Global Information Assurance Certification Paper

Deploying Nagios Monitoring Services

on Secured
Red Hat Enterprise Linux 3
Environment

GCUX
Practical Assignment
v. 3.0

Option 1
Securely Administering UNIX

Alexey Rogozhkin

03/10/2005

Table of Contents

List of Figures

Abstract

This paper describes how to securely deploy and maintain Nagios monitoring services in big enterprise environment - fictitious company GIAC Enterprises.

> *Nagios is a host and service monitor designed to inform you of network problems before your clients, end-users or managers do. The monitoring daemon runs intermittent checks on hosts and services you specify using external "plugins" which return status information to Nagios. When problems are encountered, the daemon can send notifications out to administrative contacts in a variety of different ways (email, instant message, SMS, etc.). Current status information, historical logs, and reports can all be accessed via a web browser.*[1]

The monitoring of IT infrastructure is highly visible and important component of any company. Solid and nonrepudiate monitoring process can improve service availability, increase security, and enforce the following of the Service Level Agreements with customers and service providers.

This paper will provide details of deployment of multiple software components collectively called "Nagios server" in secure and manageable bundle. Nagios server includes Nagios core services to provide actual monitoring, mail server to sends alerts, and HTTP server to display monitoring status and control all service parameters.

Nagios services will be installed on the already secured single Red Hat Enterprise Linux 3.0 server.

This paper assumes that the software will be deployed and hardened by a security professional, although all maintenance and update procedures come to hands of regular system administrators.

Document Conventions

When you read this practical assignment, you will see that certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

`command`	Operating system commands are represented in this font style. This style indicates a command that is entered at a command prompt or shell.
`filename`	Filenames, paths, and directory names are represented in this style.
`computer output`	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.
Quotation	A citation or quotation from a book or web site is in this style.

[1] Nagios monitoring program <http://www.nagios.org/about.php> [7]

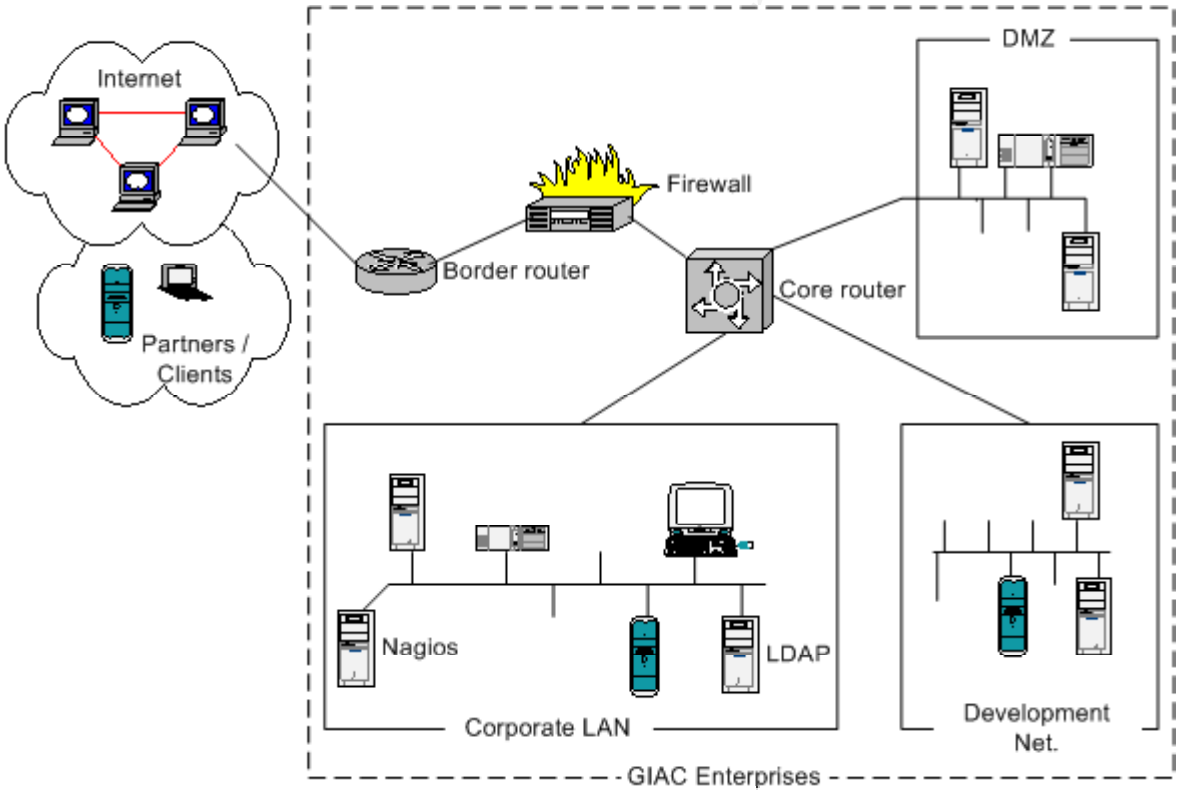
1. Environment

Nagios monitoring server will operate in the secured corporate network. The network is protected by the border firewalls. All computers in corporate network use IP addresses from the private IP space. The computers can access the Internet, although are not visible directly from it.

Nagios control interfaces should be accessible from the Corporate LAN. Access for mobile employee or external customers (HTTPS through the application firewall in the DMZ) can be considered later.

Nagios server will monitor entire GIAC Enterprises IT infrastructure including servers, printers, networking equipment, and devices in the special use subnets.

Figure 1: Network diagram



1.1. Physical Security

Server is located in the secured server room with access limited to authorized system and network administrators. Servers and terminals are locked in the racks and only system administrators and management have keys to access them.

Network devices and cables are managed by the networking team. Server room has redundant cooling systems and additional power sources.

All environmental risks are mitigated to acceptable level.

1.2. Hardware

A hardware specification of the server is:

- Model: IBM xSeries x335 server
  - CPUs: 2 x 3.0GHz Xeon, Hyperthreading is turned on;
  - RAM: 2GB;
  - HDDs: 2 x 35GB SCSI; organized in RAID1 logical volume;
  - NICs: 2 x Built-in Gigabit Broadcom cards.

1.3. Operation System

GIAC Enterprises use System Imager[2] server to build new production servers using the unified "golden" images. The "golden image" is based on Red Hat Enterprise Linux AS release 3 Update 4 and has 245 original Red Hat's RPMs. Complete list of installed components can be found in Appendix A. The image has reasonable minimum of components patched recently, and their security settings conform to local security policies and guidelines.

All existing network listening services (e.g. `SSH`, `NTP`) are hardened properly, and after OS deployment process is completed the server can be connected to the office network.

New system comes with CD-R containing `AIDE`[3] integrity checking tool and reference `AIDE`'s database for the "golden image".

1.3.1. Networking settings

```
HOSTNAME=nagios.example.com    DOMAINNAME=example.com
IPADDR="10.0.2.133"            GATEWAY="10.0.0.252"
NETMASK="255.255.0.0"          NETWORK="10.0.0.0"
```

Local firewall (`iptables`) is installed and activated on the server and allows any established and outgoing network traffic, but blocks any new incoming traffic except for `SSH(TCP/22)` and `NTP(UDP/123)`. GIAC Enterprises does not use `IPv6` currently.

1.3.2. Filesystem Layout

FS mount point	size	mount options	FS type
/	640MB	rw	ext3
/boot	128MB	rw,nosuid,nodev,noatime	ext3
/var	2GB	rw,noexec,nosuid,nodev,noatime	ext3
Swap	2GB	n/a	swap
/tmp	2GB	rw,nosuid,nodev	ext3
/usr	4GB	rw,nodev,noatime	ext3

[2] System Imager automates Linux installs and software distribution <http://www.systemimager.org> [16]
[3] AIDE, version 0.10. <http://sourceforge.net/projects/aide> [18]

/home	512MB	rw,nosuid,nodev	ext3

The filesystem layout and directory structures perfectly fit to current FHS standard[4] and also allow applying granular access limitations for each disk partition.[5]

Newly deployed server has a disk drive with bigger capacity than required by OS components. Unallocated disk space can be used to create filesystems for third party applications and data.

1.4. Pre-deployment System Audit

To better describe the operational environment and to verify our assumptions about security of the server, a few assessment tools will be used. This audit will also help to identify vulnerabilities which were ignored during server initial deployment because no one thought about the specific role the new server will play.

1.4.1. System Security Rating by CIS Benchmark Tool

CIS Security Benchmark Checker[6] was chosen to provide local system settings security assessment.

Although the current version of this tool[7] does not officially support Red Hat Enterprise Linux release 3, it can be used after minor modification[8].

Report about possible security problems is included in the Appendix B.

GIAC Enterprises site security policy does not allow keeping security assessment tools on the mission critical servers. So, when the software deployment is verified, this tool will be uninstalled.

1.4.2. Network Services Survey

1.4.2.1. Listening ports

Pre-deployment check of network services have confirmed that only specified ports UDP/123 (ntp) and TCP/22 (ssh) were listening to the network:

```
lsof -i -P -n

COMMAND   PID USER   FD   TYPE  DEVICE SIZE NODE NAME
sshd      883 root    3u  IPv4    1969       TCP *:22 (LISTEN)
ntpd    16593  ntp    4u  IPv4 2578258       UDP *:123
ntpd    16593  ntp    5u  IPv4 2578259       UDP 127.0.0.1:123
ntpd    16593  ntp    6u  IPv4 2578260       UDP 10.0.2.133:123
```

This survey also confirms that no existing system process is occupying ports which will be used by new software (ports `TCP/25,80,389,433,5666`).

[4] <http://www.pathname.com/fhs/> [2]
[5] See chapter “My considerations on partitioning scheme”. Simon Ostengaard [6]
[6] CIS Security Benchmark Checker. CIS-Scan. <http://www.cisecurity.org> [1717]
[7] Current CIS benchmark tool Version for Linux is 1.4.2-1.0 (as of January 2005)
[8] For modification details see [8].

1.4.2.2. Test by Nessus[9]

The test was run for all ports in range 1-65365, with all plugins enabled, including those marked as "unsafe".

The significant findings are:

```
     Number of security holes found: 1
     - You are running a version of OpenSSH which is older than
3.7.1.

     Number of security warnings found: 2
     - The remote host does not discard TCP SYN packets which
have the FIN flag set.

     - The remote host answers to an ICMP timestamp request. This
allows an attacker to know the date which is set on your machine.
```

1.4.3. Summary of Security Assessment Results

Results of pre-deployment tests confirm that level of hardening of the server is acceptable and conforms to GIAC Enterprises security policies. However, because of importance of Nagios services which makes them highly probable target for attacks, the server could be tightened more, to close some of the identified vulnerabilities. The following groups have been identified by the assessment:

1: Real risks. Despite of little possible security exposure in the most of them, it does not hurt to make them fixed:

- kernel's network options (CIS)
- ICMP timestamp messages (Nessus)
- TCP packets with *ACK,FIN* options (Nessus)

2. Questionable risks. Hardening of these items has little sense in the current operational environment, but can negatively affect operational procedures, productivity, monitoring, and service availability:

- restrictions for single user mode in the /etc/inittab file (CIS)
- "gpm not deactivated" (CIS)
- SUID bit of /bin/traceroute file (CIS)

3. Obvious "False Positives":

- "vulnerability" in the SSH daemon[10] (Nessus)

2. Pre-deployment Security Hardening

The following procedures include risk mitigation for the items identified in the previous chapter, and some additional security hardening required for the new applications.

[9] Nessus. v. 2.2.2a for Linux[9] with latest plugins [20]

[10] Red Hat components may have "backported" security fixes, which do not reflect mainstream version numbers of the corresponding software package.

2.1. Networking Settings Hardening

- Kernel parameters for the networking have been modified by editing `/etc/sysctl.conf` file. Context of this file is included in Appendix G.
- Added protection against `SYN/FIN` attack (`Nessus ID : 11618`) by modifying firewall rules in the `/etc/sysconfig/iptables` file:

```
-A INPUTCHAIN -p tcp --tcp-flags SYN,FIN SYN,FIN -m state \
--state NEW -j REJECT --reject-with tcp-reset
```

- Added protection for ICMP timestamp information leak (`CVE : CAN-1999-0524`) in the `/etc/sysconfig/iptables` file:

```
-A INPUTCHAIN -p icmp -m icmp --icmp-type 13 -j DROP
-A INPUTCHAIN -p icmp -m icmp --icmp-type 14 -j DROP
```

2.2. Stack Protection

In the recent kernel,[11] new features are available to mitigate some risks of the Buffer Overflow exploits:

> *The Red Hat Enterprise Linux 3 Update 3 kernel includes a new security feature known as Exec-shield. Exec-shield is a security-enhancing modification to the Linux kernel that makes large parts of specially-marked programs — including their stack — not executable. This can reduce the potential damage of some security holes, such as buffer overflow exploits.*
>
> *Exec-shield can also randomize the virtual memory addresses at which certain binaries are loaded. This randomized VM mapping makes it more difficult for a malicious application to improperly access code or data based on knowledge of the code or data's virtual address.*[12]

To ensure this protection is active, the following lines have been added to the `/etc/sysctl.conf` file[13]:

```
kernel.exec-shield-randomize = 1
kernel.exec-shield = 1
```

3. New Software to Deploy

To deploy fully functioning central monitoring Nagios server, the following main components must be installed on the server:

- **Nagios engine** *"is a host/service/network monitoring program written in C and released under the GNU General Public License. CGI programs are included to allow you to view the current status, history, etc via a web interface if you so desire"*.[14]

[11] E.g. 2.4.20-1.0.2ENT

[12] < http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/release-notes/as-x86/RELEASE-NOTES-U3-x86-en.html> [11]

[13] For more information about stack protection read the chapters "Testing and Validation" and "Summary and Research" in this paper.

