



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

Check out the list of upcoming events offering  
"Securing Linux/Unix (Security 506)"  
at <http://www.giac.org/registration/gcux>

# SANS Network Security 2000, GIAC Practical: Track 6, UNIX Security

---

## List of contents:

- Executive Summary
- Introduction
- Brief History
- Physical Access Security
- System Access Security
- Analysis of Operation System & Services
  - Patches
  - Stripping down OS
  - Hardening tools
  - Stripping down services (Nessus)
  - Stripping down services in general
  - User access control
  - Logging
- Third-party software
- Backup
- File Integrity
- Vulnerabilities
- Recommendations
- Conclusion
- Refence list

---

## Executive Summary

The purpose of this audit was to identify & examine weaknesses in the configuration of our DHCP server, called *dhcp01* (SUN Enterprise Ultra 250 with Solaris 7 installed). The server is not a total show stopper for the company in case of break-down, but it is an extremely important server in our infrastructure.

All Windows machines and printers are depending on the DHCP service from it.

Even though the server is behind a coporate firewall and in that case is not directly accessible to outside hackers, the security level on the server *dhcp01* is below avarage. Therefore the issues in this audit should be addressed as soon as possible.

I have made some recommendations in order to bring security within acceptable limits. Some of the more common vulnerabilities are found within the configuration of the Operating System (OS) and in administrative working methods. I have used Nessus (a remote security scanner) to see if there were any security leaks on the server.

The company has some security policies, but these are not inforced. That is off course a problem. There are no written policies about backup, disaster recovery etc. In case of a emergency the server and the company

will be very vulnerability because the administrative practices are not well planned and tested.

The DHCP server *dhcp01* moved directly from test into operation. Not enough time for test and verification. That means that the server is not configured correctly. No unnecessary services has been disabled. No hardened programs has been run on the server. Another big problem, the personal was not trained to actually handle the DHCP server. Additional training is recommended.

In conclusion the server has too many open services and improperly trained administrators maintaining it, which adds up to a potential high level of security vulnerability.

[Back to top](#)

---

## Introduction

During this audit I will identify & examine weaknesses in the configuration of our DHCP server, called *dhcp01* (SUN Enterprise Ultra 250 with Solaris 7 installed). After a brief history concerning the server in focus I will look upon physical- and system access security. Then I will go into details about an analysis of the operation system, network based services, third-party software, backup and file integrity.

In the light of the analysis, I will list a prioritized list of security vulnerabilities and recommendations.

[Back to top](#)

---

## Brief History

Some time ago, we needed a DHCP solution. A SUN Enterprise Ultra 250 machine was bought for testing purposes. We chose Cisco Network Registrar (CNR) as the DHCP server software. Then after a while the CNR server *dhcp01* was directly moved from test into operation. This was never the intention. The test and verification was not completed. The server was still running on demo software and there was not any trained personal to actually handle the CNR. The DHCP server was not configured correctly and so on. But we needed the service here and now (a management decision) and at the time there was no alternative.

[Back to top](#)

---

## Physical Access Security

All operators have access to the main server room, this is controlled by a smart-card lock. So everyone who enters the door will be logged. But that also means that operators without or very little knowledge have physical access to the various servers (NT/UNIX).

There is no "janitor" in the server room who knows information like how much power is used for each server, which network cables belongs to which machines etc. That could mean that operators are plugging in or unplugging servers and could cause a lot of harm (downtime etc.). A plan or a logbook would be a very

good idea as every change in that room should be logged.

The physical security could be improved if you limited the number of operators who have access to the server room. Or perhaps splitting the server room up into more parts, one for NT and one for UNIX with locks on the doors. Therefore only people with the correct knowledge could enter the correct room to configure the servers!

[Back to top](#)

---

## System Access Security

The DHCP server has the same root password as almost all of the servers, but because it is a very important server for our infrastructure, the password should be changed to something else. Only a few administrators should have access to the machine as root.

SSH is installed on the server, but still both 'rsh' and 'rlogin' are enabled in `/etc/inetd.conf`. These kind of services should be disabled. Only through SSH should it be allowed to login to the server.

Right now the only user that is allowed to login is root. But that has to be changed because as a result you would lose trace ability. Use instead ssh as local user and su to root:

For example: `ssh dhcp01 -l root`

Then you are logging in as a local user with root privileges. Your public key has to be inserted in the `authorized_keys` file in the ssh directory on *dhcp01*. Everything you do will then be logged as you, therefore you have the trace-ability. The login procedure can be done with out knowing the root password, so it could be disabled, but I would suggest keeping it for login as console.

We therefore make sure that the password is hard to break. For instance try running crack on root's account on *dhcp01*.

If sometime in the future another user other than 'root' needs to login on *dhcp01*, I would suggest using 'sudo' instead of enabling rlogin or telnet. Sudo is a program that allows non-root users to grant access to the administrative account and execute programs "as root".

