nagios information is the mailing list nagios-users. This list is frequented by expert Nagios developers and users alike and is *the* place to be if you're serious about Nagios. A great deal of information in this book has been obtained through the help of people on this list, I highly recommend subscribing and participating - to learn, and to help others learn. Almost every Nagios related question I've ever had has been answered and documented in the list archives. It's a good idea to check there before posting. There are several places that you can find information on Nagios. Here are some places you should bookmark. Nagios.org Documentation -- Nagios.org FAQ's --Nagios Mail Lists.The complexity of modern networks and systems is somewhat astounding, as any experienced System Administrator will tell you. Even seemingly small networks found in many Small/Medium Enterprises (SME's) can have extremely high levels of complexity in the systems they run. Nagios was designed as a rock solid framework for monitoring, scheduling and alerting. Nagios contains some very powerful features, harnessing them is not only a matter of understanding how Nagios works, but also how the system you're monitoring also works. This is an important realization. Nagios can't automatically teach you about complex systems, but it will be an valuable tool to help you in your journey. So what are the sorts of things Nagios can do? Nagios can do much more than this, but nevertheless here's a

list of A more technically accurate and complete list of Nagios' features can be found in the official documentation (http://www.nagios.org/docs/). The Nagios package doesn't contain any checking tools (called plugins) at all. Does that statement sound crazy? Sure, but let me explain. Nagios focuses on doing what it does best - providing a robust, reliable and extensible framework for any type of check that a user can come up with. So how does Nagios perform it's checking? A huge number of plugins already exist that extend Nagios to perform every type of check maginable. And if there isn't an existing check that already exists, you're free to write your own. The nagios-plugins package is separately maintained and can be downloaded from various sources.We will cover the Nagios plugins later on. than Galstad is the creator of Nagios. Karl DeBisschop, Subhendu Ghosh, Ton Voon, and Stanley Hopcroft arethe main plugin developers. Many other people have contributed to the project over the years by submitting bug reports, atches, ideas, suggestions, add-ons, plugins, etc. A list of some of the contributors can be found at the Nagios website. As with anything, each tool has it's own set of strengths and weaknesses[1]. Some are the applications may seem similar, but are very different and range from full blown SNMP management solutions, to simple applications with not much flexibility. Big Brother, OpenNMS, OpenView and SysMon (there are dozens more) are often compared to Nagios, however they are quite different in many respects. In my travels as an IT professional, Nagios is the most commonly used monitoring tool by far. There are lots of specialist companies that offer monitoring as a service. A large number of these use Nagios.It's very useful to conceptually understand how Nagios works. The following is a very simplified view on how Nagios works. Nagios runs on a server, usually as a daemon (or service). Nagios periodically run plugins residing (usually) on the same server, they contact (PING etc.) hosts and servers on your network or on the Internet. You can also have information sent to Nagios. You then view the status information using the web interface. You can also receive email or SMS notifications if something happens. Event Handlers can also be configured to "act" if something happens.For the cost of hardware these days, I usually recommend running a dedicated server. One complaint I often here from users (usually new to Linux/Unix or Open Source), is that Nagios is difficult to configure. The Nagios Book aims to dispel this myth, showing you how to build a Nagios server from scratch monitoring several hosts and sending alerts in a matter of hours. Note that it's not a competition when it comes to setting up any server, however I've heard reports of people complaining of taking days and even weeks to set up a Nagios server. This is simply not true, as you will find out! Please note that this document does not explain the how's and why's of how to build a server, that is beyond the scope of this material. My goal is to try and get users up and running quickly so they can experience the power of Nagios. Having said that, I have created some basic steps to build a FreeBSD server from scratch. These instructions were written and tested using FreeBSD 6.0, however they should also work just as well on earlier versions. If you're using Linux, some steps may be slightly different. I hope to add steps for each popular distro very soon. I plan to add some other distro specific information here eventually. If you have installation notes for another platform, 'We will glaadly add it.