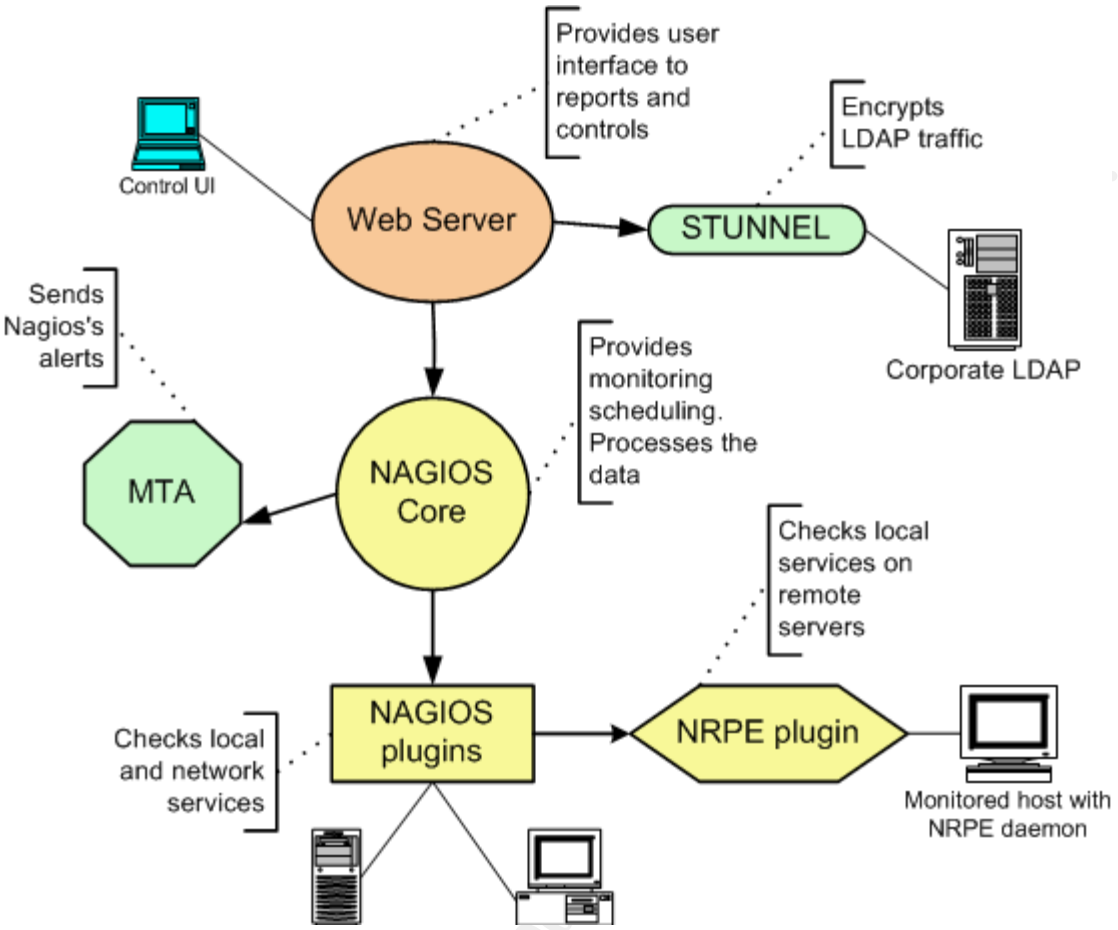
[14] Nagios 2.0 README [15]

- **Nagios plugins** provide actual checks for the Nagios engine. They able to check: disk space, memory utilization, SNMP status, check DNS services, etc.
- **Web server** provides GUI interface to Nagios engine. It uses SSL plugin to encrypt traffic and LDAP plugin to authenticate users using corporate LDAP server.
-
- **Stunnel** will be used to connect LDAP module of the Apache server to the corporate LDAP server over SSL tunnel[15]
- **MTA (mail server)** will send Nagios's notifications
- **NRPE (Nagios Remote Plugin Extender)** *"is a plugin that is run on the Nagios host and is used to contact the NRPE process on remote hosts. The plugin requests that a plugin be executed on the remote host and wait for the NRPE process to execute the plugin and return the result. The plugin then uses the output and return code from the plugin execution on the remote host for its own output and return code"*[16].

The following draft shows component dependencies, control and data flows which will help to identify requirements for application security hardening.

Figure 2: Software components and dataflow

[15] Support for LDAP over SSL may be also done by Netscape SDK, if these libraries are installed on the server [9]. I selected `stunnel` because it is small, secure, and simple.
[16] NRPE 2.0 README [21]



Benefits of Nagios monitoring already have been described in the Introduction; however, security benefits not identified yet, are:

- It can monitor network security devices such as surveillance cameras, NIDS agents, and firewalls.
- It can serve as abnormality monitor for different services.
- It can integrate plugins with direct security application: NMAP[19], Prelude[17] monitoring, system logs monitoring, etc.
- It provides accurate events history to enforce accountability and change control procedures.

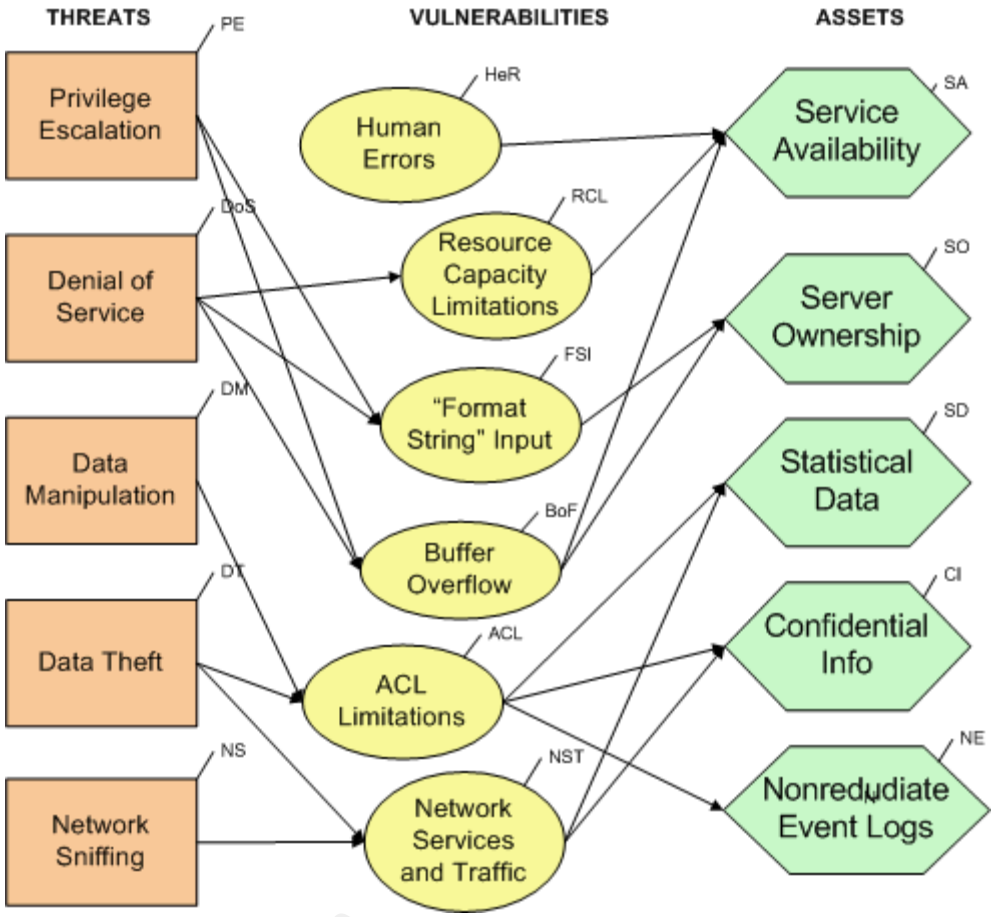
Most important quality of the Nagios server is ability to monitor thousands of various services, and instantly notify right persons about any problems. That's why application availability is most critical asset to be protected.

3.1. Risks Introduced by New Software

The risks, associated with the new software deployment are presented on the following picture:

[17] <http://www.prelude-ids.org/>

Figure 3: Risk model for the new software



Assets are valuable data and services which should be protected. Most of them are introduced by the new software, but some (like system passwords) are generic for any system.

- Service Availability `{SA}` is ability of Nagios server to monitor services and hosts, and an ability to make appropriate actions (e.g. send alerts via e-mail) immediately.
- Server Ownership `{SO}` defines who controls the `root` account on the server.
- Statistical Data `{SD}` are performance logs, network topology, usage patterns, maintenance schedules (e.g. typical maintenance windows for company's NIDS servers), etc. This information could be used by intruder to plan attacks against local and remote servers in the company.
- Confidential Information `{CI}` includes access passwords, login names, e-mails, shared secret keys, SNMP community names, etc.
- Nonrepudiate Event Logs `{NE}` are important for personnel accountability[18] , and for controlling the Service Level Agreements (SLAs).

[18] Usually, the problem reported by Nagios must be resolved or acknowledged by someone within pre-defined time frame or else an escalation alert message will be sent to the management.

Threats are conditions which potentially cause harm to assets.

- Privilege Escalation {**PE**} is a threat associated with limitations or wrong implementation of access control methods. In short, privilege escalation occurs when some user of process gets resources which are not meant to be available to him.
- Denial of Service {**DoS**} threatens the ability of Nagios to notice or report the problem. An intruder may want to cause {DoS} of Nagios services to hide his attack on another monitored host.
- Data Manipulation {**DM**} threat arise when somebody wants to change or erase some data. Most probable targets of data manipulations are: Nagios logs and service availability history.
  Possible reasons to do it: negligent system administrator who did not fix a problem in reasonable time frame and is now trying to hide this fact. Or, an attacker who wants to eliminate any possible traces of attack against some host, including performance abnormality or service restarts, which usually are noticed by Nagios. The analysis of this threat must address as well as local data manipulation, also the risk of remote data manipulation by means of exploiting vulnerabilities in the remote monitoring Nagios agents.
- Data Theft {**DT**}. Possible target is Confidential Information **{CI}**.
- Network Sniffing {**DS**}. The usual mitigation of this threat is data encryption.

Vulnerabilities are characteristics of assets or assets' environment which could be exploited by the treat.

- Buffer Overflow {**BoF**}. Most probable targets: CGIs and Nagios plugins.
- Human Errors {**HeR**}. Despite no threat targets this vulnerability directly, human errors may contribute to asset damage more than an intentional attack. Unusual security hardening, custom components, and complicated update procedures are main sources of human errors. The main countermeasures are: properly documented security hardening and change control procedures.
  Another specific target for this vulnerability is programmatic errors in the underdeveloped software components, e.g. in the custom Nagios plugins.
- Resource Capacity Limitations {**RCL**}. Everything good is limited. Most possible targets of this vulnerability are: CPU resources, network sockets, and human ability to read numerous alerts send by Nagios.
- "Format String" Input {**FSI**}. This vulnerability was named after famous cases of misuse of `printf()` function; but in this paper, I will also put under this category the related vulnerabilities: careless use of input parameters in the scripts, and problems with non-sanitized program environment. Most probable targets: Nagios plugins.
- ACL Limitations {**ACL**}. Existing security model and its implementation make proper access separation difficult or not efficient [4]. Setting of resource access permission and ownership may not be sufficient, and other means like "chroot" environment should be considered to mitigate this vulnerability.
- Network Services and Traffic {**NST**}. Possible targets: any component listening

`to or initiating a network connection. New network traffic also bears sensi`tive information and has valuable statistical characteristics.

The associated abbreviations (e.g. `{PE}` for "Privilege Escalation") are the "risk codes" which is used in this document to identify the risk which is mitigated by appropriate security measure.

For example, process of mitigating treat "Deny of Service" for asset "Service Availability", where application vulnerability is "Buffer Overflow" will have risk code `{DoS->BoF->SA}`.

Any security hardening without corresponding risk code should be considered as a following common good system administration and security practice.

---

4. Software Deployment and Hardening

The following components were chosen to provide Nagios monitoring services:

- **Apache HTTP server**. Version 2.0.46-44.ent.
  - SSL support components for HTTPS (**`mod_ssl`**). Version 2.0.46-44.ent.
  - LDAP support components (**`mod_authz_ldap`**). Version 0.22-5.
    - **stunnel**. Version 4.04-4.
- **Postfix mail server**. Version 2.0.16-14.RHEL3.
- **Nagios engine**. Version 2.0b1-1.
  - **Nagios plugins.** Version 1.4-1.
    - **NRPE** (plugin and daemon). Version 2.0-3.dag.

To increase application availability and simplify maintenance procedures, standard software RPMs packages supplied with Red Hat Enterprise Linux have been used whenever it was possible. Custom-built applications usually lack of proper testing and support, and also have complicated maintenance and update procedures.

The server must not have any development packages[19] or `C/gcc` compilers. This has some sense for security, but has much bigger value for services availability and version control. All third-party packages were built in RPM form on the separate "build" server. Then compiled RPM files were installed on the Nagios server. For compilation and install procedures please check Appendix C.

The first step of application bundle hardening will be the same as in hardening of any OS - disabling the components which are not required for the specified functionality.

The second step will be the mitigation of the identified risks by using technical means such as ACL, chroot, encryption, etc.

[19] ... unless they are required for the installed applications. Example: `make` utility, which is used to create/update some configuration files.

4.1. Securing Applications by Disabling Useless Components

We will not try to erase any single useless component because on the next RPM update it will reappear in the same place again. Instead, we will disable them whenever possible via proper configuration options.

4.1.1. Apache HTTP Server

At first step of securing web server, we will reduce number of components (loadable modules) to minimal acceptable level.
According to the requirements for the HTTP server, it must:

- Execute CGIs;
- Keep detailed connection and access logs;
- Provide access control, authentication, and authorization;
- Encrypt traffic via SSL;
- Use "`mod_rewrite`" to redirect requests to SSL before authentication;
- Authenticate users via LDAP.

Everything else can be disabled. To do it, comment out appropriate `LoadModule` directives in the `httpd.conf` file. The resulting Apache HTTP configuration files are included in the Appendix D.

Possible implications of this security measure: None. Some performance improvement is expected due to resulting smaller application memory footprint.

4.1.2. Postfix Mail Server

The default configuration of the current version[20] of Postfix has good settings. The server is not listening to incoming connections from the external network interfaces. It is impossible to restrict the MTA role to act as "null-client" because Nagios server should not rely on any "mail relay" server to send alerts. However, despite almost perfect default configuration files, some additional hardening can be done:

- Disable components we won't use:
  - Interfaces to additional non-Postfix software (e.g. `Cyrus` or `uucp`)
- Disable transports and services which won't be used:
  - `lmtp`, `virtual`, `relay`

The resulting `master.cfg` configuration file is included in the Appendix F.

4.2. Securing Applications via ACL, Chroot, Encryption, and Privilege Separation.

4.2.1. SSL

For this application, the encryption quality can be improved by specifying additional random sources in the configuration file (risk code is `{NS->NST->CI}`[21]).

[20] From the RPM `postfix-2.0.16-14.RHEL3`
[21] For the "risk codes" please refer to *Figure 3: Risk model for the* new software on page 8.

