Monitoring a Linux Mail Server

Mike Weber

mweber@spidertools.com
Various Methods to Monitor Mail Server

- **Public Ports**
  - SMTP on Port 25
  - POP3 on Port 995
  - IMAPS on Port 993

- **SNMP**
  - Amavis on Port 10024
  - Reinjection Port on 10025
  - Spamassassin on Port 783

- **NRPE**
  - Virus Signatures
  - Virus Activity
  - Virus Numbers

- **Perl Plugin**
  - Email Delivery
  - Verify Read Email Headers
  - Verify Read Headers and Content
Various Methods to Monitor Mail Server

SSH
- Amavis on Port 10024
- Reinjection Port on 10025
- Spamassassin on Port 783
- Virus Signatures
- Virus Activity
- Virus Numbers
- Email Delivery
- Verify Read Email Headers
- Verify Read Headers and Content
Monitor Public Mail Ports

- **SMTP Port 25**
  - Port Status
  - Response Times
  - Graph Response Times

- **IMAPS Port 993**
  - Port Status
  - Response Times
  - Graph Response Times

- **POP3S Port 995**
  - Port Status
  - Response Times
  - Graph Response Times
Monitor Email Delivery

**Host:** mail  **Service:** SMTP Port 25

**4 Hours** 10.08.11 6:45 - 10.08.11 10:45

![Graph showing response time for mail/SMTP_Port_25]

**Host:** mail  **Service:** SMTP Port 25

**25 Hours** 09.08.11 9:45 - 10.08.11 10:45
Monitor Public Mail Ports

Port Status – Connection Time

```plaintext
define service{
    use generic-service
generic-service
    hostgroup_name debian-servers
debian-servers
    service_description Postfix Port
Postfix Port
    check_command check_tcp!25 -w 03 -c 05
check_tcp!25 -w 03 -c 05
}
define service{
    use generic-service
generic-service
    hostgroup_name debian-servers
debian-servers
    service_description Secure IMAPS
Secure IMAPS
    check_command check_tcp!993 -w 03 -c 06
check_tcp!993 -w 03 -c 06
}
define service{
    use generic-service
generic-service
    host_name db
db
    service_description POP3S Port 995
POP3S Port 995
    check_command check_tcp!995 -w 03 -c 06
check_tcp!995 -w 03 -c 06
}
```
### Monitor Public Mail Ports

#### Nagios Core Config Manager

**Service Management**

<table>
<thead>
<tr>
<th>Common Settings</th>
<th>Check Settings</th>
<th>Alert Settings</th>
<th>Misc Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config Name*</td>
<td>192.168.5.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hosts*</td>
<td>*</td>
<td>192.168.5.45</td>
<td>localhost</td>
</tr>
<tr>
<td>Host groups*</td>
<td></td>
<td></td>
<td>linux-servers</td>
</tr>
<tr>
<td>Service description*</td>
<td>SMTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check command*</td>
<td>check_tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command view</td>
<td>$USER1$/check_tcp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ARG1$</td>
<td>25</td>
<td>$ARG5$</td>
<td></td>
</tr>
<tr>
<td>$ARG2$</td>
<td>-t 6</td>
<td>$ARG6$</td>
<td></td>
</tr>
<tr>
<td>$ARG3$</td>
<td></td>
<td>$ARG7$</td>
<td></td>
</tr>
<tr>
<td>$ARG4$</td>
<td></td>
<td>$ARG8$</td>
<td></td>
</tr>
</tbody>
</table>

**Additional templates**

*Template Name*

xiwizard_smtp_service
Monitoring Content Filter, Reinjection and Spamassassin with SNMP

- Content Filter Port 10024
- Reinjection Port 10025
- Spamassassin Port 783
Monitoring Content Filter and Reinjection

Mail Client

content_filter
port 10024

Postfix content_filter Process

qmgr

amavisd-new

SpamAssassin

ClamAv

Reinjection
port 10025
Creating Bash Scripts for SNMP

Command Definition
define command{
    command_name   check_amavis
    command_line   $USER1$/check_amavis
}

Service Definition
define service{
    use                        generic-service
    host_name                  mail
    service_description        Amavis: Virus Protection
    check_command              check_amavis
}