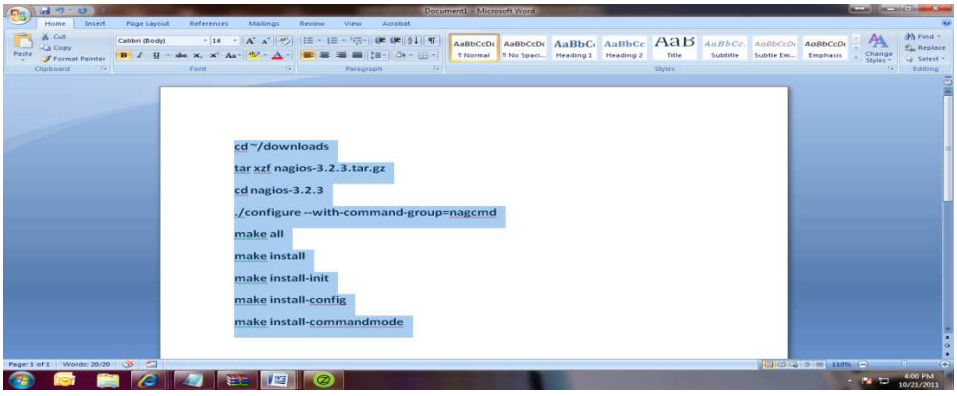

Figure 1: *Nagios install step by step command line.*

The simplest method of installation is for you to install the Nagios packages that are supplied with the distribution you are using. However, Nagios 2.0 is relatively new, so you may have to make do with an older Nagios version using this method. Configuring this is quite different from the version 2.0 described here, which is why it is recommended that you take things into your own hands and compile Nagios yourself if the distributor does not provide any Nagios 2.0 packages. If you are compiling Nagios yourself, you also have an influence on directory structures and several other parameters. A Nagios system compiled in this way also provides an almost complete main configuration file, in which, initially, nothing has to be changed. But it should be mentioned here that compiling Nagios yourself might involve a laborious search for the necessary development packages, depending on what is already installed on the computer. OK, so you know all about what Nagios can do, it's time to get on with it and choose an installation method[2].

You have two main choices when installing Nagios: For compiling Nagios itself you require gcc, make, autoconf and automake. Required libraries are libgd1 and openssl2. The development packages for these must also be installed (depending on the distribution, with either the ending -dev or -devel): libssl-dev, libgd-dev, libc6-dev. For the plugins it is recommended that you also install the following packages at the same time: ntpdate,3 snmp,4 smbclient,5 libldap2, and libldap2-dev,6 as well as the client and developer packages for the database to be used (e.g., postgresqlclient and postgresql-dev). The three commands unpack the source code into the directory created for this purpose, /usr/local/src. When this is done, a subdirectory with the name nagios-2.0b3 is also created. Before the actual compilation and installation, the groups required for operation, namely nagios and nagcmd, are set up with groupadd, and the user nagios, who is assigned to these groups and with whose permissions the Nagios server runs is set up with useradd: linux:˜ # groupadd -g 9000 nagios linux:˜ # groupadd -g 9001 nagcmd linux:˜ # useradd -u 9000 -g nagios -G nagcmd -d /usr/local/nagios \ -c "Nagios Admin" nagios Instead of the user (9000) and group IDs (9000 or 9001) used here, any other (available) ID may be used. The primary group nagios of the user nagios should remain reserved exclusively for this user.1 http://www.boutell.com/gd/2 http://www.openssl.org/ Depending on the distribution, the required RPM and Debian packages are sometimes named differently. Here you need to refer to the search help in the corresponding distribution. For Debian, the homepage will be of help. If a configure instruction complains, for example, of a missing gd.h file, you can search specifically at http://www.debian.org/distrib/packages for the contents of packages. For the configure command, parameters are specified that differ from the standard; The values chosen here ensure that the installation routine selects the directories used here in the book and that all parameters are correctly set when the main configuration file is generated. This considerably simplifies the fine-tuning of the configuration. If --prefix is not specified, Nagios installs itself in the directory /usr/local/nagios. We recommend that you stick to this directory.The system normally stores its configuration files in the directory etc beneath its root directory. In general it is better to store these in the /etc hierarchy, however. Here we use /etc/nagios.9 Variable data such as the log file and the status file are by default stored by Nagios in the directory /usr/local/nagios/var. This is in the /usr hierarchy, which should only contain programs and other read-only files, not writable ones. In order to ensure that this is the case, we use /var/nagios.10 Irrespective of these changes, in most cases configure does not run through faultlessly the very first time, since one package or another is missing. For required libraries such as libgd, Nagios almost always demands the relevant developer package with the header files (here, libgd-dev or libgd-devel). Depending on the distribution, their names end in -devel or -dev. After all the tests have been run through, configure presents a summary of all the important configuration parameters:

2. Installing and Testing Plugins

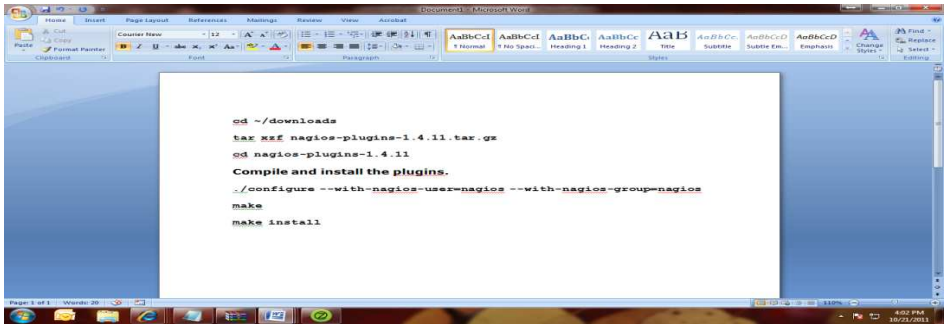

Figure 2: *Nagios Plugins install step by step command line.*

What is now still missing are the plugins. They must be downloaded separately from http://www.nagios.org/ and installed. As independent programs, they are subject to a different versioning system than Nagios. The current version at the time of going to press was version 1.4, but you can, for example, also use plugins from version 1.3.1 if you don't mind doing without the most recent features. Although the plugins are distributed in a common source distribution, they are independent of one another, so that you can replace one version of an individual plugin with another one at any time, or with one you have written yourself. Installation The installation of the plugin sources takes place, like the Nagios ones, in the directoryIf you are not using Radius, you need have no qualms in ignoring the corresponding error messages. Otherwise you should install the missing packages and repeat the configure procedure. The quite frequently required SNMP functionality is missing a Perl module in the example. This is installed either in the form of the distribution package or online via the CPAN archive:13 linux:˜ # perl -MCPAN -e install Net::SNMP If we are running the CPAN procedure for the first time, it will guide you interactively through a self-explanatory setup, and you can answer nearly all of the questions with the default option. Running make in the directory nagios-plugins-1.4 will compile all plugins. Afterwards you have the opportunity to perform tests, with make check. Because these have not been particularly carefully programmed, you will often see many error messages that have more to do with the test itself than with

the plugin. if you still want to try it, then the Cache Perl module must also be installed. Irrespective of make check, the most important plugins should be tested manually anyway after the installation. make install finally anchors the plugins in the subdirectory libexec (which in our case is /usr/local/nagios/libexec), but not all of them: the source directory contrib. contains a number of plugins that make install does not install automatically. Most plugins in this directory are shell or Perl scripts. Where needed, these are simply copied to the plugin directory /usr/local/nagios/libexec. The few C programs there are must first be compiled, which in some cases may be no laughing matter, since a corresponding makefile, and often even a description of the required libraries, is missing. If a simple make is not sufficient, as in the case of linux:nagios-plugins-1.4/contrib # make check_cluster214 cc check_cluster2.c -o check_cluster2 then it is best to look for help in the mailing list nagiosplug-help.15 The compiled program must also be copied to the plugin directory. The Comprehensive Perl Archive Network at http://www.cpan.org/.With check_cluster, hosts and services of a cluster can be monitored. Here you usually want to be notified if all nodes or redundant services provided fail at the same time. If one specific service fails on the other hand, this is not critical, as long as other hosts in the cluster provide this service.

http://lists.sourceforge.net/lists/listinfo/nagiosplug-help

Because plugins are independent programs, they can already be used manually for test purposes right now—before the installation of Nagios has been completed. In any case you should check the check_icmp plugin, which plays an essential role: it checks whether another computer can be reached via ping and is the only plugin to be used both as a service check and a host check. If it is not working correctly, Nagios will also not work correctly, since the system cannot perform any service checks as long as it categorizes a host as “down”. describes check_icmp in, which is why there is only short introduction here describing its manual use. In order for the plugin to function correctly it must, like the /bin/ping program, be run as the user root. This is done by providing it In order for the Web front end of Nagios to function, the Web server must know the CGI directory and the basis Web directory. The following description, with a slight deviation, applies to both Apache 1.3 and Apache 2.0. 1.3.1 Setting Up Apache As long as you have not added a different address for the front end, through the configure script with -with-cgiurl, it can be addressed under /nagios/cgi-bin. Since the actual CGI scripts are located in the directory /usr/local/nagios/sbin, a corresponding script alias is set in the Apache configuration: Here the directives Order and Allow also allow access only from the specified network. It is recommended that you write the above details in your own configuration file, called nagios.conf, so that this configuration is not lost during an Apache update, and place it in the Apache directory for individual configurations. This is usually to be found under /etc/apache/conf.d, but depending on the distribution and the Apache ersion, this could also be under /etc/httpd/conf.d or /etc/apache2/conf.d. In any case the Apache configuration file must integrate this directory with the directive Include. More recent SuSE distributions only accept files in the subdirectory conf.d that end in .conf. In the state in which it is delivered, Nagios allows only authenticated users access to the CGI directory. This means that users not “logged in” have no way to see anything other than the home page and the documentation. They are blocked off from access to other functions. There is a good reason for this: apart from status queries and other display functions, Nagios has the ability to send commands via the Web interface. The interface for external commands is used for this purpose (Section 13.1, page 240). If this is active, checks can be switched on and off via the Web browser, for example, and Nagios can even be restarted. Only authorized users should be in a position to do this. The easiest way to implement a corresponding authentication is via a .htaccess file in the CGI directory /usr/local/nagios/sbin.17 The document directory, on the other hand, requires no special protection. In addition, the parameter use_authentication in the CGI configuration file cgi.cfg18 of Nagios must be set to 1: The access rule described here, via .htaccess in the CGI directory, adheres to the official Nagios documentation. Those more familiar with Apache will have other configuration possibilities available, Name is just a comment that the browser displays if the Web server requests authentication. AuthType Basic stands for simple authentication, in which the password is transmitted without encryption, as long as no SSL connection is used. It is best to save the password file—here htpasswd—in the Nagios configuration directory /etc/nagios. The final parameter, require valid-user, means that all authenticated users have access (there are no restrictions for specific groups; only the user-password pair must be valid). In combination with its own modules and those of third parties, Apache allows a series of other authentication methods. These include authentication via an LDAP directory, via Pluggable Authentication Modules (PAM),19 or using SMB via a Windows server. Here we refer you to the relevant literature and the highly detailed documentation on the Apache home page at http://httpd.apache.org/. The (basically freely selectable) name of the password file will be specified here so that it displays what type of password file is involved. It is generated with the htpasswd2 program included in Apache (in Apache 1.3 the program is called htpasswd)Even though configuration of the Web interface is now finished, at the moment only the documentation is properly displayed: Nagios itself must first be correspondingly adjusted—as described in detail in the following chapter– before it can make usable monitoring data available in this way.Although the Nagios configuration can become quite large, you only need to handle a small part of this to get a system up and running. Luckily many parameters in Nagios are already set to sensible default settings. So this chapter will be primarily concerned with the most basic and frequently used parameters, which is quite sufficient for an initial configuration. Further details on the configuration are

