Centralized Linux Security Server

Unisys Scholars Program – Final Report

By: Lucas Machado

May 3rd, 2004

Overview

- Introduction
- Linux System Logging
- Windows System Logging
- Windows/Linux Bridge
- Logwatch
- Security
- Server/Client Status Monitoring
- Conclusion

Introduction

Infrastructure Configuration

Clients on the network send security information to the Linux server. Here, the system administrator monitors the activity on all of the client machines.

System Logging

- It can be important to know what is happening on the machines both when you are and aren't there.
- Why?
  - Finding/Debugging errors on the system
  - Keep track of who is doing what
  - Security

Linux System Logging
Modifying syslog.conf

- Location: /etc/syslog.conf
- Syslog Facilities:
  - authpriv — login authentication
  - cron — cron subsystem
  - daemon — system server processes
  - user — user processes
  - kern — linux kernel
  - lpr — spooling subsystem
  - mail — mail subsystem
  - news — news subsystem

Linux System Logging
Modifying syslog.conf

- Severity Levels:
  - emerg — system unusable
  - alert — take immediate action
  - crit — critical condition
  - err — error message
  - warn — warning message
  - notice — normal but significant condition
  - info — informational
  - debug — debug message
  - none — N/A
- General Syntax:

  facility.level *destination*
  - Sends logs for "*facility*" at severity level "*level*" or higher to "*destination*"

Linux System Logging
Modifying syslog.conf

- Defining Multiple Facilities & Severity Levels:

  facility1, facility2.level1

  - Logs for "*facility1*" & "*facility2*" at severity level "*level1*" and higher are sent to the same destination

  facility1.level1;facility2.level2

  - Logs for "*facility1*" at "*level1*" & logs for "*facility2*" at "*level2*" are sent to the same destination

- An "*" can be used as a wildcard for both facility and level

Linux System Logging
Modifying syslog.conf

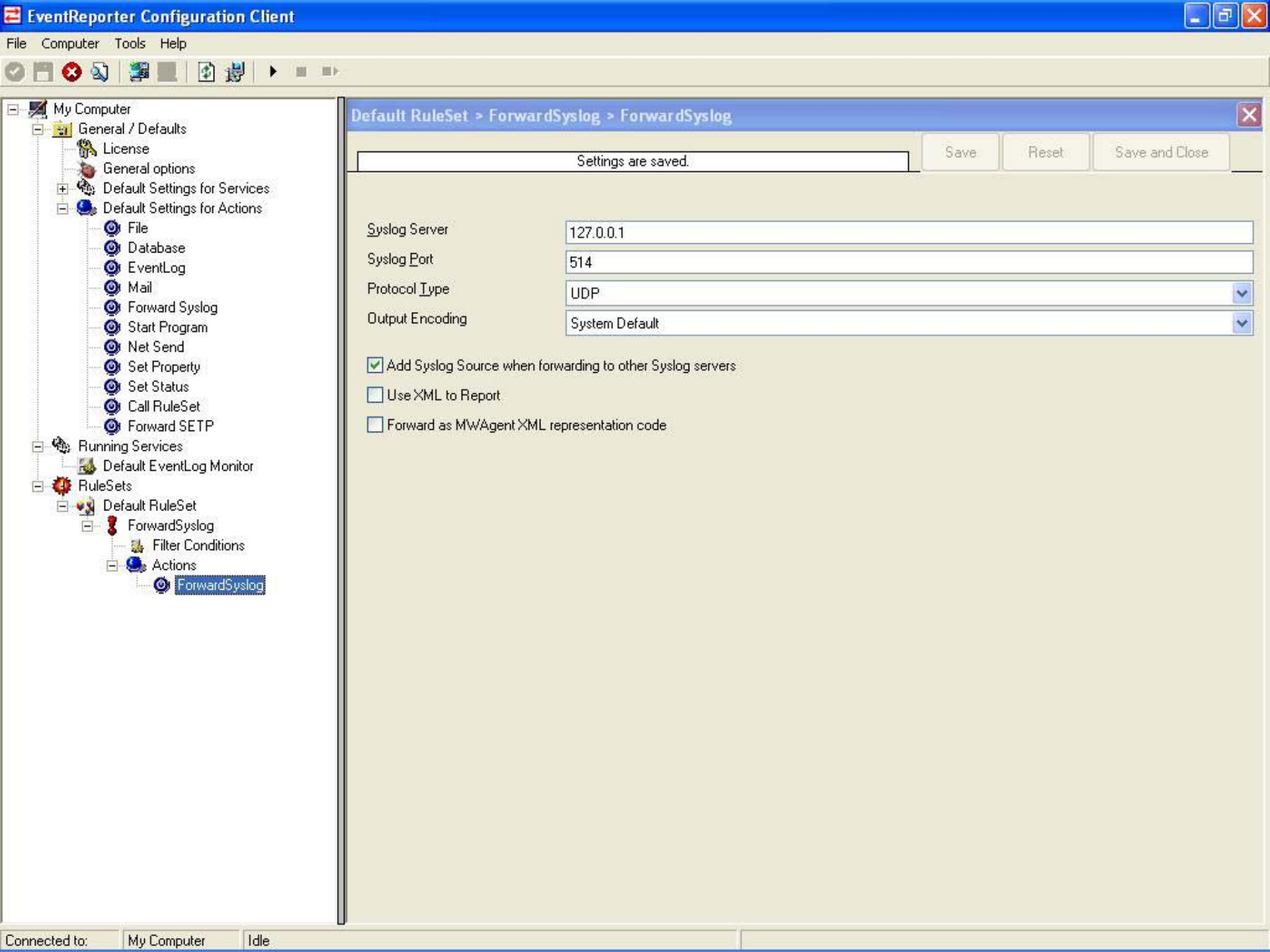
- Syntax Formats:
  - *facility.level* logs *level* and higher messages
  - *facility.=level* logs only *level* messages
  - *facility.!level* logs *level* and lower messages
  - *facility.!=level* logs all but *level* messages
- Destination:
  - A filename
  - A device (i.e. terminal)
  - A list of one or more users (comma-separated)
  - An "*" sends messages immediately to all logged-in users
  - @*hostname*: messages are sent to the syslog facility on the specified host for processing