> *The drawback* [of using default "builtin" method] *is that this is not really a strong source and at startup time (where the scoreboard is still not available) this source just produces a few bytes of entropy. So you should always, at least for the startup, use an additional seeding source.*[22]

I left "builtin" PRNG for new connections for better performance, but added `/dev/urandom` for startup PRNG generation:

```
SSLRandomSeed startup file:/dev/urandom 1024
```

Possible implications of this security measure: Low -> 1.5 second delay during Apache server startup.

4.2.2. Stunnel

To reduce the damage which could be done if some vulnerability in this application is exploited, the applications should run in chrooted environment under non privileged system ID "`stunnel`" `{PE->BoF->SO}`.

Hardening procedure

1. Create new user account "`stunnel`":
   ```
   useradd -d /var/run/stunnel -r -s /sbin/nologin stunnel
   ```
2. Lock the account: `passwd -l stunnel`
3. Change the ownership of the run and log directory, setup proper access permissions `{DT->ACL->CI}`:
   ```
   chown -R stunnel.stunnel  /var/run/stunnel
   chmod 700 /var/run/stunnel
   ```
4. Make `stunnel` to run chrooted. Only configuration change is required; the complete configuration file `/etc/stunnel/stunnel.conf` will be:

```
chroot = /var/run/stunnel/
# PID is created inside chroot jail
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel
client = yes
debug = 2
output = /var/run/stunnel/stunnel.log
[ldaps]
    accept  = 127.0.0.1:389
    connect = ldap.example.com:636
```

Possible implications of these security measures: None.

4.2.3. Apache HTTP Server

Good countermeasure against various attacks to HTTP server, i.e. risk `{PE->BoF->CI}`, is to run server in chrooted environment.

[22] Apache Module mod_ssl. <http://httpd.apache.org/docs-2.0/mod/mod_ssl.html> [9]

The usual method of making chrooted environment for Apache HTTP consists of copying directory structures, libraries, and binaries into the target chroot directory. Unfortunately, this creates problems for any software update. E.g., during the update of any OS component, like `imbd`, the system administrator must decide if he has to update files and links in the chroot directory or not. Even tools like `ldd` cannot ensure the chrooted service will survive the update. Various Nagios CGI components which would run under the same chroot also add the complexity.

The method used in this paper[23] utilizes links from binary components and libraries to the chroot directory via "`--bind`" mount option which is supported by current Linux kernels. The update procedures will be as simple as it was without any chroot. The tradeoff of this approach: although the chroot environment which is built using "`--bind`" mounts will have only minimum of required binaries, it will have more libraries than is required.

To understand which components must be chrooted, check what constitutes the installed HTTP packages by running command:

```
rpm -ql httpd mod_authz_ldap mod_ssl
```

Files included in `httpd` package can be classified to the following functional groups:

Group	Top level directory	Example of the file	chroot
Configuration files	`/etc/http`	`/etc/httpd/conf/httpd.conf`	Yes
Logs	`/var/log`	`/var/log/httpd/ssl_error_log`	Yes
Control scripts	`/etc`	`/etc/logrotate.d/httpd`	No
PID files	`/var/run`	`/var/run/httpd.pid`	Yes
Core binaries	`/usr/sbin/`	`/usr/sbin/httpd`	Yes
Support binaries	`/usr/bin/`	`/usr/bin/ab`	No
Runtime libraries	`/usr/lib`	`/usr/lib/libapr-0.so.0`	Yes
Modules	`/usr/lib/httpd/modules`	`/usr/lib/httpd/modules/mod_access.so`	Yes

Misc. components	`/usr/sbin`
	`/usr/share`
	`/var/lib/dav`
	`/var/cache/mod_proxy`
		`/usr/sbin/suexec`
		`/usr/share/man/man1/htdigest.1.gz`
		`/var/lib/dav`
		`/var/cache/mod_proxy`
			No

Sample files	`/var/www`	`/var/www/icons/bomb.png`	No

There are additional components which were not installed from the Apache RPMs but required for the `httpd`: system libraries, devices, and pipes.

Additional libraries identified by `lsof` command must be included to chroot environment:

```
lsof -u apache | awk '{print $NF}' | sort -u
```

[23] See also publication [5]

```
/lib/ld-2.3.2.so
/lib/libc-2.3.2.so
/lib/libcrypt-2.3.2.so
/lib/libcrypto.so.0.9.7a
/lib/libdb-4.1.so
/lib/libdl-2.3.2.so
/lib/liblaus.so.1.0.0
/lib/libm-2.3.2.so
/lib/libnsl-2.3.2.so
/lib/libnss_files-2.3.2.so
/lib/libpam.so.0.75
/lib/libpcre.so.0.0.1
/lib/libpthread-0.10.so
/lib/libresolv-2.3.2.so
/lib/librt-2.3.2.so
/lib/libssl.so.0.9.7a
```

In addition, some files will be used occasionally, usually while starting new child process, e.g.:

```
/etc/passwd
/etc/group
/etc/mime.types
```

We will name these files "Additional components".

Creating the chroot environment for Apache HTTP

1. Create new `ext3` filesystem and mount it as `/chroot` with the following mount options: `rw,noatime,nosuid,noexec`

2. Create chroot structure for the functional groups:

Group "Logs"

- Create target directory

```
mkdir -p /chroot/apache/var/log/httpd
```

- Create "bind" links from real environment to jailed one[24]

```
mount --bind /var/log/httpd /chroot/apache/var/log/httpd
```

Groups "Runtime libraries", "Additional libraries", and "Modules"

- Create target directories

```
mkdir -p /chroot/apache/usr/lib/httpd
mkdir -p /chroot/apache/usr/kerberos/lib
mkdir /chroot/apache/lib
```

- Create "bind" links from real environment

```
mount --bind /usr/lib  /chroot/apache/usr/lib
mount --bind /usr/kerberos/lib  \
```

[24] This measure will provide additional protection because the original `/var` filesystem is mounted with more restrictive mount options (`rw,noexec,nosuid,nodev`) than if we would use log directories on `/chroot` filesystem. Another benefit of using "`--bind`" mount is: log analyzer software, e.g. `logwatch` and `webalizer` can use default settings to analyze http logs.

```
        /chroot/apache/usr/kerberos/lib
mount --bind /lib /chroot/apache/lib
```

Group "Core binaries"

Since we cannot bind the single file, we will make a "hard" link for this file to the directory already bound to the jail environment:

```
mkdir /usr/lib/httpd/sbin
ln /usr/sbin/httpd /usr/lib/httpd/sbin/httpd
cd /chroot/apache/usr && ln -s lib/httpd/sbin
```

Group "Devices"

- Create target directory

```
mkdir /chroot/apache/dev
```

- Create devices[25]

```
mknod  -m 644 /chroot/apache/dev/zero c 1 5
mknod  -m 666 /chroot/apache/dev/null c 1 3
mknod  -m 644 /chroot/apache/dev/urandom c 1 9
```

Group "Additional components"

- Create target directory

```
mkdir -p /chroot/apache/etc/httpd
```

- Create "bind" mount from real environment

```
mount --bind /etc/httpd /chroot/apache/etc/httpd
```

- For `syslog` support, we need a socket `/chroot/apache/dev/log`. To make it, specify in `/etc/sysconfig/syslog` file additional options:

```
SYSLOGD_OPTIONS="-m 0 -a /chroot/apache/dev/log"
```

and restart `syslog` daemon by the command:

```
service syslog restart
```

- Add required configuration files to the chrooted environment:

```
cp /etc/mime.types /chroot/apache/etc/
cp /etc/ld.so.cache /chroot/apache/etc
cp /etc/hosts /chroot/apache/etc
```

- Create files `/chroot/apache/etc/passwd` and `/chroot/apache/etc/group` containing the following lines:

`apache:x:48:48:Apache: /var/www/html:/sbin/nologin` in the `password`

`apache:x:48:` in the `group`

- Create place for PID files:

```
mkdir /chroot/apache/var/run
```

3. Create the content directory structure for HTML documents and CGIs:

```
mkdir /chroot/apache/www/error
mkdir /chroot/apache/www/cgi-bin
mkdir /chroot/apache/www/icons
mkdir /chroot/apache/www/html
```

4. Modify the server's startup file `/etc/init.d/httpd` to include the required mount commands (See Appendix E).

[25] The last device (`urandom`), although is not strictly required for the Apache http server, will be used for more reliable generation of PRNG in the SSL module.

Hardening Apache configuration options:
The following modifications of the configuration files should be done:

- Increase the number of processes available for new requests. {DoS->RCL->SA}
```
StartServers       10
```
- Decrease the request timeout. {DoS->RCL-SA}
```
Timeout 60
```
- The following LocationMatch directive reduces an accuracy of vulnerability scans which would run against the web server. {NST->SD}
```
<LocationMatch /(scripts|cgi-local|htbin|cgibin|cgis|cgi/|
win-cgi|cgi-win|finger\.pl|guestbook\.cgi|campas|files\.pl|
count\.cgi|*\~|*\.bak|*\.sav|*\.orig|*\.old)/>
   deny from all
</LocationMatch>
```
- Disable "inherited" options which will not be used (in the <Directory "/var/www/html">). {HeR}
```
Options -MultiViews -Indexes
```
- Limit allowed request methods to "POST" and "READ" (in the <Directory "/var/www/html">) {NST->CI}
```
<LimitExcept GET POST>
   deny from all
</LimitExcept>
```

The resulting Apache HTTP configuration files are included in the Appendix D.

4.2.4. Nagios Core

Hardening procedures:
1. Replace default interactive shell of the account nagios to /sbin/nologin {PE}
```
chsh -s /sbin/nologin nagios
```
Possible implications of this security measure: This change will break functionality of the Nagios startup script, see Appendix E for the required modification.
2. Lock the account:
```
passwd -l nagios
```
3. Restrict directories access {DT->ACL->CI}
```
chmod 700 /var/log/nagios/rw /var/log/nagios
chown -R root.root /usr/lib/nagios
chmod 755 /usr/lib/nagios
```
4. Add user nagios to /etc/ftpusers file
5. Make sure, if resource configuration file which may have user specific configuration options is owned by root and readable by the group nagios {DT->ACL->CI}:
```
chown -R root.nagios /etc/nagios/private
chmod 750 /etc/nagios/private
chmod 640 /etc/nagios/private/resource.cfg
```
6. Make Nagios log and configuration files to be accessible by the CGIs:
- Create target directories
```
mkdir /chroot/apache/var/log/nagios
mkdir /chroot/apache/etc/nagios
mkdir -p /chroot/apache/var/spool/nagios
```

- Create "bind" links from real environment[26]

```
mount --bind /var/log/nagios /chroot/apache/var/log/nagios
mount --bind /etc/nagios /chroot/apache/etc/nagios
mount --bind /var/spool/nagios \
      /chroot/apache/var/spool/nagios
```

7. Edit the main configuration file `nagios.cfg` and set the following options:
   - Set effective user and group that Nagios should run as

     ```
     nagios_user=nagios
     nagios_group=nagios
     ```

   - Disable event handlers. Risk code: `{PE->ACL|BoF|FSI->SO}`.

     ```
     enable_event_handlers=0
     ```

     "Event handlers" is powerful mechanism which allows Nagios not only to report the problems but also execute corrective actions, e.g. restart failed services. Typically, the only service which could be restarted using `nagios` UID is Nagios engine itself. To restart other services, `SUDO` or `SUID` methods may be required, which can introduce new security risks.
   - Protect CGIs and Nagios plugins from potentially dangerous characters in host names, service descriptions, or names of other object types `{HeR|FSI->SA}`.

     ```
     illegal_object_name_chars=`~!$%^&*|'"<>?,()=\+;
     ```

User access

Depending on the access level, users are able to do the following actions using Nagios CGI Interface:

- View system information about hosts they own or responsible for.
- View Nagios configuration for their hosts.
- Run commands to modify Nagios runtime configuration for their hosts.
- View system information about any hosts.
- View complete Nagios configuration for all services and hosts.
- Run commands to modify Nagios runtime configuration or stop any monitoring and alerting for any hosts and services.

Although it is possible to assign the access levels in the corporate LDAP database and let Apache HTTP server to control them by resource access directives, more simple and secure approach is to separate these functions:

- Do users' authentication via Apache server (line "`require valid-user`" in the `authz_ldap.conf` file).
- Do authorization via Nagios CGI programs. Assign the roles in the Nagios configuration file `cgi.cfg`.

4.2.5. Postfix Mail Server

To provide required functionality for the Nagios operations, the mail server should only send alerts from the account `nagios`. There is no need to store any messages for this account because nobody will read them. Therefore, any e-mail sent by Nagios which cannot be delivered, must be discarded. It will also provide protection from overfilling

[26] These "`--bind`" links will be mounted during Apache HTTP server startup process.

the /var/spool directory by discarded e-mail messages {DoS->RCL->SA}.

- Disable mail bouncing for the user nagios:
  - file main.cf: double_bounce_sender = nagios

Most of the Postfix services can run in the chroot environment.[27] {PE->ACL->CI}
Default chroot directory is /var/spool/postfix.
Making Postfix components to run in the chroot environment:

- Modify the master.cf file and specify chroot->"y" for all services except the "local"[28].
- To log the Postfix processes, modify /etc/sysconfig/syslog file and specify
  SYSLOGD_OPTIONS="-m 0 -a /chroot/apache/dev/log -a \
  /var/spool/postfix/var/run/log"[29]
- Restart the syslog daemon: service syslog restart
- Create the chroot directories, add required components:

```
mkdir /var/spool/postfix/etc
cp /etc/localtime /etc/resolv.conf /etc/services \
     /var/spool/postfix/etc
```

4.3. Target Nodes' NRPE Security Hardening

Why to worry about NRPE security on the target nodes? - Insecure or too restrictive configuration of NRPE on the target nodes can damage the whole Nagios monitoring process. For monitored-from-the-network services, like FTP or SSH one must make sure, if access from the Nagios server to the monitored services is not blocked by network devices or by local firewalls.