Script Using SNMP
#!/bin/bash
amavis=$(snmpnetstat -v 2c 192.168.5.191 -c public -Ca | grep 10024 |wc -l)
if (($amavis >= 1 ))
then echo "Amavis is Running"
    stateid=0
else
    echo "Danger: Amavis is NOT running, no virus protection"
    stateid=2
fi
exit $stateid
Creating Bash Scripts for SNMP

```
snmpnetstat -v 2c 192.168.5.45 -c public -Ca
```

Active Internet (tcp) Connections (including servers)

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Remote Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>*.ssh</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.smtp</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.pop3</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.sunrpc</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.imap</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.imaps</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.pop3s</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.5666</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>*.38922</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>localhost.ipp</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>localhost.783</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>localhost.10025</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.37932</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.39143</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.44947</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.46752</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.50184</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.55465</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.55674</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.59800</td>
<td>CLOSEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>192.168.5.4.http</td>
<td>TIMEWAIT</td>
</tr>
<tr>
<td>tcp</td>
<td>192.168.5.45.smtp</td>
<td>a69-192-195-51.d.https</td>
<td>CLOSEWAIT</td>
</tr>
</tbody>
</table>
Checking Amavis - SNMP

**Install Script**
Install any script you want to use in the /usr/local/nagios/libexec with the correct permissions.

**Create Command**
Whenever you use your own script, you will need to create a command to access the script.

**Create Check**
Once the command has been created, you will be able to use it for any hosts.
Checking Spamassassin - SNMP

- **Install Script**
  Install any script you want to use in the /usr/local/nagios/libexec with the correct permissions

- **Create Command**
  Whenever you use your own script, you will need to create a command to access the script.

- **Create Check**
  Once the command has been created you will be able to use it for any hosts.
Checking Spamassassin - SNMP

Nagios Core Config Manager

Service Management

Common Settings

Config Name: check_spamassassin

Hosts:
- 192.168.5.45
- localhost

Host groups:
- linux-servers

Service description: check_spamassassin

Display name:

Active:

Check command: check_spamassassin

Command view:

$USER1$/check_spamassassin

Additional templates

Template Name:
- generic-service
Monitor Virus Activity with NRPE

- Virus Signatures
- Quarantine Status
- Number ofViruses Captured
You will need to install xinetd and make sure you have a file in `/etc/xinetd.d` called `nrpe` on the client and it looks like this:

```
# default: off
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
  flags             = REUSE
  type              = UNLISTED
  port              = 5666
  socket_type       = stream
  wait              = no
  user              = nagios
  group             = nagios
  server            = /usr/sbin/nrpe
  server_args       = -c /usr/local/nagios/etc/nrpe.cfg --inetd
  log_on_failure    += USERID
  disable           = no
  only_from         = 127.0.0.1 192.168.5.50
}
```
Checking Virus Signatures - NRPE

define command{
    command_name   check_nrpe
    command_line   $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

define service{
    use                      generic-service
    host_name                mail
    service_description      Virus Signatures
    check_command            check_nrpe!check_signatures
}

command[check_signatures]=/usr/local/nagios/libexec/check_signatures

Bash shell script

#!/bin/bash

dbase=$(tail -300 /var/log/clamav/clamd.log| grep "Database correctly reloaded"|wc -l)
sigs=$(tail -300 /var/log/clamav/clamd.log| grep "Database correctly reloaded"| awk -F"( '{print $2}'|tail -1)
dbdate=$(tail -300 /var/log/clamav/clamd.log| grep "Database correctly reloaded"| awk -F' ' '{print $1,$2,$3}'|tail -1)
if [ "$dbase" -eq 0 ]
    then
    echo "Virus Signatures Out of Date"
    stateid=2
else
    echo "Virus Database Updated $dbdate with ($sigs"
    stateid=0
fi
exit $stateid
# Checking Virus Signatures - NRPE

