Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Multi-client MRTG platform

GIAC Certified Unix Security Administrator (GCUX)
Practical Assignment - Version 2.0 Option 1
CDI East 2003 – Washington DC

T. Brian Granier

© SANS Institute 2004, Author retains full rights.

Abstract....4
Server Specification....5
Server role....5
Hardware Requirements....5
OS Version....5
Third party software....5
Additional processes....6
Available services....6
User access....6
Risk Mitigation Plan....7
Risk Identification....7
Business Criticality....7
System exposure....8
Accepted risks....8
Steps to Install the Server....9
Gather the media....9
Initial OS installation....9
Post-Installation....15
Install third party applications....15
System hardening and configuration....18
Disable unnecessary packages....18
Disable mingetty processes....18
Sendmail configuration....19
Configure NTP....19
System Resource and Kernel Limits....19
Network parameter configuration....20
File system options....21
User account creation....22
SSH Configuration....23
Protecting single-user shell....24
Disable anonymous shutdown....24
Configuring MRTG....24
Configuring Apache....27
Design and Implement Ongoing Maintenance Procedures....31
Overall Plan....31
Backup Plan....31
Patch Maintenance....31
Log monitoring....32
Tripwire and integrity checking....37
Vulnerabilty checks....37
Test and Verify the Setup....38
Verifying SSH configuration....38
Checking httpd configuration....40
Verifying boot loader password protection....43
Verify Single User Shell password protection....44

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Checking file system options 44
References 45

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Abstract

This paper covers the installation of a system on RedHat 9 designed to run mrtg to gather data via SNMP and store it in rrd format and then to provide reports to customers in web format with Apache running SSL. The risks are identified followed by a detailed installation and lockdown procedure and finished off with ongoing maintenance procedures and verification steps.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Server Specification

Server role

The system explained in this environment is created to provide MRTG services to clients of the company SANSCO. SANSCO is a company that provides managed IT services for their clients ranging from a broad spectrum of small, medium and large businesses. Their managed services range from maintaining a simple static web server to being ultimately responsible for all servers, applications, WAN connectivity, LAN configuration and workstation support. As a part of the value added services they provide to their clients, this system offers monitoring services that not only monitors uptime, but also provides a baseline for long-term system and network health by querying objects via SNMP and storing the values in a file that can be graphed and provided in a web browser. This system runs MRTG/RRDTool to obtain this information and provide the web based utilization statistics to all of their clients. SANSCO clients access this system over the Internet, through site to site VPN's or through private line connections depending upon their existing connectivity requirements.

Hardware Requirements

The system required to perform this function does not need to be very powerful. For SANSCO, the following system has been identified as the MRTG/RRDTool system:
Type: Deskpro EXS P733
Processor: P3/733 Mhz
Memory: 256 MB of RAM
Storage: Single Ultra ATA/66 - 40 GB Hard Drive
Network: Built in 3Com 10/100
CD ROM: 48X IDE CD-Rom
Floppy: 3.5" floppy drive

OS Version

The OS Version selected for this installation is Redhat 9. While there are a wide variety of operating system choices available, Redhat 9 was select predominantly due to the comfort level of the administrator(s) responsible for the system.

Third party software

The third party software that will be used in this installation is RRDTool available from http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/download.html, OpenSSH available from http://www.openssh.org/portable.html and OpenSSL available at http://www.openssl.org/source/. While OpenSSH and OpenSSL is currently distributed as part of the base operating system bundle, these applications will be specifically downloaded and installed from these third party sites.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Additional processes

In addition to the third party applications mentioned above this system will be running an SSH server for management and httpd (with SSL configured) for providing the web front-end for customers to access the RRDTool generated images.

Available services

Once the system has been completely installed, SSH (TCP port 22) and HTTPS (TCP port 443) are expected to be the only ports listening on the system.

User access

Administrative users are the only ones expected to access the system over SSH for administrative purposes. The customers who are the primary users of the system will only be able to access the system through the web front-end and will, therefore, not have actual Operating System level accounts.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Risk Mitigation Plan

Risk Identification

The largest risk that will take the most amount of work to mitigate is preventing one customer from being able to view data meant for another client. By default, the cgi script used to deliver the MRTG/RRDTool data is not designed to prevent one web session from being able to view data meant for another user. To mitigate this risk, special attention will be paid to the configuration of the website within apache and steps will be taken to modify the cgi script.

Another concern is due to the fact the Redhat 9 will be discontinued in the near future. The biggest risk here is that Redhat will stop supplying patches to the operating system. SANSCO is aware of this issue and anticipates either changing to another operating system or upgrading to one of the Redhat supported operating systems. It is anticipated that the steps taken in the installation and configuration of this system can be used to facilitate this change when the time comes.

Beyond this risk, the same issues apply to this system as nearly all systems in an enterprise. Due to improper patch management and system administrative maintenance, the system could fall behind and become vulnerable to a system level compromise. To mitigate this risk, IPTables will be employed to restrict the traffic to only the required ports, patch maintenance will be implemented as a routine event and logs will be automatically processed to look for any suspicious activity. Attention will also be paid to obtaining regular backups of the data to recover in the event of a hardware failure.

Business Criticality

From a technical perspective, this system is not mission critical. However, given that the source IP address of this host is expected to be used throughout the enterprise in access lists and firewall policies that restrict the usage of SNMP, it's important to ensure it safe operation in order to prevent a potential cascading compromise whereby this system is compromised and used as a launching point to obtain even more data or to reconfigure remote devices with SNMP. More likely of a threat is that if this system were known to be compromised, it could potentially create a bad public relations scenario where SANSCO loses credibility with its clients who depend upon SANSCO to maintain a secure networking environment. Given these considerations, it seems appropriate that the steps taken are reasonable given the potential risks of the system not being up. It is noted that additional hardening could be performed (such as removing the floppy and CD-ROM drive from the system, removing GCC, permitting console only administrative access, etc…) but these are deemed unwarranted for the situation. Additional steps could also be taken to ensure the availability of the data, but the potential for hardware failure that requires a restoration and likely causing an outage for as long as 24 hours is deemed acceptable.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

System exposure

This system is available to hosts on the Internet. SANSCO maintains an appropriately configured border router that blocks a large amount of illegitimate traffic. Next, the SANSCO corporate firewall filters traffic, and finally the IPTables process running on the system directly provides filtering to help protect against Internet based threats. Only the https service is permitted for access from hosts outside the SANSCO firewall. Administrative users who connect via SSH to manage the system are being restricted to source IP addresses on RFC 1918 address space and, therefore, must connect from within the SANSCO private network.

Accepted risks

These risks (some noted previously) are a few of the ones deemed acceptable:
The usage of SNMP: SNMP is known to have security flaws. In this environment, the usage of SNMP will be limited by access lists and firewall policies. Additionally, only the read only community string will be used
The usage of a CD-ROM drive: The installation of a CD-ROM drive could lead to a potential intruder being able to load or reboot the system into an environment that could be hostile to the security of the system. Since this system is located in a secure access controlled facility, this risk is considered acceptable.
The usage of a floppy drive: Risks the same as for a CD-ROM drive.
Single point of failure: From a hardware perspective, this system does not attempt to mitigate single points of failure. This is considered an acceptable risk since the application is not business critical and a 24 hour restoration window is deemed acceptable.
GCC compiler available: Some security experts suggest that no compiler should be available on a production system. The purpose is to make it more difficult for a potential attacker to be able to upload and compile code for purpose of furthering their attack. Due to the combination of mitigation techniques and business criticality, it was determined that the loss in ease of management (compiling and installing software updates) was more important and that the fact that a potential intruder could easily upload their own compiler or compile attack scripts off-line makes this issue less significant.
SSH available for remote access: In a more secure environment, administrative access would only be permitted from the console. The criticality of this application does not warrant that additional administrative overhead.
Apache running in normal environment: It would be possible to run httpd in a chrooted environment. However, given the likelihood of compromise and the potential impact of a compromise when weighed against the added system administration requirements, it has been determined that this is unnecessary in this case.
RedHat 9 retirement: With the impending retirement of RedHat 9, patches will cease to be available to RH9 from Redhat. SANSCO is already making plans for either upgrading to a newer version of RedHat or migrating to a different operating system altogether.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Steps to Install the Server

Gather the media

This system will be built offline until it has reached a reasonable state and connected to the network. For this reason, the media necessary to install this system is gathered to be installed from CD-ROM. The following materials are gathered in order to perform the installation:

1. Redhat 9 CDs purchased from Redhat
2. A copy of the latest version of OpenSSH (currently 3.7.1p2).
3. A copy of the latest version of OpenSSL (currently 0.9.7c).
4. A copy of the latest version of RRDTool (currently 1.0.46).
5. A copy of the latest version of GD (currently gd-2.0.20)
6. A copy of the latest version of libpng (currently libpng-1.2.5)
7. A copy of the latest version of httpd (currently httpd-2.0.48)
8. A copy of the latest version of MRTG (currently -2.10.12)
9. A copy of the latest version of zlib (currently zlib-1.2.1)
10. A copy of the latest version of sendmail (currently sendmail-8.12.10)

When available, the MD5 hash is compared to ensure the download has not been tampered with.