provided by the chapters on individual Nagios features: in Chapter 6 about network plugins (page 85) there are many examples on the configuration of services. All parameters of the Nagios messaging system are explained in detail in Chapter 12, page 215, and the parameters for controlling the Web interface are described in Chapter 16 from page 273. In addition to this, Nagios includes its own extensive documentation, once it is installed, in the directory /usr/local/nagios/share/docs, which can also be reached from the Web interface. This can always be recommended as a useful source for further information, which is why each of the sections below refer to the corresponding location in the original documentation. The installation routine in make install-config stores examples of individual configuration files in the directory /etc/nagios. They all end in -sample, so that a possible update will not overwrite the files needed for productive operation. All subsequent work should be carried out as the user nagios. If you are editing files as the superuser, you must ensure yourself that the contents of directory /etc/nagios afterwards belong to the user nagios again. With the exception of the file resource.cfg—this may contain passwords, which is why only the owner nagios should have the read permission set—all other files may be readable for all.The central configuration takes place in nagios.cfg. Instead of storing all configuration options there, it makes links to other configuration files (with the exception of the CGI configuration). The easiest method is first to copy the example file: nagios@linux:/etc/nagios$ cp nagios.cfg-sample nagios.cfg Those who compile and install Nagios themselves have the advantage that at first they do not even need to adjust nagios.cfg, since all paths are already correctly set.1 And that's as much as you need to do. Nevertheless one small modification is recommended, which helps to maintain a clear picture and considerably simplifies configuration where larger networks are involved. The parameter concerned is cfg_file, which integrates files with object definitions). The file nagios.cfg-sample, included in the package, As an alternative to cfg_file, you can also use the parameter cfg_dir: this requests you to specify the name of a directory from which Nagios should integrate all configuration files ending in .cfg (files with other extensions are simply ignored). This also works recursively; Nagios thus evaluates all *.cfg files from all subdirectories. With the parameter cfg_dir you therefore only need to specify a signal directory, instead of calling all configuration files, with cfg_file, individually. The only restriction: these must be configuration files that describe objects. The configuration files cgi.cfg and resource.cfg are excluded from this, which is why, like the main configuration file nagios.cfg, they remain in the main directory /etc/nagios. For the object-specific configuration, it is best to create a directory called /etc/ nagios/mysite, then remove all cfg_file directives in nagios.cfg (or comment them out with a # at the beginning of the line) and replace them with the following:The main directory /etc/nagios contains only three configuration files and the password file for protected Web access. For the sake of clarity, the configuration examples *-sample should be moved to the directory sample. In this doc we will include all objects of a type in a file of its own, that is, all host definitions in the file hosts.cfg, all services in services.cfg, and so on. But you could just as well save each of the host definitions in a separate file for each host and use a directory structure to reflect this:http://mama.indstate.edu/users/ice/tree/Extended Service Information, like Extended Host Information. Not all object types are absolutely essential; especially at the beginning, you can easily do without the *dependency, *escalation, and *extinfo objects, as well as the servicegroup. looks at escalation and dependencies in detail. The extended information objects are used to provide a "more colorful" graphical representation, but they are not at all necessary for running Nagios. We refer here to the original documentation.3 Notes on the object examples below Although the following chapters describe individual object types in detail, only the mandatory parameters are described there and those that are absolutely essential for meaningful operation. Mandatory parameters here are always printed in bold type. The first (comment) line in each example lists the file in which the recorded object definition is to be stored. When you first start using Nagios, it is recommended that you restrict yourself to a minimal configuration with only one or two objects per object type, in order to keep potential sources of error to a minimum and to obtain a running system as quickly as possible. Afterwards extensions can be implemented very simply and quickly, especially if you take on board the tips mentioned in Section 2.11 on templates Time details in general refer to time units. A time unit consists of 60 seconds by default. It can be set to a different value in the configuration file nagios.cfg, using the parameter interval_length. You should really change this parameter only if you know exactly what you are doing.In order for the Web front end to work correctly, Nagios needs the configuration file cgi.cfg. The example included, called cgi.cfg-sample, can initially be taken over one-to-one, since the paths contained in it were set correctly during installation: nagios@linux:/etc/nagios$ cp sample/cgi.cfg-sample ./cgi.cfg Important: the file cgi.cfg should be located in the same directory as nagios.cfg, because the CGI programs have been compiled in this path permanently. If cgi.cfg is located in a different directory, the Web server must also be given an environment variable with the correct path, called NAGIOS_CGI_CONFIG. How this is set in the case of Apache is described in the corresponding online documentation at http://httpd.apache.org/docs-2.0/env.html. Out of the box, only a few parameters are enabled in the CGI configuration file. pre-flight check Although warnings displayed here can in principle be ignored, this is not always what the inventor had in mind: perhaps you made a mistake in the configuration, and Nagios is ignoring a specific object, which you would actually like to use. The first warning in the example refers to a host called linux02, which has not been allocated any services. Since Nagios works primarily with service checks, and uses host checks only if it needs them, a computer should basically always be allocated at least one service. Nagios