NRPE hardening steps

1. Lock the account nagios: passwd -l nagios
2. Add user nagios to /etc/ftpusers file
3. Protect the NRPE configuration directory:
   chmod 750 /etc/nagios && chown -R root.nagios /etc/nagios
4. Setup proper NRPE settings in the configuration file /etc/nagios/nrpe.cfg:
   - Set the effective user and group that the NRPE daemon will use to run as
     nrpe_user=nagios \ nrpe_group=nagios
   - Disable command argument processing: dont_blame_nrpe=0
   - Specify servers which are allowed to talk to the NRPE daemon:
     allowed_hosts=10.0.2.133[30]

There is the risk associated with the system resource allocation {DM->RCL->NE}.
NRPE daemon is running on non privileged port 5666, and therefore when this port is available, any user on the target host can start his own daemon on the port TCP/5666

[27] <http://www.postfix.org/BASIC_CONFIGURATION_README.html#chroot_setup>[13]
[28] All other services which cannot be chrooted, are already disabled during the first step of the Postfix Mail Server hardening.
[29] This option for the syslog daemon also includes the support for chrooted Apache HTTP server
[30] This is the Nagios server IP address

and supply bogus information for the Nagios server.
Risk mitigation:

1. Make NRPE started automatically on the system startup as a daemon:
   ```
   chkconfig nrpe on
   ```
2. Additionally, for the case NRPE daemon would crash due to attack to potential vulnerability, local iptables firewall should be configured to prevent outgoing NRPE traffic from any user, except nagios:

```
-A OUTPUT -m owner ! --uid-owner nagios -p tcp --sport 5666 -j LOG
-A OUTPUT -m owner ! --uid-owner nagios -p tcp --sport 5666 -j REJECT
```

4.4. Post-Deployment System Changes

Before final application security assessment tests can be executed, the following system changes must be performed:

1. Clean up directories which have been used to store installable packages.
2. Change the configuration file for the local iptables firewall:
   - Enable outgoing traffic for all ports which will be monitored by Nagios[31].
   - Enable incoming traffic from the GIAC Enterprises LAN to the ports: 80/TCP (HTTP server), 443/TCP (HTTP server / SSL), and 5666/TCP (NRPE)
3. Reboot the server to verify if all required services are started automatically on the system startup.
4. Test Nagios bundle functionality.

Nagios application bundle is ready for testing and validation.

5. Testing and Validation

We will start validation of the security hardening from the basic simple techniques like integrity checks. Then, if no security problems are found, we will continue with network scanners, and finish by Nagios bundle specific tests.

5.1. Baseline Security Check

The following security tests were done before the software deployment. By running them again, we will achieve the following objectives:

- Mitigate the risk of new security holes created by human error during software install or introduced by software install scripts[32];
- Verify if the integrity check procedures captures important changes.

5.1.1. Integrity Check

[31] Most sites have almost no restrictions for outgoing traffic in the LAN

[32] During the RPM install or update process, the internal RPM scripts are executed silently with root ID privileges. I have seen multiple examples that such scripts undermined the security of the system by changing system directory permissions, or changing firewall settings to allow the software to bypass firewall automatically, etc.

AIDE. The server files were verified against AIDE database from the supplied CD-R.

```
mount /mnt/cdrom && /mnt/cdrom/aide -c /mnt/cdrom/aide.conf

AIDE found differences between database and filesystem!!
Summary: Total number of files=50505,
added files=1276,removed files=0,changed files=44
```

The result of this output illustrated that all changes were expected. All "added" and "changed" files came from the Nagios deployment process.

5.1.2. Post-deployment System Security Rating

Most of the following security assessment tools already have been used during the pre-deployment assessment and hardening. I will not duplicate pre-deployment test results. Only the changes will be reported.

CIS Benchmark[33]. The score is 8.91 /10.00 (pre-deployment score was: 9.06).

- New "potential vulnerabilities" reported:
  - `Negative: 3.3 Mail daemon is still listening on TCP 25.`
    - This is the expected change. Postfix MTA is listening to the internal interface only (`127.0.0.1:25`).
  - `Negative: 3.15 Web server not deactivated.`
    - This is the expected change. Apache HTTP is running.
  - `Negative: 6.1 /chroot is not mounted nodev.`
    - It cannot be mounted with "`nodev`" because it uses the devices[34].
- Pre-deployment "potential vulnerabilities" are fixed:
  - Fixed networking kernel options for `/proc/sys/net/ipv4/*`

CIS Apache Benchmark[35]. Score: 5.81 out of 10.00.
The tool was run to verify both configuration files and running HTTP server:

```
./benchmark.pl -c /chroot/apache/etc/httpd/conf/httpd.conf \
-s  http://nagios.example.com
```

The report with comments is included in the Appendix H.

5.1.3. Network Services Testing

The network scanners have been run with the same options as during the pre-deployment assessment - i.e. for all possible ports and with all types of test enabled (even if marked as "dangerous"). For the HTTP-specific tests in the Nessus, Nikto[36] plugin was configured to test the actual server URLs.

[33] By the time the paper was completed new CIS Security Benchmark Checker became available (v.1.6.7). Test by the new version gave score 9.17 and less "Negative" warnings than the old CIS tool.
[34] Anyway, the current mount options `rw,noexec,nosuid,noatime` are not too bad.
[35] CIS Apache Benchmark. Version: 2.08. <http://www.cisecurity.org/bench_linux.html> [23]
[36] Nikto, v1.34. <http://www.cirt.net/code/nikto.shtml>

Check listening ports
Check by `lsof` has confirmed that only expected ports are listening to the network interfaces:

```
lsof -i -P -n

COMMAND   PID    USER   FD   TYPE DEVICE SIZE NODE NAME
sshd      815    root    3u  IPv4   1400       TCP *:22 (LISTEN)
ntpd      828     ntp    4u  IPv4   1429       UDP *:123
ntpd      828     ntp    5u  IPv4   1430       UDP 127.0.0.1:123
ntpd      828     ntp    6u  IPv4   1431       UDP 10.0.2.133:123
master    880    root   11u  IPv4   1580       TCP 127.0.0.1:25 (LISTEN)
stunnel   929 stunnel    4u  IPv4   1783       TCP 127.0.0.1:389 (LISTEN)
httpd   17612    root    3u  IPv4 135665       TCP *:80 (LISTEN)
httpd   17612    root    4u  IPv4 135667       TCP *:443 (LISTEN)
```

Scan by NMAP[37]
The result of the scan confirms that only expected ports are visible from the network:

```
PORT      STATE  SERVICE         VERSION
22/tcp    open   ssh             OpenSSH 3.6.1p2 (protocol 2.0)
80/tcp    open   http            Apache httpd
123/udp   open   ntp             NTP v4
443/tcp   open   ssl              OpenSSL
```

Scan by Nessus Security Scanner
The report is included in the Appendix I. Note: the test was run from the trusted network; some services (e.g. `NTP`) have tight access restrictions for non-trusted hosts.

- New "potential vulnerabilities" reported:
  - `The SSLv2 server offers 5 strong ciphers, but also 0 medium strength` **`and 2 weak "export class" ciphers.`**
    - Action taken: `SSLCipherSuite` parameter in the `ssl.conf` Apache HTTP configuration file have been modified to disallow use of the weak ciphers: `{DT->NST->CI}`

```
SSLCipherSuite RSA:!EXP:!NULL:RC4+RSA:+HIGH:+MEDIUM:-LOW:+SSLv2
```

- Pre-deployment "potential vulnerabilities" are fixed:
  - ICMP timestamp messages
  - TCP packets with *`ACK,FIN`* options

5.2. Software Bundle Check

These tests will verify the results of security hardening of the Nagios server bundle components.

5.2.1. NRPE

1. Verify if traffic between Nagios server and remote target host is **encrypted** by SSL/TLS when checks are done via `check_nrpe` plugin:

- Run `tcpdump` or `tethereal` on the target remote server, e.g:

```
tcpdump -s 0 -xX -vvv host nagios.example.com
```

- Try NRPE request from the Nagios server:

[37] NMAP, v.3.81. <http://www.insecure.org/nmap/> [19]

```
/usr/lib/nagios/plugins/check_nrpe -H \
     anyhost.example.com  -c check_users
USERS OK - 1 users currently logged in |users=1;5;10;0
```

- The response confirms that communication was established between the Nagios server and target host (network access to this service is not blocked).
- The result of packet sniffing confirms that all traffic is obscured (presumably encrypted by SSL/TLS).

2. Verify if SSL/TLS encryption connection is **required** when Nagios server checks remote hosts via `check_nrpe` plugin:

- Try to connect to server running older version of NRPE (1.8) which is not requires SSL encryption:

```
/usr/lib/nagios/plugins/check_nrpe -H \
     oldhost.example.com -c check_users
CHECK_NRPE: Error - Could not complete SSL handshake.
```

- The response confirms that the `check_nrpe` plugin on the Nagios server refused to communicate with the target host without an encryption.

3. Verify if NRPE responds only to requests from the **authorized** Nagios server:

- Try to make NRPE request from the unauthorized server:

```
/usr/lib/nagios/plugins/check_nrpe -H \
     anyhost.example.com  -c check_users
CHECK_NRPE: Error - Could not complete SSL handshake.
```

- The response confirms NRPE does not give information for unauthorized hosts.

Additionally, this unauthorized attempt was **logged** in the `/var/log/messages` file on the target host:

```
Feb 23 17:43:11 anyhost nrpe[xx]: Host 10.0.7.7 is not allowed to talk to
us!
```

5.2.2. Apache HTTP

1. Check if `httpd` is running **chrooted**:

```
lsof -d rtd -a -c httpd

COMMAND  PID   USER   FD   TYPE DEVICE SIZE  NODE NAME
httpd   4900   root  rtd    DIR   8,10 4096 32769 /chroot/apache
     <... Similar lines was skipped ...>
httpd   4909 apache  rtd    DIR   8,10 4096 32769 /chroot/apache
```

... or run the `ls` command to see what is available for this process (use the PID reported by the previous command):

```
ls -l /proc/4900/root/

total 20
```

```
drwxr-xr-x    2 root     root         4096 Feb  7 14:17 dev
drwxr-xr-x    3 root     root         4096 Feb 15 15:34 etc
drwxr-xr-x    7 root     root         4096 Dec 23 10:45 lib
drwxr-xr-x    5 root     root         4096 Feb  1 22:02 usr
drwxr-xr-x    5 root     root         4096 Feb 15 15:58 var
```

2. **Performance** test helps to verify how this component of the Nagios server could bear normal load and possible {DoS} attacks:

- Start vmstat utility on the Nagios server
- Start HTTP heavy performance test using the analyzing tool ApacheBench (part of the Apache HTTP server package) on the remote server:

```
ab -n 100000 -c 20 -k http://nagios.example.com/error/403.html
```

```
Server Port:            80
```

The version of ApacheBench which we have used does not support SSL. Therefore, the test was done on the port 80.

```
Document Path:          /error/403.html
Document Length:        1644 bytes
Concurrency Level:      20
Time taken for tests:   8.735315 seconds
Complete requests:      100000
Failed requests:        0
Write errors:           0
Total transferred:      191455828 bytes
Requests per second:    11659.97 [#/sec] (mean)
Time per request:       1.715 [ms] (mean)
Time per request:       0.086 [ms] (mean, across all
concurrent requests)
Transfer rate:          21800.40 [Kbytes/sec] received
```

The results of vmstat indicated that the server had 100% CPU utilization during this test. Therefore, the experimental {DoS} attack was successful. However, it was unlikely to see that problem if the attack would be started from outside of the local LAN because the HTTP transfer rate during the test was 175Mb/s, i.e. 26 times more than capacity of T2 line by which GIAC Enterprises is connected to the Internet.

Less stressful test (with the same parameters but concurrency level equal to 4) shows acceptable level of CPU utilization (80%) and HTTP transfer rate of 16MB/s.

This result is acceptable, and no performance tuning methods will be described here. However, some {DoS} mitigation techniques will be presented later in the chapter "7. Summary and Research".

5.2.3. Stunnel

1. Check if stunnel is running in **chrooted** environment:

- lsof -d rtd -a -c stunnel

```
COMMAND   PID    USER   FD   TYPE DEVICE SIZE  NODE NAME
stunnel 30545 stunnel  rtd    DIR    8,5 4096 81927 /var/run/stunnel
stunnel 31038 stunnel  rtd    DIR    8,5 4096 81927 /var/run/stunnel
```

... or run the `ls` command to see what is available for this process (use the PID reported by the previous command):

```
ls -l /proc/30545/root/
```

```
total 4
-rw-r-----    1 root     root            0 Feb 22 18:07 stunnel.log
-rw-r--r--    1 stunnel  stunnel         6 Feb 22 18:07 stunnel.pid
```

2. Verify if traffic between stunnel and corporate LDAP server is **encrypted** by SSL:

- Run `tcpdump` on the Nagios server:
  ```
  tcpdump -s 0 -xX -vvv port ldap or port ldaps -i eth0 -w stunnel.in&
  tcpdump -s 0 -xX -vvv port ldap or port ldaps -i lo -w stunnel.out&
  ```
- Try to authenticate on Nagios server (the URL is <https://nagios.example.com/nagios>)
- Stop (kill) `tcpdump`
- Compare files `stunnel.in` (dump of communication between `stunnel` and the corporate LDAP server) and `stunnel.out` (communication dump between `stunnel` and local Apache HTTP server (`mod_ldapz`))
- The result of packet sniffing confirms that all external LDAP traffic is obscured (presumably **encrypted by SSL**) while internal one is unencrypted.

5.2.4. Postfix Mail Server

1. Check if **mail bouncing** is disabled for the user `nagios`:

- Send a mail to the non existent recipient from the user ID `nagios`
- Check the `/var/log/maillog` for the error messages:

Feb 27 16:09:09 nagios postfix/bounce[1049]: warning: 7B11914022: undeliverable postmaster notification discarded

- Check `/var/spool/mail/nagios` mailbox.
- Result: message **did not bounce** back to `nagios` account.

2. Check if application is running in **chrooted** environment[38]:

- `lsof -d rtd -a -u postfix`

```
COMMAND    PID    USER   FD   TYPE DEVICE SIZE   NODE NAME
pickup     992 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
nqmgr      993 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
trivial-r  994 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
smtp       995 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
cleanup   1043 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
```

[38] The Postfix must be doing something useful during the test. Idle state does not display some Postfix processes.

```
smtp      1044 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
local     1048 postfix  rtd    DIR    8,7 4096      2 /
bounce    1049 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
showq     1051 postfix  rtd    DIR    8,5 4096 213001 /var/spool/postfix
```

Result: all configured Postfix components are running **chrooted**.
Note: the component "`local`" cannot be chrooted. This is the expected result.

5.2.5. Nagios Plugins

This component of the Nagios application bundle is most likely to have some `{BoF}` or `{FCI}` vulnerabilities. The reason for this assumption is: Nagios plugins are simple, and could be written even by junior software developer. Moreover, if the component has limited applicability, it is a good chance if this component never has been under proper testing. The BoFCheck[39] tool was used to test all installed Nagios plugins.