The SUID (Set-UID) bit on the program 'su' should be removed, since it is not being used.

Another good thing is to prevent unauthorized people to just reboot the machine and then change the root password. A way to prevent that is to set the EEPROM security mode (firmware security level):

```
# eeprom security-mode=command
```

That means every time you make a normal reboot, or try to boot in single user mode, you will be prompted for a password. If you forget the password, you have to get a new EEPROM. There is no way to recover the password, you have to get a new EEPROM from Sun.

[Back to top](#)

---

# Analysis of Operation System & Services

Background for the analysis.

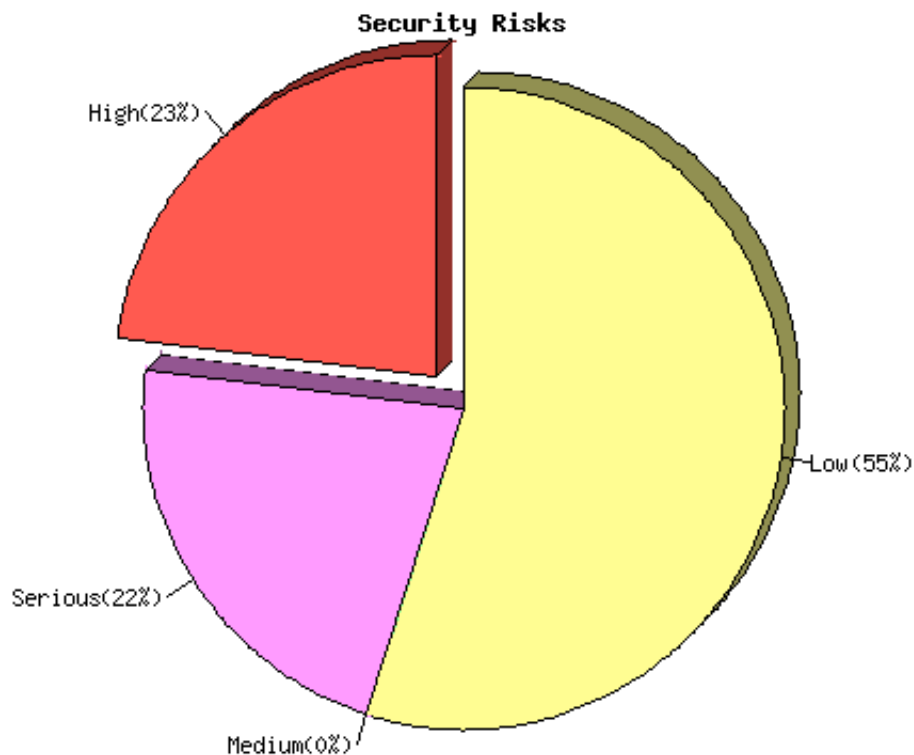
Many of the vulnerabilities are found with [Nessus version 1.0.6](#) and are described in details later in the report.

The Nessus Security Scanner was used to assess the security of 1 host

7 security warnings have been found

2 security notes have been found

Repartition of the level of the security problems (taking from Nessus) :



The DHCP server *dhcp01* is running on Sun Solaris 7. The server was installed in early in 1999. Since it went from test/demo to production, nothing has been done.

Potential problems:

- No patches installed.
- No hardning script has be run.

[Back to top](#)

## Patches

The output from 'showrev -p' on *dhcp01*

```
# showrev -p
No patches are installed
#
```

That means of course that you must do some patching immediately. I would install the latest '[Sun Recommended Patch Cluster](#)' (from Sun) to start with. It is not always the latest patches that are in the Patch Cluster. So I advise to checkout [Patch Report](#), that contains information about all patches available for a specific release.

It is not a good idea to just install brand new patches. It is better to wait a week or two to see if any updates are coming to the specific patch. Special patches should only be applied to system, if Sun specific ask you to do it.

Here is contents of the November Patch Cluster:

```
# CLUSTER_README
```

```
NAME: Solaris 7 Recommended Patch Cluster
DATE: Nov/02/00
```

### CLUSTER DESCRIPTION

-----

*These Solaris Recommended patches are considered the most important and highly recommended patches that avoid the most critical system, user, or security related bugs which have been reported and fixed to date. In most cases a Solaris security patch will be included in the recommended patch set. It is possible, however, that a security patch may not be included in the recommended set if it is determined to be a more obscure application specific issue and not generally applicable.*

*During initial installation of the Solaris product other patches or patch sets may be provided with the product and required with product installation. Refer to the Solaris product installation documentation to be sure that all the patches required at product installation are already installed. This patch cluster can then be used to update or augment the system with the recommended patches included.*