issues a warning, as here, if no service at all has been defined for a particular host.It is also recommended, however, to always define a “PING” service for every host, although this is not absolutely essential. Even if the same plugin, check_icmp, is used here as with the host check, this is not the same thing: the host check is satisfied with a single response packet, after all, it only wants to find out if the host “is alive”. As a service check, check_icmp registers packet run times and loss rates, which can be used to draw conclusions, if necessary, concerning existing problems with a network card. The second warning refers to a contact named wob, who, although defined, is not used, because he does not belong to any contact group. In contrast to warnings, genuine errors must be eliminated, because Nagios will usually not start if the parser finds an error, as in the following example:Here the configuration mistakenly contains a host called linux03, for which there is no definition. If you read through the error message carefully, you will quickly realize that the error can be found in the file /etc/nagios/mysite/services.cfg. In the definition of independencies (host and service dependencies, see Section 12.6 page 234) there is a fundamental risk that circular dependencies could be specified by mistake. Because Nagios cannot automatically resolve such dependencies, this is also checked before the start, and if necessary, an error is displayed. When using the parents parameter, it is also possible that two hosts may inadvertently serve mutually as “parents”; Nagios also test this.

3. Getting Monitoring Started

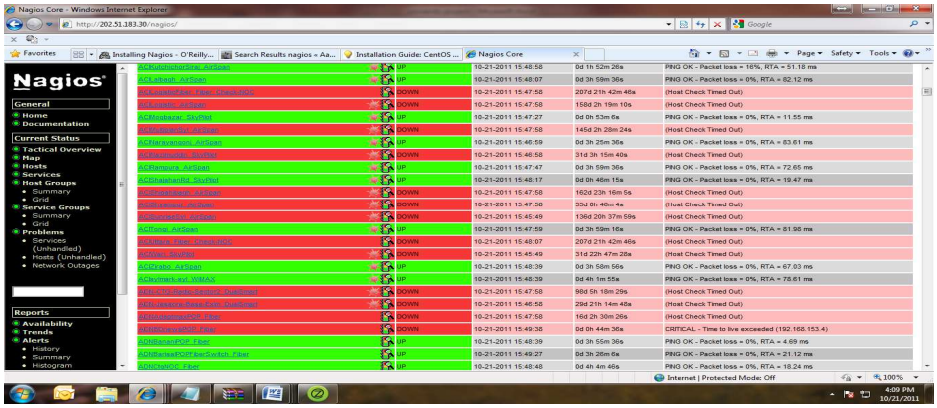

Figure 3: *Host and service status page.*

is causes Nagios to reread the configuration, end tests for hosts and services that no longer exist, and integrate new computers and services into the test. However, with each reload there is a renewed scheduling of checks, meaning that Nagios plans to carry out all tests afresh. To prevent all tests from being started simultaneously at bootup, Nagios performs a so-called spreading. Here the server spreads the start times of the tests over a configurable period.1 For a large number of services, it can therefore take a while before Nagios continues the test for a specific service. For this reason you should never run reloads at short intervals: in the worst case, Nagios will not manage to perform some checks in the intervening period and will perform them only some time after the most recent reload. Before being reloaded, the configuration is tested to eliminate any existing errors.With the Simple Network Management Protocol, SNMP, local resources can also Be queried over the network (see also Client 4 in Figure 5.1, page 80). If an SNMP daemon is installed (NET-SNMPD is very extensively used, and is described in Section 11.2.2 from page 187), Nagios can use it to query local resources such as processes, hard drive and interface load. The advantage of SNMP lies in the fact that it is widely used: there are corresponding services for both UNIX and Windows systems, and almost all modern network components such as routers and switches can be queried via SNMP. Even uninterruptable power supplies (USPs) and other equipment sometimes have a network connection and can provide current status information via SNMP. Apart from the standard plugin check_snmp, a generic SNMP plugin, there are various specialized plugins that concentrate on specific SNMP queries but are sometimes more simple to use. So check_ifstatus and check_ifoperstatus[5], for example, focus precisely on the status of network interfaces. If you are grappling with SNMP for the first time, you will soon come to realize that the term “readable for human beings” did not seem to be high up on the list of priorities when the protocol was defined. SNMP queries are optimized for machine processing, such as for a network monitoring tool. If you use the tool available from the vendor for its network components, SNMP will basically remain hidden to the user. But to use it with Nagios, you have to get your hands dirty and get involved with the protocol and its underlying syntax. It takes some getting used to, but it’s not really as difficult as it seems at first sight.

4. Plugins for Network Services

Every plugin that is used for host and service checks is a separate and independent program that can also be used independently of Nagios. The other way round, it is not so

easy: in order for Nagios to use an external program, it must stick to certain rules. The most important of these concerns the return status that is returned by the program. Using this, A plugin therefore does not distinguish by using the pattern "OK—Not OK", but is more differentiated. In order for it to be able to categorize a status as WARNING, it requires details of up to what measured value a certain event is regarded as OK, when it is seen as a WARNING, and when it is CRITICAL. An example: apart from the response time, a ping also returns the rate of packet loss. For a slow network connection (ISDN, DSL), a response time of 1000 milliseconds could be seen as a warning limit and 5000 milliseconds as critical, because that would mean that interactive working is no longer possible. If there is a high load on the network connection, occasional packet loss could also occur,1 so that 20 percent packet loss can be specified as a warning limit, 60 percent as the critical limit The classic reachability test in UNIX systems has always been a ping, which