Windows System Logging
Logging Using gpedit.msc

- Telling Windows what to audit:
  - Start→Run→gpedit.msc
  - Local Computer Policy→Computer Config→ Windows Settings→Security Settings→Local Policies→Audit Policy
  - Policies:
    - Account management
    - Logon events
    - Object access
    - Policy change
    - Process tracking
    - System events

Windows/Linux Bridge

- EventReporter
  - Sends Windows 2k logs to server running syslog daemon (includes many other features)
  - Free to try/$59 to buy
  - Easy to use interface; Configuring to send logs to server takes less than 5 minutes


EventReporter Configuration Client
File
Computer
Tools
Help
My Computer
General / Defaults
License
General options
Default Settings for Services
Default Settings for Actions
File
Database
EventLog
Mail
Forward Syslog
Start Program
Net Send
Set Property
Set Status
Call RuleSet
Forward SETP
Running Services
Default EventLog Monitor
RuleSets
Default RuleSet
ForwardSyslog
Filter Conditions
Actions
ForwardSyslog
Default RuleSet > ForwardSyslog > ForwardSyslog
Settings are saved.
Save
Reset
Save and Close
Syslog Server
127.0.0.1
Syslog Port
514
Protocol Type
UDP
Output Encoding
System Default
Add Syslog Source when forwarding to other Syslog servers
Use XML to Report
Forward as MWAgent XML representation code
Connected to:
My Computer
Idle

Logfile: /var/log/messages

```
Apr 16 09:48:37 big modprobe: modprobe: Can't locate module char-major-188
Apr 16 09:48:37 big last message repeated 15 times
Apr 16 09:48:37 big cups: cupsd startup succeeded
Apr 16 09:48:37 big cups: cupsd startup succeeded
Apr 16 09:48:38 big anacron: anacron startup succeeded
Apr 16 09:48:38 big anacron: anacron startup succeeded
Apr 16 09:48:38 big atd: atd startup succeeded
Apr 16 09:48:38 big atd: atd startup succeeded
Apr 16 09:50:19 big sshd(pam_unix)[734]: authentication failure; logname= uid=0 euid=0 tty=NODEVss
h ruser= rhost=unisys.engr.rutgers.edu  user=lmachado
Apr 16 09:50:28 big sshd(pam_unix)[736]: session opened for user lmachado by (uid=500)
Apr 16 09:51:17 node01 sshd(pam_unix)[11619]: session opened for user root by (uid=0)
Apr 16 09:52:49 node01 sshd(pam_unix)[11619]: session closed for user root
Apr 16 09:52:50 big sshd(pam_unix)[736]: session closed for user lmachado
Apr 16 10:55:47 node01 sshd(pam_unix)[11680]: session opened for user root by (uid=0)
Apr 16 11:06:17 big sshd(pam_unix)[782]: authentication failure; logname= uid=0 euid=0 tty=NODEVss
h ruser= rhost=unisys.engr.rutgers.edu  user=lmachado
Apr 16 11:06:23 big sshd(pam_unix)[784]: session opened for user lmachado by (uid=500)
Apr 16 11:06:32 node01 sshd(pam_unix)[11742]: session opened for user root by (uid=0)
Apr 16 12:14:39 node01 sshd(pam_unix)[11680]: session closed for user root
Apr 16 12:14:42 big sshd(pam_unix)[784]: session closed for user lmachado
Apr 16 12:14:42 node01 sshd(pam_unix)[11742]: session closed for user root
Apr 16 12:36:44 node01 sshd(pam_unix)[18567]: session opened for user root by (uid=0)
Apr 16 12:39:44 big sshd(pam_unix)[822]: session opened for user root by (uid=0)
Apr 16 12:39:52 big sshd(pam_unix)[822]: session closed for user root
Apr 16 12:40:14 big sshd(pam_unix)[866]: session opened for user root by (uid=0)
Apr 16 12:42:20 big sshd(pam_unix)[866]: session closed for user root
Apr 16 12:42:35 node01 shutdown: shutting down for system reboot
Apr 16 12:42:35 node01 init: Switching to runlevel: 6
Apr 16 12:42:36 node01 atd: atd shutdown succeeded
Apr 16 12:42:36 node01 rc: Stopping keytable:  succeeded
Apr 16 12:42:36 node01 cannaserver[699]: irw_killserver:cannaserver end
```