### PATCHES INCLUDED:

-----

*106960-01 SunOS 5.7: Manual Pages for patchadd.1m and patchrm.1m*

107038-01 SunOS 5.7: *apropos/catman/man/whatis patch*  
107171-06 SunOS 5.7: *Fixes for patchadd and patchrm*  
106793-05 SunOS 5.7: *ufsdump and ufsrestore patch*  
106934-03 CDE 1.3: *libDtSvc Patch*  
106725-02 OpenWindows 3.6.1: *mailtool vacation security patch*  
107544-03 SunOS 5.7: */usr/lib/fs/ufs/fsck patch*  
109104-04 SunOS 5.7: */kernel/fs/sockfs patch*  
106541-12 SunOS 5.7: *Kernel update patch*  
107587-01 SunOS 5.7: */usr/lib/acct/lastlogin patch*  
107359-02 SunOS 5.7: *Patch for SPARCompiler Binary Compatibility Libraries*  
107636-05 SunOS 5.7: *X Input & Output Method patch*  
107887-10 CDE 1.3: *Actions Patch*  
106944-03 SunOS 5.7: */kernel/fs/fifofs and /kernel/fs/sparcv9/fifofs patch*  
106952-01 SunOS 5.7: */usr/bin/uux patch*  
107456-01 SunOS 5.7: */etc/nsswitch.dns patch*  
106978-10 SunOS 5.7: *sysid patch*  
107115-05 SunOS 5.7: *LP patch*  
107259-01 SunOS 5.7: */usr/sbin/vold patch*  
107451-05 SunOS 5.7: */usr/sbin/cron patch*  
107454-05 SunOS 5.7: */usr/bin/ftp patch*  
107684-01 SunOS 5.7: *Sendmail patch*  
107792-02 SunOS 5.7: */usr/bin/pax patch*  
107972-01 SunOS 5.7: */usr/sbin/static/rcp patch*  
108301-02 SunOS 5.7: */usr/sbin/in.tftpd patch*  
107337-01 OpenWindows 3.6.1: *KCMS configure tool has a security vulnerability*  
107893-09 OpenWindows 3.6.1: *Tooltalk patch*  
108219-01 CDE 1.3: *dtaction Patch*  
108221-01 CDE 1.3: *dtspcd Patch*  
107885-06 CDE 1.3: *dtprintinfo Patch*  
108482-02 SunOS 5.7: */usr/sbin/snoop patch*  
108662-01 SunOS 5.7: *Patch for sadmind*  
107709-07 SunOS 5.7: *libssasmp/libssagent/snmpdx/mibiisa patch*  
108484-01 SunOS 5.7: *aset patch*  
108721-01 SunOS 5.7: *admintool patch*  
106950-13 SunOS 5.7: *Linker patch*  
106938-04 SunOS 5.7: *libresolv patch*  
107018-02 SunOS 5.7: */usr/sbin/in.named patch*  
108376-16 OpenWindows 3.6.1: *Xsun Patch*  
108331-01 SunOS 5.7: */usr/bin/uustat patch*  
108327-01 SunOS 5.7: */usr/bin/cu patch*  
109253-01 SunOS 5.7: */usr/bin/mail patch*  
109404-01 SunOS 5.7: */usr/vmsys/bin/chkperm patch*  
108798-01 SunOS 5.7: */usr/bin/tip patch*  
108838-02 SunOS 5.7: *allocate/mkdevmaps/mkdevalloc patch*  
107650-08 OpenWindows 3.6.1: *X11R6.4 Xprint Extension Patch*  
109949-01 SunOS 5.7: *jserver buffer overflow*  
107794-01 SunOS 5.7: *ASET patch*  
106942-07 SunOS 5.7: *libnsl, rpc.nisd and nis\_cachemgr patch*  
109744-01 SunOS 5.7: */usr/lib/nfs/nfsd patch*  
108374-04 CDE 1.3: *libDtWidget Patch*  
106646-03 SNC 3.2: *rpc.pcnfsd has security problem, also hangs and dumps core*

106327-08 SunOS 5.7: Shared library patch for C++  
106300-09 SunOS 5.7: Shared library patch for 64bit C++  
107443-12 SunOS 5.7: packaging utilities patch  
107022-06 CDE 1.3: Calendar Manager patch  
107200-12 CDE 1.3: dtmail patch

There is a need for some Patch-policies to be created, as normally it is not a good idea just to install patches, this can cause damage on a running system or introduce old bugs etc

Therefore you have to divide the patches up into three different levels.

1. Major risk : The system become vulnerable. The necessary patches should be installed at once
2. Medium risk : The system can potentially crash etc. The necessary patches should be tested before installed.
3. Low risk : "Nice to have" etc. Wait for the next Recommended Patch Cluster.

Subscribing to different mailing lists to know which security problems can cause damage on your system is always a good idea, as well as trying to find the right patch to fit the problem. An example of these mailing lists could be [BugTraq](#) and [Focus-sun](#).

[Back to top](#)

## Stripping down OS

All SUID (Set-UID) bit programs should be "removed" from the system, since they are not being used. Only root is allowed to login.