sends an "ICMP echo request" packet and waits for an "ICMP echo response" packet. The Nagios plugin package includes two programs that carry out this ping check: check_icmp and check_ping. Even though check_ping is used in the standard configuration, you should replace it with the more efficient check_icmp, which has been included since plugin version 1.4. Whereas check_ping calls the UNIX program /bin/ping, which is why there are always compatibility problems [4] with the existing ping version, check_icmp sends ICMP without any external help programs. check_icmp basically works more efficiently, since it does not wait for one second between individual packets, as ping Four pseudo plugins are available for testing the POP and IMAP protocols: check_ pop, check_spop, check_imap, and check_simap. They are called pseudo plugins because they are just symbolic links to the plugin check_tcp. By means of the name with which the plugin is called, this determines its intended use and correspondingly sets the required parameters, such as the standard port, whether something should be sent to the server, the expected response and how the connection should be terminated. The options are the same for all plugins, which is why we shall introduce them all together:Instead of ip, the host name or IP address of the target computer is given. For systems with several virtual environments, you will land in the default environment, and for most Web hosting providers you will then receive an error message: nagios@linux:nagios/libexec$ ./check_http -I www.swobspace.de HTTP WARNING: HTTP/1.1 404 Not Found -u url or path / --url=url or path The argument is the URL to be sent to the Web server. If the design document lies on the server to be tested, it is sufficient to enter the directory path, starting from the document root of the server: nagios@linux:nagios/libexec$ ./check_http -H linux.swobspace.net\ -u /mailinglisten/index.html HTTP OK HTTP/1.1 200 OK - 5858 bytes in 3.461 seconds If this option is not specified, the plugin asks for the document root /. -p port / --port=port This is an alternative port specification for HTTP.It is not so simple to monitor UDP ports, since there is no standard connection setup, such as the three-way-handshake for TCP, in the course of which a connection is opened, but data is not yet transferred. For a stateless protocol such as UDP there is no regulated sequence for sent and received packets. Nagios provides three plugins for monitoring databases: check_pgsql for PostgreSQL, check_mysql for MySQL, and check_oracle for Oracle. The last will not be covered in this book.13 They all have in common the fact that they can be used both locally and over the network. The latter has the advantage that the plugin in question does not have to be installed on the database server. The disadvantage is that you have to get more deeply involved with the subject of authentication, because configuring a secure local access to the database is somewhat more simple. For less critical systems, network access by the plugin can be done without a password. To do this, the user nagios is set up with its own database in the database management system to be tested, which does not contain any (important) data.There are two possibilities for monitoring uninterruptible power supplies (UPS): the Network UPS Tools support nearly all standard devices. The apcupsd daemon is specifically tailored to UPS's from the company APC The following rule generally applies: no plugin directly accesses the UPS interface. Rather they rely on a corresponding daemon that monitors the UPS and provides status information. This daemon primarily serves the purpose of shutting down the connected servers in time in case of a power failure. But it also always provides status information, which plugins can query and which can be processed by Nagios. Both the solution with the Network UPS Tools and that with apcupsd are fundamentally network-capable, that is, the daemon is always queried via TCP/IP (through a proprietary protocol, or alternatively SNMP). But you should be aware here that a power failure may affect the transmission path, so that the corresponding information might no longer even reach Nagios. Monitoring via the network therefore makes sense only if the entire network path is safeguarded properly against power failure. In the ideal scenario, the UPS is connected directly to the Nagios server. Calling the check_ups plugin is no different in this case from that for the network configuration, since even for local use it communicates via TCP/IP—but in this case, with the host localhost). The Network UPS Tools The Network UPS Tools is a manufacturer-independent package containing tools for monitoring uninterruptible power supplies. Different specific drivers take care of hardware access, so that new power supplies can be easily supported, provided their protocols are known. The remaining functionality is also spread across various programs: while the daemon upsd provides information, the program upsmon shuts down the computers supplied by the UPS in a controlled manner. It takes care both of machines connected via serial interface to the UPS and, in client/server mode, of computers supplied via the network. The homepage http://www.networkupstools.org/ lists the currently supported models and provides further information on the topic of UPS. Standard distributions already contain the software, but not always with package names that are very obvious: in SuSE and Debian they are known by the name of