Logwatch

- Provides overview of logfiles
  - Configuration information: /etc/log.d
  - Modifying Default Settings:
    - Command line options
    - /etc/log.d/logwatch.conf
  - Options:
    - Detail level
    - Process specific logfile logs
    - Process specific service logs
    - Scan/don't scan archives
    - Date range: yesterday, today, all
    - Where to send logwatch output (print, mail, file, hostname)

LogWatch

```
[root@node01 log]# logwatch --detail low --print

 ################### LogWatch 4.3.1 (01/13/03) ####################
       Processing Initiated: Thu Apr 29 18:30:29 2004
       Date Range Processed: yesterday
     Detail Level of Output: 0
          Logfiles for Host: node01.rutgers.edu
 ##################################################################

 --------------------- pam_unix Begin ------------------------

sshd:
   Authentication Failures:
      root (unisys.engr.rutgers.edu ): 1 Time(s)


 ---------------------- pam_unix End -------------------------


 --------------------- sendmail Begin ------------------------



12 messages returned after 5 days
 ---------------------- sendmail End -------------------------


 --------------------- SSHD Begin ------------------------


Failed logins from these:
   root/password from 192.168.6.1: 1 Time(s)

Users logging in through sshd:
   root logged in from node01.rutgers.edu (192.168.6.3) using password: 4 Time(s)
   root logged in from unisys.engr.rutgers.edu (192.168.6.1) using password: 2 Time(s)

 ---------------------- SSHD End -------------------------


 ###################### LogWatch End #########################
```

Security

- A good way to check if a client has been breached is to check whether key files that should not be modified have in fact been modified. Examples:
  - /etc/groups
  - /etc/passwd
  - /sbin
  - /var/logs
- How to check? Tripwire

Security
Tripwire

- Monitors file attributes that should not change:
  - Size
  - File permissions/Ownership
  - Last access time
  - Last modification time
  - Binary Signatures

Security
Tripwire

- Needs two databases to compare:
  - 1) Original database containing all the information of files being monitored
  - 2) Database with current file information
- Must make sure original database cannot be tampered with. Solution: Read-Only media.
  - Great system for a few or less machines, however, with more machines this system is not sufficient:
    - Write-once hardware for each machine is expensive.
    - CD's can be damaged, burning them after each change is not practical, and checking each machine one-by-one wastes time/money.
- There must be a better solution…..

Security
LANTrip

- Requires a simple system: 133-MHz w/64MB RAM. Hard Drive size depends on how many client machines.
- Steps:
  - LANTrip stores a copy of the Tripwire installation and the original database on the security server.
  - At a certain time (or at random), a copy of the installation & database are copied from the host to a random directory on each client.
  - Tripwire is executed and the results are recorded.
  - The results are sent back to the server.
  - The Tripwire installation & database are deleted from the client machine

Security
Configuration on Server

- Create a tripwire user who will run Tripwire on the clients
- <lantrip_directory>/data/hosts.conf:
  - Determines what hosts to check for integrity.
  - Format:

    Hostname *Remote User* *OS_Architecture*
  - Example:

unisys01	tripwire	Linux_x86
unisys02	tripwire	OpenBSD_Alpha
unisys03	tripwire	Solaris_SPARC