Here is a script that will find all SUID programs on the system and remove the correct bits:

```
#!/bin/sh
echo "Disabling all suid/gid bits"
find / -perm -4000 -type f -exec chmod u-s {} \;
find / -perm -2000 -type f -exec chmod g-s {} \;
echo "Applying suid to (some) files"
chmod u+s /usr/bin/su
chmod u+s /sbin/passwd
echo "Securing writable dirs"
find / -type d -perm -0007 -exec chmod o-w {} \;
chmod 1777 /var/tmp /tmp
chmod 777 /usr/share/man/cat*.Z /var/preserve
#
```

Here is the list of all SUID programs on *dhcp01* that should be removed:

```
# find / -perm -4000 -type f -print | xargs ls -la
-r-sr-xr-x  1 lp      lp          203 Sep 11  1998
/etc/lp/alerts/printer
-r-s--x--x  1 root    sys        351872 Sep 11  1998
/usr/bin/admintool
```



-rwsr-xr-x	1	root	sys	35916	Oct	6	1998	/usr/bin/at
-rwsr-xr-x	1	root	sys	13996	Oct	6	1998	/usr/bin/atq
-rwsr-xr-x	1	root	sys	12704	Oct	6	1998	/usr/bin/atrm
-r-s--x--x	1	root	lp	10400	Sep	1	1998	/usr/bin/cancel
-r-sr-xr-x	1	root	sys	36684	Sep	1	1998	/usr/bin/chkey
-r-sr-xr-x	1	root	bin	17044	Oct	6	1998	/usr/bin/crontab
---s--x--x	1	root	uucp	71156	Sep	1	1998	/usr/bin/ct
---s--x--x	1	uucp	uucp	84588	Sep	1	1998	/usr/bin/cu
-r-sr-xr-x	1	root	bin	14352	Oct	6	1998	/usr/bin/eject
-r-sr-xr-x	1	root	bin	28776	Oct	6	1998	
/usr/bin/fdformat								
-r-sr-xr-x	1	root	bin	29292	Oct	6	1998	/usr/bin/login
-r-s--x--x	1	root	lp	22524	Sep	1	1998	/usr/bin/lp
-r-s--x--x	1	root	lp	6920	Sep	1	1998	/usr/bin/lpset
-r-s--x--x	1	root	lp	20884	Sep	1	1998	/usr/bin/lpstat
-rwsr-xr-x	1	root	sys	7736	Oct	6	1998	/usr/bin/newgrp
-r-sr-sr-x	3	root	sys	99640	Oct	6	1998	
/usr/bin/nispasswd								
-r-sr-sr-x	3	root	sys	99640	Oct	6	1998	/usr/bin/passwd
-r-sr-xr-x	1	root	bin	21368	Oct	6	1998	/usr/bin/rcp
-r-sr-xr-x	1	root	bin	56280	Oct	6	1998	/usr/bin/rdist
-r-sr-xr-x	1	root	bin	16772	Oct	6	1998	/usr/bin/rlogin
-r-sr-xr-x	1	root	bin	9332	Oct	6	1998	/usr/bin/rsh
-r-sr-xr-x	1	root	sys	27628	Oct	6	1998	
/usr/bin/sparcv7/ps								
-r-sr-xr-x	2	root	bin	11528	Oct	6	1998	
/usr/bin/sparcv7/uptime								
-r-sr-xr-x	2	root	bin	11528	Oct	6	1998	
/usr/bin/sparcv7/w								
-r-sr-xr-x	1	root	sys	36456	Sep	1	1998	
/usr/bin/sparcv9/ps								
-r-sr-xr-x	2	root	bin	15584	Sep	1	1998	
/usr/bin/sparcv9/uptime								
-r-sr-xr-x	2	root	bin	15584	Sep	1	1998	
/usr/bin/sparcv9/w								
-r-sr-xr-x	1	root	sys	17976	Oct	6	1998	/usr/bin/su
-rws--x--x	1	uucp	bin	56140	Oct	6	1998	/usr/bin/tip
---s--x--x	1	uucp	uucp	68584	Sep	1	1998	/usr/bin/uucp
---s--x--x	1	uucp	uucp	23612	Sep	1	1998	/usr/bin/uuglist
---s--x--x	1	uucp	uucp	20504	Sep	1	1998	/usr/bin/uuname
---s--x--x	1	uucp	uucp	63336	Sep	1	1998	/usr/bin/uustat
---s--x--x	1	uucp	uucp	72016	Sep	1	1998	/usr/bin/uux
-r-sr-xr-x	1	root	bin	6264	Sep	1	1998	
/usr/bin/volcheck								
-r-sr-xr-x	1	root	bin	11176	Sep	1	1998	
/usr/bin/volrmmount								
-r-sr-sr-x	3	root	sys	99640	Oct	6	1998	
/usr/bin/yppasswd								
-r-sr-sr-x	1	root	sys	24356	Sep	12	1998	