nut. To query the information provided by the daemon upsd, there is the check_ups plugin from the Nagios Plugin package. It queries the status of the UPS through the network UPS Tools' own network protocol. A subproject also allows it to query the power supplies via SNMP.19 However, further development on it is not taking place at the present time.The following example tests the above defined local UPS with the name upsfw. The -T switch should ensure that the output of the temperature is given in degrees Celsius, which only partially works here: the text displayed by Nagios before the pipe sign | contains the correct details, but in the performance data after the |, the plugin version 1.4 still shows the information in degrees Fahrenheit.The check_procs plugin monitors processes according to various criteria. Usually it is used to monitor the running processes of just one single program. Here the upper and lower limits can also be specified. nmbd, for example, the name service of Samba, always runs as a daemon with two processes. A larger number of nmbd entries in the process table is always a sure sign of a problem; it is commonly encountered, especially in older Samba versions.Services such as [7] Nagios itself should only have one main process. This can be seen by the fact that its parent process has the process ID 1, marking it is a child of the init process. It was often the case, in the development phase of Nagios 2.0, that several such processes were active in parallel after a failed restart or reload, which led to undesirable side effects. You can test to see whether there really is just one single If you'd like to build your own SMS gateway using Gnokii, I wrote a "SMS Gateway How To" a few years ago. You can read the original article here (http://www.chrisburgess.com.au/sms-gateway-how-to/), or read on for the simplified steps. I am assuming you are using the same server to run both Nagios and Gnokii. Installing and Configuring Gnokii for SMS Alerts cd /usr/ports/comms/gnokii make install clea

If we'd prefer another messaging type, the Nagios mail list archives mention all sorts of methods people use.

Optimizing and Enhancing Nagios

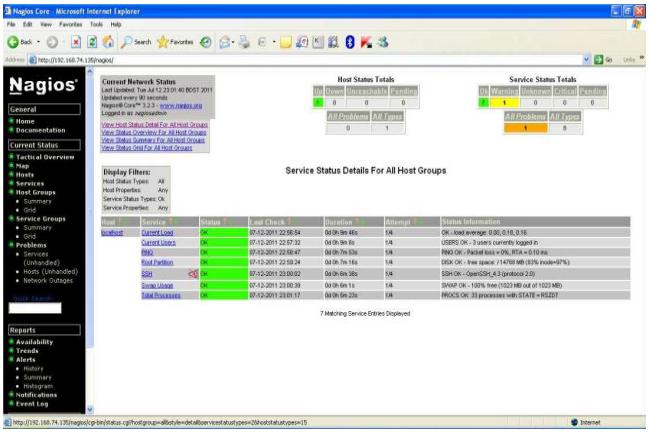

Figure 5: *Specific service status page.*

So far, we've modified the sample file called minimal.cfg (originally minimal.sample). Using the minimal sample is ideal for getting started, and by all means, you're free to use run Nagios in that configuration if you wish. At this stage however, it's worth pointing out that you can split your configurations into separate config files to help make things more manageable. Straight from the documentation, we can see how this works: NagiosQL is a web front end for simplifying the set up of the configuration files for Nagios[3]. In this guide I will only be installing NagiosQL. If we also need to install Nagios, an excellent guide to use for installing from source can be found at http://nagios.sourceforge.net/docs/3_0/quickstart-fedora.html. Generally we prefer to install using an PM package from a Yum repository, [6] but I have not yet found a repository that contains the Nagios 3 package for entOS/Red Hat. First install the MySQL and PHP packages needed to support NagiosQL. I am reinstalling, so hopefully the packages I've listed will pick up everything that is required[8]:

Figure 4: *This is web authentication feature of NagiosQL . If User login this page then write the correct user name and password.*

Host and Service Add at Nagiosql:

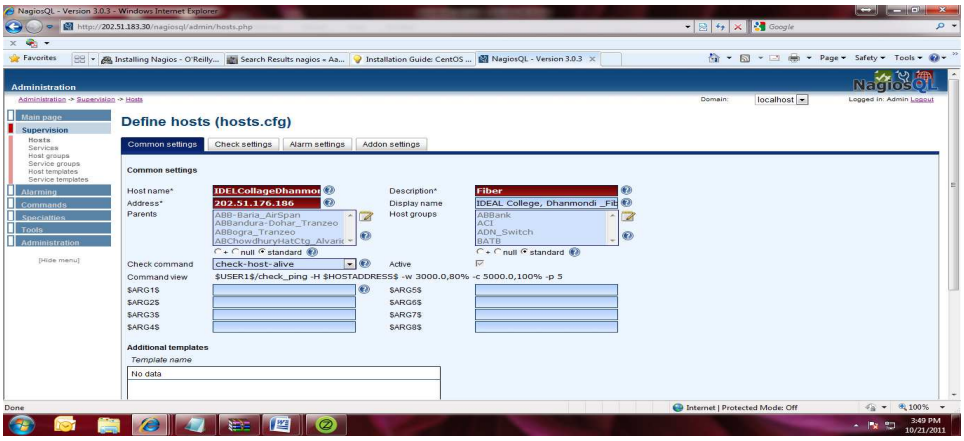

Figure 6: *Admin user add the host and service with required field like as IP Address, time, hostgroup, email.*

5. Conclusion

All in all, it is the people behind the scenes that play a crucial role in managing security in today's networks. Nagios is just a network monitoring tool. And although it is useful to further enhance the confidentality, integrity and availability of resources, it really depends on how intelligently the plugins work and how security analysts configure them to work on crucial company's servers. "Monitoring provides immediate security in a way that just doing a vulnerability assessment or dropping a firewall into a network can never provide. Monitoring provides dynamic security in a way that yet another security product can never provide. And, as security products are added into a network - firewalls, IDSs, specialized security devices - monitoring only gets better"35. This is because faster detection and response can be achieved through active network monitoring of events from a number of security components. It is always crucial for security [9] analysts to maintain monitoring 24x7 with an 'active' mind. In other words, they must work intelligently. For instance, if there are no new alerts generated for some time, security analysts should always make it a habit to check around to make sure things are still moving fine. With GNU license open source software like Nagios,[6] it is also advisable for one to subscribe to relevant mailing lists to keep in touch with new bug fixes and patch to the latest version of software applications used. Currently, Nagios is at version 1.1 and is expected to roll into 2.0 by summer of 2003 with further enhancements including passive host checks, which enable the central monitoring server to be able to receive not only remote services results but host status as well.For more interests in the next version, one can view http://www.Nagios.org/upcoming.php. "It takes constant monitoring. It's not one tool over another; it's the mind-set of the staff who review our systems, read information, put in proper patches and do proper testing.

References

[1] Lamsal, P. "Management of the Next Generation IP Core Network." 16th April 1999. URL: http://www.tml.hut.fi/Opinnot/Tik-110.551/1999/papers/12ManagementOfIPngCore/ipcore.html

[2] Winkler, Ira. "Ounce of Prevention." November 1999. URL: http://www.infosecuritymag.com/articles/1999/winkler.shtml

[3] Walker, L. "The View From Symantec's Security Central." 9th January 2003. URL: http://www.washingtonpost.com/wp-dyn/articles/A28625-2003Jan8.html

[4] Messmer, E. and Pappalardo, D. "A Year After Meltdown: No Silver Bullet for DoS." 2nd May 2001. URL: http://www.nwfusion.com/news/2001/0205ddos.html FA27 2F94 998D FDB5 DE3D F8B5 06E4 A169 4E46

[5] CERT/CC "CERT Advisory CA-2002-03 Multiple Vulnerabilities in Implementations of the Simple Network Management Protocol (SNMP)." 4th August 2003. URL: http://www.cert.org/advisories/CA-2002-03.html

[6] Kamthan, P. "CGI Security: Better Safe Than Sorry." 19th September 1999. URL: http://tech.irt.org/articles/js184/[28] - [30] Refer to [15].

[7] Polombo, D. "Prelude HOWTO." 16th September 2002. URL: http://www.prelude-ids.org/article.php3?id_article=6[32] Refer to [11].

[8] Habib, A., Hefeeda, M. M. and Bhargava, B. K. "Detecting Service Violations and DoS Attacks." 2002. URL: http://216.239.39.104/search?q=cache:m4uV_rBx9hIJ:www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/12.pdf+detecting+service+violations+and+dos+attacks&hl=en&ie=UTF-8