Security
Configuration on Server

- <lantrip_directory>/data/rand_dir.conf:
  - Includes directories where LANTrip should copy the installation & database to (these directories should exist on all machines and must be writable by the tripwire user).
  - Example:

    /var/tmp

    /tmp
  - The more directories the better. Increased Randomness!!
- < lantrip_directory>/lantrip_report.pl:
  - Edit first couple lines of the script to send report emails to proper email address.
- Add a crontab entry to run LANTrip at a specific or random time (this is not necessary, as you can run LANTrip manually)
- Sendmail must also be installed

Security
Configuration on Clients

- Run: <lantrip_directory>/lantrip_client.pl
  - Prompts for location of Tripwire source and sshd configuration file
  - Automatically edits configures Tripwire to be able to be run from any directory (this will be used when the tripwire user connects remotely to run Tripwire)
  - Modifies sshd_config and sets RSAHostsAllow to “Yes”
  - Compiles Tripwire
- Copy Tripwire installation & database to security server (this only needs to be done once for each architecture)
- Modify <tripwire_directory>/configs/tw.config to tell Tripwire what files/folders to monitor. For assistance, refer to the sample tw.config file.
- Enable password-less access to the tripwire user

Security
Running & Updating LANTrip

- Updating:
  - To update Tripwire databases on the server, run: <server_lantrip_directory>/lantrip.pl –i
- Running:
  - To run LANTrip manually, on the server, run: <server_lantrip_directory>/run_lantrip.pl
  - To send the LANTrip report manually, on the server, run: <server_lantrip_directory>/lantrip_report.pl

LANTrip Report

root@big:/opt/tripwire/tw_ASR_1.3.1_src/src

```
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
### Phase 4:   Searching for inconsistencies
###
###                    Total files scanned:            67119
###                          Files added:              1
###                          Files deleted:            1
###                          Files changed:            4
###
###                    Total file violations:          6
###
added:   -rw------- tripwire  9284338 Apr 27 14:59:22 2004 /opt/tripwire/tw_ASR_
1.3.1_src/src/data/databases/tw.db_big.rutgers.edu
deleted: -rw------- tripwire  2387968 Apr 27 14:49:53 2004 /opt/tripwire/tw_ASR_
1.3.1_src/src/databases/tw.db_big.rutgers.edu
changed: -rw------- root         8272 Apr 29 19:46:11 2004 /root/.viminfo
changed: -rw------- root         9970 Apr 29 19:52:06 2004 /root/.bash_history
changed: -rw-r--r-- root          551 Apr 29 19:12:18 2004 /etc/group
changed: -r--r----- root           32 Apr 29 19:52:10 2004 /etc/cups/certs/0
### Phase 5:   Generating observed/expected pairs for changed files
###
### Attr        Observed (what it is)         Expected (what it should be)
### =========== ============================= =============================
/root/.viminfo
        st_ino: 87748                         90411
       st_size: 8272                          8182
      st_mtime: Thu Apr 29 19:46:11 2004      Tue Apr 27 14:48:45 2004
      st_ctime: Thu Apr 29 19:46:11 2004      Tue Apr 27 14:48:45 2004
    md5 (sig1): 0q7:gZ4BB6Es67bI6UKVHW        3DMrapGVr5ksxKMnvqMmn9
 snefru (sig2): 2Bvp2pE4xONCBQmZlys:o1        1ybgABJGmegUAJs7X68Oyc

/root/.bash_history
       st_size: 9970                          8738
      st_mtime: Thu Apr 29 19:52:06 2004      Tue Apr 27 14:36:07 2004
      st_ctime: Thu Apr 29 19:52:06 2004      Tue Apr 27 14:36:07 2004
    md5 (sig1): OQXFnFrgDTCzu5BZPrwSvg        2mMAqQrEyB7TYJqkytBCAG
 snefru (sig2): 1Fx:yN5u:qXaqNq16P1Z1f        1ZU.7OUKf91SuB17sTros3

/etc/group
        st_ino: 33745                         33713
      st_mtime: Thu Apr 29 19:12:18 2004      Thu Apr 15 19:10:09 2004
      st_ctime: Thu Apr 29 19:12:18 2004      Thu Apr 15 19:10:09 2004
    md5 (sig1): 3cC5xErqw1LgVE.8YipzTY        3byL5maAB:Gx2tZ6Q8vEuH
 snefru (sig2): 2f6jDSYStfIN:5gbFBkMuT        OGmaCbv98Na8SFaAUltqvi
```

Server/Client Status Monitoring

- Often times you want to monitor:
  - Services running on the server & clients, for example:
    - sshd
    - ftp
    - http
  - Important attributes on server & client machines:
    - Disk/memory usage
    - Uptime/downtime
    - System load
    - Running processes
- Great tool for status monitoring: Nagios

Server/Client Status Monitoring

- Nagios
  - Capable of monitoring all of the most crucial information/services
  - Immediate notification of problems (ie. E-mail)
  - Ability to define event handlers
  - Manual checking of services/client attributes
  - Capable of scheduling downtime of services/clients
  - Web interface! (Requires Web server)
  - Open source – **FREE!!**
- It takes time to learn how to properly configure Nagios and use it to it's full potential, however, it is definitely worth it.