Test 1. Test for simple buffer overflow:

```
for i in `find /usr/lib/nagios/plugins -type f`; do
/root/bofcheck -b 99999 -f $i ; done
```

The result of the test is: **No vulnerabilities found**.

Test 2. Test for generic format strings bugs:

```
for i in `find /usr/lib/nagios/plugins -type f`; do
/root/bofcheck -s -f $i ; done
```

As a result of the test we have **new discovered vulnerability**:

```
[/usr/lib/nagios/plugins/check_smtp] with option -S
CAUGHT[SIGILL]Signal[14]
* [ESP] at time of crash [0xfeff8770]
* [0x41414141] string possibly overwrote address[0xfeff88f1]

[/usr/lib/nagios/plugins/check_smtp] with option -S
CAUGHT[SIGALRM]Signal[14]
* [ESP] at time of crash [0xfeff8770]
* [0x41414141] string possibly overwrote address[0xfeff88f1]
```

Risk assessment: the risk factor is **low.** Risk code is `{PE->FSI->SA}`. Normally, the plugin runs under unprivileged system ID `nagios`, has no `SUID/SGUID` bits set, and uses command arguments supplied either by Nagios server or by NRPE daemon.

Mitigation of this vulnerability: The C source code of the `check_smtp` plugin can be reviewed and fixed by GIAC Enterprise programmers. Meanwhile, the stack protection must be activated for this binary file by `execstack` utility to prevent this vulnerability to be exploited:

```
execstack  -s /usr/lib/nagios/plugins/check_smtp
```

To verify if stack protection is activated for this binary, run:

[39] BoFCheck. <http://www.securiteam.com/tools/5DP0C1PB6G.html> [22]

```
execstack  -q /usr/lib/nagios/plugins/check_smtp

X /usr/lib/nagios/plugins/check_smtp
```

"X" means the stack protection is supported and activated.

Test 3. Test for overflow over specified environment variables[40]:

```
for i in `find /usr/lib/nagios/plugins -type f`; do
/root/bofcheck -e `set | awk 'FS="=" {print $1}' |
xargs | sed -e 's/ /,/g'` -f $i ;  done
```

The complete list of environment variables which have been used for the test is:

```
BASH,BASH_ENV,BASH_VERSINFO,BASH_VERSION,COLORS,COLUMNS,DIRSTACK,EUID
,GROUPS,G_BROKEN_FILENAMES,HISTFILE,HISTFILESIZE,HISTSIZE,
HOME,HOSTNAME,HOSTTYPE,IFS,INPUTRC,LANG,LANGVAR,LESSOPEN,LINES,
LOGNAME,LS_COLORS,MACHTYPE,MAIL,MAILCHECK,OPTERR,OPTIND,OSTYPE,
PATH,PIPESTATUS,PPID,PS1,PS2,PS4,PWD,SHELL,SHELLOPTS,SHLVL,
SSH_CLIENT,SSH_CONNECTION,SSH_TTY,SUPPORTED,TERM,TMOUT,UID,USER,
USERNAME
```

The result of the test is: **No vulnerabilities found.**

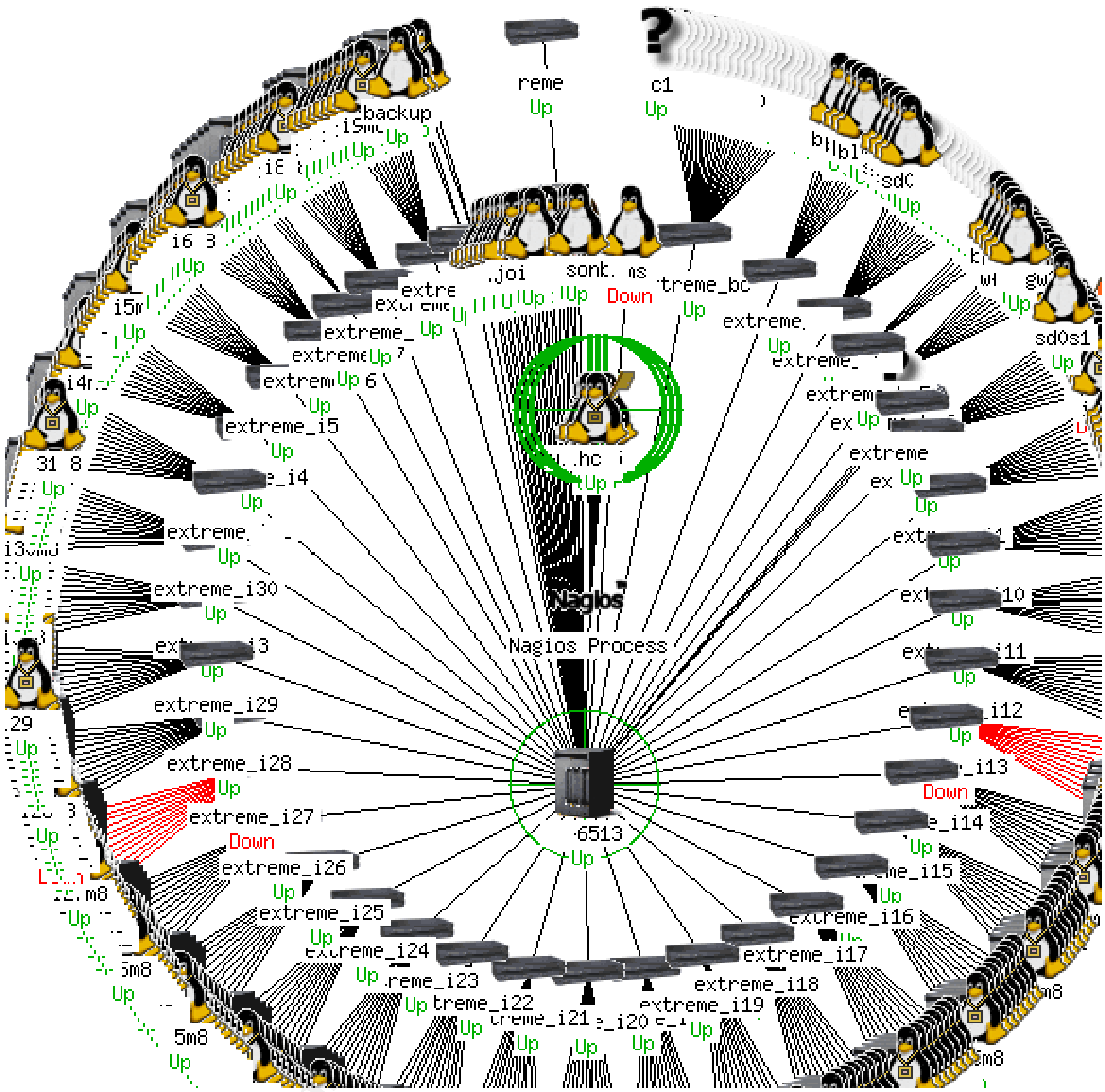
5.2.6. Nagios Core

Since all interfaces of the Nagios core process (Nagios plugins and CGI programs) already have been tested, we can start the last test of the Nagios bundle - the capacity test.

The capacity test in the lab network of GIAC Enterprises proved that installed Nagios server is capable to monitor 500 servers and 3500 services with reasonable system resources utilization.

Figure 4: Nagios Core performance test

[40] List of the environment variables was extracted from output of the `set` command.



The `sar` utility reported CPU utilization during this test:

```
12:00:00 AM       CPU      %user      %nice    %system        %idle
12:10:00 AM       all      14.72       0.00       7.94        77.34
...
Average:          all      14.41       0.00       7.92        77.66
```

5.3. When the Tests are Completed...

1. Remove security assessment tools (as mentioned earlier, GIAC Enterprises security policy does not allow to have such tools permanently installed on the critical servers).
2. Update `AIDE` database, save it on secured external media.

6. Maintenance

6.1. Change Control

Change control is very important part of software maintenance. There is no version control software installed on the server, but all changes are documented and reported in the repository, according to GIAC Enterprises access control procedures. To ensure that all system changes are reported properly, periodic report generated by integrity checking tool (`AIDE`) is used.

6.2. Security Announcements

Beyond the usual Linux and Red Hat security announcements, the following application mailing list must be monitored:

- Nagios engine and plugins components
  - mailing list <nagios-users@lists.sourceforge.net>
- Apache HTTP server and modules
  - mailing list <security@apache.org>
  - Apache 2.0 Security Vulnerabilities <http://www.apacheweek.com/features/security-20>

6.3. Updates and Patches

Production servers in GIAC Enterprises are never patched directly from the Internet using `up2date, yam, ap-get`, or similar utilities. All patches and updates are copied from the special internal servers after being carefully examined in the test lab. Special servers are used to create new binaries, scripts, and third-party software; and to keep all versions of RPM update files.

Nagios Engine and Plugins Update strategy
In addition to the security problems which can be discovered in the Nagios Engine of plugins, the following packages used for compiling Nagios components must trigger the update process:

```
gd-devel, zlib-devel, libpng-devel, libjpeg-devel
```

Update procedure:

1. Build new RPMs, using instruction in the Appendix C.
2. When you have the RPMs, update the components using command:
   ```
   rpm -Uvh nagios-<component>-<new-version>.i386.rpm
   ```
3. Report the completion of the update in the Enterprise's change control system.

Postfix Mail Server, stunnel, and Apache HTTP Update strategy
These components are typically updated when Red Hat sends security alert for some vulnerability and the corresponding RPM files.

Update procedure:

1. Update the affected components, using simple "`rpm -U`" command.
2. Report the completion of the update in the Enterprise's change control system.

Chroot environment for these programs does not require any manual changes after the RPM update is completed because libraries and binaries in the `/chroot` are always linked to the current RPM components in the original /`usr` and `/lib` directories.

6.4. Ongoing System Audit

There are Nagios specific system audits which must be done in addition to system audits run for any common server in the GIAC Enterprises:

- Check Integrity of Nagios bundle components by AIDE.
- Test by Nessus Security Scanner, always use additional plugins to test for HTTP, SSL, and CGI vulnerabilities.
- Check http/ssl error logs for unsuccessful authentication attempts to the Nagios GUI (this attempts will also be recorded on the corporate LDAP server).
- Check target node system log files for unauthorized NRPE access attempts.

6.5. User Access and Passwords

In the current authentication schema, Nagios server does not use any locally stored user authentication information. Corporate LDAP server is used for authentication. User authorization is controlled by `cgi.cfg` configuration file and directly linked to the roles and responsibilities the user is assigned to. By default, users can only view information about a hosts or services that they are registered contacts for.

System accounts created during the Nagios software deployment are locked and do not have any passwords.

6.6. Backup Procedures

Once the software is deployed and tested, the entire server must be backed up, using the System Imager server. The total disk usage of the server is below 1GB and full backup can be done after any significant server change or update.
The Nagios specific files which must be backed up frequently (perhaps, daily) are:

- Nagios configuration files (`/etc/nagios/*`): <1MB
- Nagios logs (`/var/log/nagios/*`): 20-200MB

The backups of these files can be done on-line without stopping any Nagios processes.

6.7. Monitoring Procedures

Even the server which monitors others must be monitored. While some system components like filesystem utilization, zombie processes, etc. can be monitored and reported by Nagios core locally, the server will not alert anybody about its problems with mail server, network interfaces, or high CPU utilization. Best mitigation of this problem is distributed or "HA" Nagios infrastructure, where multiple Nagios servers can

send alerts about problems with their counterparts.

Without the distributed infrastructure, Nagios can be monitored from any other server (let's call it "`serverX`") using the same tool which is used by Nagios to monitor target hosts - `NRPE`.

How to implement this "poor guy" solution:

1. Install `NRPE` daemon on the Nagios server.
2. Configure `nrpe.cfg` to allow `NRPE` requests from the `serverX`
3. Change local firewall rules (`iptables`) to allow incoming `NRPE` traffic from the `serverX`.
4. Configure `nrpe.cfg` on the Nagios server to allow `NRPE` to check the following local services: Nagios core process, Postfix MTA, Apache server, network interfaces, etc.
5. Setup `cron` job on the `serverX` to run `check_nrpe` command, check the specified services on the Nagios server, and send alerts if something is wrong. Example of the command to check Nagios engine remotely is:

```
/usr/lib/nagios/plugins/check_nrpe -H nagios.example.com \
check_nagios
```

7. Summary and Research

Securing a software bundle is task which is quite different from securing individual applications.

On the positive side, one doesn't need to harden all functional components of each individual application because the role of some application is strictly defined by the required functionality. E.g. if there is no requirements for having `PHP` on the web server, where is no need to harden that component - just disabling it is enough for the hardening.

On the negative side, one have to solve the problem how to separate bundle components (by using special system accounts, Access Control List, and "chroot" environment) for better security, but provide enough access rights for the bundle components to let them communicate to each other.

In the course of security hardening of the Nagios software bundle the following risk mitigation actions have been performed:

- ✓ Made the following applications running in the separate chroot environments:
  - Apache HTTP and Nagios GUI.
  - Stunnel
  - Postfix Mail Server
- ✓ Improved security of SSL encryption of the HTTP server
- ✓ Setup SSL/TLS encryption for remote NRPE checks and LDAP authentication
- ✓ Increased component separation by running bundle components under separate non-privileged system accounts with minimal privileges: `apache, stunnel, nagios, postfix`
- ✓ Tightened applications' file and directory access permissions.
- ✓ Implemented countermeasures against various Denial of Service attacks.
- ✓ Increased application availability by improving performance of core components.
- ✓ Provided application hardening procedures which make possible to use simple and robust RPM update methods for the bundle software management.
- ✓ Provided hardening procedures for the target hosts which are monitored by Nagios.

However, the following future improvements should be considered:

7.1. Improving Performance and Availability

To improve performance of the Nagios server the following techniques could be implemented:

- Create distributed Nagios environment.
- Use passive checks and NSCA[41].
- Replace or rewrite shell script Nagios plugins to faster alternatives: compiled binaries or Perl scripts (provided in this paper configuration for the Nagios software utilizes embedded Perl to make Perl scripts almost as fast as compiled

[41] NSCA. < http://nagios.sourceforge.net/docs/1_0/addons.html >

programs).

One performance related security issue was identified in the chapter "Testing and Validation" of this paper. HTTP server could utilize too many CPU resources of the server (risk code is `{DoS->RCL->SA})`, which could slow down or stop the most important Nagios's functions: monitoring and alerting.
To mitigate this risk one of the following methods can be implemented:

- Change CPU priority of the HTTP server by modifying its startup script.
- Use local `iptables` firewall to limit the rate of requests to the HTTP server.
- Use request rate profiling capability of Apache `mod_throttle` and `mod_security` modules when they became more mature (the current experimental status of these modules prevents its use in the mission critical servers at the GIAC Enterprises).