Initial OS installation

The OS is initially installed form the RedHat 9 media. The operating system installation is described with the tables attached below. Note that these tables were originally found in a previous GCUX practical located at http://www.giac.org/practical/GCUX/Jacqui_Chau_GCUX.pdf. The table found in this practical was used as a baseline and the text was edited to fit this specific installation and to augment the explanations where it was appropriate.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Question/Option	Explanation	Action
Option to install or upgrade RedHat Linux in graphical of text mode	Choose how you would like to view the installation screens. Graphical or text based. Graphical is much easier and intuitive to use.	Press the <Enter> key to install in graphical mode
RedHat 9 GUI	Explanation and Welcome	Click on Next
Language Selection	Select your language	English
Keyboard	Select your keyboard type	U.S. English
Mouse Configuration	Select type of mouse you will be using with the system. Usually you should select the default	3 button mouse (PS/2)
Installation Type	**Server** (allows file sharing, print sharing and web services) **Custom** (more configurable)	Select <Custom>
Disk Partitioning Setup	**Automatic Partitioning**: Selects defaults of: /boot / swap **Manual Partitioning** (Disk Druid): Allows you to configure how many partitions you require and the size of each partition.	**Manual Partitioning** /boot 100 MB / 10 00 MB /tmp 1000 MB /usr 1000 MB /home 1000 MB swap 768 MB /var 100 MB and select fill to maximum available size
Boot Loader Configuration	Grand Unified Boot Loader (GNU GRUB) is the default boot loader. It can load multiple operating systems. The boot loader is required in order to boot a system without a boot diskette. It is the first software program that runs when a computer starts and is responsible for loading and transferring control to the operating system kernel software.[1]	Leave default selection of GRUB boot loader /dev/hda6 Select 'Use a boot loader password' and enter the password

[1] http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/s1-x86-bootloader.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

	A boot loader password prevents users from changing options passed to the kernel. For greater system security, it is recommended that you set a password.	
Network configuration	The system will have a statically configured IP address. The settings shown on the right demonstrate a possible configuration. The details will be dependant upon your networking environment.	Manually configure IP information Click on edit on Network Interface Deselect Configure using DHCP Active on boot is selected IP: 192.168.1.40 Netmask: 255.255.255.0 Select OK Set the hostname manually: Hostname: mrtg.sansco.com Gateway: 192.168.1.1 Primary DNS: 192.168.1.5 Secondary DNS: 192.168.3.5 Tertiary DNS: 192.168.5.5
Firewall Configuration	**High Security:** Should use this option if you are connecting your system to the Internet, but do not plan to run a server. Add trusted devices or to allow additional incoming interfaces. https (port 443) and SSH (port 22) are the only expected inbound ports required.	Select a security level for the system: **High** Select <Customize> Select Trused devices: <eth0> Allow incoming: SSH Other ports: 443
Additional Language Support	Select your language	English (USA)
Time Zone Selection	Select your timezone. Note that not all cities are loaded in the list. Look for an entry that is in the same timezone as your location and use that.	Location: America/Chicago
Set Root Password	Ensure that you select a complex password. Use	Enter the root (administrator) password

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

	at least 8 characters with mixed uppercase/lowercase, symbols and numbers.	for the system
Authentication Configuration	The default options will use MD5 encryption and will shadow the passwords. In most environments this will do very well. If you have some type of external authentication mechanism, than other options might be appropriate, but in most cases, the default will do.	Leave default
Package Group Selection		Select Minimum Installation Click on the individual packages check box Click on next to view in a tree view
Individual Package Selection		On the left pane, select All packages and then click “Unselect all in group” Add the following packages: • apmd • binutils • cpp • crontabs • devlabel • diffutils • elfutils • gcc • glibc-devel • glibc-kernheaders • gnupg • groff • iptables • libcap • libstdc++ • logrotate

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

		• logwatch • lsof • ltrace • m4 • mailcap • mailx • make • man • man-pages • ntp • perl • perl-CGI • perl-Filter • perl-URI • pyOpenSSL • python-optik • rhnlib • rpm-python • slocate • strace • tcp_wrappers • tmpwatch • tripwire • unzip • up2date • vixiecron • zip
Insert disks 2 and 3 when prompted		Insert disks 2 and 3 when prompted
Boot Diskette Creation	Boot disks are often very important in a disaster recovery situation. They provide information about how to boot the operating system that is being installed and are unique to the specific disk	Yes, I do want to create a boot diskette

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

	configuration that was installed on this system. It should be stored with any backup media.	
INSTALLATION COMPLETE	INSTALLATION COMPLETE	CD should eject. Then click on exit and system should reboot.

Of importance is the partitioning, so I'd like to take a moment to discuss it.

/boot 100 MB – The boot partition is separately mounted so that it can be mounted ro.
/ 1000 MB - The root partition is required on every installation.
/tmp 1000 MB - The /tmp is created as its own partition in order to help protect against a possible situation where completely filling up the tmp directory could cause the system to crash if it were part of the / partition. This comes at the cost of a little bit of performance loss due to having an additional partition. This is not an overutilized system, so the performance concern is not a big issue.
/usr 1000 MB - The /usr partition is separately created so that specific mount configuration can be applied.
/home 1000 MB - The /home partition is separately created so that specific mount configuration can be applied.
swap 768 MB – Required by the operating system. Note that it is more than twice the size of the RAM.
/var remainder – This is httpd and mrtg will be located. It is the largest partition since this is where the MRTG data files will be stored and is expected to be the largest partition.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Post-Installation

In order to facilitate repeating this build, the anaconda-ks.cfg stored in the /root directory should be saved. This file can be used to install an identical system in the future. Prior to installing third party applications, we will check for and install updates. Since we did not install a GUI environment, this will be done during the command line version of up2date. Information about doing this can be found at http://www.redhat.com/docs/manuals/RHNetwork/ref-guide/2.8/up2date-config.html#UP2DATE-CONFIG-TEXT.

The following steps update the installed packages:

- up2date --nox --configure
- Press enter
- rpm --import /usr/share/rhn/RPM-GPG-KEY
- mv /usr/share/rhn/RHNS-CA-CERT /usr/share/rhn/RHNS-CA-CERT.old
- copy in new RHNS-CA-CERT available at https://rhn.redhat.com/help/RHNS-CA-CERT
- up2date --register
- Complete the wizard
- Ensure that entitlements are appropriately applied
- up2date –u
- up2date –f kernel
- reboot

Note that the steps involving the RHNS-CA-CERT file are only necessary if the installation media is not current. In newer distributions, these steps would be unnecessary.

Install third party applications

Several applications that will be used on the system have not yet been installed. Some of them could have been installed during the OS installation, but were not due to concerns about the version that would be installed. Specifically, OpenSSH[2], OpenSSL[3] and Sendmail[4] were saved for post installation for these reasons. It is important to note that the Redhat release notes claim that although the version numbers for these packages appear to be affected by these advisories, the patches have been back-ported and the RHN released versions of these applications are properly patched. By going ahead with externally downloading and installing these applications, system audits will be less confusing. RRDTool also needs to be installed. MRTG could have been installed during the OS installation, but there are some known application bugs that aren't security related with the version it installs. Therefore, I have chosen to install it and the packages it depends upon separately.

[2] Concern about advisory posted at http://www.cert.org/advisories/CA-2003-24.html
[3] Concern about advisory posted at http://www.cert.org/advisories/CA-2003-26.html
[4] Concern about advisory posted at http://www.cert.org/advisories/CA-2003-25.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

OpenSSL

- tar –zxvf openssl-0.9.7c.tar.gz
- mv openssl-0.9.7c openssl
- cd openssl
- ./config
- make
- make test
- make install

OpenSSH

- rpm –i openssh-3.7.1p2-1.i386.rpm
- rpm –i openssh-clients-3.7.1p2-1.i386.rpm
- rpm –i openssh-server-3.7.1p2-1.i386.rpm

HTTPD

- tar -zxvf httpd-2.0.48.tar.gz
- mv httpd-2.0.48 httpd
- cd httpd
- . /configure --prefix=/var/www --enable-ssl=/usr/local/ssl
- make
- make install

Sendmail

- tar –zxvf sendmail.8.12.10.tar.gz
- mv sendmail-8.12.10 sendmail
- cd sendmail
- sh Build
- cd cf/cf
- cp generic-linux.mc sendmail.mc
- sh Build sendmail.cf
- mkdir /etc/mail
- sh Build install-cf
- groupadd smmsp
- useradd –g smmsp smmsp
- cd ../..
- sh Build install

The smssp group and user are created as recommended by the sendmail installation documentation. This user and group is intended as the group the owns and runs the sendmail process so that it doesn't have to run as root.

The steps for zlib, libpng, gd and mrtg installation are available from http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-unix-guide.html.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

ZLIB

- tar –zxvf zlib-1.2.1.tar.gz
- mv zlib-1.2.1 zlib
- cd zlib
- ./configure
- make

LIBPNG

- tar –zxvf libpng-1.2.5.tar.gz
- mv libpng-1.2.5 libpng
- cd libpng
- make –f scripts/makefile.std CC=gcc ZLIBLIB=../zlib ZLIBINC=../zlib
- rm *.so.* *.so

GD

- tar -zxvf gd-2.0.20.tar.gz
- mv gd-2.0.20 gd
- cd gd
- env CPPFLAGS="-I../zlib –I../libpng" LDFLAGS="-L../zlib –L../libpng" ./configure --disable-shared --without-freetype --without-jpeg
- make
- cp .libs/* .

MRTG

- tar -zxvf mrtg-2.10.12.tar.gz
- mv mrtg-2.10.12 mrtg
- cd mrtg
- ./configure --prefix=/usr/local/mrtg-2 --with-gd=../gd --with-z=../zlib --with-png=../libpng
- make
- make install

RRDTool

- tar –zxvf rrdtool-1.0.46.tar.gz
- mv rrdtool-1.0.46 rrdtool
- cd rrdtool
- sh configure
- make
- make install

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

System hardening and configuration

Now it's time to complete the configuration for each of the installed services and to harden the system. Many of the steps taken here are outlined in the "6.6 Unix Security Lab" book distributed as part of the SANS Track 6 courseware.