/usr/dt/bin/dtaction							
-r-sr-xr-x	1	root	bin	36180	Sep 12	1998	
/usr/dt/bin/dtappgather							
-r-sr-xr-x	1	root	bin	357220	Sep 12	1998	
/usr/dt/bin/dtprintinfo							
-r-sr-xr-x	1	root	bin	147876	Sep 12	1998	
/usr/dt/bin/dtsession							
-r-sr-sr-x	1	root	daemon	311576	Sep 12	1998	
/usr/dt/bin/sdtcm_convert							
-rwsr-xr-x	1	root	adm	5304	Sep 1	1998	
/usr/lib/acct/accton							
-r-sr-xr-x	1	root	bin	15260	Oct 6	1998	
/usr/lib/fs/ufs/ufsquota							
-r-sr-sr-x	1	root	tty	163892	Oct 6	1998	
/usr/lib/fs/ufs/ufsdump							
-r-sr-xr-x	1	root	bin	848088	Oct 6	1998	
/usr/lib/fs/ufs/ufsrestore							
-r-s--x--x	1	root	bin	19564	Sep 1	1998	
/usr/lib/lp/bin/netpr							
---s--x--x	1	root	bin	4316	Oct 6	1998	
/usr/lib/pt_chmod							
-r-sr-xr-x	1	root	bin	643464	Sep 1	1998	
/usr/lib/sendmail							
-r-sr-xr-x	1	root	bin	8576	Oct 6	1998	
/usr/lib/utmp_update							
---s--x--x	1	uucp	uucp	6156	Sep 1	1998	
/usr/lib/uucp/remote.unknown							
---s--x--x	1	uucp	uucp	168020	Sep 1	1998	
/usr/lib/uucp/uucico							
---s--x--x	1	uucp	uucp	34408	Sep 1	1998	
/usr/lib/uucp/uusched							
---s--x--x	1	uucp	uucp	84904	Sep 1	1998	
/usr/lib/uucp/uuxqt							
-rws--x--x	1	root	15	1913780	Jun 12	1998	
/usr/local/bin/ssh1							
-r-sr-sr-x	1	root	bin	14908	Aug 14	1998	
/usr/openwin/bin/ff.core							
-rwsr-sr-x	1	root	bin	94044	Jul 9	1998	
/usr/openwin/bin/kcms_calibrate							
-rwsr-sr-x	1	root	bin	27752	Jul 9	1998	
/usr/openwin/bin/kcms_configure							
-rwsrwxr-x	1	root	bin	42904	Aug 4	1998	
/usr/openwin/bin/sys-suspend							
-rwsrwxr-x	1	root	bin	67920	Aug 27	1998	
/usr/openwin/bin/xlock							
-rwsr-xr-x	1	root	bin	24176	Aug 25	1998	
/usr/openwin/lib/mkcookie							
-r-sr-xr-x	1	root	bin	57124	Aug 11	1998	
/usr/sbin/afbconfig							

```

-rwsr-xr-x   3 root   bin   17584 Oct  6  1998
/usr/sbin/allocate
-rwsr-xr-x   3 root   bin   17584 Oct  6  1998
/usr/sbin/deallocate
-r-sr-xr-x   1 root   bin   56152 Aug 11  1998
/usr/sbin/ffbconfig
-rwsr-xr-x   3 root   bin   17584 Oct  6  1998
/usr/sbin/list_devices
-r-s--x--x   1 root   lp    6940 Sep  1  1998 /usr/sbin/lpmove
-r-sr-xr-x   1 root   bin  28592 Aug 11  1998
/usr/sbin/m64config
-rwsr-xr-x   1 root   bin   9992 Oct  6  1998
/usr/sbin/mkdevalloc
-rwsr-xr-x   1 root   bin  10244 Oct  6  1998
/usr/sbin/mkdevmaps
-r-sr-xr-x   1 root   bin  20404 Oct  6  1998 /usr/sbin/ping
-r-sr-xr-x   1 root   bin  14420 Sep  1  1998
/usr/sbin/pmconfig
-rwsr-xr-x   1 root   sys  23048 Oct  6  1998 /usr/sbin/sacadm
-r-sr-xr-x   1 root   bin  12948 Oct  6  1998
/usr/sbin/sparcv7/whodo
-r-sr-xr-x   1 root   bin  17472 Sep  1  1998
/usr/sbin/sparcv9/whodo
-r-sr-xr-x   1 root   bin  691620 Sep  1  1998
/usr/sbin/static/rcp
-r-sr-xr-x   1 root   bin  22056 Oct  6  1998
/usr/sbin/traceroute
-r-sr-xr-x   1 root   sys  23060 Sep  1  1998
/usr/ucb/sparcv7/ps
-r-sr-xr-x   1 root   sys  31624 Sep  1  1998
/usr/ucb/sparcv9/ps
-rwsr-sr-x   1 bin    bin  10080 Sep  1  1998
/usr/vmsys/bin/chkperm
#

```

[Back to top](#)

## Hardening tools

As part of an installation of a very important server, I would suggest running [YASSP](#) (Yet Another Solaris Security Package) by Jean Chouanard. YASSP is a tool for automatically hardening a Solaris machine. This incorporates automatic fix-modes. Fix-modes is a script written by Casper Dik. It contains a huge number of file permission improvements for most Solaris versions.