7.2. Further Security Enhancements

7.2.1. Some of identified potential vulnerabilities (e.g. `{FSI}, {ACL}, {HeR})` could be mitigated by additional layers of security:

Access control limitations in the current security DAC model still leave a lot of opportunities to discover and exploit new vulnerability. For details, please check paper [4]. Thus, crucial part of the next security hardening should be implementing of MAC security model controls (e.g. as a SELinux component of RHEL4).

> *In a MAC-based environment, application capabilities and privileges are set by predefined policies and enforced by the kernel. This prevents errant applications from compromising system security.* [42]

7.2.2. To extend security specific usage of the Nagios server, the following components could be installed:

- Portsentry Integration[43]
- NMAP plugin
- Prelude[44] Integration

7.2.3. The SSL/TLS encryption schema which is used in this software bundle (components: `stunnel` and `NRPE`) susceptible to the "man-in-the-middle" attack because there is no solid authentication of server and client during the initial key exchange.

To mitigate this risk for the `stunnel`, the corporate LDAP server's SSL certificate must be copied and installed on the Nagios server. While the new SSL certificate must be generated on the Nagios server to be sent and installed on the LDAP server.

[42] < http://www.redhat.com/software/rhel/features/ > [10]
[43] <http://nagios.sourceforge.net/docs/1_0/int-portsentry.html> and <http://nagios.sourceforge.net/docs/2_0/volatileservices.html>
[44] <http://www.prelude-ids.org/>

There is no easy way to mitigate this risk in the NRPE because the way it was designed. Current design of the NRPE uses Anon-DH encryption schema. "*This allows for an encrypted SSL/TLS connection without using pre-generated keys or certificates*[45]*"*. Once started NRPE generates new SSL keys dynamically. This approach helps to avoid the problems usually associated with certificate distribution process. However, if this risk is unacceptable, an additional layer of encryption can be created using mutual authentication provided by stunnel.

7.2.4. Memory stack protection methods, although already activated on the system level, require additional activation for each individual component (for example see how it was activated for the plugin check_smtp in the chapter "Testing" of this paper).

The reason why not all system binaries already have this protection activated is explained in the documentation for the execstack:

> *To avoid breaking binaries and shared libraries which need executable stack, ELF binaries and shared libraries now can be marked as requiring executable stack or not requiring it. This marking is done through the p_flags field in the PT_GNU_STACK program header entry. If the marking is missing, kernel or dynamic linker need to assume it might need executable stack*[46]*.*

Quick examination of the all system programs available on the server was done by command:

```
find /boot /lib /usr /bin /sbin /opt -type f \
| xargs -n 1  execstack -q
```

As a result, only 45 binaries (from the total number of 1631 "ELF" files) was identified as having a stack protection flag set.

This justifies the decision not to set stack protection flag on all binaries of the Nagios bundle right now -- if things may break, implementing complex protection techniques without extensive testing process will affect application availability (one of the main asset of the Nagios bundle we are protecting).

However, as future security enhancement, implementing of this measure is highly recommended when GIAC Enterprises QA department or open source community properly tested the possible implications of stack protection in the Nagios software bundle.

[45] README.SSL from the NRPE distribution [21]
[46] Linux Programmer's Manual (8) for the execstack. [12]

References

1. The Center for Internet Security. Linux Benchmark v1.1.0. February 17, 2005. <http://www.cisecurity.org/bench_linux.html>[47].

2. R. Russell, D. Quinlan, C. Yeoh. Filesystem Hierarchy Standard. January 2004. <http://www.pathname.com/fhs/>.

3. Oskar Andreasson. Iptables Tutorial 1.1.19. 2003. <http://www.faqs.org/docs/iptables/>

4. P. Loscocco, S. Smalley, P. Muckelbauer, etc. The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments. NSA, 2003. <http://www.nsa.gov/selinux/papers/inevitability.pdf>.

5. Jarno Gassenbauer . " \"chroot\" + \"mount --bind\" + \"ln\" _or_ \"chroot\" + \"cp -a\"?. ". <debian-security@lists.debian.org>. May, 2003. <http://lists.debian.org/debian-security/2003/05/msg00355.html>

6. Simon Ostengaard. Securing PowerDNS on Debian GNU/Linux 3.0. August 2004. SANS Institute. GIAC practical repository. <http://www.giac.org/practical/GCUX/Simon_Ostengaard_GCUX.pdf>

7. Ethan Galstad, Nagios Version 2.0 Documentation, 2004 <http://nagios.sourceforge.net/docs/2_0/>

8. Randy Warner. Securing Apache Web Server on RHEL3.0. Sep. 2004. SANS Institute. GIAC practical repository.

9. Apache Software Foundation. Apache HTTP Server Version 2.0 Documentation. 2005. <http://httpd.apache.org/docs-2.0/>

10. Red Hat Inc. Red Hat Enterprise Linux v.4 Features and Benefits. 2005. <http://www.redhat.com/software/rhel/features/>

11. Red Hat, Inc. Red Hat Enterprise Linux 3 Update 3 Release Notes (x86 Edition). 2004. < http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/release-notes/as-x86/RELEASE-NOTES-U3-x86-en.html >

12. Jakub Jelinek. execstack (8) Linux Programmer's Manual. 2003. Red Hat Inc.

13. Wietse Venema. Postfix Basic Configuration. <http://www.postfix.org/BASIC_CONFIGURATION_README.html>

14. The Center for Internet Security. Apache Benchmark for Unix. For Apache Versions

[47] Free registration is required to download documentation and tools from this server.

1.3 and 2.0. 2004. <http://www.cisecurity.org/bench_linux.html>[47].

Software Distributives

15. NAGIOS <http://sourceforge.net/projects/nagios/>
Plugins: <http://sourceforge.net/project/showfiles.php?group_id=29880>

16. SystemImager. Bald Guy Software project. http://www.systemimager.org/

17. CIS Security Benchmark Checker. The Center for Internet Security.
<http://www.cisecurity.org/bench_linux.html>[48].

18. AIDE (Advanced Intrusion Detection Environment)
<http://www.cs.tut.fi/~rammer/aide.html> and
<http://sourceforge.net/projects/aide>

19. NMAP ("Network Mapper").
<http://www.insecure.org/nmap/nmap_download.html>

20. Nessus.
<http://www.nessus.org/download/>

21. Nagios Remote Plugin Executor (NRPE)
<http://prdownloads.sourceforge.net/nagios/nrpe-2.0.tar.gz?download>, also
<ftp://fr2.rpmfind.net/linux/dag/fedora/3/en/i386/SRPMS.dag/>

22. BofCheck.c - Coded by sw @ .:[oc192.us]:. Security.
<http://www.securiteam.com/tools/5DP0C1PB6G.html>

23. Apache Benchmark for Unix For Apache Versions 1.3 and 2.0. Center for Internet Security. <http://www.cisecurity.org/bench_linux.html>[48].

[48] Free registration is required to download documentation and tools from this server.

Appendixes

Appendix A Installed Software Packages

Initial system RPMs before Nagios software have been installed.

```
acl     aspell  aspell-config   at      atk     attr        authconfig
basesystem      bash    bc      beecrypt        bind-libs  bind-utils
binutils        bison   byacc   bzip2   bzip2-libs        cdecl
chkconfig       compat-glibc    compat-libstdc++    coreutils
cpio    cracklib        cracklib-dicts  crontabs    curl    cyrus-
sasl
cyrus-sasl-gssapi       cyrus-sasl-md5         cyrus-sasl-plain
db4     db4-utils       dev     dev86   devlabel        dialog
diffutils       dos2unix        dosfstools      e2fsprogs       ed
eject   elfutils        elfutils-libelf         elinks  ethtool
expat   expect  file    filesystem      findutils       flex    ftp
gawk    gdb     gdbm    glib    glib2   glibc   glibc-common    gmp
gnupg   gpm     grep    groff   grub    gzip    hdparm  hwdata  info
initscripts     iproute         ipsec-tools     iptables
iputils  kbd     kernel-smp     kernel-utils    krb5-libs      krb5-
workstation
kudzu   laus-libs       less    libacl  libaio  libart_lgpl  libattr
libcap  libgcc  libjpeg         libmng  libobjc         libpng
libstdc++       libtermcap      libtiff         libtool-libs
libuser         libxml2         libxslt        linc    lockdev
logrotate       logwatch        losetup         lslk    lsof    lvm
m4      mailcap         mailx   make    MAKEDEV      man     man-
pages mgetty  mingetty       mkbootdisk      mkinitrd  mktemp
modutils  mount   mtools  nano    nc      ncompress       ncurses
net-snmp
net-snmp-libs   net-snmp-utils  net-tools      newt    nfs-utils
nscd    nss_ldap        ntp     openldap        openldap-clients
openssh         openssh-clients         openssh-server  openssl
openssl096b     pam     passwd  patch   patchutils      pax
pciutils        pcre    pdksh   perl    perl-AppConfig  perl-CGI
perl-CPAN       perl-DateManip  perl-DB_File    perl-DBI
perl-Digest-HMAC        perl-Digest-SHA1        perl-Filter
perl-HTML-Parser        perl-HTML-Tagset        perl-libwww-perl
perl-libxml-enno        perl-libxml-perl        perl-Net-DNS    perl-
Parse-Yapp      perl-Time-HiRes    perl-URI    perl-XML-Dumper
perl-XML-Encoding       perl-XML-Grove         perl-XML-Parser
perl-XML-Twig   pinfo   pkgconfig     popt    portmap         prelink
procps  psacct  psmisc         pspell  pstack  pyOpenSSL       python
python-optik    quota   readline        redhat-logos    redhat-menus
redhat-release  rootfiles       rpm     rpmdb-redhat    rpm-libs
rsync   schedutils      sed     setarch         setup   sgml-common
shadow-utils    sharutils       slang   slocate         star    strace
sudo    symlinks        sysklogd        syslinux        sysreport
sysstat         SysVinit        tar     tcl     tcp_wrappers    tcsh
telnet  termcap         texinfo         time    tmpwatch
traceroute      tzdata  unix2dos        unzip   usermode    utempter
```

```
util-linux      vim-common      vim-enhanced    vim-minimal
vixie-cron    vlock   wget    which   words   zip     zlib    zsh
```

Appendix B Pre-deployment Security Benchmarks

CIS Benchmarks

Rating = 9.06 / 10.00

Possible security problems only. Output was formatted. Long comments been removed.

```
#CIS Benchmark results before additional hardening (Negative only)
Negative: 3.6 gpm not deactivated.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_source_route should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_source_route should be set to 0.
Negative: 4.2 /proc/sys/net/ipv4/conf/eth0/send_redirects should be set to 0.
Negative: 4.2 /proc/sys/net/ipv4/conf/lo/send_redirects should be set to 0.
Negative: 7.11 /etc/inittab needs a /sbin/sulogin line for single user mode.
Negative: 8.10 Current umask setting in file /etc/profile is 000
Negative: 8.10 Current umask setting in file /etc/profile is 000
Negative: 8.10 Current umask setting in file /etc/csh.login is 000
Negative: 8.10 Current umask setting in file /etc/csh.login is 000
Negative: 8.10 Current umask setting in file /etc/bashrc is 022
Negative: 8.10 Current umask setting in file /etc/bashrc is 022
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000
Negative: 8.10 Current umask setting in file /root/.bashrc is 000
Negative: 8.10 Current umask setting in file /root/.bashrc is 000
Negative: 8.10 Current umask setting in file /root/.cshrc is 000
Negative: 8.10 Current umask setting in file /root/.cshrc is 000
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000
Negative: 6.7 Non-standard SUID program /bin/traceroute
Negative: 6.7 Non-standard SUID program /bin/ping6
Negative: 6.7 Non-standard SUID program /bin/traceroute6
```

Appendix C Software Compilation and Install Procedures

This Appendix includes instructions how to build and install Nagios bundle components.
The following procedures do not include security hardening steps.
All compilation must be done on separate "build" server.

Apache HTTP Server, SSL and LDAP Plugins

Source files: Binary RPMs from Red Hat.
Install procedure:

1. Check RPM signatures

```
rpm -K httpd-2.0.46-44.ent.i386.rpm mod_authz_ldap-0.22-5.i386.rpm  mod_ssl-2.0.46-44.ent.i386.rpm

httpd-2.0.46-44.ent.i386.rpm: (sha1) dsa sha1 md5 gpg OK
mod_authz_ldap-0.22-5.i386.rpm: (sha1) dsa sha1 md5 gpg OK
mod_ssl-2.0.46-44.ent.i386.rpm: (sha1) dsa sha1 md5 gpg OK
```

2. Install RPMs

```
rpm -Uvh httpd-2.0.46-44.ent.i386.rpm mod_ssl-2.0.46-44.ent.i386.rpm mod_authz_ldap-0.22-5.i386.rpm
```

3. Configure the server:
   - HTTPd: the configuration file included in the Appendix D.
   - SSL plugin: use the default configuration file `ssl.conf`
   - LDAP plugin: configuration file is included in Appendix D.