Disable unnecessary packages

In this stage, processes that are not needed will be disabled and general hardening will be taken to help secure the system. To check what services are currently running the system, the command chckconfig --list is issued. The following output is displayed:

```
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
keytable        0:off   1:on    2:on    3:on    4:on    5:on    6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

After reviewing the running services, the following commands were run:

```
chkconfig --level 123456 netfs off
chkconfig --level 345 ntpd on
```

netfs is used to map file shared for NFS, SMB and NCP mount points and is not needed for this system. httpd and ntpd will be used in this installation and so they are turned on.

Disable mingetty processes

By default, there are six tty consoles available on a RedHat system. To protect against other tty consoles being in use and therefore potentially hidden from the monitor output, tty2 through tty6 should be disabled. To accomplish this task, the /etc/inittab file is modified as follows:

Default Setting:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Modified Setting:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
#2:2345:respawn:/sbin/mingetty tty2
#3:2345:respawn:/sbin/mingetty tty3
#4:2345:respawn:/sbin/mingetty tty4
#5:2345:respawn:/sbin/mingetty tty5
#6:2345:respawn:/sbin/mingetty tty6
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Sendmail configuration

In this environment, sendmail will be used to send out notifications and reports. The sendmail configuration needs to be modified to ensure that it is only enabled to send mail outbound. Since we are running an 8.12.x version of sendmail and we're only sending outgoing mail, we're only interested in configuring the submit.cf file. To do so, we will modify the submit.mc file that came with the sendmail distribution. First the submit.mc file that came with the distribution is copied to the /etc/mail directory and then it is modified as follows:

Default: `define(`_USE_DECNET_SYNTAX_', `1')dnl support DECnet`
Change to: `define(`_USE_DECNET_SYNTAX_', `0')dnl support DECnet`

Default: `FEATURE(`msp', `[127.0.0.1]')dnl`
Change to: `FEATURE(`msp', `mail.SANSCO.com')dnl`

Next the new submit.mc needs to be converted to .cf format. To do that, run the command m4 /etc/mail/submit.mc > /etc/mail/submit.cf

Since sendmail is only being used to process outgoing emails with the MSP, it does not need to be invoked in cron or startup, so no modification is needed to do so. Applications attempting to send outgoing emails will invoke sendmail directly.

Configure NTP

Next, we want to configure the NTP synchronization on the system. NTP is important on this system predominantly since it will ensure accuracy in the timestamps on logs that will come from this system. It is expected that SANSCO.com has an existing ntp infrastructure. The /etc/ntp.conf file is modified as follows:

```
server time1.sansco.com
server time2.sansco.com
server time3.sansco.com
driftfile /etc/ntp/drift
restrict default nomodify
restrict 127.0.0.1
```

System Resource and Kernel Limits

A choice was made to disable core files on this system. While core files can contain useful information when troubleshooting a problem, they can also contain things such as passwords and are usually world readable. To prevent a condition where an unauthorized user manages to cause a core dump and reads the file and manages to gain a password they shouldn't have, we will turn it off at the system level. Additionally, we'll place a hard and soft limit on the number of processes and open files that any given user may have open at a time. We do expect that the user account that runs the mrtg process will have the most number of processes and files open at any given point in time, so the settings

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

used will be based upon the number of processes and files we expect that account to have open. Basically, take the number of unique mrtg configuration files (and therefore the number of processes initiated of mrtg) and use that number as the soft limit of processes. Double or triple the number and use that as the hard limit for the processes. Quadruple these numbers for the file limits because each instance of MRTG typically opens up to four files. If the environment is accustomed to a lot of additions and removals of mrtg configuration files, it might be better to increase these numbers accordingly to make room for growth. For this example, we'll assume 64 configuration files are in existence and the mrtg configuration files are fairly static. The /etc/security/limits.conf file is modified as follows:

```
# Prevent core dumps at system level
*           hard  core  0

# User processes
*           soft  nproc 64
*           hard  nproc 128

# Open files
*           soft  nofiles     256
*           hard  nofiles     512
```

Network parameter configuration

The default behavior of the network is modified to help enhance the network level security of the system. The configuration file modified is located at /etc/sysctl.conf. This file is modified as follows:

```
# Helps protect against syn flooding
net.ipv4.tcp_max_syn_backlog = 4096

# Interface parameters - active interfaces
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.send_redirect = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0

# Interface parameters - future interfaces
net.ipv4.conf.default.log_martians = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.send_redirect = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
```

The net.ipv4.conf.* settings are duplicated for "all" and for "default". This is because the "all" keyword applies to all interfaces currently active at boot time. The "default" keyword applies to any interface activated later. The tcp_max_syn_backlog settings sets the number of half-open connections that

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

can be in the queue. Note that this doesn't completely prevent a syn flood attack, since it's possible to deplete the queue before all half-open connections timeout. The log_martians keyword "variable tells the kernel to log all packets that contains impossible addresses to the kernel logging facility. An impossible IP address may mean an IP address that we do not know how to contact, since the IP address is not contained in the routing tables."[5] The rp_filter keyword talks about when a packet from a source interface enters on an interface other than the one that would be used when routing a packet back to the same system. This is really only useful with a multi-homed system and essentially helps prevent either spoofing attacks or asymmetric route issues. The accept_source_route parameter deals with a packet whose IP headers define the routing path that is to be taken for the packet. This parameter could be used for a man in the middle or IP spoofing attack to allow an attacker to appear to come from an address they aren't really coming from. The three redirect options deal with route redirections whereby a hosts routing table can be modified for specific hosts by other devices on the network. By disabling this functionality, you help to protect against potential man in the middle attacks where a malicious host takes over the routing functionality on the local network and is able to, therefore, have a complete view of the network traffic and potentially even hijack a connection as a result.

File system options

The mounting options for the file system needs to be set. The /etc/fstab file is modified as follows:

```
LABEL=/       /             ext3          defaults                          1 1
LABEL=/boot   /boot         ext3          ro,nodev                          1 2
none          /dev/pts      devpts        gid=5,mode=620                    0 0
LABEL=/home   /home         ext3          rw,nosuid,nodev                   1 2
none          /proc         proc          defaults                          0 0
none          /dev/shm      tmpfs         defaults                          0 0
LABEL=/tmp    /tmp          ext3          rw,nosuid,nodev                   1 2
LABEL=/usr    /usr          ext3          ro,nodev                          1 2
LABEL=/var    /var          ext3          rw,nosuid,nodev                   1 2
/dev/hda7     swap          swap          default                           0 0
/dev/cdrom    /mnt/cdrom    udf,iso9660 noauto,owner,kudzu,ro,nosuid,nodev 0 0
/dev/fd0      /mnt/floppy   auto          noauto,owner,kudzu,nosuid,nodev   0 0
```

The / directory on a Redhat system needs to be more or less open in order for the operating system to operate correctly. While there's useful binaries in /lib and /bin, there's not much that can be done to set this partition read only. However, the /usr directory where the majority of useful binaries are stored can be. This can help prevent against binaries being replaced with trojanned versions in order to help an attacker escalate their privileges to root. The remaining disk partitions, including cdrom and floppy, are set to nosuid and nodev to help protect against suid binaries being placed in these directories as part of a privilege escalation attack. The rw option is set on the disk partitions so that data can be read and written to them.

[5] http://ipsysctl-tutorial.frozentux.net/chunkyhtml/theconfvariables.html#AEN612

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Next we want to disable automatic mounting of floppies and CD-ROMs. Since the only users who have the authority to login to the system are system administrators and should therefore be able to su to root, there is no need for auto-mounting. To turn it off, the /etc/security/console.perms file is modified by commenting out the following lines as shown:

```
#<floppy>=/dev/fd[0-1]* \
#         /dev/floppy/*/* /mnt/floppy*
#<cdrom>=/dev/cdrom/* /dev/cdroms/* /dev/cdwriter* /mnt/cdrom*
#<pilot>=/dev/pilot
#<jaz>=/mnt/jaz*
#<zip>=/mnt/pocketzip* /mnt/zip*
#<ls120>/dev/ls120 /mnt/ls120*
#<camera>=/mnt/camera* /dev/usb/scanner*
#<memstick>=/mnt/memstick*

#<console> 0660 <floppy>      0660 root.floppy
#<console> 0600 <cdrom> 0660 root.disk
#<console> 0600 <pilot> 0660 root.uucp
#<console> 0600 <jaz>   0660 root.disk
#<console> 0600 <zip>   0660 root.disk
#<console> 0600 <ls120> 0660 root.disk
#<console> 0600 <camera>      0600 root
#<console> 0600 <memstick>    0600 root
```

User account creation

There are a total of five user accounts that need to be created. One will be used for running the mrtg process. The remaining accounts will be used by system administrators at SANSCO in order to access the system. These accounts are created as follows:

```
groupadd sysadmin
useradd -g sysadmin fred
useradd -g sysadmin wilma
useradd -g sysadmin barnie
useradd -g sysadmin betty
groupadd mrtg
useradd -g mrtg mrtg
passwd fred
passwd wilma
passwd barnie
passwd betty
```

Next, accounts created by default, but that aren't used are deleted. The idea for doing this comes from Jacquiline Chau's practica[6]l. Issue the following commands:

```
userdel adm
userdel lp
userdel shutdown
userdel halt
userdel news
userdel uucp
```

[6] http://www.giac.org/practical/GCUX/Jacqui_Chau_GCUX.pdf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
userdel operator
userdel games
userdel gopher
userdel ftp
```

Next the unnecessary groups are deleted:

```
groupdel adm
groupdel lp
groupdel news
groupdel uucp
groupdel games
groupdel dip
groupdel users
groupdel lock
```

SSH Configuration

The configuration file located at /etc/ssh/sshd_config is modified in order to configure the SSH server. The following configuration represents all the uncommented lines in the configuration file:

```
Port 22
Protocol 2

HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

KeyRegenerationInterval 1h
ServerKeyBits 768

SyslogFacility AUTHPRIV
LogLevel INFO

LoginGraceTime 2m
PermitRootLogin no
StrictModes yes

RSAAuthnetication no
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys

RhostsRSAAuthentication no
HostBasedAuthentication no
IgnoreUserKnownHosts yes
IgnoreRhosts yes

PasswordAuthentication yes
PermitEmptyPasswords no

ChallengeResponseAuthentication no

KerberosAuthentication no
KerberosOrLocalPasswd no
KerberosTicketCleanup no
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
AllowTcpForwarding no
X11Forwarding no
PrintMotd yes
PrintLastLog yes
KeepAlive yes
UseLogin no
UsePrivilegeSeparation yes
PermitUserEnvironment no
Compression yes

Banner /etc/ssh/sshbanner

Subsystem   sftp  /usr/libexec/openssh/sftp-server
```

The most important aspects of this configuration file is the disabling of root login (PermitRootLogin no) and the usage of version 2 and not 1 (Protocol 2).