[Back to top](#)

## Stripping down services (Nessus)

As mentioned above, Nessus found 7 security warnings and 2 security notes. Here is the output with recommendations on the security warnings

List of open ports :

1. ssh (22/tcp) (Security warnings found)
2. domain (53/tcp) (Security warnings found)
3. login (513/tcp) (Security warnings found)
4. shell (514/tcp) (Security warnings found)
5. general/tcp (Security notes found)
6. general/udp (Security notes found)
7. unknown (32801/udp) (Security warnings found)
8. lockd (4045/udp) (Security warnings found)
9. general/icmp (Security warnings found)

1. Warning found on port ssh (22/tcp)

You are running a version of SSH which is older than (or as old as) version 1.2.27. If this version was compiled against the RSAREF library, then it is very likely to be vulnerable to a buffer overflow which may be exploited by a cracker to gain root on your system.

To determine if you compiled ssh against the RSAREF library, type 'ssh -V' on the remote host.

**Risk factor : High**

**Solution : Use ssh 2.x, or do not compile ssh against the RSAREF library**

**CVE : CVE-1999-0834** (link to [Common Vulnerabilities and Exposures](#))

Information found on port ssh (22/tcp)

Remote SSH version : ssh-1.5-1.2.25

Recommendation: Upgrade ssh v. 1.2.25 to a newer version at once. I would recommend [Open ssh](#)

2. Warning found on port domain (53/tcp)

The remote nameserver allows recursive queries to be performed by the host running nessusd.

If this is your internal nameserver, then forget this warning.

If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as `www.nessus.org`).

This allows hackers to do cache poisoning attacks against this nameserver.

**Risk factor : Serious**

**Solution : Restrict recursive queries to the hosts that should use this nameserver (such as those of the LAN connected to it).**

**If you are using bind 8, you can do this by using the instruction**

**'allow-recursive' in the 'options' section of your named.conf**

**If you are using another nameserver, consult its documentation.**

Information found on port domain (53/tcp)

The remote bind version is : 8.2.2-P5-NOESW

Recommadation:

This is ok as the DHCP server is an internal nameserver. It is not a normal bind that is running, it is Cisco Network Registrar's bind. I would recommend to limit the numbers of nameservers that can do recursive queries this can be done within CNR.

3. Warning found on port login (513/tcp)

The rlogin service is running.

This service is dangerous in the sense that it is not ciphered - that is, everyone can sniff the data that passes between the rlogin client and the rlogin server. This includes logins and passwords.

You should disable this service and use openssh instead (`www.openssh.com`)

**Risk factor : Low**

**Solution : Comment out the 'rlogin' line in /etc/inetd.conf.**

**CVE : CAN-1999-0651** (link to [Common Vulnerabilities and Exposures](#))

Recommadation: Comment out the 'login' line in /etc/inetd.conf. Use ssh instead.

#### 4. Warning found on port shell (514/tcp)

The rsh service is running.

This service is dangerous in the sense that it is not ciphered - that is, everyone can sniff the data that passes between the rsh client and the rsh server. This includes logins and passwords.

You should disable this service and use ssh instead.

**Risk factor : Low**

**Solution : Comment out the 'rsh' line in /etc/inetd.conf.**

**CVE : CAN-1999-0651** (link to [Common Vulnerabilities and Exposures](#))

Recommadation: Comment out the 'shell' line in /etc/inetd.conf. Use ssh instead.

#### 5. Information found on port general/tcp

Nmap found that this host is running Solaris 2.6 - 2.7 with tcp\_strong\_iss=2

#### 6. Information found on port general/udp

For your information, here is the traceroute to 172.24.88.60 :?

#### 7. Warning found on port unknown (32801/udp)

The statd RPC service is running.

This service has a long history of security holes, so you should really know what you are doing if you decide to let it run.

\* NO SECURITY HOLE REGARDING THIS PROGRAM HAVE BEEN TESTED, SO THIS MIGHT BE A FALSE POSITIVE \*

We suggest you to disable this service.

**Risk factor : High**

**CVE : CVE-1999-0018** (link to [Common Vulnerabilities and Exposures](#))

Recommendation: Comment out the start of 'statd' in /etc/init.d/nfs.client

8. Warning found on port lockd (4045/udp)

The nlockmgr RPC service is running.  
If you do not use this service, then  
disable it as it may become a security  
threat in the future, if a vulnerability  
is discovered.