## Vagios Core Config Manager

### Service Management

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config Name</td>
<td>check_signatures</td>
</tr>
<tr>
<td>Hosts</td>
<td>192.168.5.45, localhost</td>
</tr>
<tr>
<td>Host groups</td>
<td>linux-servers</td>
</tr>
<tr>
<td>Service description</td>
<td>check_signatures</td>
</tr>
<tr>
<td>Display name</td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>checked</td>
</tr>
<tr>
<td>Check command</td>
<td>check_nrpe</td>
</tr>
<tr>
<td>Command view</td>
<td>$USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$ $ARG2$</td>
</tr>
<tr>
<td>$ARG1$</td>
<td>check_signatures</td>
</tr>
<tr>
<td>$ARG2$</td>
<td></td>
</tr>
<tr>
<td>$ARG3$</td>
<td></td>
</tr>
<tr>
<td>$ARG4$</td>
<td></td>
</tr>
<tr>
<td>$ARG5$</td>
<td></td>
</tr>
<tr>
<td>$ARG6$</td>
<td></td>
</tr>
<tr>
<td>$ARG7$</td>
<td></td>
</tr>
<tr>
<td>$ARG8$</td>
<td></td>
</tr>
</tbody>
</table>

### Additional templates

<table>
<thead>
<tr>
<th>Template Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>generic-service</td>
<td></td>
</tr>
</tbody>
</table>
Checking Virus Activity - NRPE

Command Definition
define command{
    command_name  check_nrpe
    command_line  $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

Service Definition
define service{
    use              generic-service
    host_name        mail
    service_description Quarantine Status
    check_command    check_nrpe!check_virus_activity
}

NRPE Command
command[check_virus_activity]=/usr/local/nagios/libexec/check_virus_activity

Bash Shell Script
#!/bin/bash
vmail=$(ls /var/virusmails | grep virus|wc -l)
echo "Virus Activity $vmail"
exit 1
Checking Quarantine - NRPE

**Command Definition**
define command{
    command_name    check_nrpe
    command_line    $USER1$ /check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

**Service Definition**
define service{
    use              generic-service
    host_name        mail
    service_description Quarantine Status
    check_command    check_nrpe!check_virusmail
}

**NRPE Command**
command [check_virusmail] = /usr/local/nagios/libexec/check_virusmail

**Bash Shell Script**
#!/bin/bash

vmail=$(ls /var/virusmails | grep virus | wc -l)
vmail_date=$(ls -l /var/virusmails | grep virus | awk -F' ' '{print $6,$7,$8}' | tail -1)

if [ "$vmail" -eq 0 ]
then
    echo "No Viruses in Quarantine"
    stateid=0
else
    echo "Viruses Detected!!! Last Virus Captured $vmail_date"
    stateid=1
fi
exit $stateid
Monitor Email Delivery – Perl Plugin

- Delivery Confirmation to INBOX
  Verify that mail was is deliverable.

- Delivery Confirmation: Read Header
  Read mail header to verify delivery.

- Delivery Confirmation: Read Header/Content
  Read header and content to verify readability.
**Create Command**

Whenever you use your own script, you will need to create a command to access the script.

**Create Check**

This example “hard codes” the check until you know it works, then add arguments.
Monitor with SSH Proxy: Secure Communication

- Amavis - SNMP
- Reinjection Port - SNMP
- Spamassassin - SNMP
- Virus Signatures
- Quarantine Status
- Number of Viruses Captured
SSH Proxy

This wizard monitors the remote host using SSH to execute the plugins and scripts.

Monitoring Wizard - Step 1

Monitoring wizards guide you through the process of monitoring devices, servers, or other components. You can find additional configuration wizards for Nagios XI at Nagios Exchange.

- **Generic Network Device**
  Monitor a generic IP network device.

- **Printer**
  Monitor an HP JetDirect® compatible network printer.

- **SNMP**
  Monitor a device, service, or application using SNMP.

- **SSH Proxy**
  Monitor a remote Linux/Unix server using SSH.

Download and install the SSH Proxy wizard. Once it is installed select the wizard from the list.
SSH Proxy

In Step 2 you will need to add an IP Address or fully qualified domain name. You will also need to select the operating system of the machine you will connect up to using SSH.

**SSH Proxy Monitoring Wizard - Step 2**

**Server Information**

**IP Address:**
mail.linuxtrainingcenters.com
The IP address or FQDN name of the server you'd like to monitor.

**Operating System:**
Linux - CentOS
The operating system running on the server you'd like to monitor.
In Step 2 you will need to add an IP Address or fully qualified domain name. You will also need to select the operating system of the machine you will connect up to using SSH.
SSH Proxy

Server Details

IP Address: mail.linunxtrainingcenters.com
Operating System: CentOS
Host Name: mail.linunxtrainingcenters.com

The name you’d like to have associated with this server.

Server Metrics

Specify which services you’d like to monitor for the server.