Once this has been completed, a banner needs to be created at /etc/ssh/sshbanner. The contents of this banner file is to conform with the corporate approved banner that has been reviewed by legal counsel. For purpose of this paper, the following is used to create the file:

```
echo If you do not belong here, go away > /etc/ssh/sshbanner
```

Since SSH is already configured to start during bootup, nothing more is required.

Protecting single-user shell

In order to ensure that only a user who knows the root password can use the system in single user mode, the /etc/inittab file will be modified to prompt for root login when the system enters single user mode. The following line is added to this file:

```
sum:S:wait:/sbin/sulogin
```

Disable anonymous shutdown

In order to prevent the system from being shutdown anonymously by pressing <ctrl><alt><del> from the system keyboard, the /etc/inittab file is modified as follows:

```
ca::ctrlaltdel:/usr/bin/logger -p authpriv.info 'ctrl-alt-del trapped'
```

Configuring MRTG

The detailed MRTG configuration files are an exercise that is outside the scope of this document. Information about configuring the MRTG files is found at http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-reference.html. One important thing to note is that a configuration file is capable of running scripts, so care must be taken not to run any scripts as a result of thresholds or in the attempt to gather data that can be used to compromise the system. The only

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

portion of the configuration that is important to the security of the configured environment is the global settings.

The configuration files will be placed in the /var/mrtgcfg directory. The data files will be located in /var/mrtgdata.

Here is an example working configuration file global settings:

```
HtmlDir: /var/www/html/mrtg
ImageDir: /var/www/html/img
LogDir: /var/mrtgdata
Logformat: rrdtool
```

Since the cgi script that is used to display the reports is able to generate reports on any objects within a given configuration file, each customer will have a separate configuration file and, therefore, a different instance of mrtg. The mrtg process needs to be started as a non-root user in order to prevent potential account escalation as a result of modifying scripts that might be run by the mrtg process. Instead, the process is started as the mrtg user as follows:

Create a file called runmrtg with the following in the script:

```
#!/bin/sh
env LANG=C /usr/local/mrtg-2/bin/mrtg abcmrtg.cfg &
env LANG=C /usr/local/mrtg-2/bin/mrtg defmrtg.cfg &
```

Make mrtg the owner of the file and set permissions:

```
mkdir /var/mrtgcfg
mkdir /var/mrtgdata
mkdir /var/www/html
mkdir /var/www/html/mrtg
mkdir /var/www/html/img
chown mrtg /var/mrtgcfg/*
chgrp mrtg /var/mrtgcfg/*
chmod 744 /var/mrtgcfg/runmrtg
chown mrtg /var/mrtgcfg
chgrp mrtg /var/mrtgcfg
chown mrtg /var/mrtgdata
chgrp mrtg /var/mrtgdata
```

Set the script to run every 5 minutes:

```
crontab –u mrtg –e
```

The following lines are added the mrtg's crontab:

```
0-59/5 * * * * /var/mrtgcfg/runmrtg
```

This runs each instance of mrtg once every 5 minutes, which is the default and recommended interval between each data gathering instance.

New configuration files are added simply by adding an additional line point to each new mrtg configuration file.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

For purposes of this document, the following two config files were used:

abcmrtg.cfg

```
HtmlDir: /var/www/html/mrtg
ImageDir: /var/www/html/img
LogDir: /var/mrtgdata
Logformat: rrdtool
PathAdd: /usr/local/rrdtool-1.0.46/bin
LibAdd: /usr/local/rrdtool-1.0.46/lib/perl
Options[_]: growright, bits

EnableIPv6: no

######################################################################
# System: Router
# Description: Cisco Internetwork Operating System Software
#          IOS (tm) 3000 Software (IGS-J-L), Version 11.1(17), RELEASE
SOFTWARE (fc1)
#          Copyright (c) 1986-1998 by cisco Systems, Inc.
#          Compiled Tue 27-Jan-98 12:14 by phester
# Contact: Betty Rubble
# Location: SANSCO
######################################################################

### Interface 1 >> Descr: 'Ethernet0' | Name: 'Et0' | Ip:
'192.168.1.20' | Eth: '00-01
-02-03-04-05' ###

Target[abceth0]: 1:SANSCORocks@192.168.1.20:
SetEnv[abceth0]: MRTG_INT_IP="192.168.1.20" MRTG_INT_DESCR="Ethernet0"
MaxBytes[abceth0]: 1250000
Title[abceth0]: Traffic Analysis for abc -- Router Enet
PageTop[abceth0]: <H1>Traffic Analysis for abc -- Router Enet</H1>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Router in SANSCO</TD></TR>
   <TR><TD>Maintainer:</TD> Betty Rubble<TD></TD></TR>
   <TR><TD>Description:</TD><TD>Ethernet0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD>Et0</TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>1250.0 kBytes/s</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.1.20 ()</TD></TR>
 </TABLE>
```

defmrtg.cfg

```
HtmlDir: /var/www/html/mrtg
ImageDir: /var/www/html/img
LogDir: /var/mrtgdata
Logformat: rrdtool
PathAdd: /usr/local/rrdtool-1.0.46/bin
LibAdd: /usr/local/rrdtool-1.0.46/lib/perl
Options[_]: growright, bits

EnableIPv6: no

######################################################################
# System: Router
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
# Description: Cisco Internetwork Operating System Software
#          IOS (tm) 2500 Software (C2500-D-L), Version 11.3(3), RELEASE
SOFTWARE (fc1)
#          Copyright (c) 1986-1998 by cisco Systems, Inc.
#          Compiled Mon 20-Apr-98 18:46 by phanguye
# Contact: Betty Rubble
# Location: SANSCO
##############################################################################

### Interface 1 >> Descr: 'Ethernet0' | Name: 'Et0' | Ip:
'192.168.1.30' | Eth: '02-03
-04-05-06-07' ###

Target[defeth0]: 1:SANSCORocks@192.168.1.30:
SetEnv[defeth0]: MRTG_INT_IP="192.168.1.30" MRTG_INT_DESCR="Ethernet0"
MaxBytes[defeth0]: 1250000
Title[defeth0]: Traffic Analysis for def -- Router enet
PageTop[defeth0]: <H1>Traffic Analysis for def -- Router enet</H1>
 <TABLE>
   <TR><TD>System:</TD>     <TD>Router in SANSCO</TD></TR>
   <TR><TD>Maintainer:</TD> Betty Rubble<TD></TD></TR>
   <TR><TD>Description:</TD><TD>Ethernet0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD>Et0</TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>1250.0 kBytes/s</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.1.30 ()</TD></TR>
 </TABLE>
```

Configuring Apache

To configure httpd, the /var/www/conf/httpd.conf file needs to be modified. First, the default policy to block all access needs to be configured:

```
<Directory "/">
      Options None
      Order allow,deny
      Deny from all
      Satisfy all
      AllowOverride None
</Directory>
```

Next, we setup the normal directory where standard html files will be located:

```
DocumentRoot "/var/www/html"

<Directory "/var/www/html">
      Options Indexes SymLinksIfOwnerMatch IncludesNOEXEC
MultiViews
      Order deny,allow
      Allow from all
      AllowOverride None
</Directory>
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

In the html directory will be directories created uniquely for each client. With the exception of these client specific index pages, any html pages should be viewable to all users. To implement this security on the html pages, user access will be setup for each of these client specific directories. In this example setup, these files are abc and def fodlers for clients abc and def respectively.

First the passwords need to be setup for these users:

```
mkdir /var/www/etc
touch /var/www/etc/passwd
/var/www/bin/htpasswd -m /var/www/etc/passwd abc
/var/www/bin/htpasswd -m /var/www/etc/passwd def
```

And the folders are created:

```
mkdir /var/www/html/abc
mkdir /var/www/html/def
```

And finally, these folders are configured to require a password:

```
<Directory "/var/www/html/abc">
      AuthType Basic
      AuthName "ABC Client access"
      AuthUserFile "/var/www/etc/passwd"
      Require user abc
</Directory>

<Directory "/var/www/html/def">
      AuthType Basic
      AuthName "DEF Client access"
      AuthUserFile "/var/www/etc/passwd"
      Require user def
</Directory>
```

Next, the cgi access needs to be setup:

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory "/var/www/cgi-bin/">
      Options None
      AllowOverride None
      Order deny,allow
</Directory>
```