Server/Client Status Monitoring

- Configuring Nagios:
  - Create a nagios user who will run Nagios related scripts.
  - You must modify most of the configuration files. Some of the most important are:
    - <nagios_directory>/etc/hosts.cfg
    - <nagios_directory>/etc/services.cfg
    - <nagios_directory>/etc/contacts.cfg
    - <nagios_directory>/etc/contactgroups.cfg
    - <nagios_directory>/etc/hostgroups.cfg
  - Set up web server (with authentication) & Nagios web interface
  - Start Nagios: /etc/rc.d/init.d/nagios start
- Scripts (located in <nagios_directory>/libexec/) can also be run manually
- Detailed tutorials can be found all over the internet

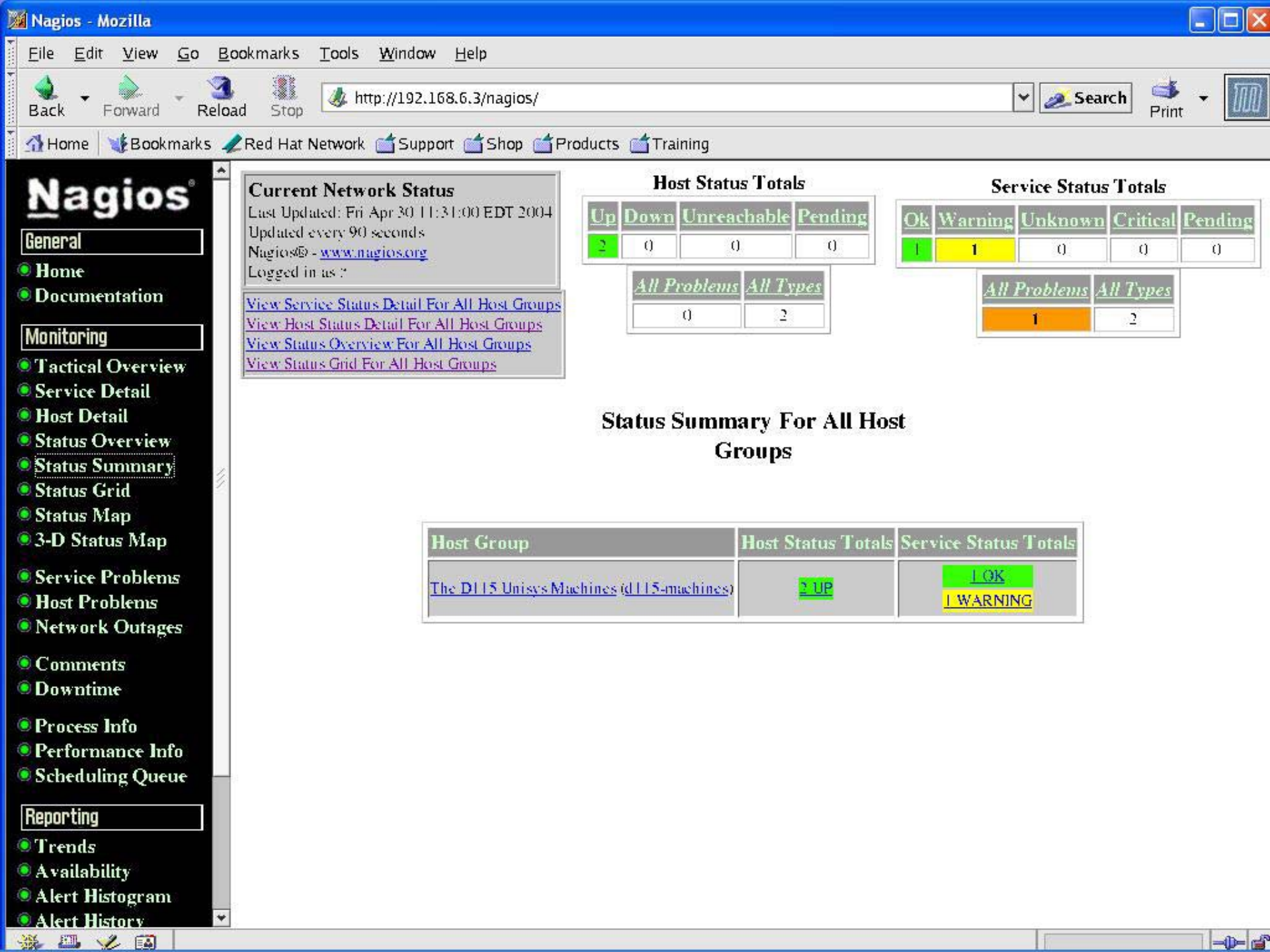
Nagios – contacts.cfg

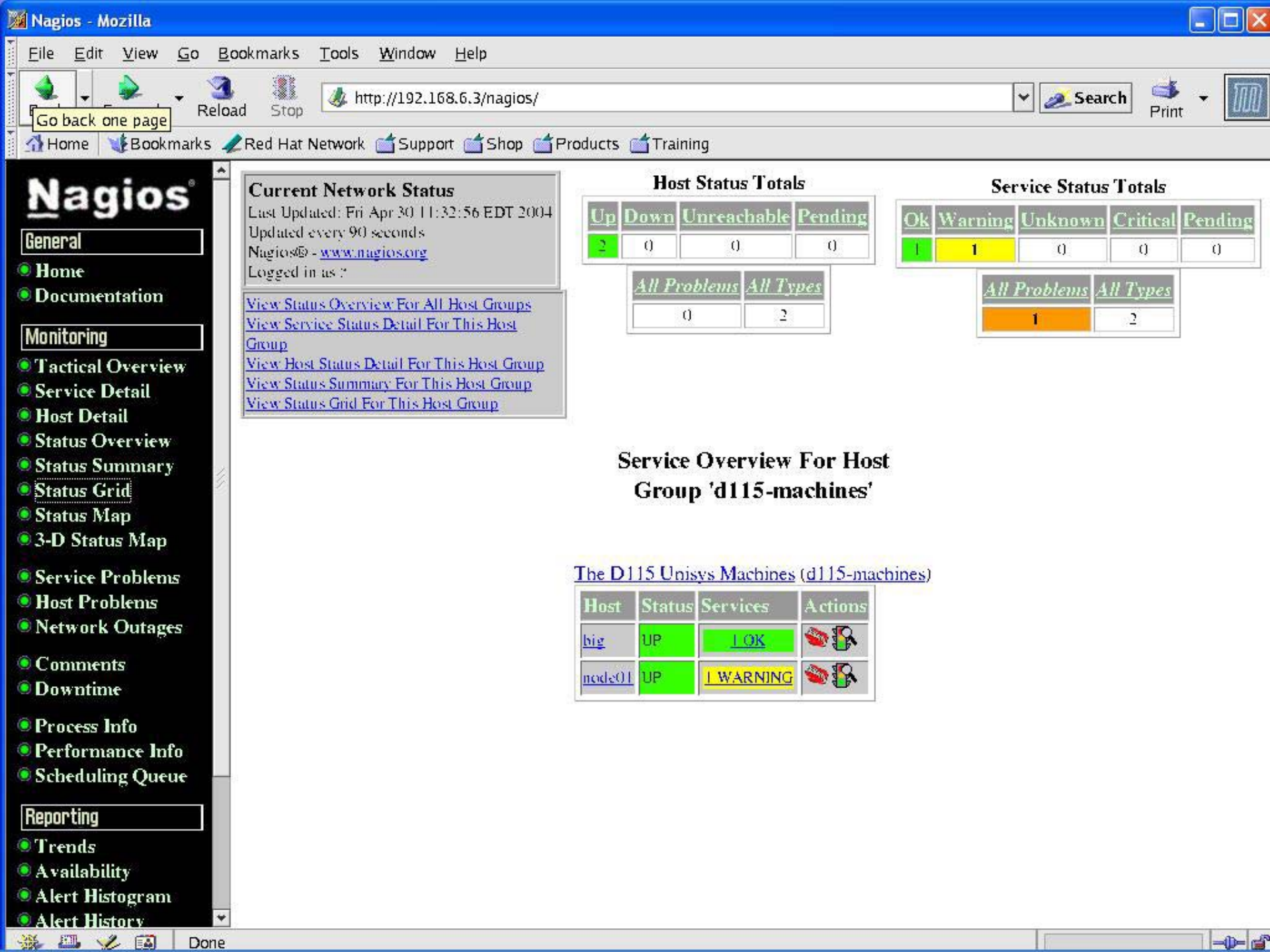
root@node01:/usr/local/nagios/etc

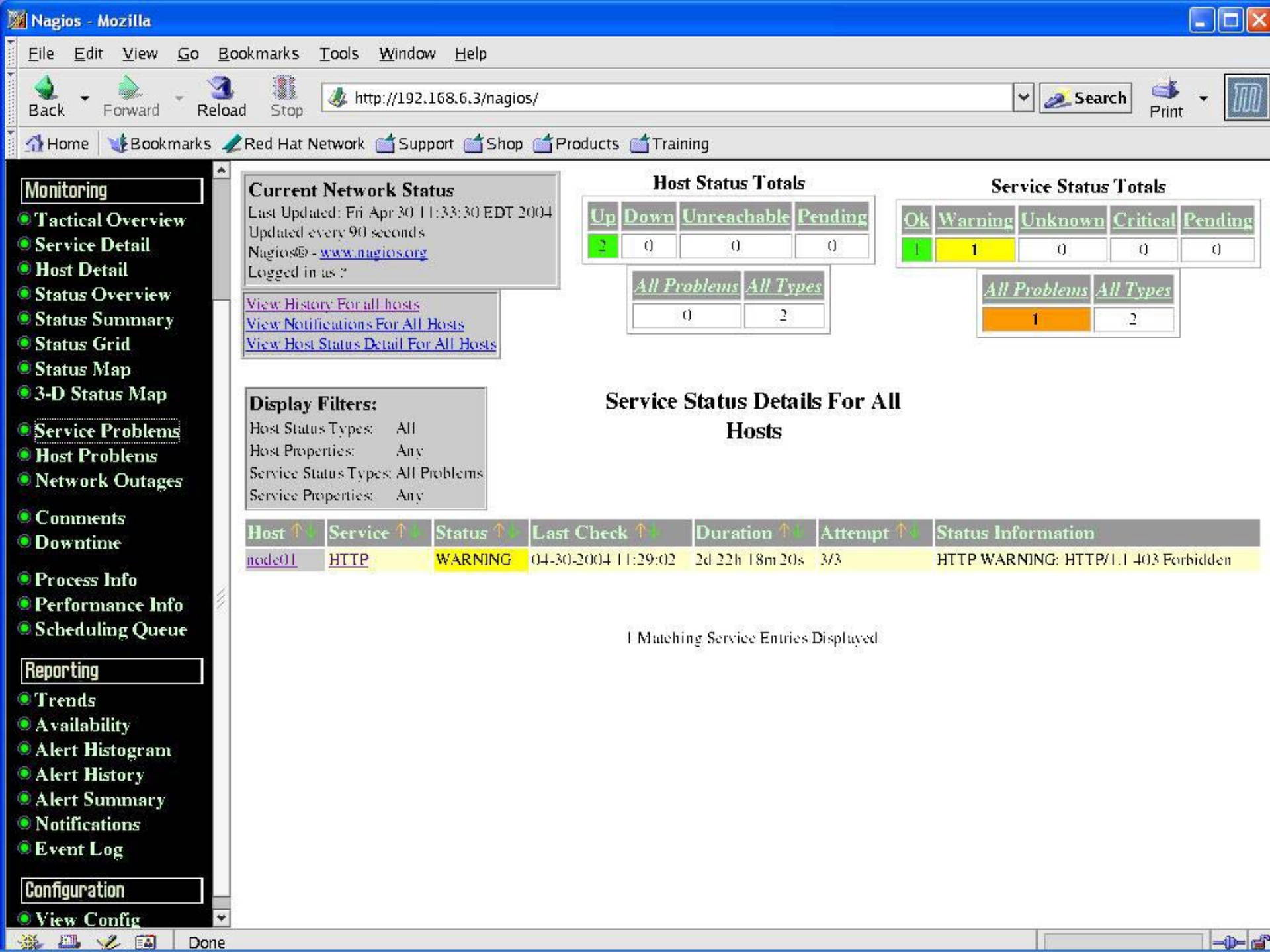
```
define contact{
 contact_name                    lmachado
 alias                           big
 service_notification_period     24x7
 host_notification_period        24x7
 service_notification_options    w,u,c,r
 host_notification_options       d,u,r
 service_notification_commands   notify-by-email
 host_notification_commands      host-notify-by-email
 email                           lmachado@eden.rutgers.edu
 }

define contact{
 contact_name                    server_root
 alias                           server-admin
 service_notification_period     24x7
 host_notification_period        24x7
 service_notification_options    w,u,c,r
 host_notification_options       d,u,r
 service_notification_commands   notify-by-email
 host_notification_commands      host-notify-by-email
 email                           root@node01
 }
```