- Ping
  Monitors the server with an ICMP “ping”. Useful for watching network latency and general uptime.

SSH Commands

Specify any remote commands that should be executed/monitored on the server using SSH.

<table>
<thead>
<tr>
<th>Remote Command</th>
<th>Display Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/local/nagios/libexec/check_amavis</td>
<td>Amavis: Virus Protection</td>
</tr>
<tr>
<td>/usr/local/nagios/libexec/check_spamassassin</td>
<td>Spamassassin</td>
</tr>
<tr>
<td>/usr/local/nagios/libexec/check_reinjection</td>
<td>Reinjection Port</td>
</tr>
<tr>
<td>/usr/local/nagios/libexec/check_signatures</td>
<td>Virus Signatures</td>
</tr>
<tr>
<td>/usr/local/nagios/libexec/check_virus_activity</td>
<td>Virus Activity</td>
</tr>
<tr>
<td>/usr/local/nagios/libexec/check_virusmail</td>
<td>Virus Mail Level</td>
</tr>
</tbody>
</table>
### SSH Proxy

**Nagios Core Config Manager**

**Service Management**

<table>
<thead>
<tr>
<th>Common Settings</th>
<th>Check Settings</th>
<th>Alert Settings</th>
<th>Misc Settings</th>
</tr>
</thead>
</table>

#### Common Settings

- **Config Name**: mail.linuxtrainingcenter
- **Hosts**: localhost, mail.linuxtrainingcenters.com
- **Host groups**: linux-servers
- **Service description**: Amavis: Virus Protection
- **Display name**:
- **Active**: checked
- **Check command**: check_xi_by_ssh
- **Command view**:

```
$USER1$/check_xi_by_ssh -H $HOSTADDRESS$ $ARG1$ $ARG2$
$ARG1$
$ARG2$
$ARG3$
$ARG4$

- **ARG5**
- **ARG6**
- **ARG7**
- **ARG8**
```

- **Additional templates**
  - **Template Name**: generic-service

- `-C "/usr/local/nagios/libexec/check_amavis"`
SSH Proxy – Creating Keys

The key to getting the whole thing to work is setting up the passwordless login ability of the nagios user. On the XI box login as the nagios user:

```
su - nagios
cd /home/nagios
ssh-keygen
```

Use ENTER to select all options as you want to take default locations and you want a password that is empty (be sure to set up the security requirements listed below). On the host to be monitored follow the same steps. Then on the XI server, log in as nagios and go to the ssh directory.

```
su - nagios
cd /home/nagios/ssh
cp id_rsa.pub nagios_key
scp nagios_key nagios@remote_client:/home/nagios/.ssh/nagios_key
```

You copy the public key to a different name, otherwise you will wipe out the public key on the remote client. Now log into the remote client as nagios and move to the /home/nagios/.ssh directory. Execute these commands:

```
cat nagios_key > authorized_keys
chmod 600 authorized_keys

ls -l
-rw------- 1 nagios nagios 394 Sep 14 16:24 authorized_keys
-rw------- 1 nagios nagios 1671 Sep 14 16:18 id_rsa
-rw-r--r-- 1 nagios nagios 418 Sep 14 16:18 id_rsa.pub
```

You should now be able to log in to the remote host from Nagios XI without a password.
SSH Proxy – Security

If you are using the nagios login without a password and with an empty key-phrase, it is important that you set a firewall rule to only allow connections using SSH from trusted hosts. Here is an iptables rule (on a CentOS box) which uses one rule to allow the Nagios XI to use several different ports. Notice the rule order is used with this rule being “7” so that you can block all access after this rule.

Firewall

iptables -I RH-Firewall-1-INPUT 7 -p tcp -m state --state NEW -m multiport -s 192.168.1.1 --dports 110,995,993,9202,22 -j ACCEPT

In addition set your tcp_wrappers file in /etc/hosts.allow so that only trusted hosts can get access to the server using SSH. Be sure to edit this file carefully so you do not lock yourself out. You will also need to edit /etc/hosts.deny to deny everything you do not allow.

# hosts.allow
This file describes the names of the hosts which are allowed to use the local INET services, as decided by the '/usr/sbin/tcpd' server.

# ALL: 127.0.0.1
SSHD: 192.168.1.1
SMTP: ALL
POP3: ALL
IMAPS: ALL

# hosts.deny
ALL: ALL