Next, the cgi script used to create the reports needs to be reviewed and modified to lockdown which config files can be opened. The crux of the problem with the cgi script being able to view data for other clients has to do with the fact that the config file can be specifically stated by assigned the config file in the url with the directive ?cfg=<filename>. This will override the hard-coded config file also configured in the cgi file. In order to combat this problem, the cgi script found at http://my14all.sourceforge.net/14all-1.1.txt is modified as follows:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Before:

```
if (defined $q->param('cfg')) {
  $l_cfgfile = $q->param('cfg');
  # security fix: don't allow ./ in the config file name
  print_error($q, "Illegal characters in cfg param: ./")
   if $l_cfgfile =~ m'(^/)|(\./)';
  $l_cfgfile = $cfgfiledir.$l_cfgfile unless -r $l_cfgfile;
  print_error($q, "Cannot find the given config file:
\<tt>$l_cfgfile\</tt>")
   unless -r $l_cfgfile;
 } elsif (!$cfgfile) {
  $l_cfgfile = $my14all::meurl;
  $l_cfgfile =~ s|.*\Q$MRTG_lib::SL\E||;
  $l_cfgfile =~ s/\.(cgi|pl|perl)$/.cfg/;
  #$my14all::meurl =~
m{\Q$MRTG_lib::SL\E([^\Q$MRTG_lib::SL\E]*)\.(cgi|pl)$};
  #$l_cfgfile = $1 . '.cfg';
 $l_cfgfile = $cfgfiledir.$l_cfgfile unless -r $l_cfgfile;
 } else {
 $l_cfgfile = $cfgfile;
}
```

After:

```
$l_cfgfile = $cfgfile;
```

This modified file will then be used as the template. Each client will have their own cgi script where the config is hard-coded by configuring the following lines from the cgi script:

```
$cfgfile = '';
$cfgfiledir = '/var/mrtgcfg/';
```

In this case the abc.cgi will be configured with the cfgfile pointed at abcmrtg.cfg and the def.cgi will be configured with the cfgfile pointed at defmrtg.cfg. For convenience in configuring access rights to the appropriate cgi scripts, these cgi scripts are placed in sub-folders in the cgi-bin. This will enable cgi scripts that are intended to be accessible to everyone to be placed in the cgi-bin directory and sub directories are known to be for restricted access.

Setup these cgi files so only the authenticated user can access it:

```
<Directory "/var/www/cgi-bin/abc">
      AuthType Basic
      AuthName "ABC Client access"
      AuthUserFile "/var/www/etc/passwd"
      Require user abc
</Directory>

<Directory "/var/www/cgi-bin/def">
      AuthType Basic
      AuthName "DEF Client access"
      AuthUserFile "/var/www/etc/passwd"
      Require user def
</Directory>
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Finally, it's time to ensure that httpd will run as the intended user (apache) and that file permissions are appropriately set.

```
chown –R apache /var/www
chgrp –R apache /var/www
chmod 400 /var/www/html/*.*
chmod 400 /var/www/html/img/*.*
chmod 400 /var/www/html/mrtg/*.*
chmod 400 /var/www/html/abc/*.*
chmod 400 /var/www/html/def/*.*
chmod –R 500 /var/www/cgi-bin
```

Although this is the default configuration, ensure that the following entries in the httpd.conf file:

```
User apache
Group apache
```

Setup ssl:

```
cd /usr/local/ssl/certs
mount –o remount,rw /usr
openssl genrsa –out server.key 1024
openssl req –new –key server.key –out server.csr
chmod 400 server.key
mount –o remount,ro,nodev /usr
```

(Typically the csr file must be sent off to get a .crt signed certificate from a certificate authority. In this case I issued the command `umask 77 ; openssl req -new -key server.key -x509 -days 365 -out server.crt` to generate the self-signed certificate).

Finally, modify the httpd.conf file as follows:

```
#Listen 80
Listen 443

<VirtualHost _default_:443>
        ServerName mrtg.sansco.com
        DocumentRoot "/var/www/html"
        SSLEngine on
        SSLCertificateFile /usr/local/ssl/certs/server.crt
        SSLCertificateKeyFile /usr/local/ssl/certs/server.key
        SSLCipherSuite HIGH
</VirtualHost>
```

The <Directory> tabs previously discussed are then copied inside of this Virtual Host configuration.

Finally, to ensure that httpd is started with SSL, issue the command "apachectl startssl" and modify the startup script accordingly. The startup script needs to be added to the startup process. After the script has been created and placed into /etc/rc.d/init.d execute the following commands:

```
chkconfig --add httpd
chkconfig --level 345 httpd on
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Design and Implement Ongoing Maintenance Procedures

Overall Plan

In order to ensure the ongoing success of the system, it is important to consider ongoing maintenance procedures. This section covers the following steps:

- Backup Plan
- Patch Maintenance
- Log monitoring
- Tripwire and integrity checking
- Vulnerability checks

Backup Plan

For the most part, the files on this system will remain static. Since the installation is fairly well documented, a disaster situation can easily be addressed by reinstalling the system from scratch. In order to maintain the continued operation of the system, the only files that need to be backed up are any web pages that are generated for providing user friendly access to the reports, the cgi files, the mrtg configuration files and the data files. With the exception of the mrtg data files, it is not anticipated that the other files will change at a high frequency. For these files, a daily backup is more than sufficient. In regards to the data files, the tolerance for loss of data is a business decision. In this case, it's determined that a daily backup of the data files will be sufficient. A script is written to capture the /var/mrtgcfg directory along with the /var/www/html, /var/www/cgi-bin directories and /var/mrtgdata directory is similarly stored in a tar.gz format file on a daily basis. After the scheduled time for these backups has run, an external system is configured to copy the files off via sftp to store the files.

Backup script (backup.sh):

```
#!/bin/sh
cd /var/backup
rm weeklybackup.tar.gz
tar -c /var/mrtgcfg /var/www/html /var/www/cgi-bin /var/mrtgdata
| gzip > mrtgbackup.tar.gz
```

This script is configured to run as the mrtg user as follows:

```
crontab –u mrtg –e
```

And the following line is added:

```
0 2 * * * /var/back/backup.sh
```

Patch Maintenance

Since this system is registered with RHN, as discussed previously in the installation section, it is anticipated that the email address associated with the appropriate entitlement is checked very frequently. Therefore, critical patches released by RedHat will have notifications sent as they become available. In

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

addition to this notification method, the system administrators for SANSCO subscribe to several security notification lists and are generally aware of new vulnerabilities as the become apparent. Since it is noted that httpd, openssh and openssl are running on these systems, these will be watches most closely for potential vulnerabilities since they are the packages most likely to be susceptible to a security event. On a scheduled weekly basis, the MRTG administrator will run through the following checklist:

	Log into the system and run up2date –l and determine is any new packages are available for install
	Visit http://www.openssh.org and determine if there is a newer version than 3.7.1p2
	Visit http://www.openssl.org and determine if there is a newer version than 0.9.7c
	Visit http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/?M=D and determine if there is a newer version of rrdtool than 1.0.46
	Visit http://www.boutell.com/gd/ and determine if there is a newer version of gd than 2.0.20
	Visit http://www.libpng.org/pub/png/src/ and determine if there is a newer version of libpng than 1.2.5
	Visit http://www.gzip.org/zlib/ and determine if there is a newer version of zlib than 1.2.1
	Visit http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pub/ and determine if there is a newer version of mrtg than 2.10.12
	Visit http://www.apache.org/ and determine if there is a newer version of httpd than 2.0.48
	Visit http://www.sendmail.org/ and determine if there is a newer version of sendmail than 8.12.10