Nagios – services.cfg

```
# Service definition
define service{
 use     generic-service

 host_name    node01
 service_description  HTTP
 is_volatile    0
 check_period    24x7
 max_check_attempts  3
 normal_check_interval  5
 retry_check_interval  1
 contact_groups    Sec-Server-Admin
 notification_interval  120
 notification_period  24x7
 notification_options  w,u,c,r
 check_command    check_http
 }


# Service definition
define service{
 use     generic-service

 host_name    big
 service_description  PING
 is_volatile    0
 check_period    24x7
 max_check_attempts  3
 normal_check_interval  5
 retry_check_interval  1
 contact_groups    Sec-Server-Admin
 notification_interval  120
 notification_period  24x7
 notification_options  c,r
 check_command    check_ping!100.0,20%!500.0,60%
 }
"services.cfg" 53L, 1163C
```


Nagios - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://192.168.6.3/nagios/
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Status Overview
Status Summary
Status Grid
Status Map
3-D Status Map
Service Problems
Host Problems
Network Outages
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Current Network Status
Last Updated: Fri Apr 30 11:31:00 EDT 2004
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as ?
View Service Status Detail For All Host Groups
View Host Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
1 1 0 0 0
All Problems All Types
1 2
Status Summary For All Host Groups
Host Group Host Status Totals Service Status Totals
The D115 Unisys Machines (d115-machines) 2 UP 1 OK 1 WARNING


Nagios - Mozilla
File Edit View Go Bookmarks Tools Window Help
Go back one page
Reload
Stop
http://192.168.6.3/nagios/
Search
Print
Home
Bookmarks
Red Hat Network
Support
Shop
Products
Training
Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Status Overview
Status Summary
Status Grid
Status Map
3-D Status Map
Service Problems
Host Problems
Network Outages
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Current Network Status
Last Updated: Fri Apr 30 11:32:56 EDT 2004
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *
View Status Overview For All Host Groups
View Service Status Detail For This Host Group
View Host Status Detail For This Host Group
View Status Summary For This Host Group
View Status Grid For This Host Group
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
1 1 0 0 0
All Problems All Types
1 2
Service Overview For Host Group 'd115-machines'
The D115 Unisys Machines (d115-machines)
Host Status Services Actions
big UP 1 OK
node01 UP 1 WARNING
Done


Nagios - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://192.168.6.3/nagios/
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Monitoring
Tactical Overview
Service Detail
Host Detail
Status Overview
Status Summary
Status Grid
Status Map
3-D Status Map
Service Problems
Host Problems
Network Outages
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log
Configuration
View Config
Current Network Status
Last Updated: Fri Apr 30 11:33:30 EDT 2004
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as ?
View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts
Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2
Service Status Totals
Ok Warning Unknown Critical Pending
1 1 0 0 0
All Problems All Types
1 2
Display Filters:
Host Status Types: All
Host Properties: Any
Service Status Types: All Problems
Service Properties: Any
Service Status Details For All Hosts
Host Service Status Last Check Duration Attempt Status Information
node01 HTTP WARNING 04-30-2004 11:29:02 2d 22h 18m 20s 3/3 HTTP WARNING: HTTP/1.1 403 Forbidden
1 Matching Service Entries Displayed
Done

Conclusion

- This project was aimed at putting together a security/monitoring server utilizing many popular open-source software. This type of centralized server is perfect for a network that contains many crucial servers (i.e. University, Business Corporations) or even for a classroom network.
- With this type of server you will be able to:
  - Easily find any errors that may be plaguing a client machine
  - Find out if someone is attempting to do something they shouldn't be doing
  - Detect unauthorized access to your client machines without letting the intruder know you are doing so
  - Monitor key services and clients so that the problems may be fixed quickly and efficiently
- Some of the tools used in this project would be a great addition to the servers used in the School of Engineering network.

References

- http://www.linux-mag.com/2000-10/security_01.html
- http://www.linux-mag.com/2000-09/guru_01.html
- http://www.linux-mag.com/2000-10/guru_01.html
- http://www.eventreporter.com/en/
- "Using Tripwire on a Network with LANTrip" in Sysadmin magazine, Oct. 2003
- http://www.nagios.org/
- http://www.onlamp.com/pub/a/onlamp/2002/09/05/nagios.html

Acknowledgments

- Through the Unisys Scholars Program I was able to learn a great deal about Linux, networking, and many system administration skills while helping the students in the course at the same time.
- Thanks to:
  - Dr. David Gardiner
  - Dr. Doyle Knight
  - Dr. Alexei Kotelnikov