**Risk factor : Low**

**CVE : CAN-2000-0508** (link to [Common Vulnerabilities and Exposures](#))

Recommendation: Comment out the start of 'lockd' in /etc/init.d/nfs.client

9. Warning found on port general/icmp

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

This may help him to defeat all your  
time based authentications protocols.

**Solution : filter out the icmp timestamp  
requests (13), and the outgoing icmp  
timestamp replies (14).**

**Risk factor : Low**

**CVE : CAN-1999-0524** (link to [Common Vulnerabilities and Exposures](#))

Warning found on port general/icmp

The remote host answered to an ICMP\_MASKREQ query and sent us its netmask.

An attacker can use this information to understand how your network is set up and how the routing is done. This may help him to bypass your filters.

**Solutioin : reconfigure the remote host so that it does not answer to those requests. Set up filters that deny ICMP packets of type 17.**

**Risk factor : Low**

**CVE : CAN-1999-0524**

(link to [Common Vulnerabilities and Exposures](#))

Recommadation:

Use ndd to filter out the above mentioned warnings. Append the following to the end of /etc/init.d/inetinit

```
ndd - set /dev/ip_respond_to_timestamp = 0
ndd - set /dev/ip_respond_to_timestamp_broadcast = 0
```

These settings need more verification and I was unable test them because in worst case it could mean downtime. *dhcp01* is an important infrastructure server that many users are dependent on so no "testing" is allowed in production time. To do this I would have to wait to the next maintanance time-slice. As an alternative, you could set up a new test server with all the needed parameters.

A nice script made by [Andres Kroonmaa](#) is used to list the ndd values of *dhcp01*.

```
#!/bin/ksh
#
# ndd.sh
#
PATH=/usr/sbin:$PATH
if [ -z "$1" ]; then
    echo "Usage: $0 [udp | tcp | ip | icmp | arp | ... ]"
    exit
fi

ndd /dev/$1 '?' | nawk -v c="ndd /dev/$1" '
/write/ {
    split($0,a,/[\t(]/);
    n=c t " " a[1];
    printf "echo %s = ",a[1];
    printf "`%s`\n",n;
}' | sh
```



[Back to top](#)

## Stripping down services in general

In our department we have a policy that on all servers only the services that are needed should run. The rest should be disabled.

Since there is no longer any services running from `/etc/inetd.conf`, it should be removed, or at least renamed so it would start up at boot time.

```
rm /etc/inet/inetd.conf
rm /etc/inetd.conf
mv /etc/rc.2d/S72inetsvc /etc/rc2.d/.NOS72inetsvc
```

Here there are a lot of unnecessary services running, these should be disabled. They are not needed and many of them are considered to be a high security risk.

- dtlogin
- powerd
- rpcbind
- keyserv
- automountd

The server is not running NIS and NFS. The services are down to a minimum.

[Back to top](#)

## User access control

There are too many unnecessary users in the `/etc/passwd` and `/etc/shadow` which must be deleted. All other accounts in those files should have their login shell changed to `/dev/null`.

The accounts that should be deleted:

- uucp
- nuucp
- listen

As a extra bonus install [Titan](#) and run the 'noshell' script. This script will log every failed attempt to login to the `syslogd`.

[Back to top](#)

## Logging

There are some logging on `dhcp01` but there are still room for improvement.

Most log messages are written to `/var/adm/messages` and the rest are displayed on the console. No one person is actually looking at the console very often so therefore a lot of messages are lost. Some messages

were set up to mail to LOGHOST, which is a machine that does not exist anymore.

I would suggest setting up the following in `/etc/syslog.conf` to ensure that failed attempts to login, reboots and other security features are logged in a file. Also logging in remote to a secure log host in real time would be a good idea.

```
auth.info    /var/log/authlog
```

Create `/var/adm/loginlog` to capture failed logins

It is also a good idea to create the log files with 600 permissions owned by root, and then create a script to rotate the syslog files on a regular basis.

It is a good idea to get the program [swatch](#). This is a program that can be configured to report any extraordinary events based on the information in the syslogd messages and then forward them to a central Log Host.

[Back to top](#)

---

## Third-party software

*dhcp01* is running a DHCP server Cisco Network Registrar (CNR). We are right now running on a old version v. 3.0. The new version is v. 5.0.

About patching, Cisco recomend:

[CNR patch info:](#)

System Requirements

For all OS platforms, Cisco recommends that you install the latest recommended patch cluster (Solaris).

About logging:

CNR has some built in log-rolling facilities. If the server is very busy, log messages could be lost due to the log-rolling. Version 3 does not support a real time login to a remote client. I need to check up on the release notes in version 5 for that feature.