Log monitoring

Logwatch is used to monitor the logs. This is convenient since it can process the logs and filter out the noise and send regular reports to the administrator of the system via email. Logwatch was installed during the operating system installation. Information about configuration logwatch can be found at http://www2.logwatch.org:81/tabs/docs/. By configuring logwatch to run on a daily basis and emailing a report to the mrtg system administrator, it is anticipated that the logs will be viewed in a timely manner. A sample report generated from logwatch appears as follows:

```
################### LogWatch 4.3.1 (01/13/03) ####################
       Processing Initiated: Thu Jan 22 13:11:48 2004
       Date Range Processed: yesterday
     Detail Level of Output: 0
         Logfiles for Host: mrtg.sansco.com
 ################################################################

 --------------------- Cron Begin ------------------------
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
**Unmatched Entries**
MAIL (mailed 64 bytes of output but got status 0x004e )

 ---------------------- Cron End -------------------------


------------------ Disk Space --------------------

Filesystem            Size  Used Avail Use% Mounted on
/dev/hda2             981M  108M  824M  12% /
/dev/hda1             101M   14M   81M  15% /boot
/dev/hda6             981M   17M  915M   2% /home
none                  125M     0  125M   0% /dev/shm
/dev/hda3             981M   17M  915M   2% /tmp
/dev/hda5             981M  351M  581M  38% /usr
/dev/hda8              33G  242M   31G   1% /var

 ---------------------- Init Begin -------------------------

**Unmatched Entries**
Trying to re-exec init

 ---------------------- Init End -------------------------

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

su:
   Sessions Opened:
      betty(uid=0) -> mrtg: 3 Time(s)
      betty(uid=501) -> root: 3 Time(s)

login:
   Sessions Opened:
      root: 2 Time(s)

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

 --------------------- Connections (secure-log) Begin -------------------
-----


New Users:
   sshd(74)
    smmsp(500)
    betty(501)
    fred(502)
    wilma(503)
    barnie(504)
    mrtg(505)
    apache(506)

**Unmatched Entries**
groupadd[9630]: new group: name=sshd, gid=74
groupadd[25617]: new group: name=smmsp, gid=500
groupadd[16905]: new group: name=sysadmin, gid=501
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
groupadd[658]: new group: name=mrtg, gid=502
userdel[663]: delete user `adm'
userdel[663]: delete `adm' from group `sys'
userdel[663]: delete `adm' from group `adm'
userdel[663]: delete `adm' from shadow group `sys'
userdel[663]: delete `adm' from shadow group `adm'
userdel[664]: delete user `lp'
userdel[664]: delete `lp' from group `lp'
userdel[664]: delete `lp' from shadow group `lp'
userdel[665]: delete user `shutdown'
userdel[666]: delete user `halt'
userdel[667]: delete user `news'
userdel[667]: delete `news' from group `news'
userdel[667]: delete `news' from shadow group `news'
userdel[668]: delete user `uucp'
userdel[668]: delete `uucp' from group `uucp'
userdel[668]: delete `uucp' from shadow group `uucp'
userdel[669]: delete user `operator'
userdel[670]: delete user `games'
userdel[671]: delete user `gopher'
userdel[671]: remove group `gopher'
userdel[672]: delete user `ftp'
userdel[672]: remove group `ftp'
groupdel[673]: remove group `adm'
groupdel[674]: remove group `lp'
groupdel[675]: remove group `news'
groupdel[676]: remove group `uucp'
groupdel[677]: remove group `games'
groupdel[678]: remove group `dip'
groupdel[679]: remove group `users'
groupdel[680]: remove group `lock'
groupadd[752]: new group: name=apache, gid=503

 ---------------------- Connections (secure-log) End --------------------
-----


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


SSHD Killed: 2 Time(s)

SSHD Started: 3 Time(s)

Users logging in through sshd:
   betty logged in from 192.168.1.100 using password: 3 Time(s)

**Unmatched Entries**
RSA1 key generation succeeded
RSA key generation succeeded
DSA key generation succeeded
 succeeded
sshd -TERM succeeded
 succeeded
sshd -TERM succeeded
 succeeded

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


 --------------------- up2date Begin ------------------------
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
**Unmatched Entries**
[Wed Jan 21 20:37:07 2004] up2date updating login info
[Wed Jan 21 20:37:07 2004] up2date logging into up2date server
[Wed Jan 21 20:37:07 2004] up2date successfully retrieved authentication
token from up2date server
[Wed Jan 21 20:37:16 2004] up2date updating login info
[Wed Jan 21 20:37:16 2004] up2date logging into up2date server
[Wed Jan 21 20:37:16 2004] up2date successfully retrieved authentication
token from up2date server
[Wed Jan 21 20:37:17 2004] up2date availablePackageList from network
[Wed Jan 21 20:39:48 2004] up2date installing packages: ['bash-2.05b-
20.1', 'coreutils-4.5.3-19.0.2', 'glibc-2.3.2-27.9.7', 'glibc-common-
2.3.2-27.9.7', 'glibc-devel-2.3.2-27.9.7', 'gnupg-1.2.1-9', 'initscripts-
7.14-1', 'iproute-2.4.7-7.90.1', 'krb5-libs-1.2.7-14', 'openssl-0.9.7a-
20', 'perl-5.8.0-88.3', 'perl-CGI-2.81-88.3', 'rhpl-0.93.4-1', 'unzip-
5.50-33', 'up2date-3.1.23.2-1']
[Wed Jan 21 20:40:58 2004] up2date Removing packages from package
profile: ['bash-2.05b-20', 'coreutils-4.5.3-19', 'glibc-2.3.2-5', 'glibc-
common-2.3.2-5', 'glibc-devel-2.3.2-5', 'gnupg-1.2.1-3', 'initscripts-
7.13-1', 'iproute-2.4.7-7', 'krb5-libs-1.2.7-8', 'openssl-0.9.7a-2',
'perl-5.8.0-88', 'perl-CGI-2.81-88', 'rhpl-0.93-1', 'unzip-5.50-7',
'up2date-3.1.23-1']
[Wed Jan 21 20:40:58 2004] up2date Adding packages to package profile:
['bash-2.05b-20.1', 'coreutils-4.5.3-19.0.2', 'glibc-2.3.2-27.9.7',
'glibc-common-2.3.2-27.9.7', 'glibc-devel-2.3.2-27.9.7', 'gnupg-1.2.1-9',
'initscripts-7.14-1', 'iproute-2.4.7-7.90.1', 'krb5-libs-1.2.7-14',
'openssl-0.9.7a-20', 'perl-5.8.0-88.3', 'perl-CGI-2.81-88.3', 'rhpl-
0.93.4-1', 'unzip-5.50-33', 'up2date-3.1.23.2-1']
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/bash-
2.05b-20.1.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/bash-
2.05b-20.1.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/coreutils-
4.5.3-19.0.2.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/coreutils-
4.5.3-19.0.2.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
2.3.2-27.9.7.i686.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
2.3.2-27.9.7.i686.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
common-2.3.2-27.9.7.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
common-2.3.2-27.9.7.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
devel-2.3.2-27.9.7.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/glibc-
devel-2.3.2-27.9.7.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/gnupg-
1.2.1-9.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/gnupg-
1.2.1-9.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting
/var/spool/up2date/initscripts-7.14-1.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting
/var/spool/up2date/initscripts-7.14-1.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/iproute-
2.4.7-7.90.1.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/iproute-
2.4.7-7.90.1.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/krb5-libs-
1.2.7-14.i386.hdr
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/krb5-libs-
1.2.7-14.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/openssl-
0.9.7a-20.i686.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/openssl-
0.9.7a-20.i686.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/perl-
5.8.0-88.3.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/perl-
5.8.0-88.3.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/perl-CGI-
2.81-88.3.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/perl-CGI-
2.81-88.3.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/rhpl-
0.93.4-1.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/rhpl-
0.93.4-1.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/unzip-
5.50-33.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/unzip-
5.50-33.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/up2date-
3.1.23.2-1.i386.hdr
[Wed Jan 21 20:40:59 2004] up2date deleting /var/spool/up2date/up2date-
3.1.23.2-1.i386.rpm
[Wed Jan 21 20:40:59 2004] up2date updating login info
[Wed Jan 21 20:40:59 2004] up2date logging into up2date server
[Wed Jan 21 20:41:00 2004] up2date successfully retrieved authentication
token from up2date server
[Wed Jan 21 20:41:00 2004] up2date availablePackageList from network
[Wed Jan 21 20:41:44 2004] up2date installing packages: ['kernel-2.4.20-
28.9']
[Wed Jan 21 20:41:59 2004] up2date Adding packages to package profile:
['kernel-2.4.20-28.9']
[Wed Jan 21 20:42:00 2004] up2date deleting /var/spool/up2date/kernel-
2.4.20-28.9.i686.hdr
[Wed Jan 21 20:42:00 2004] up2date deleting /var/spool/up2date/kernel-
2.4.20-28.9.i686.rpm
[Wed Jan 21 20:42:00 2004] up2date Modifying bootloader config to include
the new kernel info
[Wed Jan 21 20:42:00 2004] up2date Adding 2.4.20-28.9 to bootloader
config
[Wed Jan 21 20:42:00 2004] up2date Installing the kernel via grub
[Wed Jan 21 20:42:00 2004] up2date Running /sbin/grubby --default-kernel
[Thu Jan 22 00:55:18 2004] up2date updating login info
[Thu Jan 22 00:55:18 2004] up2date logging into up2date server
[Thu Jan 22 00:55:18 2004] up2date successfully retrieved authentication
token from up2date server
[Thu Jan 22 00:55:23 2004] up2date updating login info
[Thu Jan 22 00:55:23 2004] up2date logging into up2date server
[Thu Jan 22 00:55:24 2004] up2date successfully retrieved authentication
token from up2date server
[Thu Jan 22 00:55:35 2004] up2date updating login info
[Thu Jan 22 00:55:35 2004] up2date logging into up2date server
[Thu Jan 22 00:55:36 2004] up2date successfully retrieved authentication
token from up2date server
[Thu Jan 22 00:55:37 2004] up2date availablePackageList from network
[Thu Jan 22 02:54:19 2004] up2date Updating package profile
[Thu Jan 22 02:54:21 2004] up2date Updating package profile

 ---------------------- up2date End -------------------------
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

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

Tripwire and integrity checking

In order to have a baseline for comparison in the event that it is suspected that this system has been compromised, tripwire is run on the system before it is put into production. This will give a signature database that can be used for comparison should the need arise.

Tripwire was installed during the initial OS installation and is located in the /etc/tripwire directory. The configuration files, twpol.txt and twcfg.txt, could be modified, but for this environment, the default configuration is fine.

Tripwire is initiated by running the /etc/tripwire/twinstall.sh script. Finally, initialize the database by issuing the command /usr/sbin/tripwire –init. A database file will be created (Wrote database file: /var/lib/tripwire/mrtg.sansco.com.twd). This database file should be moved off the same and stored for use in case of an incident.

To ensure continued integrity assessments, tripwire is scheduled to run daily in cron:

```
crontab –u root –e

0 6 * * * tripwire –check
```

Vulnerabilty checks

To complete the maintenance process, vulnerability scans are run against the system on a weekly basis. It is anticipated that SANSCO has already implemented a nessus server. The scripting for these weekly checks will take place external to this system and so the details of the configuration are not discussed here. Essentially, the nessus server must be kept current with the latest nessus signatures and a complete scan of the system will occur on a weekly basis. Initially, a complete scan (including dangerous) are run against the system. Since the system is not expected to be accessed on weekends, the regularly scheduled scans will happen automatically every Saturday. On a quarterly scheduled basis, the dangerous scans will be run manually against the system to help mitigate against potential DoS conditions.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

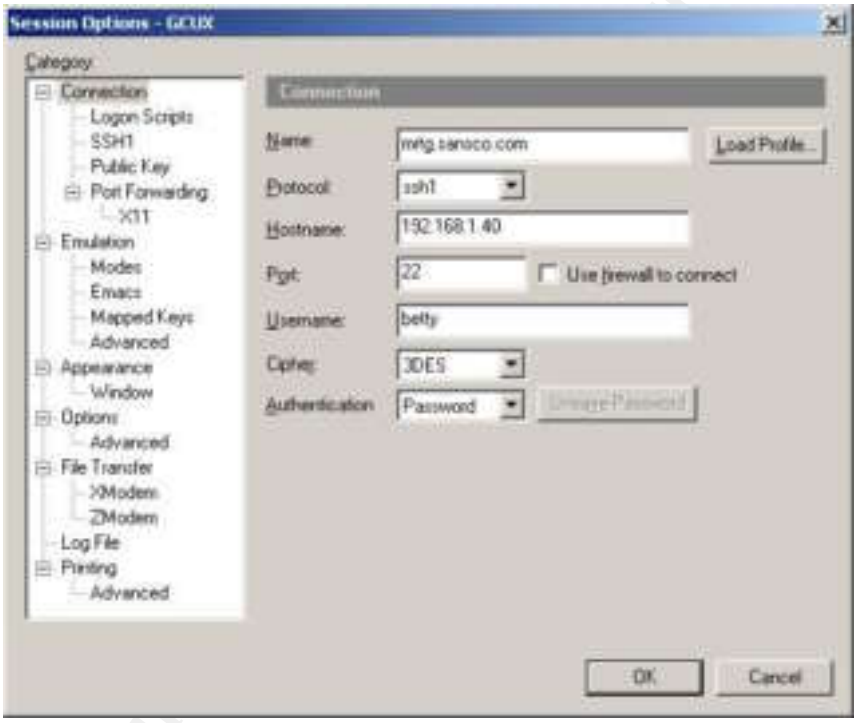
Test and Verify the Setup

Verifying SSH configuration

There are several things we want to verify with SSH. They are as follows:

- SSHv1 connections are not permitted
- Root can not login via SSH
- Login banner is working

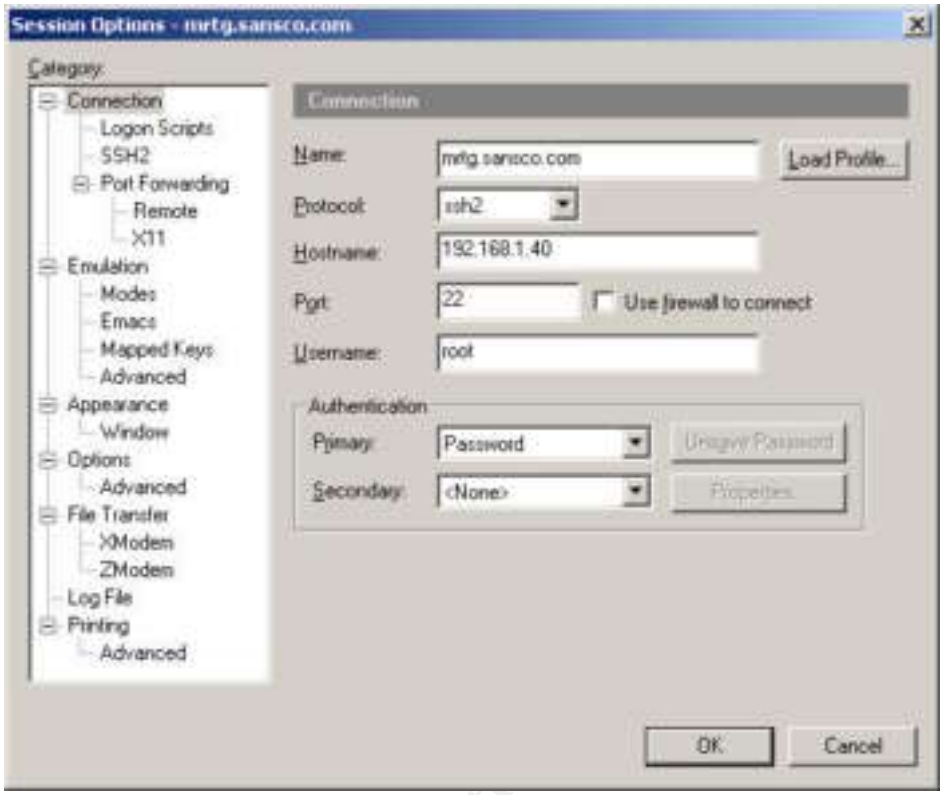
To test these items, SecureCRT is used. To test if the system will accept an SSHv1 connection, a host is configured in SecureCRT as follows:



Next, a connection is attempted and we expect to see the following window:

The configuration is then changed to SSHv2 and the user is changed to root in order to test if root has the ability to login via SSH:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



The root user should be treated as an invalid username/password combination and the following screen will appear:

Next, we'll configure to connect as betty and verify the sshbanner is displayed:

Additional steps could be taken to verify the configuration of the SSH server, such as testing to ensure RSA keys are disabled, testing X11 port forwarding, etc…

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Checking httpd configuration

There are several things that need to be checked for the http server. They are as follows:

- User authentication is required for the https://192.168.1.40/abc directory
- User authentication is required for the https://192.168.1.40/def directory
- User authentication is required for the https://192.168.1.40/cgi-bin/abc directory
- User authentication is required for the https://192.168.1.40/cgi-bin/def directory
- Verify that the modified cgi scripts can not access configuration scripts for another client

To test for user authentication requirements for the https://192.168.1.40/abc directory, we simply browse to that directory. We should be prompted with the following login:

To perform the test we first enter a known non-existent user name and password. This should fail to login and we will see the same screen as above again. Next we login with the def account. Again, we should fail. Finally, we login as the abc user and access should be permitted.

To test for user authentication requirements for the https://192.168.1.40/def directory, we repeat the steps as above, only reverse the usage of the abc and def account when testing. For this directory, we should be prompted with the following login prompt:

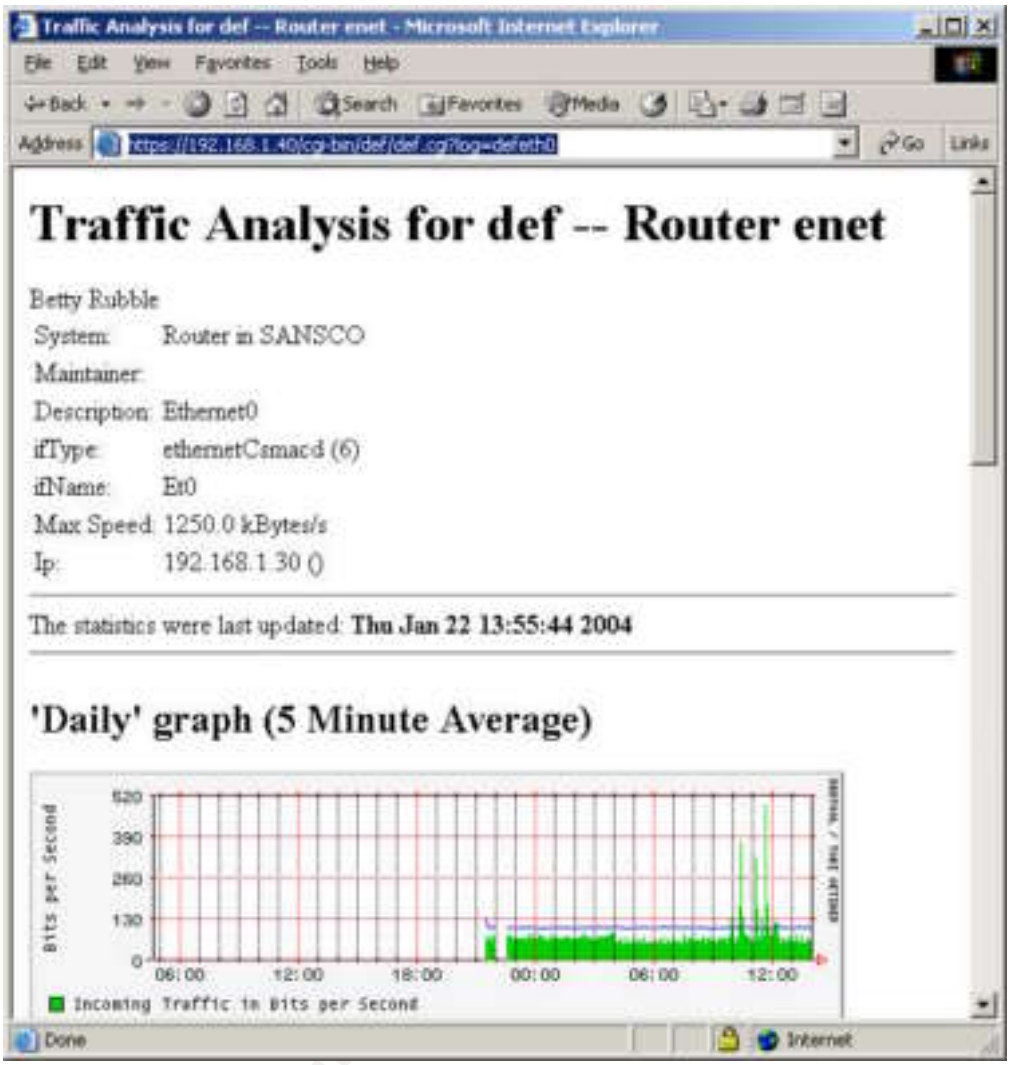
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Next we repeat the steps for the https://192.168.1.40/cgi-bin/abc/abc.cgi and https://192.168.1.40/cgi-bin/def/def.cgi sites using the same methodolofy explained above.

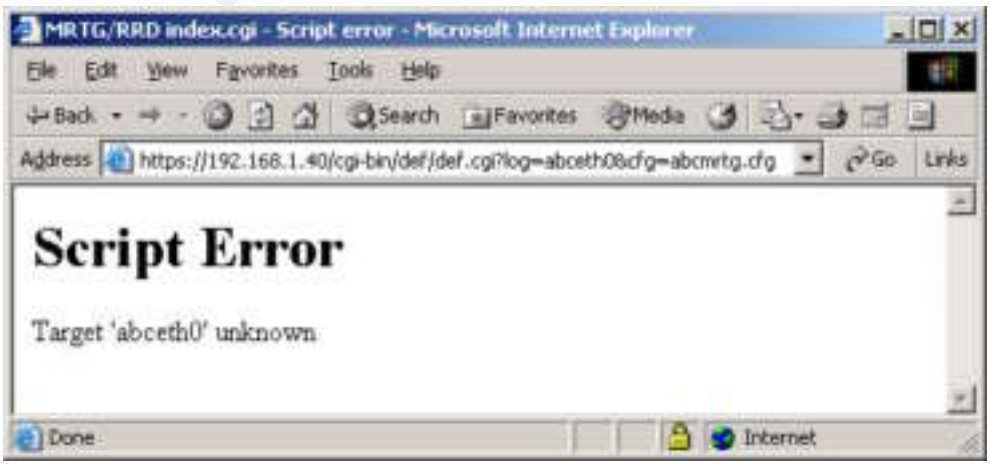
Finally, we wish to test to verify that the modification to the cgi script works as expected. The big deal with a client being able to access another client's data is the ability to use ?cfg=<config file> directive in the url path. To test that this ability has been disabled, we will browse to the url: https://192.168.1.40/cgi-bin/def/def.cgi?log=defeth0 as displayed on the next page.

© SANS Institute 2004, Author retains full

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

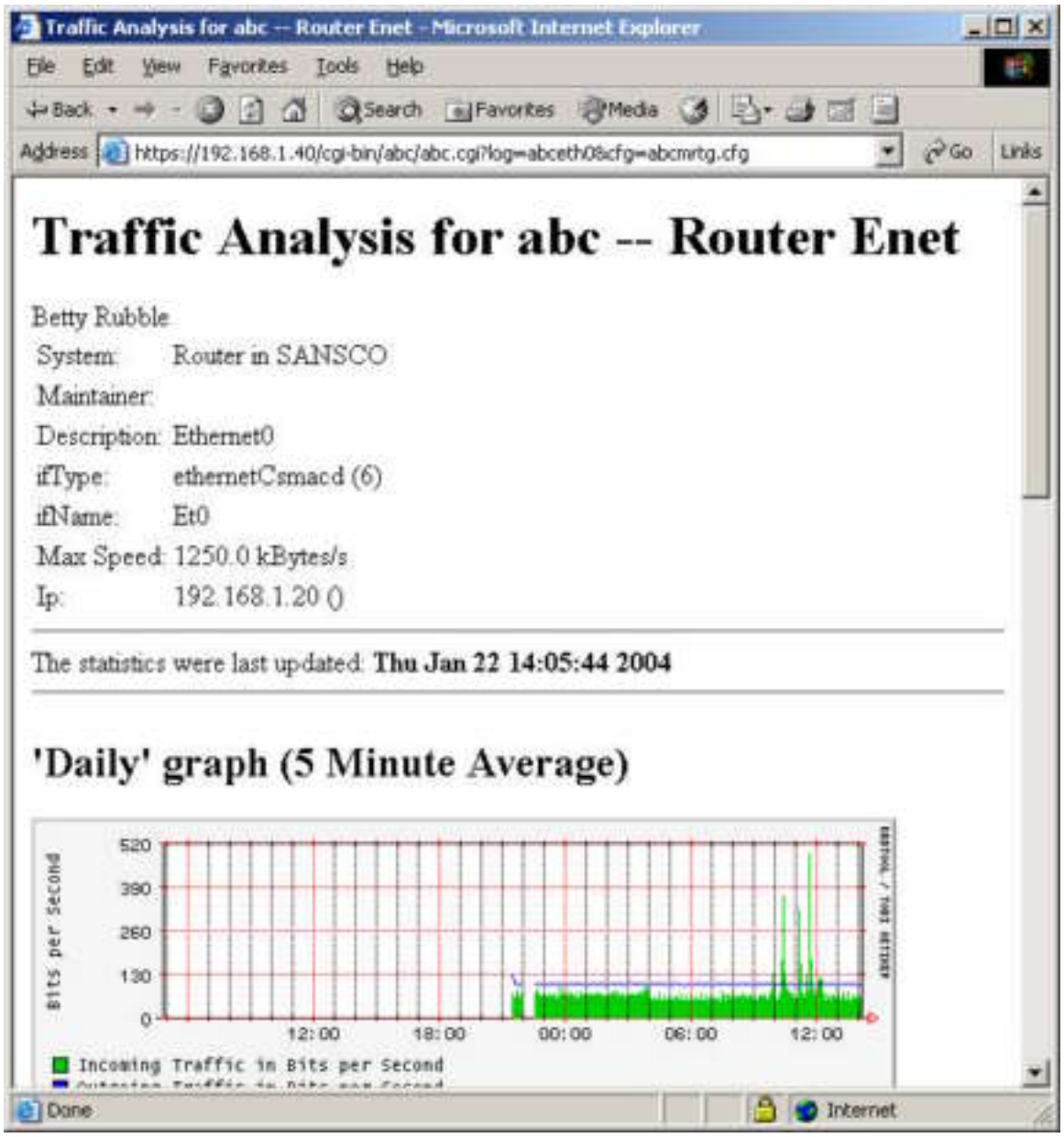


Next, the url path is manually modified as follows: https://192.168.1.40/cgi-bin/def/def.cgi?cfg=abcmrtg.cfg&log=abceth0. The following screen is displayed:



© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Since the def.cgi script is hard-coded to only open the defmrtg.cfg file, the abceth0 MRTG target is not found and so the message above is displayed. To ensure that the syntax is correct, the url is modified to read: https://192.168.1.40/cgi-bin/abc/abc.cgi?log=abceth0&cfg=abcmrtg.cfg. The following screen is displayed after logging in as the abc user:



Verifying boot loader password protection

In order to verify that the boot loader password is functioning as expected, reboot the system. On the Grub boot loader screen, it is necessary to type “p” and then enter the boot loader password entered during OS installation in order to modify the boot command.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Verify Single User Shell password protection

Since we've reboot the system to test that the boot loader password works, we will configure the system to enter single-user mode. This is accomplished by adding the keyword single to the end of the boot command. We expect to be prompted to login with the root password in order to complete the boot sequence. If we are prompted with the password to enter "maintenance" mode, then the modification to require a password for a single user shell is successful.

Checking file system options

To check and verify that the file system was mounted as intended, the mount command is issued with no options. The output should display as follows:

```
/dev/hda2 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/hda1 on /boot type ext3 (ro,nodev)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda6 on /home type ext3 (rw,nosuid,nodev)
none on /dev/shm type tmpfs (rw)
/dev/hda3 on /tmp type ext3 (rw,nosuid,nodev)
/dev/hda5 on /usr type ext3 (ro,nodev)
/dev/hda8 on /var type ext3 (rw,nosuid,nodev)
```

Simply compare this against the expected configuration form the /etc/fstab file.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

References

1. Andreasson, Oskar “The conf/ variables” URL: http://ipsysctl-tutorial.frozentux.net/chunkyhtml/theconfvariables.html#AEN612 (January 2004)

2. “Boot Loader Configuration” URL: http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/s1-x86-bootloader.html (January 2004)

3. “CERT Advisory CA-2003-24 Buffer Management Vulnerability in OpenSSH” URL: http://www.cert.org/advisories/CA-2003-24.html (January 2004)

4. “CERT Advisory CA-2003-25 Buffer Overflow in Sendmail” URL: http://www.cert.org/advisories/CA-2003-25.html (January 2004)

5. “CERT Advisory CA-2003-26 Multiple Vulnerabilities in SSL/TLS Implementations” URL: http://www.cert.org/advisories/CA-2003-26.html (January 2004)

6. Chau, Jacquiline ”Hardening a Red Hat Linux Apache Web Server with Snort Installed” URL: http://www.giac.org/practical/GCUX/Jacqui_Chau_GCUX.pdf (January 2004)

7. “Configuration” URL: http://www.redhat.com/docs/manuals/RHNetwork/ref-guide/2.8/up2date-config.html#UP2DATE-CONFIG-TEXT (January 2004)

8. “GD Graphics Library” URL: http://www.boutell.com/gd/ (January 2004)

9. “Index of /pub/png/src” URL: http://www.libpng.org/pub/png/src/ (January 2004)

10. Oetiker, Tobias “doc/mrtg-reference” URL: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-reference.html (January 2004)

11. Oetiker, Tobias “doc/mrtg-unix-guide” URL: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg-unix-guide.html (January 2004)

12. Oetiker, Tobias “Index of /~oetiker/webtools/mrtg/pub” URL: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pub/ (January 2004)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

13. Oetiker, Tobias “RRD TOOL -- RRDtool Download “ URL: http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/download.html (January 2004)

14. Oetiker, Tobias “Index of /~oetiker/webtools/rrdtool/pub” URL: http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/?M=D (January 2004)

15. “OpenSSH” http://www.openssh.org (January 2004)

16. “OpenSSH Portable Release for Linux/Solaris/etc” URL: http://www.openssh.org/portable.html (January 2004)

17. “OpenSSL: The Open Source toolkit for SSL/TLS” URL: http://www.openssl.org
18. http://www.openssl.org/source/ (January 2004)

19. Roelofs, Greg “zlib Home Site” URL: http://www.gzip.org/zlib/ (January 2004)

20. “Sendmail Home Page” URL: http://www.sendmail.org/ (January 2004)

21. URL: http://my14all.sourceforge.net/14all-1.1.txt (January 2004)

22. “Welcome! - The Apache Software Foundation” URL: http://www.apache.org/ (January 2004)

23. “www.logwatch.org” URL: http://www2.logwatch.org:81/tabs/docs/ (January 2004)

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.