Stunnel

Source files: Binary RPMs from Red Hat.
Install procedure:

1. Check RPM signatures

```
rpm -K stunnel-4.04-4.i386.rpm

stunnel-4.04-4.i386.rpm: (sha1) dsa sha1 md5 gpg OK
```

2. Install it

```
rpm -Uvh stunnel-4.04-4.i386.rpm
```

3. Create startup file `/etc/init.d/stunnel` to make this service started automatically on system reboot.

4. Create log directory

```
mkdir /var/run/stunnel
```

5. Create configuration file `/etc/stunnel/stunnel.conf`

```
client = yes
debug = 2
output = /var/run/stunnel/stunnel.log
[ldaps]
    accept  = 127.0.0.1:389
    connect = ldap.example.com:636
```

Postfix Mail Server

Source files: Binary RPMs from Red Hat.
Install procedure:

1. Check RPM signatures

```
rpm -K postfix-2.0.16-14.RHEL3.i386.rpm

postfix-2.0.16-14.RHEL3.i386.rpm:(sha1) dsa sha1 md5 gpg OK
```

2. Install it:

```
rpm -Uvh postfix-2.0.16-14.RHEL3.i386.rpm
```

Nagios Engine

Source files: Source file `nagios-2.0b1.tar.gz` from the SourceForge [35], binary RPMs from Red Hat.
Build procedure:

1. Download file `nagios-2.0b1.tar.gz`
2. Check MD5 checksum of the archive file.
   Expected value is: `994a89bd6d344f72a0520b72fb615857`
3. Unpack the archive, Copy `nagios.spec` into `/usr/src/redhat/SPECS`
4. Copy `nagios-2.0b1.tar.gz` in `/usr/src/redhat/SOURCE`
5. Fix the `nagios.spec` file.
   The supplied `nagios.spec` file is broken and must be fixed before the RPMs could be built on RHEL3 system. Default access permissions for some files also have been tightened there.
   List of required modifications of `nagios.spec` file from Nagios distributive v. 2.01b:

```
diff -w nagios.spec.orig nagios.spec

208a209
>       --with-perlcache \
271c272,273
< cp contrib/htaccess.sample
${RPM_BUILD_ROOT}/etc/httpd/conf.d/nagios.conf
---
> [ -f contrib/htaccess.sample ] && cp contrib/htaccess.sample
${RPM_BUILD_ROOT}/etc/httpd/conf.d/nagios.conf
> [ -f sample-config/httpd.conf ] && cp sample-config/httpd.conf
${RPM_BUILD_ROOT}/etc/httpd/conf.d/nagios.conf
276,277c278,280
< mv ${RPM_BUILD_ROOT}%{_prefix}/lib/nagios/cgi/convertcfg
${RPM_BUILD_ROOT}%{_prefix}/lib/nagios/
```

```
< mv ${RPM_BUILD_ROOT}%{_prefix}/lib/nagios/cgi/mini_epn
${RPM_BUILD_ROOT}%{_prefix}/sbin/
---
> # install Tools
> mv /usr/sbin/convertcfg ${RPM_BUILD_ROOT}%{_prefix}/sbin/
> mv /usr/sbin/mini_epn ${RPM_BUILD_ROOT}%{_prefix}/sbin/
304c307
< %{_prefix}/lib/nagios/convertcfg
---
> %{_prefix}/sbin/convertcfg
315c319
< %defattr(2775,%{nsusr},%{nsgrp})
---
> %defattr(2750,%{nsusr},%{nsgrp})
316a321
> %defattr(644,root,root)
```

6. Build RPMs:

```
rpmbuild -ba nagios.spec --define 'EMBPERL 1'
```

Install procedure:

1. Check RPM integrity

```
rpm -K nagios-2.0b1-1.i386.rpm nagios-www-2.0b1-1.i386.rpm
freetype-2.1.4-4.0.i386.rpm gd-1.8.4-12.3.1.i386.rpm

nagios-2.0b1-1.i386.rpm: sha1 md5 OK
nagios-www-2.0b1-1.i386.rpm: sha1 md5 OK
freetype-2.1.4-4.0.i386.rpm: (sha1) dsa sha1 md5 gpg OK
gd-1.8.4-12.3.1.i386.rpm: (sha1) dsa sha1 md5 gpg OK
```

2. Install them

```
rpm -Uvh nagios-2.0b1-1.i386.rpm nagios-www-2.0b1-
1.i386.rpm gd-1.8.4-12.3.1.i386.rpm  freetype-2.1.4-4.0.i386.rpm
```

3. Copy prepared Nagios configuration files to `/etc/nagios` directory. For detailed explanation of Nagios configuration options, please check Nagios project documentation [15].

   Edit the main configuration file `nagios.cfg` and set the following options:

   ```
   check_external_commands=1
   ```

   This enables user commands to be executed. User commands are predefined set of actions which Nagios will execute to modify actions specified in the configuration files for specific monitored host or service. Example of such action: disable notifications about problems on the specific server to be sent to system administrators.

   The CGI script will store these commands in the file:

   ```
   command_file=/var/spool/nagios/nagios.cmd
   ```

Both `nagios` and `apache` accounts must have "read", "write", and "delete" access permissions for this file. Therefore, proper ownership and permissions to this directory should be the following:

```
chmod 2750 /var/spool/nagios
chown nagios.apache /var/spool/nagios
```

Disable use of syslog:

```
use_syslog=0
```

It is bad idea to dump Nagios debug information into system log files. This is also bad for performance `{DoS->RCL->SA}`.

Make sure, if no service check will be lost in event if passive checking command is slow or hung

```
check_for_orphaned_services=1
check_service_freshness=1
service_freshness_check_interval=120
```

4. Create "external commands" directory

```
mkdir /var/log/nagios/rw
chown nagios.nagios /var/log/nagios/rw
```

Nagios Plugins

Source files: Source file `nagios-plugins-1.4.tar.gz`, binary RPMs from Red Hat.
Build procedure:

1. Download file `nagios-plugins-1.4.tar.gz`
2. Check the MD5 signature of the file.
3. Build RPMs:

```
rpmbuild --define 'custom 1' -ta nagios-plugins-1.4.tar.gz
```

Install procedure:

1. Check RPM signatures

```
rpm -K nagios-plugins-1.4-1.i386.rpm \
     net-snmp-perl-5.0.9-2.90.1.i386.rpm
nagios-plugins-1.4-1.i386.rpm: sha1 md5 OK
net-snmp-perl-5.0.9-2.90.1.i386.rpm: (sha1) dsa sha1
md5 gpg OK
```

2. Install them:

```
rpm -Uvh net-snmp-perl-5.0.9-2.90.1.i386.rpm
rpm -Uvh --nodeps nagios-plugins-1.4-1.i386.rpm
```

3. Install additional plugins and components to the target directory

```
cp utils.pm /usr/lib/nagios/plugins
chmod 644 /usr/lib/nagios/plugins/utils.??
```

NRPE

Source files: source RPM from DAG repository[49]
Build procedure:

1. Download the source RPM `nagios-nrpe-2.0-3.dag.src.rpm`
2. Check RPM signatures
```
rpm -K nagios-nrpe-2.0-3.dag.src.rpm

nagios-plugins-nrpe-2.0-3.dag.src.rpm: sha1 md5 OK
```
3. Build it
```
rpmbuild -ba nagios-nrpe-2.0-3.dag.src.rpm
```

Install procedure:

1. Check RPM signatures
```
rpm -K nagios-plugins-nrpe-2.0-3.dag.i386.rpm

nagios-plugins-nrpe-2.0-3.dag.i386.rpm: sha1 md5 OK
```
2. Install it:
```
rpm -Uvh nagios-plugins-nrpe-2.0-3.dag.i386.rpm
```

Target Nodes' NRPE Deployment

Source files: binary RPMs compiled on previous steps.
Install procedure:

1. Check RPM signatures
```
rpm -K nagios-plugins-1.4-1.i386.rpm \
nagios-nrpe-2.0-3.dag.i386.rpm[49]

nagios-plugins-1.4-1.i386.rpm: sha1 md5 OK
nagios-nrpe-2.0-3.dag.i386.rpm: sha1 md5 OK
```
2. Create user "`nagios`"
```
useradd -d / -r -s /sbin/nologin nagios
```
3. Install the RPMs
```
rpm -Uvh nagios-plugins-1.4-1.i386.rpm \
    nagios-nrpe-2.0-3.dag.i386.rpm
```
4. Copy prepared NRPE configuration file `nrpe.cfg` into `/etc/nagios` directory
5. Enable NRPE to be started automatically on the system reboot
```
chkconfig nagios on
```

[49] Source RPM was taken from <ftp://fr2.rpmfind.net/linux/dag/fedora/3/en/i386/SRPMS.dag/>

Appendix D HTTPd Configuration Files

Apache HTTPd configuration file (`httpd.conf`)

```
# This is the main Apache server configuration file.
ServerTokens Prod
ServerSignature Off
ServerAdmin root@localhost

ServerRoot "/etc/httpd"
PidFile /var/run/httpd.pid

Timeout 60
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule prefork.c>
StartServers       10
MinSpareServers    5
MaxSpareServers   20
MaxClients       150
MaxRequestsPerChild  1000
</IfModule>

Listen 0.0.0.0:80
ServerName nagios.example.com
UseCanonicalName On

LoadModule access_module modules/mod_access.so
LoadModule auth_module modules/mod_auth.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule dir_module modules/mod_dir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule cgi_module modules/mod_cgi.so

Include conf.d/ssl.conf
Include conf.d/authz_ldap.conf
Include conf.d/nagios.conf

User apache
Group apache

DocumentRoot "/var/www/html"

<Directory />
    Options None
    AllowOverride None
    Deny from all
</Directory>
<Directory "/var/www/html">
```

```
    Options FollowSymLinks -MultiViews -Indexes
    AllowOverride None
    Order allow,deny
    Allow from all
    <LimitExcept GET POST>
        deny from all
    </LimitExcept>
</Directory>

<LocationMatch /(scripts|cgi-local|htbin|cgibin|cgis|cgi/|win-cgi|
  cgi-win|finger\.pl|guestbook\.cgi|campas|files\.pl|count\.cgi|*\~|
  *\.bak|*\.sav|*\.orig|*\.old)/>
   deny from all
</LocationMatch>

DirectoryIndex index.html

AccessFileName /dev/null

TypesConfig /etc/mime.types
DefaultType text/plain

HostnameLookups Off

EnableMMAP on
EnableSendfile on

ErrorLog logs/error_log
LogLevel notice
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
CustomLog logs/access_log combined

Alias /icons/ "/var/www/icons/"
<Directory "/var/www/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Alias /error/ "/var/www/error/"
ErrorDocument 400 /error/400.html
ErrorDocument 401 /error/401.html
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
ErrorDocument 405 /error/405.html
ErrorDocument 408 /error/408.html
ErrorDocument 500 /error/500.html

#EoF#########################################################
```

SSL configuration file (`ssl.conf`)

```
# This is the Apache server configuration file providing SSL support.
#
LoadModule ssl_module modules/mod_ssl.so
Listen 0.0.0.0:443

SSLPassPhraseDialog  builtin

SSLSessionCache         shmcb:/var/cache/mod_ssl/scache(512000)
SSLSessionCacheTimeout  300

SSLMutex  file:logs/ssl_mutex

SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
SSLRandomSeed startup file:/dev/urandom 1024

SSLCryptoDevice builtin

<VirtualHost _default_:443>

DocumentRoot "/var/www/html"
ServerAdmin root@localhost

ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log

SSLEngine on

SSLCipherSuite RSA:!EXP:!NULL:RC4+RSA:+HIGH:+MEDIUM:-LOW:+SSLv2

SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

#EoF#####################################################
```

Auth LDAP configuration file (`authz_ldap.conf`)

```
#
# mod_authz_ldap can be used to implement access control and
# authenticate users against an LDAP database.

LoadModule authz_ldap_module modules/mod_authz_ldap.so

<IfModule mod_authz_ldap.c>
   <Location /nagios>
        AuthzLDAPEngine on
        Satisfy all
        AuthName "IBM Intranet access ID"
        AuthType basic
        AuthzLDAPServer localhost:389
        AuthzLDAPUserBase ou=access,o=example,o=com
        AuthzLDAPUserKey mail
        AuthzLDAPUserScope subtree
        AuthzLDAPLogLevel debug
        AuthzLDAPSetAuthorization off
        AuthzLDAPAuthoritative off

        require valid-user

        #Force all to do SSL
        RewriteEngine on
        RewriteCond %{SERVER_PORT} =80
        RewriteRule ^(.*) https://%{SERVER_NAME}%{REQUEST_URI}
   </Location>
</IfModule>

#EoF#########################################################
```

Nagios UI configuration file (`nagios.conf`)

```
# CONFIG SNIPPETS FOR APACHE WEB SERVER
ScriptAlias /nagios/cgi-bin "/usr/lib/nagios/cgi"
<Directory "/usr/lib/nagios/cgi">
   SSLRequireSSL
   Options FollowSymLinks ExecCGI
   AllowOverride none
   Order allow,deny
   Allow from all
</Directory>

Alias /nagios "/var/www/html/nagios"
<Directory "/var/www/html/nagios">
   Options FollowSymLinks
   AllowOverride none
   Order allow,deny
   Allow from all
</Directory>
#EoF#########################################################
```

Appendix E Modifications of the Standard Startup Scripts

Modifications for the Apache HTTPd startup script.

```
# diff -w /etc/init.d/httpd /etc/init.d/httpd.chroot
10a11
> CHROOT=/chroot/apache
48a50,81
> makeajail () {
> echo ""
> echo "Making chroot ENV:"
> if [ -x $CHROOT/usr/sbin/httpd ]
> then
>     echo "Probably jail is built already"
>     return 0
> else if [ -d $CHROOT/var/log/httpd ]
>     then
>         /bin/mount --bind /var/log/httpd $CHROOT/var/log/httpd
>         /bin/mount --bind /usr/lib  $CHROOT/usr/lib
>         /bin/mount --bind /lib $CHROOT/lib
>         /bin/mount --bind /etc/httpd $CHROOT/etc/httpd
>         /bin/mount --bind /usr/kerberos/lib $CHROOT/usr/kerberos/lib
>         /bin/mount --bind /var/log/nagios $CHROOT/var/log/nagios
>         /bin/mount --bind /etc/nagios $CHROOT/etc/nagios
>         /bin/mount --bind /var/spool/nagios $CHROOT/var/spool/nagios
>     else
>         echo "Cannot find the mount point. Exiting...."
>         return 1
>     fi
> fi
>
> if [ -x $CHROOT/usr/sbin/httpd ]
> then
>     echo "The jail is ready, daemons are welcome!"
>     return 0
> else
>     echo "The jail is not ready, exiting..."
>     return 2
> fi
> }
56c89,91
<         daemon $httpd $OPTIONS
---
>       makeajail || exit 2
>
>       /usr/sbin/chroot $CHROOT $httpd $OPTIONS
```

Modifications for the Nagios Engine startup script.

```
#diff -w nagios.orig /etc/init.d/nagios
113c114,115
<  su - $Nagios -c "touch $NagiosVar/nagios.log $NagiosSav"
---
>  touch $NagiosVar/nagios.log $NagiosSav $NagiosRun && chown
$Nagios.$Nagios $NagiosVar/nagios.log $NagiosSav $NagiosRun
```

Appendix F Configuration Files for Postfix

#master.cf

```
# Postfix master process configuration file.  Each logical line
# describes how a Postfix daemon program should be run.

#==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
#==========================================================================
smtp      inet  n       -       y       -       -       smtpd
pickup    fifo  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr      fifo  n       -       y       300     1       nqmgr
rewrite   unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
flush     unix  n       -       y       1000?   0       flush
proxymap  unix  -       -       y       -       -       proxymap
smtp      unix  -       -       y       -       -       smtp
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
local     unix  -       n       n       -       -       local
```