About failover:

The fact that the CNR is running on one server without failover capability makes it very vulnerable to spontaneous 'system break-downs'. I would suggest implementering a failover solution using 3 different architectures Hp, Sun and Windows NT. Then you would not be vulnerable to particular vendor problem. Those servers should be physical placed in different buildings. Most likely one or more server would always be up a running which increases overall availability.

## Backup

'*dhcp01*' is backed up every day. Incremental Monday-Thursday and a level 0 dump on Friday using Solaris 'dump' command. The backup is a part of the big scheduled backup pool. Here large StorageTek Library Storage Modules (Tape robot's) are used for the backup. That means that the tape is not being removed from the robot, before the tape is swapped out of circulation. The tapes are then stored in a local fire safe. At the end of the month the tapes are moved off site.

That's means that the backup tape when still in the robot is vulnerable to some kind of disaster before the swapping of tapes.

There is no verification of data after each backup. Because the backup is so big and is taking a very long time to complete.

Specifically the Cisco Network Registrar's database is a very important file. If that database is found to be corrupt the DHCP would not work. I suggest there should be established some kind of comparison of the backup to the original file.

There should be written a backup policy with incorporates an audit of the backup.

## File Integrity

Right now there is no integrity check on *dhcp01*. I would suggest [Tripwire](#) to be installed. Tripwire creates a database with all files on your system. In that database variant file attributes are stored. Then you can use these attributes to check your system (scripts) to see if anything has been changed. If a change has been made, an alarm can be sent.

## Vulnerabilities

Here is the vulnerability list, from most important to less important:

1. No patches are installed on *dhcp01*.
2. Hardening tools such as YASSP and fix-modes have not been run.
3. The current version of 'ssh' has a buffer overflow exploit.
4. A lot of unnecessary network services are running (rpc based, inetd.conf based etc.)
5. Many unnecessary users are in /etc/passwd and /etc/shadow
6. No one is looking in the logfiles for extraordinary events

7. The company is not enforcing there security policies.
8. All Operators have access to the server room.
9. The DHCP services is a single server with no failover facility.

[Back to top](#)

---

## Recommendations

Here is the recommendations list, from most important to less important:

1. Create a patch policy for *dhcp01* and keep up to date with patches.
2. Install and run YASSP and fix-modes
3. Remove all SUID programs on the server.
4. Clean up unused users */etc/passwd* and */etc/shadow*. Install Titan.
5. Implement all suggested solutions in this report and try to run Nessus again.
6. Try several other security tools (commercial or none commercial products) to verify the vulnerabilities Nessus has found.
7. Set the EEPROM password (Firmware security level).
8. Install swatch to help to report any extraordinary events based on the information in the syslogd messages and forward them to a central Log Host.
9. Create a script to rotate the syslog files on a regular basis.
10. Install and run Tripwire.
11. Use a logbook every time changes happen. Use for trace-ability in case of emergency.
12. Create a backup policy with incorporates an audit of the backup.
13. Build a test environment to vefify all parameters mentioned in this audit, before installing them to the production server *dhcp01*.
14. Implement the DHCP services with failover on 3 different architectures (Hp, Sun and Win NT). Then you would not be vulnerable to particular vendor problem.

[Back to top](#)

---

## Conclusion

To secure the *dhcp01* to an above avarage security level the above mentioned recommendations should be followed. It is possible to make a much more radical analysis, but I have chosen an anlysis level that gives me enough security benefits compared to the time, money and resources my company will spend on this project.

Tighten up security on servers is never a one-time asignment, it is an on-going process which demands resouces and time. You are never finish with security. Every time a additional functionality has been added to the system it is crucial for keeping the security level status quo or better, that the administrators have the necessary resouces and suffcient time to complete these tasks.

[Back to top](#)

---

## Refence list

6.6 Solaris Practicum, Hal Pomeranz

Solaris Security, Step By Step version 1.0

<http://www.sans.org>

<http://www.sans.org/newlook/digests/SAC/solaris.htm>

<http://www.nessus.org>

<http://sunsolve.sun.com>

<http://sunsolve.sun.com/securitypatch>

<http://www.securityfocus.com>

<http://www.yassp.org>

<http://cve.mitre.org>

<http://www.openssh.com>

<http://www.rvs.uni-hannover.de/people/voeckler/tune/EN/tune.html>

<http://www.fish.com/titan>

<http://www.stanford.edu/~atkins/swatch>

<http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/nr/nr3-5/relnote/nrrn35.htm>

<http://www.tripwire.com>

[Back to top](#)

---

Created by : Claus Moerkhoej  
Last Modified: 20 November 2000

# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



<b>SANS 2018</b>	<b>Orlando, FL</b>	<b>Apr 03, 2018 - Apr 10, 2018</b>	<b>Live Event</b>
<b>SANS OnDemand</b>	<b>Online</b>	<b>Anytime</b>	<b>Self Paced</b>
<b>SANS SelfStudy</b>	<b>Books &amp; MP3s Only</b>	<b>Anytime</b>	<b>Self Paced</b>