#main.cf

```
# Global Postfix configuration file.

# All values are default for the
# version postfix-2.0.16-14.RHEL3 , except:

# do not tell him bad news ;-)
double_bounce_sender = nagios

# SMTP VRFY must be disabled
disable_vrfy_command = yes
```

Appendix G Kernel Security Settings in sysctl.conf File

```
# Kernel sysctl configuration file for Red Hat Linux

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core
# filename.
kernel.core_uses_pid = 1

# NETWORK SECURITY PART
# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1

# Controls SYN pool
# Risk code:{DoS->RCL->SA}
net.ipv4.tcp_max_syn_backlog = 4096

# Minor nice-to-have things
net.ipv4.tcp_syncookies = 1

# Ignore echo broadcasts and bogus errors
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Controls redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Disabling source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0

# Controls exec shield
kernel.exec-shield-randomize = 1
kernel.exec-shield = 1

#EoF#########################
```

Appendix H Apache HTTP Server Benchmarks

This is result of the final test by CIS Apache Benchmark Scoring Tool.
All software is deployed; most of the security hardening is completed.
Comments for all "FAILED" results are included.

```
########### CIS Apache Benchmark Scoring Tool 2.08 ############
Version: 2.08
Description: Check Apache configuration file against the CIS Apache
Benchmark.
Copyright 2003-2004, CISecurity. All rights reserved.
################################################################

Level
-----------------------------------------------
Section 1.1     Harden Underlying Operating System
 [PASSED]       Has the Operating System been hardened according to any and
all applicable OS system security benchmark guidance? (Answer: Yes)

Section 1.2     Create the Web Groups
 [FAILED]       Created three dedicated web groups? (Answer: No)
```

COMMENT: N/A[50]. This is not required for Nagios operations of maintenance.

```
Section 1.3     Create the Apache Web User Account
 [PASSED]       The Apache Configuration User (apache) home directory
"/var/www/html" is the same as DocumentRoot for Virtual Host:
_default_:443.

Section 1.4     Lock Down the Apache Web User Account
 [FAILED]       User (apache) has an active shell "/bin/sh".
```

COMMENT: "False Positive". User `apache` has shell `/sbin/nologin`.

```
Section 1.5     Apache Distribution Download
 [FAILED]       Downloaded the Apache source and MD5 Checksums from
httpd.apache.org? (Answer: No)
```

COMMENT: N/A. Software was installed from Red Had RPM.

```
Section 1.6     Verify the MD5 Checksums
 [PASSED]       Verified the Apache MD5 Checksums? (Answer: Yes)

Section 1.7     Apply Current Patches (Applicable to your OS Platform and
Apache Version)
 [PASSED]       Applied the current distribution patches? (Answer: Yes)

Section 1.8     Update the Apache Banner Information
 [FAILED]       Apache banner "Apache" not sufficiently altered. Either
edit the httpd.h file or implement the Mod_Security SecServerSignature
Directive.
```

COMMENT: Cannot be fixed in the installed software.

```
Section 1.9     Configure the Apache Software
 [PASSED]       "mod_imap.c" is not be compiled into Apache.
 [FAILED]       "mod_headers.c" should be compiled into Apache.
```

[50] N/A means "Not Applicable".

COMMENT: N/A. This module was disabled by hardening procedures.

```
 [PASSED]       "mod_status.c" is not be compiled into Apache.
 [PASSED]       "mod_autoindex.c" is not be compiled into Apache.
 [FAILED]       "mod_rewrite.c" should be compiled into Apache.
```

COMMENT: Cannot be fixed in the installed software.

```
 [FAILED]       "mod_auth_digest.c" should be compiled into Apache.
```

COMMENT: N/A. This module was disabled by hardening procedures.

```
 [PASSED]       "mod_userdir.c" is not be compiled into Apache.
 [FAILED]       "mod_vhost_alias.c" should be compiled into Apache.
```

COMMENT: N/A. This module was disabled by hardening procedures.

```
Section 1.10    Compile and Install the Apache Software
 [FAILED]       Compiled and installed Apache distribution? (Answer: No)
```

COMMENT: N/A. Software was installed from Red Had RPM.

```
Section 1.11    Server Oriented General Directives
 [PASSED]       Server type is "standalone"
 [FAILED]       HostnameLookups is off
```

COMMENT: Ignored. Turning "on" this option will impact the server performance.

```
 [FAILED]       HostnameLookups is off for <VirtualHost _default_:443>
```

COMMENT: Ignored. Turning "on" this option will impact the server performance.

```
Section 1.12    User Oriented General Directives
 [PASSED]       User is "apache"
 [PASSED]       Group is "apache"
 [PASSED]       Is the root@localhost address a valid email alias? (Answer:
Yes)

Section 1.13    Denial of Service (DoS) Protective General Directives
 [PASSED]       TimeOut value is "60"
 [PASSED]       KeepAlive value is "On"
 [PASSED]       KeepAliveTimeout is "15"
 [PASSED]       StartServers is "10"
 [PASSED]       MinSpareServers is "5"
 [PASSED]       MaxSpareServers is "10"
 [PASSED]       MaxClients is "256"

Section 1.14    Web Server Software Obfuscation General Directives
 [PASSED]       ServerTokens is "Prod"
 [PASSED]       ServerSignature is "Off"
 [PASSED]       ErrorDocument is set for status code "400".
 [PASSED]       ErrorDocument is set for status code "401".
 [PASSED]       ErrorDocument is set for status code "403".
 [PASSED]       ErrorDocument is set for status code "404".
 [PASSED]       ErrorDocument is set for status code "405".
 [PASSED]       ErrorDocument is set for status code "500".

Section 1.15    Web Server Fingerprinting
 [FAILED]       No fake headers have been specified.
```

COMMENT: Cannot be fixed in the installed software.

```
Section 1.16    Intrusion Detection Options
```

```
 [FAILED]       Are fake CGI scripts used? (Answer: No)
```
COMMENT: Ignored because the risk of implementing of the proposed solution is more than not implementing it.

```
 [PASSED]       LocationMatch is used to limit scans
 [FAILED]       ScriptAliasMatch is not used
```
COMMENT: Ignored.

```
Section 1.17    Mod_Security
 [FAILED]       Module mod_security is not compiled into apache binary.
```
COMMENT: Cannot be fixed in the installed software because this module is not a part of the Red Hat RPM package. See comments for this module in the chapter "Summary and Research".

```
Section 1.18    Access Control Directives
 [FAILED]       Directory entry for "/" is not properly configured.
```
COMMENT: Probable "N/A". The "/" is configured according to the CIS Apache Benchmark documents [14].

```
Section 1.19    Authentication Mechanisms
 [PASSED]       Have you implemented SSL to encrypt the Basic Auth Session?
(Answer: Yes)

Section 1.20    Directory Functionality/Features Directives
 [PASSED]       Option directive "Includes" for directory "/var/www/html"
is disabled.
 [PASSED]       Option directive "MultiViews" for directory "/var/www/html"
is disabled.
 [PASSED]       Option directive "Indexes" for directory "/var/www/html" is
disabled.
 [FAILED]       Remove Option directive "FollowSymLinks" for directory
"/var/www/html" unless required.
```
COMMENT: This is required for the server performance.

```
Section 1.21    Limiting HTTP Request Methods
 [FAILED]       LimitExcept directive on Virtual Host "_default_:443" is
not properly set for GET and POST.
```
COMMENT: "False Positive".

```
 [PASSED]       LimitExcept directive on "/var/www/html" is properly set
for GET and POST.

Section 1.22    Logging General Directives
 [PASSED]       LogLevel is set to "notice".

Section 1.23    Remove Default/Unneeded Apache Files
 [VERIFY]       Verify DocumentRoot "/var/www/html" files (1) are not
default Apache files.
 [SKIPPED]      User "apache" home directory (/var/www/html) does not
exist.

Section 1.24    Update Ownership and Permissions for Enhanced Security
 [FAILED]       Owner of Server Conf directory "/etc/httpd/conf/" should be
root.
```
COMMENT: "False Positive".

```
 [VERIFY]       Server Conf directory "/etc/httpd/conf/" group is properly
set.
 [PASSED]       Permissions on Server Conf directory "/etc/httpd/conf/" set
to "660".
 [PASSED]       Document Root "/var/www/html" group is "root".
 [FAILED]       Permissions on Document Root "/var/www/html" should be
"664".
```

COMMENT: "False Positive".

```
 [PASSED]       Owner of Document Root "/var/www/html" is root.
 [FAILED]       Log directory "logs" does not exist.
```

COMMENT: "False Positive".

```
 [FAILED]       CGI directory "" does not exist.
```

COMMENT: "False Positive".

```
 [FAILED]       Server Bin directory "/etc/httpd/bin/" does not exist.
```

COMMENT: "False Positive".

```
Section 1.25    Update the Apachectl Script for Email Notification
 [FAILED]       Updated the default apachectl start script's code to send
alerts to the appropriate personnel? (Answer: No)
```

COMMENT: N/A. This is not required for Nagios operations of maintenance.

```
-------------------------------------------------------------------------

 [Apache Benchmark Score: 5.81 out of 10.00]
```

Appendix I Nessus Scan Report

This is result of the final test by Nessus Security Scanner.
All software is deployed; most of the security hardening is completed.
Report is formatted. Some miscellaneous information is removed.

```
SUMMARY
 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 1
 - Number of security warnings found : 2
 - Number of security notes found : 17

TESTED HOSTS  nagios.example.com (Security holes found)

DETAILS
+ nagios.example.com :
 . List of open ports :
   o ssh (22/tcp) (Security hole found)
   o http (80/tcp) (Security warnings found)
   o https (443/tcp) (Security warnings found)
   o general/tcp (Security notes found)
   o ntp (123/udp) (Security notes found)
   o general/udp (Security notes found)

 . Vulnerability found on port ssh (22/tcp) :
You are running a version of OpenSSH which is older than 3.7.1
Versions older than 3.7.1 are vulnerable to a flaw in the buffer management
functions which might allow an attacker to execute arbitrary commands on
this host. An exploit for this issue is rumored to exist.

Note that several distribution patched this hole without changing the
version number of OpenSSH. Since Nessus solely relied on the banner of the
remote SSH server to perform this check, this might be a false positive.
<...skipped...>

 . Information found on port ssh (22/tcp)
    Remote SSH version : SSH-2.0-OpenSSH_3.6.1p2
    Remote SSH supported authentication :
     publickey,password,keyboard-interactive

 . Information found on port ssh (22/tcp)
   The remote SSH daemon supports the following versions of the SSH
protocol:
      . 1.99
      . 2.0

    SSHv2 host key fingerprint : <...sanitized...>

. Warning found on port http (80/tcp)

It seems that your web server tries to hide its version or name, which is a
good thing. However, using a special crafted request, Nessus was able to
determine that is is running :
     Apache

Risk factor : None
```

```
Solution : Fix your configuration.

. Information found on port http (80/tcp)
    Nessus was not able to reliably identify this server. It might be:
    IBM_HTTP_SERVER/1.3.19.3  Apache/1.3.20 (Win32)
    The fingerprint differs from these known signatures on 11 point(s)

 . Warning found on port https (443/tcp)

It seems that your web server tries to hide its version or name, which is a
good thing. However, using a special crafted request, Nessus was able to
determine that is is running :
     Apache

    Risk factor : None
    Solution : Fix your configuration.

 . Information found on port https (443/tcp)

    A SSLv2 server answered on this port

 . Information found on port https (443/tcp)

    Nessus was not able to reliably identify this server. It might be:
    IBM_HTTP_SERVER/1.3.19.3  Apache/1.3.20 (Win32)
    The fingerprint differs from these known signatures on 11 point(s)

 . Information found on port https (443/tcp)

    Here is the SSLv2 server certificate:

    Certificate: <...skipped...>

Here is the list of available SSLv2 ciphers:
RC4-MD5
EXP-RC4-MD5
RC2-CBC-MD5
EXP-RC2-CBC-MD5
DES-CBC-MD5
DES-CBC3-MD5
RC4-64-MD5
The SSLv2 server offers 5 strong ciphers, but also 0 medium strength and 2
weak "export class" ciphers. The weak/medium ciphers may be chosen by an
export-grade or badly configured client software. They only offer a
limited protection against a brute force attack

Solution: disable those ciphers and upgrade your client software if
necessary.
See http://support.microsoft.com/default.aspx?scid=kb;en-us;216482
or http://httpd.apache.org/docs-2.0/mod/mod_ssl.html#sslciphersuite
This SSLv2 server also accepts SSLv3 connections.
This SSLv2 server also accepts TLSv1 connections.

 . Information found on port general/tcp
    TCP split NIDS evasion function is enabled. Some tests might
    run slowly and you may get some false negative results

 . Information found on port general/tcp
```

```
    TCP fake RST NIDS evasion function is enabled. Some tests might
    run slowly and you may get some false negative results.

 . Information found on port general/tcp
    HTTP NIDS evasion functions are enabled.
    You may get some false negative results

 . Information found on port general/tcp
    The remote host is running Linux Kernel 2.4.21-27.0.2.ELhugemem (i386)

 . Information found on port general/tcp
    Nessus was not able to reliably identify the remote operating system.
It
     might be:
    IPCop (Linux Kernel 2.4 firewall)
    IBM OS/400
    Netilla Service Platform 4.0
    The fingerprint differs from these known signatures on 8 points.
<...skipped...>

. Information found on port general/tcp
    10.0.2.133 resolves as nagios.example.com.

 . Information found on port ntp (123/udp)
    It is possible to determine a lot of information about the remote host
    by querying the NTP (Network Time Protocol) variables - these include
    OS descriptor, and time settings.

It was possible to gather the following information from the remote NTP
host:
    version='ntpd 4.1.2@1.892 Tue Feb 24 06:32:25 EST 2004 (1)',
    processor='i686', system='Linux2.4.21-27.0.2.ELhugemem', leap=0,
    stratum=3, precision=-18, rootdelay=78.964, rootdispersion=19.377,
    peer=49836, refid=10.0.0.1, reftime=0xc5ca5606.5fc83a96, poll=6,
    clock=0xc5ca5621.0068db8b, state=4, offset=0.340, frequency=8.113,
    jitter=0.122, stability=0.023

    Quickfix: Set NTP to restrict default access to ignore all info
packets:
     restrict default ignore

    Risk factor : Low

 . Information found on port general/udp
    For your information, here is the traceroute to 10.0.2.133:
    10.0.1.14
    10.0.2.133

-------------------------------------------------------
This file was generated by the Nessus Security Scanner
```